

# Medium-Sized PLC Instruction Guide



For models of LC\MCVS series

©2020 Leadshine Technology Co., Ltd.

Address: 15-20/F, Block B, Nanshan I Valley, No.3185, Shahe West Road, Nanshan District,  
Shenzhen, Guangdong, 518055, China

Tel: (86)755-26409254

Fax: (86)755-26402718

Web: [www.leadshine.com](http://www.leadshine.com)

Sales: [sales@leadshine.com](mailto:sales@leadshine.com)

Support: [tech@leadshine.com](mailto:tech@leadshine.com)

**Notice**

This manual is for leadshine LCMCAS series products, it mainly introduces the programming instructions of Leadsys Studio software, if you have any doubts about the function application and performance of the product, please consult our technical support staff for assistance.

- ◆ **Thank you for purchasing Leadshine series products**
- ◆ **Please read this manual carefully before operating**
- ◆ **Please keep this manual appropriately**

**Record of Revisions**

Manual Reversion	Data	Description of Release
V0.1	08/23/2023	Initial Release

# Contents

<b>Medium-Sized PLC Instruction Guide</b> .....	1
<b>Notice</b> .....	2
<b>Record of Revisions</b> .....	2
<b>Contents</b> .....	3
<b>1. Instruction Overview</b> .....	8
<b>1.1 Overview</b> .....	8
<b>1.2 Standard Data Type</b> .....	8
<b>2. List of Instructions</b> .....	9
<b>3. Basic instruction</b> .....	17
<b>3.1 Comparisons</b> .....	17
<b>3.1.1 Instruction List</b> .....	17
<b>3.1.2 GT</b> .....	17
<b>3.1.3 LT</b> .....	19
<b>3.1.4 GE</b> .....	20
<b>3.1.5 LE</b> .....	21
<b>3.1.6 EQ</b> .....	22
<b>3.1.7 NE</b> .....	23
<b>3.2 Option</b> .....	24
<b>3.2.1 Instruction List</b> .....	24
<b>3.2.2 SEL</b> .....	24
<b>3.2.3 MUX</b> .....	26
<b>3.2.4 MAX</b> .....	27
<b>3.2.5 MIN</b> .....	28
<b>3.2.6 LIMIT</b> .....	29
<b>3.3 Count</b> .....	30
<b>3.3.1 Instruction List</b> .....	30
<b>3.3.2 CTD</b> .....	30
<b>3.3.3 CTU</b> .....	32
<b>3.3.4 CTUD</b> .....	34
<b>3.4 Timers</b> .....	36
<b>3.4.1 Instruction List</b> .....	36
<b>3.4.2 TP</b> .....	36
<b>3.4.3 TON</b> .....	38
<b>3.4.4 TOF</b> .....	39
<b>3.4.5 RTC</b> .....	41
<b>3.5 Bits and Logic Words</b> .....	42
<b>3.5.1 Instruction List</b> .....	42
<b>3.5.2 AND</b> .....	42
<b>3.5.3 OR</b> .....	43
<b>3.5.4 NOT</b> .....	44
<b>3.5.5 XOR</b> .....	45
<b>3.5.6 SR</b> .....	46
<b>3.5.7 RS</b> .....	47
<b>3.5.8 R_TRIG</b> .....	48
<b>3.5.9 F_TRIG</b> .....	49
<b>3.6 Data shifting</b> .....	50
<b>3.6.1 Instruction List</b> .....	50
<b>3.6.2 SHR</b> .....	50
<b>3.6.3 SHL</b> .....	52
<b>3.6.4 ROR</b> .....	53
<b>3.6.5 ROL</b> .....	54
<b>3.7 Data type conversion instruction</b> .....	55
<b>3.7.1 Instruction List</b> .....	55

3.7.2	BOOL_TO_<TYPE>	55
3.7.3	BYTE_TO_<TYPE>	57
3.7.4	<Integer data>_TO_<TYPE>	58
3.7.5	<Real data>_TO_<TYPE>	59
<b>3.8</b>	<b>Data processing instructions</b>	<b>60</b>
3.8.1	Instruction List	60
3.8.2	MOVE	60
3.8.3	HEXinASCII_TO_BYTE	61
3.8.4	BYTE_TO_HEXinASCII	62
3.8.5	WORD_AS_STRING	63
3.8.6	BYTE_TO_HEXSTRING	64
3.8.7	WORD_TO_HEXSTRING	65
3.8.8	DWORD_TO_HEXSTRING	66
<b>3.9</b>	<b>Mathematical operation instructions</b>	<b>67</b>
3.9.1	Instruction List	67
3.9.2	ADD	67
3.9.3	SUB	69
3.9.4	MUL	70
3.9.5	DIV	71
3.9.6	MOD	72
3.9.7	ABS	73
3.9.8	SQRT	74
3.9.9	LN	75
3.9.10	LOG	76
3.9.11	EXP	77
3.9.12	EXPT	78
3.9.13	SIN	79
3.9.14	COS	80
3.9.15	TAN	81
3.9.16	ASIN	82
3.9.17	ACOS	83
3.9.18	ATAN	84
3.9.19	SIZEOF	85
<b>3.10</b>	<b>String</b>	<b>86</b>
3.10.1	Instruction List	86
3.10.2	LEN	86
3.10.3	LEFT	87
3.10.4	RIGHT	88
3.10.5	MID	89
3.10.6	CONCAT	90
3.10.7	INSERT	91
3.10.8	DELETE	92
3.10.9	FIND	93
3.10.10	REPLACE	94
<b>3.11</b>	<b>Address operation</b>	<b>95</b>
3.11.1	Instruction List	95
3.11.2	ADR	95
3.11.3	^	96
3.11.4	BITADR	97
<b>3.12</b>	<b>File Operations</b>	<b>98</b>
3.12.1	Instruction List	98
3.12.2	SysFileOpen	98
3.12.3	SysFileClose	100
3.12.4	SysFileWrite	101
3.12.5	SysFileRead	102
3.12.6	SysFileDelete	103

3.12.7 SysFileCopy.....	104
3.12.8 SysFileRename.....	105
3.12.9 SysFileSetPos.....	106
3.12.10 SysFileGetPos.....	107
3.12.11 SysFileGetSize.....	108
3.13 Analog calculation.....	109
3.13.1 Instruction List.....	109
3.13.2 PD.....	109
3.13.3 PID.....	111
3.13.4 PID_FIXCYCLE.....	113
3.14 BCD code conversion.....	115
3.14.1 Instruction List.....	115
3.14.2 BCD_TO_INT.....	115
3.14.3 INT_TO_BCD.....	116
3.14.4 BCD_TO_BYTE.....	117
3.14.5 BYTE_TO_BCD.....	118
3.14.6 BCD_TO_WORD.....	119
3.14.7 WORD_TO_BCD.....	120
3.14.8 BCD_TO_DWORD.....	121
3.14.9 DWORD_TO_BCD.....	122
3.15 Analog Waveform.....	123
3.15.1 Instruction List.....	123
3.15.2 BLINK.....	123
3.15.3 GEN.....	125
3.15.4 FREQ_MEASURE.....	127
4. Motion instruction.....	129
4.1 Single-axis status monitoring.....	129
4.1.1 Instruction List.....	129
4.1.2 MC_Power.....	129
4.1.3 MC_Reset.....	132
4.1.4 MC_ReadStatus.....	134
4.1.5 MC_ReadAxisError.....	137
4.1.6 MC_ReadParameter.....	139
4.1.7 MC_ReadBoolParameter.....	141
4.1.8 MC_WriteParameter.....	143
4.1.9 MC_WriteBoolParameter.....	145
4.1.10 MC_ReadActualPosition.....	147
4.1.11 MC_ReadActualVelocity.....	149
4.1.12 MC_ReadActualTorque.....	151
4.1.13 MC_SetPosition.....	153
4.1.14 SMC_ReadSetPosition.....	155
4.1.15 SMC_ReadFBError.....	157
4.1.16 SMC_ClearFBError.....	159
4.2 Single-axis motion control.....	160
4.2.1 Instruction List.....	160
4.2.2 MC_Home.....	160
4.2.3 MC_MoveAbsolute.....	163
4.2.4 MC_MoveRelative.....	167
4.2.5 MC_MoveVelocity.....	171
4.2.6 MC_Stop.....	175
4.2.7 MC_Halt.....	178
4.2.8 MC_Jog.....	181
4.2.9 MC_MoveAdditive.....	183
4.2.10 MC_MoveSuperImposed.....	187
4.2.11 MC_PositionProfile.....	191
4.2.12 MC_VelocityProfile.....	195

4.2.13 MC_AccelerationProfile .....	198
4.2.14 SMC_Homing .....	201
4.2.15 SMC_Inch .....	207
4.3 Axis synchronization function .....	210
4.3.1 Instruction List .....	210
4.3.2 MC_GearIn .....	210
4.3.3 MC_GearInPos .....	213
4.3.4 MC_GearOut .....	217
4.3.5 MC_CamTableSelect .....	219
4.3.6 MC_CamIn .....	222
4.3.7 MC_CamOut .....	226
4.3.8 SMC_GetTappetValue .....	228
4.4 Axis synchronization function .....	230
4.4.1 Instruction List .....	230
4.4.2 LS_2AxisLine .....	230
4.4.3 LS_3AxisLine .....	233
4.4.4 LS_4AxisLine .....	236
4.4.5 LS_5AxisLine .....	239
4.4.6 LS_6AxisLine .....	242
4.4.7 LS_2AxisLineA_Ratio .....	245
4.4.8 LS_LineFollow .....	248
4.4.9 LS_2AxisCircle .....	251
4.4.10 LS_3AxisCircle .....	256
4.4.11 LS_2AxisEllipses .....	259
4.4.12 LS_2AxisCircle_Helical .....	263
4.4.13 LS_3AxisMoveSequence .....	267
4.4.14 LS_4AxisMoveSequence .....	271
4.4.15 LS_6AxisMoveSequence .....	275
5. Communications instruction .....	279
5.1 Free Communications TCP .....	279
5.1.1 Instruction List .....	279
5.1.2 TCP_Client .....	279
5.1.3 TCP_Write .....	282
5.1.4 TCP_Read .....	285
5.1.5 TCP_Connection .....	288
5.1.6 TCP_Server .....	291
5.2 Free Communication UDP .....	294
5.2.1 Instruction List .....	294
5.2.2 UDP_Peer .....	294
5.2.3 UDP_Receive .....	297
5.2.4 UDP_Send .....	300
5.3 EtherCAT communication .....	303
5.3.1 Instruction List .....	303
5.3.2 ETC_CO_SdoReadDWord .....	303
5.3.3 ETC_CO_SdoRead4 .....	306
5.3.4 ETC_CO_SdoRead .....	309
5.3.5 ETC_CO_SdoWrite_Dword .....	312
5.3.6 ETC_CO_SdoWrite4 .....	315
5.3.7 ETC_CO_SdoWrite .....	318
5.3.8 IoDrvEtherCAT_Diag .....	321
5.3.9 ETCslave .....	323
5.4 Ethernet-IP communication .....	325
5.4.1 Instruction List .....	325
5.4.2 IoDrvEtherNetIP_Diag .....	325
5.4.3 RemoteAdapter .....	327
5.4.4 Generic_Service .....	329

5.4.5 Get_Attributes_All .....	336
5.4.6 Get_Attribute_Single .....	343
5.4.7 Set_Attributes_All .....	350
5.4.8 Get_Attribute_Single .....	357
<b>6. High-speed IO instruction .....</b>	<b>364</b>
6.1 Pulse axis control .....	364
6.1.1 Instruction List .....	364
6.1.2 LS_Home_P .....	364
6.1.3 LS_MotionControl_P .....	366
6.1.4 LS_ReadAxisPara_P .....	368
6.1.5 LS_ResetAxis_P .....	371
6.2 High-speed counter instruction .....	373
6.2.1 Instruction List .....	373
6.2.2 LS_Counter .....	373
6.2.3 LS_PresetValue .....	375
6.3 LC1000 Local High Speed Counter .....	377
6.3.1 Instruction List .....	377
6.3.2 LS_Counter .....	377
6.3.3 LS_PresetValue .....	380
6.3.4 LS_TouchProbe .....	382
6.3.5 LS_Compare .....	387
6.3.6 LS_CompareStep .....	390
6.3.7 LS_CompareFIFO .....	393
6.3.8 LS_PWM .....	397
6.3.9 LS_EnableInterrupt .....	399
<b>7. Special instruction .....</b>	<b>401</b>
7.1 SD card file operations .....	401
7.1.1 Instruction List .....	401
7.1.2 LS_CopyFromSDCard .....	401
7.1.3 LS_CopyToSDCard .....	403
7.1.4 LS_GetSDCardInformation .....	405
7.2 System instruction .....	407
7.2.1 Instruction List .....	407
7.2.2 LS_GetIpAddress .....	407
7.2.3 LS_SetIpAddress .....	409
7.2.4 GetPLCLoad .....	411
7.2.5 GetPLCVersion .....	413
7.2.6 ColdResetApp .....	415
7.2.7 WarmResetApp .....	416
7.2.8 LS_ReadDintDT .....	417
7.2.9 LS_ReadStringDT .....	419
7.2.10 LS_SetDintDT .....	421
7.2.11 LS_SetStringDT .....	423
Contact us .....	424

# 1. Instruction Overview

## 1.1 Overview

This manual covers all the required commands and programs for LC\MC\S series products, and provides a demonstration of the use of the relevant commands to guide customers in quick use.

## 1.2 Standard Data Type

Form	Data type	Keywords	Bit	Value range
Boolean type	Boolean	BOOL	8	FALSE(0) or TRUE(1)
Integers	Byte	BYTE	8	0~255
	Word	WORD	16	0~65535
	double word	DWORD	32	0~4294967295
	Long word	LWORD	64	0~(2 <sup>64</sup> -1)
	Signed short integer	SINT	8	-128~127
	Unsigned short integer	USINT	8	0~255
	integers	INT	16	-32768~32767
	unsigned integer	UINT	16	0~65535
	double integer	DINT	32	-2147483648~2147483647
	Unsigned double integer	UDINT	32	0~4294967295
	long integer	LINT	64	-2 <sup>63</sup> ~(2 <sup>63</sup> -1)
	unsigned long integer	ULINT	64	0~(2 <sup>64</sup> -1)
Float type	single precision	REAL	32	1.175494351e-38~3.402823466e+38
	double precision	LREAL	64	2.2250738585072014e-308~1.7976931348623158e+308
String	string	STRING	8*N	

## 2. List of Instructions

Instruction Type	Name	FB/FC	Functionality
Comparisons	GT	FC	Greater than >
	LT	FC	Less than <
	GE	FC	Greater than or equal $\geq$
	LE	FC	Less than or equal $\leq$
	EQ	FC	Equal to =
	NE	FC	Not equal $\neq$
Option	SEL	FC	Choose two instruction
	MUX	FC	Choose any one of instruction
	MAX	FC	Maximum values instruction
	MIN	FC	Minimum value instruction
	LIMIT	FC	Limit value instruction
Counter	CTD	FB	Subtraction counter
	CTU	FB	Additive counter
	CTUD	FB	Reversible counter
Timer	TP	FB	Timers
	TON	FB	Power-on delay timer
	TOF	FB	Power-off delay timer
	RTC	FB	Clock
Bits and Logical Words	AND	FC	And &&
	OR	FC	Or
	NOT	FC	Not !
	XOR	FC	Different or
	SR	FB	Priority setting
	RS	FB	Priority reset
	R_TRIG	FC	Rising edge
	F_TRIG	FC	Falling edge
Data shift	SHR	FC	Bit-Right Shift
	SHL	FC	Bit-Left Shift
	ROR	FC	Cyclic right shift
	ROL	FC	Cyclic Left shift
Data type conversion	BOOL_TO_<TYPE>	FC	Boolean type conversion
	BYTE_TO_<TYPE>	FC	Byte type conversion
	WORD_TO_<TYPE>	FC	Word type conversion
	DWORD_TO_<TYPE>	FC	Double word type conversion
	INT_TO_<TYPE>	FC	Integer type conversion
	SINT_TO_<TYPE>	FC	short integer type conversion
	DINT_TO_<TYPE>	FC	long integer type conversion
	UDINT_TO_<TYPE>	FC	Unsigned long integer type conversion
	REAL_TO_<TYPE>	FC	Real type conversion
	STRING_TO_<TYPE>	FC	character type conversion
	TIME_TO_<TYPE>	FC	Clock type conversion
TOD_TO_<TYPE>	FC	Time type conversion	

Instruction Type	Name	FB/FC	Functionality
	DATE_TO_<TYPE>	FC	Date type conversion
	DT_TO_<TYPE>	FC	Date Time type conversion
Data processing	MOVE	FC	Set value
	HEXinASCII_TO_BYTE	FC	ASCII Convert Digital
	BYTE_TO_HEXinASCII	FC	Convert Digital ASCII
	WORD_AS_STRING	FC	Convert Digital string
	BYTE_TO_HEXSTRING	FC	Byte Type to hexadecimal string
	WORD_TO_HEXSTRING	FC	Word Type to hexadecimal string
	DWORD_TO_HEXSTRING	FC	Double word Type to hexadecimal string
Math operation	ADD	FC	Addition
	SUB	FC	Subtractive
	MUL	FC	Multiplication
	DIV	FC	Division
	MOD	FC	Remainder
	ABS	FC	Absolute value
	SQRT	FC	Square root
	LN	FC	Natural logarithm
	LOG	FC	Common logarithm
	EXP	FC	Index
	EXPT	FC	Power index
	SIN	FC	Sine
	COS	FC	Cosine
	TAN	FC	Tangent
	ASIN	FC	Arcsine
	ACOS	FC	Arccosine
	ATAN	FC	Arctangent
String	SIZEOF	FC	Get data size
	LEN	FC	Get string length
	LEFT	FC	String from the left
	RIGHT	FC	Take the string from the right
	MID	FC	Taking strings from the middle
	CONCAT	FC	String concatenation
	INSERT	FC	String Insertion
	DELETE	FC	String Deletion
	FIND	FC	String lookup
REPLACE	FC	String Replacement	
Address operation	ADR	FC	Retrieve an address
	^	FC	Fetch address content
	BITADR	FC	Bit address
File Operations	SysFileOpen	FB	Open file
	SysFileClose	FB	Close file
	SysFileWrite	FB	Write to a file
	SysFileRead	FB	Read file

Instruction Type	Name	FB/FC	Functionality
	SysFileDelete	FB	Delete file
	SysFileCopy	FB	Copy file
	SysFileRename	FB	Rename a file
	SysFileSetPos	FB	Setting the file read/write location
	SysFileGetPos	FB	Read file read/write location
	SysFileGetSize	FB	Get file size
Analog calculation	PD	FB	Proportional Differential Control
	PID	FB	Proportional integral/differential control
	PID_FIXCYCLE	FB	Proportional Integral Differential Control with manually settable cyclic cycles
BCD code conversion	BCD_TO_INT	FC	BCD to Integer
	INT_TO_BCD	FC	Integer to BCD
	BCD_TO_BYTE	FC	BCD to byte
	BYTE_TO_BCD	FC	Byte to BCD
	BCD_TO_WORD	FC	BCD to Word
	WORD_TO_BCD	FC	Word to BCD
	BCD_TO_DWORD	FC	BCD to double word
Analog Waveform	DWORD_TO_BCD	FC	Double word to BCD
	BLINK	FB	Pulse Signal Generator
	GEN	FB	Periodic Signal Generator
Axis state	FREQ_MEASURE	FB	Frequency measurement
	MC_Power	FB	Axis enable
	MC_Reset	FB	Axis reset
	MC_ReadStatus	FB	Read Axis Status
	MC_ReadAxisError	FB	Read Axis Error
	MC_ReadParameter	FB	Read parameters
	MC_ReadBoolParameter	FB	Read Boolean parameter
	MC_WriteParameter	FB	Write parameters
	MC_WriteBoolParameter	FB	Write Boolean parameters
	MC_ReadActualPosition	FB	Read actual position
	MC_ReadActualVelocity	FB	Read actual speed
	MC_ReadActualTorque	FB	Read actual torque
	MC_SetPosition	FB	Set position
	SMC_ReadSetPosition	FB	Read set position
	SMC_ReadFBError	FB	Read historical error messages
SMC_ClearFBError	FC	Clear Historical Error Messages	
SMC_ErrorString	FB	Error Code to Error Message	
Axis motion control	MC_Home	FB	Home
	MC_MoveAbsolute	FB	Absolute motion
	MC_MoveRelative	FB	Relative motion
	MC_MoveVelocity	FB	Constant velocity
	MC_Stop	FB	Stop
	MC_Halt	FB	Halt

Instruction Type	Name	FB/FC	Functionality
	MC_Jog	FB	Jog
	MC_MoveAdditive	FB	Additive motion
	MC_MoveSuperImposed	FB	Super Imposed motion
	MC_PositionProfile	FB	Position Profile
	MC_VelocityProfile	FB	Velocity Profile
	MC_AccelerationProfile	FB	Acceleration Profile
	SMC_Homing	FB	OP homing
	SMC_Inch	FB	Move an inch
Axis synchronization function	MC_GearIn	FB	E-Gear Input
	MC_GearInPos	FB	E-Gear Smooth Coupling
	MC_GearOut	FB	E-Gear Output
	MC_CamTableSelect	FB	E-Cam tappet association
	MC_CamIn	FB	E-cam association
	MC_CamOut	FB	E-cam disengagement
	SMC_GetTappetValue	FB	Read tappet status
Interpolation	LS_2AxisLine	FB	Two axis linear interpolation
	LS_3AxisLine	FB	Three axis linear interpolation
	LS_4AxisLine	FB	Four axis linear interpolation
	LS_5AxisLine	FB	Five axis linear interpolation
	LS_6AxisLine	FB	Six axis linear interpolation
	LS_2AxisLineA_Ratio	FB	Two axis Variable speed linear interpolation
	LS_LineFollow	FB	Follow the motion
	LS_2AxisCircle	FB	Two axis circular interpolation
	LS_3AxisCircle	FB	Three axis circular interpolation
	LS_2AxisEllipses	FB	Two axis elliptic interpolation
	LS_2AxisCircle_Helical	FB	Helical interpolation
	LS_3AxisMoveSequence	FB	Three axis continuous interpolation motion
	LS_4AxisMoveSequence	FB	Four axis continuous interpolation motion
LS_6AxisMoveSequence	FB	Six axis continuous interpolation motion	
Free Communications TCP	TCP_Client	FB	Creating a TCP Client Communication Service
	TCP_Write	FB	TCP Communication data transmission
	TCP_Read	FB	TCP Communication data reception
	TCP_Connection	FB	Create a TCP connection and connect to the server
	TCP_Server	FB	Creating a TCP server-side communication service
Free Communication UDP	UDP_Peer	FB	Creating a UDP Communication Connection
	UDP_Receive	FB	UDP communication data reception
	UDP_Send	FB	UDP communication data sending
EtherCAT communication	ETC_CO_SdoReadDWord	FB	EtherCAT Slave SDO Read
	ETC_CO_SdoRead4	FB	EtherCAT Slave SDO Read
	ETC_CO_SdoRead	FB	EtherCAT Slave SDO Read
	ETC_CO_SdoWrite_Dword	FB	EtherCAT Slave SDO Write
	ETC_CO_SdoWrite4	FB	EtherCAT Slave SDO Write
	ETC_CO_SdoWrite	FB	EtherCAT Slave SDO Write

Instruction Type	Name	FB/FC	Functionality
	IoDrvEtherCAT_Diag	FB	EtherCAT Master Example
	ETCSlave	FB	EtherCAT Master Example
Ethernet-IP communication	IoDrvEtherNetIP_Diag	FB	Ethernet/IP Master Example
	RemoteAdapter	FB	Ethernet/IP Master Example
	Generic_Service	FB	CIP Universal Display Service
	Get_Attributes_All	FB	Get all properties of an object instance
	Get_Attribute_Single	FB	Getting a single property of an object instance
	Set_Attributes_All	FB	Setting all properties of an object instance
	Set_Attribute_Single	FB	Setting a single property of an object instance
Pulse axis control	LS_Home_P	FB	Pulse axis return to original
	LS_MotionControl_P	FB	Pulse Axis Binding
	LS_ReadAxisPara_P	FB	Getting the pulse equivalent value of the pulse axis
	LS_ResetAxis_P	FB	Pulse axis reset
High Speed Counter	LS_Counter	FB	High Speed Counter
	LS_PresetValue	FB	Preset count value
PWM output	LS_PWM_SetVal	FC	PWM Output
	LS_PWM_GetVal	FC	PWM Parameter read
System time	LS_DateAndTime_ReadDint	FB	Read system time
	LS_DateAndTime_ReadString	FB	Read system time
	LS_DateAndTime_SetDint	FB	Set system time
	LS_DateAndTime_SetString	FB	Set system time
Firmware version	LS_PLC_ControllerVer	FC	Read hardware version
	LS_PLC_CPUID	FC	Read CPU ID number
	LS_PLC_HardwareVerInformation	FC	Read firmware FPGA version
	LS_PLC_SoftwareVerInformation	FC	Read firmware ARM version
	LS_PLC_VerInformation	FC	Retrieve all version information of the controller
File operations within UsrData	LS_FileWrite	FB	File Write
	LS_FileWriteAppend	FB	File Append Write
	LS_FileRead	FB	File reading
	LS_FileCopy	FB	File copy
	LS_FileRename	FB	File renaming
	LS_FileDelete	FB	File deletion
	LS_FileDeleteAll	FB	Batch file deletion
	LS_GetDirectoryFile	FB	Getting information about files in a directory
	LS_GetFileInformation	FB	Getting information about a specified file
	LS_UDisk_CopyFromUDisk	FB	Copying files from a USB flash drive
	LS_UDisk_CopyToUDisk	FB	Copying files to a USB flash drive
LS_UDisk_GetDirectoyFile	FB	Getting information about files in the root directory of a USB flash drive	
File manipulation commands within UsrConfig	LS_ConfigFile_Write	FB	File Write
	LS_ConfigFile_WriteAppend	FB	File Append Write
	LS_ConfigFile_Read	FB	File Read

Instruction Type	Name	FB/FC	Functionality
	LS_ConfigFile_Del	FB	File deletion
	LS_ConfigFile_GetDirectoryFile	FB	Get information about all files in a directory
	LS_ConfigFile_CopyFromUDisk	FB	Copying files from a USB flash drive
	LS_ConfigFile_CopyToUDisk	FB	Copying files to a USB flash drive
HMI library variable batch operation	LS_Modbus_SetM	FB	BOOL variable batch set
	LS_Modbus_ResetM	FB	BOOL variable bulk reset
	LS_Modbus_SetByteToM	FB	BYTE variable bulk set
	LS_Modbus_SetDWS	FB	INT variable batch set
	LS_Modbus_SetDWU	FB	UINT variable batch set
	LS_Modbus_SetDDS	FB	DINT Variable Batch Set
	LS_Modbus_SetDDU	FB	UDINT Variable Batch Set
	LS_Modbus_SetDR	FB	REAL Variable Batch Set
RS232 protocol-free communication	LS_RS232_Open	FB	Open the specified 232 port
	LS_RS232_Write	FB	Protocol-free data transmission from port 232
	LS_RS232_Read	FB	Read receive data without protocol from port 232
	LS_RS232_Close	FB	Shut down the specified 232 port command
RS485 protocol-free communication	LS_RS485_Open	FB	Open Specified RS485 Port Command
	LS_RS485_Write	FB	Protocol-free data transmission from RS485 port
	LS_RS485_Read	FB	Protocol-free receive count from RS485 port
	LS_RS485_Close	FB	Close the specified RS485 port
Ethernet communication	LS_TCP_SetIpAddress	FB	Set the network port IP
	LS_TCP_GetIpAddress	FB	Read the IP of the network port
	LS_TCP_SetServer	FB	Set the controller as a TCP server
	LS_TCP_SetClient	FB	Set the controller as a TCP client
	LS_TCP_WriteData	FB	TCP Send Data
	LS_TCP_ReadData	FB	TCP Receive Data
	LS_UDP_SetConnect	FB	Starting UDP/IP services
	LS_UDP_WriteData	FB	UDP Send Data
	LS_UDP_ReadData	FB	UDP Receive Data
LS interpolation instruction	LS_2AxisLineAbs	FB	Two axis absolute linear interpolation
	LS_2AxisLineRel	FB	Two axis Relative linear interpolation
	LS_3AxisLineAbs	FB	Three axis absolute linear interpolation
	LS_3AxisLineRel	FB	Three axis Relative linear interpolation
	LS_4AxisLineAbs	FB	Four axis absolute linear interpolation
	LS_4AxisLineRel	FB	Four axis Relative linear interpolation
	LS_5AxisLineAbs	FB	Five axis absolute linear interpolation
	LS_5AxisLineRel	FB	Five axis Relative linear interpolation
	LS_6AxisLineAbs	FB	Six axis absolute linear interpolation
	LS_6AxisLineRel	FB	Six axis Relative linear interpolation
	LS_2AxisLineAbs_Ratio	FB	Two axis variable speed absolute linear interpolation

Instruction Type	Name	FB/FC	Functionality
	LS_2AxisLineRel_Ratio	FB	Two axis variable speed relative linear interpolation
	LS_LineFollowAbs	FB	Absolutely follow
	LS_LineFollowRel	FB	Relative follow
	LS_2AxisCircleAbs	FB	Two axis absolute arc interpolation
	LS_2AxisCircleRel	FB	Two axes Relative arc interpolation
	LS_3AxisCircleAbs	FB	Three axis absolute arc interpolation
	LS_3AxisCircleRel	FB	Three axis relative arc interpolation
	LS_2AxisEllipsesAbs	FB	Two axis absolute elliptic interpolation
	LS_2AxisEllipsesRel	FB	Two axes Relative elliptic interpolation
	LS_2AxisCircleAbs_Helical	FB	Absolute Helical Interpolation
	LS_2AxisCircleRel_Helical	FB	Relative Helical Interpolation
	LS_3AxisGCode	FB	Three axis G-code interpolation
	LS_3AxisGCode_File	FB	Three axis G-code interpolation file
	LS_4AxisGCode	FB	Four axis G-code interpolation
	LS_4AxisGCode_File	FB	Four axis G-code interpolation file
	LS_4AxisGCodeAxisP	FB	Four axis G-code interpolation -P
	LS_4AxisGCodeAxisP_File	FB	Four axis G-code interpolation file -P
	LS_6AxisGCodeAxisUVW_file	FB	Six axis G-code interpolation file
	LS_6Axis_ZeroOffset	FB	Six axis zero offset
	LS_3AxisMoveSequence	FB	Three axis continuous interpolation motion
LS_4AxisMoveSequence	FB	Four-axis continuous interpolation motion	
LS_6AxisMoveSequence	FB	Six-axis continuous interpolation motion	
LS axis motion control commands	LS_EtherCATAxis_HomeMove	FB	Home(Drive Home)
	LS_MoveAbsChangePosVel	FB	Real time speed change position
	LS_MoveChangeVel	FB	Real time speed change
	LS_MoveAbs	FB	Single-axis jump speed absolute motion
	LS_MoveRel	FB	Single-axis jump speed relative motion
	LS_MoveVel	FB	Single-axis jump speed constant velocity motion
	LS_ResetVirtualAxis	FB	Single-axis reset function
Pulse control	LS_PulseAxis_MachineState	FC	Axis pulse output control
	LS_PulseAxis_SetPulseMode	FC	Set the axis pulse mode
	LS_PulseAxis_GetPulseMode	FC	Read axis pulse mode
	LS_PulseAxis_SetSpecialIOLogicalState	FC	Set axis special IO signals
	LS_PulseAxis_GetSpecialIOLogicalState	FC	Read axis special IO signals
	LS_PulseAxis_GetSpecialIOState	FC	Get axis specific IO status
	LS_PulseAxis_SetRatio	FC	Set axis E-Gear ratio
	LS_PulseAxis_ReadRatio	FC	Read axis E-Gear ratio
Encoder	LS_Encoder_EzClearSet	FC	EZ Zeroing parameter set

<b>Instruction Type</b>	<b>Name</b>	<b>FB/FC</b>	<b>Functionality</b>
	LS_Encoder_EzClearRead	FC	EZ Clear Configuration Read
	LS_Encoder_EzSetFilterTime	FC	EZ Filter Parameter Set
	LS_Encoder_EzGetFilterTime	FC	EZ Filter parameter read
	LS_Encoder_SetVal	FC	Set the encoder value
	LS_Encoder_GetVal	FC	Read encoder values
	LS_Encoder_SetWorkMode	FC	Set the encoder work mode
	LS_Encoder_SetAxisMove	FC	Encoder value conversion axis motion
SD card file operations	LS_CopyFromSDCard	FB	Copy files from SD card to local
	LS_CopyToSDCard	FB	Copy local files to SD card
	LS_GetSDCardInformation	FB	Get information about files on the SD card
System command	LS_GetIpAddress	FB	Get PLC IP address
	LS_SetIpAddress	FB	Set PLC IP address
	GetPLCLoad	FC	Get PLC load factor
	GetPLCVersion	FC	Get PLC firmware version
	ColdResetApp	FC	Cold reset PLC
	WarmResetApp	FC	Thermal Reset PLC
	LS_ReadDintDT	FB	Get system time (Dint type)
	LS_ReadStringDT	FB	Get system time (String type)
	LS_SetDintDT	FB	Set system time (Dint type)
LS_SetStringDT	FB	Set system time(Dint type)	

## 3. Basic instruction

### 3.1 Comparisons

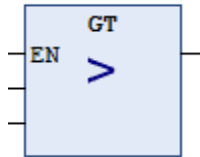
#### 3.1.1 Instruction List

Instruction Type	Name	FB/FC	Functionality
Comparisons	GT	FC	Greater than >
	LT	FC	Less than <
	GE	FC	Greater than or equal ≥
	LE	FC	Less than or equal ≤
	EQ	FC	Equal =
	NE	FC	Not equal ≠

#### 3.1.2 GT

Judge the size of the two input values, when Input 1 is greater than Input 2 output TRUE, otherwise output FALSE.

##### ⊙ Command Format

Instruction	Name	FB/FC	LD	ST	File
GT	Greater than	FC		Out: =In1>In2;	-

##### ⊙ Related Variables

###### Input variable

Input variable	Name	Data type	Range	initialization	Descriptive
In 1	Input 1	All	Compliance data type	0	Data 1
In 2	Input 2	All	Compliance data type	0	Data 2

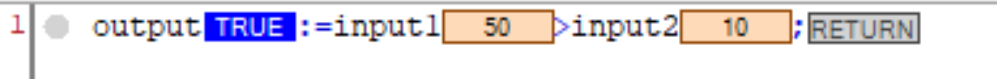
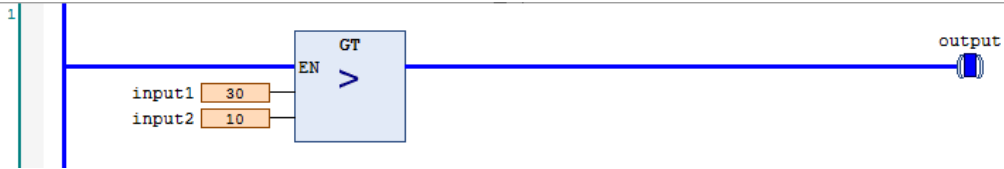
###### Output variable

Output variable	Name	Data type	Range	initialization	Descriptive
Out	return value	BOOL	TRUE-FALSE	FALSE	Comparative results

	Boolean	Bit string					Integer							Real number		Moment, Duration, Date, String				
	BOOL	BYTE	WORD	DWORD	LWORD	USINT	UINT	UDINT	ULINT	SINT	INT	DINT	LINT	REAL	LREAL	TIME	DATE	TOD	DT	STRING
In1	√	√	√	√	√	√	√	√	√	√	√	√	√	√	√	√	√	√	√	√
In2	√	√	√	√	√	√	√	√	√	√	√	√	√	√	√	√	√	√	√	√
Out	√	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-

**⊙ Program demo**

If Input\_1 is greater than Input\_2, output TRUE, otherwise output FALSE.

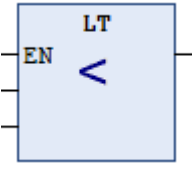
**ST:****LD:****⊙ Notes**

When the Input data type is inconsistent, the compilation will report an error.

### 3.1.3 LT

Judge the size of the two input values, when Input 1 is smaller than Input 2 output TRUE, otherwise output FALSE.

#### ⊙ Command Format

Instruction	Name	FB/FC	LD	ST	File
LT	Less than	FC		Out: =In1<In2:	-

#### ⊙ Related Variables

##### Input variable

Input variable	Name	Data type	Range	initialization	Descriptive
In 1	Input 1	All	Compliance data type	0	Data 1
In 2	Input 2	All	Compliance data type	0	Data 2

##### Output variable

Output variable	Name	Data type	Range	initialization	Descriptive
Out	return value	BOOL	TRUE-FALSE	FALSE	Comparative results

	Boolean	Bit string					Integer							Real number		Moment, Duration, Date, String				
	BOOL	BYTE	WORD	DWORD	LWORD	USINT	UINT	UDINT	ULINT	SINT	INT	DINT	LINT	REAL	LREAL	TIME	DATE	TOD	DT	STRING
In1	√	√	√	√	√	√	√	√	√	√	√	√	√	√	√	√	√	√	√	√
In2	√	√	√	√	√	√	√	√	√	√	√	√	√	√	√	√	√	√	√	√
Out	√	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-

#### ⊙ Program demo

If Input\_1 is less than Input\_2, output TRUE, otherwise output FALSE.

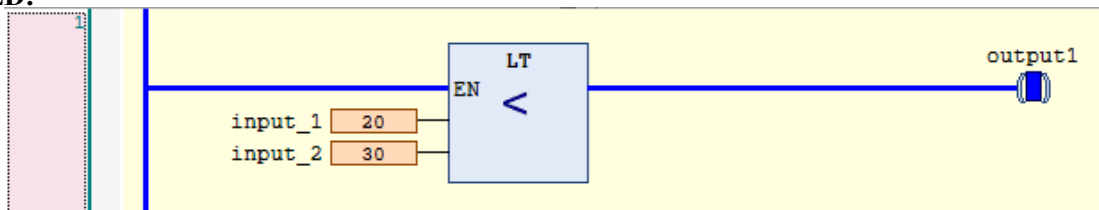
ST:

```

1 | ● POU();
2 | ● output1 TRUE :=input_1 20 <input_2 30 ;RETURN

```

LD:



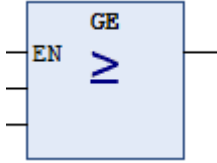
#### ⊙ Notes

When the Input data type is inconsistent, the compilation will report an error.

### 3.1.4 GE

Judge the size of the two input values, when Input 1 is greater than or equal Input 2 output TRUE, otherwise output FALSE.

#### ⊙ Command Format

Instruction	Name	FB/FC	LD	ST	File
GE	Greater than or equal	FC		Out: =In1>=In2:	-

#### ⊙ Related Variables

##### Input variable

Input variable	Name	Data type	Range	initialization	Descriptive
In 1	Input 1	All	Compliance data type	0	Data 1
In 2	Input 2	All	Compliance data type	0	Data 2

##### Output variable

Output variable	Name	Data type	Range	initialization	Descriptive
Out	return value	BOOL	TRUE-FALSE	FALSE	Comparative results

	Boolean	Bit string				Integer								Real number		Moment, Duration, Date, String				
	BOOL	BYTE	WORD	DWORD	LWORD	USINT	UINT	UDINT	ULINT	SINT	INT	DINT	LINT	REAL	LREAL	TIME	DATE	TOD	DT	STRING
In1	√	√	√	√	√	√	√	√	√	√	√	√	√	√	√	√	√	√	√	√
In2	√	√	√	√	√	√	√	√	√	√	√	√	√	√	√	√	√	√	√	√
Out	√	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-

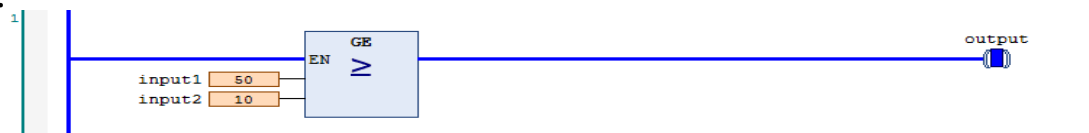
#### ⊙ Program demo

If Input\_1 is Greater than or equal Input\_2, output TRUE, otherwise output FALSE.

ST:

```
1 | output TRUE :=input1 50 >=input2 10 ;RETURN
```

LD:



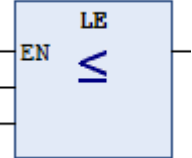
#### ⊙ Notes

When the Input data type is inconsistent, the compilation will report an error.

### 3.1.5 LE

Judge the size of the two input values, when Input 1 is less than or equal to Input 2 output TRUE, otherwise output FALSE.

#### ⊙ Command Format

Instruction	Name	FB/FC	LD	ST	File
LE	Less than or equal	FC		Out: =In1<=In2:	-

#### ⊙ Related Variables

##### Input variable

Input variable	Name	Data type	Range	initialization	Descriptive
In 1	Input 1	All	Compliance data type	0	Data 1
In 2	Input 2	All	Compliance data type	0	Data 2

##### Output variable

Output variable	Name	Data type	Range	initialization	Descriptive
Out	return value	BOOL	TRUE-FALSE	FALSE	Comparative results

	Boolean	Bit string				Integer							Real number		Moment, Duration, Date, String					
	BOOL	BYTE	WORD	DWORD	LWORD	USINT	UINT	UDINT	ULINT	SINT	INT	DINT	LINT	REAL	LREAL	TIME	DATE	TOD	DT	STRING
In1	√	√	√	√	√	√	√	√	√	√	√	√	√	√	√	√	√	√	√	√
In2	√	√	√	√	√	√	√	√	√	√	√	√	√	√	√	√	√	√	√	√
Out	√	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-

#### ⊙ Program demo

If Input\_1 is Less than or equal Input\_2, output TRUE, otherwise output FALSE.

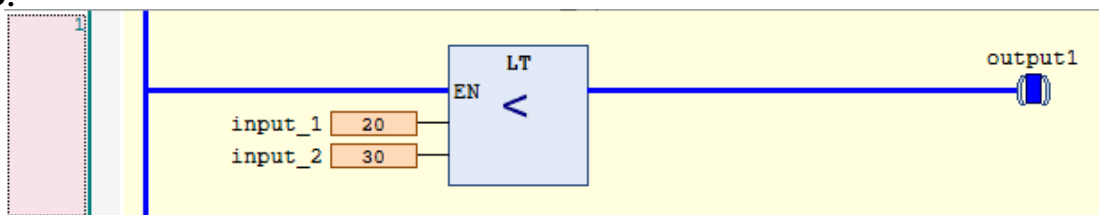
ST:

```

1 | ● POU();
2 | ● output1 TRUE :=input_1 20 <input_2 30 ;RETURN

```

LD:



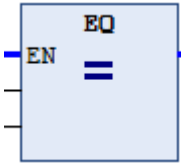
#### ⊙ Notes

When the Input data type is inconsistent, the compilation will report an error.

### 3.1.6 EQ

Judge the size of the two input values, when Input 1 is equal Input 2, output TRUE, otherwise output FALSE.

#### ⊙ Command Format

Instruction	Name	FB/FC	LD	ST	File
EQ	Equal	FC		Out: =In1=In2:	-

#### ⊙ Related Variables

##### Input variable

Input variable	Name	Data type	Range	initialization	Descriptive
In 1	Input 1	All	Compliance data type	0	Data 1
In 2	Input 2	All	Compliance data type	0	Data 2

##### Output variable

Output variable	Name	Data type	Range	initialization	Descriptive
Out	return value	BOOL	TRUE-FALSE	FALSE	Comparative results

	Boolean	Bit string				Integer							Real number		Moment, Duration, Date, String					
	BOOL	BYTE	WORD	DWORD	LWORD	USINT	UINT	UDINT	ULINT	SINT	INT	DINT	LINT	REAL	LREAL	TIME	DATE	TOD	DT	STRING
In1	√	√	√	√	√	√	√	√	√	√	√	√	√	√	√	√	√	√	√	√
In2	√	√	√	√	√	√	√	√	√	√	√	√	√	√	√	√	√	√	√	√
Out	√	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-

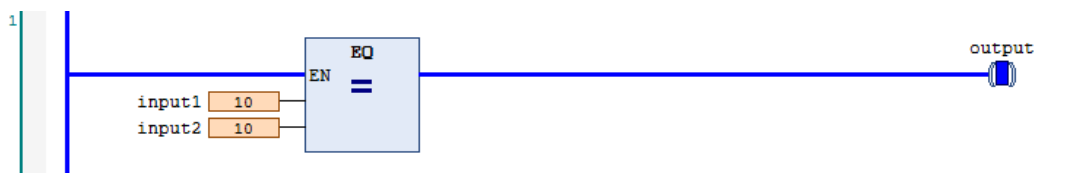
#### ⊙ Program demo

If Input\_1 is equal Input\_2, output TRUE, otherwise output FALSE.

ST:

```
1 output TRUE :=input1 10 =input2 10 ;RETURN
```

LD:



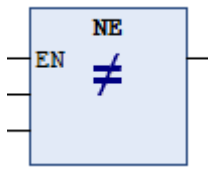
#### ⊙ Notes

When the Input data type is inconsistent, the compilation will report an error.

### 3.1.7 NE

Judge the size of the two input values, when Input 1 is Not equal Input 2, output TRUE, otherwise output FALSE.

#### ⊙ Command Format

Instruction	Name	FB/FC	LD	ST	File
NE	Not equal	FC		Out: =In1≠In2:	-

#### ⊙ Related Variables

##### Input variable

Input variable	Name	Data type	Range	initialization	Descriptive
In 1	Input 1	All	Compliance data type	0	Data 1
In 2	Input 2	All	Compliance data type	0	Data 2

##### Output variable

Output variable	Name	Data type	Range	initialization	Descriptive
Out	return value	BOOL	TRUE-FALSE	FALSE	Comparative results

	Boolean	Bit string				Integer							Real number		Moment, Duration, Date, String					
	BOOL	BYTE	WORD	DWORD	LWORD	USINT	UINT	UDINT	ULINT	SINT	INT	DINT	LINT	REAL	LREAL	TIME	DATE	TOD	DT	STRING
In1	√	√	√	√	√	√	√	√	√	√	√	√	√	√	√	√	√	√	√	√
In2	√	√	√	√	√	√	√	√	√	√	√	√	√	√	√	√	√	√	√	√
Out	√	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-

#### ⊙ Program demo

If Input\_1 is Not equal Input\_2, output TRUE, otherwise output FALSE.

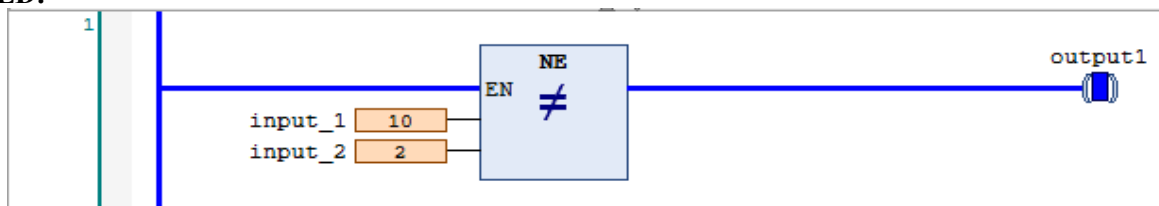
ST:

```

1 | ● POU();
2 | ● output1 TRUE :=input_1 10 <>input_2 2 ;RETURN

```

LD:



#### ⊙ Notes

When the Input data type is inconsistent, the compilation will report an error.

## 3.2 Option

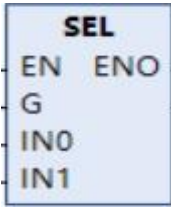
### 3.2.1 Instruction List

Instruction Type	Name	FB/FC	Functionality
Option	SEL	FC	Choose two instruction
	MUX	FC	Choose any one of instruction
	MAX	FC	Maximum values instruction
	MIN	FC	Minimum value instruction
	LIMIT	FC	Limit value instruction

### 3.2.2 SEL

Select one of the two operands. G determines whether IN0 or IN1 is the output.

#### ⊙ Command Format

Instruction	Name	FB/FC	LD	ST	File
SEL	Choose two instruction	FC		SEL	-

#### ⊙ Related Variables

##### Input variable

Input variable	Name	Data type	Range	initialization	Descriptive
G	Output control	BOOL	TRUE-FALSE	-	G=FALSE: output IN0; G=TRUE: output IN1
In 0	Input 0	All	-	-	Data 1
In 1	Input 1	All	-	-	Data 2

##### Output variable

Output variable	Name	Data type	Range	initialization	Descriptive
Output	Selection results	All	-	0	Comparative results

	Boolean	Bit string				Integer							Real number		Moment, Duration, Date, String					
	BOOL	BYTE	WORD	DWORD	LWORD	USINT	UINT	UDINT	ULINT	SINT	INT	DINT	LINT	REAL	LREAL	TIME	DATE	TOD	DT	STRING
G	√	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
In1	√	√	√	√	√	√	√	√	√	√	√	√	√	√	√	√	√	√	√	√
In2	√	√	√	√	√	√	√	√	√	√	√	√	√	√	√	√	√	√	√	√
Out	√	√	√	√	√	√	√	√	√	√	√	√	√	√	√	√	√	√	√	√

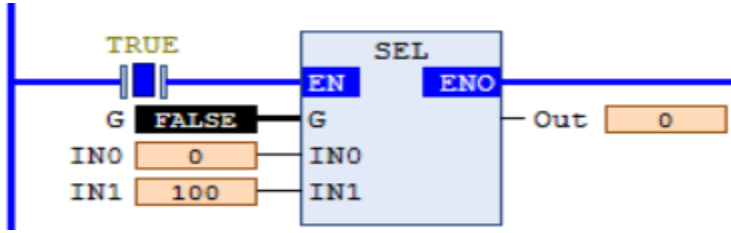
#### ⊙ Program demo

A set of data IN0=0, IN1=100. when G=True, the output is 100.

ST:

```
iOut 100 := SEL( G TRUE, IN0 0, IN1 100 );
```

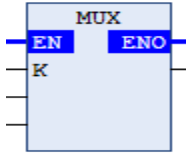
LD:



### 3.2.3 MUX

Selects the Kth number from a set of numbers to output. When K=0, if K is larger than the other inputs, then output passes the last value.

#### ⊙ Command Format

Instruction	Name	FB/FC	LD	ST	File
MUX	multiple choice instruction	FC		MUX	-

#### ⊙ Related Variables

##### Input variable

Input variable	Name	Data type	Range	initialization	Descriptive
K	Index	All	-	-	Selected digits
In 0	Comparator 1	All	-	-	Data 1
In 1	Comparator 2	All	-	-	Data 2
...	Comparator...	All	-	-	Data ...

##### Output variable

Output variable	Name	Data type	Range	initialization	Descriptive
Output	Selection results	All	-	0	Output the result after selection

	Boolean	Bit string				Integer							Real number		Moment, Duration, Date, String					
	BOOL	BYTE	WORD	DWORD	LWORD	USINT	UINT	UDINT	ULINT	SINT	INT	DINT	LINT	REAL	LREAL	TIME	DATE	TOD	DT	STRING
G	√	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
In1	√	√	√	√	√	√	√	√	√	√	√	√	√	√	√	√	√	√	√	√
In2	√	√	√	√	√	√	√	√	√	√	√	√	√	√	√	√	√	√	√	√
Out	√	√	√	√	√	√	√	√	√	√	√	√	√	√	√	√	√	√	√	√

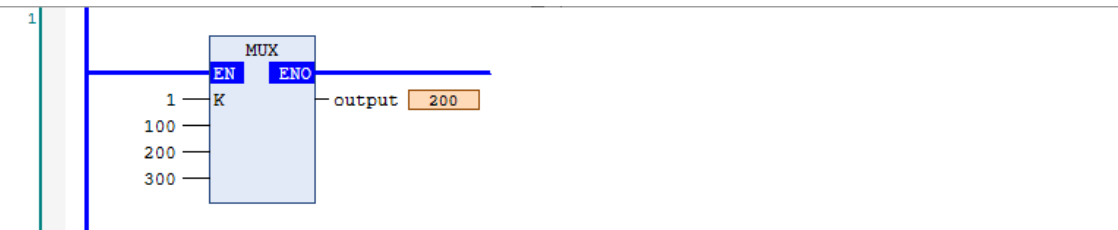
#### ⊙ Program demo

Set a set of data 100, 200, 300. when K=1, the output is 200.

ST:

```
1 output 200 :=MUX(1,100,200,300);
2 RETURN
```

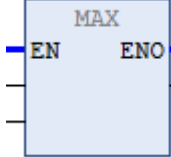
LD:



### 3.2.4 MAX

Take the maximum value of the input value, compare between the two input values, output the maximum value, eg: input1 and input2 are compared, the output takes the maximum value between the two.

#### ⊙ Command Format

Instruction	Name	FB/FC	LD	ST	File
MAX	Maximum value instruction	FC		MAX	-

#### ⊙ Related Variables

##### Input variable

Input variable	Name	Data type	Range	initialization	Descriptive
In 1	Comparator 1	All	-	0	Comparator Data 1
In 2	Comparator 2	All	-	0	Comparator Data 2

##### Output variable

Output variable	Name	Data type	Range	initialization	Descriptive
Output	maximum number	All	-	0	Output the largest of the two input numbers

	Boolean	Bit string				Integer							Real number		Moment, Duration, Date, String					
	BOOL	BYTE	WORD	DWORD	LWORD	USINT	UINT	UDINT	ULINT	SINT	INT	DINT	LINT	REAL	LREAL	TIME	DATE	TOD	DT	STRING
In1	√	√	√	√	√	√	√	√	√	√	√	√	√	√	√	√	√	√	√	√
In2	√	√	√	√	√	√	√	√	√	√	√	√	√	√	√	√	√	√	√	√
Output	√	√	√	√	√	√	√	√	√	√	√	√	√	√	√	√	√	√	√	√

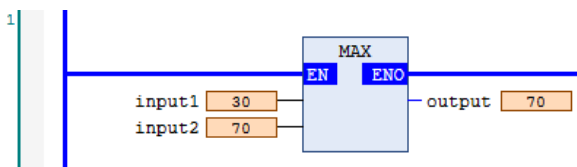
#### ⊙ Program demo

Of the two numbers input1 and input2, Output the largest one.

ST:

```
1 output 70 := (max (input1 30, input2 70)); RETURN
```

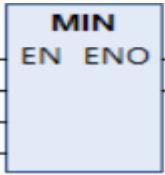
LD:



### 3.2.5 MIN

Takes the minimum of the input values, and after comparing the two inputs, outputs the minimum, eg:IN1 and IN2 are compared, and the output takes the minimum between the two.

#### ⊙ Command Format

Instruction	Name	FB/FC	LD	ST	File
MIN	Minimum value instruction	FC		MIN	-

#### ⊙ Related Variables

##### Input variable

Input variable	Name	Data type	Range	initialization	Descriptive
In 1	Comparator 1	All	-	0	Comparator Data 1
In 2	Comparator 2	All	-	0	Comparator Data 2

##### Output variable

Output variable	Name	Data type	Range	initialization	Descriptive
Output	Minimum number	All	-	0	Output the smallest of the two input numbers

	Boolean	Bit string					Integer							Real number		Moment, Duration, Date, String				
	BOOL	BYTE	WORD	DWORD	LWORD	USINT	UINT	UDINT	ULINT	SINT	INT	DINT	LINT	REAL	LREAL	TIME	DATE	TOD	DT	STRING
In1	√	√	√	√	√	√	√	√	√	√	√	√	√	√	√	√	√	√	√	√
In2	√	√	√	√	√	√	√	√	√	√	√	√	√	√	√	√	√	√	√	√
Output	√	√	√	√	√	√	√	√	√	√	√	√	√	√	√	√	√	√	√	√

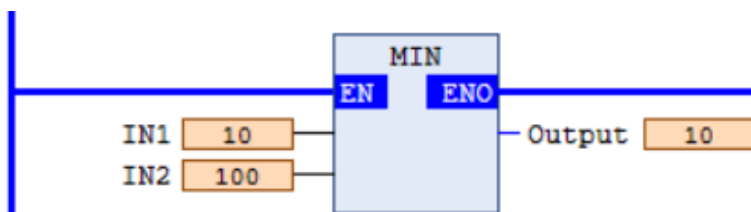
#### ⊙ Program demo

Of the two numbers input1 and input2,Output the smallest one.

ST:

```
Output 10 := MIN (IN1 10 , IN2 100 ) ; RETURN
```

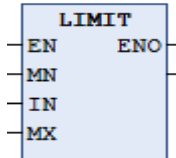
LD:



### 3.2.6 LIMIT

The limit value instruction, MX is the upper limit of the result and MN is the lower limit of the result. If the value of IN exceeds the upper limit of MX, the result is MX, and if the value of IN is less than the lower limit of MN, the result is MN, and if the value of IN is within the range of MN and MX, the result is the value of IN.

#### ⊙ Command Format

Instruction	Name	FB/FC	LD	ST	File
LIMIT	limit value instruction	FC		LIMIT	-

#### ⊙ Related Variables

##### Input variable

Input variable	Name	Data type	Range	initialization	Descriptive
MN	Comparator 1	All			lower value
IN	Input value	All	-	-	Comparator Data
MX	Comparator 2	All	-	-	upper value

##### Output variable

Output variable	Name	Data type	Range	initialization	Descriptive
Output	Output result	All	-	0	Output numerical result after selection

	Boolean	Bit string				Integer							Real number		Moment, Duration, Date, String					
	BOOL	BYTE	WORD	DWORD	LWORD	USINT	UINT	UDINT	ULINT	SINT	INT	DINT	LINT	REAL	LREAL	TIME	DATE	TOD	DT	STRING
MN	√	√	√	√	√	√	√	√	√	√	√	√	√	√	√	√	√	√	√	√
IN	√	√	√	√	√	√	√	√	√	√	√	√	√	√	√	√	√	√	√	√
MX	√	√	√	√	√	√	√	√	√	√	√	√	√	√	√	√	√	√	√	√
Output	√	√	√	√	√	√	√	√	√	√	√	√	√	√	√	√	√	√	√	√

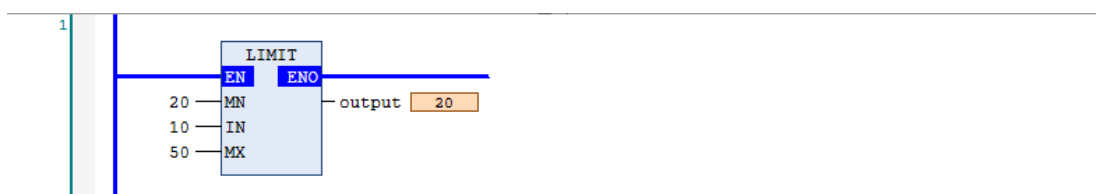
#### ⊙ Program demo

Compare IN value, select output limit value;  $IN > MX$ : Output MX;  $IN < MN$ : Output MN;  
 $MN < IN < MX$ : Output IN.

**ST:**

```
1 output 20 :=LIMIT(20,10,50);RETURN
```

**LD:**



### 3.3 Count

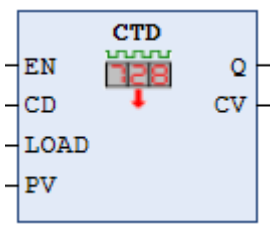
#### 3.3.1 Instruction List

Instruction Type	Name	FB/FC	Functionality
Counters instructions	CTD	FB	Subtraction Counter Set
	CTU	FB	Addition Counter Set
	CTUD	FB	Reversible Counter Set

#### 3.3.2 CTD

Decrementing counter. Preset value variable that decrements by 1 every time a rising edge is detected, and outputs the status TRUE when it reaches 0.

##### ⊙ Command Format

Instruction	Name	FB/FC	LD	ST	File
CTD	Subtraction Counter instruction	FB		CTD(CD:=,LOAD:=,PV:=, Q=>,CV=>);	Standard

##### ⊙ Related Variables

###### Input variable

Input variable	Name	Data type	Range	initialization	Descriptive
CD	Counter Input	BOOL	TRUE-FALSE	FALSE	Counter Input
LOAD	Counter Reset	BOOL	TRUE-FALSE	FALSE	PV assigns a value to the parameter CV
PV	Default value	WORD	0~65535	0	Counter Preset

###### Output variable

Output variable	Name	Data type	Range	initialization	Descriptive
Q	Counter Output	BOOL	TRUE-FALSE	FALSE	Counter output: TRUE: ON FALSE: OFF
CV	Counting real-time values	WORD	0~65535	0	Counting real-time values

	Boolean					Bit string							Integer		Real number		Moment, Duration, Date, String				
	BOOL	BYTE	WORD	DWORD	LWORD	USINT	UINT	UDINT	ULINT	SINT	INT	DINT	LINT	REAL	LREAL	TIME	DATE	TOD	DT	STRING	
CD	√	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	
LOAD	√	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	
PV	-	-	√	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	
Q	√	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	
CV	-	-	√	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	

**⊙ Functional Description**

LOAD is TRUE: the value of PV is transferred to CV and the output Q becomes FALSE.

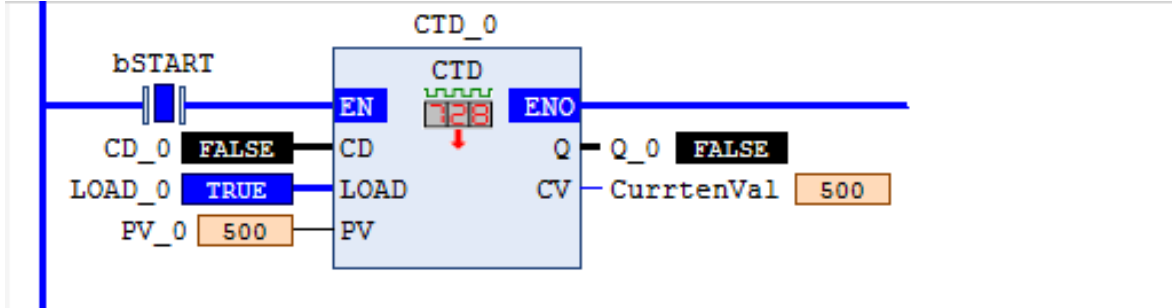
LOAD is FALSE: CD changes from FALSE to TRUE on the rising edge.

When CV is greater than 0, CV is reduced by 1; when CV is equal to 0, Q returns to TRUE.

The value of CV is less than 0, CD has a rising edge signal, and CV is not subtracted.

**⊙ Program demo**
**ST:**

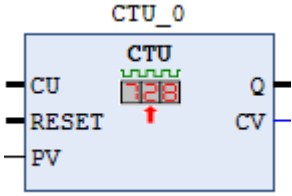
```
CTD_0(CD FALSE := CD_0 FALSE, LOAD TRUE := LOAD_0 TRUE, PV 500 := pv_0 500, Q FALSE => Q_0 FALSE, CV 500 => CV_0 500);RETURN
```

**LD:**


### 3.3.3 CTU

Incremental counter. Preset value variable, incremented by 1 every time a rising edge is detected, output status TRUE when CV value is greater than or equal to PV value.

#### ⊙ Command Format

Instruction	Name	FB/FC	LD	ST	File
CTU	Addition Counter instruction	FB		CTU_0( CU:=start, RESET:=, PV:=, Q=>, CV=> );	Standard

#### ⊙ Related Variables

##### Input variable

Input variable	Name	Data type	Range	initialization	Descriptive
CU	Counter Input	BOOL	TRUE-FALSE	FALSE	Counter Input
RESET	Counter Reset	BOOL	TRUE-FALSE	FALSE	TRUE:CV reset to 0
PV	Default value	WORD	0~65535	0	Counter Preset

##### Output variable

Output variable	Name	Data type	Range	initialization	Descriptive
Q	Counter Output	BOOL	TRUE-FALSE	FALSE	Counter output: TRUE: ON FALSE: OFF
CV	Counting real-time values	WORD	0~65535	0	Counting real-time values

	Boolean	Bit string					Integer						Real number		Moment, Duration, Date, String					
	BOOL	BYTE	WORD	DWORD	LWORD	USINT	UINT	UDINT	ULINT	SINT	INT	DINT	LINT	REAL	LREAL	TIME	DATE	TOD	DT	STRING
CU	√	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
RESET	√	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
PV	-	-	√	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
Q	√	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
CV	-	-	√	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-

#### ⊙ Functional Description

RESET is TRUE: The value of CV becomes 0 and the output Q becomes FALSE.

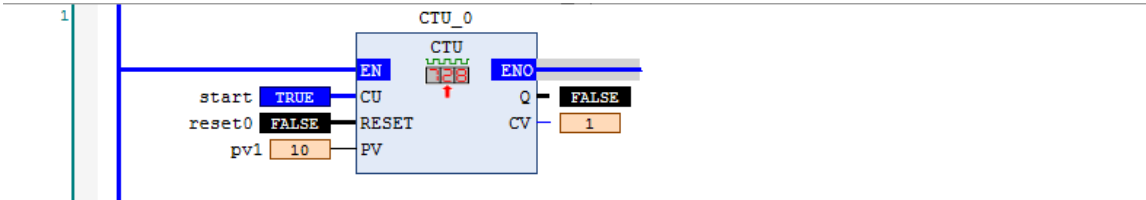
RESET is FALSE: CU has a rising edge and CV is added 1. When the value of CV is greater than the value of PV, Q becomes TRUE.

⊙ **Program demo**

**ST:**

```
1 ● CTU_0(  
2   CU TRUE :=start TRUE,  
3   RESET FALSE :=reset0 FALSE,  
4   PV 10 :=pv1 10,  
5   Q=> ,  
6 ●   CV=> );RETURN
```

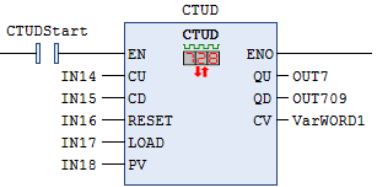
**LD:**



### 3.3.4 CTUD

A counter that performs addition and subtraction operations based on addition counter input and subtraction counter input. The data type of the preset value and count value is WORD.

#### ⊙ Command Format

Instruction	Name	FB/FC	LD	ST	File
CTUD	Reversible Counter Set instruction	FB		CTUD(CU:=, CD:=, RESET:=, LOAD:=, PV:=, QU=>, QD=>, CV=>);	Standard

#### ⊙ Related Variables

##### Input variable

Input variable	Name	Data type	Range	initialization	Descriptive
CU	Addition Counter Input	BOOL	TRUE-FALSE	FALSE	Addition Counter Input
CD	Subtraction Counter Input	BOOL	TRUE-FALSE	FALSE	Subtraction Counter Input
RESET	Counter Reset	BOOL	TRUE-FALSE	FALSE	TRUE:CV reset to 0
LOAD	Counter Reset	BOOL	TRUE-FALSE	FALSE	PV assigns a value to the parameter CV
PV	Default value	WORD	0~65535	0	Counter Preset

##### Output variable

Output variable	Name	Data type	Range	initialization	Descriptive
QU	Addition Counter Output	BOOL	TRUE-FALSE	FALSE	Counter output: TRUE: ON FALSE: OFF
QD	Subtraction Counter Output	BOOL	TRUE-FALSE	FALSE	Counter output: TRUE: ON FALSE: OFF
CV	Counting real-time values	WORD	0~65535	0	Counting real-time values

	Boolean	Bit string				Integer							Real number		Moment, Duration, Date, String					
	BOOL	BYTE	WORD	DWORD	LWORD	USINT	UINT	UDINT	ULINT	SINT	INT	DINT	LINT	REAL	LREAL	TIME	DATE	TOD	DT	STRING
CU	√	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
CD	√	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
RESET	√	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
LOAD	√	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
PV	-	-	√	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
QU	√	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
QD	√	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-

CV	-	-	√	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
----	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---

**⊙ Functional Description**

Addition Counter Function: Usage is the same as CTU;

Subtraction Counter Function: Usage is the same as CTD;

CTD and CTU Common Functions: When LOAD or RESET is TRUE, CV does not operate; when CU and CD rise at the same time, CV does not operate.

When RESET and LOAD are TRUE, RESET takes precedence and CV becomes 0.

RESET is TRUE, CV becomes 0 and QD becomes TRUE.

LOAD is TRUE, CV is equal to PV, QU is TRUE.

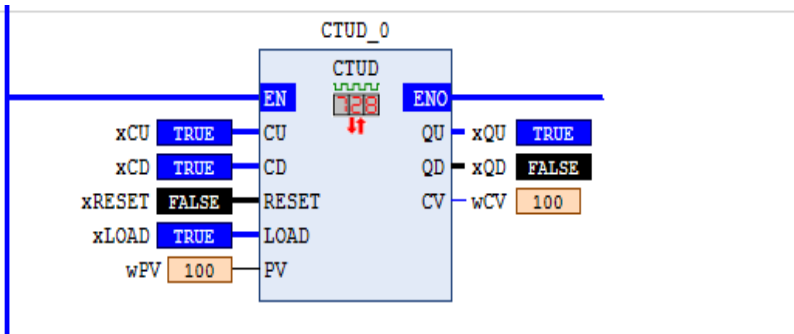
**⊙ Program demo**

**ST:**

```

CTUD_0 (
  CU TRUE :=xCU TRUE ,
  CD TRUE :=xCD TRUE ,
  RESET FALSE :=xRESET FALSE ,
  LOAD TRUE :=xload TRUE ,
  PV 100 :=wPV 100 ,
  QU TRUE =>xQU TRUE ,
  QD FALSE =>xQD FALSE ,
  CV 100 =>wCV 100 );
  
```

**LD:**



### 3.4 Timers

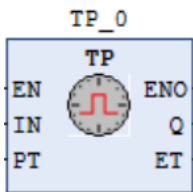
#### 3.4.1 Instruction List

Instruction Type	Name	FB/FC	Functionality
timers instructions	TP	FB	Ordinary timer
	TON	FB	Power-on delay timer
	TOF	FB	Power failure delay timer
	RTC	FB	Real time clock

#### 3.4.2 TP

Input timing, output current timing time and timing status signal.

##### ⊙ Command Format

Instruction	Name	FB/FC	LD	ST	File
TP	Ordinary timer instruction	FB		TP (IN:= , PT:= , Q=> , ET=>);	Standard

##### ⊙ Related Variables

###### Input variable

Input variable	Name	Data type	Range	initialization	Descriptive
IN	Input	BOOL	TRUE-FALSE	FALSE	Start counting
PT	Timekeeping	TIME	0~4294967295	0	Set time

###### Output variable

Output variable	Name	Data type	Range	initialization	Descriptive
Q	Timing complete	BOOL	TRUE-FALSE	FALSE	Timing complete
ET	Timekeeping	TIME	0~4294967295	0	Set time

	Boolean	Bit string					Integer						Real number		Moment, Duration, Date, String					
	BOOL	BYTE	WORD	DWORD	LWORD	USINT	UINT	UDINT	ULINT	SINT	INT	DINT	LINT	REAL	LREAL	TIME	DATE	TOD	DT	STRING
IN	√	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
PT	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	√	-	-	-	-
Q	√	-	√	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
ET	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	√	-	-	-	-

**⊙ Functional Description**

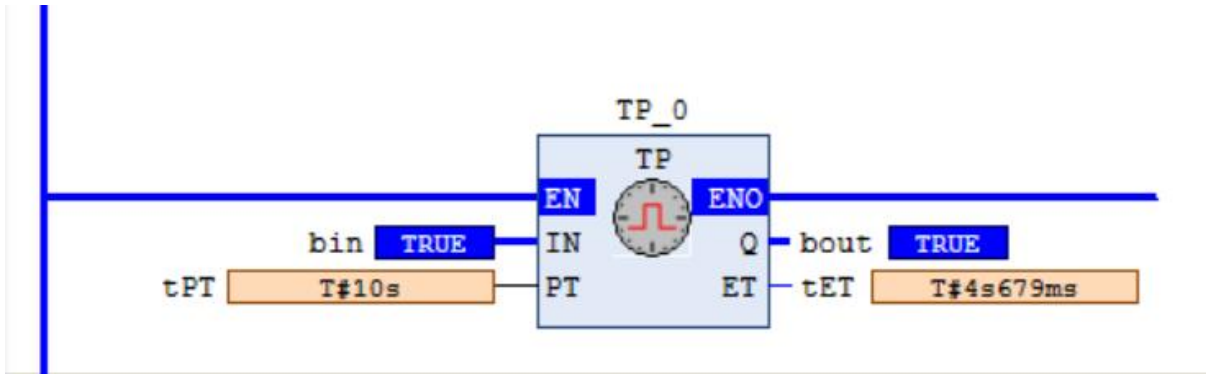
IN is TRUE, Q outputs TRUE for PT time; ET displays current time; ET=PT, Q becomes FALSE.

**⊙ Program demo**
**ST:**

```

1 ● TP_2 (
2   IN TRUE := bIn TRUE ,
3   PT T#5s999ms := tTime T#5s999ms ,
4   Q TRUE => xout TRUE ,
5 ● ET T#2s300ms => tTime_1 T#2s300ms ); RETURN

```

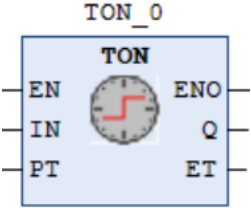
**LD:**

**⊙ Note**

During timing, the PT value can be changed. When the changed PT value is greater than the ET value, the timing continues, and when the changed PT value is less than or equal to the ET value, the timing stops and Q changes to FALSE.

### 3.4.3 TON

Outputs a timing completion signal after the power-on delay time is set.

#### ⊙ Command Format

Instruction	Name	FB/FC	LD	ST	File
TON	Power-on delay timer instruction	FB		TON(IN:=,PT:=,Q=>,ET=>);	Standard

#### ⊙ Related Variables

##### Input variable

Input variable	Name	Data type	Range	initialization	Descriptive
IN	Input	BOOL	TRUE-FALSE	FALSE	Start counting
PT	Timekeeping	TIME	0~4294967295	0	Set time

##### Output variable

Output variable	Name	Data type	Range	initialization	Descriptive
Q	Timing complete	BOOL	TRUE-FALSE	FALSE	Timing complete
ET	Timekeeping	TIME	0~4294967295	0	Set time

	Boolean	Bit string				Integer						Real number		Moment, Duration, Date, String						
	BOOL	BYTE	WORD	DWORD	LWORD	USINT	UINT	UDINT	ULINT	SINT	INT	DINT	LINT	REAL	LREAL	TIME	DATE	TOD	DT	STRING
IN	√	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
PT	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	√	-	-	-	-
Q	√	-	√	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
ET	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	√	-	-	-	-

#### ⊙ Functional Description

IN is TRUE, Q outputs TRUE for PT time; ET displays current time; ET=PT, Q becomes FALSE.

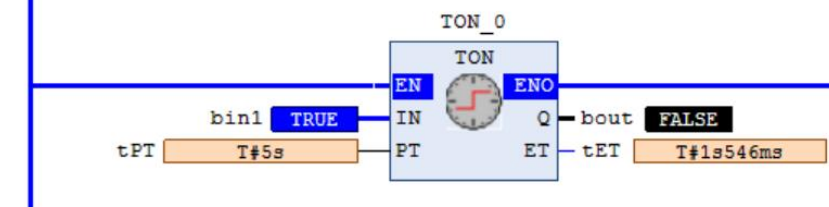
#### ⊙ Program demo

```

ST: 3 ● TON_1(
4   IN TRUE :=bIn1 TRUE ,
5   PT T#5s999ms :=tTime T#5s999ms ,
6   Q FALSE =>xOut FALSE ,
7   ET T#2s656ms =>tTime_1 T#2s656ms ) :RETURN

```

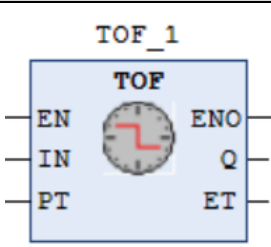
#### LD:



### 3.4.4 TOF

Outputs a timing completion signal after the set time of the power failure delay.

#### ⊙ Command Format

Instruction	Name	FB/FC	LD	ST	File
TOF	Power failure delay timer instruction	FB		TOF(IN:= ,PT:= ,Q=> ,ET=>);	Standard

#### ⊙ Related Variables

##### Input variable

Input variable	Name	Data type	Range	initialization	Descriptive
IN	Input	BOOL	TRUE-FALSE	FALSE	Start counting
PT	Timekeeping	TIME	0~4294967295	0	Set time

##### Output variable

Output variable	Name	Data type	Range	initialization	Descriptive
Q	Timing complete	BOOL	TRUE-FALSE	FALSE	Timing complete
ET	Timekeeping	TIME	0~4294967295	0	Set time

	Boolean	Bit string					Integer						Real number		Moment, Duration, Date, String					
	BOOL	BYTE	WORD	DWORD	LWORD	USINT	UINT	UDINT	ULINT	SINT	INT	DINT	LINT	REAL	LREAL	TIME	DATE	TOD	DT	STRING
IN	√	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
PT	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	√	-	-	-	-
Q	√	-	√	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
ET	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	√	-	-	-	-

#### ⊙ Functional Description

IN is TRUE, Q outputs TRUE for PT time; ET displays current time; ET=PT, Q becomes FALSE.

#### ⊙ Program demo

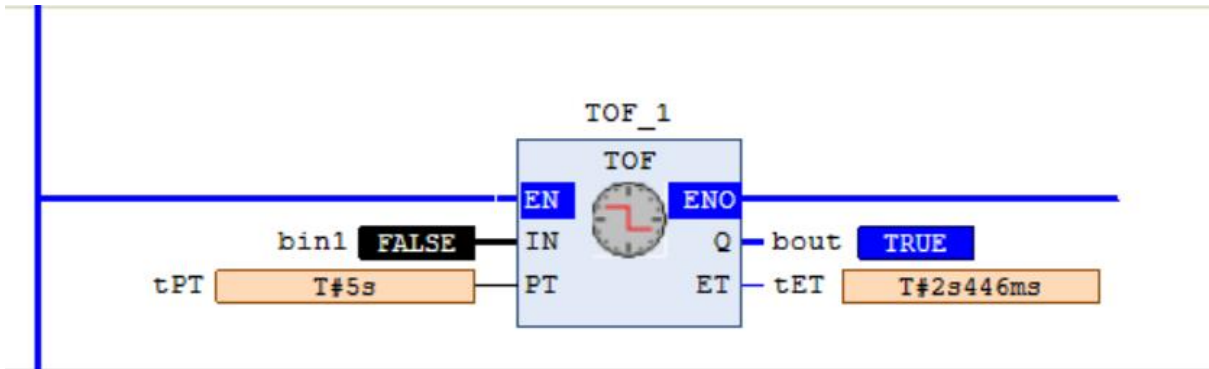
ST:

```

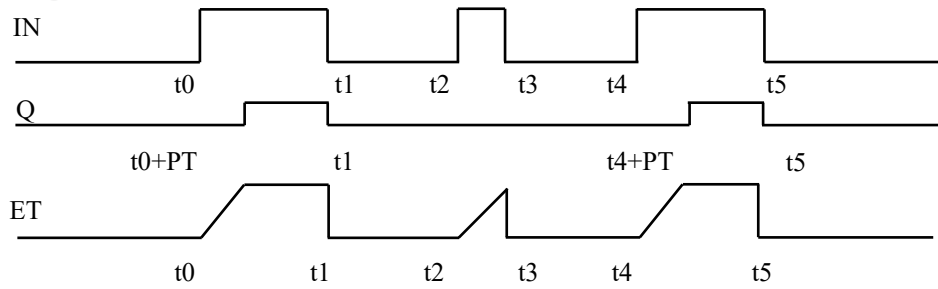
5 ● TOF_1(
6   IN FALSE := bin1 FALSE ,
7   PT T#5s999ms := tTime T#5s999ms ,
8   Q TRUE => xOut TRUE ,
9   ET T#3s761ms => tTime_1 T#3s761ms
10 ● ); RETURN

```

LD:



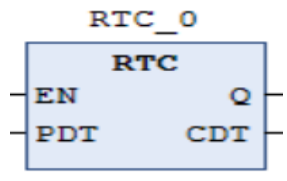
⊙ Timing Diagram



### 3.4.5 RTC

Clock function.

#### ⊙ Command Format

Instruction	Name	FB/FC	LD	ST	File
RTC	Real time clock instruction	FB		RTC(EN:=,PDT:=,Q=>,CDT=>);	Standard

#### ⊙ Related Variables

##### Input variable

Input variable	Name	Data type	Range	initialization	Descriptive
PDT	Real time clock	DATE_AND_TIME	0(1970-01-01,00:00:00)– 4294967295 (210602-07,06:28:15)	FALSE	start counting

##### Output variable

Output variable	Name	Data type	Range	initialization	Descriptive
Q	Timing complete	BOOL	TRUE-FALSE	FALSE	Timing complete
CDT	Timekeeping	DATE_AND_TIME	0(1970-01-01,00:00:00) – 4294967295 (210602-07,06:28:15)	1970-01-01, 00:00:00	Timekeeping

	Boolean	Bit string				Integer						Real number		Moment, Duration, Date, String						
	BOOL	BYTE	WORD	DWORD	LWORD	USINT	UINT	UDINT	ULINT	SINT	INT	DINT	LINT	REAL	LREAL	TIME	DATE	TOD	DT	STRING
PDT	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	√	-
Q	√	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
CDT	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	√	-

#### ⊙ Functional Description

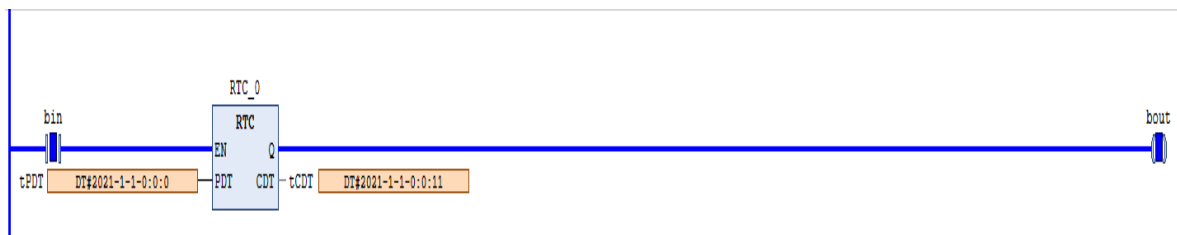
EN is TRUE to time with PDT, output Q is TRUE to output CDT to show current timing time.

#### ⊙ Program demo

```

ST: 7 ● RTC_1 (
8     EN TRUE :=bIn TRUE,
9     PDT DT#2021-1-1-0:0:0 :=tPDT DT#2021-1-1-0:0:0,
10    Q TRUE =>xOut TRUE,
11    CDT DT#2021-1-1-0:0:16 =>tCDT DT#2021-1-1-0:0:16);RETURN
    
```

#### LD:



### 3.5 Bits and Logic Words.

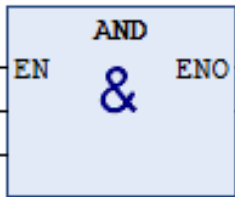
#### 3.5.1 Instruction List

Instruction Type	Name	FB/FC	Functionality
timers instructions	AND	FC	And
	OR	FC	Or
	NOT	FC	Not
	XOR	FC	Exclusive OR
	SR	FB	Bistable set-dominat latch
	RS	FB	Bistable reset-dominat latch
	R_TRIG	FC	Rising edge of a boolean signal
	F_TRIG	FC	Falling edge of a boolean signal

#### 3.5.2 AND

The IEC operator is used for the bitwise AND of bit operands.

##### ⊙ Command Format

Instruction	Name	FB/FC	LD	ST	File
AND	Operands and instruction	FC		AND	-

##### ⊙ Related Variables

###### Input variable

Input variable	Name	Data type	Range	initialization	Descriptive
IN 1	Input 1	-	-	0	Data 1
IN 2	Input 2	-	-	0	Data 2

###### Output variable

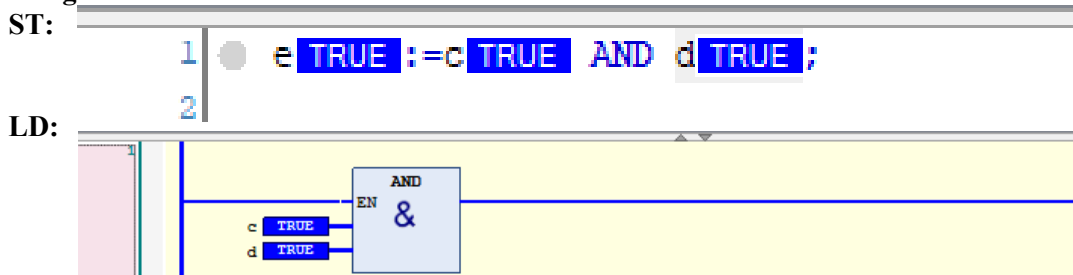
Output variable	Name	Data type	Range	initialization	Descriptive
OUT	Output	-	0	0	Output results

	Boolean	Bit string				Integer							Real number		Moment, Duration, Date, String					
	BOOL	BYTE	WORD	DWORD	LWORD	USINT	UINT	UDINT	ULINT	SINT	INT	DINT	LINT	REAL	LREAL	TIME	DATE	TOD	DT	STRING
IN 1	√	√	√	√	√	-	-	-	-	-	-	-	-	-	-	-	-	-	√	-
IN 2	√	√	√	√	√	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
OUT	√	√	√	√	√	-	-	-	-	-	-	-	-	-	-	-	-	-	√	-

##### ⊙ Functional Description

When the input bits all yield 1, the output bit also yields 1; otherwise 0.

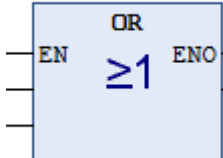
##### ⊙ Program demo



### 3.5.3 OR

The IEC operator is used for the bitwise OR of bit operands.

#### ⊙ Command Format

Instruction	Name	FB/FC	LD	ST	File
OR	Operands or instruction	FC		OR	-

#### ⊙ Related Variables

##### Input variable

Input variable	Name	Data type	Range	initialization	Descriptive
IN 1	Input 1	-	-	0	Data 1
IN 2	Input 2	-	-	0	Data 2

##### Output variable

Output variable	Name	Data type	Range	initialization	Descriptive
OUT	Output	-	0	0	Output results

	Boolean	Bit string					Integer						Real number		Moment, Duration, Date, String					
	BOOL	BYTE	WORD	DWORD	LWORD	USINT	UINT	UDINT	ULINT	SINT	INT	DINT	LINT	REAL	LREAL	TIME	DATE	TOD	DT	STRING
IN 1	√	√	√	√	√	-	-	-	-	-	-	-	-	-	-	-	-	-	√	-
IN 2	√	√	√	√	√	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
OUT	√	√	√	√	√	-	-	-	-	-	-	-	-	-	-	-	-	-	√	-

#### ⊙ Functional Description

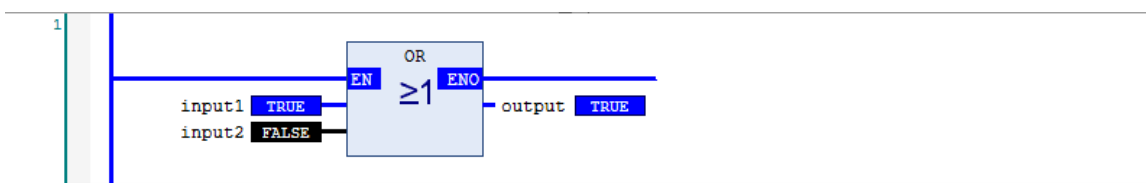
When at least one of the input bits yields 1, the output bit also yields 1; otherwise 0.

#### ⊙ Program demo

ST:

```
1 output TRUE := input1 TRUE OR input2 FALSE ; RETURN
```

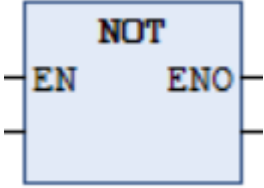
LD:



### 3.5.4 NOT

The IEC operator is used for the bitwise NOT of a bit operand.

#### ⊙ Command Format

Instruction	Name	FB/FC	LD	ST	File
NOT	Operands not instruction	FC		NOT	-

#### ⊙ Related Variables

##### Input variable

Input variable	Name	Data type	Range	initialization	Descriptive
IN 1	Input 1	-	-	0	Data 1

##### Output variable

Output variable	Name	Data type	Range	initialization	Descriptive
OUT	Output	-	0	0	Output results

	Boolean	Bit string				Integer							Real number		Moment, Duration, Date, String					
	BOOL	BYTE	WORD	DWORD	LWORD	USINT	UINT	UDINT	ULINT	SINT	INT	DINT	LINT	REAL	LREAL	TIME	DATE	TOD	DT	STRING
IN 1	√	√	√	√	√	-	-	-	-	-	-	-	-	-	-	-	-	-	√	-
OUT	√	√	√	√	√	-	-	-	-	-	-	-	-	-	-	-	-	-	√	-

#### ⊙ Functional Description

When the respective input bit yields 0, the output bit also yields 1, and the other way around.

#### ⊙ Program demo

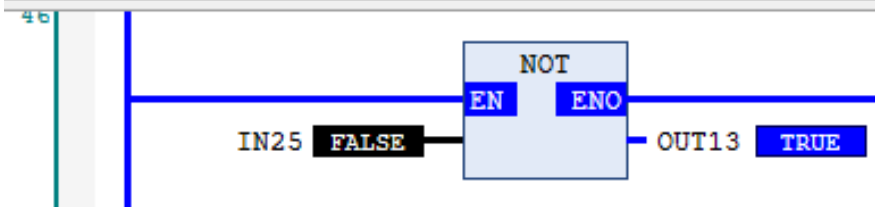
ST:

◆ IN25	BOOL	TRUE			NOT
◆ OUT13	BOOL	FALSE			

```
143 //NOT
144 ● OUT13 FALSE :=NOT (IN25 TRUE) ;
```

LD:

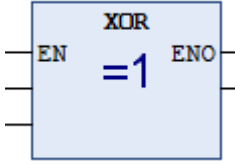
◆ IN25	BOOL	FALSE			NOT
◆ OUT13	BOOL	TRUE			



### 3.5.5 XOR

The IEC operator is used for the bitwise XOR of bit operands.

#### ⊙ Command Format

Instruction	Name	FB/FC	LD	ST	File
XOR	Operands xor instruction	FC		XOR	-

#### ⊙ Related Variables

##### Input variable

Input variable	Name	Data type	Range	initialization	Descriptive
IN 1	Input 1	-	-	0	Data 1
IN 2	Input 2	-	-	0	Data 2

##### Output variable

Output variable	Name	Data type	Range	initialization	Descriptive
OUT	Output	-	0	0	Output results

	Boolean	Bit string				Integer							Real number		Moment, Duration, Date, String					
	BOOL	BYTE	WORD	DWORD	LWORD	USINT	UINT	UDINT	ULINT	SINT	INT	DINT	LINT	REAL	LREAL	TIME	DATE	TOD	DT	STRING
IN 1	√	√	√	√	√	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
IN 2	√	√	√	√	√	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
OUT	√	√	√	√	√	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-

#### ⊙ Functional Description

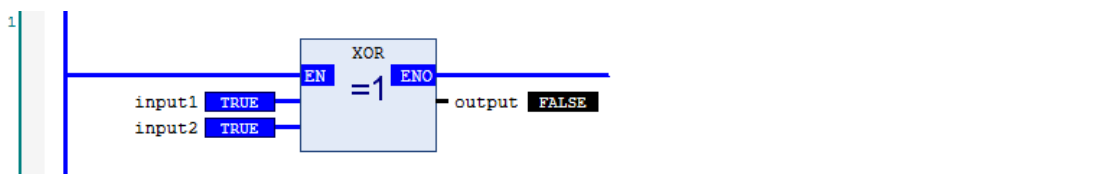
When only one of the two input bits yields 1, the output bit also yields 1. When both inputs yield 1 or 0, then the output yields 0.

#### ⊙ Program demo

ST:

```
1 | ● output FALSE := input1 TRUE XOR input2 TRUE : RETURN
```

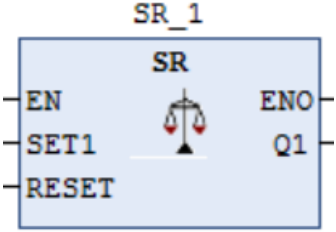
LD:



### 3.5.6 SR

Realizes a bistable set-dominant latch.

#### ⊙ Command Format

Instruction	Name	FB/FC	LD	ST	File
SR	Set latch instruction	FB		SR_1( SET1:=, RESET:=, Q1=>);	-

#### ⊙ Related Variables

##### Input variable

Input variable	Name	Data type	Range	initialization	Descriptive
SET1	SET	BOOL	TRUE-FALSE	FALSE	Data 1
RESET	RESET	BOOL	TRUE-FALSE	FALSE	Data 2

##### Output variable

Output variable	Name	Data type	Range	initialization	Descriptive
Q1	Output	BOOL	TRUE-FALSE	FALSE	SET Priority Output

	Boolean	Bit string					Integer						Real number		Moment, Duration, Date, String					
	BOOL	BYTE	WORD	DWORD	LWORD	USINT	UINT	UDINT	ULINT	SINT	INT	DINT	LINT	REAL	LREAL	TIME	DATE	TOD	DT	STRING
SET1	√	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
RESET	√	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
Q1	√	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-

#### ⊙ Functional Description

Rising edge: Set Q1 to TRUE (dominant).

Rising edge: Reset Q1 to FALSE

#### ⊙ Program demo

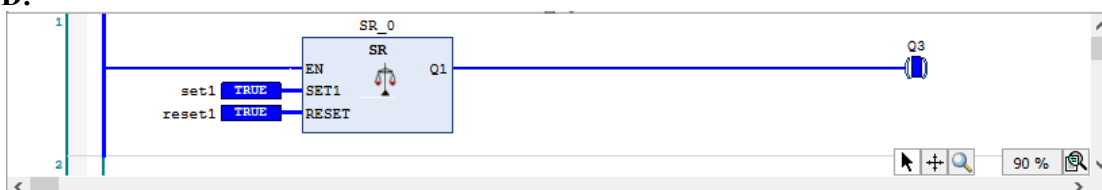
ST:

```

1 //-----SR
2 ● SR_0(SET1 TRUE := set1 TRUE , RESET TRUE :=reset1 TRUE | Q1 TRUE =>Q3 TRUE);

```

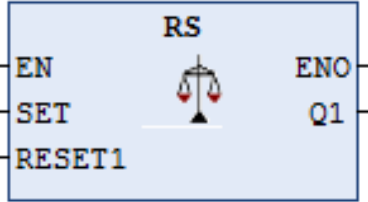
LD:



### 3.5.7 RS

Realizes a bistable reset-dominant latch.

#### ⊙ Command Format

Instruction	Name	FB/FC	LD	ST	File
SR	Reset latch instruction	FB		RS_1( SET1:=, RESET:=, Q1=>);	-

#### ⊙ Related Variables

##### Input variable

Input variable	Name	Data type	Range	initialization	Descriptive
SET1	SET	BOOL	TRUE-FALSE	FALSE	Data 1
RESET	RESET	BOOL	TRUE-FALSE	FALSE	Data 2

##### Output variable

Output variable	Name	Data type	Range	initialization	Descriptive
Q1	Output	BOOL	TRUE-FALSE	FALSE	SET Priority Output

	Boolean	Bit string				Integer							Real number		Moment, Duration, Date, String					
	BOOL	BYTE	WORD	DWORD	LWORD	USINT	UINT	UDINT	ULINT	SINT	INT	DINT	LINT	REAL	LREAL	TIME	DATE	TOD	DT	STRING
SET1	√	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
RESET	√	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
Q1	√	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-

#### ⊙ Functional Description

Rising edge: Set Q1 to TRUE.

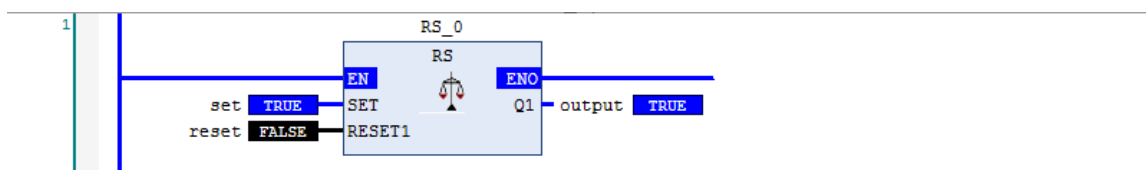
Rising edge: Reset Q1 to FALSE (dominant).

#### ⊙ Program demo

ST:

```
1 | RS_0(SET TRUE :=set TRUE, RESET1 FALSE :=reset FALSE, Q1 TRUE =>output TRUE);RETURN
```

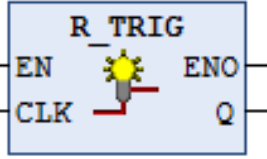
LD:



### 3.5.8 R\_TRIG

Detects a rising edge of a boolean signal.

#### ⊙ Command Format

Instruction	Name	FB/FC	LD	ST	File
R_TRIG	Rising edge instruction	FB		R_TRIG (CLK:=, Q=>);	-

#### ⊙ Related Variables

##### Input variable

Input variable	Name	Data type	Range	initialization	Descriptive
CLK	Input	BOOL	TRUE-FALSE	FALSE	BOOL, detects its rising edge

##### Output variable

Output variable	Name	Data type	Range	initialization	Descriptive
Q1	Output	BOOL	TRUE-FALSE	FALSE	Output

	Boolean	Bit string					Integer						Real number		Moment, Duration, Date, String					
	BOOL	BYTE	WORD	DWORD	LWORD	USINT	UINT	UDINT	ULINT	SINT	INT	DINT	LINT	REAL	LREAL	TIME	DATE	TOD	DT	STRING
CLK	√	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
Q1	√	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-

#### ⊙ Functional Description

Checking Boolean signals, Rising edge detected: Q = TRUE.

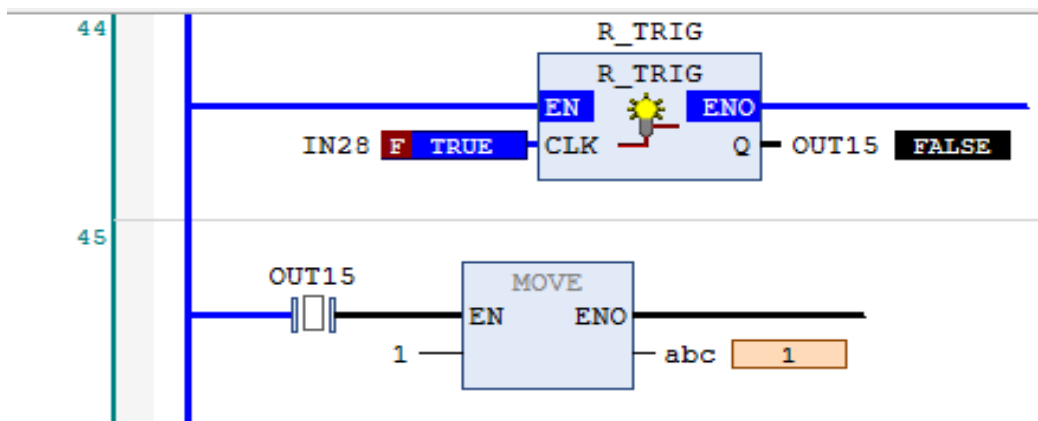
#### ⊙ Program demo

```

ST:
1 //-----R_TRIG
2 R_TRIG1(CLK:=VarBool5 TRUE , Q:=FALSE=>QBool FALSE );
3 IF QBool FALSE THEN
4   abc 1 :=1;
5 END_IF

```

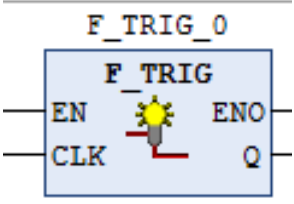
#### LD:



### 3.5.9 F\_TRIG

Detects a falling edge of a boolean signal.

#### ⊙ Command Format

Instruction	Name	FB/FC	LD	ST	File
F_TRIG	Falling edge instruction	FB		F_TRIG (CLK:=, Q=>);	-

#### ⊙ Related Variables

##### Input variable

Input variable	Name	Data type	Range	initialization	Descriptive
CLK	Input	BOOL	TRUE-FALSE	FALSE	BOOL, detects its Falling edge

##### Output variable

Output variable	Name	Data type	Range	initialization	Descriptive
Q1	Output	BOOL	TRUE-FALSE	FALSE	Output

	Boolean	Bit string				Integer						Real number		Moment, Duration, Date, String						
	BOOL	BYTE	WORD	DWORD	LWORD	USINT	UINT	UDINT	ULINT	SINT	INT	DINT	LINT	REAL	LREAL	TIME	DATE	TOD	DT	STRING
CLK	√	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
Q1	√	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-

#### ⊙ Functional Description

Checking Boolean signals, Falling edge detected: Q = TRUE.

#### ⊙ Program demo

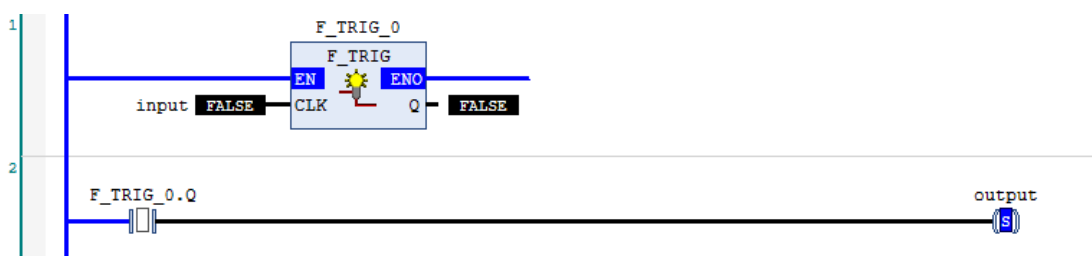
ST:

```

1 F_TRIG_0(CLK:=input FALSE, Q=>);
2 IF F_TRIG_0.Q THEN
3   output TRUE :=TRUE;
4 END_IF RETURN

```

LD:



### 3.6 Data shifting.

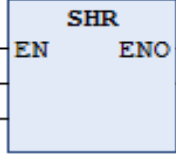
#### 3.6.1 Instruction List

Instruction Type	Name	FB/FC	Functionality
data shift instruction	SHR	FC	And
	SHL	FC	Or
	ROR	FC	Exclusive OR
	ROL	FB	Bistable set-dominat latch

#### 3.6.2 SHR

The IEC operator is used for bitwise shift of an operand to the right.

##### ⊙ Command Format

Instruction	Name	FB/FC	LD	ST	File
SHR	Bit Shift Right instruction	FC		SHR(IN,N);	-

##### ⊙ Related Variables

###### Input variable

Input variable	Name	Data type	Range	initialization	Descriptive
IN	Operand which is shifted to the right	-	-	0	Data sources
N	Number of bits to shift in to the right	-	-	0	Shift Right

###### Output variable

Output variable	Name	Data type	Range	initialization	Descriptive
OUT	Resulting data after offset	-	-	0	Converted data

	Boolean	Bit string				Integer							Real number		Moment, Duration, Date, String					
	BOOL	BYTE	WORD	DWORD	LWORD	USINT	UINT	UDINT	ULINT	SINT	INT	DINT	LINT	REAL	LREAL	TIME	DATE	TOD	DT	STRING
IN	-	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	-	-	-	-	-	-	-
N	-	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	-	-	-	-	-	-	-
OUT	-	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	-	-	-	-	-	-	-

##### ⊙ Functional Description

The data of IN is shifted to the right by n bits and output OUT. The binary is shifted to the right by one bit, which is equal to dividing the original number by 2.

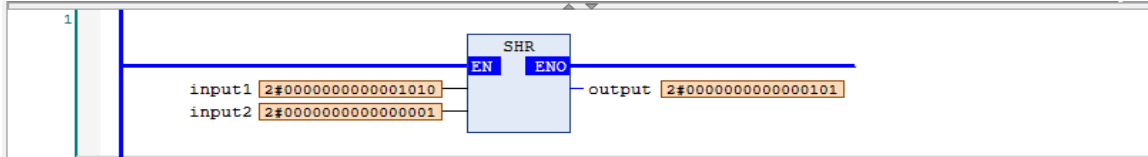
If n exceeds the data type width, then it depends on the target system how the BYTE, WORD, DWORD, and LWORD operands are padded. The target systems cause padding with zeros or n MOD <register size>.

⊙ **Program demo**

**ST:**

```
1 | ● output 2#00000000000000101 :=SHR(input1 2#0000000000001010 ,input2 2#0000000000000001) ;RETURN
```

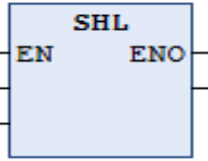
**LD:**



### 3.6.3 SHL

The IEC operator is used for bitwise shift of an operand to the left.

#### ⊙ Command Format

Instruction	Name	FB/FC	LD	ST	File
SHL	Bit Shift Lift instruction	FC		SHLN,N);	-

#### ⊙ Related Variables

##### Input variable

Input variable	Name	Data type	Range	initialization	Descriptive
IN	Operand which is shifted to the left.	-	-	0	Data sources
N	Number of bits to shift in to the left.	-	-	0	Shift Left

##### Output variable

Output variable	Name	Data type	Range	initialization	Descriptive
OUT	Resulting data after offset	-	-	0	Converted data

	Boolean	Bit string					Integer							Real number		Moment, Duration, Date, String				
	BOOL	BYTE	WORD	DWORD	LWORD	USINT	UINT	UDINT	ULINT	SINT	INT	DINT	LINT	REAL	LREAL	TIME	DATE	TOD	DT	STRING
IN	-	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	-	-	-	-	-	-	-
N	-	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	-	-	-	-	-	-	-
OUT	-	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	-	-	-	-	-	-	-

#### ⊙ Functional Description

The data from IN is shifted left by n bits and output to OUT. A one-bit left shift in binary is equal to multiplying the original number by 2.

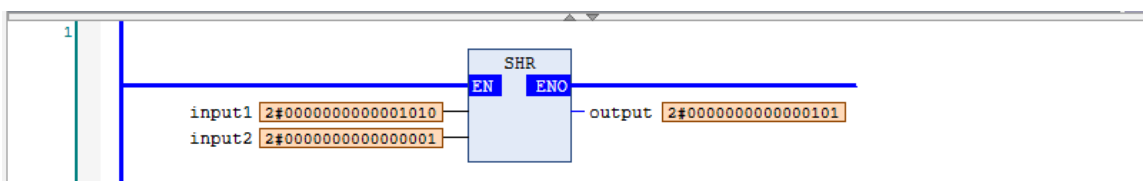
If n exceeds the data type width, then it depends on the target system how the BYTE, WORD, DWORD, and LWORD operands are padded. The target systems cause padding with zeros or n MOD <register size>.

#### ⊙ Program demo

ST:

```
1 | output 2#000000000000000101 :=SHR(input1 2#0000000000001010 ,input2 2#0000000000000001) ;RETURN
```

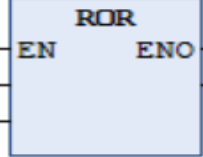
LD:



### 3.6.4 ROR

The IEC operator is used for bitwise rotation of an operand to the right.  
Permitted data types: BYTE, WORD, DWORD, LWORD

#### ⊙ Command Format

Instruction	Name	FB/FC	LD	ST	File
ROR	Cyclic Right Shift instruction	FC		ROR(IN,N);	-

#### ⊙ Related Variables

##### Input variable

Input variable	Name	Data type	Range	initialization	Descriptive
IN	Cyclic Right Shift.	-	-	0	Data sources
N	Number of bits to shift in to the right.	-	-	0	Shift Right

##### Output variable

Output variable	Name	Data type	Range	initialization	Descriptive
OUT	Resulting data after offset	-	-	0	Converted data

	Boolean	Bit string				Integer							Real number		Moment, Duration, Date, String						
	BOOL	BYTE	WORD	DWORD	LWORD	USINT	UINT	UDINT	ULINT	SINT	INT	DINT	LINT	REAL	LREAL	TIME	DATE	TOD	DT	STRING	
IN	-	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	-	-	-	-	-	-	-	-
N	-	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	-	-	-	-	-	-	-	-
OUT	-	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	-	-	-	-	-	-	-	-

#### ⊙ Functional Description

Instruction moves in n-times one bit to the right and adds the bit to the rightmost position from the left.

The results for `erg_byte` and `erg_word` are different depending on the data type of the input variables, although the values of the `in_byte` and `in_word` input variables are the same.

#### ⊙ Program demo

ST:

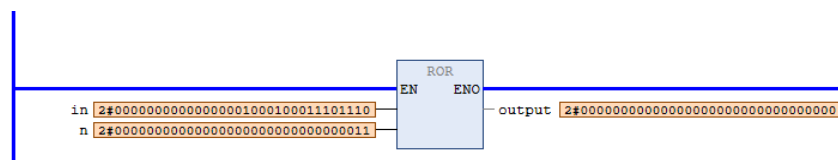
```

1  PROGRAM PLC_FRG
2  VAR
3    in_byte : BYTE := 16#45;
4    in_word : WORD := 16#45;
5    erg_byte : BYTE;
6    erg_word : WORD;
7    n : BYTE := 2;
8  END_VAR

1  erg_byte := ROR(in_byte,n); (* Result: 16#51 *)
2  erg_word := ROR(in_word,n); (* Result: 16#4011 *)

```

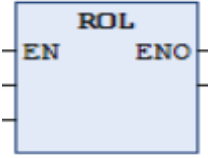
LD:



### 3.6.5 ROL

The IEC operator is used for bitwise rotation of an operand to the left.  
Permitted data types: BYTE, WORD, DWORD, LWORD

#### ⊙ Command Format

Instruction	Name	FB/FC	LD	ST	File
ROL	Cyclic Lift Shift instruction	FC		ROL(IN,N);	-

#### ⊙ Related Variables

##### Input variable

Input variable	Name	Data type	Range	initialization	Descriptive
IN	Cyclic Lift Shift instruction.	-	-	0	Data sources
N	Number of bits to shift in to the lift	-	-	0	Shift Left

##### Output variable

Output variable	Name	Data type	Range	initialization	Descriptive
OUT	Resulting data after offset	-	-	0	Converted data

	Boolean	Bit string					Integer							Real number		Moment, Duration, Date, String				
	BOOL	BYTE	WORD	DWORD	LWORD	USINT	UINT	UDINT	ULINT	SINT	INT	DINT	LINT	REAL	LREAL	TIME	DATE	TOD	DT	STRING
IN	-	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	-	-	-	-	-	-	-
N	-	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	-	-	-	-	-	-	-
OUT	-	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	-	-	-	-	-	-	-

#### ⊙ Functional Description

Instruction moves in n-times one bit to the left and adds the bit to the leftmost position from the right.  
The results for erg\_byte and erg\_word are different depending on the data type of the input variables, although the values of the in\_byte and in\_word input variables are the same.

#### ⊙ Program demo

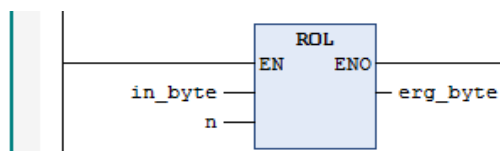
ST:

```

1 PROGRAM POU
2 VAR
3   in_byte : BYTE := 16#45;
4   in_word : WORD := 16#45;
5   erg_byte : BYTE;
6   erg_word : WORD;
7   n : BYTE := 2;
8 END_VAR
9
10
11 erg_byte := ROL(in_byte,n); (* Result: 16#15 *)
12 erg_word := ROL(in_word,n); (* Result: 16#0114 *)

```

LD:



### 3.7 Data type conversion instruction.

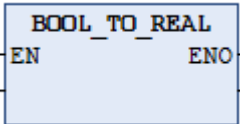
#### 3.7.1 Instruction List

Instruction Type	Name	FB/FC	Functionality
Data type conversion instruction	BOOL TO <TYPE>	FC	BOOL type conversion instruction
	BYTE TO <TYPE>	FC	Byte Type Conversion Instruction
	WORD TO <TYPE>	FC	Word Type Conversion Instruction
	DWORD TO <TYPE>	FC	Double Word Type Conversion Instruction
	INT TO <TYPE>	FC	Integer Type Conversion
	SINT TO <TYPE>	FC	Short Integer Type Conversion Instruction
	DINT TO <TYPE>	FC	Long Integer Type Conversion
	UDINT TO <TYPE>	FC	Unsigned Long Integer Conversion
	REAL TO <TYPE>	FC	Real Type Conversion
	STRING TO <TYPE>	FC	Character Type Conversion
	TIME TO <TYPE>	FC	Clock Type Conversion
	TOD TO <TYPE>	FC	Time Type Conversion
	DATE TO <TYPE>	FC	Date Type Conversion
	DT TO <TYPE>	FC	Date-Time Type Conversion

#### 3.7.2 BOOL\_TO\_<TYPE>

Converts Boolean data types to other data types.

##### ⊙ Command Format

Instruction	Name	FB/FC	LD	ST	File
BOOL_TO_<TYPE>	Bool type conversion instruction	FC		Output:= BOOL_TO_REAL (var1);	-

##### ⊙ Related Variables

###### Input variable

Input variable	Name	Data type	Range	initialization	Descriptive
IN	Data sources	BOOL	TRUE-FALSE	FALSE	Data sources

###### Output variable

Output variable	Name	Data type	Range	initialization	Descriptive
OUT	Data sources	-	-	0	Converted data

	Boolean	Bit string				Integer							Real number		Moment, Duration, Date, String					
	BOOL	BYTE	WORD	DWORD	LWORD	USINT	UINT	UDINT	ULINT	SINT	INT	DINT	LINT	REAL	LREAL	TIME	DATE	TOD	DT	STRING
IN	✓	✓	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
OUT	-	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓

##### ⊙ Functional Description

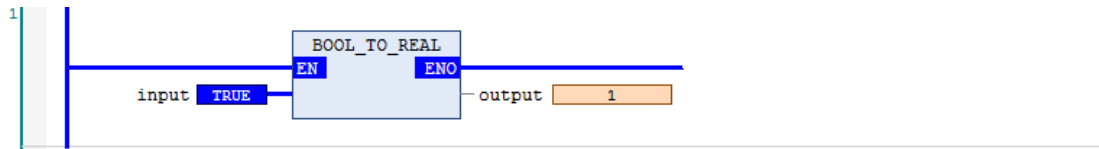
Supported data types: BYTE, WORD, DWORD, SINT, USINT, INT, UINT, DINT, UDINT, REAL, TIME, DATE, TOD, DT and STRING.

When the output is a numeric type, if the input is TRUE, the output is 1; if the input is FALSE, the output is 0.

When the output is a string type, if the input is TRUE, the output is the string 'TRUE'; if the input is FALSE, the output is the string 'FALSE'.

**⊙ Program demo****ST:**

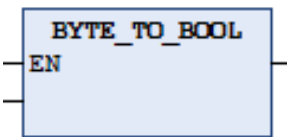
```
1 ● output 1 :=BOOL_TO_real(input TRUE);RETURN
```

**LD:**

### 3.7.3 BYTE\_TO\_<TYPE>

Converts byte data types to other data types.

#### ⊙ Command Format

Instruction	Name	FB/FC	LD	ST	File
BYTE_TO_<TYPE>	Byte type conversion instruction	FC		Output:= BYTE_TO_TYPE (var1);	-

#### ⊙ Related Variables

##### Input variable

Input variable	Name	Data type	Range	initialization	Descriptive
IN	Data sources	BYTE	0-255	0	Data sources

##### Output variable

Output variable	Name	Data type	Range	initialization	Descriptive
OUT	Data sources	-	-	0	Converted data

	Boolean	Bit string				Integer							Real number		Moment, Duration, Date, String					
	BOOL	BYTE	WORD	DWORD	LWORD	USINT	UINT	UDINT	ULINT	SINT	INT	DINT	LINT	REAL	LREAL	TIME	DATE	TOD	DT	STRING
IN	-	✓	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
OUT	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓

#### ⊙ Functional Description

Supported data types: BOOL, WORD, DWORD, SINT, USINT, INT, UINT, DINT, UDINT, REAL, TIME, DATE, TOD, DT and STRING.

When the output is BOOL: the output will be TRUE if the input is not equal to 0, or FALSE if the input is equal to 0. When the output is TIME or TOD: the input will be replaced in milliseconds. When the output is DATE or DT: the input will be replaced in seconds.

#### ⊙ Program demo

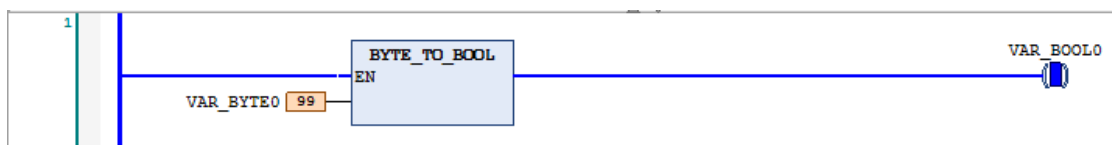
ST:

```

1 //-----BYTE_TO_BOOL
2 VAR_BOOL0 TRUE := BYTE_TO_BOOL(VAR_BYTE0 10);

```

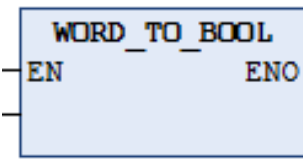
LD:



### 3.7.4 <Integer data>\_TO\_<TYPE>

Converts integer data types to other data types. Since there are many integer data types and the conversion process is similar, we use WORD\_TO\_BOOL as an example.

#### ⊙ Command Format

Instruction	Name	FB/FC	LD	ST	File
WORD_TO_BOOL	Word type conversion instruction	FC		Output:= WORD_TO_TYP E(IN);	-

#### ⊙ Related Variables

##### Input variable

Input variable	Name	Data type	Range	initialization	Descriptive
IN	Data sources	WORD	0-65535	0	Data sources

##### Output variable

Output variable	Name	Data type	Range	initialization	Descriptive
OUT	Data results	BOOL	0-255	0	Converted data

	Boolean	Bit string					Integer						Real number		Moment, Duration, Date, String					
	BOOL	BYTE	WORD	DWORD	LWORD	USINT	UINT	UDINT	ULINT	SINT	INT	DINT	LINT	REAL	LREAL	TIME	DATE	TOD	DT	STRING
IN	-	-	✓	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
OUT	✓	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-

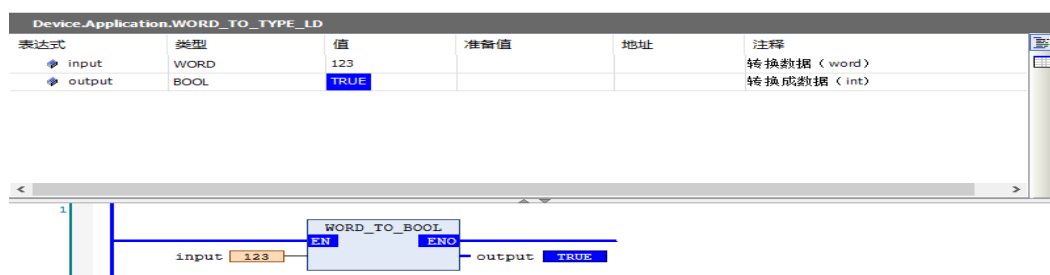
#### ⊙ Program demo

Take WORD\_TO\_BOOL for example

ST:



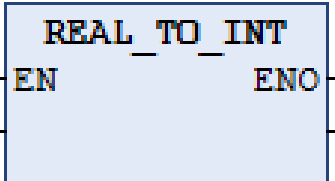
LD:



### 3.7.5 <Real data>\_TO\_<TYPE>

Converts real data types to other data types. Since there are more data types that can be converted and the conversion process is similar, REAL\_TO\_INT is used as an example.

#### ⊙ Command Format

Instruction	Name	FB/FC	LD	ST	File
REAL_TO_INT	Real type conversion instruction	FC		Output:= REAL_TO_INT(I N);	-

#### ⊙ Related Variables

##### Input variable

Input variable	Name	Data type	Range	initialization	Descriptive
IN	Data sources	REAL	-	-	Data sources

##### Output variable

Output variable	Name	Data type	Range	initialization	Descriptive
OUT	Data sources	-	-	-	Converted data

	Boolean	Bit string					Integer							Real number		Moment, Duration, Date, String				
	BOOL	BYTE	WORD	DWORD	LWORD	USINT	UINT	UDINT	ULINT	SINT	INT	DINT	LINT	REAL	LREAL	TIME	DATE	TOD	DT	STRING
IN	-	-	✓	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
OUT	-	-	-	-	-	-	-	-	-	✓	-	-	-	-	-	-	-	-	-	-

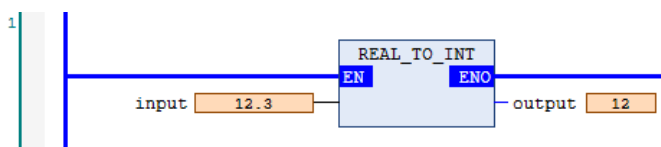
#### ⊙ Program demo

Take REAL\_TO\_INT for example.

ST:

```
1 output 12 :=REAL_TO_INT(input 12.3);RETURN
```

LD:



### 3.8 Data processing instructions.

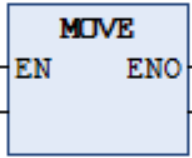
#### 3.8.1 Instruction List

Instruction Type	Name	FB/FC	Functionality
Data processing instruction	MOVE	FC	Assignment
	HEXinASCII TO BYTE	FC	ASCII to Numeric Instructions
	BYTE TO HEXinASCII	FC	Numbers to ASCII
	WORD AS STRING	FC	Numeric to String
	BYTE TO HEXSTRING	FC	BYTE to hexadecimal string
	WORD TO HEXSTRING	FC	WORD type to hexadecimal string
	DWORD TO HEXSTRING	FC	DWORD to hexadecimal string

#### 3.8.2 MOVE

This operator is used to assign the value of one variable to another variable of the same type.

##### ⊙ Command Format

Instruction	Name	FB/FC	LD	ST	File
MOVE	Assignment	FC		A2:=MOVE(A1);	-

##### ⊙ Related Variables

###### Input variable

Input variable	Name	Data type	Range	initialization	Descriptive
IN	Sending data	ALL	-	0	Input data

###### Output variable

Output variable	Name	Data type	Range	initialization	Descriptive
OUT	Receive data	ALL	-	0	Output data

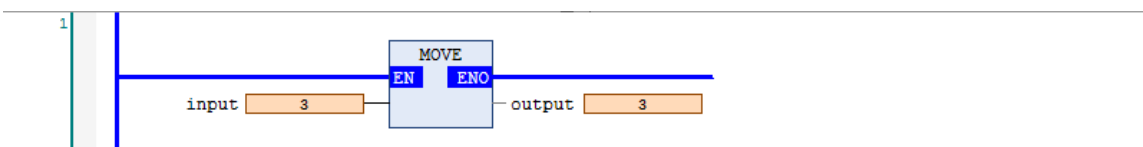
	Boolean	Bit string					Integer							Real number		Moment, Duration, Date, String				
	BOOL	BYTE	WORD	DWORD	LWORD	USINT	UINT	UDINT	ULINT	SINT	INT	DINT	LINT	REAL	LREAL	TIME	DATE	TOD	DT	STRING
IN	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
OUT	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓

##### ⊙ Program demo

ST:

```
1 output 3 :=MOVE (input 3) ;RETURN
```

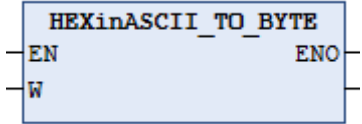
LD:



### 3.8.3 HEXinASCII\_TO\_BYTE

This operator converts hexadecimal ASCII code to BYTE type data.

#### ⊙ Command Format

Instruction	Name	FB/FC	LD	ST	File
HEXinASCII_TO_BYTE	ASCII to Numeric Instructions	FC		output:= HEXinASCII_TO_BYTE(var);	-

#### ⊙ Related Variables

##### Input variable

Input variable	Name	Data type	Range	initialization	Descriptive
W	Sending data	WORD	0-FFFF	0	Input data

##### Output variable

Output variable	Name	Data type	Range	initialization	Descriptive
HEXinASCII_TO_BYTE	Receive data	BYTE	0-255	0	Output data

	Boolean	Bit string				Integer							Real number		Moment, Duration, Date, String						
	BOOL	BYTE	WORD	DWORD	LWORD	USINT	UINT	UDINT	ULINT	SINT	INT	DINT	LINT	REAL	LREAL	TIME	DATE	TOD	DT	STRING	
W	-	-	✓	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
HEXinASCII_TO_BYTE	-	✓	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-

#### ⊙ Program demo

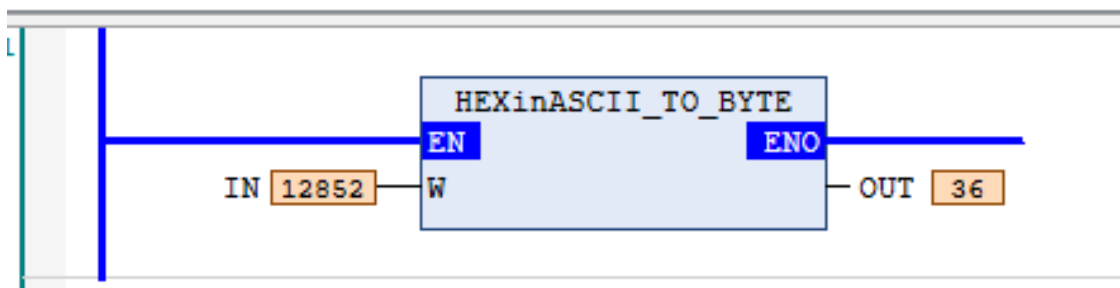
ST:

```

1 | ● out_36 := HEXinASCII_TO_BYTE (IN 12852) ; RETURN

```

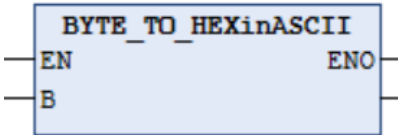
LD:



### 3.8.4 BYTE\_TO\_HEXinASCII

This operator converts hexadecimal ASCII code to BYTE type data.

#### ⊙ Command Format

Instruction	Name	FB/FC	LD	ST	File
BYTE_TO_HEXinASCII	Numbers to ASCII Instructions	FC		output:= BYTE_TO_HEXinASCII (B:=var);	Util

#### ⊙ Related Variables

##### Input variable

Input variable	Name	Data type	Range	initialization	Descriptive
B	Sending data	BYTE	0-255	0	Input data

##### Output variable

Output variable	Name	Data type	Range	initialization	Descriptive
BYTE_TO_HEXinASCII	Receive data	WORD	0-FFFF	0	Output data

	Boolean	Bit string			Integer							Real number		Moment, Duration, Date, String							
	BOOL	BYTE	WORD	DWORD	LWORD	USINT	UINT	UDINT	ULINT	SINT	INT	DINT	LINT	REAL	LREAL	TIME	DATE	TOD	DT	STRING	
W	-	√	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
BYTE_TO_HEXinASCII	-	-	√	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-

#### ⊙ Program demo

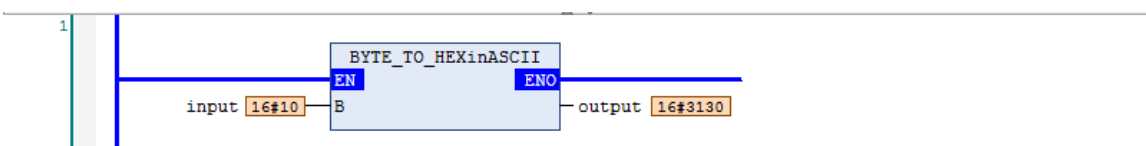
ST:

```

1 | output 16#3130 :=BYTE_TO_HEXinASCII(input 16#10);RETURN

```

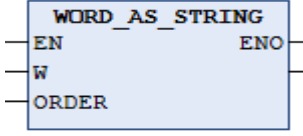
LD:



### 3.8.5 WORD\_AS\_STRING

This operator converts BYTE type data to hexadecimal ASCII code.

#### ⊙ Command Format

Instruction	Name	FB/FC	LD	ST	File
WORD_AS_STRING	Numeric to String Instructions	FC		STR:=WORD_AS_STRING(val_1,ORDER);	Util

#### ⊙ Related Variables

##### Input variable

Input variable	Name	Data type	Range	initialization	Descriptive
W	Sending data	WORD	0-FFFF	0	Input data
DRDER	high-low byte conversion	BOOL	FALSE-TRUE	FALSE	Converting the high and low bytes of a variable

##### Output variable

Output variable	Name	Data type	Range	initialization	Descriptive
WORD_AS_STRING	Receive data	STRING	-	'	Output data

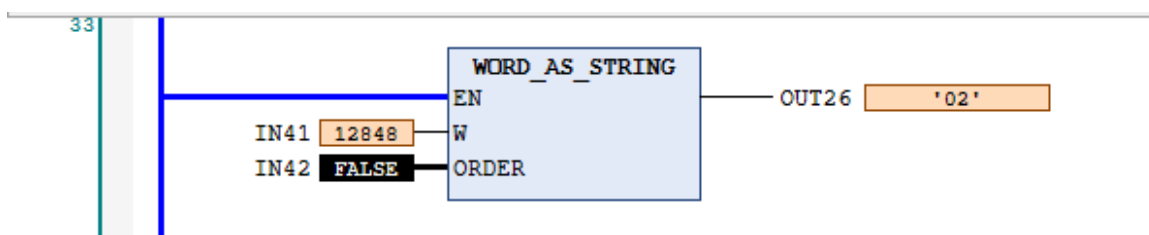
	Boole an	Bit string					Integer							Real number		Moment, Duration, Date, String				
	BOOL	BYTE	WORD	DWORD	LWORD	USINT	UINT	UDINT	ULINT	SINT	INT	DINT	LINT	REAL	LREAL	TIME	DATE	TOD	DT	STRING
W	-	-	✓	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
ORDER	✓	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
WORD_AS_STRING	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	✓

#### ⊙ Program demo

ST:

```
17 //WORD_AS_STRING
18 out57 '1' :=WORD_AS_STRING(IN57 49, IN58 FALSE);
```

LD:



#### Note:

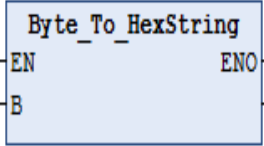
The command does not use the ORDER pin to toggle the high and low bits and does not modify the value. When using the ORDER pin as True, it toggles the high and low bits of the value.

STRING defaults to 80 character quantities, and the variable can be defined as STRING (specific character quantity).

### 3.8.6 BYTE\_TO\_HEXSTRING

This operator converts BYTE type data to hexadecimal ASCII code.

#### ⊙ Command Format

Instruction	Name	FB/FC	LD	ST	File
BYTE_TO_HEXSTRING	BYTE to hexadecimal string Instructions	FC		STR:=BYTE_TO_HEXSTRING(val);	SM3_Shared

#### ⊙ Related Variables

##### Input variable

Input variable	Name	Data type	Range	initialization	Descriptive
B	Sending data	BYTE	0-255	0	Input data

##### Output variable

Output variable	Name	Data type	Range	initialization	Descriptive
BYTE_TO_HEXSTRING	Receive data	STRING	-	'	Output data

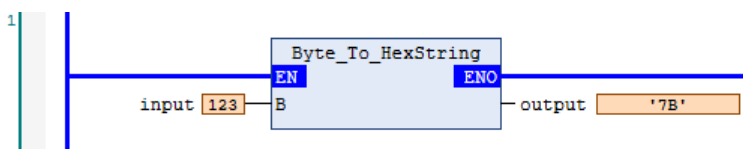
	Boole an	Bit string				Integer							Real number		Moment, Duration, Date, String					
	BOOL	BYTE	WORD	DWORD	LWORD	USINT	UINT	UDINT	ULINT	SINT	INT	DINT	LINT	REAL	LREAL	TIME	DATE	TOD	DT	STRING
B	-	✓	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
BYTE_TO_HEXSTRING	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	✓

#### ⊙ Program demo

ST:

```
1 output[7B] :=Byte_To_HexString(input[123]);RETURN
```

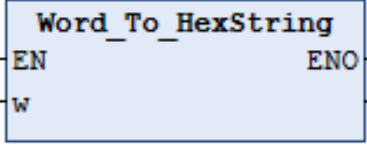
LD:



### 3.8.7 WORD\_TO\_HEXSTRING

This operator converts WORD type data to hexadecimal ASCII code.

#### ⊙ Command Format

Instruction	Name	FB/FC	LD	ST	File
WORD_TO_HEXSTRING	WORD to hexadecimal string Instruction	FC		STR:=BYTE_TO_HEXSTRING(val);	SM3_Shared

#### ⊙ Related Variables

##### Input variable

Input variable	Name	Data type	Range	initialization	Descriptive
W	Sending data	WORD	0-FFFF	0	Input data

##### Output variable

Output variable	Name	Data type	Range	initialization	Descriptive
WORD_TO_HEXSTRING	Receive data	STRING	-	''	Output data

	Boole an	Bit string					Integer							Real number		Moment, Duration, Date, String				
	BOOL	BYTE	WORD	DWORD	LWORD	USINT	UINT	UDINT	ULINT	SINT	INT	DINT	LINT	REAL	LREAL	TIME	DATE	TOD	DT	STRING
B	-	-	✓	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
WORD_TO_HEXSTRING	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	✓

#### ⊙ Program demo

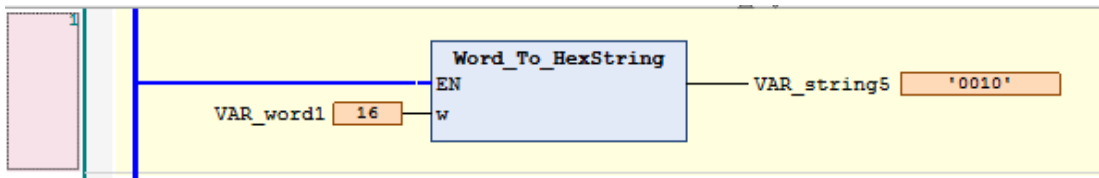
ST:

```

1 //Word_To_HexString
2 VAR_string5 '0010' :=Word_To_HexString(VAR_word1 16 );

```

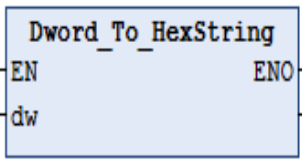
LD:



### 3.8.8 DWORD\_TO\_HEXSTRING

This operator converts DWORD type data to hexadecimal ASCII code.

#### ⊙ Command Format

Instruction	Name	FB/FC	LD	ST	File
DWORD_TO_HEXSTRING	DWORD to hexadecimal string Instructions	FC		STR:=BYTE_TO_HEXSTRING(val);	SM3_Shared

#### ⊙ Related Variables

##### Input variable

Input variable	Name	Data type	Range	initialization	Descriptive
W	Sending data	WORD	0-FFFF	0	Input data

##### Output variable

Output variable	Name	Data type	Range	initialization	Descriptive
WORD_TO_HEXSTRING	Receive data	STRING	-	'	Output data

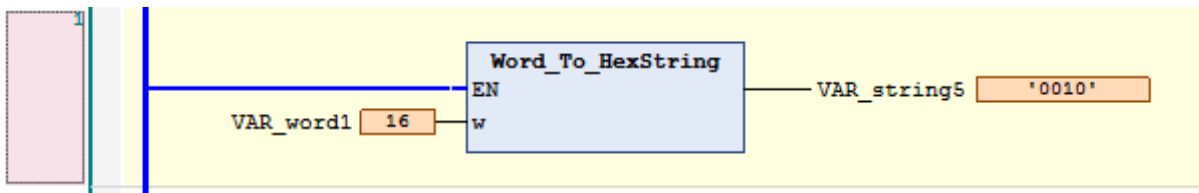
	Boolean	Bit string					Integer						Real number		Moment, Duration, Date, String					
	BOOL	BYTE	WORD	DWORD	LWORD	USINT	UINT	UDINT	ULINT	SINT	INT	DINT	LINT	REAL	LREAL	TIME	DATE	TOD	DT	STRING
B	-	-	-	✓	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
WORD_TO_HEXSTRING	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	✓

#### ⊙ Program demo

ST:

```
STR '00003039' := DWORD_TO_HEXSTRING (VAL 12345);
```

LD:



### 3.9 Mathematical operation instructions.

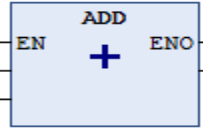
#### 3.9.1 Instruction List

Instruction Type	Name	FB/FC	Functionality
Mathematical operation instruction	ADD	FC	Addition
	SUB	FC	Subtractive
	MUL	FC	Multiplication
	DIV	FC	Division
	MOD	FC	Remainder
	ABS	FC	Absolute value
	SQRT	FC	Square root
	LN	FC	Natural logarithm
	LOG	FC	Common logarithm
	EXP	FC	Index
	EXPT	FC	Power index
	SIN	FC	Sine
	COS	FC	Cosine
	TAN	FC	Tangent
	ASIN	FC	Arcsine
	ACOS	FC	Arccosine
ATAN	FC	Arctangent	
SIZEOF	FC	Get data size	

#### 3.9.2 ADD

Adds the input variables and outputs the result of the addition.

##### ⊙ Command Format

Instruction	Name	FB/FC	LD	ST	File
ADD	Addition Instructions	FC		OUT:=IN1 + IN2;	-

##### ⊙ Related Variables

###### Input variable

Input variable	Name	Data type	Range	initialization	Descriptive
IN1	Additive value	-	ALL	-	Input data
IN2	Additive value	-	ALL	-	Input data

###### Output variable

Output variable	Name	Data type	Range	initialization	Descriptive
OUT	Output value	-	ALL	-	Additive value

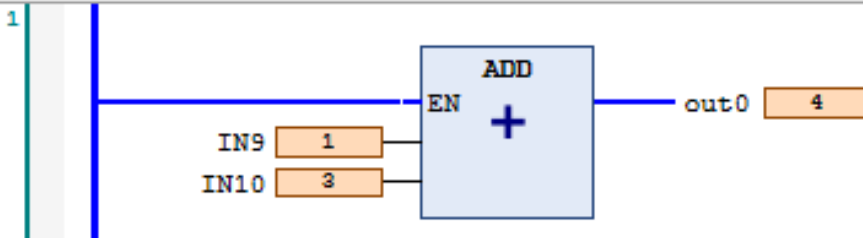
	Boolean	Bit string				Integer							Real number		Moment, Duration, Date, String						
	BOOL	BYTE	WORD	DWORD	LWORD	USINT	UINT	UDINT	ULINT	SINT	INT	DINT	LINT	REAL	LREAL	TIME	DATE	TOD	DT	STRING	
IN1	-	√	√	√	√	√	√	√	√	√	√	√	√	√	√	-	-	√	√	-	
IN2	-	√	√	√	√	√	√	√	√	√	√	√	√	√	√	√	-	-	√	√	-
OUT	-	√	√	√	√	√	√	√	√	√	√	√	√	√	√	√	-	-	√	√	-

⊙ Program demo

ST:

```
1 //ADD  
2 ● out0 4 :=IN9 1 +IN10 3 ;
```

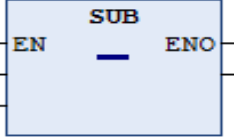
LD:



### 3.9.3 SUB

Subtract the input variables and output the result of the subtraction.

#### ⊙ Command Format

Instruction	Name	FB/FC	LD	ST	File
SUB	Subtractive Instructions	FC		OUT:=IN1 - IN2;	-

#### ⊙ Related Variables

##### Input variable

Input variable	Name	Data type	Range	initialization	Descriptive
IN1	Subtractive value	-	ALL	-	Input data
IN2	Subtractive value	-	ALL	-	Input data

##### Output variable

Output variable	Name	Data type	Range	initialization	Descriptive
OUT	Output value	-	ALL	-	Subtractive value

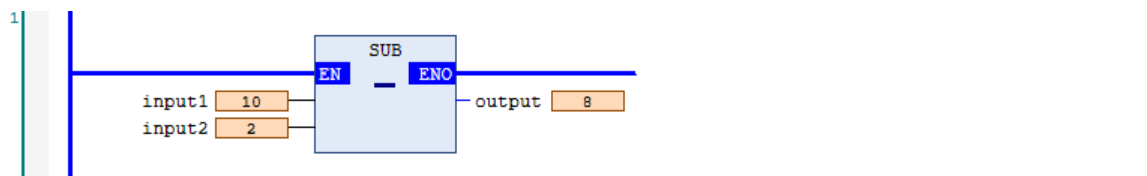
	Boolean	Bit string				Integer							Real number		Moment, Duration, Date, String					
	BOOL	BYTE	WORD	DWORD	LWORD	USINT	UINT	UDINT	ULINT	SINT	INT	DINT	LINT	REAL	LREAL	TIME	DATE	TOD	DT	STRING
IN1	-	√	√	√	√	√	√	√	√	√	√	√	√	√	√	-	-	√	√	-
IN2	-	√	√	√	√	√	√	√	√	√	√	√	√	√	√	√	-	√	√	-
OUT	-	√	√	√	√	√	√	√	√	√	√	√	√	√	√	√	-	√	√	-

#### ⊙ Program demo

##### ST:

```
1 output 8 :=input1 10 -input2 2 ;RETURN
```

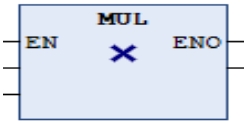
##### LD:



### 3.9.4 MUL

Multiplication of variables or constants.

#### ⊙ Command Format

Instruction	Name	FB/FC	LD	ST	File
MUL	Multiplication Instructions	FC		OUT:=IN1 * IN2;	-

#### ⊙ Related Variables

##### Input variable

Input variable	Name	Data type	Range	initialization	Descriptive
IN1	Multiplication value	-	ALL	-	Input data
IN2	Multiplication value	-	ALL	-	Input data

##### Output variable

Output variable	Name	Data type	Range	initialization	Descriptive
OUT	Output value	-	ALL	-	Multiplication value

	Boolean	Bit string				Integer							Real number		Moment, Duration, Date, String					
	BOOL	BYTE	WORD	DWORD	LWORD	USINT	UINT	UDINT	ULINT	SINT	INT	DINT	LINT	REAL	LREAL	TIME	DATE	TOD	DT	STRING
IN1	-	√	√	√	√	√	√	√	√	√	√	√	√	√	√	√	-	-	-	-
IN2	-	√	√	√	√	√	√	√	√	√	√	√	√	√	√	√	-	-	-	-
OUT	-	√	√	√	√	√	√	√	√	√	√	√	√	√	√	√	-	-	-	-

#### ⊙ Program demo

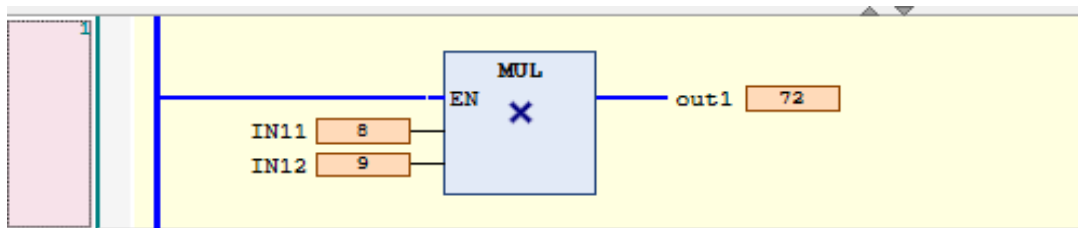
ST:

```

1 //MUL
2 ● out2 18 :=IN11 2 *IN12 9 ;
3 //ADD

```

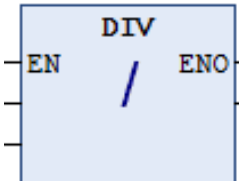
LD:



### 3.9.5 DIV

Dividing two inputs.

☉ **Command Format**

Instruction	Name	FB/FC	LD	ST	File
DIV	Division Instructions	FC		OUT:=IN1 / IN2;	-

☉ **Related Variables**

**Input variable**

Input variable	Name	Data type	Range	initialization	Descriptive
IN1	Dividend	-	ALL	-	Input data
IN2	Divisor	-	ALL	-	Input data

**Output variable**

Output variable	Name	Data type	Range	initialization	Descriptive
OUT	Output value	-	ALL	-	Division value

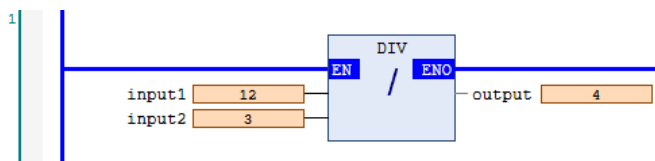
	Boolean	Bit string					Integer							Real number		Moment, Duration, Date, String				
	BOOL	BYTE	WORD	DWORD	LWORD	USINT	UINT	UDINT	ULINT	SINT	INT	DINT	LINT	REAL	LREAL	TIME	DATE	TOD	DT	STRING
IN1	-	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	-	-	-	-
IN2	-	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	-	-	-	-
OUT	-	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	-	-	-	-

☉ **Program demo**

**ST:**

```
1 output:=input1 / input2;RETURN
```

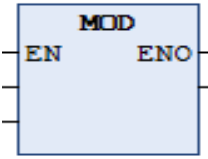
**LD:**



### 3.9.6 MOD

Dividing two inputs.

#### ⊙ Command Format

Instruction	Name	FB/FC	LD	ST	File
MOD	Remainder Instructions	FC		OUT:=IN1 MOD IN2;	-

#### ⊙ Related Variables

##### Input variable

Input variable	Name	Data type	Range	initialization	Descriptive
IN1	Division value	-	ALL	-	Input data
IN2	Division value	-	ALL	-	Input data

##### Output variable

Output variable	Name	Data type	Range	initialization	Descriptive
OUT	Remainder value	-	ALL	-	Division value

	Boolean	Bit string			Integer								Real number		Moment, Duration, Date, String					
	BOOL	BYTE	WORD	DWORD	LWORD	USINT	UINT	UDINT	ULINT	SINT	INT	DINT	LINT	REAL	LREAL	TIME	DATE	TOD	DT	STRING
IN1	-	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	-	-	-	-	-	-	-
IN2	-	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	-	-	-	-	-	-	-
OUT	-	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	-	-	-	-	-	-	-

#### ⊙ Program demo

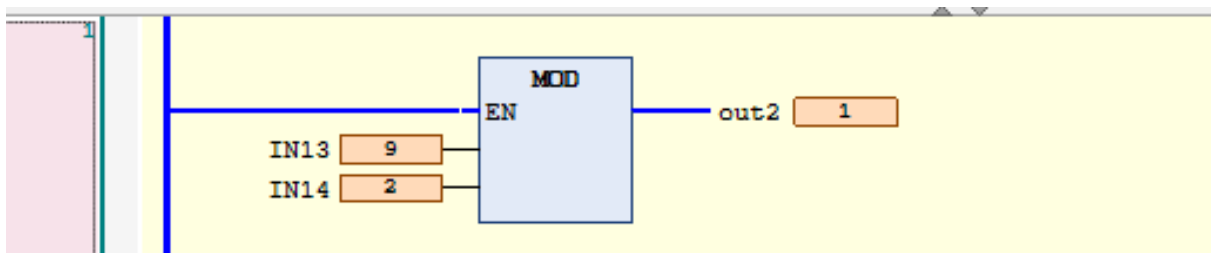
ST:

```

9 //MOD
10 out3 1 :=IN13 9 MOD IN14 2 ;

```

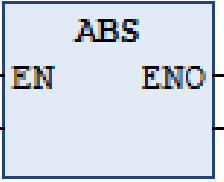
LD:



### 3.9.7 ABS

Dividing two inputs.

⊙ **Command Format**

Instruction	Name	FB/FC	LD	ST	File
ABS	Absolute address Instructions	FC		OUT:=ABS(IN);	-

⊙ **Related Variables**

**Input variable**

Input variable	Name	Data type	Range	initialization	Descriptive
IN	Data value	-	ALL	-	Input data

**Output variable.**

Output variable	Name	Data type	Range	initialization	Descriptive
OUT	Absolute value	-	ALL	-	Output value

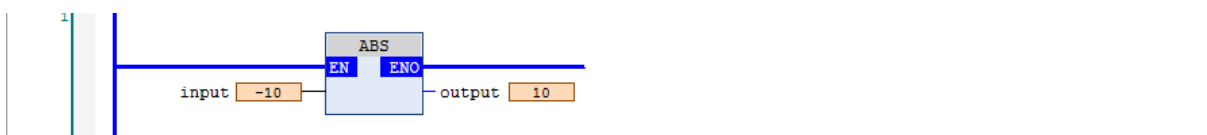
	Boolean	Bit string				Integer							Real number		Moment, Duration, Date, String					
	BOOL	BYTE	WORD	DWORD	LWORD	USINT	UINT	UDINT	ULINT	SINT	INT	DINT	LINT	REAL	LREAL	TIME	DATE	TOD	DT	STRING
IN	-	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	-	-	-	-	-
OUT	-	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	-	-	-	-	-

⊙ **Program demo**

**ST:**

```
1 | output:=ABS(input);RETURN
```

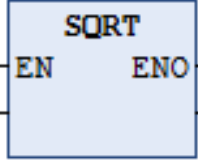
**LD:**



### 3.9.8 SQRT

Perform square root operation on the input data and output the result of the square root operation.

#### ⊙ Command Format

Instruction	Name	FB/FC	LD	ST	File
SQRT	Calculate the square root value	FC		OUT:=SQRT(IN);	-

#### ⊙ Related Variables

##### Input variable

Input variable	Name	Data type	Range	initialization	Descriptive
IN	Data value	-	ALL	-	Input data

##### Output variable.

Output variable	Name	Data type	Range	initialization	Descriptive
OUT	Output value	-	ALL	-	Output value

	Boolean	Bit string					Integer							Real number		Moment, Duration, Date, String				
	BOOL	BYTE	WORD	DWORD	LWORD	USINT	UINT	UDINT	ULINT	SINT	INT	DINT	LINT	REAL	LREAL	TIME	DATE	TOD	DT	STRING
IN	-	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	-	-	-	-	-
OUT	-	-	-	-	-	-	-	-	-	-	-	-	-	✓	✓	-	-	-	-	-

#### ⊙ Program demo

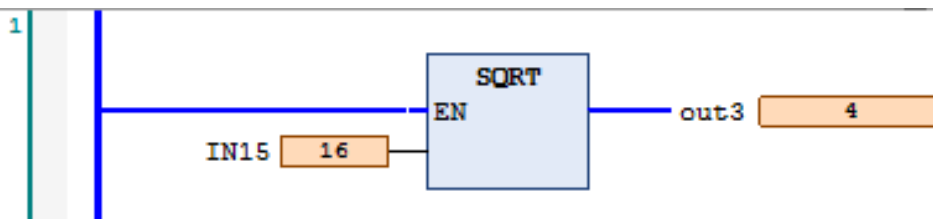
ST:

```

1 //SQRT
2 ● out4 3 :=SQRT (IN15 9 );

```

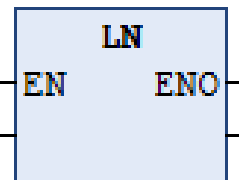
LD:



### 3.9.9 LN

Performs logarithmic operations on input data and outputs logarithmic results.

#### ⊙ Command Format

Instruction	Name	FB/FC	LD	ST	File
LN	logarithmic instruction	FC		OUT:=LN(IN);	-

#### ⊙ Related Variables

##### Input variable

Input variable	Name	Data type	Range	initialization	Descriptive
IN	Data value	-	ALL	-	Input data

##### Output variable.

Output variable	Name	Data type	Range	initialization	Descriptive
OUT	Output value	-	ALL	-	Output value

	Boolean	Bit string					Integer							Real number		Moment, Duration, Date, String				
	BOOL	BYTE	WORD	DWORD	LWORD	USINT	UINT	UDINT	ULINT	SINT	INT	DINT	LINT	REAL	LREAL	TIME	DATE	TOD	DT	STRING
IN	-	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	-	-	-	-	-
OUT	-	-	-	-	-	-	-	-	-	-	-	-	-	✓	✓	-	-	-	-	-

#### ⊙ Program demo

ST:

```
1 output:=LN(10);RETURN
```

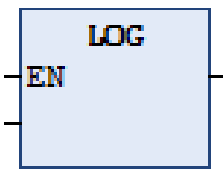
LD:



### 3.9.10 LOG

Perform a logarithmic operation on the input data with base 10 and output the logarithmic result.

#### ⊙ Command Format

Instruction	Name	FB/FC	LD	ST	File
LOG	Common logarithm instruction	FC		OUT:=LOG(IN);	-

#### ⊙ Related Variables

##### Input variable

Input variable	Name	Data type	Range	initialization	Descriptive
IN	Data value	-	ALL	-	Input data

##### Output variable.

Output variable	Name	Data type	Range	initialization	Descriptive
OUT	Output value	-	ALL	-	Output value

	Boolean	Bit string					Integer							Real number		Moment, Duration, Date, String				
	BOOL	BYTE	WORD	DWORD	LWORD	USINT	UINT	UDINT	ULINT	SINT	INT	DINT	LINT	REAL	LREAL	TIME	DATE	TOD	DT	STRING
IN	-	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	-	-	-	-	-
OUT	-	-	-	-	-	-	-	-	-	-	-	-	-	✓	✓	-	-	-	-	-

#### ⊙ Program demo

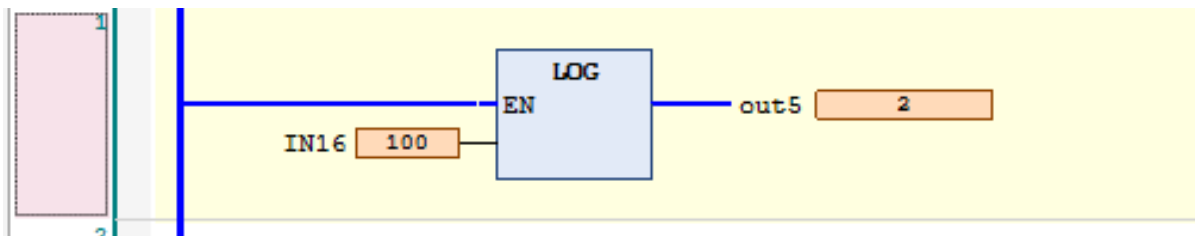
ST:

```

1 | //log
2 | ● out5 3 :=LOG(IN16 1000);

```

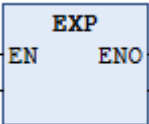
LD:



### 3.9.11 EXP

Perform an exponential operation on the input data and output the result.

#### ⊙ Command Format

Instruction	Name	FB/FC	LD	ST	File
EXP	Index instruction	FC		OUT:=EXP(IN);	-

#### ⊙ Related Variables

##### Input variable

Input variable	Name	Data type	Range	initialization	Descriptive
IN	Data value	-	ALL	-	Input data

##### Output variable.

Output variable	Name	Data type	Range	initialization	Descriptive
OUT	Output value	-	ALL	-	Output value

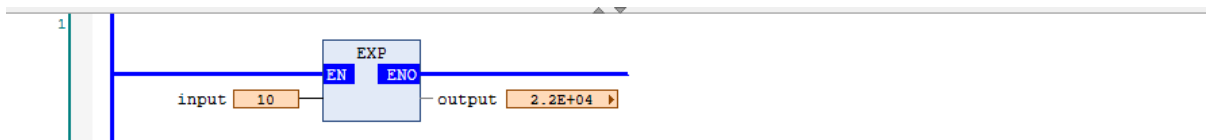
	Boolean	Bit string					Integer						Real number		Moment, Duration, Date, String					
	BOOL	BYTE	WORD	DWORD	LWORD	USINT	UINT	UDINT	ULINT	SINT	INT	DINT	LINT	REAL	LREAL	TIME	DATE	TOD	DT	STRING
IN	-	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	-	-	-	-	-	-
OUT	-	-	-	-	-	-	-	-	-	-	-	-	-	✓	✓	-	-	-	-	-

#### ⊙ Program demo

##### ST:

```
1 | output:=EXP(input);RETURN
```

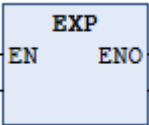
##### LD:



### 3.9.12 EXPT

Output the result of the power operation using input data 1 as the base and input data 2 as the second power

#### ⊙ Command Format

Instruction	Name	FB/FC	LD	ST	File
EXPT	Power index instruction	FC		OUT:=EXPT(IN);	-

#### ⊙ Related Variables

##### Input variable

Input variable	Name	Data type	Range	initialization	Descriptive
IN1	Data 1 value	-	ALL	-	Input data
IN2	Data 2 value	-	ALL	-	Input data

##### Output variable.

Output variable	Name	Data type	Range	initialization	Descriptive
OUT	Output value	-	ALL	-	Output value

	Boolean	Bit string					Integer							Real number		Moment, Duration, Date, String				
	BOOL	BYTE	WORD	DWORD	LWORD	USINT	UINT	UDINT	ULINT	SINT	INT	DINT	LINT	REAL	LREAL	TIME	DATE	TOD	DT	STRING
IN1	-	√	√	√	√	√	√	√	√	√	√	√	√	√	√	-	-	-	-	-
IN2	-	√	√	√	√	√	√	√	√	√	√	√	√	√	√	-	-	-	-	-
OUT	-	-	-	-	-	-	-	-	-	-	-	-	-	√	√	-	-	-	-	-

#### ⊙ Program demo

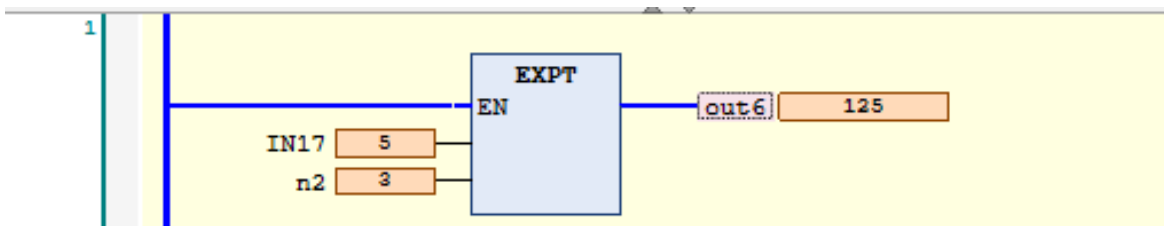
ST:

```

1 //EXPT
2 out6 125 :=EXPT (IN17 5 , n2 3 );

```

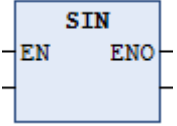
LD:



### 3.9.13 SIN

Output the result of sinusoidal operation by using the input data as a sinusoidal operation.

#### ⊙ Command Format

Instruction	Name	FB/FC	LD	ST	File
SIN	Sine instruction	FC		OUT:=SIN(IN);	-

#### ⊙ Related Variables

##### Input variable

Input variable	Name	Data type	Range	initialization	Descriptive
IN	Data value	-	ALL	-	Input data

##### Output variable.

Output variable	Name	Data type	Range	initialization	Descriptive
OUT	Output value	-	ALL	-	Output value

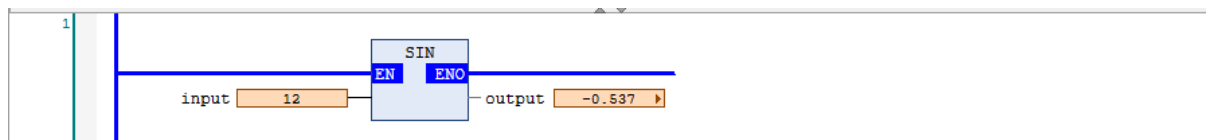
	Boolean	Bit string				Integer							Real number		Moment, Duration, Date, String					
	BOOL	BYTE	WORD	DWORD	LWORD	USINT	UINT	UDINT	ULINT	SINT	INT	DINT	LINT	REAL	LREAL	TIME	DATE	TOD	DT	STRING
IN	-	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	-	-	-	-	-
OUT	-	-	-	-	-	-	-	-	-	-	-	-	-	✓	✓	-	-	-	-	-

#### ⊙ Program demo

##### ST:

```
1 | output:=SIN(input);RETURN
```

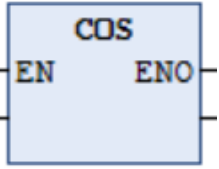
##### LD:



### 3.9.14 COS

Output the result of the cosine operation by using the input data as the cosine operation.

#### ⊙ Command Format

Instruction	Name	FB/FC	LD	ST	File
COS	Cosine instruction	FC		OUT:=COS(IN);	-

#### ⊙ Related Variables

##### Input variable

Input variable	Name	Data type	Range	initialization	Descriptive
IN	Data value	-	ALL	-	Input data

##### Output variable.

Output variable	Name	Data type	Range	initialization	Descriptive
OUT	Output value	-	ALL	-	Output value

	Boolean	Bit string				Integer							Real number		Moment, Duration, Date, String					
	BOOL	BYTE	WORD	DWORD	LWORD	USINT	UINT	UDINT	ULINT	SINT	INT	DINT	LINT	REAL	LREAL	TIME	DATE	TOD	DT	STRING
IN	-	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	-	-	-	-	-
OUT	-	-	-	-	-	-	-	-	-	-	-	-	-	✓	✓	-	-	-	-	-

#### ⊙ Program demo

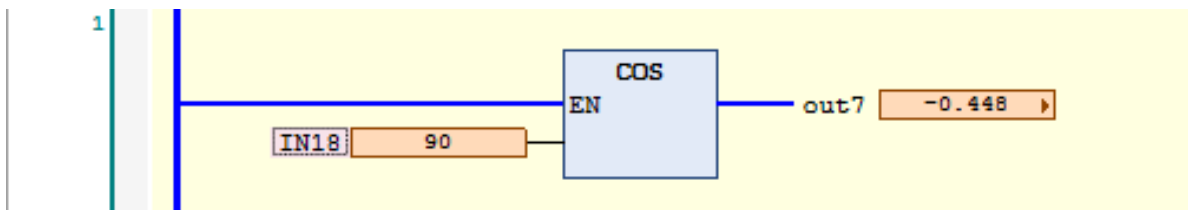
ST:

```

1 | //cos
2 | ● out7 [-0.448] :=COS (IN18 [90] );

```

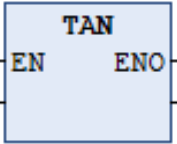
LD:



### 3.9.15 TAN

Output the result of the tangent operation by using the input data as the tangent operation.

#### ⊙ Command Format

Instruction	Name	FB/FC	LD	ST	File
TAN	Tangent instruction	FC		OUT:=TAN(IN);	-

#### ⊙ Related Variables

##### Input variable

Input variable	Name	Data type	Range	initialization	Descriptive
IN	Data value	-	ALL	-	Input data

##### Output variable.

Output variable	Name	Data type	Range	initialization	Descriptive
OUT	Output value	-	ALL	-	Output value

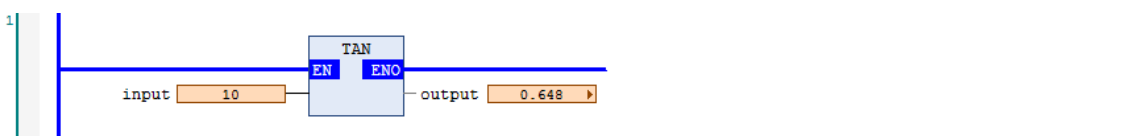
	Boolean	Bit string				Integer							Real number		Moment, Duration, Date, String					
	BOOL	BYTE	WORD	DWORD	LWORD	USINT	UINT	UDINT	ULINT	SINT	INT	DINT	LINT	REAL	LREAL	TIME	DATE	TOD	DT	STRING
IN	-	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	-	-	-	-	-
OUT	-	-	-	-	-	-	-	-	-	-	-	-	-	✓	✓	-	-	-	-	-

#### ⊙ Program demo

ST:

```
1 output:=TAN(input);RETURN
```

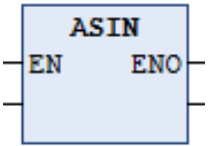
LD:



### 3.9.16 ASIN

Use the input data as an inverse chord operation and output the result of the inverse chord operation.

#### ⊙ Command Format

Instruction	Name	FB/FC	LD	ST	File
ASIN	Arcsine instruction	FC		OUT:=ASIN(IN);	-

#### ⊙ Related Variables

##### Input variable

Input variable	Name	Data type	Range	initialization	Descriptive
IN	Data value	-	ALL	-	Input data

##### Output variable.

Output variable	Name	Data type	Range	initialization	Descriptive
OUT	Output value	-	ALL	-	Output value

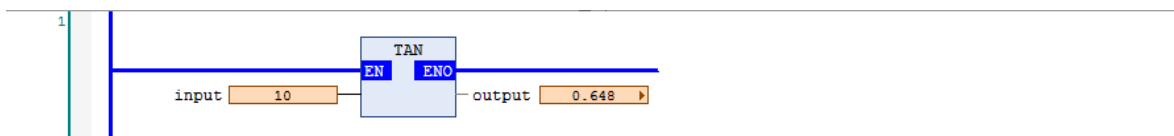
	Boolean	Bit string					Integer						Real number		Moment, Duration, Date, String					
	BOOL	BYTE	WORD	DWORD	LWORD	USINT	UINT	UDINT	ULINT	SINT	INT	DINT	LINT	REAL	LREAL	TIME	DATE	TOD	DT	STRING
IN	-	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	-	-	-	-	-	-
OUT	-	-	-	-	-	-	-	-	-	-	-	-	-	✓	✓	-	-	-	-	-

#### ⊙ Program demo

##### ST:

```
1 output := TAN(input); RETURN
```

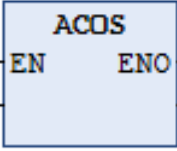
##### LD:



### 3.9.17 ACOS

Input data as inverse cosine operation, output the result of inverse cosine operation.

#### ⊙ Command Format

Instruction	Name	FB/FC	LD	ST	File
ACOS	Arcosine instruction	FC		OUT:=ACOS(IN);	-

#### ⊙ Related Variables

##### Input variable

Input variable	Name	Data type	Range	initialization	Descriptive
IN	Data value	-	ALL	-	Input data

##### Output variable.

Output variable	Name	Data type	Range	initialization	Descriptive
OUT	Output value	-	ALL	-	Output value

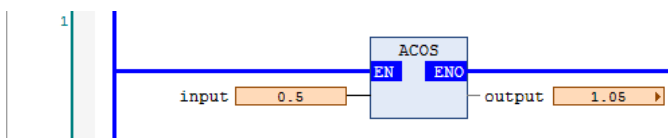
	Boolean	Bit string				Integer							Real number		Moment, Duration, Date, String					
	BOOL	BYTE	WORD	DWORD	LWORD	USINT	UINT	UDINT	ULINT	SINT	INT	DINT	LINT	REAL	LREAL	TIME	DATE	TOD	DT	STRING
IN	-	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	-	-	-	-	-
OUT	-	-	-	-	-	-	-	-	-	-	-	-	-	✓	✓	-	-	-	-	-

#### ⊙ Program demo

ST:

```
1 output:=ACOS(input);RETURN
```

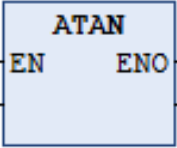
LD:



### 3.9.18 ATAN

Use the input data as an inverse tangent operation and output the result of the inverse tangent operation.

#### ⊙ Command Format

Instruction	Name	FB/FC	LD	ST	File
ATAN	Arctangent instruction	FC		OUT:=ATAN(IN);	-

#### ⊙ Related Variables

##### Input variable

Input variable	Name	Data type	Range	initialization	Descriptive
IN	Data value	-	ALL	-	Input data

##### Output variable.

Output variable	Name	Data type	Range	initialization	Descriptive
OUT	Output value	-	ALL	-	Output value

	Boolean	Bit string				Integer							Real number		Moment, Duration, Date, String					
	BOOL	BYTE	WORD	DWORD	LWORD	USINT	UINT	UDINT	ULINT	SINT	INT	DINT	LINT	REAL	LREAL	TIME	DATE	TOD	DT	STRING
IN	-	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	-	-	-	-	-
OUT	-	-	-	-	-	-	-	-	-	-	-	-	-	✓	✓	-	-	-	-	-

#### ⊙ Program demo

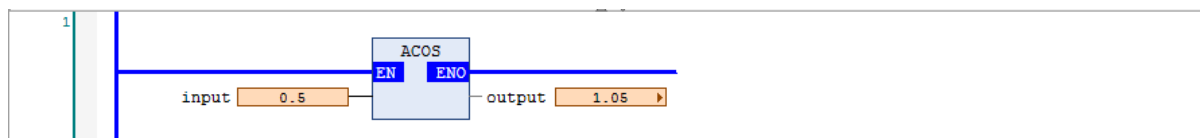
ST:

```

97 //ATAN
98 ● OUT36 0.785 :=ATAN (IN56 1);

```

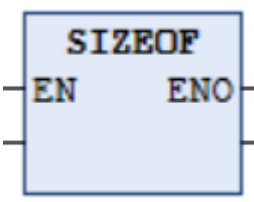
LD:



### 3.9.19 SIZEOF

Used to determine the number of bytes occupied by an input variable. The SIZEOF operator usually returns an unsigned number. The size of the return value is the number of bytes occupied by the type of the input variable.

#### ⊙ Command Format

Instruction	Name	FB/FC	LD	ST	File
SIZEOF	Get data size instruction	FC		OUT:=SIZEOF(IN);	-

#### ⊙ Related Variables

##### Input variable

Input variable	Name	Data type	Range	initialization	Descriptive
IN	Data value	-	ALL	-	Input data

##### Output variable.

Output variable	Name	Data type	Range	initialization	Descriptive
OUT	Output value	-	ALL	-	Output value

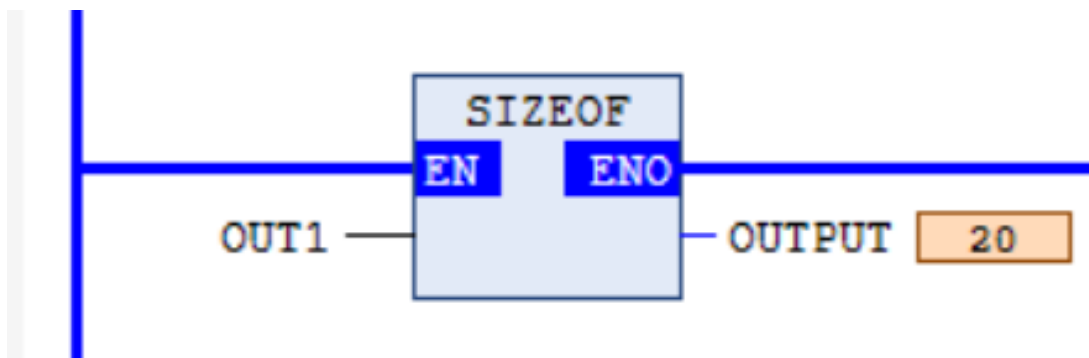
	Boolean	Bit string				Integer							Real number		Moment, Duration, Date, String					
	BOOL	BYTE	WORD	DWORD	LWORD	USINT	UINT	UDINT	ULINT	SINT	INT	DINT	LINT	REAL	LREAL	TIME	DATE	TOD	DT	STRING
IN	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
OUT	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓

#### ⊙ Program demo

ST:

```
● OUTPUT1 [20] :=SIZEOF (IN1) ;
```

LD:



### 3.10 String.

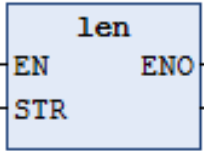
#### 3.10.1 Instruction List

Instruction Type	Name	FB/FC	Functionality
String instruction	LEN	FC	Get string length
	LEFT	FC	String from the left
	RIGHT	FC	Take the string from the right
	MID	FC	Taking strings from the middle
	CONCAT	FC	String concatenation
	INSERT	FC	String Insertion
	DELETE	FC	String Deletion
	FIND	FC	String lookup
REPLACE	FC	String Replacement	

#### 3.10.2 LEN

Get the length of the string command.

##### ⊙ Command Format

Instruction	Name	FB/FC	LD	ST	File
LEN	Get string length instruction	FC		OUT:= LEN (IN);	-

##### ⊙ Related Variables

###### Input variable

Input variable	Name	Data type	Range	initialization	Descriptive
STR	Data value	STRING(255)	-	“	Input data

###### Output variable.

Output variable	Name	Data type	Range	initialization	Descriptive
LEN	Return value	INT	0 - 65535	0	Output value

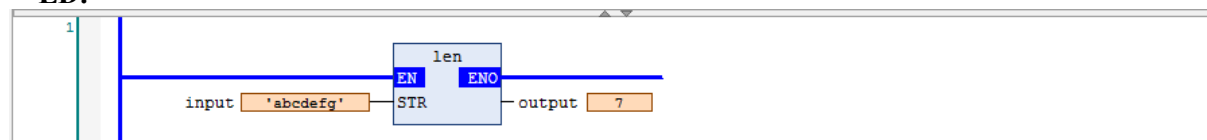
	Boolean	Bit string				Integer							Real number		Moment, Duration, Date, String						
	BOOL	BYTE	WORD	DWORD	LWORD	USINT	UINT	UDINT	ULINT	SINT	INT	DINT	LINT	REAL	LREAL	TIME	DATE	TOD	DT	STRING	
STR	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	✓
LEN	-	-	-	-	-	-	-	-	-	✓	-	-	-	-	-	-	-	-	-	-	-

##### ⊙ Program demo

ST:

```
1 output:=LEN(input);RETURN
```

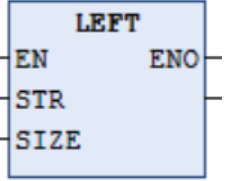
LD:



### 3.10.3 LEFT

Extracts size characters from the left side of the string to the right, and returns the length of the string.

#### ⊙ Command Format

Instruction	Name	FB/FC	LD	ST	File
LEFT	String from the left instruction	FC		STR1:= LEFT (STR,SIZE);	-

#### ⊙ Related Variables

##### Input variable

Input variable	Name	Data type	Range	initialization	Descriptive
STR	Source string	STRING(255)	-	-	Input string
SIZE	Number of characters	INT	0 - 65535	0	

##### Output variable.

Output variable	Name	Data type	Range	initialization	Descriptive
LEFT	Return value	STRING(255)	ALL	,	Number of target strings

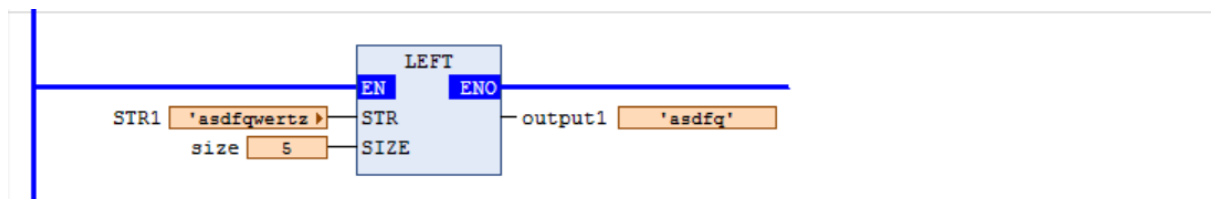
	Boolean	Bit string				Integer							Real number		Moment, Duration, Date, String						
	BOOL	BYTE	WORD	DWORD	LWORD	USINT	UINT	UDINT	ULINT	SINT	INT	DINT	LINT	REAL	LREAL	TIME	DATE	TOD	DT	STRING	
STR	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	✓
SIZE	-	-	-	-	-	-	-	-	-	-	✓	-	-	-	-	-	-	-	-	-	-
LEFT	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	✓

#### ⊙ Program demo

ST:

```
OUTPUT2 'lead' := LEFT (STR1 'leadshine', SIZE 4);
```

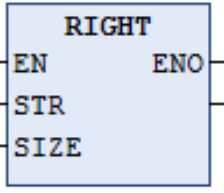
LD:



### 3.10.4 RIGHT

Extracts size characters from the left side of the string to the right, and returns the length of the string.

#### ⊙ Command Format

Instruction	Name	FB/FC	LD	ST	File
RIGHT	String from the right instruction	FC		STR1:= RIGHT (STR,SIZE);	-

#### ⊙ Related Variables

##### Input variable

Input variable	Name	Data type	Range	initialization	Descriptive
STR	Source string	STRING(255)	-	-	Input string
SIZE	Number of characters	INT	0 - 65535	0	

##### Output variable.

Output variable	Name	Data type	Range	initialization	Descriptive
RIGHT	Return value	STRING(255)	ALL	‘	Number of target strings

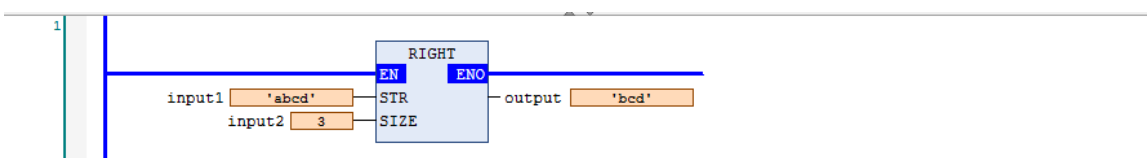
	Boolean	Bit string				Integer							Real number		Moment, Duration, Date, String						
	BOOL	BYTE	WORD	DWORD	LWORD	USINT	UINT	UDINT	ULINT	SINT	INT	DINT	LINT	REAL	LREAL	TIME	DATE	TOD	DT	STRING	
STR	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	✓
SIZE	-	-	-	-	-	-	-	-	-	✓	-	-	-	-	-	-	-	-	-	-	-
RIGHT	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	✓

#### ⊙ Program demo

##### ST:

```
1 output:=RIGHT(input1,input2);RETURN
```

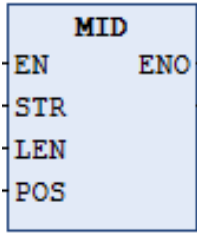
##### LD:



### 3.10.5 MID

Takes a string of the specified length from the specified position in the string and returns the string of the taken length.

☉ **Command Format**

Instruction	Name	FB/FC	LD	ST	File
MID	Strings from the middle instruction	FC		STR1:= MID (STR,LEN,POS);	-

☉ **Related Variables**

**Input variable**

Input variable	Name	Data type	Range	initialization	Descriptive
STR	Source string	STRING(255)	-	-	Input string
LEN	Number of characters	INT	0 - 65535	0	Data length
POS	String position	INT	0 - 65535	0	

**Output variable.**

Output variable	Name	Data type	Range	initialization	Descriptive
MID	Return value	STRING(255)	ALL	,	Number of target strings

	Boolean	Bit string					Integer						Real number		Moment, Duration, Date, String						
	BOOL	BYTE	WORD	DWORD	LWORD	USINT	UINT	UDINT	ULINT	SINT	INT	DINT	LINT	REAL	LREAL	TIME	DATE	TOD	DT	STRING	
STR	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	✓
LEN	-	-	-	-	-	-	-	-	-	✓	-	-	-	-	-	-	-	-	-	-	-
POS	-	-	-	-	-	-	-	-	-	✓	-	-	-	-	-	-	-	-	-	-	-
(MID)	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	✓

☉ **Program demo**

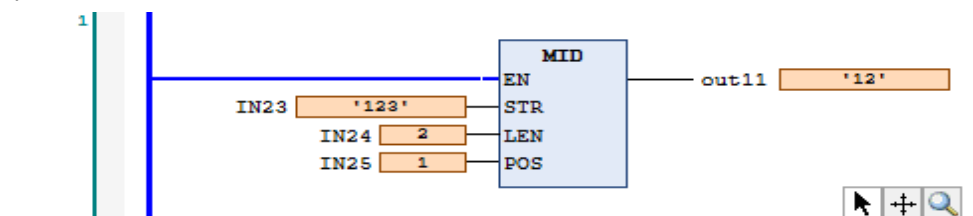
**ST:**

```

1 //MID
2 out11 '12' := MID (IN23 '123', IN24 2, IN25 1);

```

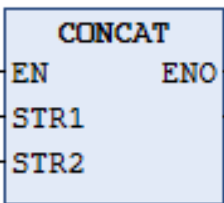
**LD:**



### 3.10.6 CONCAT

Takes a string of the specified length from the specified position in the string and returns the string of the taken length.

#### ⊙ Command Format

Instruction	Name	FB/FC	LD	ST	File
CONCAT	String concatenation instruction	FC		STR:= CONCAT (STR1, STR2);	-

#### ⊙ Related Variables

##### Input variable

Input variable	Name	Data type	Range	initialization	Descriptive
STR1	Source string 1	STRING(255)	-	“	Input string 1
STR2	Source string 1	STRING(255)	-	“	Input string 2

##### Output variable.

Output variable	Name	Data type	Range	initialization	Descriptive
CONCAT	Return value	STRING(255)	-	“	Output the synthesised string

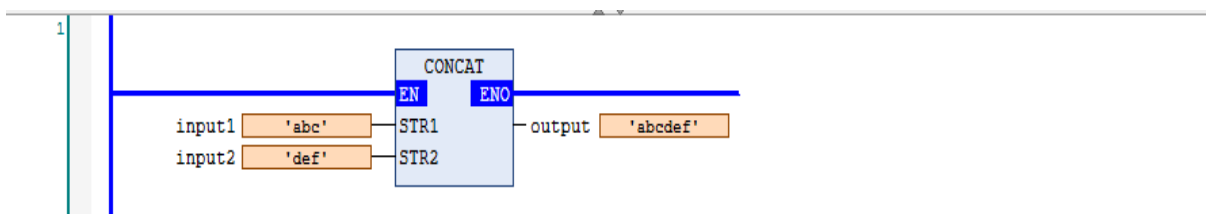
	Boolean	Bit string				Integer							Real number		Moment, Duration, Date, String						
	BOOL	BYTE	WORD	DWORD	LWORD	USINT	UINT	UDINT	ULINT	SINT	INT	DINT	LINT	REAL	LREAL	TIME	DATE	TOD	DT	STRING	
STR1	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	✓
STR2	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	✓
CONCAT	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	✓

#### ⊙ Program demo

##### ST:

```
1 output:=CONCAT(input1,input2);RETURN
```

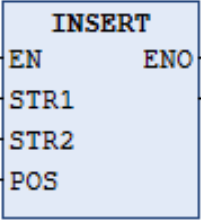
##### LD:



### 3.10.7 INSERT

Takes a string of the specified length from the specified position in the string and returns the string of the taken length.

☉ **Command Format**

Instruction	Name	FB/FC	LD	ST	File
INSERT	String Insertion instruction	FC		STR:= INSERT (STR1, STR2,POS);	-

☉ **Related Variables**

**Input variable**

Input variable	Name	Data type	Range	initialization	Descriptive
STR1	Source string 1	STRING(255)	-	“	Input string 1
STR2	Source string 1	STRING(255)	-	“	Input string 2
POS	Data insertion position	INT	0-65535	0	Data insertion position

**Output variable.**

Output variable	Name	Data type	Range	initialization	Descriptive
INSERT	Return value	STRING(255)	-	“	Output the new string

	Boolean	Bit string				Integer							Real number		Moment, Duration, Date, String						
	BOOL	BYTE	WORD	DWORD	LWORD	USINT	UINT	UDINT	ULINT	SINT	INT	DINT	LINT	REAL	LREAL	TIME	DATE	TOD	DT	STRING	
STR1	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	✓
STR2	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	✓
POS	-	-	-	-	-	-	-	-	-	-	✓	-	-	-	-	-	-	-	-	-	-
INSERT	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	✓

☉ **Program demo**

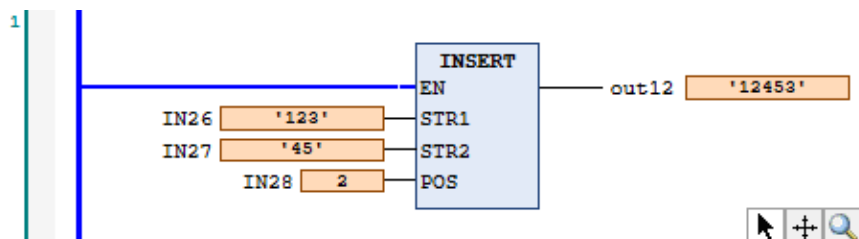
**ST:**

```

1 //INSERT
2 out12 ['12453'] := INSERT (IN26 ['123'], IN27 ['45'], IN28 [2]);

```

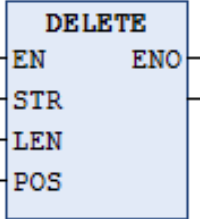
**LD:**



### 3.10.8 DELETE

Takes a string of the specified length from the specified position in the string and returns the string of the taken length.

#### ⊙ Command Format

Instruction	Name	FB/FC	LD	ST	File
DELETE	String Insertion instruction	FC		STR:= DELETE (STR1, STR2,POS);	-

#### ⊙ Related Variables

##### Input variable

Input variable	Name	Data type	Range	initialization	Descriptive
STR1	Source string	STRING(255)	-	''	Input string 1
LEN	Number of characters	INT	0 - 65535	0	Data length
POS	String position	INT	0 - 65535	0	Deleted Character Position

##### Output variable.

Output variable	Name	Data type	Range	initialization	Descriptive
DELETE	Return value	STRING(255)	-	''	Output the synthesised string

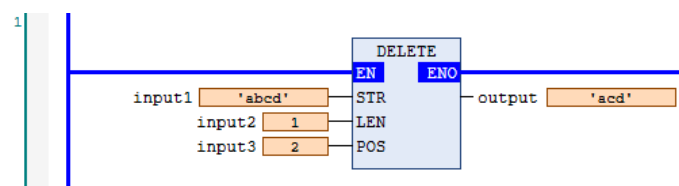
	Boolean	Bit string				Integer							Real number		Moment, Duration, Date, String						
	BOOL	BYTE	WORD	DWORD	LWORD	USINT	UINT	UDINT	ULINT	SINT	INT	DINT	LINT	REAL	LREAL	TIME	DATE	TOD	DT	STRING	
STR	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	✓
LEN	-	-	-	-	-	-	-	-	-	✓	-	-	-	-	-	-	-	-	-	-	-
POS	-	-	-	-	-	-	-	-	-	✓	-	-	-	-	-	-	-	-	-	-	-
DELETE	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	✓

#### ⊙ Program demo

ST:

```
1 output:=DELETE(input1,input2,input3);RETURN
```

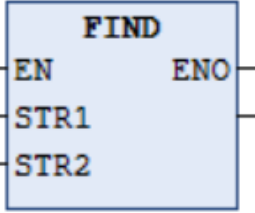
LD:



### 3.10.9 FIND

Detecting the position of the target string in the source string.

#### ⊙ Command Format

Instruction	Name	FB/FC	LD	ST	File
FIND	String lookup instruction	FC		STR:= FIND (STR1, STR2);	-

#### ⊙ Related Variables

##### Input variable

Input variable	Name	Data type	Range	initialization	Descriptive
STR1	Source string	STRING(255)	-	''	Input string 1
STR2	Source string	STRING(255)	-	''	Input string 1

##### Output variable.

Output variable	Name	Data type	Range	initialization	Descriptive
FIND	Return value	INT	0-65536	0	Position of the target string in the source string

	Boolean	Bit string				Integer							Real number		Moment, Duration, Date, String						
	BOOL	BYTE	WORD	DWORD	LWORD	USINT	UINT	UDINT	ULINT	SINT	INT	DINT	LINT	REAL	LREAL	TIME	DATE	TOD	DT	STRING	
STR1	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	✓
STR2	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	✓
Output	-	-	-	-	-	-	-	-	-	-	✓	-	-	-	-	-	-	-	-	-	-

#### ⊙ Program demo

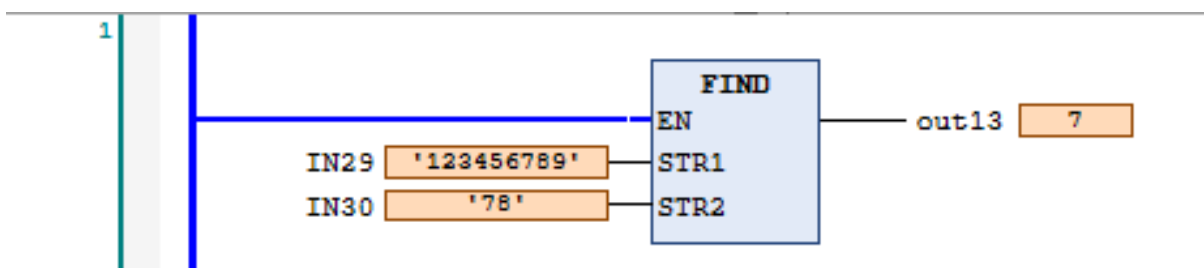
ST:

```

1 //FIND
2 out13 7 := FIND (IN29 '123456789', IN30 '78');

```

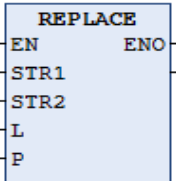
LD:



### 3.10.10 REPLACE

Detecting the position of the target string in the source string.

#### ⊙ Command Format

Instruction	Name	FB/FC	LD	ST	File
REPLACE	String Replacement instruction	FC		STR:= REPLACE (STR1, STR2,L,P);	-

#### ⊙ Related Variables

##### Input variable

Input variable	Name	Data type	Range	initialization	Descriptive
STR1	Source string	STRING(255)	-	‘	Input string 1
STR2	Source string	STRING(255)	-	‘	Input string 1
L	Source string	INT	0-65536	0	Source string
P	Data position	INT	0-65536	0	Data position

##### Output variable.

Output variable	Name	Data type	Range	initialization	Descriptive
REPLACE	Return value	STRING(255)	-	‘	Target string

	Boolean	Bit string				Integer							Real number		Moment, Duration, Date, String						
	BOOL	BYTE	WORD	DWORD	LWORD	USINT	UINT	UDINT	ULINT	SINT	INT	DINT	LINT	REAL	LREAL	TIME	DATE	TOD	DT	STRING	
STR1	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	√
STR2	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	√
L	-	-	-	-	-	-	-	-	-	-	√	-	-	-	-	-	-	-	-	-	-
P	-	-	-	-	-	-	-	-	-	-	√	-	-	-	-	-	-	-	-	-	-
REPLACE	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	√

#### ⊙ Program demo

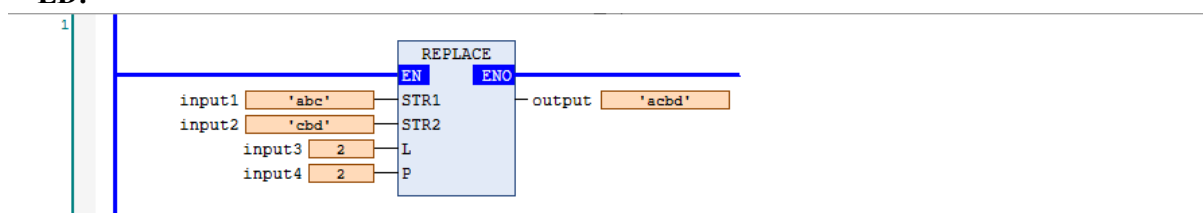
ST:

```

1 output:=REPLACE(input1,input2,input3,input4);RETURN

```

LD:



### 3.11 Address operation.

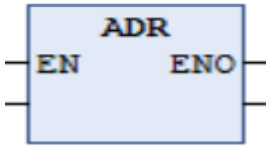
#### 3.11.1 Instruction List

Instruction Type	Name	FB/FC	Functionality
Address operation	ADR	FC	Retrieve an address
	^	FC	Fetch address content
	BITADR	FC	Bit address

#### 3.11.2 ADR

Detecting the position of the target string in the source string.

##### ⊙ Command Format

Instruction	Name	FB/FC	LD	ST	File
ADR	Retrieve an address instruction	FC		OUT:= ADR (IN);	-

##### ⊙ Related Variables

###### Input variable

Input variable	Name	Data type	Range	initialization	Descriptive
IN	Target data	-	-	-	Target data

###### Output variable.

Output variable	Name	Data type	Range	initialization	Descriptive
OUT	Target data address	POINTER_TO_<TYPE>	-	0	Target data address

	Boolean	Bit string				Integer							Real number		Moment, Duration, Date, String					
	BOOL	BYTE	WORD	DWORD	LWORD	USINT	UINT	UDINT	ULINT	SINT	INT	DINT	LINT	REAL	LREAL	TIME	DATE	TOD	DT	STRING
IN	√	√	√	√	√	√	√	√	√	√	√	√	√	√	√	√	√	√	√	√
OUT	POINTER_TO_<TYPE>																			

##### ⊙ Program demo

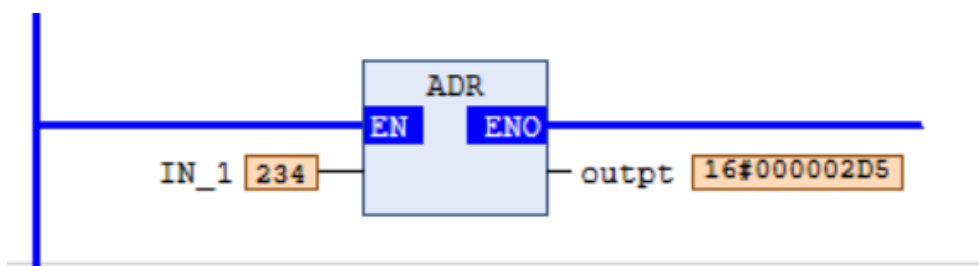
ST:

```

1  outputs[000002D55B609F7] = ADR (IN_1 0);

```

LD:



### 3.11.3 ^

Fetch address content instruction.

#### ⊙ Command Format

Instruction	Name	FB/FC	LD	ST	File
^	Fetch address content instruction	FC	^	^	-

#### ⊙ Related Variables

##### Input variable

Input variable	Name	Data type	Range	initialization	Descriptive
IN	Target data	-	-	-	Target data

##### Output variable.

Output variable	Name	Data type	Range	initialization	Descriptive
OUT	Contents of a pointer to an address	-	-	‘	Contents of a pointer to an address

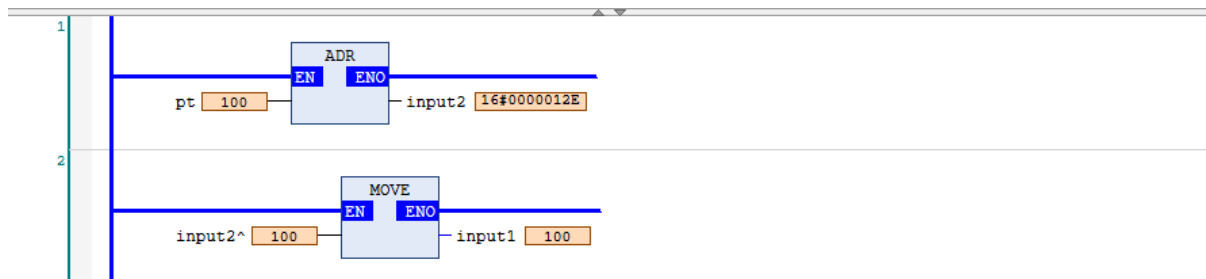
	Boolean	Bit string				Integer							Real number	Moment, Duration, Date, String						
	BOOL	BYTE	WORD	DWORD	LWORD	USINT	UINT	UDINT	ULINT	SINT	INT	DINT	LINT	REAL	LREAL	TIME	DATE	TOD	DT	STRING
OUT	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-

#### ⊙ Program demo

ST:

```
1 input2:=ADR(pt);
2 input1:=input2^;
```

LD:



#### ⊙ Functional Description

The pointer automatically generates the address of the parameter after the programme is run, adding ^ to the pointer represents the contents of the address to which the pointer points.

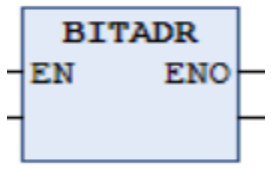
When on-line changes are made, the contents of the address will change and will cause the POINTER variable to point to an unavailable area of memory. To avoid problems, use ADR to update the pointer value once before using the pointer variable.

Do not return POINTER\_TO variables and methods of functions to the calling function or assign them to global variables.

### 3.11.4 BITADR

Take the memory address of the BOOL variable and assign the result to the output variable.

#### ⊙ Command Format

Instruction	Name	FB/FC	LD	ST	File
BITADR	Bit address instruction	FC		OUT:= BITADR (IN);	-

#### ⊙ Related Variables

##### Input variable

Input variable	Name	Data type	Range	initialization	Descriptive
IN	Target data	BOOL	-	-	Target data

##### Output variable.

Output variable	Name	Data type	Range	initialization	Descriptive
OUT	Destination address	DWORD	-	0	Destination address

	Boolean	Bit string				Integer							Real number		Moment, Duration, Date, String					
	BOOL	BYTE	WORD	DWORD	LWORD	USINT	UINT	UDINT	ULINT	SINT	INT	DINT	LINT	REAL	LREAL	TIME	DATE	TOD	DT	STRING
IN	√	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
OUT	-	-	-	√	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-

#### ⊙ Program demo

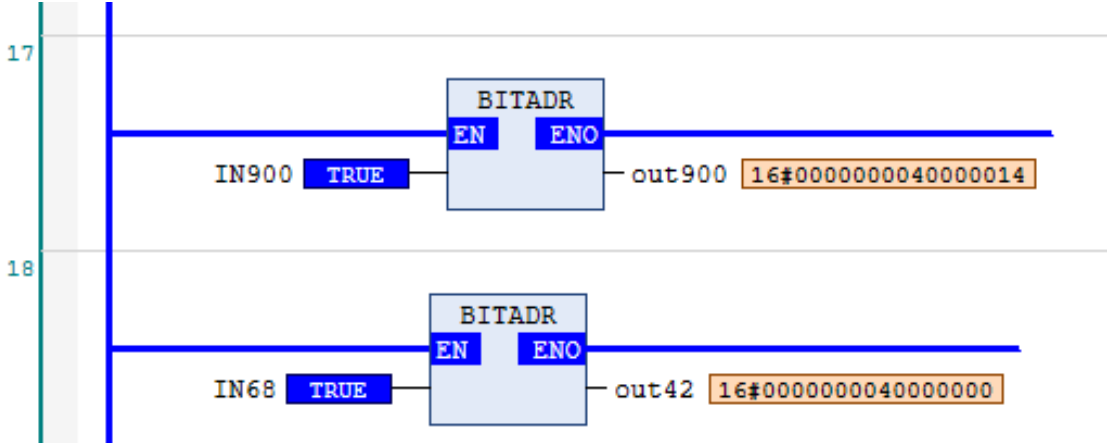
ST:

```

85 //BITADR
86 ● out42 16#0000000040000000 := BITADR(IN68 TRUE);
87 //BITADR
88 ● out900 16#0000000040000014 := BITADR(IN900 TRUE);

```

LD:



## 3.12 File Operations.

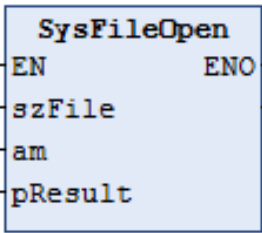
### 3.12.1 Instruction List

Instruction Type	Name	FB/FC	Functionality
File Operations	SysFileOpen	FB	Open file
	SysFileClose	FB	Close file
	SysFileWrite	FB	Write to a file
	SysFileRead	FB	Read file
	SysFileDelete	FB	Delete file
	SysFileCopy	FB	Copy file
	SysFileRename	FB	Rename a file
	SysFileSetPos	FB	Setting the file read/write location
	SysFileGetPos	FB	Read file read/write location
	SysFileGetSize	FB	Get file size
	File.Open	FB	Open files by directory
	File.Close	FB	Close file by directory
	File.Copy	FB	Copy by directory
	File.Create	FB	Create by directory
File.Delete	FB	Delete by directory	

### 3.12.2 SysFileOpen

Open an existing file or create a new one.

#### ⊙ Command Format

Instruction	Name	FB/FC	LD	ST	File
SysFileOpen	Open file instruction	FC		SysFileOpen( szFile:=, am:=, pResult:=);	SysFile

#### ⊙ Related Variables

##### Input variable

Input variable	Name	Data type	Range	initialization	Descriptive
szFile	filename	STRING	-	-	Path to open or create a file
am	Document access modalities	ACCESS_MODE	-	-	Requested document access patterns
pResult	Pointer to error code	POINTER TO RTS_IEC_RESULT	-	-	Pointer to runtime system error

##### Output variable.

Output variable	Name	Data type	Range	initialization	Descriptive
SysFileOpen	Return value	RTS_IEC_HANDLE	-	-	File handles

	Boolean	Bit string					Integer						Real number		Moment, Duration, Date, String					
	BOOL	BYTE	WORD	DWORD	LWORD	USINT	UINT	UDINT	ULINT	SINT	INT	DINT	LINT	REAL	LREAL	TIME	DATE	TOD	DT	STRING
SZFILE	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	~
AM	ACCESS_MODE																			
PRESULT	POINTER TO RTS_IEC_RESULT																			
SysFileOpen	RTS_IEC_HANDLE																			

### ⊙ Program demo

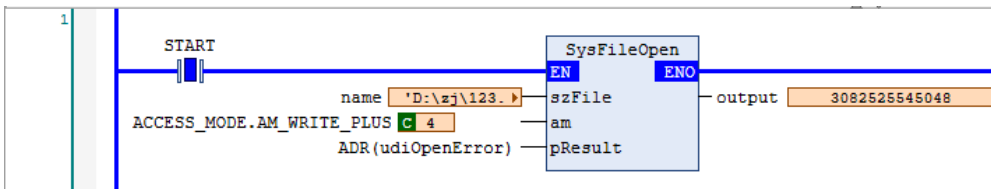
#### ST:

```

1 IF start THEN
2   SysFileOpen(szFile:=name, am:=SysFile.AM_WRITE_PLUS, pResult:=ADR(udiopenererror));
3   //start:=FALSE;
4 END IF

```

#### LD:



### ⊙ Functional Description

The data type of "AM" is ACCESS\_MODE and its meaning is shown below:

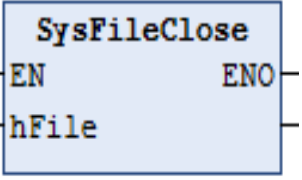
Enumerated values	Descriptions
AM_READ	Opens an existing file with read permissions. If the file does not exist, opening fails.
AM_WRITE	Creates a new file with write access. If the file does exist, discard the contents
AM_APPEND	Opens an existing file using Append (write-only) access. If the file does not exist, the open fails
AM_READ_PLUS	Opens an existing file with read/write access. If the file does not exist, the open fails
AM_WRITE_PLUS	Creates a new file with read/write access. If the file does exist, discard the contents
AM_APPEND_PLUS	Open an existing file with Append (read/write) access. If the file does not exist, Open will create a new file.

1. Please add the "sysfile" library first.
2. This instruction is used to open or create a file under the specified path, the file path and file name are specified by szFile; the file open mode is specified by am; the file handle is returned, which is used as the input of the function blocks such as WriteFile; for the error code of OpenFile, please refer to the library CmpErrors,3.3.1.40(System).
3. There are two libraries that support file-related operations, SYSFILE and CAA.FILE, and the difference between the two is:
  - 1). Difference in file access mechanism, SYSFILE is implemented for synchronous access while CAA .FIFLE is for asynchronous access.
  - 2). Simulation differences, SYSFILE can be simulated while CAA .FIFLE can not be simulated.
  - 3). SysFile function type, File Access for the function block type.
  - 4). File path difference, SYSFILE prohibits the use of backslash.

### 3.12.3 SysFileClose

Close File Operation.

⊙ **Command Format**

Instruction	Name	FB/FC	LD	ST	File
SysFileClose	Close file instruction	FC		SysFileClose(hFile);	SysFile

⊙ **Related Variables**

**Input variable**

Input variable	Name	Data type	Range	initialization	Descriptive
hFile	File handle	RTS_IEC_HANDLE	-	-	File handle

**Output variable.**

Output variable	Name	Data type	Range	initialization	Descriptive
SysFileClose	Return value	RTS_IEC_RESULT	-	-	File handles

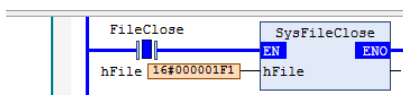
	Boolea n	Bit string				Integer							Real number		Moment, Duration, Date, String				
	BOOL	BYTE	WORD	DWORD	LWORD	USINT	UINT	UDINT	ULINT	SINT	INT	DINT	LINT	REAL	LREAL	TIME	DATE	TOD	DT
hFile						RTS_IEC_HANDLE													
SysFileClose						RTS_IEC_RESULT													

⊙ **Program demo**

**ST:**

```
81 | udiCloseError1 0 :=SysFile.SysFileClose (hFile:=hFile#0000019E4293D7);
```

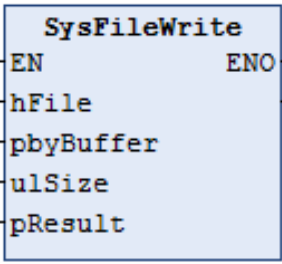
**LD:**



### 3.12.4 SysFileWrite

Writing the contents of a file.

⊙ **Command Format**

Instruction	Name	FB/FC	LD	ST	File
SysFileWrite	Write to a file instruction	FC		SysFileWrite(hFile,pbyBuffer,ulSize,pResult);	SysFile

⊙ **Related Variables**

**Input variable**

Input variable	Name	Data type	Range	initialization	Descriptive
szFile	Filename	RTS_IEC_HANDLE	-	-	Filename
pbyBuffer	Pointer to the contents of the file being written	POINTER TO BYTE	-	-	Pointer to the contents of the file being written
ulSize	Number of bytes written	__XWORD	-	-	Number of bytes written
pResult	Write file content error code pointer	POINTER TO RTS IEC RESULT	-	-	Write file content error code pointer

**Output variable.**

Output variable	Name	Data type	Range	initialization	Descriptive
SysFileWrite	Return value	__XWORD	-	-	Actual size of data written to the file, in bytes

	Boolean	Bit string				Integer							Real number		Moment, Duration, Date, String						
	BOOL	BYTE	WORD	DWORD	LWORD	USINT	UINT	UDINT	ULINT	SINT	INT	DINT	LINT	REAL	LREAL	TIME	DATE	TOD	DT	STRING	
hFile																					
pbyBuffer																					
ulSize	-	-	√	√	√	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
pResult																					
SysFileWrite																					

⊙ **Program demo**

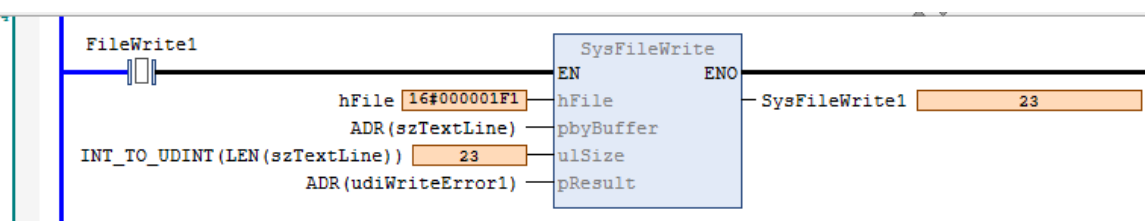
ST:

```

77 | udiWrite 23 := SysFileWrite(hFile:=hFile#0000019E4293D718, pbyBuffer:=ADR(szTextLine This is th ),
78 | ulSize:= INT TO UDINT(LEN(szTextLine This is th )), pResult:= ADR(udiWriteError1#000000000000000000));

```

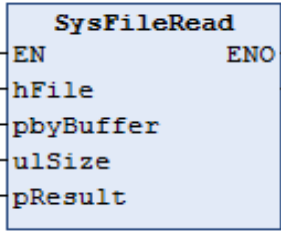
LD:



### 3.12.5 SysFileRead

Read the data content of the file.

⊙ **Command Format**

Instruction	Name	FB/FC	LD	ST	File
SysFileRead	Read file instruction	FC		SysFileRead(hFile, pbyBuffer, ulSize, pResult);	SysFile

⊙ **Related Variables**

**Input variable**

Input variable	Name	Data type	Range	initialization	Descriptive
szFile	Filename	RTS_IEC_HANDLE	-	-	Filename
pbyBuffer	Pointer to the contents of the read file	POINTER TO BYTE	-	-	Pointer to the contents of the read file
ulSize	Number of bytes read from the file	__XWORD			Number of bytes read from the file
pResult	Read file content error code pointer	POINTER TO RTS_IEC_RESULT	-		Read file content error code pointer

**Output variable.**

Output variable	Name	Data type	Range	initialization	Descriptive
SysFileRead	Return value	__XWORD	-	-	The actual size of the read file data, in bytes

	Boolean	Bit string				Integer							Real number		Moment, Duration, Date, String						
	BOOL	BYTE	WORD	DWORD	LWORD	USINT	UINT	UDINT	ULINT	SINT	INT	DINT	LINT	REAL	LREAL	TIME	DATE	TOD	DT	STRING	
hFile																					
pbyBuffer																					
ulSize	-	-	√	√	√	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
pResult																					
SysFileRead																					

⊙ **Program demo**

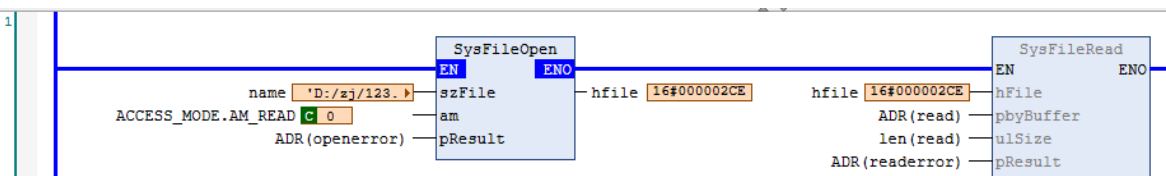
ST:

```

1 SysFileOpen(szFile:=name, am:=SysFile.AM_READ, pResult:=ADR(openererror));
2 hfile:=SysFileOpen(szFile:=name, am:=SysFile.AM_READ, pResult:=ADR(openererror));
3 SysFileRead(hFile:=hfile, pbyBuffer:=ADR(read), ulSize:=LEN(read), pResult:=ADR(readerror));

```

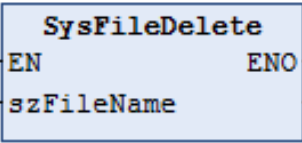
LD:



### 3.12.6 SysFileDelete

Deleting files.

⊙ **Command Format**

Instruction	Name	FB/FC	LD	ST	File
SysFileDelete	Delete file instruction	FC		SysFileDelete(szFileName);	SysFile

⊙ **Related Variables**

**Input variable**

Input variable	Name	Data type	Range	initialization	Descriptive
szFileName	Specify file name	STRING	-	-	Specify the file path and file name

**Output variable.**

Output variable	Name	Data type	Range	initialization	Descriptive
SysFileDelete	Return value	RTS_IEC_RESULT	-	-	Delete File Error Code

	Boolean	Bit string				Integer							Real number		Moment, Duration, Date, String						
	BOOL	BYTE	WORD	DWORD	LWORD	USINT	UINT	UDINT	ULINT	SINT	INT	DINT	LINT	REAL	LREAL	TIME	DATE	TOD	DT	STRING	
szFileName	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
SysFileDelete	RTS_IEC_RESULT																				

⊙ **Program demo**

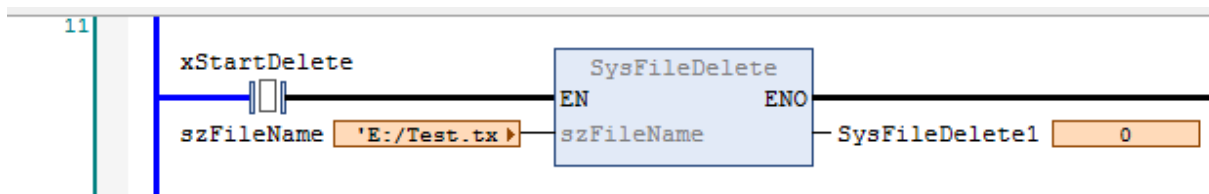
ST:

```

SysFileDelete (szFileName := 'E:/Test.txt');

```

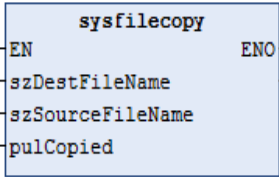
LD:



### 3.12.7 SysFileCopy

Copy file operations.

#### ⊙ Command Format

Instruction	Name	FB/FC	LD	ST	File
SysFileCopy	Copy file instruction	FC		SysFileCopy(szDestFileName, szSourceFileName, pulCopied);	SysFile

#### ⊙ Related Variables

##### Input variable

Input variable	Name	Data type	Range	initialization	Descriptive
szDestFileName	Target documents	STRING	-	-	Target file path and file name
szsourceFileName	Source files	STRING	-	-	Source file path and file name
pulCopied	Pointer to actual read data size	POINTER TO _XWORD			Pointer to actual read data size; unit: byte

##### Output variable.

Output variable	Name	Data type	Range	initialization	Descriptive
SysFileCopy	Return value	RTS_IEC_RESULT	-	-	Copy File Error Code

	Boolean	Bit string				Integer							Real number		Moment, Duration, Date, String						
	BOOL	BYTE	WORD	DWORD	LWORD	USINT	UINT	UDINT	ULINT	SINT	INT	DINT	LINT	REAL	LREAL	TIME	DATE	TOD	DT	STRING	
szDestFileName	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	√
szSourceFileName	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	√
pulCopied	POINTER TO _XWORD																				
SysFileCopy	RTS_IEC_RESULT																				

#### ⊙ Program demo

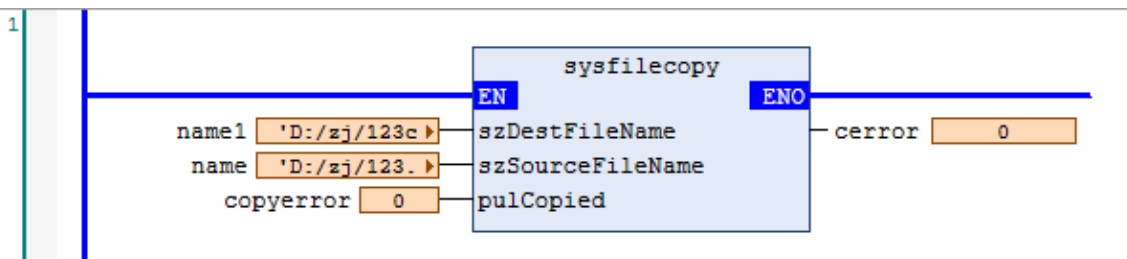
ST:

```

1 | SysFileCopy(szDestFileName:=name1, szSourceFileName:=name, pulCopied:=copyerror);
2 | RETURN

```

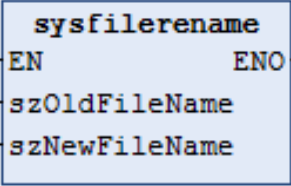
LD:



### 3.12.8 SysFileRename

Rename File Operations.

#### ⊙ Command Format

Instruction	Name	FB/FC	LD	ST	File
SysFileRename	Rename a file instruction	FC		SysFileCopy(szOldFileName,szNewFileName);	SysFile

#### ⊙ Related Variables

##### Input variable

Input variable	Name	Data type	Range	initialization	Descriptive
szOldFileName	Source file	STRING	-	-	Old filename
szNewFileName	Target documents	STRING	-	-	New filename

##### Output variable.

Output variable	Name	Data type	Range	initialization	Descriptive
SysFileRename	Return value	RTS_IEC_RESULT	-	-	Rename File Error Code

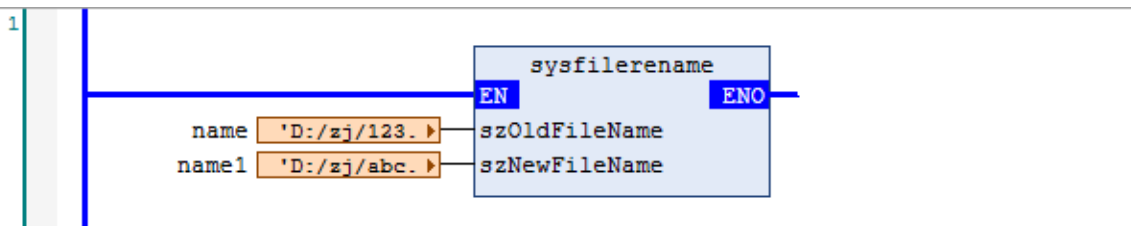
	Boolean	Bit string				Integer						Real number		Moment, Duration, Date, String							
	BOOL	BYTE	WORD	DWORD	LWORD	USINT	UINT	UDINT	ULINT	SINT	INT	DINT	LINT	REAL	LREAL	TIME	DATE	TOD	DT	STRING	
szOldFileName	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	✓
szNewFileName	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	✓
SysFileRename	RTS_IEC_RESULT																				

#### ⊙ Program demo

ST:

```
1 | SysFileRename (szOldFileName:=name, szNewFileName:=name1);
```

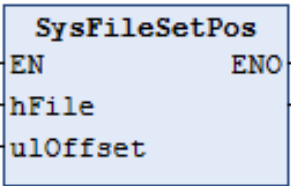
LD:



### 3.12.9 SysFileSetPos

Set the file pointer to the specified location.

#### ⊙ Command Format

Instruction	Name	FB/FC	LD	ST	File
SysFileSetPos	Setting the file read/write location instruction	FC		SysFileCopy(hFile,ulOffset);	SysFile

#### ⊙ Related Variables

##### Input variable

Input variable	Name	Data type	Range	initialization	Descriptive
hFile	File handles	RTS_IEC_HANDLE	-	-	File handles
ulOffset	Offset position	_XWORD	-	-	File Pointer Offset

##### Output variable.

Output variable	Name	Data type	Range	initialization	Descriptive
SysFileSetPos	Return value	RTS_IEC_RESULT	-	-	Set file pointer to specified location error code

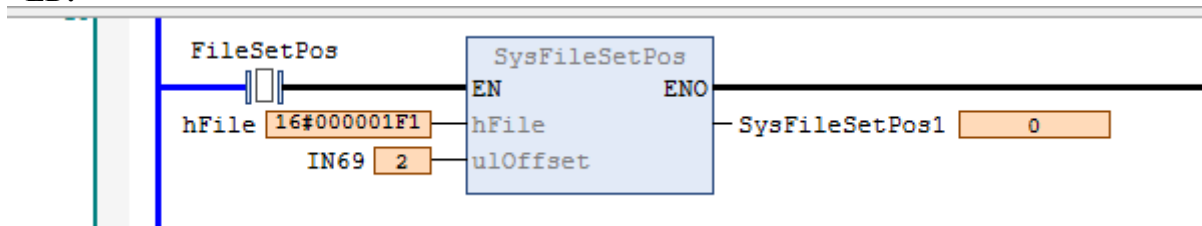
	Boolean	Bit string				Integer							Real number		Moment, Duration, Date, String					
	BOOL	BYTE	WORD	DWORD	LWORD	USINT	UINT	UDINT	ULINT	SINT	INT	DINT	LINT	REAL	LREAL	TIME	DATE	TOD	DT	STRING
hFile																				
uloffset	-	-	√	√	√	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
SysFileSetPos																				

#### ⊙ Program demo

ST:

```
76 | udiPosError3 := SysFileSetPos (hFile:= hFile#0000019E4293D);ulOffset:=IN69 );
```

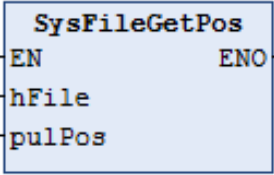
LD:



### 3.12.10 SysFileGetPos

Get the location of the file pointer command.

#### ⊙ Command Format

Instruction	Name	FB/FC	LD	ST	File
SysFileGetPos	Read file read/write location instruction	FC		SysFileGetPos(hFile, pulPos);	SysFile

#### ⊙ Related Variables

##### Input variable

Input variable	Name	Data type	Range	initialization	Descriptive
hFile	File handles	RTS_IEC_HANDLE	-	-	File handles
pulPos	Offset position	POINTER TO _XWORD	-	-	Get file pointer offset

##### Output variable.

Output variable	Name	Data type	Range	initialization	Descriptive
SysFileGetPos	Return value	RTS_IEC_RESULT	-	-	Get file pointer to specified location error code

	Boolean	Bit string				Integer							Real number		Moment, Duration, Date, String					
	BOOL	BYTE	WORD	DWORD	LWORD	USINT	UINT	UDINT	ULINT	SINT	INT	DINT	LINT	REAL	LREAL	TIME	DATE	TOD	DT	STRING
hFile																				
uloffset	-	-	√	√	√	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
SysFileGetPos																				

#### ⊙ Program demo

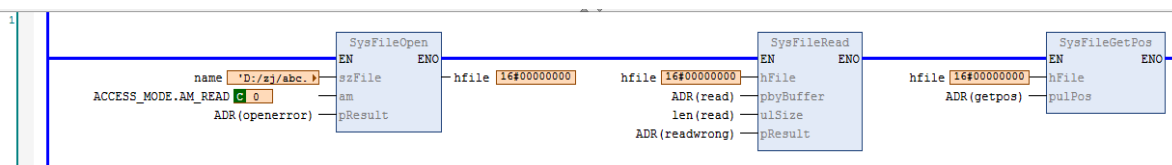
ST:

```

1 SysFileOpen(szFile:=name , am:=SysFile.AM_READ , pResult:=ADR(openererror));
2 hfile:=SysFileOpen(szFile:=name , am:=SysFile.AM_READ , pResult:=ADR(openererror));
3 SysFileRead(hFile:=hfile, pbyBuffer:=ADR(read), ulSize:=len(read), pResult:=ADR(readwrong));
4 SysFileGetPos(hFile:=hfile, pulPos:=ADR(getpos)); RETURN

```

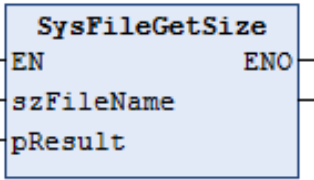
LD:



### 3.12.11 SysFileGetSize

Get file size command.

⊙ **Command Format**

Instruction	Name	FB/FC	LD	ST	File
SysFileGetSize	Get file size instruction	FC		SysFileGetSize(szFileName, pResult);	SysFile

⊙ **Related Variables**

**Input variable**

Input variable	Name	Data type	Range	initialization	Descriptive
szFileName	File name	STRING	-	-	File name
pResult	Offset position	POINTER TO RTS_IEC_RESULT	-	-	Get a pointer to the file size command error code

**Output variable.**

Output variable	Name	Data type	Range	initialization	Descriptive
SysFileGetSize	Return value	_XWORD	-	-	Get file pointer to specified location error code

	Boolean	Bit string					Integer							Real number		Moment, Duration, Date, String				
	BOOL	BYTE	WORD	DWORD	LWORD	USINT	UINT	UDINT	ULINT	SINT	INT	DINT	LINT	REAL	LREAL	TIME	DATE	TOD	DT	STRING
szFileName	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
pResult	POINTER TO RTS_IEC_RESULT																			
SysFileGetSize	-	-	√	√	√	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-

⊙ **Program demo**

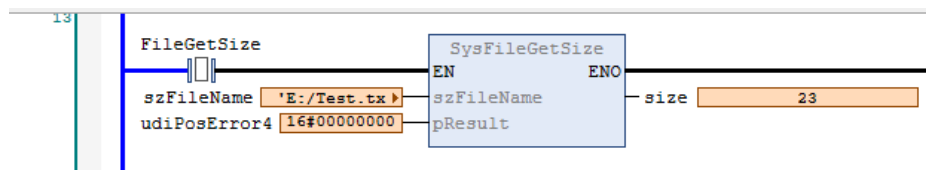
**ST:**

```

80 | size 23 := SysFileGetSize(szFileName:='E:/Test.txt', pResult:=udiPosError4#000000000000);

```

**LD:**



### 3.13 Analog calculation.

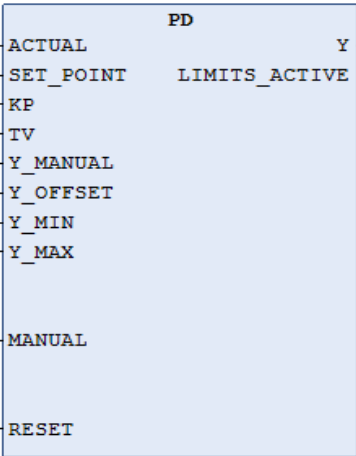
#### 3.13.1 Instruction List

Instruction Type	Name	FB/FC	Functionality
Analog calculation	PD	FB	Proportional Differential Control
	PID	FB	Proportional integral/differential control
	PID_FIXCYCLE	FB	Proportional Integral Differential Control with manually settable cyclic cycles

#### 3.13.2 PD

Proportional Differential Controller.

##### ⊙ Command Format

Instruction	Name	FB/FC	LD	ST	File
PD	Proportional Differential Control instruction	FC		PD(ACTUAL:=, SET_POINT:=, KP:=, TV:=, Y_MANUAL:=, Y_OFFSET:=, Y_MIN:=, Y_MAX:=, MANUAL:=, RESET:=, Y=>, LIMITS_ACTIVE= > );	Util

##### ⊙ Related Variables

###### Input variable

Input variable	Name	Data type	Range	initialization	Descriptive
ACTUAL	Control variable current value	REAL	-	0	Control variable current value
SET_POINT	Setting value	REAL	-	0	Setting value
KP	Scaling factor	REAL	-	0	Scaling factor
TV	Differential time	REAL	-	0	Differential time
Y_MANUAL	Manual output value	REAL	-	0	Manual output value
Y_OFFSET	Offset of output Y	REAL	-	0	Offset of output Y
Y_MIN	Minimum value of output Y	REAL	-	0	Minimum value of output Y
Y_MAX	Maximum value of Y is output	REAL	-	0	Maximum value of Y is output
MANUAL	Manual output	BOOL	FALSE - TRUE	FALSE	Manual output
RESET	Reset	BOOL	FALSE - TRUE	FALSE	Reset

**Output variable.**

Output variable	Name	Data type	Range	initialization	Descriptive
Y	Output value	REAL	-	0	Output value
LIMITS_ACTIVE	Reaching a given limit	BOOL	FALSE-TRUE	FALSE	Reaching a given limit

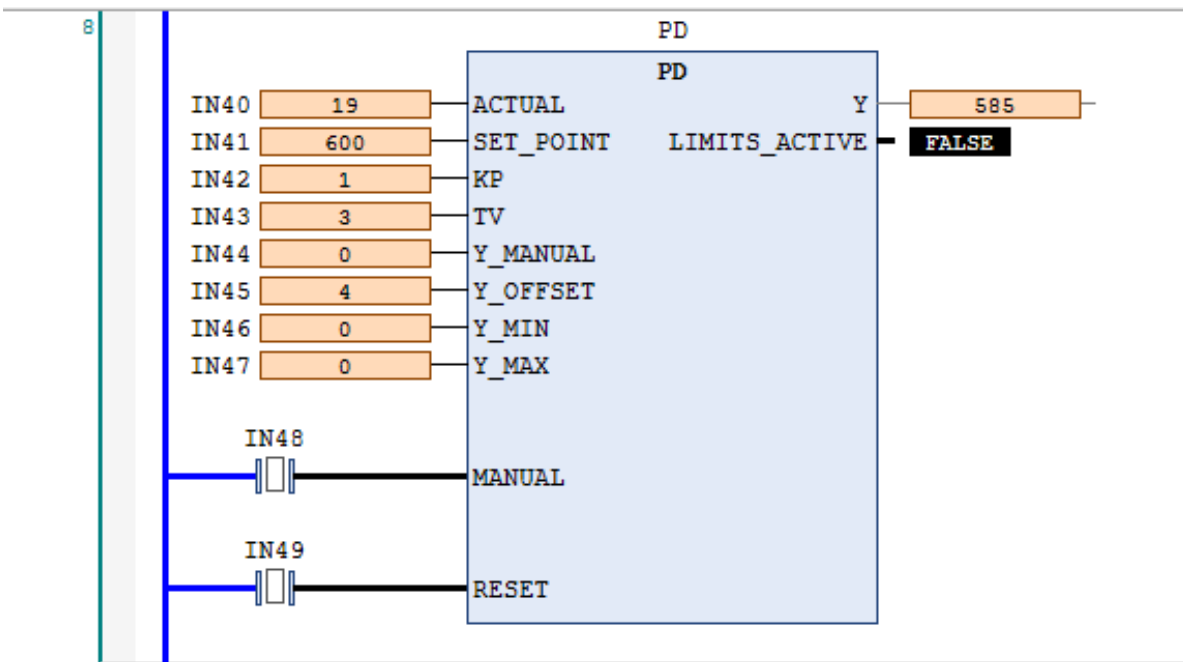
	Boolean	Bit string					Integer							Real number		Moment, Duration, Date, String				
	BOOL	BYTE	WORD	DWORD	LWORD	USINT	UINT	UDINT	ULINT	SINT	INT	DINT	LINT	REAL	LREAL	TIME	DATE	TOD	DT	STRING
ACTUAL	-	-	-	-	-	-	-	-	-	-	-	-	-	✓	-	-	-	-	-	-
SET_POINT	-	-	-	-	-	-	-	-	-	-	-	-	-	✓	-	-	-	-	-	-
KP	-	-	-	-	-	-	-	-	-	-	-	-	-	✓	-	-	-	-	-	-
TV	-	-	-	-	-	-	-	-	-	-	-	-	-	✓	-	-	-	-	-	-
Y_MANUAL	-	-	-	-	-	-	-	-	-	-	-	-	-	✓	-	-	-	-	-	-
Y_OFFSET	-	-	-	-	-	-	-	-	-	-	-	-	-	✓	-	-	-	-	-	-
Y_MIN	-	-	-	-	-	-	-	-	-	-	-	-	-	✓	-	-	-	-	-	-
Y_MAX	-	-	-	-	-	-	-	-	-	-	-	-	-	✓	-	-	-	-	-	-
MANUAL	✓	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
RESET	✓	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
Y	-	-	-	-	-	-	-	-	-	-	-	-	-	✓	-	-	-	-	-	-
LIMITS_ACTIVE	✓	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-

**⊙ Program demo**  
**ST:**

```

21 //PD
22 PD(ACTUAL:=IN40_19, SET_POINT:=IN41_600, KP:=IN42_1, TV:=IN43_3,
23 Y_MANUAL:=IN44_0, Y_OFFSET:=IN45_4, Y_MIN:=IN46_0, Y_MAX:=IN47_0,
24 MANUAL:=IN48_FALSE, RESET:=IN49_FALSE, Y:=Y1_585, LIMITS_ACTIVE=>);
25

```

**LD:**


### 3.13.3 PID

Proportional Integral Differential Controller.

#### ⊙ Command Format

Instruction	Name	FB/FC	LD	ST	File
PID	Proportional integral/differential control instruction	FC	<div style="border: 1px solid black; padding: 5px;"> <p style="text-align: center;">PID</p> <pre> -EN          ENO- -ACTUAL      Y- -SET_POINT   LIMITS_ACTIVE- -KP          OVERFLOW- -TN -TV -Y_MANUAL -Y_OFFSET -Y_MIN -Y_MAX -MANUAL -RESET </pre> </div>	PID (ACTUAL:=, SET_POINT:=, KP:=, TN:=, TV:=, Y_MANUAL:=, Y_OFFSET:=, Y_MIN:=, Y_MAX:=, MANUAL:=, RESET:=, Y=>, LIMITS_ACTIVE= > OVERFLOW);	Util

#### ⊙ Related Variables

##### Input variable

Input variable	Name	Data type	Range	initialization	Descriptive
ACTUAL	Control variable current value	REAL	-	0	Control variable current value
SET_POINT	Setting value	REAL	-	0	Setting value
KP	Scaling factor	REAL	-	0	Scaling factor
TN	Points time	REAL	-	0	Reset time, mutual unit gain of I-shaped components
TV	Differential time	REAL	-	0	Differential action time, unit gain of part D
Y_MANUAL	Manual output value	REAL	-	0	Manual output value
Y_OFFSET	Offset of output Y	REAL	-	0	Offset of output Y
Y_MIN	Minimum value of output Y	REAL	-	0	Minimum value of output Y
Y_MAX	Maximum value of Y is output	REAL	-	0	Maximum value of Y is output
MANUAL	Manual output	BOOL	FALSE - TRUE	FALSE	Manual output
RESET	Reset	BOOL	FALSE - TRUE	FALSE	Reset

**Output variable.**

Output variable	Name	Data type	Range	initialization	Descriptive
Y	Output value	REAL	-	0	Output value
LIMITS_ACTIVE	Reaching a given limit	BOOL	FALSE-TRUE	FALSE	Reaching a given limit
OVERFLOW	Overflow marker	BOOL	FALSE-TRUE	FALSE	Overflow marker

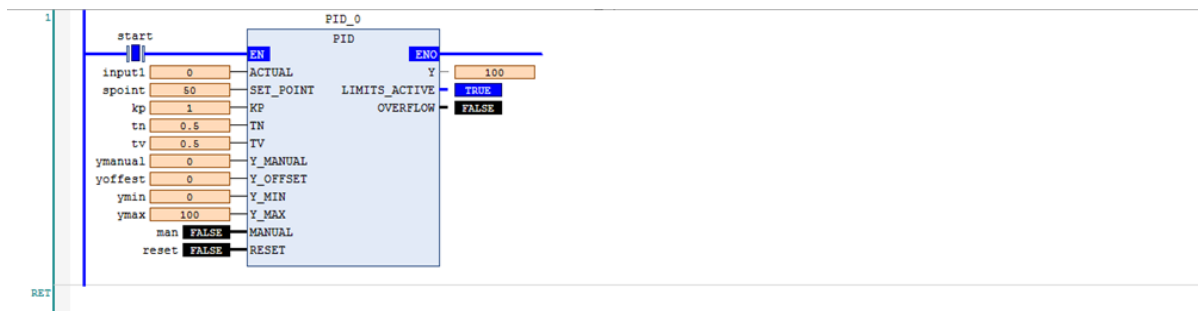
	Boo lea n	Bit string					Integer							Real number		Moment, Duration, Date, String				
	BOO L	BY TE	WO RD	DWO RD	LWO RD	US INT	UI NT	UD INT	UL INT	SI NT	IN T	DI NT	LI NT	RE AL	LR EAL	TI ME	DA TE	TO D	DT	STR ING
ACTUAL	-	-	-	-	-	-	-	-	-	-	-	-	-	✓	-	-	-	-	-	-
SET_POINT	-	-	-	-	-	-	-	-	-	-	-	-	-	✓	-	-	-	-	-	-
KP	-	-	-	-	-	-	-	-	-	-	-	-	-	✓	-	-	-	-	-	-
TN	-	-	-	-	-	-	-	-	-	-	-	-	-	✓	-	-	-	-	-	-
TV	-	-	-	-	-	-	-	-	-	-	-	-	-	✓	-	-	-	-	-	-
Y_MANUAL	-	-	-	-	-	-	-	-	-	-	-	-	-	✓	-	-	-	-	-	-
Y_OFFSET	-	-	-	-	-	-	-	-	-	-	-	-	-	✓	-	-	-	-	-	-
Y_MIN	-	-	-	-	-	-	-	-	-	-	-	-	-	✓	-	-	-	-	-	-
Y_MAX	-	-	-	-	-	-	-	-	-	-	-	-	-	✓	-	-	-	-	-	-
MANUAL	✓	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
RESET	✓	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
Y	-	-	-	-	-	-	-	-	-	-	-	-	-	✓	-	-	-	-	-	-
LIMITS_ACTIVE	✓	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
OVERFLOW	✓	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-

**⊙ Program demo**
**ST:**

```

1  @ PID_0(
2  ACTUAL:=input1 ,
3  SET_POINT:=spoint ,
4  KP:=kp ,
5  TN:=tn ,
6  TV:=tv ,
7  Y_MANUAL:=ymanual,
8  Y_OFFSET:=yoffset,
9  Y_MIN:=ymin,
10 Y_MAX:=ymax,
11 MANUAL:=man,
12 RESET:=reset,
13 Y=> ,
14 LIMITS_ACTIVE=> ,
15 @ OVERFLOW=> );RETURN

```

**LD:**


### 3.13.4 PID\_FIXCYCLE

Proportional Integral Differential Controller.

☉ **Command Format**

Instruction	Name	FB/FC	LD	ST	File
PID_FIXCYCLE	Proportional Integral Differential Control instruction	FC	<pre> PID_FIXCYCLE   ACTUAL          Y   SET_POINT      LIMITS_ACTIVE   KP              OVERFLOW   TN   TV   Y_MANUAL   Y_OFFSET   Y_MIN   Y_MAX   MANUAL   RESET   CYCLE           </pre>	<pre> PD(ACTUAL:=, SET_POINT:=,   KP:=,   TN:=,   TV:=, Y_MANUAL:=, Y_OFFSET:=,   Y_MIN:=,   Y_MAX:=, MANUAL:=, RESET:=, CYCLE:=,   Y=&gt;, LIMITS_ACTIVE= &gt; OVERFLOW=&gt;);           </pre>	Util

☉ **Related Variables**

**Input variable**

Input variable	Name	Data type	Range	initialization	Descriptive
ACTUAL	Control variable current value	REAL	-	0	Control variable current value
SET_POINT	Setting value	REAL	-	0	Setting value
KP	Scaling factor	REAL	-	0	Scaling factor
TN	Points time	REAL	-	0	Reset time, mutual unit gain of I-shaped components
TV	Differential time	REAL	-	0	Differential action time, unit gain of part D
Y_MANUAL	Manual output value	REAL	-	0	Manual output value
Y_OFFSET	Offset of output Y	REAL	-	0	Offset of output Y
Y_MIN	Minimum value of output Y	REAL	-	0	Minimum value of output Y
Y_MAX	Maximum value of Y is output	REAL	-	0	Maximum value of Y is output
MANUAL	Manual output	BOOL	FALSE - TRUE	FALSE	Manual output
RESET	Reset	BOOL	FALSE - TRUE	FALSE	Reset
CYCLE	Sampling cycle time	REAL	-	0	Sampling cycle time

**Output variable.**

Output variable	Name	Data type	Range	initialization	Descriptive
Y	Output value	REAL	-	0	Output value
LIMITS_ACTIVE	Reaching a given limit	BOOL	FALSE-TRUE	FALSE	Reaching a given limit
OVERFLOW	Overflow marker	BOOL	FALSE-TRUE	FALSE	Overflow marker

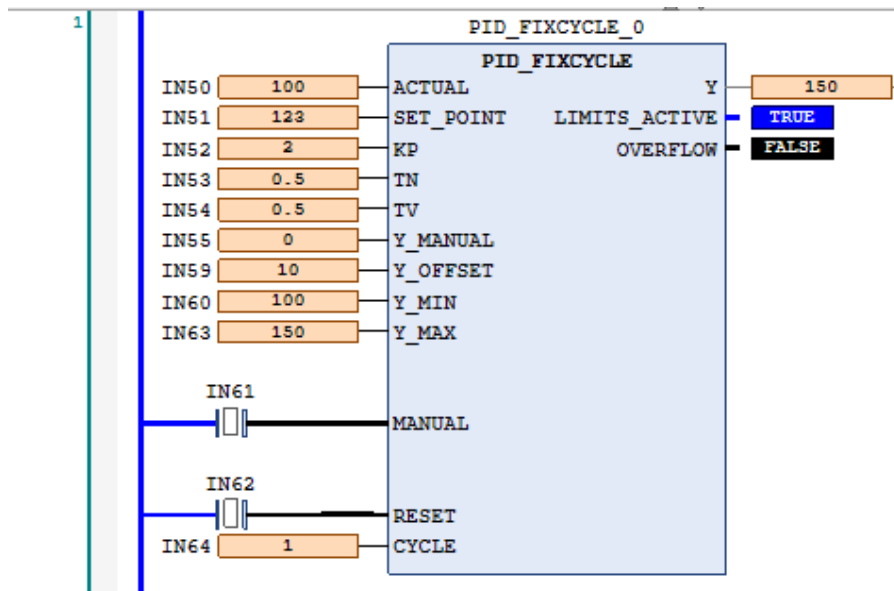
	Boo	Bit string					Integer							Real number		Moment, Duration, Date, String				
	le	BY	WO	DWO	LWO	US	UI	UD	UL	SI	IN	DI	LI	RE	LR	TI	DA	TO	DT	STR
	an	TE	RD	RD	RD	INT	NT	INT	INT	INT	INT	INT	INT	AL	REAL	ME	TE	OD	DT	ING
ACTUAL	-	-	-	-	-	-	-	-	-	-	-	-	-	✓	-	-	-	-	-	-
SET_POINT	-	-	-	-	-	-	-	-	-	-	-	-	-	✓	-	-	-	-	-	-
KP	-	-	-	-	-	-	-	-	-	-	-	-	-	✓	-	-	-	-	-	-
TN	-	-	-	-	-	-	-	-	-	-	-	-	-	✓	-	-	-	-	-	-
TV	-	-	-	-	-	-	-	-	-	-	-	-	-	✓	-	-	-	-	-	-
Y_MANUAL	-	-	-	-	-	-	-	-	-	-	-	-	-	✓	-	-	-	-	-	-
Y_OFFSET	-	-	-	-	-	-	-	-	-	-	-	-	-	✓	-	-	-	-	-	-
Y_MIN	-	-	-	-	-	-	-	-	-	-	-	-	-	✓	-	-	-	-	-	-
Y_MAX	-	-	-	-	-	-	-	-	-	-	-	-	-	✓	-	-	-	-	-	-
MANUAL	✓	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
RESET	✓	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
CYCLE	-	-	-	-	-	-	-	-	-	-	-	-	-	✓	-	-	-	-	-	-
Y	-	-	-	-	-	-	-	-	-	-	-	-	-	✓	-	-	-	-	-	-
LIMITS_ACTIVE	✓	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
OVERFLOW	✓	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-

**⊙ Program demo**
**ST:**

```

14 //PID_FIXCYCLE
15 PID_FIXCYCLE(ACTUAL:=IN50[19], SET_POINT:=IN51[21], KP:=IN52[1],
16 TN:=IN53[10], TV:=IN54[0], Y_MANUAL:=IN55[0], Y_OFFSET:=IN59[0],
17 Y_MIN:=IN60[0], Y_MAX:=IN61[0], MANUAL:=IN62[FALSE], RESET:=IN63[FALSE],
18 CYCLE:=IN64[0.05], Y:=Y2[49.2], LIMITS_ACTIVE=>, OVERFLOW=>);

```

**LD:**


### 3.14 BCD code conversion.

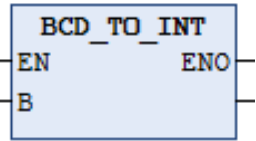
#### 3.14.1 Instruction List

Instruction Type	Name	FB/FC	Functionality
BCD code conversion	BCD TO INT	FC	BCD to Integer
	INT TO BCD	FC	Integer to BCD
	BCD TO BYTE	FC	BCD to byte
	BYTE TO BCD	FC	Byte to BCD
	BCD TO WORD	FC	BCD to Word
	WORD TO BCD	FC	Word to BCD
	BCD TO DWORD	FC	BCD to double word
	DWORD TO BCD	FC	Double word to BCD

#### 3.14.2 BCD\_TO\_INT

Convert BCD code of source data to INT type data.

##### ⊙ Command Format

Instruction	Name	FB/FC	LD	ST	File
BCD_TO_INT	BCD to Integer instruction	FC		Output: =BCD_TO_INT(B )	Util

##### ⊙ Related Variables

###### Input variable

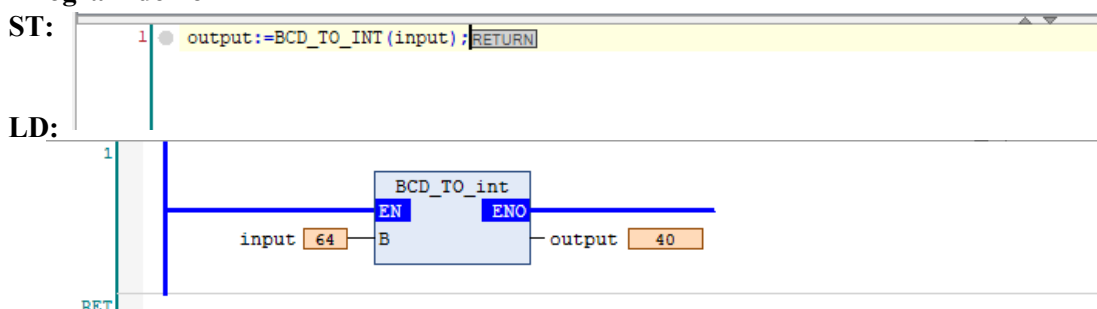
Input variable	Name	Data type	Range	initialization	Descriptive
B	Input data	BYTE	0-255	0	Input data

###### Output variable.

Output variable	Name	Data type	Range	initialization	Descriptive
BCD_TO_INT	Output data	INT	-	-	Output data

	Boole an	Bit string					Integer						Real number		Moment, Duration, Date, String					
	BOOL	BYTE	WORD	DWORD	LWORD	USINT	UINT	UDINT	ULINT	SINT	INT	DINT	LINT	REAL	LREAL	TIME	DATE	TOD	DT	STRING
B	-	✓	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
BCD_TO_INT	-	-	-	-	-	-	-	-	-	✓	-	-	-	-	-	-	-	-	-	-

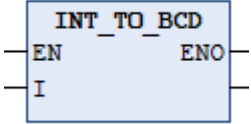
##### ⊙ Program demo



### 3.14.3 INT\_TO\_BCD

Convert INT type data of source data to BCD code.

⊙ **Command Format**

Instruction	Name	FB/FC	LD	ST	File
INT_TO_BCD	Integer to BCD instruction	FC		Output: = INT_TO_BCD (I)	Util

⊙ **Related Variables**

**Input variable**

Input variable	Name	Data type	Range	initialization	Descriptive
I	Input data	INT	0-99	0	Input data

**Output variable.**

Output variable	Name	Data type	Range	initialization	Descriptive
INT_TO_BCD	Output data	BYTE	0-255	0	Output data

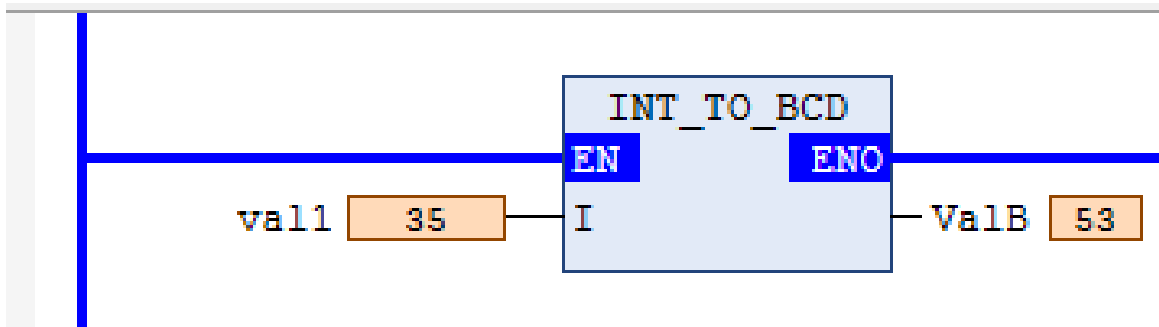
	Boole an	Bit string					Integer						Real number		Moment, Duration, Date, String					
	BOOL	BYTE	WORD	DWORD	LWORD	USINT	UINT	UDINT	ULINT	SINT	INT	DINT	LINT	REAL	LREAL	TIME	DATE	TOD	DT	STRING
I	-	-	-	-	-	-	-	-	-	-	✓	-	-	-	-	-	-	-	-	-
INT_TO_BCD	-	✓	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-

⊙ **Program demo**

ST:

```
1 | ● OUTPUT 53 := INT_TO_BCD (I 35) ; RETURN
```

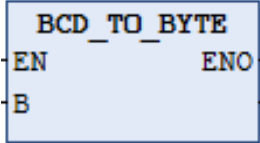
LD:



### 3.14.4 BCD\_TO\_BYTE

Convert BCD code of source data to BYTE type data.

⊙ **Command Format**

Instruction	Name	FB/FC	LD	ST	File
BCD_TO_BYTE	BCD to byte instruction	FC		Output: = BCD_TO_BYTE (B)	Util

⊙ **Related Variables**

**Input variable**

Input variable	Name	Data type	Range	initialization	Descriptive
B	Input data	BYTE	0-255	0	Input data

**Output variable.**

Output variable	Name	Data type	Range	initialization	Descriptive
BCD_TO_BYTE	Output data	BYTE	0-255	0	Output data

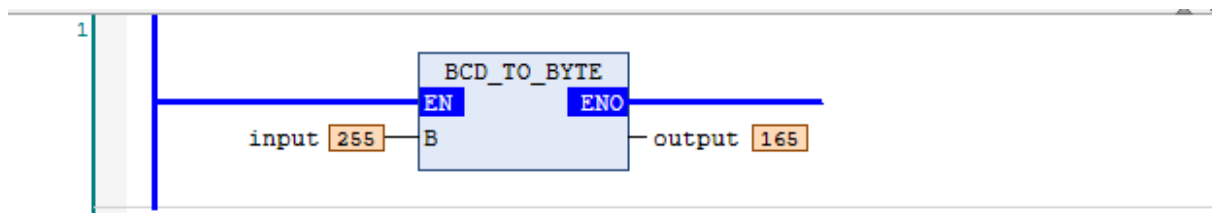
	Boo	Bit string				Integer							Real number		Moment, Duration, Date, String					
	lea	BYTE	WORD	DWORD	LWORD	USINT	UINT	UDINT	ULINT	SINT	INT	DINT	LINT	REAL	LREAL	TIME	DATE	TOD	DT	STRING
B	-	√	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
BCD_TO_BYTE	-	√	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-

⊙ **Program demo**

**ST:**

```
1 ● output:=BCD_TO_BYTE(input);RETURN
```

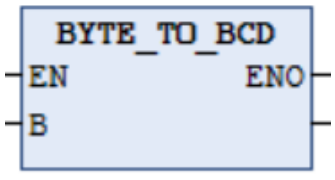
**LD:**



### 3.14.5 BYTE\_TO\_BCD

Convert BYTE type data of source data to BCD code.

⊙ **Command Format**

Instruction	Name	FB/FC	LD	ST	File
BYTE_TO_BCD	Byte to BCD instruction	FC		Output: = BYTE_TO_BCD (B)	Util

⊙ **Related Variables**

**Input variable**

Input variable	Name	Data type	Range	initialization	Descriptive
B	Input data	BYTE	0–255	0	Input data

**Output variable.**

Output variable	Name	Data type	Range	initialization	Descriptive
BYTE_TO_BCD	Output data	BYTE	0–255	0	Output data

	Boolean	Bit string				Integer							Real number		Moment, Duration, Date, String					
	BOOL	BYTE	WORD	DWORD	LWORD	USINT	UINT	UDINT	ULINT	SINT	INT	DINT	LINT	REAL	LREAL	TIME	DATE	TOD	DT	STRING
B	-	✓	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
Output	-	✓	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-

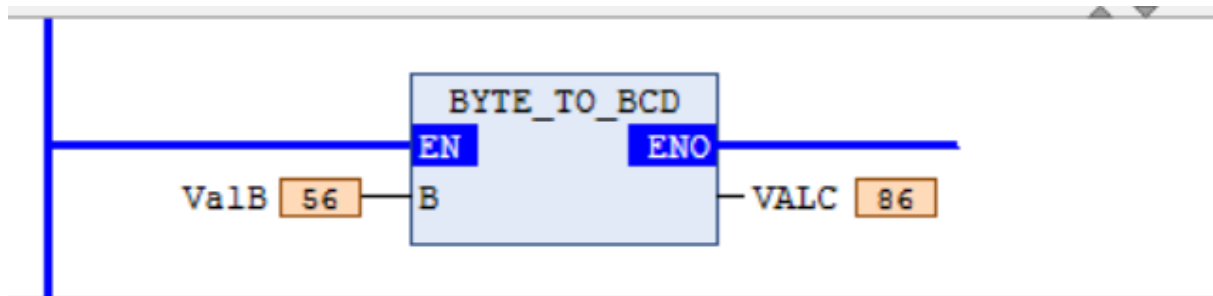
⊙ **Program demo**

ST:

```

OUTPUT 86 := BYTE_TO_BCD (B 56) ; RETURN
    
```

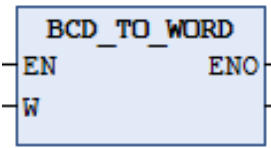
LD:



### 3.14.6 BCD\_TO\_WORD

Convert BCD code of source data to WORD type data.

⊙ **Command Format**

Instruction	Name	FB/FC	LD	ST	File
BCD_TO_WORD	BCD to Word instruction	FC		Output: = BCD_TO_WORD (B)	Util

⊙ **Related Variables**

**Input variable**

Input variable	Name	Data type	Range	initialization	Descriptive
W	Input data	WORD	0–FFFF	0	Input data

**Output variable.**

Output variable	Name	Data type	Range	initialization	Descriptive
BCD_TO_WORD	Output data	WORD	0–FFFF	0	Output data

	Boolean	Bit string				Integer							Real number		Moment, Duration, Date, String						
		BOOL	BYTE	WORD	DWORD	LWORD	USINT	UINT	UDINT	ULINT	SINT	INT	DINT	LINT	REAL	LREAL	TIME	DATE	TOD	DT	STRING
B	-	-	✓	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
BCD_TO_WORD	-	-	✓	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-

⊙ **Program demo**

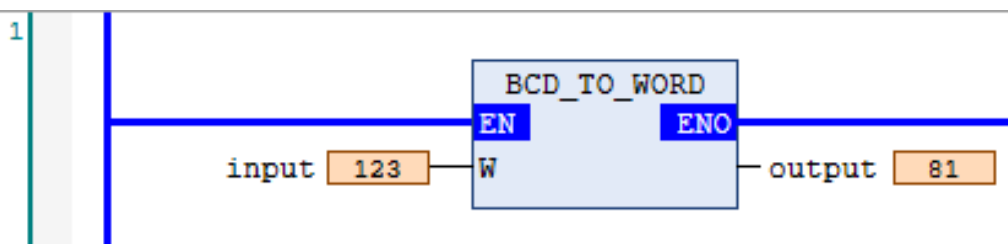
ST:

```

1  ● output:=BCD_TO_WORD(input);
2  ● RETURN

```

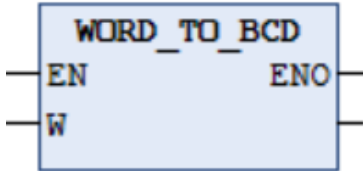
LD:



### 3.14.7 WORD\_TO\_BCD

Convert BCD code of source data to WORD type data.

⊙ **Command Format**

Instruction	Name	FB/FC	LD	ST	File
WORD_TO_BCD	Word to BCD instruction	FC		Output: = WORD_TO_BCD (W)	Util

⊙ **Related Variables**

**Input variable**

Input variable	Name	Data type	Range	initialization	Descriptive
W	Input data	WORD	0–FFFF	0	Input data

**Output variable.**

Output variable	Name	Data type	Range	initialization	Descriptive
WORD_TO_BCD	Output data	WORD	0–FFFF	0	Output data

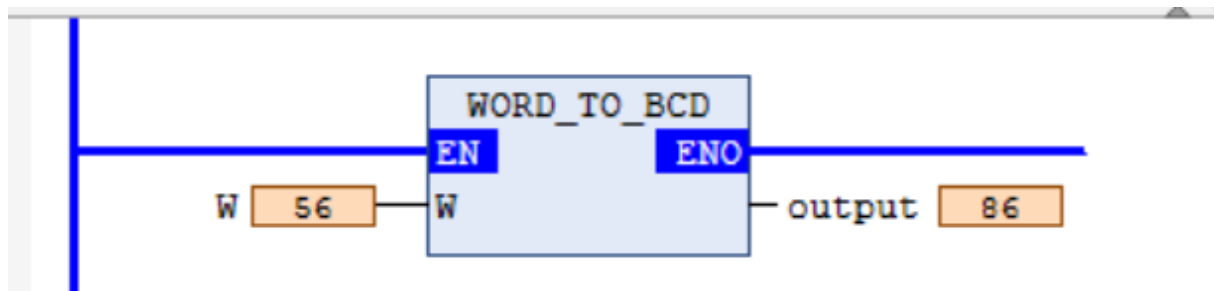
	Boolean	Bit string				Integer							Real number		Moment, Duration, Date, String					
	BOOL	BYTE	WORD	DWORD	LWORD	USINT	UINT	UDINT	ULINT	SINT	INT	DINT	LINT	REAL	LREAL	TIME	DATE	TOD	DT	STRING
W	-	-	✓	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
WORD_TO_BCD	-	-	✓	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-

⊙ **Program demo**

ST:

```
wOUTPUT 86 :=WORD_TO_BCD (W 56) ; RETURN
```

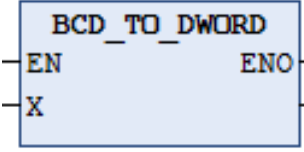
LD:



### 3.14.8 BCD\_TO\_DWORD

Convert BCD code of source data to DWORD type data.

⊙ **Command Format**

Instruction	Name	FB/FC	LD	ST	File
BCD_TO_DWORD	BCD to double word instruction	FC		Output: = BCD_TO_DWORD(X)	Util

⊙ **Related Variables**

**Input variable**

Input variable	Name	Data type	Range	initialization	Descriptive
X	Input data	DWORD	0–FFFF FFFF	0	Input data

**Output variable.**

Output variable	Name	Data type	Range	initialization	Descriptive
BCD_TO_DWORD	Output data	DWORD	0–FFFF FFFF	0	Output data

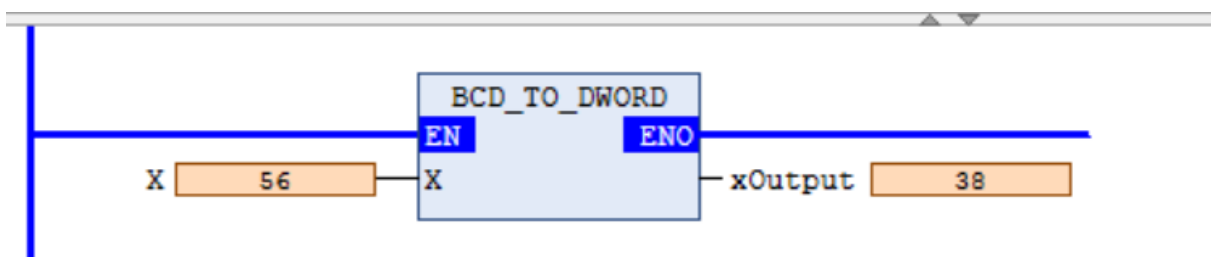
	Boolean	Bit string				Integer							Real number		Moment, Duration, Date, String					
	BOOL	BYTE	WORD	DWORD	LWORD	USINT	UINT	UDINT	ULINT	SINT	INT	DINT	LINT	REAL	LREAL	TIME	DATE	TOD	DT	STRING
X	-	-	-	✓	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
BCD_TO_DWORD	-	-	-	✓	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-

⊙ **Program demo**

ST:

```
xOUTPUT [38] := BCD_TO_DWORD (X [56]) ;|
```

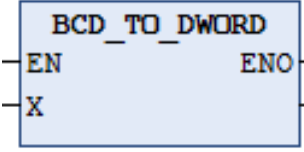
LD:



### 3.14.9 DWORD\_TO\_BCD

Convert DWORD type data of source data to BCD code.

⊙ **Command Format**

Instruction	Name	FB/FC	LD	ST	File
DWORD_TO_BCD	Double word to BCD instruction	FC		Output: = DWORD_TO_BCD (X)	Util

⊙ **Related Variables**

**Input variable**

Input variable	Name	Data type	Range	initialization	Descriptive
X	Input data	DWORD	0–FFFF FFFF	0	Input data

**Output variable.**

Output variable	Name	Data type	Range	initialization	Descriptive
DWORD_TO_BCD	Output data	DWORD	0–FFFF FFFF	0	Output data

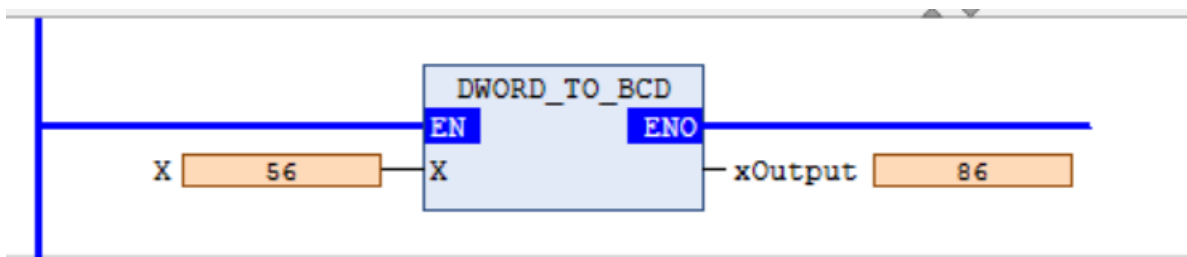
	Boolean	Bit string				Integer							Real number		Moment, Duration, Date, String					
	BOOL	BYTE	WORD	DWORD	LWORD	USINT	UINT	UDINT	ULINT	SINT	INT	DINT	LINT	REAL	LREAL	TIME	DATE	TOD	DT	STRING
X	-	-	-	✓	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
DWORD_TO_BCD	-	-	-	✓	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-

⊙ **Program demo**

ST:

```
xOUTPUT 86 :=DWORD_TO_BCD (X 56) ;
```

LD:



## 3.15 Analog Waveform.

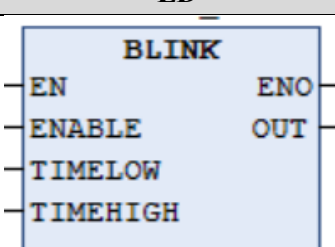
### 3.15.1 Instruction List

Instruction Type	Name	FB/FC	Functionality
Analog Waveform	BLINK	FB	Pulse Signal Generator
	GEN	FB	Periodic Signal Generator
	FREQ_MEASURE	FB	Frequency measurement

### 3.15.2 BLINK

A pulse signal generator that sets the output to TRUE during the time period TIMEHIGH and subsequently sets the output to FALSE during the TIMELOW period.

#### ⊙ Command Format

Instruction	Name	FB/FC	LD	ST	File
BLINK	Pulse Signal Generator instruction	FC		BLINK_0( ENABLE:=, TIMELOW:=, TIMEHIGH:=, OUT=>);	Util

#### ⊙ Related Variables

##### Input variable

Input variable	Name	Data type	Range	initialization	Descriptive
ENABLE	Enable	BOOL	0-255	0	TRUE, the command starts working
TIMELOW	Low Level Time	TIME	-	-	The time the pulse signal remains low
TIMEHIGH	High Level Time	TIME	-	-	The time the pulse signal remains high

##### Output variable.

Output variable	Name	Data type	Range	initialization	Descriptive
OUT	Pulse signal output	BOOL	TRUE-FALSE	FALSE	Pulse signal output

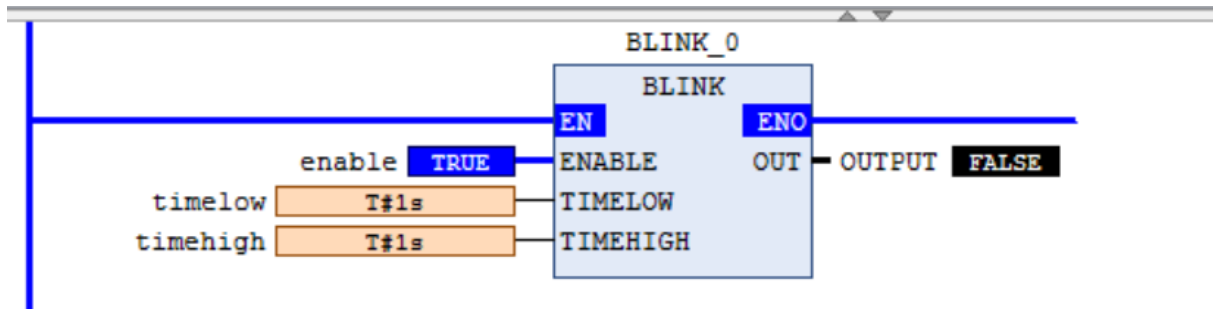
	Boo lea n	Bit string				Integer							Real number		Moment, Duration, Date, String					
	BOOL	BYTE	WORD	DWORD	LWORD	USINT	UINT	UDINT	ULINT	SINT	INT	DINT	LINT	REAL	LREAL	TIME	DATE	TOD	DT	STRING
ENABLE	√	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
TIMELOW	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	√	-	-	-	-
TIMEHIGH	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	√	-	-	-	-
OUT	√	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-

⊙ Program demo

ST:

```
BLINK_0(ENABLE:=enable, TIMELOW:=timelow, TIMEHIGH:=timehigh, OUT=>OUTPUT);
```

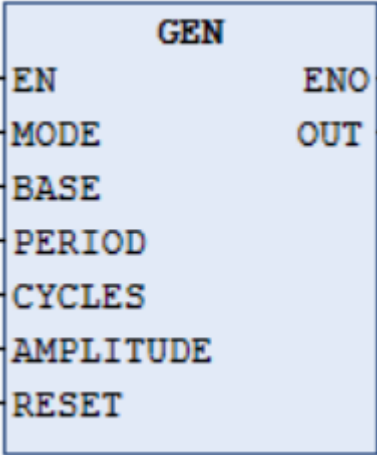
LD:



### 3.15.3 GEN

Generate the specified function waveform according to the input parameters.

#### ⊙ Command Format

Instruction	Name	FB/FC	LD	ST	File
GEN	Periodic Signal Generator instruction	FC		GEN_0( MODE:=, BASE:=, PERIOD:=, CYCLES:=, AMPLITUDE:=, RESET:=, OUT=>);	Util

#### ⊙ Related Variables

##### Input variable

Input variable	Name	Data type	Range	initialization	Descriptive
MODE	Signal Type	GEN_MODE	-	TRIANGLE	Enumerated variables of type GEN_MODE
BASE	Cycle method selection	BOOL	FALSE-TRUE	FALSE	TRUE: output signal with PERIOD as cycle time. FALSE: output signal with CYCLES as the cycle number.
PERIOD	Cycle period	TIME	-	TIME#1s0ms	If BASE is TRUE, the cycle time of the output signal is valid.
CYCLES	Cycle times	INT	0-65535	1000	Valid when BASE is FALSE, the cycle time of the output signal.
AMPLITUDE	Signal Amplitude	INT	0-65535	0	Amplitude value of output signal
RESET	Reset	BOOL	FALSE-TRUE	FALSE	When TRUE, the output signal is reset to 0, when FALSE, the signal can be output normally.

##### Output variable.

Output variable	Name	Data type	Range	initialization	Descriptive
OUT	Signal output value	INT	0-65535	0	Output value of the periodic signal

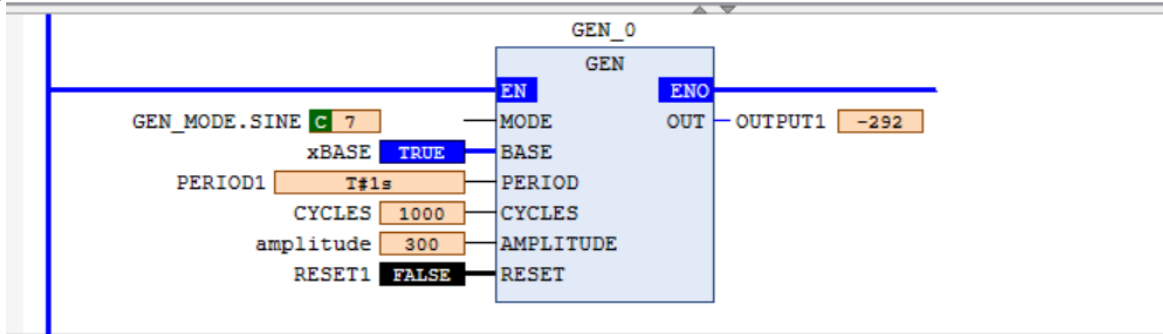
	Bo	Bit string					Integer						Real number		Moment, Duration, Date, String						
	olean	BOOL	BYTE	WORD	DWORD	LWORD	USINT	UINT	UDINT	ULINT	SINT	INT	DINT	LINT	REAL	LREAL	TIME	DATE	TOD	DT	STRING
MODE		GEN_MODE																			
TIMELOW	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	√	-	-	-	-

BASE	√	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
PERIOD	-	-	-	-	-	-	-	-	-	-	-	-	-	-	√	-	-	-	-	-
CYCLES	-	-	-	-	-	-	-	-	-	-	√	-	-	-	-	-	-	-	-	-
AMPLITUDE	-	-	-	-	-	-	-	-	-	-	√	-	-	-	-	-	-	-	-	-
RESET	√	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-

**⊙ Program demo**
**ST:**

```

GEN_0 (
  MODE SINE :=GEN_MODE.SINE,
  BASE FALSE :=BASE FALSE,
  PERIOD T#1s :=PERIOD1 T#1s,
  CYCLES 1000 :=CYCLES 1000,
  AMPLITUDE 300 :=amplitude 300,
  RESET FALSE :=RESET FALSE,
  OUT -146 =>OUTPUT1 -146 );RETURN
  
```

**LD:**

**⊙ Functional Description**

MODE is an enumerated variable of type GEN\_MODE, there are 7 enumerated variables, such as:

0-TRIANGLE: output triangle wave function, amplitude: -Amplitude~+Amplitude.

1-TRIANGLE\_POS: output triangle wave function, amplitude: 0~+Amplitude.

2-SAWTOOTH\_RISE: Rising sawtooth wave function, amplitude: -Amplitude~+Amplitude.

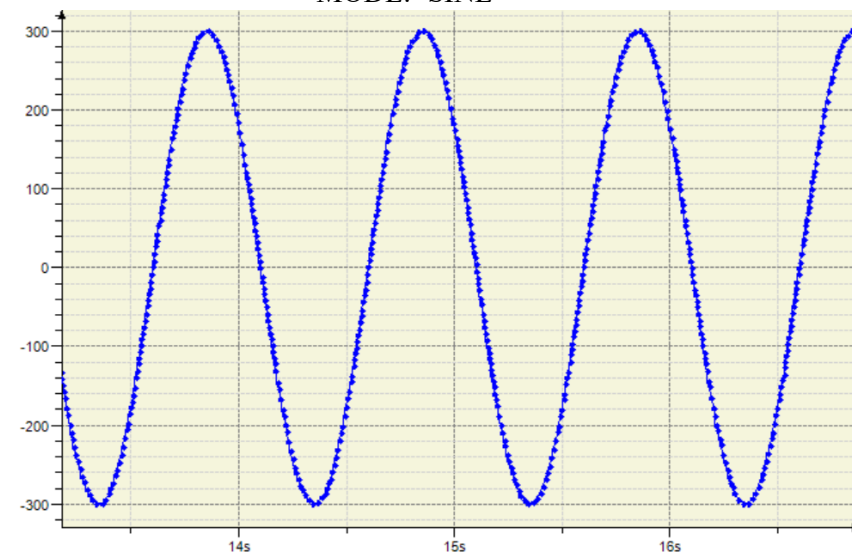
3-SAWTOOTH\_FALL: Falling sawtooth wave function, amplitude: -Amplitude~+Amplitude.

4-RECTANGLE: Rectangle sawtooth wave function conversion, amplitude: -Amplitude~+Amplitude.

5-SINE: Sine function

6-COSINE: cosine function

MODE:=SINE



### 3.15.4 FREQ\_MEASURE

Measurement of the (average) frequency value of the incoming Boolean type signal (Hz).

#### ⊙ Command Format

Instruction	Name	FB/FC	LD	ST	File
FREQ_MEASURE	Frequency measurement instruction	FC	<p style="text-align: center;">FREQ_MEASURE_0</p> <div style="border: 1px solid black; padding: 5px; width: fit-content; margin: auto;"> <p style="text-align: center;"><b>FREQ_MEASURE</b></p> <p>EN                      ENO</p> <p>IN                        OUT</p> <p>PERIODS                VALID</p> <p>RESET</p> </div>	FREQ_MEASURE_0(IN:=, PERIODS:=, RESET:=, OUT=>, VALID=>);	Util

#### ⊙ Related Variables

##### Input variable

Input variable	Name	Data type	Range	initialization	Descriptive
IN	Input Signal	BOOL	TRUE-FALSE	FALSE	TRUE. Command is active.
PERIODS	Measurement cycle	INT	0-65535	1	Range 1 to 10, default 1, the time interval between two rising edges of the input signal is one cycle, the frequency of the input signal is obtained by averaging the frequency values of N cycles, this parameter is the number of cycles of the averaging operation.
RESET	Reset	BOOL	TRUE-FALSE	FALSE	If BASE is TRUE, the cycle time of the output signal is valid.

##### Output variable.

Output variable	Name	Data type	Range	initialization	Descriptive
OUT	Frequency Output	REAL	-2147483648~2147483647	0	Frequency of input signal, unit Hz
VALID	operation flag	BOOL	TRUE-FALSE	FALSE	TRUE when the first operation is completed or when there is an arithmetic error, FALSE for the rest of the time.

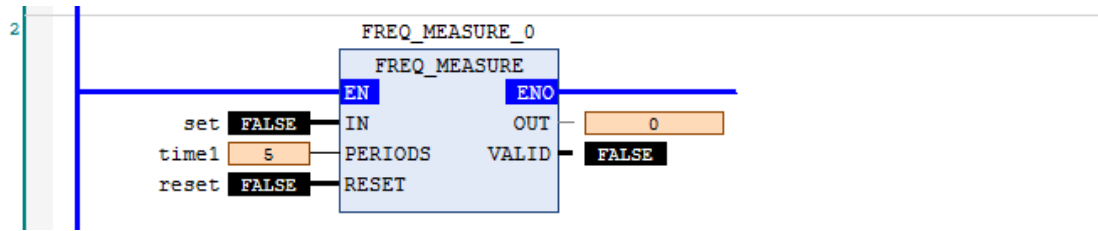
	Bo ole an	Bit string					Integer						Real number		Moment, Duration, Date, String					
	BOOL	BYTE	WORD	DWORD	LWORD	USINT	UINT	UDINT	ULINT	SINT	INT	DINT	LINT	REAL	LREAL	TIME	DATE	TOD	DT	STRING
IN	√	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
PERIODS	-	-	-	-	-	-	-	-	-	√	-	-	-	-	-	-	-	-	-	-
RESET	√	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
OUT	-	-	-	-	-	-	-	-	-	-	-	-	-	√	-	-	-	-	-	-
VALID	√	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-

⊙ Program demo

ST:

```
1 ● FREQ_MEASURE_0(  
2   IN:=set,  
3   PERIODS:=time1,  
4   RESET:=reset,  
5   OUT=> ,  
6   VALID=> );
```

LD:



## 4. Motion instruction

### 4.1 Single-axis status monitoring

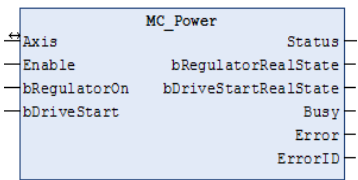
#### 4.1.1 Instruction List

Instruction Type	Name	FB/FC	Functionality
Axis state	MC Power	FB	Axis enable
	MC Reset	FB	Axis reset
	MC ReadStatus	FB	Read Axis Status
	MC ReadAxisError	FB	Read Axis Error
	MC ReadParameter	FB	Read parameters
	MC ReadBoolParameter	FB	Read Boolean parameter
	MC WriteParameter	FB	Write parameters
	MC WriteBoolParameter	FB	Write Boolean parameters
	MC ReadActualPosition	FB	Read actual position
	MC ReadActualVelocity	FB	Read actual speed
	MC ReadActualTorque	FB	Read actual torque
	MC SetPosition	FB	Set position
	SMC_ReadSetPosition	FB	Read set position
	SMC_ReadFBError	FB	Read historical error messages
	SMC_ClearFBError	FC	Clear Historical Error Messages
SMC_ErrorString	FB	Error Code to Error Message	

#### 4.1.2 MC\_Power

Enables the specified axis to enter or exit the runnable state, also called axis enable.

##### ⊙ Command Format

Instruction	Name	FB/FC	LD	ST	File
MC_Power	Axis enable instruction	FB		<pre>MC_Power( Axis:=, Enable:=, bRegulatorOn:=, bDriveStart:=, Status=&gt;, bRegulatorRealState= &gt;, bDriveStartRealState= &gt;, Busy=&gt;, Error=&gt;, ErrorID=&gt;);</pre>	SM3_Basic

##### ⊙ Related Variables

###### I/O variable

I/O variable	Name	Data type	Range	initialization	Descriptive
Axis	Axis	AXIS_REF_SM3	-	-	Specified axis

###### Input variable

Input variable	Name	Data type	Range	initialization	Descriptive
Enable	effective	BOOL	TRUE-FALSE	FALSE	TRUE to activate the processing of the function block.
bRegulatorOn	Enable	BOOL	TRUE-FALSE	FALSE	TRUE, activates the enabling state of the axis.
bDriveStart	Driver Enable	BOOL	TRUE-FALSE	FALSE	TRUE, disables the emergency stop processing of the axis.

**Output variable.**

Output variable	Name	Data type	Range	initialization	Descriptive
Status	Operable	BOOL	TRUE-FALSE	FALSE	TRUE if the axis is ready.
bRegulatorRealState	Enable active	BOOL	TRUE-FALSE	FALSE	Valid state for axis enable.
bDriveStartRealState	Driver available	BOOL	TRUE-FALSE	FALSE	TRUE if the drive has not been interrupted by the Quick Stop mechanism.
Busy	Execution in progress	BOOL	TRUE-FALSE	FALSE	TRUE if processing of the function block is not completed.
Error	Error	BOOL	TRUE-FALSE	FALSE	TRUE if an exception occurs
ErrorID	Error Code	SMC_ER ROR	-	0	The value is 0 when normal, and an error code is output when an exception occurs.

	Boolean	Bit string					Integer							Real number		Moment, Duration, Date, String				
	BOOL	BYTE	WORD	DWORD	LWORD	USINT	UINT	UDINT	ULINT	SINT	INT	DINT	LINT	REAL	LREAL	TIME	DATE	TOD	DT	STRING
Axis	AXIS_REF_SM3																			
Enable	√	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
bRegulatorOn	√	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
bDriveStart	√	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
Status	√	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
bRegulatorRealState	√	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
bDriveStartRealState	√	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
Busy	√	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
Error	√	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
ErrorID	SMC_ERROR																			

**⊙ Functional Description**

When Enable is set to TRUE, the function block enters the runnable state; when Enable is set to FALSE, the function block does not run, but it can execute MC\_Power instruction, MC\_Reset (axis error reset) instruction.

Set bRegulatorOn to TRUE, if the axis has no error, the axis state will be standstill, if the axis has error, the axis state will be errorstop.

Setting bRegulatorOn to FALSE results in the axis state being Disabled, indicating that the axis is not ready for motion.

Set bDriveStart to FALSE and the axis stops sharply.

**⊙ Program demo**

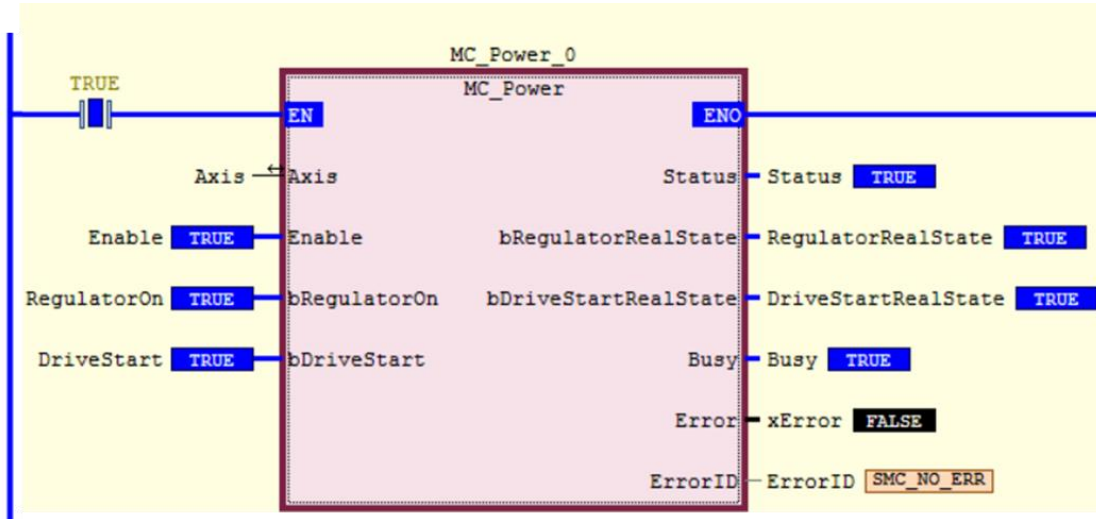
**ST:** When Enable, RegulatorOn, DriveStart is TRUE, Status changes to TRUE, indicating that axis Axis is enabled.

```

1  MC_Power_0(
2      Axis:= Axis,
3      Enable TRUE := Enable TRUE,
4      bRegulatorOn TRUE := RegulatorOn TRUE,
5      bDriveStart TRUE := DriveStart TRUE,
6      Status TRUE => Status TRUE,
7      bRegulatorRealState TRUE => RegulatorRealState TRUE,
8      bDriveStartRealState TRUE => DriveStartRealState TRUE,
9      Busy TRUE => Busy TRUE,
10     Error FALSE => xError FALSE,
11     ErrorID[SMC_NO_ERR] => ErrorID[SMC_NO_ERR]; RETURN

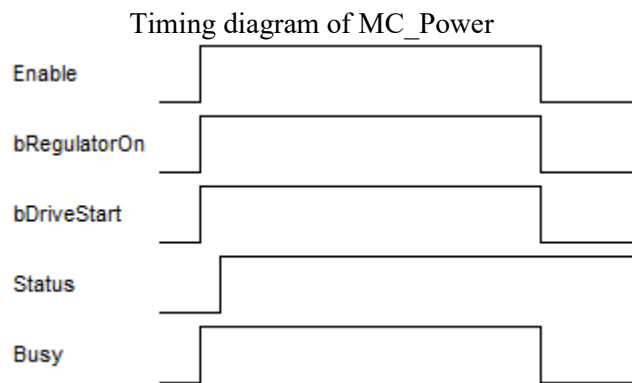
```

**LD:** When Enable, RegulatorOn, DriveStart is TRUE, Status changes to TRUE, indicating that axis Axis is enabled.



⊙ **Timing Diagram:**

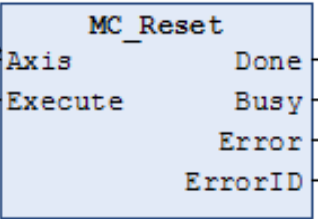
Setting Enable, bRegulatorOn and bDriveStart to TRUE respectively, the Busy status of the function block will become TRUE, and then the Status signal will also become TRUE, and the axis becomes enabled. As shown in the figure.



### 4.1.3 MC\_Reset

Reset (clear) axis error.

#### ⊙ Command Format

Instruction	Name	FB/FC	LD	ST	File
MC_Reset	Axis reset instruction	FB		Axis:=, Execute:=, Done=>, Busy=>, Error=>, ErrorID=>;	SM3_Basic

#### ⊙ Related Variables

##### I/O variable

I/O variable	Name	Data type	Range	initialization	Descriptive
Axis	Axis	AXIS_REF_SM3	-	-	Specified axis

##### Input variable

Input variable	Name	Data type	Range	initialization	Descriptive
Execute	Activate	BOOL	TRUE-FALSE	FALSE	Rising edge activates the function block.

##### Output variable.

Output variable	Name	Data type	Range	initialization	Descriptive
Done	Completion	BOOL	TRUE-FALSE	FALSE	TRUE if the reset is executed.
Busy	Executing	BOOL	TRUE-FALSE	FALSE	TRUE when function block execution has not been completed.
Error	Error	BOOL	TRUE-FALSE	FALSE	Function Block Execution Error
ErrorID	Error Code	SMC_ERROR	-	0	Error indication, see SMC_Error.

	Boolean	Bit string					Integer							Real number		Moment, Duration, Date, String				
	BOOL	BYTE	WORD	DWORD	LWORD	USINT	UINT	UDINT	ULINT	SINT	INT	DINT	LINT	REAL	LREAL	TIME	DATE	TOD	DT	STRING
Axis		AXIS_REF_SM3																		
Execute	√	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
Done	√	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
Busy	√	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
Error	√	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
ErrorID		SMC_ERROR																		

#### ⊙ Functional Description

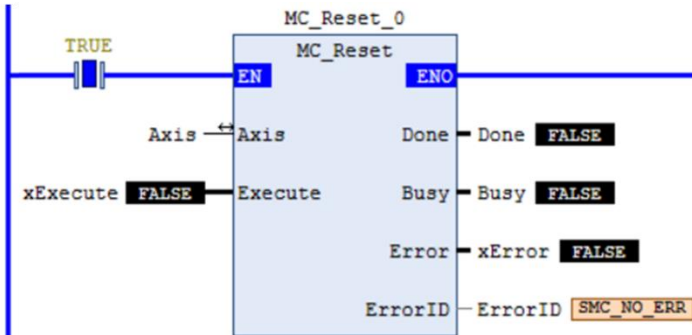
After an error has been reported, the axis cannot continue to execute motion commands and can only be operated after the error has been cleared by calling the MC\_Reset command, which is used to change the state of the axis from ErrorStop to StandStill, eliminating the error reporting state of the axis and turning it into an executable state.

⊙ **Program demo**

**ST:** When xExecute becomes TRUE, Done becomes TRUE, indicating that the Axis error has been cleared!

```
MC_Reset_0(
  Axis:= Axis ,
  Execute FALSE := xExecute FALSE ,
  Done FALSE => Done FALSE ,
  Busy FALSE => Busy FALSE ,
  Error FALSE => xError FALSE ,
  ErrorID SMC_NO_ERR => ErrorID SMC_NO_ERR );
```

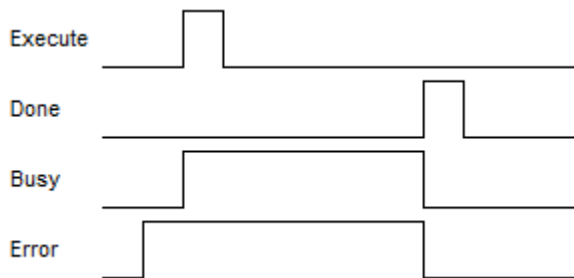
**LD:** When xExecute becomes TRUE, Done becomes TRUE, indicating that the Axis error has been cleared!



⊙ **Timing Diagram**

When Error becomes TRUE and Execute is set to TRUE, the Busy state of the function block will become TRUE, and then the Done signal also becomes TRUE, and this axis clears the error state. As shown in the figure.

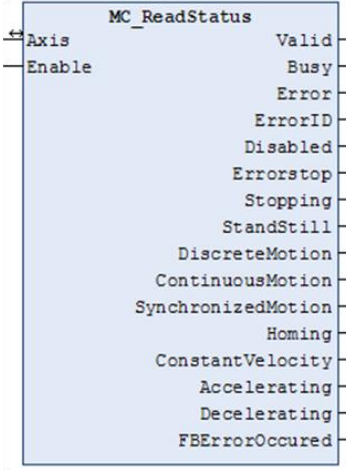
Timing diagram of MC\_Peset



### 4.1.4 MC\_ReadStatus

This command is used to read the detailed status of the axis.

#### ⊙ Command Format

Instruction	Name	FB/FC	LD	ST	File
MC_ReadStatus	Read Axis Status instruction	FB		MC_ReadStatus( Axis: =, Enable: =, Valid=>, Busy=>, Error=>, ErrorID=>, Disabled=>, Errorstop=>, Stopping=>, StandStill=>, DiscreteMotion=>, ContinuousMotion=>, SynchronizedMotion=>, Homing=>, ConstantVelocity=>, Accelerating=>, Decelerating=>, FBErrorOccured=>);	SM3_Basic

#### ⊙ Related Variables

##### I/O variable

I/O variable	Name	Data type	Range	initialization	Descriptive
Axis	Axis	AXIS_REF_SM3	-	-	Specified axis

##### Input variable

Input variable	Name	Data type	Range	initialization	Descriptive
Execute	Effective	BOOL	TRUE-FALSE	FALSE	TRUE to activate the use of function blocks.

##### Output variable.

Output variable	Name	Data type	Range	initialization	Descriptive
Valid	Effective	BOOL	TRUE-FALSE	FALSE	TRUE if the axis is ready.
Busy	In progress	BOOL	TRUE-FALSE	FALSE	TRUE when function block execution is not finished.
Error	Error	BOOL	TRUE-FALSE	FALSE	Function block execution error
ErrorID	Error Code	SMC_ERROR	-	0	Error indication, see SMC_Error.
Disabled	Not enabled	BOOL	TRUE-FALSE	FALSE	TRUE if the axis status is Disabled
Errorstop	Error	BOOL	TRUE-FALSE	FALSE	TRUE if the axis status is Errorstop
Stopping	Stopping	BOOL	TRUE-FALSE	FALSE	TRUE if the axis state is Stopping
StandStill	Ready	BOOL	TRUE-FALSE	FALSE	TRUE if the axis state is Standstill.

DiscreteMotion	Point motion	BOOL	TRUE-FALSE	FALSE	TRUE if the axis state is DiscreteMotion.
ContinuousMotion	Continuous motion	BOOL	TRUE-FALSE	FALSE	TRUE if the axis state is ContinuousMotion.
SynchronizedMotion	Synchronised motion	BOOL	TRUE-FALSE	FALSE	TRUE if the axis state is SynchronisedMotion.
Homing	Zero return in progress	BOOL	TRUE-FALSE	FALSE	TRUE if the axis state is Homing.
ConstantVelocity	Constant motion	BOOL	TRUE-FALSE	FALSE	TRUE if the motor is moving at a constant speed.
Accelerating	Accelerating	BOOL	TRUE-FALSE	FALSE	TRUE if the motor is accelerating
Decelerating	Decelerating	BOOL	TRUE-FALSE	FALSE	TRUE if the motor is decelerating
FBErrorOccured	FB error	BOOL	TRUE-FALSE	FALSE	TRUE if a function block error has been detected and not yet cleared by SMC_ClearFBError

	Boolean	Bit string				Integer							Real number		Moment, Duration, Date, String					
	BOOL	BYTE	WORD	DWORD	LWORD	USINT	UINT	UDINT	ULINT	SINT	INT	DINT	LINT	REAL	LREAL	TIME	DATE	TOD	DT	STRING
Axis	AXIS_REF_SM3																			
Enable	√	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
Valid	√	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
Busy	√	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
Error	√	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
ErrorID	SMC_ERROR																			
Disabled	√	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
Errorstop	√	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
Stopping	√	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
StandStill	√	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
DiscreteMotion	√	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
ContinuousMotion	√	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
SynchronizedMotion	√	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
Homing	√	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
ConstantVelocity	√	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
Accelerating	√	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
Decelerating	√	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
FBErrorOccured	√	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-

**⊙ Functional Description**

It is used to read the state of the axis and refer the state to the corresponding program step. when Enable is false, all state outputs will be set to false; when Enable is true, the state of the axis will be read continuously.

**⊙ Program demo**

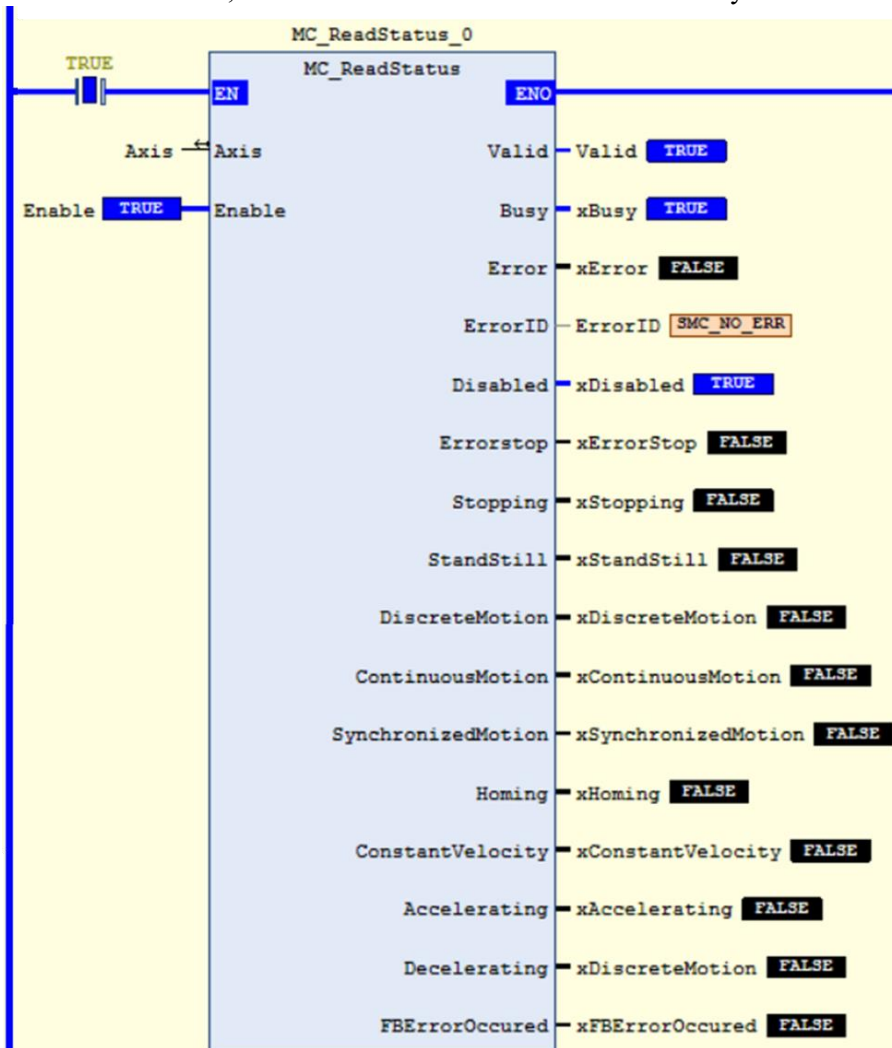
**ST:** When Enable is TRUE, the current state of the axis is continuously read.

```

● MC_ReadStatus_0(
    Axis:=Axis ,
    Enable TRUE :=Enable TRUE ,
    Valid TRUE =>Valid TRUE ,
    Busy TRUE =>xBusy TRUE ,
    Error FALSE =>xError FALSE ,
    ErrorID SMC_NO_ERR =>ErrorID SMC_NO_ERR ,
    Disabled TRUE =>xDisabled TRUE ,
    Errorstop FALSE => xErrorStop FALSE ,
    Stopping FALSE => xStopping FALSE ,
    StandStill FALSE => xStandStill FALSE ,
    DiscreteMotion FALSE => xDiscreteMotion FALSE ,
    ContinuousMotion FALSE =>xContinuousMotion FALSE ,
    SynchronizedMotion FALSE =>xSynchronizedMotion FALSE ,
    Homing FALSE =>xHoming FALSE ,
    ConstantVelocity FALSE =>xConstantVelocity FALSE ,
    Accelerating FALSE =>xAccelerating FALSE ,
    Decelerating FALSE =>xDecelerating FALSE ,
    FBErrorOccured FALSE =>xFBErrorOccured FALSE );RETURN

```

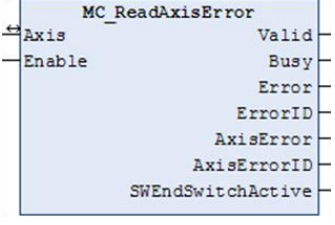
**LD:** When Enable is TRUE, the current state of the axis is continuously read.



### 4.1.5 MC\_ReadAxisError

Errors in reading axes.

#### ⊙ Command Format

Instruction	Name	FB/FC	LD	ST	File
MC_ReadAxisError	Read Axis Error instruction	FB		MC_ReadAxisError( Axis: =, Enable: =, Valid=>, Busy=>, Error=>, ErrorID=>, AxisError=>, AxisErrorID=>, SWEndSwitchActive=>);	SM3_Basic

#### ⊙ Related Variables

##### I/O variable

I/O variable	Name	Data type	Range	initialization	Descriptive
Axis	Axis	AXIS_REF_SM3	-	-	Specified axis

##### Input variable

Input variable	Name	Data type	Range	initialization	Descriptive
Execute	Effective	BOOL	TRUE-FALSE	FALSE	TRUE to activate the use of function blocks.

##### Output variable.

Output variable	Name	Data type	Range	initialization	Descriptive
Valid	Effective	BOOL	TRUE-FALSE	FALSE	TRUE if the axis is ready.
Busy	In progress	BOOL	TRUE-FALSE	FALSE	TRUE when function block execution is not finished.
Error	Error	BOOL	TRUE-FALSE	FALSE	Function block execution error
ErrorID	Error code	SMC_ERROR	-	0	Error indication, see SMC_Error.
AxisError	Axis error	BOOL	TRUE-FALSE	FALSE	Axis error flag
AxisErrorID	Axis error value	DWORD	INT	0	Axis error value
SWEndSwitchActive	Soft limit	BOOL	TRUE-FALSE	FALSE	TRUE if the soft limit is exceeded.

	Boolean	Bit string				Integer							Real number		Moment, Duration, Date, String					
	BOOL	BYTE	WORD	DWORD	LWORD	USINT	UINT	UDINT	ULINT	SINT	INT	DINT	LINT	REAL	LREAL	TIME	DATE	TOD	DT	STRING
Axis		AXIS_REF_SM3																		
Enable	√	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
Valid	√	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
Busy	√	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-

Error	√	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
ErrorID	SMC_ERROR																		
AxisError	√	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
AxisErrorID	√	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
SWEndSwitchActive	√	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-

⊙ **Functional Description**

Used to read the error of the axis and refer the error status to the corresponding program step. enable is false, all outputs will be set to false; enable is true, the error of the axis will be read continuously.

⊙ **Program demo**

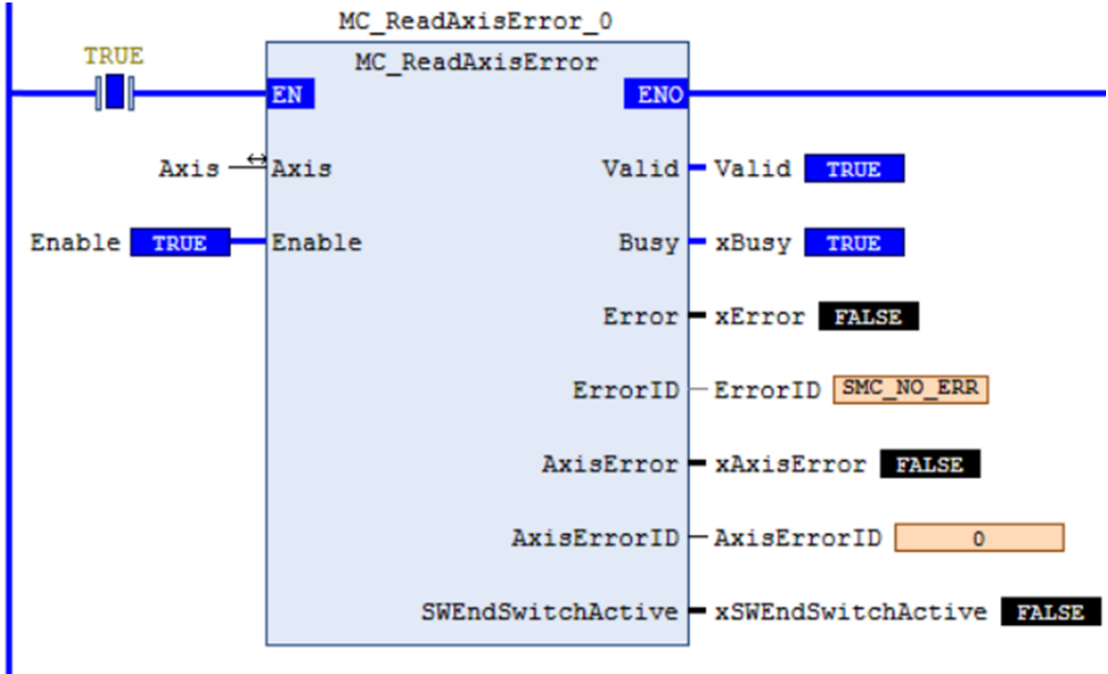
**ST:** Continuous Read Axis Axis Error when Enable is TRUE.

```

● MC_ReadAxisError_0(
    Axis:= Axis ,
    Enable TRUE := Enable TRUE ,
    Valid TRUE => Valid TRUE ,
    Busy TRUE => xBusy TRUE ,
    Error FALSE => xError FALSE ,
    ErrorID SMC_NO_ERR => ErrorID SMC_NO_ERR ,
    AxisError FALSE => xAxisError FALSE ,
    AxisErrorID 0 => AxisErrorID 0 ,
● SWEndSwitchActive FALSE => xSWEndSwitchActive FALSE );RETURN

```

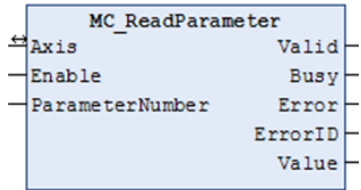
**LD:** Continuous Read Axis Axis Error when Enable is TRUE.



### 4.1.6 MC\_ReadParameter

Retrieve the value of the specified parameter.

#### ⊙ Command Format

Instruction	Name	FB/FC	LD	ST	File
MC_ReadParameter	Read parameters instruction	FB		MC_ReadParameter( Axis: =, Enable: =, ParameterNumber: =, Valid=>, Busy=>, Error=>, ErrorID=>, Value=>);	SM3_Basic

#### ⊙ Related Variables

##### I/O variable

I/O variable	Name	Data type	Range	initialization	Descriptive
Axis	Axis	AXIS_REF_SM3	-	-	Specified axis

##### Input variable

Input variable	Name	Data type	Range	initialization	Descriptive
Enable	Start	BOOL	TRUE-FALSE	FALSE	TRUE to activate the use of function blocks.
ParameterNumber	Axis parameter number	DINT	ALL	0	Index and sub-index and serial number of the access axis parameter

##### Output variable.

Output variable	Name	Data type	Range	initialization	Descriptive
Valid	Getting Flags	BOOL	TRUE-FALSE	FALSE	TRUE if the parameter has been read.
Busy	Execution in progress	BOOL	TRUE-FALSE	FALSE	TRUE when function block execution has not finished.
Error	Error	BOOL	TRUE-FALSE	FALSE	Function Block Execution Error
ErrorID	Error Code	SMC_ERROR	-	0	Error indication, see SMC_Error.
Value	Parameter value	LREAL	ALL	0	Read the value of the parameter

	Boolean	Bit string					Integer						Real number		Moment, Duration, Date, String					
	BOOL	BYTE	WORD	DWORD	LWORD	USINT	UINT	UDINT	ULINT	SINT	INT	DINT	LINT	REAL	LREAL	TIME	DATE	TOD	DT	STRING
Axis		AXIS_REF_SM3																		
Enable	√	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
ParameterNumber	-	-	-	-	-	-	-	-	-	-	√	-	-	-	-	-	-	-	-	-
Valid	√	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
Busy	√	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-

Error	√	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	
ErrorID	SMC_ERROR																			
Value	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	√	-	-	-	-

**⊙ Functional Description**

In the AXIS\_REF\_SM3 section of the SM3\_Basic library, find the parameter usage represented by the parameter number specific to the ParameterNumber parameter.

Enable is false, all state outputs will be set to false; Enable is true, the parameters of the axis are continuously read.

**⊙ Program demo**

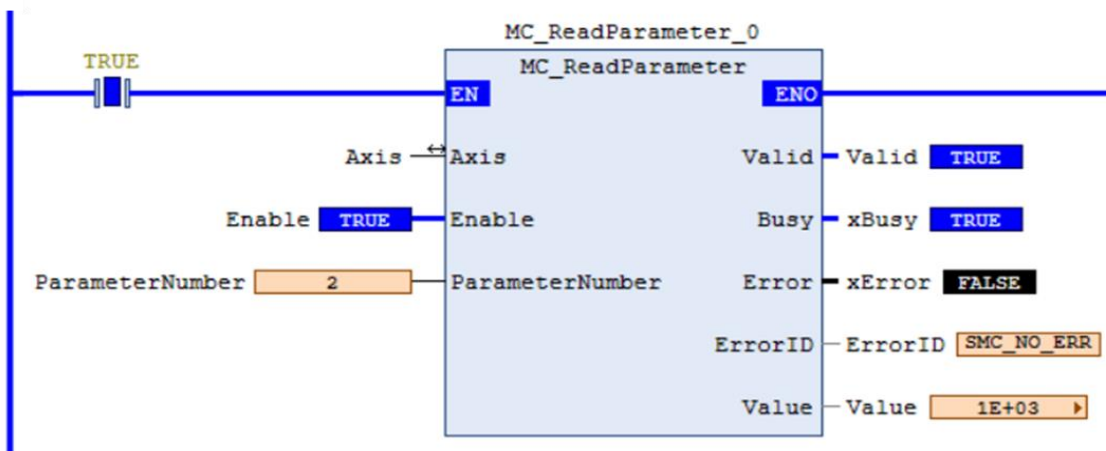
**ST:** When Enable is TRUE, the parameters of Axis Axis are read continuously.

```

● MC_ReadParameter_0 (
  Axis:= Axis,
  Enable TRUE := Enable TRUE,
  ParameterNumber 2 := ParameterNumber 2,
  Valid TRUE => Valid TRUE,
  Busy TRUE => xBusy TRUE,
  Error FALSE => xError FALSE,
  ErrorID SMC_NO_ERR => ErrorID SMC_NO_ERR,
  Value 1E+03 => Value 1E+03 ); RETURN
●

```

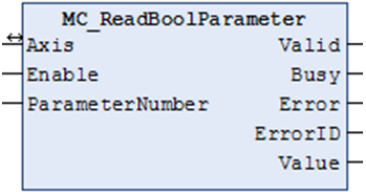
**LD:** When Enable is TRUE, the parameters of Axis Axis are read continuously.



### 4.1.7 MC\_ReadBoolParameter

Reads the value of the specified BOOL variable.

#### ⊙ Command Format

Instruction	Name	FB/FC	LD	ST	File
MC_ReadBoolParameter	Read Boolean parameter instruction	FB		MC_ReadBoolParameter( Axis: =, Enable: =, ParameterNumber: =, Valid=>, Busy=>, Error=>, ErrorID=>, Value=> );ErrorID=>, Value=> );	SM3_Basic

#### ⊙ Related Variables

##### I/O variable

I/O variable	Name	Data type	Range	initialization	Descriptive
Axis	Axis	AXIS_REF_SM3	-	-	Specified axis

##### Input variable

Input variable	Name	Data type	Range	initialization	Descriptive
Enable	Start	BOOL	TRUE-FALSE	FALSE	TRUE to activate the use of function blocks.
ParameterNumber	Axis parameter number	DINT	INT	0	Index and sub-index and serial number of the access axis parameter

##### Output variable.

Output variable	Name	Data type	Range	initialization	Descriptive
Valid	Getting Flags	BOOL	TRUE-FALSE	FALSE	TRUE if the parameter has been read.
Busy	Execution in progress	BOOL	TRUE-FALSE	FALSE	TRUE when function block execution has not finished.
Error	Error	BOOL	TRUE-FALSE	FALSE	Function Block Execution Error
ErrorID	Error Code	SMC_ERROR	-	0	Error indication, see SMC_Error.
Value	Parameter value	LREAL	TRUE-FALSE	FALSE	Read the value of the parameter

	Boolean	Bit string					Integer							Real number		Moment, Duration, Date, String				
	BOOL	BYTE	WORD	DWORD	LWORD	USINT	UINT	UDINT	ULINT	SINT	INT	DINT	LINT	REAL	LREAL	TIME	DATE	TOD	DT	STRING
Axis																				
Enable	√	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
ParameterNumber	-	-	-	-	-	-	-	-	-	-	√	-	-	-	-	-	-	-	-	-
Valid	√	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
Busy	√	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-

Error	√	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
ErrorID	SMC_ERROR																		
Value	√	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-

**⊙ Functional Description**

In the AXIS\_REF\_SM3 section of the SM3\_Basic library, find the parameter usage represented by the parameter number specific to the ParameterNumber parameter.

Enable is false, all status outputs will be set to false; Enable is true, the parameters of the axis are read continuously.

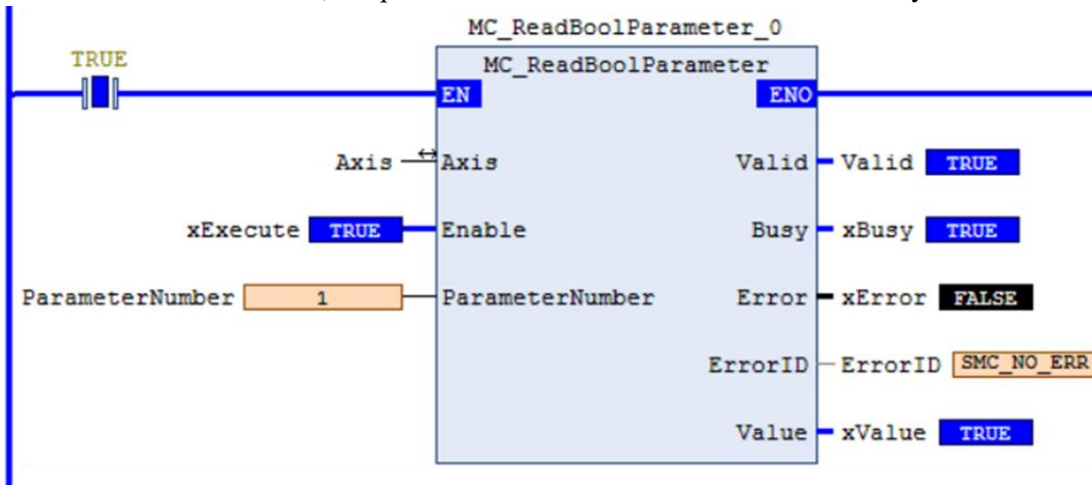
**⊙ Program demo**

**ST:** When Enable is TRUE, the parameters of Axis Axis are read continuously.

```

MC_ReadBoolParameter_0(
  Axis:= Axis,
  Enable TRUE := Enable TRUE ,
  ParameterNumber 1 := ParameterNumber 1 ,
  Valid TRUE => Valid TRUE ,
  Busy TRUE => Busy TRUE ,
  Error FALSE => xError FALSE ,
  ErrorID SMC_NO_ERR => ErrorID SMC_NO_ERR ,
  Value TRUE => xValue TRUE );RETURN
  
```

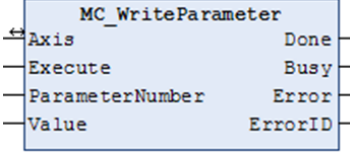
**LD:** When Enable is TRUE, the parameters of Axis Axis are read continuously.



### 4.1.8 MC\_WriteParameter

Reads the value of the specified BOOL variable.

#### ⊙ Command Format

Instruction	Name	FB/FC	LD	ST	File
MC_WriteParameter	Write parameter's instruction	FB		MC_WriteParameter( Axis: =, Execute: =, ParameterNumber: =, Value: =, Done=>, Busy=>, Error=>, ErrorID=> );	SM3_Basic

#### ⊙ Related Variables

##### I/O variable

I/O variable	Name	Data type	Range	initialization	Descriptive
Axis	Axis	AXIS_REF_SM3	-	-	Specified axis

##### Input variable

Input variable	Name	Data type	Range	initialization	Descriptive
Enable	Start	BOOL	TRUE-FALSE	FALSE	TRUE to activate the use of function blocks.
ParameterNumber	Axis parameter number	DINT	INT	0	Index and sub-index and serial number of the access axis parameter
Value	Parameter value	LREAL	ALL	0	Value to be written

##### Output variable.

Output variable	Name	Data type	Range	initialization	Descriptive
Valid	Getting Flags	BOOL	TRUE-FALSE	FALSE	TRUE if the parameter has been read.
Busy	Execution in progress	BOOL	TRUE-FALSE	FALSE	TRUE when function block execution has not finished.
Error	Error	BOOL	TRUE-FALSE	FALSE	Function Block Execution Error
ErrorID	Error Code	SMC_ERROR	-	0	Error indication, see SMC_Error.

	Boolean	Bit string					Integer							Real number		Moment, Duration, Date, String				
	BOOL	BYTE	WORD	DWORD	LWORD	USINT	UINT	UDINT	ULINT	SINT	INT	DINT	LINT	REAL	LREAL	TIME	DATE	TOD	DT	STRING
Axis		AXIS_REF_SM3																		
Enable	√	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
Value	-	-	-	-	-	-	-	-	-	-	-	-	-	√	-	-	-	-	-	-
ParameterNumber	-	-	-	-	-	-	-	-	-	-	√	-	-	-	-	-	-	-	-	-
Valid	√	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
Busy	√	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-

Error	√	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
ErrorID	SMC_ERROR																		

⊙ **Functional Description**

In the AXIS\_REF\_SM3 section of the SM3\_Basic library, find the parameter usage represented by the parameter number specific to the ParameterNumber parameter.  
 Enable is true to write the specified axis parameters.

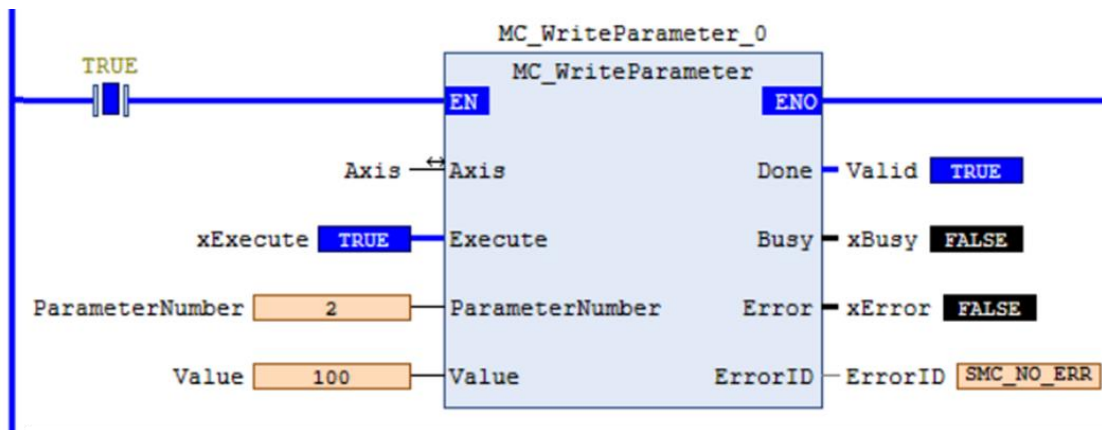
⊙ **Program demo**

**ST:** When xExecute becomes TRUE, write the specified axis parameters.

```

MC_WriteParameter_0(
  Axis:= Axis,
  Execute TRUE := xExecute TRUE,
  ParameterNumber 2 := ParameterNumber 2,
  Value 100 := Value 100,
  Done TRUE => xDone TRUE,
  Busy FALSE => xBusy FALSE,
  Error FALSE => xError FALSE,
  ErrorID SMC_NO_ERR => ErrorID SMC_NO_ERR ); RETURN
  
```

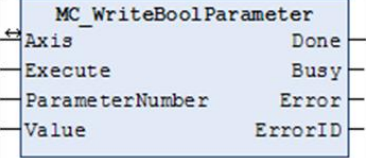
**LD:** When xExecute becomes TRUE, write the specified axis parameters.



### 4.1.9 MC\_WriteBoolParameter

Write parameter values of type BOOL.

#### ⊙ Command Format

Instruction	Name	FB/FC	LD	ST	File
MC_WriteBoolParameter	Write Boolean parameter's instruction	FB		MC_WriteBoolParameter(Axis: =, Execute: =, ParameterNumber: =, Value: =, Done=>, Busy=>, Error=>, ErrorID=> );	SM3_Basic

#### ⊙ Related Variables

##### I/O variable

I/O variable	Name	Data type	Range	initialization	Descriptive
Axis	Axis	AXIS_REF_SM3	-	-	Specified axis

##### Input variable

Input variable	Name	Data type	Range	initialization	Descriptive
Enable	Start	BOOL	TRUE-FALSE	FALSE	TRUE to activate the use of function blocks.
ParameterNumber	Axis parameter number	DINT	INT	0	Index and sub-index and serial number of the access axis parameter
Value	Parameter value	LREAL	ALL	0	Value to be written

##### Output variable.

Output variable	Name	Data type	Range	initialization	Descriptive
Valid	Getting Flags	BOOL	TRUE-FALSE	FALSE	TRUE if the parameter has been read.
Busy	Execution in progress	BOOL	TRUE-FALSE	FALSE	TRUE when function block execution has not finished.
Error	Error	BOOL	TRUE-FALSE	FALSE	Function Block Execution Error
ErrorID	Error Code	SMC_ERROR	-	0	Error indication, see SMC_Error.

	Boolean	Bit string					Integer							Real number		Moment, Duration, Date, String				
	BOOL	BYTE	WORD	DWORD	LWORD	USINT	UINT	UDINT	ULINT	SINT	INT	DINT	LINT	REAL	LREAL	TIME	DATE	TOD	DT	STRING
Axis		AXIS_REF_SM3																		
Enable	√	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
Value	-	-	-	-	-	-	-	-	-	-	-	-	-	√	-	-	-	-	-	-
ParameterNumber	-	-	-	-	-	-	-	-	-	-	-	√	-	-	-	-	-	-	-	-
Valid	√	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
Busy	√	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-

Error	√	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
ErrorID	SMC_ERROR																	

⊙ **Functional Description**

In the AXIS\_REF\_SM3 section of the SM3\_Basic library, find the parameter usage represented by the parameter number specific to the ParameterNumber parameter.  
 Enable is true to write the specified axis parameters.

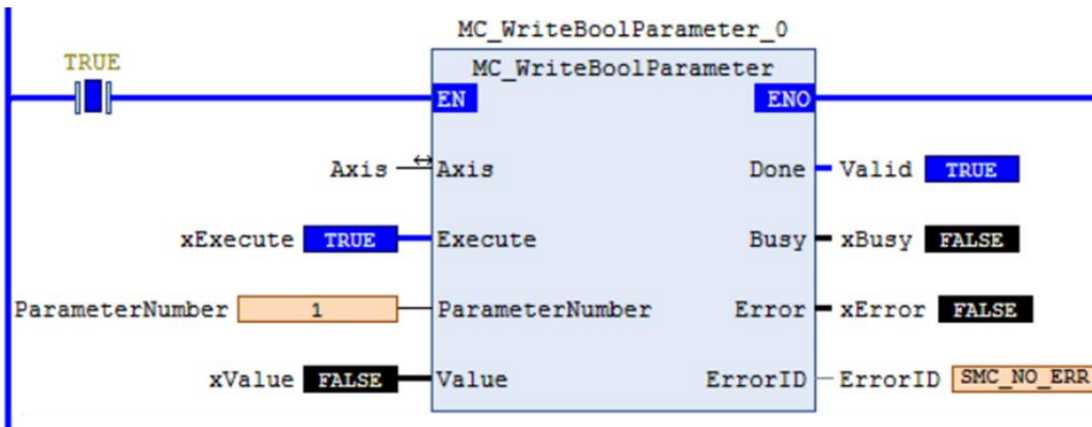
⊙ **Program demo**

**ST:** When xExecute becomes TRUE, write the specified axis parameters.

```

MC_WriteBoolParameter_0(
  Axis:= Axis,
  Execute TRUE := xExecute TRUE,
  ParameterNumber 1 := ParameterNumber 1,
  Value FALSE := xValue FALSE,
  Done TRUE => xDone TRUE,
  Busy FALSE => xBusy FALSE,
  Error FALSE => xError FALSE,
  ErrorID SMC_NO_ERR => ErrorID SMC_NO_ERR ); RETURN
  
```

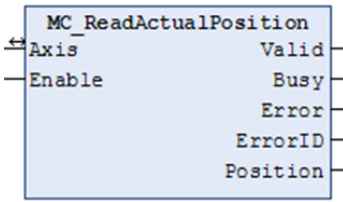
**LD:** When xExecute becomes TRUE, write the specified axis parameters.



### 4.1.10 MC\_ReadActualPosition

Write parameter values of type BOOL.

#### ⊙ Command Format

Instruction	Name	FB/FC	LD	ST	File
MC_ReadActualPosition	Read actual position instruction	FB		MC_ReadActualPosition( Axis: =, Enable: =, Valid=>, Busy=>, Error=>, ErrorID=>, Position=>);	SM3_Basic

#### ⊙ Related Variables

##### I/O variable

I/O variable	Name	Data type	Range	initialization	Descriptive
Axis	Axis	AXIS_REF_SM3	-	-	Specified axis

##### Input variable

Input variable	Name	Data type	Range	initialization	Descriptive
Enable	Start	BOOL	TRUE-FALSE	FALSE	TRUE to activate the use of function blocks.

##### Output variable.

Output variable	Name	Data type	Range	initialization	Descriptive
Valid	Getting Flags	BOOL	TRUE-FALSE	FALSE	TRUE if the parameter has been read.
Busy	Execution in progress	BOOL	TRUE-FALSE	FALSE	TRUE when function block execution has not finished.
Error	Error	BOOL	TRUE-FALSE	FALSE	Function Block Execution Error
ErrorID	Error Code	SMC ERROR	-	0	Error indication, see SMC Error.
Position	Position	LREAL	All	0	New absolute position (expressed in units of axes [u])

	Boolean	Bit string					Integer							Real number		Moment, Duration, Date, String				
	BOOL	BYTE	WORD	DWORD	LWORD	USINT	UINT	UDINT	ULINT	SINT	INT	DINT	LINT	REAL	LREAL	TIME	DATE	TOD	DT	STRING
Axis		AXIS_REF_SM3																		
Enable	√	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
Valid	√	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
Busy	√	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
Error	√	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
ErrorID		SMC_ERROR																		
Position	-	-	-	-	-	-	-	-	-	-	-	-	-	-	√	-	-	-	-	-

#### ⊙ Functional Description

Reads the actual position of the axis in the drive, i.e. fActPosition.

Enable is false, all status outputs will be set to false; Enable is true, the actual position of the axis is continuously read.

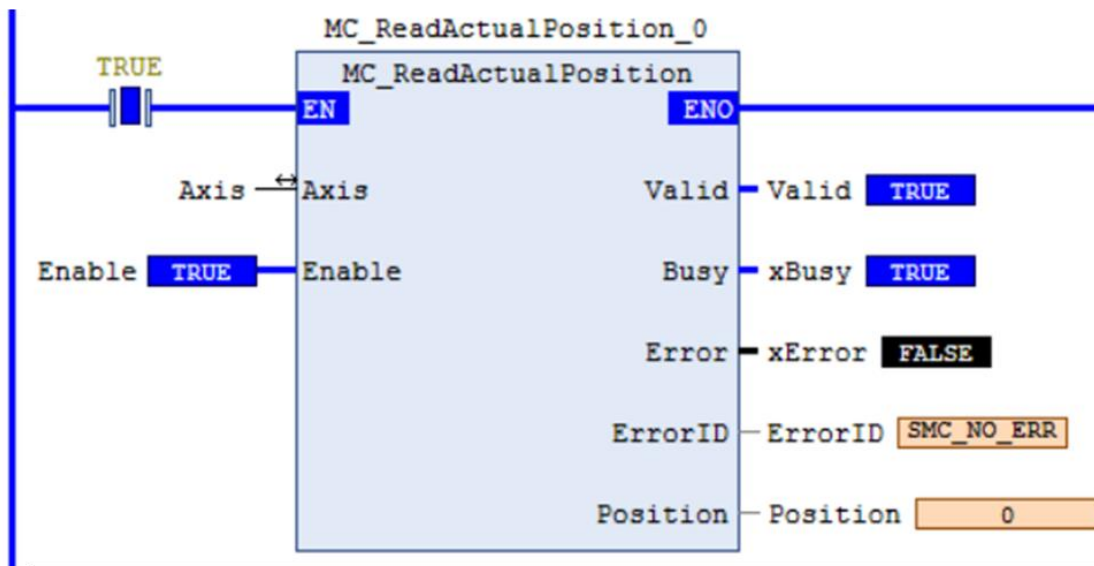
⊙ **Program demo**

**ST:** When Enable is TRUE, the actual position of the axis is continuously read.

```

MC_ReadActualPosition_0(
    Axis:= Axis,
    Enable TRUE := Enable TRUE,
    Valid TRUE => Valid TRUE,
    Busy TRUE => xBusy TRUE,
    Error FALSE => xError FALSE,
    ErrorID SMC_NO_ERR => ErrorID SMC_NO_ERR,
    Position 0 => Position 0 );RETURN
    
```

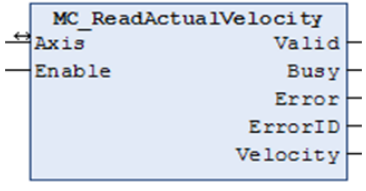
**LD:** When Enable is TRUE, the actual position of the axis is continuously read.



### 4.1.11 MC\_ReadActualVelocity

Read the actual speed value of the axis.

#### ⊙ Command Format

Instruction	Name	FB/FC	LD	ST	File
MC_ReadActualVelocity	Read actual speed instruction	FB		MC_ReadActualVelocity( Axis: =, Enable: =, Valid=>, Busy=>, Error=>, ErrorID=>, Velocity=>);	SM3_Basic

#### ⊙ Related Variables

##### I/O variable

I/O variable	Name	Data type	Range	initialization	Descriptive
Axis	Axis	AXIS_REF_SM3	-	-	Specified axis

##### Input variable

Input variable	Name	Data type	Range	initialization	Descriptive
Enable	Start	BOOL	TRUE-FALSE	FALSE	TRUE to activate the use of function blocks.

##### Output variable.

Output variable	Name	Data type	Range	initialization	Descriptive
Valid	Getting Flags	BOOL	TRUE-FALSE	FALSE	TRUE if the parameter has been read.
Busy	Execution in progress	BOOL	TRUE-FALSE	FALSE	TRUE when function block execution has not finished.
Error	Error	BOOL	TRUE-FALSE	FALSE	Function Block Execution Error
ErrorID	Error Code	SMC ERROR	-	0	Error indication, see SMC Error.
Velocity	Speed	LREAL	ALL	0	The value of the actual speed (expressed in [counting units / second]).

	Boolean	Bit string					Integer							Real number		Moment, Duration, Date, String				
	BOOL	BYTE	WORD	DWORD	LWORD	USINT	UINT	UDINT	ULINT	SINT	INT	DINT	LINT	REAL	LREAL	TIME	DATE	TOD	DT	STRING
Axis		AXIS_REF_SM3																		
Enable	√	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
Valid	√	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
Busy	√	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
Error	√	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
ErrorID		SMC_ERROR																		
Velocity	-	-	-	-	-	-	-	-	-	-	-	-	-	-	√	-	-	-	-	-

#### ⊙ Functional Description

Reads the actual velocity of the axes in the drive, i.e. fActVelocity.

Enable is false, all status outputs will be set to false; Enable is true, the current speed of the axis is continuously read.

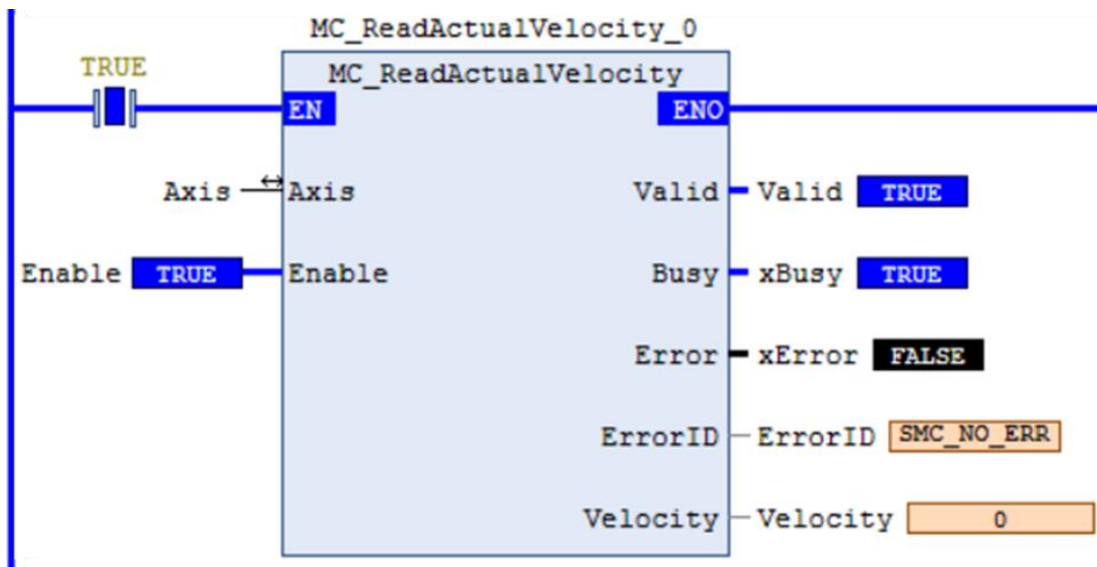
⊙ **Program demo**

**ST:** When Enable is TRUE, the current speed of the axis is continuously read.

```

MC_ReadActualVelocity_0 (
    Axis:= Axis,
    Enable TRUE := Enable TRUE ,
    Valid TRUE => Valid TRUE ,
    Busy TRUE => xBusy TRUE ,
    Error FALSE => xError FALSE ,
    ErrorID SMC_NO_ERR => ErrorID SMC_NO_ERR ,
    Velocity 0 => Velocity 0 ) ;RETURN
    
```

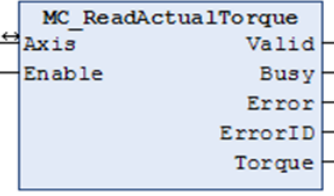
**LD:** When Enable is TRUE, the current speed of the axis is continuously read.



### 4.1.12 MC\_ReadActualTorque

Read the actual torque value of the shaft.

#### ⊙ Command Format

Instruction	Name	FB/FC	LD	ST	File
MC_ReadActualTorque	Read actual torque instruction	FB		MC_ReadActualTorque( Axis:=, Enable:=, Valid=>, Busy=>, Error=>, ErrorID=>, Torque=>);	SM3_Basic

#### ⊙ Related Variables

##### I/O variable

I/O variable	Name	Data type	Range	initialization	Descriptive
Axis	Axis	AXIS_REF_SM3	-	-	Specified axis

##### Input variable

Input variable	Name	Data type	Range	initialization	Descriptive
Enable	Start	BOOL	TRUE-FALSE	FALSE	TRUE to activate the use of function blocks.

##### Output variable.

Output variable	Name	Data type	Range	initialization	Descriptive
Valid	Getting Flags	BOOL	TRUE-FALSE	FALSE	TRUE if the parameter has been read.
Busy	Execution in progress	BOOL	TRUE-FALSE	FALSE	TRUE when function block execution has not finished.
Error	Error	BOOL	TRUE-FALSE	FALSE	Function Block Execution Error
ErrorID	Error Code	SMC ERROR	-	0	Error indication, see SMC Error.
Torque	Torques	LREAL	ALL	0	Value of actual torque or moment (expressed in counting units)

	Boolean	Bit string					Integer							Real number		Moment, Duration, Date, String				
	BOOL	BYTE	WORD	DWORD	LWORD	USINT	UINT	UDINT	ULINT	SINT	INT	DINT	LINT	REAL	LREAL	TIME	DATE	TOD	DT	STRING
Axis		AXIS_REF_SM3																		
Enable	√	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
Valid	√	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
Busy	√	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
Error	√	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
ErrorID		SMC_ERROR																		
Torque	-	-	-	-	-	-	-	-	-	-	-	-	-	-	√	-	-	-	-	-

#### ⊙ Functional Description

Read the actual torque of the centre shaft of the drive.

If Enable is false, all status outputs will be set to false; if Enable is true, the current torque of the axis is continuously read.

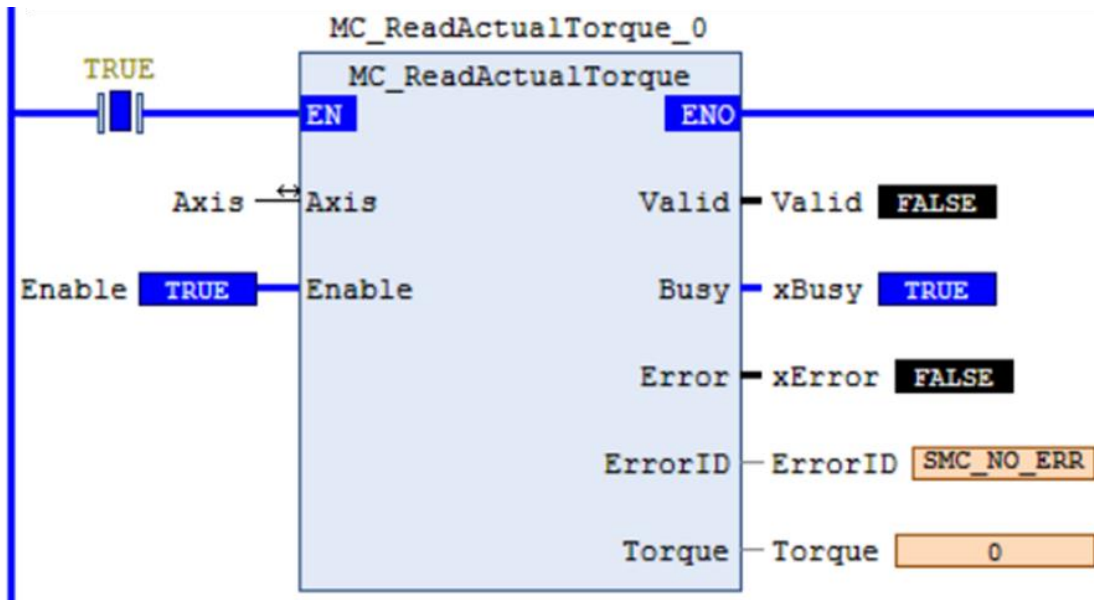
⊙ **Program demo**

**ST:** When Enable is TRUE, the current torque of the axis is continuously read.

```

MC_ReadActualTorque_0 (
  Axis:= Axis,
  Enable TRUE := Enable TRUE ,
  Valid FALSE => Valid FALSE ,
  Busy TRUE => xBusy TRUE ,
  Error FALSE => xError FALSE ,
  ErrorID SMC_NO_ERR => ErrorID SMC_NO_ERR ,
  Torque 0 => Torque 0 ) ;RETURN
  
```

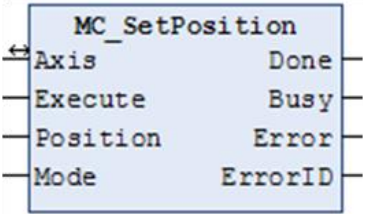
**LD:** When Enable is TRUE, the current torque of the axis is continuously read.



### 4.1.13 MC\_SetPosition

Sets the current position of the axis (does not cause the axis to move).

#### ⊙ Command Format

Instruction	Name	FB/FC	LD	ST	File
MC_SetPosition	Set position instruction	FB		MC_SetPosition( Axis:=, Execute:=, Position, Mode:=, Done=>, Busy=>, Error=>, ErrorID=>);	SM3_Basic

#### ⊙ Related Variables

##### I/O variable

I/O variable	Name	Data type	Range	initialization	Descriptive
Axis	Axis	AXIS_REF_SM3	-	-	Specified axis

##### Input variable

Input variable	Name	Data type	Range	initialization	Descriptive
Enable	Start	BOOL	TRUE-FALSE	FALSE	TRUE to activate the use of function blocks.
Position	Position	LREAL	ALL	0	Position unit [u] (indicates "distance" if Mode = RELATIVE)
Mode	mode	BOOL	TRUE-FALSE	FALSE	TRUE = relative, FALSE = absolute

##### Output variable.

Output variable	Name	Data type	Range	initialization	Descriptive
Done	Done	BOOL	TRUE-FALSE	FALSE	Function block execution completion is TRUE.
Busy	Execution in progress	BOOL	TRUE-FALSE	FALSE	TRUE when function block execution has not finished.
Error	Error	BOOL	TRUE-FALSE	FALSE	Function Block Execution Error
ErrorID	Error Code	SMC_ERROR	-	0	Error indication, see SMC_Error.

	Boolean	Bit string				Integer							Real number		Moment, Duration, Date, String					
	BOOL	BYTE	WORD	DWORD	LWORD	USINT	UINT	UDINT	ULINT	SINT	INT	DINT	LINT	REAL	LREAL	TIME	DATE	TOD	DT	STRING
Axis		AXIS_REF_SM3																		
Execute	√	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
Position	-	-	-	-	-	-	-	-	-	-	-	-	-	√	-	-	-	-	-	-
Mode	√	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
Done	√	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
Busy	√	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
Error	√	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
ErrorID		SMC_ERROR																		

**⊙ Functional Description**

The rising edge of Execute will trigger the execution of this instruction. After the instruction is executed, it will set the Position parameter to the current position of the axis according to the Mode setting mode, but it will not cause the axis to move.

**⊙ Program demo**

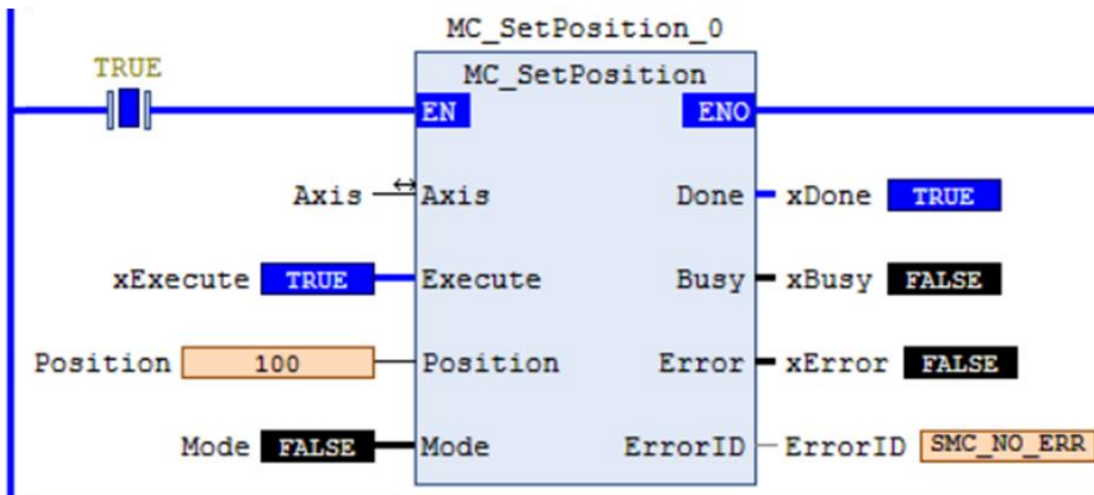
**ST:** When xExecute becomes TRUE, set the Position parameter to the current position of the axis.

```

MC_SetPosition_0(
  Axis:= Axis,
  Execute TRUE := xExecute TRUE ,
  Position 100 := Position 100 ,
  Mode FALSE := Mode FALSE ,
  Done TRUE => xDone TRUE ,
  Busy FALSE => xBusy FALSE ,
  Error FALSE => xError FALSE ,
  ErrorID SMC_NO_ERR => ErrorID SMC_NO_ERR ) ;RETURN

```

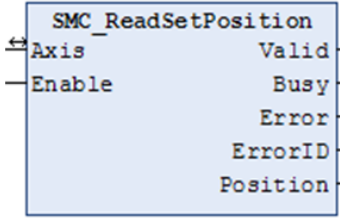
**LD:** When xExecute becomes TRUE, set the Position parameter to the current position of the axis.



### 4.1.14 SMC\_ReadSetPosition

Sets the current position of the axis (does not cause the axis to move).

#### ⊙ Command Format

Instruction	Name	FB/FC	LD	ST	File
SMC_ReadSetPosition	Read set position instruction	FB		SMC_ReadSetPosition( Axis:=, Enable:=, Valid=>, Busy=>, Error=>, ErrorID=>, Position=>);	SM3_Basic

#### ⊙ Related Variables

##### I/O variable

I/O variable	Name	Data type	Range	initialization	Descriptive
Axis	Axis	AXIS_REF_SM3	-	-	Specified axis

##### Input variable

Input variable	Name	Data type	Range	initialization	Descriptive
Enable	Start	BOOL	TRUE-FALSE	FALSE	TRUE to activate the use of function blocks.

##### Output variable.

Output variable	Name	Data type	Range	initialization	Descriptive
Valid	Getting Flags	BOOL	TRUE-FALSE	FALSE	TRUE if the parameter has been read.
Busy	Execution in progress	BOOL	TRUE-FALSE	FALSE	TRUE when function block execution has not finished.
Error	Error	BOOL	TRUE-FALSE	FALSE	Function Block Execution Error
ErrorID	Error Code	SMC_ERROR	-	0	Error indication, see SMC_Error.
Position	Position	LREAL	All	0	Set position value

	Boolean	Bit string				Integer							Real number		Moment, Duration, Date, String					
	BOOL	BYTE	WORD	DWORD	LWORD	USINT	UINT	UDINT	ULINT	SINT	INT	DINT	LINT	REAL	LREAL	TIME	DATE	TOD	DT	STRING
Axis		AXIS_REF_SM3																		
Enable	√	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
Valid	√	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
Busy	√	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
Error	√	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
ErrorID		SMC_ERROR																		
Position	-	-	-	-	-	-	-	-	-	-	-	-	-	-	√	-	-	-	-	-

#### ⊙ Functional Description

TRead the set position of the axis.

Enable is false, all status outputs will be set to false; Enable is true, the set position of the axis is continuously read.

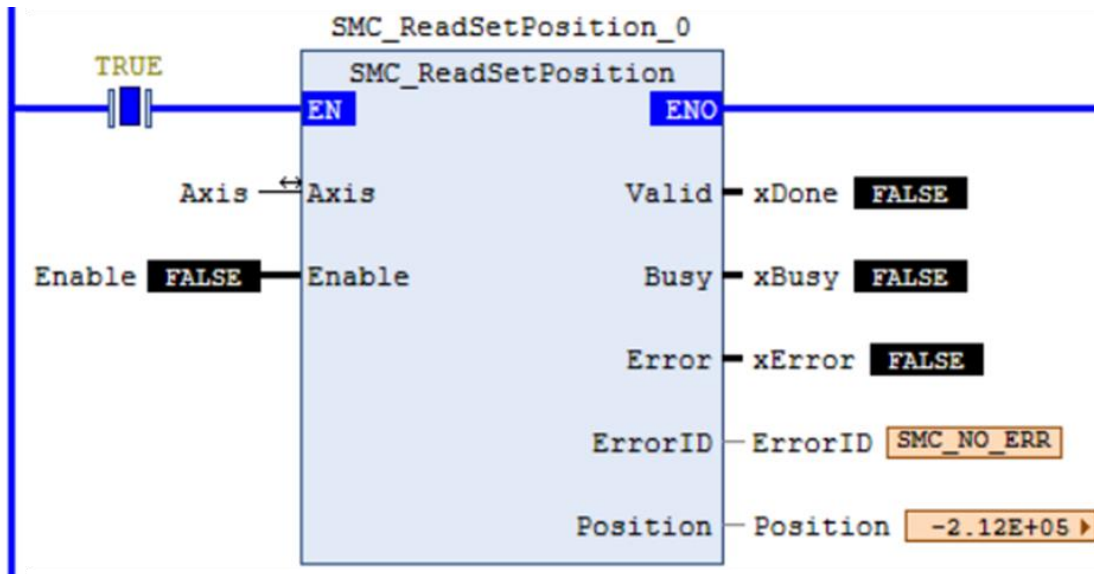
**⊙ Program demo**

**ST:** When Enable is TRUE, the set position of the axis is continuously read.

```

SMC_ReadSetPosition_0(
  Axis:= Axis,
  Enable TRUE := Enable TRUE,
  Valid TRUE => Valid TRUE,
  Busy TRUE => xBusy TRUE,
  Error FALSE => xError FALSE,
  ErrorID SMC_NO_ERR => ErrorID SMC_NO_ERR,
  Position -2.12E+05 => Position -2.12E+05 ); RETURN
  
```

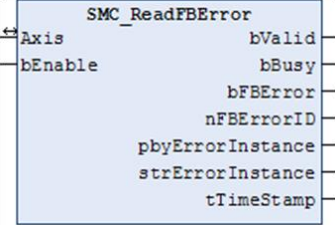
**LD:** When Enable is TRUE, the set position of the axis is continuously read.



### 4.1.15 SMC\_ReadFBError

Sets the current position of the axis (does not cause the axis to move).

#### ⊙ Command Format

Instruction	Name	FB/FC	LD	ST	File
SMC_ReadFBError	Read historical error messages instruction	FB		SMC_ReadFBError( Axis:=, bEnable:=, bValid=>, bBusy=>, bFBError=>, nFBErrorID=>, pbyErrorInstance=>, strErrorInstance=>, tTimeStamp=>);	SM3_Basic

#### ⊙ Related Variables

##### I/O variable

I/O variable	Name	Data type	Range	initialization	Descriptive
Axis	Axis	AXIS_REF_SM3	-	-	Specified axis

##### Input variable

Input variable	Name	Data type	Range	initialization	Descriptive
bEnable	Efficiently	BOOL	TRUE-FALSE	FALSE	TRUE to activate the use of function blocks.

##### Output variable.

Output variable	Name	Data type	Range	initialization	Descriptive
bValid	Getting Flags	BOOL	TRUE-FALSE	FALSE	TRUE if the parameter has been read.
bBusy	Execution in progress	BOOL	TRUE-FALSE	FALSE	TRUE when function block execution has not finished.
bFBError	Error	BOOL	TRUE-FALSE	FALSE	Function Block Execution Error
nFBErrorID	Error Code	SMC_ERROR	-	0	Error indication, see SMC Error.
pbyErrorInstance	Error pointer	POINTER TO BYTE	-	-	Pointer to the FB that reported the error
strErrorInstance		STRING	-	-	Pointer to error function block (program, subroutine, function block)
tTimeStamp	Timestamp	TIME	-	0	Timestamp when the error occurred

	Boolean	Bit string					Integer						Real number		Moment, Duration, Date, String					
	BOOL	BYTE	WORD	DWORD	LWORD	USINT	UINT	UDINT	ULINT	SINT	INT	DINT	LINT	REAL	LREAL	TIME	DATE	TOD	DT	STRING
Axis		AXIS_REF_SM3																		
bEnable	√	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
bValid	√	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
bBusy	√	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
bFBError	√	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
nFBErrorID		SMC_ERROR																		

pbyErrorInstance	POINTER TO BYTE																			
strErrorInstance	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	√
tTimeStamp	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	√	-	-	-	-

**⊙ Functional Description**

If Enable is TRUE, Valid, Busy outputs TRUE if there is no error; bFBError outputs true if there is a function block alarm.

If Enable is FALSE, then Valid, Busy outputs FALSE.

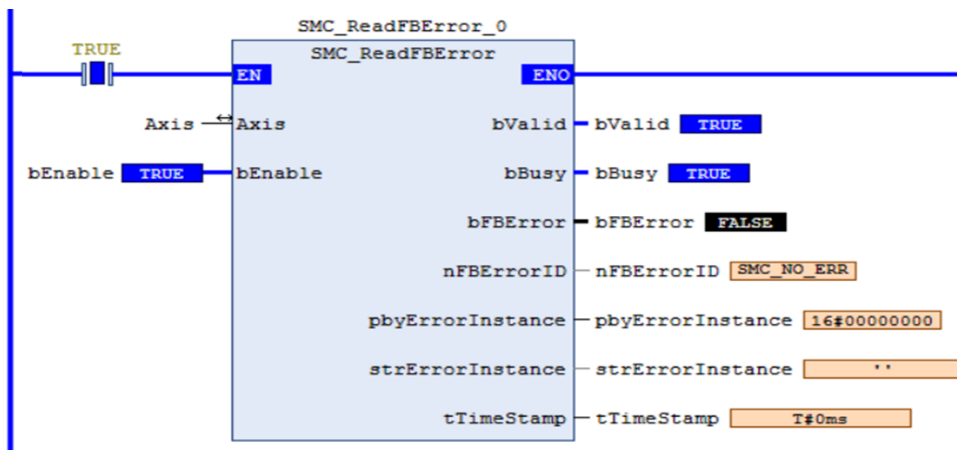
**⊙ Program demo**

**ST:** When Enable is TRUE, the axis function block error message is read.

```

SMC_ReadFBError_0(
  Axis:= Axis,
  bEnable TRUE := bEnable TRUE,
  bValid TRUE => bValid TRUE,
  bBusy TRUE => bBusy TRUE,
  bFBError FALSE => bFBError FALSE,
  nFBErrorID SMC_NO_ERR => nFBErrorID SMC_NO_ERR,
  pbyErrorInstance 16#00000000 => pbyErrorInstance 16#00000000,
  strErrorInstance "" => strErrorInstance "",
  tTimeStamp T#0ms => tTimeStamp T#0ms ) :RETURN
  
```

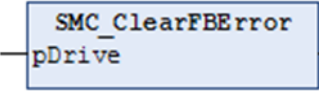
**LD:** When Enable is TRUE, the axis function block error message is read.



### 4.1.16 SMC\_ClearFBError

Clearing Historical Error Messages for Function Blocks.

#### ⊙ Command Format

Instruction	Name	FB/FC	LD	ST	File
SMC_ClearFBError	Clear Historical Error Messages instruction	FC		SMC_ClearFBError(pDrive:=);	SM3_Basic

#### ⊙ Related Variables

##### Input variable

Input variable	Name	Data type	Range	initialization	Descriptive
pDrive	Axis pointer	POINTER TO AXIS_REF_SM3	-	-	Specified axis

##### Output variable

Output variable	Name	Data type	Range	initialization	Descriptive
SMC_ClearFBError	return value	BOOL	TRUE-FALSE	FALSE	FALSE if the function ends normally.

	Boolean	Bit string					Integer							Real number		Moment, Duration, Date, String				
	BOOL	BYTE	WORD	DWORD	LWORD	USINT	UINT	UDINT	ULINT	SINT	INT	DINT	LINT	REAL	LREAL	TIME	DATE	TOD	DT	STRING
pDrive		POINTER TO AXIS_REF_SM3																		
SMC_ClearFBError	√	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-

#### ⊙ Functional Description

This function block needs to be called to clear the historical error status of the axis when an error occurs and the reset function block is called to reset the axis.

The rising edge of Execute will trigger the execution of this instruction.

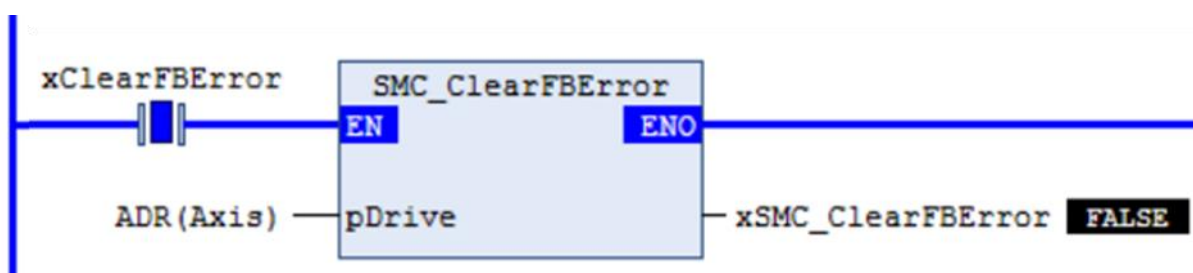
#### ⊙ Program demo

**ST:** Clears the Axis historical error state when xClearFBError becomes TRUE.

```

IF xClearFBError TRUE THEN
    xSMC_ClearFBError FALSE := SMC_ClearFBError(pDrive:= ADR(Axis) );
END_IF
RETURN
    
```

**LD:** Clears the Axis historical error state when xClearFBError becomes TRUE.



## 4.2 Single-axis motion control

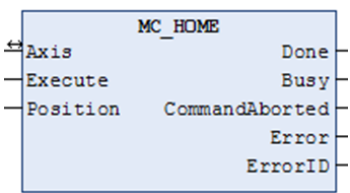
### 4.2.1 Instruction List

Instruction Type	Name	FB/FC	Functionality
Axis motion control	MC Home	FB	Home
	MC MoveAbsolute	FB	Absolute motion
	MC MoveRelative	FB	Relative motion
	MC MoveVelocity	FB	Constant velocity
	MC_Stop	FB	Stop
	MC Halt	FB	Halt
	MC Jog	FB	Jog
	MC MoveAdditive	FB	Additive motion
	MC MoveSuperImposed	FB	Super Imposed motion
	MC PositionProfile	FB	Position Profile
	MC VelocityProfile	FB	Velocity Profile
	MC AccelerationProfile	FB	Acceleration Profile
	SMC Homing	FB	OP homing
	SMC Inch	FB	Move an inch

### 4.2.2 MC\_Home

The homing action of the EtherCAT bus motor, the specific homing process is determined by the homing mode of the bus motor design.

#### ⊙ Command Format

Instruction	Name	FB/FC	LD	ST	File
MC_Home	Homing instruction	FC		MC_Home(Axis:=, Execute:=, Position:=, Done=>, Busy=>, CommandAborted=>, Error=>, ErrorID=>);	SM3_Basic

#### ⊙ Related Variables

##### I/O variable

I/O variable	Name	Data type	Range	initialization	Descriptive
Axis	Axis	AXIS_REF_SM3	-	-	Specified axis

##### Input variable

Input variable	Name	Data type	Range	initialization	Descriptive
Execute	start	BOOL	TRUE-FALSE	FALSE	Rising edge trigger
Position	Position	LREAL	ALL	0	Drive homing bias 16#607C

##### Output variable

Output variable	Name	Data type	Range	initialization	Descriptive
Done	Completion	BOOL	TRUE-FALSE	FALSE	TRUE: Homing is done.
Busy	Executing	BOOL	TRUE-FALSE	FALSE	TRUE: In progress
Command Aborted	Command Interrupt	BOOL	TRUE-FALSE	FALSE	TRUE: Terminated by other commands
Error	Error	BOOL	TRUE-FALSE	FALSE	TRUE: An error occurred within the function block
ErrorID	Error Code	SMC_ERRO R	-	0	Error indication, see SMC Error.

	Boolean	Bit string					Integer							Real number		Moment, Duration, Date, String				
	BOOL	BYTE	WORD	DWORD	LWORD	USINT	UINT	UDINT	ULINT	SINT	INT	DINT	LINT	REAL	LREAL	TIME	DATE	TOD	DT	STRING
Axis	AXIS_REF_SM3																			
Execute	√	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
Position	-	-	-	-	-	-	-	-	-	-	-	-	-	-	√	-	-	-	-	-
Done	√	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
Busy	√	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
Command Aboned	√	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
Error	√	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
ErrorID	SMC_ERROR																			

### ⊙ Functional Description

The homing motion instruction for the bus drive needs to be called when the motion axis being controlled is a bus axis in order to homing the motion axis.

The homing mode of the homing instruction is decided by the slave connected to the bus. The control terminal only sends the required homing parameters to the driver, and the specific homing action is completed at the driver terminal.

The standard drive's homing offset address is 16#607C, which is handled differently by each drive, please refer to the description of the homing process and homing parameters for that drive.

Before performing a homing, you need to configure the bus driver's homing parameters such as homing mode, speed, acceleration, etc. Please refer to the manual of the driver you are using for which parameters need to be configured for bus driver homing.

Standard bus drive homing requires setting the data for indexes and sub-indexes. This is shown in the table below.

parametric	indexing	subindex	descriptions
Homing method	0x6098		Drive homing methods vary by manufacturer, so you need to select the appropriate homing method according to the specific drive manufacturer's manual.
Homing velocity(fast)	0x6099	0x01	The speed of the homing process from the start of homing to the home position, with a higher value to reduce the homing time.
Homing velocity(slow)	0x6099	0x02	The speed of the process from finding the origin to the completion of homing, with a lower value to improve accuracy
Homing acceleration	0x609A		Acceleration and deceleration changes during home positioning

### Three ways to set the regression mode:

- 1) The controller writes the return-to-zero parameter to the drive via the SDO service;
- 2) Setting back to zero by configuring the PDO parameters and mapping the relevant back-to-zero parameters;
- 3) Write back to zero parameters to the drive by configuring the startup parameters.

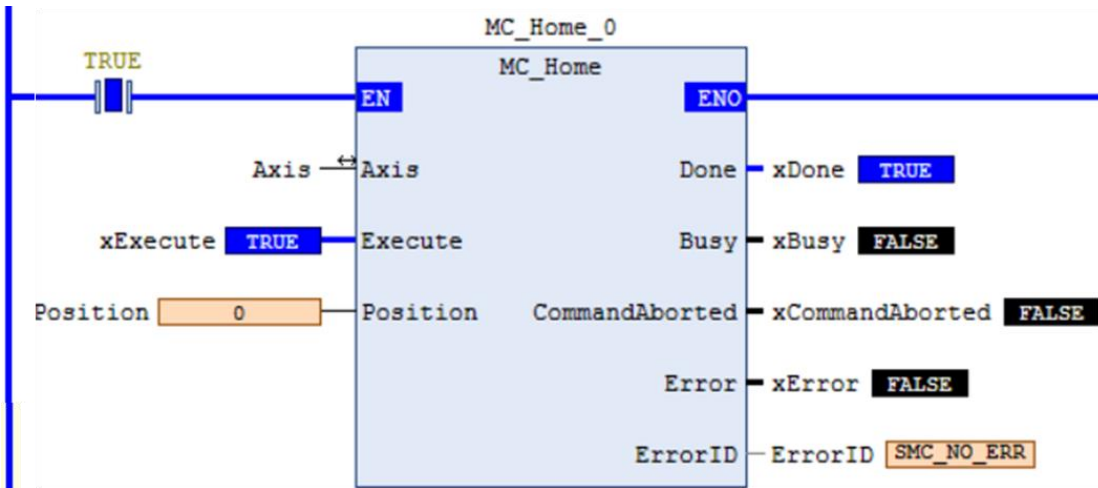
⊙ **Program demo**

**ST:** When xExecute becomes TRUE, execute the homing instruction.

```

MC_Home_0(
  Axis:= Axis,
  Execute TRUE := xExecute TRUE,
  Position 0 := Position 0,
  Done TRUE => xDone TRUE,
  Busy FALSE => xBusy FALSE,
  CommandAborted FALSE => xCommandAborted FALSE,
  Error FALSE => xError FALSE,
  ErrorID SMC_NO_ERR => ErrorID SMC_NO_ERR ); RETURN
  
```

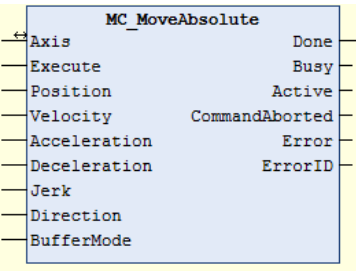
**LD:** When xExecute becomes TRUE, execute the homing instruction.



### 4.2.3 MC\_MoveAbsolute

Moves the control axis to the specified absolute position according to the set parameters.

#### ⊙ Command Format

Instruction	Name	FB/FC	LD	ST	File
MC_MoveAbsolute	Absolute motion instruction	FC		MC_MoveAbsolute( Axis:=, Execute:=, Position:=, Velocity:=, Acceleration:=, Deceleration:=, Jerk:=, Direction:=, BufferMode:=, Done=>, Busy=>, Active=>, CommandAborted=>, Error=>, ErrorID=>);	SM3_Basic

#### ⊙ Related Variables

##### I/O variable

I/O variable	Name	Data type	Range	initialization	Descriptive
Axis	Axis	AXIS REF SM3	-	-	Specified axis

##### Input variable

Input variable	Name	Data type	Range	initialization	Descriptive
Execute	start	BOOL	TRUE-FALSE	FALSE	Rising edge trigger
Position	Position	LREAL	ALL	0	Target position of movement [u] (negative or positive)
Velocity	Target velocity	LREAL	Is always positive.	0	Maximum velocity in technical units per second [u/s]. Is not necessarily reached.
Acceleration	Target acceleration	LREAL	Is always positive.	0	Acceleration in [u/s <sup>2</sup> ]. Increasing motor energy
Deceleration	Target Deceleration	LREAL	Is always positive.	0	Deceleration in [u/s <sup>2</sup> ]. Decreasing motor energy
Jaek	Target jaek	LREAL	Is always positive.	0	Maximum magnitude of jerk in [u/s <sup>3</sup> ]
Direction	Direction	MC_DIRECTION	-1, 1	Positive	1, Positive; -1, Negative
BufferMode	Cache Mode	MC_BUFFER_MODE	0-5	0	Specifies the action to be taken when a multi-initiation motion instruction is initiated. 0: mcAborting, interrupt; 1: mcBuffered, wait; 2: mcBlendingLow, merging at low speed; 3: mcBlendingPrevious, merge at previous speed; 4: mcBlendingNext, merging at the next speed; 5: mcBlendingHigh, merge at high speed

**Output variable**

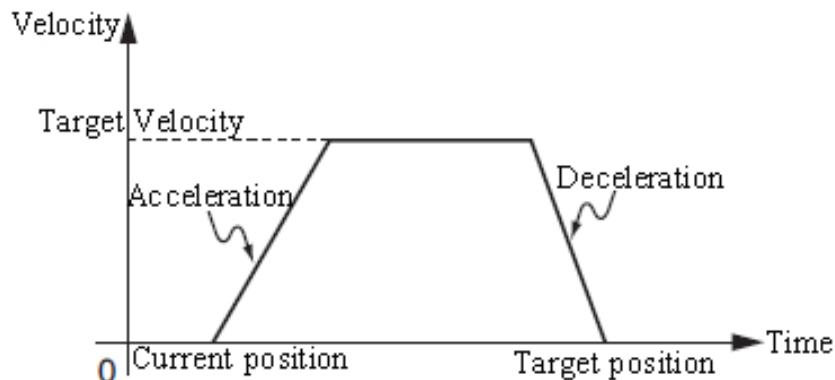
Output variable	Name	Data type	Range	initialization	Descriptive
Done	Completion	BOOL	TRUE-FALSE	FALSE	TRUE: Homing is done.
Busy	Executing	BOOL	TRUE-FALSE	FALSE	TRUE: In progress
Active		BOOL			Axis is working
Command Aborted	Command Interrupt	BOOL	TRUE-FALSE	FALSE	TRUE: Terminated by other commands
Error	Error	BOOL	TRUE-FALSE	FALSE	TRUE: An error occurred within the function block
ErrorID	Error Code	SMC_ERROR	-	0	Error indication, see SMC_Error.

	Boolean	Bit string					Integer							Real number		Moment, Duration, Date, String				
	BOOL	BYTE	WORD	DWORD	LWORD	USINT	UINT	UDINT	ULINT	SINT	INT	DINT	LINT	REAL	LREAL	TIME	DATE	TOD	DT	STRING
Axis	AXIS_REF_SM3																			
Execute	√	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
Position	-	-	-	-	-	-	-	-	-	-	-	-	-	-	√	-	-	-	-	-
Velocity	-	-	-	-	-	-	-	-	-	-	-	-	-	-	√	-	-	-	-	-
Acceleration	-	-	-	-	-	-	-	-	-	-	-	-	-	-	√	-	-	-	-	-
Deceleration	-	-	-	-	-	-	-	-	-	-	-	-	-	-	√	-	-	-	-	-
Jerk	-	-	-	-	-	-	-	-	-	-	-	-	-	-	√	-	-	-	-	-
Direction	MC_DIRECTION																			
BufferMode	MC_BUFFER_MODE																			
Done	√	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
Busy	√	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
Active	√	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
Command Aboned	√	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
Error	√	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
ErrorID	SMC_ERROR																			

**⊙ Functional Description**

In case of absolute pointwise motion of a linear axis, the direction value will be ignored.

When the speed curve is a trapezoidal curve, the "speed-time" curve when the instruction is executed is shown below.



If the distance is too short, the maximum speed may not be reached.

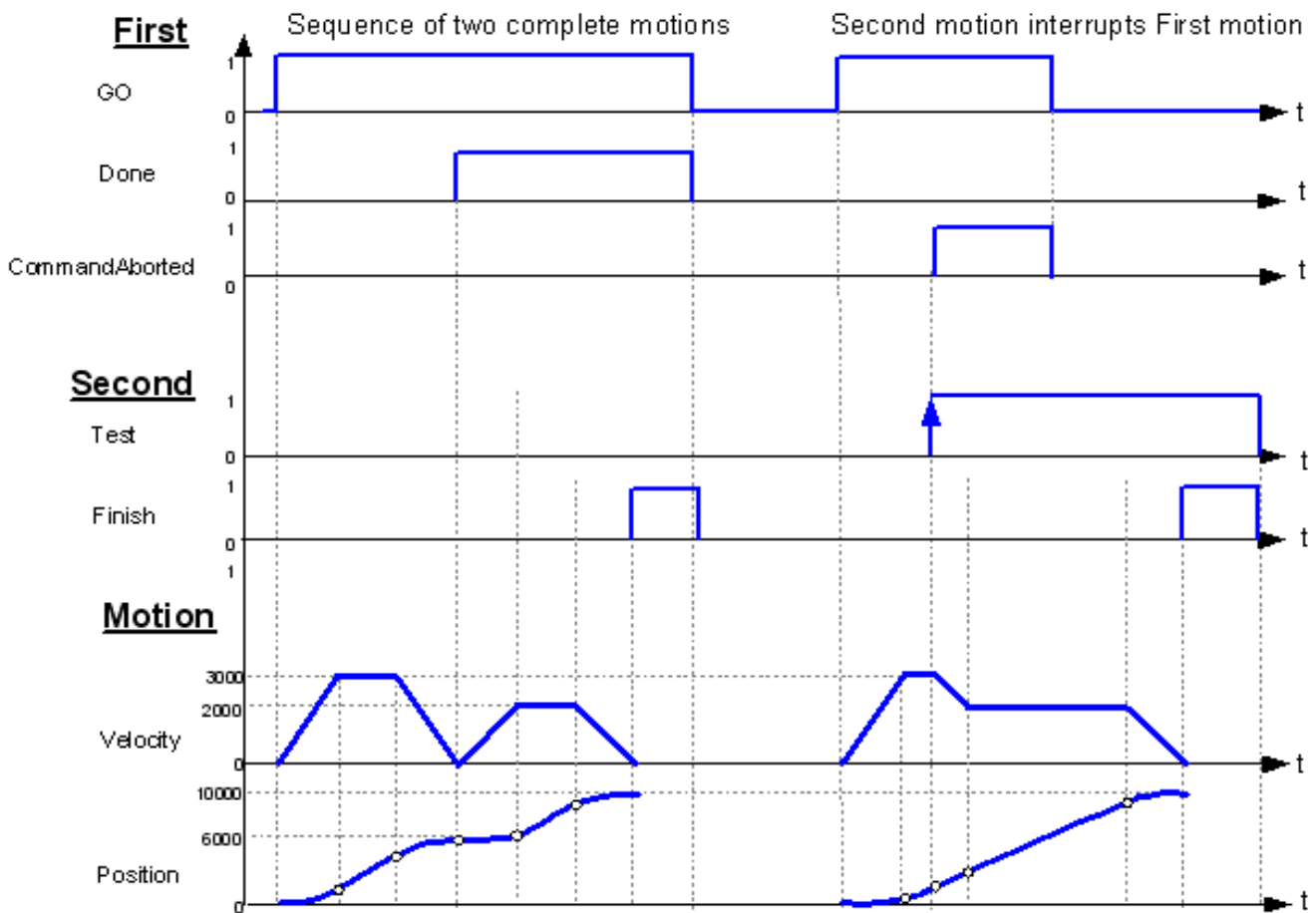
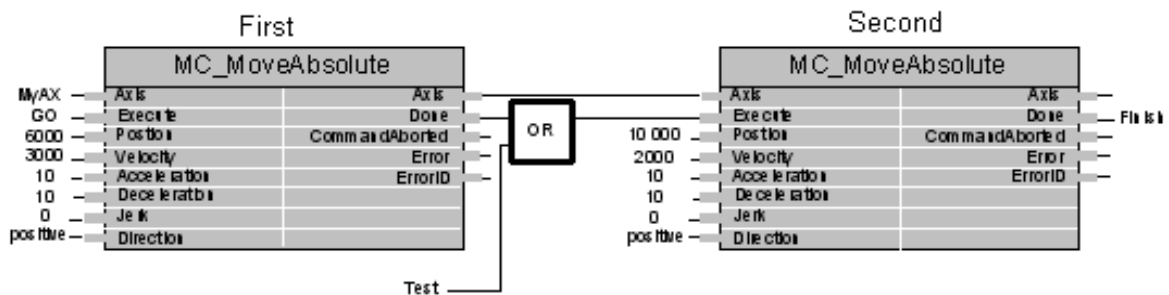
Setting the acceleration/deceleration speed to 0 will reach the target speed directly without acceleration/deceleration.

### MC\_MoveAbsolute Demo

The following is a routine that connects two instances of the MC\_MoveAbsolute function block, "First" and "Second":

- 1) The first part of the timing diagram shows the situation when "Second" is called after "First". When "First" reaches the required position 6000 (speed 0), the output Done will trigger "Second" to move to position 10000.
- 2) The latter part of the timing diagram shows the case when "Second" is activated while "First" is still executing. In this case, the movement of "First" is terminated by the Test signal during the constant velocity. The "Second" function block will move directly to position 10000, although it does not reach position 6000.

### MoveAbsolute - Example



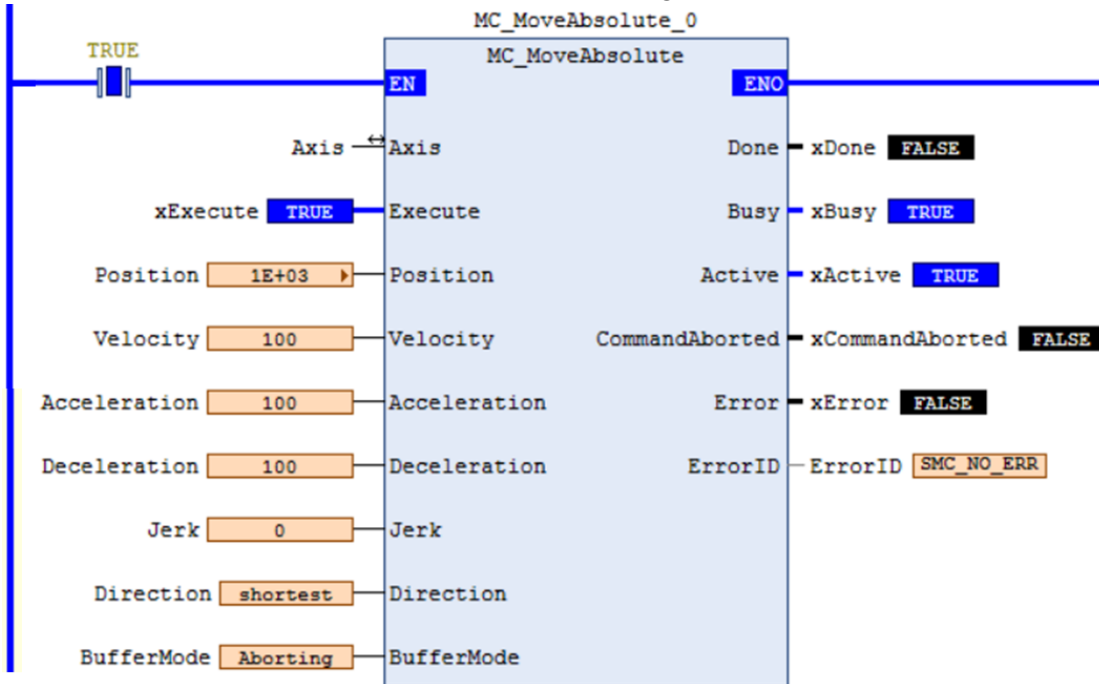
**⊙ Program demo**

**ST:** When xExecute becomes TRUE, execute the homing instruction.

```

MC_MoveAbsolute_0(
  Axis:= Axis,
  Execute TRUE := xExecute TRUE,
  Position  := Position ,
  Velocity  := Velocity ,
  Acceleration  := Acceleration ,
  Deceleration  := Deceleration ,
  Jerk  := Jerk ,
  Direction  := Direction ,
  BufferMode  := BufferMode ,
  Done FALSE => xDone FALSE,
  Busy TRUE => xBusy TRUE,
  Active TRUE => xActive TRUE,
  CommandAborted FALSE => xCommandAborted FALSE,
  Error FALSE => xError FALSE,
  ErrorID  => ErrorID ); RETURN
  
```

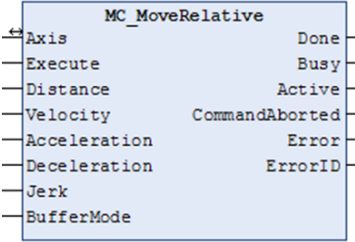
**LD:** When xExecute becomes TRUE, execute the homing instruction.



### 4.2.4 MC\_MoveRelative

Move the control axis a relative distance according to the set parameters.

#### ⊙ Command Format

Instruction	Name	FB/FC	LD	ST	File
MC_MoveRelative	Relative motion instruction	FC		MC_MoveRelative( Axis:=, Execute:=, Distance:=, Velocity:=, Acceleration:=, Deceleration:=, Jerk:=, BufferMode:=, Done=>, Busy=>, Active=>, CommandAborted=>, Error=>, ErrorID=>);	SM3_Basic

#### ⊙ Related Variables

##### I/O variable

I/O variable	Name	Data type	Range	initialization	Descriptive
Axis	Axis	AXIS REF SM3	-	-	Specified axis

##### Input variable

Input variable	Name	Data type	Range	initialization	Descriptive
Execute	Start	BOOL	TRUE-FALSE	FALSE	Rising edge trigger
Distance	Distance	LREAL	ALL	0	Relative distance between the target position and the current position [u] (negative or positive)
Velocity	Target velocity	LREAL	Is always positive.	0	Maximum velocity in technical units per second [u/s]. Is not necessarily reached.
Acceleration	Target acceleration	LREAL	Is always positive.	0	Acceleration in [u/s <sup>2</sup> ]. Increasing motor energy
Deceleration	Target Deceleration	LREAL	Is always positive.	0	Deceleration in [u/s <sup>2</sup> ]. Decreasing motor energy
Jack	Target jaek	LREAL	Is always positive.	0	Maximum magnitude of jerk in [u/s <sup>3</sup> ]
BufferMode	Cache Mode	MC_BUFFER_MODE	0-5	0	Specifies the action to be taken when a multi-initiation motion instruction is initiated. 0: mcAborting, interrupt; 1: mcBuffered, wait; 2: mcBlendingLow, merging at low speed; 3: mcBlendingPrevious, merge at previous speed; 4: mcBlendingNext, merging at the next speed; 5: mcBlendingHigh, merge at high speed

**Output variable**

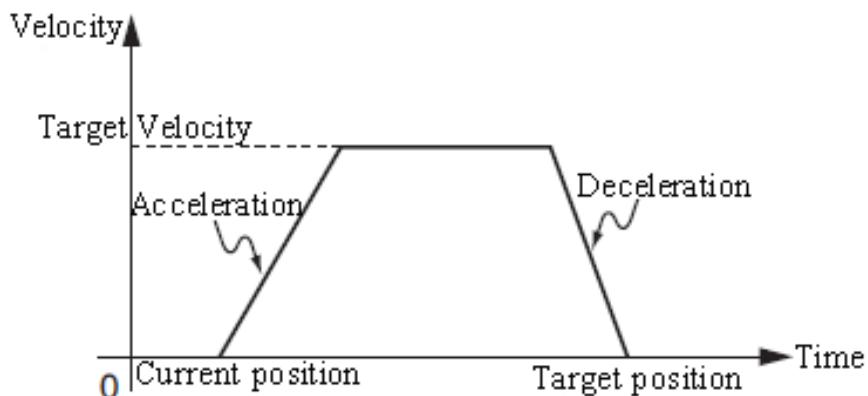
Output variable	Name	Data type	Range	initialization	Descriptive
Done	Completion	BOOL	TRUE-FALSE	FALSE	TRUE: Homing is done.
Busy	Executing	BOOL	TRUE-FALSE	FALSE	TRUE: In progress
Active		BOOL			Axis is working
Command Aborted	Command Interrupt	BOOL	TRUE-FALSE	FALSE	TRUE: Terminated by other commands
Error	Error	BOOL	TRUE-FALSE	FALSE	TRUE: An error occurred within the function block
ErrorID	Error Code	SMC_ERROR	-	0	Error indication, see SMC_Error.

	Boolean	Bit string					Integer							Real number		Moment, Duration, Date, String				
	BOOL	BYTE	WORD	DWORD	LWORD	USINT	UINT	UDINT	ULINT	SINT	INT	DINT	LINT	REAL	LREAL	TIME	DATE	TOD	DT	STRING
Axis	AXIS_REF_SM3																			
Execute	√	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
Distance	-	-	-	-	-	-	-	-	-	-	-	-	-	-	√	-	-	-	-	-
Velocity	-	-	-	-	-	-	-	-	-	-	-	-	-	-	√	-	-	-	-	-
Acceleration	-	-	-	-	-	-	-	-	-	-	-	-	-	-	√	-	-	-	-	-
Deceleration	-	-	-	-	-	-	-	-	-	-	-	-	-	-	√	-	-	-	-	-
Jerk	-	-	-	-	-	-	-	-	-	-	-	-	-	-	√	-	-	-	-	-
BufferMode	MC_BUFFER_MODE																			
Done	√	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
Busy	√	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
Active	√	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
Command Aborted	√	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
Error	√	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
ErrorID	SMC_ERROR																			

**⊙ Functional Description**

Relative motion with the current position as the origin, the end point coordinate is the distance from the start point to the end point.

When the speed curve is a trapezoidal curve, the "speed-time" curve when the instruction is executed is shown below.

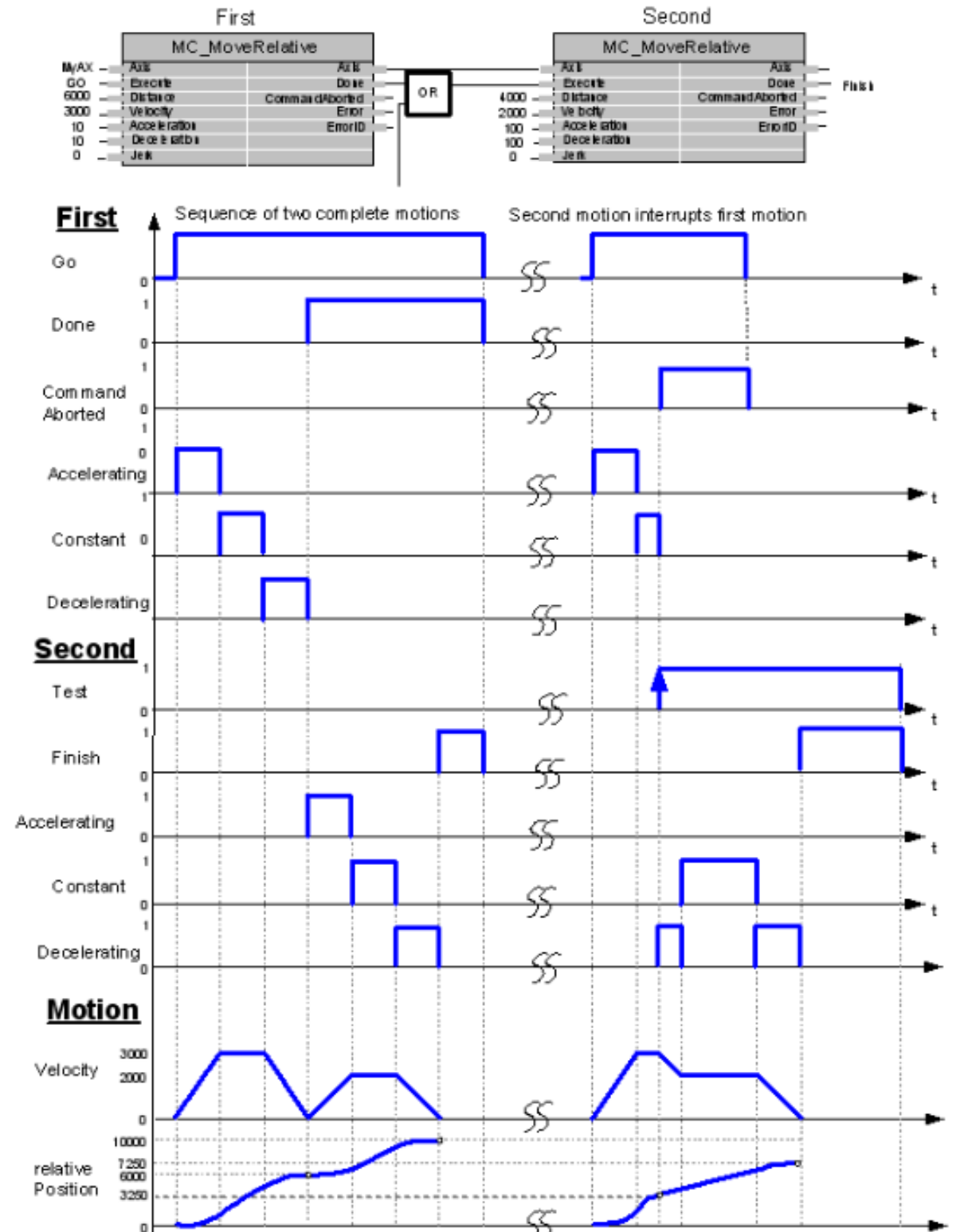


**MC\_MoveRelative Demo**

The following is a routine that connects two instances of the MC\_MoveRelative function block, "First" and "Second":

1) The first part of the timing diagram shows the situation when "Second" is called after "First". When "First" reaches the required position 6000 (speed 0), the output Done will trigger "Second" to move to position 10000.

2) The latter part of the timing diagram shows the case when "Second" is started while "First" is still executing. In this case, "First" is terminated by the Test signal during constant speed. After adding the distance 4000 to the actual position 3250, the "Second" function block moves the axis to the new position 7250.

**MoveRelative - Example**


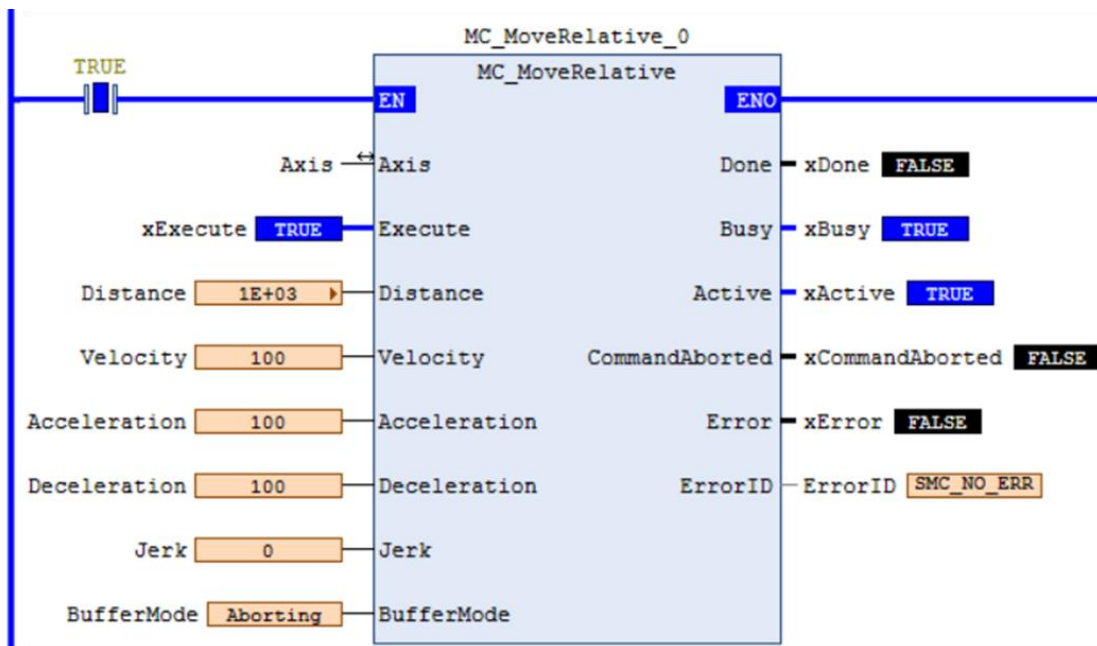
**⊙ Program demo**

**ST:** When xExecute becomes TRUE, execute the homing instruction.

```

MC_MoveRelative_0(
  Axis:= Axis,
  Execute TRUE := xExecute TRUE,
  Distance 1E+03 := Distance 1E+03,
  Velocity 100 := Velocity 100,
  Acceleration 100 := Acceleration 100,
  Deceleration 100 := Deceleration 100,
  Jerk 0 := Jerk 0,
  BufferMode Aborting := BufferMode Aborting,
  Done FALSE => xDone FALSE,
  Busy TRUE => xBusy TRUE,
  Active TRUE => xActive TRUE,
  CommandAborted FALSE => xCommandAborted FALSE,
  Error FALSE => xError FALSE,
  ErrorID SMC_NO_ERR => ErrorID SMC_NO_ERR);
  
```

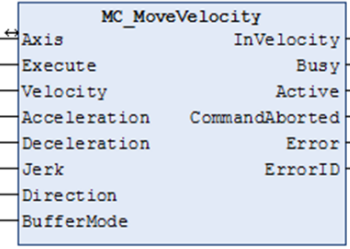
**LD:** When xExecute becomes TRUE, execute the homing instruction.



### 4.2.5 MC\_MoveVelocity

The control axis maintains a constant speed movement according to the set parameters.

#### ⊙ Command Format

Instruction	Name	FB/FC	LD	ST	File
MC_MoveVelocity	Constant velocity instruction	FC		MC_MoveVelocity( Axis:=, Execute:=, Velocity:=, Acceleration:=, Deceleration:=, Jerk:=, Direction:=, BufferMode:=, InVelocity=>, Busy=>, Active=>, CommandAborted=>, Error=>, ErrorID=>);	SM3_Basic

#### ⊙ Related Variables

##### I/O variable

I/O variable	Name	Data type	Range	initialization	Descriptive
Axis	Axis	AXIS_REF_SM3	-	-	Specified axis

##### Input variable

Input variable	Name	Data type	Range	initialization	Descriptive
Execute	Start	BOOL	TRUE-FALSE	FALSE	Rising edge trigger
Velocity	Target velocity	LREAL	Is always positive.	0	Maximum velocity in technical units per second [u/s]. Is not necessarily reached.
Acceleration	Target acceleration	LREAL	Is always positive.	0	Acceleration in [u/s <sup>2</sup> ]. Increasing motor energy
Deceleration	Target Deceleration	LREAL	Is always positive.	0	Deceleration in [u/s <sup>2</sup> ]. Decreasing motor energy
Jaek	Target jaek	LREAL	Is always positive.	0	Maximum magnitude of jerk in [u/s <sup>3</sup> ]
Direction	Direction	MC_DIRECTION	-1, 1	Positive	1, Positive; -1, Negative
BufferMode	Cache Mode	MC_BUFFER_MODE	0-5	0	Specifies the action to be taken when a multi-initiation motion instruction is initiated. 0: mcAborting, interrupt; 1: mcBuffered, wait; 2: mcBlendingLow, merging at low speed; 3: mcBlendingPrevious, merge at previous speed; 4: mcBlendingNext, merging at the next speed; 5: mcBlendingHigh, merge at high speed

**Output variable**

Output variable	Name	Data type	Range	initialization	Descriptive
InVelocity	Set velocity has been reached	BOOL	TRUE-FALSE	FALSE	TRUE: The set velocity has been reached for the first time.
Busy	Executing	BOOL	TRUE-FALSE	FALSE	TRUE: In progress
Active		BOOL			Axis is working
Command Aborted	Command Interrupt	BOOL	TRUE-FALSE	FALSE	TRUE: Terminated by other commands
Error	Error	BOOL	TRUE-FALSE	FALSE	TRUE: An error occurred within the function block
ErrorID	Error Code	SMC_ERRO R	-	0	Error indication, see SMC Error.

	Boo lea n	Bit string				Integer							Real number		Moment, Duration, Date, String					
	BOOL	BYTE	WORD	DWORD	LWORD	USINT	UINT	UDINT	ULINT	SINT	INT	DINT	LINT	REAL	LREAL	TIME	DATE	TOD	DT	STRING
Axis		AXIS_REF_SM3																		
Execute	√	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
Velocity	-	-	-	-	-	-	-	-	-	-	-	-	-	-	√	-	-	-	-	-
Acceleration	-	-	-	-	-	-	-	-	-	-	-	-	-	-	√	-	-	-	-	-
Deceleration	-	-	-	-	-	-	-	-	-	-	-	-	-	-	√	-	-	-	-	-
Jerk	-	-	-	-	-	-	-	-	-	-	-	-	-	-	√	-	-	-	-	-
Direction		MC_DIRECTION																		
BufferMode		MC_BUFFER_MODE																		
InVelocity	√	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
Busy	√	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
Active	√	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
Command Aboned	√	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
Error	√	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
ErrorID		SMC_ERROR																		

**© Functional Description**

The rising edge of Execute starts the constant velocity motion, and after that, the constant velocity motion can only be aborted by other commands.

When the constant velocity motion reaches the set speed, the InVelocity signal will change to TRUE.

When the constant velocity motion is interrupted by another motion, the InVelocity signal will be reset to FALSE.

Negative velocity \* negative direction = positive velocity.

Controls the motor to accelerate to the maximum speed at the specified acceleration, and then keep running at that maximum speed until a stop command or other interrupt command is called to interrupt the command.

**MC\_MoveVelocity Demo**

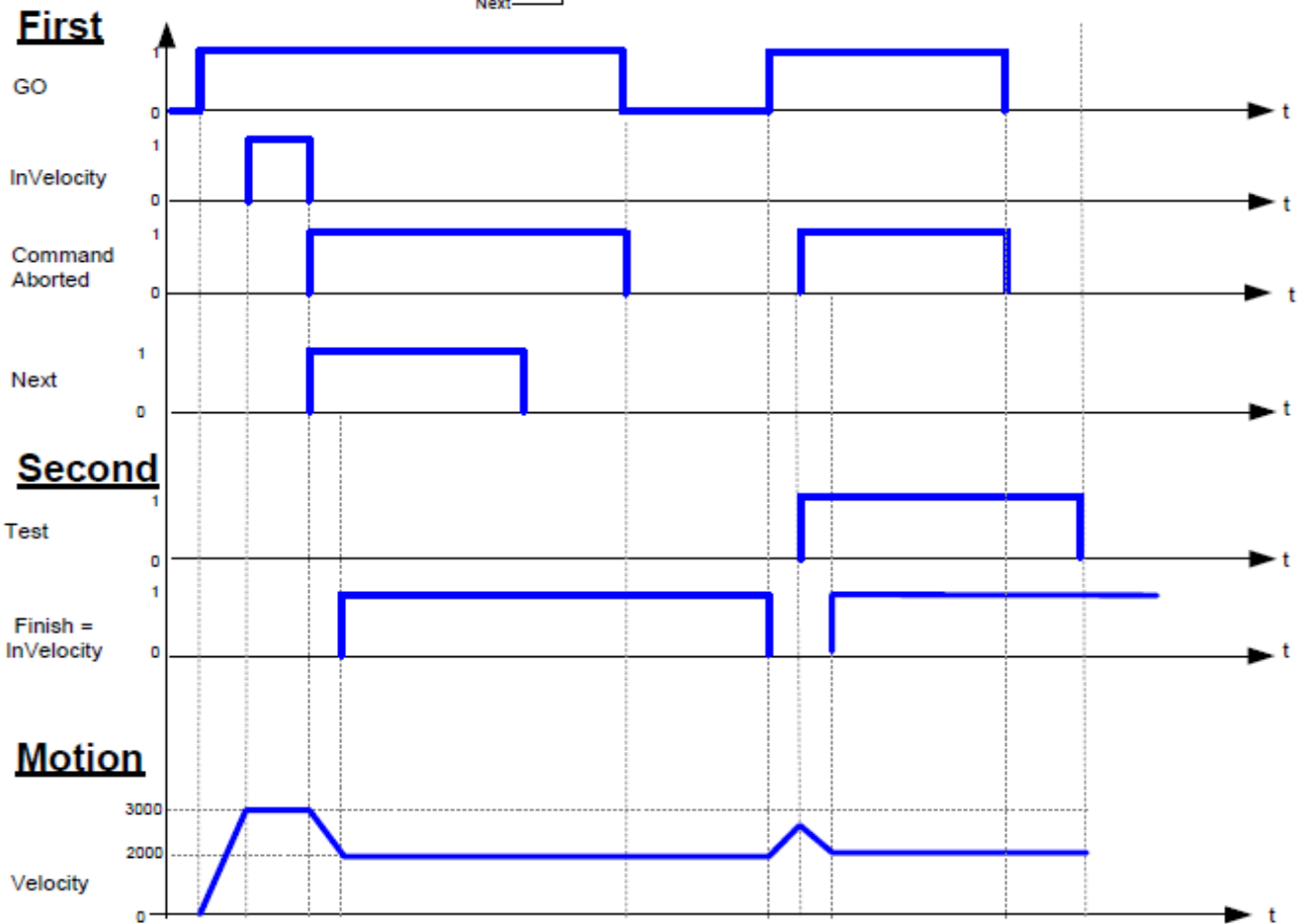
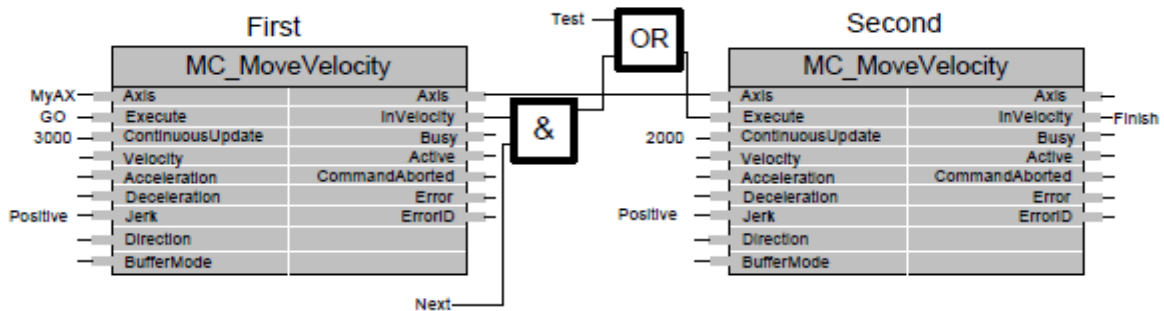
The following is a routine that connects two instances of the MC\_MoveVelocity function block, "First" and "Second":

1) The left part of the time diagram shows the situation when "Second" is called after "First". When "First" reaches the set speed of 3000, the first instruction outputs the InVelocity signal, which performs an "and" operation with

the next instruction, triggering the second instruction to move at a speed of 2000.

2) The right part of the time diagram shows the situation when "Second" is started while "First" is still executing. At this point, the motion of "First" is interrupted and terminated by the Test signal that was transmitted during the period when the speed of "First" was constant. Although the first instruction is still in the process of accelerating towards 3000, the acceleration of the first is interrupted and aborted due to the execution of the second instance in the middle of the process. The second instance will then decelerate to 2000, after which the InVelocity of the second instance is set to TRUE.

### MoveVelocity - Example



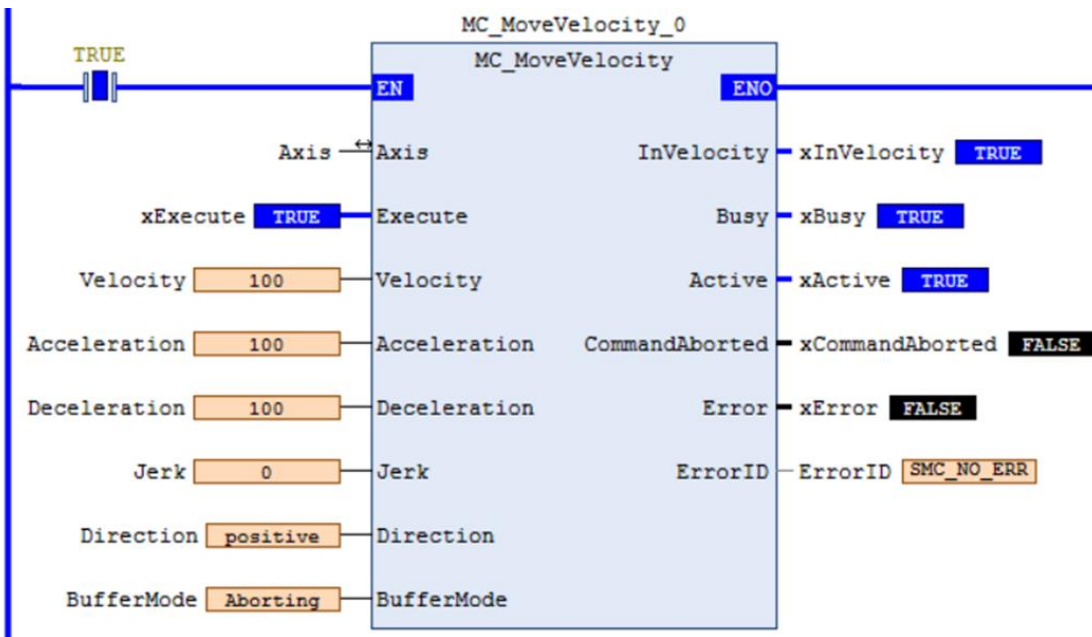
**⊙ Program demo**

**ST:** When xExecute becomes TRUE, the constant velocity motion instruction is executed.

```

MC_MoveVelocity_0(
  Axis:= Axis,
  Execute TRUE := xExecute TRUE,
  Velocity 100 := Velocity 100,
  Acceleration 100 := Acceleration 100,
  Deceleration 100 := Deceleration 100,
  Jerk 0 := Jerk 0,
  Direction positive := Direction positive,
  BufferMode Aborting := BufferMode Aborting,
  InVelocity TRUE => InVelocity TRUE,
  Busy TRUE => xBusy TRUE,
  Active TRUE => xActive TRUE,
  CommandAborted FALSE => xCommandAborted FALSE,
  Error FALSE => xError FALSE,
  ErrorID SMC_NO_ERR => ErrorID SMC_NO_ERR );
  
```

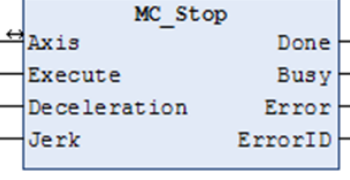
**LD:** When xExecute becomes TRUE, the constant velocity motion instruction is executed.



### 4.2.6 MC\_Stop

The control axis maintains a constant speed movement according to the set parameters.

#### ⊙ Command Format

Instruction	Name	FB/FC	LD	ST	File
MC_Stop	Stop instruction	FC		MC_Stop( Axis:=, Execute:=, Deceleration:=, Jerk:=, Done=>, Busy=>, Error=>, ErrorID=>);	SM3_Basic

#### ⊙ Related Variables

##### I/O variable

I/O variable	Name	Data type	Range	initialization	Descriptive
Axis	Axis	AXIS_REF_SM3	-	-	Specified axis

##### Input variable

Input variable	Name	Data type	Range	initialization	Descriptive
Execute	Start	BOOL	TRUE-FALSE	FALSE	Rising edge trigger
Deceleration	Target Deceleration	LREAL	Is always positive.	0	Deceleration in [u/s <sup>2</sup> ]. Decreasing motor energy
Jack	Target jack	LREAL	Is always positive.	0	Maximum magnitude of jerk in [u/s <sup>3</sup> ]

##### Output variable

Output variable	Name	Data type	Range	initialization	Descriptive
Done	Completion	BOOL	TRUE-FALSE	FALSE	TRUE: Homing is done.
Busy	Executing	BOOL	TRUE-FALSE	FALSE	TRUE: In progress
Error	Error	BOOL	TRUE-FALSE	FALSE	TRUE: An error occurred within the function block
ErrorID	Error Code	SMC_ERRO R	-	0	Error indication, see SMC Error.

	Boo lea n	Bit string				Integer							Real number		Moment, Duration, Date, String					
	BOOL	BYTE	WORD	DWORD	LWORD	USINT	UINT	UDINT	ULINT	SINT	INT	DINT	LINT	REAL	LREAL	TIME	DATE	TOD	DT	STRING
Axis		AXIS_REF_SM3																		
Execute	√	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
Deceleration	-	-	-	-	-	-	-	-	-	-	-	-	-	-	√	-	-	-	-	
Jerk	-	-	-	-	-	-	-	-	-	-	-	-	-	-	√	-	-	-	-	
Done	√	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
Busy	√	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
Error	√	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
ErrorID		SMC_ERROR																		

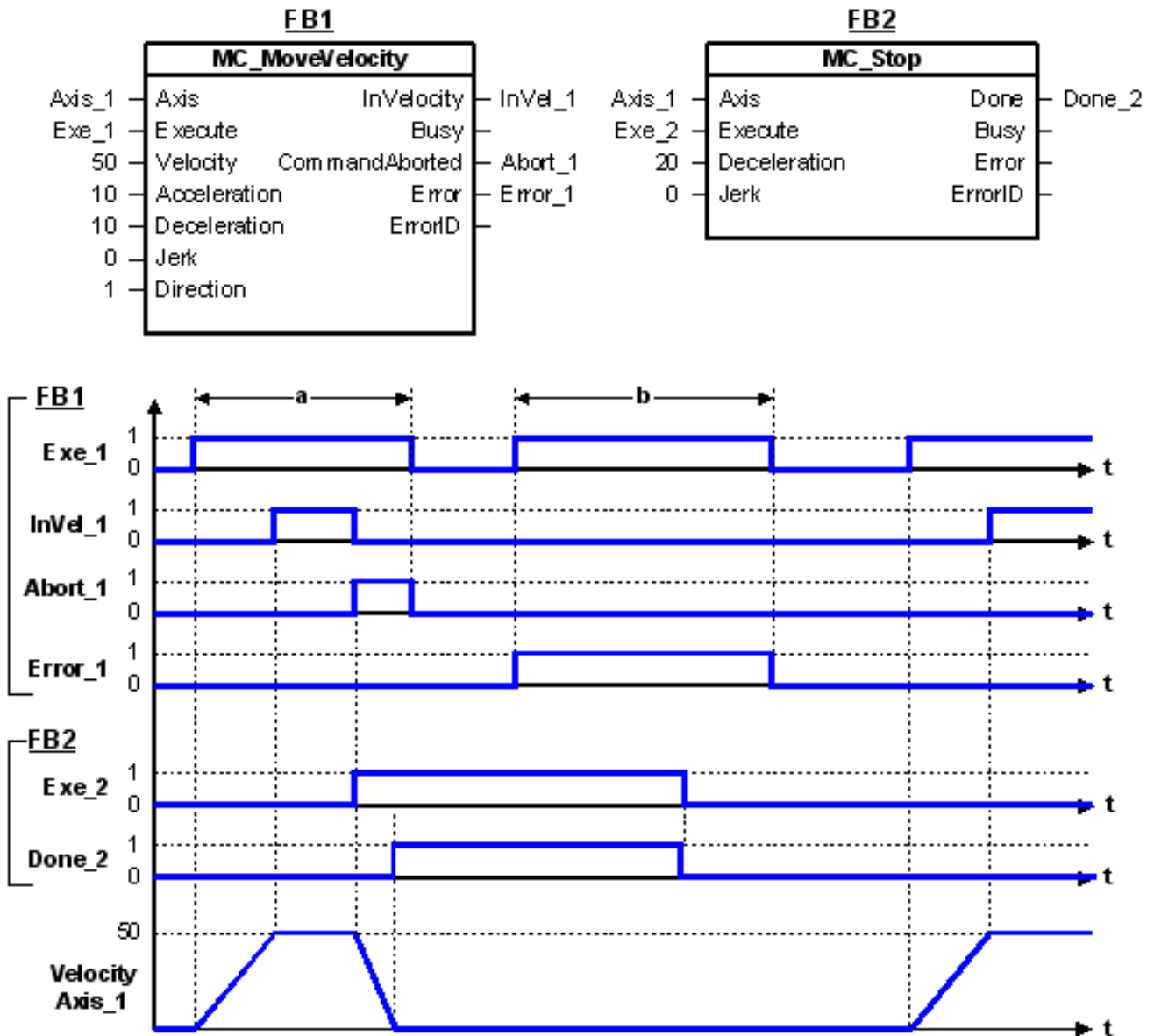
**⊙ Functional Description**

The movement stops at the rising edge of Execute, and the control axis decelerates from the current speed to "0". When this instruction is running, it cannot be terminated by any other instruction. When the MC\_Stop instruction is activated, the running instruction executes CommandAborted.

**MC\_Stop Demo**

The following figure shows how the FB2 (MC\_Stop) instance is used when combined with the FB1 (MC\_MoveVelocity) instance.

- 1) As can be seen from the timing diagram in the figure, when FB2 is activated during the motion of FB1, the speed of the axis will tend to tilt downwards until it is 0.
- 2) As long as the Execute signal of FB2 is TRUE, the corresponding axis will not execute any motion command.
- 3) After FB2 is started, the Error signal of FB1 is set to TRUE, just that the current axis is in stop state.



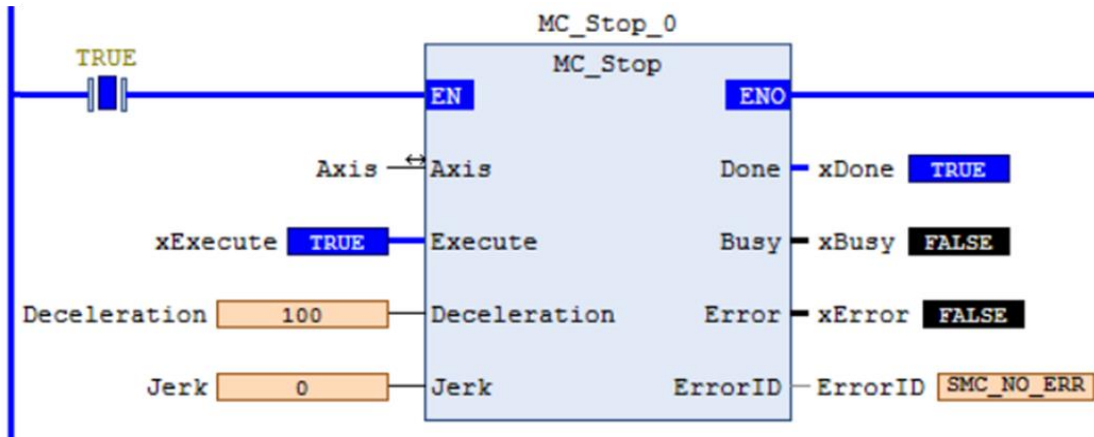
**⊙ Program demo**

**ST:** When xExecute becomes TRUE, the axis is decelerated to a stop.

```

MC_Stop_0(
  Axis:= Axis,
  Execute TRUE := xExecute TRUE,
  Deceleration 100 := Deceleration 100,
  Jerk 0 := Jerk 0,
  Done TRUE => xDone TRUE,
  Busy FALSE => xBusy FALSE,
  Error FALSE => xError FALSE,
  ErrorID SMC_NO_ERR => ErrorID SMC_NO_ERR );
  
```

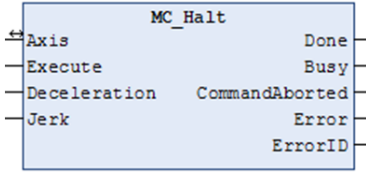
**LD:** When xExecute becomes TRUE, the constant velocity motion instruction is executed.



### 4.2.7 MC\_Halt

Deceleration stops the motion being executed by the axis, and the stopped motion resumes execution of the unfinished portion.

#### ⊙ Command Format

Instruction	Name	FB/FC	LD	ST	File
MC_Halt	Halt instruction	FC		MC_Halt( Axis:=, Execute:=, Deceleration:=, Jerk:=, Done=>, Busy=>, CommandAborted=>, Error=>, ErrorID=>);	SM3_Basic

#### ⊙ Related Variables

##### I/O variable

I/O variable	Name	Data type	Range	initialization	Descriptive
Axis	Axis	AXIS REF SM3	-	-	Specified axis

##### Input variable

Input variable	Name	Data type	Range	initialization	Descriptive
Execute	Start	BOOL	TRUE-FALSE	FALSE	Rising edge trigger
Deceleration	Target Deceleration	LREAL	Is always positive.	0	Deceleration in [u/s <sup>2</sup> ]. Decreasing motor energy
Jack	Target jack	LREAL	Is always positive.	0	Maximum magnitude of jerk in [u/s <sup>3</sup> ]

##### Output variable

Output variable	Name	Data type	Range	initialization	Descriptive
Done	Completion	BOOL	TRUE-FALSE	FALSE	TRUE: Homing is done.
Busy	Executing	BOOL	TRUE-FALSE	FALSE	TRUE: In progress
bCommandAborted	Instruction interrupted	BOOL	TRUE-FALSE	FALSE	TRUE: This command has been terminated by other commands
Error	Error	BOOL	TRUE-FALSE	FALSE	TRUE: An error occurred within the function block
ErrorID	Error Code	SMC_ERROR	-	0	Error indication, see SMC_Error.

	Boolean	Bit string					Integer							Real number		Moment, Duration, Date, String				
	BOOL	BYTE	WORD	DWORD	LWORD	USINT	UINT	UDINT	ULINT	SINT	INT	DINT	LINT	REAL	LREAL	TIME	DATE	TOD	DT	STRING
Axis																				
Execute	√	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
Deceleration	-	-	-	-	-	-	-	-	-	-	-	-	-	√	-	-	-	-	-	-
Jerk	-	-	-	-	-	-	-	-	-	-	-	-	-	√	-	-	-	-	-	-

Done	√	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
Busy	√	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
bCommandAborted	√	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
Error	√	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
ErrorID		SMC_ERROR																		

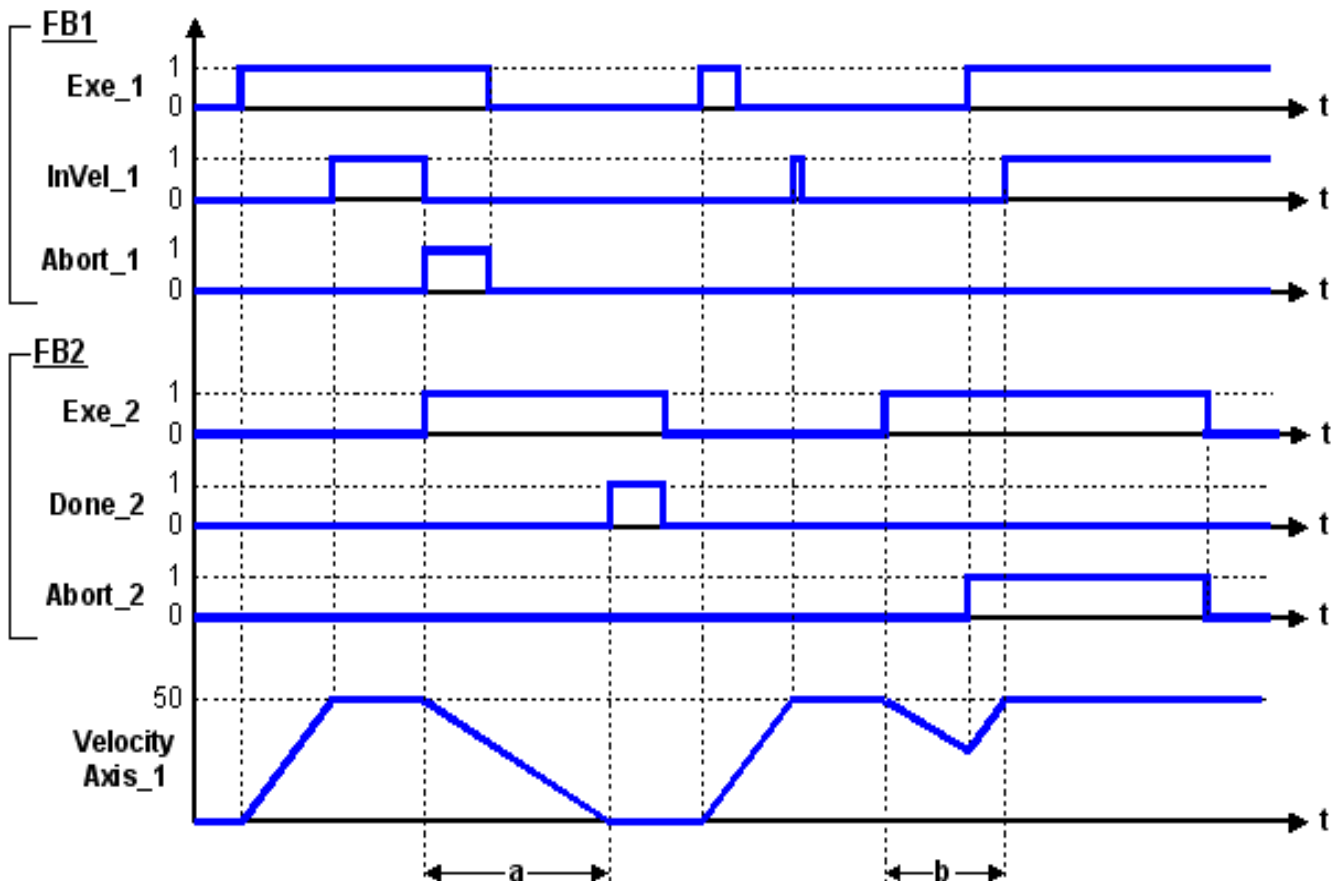
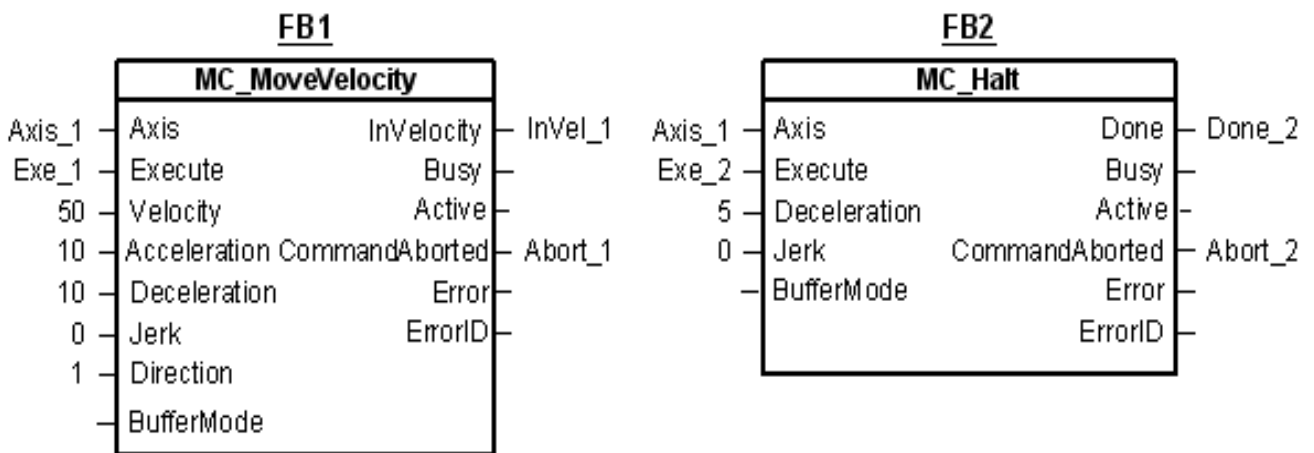
**Ⓢ Functional Description**

For the set axis, motion is paused starting at the rising edge of Execute.  
 Execution of the original motion can be resumed by re-executing the original motion instruction.

**MC\_Halt Demo**

The following diagram shows the behaviour of the MC\_MoveVelocity instruction being interrupted by the MC\_Halt instruction while it is executing motion.

- 1) Compared to MC\_Stop, MC\_Halt only pauses the motion, and the motion can continue to be executed until the instruction is completed.
- 2) After being paused by the MC\_Halt instruction, the axis can be accelerated again by re-calling the motion instruction when the velocity is not zero.



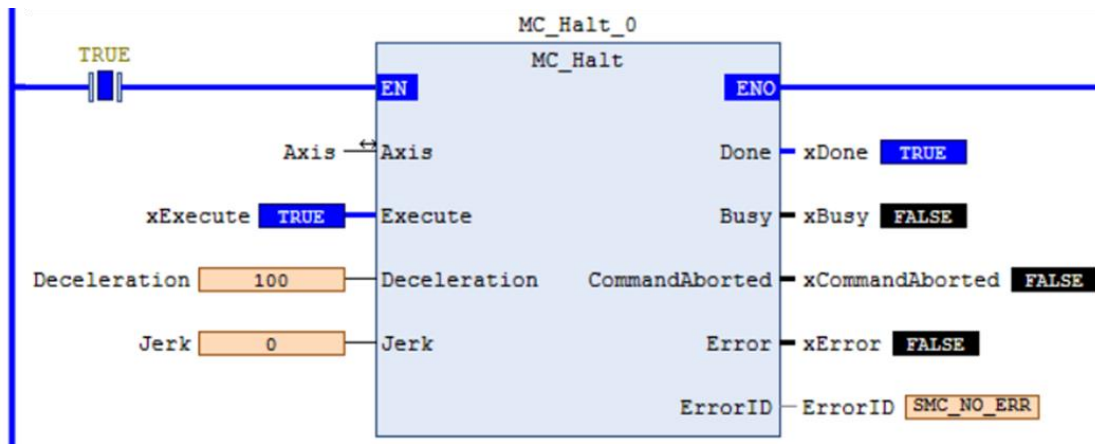
⊙ **Program demo**

**ST:** When xExecute becomes TRUE, a deceleration pause command is executed on the axis.

```

MC_Halt_0(
  Axis:= Axis,
  Execute TRUE := xExecute TRUE ,
  Deceleration 100 := Deceleration 100 ,
  Jerk 0 := Jerk 0 ,
  Done TRUE => xDone TRUE ,
  Busy FALSE => xBusy FALSE ,
  CommandAborted FALSE => xCommandAborted FALSE ,
  Error FALSE => xError FALSE ,
  ErrorID SMC_NO_ERR => ErrorID SMC_NO_ERR );
  
```

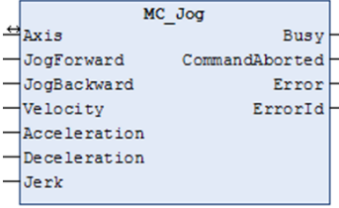
**LD:** When xExecute becomes TRUE, a deceleration pause command is executed on the axis.



### 4.2.8 MC\_Jog

Manual control of axis movement in a specified direction.

#### ⊙ Command Format

Instruction	Name	FB/FC	LD	ST	File
MC_Jog	Jog instruction	FC		MC_Jog( Axis:=, JogForward:=, JogBackward:=, Velocity:=, Acceleration:=, Deceleration:=, Jerk:=, Busy=>, CommandAborted=>, Error=>, ErrorId=> );	SM3_Basic

#### ⊙ Related Variables

##### I/O variable

I/O variable	Name	Data type	Range	initialization	Descriptive
Axis	Axis	AXIS REF SM3	-	-	Specified axis

##### Input variable

Input variable	Name	Data type	Range	initialization	Descriptive
JogForward	Tap to advance	BOOL	TRUE-FALSE	FALSE	TRUE: the axis moves in the positive direction according to the given parameters
JogBackward	Tap to move back	BOOL	TRUE-FALSE	FALSE	TRUE: the axis moves in the negative direction according to the given parameters
Velocity	Target Velocity	LREAL	Is always positive.	0	Maximum velocity in technical units per second [u/s]. Is not necessarily reached.
Acceleration	Target acceleration	LREAL	Is always positive.	0	Acceleration in [u/s <sup>2</sup> ]. Increasing motor energy
Deceleration	Target Deceleration	LREAL	Is always positive.	0	Deceleration in [u/s <sup>2</sup> ]. Decreasing motor energy
Jerk	Target jaek	LREAL	Is always positive.	0	Maximum magnitude of jerk in [u/s <sup>3</sup> ]

##### Output variable

Output variable	Name	Data type	Range	initialization	Descriptive
Busy	Executing	BOOL	TRUE-FALSE	FALSE	TRUE: In progress
Command Aborted	Command Interrupt	BOOL	TRUE-FALSE	FALSE	TRUE: Terminated by other commands
Error	Error	BOOL	TRUE-FALSE	FALSE	TRUE: An error occurred within the function block
ErrorID	Error Code	SMC_ERRO R	-	0	Error indication, see SMC_Error.

	Boolean	Bit string					Integer							Real number		Moment, Duration, Date, String				
	BOOL	BYTE	WORD	DWORD	LWORD	USINT	UINT	UDINT	ULINT	SINT	INT	DINT	LINT	REAL	LREAL	TIME	DATE	TOD	DT	STRING
Axis		AXIS_REF_SM3																		
JogForward	√	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
JogBackward	√	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
Velocity	-	-	-	-	-	-	-	-	-	-	-	-	-	√	-	-	-	-	-	-
Acceleration	-	-	-	-	-	-	-	-	-	-	-	-	-	√	-	-	-	-	-	-
Deceleration	-	-	-	-	-	-	-	-	-	-	-	-	-	√	-	-	-	-	-	-
Jerk	-	-	-	-	-	-	-	-	-	-	-	-	-	√	-	-	-	-	-	-
Busy	√	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
Command Aboned	√	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
Error	√	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
ErrorID		SMC_ERROR																		

### ⊙ Functional Description

When "JogForward" or "JogBackward" is TRUE, the specified axis performs constant velocity motion in the positive or negative direction, respectively.

If "JogForward" or "JogBackward" is TRUE, the specified axis performs constant velocity motion in the positive or negative direction, respectively.

If the command speed setting value of the MC\_Jog instruction exceeds the maximum speed of the point motion in the axis parameter, the motion is executed at the maximum speed of the point motion.

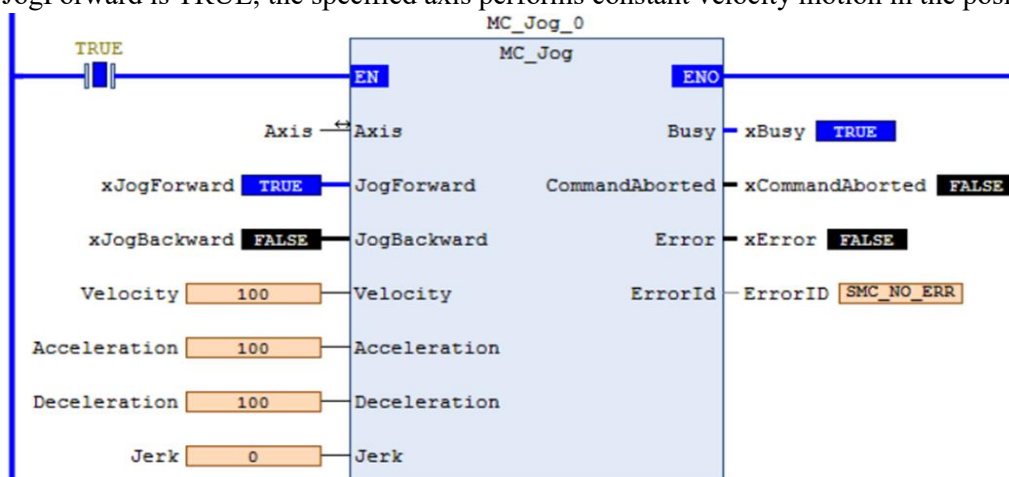
### ⊙ Program demo

**ST:** When JogForward is TRUE, the specified axis performs constant velocity motion in the positive direction.

```

MC_Jog_0(
  Axis:= Axis,
  JogForward TRUE := xJogForward TRUE,
  JogBackward FALSE := xJogBackward FALSE,
  Velocity 100 := Velocity 100,
  Acceleration 100 := Acceleration 100,
  Deceleration 100 := Deceleration 100,
  Jerk 0 := Jerk 0,
  Busy TRUE => xBusy TRUE,
  CommandAborted FALSE => xCommandAborted FALSE,
  Error FALSE => xError FALSE,
  ErrorId SMC_NO_ERR => ErrorId SMC_NO_ERR);
  
```

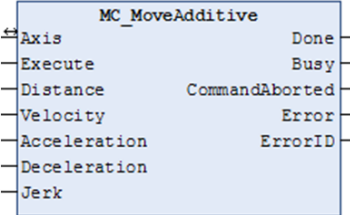
**LD:** When JogForward is TRUE, the specified axis performs constant velocity motion in the positive direction.



### 4.2.9 MC\_MoveAdditive

Control of end-actuators in positional superposition during movement.

#### ⊙ Command Format

Instruction	Name	FB/FC	LD	ST	File
MC_MoveAdditive	Additive motion instruction	FC		MC_MoveAdditive( Axis:=, Execute:=, Distance:=, Velocity:=, Acceleration:=, Deceleration:=, Jerk:=, Done=>, Busy=>, CommandAborted=>, Error=>, ErrorID=>);	SM3_Basic

#### ⊙ Related Variables

##### I/O variable

I/O variable	Name	Data type	Range	initialization	Descriptive
Axis	Axis	AXIS_REF_SM3	-	-	Specified axis

##### Input variable

Input variable	Name	Data type	Range	initialization	Descriptive
Execute	Start	BOOL	TRUE-FALSE	FALSE	Rising edge trigger
Distance	Distance	LREAL	ALL	0	Relative distance of movement
Velocity	Target velocity	LREAL	Is always positive.	0	Maximum velocity in technical units per second [u/s]. Is not necessarily reached.
Acceleration	Target acceleration	LREAL	Is always positive.	0	Acceleration in [u/s <sup>2</sup> ]. Increasing motor energy
Deceleration	Target Deceleration	LREAL	Is always positive.	0	Deceleration in [u/s <sup>2</sup> ]. Decreasing motor energy
Jack	Target jack	LREAL	Is always positive.	0	Maximum magnitude of jerk in [u/s <sup>3</sup> ]

##### Output variable

Output variable	Name	Data type	Range	initialization	Descriptive
Done	Completion	BOOL	TRUE-FALSE	FALSE	TRUE: Homing is done.
Busy	Executing	BOOL	TRUE-FALSE	FALSE	TRUE: In progress
Command Aborted	Command Interrupt	BOOL	TRUE-FALSE	FALSE	TRUE: Terminated by other commands
Error	Error	BOOL	TRUE-FALSE	FALSE	TRUE: An error occurred within the function block
ErrorID	Error Code	SMC_ERRO R	-	0	Error indication, see SMC_Error.

	Boolean	Bit string					Integer							Real number		Moment, Duration, Date, String				
	BOOL	BYTE	WORD	DWORD	LWORD	USINT	UINT	UDINT	ULINT	SINT	INT	DINT	LINT	REAL	LREAL	TIME	DATE	TOD	DT	STRING
Axis		AXIS_REF_SM3																		
Execute	√	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
Distance	-	-	-	-	-	-	-	-	-	-	-	-	-	√	-	-	-	-	-	-
Velocity	-	-	-	-	-	-	-	-	-	-	-	-	-	√	-	-	-	-	-	-
Acceleration	-	-	-	-	-	-	-	-	-	-	-	-	-	√	-	-	-	-	-	-
Deceleration	-	-	-	-	-	-	-	-	-	-	-	-	-	√	-	-	-	-	-	-
Jerk	-	-	-	-	-	-	-	-	-	-	-	-	-	√	-	-	-	-	-	-
Done	√	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
Busy	√	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
Command Aboned	√	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
Error	√	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
ErrorID		SMC_ERROR																		

⊙ **Functional Description**

Controls the terminal actuator to move an additional distance at a given speed and acceleration.

The final position of the end-actuator is the sum of the distance given by the previous displacement command and this command.

When interrupting the movement of other instructions, the speed of the axis is the speed of this function block.

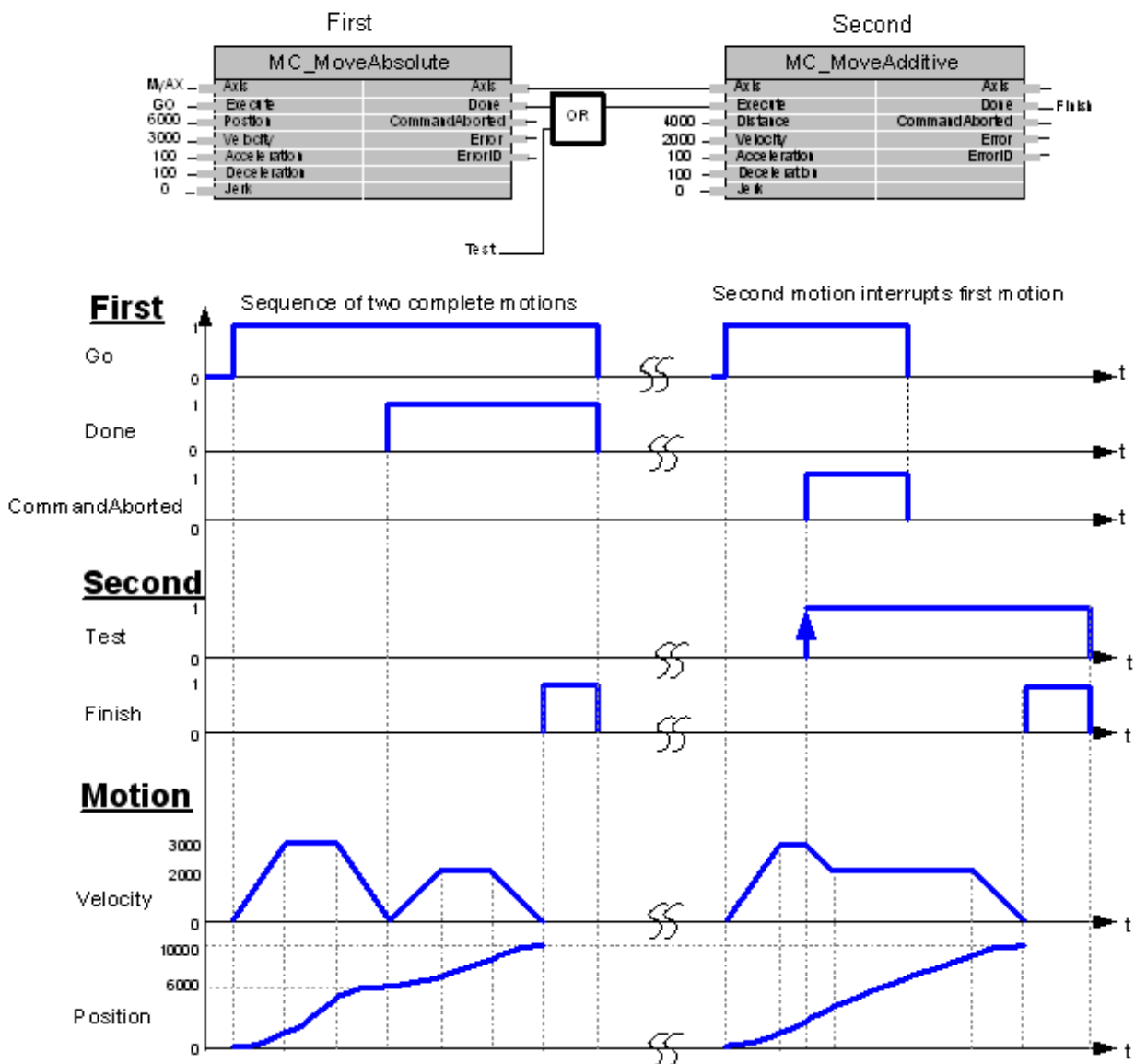
When the previous instruction is a speed instruction, the execution of this instruction terminates the speed instruction and stops after moving the given distance at the given speed with acceleration and deceleration.

**MC\_MoveAdditive Demo**

1) The figure below shows the movement of the first instance "First" (MC\_MoveAbsolute) combined with the second instance "Second" (MC\_MoveAdditive).

2) In the first part of the diagram, the Second instance is called after the First instance. When the First instance moves to the specified position 6000, the velocity is 0 and the Done signal is True, and the execution of the Second instance starts to move the axis to position 10000.

3) At the end of the diagram, during the execution of the First instance, the Second instance is started. The movement of the First instance is now at position 4000, at which point the movement of the First instance is interrupted and it starts to decelerate, and when it reaches position 6000, it moves the axis to the final position 10,000 at the speed specified by the Second instance.

**MoveAdditive - Example**


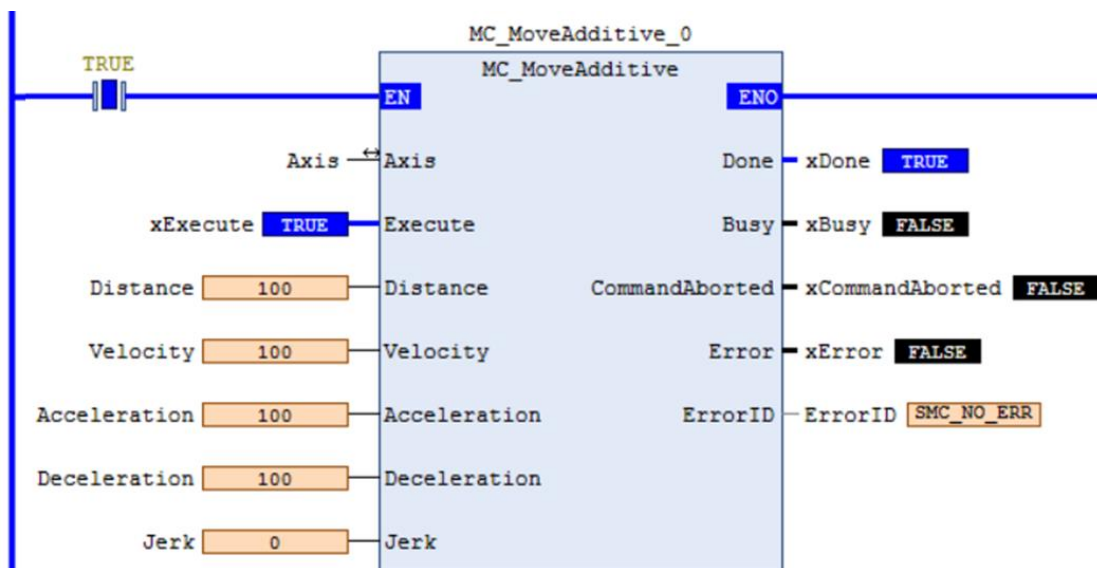
**Ⓞ Program demo**

**ST:** When xExecute becomes TRUE, the position overlay motion instruction is executed

```

MC_MoveAdditive_0(
  Axis:= Axis,
  Execute TRUE := xExecute TRUE,
  Distance 100 := Distance 100,
  Velocity 100 := Velocity 100,
  Acceleration 100 := Acceleration 100,
  Deceleration 100 := Deceleration 100,
  Jerk 0 := Jerk 0,
  Done FALSE => xDone FALSE,
  Busy TRUE => xBusy TRUE,
  CommandAborted FALSE => xCommandAborted FALSE,
  Error FALSE => xError FALSE,
  ErrorID SMC_NO_ERR => ErrorID SMC_NO_ERR);
  
```

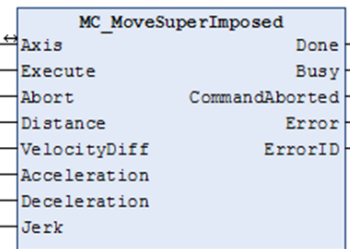
**LD:** When xExecute becomes TRUE, the constant velocity motion instruction is executed.



### 4.2.10 MC\_MoveSuperImposed

Control of end-actuators with superimposed motion during movement.

#### ☉ Command Format

Instruction	Name	FB/FC	LD	ST	File
MC_MoveSuperImposed	Super Imposed motion instruction	FC		MC_MoveSuperImposed( Axis:=, Execute:=, Abort:=, Distance:=, VelocityDiff:=, Acceleration:=, Deceleration:=, Jerk:=, Done=>, Busy=>, CommandAborted=>, Error=>, ErrorID=>);	SM3_Basic

#### ☉ Related Variables

##### I/O variable

I/O variable	Name	Data type	Range	initialization	Descriptive
Axis	Axis	AXIS REF SM3	-	-	Specified axis

##### Input variable

Input variable	Name	Data type	Range	initialization	Descriptive
Execute	Start	BOOL	TRUE-FALSE	FALSE	Rising edge trigger
Abort	Terminate (law)	BOOL	TRUE-FALSE	FALSE	Motion terminated, output parameters reset
Distance	Distance	LREAL	ALL	0	Relative distance of movement
VelocityDiff	Stacked acceleration	LREAL	Is always positive.	0	Stacked acceleration.
Acceleration	Target acceleration	LREAL	Is always positive.	0	Acceleration in [u/s <sup>2</sup> ]. Increasing motor energy
Deceleration	Target Deceleration	LREAL	Is always positive.	0	Deceleration in [u/s <sup>2</sup> ]. Decreasing motor energy
Jack	Target jaek	LREAL	Is always positive.	0	Maximum magnitude of jerk in [u/s <sup>3</sup> ]

##### Output variable

Output variable	Name	Data type	Range	initialization	Descriptive
Done	Completion	BOOL	TRUE-FALSE	FALSE	TRUE: Homing is done.
Busy	Executing	BOOL	TRUE-FALSE	FALSE	TRUE: In progress
Command Aborted	Command Interrupt	BOOL	TRUE-FALSE	FALSE	TRUE: Terminated by other commands
Error	Error	BOOL	TRUE-FALSE	FALSE	TRUE: An error occurred within the function block
ErrorID	Error Code	SMC_ERROR	-	0	Error indication, see SMC_Error.

	Boolean	Bit string					Integer							Real number		Moment, Duration, Date, String				
	BOOL	BYTE	WORD	DWORD	LWORD	USINT	UINT	UDINT	ULINT	SINT	INT	DINT	LINT	REAL	LREAL	TIME	DATE	TOD	DT	STRING
Axis		AXIS_REF_SM3																		
Execute	√	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
Abort	√	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
Distance	-	-	-	-	-	-	-	-	-	-	-	-	-	-	√	-	-	-	-	-
VelocityDiff	-	-	-	-	-	-	-	-	-	-	-	-	-	-	√	-	-	-	-	-
Acceleration	-	-	-	-	-	-	-	-	-	-	-	-	-	-	√	-	-	-	-	-
Deceleration	-	-	-	-	-	-	-	-	-	-	-	-	-	-	√	-	-	-	-	-
Jerk	-	-	-	-	-	-	-	-	-	-	-	-	-	-	√	-	-	-	-	-
Done	√	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
Busy	√	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
Command Aboned	√	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
Error	√	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
ErrorID		SMC_ERROR																		

© **Functional Description**

This instruction is used to control the terminal actuator to move a superimposed distance at the given speed and acceleration.

The final position of the terminal actuator is the sum of the distance given by the previous displacement instruction and this instruction.

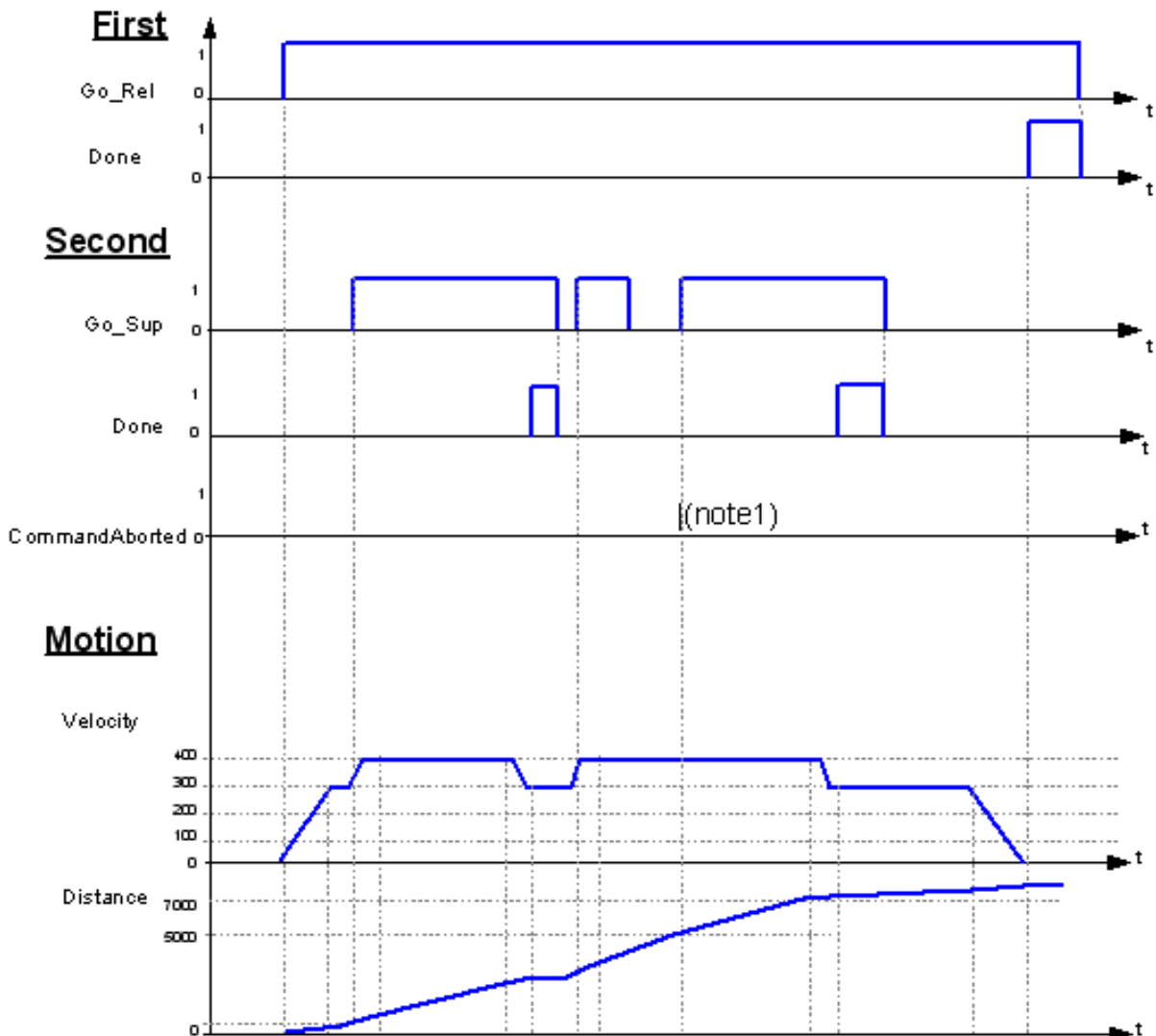
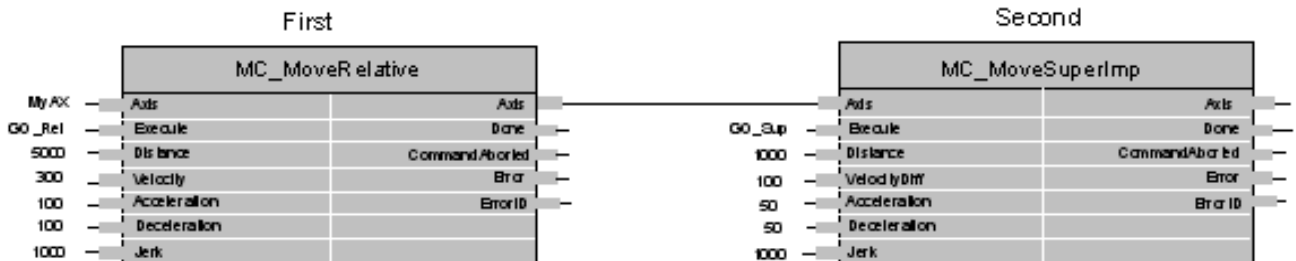
When the first movement is not completed, the speed is the sum of two movements; when the first movement is completed, the speed is the speed of this function block.

When the previous instruction is a velocity instruction, the execution of this instruction terminates the velocity instruction and stops after moving the given distance at the given speed, plus or minus.

**MC\_MoveSuperImposed Demo**

- 1) The following figure shows an example of connecting the instance of MC\_MoveRelative function block "First" and the instance of MC\_MoveSuperImposed function block "Second":
- 2) You can see that the curve of the CommandAborted signal is always zero, because the "Second" instance is acting on the same instance as the "First" command.
- 3) The final position of the end of motion is between 7000 and 8000, depending on the actual start/stop time of the "Second" instance.

### MoveSuperimposed - Example

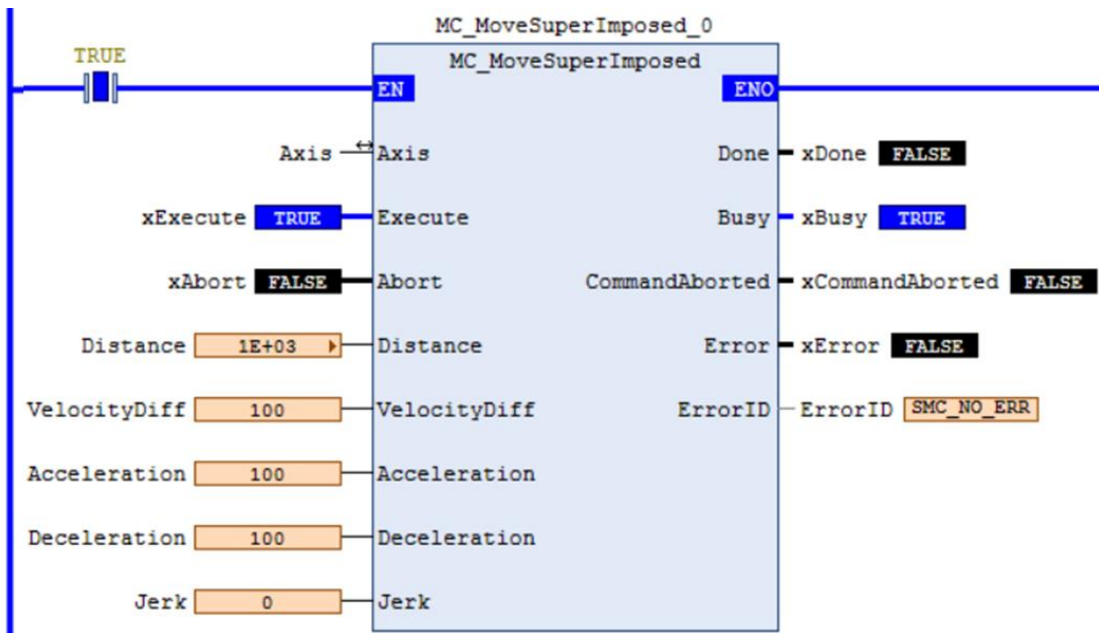


**⊙ Program demo**

**ST:** When xExecute becomes TRUE, the position velocity superimposed motion instruction is executed.

```
MC_MoveSuperImposed_0 (
  Axis:= Axis,
  Execute TRUE := xExecute TRUE,
  Abort FALSE := xAbort FALSE,
  Distance  := Distance ,
  VelocityDiff  := VelocityDiff ,
  Acceleration  := Acceleration ,
  Deceleration  := Deceleration ,
  Jerk  := Jerk ,
  Done FALSE => xDone FALSE,
  Busy TRUE => xBusy TRUE,
  CommandAborted FALSE => xCommandAborted FALSE,
  Error FALSE => xError FALSE,
  ErrorID  => ErrorID  );
```

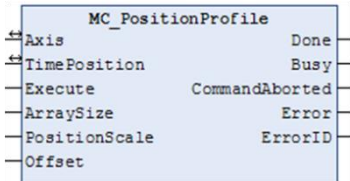
**LD:** When xExecute becomes TRUE, the position velocity superimposed motion instruction is executed.



### 4.2.11 MC\_PositionProfile

The user can plan his own "time-position" data table and the controller will complete the movement according to the planned data.

#### ⊙ Command Format

Instruction	Name	FB/FC	LD	ST	File
MC_PositionProfile	Position Profile instruction	FC		MC_PositionProfile( Axis:=, TimePosition:=, Execute:=, ArraySize:=, PositionScale:=, Ffset:=, Done=>, Busy=>, CommandAborted=>, Error=>, ErrorID=>);	SM3_Basic

#### ⊙ Related Variables

##### I/O variable

I/O variable	Name	Data type	Range	initialization	Descriptive
Axis	Axis	AXIS REF SM3	-	-	Specified axis
TimePosition	Data sheet	MC TP REF	-	-	Planned time-location data table

##### Input variable

Input variable	Name	Data type	Range	initialization	Descriptive
Execute	Start	BOOL	TRUE-FALSE	FALSE	Rising edge trigger
ArraySize	Number of data points	INT	Is always positive.	0	Data table size
PositionScale	Scale	LREAL	ALL	1	Scale factor for the entire data table location
Offset	Offset	LREAL	ALL	0	position bias

##### Output variable

Output variable	Name	Data type	Range	initialization	Descriptive
Done	Completion	BOOL	TRUE-FALSE	FALSE	TRUE: Homing is done.
Busy	Executing	BOOL	TRUE-FALSE	FALSE	TRUE: In progress
Command Aborted	Command Interrupt	BOOL	TRUE-FALSE	FALSE	TRUE: Terminated by other commands
Error	Error	BOOL	TRUE-FALSE	FALSE	TRUE: An error occurred within the function block
ErrorID	Error Code	SMC_ERROR	-	0	Error indication, see SMC Error.

	Boolean	Bit string					Integer					Real number		Moment, Duration, Date, String						
	BOOL	BYTE	WORD	DWORD	LWORD	USINT	UINT	UDINT	ULINT	SINT	INT	DINT	LINT	REAL	LREAL	TIME	DATE	TOD	DT	STRING
Axis		AXIS_REF_SM3																		
TimePosition		MC_TP_REF																		

Execute	√	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
ArraySize	-	-	-	-	-	-	-	-	-	-	√	-	-	-	-	-	-	-	-	-
PositionScale	-	-	-	-	-	-	-	-	-	-	-	-	-	-	√	-	-	-	-	-
Offset	-	-	-	-	-	-	-	-	-	-	-	-	-	-	√	-	-	-	-	-
Done	√	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
Busy	√	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
Command Aboned	√	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
Error	√	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
ErrorID	SMC_ERROR																			

**⊙ Functional Description**

The user can plan his own "time-position" data table and the controller will complete the motion according to the planned data.

PVT motion refers to planning a motion by defining the position, velocity, acceleration and other parameters reached by the point axis at each moment.

There are three kinds of PVT motions supported by the controller, as shown in the table below.

**MC\_MoveSuperImposed Demo**

- 1) The following figure shows an example of connecting the instance of MC\_MoveRelative function block "First" and the instance of MC\_MoveSuperImposed function block "Second":
- 2) You can see that the curve of the CommandAborted signal is always zero, because the "Second" instance is acting on the same instance as the "First" command.
- 3) The final position of the end of motion is between 7000 and 8000, depending on the actual start/stop time of the "Second" instance.

Command Name	Function
MC_PositionProfile	Planning of axis movements by defining time-position tables
MC_VelocityProfile	Planning the motion of an axis by defining a time-velocity table
MC_AccelerationProfile	Planning the motion of an axis by defining a time-acceleration table

The data structures used for the MC\_PositionProfile motion are defined below:

- 1) Structure of the "Time-Position" data table:

```

TYPE MC_TP_TABLE
STRUCT
Number_of_pairs : INT; //number of position-time pairs
IsAbsolute: BOOL; //Position value absolute relative selection
MC_TP_Array : ARRAY [1..N] of MC_TP; //position-time data
END_STRUCT
END_TYPE

```

- 2) Structure of "time-position" data:

```

TYPE MC_TP
STRUCT
delta_time: TIME; //time of position-time data
position: REAL; //position of position-time data (absolute value or relative value).
END_STRUCT
END_TYPE

```

**MC\_PositionProfile Demo**

- 1) MC\_TP\_REF supports a special data type. The code in the following figure gives an example of this data structure.
- 2) A time/position content can be expressed as DeltaTime/Pos, where DeltaTime is the time difference between two consecutive points:

```

//PT运动参数的赋值
arPT.IsAbsolute:=FALSE ; //相对位置
arPT.MC_IP_Array:=arNUM ; //PT点的数据
arPT.Number_of_pairs:=4 ; //PT点的个数

//PT点的数据赋值
arNUM[1].delta_time:=T#1S; //第一个PT点的时间数据
arNUM[1].position:=10000; //第一个PT点的位置数据
arNUM[2].delta_time:=T#2S;
arNUM[2].position:=40000;
arNUM[3].delta_time:=T#1S;
arNUM[3].position:=-10000;
arNUM[4].delta_time:=T#0.5S;
arNUM[4].position:=-5000;
    
```

This example implements the PT motion of AXIS\_0 in 4 phases with continuous motion between phases:

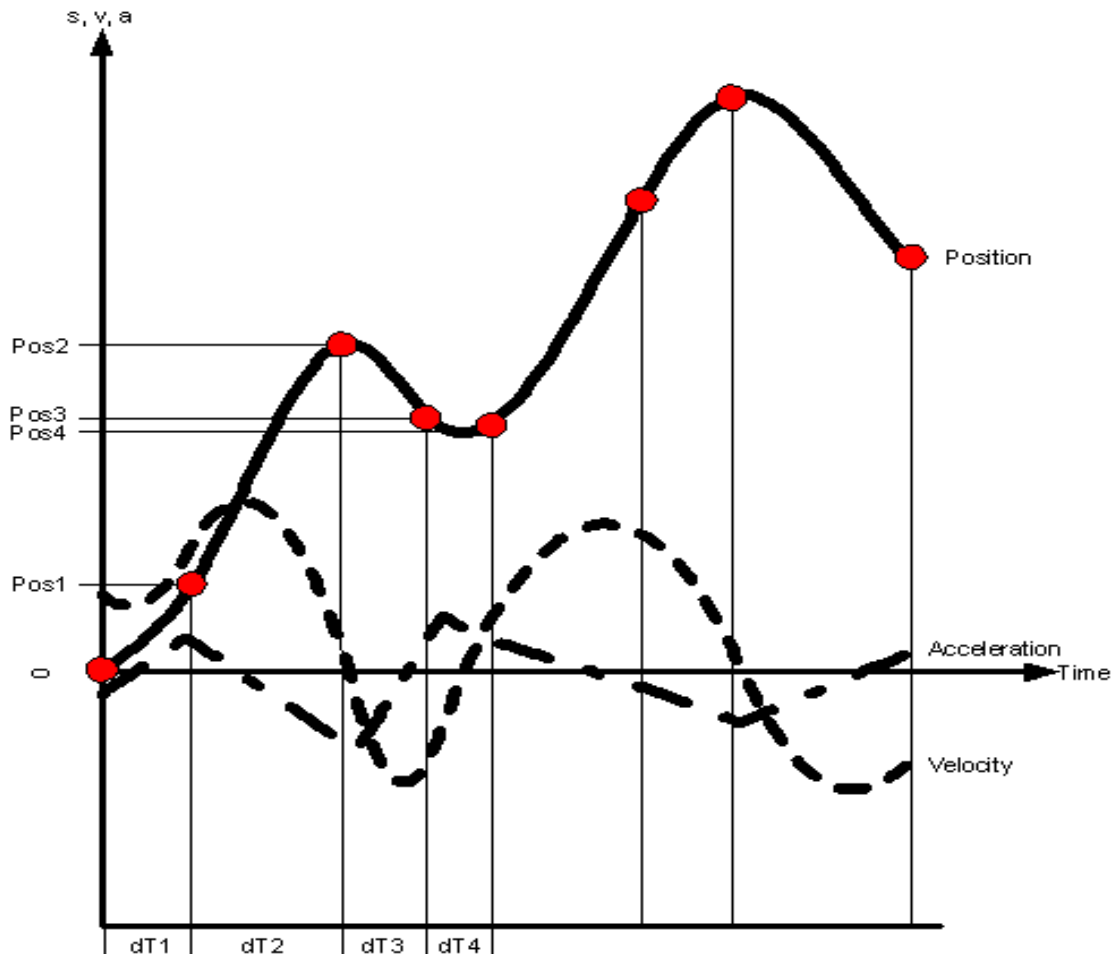
Phase 1: 1 second, motion 10000, and

Phase 2: 2 seconds, motion 40,000.

Third stage: 1 second, motion -10000, and

Fourth stage: 0.5 seconds, motion -5000.

The result of the routine "PT" is shown in the figure below.



deltaTime	absPos
dT1	Pos1
dT2	Pos2
dT3	Pos3
dT4	Pos4

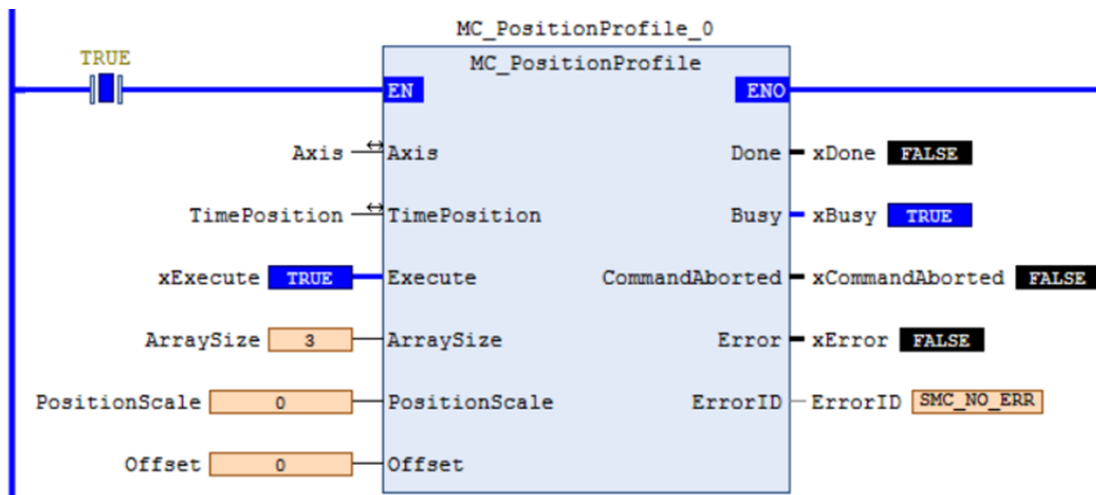
**⊙ Program demo**

**ST:** When xExecute becomes TRUE, the time-position planning motion instruction is executed.

```

MC_PositionProfile_0(
  Axis:=Axis ,
  TimePosition:= TimePosition,
  Execute TRUE := xExecute TRUE ,
  ArraySize 3 := ArraySize 3 ,
  PositionScale 0 := PositionScale 0 ,
  Offset 0 := Offset 0 ,
  Done FALSE => xDone FALSE ,
  Busy TRUE => xBusy TRUE ,
  CommandAborted FALSE => xCommandAborted FALSE ,
  Error FALSE => xError FALSE ,
  ErrorID SMC_NO_ERR => ErrorID SMC_NO_ERR );
  
```

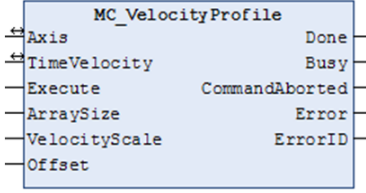
**LD:** When xExecute becomes TRUE, the time-position planning motion instruction is executed.



### 4.2.12 MC\_VelocityProfile

This instruction is similar to the MC\_PositionProfile instruction, MC\_VelocityProfile plans the motion by defining the "time-velocity" data.

#### ⊙ Command Format

Instruction	Name	FB/FC	LD	ST	File
MC_VelocityProfile	Velocity Profile instruction	FC		MC_VelocityProfile( Axis:=, TimeVelocity:=, Execute:=, ArraySize:=, VelocityScale:=, Offset:=, Done=>, Busy=>, CommandAborted=>, Error=>, ErrorID=>);	SM3_Basic

#### ⊙ Related Variables

##### I/O variable

I/O variable	Name	Data type	Range	initialization	Descriptive
Axis	Axis	AXIS REF SM3	-	-	Specified axis
TimePosition	Data sheet	MC TV REF	-	-	Planned time-location data table

##### Input variable

Input variable	Name	Data type	Range	initialization	Descriptive
Execute	Start	BOOL	TRUE-FALSE	FALSE	Rising edge trigger
ArraySize	Number of data points	INT	Is always positive.	0	Data table size
VelocityScale	proportions	LREAL	ALL	1	Scale factor
Offset	Offset	LREAL	ALL	0	position bias

##### Output variable

Output variable	Name	Data type	Range	initialization	Descriptive
Done	Completion	BOOL	TRUE-FALSE	FALSE	TRUE: Homing is done.
Busy	Executing	BOOL	TRUE-FALSE	FALSE	TRUE: In progress
Command Aborted	Command Interrupt	BOOL	TRUE-FALSE	FALSE	TRUE: Terminated by other commands
Error	Error	BOOL	TRUE-FALSE	FALSE	TRUE: An error occurred within the function block
ErrorID	Error Code	SMC_ERROR	-	0	Error indication, see SMC Error.

	Boolean	Bit string					Integer						Real number		Moment, Duration, Date, String					
	BOOL	BYTE	WORD	DWORD	LWORD	USINT	UINT	UDINT	ULINT	SINT	INT	DINT	LINT	REAL	LREAL	TIME	DATE	TOD	DT	STRING
Axis		AXIS_REF_SM3																		
TimeVelocity		MC_TV_REF																		
Execute	√	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-

ArraySize	-	-	-	-	-	-	-	-	-	-	√	-	-	-	-	-	-	-	-	-
VelocityScale	-	-	-	-	-	-	-	-	-	-	-	-	-	-	√	-	-	-	-	-
Offset	-	-	-	-	-	-	-	-	-	-	-	-	-	-	√	-	-	-	-	-
Done	√	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
Busy	√	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
Command Aboned	√	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
Error	√	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
ErrorID	SMC_ERROR																			

☉ **Functional Description**

The MC\_VelocityProfile function block models the profile motion in time and velocity, and executes the motion according to the data set by the user in the "Time-Velocity" table variable.

The data table used for the VT motion:

1) The structure of the "Time-Speed" data table is as follows:

```

TYPE MC_TV_TABLE
STRUCT
Number_of_pairs : INT; //number of speed-time pairs
IsAbsolute: BOOL; //Speed value absolute relative selection
MC_TV_Array : ARRAY [1..N] of MC_TV; //Speed-time data
END_STRUCT
END_TYPE
  
```

2) Type of "Time-Speed" Data

```

TYPE MC_TV
STRUCT
delta_time : TIME; //Time of velocity-time data.
velocity : REAL; //Velocity (absolute or relative) of the velocity-time data.
END_STRUCT
END_TYPE
  
```

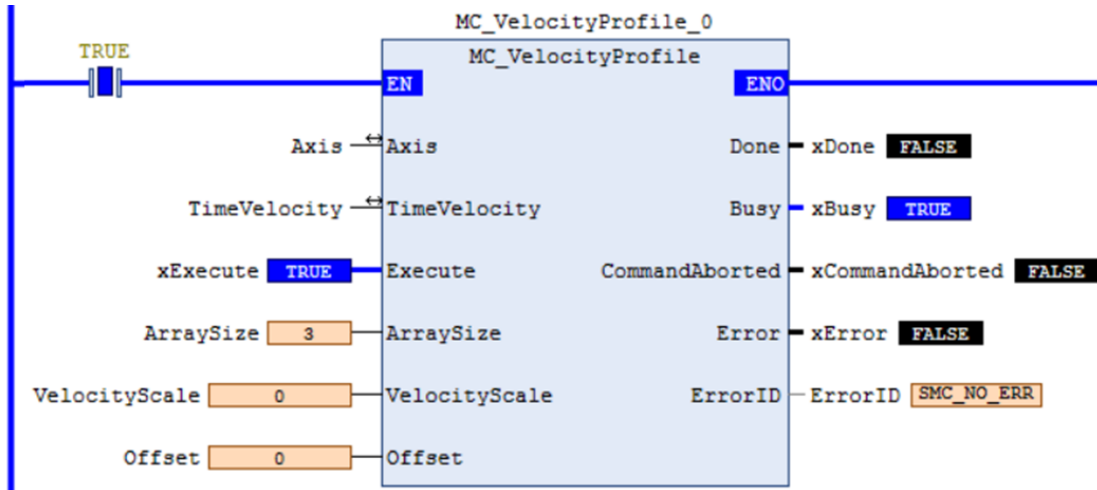
☉ **Program demo**

**ST:** When xExecute becomes TRUE, the time-velocity planning motion instruction is executed.

```

MC_VelocityProfile_0(
  Axis:=Axis ,
  TimeVelocity:= TimeVelocity,
  Execute TRUE := xExecute TRUE,
  ArraySize 3 := ArraySize 3,
  VelocityScale 0 := VelocityScale 0,
  Offset 0 := Offset 0,
  Done FALSE => xDone FALSE,
  Busy TRUE => xBusy TRUE,
  CommandAborted FALSE => xCommandAborted FALSE,
  Error FALSE => xError FALSE,
  ErrorID SMC_NO_ERR => ErrorID SMC_NO_ERR );
  
```

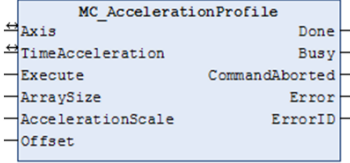
**LD:** When xExecute becomes TRUE, the time-velocity planning motion instruction is executed.



### 4.2.13 MC\_AccelerationProfile

This instruction is similar to the MC\_PositionProfile instruction, MC\_VelocityProfile plans the motion by defining the "time-velocity" data.

#### ⊙ Command Format

Instruction	Name	FB/FC	LD	ST	File
MC_AccelerationProfile	Acceleration Profile instruction	FC		MC_AccelerationProfile ( Axis:=, TimeAcceleration:=, Execute:=, ArraySize:=, AccelerationScale:=, Offset:=, Done=>, Busy=>, CommandAborted=>, Error=>, ErrorID=>);	SM3_Basic

#### ⊙ Related Variables

##### I/O variable

I/O variable	Name	Data type	Range	initialization	Descriptive
Axis	Axis	AXIS_REF_SM3	-	-	Specified axis
TimePosition	Data sheet	MC_TV_REF	-	-	Planned time-location data table

##### Input variable

Input variable	Name	Data type	Range	initialization	Descriptive
Execute	Start	BOOL	TRUE-FALSE	FALSE	Rising edge trigger
ArraySize	Number of data points	INT	Is always positive.	0	Data table size
AccelerationScale	proportions	LREAL	ALL	1	Scale factor
Offset	Offset	LREAL	ALL	0	position bias

##### Output variable

Output variable	Name	Data type	Range	initialization	Descriptive
Done	Completion	BOOL	TRUE-FALSE	FALSE	TRUE: Homing is done.
Busy	Executing	BOOL	TRUE-FALSE	FALSE	TRUE: In progress
Command Aborted	Command Interrupt	BOOL	TRUE-FALSE	FALSE	TRUE: Terminated by other commands
Error	Error	BOOL	TRUE-FALSE	FALSE	TRUE: An error occurred within the function block
ErrorID	Error Code	SMC_ERROR	-	0	Error indication, see SMC_Error.

	Boolean	Bit string					Integer					Real number		Moment, Duration, Date, String						
	BOOL	BYTE	WORD	DWORD	LWORD	USINT	UINT	UDINT	ULINT	SINT	INT	DINT	LINT	REAL	LREAL	TIME	DATE	TOD	DT	STRING
Axis		AXIS_REF_SM3																		
TimeAcceleration		MC_TA_REF																		

Execute	√	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
ArraySize	-	-	-	-	-	-	-	-	-	-	√	-	-	-	-	-	-	-	-	-
AccelerationScale	-	-	-	-	-	-	-	-	-	-	-	-	-	-	√	-	-	-	-	-
Offset	-	-	-	-	-	-	-	-	-	-	-	-	-	-	√	-	-	-	-	-
Done	√	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
Busy	√	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
Command Aboned	√	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
Error	√	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
ErrorID	SMC_ERROR																			

⊙ **Functional Description**

The MC\_AccelerationProfile function block models the profile motion in time and acceleration, and executes the motion according to the data set by the user in the "Time-Acceleration" table variable.

The data table used for the AT motion:

1) The structure of the "Time-Acceleration" data table is as follows:

```

TYPE MC_TA_TABLE
STRUCT
Number_of_pairs : INT; //number of acceleration-time pairs
IsAbsolute : BOOL; //Acceleration value absolute relative selection.
MC_TA_Array : ARRAY [1..N] of MC_TA; //acceleration-time data
END_STRUCT
END_TYPE.

```

2) The type of "time-acceleration" data.

```

TYPE MC_TA
STRUCT
delta_time : TIME; //Time of acceleration/time data.
acceleration: REAL; //acceleration (absolute or relative) of acceleration-time data.
END_STRUCT
END_TYPE.

```

⊙ **Program demo**

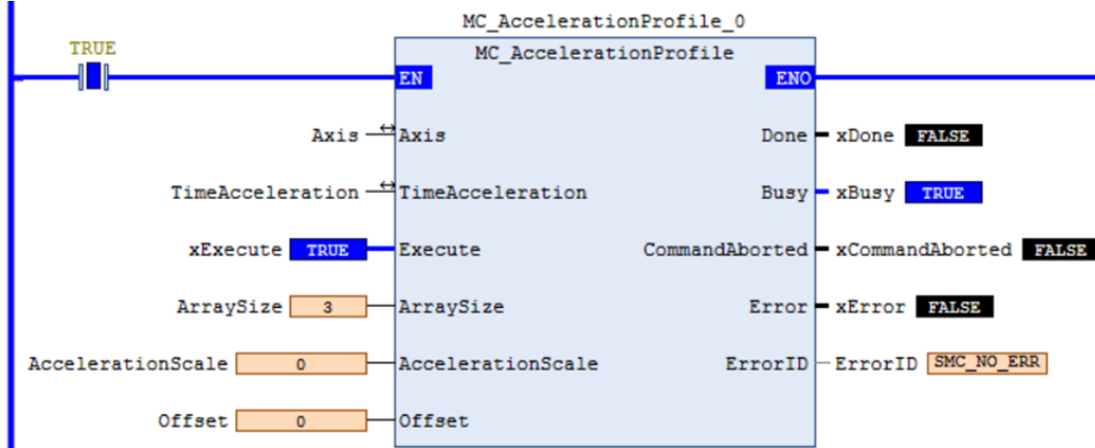
**ST:** When xExecute becomes TRUE, the time-acceleration planning motion instruction is executed.

```

MC_AccelerationProfile_0(
  Axis:=Axis ,
  TimeAcceleration := TimeAcceleration,
  Execute TRUE := xExecute TRUE ,
  ArraySize 3 := ArraySize 3 ,
  AccelerationScale 0 := AccelerationScale 0 ,
  Offset 0 := Offset 0 ,
  Done FALSE => xDone FALSE ,
  Busy TRUE => xBusy TRUE ,
  CommandAborted FALSE => xCommandAborted FALSE ,
  Error FALSE => xError FALSE ,
  ErrorID SMC_NO_ERR => ErrorID SMC_NO_ERR );

```

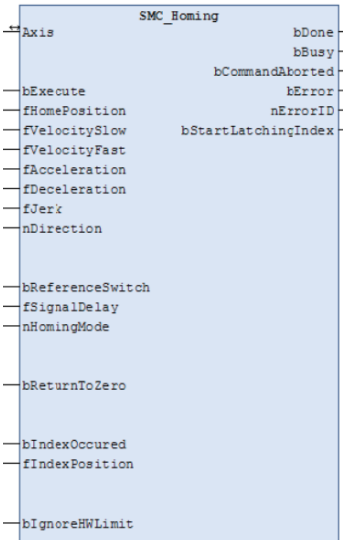
**LD:** When xExecute becomes TRUE, the time-acceleration planning motion instruction is executed.



### 4.2.14 SMC\_Homing

This instruction is similar to the MC\_PositionProfile instruction, MC\_VelocityProfile plans the motion by defining the "time-velocity" data.

#### ⊙ Command Format

Instruction	Name	FB/FC	LD	ST	File
SMC_Homing	OP homing instruction	FC	 <p>The LD diagram shows the SMC_Homing instruction with the following inputs and outputs:</p> <ul style="list-style-type: none"> <li>Inputs: Axis, bExecute, fHomePosition, fVelocitySlow, fVelocityFast, fAcceleration, fDeceleration, fJerk, nDirection, bReferenceSwitch, fSignalDelay, nHomingMode, bReturnToZero, bIndexOccured, fIndexPosition, bIgnoreHWLimit.</li> <li>Outputs: bDone, bBusy, bCommandAborted, bError, nErrorID, bStartLatchingIndex.</li> </ul>	<pre>SMC_Homing( Axis:= , bExecute:=, fHomePosition:=, fVelocitySlow:=, fVelocityFast:=, fAcceleration:=, fDeceleration:=, fJerk:=, nDirection:=, bReferenceSwitch:=, fSignalDelay:=, nHomingMode:=, bReturnToZero:=, bIndexOccured:=, fIndexPosition:=, bIgnoreHWLimit:=, bDone=&gt;, bBusy=&gt; , bCommandAborted=&gt;, bError=&gt; , nErrorID=&gt;, bStartLatchingIndex=&gt;);</pre>	SM3_Basic

#### ⊙ Related Variables

##### I/O variable

I/O variable	Name	Data type	Range	initialization	Descriptive
Axis	Axis	AXIS_REF SM3	-	-	Specified axis

##### Input variable

Input variable	Name	Data type	Range	initialization	Descriptive
bExecute	Execute	BOOL	TRUE-FALSE	FALSE	TRUE:Execute function block FALSE: Function block is not executed.
fHomePosition	Home Setting Position	LREAL	ALL		Home set position after homing in user calibrated units
fVelocitySlow	Slow	LREAL	ALL	0	Slow setting speed after leaving reference switch set
fVelocityFast	Fast	LREAL	ALL	0	Fast setting speed when leaving the reference switch set position
fAcceleration	Acceleration	LREAL	ALL	0	Acceleration set value
fDeceleration	Deceleration	LREAL	ALL	0	Deceleration setting value
fJerk	Acceleration	LREAL	ALL	0	Jerk [u/s <sup>3</sup> ]
nDirection	Zero direction	MC_DIRECTION	ALL	negative	Homing start direction, reference MC_DIRECTION
bReferenceSwitch	Reference switch	BOOL	ALL	FALSE	Connection of reference switch, TRUE: reference switch triggered, FALSE: reference switch closed
fSignalDelay	Delay	LREAL	ALL	0	Transmission time of the reference switch to compensate the dead time in seconds.

nHomingMode	Homing mode	SMC_HOMING_MODE	-	-	Reference SMC_HOMING_MODE
bReturnTozero	Return to Home	BOOL	TRUE-FALSE	FALSE	TRUE:Axis run to position zero after homing is completed (note: if fHomePosition=10, the axis position becomes 10 after homing is completed, and bReturnTozero is true then the axis reverses to go 10 units to position 0 after homing is completed)
bIndexOccured	Pulse Signal	BOOL	TRUE-FALSE	FALSE	TRUE: flag pulse recording, effective when homing mode is FAST_BSLow_I_S_STOP, FAST_SLOW_I_S_STOP
fIndexPosition	Record position	LREAL	ALL	0	Position recorded during flag pulse
bIgnoreHWLimit	Ignore hard limits	BOOL	TRUE-FALSE	FALSE	TRUE sets hardware limit switch enable to false, if the same physical switch is used for both the hardware limit switch and the reference switch, then the hardware control will be set to false

**Output variable**

Output variable	Name	Data type	Range	initialization	Descriptive
bDone	Completion	BOOL	TRUE-FALSE	FALSE	TRUE: Homing is done.
bBusy	Executing	BOOL	TRUE-FALSE	FALSE	TRUE: In progress
bCommand Aborted	Command Interrupt	BOOL	TRUE-FALSE	FALSE	TRUE: Terminated by other commands
bError	Error	BOOL	TRUE-FALSE	FALSE	TRUE: An error occurred within the function block
nErrorID	Error Code	SMC_ERRO R	-	0	Error indication, see SMC_Error.
bStartLatchingIndex	Start Latch	BOOL	TRUE-FALSE	FALSE	Generated by the combination of "bIndexOccured" and "fIndexPosition".

	Boo lea n	Bit string					Integer							Real number		Moment, Duration, Date, String				
	BOOL	BYTE	WORD	DWORD	LWORD	USINT	UINT	UDINT	ULINT	SINT	INT	DINT	LINT	REAL	LREAL	TIME	DATE	TOD	DT	STRING
Axis		AXIS_REF_SM3																		
bExecute	√	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
fHomePosition	-	-	-	-	-	-	-	-	-	-	-	-	-	-	√	-	-	-	-	-
fVelocitySlow	-	-	-	-	-	-	-	-	-	-	-	-	-	-	√	-	-	-	-	-
fVelocityFast	-	-	-	-	-	-	-	-	-	-	-	-	-	-	√	-	-	-	-	-
fAcceleration	-	-	-	-	-	-	-	-	-	-	-	-	-	-	√	-	-	-	-	-
fDeceleration	-	-	-	-	-	-	-	-	-	-	-	-	-	-	√	-	-	-	-	-
fJerk	-	-	-	-	-	-	-	-	-	-	-	-	-	-	√	-	-	-	-	-
nDirection		MC DIRECTION																		
bReferenceSwitch	√	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
fSignalDelay	-	-	-	-	-	-	-	-	-	-	-	-	-	-	√	-	-	-	-	-
nHomingMode		SMC HOMING MODE																		

bReturnTozero	√	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
bIndexOccured	√	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
fIndexPosition	-	-	-	-	-	-	-	-	-	-	-	-	-	√	-	-	-	-	-	-
bIgnoreHWLimit	√	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
bDone	√	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
bBusy	√	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
bCommandAborted	√	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
bError	√	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
nErrorID		SMC_ERROR																		
bStartLatchingIndex	√	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-

### ⊙ Functional Description

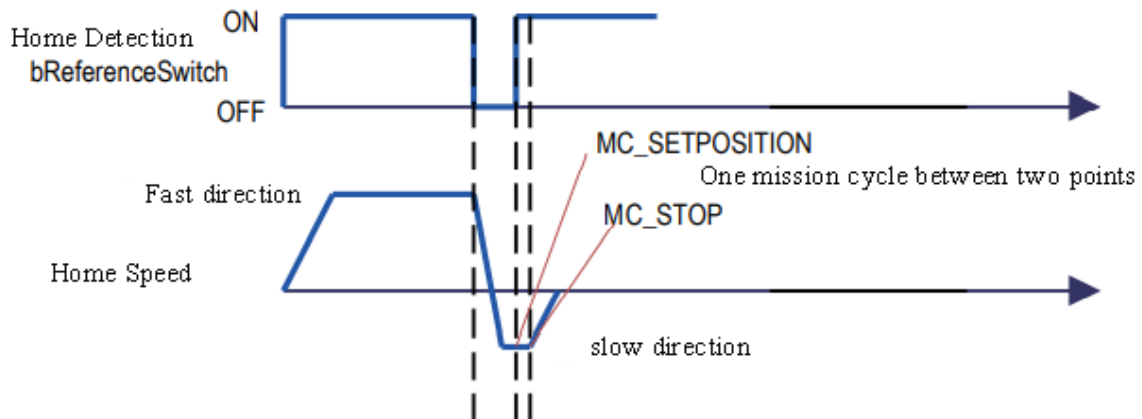
After SMC\_HOMING is initiated by the rising edge of bExecute, the axis will start moving at velocity fVelocityFast and in the direction defined by nDirection until bReferenceSwitch = FALSE. then the axis will slowly stop and leave the reference switch at velocity fVelocitySlow in the opposite direction. bReferenceSwitch = TRUE and the state of bReferenceSwitch is ON->OFF->ON after enabling the zero return instruction. Homing is complete after bReferenceSwitch = TRUE. bReferenceSwitch is ON->OFF->ON when the homing command is enabled, and homing is complete at the rising edge of OFF->ON, setting the reference position.

Reference position = fHomePosition + ((fSignalDelay \* 1000 + 1 DC clock cycle) / 1000 \* fVelocitySlow) Actually, it is to compensate for the sampling delay of bReferenceSwitch and one communication cycle displacement delay. If bReturnToZero = TRUE, the state of bReferenceSwitch on the rising edge of OFF->ON sets the reference position to fHomePosition + ((fSignalDelay \* 1000 + 1 DC clock cycle) / 1000 \* fVelocitySlow, and then run to position 0 according to the velocity fVelocityFast.

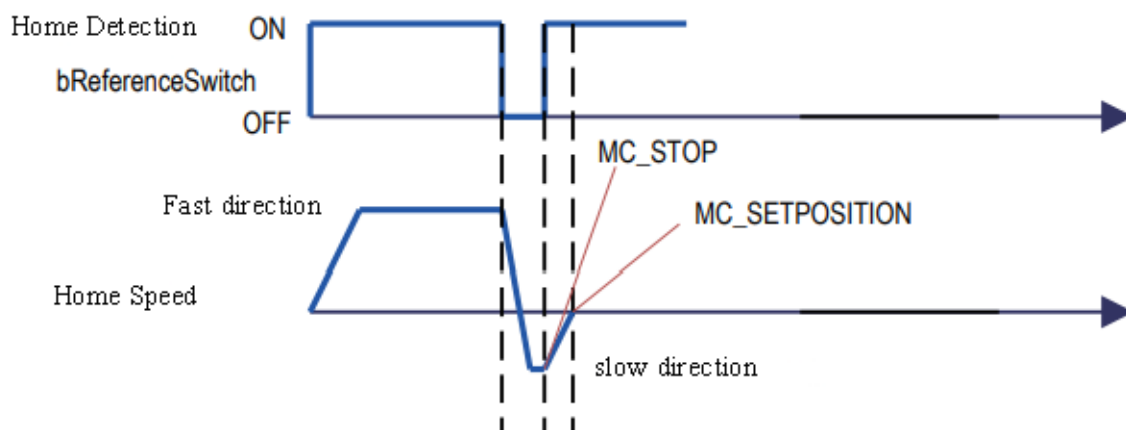
Note: After the Done signal, the axis position is set to: fHomePosition. The timing of setting is related to nHomingMode (refer to SMC\_HOMING\_MODE for details).

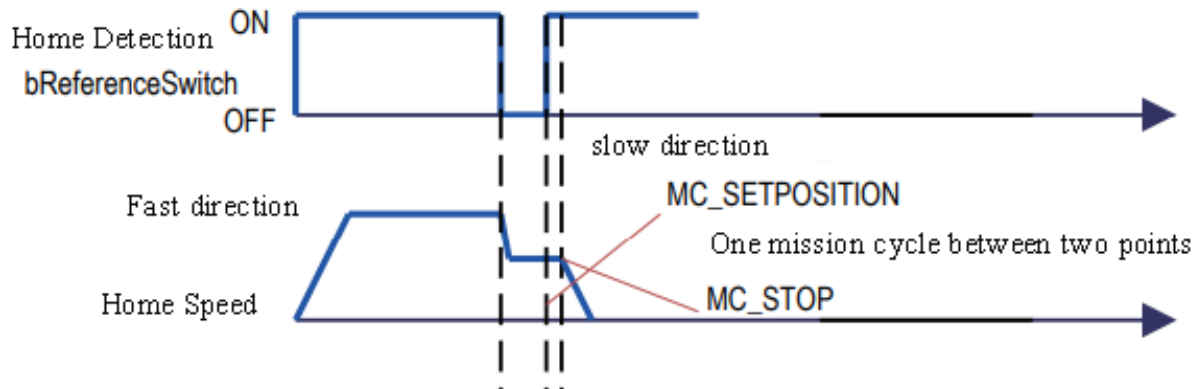
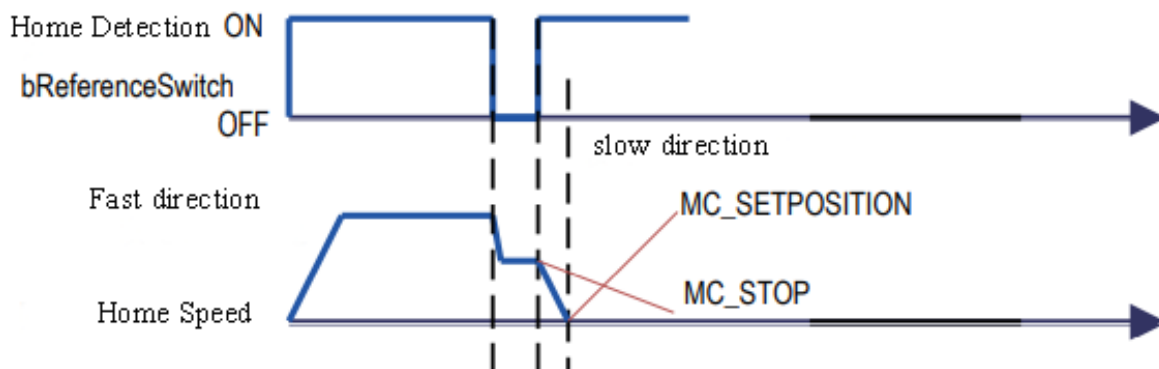
The following figure shows several return-to-zero modes:

#### 1) When homing mode is "0"



#### 2) When homing mode is "1"



**3) When homing mode is "4"**

**3) When homing mode is "5"**

**© Program demo**

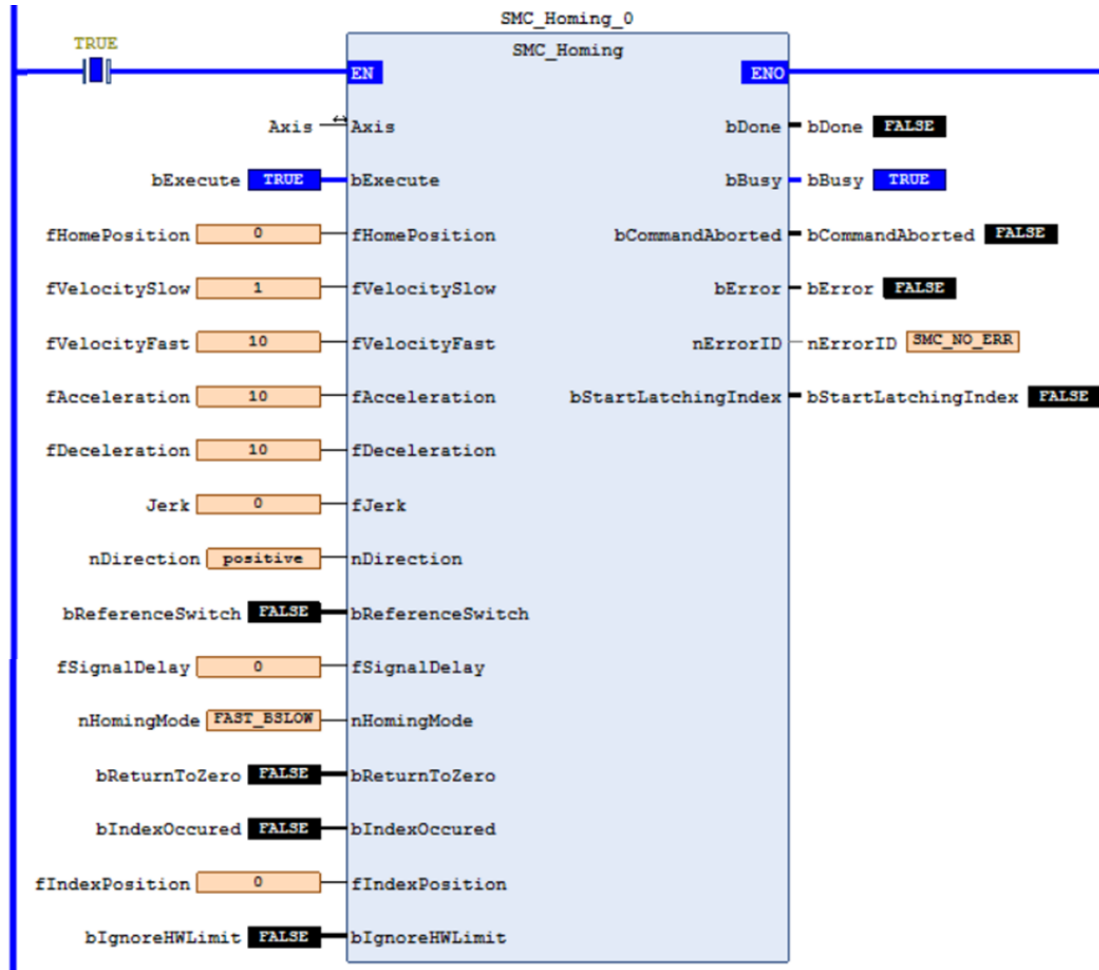
**ST:** When bExecute becomes TRUE, the axis return to zero instruction is executed.

```

SMC_Homing_0(
  Axis:= Axis,
  bExecute TRUE := bExecute TRUE,
  fHomePosition 0 := fHomePosition 0,
  fVelocitySlow 1 := fVelocitySlow 1,
  fVelocityFast 10 := fVelocityFast 10,
  fAcceleration 10 := fAcceleration 10,
  fDeceleration 10 := fDeceleration 10,
  fJerk 0 := fJerk 0,
  nDirection positive := nDirection positive,
  bReferenceSwitch FALSE := bReferenceSwitch FALSE,
  fSignalDelay 0 := fSignalDelay 0,
  nHomingMode FAST_BSLOW := nHomingMode FAST_BSLOW,
  bReturnToZero FALSE := bReturnToZero FALSE,
  bIndexOccured FALSE := bIndexOccured FALSE,
  fIndexPosition 0 := fIndexPosition 0,
  bIgnoreHWLimit FALSE := bIgnoreHWLimit FALSE,
  bDone FALSE => bDone FALSE,
  bBusy TRUE => bBusy TRUE,
  bCommandAborted FALSE => bCommandAborted FALSE,
  bError FALSE => bError FALSE,
  nErrorID SMC_NO_ERR => nErrorID SMC_NO_ERR,
  bStartLatchingIndex FALSE => bStartLatchingIndex FALSE );

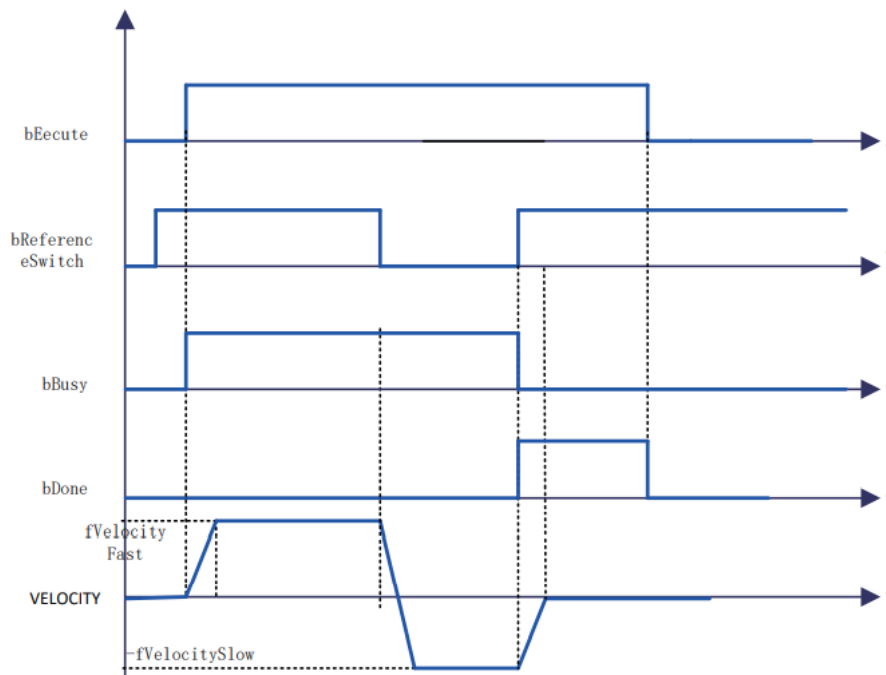
```

**LD:** When bExecute becomes TRUE, the axis return to zero instruction is executed.

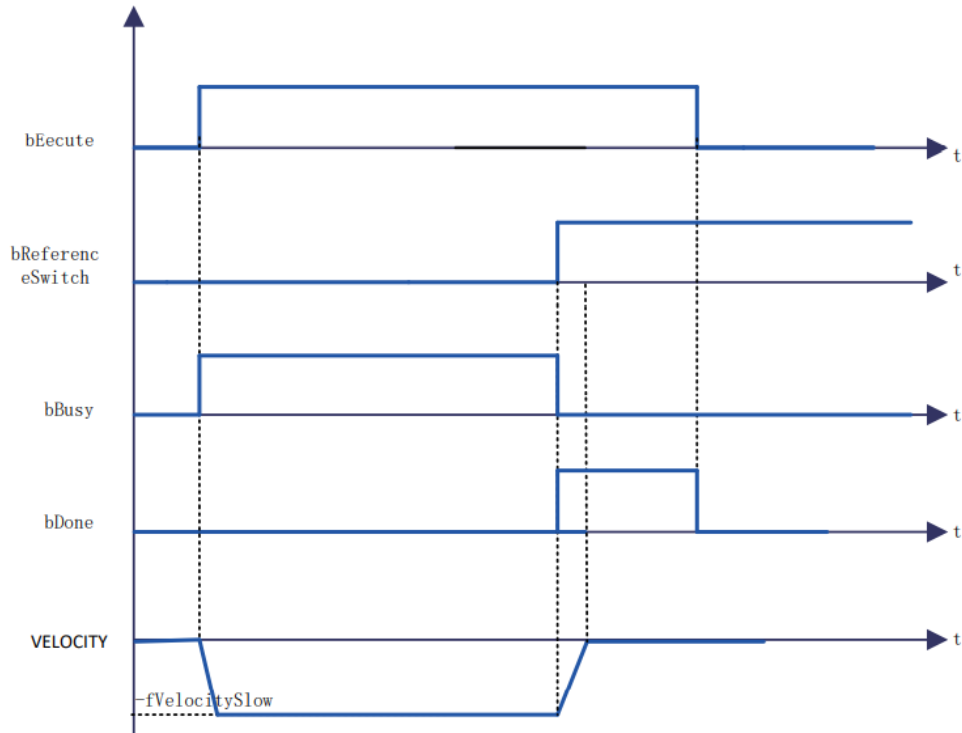


⊙ **Timing diagram:**

Instruction execution when bReferenceSwitch = TRUE



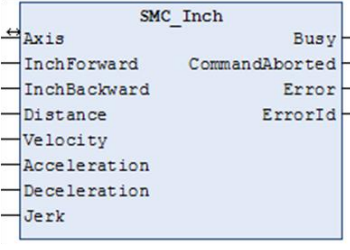
When bReferenceSwitch = FALSE at the time of instruction execution



### 4.2.15 SMC\_Inch

Manually control the axes to move in the specified direction, segment by segment, in the specified distance units.

#### ⊙ Command Format

Instruction	Name	FB/FC	LD	ST	File
SMC_Inch	Move an inch instruction	FC		SMC_Inch( Axis:=, InchForward:=, InchBackward:=, Distance:=, Velocity:=, Acceleration:=, Deceleration:=, Jerk:=, Busy=>, CommandAborted=>, Error=>, ErrorId=>);	SM3_Basic

#### ⊙ Related Variables

##### I/O variable

I/O variable	Name	Data type	Range	initialization	Descriptive
Axis	Axis	AXIS REF SM3	-	-	Specified axis

##### Input variable

Input variable	Name	Data type	Range	initialization	Descriptive
InchForward	Inch forward.	BOOL	TRUE-FALSE	FALSE	TRUE, the axis moves a set distance in the forward direction, and if InchForward is set FALSE to TRUE again, the axis moves another set distance. If the axis has not yet moved the set distance, InchForward becomes FALSE, the axis immediately decelerates to 0, and the Busy output is FALSE. If InchBackward is also true, the axis does not move.
InchBackward	Inching backwards	BOOL	TRUE-FALSE	FALSE	TRUE, the axis moves a set distance in the reverse direction. if InchBackward is set FALSE to TRUE again, the axis moves another set distance. If the axis has not yet moved the set distance, InchBackward becomes FALSE, the axis immediately decelerates to 0, and the Busy output is FALSE. if InchForward is true at the same time, the axis does not move.
Distance	Inching Distance	LREAL	ALL	0	Define the distance to be moved by the input axis at one time
Velocity	Target Velocity	LREAL	ALL	0	Maximum speed [u/s]
Acceleration	Target acceleration	LREAL	ALL	0	Acceleration in [u/s <sup>2</sup> ]. Increasing motor energy
Deceleration	Target Deceleration	LREAL	ALL	0	Deceleration in [u/s <sup>2</sup> ]. Decreasing motor energy
Jaek	Target jaek	LREAL	ALL	0	Maximum magnitude of jerk in [u/s <sup>3</sup> ]

**Output variable**

Output variable	Name	Data type	Range	initialization	Descriptive
Busy	Executing	BOOL	TRUE-FALSE	FALSE	TRUE: In progress
Command Aborted	Command Interrupt	BOOL	TRUE-FALSE	FALSE	TRUE: Terminated by other commands
Error	Error	BOOL	TRUE-FALSE	FALSE	TRUE: An error occurred within the function block
ErrorID	Error Code	SMC_ERRO R	-	0	Error indication, see SMC Error.

	Boo lea n	Bit string				Integer							Real number		Moment, Duration, Date, String					
	BOOL	BYTE	WORD	DWORD	LWORD	USINT	UINT	UDINT	ULINT	SINT	INT	DINT	LINT	REAL	LREAL	TIME	DATE	TOD	DT	STRING
Axis		AXIS_REF_SM3																		
InchForward	√	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
InchBackward	√	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
Distance	-	-	-	-	-	-	-	-	-	-	-	-	-	√	-	-	-	-	-	-
Velocity	-	-	-	-	-	-	-	-	-	-	-	-	-	√	-	-	-	-	-	-
Acceleration	-	-	-	-	-	-	-	-	-	-	-	-	-	√	-	-	-	-	-	-
Deceleration	-	-	-	-	-	-	-	-	-	-	-	-	-	√	-	-	-	-	-	-
Jerk	-	-	-	-	-	-	-	-	-	-	-	-	-	√	-	-	-	-	-	-
Busy	√	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
Command Aboned	√	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
Error	√	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
ErrorID		SMC_ERROR																		

**© Functional Description**

The maximum distance for one movement is fixed and defined by the parameter Distance.

If the distance Distance needs to be run again, the input (InchForward OR InchBackward) needs to be reset.

If the distance Distance has not yet been reached, the input (InchForward OR InchBackward) is reset to FALSE and the movement immediately decelerates to a stop.

When the inputs InchForward and InchBackward are both TRUE, the axis does not move. At this point, when one of the signals becomes FALSE, the axis will perform the motion specified by the signal that is also TRUE.

Parameters such as velocity, acceleration and deceleration for each forward or reverse motion are determined by Velocity, Acceleration, Deceleration and Jerk.

**Ⓞ Program demo**

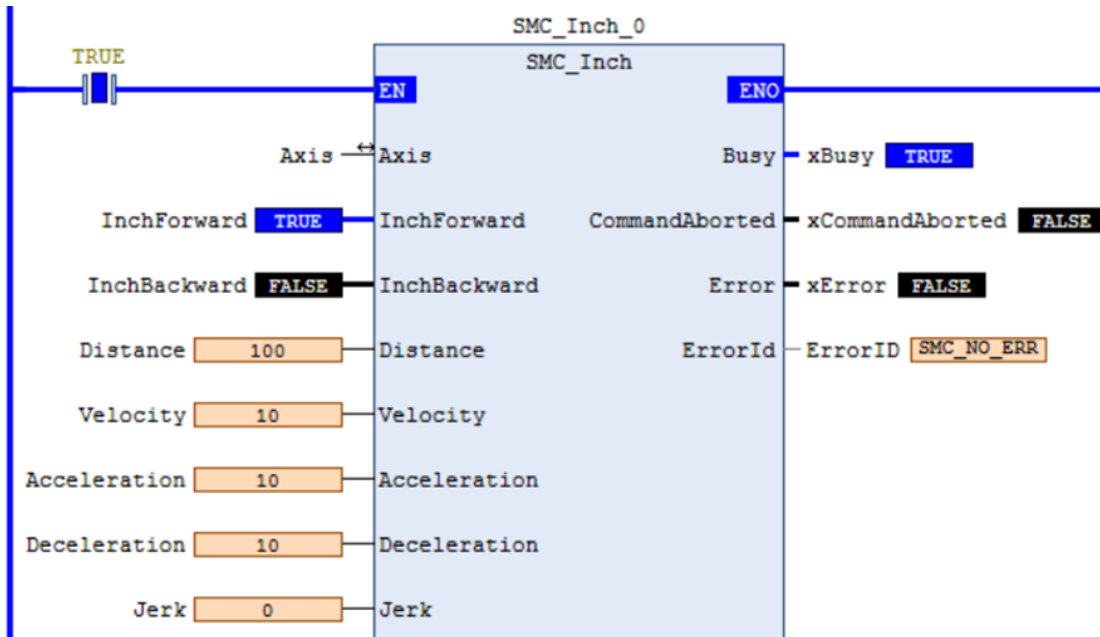
**ST:** When InchForward becomes TRUE, the specified axis moves the specified distance in the positive direction.

```

SMC_Inch_0(
  Axis:= Axis,
  InchForward TRUE := InchForward TRUE ,
  InchBackward FALSE := InchBackward FALSE ,
  Distance 100 := Distance 100 ,
  Velocity 10 := Velocity 10 ,
  Acceleration 10 := Acceleration 10 ,
  Deceleration 10 := Deceleration 10 ,
  Jerk 0 := Jerk 0 ,
  Busy TRUE => xBusy TRUE ,
  CommandAborted FALSE => xCommandAborted FALSE ,
  Error FALSE => xError FALSE ,
  ErrorId SMC_NO_ERR => ErrorId SMC_NO_ERR );

```

**LD:** When InchForward becomes TRUE, the specified axis moves the specified distance in the positive direction.



## 4.3 Axis synchronization function

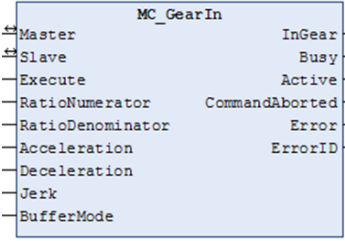
### 4.3.1 Instruction List

Instruction Type	Name	FB/FC	Functionality
Axis synchronization function	MC_GearIn	FB	E-Gear Input
	MC_GearInPos	FB	E-Gear Smooth Coupling
	MC_GearOut	FB	E-Gear Output
	MC_CamTableSelect	FB	E-Cam tappet association
	MC_CamIn	FB	E-cam association
	MC_CamOut	FB	E-cam disengagement
	SMC_GetTappetValue	FB	Read tappet status

### 4.3.2 MC\_GearIn

Setting the master-slave gear ratio and activating the electronic gears

#### ⊙ Command Format

Instruction	Name	FB/FC	LD	ST	File
MC_GearIn	E-Gear Input instruction	FC		MC_GearIn( Master:= , Slave:=, Execute:= , RatioNumerator:=, RatioDenominator:=, Acceleration:= , Deceleration:=, Jerk:=, BufferMode:=, InGear=>, Busy=>, Active=>, CommandAborted=>, Error=>, ErrorID=> );	SM3_Basic

#### ⊙ Related Variables

##### I/O variable

I/O variable	Name	Data type	Range	initialization	Descriptive
Master	Main axis	AXIS_REF_SM3	-	-	Specify spindle
Slave	axle	AXIS_REF_SM3	-	-	Specify slave axis

##### Input variable

Input variable	Name	Data type	Range	initialization	Descriptive
Execute	Start	BOOL	TRUE-FALSE	FALSE	Rising edge trigger
RatioNumerator	Numerator	DINT	Positive number	1	Gear ratio numerator
RatioDenominator	Denominator	UDINT	Positive number	1	Gear ratio denominator
Acceleration	Target acceleration	LREAL	ALL	0	Electronic Gear Ratio Acceleration
Deceleration	Target Deceleration	LREAL	ALL	0	Electronic gear ratio deceleration
Jaek	Target jaek	LREAL	ALL	0	Acceleration
BufferMode	caching mode	MC_BUFFER_MODE	-	-	Buffer mode

**Output variable**

Output variable	Name	Data type	Range	initialization	Descriptive
InGear	Gears mesh	BOOL	TRUE-FALSE	FALSE	TRUE: Processing of gear ratios is complete!
Busy	Executing	BOOL	TRUE-FALSE	FALSE	TRUE: Processing of the function block is not completed
Active	Medium term	BOOL	TRUE-FALSE	FALSE	The shaft works
Command Aborted	Command Interrupt	BOOL	TRUE-FALSE	FALSE	TRUE: Terminated by other commands
Error	Error	BOOL	TRUE-FALSE	FALSE	TRUE: An error occurred within the function block
ErrorID	Error Code	SMC_ERRO R	-	0	Error indication, see SMC Error.

	Boo lea n	Bit string					Integer							Real number		Moment, Duration, Date, String				
	BOOL	BYTE	WORD	DWORD	LWORD	USINT	UINT	UDINT	ULINT	SINT	INT	DINT	LINT	REAL	LREAL	TIME	DATE	TOD	DT	STRING
Master		AXIS_REF_SM3																		
Slave		AXIS_REF_SM3																		
Execute	√	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
RatioNumerator	-	-	-	-	-	-	-	-	-	-	-	√	-	-	-	-	-	-	-	-
RatioDenominator	-	-	-	-	-	-	-	√	-	-	-	-	-	-	-	-	-	-	-	-
Acceleration	-	-	-	-	-	-	-	-	-	-	-	-	-	-	√	-	-	-	-	-
Deceleration	-	-	-	-	-	-	-	-	-	-	-	-	-	-	√	-	-	-	-	-
Jerk	-	-	-	-	-	-	-	-	-	-	-	-	-	-	√	-	-	-	-	-
BufferMode		MC_BUFFER_MODE																		
InGear	√	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
Busy	√	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
Active	√	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
Command Aboned	√	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
Error	√	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
ErrorID		SMC_ERROR																		

**© Functional Description**

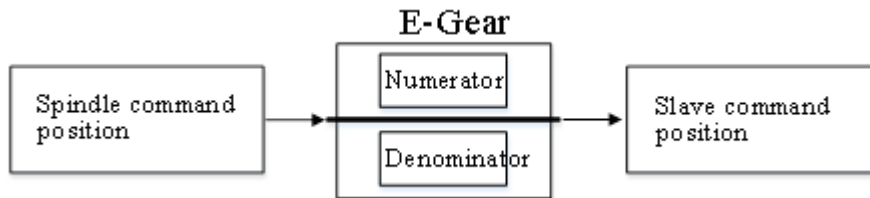
This instruction is implemented by the "SM3\_Basic" library.

After the execution of the electronic gear, the slave axis will be synchronised and to decouple it, the GearOut instruction must be called.

The instruction is of the speed synchronisation type and when activated, the slave axis speed will increase to the given gear ratio. At the completion of this process, the slave axis coupling is complete. Loss of synchronisation distance caused during coupling is not compensated.

While the MC\_GearIn instruction is running, the master-slave electronic gear ratio can be changed by retriggering the MC\_GearIn instruction without first calling the MC\_GearOut instruction to pause the original electronic gear movement.

When the target speed is reached, the InGear signal is TRUE, and after that, the slave axis movement = master axis movement \* RatioNumerator / RatioDenominator



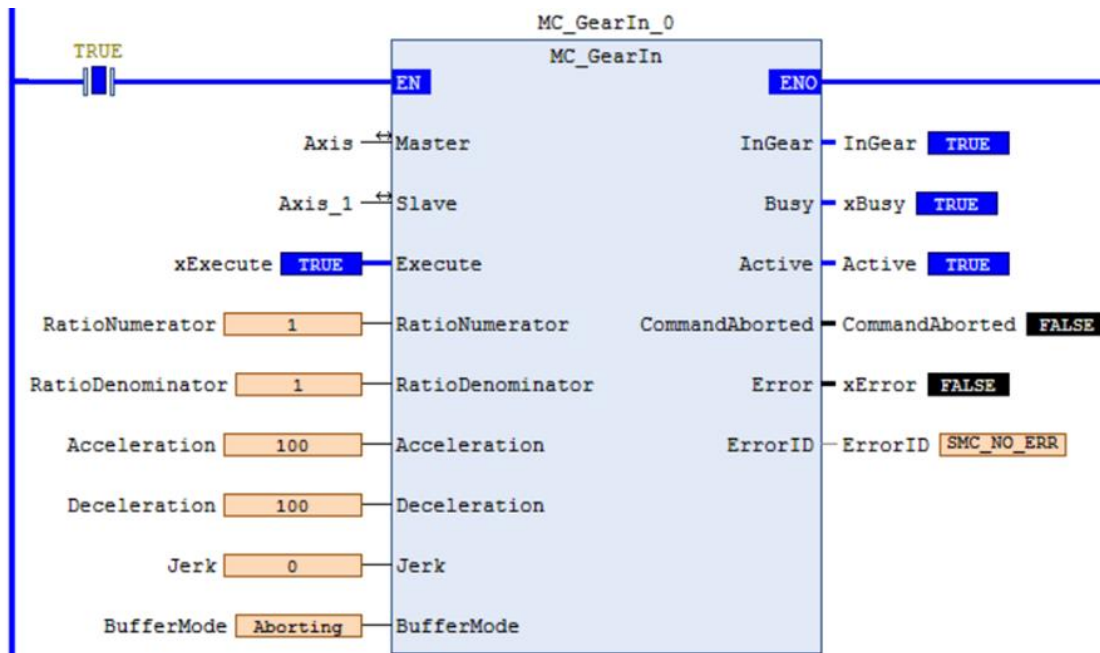
This is used when the spindle speed is stable. If the spindle speed changes in real time, please note that this command should be used carefully.

### ⊙ Program demo

**ST:** When xExecute becomes TRUE, the master-slave gear ratio is set and the electronic gear is activated.

```
MC_GearIn_0(
    Master:= Axis,
    Slave:= Axis_1,
    Execute TRUE := xExecute TRUE,
    RatioNumerator 1 := RatioNumerator 1,
    RatioDenominator 1 := RatioDenominator 1,
    Acceleration 100 := Acceleration 100,
    Deceleration 100 := Deceleration 100,
    Jerk 0 := Jerk 0,
    BufferMode Aborting := BufferMode Aborting,
    InGear TRUE => InGear TRUE,
    Busy TRUE => Busy TRUE,
    Active TRUE => Active TRUE,
    CommandAborted FALSE => CommandAborted FALSE,
    Error FALSE => xError FALSE,
    ErrorID SMC_NO_ERR => ErrorID SMC_NO_ERR );
```

**LD:** When xExecute becomes TRUE, the master-slave gear ratio is set and the electronic gear is activated.



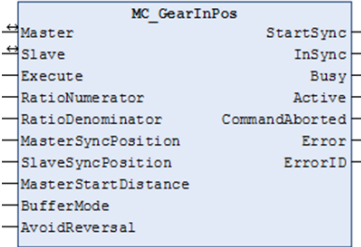
### Note:

Do not use the MC\_SetPosition command during the execution of the command as it may cause the motor to run rapidly and cause accidents.

### 4.3.3 MC\_GearInPos

Sets the master-slave axis electronic gear ratio and executes the electronic gear movement. The difference with the MC\_GearIn instruction is that this instruction needs to specify the master axis position where synchronisation starts, the slave axis position, and the distance at which synchronisation of the master axis starts, and this is used to complete the cut-in electronic gear movement.

#### ⊙ Command Format

Instruction	Name	FB/FC	LD	ST	File
MC_GearInPos	E-Gear Smooth Coupling instruction	FC		MC_GearInPos( Master:=, Slave:=, Execute:=, RatioNumerator:=, RatioDenominator:=, MasterSyncPosition:=, SlaveSyncPosition:=, MasterStartDistance:=, BufferMode:=, AvoidReversal:=, StartSync=>, InSync=>, Busy=>, Active=>, CommandAborted=>, Error=>, ErrorID=> );	SM3_Basic

#### ⊙ Related Variables

##### I/O variable

I/O variable	Name	Data type	Range	initialization	Descriptive
Master	Main axis	AXIS_REF_SM3	-	-	Specify spindle
Slave	axle	AXIS_REF_SM3	-	-	Specify slave axis

##### Input variable

Input variable	Name	Data type	Range	initialization	Descriptive
Execute	Start	BOOL	TRUE-FALSE	FALSE	Rising edge trigger
RatioNumerator	Numerator	DINT	Positive number	1	Gear ratio numerator
RatioDenominator	Denominator	DINT	Positive number	1	Gear ratio denominator
MasterSyncPosition	Master axis synchronised position	REAL	ALL	0	Position of the master axis when the master-slave axes are synchronised
SlaveSyncPosition	Slave shaft synchronised position	REAL	ALL	0	Position of the slave axis when the master and slave axes are synchronised
MasterStartDistance	Engaging segment spindle travel distance	REAL	Positive number	0	Distance of the master axis movement from the start of the slave axis to the time when it is synchronised with the master axis.
BufferMode	Cache mode	MC_BUFFER_MODE	-	-	Only BufferModes Aborting, Buffered and BlendingPrevious are supported.
AvoidReversal	Reverse rotation prohibited	BOOL	TRUE-FALSE	FALSE	FALSE: Reversal is performed if the slave axis physically overruns its position; \n TRUE: reversal is not physically possible

					from the axis or causes danger, only applicable with modal axes. If the reversal cannot be avoided, then the axis will stop incorrectly.
--	--	--	--	--	---

**Output variable**

Output variable	Name	Data type	Range	initialization	Descriptive
StartSync	Start synchronisation	BOOL	TRUE-FALSE	FALSE	TRUE: Electronic Gear starts processing
InSync	Arrival synchronisation	BOOL	TRUE-FALSE	FALSE	TRUE: Electronic Gear command completed
Busy	Executing	BOOL	TRUE-FALSE	FALSE	TRUE: Processing of the function block is not completed
Active	Medium term	BOOL	TRUE-FALSE	FALSE	The shaft works
Command Aborted	Command Interrupt	BOOL	TRUE-FALSE	FALSE	TRUE: Terminated by other commands
Error	Error	BOOL	TRUE-FALSE	FALSE	TRUE: An error occurred within the function block
ErrorID	Error Code	SMC_ERRO R	-	0	Error indication, see SMC Error.

	Boo lea n	Bit string					Integer							Real number		Moment, Duration, Date, String				
	BOOL	BYTE	WORD	DWORD	LWORD	USINT	UINT	UDINT	ULINT	SINT	INT	DINT	LINT	REAL	LREAL	TIME	DATE	TOD	DT	STRING
Master		AXIS_REF_SM3																		
Slave		AXIS_REF_SM3																		
Execute	√	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
RatioNumerator	-	-	-	-	-	-	-	-	-	-	-	√	-	-	-	-	-	-	-	-
RatioDenominator	-	-	-	-	-	-	-	-	-	-	-	√	-	-	-	-	-	-	-	-
MasterSyncPosition	-	-	-	-	-	-	-	-	-	-	-	-	-	-	√	-	-	-	-	-
SlaveSyncPosition	-	-	-	-	-	-	-	-	-	-	-	-	-	-	√	-	-	-	-	-
MasterStartDistance	-	-	-	-	-	-	-	-	-	-	-	-	-	-	√	-	-	-	-	-
BufferMode		MC_BUFFER_MODE																		
AvoidReversal	√	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
StartSync	√	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
InSync	√	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
Busy	√	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
Active	√	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
Command Aboned	√	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
Error	√	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
ErrorID		SMC_ERROR																		

**⊙ Functional Description**

This instruction is implemented by the "SM3\_Basic" library.

After the start of the synchronisation, the slave axis performs acceleration and deceleration with the speed obtained by multiplying the spindle speed by the gear ratio as the target speed.

The process from the start of synchronisation to the end of synchronisation of this function block is essentially an electronic cam in which the slave axis follows the master axis during the synchronisation interval. At this point, according to the range of the master axis (MasterSyncPosition - MasterStartDistance, MasterSyncPosition), the range of the slave axis (CurrentPosition, SlaveSyncPosition), the instruction will automatically design a cam curve according to the set gear ratio and the above three parameters, and the slave axis will follow the main axis to complete the cam action when performing synchronisation.

Note that if the master and slave axes are working in linear mode, it is necessary to ensure that the above parameters are set appropriately or else the gear movements will not be performed correctly, so it is recommended that the master and slave axes are in modular mode when using this command. For example, if the master and slave axes are working in linear mode and both are moving in the forward direction, if the command is executed with MasterSyncPosition - MasterStartDistance or SlaveSyncPosition > SlaveSyncPosition, the electronic gearing will not be able to cut in.

The target speed is reached while synchronisation is completed (InSync is TRUE).

$SlaveSyncPosition = MasterSyncPosition * RatioNumerator / RatioDenominator$

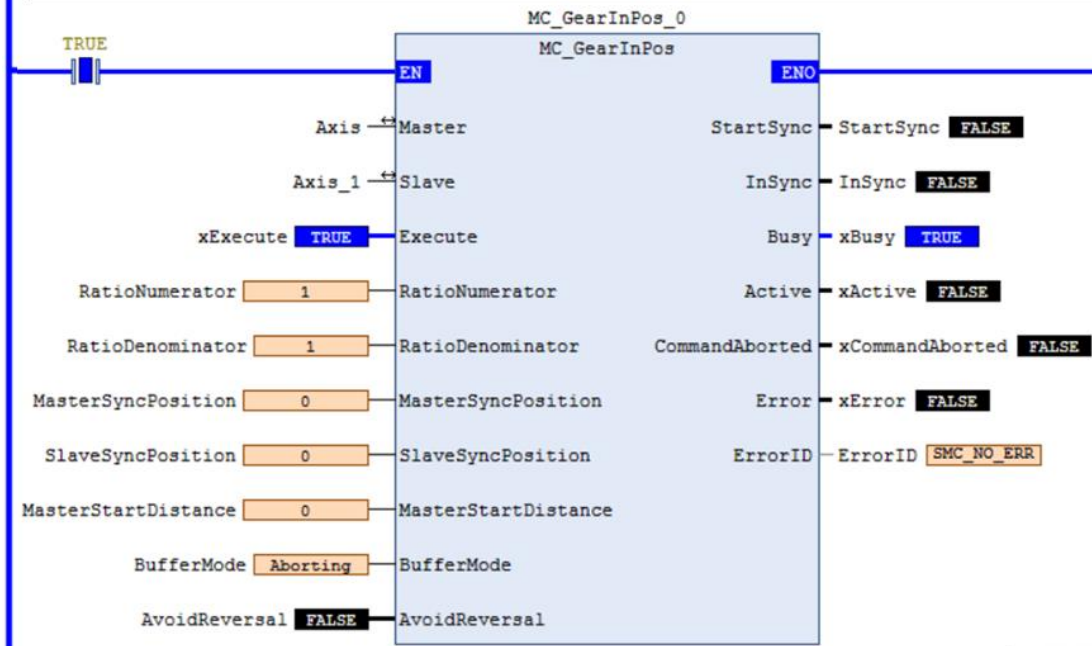
For AvoidReversal: If the slave axis is a modal axis and the master axis speed (a multiple of the gear ratio) is not relative to the slave axis speed, then MC\_GearInPos tries to avoid the slave axis reversal. It tries to "stretch" the motion of the slave axis by adding 5 slave cycles. If this "stretching" is not effective, then an error occurs and the slave axis stops incorrectly. If the slave speed is related to the master speed (a multiple of the gear ratio), then an error occurs and the axis stops incorrectly. If the slave axis is a linear mode axis, then an error will be generated when the rising edge of the Execute input is processed.

**⊙ Program demo**

**ST:** When xExecute becomes TRUE, the gear smoothing coupling instruction is executed.

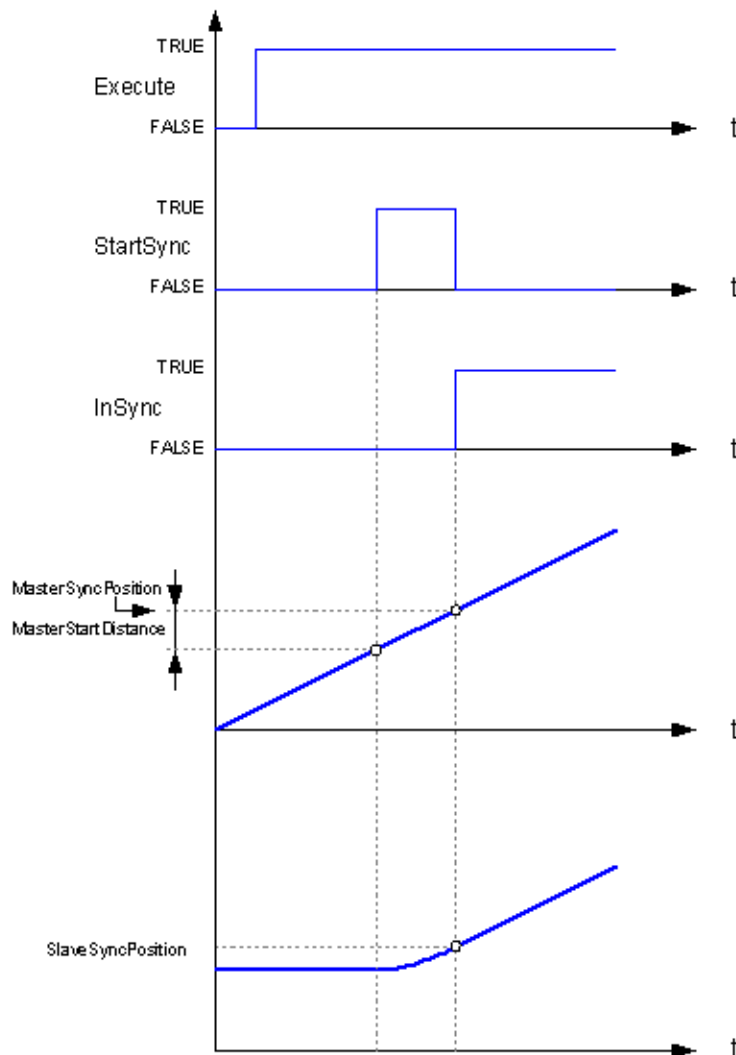
```
MC_GearInPos_0(
    Master:= Axis,
    Slave:= Axis_1,
    Execute TRUE := xExecute TRUE,
    RatioNumerator 1 := RatioNumerator 1,
    RatioDenominator 1 := RatioDenominator 1,
    MasterSyncPosition 0 := MasterSyncPosition 0,
    SlaveSyncPosition 0 := SlaveSyncPosition 0,
    MasterStartDistance 0 := MasterStartDistance 0,
    BufferMode Aborting := BufferMode Aborting,
    AvoidReversal FALSE := AvoidReversal FALSE,
    StartSync FALSE => StartSync FALSE,
    InSync FALSE => InSync FALSE,
    Busy TRUE => xBusy TRUE,
    Active FALSE => xActive FALSE,
    CommandAborted FALSE => xCommandAborted FALSE,
    Error FALSE => xError FALSE,
    ErrorID SMC_NO_ERR => ErrorID SMC_NO_ERR );
```

**LD:** When xExecute becomes TRUE, the gear smoothing coupling instruction is executed.



**Timing Diagram:**

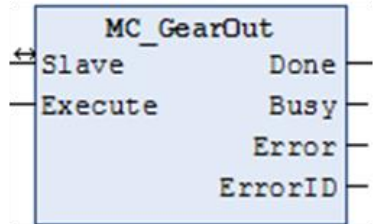
The timing diagram below shows the timing of the main signals of the MC\_GearInPos instruction after the Execute signal is set to True.



### 4.3.4 MC\_GearOut

Sets the master-slave axis electronic gear ratio and executes the electronic gear movement. The difference with the MC\_GearIn instruction is that this instruction needs to specify the master axis position where synchronisation starts, the slave axis position, and the distance at which synchronisation of the master axis starts, and this is used to complete the cut-in electronic gear movement.

#### ⊙ Command Format

Instruction	Name	FB/FC	LD	ST	File
MC_GearOut	E-Gear Output instruction	FC		MC_GearOut( Slave:=, Execute:=, one=>, Busy=>, Error=>, ErrorID=> );	SM3_Basic

#### ⊙ Related Variables

##### I/O variable

I/O variable	Name	Data type	Range	initialization	Descriptive
Slave	axle	AXIS_REF_SM3	-	-	Specify slave axis

##### Input variable

Input variable	Name	Data type	Range	initialization	Descriptive
Execute	Start	BOOL	TRUE-FALSE	FALSE	Rising edge trigger

##### Output variable

Output variable	Name	Data type	Range	initialization	Descriptive
Done	Implementation completed	BOOL	TRUE-FALSE	FALSE	TRUE: Electronic cam disconnection
Busy	Executing	BOOL	TRUE-FALSE	FALSE	TRUE: Processing of the function block is not completed
Error	Error	BOOL	TRUE-FALSE	FALSE	TRUE: An error occurred within the function block
ErrorID	Error Code	SMC_ERROR	-	0	Error indication, see SMC Error.

	Boolean	Bit string					Integer							Real number		Moment, Duration, Date, String				
	BOOL	BYTE	WORD	DWORD	LWORD	USINT	UINT	UDINT	ULINT	SINT	INT	DINT	LINT	REAL	LREAL	TIME	DATE	TOD	DT	STRING
Slave		AXIS_REF_SM3																		
Execute	√	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
Done	√	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
Busy	√	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
Error	√	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
ErrorID		SMC_ERROR																		

**⊙ Functional Description**

This instruction is implemented by the "SM3\_Basic" library.

After cutting out the electronic gear, the speed of the slave axis is the speed before cutting out, so it is necessary to stop the slave axis with the MC\_Stop instruction.

Specify the axis to which the action is to be directed by Slave, specify Deceleration, and stop the MC\_GearIn (start of gear action) and MC\_GearInPos (position-designated gear action) instructions that are being executed.

This instruction has no effect on the spindle action of the MC\_GearIn (gear action start) instruction and MC\_GearInPos (position-specified gear action) instruction.

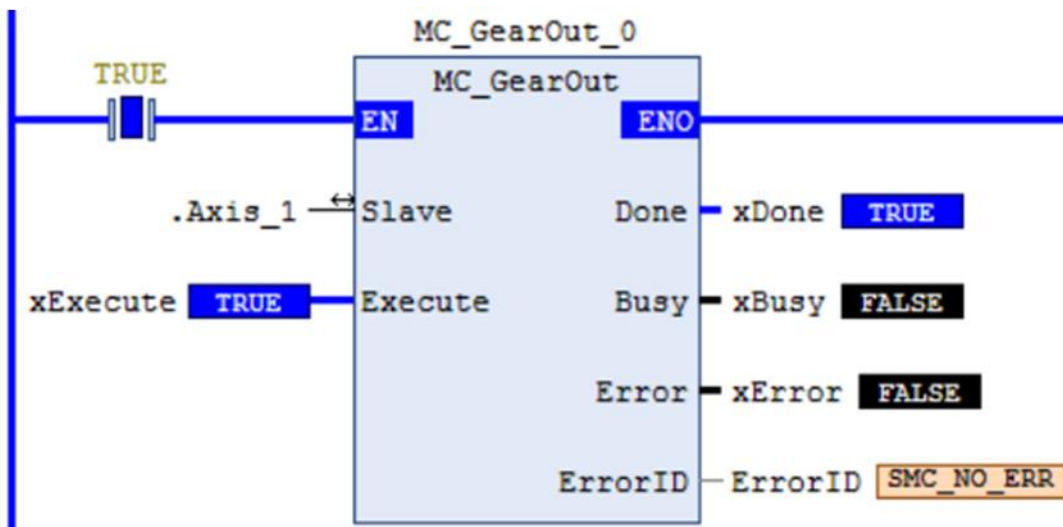
**⊙ Program demo**

**ST:** When xExecute becomes TRUE, the gear disengagement instruction is executed.

```

MC_GearOut_0 (
  Slave:= Axis_1 ,
  Execute TRUE := xExecute TRUE ,
  Done TRUE => xDone TRUE ,
  Busy FALSE => xBusy FALSE ,
  Error FALSE => xError FALSE ,
  ErrorID SMC_NO_ERR => ErrorID SMC_NO_ERR ) ;
  
```

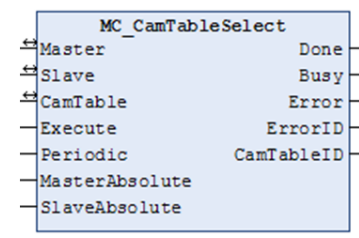
**LD:** When xExecute becomes TRUE, the gear disengagement instruction is executed.



### 4.3.5 MC\_CamTableSelect

Used to select the cam table to be executed, and needs to be used in conjunction with the MC\_CamIn instruction.

#### ⊙ Command Format

Instruction	Name	FB/FC	LD	ST	File
MC_CamTableSelect	E-Cam tappet association instruction	FC		MC_CamTableSelect( Master:=, Slave:=, CamTable:=, Execute:=, Periodic:=, MasterAbsolute:=, SlaveAbsolute:=, Done=>, Busy=>, Error=>, ErrorID=>, CamTableID=>);	SM3_Basic

#### ⊙ Related Variables

##### I/O variable

I/O variable	Name	Data type	Range	initialization	Descriptive
Master	Main axis	AXIS REF	-	-	Specify spindle
Slave	axle	AXIS REF	-	-	Specify slave axis
CamTable	Cam watch	MC CAM REF	-	-	Designated Cam Table

##### Input variable

Input variable	Name	Data type	Range	initialization	Descriptive
Execute	Start	BOOL	TRUE-FALSE	FALSE	Rising edge trigger
Periodic	Repeat mode	BOOL	TRUE-FALSE	FALSE	TRUE: periodic, repeated execution of the specified cam table FALSE: The cam table is executed only once.
MasterAbsolute	Master axis absolute mode	BOOL	TRUE-FALSE	FALSE	TRUE: absolute coordinates, FALSE: Relative coordinates.
SlaveAbsolute	Slave absolute mode	BOOL	TRUE-FALSE	FALSE	TRUEp: Absolute coordinates, FALSE: Relative coordinates

##### Output variable

Output variable	Name	Data type	Range	initialization	Descriptive
Done	Implementation completed	BOOL	TRUE-FALSE	FALSE	TRUE:Electronic cam disconnection
Busy	Executing	BOOL	TRUE-FALSE	FALSE	TRUE: Processing of the function block is not completed
Error	Error	BOOL	TRUE-FALSE	FALSE	TRUE: An error occurred within the function block
ErrorID	Error Code	SMC_ERROR	-	0	Error indication, see SMC_Error.
CamTableID	Tappet ID	MC_CAM_ID	-	-	CAM: Form definition for function block MC_CamID

	Boolean	Bit string					Integer							Real number		Moment, Duration, Date, String				
	BOOL	BYTE	WORD	DWORD	LWORD	USINT	UINT	UDINT	ULINT	SINT	INT	DINT	LINT	REAL	LREAL	TIME	DATE	TOD	DT	STRING
Master																				
Slave																				
CamTable																				
Execute	√	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
Periodic	√	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
MasterAbsolute	√	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
SlaveAbsolute	√	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
Done	√	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
Busy	√	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
Error	√	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
ErrorID																				
CamTableID																				

### ⊙ Functional Description

This instruction is implemented by the "SM3\_Basic" library.

This instruction is used to specify the cam table required for the operation of the electronic cam, so it is necessary to edit the cam table (in the cam editor or online) before using this instruction.

The rising edge of Execute executes the specified cam table, and also refreshes the specified cam table when it is updated.

When the Done signal is output as TRUE, the output variable "CamTableID" is generated and becomes effective.

The master and slave axes cannot be specified as the same axis, otherwise an error will be reported.

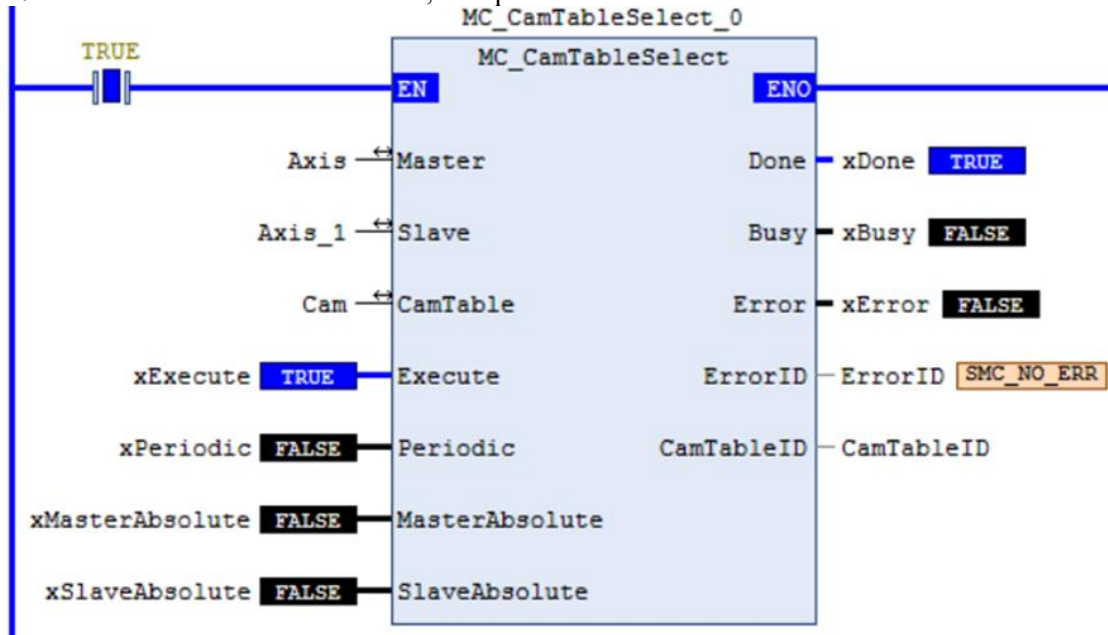
### ⊙ Program demo

**ST:** When xExecute becomes TRUE, the specified cam table instruction is executed.

```

MC_CamTableSelect_0(
  Master:= Axis,
  Slave:= Axis_1,
  CamTable:= Cam,
  Execute TRUE := xExecute TRUE,
  Periodic FALSE := xPeriodic FALSE,
  MasterAbsolute FALSE := xMasterAbsolute FALSE,
  SlaveAbsolute FALSE := xSlaveAbsolute FALSE,
  Done TRUE => xDone TRUE,
  Busy FALSE => xBusy FALSE,
  Error FALSE => xError FALSE,
  ErrorID SMC_NO_ERR => ErrorID SMC_NO_ERR,
  CamTableID=> CamTableID);
  
```

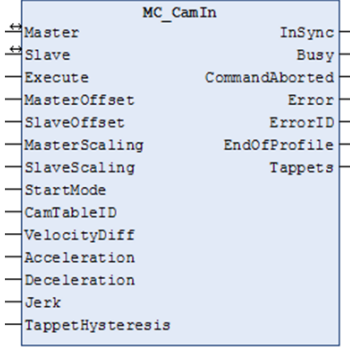
**LD:** When xExecute becomes TRUE, the specified cam table instruction is executed.



### 4.3.6 MC\_CamIn

Use the specified cam table to start executing the electronic cam action.

#### ⊙ Command Format

Instruction	Name	FB/FC	LD	ST	File
MC_CamIn	E-cam association instruction	FC		MC_CamIn( Master:=, Slave:=, Execute:=, MasterOffset:=, SlaveOffset:=, MasterScaling:=, SlaveScaling:=, StartMode:=, CamTableID:=, VelocityDiff:=, Acceleration:=, Deceleration:=, Jerk:=, TappetHysteresis:=, InSync=>, Busy=>, CommandAborted=>, Error=>, ErrorID=>, EndOfProfile=>, Tappets=> );	SM3_Basic

#### ⊙ Related Variables

##### I/O variable

I/O variable	Name	Data type	Range	initialization	Descriptive
Master	Main axis	AXIS_REF	-	-	Specify spindle
Slave	axle	AXIS_REF	-	-	Specify slave axis

##### Input variable

Input variable	Name	Data type	Range	initialization	Descriptive
Execute	Start	BOOL	TRUE-FALSE	FALSE	Rising edge trigger
MasterOffset	Master Axis Offset	LREAL	ALL	0	Offset from master axis list
SlaveOffset	Slave axis offset	LREAL	ALL	0	Offset from axis list
MasterScaling	Master Axis Ratio	LREAL	Positive number	1	Master axis scaling
SlaveScaling	Slave Ratio	LREAL	Positive number	1	Slave axis scaling
StartMode	Follower shaft engagement mode	MC_StartMode	0-4	0: absolute	0: absolute absolute position: 1: relative Relative position: 2: ramp_in (ramp cut) 3: ramp_in_pos(Positive slope cut-in) 4: ramp_in_neg (reverse slope cut)
CamTableID	Cam table	MC_CAM_ID	-	-	Defines the use of the cam form, which is an output of MC_CamID
VelocityDiff	Stacked	LREAL	ALL	0	Maximum speed different from ramp_in

	acceleration				
Acceleration	Target acceleration	LREAL	Positive number	0	Maximum acceleration against ramp_in
Deceleration	Target Deceleration	LREAL	Positive number	0	Maximum deceleration against ramp_in
Jerk	Target jaek	LREAL	ALL	0	Acceleration
TappetHysteresis	Tappet damping	LREAL	Positive number	0	Size of tappet lag

**Output variable**

Output variable	Name	Data type	Range	initialization	Descriptive
InSync	Synchronised	BOOL	TRUE-FALSE	FALSE	TRUE: Slave axes are synchronised with the spindle according to the cam table
Busy	Executing	BOOL	TRUE-FALSE	FALSE	TRUE: Processing of the function block is not completed
Command Aborted	Command Interrupt	BOOL	TRUE-FALSE	FALSE	TRUE: Terminated by other commands
Error	Error	BOOL	TRUE-FALSE	FALSE	TRUE: An error occurred within the function block
ErrorID	Error Code	SMC_ERROR	-	0	Error indication, see SMC_Error.
EndOfProfile	Curve Finish	BOOL	TRUE-FALSE	FALSE	Pulse signal output at the end of the cam compilation cycle
Tappets	Tappet table	SMC_TappetData	-	-	Tappet signal for SMC_GetTappetValue processing

	Boo	Bit string					Integer							Real number		Moment, Duration, Date, String					
	lea	BOO	BY	WO	DWO	LWO	US	UI	UD	UL	S	I	D	L	REAL	LREAL	TIME	DATE	TOD	DT	STRING
Master		AXIS_REF_SM3																			
Slave		AXIS_REF_SM3																			
Execute	√	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
MasterOffset	-	-	-	-	-	-	-	-	-	-	-	-	-	-	√	-	-	-	-	-	-
SlaveOffset	-	-	-	-	-	-	-	-	-	-	-	-	-	-	√	-	-	-	-	-	-
MasterScaling	-	-	-	-	-	-	-	-	-	-	-	-	-	-	√	-	-	-	-	-	-
SlaveScaling	-	-	-	-	-	-	-	-	-	-	-	-	-	-	√	-	-	-	-	-	-
StartMode		MC_StartMode																			
CamTableID		MC_CAM_ID																			
VelocityDiff	-	-	-	-	-	-	-	-	-	-	-	-	-	-	√	-	-	-	-	-	-
Acceleration	-	-	-	-	-	-	-	-	-	-	-	-	-	-	√	-	-	-	-	-	-
Deceleration	-	-	-	-	-	-	-	-	-	-	-	-	-	-	√	-	-	-	-	-	-
Jerk	-	-	-	-	-	-	-	-	-	-	-	-	-	-	√	-	-	-	-	-	-
TappetHysteresis	-	-	-	-	-	-	-	-	-	-	-	-	-	-	√	-	-	-	-	-	-
InSync	√	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
Busy	√	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
CommandAborted	√	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
Error	√	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
ErrorID		SMC_ERROR																			

EndOfProfile	√	√	√	√	√	√	√	√	√	√	√	√	√	√	√	√	√	√	√	√	√	√
Tappets	SMC_TappetData																					

### ⊙ Functional Description

This instruction is implemented by the "SM3\_Basic" library.

This instruction synchronises the phase (master axis) and displacement (slave axis) cam movements according to the cam table.

The cam table specified by this instruction can be created in two ways:

- 1) Prepared using the Cam Editor;
- 2) by programming a self-built cam table data structure.

In a cam system, to call a cam curve, first call the MC\_CamTableSelect instruction to select the corresponding cam table, and then execute MC\_CamIn; if you want to replace the cam curve, then call the MC\_CamTableSelect instruction to select the cam table again.

If you want to change the cam curve, call MC\_CamTableSelect instruction to select the cam table again. It is necessary to use MC\_CamOut instruction to disconnect the cam coupling relationship between master and slave axis.

When this instruction is executed, when the slave axis of this instruction executes other motion instructions, the cam relationship between the slave axis and the master axis will be lifted, and the output of the Command-Aborted variable of MC\_CamIn will be TRUE.

### ⊙ Program demo

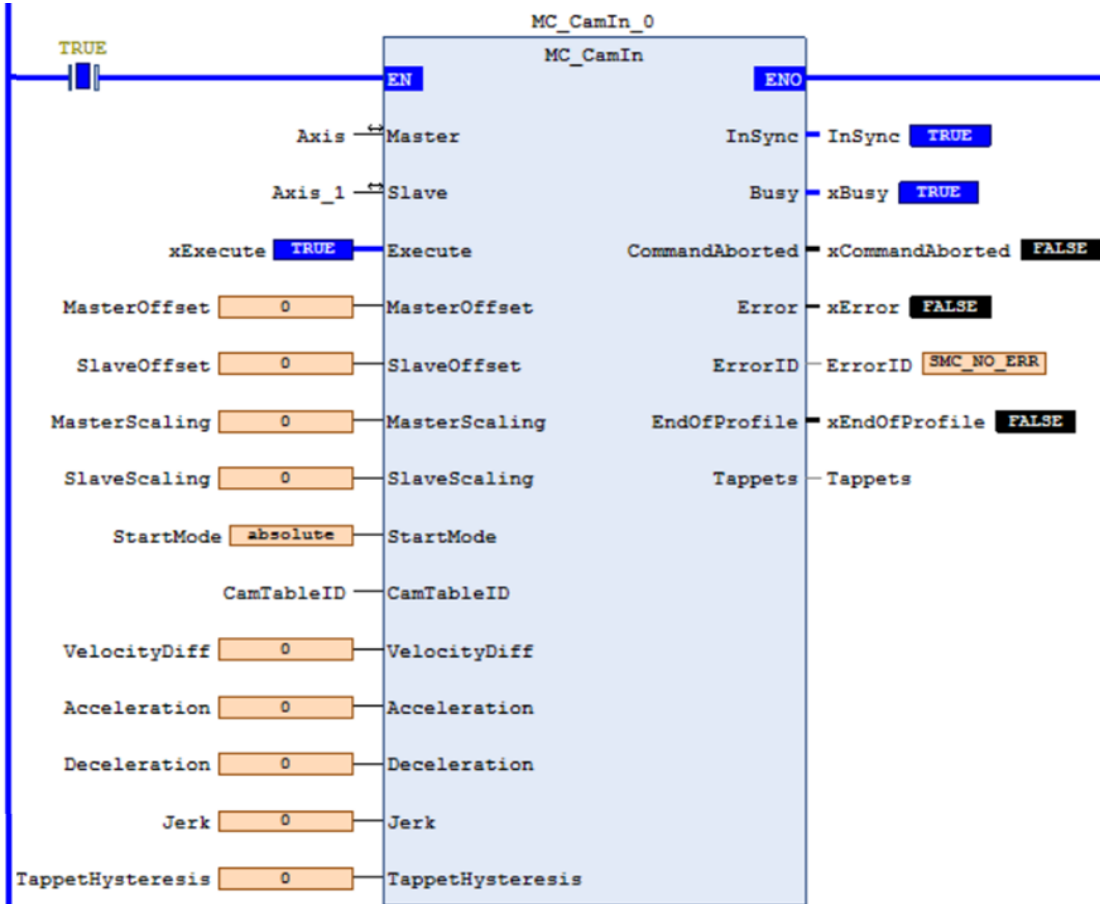
**ST:** When xExecute becomes TRUE, the electronic cam coupling instruction is executed.

```

MC_CamIn_0(
  Master:= Axis,
  Slave:= Axis_1,
  Execute TRUE := xExecute TRUE,
  MasterOffset 0 := MasterOffset 0,
  SlaveOffset 0 := SlaveOffset 0,
  MasterScaling 0 := MasterScaling 0,
  SlaveScaling 0 := SlaveScaling 0,
  StartMode absolute := StartMode absolute,
  CamTableID:= CamTableID,
  VelocityDiff 0 := VelocityDiff 0,
  Acceleration 0 := Acceleration 0,
  Deceleration 0 := Deceleration 0,
  Jerk 0 := Jerk 0,
  TappetHysteresis 0 := TappetHysteresis 0,
  InSync TRUE => xInSync TRUE,
  Busy TRUE => xBusy TRUE,
  CommandAborted FALSE => xCommandAborted FALSE,
  Error FALSE => xError FALSE,
  ErrorID SMC_NO_ERR => ErrorID SMC_NO_ERR,
  EndOfProfile FALSE => xEndOfProfile FALSE,
  Tappets=> Tappets);

```

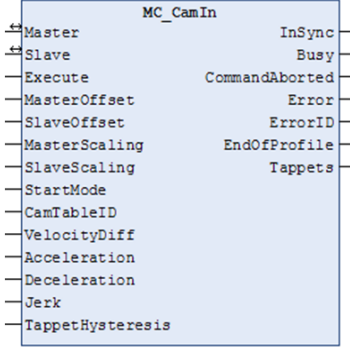
**LD:** When xExecute becomes TRUE, the electronic cam coupling instruction is executed.



### 4.3.7 MC\_CamOut

Use the specified cam table to start executing the electronic cam action.

#### ⊙ Command Format

Instruction	Name	FB/FC	LD	ST	File
MC_CamOut	E-cam disengagement instruction	FC		MC_CamIn( Master:=, Slave:=, Execute:=, MasterOffset:=, SlaveOffset:=, MasterScaling:=, SlaveScaling:=, StartMode:=, CamTableID:=, VelocityDiff:=, Acceleration:=, Deceleration:=, Jerk:=, TappetHysteresis:=, nSync=>, Busy=>, CommandAborted=>=, Error=>, ErrorID=>, EndOfProfile=>, Tappets=> );	SM3_Basic

#### ⊙ Related Variables

##### I/O variable

I/O variable	Name	Data type	Range	initialization	Descriptive
Slave	axle	AXIS_REF_SM3	-	-	Specify slave axis

##### Input variable

Input variable	Name	Data type	Range	initialization	Descriptive
Execute	Start	BOOL	TRUE-FALSE	FALSE	Rising edge trigger

##### Output variable

Output variable	Name	Data type	Range	initialization	Descriptive
Done	Implementation completed	BOOL	TRUE-FALSE	FALSE	TRUE:Electronic cam disconnection
Busy	Executing	BOOL	TRUE-FALSE	FALSE	TRUE: Processing of the function block is not completed
Error	Error	BOOL	TRUE-FALSE	FALSE	TRUE: An error occurred within the function block
ErrorID	Error Code	SMC_ERRO R	-	0	Error indication, see SMC Error.

	Boolean	Bit string					Integer							Real number		Moment, Duration, Date, String				
	BOOL	BYTE	WORD	DWORD	LWORD	USINT	UINT	UDINT	ULINT	SINT	INT	DINT	LINT	REAL	LREAL	TIME	DATE	TOD	DT	STRING
Slave		AXIS_REF_SM3																		
Execute	√	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
Done	√	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
Busy	√	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
Error	√	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
ErrorID		SMC_ERROR																		

**⊙ Functional Description**

This instruction is implemented by the "SM3\_Basic" library.

This instruction is used to uncouple the cams of the specified slave axis from its corresponding master axis.

**⊙ Program demo**

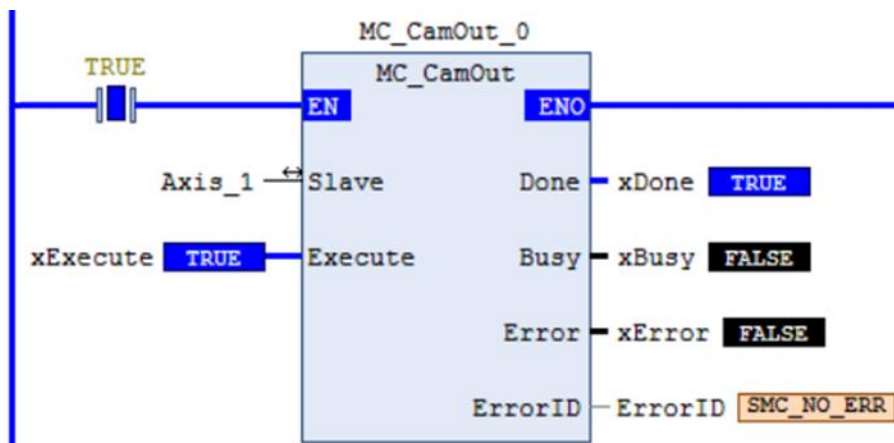
**ST:** When xExecute becomes TRUE, the electronic cam disengagement command is executed.

```

MC_CamOut_0 (
  Slave:= Axis_1,
  Execute TRUE := xExecute TRUE ,
  Done TRUE => xDone TRUE ,
  Busy FALSE => xBusy FALSE ,
  Error FALSE => xError FALSE ,
  ErrorID SMC_NO_ERR => ErrorID SMC_NO_ERR );

```

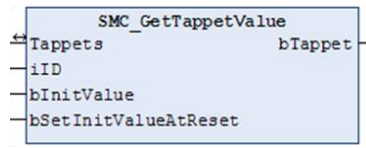
**LD:** When xExecute becomes TRUE, the electronic cam disengagement command is executed.



### 4.3.8 SMC\_GetTappetValue

Reading the current tappet status needs to be used in conjunction with the MC\_CamIn instruction.

#### ⊙ Command Format

Instruction	Name	FB/FC	LD	ST	File
SMC_GetTappetValue	Read tappet status instruction	FC		SMC_GetTappetValue( Tappets:=, iID:=, bInitValue:=, bSetInitValueAtReset:=, bTappet=> );	SM3_Basic

#### ⊙ Related Variables

##### I/O variable

I/O variable	Name	Data type	Range	initialization	Descriptive
Tappets	Tappet	SMC_TappetData	-	-	Tappet signal evaluated via MC_GetTappetValue, see SMC_TappetData

##### Input variable

Input variable	Name	Data type	Range	initialization	Descriptive
iID	Tappet group number	INT	Positive number	0	Tappet group ID to be evaluated
bInitValue	Initial value	BOOL	TRUE-FALSE	FALSE	Initial value of the tappet assigned at the first call
bSetInitValueAtReset	Tappet Reset	BOOL	TRUE-FALSE	FALSE	TRUE: The output value of the tappet will be set to the initial value when the CamIn function block is restarted. FALSE: The value of the tappet will be maintained when the CamIn function block is restarted.

##### Output variable

Output variable	Name	Data type	Range	initialization	Descriptive
bTappet	Tappet value	BOOL	TRUE /FALSE	FALSE	Tappet value

	Boolean	Bit string					Integer							Real number		Moment, Duration, Date, String				
	BOOL	BYTE	WORD	DWORD	LWORD	USINT	UINT	UDINT	ULINT	SINT	INT	DINT	LINT	REAL	LREAL	TIME	DATE	TOD	DT	STRING
Tappets		SMC_TappetData																		
iID	-	-	-	-	-	-	-	-	-	-	√	-	-	-	-	-	-	-	-	-
bInitValue	√	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
bSetInitValueAtReset	√	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
bTappet	√	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
Tappets	-	-	-	-	-	-	-	-	-	-	√	-	-	-	-	-	-	-	-	-
Busy	√	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-

#### ⊙ Functional Description

This instruction is implemented in the "SM3\_Basic" library.

In a mechanical cam structure, the rotary motion of the cam is converted to vertical motion by a tappet. The tappet of an electronic cam is similar to a mechanical tappet and is equivalent to an electronic mechanical tappet. An electronic cam tappet is a cam spindle that gives a high or low level at a defined position, which is used as a switch. This function block needs to be used in conjunction with the MC\_CamIn instruction to get the value of the tappet managed by the cam table.

☉ **Program demo**

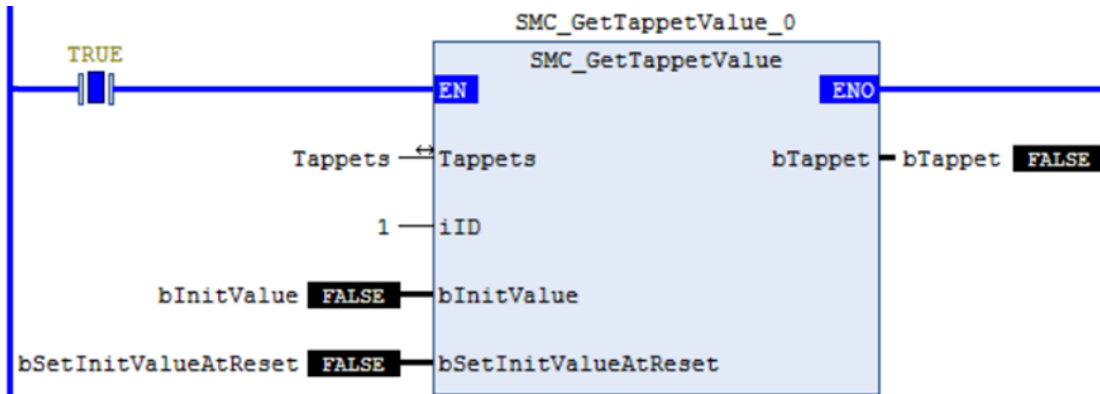
**ST:** When xExecute becomes TRUE, the Read Tappet Status command is executed.

```

SMC_GetTappetValue_0(
Tappets:= Tappets,
iID 1 := 1,
bInitValue FALSE := bInitValue FALSE,
bSetInitValueAtReset FALSE := bSetInitValueAtReset FALSE,
bTappet FALSE => bTappet FALSE );

```

**LD:** When xExecute becomes TRUE, the Read Tappet Status command is executed.



## 4.4 Axis synchronization function

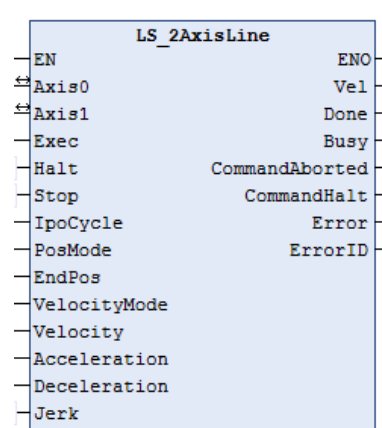
### 4.4.1 Instruction List

Instruction Type	Name	FB/FC	Functionality
Interpolation	LS_2AxisLine	FB	Two axis linear interpolation
	LS_3AxisLine	FB	Three axis linear interpolation
	LS_4AxisLine	FB	Four axis linear interpolation
	LS_5AxisLine	FB	Five axis linear interpolation
	LS_6AxisLine	FB	Six axis linear interpolation
	LS_2AxisLineA_Ratio	FB	Two axis Variable speed linear interpolation
	LS_LineFollow	FB	Follow the motion
	LS_2AxisCircle	FB	Two axis circular interpolation
	LS_3AxisCircle	FB	Three axis circular interpolation
	LS_2AxisEllipses	FB	Two axis elliptic interpolation
	LS_2AxisCircle_Helical	FB	Helical interpolation
	LS_3AxisMoveSequence	FB	Three axis continuous interpolation motion
	LS_4AxisMoveSequence	FB	Four axis continuous interpolation motion
LS_6AxisMoveSequence	FB	Six axis continuous interpolation motion	

### 4.4.2 LS\_2AxisLine

Two axis linear interpolation instructions.

#### ⊙ Command Format

Instruction	Name	FB/FC	LD	ST	File
LS_2AxisLine	Two axis linear interpolation instruction	FC		<pre> LS_2AxisLine( Axis0:=, Axis1:=, Exec:=, Halt:=, Stop:=, IpoCycle:=, PosMode:=, EndPos:=, VelocityMode:=, Velocity:=, Acceleration:=, Deceleration:=, Jerk:=, Vel=&gt;, Done=&gt;, Busy=&gt;, CommandAborted=&gt;, CommandHalt=&gt;, Error=&gt;, ErrorID=&gt;); </pre>	LS_IpoLib

#### ⊙ Related Variables

##### I/O variable

I/O variable	Name	Data type	Range	initialization	Descriptive
Axis0	Axis0	AXIS_REF_VIRTUAL_SM3	-	-	The 0-axis involved in interpolation
Axis1	Axis1	AXIS_REF_VIRTUAL_SM3	-	-	The 1-axis involved in interpolation

##### Input variable

Input variable	Name	Data type	Range	initialization	Descriptive
Exec	Start	BOOL	TRUE-FALSE	FALSE	Rising edge trigger

Halt	Pause	BOOL	TRUE-FALSE	FALSE	TRUE: the interpolation motion is suspended; when the state is switched to False again, the interpolation module continues the previous unfinished interpolation tasks
Stop	Stop	BOOL	TRUE-FALSE	FALSE	TRUE: The interpolation module is stopped.
IpoCycle	Interpolation cycle	DWORD	ALL	2000	Interpolation period, unit: us.
PosMode	Position Mode	INT	[0,1]	0	Positional mode: 0 Absolute mode; 1 Relative mode.
Endpos	Target position	ARRAY[0...1] OF REAL	ALL	0	Target position of the axis. Unit: pulse
VelocityMode	Velocity mode	SMC_INT_VE LMODE	0-3	SIGMOID	Velocity mode: Trapezoidal: 0 (TRAPEZOID); S-shaped: 1 (SIGMOID); Quadratic: 3 (QUADRATIC).
Velocity	Target velocity	LREAL	Is always positive.	0	Interpolate synthetic velocity.
Acceleration	Target acceleration	LREAL	Is always positive.	0	Interpolate synthetic acceleration.
Deceleration	Target Deceleration	LREAL	Is always positive.	0	Interpolate the synthetic deceleration.
Jaek	Target jaek	LREAL	ALL	90000000	Acceleration, when the speed mode is 3, you need to set this parameter, the value can not be 0.

**Output variable**

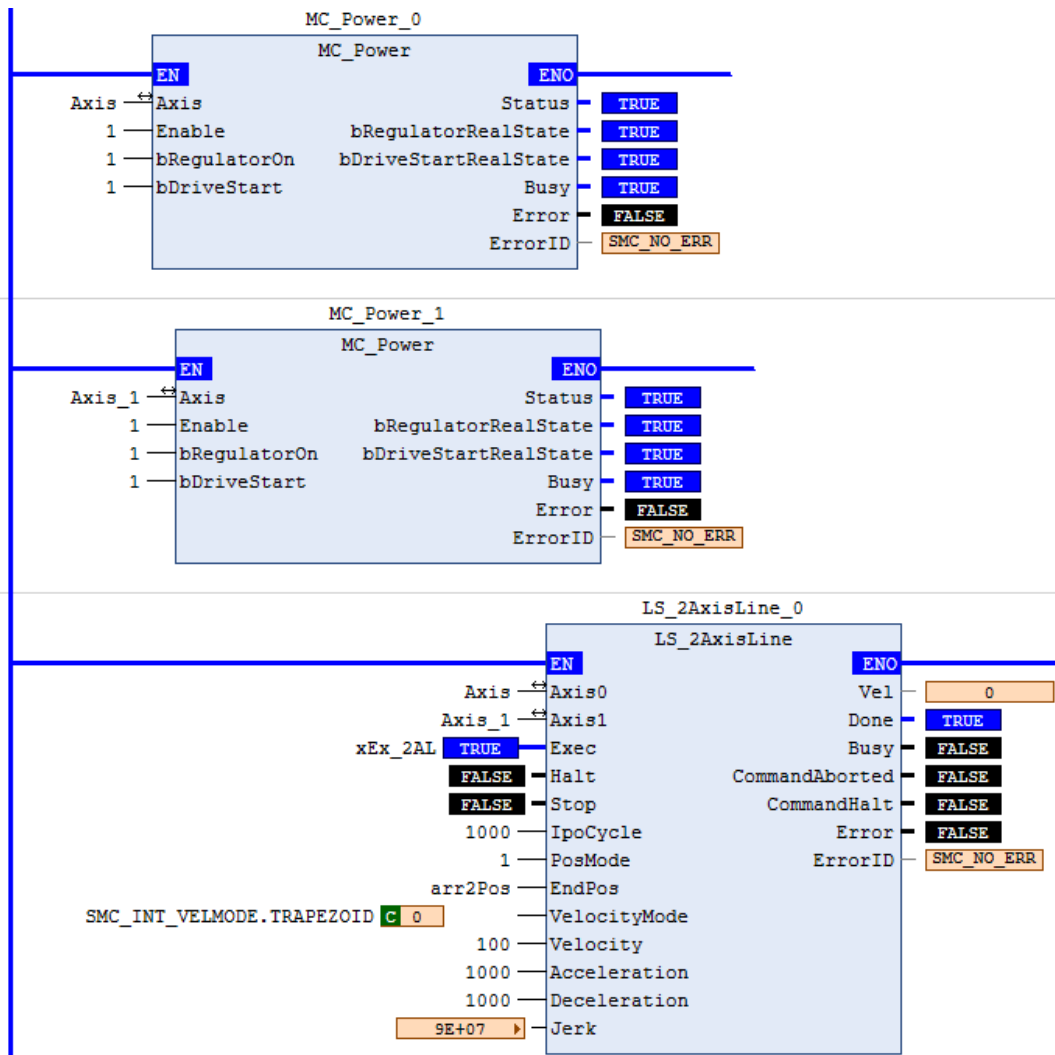
Output variable	Name	Data type	Range	initialization	Descriptive
Vel	Closing speed	LREAL	0, positive number	0	Kinematic synthetic velocity
Done	Completion	BOOL	TRUE-FALSE	FALSE	TRUE: Homing is done.
Busy	Executing	BOOL	TRUE-FALSE	FALSE	TRUE: In progress
Command Aborted	Command Interrupt	BOOL	TRUE-FALSE	FALSE	TRUE: Terminated by other commands
CommandHalt	motion pause	BOOL	TRUE, FALSE	FALSE	When Halt is True, this variable is True and the interpolation module is in a suspended state.
Error	Error	BOOL	TRUE-FALSE	FALSE	TRUE: An error occurred within the function block
ErrorID	Error Code	SMC_ERRO R	-	0	Error indication, see SMC_Error.

	Bo le an	Bit string				Integer							Real number		Moment, Duration, Date, String						
	BOOL	BYTE	WORD	DWORD	LWORD	USINT	UINT	UDINT	ULINT	SINT	INT	DINT	LINT	REAL	LREAL	TIME	DATE	TOD	DT	STRING	
Axis																					
Exec	√	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
Halt	√	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-

Stop	√	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
Endpos	-	-	-	-	-	-	-	-	-	-	-	-	-	-	√	-	-	-	-	-
Velocity	-	-	-	-	-	-	-	-	-	-	-	-	-	-	√	-	-	-	-	-
Acceleration	-	-	-	-	-	-	-	-	-	-	-	-	-	-	√	-	-	-	-	-
Deceleration	-	-	-	-	-	-	-	-	-	-	-	-	-	-	√	-	-	-	-	-
Jerk	-	-	-	-	-	-	-	-	-	-	-	-	-	-	√	-	-	-	-	-
Done	√	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
Busy	√	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
Command Aboned	√	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
Error	√	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
ErrorID	SMC_ERROR																			

**⊙ Program demo**

**LD:** The two axes move in relative mode with linear interpolation to the target position 1000, 1000.

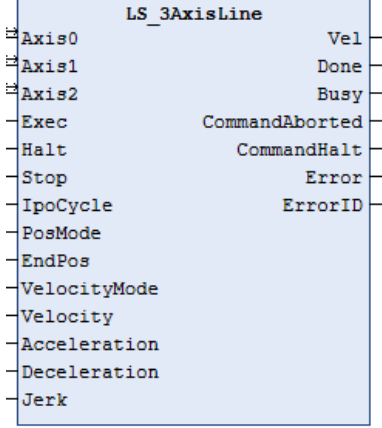

**Note**

The IpoCycle parameter needs to be aligned with the task cycle time where the motion instruction is located, otherwise an Error may be reported when the instruction is executed.  
 The parameters IpoCycle, Jerk, Velocity, Acceleration, Deceleration cannot be set to 0.  
 During the instruction running period, neither master nor slave axis can be called by other motion instructions.

### 4.4.3 LS\_3AxisLine

Three-axis linear interpolation commands.

#### ⊙ Command Format

Instruction	Name	FB/FC	LD	ST	File
LS_3AxisLine	Three axis linear interpolation instruction	FC		<pre> LS_3AxisLine( Axis0:=, Axis1:=, Axis2:=, Exec:=, Halt:=, Stop:=, IpoCycle:=, PosMode:=, EndPos:=, VelocityMode:=, Velocity:=, Acceleration:=, Deceleration:=, Jerk:=, Vel=&gt;, Done=&gt;, Busy=&gt;, CommandAborted=&gt;, CommandHalt=&gt;, Error=&gt;, ErrorID=&gt;); </pre>	LS_IpoLib

#### ⊙ Related Variables

##### I/O variable

I/O variable	Name	Data type	Range	initialization	Descriptive
Axis0	Axis0	AXIS_REF_ VIRTUAL_SM3	-	-	The 0-axis involved in interpolation
Axis1	Axis1	AXIS_REF_ VIRTUAL_SM3	-	-	The 1-axis involved in interpolation
Axis2	Axis2	AXIS_REF_ VIRTUAL_SM3	-	-	The 2-axis involved in interpolation

##### Input variable

Input variable	Name	Data type	Range	initialization	Descriptive
Exec	Start	BOOL	TRUE-FALSE	FALSE	Rising edge trigger
Halt	Pause	BOOL	TRUE-FALSE	FALSE	TRUE: the interpolation motion is suspended; when the state is switched to False again, the interpolation module continues the previous unfinished interpolation tasks
Stop	Stop	BOOL	TRUE-FALSE	FALSE	TRUE: The interpolation module is stopped.
IpoCycle	Interpolation cycle	DWORD	ALL	2000	Interpolation period, unit: us.
PosMode	Position Mode	INT	[0,1]	0	Positional mode: 0 Absolute mode; 1 Relative mode.
Endpos	Target position	ARRAY[0...1] OF REAL	ALL	0	Target position of the axis. Unit: pulse
VelocityMode	Velocity mode	SMC_INT_VE LMODE	0-3	SIGMOID	Velocity mode: Trapezoidal: 0 (TRAPEZOID); S-shaped: 1 (SIGMOID); Quadratic: 3 (QUADRATIC).

Velocity	Target velocity	LREAL	Is always positive.	0	Interpolate synthetic velocity.
Acceleration	Target acceleration	LREAL	Is always positive.	0	Interpolate synthetic acceleration.
Deceleration	Target Deceleration	LREAL	Is always positive.	0	Interpolate the synthetic deceleration.
Jack	Target jack	LREAL	ALL	90000000	Acceleration, when the speed mode is 3, you need to set this parameter, the value can not be 0.

**Output variable**

Output variable	Name	Data type	Range	initialization	Descriptive
Vel	Closing speed	LREAL	0,positive number	0	Kinematic synthetic velocity
Done	Completion	BOOL	TRUE-FALSE	FALSE	TRUE: Homing is done.
Busy	Executing	BOOL	TRUE-FALSE	FALSE	TRUE: In progress
Command Aborted	Command Interrupt	BOOL	TRUE-FALSE	FALSE	TRUE: Terminated by other commands
CommandHalt	motion pause	BOOL	TRUE, FALSE	FALSE	When Halt is True, this variable is True and the interpolation module is in a suspended state.
Error	Error	BOOL	TRUE-FALSE	FALSE	TRUE: An error occurred within the function block
ErrorID	Error Code	SMC_ERRO R	-	0	Error indication, see SMC Error.

	Bo le an	Bit string					Integer							Real number		Moment, Duration, Date, String				
	BOOL	BYTE	WORD	DWORD	LWORD	USINT	UINT	UDINT	ULINT	SINT	INT	DINT	LINT	REAL	LREAL	TIME	DATE	TOD	DT	STRING
Axis		AXIS_REF_SM3																		
Exec	√	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
Halt	√	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
Stop	√	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
Endpos	-	-	-	-	-	-	-	-	-	-	-	-	-	-	√	-	-	-	-	-
Velocity	-	-	-	-	-	-	-	-	-	-	-	-	-	-	√	-	-	-	-	-
Acceleration	-	-	-	-	-	-	-	-	-	-	-	-	-	-	√	-	-	-	-	-
Deceleration	-	-	-	-	-	-	-	-	-	-	-	-	-	-	√	-	-	-	-	-
Jerk	-	-	-	-	-	-	-	-	-	-	-	-	-	-	√	-	-	-	-	-
Done	√	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
Busy	√	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
Command Aboned	√	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
Error	√	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
ErrorID		SMC_ERROR																		

**⊙ Program demo**

**LD:** The three axes move in relative mode with linear interpolation to the target position 1000, 1000, 1000.

表达式	类型	值
⊕  LS_3AxisLine_0	LS_3AxisLine	
xEx	BOOL	TRUE
⊖  arrPos	ARRAY [0..2] OF LREAL	
arrPos[0]	LREAL	1000
arrPos[1]	LREAL	1000
arrPos[2]	LREAL	1000
⊕  MC_Power_0	MC_Power	
⊕  MC_Power_1	MC_Power	

```

1  ● ACT();
2
3  ● LS_3AxisLine_0(
4      Axis0:= axis,
5      Axis1:= axis_1,
6      Axis2:= axis_2,
7      ExecTRUE:= xExTRUE,
8      Halt:= ,
9      Stop:= ,
10     IpoCycle 1000 := 1000,
11     PosMode 1 := 1,
12     EndPos:= arrPos,
13     VelocityMode SIGMOID := SMC_INT_VELMODE.SIGMOID,
14     Velocity 50 := 50,
15     Acceleration 1E+03 := 1000,
16     Deceleration 1E+03 := 1000,
17     Jerk 1E+04 := 10000,
18     Vel=> ,
19     Done=> ,
20     Busy=> ,
21     CommandAborted=> ,
22     CommandHalt=> ,
23     Error=> ,
24     ErrorID=> );
--

```

**Note**

The IpoCycle parameter needs to be aligned with the task cycle time where the motion instruction is located, otherwise an Error may be reported when the instruction is executed.

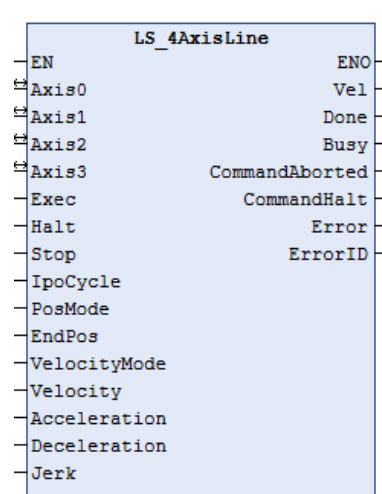
The parameters IpoCycle, Jerk, Velocity, Acceleration, Deceleration cannot be set to 0.

During the instruction running period, neither master nor slave axis can be called by other motion instructions.

### 4.4.4 LS\_4AxisLine

Three-axis linear interpolation commands.

#### ⊙ Command Format

Instruction	Name	FB/FC	LD	ST	File
LS_4AxisLine	Four axis linear interpolation instruction	FC		<pre>LS_4AxisLine( Axis0:=, Axis1:=, Axis2:=, Axis3:=, Exec:=, Halt:=, Stop:=, IpoCycle:=, PosMode:=, EndPos:=, VelocityMode:=, Velocity:=, Acceleration:=, Deceleration:=, Jerk:=, Vel=&gt;, Done=&gt;, Busy=&gt;, CommandAborted=&gt;, CommandHalt=&gt;, Error=&gt;, ErrorID=&gt;);</pre>	LS_IpoLib

#### ⊙ Related Variables

##### I/O variable

I/O variable	Name	Data type	Range	initialization	Descriptive
Axis0	Axis0	AXIS_REF_ VIRTUAL_SM3	-	-	The 0-axis involved in interpolation
Axis1	Axis1	AXIS_REF_ VIRTUAL_SM3	-	-	The 1-axis involved in interpolation
Axis2	Axis2	AXIS_REF_ VIRTUAL_SM3	-	-	The 2-axis involved in interpolation
Axis3	Axis3	AXIS_REF_ VIRTUAL_SM3	-	-	The 3-axis involved in interpolation

##### Input variable

Input variable	Name	Data type	Range	initialization	Descriptive
Exec	Start	BOOL	TRUE-FALSE	FALSE	Rising edge trigger
Halt	Pause	BOOL	TRUE-FALSE	FALSE	TRUE: the interpolation motion is suspended; when the state is switched to False again, the interpolation module continues the previous unfinished interpolation tasks
Stop	Stop	BOOL	TRUE-FALSE	FALSE	TRUE: The interpolation module is stopped.
IpoCycle	Interpolation cycle	DWORD	ALL	2000	Interpolation period, unit: us.
PosMode	Position Mode	INT	[0,1]	0	Positional mode: 0 Absolute mode; 1 Relative mode.
Endpos	Target position	ARRAY[0...1] OF REAL	ALL	0	Target position of the axis. Unit: pulse
VelocityMode	Velocity mode	SMC_INT_VE LMODE	0-3	SIGMOID	Velocity mode: Trapezoidal: 0 (TRAPEZOID);

					S-shaped: 1 (SIGMOID); Quadratic: 3 (QUADRATIC).
Velocity	Target velocity	LREAL	Is always positive.	0	Interpolate synthetic velocity.
Acceleration	Target acceleration	LREAL	Is always positive.	0	Interpolate synthetic acceleration.
Deceleration	Target Deceleration	LREAL	Is always positive.	0	Interpolate the synthetic deceleration.
Jaek	Target jaek	LREAL	ALL	90000000	Acceleration, when the speed mode is 3, you need to set this parameter, the value can not be 0.

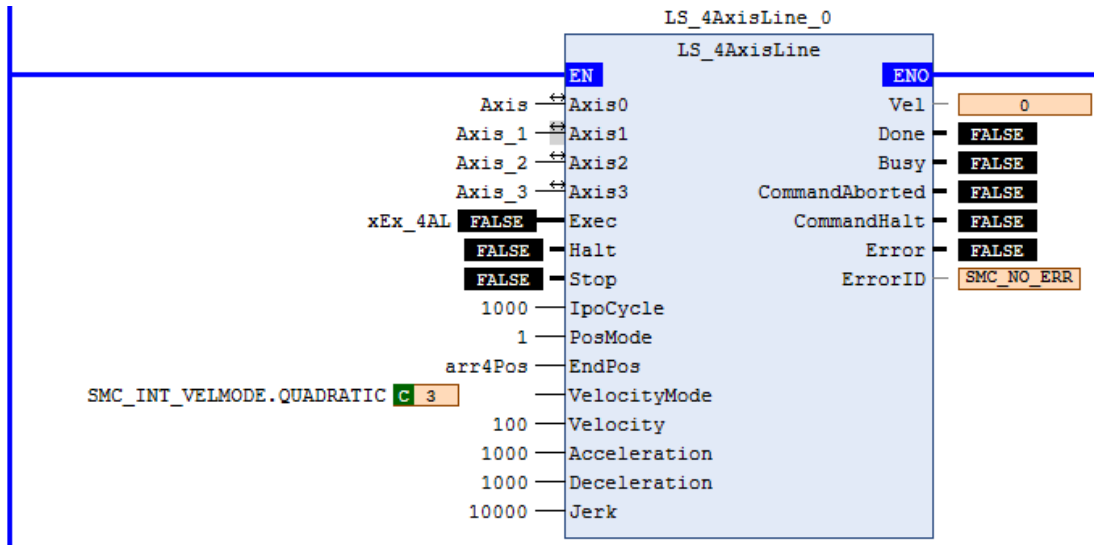
**Output variable**

Output variable	Name	Data type	Range	initialization	Descriptive
Vel	Closing speed	LREAL	0,positive number	0	Kinematic synthetic velocity
Done	Completion	BOOL	TRUE-FALSE	FALSE	TRUE: Homing is done.
Busy	Executing	BOOL	TRUE-FALSE	FALSE	TRUE: In progress
Command Aborted	Command Interrupt	BOOL	TRUE-FALSE	FALSE	TRUE: Terminated by other commands
CommandHalt	motion pause	BOOL	TRUE, FALSE	FALSE	When Halt is True, this variable is True and the interpolation module is in a suspended state.
Error	Error	BOOL	TRUE-FALSE	FALSE	TRUE: An error occurred within the function block
ErrorID	Error Code	SMC_ERROR	-	0	Error indication, see SMC_Error.

	Bo le an	Bit string					Integer						Real number		Moment, Duration, Date, String					
	BOOL	BYTE	WORD	DWORD	LWORD	USINT	UINT	UDINT	ULINT	SINT	INT	DINT	LINT	REAL	LREAL	TIME	DATE	TOD	DT	STRING
Axis		AXIS_REF_SM3																		
Exec	√	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
Halt	√	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
Stop	√	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
Endpos	-	-	-	-	-	-	-	-	-	-	-	-	-	√	-	-	-	-	-	-
Velocity	-	-	-	-	-	-	-	-	-	-	-	-	-	√	-	-	-	-	-	-
Acceleration	-	-	-	-	-	-	-	-	-	-	-	-	-	√	-	-	-	-	-	-
Deceleration	-	-	-	-	-	-	-	-	-	-	-	-	-	√	-	-	-	-	-	-
Jerk	-	-	-	-	-	-	-	-	-	-	-	-	-	√	-	-	-	-	-	-
Done	√	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
Busy	√	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
Command Aboned	√	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
Error	√	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
ErrorID		SMC_ERROR																		

**Ⓞ Program demo**

**LD:** 4-axis in relative mode, linear interpolation motion to target position 1000, 2000, 3000, 4000. variable definition: Arr4Pos: ARRAY [0..3] OF LREAL:=[1000,2000,3000,4000];.


**Note**

The IpoCycle parameter needs to be aligned with the task cycle time where the motion instruction is located, otherwise an Error may be reported when the instruction is executed.

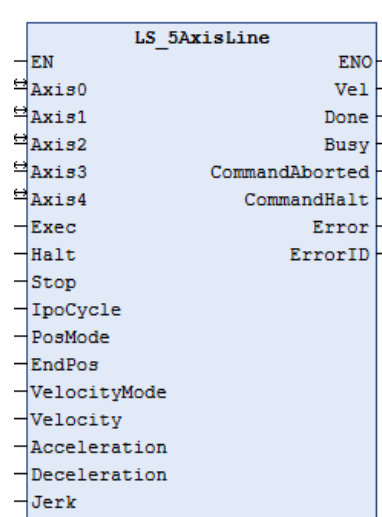
The parameters IpoCycle, Jerk, Velocity, Acceleration, Deceleration cannot be set to 0.

During the instruction running period, neither master nor slave axis can be called by other motion instructions

### 4.4.5 LS\_5AxisLine

Three-axis linear interpolation commands.

#### ⊙ Command Format

Instruction	Name	FB/FC	LD	ST	File
LS_5AxisLine	Five axis linear interpolation instruction	FC		<pre> LS_5AxisLine( Axis0:=, Axis1:=, Axis2:=, Axis3:=, Axis4:=, Exec:=, Halt:=, Stop:=, IpoCycle:=, PosMode:=, EndPos:=, VelocityMode:=, Velocity:=, Acceleration:=, Deceleration:=, Jerk:=, Vel=&gt;, Done=&gt;, Busy=&gt;, CommandAborted=&gt;, CommandHalt=&gt;, Error=&gt;, ErrorID=&gt;); </pre>	LS_IpoLib

#### ⊙ Related Variables

##### I/O variable

I/O variable	Name	Data type	Range	initialization	Descriptive
Axis0	Axis0	AXIS_REF_ VIRTUAL_SM3	-	-	The 0-axis involved in interpolation
Axis1	Axis1	AXIS_REF_ VIRTUAL_SM3	-	-	The 1-axis involved in interpolation
Axis2	Axis2	AXIS_REF_ VIRTUAL_SM3	-	-	The 2-axis involved in interpolation
Axis3	Axis3	AXIS_REF_ VIRTUAL_SM3	-	-	The 3-axis involved in interpolation
Axis4	Axis4	AXIS_REF_ VIRTUAL_SM3	-	-	The 4-axis involved in interpolation

##### Input variable

Input variable	Name	Data type	Range	initialization	Descriptive
Exec	Start	BOOL	TRUE-FALSE	FALSE	Rising edge trigger
Halt	Pause	BOOL	TRUE-FALSE	FALSE	TRUE: the interpolation motion is suspended; when the state is switched to False again, the interpolation module continues the previous unfinished interpolation tasks
Stop	Stop	BOOL	TRUE-FALSE	FALSE	TRUE: The interpolation module is stopped.
IpoCycle	Interpolation cycle	DWORD	ALL	2000	Interpolation period, unit: us.
PosMode	Position Mode	INT	[0,1]	0	Positional mode: 0 Absolute mode; 1 Relative mode.
Endpos	Target position	ARRAY[0...1] OF REAL	ALL	0	Target position of the axis. Unit: pulse

VelocityMode	Velocity mode	SMC_INT_VE LMODE	0-3	SIGMOID	Velocity mode: Trapezoidal: 0 (TRAPEZOID); S-shaped: 1 (SIGMOID); Quadratic: 3 (QUADRATIC).
Velocity	Target velocity	LREAL	Is always positive.	0	Interpolate synthetic velocity.
Acceleration	Target acceleration	LREAL	Is always positive.	0	Interpolate synthetic acceleration.
Deceleration	Target Deceleration	LREAL	Is always positive.	0	Interpolate the synthetic deceleration.
Jack	Target jack	LREAL	ALL	90000000	Acceleration, when the speed mode is 3, you need to set this parameter, the value can not be 0.

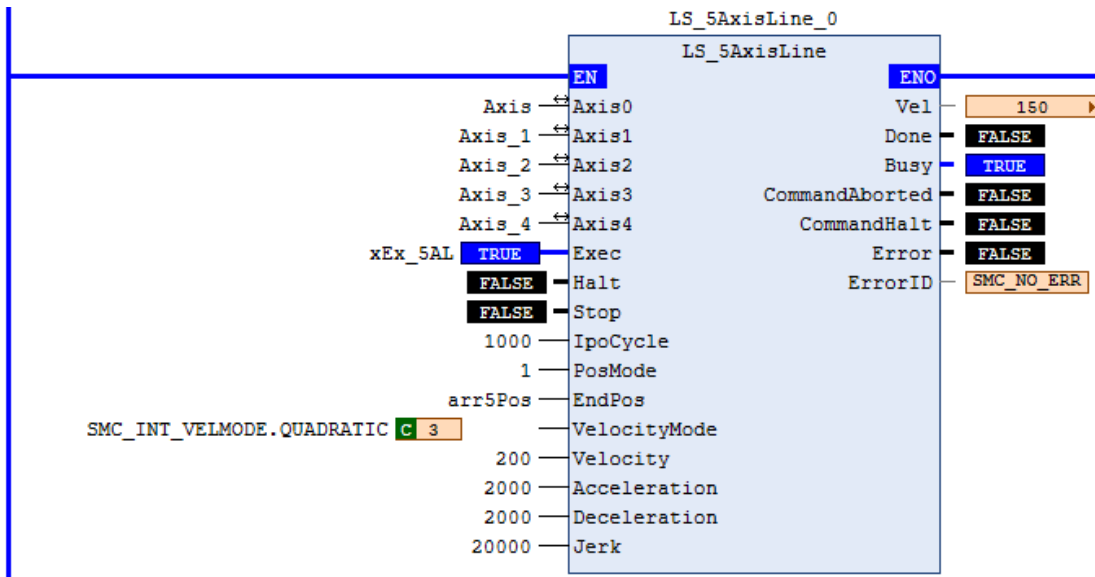
**Output variable**

Output variable	Name	Data type	Range	initialization	Descriptive
Vel	Closing speed	LREAL	0, positive number	0	Kinematic synthetic velocity
Done	Completion	BOOL	TRUE-FALSE	FALSE	TRUE: Homing is done.
Busy	Executing	BOOL	TRUE-FALSE	FALSE	TRUE: In progress
Command Aborted	Command Interrupt	BOOL	TRUE-FALSE	FALSE	TRUE: Terminated by other commands
CommandHalt	motion pause	BOOL	TRUE, FALSE	FALSE	When Halt is True, this variable is True and the interpolation module is in a suspended state.
Error	Error	BOOL	TRUE-FALSE	FALSE	TRUE: An error occurred within the function block
ErrorID	Error Code	SMC_ERRO R	-	0	Error indication, see SMC Error.

	Bo le an	Bit string					Integer						Real number		Moment, Duration, Date, String					
	BOOL	BYTE	WORD	DWORD	LWORD	USINT	UINT	UDINT	ULINT	SINT	INT	DINT	LINT	REAL	LREAL	TIME	DATE	TOD	DT	STRING
Axis		AXIS_REF_SM3																		
Exec	√	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
Halt	√	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
Stop	√	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
Endpos	-	-	-	-	-	-	-	-	-	-	-	-	-	√	-	-	-	-	-	-
Velocity	-	-	-	-	-	-	-	-	-	-	-	-	-	√	-	-	-	-	-	-
Acceleration	-	-	-	-	-	-	-	-	-	-	-	-	-	√	-	-	-	-	-	-
Deceleration	-	-	-	-	-	-	-	-	-	-	-	-	-	√	-	-	-	-	-	-
Jerk	-	-	-	-	-	-	-	-	-	-	-	-	-	√	-	-	-	-	-	-
Done	√	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
Busy	√	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
Command Aboned	√	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
Error	√	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
ErrorID		SMC_ERROR																		

**Ⓞ Program demo**

**LD:** 5-axis in relative mode, linear interpolation running to target position 1000, 2000, 3000, 4000, 5000.  
 variable definition: arr5Pos: ARRAY [0..4] OF LREAL:= [1000,2000,3000,4000,5000];.


**Note**

The IpoCycle parameter needs to be aligned with the task cycle time where the motion instruction is located, otherwise an Error may be reported when the instruction is executed.

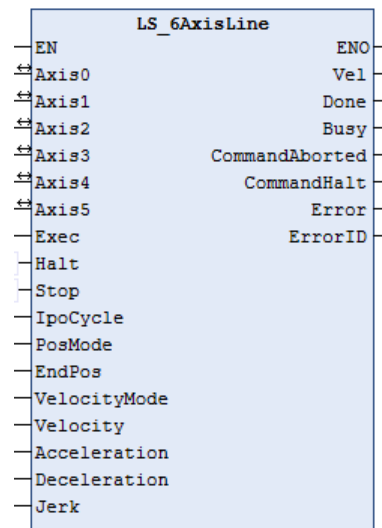
The parameters IpoCycle, Jerk, Velocity, Acceleration, Deceleration cannot be set to 0.

During the instruction running period, neither master nor slave axis can be called by other motion instructions

### 4.4.6 LS\_6AxisLine

Three-axis linear interpolation commands.

#### ⊙ Command Format

Instruction	Name	FB/FC	LD	ST	File
LS_6AxisLine	Six axis linear interpolation instruction	FC		<pre> LS_6AxisLine( Axis0:=, Axis1:=, Axis2:=, Axis3:=, Axis4:=, Axis5:=, Exec:=, Halt:=, Stop:=, IpoCycle:=, PosMode:=, EndPos:=, VelocityMode:=, Velocity:=, Acceleration:=, Deceleration:=, Jerk:=, Vel=&gt;, Done=&gt;, Busy=&gt;, CommandAborted=&gt;, CommandHalt=&gt;, Error=&gt;, ErrorID=&gt;); </pre>	LS_IpoLib

#### ⊙ Related Variables

##### I/O variable

I/O variable	Name	Data type	Range	initialization	Descriptive
Axis0	Axis0	AXIS_REF_ VIRTUAL_SM3	-	-	The 0-axis involved in interpolation
Axis1	Axis1	AXIS_REF_ VIRTUAL_SM3	-	-	The 1-axis involved in interpolation
Axis2	Axis2	AXIS_REF_ VIRTUAL_SM3	-	-	The 2-axis involved in interpolation
Axis3	Axis3	AXIS_REF_ VIRTUAL_SM3	-	-	The 3-axis involved in interpolation
Axis4	Axis4	AXIS_REF_ VIRTUAL_SM3	-	-	The 4-axis involved in interpolation
Axis5	Axis5	AXIS_REF_ VIRTUAL_SM3	-	-	The 5-axis involved in interpolation

##### Input variable

Input variable	Name	Data type	Range	initialization	Descriptive
Exec	Start	BOOL	TRUE-FALSE	FALSE	Rising edge trigger
Halt	Pause	BOOL	TRUE-FALSE	FALSE	TRUE: the interpolation motion is suspended; when the state is switched to False again, the interpolation module continues the previous unfinished interpolation tasks
Stop	Stop	BOOL	TRUE-FALSE	FALSE	TRUE: The interpolation module is stopped.
IpoCycle	Interpolation cycle	DWORD	ALL	2000	Interpolation period, unit: us.
PosMode	Position Mode	INT	[0,1]	0	Positional mode: 0 Absolute mode; 1 Relative mode.
Endpos	Target position	ARRAY[0...1]	ALL	0	Target position of the axis.

		OF REAL			Unit: pulse
VelocityMode	Velocity mode	SMC_INT_VE LMODE	0-3	SIGMOID	Velocity mode: Trapezoidal: 0 (TRAPEZOID); S-shaped: 1 (SIGMOID); Quadratic: 3 (QUADRATIC).
Velocity	Target velocity	LREAL	Is always positive.	0	Interpolate synthetic velocity.
Acceleration	Target acceleration	LREAL	Is always positive.	0	Interpolate synthetic acceleration.
Deceleration	Target Deceleration	LREAL	Is always positive.	0	Interpolate the synthetic deceleration.
Jack	Target jack	LREAL	ALL	90000000	Acceleration, when the speed mode is 3, you need to set this parameter, the value can not be 0.

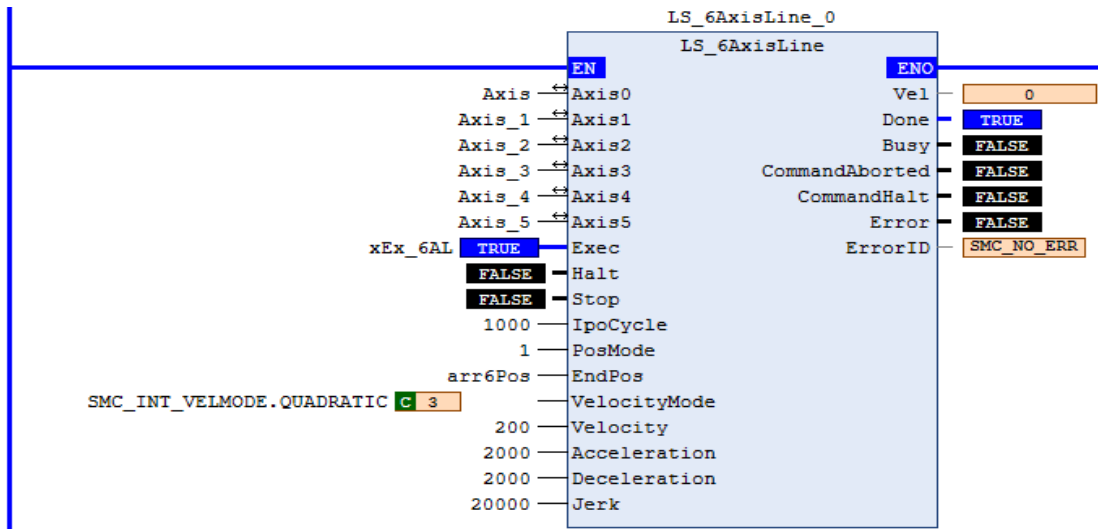
**Output variable**

Output variable	Name	Data type	Range	initialization	Descriptive
Vel	Closing speed	LREAL	0, positive number	0	Kinematic synthetic velocity
Done	Completion	BOOL	TRUE-FALSE	FALSE	TRUE: Homing is done.
Busy	Executing	BOOL	TRUE-FALSE	FALSE	TRUE: In progress
Command Aborted	Command Interrupt	BOOL	TRUE-FALSE	FALSE	TRUE: Terminated by other commands
CommandHalt	motion pause	BOOL	TRUE, FALSE	FALSE	When Halt is True, this variable is True and the interpolation module is in a suspended state.
Error	Error	BOOL	TRUE-FALSE	FALSE	TRUE: An error occurred within the function block
ErrorID	Error Code	SMC_ERRO R	-	0	Error indication, see SMC_Error.

	Bo ole an	Bit string					Integer							Real number		Moment, Duration, Date, String				
	BOOL	BYTE	WORD	DWORD	LWORD	USINT	UINT	UDINT	ULINT	SINT	INT	DINT	LINT	REAL	LREAL	TIME	DATE	TOD	DT	STRING
Axis		AXIS_REF_SM3																		
Exec	√	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
Halt	√	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
Stop	√	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
Endpos	-	-	-	-	-	-	-	-	-	-	-	-	-	√	-	-	-	-	-	-
Velocity	-	-	-	-	-	-	-	-	-	-	-	-	-	√	-	-	-	-	-	-
Acceleration	-	-	-	-	-	-	-	-	-	-	-	-	-	√	-	-	-	-	-	-
Deceleration	-	-	-	-	-	-	-	-	-	-	-	-	-	√	-	-	-	-	-	-
Jerk	-	-	-	-	-	-	-	-	-	-	-	-	-	√	-	-	-	-	-	-
Done	√	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
Busy	√	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
Command Aboned	√	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
Error	√	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
ErrorID		SMC_ERROR																		

**Ⓞ Program demo**

**LD:** The six axes are run in relative mode with linear interpolation to target positions 1000, 2000, 3000, 4000, 5000, 6000. variable definition: arr6Pos: ARRAY [0..5] OF LREAL:= [1000,2000,3000,4000,5000,6000];


**Note**

The IpoCycle parameter needs to be aligned with the task cycle time where the motion instruction is located, otherwise an Error may be reported when the instruction is executed.

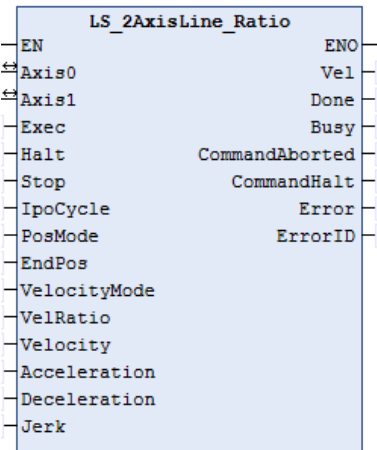
The parameters IpoCycle, Jerk, Velocity, Acceleration, Deceleration cannot be set to 0.

During the instruction running period, neither master nor slave axis can be called by other motion instructions

### 4.4.7 LS\_2AxisLineA\_Ratio

Two axes adjustable speed linear interpolation instruction. Compared with LS\_2AxisLine, the adjustable speed function is added, i.e., the interpolation speed can be changed online by changing the value of VelRatio during the interpolation process.

#### ⊙ Command Format

Instruction	Name	FB/FC	LD	ST	File
LS_2AxisLineA_Ratio	Two axis Variable speed linear interpolation instruction	FC		LS_2AxisLine_Ratio( Axis0:= , Axis1:= , Exec:= , Halt:= , Stop:= , IpoCycle:= , PosMode:= , EndPos:= , VelocityMode:= , VelRatio:= , Velocity:= , Acceleration:= , Deceleration:= , Jerk:= , Vel=> , Done=> , Busy=> , CommandAborted=> , CommandHalt=> , Error=> , ErrorID=> );	LS_IpoLib

#### ⊙ Related Variables

##### I/O variable

I/O variable	Name	Data type	Range	initialization	Descriptive
Axis0	Axis0	AXIS_REF_VIRTUAL_SM3	-	-	The 0-axis involved in interpolation
Axis1	Axis1	AXIS_REF_VIRTUAL_SM3	-	-	The 1-axis involved in interpolation

##### Input variable

Input variable	Name	Data type	Range	initialization	Descriptive
Exec	Start	BOOL	TRUE-FALSE	FALSE	Rising edge trigger
Halt	Pause	BOOL	TRUE-FALSE	FALSE	TRUE: the interpolation motion is suspended; when the state is switched to False again, the interpolation module continues the previous unfinished interpolation tasks
Stop	Stop	BOOL	TRUE-FALSE	FALSE	TRUE: The interpolation module is stopped.
IpoCycle	Interpolation cycle	DWORD	ALL	2000	Interpolation period, unit: us.
PosMode	Position Mode	INT	[0,1]	0	Positional mode: 0 Absolute mode; 1 Relative mode.
Endpos	Target position	ARRAY[0...1] OF REAL	ALL	0	Target position of the axis. Unit: pulse
VelocityMode	Velocity mode	SMC_INT_VE LMODE	0-3	SIGMOID	Velocity mode: Trapezoidal: 0 (TRAPEZOID); S-shaped: 1 (SIGMOID);

					Quadratic: 3 (QUADRATIC).
VelRatio	Velocity multiplier	LREAL	0.01-2	1	Minimum 0.01, maximum 2
Velocity	Target velocity	LREAL	Is always positive.	0	Interpolate synthetic velocity.
Acceleration	Target acceleration	LREAL	Is always positive.	0	Interpolate synthetic acceleration.
Deceleration	Target Deceleration	LREAL	Is always positive.	0	Interpolate the synthetic deceleration.
Jack	Target jack	LREAL	ALL	90000000	Acceleration, when the speed mode is 3, you need to set this parameter, the value can not be 0.

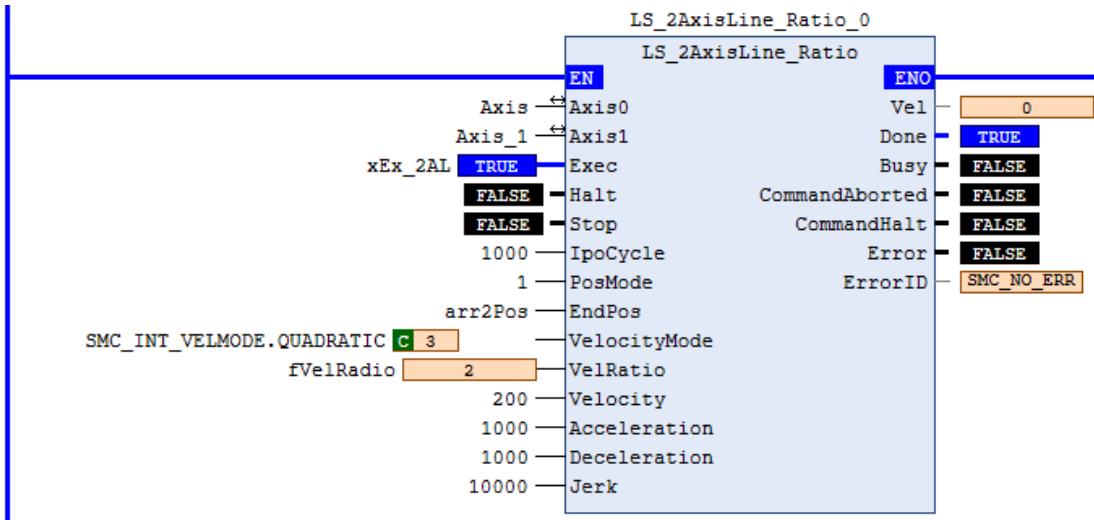
**Output variable**

Output variable	Name	Data type	Range	initialization	Descriptive
Vel	Closing speed	LREAL	0,positive number	0	Kinematic synthetic velocity
Done	Completion	BOOL	TRUE-FALSE	FALSE	TRUE: Homing is done.
Busy	Executing	BOOL	TRUE-FALSE	FALSE	TRUE: In progress
Command Aborted	Command Interrupt	BOOL	TRUE-FALSE	FALSE	TRUE: Terminated by other commands
CommandHalt	motion pause	BOOL	TRUE, FALSE	FALSE	When Halt is True, this variable is True and the interpolation module is in a suspended state.
Error	Error	BOOL	TRUE-FALSE	FALSE	TRUE: An error occurred within the function block
ErrorID	Error Code	SMC_ERRO R	-	0	Error indication, see SMC Error.

	Bo le an	Bit string					Integer							Real number		Moment, Duration, Date, String				
	BOOL	BYTE	WORD	DWORD	LWORD	USINT	UINT	UDINT	ULINT	SINT	INT	DINT	LINT	REAL	LREAL	TIME	DATE	TOD	DT	STRING
Axis		AXIS_REF_SM3																		
Exec	√	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
Halt	√	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
Stop	√	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
Endpos	-	-	-	-	-	-	-	-	-	-	-	-	-	√	-	-	-	-	-	-
Velocity	-	-	-	-	-	-	-	-	-	-	-	-	-	√	-	-	-	-	-	-
Acceleration	-	-	-	-	-	-	-	-	-	-	-	-	-	√	-	-	-	-	-	-
Deceleration	-	-	-	-	-	-	-	-	-	-	-	-	-	√	-	-	-	-	-	-
Jerk	-	-	-	-	-	-	-	-	-	-	-	-	-	√	-	-	-	-	-	-
Done	√	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
Busy	√	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
Command Aboned	√	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
Error	√	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
ErrorID		SMC_ERROR																		

**⊙ Program demo**

**LD:** Two axes in relative mode with interpolation speed of 2\*200 and linear interpolation motion to target position 2000, 2000.


**Note**

The IpoCycle parameter needs to be aligned with the task cycle time where the motion instruction is located, otherwise an Error may be reported when the instruction is executed.

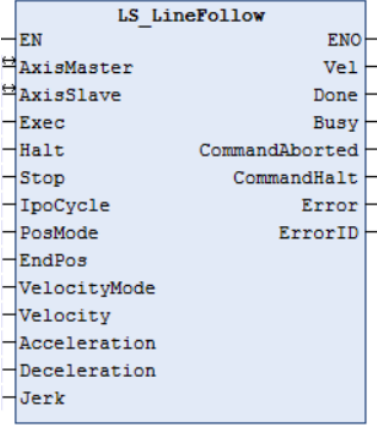
The parameters IpoCycle, Jerk, Velocity, Acceleration, Deceleration cannot be set to 0.

During the instruction running period, neither master nor slave axis can be called by other motion instructions

### 4.4.8 LS\_LineFollow

One axis points and the other axis follows the motion command.

#### ⊙ Command Format

Instruction	Name	FB/FC	LD	ST	File
LS_LineFollow	Follow the motion instruction	FC		LS_LineFollow( AxisMaster:=, AxisSlave:=, Exec:=, Halt:=, Stop:=, IpoCycle:=, PosMode:=, EndPos:=, VelocityMode:=, Velocity:=, Acceleration:=, Deceleration:=, Jerk:=, Vel=>, Done=>, Busy=>, CommandAborted=>, CommandHalt=>, Error=>, ErrorID=> );	LS_IpoLib

#### ⊙ Related Variables

##### I/O variable

I/O variable	Name	Data type	Range	initialization	Descriptive
AxisMaster	Master	AXIS_REF_VIRTUAL_SM3	-	-	AxisMaster
AxisSlave	Slave	AXIS_REF_VIRTUAL_SM3	-	-	AxisSlave

##### Input variable

Input variable	Name	Data type	Range	initialization	Descriptive
Exec	Start	BOOL	TRUE-FALSE	FALSE	Rising edge trigger
Halt	Pause	BOOL	TRUE-FALSE	FALSE	TRUE: the interpolation motion is suspended; when the state is switched to False again, the interpolation module continues the previous unfinished interpolation tasks
Stop	Stop	BOOL	TRUE-FALSE	FALSE	TRUE: The interpolation module is stopped.
IpoCycle	Interpolation cycle	DWORD	ALL	2000	Interpolation period, unit: us.
PosMode	Position Mode	INT	[0,1]	0	Positional mode: 0 Absolute mode; 1 Relative mode.
Endpos	Target position	ARRAY[0...1] OF REAL	ALL	0	Target position of the axis. Unit: pulse
VelocityMode	Velocity mode	SMC_INT_VE LMODE	0-3	SIGMOID	Velocity mode: Trapezoidal: 0 (TRAPEZOID); S-shaped: 1 (SIGMOID); Quadratic: 3 (QUADRATIC).
Velocity	Target velocity	LREAL	Is always positive.	0	Interpolate synthetic velocity.
Acceleration	Target	LREAL	Is always	0	Interpolate synthetic acceleration.

	acceleration		positive.		
Deceleration	Target Deceleration	LREAL	Is always positive.	0	Interpolate the synthetic deceleration.
Jack	Target jack	LREAL	ALL	90000000	Acceleration, when the speed mode is 3, you need to set this parameter, the value can not be 0.

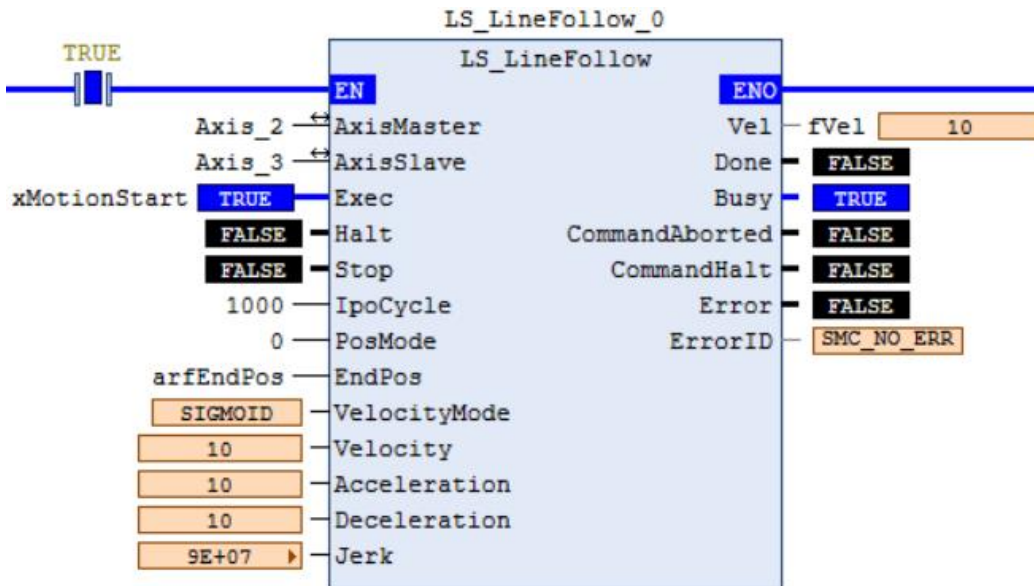
**Output variable**

Output variable	Name	Data type	Range	initialization	Descriptive
Vel	Closing speed	LREAL	0,positive number	0	Kinematic synthetic velocity
Done	Completion	BOOL	TRUE-FALSE	FALSE	TRUE: Homing is done.
Busy	Executing	BOOL	TRUE-FALSE	FALSE	TRUE: In progress
Command Aborted	Command Interrupt	BOOL	TRUE-FALSE	FALSE	TRUE: Terminated by other commands
CommandHalt	motion pause	BOOL	TRUE, FALSE	FALSE	When Halt is True, this variable is True and the interpolation module is in a suspended state.
Error	Error	BOOL	TRUE-FALSE	FALSE	TRUE: An error occurred within the function block
ErrorID	Error Code	SMC_ERROR	-	0	Error indication, see SMC_Error.

	Bo ole an	Bit string					Integer							Real number		Moment, Duration, Date, String				
	BOOL	BYTE	WORD	DWORD	LWORD	USINT	UINT	UDINT	ULINT	SINT	INT	DINT	LINT	REAL	LREAL	TIME	DATE	TOD	DT	STRING
Axis		AXIS_REF_SM3																		
Exec	√	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
Halt	√	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
Stop	√	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
Endpos	-	-	-	-	-	-	-	-	-	-	-	-	-	-	√	-	-	-	-	-
Velocity	-	-	-	-	-	-	-	-	-	-	-	-	-	-	√	-	-	-	-	-
Acceleration	-	-	-	-	-	-	-	-	-	-	-	-	-	-	√	-	-	-	-	-
Deceleration	-	-	-	-	-	-	-	-	-	-	-	-	-	-	√	-	-	-	-	-
Jerk	-	-	-	-	-	-	-	-	-	-	-	-	-	-	√	-	-	-	-	-
Done	√	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
Busy	√	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
Command Aboned	√	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
Error	√	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
ErrorID		SMC_ERROR																		

**Ⓞ Program demo**

**LD:** EndPos is set to [100,100] and the spindle moves synchronously from the axis to the specified position.


**Note**

The `IpoCycle` parameter needs to be aligned with the task cycle time where the motion instruction is located, otherwise an Error may be reported when the instruction is executed.

The parameters `IpoCycle`, `Jerk`, `Velocity`, `Acceleration` and `Deceleration` cannot be set to 0.

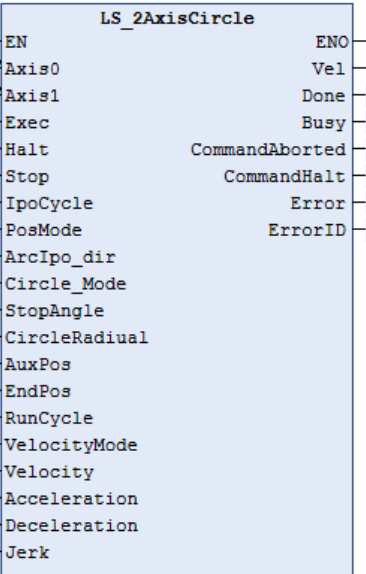
The follow motion instruction is only valid during the period from the start to the end of the instruction, and it has the effect of following control on master and slave axes, after reaching the instruction position, the master and slave axes are no longer synchronised, and both of them restore to the state of standstill.

After reaching the commanded position, the master and slave axes are no longer synchronised and will return to the standstill state. During the operation of the command, the master and slave axes cannot be called by other motion commands.

### 4.4.9 LS\_2AxisCircle

The two-axis plane arc interpolation command supports four modes: three-point arc, centre arc, radius arc and angle arc.

#### ⊙ Command Format

Instruction	Name	FB/FC	LD	ST	File
LS_2AxisCircle	Two axis circular interpolation instruction	FC		<pre> LS_2AxisCircle( Axis0:= , Axis1:= , Exec:= , Halt:= , Stop:= , IpoCycle:= , PosMode:= , ArcIpo_dir:= , Circle_Mode:= , StopAngle:= , CircleRadial:= , AuxPos:= , EndPos:= , RunCycle:= , VelocityMode:= , Velocity:= , Acceleration:= , Deceleration:= , Jerk:= , Vel=&gt; , Done=&gt; , Busy=&gt; , CommandAborted=&gt; , CommandHalt=&gt; , Error=&gt; , ErrorID=&gt; ); </pre>	LS_IpoLib

#### ⊙ Related Variables

##### I/O variable

I/O variable	Name	Data type	Range	initialization	Descriptive
Axis0	Axis0	AXIS_REF_VIRTUAL_SM3	-	-	The 0-axis involved in interpolation
Axis1	Axis1	AXIS_REF_VIRTUAL_SM3	-	-	The 1-axis involved in interpolation

##### Input variable

Input variable	Name	Data type	Range	initialization	Descriptive
Exec	Start	BOOL	TRUE-FALSE	FALSE	Rising edge trigger
Halt	Pause	BOOL	TRUE-FALSE	FALSE	TRUE: the interpolation motion is suspended; when the state is switched to False again, the interpolation module continues the previous unfinished interpolation tasks
Stop	Stop	BOOL	TRUE-FALSE	FALSE	TRUE: The interpolation module is stopped.
IpoCycle	Interpolation cycle	DWORD	ALL	2000	Interpolation period, unit: us.
PosMode	Position Mode	INT	[0,1]	0	Positional mode: 0 Absolute mode; 1 Relative mode.
ArcIpo_dir	Interpolation direction	INT	0, 1	0	The direction of circular interpolation: 0: counterclockwise direction; 1:

					clockwise direction. Circular arc mode 1, 2 and 3 need to be set.
Circle_Mode	Interpolation Mode	INT	0-3	1	Circular arc mode: 0: three-point circular arc mode; 1: circular centre position and endpoint determine circular arc; 2: endpoint radius mode; 3: target angle mode.
StopAngle	Stop Angle	REAL	0-360	0	The target angle of stop, relative to the starting point, in degrees, with a value in the range of [0, 360.0]; Arc Mode 3 requires this value to be set.
CircleRadial	Arc radius	REAL	positive number	0	The radius of the arc, arc mode 2 need to set this value. A negative value means that the arc angle is less than 180 degrees, and a positive radius means that the arc angle is greater than 180 degrees.
AuxPos	Auxiliary position	ARRAY [0..1] OF REAL	ALL	0	Auxiliary point in Pulse. arc mode 0 is the passing point of the three-point arc. Arc mode 1 and 3 are the position of the centre of the circle.
Endpos	Target position	ARRAY[0...1] OF REAL	ALL	0	Target position of the axis. Unit: pulse
RunCycle	Number of cycles	UINT	Is always positive.	0	The number of cycles of circular motion (i.e., the number of revolutions of the arc).
VelocityMode	Velocity mode	SMC_INT_VE LMODE	0-3	SIGMOID	Velocity mode: Trapezoidal: 0 (TRAPEZOID); S-shaped: 1 (SIGMOID); Quadratic: 3 (QUADRATIC).
Velocity	Target velocity	LREAL	Is always positive.	0	Interpolate synthetic velocity.
Acceleration	Target acceleration	LREAL	Is always positive.	0	Interpolate synthetic acceleration.
Deceleration	Target Deceleration	LREAL	Is always positive.	0	Interpolate the synthetic deceleration.
Jack	Target jack	LREAL	ALL	90000000	Acceleration, when the speed mode is 3, you need to set this parameter, the value can not be 0.

**Output variable**

Output variable	Name	Data type	Range	initialization	Descriptive
Vel	Closing speed	LREAL	0, positive number	0	Kinematic synthetic velocity
Done	Completion	BOOL	TRUE-FALSE	FALSE	TRUE: Homing is done.
Busy	Executing	BOOL	TRUE-FALSE	FALSE	TRUE: In progress
Command Aborted	Command Interrupt	BOOL	TRUE-FALSE	FALSE	TRUE: Terminated by other commands
CommandHalt	motion pause	BOOL	TRUE, FALSE	FALSE	When Halt is True, this variable is True and the interpolation module is in a suspended state.
Error	Error	BOOL	TRUE-FALSE	FALSE	TRUE: An error occurred within the function block

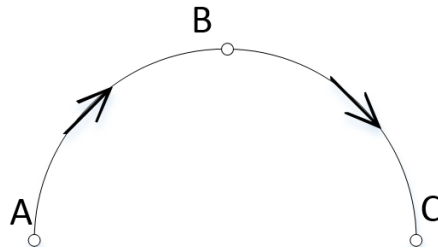
ErrorID	Error Code	SMC_ERRO R	-	0	Error indication, see SMC Error.
---------	------------	---------------	---	---	-------------------------------------

	Bo le an	Bit string					Integer						Real number		Moment, Duration, Date, String					
	BOOL	BYTE	WORD	DWORD	LWORD	USINT	UINT	UDINT	ULINT	SINT	INT	DINT	LINT	REAL	LREAL	TIME	DATE	TOD	DT	STRING
Axis	AXIS_REF_SM3																			
Exec	√	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
Halt	√	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
Stop	√	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
Endpos	-	-	-	-	-	-	-	-	-	-	-	-	-	-	√	-	-	-	-	-
Velocity	-	-	-	-	-	-	-	-	-	-	-	-	-	-	√	-	-	-	-	-
Acceleration	-	-	-	-	-	-	-	-	-	-	-	-	-	-	√	-	-	-	-	-
Deceleration	-	-	-	-	-	-	-	-	-	-	-	-	-	-	√	-	-	-	-	-
Jerk	-	-	-	-	-	-	-	-	-	-	-	-	-	-	√	-	-	-	-	-
Done	√	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
Busy	√	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
Command Aboned	√	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
Error	√	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
ErrorID	SMC_ERROR																			

**⊙ Functional Description**

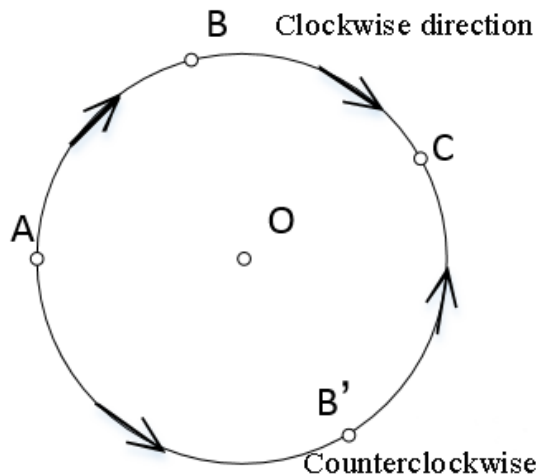
The 'Circle\_Mode' parameter is used to set the mode of circular interpolation as follows:

1) Circle\_Mode=0, indicates three-point circular arc mode, as shown in the figure.



When using the three-point arc mode, you need to determine three non-overlapping points on the arc trajectory, such as A, B and C points in the figure. In the instruction, A represents the current position of arc interpolation axes 0 and 1, which does not need to be set; B represents the point through which the arc passes, which needs to be set as parameter AuxPos; C represents the end position of the arc, which needs to be set as parameter Endpos.

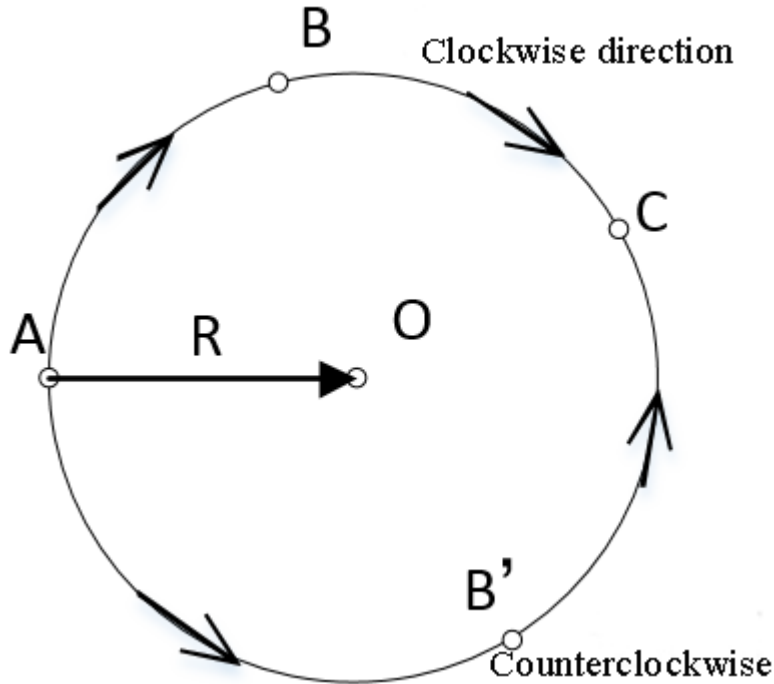
2) Circle\_Mode=1, denotes the arc of the circle with centre and end mode as shown in the figure.



When using the Circle Centre Endpoint Arc mode, you need to set the parameters of Circle Centre O, Endpoint C

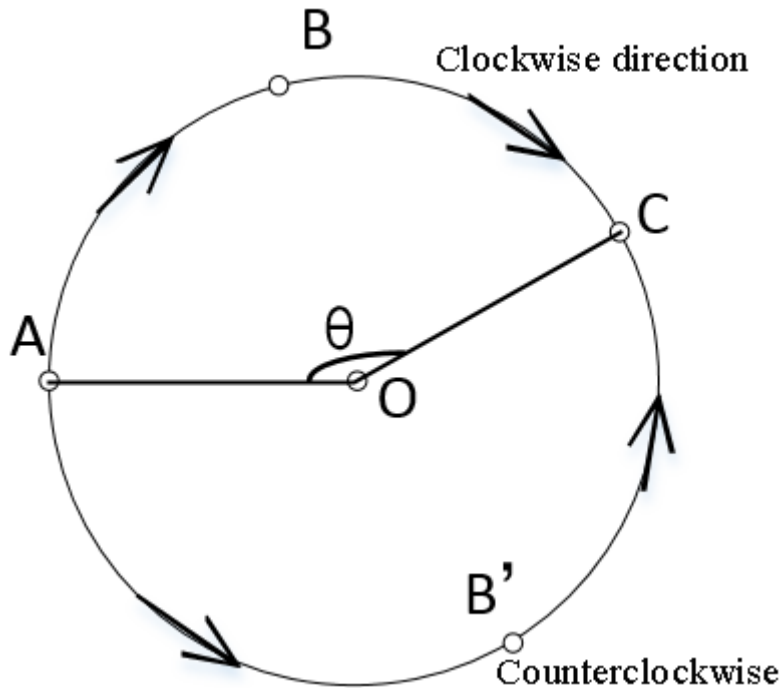
and Movement Direction. In the instruction O represents the coordinates of the centre of the circle, which needs to be set to the parameter AuxPos; C represents the coordinates of the end point of the arc, which needs to be set to the parameter Endpos; and the direction of motion needs to be set to the parameter ArcIpo\_dir.

3) Circle\_Mode=2, indicates the end point radius mode, as shown in the figure.



When using the endpoint radius mode, you need to set parameters such as endpoint C, radius R and motion direction. In the instruction, C represents the end position of the arc and needs to be set to the parameter Endpos; the radius R needs to be set to the parameter CircleRadial; and the movement direction needs to be set to the parameter ArcIpo\_dir.

4) Circle\_Mode=3, indicating the target angle mode, as shown in the figure.

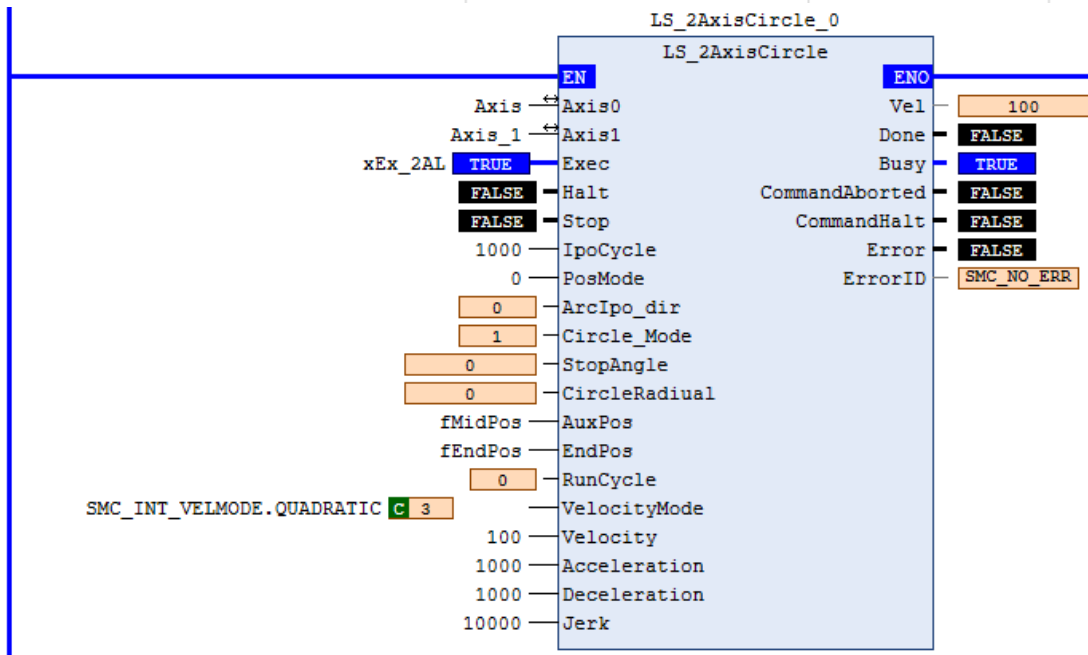


When using the target angle mode, you need to set the parameters of circle centre O, angle  $\theta$  and motion direction. In the instruction O represents the position of the circle centre, which needs to be set as parameter Auxpos; the target angle  $\theta$  needs to be set as parameter StopAngle; and the motion direction needs to be set as parameter ArcIpo\_dir.

**⊙ Program demo**

**LD:** The two-axis planar arc performs a three-point drawing of a semicircle with a start point of 0,0, a three-point arc passing through points 1000,1000, and a target point of 2000,0.

[-] ◆ fMidPos	ARRAY [0..1] OF LREAL	
◆ fMidPos[0]	LREAL	1000
◆ fMidPos[1]	LREAL	1000
[-] ◆ fEndPos	ARRAY [0..1] OF LREAL	
◆ fEndPos[0]	LREAL	2000
◆ fEndPos[1]	LREAL	0


**Note**

The IpoCycle parameter needs to be aligned with the task cycle time where the motion instruction is located, otherwise an Error may be reported when the instruction is executed.

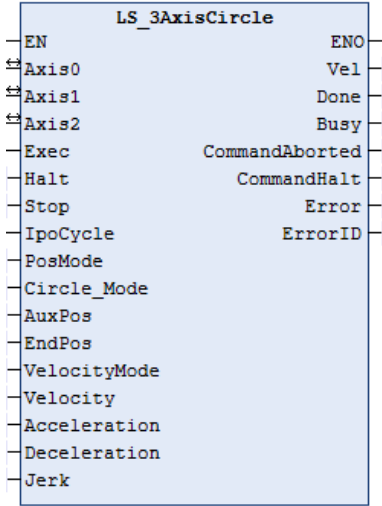
The parameters IpoCycle, Jerk, Velocity, Acceleration and Deceleration cannot be set to 0.

The master and slave axes can not be called by other motion instructions during the instruction running time.

### 4.4.10 LS\_3AxisCircle

3-axis spatial arc interpolation command, only supports 3-point arc mode.

#### ⊙ Command Format

Instruction	Name	FB/FC	LD	ST	File
LS_3AxisCircle	Three axis circular interpolation instruction	FC		<pre> LS_2AxisCircle( Axis0:= , Axis1:= , Axis2:= , Exec:= , Halt:= , Stop:= , IpoCycle:= , PosMode:= , Circle_Mode:= , AuxPos:= , EndPos:= , VelocityMode:= , Velocity:= , Acceleration:= , Deceleration:= , Jerk:= , Vel=&gt; , Done=&gt; , Busy=&gt; , CommandAborted=&gt; , CommandHalt=&gt; , Error=&gt; , ErrorID=&gt; ); </pre>	LS_IpoLib

#### ⊙ Related Variables

##### I/O variable

I/O variable	Name	Data type	Range	initialization	Descriptive
Axis0	Axis0	AXIS_REF_VIRTUAL_SM3	-	-	The 0-axis involved in interpolation
Axis1	Axis1	AXIS_REF_VIRTUAL_SM3	-	-	The 1-axis involved in interpolation
Axis2	Axis2	AXIS_REF_VIRTUAL_SM3	-	-	The 2-axis involved in interpolation

##### Input variable

Input variable	Name	Data type	Range	initialization	Descriptive
Exec	Start	BOOL	TRUE-FALSE	FALSE	Rising edge trigger
Halt	Pause	BOOL	TRUE-FALSE	FALSE	TRUE: the interpolation motion is suspended; when the state is switched to False again, the interpolation module continues the previous unfinished interpolation tasks
Stop	Stop	BOOL	TRUE-FALSE	FALSE	TRUE: The interpolation module is stopped.
IpoCycle	Interpolation cycle	DWORD	ALL	2000	Interpolation period, unit: us.
PosMode	Position Mode	INT	[0,1]	0	Positional mode: 0 Absolute mode; 1 Relative mode.
Circle_Mode	interpolation mode	INT	0	0	Arc Mode: 0: 3-point arc mode, only this mode is supported.
AuxPos	Auxiliary position	ARRAY [0..1] OF REAL	ALL	0	Auxiliary point in Pulse. arc mode 0 is the passing point of the three-point arc. Arc mode 1 and 3 are the position of the

					centre of the circle.
Endpos	Target position	ARRAY[0...1] OF REAL	ALL	0	Target position of the axis. Unit: pulse
VelocityMode	Velocity mode	SMC_INT_VE LMODE	0-3	SIGMOID	Velocity mode: Trapezoidal: 0 (TRAPEZOID); S-shaped: 1 (SIGMOID); Quadratic: 3 (QUADRATIC).
Velocity	Target velocity	LREAL	Is always positive.	0	Interpolate synthetic velocity.
Acceleration	Target acceleration	LREAL	Is always positive.	0	Interpolate synthetic acceleration.
Deceleration	Target Deceleration	LREAL	Is always positive.	0	Interpolate the synthetic deceleration.
Jack	Target jack	LREAL	ALL	90000000	Acceleration, when the speed mode is 3, you need to set this parameter, the value can not be 0.

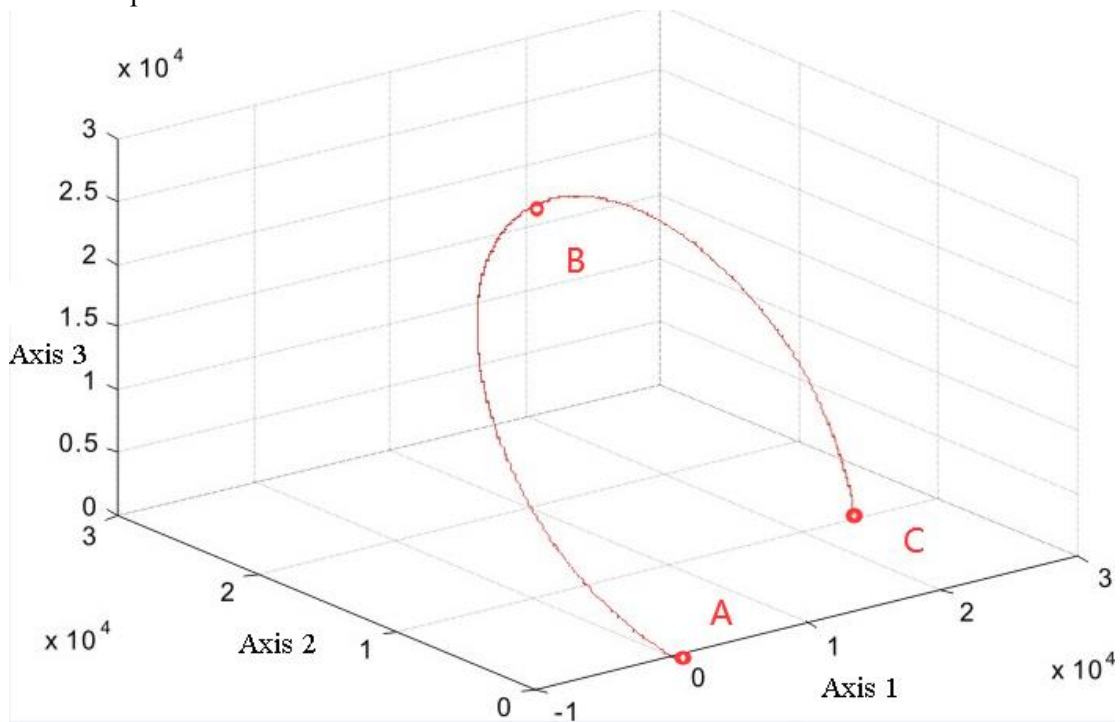
**Output variable**

Output variable	Name	Data type	Range	initialization	Descriptive
Vel	Closing speed	LREAL	0,positive number	0	Kinematic synthetic velocity
Done	Completion	BOOL	TRUE-FALSE	FALSE	TRUE: Homing is done.
Busy	Executing	BOOL	TRUE-FALSE	FALSE	TRUE: In progress
Command Aborted	Command Interrupt	BOOL	TRUE-FALSE	FALSE	TRUE: Terminated by other commands
CommandHalt	motion pause	BOOL	TRUE, FALSE	FALSE	When Halt is True, this variable is True and the interpolation module is in a suspended state.
Error	Error	BOOL	TRUE-FALSE	FALSE	TRUE: An error occurred within the function block
ErrorID	Error Code	SMC_ERRO R	-	0	Error indication, see SMC Error.

	Bo le an	Bit string				Integer							Real number		Moment, Duration, Date, String					
	BOOL	BYTE	WORD	DWORD	LWORD	USINT	UINT	UDINT	ULINT	SINT	INT	DINT	LINT	REAL	LREAL	TIME	DATE	TOD	DT	STRING
Axis		AXIS_REF_SM3																		
Exec	√	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
Halt	√	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
Stop	√	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
Endpos	-	-	-	-	-	-	-	-	-	-	-	-	-	-	√	-	-	-	-	-
Velocity	-	-	-	-	-	-	-	-	-	-	-	-	-	-	√	-	-	-	-	-
Acceleration	-	-	-	-	-	-	-	-	-	-	-	-	-	-	√	-	-	-	-	-
Deceleration	-	-	-	-	-	-	-	-	-	-	-	-	-	-	√	-	-	-	-	-
Jerk	-	-	-	-	-	-	-	-	-	-	-	-	-	-	√	-	-	-	-	-
Done	√	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
Busy	√	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
Command Aboned	√	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
Error	√	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
ErrorID		SMC_ERROR																		

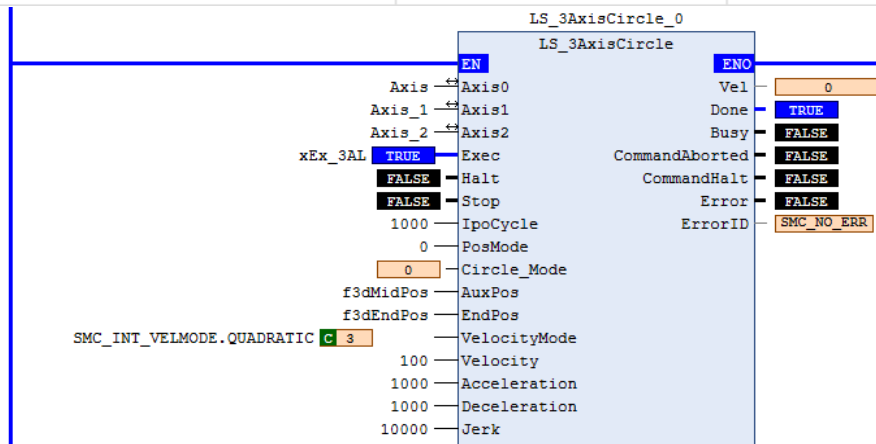
**⊙ Functional Description**

The 'Circle\_Mode' parameter is used to set the arc mode of the three-axis arc, the default mode is three-point arc, the corresponding parameter value is 0, and only this arc mode is supported, the arc trajectory is as shown in the figure, point A is the current position of axes 0, 1, and 2; point B is the point where the arc passes through, and point C is the end point of the arc.


**⊙ Program demo**

**LD:** The two-axis planar arc performs a three-point drawing of a semicircle with a start point of 0,0, a three-point arc passing through points 1000,1000, and a target point of 2000,0.

表达式	类型	值
f3dMidPos	ARRAY [0..2] OF LREAL	
f3dMidPos[0]	LREAL	1000
f3dMidPos[1]	LREAL	1000
f3dMidPos[2]	LREAL	100
f3dEndPos	ARRAY [0..2] OF LREAL	
f3dEndPos[0]	LREAL	2000
f3dEndPos[1]	LREAL	0
f3dEndPos[2]	LREAL	0

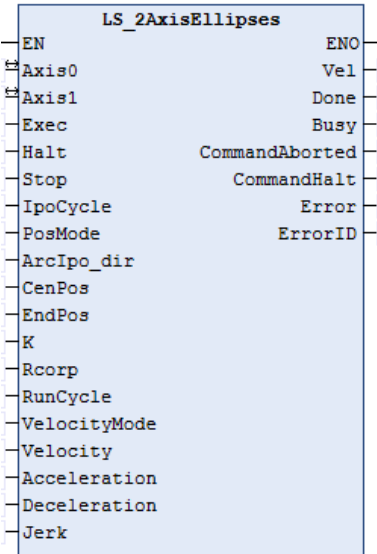

**Note**

The IpoCycle parameter needs to be aligned with the task cycle time where the motion instruction is located, otherwise an Error may be reported when the instruction is executed.  
 The parameters IpoCycle, Jerk, Velocity, Acceleration and Deceleration cannot be set to 0.  
 The master and slave axes can not be called by other motion instructions during the instruction running time.

### 4.4.11 LS\_2AxisEllipses

Two axis elliptic interpolation command.

#### ⊙ Command Format

Instruction	Name	FB/FC	LD	ST	File
LS_2AxisEllipses	Two axis elliptic interpolation instruction	FC		<pre> LS_2AxisEllipses( Axis0:= , Axis1:= , Exec:= , Halt:= , Stop:= , IpoCycle:= , PosMode:= , ArcIpo_dir:= , CenPos:= , EndPos:= , K:= , Rcorp:= , RunCycle:= , VelocityMode:= , Velocity:= , Acceleration:= , Deceleration:= , Jerk:= , Vel=&gt; , Done=&gt; , Busy=&gt; , CommandAborted=&gt; , CommandHalt=&gt; , Error=&gt; , ErrorID=&gt; ); </pre>	LS_IpoLib

#### ⊙ Related Variables

##### I/O variable

I/O variable	Name	Data type	Range	initialization	Descriptive
Axis0	Axis0	AXIS_REF_VIRTUAL_SM3	-	-	The 0-axis involved in interpolation
Axis1	Axis1	AXIS_REF_VIRTUAL_SM3	-	-	The 1-axis involved in interpolation

##### Input variable

Input variable	Name	Data type	Range	initialization	Descriptive
Exec	Start	BOOL	TRUE-FALSE	FALSE	Rising edge trigger
Halt	Pause	BOOL	TRUE-FALSE	FALSE	TRUE: the interpolation motion is suspended; when the state is switched to False again, the interpolation module continues the previous unfinished interpolation tasks
Stop	Stop	BOOL	TRUE-FALSE	FALSE	TRUE: The interpolation module is stopped.
IpoCycle	Interpolation cycle	DWORD	ALL	2000	Interpolation period, unit: us.
PosMode	Position Mode	INT	[0,1]	0	Positional mode: 0 Absolute mode; 1 Relative mode.
ArcIpo_dir	Interpolation direction	INT	0, 1	0	Direction of ellipse interpolation: 0:counterclockwise direction; 1:clockwise direction.
CenPos	Centre position	ARRAY [0..1]	ALL	0	The position of the centre of the ellipse,

		OF REAL			in absolute coordinates. Unit: Pulse.
Endpos	End position	ARRAY [0..1] OF REAL	ALL	0	End position, absolute coordinates. Unit: Pulse.
K	Spindle angle	REAL	0-360	0	The angle of the spindle, the angle of the spindle relative to the X-axis; the angle between the long axis and the X-axis, used to determine the direction of the spindle.
Rcorp	Short axis long axis ratio	REAL	0, 1	0	The ratio of the short axis to the long axis, taking values in the range [0,1].
RunCycle	Number of cycles	UINT	Is always positive.	0	The number of cycles of elliptical motion (i.e., the number of revolutions of the ellipse).
VelocityMode	Velocity mode	SMC_INT_VE LMODE	0-3	SIGMOID	Velocity mode: Trapezoidal: 0 (TRAPEZOID); S-shaped: 1 (SIGMOID); Quadratic: 3 (QUADRATIC).
Velocity	Target velocity	LREAL	Is always positive.	0	Interpolate synthetic velocity.
Acceleration	Target acceleration	LREAL	Is always positive.	0	Interpolate synthetic acceleration.
Deceleration	Target Deceleration	LREAL	Is always positive.	0	Interpolate the synthetic deceleration.
Jaek	Target jaek	LREAL	ALL	90000000	Acceleration, when the speed mode is 3, you need to set this parameter, the value can not be 0.

**Output variable**

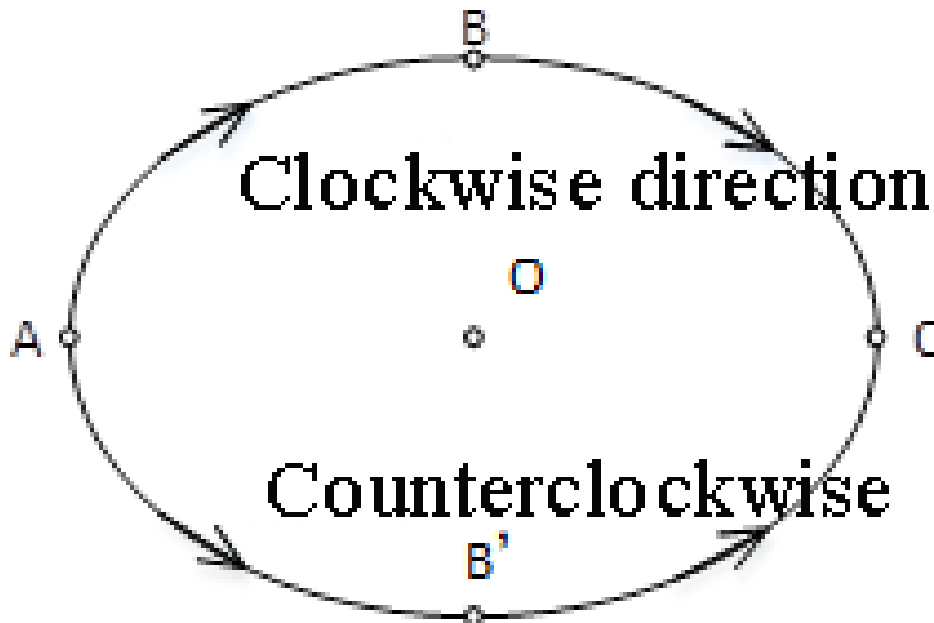
Output variable	Name	Data type	Range	initialization	Descriptive
Vel	Closing speed	LREAL	0, positive number	0	Kinematic synthetic velocity
Done	Completion	BOOL	TRUE-FALSE	FALSE	TRUE: Homing is done.
Busy	Executing	BOOL	TRUE-FALSE	FALSE	TRUE: In progress
Command Aborted	Command Interrupt	BOOL	TRUE-FALSE	FALSE	TRUE: Terminated by other commands
CommandHalt	motion pause	BOOL	TRUE, FALSE	FALSE	When Halt is True, this variable is True and the interpolation module is in a suspended state.
Error	Error	BOOL	TRUE-FALSE	FALSE	TRUE: An error occurred within the function block
ErrorID	Error Code	SMC_ERRO R	-	0	Error indication, see SMC_Error.

	Bo ole an	Bit string				Integer							Real number		Moment, Duration, Date, String					
	BOOL	BYTE	WORD	DWORD	LWORD	USINT	UINT	UDINT	ULINT	SINT	INT	DINT	LINT	REAL	LREAL	TIME	DATE	TOD	DT	STRING
Axis		AXIS_REF_SM3																		
Exec	√	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
Halt	√	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-

Stop	√	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
Endpos	-	-	-	-	-	-	-	-	-	-	-	-	-	-	√	-	-	-	-	-
Velocity	-	-	-	-	-	-	-	-	-	-	-	-	-	-	√	-	-	-	-	-
Acceleration	-	-	-	-	-	-	-	-	-	-	-	-	-	-	√	-	-	-	-	-
Deceleration	-	-	-	-	-	-	-	-	-	-	-	-	-	-	√	-	-	-	-	-
Jerk	-	-	-	-	-	-	-	-	-	-	-	-	-	-	√	-	-	-	-	-
Done	√	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
Busy	√	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
Command Aboned	√	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
Error	√	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
ErrorID	SMC_ERROR																			

⊙ **Functional Description**

The two axes specified by the command do elliptical interpolation motion, the trajectory of elliptical interpolation, as shown in Fig.

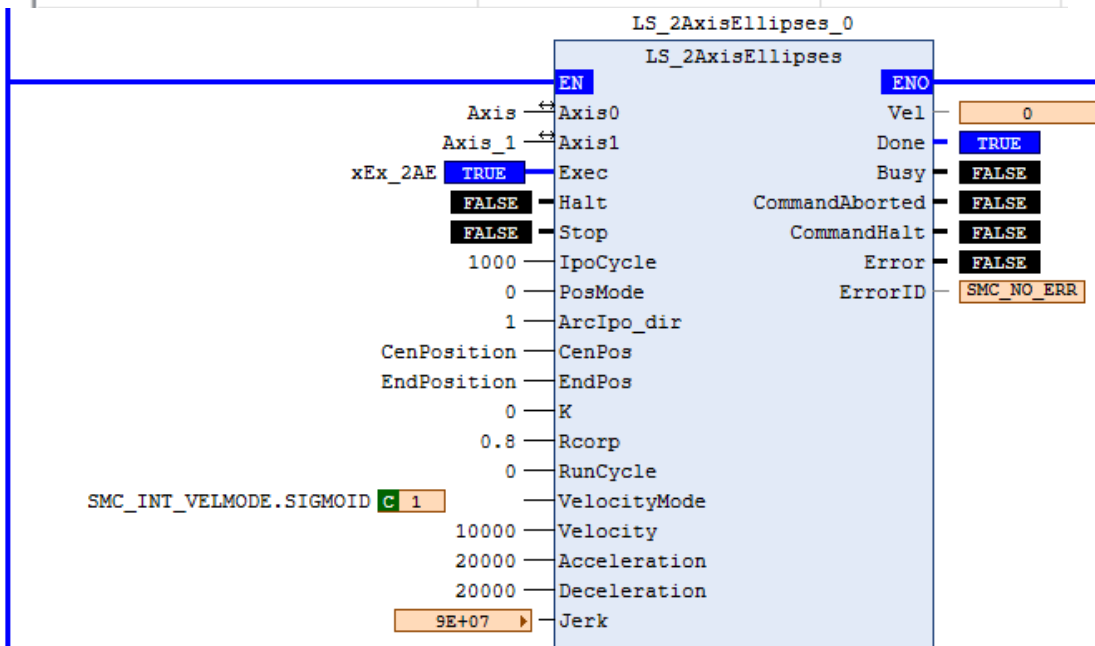


When using elliptical interpolation motion, you need to set parameters such as circle centre O, end point C and motion direction. In the instruction, A represents the current position of arc interpolation axes 0 and 1, which does not need to be set; O represents the position of the centre of the circle, which needs to be set to the parameter CenPos; C represents the end position of the arc, which needs to be set to the parameter Endpos; the direction of movement needs to be set to the parameter ArcIpo\_dir; and the number of laps of the elliptical trajectory execution needs to be set to the parameter RunCycle.

**⊙ Program demo**

**LD:** XY axis elliptic interpolation movement, current position (0, 0) clockwise movement to (10000, 0), X axis is the long axis, length 10000, Y is the short axis, length 8000, the centre point (5000, 0), the unit is Pulse, the running speed 10000Pulse/s, acceleration and deceleration 20000Pulse/s<sup>2</sup> .

表达式	类型	值
[-]  CenPosition	ARRAY [0..1] OF LREAL	
CenPosition[0]	LREAL	5000
CenPosition[1]	LREAL	0
[-]  EndPosition	ARRAY [0..1] OF LREAL	
EndPosition[0]	LREAL	10000
EndPosition[1]	LREAL	0


**Note**

The IpoCycle parameter needs to be aligned with the task cycle time where the motion instruction is located, otherwise an Error may be reported when the instruction is executed.

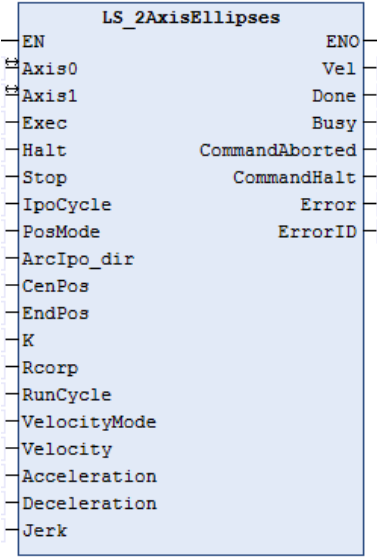
The parameters IpoCycle, Jerk, Velocity, Acceleration and Deceleration cannot be set to 0.

The master and slave axes can not be called by other motion instructions during the instruction running time.

### 4.4.12 LS\_2AxisCircle\_Helical

Spiral interpolation instructions.

#### ⊙ Command Format

Instruction	Name	FB/FC	LD	ST	File
LS_2AxisCircle_Helical	Helical interpolation instruction	FC		LS_2AxisCircle_Helical( Axis0:= , Axis1:= , Axis2:= , Exec:= , Halt:= , Stop:= , IpoCycle:= , PosMode:= , ArcIpo_dir:= , Circle_Mode:= , StopAngle:= , CircleRadiual:= , AuxPos:= , EndPos:= , HelicalPos:= , RunCycle:= , VelocityMode:= , Velocity:= , Acceleration:= , Deceleration:= , Jerk:= , Vel=> , Done=> , Busy=> , CommandAborted=> , CommandHalt=> , Error=> , ErrorID=> );	LS_IpoLib

#### ⊙ Related Variables

##### I/O variable

I/O variable	Name	Data type	Range	initialization	Descriptive
Axis0	Axis0	AXIS_REF_VIRTUAL_SM3	-	-	The 0-axis involved in interpolation
Axis1	Axis1	AXIS_REF_VIRTUAL_SM3	-	-	The 1-axis involved in interpolation

##### Input variable

Input variable	Name	Data type	Range	initialization	Descriptive
Exec	Start	BOOL	TRUE-FALSE	FALSE	Rising edge trigger
Halt	Pause	BOOL	TRUE-FALSE	FALSE	TRUE: the interpolation motion is suspended; when the state is switched to False again, the interpolation module continues the previous unfinished interpolation tasks
Stop	Stop	BOOL	TRUE-FALSE	FALSE	TRUE: The interpolation module is stopped.
IpoCycle	Interpolation cycle	DWORD	ALL	2000	Interpolation period, unit: us.
PosMode	Position Mode	INT	[0,1]	0	Positional mode: 0 Absolute mode; 1 Relative mode.
ArcIpo_dir	Interpolation	INT	0, 1	0	Direction of ellipse interpolation:

	direction				0:counterclockwise direction; 1:clockwise direction.
Circle_Mode	Interpolation Mode	INT	0-3	1	Arc mode: 0: three-point arc mode; 1: circle centre position and end point to determine the arc; 2: end point radius mode; 3: target angle mode.
StopAngle	Stop Angle	REAL	0-360	0	The target angle of the stop, relative to the start point, in degrees, with a value in the range [0, 360.0]. Arc mode 3 requires this value to be set.
CircleRadiual	Arc radius	REAL	Is always positive.	0	The radius of the arc, which needs to be set for arc mode 2. A negative radius for this value indicates a large arc and a positive radius indicates a small arc.
AuxPos	Auxiliary position	ARRAY [0..1] OF REAL	ALL	0	Auxiliary point in Pulse.
Endpos	End position	ARRAY [0..1] OF REAL	ALL	0	End position, absolute coordinates. Unit: Pulse.
HelicalPos	Screw shaft position	LREAL	ALL	0	Spiral axis position.Unit: Pulse.
RunCycle	Number of cycles	UINT	Is always positive.	0	The number of cycles of elliptical motion (i.e., the number of revolutions of the ellipse).
VelocityMode	Velocity mode	SMC_INT_VE LMODE	0-3	SIGMOID	Velocity mode: Trapezoidal: 0 (TRAPEZOID); S-shaped: 1 (SIGMOID); Quadratic: 3 (QUADRATIC).
Velocity	Target velocity	LREAL	Is always positive.	0	Interpolate synthetic velocity.
Acceleration	Target acceleration	LREAL	Is always positive.	0	Interpolate synthetic acceleration.
Deceleration	Target Deceleration	LREAL	Is always positive.	0	Interpolate the synthetic deceleration.
Jack	Target jack	LREAL	ALL	90000000	Acceleration, when the speed mode is 3, you need to set this parameter, the value can not be 0.

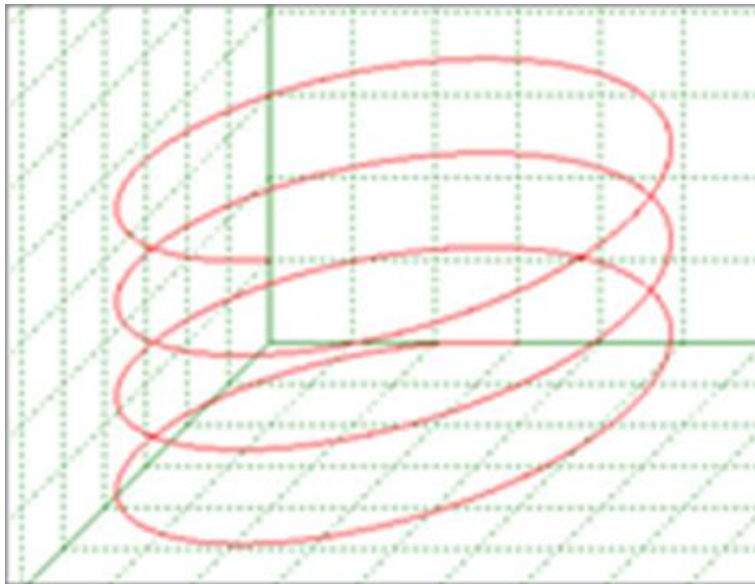
**Output variable**

Output variable	Name	Data type	Range	initialization	Descriptive
Vel	Closing speed	LREAL	0,positive number	0	Kinematic synthetic velocity
Done	Completion	BOOL	TRUE-FALSE	FALSE	TRUE: Homing is done.
Busy	Executing	BOOL	TRUE-FALSE	FALSE	TRUE: In progress
Command Aborted	Command Interrupt	BOOL	TRUE-FALSE	FALSE	TRUE: Terminated by other commands
CommandHalt	motion pause	BOOL	TRUE, FALSE	FALSE	When Halt is True, this variable is True and the interpolation module is in a suspended state.
Error	Error	BOOL	TRUE-FALSE	FALSE	TRUE: An error occurred within the function block
ErrorID	Error Code	SMC_ERRO R	-	0	Error indication, see SMC_Error.

	Bo le an	Bit string				Integer							Real number		Moment, Duration, Date, String					
	BOOL	BYTE	WORD	DWORD	LWORD	USINT	UINT	UDINT	ULINT	SINT	INT	DINT	LINT	REAL	LREAL	TIME	DATE	TOD	DT	STRING
Axis		AXIS_REF_SM3																		
Exec	√	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
Halt	√	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
Stop	√	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
Endpos	-	-	-	-	-	-	-	-	-	-	-	-	-	-	√	-	-	-	-	-
Velocity	-	-	-	-	-	-	-	-	-	-	-	-	-	-	√	-	-	-	-	-
Acceleration	-	-	-	-	-	-	-	-	-	-	-	-	-	-	√	-	-	-	-	-
Deceleration	-	-	-	-	-	-	-	-	-	-	-	-	-	-	√	-	-	-	-	-
Jerk	-	-	-	-	-	-	-	-	-	-	-	-	-	-	√	-	-	-	-	-
Done	√	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
Busy	√	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
Command Aboned	√	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
Error	√	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
ErrorID		SMC_ERROR																		

**⊙ Functional Description**

The specified three axes do spatial helical interpolation motion. The trajectory of the helical interpolation, as shown in Fig.

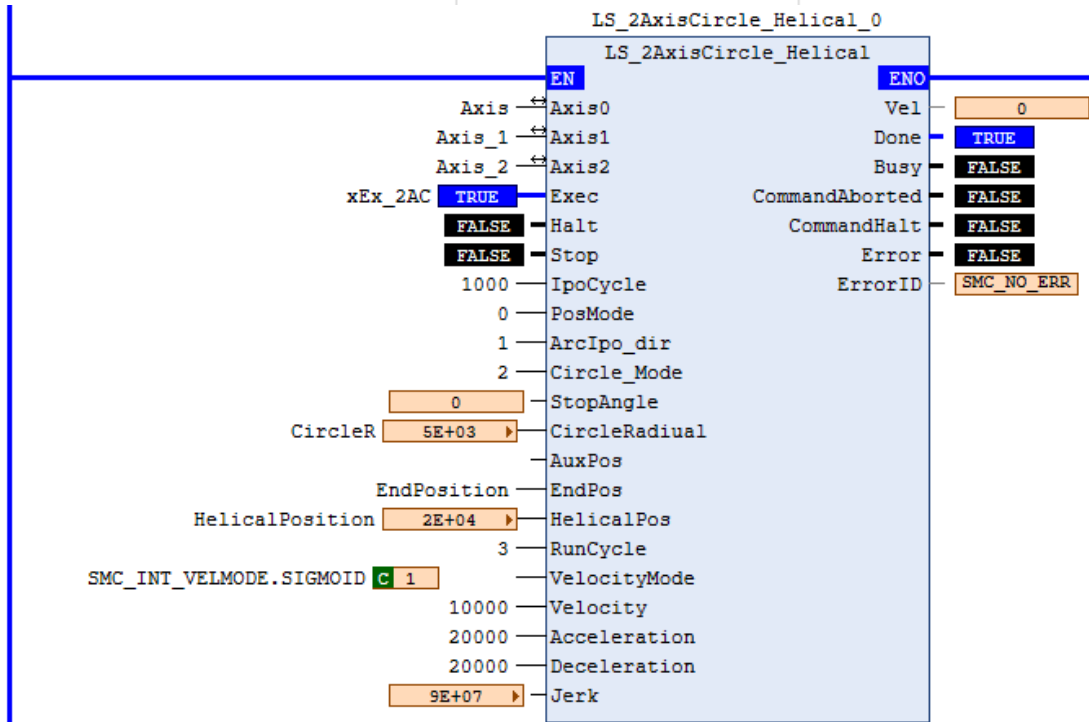


In the helical interpolation motion, axis 0 and axis 1 are used to realise circular motion, the 'Circle\_Mode' parameter is used to set the mode of circular interpolation of axis 0 and axis 1, and the definition is the same as that of the two-axis circular interpolation instruction.

**⊙ Program demo**

**LD:** The XYZ axis does three-axis circular arc helix interpolation motion. Among them, XY axis does circular arc interpolation, using mode 2 end point radius mode, radius 5000, end point (10000, 0), Z axis does following motion, distance is 10000, unit is Pulse, the number of circular arc cycle is 3 times, set the running speed of 10000Pulse/s, and the speed of acceleration and deceleration 20000Pulse/s<sup>2</sup>.

表达式	类型	值
CircleR	LREAL	5000
EndPosition	ARRAY [0..1] OF LREAL	
EndPosition[0]	LREAL	10000
EndPosition[1]	LREAL	0
HelicalPosition	LREAL	20000


**Note**

The IpoCycle parameter needs to be aligned with the task cycle time where the motion instruction is located, otherwise an Error may be reported when the instruction is executed.

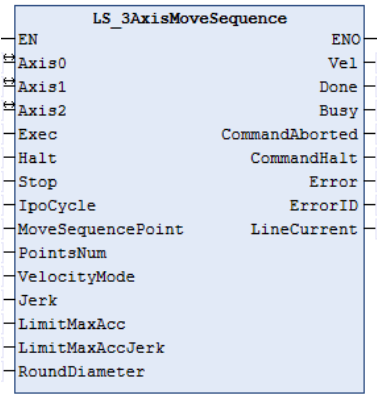
The parameters IpoCycle, Jerk, Velocity, Acceleration and Deceleration cannot be set to 0.

The master and slave axes can not be called by other motion instructions during the instruction running time.

### 4.4.13 LS\_3AxisMoveSequence

Three-axis continuous interpolation command with continuous speed.

#### ⊙ Command Format

Instruction	Name	FB/FC	LD	ST	File
LS_3AxisMoveSequence	Three axis continuous interpolation motion instruction	FC		<pre> LS_3AxisMoveSequence( Axis0:= , Axis1:= , Axis2:= , Exec:= , Halt:= , Stop:= , IpoCycle:= , MoveSequencePoint:= , PointsNum:= , VelocityMode:= , Jerk:= , LimitMaxAcc:= , LimitMaxAccJerk:= , RoundDiameter:= , Vel=&gt; , Done=&gt; , Busy=&gt; , CommandAborted=&gt; , CommandHalt=&gt; , Error=&gt; , ErrorID=&gt; , LineCurrent=&gt; );                     </pre>	LS_IpoLib

#### ⊙ Related Variables

##### I/O variable

I/O variable	Name	Data type	Range	initialization	Descriptive
Axis0	Axis0	AXIS_REF_VIRTUAL_SM3	-	-	The 0-axis involved in interpolation
Axis1	Axis1	AXIS_REF_VIRTUAL_SM3	-	-	The 1-axis involved in interpolation
Axis2	Axis2	AXIS_REF_VIRTUAL_SM3	-	-	The 2-axis involved in interpolation

##### Input variable

Input variable	Name	Data type	Range	initialization	Descriptive
Exec	Start	BOOL	TRUE-FALSE	FALSE	Rising edge trigger
Halt	Pause	BOOL	TRUE-FALSE	FALSE	TRUE: the interpolation motion is suspended; when the state is switched to False again, the interpolation module continues the previous unfinished interpolation tasks
Stop	Stop	BOOL	TRUE-FALSE	FALSE	TRUE: The interpolation module is stopped.
IpoCycle	Interpolation cycle	DWORD	ALL	2000	Interpolation period, unit: us.
MoveSequencePoint	Interpolation Data Sheet	ARRAY [0..MoveSequence_MAX] OF MoveSequence	-	-	Input data points.
PointsNum	check numbers	UINT	positive number	0	Number of data points entered.
VelocityMode	Velocity mode	SMC_INT_VE LMODE	0-3	SIGMOID	Velocity mode: Trapezoidal: 0 (TRAPEZOID); S-shaped: 1 (SIGMOID); Quadratic: 3 (QUADRATIC).

Jerk	Acceleration	REAL	positive number	90000000	Acceleration, you need to set this parameter when the speed mode is 3.
LimitMaxAcc	Maximum acceleration	REAL	positive number	9000000	Maximum acceleration limit for corner transition arc.
LimitMaxAccJerk	Maximum deceleration	REAL	positive number	90000000	Maximum acceleration limit for corner transition arc.
RoundDiameter	Corner radius	REAL	positive number	100	Corner transition radius, unit: Pulse.

**Output variable**

Output variable	Name	Data type	Range	initialization	Descriptive
Vel	Closing speed	LREAL	0, positive number	0	Kinematic synthetic velocity
Done	Completion	BOOL	TRUE-FALSE	FALSE	TRUE: Homing is done.
Busy	Executing	BOOL	TRUE-FALSE	FALSE	TRUE: In progress
Command Aborted	Command Interrupt	BOOL	TRUE-FALSE	FALSE	TRUE: Terminated by other commands
CommandHalt	motion pause	BOOL	TRUE, FALSE	FALSE	When Halt is True, this variable is True and the interpolation module is in a suspended state.
Error	Error	BOOL	TRUE-FALSE	FALSE	TRUE: An error occurred within the function block
ErrorID	Error Code	SMC_ERROR	-	0	Error indication, see SMC Error.
LineCuren	Line Number	UINT	Follows the data type	0	The line number of the current execution.

	Bo	Bit string					Integer							Real number		Moment, Duration, Date, String					
	olean	BOOL	BYTE	WORD	DWORD	LWORD	USINT	UINT	UDINT	ULINT	SINT	INT	DINT	LINT	REAL	LREAL	TIME	DATE	TOD	DT	STRING
Axis		AXIS_REF_SM3																			
Exec	√	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
Halt	√	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
Stop	√	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
Endpos	-	-	-	-	-	-	-	-	-	-	-	-	-	-	√	-	-	-	-	-	-
Velocity	-	-	-	-	-	-	-	-	-	-	-	-	-	-	√	-	-	-	-	-	-
Acceleration	-	-	-	-	-	-	-	-	-	-	-	-	-	-	√	-	-	-	-	-	-
Deceleration	-	-	-	-	-	-	-	-	-	-	-	-	-	-	√	-	-	-	-	-	-
Jerk	-	-	-	-	-	-	-	-	-	-	-	-	-	-	√	-	-	-	-	-	-
Done	√	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
Busy	√	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
Command Aboned	√	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
Error	√	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
ErrorID		SMC_ERROR																			

**⊙ Functional Description**

Specify 3 axes for continuous interpolation motion, the position coordinates are absolute coordinates, and you can input up to 200 interpolation motion points at a time.

In the continuous interpolation motion, each motion point position, is in the form of MoveSequence structure.

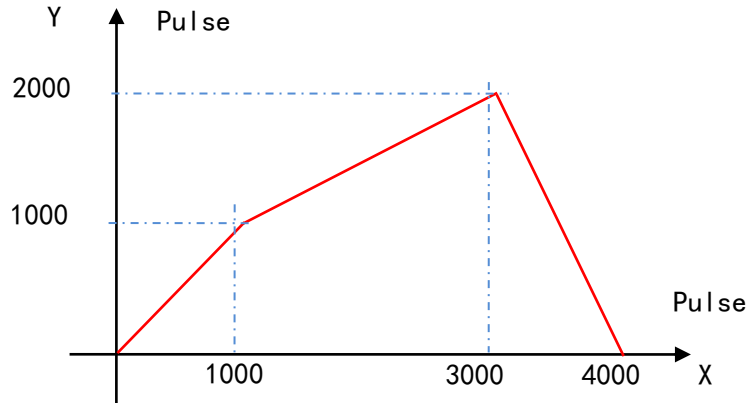
When the selected interpolation type is linear interpolation, if the trajectories of two adjacent interpolation motions are tangent or corner, the planned combined velocity will be continuous, if the corner is too large, the velocity will be discontinuous; when the selected interpolation type is circular interpolation, if the trajectories of two adjacent

interpolation motions are tangent at the intersection point, the velocity will be continuous, if they are not tangent, the velocity will be discontinuous.

After starting the continuous interpolation instruction, the instruction needs to carry out the operation of compression cache first, and the more interpolation points, the longer the time of compression cache will be.

### ☉ Program demo

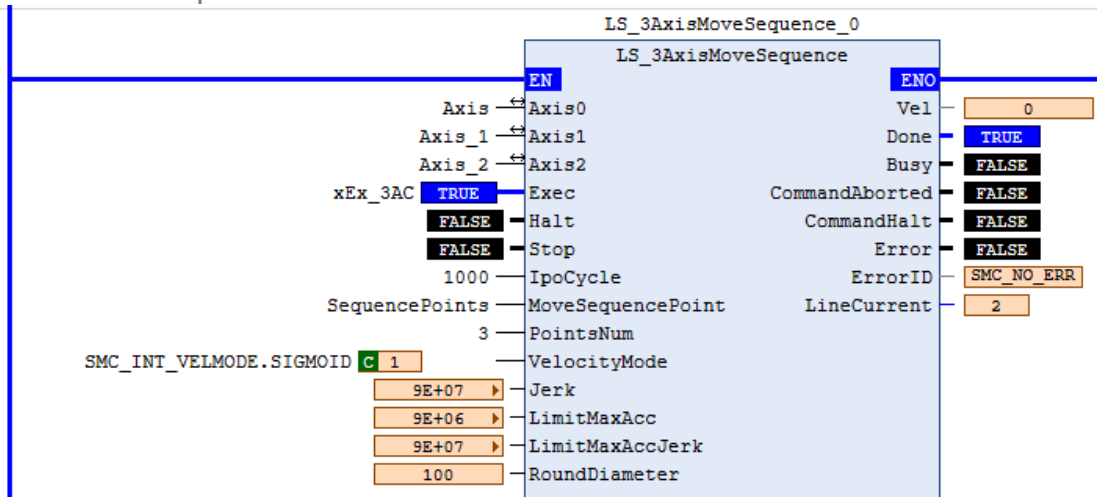
**LD:** Write a programme to achieve XYZ axis with continuous interpolation instruction to carry out three segments of linear continuous interpolation, the speed of all 1000Pulse/s, plus or minus 2000Pulse/s<sup>2</sup>, the first segment of the endpoint coordinates (1000, 1000, 0), the second segment of the endpoint coordinates (3000, 2000, 0), and the third segment of the endpoint coordinates (4000, 0, 0), as shown in the figure.



Add the parameters related to the data points in the main programme. Once the program is complete, compile and download it to the controller for execution and force the variable iState to 1 to initiate continuous interpolation.

```

3  ● CASE iState[ 0 ] OF
4      1://配置连续插补数据点参数
5      ● FOR i[ 0 ]:=0 TO 5 DO
6      ● SequencePoints[i[ 0 ]].Acc[ 0 ]:=20000;
7      ● SequencePoints[i[ 0 ]].DataType[ 0 ]:=1;
8      ● SequencePoints[i[ 0 ]].LineNum[ 0 ]:=i[ 0 ];
9      ● SequencePoints[i[ 0 ]].Vel[ 0 ]:=10000;
10     ● SequencePoints[i[ 0 ]].Z_EndPos[ 0 ]:=0;
11     END_FOR
12     SequencePoints[0].X_EndPos[ 0 ]:=1000;
13     SequencePoints[0].Y_EndPos[ 0 ]:=1000;
14     SequencePoints[1].X_EndPos[ 0 ]:=3000;
15     SequencePoints[1].Y_EndPos[ 0 ]:=2000;
16     SequencePoints[2].X_EndPos[ 0 ]:=4000;
17     SequencePoints[2].Y_EndPos[ 0 ]:=0;
18     iState[ 0 ]:=2;
19     2://启动运动
20     xEx_3AC[FALSE]:=TRUE;
21     iState[ 0 ]:=3;
22     3://等待运动完成
23     IF LS_3AxisMoveSequence_0.Done[FALSE] THEN
24     xEx_3AC[FALSE]:=FALSE;
25     iState[ 0 ]:=0;
26     END_IF
27     END_CASE
28     ACT();RETURN
    
```



**Note**

The IpoCycle parameter needs to be aligned with the task cycle time where the motion instruction is located, otherwise an Error may be reported when the instruction is executed.

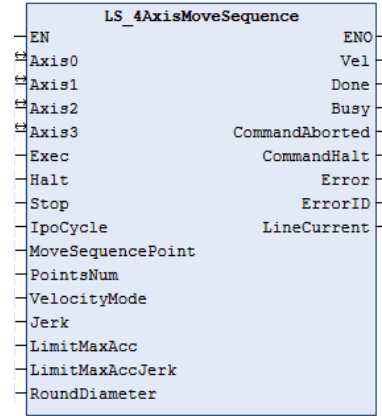
The parameters IpoCycle, Jerk, Velocity, Acceleration and Deceleration cannot be set to 0.

The master and slave axes can not be called by other motion instructions during the instruction running time.

### 4.4.14 LS\_4AxisMoveSequence

Four-axis continuous interpolation command with continuous speed.

#### ⊙ Command Format

Instruction	Name	FB/FC	LD	ST	File
LS_4AxisMoveSequence	Four axis continuous interpolation motion instruction	FC		<pre> LS_4AxisMoveSequence( Axis0:= , Axis1:= , Axis2:= , Axis3:= , Exec:= , Halt:= , Stop:= , IpoCycle:= , MoveSequencePoint:= , PointsNum:= , VelocityMode:= , Jerk:= , LimitMaxAcc:= , LimitMaxAccJerk:= , RoundDiameter:= , Vel=&gt; , Done=&gt; , Busy=&gt; , CommandAborted=&gt; , CommandHalt=&gt; , Error=&gt; , ErrorID=&gt; , LineCurrent=&gt; ); </pre>	LS_IpoLib

#### ⊙ Related Variables

##### I/O variable

I/O variable	Name	Data type	Range	initialization	Descriptive
Axis0	Axis0	AXIS_REF_VIRTUAL_SM3	-	-	The 0-axis involved in interpolation
Axis1	Axis1	AXIS_REF_VIRTUAL_SM3	-	-	The 1-axis involved in interpolation
Axis2	Axis2	AXIS_REF_VIRTUAL_SM3	-	-	The 2-axis involved in interpolation
Axis3	Axis3	AXIS_REF_VIRTUAL_SM3	-	-	Follow axis, speed discontinuity

##### Input variable

Input variable	Name	Data type	Range	initialization	Descriptive
Exec	Start	BOOL	TRUE-FALSE	FALSE	Rising edge trigger
Halt	Pause	BOOL	TRUE-FALSE	FALSE	TRUE: the interpolation motion is suspended; when the state is switched to False again, the interpolation module continues the previous unfinished interpolation tasks
Stop	Stop	BOOL	TRUE-FALSE	FALSE	TRUE: The interpolation module is stopped.
IpoCycle	Interpolation cycle	DWORD	ALL	2000	Interpolation period, unit: us.
MoveSequencePoint	Interpolation Data Sheet	ARRAY [0..MoveSequence_MAX] OF MoveSequence	-	-	Input data points.
PointsNum	check numbers	UINT	positive number	0	Number of data points entered.
VelocityMode	Velocity mode	SMC_INT_VE_LMODE	0-3	SIGMOID	Velocity mode: Trapezoidal: 0 (TRAPEZOID);

					S-shaped: 1 (SIGMOID); Quadratic: 3 (QUADRATIC).
Jerk	Acceleration	REAL	positive number	90000000	Acceleration, you need to set this parameter when the speed mode is 3.
LimitMaxAcc	Maximum acceleration	REAL	positive number	9000000	Maximum acceleration limit for corner transition arc.
LimitMaxAccJerk	Maximum deceleration	REAL	positive number	90000000	Maximum acceleration limit for corner transition arc.
RoundDiameter	Corner radius	REAL	positive number	100	Corner transition radius, unit: Pulse.

**Output variable**

Output variable	Name	Data type	Range	initialization	Descriptive
Vel	Closing speed	LREAL	0,positive number	0	Kinematic synthetic velocity
Done	Completion	BOOL	TRUE-FALSE	FALSE	TRUE: Homing is done.
Busy	Executing	BOOL	TRUE-FALSE	FALSE	TRUE: In progress
Command Aborted	Command Interrupt	BOOL	TRUE-FALSE	FALSE	TRUE: Terminated by other commands
CommandHalt	motion pause	BOOL	TRUE, FALSE	FALSE	When Halt is True, this variable is True and the interpolation module is in a suspended state.
Error	Error	BOOL	TRUE-FALSE	FALSE	TRUE: An error occurred within the function block
ErrorID	Error Code	SMC_ERROR	-	0	Error indication, see SMC Error.
LineCuren	Line Number	UINT	Follows the data type	0	The line number of the current execution.

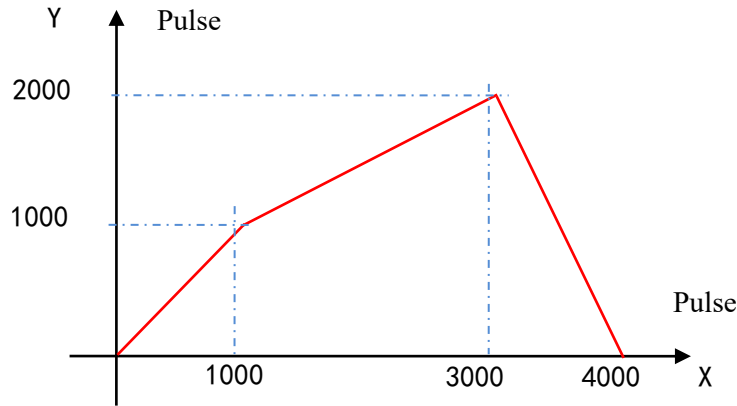
	Boo le an	Bit string					Integer							Real number		Moment, Duration, Date, String				
	BOOL	BYTE	WORD	DWORD	LWORD	USINT	UINT	UDINT	ULINT	SINT	INT	DINT	LINT	REAL	LREAL	TIME	DATE	TOD	DT	STRING
Axis		AXIS_REF_SM3																		
Exec	√	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
Halt	√	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
Stop	√	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
Endpos	-	-	-	-	-	-	-	-	-	-	-	-	-	√	-	-	-	-	-	-
Velocity	-	-	-	-	-	-	-	-	-	-	-	-	-	√	-	-	-	-	-	-
Acceleration	-	-	-	-	-	-	-	-	-	-	-	-	-	√	-	-	-	-	-	-
Deceleration	-	-	-	-	-	-	-	-	-	-	-	-	-	√	-	-	-	-	-	-
Jerk	-	-	-	-	-	-	-	-	-	-	-	-	-	√	-	-	-	-	-	-
Done	√	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
Busy	√	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
Command Aboned	√	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
Error	√	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
ErrorID		SMC_ERROR																		

**⊙ Functional Description**

Specify Axis0-Axis2 for continuous interpolation motion and Axis3 for following motion, the position coordinates are absolute coordinates, and up to 200 interpolation motion points can be input at one time.  
The settings of other parameters are the same as the LS\_3AxisMoveSequence instruction.

**⊙ Program demo**

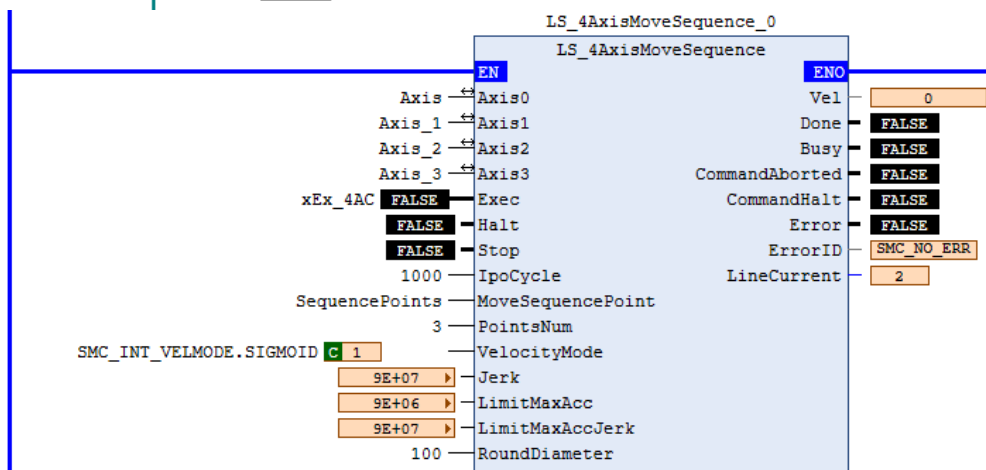
**LD:** Write a programme to achieve XYZ axis with continuous interpolation instruction to carry out three segments of linear continuous interpolation, the speed of all 1000Pulse/s, plus or minus 2000Pulse/s<sup>2</sup>, the first segment of the endpoint coordinates (1000, 1000, 0), the second segment of the endpoint coordinates (3000, 2000, 0), and the third segment of the endpoint coordinates (4000, 0, 0), as shown in the figure.



Add the parameters related to the data points in the main programme. Once the program is complete, compile and download it to the controller for execution and force the variable iState to 1 to initiate continuous interpolation.

```

3 ● CASE iState 0 OF
4   1://配置连续插补数据点参数
5     FOR i 4 :=0 TO 3 DO
6       SequencePoints[i 4].Acc 0 :=20000;
7       SequencePoints[i 4].DataType 0 :=1;
8       SequencePoints[i 4].LineNum 0 :=i 4;
9       SequencePoints[i 4].U_EndPos 0 :=0;
10      SequencePoints[i 4].Vel 0 :=10000;
11      SequencePoints[i 4].Z_EndPos 0 :=0;
12    END_FOR
13    SequencePoints[0].X_EndPos 1E+03 :=1000;
14    SequencePoints[0].Y_EndPos 1E+03 :=1000;
15    SequencePoints[1].X_EndPos 3E+03 :=3000;
16    SequencePoints[1].Y_EndPos 2E+03 :=2000;
17    SequencePoints[2].X_EndPos 4E+03 :=4000;
18    SequencePoints[2].Y_EndPos 0 :=0;
19    iState 0 :=2;
20  2://启动运动
21    xEx_4AC FALSE :=TRUE;
22    iState 0 :=3;
23  3://等待运动完成
24    IF LS_4AxisMoveSequence_0.Done FALSE THEN
25      xEx_4AC FALSE :=FALSE;
26      iState 0 :=0;
27    END_IF
28  END_CASE
29 ● ACT();RETURN
  
```



**Note**

The IpoCycle parameter needs to be aligned with the task cycle time where the motion instruction is located, otherwise an Error may be reported when the instruction is executed.

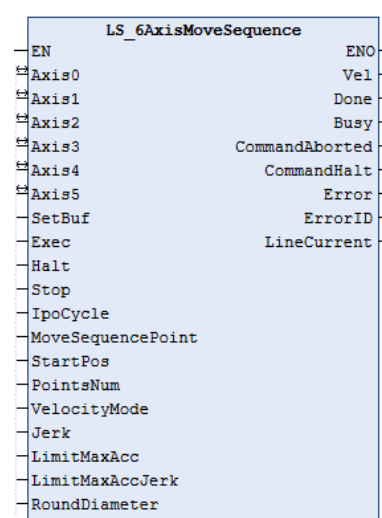
The parameters IpoCycle, Jerk, Velocity, Acceleration and Deceleration cannot be set to 0.

The master and slave axes can not be called by other motion instructions during the instruction running time.

### 4.4.15 LS\_6AxisMoveSequence

Four-axis continuous interpolation command with continuous speed.

#### ⊙ Command Format

Instruction	Name	FB/FC	LD	ST	File
LS_6AxisMoveSequence	Six axis continuous interpolation motion instruction	FC		<pre> LS_6AxisMoveSequence( Axis0:= , Axis1:= , Axis2:= , Axis3:= , Axis4:= , Axis5:= , SetBuf:= , Exec:= , Halt:= , Stop:= , IpoCycle:= , MoveSequencePoint:= , StartPos:= , PointsNum:= , VelocityMode:= , Jerk:= , LimitMaxAcc:= , LimitMaxAccJerk:= , RoundDiameter:= , Vel=&gt; , Done=&gt; , Busy=&gt; , CommandAborted=&gt; , CommandHalt=&gt; , Error=&gt; , ErrorID=&gt; , LineCurrent=&gt; ); </pre>	LS_Ipo Lib

#### ⊙ Related Variables

##### I/O variable

I/O variable	Name	Data type	Range	initialization	Descriptive
Axis0	Axis0	AXIS_REF_VIRTUAL_SM3	-	-	The 0-axis involved in interpolation
Axis1	Axis1	AXIS_REF_VIRTUAL_SM3	-	-	The 1-axis involved in interpolation
Axis2	Axis2	AXIS_REF_VIRTUAL_SM3	-	-	The 2-axis involved in interpolation
Axis3	Axis3	AXIS_REF_VIRTUAL_SM3	-	-	Follow axis, speed discontinuity
Axis4	Axis4	AXIS_REF_VIRTUAL_SM3	-	-	Follow axis, speed discontinuity
Axis5	Axis5	AXIS_REF_VIRTUAL_SM3	-	-	Follow axis, speed discontinuity

##### Input variable

Input variable	Name	Data type	Range	initialization	Descriptive
SetBuf	Trigger press-in data	BOOL	TRUE, FALSE	FALSE	The interpolated data is pressed into the cache.
Exec	Start	BOOL	TRUE-FALSE	FALSE	Rising edge trigger
Halt	Pause	BOOL	TRUE-FALSE	FALSE	TRUE: the interpolation motion is suspended; when the state is switched to False again, the interpolation module continues the previous unfinished interpolation tasks
Stop	Stop	BOOL	TRUE-FALSE	FALSE	TRUE: The interpolation module is stopped.

IpoCycle	Interpolation cycle	DWORD	ALL	2000	Interpolation period, unit: us.
MoveSequencePoint	Interpolation Data Sheet	ARRAY [0..MoveSequence_MAX] OF MoveSequence	-	-	Input data points.
PointsNum	check numbers	UINT	positive number	0	Number of data points entered.
VelocityMode	Velocity mode	SMC_INT_VE LMODE	0-3	SIGMOID	Velocity mode: Trapezoidal: 0 (TRAPEZOID); S-shaped: 1 (SIGMOID); Quadratic: 3 (QUADRATIC).
Jerk	Acceleration	REAL	positive number	90000000	Acceleration, you need to set this parameter when the speed mode is 3.
LimitMaxAcc	Maximum acceleration	REAL	positive number	9000000	Maximum acceleration limit for corner transition arc.
LimitMaxAccJerk	Maximum deceleration	REAL	positive number	90000000	Maximum acceleration limit for corner transition arc.
RoundDiameter	Corner radius	REAL	positive number	100	Corner transition radius, unit: Pulse.

**Output variable**

Output variable	Name	Data type	Range	initialization	Descriptive
Vel	Closing speed	LREAL	0, positive number	0	Kinematic synthetic velocity
Done	Completion	BOOL	TRUE-FALSE	FALSE	TRUE: Homing is done.
Busy	Executing	BOOL	TRUE-FALSE	FALSE	TRUE: In progress
Command Aborted	Command Interrupt	BOOL	TRUE-FALSE	FALSE	TRUE: Terminated by other commands
CommandHalt	motion pause	BOOL	TRUE, FALSE	FALSE	When Halt is True, this variable is True and the interpolation module is in a suspended state.
Error	Error	BOOL	TRUE-FALSE	FALSE	TRUE: An error occurred within the function block
ErrorID	Error Code	SMC_ERROR	-	0	Error indication, see SMC_Error.
LineCuren	Line Number	UINT	Follows the data type	0	The line number of the current execution.

	Bo	Bit string					Integer							Real number		Moment, Duration, Date, String					
	olean	BOOL	BYTE	WORD	DWORD	LWORD	USINT	UINT	UDINT	ULINT	SINT	INT	DINT	LINT	REAL	LREAL	TIME	DATE	TOD	DT	STRING
Axis		AXIS_REF_SM3																			
Exec	√	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
Halt	√	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
Stop	√	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
Endpos	-	-	-	-	-	-	-	-	-	-	-	-	-	-	√	-	-	-	-	-	-
Velocity	-	-	-	-	-	-	-	-	-	-	-	-	-	-	√	-	-	-	-	-	-
Acceleration	-	-	-	-	-	-	-	-	-	-	-	-	-	-	√	-	-	-	-	-	-
Deceleration	-	-	-	-	-	-	-	-	-	-	-	-	-	-	√	-	-	-	-	-	-
Jerk	-	-	-	-	-	-	-	-	-	-	-	-	-	-	√	-	-	-	-	-	-
Done	√	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-

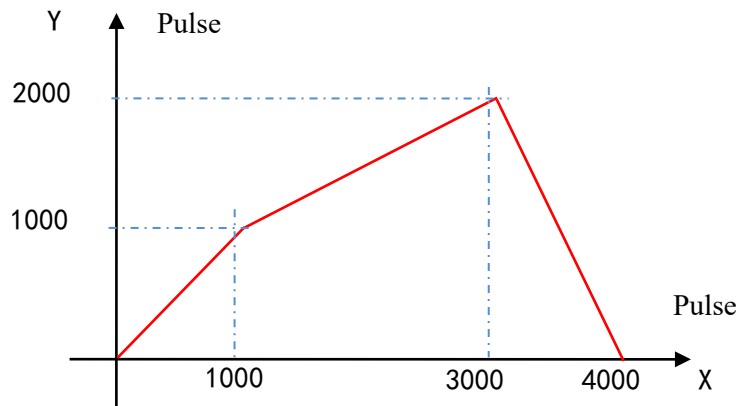
Busy	√	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
Command Aboned	√	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
Error	√	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
ErrorID	SMC_ERROR																			

⊙ **Functional Description**

Six-axis continuous interpolation and three-axis following motion. A maximum of MoveSequence\_MAX points can be entered at one time. Parameters IpoCycle, Jerk, LimitMaxAcc, LimitMaxAccJerk, and RoundDiameter cannot be set to 0. The LT\_6AxisMoveSequence instruction triggers the execution of the pressure buffer and the instruction after setting Exec to TRUE.

⊙ **Program demo**

**LD:** Write a program to achieve three segments of X/Y/Z/U/V/W axes for continuous linear interpolation with continuous interpolation instruction, the speed of all of them is 1000Pulse/s, the acceleration and deceleration speed is 2000Pulse/s<sup>2</sup>, the end point coordinates of the first segment are (1000, 1000, 0, 0, 0, 0), the end point coordinates of the second segment are (3000, 2000, 0, 0, 0, 0), the third segment endpoint coordinates are (4000, 0, 0, 0, 0, 0), as shown in Fig.

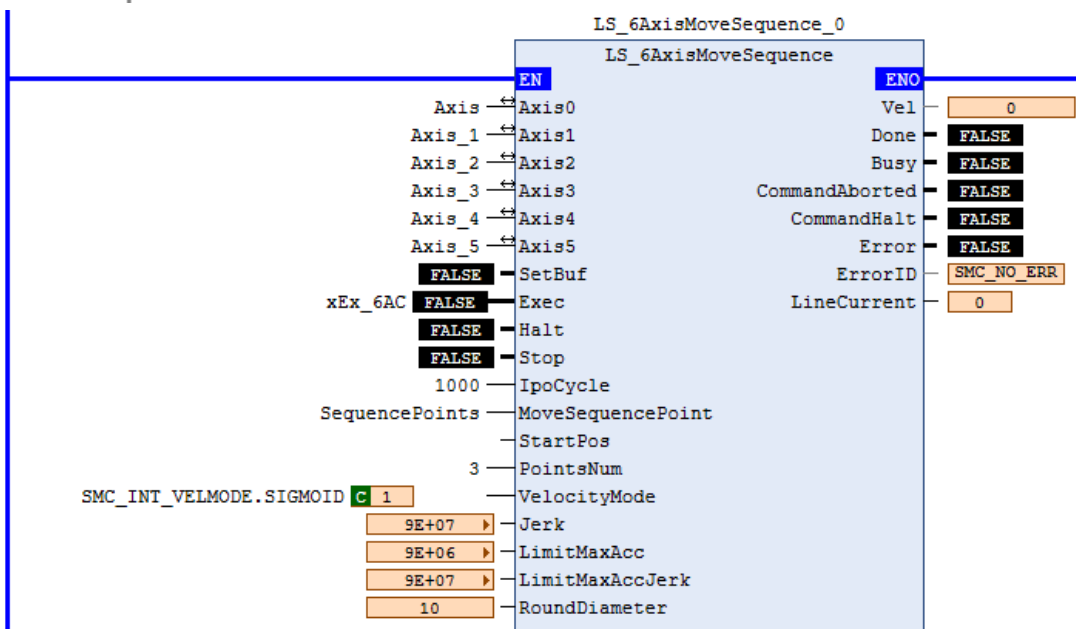


Add the parameters related to the data points in the main programme. Once the program is complete, compile and download it to the controller for execution and force the variable Exec to 1 to initiate continuous interpolation.

```

3 ● CASE iState 0 OF
4   1://配置连续插补数据点参数
5 ●   FOR i 0 :=0 TO 5 DO
6 ●     SequencePoints[i 0].Acc 0 :=20000;
7 ●     SequencePoints[i 0].DataType 0 :=1;
8 ●     SequencePoints[i 0].LineNum 0 :=i 0;
9 ●     SequencePoints[i 0].Vel 0 :=10000;
10 ●    SequencePoints[i 0].Z_EndPos 0 :=0;
11 ●    SequencePoints[i 0].U_EndPos 0 :=0;
12 ●    SequencePoints[i 0].V_EndPos 0 :=0;
13 ●    SequencePoints[i 0].W_EndPos 0 :=0;
14 ●    END_FOR
15 ●    SequencePoints[0].X_EndPos 0 :=1000;
16 ●    SequencePoints[0].Y_EndPos 0 :=1000;
17 ●    SequencePoints[1].X_EndPos 0 :=3000;
18 ●    SequencePoints[1].Y_EndPos 0 :=2000;
19 ●    SequencePoints[2].X_EndPos 0 :=4000;
20 ●    SequencePoints[2].Y_EndPos 0 :=0;
21 ●    iState 0 :=2;
22 ●    2://启动运动
23 ●    xEx_6AC FALSE :=TRUE;
24 ●    iState 0 :=3;
25 ●    3://等待运动完成
26 ●    IF LS_6AxisMoveSequence_0.Done FALSE THEN
27 ●      xEx_6AC FALSE :=FALSE;
28 ●      iState 0 :=0;
29 ●    END_IF
30 ●  END_CASE
31 ●  ACT (); RETURN

```


**Note**

The IpoCycle parameter needs to be aligned with the task cycle time where the motion instruction is located, otherwise an Error may be reported when the instruction is executed.

The parameters IpoCycle, Jerk, Velocity, Acceleration and Deceleration cannot be set to 0.

The master and slave axes can not be called by other motion instructions during the instruction running time.

## 5. Communications instruction

### 5.1 Free Communications TCP

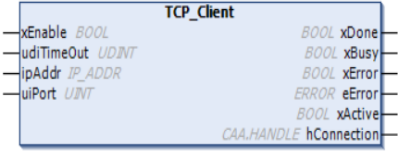
#### 5.1.1 Instruction List

Instruction Type	Name	FB/FC	Functionality
Free Communications TCP	TCP_Client	FB	Creating a TCP Client Communication Service
	TCP_Write	FB	TCP Communication data transmission
	TCP_Read	FB	TCP Communication data reception
	TCP_Connection	FB	Create a TCP connection and connect to the server
	TCP_Server	FB	Creating a TCP server-side communication service

#### 5.1.2 TCP\_Client

Creating a TCP client communication service

##### ⊙ Command Format

Instruction	Name	FB/FC	LD	ST	File
TCP_Client	Creating a TCP Client Communication Service instruction	FB		<pre>TCP_Client( xEnable:= , xDone=&gt; , xBusy=&gt; , xError=&gt; , udiTimeOut:= , ipAddr:= , uiPort:= , eError=&gt; , xActive=&gt; , hConnection=&gt; )</pre>	CAA Net Base Services

##### ⊙ Related Variables

###### Input variable

Input variable	Name	Data type	Range	initialization	Descriptive
xEnable	Function Block Enable	BOOL	TRUE-FALSE	-	TRUE: start function block running
udiTimeOut	Timeout	UDINT	0- 4294967295	0	Timeout for requesting connection establishment. Unit: μs
ipAddr	Server IP address	NBS.IP ADDR	-	-	IP address of the server to which the client is connected, refer to structure IP_ADDR in CAA Net Base Services library.
uiPort	Port number	UINT	0-65535	-	Port number of the server to which the client is connected

###### Output variable

Output variable	Name	Data type	Range	initialization	Descriptive
xDone	Completion	BOOL	TRUE-FALSE	-	Client communication complete
xBusy	Function Block Running	BOOL	TRUE-FALSE	-	Function block running flag
xError	Error Flag	BOOL	TRUE-FALSE	-	Error Status

eError	Error	NBS.ERROR	--	-	Please refer to the CAA Net Base Services library enumeration ERROR for specific errors.
xActive	Connection success flag	BOOL	TRUE-FALSE	-	Client-server connection success flag
hConnection	Connection handle	CAA.HANDLE	--	-	Client-server connection handle

	Boo le an	Bit string					Integer							Real number		Moment, Duration, Date, String				
	BOO L	BY TE	WO RD	DWO RD	LWO RD	USI NT	UI NT	UDI NT	ULI NT	SIN T	IN T	DIN T	LIN T	REA L	LR EAL	TI ME	DA TE	TO D	DT	STR ING
xEnable	√	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
udiTimeOut	-	-	-	-	-	-	-	√	-	-	-	-	-	-	-	-	-	-	-	-
ipAddr	NBS.IP ASSR																			
uiPort	-	-	-	-	-	-	-	√	-	-	-	-	-	-	-	-	-	-	-	-
xDone	√	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
xBusy	√	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
xError	√	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
eError	NBS.ERROR																			
xActive	√	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
hConnec tion	CAA.HANDLE																			

### ⊙ Functional Description

Create a TCP client communication service.

The data type of 'ipAddr' is structure IP\_ADDR, and the meanings of the structure members are shown in the following table.

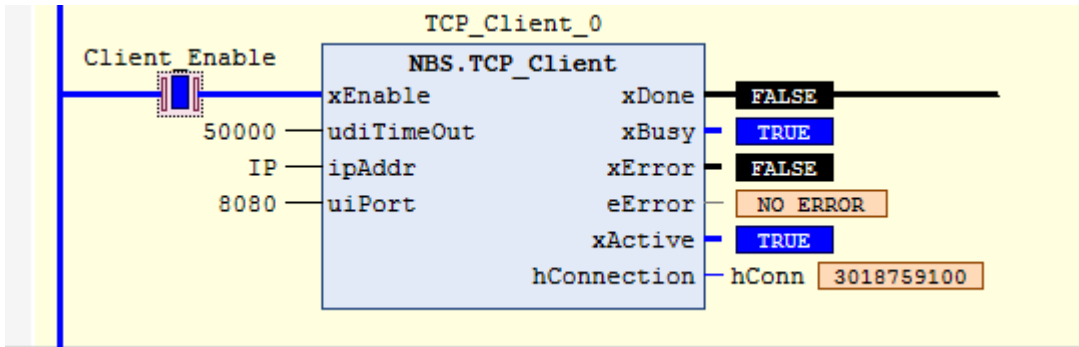
Structures Members	Data Type	Description of the structure
sAddr	STRING(80)	IP address of the server to which the client is connected

### ⊙ Program demo

**ST:** Create a TCP client communication service. When the TCP\_Client directive xEnable is TRUE, a valid handle for TCP communication between the local server side and the remote client side will be created (hServer is greater than 0) and xBusy is TRUE.

```
TCP_Client_0(
  xEnable TRUE := Client_Enable TRUE,
  xDone=>,
  xBusy TRUE => Client_Busy TRUE,
  xError=>,
  udiTimeOut 50000 := 50000,
  ipAddr:=IP,
  uiPort 8080 := 8080,
  eError=>,
  xActive=>,
  hConnection 3018759100 => hConn 3018759100);
```

**LD:** Create a TCP client communication service. When the TCP\_Client directive xEnable is TRUE, a valid handle for TCP communication between the local server side and the remote client side will be created (hServer is greater than 0) and xBusy is TRUE.



**Note**

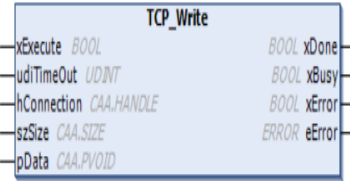
If there is an error in the function block, xError will be set to TRUE, eError will be assigned the relevant error value (various values in the enumerated ERROR) , and hConnection will be invalid (equal to 0). See the following error codes for details..

Error Code	Definition	Descripción
0	NO_ERROR	No errors
6000	FIRST_ERROR	Reservations
6001	TIME_OUT	Reserved
6002	INVALID_ADDR	The IP address of the server to which the client is connected is invalid
6003	INVALID_HANDLE	Connection handle is invalid
6004	INVALID_DATAPOINTER	Invalid data pointer
6005	INVALID_DATASIZE	Invalid data size
6006	UDP_RECEIVE_ERROR	UDP receive error
6007	UDP_SEND_ERROR	UDP transmit error
6008	UDP_SEND_NOT_COMPLETE	UDP send incomplete
6009	UDP_OPEN_ERROR	UDP open error
6010	UDP_CLOSE_ERROR	UDP close error
6011	TCP_SEND_ERROR	TCP send error
6012	TCP_RECEIVE_ERROR	TCP send incomplete
6013	TCP_OPEN_ERROR	TCP open error
6014	TCP_CONNECT_ERROR	TCP connection error
6015	TCP_CLOSE_ERROR	TCP close error
6016	TCP_SERVER_ERROR	TCP server error
6017	WRONG_PARAMETER	Parameter error
6018	ERROR_UNKNOWN	Unknown error
6019	TCP_NO_CONNECTION	No TCP connection
6020	LOCTL_ERROR	Internal error (not supported by this machine)
6050	FIRST_MF	Reserved
6099	LAST_ERROR	Reserved

### 5.1.3 TCP\_Write

Writes data to the connection established at hConnection.

#### ⊙ Command Format

Instruction	Name	FB/FC	LD	ST	File
TCP_Write	TCP Communication data transmission instruction	FB		TCP_Write( xExecute:= , udiTimeout:= , xDone=> , xBusy=> , xError=> , hConnection:= , szSize:= , pData:= , eError=> )	CAA Net Base Services

#### ⊙ Related Variables

##### Input variable

Input variable	Name	Data type	Range	initialization	Descriptive
xEnable	Function Block Enable	BOOL	TRUE-FALSE	-	TRUE: start function block running
udiTimeout	Timeout	UDINT	0- 4294967295	0	Timeout for requesting connection establishment. Unit: μs
hConnection	Connection handle	CAA.HANDLE	-	-	Client-server connection handle
szSize	Data Size	CAA.SIZE	-	-	Size of the sent data area, byte
pData	Send Cache	CAA.PVOID	-	-	Pointer to the attribute data to send to the target device

##### Output variable

Output variable	Name	Data type	Range	initialization	Descriptive
xDone	Completion	BOOL	TRUE-FALSE	-	Client communication complete
xBusy	Function Block Running	BOOL	TRUE-FALSE	-	Function block running flag
xError	Error Flag	BOOL	TRUE-FALSE	-	Error Status
eError	Error	NBS.ERROR	--	-	Please refer to the CAA Net Base Services library enumeration ERROR for specific errors.

	Boo lea n	Bit string				Integer							Real number		Moment, Duration, Date, String					
	BOOL	BYTE	WORD	DWORD	LWORD	USINT	UINT	UDINT	ULINT	SINT	INT	DINT	LINT	REAL	LREAL	TIME	DATE	TOD	DT	STRING
xEnable	√	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
udiTimeout	-	-	-	-	-	-	-	√	-	-	-	-	-	-	-	-	-	-	-	-
hConnection	CAA.HANDLE																			
szSize	CAA.SIZE																			
pData	CAA.PVOID																			

xDone	√	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
xBusy	√	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
xError	√	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
eError	NBS.ERROR																			

**Program demo**

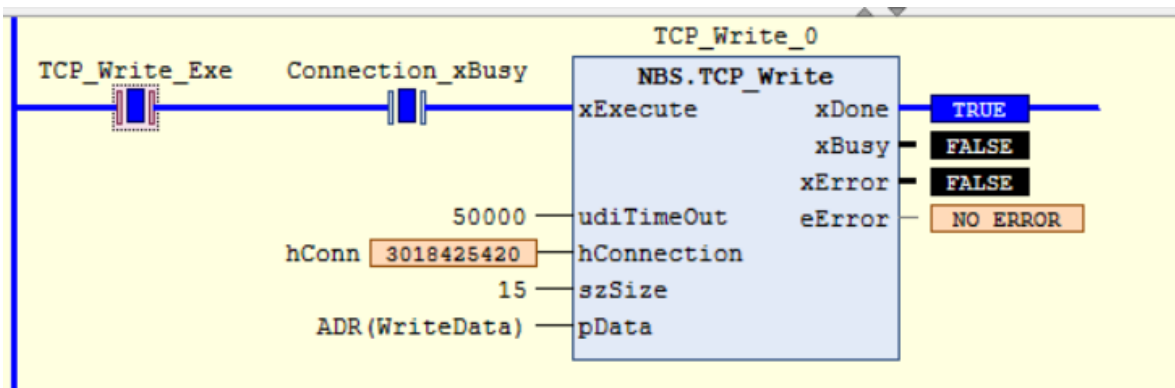
**ST:** This function block serves to write data to the connection established at hConnection. When the TCP\_Write command xExecute is TRUE, data of length szSize is sent to the target device with the send buffer pData set by the user as the first address. If the send is successful within the timeout period, xDone is set to TRUE.

```

1  TCP_Write_0(
2    xExecute := TCP_Write_Exe TRUE,
3    udiTimeOut := 50000,
4    xDone=> ,
5    xBusy=> ,
6    xError=> ,
7    hConnection := hConn 3018759840 ,
8    szSize := 15 ,
9    pData := ADR(WriteData) ,
10   eError=> );
11

```

**LD:** This function block serves to write data to the connection established at hConnection. When the TCP\_Write command xExecute is TRUE, data of length szSize with the user set send buffer pData as the first address is sent to the target device. If the send is successful within the timeout period, xDone is set to TRUE.


**Note**

If there is an error in the function block, xError will be set to TRUE, eError will be assigned the relevant error value (various values in the enumerated ERROR) and hConnection will be invalid (equal to 0). See the following error codes for details:

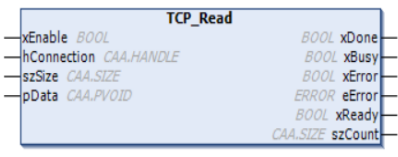
Error Code	Definition	Descripción
0	NO_ERROR	No errors
6000	FIRST_ERROR	Reservations
6001	TIME_OUT	Reserved
6002	INVALID_ADDR	The IP address of the server to which the client is connected is invalid
6003	INVALID_HANDLE	Connection handle is invalid
6004	INVALID_DATAPOINTER	Invalid data pointer
6005	INVALID_DATASIZE	Invalid data size

6006	UDP_RECEIVE_ERROR	UDP receive error
6007	UDP_SEND_ERROR	UDP transmit error
6008	UDP_SEND_NOT_COMPLETE	UDP send incomplete
6009	UDP_OPEN_ERROR	UDP open error
6010	UDP_CLOSE_ERROR	UDP close error
6011	TCP_SEND_ERROR	TCP send error
6012	TCP_RECEIVE_ERROR	TCP send incomplete
6013	TCP_OPEN_ERROR	TCP open error
6014	TCP_CONNECT_ERROR	TCP connection error
6015	TCP_CLOSE_ERROR	TCP close error
6016	TCP_SERVER_ERROR	TCP server error
6017	WRONG_PARAMETER	Parameter error
6018	ERROR_UNKNOWN	Unknown error
6019	TCP_NO_CONNECTION	No TCP connection
6020	LOCTL_ERROR	Internal error (not supported by this machine)
6050	FIRST_MF	Reserved
6099	LAST_ERROR	Reserved

### 5.1.4 TCP\_Read

Reads data from the communication buffer of the connection established by hConnection.

#### ⊙ Command Format

Instruction	Name	FB/FC	LD	ST	File
TCP_Read	TCP Communication data reception instruction	FB		<pre>TCP_Read( xEnable:= , xDone=&gt; , xBusy=&gt; , xError=&gt; , hConnection:= , szSize:= , pData:= , eError=&gt; , xReady=&gt; , szCount=&gt; );</pre>	CAA Net Base Services

#### ⊙ Related Variables

##### Input variable

Input variable	Name	Data type	Range	initialization	Descriptive
xEnable	Function Block Enable	BOOL	TRUE-FALSE	-	TRUE: start function block running
hConnection	Connection handle	CAA.HANDLE	-	-	Client-server connection handle
szSize	Data Size	CAA.SIZE	-	-	Size of the sent data area, byte
pData	Send Cache	CAA.PVOID	-	-	Pointer to the attribute data to send to the target device

##### Output variable

Output variable	Name	Data type	Range	initialization	Descriptive
xDone	Completion	BOOL	TRUE-FALSE	-	Client communication complete
xBusy	Function Block Running	BOOL	TRUE-FALSE	-	Function block running flag
xError	Error Flag	BOOL	TRUE-FALSE	-	Error Status
eError	Error	NBS.ERROR	--	-	Please refer to the CAA Net Base Services library enumeration ERROR for specific errors.
xReady	Successful connection flag	BOOL	TRUE-FALSE	-	Read data from the buffer and set the flag bit for one scan cycle if the data is not empty
szCount	Data Size	CAA.SIZE	-	-	Actual size of the received data area

	Boo	Bit string					Integer					Real number		Moment, Duration, Date, String					
	lean	WORD	DWORD	LWORD	USINT	UINT	UDINT	ULINT	SINT	INT	DINT	LINT	REAL	LREAL	TIME	DATE	TOD	DT	STRING
xEnable	√	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-

hConnection	CAA.HANDLE																		
szSize	CAA.SIZE																		
pData	CAA.PVOID																		
xDone	√	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
xBusy	√	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
xError	√	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
eError	NBS.ERROR																		
xReady	√	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
szCount	CAA.SIZE																		

**⊙ Program demo**

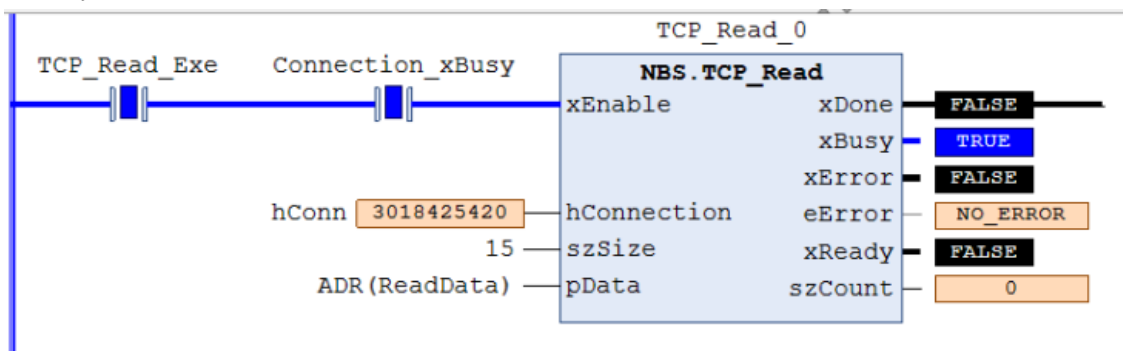
**ST:** Reads data from the communication buffer of the connection established by hConnection. When the TCP\_Read instruction xEnable is TRUE, the data will be read from the TCP communication buffer, and xBusy is TRUE. xDone will be set for one scan cycle if the read is successful; the read data will be placed into the variable with the address pData; at the same time, xReady will be set for one scan cycle; the value of the actual size of the received data area will be given to szCount, and the value of szCount will be cleared after one scan cycle. The actual size of the received data area will be assigned to szCount, and the value of szCount will be cleared to zero after one scan cycle.

```

1
2 ● TCP_Read_0 (
3   xEnable TRUE := TCP_Read_Exe TRUE ,
4   xDone=> ,
5   xBusy=> ,
6   xError=> ,
7   hConnection 3017879200 := hConn 3017879200 ,
8   szSize 15 :=15 ,
9   pData 3017879692 :=ADR (ReadData) ,
10  eError=> ,
11  xReady=> ,
12  szCount=> );
13
14
15

```

**LD:** Reads data from the communication buffer of the connection established by hConnection. When the TCP\_Read instruction xEnable is TRUE, the data will be read from the TCP communication buffer, and xBusy is TRUE. xDone will be set for one scan cycle if the read is successful; the read data will be placed into the variable with the address pData; at the same time, xReady will be set for one scan cycle; the value of the actual size of the received data area will be given to szCount, and the value of szCount will be cleared after one scan cycle. The actual size of the received data area will be assigned to szCount, and the value of szCount will be cleared to zero after one scan cycle.


**Note**

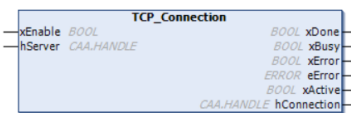
If there is an error in the function block, xError will be set to TRUE, eError will be assigned the relevant error value (various values in the enumerated ERROR) , and hConnection will be invalid (equal to 0). See the following error codes for details:.

<b>Error Code</b>	<b>Definition</b>	<b>Descripción</b>
0	NO_ERROR	No errors
6000	FIRST_ERROR	Reservations
6001	TIME_OUT	Reserved
6002	INVALID_ADDR	The IP address of the server to which the client is connected is invalid
6003	INVALID_HANDLE	Connection handle is invalid
6004	INVALID_DATAPOINTER	Invalid data pointer
6005	INVALID_DATASIZE	Invalid data size
6006	UDP_RECEIVE_ERROR	UDP receive error
6007	UDP_SEND_ERROR	UDP transmit error
6008	UDP_SEND_NOT_COMPLETE	UDP send incomplete
6009	UDP_OPEN_ERROR	UDP open error
6010	UDP_CLOSE_ERROR	UDP close error
6011	TCP_SEND_ERROR	TCP send error
6012	TCP_RECEIVE_ERROR	TCP send incomplete
6013	TCP_OPEN_ERROR	TCP open error
6014	TCP_CONNECT_ERROR	TCP connection error
6015	TCP_CLOSE_ERROR	TCP close error
6016	TCP_SERVER_ERROR	TCP server error
6017	WRONG_PARAMETER	Parameter error
6018	ERROR_UNKNOWN	Unknown error
6019	TCP_NO_CONNECTION	No TCP connection
6020	LOCTL_ERROR	Internal error (not supported by this machine)
6050	FIRST_MF	Reserved
6099	LAST_ERROR	Reserved

### 5.1.5 TCP\_Connection

Reads data from the communication buffer of the connection established by hConnection.

#### ⊙ Command Format

Instruction	Name	FB/FC	LD	ST	File
TCP_Connection	Create a TCP connection and connect to the server instruction	FB		TCP_Connection( xEnable:= , xDone=> , xBusy=> , xError=> , hServer:= , eError=> , xActive=> , hConnection=> );	CAA Net Base Services

#### ⊙ Related Variables

##### Input variable

Input variable	Name	Data type	Range	initialization	Descriptive
xEnable	Function Block Enable	BOOL	TRUE-FALSE	-	TRUE: start function block running
hServer	server-side handle	CAA.HANDLE	-	-	TCP server-side handle

##### Output variable

Output variable	Name	Data type	Range	initialization	Descriptive
xDone	Completion	BOOL	TRUE-FALSE	-	Client communication complete
xBusy	Function Block Running	BOOL	TRUE-FALSE	-	Function block running flag
xError	Error Flag	BOOL	TRUE-FALSE	-	Error Status
eError	Error	NBS.ERROR	--	-	Please refer to the CAA Net Base Services library enumeration ERROR for specific errors.
xReady	Successful connection flag	BOOL	TRUE-FALSE	-	Read data from the buffer and set the flag bit for one scan cycle if the data is not empty
szCount	Data Size	CAA.SIZE	-	-	Actual size of the received data area

	Boo	Bit string					Integer							Real number		Moment, Duration, Date, String					
	le	BOO	BY	WO	DWO	LWO	US	UI	UD	UL	SI	IN	DI	LI	RE	LR	TI	DA	TO	DT	STR
xEnable	√	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
hServer		CAA.HANDLE																			
xDone	√	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
xBusy	√	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
xError	√	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
eError		NBS.ERROR																			

xReady	√	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
szCount	CAA.HANDLE																		

**⊙ Program demo**

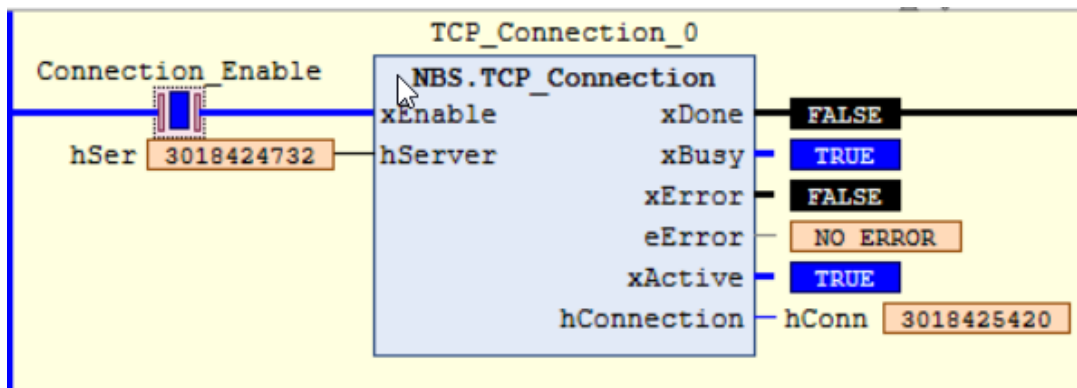
**ST:** Establish a TCP connection and connect to the server. When the TCP\_Connection directive xEnable is TRUE, the local server side will listen to the connection request from the remote client, and if the client and the server are connected successfully, it will create a handle to hConnection, the communication connection between the server and the remote client.

```

1  TCP_Connection_0(
2    xEnable TRUE := Connection_Enable TRUE,
3    xDone=>,
4    xBusy TRUE => Connection_xBusy TRUE,
5    xError=>,
6    hServer 3017878424 :=hSer 3017878424,
7    eError=>,
8    xActive=>,
9    hConnection 3017879200 => hConn 3017879200);
10
11

```

**LD:** Establish a TCP connection and connect to the server. When the TCP\_Connection directive xEnable is TRUE, the local server side will listen to the connection request from the remote client, and if the client and the server are connected successfully, it will create a handle to hConnection, the communication connection between the server and the remote client.


**Note**

If there is an error in the function block, xError will be set to TRUE, eError will be assigned the relevant error value (various values in the enumerated ERROR) , and hConnection will be invalid (equal to 0). See the following error codes for details:.

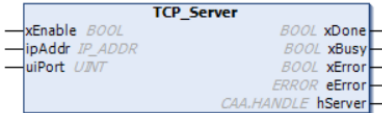
Error Code	Definition	Descripción
0	NO_ERROR	No errors
6000	FIRST_ERROR	Reservations
6001	TIME_OUT	Reserved
6002	INVALID_ADDR	The IP address of the server to which the client is connected is invalid
6003	INVALID_HANDLE	Connection handle is invalid
6004	INVALID_DATAPOINTER	Invalid data pointer
6005	INVALID_DATASIZE	Invalid data size

6006	UDP_RECEIVE_ERROR	UDP receive error
6007	UDP_SEND_ERROR	UDP transmit error
6008	UDP_SEND_NOT_COMPLETE	UDP send incomplete
6009	UDP_OPEN_ERROR	UDP open error
6010	UDP_CLOSE_ERROR	UDP close error
6011	TCP_SEND_ERROR	TCP send error
6012	TCP_RECEIVE_ERROR	TCP send incomplete
6013	TCP_OPEN_ERROR	TCP open error
6014	TCP_CONNECT_ERROR	TCP connection error
6015	TCP_CLOSE_ERROR	TCP close error
6016	TCP_SERVER_ERROR	TCP server error
6017	WRONG_PARAMETER	Parameter error
6018	ERROR_UNKNOWN	Unknown error
6019	TCP_NO_CONNECTION	No TCP connection
6020	LOCTL_ERROR	Internal error (not supported by this machine)
6050	FIRST_MF	Reserved
6099	LAST_ERROR	Reserved

### 5.1.6 TCP\_Server

Reads data from the communication buffer of the connection established by hConnection.

#### ⊙ Command Format

Instruction	Name	FB/FC	LD	ST	File
TCP_Server	Creating a TCP server instruction	FB		TCP_Server( xEnable:= , xDone=> , xBusy=> , xError=> , ipAddr:= , uiPort:= , eError=> , hServer=> )	CAA Net Base Services

#### ⊙ Related Variables

##### Input variable

Input variable	Name	Data type	Range	initialization	Descriptive
xEnable	Function Block Enable	BOOL	TRUE-FALSE	-	TRUE: start function block running
ipAddr	Server IP address	NBS.IP ADDR	-	-	IP address of the server to which the client is connected, refer to structure IP_ADDR in CAA Net Base Services library.
uiPort	Port number	UINT	0-4294967297	-	Port number of the server to which the client is connected

##### Output variable

Output variable	Name	Data type	Range	initialization	Descriptive
xDone	Completion	BOOL	TRUE-FALSE	-	Client communication complete
xBusy	Function Block Running	BOOL	TRUE-FALSE	-	Function block running flag
xError	Error Flag	BOOL	TRUE-FALSE	-	Error Status
eError	Error	NBS.ERROR	--	-	Please refer to the CAA Net Base Services library enumeration ERROR for specific errors.
hServer	server-side handle	CAA.HANDLE	-	-	TCP server-side handle

	Boo	Bit string					Integer							Real number		Moment, Duration, Date, String					
	le	BOO	BYT	WORD	DWORD	LWORD	USINT	UINT	UDINT	ULINT	SINT	INT	DINT	LINT	REAL	LREAL	TIME	DATE	TOD	DT	STRING
xEnable	√	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
ipAddr		NBS.IP ADDR																			
uiPort	-	-	-	-	-	-	√	-	-	-	-	-	-	-	-	-	-	-	-	-	-
xDone	√	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
xBusy	√	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-

xError	√	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
eError	NBS.ERROR																	
hServer	CAA.HANDLE																	

**Program demo**

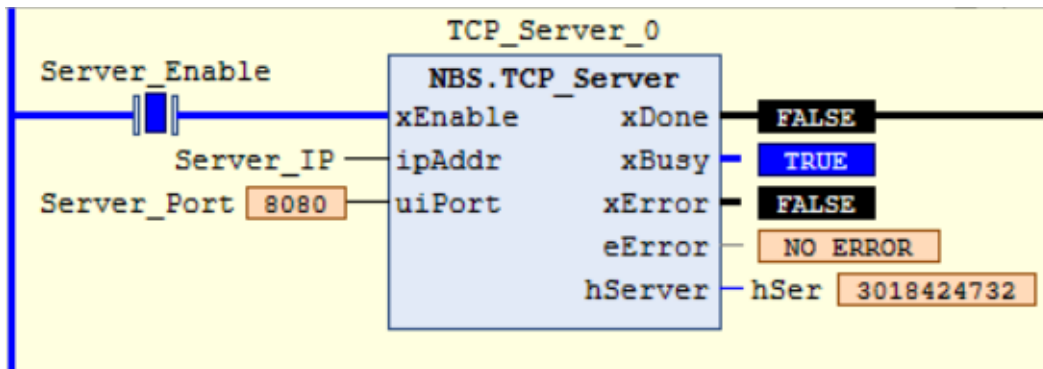
**ST:** Creates a TCP server-side communication service. When the TCP\_Server directive xEnable is TRUE, a valid handle for local server-side TCP communication with remote clients will be created (hServer is greater than 0) and xBusy is TRUE.

```

1  ● Server_IP.sAddr '192.168.1.' := '192.168.1.10';
2  ● Server_Port 8080 := 8080;
3  ● TCP_Server_0(
4      xEnable TRUE := Server_Enable TRUE,
5      xDone=> ,
6      xBusy TRUE => Server_xBusy TRUE ,
7      xError=> ,
8      ipAddr:= Server_IP,
9      uiPort 8080 :=Server_Port 8080 ,
10     eError=> ,
11     hServer 3017878424 => hSer 3017878424 );
12

```

**LD:** Creates a TCP server-side communication service. When the TCP\_Server directive xEnable is TRUE, a valid handle for local server-side TCP communication with remote clients will be created (hServer is greater than 0) and xBusy is TRUE.


**Note**

If there is an error in the function block, xError will be set to TRUE, eError will be assigned the relevant error value (various values in the enumerated ERROR) , and hConnection will be invalid (equal to 0). See the following error codes for details:.

Error Code	Definition	Descripción
0	NO_ERROR	No errors
6000	FIRST_ERROR	Reservations
6001	TIME_OUT	Reserved
6002	INVALID_ADDR	The IP address of the server to which the client is connected is invalid
6003	INVALID_HANDLE	Connection handle is invalid
6004	INVALID_DATAPOINTER	Invalid data pointer
6005	INVALID_DATASIZE	Invalid data size

6006	UDP_RECEIVE_ERROR	UDP receive error
6007	UDP_SEND_ERROR	UDP transmit error
6008	UDP_SEND_NOT_COMPLETE	UDP send incomplete
6009	UDP_OPEN_ERROR	UDP open error
6010	UDP_CLOSE_ERROR	UDP close error
6011	TCP_SEND_ERROR	TCP send error
6012	TCP_RECEIVE_ERROR	TCP send incomplete
6013	TCP_OPEN_ERROR	TCP open error
6014	TCP_CONNECT_ERROR	TCP connection error
6015	TCP_CLOSE_ERROR	TCP close error
6016	TCP_SERVER_ERROR	TCP server error
6017	WRONG_PARAMETER	Parameter error
6018	ERROR_UNKNOWN	Unknown error
6019	TCP_NO_CONNECTION	No TCP connection
6020	LOCTL_ERROR	Internal error (not supported by this machine)
6050	FIRST_MF	Reserved
6099	LAST_ERROR	Reserved

## 5.2 Free Communication UDP

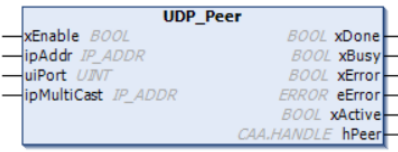
### 5.2.1 Instruction List

Instruction Type	Name	FB/FC	Functionality
Free CommunicationUDP	UDP_Peer	FB	Creating a UDP Communication Connection
	UDP_Receive	FB	UDP communication data reception
	UDP_Send	FB	UDP communication data sending

### 5.2.2 UDP\_Peer

Creating a UDP Communication Connection.

#### ⊙ Command Format

Instruction	Name	FB/FC	LD	ST	File
UDP_Peer	Creating a UDP Communication Connection instruction	FB		<pre> UDP_Peer(   xEnable:=,   xDone=&gt;,   xBusy=&gt;,   xError=&gt;,   ipAddr:=,   uiPort:=,   ipMultiCast:=,   eError=&gt;,   xActive=&gt;,   hPeer=&gt;);           </pre>	CAA Net Base Services

#### ⊙ Related Variables

##### Input variable

Input variable	Name	Data type	Range	initialization	Descriptive
xEnable	Function Block Enable	BOOL	TRUE-FALSE	-	TRUE: start function block running
ipAddr	Server IP address	NBS.IP_ADDR	-	-	IP address of the server to which the client is connected, refer to structure IP_ADDR in CAA Net Base Services library.
uiPort	Port number	UINT	0-4294967297	-	Port number of the server to which the client is connected
ipMultiCast		NBS.IP_ADDR	-	-	

##### Output variable

Output variable	Name	Data type	Range	initialization	Descriptive
xDone	Completion	BOOL	TRUE-FALSE	-	Client communication complete
xBusy	Function Block Running	BOOL	TRUE-FALSE	-	Function block running flag
xError	Error Flag	BOOL	TRUE-FALSE	-	Error Status
eError	Error	NBS.ERROR	--	-	Please refer to the CAA Net Base Services library enumeration ERROR for specific errors.
hServer	server-side handle	CAA.HANDLE	-	-	UDP server-side handle

	Bo le an	Bit string					Integer							Real number		Moment, Duration, Date, String				
	BO OL	BY TE	WO RD	DW ORD	LW ORD	US INT	UI NT	UD INT	UL INT	SI NT	IN T	DI NT	LI NT	RE AL	LR EAL	TI ME	DA TE	TO D	DT	STR ING
xEnable	√	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
ipAddr	NBS.IP_ADDR																			
uiPort	-	-	-	-	-	-	√	-	-	-	-	-	-	-	-	-	-	-	-	-
ipMultiCast	NBS.IP_ADDR																			
xDone	√	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
xBusy	√	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
xError	√	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
eError	NBS.ERROR																			
hServer	CAA.HANDLE																			

### ⊙ Program demo

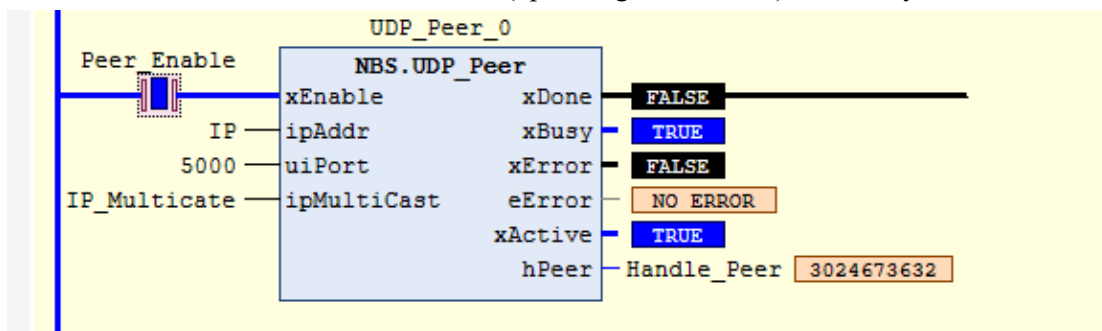
**ST:** Creates a UDP communication connection. When the UDP\_Peer instruction xEnable is TRUE, a valid handle for UDP communication will be created (hpeer greater than 0) and xBusy and xActive are TRUE.

```

1 ● IP.sAddr '192.168.1.' := '192.168.1.10';
2 ● IP_Multicate.sAddr '255.255.255.' := '255.255.255.255';
3 ● IP_Target.sAddr '192.168.1.' := '192.168.1.90';
4 ● UDP_Peer_0 (
5     xEnable TRUE := Peer_Enable TRUE ,
6     xDone => ,
7     xBusy TRUE => Peer_xBusy TRUE ,
8     xError => ,
9     ipAddr := IP ,
10    uiPort 5000 := 5000 ,
11    ipMultiCast := IP_Multicate ,
12    eError => ,
13    xActive TRUE => Peer_xActive TRUE ,
14    hPeer 3024673632 => Handle_Peer 3024673632 );

```

**LD:** Creates a UDP communication connection. When the UDP\_Peer instruction xEnable is TRUE, a valid handle for UDP communication will be created (hpeer greater than 0) and xBusy and xActive are TRUE.



### Note

If an error occurs during communication, xError will be set to TRUE and eError will be assigned the relevant error value (various values in the enumerated ERROR), see the following error codes for details.

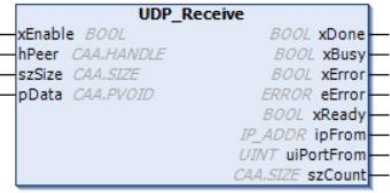
Error Code	Definition	Descripción
0	NO_ERROR	No errors
6000	FIRST_ERROR	Reserved
6001	TIME_OUT	Reserved

6002	INVALID_ADDR	The IP address of the server to which the client is connected is invalid
6003	INVALID_HANDLE	Connection handle is invalid
6004	INVALID_DATAPOINTER	Invalid data pointer
6005	INVALID_DATASIZE	Invalid data size
6006	UDP_RECEIVE_ERROR	UDP receive error
6007	UDP_SEND_ERROR	UDP transmit error
6008	UDP_SEND_NOT_COMPLETE	UDP send incomplete
6009	UDP_OPEN_ERROR	UDP open error
6010	UDP_CLOSE_ERROR	UDP close error
6011	TCP_SEND_ERROR	TCP send error
6012	TCP_RECEIVE_ERROR	TCP send incomplete
6013	TCP_OPEN_ERROR	TCP open error
6014	TCP_CONNECT_ERROR	TCP connection error
6015	TCP_CLOSE_ERROR	TCP close error
6016	TCP_SERVER_ERROR	TCP server error
6017	WRONG_PARAMETER	Parameter error
6018	ERROR_UNKNOWN	Unknown error
6019	TCP_NO_CONNECTION	No TCP connection
6020	LOCTL_ERROR	Internal error (not supported by this machine)
6050	FIRST_MF	Reserved
6099	LAST_ERROR	Reserved

### 5.2.3 UDP\_Receive

Creating a UDP Communication Connection.

#### ⊙ Command Format

Instruction	Name	FB/FC	LD	ST	File
UDP_Receive	UDP communication data reception instruction	FB		<pre> UDP_Receive( xEnable:= , xDone=&gt; , xBusy=&gt; , xError=&gt; , hPeer:= , szSize:= , pData:= , eError=&gt; , xReady=&gt; , ipFrom=&gt; , uiPortFrom=&gt; , szCount=&gt; ); </pre>	CAA Net Base Services

#### ⊙ Related Variables

##### Input variable

Input variable	Name	Data type	Range	initialization	Descriptive
xEnable	Function Block Enable	BOOL	TRUE-FALSE	-	TRUE: start function block running
hPeer	Communication handle	CAA.HANDLE	-	-	UDP communication handle
szSize	Data Size	CAA.SIZE	-	-	Receive data area size, byte
pData	Send Cache	CAA.PVOID	-	-	Pointer to attribute data to receive data from the target device

##### Output variable

Output variable	Name	Data type	Range	initialization	Descriptive
xDone	Completion	BOOL	TRUE-FALSE	-	Client communication complete
xBusy	Function Block Running	BOOL	TRUE-FALSE	-	Function block running flag
xError	Error Flag	BOOL	TRUE-FALSE	-	Error Status
eError	Error	NBS.ERROR	--	-	Please refer to the CAA Net Base Services library enumeration ERROR for specific errors.
xReady	Successful connection	BOOL	TRUE-FALSE	-	Read data from the buffer, if the data is not empty, set the flag bit for one scan cycle
ipFrom	Current packet: Source IP	NBS.IP_ADDR	-	-	Current packet source IP
uiPortFrom	Current packet: source port number	UINT	0-4294967297	-	Current packet source port number
szCount	Data Size	CAA.SIZE	-	-	Actual size of received data area

	Bo ole an	Bit string					Integer							Real number		Moment, Duration, Date, String				
	BOOL	BYTE	WORD	DWORD	LWORD	USINT	UINT	UDINT	ULINT	SINT	INT	DINT	LINT	REAL	LREAL	TIME	DATE	TOD	DT	STRING
xEnable	√	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
hPeer	CAA.HANDLE																			
szSize	CAA.SIZE																			
pData	CAA.PVOID																			
xDone	√	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
xBusy	√	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
xError	√	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
eError	NBS.ERROR																			
xReady	√	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
ipFrom	NBS. IP ADDR																			
uiPortFrom	-	-	-	-	-	-	√	-	-	-	-	-	-	-	-	-	-	-	-	-
szCount	CAA.HANDLE																			

### ⊙ Program demo

**ST:** UDP communication data receive instruction. When the UDP\_Receive instruction xEnable is TRUE, the data will be read from the UDP communication buffer, and xBusy is TRUE. xDone is set for one scan cycle if the read is successful; the read data will be placed into the variable with the address pData; meanwhile, xReady is set for one scan cycle; the value of the actual size of the received data area will be assigned to the value of The actual size of the received data area will be assigned to szCount, and the value of szCount will be cleared to zero after one scan cycle.

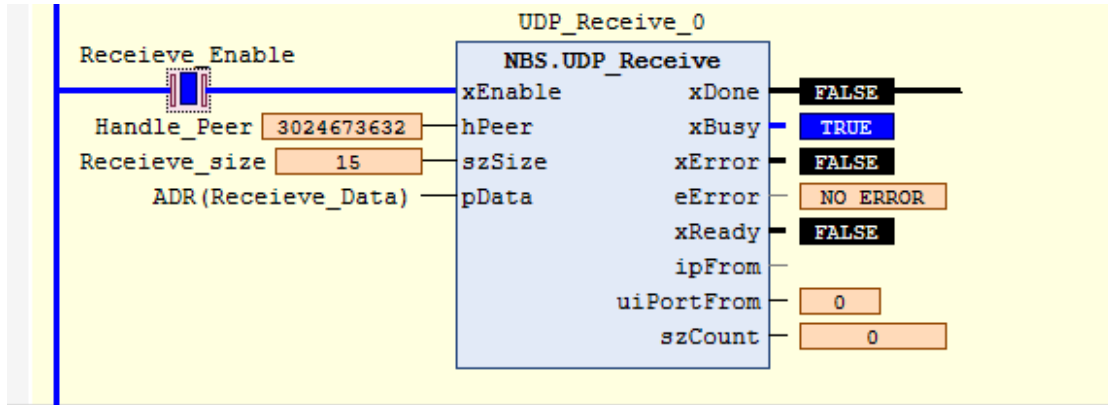
```

16 ● UDP_Receive_0(
17     xEnable TRUE := Receive_Enable TRUE ,
18     xDone FALSE => Receive_xDone FALSE ,
19     xBusy TRUE => Receive_xBusy TRUE ,
20     xError => ,
21     hPeer 3024673632 := Handle_Peer 3024673632 ,
22     szSize 15 := 15 ,
23     pData 3025087476 :=ADR(Receive_Data) ,
24     eError => ,
25     xReady FALSE =>Receive_xReady FALSE ,
26     ipFrom => ,
27     uiPortFrom => ,
28     szCount => );
29

```

**LD:** UDP communication data receive instruction. When the UDP\_Receive instruction xEnable is TRUE, the data will be read from the UDP communication buffer, and xBusy is TRUE. xDone is set for one scan cycle if the read is successful; the read data will be placed into the variable with the address pData; meanwhile, xReady is set for one scan cycle; the value of the actual size of the received data area will be assigned to the value of The actual size of the received data area will be assigned to szCount, and the value of szCount will be cleared to zero after one

scan cycle.



**Note**

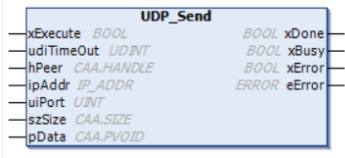
If an error occurs during communication, xError will be set to TRUE and eError will be assigned the relevant error value (various values in the enumerated ERROR), see the following error codes for details.

Error Code	Definition	Descripción
0	NO_ERROR	No errors
6000	FIRST_ERROR	Reserved
6001	TIME_OUT	Reserved
6002	INVALID_ADDR	The IP address of the server to which the client is connected is invalid
6003	INVALID_HANDLE	Connection handle is invalid
6004	INVALID_DATAPOINTER	Invalid data pointer
6005	INVALID_DATASIZE	Invalid data size
6006	UDP_RECEIVE_ERROR	UDP receive error
6007	UDP_SEND_ERROR	UDP transmit error
6008	UDP_SEND_NOT_COMPLETE	UDP send incomplete
6009	UDP_OPEN_ERROR	UDP open error
6010	UDP_CLOSE_ERROR	UDP close error
6011	TCP_SEND_ERROR	TCP send error
6012	TCP_RECEIVE_ERROR	TCP send incomplete
6013	TCP_OPEN_ERROR	TCP open error
6014	TCP_CONNECT_ERROR	TCP connection error
6015	TCP_CLOSE_ERROR	TCP close error
6016	TCP_SERVER_ERROR	TCP server error
6017	WRONG_PARAMETER	Parameter error
6018	ERROR_UNKNOWN	Unknown error
6019	TCP_NO_CONNECTION	No TCP connection
6020	LOCTL_ERROR	Internal error (not supported by this machine)
6050	FIRST_MF	Reserved
6099	LAST_ERROR	Reserved

### 5.2.4 UDP\_Send

Creating a UDP Communication Connection.

#### ⊙ Command Format

Instruction	Name	FB/FC	LD	ST	File
UDP_Send	UDP communication data sending instruction	FB		<pre> UDP_Send( xExecute:= , udiTimeOut:= , xDone=&gt; , xBusy=&gt; , xError=&gt; , hPeer:= , ipAddr:= , uiPort:= , szSize:= , pData:= , eError=&gt; ); </pre>	CAA Net Base Services

#### ⊙ Related Variables

##### Input variable

Input variable	Name	Data type	Range	initialization	Descriptive
xEnable	Function Block Enable	BOOL	TRUE-FALSE	-	TRUE: start function block running
xExecute	Function Block Enable	BOOL	TRUE-FALSE	-	TRUE (rising edge signal), then start function block operation
hPeer	Communication handle	CAA.HANDLE	-	-	UDP communication handle
ipAddr	Target IP	NBS.IP_ADDR	-	-	Destination IP for sending data
uiPort	communications port number	UINT	0-4294967297	-	Communications port number
szSize	Data Size	CAA.SIZE	-	-	Receive data area size, byte
pData	Send Cache	CAA.PVOID	-	-	Pointer to attribute data to receive data from the target device

##### Output variable

Output variable	Name	Data type	Range	initialization	Descriptive
xDone	Completion	BOOL	TRUE-FALSE	-	Client communication complete
xBusy	Function Block Running	BOOL	TRUE-FALSE	-	Function block running flag
xError	Error Flag	BOOL	TRUE-FALSE	-	Error Status
eError	Error	NBS.ERROR	--	-	Please refer to the CAA Net Base Services library enumeration ERROR for specific errors.

	Bo le an	Bit string					Integer							Real number		Moment, Duration, Date, String				
	BOOL	BYTE	WORD	DWORD	LWORD	USINT	UINT	UDINT	ULINT	SINT	INT	DINT	LINT	REAL	LREAL	TIME	DATE	TOD	DT	STRING
xEnable	√	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
udiTimeOut	-	-	-	-	-	-	-	√	-	-	-	-	-	-	-	-	-	-	-	-
hPeer	CAA.HANDLE																			
ipAddr	CAA.SIZE																			
uiPort	-	-	-	-	-	-	√	-	-	-	-	-	-	-	-	-	-	-	-	-
szSize	CAA.SIZE																			
pData	CAA.PVOID																			
xDone	√	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
xBusy	√	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
xError	√	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
eError	NBS. IP_ADDR																			

**⊙ Program demo**

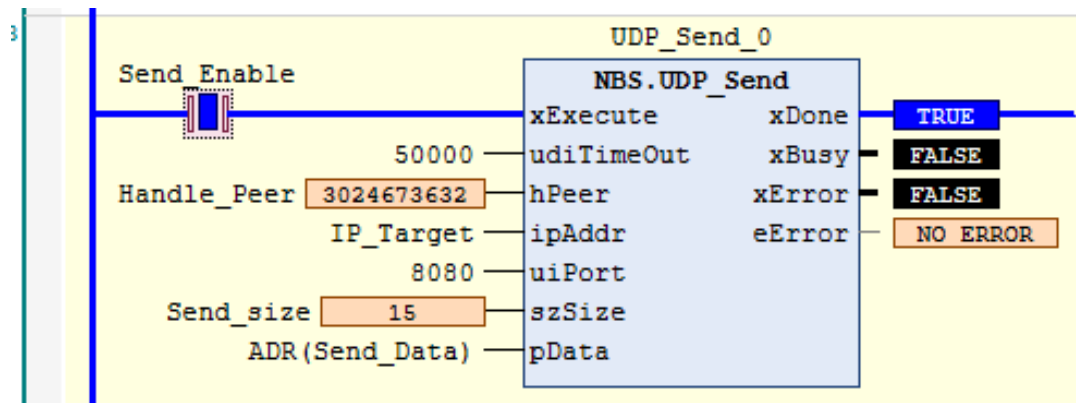
**ST:** UDP communication data send instruction. When the UDP\_Send instruction xExecute is TRUE, it sends the data with the length of szSize set by the user with the send buffer pData as the first address to the target device. If the send is successful within the timeout period, xDone is set to TRUE.

```

30  ● UDP_Send_0(
31      xExecute TRUE := Send_Enable TRUE ,
32      udiTimeOut 50000 :=50000 ,
33      xDone TRUE => Send_xDone TRUE ,
34      xBusy FALSE => Send_xBusy FALSE ,
35      xError=> ,
36      hPeer 3024673632 :=Handle_Peer 3024673632 ,
37      ipAddr:= IP_Target,
38      uiPort 8080 := 8080,
39      szSize 15 :=15 ,
40      pData 3025087491 := ADR(Send_Data),
41      eError=> );
42

```

**LD:** UDP communication data send instruction. When the UDP\_Send instruction xExecute is TRUE, it sends the data with the length of szSize set by the user with the send buffer pData as the first address to the target device. If the send is successful within the timeout period, xDone is set to TRUE.



**Note**

If an error occurs during communication, xError will be set to TRUE and eError will be assigned the relevant error value (various values in the enumerated ERROR), see the following error codes for details.

<b>Error Code</b>	<b>Definition</b>	<b>Descripción</b>
0	NO_ERROR	No errors
6000	FIRST_ERROR	Reserved
6001	TIME_OUT	Reserved
6002	INVALID_ADDR	The IP address of the server to which the client is connected is invalid
6003	INVALID_HANDLE	Connection handle is invalid
6004	INVALID_DATAPOINTER	Invalid data pointer
6005	INVALID_DATASIZE	Invalid data size
6006	UDP_RECEIVE_ERROR	UDP receive error
6007	UDP_SEND_ERROR	UDP transmit error
6008	UDP_SEND_NOT_COMPLETE	UDP send incomplete
6009	UDP_OPEN_ERROR	UDP open error
6010	UDP_CLOSE_ERROR	UDP close error
6011	TCP_SEND_ERROR	TCP send error
6012	TCP_RECEIVE_ERROR	TCP send incomplete
6013	TCP_OPEN_ERROR	TCP open error
6014	TCP_CONNECT_ERROR	TCP connection error
6015	TCP_CLOSE_ERROR	TCP close error
6016	TCP_SERVER_ERROR	TCP server error
6017	WRONG_PARAMETER	Parameter error
6018	ERROR_UNKNOWN	Unknown error
6019	TCP_NO_CONNECTION	No TCP connection
6020	LOCTL_ERROR	Internal error (not supported by this machine)
6050	FIRST_MF	Reserved
6099	LAST_ERROR	Reserved

## 5.3 EtherCAT communication

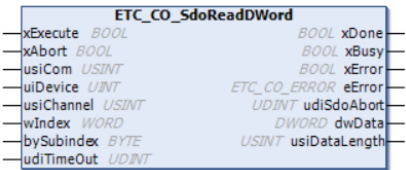
### 5.3.1 Instruction List

Instruction Type	Name	FB/FC	Functionality
EtherCAT communication	ETC_CO_SdoReadDWord	FB	EtherCAT Slave SDO Read
	ETC_CO_SdoRead4	FB	EtherCAT Slave SDO Read
	ETC_CO_SdoRead	FB	EtherCAT Slave SDO Read
	ETC_CO_SdoWrite_Dword	FB	EtherCAT Slave SDO Write
	ETC_CO_SdoWrite4	FB	EtherCAT Slave SDO Write
	ETC_CO_SdoWrite	FB	EtherCAT Slave SDO Write
	IoDrvEtherCAT_Diag	FB	EtherCAT Master Example
	ETCSlave	FB	EtherCAT Master Example

### 5.3.2 ETC\_CO\_SdoReadDWord

EtherCAT slave SDO data read, and read object dictionary value length not greater than 4 bytes.

#### ⊙ Command Format

Instruction	Name	FB/FC	LD	ST	File
ETC_CO_SdoReadDWord	EtherCAT Slave SDO Read instruction	FB		ETC_CO_SdoReadDWord( xExecute:= , xAbort:= , usiCom:= , uiDevice:= , usiChannel:= , wIndex:= , bySubindex:= , udiTimeout:= , xDone=> , xBusy=> , xError=> , eError=> , udiSdoAbort=> , dwData=> , usiDataLength=> );	IODrvEtherCAT

#### ⊙ Related Variables

##### Input variable

Input variable	Name	Data type	Range	initialization	Descriptive
xEnable	Function Block Enable	BOOL	TRUE-FALSE	-	TRUE: start function block running
xAbort	Function block abort	BOOL	TRUE-FALSE	FALSE	If TRUE, the current function block read command is terminated!
usiCom	Number of EtherCAT masters	USINT	1-2	1	If only one EtherCAT master is used, usiCom is 1. If more than one master is used, the first master is 1, the second master is 2, and so on.
uiDevice	Slave station number	UINT	1-65535	0	Slave station number e.g. 1001,1002 etc.
usiChannel	Reserved	USINT	1	1	Reservations
wIndex	Object dictionary	WORD	1-65535	0	e.g. servo control word object dictionary 16#6040

	master index				
bySubindex	Object dictionary sub-index	BYTE	0-255	0	e.g. servo control word object dictionary subindex 16#00
udiTimeOut	Timeout	UDINT	1-65535	0	Timeout for reading slave parameters in ms

**Output variable**

Output variable	Name	Data type	Range	initialization	Descriptive
xDone	Completion	BOOL	TRUE-FALSE	-	Client communication complete
xBusy	Function Block Running	BOOL	TRUE-FALSE	-	Function block running flag
xError	Error Flag	BOOL	TRUE-FALSE	-	Error Status
eError	Error	NBS.ERROR	--	-	Please refer to the CAA Net Base Services library enumeration ERROR for specific errors.
udiSdoAbort	Abort	UDINT	0-4294967295	0	More error information can be read from this output variable when a device error occurs
dwData	Data value read	DWORD	0-4294967295	0	SDO data value read from slave.
usiDataLength	Length of data read	USINT	0-255	0	Length of the SDO data value read from the slave.

	Bo ole an	Bit string				Integer							Real number		Moment, Duration, Date, String					
	BOOL	BYTE	WORD	DWORD	LWORD	USINT	UINT	UDINT	ULINT	SINT	INT	DINT	LINT	REAL	LREAL	TIME	DATE	TOD	DT	STRING
xExecute	√	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
xAbort	√	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
usiCom	-	-	-	-	-	√	-	-	-	-	-	-	-	-	-	-	-	-	-	-
uiDevice	-	-	-	-	-	-	√	-	-	-	-	-	-	-	-	-	-	-	-	-
usiChannel	-	-	-	-	-	√	-	-	-	-	-	-	-	-	-	-	-	-	-	-
wIndex	-	-	√	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
bySubindex	-	√	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
udiTimeOut	-	-	-	-	-	-	-	√	-	-	-	-	-	-	-	-	-	-	-	-
xDone	√	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
xBusy	√	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
xError	√	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
eError	ETC CO ERROR																			
udiSdoAbort	-	-	-	-	-	-	-	√	-	-	-	-	-	-	-	-	-	-	-	-
dwData	-	-	-	√	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
usiDataLength	-	-	-	-	-	√	-	-	-	-	-	-	-	-	-	-	-	-	-	-

**⊙ Program demo**

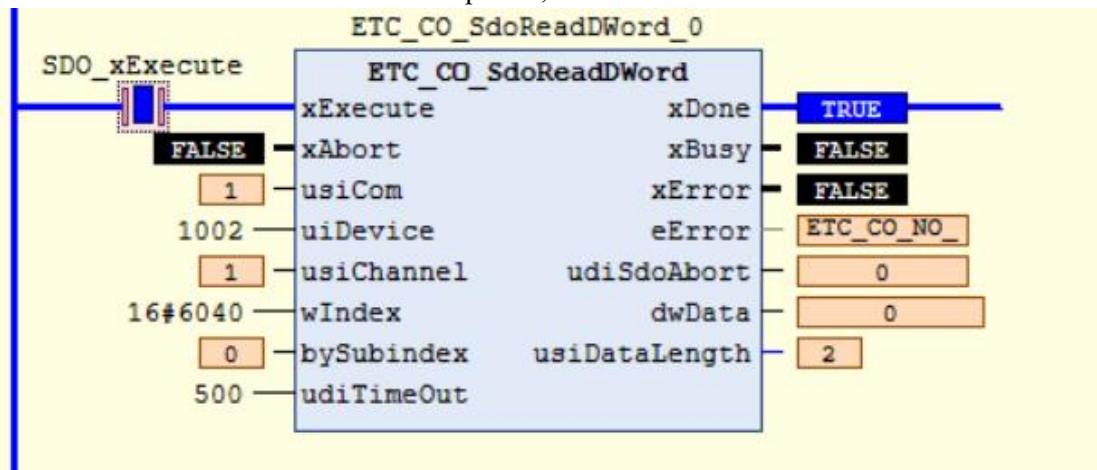
**ST:** Write servo slave control word mode object dictionary index 16#6040, subindex 16#00.

```

1  ● ETC_CO_SdoReadDWord_0(
2     xExecute TRUE := SDO_xExecute TRUE ,
3     xAbort := ,
4     usiCom := ,
5     uiDevice 1002 := 1002 ,
6     usiChannel := ,
7     wIndex 24640 := 16#6040,
8     bySubindex := ,
9     udiTimeOut 500 := 500,
10    xDone => ,
11    xBusy => ,
12    xError => ,
13    eError => ,
14    udiSdoAbort => ,
15    dwData 0 => SDO_Data 0 ,
16    usiDataLength 2 => SDO_Length 2 );
17

```

**LD:** UDP communication data send instruction. When the UDP\_Send instruction xExecute is TRUE, it sends the data with the length of szSize set by the user with the send buffer pData as the first address to the target device. If the send is successful within the timeout period, xDone is set to TRUE.


**Note**

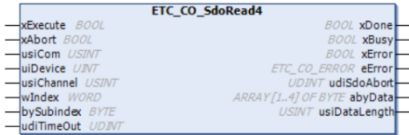
If an error occurs during communication, xError will be set to TRUE and eError will be assigned the relevant error value (various values in the enumerated ERROR), see the following error codes for details.

Name	Data Type	Description
ETC_CO_NO_ERROR	WORD	-
ETC_CO_FIRST_ERROR	WORD	-
ETC_CO_OTHER_ERROR	WORD	This error is reserved for all SDO channels in busy! Prohibit changes
ETC_CO_DATA_OVERFLOW	WORD	-
ETC_CO_TIMEOUT	WORD	-
ETC_CO_FIRST_MF	WORD	-
ETC_CO_LAST_ERROR	WORD	-

### 5.3.3 ETC\_CO\_SdoRead4

The EtherCAT slave SDO data is read and the length of the read object dictionary value is not greater than 4 bytes. The difference with the ETC\_CO\_SdoReadDWord function block is that ETC\_CO\_SdoRead4 stores the read values in a byte type array.

#### ⊙ Command Format

Instruction	Name	FB/FC	LD	ST	File
ETC_CO_SdoRead4	EtherCAT Slave SDO Read instruction	FB		ETC_CO_SdoRead4( xExecute:= , xAbort:= , usiCom:= , uiDevice:= , usiChannel:= , wIndex:= , bySubindex:= , udiTimeOut:= , xDone=> , xBusy=> , xError=> , eError=> , udiSdoAbort=> , abyData=> , usiDataLength=> );	IODrvEtherCAT

#### ⊙ Related Variables

##### Input variable

Input variable	Name	Data type	Range	initialization	Descriptive
xEnable	Function Block Enable	BOOL	TRUE-FALSE	-	TRUE: start function block running
xAbort	Function block abort	BOOL	TRUE-FALSE	FALSE	If TRUE, the current function block read command is terminated!
usiCom	Number of EtherCAT masters	USINT	1-2	1	If only one EtherCAT master is used, usiCom is 1. If more than one master is used, the first master is 1, the second master is 2, and so on.
uiDevice	Slave station number	UINT	1-65535	0	Slave station number e.g. 1001,1002 etc.
usiChannel	Reserved	USINT	1	1	Reservations
wIndex	Object dictionary master index	WORD	1-65535	0	e.g. servo control word object dictionary 16#6040
bySubindex	Object dictionary sub-index	BYTE	0-255	0	e.g. servo control word object dictionary subindex 16#00
udiTimeOut	Timeout	UDINT	1-65535	0	Timeout for reading slave parameters in ms

##### Output variable

Output variable	Name	Data type	Range	initialization	Descriptive
xDone	Completion	BOOL	TRUE-FALSE	-	Client communication complete
xBusy	Function Block	BOOL	TRUE-FALSE	-	Function block running flag

	Running				
xError	Error Flag	BOOL	TRUE-FALSE	-	Error Status
eError	Error	NBS.ERROR	--	-	Please refer to the CAA Net Base Services library enumeration ERROR for specific errors.
udiSdoAbort	Abort	UDINT	0-4294967295	0	More error information can be read from this output variable when a device error occurs
abyData	The value of the data read	ARRAY [1..4] OF BYTE	-	0	SDO data values read from the slave, values filled in order from array index 1 to 4
usiDataLength	Length of data read	USINT	0-255	0	Length of the SDO data value read from the slave.

	Bo	Bit string					Integer							Real number		Moment, Duration, Date, String					
	olean	BOOL	BYTE	WORD	DWORD	LWORD	USINT	UINT	UDINT	ULINT	SINT	INT	DINT	LINT	REAL	LREAL	TIME	DATE	TOD	DT	STRING
xExecute	√	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
xAbort	√	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
usiCom	-	-	-	-	-	√	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
uiDevice	-	-	-	-	-	-	√	-	-	-	-	-	-	-	-	-	-	-	-	-	-
usiChannel	-	-	-	-	-	√	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
wIndex	-	-	√	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
bySubindex	-	√	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
udiTimeOut	-	-	-	-	-	-	-	√	-	-	-	-	-	-	-	-	-	-	-	-	-
xDone	√	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
xBusy	√	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
xError	√	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
eError		ETC CO ERROR																			
udiSdoAbort	-	-	-	-	-	-	-	√	-	-	-	-	-	-	-	-	-	-	-	-	-
abyData		ARRAY [1..4] OF BYTE																			
usiDataLength	-	-	-	-	-	√	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-

**⊙ Program demo**

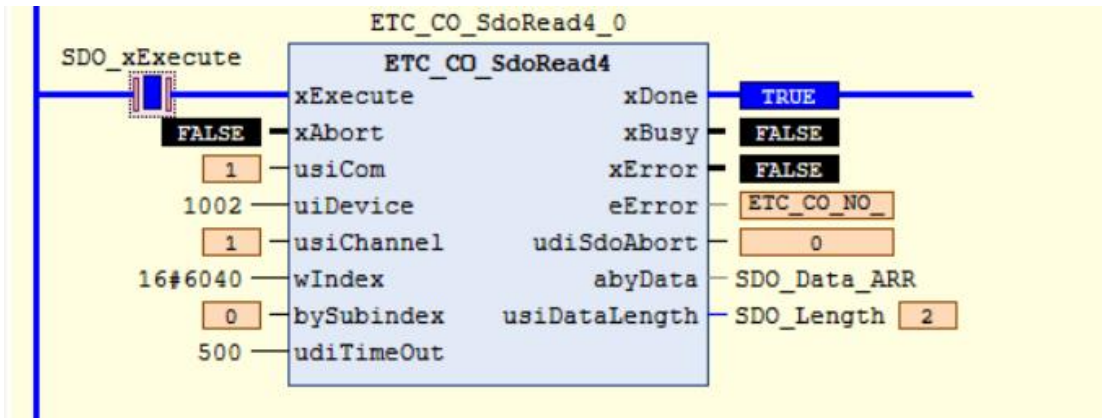
**ST:** Write servo slave control word mode object dictionary index 16#6040, subindex 16#00. (Note: values read are stored in a byte type array with an array size of 4 bytes).

```

1  ETC_CO_SdoRead4_0 (
2    xExecute TRUE := SDO_xExecute TRUE ,
3    xAbort:= ,
4    usiCom:= ,
5    uiDevice 1002 :=1002 ,
6    usiChannel:= ,
7    wIndex 24640 := 16#6040,
8    bySubindex:= ,
9    udiTimeOut 500 :=500 ,
10   xDone=> ,
11   xBusy=> ,
12   xError=> ,
13   eError=> ,
14   udiSdoAbort=> ,
15   abyData=> SDO_Data_ARR,
16   usiDataLength 2 => SDO_Length 2 );
17

```

**LD:** Write servo slave control word mode object dictionary index 16#6040, subindex 16#00. (Note: values read are stored in a byte type array with an array size of 4 bytes)


**Note**

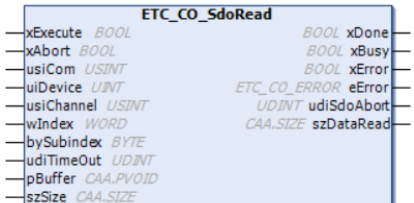
If an error occurs during communication, xError will be set to TRUE and eError will be assigned the relevant error value (various values in the enumerated ERROR), see the following error codes for details.

Name	Data Type	Description
ETC_CO_NO_ERROR	WORD	-
ETC_CO_FIRST_ERROR	WORD	-
ETC_CO_OTHER_ERROR	WORD	This error is reserved for all SDO channels in busy! Prohibit changes
ETC_CO_DATA_OVERFLOW	WORD	-
ETC_CO_TIMEOUT	WORD	-
ETC_CO_FIRST_MF	WORD	-
ETC_CO_LAST_ERROR	WORD	-

### 5.3.4 ETC\_CO\_SdoRead

EtherCAT slave SD0 data reading, unlike the ETC\_CO\_SdoReadDWord or ETC\_CO\_SdoRead4 function blocks, the ETC\_CO\_SdoRead function block reads object dictionary values that can be greater than 4 bytes in length.

#### ⊙ Command Format

Instruction	Name	FB/FC	LD	ST	File
ETC_CO_SdoRead	EtherCAT Slave SDO Read instruction	FB		<pre>ETC_CO_SdoRead(   xExecute:= ,   xAbort:= ,   usiCom:= ,   uiDevice:= ,   usiChannel:= ,   wIndex:= ,   bySubindex:= ,   udiTimeOut:= ,   pBuffer:= ,   szSize:= ,   xDone=&gt; ,   xBusy=&gt; ,   xError=&gt; ,   eError=&gt; ,   udiSdoAbort=&gt; ,   szDataRead=&gt; );</pre>	IODrvEtherCAT

#### ⊙ Related Variables

##### Input variable

Input variable	Name	Data type	Range	initialization	Descriptive
xEnable	Function Block Enable	BOOL	TRUE-FALSE	-	TRUE: start function block running
xAbort	Function block abort	BOOL	TRUE-FALSE	FALSE	If TRUE, the current function block read command is terminated!
usiCom	Number of EtherCAT masters	USINT	1-2	1	If only one EtherCAT master is used, usiCom is 1. If more than one master is used, the first master is 1, the second master is 2, and so on.
uiDevice	Slave station number	UINT	1-65535	0	Slave station number e.g. 1001,1002 etc.
usiChannel	Reserved	USINT	1	1	Reservations
wIndex	Object dictionary master index	WORD	1-65535	0	e.g. servo control word object dictionary 16#6040
bySubindex	Object dictionary sub-index	BYTE	0-255	0	e.g. servo control word object dictionary subindex 16#00
udiTimeOut	Timeout	UDINT	1-65535	0	Timeout for reading slave parameters in ms
pBuffer	Pointer to data buffer	CAA.PVOID	—	0	Storage area after successful data read
szSize	Data buffer size	CAA.SIZE	-	0	Size of the data buffer (pBuffer)

**Output variable**

Output variable	Name	Data type	Range	initialization	Descriptive
xDone	Completion	BOOL	TRUE-FALSE	-	Client communication complete
xBusy	Function Block Running	BOOL	TRUE-FALSE	-	Function block running flag
xError	Error Flag	BOOL	TRUE-FALSE	-	Error Status
eError	Error	NBS.ERROR	--	-	Please refer to the CAA Net Base Services library enumeration ERROR for specific errors.
udiSdoAbort	Abort	UDINT	0-4294967295	0	More error information can be read from this output variable when a device error occurs
szDataRead	Size of data successfully read (bytes)	CAA.SIZE		0	Number of bytes normally read

	Bo ole an	Bit string				Integer							Real number		Moment, Duration, Date, String					
	BOOL	BYTE	WORD	DWORD	LWORD	USINT	UINT	UDINT	ULINT	SINT	INT	DINT	LINT	REAL	LREAL	TIME	DATE	TOD	DT	STRING
xExecute	√	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
xAbort	√	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
usiCom	-	-	-	-	-	√	-	-	-	-	-	-	-	-	-	-	-	-	-	-
uiDevice	-	-	-	-	-	-	√	-	-	-	-	-	-	-	-	-	-	-	-	-
usiChannel	-	-	-	-	-	√	-	-	-	-	-	-	-	-	-	-	-	-	-	-
wIndex	-	-	√	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
bySubindex	-	√	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
udiTimeOut	-	-	-	-	-	-	-	√	-	-	-	-	-	-	-	-	-	-	-	-
pBuffer	CAA.PVOID																			
szSize	CAA.SIZE																			
xDone	√	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
xBusy	√	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
xError	√	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
eError	ETC CO ERROR																			
udiSdoAbort	-	-	-	-	-	-	-	√	-	-	-	-	-	-	-	-	-	-	-	-
szDataRead	CAA.SIZE																			

**⊙ Program demo**

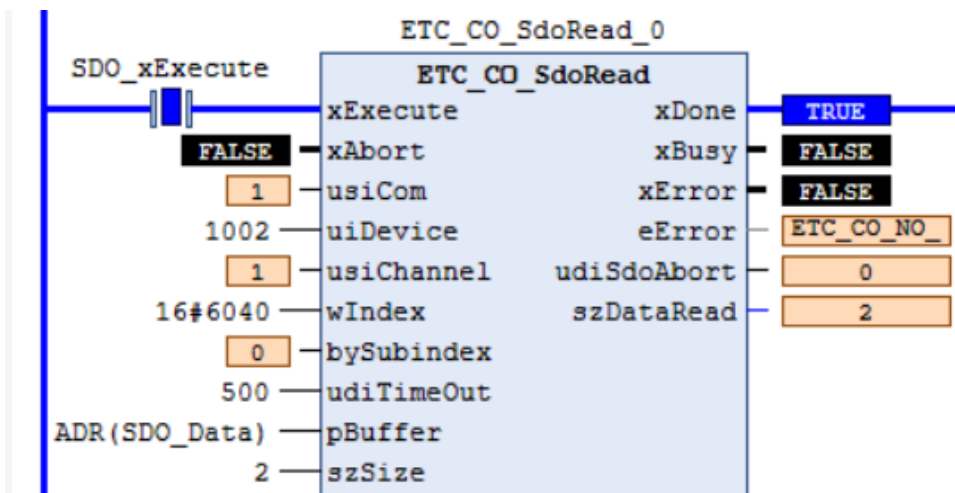
**ST:** Write servo slave control word mode object dictionary index 16#6040, subindex 16#00.

```

1  ● ETC_CO_SdoRead_0(
2      xExecute TRUE := SDO_xExecute TRUE,
3      xAbort:= ,
4      usiCom:= ,
5      uiDevice 1002 :=1002 ,
6      usiChannel:= ,
7      wIndex 24640 := 16#6040,
8      bySubindex:= ,
9      udiTimeOut 500 :=500 ,
10     pBuffer 3019616116 := ADR(SDO_Data 0) ,
11     szSize 2 :=2 ,
12     xDone=> ,
13     xBusy=> ,
14     xError=> ,
15     eError=> ,
16     udiSdoAbort=> ,
17     szDataRead 2 =>SDO_Length 2 );

```

**LD:** Write servo slave control word mode object dictionary index 16#6040, subindex 16#00.


**Note**

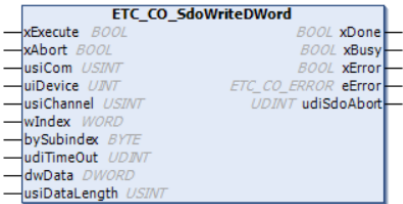
If an error occurs during communication, xError will be set to TRUE and eError will be assigned the relevant error value (various values in the enumerated ERROR), see the following error codes for details.

Name	Data Type	Description
ETC_CO_NO_ERROR	WORD	-
ETC_CO_FIRST_ERROR	WORD	-
ETC_CO_OTHER_ERROR	WORD	This error is reserved for all SDO channels in busy! Prohibit changes
ETC_CO_DATA_OVERFLOW	WORD	-
ETC_CO_TIMEOUT	WORD	-
ETC_CO_FIRST_MF	WORD	-
ETC_CO_LAST_ERROR	WORD	-

### 5.3.5 ETC\_CO\_SdoWrite\_Dword

EtherCAT slave SDO data is written and the length of the written object dictionary value is not greater than 4 bytes.

#### ⊙ Command Format

Instruction	Name	FB/FC	LD	ST	File
ETC_CO_SdoWrite_Dword	EtherCAT Slave SDO Write instruction	FB		<pre>ETC_CO_SdoWriteDWord(   xExecute:= ,   xAbort:= ,   usiCom:= ,   uiDevice:= ,   usiChannel:= ,   wIndex:= ,   bySubindex:= ,   udiTimeOut:= ,   dwData:= ,   usiDataLength:= ,   xDone=&gt; ,   xBusy=&gt; ,   xError=&gt; ,   eError=&gt; ,   udiSdoAbort=&gt; );</pre>	IODrvEtherCAT

#### ⊙ Related Variables

##### Input variable

Input variable	Name	Data type	Range	initialization	Descriptive
xEnable	Function Block Enable	BOOL	TRUE-FALSE	-	TRUE: start function block running
xAbort	Function block abort	BOOL	TRUE-FALSE	FALSE	If TRUE, the current function block read command is terminated!
usiCom	Number of EtherCAT masters	USINT	1-2	1	If only one EtherCAT master is used, usiCom is 1. If more than one master is used, the first master is 1, the second master is 2, and so on.
uiDevice	Slave station number	UINT	1-65535	0	Slave station number e.g. 1001,1002 etc.
usiChannel	Reserved	USINT	1	1	Reservations
wIndex	Object dictionary master index	WORD	1-65535	0	e.g. servo control word object dictionary 16#6040
bySubindex	Object dictionary sub-index	BYTE	0-255	0	e.g. servo control word object dictionary subindex 16#00
udiTimeOut	Timeout	UDINT	1-65535	0	Timeout for reading slave parameters in ms
dwData	Value of data written	DWORD	0-4294967295	0	Write slave SDO data value
usiDataLength	Length of data read	USINT	0-255	0	Length of the data value written to the slave SDO

**Output variable**

Output variable	Name	Data type	Range	initialization	Descriptive
xDone	Completion	BOOL	TRUE-FALSE	-	Client communication complete
xBusy	Function Block Running	BOOL	TRUE-FALSE	-	Function block running flag
xError	Error Flag	BOOL	TRUE-FALSE	-	Error Status
eError	Error	NBS.ERROR	--	-	Please refer to the CAA Net Base Services library enumeration ERROR for specific errors.
udiSdoAbort	Abort	UDINT	0-4294967295	0	More error information can be read from this output variable when a device error occurs

	Bo ole an	Bit string				Integer								Real number		Moment, Duration, Date, String					
	BO OL	BY TE	WO RD	DWO RD	LWO RD	US INT	UI NT	UD INT	UL INT	SI NT	IN T	DI NT	LI NT	RE AL	LR EAL	TI ME	DA TE	TO D	DT	STR ING	
xExecute	√	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
xAbort	√	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
usiCom	-	-	-	-	-	√	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
uiDevice	-	-	-	-	-	-	√	-	-	-	-	-	-	-	-	-	-	-	-	-	-
usiChannel	-	-	-	-	-	√	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
wIndex	-	-	√	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
bySubindex	-	√	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
udiTimeOut	-	-	-	-	-	-	-	√	-	-	-	-	-	-	-	-	-	-	-	-	-
dwData	-	-	-	√	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
usiDataLength	-	-	-	-	-	√	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
xDone	√	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
xBusy	√	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
xError	√	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
eError	ETC CO ERROR																				
udiSdoAbort	-	-	-	-	-	-	-	√	-	-	-	-	-	-	-	-	-	-	-	-	-

**⊙ Program demo**

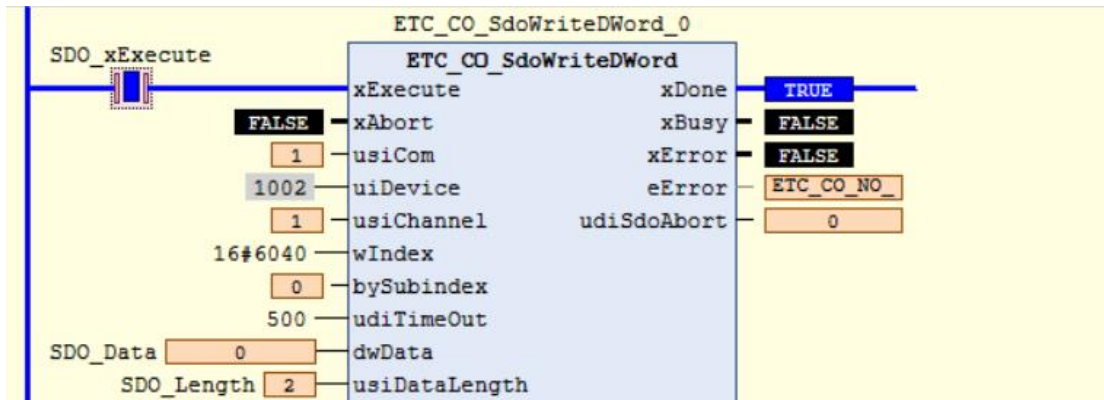
**ST:** Write servo slave control word mode object dictionary index 16#6040, subindex 16#00. (Note: values written are stored in a byte type array with an array size of 4 bytes).

```

1  ● ETC_CO_SdoWriteDWord_0(
2      xExecute TRUE := SDO_xExecute TRUE ,
3      xAbort:= ,
4      usiCom:= ,
5      uiDevice 1002 := 1002,
6      usiChannel:= ,
7      wIndex 24640 := 16#6040,
8      bySubindex:= ,
9      udiTimeOut 500 := 500,
10     dwData 0 := SDO_Data 0,
11     usiDataLength 2 := SDO_Length 2,
12     xDone=> ,
13     xBusy=> ,
14     xError=> ,
15     eError=> ,
16     udiSdoAbort=> );
17

```

**LD:** Write servo slave control word mode object dictionary index 16#6040, subindex 16#00. (Note: values written are stored in a byte type array with an array size of 4 bytes).


**Note**

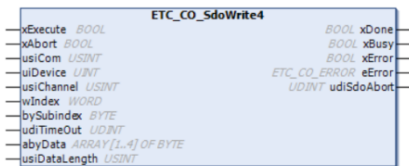
If an error occurs during communication, xError will be set to TRUE and eError will be assigned the relevant error value (various values in the enumerated ERROR), see the following error codes for details.

Name	Data Type	Description
ETC_CO_NO_ERROR	WORD	-
ETC_CO_FIRST_ERROR	WORD	-
ETC_CO_OTHER_ERROR	WORD	This error is reserved for all SDO channels in busy! Prohibit changes
ETC_CO_DATA_OVERFLOW	WORD	-
ETC_CO_TIMEOUT	WORD	-
ETC_CO_FIRST_MF	WORD	-
ETC_CO_LAST_ERROR	WORD	-

### 5.3.6 ETC\_CO\_SdoWrite4

EtherCAT slave SDO data is written and the length of the written object dictionary value is not greater than 4 bytes. The difference with the ETC\_CO\_SdoWriteDWord function block is that ETC\_CO\_SdoWrite4 writes the values as a byte type array.

#### ⊙ Command Format

Instruction	Name	FB/FC	LD	ST	File
ETC_CO_SdoWrite4	EtherCAT Slave SDO Write instruction	FB		<pre>ETC_CO_SdoWrite4(   xExecute:= ,   xAbort:= ,   usiCom:= ,   uiDevice:= ,   usiChannel:= ,   wIndex:= ,   bySubindex:= ,   udiTimeOut:= ,   abyData:= ,   usiDataLength:= ,   xDone=&gt; ,   xBusy=&gt; ,   xError=&gt; ,   eError=&gt; ,   udiSdoAbort=&gt; );</pre>	IODrvEtherCAT

#### ⊙ Related Variables

##### Input variable

Input variable	Name	Data type	Range	initialization	Descriptive
xEnable	Function Block Enable	BOOL	TRUE-FALSE	-	TRUE: start function block running
xAbort	Function block abort	BOOL	TRUE-FALSE	FALSE	If TRUE, the current function block read command is terminated!
usiCom	Number of EtherCAT masters	USINT	1-2	1	If only one EtherCAT master is used, usiCom is 1. If more than one master is used, the first master is 1, the second master is 2, and so on.
uiDevice	Slave station number	UINT	1-65535	0	Slave station number e.g. 1001,1002 etc.
usiChannel	Reserved	USINT	1	1	Reservations
wIndex	Object dictionary master index	WORD	1-65535	0	e.g. servo control word object dictionary 16#6040
bySubindex	Object dictionary sub-index	BYTE	0-255	0	e.g. servo control word object dictionary subindex 16#00
udiTimeOut	Timeout	UDINT	1-65535	0	Timeout for reading slave parameters in ms
abyData	Data value read	ARRAY [1..4] OF BYTE	-	0	SDO data values read from the slave, values filled in order from array index 1 to 4
usiDataLength	Length of data read	USINT	0-255	0	Length of the data value written to the slave SDO

**Output variable**

Output variable	Name	Data type	Range	initialization	Descriptive
xDone	Completion	BOOL	TRUE-FALSE	-	Client communication complete
xBusy	Function Block Running	BOOL	TRUE-FALSE	-	Function block running flag
xError	Error Flag	BOOL	TRUE-FALSE	-	Error Status
eError	Error	NBS.ERROR	--	-	Please refer to the CAA Net Base Services library enumeration ERROR for specific errors.
udiSdoAbort	Abort	UDINT	0-4294967295	0	More error information can be read from this output variable when a device error occurs

	Bo le an	Bit string				Integer								Real number		Moment, Duration, Date, String				
	BOOL	BYTE	WORD	DWORD	LWORD	USINT	UINT	UDINT	ULINT	SINT	INT	DINT	LINT	REAL	LREAL	TIME	DATE	TOD	DT	STRING
xExecute	√	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
xAbort	√	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
usiCom	-	-	-	-	-	√	-	-	-	-	-	-	-	-	-	-	-	-	-	-
uiDevice	-	-	-	-	-	-	√	-	-	-	-	-	-	-	-	-	-	-	-	-
usiChannel	-	-	-	-	-	√	-	-	-	-	-	-	-	-	-	-	-	-	-	-
wIndex	-	-	√	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
bySubindex	-	√	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
udiTimeOut	-	-	-	-	-	-	-	√	-	-	-	-	-	-	-	-	-	-	-	-
abyData	ARRAY [1..4]OF BYTE																			
usiDataLength	-	-	-	-	-	√	-	-	-	-	-	-	-	-	-	-	-	-	-	-
xDone	√	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
xBusy	√	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
xError	√	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
eError	ETC CO ERROR																			
udiSdoAbort	-	-	-	-	-	-	-	√	-	-	-	-	-	-	-	-	-	-	-	-

**⊙ Program demo**

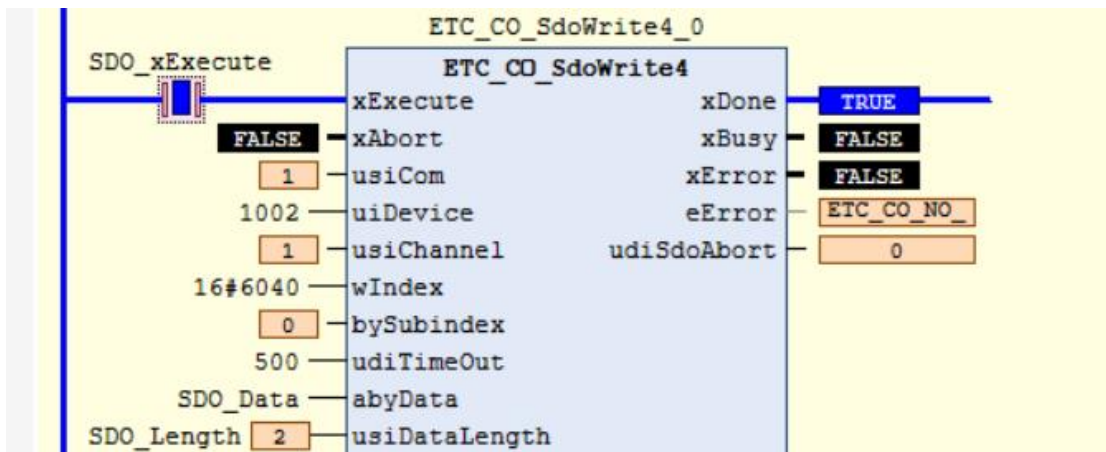
**ST:** Write servo slave control word mode object dictionary index 16#6040, subindex 16#00. (Note: values written are stored in a byte type array with an array size of 4 bytes)

```

1  ● ETC_CO_SdoWrite4_0(
2      xExecute TRUE := SDO_xExecute TRUE,
3      xAbort:= ,
4      usiCom:= ,
5      uiDevice 1002 := 1002,
6      usiChannel:= ,
7      wIndex 24640 := 16#6040,
8      bySubindex:= ,
9      udiTimeOut 500 :=500 ,
10     abyData:=SDO_Data ,
11     usiDataLength 2 :=SDO_Length 2 ,
12     xDone=> ,
13     xBusy=> ,
14     xError=> ,
15     eError=> ,
16     udiSdoAbort=> );
17

```

**LD:** Write servo slave control word mode object dictionary index 16#6040, subindex 16#00. (Note: values written are stored in a byte type array with an array size of 4 bytes)


**Note**

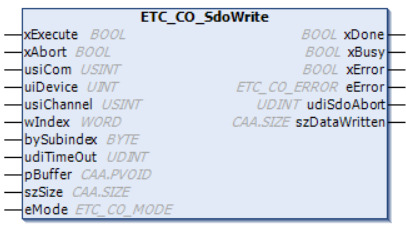
If an error occurs during communication, xError will be set to TRUE and eError will be assigned the relevant error value (various values in the enumerated ERROR), see the following error codes for details.

Name	Data Type	Description
ETC_CO_NO_ERROR	WORD	-
ETC_CO_FIRST_ERROR	WORD	-
ETC_CO_OTHER_ERROR	WORD	This error is reserved for all SDO channels in busy! Prohibit changes
ETC_CO_DATA_OVERFLOW	WORD	-
ETC_CO_TIMEOUT	WORD	-
ETC_CO_FIRST_MF	WORD	-
ETC_CO_LAST_ERROR	WORD	-

### 5.3.7 ETC\_CO\_SdoWrite

EtherCAT slave SD0 data is written and the length of the written object dictionary value is not greater than 4 bytes. The difference with the ETC\_CO\_SdoWriteDWord function block is that ETC\_CO\_SdoWrite4 writes the values as a byte type array.

#### ⊙ Command Format

Instruction	Name	FB/FC	LD	ST	File
ETC_CO_SdoWrite	EtherCAT Slave SDO Write instruction	FB		<pre>ETC_CO_SdoWrite( xExecute:= , xAbort:= , usiCom:= , uiDevice:= , usiChannel:= , wIndex:= , bySubindex:= , udiTimeOut:= , pBuffer:= , szSize:= , eMode:= , xDone=&gt; , xBusy=&gt; , xError=&gt; , eError=&gt; , udiSdoAbort=&gt; , szDataWritten=&gt; );</pre>	IODrvEtherCAT

#### ⊙ Related Variables

##### Input variable

Input variable	Name	Data type	Range	initialization	Descriptive
xEnable	Function Block Enable	BOOL	TRUE-FALSE	-	TRUE: start function block running
xAbort	Function block abort	BOOL	TRUE-FALSE	FALSE	If TRUE, the current function block read command is terminated!
usiCom	Number of EtherCAT masters	USINT	1-2	1	If only one EtherCAT master is used, usiCom is 1. If more than one master is used, the first master is 1, the second master is 2, and so on.
uiDevice	Slave station number	UINT	1-65535	0	Slave station number e.g. 1001,1002 etc.
usiChannel	Reserved	USINT	1	1	Reservations
wIndex	Object dictionary master index	WORD	1-65535	0	e.g. servo control word object dictionary 16#6040
bySubindex	Object dictionary sub-index	BYTE	0-255	0	e.g. servo control word object dictionary subindex 16#00
udiTimeOut	Timeout	UDINT	1-65535	0	Timeout for reading slave parameters in ms
pBuffer	Pointer to data buffer	CAA.PVOID	—	—	Write data pointer start address
szSize	Data buffer size	CAA.SIZE	-	—	Write data length
eMode	Write mode selection	ETC_CO_MODE	-	—	Write mode selection

**Output variable**

Output variable	Name	Data type	Range	initialization	Descriptive
xDone	Completion	BOOL	TRUE-FALSE	-	Client communication complete
xBusy	Function Block Running	BOOL	TRUE-FALSE	-	Function block running flag
xError	Error Flag	BOOL	TRUE-FALSE	-	Error Status
eError	Error	NBS.ERROR	--	-	Please refer to the CAA Net Base Services library enumeration ERROR for specific errors.
udiSdoAbort	Abort	UDINT	0-4294967295	0	More error information can be read from this output variable when a device error occurs
szDataWritten	Completion of write data length	CAA.SIZE	-	-	Completion of write data length

	Bo ole an	Bit string				Integer								Real number		Moment, Duration, Date, String				
	BOOL	BYTE	WORD	DWORD	LWORD	USINT	UINT	UDINT	ULINT	SINT	INT	DINT	LINT	REAL	LREAL	TIME	DATE	TOD	DT	STRING
xExecute	√	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
xAbort	√	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
usiCom	-	-	-	-	-	√	-	-	-	-	-	-	-	-	-	-	-	-	-	-
uiDevice	-	-	-	-	-	-	√	-	-	-	-	-	-	-	-	-	-	-	-	-
usiChannel	-	-	-	-	-	√	-	-	-	-	-	-	-	-	-	-	-	-	-	-
wIndex	-	-	√	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
bySubindex	-	√	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
udiTimeOut	-	-	-	-	-	-	-	√	-	-	-	-	-	-	-	-	-	-	-	-
pBuffer	CAA.PVOID																			
szSize	CAA.SIZE																			
eMode	ETC CO MODE																			
xDone	√	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
xBusy	√	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
xError	√	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
eError	ETC CO ERROR																			
udiSdoAbort	-	-	-	-	-	-	-	√	-	-	-	-	-	-	-	-	-	-	-	-
szDatWritten	CAA.SIZE																			

**⊙ Program demo**

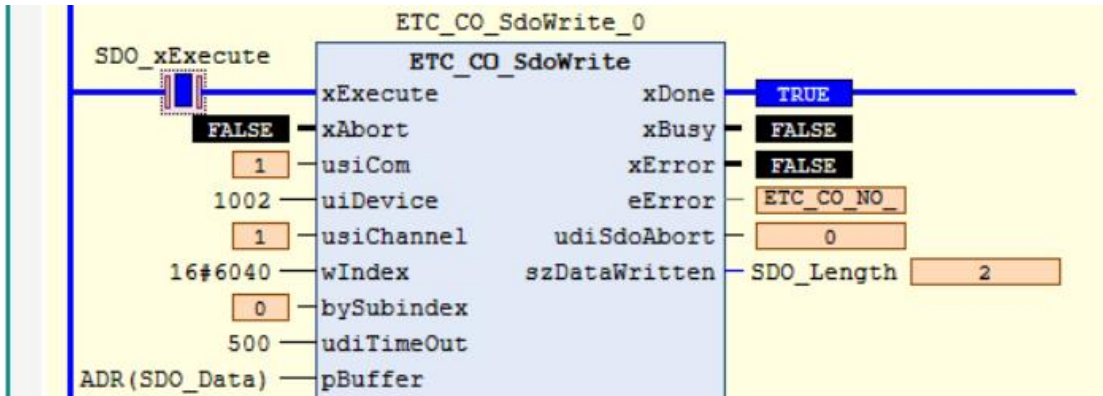
**ST:** Write servo slave control word mode object dictionary index 16#6040, subindex 16#00.

```

1
2 ● ETC_CO_SdoWrite_0(
3   xExecute TRUE := SDO_xExecute TRUE,
4   xAbort:= ,
5   usiCom:= ,
6   uiDevice 1002 := 1002,
7   usiChannel:= ,
8   wIndex 24640 := 16#6040 ,
9   bySubindex:= ,
10  udiTimeOut 500 := 500,
11  pBuffer 3019816116 := ADR(SDO_Data 16#00000000) ,
12  szSize 2 := 2,
13  eMode:= ,
14  xDone=> ,
15  xBusy=> ,
16  xError=> ,
17  eError=> ,
18  udiSdoAbort=> ,
19  szDataWritten 2 => SDO_Length 2 );
20

```

**LD:** Write servo slave control word mode object dictionary index 16#6040, subindex 16#00.


**Note**

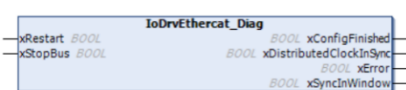
If an error occurs during communication, xError will be set to TRUE and eError will be assigned the relevant error value (various values in the enumerated ERROR), see the following error codes for details.

Name	Data Type	Description
ETC_CO_NO_ERROR	WORD	-
ETC_CO_FIRST_ERROR	WORD	-
ETC_CO_OTHER_ERROR	WORD	This error is reserved for all SDO channels in busy! Prohibit changes
ETC_CO_DATA_OVERFLOW	WORD	-
ETC_CO_TIMEOUT	WORD	-
ETC_CO_FIRST_MF	WORD	-
ETC_CO_LAST_ERROR	WORD	-

### 5.3.8 IoDrvEtherCAT\_Diag

EtherCAT slave SD0 data is written and the length of the written object dictionary value is not greater than 4 bytes. The difference with the ETC\_CO\_SdoWriteDWord function block is that ETC\_CO\_SdoWrite4 writes the values as a byte type array.

#### ☉ Command Format

Instruction	Name	FB/FC	LD	ST	File
IoDrvEtherCAT_Diag	EtherCAT Master Example instruction	FB		IoDrvEtherCAT_Diag( xRestart:= , xStopBus:= , xConfigFinished=> , xDistributedClockInSync => , xError=> , xSyncInWindow=> );	IODrvEtherCAT

#### ☉ Related Variables

##### Input variable

Input variable	Name	Data type	Range	initialization	Descriptive
xRestart	EtherCAT master restart	BOOL	TRUE-FALSE	FALSE	TRUE: (rising edge signal) EtherCAT master reboot
xStopBus	EtherCAT master stop	BOOL	TRUE-FALSE	FALSE	Stop EtherCAT master

##### Output variable

Output variable	Name	Data type	Range	initialization	Descriptive
xConfigFinished	EtherCAT master configuration complete	BOOL	TRUE-FALSE	FALSE	If EtherCAT configuration is complete and communication is running, the output is TRUE.
xDistributedClockInSync	DC synchronisation signal	BOOL	TRUE-FALSE	FALSE	TRUE if the synchronisation mode is completed in DC mode.
xError	Error flags	BOOL	TRUE-FALSE	-	Error status
xSyncInWindow	Synchronisation window	BOOL	TRUE-FALSE	FALSE	If the synchronisation is within the synchronisation window, the output is TRUE.

	Boo le an	Bit string				Integer							Real number		Moment, Duration, Date, String						
	BOO L	BY TE	WO RD	DWO RD	LWO RD	US INT	UI NT	UD INT	UL INT	SI NT	IN T	DI NT	LI NT	RE AL	LR EAL	TI ME	DA TE	TO D	DT	STR ING	
xRestart	√	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
xStopBus	√	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
xConfigFinished	√	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
xDistributedClockInSync	√	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
xError	√	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
xSyncnWindow	√	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-

**⊙ Program demo**

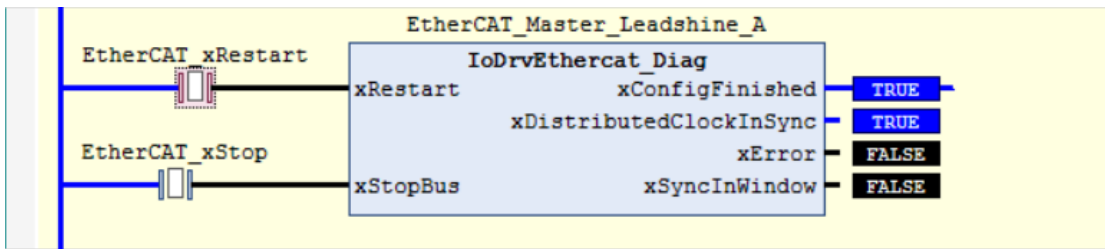
**ST:** If the device has only one master, the instantiation name is EtherCAT\_Master\_Leadshine\_A. If the EtherCAT configuration is complete, the communication is running and the synchronisation mode is completed in DC mode, the values xConfigFinished and xDistributedClockInSync are output as TRUE.

```

1 ● EtherCAT_Master_Leadshine_A(
2     xRestartFALSE := EtherCAT_xRestartFALSE,
3     xStopBusFALSE := EtherCAT_xStopFALSE,
4     xConfigFinishedTRUE => EtherCAT_xConfigFinishedTRUE,
5     xDistributedClockInSyncTRUE => EtherCAT_xDistributedClockInSyncTRUE,
6     xError=>,
7     xSyncInWindow=> );
8

```

**LD:** If the device has only one master, the instantiation name is EtherCAT\_Master\_Leadshine\_A. If the EtherCAT configuration is complete, the communication is running and the synchronisation mode is completed in DC mode, the values xConfigFinished and xDistributedClockInSync are output as TRUE.



**Note**

The name of the instantiation needs to be the same as the name of the master bus so that the master bus can be restarted.

### 5.3.9 ETCSlave

EtherCAT slave SD0 data is written and the length of the written object dictionary value is not greater than 4 bytes. The difference with the ETC\_CO\_SdoWriteDWord function block is that ETC\_CO\_SdoWrite4 writes the values as a byte type array.

#### ⊙ Command Format

Instruction	Name	FB/FC	LD	ST	File
ETCSlave	EtherCAT Master Example instruction	FB		ETCSlave(xSetOperational:= , wState=> );	IODrvEtherCAT

#### ⊙ Related Variables

##### Input variable

Input variable	Name	Data type	Range	initialization	Descriptive
xSetOperational	Try to set the slave to the ETC_SLAVE_OPERATIONAL state	BOOL	TRUE-FALSE	FALSE	If the input is TRUE (rising edge signal) then an attempt is made to set the slave's state machine to the ETC_SLAVE_OPERATIONAL state

##### Output variable

Output variable	Name	Data type	Range	initialization	Descriptive
wState	Slave status	ETC_SLAVE_STATE	-	-	Read slave status

	Boo	Bit string					Integer					Real number		Moment, Duration, Date, String							
	lean	BOOL	BYTE	WORD	DWORD	LWORD	USINT	UINT	UDINT	ULINT	SINT	INT	DINT	LINT	REAL	LREAL	TIME	DATE	TOD	DT	STRING
xSetOperational	√	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
wState		ETC SLAVE STATE																			

#### ⊙ Program demo

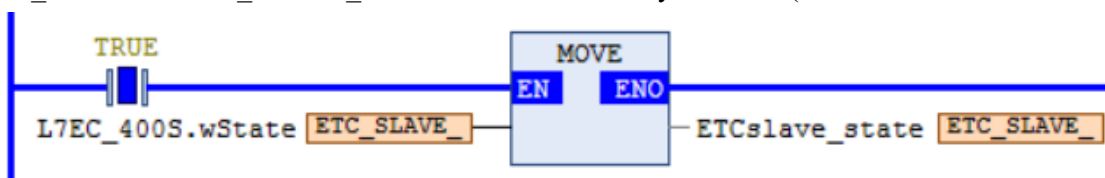
**ST:** Take L7EC\_400S as an example to add the instance name of the relevant servo slave, and define ETCSlave\_state as the ETC\_SLAVE\_STATE enumeration body variable. (See Notes for detailed state)

```

1 L7EC_400S(xSetOperational:= , wState 0 =>ETCSlave_state 0 );
2

```

**LD:** Take L7EC\_400S as an example to add the instance name of the relevant servo slave, and define ETCSlave\_state as the ETC\_SLAVE\_STATE enumeration body variable. (See Notes for detailed state)



**Note**

wState is the enumerator ETC\_SLAVE\_STATE variable, which reads the slave state, and the following table shows the information of the ETC\_SLAVE\_STATE enumerator variable.

<b>Name</b>	<b>Data Type</b>	<b>Description</b>
ETC_SLAVE_BOOT	WORD	Takes the value 0
ETC_SLAVE_Init	WORD	Takes the value 1
ETC_SLAVE_PREOPERATIONAL	WORD	Takes the value 2
ETC_SLAVE_SAVEOPERATIONAL	WORD	Takes the value 4
ETC_SLAVE_OPERATIONAL	WORD	Takes the value 8

## 5.4 Ethernet-IP communication

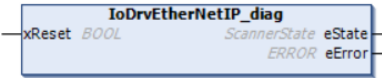
### 5.4.1 Instruction List

Instruction Type	Name	FB/FC	Functionality
Ethernet-IP communication	IoDrvEtherNetIP_Diag	FB	Ethernet/IP Master Example
	RemoteAdapter	FB	Ethernet/IP Slave Example
	Generic_Service	FB	CIP Universal Display Service
	Get_Attributes_All	FB	Get all properties of an object instance
	Get_Attribute_Single	FB	Getting a single property of an object instance
	Set_Attributes_All	FB	Setting all properties of an object instance
	Set_Attribute_Single	FB	Setting a single property of an object instance

### 5.4.2 IoDrvEtherNetIP\_Diag

Ethernet/IP Master Example

#### ⊙ Command Format

Instruction	Name	FB/FC	LD	ST	File
IoDrvEtherNetIP_Diag	Ethernet/IP Master Example instruction	FB		IoDrvEtherNetIP(xReset:= , eState=> , eError=> );	IoDrvEtherNetIP

#### ⊙ Related Variables

##### Input variable

Input variable	Name	Data type	Range	initialization	Descriptive
xReset	Rest	BOOL	TRUE-FALSE	FALSE	TRUE: (Rising edge signal) reset master

##### Output variable

Output variable	Name	Data type	Range	initialization	Descriptive
eState	Slave status	ETC_SLAVE STATE	-	-	Status of EtherNet/IP when doing scanners
eError	Error	ERROR	-	-	Please refer to the IoDrvEtherNetIP library enumeration ERROR for specific errors.

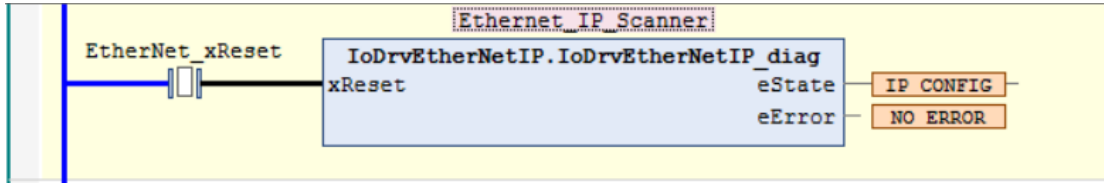
	Boo le an	Bit string				Integer							Real number		Moment, Duration, Date, String					
	BOOL	BYTE	WORD	DWORD	LWORD	USINT	UINT	UDINT	ULINT	SINT	INT	DINT	LINT	REAL	LREAL	TIME	DATE	TOD	DT	STRING
xReset	√	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
eState		ScannerState																		
eError		ERROR																		

**Ⓞ Program demo**

**ST:** The instantiated name is Ethernet\_IP\_Scanner, and eState is the current state when you are a scanner.

```
1 | Ethernet_IP_Scanner(xReset:=FALSE, EtherNet_xReset:=FALSE, eState:=IP_CONFIG => EtherNet_state:=IP_CONFIG, eError=> );
```

**LD:** The instantiated name is Ethernet\_IP\_Scanner, and eState is the current state when you are a scanner.


**Note**

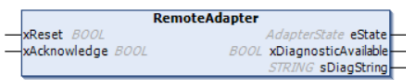
If an error occurs in the use of the function block, eError will be assigned the relevant error value (the various values in the enumerated ERROR). See the following error codes for details.

Error Code	Definition	Descripción
0	NO_ERROR	-
16#1	INVALID_COMMAND	-
16#2	OUT_OF_MEMORY	-
16#3	INVALID_DATA	-
16#64	INVALID_SESSION_HANDLE	-
16#65	INVALID_LENGTH	-
16#69	UNSUPPORTED_PROTOCOL_VERSION	-
16#AA	NBS_ERROR	-
	NBS_RCV_ERROR	-
	NBS_SND_ERROR	-
	ENCAPSULATION_ERROR	-
	TCPIP_CONFIG_ERROR	-
	UDP_CONFIG_ERROR	-
	UDP_RECV_ERROR	-
	UDP_SEND_ERROR	-
	UDP_CLOSE_ERROR	-
	NULL_POINTER	-
	DEVICE_STATE_ERROR	-
	RECONFIGURATION_FAILED	-
	PERFORMANCE_MONITOR_DISABLE D	-
	INVALID_MEASURING_POINT	-
	IP_CONFIG_ERROR	-

### 5.4.3 RemoteAdapter

Ethernet/IP Slave Example

#### ⊙ Command Format

Instruction	Name	FB/FC	LD	ST	File
RemoteAdapter	Ethernet/IP Slave Example instruction	FB		RemoteAdapter( xReset:= , xAcknowledge:= , eState=> , xDiagnosticAvailable=> , sDiagString=> );	IoDrvEtherNetIP

#### ⊙ Related Variables

##### Input variable

Input variable	Name	Data type	Range	initialization	Descriptive
xReset	Rest	BOOL	TRUE-FALSE	FALSE	TRUE: (Rising edge signal) reset master
xAcknowledge	Confirmation of diagnostic information	BOOL	TRUE-FALSE	FALSE	TRUE: (Rising edge signal) Acknowledgement of diagnostic information

##### Output variable

Output variable	Name	Data type	Range	initialization	Descriptive
eState	Slave status	ETC_SLAVE STATE	-	-	Status of EtherNet/IP when doing scanners
xDiagnosticAvailable	Diagnostics available	BOOL	TRUE-FALSE	FALSE	
sDiagString	Diagnostic Information	STRING	-	-	Diagnostic information string, displays diagnostic information about the adapter

	Boo le an	Bit string					Integer							Real number		Moment, Duration, Date, String				
	BOOL	BYTE	WORD	DWORD	LWORD	USINT	UINT	UDINT	ULINT	SINT	INT	DINT	LINT	REAL	LREAL	TIME	DATE	TOD	DT	STRING
xReset	√	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
xAcknowledge	√	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
eState		AdapterState																		
xDiagnosticAvailable	√	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
sDiagString	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	√

**⊙ Program demo**

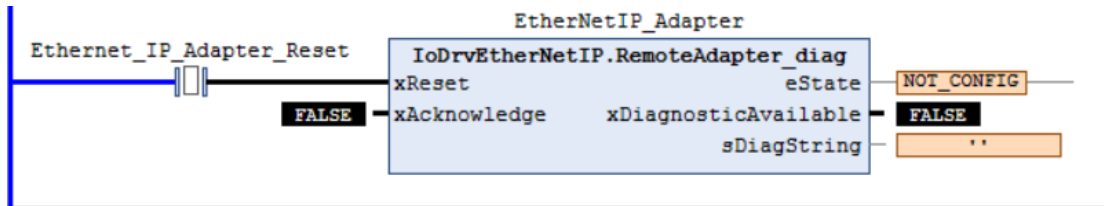
**ST:** The instantiated name is EtherNetIP\_Adapter, and eState is the current state of EtherNet/IP when it is an adapter.

```

● EtherNetIP_Adapter (
    xReset FALSE := Ethernet_IP_Adapter_Reset FALSE ,
    xAcknowledge := ,
    eState NOT_CONFIG => Ethernet_IP_Adapter_State NOT_CONFIG ,
    xDiagnosticAvailable => ,
    sDiagString => ); RETURN

```

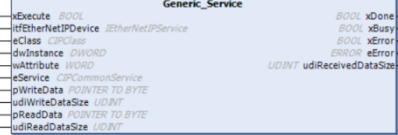
**LD:** The instantiated name is EtherNetIP\_Adapter, and eState is the current state of EtherNet/IP when it is an adapter.



### 5.4.4 Generic\_Service

Accessing the CIP Common Display Services function block of the EtherNetIP adapter.

#### ⊙ Command Format

Instruction	Name	FB/FC	LD	ST	File
Generic_Service	CIP Universal Display Service instruction	FB		<pre>Generic_Service( xExecute:= , xDone=&gt; , xBusy=&gt; , xError=&gt; , itfEtherNetIPDevice:= , eClass:= , dwInstance:= , eError=&gt; , wAttribute:= , eService:= , pWriteData:= , udiWriteDataSize:= , pReadData:= , udiReadDataSize:= , udiReceivedDataSize=&gt; );</pre>	EtherNetIPService

#### ⊙ Related Variables

##### Input variable

Input variable	Name	Data type	Range	initialization	Descriptive
xExecute	Function Block Enable	BOOL	TRUE-FALSE	-	TRUE: (rising edge signal), start function block operation
eClass	Class	ENIP. CIP Class	-	-	The class of objects to which the service refers, either CIP standard objects or third party vendors. Custom objects
itfEtherNetIPDevice	Remote Slave Instantiation Name	IEtherNetIPService	-	-	Remote slave instantiation name
dwInstance	Instance	DWORD	-	-	The actual number of instances when CIP is executed, which should be greater than or equal to 1.
wAttribute	Attribute	WORD	-	-	Attributes pointed to by the service
eService	Public Services	ENIP. CIPCommonService	-	-	CIP Common Public Service
pWriteData	Write Cache	POINTER TO BYTE	-	-	Pointer to attribute data to be written to the target adapter
udiWriteDataSize	Write data length	UDINT	1~65535	-	Write Data Length
pReadData	Receive Cache	POINTER TO BYTE	-	-	Pointer to attribute data, data received from adapter
udiReadDataSize	Receive Length	UDINT	1~65535	-	Length of data received from the adapter

**Output variable**

Output variable	Name	Data type	Range	initialization	Descriptive
xDone	Completion	BOOL	TRUE-FALSE	-	Client communication complete
xBusy	Function Block Running	BOOL	TRUE-FALSE	-	Function block running flag
xError	Error Flag	BOOL	TRUE-FALSE	-	Error Status
eError	Error	NBS.ERROR	--	-	Please refer to the CAA Net Base Services library enumeration ERROR for specific errors.
udiReceivedDataSize	Reception length	UDINT	-	-	Reception length

	Bo le an	Bit string					Integer							Real number		Moment, Duration, Date, String				
	BOOL	BYTE	WORD	DWORD	LWORD	USINT	UINT	UDINT	ULINT	SINT	INT	DINT	LINT	REAL	LREAL	TIME	DATE	TOD	DT	STRING
xExecute	√	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
eClass	ENIP. CIP Class																			
itfEtherNetIPDevice	IEtherNetIPService																			
dwInstance	-	-	√	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
wAttribute	-	√	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
eService	ENIP. CIPCommonService																			
pWriteData	POINTER TO BYTE																			
udiWriteDataSize	-	-	-	-	-	-	√	-	-	-	-	-	-	-	-	-	-	-	-	-
pReadData	POINTER TO BYTE																			
udiReadDataSize	-	-	-	-	-	-	√	-	-	-	-	-	-	-	-	-	-	-	-	-
xDone	√	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
xBusy	√	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
xError	√	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
eError	ENIP. ERROR																			
udiReceivedDataSize	-	-	-	-	-	-	√	-	-	-	-	-	-	-	-	-	-	-	-	-

**⊙ Program demo**

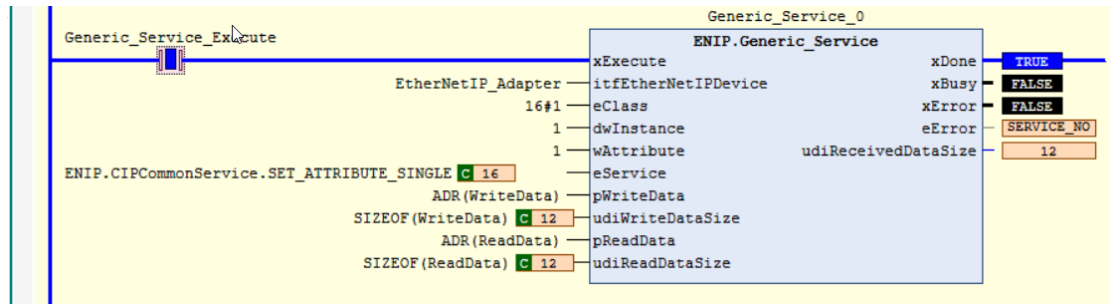
**ST:** Attribute and service data in the user program can be obtained through the function block Generic\_Service.

```

● Generic_Service_0(
  xExecute TRUE := Generic_Service_Execute TRUE,
  xDone TRUE => Generic_Service_Done TRUE,
  xBusy=> ,
  xError=> ,
  itfEtherNetIPDevice:=EtherNetIP_Adapter ,
  eClass IdentityOb :=16#1 ,
  dwInstance 1 := 1,
  eError=> ,
  wAttribute 1 :=1 ,
  eService SET_ATTRIB := ENIP.CIPCommonService.SET_ATTRIBUTE_SINGLE,
  pWriteData 16#B43AEE34 := ADR(WriteData),
  udiWriteDataSize 12 := SIZEOF(WriteData),
  pReadData 16#B43AEE40 :=ADR(ReadData) ,
  udiReadDataSize 12 :=SIZEOF(ReadData) ,
  udiReceivedDataSize=> );RETURN

```

**LD:** Attribute and service data in the user program can be obtained through the function block `Generic_Service`.



**Note**

itfEtherNetIPDevice is the instantiated name of the slave and eService is the CIP Public Service ID and name (see table below for details). If an error occurs in the usage function block, eError will be assigned the relevant error value (various values in the enumerated ERROR). See the following error codes for details.

**CIP Common Services and Attributes**

ID	Causality
16#1	IdentityObject
16#2	MessageRouterObject
16#3	DeviceNetObject
16#4	AssemblyObject
16#5	ConnectionObject
16#6	ConnectionManagerObject
16#7	RegisterObject
16#8	DiscreteInputPointObject
16#9	DiscreteOutputPointObject
16#A	AnalogInputPointObject
16#B	AnalogOutputPointObject
16#E	PresenceSensingObject
16#F	ParameterObject
16#10	ParameterGroupObject
16#12	GroupObject
16#1D	DiscreteInputGroupObject
16#1E	DiscreteOutputGroupObject
16#1F	DiscreteGroupObject
16#20	AnalogInputGroupObject
16#21	AnalogOutputGroupObject
16#22	AnalogGroupObject
16#23	PositionSensorObject
16#24	PositionControllerSupervisorObject
16#25	PositionControllerObject
16#26	BlockSequencerObject
16#27	CommandBlockObject
16#28	MotorDataObject
16#29	ControlSupervisorObject

16#2A	ACDCDriveObject
16#2B	AcknowledgeHandlerObject
16#2C	OverloadObject
16#2D	SoftstartObject
16#2E	SelectionObject
16#30	S_DeviceSupervisorObject
16#31	S_AnalogSensorObject
16#32	S_AnalogActuatorObject
16#33	S_SingleStageControllerObject
16#34	S_GasCalibrationObject
16#35	TripPointObject
16#37	FileObject
16#38	S_PartialPressureObject
16#39	SafetySupervisorObject
16#3A	SafetyValidatorObject
16#3B	SafetyDiscreteOutputPointObject
16#3C	SafetyDiscreteOutputGroupObject
16#3D	SafetyDiscreteInputPointObject
16#3E	SafetyDiscreteInputGroupObject
16#3F	SafetyDualChannelOutputObject
16#40	S_SensorCalibrationObject
16#41	EventLogObject
16#42	MotionDeviceAxisObject
16#43	TimeSyncObject
16#44	ModbusObject
16#45	OriginatorConnectionListObject
16#46	ModbusSerialLinkObject
16#47	DeviceLevelRingObject
16#48	QoSObject
16#49	SafetyAnalogInputPointObject
16#4A	SafetyAnalogInputGroupObject
16#4B	SafetyDualChannelAnalogInputObject
16#4C	SERCOSIIILinkObject
16#4D	TargetConnectionListObject
16#4E	EnergyObject
16#4F	ElectricalEnergyObject
16#50	Non_ElectricalEnergyObject
16#51	BaseSwitchObject

16#52	SNMPObject
16#53	PowerManagementObject
16#F0	ControlNetObject
16#F1	ControlNetKeeperObject
16#F2	ControlNetSchedulingObject
16#F3	ConnectionConfigurationObject
16#F4	PortObject
16#F5	TCPIPInterfaceObject
16#F6	EthernetLinkObject
16#F7	CompoNetLink
16#F8	CompoNetRepeater

**CIP and library-specific errors**

<b>Error Code</b>	<b>Definition</b>	<b>Descripción</b>
0	NO_ERROR	Service was successfully performed by the object specified.
16#1	CONNECTION_FAILURE	A connection related service failed along the connection path.
16#2	RESOURCE_UNAVAILABLE	Resources needed for the object to perform the requested service were unavailable
16#3	INVALID_PARAM_VALUE	See Status Code 0x20, which is the preferred value to use for this condition.
16#4	PATH_SEGMENT_ERROR	The path segment identifier or the segment syntax was not understood by the processing node. Path processing shall stop when a path segment error is encountered.
16#5	PATH_DESTINATION_UNKNOWN	The path is referencing an object class, instance or structure element that is not known or is not contained in the processing node. Path processing shall stop when a path destination unknown error is encountered.
16#6	PARTIAL_TRANSFER	Only part of the expected data was transferred.
16#7	CONNECTION_LOST	The messaging connection was lost.
16#8	SERVICE_NOT_SUPPORTED	The requested service was not implemented or was not defined for this Object Class/Instance.
16#9	INVALID_ATTRIBUTE_VALUE	Invalid attribute data detected.
16#A	ATTRIBUTE_LIST_ERROR	An attribute in the Get_Attribute_List or Set_Attribute_List response has a non-zero status.
16#B	ALREADY_IN_REQUEST_STATE	The object is already in the mode/state being requested by the service
16#C	OBJECT_STATE_ERROR	The object cannot perform the requested service in its current mode/state.
16#D	OBJECT_ALREADY_EXISTS	The requested instance of object to be created already exists.
16#E	ATTRIBUTE_NOT_SETTABLE	A request to modify a nonmodifiable attribute was received.

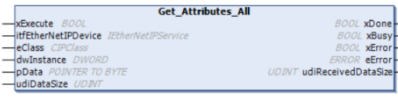
16#F	PRIVILEGE_VIOLATION	A permission/privilege check failed
16#10	DEVICE_STATE_ERROR	The device's current mode/state prohibits the execution of the requested service.
16#11	REPLY_DATA_TOO_LARGE	The data to be transmitted in the response buffer is larger than the allocated response buffer
16#12	FRAGMENTATION_OF_VALUE	The service specified an operation that is going to fragment a primitive data value, i.e. half a REAL data type.
16#13	NOT_ENOUGH_DATA	The service did not supply enough data to perform the specified operation.
16#14	ATTRIBUTE_NOT_SUPPORTED	The attribute specified in the request is not supported.
16#15	TOO_MUCH_DATA	The service supplied more data than was expected.
16#16	OBJECT_DOES_NOT_EXIST	The object specified does not exist in the device.
16#17	SERVICE_FRAGMENTATION_SEQUENCE_NOT_IN_PROGRESS	The fragmentation sequence for this service is not currently active for this data.
16#18	NO_STORED_ATTRIBUTE_DATA	The attribute data of this object was not saved prior to the requested service.
16#19	STORE_OPERATION_FAILURE	The attribute data of this object was not saved due to a failure during the attempt.
16#1A	ROUTING_FAILURE_REQUEST_PACKET_TOO_LARGE	The service request packet was too large for transmission on a CAA Net Base Services in the path to the destination. The routing device was forced to abort the service.
16#1B	ROUTING_FAILURE_RESPONSE_PACKET_TOO_LARGE	The service response packet was too large for transmission on a CAA Net Base Services in the path from the destination. The routing device was forced to abort the service.
16#1C	MISSING_ATTRIBUTE_LIST_ENTRY_DATA	The service did not supply an attribute in a list of attributes that was needed by the service to perform the requested behavior.
16#1D	INVALID_ATTRIBUTE_VALUE_LIST	The service is returning the list of attributes supplied with status information for those attributes that were invalid.
16#1E	EMBEDDED_SERVICE_ERROR	An embedded service resulted in an error.
16#1F	VENDOR_SPECIFIC_ERROR	A vendor specific error has been encountered. The Additional Code Field of the Error Response defines the particular error encountered. Use of this General Error Code should only be performed when none of the Error Codes presented in this table or within an Object Class definition accurately reflect the error.
16#20	INVALID_PARAMETER	A parameter associated with the request was invalid. This code is used when a parameter does not meet the requirements of this specification and/or the requirements defined in an Application Object Specification.
16#21	WRITE_ONCE_VALUE_OR_MEDIUM_ALREADY_WRITTEN	An attempt was made to write to a write-once medium (e.g. WORM drive, PROM) that has already been written, or to modify a value that cannot be changed once established.

16#22	INVALID_REPLY_RECEIVED	An invalid reply is received (e.g. reply service code does not match the request service code, or reply message is shorter than the minimum expected reply size). This status code can serve for other causes of invalid replies.
16#23	BUFFER_OVERFLOW	The message received is larger than the receiving buffer can handle. The entire message was discarded.
16#24	MESSAGE_FORMAT_ERROR	The format of the received message is not supported by the server.
16#25	KEY_FAILURE_IN_PATH	The Key Segment that was included as the first segment in the path does not match the destination module. The object specific status shall indicate which part of the key check failed.
16#26	PATH_SIZE_INVALID	The size of the path which was sent with the Service Request is either not large enough to allow the Request to be routed to an object or too much routing data was included.
16#27	UNEXPECTED_ATTRIBUTE_IN_LIST	An attempt was made to set an attribute that is not able to be set at this time.
16#28	INVALID_MEMBER_ID	The Member ID specified in the request does not exist in the specified Class/Instance/Attribute
16#29	MEMBER_NOT_SETTABLE	A request to modify a non-modifiable member was received
16#2A	GROUP_2_ONLY_SERVER_GENERAL_FAILURE	This error code may only be reported by DeviceNet Group 2 Only servers with 4K or less code space and only in place of Service not supported, Attribute not supported and Attribute not settable.
16#2B	UNKNOWN_MODBUS_ERROR	A CIP to Modbus translator received an unknown Modbus Exception Code.
16#2C	ATTRIBUTE_NOT_GETTABLE	A request to read a non-readable attribute was received
16#2D	INSTANCE_NOT_DELETABLE	The requested object instance cannot be deleted
16#2E	SERVICE_NOT_SUPPORTED_FOR_SPECIFIED_PATH	The object supports the service, but not for the designated application path (e.g. attribute). NOTE: Not to be used for any set service (use General Status Code 0x0E or 0x29 instead)
16#100	TIME_OUT	Request timed out.
	INTERFACE_MISSING	IEtherNetIPService is not implemented.
	REMOTE_CALL_FAILED	No physical connection.
	NULL_POINTER	Wrong input value NULL.
	INVALID_DATA_SIZE	Size of data unacceptable.
	WRONG_INTERFACE_VERSION	Version mismatch. Device implements not the same version of interface for the called method.
	NO_MEMORY	Not enough memory
	UNKNOWN_ERROR	An unknown error occurred.
	ABORTED	Service was aborted

### 5.4.5 Get\_Attributes\_All

Getting the properties and service data of an object instance.

#### ⊙ Command Format

Instruction	Name	FB/FC	LD	ST	File
Get_Attributes_All	Get all properties of an object instance instruction	FB		<pre>Get_Attributes_All( xExecute:= , xDone=&gt; , xBusy=&gt; , xError=&gt; , itfEtherNetIPDevice:= , eClass:= , dwInstance:= , eError=&gt; , pData:= , udiDataSize:= , udiReceivedDataSize=&gt; ) ;</pre>	EtherNetIPService

#### ⊙ Related Variables

##### Input variable

Input variable	Name	Data type	Range	initialization	Descriptive
xExecute	Function Block Enable	BOOL	TRUE-FALSE	-	TRUE: (rising edge signal), start function block operation
itfEtherNetIPDevice	Remote Slave Instantiation Name	IEtherNetIPService	-	-	Remote slave instantiation name
eClass	Class	ENIP. CIP Class	-	-	The class of objects to which the service refers, either CIP standard objects or third party vendors. Custom objects
dwInstance	Instance	DWORD	-	-	The actual number of instances when CIP is executed, which should be greater than or equal to 1.
pData	Receive Buffer	POINTER TO BYTE	-	-	Pointer to the attribute data, received from the target slave.
udiDataSize	Receive Length	UDINT	-	-	Received data length

##### Output variable

Output variable	Name	Data type	Range	initialization	Descriptive
xDone	Completion	BOOL	TRUE-FALSE	-	Client communication complete
xBusy	Function Block Running	BOOL	TRUE-FALSE	-	Function block running flag
xError	Error Flag	BOOL	TRUE-FALSE	-	Error Status
eError	Error	NBS.ERROR	--	-	Please refer to the CAA Net Base Services library enumeration ERROR for specific errors.
udiReceivedDataSize	Reception length	UDINT	-	-	Reception length

	Bo ole an	Bit string					Integer							Real number		Moment, Duration, Date, String				
	BOOL	BYTE	WORD	DWORD	LWORD	USINT	UINT	UDINT	ULINT	SINT	INT	DINT	LINT	REAL	LREAL	TIME	DATE	TOD	DT	STRING
xExecute	√	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
itfEtherNetIPDevice	IEtherNetIPService																			
eClass	ENIP. CIP Class																			
dwInstance	-	-	-	√	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
pData	POINTER TO BYTE																			
udiDataSize	-	-	-	-	-	-	-	√	-	-	-	-	-	-	-	-	-	-	-	-
xDone	√	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
xBusy	√	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
xError	√	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
eError	ENIP. ERROR																			
udiReceivedDataSize	-	-	-	-	-	-	-	√	-	-	-	-	-	-	-	-	-	-	-	-

### ⊙ Program demo

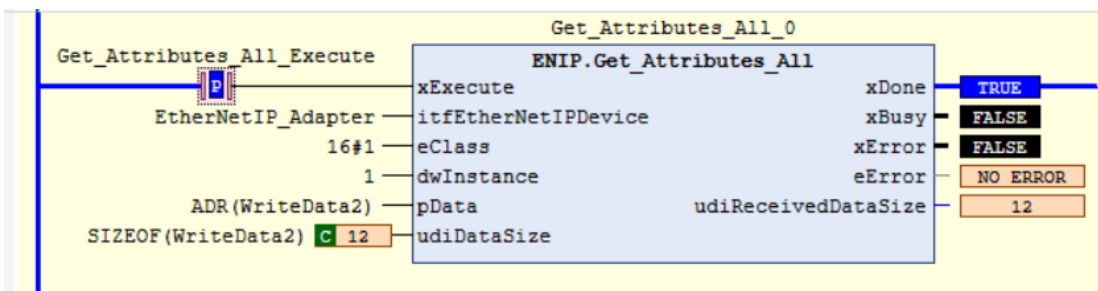
**ST:** Attribute and service data in the user program can be obtained through the function block Generic\_Service.

```

12 ● Get_Attributes_All_0(
13   xExecute TRUE := Get_Attributes_All_Execute TRUE ,
14   xDone=> ,
15   xBusy=> ,
16   xError=> ,
17   itfEtherNetIPDevice:= EtherNetIP_Adapter,
18   eClass IdentityOb := 16#1 ,
19   dwInstance 1 := 1,
20   eError=> ,
21   pData 16#B43C2014 := ADR(WriteData2),
22   udiDataSize 12 := SIZEOF(WriteData2),
23   udiReceivedDataSize=> );

```

**LD:** Attribute and service data in the user program can be obtained through the function block Generic\_Service.



### Note

itfEtherNetIPDevice is the instantiated name of the slave and eService is the CIP Public Service ID and name (see table below for details). If an error occurs in the usage function block, eError will be assigned the relevant error value (various values in the enumerated ERROR). See the following error codes for details.

#### CIP Common Services and Attributes

ID	Causality
16#1	IdentityObject
16#2	MessageRouterObject
16#3	DeviceNetObject
16#4	AssemblyObject
16#5	ConnectionObject

16#6	ConnectionManagerObject
16#7	RegisterObject
16#8	DiscreteInputPointObject
16#9	DiscreteOutputPointObject
16#A	AnalogInputPointObject
16#B	AnalogOutputPointObject
16#E	PresenceSensingObject
16#F	ParameterObject
16#10	ParameterGroupObject
16#12	GroupObject
16#1D	DiscreteInputGroupObject
16#1E	DiscreteOutputGroupObject
16#1F	DiscreteGroupObject
16#20	AnalogInputGroupObject
16#21	AnalogOutputGroupObject
16#22	AnalogGroupObject
16#23	PositionSensorObject
16#24	PositionControllerSupervisorObject
16#25	PositionControllerObject
16#26	BlockSequencerObject
16#27	CommandBlockObject
16#28	MotorDataObject
16#29	ControlSupervisorObject
16#2A	ACDCDriveObject
16#2B	AcknowledgeHandlerObject
16#2C	OverloadObject
16#2D	SoftstartObject
16#2E	SelectionObject
16#30	S_DeviceSupervisorObject
16#31	S_AnalogSensorObject
16#32	S_AnalogActuatorObject
16#33	S_SingleStageControllerObject
16#34	S_GasCalibrationObject
16#35	TripPointObject
16#37	FileObject
16#38	S_PartialPressureObject
16#39	SafetySupervisorObject
16#3A	SafetyValidatorObject

16#3B	SafetyDiscreteOutputPointObject
16#3C	SafetyDiscreteOutputGroupObject
16#3D	SafetyDiscreteInputPointObject
16#3E	SafetyDiscreteInputGroupObject
16#3F	SafetyDualChannelOutputObject
16#40	S_SensorCalibrationObject
16#41	EventLogObject
16#42	MotionDeviceAxisObject
16#43	TimeSyncObject
16#44	ModbusObject
16#45	OriginatorConnectionListObject
16#46	ModbusSerialLinkObject
16#47	DeviceLevelRingObject
16#48	QoSObject
16#49	SafetyAnalogInputPointObject
16#4A	SafetyAnalogInputGroupObject
16#4B	SafetyDualChannelAnalogInputObject
16#4C	SERCOSIIILinkObject
16#4D	TargetConnectionListObject
16#4E	EnergyObject
16#4F	ElectricalEnergyObject
16#50	Non_ElectricalEnergyObject
16#51	BaseSwitchObject
16#52	SNMPObject
16#53	PowerManagementObject
16#F0	ControlNetObject
16#F1	ControlNetKeeperObject
16#F2	ControlNetSchedulingObject
16#F3	ConnectionConfigurationObject
16#F4	PortObject
16#F5	TCPIPInterfaceObject
16#F6	EthernetLinkObject
16#F7	CompoNetLink
16#F8	CompoNetRepeater

**CIP and library-specific errors**

<b>Error Code</b>	<b>Definition</b>	<b>Descripción</b>
0	NO_ERROR	Service was successfully performed by the object specified.
16#1	CONNECTION_FAILURE	A connection related service failed along the connection path.
16#2	RESOURCE_UNAVAILABLE	Resources needed for the object to perform the requested service were unavailable
16#3	INVALID_PARAM_VALUE	See Status Code 0x20, which is the preferred value to use for this condition.
16#4	PATH_SEGMENT_ERROR	The path segment identifier or the segment syntax was not understood by the processing node. Path processing shall stop when a path segment error is encountered.
16#5	PATH_DESTINATION_UNKNOWN	The path is referencing an object class, instance or structure element that is not known or is not contained in the processing node. Path processing shall stop when a path destination unknown error is encountered.
16#6	PARTIAL_TRANSFER	Only part of the expected data was transferred.
16#7	CONNECTION_LOST	The messaging connection was lost.
16#8	SERVICE_NOT_SUPPORTED	The requested service was not implemented or was not defined for this Object Class/Instance.
16#9	INVALID_ATTRIBUTE_VALUE	Invalid attribute data detected.
16#A	ATTRIBUTE_LIST_ERROR	An attribute in the Get_Attribute_List or Set_Attribute_List response has a non-zero status.
16#B	ALREADY_IN_REQUEST_STATE	The object is already in the mode/state being requested by the service
16#C	OBJECT_STATE_ERROR	The object cannot perform the requested service in its current mode/state.
16#D	OBJECT_ALREADY_EXISTS	The requested instance of object to be created already exists.
16#E	ATTRIBUTE_NOT_SETTABLE	A request to modify a nonmodifiable attribute was received.
16#F	PRIVILEGE_VIOLATION	A permission/privilege check failed
16#10	DEVICE_STATE_ERROR	The device's current mode/state prohibits the execution of the requested service.
16#11	REPLY_DATA_TOO_LARGE	The data to be transmitted in the response buffer is larger than the allocated response buffer
16#12	FRAGMENTATION_OF_VALUE	The service specified an operation that is going to fragment a primitive data value, i.e. half a REAL data type.
16#13	NOT_ENOUGH_DATA	The service did not supply enough data to perform the specified operation.
16#14	ATTRIBUTE_NOT_SUPPORTED	The attribute specified in the request is not supported.
16#15	TOO_MUCH_DATA	The service supplied more data than was expected.
16#16	OBJECT_DOES_NOT_EXIST	The object specified does not exist in the device.


16#17	SERVICE_FRAGMENTATION_SEQUENCE_NOT_IN_PROGRESS	The fragmentation sequence for this service is not currently active for this data.
16#18	NO_STORED_ATTRIBUTE_DATA	The attribute data of this object was not saved prior to the requested service.
16#19	STORE_OPERATION_FAILURE	The attribute data of this object was not saved due to a failure during the attempt.
16#1A	ROUTING_FAILURE_REQUEST_PACKET_TOO_LARGE	The service request packet was too large for transmission on a CAA Net Base Services in the path to the destination. The routing device was forced to abort the service.
16#1B	ROUTING_FAILURE_RESPONSE_PACKET_TOO_LARGE	The service response packet was too large for transmission on a CAA Net Base Services in the path from the destination. The routing device was forced to abort the service.
16#1C	MISSING_ATTRIBUTE_LIST_ENTRY_DATA	The service did not supply an attribute in a list of attributes that was needed by the service to perform the requested behavior.
16#1D	INVALID_ATTRIBUTE_VALUE_LIST	The service is returning the list of attributes supplied with status information for those attributes that were invalid.
16#1E	EMBEDDED_SERVICE_ERROR	An embedded service resulted in an error.
16#1F	VENDOR_SPECIFIC_ERROR	A vendor specific error has been encountered. The Additional Code Field of the Error Response defines the particular error encountered. Use of this General Error Code should only be performed when none of the Error Codes presented in this table or within an Object Class definition accurately reflect the error.
16#20	INVALID_PARAMETER	A parameter associated with the request was invalid. This code is used when a parameter does not meet the requirements of this specification and/or the requirements defined in an Application Object Specification.
16#21	WRITE_ONCE_VALUE_OR_MEDIUM_ALREADY_WRITTEN	An attempt was made to write to a write-once medium (e.g. WORM drive, PROM) that has already been written, or to modify a value that cannot be changed once established.
16#22	INVALID_REPLY_RECEIVED	An invalid reply is received (e.g. reply service code does not match the request service code, or reply message is shorter than the minimum expected reply size). This status code can serve for other causes of invalid replies.
16#23	BUFFER_OVERFLOW	The message received is larger than the receiving buffer can handle. The entire message was discarded.
16#24	MESSAGE_FORMAT_ERROR	The format of the received message is not supported by the server.
16#25	KEY_FAILURE_IN_PATH	The Key Segment that was included as the first segment in the path does not match the destination module. The object specific status shall indicate which part of the key check failed.
16#26	PATH_SIZE_INVALID	The size of the path which was sent with the Service Request is either not large enough to allow the Request to be routed to an object or

		too much routing data was included.
16#27	UNEXPECTED_ATTRIBUTE_IN_LIST	An attempt was made to set an attribute that is not able to be set at this time.
16#28	INVALID_MEMBER_ID	The Member ID specified in the request does not exist in the specified Class/Instance/Attribute
16#29	MEMBER_NOT_SETTABLE	A request to modify a non-modifiable member was received
16#2A	GROUP_2_ONLY_SERVER_GENERAL_FAILURE	This error code may only be reported by DeviceNet Group 2 Only servers with 4K or less code space and only in place of Service not supported, Attribute not supported and Attribute not settable.
16#2B	UNKNOWN_MODBUS_ERROR	A CIP to Modbus translator received an unknown Modbus Exception Code.
16#2C	ATTRIBUTE_NOT_GETTABLE	A request to read a non-readable attribute was received
16#2D	INSTANCE_NOT_DELETABLE	The requested object instance cannot be deleted
16#2E	SERVICE_NOT_SUPPORTED_FOR_SPECIFIED_PATH	The object supports the service, but not for the designated application path (e.g. attribute). NOTE: Not to be used for any set service (use General Status Code 0x0E or 0x29 instead)
16#100	TIME_OUT	Request timed out.
	INTERFACE_MISSING	IEtherNetIPService is not implemented.
	REMOTE_CALL_FAILED	No physical connection.
	NULL_POINTER	Wrong input value NULL.
	INVALID_DATA_SIZE	Size of data unacceptable.
	WRONG_INTERFACE_VERSION	Version mismatch. Device implements not the same version of interface for the called method.
	NO_MEMORY	Not enough memory
	UNKNOWN_ERROR	An unknown error occurred.
	ABORTED	Service was aborted

### 5.4.6 Get\_Attribute\_Single

Getting the properties and service data of an object instance.

#### ⊙ Command Format

Instruction	Name	FB/FC	LD	ST	File
Get_Attribute_Single	Getting a single property of an object instance instruction	FB		<pre>Get_Attribute_Single( xExecute:= , xDone=&gt; , xBusy=&gt; , xError=&gt; , itfEtherNetIPDevice:= , eClass:= , dwInstance:= , eError=&gt; , wAttribute:= , pData:= , udiDataSize:= , udiReceivedDataSize=&gt; );</pre>	EtherNetIPServices

#### ⊙ Related Variables

##### Input variable

Input variable	Name	Data type	Range	initialization	Descriptive
xExecute	Function Block Enable	BOOL	TRUE-FALSE	-	TRUE: (rising edge signal), start function block operation
itfEtherNetIPDevice	Remote Slave Instantiation Name	IEtherNetIPService	-	-	Remote slave instantiation name
eClass	Class	ENIP. CIP Class	-	-	The class of objects to which the service refers, either CIP standard objects or third party vendors. Custom objects
dwInstance	Instance	DWORD	-	-	The actual number of instances when CIP is executed, which should be greater than or equal to 1.
wAttribute	Causality	WORD	-	-	Attributes pointed to by the service
pData	Receive Buffer	POINTER TO BYTE	-	-	Pointer to the attribute data, received from the target slave.
udiDataSize	Receive Length	UDINT	-	-	Received data length

##### Output variable

Output variable	Name	Data type	Range	initialization	Descriptive
xDone	Completion	BOOL	TRUE-FALSE	-	Client communication complete
xBusy	Function Block Running	BOOL	TRUE-FALSE	-	Function block running flag
xError	Error Flag	BOOL	TRUE-FALSE	-	Error Status
eError	Error	NBS.ERROR	--	-	Please refer to the CAA Net Base Services library enumeration ERROR for specific errors.
udiReceivedDataSize	Reception length	UDINT	-	-	Reception length

	Bo le an	Bit string					Integer							Real number		Moment, Duration, Date, String				
	BO OL	BY TE	WO RD	DWO RD	LWO RD	US INT	UI NT	UD INT	UL INT	SI NT	IN T	DI NT	LI NT	RE AL	LR EAL	TI ME	DA TE	TO D	DT	STR ING
xExecute	√	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
itfEtherNetIPDevice	IEtherNetIPService																			
eClass	ENIP. CIP Class																			
dwInstance	-	-	-	√	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
wAttribute	-	-	√	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
pData	POINTER TO BYTE																			
udiDataSize	-	-	-	-	-	-	-	√	-	-	-	-	-	-	-	-	-	-	-	-
xDone	√	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
xBusy	√	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
xError	√	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
eError	ENIP. ERROR																			
udiReceivedDataSize	-	-	-	-	-	-	-	√	-	-	-	-	-	-	-	-	-	-	-	-

### ⊙ Program demo

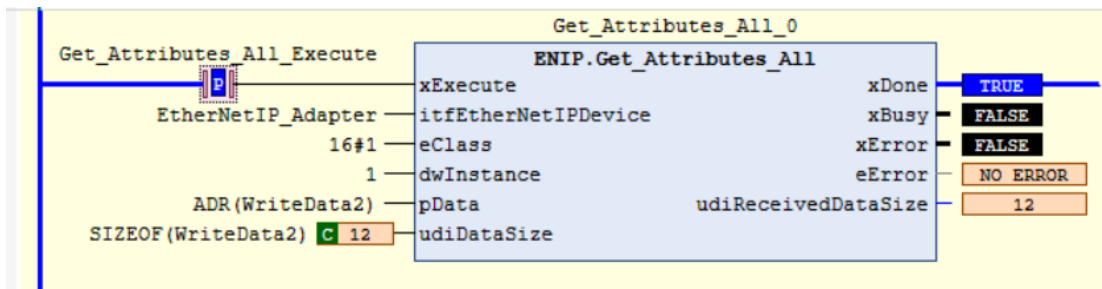
**ST:** Individual attributes of an object instance can be obtained through the function block Get\_Attributes\_Single.

```

11  ● Get_Attribute_Single_0(
12      xExecute TRUE := Get_Attribute_Single_Execute TRUE ,
13      xDone=> ,
14      xBusy=> ,
15      xError=> ,
16      itfEtherNetIPDevice:=EtherNetIP_Adapter ,
17      eClass IdentityOb :=16#1 ,
18      dwInstance 1 :=1 ,
19      eError=> ,
20      wAttribute 16 :=ENIP.CIPCommonService.SET_ATTRIBUTE_SINGLE ,
21      pData 16#B43ABE4C := ADR(WriteData3) ,
22      udiDataSize 12 :=SIZEOF(WriteData3) ,
23      udiReceivedDataSize=> );
24

```

**LD:** Individual attributes of an object instance can be obtained through the function block Get\_Attributes\_Single.



### Note

The itfEtherNetIPDevice is the instantiated name of the slave, and the eClass service points to the object class. If an error occurs in the usage function block, eError will be assigned the relevant error value (various values in the enumerated ERROR) . See the following error codes for details.

**CIP Common Services and Attributes**

<b>ID</b>	<b>Causality</b>
16#1	IdentityObject
16#2	MessageRouterObject
16#3	DeviceNetObject
16#4	AssemblyObject
16#5	ConnectionObject
16#6	ConnectionManagerObject
16#7	RegisterObject
16#8	DiscreteInputPointObject
16#9	DiscreteOutputPointObject
16#A	AnalogInputPointObject
16#B	AnalogOutputPointObject
16#E	PresenceSensingObject
16#F	ParameterObject
16#10	ParameterGroupObject
16#12	GroupObject
16#1D	DiscreteInputGroupObject
16#1E	DiscreteOutputGroupObject
16#1F	DiscreteGroupObject
16#20	AnalogInputGroupObject
16#21	AnalogOutputGroupObject
16#22	AnalogGroupObject
16#23	PositionSensorObject
16#24	PositionControllerSupervisorObject
16#25	PositionControllerObject
16#26	BlockSequencerObject
16#27	CommandBlockObject
16#28	MotorDataObject
16#29	ControlSupervisorObject
16#2A	ACDCDriveObject
16#2B	AcknowledgeHandlerObject
16#2C	OverloadObject
16#2D	SoftstartObject
16#2E	SelectionObject
16#30	S_DeviceSupervisorObject
16#31	S_AnalogSensorObject
16#32	S_AnalogActuatorObject

16#33	S_SingleStageControllerObject
16#34	S_GasCalibrationObject
16#35	TripPointObject
16#37	FileObject
16#38	S_PartialPressureObject
16#39	SafetySupervisorObject
16#3A	SafetyValidatorObject
16#3B	SafetyDiscreteOutputPointObject
16#3C	SafetyDiscreteOutputGroupObject
16#3D	SafetyDiscreteInputPointObject
16#3E	SafetyDiscreteInputGroupObject
16#3F	SafetyDualChannelOutputObject
16#40	S_SensorCalibrationObject
16#41	EventLogObject
16#42	MotionDeviceAxisObject
16#43	TimeSyncObject
16#44	ModbusObject
16#45	OriginatorConnectionListObject
16#46	ModbusSerialLinkObject
16#47	DeviceLevelRingObject
16#48	QoSObject
16#49	SafetyAnalogInputPointObject
16#4A	SafetyAnalogInputGroupObject
16#4B	SafetyDualChannelAnalogInputObject
16#4C	SERCOSIIILinkObject
16#4D	TargetConnectionListObject
16#4E	EnergyObject
16#4F	ElectricalEnergyObject
16#50	Non_ElectricalEnergyObject
16#51	BaseSwitchObject
16#52	SNMPObject
16#53	PowerManagementObject
16#F0	ControlNetObject
16#F1	ControlNetKeeperObject
16#F2	ControlNetSchedulingObject
16#F3	ConnectionConfigurationObject
16#F4	PortObject
16#F5	TCPIPIInterfaceObject

16#F6	EthernetLinkObject
16#F7	CompoNetLink
16#F8	CompoNetRepeater

**CIP and library-specific errors**

<b>Error Code</b>	<b>Definition</b>	<b>Descripción</b>
0	NO_ERROR	Service was successfully performed by the object specified.
16#1	CONNECTION_FAILURE	A connection related service failed along the connection path.
16#2	RESOURCE_UNAVAILABLE	Resources needed for the object to perform the requested service were unavailable
16#3	INVALID_PARAM_VALUE	See Status Code 0x20, which is the preferred value to use for this condition.
16#4	PATH_SEGMENT_ERROR	The path segment identifier or the segment syntax was not understood by the processing node. Path processing shall stop when a path segment error is encountered.
16#5	PATH_DESTINATION_UNKNOWN	The path is referencing an object class, instance or structure element that is not known or is not contained in the processing node. Path processing shall stop when a path destination unknown error is encountered.
16#6	PARTIAL_TRANSFER	Only part of the expected data was transferred.
16#7	CONNECTION_LOST	The messaging connection was lost.
16#8	SERVICE_NOT_SUPPORTED	The requested service was not implemented or was not defined for this Object Class/Instance.
16#9	INVALID_ATTRIBUTE_VALUE	Invalid attribute data detected.
16#A	ATTRIBUTE_LIST_ERROR	An attribute in the Get_Attribute_List or Set_Attribute_List response has a non-zero status.
16#B	ALREADY_IN_REQUEST_STATE	The object is already in the mode/state being requested by the service
16#C	OBJECT_STATE_ERROR	The object cannot perform the requested service in its current mode/state.
16#D	OBJECT_ALREADY_EXISTS	The requested instance of object to be created already exists.
16#E	ATTRIBUTE_NOT_SETTABLE	A request to modify a nonmodifiable attribute was received.
16#F	PRIVILEGE_VIOLATION	A permission/privilege check failed
16#10	DEVICE_STATE_ERROR	The device's current mode/state prohibits the execution of the requested service.
16#11	REPLY_DATA_TOO_LARGE	The data to be transmitted in the response buffer is larger than the allocated response buffer
16#12	FRAGMENTATION_OF_VALUE	The service specified an operation that is going to fragment a primitive data value, i.e. half a REAL data type.
16#13	NOT_ENOUGH_DATA	The service did not supply enough data to perform the specified operation.
16#14	ATTRIBUTE_NOT_SUPPORTED	The attribute specified in the request is not supported.


16#15	TOO_MUCH_DATA	The service supplied more data than was expected.
16#16	OBJECT_DOES_NOT_EXIST	The object specified does not exist in the device.
16#17	SERVICE_FRAGMENTATION_SEQUENCE_NOT_IN_PROGRESS	The fragmentation sequence for this service is not currently active for this data.
16#18	NO_STORED_ATTRIBUTE_DATA	The attribute data of this object was not saved prior to the requested service.
16#19	STORE_OPERATION_FAILURE	The attribute data of this object was not saved due to a failure during the attempt.
16#1A	ROUTING_FAILURE_REQUEST_PACKET_TOO_LARGE	The service request packet was too large for transmission on a CAA Net Base Services in the path to the destination. The routing device was forced to abort the service.
16#1B	ROUTING_FAILURE_RESPONSE_PACKET_TOO_LARGE	The service response packet was too large for transmission on a CAA Net Base Services in the path from the destination. The routing device was forced to abort the service.
16#1C	MISSING_ATTRIBUTE_LIST_ENTRY_DATA	The service did not supply an attribute in a list of attributes that was needed by the service to perform the requested behavior.
16#1D	INVALID_ATTRIBUTE_VALUE_LIST	The service is returning the list of attributes supplied with status information for those attributes that were invalid.
16#1E	EMBEDDED_SERVICE_ERROR	An embedded service resulted in an error.
16#1F	VENDOR_SPECIFIC_ERROR	A vendor specific error has been encountered. The Additional Code Field of the Error Response defines the particular error encountered. Use of this General Error Code should only be performed when none of the Error Codes presented in this table or within an Object Class definition accurately reflect the error.
16#20	INVALID_PARAMETER	A parameter associated with the request was invalid. This code is used when a parameter does not meet the requirements of this specification and/or the requirements defined in an Application Object Specification.
16#21	WRITE_ONCE_VALUE_OR_MEDIUM_ALREADY_WRITTEN	An attempt was made to write to a write-once medium (e.g. WORM drive, PROM) that has already been written, or to modify a value that cannot be changed once established.
16#22	INVALID_REPLY_RECEIVED	An invalid reply is received (e.g. reply service code does not match the request service code, or reply message is shorter than the minimum expected reply size). This status code can serve for other causes of invalid replies.
16#23	BUFFER_OVERFLOW	The message received is larger than the receiving buffer can handle. The entire message was discarded.
16#24	MESSAGE_FORMAT_ERROR	The format of the received message is not supported by the server.
16#25	KEY_FAILURE_IN_PATH	The Key Segment that was included as the first segment in the path does not match the destination module. The object specific status shall indicate which part of the key check

		failed.
16#26	PATH_SIZE_INVALID	The size of the path which was sent with the Service Request is either not large enough to allow the Request to be routed to an object or too much routing data was included.
16#27	UNEXPECTED_ATTRIBUTE_IN_LIST	An attempt was made to set an attribute that is not able to be set at this time.
16#28	INVALID_MEMBER_ID	The Member ID specified in the request does not exist in the specified Class/Instance/Attribute
16#29	MEMBER_NOT_SETTABLE	A request to modify a non-modifiable member was received
16#2A	GROUP_2_ONLY_SERVER_GENERAL_FAILURE	This error code may only be reported by DeviceNet Group 2 Only servers with 4K or less code space and only in place of Service not supported, Attribute not supported and Attribute not settable.
16#2B	UNKNOWN_MODBUS_ERROR	A CIP to Modbus translator received an unknown Modbus Exception Code.
16#2C	ATTRIBUTE_NOT_GETTABLE	A request to read a non-readable attribute was received
16#2D	INSTANCE_NOT_DELETABLE	The requested object instance cannot be deleted
16#2E	SERVICE_NOT_SUPPORTED_FOR_SPECIFIED_PATH	The object supports the service, but not for the designated application path (e.g. attribute). NOTE: Not to be used for any set service (use General Status Code 0x0E or 0x29 instead)
16#100	TIME_OUT	Request timed out.
	INTERFACE_MISSING	IEtherNetIPService is not implemented.
	REMOTE_CALL_FAILED	No physical connection.
	NULL_POINTER	Wrong input value NULL.
	INVALID_DATA_SIZE	Size of data unacceptable.
	WRONG_INTERFACE_VERSION	Version mismatch. Device implements not the same version of interface for the called method.
	NO_MEMORY	Not enough memory
	UNKNOWN_ERROR	An unknown error occurred.
	ABORTED	Service was aborted

### 5.4.7 Set\_Attributes\_All

Getting the properties and service data of an object instance.

#### ⊙ Command Format

Instruction	Name	FB/FC	LD	ST	File
Set_Attributes_All	Setting all properties of an object instance instruction	FB		<pre>Set_Attributes_All( xExecute:= , xDone=&gt; , xBusy=&gt; , xError=&gt; , itfEtherNetIPDevice:= , eClass:= , dwInstance:= , eError=&gt; , pData:= , udiDataSize:= );</pre>	EtherNetIPServices

#### ⊙ Related Variables

##### Input variable

Input variable	Name	Data type	Range	initialization	Descriptive
xExecute	Function Block Enable	BOOL	TRUE-FALSE	-	TRUE: (rising edge signal), start function block operation
itfEtherNetIPDevice	Remote Slave Instantiation Name	IEtherNetIPService	-	-	Remote slave instantiation name
eClass	Class	ENIP_Class	-	-	The class of objects to which the service refers, either CIP standard objects or third party vendors. Custom objects
dwInstance	Instance	DWORD	-	-	The actual number of instances when CIP is executed, which should be greater than or equal to 1.
pData	Receive Buffer	POINTER TO BYTE	-	-	Pointer to the attribute data, received from the target slave.
udiDataSize	Receive Length	UDINT	-	-	Received data length

##### Output variable

Output variable	Name	Data type	Range	initialization	Descriptive
xDone	Completion	BOOL	TRUE-FALSE	-	Client communication complete
xBusy	Function Block Running	BOOL	TRUE-FALSE	-	Function block running flag
xError	Error Flag	BOOL	TRUE-FALSE	-	Error Status
eError	Error	NBS.ERROR	--	-	Please refer to the CAA Net Base Services library enumeration ERROR for specific errors.

	Bo le an	Bit string					Integer							Real number		Moment, Duration, Date, String				
	BO OL	BY TE	WO RD	DWO RD	LWO RD	US INT	UI NT	UD INT	UL INT	SI NT	IN T	DI NT	LI NT	RE AL	LR EAL	TI ME	DA TE	TO D	DT	ST RING
xExecute	√	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
itfEtherNetIPDevice	IEtherNetIPService																			
eClass	ENIP. CIP Class																			
dwInstance	-	-	-	√	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
pData	POINTER TO BYTE																			
udiDataSize	-	-	-	-	-	-	-	√	-	-	-	-	-	-	-	-	-	-	-	-
xDone	√	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
xBusy	√	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
xError	√	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
eError	ENIP. ERROR																			

**⊙ Program demo**

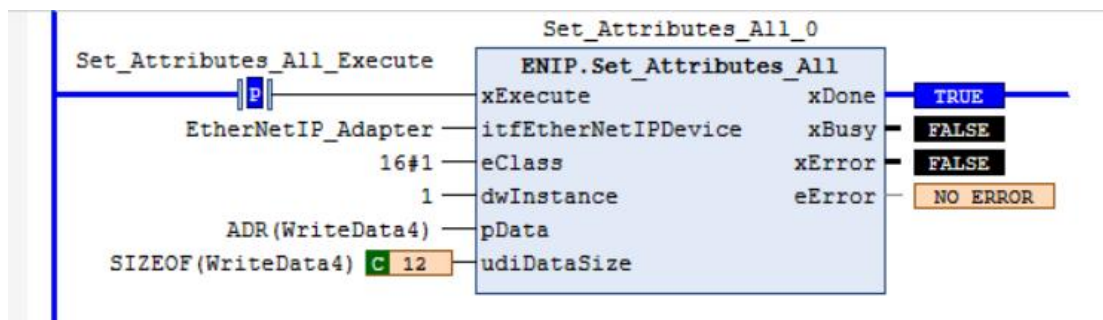
**ST:** All attributes and service data of an object instance can be set via the function block Set\_Attributes\_All.

```

● Set_Attributes_All_0(
    xExecute TRUE := Set_Attributes_All_Execute TRUE,
    xDone=> ,
    xBusy=> ,
    xError=> ,
    itfEtherNetIPDevice:=EtherNetIP_Adapter ,
    eClass IdentityOb := 16#1 ,
    dwInstance 1 := 1,
    eError=> ,
    pData 16#B43F0008 := ADR(WriteData4),
    udiDataSize 12 := SIZEOF(WriteData4));

```

**LD:** All attributes and service data of an object instance can be set via the function block Set\_Attributes\_All.


**Note**

The itfEtherNetIPDevice is the instantiated name of the slave, and the eClass service points to the object class. If an error occurs in the usage function block, eError will be assigned the relevant error value (various values in the enumerated ERROR) . See the following error codes for details.

**CIP Common Services and Attributes**

<b>ID</b>	<b>Causality</b>
16#1	IdentityObject
16#2	MessageRouterObject
16#3	DeviceNetObject
16#4	AssemblyObject
16#5	ConnectionObject
16#6	ConnectionManagerObject
16#7	RegisterObject
16#8	DiscreteInputPointObject
16#9	DiscreteOutputPointObject
16#A	AnalogInputPointObject
16#B	AnalogOutputPointObject
16#E	PresenceSensingObject
16#F	ParameterObject
16#10	ParameterGroupObject
16#12	GroupObject
16#1D	DiscreteInputGroupObject
16#1E	DiscreteOutputGroupObject
16#1F	DiscreteGroupObject
16#20	AnalogInputGroupObject
16#21	AnalogOutputGroupObject
16#22	AnalogGroupObject
16#23	PositionSensorObject
16#24	PositionControllerSupervisorObject
16#25	PositionControllerObject
16#26	BlockSequencerObject
16#27	CommandBlockObject
16#28	MotorDataObject
16#29	ControlSupervisorObject
16#2A	ACDCDriveObject
16#2B	AcknowledgeHandlerObject
16#2C	OverloadObject
16#2D	SoftstartObject
16#2E	SelectionObject
16#30	S_DeviceSupervisorObject
16#31	S_AnalogSensorObject
16#32	S_AnalogActuatorObject
16#33	S_SingleStageControllerObject

16#34	S_GasCalibrationObject
16#35	TripPointObject
16#37	FileObject
16#38	S_PartialPressureObject
16#39	SafetySupervisorObject
16#3A	SafetyValidatorObject
16#3B	SafetyDiscreteOutputPointObject
16#3C	SafetyDiscreteOutputGroupObject
16#3D	SafetyDiscreteInputPointObject
16#3E	SafetyDiscreteInputGroupObject
16#3F	SafetyDualChannelOutputObject
16#40	S_SensorCalibrationObject
16#41	EventLogObject
16#42	MotionDeviceAxisObject
16#43	TimeSyncObject
16#44	ModbusObject
16#45	OriginatorConnectionListObject
16#46	ModbusSerialLinkObject
16#47	DeviceLevelRingObject
16#48	QoSObject
16#49	SafetyAnalogInputPointObject
16#4A	SafetyAnalogInputGroupObject
16#4B	SafetyDualChannelAnalogInputObject
16#4C	SERCOSIIILinkObject
16#4D	TargetConnectionListObject
16#4E	EnergyObject
16#4F	ElectricalEnergyObject
16#50	Non_ElectricalEnergyObject
16#51	BaseSwitchObject
16#52	SNMPObject
16#53	PowerManagementObject
16#F0	ControlNetObject
16#F1	ControlNetKeeperObject
16#F2	ControlNetSchedulingObject
16#F3	ConnectionConfigurationObject
16#F4	PortObject
16#F5	TCPIPIInterfaceObject
16#F6	EthernetLinkObject

16#F7	CompoNetLink
16#F8	CompoNetRepeater

**CIP and library-specific errors**

<b>Error Code</b>	<b>Definition</b>	<b>Descripción</b>
0	NO_ERROR	Service was successfully performed by the object specified.
16#1	CONNECTION_FAILURE	A connection related service failed along the connection path.
16#2	RESOURCE_UNAVAILABLE	Resources needed for the object to perform the requested service were unavailable
16#3	INVALID_PARAM_VALUE	See Status Code 0x20, which is the preferred value to use for this condition.
16#4	PATH_SEGMENT_ERROR	The path segment identifier or the segment syntax was not understood by the processing node. Path processing shall stop when a path segment error is encountered.
16#5	PATH_DESTINATION_UNKNOWN	The path is referencing an object class, instance or structure element that is not known or is not contained in the processing node. Path processing shall stop when a path destination unknown error is encountered.
16#6	PARTIAL_TRANSFER	Only part of the expected data was transferred.
16#7	CONNECTION_LOST	The messaging connection was lost.
16#8	SERVICE_NOT_SUPPORTED	The requested service was not implemented or was not defined for this Object Class/Instance.
16#9	INVALID_ATTRIBUTE_VALUE	Invalid attribute data detected.
16#A	ATTRIBUTE_LIST_ERROR	An attribute in the Get_Attribute_List or Set_Attribute_List response has a non-zero status.
16#B	ALREADY_IN_REQUEST_STATE	The object is already in the mode/state being requested by the service
16#C	OBJECT_STATE_ERROR	The object cannot perform the requested service in its current mode/state.
16#D	OBJECT_ALREADY_EXISTS	The requested instance of object to be created already exists.
16#E	ATTRIBUTE_NOT_SETTABLE	A request to modify a nonmodifiable attribute was received.
16#F	PRIVILEGE_VIOLATION	A permission/privilege check failed
16#10	DEVICE_STATE_ERROR	The device's current mode/state prohibits the execution of the requested service.
16#11	REPLY_DATA_TOO_LARGE	The data to be transmitted in the response buffer is larger than the allocated response buffer
16#12	FRAGMENTATION_OF_VALUE	The service specified an operation that is going to fragment a primitive data value, i.e. half a REAL data type.
16#13	NOT_ENOUGH_DATA	The service did not supply enough data to perform the specified operation.
16#14	ATTRIBUTE_NOT_SUPPORTED	The attribute specified in the request is not supported.


16#15	TOO_MUCH_DATA	The service supplied more data than was expected.
16#16	OBJECT_DOES_NOT_EXIST	The object specified does not exist in the device.
16#17	SERVICE_FRAGMENTATION_SEQUENCE_NOT_IN_PROGRESS	The fragmentation sequence for this service is not currently active for this data.
16#18	NO_STORED_ATTRIBUTE_DATA	The attribute data of this object was not saved prior to the requested service.
16#19	STORE_OPERATION_FAILURE	The attribute data of this object was not saved due to a failure during the attempt.
16#1A	ROUTING_FAILURE_REQUEST_PACKET_TOO_LARGE	The service request packet was too large for transmission on a CAA Net Base Services in the path to the destination. The routing device was forced to abort the service.
16#1B	ROUTING_FAILURE_RESPONSE_PACKET_TOO_LARGE	The service response packet was too large for transmission on a CAA Net Base Services in the path from the destination. The routing device was forced to abort the service.
16#1C	MISSING_ATTRIBUTE_LIST_ENTRY_DATA	The service did not supply an attribute in a list of attributes that was needed by the service to perform the requested behavior.
16#1D	INVALID_ATTRIBUTE_VALUE_LIST	The service is returning the list of attributes supplied with status information for those attributes that were invalid.
16#1E	EMBEDDED_SERVICE_ERROR	An embedded service resulted in an error.
16#1F	VENDOR_SPECIFIC_ERROR	A vendor specific error has been encountered. The Additional Code Field of the Error Response defines the particular error encountered. Use of this General Error Code should only be performed when none of the Error Codes presented in this table or within an Object Class definition accurately reflect the error.
16#20	INVALID_PARAMETER	A parameter associated with the request was invalid. This code is used when a parameter does not meet the requirements of this specification and/or the requirements defined in an Application Object Specification.
16#21	WRITE_ONCE_VALUE_OR_MEDIUM_ALREADY_WRITTEN	An attempt was made to write to a write-once medium (e.g. WORM drive, PROM) that has already been written, or to modify a value that cannot be changed once established.
16#22	INVALID_REPLY_RECEIVED	An invalid reply is received (e.g. reply service code does not match the request service code, or reply message is shorter than the minimum expected reply size). This status code can serve for other causes of invalid replies.
16#23	BUFFER_OVERFLOW	The message received is larger than the receiving buffer can handle. The entire message was discarded.
16#24	MESSAGE_FORMAT_ERROR	The format of the received message is not supported by the server.
16#25	KEY_FAILURE_IN_PATH	The Key Segment that was included as the first segment in the path does not match the destination module. The object specific status shall indicate which part of the key check

		failed.
16#26	PATH_SIZE_INVALID	The size of the path which was sent with the Service Request is either not large enough to allow the Request to be routed to an object or too much routing data was included.
16#27	UNEXPECTED_ATTRIBUTE_IN_LIST	An attempt was made to set an attribute that is not able to be set at this time.
16#28	INVALID_MEMBER_ID	The Member ID specified in the request does not exist in the specified Class/Instance/Attribute
16#29	MEMBER_NOT_SETTABLE	A request to modify a non-modifiable member was received
16#2A	GROUP_2_ONLY_SERVER_GENERAL_FAILURE	This error code may only be reported by DeviceNet Group 2 Only servers with 4K or less code space and only in place of Service not supported, Attribute not supported and Attribute not settable.
16#2B	UNKNOWN_MODBUS_ERROR	A CIP to Modbus translator received an unknown Modbus Exception Code.
16#2C	ATTRIBUTE_NOT_GETTABLE	A request to read a non-readable attribute was received
16#2D	INSTANCE_NOT_DELETABLE	The requested object instance cannot be deleted
16#2E	SERVICE_NOT_SUPPORTED_FOR_SPECIFIED_PATH	The object supports the service, but not for the designated application path (e.g. attribute). NOTE: Not to be used for any set service (use General Status Code 0x0E or 0x29 instead)
16#100	TIME_OUT	Request timed out.
	INTERFACE_MISSING	IEtherNetIPService is not implemented.
	REMOTE_CALL_FAILED	No physical connection.
	NULL_POINTER	Wrong input value NULL.
	INVALID_DATA_SIZE	Size of data unacceptable.
	WRONG_INTERFACE_VERSION	Version mismatch. Device implements not the same version of interface for the called method.
	NO_MEMORY	Not enough memory
	UNKNOWN_ERROR	An unknown error occurred.
	ABORTED	Service was aborted

### 5.4.8 Get\_Attribute\_Single

Getting the properties and service data of an object instance.

#### ⊙ Command Format

Instruction	Name	FB/FC	LD	ST	File
Set_Attribute_Single	Setting a single property of an object instance instruction	FB		<pre>Set_Attribute_Single( xExecute:= , xDone=&gt; , xBusy=&gt; , xError=&gt; , itfEtherNetIPDevice:= , eClass:= , dwInstance:= , eError=&gt; , wAttribute:= , pData:= , udiDataSize:= );</pre>	EtherNetIPServices

#### ⊙ Related Variables

##### Input variable

Input variable	Name	Data type	Range	initialization	Descriptive
xExecute	Function Block Enable	BOOL	TRUE-FALSE	-	TRUE: (rising edge signal), start function block operation
itfEtherNetIPDevice	Remote Slave Instantiation Name	IEtherNetIPService	-	-	Remote slave instantiation name
eClass	Class	ENIP. CIP Class	-	-	The class of objects to which the service refers, either CIP standard objects or third party vendors. Custom objects
dwInstance	Instance	DWORD	-	-	The actual number of instances when CIP is executed, which should be greater than or equal to 1.
wAttribute	Causality	WORD	-	-	Attributes pointed to by the service
pData	Receive Buffer	POINTER TO BYTE	-	-	Pointer to the attribute data, received from the target slave.
udiDataSize	Receive Length	UDINT	-	-	Received data length

##### Output variable

Output variable	Name	Data type	Range	initialization	Descriptive
xDone	Completion	BOOL	TRUE-FALSE	-	Client communication complete
xBusy	Function Block Running	BOOL	TRUE-FALSE	-	Function block running flag
xError	Error Flag	BOOL	TRUE-FALSE	-	Error Status
eError	Error	NBS.ERROR	--	-	Please refer to the CAA Net Base Services library enumeration ERROR for specific errors.

	Bo le an	Bit string					Integer						Real number		Moment, Duration, Date, String					
	BOOL	BYTE	WORD	DWORD	LWORD	USINT	UINT	UDINT	ULINT	SINT	INT	DINT	LINT	REAL	LREAL	TIME	DATE	TOD	DT	STRING
xExecute	√	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
itfEtherNetIPDevice	IEtherNetIPService																			
eClass	ENIP. CIP Class																			
dwInstance	-	-	-	√	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
wAttribute	-	-	√	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
pData	POINTER TO BYTE																			
udiDataSize	-	-	-	-	-	-	-	√	-	-	-	-	-	-	-	-	-	-	-	-
xDone	√	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
xBusy	√	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
xError	√	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
eError	ENIP. ERROR																			

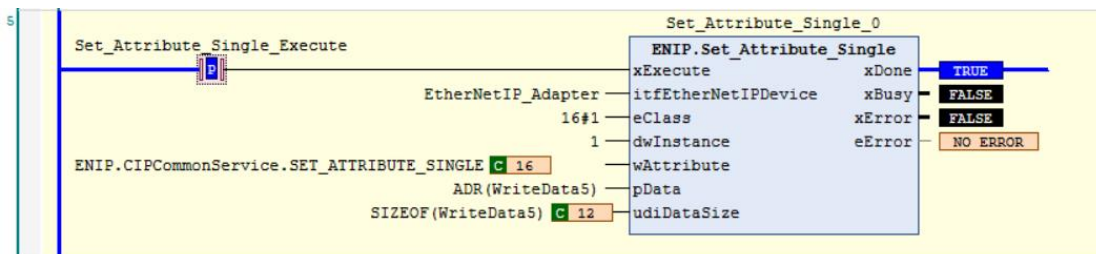
### ⊙ Program demo

**ST:** Individual attributes and service data for an object instance can be set via the function block Set\_Attributes\_Single.

```

● Set_Attribute_Single_0 (
  xExecute TRUE := Set_Attribute_Single_Execute TRUE ,
  xDone=> ,
  xBusy=> ,
  xError=> ,
  itfEtherNetIPDevice:= EtherNetIP_Adapter,
  eClass IdentityOb := 16#1,
  dwInstance 1 := 1,
  eError=> ,
  wAttribute 16 := ENIP.CIPCommonService.SET_ATTRIBUTE_SINGLE ,
  pData 16#B43EFFC8 := ADR(WriteData5) ,
  udiDataSize 12 := SIZEOF(WriteData5));
  
```

**LD:** Individual attributes and service data for an object instance can be set via the function block Set\_Attributes\_Single.



### Note

The itfEtherNetIPDevice is the instantiated name of the slave, and the eClass service points to the object class. If an error occurs in the usage function block, eError will be assigned the relevant error value (various values in the enumerated ERROR) . See the following error codes for details.

**CIP Common Services and Attributes**

<b>ID</b>	<b>Causality</b>
16#1	IdentityObject
16#2	MessageRouterObject
16#3	DeviceNetObject
16#4	AssemblyObject
16#5	ConnectionObject
16#6	ConnectionManagerObject
16#7	RegisterObject
16#8	DiscreteInputPointObject
16#9	DiscreteOutputPointObject
16#A	AnalogInputPointObject
16#B	AnalogOutputPointObject
16#E	PresenceSensingObject
16#F	ParameterObject
16#10	ParameterGroupObject
16#12	GroupObject
16#1D	DiscreteInputGroupObject
16#1E	DiscreteOutputGroupObject
16#1F	DiscreteGroupObject
16#20	AnalogInputGroupObject
16#21	AnalogOutputGroupObject
16#22	AnalogGroupObject
16#23	PositionSensorObject
16#24	PositionControllerSupervisorObject
16#25	PositionControllerObject
16#26	BlockSequencerObject
16#27	CommandBlockObject
16#28	MotorDataObject
16#29	ControlSupervisorObject
16#2A	ACDCDriveObject
16#2B	AcknowledgeHandlerObject
16#2C	OverloadObject
16#2D	SoftstartObject
16#2E	SelectionObject
16#30	S_DeviceSupervisorObject
16#31	S_AnalogSensorObject
16#32	S_AnalogActuatorObject
16#33	S_SingleStageControllerObject

16#34	S_GasCalibrationObject
16#35	TripPointObject
16#37	FileObject
16#38	S_PartialPressureObject
16#39	SafetySupervisorObject
16#3A	SafetyValidatorObject
16#3B	SafetyDiscreteOutputPointObject
16#3C	SafetyDiscreteOutputGroupObject
16#3D	SafetyDiscreteInputPointObject
16#3E	SafetyDiscreteInputGroupObject
16#3F	SafetyDualChannelOutputObject
16#40	S_SensorCalibrationObject
16#41	EventLogObject
16#42	MotionDeviceAxisObject
16#43	TimeSyncObject
16#44	ModbusObject
16#45	OriginatorConnectionListObject
16#46	ModbusSerialLinkObject
16#47	DeviceLevelRingObject
16#48	QoSObject
16#49	SafetyAnalogInputPointObject
16#4A	SafetyAnalogInputGroupObject
16#4B	SafetyDualChannelAnalogInputObject
16#4C	SERCOSIIILinkObject
16#4D	TargetConnectionListObject
16#4E	EnergyObject
16#4F	ElectricalEnergyObject
16#50	Non_ElectricalEnergyObject
16#51	BaseSwitchObject
16#52	SNMPObject
16#53	PowerManagementObject
16#F0	ControlNetObject
16#F1	ControlNetKeeperObject
16#F2	ControlNetSchedulingObject
16#F3	ConnectionConfigurationObject
16#F4	PortObject
16#F5	TCPIPIInterfaceObject
16#F6	EthernetLinkObject

16#F7	CompoNetLink
16#F8	CompoNetRepeater

**CIP and library-specific errors**

<b>Error Code</b>	<b>Definition</b>	<b>Descripción</b>
0	NO_ERROR	Service was successfully performed by the object specified.
16#1	CONNECTION_FAILURE	A connection related service failed along the connection path.
16#2	RESOURCE_UNAVAILABLE	Resources needed for the object to perform the requested service were unavailable
16#3	INVALID_PARAM_VALUE	See Status Code 0x20, which is the preferred value to use for this condition.
16#4	PATH_SEGMENT_ERROR	The path segment identifier or the segment syntax was not understood by the processing node. Path processing shall stop when a path segment error is encountered.
16#5	PATH_DESTINATION_UNKNOWN	The path is referencing an object class, instance or structure element that is not known or is not contained in the processing node. Path processing shall stop when a path destination unknown error is encountered.
16#6	PARTIAL_TRANSFER	Only part of the expected data was transferred.
16#7	CONNECTION_LOST	The messaging connection was lost.
16#8	SERVICE_NOT_SUPPORTED	The requested service was not implemented or was not defined for this Object Class/Instance.
16#9	INVALID_ATTRIBUTE_VALUE	Invalid attribute data detected.
16#A	ATTRIBUTE_LIST_ERROR	An attribute in the Get_Attribute_List or Set_Attribute_List response has a non-zero status.
16#B	ALREADY_IN_REQUEST_STATE	The object is already in the mode/state being requested by the service
16#C	OBJECT_STATE_ERROR	The object cannot perform the requested service in its current mode/state.
16#D	OBJECT_ALREADY_EXISTS	The requested instance of object to be created already exists.
16#E	ATTRIBUTE_NOT_SETTABLE	A request to modify a nonmodifiable attribute was received.
16#F	PRIVILEGE_VIOLATION	A permission/privilege check failed
16#10	DEVICE_STATE_ERROR	The device's current mode/state prohibits the execution of the requested service.
16#11	REPLY_DATA_TOO_LARGE	The data to be transmitted in the response buffer is larger than the allocated response buffer
16#12	FRAGMENTATION_OF_VALUE	The service specified an operation that is going to fragment a primitive data value, i.e. half a REAL data type.
16#13	NOT_ENOUGH_DATA	The service did not supply enough data to perform the specified operation.
16#14	ATTRIBUTE_NOT_SUPPORTED	The attribute specified in the request is not supported.

16#15	TOO_MUCH_DATA	The service supplied more data than was expected.
16#16	OBJECT_DOES_NOT_EXIST	The object specified does not exist in the device.
16#17	SERVICE_FRAGMENTATION_SEQUENCE_NOT_IN_PROGRESS	The fragmentation sequence for this service is not currently active for this data.
16#18	NO_STORED_ATTRIBUTE_DATA	The attribute data of this object was not saved prior to the requested service.
16#19	STORE_OPERATION_FAILURE	The attribute data of this object was not saved due to a failure during the attempt.
16#1A	ROUTING_FAILURE_REQUEST_PACKET_TOO_LARGE	The service request packet was too large for transmission on a CAA Net Base Services in the path to the destination. The routing device was forced to abort the service.
16#1B	ROUTING_FAILURE_RESPONSE_PACKET_TOO_LARGE	The service response packet was too large for transmission on a CAA Net Base Services in the path from the destination. The routing device was forced to abort the service.
16#1C	MISSING_ATTRIBUTE_LIST_ENTRY_DATA	The service did not supply an attribute in a list of attributes that was needed by the service to perform the requested behavior.
16#1D	INVALID_ATTRIBUTE_VALUE_LIST	The service is returning the list of attributes supplied with status information for those attributes that were invalid.
16#1E	EMBEDDED_SERVICE_ERROR	An embedded service resulted in an error.
16#1F	VENDOR_SPECIFIC_ERROR	A vendor specific error has been encountered. The Additional Code Field of the Error Response defines the particular error encountered. Use of this General Error Code should only be performed when none of the Error Codes presented in this table or within an Object Class definition accurately reflect the error.
16#20	INVALID_PARAMETER	A parameter associated with the request was invalid. This code is used when a parameter does not meet the requirements of this specification and/or the requirements defined in an Application Object Specification.
16#21	WRITE_ONCE_VALUE_OR_MEDIUM_ALREADY_WRITTEN	An attempt was made to write to a write-once medium (e.g. WORM drive, PROM) that has already been written, or to modify a value that cannot be changed once established.
16#22	INVALID_REPLY_RECEIVED	An invalid reply is received (e.g. reply service code does not match the request service code, or reply message is shorter than the minimum expected reply size). This status code can serve for other causes of invalid replies.
16#23	BUFFER_OVERFLOW	The message received is larger than the receiving buffer can handle. The entire message was discarded.
16#24	MESSAGE_FORMAT_ERROR	The format of the received message is not supported by the server.
16#25	KEY_FAILURE_IN_PATH	The Key Segment that was included as the first segment in the path does not match the destination module. The object specific status shall indicate which part of the key check

		failed.
16#26	PATH_SIZE_INVALID	The size of the path which was sent with the Service Request is either not large enough to allow the Request to be routed to an object or too much routing data was included.
16#27	UNEXPECTED_ATTRIBUTE_IN_LIST	An attempt was made to set an attribute that is not able to be set at this time.
16#28	INVALID_MEMBER_ID	The Member ID specified in the request does not exist in the specified Class/Instance/Attribute
16#29	MEMBER_NOT_SETTABLE	A request to modify a non-modifiable member was received
16#2A	GROUP_2_ONLY_SERVER_GENERAL_FAILURE	This error code may only be reported by DeviceNet Group 2 Only servers with 4K or less code space and only in place of Service not supported, Attribute not supported and Attribute not settable.
16#2B	UNKNOWN_MODBUS_ERROR	A CIP to Modbus translator received an unknown Modbus Exception Code.
16#2C	ATTRIBUTE_NOT_GETTABLE	A request to read a non-readable attribute was received
16#2D	INSTANCE_NOT_DELETABLE	The requested object instance cannot be deleted
16#2E	SERVICE_NOT_SUPPORTED_FOR_SPECIFIED_PATH	The object supports the service, but not for the designated application path (e.g. attribute). NOTE: Not to be used for any set service (use General Status Code 0x0E or 0x29 instead)
16#100	TIME_OUT	Request timed out.
	INTERFACE_MISSING	IEtherNetIPService is not implemented.
	REMOTE_CALL_FAILED	No physical connection.
	NULL_POINTER	Wrong input value NULL.
	INVALID_DATA_SIZE	Size of data unacceptable.
	WRONG_INTERFACE_VERSION	Version mismatch. Device implements not the same version of interface for the called method.
	NO_MEMORY	Not enough memory
	UNKNOWN_ERROR	An unknown error occurred.
	ABORTED	Service was aborted

## 6. High-speed IO instruction

### 6.1 Pulse axis control


#### 6.1.1 Instruction List

Instruction Type	Name	FB/FC	Functionality
Pulse axis control	LS_Home_P	FB	Pulse axis return to original
	LS_MotionControl_P	FB	Pulse Axis Binding
	LS_ReadAxisPara_P	FB	Getting the pulse equivalent value of the pulse axis
	LS_ResetAxis_P	FB	Pulse axis reset

#### 6.1.2 LS\_Home\_P

The homing action of the EtherCAT bus motor, the specific homing process is determined by the homing mode of the bus motor design.

#### ⊙ Command Format

Instruction	Name	FB/FC	LD	ST	File
LS_Home_P	Pulse axis return to original instruction	FB		<pre>LS_Home_P( Axis:=, xExecute=, xDone=&gt;, xBusy:=&gt;, xCommandAborted:=&gt;, xError=&gt;, nErrorID=&gt;);</pre>	MC_HSI O

#### ⊙ Related Variables

##### I/O variable

I/O variable	Name	Data type	Range	initialization	Descriptive
Axis	High speed shaft	AXIS_REF_SM3	-	-	Specified axis

##### Input variable

Input variable	Name	Data type	Range	initialization	Descriptive
Execute	start	BOOL	TRUE-FALSE	FALSE	Rising edge trigger

##### Output variable

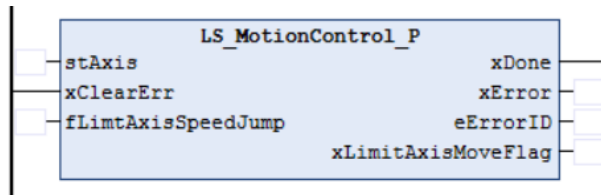
Output variable	Name	Data type	Range	initialization	Descriptive
xDone	Completion	BOOL	TRUE-FALSE	FALSE	TRUE: Homing is done.
xBusy	Executing	BOOL	TRUE-FALSE	FALSE	TRUE: In progress
xCommand Aborted	Command Interrupt	BOOL	TRUE-FALSE	FALSE	TRUE: Terminated by other commands
xError	Error	BOOL	TRUE-FALSE	FALSE	TRUE: An error occurred within the function block
diErrorID	Error ID	SMC_ERRO R	-	0	For the error code, please look for 'MC_HSIO.LS_BasicModule.LS_ERROR'.

	Boo lea n	Bit string				Integer							Real number		Moment, Duration, Date, String				
	BOOL	BYTE	WORD	DWORD	LWORD	USINT	UINT	UDINT	ULINT	SINT	INT	DINT	LINT	REAL	LREAL	TIME	DATE	TOD	DT

Axis	AXIS_REF_SM3																		
Execute	√	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
xDone	√	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
xBusy	√	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
xCommand Aboned	√	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
xError	√	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
diErrorID		SMC_ERROR																	

### ⊙ Functional Description

This function block needs to be used together with the LS\_MotionControl\_P function block, which only needs to be called in the programme, and the input and output pins do not need to be mapped to variables. Meanwhile, no matter how many high-speed pulse outputs are used in the programme, it only needs to be called once. As shown in the figure:



In the procedure to associate the axis that needs to return to the home position with the pulse axis channel number, MC500 series supports maximum 6 pulse axes, i.e. PulAxis\_0 to PulAxis\_5, as follows  
 'MC\_HSI0.LS\_MotionControl\_P.stAxis.PulAxis\_0 := ADR(LS\_Axis\_0);', where LS\_Axis\_0 is named according to the virtual drive in the project. Axis\_0;', where LS\_Axis\_0 is named according to the virtual driver in the project.

When switching xEnable from FALSE to TRUE, a rising edge signal is used to trigger the execution of the function block. xEnable's falling edge does not stop or affect the function block being executed.

When the specified axis is performing a home return action, reset xEnable will not stop the axis and the MC\_Stop function block is required.

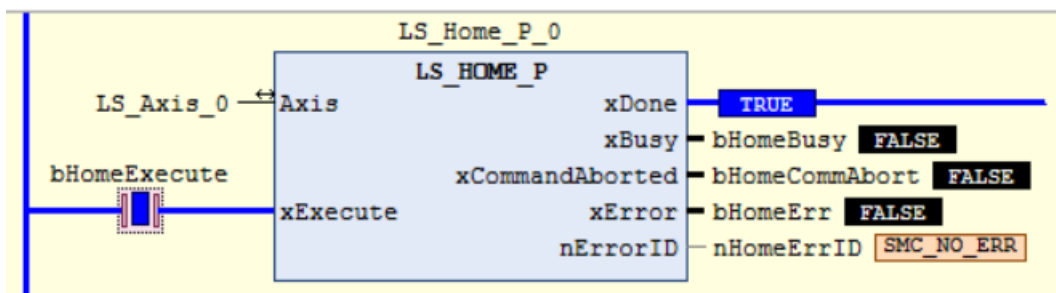
### ⊙ Program demo

ST:

```

LS_Home_P_0(Axis:= LS_Axis_0,
  xExecute TRUE := bHomeExecute TRUE,
  xDone FALSE => bHomeDone FALSE,
  xBusy TRUE => bHomeBusy TRUE,
  xCommandAborted FALSE => bHomeCommAbort FALSE,
  xError FALSE => bHomeErr FALSE,
  nErrorID SMC_NO_ERR => nHomeErrID SMC_NO_ERR);
  
```


LD:



### 6.1.3 LS\_MotionControl\_P

The homing action of the EtherCAT bus motor, the specific homing process is determined by the homing mode of the bus motor design.

#### ⊙ Command Format

Instruction	Name	FB/FC	LD	ST	File
LS_MotionControl_P	Pulse Axis Binding instruction	FB		<pre>LS_MotionControl_P( stAxis:=, xClearErr:=, fLimitAxisSpeedJump:=, xDone=&gt;, xError=&gt;, nErrorID=&gt;, xLimitAxisMoveFlag=&gt;);</pre>	MC_HS IO

#### ⊙ Related Variables

##### Input variable

Input variable	Name	Data type	Range	initialization	Descriptive
stAxis	High Speed Axis	DUT_Pulse_Axis	-	-	6 pulse planning axis structure, need to bind the specific pulse axis in the programme.
xClearErr	Active High	BOOL	TRUE-FALSE	FALSE	The rising edge is valid, the state is switched from FALSE to TRUE, and the function block is executed. Note: When using this pin, you must first call MC_Reset to clear the status of axis error reporting, and then call ClearErr to clear the error reporting within the module. (This pin is only used to clear the error flag and error code within the module).
fLimitAxisSpeedJump	Limit single-cycle speed jump at actual axis output	LREAL	1~500000	500000	When the speed jump is greater than this value, the module reports an error and limits the pulse output.

##### Output variable

Output variable	Name	Data type	Range	initialization	Descriptive
xDone	Completion	BOOL	TRUE-FALSE	FALSE	TRUE: Homing is done.
xError	Error	BOOL	TRUE-FALSE	FALSE	TRUE: An error occurred within the function block
eErrorID	Error ID	SMC_ERROR	-	0	For the error code, please look for 'MC_HSIO.LS_BasicModule.LS_ERROR'.
xLimitAxisMoveFlag	Restricted Axis Motion Symbol	BOOL	TRUE-FALSE	FALSE	Restricted Axis Motion Symbol

	Boolean	Bit string					Integer							Real number		Moment, Duration, Date, String				
	BOOL	BYTE	WORD	DWORD	LWORD	USINT	UINT	UDINT	ULINT	SINT	INT	DINT	LINT	REAL	LREAL	TIME	DATE	TOD	DT	STRING
stAxis		AXIS_REF_SM3																		
xClearErr	√	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
xDone	√	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
xLimitAxisMoveFlag	√	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
xError	√	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
eErrorID		SMC_ERROR																		

### ⊙ Functional Description

Used for pulse axis binding, and only needs to be called once.

After calling this function block, it is also necessary to associate the pulse axis with the actual output channel number in the program. MC500 series supports 6-way pulse axis at most, i.e. PulAxis\_0~PulAxis\_5, as follows 'MC\_HSIO.LS\_MotionControl\_P.stAxis.PulAxis\_0 . = ADR(LS\_Axis\_0);', where LS\_Axis\_0 is named according to the virtual driver in the project.

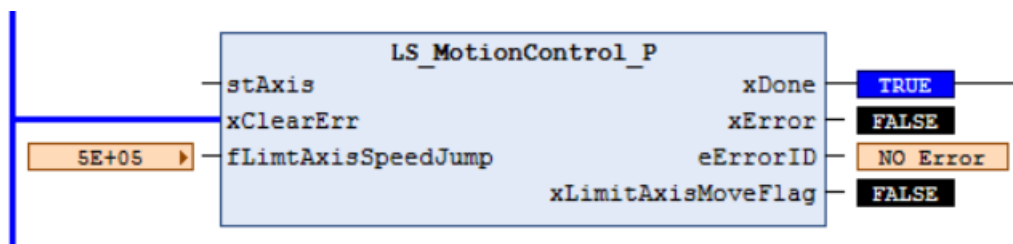
### ⊙ Program demo

ST:

```

MC_HSIO.LS_MotionControl_P(stAxis:= ,
    xClearErr:= ,
    fLimitAxisSpeedJump:= ,
    xDone=> ,
    xError=> ,
    eErrorID=> ,
    xLimitAxisMoveFlag=> );
    
```

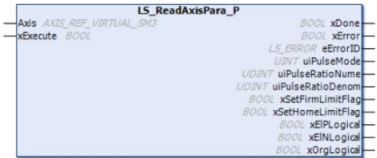
LD:



### 6.1.4 LS\_ReadAxisPara\_P

The homing action of the EtherCAT bus motor, the specific homing process is determined by the homing mode of the bus motor design.

#### ⊙ Command Format

Instruction	Name	FB/FC	LD	ST	File
LS_ReadAxisPara_P	Getting the pulse equivalent value of the pulse axis instruction	FB		<pre> LS_ReadAxisPara_P( Axis:=, xExecute:=, xDone=&gt;, xError=&gt;, eErrorID=&gt;, uiPulseMode=&gt;, uiPulseRatioNum=&gt;, uiPulseRatioDenom=&gt;, xSetFirmLimitFlag=&gt;, xSetHomeLimitFlag=&gt;, xEIPLogical=&gt;, xEINLogical=&gt;, xOrgLogical=&gt;); </pre>	MC_HSI O

#### ⊙ Related Variables

##### I/O variable

I/O variable	Name	Data type	Range	initialization	Descriptive
Axis	High speed shaft	AXIS_REF_SM3	-	-	Specified axis

##### Input variable

Input variable	Name	Data type	Range	initialization	Descriptive
xExecute	Active High	BOOL	TRUE-FALSE	FALSE	The rising edge is valid, the state switches from FALSE to TRUE, and the function block is executed.

##### Output variable

Output variable	Name	Data type	Range	initialization	Descriptive
xDone	Completion	BOOL	TRUE-FALSE	FALSE	TRUE: Homing is done.
xError	Error	BOOL	TRUE-FALSE	FALSE	TRUE: An error occurred within the function block
eErrorID	Error ID	SMC_ERROR	-	0	For the error code, please look for 'MC_HSI0.LS_BasicModule.LS_ERROR'.
uiPulseMode	Pulse shaft output mode	UINT	0-6	0	Pulse output modes of the axis: 0-3: pulse direction mode; 4-5: double pulse mode; 6: AB phase mode.
uiPulseRatioNum	Shaft Ratio Molecule	UDINT	Follow the data type	0	Shaft Ratio Molecule
uiPulseRatioDenom	Shaft Ratio Denominator	UDINT	Follow the data type	0	Shaft ratio denominator.
xSetFirmLimitFlag	Hardware limit valid or not	BOOL	TRUE-FALSE	FALSE	Whether the hardware limit set by the system is valid or not; FALSE indicates that the hardware limit is not processed, TRUE indicates that the hardware limit is processed.

xSetHomeLimitFlag	Return limit valid or not	BOOL	TRUE-FALSE	FALSE	Whether the return time limit is valid; Note: If SetFirmLimitFlag is set to TRUE, it is also valid for the return motion.
xEIPLogical	Positive limit logic level	BOOL	TRUE-FALSE	FALSE	Positive limit signal logic level, FALSE: normally open; TRUE: normally closed.
xEINLogical	Negative limit logic level	BOOL	TRUE-FALSE	FALSE	Negative limit signal logic level, FALSE: normally open; TRUE: normally closed.
xOrgLogical	Home signal logic level	BOOL	TRUE-FALSE	FALSE	Home signal logic level, FALSE: normally open; TRUE: normally closed.

	Boo lea n	Bit string					Integer							Real number		Moment, Duration, Date, String				
	BOOL	BYTE	WORD	DWORD	LWORD	USINT	UINT	UDINT	ULINT	SINT	INT	DINT	LINT	REAL	LREAL	TIME	DATE	TOD	DT	STRING
Axis		AXIS_REF_SM3																		
xExecute	√	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
xDone	√	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
xSetFirmLimitFlag	√	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
xError	√	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
eErrorID		SMC_ERROR																		

**⊙ Functional Description**

When setting xEnable from FALSE to TRUE, a rising edge signal is taken to trigger the function block.

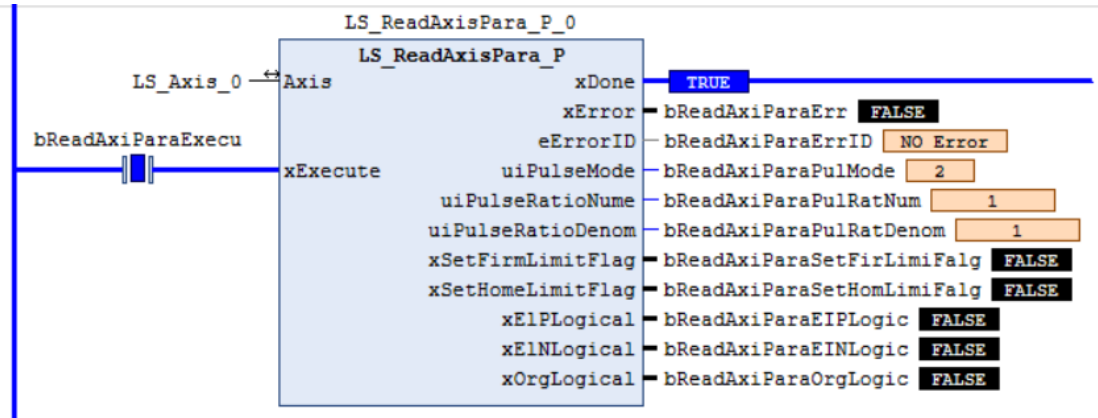
**⊙ Program demo**

ST:

```

LS_ReadAxisPara_P_0(Axis:= LS_Axis_0,
    xExecute TRUE := bReadAxiParaExecu TRUE,
    xDone TRUE => bReadAxiParaDone TRUE,
    xError FALSE => bReadAxiParaErr FALSE,
    eErrorID NO_Error => bReadAxiParaErrID NO_Error,
    uiPulseMode 2 => uiReadAxiParaPulMode 2,
    uiPulseRatioNum 1 => udReadAxiParaPulRatNum 1,
    uiPulseRatioDenom 1 => udReadAxiParaPulRatDenom 1,
    xSetFirmLimitFlag FALSE => bReadAxiParaSetFirLimiFalg FALSE,
    xSetHomeLimitFlag FALSE => bReadAxiParaSetHomLimiFalg FALSE,
    xEIPLogical FALSE => bReadAxiParaEIPLogic FALSE,
    xEINLogical FALSE => bReadAxiParaEINLogic FALSE,
    xOrgLogical FALSE => bReadAxiParaOrgLogic FALSE);


```

**LD:**


### 6.1.5 LS\_ResetAxis\_P

Pulse axis reset. Mainly realises pulse output axis error reset operation.

#### ⊙ Command Format

Instruction	Name	FB/FC	LD	ST	File
LS_ResetAxis_P	Pulse axis reset instruction	FB		LS_ResetAxis_P( Axis:=, xExecute:=, xDone=>, xBusy=>, xError=>, eErrorID=>);	MC_HSIO

#### ⊙ Related Variables

##### I/O variable

I/O variable	Name	Data type	Range	initialization	Descriptive
Axis	High speed shaft	AXIS_REF_SM3	-	-	Specified axis

##### Input variable

Input variable	Name	Data type	Range	initialization	Descriptive
xExecute	Active High	BOOL	TRUE-FALSE	FALSE	The rising edge is valid, the state switches from FALSE to TRUE, and the function block is executed.

##### Output variable

Output variable	Name	Data type	Range	initialization	Descriptive
xDone	Completion	BOOL	TRUE-FALSE	FALSE	TRUE: Homing is done.
xBusy	Enforcement marks	BOOL	TRUE-FALSE	FALSE	The function block has not yet finished executing and the new output value is being calculated
xError	Error	BOOL	TRUE-FALSE	FALSE	TRUE: An error occurred within the function block
eErrorID	Error ID	SMC_ERRO R	-	0	For the error code, please look for 'MC_HSIO.LS_BasicModule.LS_ERROR'.

	Boo lea n	Bit string					Integer							Real number		Moment, Duration, Date, String				
	BOOL	BYTE	WORD	DWORD	LWORD	USINT	UINT	UDINT	ULINT	SINT	INT	DINT	LINT	REAL	LREAL	TIME	DATE	TOD	DT	STRING
Axis		AXIS_REF_SM3																		
xExecute	√	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
xDone	√	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
xBusy	√	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
xError	√	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
eErrorID		SMC_ERROR																		

**⊙ Functional Description**

When setting xEnable from FALSE to TRUE, a rising edge signal is taken to trigger the function block.

**⊙ Program demo**

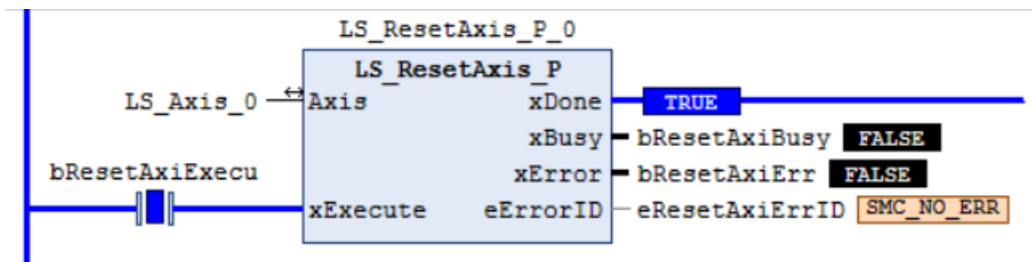
**ST:**

```

LS_ResetAxis_P_0(Axis:= LS_Axis_0,
  xExecute TRUE := bResetAxiExecu TRUE,
  xDone TRUE => bResetAxiDone TRUE,
  xBusy FALSE => bResetAxiBusy FALSE,
  xError FALSE => bResetAxiErr FALSE,
  eErrorID SMC_NO_ERR => eResetAxiErrID SMC_NO_ERR );

```

**LD:**



## 6.2 High-speed counter instruction

### 6.2.1 Instruction List

Instruction Type	Name	FB/FC	Functionality
High Speed Counter	LS_Counter	FB	High Speed Counter
	LS_PresetValue	FB	Preset count value

### 6.2.2 LS\_Counter

High Speed Counter Command. To get the setting parameters of the high-speed counter module as well as the count value, you need to configure the corresponding hardware port function and parameters in High Speed IO Module under Device first, otherwise the function is invalid.

#### ⊙ Command Format

Instruction	Name	FB/FC	LD	ST	File
LS_Counter	High Speed Counter instruction	FB		<pre>LS_Counter( xEnable:=, eChannel:=, xValid=&gt;, xError=&gt;, eErrorID=&gt;, diValue=&gt;);</pre>	MC_HSIO

#### ⊙ Related Variables

##### Input variable

Input variable	Name	Data type	Range	initialization	Descriptive
xExecute	Active High	BOOL	TRUE-FALSE	FALSE	TRUE: Function block active.
eChannel	Channel No.	LS_ENCODER_CHAN	0-8	0	Counter 0-5, encoder 6-8.

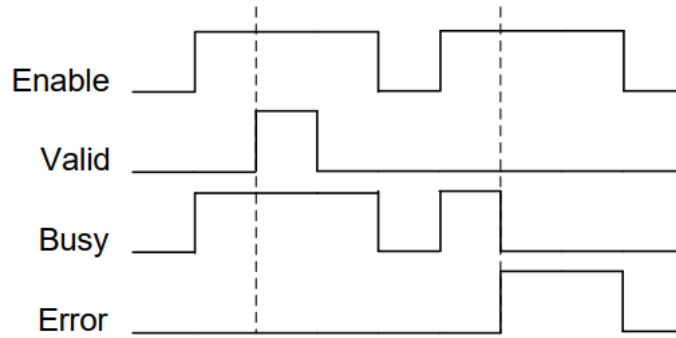
##### Output variable

Output variable	Name	Data type	Range	initialization	Descriptive
xValid	Implementation completed	BOOL	TRUE-FALSE	FALSE	TRUE if the axis is ready.
xError	Error	BOOL	TRUE-FALSE	FALSE	TRUE: An error occurred within the function block
eErrorID	Error ID	SMC_ERROR	-	0	For the error code, please look for 'MC_HSIO.LS_BasicModule.LS_ERROR'.
diValue	Current value	DINT	Follow the data type	0	The current count value of the selected channel.

	Boolean	Bit string					Integer							Real number		Moment, Duration, Date, String				
	BOOL	BYTE	WORD	DWORD	LWORD	USINT	UINT	UDINT	ULINT	SINT	INT	DINT	LINT	REAL	LREAL	TIME	DATE	TOD	DT	STRING
eChannel		LS_ENCODER_CHAN																		
xExecute	√	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
xValid	√	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
xError	√	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
eErrorID		SMC_ERROR																		

**⊙ Functional Description**

It mainly realises the functions of counter enable, counting, frequency measurement and counting direction output.

**⊙ Timing Diagram: LS\_Counter****⊙ Program demo**


**ST:**

**LD:**

### 6.2.3 LS\_PresetValue

Preset count value command. You need to configure the corresponding hardware port function and parameters in High Speed IO Module under Device first, otherwise this function is invalid.

#### ⊙ Command Format

Instruction	Name	FB/FC	LD	ST	File
LS_PresetValue	Preset count value instruction	FB		<pre>LS_PresetValue( xExecute:=, eChannel:=, eTriggerEdge:=, diPresetValue:=, xDone=&gt;, xBusy=&gt;, xError=&gt;, eErrorID=&gt;);</pre>	MC_HSIO

#### ⊙ Related Variables

##### Input variable

Input variable	Name	Data type	Range	initialization	Descriptive
xExecute	Active High	BOOL	TRUE-FALSE	FALSE	TRUE: Function block active.
eChannel	Channel No.	LS_ENCODER_CHAN	0-8	0	Counter 0-5, encoder 6-8.
eTriggerEdge	Manufacturer Parameters	LS_PRESET_EDGE_TYPE	-	-	Not supported at this time.
diPresetValue	Counter Preset	DINT	Follow the data type	0	Counter preset.

##### Output variable

Output variable	Name	Data type	Range	initialization	Descriptive
xDone	Completion	BOOL	TRUE-FALSE	FALSE	TRUE: Homing is done.
xBusy	Enforcement marks	BOOL	TRUE-FALSE	FALSE	The function block has not yet finished executing and the new output value is being calculated
eErrorID	Error ID	SMC_ERROR	-	0	For the error code, please look for 'MC_HSIO.LS_BasicModule.LS_ERROR'.
diValue	Current value	DINT	Follow the data type	0	The current count value of the selected channel.

	Boolean	Bit string					Integer							Real number		Moment, Duration, Date, String				
	BOOL	BYTE	WORD	DWORD	LWORD	USINT	UINT	UDINT	ULINT	SINT	INT	DINT	LINT	REAL	LREAL	TIME	DATE	TOD	DT	STRING
eChannel		LS_ENCODER_CHAN																		
xExecute	√	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
xValid	√	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
xError	√	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
eErrorID		SMC_ERROR																		

**⊙ Functional Description**

This function block is a rising edge single trigger valid.

**⊙ Program demo**

**ST:**

**LD:**

## 6.3 LC1000 Local High Speed Counter

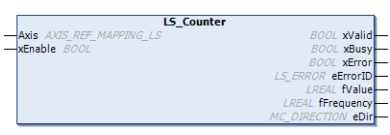
### 6.3.1 Instruction List

Instruction Type	Name	FB/FC	Functionality
LC1000 Local High Speed Counter	LS_Counter	FB	High Speed Counter Instruction
	LS_PresetValue	FB	Preset Count Instruction
	LS_Compare	FB	Single-point compare output
	LS_CompareStep	FB	Equidistant compare output
	LS_CompareFIFO	FB	Queue compare output
	LS_PWM	FB	PWM output
	LS_EnableInterrupt	FB	Interrupt enable

### 6.3.2 LS\_Counter

High-speed counter enable.

#### ⊙ Command Format

Instruction	Name	FB/FC	LD	ST	File
LS_Counter	High Speed Counter instruction	FB		<pre>LS_Counter( Axis:= , xEnable:= , xValid=&gt; , xBusy=&gt; , xError=&gt; , eErrorID=&gt; , fValue=&gt; , fFrequency=&gt; , eDir=&gt; );</pre>	MC_HSIO

#### ⊙ Related Variables

##### I/O variable

I/O variable	Name	Data type	Range	initialization	Descriptive
Axis	Encoder shaft	AXIS_REF_MAPPING_LS	-	-	AXIS_REF_MAPPING_LS type encoder axis example

##### Input variable

Input variable	Name	Data type	Range	initialization	Descriptive
xExecute	Active High	BOOL	TRUE-FALSE	FALSE	TRUE: Function block active.

##### Output variable

Output variable	Name	Data type	Range	initialization	Descriptive
xValid	Implementation completed	BOOL	TRUE-FALSE	FALSE	TRUE if the axis is ready.
xBusy	Executing	BOOL	TRUE-FALSE	FALSE	TRUE: In progress
xError	Error	BOOL	TRUE-FALSE	FALSE	TRUE: An error occurred within the function block
eErrorID	Error ID	SMC_ERROR	-	0	For the error code, please look for 'MC_HSIO.LS_BasicModule.LS_ERROR'.
fValue	Count value	LREAL	Follow data types	0.0	Current count value, converted from scaling, in units of unit
fFrequency	Counting Frequency	LREAL	Follow data types	0.0	Current count frequency, unit unit/s

eDir	Direction	LS_DIR	[-1,1]	0	Current count direction, in unit/s -1:Negative direction 1:Positive direction
------	-----------	--------	--------	---	---

	Boolean	Bit string					Integer							Real number		Moment, Duration, Date, String				
	BOOL	BYTE	WORD	DWORD	LWORD	USINT	UINT	UDINT	ULINT	SINT	INT	DINT	LINT	REAL	LREAL	TIME	DATE	TOD	DT	STRING
xExecute	√	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
xValid	√	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
xBusy	√	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
xError	√	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
eErrorID	SMC_ERROR																			

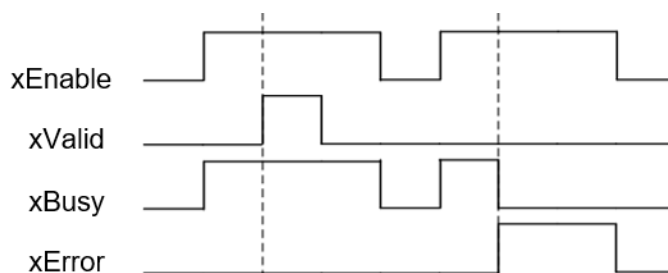
### ⊙ Functional Description

It mainly realises the local counter start/stop, after detecting xEnable set to TRUE, the counter responds to the external pulse signal and starts counting, after detecting xEnable set to FALSE, the counter stops counting. The counter position changes within the range of counter mode and the unit of position is Unit.

The counting direction should be changed through the software interface. The counting direction of different counting modes is defined as follows. After changing the direction, you need to Counting direction change, you need to modify through the software interface, the counting direction of different counting modes is defined as follows. To change the counting direction, you need to re-enable the counter function block.

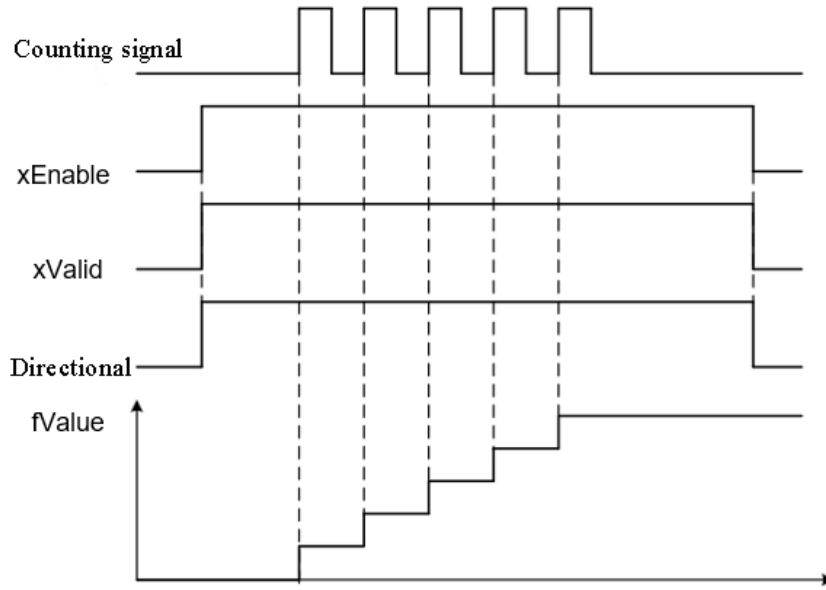
Direction	Phase AB	Pulse + Direction	CW/CCW	CW/CCW Single phase counting
positive direction	A Phase ahead B Phase increasing count B phase overrun A phase decrease count	Direction signal low level minus count Direction signal high level increment count	A Incremental counting B Decreasing Count	Incremental
negative	A phase overruns B phase underruns B phase overrun A phase increase count	Direction signal low level increase count Direction signal high level decrement count	A Decremental counting B Incremental counting	decimation

### ⊙ Timing Diagram: LS\_Counter



⊙ **Program demo**

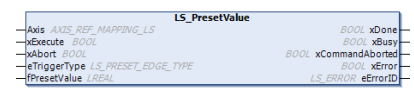
Take the pulse + method of forward counting as an example, the counter increments as shown in the figure below.



### 6.3.3 LS\_PresetValue

High-speed counter preset value.

#### ⊙ Command Format

Instruction	Name	FB/FC	LD	ST	File
LS_PresetValue	Preset count value instruction	FB		<pre> LS_PresetValue( Axis:= , xExecute:= , xAbort:= , eTriggerType:= , fPresetValue:= , xDone=&gt; , xBusy=&gt; , xCommandAborted=&gt; , xError=&gt; , eErrorID=&gt; ); </pre>	MC_HSIO

#### ⊙ Related Variables

##### I/O variable

I/O variable	Name	Data type	Range	initialization	Descriptive
Axis	Encoder shaft	AXIS_REF_MAPPING_LS	-	-	AXIS_REF_MAPPING_LS type encoder axis example

##### Input variable

Input variable	Name	Data type	Range	initialization	Descriptive
xExecute	Active High	BOOL	TRUE-FALSE	FALSE	TRUE: Function block active.
xAbort	Abort	BOOL	TRUE FALSE	FALSE	Level trigger TRUE: Abort function block
eTriggerType	Trigger mode	LS_PRESET_EDGE_TYPE	[0,3]	0	0: Command rising edge trigger 1: External DI rising edge trigger 2: External DI falling edge trigger 3: External DI rising or falling edge trigger
fPresetValue	Preset value	LREAL	Follow data types	0.0	Counter preset value, unit: unit, conversion to Pulse unit need to meet-2147483648~ 2147483647 range

##### Output variable

Output variable	Name	Data type	Range	initialization	Descriptive
xDone	Completion	BOOL	TRUE-FALSE	FALSE	TRUE: Homing is done.
xBusy	Enforcement marks	BOOL	TRUE-FALSE	FALSE	The function block has not yet finished executing and the new output value is being calculated
xCommandAborted	Stop sign	BOOL	TRUE FALSE	FALSE	TRUE: function block execution is interrupted
xError	Error	BOOL	TRUE-FALSE	FALSE	TRUE: An error occurred within the function block
eErrorID	Error ID	SMC_ERROR	-	0	For the error code, please look for 'MC_HSIO.LS_BasicModule.LS_ERROR'.

	Boolean	Bit string					Integer							Real number		Moment, Duration, Date, String				
	BOOL	BYTE	WORD	DWORD	LWORD	USINT	UINT	UDINT	ULINT	SINT	INT	DINT	LINT	REAL	LREAL	TIME	DATE	TOD	DT	STRING
xExecute	√	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
xValid	√	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
xError	√	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
eErrorID	SMC_ERROR																			

### ⊙ Functional Description

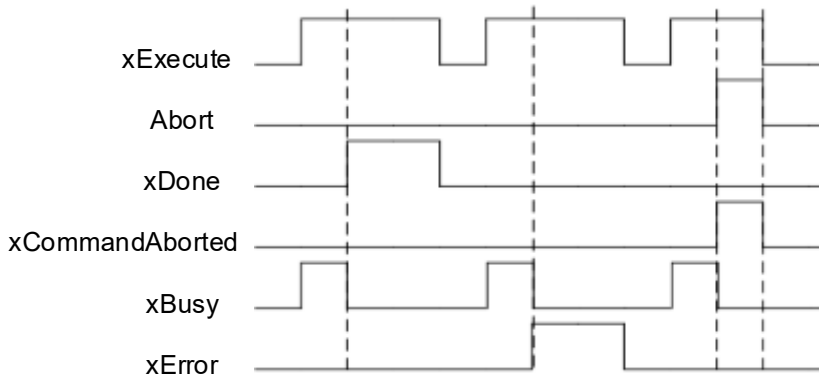
This function block mainly implements the preset function of the counter, the rising edge triggers the function block to execute, and outputs the xDone signal after the preset is completed.

There are 4 types of preset trigger types eTriggerType, which can be triggered by rising edge of instruction, or external DI input (rising edge, falling edge, arbitrary edge).

When the preset condition selects command rising edge trigger, xExecute is the command trigger pin, and the rising edge trigger is selected.

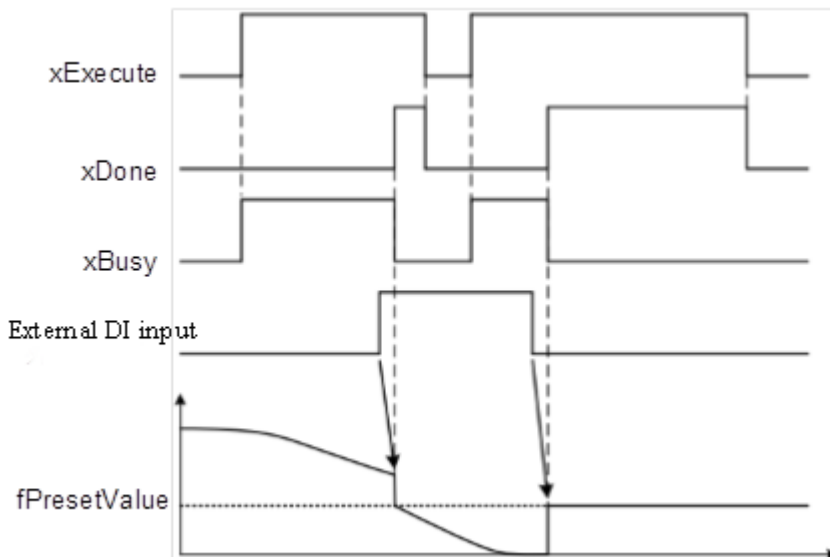
When the preset condition selects external DI input trigger, it is necessary to select preset inputs IN0~IN7 in the counter parameter configuration interface, and when external DI trigger is selected, xExecute is the level enable signal.

### ⊙ Timing Diagram: LS\_Counter



### ⊙ Program demo

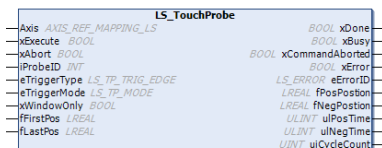
With eTriggerType = 3 (external DI arbitrary edge), the instruction timing is as follows.



### 6.3.4 LS\_TouchProbe

High-speed counter probes.

⊙ **Command Format**

Instruction	Name	FB/FC	LD	ST	File
LS_TouchProbe	Single-point compare output instruction	FB		<pre> LS_TouchProbe( Axis:= , xExecute:= , xAbort:= , iProbeID:= , eTriggerType:= , eTriggerMode:= , xWindowOnly:= , fFirstPos:= , fLastPos:= , xDone=&gt; , xBusy=&gt; , xCommandAborted=&gt; , xError=&gt; , eErrorID=&gt; , fPosPosition=&gt; , fNegPosition=&gt; , ulPosTime=&gt; , ulNegTime=&gt; , uiCycleCount=&gt; ); </pre>	MC_HS IO

⊙ **Related Variables**

**I/O variable**

I/O variable	Name	Data type	Range	initialization	Descriptive
Axis	Encoder shaft	AXIS_REF_MAPPING_LS	-	-	AXIS_REF_MAPPING_LS type encoder axis example

**Input variable**

Input variable	Name	Data type	Range	initialization	Descriptive
xExecute	Active High	BOOL	TRUE-FALSE	FALSE	TRUE: Function block active.
xAbort	Abort	BOOL	TRUE FALSE	FALSE	Level trigger TRUE: Abort function block
iProbeID	Probe ID	INT	[0,1]	0	Probe ID No.
eTriggerType	Trigger mode	LS_PRESET_EDGE_TYPE	[0,3]	0	0: Command rising edge trigger 1: External DI rising edge trigger 2: External DI falling edge trigger 3: External DI rising or falling edge trigger
eTriggerMode	Trigger Mode	LS_TP_MODE	[0,1]	0	Trigger mode: 0: Single latch 1: Continuous latch
xWindowOnly	Window Effective	BOOL	TRUE FALSE	FALSE	Latching window takes effect when True, only the position inside the window is accepted
fFirstPos	Start position	LREAL	Follow data types	0.0	Specify the position to start receiving triggers, unit unit
fLastPos	End position	LREAL	Follow data types	0.0	Specify the position to stop receiving triggers, unit.

**Output variable**

Output variable	Name	Data type	Range	initialization	Descriptive
xDone	Completion	BOOL	TRUE-FALSE	FALSE	TRUE: Homing is done.
xBusy	Enforcement marks	BOOL	TRUE-FALSE	FALSE	The function block has not yet finished executing and the new output value is being calculated
xCommandAborted	Stop sign	BOOL	TRUE FALSE	FALSE	TRUE: function block execution is interrupted
xError	Error	BOOL	TRUE-FALSE	FALSE	TRUE: An error occurred within the function block
eErrorID	Error ID	SMC_ERRO R	-	0	For the error code, please look for 'MC_HSIO.LS_BasicModule.LS_ERROR'.
fPosPostion	Rising edge latch position	LREAL	Follow data types	0.0	Rising edge latch position, unit
fNegPosition	Falling edge latch position	LREAL	Follow data types	0.0	Falling edge latch position, unit
ulPosTime	Rising edge latch time	ULINT	[0,2^63)	0	Rising edge latch time (in ns)
ulNegTime	Falling edge latch time	ULINT	[0,2^63)	0	Falling edge latch time (unit ns)
uiCycleCount	Latch count	UINT	[0,2^16)	0	Continuous latch mode, accumulate once for latch completion

	Boo lea n	Bit string				Integer							Real number		Moment, Duration, Date, String					
	BOOL	BYTE	WORD	DWORD	LWORD	USINT	UINT	UDINT	ULINT	SINT	INT	DINT	LINT	REAL	LREAL	TIME	DATE	TOD	DT	STRING
xExecute	√	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
xValid	√	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
xError	√	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
eErrorID	SMC_ERROR																			

**© Functional Description**

This function block mainly implements the function of latching the counter value at the moment of external DI, and also supports the function of latching the internal timestamp of the local high-speed IO module, the rising edge triggers the execution of the function block, and outputs the xDone signal after the latching is completed.

The output value of the latched position will remain unchanged until the counter position is latched again in the same instance of the LS\_TouchProbe instruction.

Each counter axis supports two probes, software parameter configuration interface, select probe input port DI0~DI7. Each probe supports DI rising edge, falling edge and double edge latch.

Each probe supports single latch and continuous latch.

Supports window valid setting. When xWindowOnly is specified as TRUE (valid), only the triggers within the range of fFirstPos (start position) and fLastPos (end position) are detected at the counter position.

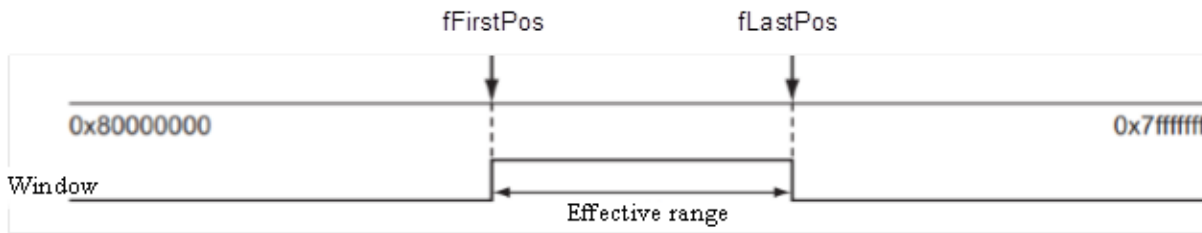
The counter can be selected from linear mode and rotary mode in the parameter configuration screen.

**⊙ Linear pattern:**

The effective range of the window is expressed by the following formula.  $fFirstPos$  (Start position)  $\cong$  Window range  $\cong$   $fLastPos$  (End position)

When  $fFirstPos$  (start position)  $>$   $fLastPos$  (end position) is specified, an exception occurs.

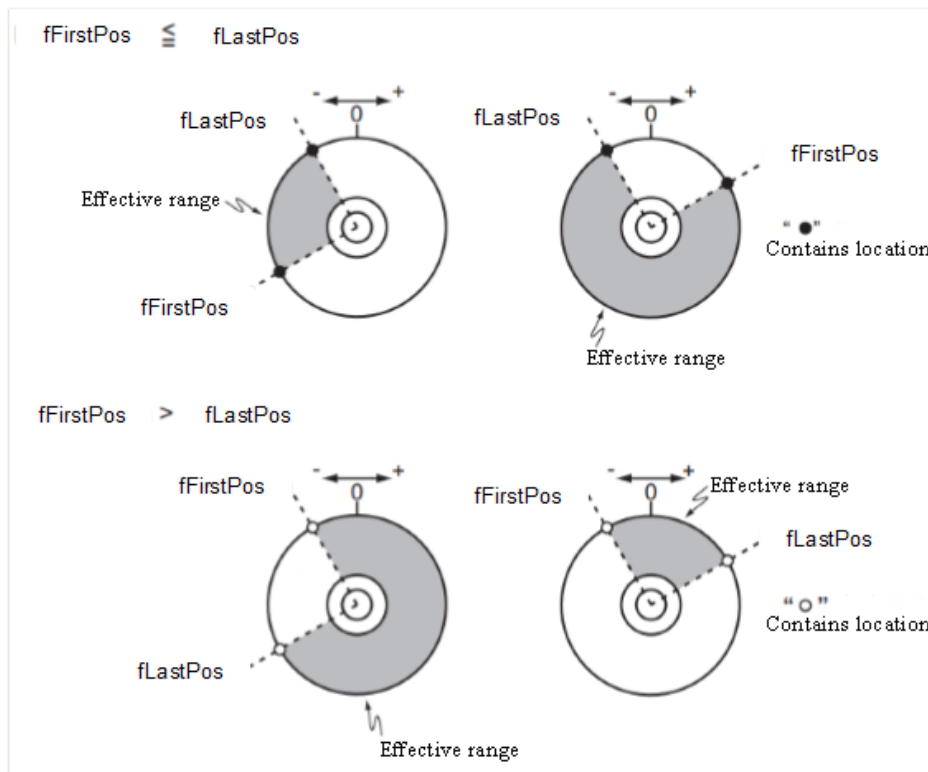
An exception also occurs when the position range is specified to be outside the [Linear Mode] range.

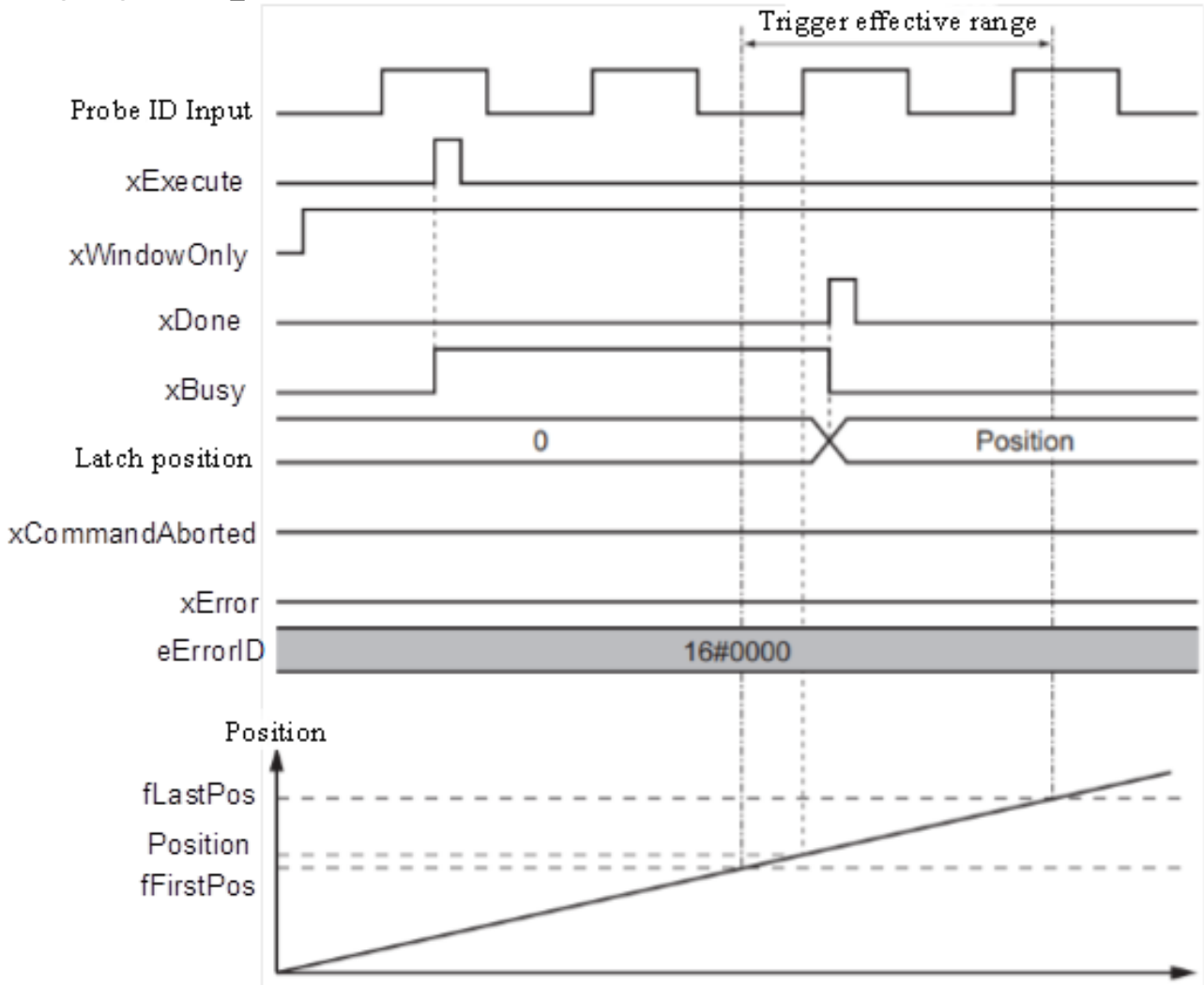

**⊙ Rotary mode:**

Can be specified as  $fFirstPos$  (start position)  $\cong$   $fLastPos$  (end position) and  $fFirstPos$  (start position)  $>$   $fLastPos$  (end position).

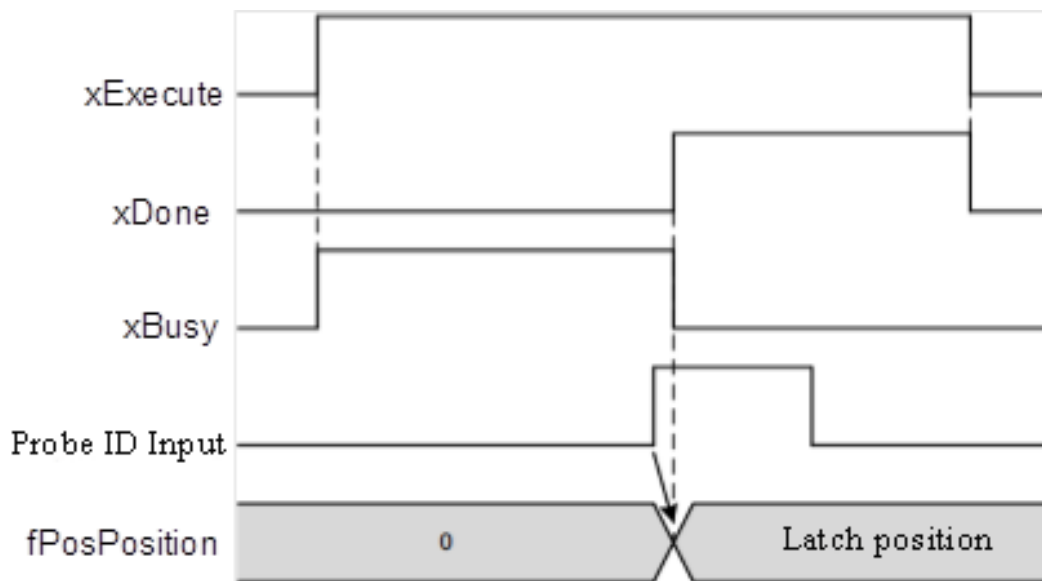
When  $fFirstPos$  (start position)  $>$   $fLastPos$  (end position), it will span the upper and lower limit positions of the loop counter.

If the specified position is outside the upper and lower limit of the loop counter, an exception occurs.

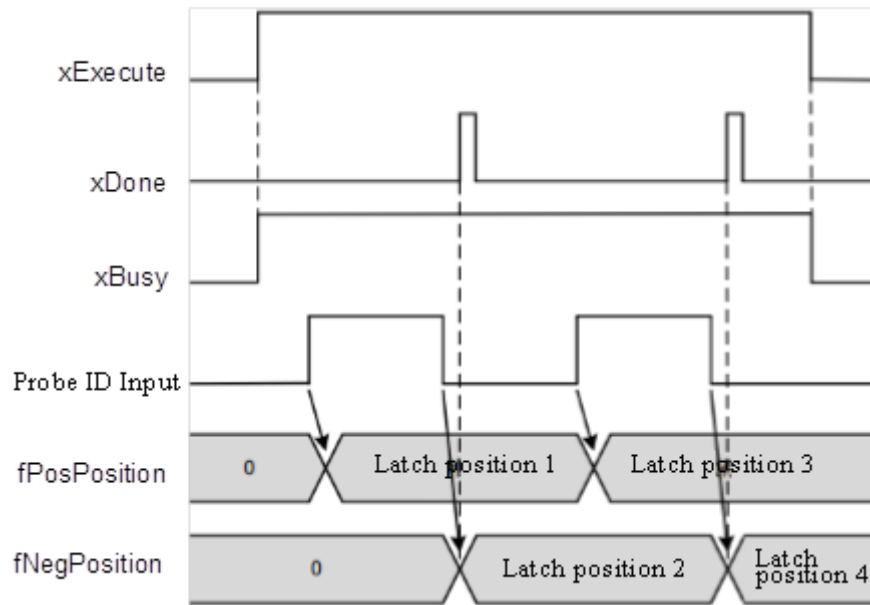


**⊙ Timing Diagram: LS\_Counter**

**⊙ Program demo**

External DI rising edge trigger (eTriggerType = 0), single trigger mode (eTriggerMode = 0), the instruction timing diagram is shown below.



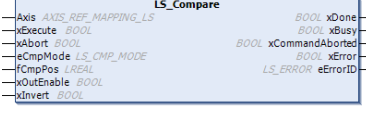
External DI double edge trigger (eTriggerType =2), single trigger mode (eTriggerMode =0), first rising edge, then falling edge, the instruction timing diagram is shown below.



### 6.3.5 LS\_Compare

High-speed one-dimensional single-point comparisons.

#### ⊙ Command Format

Instruction	Name	FB/FC	LD	ST	File
LS_Compar	Single-point compare output instruction	FB		<pre> LS_Compare( Axis:= , xExecute:= , xAbort:= , fCmpPos:= , xOutEnable:= , eOutType:= , udiOutPara:= , xDone=&gt; , xBusy=&gt; , xCommandAborted=&gt; , xError=&gt; , eErrorID=&gt; ); </pre>	MC_HS IO

#### ⊙ Related Variables

##### I/O variable

I/O variable	Name	Data type	Range	initialization	Descriptive
Axis	Encoder shaft	AXIS_REF_MAPPING_LS	-	-	AXIS_REF_MAPPING_LS type encoder axis example

##### Input variable

Input variable	Name	Data type	Range	initialization	Descriptive
xExecute	Active High	BOOL	TRUE-FALSE	FALSE	TRUE: Function block active.
xAbort	Abort	BOOL	TRUE FALSE	FALSE	Level trigger TRUE: Abort function block
fCmpPos	fCmpPos	LREAL	Follow data types	0.0	Comparison value, unit: unit
xOutEnable	xOutEnable	BOOL	TRUE FALSE	FALSE	Enable hardware port output function. When using this function, you need to configure the corresponding output port in the parameter configuration interface. When this function is not enabled, the function block executes normally, only the hardware port has no output.
eOutType	eOutType	LS_CMP_OUT_TPYE	[0,3]	0	0: Time mode - output FALSE 1: Time mode - output TRUE 2: Pulse mode - output FALSE 3: Pulse mode - output TRUE
udiOutPara	udiOutPara	UDINT	Follow data types	0	① Time mode: Hold time of open output port, us unit, [1,20000000]. ② Pulse mode: distance value, pulse unit, pulse value range [1~2147483647].

##### Output variable

Output variable	Name	Data type	Range	initialization	Descriptive
xDone	Completion	BOOL	TRUE-FALSE	FALSE	TRUE: Homing is done.
xBusy	Enforcement	BOOL	TRUE-FALSE	FALSE	The function block has not

	marks				yet finished executing and the new output value is being calculated
xCommandAborted	Stop sign	BOOL	TRUE FALSE	FALSE	TRUE: function block execution is interrupted
xError	Error	BOOL	TRUE-FALSE	FALSE	TRUE: An error occurred within the function block
eErrorID	Error ID	SMC_ERROR	-	0	For the error code, please look for 'MC_HSIO.LS_BasicModule.LS_ERROR'.

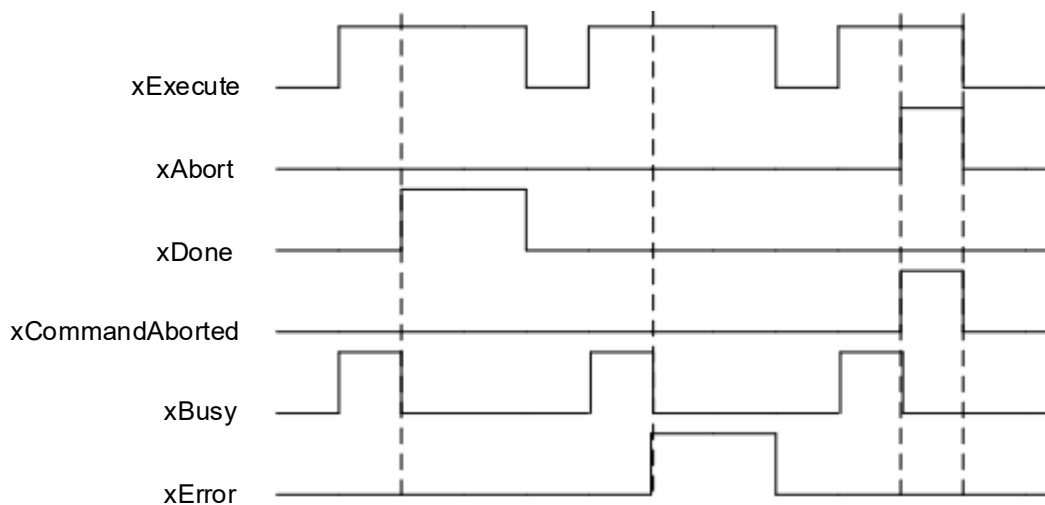
	Boolean	Bit string					Integer							Real number		Moment, Duration, Date, String				
	BOOL	BYTE	WORD	DWORD	LWORD	USINT	UINT	UDINT	ULINT	SINT	INT	DINT	LINT	REAL	LREAL	TIME	DATE	TOD	DT	STRING
xExecute	√	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
xValid	√	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
xError	√	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
eErrorID	SMC_ERROR																			

### ⊙ Functional Description

This function block implements a high-speed output port to be triggered when the current count value of the counter is equal to the set comparison value, and keep it for a period of time or a distance. Comparison and consistency output function is a single trigger, when xDone=TRUE, the comparison and consistency function is finished, if you need to continue to use the comparison function, trigger xExecute again.

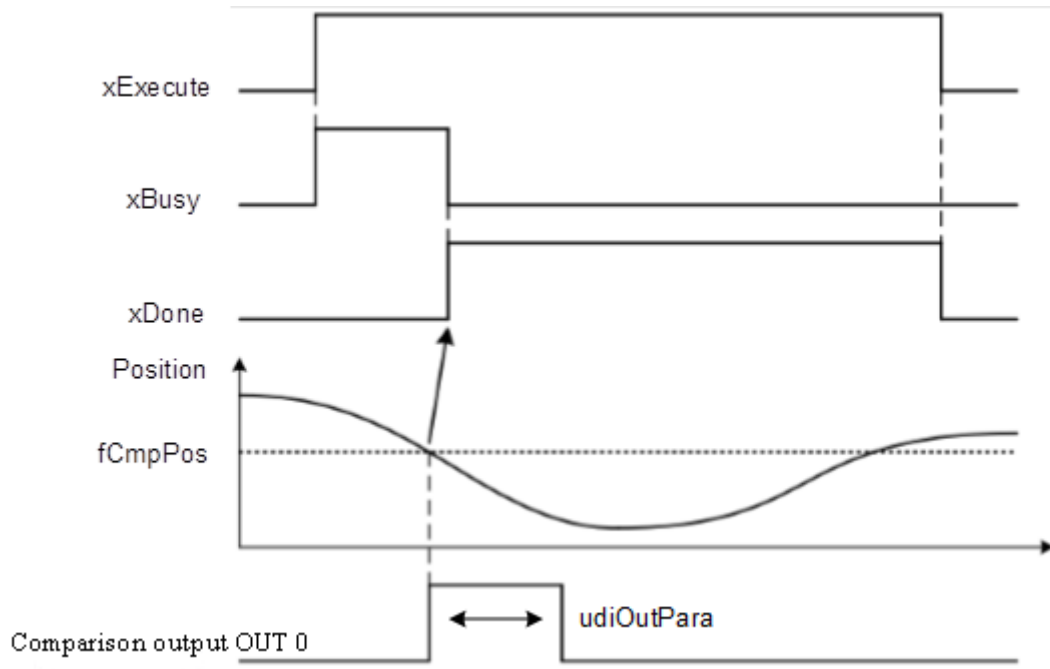
Before using the compare function block, the output port must be configured in the corresponding counter parameter interface, and the output port can be configured OUT0~OUT7.

### ⊙ Timing Diagram:



**⊙ Program demo**

The counter interface is configured with the comparison output port OUT0. The function block enables the comparison output port ( $xOutEnable = TRUE$ ), the comparison position ( $fCmpPos = 1000.0$ , assuming a 1:1 gear ratio and  $1unit = 1pulse$ ), the output type Pulse mode - Output TRUE ( $eOutType = 3$ ), the output hold parameter ( $udiOutPara = 100$ ), the OUT0 output port conducts when the count value reaches the comparison position 1000, and the output port closes after 100 more pulses are accumulated.



**Note**

When the high-speed compare output function block is activated, the initial state of the output OUT point is automatically flipped to the opposite state of the first compare output point.

### 6.3.6 LS\_CompareStep

High-speed one-dimensional single-point comparisons.

#### ⊙ Command Format

Instruction	Name	FB/FC	LD	ST	File
LS_CompareStep	Equidistant compare output instruction	FB		<pre> LS_CompareStep( Axis:= , xExecute:= , xAbort:= , fStartPos:= , fStep:= , uiCmpNum:= , xOutEnable:= , eOutType:= , udiOutPara:= , xDone=&gt; , xBusy=&gt; , xCommandAborted=&gt; , xError=&gt; , eErrorID=&gt; , uiPoints=&gt; , fCurrentPos=&gt; ); </pre>	MC_HS IO

#### ⊙ Related Variables

##### I/O variable

I/O variable	Name	Data type	Range	initialization	Descriptive
Axis	Encoder shaft	AXIS_REF_MAPPING_LS	-	-	AXIS_REF_MAPPING_LS type encoder axis example

##### Input variable

Input variable	Name	Data type	Range	initialization	Descriptive
xExecute	Active High	BOOL	TRUE-FALSE	FALSE	TRUE: Function block active.
xAbort	Abort	BOOL	TRUE FALSE	FALSE	Level trigger TRUE: Abort function block
fStartPos	Start position	LREAL	Follow the data range	0.0	Start comparison value, unit
fStep	Step length	LREAL	Follow the data range	0.0	Comparison isometric step, unit, positive position increment, negative position decrement
uiCmpNum	Number of comparison points	UINT	[1,65535]	0	Number of comparison points
xOutEnable	xOutEnable	BOOL	TRUE FALSE	FALSE	Enable hardware port output function. When using this function, you need to configure the corresponding output port in the parameter configuration interface. When this function is not enabled, the function block executes normally, only the hardware port has no output.
eOutType	eOutType	LS_CMP_OUTTYPE	[0,3]	0	0: Time mode - output FALSE 1: Time mode - output TRUE 2: Pulse mode - output FALSE 3: Pulse mode - output TRUE
udiOutPara	udiOutPara	UDINT	Follow data types	0	① Time mode: Hold time of

					open output port, us unit, [1,20000000]. ② Pulse mode: distance value, pulse unit, pulse value range [1~2147483647].
--	--	--	--	--	---

**Output variable**

Output variable	Name	Data type	Range	initialization	Descriptive
xDone	Completion	BOOL	TRUE-FALSE	FALSE	TRUE: Homing is done.
xBusy	Enforcement marks	BOOL	TRUE-FALSE	FALSE	The function block has not yet finished executing and the new output value is being calculated
xCommandAborted	Stop sign	BOOL	TRUE-FALSE	FALSE	TRUE: function block execution is interrupted
xError	Error	BOOL	TRUE-FALSE	FALSE	TRUE: An error occurred within the function block
eErrorID	Error ID	SMC_ERRO R	-	0	For the error code, please look for 'MC_HSIO.LS_BasicModule.LS_ERROR'.
uiPoints	Compare Points	UINT	[0,65535]	0	Comparison Points
fCurrentPos	Current Comparison Position	LREAL	Follow data types	0.0	Current comparison position, unit

	Boo lea n	Bit string					Integer							Real number		Moment, Duration, Date, String				
	BOOL	BYTE	WORD	DWORD	LWORD	USINT	UINT	UDINT	ULINT	SINT	INT	DINT	LINT	REAL	LREAL	TIME	DATE	TOD	DT	STRING
xExecute	√	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
xValid	√	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
xError	√	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
eErrorID	SMC_ERROR																			

**© Functional Description**

This function block implements a hardware high-speed output port to be triggered when the current count value of the counter is equal to the set comparison value and held for a period of time or a distance. The comparison value is incremented or decremented according to the step size. Assuming that the initial value of comparison is 5, the comparison step size is 2, and the number of comparison points is 5, then the comparison values are 5,7,9,11,13. Before using the compare function block, the output port must be configured on the corresponding counter interface, and the output port can be configured OUT0~OUT7.

The compare output hold parameter supports two ways. Time mode: When the count value is equal to the set comparison value, the comparison output OUT point will be turned on, and the output OUT point will be turned off after holding udiOutPara \*1us time. Pulse mode: when the count value is equal to the set comparison value, turn on the compare output OUT point, hold udiOutPara \*1pulse count number, then turn off the output OUT point.

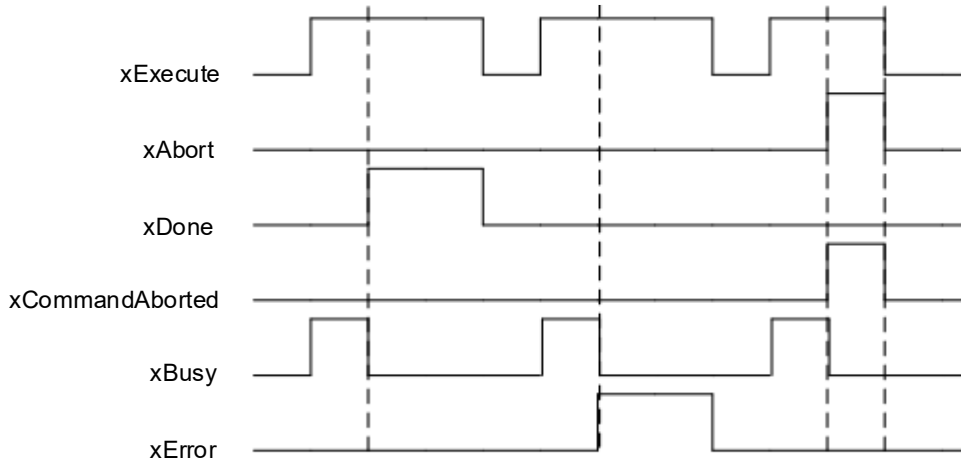
xOutEnable is to enable the output function of hardware port. when FALSE, the function block executes normally and there is no output from hardware OUT point. when TRUE, the function block executes normally and there is normal output from hardware OUT point.

If the interval between two adjacent triggers is less than the compare output hold time, the output port will continuously hold the output state, and the current output time will directly overwrite the previous output time.

During the comparison process, trigger xExecute rising edge again, the function block will re-execute the compare

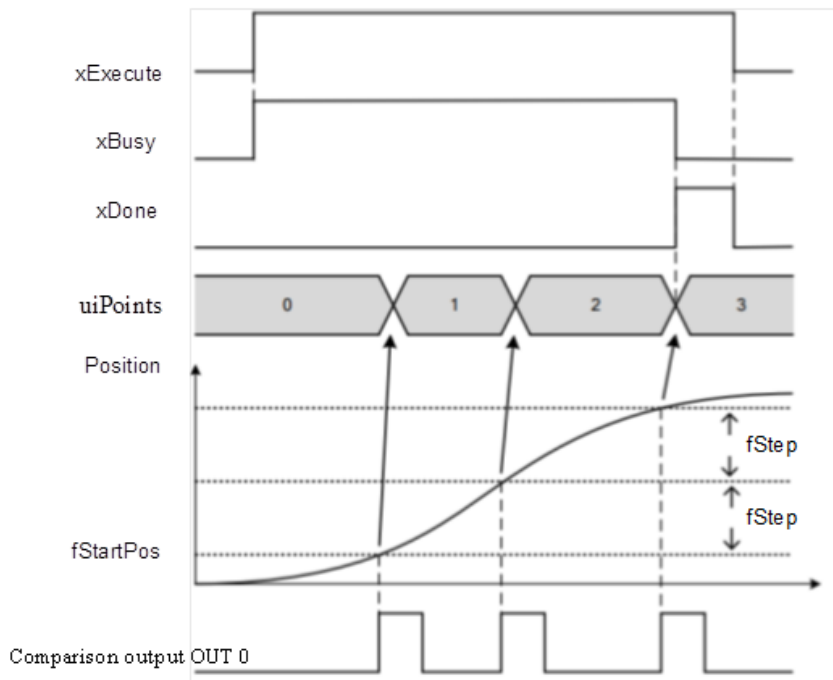
output function from the first point.

⊙ **Timing Diagram:**



⊙ **Program demo**

When fStep is positive, the comparison position is incremented, and the instruction timing diagram is shown below. fStep is negative, similarly, the comparison position is decremented.




**Note**

When the high-speed compare output function block is activated, the initial state of the output OUT point is automatically flipped to the opposite state of the first compare output point.

### 6.3.7 LS\_CompareFIFO

High-speed counter queue comparison output.

#### ⊙ Command Format

Instruction	Name	FB/FC	LD	ST	File
LS_CompareFIFO	Queue compare output instruction	FB		<pre> LS_CompareFIFO( Axis:= , xExecute:= , xAbort:= , uiCmpNum:= , xPush:= , afCmpPos:= , xOutEnable:= , eOutType:= , udiOutPara:= , axInvert:= , xDone=&gt; , xBusy=&gt; , xCommandAborted=&gt; , xError=&gt; , eErrorID=&gt; , xPushDone=&gt; , uiFIFOSize=&gt; , uiPoints=&gt; , fCurrentPos=&gt; ); </pre>	MC_HS IO

#### ⊙ Related Variables

##### I/O variable

I/O variable	Name	Data type	Range	initialization	Descriptive
Axis	Encoder shaft	AXIS_REF_MAPPING_LS	-	-	AXIS_REF_MAPPING_LS type encoder axis example

##### Input variable

Input variable	Name	Data type	Range	initialization	Descriptive
xExecute	Active High	BOOL	TRUE-FALSE	FALSE	TRUE: Function block active.
xAbort	Abort	BOOL	TRUE FALSE	FALSE	Level trigger TRUE: Abort function block
uiCmpNum	Number of comparison points	UINT	[1,65535]	0	Number of comparison points
xPush	FIFO cache point press-in	BOOL	TRUE FALSE	FALSE	Press point into FIFO, support dynamic press in.
afCmpPos	Compare Point Location	ARRAY[0..999] OF LREAL	Follow data types	0.0	Comparison point location set, unit unit, maximum 1000 points
xOutEnable	xOutEnable	BOOL	TRUE FALSE	FALSE	Enable hardware port output function. When using this function, you need to configure the corresponding output port in the parameter configuration interface. When this function is not enabled, the function block executes normally, only the hardware port has no output.
eOutType	eOutType	LS_CMP_OUTTYPE	[0,3]	0	0: Time mode - output FALSE 1: Time mode - output TRUE 2: Pulse mode - output FALSE 3: Pulse mode - output TRUE

udiOutPara	udiOutPara	UDINT	Follow data types	0	① Time mode: Hold time of open output port, us unit, [1,20000000]. ② Pulse mode: distance value, pulse unit, pulse value range [1~2147483647].
axInvert	Level	ARRAY[0..999] ] OF BOOL	TRUE FALSE	FALSE	This parameter is set when the output type is level mode. Output status of each comparison point TRUE: Output valid FALSE: Output is invalid

**Output variable**

Output variable	Name	Data type	Range	initialization	Descriptive
xDone	Completion	BOOL	TRUE-FALSE	FALSE	TRUE: Homing is done.
xBusy	Enforcement marks	BOOL	TRUE-FALSE	FALSE	The function block has not yet finished executing and the new output value is being calculated
xCommandAborted	Stop sign	BOOL	TRUE-FALSE	FALSE	TRUE: function block execution is interrupted
xError	Error	BOOL	TRUE-FALSE	FALSE	TRUE: An error occurred within the function block
eErrorID	Error ID	SMC_ERROR	-	0	For the error code, please look for 'MC_HSIO.LS_BasicModule.LS_ERROR'.
xPushDone	Completion of press-in compare points	BOOL	TRUE FALSE	FALSE	TRUE: FIFO cache press-in comparison points complete
uiFIFOSize	Points in FIFO	UINT	[0,128]	0	Number of points in FIFO cache
uiPoints	Compare Points	UINT	[0,65535]	0	Comparison Points
fCurrentPos	Current Comparison Position	LREAL	Follow data types	0.0	Current comparison position, unit

	Boolean	Bit string					Integer							Real number		Moment, Duration, Date, String				
	BOOL	BYTE	WORD	DWORD	LWORD	USINT	UINT	UDINT	ULINT	SINT	INT	DINT	LINT	REAL	LREAL	TIME	DATE	TOD	DT	STRING
xExecute	√	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
xValid	√	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
xError	√	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
eErrorID	SMC_ERROR																			

**⊙ Functional Description**

This function block implements a hardware high-speed output port to be triggered when the current count value of the counter is equal to the comparison value set by the array, and hold it for a period of time or a distance or maintain the output state.

Before using the compare function block, the output port must be configured on the corresponding counter interface, and the output port can be configured OUT0~OUT7.

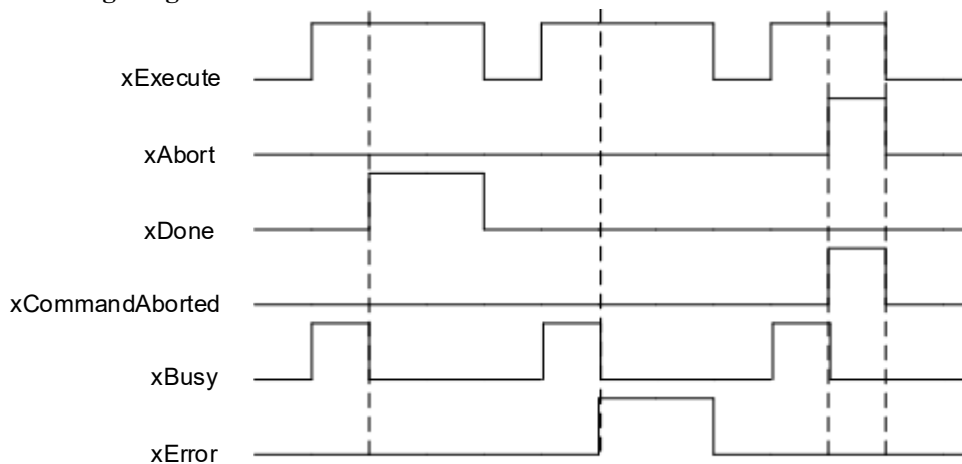
The comparison output type supports three ways. Time mode: When the count value is equal to the set comparison value, the comparison output OUT point will be turned on, and the output OUT point will be turned off after holding  $udiOutPara * 1\mu s$  time. Pulse mode: when the count value is equal to the set comparison value, open the comparison output OUT point, hold  $udiOutPara * 1$  pulse count number, then close the output OUT point. Level mode, when the count value is equal to the set comparison value, according to the level parameter  $axInvert$  set for each point, turn on the comparison output OUT point of that point, when  $axInvert$  is TRUE, the hardware OUT point has output, when  $axInvert$  is FALSE, the hardware OUT point has no output.

When  $axInvert$  is TRUE, there is output from hardware OUT point, when  $axInvert$  is FALSE, there is no output from hardware OUT point.  $xOutEnable$  is to enable the hardware port output function.  $xOutEnable$  is to enable the hardware port output function.  $xOutEnable$  is to enable the hardware port output function.

If the interval between two adjacent triggers is less than the compare output hold time, the output port will continuously hold the output state, and the current output time will directly overwrite the previous output time.

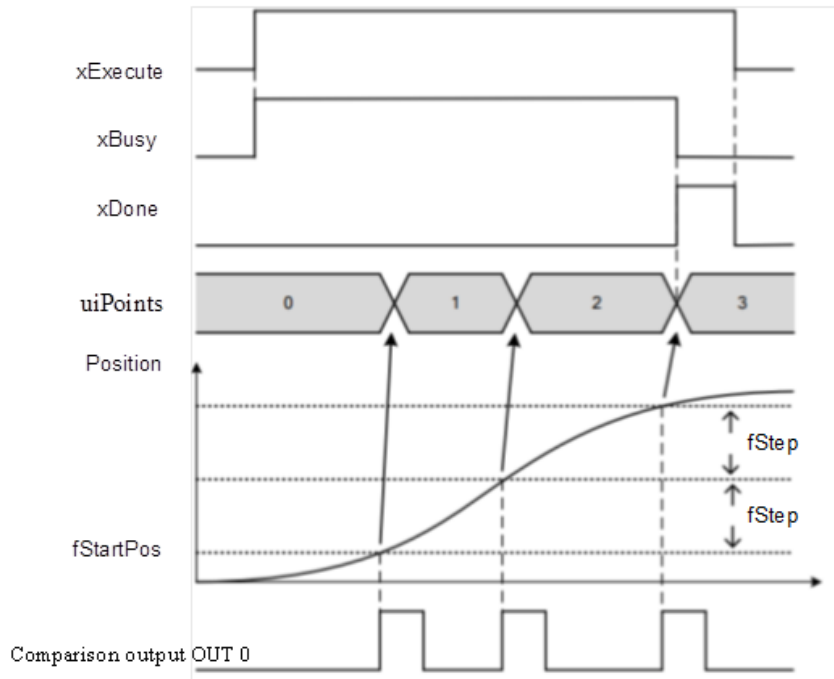
If the rising edge of Execute is triggered again during multi-point comparison, the function block will re-execute the comparison output function from the first point.

In queue comparison mode, the maximum number of comparison points added in a single time is 1000. When it is lower than 1000 points, only the comparison points can be stored in  $afCmpPos$ , which triggers  $xExecute$  to execute the queue comparison without using  $xPush$  to press the cached data. When more than 1000 points, you can dynamically add comparison points during the comparison process, and press the comparison point data  $afCmpPos$  into the cache queue through  $xPush$ , when  $xPushDone$  is TRUE, the press into the cache is complete. At this time, the user can again press the new cache comparison point data.

**⊙ Timing Diagram:**


**⊙ Program demo**

The queue compares 3 positions (uiCmpNum =3) and the instruction timing diagram is shown below.

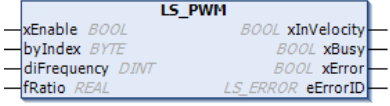

**Note**

When the high-speed compare output function block is activated, the initial state of the output OUT point is automatically flipped to the opposite state of the first compare output point.

### 6.3.8 LS\_PWM

Pulse Width Modulation PWM output.

#### ⊙ Command Format

Instruction	Name	FB/FC	LD	ST	File
LS_PWM	PWM output instruction	FB		<pre>LS_PWM( xEnable:= , byIndex:= , diFrequency:= , fRatio:= , xInVelocity=&gt; , xBusy=&gt; , xError=&gt; , eErrorID=&gt; );</pre>	MC_HS IO

#### ⊙ Related Variables

##### Input variable

Input variable	Name	Data type	Range	initialization	Descriptive
xEnable	Enable	BOOL	TRUE FALSE	FALSE	Function block enable, level trigger
byIndex	PWM serial number	BYTE	0 - 3	0	PWM serial number (0-3)
diFrequency	Frequency	DINT	[1,200000]	1	PWM frequency
fRatio	Duty Cycle	REAL	[0,1]	0	PWM Duty Cycle

##### Output variable

Output variable	Name	Data type	Range	initialization	Descriptive
xInVelocity	Frequency value reaches set value flag	BOOL	TRUE FALSE	FALSE	TRUE Pulse width frequency reaches set value
xBusy	Enforcement marks	BOOL	TRUE-FALSE	FALSE	The function block has not yet finished executing and the new output value is being calculated
xError	Error	BOOL	TRUE-FALSE	FALSE	TRUE: An error occurred within the function block
eErrorID	Error ID	SMC_ERRO R	-	0	For the error code, please look for 'MC_HSIO.LS_BasicModule.LS_ERROR'.

	Boo lea n	Bit string					Integer							Real number		Moment, Duration, Date, String					
	BOOL	BYTE	WORD	DWORD	LWORD	USINT	UINT	UDINT	ULINT	SINT	INT	DINT	LINT	REAL	LREAL	TIME	DATE	TOD	DT	STRING	
xEnable	√	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	
xInVelocity	√	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	
xError	√	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	
eErrorID		SMC_ERROR																			

**⊙ Functional Description**

This function block implements high-speed output OUT point output pulse width modulation PWM. MC508 supports 4-channel PWM output. Before using, the PWM function must be enabled on the corresponding PWM interface, and the correspondence between the output point and the PWM serial number (byIndex) is shown in the table below.

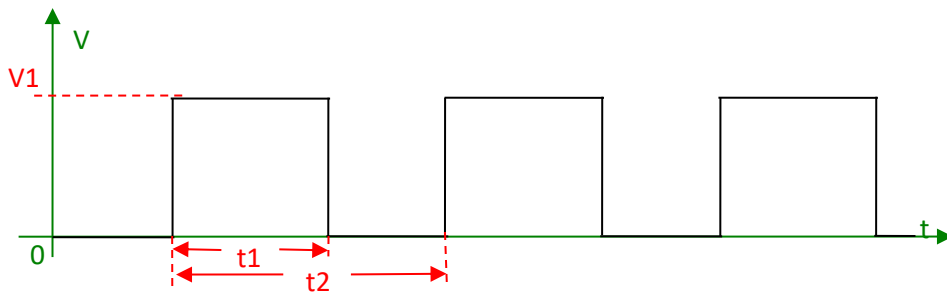
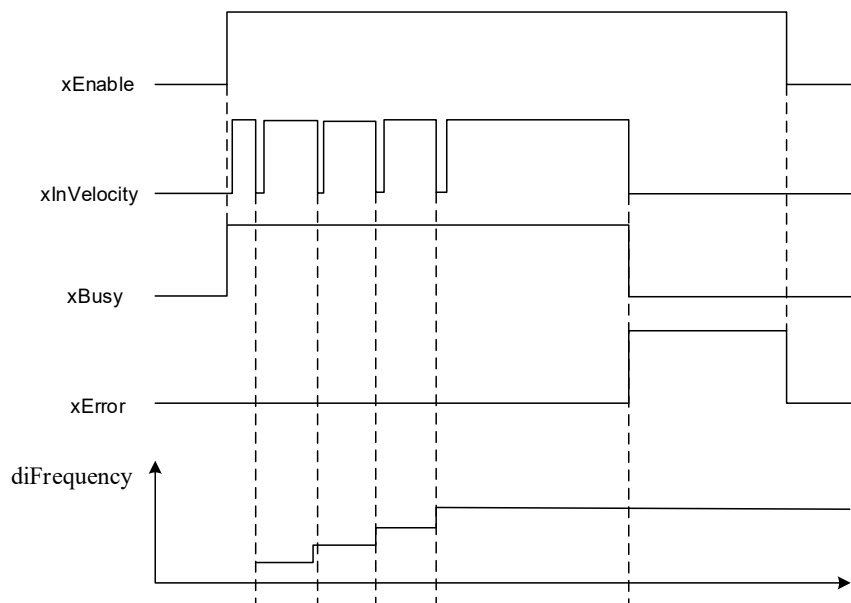
PWM Serial No.	output point
PWM0	OUT4
PWM1	OUT5
PWM2	OUT6
PWM3	OUT7

\* Neither the MC516 nor the MC532 supports PWM

SC508,SC516,SC532 supports 6-channel PWM output. The output point and the PWM serial number (byIndex) is shown in the table below.

PWM Serial No.	output point
PWM0	OUT1
PWM1	OUT3
PWM2	OUT5
PWM3	OUT7
PWM4	OUT11
PWM5	OUT13

Under the enable state ( $xEnable = TRUE$ ), the duty cycle and frequency of pulse width modulation can be modified in real time, the output frequency range: 1HZ~200KHZ. duty cycle ( $t1/t2$ ) range: 0~1, the amplitude is  $V1 = 24V$ , the output waveform is as follows.


**⊙ Timing Diagram:**


### 6.3.9 LS\_EnableInterrupt

Interrupt Enable.

#### ⊙ Command Format

Instruction	Name	FB/FC	LD	ST	File
LS_EnableInterrupt	Interrupt enable instruction	FB		<pre>LS_EnableInterrupt( xEnable:= , uiExternal:= , uiCompare:= , xValid=&gt; , xBusy=&gt; , xError=&gt; , eErrorID=&gt; );</pre>	MC_HS IO

#### ⊙ Related Variables

##### Input variable

Input variable	Name	Data type	Range	initialization	Descriptive
xEnable	Execute	BOOL	[FALSE,TRUE]	FALSE	Enable interrupt, TRUE: enable, FALSE: disable TRUE: enable, FALSE: disable
uiExternal	Turn on external input interrupt	UINT	-	0	Turn on the external input interrupt, e.g. 3, binary 2#11, then the input terminals IN0 and IN1 bits are turned on
uiCMPNum	Turn on comparator interrupt	UINT	-	0	Turn on the Compare Consistent interrupt, e.g. 3, binary 2#11, then the high-speed counting comparators Counter0 and Counter1 are turned on.

##### Output variable

Output variable	Name	Data type	Range	initialization	Descriptive
xValid	Output valid	BOOL	TRUE FALSE	FALSE	TRUE: Function block output valid
xBusy	Enforcement marks	BOOL	TRUE-FALSE	FALSE	The function block has not yet finished executing and the new output value is being calculated
xError	Error	BOOL	TRUE-FALSE	FALSE	TRUE: An error occurred within the function block
eErrorID	Error ID	SMC_ERROR	-	0	For the error code, please look for 'MC_HSIO.LS_BasicModule.LS_ERROR'.

	Boolean	Bit string				Integer							Real number		Moment, Duration, Date, String						
	BOOL	BYTE	WORD	DWORD	LWORD	USINT	UINT	UDINT	ULINT	SINT	INT	DINT	LINT	REAL	LREAL	TIME	DATE	TOD	DT	STRING	
xEnable	√	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
xValid	√	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-

xError	√	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
eErrorID	SMC_ERROR																		

⊙ **Functional Description**

This function block is used to open the external input interrupt, high-speed counter comparison consistent interrupt (single-point comparison); when an interrupt occurs, the associated interrupt task will be executed.

The interrupt task should be configured as 'external' in the software task interface, and the external event selection is 'IN[0~7] InteruptEvent' or 'Counter[0~3] CmpEvent'. When external input interrupt 'IN[0~7] InteruptEvent' is selected, it is also necessary to specify the trigger edge of the input point in the high-speed IO general setup interface, such as a rising edge, a falling edge, or an arbitrary edge.

## 7. Special instruction

### 7.1 SD card file operations


#### 7.1.1 Instruction List

Instruction Type	Name	FB/FC	Functionality
SD card file operations	LS_CopyFromSDCard	FB	Copy files from SD card to local
	LS_CopyToSDCard	FB	Copy local files to SD card
	LS_GetSDCardInformation	FB	Get information about files on the SD card

#### 7.1.2 LS\_CopyFromSDCard

Copy files from SD card to local.

##### ⊙ Command Format

Instruction	Name	FB/FC	LD	ST	File
LS_CopyFromSDCard	Copy files from SD card to local instruction	FB		<pre>LS_CopyFromSDCard( xExecute:=, strSourceFileName:=, strDestFileName:=, xDone=&gt;, xError=&gt;, diErrorID=&gt;, dwCopyedSizeBytes=&gt;);</pre>	MC_Sy sLib

##### ⊙ Related Variables

###### Input variable

Input variable	Name	Data type	Range	initialization	Descriptive
xExecute	Active High	BOOL	TRUE-FALSE	FALSE	The rising edge is valid, the state switches from FALSE to TRUE, and the function block is executed.
strSourceFileName	Source File Name	STRING(32)	-	'tmp.data'	Source file name.
strDestFileName	Destination file name	STRING(32)	-	'tmp2.data'	Target file name.

###### Output variable

Output variable	Name	Data type	Range	initialization	Descriptive
xDone	Output valid	BOOL	TRUE-FALSE	FALSE	TRUE indicates that writing data is complete.
xError	Error	BOOL	TRUE-FALSE	FALSE	FALSE - no error. TRUE - execution error.
diErrorID	Error ID	DINT	Follow the data type	0	Error code. 0 means no error, 1 means SD card not found, 2 means create local file error, 3 means open SD card file error, 4 means file name exception, 16 means file does not exist.
dwCopyedSizeBytes	Length of bytes copied	DWORD	Follow the data type	0	Length of bytes copied.

	Boolean	Bit string					Integer							Real number		Moment, Duration, Date, String				
	BOOL	BYTE	WORD	DWORD	LWORD	USINT	UINT	UDINT	ULINT	SINT	INT	DINT	LINT	REAL	LREAL	TIME	DATE	TOD	DT	STRING
xExecute	√	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
xDone	√	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
xError	√	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
eErrorID	SMC_ERROR																			

**⊙ Functional Description**

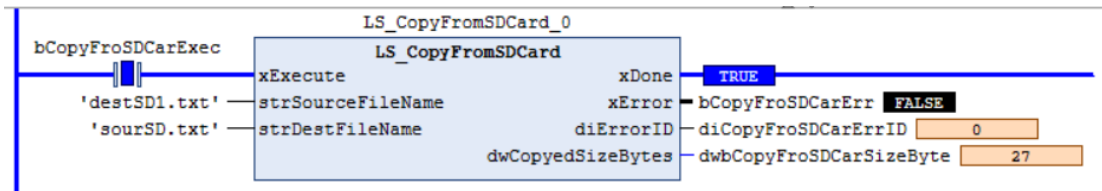
Setting xEnable to TRUE puts the function block into a runnable state; setting xEnable to FALSE keeps the function block from running.

Files copied from the SD card are saved in the local UsrData file.

**⊙ Program demo**
**ST:**

```


LS_CopyFromSDCard_0(xExecute TRUE := bCopyFroSDCarExec TRUE,
    strSourceFileName 'destSD1.txt' := 'destSD1.txt',
    strDestFileName 'sourSD.txt' := 'sourSD.txt',
    xDone TRUE => bCopyFroSDCarDone TRUE,
    xError FALSE => bCopyFroSDCarErr FALSE,
    diErrorID 0 => diCopyFroSDCarErrID 0,
    dwCopiedSizeBytes 27 => dwbCopyFroSDCarSizeByte 27);
    
```

**LD:**


### 7.1.3 LS\_CopyToSDCard

Copy local files to the SD card.

#### ⊙ Command Format

Instruction	Name	FB/FC	LD	ST	File
LS_CopyToSDCard	Copy local files to SD card instruction	FB		<pre> LS_CopyToSDCard( xExecute:=, strSourceFileName:=, strDestFileName:=, xDone=&gt;, xError=&gt;, diErrorID=&gt;, dwCopyedSizeBytes=&gt;);                     </pre>	MC_SysLib

#### ⊙ Related Variables

##### Input variable

Input variable	Name	Data type	Range	initialization	Descriptive
xExecute	Active High	BOOL	TRUE-FALSE	FALSE	The rising edge is valid, the state switches from FALSE to TRUE, and the function block is executed.
strSourceFileName	Source File Name	STRING(32)	-	'tmp.data'	Source file name.
strDestFileName	Destination file name	STRING(32)	-	'tmp2.data'	Target file name.

##### Output variable

Output variable	Name	Data type	Range	initialization	Descriptive
xDone	Output valid	BOOL	TRUE-FALSE	FALSE	TRUE indicates that writing data is complete.
xError	Error	BOOL	TRUE-FALSE	FALSE	FALSE - no error. TRUE - execution error.
diErrorID	Error ID	DINT	Follow the data type	0	Error code. 0 means no error, 1 means SD card not found, 2 means create local file error, 3 means open SD card file error, 4 means file name exception, 16 means file does not exist.
dwCopyedSizeBytes	Length of bytes copied	DWORD	Follow the data type	0	Length of bytes copied.

	Boo	Bit string					Integer							Real number		Moment, Duration, Date, String						
	lea	BOO	BYT	WORD	DWORD	LWORD	USINT	UINT	UDINT	ULINT	SINT	INT	DINT	LINT	REAL	LREAL	TIME	DATE	TOD	DT	STRING	
xExecute	√	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
xDone	√	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
xError	√	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
eErrorID		SMC_ERROR																				

**⊙ Functional Description**

Setting xEnable to TRUE puts the function block into a runnable state; setting xEnable to FALSE keeps the function block from running.

Before executing the function block, you need to place the copied file in the local UsrData file.

**⊙ Program demo**

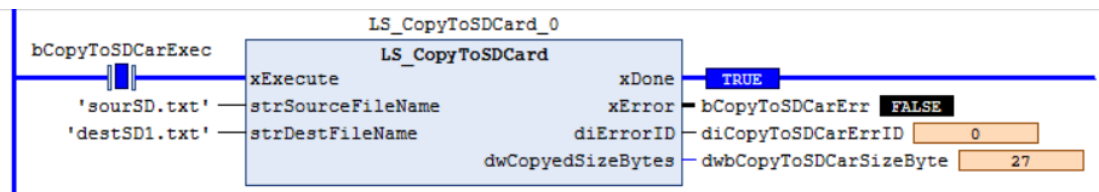
ST:

```

LS_CopyToSDCard_0(xExecute TRUE := bCopyToSDCarExec TRUE ,
  strSourceFileName 'sourSD.txt' := 'sourSD.txt',
  strDestFileName 'destSD1.tx' := 'destSD1.txt',
  xDone TRUE => bCopyToSDCarDone TRUE ,
  xError FALSE => bCopyToSDCarErr FALSE ,
  diErrorID 0 => diCopyToSDCarErrID 0 ,
  dwCopedSizeBytes 27 => dwbCopyToSDCarSizeByte 27 );

```


LD:



### 7.1.4 LS\_GetSDCardInformation

Copy local files to the SD card.

#### ⊙ Command Format

Instruction	Name	FB/FC	LD	ST	File
LS_GetSDCardInformation	Get information about files on the SD card instruction	FB		LS_GetSDCardInformation( xExecute:=, strFileExtension:=, iMaxFileNums:=, pstFile_Information_Point:= , xDone=>, xError=>, diErrorID=>, iNumberOfFiles=>);	MC_Sy sLib

#### ⊙ Related Variables

##### Input variable

Input variable	Name	Data type	Range	initialization	Descriptive
xExecute	Active High	BOOL	TRUE-FALSE	FALSE	The rising edge is valid, the state switches from FALSE to TRUE, and the function block is executed.
strFileExtension	Source file name	STRING(32)	-	-	File extension '.data'
iMaxFileNums	Maximum number of files allowed to be read	INT	-	20	Maximum number of files allowed to be read
pstFile_Information_Point	File information pointer	POINTER TO File_Information	-	-	File information pointer

##### Output variable

Output variable	Name	Data type	Range	initialization	Descriptive
xDone	Output valid	BOOL	TRUE-FALSE	FALSE	TRUE indicates that writing data is complete.
xError	Error	BOOL	TRUE-FALSE	FALSE	FALSE - no error. TRUE - execution error.
diErrorID	Error ID	DINT	Follow the data type	0	Error code. 0 means no error, 1 means SD card not found, 2 means create local file error, 3 means open SD card file error, 4 means file name exception, 16 means file does not exist.
iNumberOfFiles	Length of bytes copied	INT	Follow the data type	0	The number of files read.

	Boolean	Bit string					Integer							Real number		Moment, Duration, Date, String				
	BOOL	BYTE	WORD	DWORD	LWORD	USINT	UINT	UDINT	ULINT	SINT	INT	DINT	LINT	REAL	LREAL	TIME	DATE	TOD	DT	STRING
xExecute	√	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
xDone	√	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
xError	√	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
eErrorID	SMC_ERROR																			

**⊙ Functional Description**

Set xEnable to TRUE, the function block enters the runnable state; set xEnable to FALSE, the function block does not run.

pstFile\_Information\_Point input/output pin, you need to define the file information structure variable first, it is recommended to use array type, when there are more than one file in the SD card, you can read the file information together. Then use the file pointer variable to get the first address of the file information structure, as follows:

File\_Infor:ARRAY[0..19] OF File\_Information.

fileinfor: POINTER TO File\_Information := ADR(File\_Infor[0]);

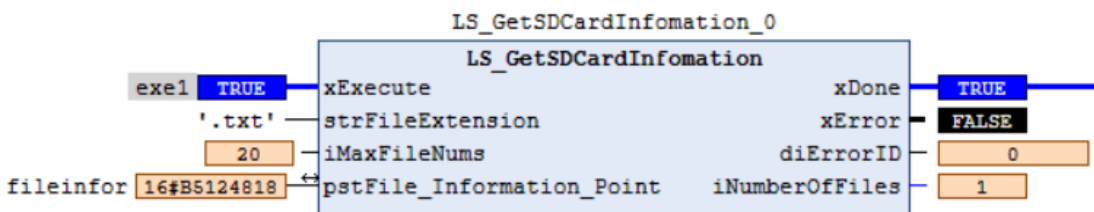
**⊙ Program demo**

ST:

```

LS_GetSDCardInfomation_0(xExecute TRUE := exel TRUE,
    strFileExtension '.txt' := '.txt',
    iMaxFileNums := ,
    pstFile_Information_Point 16#B5124818 := fileinfor 16#B5124818,
    xDone TRUE => done TRUE,
    xError FALSE => err FALSE,
    diErrorID => ,
    iNumberOfFiles 1 => NumberofFiles 1 ); RETURN
    
```

LD:



## 7.2 System instruction

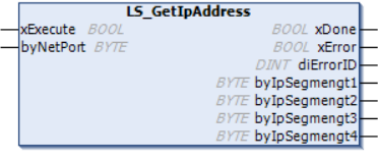
### 7.2.1 Instruction List

Instruction Type	Name	FB/FC	Functionality
System command	LS_GetIpAddress	FB	Get PLC IP address
	LS_SetIpAddress	FB	Set PLC IP address
	GetPLCLoad	FC	Get PLC load factor
	GetPLCVersion	FC	Get PLC firmware version
	ColdResetApp	FC	Cold reset PLC
	WarmResetApp	FC	Thermal Reset PLC
	LS_ReadDintDT	FB	Get system time (Dint type)
	LS_ReadStringDT	FB	Get system time (String type)
	LS_SetDintDT	FB	Set system time (Dint type)
	LS_SetStringDT	FB	Set system time(Dint type)

### 7.2.2 LS\_GetIpAddress

Copy local files to the SD card.

#### ⊙ Command Format

Instruction	Name	FB/FC	LD	ST	File
LS_GetIpAddress	Get PLC IP address instruction	FB		<pre>LS_GetIpAddress( xExecute:=, byNetPort:=, xDone=&gt;, xError=&gt;, diErrorID=&gt;, byIpSegmengt1=&gt;, byIpSegmengt2=&gt;, byIpSegmengt3=&gt;, byIpSegmengt4=&gt;);</pre>	MC_SysLib

#### ⊙ Related Variables

##### Input variable

Input variable	Name	Data type	Range	initialization	Descriptive
xExecute	Active High	BOOL	TRUE-FALSE	FALSE	The rising edge is valid, the state switches from FALSE to TRUE, and the function block is executed.
byNetPort	EtherNet Port ID	BYTE	0-255	0	EtherNet Port numbering, starting with 0.

##### Output variable

Output variable	Name	Data type	Range	initialization	Descriptive
xDone	Output valid	BOOL	TRUE-FALSE	FALSE	TRUE indicates that writing data is complete.
xError	Error	BOOL	TRUE-FALSE	FALSE	FALSE - no error. TRUE - execution error.
diErrorID	Error ID	DINT	Follow the data type	0	Error code. 0 means no error, 1 means SD card not found, 2 means create local file error, 3 means open SD card file error, 4 means file name exception, 16 means file does not exist.
byIpSegmengt1	IP address segment 1	BYTE	0-255	0	The first segment of the obtained IP address, i.e. the network type.

byIpSegmengt2	IP address segment 2	BYTE	0-255	0	The 2nd segment of the obtained IP address, i.e. the network range.
byIpSegmengt3	IP address segment 3	BYTE	0-255	0	The 3rd segment of the obtained IP address, i.e., the network bits.
byIpSegmengt4	IP address segment 4	BYTE	0-255	0	The 4th segment of the obtained IP address, i.e., host bits.

	Boolean	Bit string					Integer							Real number		Moment, Duration, Date, String				
	BOOL	BYTE	WORD	DWORD	LWORD	USINT	UINT	UDINT	ULINT	SINT	INT	DINT	LINT	REAL	LREAL	TIME	DATE	TOD	DT	STRING
xExecute	√	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
xDone	√	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
xError	√	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
eErrorID	SMC_ERROR																			

### ⊙ Functional Description

Setting xExecute to TRUE, the function block enters the runnable state; setting xExecute to FALSE, the function block does not run.

According to the number of network port channels configured by the host, input the corresponding number of the network port you need to acquire into byNetPort, the default starts from 0. If the host has only 1 Ethernet port, input 0.

If you need to convert the acquired IP address into string format, first use the BYTE\_TO\_STRING instruction to convert each segment of the address into string format, and then use the CONCAT string concatenation function to combine them..

### ⊙ Program demo

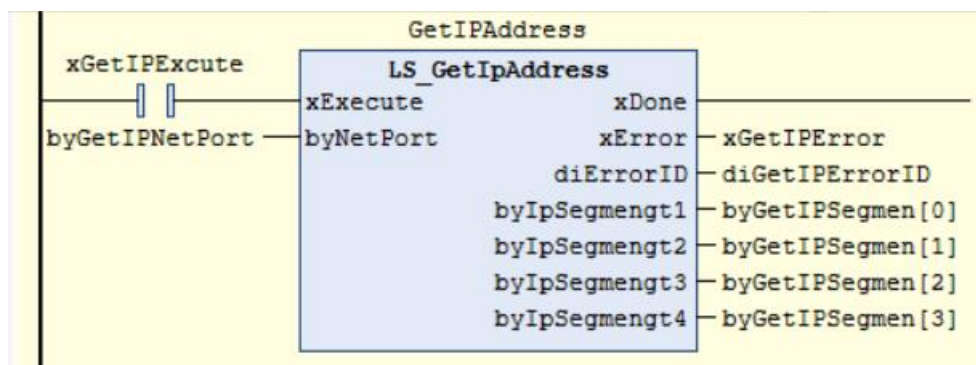
ST:

```

1  GetIPAddress (
2      xExecute TRUE := xGetIPExcute TRUE ,
3      byNetPort 0 := byGetIPNetPort 0 ,
4      xDone TRUE => xGetIPDone TRUE ,
5      xError FALSE => xGetIPError FALSE ,
6      diErrorID 0 => diGetIPErrorID 0 ,
7      byIpSegmengt1 192 => byGetIPSegmen [0] 192 ,
8      byIpSegmengt2 168 => byGetIPSegmen [1] 168 ,
9      byIpSegmengt3 1 => byGetIPSegmen [2] 1 ,
10     byIpSegmengt4 3 => byGetIPSegmen [3] 3 ) ; RETURN

```

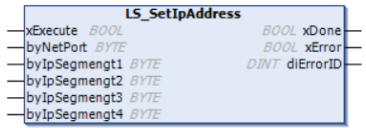
LD:



### 7.2.3 LS\_SetIpAddress

Configure the PLC network IP address. The IP address of the PLC network port is configured according to the network number through the external library method in 4 segments: network type, network range, network bits, and host bits.

#### ⊙ Command Format

Instruction	Name	FB/FC	LD	ST	File
LS_SetIpAddress	Set PLC IP address instruction	FB		<pre>LS_SetIpAddress( xExecute:=, byNetPort:=, byIpSegmengt1:=, byIpSegmengt2:=, byIpSegmengt3:=, byIpSegmengt4:=, xDone=&gt;, xError=&gt;, diErrorID=&gt;);</pre>	MC_SysLib

#### ⊙ Related Variables

##### Input variable

Input variable	Name	Data type	Range	initialization	Descriptive
xExecute	Active High	BOOL	TRUE-FALSE	FALSE	The rising edge is valid, the state switches from FALSE to TRUE, and the function block is executed.
byNetPort	EtherNet Port ID	BYTE	0-255	0	EtherNet Port numbering, starting with 0.
ByIpSegmengt1	IP address segment 1	BYTE	0-255	192	IP address segment 1 that needs to be configured, i.e. network type.
ByIpSegmengt2	IP address segment 2	BYTE	0-255	168	The 2nd segment of the IP address to be configured, i.e. the network range.
ByIpSegmengt3	IP address segment 3	BYTE	0-255	1	The 3rd segment of the IP address to be configured, i.e. the network bits.
ByIpSegmengt4	IP address segment 4	BYTE	0-255	3	The IP address to be configured, segment 4, i.e., host bits.

##### Output variable

Output variable	Name	Data type	Range	initialization	Descriptive
xDone	Output valid	BOOL	TRUE-FALSE	FALSE	TRUE indicates that writing data is complete.
xError	Error	BOOL	TRUE-FALSE	FALSE	FALSE - no error. TRUE - execution error.
diErrorID	Error ID	DINT	Follow the data type	0	Error code. 01:IP address segment 1 data value cannot be 0 02::NetPort value is out of range.

	Boolean	Bit string					Integer							Real number		Moment, Duration, Date, String				
	BOOL	BYTE	WORD	DWORD	LWORD	USINT	UINT	UDINT	ULINT	SINT	INT	DINT	LINT	REAL	LREAL	TIME	DATE	TOD	DT	STRING
xExecute	√	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
xDone	√	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
xError	√	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
eErrorID	SMC_ERROR																			

### ⊙ Functional Description

Setting xExecute to TRUE, the function block enters the runnable state; setting xExecute to FALSE, the function block does not run.

According to the number of network port channels configured by the host, input the corresponding number of the network port that needs to be acquired into byNetPort, the default starts from 0. If the host has only 1 Ethernet port, input 0.

Write the IP address segments that need to be configured into ByIpSegmengt1, ByIpSegmengt2, ByIpSegmengt3, ByIpSegmengt4.

Note: After successfully configuring the new IP address, the PLC must be powered off and restarted to take effect.

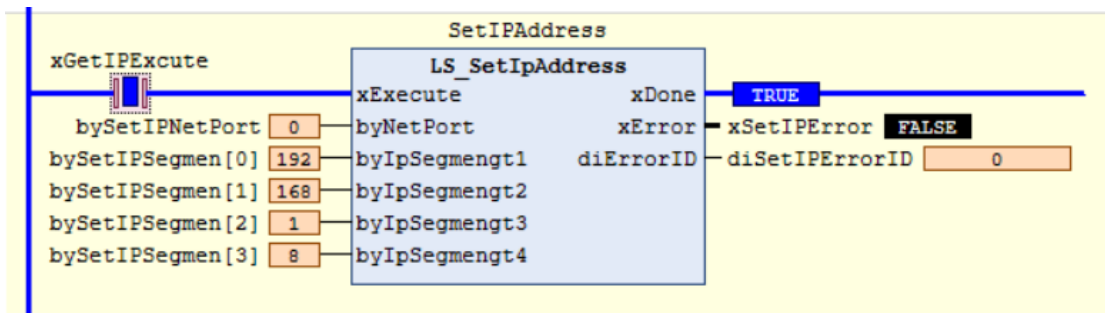
### ⊙ Program demo

ST:

```

SetIPAddress (
    xExecute TRUE := xSetIPExecute TRUE, //写入IP
    byNetPort 0 := bySetIPNetPort 0,
    byIpSegmengt1 192 := bySetIPSegmen[0] 192,
    byIpSegmengt2 168 := bySetIPSegmen[1] 168,
    byIpSegmengt3 1 := bySetIPSegmen[2] 1,
    byIpSegmengt4 12 := bySetIPSegmen[3] 12,
    xDone TRUE => xSetIPDone TRUE,
    xError FALSE => xSetIPError FALSE,
    diErrorID 0 => diSetIPErrorID 0 );
RETURN
    
```

LD:



### 7.2.4 GetPLCLoad

Get the CPU load rate of the PLC. Stores the CPU load rate information in an internal property variable.

#### ⊙ Command Format

Instruction	Name	FB/FC	LD	ST	File
GetPLCLoad dress	Get PLC load factor instruction	FB		<pre>GetPLCLoad( xEnable:=, xValid=&gt;, xError=&gt;, diErrorID=&gt;, fLoad=&gt;);</pre>	MC_Sy sLib

#### ⊙ Related Variables

##### Input variable

Input variable	Name	Data type	Range	initialization	Descriptive
xEnable	Active High	BOOL	TRUE-FALSE	FALSE	Active High

##### Output variable

Output variable	Name	Data type	Range	initialization	Descriptive
xValid	Output valid	BOOL	TRUE-FALSE	FALSE	TRUE indicates read enable.
xError	Error	BOOL	TRUE-FALSE	FALSE	FALSE - no error. TRUE - execution error.
diErrorID	Error ID	DINT	Follow the data type	0	Error code. 01:IP address segment 1 data value cannot be 0 02::NetPort value is out of range.
fLoad	CPU load factor	REAL	Follow the data type		The obtained CPU load factor.

	Boo lea n	Bit string					Integer							Real number		Moment, Duration, Date, String				
	BOOL	BYTE	WORD	DWORD	LWORD	USINT	UINT	UDINT	ULINT	SINT	INT	DINT	LINT	REAL	LREAL	TIME	DATE	TOD	DT	STRING
xEnable	√	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
xValid	√	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
xError	√	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
eErrorID		SMC_ERROR																		

#### ⊙ Functional Description

When xEnable is set to TRUE, the function block enters the runnable state; when xEnable is set to FALSE, the function block does not run.

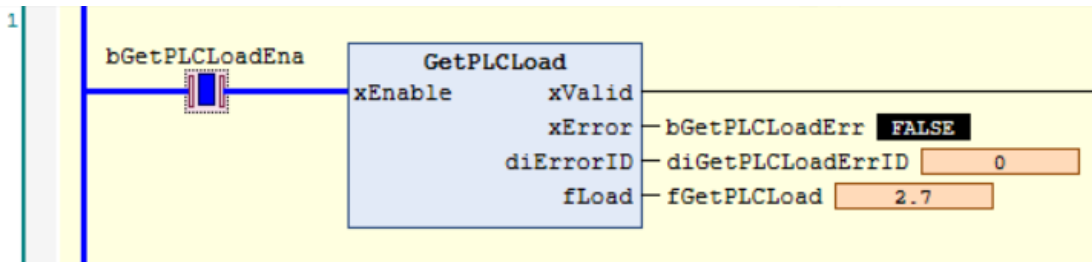
If xEnable is kept as TRUE, fLoad follows the system CPU load in real time.

**⊙ Program demo**
**ST:**

```

GetPLCLoad(xEnable:= bGetPLCLoadEna TRUE,
           xValid=>bGetPLCLoadValid TRUE,
           xError=>bGetPLCLoadErr FALSE,
           diErrorID=> diGetPLCLoadErrID 0,
           fLoad=> fGetPLCLoad 2.93);
RETURN


```

**LD:**


### 7.2.5 GetPLCVersion

Get the CPU load rate of the PLC. Stores the CPU load rate information in an internal property variable.

#### ☉ Command Format

Instruction	Name	FB/FC	LD	ST	File
GetPLCVersion	Get PLC firmware version instruction	FB		<pre>GetPLCLoad( xEnable:=, xValid=&gt;, xError=&gt;, diErrorID=&gt;, udInfo_CPU_ID=&gt;, diHardware_Ver=&gt;, diFirmwareH_Ver=&gt;, diFirmwareS_Ver=&gt;);</pre>	MC_Sy sLib

#### ☉ Related Variables

##### Input variable

Input variable	Name	Data type	Range	initialization	Descriptive
xEnable	Active High	BOOL	TRUE-FALSE	FALSE	Active High

##### Output variable

Output variable	Name	Data type	Range	initialization	Descriptive
xValid	Output valid	BOOL	TRUE-FALSE	FALSE	TRUE indicates read enable.
xError	Error	BOOL	TRUE-FALSE	FALSE	FALSE - no error. TRUE - execution error.
diErrorID	Error ID	DINT	Follow the data type	0	Error code. 01:IP address segment 1 data value cannot be 0 02::NetPort value is out of range.
udInfo_CPU_ID	CPU ID	UDINT	Follow the data type	-	CPU ID number.
diHardware_Ver	Controller Model	DINT	Follow the data type	-	Controller model number.
diFirmwareH_Ver	FPGA Version	DINT	Follow the data type	-	Hexadecimal display is required.
diFirmwareS_Ver	ARM Version	DINT	Follow the data type	-	Hexadecimal display is required.

	Boolean	Bit string					Integer							Real number		Moment, Duration, Date, String					
	BOOL	BYTE	WORD	DWORD	LWORD	USINT	UINT	UDINT	ULINT	SINT	INT	DINT	LINT	REAL	LREAL	TIME	DATE	TOD	DT	STRING	
xEnable	√	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	
xValid	√	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	
xError	√	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	
eErrorID		SMC_ERROR																			

**⊙ Functional Description**

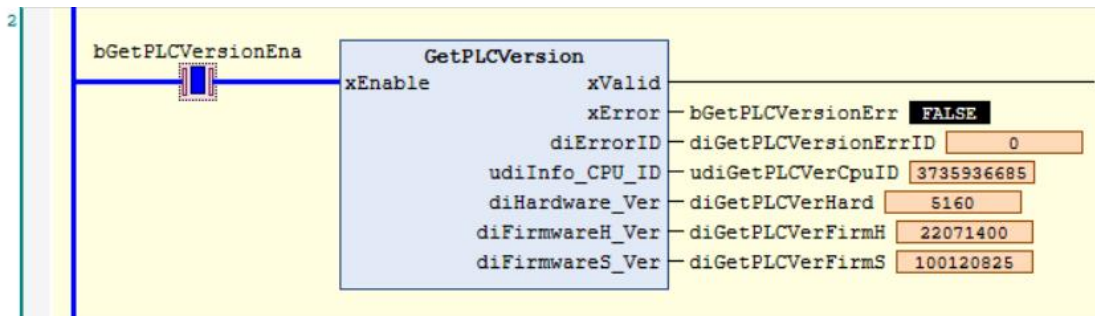
When xEnable is set to TRUE, the function block enters the runnable state; when xEnable is set to FALSE, the function block does not run.

If xEnable is kept as TRUE, fLoad follows the system CPU load in real time.

**⊙ Program demo**
**ST:**

```

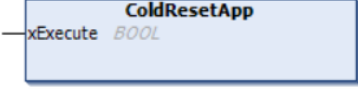
GetPLCVersion(xEnable:= bGetPLCLoadEna TRUE,
              xValid=> bGetPLCLoadValid TRUE,
              xError=> bGetPLCLoadErr FALSE,
              diErrorID=> diGetPLCLoadErrID 0,
              udiInfo_CPU_ID=> udiGetPLCVerCpuID 3735936685,
              diHardware_Ver=> diGetPLCVerHard 5160,
              diFirmwareH_Ver=> diGetPLCVerFirmH 22071400,
              diFirmwareS_Ver=> diGetPLCVerFirmS 100120825 );RETURN
    
```

**LD:**


### 7.2.6 ColdResetApp

Cold reset PLC. after executing a function block, the PLC switches from the RUN state to the STOP state and then back to the RUN state. After reset, the VAR type variable and VAR RETAIN type variable in the programme will initialise the value, and the VAR PERSISTENT RETAIN or VAR RETAIN PERSISTENT type variable will keep the original value.

#### ⊙ Command Format

Instruction	Name	FB/FC	LD	ST	File
ColdResetApp	Cold reset PLC instruction	FB		ColdResetApp(xEnable:=);	MC_SysLib

#### ⊙ Related Variables

##### Input variable

Input variable	Name	Data type	Range	initialization	Descriptive
xEnable	Active High	BOOL	TRUE-FALSE	FALSE	Active High

#### ⊙ Functional Description

When xEnable is set to TRUE, the function block enters the runnable state; when xEnable is set to FALSE, the function block does not run.

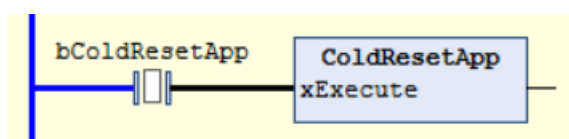
After executing the function block, xEnable automatically returns to the FALSE state.

#### ⊙ Program demo

ST:

```
ColdResetApp(xExecute:= bColdResetApp FALSE);
```

LD:



### 7.2.7 WarmResetApp

Thermal Reset PLC. after executing a function block, the PLC switches from the RUN state to the STOP state and back to the RUN state. After the reset, the VAR type variables in the programme will initialise their values, and the VAR RETAIN type variables and the VAR PERSISTENT RETAIN or VAR RETAIN PERSISTENT type variables will keep their original values.

#### ⊙ Command Format

Instruction	Name	FB/FC	LD	ST	File
WarmResetApp	Thermal Reset PLC instruction	FB		WarmResetApp(xEnable:=);	MC_SysLib

#### ⊙ Related Variables

##### Input variable

Input variable	Name	Data type	Range	initialization	Descriptive
xExecute	High level active	BOOL	TRUE-FALSE	FALSE	TRUE: Activates the processing of the function block.

#### ⊙ Functional Description

When xEnable is set to TRUE, the function block enters the runnable state; when xEnable is set to FALSE, the function block does not run.

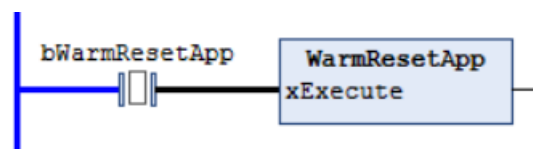
After executing the function block, xEnable automatically returns to the FALSE state.

#### ⊙ Program demo

ST:

```
WarmResetApp(xExecute:= bWarmResetApp FALSE);RETURN
```

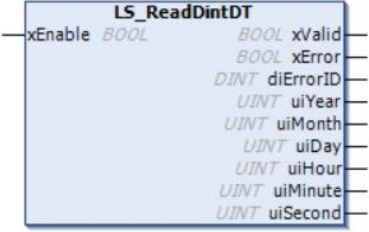
LD:



### 7.2.8 LS\_ReadDintDT

Get the CPU load rate of the PLC. Stores the CPU load rate information in an internal property variable.

#### ⊙ Command Format

Instruction	Name	FB/FC	LD	ST	File
LS_ReadDintDT	Get system time (Dint type) instruction	FB		<pre>LS_ReadDintDT( xEnable=, xValid=&gt;, xError=&gt;, diErrorID=&gt;, uiYear=&gt;, uiMonth=&gt;, uiDay=&gt;, uiHour=&gt;, uiMinute=&gt;, uiSecond=&gt;);</pre>	MC_Sy sLib

#### ⊙ Related Variables

##### Input variable

Input variable	Name	Data type	Range	initialization	Descriptive
xEnable	Active High	BOOL	TRUE-FALSE	FALSE	Active High

##### Output variable

Output variable	Name	Data type	Range	initialization	Descriptive
xValid	Output valid	BOOL	TRUE-FALSE	FALSE	TRUE indicates read enable.
xError	Error	BOOL	TRUE-FALSE	FALSE	FALSE - no error. TRUE - execution error.
diErrorID	Error ID	DINT	Follow the data type	0	Error code. 01:IP address segment 1 data value cannot be 0 02::NetPort value is out of range.
uiYear	year	UINT	Follow the data type	1970~2106	year
uiMonth	month	UINT		1~12	month
uiDay	day	UINT		1~31	day
uiHour	hour	UINT		0~23	hour
uiMinute	minutes	UINT		0~59	minutes
uiSecond	seconds	UINT		0~59	seconds

	Boolean	Bit string				Integer							Real number		Moment, Duration, Date, String					
	BOOL	BYTE	WORD	DWORD	LWORD	USINT	UINT	UDINT	ULINT	SINT	INT	DINT	LINT	REAL	LREAL	TIME	DATE	TOD	DT	STRING
xEnable	√	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
xValid	√	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
xError	√	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
eErrorID	SMC_ERROR																			

**⊙ Functional Description**

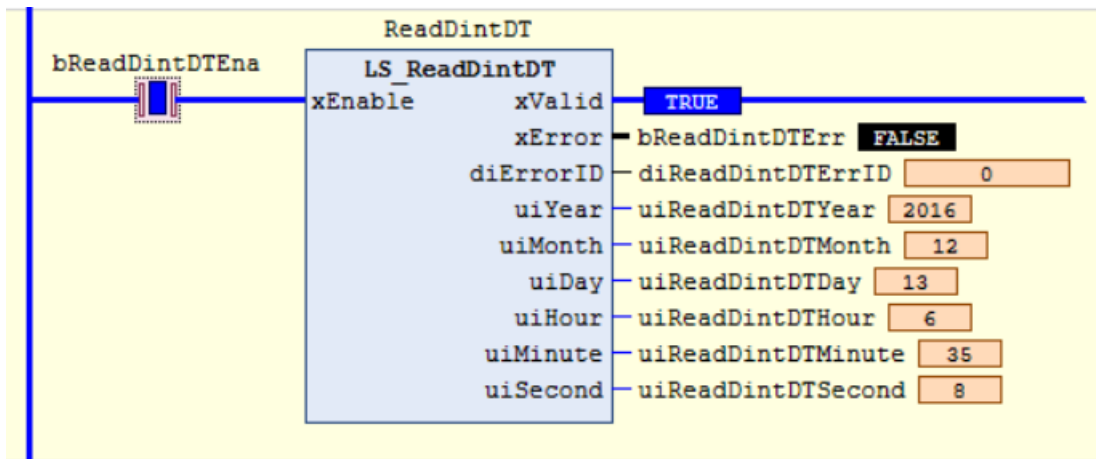
Setting xEnable to TRUE puts the function block into a runnable state; setting xEnable to FALSE keeps the function block from running.

If xEnable is held to TRUE, the PLC system time is read in real time into the internal property variable.

**⊙ Program demo**
**ST:**

```

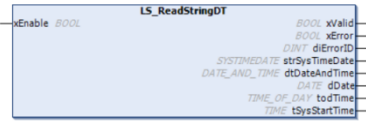
ReadDintDT (xEnable TRUE := bReadDintDTena TRUE ,
            xValid TRUE => bReadDintDTValid TRUE ,
            xError FALSE => bReadDintDTerr FALSE ,
            diErrorID 0 => diReadDintDTerrID 0 ,
            uiYear 2016 => uiReadDintDTYear 2016 ,
            uiMonth 12 => uiReadDintDTMonth 12 ,
            uiDay 13 => uiReadDintDTDay 13 ,
            uiHour 6 => uiReadDintDTHour 6 ,
            uiMinute 28 => uiReadDintDTMinute 28 ,
            uiSecond 22 => uiReadDintDTSecond 22 ) ;
    
```

**LD:**


## 7.2.9 LS\_ReadStringDT

Get the CPU load rate of the PLC. Stores the CPU load rate information in an internal property variable.

### ⊙ Command Format

Instruction	Name	FB/FC	LD	ST	File
LS_ReadStringDT	Get system time (String type) instruction	FB		<pre>LS_ReadStringDT( xEnable:=, xValid=&gt;, xError=&gt;, diErrorID=&gt;, strSysTimeDate=&gt;, dtDateAndTime=&gt;, dDate=&gt;, todTime=&gt;, tSysStartTime=&gt;);</pre>	MC_SysLib

### ⊙ Related Variables

#### Input variable

Input variable	Name	Data type	Range	initialization	Descriptive
xEnable	Active High	BOOL	TRUE-FALSE	FALSE	Active High

#### Output variable

Output variable	Name	Data type	Range	initialization	Descriptive
xValid	Output valid	BOOL	TRUE-FALSE	FALSE	TRUE indicates read enable.
xError	Error	BOOL	TRUE-FALSE	FALSE	FALSE - no error. TRUE - execution error.
diErrorID	Error ID	DINT	Follow the data type	0	Error code. 01:IP address segment 1 data value cannot be 0 02::NetPort value is out of range.
strSysTimeDate	System Time Structures	SYSTIMEDATE	Follow the data type		The structure variables contain: wYear, wMonth, wDay, wHour, wMinute, wSecond, wMilliseconds, wDayofWeek, wYday.
dtDateAndTime	System Time	DATE_AND_TIME	Follow the data type		Format:DT#2016-12-13-10:20: 43.
dDate	System Date	DATE	Follow the data type		Format:D#2016-12-13.
todTime	System hours, minutes and seconds	TIME_OF_DAY	Follow the data type		Format:TOD#10: 20: 43.
tSysStartTime	System runtime constant	TIME	Follow the data type		Format:T#4h22m22s333ms.

### ⊙ Functional Description

Setting xEnable to TRUE puts the function block into a runnable state; setting xEnable to FALSE keeps the function block from running.

If xEnable is held to TRUE, the PLC system time is read in real time into the internal property variable.

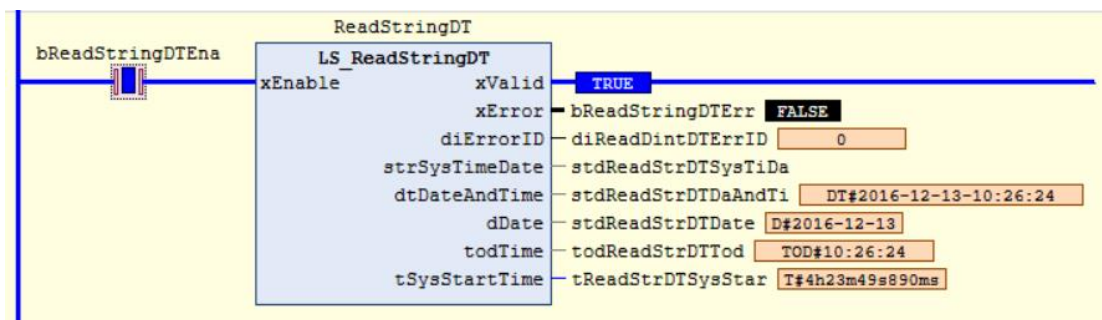
SYSTIMEDATE structure library file path: MC\_SysLib.SysTimeRtc.

**⊙ Program demo**
**ST:**

```

ReadStringDT(xEnable TRUE := bReadStringDTena TRUE,
  xValid TRUE => bReadStrDTValid TRUE,
  xError FALSE => bReadDintDTerr FALSE,
  diErrorID 0 => diReadStrDTerrID 0,
  strSysTimeDate => stdReadStrDTSysTiDa,
  dtDateAndTime DT#2016-12-13-10:25:24 => stdReadStrDTDaAndTi DT#2016-12-13-10:25:24,
  dDate D#2016-12-13 => stdReadStrDTDate D#2016-12-13,
  todTime TOD#10:25:24 => todReadStrDTTod TOD#10:25:24,
  tSysStartTime T#4h22m49s706ms => tReadStrDTSysStar T#4h22m49s706ms );
RETURN

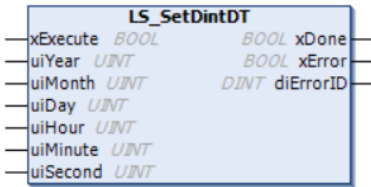
```

**LD:**


### 7.2.10 LS\_SetDintDT

Set the PLC system time (Dint type) and write the information in the internal attribute variable to the PLC system time.

#### ☉ Command Format

Instruction	Name	FB/FC	LD	ST	File
LS_SetDintDT	Set system time (Dint type) instruction	FB		<pre>LS_SetDintDT( xExecute:=, uiYear:=, uiMonth:=, uiDay:=, uiHour:=, uiMinute:=, uiSecond:=, xDone=&gt;, xError=&gt;, diErrorID=&gt;);</pre>	MC_Sy sLib

#### ☉ Related Variables

##### Input variable

Input variable	Name	Data type	Range	initialization	Descriptive
xExecute	High level active	BOOL	TRUE-FALSE	FALSE	The rising edge is valid, the state switches from FALSE to TRUE, and the function block is executed.
uiYear	Year	UINT	1970~2106	2020	Year.
uiMonth	Month	UINT	1~12	4	Month.
uiDay	Day	UINT	1~31	17	Day.
uiHour	hour	UINT	0~23	10	Hour.
uiMinute	minutes	UINT	0~59	10	Minutes.
uiSecond	seconds	UINT	0~59	10	Seconds.

##### Output variable

Output variable	Name	Data type	Range	initialization	Descriptive
xDone	Implementation completed	BOOL	TRUE-FALSE	FALSE	The rising edge is valid, the state switches from FALSE to TRUE, and the function block is executed.
xError	Error	BOOL	TRUE-FALSE	FALSE	FALSE - no error. TRUE - execution error.
diErrorID	Error ID	DINT	Follow the data type	0	Error code. 01:IP address segment 1 data value cannot be 0 02::NetPort value is out of range.

#### ☉ Functional Description

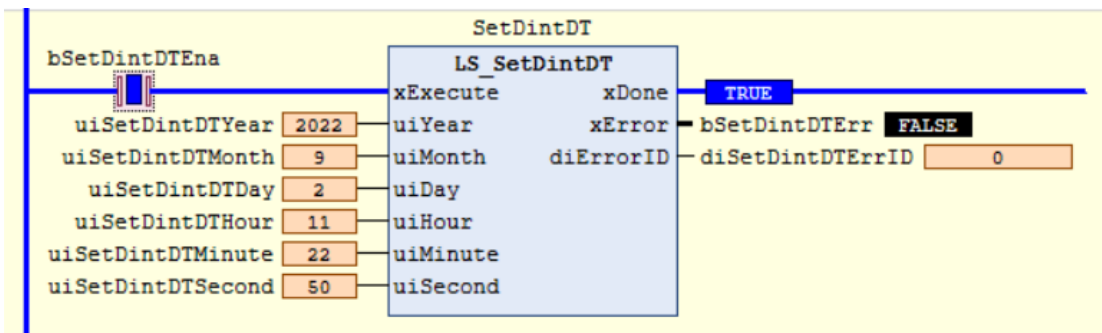
Setting xExecute to TRUE puts the function block into a runnable state; setting xExecute to FALSE keeps the function block from running.

**Ⓞ Program demo**
**ST:**

```

SetDintDT (xExecute TRUE := bSetDintDTena TRUE ,
           uiYear 2022 := uiSetDintDTYear 2022 ,
           uiMonth 9 := uiSetDintDTMonth 9 ,
           uiDay 2 := uiSetDintDTDay 2 ,
           uiHour 11 := uiSetDintDTHour 11 ,
           uiMinute 22 := uiSetDintDTMinute 22 ,
           uiSecond 50 := uiSetDintDTSecond 50 ,
           xDone TRUE => bSetDintDTDone TRUE ,
           xError FALSE => bSetDintDTerr FALSE ,
           diErrorID 0 => diSetDintDTerrID 0 );

```

**LD:**


### 7.2.11 LS\_SetStringDT

Set the PLC system time (Dint type) and write the information in the internal attribute variable to the PLC system time.

#### ⊙ Command Format

Instruction	Name	FB/FC	LD	ST	File
LS_SetDintDT	Set system time(Dint type) instruction	FB		LS_SetStringDT( xExecute=, diDateAndTime=, xDone=>, xError=>, diErrorID=>);	MC_SysLib

#### ⊙ Related Variables

##### Input variable

Input variable	Name	Data type	Range	initialization	Descriptive
xExecute	Active High	BOOL	TRUE-FALSE	FALSE	TRUE: Activates processing of the function block.
diDateAndTime	Date and time to be set	DATE_AND_TIME	Follow the data type	2020-4-17-10:10:10	Date and time to be set.

##### Output variable

Output variable	Name	Data type	Range	initialization	Descriptive
xValid	Output valid	BOOL	TRUE-FALSE	FALSE	TRUE: Read enable.
xError	Error	BOOL	TRUE-FALSE	FALSE	FALSE - no error. TRUE - execution error.
diErrorID	Error ID	DINT	Follow the data type	0	Error code.

#### ⊙ Functional Description

Setting xEnable to TRUE puts the function block into a runnable state; setting xEnable to FALSE keeps the function block from running.

The diDateAndTime input format is as DT#2022-8-13-10:20:43.

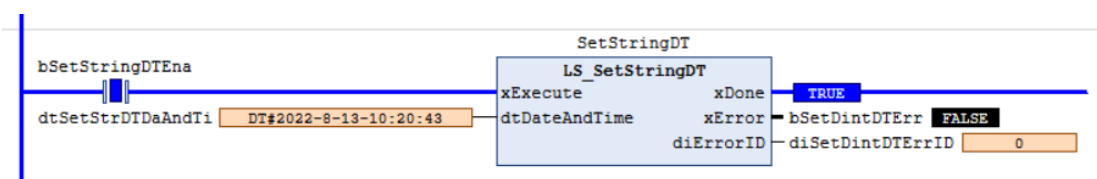
#### ⊙ Program demo

ST:

```

SetStringDT(xExecute TRUE := bSetStringDTEna TRUE,
dtDateAndTime DT#2022-8-13-10:20:43 := dtSetStrDTDaAndTi DT#2022-8-13-10:20:43,
xDone TRUE => bSetStrDTDone TRUE,
xError FALSE => bSetStringDTErr FALSE,
diErrorID 0 => diSetStringDTErrID 0);
RETURN
  
```

LD:



# Contact us

## Leadshine Technology Co., Ltd.

Headquarters Address:

15-20/F, Block B, Nanshan i-Valley, No.3157, Nanshan District,  
Shenzhen City, Guangdong Province, China

Tel: +86 755 26411692

Fax: +86 755 26402718

Website: [www.leadshine.com](http://www.leadshine.com)

Emails: [sales@leadshine.com](mailto:sales@leadshine.com)

### Leadshine Global Retailers Network



Get in touch with us or any of your local Leadshine certified retailers by visiting our global website.

Technical Support

Tel: 86-755-2641-8447

86-755-2641-8774 (Asia, Australia, Africa)

86-755-2665-5136 (North and South America)

86-755-8654-2465 (Europe)

Fax: 86-755-2640-2718

Email: [tech@leadshine.com](mailto:tech@leadshine.com)

Author Email: [Sean@leadshine.com](mailto:Sean@leadshine.com)

Sales Hot Line

Tel: 86-755-2641-7674 (Asia, Australia, Africa)

86-755-2641-7617 (North and South America)

86-755-2640-9254 (Europe)

Email: [sales@leadshine.com](mailto:sales@leadshine.com)

Leadshine  
overseas



Leadshine  
America, Inc.

North America Office

Address: 26050 Towne  
Centre Dr. Foothill  
Ranch California  
United States

Tel: 1-949-608-7270

Fax: 1-949-638-7298

Website:

[www.leadshineusa.com](http://www.leadshineusa.com)

Emails:

[sales@leadshineusa.com](mailto:sales@leadshineusa.com)