TS600 Series Programmable Logic Controller

Command Manual

SHENZHEN INVT ELECTRIC CO., LTD.

No.	Change Description	Version	Release Date
1	First release.	V1.0	March 2024

Preface

Data Introduction

The TS600 series PLC is a small compact and high-performance PLC with a full function model that supports EtherCAT master stations. Its body comes with 16 inputs and 16 outputs. The TS600 series PLC meets various needs of users for small and medium-sized automated devices, and applies to suitable for scenarios such as demanding volume, multi-axis operation control, temperature control, and communication networking.

This manual introduces basic commands and command examples, as well as complex application commands and command examples, which are used in product programming applications.

Targeted Readers

This manual applies to the following readers:

- Electrical engineers
- Software engineers
- Application engineers

Initial Use

The users using this product for the first time should read this manual carefully first. If you have any doubts about some specific functions and performance, feel free to consult our technical support personnel for assistance, which is beneficial for the correct use of this product.

List of Related Manuals

Manual Type	Manual Name	Manual Version
User's Manual	TS600 Series Programmable Logic Controller User Manual	V1.0
Programming and Application Manual	TS600 Series Programmable Logic Controller Programming and Application Manual	V1.0

This manual is not delivered along with the product. To obtain an electronic version of the PDF file, you can:

Log in to the official website of INVT at www.invt.com to download PDF files.

Contents

1 Command Overview

1.1 Command Composition

Commands consist of command symbol codes and operands. See below for meanings of command symbol codes and operands.

- Command symbol number: The description of a command function.
- Command operand: The data used in a command.

Command operands includes input data, output data, and constant numerical data.

Input Data:

Input data are data used in operations, and commands read their data for operation processing. In the command instructions, a single input data is represented by S. If there are more than one input data, they are represented by S1, S2, S3, etc, respectively. According to variables and soft elements specified in individual commands, the use of input data is listed as follows.

Table 1-1 Use of Input Data

Category	Description
Constant	It specifies a numerical value used in the operation. It cannot be changed during program execution, since it is set during program creation.
Soft element and variable	During program execution, the data used in the command can be changed by changing the data stored in the specified software element.

Output Data:

Commands control or output the data of output operands In the command instructions, a single output data is represented by D. If there are more than one output data, they are represented by D1, D2, etc, respectively. In addition to bit elements, operands have single-word or double-word elements and also constants.

Example - Block Transfer Command:

BMOV	S	D	n

①

① The data to be transmitted is specified through the BMOV command.

1.2 List of Soft Elements and Variables

Bit soft elements, word soft elements, special soft elements, variables, arrays, structures, and custom variables are supported. See below for details:

Bit Soft Elements

Type	Range	Points	Data Type	Description
X	X0~X1777	1024, encoded on the octal basis	BOOL	Unsaved after power down
Y	Y0~Y1777	1024, encoded on the octal basis	BOOL	Unsaved after power down
M	M0~M32767	32768	BOOL	M0~M999 unsaved after power down,

Type	Range	Points	Data Type	Description
				M1000 and later saved after power down
S	S0~S4095	4096	BOOL	S0~S999 unsaved after power down, S1000 and later saved after power down
LM local auxiliary relay	LM0~LM63	64	-	-

Word Soft Elements

Type	Range	Points	Data Type	Description
D	D0~D32767	32768	BOOL/INT/WORD/DWORD/DINT/REAL	D0~D999 unsaved after power down, D1000 and later saved after power down
R	R0~R32767	32768	BOOL/INT/WORD/DWORD/DINT/REAL	R0~R999 unsaved after power down, R1000 and later saved after power down
V local data register	V0~V63	64	-	-
Z indexed addressing register	Z0~Z15	16	-	-

Custom Variable

Type	Capacity	Data Form	Description
BOOL	2 MB (8 bits)	Variable, array, structure	256 KB saved after power down, others unsaved after power down
INT			
DINT			
WORD			
DWORD			
REAL			

Special Soft Elements

Type	Function	Range	Points	Description
L	Jump tag	L0~L1023	1024	Used in conjunction with CJ and LBL commands
SBR	Subroutine label	SBR0~SBR63	64	Used in conjunction with the CALL command to call the SBR subroutine (whose properties can be set to normal and encrypted), jointly occupying the system program area capacity
Character	Character, string	-	-	Character or string, used as a command parameter

Special SM Elements

Special Soft Elements	Function Description	R/W Access Permission
SM0	Monitoring run bit, which is always ON in RUN state and always OFF in STOP state	R
SM1	Initial running pulse bit, which is set ON when the user program switches from STOP to RUN and set OFF after one running cycle	R
SM2	Power-on flag bit, which is set ON when the system is energized and set OFF after the user program has run for one cycle	R

Special Soft Elements	Function Description	R/W Access Permission
SM3	Set ON after power-on or a system error is detected when he user program switches from STOP to RUN, or reset to zero if no system error occurs	R
...	-	-
SM10	Clock oscillation with a cycle of 10 ms (flipped in half a cycle, with the first half cycle set OFF when the user program runs)	R
SM11	Clock oscillation with a cycle of 100 ms (flipped in half a cycle, with the first half cycle set OFF when the user program runs)	R
SM12	Clock oscillation with a cycle of 1 s (flipped in half a cycle, with the first half cycle set OFF when the user program runs)	R
SM13	Clock oscillation with a cycle of 1 min (flipped in half a cycle, with the first half cycle set OFF when the user program runs)	R
SM14	Clock oscillation with a cycle of 1 hour (flipped in half a cycle, with the first half cycle set OFF when the user program runs)	R
SM15	Scan cycle oscillation bit, which is flipped once per scan cycle (with the first cycle set OFF when the user program runs)	R
...	-	-
SM18	Operation zero flag	R
SM19	Operation borrow flag	R
SM20	Operation carry flag	R
SM22	Bit set for command execution error	R
SM23	Bit set for overflow of command element number subscript	R
SM24	Bit set for illegal command parameter	R
...	-	-
SM30	Flag for multi-cycle command completion	R
SM31	Flag for BINDA command output character	R/W
SM32	Flag for processing mode of ATI/ITA/ASC/CCITT/CRC16/LRC/CCD command bit	R/W
SM33	SORTR/SORTC command descending sort enable	R/W
SM34	Bit for data format settings of SMOV command	R/W
SM35	Flag for all comparison results of BKCMP command matrices being 1	R

Description of Access Permissions:

- Read only: The output controlled by the PLC is only readable but not writable by users.
- Read and write: The input controlled by the PLC is both readable and writable by users.

1.3 Soft Elements

1.3.1 Bit Soft Elements

This PLC provides programming supports for bit soft elements, whose specific types, ranges, number of points, and related descriptions are shown in the following table:

Type	Range	Points	Data Type	Description
X	X0~X1777	1024, encoded on the octal basis	BOOL	Input
Y	Y0~Y1777	1024, encoded on the octal basis	BOOL	Output
M	M0~M32767	32768	BOOL	M0~M999 unsaved after power down, M1000 and later saved after power down

Type	Range	Points	Data Type	Description
S	S0~S4095	4096	BOOL	S0~S999 unsaved after power down, S1000 and later saved after power down

Note: The power-down keeping range cannot be changed.

1.3.2 Word Soft Elements

This PLC provides programming supports for word soft elements, whose specific types, ranges, number of points, and related descriptions are shown in the following table:

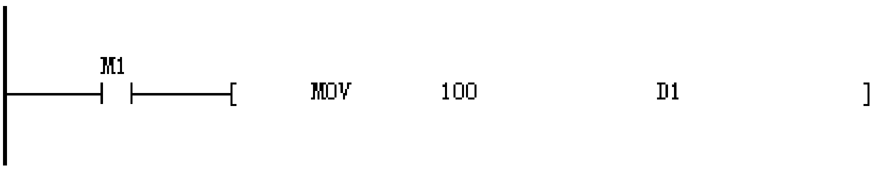
Type	Range	Points	Data Type	Description
D	D0~D32767	32768	BOOL/INT/DINT/WORD/ DWORD/REAL	D0~M999 unsaved after power down, M1000 and later saved after power down
R	R0~R32767	32768	BOOL/INT/DINT/WORD/ DWORD/REAL	R0~R999 unsaved after power down, R1000 and later saved after power down
T	T0~T399	400	INT	T0 ~ T199: 100ms accuracy T200 ~ T299: 10ms accuracy T300 ~ T399: 1ms accuracy
C	C0~C235	256	INT/DINT	C0 ~ C199: 16-bit increment counter or 16-bit loop counter C200 ~ C255: 32-bit CTUD

Note:

- The saving range after power down cannot be changed.
- Word soft elements can be used as integers or floating-point numbers. The soft elements themselves do not have data type attribute, and the elements are interpreted as integers or floating-point numbers according to the parameter attributes of instructions.
- When interpreted as integers, word soft elements can be used as 16-bit or 32-bit data depending on command parameters. When used as a 16-bit data, a word soft element occupies 1 soft element; when used as a 32-bit data, it occupies 2 soft elements. When interpreted as a floating-point number, every word soft element occupies 2 soft elements.

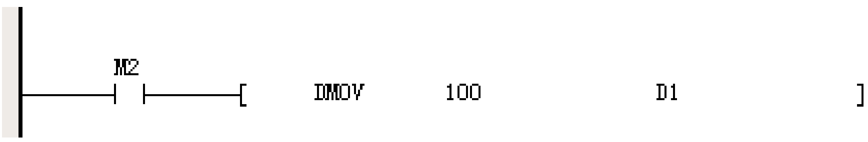
Example

1. Use word soft elements as 16-bit integers.



Using a 16-bit assignment command, a value of 100 is assigned to the word soft element D1, which occupies the soft element D1.

2. Use word soft elements as 32-bit integers.



Using a 32-bit assignment command, a value of 100 is assigned to the word soft element D1, which occupies the soft elements D1 (low bit) and D2 (high bit).

3. Use word soft elements as floating-point numbers.

Using a floating-point number command, a value of 100 is assigned to the word soft element D1, which occupies the soft elements D1 and D2.

1.3.3 Special Soft Elements

This PLC provides programming supports for special soft elements, whose specific functions, ranges, and related descriptions are shown in the following table:

Type	Function	Range	Points	Description
SBR	Subroutine label	SBR0–SBR1023	1024	Used by the CALL command to call the SBR subroutine (whose properties can be set to normal and encrypted), jointly occupying the system program area capacity
L	Jump tag	L0~L1023	1024	Used in conjunction with CJ and LBL commands
Constant	Decimal	-32768~32767 (16-bit), -2147483648~2147483647 (32-bit)	-	-
Character	Character, string	-	-	Character or string, used as a command parameter

Special Soft Elements	Function Description	R/W Access Permission
SM0	Monitoring run bit, which is always ON in RUN state and always OFF in STOP state	R
SM1	Initial running pulse bit, which is set ON when the user program switches from STOP to RUN and set OFF after one running cycle	R
SM2	Power-on flag bit, which is set ON when the system is energized and set OFF after the user program has run for one cycle	R
SM3	Set ON after power-on or a system error is detected when he user program switches from STOP to RUN, or reset to zero if no system error occurs	R
...	-	-
SM10	Clock oscillation with a cycle of 10 ms (flipped in half a cycle, with the first half cycle set OFF when the user program runs)	R
SM11	Clock oscillation with a cycle of 100 ms (flipped in half a cycle, with the first half cycle set OFF when the user program runs)	R
SM12	Clock oscillation with a cycle of 1 s (flipped in half a cycle, with the first half cycle set OFF when the user program runs)	R
SM13	Clock oscillation with a cycle of 1 min (flipped in half a cycle, with the first half cycle set OFF when the user program runs)	R
SM14	Clock oscillation with a cycle of 1 hour (flipped in half a cycle, with the first half cycle set OFF when the user program runs)	R
SM15	Scan cycle oscillation bit, which is flipped once per scan cycle (with the first cycle set OFF when the user program runs)	R
...	-	-

Special Soft Elements	Function Description	R/W Access Permission
SM18	Operation zero flag	R
SM19	Operation borrow flag	R
SM20	Operation carry flag	R
...	-	-
SM22	Bit set for command execution error	R
SM23	Bit set for overflow of command element number subscript	R
SM24	Bit set for illegal command parameter	R
...	-	-
SM30	Flag for multi-cycle command completion	R
SM31	Flag for BINDA command output character	R/W
SM32	Flag for processing mode of ATI/ITA/ASC/CCITT/CRC16/LRC/CCD command bit	R/W
SM33	SORTR/SORTC command descending sort enable	R/W
SM34	Bit for data format settings of SMOV command	R/W
SM35	Flag for all comparison results of BKCMP command matrices being 1	R

1.3.4 Bit Operation of Word Elements

Bit operation of word elements can be done by (.). For example, D100.8 means operation shall be done to the 8th bit on D100 word element, and the lowest bit is the 0th bit.

For example:

D100.8

1	1	0	0	1	1	0	0	1	1	0	0	1	1	0	0
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

The bit count of a word element starts from the 0th bit: D100.8 can be seen as a BOOL element, which is suitable for bit operation commands.

1.4 Variable

1.4.1 Custom Variable

In programming engineering, in addition to directly using direct addresses such as X, Y, M, D, R, and other elements for programming, programming can also be done in the form of variables without specific storage addresses to achieve the required control logic or machining control processes. This can both improve the efficiency of code writing and enhance the code readability.

Table 1-2 Supported Custom Variables

Type	Capacity	Data Form	Description
BOOL	2 MB (8 bits)	Variable, array, structure	256 KB saved after power down, others unsaved after power down
INT			
DINT			
WORD			
DWORD			
REAL			

1.4.2 Defining Variables

This series PLC supports custom variables, and users can directly use variable names in programs for programming by defining global variables. You need to follow the following rules when defining a global variable name:

1. It can only contain "_, letters, numbers, Chinese characters" and cannot start with "_, numbers".
2. Global variables cannot have the same name as "soft element forms, constants, standard data types, and commands".
3. Global variable names cannot be keywords such as "ARRAY, TRUE, FALSE, ON, OFF, and NULL".

Variable Data Type

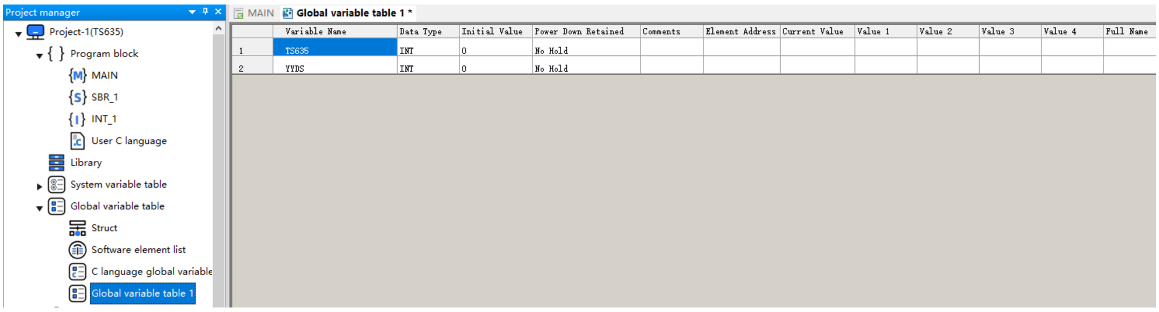
Variable definitions support structures and arrays, and supported variable data types are listed as follows:

Table 1-3 Supported Variable Data Types

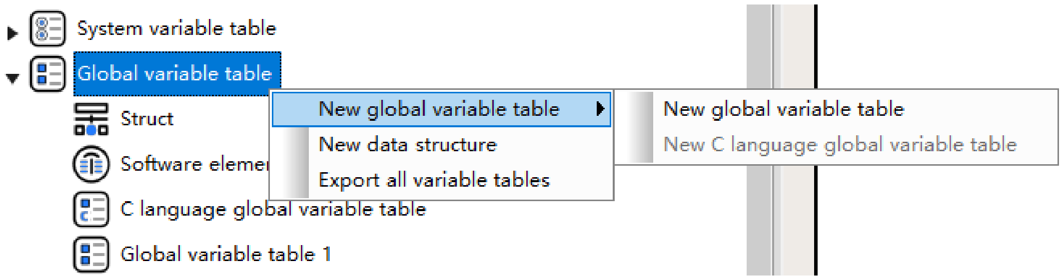
Data Type	Description
BOOL	Boolean
INT	Single-word integer
DINT	Double-word integer
WORD	Single-word unsigned integer
DWORD	Double-word unsigned integer
REAL	Real number

Defining Global Variables

"Global Variable table" in the engineering management column of the programming software Auto Station Pro can be used for variable management to add, delete, and edit variables.



1. Adding a variable table and variables: Right click on "Global Variable table" and select "New global variable table" to create a new global variable table.



2. Double click on the variable table to enter the variable editing interface.

(1) In the variable table, right click on the pop-up menu to insert or delete variables.

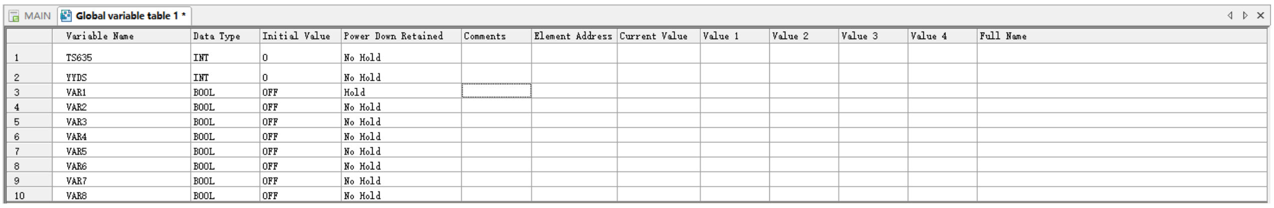
(2) If a custom variable name is entered in the "Variable name" column of the variable table, you can directly use the variable name for programming.

(3) "Data Type" can be set to BOOL, INT, DINT, WORD, DWORD, REAL, array, or structure (structures need to be defined in advance). When selecting an array as the data type, you can set the type and length

of the array variable in the pop-up dialog box. A structure variable can be defined by selecting the structured defined in advance.

(4) The "Initial Value" column defines initial values for variables. For arrays and structures, the initial value of each element can be defined separately.

(5) "Power Down Retained" can be set to either "Hold" or "No Hold", and the settings of the initial values are valid to only non-holding variables.

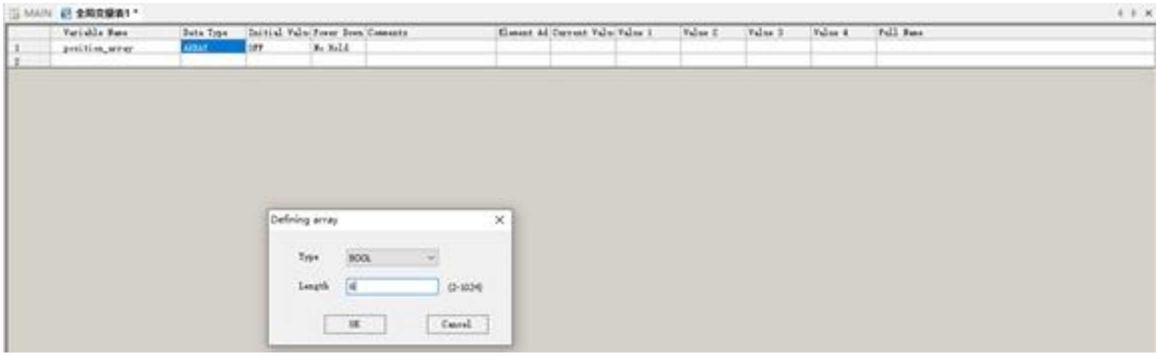


	Variable Name	Data Type	Initial Value	Power Down Retained	Comments	Element Address	Current Value	Value 1	Value 2	Value 3	Value 4	Full Name
1	TS635	INT	0	No Hold								
2	YYDS	INT	0	No Hold								
3	VAR1	BOOL	OFF	Hold								
4	VAR2	BOOL	OFF	No Hold								
5	VAR3	BOOL	OFF	No Hold								
6	VAR4	BOOL	OFF	No Hold								
7	VAR5	BOOL	OFF	No Hold								
8	VAR6	BOOL	OFF	No Hold								
9	VAR7	BOOL	OFF	No Hold								
10	VAR8	BOOL	OFF	No Hold								

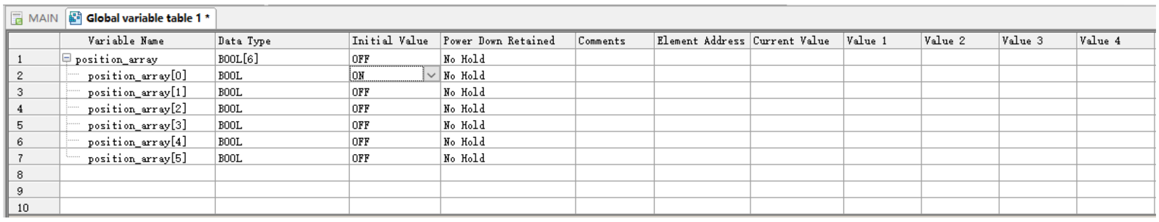
1.4.3 Defining Arrays

Users can define arrays if ARRAY is chosen as the data type when programming.

1. To define an array, select the type and length of the array variable in the pop-up dialog box, and click "OK".



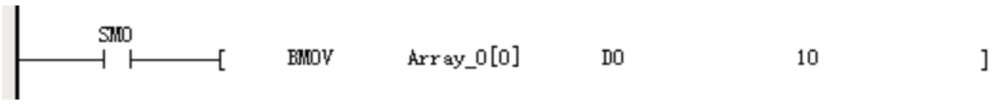
2. Click on the "Initial Value" column of the array variable to initial value setting interface of the array variable.



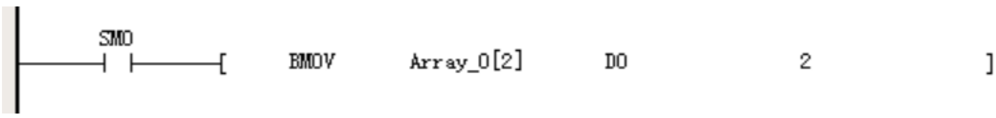
	Variable Name	Data Type	Initial Value	Power Down Retained	Comments	Element Address	Current Value	Value 1	Value 2	Value 3	Value 4
1	position_array	BOOL[6]	OFF	No Hold							
2	position_array[0]	BOOL	ON	No Hold							
3	position_array[1]	BOOL	OFF	No Hold							
4	position_array[2]	BOOL	OFF	No Hold							
5	position_array[3]	BOOL	OFF	No Hold							
6	position_array[4]	BOOL	OFF	No Hold							
7	position_array[5]	BOOL	OFF	No Hold							
8											
9											
10											

3. When using an array in a command, if you do not enter an array subscript, access starts from the first element of the array. If you enter an array subscript, access starts from the element specified by the subscript. For example:

1) Assign the 10 elements from Array_0[0] through Array_0[9] to D0~D9.



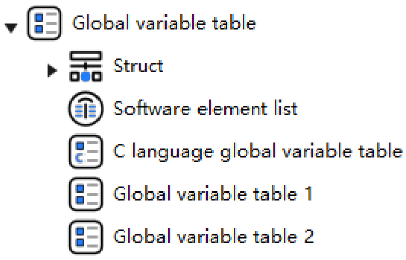
2) Assign the 2 elements from Array_0[2] through Array_0[3] to D0~D1.



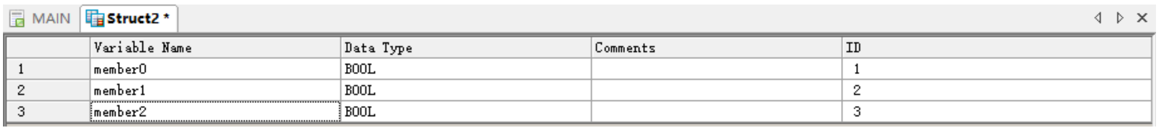
1.4.4 Defining Structure

During variable definition, if it is necessary to define a structural variable, you should define the data structure of the structure variable in advance.

1. Right click on "Struct" under "Global Variable table", select "New date structure", and enter a structure name to define the structure.
2. When defining a variable in the variable table, you can select the type of this structure as the data type of the variable to define the variable as a structure variable.



3. After establishing structure and member variables, you can select a structure from the data type defined by the variable to define a structure variable.



	Variable Name	Data Type	Comments	ID
1	member0	BOOL		1
2	member1	BOOL		2
3	member2	BOOL		3

4. Click on the "Initial Value" column of structure variables to enter the initial value setting interface of structure variables, where you can set initial values of structure variable members.

1.4.5 How to Use Variables

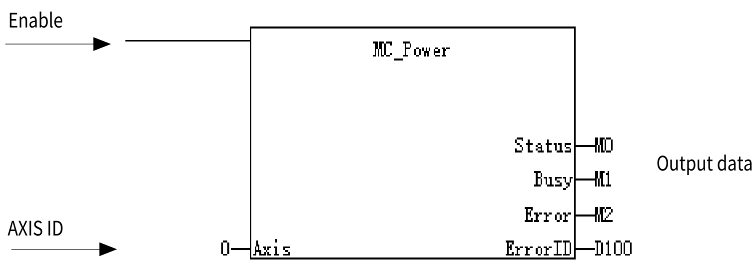
After defining variables, variable names can be directly used for programming variables, eliminating the need to allocate soft elements.

- When using array variables, arrays in programming are represented by "[numbers]", which start from 0.
- When using structure variables, every structure member in programming is represented by "structure variable name. member variable".

1.5 Special Functions

1.5.1 Graphic Block Commands

Some commands of this PLC support graphic block programming, and graphic block commands consist of command names, energy flow signals, input sides, and output sides. Taking the graphic block command of a motion control axis as an example, its specific composition is shown in the figure below.



When programming, after entering the name of the graphic block command, you can add the graphic block command to the program network by simply pressing the "ENTER" key. In the input parameter command of the graphic block command, the item displayed as "???" is a mandatory parameter, which must be assigned with a value. For a non-mandatory parameter, the command input automatically defaults to the parameter value in the command, and the command output cannot obtain the status in the command during programming or monitoring debugging.

Graphic blocks support directly double clicking on any graphic block command under the toolbox command set node during programming, and you can drag the graphic block command with the left mouse button to add it to the current focus position of the ladder diagram.

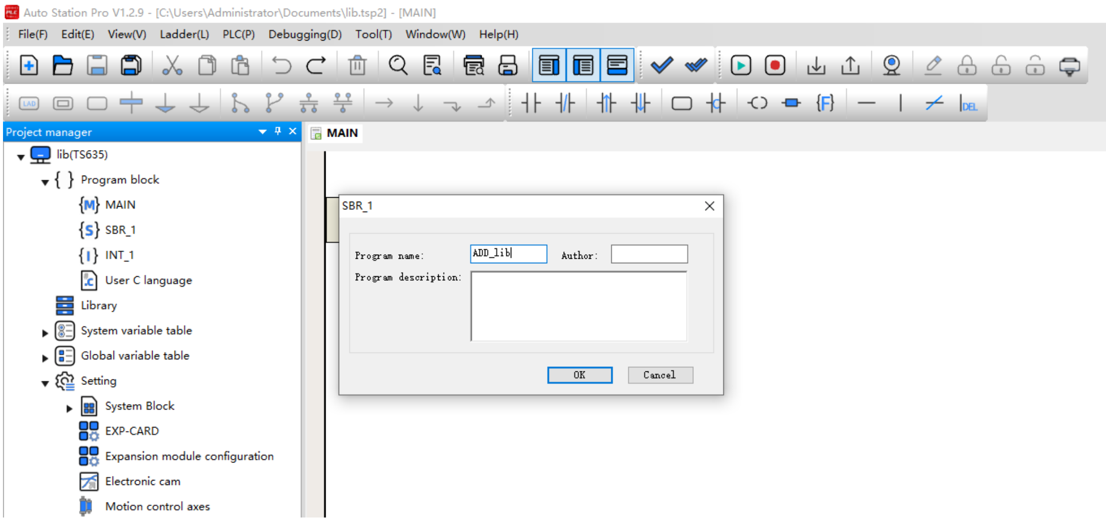
1.5.2 Library Functions

Library functions abstract and encapsulate reusable parts of a program into a universal program block, which can be repeatedly called in the program. Using encapsulated libraries in programming enhances program development efficiency, reduces programming errors, and improves program quality.

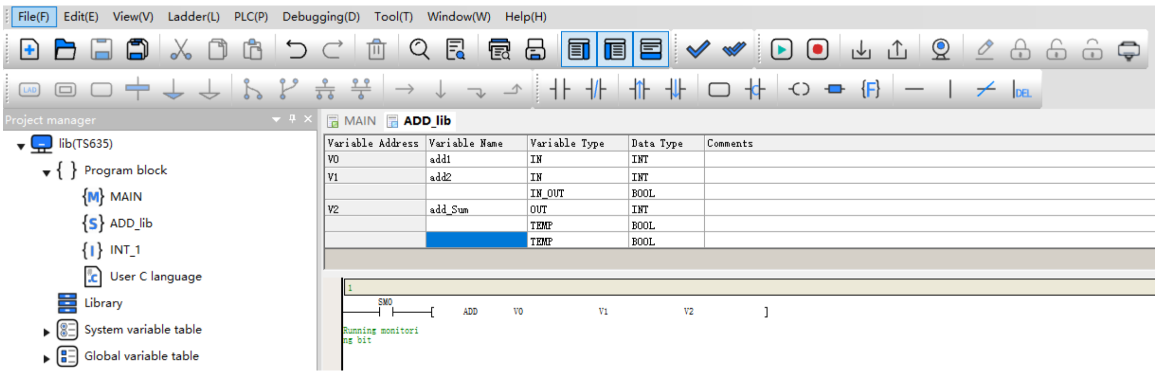
The basic step to use library functions is as follows: Create a library project to be encapsulated -> Write the program -> Export a library file -> Create current project -> Import the library file.

See below for an example of exporting an addition library:

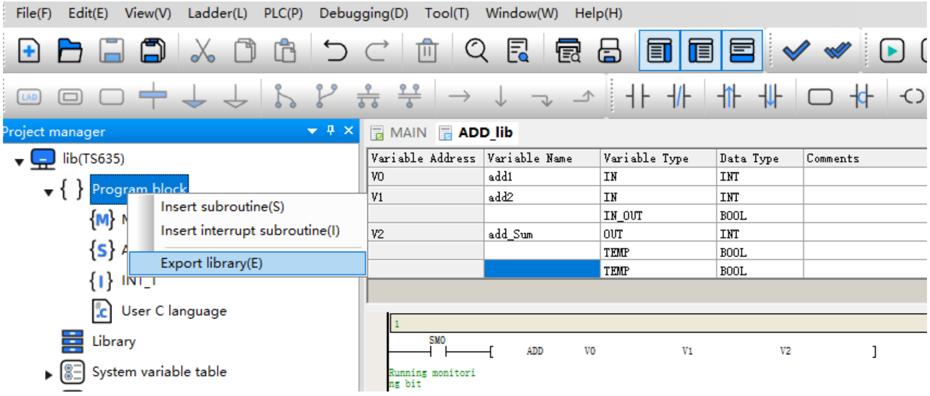
Step 1 Create a new addition library project and then a new subroutine Add_lib.



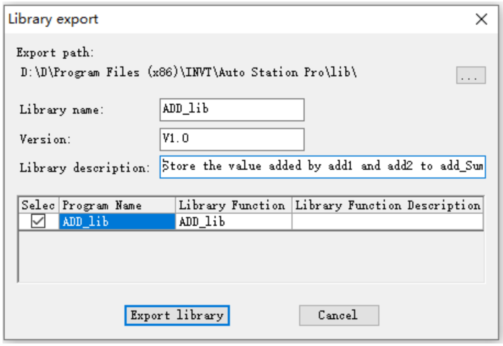
Step 2 Write the input variables add1 and add2, then the output variable add_Sum, and finally the core code.



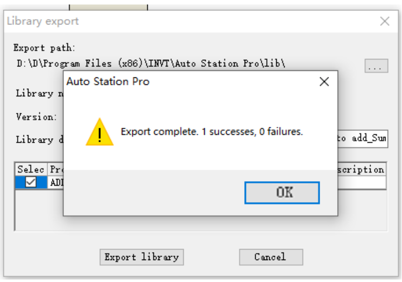
Step 3 Right click on "Program block" in the "Project manager" column, and select "Export library".



Step 4 Select the save path, fill in the "Library name", "Version", and "Library description" fields, and select the subroutine to save.

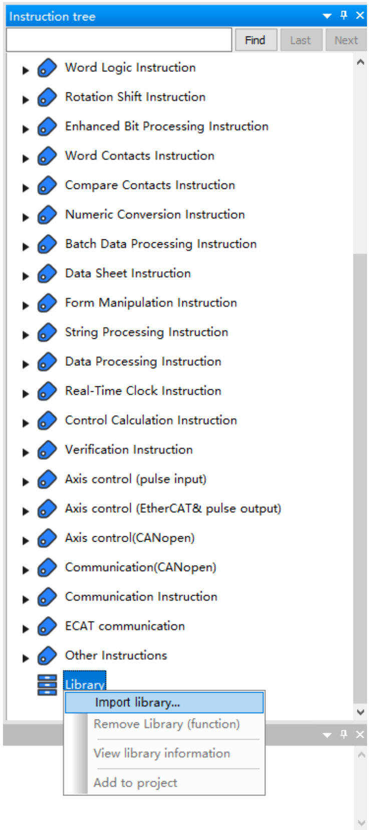


Step 5 Export is completed.

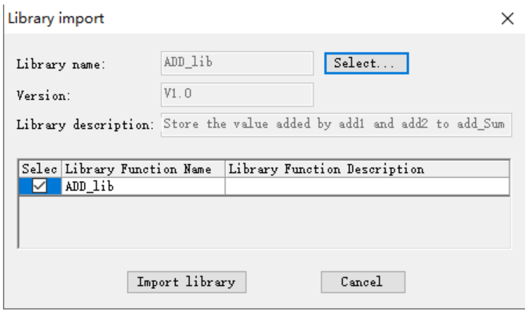


See below for an example of importing an addition library:

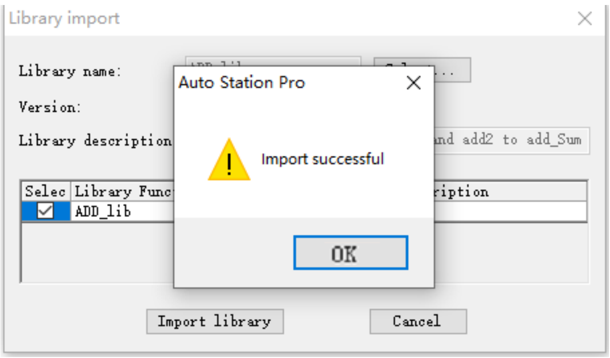
Step 1 Open the project that requires the use of library functions, right click on "Library" at the bottom of the "Instruction tree" column, and select "Import library".



Step 2 Select the library save path (which is the lib file under the installation path of the host computer software by default), then select the load library function, and finally click "Import library".

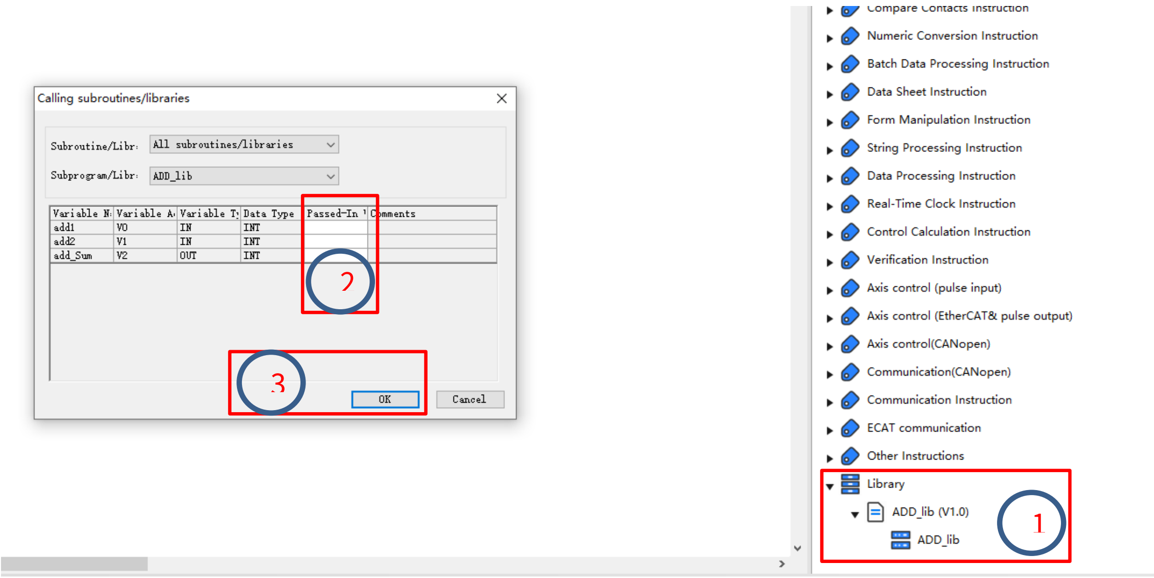


Step 3 Import is successful.

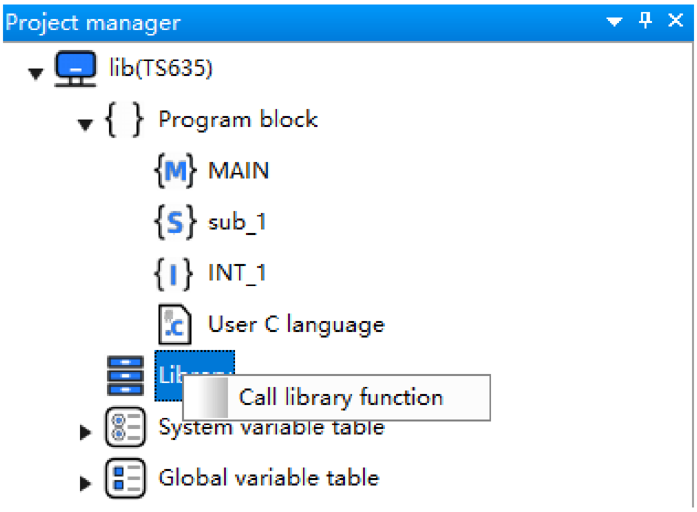


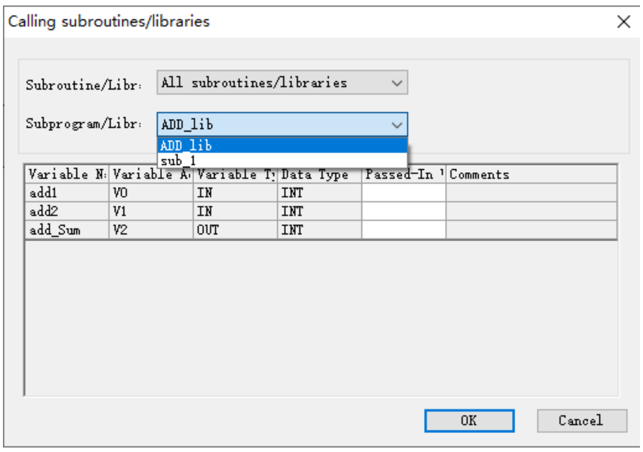
There are two ways to call library functions:

Step 1 In the "Instruction tree" column, select the newly imported library function, double click on the functional function or drag it into the project, and fill in the value passed in.

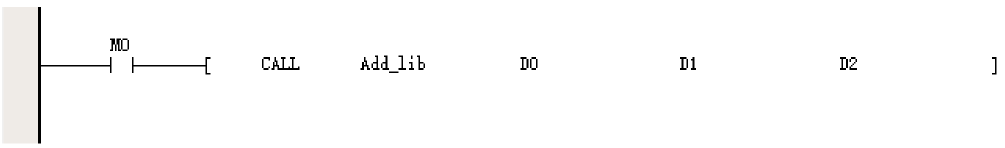


Step 2 In the "Project manager" column, right click on "Library", select "Subroutine/Libr" and " Subprogram/Libr", fill in the "Passed-In Value" field, and click "OK".

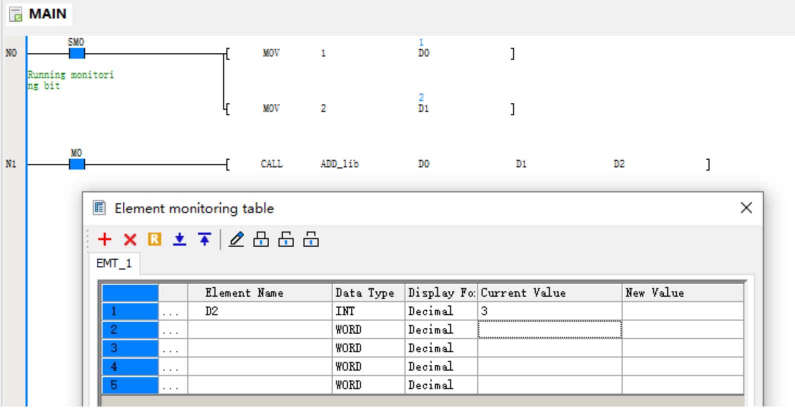




Using either of the above two methods generates the following ladder diagram code, which indicates a successful library import.



If D0=1 and D1=3 are assigned, D2=4 can be obtained, as shown in the figure below.



To update the library, you need to delete original library functions, including those under "Library" in "Project manager" and "Instruction tree", and then re-import the library file.

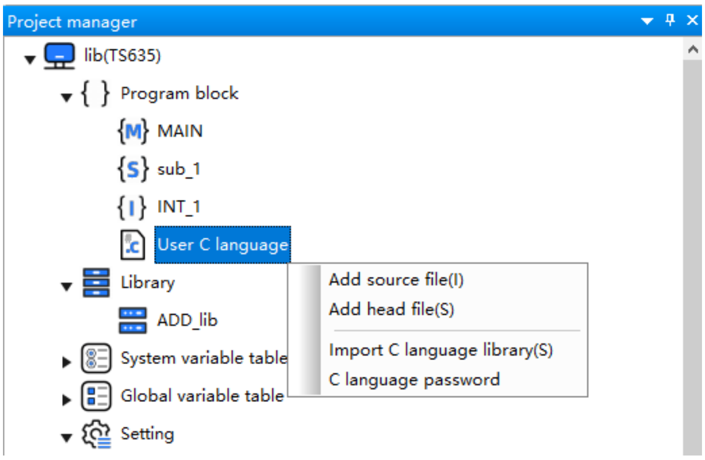
1.5.3 C Language Functions

The TS series PLC allows users to use C language to write function blocks in programming software, calls them where needed, supports commonly used C language attribute libraries, and uses the CALL command to call C language in ladder diagrams. Parameters can be passed to bit elements M, word elements D, and word elements R to read and write element values. By replacing complex logic, arithmetic operations, and other functions implemented in other programming languages of PLC, C language enables programmer to remarkably improve development efficiency. Here is a brief explanation of the creation and use of C language. For specific usage methods, refer to Chapter 4 of the Programming and Application Manual.

Operation Steps:

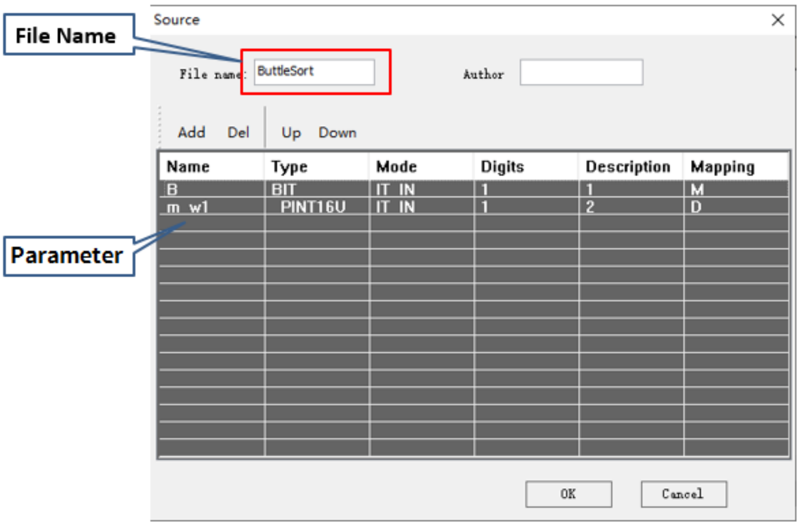
Step 1 Create C language

Open the PLC editing software, choose "User C language" node in the "Project manager" toolbar on the left, right-click and select "Add source file", and the software interface will pop up the user C language interface design window.



Step 2 Design C language interface

In the C language source file interface design dialog box, fill in the user C language function information. The function name is a mandatory item and cannot be the same as names of subroutines, interrupt subroutines, and other C language functions; the use of function names that include strings SBR_ and INT_ should be avoided; the addition of up to 16 parameters are supported; parameters cannot be empty; parameter names cannot be duplicate; and PLC soft element names cannot be used.

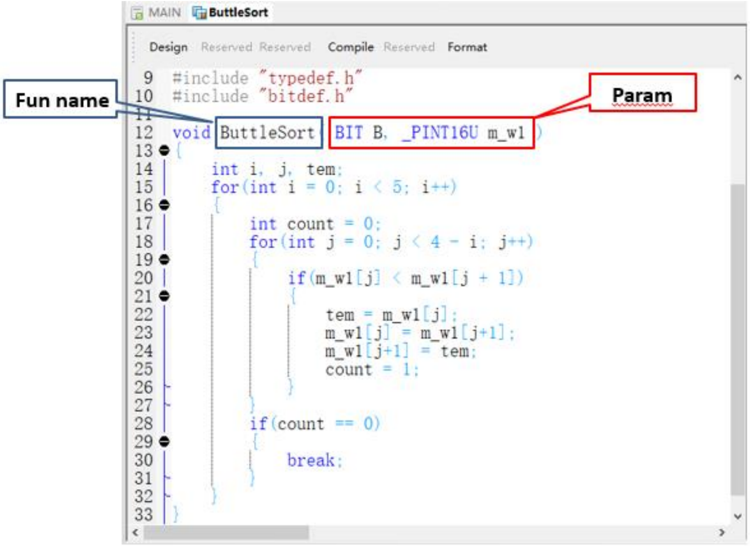


Supported Data Types:

Type	Description
BIT	Boolean quantity
_INT16U	16-bit unsigned integer
_INT16S	16-bit signed integer
_INT32U	32-bit unsigned integer
_INT32S	32-bit signed integer
_FP32	32-bit floating point
_PINT16U	16-bit unsigned int pointer
_PINT16S	16-bit signed int pointer
_PINT32U	32-bit unsigned int pointer
_PINT32S	32-bit signed int pointer
_PFP32S	32-bit float pointer

Step 3 C language editing

After creation, enter the C language editing interface where users can write the functions needed to realize. The default generated part includes the contained header files (the three header files plcstdafx. h, typedef. h, and bitdef. h are contained by default) and C language interface function body. Users do not need to manually change or delete default header files, function interface names, return value types, and function parameters, otherwise compilation errors will be caused. After you click the design button and re-edit the interface design, this part will be reproduced, and the previous section needs to be manually deleted to avoid compilation errors. In the example, the function is of C language bubble sort algorithm.



Parameter passing mode: When ladder diagram is called, the passed-in M and D are the start addresses of B and m_w1. As shown in the above figure, if the elements in the command ButtleSort are M0 and D0, then in the C language function, B[0] is M0, B[10] is M10, m_w1[0] is D0, m_w1[10] is D10; if the parameters used in the ladder diagram are M100 and D100, then B[0] is M100, and m_w1[0] is D100.

Double-word operation: A D is added before m_w1, as in Dm_w1[10]=100000, which means to assign a value to the combined doubleword m_w1[10] m_w1[11].

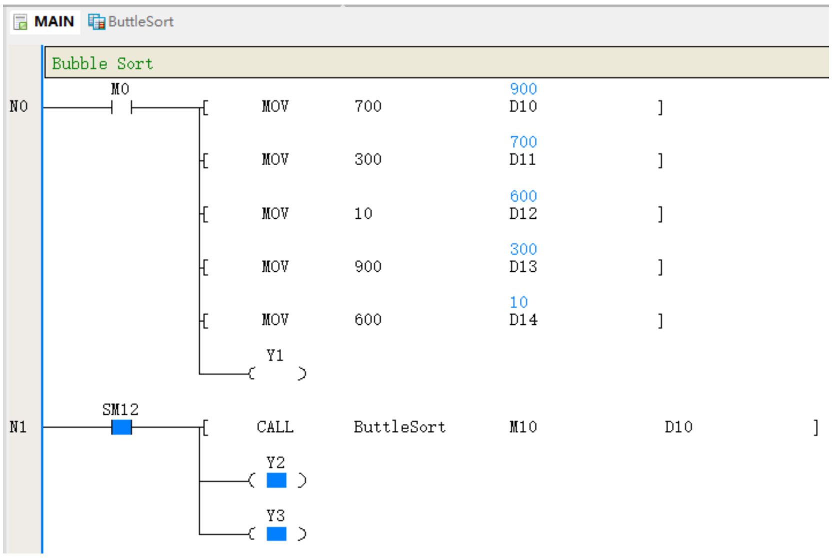
Floating-point operation: This PLC supports defining floating-point variables in functions and performing floating-point operations (for example, the floating-point register D0 (doubleword) can be represented as Fm_w1[0], Fm_w1[0]=100.01).

Step 4 Using C Language Programs

User C language functions are called by CALL commands. For example, for the bubble sort function mentioned above, the following shall be input in ladder diagram:

- CALL Command
- Function Name (ButtleSort)
- Parameters (M10...D10)

When compiling a ladder diagram, check the command block. If it is a CALL command, check whether its function name is a C language function (by distinguishing between ladder subroutines and interrupt subroutines). If it is a C language function, check parameters and match the types and quantity of parameters. If no error is found during the command block check is correct, compile the C language file to generate an executable file.


MAIN
ButtleSort
Bubble Sort
N0
M0
MOV 700 D10
900
MOV 300 D11
700
MOV 10 D12
600
MOV 900 D13
300
MOV 600 D14
10
Y1
N1
SM12
CALL ButtleSort M10 D10
Y2
Y3

2 Command Cheat Sheet

All commands supported by this PLC are summarized in the command cheat sheet and classified according to the corresponding function categories.

Table 2-1 Command Cheat Sheet

Command Category	Name	Function
Contact Logic Command	LD	Normally open contact
	LDI	Normally closed contact
	LDP	Take rising pulse edge
	LDF	Take falling pulse edge
	AND	Normally open contact AND
	ANI	Normally closed contact OR
	ANDP	Serial connection of AND rising pulse edge detection
	ANDF	Serial connection of AND falling pulse edge detection
	OR	Normally open contact OR
	ORI	Normally closed contact OR
	ORP	Serial connection of OR rising pulse edge detection
	ORF	Serial connection of OR falling pulse edge detection
	ANB	Energy flow block AND
	ORB	Energy flow block OR
	EU	Rising energy flow edge detection
	ED	Falling energy flow edge detection
Output Control Command	OUT	Coil output
	SET	Coil set
	RST	Coil reset
	PLS	Rising pulse edge detection coil
	PLF	Falling pulse edge detection coil
	ALT	Alternating output
	NOP	Null operation
Energy Flow Control Command	INV	Energy flow inversion
SFC Command	STL	SFC state load
	SET Sxx	SFC state transition
	OUT Sxx	SFC state jump
	RST Sxx	SFC state clear
	RET	SFC program segment end
Program Flow Control Command	FOR	Loop operation
	NEXT	Loop return
	LBL	Jump label definition
	CJ	Conditional jump
	CFEND	Conditional return of main user program
	WDT	User program watchdog reset
	EI	Interrupt enable
	DI	Interrupt disable
	CIRET	Conditional return of user interrupt

Command Category	Name	Function
		program
	STOP	User program stop
	CALL	User subroutine call
	CSRET	Conditional return of user subroutine
Timing and Counting Command	TON	ON delay timing
	TONR	Memory-type ON delay timing
	TOF	OFF delay timing
	TMON	Non-retriggering single stable timing
	CTU	16-bit increment counter
	CTR	16-bit loop counter
	DCNT	32-bit increment-decrement counter
Data Transmission Command	*MOV	Word/doubleword data transmission
	RMOV	Floating-point number data transmission
	BMOV	Block data transmission
	*FMOV	Data block word/doubleword stuffing
	SMOV	Word/doubleword shift transmission
	SWAP	High-low byte swap
	*XCH	Word/doubleword exchange
	PUSH	Data push
	FIFO	First in first out
	LIFO	Last in first out
	WSFR	Word string shift right
	WSFL	Word string shift left
Arithmetic Operation Command for Integers	*ADD	Integer/long integer addition
	*SUB	Integer/long integer subtraction
	*MUL	Integer/long integer multiplication
	*DIV	Integer/long integer division
	*SQT	Arithmetic square root of integer/long integer
	*INC	Integer/long integer increment by 1
	*DEC	Integer/long integer decrement by 1
	*VABS	Absolute value of integer/long integer
	*NEG	Integer/long integer negation
	*SUM	Integer/long integer accumulation
	*MEAN	Mean value of integers/long integers
Arithmetic Operation Command for Floating-Point Numbers	RADD	Floating-point number addition
	RSUB	Floating-point number subtraction
	RMUL	Floating-point number multiplication
	RDIV	Floating-point number division
	RSQT	Arithmetic square root of floating-point number
	RVABS	Absolute value of floating-point number
	RNEG	Floating-point number negation
	SIN	Sine operation of floating-point number
	COS	Cosine operation of floating-point number
	TAN	Tangent operation of floating-point number

Command Category	Name	Function
	POWER	Power operation of floating-point number
	LN	Natural logarithm operation of floating-point number
	EXP	Natural number power operation of floating-point number
	RSUM	Accumulation operation of floating-point number
	RMEAN	Mean operation of floating-point numbers
	ASIN	Anti-sine operation of floating-point number
	ACOS	Anti-cosine operation of floating-point number
	ATAN	Anti-tangent operation of floating-point number
	SINH	Hyperbolic sine operation of floating-point number
	COSH	Hyperbolic cosine operation of floating-point number
	TANH	Hyperbolic tangent operation of floating-point number
	LOG	Common logarithm operation of floating-point number
	RAD	Angle-to-radian conversion of floating-point number
	DEG	Radian-to-angle conversion of floating-point number
Word Logic Operation Command	*WAND	Word/doubleword AND operation
	*WOR	Word/doubleword OR operation
	*WXOR	Word/doubleword XOR operation
	*WINV	Word/doubleword negation operation
Bit shift rotation command	*ROR	16-bit/32-bit cyclic shift right
	*ROL	16-bit/32-bit cyclic shift left
	*RCR	16-bit/32-bit cyclic shift right with carry
	*RCL	16-Bit/32-bit cyclic shift left with carry
	*SHR	16-bit/32-bit shift right
	*SHL	16-bit/32-bit shift left
	SFTR	Bit string shift right
	SFTL	Bit string shift left
Enhanced Bit Processing Command	ZRST	Batch bit reset
	ZSET	Batch bit set
	DECO	Decode
	ENCO	Encode
	*BITS	ON bit statistics in word/doubleword
	BON	ON bit judgment in word
Word Contact Command	BLD	Contact command of word bit data
	BLDI	Inversion contact command of word bit data
	BAND	Serial connection to contact command of word bit data

Command Category	Name	Function
	BANI	Serial connection to inversion contact command of word bit data
	BOR	Parallel connection to contact command of word bit data
	BORI	Parallel connection to inversion contact command of word bit data
	LD*&	Logical AND operation of word/doubleword LD contact
	LD*|	Logical OR operation of word/doubleword LD contact
	LD*^	Logical XOR operation of word/doubleword LD contact
	AND*&	Logical AND operation of word/doubleword AND contact
	AND*|	Logical OR operation of word/doubleword AND contact
	AND*^	Logical XOR operation of word/doubleword AND contact
	OR*&	Logical AND operation of word/doubleword OR contact
	OR*|	Logical OR operation of word/doubleword OR contact
	OR*^	Logical XOR operation of word/doubleword OR contact
	BOUT	Word bit data coil output
	BSET	Word bit data coil set
	BRST	Word bit data coil reset
Contact Comparison Command	LD*=	Integer/long integer LD contact comparison equal to
	LD*>	Integer/long integer LD contact comparison greater than
	LD*<	Integer/long integer LD contact comparison less than
	LD*<>	Integer/long integer LD contact comparison not equal to
	LD*>=	Integer/long integer LD contact comparison greater than or equal to
	LD*<=	Integer/long integer LD contact comparison less than or equal to
	AND*=	Integer/long integer AND contact comparison equal to
	AND*>	Integer/long integer AND contact comparison greater than
	AND*<	Integer/long integer AND contact comparison less than
	AND*<>	Integer/long integer AND contact comparison not equal to
	AND*>=	Integer/long integer AND contact comparison greater than or equal to
	AND*<=	Integer/long integer AND contact comparison less than or equal to
	OR*=	Integer/long integer OR contact

Command Category	Name	Function
		comparison equal to
	OR*>	Integer/long integer OR contact comparison greater than
	OR*<	Integer/long integer OR contact comparison less than
	OR*<>	Integer/long integer OR contact comparison not equal to
	OR*>=	Integer/long integer OR contact comparison greater than or equal to
	OR*<=	Integer/long integer OR contact comparison less than or equal to
	LDR=	Floating-point number LD contact comparison equal to
	LDR>	Floating-point number LD contact comparison greater than
	LDR<	Floating-point number LD contact comparison less than
	LDR<>	Floating-point number LD contact comparison not equal to
	LDR>=	Floating-point number LD contact comparison greater than or equal
	LDR<=	Floating-point number LD contact comparison less than or equal
	ANDR=	Floating-point number AND contact comparison equal to
	ANDR>	Floating-point number AND contact comparison greater than
	ANDR<	Floating-point number AND contact comparison less than
	ANDR<>	Floating-point number AND contact comparison not equal to
	ANDR>=	Floating-point number AND contact comparison greater than or equal
	ANDR<=	Floating-point number AND contact comparison less than or equal
	ORR=	Floating-point number OR contact comparison equal to
	ORR<	Floating-point number OR contact comparison greater than
	ORR>	Floating-point number OR contact comparison less than
	ORR<>	Floating-point number OR contact comparison not equal to
	ORR>=	Floating-point number OR contact comparison greater than or equal
	ORR<=	Floating-point number OR contact comparison less than or equal
	CMP	Integer comparison set
	LCMP	Long integer comparison set
	RCMP	Floating-point number comparison set
	*ZCP	Word/doubleword data region

Command Category	Name	Function
		comparison set
	RZCP	Floating-point number region comparison set
Logic Contact Command	LD*&	Word/doubleword LD contact AND
	LD*|	Word/doubleword LD contact OR
	LD*^	Word/doubleword LD contact XOR
	AND*&	Word/doubleword AND contact AND
	AND*|	Word/doubleword AND contact OR
	AND*^	Word/doubleword AND contact XOR
	OR*&	Word/doubleword OR contact AND
	OR*|	Word/doubleword OR contact OR
	OR*^	Word/doubleword OR contact XOR
Numerical Conversion Command	DTI	Conversion from long integer to integer
	ITD	Conversion from integer to long integer
	*FLT	Conversion from Integer/long integer to floating-point number
	*INT	Conversion from floating-point number to Integer/long integer
	*BCD	Conversion from word/doubleword data to 16-bit/32-bit BCD code
	*BIN	Conversion from 16-bit/32-bit BCD code to word/doubleword data
	*GRY	Conversion from word/doubleword to 16-bit/32-bit Gray code
	*GBIN	Conversion from 16-bit/32-bit Gray code to word/doubleword data
	SEG	Conversion from word data to 7-segment code
	ITA	Conversion from 16-bit hexadecimal number to ASCII code
	ATI	Conversion from ASCII code to 16-bit hexadecimal number
	LCNV	Engineering conversion
	RLCNV	Floating-point number engineering conversion
	DABIN	Conversion from decimal ASCII code to integer/long integer
	BINDA	Conversion from integer/long integer to decimal ASCII code
Batch Data Processing Command	*BKADD	Addition operation of word/doubleword data block
	*BKSUB	Subtraction operation of word/doubleword data block
	*BKCMP=	Set word/doubleword data block comparison equal to
	*BKCMP>	Set word/doubleword data block comparison greater than
	*BKCMP<	Set word/doubleword data block comparison less than
	*BKCMP<>	Set word/doubleword data block

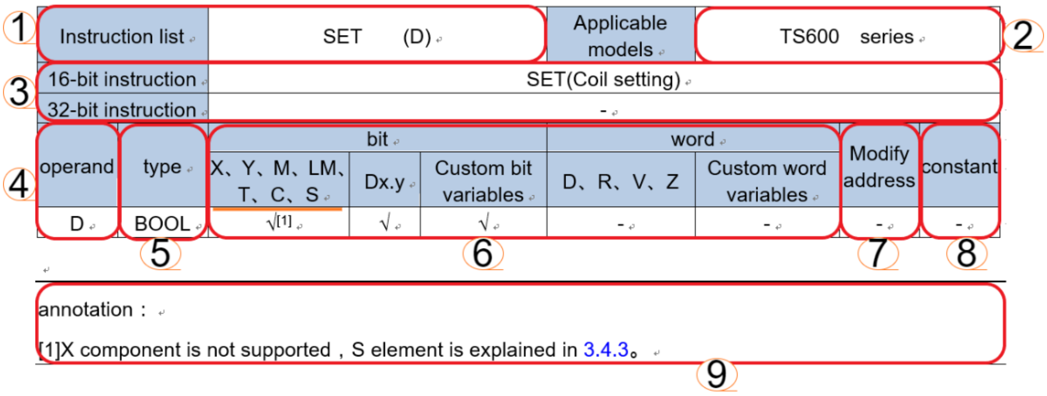
Command Category	Name	Function
		comparison not equal to
	*BKCMP>=	Set word/doubleword data block comparison greater than or equal to
	*BKCMP<=	Set word/doubleword data block comparison less than or equal to
	BKITD	Batch conversion from integers to long integers
	BKDTI	Batch conversion from long integers to integers
	*BKFLT	Batch conversion from integers/long integers to floating-point numbers
	*BKINT	Batch conversion from floating-point numbers to integer/long integers
	BKWBIT	Assign word element to bit element combination
	BKBITW	Assign bit element combination to word element
	*BKAND	AND operation of word/doubleword data block
	*BKOR	OR operation of word/doubleword data block
	*BKXNR	XNOR operation of word/doubleword data block
	*BKXOR	XOR operation of word/doubleword data block
	*BKINV	Inversion operation of word/doubleword data block
Data Table Command	LIMIT	Upper-lower limit control
	DBAND	Deadband control
	ZONE	Zone control
	*SCL	Coordinate determination of word/doubleword data
	SER	Data retrieval
Table Operation Command	*SORTR	Sort word/doubleword data by row
	*SORTC	Sort word/doubleword data by column
	FDEL	Data deletion of data table
	FINS	Data insertion of data table
String Command	STRADD	String combination
	STRLEN	String length detection
	STRRIGHT	Read from right side of string
	STRLEFT	Read from left side of string
	STRMIDR	Randomly read from string
	STRMIDW	Randomly replace from string
	STRINSTR	String retrieval
	STRMOV	String transfer
Data Processing Command	WTOB	Data separation of byte unit
	BTOW	Data combination of byte unit
	UNI	4-bit combination of 16-bit data
	DIS	4-bit separation of 16-bit data
	ANS	Signal alarm set
	ANR	Signal alarm reset

Command Category		Name	Function
CANopen Motion Control Axis Command (Developed in Mid-2023)		MC_Power_CO	Communication control servo axis enabled
		MC_Reset_CO	Fault reset of communication control servo axis
		MC_ReadActualPosition_CO	Read current actual axis position by communication control
		MC_ReadActualVelocity_CO	Read current actual velocity by communication control
		MC_Halt_CO	Communication control servo axis halt (interruptible)
		MC_Stop_CO	Communication control servo axis stop (uninterruptible)
		MC_MoveAbsolute_CO	Absolute positioning of communication control axis
		MC_MoveRelative_CO	Relative positioning of communication control axis
		MC_MoveVelocity_CO	Velocity operation mode of communication control axis
		MC_Jog_CO	Communication control axis jog
		MC_Home_CO	Communication control axis home
		MC_WriteParameter_CO	Write axis parameter by communication control
		MC_ReadParameter_CO	Read axis parameter by communication control
Encoder Axis (TS600)		ENC_Counter	Encoder enable
		ENC_Reset	Encoder reset
		ENC_Preset	Encoder preset
		ENC_TouchProbe	Encoder probe
		ENC_ArrayCompare	Unidimensional array comparison of encoder
		ENC_StepCompare	Unidimensional step size comparison of encoder
		ENC_Compare	Single-point comparison output
		ENC_GroupArrayCompare	Bidimensional array comparison of encoder
		ENC_ReadStatus	Encoder status acquisition
		ENC_DigitalOutput	Encoder digital output control
		ENC_ResetCompare	Encoder reset comparison output
		ENC_SetUnit	Set axis gear ratio
		ENC_SetLineRotationMode	Set axis operation mode
High-speed counter		HC_Counter	High-speed counter enable
		HC_Preset	High-speed counter preset value
		HC_TouchProbe	High-speed counter probe
		HC_Compare	High-speed counter comparison
		HC_ArrayCompare	High-speed counter array comparison
		HC_StepCompare	High-speed counter equidistance comparison
Communication Protocol Command	Free Protocol for Serial Ports	Free_Serial	Sending and receiving under free protocol for serial ports
	Free Protocol for	TCP_Server	Server socket creation
		TCP_Accept	Reception of client connection

Command Category		Name	Function
	TCP/IP		request by server
		TCP_Client	Client socket creation
		TCP_Send	Send TCP data
		TCP_Recv	Receive TCP data
		TCP_Close	Close TCP socket
	Free Protocol for UDP/IP	UDP_Peer	UDP socket creation
		UDP_Send	Send UDP data
		UDP_Recv	Receive UDP data
	EtherCAT	ECAT_ReadParameter_CoE	Read SDO parameter from slave station
		ECAT_WriteParameter_CoE	Write SDO parameter to slave station
		ECAT_RestartMaster_CoE	Restart EtherCAT master station
Real-time Clock Command		TRD	Real-time clock read
		TWR	Real-time clock write
		TADD	Clock addition operation
		TSUB	Clock subtraction operation
		HOUR	Hour meter
		DCMP=	Date comparison equal to
		DCMP>	Date comparison greater than
		DCMP<	Date comparison less than
		DCMP<>	Date comparison not equal to
		DCMP>=	Date comparison greater than or equal to
		DCMP<=	Date comparison less than or equal to
		TCMP=	Time comparison equal to
		TCMP>	Time comparison greater than
		TCMP<	Time comparison less than
		TCMP<>	Time comparison not equal to
		TCMP>=	Time comparison greater than or equal to
		TCMP<=	Time comparison less than or equal to
		HTO*S	Conversion from hours, minutes, or seconds to word/doubleword second data
		*STOH	Conversion from word/doubleword second data to hours, minutes, or seconds
Control Calculation Command		PID	PID function
		RAMP	Ramp signal output
		HACKLE	Hackled wave signal output
		TRIANGLE	Triangular wave signal output
Verification Command		CCITT	CCITT checksum calculation
		CRC16	CRC16 checksum calculation
		LRC	LRC checksum calculation
		CCD	CCD checksum calculation
Other Commands		RND	Generate random number
		DUTY	Generate timed pulse

3 Command Instructions

In Chapter "Command Instructions", command contents are explained in details through "Command Table", "Operand Content Description", "Command Function", "Precautions", and "Application Example". Among them, the SET command is taken as an example for "Command Table".



No.	Item	Description
①	Command table	Describes how to write the command code and the order in which the operands are filled in
②	Applicable model	Describes the model of the INVT PLC product supported by the command
③	Command Name	Displays the names of 16-bit and 32-bit commands
④	Operand	Displays the method to input the operand, which is either an source operand or an destination operand. S represents the source operand, D represents the destination operand, and n is the source operand that represents the quantity
⑥	Type	Indicates the data type supported by the operand, which can be BOOL (bit), WORD (unsigned 16-bit), DWORD (unsigned 32-bit), INT (signed 16-bit), or DINT (signed 32-bit). Some commands support array input
⑦	Indexing	Indicates that the operand supports indexing through the Z element in the form of, for example, D0Z0, where the Z element itself does not support indexing
⑧	Constant	Indicates that the operand supports constant inputs, which include floating-point number constants and integer constants. In case of a string command, string constants are supported
⑨	Remarks	Further details the supported element types
Others	Supported element	Lists all the bit and word elements supported in PLC, where "√" is used to indicates that the operand supports a certain part of the elements, and "⑨" (remark) is used to provide detailed explanations

3.1 Contact Logic Command

3.1.1 Command Table

Command Category	Name	Function
Contact Logic Command	LD	Normally open contact
	LDI	Normally closed contact
	LDP	Take rising pulse edge
	LDF	Take falling pulse edge
	AND	Normally open contact AND
	ANI	Normally closed contact OR

Command Category	Name	Function
	ANDP	Serial connection of AND rising pulse edge detection
	ANDF	Serial connection of AND falling pulse edge detection
	OR	Normally open contact OR
	ORI	Normally closed contact OR
	ORP	Serial connection of OR rising pulse edge detection
	ORF	Serial connection of OR falling pulse edge detection
	ANB	Energy flow block AND
	ORB	Energy flow block OR
	EU	Rising energy flow edge detection
	ED	Falling energy flow edge detection

3.1.2 LD&LDI&LDP&LDF: Contact Operation Commands

Command table	LD* (S)			Applicable model	TS600 series		
16-Bit command	LD: Normally open contact						
32-Bit command	-						
16-Bit command	LDI: Normally closed contact						
32-Bit command	-						
16-Bit command	LDP: Take rising pulse edge						
32-Bit command	-						
16-Bit command	LDF: Take falling pulse edge						
32-Bit command	-						

Operand	Type	Bit			Word		Indexing	Constant
		X, Y, M, LM, T, C, S	Dx.y	Custom bit variable	D, R, V, Z, T, C	Custom word variable		
S	BOOL	√[1]	√	√	-	-	-	-

Remark:

[1] LDP and LDF do not support LM elements.

Operand Description

S: The source operand, which determines the soft elements or variables of the energy flow state.

Function Description

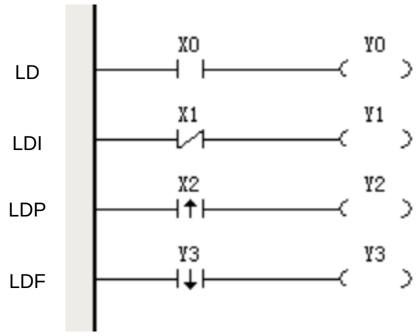
LD&LDI&LDP&LDF commands are used for the output state operation starting from the left busbar to obtain the output energy flow state. Among them:

1. The LD command is used for normally open contact to take the energy flow state. If the corresponding signal is detected to be high level during this scan, the contact is valid.
2. The LDI command is used for normally closed contact to take the energy flow state. If the corresponding signal is detected to be high level during this scan, the contact is valid.
3. The LDP command is used to take the rising edge of a bit element. If a rising jump of the corresponding signal is detected during this scan, the contact is valid. However, the contact becomes invalid as soon as the next scan.
4. The LDF command is used to take the falling edge of a bit element. If a falling jump of the corresponding signal is detected during this scan, the contact is valid. However, the contact becomes invalid as soon as the next scan.

Precautions

Up to 4096 edge commands such as LDP and LDF can be present at the same time.

Application Example



3.1.3 AND&ANI&ANDP&ANDF: Serial Contact Operation Commands

<table>
<tr><td colspan="2">Command table</td><td colspan="3">AND* (S)</td><td>Applicable model</td><td colspan="3">TS600 series</td></tr>
<tr><td colspan="2">16-Bit command</td><td colspan="7">AND: Normally open contact AND</td></tr>
<tr><td colspan="2">32-Bit command</td><td colspan="7">-</td></tr>
<tr><td colspan="2">16-Bit command</td><td colspan="7">ANI: Normally closed contact OR</td></tr>
<tr><td colspan="2">32-Bit command</td><td colspan="7">-</td></tr>
<tr><td colspan="2">16-Bit command</td><td colspan="7">ANDP: Serial connection of AND rising pulse edge detection</td></tr>
<tr><td colspan="2">32-Bit command</td><td colspan="7">-</td></tr>
<tr><td colspan="2">16-Bit command</td><td colspan="7">ANDF Serial connection of AND falling pulse edge detection</td></tr>
<tr><td colspan="2">32-Bit command</td><td colspan="7">-</td></tr>
<tr><td rowspan="2">Operand</td><td rowspan="2">Type</td><td colspan="3">Bit</td><td colspan="2">Word</td><td rowspan="2">Indexing</td><td rowspan="2">Constant</td></tr>
<tr><td>X, Y, M, LM, T, C, S</td><td>Dx.y</td><td>Custom bit variable</td><td>D, R, V, Z, T, C</td><td>Custom word variable</td></tr>
<tr><td>S</td><td>BOOL</td><td>√[1]</td><td>√</td><td>√</td><td>-</td><td>-</td><td>-</td><td>-</td></tr>
</table>

Remark:

[1] ANDP and ANDF do not support LM elements.

Operand Description

S: The source operand, which determines the soft elements or variables of the energy flow state.

Function Description

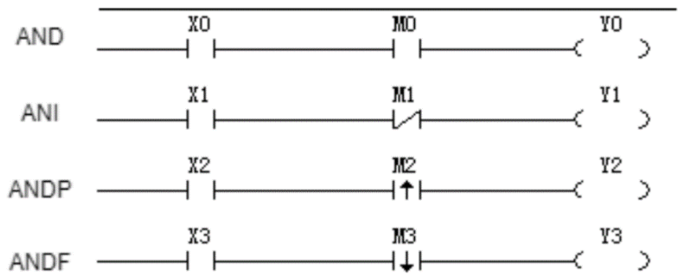
AND&ANI&ANDP&ANDF commands are used for the output state operation after connecting soft elements in series. The command first reads the state of the bit element and then performs a logical "AND" operation with its previous energy flow state to obtain the output energy flow state. Among them:

1. The AND command is used for normally open contact to take the energy flow state. If the corresponding signal is detected to be high level during this scan, the contact is valid. The "AND" operation is done through this logic.
2. The ANI command is used for normally closed contact to take the energy flow state. If the corresponding signal is detected to be high level during this scan, the contact is valid. The "AND" operation is done through this logic.
3. The ANDP command is used to take the rising edge of a bit element. If a rising jump of the corresponding signal is detected during this scan, the contact is valid. However, the contact becomes invalid as soon as the next scan. The "AND" operation is done through this logic.
4. The ANDF command is used to take the falling edge of a bit element. If a falling jump of the corresponding signal is detected during this scan, the contact is valid. However, the contact becomes invalid as soon as the next scan. The "AND" operation is done through this logic.

Precautions

Up to 4096 edge commands such as ANDP and ANDF can be present at the same time.

Application Example



3.1.4 OR&ORI&ORP&ORF: Parallel Contact Operation Commands

Command table	OR* (S)	Applicable model	TS600 series
16-Bit command	OR: Normally open contact OR		
32-Bit command	-		
16-Bit command	ORI: Normally closed contact OR		
32-Bit command	-		
16-Bit command	ORP: Serial connection of OR rising pulse edge detection		
32-Bit command	-		
16-Bit command	ORF: Serial connection of OR falling pulse edge detection		
32-Bit command	-		

Operand	Type	Bit			Word		Indexing	Constant
		X, Y, M, LM, T, C, S	Dx.y	Custom bit variable	D, R, V, Z, T, C	Custom word variable		
S	BOOL	√[1]	√	√	-	-	-	-

Remark:

[1] ORP and ORF do not support LM elements.

Operand Description

S: The source operand, which determines the soft elements or variables of the energy flow state.

Function Description

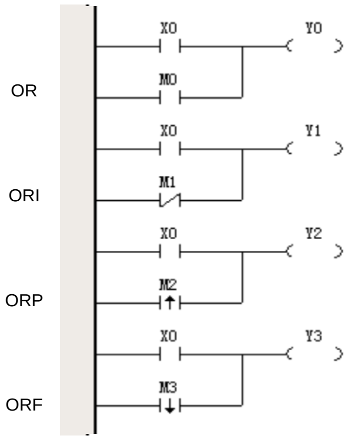
OR&ORI&ORP&ORF commands are used for the output state operation after connecting soft elements in parallel. The command first reads the state of the bit element and then performs a logical "OR" operation with its previous energy flow state to obtain the output energy flow state. Among them:

1. The OR command is used for normally open contact to take the energy flow state. If the corresponding signal is detected to be high level during this scan, the contact is valid. The "OR" operation is done through this logic.
2. The ORI command is used for normally closed contact to take the energy flow state. If the corresponding signal is detected to be high level during this scan, the contact is valid. The "OR" operation is done through this logic.
3. The ORP command is used to take the rising edge of a bit element. If a rising jump of the corresponding signal is detected during this scan, the contact is valid. However, the contact becomes invalid as soon as the next scan. The "OR" operation is done through this logic.
4. The ORF command is used to take the falling edge of a bit element. If a falling jump of the corresponding signal is detected during this scan, the contact is valid. However, the contact becomes invalid as soon as the next scan. The "OR" operation is done through this logic.

Precautions

Up to 4096 edge commands such as ORP and ORF can be present at the same time.

Application Example

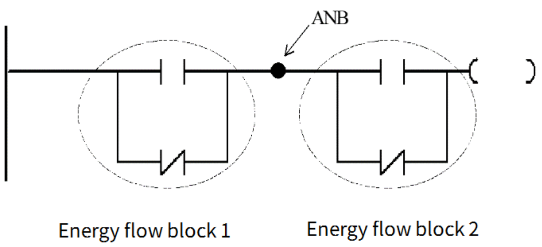


3.1.5 ANB&ORB: Operation Commands for Energy Flow Block Connection

Command table		*B			Applicable model		TS600 series	
16-Bit command		ANB: Energy flow block AND						
32-Bit command		-						
16-Bit command		ORB: Energy flow block OR						
32-Bit command		-						
Operand	Type	Bit			Word		Indexing	Constant
		X, Y, M, LM, T, C, S	Dx.y	Custom bit variable	D, R, V, Z, T, C	Custom word variable		
-	-	-	-	-	-	-	-	-

Function Description

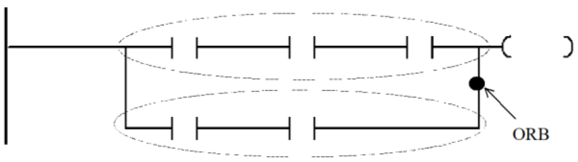
ANB Command:



It performs an "AND" operation on the energy flow values of two energy flow blocks and assigns the values to the current energy flow.

ORB Command:

Energy flow block 1

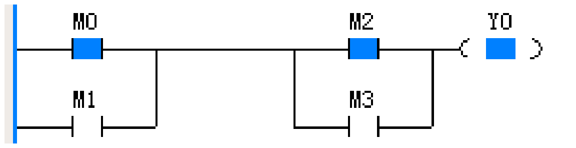


Energy flow block 2

It performs an "OR" operation on the energy flow values of two energy flow blocks and assigns the values to the current energy flow.

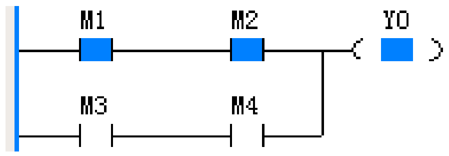
Application Example

ANB Command:



When at least one of M0 and M1 is ON, and at least one of M2 and M3 is ON, Y0 outputs ON.

ORB Command:



In case of M1=M2=ON or M3=M5=ON, Y0 outputs ON.

3.1.6 EU&ED: Energy Flow Edge Detection Commands

Command table		E*			Applicable model	TS600 series		
16-Bit command		EU: Rising energy flow edge detection						
32-Bit command		-						
16-Bit command		ED: Falling energy flow edge detection						
32-Bit command		-						
Operand	Type	Bit			Word		Indexing	Constant
		X, Y, M, LM, T, C, S	Dx.y	Custom bit variable	D, R, V, Z, T, C	Custom word variable		
-	-	-	-	-	-	-	-	-

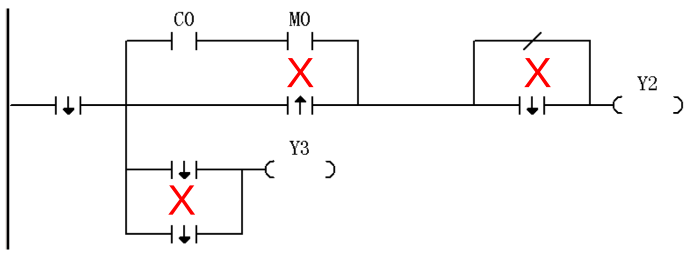
Function Description

EU: It compares the changes in input energy flow between this scan and the last scan. When the energy flow has a rising edge change (OFF → ON), the output is valid during this scan cycle.

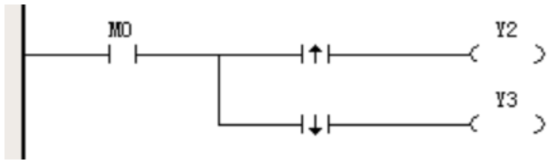
ED: It compares the changes in input energy flow between this scan and the last scan. When the energy flow has a falling edge change (ON → OFF), the output is valid during this scan cycle.

Precautions

- In the ladder diagram, the rising or falling edge contact command should be used in series with other contact elements and cannot be used in parallel with other contact elements.
- In the ladder diagram, the rising or falling edge contact command cannot be directly connected to the left energy flow busbar.
- Up to 4096 edge commands such as EU and ED can be present at the same time.
- An example of incorrect use of EU/ED commands in the ladder diagram is shown below:



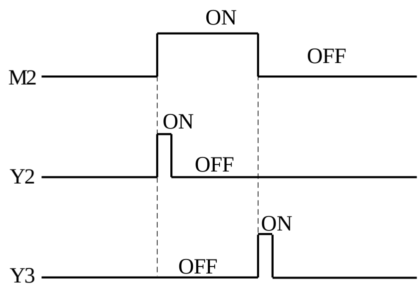
Application Example



In two consecutive scan cycles, the states of the M0 contact are OFF and ON, respectively. The EU command detects a change in the rising edge, causing Y2 to output the ON state for one scan cycle width.

In two consecutive scan cycles, the states of the M0 contact are ON and OFF, respectively. The ED command detects a change in the falling edge, causing Y3 to output the ON state for one scan cycle width.

Timing diagram



3.2 Output Control Command

3.2.1 Command Table

Command Category	Name	Function
Output Control Command	OUT	Coil output
	SET	Coil set
	RST	Coil reset
	PLS	Rising pulse edge detection coil
	PLF	Falling pulse edge detection coil
	ALT	Alternating output
	NOP	Null operation

3.2.2 OUT: Coil Output Commands

Command table	OUT (D)			Applicable model	TS600 series		
16-Bit command	OUT: Coil output						
32-Bit command	-						

Operand	Type	Bit			Word		Indexing	Constant
		X, Y, M, LM, T, C, S	Dx.y	Custom bit variable	D, R, V, Z, T, C	Custom word variable		
D	BOOL	√[1]	√	√	-	-	-	-

Remark:

[1] The X element is not supported, and the S element is separately listed in 3.2.3 SET: Coil Set Commands.

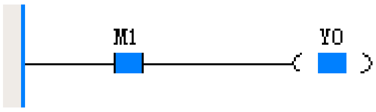
Operand Description

D: The destination operand.

Function Description

It assigns the current energy flow value to the specified coil (D).

Application Example



When M1 is ON, Y0 outputs ON.

3.2.3 SET: Coil Set Commands

Command table		SET (D)			Applicable model	TS600 series		
16-Bit command		SET: Coil set						
32-Bit command		-						
Operand	Type	Bit			Word		Indexing	Constant
		X, Y, M, LM, T, C, S	Dx.y	Custom bit variable	D, R, V, Z	Custom word variable		
D	BOOL	√[1]	√	√	-	-	-	-

Remark:

[1] The X element is not supported, and the S element is separately listed in 3.4.3 SET/RST/OUT S (label): SFC State Operation Commands.

Operand Description

D: The destination operand.

Function Description

When the energy flow is valid, the bit element specified by (D) will be set. After setting, the bit element specified by (D) will still remain in the set state, regardless of whether the command is driven or not. This state can be reset by the RST command.

Application Example

In case M0=ON, the M1 element is set.

3.2.4 RST: Coil Reset Commands

Command table		RST (D)			Applicable model	TS600 series		
16-Bit command		RST: Coil reset						
32-Bit command		-						
Operand	Type	Bit			Word		Indexing	Constant
		X, Y, M, LM, T, C, S	Dx.y	Custom bit variable	D, R, V, Z, T, C	Custom word variable		
D	BOOL	√[1]	√	√	-	-	-	-

Remark:

[1] The X element is not supported, and the S element is separately listed in 3.4.3 SET/RST/OUT S (label): SFC State Operation Commands.

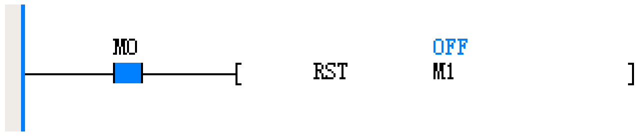
Operand Description

D: The destination operand.

Function Description

When the energy flow is valid, the bit element specified by (D) will be reset. After resetting, the bit element specified by (D) will still remain in the set state, regardless of whether the command is driven or not. This state can be set by the SET command.

Application Example



In case M0=ON, the M1 element is reset.

3.2.5 PLS&PLF: Pulse Edge Detection Coil Commands

Command table		PL* (D)			Applicable model	TS600 series		
16-Bit command	PLS: Rising pulse edge detection coil							
32-Bit command	-							
16-Bit command	PLF: Falling pulse edge detection coil							
32-Bit command	-							
Operand	Type	Bit			Word		Indexing	Constant
		X, Y, M, LM, T, C, S	Dx.y	Custom bit variable	D, R, V, Z, T, C	Custom word variable		
D	BOOL	√[1]	√	√	-	-	-	-

Remark:

[1] The X element is not supported.

Operand Description

D: The destination operand.

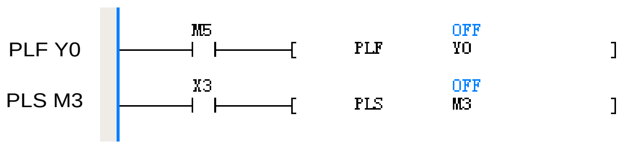
Function Description

1. PLS command: When the rising edge of the energy flow appears, the specified (D) element is set and the set state is maintained for one scan cycle.
2. PLF command: When the falling edge of the energy flow appears, the specified (D) element is set and the set state is maintained for one scan cycle.

Precautions

Up to 4096 edge commands such as PLS and PLF can be present at the same time.

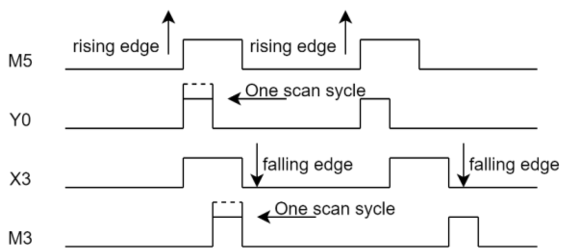
Application Example



When M5 is switched from ON to OFF, the Y0 element is set.

When X3 is switched from OFF to ON, the M3 element is set.

Timing diagram



3.2.6 ALT: Alternating Output Commands

Command table	ALT (D)			Applicable model	TS600 series		
16-Bit command	ALT: Alternating output						
32-Bit command	-						

Operand	Type	Bit			Word		Indexing	Constant
		X, Y, M, LM, T, C, S	Dx.y	Custom bit variable	D, R, V, Z, T, C	Custom word variable		
D	BOOL	√[1]	√	√	-	-	-	-

Remark:

[1] Only the Y, M, and S elements are supported.

Operand Description

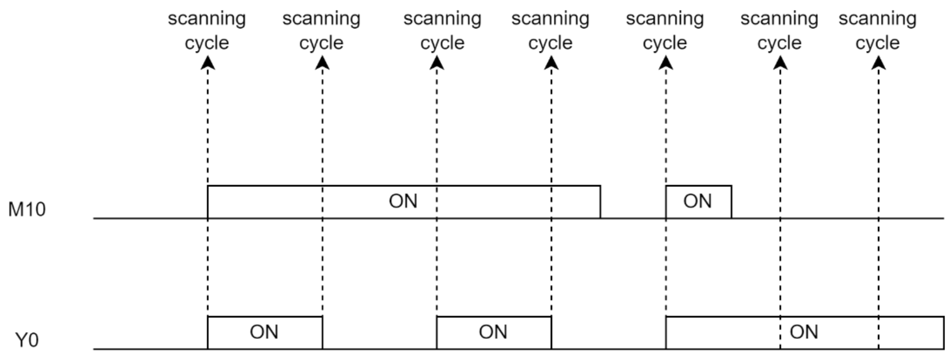
D: The destination operand, which alternately outputs bit elements.

Function Description

When the energy flow is effective, the bit element pointed to by (D) acts in reverse for each scan cycle.

Application Example

When M10 is set to ON, Y0 flips the element values at each scan cycle, as shown in the following figure.



3.2.7 NOP: Null Operation Commands

Command table	NOP			Applicable model	TS600 series		
16-Bit command	NOP: Null operation						
32-Bit command	-						

Operand	Type	Bit			Word		Indexing	Constant
		X, Y, M, LM, T, C, S	Dx.y	Custom bit variable	D, R, V, Z, T, C	Custom word variable		
-	-	-	-	-	-	-	-	-

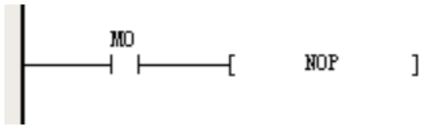
Function Description

This command does not generate any action.

Precautions

In the ladder diagram, this command cannot be directly connected to the left energy flow busbar.

Application Example



In case of M0=ON, the PLC performs a null operation.

3.3 Energy Flow Control Command

3.3.1 INV: Energy Flow Inversion Commands

Command table		INV			Applicable model	TS600 series		
16-Bit command		INV: Energy flow inversion						
32-Bit command		-						
Operand	Type	Bit			Word		Indexing	Constant
		X, Y, M, LM, T, C, S	Dx.y	Custom bit variable	D, R, V, Z, T, C	Custom word variable		
-	-	-	-	-	-	-	-	-

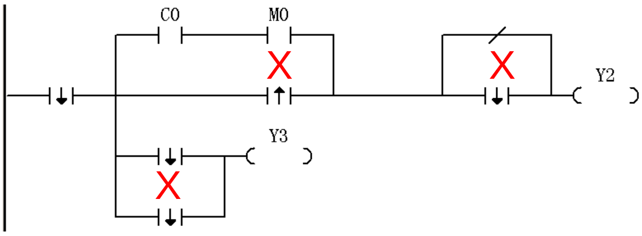
Function Description

It inverts the current energy flow value and assign the resultant value to the current energy flow.

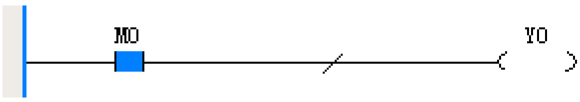
Precautions

- In the ladder diagram, the energy flow inversion command should be used in series with contact elements and cannot be used in parallel with other contact elements.
- INV cannot be used as the first command for parallel branches of the output.
- In the ladder diagram, the energy flow inversion command cannot be directly connected to the left energy flow busbar.

An example of incorrect use of INV commands in the ladder diagram is shown below:



Application Example



In case of M0=ON, the current energy flow is ON. After inversion, the value is assigned to Y0, which means that Y0 is reset.

3.4 SFC Command

3.4.1 Command Table

Command Category	Name	Function
SFC Command	STL	SFC state load
	SET S(label)	SFC state transition
	OUT S(label)	SFC state jump
	RST S(label)	SFC state clear
	RET	SFC program end

3.4.2 STL: SFC State Load Commands

Command table	STL (S)			Applicable model	TS600 series		
16-Bit command	STL: SFC state load						
32-Bit command	-						

Operand	Type	Bit			Word		Indexing	Constant
		X, Y, M, LM, T, C, S	Dx.y	Custom bit variable	D, R, V, Z, T, C	Custom word variable		
S	BOOL	√[1]	-	-	-	-	-	-

Remark:

[1] Only the S element is supported.

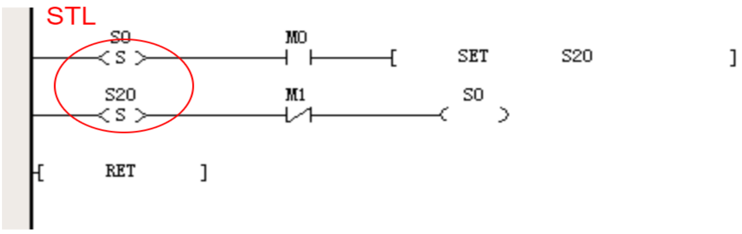
Operand Description

S: The S element number of the STL statement to be executed.

Function Description

1. It represents the beginning of a step state (S) processing.
2. If the step state is valid (ON), its built-in commands will be executed.
3. If the step state changes from valid to invalid (falling edge change), its built-in command sequence will not be executed, and the built-in ladder diagram program will be reset.
4. If the step state is invalid,, its built-in command sequence will not be executed.
5. Continuous STL commands (serial connection of STL components) represent a defined parallel merging structure. STL commands can be consecutively used for up to 16 times, (the parallel branch merging structure has a maximum of 16 branches).

Application Example



As shown in the above figure, STL is used to load a stepping state. In case of S0=ON, this step state is enabled; when M0 is set, it transitions to the S20 step state.

3.4.3 SET/RST/OUT S (label): SFC State Operation Commands

Command table	SET/RST/OUT S(label) (D)			Applicable model	TS600 series		
16-Bit command	SET S(label): SFC state transition						
32-Bit command	-						
16-Bit command	RST S(label): SFC state clear						
32-Bit command	-						
16-Bit command	OUT S(label): SFC state jump						
32-Bit command	-						

Operand	Type	Bit			Word		Indexing	Constant
		X, Y, M, LM, T, C, S	Dx.y	Custom bit variable	D, R, V, Z, T, C	Custom word variable		
S	BOOL	√[1]	-	-	-	-	-	-

Remark:

[1] Only the S element is supported.

Operand Description

S: The S element number of the STL statement to be executed.

Function Description

1. SET S(label): When the command is driven, it sets the specified step state (D) to valid and the currently valid step state to invalid, completing the step state jump action.
2. RST S(label): When the command is driven, it sets the specified step state (D) to invalid.
3. OUT S(label): When the command is driven, it sets the specified step state (D) to valid and the currently valid step state to invalid, completing the step state transition action.

Application Example

When the S0 step state is activated:

- In case of M0=ON, this step state transitions to the S20 step state;
- In case of M1=ON, this step state resets to the S21 step state;
- In case of M2=ON, this step state jumps to the S22 step state.

3.4.4 RET: SFC Program Segment End

Command table	RET				Applicable model	TS600 series		
16-Bit command	RET: SFC Program Segment End							
32-Bit command	-							
Operand	Type	Bit			Word		Indexing	Constant
		X, Y, M, LM, T, C, S	Dx.y	Custom bit variable	D, R, V, Z, T, C	Custom word variable		
-	-	-	-	-	-	-	-	-

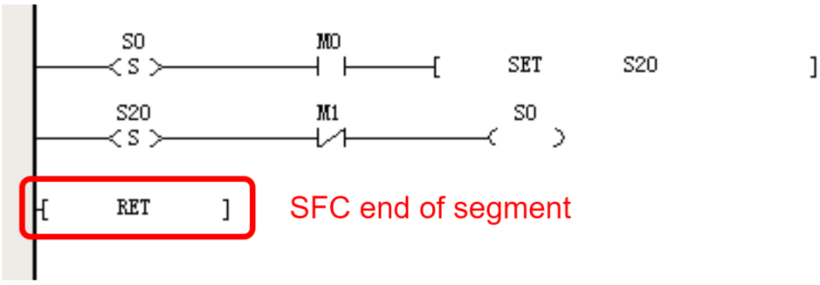
Function Description

1. The step ladder diagram is a controlled device based run procedure, which is decomposed into several states or processes, logically programs each state, and then switch states based on signal conditions. During programming, the STL ladder diagram is used, which provides a clear thought, simplifies logical design, and facilitates debugging and maintenance.
2. The step ladder diagram command can be represented by a ladder diagram, where the state (S) is considered as a control process, and the input conditions and output control are programmed in sequence. The most significant feature of this control is that the current process is not connected to the previous process when running and the device can be controlled in a simple sequence of each process.
3. The step ladder diagram has corresponding programming rules, which include the programming methods of ordinary ladder diagrams and differ from programming rules of ordinary ladder diagrams to a certain extent, as explained below.
4. The step ladder program starts with the STL command (note that it is different from the S state in the ordinary ladder diagram) and ends with the RET command. The intermediate program is led by the S state and followed by all operation logics of the S state, including the operations used for switching to the next state when the conditions are met.

Precautions

It can be used only in the main program.

Application Example



The RET command indicates that the SFC program segment ends and separates from the ordinary ladder diagram, and the ladder diagram program after the segment continues to run.

3.5 Program Flow Control Command

3.5.1 Command Table

Command Category	Name	Function
Program Flow Control Command	FOR	Loop operation
	NEXT	Loop return
	LBL	Jump label definition
	CJ	Conditional jump
	CFEND	Conditional return of main user program
	WDT	User program watchdog reset
	EI	Interrupt enable
	DI	Interrupt disable
	CIRET	Conditional return of user interrupt program
	STOP	User program stop
	CALL	User subroutine call
	CSRET	Conditional return of user subroutine

3.5.2 FOR: Loop Operation

<table>
<tr><td colspan="2">Command table</td><td colspan="3">FOR (S)</td><td>Applicable model</td><td colspan="3">TS600 series</td></tr>
<tr><td colspan="2">16-Bit command</td><td colspan="7">FOR: Loop operation</td></tr>
<tr><td colspan="2">32-Bit command</td><td colspan="7">-</td></tr>
<tr><td rowspan="2">Operand</td><td rowspan="2">Type</td><td colspan="3">Bit</td><td colspan="2">Word</td><td rowspan="2">Indexing</td><td rowspan="2">Constant</td></tr>
<tr><td>X, Y, M, LM, T, C, S</td><td>Dx.y</td><td>Custom bit variable</td><td>D, R, V, Z, T, C</td><td>Custom word variable</td></tr>
<tr><td>S</td><td>INT</td><td>-</td><td>-</td><td>-</td><td>√</td><td>√</td><td>√</td><td>√</td></tr>
</table>

Operand Description

S: The source operand, which indicates the number of loops.

Function Description

It forms a FOR-NEXT structure with the NEXT command, as explained in the section for NEXT.

3.5.3 NEXT: Loop Return

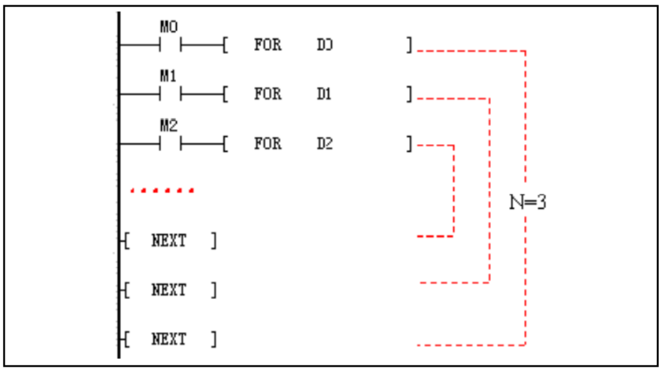
<table>
<tr><td>Command table</td><td colspan="4">NEXT-</td><td>Applicable model</td><td colspan="3">TS600 series</td></tr>
<tr><td>16-Bit command</td><td colspan="8">NEXT: Loop return</td></tr>
<tr><td>32-Bit command</td><td colspan="8">-</td></tr>
<tr><td rowspan="2">Operand</td><td rowspan="2">Type</td><td colspan="3">Bit</td><td colspan="2">Word</td><td rowspan="2">Indexing</td><td rowspan="2">Constant</td></tr>
<tr><td>X, Y, M, LM, T, C, S</td><td>Dx.y</td><td>Custom bit variable</td><td>D, R, V, Z, T, C</td><td>Custom word variable</td></tr>
<tr><td>-</td><td>-</td><td>-</td><td>-</td><td>-</td><td>-</td><td>-</td><td>-</td><td>-</td></tr>
</table>

Function Description

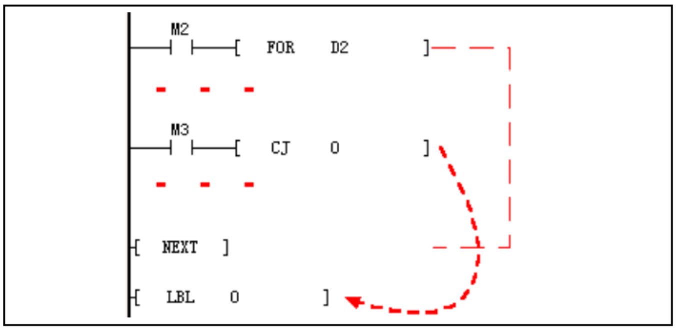
1. The FOR command matches with the NEXT command to form a FOR NEXT structure.
2. When the energy flow before the FOR command is valid and the number of loop times (S) is greater than zero, the command in the middle of the FOR-NEXT structure is consecutively looped for S times. When the command has been looped for S times, the program continues to execute the commands after the FOR-NEXT structure.
3. If the energy flow before the FOR command is invalid, or the number of loop times (S) is less than or equal to zero, the command in the middle of the FOR-NEXT structure is not executed, and the program directly jumps after the FOR-NEXT structure and continues to execute.

Precautions

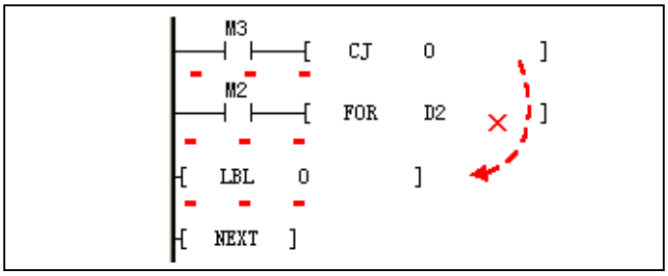
- FOR-NEXT commands must be used in pairs within a program body (POU), otherwise the user program cannot be compiled correctly.
- Nesting multiple FOR-NEXT structures is supported, and the TS600 series only supports up to 8 layers of nested FOR-NEXT structures. An example of 3-layer nested FOR-NEXT structures is shown in the figure below.



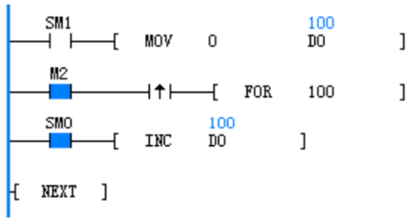
- Conditional jump commands (CJ) can be used within the loop body to jump out of the loop body, thereby early terminating the loop body execution, as shown in the ladder diagram below.



- Users are prohibited from using jump statements (CJ) to jump into a loop body, otherwise the following ladder diagram cannot be compiled correctly.



Application Example



The initial conditions for running are D0=0 and M2=OFF. When M2 changes from OFF to ON, the command within the FOR-NEXT structure are executed for 100 times continuously, and D0 is increased by 1 for 100 times. After the loop ends, the result is D0=100.

3.5.4 LBL: Jump Label Definition Commands

Command table		LBL (S)			Applicable model	TS600 series		
16-Bit command		LBL: Jump label definition						
32-Bit command		-						
Operand	Type	Bit			Word		Indexing	Constant
		X, Y, M, LM, T, C, S	Dx.y	Custom bit variable	D, R, V, Z, T, C	Custom word variable		
S	INT	-	-	-	-	-	-	√

Operand Description

S: The label value.

Function Description

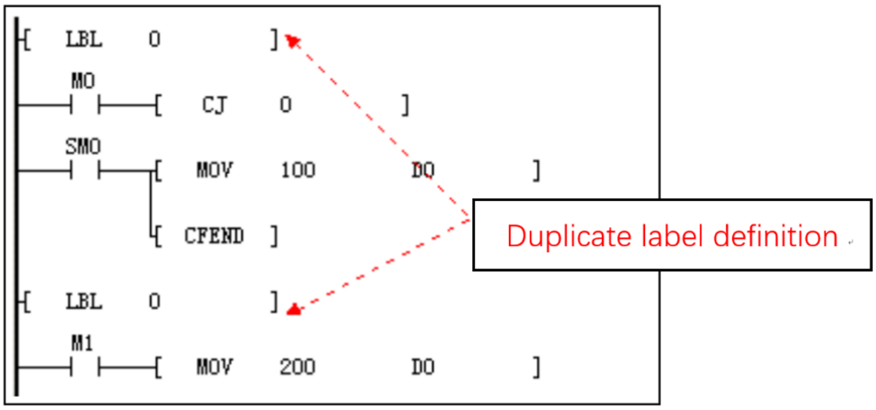
1. It defines a label that has a value of S.
2. It does not generate any substantive operation, but only indicates the specific jump position for the conditional jump command (CJ).

Precautions

- The range of label value S: $0 \leqslant S \leqslant 1023$.
- In a user program, it is not allowed to have two duplicate defined labels in the same program body, otherwise the user program cannot be compiled. However, duplicate label definitions are allowed in different program bodies (such as different subroutines).

Error Example

As shown in the figure, there are the same label definitions and therefore a compilation error occurs.



3.5.5 CJ: Conditional Jump Commands

Command table	CJ (S)			Applicable model	TS600 series		
16-Bit command	CJ: Conditional jump						
32-Bit command	-						

Operand	Type	Bit			Word		Indexing	Constant
		X, Y, M, LM, T, C, S	Dx.y	Custom bit variable	D, R, V, Z, T, C	Custom word variable		
S	INT	-	-	-	-	-	-	√

Operand Description

S: The label value.

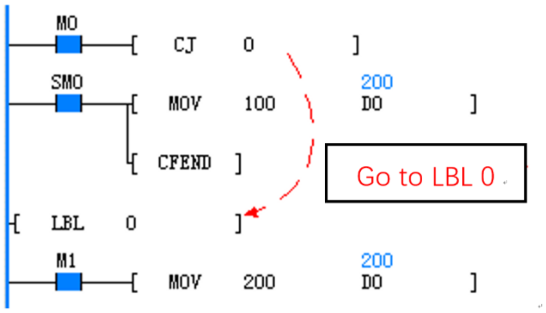
Function Description

1. When the energy flow is valid, the user program jumps to the legally labeled command with the number S for execution.
2. If the energy flow is invalid, no jump operation occurs and the next command after CJ is executed in sequence.

Precautions

- The label S (0 ⩽ S ⩽ 1023) that the CJ command needs to jump to should be a valid and defined label, otherwise the user program cannot be compiled correctly.
- It is not allowed to use the CJ command for jumping into a FOR-NEXT structure.
- It is allowed to use the CJ command for jumping out of or into the SFC state, but this will disrupt the logic of the SFC state and make the program more complex. Therefore, it is not recommended to use the command in this way.

Application Example



The initial conditions are M0=OFF, M1=ON, and D0=100, with CJ 0 not jumping. After executing CFEND, the program flow exits the main program prematurely, and the commands LD M1 and MOV 200 D0 are not executed.

In case of M0=ON and M1=ON, the command CJ 0 is executed, and the commands MOV 100 D0 and CFEND are skipped. After jumping to LBL 0, the program executes the command MOV 200 D0, with D0=200 obtained as the result.

3.5.6 CFEND: Conditional Return of Main User Program

Command table		CFEND (S)			Applicable model	TS600 series		
16-Bit command		CFEND: Conditional return of main user program						
32-Bit command		-						
Operand	Type	Bit			Word		Indexing	Constant
		X, Y, M, LM, T, C, S	Dx.y	Custom bit variable	D, R, V, Z, T, C	Custom word variable		
S	INT	-	-	-	-	-	-	√

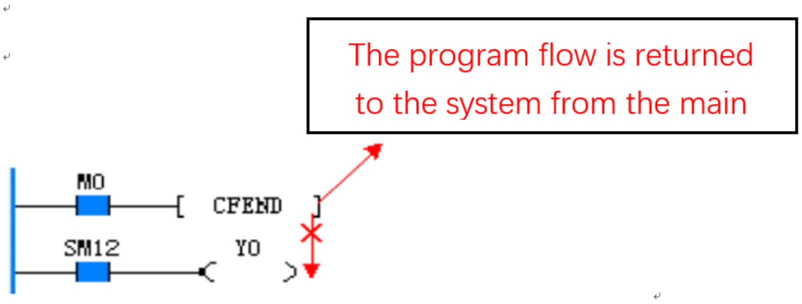
Function Description

1. When the energy flow of the command is valid, the main program returns to the system from the current scan cycle (the main program of the user program is called and executed by the system repeatedly according to the scan cycle), and subsequent commands in the main program are not executed.
2. When the energy flow of the command is invalid, the command does not generate any action, and subsequent commands are executed in sequence.

Precautions

The CFEND command must be present in the main user program, otherwise the program cannot be compiled.

Application Example



When the program runs with M0=OFF, the command CFEND command does not generate any action, the subsequent commands LD SM12 and OUT Y0 are executed, and the Y0 cycle flashing output is visible. In case of M0=ON, the command CFEND generates an action, and the program flow returns to the system prematurely from the main program, the subsequent commands LD SM12 and OUT Y0 are not executed, and the Y0 cycle flashing output disappears.

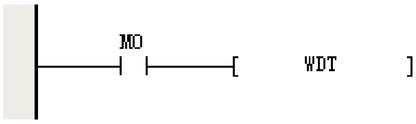
3.5.7 WDT: User Program Watchdog Reset

Command table		WDT			Applicable model	TS600 series		
16-Bit command		WDT: User program watchdog reset						
32-Bit command		-						
Operand	Type	Bit			Word		Indexing	Constant
		X, Y, M, LM, T, C, S	Dx.y	Custom bit variable	D, R, V, Z, T, C	Custom word variable		
-	-	-	-	-	-	-	-	-

Function Description

When the energy flow is valid, this command resets the timing value of the user program watchdog to zero, and the user program watchdog of the system restarts timing.

Application Example



In case of M0=ON, the system watchdog timing value is reset to zero.

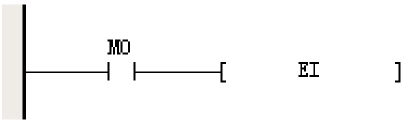
3.5.8 EI: Interrupt Enable

Command table		EI			Applicable model	TS600 series		
16-Bit command		EI: Interrupt enable						
32-Bit command		-						
Operand	Type	Bit			Word		Indexing	Constant
		X, Y, M, LM, T, C, S	Dx.y	Custom bit variable	D, R, V, Z, T, C	Custom word variable		
-	-	-	-	-	-	-	-	-

Function Description

1. When the energy flow is valid, the interrupt is enabled.
2. When the EI command is valid, interrupt requests are allowed to be added to the interrupt request queue, waiting for the system to respond.
3. Interrupt enable can be disabled using the DI command.

Application Example



In case of M0=ON, the enable system is interrupted, and interrupt requests are allowed.

3.5.9 DI: Interrupt Disable

Command table		DI			Applicable model	TS600 series		
16-Bit command		DI: Interrupt disable						
32-Bit command		-						
Operand	Type	Bit			Word		Indexing	Constant
		X, Y, M, LM, T, C, S	Dx.y	Custom bit variable	D, R, V, Z, T, C	Custom word variable		
-	-	-	-	-	-	-	-	-

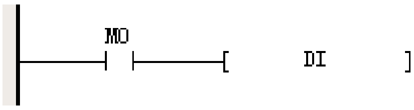
Function Description

1. When the energy flow is valid, the global interrupt enable flag becomes invalid, which means that the global interrupt is turned off.
2. When the global interrupt enable flag is invalid, various interrupt events cannot generate interrupt requests.
3. Interrupt disable can be enabled using the EI command.

Precautions

When the interrupt disable request command takes effect, if there are still pending interrupt requests in the interrupt request queue, the remaining interrupt requests must still be responded to, but new interrupt events cannot generate interrupt requests.

Application Example



In case of M0=ON, the enable system is interrupted, and interrupt requests are prohibited.

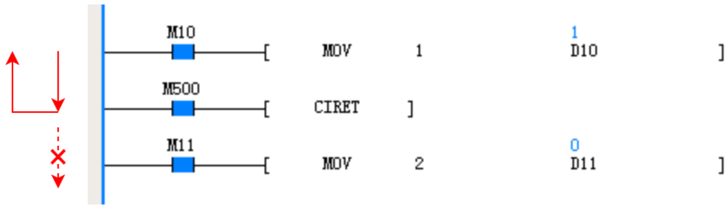
3.5.10 CIRET: Conditional Return of User Interrupt Program

Command table		CIRET			Applicable model	TS600 series		
16-Bit command		CIRET: Conditional return of user interrupt program						
32-Bit command		-						
Operand	Type	Bit			Word		Indexing	Constant
		X, Y, M, LM, T, C, S	Dx.y	Custom bit variable	D, R, V, Z, T, C	Custom word variable		
-	-	-	-	-	-	-	-	-

Function Description

When the energy flow is valid, the interrupt program being executed is exited prematurely.

Application Example



When this command is used in the interrupt subroutine, the subsequent program does not run. Note that when using the interrupt subroutine, the EI command is required to enable interrupt requests.

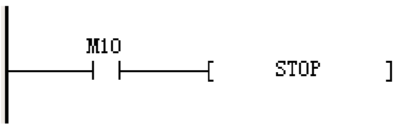
3.5.11 STOP: User Program Stop

Command table		STOP			Applicable model	TS600 series		
16-Bit command		STOP: User program stop						
32-Bit command		-						
Operand	Type	Bit			Word		Indexing	Constant
		X, Y, M, LM, T, C, S	Dx.y	Custom bit variable	D, R, V, Z, T, C	Custom word variable		
-	-	-	-	-	-	-	-	-

Function Description

When the energy flow is valid, the system immediately stops the execution of the user program.

Application Example



In case of M10=ON, the user program stops.

3.5.12 CALL: User Subroutine Call

Command table	CALL (subroutine name) (parameter 1) (parameter 2)			Applicable model	TS600 series			
16-Bit command	-							
32-Bit command	-							
Operand	Type	Bit			Word		Indexing	Constant
		X, Y, M, LM, T, C, S	Dx.y	Custom bit variable	D, R, V, Z, T, C	Custom word variable		
-	-	-	-	-	-	-	-	-

Function Description

When the energy flow is valid, the system calls a subroutine with the specified name for execution. After the subroutine is executed, the system returns to the command after the CALL command to continue execution.

Precautions

- The subroutine called in the CALL command must be defined beforehand in the user program. If an undefined subroutine appears in the CALL command, the program cannot be compiled.
- The element type of the operand carried in the CALL command should match the data type defined in the local variable table of the subroutine, otherwise the program cannot be compiled.
- The number of operands carried in the CALL command should match the local variable table of the subroutine, otherwise the program cannot be compiled.
- When the CALL command calls the program as a C language function, the subroutine name and parameter input follow the same rules described above.

The following examples explain illegal matching uses:

Example 1: In the local variable table of the SBR1 subroutine, the data type of operand 1 is DINT/DWORD.

The following uses are illegal:

1. CALL SBR1 Z0 (the Z element cannot be used for the data type DINT/DWORD).
2. CALL SBR1 C199 (elements C0~C199 cannot be used for the data type DINT/DWORD).

Example 2: In the local variable table of the SBR1 subroutine, the data type of operand 1 is INT/WORD.

The following uses are illegal:

1. CALL SBR1 C200 (elements C200~C255 cannot be used for the data type INT/WORD).
2. The element type of the operand carried in the CALL command should match the variable type defined in the local variable table of the subroutine, otherwise the program cannot be compiled.

The following examples explain illegal matching uses:

Example 3: In the local variable table of the SBR1 subroutine, the operand type of operand 1 is OUT or IN_OUT.

The following uses are illegal:

1. CALL SBR1 321 (the constant cannot be changed, and therefore it does not match the OUT or IN-OUT type operand).
2. CALL SBR1 X0 (X0 is read-only, and therefore it does not match the OUT or IN-OUT type operand).

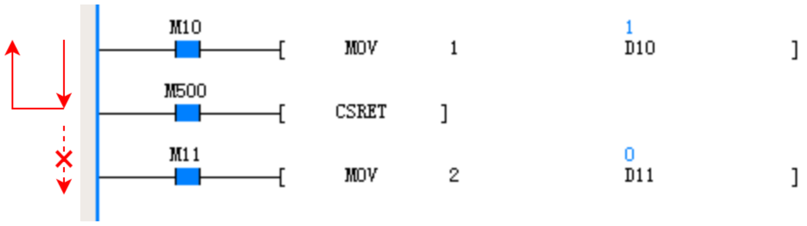
3.5.13 CSRET: Conditional Return of User Subroutine

Command table	CSRET			Applicable model	TS600 series			
16-Bit command	CSRET: Conditional return of user subroutine							
32-Bit command	-							
Operand	Type	Bit			Word		Indexing	Constant
		X, Y, M, LM, T, C, S	Dx.y	Custom bit variable	D, R, V, Z, T, C	Custom word variable		
-	-	-	-	-	-	-	-	-

Function Description

When the energy flow is valid, the system exits the subroutine being executed and returns to the subroutine at the previous level.

Application Example



In case of M500=ON, the system does not execute the programs after the CSRET command and returns to the main program.

3.6 Timing and Counting Command

3.6.1 Command Table

Command Category	Name	Function
Timing and Counting Command	TON	ON delay timing
	TONR	Memory-type ON delay timing
	TOF	OFF delay timing
	TMON	Non-retriggering single stable timing
	CTU	16-bit increment counter
	CTR	16-bit loop counter
	DCNT	32-bit increment-decrement counter

3.6.2 TON: ON Delay Timing Commands

Command table		TON (D) (S)			Applicable model	TS600 series		
16-Bit command		TON: ON delay timing						
32-Bit command		-						
Operand	Type	Bit			Word		Indexing	Constant
		X, Y, M, LM, T, C, S	Dx.y	Custom bit variable	D, R, V, Z, T, C	Custom word variable		
D	INT	-	-	-	√[1]	-	-	-
S	INT	-	-	-	√	√	√	√

Remark:

[1] The T element is supported.

Operand Description

D: The destination operand, which indicates the specified T element.

S: The source operand, which indicates the preset value of timing.

Function Description

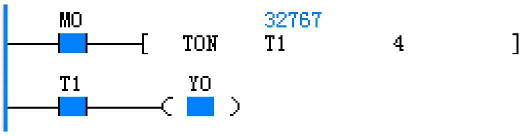
1. When the energy flow is valid and the timing value is ＜ 32767, the specified T element (D) is timed (the timing value accumulates over time). When the timing value reaches 32767, the timing value will remain unchanged at 32767.
2. When the timing value is ⩾ the preset value (S), the timing coil output of the specified T element is ON.
3. When the energy flow is OFF, timing stops, the timing value is reset to zero, and the timing coil output is OFF.

4. When the system executes this instruction for the first time, the timing coil value of the specified T element will be reset to OFF and the timing value will be reset to zero.

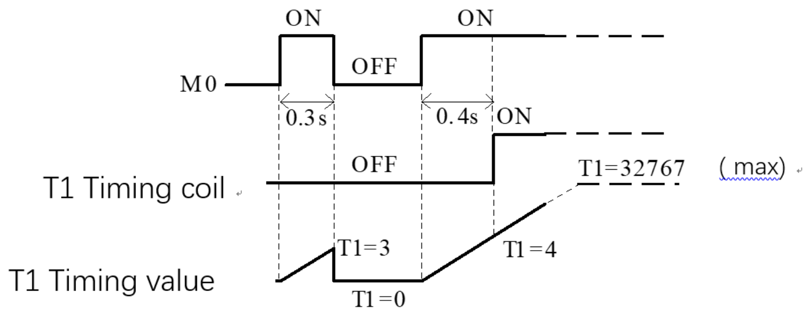
Precautions

The subscript value of the T element ranges between T0 and T399.

Application Example



Timing diagram



3.6.3 TONR: Memory-Type ON Delay Timing Commands

Command table	TONR (D) (S)				Applicable model	TS600 series		
16-Bit command	TONR: Memory-type ON delay timing							
32-Bit command	-							

Operand	Type	Bit			Word		Indexing	Constant
		X, Y, M, LM, T, C, S	Dx.y	Custom bit variable	D, R, V, Z, T, C	Custom word variable		
D	INT	-	-	-	√[1]	-	-	-
S	INT	-	-	-	√	√	√	√

Remark:

[1] The T element is supported.

Operand Description

D: The destination operand, which indicates the specified T element.

S: The source operand, which indicates the preset value of timing.

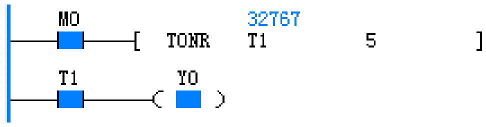
Function Description

1. When the energy flow is valid and the timing value is < 32767, the specified T element (D) is timed (the timing value increases over time). When the timing value reaches 32767, the timing value will remain unchanged at 32767.
2. When the timing value is ≥ the preset value (S), the timing coil output of the specified T element is ON.
3. When the energy flow is OFF, timing stops, and the timing coil and the timing value remain the current timing value.

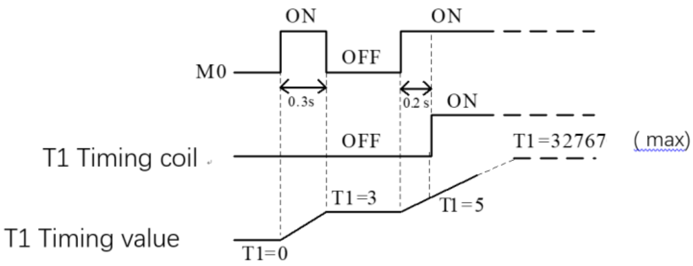
Precautions

The subscript value of the T element ranges between T0 and T399.

Application Example



Timing diagram



3.6.4 TOF: OFF Delay Timing Commands

Command table	TOF (D) (S)			Applicable model	TS600 series		
16-Bit command	TOF: OFF delay timing						
32-Bit command	-						

Operand	Type	Bit			Word		Indexing	Constant
		X, Y, M, LM, T, C, S	Dx.y	Custom bit variable	D, R, V, Z, T, C	Custom word variable		
D	INT	-	-	-	√[1]	-	-	-
S	INT	-	-	-	√	√	√	√

Remark:

[1] The T element is supported.

Operand Description

D: The destination operand, which indicates the specified T element.

S: The source operand, which indicates the preset value of timing.

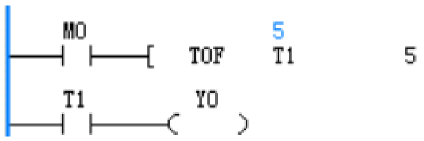
Function Description

1. When the energy flow changes from ON to OFF (falling edge), the specified timer T (D) starts timing.
2. When the energy flow is OFF, and the specified timer T has started timing, the timing is continued. Once the timing value equals the preset value (S), the timing coil output of the specified T element is OFF, and thereafter the timing value will remain unchanged at the preset value.
3. If the timing is not started, it will not be started even if the energy flow input is OFF.
4. When the energy flow is ON, timing stops, the timing value is reset to zero, and the timing coil output is ON.

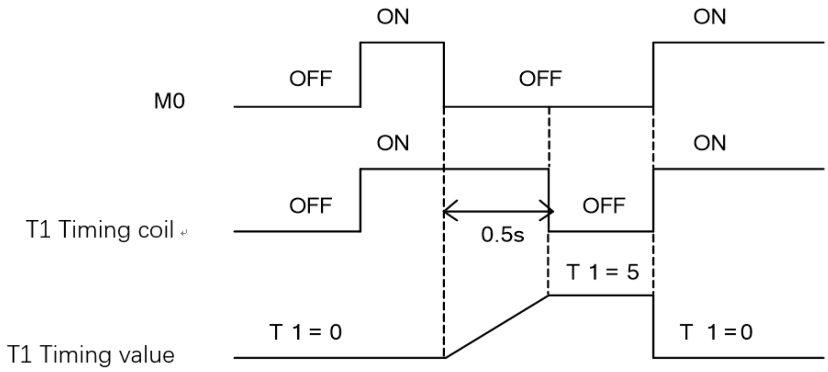
Precautions

The subscript value of the T element ranges between T0 and T399.

Application Example



Timing diagram



3.6.5 TMON: Non-Triggering Timing Commands

Command table		TMON (D) (S)			Applicable model	TS600 series		
16-Bit command		TMON: Non-retriggering single stable timing						
32-Bit command		-						
Operand	Type	Bit			Word		Indexing	Constant
		X, Y, M, LM, T, C, S	Dx.y	Custom bit variable	D, R, V, Z, T, C	Custom word variable		
D	INT	-	-	-	√[1]	-	-	-
S	INT	-	-	-	√	√	√	√

Remark:

[1] The T element is supported.

Operand Description

D: The destination operand, which indicates the specified T element.

S: The source operand, which indicates the preset value of timing.

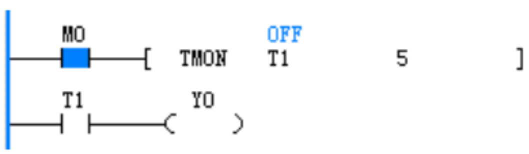
Function Description

1. When the input energy flow changes from OFF to ON (rising edge) and is in the non-timing state, the specified timer T (D) starts timing (from the current value). In the timing state (the timing length is determined by S), the timing coil output keeps ON.
2. In the timing state (the timing length is determined by S), the timing keeps and the timing coil output remains ON, regardless of how the energy flow changes.
3. When the timing value is reached, timing stops, the timing value is reset to zero, and the coil output is reset to OFF.

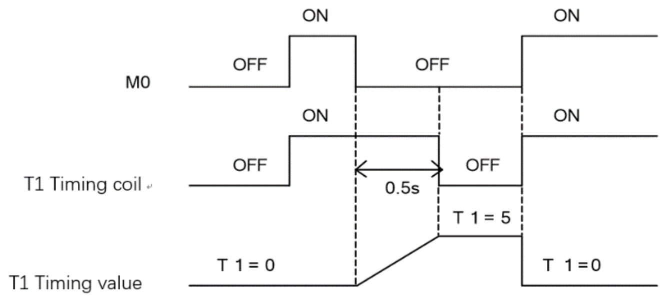
Precautions

The subscript value of the T element ranges between T0 and T399.

Application Example



Timing diagram



3.6.6 TPR: Pulse Timing Commands

Command table		TPR (P) (R) (O) (E)			Applicable model	TS600 series		
16-Bit command		TPR: Pulse timing						
32-Bit command		-						
Operand	Type	Bit			Word		Indexing	Constant
		X, Y, M, LM, T, C, S	Dx.y	Custom bit variable	D, R, V, Z, T, C	Custom word variable		
PreTime	DINT	-	-	-	√[2]	√	-	√
ReSet	BOOL	√[1]	-	√	-	-	-	-
OutPut	BOOL	√[1]	-	√	-	-	-	-
ElapTime	DINT	-	-	-	√[2]	√	-	-

Remark:

[1]The Y, M, and S elements are supported.

[2]The D and R elements are supported.

Operand Description

PreTime: The preset timing, measured in milliseconds. The set value ranges between 0 and 2147483647 ms (up to approximately 24 days); if the PT set value is ⩽ 0, the timing is done according to 0.

ReSet: The reset timer.

OutPut: The output result.

ElapTime: The elapsed time.

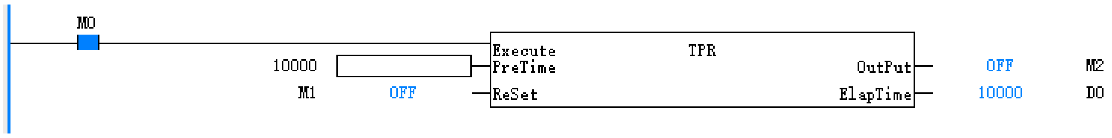
Function Description

1. When the IN input energy flow of the timer command changes from OFF to ON, the timer starts timing and the output Q becomes ON. At this point, Q remains ON within the time specified by the PT parameter, regardless of how the input energy flow of IN changes. After the timing reaches the time specified by the PT parameter, Q becomes OFF.
2. During the timing operation of the timer, ET outputs the current timing. After the timing of the timer reaches the time specified by the PT parameter: If the IN input energy flow is ON, the ET value remains; if the IN input energy flow is OFF, the ET value becomes 0.
3. During the timing process of the timer, if the reset input R changes from OFF to ON, the timing of the TPR timer is reset to 0, and the output Q becomes OFF. After the reset input becomes OFF, if the energy flow of IN is valid, the timing of the timer can be restored.

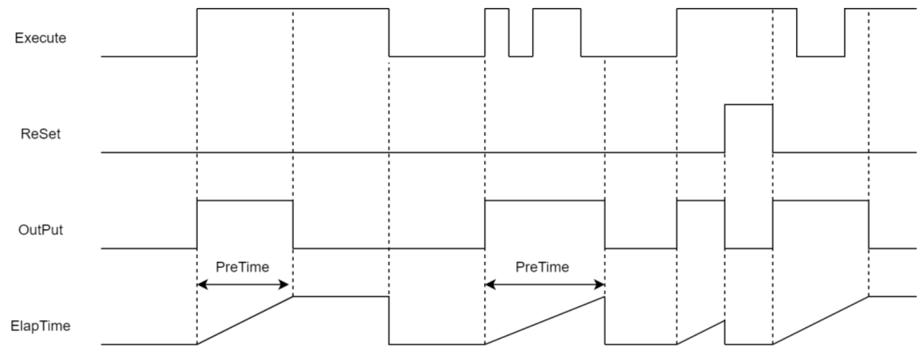
Precautions

Up to 1024 TPR commands are supported.

Application Example



Timing diagram



3.6.7 TONG: ON Delay Timing Commands

Command table		TONG (P) (R) (O) (E)			Applicable model	TS600 series		
16-Bit command		TONG: ON delay timing						
32-Bit command		-						
Operand	Type	Bit			Word		Indexing	Constant
		X, Y, M, LM, T, C, S	Dx.y	Custom bit variable	D, R, V, Z, T, C	Custom word variable		
PreTime	DINT	-	-	-	√[2]	√	-	√
ReSet	BOOL	√[1]	-	√	-	-	-	-
OutPut	BOOL	√[1]	-	√	-	-	-	-
ElapTime	DINT	-	-	-	√[2]	√	-	-

Remark:

[1] The Y, M, and S elements are supported.

[2] The D and R elements are supported.

Operand Description

PreTime: The preset timing, measured in milliseconds. The set value ranges between 0 and 2147483647 ms (up to approximately 24 days); if the PT set value is ⩽ 0, the timing is done according to 0.

ReSet: The reset timer.

OutPut: The output result.

ElapTime: The elapsed time.

Function Description

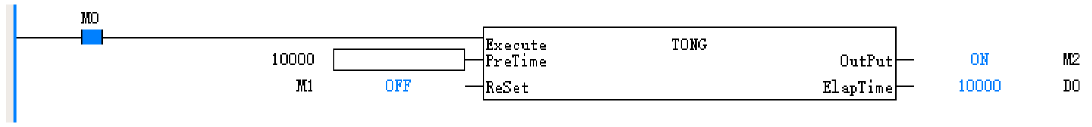
1. When the timer command IN changes the input energy flow from OFF to ON, the timer starts timing and the output Q becomes OFF. During the period when the IN input energy flow remains ON, the running time of the timer is the time specified by the PT parameter. After the timing reaches the time specified by the PT parameter, Q becomes ON. During or after the timing process, if the IN input energy flow becomes OFF, the timing ends, and Q becomes OFF. During this period, when the IN input energy flow is OFF, the output Q remains OFF.
2. During the timing operation of the timer with the IN input energy flow being ON, ET outputs the current timing. After the timing of the timer reaches the time specified by the PT parameter, the ET value remains unchanged; if the IN input energy flow is OFF, the ET value becomes 0.

3. During the timing process of the timer, if the reset input R changes from OFF to ON, the timing of the TONR timer is reset to 0, and the output Q becomes OFF. After the reset input R becomes OFF, it is necessary to change the IN input energy flow from OFF to ON again in order to restore the timing of the timer.

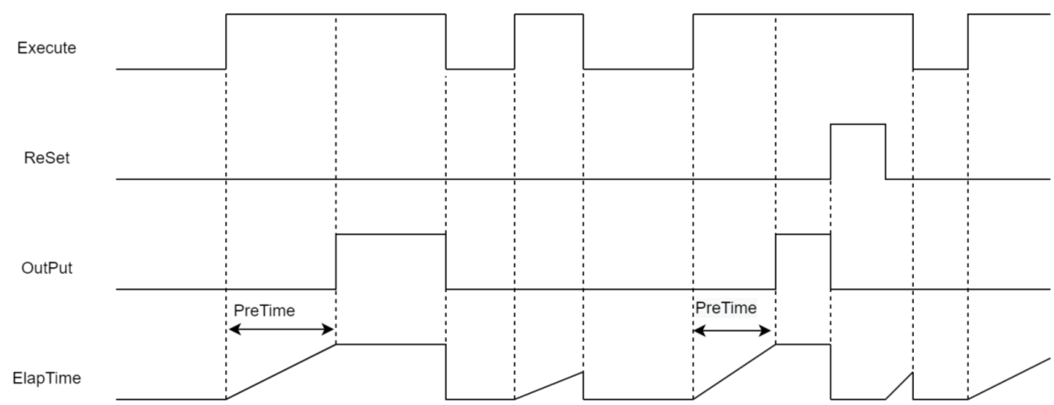
Precautions

Up to 1024 TONG commands are supported.

Application Example



Timing diagram



3.6.8 TOFG: OFF Delay Timing Commands

Command table		TOFG (P) (R) (O) (E)			Applicable model	TS600 series		
16-Bit command		TOFG: OFF delay timing						
32-Bit command		-						
Operand	Type	Bit			Word		Indexing	Constant
		X, Y, M, LM, T, C, S	Dx.y	Custom bit variable	D, R, V, Z, T, C	Custom word variable		
PreTime	DINT	-	-	-	√[2]	√	-	√
ReSet	BOOL	√[1]	-	√	-	-	-	-
OutPut	BOOL	√[1]	-	√	-	-	-	-
ElapTime	DINT	-	-	-	√[2]	√	-	-

Remark:

[1] The Y, M, and S elements are supported.

[2] The D and R elements are supported.

Operand Description

PreTime: The preset timing, measured in milliseconds. The set value ranges between 0 and 2147483647 ms (up to approximately 24 days); if the PT set value is ⩽ 0, the timing is done according to 0.

ReSet: The reset timer.

OutPut: The output result.

ElapTime: The elapsed time.

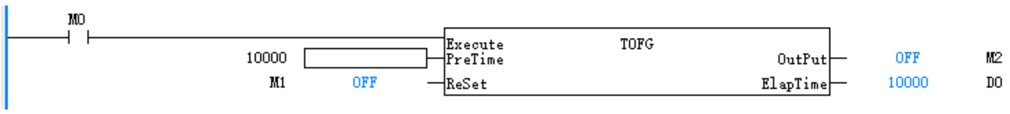
Function Description

1. When the IN input energy flow of the timer command changes from OFF to ON, the timer starts timing and the output Q becomes ON. When the IN input energy flow changes from ON to OFF with IN remaining OFF, the running time of the timer is the time specified by the PT parameter. When the timing of the timer reaches the time specified by the PT parameter, Q becomes OFF. During the period when the IN input energy flow is OFF, Q remains OFF.
2. When the IN input energy flow is ON, the ET output value becomes 0. When IN changes from ON to OFF, during the timing operation of the timer, ET outputs the current timing. After the timing of the timer reaches the time specified by the PT parameter, the ET value remains unchanged.
3. When the IN input energy flow is ON, if the reset input R changes from OFF to ON, the output Q becomes OFF; if R returns to OFF, the output Q returns to ON. When the IN input energy flow changes from ON to OFF, for the TOFR timer during or after the timing process, if the reset input R changes from OFF to ON, the output Q becomes OFF, and ET is reset to 0. After the reset input R becomes OFF, it is necessary to change the IN input energy flow from ON to OFF again in order to restore the timing of the timer.

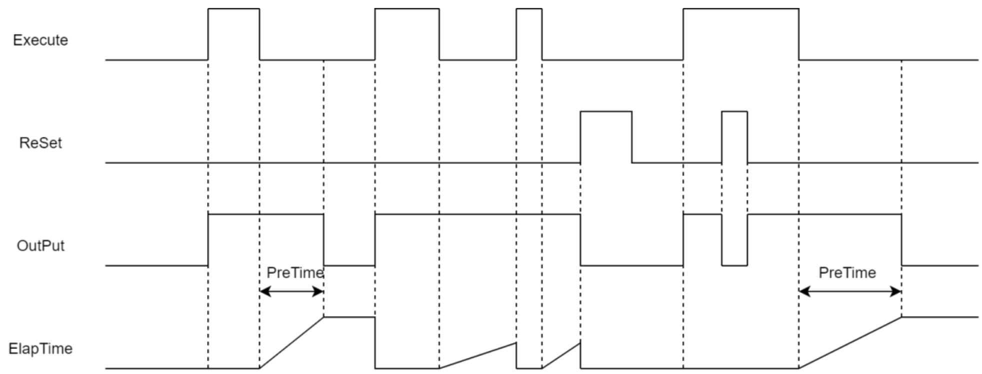
Precautions

Up to 1024 TOFG commands are supported.

Application Example



Timing diagram



3.6.9 TACR: Temporal Accumulation Timing Commands

Command table		TACR (P) (R) (O) (E)			Applicable model	TS600 series		
16-Bit command		TACR: Temporal accumulation timing						
32-Bit command		-						
Operand	Type	Bit			Word		Indexing	Constant
		X, Y, M, LM, T, C, S	Dx.y	Custom bit variable	D, R, V, Z, T, C	Custom word variable		
PreTime	DINT	-	-	-	√[2]	√	-	√
ReSet	BOOL	√[1]	-	√	-	-	-	-
OutPut	BOOL	√[1]	-	√	-	-	-	-
ElapTime	DINT	-	-	-	√[2]	√	-	-

Remark:

[1] The Y, M, and S elements are supported.

[2] The D and R elements are supported.

Operand Description

PreTime: The preset timing, measured in milliseconds. The set value ranges between 0 and 2147483647 ms (up to approximately 24 days); if the PT set value is ⩽ 0, the timing is done according to 0.

ReSet: The reset timer.

OutPut: The output result.

ElapTime: The elapsed time.

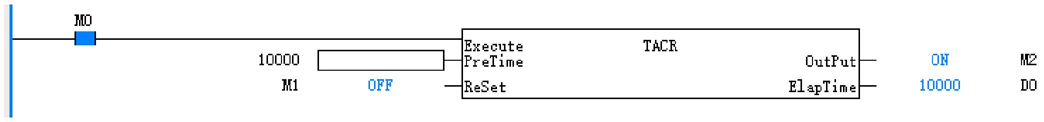
Function Description

1. When the IN input energy flow of the timer command is ON, if the timing value of the timer does not reach the time specified by the PT parameter, the timer continues to count and the output Q becomes OFF. When the timer timing time reaches the time specified by the PT parameter, Q becomes ON. During the timing period of the timer with IN being ON, if IN becomes OFF, the timing of the timer remains unchanged. After IN becomes ON again, the timer starts counting from the current holding value. After the time specified by the PT parameter is reached, Q becomes ON.
2. When the IN input energy flow is ON, ET outputs the current timing value. After the timing reaches the time specified by the PT parameter, the ET value remains. When the IN input energy flow is OFF, ET remains unchanged.
3. For the timer during or after the timing process, if the reset input R changes from OFF to ON, the output Q becomes OFF, and ET is reset to 0. After the reset input R becomes OFF, it is necessary to change the IN input energy flow from OFF to OFF again in order to restore the timing of the timer.

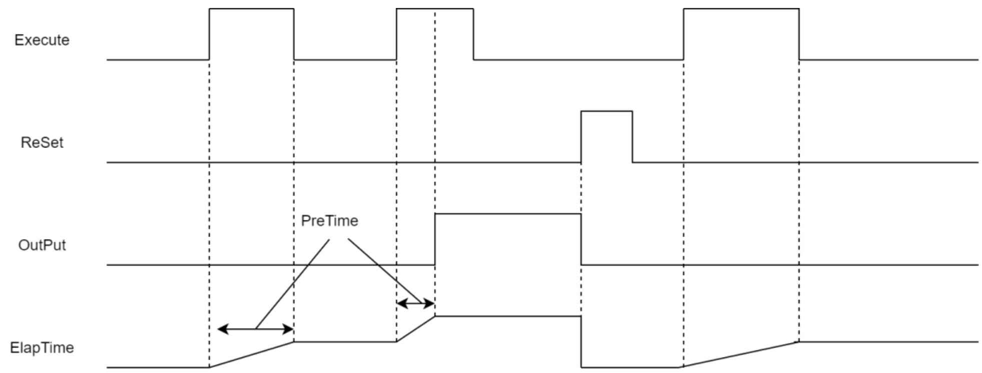
Precautions

Up to 1024 TACR commands are supported.

Application Example



Timing diagram



3.6.10 CTU: 16-Bit Increment Counter Commands

Command table		CTU (D) (S)			Applicable model	TS600 series		
16-Bit command		CTU: 16-bit increment counter						
32-Bit command		-						
Operand	Type	Bit			Word		Indexing	Constant
		X, Y, M, LM, T, C, S	Dx.y	Custom bit variable	D, R, V, Z, T, C	Custom word variable		
D	INT	-	-	-	√[1]	-	-	-
S	INT	-	-	-	√	√	√	√

Remark:

[1] The C element is supported.

Operand Description

D: The destination operand, which indicates the specified C element.

S: The source operand, which indicates the preset counting value.

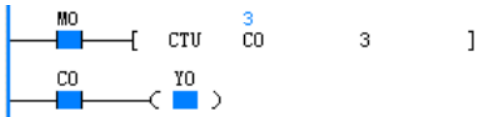
Function Description

1. When the energy flow changes from OFF to ON (rising edge), the counting value of the specified 16-bit counter C (D) increases by one.
2. After reaching 32767, the counting value remains unchanged.
3. When the counting value is ⩾ the preset counting value (S), the counting coil is set to ON.

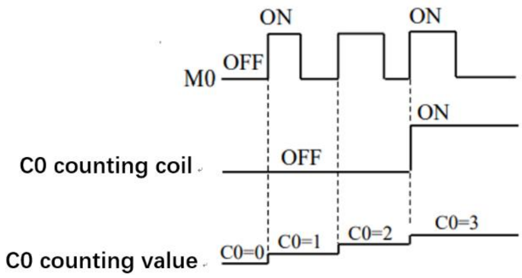
Precautions

The subscript values of the C element ranges between 0 and 199.

Application Example



Timing diagram



3.6.11 CTR: 16-Bit Loop Counter Commands

<table>
<tr><td colspan="2">Command table</td><td colspan="3">CTR (D) (S)</td><td>Applicable model</td><td colspan="3">TS600 series</td></tr>
<tr><td colspan="2">16-Bit command</td><td colspan="7">CTR: 16-bit loop counter</td></tr>
<tr><td colspan="2">32-Bit command</td><td colspan="7">-</td></tr>
<tr><td rowspan="2">Operand</td><td rowspan="2">Type</td><td colspan="3">Bit</td><td colspan="2">Word</td><td rowspan="2">Indexing</td><td rowspan="2">Constant</td></tr>
<tr><td>X, Y, M, LM, T, C, S</td><td>Dx.y</td><td>Custom bit variable</td><td>D, R, V, Z, T, C</td><td>Custom word variable</td></tr>
<tr><td>D</td><td>INT</td><td>-</td><td>-</td><td>-</td><td>√[1]</td><td>-</td><td>-</td><td>-</td></tr>
<tr><td>S</td><td>INT</td><td>-</td><td>-</td><td>-</td><td>√</td><td>√</td><td>√</td><td>√</td></tr>
</table>

Remark:

[1] The C element is supported.

Operand Description

D: The destination operand, which indicates the specified C element.

S: The source operand, which indicates the preset counting value.

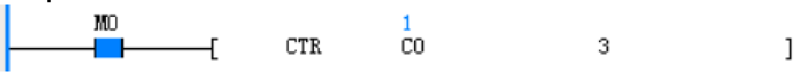
Function Description

1. When the input energy flow changes from OFF to ON (rising edge), the counting value of the specified 16-bit counter C (D) increases by one.
2. When the counting value is equal to the preset counting value (S), the counting coil is set to ON.
3. When the counting value is equal to the preset counting value (S), if the input energy flow changes from OFF to ON (rising edge) once again, the counting value is set to 1 and the counting coil is reset to OFF.

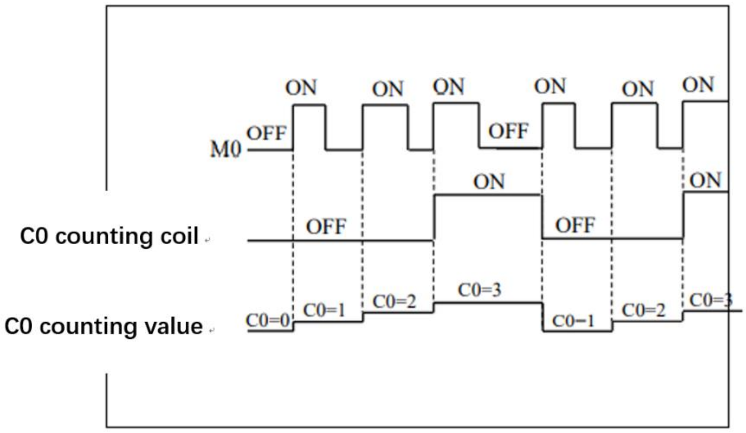
Precautions

The subscript values of the C element ranges between 0 and 199.

Application Example



Timing diagram



3.6.12 DCNT: 32-Bit Increment-Decrement Counter Commands

Command table	DCNT (D) (S1) (S2)				Applicable model	TS600 series		
16-Bit command	DCNT: 32-bit increment-decrement counter							
32-Bit command	-							
Operand	Type	Bit			Word		Indexing	Constant
		X, Y, M, LM, T, C, S	Dx.y	Custom bit variable	D, R, V, Z, T, C	Custom word variable		
D	DINT	-	-	-	√[1]	-	-	-
S1	DINT	-	-	-	√[2]	√	√	√
S2	BOOL	√[3]	-	√	-	-	-	√

Remark:

[1]Only the C element is supported.

[2]The Z and T elements are not supported.

[3]Only the M element is supported.

Operand Description

D: The destination operand, which indicates the specified C element (C200~C255).

S1: Source operand 1, which indicates the preset counting value.

S2: Source operand 2, which indicates the counting direction flag bit, where OFF means increasing and ON means decreasing.

Function Description

1. When the input energy flow changes from OFF to ON (rising edge), the counting value of the specified 32-bit counter C (D) increases or decrease by 1 (the counting direction, either increasing or decreasing, depends on the S2 operand).
2. For the increment counter, when the counting value is ⩾ the preset counting value (S), the counting coil is set to ON.
3. For the decrement counter, when the counting value is ⩽ the preset counting value (S), the counting coil is set to OFF.

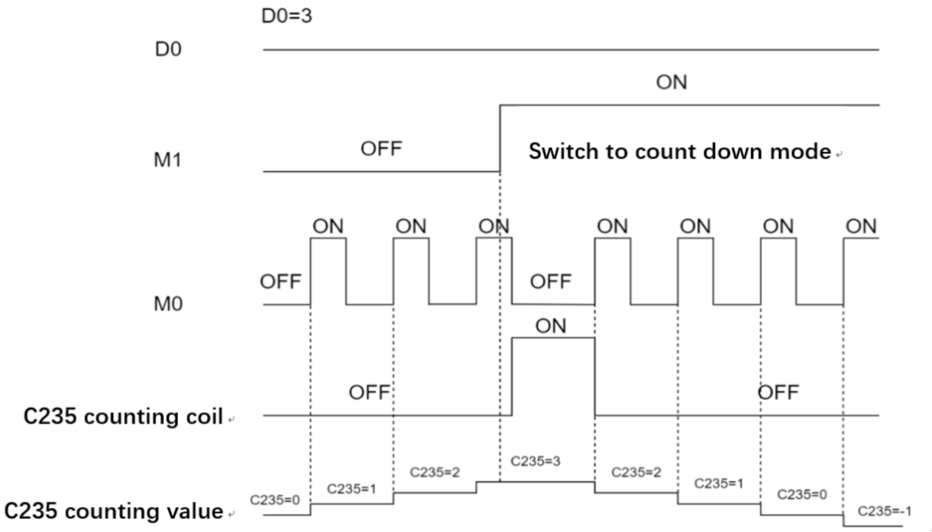
4. When the counting value is 2147483647, if the timer increases the counting by 1 once more, the count value becomes -2147483648.
5. When the counting value is -2147483648, if the timer decreases the counting by 1 once more, the count value becomes 2147483647.

Precautions

The subscript value of the C element ranges between C200 and C255.

Application Example

Timing diagram



3.7 Data Transmission Command

3.7.1 Command Table

Command Category	Name	Function
Data Transmission Command	*MOV	Word/doubleword data transmission
	RMOV	Floating-point number data transmission
	BMOV	Block data transmission
	FMOV	Data block word/doubleword stuffing
	SMOV	Word/doubleword shift transmission
	SWAP	High-low byte swap
	*XCH	Word/doubleword exchange
	PUSH	Data push
	FIFO	First in first out
	LIFO	Last in first out
	WSFR	Word string shift right
	WSFL	Word string shift left

3.7.2 MOV: Word/Doubleword Data Transmission Commands

Command table	*MOV (S) (D)	Applicable model	TS600 series
16-Bit command	MOV: Word data transmission		
32-Bit command	DMOV: Doubleword data transmission		

Operand	Type	Bit			Word		Indexing	Constant
		X, Y, M, LM, T, C, S	Dx.y	Custom bit variable	D, R, V, Z, T, C	Custom word variable		
S	INT/DINT	-	-	-	√[1]	√	√	√
D	INT/DINT	-	-	-	√[1]	√	√	-

Remark:

[1] For 32-bit commands, the T and Z elements are not supported.

Operand Description

S: The source operand.

D: The destination operand.

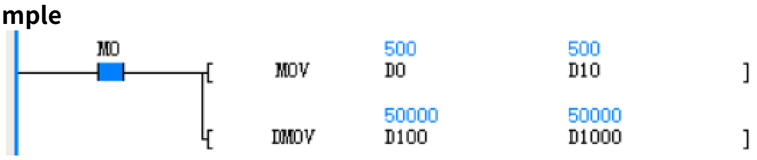
Function Description

When the energy flow is valid, the content of S is assigned to D, and the value of S remains unchanged.

Precautions

The MOV command supports both signed and unsigned integers. If both operands of the command are soft elements, then their data types are both signed integers. If the source operand of the command is a signed long integer such as (-10, +100), the destination operand is also a signed integer. If the source operand of the command is an unsigned long integer such as (100, 45535), the destination operand is also a unsigned integer.

Application Example



In case of M0=ON, the content of D0 is assigned to D10 to obtain D10=500, or the content of (D0, D1) is assigned to (D10, D11) to obtain (D10, D11)=50000.

3.7.3 RMOV: Floating-Point Number Data Transmission Commands

Command table	RMOV (S) (D)				Applicable model	TS600 series		
16-Bit command	-							
32-Bit command	RMOV: Floating-point number data transmission							
Operand	Type	Bit			Word		Indexing	Constant
		X, Y, M, LM, T, C, S	Dx.y	Custom bit variable	D, R, V, Z, T, C	Custom word variable		
S	REAL	-	-	-	√[1]	√	√	√
D	REAL	-	-	-	√[1]	√	√	-

Remark:

[1] Only the D, R, and V elements are supported.

Operand Description

S: The source operand.

D: The destination operand.

Function Description

When the energy flow is valid, the content of S is assigned to D, and the value of S remains unchanged.

Application Example

In case of M0=ON, the content of (D0, D1) is assigned to (D10, D11) to obtain (D10, D11)=50000.00.

3.7.4 BMOV: Block Data Transmission Commands

Command table	BMOV (S1) (D) (S2)				Applicable model	TS600 series		
16-Bit command	BMOV: Block data transmission							
32-Bit command	-							
Operand	Type	Bit			Word		Indexing	Constant
		X, Y, M, LM, T, C, S	Dx.y	Custom bit variable	D, R, V, Z, T, C	Custom word variable		
S1	INT, Array*S2	-	-	-	√[1]	√	√	-
D	INT, Array*S2	-	-	-	√[1]	√	√	-
S2	INT	-			√	√	√	√

Remark:

[1] The Z element is not supported.

Operand Description

S1: The source operand, which indicates the starting unit of a data block.

D: The destination operand, which indicates the starting unit of a data block.

S2: The data block size.

Function Description

When the energy flow is valid, the content of the S2 units starting from the S1 unit is assigned to the S2 units starting from the D unit, and the content of the S2 units starting from the S1 unit remains unchanged.

Application Example

In case of M0=ON, the contents of the 10 units starting from D0 are assigned to the 10 units starting from D100. D100=D0, D101=D1, ..., D109=D9.

3.7.5 FMOV: Data Block Word/Doubleword Stuffing Commands

Command table	FMOV (S1) (D) (S2)				Applicable model	TS600 series		
16-Bit command	FMOV: Data block word stuffing							
32-Bit command	DFMOV: Data block doubleword stuffing							
Operand	Type	Bit			Word		Indexing	Constant
		X, Y, M, LM, T, C, S	Dx.y	Custom bit variable	D, R, V, Z, T, C	Custom word variable		
S1	INT/DINT	-	-	-	√[1]	√	√	√
D	INT/DINT, Array*S2	-	-	-	√[2]	√	√	-
S2	INT	-			√	√	√	√

Remark:

[1]For the 32-bit command DFMOV, the T and Z elements are not supported.

[2]For the 16-bit command FMOV, the Z element is not supported; for the 32-bit command DBMOV, the Z and T elements are not supported.

Operand Description

S1: The source operand, which indicates the starting unit of a data block.

D: The destination operand, which indicates the starting unit of a data block.

S2: The data block size.

Function Description

When the energy flow is valid, the content of the S1 unit is stuffed into the S2 units starting from the D unit, and the content of the S1 unit remains unchanged.

Application Example

In case of M0=ON, the content of D0 is stuffed into the 10 units starting from D100. D100 = D101 = …… = D109 = D0 = 500.

In case of M0=ON, the content of (D0, D1) is stuffed into the 10 × 2 units starting from D10. (D10, D11) = (D12, D13) = …… = (D28, D29) = (D0, D1) = 100000.

3.7.6 SMOV: Word/Doubleword Shift Transmission Commands

Command table		SMOV (S1) (S2) (S3) (D) (n)			Applicable model	TS600 series		
16-Bit command		SMOV: Word shift transmission						
32-Bit command		DSMOV: Doubleword shift transmission						
Operand	Type	Bit			Word		Indexing	Constant
		X, Y, M, LM, T, C, S	Dx.y	Custom bit variable	D, R, V, Z, T, C	Custom word variable		
S1	INT/DINT	-	-	-	√[1]	√	-	√
S2	INT/DINT	-	-	-	√[1]	√	-	√
S3	INT/DINT	-	-	-	√[1]	√	-	√
D	INT/DINT	-	-	-	√[1]	√	-	-
n	INT/DINT	-	-	-	√[1]	√	-	√

Remark:

[1] The Z, C, and T soft elements are not supported.

Operand Description

The operands need to be driven by the contact, and there are up to 5 operational variables, among which:

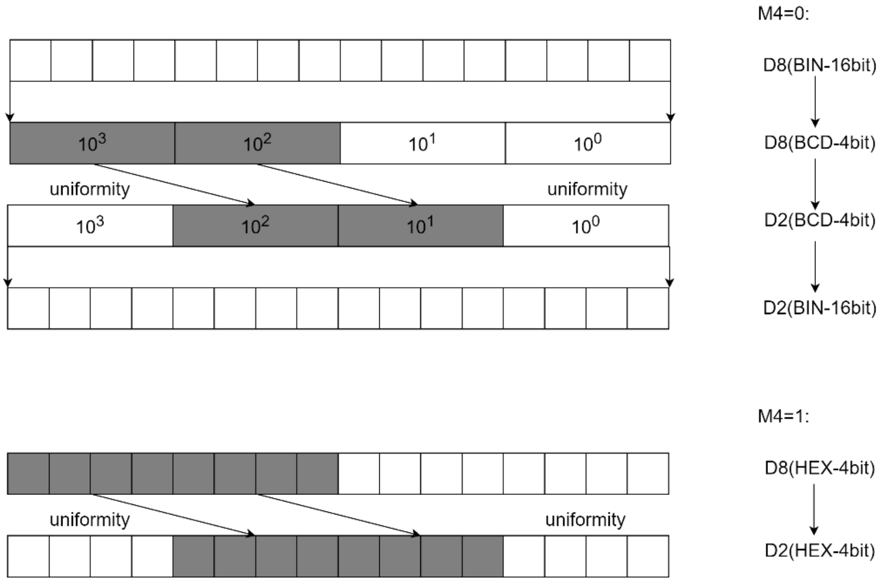
- S1 is the data source variable to be copied. When SM34 is OFF, it is in the BCD mode (decimal bit). The S1 operand ranges from 0000 to 9999 or 00000000 to 99999999 and cannot be negative. When SM34 is ON, it is in the BIN mode, and the S1 operand can be negative.
- S2 is the starting bit number for data source transmission and ranges from 1 to 4 or 1 to 8.
- S3 is the number of bits for data source transmission and ranges from or 1 to S2.
- D is the destination variable for data source transmission.
- n is the starting bit of the destination variable for data source transmission and ranges from S3 to S4 or S3 to S8.

The transmission process of data bits is related to the state of the special flag SM34. When SM34 is OFF, they are in the BCD mode (decimal); when SM34 is ON, they are is in the BIN mode, where every 4 bits (hexadecimal) are transmitted as a unit.

Function Description

Move the digit data of a total of S3 digits starting from the S2 digit (low to high) in S1 to a total of S3 digits starting from the n digit in destination D.

Application Example



Assuming D8=1234 and D2=5678, if M0 is set to ON with SM34=OFF (in the BCD mode), the value of D2 becomes 5128;

assuming D8=0x04D2=1234 and D2=0x162E=5678, if M0 is set to ON with SM34=ON (in the BIN mode), D2=0x104E=4174 is obtained.

3.7.7 SWAP: High-Low Byte Swap Commands

Command table		SWAP (D)			Applicable model	TS600 series		
16-Bit command		SWAP: High-low byte swap						
32-Bit command		-						
Operand	Type	Bit			Word		Indexing	Constant
		X, Y, M, LM, T, C, S	Dx.y	Custom bit variable	D, R, V, Z, T, C	Custom word variable		
D	INT	-	-	-	√	√	√	-

Operand Description

D: The destination operand, which indicates the word element for high-low byte swap.

Function Description

When the energy flow is valid, the value of the content of D after high-low byte swap is saved to the D unit.

Application Example

In case of M0=ON, the value of the content of D0=0x1027 (4135) after high-low byte swap is saved to D0, and D0=0x2710 (10000) is obtained.

3.7.8 XCH: Word Exchange Commands

Command table	XCH (D1) (D2)			Applicable model	TS600 series		
16-Bit command	XCH: Word exchange command						
32-Bit command	DXCH: Doubleword exchange command						
Operand	Type	Bit			Word		Indexing	Constant
		X, Y, M, LM, T, C, S	Dx.y	Custom bit variable	D, R, V, Z, T, C	Custom word variable		
D1	INT/ DINT	-	-	-	√[1]	√	√	-
D2	INT/ DINT	-	-	-	√[1]	√	√	-

Remark:

[1] For 32-bit commands, the Z and T soft elements are not supported.

Operand Description

D1: Destination operand 1

D2: Destination operand 2

Function Description

When the energy flow is valid, the values of the contents of D1 and D2 after exchange are saved to the D1 and D2 units.

Application Example

In case of M0=ON, the content of D0 is exchanged with that of D10:

Before execution: D0=5000, and D10=1000; after execution: D0=1000, and D10=5000.

In case of M1=ON, the content of (D0, D1) is exchanged with that of (D10, D11):

Before execution: (D0, D1)=5000000, and (D10, D11)=1000000; after execution: (D0, D1)=1000000, and (D10, D11)=5000000.

3.7.9 PUSH: Data Push Commands

Command table		PUSH (S1) (D) (S2)			Applicable model	TS600 series		
16-Bit command		PUSH: Data push						
32-Bit command		-						
Operand	Type	Bit			Word		Indexing	Constant
		X, Y, M, LM, T, C, S	Dx.y	Custom bit variable	D, R, V, Z, T, C	Custom word variable		
S1	INT	-	-	-	√	√	√	√
D	INT, Array*[S2]+1	-	-	-	√[1]	√	√	√
S2	WORD	-	-	-	√	√	√	√

Remark:

[1] The Z, T, and C elements are not supported.

Operand Description

S1: The push value.

D: The number of elements in the storage stack, where their labels represent the positions of the stack bottom.

S2: The stack size.

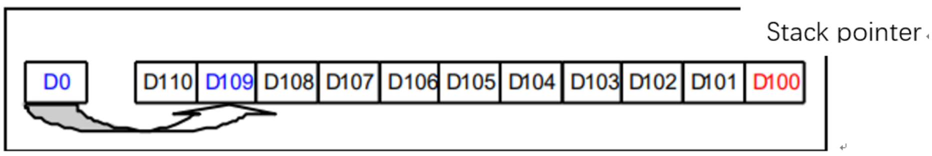
Function Description

1. When the energy flow is valid, the value of S1 is pushed into the stack top with the D unit as the bottom, and the value of D increases by 1. At this point, the number of the stack top unit is: the number of D + the value of D.
2. When the value of D is equal to the value of S2, there are still push commands to be executed, and the carry flag bit (SM20) is set to 1, and no stack pushing operation is performed.

Precautions

- When the stack definition being operated is illegal (that is, the stack size is less than or equal to zero, the number of elements in the stack is less than zero, or the number of elements in the stack exceeds the stack size limit), an error related to the illegal stack definition is reported.
- The stack size does not include the stack bottom element (the element specified by D).

Application Example



In case of M0=ON, the content of D0 is pushed into the stack with D100 as the stack bottom:

Before execution: D0=1000, D100=8, and D109=0; after execution: D0=1000, D100=9, and D109=1000.

3.7.10 FIFO: First In First Out Commands

Command table	FIFO (D1) (D2) (S)				Applicable model	TS600 series		
16-Bit command	FIFO: First in first out							
32-Bit command	-							
Operand	Type	Bit			Word		Indexing	Constant
		X, Y, M, LM, T, C, S	Dx.y	Custom bit variable	D, R, V, Z, T, C	Custom word variable		
D1	INT, Array* *[S]+1	-	-	-	√[1]	√	√	-
D2	INT	-	-	-	√	√	√	-
S	WORD	-	-	-	√	√	√	√

Remark:

[1] The Z, T, and C elements are not supported.

Operand Description

D1: The number of elements in the queue, where the element with the element number + 1 is the first element in the queue.

D2: The storage unit of the output queue value.

S: The queue size.

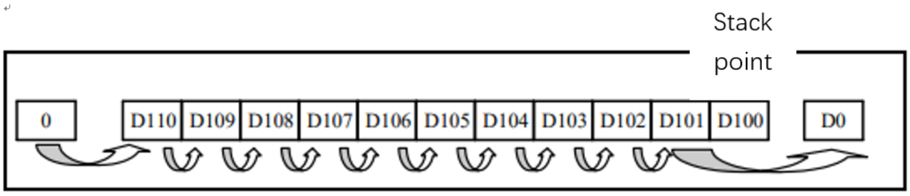
Function Description

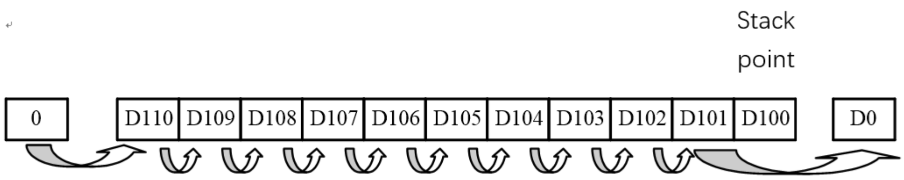
1. When the energy flow is valid, the first value of the word queue starting with D1 (the content of the next unit after D1) is assigned to the D2 unit, and the value of D1 decreases by 1, the contents of the S units after D1 move from back to front, and the last unit is filled with 0.
2. If the value of D1 is equal to 0, this indicates that the queue is empty, and the zero flag bit (SM18) is set to 1.

Precautions

- When the queue definition being operated is illegal (that is, the queue size is less than or equal to zero, the number of elements in the queue is less than zero, or the number of elements in the queue exceeds the queue size limit), an error related to the illegal queue definition is reported.
- The queue size does not include the queue bottom element (the element specified by D1).
- S indicates the queue size, which has a range greater than 0.

Application Example





In case of M0=ON, the content of D101 is stuffed into D0, the contents of units D101 to D110 move from back to front, and the content of D110 is filled with 0.

Before execution: D0=0, D100=10, D101=1000, D102=2000, ……, D109=9000, D110=10000;

After execution: D0=1000, D100=9, D101=2000, D102=3000, ……, D109=10000, D110=0.

3.7.11 LIFO: Last In First Out Commands

Command table		LIFO (D1) (D2) (S)			Applicable model	TS600 series		
16-Bit command		LIFO: Last in first out						
32-Bit command		-						
Operand	Type	Bit			Word		Indexing	Constant
		X, Y, M, LM, T, C, S	Dx.y	Custom bit variable	D, R, V, Z, T, C	Custom word variable		
D1	INT, Array**[S]+1	-	-	-	√[1]	√	√	-
D2	INT	-	-	-	√	√	√	-
S	WORD	-	-	-	√	√	√	√

Remark:

[1] The Z, T, and C elements are not supported.

Operand Description

D1: The number of elements in the stack, where the element with the element number + 1 is the first element in the stack.

D2: The storage unit of the output stack value.

S: The stack size.

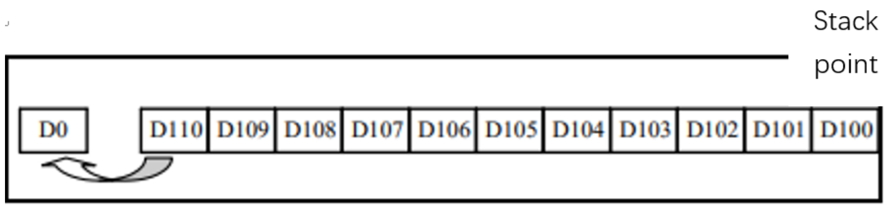
Function Description

1. When the energy flow is valid, the content of the stack top unit with D1 as the stack bottom is assigned to the D2 unit, and the value of D1 decreases by 1.
2. If the value of D1 is equal to 0, this indicates that the stack is empty, and the zero flag bit (SM18) is set to 1.

Precautions

- When the stack definition being operated is illegal (that is, the stack size is less than or equal to zero, the number of elements in the stack is less than zero, or the number of elements in the stack exceeds the stack size limit), an error related to the illegal stack definition is reported.
- The stack size does not include the stack bottom element (the element specified by D1).
- S indicates the stack size, which has a range greater than 0.

Application Example



In case of M0=ON, the content of D110 is assigned to D0, and the contents of units D101 to D110 remain unchanged:

Before execution: D0=0, D100=10, D101=1000, D102=2000, ⋯⋯, D109=9000, D110=10000;

After execution: D0=10000, D100=9, D101=1000, D102=2000, ⋯⋯, D109=9000, D110=10000.

3.7.12 WSFR: Word String Shift Right Commands

Command table		WSFR (S1) (D) (S2) (S3)			Applicable model	TS600 series		
16-Bit command		WSFR: Word string shift right						
32-Bit command		-						
Operand	Type	Bit			Word		Indexing	Constant
		X, Y, M, LM, T, C, S	Dx.y	Custom bit variable	D, R, V, Z, T, C	Custom word variable		
S1	INT, Array*S3	-	-	-	√[1]	√	√	-
D	INT, \Array*S2	-	-	-	√[1]	√	√	-
S2	INT	-	-	-	√	√	√	√
S3	WORD	-	-	-	√	√	√	√

Remark:

[1] The Z element is not supported.

Operand Description

S1: The source operand.

D2: The destination operand, which indicates the starting element of the word string.

S2: The size of the destination word queue.

S3: The number of words filled in after right shift.

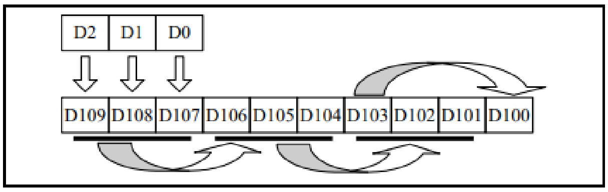
Function Description

When the energy flow is valid, the contents (taking a word as the unit) of the S2 units starting from the D unit is shifted to the right for S3 units, the rightmost S3 data are discarded, the contents of the S3 units starting from the S1 unit are shifted into the left end of the word string.

Precautions

- For the left-right order, the elements with smaller numbers indicate right, while those with larger numbers indicate left.
- S2 is greater than or equal to 0, and S3 is greater than or equal to 0.
- S2 is greater than or equal to and S3.

Application Example



In case of M0=ON, the contents (taking a word as the unit) of the 10 units starting from the D100 unit are shifted to the right for 3 units, and the data of the rightmost D102 to D100 units are discarded. At the same time, the contents of the 3 units starting from the D0 unit are shifted into the left end of the word string:

Before execution: D2=300, D1=200, and D0=100. D109=10000, D108=9000, D107=8000, D106=7000, D105=6000, D104=5000, D103=4000, D102=3000, D101=2000, and D100=1000;

After execution: The contents of D0 to D2 remain unchanged. D2=300, D1=200, and D0=100. D109=300, D108=200, D107=100, D106=10000, D105=9000, D104=8000, D103=7000, D102=6000, D101=5000, and D100=4000.

3.7.13 WSFL: Word String Shift Left Commands

Command table	WSFL (S1) (D) (S2) (S3)	Applicable model	TS600 series
16-Bit command	WSFL: Word string shift left		
32-Bit command	-		

Operand	Type	Bit			Word		Indexing	Constant
		X, Y, M, LM, T, C, S	Dx.y	Custom bit variable	D, R, V, Z, T, C	Custom word variable		
S1	INT, Array*S3	-	-	-	√[1]	√	√	-
D	INT, Array*S2	-	-	-	√[1]	√	√	-
S2	INT	-	-	-	√	√	√	√
S3	WORD	-	-	-	√	√	√	√

Remark:

[1] The Z element is not supported.

Operand Description

S1: The source operand.

D2: The destination operand, which indicates the starting element of the word string.

S2: The size of the destination word queue.

S3: The number of words filled in after right shift.

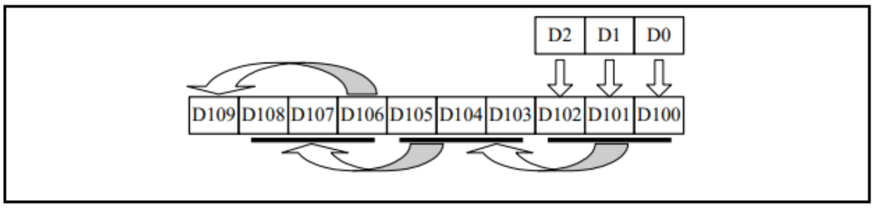
Function Description

When the energy flow is valid, the contents (taking a word as the unit) of the S2 units starting from the D unit are shifted to the left for S3 units, the leftmost S3 data are discarded, the contents of the S3 units starting from the S1 unit are shifted into the right end of the word string.

Precautions

- For the left-right order, the elements with smaller numbers indicate right, while those with larger numbers indicate left.
- S2 is greater than or equal to 0, and S3 is greater than or equal to 0.
- S2 is greater than or equal to and S3.

Application Example



In case of M0=ON, the contents (taking a word as the unit) of the first 10 units starting from the D100 unit are shifted to the left by 3 units, and the data of the leftmost D109 to D107 units are discarded. At the same time, the contents of the 3 units starting from the D0 unit are shifted into the right end of the string:

Before execution: D0=100, D1=200, and D2=300. D109=10000, D108=9000, D107=8000, D106=7000, D105=6000, D104=5000, D103=4000, D102=3000, D101=2000, and D100=1000;

After execution: The contents of D0 to D2 remain unchanged. D2=300, D1=200, and D0=100. D109=7000, D108=6000, D107=5000, D106=4000, D105=3000, D104=2000, D103=1000, D102=300, D101=200, and D100=100.

3.8 Arithmetic Operation Command for Integers

3.8.1 Command Table

Command Category	Name	Function
Arithmetic Operation Command for Integers	*ADD	Integer/long integer addition
	*SUB	Integer/long integer subtraction
	*MUL	Integer/long integer multiplication
	*DIV	Integer/long integer division
	*SQT	Arithmetic square root of integer/long integer
	*INC	Integer/long integer increment by 1
	*DEC	Integer/long integer decrement by 1
	*VABS	Absolute value of integer/long integer
	*NEG	Integer/long integer negation
	*SUM	Integer/long integer accumulation
	*MEAN	Mean value of integers/long integers

3.8.2 ADD: Integer/Long Integer Addition Commands

Command table		ADD (S1) (S2) (D)			Applicable model	TS600 series		
16-Bit command		Add: Integer addition						
32-Bit command		Add: Long integer addition						
Operand	Type	Bit			Word		Indexing	Constant
		X, Y, M, LM, T, C, S	Dx.y	Custom bit variable	D, R, V, Z, T, C	Custom word variable		
S1	INT/DINT	-	-	-	√[1]	√	√	√
S2	INT/DINT	-	-	-	√[1]	√	√	√
D	INT/DINT	-	-	-	√[1]	√	√	-

Remark:

[1] For 32-bit commands, the T and Z elements are not supported.

Operand Description

S1: Source operand 1.

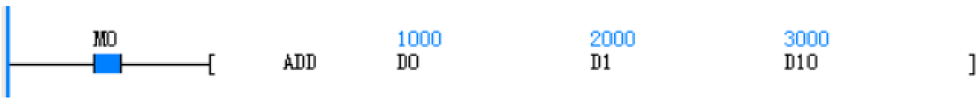
S2: Source operand 2.

D: The destination operand.

Function Description

1. When the energy flow is valid, S1 is added to S2, and the operation result is assigned to D.
2. When the operation result (D) is greater than 32767/2147483647, the carry flag bit (SM20) is set; when the operation result is equal to 0, the zero flag bit (SM18) is set; when the operation result is less than -32768/-2147483648, the borrow flag bit (SM19) is set.

Application Example



In case of M0=ON, D0 (1000) is added to D1 (2000), and the result is assigned to D10 to obtain D10=3000.

In case of M0=ON, the value (100000) of (D0, D1) is added to the value (200000) of (D2, D3), and the result is assigned to (D10, D11) to obtain (D10, D11)=300000.

3.8.3 SUB: Integer/Long Integer Subtraction Commands

Command table		SUB (S1) (S2) (D)			Applicable model	TS600 series		
16-Bit command		SUB: Integer subtraction						
32-Bit command		DSUB: Long integer subtraction						
Operand	Type	Bit			Word		Indexing	Constant
		X, Y, M, LM, T, C, S	Dx.y	Custom bit variable	D, R, V, Z, T, C	Custom word variable		
S1	INT/DINT	-	-	-	√[1]	√	√	√
S2	INT/DINT	-	-	-	√[1]	√	√	√
D	INT/DINT	-	-	-	√[1]	√	√	-

Remark:

[1] For 32-bit commands, the T and Z elements are not supported.

Operand Description

S1: Source operand 1

S2: Source operand 2

D: The destination operand

Function Description

1. When the energy flow is valid, S1 is added to S2, and the operation result is assigned to D.
2. When the operation result (D) is greater than 32767/2147483647, the carry flag bit (SM20) is set; when the operation result is equal to 0, the zero flag bit (SM18) is set; when the operation result is less than -32768/-2147483648, the borrow flag bit (SM19) is set.

Application Example

In case of M1=ON, D1 (2000) is subtracted from D0 (1000), and the result is assigned to D10 to obtain D10=-1000.

In case of M1=ON, the value (200000) of (D2, D3) is subtracted from the value (100000) of (D0, D1), and the result is assigned to (D10, D11) to obtain (D10, D11)=-100000.

3.8.4 MUL: Integer/Long Integer Multiplication Commands

Command table		MUL (S1) (S2) (D)			Applicable model	TS600 series		
16-Bit command		MUL: Integer/long integer multiplication						
32-Bit command		DMUL: Integer/long integer multiplication						
Operand	Type	Bit			Word		Indexing	Constant
		X, Y, M, LM, T, C, S	Dx.y	Custom bit variable	D, R, V, Z, T, C	Custom word variable		
S1	INT/DINT	-	-	-	√[1]	√	√	√
S2	INT/DINT	-	-	-	√[1]	√	√	√
D	DINT	-	-	-	√[1]	√	√	-

Remark:

[1] For 32-bit commands, the T and Z elements are not supported.

Operand Description

S1: Source operand 1.

S2: Source operand 2.

D: The destination operand.

Function Description

When the energy flow is valid, S1 is multiplied by S2, and the operation result is assigned to D.

Application Example

In case of M0=ON, D0 (1000) is multiplied by D1 (2000), and the result is assigned to (D10, D11) to obtain (D10, D11)=2000000.

In case of M0=ON, the value (83000) of (D0, D1) is multiplied by the value (2000) of (D2, D3), and the result is assigned to (D10, D11) to obtain (D10, D11)=1660000000.

3.8.5 DIV: Integer/Long Integer Division Commands

Command table		DIV (S1) (S2) (D)			Applicable model	TS600 series	
16-Bit command		DIV: Integer division					
32-Bit command		DDIV: Long integer devision					
Operand	Type	Bit			Word		Indexing	Constant
		X, Y, M, LM, T, C, S	Dx.y	Custom bit variable	D, R, V, Z, T, C	Custom word variable		
S1	INT/DINT	-	-	-	√[1]	√	√	√
S2	INT/DINT	-	-	-	√[1]	√	√	√
D	DINT, Array*2	-	-	-	√[1]	√	√	-

Remark:

[1] For 32-bit commands, the T and Z elements are not supported.

Operand Description

S1: Source operand 1.

S2: Source operand 2.

D: The destination operand.

Function Description

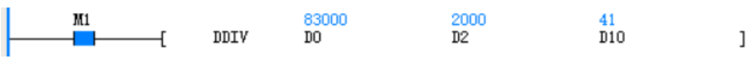
When the energy flow is valid, S1 is divided by S2, and the operation result is assigned to D. For the 16-bit command, D includes two units, where the first unit stores the quotient value and the second unit stores the remainder value; for the 32-bit command, D includes four units, where the first two units store the quotient value and the last two units store the remainder value.

Precautions

When the divisor is set to 0, the system reports an error about the divisor being 0.

Application Example

In case of M0=ON, D0 (2500) is divided by D1 (1000), and the result is assigned to (D10, D11). D10=2, and D11=500.



In case of M1=ON, the value (83000) of (D0, D1) is divided by the value (2000) of (D2, D3), and the result is assigned to (D10, D11) and (D12, D13). (D10, D11)=41, and (D12, D13)=1000。

3.8.6 SQT: Commands for Arithmetic Square Root of Integer/Long Integer

Command table		SQT (S1) (S2) (D)			Applicable model	TS600 series	
16-Bit command		SQT: Arithmetic square root of integer					
32-Bit command		DSQT: Arithmetic square root of long integer					
Operand	Type	Bit			Word		Indexing	Constant
		X, Y, M, LM, T, C, S	Dx.y	Custom bit variable	D, R, V, Z, T, C	Custom word variable		
S	INT/DINT	-	-	-	√[1]	√	√	√
D	INT/DINT	-	-	-	√[1]	√	√	-

Remark:

[1] For 32-bit commands, the T and Z elements are not supported.

Operand Description

S: The source operand.

D: The destination operand.

Function Description

1. When the energy flow is valid, the square root of S is extracted, and the operation result is assigned to D.
2. When the operation result is equal to 0, the zero flag bit (SM18) is set; when decimals are truncated from the operation result, the borrow flag bit (SM19) is set.

Application Example

In case of M0=ON, the square root of D0 (1000) is extracted, and the result is assigned to D10 to obtain D10=31.

In case of M1=ON, the square root of the value (83000) of (D0, D1) is extracted, and the result is assigned to (D10, D11) to obtain (D10, D11)=288.

3.8.7 INC: Commands for Integer/Long Integer Increment by 1

<table>
<tr><td colspan="2">Command table</td><td colspan="3">INC (D)</td><td>Applicable model</td><td colspan="3">TS600 series</td></tr>
<tr><td colspan="2">16-Bit command</td><td colspan="7">INC: Integer increment by 1</td></tr>
<tr><td colspan="2">32-Bit command</td><td colspan="7">DINC: Long integer increment by 1</td></tr>
<tr><td rowspan="2">Operand</td><td rowspan="2">Type</td><td colspan="3">Bit</td><td colspan="2">Word</td><td rowspan="2">Indexing</td><td rowspan="2">Constant</td></tr>
<tr><td>X, Y, M, LM, T, C, S</td><td>Dx.y</td><td>Custom bit variable</td><td>D, R, V, Z, T, C</td><td>Custom word variable</td></tr>
<tr><td>D</td><td>INT/DINT</td><td>-</td><td>-</td><td>-</td><td>√[1]</td><td>√</td><td>√</td><td>-</td></tr>
</table>

Remark:

[1] For 32-bit commands, the T and Z elements are not supported.

Operand Description

D: The destination operand.

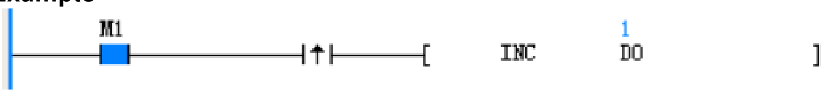
Function Description

When the energy flow is valid, D increases by 1.

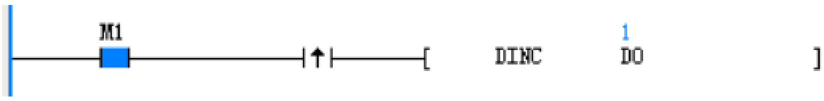
Precautions

This command is a cyclic addition command. For the 16-bit command, the value range is -32768~32767. For the 32-bit command, the value range is -2147483648~2147483647.

Application Example



In case of M1=ON, D0=0 increases by 1; after execution, D0=1 is obtained.



In case of M1=ON, (D0, D1)=0 increases by 1; after execution, (D0, D1)=1 is obtained.

3.8.8 DEC: Commands for Integer/Long Integer Decrement by 1

Command table		DEC (D)			Applicable model	TS600 series		
16-Bit command		DEC: Integer decrement by 1						
32-Bit command		DDEC: Long integer decrement by 1						
Operand	Type	Bit			Word		Indexing	Constant
		X, Y, M, LM, T, C, S	Dx.y	Custom bit variable	D, R, V, Z, T, C	Custom word variable		
D	INT/DINT	-	-	-	√[1]	√	√	-

Remark:

[1] For 32-bit commands, the T and Z elements are not supported.

Operand Description

D: The destination operand.

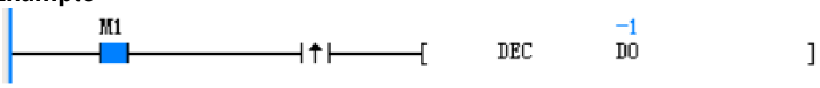
Function Description

When the energy flow is valid, D decreases by 1.

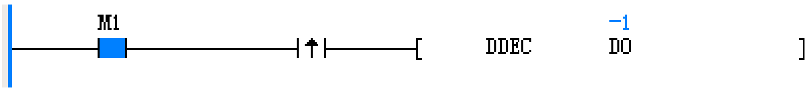
Precautions

This command is a cyclic subtraction command. For the 16-bit command, the value range is -32768~32767. For the 32-bit command, the value range is -2147483648~2147483647.

Application Example



In case of M1=ON, D0=0 decreases by 1; after execution, D0=-1 is obtained.



In case of M1=ON, (D0, D1)=0 increases by 1; after execution, (D0, D1)=-1 is obtained.

3.8.9 VABS: Commands for Absolute Value of Integer/Long Integer

Command table		VABS (S) (D)			Applicable model	TS600 series		
16-Bit command		VABS: Absolute value of integer						
32-Bit command		DVABS: Absolute value of long integer						
Operand	Type	Bit			Word		Indexing	Constant
		X, Y, M, LM, T, C, S	Dx.y	Custom bit variable	D, R, V, Z, T, C	Custom word variable		
S	INT/DINT	-	-	-	√[1]	√	√	√
D	INT/DINT	-	-	-	√[1]	√	√	-

Remark:

[1] For 32-bit commands, the T and Z elements are not supported.

Operand Description

S: The source operand.

D: The destination operand.

Function Description

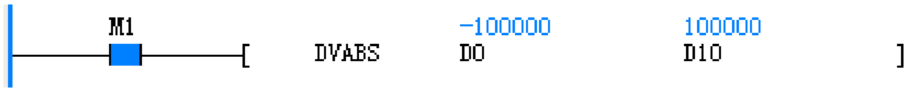
When the energy flow is valid, the absolute value of S is taken, and the operation result is assigned to D.

Precautions

This command is a cyclic subtraction command. For the 16-bit command, the value range is -32768~32767. For the 32-bit command, the value range is -2147483648~2147483647.

Application Example

In case of M0=ON, the absolute value of D0 (-1000) is taken, and the result is assigned to D10 to obtain D10=1000.



In case of M1=ON, the absolute value of the value (-100000) of (D0, D1) is taken, and the result is assigned to (D10, D11) to obtain (D10, D11)=100000.

3.8.10 NEG: Integer/Long Integer Negation Commands

Command table		NEG (S) (D)			Applicable model	TS600 series		
16-Bit command		NEG: Integer negation						
32-Bit command		DNEG: Long integer negation						
Operand	Type	Bit			Word		Indexing	Constant
		X, Y, M, LM, T, C, S	Dx.y	Custom bit variable	D, R, V, Z, T, C	Custom word variable		
S	INT/DINT	-	-	-	√[1]	√	√	√
D	INT/DINT	-	-	-	√[1]	√	√	-

Remark:

[1] For 32-bit commands, the T and Z elements are not supported.

Operand Description

S: The source operand.

D: The destination operand.

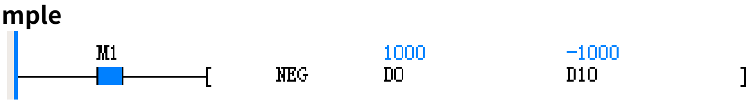
Function Description

When the energy flow is valid, S is negated, and the operation result is assigned to D.

Precautions

- In the 16-bit command, the range of S should be -32767~32767; when the value of S is -32768, an error about the illegal operand is reported, and the command does not generate any action.
- In the 32-bit command, the range of S should be -2147483647~2147483647; when the value of S is -2147483648, an error about the illegal operand is reported, and the command does not generate any action.

Application Example



In case of M1=ON, D0 (1000) is negated, and the result is assigned to D10 to obtain D10=-1000.

In case of M1=ON, (D0, D1) (100000) is negated, and the result is assigned to (D10, D11) to obtain (D10, D11)=-100000.

3.8.11 SUM: Integer/Long Integer Accumulation Commands

Command table		SUM (S1) (S2) (D)			Applicable model	TS600 series		
16-Bit command		SUM: Integer accumulation						
32-Bit command		DSUM: Long integer accumulation						
Operand	Type	Bit			Word		Indexing	Constant
		X, Y, M, LM, T, C, S	Dx.y	Custom bit variable	D, R, V, Z, T, C	Custom word variable		
S1	INT/DINT, Array*S2	-	-	-	√[1]	√	√	-
S2	WORD	-	-	-	√	√	√	√
D	DINT	-	-	-	√[2]	√	√	-

Remark:

[1]For 32-bit commands, the T and Z elements are not supported.

[2]The T and Z elements are not supported.

Operand Description

S1: The source operand, which indicates the starting unit of accumulation.

S2: The source operand, which indicates the number of accumulated data.

D: The destination operand, which indicates the accumulation result.

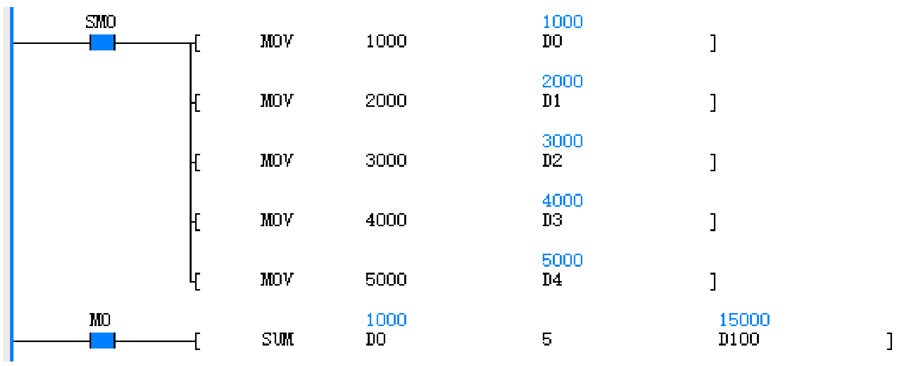
Function Description

When the energy flow is valid, the contents of the S2 units (or S2×2 units for the 32-bit command) starting from the starting unit S1 are accumulated, and the result after accumulation operation is assigned to the D unit.

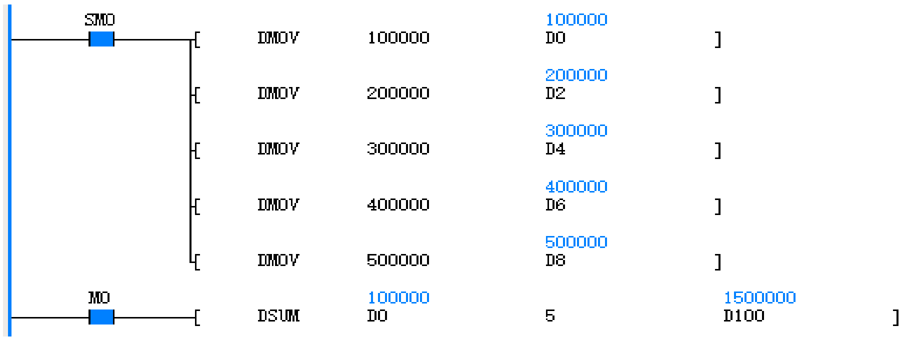
Precautions

- Ensure 0⩽S2⩽255, otherwise an operand error is reported.
- The carry flag bit SM20 and the borrow flag bit SM19 are constantly 0, since D is a 32-bit data. The zero flag bit is determined according to the result of final accumulation.

Application Example



In case of M0=ON, the data of the 5 units starting from D0 are accumulated, and the result is assigned to (D100, D101). (D100,D101) = D0 ＋ …… ＋ D4 = 15000.



In case of M0=ON, the long integers of the 5×2 units starting from D0 are accumulated, and the result is assigned to (D100, D101). (D100, D101) = (D0, D1) + ... + (D8, D9) = 1500000.

3.8.12 MEAN: Commands for Mean Value of Integers/Long Integers

Command table	MEAN (S1) (D) (S2)				Applicable model	TS600 series		
16-Bit command	MEAN: Mean value of integers							
32-Bit command	DMEAN: Mean value of long integers							
Operand	Type	Bit			Word		Indexing	Constant
		X, Y, M, LM, T, C, S	Dx.y	Custom bit variable	D, R, V, Z, T, C	Custom word variable		
S1	INT/DINT, Array*S2	-	-	-	√[1]	√	√	√
D	INT/DINT	-	-	-	√[2]	√	√	-
S2	WORD	-	-	-	√[3]	√	√	√

Remark:

[1]For the 16-bit command, the D, C, T, and R elements are supported; for the 32-bit command, the D, C, and R elements are supported.

[2]The D, C, and R elements are supported.

[3]The D and R elements are supported.

Operand Description

S1: The number of the starting word element, which saves the desired average value data.

D: The number of the word element, which saves the obtained average value data.

S2: The average number of data (1~64).

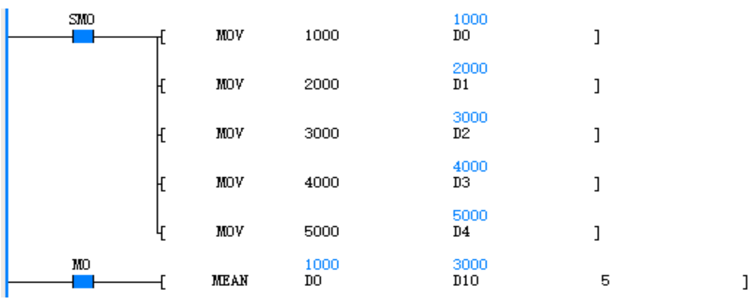
Function Description

The average value of the S2 16-bit data starting from S1 is saved to D, and the remainder is truncated.

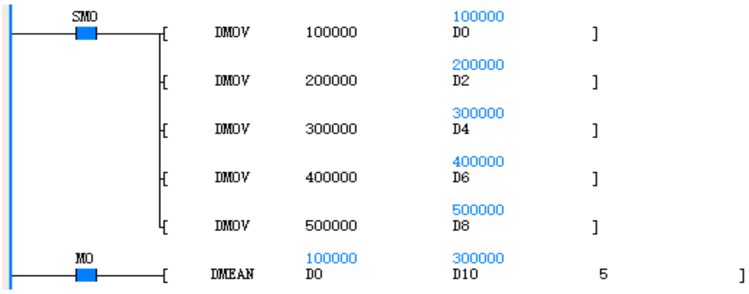
Precautions

This command resets the borrow flag bit (SM19) and the carry flag bit (SM20), while the zero flag bit (SM18) is determined according to the result of the final average.

Application Example



In case of M0=ON, the average value of the data in the 5 units starting from D0 is calculated, and the result is saved in D10 to obtain D10=3000.



In case of M0=ON, the average value of the data in the 5×2 units starting from D0 is calculated, and the result is assigned to (D10, D11) to obtain (D10, D11)=300000.

3.9 Arithmetic Operation Command for Floating-Point Numbers

3.9.1 Command Table

Command Category	Name	Function
Arithmetic Operation Command for Floating-Point Numbers	RADD	Floating-point number addition
	RSUB	Floating-point number subtraction
	RMUL	Floating-point number multiplication
	RDIV	Floating-point number division
	RSQT	Arithmetic square root of floating-point number
	RVABS	Absolute value of floating-point number
	RNEG	Floating-point number negation
	SIN	Sine operation of floating-point number
	COS	Cosine operation of floating-point number
	TAN	Tangent operation of floating-point number
	POWER	Power operation of floating-point number
	LN	Natural logarithm operation of floating-point number
	EXP	Natural number power operation of floating-point number
	RSUM	Accumulation operation of floating-point number
	RMEAN	Mean operation of floating-point numbers
	ASIN	Anti-sine operation of floating-point number
	ACOS	Anti-cosine operation of floating-point number
	ATAN	Anti-tangent operation of floating-point number
	SINH	Hyperbolic sine operation of floating-point number
	COSH	Hyperbolic cosine operation of floating-point number
	TANH	Hyperbolic tangent operation of floating-point number
	LOG	Common logarithm operation of floating-point number
	RAD	Angle-to-radian conversion of floating-point number
	DEG	Radian-to-angle conversion of floating-point number

3.9.2 RADD: Floating-Point Number Addition Commands

Command table	RADD (S1) (S2) (D)			Applicable model	TS600 series		
16-Bit command	-						
32-Bit command	RADD: Floating-point number addition						

Operand	Type	Bit			Word		Indexing	Constant
		X, Y, M, LM, T, C, S	Dx.y	Custom bit variable	D, R, V, Z, T, C	Custom word variable		
S1	REAL	-	-	-	√[1]	√	√	√
S2	REAL	-	-	-	√[1]	√	√	√
D	REAL	-	-	-	√[1]	√	√	-

Remark:

[1] The D, V, and R elements are supported.

Operand Description

S1: Source operand 1.

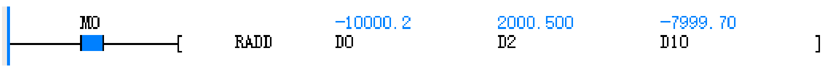
S2: Source operand 2.

D: The destination operand.

Function Description

1. When the energy flow is valid, S1 is added to S2, and the operation result is assigned to D.
2. When the operation result (D) is greater than 1.701412e+038 or less than -1.701412e+038, the carry flag bit (SM20) is set. When the operation result is equal to 0, the zero flag bit (SM18) is set.

Application Example



In case of M0=ON, the value (-10000.2) of (D0, D1) is added to the value (2000.5) of (D2, D3), and the result is assigned to (D10, D11) to obtain (D10, D11)=-7999.7.

3.9.3 RSUB: Floating-Point Number Subtraction Commands

Command table	RSUB (S1) (S2) (D)			Applicable model	TS600 series		
16-Bit command	-						
32-Bit command	RSUB: Floating-point number subtraction						

Operand	Type	Bit			Word		Indexing	Constant
		X, Y, M, LM, T, C, S	Dx.y	Custom bit variable	D, R, V, Z, T, C	Custom word variable		
S1	REAL	-	-	-	√[1]	√	√	√
S2	REAL	-	-	-	√[1]	√	√	√
D	REAL	-	-	-	√[1]	√	√	-

Remark:

[1] The D, V, and R elements are supported.

Operand Description

S1: Source operand 1

S2: Source operand 2

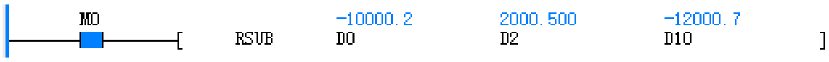
D: The destination operand

Function Description

1. When the energy flow is valid, S2 is subtracted from S1, and the operation result is assigned to D.

2. When the operation result (D) is greater than 1.701412e+038 or less than -1.701412e+038, the carry flag bit (SM20) is set. When the operation result is equal to 0, the zero flag bit (SM18) is set.

Application Example



In case of M0=ON, the value (2000.5) of (D2, D3) is subtracted from the value (-10000.2) of (D0, D1), and the result is assigned to (D10, D11) to obtain (D10, D11)=-12000.7.

3.9.4 RMUL: Floating-Point Number Multiplication Commands

Command table	RMUL (S1) (S2) (D)				Applicable model	TS600 series		
16-Bit command	-							
32-Bit command	RMUL: Floating-point number multiplication							
Operand	Type	Bit			Word		Indexing	Constant
		X, Y, M, LM, T, C, S	Dx.y	Custom bit variable	D, R, V, Z, T, C	Custom word variable		
S1	REAL	-	-	-	√[1]	√	√	√
S2	REAL	-	-	-	√[1]	√	√	√
D	REAL	-	-	-	√[1]	√	√	-

Remark:

[1] The D, V, and R elements are supported.

Operand Description

S1: Source operand 1.

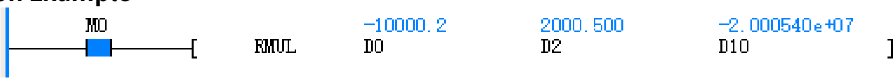
S2: Source operand 2.

D: The destination operand.

Function Description

1. When the energy flow is valid, S1 is multiplied by S2, and the operation result is assigned to D.
2. When the operation result (D) is greater than 1.701412e+038 or less than -1.701412e+038, the carry flag bit (SM20) is set. When the operation result is equal to 0, the zero flag bit (SM18) is set.

Application Example



In case of M0=ON, the value (-10000.2) of (D0, D1) is multiplied by the value (2000.5) of (D2, D3), and the result is assigned to (D10, D11) to obtain (D10,D11)=-20005400.0 (actually, the product should be -20005400.1, but 0.1 has been truncated for the sake of measurement accuracy).

3.9.5 RDIV: Floating-Point Number Division Commands

Command table	RDIV (S1) (S2) (D)				Applicable model	TS600 series		
16-Bit command	-							
32-Bit command	RDIV: Floating-point number division							
Operand	Type	Bit			Word		Indexing	Constant
		X, Y, M, LM, T, C, S	Dx.y	Custom bit variable	D, R, V, Z, T, C	Custom word variable		
S1	REAL	-	-	-	√[1]	√	√	√
S2	REAL	-	-	-	√[1]	√	√	√
D	REAL	-	-	-	√[1]	√	√	-

Remark:

[1] The D, V, and R elements are supported.

Operand Description

S1: Source operand 1.

S2: Source operand 2.

D: The destination operand.

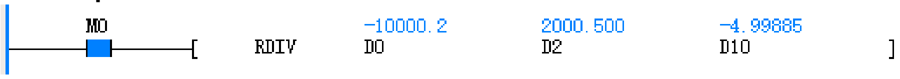
Function Description

When the energy flow is valid, S1 is divided by S2, and the operation result is assigned to D. When the operation result (D) is greater than 1.701412e+038 or less than -1.701412e+038, the carry flag bit (SM20) is set. When the operation result is equal to 0, the zero flag bit (SM18) is set.

Precautions

Ensure S2≠0, otherwise an error about the divisor being 0 is reported and the division operation is not executed.

Application Example



In case of M0=ON, the value (-10000.2) of (D0, D1) is divided by the value (2000.5) of (D2, D3), and the result is assigned to and (D10, D11). (D10,D11)=-4.998850.

3.9.6 RSQT: Commands for Square Root of Floating-Point Number

Command table	RSQT (S1) (S2) (D)			Applicable model	TS600 series		
16-Bit command	-						
32-Bit command	RSQT: Square root of floating-point number						
Operand	Type	Bit			Word		Indexing	Constant
		X, Y, M, LM, T, C, S	Dx.y	Custom bit variable	D, R, V, Z, T, C	Custom word variable		
S	REAL	-	-	-	√[1]	√[1]	√	√
D	REAL	-	-	-	√[1]	√[1]	√	-

Remark:

[1] The D, V, and R elements are supported.

Operand Description

S: The source operand.

D: The destination operand.

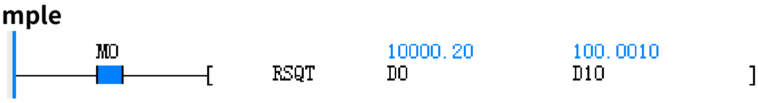
Function Description

1. When the energy flow is valid, the square root of S is extracted, and the operation result is assigned to D.
2. When the operation result (D) is equal to 0, the zero flag bit SM18 is set.

Precautions

Ensure S≥0, otherwise an operand error is reported and the square root operation is not executed.

Application Example



In case of M0=ON, the square root of the value (10000.2) of (D0, D1) is extracted, and the result is assigned to (D10, D11) to obtain (D10, D11)=100.0010.

3.9.7 RVABS: Commands for Absolute Value of Floating-Point Number

Command table	RVABS (S) (D)			Applicable model	TS600 series			
16-Bit command	-							
32-Bit command	RVABS: Absolute value of floating-point number							
Operand	Type	Bit			Word		Indexing	Constant
		X, Y, M, LM, T, C, S	Dx.y	Custom bit variable	D, R, V, Z, T, C	Custom word variable		
S	REAL	-	-	-	√[1]	√	√	√
D	REAL	-	-	-	√[1]	√	√	-

Remark:

[1] The D, V, and R elements are supported.

Operand Description

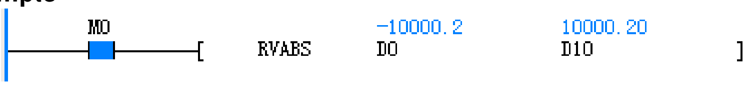
S: The source operand.

D: The destination operand.

Function Description

When the energy flow is valid, the absolute value of S is taken, and the operation result is assigned to D.

Application Example



In case of M0=ON, the absolute value of the value (-10000.2) of (D0, D1) is taken, and the result is assigned to (D10, D11) to obtain (D10, D11)=10000.2.

3.9.8 RNEG: Floating-Point Number Negation Commands

Command table	RNEG (S) (D)			Applicable model	TS600 series			
16-Bit command	-							
32-Bit command	RNEG: Floating-point number negation							
Operand	Type	Bit			Word		Indexing	Constant
		X, Y, M, LM, T, C, S	Dx.y	Custom bit variable	D, R, V, Z, T, C	Custom word variable		
S	REAL	-	-	-	√[1]	√	√	√
D	REAL	-	-	-	√[1]	√	√	-

Remark:

[1] The D, V, and R elements are supported.

Operand Description

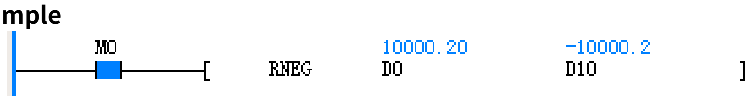
S: The source operand.

D: The destination operand.

Function Description

When the energy flow is valid, S is negated, and the operation result is assigned to D.

Application Example



In case of M0=ON, (D0, D1) (10000.2) is negated, and the result is assigned to (D10, D11) to obtain (D10, D11)=-10000.2.

3.9.9 SIN: Commands for Sine Operation of Floating-Point Number

Command table	SIN (S) (D)			Applicable model	TS600 series		
16-Bit command	-						
32-Bit command	SIN: Sine operation of floating-point number						
Operand	Type	Bit			Word		Indexing	Constant
		X, Y, M, LM, T, C, S	Dx.y	Custom bit variable	D, R, V, Z, T, C	Custom word variable		
S	REAL	-	-	-	√[1]	√	√	√
D	REAL	-	-	-	√[1]	√	√	-

Remark:

[1] The D, V, and R elements are supported.

Operand Description

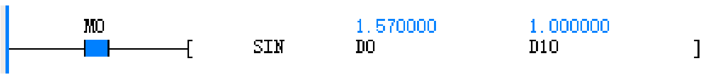
S: The source operand.

D: The destination operand.

Function Description

When the energy flow is valid, S is negated, and the operation result is assigned to D.

Application Example



In case of M0=ON, the SIN value of (D0, D1)=1.57 is calculated, and the result is assigned to (D10, D11) to obtain (D10, D11)=1.

3.9.10 COS: Commands for Cosine Operation of Floating-Point Number

Command table	COS (S) (D)			Applicable model	TS600 series		
16-Bit command	-						
32-Bit command	COS: Cosine operation of floating-point number						
Operand	Type	Bit			Word		Indexing	Constant
		X, Y, M, LM, T, C, S	Dx.y	Custom bit variable	D, R, V, Z, T, C	Custom word variable		
S	REAL	-	-	-	√[1]	√	√	√
D	REAL	-	-	-	√[1]	√	√	-

Remark:

[1] The D, V, and R elements are supported.

Operand Description

S: The source operand.

D: The destination operand.

Function Description

1. When the energy flow is valid, the COS value of S (taking a radian as the unit) is calculated, and the operation result is assigned to D.
2. When the operation result (D) is equal to 0, the zero flag bit (SM18) is set.

Application Example

In case of M0=ON, the COS value of (D0, D1)=3.14 is calculated, and the result is assigned to (D10, D11) to obtain (D10, D11)=-1.

3.9.11 RSUM: Commands for Accumulation Operation of Floating-Point Number

Command table	RSUM (S) (D)			Applicable model	TS600 series			
16-Bit command	-							
32-Bit command	RSUM: Accumulation operation of floating-point number							
Operand	Type	Bit			Word		Indexing	Constant
		X, Y, M, LM, T, C, S	Dx.y	Custom bit variable	D, R, V, Z, T, C	Custom word variable		
S1	REAL	-	-	-	√[1]	√	√	-
S2	WORD	-	-	-	√[1]	√	√	√
D	REAL	-	-	-	√[1]	√	√	-

Remark:

[1] The D, V, and R elements are supported.

Operand Description

S: The source operand.

D: The destination operand.

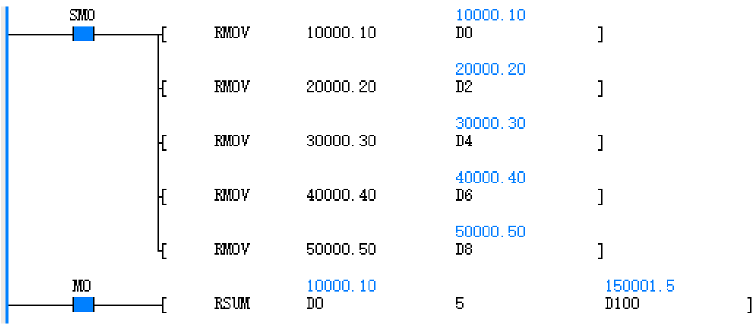
Function Description

When the energy flow is valid, the contents of the S2×2 units starting from the starting unit (S1) are accumulated as per the floating-point data, and the result after operation is assigned to the D unit.

Precautions

- Ensure 0⩽S2⩽255, otherwise an operand error is reported.
- If an overflow occurs, the accumulation operation is not executed any more.

Application Example



In case of M0=ON, the floating-point numbers of the 5×2 units starting from D0 are accumulated, and the result is assigned to (D100, D101). (D100, D101) = (D0, D1) + ... + (D8, D9) = 150001.5.

3.9.12 TAN: Commands for Tangent Operation of Floating-Point Number

Command table	TAN (S) (D)			Applicable model	TS600 series			
16-Bit command	-							
32-Bit command	TAN: Tangent operation of floating-point number							
Operand	Type	Bit			Word		Indexing	Constant
		X, Y, M, LM, T, C, S	Dx.y	Custom bit variable	D, R, V, Z, T, C	Custom word variable		
S	REAL	-	-	-	√[1]	√	√	√
D	REAL	-	-	-	√[1]	√	√	-

Remark:

[1] The D, V, and R elements are supported.

Operand Description

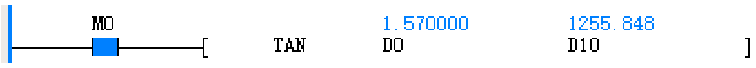
S: The source operand.

D: The destination operand.

Function Description

1. When the energy flow is valid, the TAN value of S (taking a radian as the unit) is calculated, and the operation result is assigned to D.
2. When the operation result (D) is greater than 1.701412e+038 or less than -1.701412e+038, the carry flag bit (SM20) is set. When the operation result is equal to 0, the zero flag bit (SM18) is set.

Application Example



In case of M0=ON, the TAN value of (D0, D1)=1.57 is calculated, and the result is assigned to (D10, D11) to obtain (D10, D11)=1255.848398.

3.9.13 POWER: Commands for Power Operation of Floating-Point Number

Command table		POWER (S) (D)			Applicable model	TS600 series		
16-Bit command		-						
32-Bit command		POWER: Power operation of floating-point number						
Operand	Type	Bit			Word		Indexing	Constant
		X, Y, M, LM, T, C, S	Dx.y	Custom bit variable	D, R, V, Z, T, C	Custom word variable		
S1	REAL	-	-	-	√[1]	√	√	√
S2	REAL	-	-	-	√[1]	√	√	√
D	REAL	-	-	-	√[1]	√	√	-

Remark:

[1] The D, V, and R elements are supported.

Operand Description

S: The source operand.

D: The destination operand.

Function Description

1. When the energy flow is valid, the S2-nd power of S1 is calculated, and the operation result is assigned to D.
2. When the operation result (D) is greater than 1.701412e+038 or less than -1.701412e+038, the carry flag bit (SM20) is set. When the operation result is equal to 0, the zero flag bit (SM18) is set.

Precautions

- In case of S1=0 and S2≤0, an operand error is reported and the operation is not executed.
- In case of S1<0 and the mantissa of S2 not being 0, an operand error is reported and the operation is not executed.

Application Example

In case of M0=ON, the (D2, D3)-th power of (D0, D1) (namely the 3rd power of 55.0) is calculated, and the result is assigned to (D10, D11) to obtain (D10, D11)=166375.0.

3.9.14 LN: Commands for Natural Logarithm Operation of Floating-Point Number

Command table		LN (S) (D)			Applicable model	TS600 series		
16-Bit command		-						
32-Bit command		LN: Natural logarithm operation of floating-point number						
Operand	Type	Bit			Word		Indexing	Constant
		X, Y, M, LM, T, C, S	Dx.y	Custom bit variable	D, R, V, Z, T, C	Custom word variable		
S	REAL	-	-	-	√[1]	√	√	√
D	REAL	-	-	-	√[1]	√	√	-

Remark:

[1] The D, V, and R elements are supported.

Operand Description

S: The source operand.

D: The destination operand.

Function Description

1. When the energy flow is valid, the LN value of S is calculated, and the operation result is assigned to D.
2. When the operation result (D) is greater than 1.701412e+038 or less than -1.701412e+038, the carry flag bit (SM20) is set. When the operation result is equal to 0, the zero flag bit (SM18) is set.

Application Example

In case of M0=ON, the LN value of (D0, D1)=1000.0 is calculated, and the result is assigned to (D10, D11) to obtain (D10, D11)=6.907755.

3.9.15 EXP: Commands for Natural Number Power Operation of Floating-Point Number

Command table		EXP (S) (D)			Applicable model	TS600 series		
16-Bit command		-						
32-Bit command		EXP: Natural number power operation of floating-point number						
Operand	Type	Bit			Word		Indexing	Constant
		X, Y, M, LM, T, C, S	Dx.y	Custom bit variable	D, R, V, Z, T, C	Custom word variable		
S	REAL	-	-	-	√[1]	√	√	√
D	REAL	-	-	-	√[1]	√	√	-

Remark:

[1] The D, V, and R elements are supported.

Operand Description

S: The source operand.

D: The destination operand.

Function Description

1. When the energy flow is valid, the EXP value of S is calculated, and the operation result is assigned to D.
2. When the operation result (D) is greater than 1.701412e+038 or less than -1.701412e+038, the carry flag bit (SM20) is set. When the operation result is equal to 0, the zero flag bit (SM18) is set.

Application Example

In case of M0=ON, the EXP value of (D0, D1)=10.0 is calculated, and the result is assigned to (D10, D11) to obtain (D10, D11)=22026.464844.

3.9.16 RMEAN: Commands for Mean Operation of Floating-Point Number

Command table		RMEAN (S1) (D) (S2)			Applicable model	TS600 series		
16-Bit command		-						
32-Bit command		RMEAN: Mean value of long integers						
Operand	Type	Bit			Word		Indexing	Constant
		X, Y, M, LM, T, C, S	Dx.y	Custom bit variable	D, R, V, Z, T, C	Custom word variable		
S1	REAL	-	-	-	√[1]	√	√	-
D	REAL	-	-	-	√[1]	√	√	-
S2	WORD	-	-	-	√[2]	√	-	√

Remark:

[1]The D, V, and R elements are supported.

[2]The D and R elements are supported.

Operand Description

S1: The number of the starting word element, which saves the desired average value data.

D: The number of the word element, which saves the obtained average value data.

S2: The average number of data (1~64).

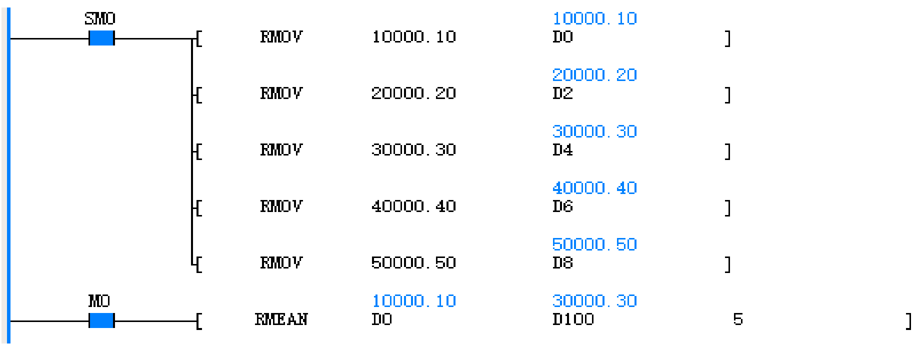
Function Description

The average value of the S2 16-bit data starting from S1 is saved to D, and the remainder is truncated.

Precautions

- Ensure 0⩽S2⩽255, otherwise an operand error is reported.
- The carry flag bit and the borrow flag bit are constantly 0, since D is a 32-bit data. The zero flag bit is determined according to the result of final accumulation.

Application Example



In case of M0=ON, the average value of the data in the 5×2 units starting from D0 is calculated, and the result is assigned to (D100,D101) to obtain (D100, D101)=30000.3.

3.9.17 ASIN: Commands for Anti-Sine Operation of Floating-Point Number

Command table	ASIN (S) (D)				Applicable model	TS600 series		
16-Bit command	-							
32-Bit command	ASIN: Anti-sine operation of floating-point number							
Operand	Type	Bit			Word		Indexing	Constant
		X, Y, M, LM, T, C, S	Dx.y	Custom bit variable	D, R, V, Z, T, C	Custom word variable		
S	REAL	-	-	-	√[1]	√	√	√
D	REAL	-	-	-	√[1]	√	√	-

Remark:

[1] The D, V, and R elements are supported.

Operand Description

S: The source operand.

D: The destination operand.

Function Description

1. When the energy flow is valid, the SIN-1 value of S is calculated, and the operation result is assigned to D.
2. When the operation result (D) is equal to 0, the zero flag bit (SM18) is set.

Precautions

In case of S＞1 or S＜-1, the system reports an operand error, the conversion is not executed, and the content of D remains unchanged.

Application Example

In case of SM0=ON, the SIN-1 value of (D0, D1)=0.500000 is calculated, and the result is assigned to (D10, D11) to obtain (D10, D11)=0.523599.

3.9.18 ACOS: Commands for Anti-Cosine Operation of Floating-Point Number

Command table	ACOS (S) (D)				Applicable model	TS600 series		
16-Bit command	-							
32-Bit command	ACOS: Anti-cosine operation of floating-point number							
Operand	Type	Bit			Word		Indexing	Constant
		X, Y, M, LM, T, C, S	Dx.y	Custom bit variable	D, R, V, Z, T, C	Custom word variable		
S	REAL	-	-	-	√[1]	√	√	√
D	REAL	-	-	-	√[1]	√	√	-

Remark:

[1] The D, V, and R elements are supported.

Operand Description

S: The source operand.

D: The destination operand.

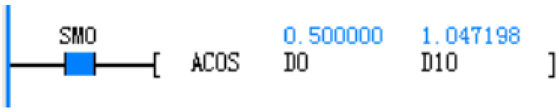
Function Description

1. When the energy flow is valid, the COS-1 value of S is calculated, and the operation result is assigned to D.
2. When the operation result (D) is equal to 0, the zero flag bit (SM180) is set.

Precautions

In case of S＞1 or S＜-1, the system reports an operand error, the conversion is not executed, and the content of D remains unchanged.

Application Example



In case of SM0=ON, the COS-1 value of (D0, D1)=0.500000 is calculated, and the result is assigned to (D10, D11) to obtain (D10, D11)=1.047198.

3.9.19 ATAN: Commands for Anti-Tangent Operation of Floating-Point Number

Command table	ATAN (S) (D)				Applicable model	TS600 series		
16-Bit command	-							
32-Bit command	ATAN: Anti-tangent operation of floating-point number							
Operand	Type	Bit			Word		Indexing	Constant
		X, Y, M, LM, T, C, S	Dx.y	Custom bit variable	D, R, V, Z, T, C	Custom word variable		
S	REAL	-	-	-	√[1]	√	√	√
D	REAL	-	-	-	√[1]	√	√	-

Remark:

[1] The D, V, and R elements are supported.

Operand Description

S: The source operand.

D: The destination operand.

Function Description

1. When the energy flow is valid, the TAN-1 value of S is calculated, and the operation result is assigned to D.
2. When the operation result (D) is equal to 0, the zero flag bit (SM180) is set.

Application Example

In case of SM0=ON, the TAN-1 value of (D0, D1)=3.14 is calculated, and the result is assigned to (D10, D11) to obtain (D10, D11)=1.262481.

3.9.20 SINH: Commands for Hyperbolic Sine Operation of Floating-Point Number

Command table	SINH (S) (D)	Applicable model	TS600 series
16-Bit command	-		
32-Bit command	SINH: Anti-sine operation of floating-point number		

Operand	Type	Bit			Word		Indexing	Constant
		X, Y, M, LM, T, C, S	Dx.y	Custom bit variable	D, R, V, Z, T, C	Custom word variable		
S	REAL	-	-	-	√[1]	√	√	√
D	REAL	-	-	-	√[1]	√	√	-

Remark:

[1] Only the D, V, and R elements are supported.

Operand Description

S: The source operand.

D: The destination operand.

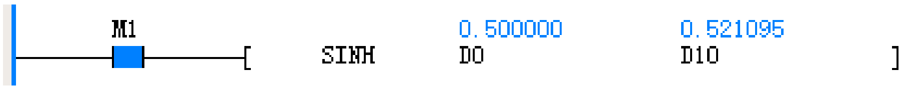
Function Description

1. When the energy flow is valid, the SINH value of S is calculated, and the operation result is assigned to D.
2. When the operation result (D) is equal to 0, the zero flag bit (SM18) is set.

Precautions

In case of S>1 or S<-1, the system reports an operand error, the conversion is not executed, and the content of D remains unchanged.

Application Example



In case of M1=ON, the SINH value of (D0, D1)=0.500000 is calculated, and the result is assigned to (D10, D11) to obtain (D10, D11)=0.521095.

3.9.21 COSH: Commands for Hyperbolic Cosine Operation of Floating-Point Number

Command table	COSH (S) (D)				Applicable model	TS600 series		
16-Bit command	-							
32-Bit command	COSH: Hyperbolic cosine operation of floating-point number							
Operand	Type	Bit			Word		Indexing	Constant
		X, Y, M, LM, T, C, S	Dx.y	Custom bit variable	D, R, V, Z, T, C	Custom word variable		
S	REAL	-	-	-	√[1]	√	√	√
D	REAL	-	-	-	√[1]	√	√	-

Remark:

[1] Only the D, V, and R elements are supported.

Operand Description

S: The source operand.

D: The destination operand.

Function Description

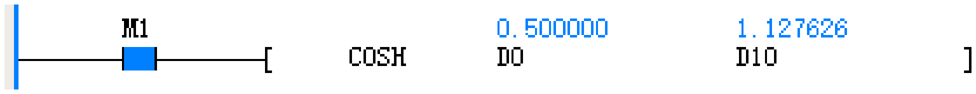
1. When the energy flow is valid, the COSH value of S is calculated, and the operation result is assigned to D.

2. When the operation result (D) is equal to 0, the zero flag bit (SM180) is set.

Precautions

In case of S>1 or S<-1, the system reports an operand error, the conversion is not executed, and the content of D remains unchanged.

Application Example



In case of M1=ON, the COSH value of (D0, D1)=0.500000 is calculated, and the result is assigned to (D10, D11) to obtain (D10, D11)=1.127626.

3.9.22 TANH: Commands for Hyperbolic Tangent Operation of Floating-Point Number

Command table	TANH (S) (D)			Applicable model	TS600 series		
16-Bit command	-						
32-Bit command	TANH: Anti-tangent operation of floating-point number						
Operand	Type	Bit			Word		Indexing	Constant
		X, Y, M, LM, T, C, S	Dx.y	Custom bit variable	D, R, V, Z, T, C	Custom word variable		
S	REAL	-	-	-	√[1]	√	√	√
D	REAL	-	-	-	√[1]	√	√	-

Remark:

[1] Only the D, V, and R elements are supported.

Operand Description

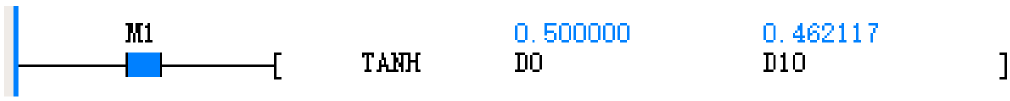
S: The source operand.

D: The destination operand.

Function Description

1. When the energy flow is valid, the TANH value of S is calculated, and the operation result is assigned to D.
2. When the operation result (D) is equal to 0, the zero flag bit (SM180) is set.

Application Example



In case of M1=ON, the TANH value of (D0, D1) is calculated, and the result is assigned to (D10, D11) to obtain (D10, D11)=0.462117.

3.9.23 LOG: Commands for Common Logarithm Operation of Floating-Point Number

Command table	LOG (S) (D)	Applicable model	TS600 series
16-Bit command	-		
32-Bit command	LOG: Common logarithm operation of floating-point number		

Operand	Type	Bit			Word		Indexing	Constant
		X, Y, M, LM, T, C, S	Dx.y	Custom bit variable	D, R, V, Z, T, C	Custom word variable		
S	REAL	-	-	-	√[1]	√	√	√
D	REAL	-	-	-	√[1]	√	√	-

Remark:

[1] Only the D, V, and R elements are supported.

Operand Description

S: The source operand.

D: The destination operand.

Function Description

1. When the energy flow is valid, the LOG value of S is calculated, and the operation result is assigned to D. LOG is a common logarithm operation with 10 as the base.
2. When the operation result (D) overflows, the carry (overflow) flag bit (SM20) is set; when the operation result is equal to 0, the zero flag bit (SM18) is set.

Application Example

In case of SM0=ON, the result is assigned to D10 (D11) to obtain D10 (D11)=0.477121.

3.9.24 RAD: Commands for Angle-to-Radian Conversion of Floating-Point Number

Command table	RAD (S) (D)				Applicable model	TS600 series		
16-Bit command	-							
32-Bit command	RAD: Angle-to-radian conversion of floating-point number							
Operand	Type	Bit			Word		Indexing	Constant
		X, Y, M, LM, T, C, S	Dx.y	Custom bit variable	D, R, V, Z, T, C	Custom word variable		
S	REAL	-	-	-	√[1]	√	√	√
D	REAL	-	-	-	√[1]	√	√	-

Remark:

[1] Only the D, V, and R elements are supported.

Operand Description

S: The source operand.

D: The destination operand.

Function Description

1. When the energy flow is valid, the angle value of the floating-point number in the S unit is converted into a radian value, and the result is assigned to D.
2. When the operation result is equal to 0, the zero flag bit (SM18) is set.

Application Example

In case of SM0=ON, the value (180.0) of D0 (D1) is converted, and the result is assigned to D10 (D11) to obtain D10 (D11)=3.141593.

3.9.25 DEG: Commands for Radian-to-Angle Conversion of Floating-Point Number

Command table	DEG (S) (D)				Applicable model	TS600 series		
16-Bit command	-							
32-Bit command	DEG: Radian-to-angle conversion of floating-point number							
Operand	Type	Bit			Word		Indexing	Constant
		X, Y, M, LM, T, C, S	Dx.y	Custom bit variable	D, R, V, Z, T, C	Custom word variable		
S	REAL	-	-	-	√[1]	√	√	√
D	REAL	-	-	-	√[1]	√	√	-

Remark:

[1] Only the D, V, and R elements are supported.

Operand Description

S: The source operand.

D: The destination operand.

Function Description

1. When the energy flow is valid, the radian value of the floating-point number in the S unit is converted into an angle value, and the result is assigned to D.
2. When the operation result is equal to 0, the zero flag bit (SM18) is set; when the operation result overflows, the carry (overflow) flag bit (SM20) is set.

Application Example

In case of SM0=ON, the value (3.0) of D0 (D1) is converted, and the result is assigned to D10 (D11) to obtain D10 (D11)=171.8873.

3.10 Word Logic Operation Command

3.10.1 Command Table

Command Category	Name	Function
Word Logic Operation Command	*WAND	Word/doubleword AND operation
	*WOR	Word/doubleword OR operation
	*WXOR	Word/doubleword XOR operation
	*WINV	Word/doubleword negation operation

3.10.2 WAND: Commands for Logical AND Operation of Word/Doubleword Data

Command table	*WAND (S1) (S2) (D)	Applicable model	TS600 series
16-Bit command	WAND: Logical AND operation of word data		
32-Bit command	DWAND: Logical AND operation of doubleword data		

Operand	Type	Bit			Word		Indexing	Constant
		X, Y, M, LM, T, C, S	Dx.y	Custom bit variable	D, R, V, Z, T, C	Custom word variable		
S1	WORD/ DWORD	-	-	-	√[1]	√	√	√
S2	WORD/ DWORD	-	-	-	√[1]	√	√	√
D	WORD/ DWORD	-	-	-	√[1]	√	√	

Remark:

[1] For 32-bit commands, the T and Z elements are not supported.

Operand Description

S1: The source operand, which indicates the address of the data or data storage word soft element that participates in the AND operation.

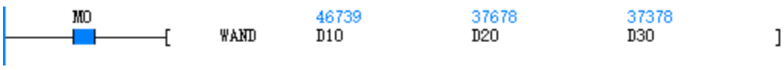
S2: The source operand, which indicates the address of the data or data storage word soft element that participates in the AND operation.

D: The destination operand, which indicates the address of the data storage word soft element of the operation result.

Function Description

1. When the command is driven, the bitwise logical AND operation is executed on S1 and S2, and the operation result is assigned to D.
2. The rule of logical AND operation: If any data is 0, the result is 0. For example: 1 • 1=1 1 • 0=0 0 • 1=0 0 • 0=0.

Application Example



		Element Name	Data Type	Display Format	Current Value
1	...	D10	WORD	Binary	2#1011011010010011
2	...	D20	WORD	Binary	2#1001001100101110
3	...	D30	WORD	Binary	2#1001001000000010

In case of M0=ON, the bitwise logical AND operation is executed on D0=2#1011011010010011 (46739) and D1=2#1001001100101110 (37678), and the result is assigned to D10 to obtain D10=2#1001001000000010 (37378).

3.10.3 WOR: Commands for Logical OR Operation of Word/Doubleword Data

Command table	*WOR (S1) (S2) (D)	Applicable model	TS600 series
16-Bit command	WOR: Logical OR operation of word data		
32-Bit command	DWOR: Logical OR operation of doubleword data		

Operand	Type	Bit			Word		Indexing	Constant
		X, Y, M, LM, T, C, S	Dx.y	Custom bit variable	D, R, V, Z, T, C	Custom word variable		
S1	WORD/ DWORD	-	-	-	√[1]	√	√	√
S2	WORD/ DWORD	-	-	-	√[1]	√	√	√
D	WORD/ DWORD	-	-	-	√[1]	√	√	

Remark:

[1] For 32-bit commands, the T and Z elements are not supported.

Operand Description

S1: The source operand, which indicates the address of the data or data storage word soft element that participates in the OR operation.

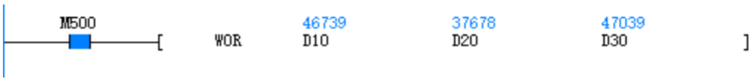
S2: The source operand, which indicates the address of the data or data storage word soft element that participates in the OR operation.

D: The destination operand, which indicates the address of the data storage word soft element of the operation result.

Function Description

1. When the command is driven, the bitwise logical OR operation is executed on S1 and S2, and the operation result is assigned to D.
2. The rule of logical OR operation: If any data is 1, the result is 1. For example: 1+1=1 1+0=1 0+1=1 0+0=0.

Application Example



		Element Name	Data Type	Display Format	Current Value
1	...	D10	WORD	Binary	2#1011011010010011
2	...	D20	WORD	Binary	2#1001001100101110
3	...	D30	WORD	Binary	2#1011011110111111

In case of M500=ON, the bitwise logical OR operation is executed on D10=2#1011011010010011 (46739) and D20=2#1001001100101110 (37678), and the result is assigned to D30 to obtain D30=2#1011011110111111 (47039).

3.10.4 WXOR: Commands for Logical XOR Operation of Word/Doubleword Data

Command table		*WXOR (S1) (S2) (D)			Applicable model	TS600 series		
16-Bit command		WXOR: Logical XOR operation of word data						
32-Bit command		DXWOR: Logical XOR operation of doubleword data						
Operand	Type	Bit			Word		Indexing	Constant
		X, Y, M, LM, T, C, S	Dx.y	Custom bit variable	D, R, V, Z, T, C	Custom word variable		
S1	WORD/ DWORD	-	-	-	√[1]	√	√	√
S2	WORD/ DWORD	-	-	-	√[1]	√	√	√
D	WORD/ DWORD	-	-	-	√[1]	√	√	-

Remark:

[1] For 32-bit commands, the T and Z elements are not supported.

Operand Description

S1: The source operand, which indicates the address of the data or data storage word soft element that participates in the XOR operation.

S2: The source operand, which indicates the address of the data or data storage word soft element that participates in the XOR operation.

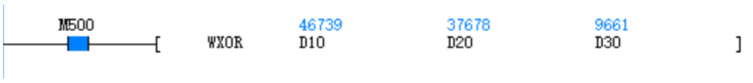
D: The destination operand, which indicates the address of the data storage word soft element of the operation result.

Function Description

1. When the command is driven, the bitwise logical XOR operation is executed on S1 and S2, and the operation result is assigned to D.
2. The rules of logical XOR operation: If two data are the same, the result is 0; if two data are different, the result is 1.

   For example: 1^1=0   1^0=1   0^1=1   0^0=0.

Application Example



		Element Name	Data Type	Display Format	Current Value
1	...	D10	WORD	Binary	2#1011011010010011
2	...	D20	WORD	Binary	2#1001001100101110
3	...	D30	WORD	Binary	2#10010110111101

In case of M500=ON, the bitwise logic XOR operation is executed on D10=2#1011011010010011 (46739) and D20=2#1001001100101110 (37678), and the result is assigned to D30 to obtain D30=2#0010010110111101 (9661).

3.10.5 WINV: Commands for Inversion Operation of Word/Doubleword Data

Command table	*WINV (S) (D)	Applicable model	TS600 series
16-Bit command	WINV: Logical inversion operation of word data		
32-Bit command	DWINV: Logical inversion operation of doubleword data		

Operand	Type	Bit			Word		Indexing	Constant
		X, Y, M, LM, T, C, S	Dx.y	Custom bit variable	D, R, V, Z, T, C	Custom word variable		
S	WORD/ DWORD	-	-	-	√[1]	√	√	√
D	WORD/ DWORD	-	-	-	√[1]	√	√	-

Remark:

[1] For 32-bit commands, the T and Z elements are not supported.

Operand Description

S: The source operand, which indicates the address of the data or data storage word soft element that participates in the inversion operation.

D: The destination operand, which indicates the address of the data storage word soft element of the operation result.

Function Description

1. When the command is driven, the bitwise logical inversion operation is executed on S, and the operation result is assigned to D.
2. The rules of logical inversion operation: If any data is 1, the result is 0; if any data is 0, the result is 1. For example: ~1=0   ~0=1.

Application Example

		Element Name	Data Type	Display Format	Current Value
1	...	D10	WORD	Binary	2#1011011010010011
2	...	D20	WORD	Binary	2#100100101101100

In case of M503=ON, the bitwise logical inversion operation is executed on D10=(46739), and the result is assigned to D20 to obtain D20=(18796).

3.11 Bit Shift Rotation Command

3.11.1 Command Table

Command Category	Name	Function
Bit shift rotation command	*ROR	16-bit/32-bit cyclic shift right
	*ROL	16-bit/32-bit cyclic shift left
	*RCR	16-bit/32-bit cyclic shift right with carry
	*RCL	16-Bit/32-bit cyclic shift left with carry
	*SHR	16-bit/32-bit shift right
	*SHL	16-bit/32-bit shift left
	SFTR	Bit string shift right
	SFTL	Bit string shift left

3.11.2 ROR: Commands for 16-Bit/32-Bit Cyclic Shift Right

Command table	*ROR (S) (D) (n)	Applicable model	TS600 series
16-Bit command	ROR: 16-bit cyclic shift right		
32-Bit command	DROR: 32-bit cyclic shift right		

Operand	Type	Bit			Word		Indexing	Constant
		X, Y, M, LM, T, C, S	Dx.y	Custom bit variable	D, R, V, Z, T, C	Custom word variable		
S	WORD/ DWORD	-	-	-	√[1]	√	√	√
D	WORD/ DWORD	-	-	-	√[1]	√	√	
n	WORD	-	-	-	√	√	√	√

Remark:

[1] For the 32-bit command DROR, the T and Z elements are not supported.

Operand Description

S: The source operand, which indicates the address of the data or data storage word soft element to be shifted to right.

D: The destination operand, which indicates the address of the data storage word soft element after being shifted to right.

n: The number of digits shifted for a single time, which ranges between 0 and 32767.

Function Description

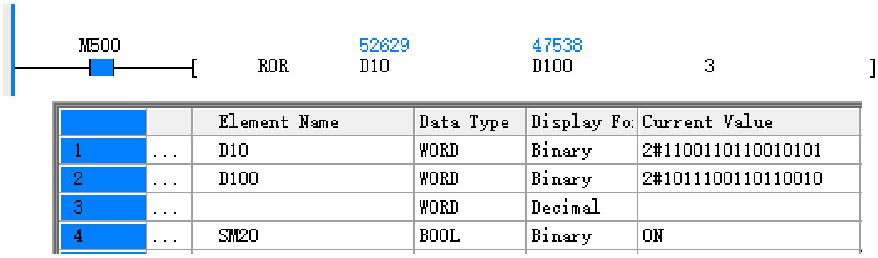
When the command is driven, the data of S is cyclically shifted to right for n digits, and the result is assigned to D. For the 32-bit command, the data consisting of S and (S+1) is cyclically shifted to right for n digits, the result is assigned to D and (D+1), and the final digit of the shift is stored in the carry flag bit SM20.

Precautions

When the value of n is greater than 32767, the system reports an operand error and the command does not run.

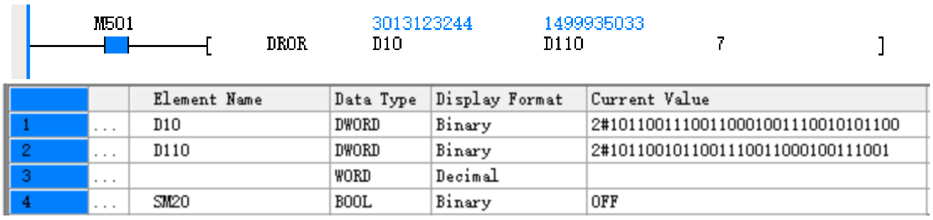
Application Example

ROR Command:



In case of M500=ON, D10=2#1100110110010101 (52629) is shifted to right for 3 digits, the result is assigned to D100, and the final digit of the shift is stored in the carry flag bit to obtain D100=2#1011100110110010 (47538) and SM20=ON.

DROR Command:



In case of M501=ON, (D10, D11)=2#10110011100110001001110010101100 (3013123244) is cyclically shifted to right for 7 digits, the result is assigned to (D110, D111), and the final digit of the shift is stored in the carry flag bit to obtain (D110,D111)=2#01011001011001110011000100111001 (1499935033) and SM20=OFF.

3.11.3 ROL: Commands for 16-Bit/32-Bit Cyclic Shift Left

Command table	*ROL (S) (D) (n)			Applicable model	TS600 series		
16-Bit command	ROL: 16-bit cyclic shift left						
32-Bit command	DROL: 32-bit cyclic shift left						

Operand	Type	Bit			Word		Indexing	Constant
		X, Y, M, LM, T, C, S	Dx.y	Custom bit variable	D, R, V, Z, T, C	Custom word variable		
S	WORD/ DWORD	-	-	-	√[1]	√	√	√
D	WORD/ DWORD	-	-	-	√[1]	√	√	-
n	WORD	-	-	-	√	√	√	√

Remark:

[1] For the 32-bit command DROL, the T and Z elements are not supported.

Operand Description

S: The source operand, which indicates the address of the data or data storage word soft element to be shifted to left.

D: The destination operand, which indicates the address of the data storage word soft element after being shifted to left.

n: The number of digits shifted for a single time, which ranges between 0 and 32767.

Function Description

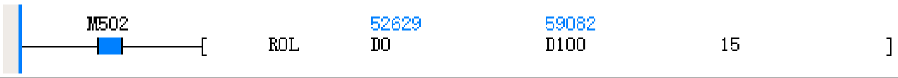
When the command is driven, the data of S is cyclically shifted to left for n digits, and the result is assigned to D. For the 32-bit command, the data consisting of S and (S+1) is cyclically shifted to left for n digits, the result is assigned to D and (D+1), and the final digit of the shift is stored in the carry flag bit (SM20).

Precautions

When the value of n is greater than 32767, the system reports an operand error and the command does not run.

Application Example

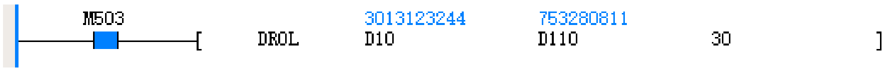
ROL Command:



		Element Name	Data Type	Display Fo:	Current Value
1	...	D0	WORD	Binary	2#1100110110010101
2	...	D100	WORD	Binary	2#1110011011001010
3	...		WORD	Decimal	
4	...	SM20	BOOL	Binary	OFF

In case of M502=ON, D0=2#1100110110010101 (52629) is cyclically shifted to left for 15 digits, the result is assigned to D100, and the final digit of the shift is stored in the carry flag bit to obtain D100=2#1110011011001010 (59082) and SM20=OFF.

DROL Command:



		Element Name	Data Type	Display Format	Current Value
1	...	D10	DWORD	Binary	2#10110011100110001001110010101100
2	...	D110	DWORD	Binary	2#101100111001100010011100101011
3	...		WORD	Decimal	
4	...	SM20	BOOL	Binary	ON

In case of M503=ON, (D10, D11)=2#10110011100110001001110010101100 (3013123244) is cyclically shifted to left for 30 digits, the result is assigned to (D110, D111), and the final digit of the shift is stored in the carry flag bit to obtain (D110,D111)=2#00101100111001100010011100101011 (753280811) and SM20=ON.

3.11.4 RCR: Commands for 16-Bit/32-Bit Cyclic Shift Right with Carry

<table>
<tr><td colspan="2">Command table</td><td colspan="4">*RCR (S) (D) (n)</td><td>Applicable model</td><td colspan="2">TS600 series</td></tr>
<tr><td colspan="2">16-Bit command</td><td colspan="7">RCR: 16-bit cyclic shift right with carry</td></tr>
<tr><td colspan="2">32-Bit command</td><td colspan="7">DRCR: 32-bit cyclic shift right with carry</td></tr>
<tr><td rowspan="2">Operand</td><td rowspan="2">Type</td><td colspan="3">Bit</td><td colspan="2">Word</td><td rowspan="2">Indexing</td><td rowspan="2">Constant</td></tr>
<tr><td>X, Y, M, LM, T, C, S</td><td>Dx.y</td><td>Custom bit variable</td><td>D, R, V, Z, T, C</td><td>Custom word variable</td></tr>
<tr><td>S</td><td>WORD/ DWORD</td><td>-</td><td>-</td><td>-</td><td>√[1]</td><td>√</td><td>√</td><td>√</td></tr>
<tr><td>D</td><td>WORD/ DWORD</td><td>-</td><td>-</td><td>-</td><td>√[1]</td><td>√</td><td>√</td><td>-</td></tr>
<tr><td>n</td><td>WORD</td><td>-</td><td>-</td><td>-</td><td>√</td><td>√</td><td>√</td><td>√</td></tr>
</table>

Remark:

[1] For the 32-bit command DRCR, the T and Z elements are not supported.

Operand Description

S: The source operand, which indicates the address of the data or data storage word soft element to be shifted to right.

D: The destination operand, which indicates the address of the data storage word soft element after being shifted to right.

n: The number of digits shifted for a single time, which ranges between 0 and 32767.

Function Description

When the command is driven, the data of S together with the carry (SM20) is cyclically shifted to right for n digits, and the result is assigned to D. For the 32-bit command, the data consisting of S and (S+1) together with the carry (SM20) is cyclically shifted to right for n digits, and the result is assigned to D and (D+1).

Precautions

When the value of n is greater than 32767, the system reports an operand error and the command does not run.

Application Example

RCR Command:

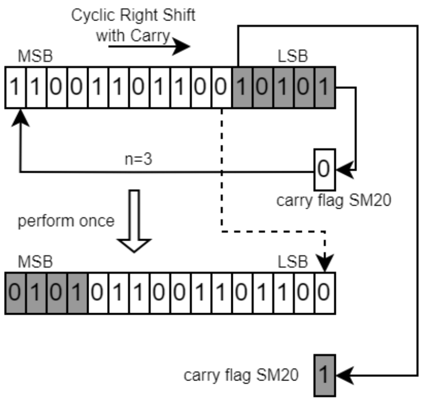
Before execution:

		Element Name	Data Type	Display Format	Current Value
1	...	D0	WORD	Binary	2#1100110110010101
2	...	D100	WORD	Binary	2#0
3	...		WORD	Decimal	
4	...	SM20	BOOL	Binary	OFF

After execution:

		Element Name	Data Type	Display Format	Current Value
1	...	D0	WORD	Binary	2#1100110110010101
2	...	D100	WORD	Binary	2#101011001101100
3	...		WORD	Decimal	
4	...	SM20	BOOL	Binary	OFF

In case of M504=ON, D0=2#1100110110010101 (52629) together with the carry (SM20=OFF) is cyclically shifted to right for 5 digits, and the result is assigned to D100 to obtain D100=2#0101011001101100 (22124) and SM20=ON. See the figure below for the process.



DRCR Command:

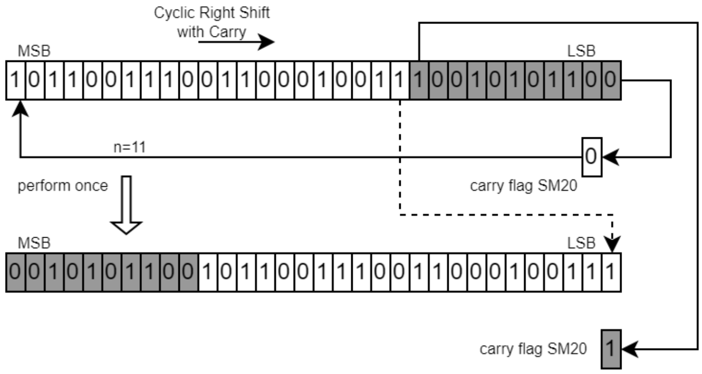
Before execution:

		Element Name	Data Type	Display Format	Current Value
1	...	D10	DWORD	Binary	2#10110011100110001001110010101100
2	...	D110	DWORD	Binary	2#0
3	...		WORD	Decimal	
4	...	SM20	BOOL	Binary	OFF
5	...		WORD	Decimal	

After execution:

		Element Name	Data Type	Display Format	Current Value
1	...	D10	DWORD	Binary	2#10110011100110001001110010101100
2	...	D110	DWORD	Binary	2#1010110011011001110011000100011
3	...		WORD	Decimal	
4	...	SM20	BOOL	Binary	ON
5	...		WORD	Decimal	

In case of M505=ON, (D0, D1)=2#10110011100110001001110010101100 (3013123244) together with the carry (SM20=OFF) is cyclically shifted to right for 11 digits, and the result is assigned to (D110, D111) to obtain (D110, D111)=2#00101011000101100111001100010011 (722891539) and SM20=ON. See the figure below for the process.



3.11.5 RCL: Commands for 16-Bit/32-Bit Cyclic Shift Left with Carry

Command table	*RCL (S) (D) (n)	Applicable model	TS600 series
16-Bit command	RCL: 16-bit cyclic shift left with carry		
32-Bit command	DRCL: 32-bit cyclic shift left with carry		

Operand	Type	Bit			Word		Indexing	Constant
		X, Y, M, LM, T, C, S	Dx.y	Custom bit variable	D, R, V, Z, T, C	Custom word variable		
S	WORD/ DWORD	-	-	-	√[1]	√	√	√
D	WORD/ DWORD	-	-	-	√[1]	√	√	-
n	WORD	-	-	-	√	√	√	√

Remark:

[1] For the 32-bit command DRCL, the T and Z elements are not supported.

Operand Description

S: The source operand, which indicates the address of the data or data storage word soft element to be shifted to left.

D: The destination operand, which indicates the address of the data storage word soft element after being shifted to left.

n: The number of digits shifted for a single time, which ranges between 0 and 32767.

Function Description

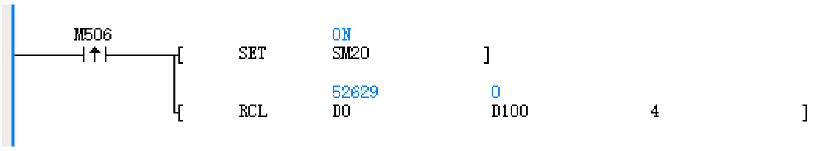
When the command is driven, the data of S together with the carry (SM20) is cyclically shifted to left for n digits, and the result is assigned to D. For the 32-bit command, the data consisting of S and (S+1) together with the carry (SM20) is cyclically shifted to left for n digits, and the result is assigned to D and (D+1).

Precautions

When the value of n is greater than 32767, the system reports an operand error and the command does not run.

Application Example

RCL Command:



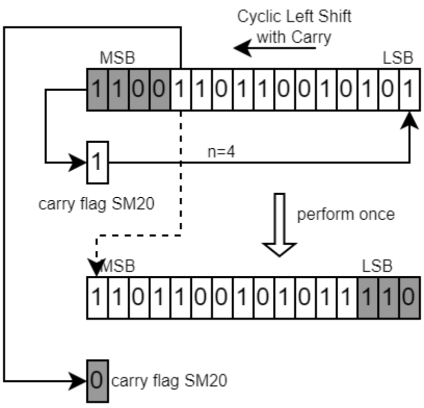
Before execution:

		Element Name	Data Type	Display Format	Current Value
1	...	D0	WORD	Binary	2#1100110110010101
2	...	D100	WORD	Binary	2#0
3	...		WORD	Decimal	
4	...	SM20	BOOL	Binary	ON

After execution:

		Element Name	Data Type	Display Format	Current Value
1	...	D0	WORD	Binary	2#1100110110010101
2	...	D100	WORD	Binary	2#1101100101011110
3	...		WORD	Decimal	
4	...	SM20	BOOL	Binary	OFF

In case of M506=ON, D0=2#1100110110010101 (52629) together with the carry (SM20=ON) is cyclically shifted to left for 4 digits, and the result is assigned to D100 to obtain D100=2#1101100101011110 (55646) and SM20=OFF. See the figure below for the process.



DRCL Command:

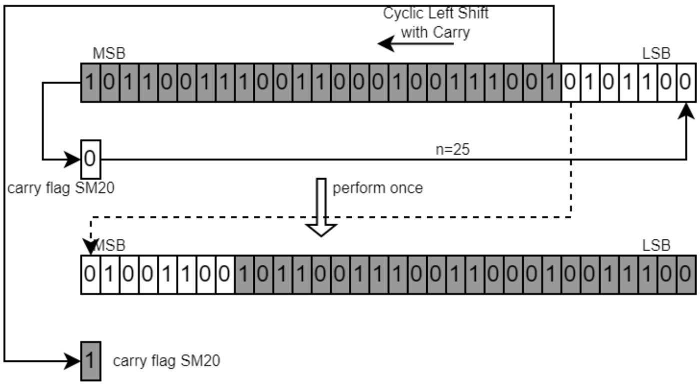
Before execution:

		Element Name	Data Type	Display Format	Current Value
1	...	D10	DWORD	Binary	2#10110011100110001001110010101100
2	...	D110	DWORD	Binary	2#0
3	...		WORD	Decimal	
4	...	SM20	BOOL	Binary	OFF

After execution:

		Element Name	Data Type	Display Format	Current Value
1	...	D10	DWORD	Binary	2#10110011100110001001110010101100
2	...	D110	DWORD	Binary	2#1011000101100111001100010011100
3	...		WORD	Decimal	
4	...	SM20	BOOL	Binary	ON

In case of M507=ON, (D10, D11)=2#10110011100110001001110010101100 (3013123244) together with the carry (SM20=OFF) is cyclically shifted to left for 25 digits, and the result is assigned to (D110, D111) to obtain (D110, D111)=2#00101100010110011100110001001110 0 (1488165020) and SM20=ON.



3.11.6 SHR: Commands for 16-Bit/32-Bit Shift Right

Command table	*SHR (S) (D) (n)				Applicable model	TS600 series		
16-Bit command	SHR: 16-bit shift right							
32-Bit command	DSHR: 32-bit shift right							
Operand	Type	Bit			Word		Indexing	Constant
		X, Y, M, LM, T, C, S	Dx.y	Custom bit variable	D, R, V, Z, T, C	Custom word variable		
S	WORD/ DWORD	-	-	-	√[1]	√	√	√
D	WORD/ DWORD	-	-	-	√[1]	√	√	
n	WORD/ DWORD	-	-	-	√	√	√	√

Remark:

[1] For the 32-bit command DSHR, the T and Z elements are not supported.

Operand Description

S: The source operand, which indicates the address of the data or data storage word soft element to be shifted to right.

D: The destination operand, which indicates the address of the data storage word soft element after being shifted to right.

n: The number of digits shifted for a single time, which ranges between 0 and 32767.

Function Description

When the command is driven, the data of S is shifted to right for n digits, and the result is assigned to D. For the 32-bit command, the data consisting of S and (S+1) together with the carry is shifted to right for n digits, and the result is assigned to D and (D+1). At the same time, the position after shift is padded with 0.

Precautions

When the value of n is greater than 32767, the system reports an operand error and the command does not run.

Application Example

		Element Name	Data Type	Display Fo:	Current Value
1	...	D0	WORD	Binary	2#111101011011100
2	...	D100	WORD	Binary	2#1111010110
3	...		WORD	Decimal	
4	...	D10	DWORD	Binary	2#1110011100110001001110010101100
5	...	D110	DWORD	Binary	2#111001110011000100111

In case of M508=ON, D0=2#0111101011011100 (31452) is shifted to right for 5 digits, and the result is assigned to D100 to obtain D100=2#0000001111010110 (982).

In case of M509=ON, (D10, D11)=2#01110011100110001001110010101100 (1939381420) is shifted to right for 10 digits, and the result is assigned to (D110, D111) to obtain (D110, D111)=2#00000000000111001110011000100111 (1893927).

3.11.7 SHL: Commands for 16-Bit/32-Bit Shift Left

Command table	*SHL (S) (D) (n)			Applicable model	TS600 series		
16-Bit command	SHL: 16-bit shift left						
32-Bit command	DSHL: 32-bit shift left						

Operand	Type	Bit			Word		Indexing	Constant
		X, Y, M, LM, T, C, S	Dx.y	Custom bit variable	D, R, V, Z, T, C	Custom word variable		
S	WORD/ DWORD	-	-	-	√[1]	√	√	√
D	WORD/ DWORD	-	-	-	√[1]	√	√	
n	WORD/ DWORD	-	-	-	√	√	√	√

Remark:

[1] For the 32-bit command DSHL, the T and Z elements are not supported.

Operand Description

S: The source operand, which indicates the address of the data or data storage word soft element to be shifted to left.

D: The destination operand, which indicates the address of the data storage word soft element after being shifted to left.

n: The number of digits shifted for a single time, which ranges between 0 and 32767.

Function Description

When the command is driven, the data of S is shifted to left for n digits, and the result is assigned to D. For the 32-bit command, the data consisting of S and (S+1) together with the carry is shifted to left for n digits, and the result is assigned to D and (D+1). At the same time, the position after shift is padded with 0.

Precautions

When the value of n is greater than 32767, the system reports an operand error and the command does not run.

Application Example

		Element Name	Data Type	Display Fo	Current Value
1	...	D0	WORD	Binary	2#111101011011100
2	...	D100	WORD	Binary	2#110111000000000
3	...		WORD	Decimal	
4	...	D10	DWORD	Binary	2#1110011100110001001110010101100
5	...	D110	DWORD	Binary	2#1001110010101100000000000000000

In case of M510=ON, D0=2#0111101011011100 (31452) is shifted to left for 7 digits, and the result is assigned to D100 to obtain D100=2#0110111000000000 (28160).

In case of M511=ON, (D10, D11)=2#01110011100110001001110010101100 (1939381420) is shifted to left for 15 digits, and the result is assigned to (D110, D111) to obtain (D110, D111)=2#01001110010101100000000000000000 (1314258944).

3.11.8 SFTR: Command for Bit String Shift Right

Command table		*SFTR (S) (D) (n1) (n2)			Applicable model	TS600 series		
16-Bit command		SFTR: Bit string shift right						
32-Bit command		-						
Operand	Type	Bit			Word		Indexing	Constant
		X, Y, M, LM, T, C, S	Dx.y	Custom bit variable	D, R, V, Z, T, C	Custom word variable		
S	BOOL, Array*n2	√	-	√	-	-	√	-
D	BOOL, Array*n1	√[1]	-	√	-	-	√	-
n1	WORD	-	-	-	√	√	√	√
n2	WORD	-	-	-	√	√	√	√

Remark:

[1] The X element is not supported.

Operand Description

S: The source operand, which indicates the starting address of the bit element to shift the data in.

D: The destination operand, which indicates the starting address of the bit element shifting the data.

n1: The number of bit elements shifting the data, which ranges between 1 and 256.

n2: The number of bit elements to shift the data in, which ranges between 1 and n1.

Function Description

When the command is driven, the contents of the n1 units starting from the D unit are shifted to the right for n2 units, and the rightmost n2 data are discarded. At the same time, the contents of the n2 units starting from the S unit are shifted into the left end of the word string.

Precautions

- Please note the left-right order. For this command, larger element numbers are arranged to left, while smaller element numbers are arranged to right.
- In case of n1>256 or n2>n1, the system reports an operand error and this command is not executed.

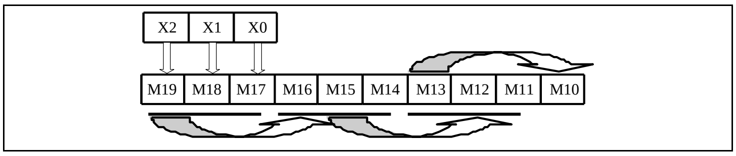
Application Example

Before execution:

		Element Name	Data Type	Display Fo:	Current Value
1	...	Y0	BOOL	Binary	ON
2	...	Y1	BOOL	Binary	OFF
3	...	Y2	BOOL	Binary	ON
4	...		WORD	Decimal	
5	...	M10	BOOL	Binary	OFF
6	...	M11	BOOL	Binary	ON
7	...	M12	BOOL	Binary	ON
8	...	M13	BOOL	Binary	OFF
9	...	M14	BOOL	Binary	OFF
10	...	M15	BOOL	Binary	ON
11	...	M16	BOOL	Binary	OFF
12	...	M17	BOOL	Binary	OFF
13	...	M18	BOOL	Binary	OFF
14	...	M19	BOOL	Binary	ON

After execution:

		Element Name	Data Type	Display Fo:	Current Value
1	...	Y0	BOOL	Binary	ON
2	...	Y1	BOOL	Binary	OFF
3	...	Y2	BOOL	Binary	ON
4	...		WORD	Decimal	
5	...	M10	BOOL	Binary	OFF
6	...	M11	BOOL	Binary	OFF
7	...	M12	BOOL	Binary	ON
8	...	M13	BOOL	Binary	OFF
9	...	M14	BOOL	Binary	OFF
10	...	M15	BOOL	Binary	OFF
11	...	M16	BOOL	Binary	ON
12	...	M17	BOOL	Binary	ON
13	...	M18	BOOL	Binary	OFF
14	...	M19	BOOL	Binary	ON



In case of M512=ON, the contents (taking a bit as the unit) of the 10 units starting from the M10 unit are shifted to the right for 3 units, and the rightmost M10 to M12 units are discarded. At the same time, the contents of the 3 units starting from the Y0 unit are shifted into the left end of the bit string:

Before execution: Y0=1, Y1=0, and Y2=1. M10=0, M11=1, M12=1, M13=0, M14=0, M15=1, M16=0, M17=0, M18=0, and M19=1;

After execution: The contents of Y0 to Y2 remain unchanged. M10=0, M11=0, M12=1, M13=0, M14=0, M15=0, M16=1, M17=1, M18=0, and M19=1.

3.11.9 SFTL: Commands for Bit String Shift Left

Command table		*SFTL (S) (D) (n1) (n2)			Applicable model	TS600 series		
16-Bit command		SFTL: Bit string shift left						
32-Bit command		-						
Operand	Type	Bit			Word		Indexing	Constant
		X, Y, M, LM, T, C, S	Dx.y	Custom bit variable	D, R, V, Z, T, C	Custom word variable		
S	BOOL, Array*n2	√	-	√	-	-	√	-
D	BOOL, Array*n1	√[1]	-	√	-	-	√	-
n1	WORD	-	-	-	√	√	√	√
n2	WORD	-	-	-	√	√	√	√

Remark:

[1] The X element is not supported.

Operand Description

S: The source operand, which indicates the starting address of the bit element to shift the data in.

D: The destination operand, which indicates the starting address of the bit element shifting the data.

n1: The number of bit elements shifting the data, which ranges between 1 and 256.

n2: The number of bit elements to shift the data in, which ranges between 1 and n1.

Function Description

When the command is driven, the contents of the n1 units starting from the D unit are shifted to the left for n2 units, and the leftmost n2 data are discarded. At the same time, the contents of the n2 units starting from the S unit are shifted into the right end of the word string.

Precautions

- Please note the left-right order. For this command, larger element numbers are arranged to left, while smaller element numbers are arranged to right.
- In case of n1>256 or n2>n1, the system reports an operand error and this command is not executed.

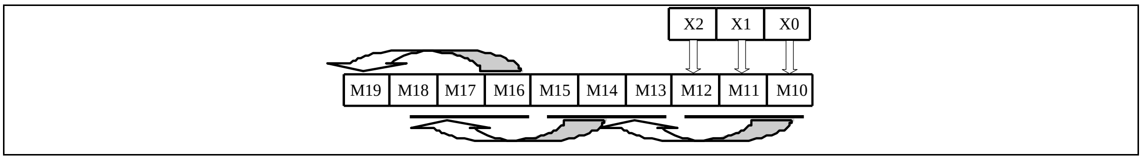
Application Example

Before execution:

		Element Name	Data Type	Display Format	Current Value
1	...	Y0	BOOL	Binary	ON
2	...	Y1	BOOL	Binary	OFF
3	...	Y2	BOOL	Binary	ON
4	...		WORD	Decimal	
5	...	M10	BOOL	Binary	OFF
6	...	M11	BOOL	Binary	ON
7	...	M12	BOOL	Binary	ON
8	...	M13	BOOL	Binary	OFF
9	...	M14	BOOL	Binary	OFF
10	...	M15	BOOL	Binary	ON
11	...	M16	BOOL	Binary	OFF
12	...	M17	BOOL	Binary	OFF
13	...	M18	BOOL	Binary	OFF
14	...	M19	BOOL	Binary	ON

After execution:

		Element Name	Data Type	Display Format	Current Value
1	...	Y0	BOOL	Binary	ON
2	...	Y1	BOOL	Binary	OFF
3	...	Y2	BOOL	Binary	ON
4	...		WORD	Decimal	
5	...	M10	BOOL	Binary	ON
6	...	M11	BOOL	Binary	OFF
7	...	M12	BOOL	Binary	ON
8	...	M13	BOOL	Binary	OFF
9	...	M14	BOOL	Binary	ON
10	...	M15	BOOL	Binary	ON
11	...	M16	BOOL	Binary	OFF
12	...	M17	BOOL	Binary	OFF
13	...	M18	BOOL	Binary	ON
14	...	M19	BOOL	Binary	OFF



In case of M513=ON, the contents (taking a bit as the unit) of the 10 units starting from the M10 unit are shifted to the left for 3 units, and the leftmost M10 to M12 units are discarded. At the same time, the contents of the 3 units starting from the Y0 unit are shifted into the right end of the bit string:

Before execution: Y0=1, Y1=0, and Y2=1. M10=0, M11=1, M12=1, M13=0, M14=0, M15=1, M16=0, M17=0, M18=0, and M19=1;

After execution: The contents of Y0 to Y2 remain unchanged. M10=1, M11=0, M12=1, M13=0, M14=1, M15=1, M16=0, M17=0, M18=1, and M19=0.

3.12 Enhanced Bit Processing Command

3.12.1 Command Table

<table>
<tr><th>Command Category</th><th>Name</th><th>Function</th></tr>
<tr><td rowspan="6">Enhanced Bit Processing Command</td><td>ZRST</td><td>Batch bit reset</td></tr>
<tr><td>ZSET</td><td>Batch bit set</td></tr>
<tr><td>DECO</td><td>Decode</td></tr>
<tr><td>ENCO</td><td>Encode</td></tr>
<tr><td>*BITS</td><td>ON bit statistics in word/doubleword</td></tr>
<tr><td>BON</td><td>ON bit judgment in word</td></tr>
</table>

3.12.2 ZRST: Commands for Batch Bit Reset

<table>
<tr><td colspan="2">Command table</td><td colspan="3">ZRST (D) (S)</td><td>Applicable model</td><td colspan="3">TS600 series</td></tr>
<tr><td colspan="2">16-Bit command</td><td colspan="7">ZRST: Batch bit reset</td></tr>
<tr><td colspan="2">32-Bit command</td><td colspan="7">-</td></tr>
<tr><td rowspan="2">Operand</td><td rowspan="2">Type</td><td colspan="3">Bit</td><td colspan="2">Word</td><td rowspan="2">Indexing</td><td rowspan="2">Constant</td></tr>
<tr><td>X, Y, M, LM, T, C, S</td><td>Dx.y</td><td>Custom bit variable</td><td>D, R, V, Z, T, C</td><td>Custom word variable</td></tr>
<tr><td>D</td><td>BOOL</td><td>√[1]</td><td>-</td><td>-</td><td>-</td><td>-</td><td>√</td><td>-</td></tr>
<tr><td>S</td><td>WORD</td><td>-</td><td>-</td><td>-</td><td>√</td><td>√</td><td>√</td><td>√</td></tr>
</table>

Remark:

[1] The X element is not supported.

Operand Description

D: The destination operand.

S: The source operand.

Function Description

When the energy flow is valid, the S consecutive bit element units starting from the D unit are reset to zero.

Precautions

- When the cleared bit element is C, the counter value in the C element is also reset to zero.
- When the cleared bit element is T, the timing value in the T element is also reset to zero.

Application Example

In case of SM0=ON, all data of the 10 units starting from M10 (which are M10, M11, M12, ..., and M19) are reset to zero.

3.12.3 ZSET: Commands for Batch Bit Set

Command table	ZSET (D) (S)			Applicable model	TS600 series		
16-Bit command	ZSET: Batch bit set						
32-Bit command	-						

Operand	Type	Bit			Word		Indexing	Constant
		X, Y, M, LM, T, C, S	Dx.y	Custom bit variable	D, R, V, Z, T, C	Custom word variable		
D	BOOL	√[1]		-	-	-	√	-
S	WORD	-	-	-	√	√	√	√

Remark:

[1] The X element is not supported.

Operand Description

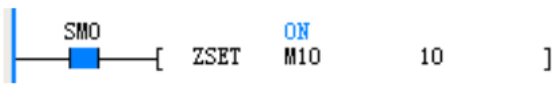
D: The destination operand.

S: The source operand.

Function Description

When the energy flow is valid, the S consecutive bit element units starting from the D unit are set to 1.

Application Example



In case of SM0=ON, all data of the 10 units starting from M10 (which are M10, M11, M12, ..., and M19) are set to 1.

3.12.4 DECO: Decode Commands

Command table	DECO (D) (S)			Applicable model	TS600 series		
16-Bit command	DECO: Decode						
32-Bit command	-						

Operand	Type	Bit			Word		Indexing	Constant
		X, Y, M, LM, T, C, S	Dx.y	Custom bit variable	D, R, V, Z, T, C	Custom word variable		
S	WORD	-	-	-	√	√	√	√
D	INT	-	-	-	√	√	√	-

Operand Description

S: The source operand.

D: The destination operand.

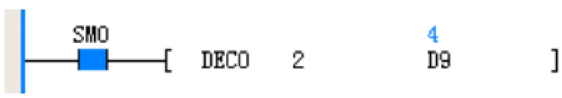
Function Description

When the energy flow is valid, the S-th bit in the word element D is set to 1 and other bits are reset to 0.

Precautions

- The effective range of S is 0~15.
- When S is greater than 15 or less than 0 and the energy flow is valid, the value of D is not changed, but a command operand value error is reported.

Application Example



When the energy flow is valid, the 2nd bit in D9 is set to 1 and other bits are reset to 0.

3.12.5 ENCO: Encode Commands

Command table		ENCO (S) (D)			Applicable model	TS600 series		
16-Bit command		ENCO: Encode						
32-Bit command		-						
Operand	Type	Bit			Word		Indexing	Constant
		X, Y, M, LM, T, C, S	Dx.y	Custom bit variable	D, R, V, Z, T, C	Custom word variable		
S	INT	-	-	-	√	√	√	√
D	INT	-	-	-	√	√	√	-

Operand Description

S: The source operand.

D: The destination operand.

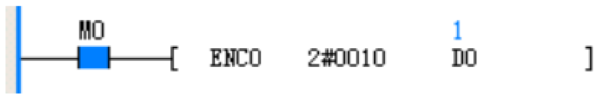
Function Description

When the energy flow is valid, the bit number with "1" in the word element S is written into D.

Precautions

When 1 appears at multiple bits in S, the smallest bit number is written into D. See the figure below.

Application Example



When the energy flow is valid, operand 1 is 2#0010, and the first bit is "1", so the result is 1 and written into D0.

3.12.6 BITS: Commands for ON Bit Statistics in Word/Doubleword

Command table		BITS (S) (D)			Applicable model	TS600 series		
16-Bit command		BITS: ON bit statistics in word						
32-Bit command		DBITS: ON bit statistics in doubleword						
Operand	Type	Bit			Word		Indexing	Constant
		X, Y, M, LM, T, C, S	Dx.y	Custom bit variable	D, R, V, Z, T, C	Custom word variable		
S	WORD/ DWORD	-	-	-	√[1]	√	√	√
D	INT	-	-	-	√	√	√	-

Remark:

[1] For 32-bit commands, the Z and T elements are not supported.

Operand Description

S: The source operand.

D: The destination operand.

Function Description

When the energy flow is valid, the number of bits with "1" in the doubleword S is counted, and the statistical result is stored in D.

Application Example

When the energy flow is valid, S in the BITS command is a constant, namely 16#F0F0, and 8 bits are "1" (ON state), so the statistical result is 8 and stored in D (D1).

When the energy flow is valid, S in the DBITS command is a constant, namely 16#FF0FF, and 16 bits are "1" (ON state), so the statistical result is 16 and stored in D (D10).

3.12.7 BON: Commands for ON Bit Judgment in Word

Command table		BON (S1) (D) (S2)			Applicable model	TS600 series		
16-Bit command		BON: ON bit judgment in word						
32-Bit command		-						
Operand	Type	Bit			Word		Indexing	Constant
		X, Y, M, LM, T, C, S	Dx.y	Custom bit variable	D, R, V, Z, T, C	Custom word variable		
S1	INT	-	-	-	√	√	√	√
D	BOOL	√[1]	-	-	√[1]	√	-	-
S2	WORD	-	-	-	√[2]	√	-	-

Remark:

[1] The X, C, and T elements are not supported.

[2] The D, V, and R elements are supported.

Operand Description

S: The source operand.

D: The destination operand.

Function Description

When the energy flow is valid, the state of the S2-nd bit in the word S1 is judged, and the result is output to D.

Application Example

When the energy flow is valid, S1 in the BON command is a constant, namely D0, and the state of the 5th bit is (ON) and output to D (Y0).

3.13 Word Contact Command

3.13.1 Command Table

Command Category	Name	Function
Word Contact Command	BLD	Normally open contact of word bit data
	BLDI	Normally closed contact of word bit data
	BAND	Serial connection to normally open contact of AND word bit data
	BANI	Serial connection to normally closed contact of AND word bit data

Command Category	Name	Function
	BOR	Serial connection to normally open contact of OR word bit data
	BORI	Serial connection to normally closed contact of OR word bit data
	LD*&	Logical AND operation of word/doubleword LD contact
	LD*|	Logical OR operation of word/doubleword LD contact
	LD*^	Logical XOR operation of word/doubleword LD contact
	AND*&	Logical AND operation of word/doubleword AND contact
	AND*|	Logical OR operation of word/doubleword AND contact
	AND*^	Logical XOR operation of word/doubleword AND contact
	OR*&	Logical AND operation of word/doubleword OR contact
	OR*|	Logical OR operation of word/doubleword OR contact
	OR*^	Logical XOR operation of word/doubleword OR contact
	BOUT	Word bit data coil output
	BSET	Word bit data coil set
	BRST	Word bit data coil reset

3.13.2 BLD&BLDI: Commands for Contact of Word Bit Data

Command table	BLD* (S1) (S2)				Applicable model	TS600 series		
16-Bit command	BLD: Normally open contact of word bit data							
32-Bit command	-							
16-Bit command	BLDI: Normally closed contact of word bit data							
32-Bit command	-							
Operand	Type	Bit			Word		Indexing	Constant
		X, Y, M, LM, T, C, S	Dx.y	Custom bit variable	D, R, V, Z, T, C	Custom word variable		
S1	WORD	-	-	-	√	√	√	-
S2	WORD	-	-	-	√	√	√	√

Operand Description

S1: The source operand.

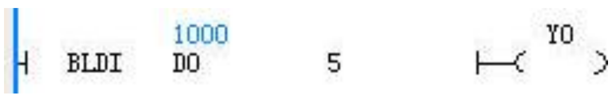
S2: The specified bit, which ranges between 0 and 15, otherwise an operand error is reported.

Function Description

1. BLD: The state of the S2-nd bit in the content of the S1 unit is taken to drive the subsequent segment operation.
2. BLDI: The logical NOT of the state of the S2-nd bit in the content of the S1 unit is taken to drive the subsequent segment operation.

Application Example

The logical NOT (OFF) of the BIT5 state (ON) of D0 (1000: 2#0000001111101000) is taken to determines the output state of the subsequent segment element Y0.



The logical NOT (OFF) of the BIT5 state (ON) of D0 (1000: 2#0000001111101000) is taken to determines the output state of the subsequent segment element Y0.

3.13.3 BAND&BANI: Commands for Serial Connection to Contact of Word Bit Data

Command table		BAN* (S1) (S2)			Applicable model	TS600 series		
16-Bit command		BAND: Serial connection to normally open contact of AND word bit data						
32-Bit command		-						
16-Bit command		BANI: Serial connection to normally closed contact of AND word bit data						
32-Bit command		-						
Operand	Type	Bit			Word		Indexing	Constant
		X, Y, M, LM, T, C, S	Dx.y	Custom bit variable	D, R, V, Z, T, C	Custom word variable		
S1	WORD	-	-	-	√	√	√	-
S2	WORD	-	-	-	√	√	√	√

Operand Description

S1: The source operand.

S2: The specified bit, which ranges between 0 and 15, otherwise an operand error is reported.

Function Description

1. BAND: The state of the S2-nd bit in the content of the S1 unit is taken and connected in series with other nodes to drive the subsequent segment operation.
2. BANI: The logical NOT of the state of the S2-nd bit in the content of the S1 unit is taken and connected in series with other nodes to drive the subsequent segment operation.

Application Example

The BIT5 state (ON) of D0 (1000: 2#0000001111101000) is taken and connected in series with other nodes (X0=ON) to determine the output state of the subsequent segment element Y0.

The logical NOT (OFF) of the BIT5 state (ON) of D0 (1000: 2#0000001111101000) is taken and connected in series with other nodes (X0=ON) to determine the output state of the subsequent segment element Y0.

3.13.4 BOR&BORI: Commands for Parallel Connection to Contact of Word Bit Data

Command table		BOR* (S1) (S2)			Applicable model	TS600 series		
16-Bit command		BOR: Parallel connection to normally open contact of OR word bit data						
32-Bit command		-						
16-Bit command		BORI: Parallel connection to normally open contact of OR word bit data						
32-Bit command		-						
Operand	Type	Bit			Word		Indexing	Constant
		X, Y, M, LM, T, C, S	Dx.y	Custom bit variable	D, R, V, Z, T, C	Custom word variable		
S1	WORD	-	-	-	√	√	√	-
S2	WORD	-	-	-	√	√	√	√

Operand Description

S1: The source operand.

S2: The specified bit, which ranges between 0 and 15, otherwise an operand error is reported.

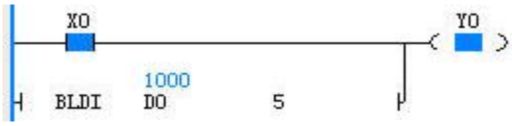
Function Description

1. BOR: The state of the S2-nd bit in the content of the S1 unit is taken and connected in parallel with other nodes to drive the subsequent segment operation.

2. BORI: The logical NOT of the state of the S2-nd bit in the content of the S1 unit is taken and connected in parallel with other nodes to drive the subsequent segment operation.

Application Example

The BIT5 state (ON) of D0 (1000: 2#0000001111101000) is taken and connected in parallel with other nodes (X0=ON) to determine the output state of the subsequent segment element Y0.



The logical NOT (X0= OFF) of the BIT5 state (ON) of D0 (1000: 2#0000001111101000) is taken and connected in parallel with other nodes (X0=ON) to determine the output state of the subsequent segment element Y0.

3.13.5 LD*: LD Logic Operation Commands

<table>
<tr><td colspan="2">Command table</td><td colspan="3">LD* (S1) (S2)</td><td>Applicable model</td><td colspan="3">TS600 series</td></tr>
<tr><td colspan="2">16-Bit command</td><td colspan="7">LD&: Logical AND operation of word LD contact</td></tr>
<tr><td colspan="2">32-Bit command</td><td colspan="7">LDD&: Logical AND operation of doubleword LD contact</td></tr>
<tr><td colspan="2">16-Bit command</td><td colspan="7">LD|: Logical OR operation of word LD contact</td></tr>
<tr><td colspan="2">32-Bit command</td><td colspan="7">LDD|: Logical OR operation of doubleword LD contact</td></tr>
<tr><td colspan="2">16-Bit command</td><td colspan="7">LD^: Logical XOR operation of word LD contact</td></tr>
<tr><td colspan="2">32-Bit command</td><td colspan="7">LDD^: Logical XOR operation of doubleword LD contact</td></tr>
<tr><td rowspan="2">Operand</td><td rowspan="2">Type</td><td colspan="3">Bit</td><td colspan="2">Word</td><td rowspan="2">Indexing</td><td rowspan="2">Constant</td></tr>
<tr><td>X, Y, M, LM, T, C, S</td><td>Dx.y</td><td>Custom bit variable</td><td>D, R, V, Z, T, C</td><td>Custom word variable</td></tr>
<tr><td>S1</td><td>INT/DINT</td><td>-</td><td>-</td><td>-</td><td>√</td><td>√</td><td>√</td><td>√</td></tr>
<tr><td>S2</td><td>INT/DINT</td><td>-</td><td>-</td><td>-</td><td>√</td><td>√</td><td>√</td><td>√</td></tr>
</table>

Operand Description

S1: Source operand, which indicates logical judgment data 1.

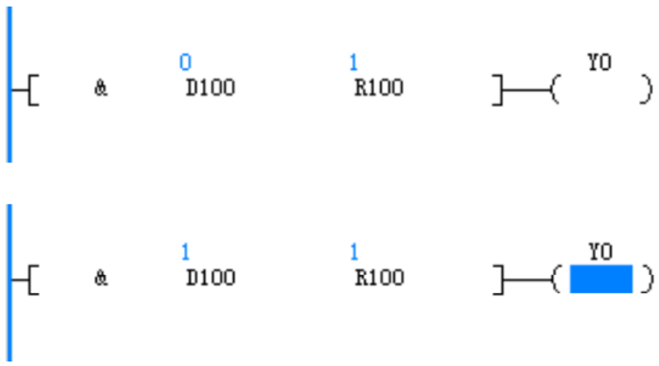
S2: Source operand, which indicates logical judgment data 2.

Function Description

The command performs a logical operation ("AND, &", "NOT, |", or "XOR, ^") on the contents of S1 and S2. If the result is not 0, the command is on-state; if the result is 0, the command is off-state.

16-Bit command	32-Bit command	On-state condition	Off-state condition
LD&	LDD&	S1&S2≠0	S1&S2=0
LD|	LDD|	S1|S2≠0	S1|S2=0
LD^	LDD^	S1^S2≠0	S1^S2=0

Application Example



3.13.6 AND*: AND Logic Operation Commands

Command table	AND* (S1) (S2)			Applicable model	TS600 series		
16-Bit command	AND &: Logical AND operation of word AND contact						
32-Bit command	ANDD &: Logical AND operation of doubleword AND contact						
16-Bit command	AND |: Logical OR operation of word AND contact						
32-Bit command	ANDD |: Logical OR operation of doubleword AND contact						
16-Bit command	AND ^: Logical XOR operation of word AND contact						
32-Bit command	ANDD ^: Logical XOR operation of doubleword AND contact						

Operand	Type	Bit			Word		Indexing	Constant
		X, Y, M, LM, T, C, S	Dx.y	Custom bit variable	D, R, V, Z, T, C	Custom word variable		
S1	INT/DINT	-	-	-	√	√	√	√
S2	INT/DINT	-	-	-	√	√	√	√

Operand Description

S1: Source operand, which indicates logical judgment data 1.

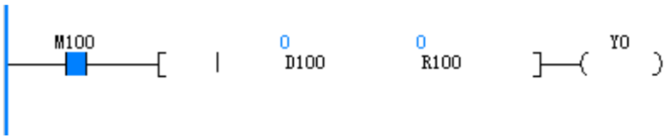
S2: Source operand, which indicates logical judgment data 2.

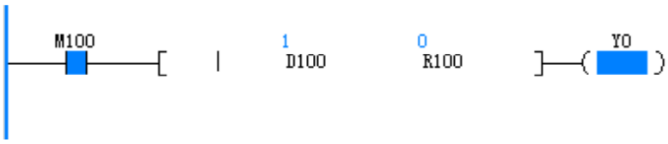
Function Description

The command performs a logical operation ("AND, &", "NOT, |", or "XOR, ^") on the contents of S1 and S2. If the result is not 0, the command is on-state; if the result is 0, the command is off-state.

16-Bit command	32-Bit command	On-state condition	Off-state condition
AND&	ANDD&	S1&S2≠0	S1&S2=0
AND|	ANDD|	S1|S2≠0	S1|S2=0
AND^	ANDD^	S1^S2≠0	S1^S2=0

Application Example





3.13.7 OR*: OR Logic Operation Commands

Command table	OR * (S1) (S2)			Applicable model	TS600 series		
16-Bit command	OR &: Logical AND operation of word OR contact						
32-Bit command	ORD &: Logical AND operation of doubleword OR contact						
16-Bit command	OR |: Logical OR operation of word OR contact						
32-Bit command	ORD |: Logical OR operation of doubleword OR contact						
16-Bit command	OR ^: Logical XOR operation of word OR contact						
32-Bit command	ORD^: Logical XOR operation of doubleword OR contact						

Operand	Type	Bit			Word		Indexing	Constant
		X, Y, M, LM, T, C, S	Dx.y	Custom bit variable	D, R, V, Z, T, C	Custom word variable		
S1	INT/DINT	-	-	-	√	√	√	√
S2	INT/DINT	-	-	-	√	√	√	√

Operand Description

S1: Source operand, which indicates logical judgment data 1.

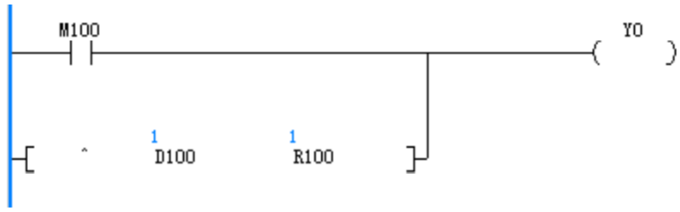
S2: Source operand, which indicates logical judgment data 2.

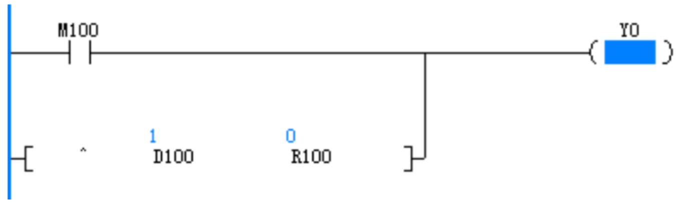
Function Description

The command performs a logical operation ("AND, &", "NOT, |", or "XOR, ^") on the contents of S1 and S2. If the result is not 0, the command is on-state; if the result is 0, the command is off-state.

16-Bit command	32-Bit command	On-state condition	Off-state condition
OR&	ORD&	S1&S2≠0	S1&S2=0
OR|	ORD|	S1|S2≠0	S1|S2=0
OR^	ORD^	S1^S2≠0	S1^S2=0

Application Example





3.13.8 BOUT: Commands for Word Bit Data Coil Output

<table>
<tr><td>Command table</td><td colspan="3">BOUT (D) (S)</td><td>Applicable model</td><td colspan="4">TS600 series</td></tr>
<tr><td colspan="2">16-Bit command</td><td colspan="7">BOUT: Word bit data coil output</td></tr>
<tr><td colspan="2">32-Bit command</td><td colspan="7">-</td></tr>
<tr><td rowspan="2">Operand</td><td rowspan="2">Type</td><td colspan="3">Bit</td><td colspan="2">Word</td><td rowspan="2">Indexing</td><td rowspan="2">Constant</td></tr>
<tr><td>X, Y, M, LM, T, C, S</td><td>Dx.y</td><td>Custom bit variable</td><td>D, R, V, Z, T, C</td><td>Custom word variable</td></tr>
<tr><td>D</td><td>WORD</td><td>-</td><td>-</td><td>-</td><td>√</td><td>√</td><td>√</td><td>-</td></tr>
<tr><td>S</td><td>WORD</td><td>-</td><td>-</td><td>-</td><td>√</td><td>√</td><td>√</td><td>√</td></tr>
</table>

Operand Description

D: The source operand.

S: The specified bit, which ranges between 0 and 15, otherwise an operand error is reported.

Function Description

The current energy flow state is assigned to the S bit of D.

Application Example

The current energy flow state (M0=ON) is assigned to BIT4 of D0(1000: 2#0000001111101000). After execution: D0=1016 (2#0000001111111000).

3.13.9 BSET: Commands for Word Bit Data Coil Set

Command table		BSET (D) (S)			Applicable model	TS600 series		
16-Bit command		BSET: Word bit data coil set						
32-Bit command		-						
Operand	Type	Bit			Word		Indexing	Constant
		X, Y, M, LM, T, C, S	Dx.y	Custom bit variable	D, R, V, Z, T, C	Custom word variable		
D	WORD	-	-	-	√	√	√	-
S	WORD	-	-	-	√	√	√	√

Operand Description

D: The source operand.

S: The specified bit, which ranges between 0 and 15, otherwise an operand error is reported.

Function Description

The command sets the S bit of the D element.

Application Example

When the energy flow is valid, the command sets BIT15 of D0 (1000: 2#0000001111101000). After execution: D0=33768 (2#1000001111101000).

3.13.10 BRST: Commands for Word Bit Data Coil Reset

Command table		BRST (D) (S)			Applicable model	TS600 series		
16-Bit command		BRST: Word bit data coil reset						
32-Bit command		-						
Operand	Type	Bit			Word		Indexing	Constant
		X, Y, M, LM, T, C, S	Dx.y	Custom bit variable	D, R, V, Z, T, C	Custom word variable		
D	WORD	-	-	-	√	√	√	-
S	WORD	-	-	-	√	√	√	√

Operand Description

D: The source operand.

S: The specified bit, which ranges between 0 and 15, otherwise an operand error is reported.

Function Description

The command resets the S bit of the D element.

Application Example

When the energy flow is valid, the command resets BIT8 of D0 (1000: 2#0000001111101000). After execution: D0=744(2#0000001011101000)

3.14 Contact Comparison Command

3.14.1 Command Table

Command Category	Name	Function
Contact Comparison Command	LD*=	Integer/long integer LD contact comparison equal to
	LD*>	Integer/long integer LD contact comparison greater than
	LD*<	Integer/long integer LD contact comparison less than
	LD*<>	Integer/long integer LD contact comparison not equal to
	LD*>=	Integer/long integer LD contact comparison greater than or equal to
	LD*<=	Integer/long integer LD contact comparison less than or equal to
	AND*=	Integer/long integer AND contact comparison equal to
	AND*>	Integer/long integer AND contact comparison greater than
	AND*<	Integer/long integer AND contact comparison less than
	AND*<>	Integer/long integer AND contact comparison not equal to
	AND*>=	Integer/long integer AND contact comparison greater than or equal to
	AND*<=	Integer/long integer AND contact comparison less than or equal to
	OR*=	Integer/long integer OR contact comparison equal to
	OR*>	Integer/long integer OR contact comparison greater than
	OR*<	Integer/long integer OR contact comparison less than
	OR*<>	Integer/long integer OR contact comparison not equal to
	OR*>=	Integer/long integer OR contact comparison greater than or equal to
	OR*<=	Integer/long integer OR contact comparison less than or equal to
	LDR=	Floating-point number LD contact comparison equal to
	LDR>	Floating-point number LD contact comparison greater than
	LDR<	Floating-point number LD contact comparison less than
	LDR<>	Floating-point number LD contact comparison not equal to
	LDR>=	Floating-point number LD contact comparison greater than or equal
	LDR<=	Floating-point number LD contact comparison less than or equal
	ANDR=	Floating-point number AND contact comparison equal to
	ANDR>	Floating-point number AND contact comparison greater than
	ANDR<	Floating-point number AND contact comparison less than
	ANDR<>	Floating-point number AND contact comparison not equal to
	ANDR>=	Floating-point number AND contact comparison greater than or equal
	ANDR<=	Floating-point number AND contact comparison less than or equal
	ORR=	Floating-point number OR contact comparison equal to
	ORR<	Floating-point number OR contact comparison greater than
	ORR>	Floating-point number OR contact comparison less than
	ORR<>	Floating-point number OR contact comparison not equal to
	ORR>=	Floating-point number OR contact comparison greater than or equal
	ORR<=	Floating-point number OR contact comparison less than or equal
	CMP	Integer comparison set
	LCMP	Long integer comparison set
	RCMP	Floating-point number comparison set
	*ZCP	Word/doubleword data region comparison set
	RZCP	Floating-point number region comparison set

3.14.2 LD (=, <, >, <>, >=, <=): Commands for Integer/Long Integer LD Contact Comparison

Command table	LD* (S1) (S2)	Applicable model	TS600 series
16-Bit command	LD =: Integer LD contact comparison equal to		
32-Bit command	LDD =: Long integer LD contact comparison equal to		
16-Bit command	LD <: Integer LD contact comparison less than		
32-Bit command	LDD <: Long integer LD contact comparison less than		
16-Bit command	LD >: Integer LD contact comparison greater than		
32-Bit command	LDD >: Long integer LD contact comparison greater than		
16-Bit command	LD <>: Integer LD contact comparison not equal to		
32-Bit command	LDD <>: Long integer LD contact comparison not equal to		
16-Bit command	LD >=: Integer LD contact comparison greater than or equal to		
32-Bit command	LDD >=: Long integer LD contact comparison greater than or equal to		
16-Bit command	LD <=: Integer LD contact comparison less than or equal to		
32-Bit command	LDD <=: Long integer LD contact comparison less than or equal to		

Operand	Type	Bit			Word		Indexing	Constant
		X, Y, M, LM, T, C, S	Dx.y	Custom bit variable	D, R, V, Z, T, C	Custom word variable		
S1	INT/DINT	-	-	-	√[1]	√	√	√
S2	INT/DINT	-	-	-	√[1]	√	√	√

Remark:

[1] For 32-bit commands, the Z and T elements are not supported.

Operand Description

S1: Comparison parameter 1.

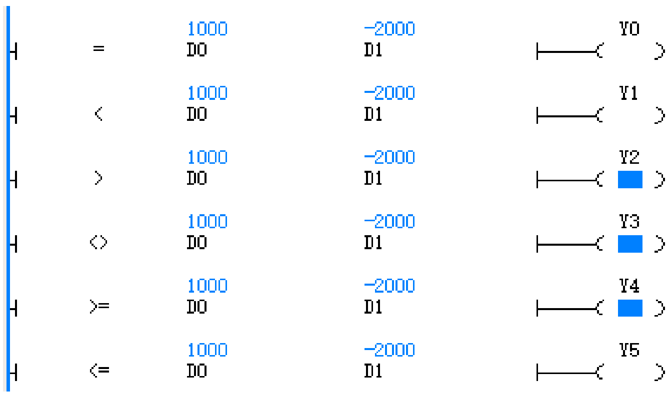
S2: Comparison parameter 2.

Function Description

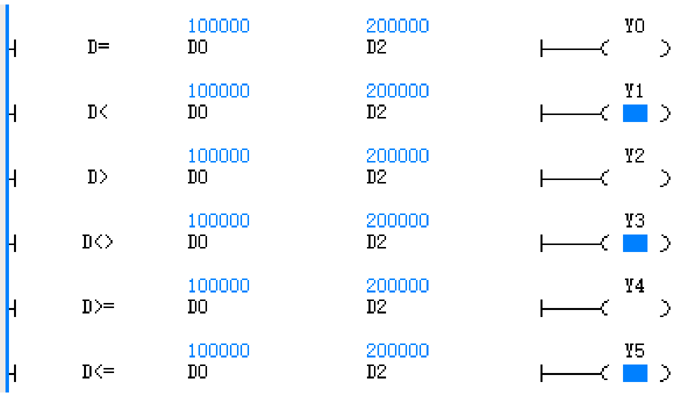
The command performs a 16-bit/32-bit BIN comparison between the contents of the S1 and S2 units and uses the comparison result to drive the subsequent segment operation.

Application Example

16-bit command:



32-bit command:



3.14.3 AND (=, <, >, <>, >=, <=): Commands for Integer/Long Integer AND Contact Comparison

Command table	AND* (S1) (S2)	Applicable model	TS600 series
16-Bit command	AND =: Integer AND contact comparison equal to		
32-Bit command	ANDD =: Long integer AND contact comparison equal to		
16-Bit command	AND <: Integer AND contact comparison less than		
32-Bit command	AND<: Long integer AND contact comparison less than		
16-Bit command	AND >: Integer AND contact comparison greater than		
32-Bit command	AND >: Long integer AND contact comparison greater than		
16-Bit command	AND <>: Integer AND contact comparison not equal to		
32-Bit command	ANDD <>: Long integer AND contact comparison not equal to		
16-Bit command	AND >=: Integer AND contact comparison greater than or equal to		
32-Bit command	ANDD >=: Long integer AND contact comparison greater than or equal to		
16-Bit command	AND <=: Integer AND contact comparison less than or equal to		
32-Bit command	ANDD <=: Long integer AND contact comparison less than or equal to		

Operand	Type	Bit			Word		Indexing	Constant
		X, Y, M, LM, T, C, S	Dx.y	Custom bit variable	D, R, V, Z, T, C	Custom word variable		
S1	INT/DINT	-	-	-	√[1]	√	√	√
S2	INT/DINT	-	-	-	√[1]	√	√	√

Remark:

[1] For 32-bit commands, the Z and T elements are not supported.

Operand Description

S1: Comparison parameter 1.

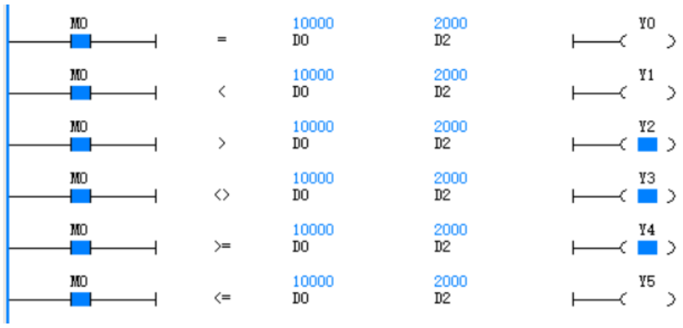
S2: Comparison parameter 2.

Function Description

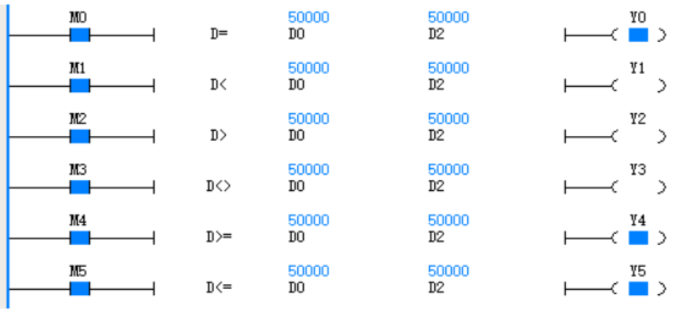
The command performs a 16-bit/32-bit BIN comparison between the contents of the S1 and S2 units and connects the comparison result in series with other nodes to drive the subsequent segment operation.

Application Example

16-bit command:



32-bit command:



3.14.4 OR (=, <, >, <>, >=, <=): Commands for Integer/Long Integer OR Contact Comparison

Command table	OR * (S1) (S2)				Applicable model	TS600 series		
16-Bit command	OR =: Integer OR contact comparison not equal to							
32-Bit command	ORD =: Long integer OR contact comparison not equal to							
16-Bit command	OR <: Integer OR contact comparison less than							
32-Bit command	ORD <: Long integer OR contact comparison less than							
16-Bit command	OR >: Integer OR contact comparison less than							
32-Bit command	ORD >: Long integer OR contact comparison less than							
16-Bit command	OR <>: Integer OR contact comparison not equal to							
32-Bit command	ORD <>: Long integer OR contact comparison not equal to							
16-Bit command	OR >=: Integer OR contact comparison greater than or equal to							
32-Bit command	ORD >=: Long integer OR contact comparison greater than or equal to							
16-Bit command	OR <=: Integer OR contact comparison less than or equal to							
32-Bit command	ORD <=: Long integer OR contact comparison less than or equal to							
Operand	Type	Bit			Word		Indexing	Constant
		X, Y, M, LM, T, C, S	Dx.y	Custom bit variable	D, R, V, Z, T, C	Custom word variable		
S1	INT/DINT	-	-	-	√[1]	√	√	√
S2	INT/DINT	-	-	-	√[1]	√	√	√

Remark:

[1] For 32-bit commands, the Z, C, and T elements are not supported.

Operand Description

S1: Comparison parameter 1.

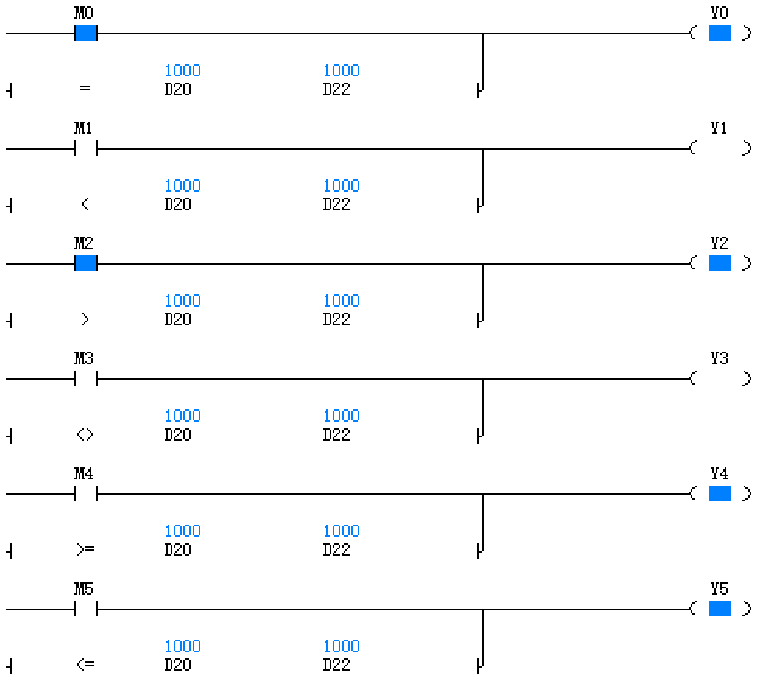
S2: Comparison parameter 2.

Function Description

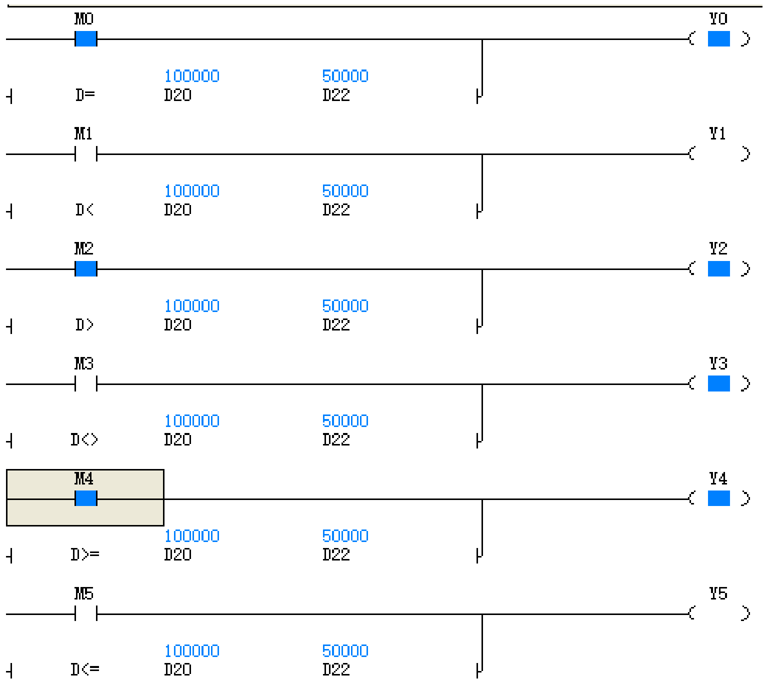
The command performs a 16-bit/32-bit BIN comparison between the contents of the S1 and S2 units and connects the comparison result in parallel with other nodes to drive the subsequent segment operation.

Application Example

16-bit command:



32-bit command:



3.14.5 LDR (=, <, >, <>, >=, <=): Commands for Floating-Point Number LD Contact Comparison

Command table		LDR (S1) (S2)			Applicable model	TS600 series		
32-Bit command		LDR =: Floating-point number LD contact comparison equal to						
32-Bit command		LDR <: Floating-point number LD contact comparison less than						
32-Bit command		LDR >: Floating-point number LD contact comparison greater than						
32-Bit command		LDR <>: Floating-point number LD contact comparison not equal to						
32-Bit command		LDR >=: Floating-point number LD contact comparison greater than or equal to						
32-Bit command		LDR <=: Floating-point number LD contact comparison less than or equal to						
Operand	Type	Bit			Word		Indexing	Constant
		X, Y, M, LM, T, C, S	Dx.y	Custom bit variable	D, R, V, Z, T, C	Custom word variable		
S1	REAL	-	-	-	√[1]	√	√	√
S2	REAL	-	-	-	√[1]	√	√	√

Remark:

[1] For 32-bit commands, the Z, C, and T elements are not supported.

Operand Description

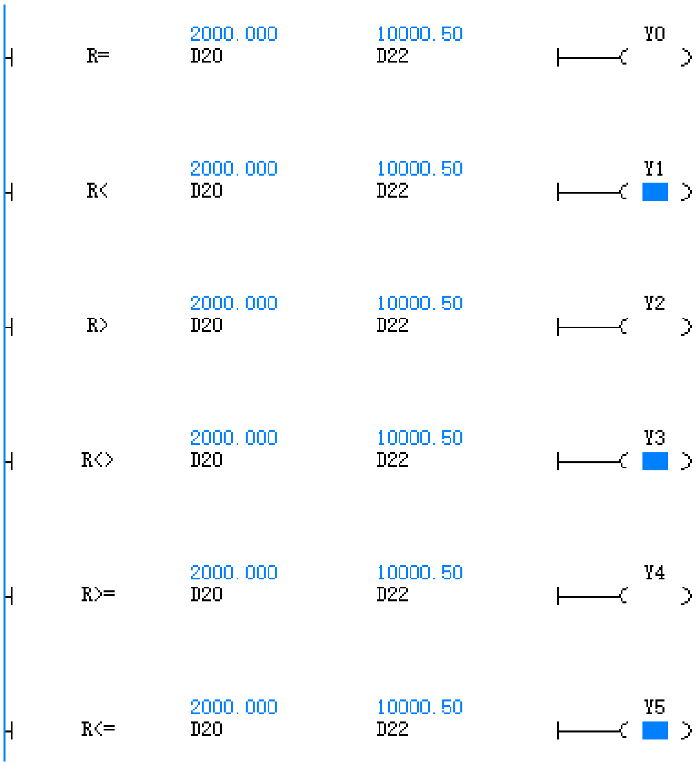
S1: Comparison parameter 1.

S2: Comparison parameter 2.

Function Description

The command performs a floating-point number comparison between the contents of the S1 and S2 units and uses the comparison result to drive the subsequent segment operation.

Application Example



3.14.6 ANDR (=, <, >, <>, >=, <=): Commands for Floating-Point Number AND Contact Comparison

Command table	ANDR (S1) (S2)				Applicable model	TS600 series		
32-Bit command	ANDR =: Floating-point number AND contact comparison equal to							
32-Bit command	ANDR <: Floating-point number AND contact comparison less than							
32-Bit command	ANDR >: Floating-point number AND contact comparison greater than							
32-Bit command	ANDR <>: Floating-point number AND contact comparison not equal to							
32-Bit command	ANDR >=: Floating-point number AND contact comparison greater than or equal to							
32-Bit command	ANDR <=: Floating-point number AND contact comparison less than or equal to							
Operand	Type	Bit			Word		Indexing	Constant
		X, Y, M, LM, T, C, S	Dx.y	Custom bit variable	D, R, V, Z, T, C	Custom word variable		
S1	REAL	-	-	-	√[1]	√	√	√
S2	REAL	-	-	-	√[1]	√	√	√

Remark:

[1] For 32-bit commands, the Z, C, and T elements are not supported.

Operand Description

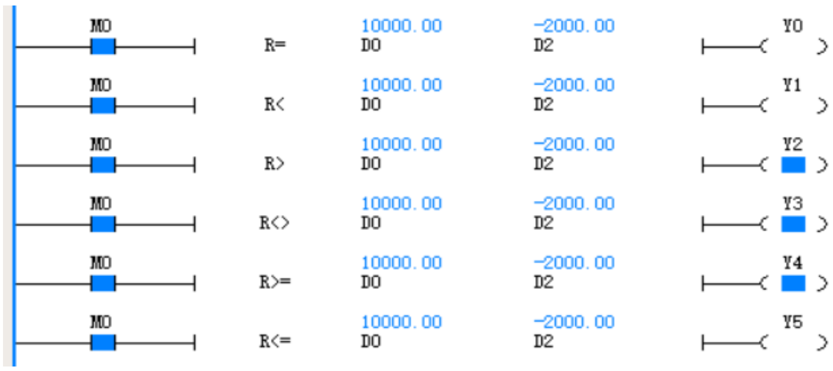
S1: Comparison parameter 1.

S2: Comparison parameter 2.

Function Description

The command performs a floating-point number comparison between the contents of the S1 and S2 units and connects the comparison result in series with other nodes to drive the subsequent segment operation.

Application Example



3.14.7 ORR (=, <, >, <>, >=, <=): Commands for Floating-Point Number OR Contact Comparison

Command table	ORR (S1) (S2)	Applicable model	TS600 series
32-Bit command	ORR =: Floating-point number OR contact comparison equal to		
32-Bit command	ORR <: Floating-point number OR contact comparison less than		
32-Bit command	ORR >: Floating-point number OR contact comparison greater than		
32-Bit command	ORR <>: Floating-point number OR contact comparison not equal to		
32-Bit command	ORR >=: Floating-point number OR contact comparison greater than or equal to		
32-Bit command	ORR <=: Floating-point number OR contact comparison less than or equal to		

Operand	Type	Bit			Word		Indexing	Constant
		X, Y, M, LM, T, C, S	Dx.y	Custom bit variable	D, R, V, Z, T, C	Custom word variable		
S1	REAL	-	-	-	√[1]	√	√	√
S2	REAL	-	-	-	√[1]	√	√	√

Remark:

[1] For 32-bit commands, the Z, C, and T elements are not supported.

Operand Description

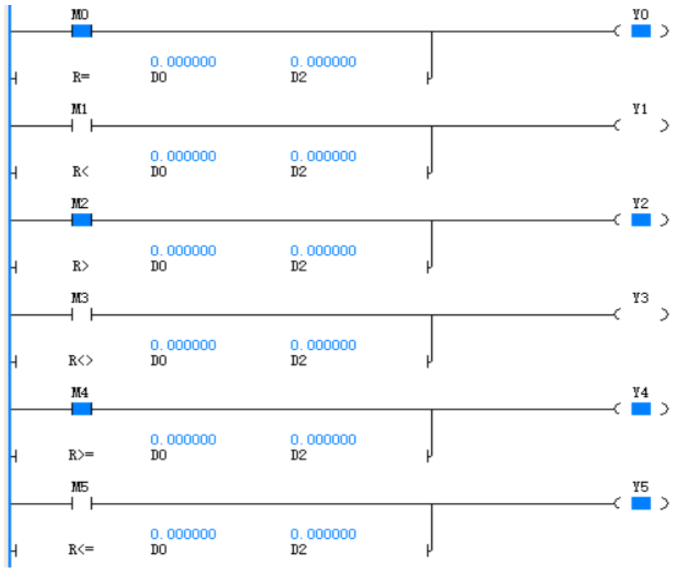
S1: Comparison parameter 1.

S2: Comparison parameter 2.

Function Description

The command performs a floating-point number comparison between the contents of the S1 and S2 units and connects the comparison result in parallel with other nodes to drive the subsequent segment operation.

Application Example



3.14.8 CMP: Integer Comparison Set

Command table	CMP (D) (S)				Applicable model	TS600 series		
16-Bit command	CMP: Integer comparison set							
32-Bit command	-							
Operand	Type	Bit			Word		Indexing	Constant
		X, Y, M, LM, T, C, S	Dx.y	Custom bit variable	D, R, V, Z, T, C	Custom word variable		
S1	INT	-	-	-	√	√	√	√
S2	INT	-	-	-	√	√	√	√
D	BOOL	√[1]	-	√	-	-	-	-

Remark:

[1] The Y, M, and S elements are supported.

Operand Description

S1: The number of the data or soft element becoming the comparison value.

S2: The number of the data or soft element becoming the comparison source.

D: The number of the starting element outputting the result.

Function Description

When the energy flow is valid, the command is executed to compare S1 with S2. Depending on the comparison result (less than, equal to, or greater than), the command sets one of (D), (D+1), and (D+2) to ON.

Application Example

3.14.9 LCMP: Long Integer Comparison Set

Command table	LCMP (S1) (S2) (D)			Applicable model	TS600 series		
16-Bit command	-						
32-Bit command	LCMP: Long integer comparison set						
Operand	Type	Bit			Word			
		X, Y, M, LM, T, C, S	Dx.y	Custom bit variable	D, R, V, Z, T, C	Custom word variable	Indexing	Constant
S1	DINT	-	-	-	√	√	√	√
S2	DINT	-	-	-	√	√	√	√
D	BOOL	√[1]	-	√	-	-	-	-

Remark:

[1] The Y, M, and S elements are supported.

Operand Description

S1: Comparison value 1.

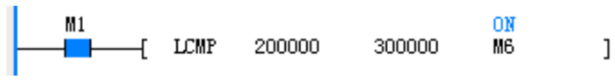
S2: Comparison value 2.

D: The number of the starting element outputting the result.

Function Description

When the energy flow is valid, the command is executed to compare S1 with S2. Depending on the comparison result (less than, equal to, or greater than), the command sets one of (D), (D+1), and (D+2) to ON.

Application Example



3.14.10 RCMP: Floating-Point Number Comparison Set

Command table	RCMP (D) (S)			Applicable model	TS600 series			
16-Bit command	-							
32-Bit command	RCMP: Floating-point number comparison set							
Operand	Type	Bit			Word			
		X, Y, M, LM, T, C, S	Dx.y	Custom bit variable	D, R, V, Z, T, C	Custom word variable	Indexing	Constant
S1	REAL	-	-	-	√[1]	√	√	√
S2	REAL	-	-	-	√[1]	√	√	√
D	BOOL	√[2]	-	√	-	-	-	-

Remark:

[1] The D and R elements are supported.

[2] The Y, M, and S elements are supported.

Operand Description

S1: Comparison value 1.

S2: Comparison value 2.

D: The number of the starting element outputting the result.

Function Description

When the energy flow is valid, the command is executed to compare S1 with S2. Depending on the comparison result (less than, equal to, or greater than), the command sets one of (D), (D+1), and (D+2) to ON.

Application Example

3.14.11 ZCP: Word/Doubleword Data Region Comparison Set

Command table		*ZCP (S1) (S2) (S) (D)			Applicable model	TS600 series		
16-Bit command		ZCP: Word data region comparison set						
32-Bit command		DZCP: Doubleword data region comparison set						
Operand	Type	Bit			Word		Indexing	Constant
		X, Y, M, LM, T, C, S	Dx.y	Custom bit variable	D, R, V, Z, T, C	Custom word variable		
S1	INT/DINT	-	-	-	√[1]	√	√	√
S2	INT/DINT	-	-	-	√[1]	√	√	√
S	INT/DINT	-	-	-	√[1]	√		√
D	BOOL	√[2]	-	√	-	√	-	-

Remark:

[1] The D and R elements are supported.

[2] The Y, M, and S elements are supported.

Operand Description

S1: The lower limit value of region comparison.

S2: The upper limit value of region comparison.

S: The comparison variable.

D: The comparison result.

Function Description

When the energy flow is valid, the command performs an algebraic comparison operation based on signed numbers, takes S1 and S2 as a region, takes the value of S at the position within the region as the result, and stores the result in the 3 consecutive bit variables with D as the starting address.

Application Example

		Element Name	Data Type	Display Fo:	Current Value
1	...	M0	BOOL	Binary	OFF
2	...	M1	BOOL	Binary	ON
3	...	M2	BOOL	Binary	OFF

In case of M10=ON, S=250 is between S1 and S2, so M1=ON is obtained.

		Element Name	Data Type	Display Fo	Current Value
1	...	M0	BOOL	Binary	OFF
2	...	M1	BOOL	Binary	ON
3	...	M2	BOOL	Binary	OFF

In case of M10=ON, S=250 is between S1 and S2, so M1=ON is obtained.

3.14.12 RZCP: Commands for Floating-Point Number Region Comparison Set

Command table		RZCP (S1) (S2) (S3) (D)			Applicable model	TS600 series		
16-Bit command		-						
32-Bit command		RZCP: Floating-point number region comparison set						
Operand	Type	Bit			Word		Indexing	Constant
		X, Y, M, LM, T, C, S	Dx.y	Custom bit variable	D, R, V, Z, T, C	Custom word variable		
S1	REAL	-	-	-	√[1]	√	√	√
S2	REAL	-	-	-	√[1]	√	√	√
S3	REAL	-	-	-	√[1]	√		√
D	BOOL	√[2]	-	√	-	√	-	-

Remark:

[1] The D and R elements are supported.

[2] The Y, M, and S elements are supported.

Operand Description

S1: The lower limit value of region comparison.

S2: The upper limit value of region comparison.

S3: The comparison variable.

D: The comparison result.

Function Description

When the energy flow is valid, the command performs an algebraic comparison operation based on signed numbers, takes S1 and S2 as a region, takes the value of S3 at the position within the region as the result, and stores the result in the 3 consecutive bit variables with D as the starting address.

Application Example

		Element Name	Data Type	Display Fo	Current Value
1	...	M0	BOOL	Binary	OFF
2	...	M1	BOOL	Binary	ON
3	...	M2	BOOL	Binary	OFF

In case of M10=ON, S3=250.0000 is between S1 and S2, so M1=ON is obtained.

3.15 Numerical Conversion Command

3.15.1 Command Table

Command Category	Name	Function
Numerical Conversion Command	DTI	Conversion from long integer to integer
	ITD	Conversion from integer to long integer
	*FLT	Conversion from Integer/long integer to floating-point number

Command Category	Name	Function
	*INT	Conversion from floating-point number to Integer/long integer
	*BCD	Conversion from word/doubleword data to 16-bit/32-bit BCD code
	*BIN	Conversion from 16-bit/32-bit BCD code to word/doubleword data
	*GRY	Conversion from word/doubleword to 16-bit/32-bit Gray code
	*GBIN	Conversion from 16-bit/32-bit Gray code to word/doubleword data
	SEG	Conversion from word data to 7-segment code
	ITA	Conversion from 16-bit hexadecimal number to ASCII code
	ATI	Conversion from ASCII code to 16-bit hexadecimal number
	LCNV	Engineering conversion
	RLCNV	Floating-point number engineering conversion
	DABIN	Conversion from decimal ASCII code to integer/long integer
	BINDA	Conversion from integer/long integer to decimal ASCII code

3.15.2 DTI: Commands for Conversion from Long Integer to Integer

Command table	DTI (S) (D)			Applicable model	TS600 series		
16-Bit command	-						
32-Bit command	DTI: Conversion from long integer to integer						
Operand / Type	Bit			Word		Indexing	Constant
	X, Y, M, LM, T, C, S	Dx.y	Custom bit variable	D, R, V, Z, T, C	Custom word variable		
S / DINT	-	-	-	√[1]	√	√	√
D / INT	-	-	-	√	√	√	-

Remark:

[1] The Z and T elements are not supported.

Operand Description

S: The source operand, which indicates a long integer data element.

D: The destination operand, which indicates an integer data element.

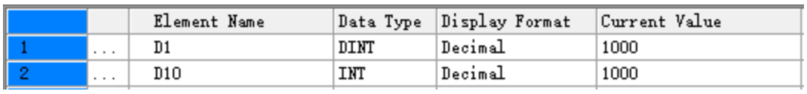
Function Description

When the command is driven, the 32-bit data of the S element is converted into the 16-bit data, and the result is stored in the D element.

Precautions

In case of S>32767 or S<-32768, the system reports an operand error, this conversion operation is not executed, and the data of the D element remains unchanged.

Application Example



		Element Name	Data Type	Display Format	Current Value
1	...	D1	DINT	Decimal	1000
2	...	D10	INT	Decimal	1000

In case of M500=ON, (D1, D2)=1000 is converted from a long integer to an integer, and the result is assigned to obtain D10: D10=1000.

3.15.3 ITD: Commands for Conversion from Integer to Long Integer

Command table	ITD (S) (D)	Applicable model	TS600 series
16-Bit command	-		
32-Bit command	ITD: Conversion from integer to long integer		

Operand	Type	Bit			Word		Indexing	Constant
		X, Y, M, LM, T, C, S	Dx.y	Custom bit variable	D, R, V, Z, T, C	Custom word variable		
S	INT	-	-	-	√	√	√	√
D	DINT	-	-	-	√[1]	√	√	-

Remark:

[1] The Z and T elements are not supported.

Operand Description

S: The source operand, which indicates an integer data element.

D: The destination operand, which indicates a long integer data element.

Function Description

When the command is driven, the 16-bit integer data of the S element is converted into the 32-bit data, and the result is stored in the D element.

Application Example

		Element Name	Data Type	Display Format	Current Value
1	...	D1	INT	Decimal	1000
2	...	D10	DINT	Decimal	1000

In case of M501=ON, D1=1000 is converted from an integer to a long integer, and the result is assigned to obtain (D10, D11)=1000.

3.15.4 FLT: Commands for Conversion from Integer/Long Integer to Floating-Point Number

Command table	*FLT (S) (D)			Applicable model	TS600 series			
16-Bit command	FLT: Conversion from integer to floating-point number							
32-Bit command	DFLT: Conversion from long integer to floating-point number							
Operand	Type	Bit			Word		Indexing	Constant
		X, Y, M, LM, T, C, S	Dx.y	Custom bit variable	D, R, V, Z, T, C	Custom word variable		
S	INT/DINT	-	-	-	√[1]	√	√	√
D	REAL	-	-	-	√[2]	√	√	-

Remark:

[1]For the 32-bit command DFLT, the Z and T elements are not supported.

[2]Only the D, V, and R elements are supported.

Operand Description

S: The source operand, which indicates an integer/long integer data element to be converted.

D: The destination operand, which indicates a storage word soft element of the floating-point number data.

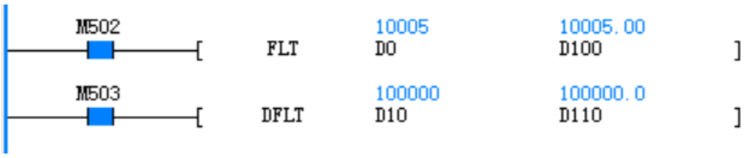
Function Description

1. When the command is driven, the 16-bit/32-bit integer data of the ***S*** element is converted into a floating-point number, and the result is stored in the D element.
2. The inverse command of this command is INT (which converts floating-point numbers into integer data).

Precautions

In case of S>32767 or S<-32768, the system reports an operand error, this conversion operation is not executed, and the data of the D element remains unchanged.

Application Example



In case of M502=ON, D0=10005 is converted from an integer to a floating-point number, and the result is assigned to (D1000, D101) to obtain (D100, D101)=10005.0.

In case of M503=ON, (D10, D11)=100000 is converted from a long integer to a floating-point number, and the result is assigned to (D110, D111) to obtain (D110, D111)=100000.0.

3.15.5 INT: Commands for Conversion from Floating-Point Number to Integer/Long Integer

Command table	*INT (S) (D)				Applicable model	TS600 series		
16-Bit command	INT: Conversion from floating-point number to integer							
32-Bit command	DINT: Conversion from floating-point number to long integer							
Operand	Type	Bit			Word		Indexing	Constant
		X, Y, M, LM, T, C, S	Dx.y	Custom bit variable	D, R, V, Z, T, C	Custom word variable		
S	REAL	-	-	-	√[1]	√	√	√
D	INT/DINT	-	-	-	√[2]	√	√	-

Remark:

[1]Only the D, V, and R elements are supported.

[2]For the 32-bit command DFLT, the Z and T elements are not supported.

Operand Description

S: The source operand, which indicates a floating-point number element to be converted.

D: The destination operand, which indicates an integer element stored after conversion

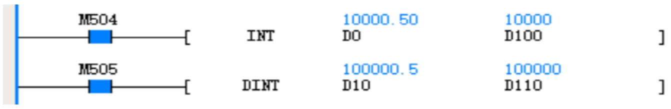
Function Description

1. When the command is driven, the 32-bit floating-point number data of the S element is rounded, the decimal part is discarded, and the 16-bit/32-bit data is stored in the D element.
2. When the conversion result is equal to 0, the zero flag bit (SM18) is set.
3. When decimals are truncated from the result, the borrow flag bit (SM19) is set.
4. When the result exceeds the range of the integer/long integer data, the borrow flag bit (SM20) is set.
5. Specifically, the range of the integer data is -32768~32767, and the range of the long integer data is -2147483648~2147483647.

Precautions

- INT command: In case of S＞32767 or S＜-32768, the system reports an illegal operand error, and the command is not executed.
- DINT command: In case of S＞2147483647 or S＜-2147483648, the system reports an illegal operand error, and the command is not executed.

Application Example



In case of M504=ON, (D0, D1)=10000.5 is converted from a floating-point number to an integer, and the result is assigned to D100 to obtain D100=10000.

In case of M505=ON, (D10, D11)=100000.5 is converted from a floating-point number to a long integer, and the result is assigned to (D110, D111) to obtain (D110, D111)=100000.

3.15.6 BCD: Commands for Conversion from Word/Doubleword Data to 16-Bit/32-Bit BCD Code

Command table		*BCD (S) (D)			Applicable model	TS600 series		
16-Bit command		BCD: Conversion from word data to 16-bit BCD code						
32-Bit command		DBCD: Conversion from doubleword data to 32-bit BCD code						
Operand	Type	Bit			Word		Indexing	Constant
		X, Y, M, LM, T, C, S	Dx.y	Custom bit variable	D, R, V, Z, T, C	Custom word variable		
S	WORD/ DWORD	-	-	-	√[1]	√	√	√
D	WORD/ DWORD	-	-	-	√[1]	√	√	

Remark:

[1] For the 32-bit command DBCD, the Z and T elements are not supported.

Operand Description

S: The source operand, which indicates the storage word soft element of the data to be converted.

D: The destination operand, which indicates the storage word soft element of the BCD code data after conversion.

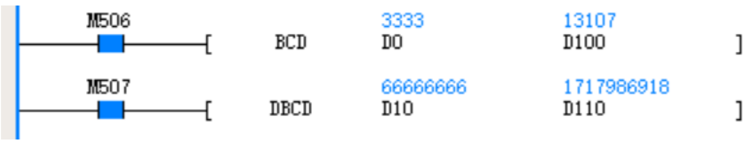
Function Description

When the command is driven, the 16-bit/32-bit data of S is converted into the BCD code, and the result is stored in the 16-bit/32-bit register of the D element. The command is usually used to format the data before displaying it.

Precautions

- BCD command: The value range of S is 0~9999. In case of S>9999 or S<0, the system reports an operand error.
- DBCD command: The value range of S is 0~99999999. In case of S>99999999 or S<0, the system reports an operand error.

Application Example



		Element Name	Data Type	Display Format	Current Value
1	...	D0	WORD	Decimal	3333
2	...	D100	WORD	Hexadecimal	16#3333
3	...		WORD	Decimal	
4	...	D10	DWORD	Decimal	66666666
5	...	D110	DWORD	Hexadecimal	16#66666666

In case of M506=ON, D0=0x0D05 (3333) is converted from an integer into a 16-bit BCD code, and the result is assigned to D100 to obtain D100=0x3333 (13107).

In case of M507=ON, (D10, D11)=0x3F940AA (66666666) is converted from a long integer to a 32-bit BCD code, and the result is assigned to (D110, D111) to obtain (D110, D111)=0x66666666 (1717986918).

3.15.7 BIN: Commands for Conversion from 16-Bit/32-Bit BCD Code to Word/Doubleword Data

Command table		*BIN (S) (D)			Applicable model	TS600 series		
16-Bit command		BIN: Conversion from 16-bit BCD code to word data						
32-Bit command		DBIN: Conversion from 32-bit BCD code to doubleword data						
Operand	Type	Bit			Word		Indexing	Constant
		X, Y, M, LM, T, C, S	Dx.y	Custom bit variable	D, R, V, Z, T, C	Custom word variable		
S	WORD/ DWORD	-	-	-	√[1]	√	√	√
D	WORD/ DWORD	-	-	-	√[1]	√	√	-

Remark:

[1] For the 32-bit command DBIN, the Z and T elements are not supported.

Operand Description

S: The source operand, which indicates the address of the BCD data or data storage word soft element.

D: The destination operand, which indicates the address of the data storage word soft element after conversion.

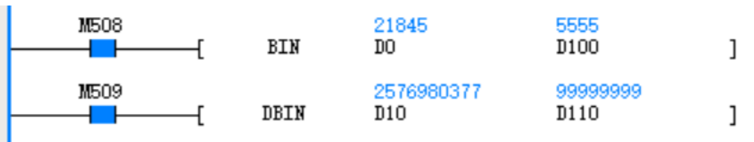
Function Description

When the command is driven, the 16-bit/32-bit BCD code data of the S element is converted into the integer data, and the result is stored in the D element. The command is usually used to process the data read from an external port into the integer data that can be directly used for calculation.

Precautions

When the data format of S does is not consistent with the BCD code format, the system reports an operand error, the conversion is not executed, and the content of D remains unchanged.

Application Example



		Element Name	Data Type	Display Format	Current Value
1	...	D0	WORD	Hexadecimal	16#5555
2	...	D100	WORD	Decimal	5555
3	...		WORD	Decimal	
4	...	D10	DWORD	Hexadecimal	16#99999999
5	...	D110	DWORD	Decimal	99999999

In case of M508=ON, D0=0x5555 (21845) is converted from a 16-bit BCD code into an integer, and the result is assigned to D100 to obtain D100=0x15B3 (5555).

In case of M509=ON, (D10, D11)=0x99999999 (2576980377) is converted from a 32-bit BCD code to a long integer, and the result is assigned to (D110, D111) to obtain (D110, D111)=0x5F5E0FF (99999999).

3.15.8 GRY: Commands for Conversion from Word/Doubleword Data to 16-Bit/32-Bit Gray Code

Command table	*GRY (S) (D)			Applicable model	TS600 series			
16-Bit command	GRY: Conversion from word data to Gray code							
32-Bit command	DGRY: Conversion from doubleword data to Gray code							
Operand	Type	Bit			Word		Indexing	Constant
		X, Y, M, LM, T, C, S	Dx.y	Custom bit variable	D, R, V, Z, T, C	Custom word variable		
S	WORD/ DWORD	-	-	-	√[1]	√	√	√
D	WORD/ DWORD	-	-	-	√[1]	√	√	-

Remark:

[1] For the 32-bit command DGRY, the Z and T elements are not supported.

Operand Description

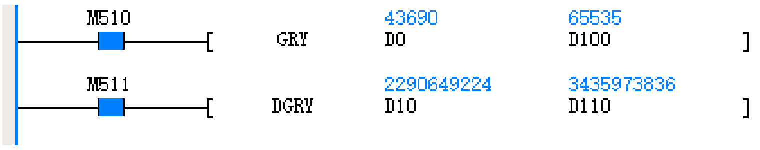
S: The source operand, which indicates the address of the data or data storage word soft element.

D: The destination operand, which indicates the address of the data storage word soft element for the gray code after conversion.

Function Description

When the command is driven, the 16-bit/32-bit data of S is converted into a Gray code, and the result is stored in the 16-bit/32-bit register of the D element.

Application Example



		Element Name	Data Type	Display Format	Current Value
1	...	D0	WORD	Hexadecimal	16#aaaa
2	...	D100	WORD	Hexadecimal	16#ffff
3	...		WORD	Decimal	
4	...	D10	DWORD	Hexadecimal	16#88888888
5	...	D110	DWORD	Hexadecimal	16#cccccccc

In case of M510=ON, D0=0xAAAA (43690) is converted from an integer into a 16-bit Gray code, and the result is assigned to D100 to obtain D100=0xFFFF (65535).

In case of M511=ON, (D10, D11)=0x88888888 (2290649224) is converted from a long integer to a 32-bit Gray code, and the result is assigned to (D110, D111) to obtain (D110, D111)=0xCCCCCCCC (3435973836).

3.15.9 GBIN: Commands for Conversion from 16-Bit/32-Bit Gray Code to Word/Doubleword Data

Command table	GBIN (S) (D)	Applicable model	TS600 series
16-Bit command	GBIN: Conversion from 16-bit Gray code to word data		
32-Bit command	DGBIN: Conversion from 32-bit Gray code to doubleword data		

Operand	Type	Bit			Word		Indexing	Constant
		X, Y, M, LM, T, C, S	Dx.y	Custom bit variable	D, R, V, Z, T, C	Custom word variable		
S	WORD/ DWORD	-	-	-	√[1]	√	√	√
D	WORD/ DWORD	-	-	-	√[1]	√	√	-

Remark:

[1] For the 32-bit command DGRY, the Z and T elements are not supported.

Operand Description

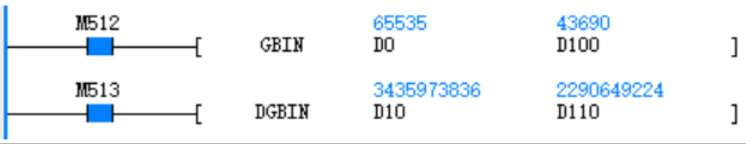
S: The source operand, which indicates the address of the Gray data or data storage word soft element.

D: The destination operand, which indicates the address of the data storage word soft element after conversion.

Function Description

When the command is driven, the 16-bit/32-bit Gray code data of the S element is converted into the integer data, and the result is stored in the 16-bit/32-bit register of the D element.

Application Example



		Element Name	Data Type	Display Format	Current Value
1	...	D0	WORD	Hexadecimal	16#ffff
2	...	D100	WORD	Hexadecimal	16#aaaa
3	...		WORD	Decimal	
4	...	D10	DWORD	Hexadecimal	16#cccccccc
5	...	D110	DWORD	Hexadecimal	16#88888888

In case of M512=ON, D0=0xFFFF (65535) is converted from a 16-bit Gray code into an integer, and the result is assigned to D100 to obtain D100=0xAAAA (43690).

In case of M513=ON, (D10, D11)=0xCCCCCCCC (3435973836) is converted from a 32-bit Gray code to a long integer, and the result is assigned to (D110, D111) to obtain (D110, D111)=0x88888888 (2290649224).

3.15.10 SEG: Commands for Conversion from Word Data to 7-Segment Code

Command table	SEG (S) (D)				Applicable model	TS600 series		
16-Bit command	SEG: Conversion from word data to 7-segment code							
32-Bit command	-							
Operand	Type	Bit			Word		Indexing	Constant
		X, Y, M, LM, T, C, S	Dx.y	Custom bit variable	D, R, V, Z, T, C	Custom word variable		
S	WORD	-	-	-	√	√	√	√
D	WORD	-	-	-	√	√	√	-

Operand Description

S: The source operand, which indicates the address of the data or data storage word soft element to be converted; S is ≤ 15.

D: The destination operand, which indicates the address of the data storage word soft element for the 7-segment code after conversion.

Function Description

When the command is driven, the 16-bit/32-bit data of the S element is converted into the 7-segment code data, and the result is stored in the 16-bit/32-bit register of the D element.

Precautions

In case of or S>15, the system reports an operand error, the conversion is not executed, and the content of D remains unchanged.

Application Example

		Element Name	Data Type	Display Format	Current Value
1	...	D0	WORD	Hexadecimal	16#f
2	...	D100	WORD	Hexadecimal	16#71

In case of M514=ON, D0=0x0F (15) is converted from an integer into a 7-segment code, and the result is assigned to D100 to obtain D100=0x71 (113).

3.15.11 ITA: Commands for Conversion from 16-Bit Hexadecimal Number to ASCII Code

Command table		ITA (S) (D) (n)			Applicable model	TS600 series		
16-Bit command		ITA: Conversion from 16-bit hexadecimal number to ASCII code						
32-Bit command		-						
Operand	Type	Bit			Word		Indexing	Constant
		X, Y, M, LM, T, C, S	Dx.y	Custom bit variable	D, R, V, Z, T, C	Custom word variable		
S	WORD/ Array*n	-	-	-	√	√	√	-
D	WORD/ Array*n	-	-	-	√	√	√	-
n	WORD	-	-	-	√	√	√	√

Operand Description

S: The source operand, which indicates the address of the hexadecimal data or data storage word soft element to be converted.

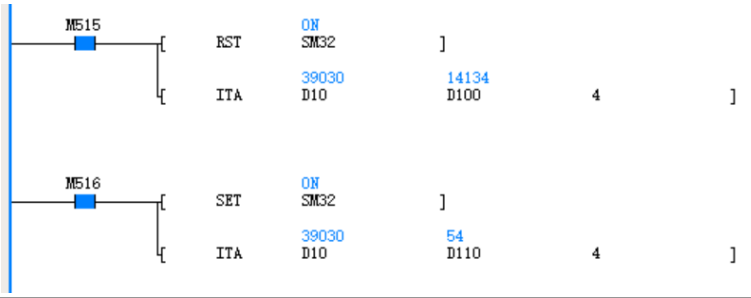
D: The destination operand, which indicates the address of the data storage word soft element for the ASCII code after conversion.

n: The number of ASCII code elements, which ranges between 1 and 256.

Function Description

1. When the command is driven, the hexadecimal numbers starting from the S element are converted into n ASCII codes, and the results are stored in the starting element of D.
2. After conversion, the results are stored in a small end format.
3. In case of SM32=OFF, each D element stores two ASCII code data in high and low bytes; in case of SM32=ON, each D element stores one ASCII code data in low bytes.

Application Example



		Element Name	Data Type	Display Format	Current Value
1	...	D10	WORD	Hexadecimal	16#9876
2	...		WORD	Hexadecimal	
3	...	D100	INT	Hexadecimal	16#3736
4	...	D101	INT	Hexadecimal	16#3938
5	...	D102	INT	Hexadecimal	16#0
6	...	D103	INT	Hexadecimal	16#0
7	...		WORD	Decimal	
8	...	D110	INT	Hexadecimal	16#36
9	...	D111	INT	Hexadecimal	16#37
10	...	D112	INT	Hexadecimal	16#38
11	...	D113	INT	Hexadecimal	16#39

In case of SM32=OFF and M515=OFF, the ITA conversion is executed, and the results are: D100=0x3736, and D101=0x3938.

In case of SM32=ON and M516=ON, the ITA conversion is executed, and the results are: D110=0x36, D111=0x37, D112=0x38, D113=0x39.

3.15.12 ATI: Commands for Conversion from ASCII Code to 16-Bit Hexadecimal Number

Command table		ATI (S) (D) (n)			Applicable model	TS600 series		
16-Bit command		ATI: Conversion from ASCII code to 16-bit hexadecimal number						
32-Bit command		-						
Operand	Type	Bit			Word		Indexing	Constant
		X, Y, M, LM, T, C, S	Dx.y	Custom bit variable	D, R, V, Z, T, C	Custom word variable		
S	WORD/ Array*n	-	-	-	√	√	√	√
D	WORD/ Array*n	-	-	-	√	√	√	-
n	WORD	-	-	-	√	√	√	√

Operand Description

S: The source operand, which indicates the address of the hexadecimal data or data storage word soft element to be converted. The value range of S is 0x30~0x39 or 0x41~0x46 (in case of FLG=OFF, both high and low bytes of S must meet the range).

D: The destination operand, which indicates the address of the data storage word soft element for the ASCII code after conversion.

n: The number of ASCII code elements, which ranges between 1 and 256.

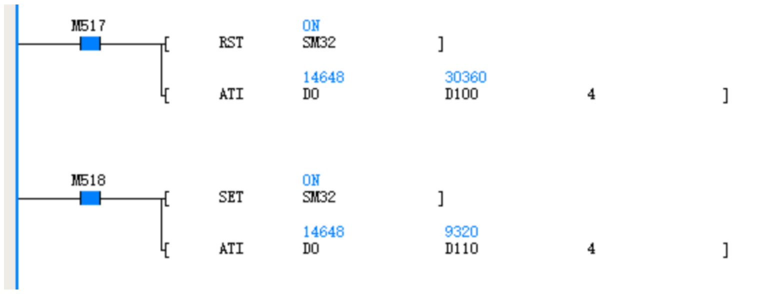
Function Description

1. When the command is driven, the hexadecimal numbers starting from the S element are converted into n ASCII codes, and the results are stored in the starting element of D.
2. After conversion, the results are stored in a small end format.
3. In case of SM32=OFF, each S element stores two ASCII code data in high and low bytes; in case of SM32=ON, each S element stores one ASCII code data in low bytes.

Precautions

- When the value of S is not within 0x30~0x39 or 0x41~0x46, the system reports an operand error, the conversion is not executed, and the content of D remains unchanged.
- When the value of n is not within 1~256, the system reports an operand error, the conversion is not executed, and the content of D remains unchanged.
- When S is a constant, in case of SM32=OFF and n＞2, the system reports an operand error, the conversion is not executed, and the content of D remains unchanged.
- When S is a constant, in case of SM32=ON and n＞1, the system reports an operand error, the conversion is not executed, and the content of D remains unchanged.

Application Example



		Element Name	Data Type	Display Fo:	Current Value
1	...	D0	WORD	Hexadecima	16#3938
2	...	D1	WORD	Hexadecima	16#3736
3	...	D2	WORD	Hexadecima	16#3534
4	...	D3	WORD	Hexadecima	16#3332
5	...		WORD	Decimal	
6	...	D100	WORD	Hexadecima	16#7698
7	...		WORD	Decimal	
8	...	D110	WORD	Hexadecima	16#2468

Source data: D0=0x3938, D1=0x3736, D2=0x3534, and D3=0x3332.

In case of SM32=OFF and M517=ON, the ATI conversion is executed, and the result is: D100=0x8967.

In case of SM32=ON and M518=ON, the ATI conversion is executed, and the result is: D110=0x8642.

3.15.13 LCNV: Engineering Conversion Commands

Command table	LCNV (S1) (S2) (D) (n)				Applicable model	TS600 series		
16-Bit command	LCNV: Engineering conversion							
32-Bit command	-							
Operand	Type	Bit			Word		Indexing	Constant
		X, Y, M, LM, T, C, S	Dx.y	Custom bit variable	D, R, V, Z, T, C	Custom word variable		
S1	INT/ Array*n	-	-	-	√[1]	√	-	-
S2	INT/ Array*4	-	-	-	√[1]	√	-	-
D	WORD/ DWORD	-	-	-	√[1]	√	-	-
n	WORD	-	-	-	√[1]	√	-	√

Remark:

[1] Only the D, R, and V soft elements are supported.

Operand Description

S1: The source operand, which indicates the starting address of the data storage word soft element to be converted.

S2: The destination operand, which indicates the starting address of the conversion table.

D: The destination operand, which indicates the starting address of the data storage word soft element after conversion.

n: The number of data to be converted, which ranges between 1 and 64.

Function Description

1. When using the analog input module to read external analog signals, you can use this command to convert the original analog reading value into the corresponding engineering reading value.
2. When using temperature or analog modules for temperature or analog measurement applications, if there are deviations between the temperatures or engineering readings measured by the PLC and the results measured by the standard thermometer or related standard instruments, you can use this command for the linear correction as the actual measurement correction.

Conversion Instructions

Fill in the conversion table with the following four parameters: low-point measurement value V_{ML}, high-point measurement value V_{MH}, and corresponding low-point standard value V_{SL} and high-point standard value V_{SH}; when performing linear conversion, the source data is calculated through the following formula to generate the corresponding target standard value, where S_n is the original input data and D_n is the conversion result data. See below for the conversion formula:

$$A = (V_{SL} - V_{SH}) / (V_{ML} - V_{MH}) * 10000$$

$$B = V_{SL} - (V_{ML} * A / 10000)$$

$$D_n = (S_n * A / 10000) + B$$

Precautions

- When the value of the conversion number n is not within 1~64, the system reports an operand error, the conversion is not executed, and the content of D remains unchanged.
- When the low-point measurement value is greater than the high-point measurement value, or the low-point standard value is greater than the high-point standard value, the system reports an upper-lower limit warning, the upper and lower limit values are exchanged, but this will not affect the continuous running of the program.
- If the output D is greater than 32767, the result is 32767; if it is less than -32768, the result is -32768.

Application Example

		Element Name	Data Type	Display Format	Current Value
1	...	D0	INT	Decimal	282
2	...	D1	INT	Decimal	3530
3	...	D2	INT	Decimal	1906
4	...	D3	INT	Decimal	0
5	...	D4	INT	Decimal	5000
6	...	D5	INT	Decimal	-115
7	...		WORD	Decimal	
8	...	D10	INT	Decimal	282
9	...	D11	INT	Decimal	3530
10	...	D12	INT	Decimal	260
11	...	D13	INT	Decimal	3650
12	...		WORD	Decimal	
13	...	D100	INT	Decimal	260
14	...	D101	INT	Decimal	3650
15	...	D102	INT	Decimal	1955
16	...	D103	INT	Decimal	-34
17	...	D104	INT	Decimal	5184
18	...	D105	INT	Decimal	-154

In case of M519=ON, the LCNV conversion is executed, and the following results are generated depending on the data storage methods:

D100=260

D101=3650

D102=1955

D103=-34

D104=5184

D105=-154

3.15.14 RLCNV: Floating-Point Engineering Conversion Commands

Command table		RLCNV (S1) (S2) (D) (n)			Applicable model	TS600 series		
16-Bit command		RLCNV: Floating-point engineering conversion						
32-Bit command		-						
Operand	Type	Bit			Word		Indexing	Constant
		X, Y, M, LM, T, C, S	Dx.y	Custom bit variable	D, R, V, Z, T, C	Custom word variable		
S1	REAL/ Array*n	-	-	-	√[1]	√	-	-
S2	REAL/ Array*4	-	-	-	√[1]	√	-	-
D	REAL/ Array*n	-	-	-	√[1]	√	-	-
n	WORD/ DWORD	-	-	-	√[1]	√	-	√

Remark:

[1] Only the D, R, and V soft elements are supported.

Operand Description

S1: The source operand, which indicates the starting address of the data storage word soft element to be converted.

S2: The destination operand, which indicates the starting address of the conversion table.

D: The destination operand, which indicates the starting address of the data storage word soft element after conversion.

n: The number of data to be converted, which ranges between 1 and 64.

Function Description

1. When using the analog input module to read external analog signals, you can use this command to convert the original analog reading value into the corresponding engineering reading value.
2. When using temperature or analog modules for temperature or analog measurement applications, if there are deviations between the temperatures or engineering readings measured by the PLC and the results measured by the standard thermometer or related standard instruments, you can use this command for the linear correction as the actual measurement correction.

Conversion Instructions

Fill in the conversion table with the following four parameters: low-point measurement value V_{ML}, high-point measurement value V_{MH}, and corresponding low-point standard value V_{SL} and high-point standard value V_{SH}; when performing linear conversion, the source data is calculated through the following formula to generate the corresponding target standard value, where S_n is the original input data and D_n is the conversion result data. See below for the conversion formula:

$$A = (V_{SL} - V_{SH})/(V_{ML} - V_{MH}) * 10000$$

$$B = V_{SL} - (V_{ML} * A/10000)$$

$$D_n = (S_n * A/10000) + B$$

Function Description

1. When the value of the conversion number n is not within 1~64, the system reports an operand error, the conversion is not executed, and the content of D remains unchanged.
2. When the low-point measurement value is greater than the high-point measurement value, or the low-point standard value is greater than the high-point standard value, the system reports an upper-lower limit warning, the upper and lower limit values are exchanged, but this will not affect the continuous running of the program.
3. If the output D is greater than 32767, the result is 32767; if it is less than -32768, the result is -32768.

Application Example

		Element Name	Data Type	Display Fo:	Current Value
1	...	D0	REAL	Decimal	282.0000
2	...	D2	REAL	Decimal	3530.000
3	...	D4	REAL	Decimal	1906.000
4	...	D6	REAL	Decimal	0.000000
5	...	D8	REAL	Decimal	5000.000
6	...	D10	REAL	Decimal	-115.0000
7	...		WORD	Decimal	
8	...	D20	REAL	Decimal	282.0000
9	...	D22	REAL	Decimal	3530.000
10	...	D24	REAL	Decimal	260.0000
11	...	D26	REAL	Decimal	3650.000
12	...		WORD	Decimal	
13	...	D100	REAL	Decimal	260.0000
14	...	D102	REAL	Decimal	3650.000
15	...	D104	REAL	Decimal	1955.000
16	...	D106	REAL	Decimal	-34.32880
17	...	D108	REAL	Decimal	5184.267
18	...	D110	REAL	Decimal	-154.3570

In case of M520=ON, the RLCNV conversion is executed, and the following results are generated depending on the data storage methods:

D200(D201)=260

D202(D203)=3650

D204(D205)=1955

D206(D207)=-34.3288

D208(D209)=5184.267

D210(D211)=-154.357

3.15.15 DABIN: Commands for Conversion from Decimal ASCII Code to Integer/Long Integer

<table>
<tr><td colspan="2">Command table</td><td colspan="3">*DABIN (S) (D)</td><td>Applicable model</td><td colspan="3">TS600 series</td></tr>
<tr><td colspan="2">16-Bit command</td><td colspan="7">DABIN: Conversion from decimal ASCII code to integer</td></tr>
<tr><td colspan="2">32-Bit command</td><td colspan="7">DDABIN: Conversion from decimal ASCII code to long integer</td></tr>
<tr><td rowspan="2">Operand</td><td rowspan="2">Type</td><td colspan="3">Bit</td><td colspan="2">Word</td><td rowspan="2">Indexing</td><td rowspan="2">Constant</td></tr>
<tr><td>X, Y, M, LM, T, C, S</td><td>Dx.y</td><td>Custom bit variable</td><td>D, R, V, Z, T, C</td><td>Custom word variable</td></tr>
<tr><td>S</td><td>WORD, Array*3/ DWORD, Array*6</td><td>-</td><td>-</td><td>-</td><td>√</td><td>√</td><td>√</td><td>-</td></tr>
<tr><td>D</td><td>INT/ DINT</td><td>-</td><td>-</td><td>-</td><td>√[1]</td><td>√</td><td>√</td><td>-</td></tr>
</table>

Remark:

[1] For 32-bit commands, the Z and T soft elements are not supported.

Operand Description

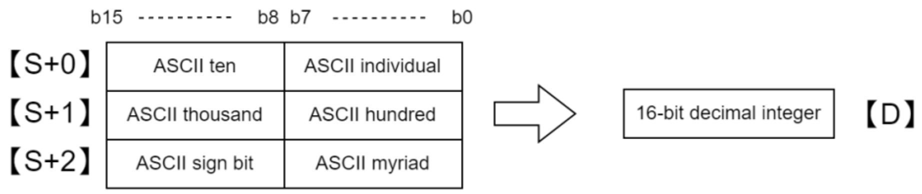
S: The source operand, which indicates the starting number of the soft element for the data (ASCII code) to be converted into an integer.

D: The destination operand, which indicates the number of the soft element storing the conversion result.

Function Description

1. For the 16-bit command:

The decimal ASCII code data (0x30~0x39) in S~S+2 is converted into the 16-bit integer data, and the result is stored in D, as shown in the figure below.



The value range of S~S+2 is 0~65535.

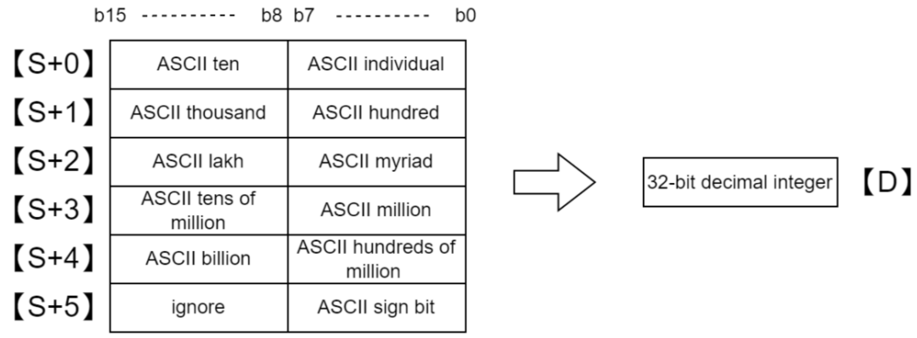
The symbol bit (lowest byte) of the data to be converted is "0x20 (space)" indicating a positive number, or "0x2D (-)" indicating a negative number.

The ASCII code of each bit ranges between 0x30 and 0x39.

If the ASCII code of each bit is "0x20 (space)" or "0x00 (NULL)", the result is treated as "0x30".

2. For the 32-bit command:

The decimal ASCII code data (30H~39H) in S~S+5 is converted into the 32-bit integer data, and the result is stored in D, as shown in the figure below.



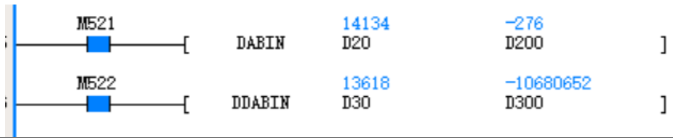
The value range of S~S+5 is 0~4,294,967,295.

The ASCII code of each bit ranges between "0x30" and "0x39".

If the ASCII code of each bit is "0x20 (space)" or "0x00 (NULL)", the result is treated as "0x30".

Application Example

1. When the symbol bit is a value other than "0x20 (space)" or "0x2D (-)", the system reports an ASCII conversion error, and the command is not executed.
2. When the ASCII code of the data bit is a value other than "0x30~0x39", "0x20 (space)", or "0x00 (NULL)", the system reports an ASCII conversion error, and the command is not executed.
3. When the converted data exceeds the value range of 16-bit or 32-bit signed integers, the system reports an operand error, and the command is not executed.
4. If the 16-bit command [S+2] or 32-bit command [S+5] exceeds the range value of the corresponding soft element, the system reports an error stating that the soft element address is out of range, and the command is not executed.



		Element Name	Data Type	Display Format	Current Value
1	...	D20	WORD	Hexadecimal	16#3736
2	...	D21	WORD	Hexadecimal	16#2032
3	...	D22	WORD	Hexadecimal	16#2d20
4	...		WORD	Decimal	
5	...	D200	INT	Decimal	-276
6	...		WORD	Decimal	
7	...	D30	WORD	Hexadecimal	16#3532
8	...	D31	WORD	Hexadecimal	16#3036
9	...	D32	WORD	Hexadecimal	16#3638
10	...	D33	WORD	Hexadecimal	16#3130
11	...	D34	WORD	Hexadecimal	16#3030
12	...	D35	WORD	Hexadecimal	16#2d
13	...		WORD	Decimal	
14	...	D300	DINT	Decimal	-10680652

3.15.16 BINDA: Commands for Conversion from Integer/Long Integer to Decimal ASCII Code

Command table	*BINDA (S) (D)	Applicable model	TS600 series
16-Bit command	BINDA: Conversion from integer to decimal ASCII code		
32-Bit command	DBINDA: Conversion from long integer to decimal ASCII code		

Operand	Type	Bit			Word		Indexing	Constant
		X, Y, M, LM, T, C, S	Dx.y	Custom bit variable	D, R, V, Z, T, C	Custom word variable		
S	WORD/ DWORD	-	-	-	√[1]	√	√	√
D	WORD, Array*3/ DWORD, Array*6	-	-	-	√	√	√	-

Remark:

[1] For 32-bit commands, the Z and T soft elements are not supported.

Operand Description

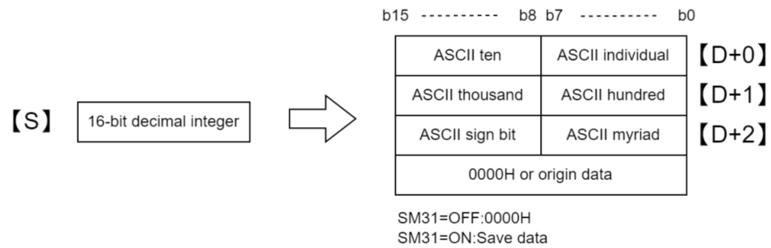
S: The source operand, which indicates the number of the soft element for the integer data to be converted into an ASCII code.

D: The destination operand, which indicates the starting number of the soft element storing the conversion result.

Function Description

1. For the 16-bit command:

The 16-bit integer data of S is converted into the ASCII code (0x30~0x39) in a decimal bitwise manner, the result is stored in the soft element starting from D.



The value range of S is 0~65535.

See below for the operation results:

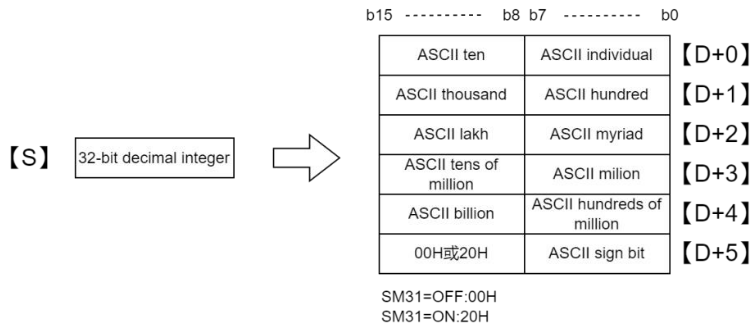
When the 16-bit data is a positive number, the symbol bit stores "0x20 (space)"; when the 16-bit data is a negative number, the symbol bit stores "0x2D (-)".

The ASCII code of each bit ranges between 0x30 and 0x39.

The value of [D+3] is determined according to the state of FLG, which is either ON or OFF.

2. For the 32-bit command:

The 32-bit integer data is converted into the ASCII code (0x30~0x39) in a decimal bitwise manner, the result is stored in the soft element starting from D.



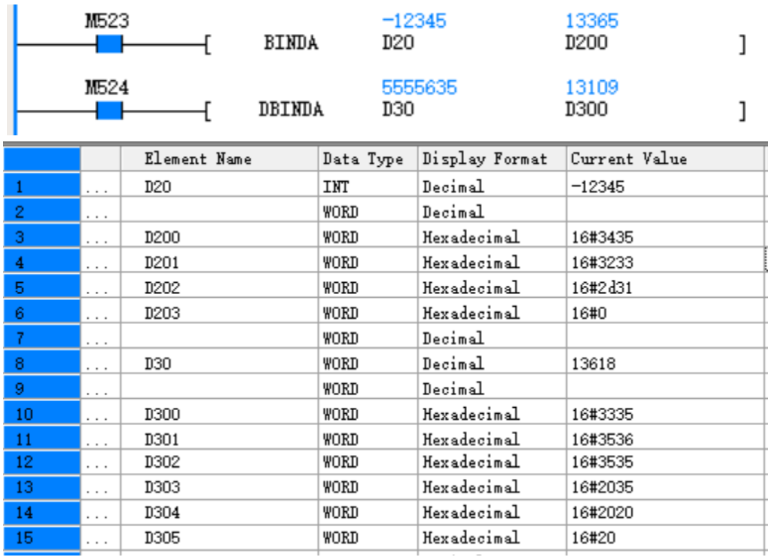
The value range of the 32-bit integer data [S+1, S] is 0~4,294,967,295.

See below for the operation results:

When the 32-bit data is a positive number, the symbol bit stores "0x20 (space)"; when the 16-bit data is a negative number, the symbol bit stores "0x2D (-)".

The value of [D+5] is determined according to the state of FLG, which is either ON or OFF.

Application Example



		Element Name	Data Type	Display Format	Current Value
1	...	D20	INT	Decimal	-12345
2	...		WORD	Decimal	
3	...	D200	WORD	Hexadecimal	16#3435
4	...	D201	WORD	Hexadecimal	16#3233
5	...	D202	WORD	Hexadecimal	16#2d31
6	...	D203	WORD	Hexadecimal	16#0
7	...		WORD	Decimal	
8	...	D30	WORD	Decimal	13618
9	...		WORD	Decimal	
10	...	D300	WORD	Hexadecimal	16#3335
11	...	D301	WORD	Hexadecimal	16#3536
12	...	D302	WORD	Hexadecimal	16#3535
13	...	D303	WORD	Hexadecimal	16#2035
14	...	D304	WORD	Hexadecimal	16#2020
15	...	D305	WORD	Hexadecimal	16#20

3.16 Batch Data Processing Command

3.16.1 Command Table

Command Category	Name	Function
Batch Data Processing Command	*BKADD	Addition operation of word/doubleword data block
	*BKSUB	Subtraction operation of word/doubleword data block
	*BKCMP=	Set word/doubleword data block comparison equal to
	*BKCMP>	Set word/doubleword data block comparison greater than
	*BKCMP<	Set word/doubleword data block comparison less than
	*BKCMP<>	Set word/doubleword data block comparison not equal to
	*BKCMP>=	Set word/doubleword data block comparison greater than or equal to
	*BKCMP<=	Set word/doubleword data block comparison less than or equal to
	BKITD	Batch conversion from integers to long integers
	BKDTI	Batch conversion from long integers to integers
	*BKFLT	Batch conversion from integers/long integers to floating-point numbers
	*BKINT	Batch conversion from floating-point numbers to integer/long integers
	BKWBIT	Assign word element to bit element combination
	BKBITW	Assign bit element combination to word element
	*BKAND	AND operation of word/doubleword data block
	*BKOR	OR operation of word/doubleword data block
	*BKXNR	XNOR operation of word/doubleword data block
	*BKXOR	XOR operation of word/doubleword data block
	*BKINV	Inversion operation of word/doubleword data block

3.16.2 BKADD: Commands for Addition Operation of Word/Doubleword Data Block

Command table		*BKADD (S1) (S2) (D) (n)			Applicable model	TS600 series		
16-Bit command		BKADD: Addition operation of word data block						
32-Bit command		DBKADD: Addition operation of doubleword data block						
Operand	Type	Bit			Word		Indexing	Constant
		X, Y, M, LM, T, C, S	Dx.y	Custom bit variable	D, R, V, Z, T, C	Custom word variable		
S1	INT/DINT, Array*n	-	-	-	√[1]	√	√	-
S2	INT/DINT, Array*n	-	-	-	√[1]	√	√	√
D	INT/DINT, Array*n	-	-	-	√[1]	√	√	-
n	WORD	-	-	-	√[2]	√	-	√

Remark:

[1] The Z element is not supported.

[2] The D, V, and R elements are supported.

Operand Description

S1: The source operand, which indicates the starting number of the soft element for the data executing the addition operation.

S2: The source operand, which indicates the constant executing the addition operation or the starting number of the soft element for the data executing the addition operation.

D: The destination operand, which indicates the starting number of the soft element storing the operation result.

n: The source operand, which indicates the number of data.

Precautions

When the operation result overflows, the carry flag bit is not set to ON.

Function Description

1. When the energy flow is valid, the command is executed, which performs the addition operation on the 16-bit data of the n points starting from S1 and the 16-bit data (BIN) of the n points starting from S2 and then stores the operation results in the n points starting from D.
2. You can directly specify a 16-bit constant in S2. When S2 is a constant, the command adds the 16-bit data of the n points starting from S1 to S2 sequentially and then stores the operation results in the n points starting from D.

Application Example

In case of M1=ON, the contents of the 5 units starting from D10 are added to the contents of the 5 units starting from D100 sequentially, and the results are stores in the 5 units starting from D1000. D1000=D10+D100, D1001=D11+D101, ……, D1004=D14+D104.

In case of M1=ON, the contents of the 3 units starting from D0 are added to the contents of the 3 units starting from D10 sequentially, and the results are stores in the 3 units starting from D100. D100=D0+D10, D101=D1+D11, D102=D12+D12.

3.16.3 BKSUB: Commands for Subtraction Operation of Word/Doubleword Data Block

Command table		*BKSUB (S1) (S2) (D) (n)			Applicable model	TS600 series		
16-Bit command		BKSUB: Subtraction operation of word data block						
32-Bit command		DBKSUB: Subtraction operation of doubleword data block						
Operand	Type	Bit			Word		Indexing	Constant
		X, Y, M, LM, T, C, S	Dx.y	Custom bit variable	D, R, V, Z, T, C	Custom word variable		
S1	INT/DINT, Array*n	-	-	-	√[1]	√	√	-
S2	INT/DINT, Array*n	-	-	-	√[1]	√	√	√
D	INT/DINT, Array*n	-	-	-	√[1]	√	√	-
n	WORD	-	-	-	√[2]	√	-	√

Remark:

[1] The Z element is not supported.

[2] The D, V, and R elements are supported.

Operand Description

S1: The source operand, which indicates the starting number of the soft element for the data executing the subtraction operation.

S2: The source operand, which indicates the constant executing the subtraction operation or the starting number of the soft element for the data executing the subtraction operation.

D: The destination operand, which indicates the starting number of the soft element storing the operation result.

n: The source operand, which indicates the number of data.

Function Description

1. When the energy flow is valid, the command is executed, which performs the subtraction operation on the 16-bit data of the n points starting from S1 and the 16-bit data (BIN) of the n points starting from S2 and then stores the operation results in the n points starting from D.
2. You can directly specify a 16-bit constant in S2. When S2 is a constant, the command subtracts S2 from the 16-bit data of the n points starting from S1 sequentially and then stores the operation results in the n points starting from D.

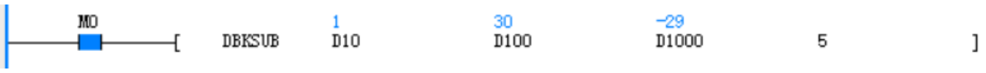
Precautions

When the operation result overflows, the carry flag bit is not set to ON.

Application Example

In case of M0=ON, the contents of the 5 units starting from D100 are subtracted from the contents of the 5 units starting from D10 sequentially, and the results are stores in the 5 units starting from D1000.

D1000=D10-D100, D1001=D11-D101, ······, D1004=D14-D104.



In case of M0=ON, the contents of the 5 doubleword units starting from D100 are subtracted from the contents of the 5 doubleword units starting from D10 sequentially, and the results are stores in the 5 units starting from D1000.

3.16.4 BKCMP =, >, <, <>, <=, >=: Commands for Word/Doubleword Data Block Comparison Set

Command table		*BKCMP (S1) (S2) (D) (n)			Applicable model	TS600 series		
16-Bit command		BKCMP =, >, <, <>, <=, >=: Word data block comparison set						
32-Bit command		DBKCMP =, >, <, <>, <=, >=: Doubleword data block comparison set						
Operand	Type	Bit			Word		Indexing	Constant
		X, Y, M, LM, T, C, S	Dx.y	Custom bit variable	D, R, V, Z, T, C	Custom word variable		
S1	INT/DINT, Array*n	-	-	-	√[1]	√	√	√
S2	INT/DINT, Array*n	-	-	-	√[1]	√	√	-
D	BOOL, Array*n	√	-	√	-	-	-	-
n	WORD	-	-	-	√[2]	√	-	√

Remark:

[1] The Z element is not supported.

[2] The D, V, and R elements are supported.

Operand Description

S1: The source operand, which indicates the starting number of the soft element comparing or storing the value data.

S2: The source operand, which indicates the starting number of the soft element storing the comparison source data.

D: The destination operand, which indicates the starting number of the soft element storing the comparison result.

n: The source operand, which indicates the number of data.

Precautions

When the operation result overflows, the carry flag bit is not set to ON.

Function Description

1. The command compares the 16-bit data of the n points starting from S1 and the 16-bit data (BIN) of the n points starting from S2 and then stores the operation results in the n points starting from D.
2. You can directly specify a 16-bit constant in S1. When S1 is a constant, the command compares S1 with the 16-bit data of the n points starting from S2 sequentially and then stores the operation results in the n points starting from D.
3. When all the comparison results in the n point starting from D are ON, the comparison set flag bit (SM35) of the data block is set.

Application Example

		Element Name	Data Type	Display Fo:	Current Value
1	...	D0	WORD	Decimal	10
2	...	D1	WORD	Decimal	11
3	...	D2	WORD	Decimal	12
4	...	D3	WORD	Decimal	13
5	...	D4	WORD	Decimal	14
6	...	D5	WORD	Decimal	15
7	...	D10	WORD	Decimal	10
8	...	D11	WORD	Decimal	11
9	...	D12	WORD	Decimal	12
10	...	D13	WORD	Decimal	13
11	...	D14	WORD	Decimal	14
12	...	D15	WORD	Decimal	15
13	...	M0	BOOL	Binary	ON
14	...	M1	BOOL	Binary	ON
15	...	M2	BOOL	Binary	ON
16	...	M3	BOOL	Binary	ON
17	...	M4	BOOL	Binary	ON

In case of M100=ON, the contents of the 5 units starting from D0 are compared with the contents of the 5 units starting from D10, and the results are stores in the 5 units starting from M0. In addition, when all the comparison results are ON, SM35 is set to ON.

3.16.5 BKITD: Commands for Batch Conversion from Integers to Long Integers

Command table		BKITD (S) (D) (n)			Applicable model	TS600 series		
16-Bit command		-						
32-Bit command		BKITD: Batch conversion from integers to long integers						
Operand	Type	Bit			Word		Indexing	Constant
		X, Y, M, LM, T, C, S	Dx.y	Custom bit variable	D, R, V, Z, T, C	Custom word variable		
S	INT, Array*n	-	-	-	√	√	√	√
D	DINT, Array*n	-	-	-	√[1]	√	√	-
n	WORD	-	-	-	√[1]	√	-	√

Remark:

[1] The D, V, and R elements are supported.

Operand Description

S: The source operand.

D: The destination operand.

n: The source operand, which indicates the number of conversions.

Function Description

When the energy flow is valid, the n integers (16-bit data) starting from the S element are converted into long integers (32-bit data), and the results are assigned to the n long integer (32-bit) elements starting from the D element.

Application Example

In case of M0=ON, D0=1000, D1=1000, and D2=1000 are converted from integers into long integers, and the results are assigned to (D10, D11), (D12, D13), and (D14, D15).

3.16.6 BKDTI: Commands for Batch Conversion from Long Integers to Integers

Command table	BKDTI (S) (D) (n)				Applicable model	TS600 series		
16-Bit command	-							
32-Bit command	BKDTI: Batch conversion from long integers to integers							
Operand	Type	Bit			Word		Indexing	Constant
		X, Y, M, LM, T, C, S	Dx.y	Custom bit variable	D, R, V, Z, T, C	Custom word variable		
S	DINT, Array*n	-	-	-	√[1]	√	√	√
D	INT, Array*n	-	-	-	√	√	√	-
n	WORD	-	-	-	√[1]	√	-	√

Remark:

[1] The D, V, and R elements are supported.

Operand Description

S: The source operand.

D: The destination operand.

n: The source operand, which indicates the number of conversions.

Function Description

When the energy flow is valid, the n long integers (32-bit data) starting from S are converted into integers (16-bit data), and the results are assigned to the n integer (16-bit) elements starting from D.

Precautions

When the n long integers starting from S are greater than 32767 or the n long integers starting from S are less than -32768, the system reports an operand error, this conversion operation is not executed, and the data of the D element remains unchanged.

Application Example

In case of M0=ON, (D0, D1)=1000, (D2, D3)=1000, and (D4, D5)=1000 are converted from long integers into integers, and the results are assigned to D10, D11, and D12.

3.16.7 BKFLT: Commands for Batch Conversion from Integers/Long Integers to Floating-Point Numbers

Command table	*BKFLT (S) (D) (n)				Applicable model	TS600 series		
16-Bit command	BKFLT: Batch conversion from integers/long integers to floating-point numbers							
32-Bit command	-							
Operand	Type	Bit			Word		Indexing	Constant
		X, Y, M, LM, T, C, S	Dx.y	Custom bit variable	D, R, V, Z, T, C	Custom word variable		
S	INT, Array*n	-	-	-	√	√	√	√
D	REAL, Array*n	-	-	-	√[1]	√	√	-
n	WORD	-	-	-	√[1]	√	-	√

Remark:

[1] The D, V, and R elements are supported.

Operand Description

S: The source operand.

D: The destination operand.

n: The source operand, which indicates the number of conversions.

Function Description

When the energy flow is valid, the n long integers (16-bit data) starting from S are converted into floating-point numbers (32-bit data), and the results are assigned to the n 32-bit data starting from D.

Application Example

In case of M0=ON, D0=10000 and D1=10000 are converted from integers into floating-point numbers, and the results are assigned to obtain (D10, D11)=10000.00 and (D12, D13)=10000.00.

3.16.8 BKINT: Batch Conversion from Floating-Point Numbers to Integers/Long integers

Command table	BKINT (S) (D) (n)			Applicable model	TS600 series		
16-Bit command	BKINT: Batch conversion from floating-point numbers to integers						
32-Bit command							

Operand	Type	Bit			Word		Indexing	Constant
		X, Y, M, LM, T, C, S	Dx.y	Custom bit variable	D, R, V, Z, T, C	Custom word variable		
S	REAL, Array*n	-	-	-	√[1]	√	√	√
D	INT, Array*n	-	-	-	√	√	√	-
n	WORD	-	-	-	√[1]	√	-	√

Remark:

[1] The D, V, and R elements are supported.

Operand Description

S: The source operand.

D: The destination operand.

n: The source operand, which indicates the number of conversions.

Function Description

1. When the energy flow is valid, the n floating-point numbers (32-bit data) starting from S are converted into integers (16-bit data), and the results are assigned to the n 16-bit data starting from D.
2. This command affects the zero flag bits and borrow flag bits. When the conversion results are 0, the zero flag bits are set. When decimals are truncated from the results, the borrow flag bits are set. When the results exceed the range of the long integer data, the carry (overflow) flag bits are set.

Precautions

- BKINT command: When the n integers starting from S are greater than 32767 or the n integers starting from S are less than -32768, the system reports an illegal operand error, and the command is not executed.
- DBKINT command: When the n long integers starting from S are greater than 2147483647 or the n long integers starting from S are less than -2147483648, the system reports an illegal operand error, and the command is not executed.

Application Example

In case of M0=ON, (D0, D1)=10000.5 and (D2, D3)=10000.5 are converted from floating-point numbers into integers, and the results are assigned to obtain D10=10000 and D11=10000.

3.16.9 BKWBIT: Commands to Assign Word Element to Bit Element Combination

Command table		BKWBIT (S) (D) (n)			Applicable model	TS600 series		
16-Bit command		BKWBIT: Assign word element to bit element combination						
32-Bit command		-						
Operand	Type	Bit			Word		Indexing	Constant
		X, Y, M, LM, T, C, S	Dx.y	Custom bit variable	D, R, V, Z, T, C	Custom word variable		
S	INT, Array*n	-	-	-	√[1]	√	√	√
D	BOOL, Array*n	√[2]	√	√	-	-	-	-
n	WORD	-	-	-	√[3]	√	-	√

Remark:

[1] The Z element is not supported.

[2] The X element is not supported.

[3] The D, V, and R elements are supported.

Operand Description

S: The source data, or the combination value to be assigned to the bit element.

D: The bit element, or the starting bit element.

n: The number of bit elements, or the quantity of bit elements.

Function Description

The command converts the binary numbers in S into the bit state and assigns the results to the n bits starting from D.

Application Example

When M0=ON, D0=2#1111 is assigned to M70, M71, M72, and M73, which therefore all become ON.

3.16.10 BKBITW: Commands to Assign Bit Element Combination to Word Element

Command table		BKWBIT (S) (D) (n)			Applicable model	TS600 series		
16-Bit command		BKWBIT: Assign bit element combination to word element						
32-Bit command		-						
Operand	Type	Bit			Word		Indexing	Constant
		X, Y, M, LM, T, C, S	Dx.y	Custom bit variable	D, R, V, Z, T, C	Custom word variable		
S	BOOL, Array*n	√	√	√	-	-	√	-
D	INT, Array*n	-	-	-	√[1]	√	√	√
n	WORD	-	-	-	√[2]	√	-	√

Remark:

[1] The Z element is not supported.

[2] The D, V, and R elements are supported.

Operand Description

S: The bit element, or the starting bit element.

D: The destination operand, which indicates the value of the bit element combination.

n: The number of bit elements, or the quantity of bit elements.

Function Description

The command converts the n bits starting from S into binary numbers and assigns the results to D.

Application Example

In case of M0=ON, M0, M1, M2, and M3 are all ON, and D0=15 is obtained.

3.16.11 BKAND: Commands for AND Operation of Word/Doubleword Data Block

Command table		*BKAND (S1) (S2) (D) (n)			Applicable model	TS600 series		
16-Bit command		BKAND: AND operation of batch word data blocks						
32-Bit command		DBKAND: AND operation of batch doubleword data blocks						
Operand	Type	Bit			Word		Indexing	Constant
		X, Y, M, LM, T, C, S	Dx.y	Custom bit variable	D, R, V, Z, T, C	Custom word variable		
S1	WORD / DWORD, Array*n	-	-	-	√[1]	√	√	√
S2	WORD / DWORD, Array*n	-	-	-	√[1]	√	√	√
D	WORD / DWORD, Array*n	-	-	-	√[1]	√	√	-
n	WORD	-	-	-	√[1]	√	-	√

Remark:

[1] The D, V, and R elements are supported.

Operand Description

S1: The source operand, which indicates the address of the data or data storage word soft element that participates in the AND operation.

S2: The source operand, which indicates the address of the data or data storage word soft element that participates in the AND operation.

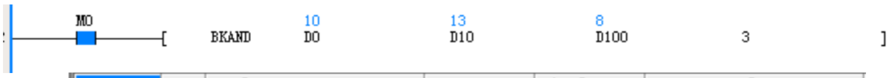
D: The destination operand, which indicates the address of the data storage word soft element of the operation result.

S3: The number of bit elements, or the quantity of comparisons.

Function Description

1. 16-bit command: When the energy flow is valid, the command is executed, which performs the bitwise logical AND operation on the n 16-bit data starting from S1 and the corresponding n 16-bit data (BIN) starting from S2 and then stores the operation results in the n 16-bit data starting from D.
2. 32-bit command: When the energy flow is valid, the command is executed, which performs the bitwise logical AND operation on the n 32-bit data starting from S1 and the corresponding n 32-bit data (BIN) starting from S2 and then stores the operation results in the n 32-bit data starting from D.
3. The rule of logical AND operation: If any data is 0, the result is 0. For example: 1・1=1 1・0=0 0・1=0 0・0=0.

Application Example



		Element Name	Data Type	Display Fo:	Current Value
1	...	D0	WORD	Decimal	10
2	...	D1	WORD	Decimal	11
3	...	D2	WORD	Decimal	12
4	...		WORD	Decimal	
5	...	D10	WORD	Decimal	13
6	...	D11	WORD	Decimal	14
7	...	D12	WORD	Decimal	15
8	...		WORD	Decimal	
9	...	D100	WORD	Decimal	8
10	...	D101	WORD	Decimal	10
11	...	D102	WORD	Decimal	12

In case of M0=ON, the command performs the bitwise logical AND operation on D0, D1, and D2 and D10, D11, and D12 and stores the results to D100, D101, and D102.

		Element Name	Data Type	Display Fo:	Current Value
1	...	D0	DINT	Decimal	100
2	...	D2	DINT	Decimal	110
3	...	D4	DINT	Decimal	120
4	...		WORD	Decimal	
5	...	D10	DINT	Decimal	130
6	...	D12	DINT	Decimal	140
7	...	D14	DINT	Decimal	150
8	...		WORD	Decimal	
9	...	D100	DINT	Decimal	0
10	...	D102	DINT	Decimal	12
11	...	D104	DINT	Decimal	16

In case of M0=ON, the command performs the bitwise logical AND operation on (D0, D1), (D2, D3), and (D4, D5) and (D10, D11), (D12, D13), and (D14, D15) and stores the results to (D100, D101), (D102, D103), and (D104, D105).

3.16.12 BKOR: Commands for OR Operation of Word/Doubleword Data Block

Command table		*BKOR (S1) (S2) (D) (n)			Applicable model	TS600 series		
16-Bit command		BKOR: OR operation of batch word data blocks						
32-Bit command		DBKOR: OR operation of batch doubleword data blocks						
Operand	Type	Bit			Word		Indexing	Constant
		X, Y, M, LM, T, C, S	Dx.y	Custom bit variable	D, R, V, Z, T, C	Custom word variable		
S1	WORD / DWORD, Array*n	-	-	-	√[1]	√	√	√
S2	WORD / DWORD, Array*n	-	-	-	√[1]	√	√	√
D	WORD / DWORD, Array*n	-	-	-	√[1]	√	√	-
n	WORD	-	-	-	√[1]	√	-	√

Remark:

[1] The D, V, and R elements are supported.

Operand Description

S1: The source operand, which indicates the address of the data or data storage word soft element that participates in the OR operation.

S2: The source operand, which indicates the address of the data or data storage word soft element that participates in the OR operation.

D: The destination operand, which indicates the address of the data storage word soft element of the operation result.

n: The number of bit elements, or the quantity of comparisons.

Function Description

1. 16-bit command: When the energy flow is valid, the command is executed, which performs the bitwise logical OR operation on the n 16-bit data starting from S1 and the corresponding n 16-bit data (BIN) starting from S2 and then stores the operation results in the n 16-bit data starting from D.
2. 32-bit command: When the energy flow is valid, the command is executed, which performs the bitwise logical OR operation on the n 32-bit data starting from S1 and the corresponding n 32-bit data (BIN) starting from S2 and then stores the operation results in the n 32-bit data starting from D.
3. The rule of logical OR operation: If any data is 1, the result is 1. For example: 1+1=1 1+0=1 0+1=1 0+0=0.

Application Example

		Element Name	Data Type	Display Fo:	Current Value
1	...	D0	WORD	Decimal	10
2	...	D1	WORD	Decimal	11
3	...	D2	WORD	Decimal	12
4	...		WORD	Decimal	
5	...		WORD	Decimal	
6	...	D10	WORD	Decimal	13
7	...	D11	WORD	Decimal	14
8	...	D12	WORD	Decimal	15
9	...		WORD	Decimal	
10	...	D100	WORD	Decimal	15
11	...	D101	WORD	Decimal	15
12	...	D102	WORD	Decimal	15

In case of M1=ON, the command performs the bitwise logical OR operation on D0, D1, and D2 and D10, D11, and D12 and stores the results to D100, D101, and D102.

		Element Name	Data Type	Display Fo:	Current Value
1	...	D0	DINT	Decimal	100
2	...	D2	DINT	Decimal	110
3	...	D4	DINT	Decimal	120
4	...		WORD	Decimal	
5	...	D10	DINT	Decimal	130
6	...	D12	DINT	Decimal	140
7	...	D14	DINT	Decimal	150
8	...		WORD	Decimal	
9	...	D100	DINT	Decimal	230
10	...	D102	DINT	Decimal	238
11	...	D104	DINT	Decimal	254

In case of M1=ON, the command performs the bitwise logical OR operation on (D0, D1), (D2, D3), and (D4, D5) and (D10, D11), (D12, D13), and (D14, D15) and stores the results to (D100, D101), (D102, D103), and (D104, D105).

3.16.13 BKXNR: Commands for XNOR Operation of Word/Doubleword Data Block

Command table	*BKXNR (S1) (S2) (D) (n)	Applicable model	TS600 series
16-Bit command	BKXNR: XNOR operation of batch word data blocks		
32-Bit command	DBKXNR: XNOR operation of batch doubleword data blocks		

Operand	Type	Bit			Word		Indexing	Constant
		X, Y, M, LM, T, C, S	Dx.y	Custom bit variable	D, R, V, Z, T, C	Custom word variable		
S1	WORD / DWORD, Array*n	-	-	-	√[1]	√	√	√
S2	WORD / DWORD, Array*n	-	-	-	√[1]	√	√	√
D	WORD / DWORD, Array*n	-	-	-	√[1]	√	√	-
n	WORD	-	-	-	√[1]	√	-	√

Remark:

[1] The D, V, and R elements are supported.

Operand Description

S1: The source operand, which indicates the address of the data or data storage word soft element that participates in the XNOR operation.

S2: The source operand, which indicates the address of the data or data storage word soft element that participates in the XNOR operation.

D: The destination operand, which indicates the address of the data storage word soft element of the operation result.

n: The number of bit elements, or the quantity of comparisons.

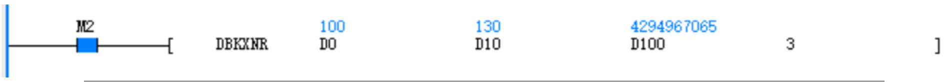
Function Description

1. 16-bit command: When the energy flow is valid, the command is executed, which performs the bitwise logical XNOR operation on the n 16-bit data starting from S1 and the corresponding n 16-bit data (BIN) starting from S2 and then stores the operation results in the n 16-bit data starting from D.
2. 32-bit command: When the energy flow is valid, the command is executed, which performs the bitwise logical XNOR operation on the n 32-bit data starting from S1 and the corresponding n 32-bit data (BIN) starting from S2 and then stores the operation results in the n 32-bit data starting from D.
3. The rules of logical XNOR operation: If two data are the same, the result is 0; if two data are different, the result is 0. For example: 1⊙1=1 1⊙0=0 0⊙1=0 0⊙0=1.

Application Example

		Element Name	Data Type	Display Fo:	Current Value
1	...	D0	WORD	Decimal	10
2	...	D1	WORD	Decimal	11
3	...	D2	WORD	Decimal	12
4	...		WORD	Decimal	
5	...		WORD	Decimal	
6	...	D10	WORD	Decimal	13
7	...	D11	WORD	Decimal	14
8	...	D12	WORD	Decimal	15
9	...		WORD	Decimal	
10	...	D100	WORD	Decimal	65528
11	...	D101	WORD	Decimal	65530
12	...	D102	WORD	Decimal	65532

In case of M2=ON, the command performs the bitwise logical AND operation on D0, D1, and D2 and D10, D11, and D12 and stores the results to D100, D101, and D102.



		Element Name	Data Type	Display Fo:	Current Value
1	...	D0	DINT	Decimal	100
2	...	D2	DINT	Decimal	110
3	...	D4	DINT	Decimal	120
4	...		WORD	Decimal	
5	...	D10	DINT	Decimal	130
6	...	D12	DINT	Decimal	140
7	...	D14	DINT	Decimal	150
8	...		WORD	Decimal	
9	...	D100	DINT	Hexadecima	16#ffffff19
10	...	D102	DINT	Hexadecima	16#ffffff1d
11	...	D104	DINT	Hexadecima	16#ffffff11

In case of M2=ON, the command performs the bitwise logical AND operation on (D0, D1), (D2, D3), and (D4, D5) and (D10, D11), (D12, D13), and (D14, D15) and stores the results to (D100, D101), (D102, D103), and (D104, D105).

3.16.14 BKXOR: Commands for XOR Operation of Word/Doubleword Data Block

Command table		*BKXOR (S1) (S2) (D) (n)			Applicable model	TS600 series		
16-Bit command		BKXOR: XOR operation of batch word data blocks						
32-Bit command		DBKXOR: XOR operation of batch doubleword data blocks						
Operand	Type	Bit			Word		Indexing	Constant
		X, Y, M, LM, T, C, S	Dx.y	Custom bit variable	D, R, V, Z, T, C	Custom word variable		
S1	WORD / DWORD, Array*n	-	-	-	√[1]	√	√	√
S2	WORD / DWORD, Array*n	-	-	-	√[1]	√	√	√
D	WORD / DWORD, Array*n	-	-	-	√[1]	√	√	-
n	WORD	-	-	-	√[1]	√	-	√

Remark:

[1] The D, V, and R elements are supported.

Operand Description

S1: The source operand, which indicates the address of the data or data storage word soft element that participates in the XOR operation.

S2: The source operand, which indicates the address of the data or data storage word soft element that participates in the XOR operation.

D: The destination operand, which indicates the address of the data storage word soft element of the operation result.

n: The number of bit elements, or the quantity of comparisons.

Function Description

1. 16-bit command: When the energy flow is valid, the command is executed, which performs the bitwise logical XOR operation on the n 16-bit data starting from S1 and the corresponding n 16-bit data (BIN) starting from S2 and then stores the operation results in the n 16-bit data starting from D.
2. 32-bit command: When the energy flow is valid, the command is executed, which performs the bitwise logical XOR operation on the n 32-bit data starting from S1 and the corresponding n 32-bit data (BIN) starting from S2 and then stores the operation results in the n 32-bit data starting from D.
3. The rules of logical XOR operation: If two data are the same, the result is 0; if two data are different, the result is 1. For example: 1^1=0 1^0=1 0^1=1 0^0=0.

Application Example

		Element Name	Data Type	Display Fo:	Current Value
1	...	D0	WORD	Decimal	10
2	...	D1	WORD	Decimal	11
3	...	D2	WORD	Decimal	12
4	...		WORD	Decimal	
5	...		WORD	Decimal	
6	...	D10	WORD	Decimal	13
7	...	D11	WORD	Decimal	14
8	...	D12	WORD	Decimal	15
9	...		WORD	Decimal	
10	...	D100	WORD	Decimal	7
11	...	D101	WORD	Decimal	5
12	...	D102	WORD	Decimal	3

In case of M3=ON, the command performs the bitwise logical XOR operation on D0, D1, and D2 and D10, D11, and D12 and stores the results to D100, D101, and D102.

		Element Name	Data Type	Display Fo:	Current Value
1	...	D0	DINT	Decimal	100
2	...	D2	DINT	Decimal	110
3	...	D4	DINT	Decimal	120
4	...		WORD	Decimal	
5	...	D10	DINT	Decimal	130
6	...	D12	DINT	Decimal	140
7	...	D14	DINT	Decimal	150
8	...		WORD	Decimal	
9	...	D100	DINT	Decimal	230
10	...	D102	DINT	Decimal	226
11	...	D104	DINT	Decimal	238

In case of M3=ON, the command performs the bitwise logical XOR operation on (D0, D1), (D2, D3), and (D4, D5) and (D10, D11), (D12, D13), and (D14, D15) and stores the results to (D100, D101), (D102, D103), and (D104, D105).

3.16.15 BKINV: Commands for Inversion Operation of Word/Doubleword Data Block

Command table	*BKINV (S) (D) (n)				Applicable model	TS600 series		
16-Bit command	BKINV: Inversion operation of batch word data blocks							
32-Bit command	DBKINV: Inversion operation of batch doubleword data blocks							
Operand	Type	Bit			Word		Indexing	Constant
		X, Y, M, LM, T, C, S	Dx.y	Custom bit variable	D, R, V, Z, T, C	Custom word variable		
S	WORD / DWORD, Array*n	-	-	-	√[1]	√	√	√
D	WORD / DWORD, Array*n	-	-	-	√[1]	√	√	-
n	WORD	-	-	-	√[1]	√	-	√

Remark:

[1] The D, V, and R elements are supported.

Operand Description

S: The source operand, which indicates the address of the data or data storage word soft element that participates in the inversion operation.

D: The destination operand, which indicates the address of the data storage word soft element of the operation result.

n: The number of bit elements, or the quantity of comparisons.

Function Description

1. 16-bit command: When the energy flow is valid, the command is executed, which performs the bitwise logical AND operation on the n 16-bit data starting from S and then stores the operation results in the n 16-bit data starting from D.
2. 32-bit command: When the energy flow is valid, the command is executed, which performs the bitwise logical AND operation on the n 32-bit data starting from S and then stores the operation results in the n 32-bit data starting from D.
3. The rules of logical inversion operation: If any data is 1, the result is 0; if any data is 0, the result is 1. For example: ~1=0 ~0=1.

Application Example

		Element Name	Data Type	Display Format	Current Value
1	...	D0	INT	Decimal	1
2	...	D1	INT	Decimal	2
3	...	D2	INT	Decimal	3
4	...		WORD	Decimal	
5	...	D10	INT	Decimal	-2
6	...	D11	INT	Decimal	-3
7	...	D12	INT	Decimal	-4

In case of M0=ON, the command performs the inversion operation on D0, D1, and D2 respectively and stores the results to D10, D11, and D12.

		Element Name	Data Type	Display Fo:	Current Value
1	...	D0	DINT	Decimal	11
2	...	D2	DINT	Decimal	22
3	...	D4	DINT	Decimal	33
4	...		WORD	Decimal	
5	...	D10	DINT	Decimal	-12
6	...	D12	DINT	Decimal	-23
7	...	D14	DINT	Decimal	-34

In case of M1=ON, the command performs the inversion operation on doublewords (D0, D1), (D2, D3), and (D4, D5) respectively and stores the results to (D10, D11), (D12, D13), and (D14, D15).

3.17 Data Table Command

3.17.1 Command Table

Command Category	Name	Function
Data Table Command	LIMIT	Upper-lower limit control
	DBAND	Deadband control
	ZONE	Zone control
	*SCL	Coordinate determination of word/doubleword data
	SER	Data retrieval

3.17.2 LIMIT: Commands for Upper-Lower Limit Control

Command table		LIMIT (S1) (S2) (S3) (D)			Applicable model	TS600 series		
16-Bit command		LIMIT: Upper-lower limit control						
32-Bit command		-						
Operand	Type	Bit			Word		Indexing	Constant
		X, Y, M, LM, T, C, S	Dx.y	Custom bit variable	D, R, V, Z, T, C	Custom word variable		
S1	INT	-	-	-	√[1]	√	√	√
S2	INT	-	-	-	√[1]	√	√	√
S3	INT	-	-	-	√[1]	√	√	-
D	INT	-	-	-	√[1]	√	√	-

Remark:

[1] The Z element is not supported.

Operand Description

S1: The lower limit value.

S2: The upper limit value.

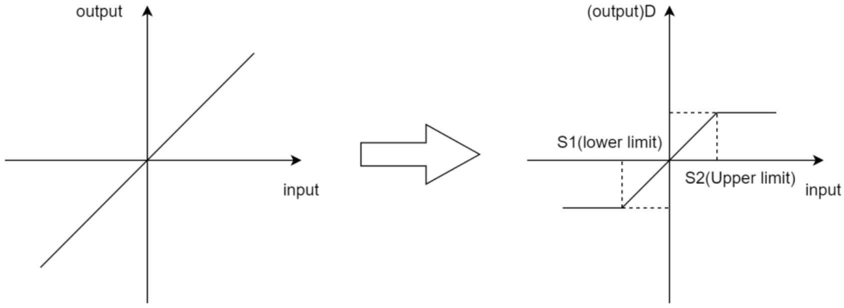
S3: The input value to be controlled through the upper and lower limits.

D: The starting number of the soft element that stores the output value controlled through the upper and lower limits.

Function Description

The command judges whether the input value specified by S3 is within the range of the upper and lower limits specified by S1 and S2:

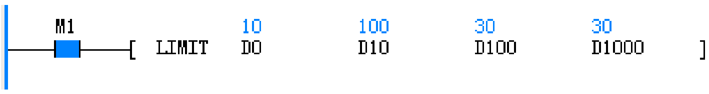
In case of S3<S1, D=S1 is output; in case of S3>S2, D=S2 is output; in case of S1⩽S3⩽S2, D=S2 is output.



Precautions

In case of S1>S2, a parameter value error is reported and the command is not executed.

Application Example



In case of M1=ON, the command performs limit control from D0 to D10 on the content of D100 unit and stores the result in D1000. D0⩽D100⩽D10, D1000=30.

3.17.3 DBAND: Commands for Deadband Control

Command table		DBAND (S1) (S2) (S3) (D)			Applicable model	TS600 series		
16-Bit command		DBAND: Deadband control						
32-Bit command		-						
Operand	Type	Bit			Word		Indexing	Constant
		X, Y, M, LM, T, C, S	Dx.y	Custom bit variable	D, R, V, Z, T, C	Custom word variable		
S1	INT	-	-	-	√[1]	√	√	√
S2	INT	-	-	-	√[1]	√	√	√
S3	INT	-	-	-	√[1]	√	√	-
D	INT	-	-	-	√[1]	√	√	-

Remark:

[1] The Z element is not supported.

Operand Description

S1: The lower limit value of the deadband.

S2: The upper limit value of the deadband.

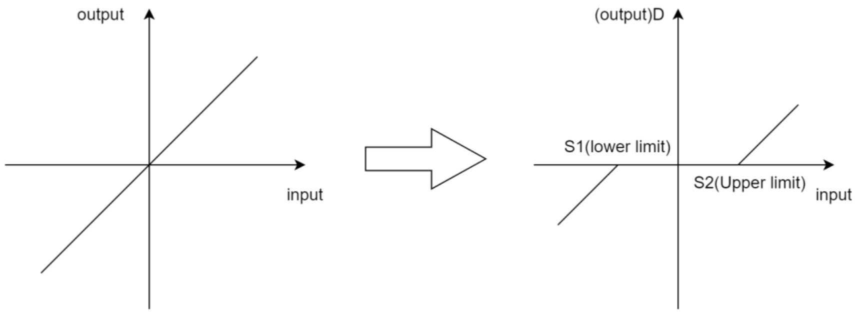
S3: The input value to be controlled through the deadband.

D: The starting number of the soft element that stores the output value controlled through the deadband.

Function Description

The command judges whether the input value specified by S3 is within the range of the upper and lower limits specified by S1 and S2:

In case of S3<S1, D=S3-S1 is output; in case of S3>S2, D=S3-S2 is output; in case of S1⩽S3⩽S2, D=0 is output.



Precautions

In case of S1>S2, a parameter value error is reported and the command is not executed.

Application Example

```
      M1                              -100       100       30        0
──┤ ├──────[  DBAND    D0         D10       D100      D1000        ]
```

In case of M1=ON, the command performs deadband control from D0 to D10 on the content of D100 unit and stores the result in D1000. D0≤D100≤D10, D1000=0.

3.17.4 ZONE: Commands for Zone Control

Command table		ZONE (S1) (S2) (S3) (D)			Applicable model	TS600 series		
16-Bit command		ZONE: Zone control						
32-Bit command		-						
Operand	Type	Bit			Word		Indexing	Constant
		X, Y, M, LM, T, C, S	Dx.y	Custom bit variable	D, R, V, Z, T, C	Custom word variable		
S1	INT	-	-	-	√[1]	√	√	√
S2	INT	-	-	-	√[1]	√	√	√
S3	INT	-	-	-	√[1]	√	√	-
D	INT	-	-	-	√[1]	√	√	-

Remark:

[1] The Z element is not supported.

Operand Description

S1: The negative deviation value added to the input value.

S2: The positive deviation value added to the input value.

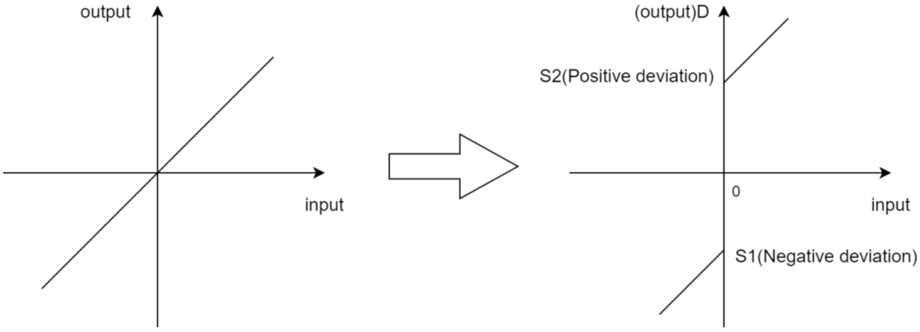
S3: The input value to be controlled through the zone.

D: The starting number of the soft element that stores the output value controlled through the zone.

Function Description

The command judges the input value specified by S3 plus the deviation value specified by S1 or S2:

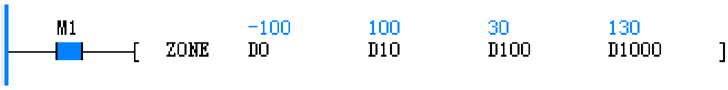
In case of S3<0, D=S3+S1 is output; in case of S3>0, D=S3+S2 is output; in case of S3=0, D=0 is output.



Precautions

In case of S1>S2, a parameter value error is reported and the command is not executed.

Application Example



In case of M1=ON, the command performs zone control from D0 to D10 on the content of D100 unit and stores the result in D1000. D100(30)>0, D1000=D100(30)+D10(100), D1000=130.

3.17.5 SCL: Commands for Coordinate Determination of Word/Doubleword Data

Command table		*SCL (S1) (S2) (D)			Applicable model	TS600 series		
16-Bit command		SCL: Coordinate determination of word data						
32-Bit command		DSCL: Coordinate determination of doubleword data						
Operand	Type	Bit			Word		Indexing	Constant
		X, Y, M, LM, T, C, S	Dx.y	Custom bit variable	D, R, V, Z, T, C	Custom word variable		
S1	INT/DINT	-	-	-	√[1]	√	√	√
S2	INT/DINT	-	-	-	√[2]	√	√	-
D	INT/DINT	-	-	-	√[1]	√	√	-

Remark:

[1] For the 16-bit command, the Z element is not supported; for the 32-bit command, the Z and T elements are not supported.

[2] The D, V, and R elements are supported.

Operand Description

S1: The number of the soft element performing coordinate determination on the input value or storing the input value. If it is less than x1, the system reports a parameter error.

S2: The starting number of the soft element of the conversion table used for coordinate determination. If it is less than 1, the system reports a parameter error.

D: The starting number of the soft element that stores the output value controlled through coordinate determination.

Function Description

1. According to the specified conversion characteristics, the command performs coordinate determination on the input value specified by S1, and then stores the result to the soft element number specified by D.
2. The conversion used for coordinate determination is done according to the data table stored at the starting soft element specified by S2. However, when the output data is not an integer value, it is rounded to the 1st decimal place and then output.

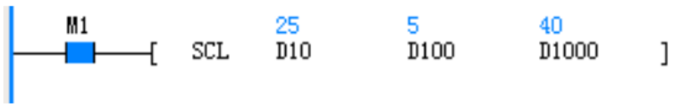
3. See below for settings of the conversion table used for coordinate determination:

Number of Coordinate Points		S2
Point 1	X coordinate	S2+1
	Y coordinate	S2+2
Point 2	X coordinate	S2+3
	Y coordinate	S2+4
...	...	...
Point n (the last)	X coordinate	S2+2n-1
	Y coordinate	S2+2n

Precautions

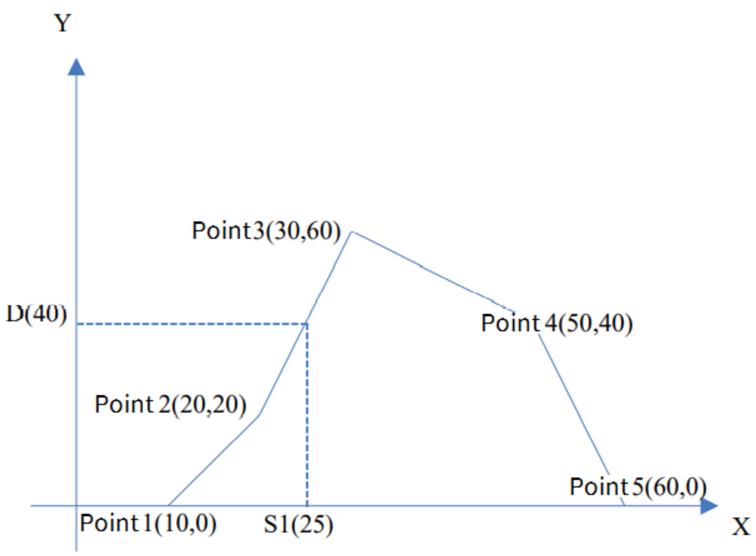
- The data of X in the data table should be arranged in ascending order. If only some parts are not in ascending order and detection starts from the low bit, the operations before these part will still be executed.
- S1 must be within the range set in the data table.

Application Example



In case of M1=ON, the command performs coordinate determination on the content of D10 unit and stores the result in D1000.

Number of Coordinate Points		D100	5
Point 1	X coordinate	D101	10
	Y coordinate	D102	0
Point 2	X coordinate	D103	20
	Y coordinate	D104	20
Point 3	X coordinate	D105	30
	Y coordinate	D106	60
Point 4	X coordinate	D107	50
	Y coordinate	D108	40
Point 5	X coordinate	D109	60
	Y coordinate	D110	0



3.17.6 SER: Commands for Data Retrieval

<table>
<tr><td colspan="2">Command table</td><td colspan="3">SER (S1) (S2) (D) (S3)</td><td>Applicable model</td><td colspan="3">TS600 series</td></tr>
<tr><td colspan="2">16-Bit command</td><td colspan="7">SER: Zone control</td></tr>
<tr><td colspan="2">32-Bit command</td><td colspan="7">-</td></tr>
<tr><td rowspan="2">Operand</td><td rowspan="2">Type</td><td colspan="3">Bit</td><td colspan="2">Word</td><td rowspan="2">Indexing</td><td rowspan="2">Constant</td></tr>
<tr><td>X, Y, M, LM, T, C, S</td><td>Dx.y</td><td>Custom bit variable</td><td>D, R, V, Z, T, C</td><td>Custom word variable</td></tr>
<tr><td>S1</td><td>INT</td><td>-</td><td>-</td><td>-</td><td>√[1]</td><td>√</td><td>√</td><td>-</td></tr>
<tr><td>S2</td><td>INT</td><td>-</td><td>-</td><td>-</td><td>√[1]</td><td>√</td><td>√</td><td>√</td></tr>
<tr><td>D</td><td>INT</td><td>-</td><td>-</td><td>-</td><td>√[1]</td><td>√</td><td>√</td><td>-</td></tr>
<tr><td>S3</td><td>INT</td><td>-</td><td>-</td><td>-</td><td>√[2]</td><td>√</td><td>-</td><td>√</td></tr>
</table>

Remark:

[1] The Z element is not supported.

[2] The D, V, and R elements are supported.

Operand Description

S1: The number of the starting soft element that retrieves the same data, maximum value, and minimum value.

S2: The number of the soft element that retrieves the reference value for the same data, maximum value, and minimum value or stores the target.

D: The number of starting soft element that retrieves the same data, maximum value, and minimum value and then stores the number of these items.

S3: The number of the retrieved same data, maximum values, and minimum values ($1 \leqslant S3 \leqslant 256$).

Function Description

1. The command retrieves the S3 data starting from S1, finds the same data as S2, and stores the results in D-D+4.
2. When the same data are present, the 5 soft elements starting from D store the number of the same data, the position of the initial value, the position of the final value, the position of the minimum value, and the position of the maximum value, respectively.
3. When the same data are not present, the first 3 soft elements store 0, while the last 2 soft elements store the position of the minimum value and the position of the maximum value, respectively.

Application Example

In case of M1=ON, the command retrieves the contents of the 8 units starting from D10 and stores the retrieval results in the 5 units starting from D1000.

Retrieved element, S1	Numeric value	Compared element value, S2	Data location	Retrieval result, D	Numeric value
D10	100	100	0	D1000	3
D11	78	-	1	D1001	0
D12	92	-	2	D1002	7
D13	100	-	3	D1003	5
D14	110	-	4	D1004	6
D15	-20	-	5	-	-
D16	145	-	6	-	-
D17	100	-	7	-	-

3.18 Table Operation Command

3.18.1 Command Table

Command Category	Name	Function
Table Operation Command	*SORTR	Commands to Sort word/doubleword data by row
	*SORTC	Commands to Sort word/doubleword data by column
	FDEL	Commands to Data deletion of data table
	FINS	Commands to Data insertion of data table

3.18.2 SORTR: Commands to Sort Word/Doubleword Data by Row

<table>
<tr><th colspan="2">Command table</th><th colspan="3">*SORTR (S1) (m1) (m2) (D1) (n) (D2)</th><th>Applicable model</th><th colspan="3">TS600 series</th></tr>
<tr><td colspan="2">16-Bit command</td><td colspan="7">SORTR: Sort word data by row</td></tr>
<tr><td colspan="2">32-Bit command</td><td colspan="7">DSORTR: Sort doubleword data by row</td></tr>
<tr><td rowspan="2">Operand</td><td rowspan="2">Type</td><td colspan="3">Bit</td><td colspan="2">Word</td><td rowspan="2">Indexing</td><td rowspan="2">Constant</td></tr>
<tr><td>X, Y, M, LM, T, C, S</td><td>Dx.y</td><td>Custom bit variable</td><td>D, R, V, Z, T, C</td><td>Custom word variable</td></tr>
<tr><td>S1</td><td>INT/DINT, Arrays m1*m2</td><td>-</td><td>-</td><td>-</td><td>√[1]</td><td>√</td><td>-</td><td>-</td></tr>
<tr><td>m1</td><td>WORD</td><td>-</td><td>-</td><td>-</td><td>√[1]</td><td>√</td><td>√</td><td>√</td></tr>
<tr><td>m2</td><td>WORD</td><td>-</td><td>-</td><td>-</td><td>√[1]</td><td>√</td><td>√</td><td>√</td></tr>
<tr><td>D1</td><td>INT/DINT, Arrays m1*m2</td><td>-</td><td>-</td><td>-</td><td>√[1]</td><td>√</td><td>-</td><td>-</td></tr>
<tr><td>n</td><td>WORD</td><td>-</td><td>-</td><td>-</td><td>√[1]</td><td>√</td><td>√</td><td>√</td></tr>
</table>

Remark:

[1] Only the D, V, and R soft elements are supported.

Operand Description

S: The source operand, which indicates the starting unit of the first variable in the first row (first record).

M1: The source operand, which indicates the number of rows in an array and is also known as the number of records.

M2: The source operand, which indicates the number of columns in an array and is also known as the number of columns per record.

D1: The destination operand, indicating the starting unit that is used for storage after sorting and occupies the same number of subsequent variable units as the number of array variables before sorting.

n: The source operand, which indicates the array row number based on row sorting. The value range of n is 1~m1.

Function Description

1. This commands sorts the array consisting of m1 rows × m2 columns (described by S, m1, and m2) by the n-th row of parameters, and then stores the results in the variable area starting from the D1 unit.

See below for the sorting process of the 3x3 data:

Before sorting:

	1	2	3
1	S	S+3	S+6
	1	2	8
2	S+1	S+4	S+7
	6	7	2
3	S+2	S+5	S+8
	3	4	3

After sorting by the second row (ascending):

	1	2	3
1	D	D+3	D+6
	8	1	2
2	D+1	D+4	D+7
	2	6	7
3	D+2	D+5	D+8
	3	3	4

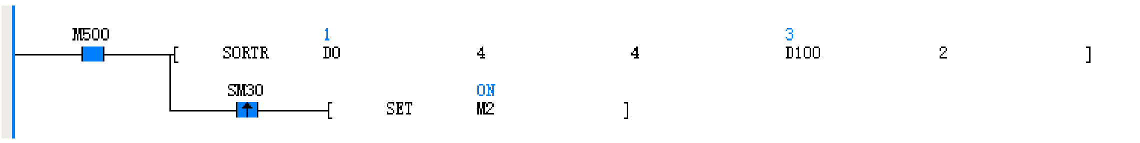
The command sets sorting according to the state (either ON or OFF) of SM33, where ON means descending, while OFF means ascending.

2. When the energy flow is valid, data sorting begins. After m1 scan cycles, the sorting is completed, and command execution is completed to obtain SM30=ON.

Precautions

- During the command execution, the operand cannot be modified.
- To re-sort, perform the OFF → ON operation on the energy flow.
- During the sorting process by the command, ensure that the operands and table content are not changed.
- The source operand S cannot partially overlap with D1, and it can only overlap with the latter completely or not at all, otherwise the system reports an error of overlapping source and destination operands.
- If you use the 32-bit command, the data table content occupies 2 16-bit soft elements.
- After sorting is completed, SM30 will be set. If multiple command are used for sorting, the value of SM30 will be overwritten by the subsequent sorting commands.
- Up to 128 SORTR commands are supported.

Application Example



In case of M500=ON, the SORTR command begins to execute, which sorts the 4*4 data table elements starting from D0 in ascending order according to the 2nd row. The sorting results are stored in the 4*4 table data starting from D100, and the sorting process data are displayed in the 4 data starting from D200. After sorting is completed, M2 is set.

Before sorting:

		Element Name	Data Type	Display Format	Current Value
1	...	D0	INT	Decimal	1
2	...	D1	INT	Decimal	2
3	...	D2	INT	Decimal	3
4	...	D3	INT	Decimal	2
5	...	D4	INT	Decimal	6
6	...	D5	INT	Decimal	4
7	...	D6	INT	Decimal	8
8	...	D7	INT	Decimal	7
9	...	D8	INT	Decimal	3
10	...	D9	INT	Decimal	1
11	...	D10	INT	Decimal	2
12	...	D11	INT	Decimal	3
13	...	D12	INT	Decimal	2
14	...	D13	INT	Decimal	6
15	...	D14	INT	Decimal	4
16	...	D15	INT	Decimal	8

After sorting:

		Element Name	Data Type	Display Format	Current Value
18	...	D100	INT	Decimal	3
19	...	D101	INT	Decimal	1
20	...	D102	INT	Decimal	2
21	...	D103	INT	Decimal	3
22	...	D104	INT	Decimal	1
23	...	D105	INT	Decimal	2
24	...	D106	INT	Decimal	3
25	...	D107	INT	Decimal	2
26	...	D108	INT	Decimal	6
27	...	D109	INT	Decimal	4
28	...	D110	INT	Decimal	8
29	...	D111	INT	Decimal	7
30	...	D112	INT	Decimal	2
31	...	D113	INT	Decimal	6
32	...	D114	INT	Decimal	4
33	...	D115	INT	Decimal	8

3.18.3 SORTC: Commands to Sort Word/Doubleword Data by Column

Command table		*SORTC (S1) (m1) (m2) (D1) (n) (D2)			Applicable model	TS600 series		
16-Bit command		SORTC: Sort word data by column						
32-Bit command		DSORTC: Sort doubleword data by column						
Operand	Type	Bit			Word		Indexing	Constant
		X, Y, M, LM, T, C, S	Dx.y	Custom bit variable	D, R, V, Z, T, C	Custom word variable		
S1	INT/DINT, Arrays m1*m2	-	-	-	√[1]	√	-	-
m1	WORD	-	-	-	√[1]	√	√	√
m2	WORD	-	-	-	√[1]	√	√	√
D1	INT/DINT, Arrays m1*m2	-	-	-	√[1]	√	-	-
n	WORD	-	-	-	√[1]	√	√	√

Remark:

[1] Only the D, V, and R soft elements are supported.

Operand Description

S: The source operand, which indicates the starting unit of the first variable in the first row (first record).

M1: The source operand, which indicates the number of rows in an array and is also known as the number of records.

M2: The source operand, which indicates the number of columns in an array and is also known as the number of columns per record.

D1: The destination operand, indicating the starting unit that is used for storage after sorting and occupies the same number of subsequent variable units as the number of array variables before sorting.

n: The source operand, which indicates the array row number based on column sorting. The value range of n is 1~m2.

Function Description

1. This commands sorts the array consisting of m1 rows × m2 columns (described by S, m1, and m2) by the n-th column of parameters, and then stores the results in the variable area starting from the D1 unit.

See below for the sorting process of the 3x3 data:

Before sorting:

	1	**2**	**3**
1	S	S+3	S+6
	1	2	8
2	S+1	S+4	S+7
	6	7	2
3	S+2	S+5	S+8
	3	4	3

After sorting by the second column (ascending):

	1	**2**	**3**
1	D	D+3	D+6
	8	1	2
2	D+1	D+4	D+7
	2	6	7
3	D+2	D+5	D+8
	3	3	4

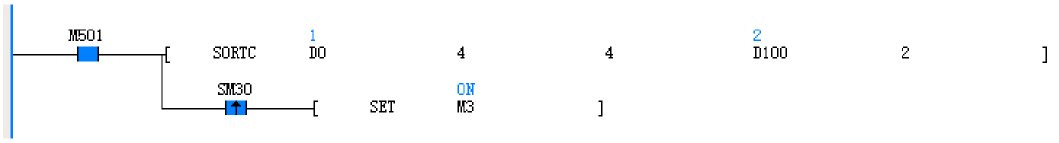
The command sets sorting according to the state (either ON or OFF) of SM33, where ON means descending, while OFF means ascending.

2. When the energy flow is valid, data sorting begins. After m2 scan cycles, the sorting is completed, and command execution is completed to obtain SM30=ON.

Precautions

- During the command execution, the operand cannot be modified.
- To re-sort, perform the OFF → ON operation on the energy flow.
- During the sorting process by the command, ensure that the operands and table content are not changed.
- The source operand S cannot partially overlap with D1, and it can only overlap with the latter completely or not at all, otherwise the system reports an error of overlapping source and destination operands.
- If you use the 32-bit command, the data table content occupies 2 16-bit soft elements.
- After sorting is completed, SM30 will be set. If multiple command are used for sorting, the value of SM30 will be overwritten by the subsequent sorting commands.
- Up to 128 SORTC commands are supported.

Application Example



In case of M501=ON, the SORTC command begins to execute, which sorts the 4*4 data table elements starting from D0 in ascending order according to the 2nd column. The sorting results are stored in the 4*4 table data starting from D100, and the sorting process data are displayed in the 4 data starting from D200. After sorting is completed, M3 is set.

Before sorting:

		Element Name	Data Type	Display Format	Current Value
1	...	D0	INT	Decimal	1
2	...	D1	INT	Decimal	2
3	...	D2	INT	Decimal	3
4	...	D3	INT	Decimal	2
5	...	D4	INT	Decimal	6
6	...	D5	INT	Decimal	4
7	...	D6	INT	Decimal	8
8	...	D7	INT	Decimal	7
9	...	D8	INT	Decimal	3
10	...	D9	INT	Decimal	1
11	...	D10	INT	Decimal	2
12	...	D11	INT	Decimal	3
13	...	D12	INT	Decimal	2
14	...	D13	INT	Decimal	6
15	...	D14	INT	Decimal	4
16	...	D15	INT	Decimal	8

After sorting:

		Element Name	Data Type	Display Format	Current Value
18	...	D100	INT	Decimal	2
19	...	D101	INT	Decimal	1
20	...	D102	INT	Decimal	2
21	...	D103	INT	Decimal	3
22	...	D104	INT	Decimal	4
23	...	D105	INT	Decimal	6
24	...	D106	INT	Decimal	7
25	...	D107	INT	Decimal	8
26	...	D108	INT	Decimal	1
27	...	D109	INT	Decimal	3
28	...	D110	INT	Decimal	3
29	...	D111	INT	Decimal	2
30	...	D112	INT	Decimal	6
31	...	D113	INT	Decimal	2
32	...	D114	INT	Decimal	8
33	...	D115	INT	Decimal	4

3.18.4 FDEL: Commands for Data Deletion of Data Table

<table>
<tr><td colspan="2">Command table</td><td colspan="3">FDEL (S) (D) (n)</td><td>Applicable model</td><td colspan="3">TS600 series</td></tr>
<tr><td colspan="2">16-Bit command</td><td colspan="7">FDEL: Data deletion of data table</td></tr>
<tr><td colspan="2">32-Bit command</td><td colspan="7">-</td></tr>
<tr><td rowspan="2">Operand</td><td rowspan="2">Type</td><td colspan="3">Bit</td><td colspan="2">Word</td><td rowspan="2">Indexing</td><td rowspan="2">Constant</td></tr>
<tr><td>X, Y, M, LM, T, C, S</td><td>Dx.y</td><td>Custom bit variable</td><td>D, R, V, Z, T, C</td><td>Custom word variable</td></tr>
<tr><td>S</td><td>INT, Array*[S]+1</td><td>-</td><td>-</td><td>-</td><td>√[1]</td><td>√</td><td>-</td><td>-</td></tr>
<tr><td>D</td><td>INT</td><td>-</td><td>-</td><td>-</td><td>√[1]</td><td>√</td><td></td><td>-</td></tr>
<tr><td>n</td><td>WORD</td><td>-</td><td>-</td><td>-</td><td>√[1]</td><td>√</td><td>√</td><td>√</td></tr>
</table>

Remark:

[1] Only the D, V, and R soft elements are supported.

Operand Description

S: The data table information.

S: The number of saved data.

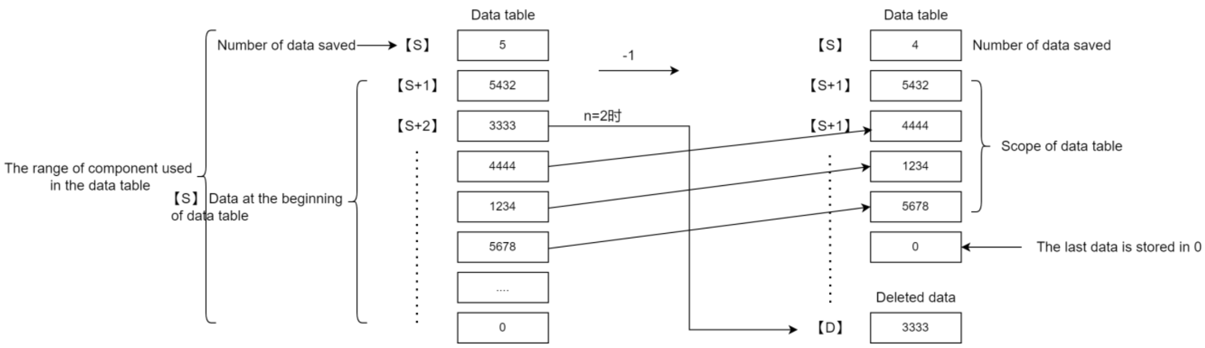
S+1: The starting position of the data table.

D: The soft element that saves the deleted data.

n: The table position of the data to be deleted.

Function Description

The command deletes the ***n***-th data in the data table starting from D+1, saves the deleted data to S, moves the data starting from the n+1-th data in D+1 forwards, and subtracts 1 from the saved data number D.



Precautions

- This command continuously performs the deletion operation, which is usually executed in conjunction with the rising edge trigger.
- When the saved data number exceeds the range of the soft elements, the system reports an out-of-range address error.
- When the deleted position n is greater than the saved data number S, the system reports an illegal operand error.
- When the set value of n is ≤ 0, the system reports an illegal operand error.
- When the set value of the saved data number is ≤ 0, the system reports an illegal operand error.

Application Example

Before execution:

		Element Name	Data Type	Display Fo:	Current Value
1	...	R100	WORD	Decimal	5
2	...	R101	WORD	Decimal	1111
3	...	R102	WORD	Decimal	2222
4	...	R103	WORD	Decimal	3333
5	...	R104	WORD	Decimal	4444
6	...	R105	WORD	Decimal	5555
7	...	R106	WORD	Decimal	0
8	...		WORD	Decimal	
9	...	D200	WORD	Decimal	3
10	...		WORD	Decimal	
11	...	D100	WORD	Decimal	0

After execution:

		Element Name	Data Type	Display Fo:	Current Value
1	...	R100	WORD	Decimal	4
2	...	R101	WORD	Decimal	1111
3	...	R102	WORD	Decimal	2222
4	...	R103	WORD	Decimal	4444
5	...	R104	WORD	Decimal	5555
6	...	R105	WORD	Decimal	0
7	...	R106	WORD	Decimal	0
8	...		WORD	Decimal	
9	...	D200	WORD	Decimal	3
10	...		WORD	Decimal	
11	...	D100	WORD	Decimal	3333

3.18.5 FINS: Commands for Data Insertion of Data Table

Command table		FINS (S1) (S2) (n)			Applicable model	TS600 series		
16-Bit command		FINS: Data insertion of data table						
32-Bit command		-						
Operand	Type	Bit			Word		Indexing	Constant
		X, Y, M, LM, T, C, S	Dx.y	Custom bit variable	D, R, V, Z, T, C	Custom word variable		
S1	INT, Array*[S1]+2	-	-	-	√[1]	√	-	-
S2	INT	-	-	-	√[1]	√	-	-
n	WORD	-	-	-	√[1]	√	√	√

Remark:

[1] Only the D, V, and R soft elements are supported.

Operand Description

S1: The data table information.

S1: The number of saved data.

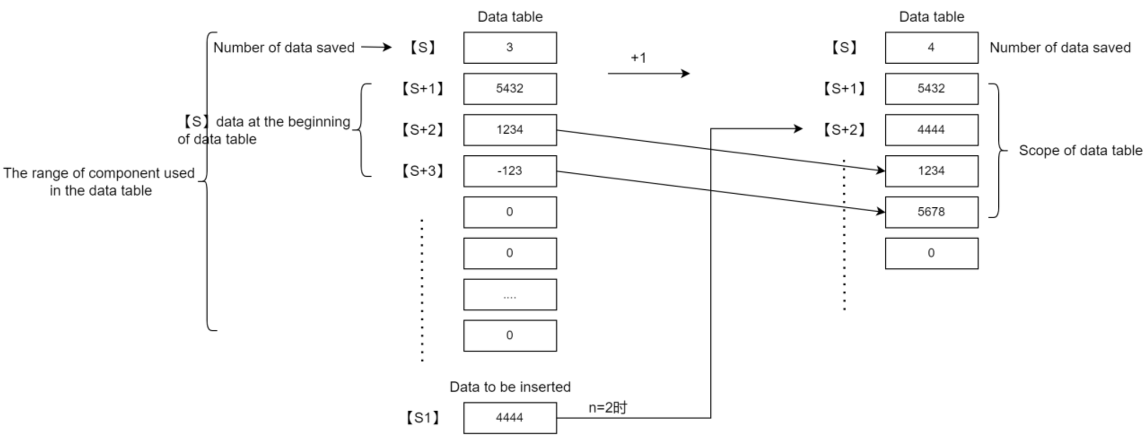
S1+1: The starting position of the data table.

S2: The soft element that saves the inserted data.

n: The table position of the data to be inserted.

Function Description

The command inserts the data of S to the n-th data in the data table starting from D+1, moves the data starting from the n-th data in the data table backwards one by one, and adds 1 to the saved data number D.



Precautions

- This command continuously performs the insertion operation, which is usually executed in conjunction with the rising edge trigger.
- When the saved data number exceeds the range of the soft elements, the system reports an out-of-range address error.
- When the deleted position n is greater than the saved data number S, the system reports an illegal operand error.
- When the set value of n is ⩽ 0, the system reports an illegal operand error.
- When the set value of the saved data number is ⩽ 0, the system reports an illegal operand error.

Application Example

Before execution:

		Element Name	Data Type	Display Fo:	Current Value
1	...	R100	WORD	Decimal	4
2	...	R101	WORD	Decimal	1111
3	...	R102	WORD	Decimal	2222
4	...	R103	WORD	Decimal	4444
5	...	R104	WORD	Decimal	5555
6	...	R105	WORD	Decimal	0
7	...	R106	WORD	Decimal	0
8	...		WORD	Decimal	
9	...	D100	WORD	Decimal	3333
10	...		WORD	Decimal	
11	...	D200	WORD	Decimal	3

After execution:

		Element Name	Data Type	Display Fo:	Current Value
1	...	R100	WORD	Decimal	5
2	...	R101	WORD	Decimal	1111
3	...	R102	WORD	Decimal	2222
4	...	R103	WORD	Decimal	3333
5	...	R104	WORD	Decimal	4444
6	...	R105	WORD	Decimal	5555
7	...	R106	WORD	Decimal	0
8	...		WORD	Decimal	
9	...	D100	WORD	Decimal	3333
10	...		WORD	Decimal	
11	...	D200	WORD	Decimal	3

3.19 String Command

3.19.1 Command Table

Command Category	Name	Function
String Command	STRADD	String combination
	STRLEN	String length detection
	STRRIGHT	Read from right side of string
	STRLEFT	Read from left side of string
	STRMIDR	Randomly read from string
	STRMIDW	Randomly replace from string
	STRINSTR	String retrieval
	STRMOV	String transfer

3.19.2 STRADD: Commands for String Combination

<table>
<tr><td colspan="2">Command table</td><td colspan="3">STRADD (S1) (S2) (D)</td><td>Applicable model</td><td colspan="3">TS600 series</td></tr>
<tr><td colspan="2">16-Bit command</td><td colspan="7">STRADD: String combination</td></tr>
<tr><td colspan="2">32-Bit command</td><td colspan="7">-</td></tr>
<tr><td rowspan="2">Operand</td><td rowspan="2">Type</td><td colspan="3">Bit</td><td colspan="2">Word</td><td rowspan="2">Indexing</td><td rowspan="2">Constant</td></tr>
<tr><td>X, Y, M, LM, T, C, S</td><td>Dx.y</td><td>Custom bit variable</td><td>D, R, V, Z, T, C</td><td>Custom word variable</td></tr>
<tr><td>S1</td><td>INT/ Array*number</td><td>-</td><td>-</td><td>-</td><td>√[1]</td><td>√</td><td>√</td><td>√[2]</td></tr>
<tr><td>S2</td><td>INT/ Array*number</td><td>-</td><td>-</td><td>-</td><td>√[1]</td><td>√</td><td>√</td><td>√[2]</td></tr>
<tr><td>D</td><td>INT/ Array*number</td><td>-</td><td>-</td><td>-</td><td>√[1]</td><td>√</td><td>√</td><td>-</td></tr>
</table>

Remark:

[1]The Z element is not supported.

[2]Here it represents a string constant.

Operand Description

S1: The first string unit.

S2: The second string unit.

D: The string storage unit after connection.

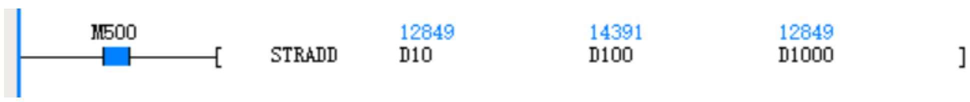
Function Description

1. When the energy flow is valid, the command connects the string units starting from S1 and S2 and stores the results to the soft elements starting from D.
2. String combination refers to connecting the first character of the S2 unit string to the last character of the S1 unit string and ignoring the end marker of the S1 unit string.
3. The valid data of a string unit is the data from the specified soft element of the string unit to the position where the first "00H" is detected.
4. If the number of connected characters is odd, "00H" is added to the high byte of the soft element that stores the last character. If it is even, "0000H" is added to the next element after the soft element that stores last character.

Precautions

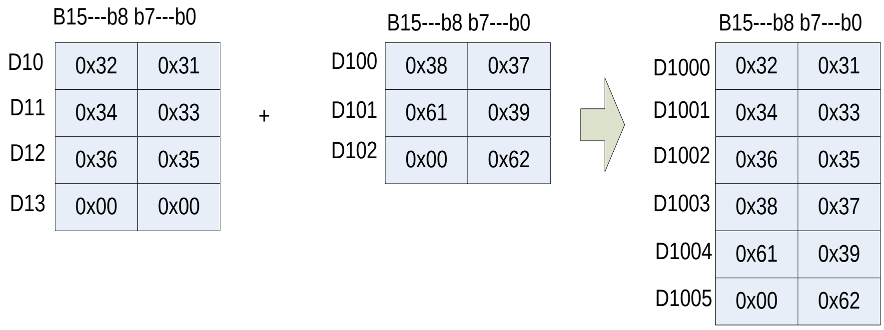
- When "00H" is not found within the allowed ranges of S1 and S2, the system reports a string data or length error.
- When the merged string exceeds the allowed range of D, the system reports a string data or length error.

Application Example



		Element Name	Data Type	Display Format	Current Value
1	...	D10	WORD	Hexadecimal	16#3231
2	...	D11	WORD	Hexadecimal	16#3433
3	...	D12	WORD	Hexadecimal	16#3635
4	...	D13	WORD	Hexadecimal	16#0
5	...		WORD	Hexadecimal	
6	...	D100	WORD	Hexadecimal	16#3837
7	...	D101	WORD	Hexadecimal	16#6139
8	...	D102	WORD	Hexadecimal	16#62
9	...		WORD	Decimal	
10	...	D1000	WORD	Hexadecimal	16#3231
11	...	D1001	WORD	Hexadecimal	16#3433
12	...	D1002	WORD	Hexadecimal	16#3635
13	...	D1003	WORD	Hexadecimal	16#3837
14	...	D1004	WORD	Hexadecimal	16#6139
15	...	D1005	WORD	Hexadecimal	16#62

In case of M500=ON, the command connects the string units starting from D10 with the string units starting from D100, and stores the results in the units starting from D1000. See the figure below for the process.



3.19.3 STRLEN: Commands for String Length Detection

<table>
<tr><td colspan="2">Command table</td><td colspan="3">STRLEN (S) (D)</td><td>Applicable model</td><td colspan="3">TS600 series</td></tr>
<tr><td colspan="2">16-Bit command</td><td colspan="7">STRLEN: String length detection</td></tr>
<tr><td colspan="2">32-Bit command</td><td colspan="7">-</td></tr>
<tr><td rowspan="2">Operand</td><td rowspan="2">Type</td><td colspan="3">Bit</td><td colspan="2">Word</td><td rowspan="2">Indexing</td><td rowspan="2">Constant</td></tr>
<tr><td>X, Y, M, LM, T, C, S</td><td>Dx.y</td><td>Custom bit variable</td><td>D, R, V, Z, T, C</td><td>Custom word variable</td></tr>
<tr><td>S</td><td>INT/ Array*number</td><td>-</td><td>-</td><td>-</td><td>√[1]</td><td>√</td><td>√</td><td>-</td></tr>
<tr><td>D</td><td>WORD</td><td>-</td><td>-</td><td>-</td><td>√</td><td>√</td><td>√</td><td>-</td></tr>
</table>

Remark:

[1] The Z element is not supported.

Operand Description

S: The string unit.

D: The length of the string unit.

Function Description

1. When the energy flow is valid, the command detects the length of the S unit string (the number of characters/bytes in the string) and stores the value in D.
2. The valid data of a string unit is the data from the specified soft element of the string unit to the position where the first "00H" is detected.

Precautions

- When "00H" is not found within the allowed range of S, the system reports a string data or length error.
- When the number of detected characters is greater than 32767, the system reports a string data or length error.

Application Example

		Element Name	Data Type	Display Format	Current Value
1	...	D0	WORD	Hexadecimal	16#1122
2	...	D1	WORD	Hexadecimal	16#3344
3	...	D2	WORD	Hexadecimal	16#55
4	...	D3	WORD	Hexadecimal	16#0
5	...		WORD	Decimal	
6	...	D100	WORD	Decimal	5

In case of M501=ON, the command retrieves the length of the string unit starting from D100 and stores the result in D100.

3.19.4 STRRIGHT: Commands Used to Read from Right Side of String

Command table		STRRIGHT (S) (D) (n)			Applicable model	TS600 series		
16-Bit command		STRRIGHT: Read from right side of string						
32-Bit command		-						
Operand	Type	Bit			Word		Indexing	Constant
		X, Y, M, LM, T, C, S	Dx.y	Custom bit variable	D, R, V, Z, T, C	Custom word variable		
S	INT/ Array*number	-	-	-	√[1]	√	√	-
D	INT/ Array*number	-	-	-	√[1]	√	√	-
n	WORD	-	-	-	√	√	√	√

Remark:

[1] The Z element is not supported.

Operand Description

S: The string unit.

D: The extracted string unit saved.

n: The number of characters extracted; 0≤n<32767.

Function Description

1. When the energy flow is valid, the command extracts n characters from the left side of the S string unit to the right, and stores them in the soft elements starting from D.
2. When n equals zero, "00H" is stored in the D soft element.
3. If the number of extracted characters is odd, "00H" is added to the high byte of the soft element that stores the last character. If it is even, "0000H" is added to the next element after the soft element that stores last character.
4. The valid data of a string unit is the data from the specified soft element of the string unit to the position where the first "00H" is detected.

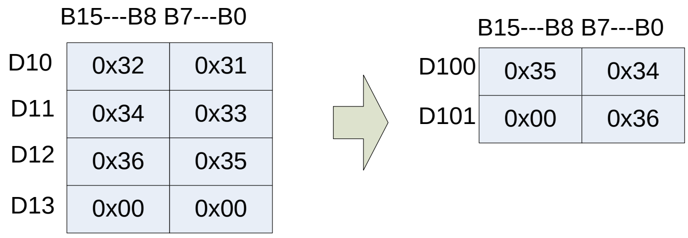
Precautions

- When "00H" is not found within the corresponding soft element range of the string unit starting from S, the system reports a string data or length error.
- n should be greater than or equal to 0 and also less than or equal to the number of characters in the S string unit, otherwise the system reports an illegal operand error.
- If the retrieved string data cannot be stored within the legal range of D, the system reports an error that the parameter exceeds the limit address range.

Application Example

		Element Name	Data Type	Display Fo	Current Value
1	...	D10	WORD	Hexadecima	16#3231
2	...	D11	WORD	Hexadecima	16#3433
3	...	D12	WORD	Hexadecima	16#3635
4	...	D13	WORD	Hexadecima	16#0
5	...		WORD	Hexadecima	
6	...	D100	WORD	Hexadecima	16#3534
7	...	D101	WORD	Hexadecima	16#36

In case of M502=ON, the command extracted 3 characters from the right side of the string unit starting from D10, and stores them in the unit starting from D100. See the figure below for the process.



3.19.5 STRLEFT: Commands Used to Read from Left Side of String

Command table		STRLEFT (S) (D) (n)			Applicable model	TS600 series		
16-Bit command		STRLEFT: Read from left side of string						
32-Bit command		-						
Operand	Type	Bit			Word		Indexing	Constant
		X, Y, M, LM, T, C, S	Dx.y	Custom bit variable	D, R, V, Z, T, C	Custom word variable		
S	INT/ Array*number	-	-	-	√[1]	√	√	-
D	INT/ Array*number	-	-	-	√[1]	√	√	-
n	WORD	-	-	-	√	√	√	√

Remark:

[1] The Z element is not supported.

Operand Description

S: The string unit.

D: The extracted string unit saved.

n: The number of characters extracted; 0≤n<32767.

Function Description

1. When the energy flow is valid, the command extracts n characters starting from the last valid character

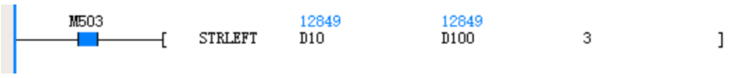
(excluding "00H") of the string in the S unit, and stores them in the soft elements starting from D.

2. When n equals zero, "00H" is stored in the D soft element.
3. If the number of extracted characters is odd, "00H" is added to the high byte of the soft element that stores the last character. If it is even, "0000H" is added to the next element after the soft element that stores last character.
4. The valid data of a string unit is the data from the specified soft element of the string unit to the position where the first "00H" is detected.

Precautions

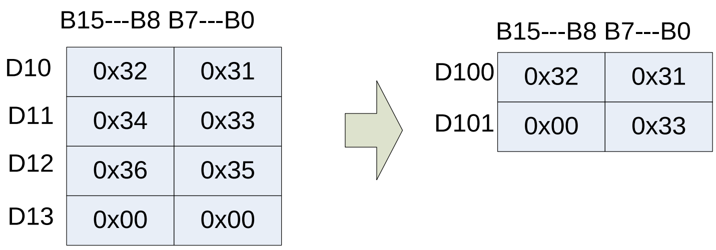
- When "00H" is not found within the corresponding soft element range of the string unit starting from S, the system reports a string data or length error.
- n should be greater than or equal to 0 and also less than or equal to the number of characters in the S string unit, otherwise the system reports an illegal operand error.
- If the retrieved string data cannot be stored within the legal range of D, the system reports an error that the parameter exceeds the limit address range.

Application Example



		Element Name	Data Type	Display Fo	Current Value
1	...	D10	WORD	Hexadecima	16#3231
2	...	D11	WORD	Hexadecima	16#3433
3	...	D12	WORD	Hexadecima	16#3635
4	...	D13	WORD	Hexadecima	16#0
5	...		WORD	Hexadecima	
6	...	D100	WORD	Hexadecima	16#3231
7	...	D101	WORD	Hexadecima	16#33

In case of M503=ON, the command extracted 3 characters from the left side of the string unit starting from D10, and stores them in the unit starting from D100. See the figure below for the process.



3.19.6 STRMIDR: Commands Used to Randomly Read from String

Command table		STRMIDR (S1) (D) (S2)			Applicable model	TS600 series		
16-Bit command		STRMIDR: Randomly read from string						
32-Bit command		-						
Operand	Type	Bit			Word		Indexing	Constant
		X, Y, M, LM, T, C, S	Dx.y	Custom bit variable	D, R, V, Z, T, C	Custom word variable		
S1	INT/ Array*number	-	-	-	√[1]	√	√	-
D	INT/ Array*number	-	-	-	√[1]	√	√	-
S2	INT/ Array*2	-	-	-	√	√	√	√

Remark:

[1] The Z element is not supported.

Operand Description

S1: The string unit.

D: The extracted string unit saved.

S2: The starting position of the string to be extracted.

S2+1: n, the number of characters to be extracted.

Function Description

1. When the energy flow is valid, the command extracts the data of n characters starting from the S2-nd characters in the S1 string unit, and stores them in the soft elements starting from D.
2. If the number of extracted characters is odd, "00H" is added to the high byte of the soft element that stores the last character. If it is even, "0000H" is added to the next element after the soft element that stores last character.
3. The valid data of a string unit is the data from the specified soft element of the string unit to the position where the first "00H" is detected.
4. When n is 0, the command performs no action.
5. When n is -1, the command extracts all characters starting from S2 in the S1 string unit and stores them in the soft elements starting from D.

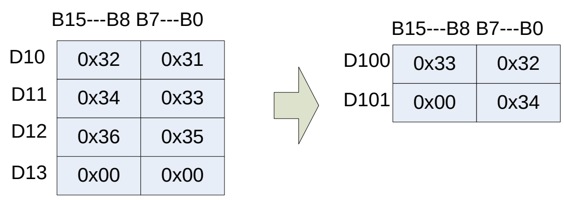
Precautions

- When "00H" is not found within the corresponding soft element range of the string unit starting from S1 or D, the system reports a string data or length error.
- When the set value of S2 is too large and exceeds the number of characters in S1, the system reports an illegal operand error.
- When the set value of S2 is less than 1, the system reports an illegal operand error.
- When the set value of S2+1 is less than -1, the system reports an illegal operand error.
- When the set value of S2+1 exceeds the number of characters in S1, the system reports an illegal operand error.

Application Example

		Element Name	Data Type	Display Format	Current Value
1	...	D10	WORD	Hexadecimal	16#3231
2	...	D11	WORD	Hexadecimal	16#3433
3	...	D12	WORD	Hexadecimal	16#3635
4	...	D13	WORD	Hexadecimal	16#0
5	...		WORD	Hexadecimal	
6	...	D100	WORD	Hexadecimal	16#3332
7	...	D101	WORD	Hexadecimal	16#34
8	...		WORD	Decimal	
9	...	D0	INT	Decimal	2
10	...	D1	INT	Decimal	3

In case of M504=ON, the command reads out the D1 (D1=3) data starting from the D0th (D0=2) data of the string unit starting from D10, and stores them in the units starting from D100. See the figure below for the process.



3.19.7 STRMIDW: Commands Used to Randomly Replace from String

Command table		STRMIDW (S1) (D) (S2)			Applicable model	TS600 series		
16-Bit command		STRMIDW: Randomly replace from string						
32-Bit command		-						
Operand	Type	Bit			Word		Indexing	Constant
		X, Y, M, LM, T, C, S	Dx.y	Custom bit variable	D, R, V, Z, T, C	Custom word variable		
S1	INT/ Array*number	-	-	-	√[1]	√	√	-
D	INT/ Array*number	-	-	-	√[1]	√	√	-
S2	INT/ Array*2	-	-	-	√	√	√	√

Remark:

[1] The Z element is not supported.

Operand Description

S1: The string unit to be used as the replacement.

D: The string unit replaced.

S2: The starting position of the replacement.

S2+1: n, the number of replaced characters.

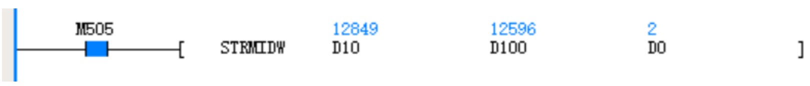
Function Description

1. When the energy flow is valid, the command uses the n characters in the S1 string unit to replace the n characters starting from the S2-nd character in the D string unit.
2. The valid data of a string unit is the data from the specified soft element of the string unit to the position where the first "00H" is detected.
3. When n is 0, the command performs no action.
4. When n is -1, all characters of the S1 string will be replaced sequentially with the characters starting from position S2 to the end character 00H of the D string (which is not replaced).

Precautions

- When "00H" is not found within the corresponding soft element range of the string unit starting from S1 or D, the system reports a string data or length error.
- When the set value of S2 is too large and exceeds the number of characters in D, the system reports an illegal operand error.
- When the set value of S2 is less than 1, the system reports an illegal operand error.
- When the set value of S2+1 is less than -1, the system reports an illegal operand error.
- When the set value of S2+1 exceeds the number of characters in S1, the system reports an illegal operand error.

Application Example



		Element Name	Data Type	Display Format	Current Value
1	...	D10	WORD	Hexadecimal	16#3231
2	...	D11	WORD	Hexadecimal	16#3433
3	...	D12	WORD	Hexadecimal	16#35
4	...	D13	WORD	Hexadecimal	16#0
5	...		WORD	Hexadecimal	
6	...	D100	WORD	Hexadecimal	16#3134
7	...	D101	WORD	Hexadecimal	16#3332
8	...	D102	WORD	Hexadecimal	16#3938
9	...	D103	WORD	Hexadecimal	16#61
10	...		WORD	Decimal	
11	...	D0	INT	Decimal	2
12	...	D1	INT	Decimal	3

In case of M505=ON, the command reads out the D1 (D1=3) data starting from the D0th (D0=2) data of the string unit starting from D10, and stores them in the units starting from D100. See the figure below for the process.

	B15---B8	B7---B0
D10	0x32	0x31
D11	0x34	0x33
D12	0x00	0x35

	B15---B8	B7---B0
D100	0x35	0x34
D101	0x37	0x36
D102	0x39	0x38
D103	0x00	0x61

	B15---B8	B7---B0
D100	0x31	0x34
D101	0x33	0x32
D102	0x39	0x38
D103	0x00	0x61

3.19.8 STRINSTR: Commands for String Retrieval

Command table		STRINSTR (S1) (S2) (D) (n)			Applicable model	TS600 series		
16-Bit command		STRINSTR: String retrieval						
32-Bit command		-						
Operand	Type	Bit			Word		Indexing	Constant
		X, Y, M, LM, T, C, S	Dx.y	Custom bit variable	D, R, V, Z, T, C	Custom word variable		
S1	INT/ Array*number	-	-	-	√[1]	√	√	√[2]
S2	INT/ Array*number	-	-	-	√[1]	√	√	-
D	INT	-	-	-	√	√	√	-
n	INT	-	-	-	√	√	√	√

Remark:

[1] The Z element is not supported.

[2] Here it represents a string constant.

Operand Description

S1: The string unit to be retrieved.

S2: The retrieval source.

D: The retrieval result.

n: The position where the retrieval starts.

Function Description

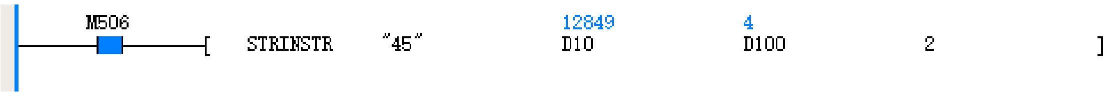
1. When the energy flow is valid, the command retrieves the same strings starting from the nth character of the S2 string unit as the S1 string unit, and stores the string position information of the retrieved result in D.

2. When there is no consistent string, the command stores "0" in D.
3. When n, the retrieval starting position, is a negative or "0", the command performs no action.
4. The valid data of a string unit is the data from the specified soft element of the string unit to the position where the first "00H" is detected.

Precautions

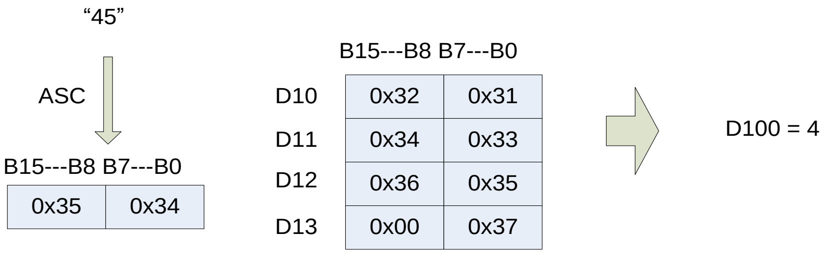
- When "00H" is not found within the corresponding soft element range of the string unit starting from S1 or S2, the system reports a string data or length error.
- When n is greater than the number of characters in S2, the system reports an illegal operand error.
- When S1 is a specified string, a maximum of 32 characters are allowed. Commas and double quotes represent separators in the host computer software, so these characters cannot be recognized by the host computer software and are not counted towards the number of characters.
- When S1 is an empty string ("00H"), the detection result is the position of "00H" in the S2 string unit (if S2 has an even number of characters, the result is the first "00H" position).

Application Example



		Element Name	Data Type	Display Format	Current Value
1	...	D10	WORD	Hexadecimal	16#3231
2	...	D11	WORD	Hexadecimal	16#3433
3	...	D12	WORD	Hexadecimal	16#3635
4	...	D13	WORD	Hexadecimal	16#37
5	...		WORD	Hexadecimal	
6	...	D100	WORD	Decimal	4

In case of M506=ON, the command retrieves the same character as "45" starting from the 2nd character of the string unit starting from D10, and stores the result in the D100 unit. See the figure below for the process.



3.19.9 STRMOV: Commands for String Transfer

Command table		STRMOV (S) (D)			Applicable model	TS600 series		
16-Bit command		STRMOV: String transfer						
32-Bit command		-						
Operand	Type	Bit			Word		Indexing	Constant
		X, Y, M, LM, T, C, S	Dx.y	Custom bit variable	D, R, V, Z, T, C	Custom word variable		
S	INT/ Array*number	-	-	-	√[1]	√	√	√[2]
D	INT/ Array*number	-	-	-	√[1]	√	√	-

Remark:

[1] The Z element is not supported.

[2] Here it represents a string constant.

Operand Description

S: The source string unit.

D: The destination unit.

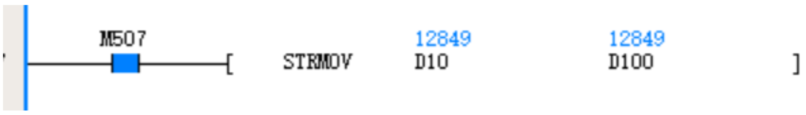
Function Description

1. The command transfers all data (including "00H") of the S string unit to the element units starting from D.
2. The valid data of a string unit is the data from the specified soft element of the string unit to the position where the first "00H" is detected.

Precautions

- When "00H" is not found within the corresponding soft element range of the string unit starting from S, the system reports a string data or length error.
- When the number of characters in the S string unit is even, the low byte stores "00H", while both high and low bytes at the corresponding positions in D stores "00H".
- When S1 is a specified string, a maximum of 32 characters are allowed. Commas and double quotes represent separators in the host computer software, so these characters cannot be recognized by the host computer software.

Application Example



		Element Name	Data Type	Display Format	Current Value
1	...	D10	WORD	Hexadecimal	16#3231
2	...	D11	WORD	Hexadecimal	16#3433
3	...	D12	WORD	Hexadecimal	16#3635
4	...	D13	WORD	Hexadecimal	16#37
5	...		WORD	Hexadecimal	
6	...	D100	WORD	Decimal	4

In case of M507=ON, the command transfers the string data starting from D10 to the unit starting from D100. See the figure below for the process.

	B15---B8	B7---B0
D10	0x32	0x31
D11	0x34	0x33
D12	0x36	0x35
D13	0x00	0x00

	B15---B8	B7---B0
D100	0x32	0x31
D101	0x34	0x33
D102	0x36	0x35
D103	0x00	0x00

3.20 Data Processing Command

3.20.1 Command Table

Command Category	Name	Function
Data Processing Command	*WTOB	Data separation of byte unit
	BTO*W	Data combination of byte unit
	UNI	4-bit combination of 16-bit data
	DIS	4-bit separation of 16-bit data
	ANS	Signal alarm set
	ANR	Signal alarm reset

3.20.2 WTOB: Commands for Data Separation of Byte Unit

Command table		WTOB (S) (D) (n)			Applicable model	TS600 series		
16-Bit command		WTOB: Data separation of byte unit						
32-Bit command		-						
Operand	Type	Bit			Word		Indexing	Constant
		X, Y, M, LM, T, C, S	Dx.y	Custom bit variable	D, R, V, Z, T, C	Custom word variable		
S	WORD/ DWORD, Array*n/2	-	-	-	√[1]	√	√	-
D	WORD/ DWORD, Array*n	-	-	-	√[1]	√	√	-
n	WORD	-	-	-	√[2]	√	√	√

Remark:

[1]The Z and V soft elements are not supported.

[2]Only the D and R soft elements are supported.

Operand Description

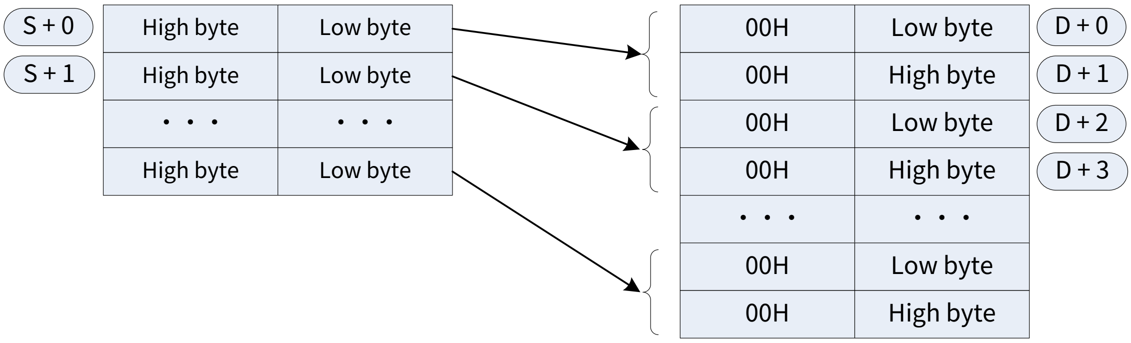
S: The starting number of the soft element that stores the data to be separated according to byte units.

D: The starting number of the soft element that stores the result already separated according to byte units.

n: The number of byte data to be separated (0≤n≤ 256).

Function Description

The commands separates the 16-bit data stored in the n/2 soft elements starting from S into n bytes, stores them to the low bytes of the n soft elements starting from D, and resets the high bytes to zero.

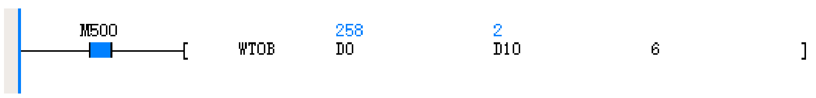


Note: When n is an odd, only the low byte (8 bits) is the object data in the last separated data.

Precautions

- In case of n=0, the command is not executed; in case of n>256 or n<0, the system reports an illegal operand error.
- The source and destination operands cannot overlap with each other, otherwise the system reports an overlapping operand error.

Application Example



		Element Name	Data Type	Display Format	Current Value
1	...	D0	INT	Hexadecimal	16#102
2	...	D1	INT	Hexadecimal	16#304
3	...	D2	INT	Hexadecimal	16#506
4	...		WORD	Decimal	
5	...	D10	INT	Hexadecimal	16#2
6	...	D11	INT	Hexadecimal	16#1
7	...	D12	INT	Hexadecimal	16#4
8	...	D13	INT	Hexadecimal	16#3
9	...	D14	INT	Hexadecimal	16#6
10	...	D15	INT	Hexadecimal	16#5

In case of M500=ON, the command separates the data of the 3 units starting from D0 into 6 units according to high and low bytes, and stores the results in the 6 units starting from D10. In case of D0=0x102, D1=0x304, and D2=0x506, the obtained results are D10=0x02, D11=0x01, D12=0x04, D13=0x03, D14=0x06, and D15=0x05.

3.20.3 BTOW: Commands for Data Combination of Byte Unit

Command table		BTOW (S) (D) (n)			Applicable model	TS600 series		
16-Bit command		BTOW: Data Separation of byte unit						
32-Bit command		-						
Operand	Type	Bit			Word		Indexing	Constant
		X, Y, M, LM, T, C, S	Dx.y	Custom bit variable	D, R, V, Z, T, C	Custom word variable		
S	WORD/ DWORD, Array*n	-	-	-	√[1]	√	√	-
D	WORD/ DWORD, Array*n/2	-	-	-	√[1]	√	√	-
n	WORD	-	-	-	√[2]	√	√	√

Remark:

[1]The Z and V soft elements are not supported.

[2]Only the D and R soft elements are supported.

Operand Description

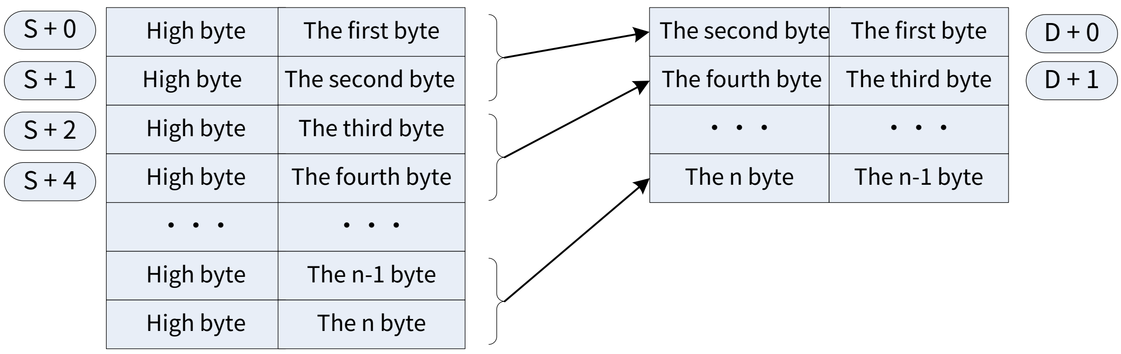
S: The starting number of the soft element that stores the data to be combined according to byte units.

D: The starting number of the soft element that stores the result already combined according to byte units.

n: The number of byte data to be separated (0⩽n⩽ 256).

Function Description

The command combines the low bytes (8 bits) of the n 16/32-bit data starting from S, and then stores the resulting 16-bit/32-bit data in a small end manner to the n/2 soft elements starting from D. The (8-bit) high bytes (after S) of the 16-bit data to be combined are ignored.



Note: When n is an odd number, the last combined high byte is reset to zero.

Precautions

- In case of n=0, the command is not executed; in case of n>256 or n<0, the system reports an illegal operand error.
- The source and destination operands cannot overlap with each other, otherwise the system reports an overlapping operand error.

Application Example

		Element Name	Data Type	Display Format	Current Value
1	...	D0	WORD	Hexadecimal	16#1
2	...	D1	WORD	Hexadecimal	16#2
3	...	D2	WORD	Hexadecimal	16#3
4	...	D3	WORD	Hexadecimal	16#4
5	...	D4	WORD	Hexadecimal	16#5
6	...	D5	WORD	Hexadecimal	16#6
7	...		WORD	Decimal	
8	...	D10	WORD	Hexadecimal	16#201
9	...	D11	WORD	Hexadecimal	16#403
10	...	D12	WORD	Hexadecimal	16#605

In case of M502=ON, the command combines the data of the 6 units starting from D0 into the data of 3 units, and stores the results in the 3 units starting from D10. In case of D0=0x01, D1=0x02, and D2=0x03, the obtained results are D3=0x04, D4=0x05, D5=0x06, D10=0x201, D11=0x403, and D12=0x605.

3.20.4 UNI: Commands for 4-Bit Combination of 16-Bit Data

Command table		UNI (S) (D) (n)			Applicable model	TS600 series		
16-Bit command		UNI: 4-bit combination of 16-bit data						
32-Bit command		-						
Operand	Type	Bit			Word		Indexing	Constant
		X, Y, M, LM, T, C, S	Dx.y	Custom bit variable	D, R, V, Z, T, C	Custom word variable		
S	WORD, Array*n	-	-	-	√[1]	√	√	-
D	WORD	-	-	-	√[1]	√	√	-
n	WORD	-	-	-	√[2]	√	√	√

Remark:

[1]The Z and V soft elements are not supported.

[2]Only the D and R soft elements are supported.

Operand Description

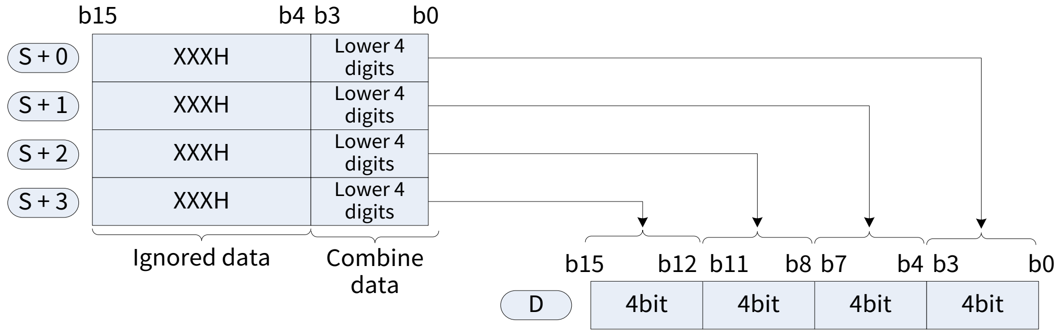
S: The starting number of the soft element that stores the data to be combined.

D: The number of the soft element that stores the data already combined.

n: The number of combinations (which is between 0 and 4; in case of n=0, the command performs no action).

Function Description

The command combines the low 4 bits of the 16-bit data of the *n* points starting from *S* into 16 bit data, and then stores the result to *D* in a small end manner.



Note: When n is between 1 and 3, the remaining low bits of D are padded with 0.

Precautions

- In case of n=0, the command is not executed; in case of n>4 or n<0, the system reports an illegal operand error.
- The source and destination operands cannot overlap with each other, otherwise the system reports an overlapping operand error.

Application Example

		Element Name	Data Type	Display Format	Current Value
1	...	D0	WORD	Hexadecimal	16#1
2	...	D1	WORD	Hexadecimal	16#2
3	...	D2	WORD	Hexadecimal	16#3
4	...	D3	WORD	Hexadecimal	16#4
5	...		WORD	Hexadecimal	
6	...	D10	WORD	Hexadecimal	16#4321

In case of M504=ON, the command combines the low 4 bits of the data of the 4 units starting from D0, and then stores the result in D10. In case of D0=0x01, D1=0x02, D2=0x03, D3=0x04, the obtained result is D10=0x4321.

3.20.5 DIS: Commands for 4-Bit Separation of 16-Bit Data

Command table		DIS (S) (D) (n)			Applicable model	TS600 series		
16-Bit command		DIS: 4-bit separation of 16-bit data						
32-Bit command		-						
Operand	Type	Bit			Word		Indexing	Constant
		X, Y, M, LM, T, C, S	Dx.y	Custom bit variable	D, R, V, Z, T, C	Custom word variable		
S	WORD	-	-	-	√[1]	√	√	-
D	WORD, Array*n	-	-	-	√[1]	√	√	-
n	WORD	-	-	-	√[2]	√	√	√

Remark:

[1]The Z and V soft elements are not supported.

[2]Only the D and R soft elements are supported.

Operand Description

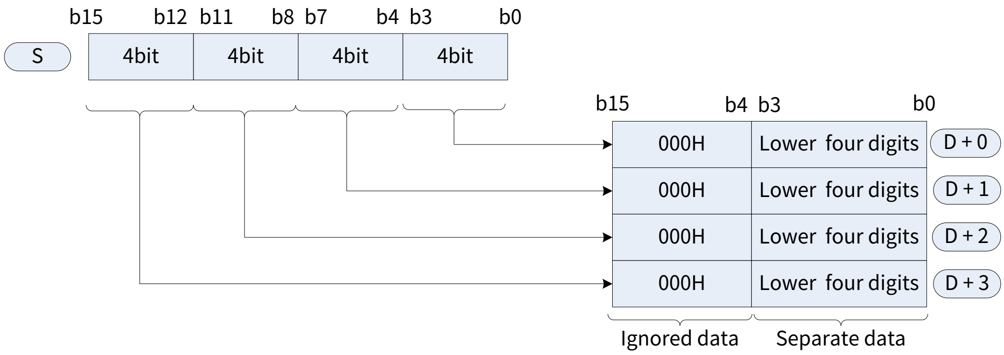
S: The starting number of the soft element that stores the data to be separated.

D: The number of the soft element that stores the data already separated.

n: The number of separations (which is between 0 and 4; in case of n=0, the command performs no action).

Function Description

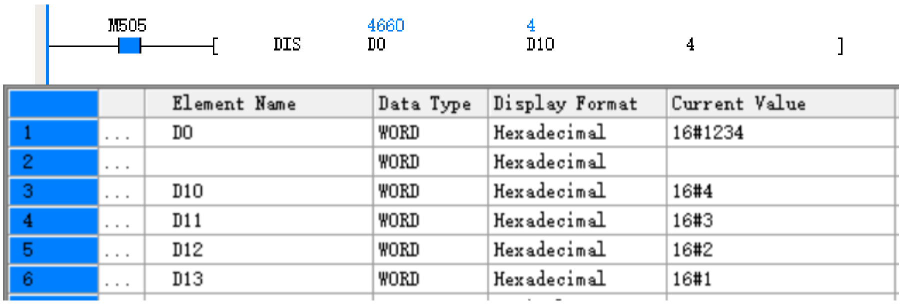
The command separates the 16-bit data of S with every 4 bits as a unit, and then stores the results to the low 4 bits of the n soft elements starting from D, with the high 12 bits padded with 000H.



Precautions

- In case of n=0, the command is not executed; in case of n>4 or n<0, the system reports an illegal operand error.
- The source and destination operands cannot overlap with each other, otherwise the system reports an overlapping operand error.
- The high 12 bits of the n soft elements starting from D are reset to zero.

Application Example



In case of M505=ON, the command separates the data of the D0 unit per 4 bits, and then stores the results in the 4 units starting from D10. In case of D0=0x1234, the obtained results are D10=0x04, D11=0x03, D12=0x02, and D13=0x01.

3.20.6 ANS: Commands for Signal Alarm Set

<table>
<tr><td colspan="2">Command table</td><td colspan="3">ANS (S1) (S2) (D)</td><td>Applicable model</td><td colspan="3">TS600 series</td></tr>
<tr><td colspan="2">16-Bit command</td><td colspan="7">ANS: Signal alarm set</td></tr>
<tr><td colspan="2">32-Bit command</td><td colspan="7">-</td></tr>
<tr><td rowspan="2">Operand</td><td rowspan="2">Type</td><td colspan="3">Bit</td><td colspan="2">Word</td><td rowspan="2">Indexing</td><td rowspan="2">Constant</td></tr>
<tr><td>X, Y, M, LM, T, C, S</td><td>Dx.y</td><td>Custom bit variable</td><td>D, R, V, Z, T, C</td><td>Custom word variable</td></tr>
<tr><td>S1</td><td>INT</td><td>-</td><td>-</td><td>-</td><td>√[1]</td><td>-</td><td>√</td><td>-</td></tr>
<tr><td>S2</td><td>WORD</td><td>-</td><td>-</td><td>-</td><td>√[2]</td><td>-</td><td>√</td><td>√</td></tr>
<tr><td>D</td><td>BOOL</td><td>√[3]</td><td>-</td><td>-</td><td>-</td><td>-</td><td>-</td><td>-</td></tr>
</table>

Remark:

[1]Only the T element is supported.

[2]Only the D and R soft elements are supported.

[3]Only the S soft element is supported.

Operand Description

S1: The number of the timer which judges the time, only applying to the 100ms timer, and ranging between T0 and T199.

S2: The data used to judge the time (ranging between 1 and 32767).

D: The set signal alarm soft element, ranging between S900 and S999.

Function Description

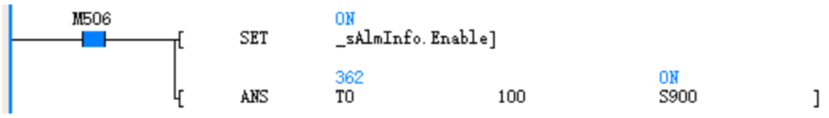
When the duration of the energy flow is greater than n, D is set; when the duration of the energy flow is less than n, the timer S is reset, and D is reset; when the energy flow is invalid, S is reset.

System variables	Name	Function
_sAlmInfo.Enable	Signal alarm enable	After Enable=ON, alarm enable acts
_sAlmInfo.ActFlg	Signal alarm act	When any of states S900~S999 acts, ActFlg=ON is set
_sAlmInfo.MinNum	Minimum number of ON state	It stores the smallest number of active alarms among S900~S999

Precautions

- When the timer number is greater than 199, the system reports an operand error.
- When the set signal alarm soft element is not between S900 and S999, the system reports an operand error.

Application Example



		Element Name	Data Type	Display Format	Current Value
1	...	T0	WORD	Decimal	233
2	...	S900	BOOL	Binary	ON
3	...		WORD	Decimal	
4	...	_sAlmInfo	_stru_ALM_INFO	Decimal	
5		_sAlmInfo.Enable	BOOL	Decimal	ON
6		_sAlmInfo.ActFlg	BOOL	Decimal	ON
7		_sAlmInfo.MinNum	INT	Decimal	900

In case of M506=ON, the alarm is enabled (that is, Enable=ON). After 10 seconds, S900 is set, the alarm action flag bit is ActFlg=ON, and the minimum recorded alarm value is MinNum=900.

3.20.7 ANR: Command for Signal Alarm Reset

Command table		ANR			Applicable model	TS600 series		
16-Bit command		ANR: Signal alarm reset						
32-Bit command		-						
Operand	Type	Bit			Word		Indexing	Constant
		X, Y, M, LM, T, C, S	Dx.y	Custom bit variable	D, R, V, Z, T, C	Custom word variable		
-	-	-	-	-	-	-	-	-

Operand Description

S1: The number of the timer which judges the time, only applying to the 100ms timer, and ranging between T0 and T199.

D: The set signal alarm soft element, ranging between S900 and S999.

S2: The data used to judge the time (ranging between 1 and 32767).

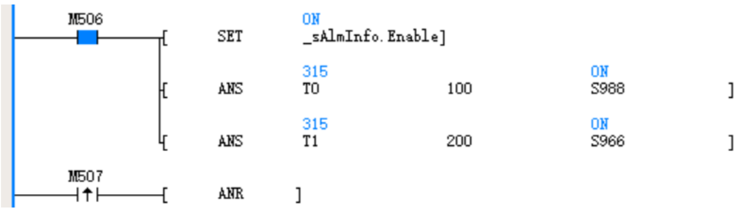
Function Description

When the energy flow is valid, the command resets the operating states of the signal alarms S900~S999; if there are multiple states acts, the command resets the one with the smallest number. When the energy flow becomes valid again, the command resets the next one with the smallest number.

System variables	Name	Function
_sAlmInfo.Enable	Signal alarm enable	After Enable=ON, alarm enable acts
_sAlmInfo.ActFlg	Signal alarm act	When any of states S900~S999 acts, ActFlg=ON is set
_sAlmInfo.MinNum	Minimum number of ON state	It stores the smallest number of active alarms among S900~S999

Application Example

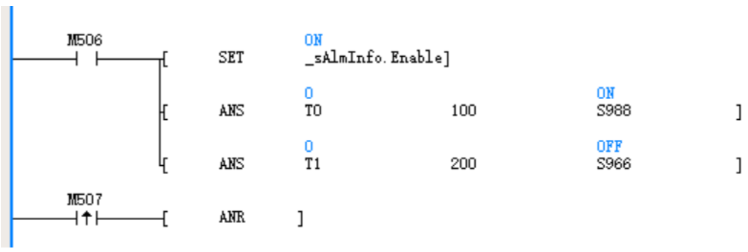
Before using the command:



		Element Name	Data Type	Display Format	Current Value
1	...	T0	WORD	Decimal	309
2	...	S900	BOOL	Binary	OFF
3	...		WORD	Decimal	
4	...	_sAlmInfo	_stru_ALM_INFO	Decimal	
5		_sAlmInfo. Enable	BOOL	Decimal	ON
6		_sAlmInfo. ActFlg	BOOL	Decimal	ON
7		_sAlmInfo. MinNum	INT	Decimal	966

In case of M506=ON, alarm enable (Enable) is set, the alarm flag bits S988 and S966 are set after 10 seconds and 20 seconds, respectively, the alarm action flag bit ActFlg is set, and set the minimum recorded alarm value is MinNum=966.

After using the command:



		Element Name	Data Type	Display Format	Current Value
1	...	T0	WORD	Decimal	196
2	...	S900	BOOL	Binary	OFF
3	...		WORD	Decimal	
4	...	_sAlmInfo	_stru_ALM_INFO	Decimal	
5		_sAlmInfo. Enable	BOOL	Decimal	ON
6		_sAlmInfo. ActFlg	BOOL	Decimal	ON
7		_sAlmInfo. MinNum	INT	Decimal	988

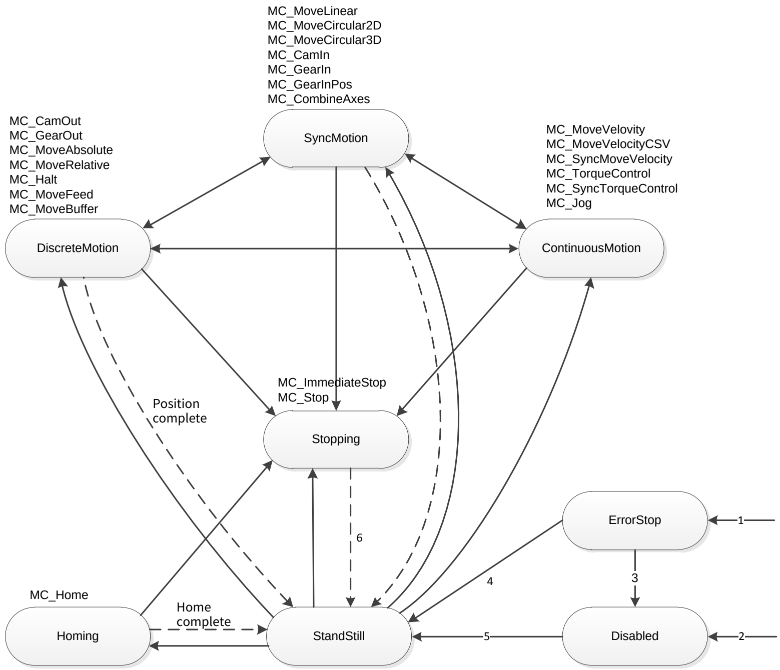
In case of M507 is ON, the minimum alarm flag bit S966 is reset, and the minimum recorded alarm value is MinNum=988.

3.21 MC Axis Control (EtherCAT & Pulse Output Commands)

3.21.1 Command Table

Command	Name
MC_SetAxisConfigPara	Axis configuration parameter
MC_Power	Axis enable
MC_Reset	Axis reset
MC_ReadStatus	Axis read status instruction
MC_ReadAxisError	Read axis error
MC_ReadDigitalInput	Read digital input instruction
MC_ReadPosition	Read actual position instruction
MC_ReadVelocity	Read actual velocity instruction
MC_SetPosition	Set position instruction
MC_MoveAbsolute	Absolute positioning instruction
MC_MoveRelative	Relative positioning instruction
MC_MoveVelocity	Velocity
MC_Jog	Continuous operation instruction
MC_Home	Home instruction
MC_SetOverride	Set override instruction
MC_Stop	Stop instruction
MC_Halt	Halt instruction
MC_ImmediateStop	Immediate stop instruction
MC_MoveSuperImposed	Motion Superimposed instruction
MC_TouchProbe	Probe instruction
MC_MoveFeed	Interrupt fixed-length instruction
MC_MoveBuffer	Multi-segment positioning
MC_MoveVelocityCSV	CSV-based velocity instruction with adjustable pulse width
MC_SyncMoveVelocity	Synchronized velocity supporting PWM waveform, based on CSV
MC_FollowPosition	Synchronized position based on CSP mode
MC_FollowVelocity	Synchronized position based on CSP mode
MC_SyncTorqueControl	Synchronous torque control instruction
MC_ReadActualTorque	Read actual torque instruction
MC_CamIn	Cam action start
MC_CamOut	Cam out
MC_DigitalCamSwitch	Tappet
MC_GenerateCamTable	Update cam table
MC_GetCamtableDistance	Get cam table displacement
MC_GetCamtablePhase	Get cam table phase (reserved)
MC_GearIn	Gear in
MC_GearOut	Gear out
MC_Phasing	Master-slave axis phase offset
MC_CombineAxes	Addition-subtraction operation positioning
MC_MoveLiear	Linear interpolation instruction
MC_MoveCircular2D	Plane arc interpolation
MC_MoveEllipse	Plane ellipse interpolation (reserved)
MC_GroupSetOverRide	Axis group velocity regulation
MC_GroupStop	Axis group stop
MC_GroupHalt	Axis group halt instruction
MC_GroupImmediateStop	Axis group immediate stop instruction
MC_ReadGroupVelocity	Read composite axis group velocity

3.21.2 Axis State Machines



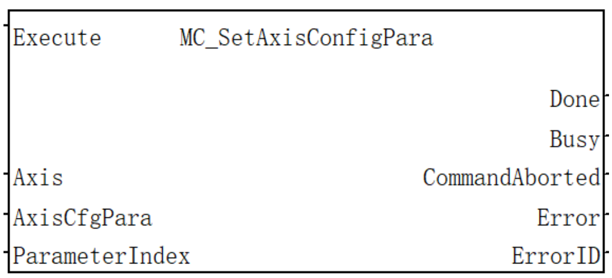
State Machine Description

State	Function Description
Disabled	Disabled state
ErrorStop	Stop due to fault
Standstill	Enabled state
Homing	Home
Stopping	Stop
Discrete Motion	Discretely move
Continuous Motion	Continuously move
Synchronized Motion	Synchronized motion state

Conversion	Conversion Condition
1	When the fault detection logic of the axis detects a fault
2	When there is no fault with the axis and the energy flow of MC_Power is OFF
3	When MC_Reset is called to reset axis failure and MC_Power energy flow is OFF
4	When MC_Reset is called to reset axis failure and MC_Power energy flow is ON
5	When the energy flow of MC_Power is ON and the output flag Status is ON
6	When MC_Stop(MC_ImmediateStop).Done=ON and the energy flow of the graphic block is OFF

3.21.3 MC_SetAxisParaAxis

Graphic Block



16-Bit command	MC_SetAxisConfigPara: Axis Configuration Parameter					
32-Bit command	-					
Operand	**Name**	**Description**	**Nullable**	**Default value**	**Range**	**Data Type**
S1	Axis	Axis name/axis ID	No	-	-	WORD
S2	AxisCfgPara	Custom axis parameter	No	0	-	_stru_AXIS_CFG
S3	ParameterIndex	Parameter ID	No	0	Positive/ negative/0	INT
D1	Done	Command execution completion	Yes	OFF	ON/OFF	BOOL
D2	Busy	Ongoing execution flag	Yes	OFF	ON/OFF	BOOL
D3	CommandAborted	Execution interrupt flag	Yes	OFF	ON/OFF	BOOL
D4	Error	Error flag	Yes	OFF	ON/OFF	BOOL
D5	ErrorID	Error code	Yes	0	-	WORD

Operand	Const	Y	M	S	D	R	Custom Variable
S1	√	-	-	-	-	-	-
S2	-	-	-	-	-	-	√
S3	-	-	-	-	√	√	√
D1	-	√	√	√	-	-	√
D2	-	√	√	√	-	-	√
D3	-	√	√	√	-	-	√
D4	-	√	√	√	-	-	√
D5	-	-	-	-	√	√	√

Function Description

This command is used to modify the configuration parameters of the axis. If the configuration parameters meet the requirements, the command modifies the configuration parameters again. After the configuration is completed, the Done signal output is valid. If they don’ t meet the requirements, the command reports an error.

AxisCfgPara represents the axis configuration parameter that the user expects to modify.

ParameterIndex is used to indicate the range of modified parameters. See below for the allowed set values:

Parameter #-1: It updates all parameters and allows modification in the Disabled state. After modification, the current position may change suddenly and therefore requires the re-execution of the homing operation.

Parameter #0: It does not update any parameter.

Parameter #100: It only updates the gear ratio and allows modification in the Disabled state. After modification, the current position may change suddenly and therefore requires the re-execution of the homing operation.

Variable	Data Type	Function Description	Valid parameter range
dwPulseData	DWORD	Number of pulses for one revolution of motor/encoder	Positive number
fDistanceData	REAL	The amount of movement when the worktable rotates for one revolution	Positive number
diGearRatioNum	DINT	Numerator of the gear ratio	Positive number
dwGearRatioDen	DWORD	Denominator of the gear ratio	Positive number

Parameter #200: It only modifies the positive and negative soft limits and allows modification in the Disabled and Standstill states.

Variable	Data Type	Function Description	Valid parameter range
bSWLimitEnable	BOOL	Soft limit enable control OFF: invalid; ON: valid	ON/OFF
fMaxPLimit	REAL	Positive limit value in linear mode	Positive/0/negative
fMaxNLimit	REAL	Negative limit value in linear mode	Positive/0/negative

Parameter #300: It only modifies the linear/rotary mode and allows modification in the Disabled state. After modification, the current position may change suddenly and therefore requires the re-execution of the homing operation.

Variable	Data Type	Function Description	Valid parameter range
iLineRotateMode	INT	Linear/rotation mode selection 0: linear mode; 1: rotary mode	0~1
fRotation	REAL	Number of rotation cycles in rotary mode	Positive number

Parameter #400: It only modifies the homing mode and allows modification in the Disabled and Standstill states.

Variable	Data Type	Function Description	Valid parameter range
iHomeMode	INT	Homing mode	Positive number
bHomeDirection	BOOL	Homing direction	Positive number
fMaxHomeSpeed	REAL	Maximum axis homing speed limit	Positive number
fMaxHomeAcc	REAL	Maximum axis homing acceleration limit	Positive number
fDecModuleSpeed	REAL	Maximum speed on deceleration module when axis homes	Positive number
fWaitZSpeed	REAL	Maximum speed while waiting for Z signal when axis homes	Positive number

Parameter #500: It only modifies the hard limit, origin signal, and Z signal and allows modification in the Disabled and Standstill states.

Variable	Data Type	Function Description	Valid parameter range
bHWPLimitEnable	BOOL	Hardware positive limit enable signal	ON/OFF
iHWPLimitID	INT	Hardware positive limit terminal ID	0~15
bHWNLimitEnable	BOOL	Hardware negative limit enable signal	ON/OFF
iHWNLimitID	INT	Hardware negative limit terminal ID	0~15
bHomeSignal	BOOL	Home enable signal	ON/OFF
iHomeSignalID	INT	Home signal terminal ID	0~15
bZSignal	BOOL	Z signal enable signal	ON/OFF
iZSignalID	INT	Z signal terminal ID	0~15

Parameter #600: It only updates the pulse output mode and allows modification in the Disabled state. After modification, the current position requires the re-execution of the homing operation.

Variable	Data Type	Function Description	Valid parameter range
iPulseMode	INT	Pulse axis control mode 0: pulse+direction; 1: forward-reverse pulse train 2: orthogonal coding pulse; 3: PWM wave mode	0~3

Parameter #700: It only updates the virtual axis mode and allows modification in the Disabled state. After modification, the current position may change suddenly and therefore requires the re-execution of the homing operation.

Variable	Data Type	Function Description	Valid parameter range
bVirtualMode	BOOL	Virtual axis mode OFF: virtual axis mode invalid; ON: virtual axis mode valid	ON/OFF

Parameter #800: It only modifies the probe signal and allows modification in the Disabled state. After modification, the current position may change suddenly and therefore requires the re-execution of the homing operation.

Variable	Data Type	Function Description	Valid parameter range
bTouchProbeID1	BOOL	Probe terminal 1 enable signal	ON/OFF
iTouchProbeID1	INT	Probe terminal 1 ID	0~15
bTouchProbeID2	BOOL	Probe terminal 2 enable signal	ON/OFF
iTouchProbeID2	INT	Probe terminal 2 ID	0~15

Parameter #900: It only modifies the software limit variables and negative soft limits and allows modification in the Disabled and Standstill states.

Variable	Data Type	Function Description	Valid parameter range
fAxisErrorDec	REAL	Axis error deceleration	Positive number
fMaxVelocity	REAL	Maximum axis velocity limit	Positive number
fMaxAcceleration	REAL	Maximum axis acceleration limit	Positive number
fMaxDeceleration	REAL	Maximum axis deceleration limit	Positive number
fMaxJerk	REAL	Maximum axis jerk limit	Positive number
fMaxJogSpeed	REAL	Maximum speed of axis in Jog mode	Positive number
fMaxPTorque	REAL	Maximum positive torque (fieldbus servo axis)	Positive number
fMaxNTorque	REAL	Maximum negative torque (fieldbus servo axis)	Positive number

Parameter #1000: It only updates the pulse servo control signal and allows modification in the Disabled state. After modification, the current position may change suddenly and therefore requires the re-execution of the homing operation.

Variable	Data Type	Function Description	Valid parameter range
bServoError	BOOL	Servo alarm enable signal	ON/OFF
iServoErrorID	INT	Servo alarm terminal ID	0~15
bServoEnable	BOOL	Servo enable signal	ON/OFF
iServoEnableID	INT	Servo enable terminal ID	0~15
bClearError	BOOL	Clear servo alarm enable signal	ON/OFF
iClearErrorID	INT	Clear servo alarm terminal ID	0~15

Resetting This Command

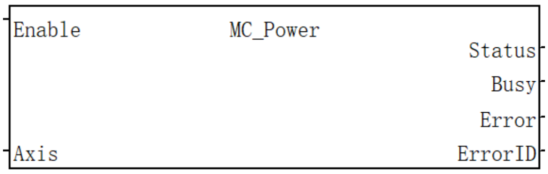
Resetting this command enables the modification of axis parameters again.

Multiple Calls of This Command

When this command is being executed, it is not allowed to call the second MC_SetAxisConfigPara command, otherwise the latter reports an error. Only after the first command is completed can the second command be executed.

3.21.4 MC_Power

Graphic Block



16-Bit command	MC_Power: Axis Enable					
32-Bit command	-					
Operand	**Name**	**Description**	**Nullable**	**Default value**	**Range**	**Data Type**
S1	Axis	Axis name/axis ID	No	-	-	WORD
D1	Status	Axis enable flag	Yes	OFF	ON/OFF	BOOL
D2	Busy	Busy flag	Yes	OFF	ON/OFF	BOOL
D3	Error	Command error flag	Yes	OFF	ON/OFF	BOOL
D4	ErrorID	Error code	Yes	0	-	WORD

Operand	Const	Y	M	S	D	R	Custom Variable
S1	√	-	-	-	-	-	
D1	-	√	√	√	-	-	√
D2	-	√	√	√	-	-	√
D3	-	√	√	√	-	-	√
D4	-	-	-	-	√	√	√

Function Description

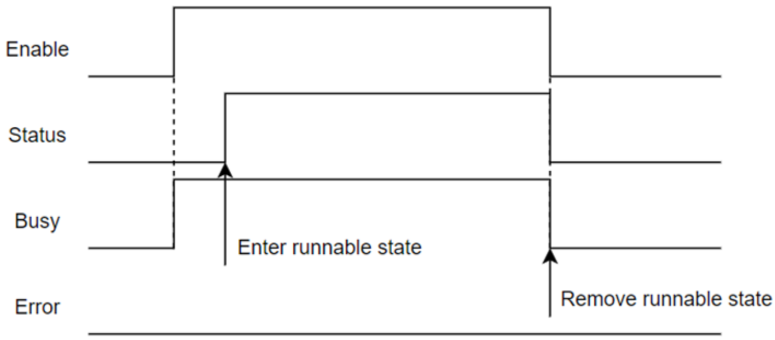
1. The MC_Power command is applicable to the local pulse axis and bus servo axis, used to set the enable state of these axes, and valid at high levels.
2. When the command sets Enable to ON, the axis enters the enable state, the Status signal of the command is valid, and the PLCOpen state machine of the axis switches from the Disabled state to the StandStill state.
3. During the axis operation, if the command sets Enable to FALSE, the axis is disabled, and the motion-related commands (such as MC-MoveAbsolute) stop. The disabled state of the axis triggers the motion-related commands to become invalid.
4. When the axis malfunctions and enters the errorstop state, enabling MC_Power again or calling MC_Reset can switch the axis to the Standstill state.

Multiple Starts of This Command

When you are using multiple MC_Power commands, the control energy flow of the last executed MC_Power command within one cycle shall prevail.

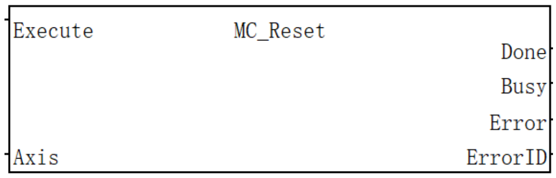
Timing diagram

Calling the MC_Power command enables the axis normally.



3.21.5 MC_Reset

Graphic Block



16-Bit command	MC_Reset: Axis Reset					
32-Bit command	-					
Operand	Name	Description	Nullable	Default value	Range	Data Type
S1	Axis	Axis name/axis ID	No	-	-	WORD
D1	Done	Axis enable flag	Yes	OFF	ON/OFF	BOOL
D2	Busy	Busy flag	Yes	OFF	ON/OFF	BOOL
D3	Error	Command error flag	Yes	OFF	ON/OFF	BOOL
D4	ErrorID	Error code	Yes	0	-	WORD

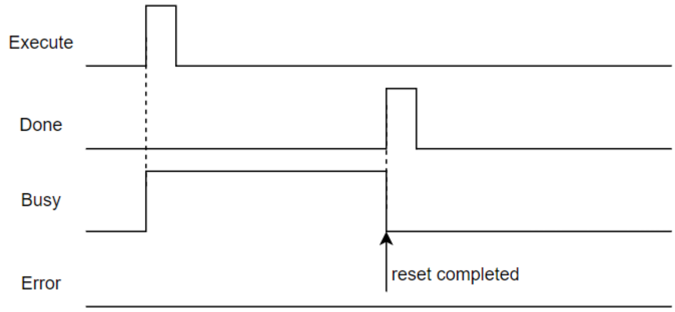
Operand	Const	Y	M	S	D	R	Custom Variable
S1	√	-	-	-	-	-	-
D1	-	√	√	√	-	-	√
D2	-	√	√	√	-	-	√
D3	-	√	√	√	-	-	√
D4	-	-	-	-	√	√	√

Function Description

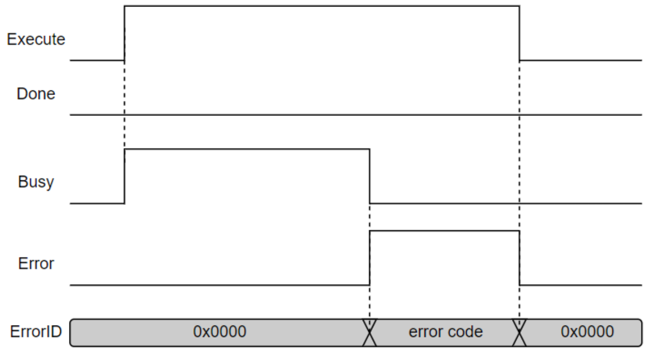
1. The MC_ Reset command is used to reset the fault of the axis, and is valid to the rising edge.
2. After a successful reset, if the drive is in the enabled state, the PLCOpen state machine of the axis enters the StandStill state; if the driver is not enabled, the state machine enters the Disabled state.
3. This command does not have an interrupt output signal and therefore cannot be interrupted during execution.
4. During the axis operation, if this command is triggered, the system reports an error.

Timing diagram

- When the axis malfunctions, calling the MC_Reset command successfully resets the axis fault.

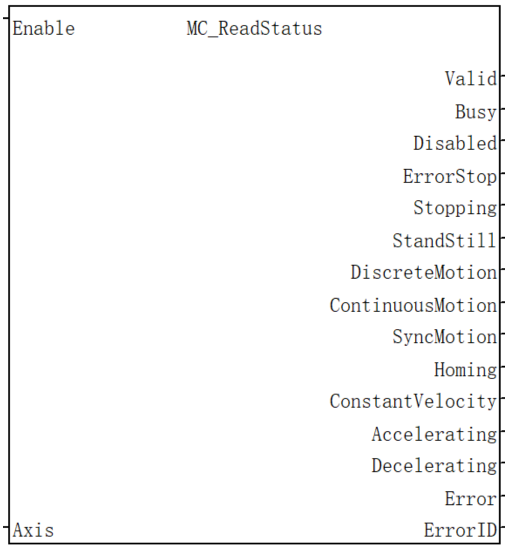


- When the driver experiences a non-resettable fault.



3.21.6 MC_ReadStatus

Graphic Block



16-Bit command	MC_ ReadStatus: Read Axis Status					
32-Bit command	-					
Operand	**Name**	**Description**	**Nullable**	**Default value**	**Range**	**Data Type**
S1	Axis	Axis name/axis ID	No	-	-	WORD
D1	Valid	Valid flag	Yes	OFF	ON/OFF	BOOL
D2	Busy	Run flag	Yes	OFF	ON/OFF	BOOL
D3	Disabled	Disabled flag	Yes	OFF	ON/OFF	BOOL
D4	ErrorStop	Fault message	Yes	OFF	ON/OFF	BOOL
D5	Stopping	Stop	Yes	OFF	ON/OFF	BOOL
D6	Standstill	Run	Yes	OFF	ON/OFF	BOOL
D7	DiscreteMotion	Discretely move	Yes	OFF	ON/OFF	BOOL
D8	ContinuousMotion	Continuously move	Yes	OFF	ON/OFF	BOOL
D9	SyncMotion	Synchronously move	Yes	OFF	ON/OFF	BOOL
D10	Homing	Home	Yes	OFF	ON/OFF	BOOL
D11	ConstantVelocity	The axis velocity is 0 or the axis moves at a constant velocity	Yes	OFF	ON/OFF	BOOL
D12	Accelerating	The axis is moving with an acceleration	Yes	OFF	ON/OFF	BOOL
D13	Decelerating	The axis is moving with a deceleration	Yes	OFF	ON/OFF	BOOL
D14	Error	Command error flag	Yes	OFF	ON/OFF	BOOL
D15	ErrorID	Error code	Yes	0	-	WORD

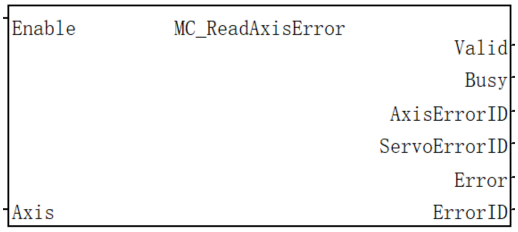
Operand	Const	Y	M	S	D	R	Custom Variable
S1	√	-	-	-	-	-	-
D1	-	√	√	√	-	-	√
D2	-	√	√	√	-	-	√
D3	-	√	√	√	-	-	√
D4	-	√	√	√	-	-	√
D5	-	√	√	√	-	-	√
D6	-	√	√	√	-	-	√
D7	-	√	√	√	-	-	√
D8	-	√	√	√	-	-	√
D9	-	√	√	√	-	-	√
D10	-	√	√	√	-	-	√
D11	-	√	√	√	-	-	√
D12	-	√	√	√	-	-	√
D13	-	√	√	√	-	-	√
D14	-	√	√	√	-	-	√
D15	-	-	-	-	√	√	√

Function Description

1. This command is used to read the states of the PLCOpen state machine, as well as the accelerating and decelerating states, of the axis, and is valid at high levels.
2. This command has no interrupt flag and therefor multiple commands can run simultaneously.

3.21.7 MC_ReadAxisError

Graphic Block



16-Bit command	-
32-Bit command	MC_ ReadAxisError: Read Axis Error

Operand	Name	Description	Nullable	Default value	Range	Data Type
S1	Axis	Axis name/axis ID	No	-	-	WORD
D1	Valid	Valid flag	Yes	OFF	ON/OFF	BOOL
D2	Busy	Ongoing execution flag	Yes	OFF	ON/OFF	BOOL
D3	AxisErrorID	Axis fault code	Yes	0	-	DINT
D4	ServoErrorID	Servo fault code	Yes	0	-	WORD
D5	Error	Command error flag	Yes	OFF	ON/OFF	BOOL
D6	ErrorID	Error code	Yes	0	-	WORD

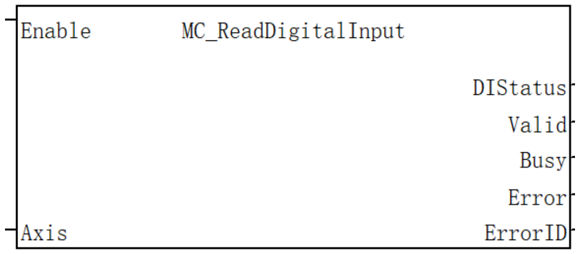
Operand	Const	Y	M	S	D	R	Custom Variable
S1	√	-	-	-	-	-	-
D1	-	√	√	√	-	-	√
D2	-	√	√	√	-	-	√
D3	-	-	-	-	√	√	√
D4	-	-	-	-	√	√	√
D5	-	√	√	√	-	-	√
D6	-	-	-	-	√	√	√

Function Description

1. The MC_ReadAxisError command is used to read bus axis or local pulse axis faults, and is valid at high levels.
2. AxisErrorID is used to display the error of the corresponding function block when the axis status is ErrorStop, while ServoErrorID corresponds to the value (0x603f) of the PDO parameter of the bus driver and is used to display the fault code when the servo driver fails.
3. This command has no interrupt flag and therefor multiple commands can run simultaneously.

3.21.8 MC_ReadDigitalInput

Graphic Block



16-Bit command	-					
32-Bit command	MC_ ReadDigitalInput: Read Digital Input					
Operand	**Name**	**Description**	**Nullable**	**Default value**	**Range**	**Data Type**
S1	Axis	Axis name/axis ID	No	-	-	WORD
D1	DIStatus	IO status	No	0	-	DINT
D2	Valid	Valid flag	Yes	OFF	ON/OFF	BOOL
D3	Busy	Ongoing execution flag	Yes	OFF	ON/OFF	BOOL
D4	Error	Command error flag	Yes	OFF	ON/OFF	BOOL
D5	ErrorID	Error code	Yes	0	-	WORD

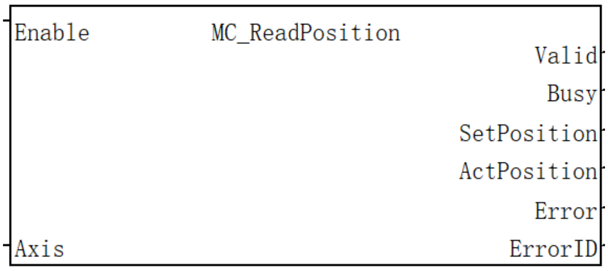
Operand	**Const**	**Y**	**M**	**S**	**D**	**R**	**Custom Variable**
S1	√	-	-	-	-	-	-
D1	-	-	-	-	√	√	√
D2	-	√	√	√	-	-	√
D3	-	√	√	√	-	-	√
D5	-	√	√	√	-	-	√
D6	-	-	-	-	√	√	√

Function Description

1. This command applies to the EtherCAT bus axis and local encoder axis, and does not support the virtual axis mode. This command is used to read the states of the digital input terminal of the axis.
2. When Enable is set to ON, if 0x60fd is configured in the PDO parameters of the bus axis, or if the left-right limit signal and origin signal of the local pulse axis are not all empty, the Valid signal output is valid.
3. If the current axis is an EtherCAT bus axis, DIStatus will display the digital input (0x60fd) of the EtherCAT bus driver. Refer to the corresponding driver manual for specific definitions.
4. If the current axis is a local pulse axis, the command is used to display the hard limit and origin signals; otherwise, it displays 0.
5. This command has no interrupt flag and therefor multiple commands can run simultaneously.

3.21.9 MC_ReadPosition

Graphic Block



16-Bit command	-					
32-Bit command	MC_ReadPosition: Read Actual Position					
Operand	**Name**	**Description**	**Nullable**	**Default value**	**Range**	**Data Type**
S1	Axis	Axis name/axis ID	No	-	-	WORD
D1	Valid	Valid flag	Yes	OFF	ON/OFF	BOOL
D2	Busy	Ongoing execution flag	Yes	OFF	ON/OFF	BOOL
D3	SetPosition	Axis command position	Yes	0	-	REAL
D4	ActPosition	Axis feedback position	Yes	0	-	REAL
D5	Error	Command error flag	Yes	OFF	ON/OFF	BOOL
D6	ErrorID	Error code	Yes	0	-	WORD

Operand	Const	Y	M	S	D	R	Custom Variable
S1	√	-	-	-	-	-	-
D1	-	√	√	√	-	-	√
D2	-	√	√	√	-	-	√
D3	-	-	-	-	√	√	√
D4	-	-	-	-	√	√	√
D5	-	√	√	√	-	-	√
D6	-	-	-	-	√	√	√

Function Description

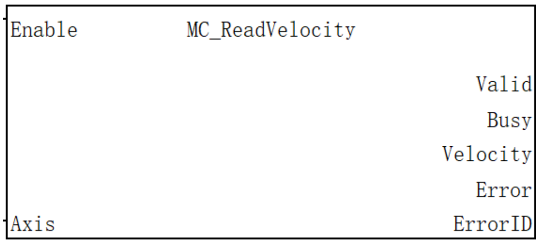
1、 The MC_ReadPosition command is used to read the axis command position or axis feedback position, and is valid at high levels.

2、 When the axis is a local pulse axis, the command output parameter ActPosition is actually the command position.

3、 This command has no interrupt flag and therefor multiple commands can run simultaneously.

3.21.10 MC_ReadVelocity

Graphic Block



16-Bit command	-					
32-Bit command	MC_ReadVelocity: Read Actual Velocity					
Operand	**Name**	**Description**	**Nullable**	**Default value**	**Range**	**Data Type**
S1	Axis	Axis name/axis ID	No	-	-	WORD
D1	Valid	Valid flag	Yes	OFF	ON/OFF	BOOL
D2	Busy	Ongoing execution flag	Yes	OFF	ON/OFF	BOOL
D3	Velocity	Axis feedback velocity	Yes	0	-	REAL
D4	Error	Command error flag	Yes	OFF	ON/OFF	BOOL
D5	ErrorID	Error code	Yes	0	-	WORD

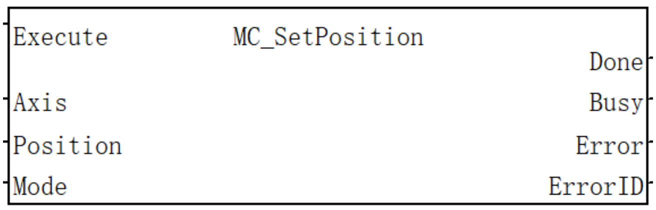
Operand	Const	Y	M	S	D	R	Custom Variable
S1	√	-	-	-	-	-	
D1	-	√	√	√	-	-	√
D2	-	√	√	√	-	-	√
D3	-	-	-	-	√	√	√
D4	-	√	√	√	-	-	√
D5	-	-	-	-	√	√	√

Function Description

1. The MC_ReadVelocity command is used to read the actual running velocity of the axis, and is valid at high levels.
2. This command has no interrupt flag and therefor multiple commands can run simultaneously.

3.21.11 MC_SetPosition

Graphic Block



16-Bit command	-					
32-Bit command	MC_SetPosition: Position Set					
Operand	**Name**	**Description**	**Nullable**	**Default value**	**Range**	**Data Type**
S1	Axis	Axis name/axis ID	No	-	-	WORD
S2	Position	Position value	Yes	0	Positive/negative/0	REAL
S3	Mode	Mode selection 0: absolute mode 1: relative mode	Yes	0	0~1	INT
D1	Done	Valid flag	Yes	OFF	ON/OFF	BOOL
D2	Busy	Ongoing execution flag	Yes	OFF	ON/OFF	BOOL
D3	Error	Command error flag	Yes	OFF	ON/OFF	BOOL
D4	ErrorID	Error code	Yes	0	-	WORD

Operand	Const	Y	M	S	D	R	Custom Variable
S1	√	-	-	-	-	-	
S2	√	-	-	-	√	√	√
S3	√	-	-	-	√	√	√
D1	-	√	√	√	-	-	√
D2	-	√	√	√	-	-	√
D3	-	√	√	√	-	-	√
D4	-	-	-	-	√	√	√

Function Description

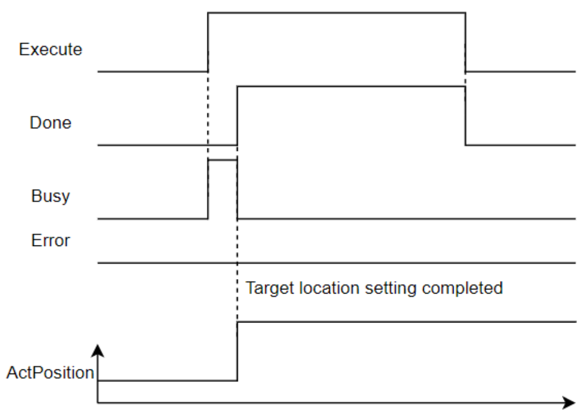
1. This command is used to set the current position of the bus servo axis or local pulse axis, and is valid to the rising edge.
2. In case of Mode=0 (absolute mode), the command writes Position to the current position of the axis; in case of Mode=1 (relative mode), it adds Position to the current position of the axis.
3. This command is valid only when triggered in the Standstall state.

Multiple Starts of This Command

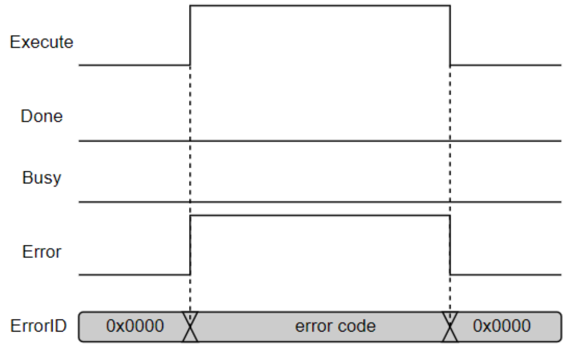
This command does not support interrupt. If there are multiple commands in one scan cycle, the first valid command will be executed. During the validity period of the Busy signal, if there is another SetPosition command running, other commands will report an error.

Timing diagram

- When the axis is in the StandStill state, this command is executed, and the relative mode is selected.

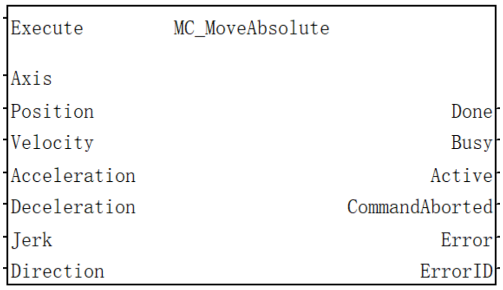


- During the axis operation, this command is executed.



3.21.12 MC_MoveAbsolute

Graphic Block



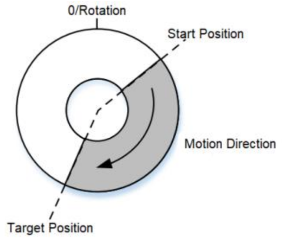
16-Bit command	-					
32-Bit command	MC_MoveAbsolute: Absolute Positioning					
Operand	**Name**	**Description**	**Nullable**	**Default value**	**Range**	**Data Type**
S1	Axis	Axis name/axis ID	No	-	-	WORD
S2	Position	Target position	No	0	Positive/negative/0	REAL
S3	Velocity	Target velocity	Yes	100	Positive number	REAL
S4	Acceleration	Acceleration	Yes	1000	Positive number	REAL
S5	Deceleration	Deceleration	Yes	1000	Positive number	REAL
S6	Jerk	Jerk value 0: T-type acceleration and deceleration >0: S-type acceleration and deceleration	Yes	0	Positive/0	REAL
S7	Direction	Direction 0: positive (velocity ＞ 0) 1: negative (velocity ＜ 0) 2: minimum distance 3: current	Yes	0	0~3	INT
D1	Done	Command execution completion	Yes	OFF	ON/OFF	BOOL
D2	Busy	Ongoing execution flag	Yes	OFF	ON/OFF	BOOL
D3	Active	Execution validity flag	Yes	OFF	ON/OFF	BOOL
D4	CommandAborted	Execution interrupt flag	Yes	OFF	ON/OFF	BOOL
D5	Error	Error flag	Yes	OFF	ON/OFF	BOOL
D6	ErrorID	Error code	Yes	0	-	WORD

Operand	Const	Y	M	S	D	R	Custom Variable
S1	√	-	-	-			
S2	√	-	-	-	√	√	√
S3	√	-	-	-	√	√	√
S4	√	-	-	-	√	√	√
S5	√	-	-	-	√	√	√
S6	√	-	-	-	√	√	√
S7	√	-	-	-	√	√	√
D1	-	√	√	√	-	-	√
D2	-	√	√	√	-	-	√
D3	-	√	√	√	-	-	√
D4	-	√	√	√	-	-	√
D5	-	√	√	√	-	-	√
D6	-	-	-	-	√	√	√

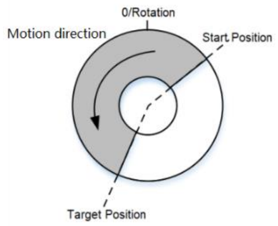
Function Description

1. The MC_MoveAbsolute command is used to control the bus servo axis or local pulse axis to achieve absolute positioning, and is valid to the rising edge.
2. On the rising edge of the Execute input, the command locks the left input parameters such as Position and Velocity, triggers the absolute positioning function, and switches the axis state to DiscreteMotion state.
3. In the linear mode, Position is used to set the target position for absolute positioning. If the current position is less than the target position, the axis will move forward and finally reach the position set by Position. If the current position is greater than the target position, the axis will move backward and finally reach the position set by Position.
4. In the rotary mode, the command first uses Position to calculate the modulus over the rotation cycle to obtain the absolute position Position_p within one rotation cycle. The actual motion direction of axis is absolute in the following 4 situations:

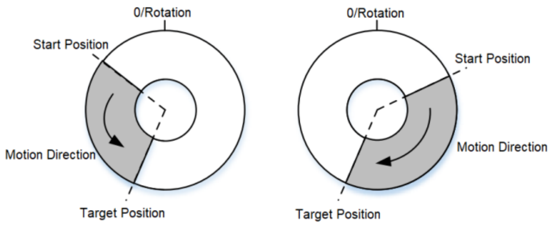
(1) Direction=0 indicates the forward motion (the target velocity greater than 0). If the current velocity is greater than 0, the axis continues running in the current direction and then stops at the position set by Position_p; if the current velocity is less than 0, the axis slows down to 0 first and then starts running backward until the position set by Position_p; if Position_p is equal to the current position, the axis does not move.



(2) Direction=1 indicates the backward motion (the target velocity less than 0). If the current velocity is less than 0, the axis continues running in the current direction and then stops at the position set by Position_p; if the current velocity is greater than 0, the axis slows down to 0 first and then starts running backward until the position set by Position_p; if Position_p is equal to the current position, the axis does not move.



(3) Direction=2 indicates the minimum distance. On the rising edge of Execute, the axis will record the current position, and assume that the current velocity is 0 to first calculate the displacement Distance_p from the 0 velocity to Position_p through the forward motion and then the displacement Distance_n from the 0 velocity to Position_p through the backward motion. In case Distance_p>Distance_n, the axis runs in the negative direction; otherwise, the axis runs in the positive direction; if Position_p is equal to the current position, the axis does not move.



(4) Direction=3 indicates the current direction. On the rising edge of Execute, the axis will move in the direction of the most recent motion before reaching the rising edge of the Execute, and stop at the position set by Position_p. If it is powered on for the first time, the axis will run in the forward direction (the target velocity greater than 0); if Position_p is equal to the current position, the axis does not move.

5. Jerk is used to set the type of the speed curve. Jerk=0 represents the T-type curve acceleration and deceleration, while Jerk>0 represents the S-type curve acceleration and deceleration.
6. This command supports the mutual interruption between functional blocks according to the PLCopen state machine standards.

Resetting This Command

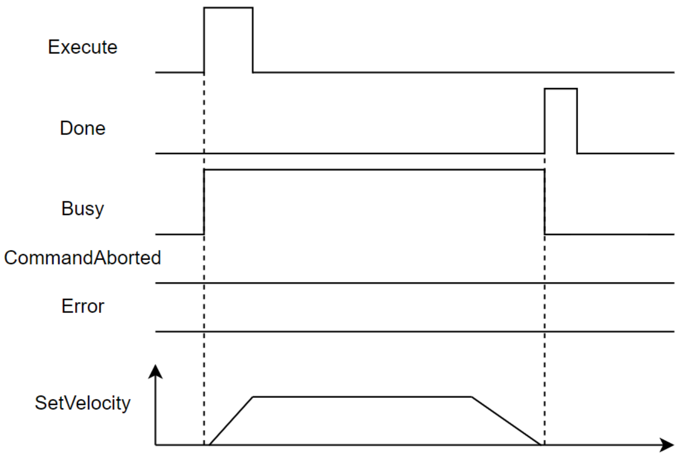
During the validity period of the Busy signal of the MoveAbsolute command, if the MC_MoveAbsolute command is triggered again, it will re-plan with new target parameters according to the current motion position, velocity, etc.

Multiple Starts of This Command

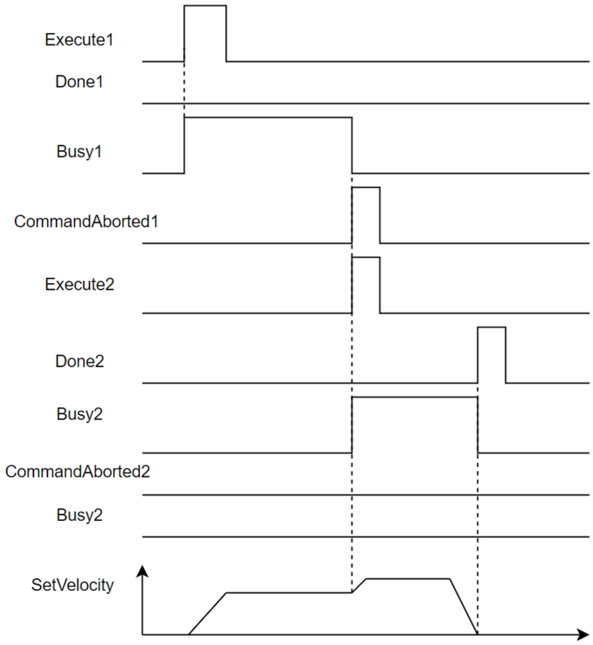
During the validity period of the Busy signal of the MoveAbsolute command, if the second MC_MoveAbsolute command is triggered, the second command will re-plan with new target parameters according to the current motion position, velocity, etc., while the first command will be interrupted and invalidated.

Timing diagram

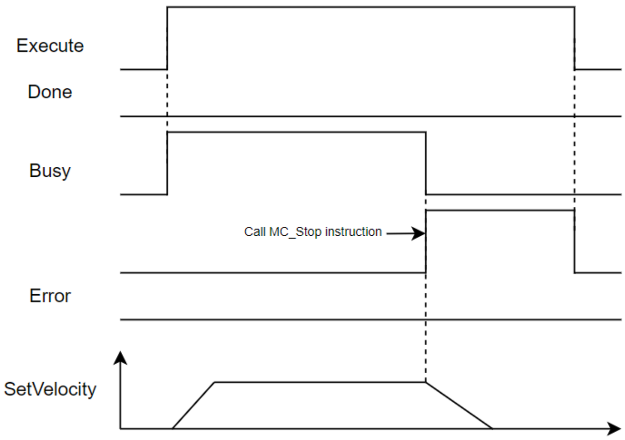
- In the StandStill state, the axis calls this command to perform the relative positioning motion with a T-typed curve.



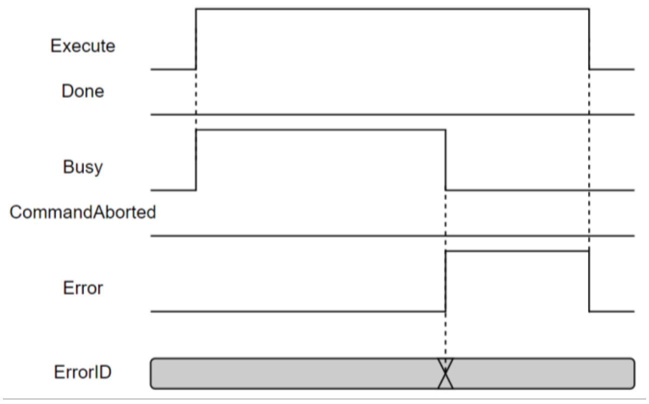
- During the operation of absolute positioning, another absolute positioning command (acting on the same axis) is triggered.



- During the relative positioning motion, the axis is interrupted by the MC_Stop command.

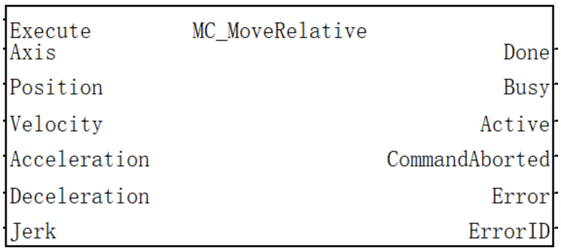


- During the axis operation, the driver experiences a fault.



3.21.13 MC_MoveRelative

Graphic Block



16-Bit command	-					
32-Bit command	MC_MoveRelative: Relative Positioning					
Operand	**Name**	**Description**	**Nullable**	**Default value**	**Range**	**Data Type**
S1	Axis	Axis name/axis ID	No	-	-	WORD
S2	Position	Target position	No	0	Positive/negative/0	REAL
S3	Velocity	Target velocity	Yes	100	Positive number	REAL
S4	Acceleration	Acceleration	Yes	1000	Positive number	REAL
S5	Deceleration	Deceleration	Yes	1000	Positive number	REAL
S6	Jerk	Jerk value 0: T-type acceleration and deceleration >0: S-type acceleration and deceleration	Yes	0	Positive/0	REAL
D1	Done	Command execution completion	Yes	OFF	ON/OFF	BOOL
D2	Busy	Ongoing execution flag	Yes	OFF	ON/OFF	BOOL
D3	Active	Execution validity flag	Yes	OFF	ON/OFF	BOOL
D4	CommandAborted	Execution interrupt flag	Yes	OFF	ON/OFF	BOOL
D5	Error	Error flag	Yes	OFF	ON/OFF	BOOL
D6	ErrorID	Error code	Yes	0	-	WORD

Operand	Const	Y	M	S	D	R	Custom Variable
S1	√	-	-	-			
S2	√	-	-	-	√	√	√
S3	√	-	-	-	√	√	√
S4	√	-	-	-	√	√	√
S5	√	-	-	-	√	√	√
S6	√	-	-	-	√	√	√
D1	-	√	√	√	-	-	√
D2	-	√	√	√	-	-	√
D3	-	√	√	√	-	-	√

Operand	Const	Y	M	S	D	R	Custom Variable
D4	-	√	√	√	-	-	√
D5	-	√	√	√	-	-	√
D6	-	-	-	-	√	√	√

Function Description

1. The MC_MoveRelative command is used to control the bus servo axis or local pulse axis to achieve relative positioning, and is valid to the rising edge.
2. On the rising edge of the Execute input, the command locks the left input parameters such as Position and Velocity, triggers the relative positioning function, and switches the axis state to DiscreteMotion state.
3. Position is used to set the distance for relative positioning. Regardless of the linear or rotary mode, if Position is a positive number, the axis will run in the positive direction for the distance specified by Position; if Position is negative number, the axis will run in the negative direction for the distance specified by Position.
4. Jerk is used to set the type of the speed curve. Jerk=0 represents the T-type curve acceleration and deceleration, while Jerk>0 represents the S-type curve acceleration and deceleration.
5. This command supports the mutual interruption between functional blocks according to the PLCopen state machine standards.

Resetting This Command

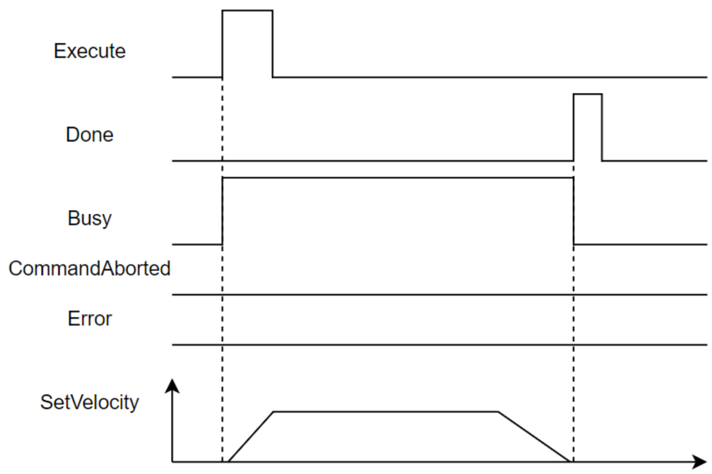
During the validity period of the Busy signal of the MC_MoveRelative command, if the MC_MoveRelative command is triggered again, it will re-plan with new target parameters according to the current motion position, velocity, etc.

Multiple Starts of This Command

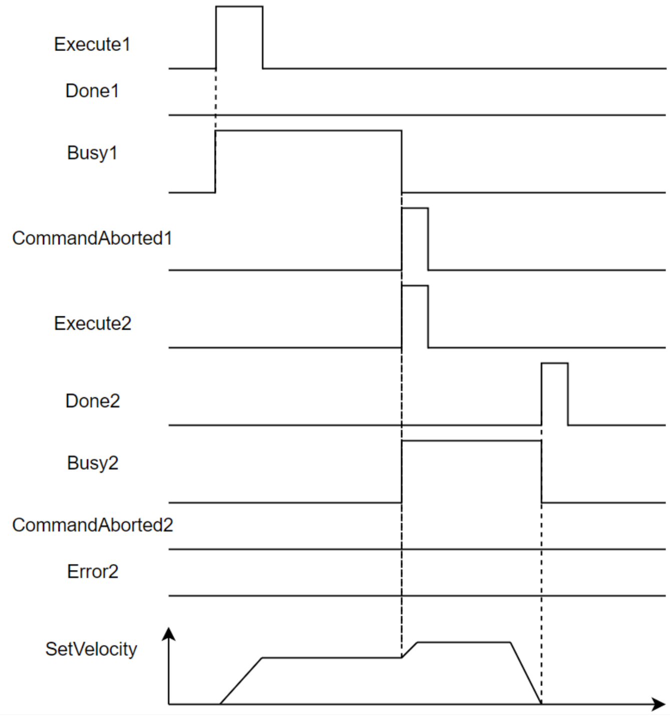
During the validity period of the Busy signal of the current MC_MoveRelative command, if the second MC_MoveRelative command is triggered, the second command will re-plan with new target parameters according to the current motion position, velocity, etc., while the current MC_MoveRelative command will be interrupted and invalidated.

Timing diagram

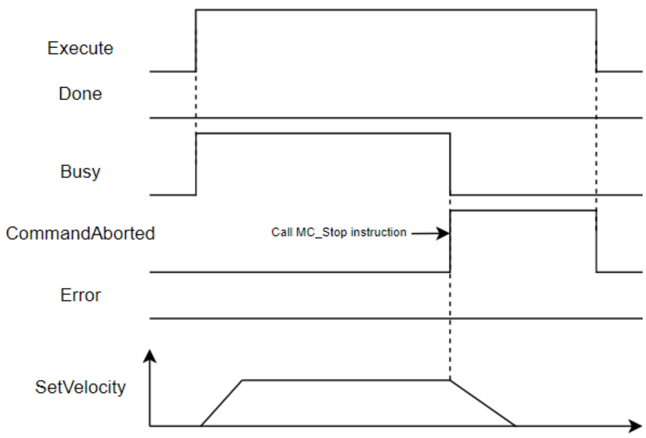
- In the StandStill state, the axis calls this command to perform the relative positioning motion with a T-typed curve.



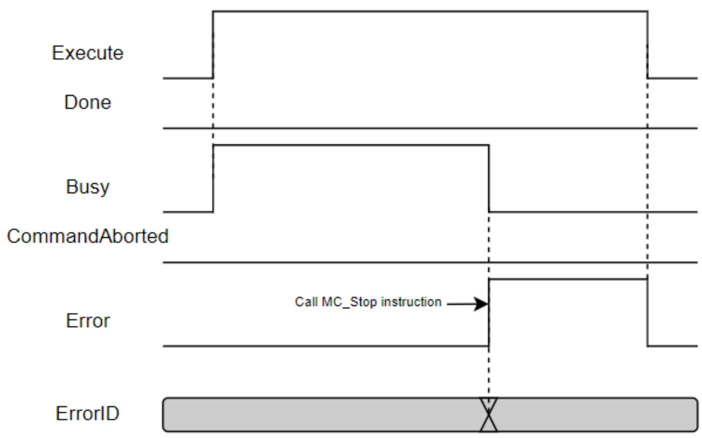
- During the operation of relative positioning, another relative positioning command (acting on the same axis) is triggered.



- During the relative positioning motion, the axis is interrupted by the MC_Stop command.

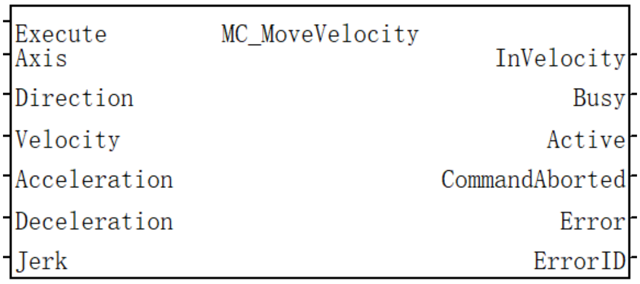


- During the axis operation, the driver experiences a fault.



3.21.14 MC_MoveVelocity

Graphic Block



16-Bit command	-					
32-Bit command	MC_MoveVelocity: Velocity					
Operand	**Name**	**Description**	**Nullable**	**Default value**	**Range**	**Data Type**
S1	Axis	Axis name/axis ID	No	-	-	WORD
S2	Direction	Axis motion direction 0: negative Others: positive	Yes	1	Positive/negative/0	INT
S3	Velocity	Target velocity	Yes	100	Positive number	REAL
S4	Acceleration	Acceleration	Yes	1000	Positive number	REAL
S5	Deceleration	Deceleration	Yes	1000	Positive number	REAL
S6	Jerk	Jerk value 0: T-type acceleration and deceleration >0: S-type acceleration and deceleration	Yes	0	Positive/0	REAL
D1	InVelocity	Reach target velocity	Yes	OFF	ON/OFF	BOOL
D2	Busy	Ongoing execution flag	Yes	OFF	ON/OFF	BOOL
D3	Active	Execution validity flag	Yes	OFF	ON/OFF	BOOL
D4	CommandAborted	Execution interrupt flag	Yes	OFF	ON/OFF	BOOL
D5	Error	Error flag	Yes	OFF	ON/OFF	BOOL
D6	ErrorID	Error code	Yes	0	-	WORD

Operand	**Const**	**Y**	**M**	**S**	**D**	**R**	**Custom Variable**
S1	√	-	-	-	-	-	
S2	√	-	-	-	√	√	√
S3	√	-	-	-	√	√	√
S4	√	-	-	-	√	√	√
S5	√	-	-	-	√	√	√
S6	√	-	-	-	√	√	√
D1	-	√	√	√	-	-	√
D2	-	√	√	√	-	-	√
D3	-	√	√	√	-	-	√
D4	-	√	√	√	-	-	√
D5	-	√	√	√	-	-	√
D6	-	-	-	-	√	√	√

Function Description

1. The MC_MoveVelocity command is used to control the velocity of the bus servo axis and local pulse axis, and is valid to the rising edge.
2. Jerk is used to set the type of the speed curve. Jerk=0 represents the T-type curve acceleration and deceleration, while Jerk>0 represents the S-type curve acceleration and deceleration.
3. This command supports the mutual interruption between functional blocks according to the PLCopen state machine standards.
4. After calling this command, you can call the MC_Stop, MC_Halt and MC_ImmediateStop commands to stop the axis from running.

Resetting This Command

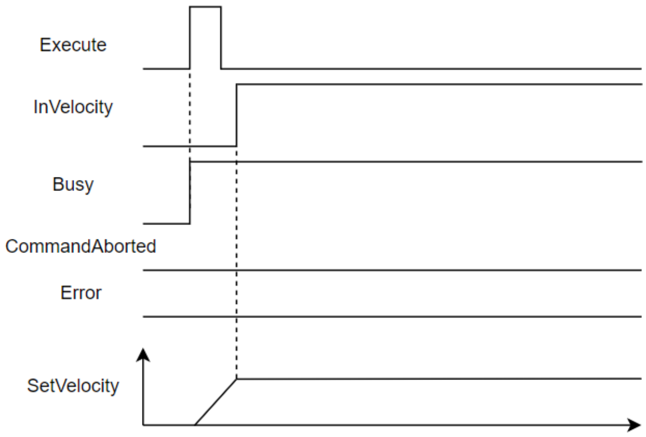
During the validity period of the Busy signal of the MC_MoveVelocity command, if the MC_MoveVelocity command is triggered again, it will re-plan with new target parameters according to the current motion position, velocity, etc.

Multiple Starts of This Command

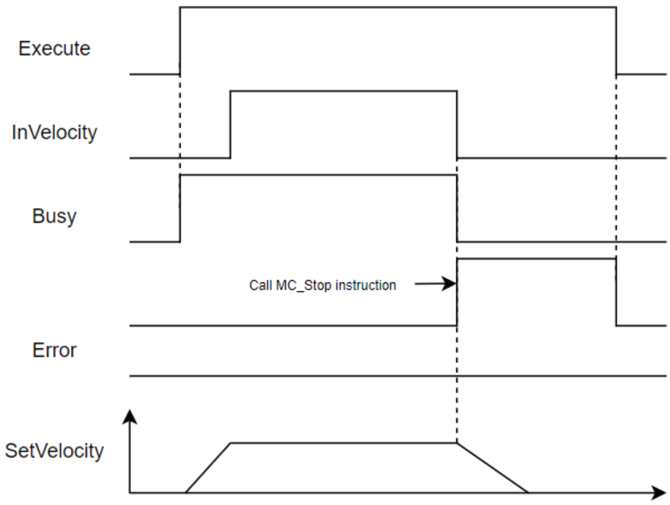
During the validity period of the Busy signal of the MC_MoveVelocity command, if the second MC_MoveVelocity command is triggered, the second command will re-plan with new target parameters according to the current motion position, velocity, etc., while the first command will be interrupted and invalidated.

Timing diagram

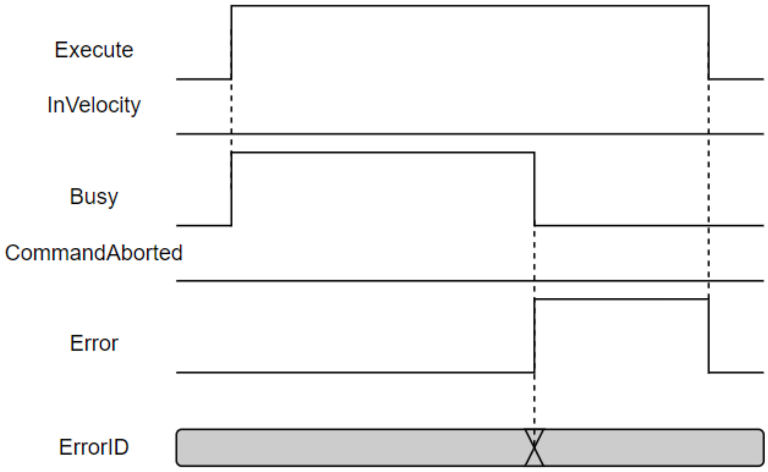
- In the StandStill state, the axis calls this command to perform the continuous motion with a T-typed curve.



- During running, the axis is interrupted by the MC_Stop command.

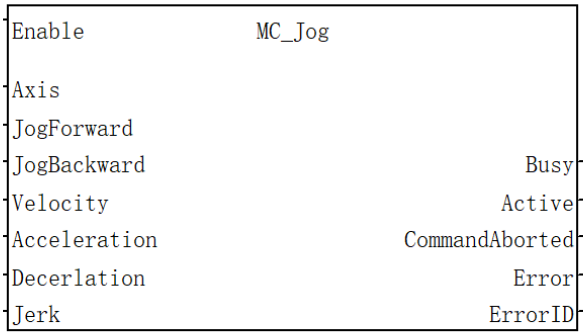


- During the axis acceleration, the driver experiences a fault.



3.21.15 MC_Jog

Graphic Block



16-Bit command	-					
32-Bit command	MC_Jog: Continuous Run					
Operand	**Name**	**Description**	**Nullable**	**Default value**	**Range**	**Data Type**
S1	Axis	Axis name/axis ID	No	-	-	WORD
S2	JogForward	Enable forward	No	OFF	ON/OFF	BOOL
S3	JogBackward	Enable backward	No	OFF	ON/OFF	BOOL
S4	Velocity	Target velocity	Yes	100	Positive number	REAL
S5	Acceleration	Acceleration	Yes	1000	Positive number	REAL
S6	Deceleration	Deceleration	Yes	1000	Positive number	REAL
S7	Jerk	Jerk value 0: T-type acceleration and deceleration >0: S-type acceleration and deceleration	Yes	0	Positive/0	REAL
D1	Busy	Ongoing execution flag	Yes	OFF	ON/OFF	BOOL
D2	Active	Execution validity flag	Yes	OFF	ON/OFF	BOOL
D3	CommandAborted	Execution interrupt flag	Yes	OFF	ON/OFF	BOOL
D4	Error	Error flag	Yes	OFF	ON/OFF	BOOL
D5	ErrorID	Error code	Yes	0	-	WORD

Operand	Const	Y	M	S	D	R	Custom Variable
S1	√	-	-	-			
S2	√	-	-	-	√	√	√
S3	√	-	-	-	√	√	√
S4	√	-	-	-	√	√	√
S5	√	-	-	-	√	√	√
S6	√	-	-	-	√	√	√
S7	√	-	-	-	√	√	√
D1	-	√	√	√	-	-	√
D2	-	√	√	√	-	-	√
D3	-	√	√	√	-	-	√
D4	-	√	√	√	-	-	√
D5	-	-	-	-	√	√	√

Function Description

1. The MC_Jog command is used to control the bus servo axis and local pulse axis to achieve continuous axis run, and the direction is determined by the enable of JogForward and JogBackward.
2. When JogForward is set to TRUE, the axis runs in the positive direction; when JogForward is set to FALSE, the axis decelerates and then stops. When JogBackward is set to TRUE, the axis runs in the negative direction; when JogBackward is set to FALSE, the axis decelerates and then stops.
3. During the positive axis run, if JogForward is set to TRUE, the axis decelerates and then stops. During the negative axis run, if JogBackward is set to TRUE, the axis decelerates and then stops.
4. Jerk is used to set the type of the speed curve. Jerk=0 represents the T-type curve acceleration and deceleration, while Jerk>0 represents the S-type curve acceleration and deceleration.
5. This command supports the mutual interruption between functional blocks according to the PLCopen state machine standards.
6. After calling this command, you can call the MC_Stop, MC_Halt and MC_ImmediateStop commands to stop the axis from running.

Resetting This Command

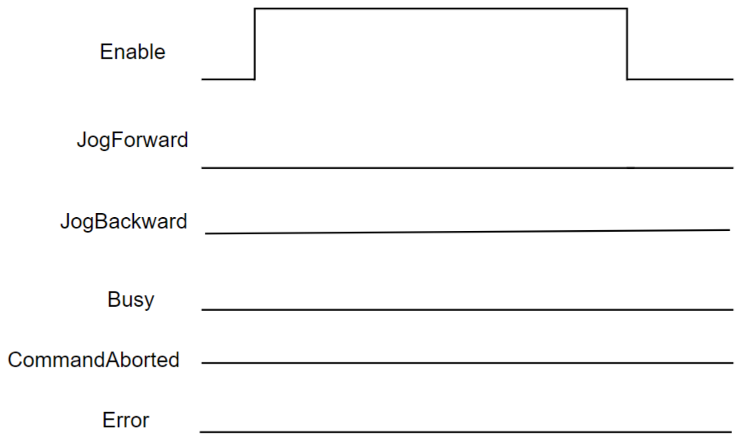
During the validity period of the Busy signal of the MC_Jog, if the MC_Jog command is triggered again, it will re-plan with new target parameters according to the current motion position, velocity, etc.

Multiple Starts of This Command

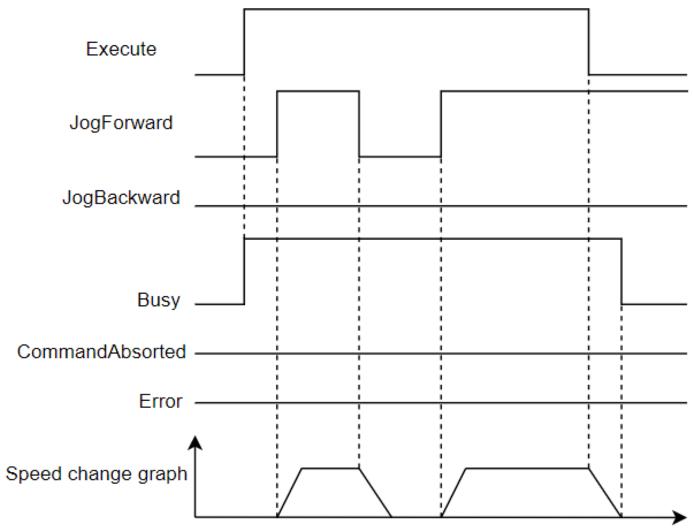
During the validity period of the Busy signal of the MC_Jog command, if the second MC_Jog command is triggered, the second command will re-plan with new target parameters according to the current motion position, velocity, etc., while the first command will be interrupted and invalidated.

Timing diagram

- When only Enable is valid, the axis is in the Standstill state.

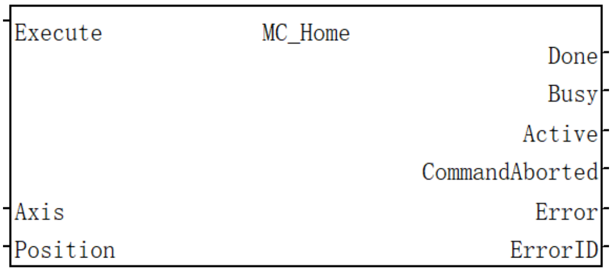


- When the inputs of Enable and JogForward are valid.



3.21.16 MC_Home

Graphic Block



16-Bit command	-					
32-Bit command	MC_Home: Home					
Operand	**Name**	**Description**	**Nullable**	**Default value**	**Range**	**Data Type**
S1	Axis	Axis name/axis ID	No	-	-	WORD
S2	Position	Offset position	No	0	-	REAL
D1	Done	Completion flag	Yes	OFF	ON/OFF	BOOL
D2	Busy	Ongoing execution flag	Yes	OFF	ON/OFF	BOOL
D3	Active	Execution validity flag	Yes	OFF	ON/OFF	BOOL
D4	CommandAborted	Execution interrupt flag	Yes	OFF	ON/OFF	BOOL
D5	Error	Error flag	Yes	OFF	ON/OFF	BOOL
D6	ErrorID	Error code	Yes	0	-	WORD

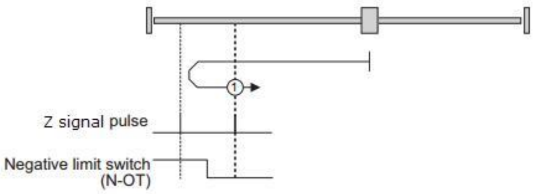
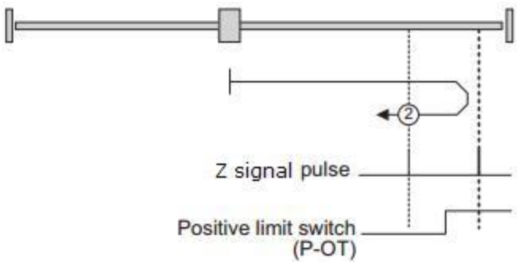
Operand	Const	Y	M	S	D	R	Custom Variable
S1	√	-	-	-	-	-	√
S2	√	-	-	-	√	√	√
D1	-	√	√	√	-	-	√
D2	-	√	√	√	-	-	√
D3	-	√	√	√	-	-	√
D4	-	√	√	√	-	-	√
D5	-	√	√	√	-	-	√
D6	-	-	-	-	√	√	√

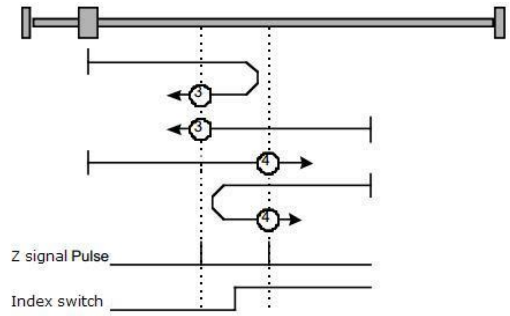
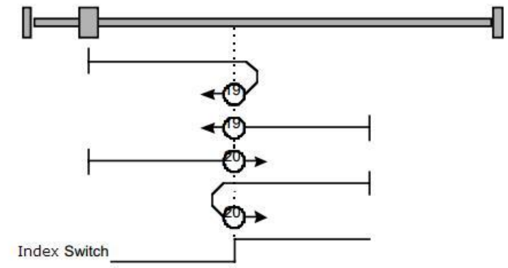
Function Description

1. The MC_Home command is used to control the bus servo axis or local pulse axis to home, and is valid to the rising edge.
2. This command can be called only by switching the axis to the enabled state using the MC_Power command.
3. On the rising edge of the command, the function block locks the Position input parameter, the axis processes the Homing state and performs the homing motion. Position is used to set the origin offset, parameters such as home mode is set in the axis configuration interface.
4. This command does not support the virtual axis mode.
5. This command does not support the mutual interruption between functional blocks.
6. After calling this command, you can call the MC_Stop and MC_ImmediateStop commands to stop the axis from running.

Bus Axis Home Modes:

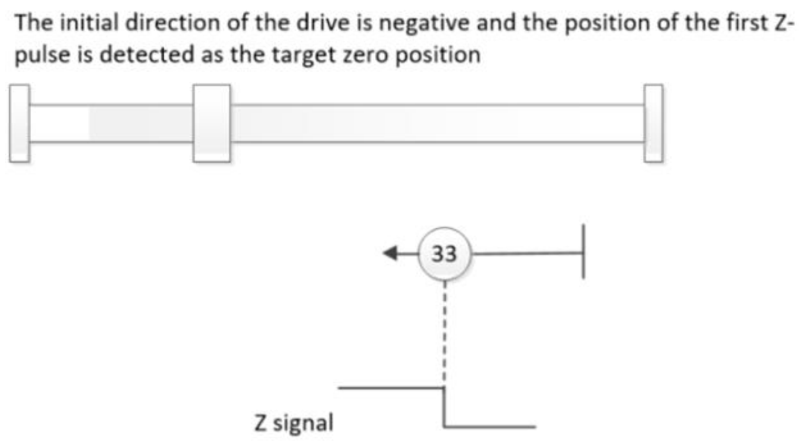
There are four types of signals related to home modes, namely: positive limit switch (POT), negative limit switch (NOT), reference point switch (Index), and encoder Z signal. See below for specific meanings of home modes:

Homing method (DS402)	Start direction	Target position	Reference point position	Homing method (P5.10)	Description
1	Negative	NOT	Z pulse	1	Using Z pulse and negative limit switch: The drive moves towards negative limit switch at high speed, then returns at low speed and searches for target zero position (the first encoder Z pulse position after leaving NOT) after reaching NOT. 
2	Positive	POT	Z pulse	0	Using Z pulse and positive limit switch: The drive moves towards positive limit switch at high speed, then returns at low speed and searches for target zero position (the first encoder Z pulse position after leaving NOT) after reaching POT. 
3	Positive	Index	Z pulse	2	The initial direction movement of the drive depends on the switch state of the reference point. The target zero position is the first Z pulse position on the left or right side of the Index.
4	Positive	Index	Z pulse	12	

Homing method (DS402)	Start direction	Target position	Reference point position	Homing method (P5.10)	Description
					
17	Negative	NOT	NOT	21	These four types of homing methods are similar to 1–4 phases except that the target zero position is related to the change of limit switch or Index switch rather than using Z pulse. The following figure is diagram for 19 and 20, which are similar to method 3 and 4. 
18	Positive	POT	POT	20	
19	Negative	Index	Index	23	
20	Positive	Index	Index	22	
35	-	Present position value	Present position value	8	The present position is the system zero point.

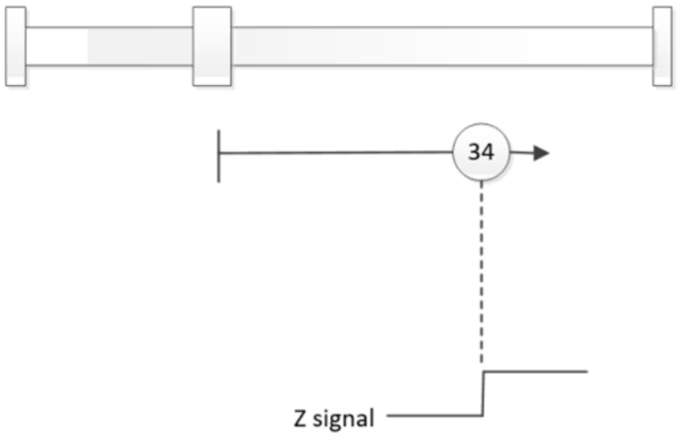
Note: Both home mode 19 and 20 correspond to home mode 22 (under P5.10), the actual home effect is consistent with that of home mode 19 shown in the figure above, and the home mode diagram drawn on the host computer interface shall prevail.

Home mode 33:



Home mode 34:

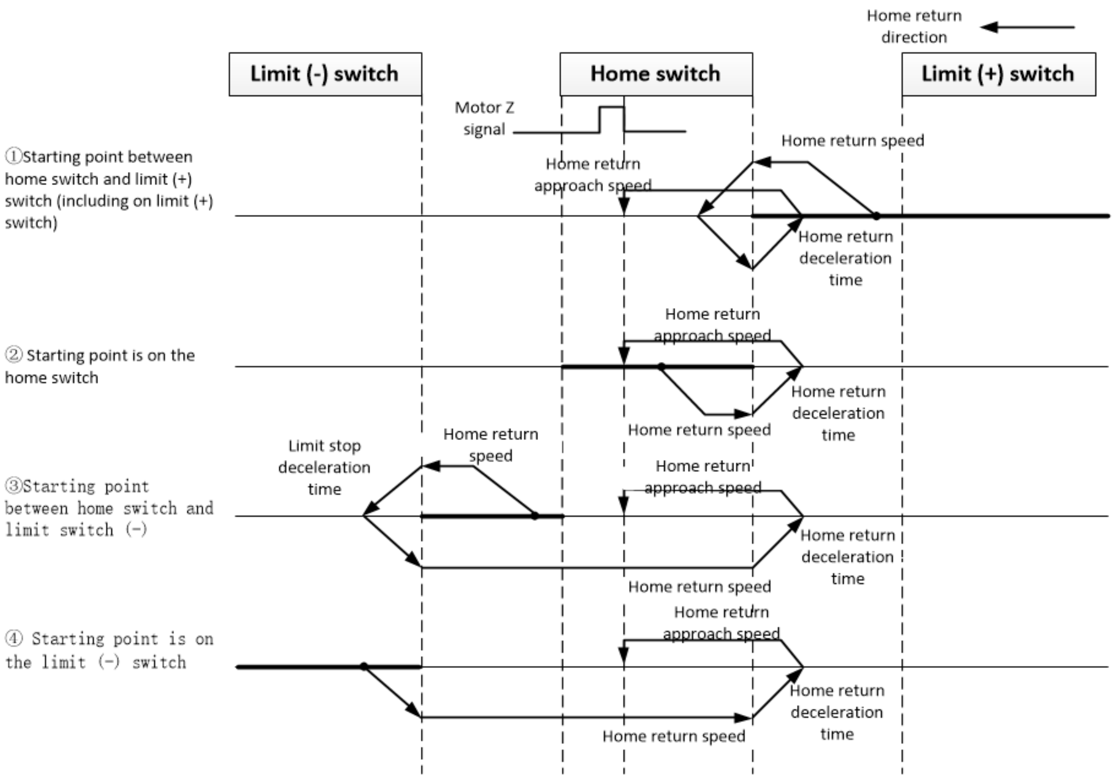
The initial direction of the drive is positive, and the position of the first Z-pulse is detected as the target zero position.



Local Pulse Axis Home Modes:

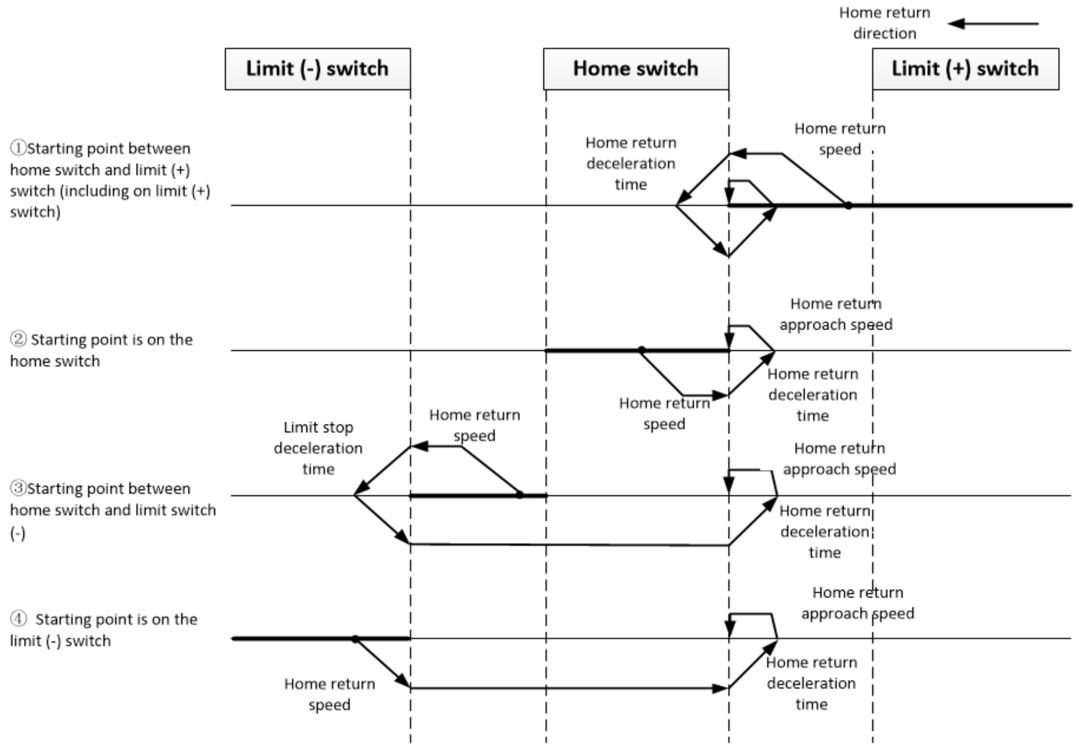
Home mode 2:

When the rising edge of the home switch is detected, the rising edge of the initial home switch is used as the home point.



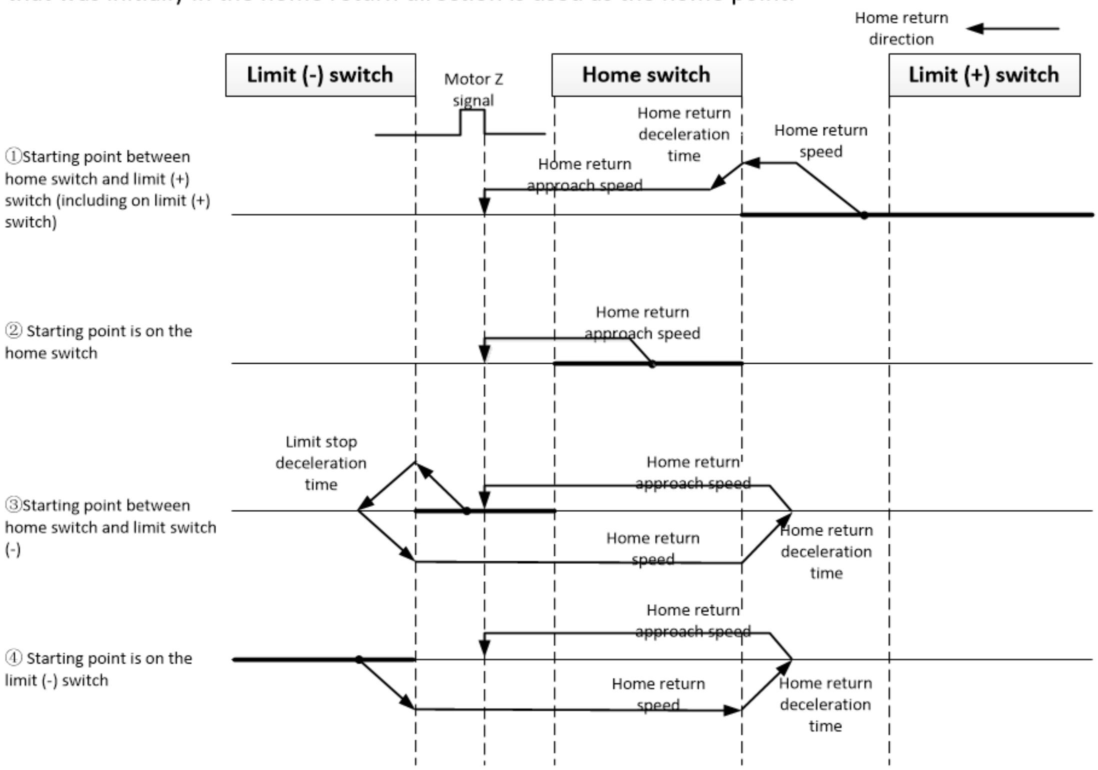
Home mode 3:

The rising edge of the home switch is detected and used as the home point.



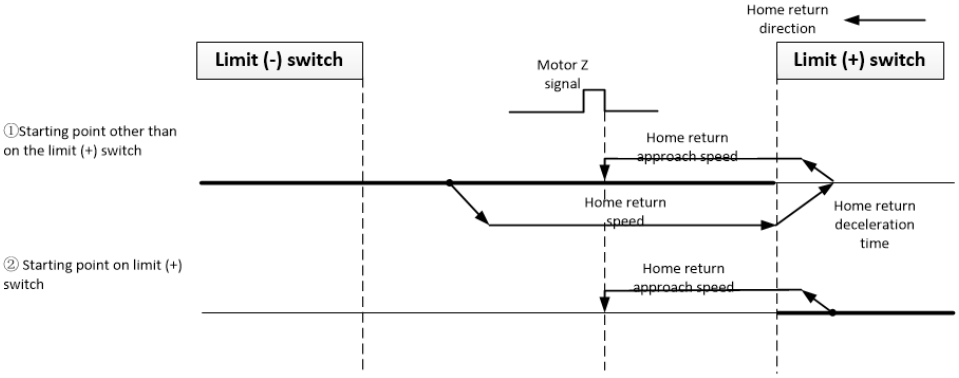
Home mode 4:

When the falling edge (rear end) of the home switch is detected, the rising edge of the home switch that was initially in the home return direction is used as the home point.



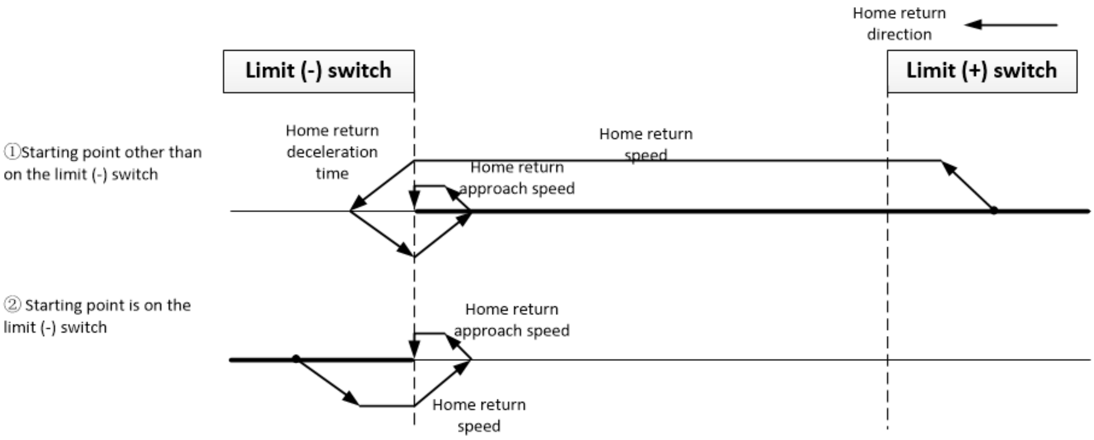
Home mode 5:

After detecting the rising edge of the limit switch in the direction opposite to the home return direction, reverse rotation is performed. Then, it stops at the rising edge of the motor Z signal, which is used as the home position.



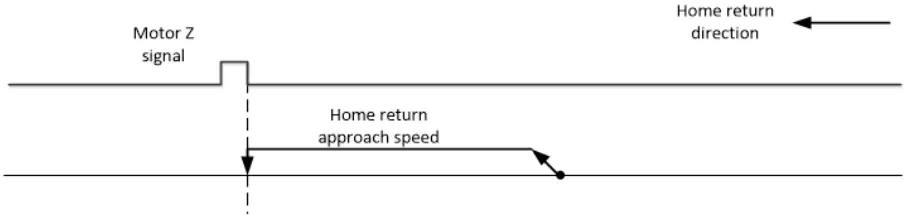
Home mode 6:

Stop after detecting the rising edge of the limit switch in the home return direction.



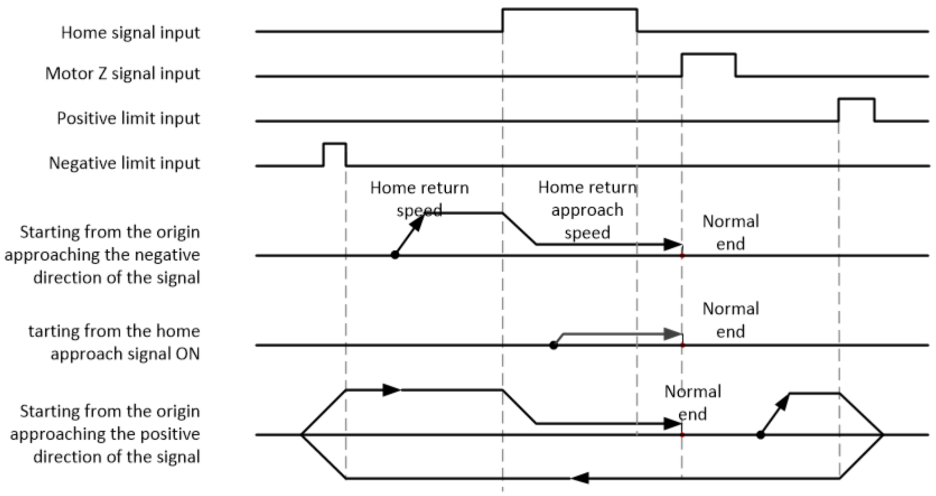
Home mode 7:

It moves from the current value to the home return direction, and stops when it detects the rising edge of the motor Z signal, which is used as the home point.

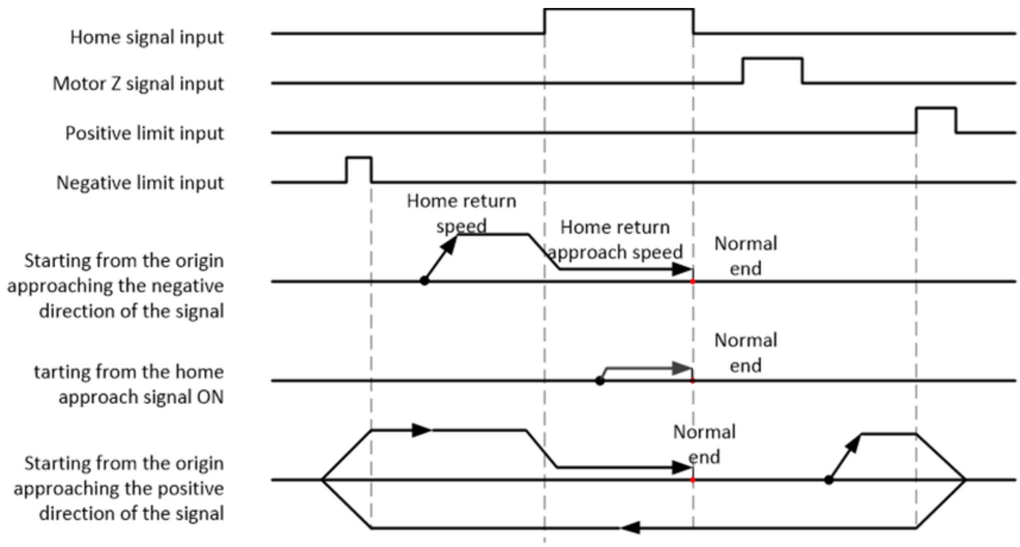


(Note) If the home position sensor is ON at startup, the home position sensor is not detected, and action is taken in the home return direction.

Home mode 8:



Home mode 9:



Note: The origin signal, positive and negative hard limit signals, home mode, and home-related parameters (homing direction, homing velocity, etc.) should be set on the host computer.

Controller Home	Applicable Scenarios
2	Origin signal (deceleration block signal) + motor Z signal, or origin signal + motor Z signal + positive and negative limit signals
3	Origin signal, or origin signal + positive and negative limit signals
4	Origin signal + motor Z signal, or origin signal + motor Z signal + positive and negative limit signals
5	Motor Z signal + positive and negative limit signals
6	Positive and negative limit signals
7	Motor Z signal
8	Origin signal + motor Z signal, or origin signal + motor Z signal + positive and negative limit signals
9	Origin signal, or origin signal + positive and negative limit signals

Note:

- If there are only positive and negative limit signals on site, home mode 6 is selected.
- If there are neither origin signal nor positive and negative limit signals on site, home mode 7 is selected in order to home the motor.
- Home modes 2, 4, and 8 are suitable for most scenarios.

Resetting This Command

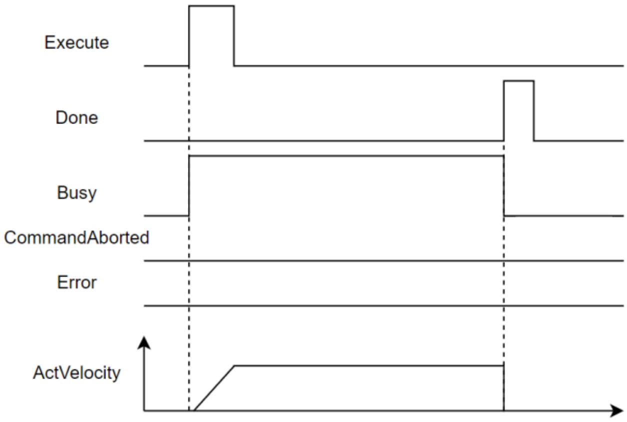
During the validity period of the Busy signal of the MC_Home command, if the MC_Home command is triggered again, the system displays that the MC_Home command is interrupted.

Multiple Starts of This Command

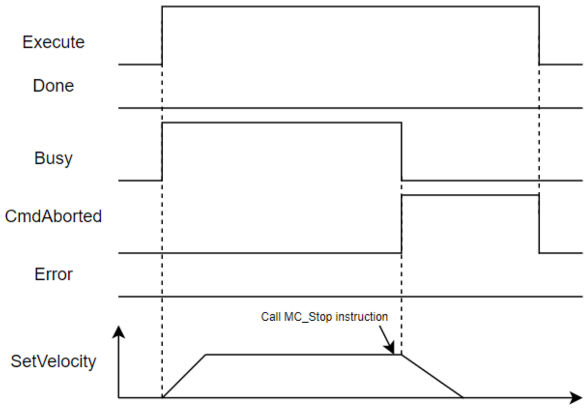
During the validity period of the Busy signal of the MC_Home command, if the second MC_Home command is triggered, the system displays that the first command is interrupted and the second command has an error.

Timing diagram

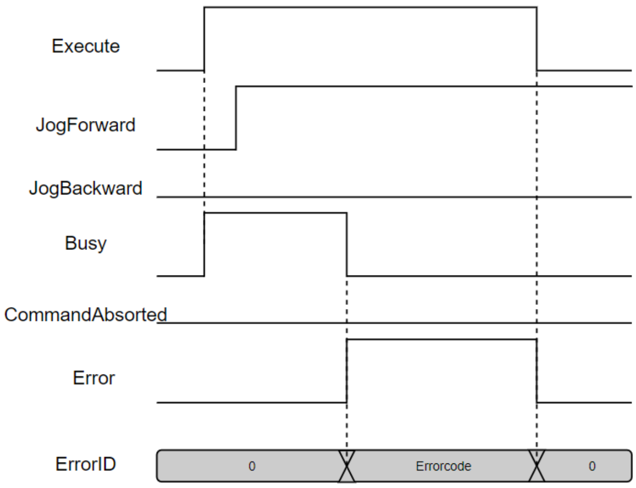
- When this command is enabled, the driver performs the homing action normally.



- During the homing process, the home command is interrupted by the MC_Stop command.

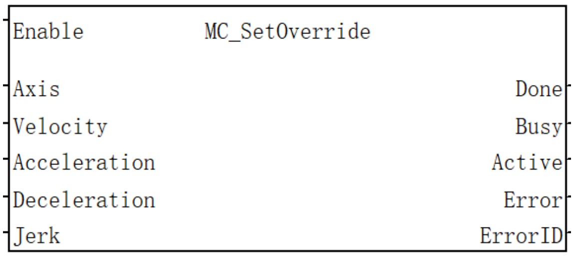


- During the homing process, the driver experiences a fault.



3.21.17 MC_SetOverride

Graphic Block



16-Bit command	-					
32-Bit command	MC_SetOverride: Variable Velocity, Acceleration, and Deceleration					
Operand	**Name**	**Description**	**Nullable**	**Default value**	**Range**	**Data Type**
S1	Axis	Axis name/axis ID	No	-	-	WORD
S2	Velocity	Target velocity after velocity regulation	Yes	100	Positive number	REAL
S3	Acceleration	Acceleration after velocity regulation	Yes	1000	Positive number	REAL
S3	Deceleration	Deceleration after velocity regulation	Yes	1000	Positive number	REAL
S4	Jerk	Jerk after velocity regulation 0: T-type acceleration and deceleration >0: S-type acceleration and deceleration	Yes	0	Positive/0	REAL
D1	Done	Completion flag	Yes	OFF	ON/OFF	BOOL
D2	Busy	Ongoing execution flag	Yes	OFF	ON/OFF	BOOL
D3	Active	Alternate waiting flag	Yes	OFF	ON/OFF	BOOL
D4	Error	Command error flag	Yes	OFF	ON/OFF	BOOL
D5	ErrorID	Error code	Yes	0	-	WORD

Operand	Const	Y	M	S	D	R	Custom Variable
S1	√	-	-	-	-	-	-
S2	√	-	-	-	√	√	√
S3	√	-	-	-	√	√	√
S4	√	-	-	-	√	√	√
S5	√	-	-	-	√	√	√
D1	-	√	√	√	-	-	√
D2	-	√	√	√	-	-	√
D3	-	√	√	√	-	-	√
D4	-	√	√	√	-	-	√
D5	-	-	-	-	√	√	√

Function Description

1. The MC_SetOverride command is used to achieve variable velocity, acceleration, and deceleration, and is valid at high levels.
2. Velocity, Acceleration, and Jerk represent the velocity, acceleration, and acceleration after velocity adjustment, respectively.

   It should be noted that setting Velocity to a negative number is only valid for the MoveVelocity command, where a positive number indicates that the command is running in the forward direction, while a negative number indicates that the command is running in the negative direction. For other commands, positive numbers are valid (processed by taking absolute values internally)

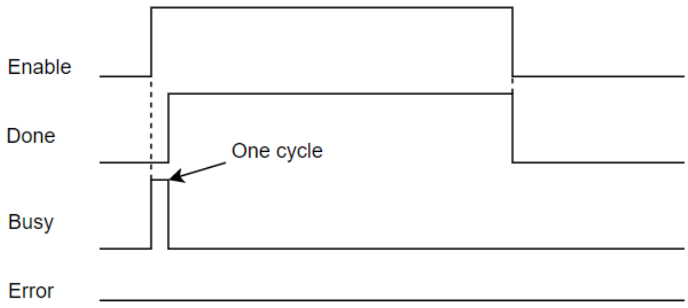
3. Velocity regulation by the MC_Jog and MC_Home commands is not supported, while velocity regulation by other single-axis motion related commands is supported.
4. This command is only used for single-axis motion function blocks and does not support velocity regulation for axis group, master axis, and slave axis.

Multiple Starts of This Command

During the validity period of the Busy signal of the MC_SetOverride command, if the second MC_SetOverride command is triggered, the system displays that the first command remains running and the second command has an error.

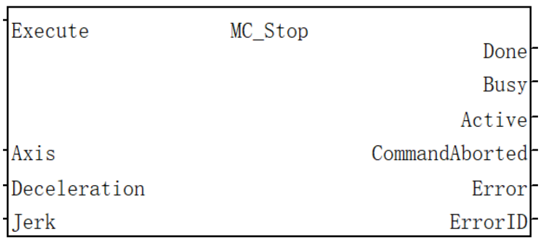
Timing diagram

During the execution of the MC_MoveVelocity command, the MC_SetOverride command is triggered.



3.21.18 MC_Stop

Graphic Block



16-Bit command	-					
32-Bit command	MC_Stop: Stop					
Operand	**Name**	**Description**	**Nullable**	**Default value**	**Range**	**Data Type**
S1	Axis	Axis name/axis ID	No	-	-	WORD
S2	Deceleration	Deceleration	Yes	1000	Positive number	REAL
S3	Jerk	Jerk value 0: T-type acceleration and deceleration >0: S-type acceleration and deceleration	Yes	0	Positive/0	REAL
D1	Done	Completion flag	Yes	OFF	ON/OFF	BOOL
D2	Busy	Ongoing execution flag	Yes	OFF	ON/OFF	BOOL
D3	Active	Alternate waiting flag	Yes	OFF	ON/OFF	BOOL
D4	CommandAborted	Execution interrupt flag	Yes	OFF	ON/OFF	BOOL
D5	Error	Command error flag	Yes	OFF	ON/OFF	BOOL
D6	ErrorID	Error code	Yes	0	-	WORD

Operand	Const	Y	M	S	D	R	Custom Variable
S1	√	-	-	-	-	-	
S2	√	-	-	-	√	√	√
S3	√	-	-	-	√	√	√
D1	-	√	√	√	-	-	√
D2	-	√	√	√	-	-	√
D3	-	√	√	√	-	-	√
D4	-	√	√	√	-	-	√
D5	-	√	√	√	-	-	√
D6	-	-	-	-	√	√	√

Function Description

1. The MC_Stop command is used to control the bus servo axis and local pulse axis to achieve a single-axis stop, and triggers the rising edge.
2. On the rising edge of Execute, the function block locks input parameters such as Deceleration and Jerk, and the axis is in the Stopping state and performs the decelerating stop motion. After the deceleration is completed, the Done signal is valid and remains in the Stopping state during the Execute=ON period. In case of Execute=OFF and Done=ON, the axis switches from the Stopping state to the Standstill state.
3. Jerk is used to set the type of the speed curve. Jerk=0 represents the T-type curve acceleration and deceleration, while Jerk>0 represents the S-type curve acceleration and deceleration.
4. This command is only used for single-axis motion function blocks and does not support axis group, master axis, slave axis, etc.
5. When the axis is in the Stopping state, other motion commands cannot interrupt this command. Only after the axis is restored to the Standstill state through the falling edge of energy flow of this command can other motion control commands run.

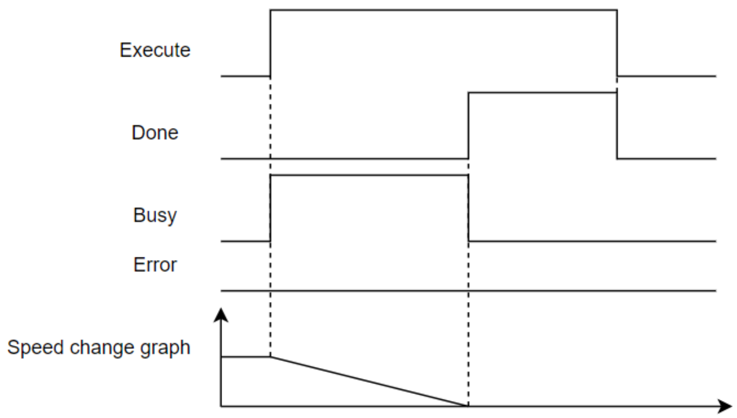
Resetting This Command

During the validity period of the Busy signal of the MC_Stop command, if the MC_Stop command is triggered again, it will re-plan with new target parameters according to the current motion position, velocity, etc.

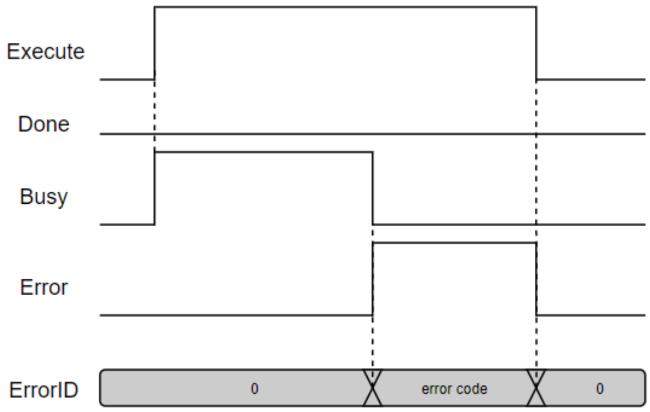
Multiple Starts of This Command

During the validity period of the Busy signal of the MC_Stop command, if the second MC_Stop command is triggered, the system displays that the first MC_Stop command remains running and the second MC_Stop command has an error.

Timing diagram

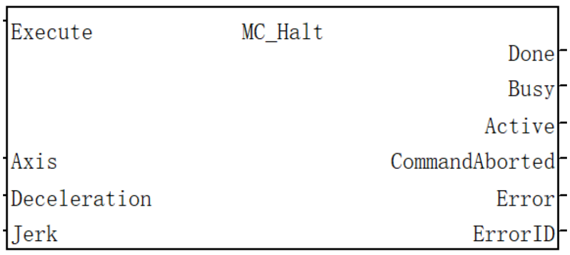


During the command operation, the driver experiences a fault.



3.21.19 MC_Halt

Graphic Block



16-Bit command	-					
32-Bit command	MC_Halt: Halt					
Operand	**Name**	**Description**	**Nullable**	**Default value**	**Range**	**Data Type**
S1	Axis	Axis name/axis ID	No	-	-	WORD
S2	Deceleration	Deceleration	Yes	1000	Positive number	REAL
S3	Jerk	Jerk value 0: T-type acceleration and deceleration >0: S-type acceleration and deceleration	Yes	0	Positive/0	REAL
D1	Done	Completion flag	Yes	OFF	ON/OFF	BOOL
D2	Busy	Ongoing execution flag	Yes	OFF	ON/OFF	BOOL
D3	Active	Alternate waiting flag	Yes	OFF	ON/OFF	BOOL
D4	CommandAborted	Execution interrupt flag	Yes	OFF	ON/OFF	BOOL
D5	Error	Command error flag	Yes	OFF	ON/OFF	BOOL
D6	ErrorID	Error code	Yes	0	-	WORD

Operand	**Const**	**Y**	**M**	**S**	**D**	**R**	**Custom Variable**
S1	√	-	-	-	-	-	
S2	√	-	-	-	√	√	√
S3	√	-	-	-	√	√	√
D1	-	√	√	√	-	-	√
D2	-	√	√	√	-	-	√

Operand	Const	Y	M	S	D	R	Custom Variable
D3	-	√	√	√	-	-	√
D4	-	√	√	√	-	-	√
D5	-	√	√	√	-	-	√
D6	-	-	-	-	√	√	√

Function Description

1. The MC_Halt command is used to control the bus servo axis and local pulse axis to achieve a single-axis stop, and triggers the rising edge.
2. On the rising edge of Execute, the function block locks input parameters such as Deceleration and Jerk, and the axis is in the DiscreteMotion state and performs the decelerating stop motion. After the deceleration is completed, the Done signal is valid, and the axis is restored to the Standstill state.
3. Jerk is used to set the type of the speed curve. Jerk=0 represents the T-type curve acceleration and deceleration, while Jerk>0 represents the S-type curve acceleration and deceleration.
4. This command is only used for single-axis motion function blocks and does not support axis group, master axis, slave axis, etc.

Resetting This Command

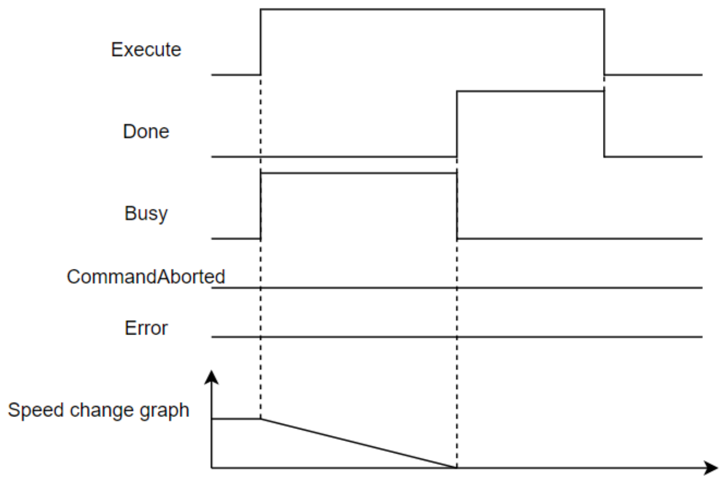
During the validity period of the Busy signal of the MC_Halt command, if the MC_Halt command is triggered again, it will re-plan with new target parameters according to the current motion position, velocity, etc.

Multiple Starts of This Command

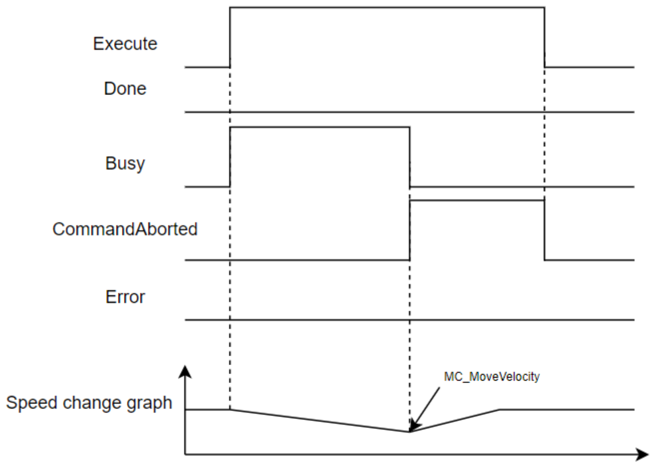
During the validity period of the Busy signal of the MC_Halt command, if the second MC_Halt command is triggered, the second command will re-plan with new target parameters according to the current motion position, velocity, etc., while the first command will be interrupted and invalidated.

Timing diagram

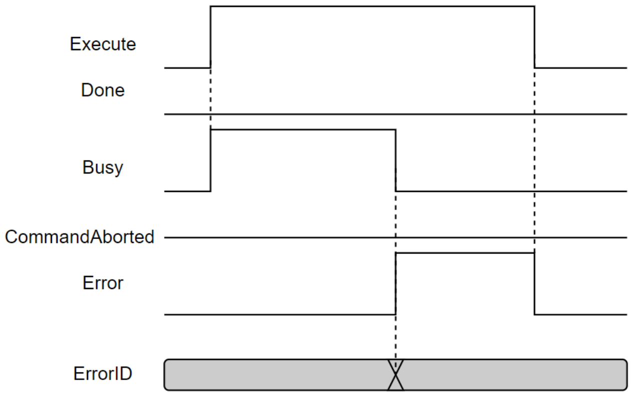
- During the execution of the positioning command, the MC_Halt command is triggered.



- After the MC_Halt command is trigged, the velocity command is called again to interrupt the running of the MC_Halt command.



- During the execution of the MC_Halt command, the machine stops due to a fault.



3.21.20 MC_ImmediateStop

Graphic Block

16-Bit command	MC_ImmediateStop: Immediate Stop					
32-Bit command	-					
Operand	**Name**	**Description**	**Nullable**	**Default value**	**Range**	**Data Type**
S1	Axis	Axis name/axis ID	No	-	-	WORD
D1	Done	Completion flag	Yes	OFF	ON/OFF	BOOL
D2	Busy	Ongoing execution flag	Yes	OFF	ON/OFF	BOOL
D3	Error	Command error flag	Yes	OFF	ON/OFF	BOOL
D4	ErrorID	Error code	Yes	0	-	WORD

Operand	Const	Y	M	S	D	R	Custom Variable
S1	√	-	-	-	-	-	-
D1	-	√	√	√	-	-	√
D2	-	√	√	√	-	-	√
D3	-	√	√	√	-	-	√
D4	-	-	-	-	√	√	√

Function Description

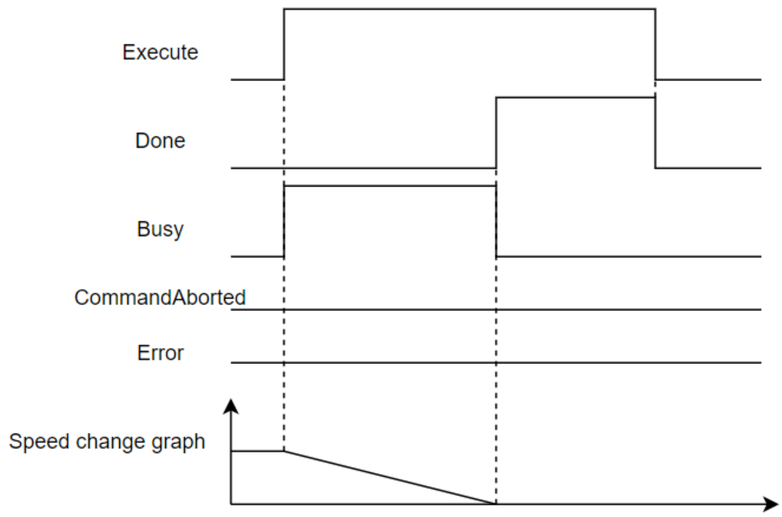
1. The MC_ImmediateStop command is used for single-axis stop, and triggers the rising edge.
2. On the rising edge of Execute, the axis immediately stops. After the stop is completed, the Done signal is valid and remains in the Stopping state during the Execute=ON period. In case of Execute=OFF and Done=ON, the axis switches from the Stopping state to the Standstill state.
3. This command is only used for single-axis motion function blocks and does not support axis group, master axis, slave axis, etc.
4. When the axis is in the Stopping state, other motion commands cannot interrupt this command. Only after the axis is restored to the Standstill state through the falling edge of energy flow of this command can other motion control commands run.

Multiple Starts of This Command

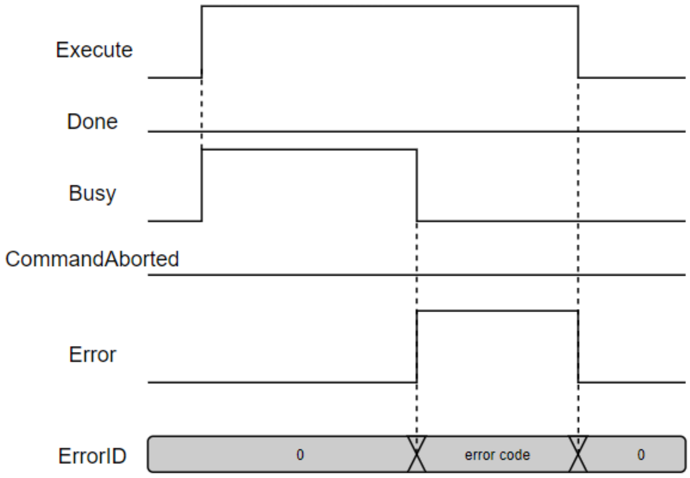
For multiple MC_ImmediateStop commands, the first triggered one shall prevail, while the rest ones report a fault.

Timing diagram

- This command is called after calling the MC_MoveVelocity command.

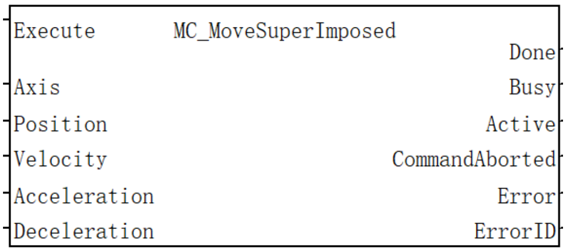


- During the command operation, the driver experiences a fault.



3.21.21 MC_MoveSuperImposed

Graphic Block



16-Bit command	-					
32-Bit command	MC_ MoveSuperImposed: Superimposed Motion					
Operand	**Name**	**Description**	**Nullable**	**Default value**	**Range**	**Data Type**
S1	Axis	Axis name/axis ID	No	-	-	WORD
S2	Position	Compensation distance	No	0	Positive/negative/0	REAL
S3	Velocity	Speed	Yes	100	Positive number	REAL
S4	Acceleration	Acceleration	Yes	1000	Positive number	REAL
S5	Deceleration	Deceleration	Yes	1000	Positive number	REAL
D1	Done	Execution completion flag	Yes	OFF	ON/OFF	BOOL
D2	Busy	Ongoing execution flag	Yes	OFF	ON/OFF	BOOL
D3	Active	Alternate waiting flag	Yes	OFF	ON/OFF	BOOL
D4	CommandAborted	Execution interrupt flag	Yes	OFF	ON/OFF	BOOL
D5	Error	Execution error flag	Yes	OFF	ON/OFF	BOOL
D6	ErrorID	Error code	Yes	0	-	WORD

Operand	Const	Y	M	S	D	R	Custom Variable
S1	√	-	-	-	-	-	

Operand	Const	Y	M	S	D	R	Custom Variable
S2	√	-	-	-	√	√	√
S3	√	-	-	-	√	√	√
S4	√	-	-	-	√	√	√
S5	√	-	-	-	√	√	√
D1	-	√	√	√	-	-	√
D2	-	√	√	√	-	-	√
D3	-	√	√	√	-	-	√
D4	-	√	√	√	-	-	√
D5	-	√	√	√	-	-	√
D6	-	-	-	-	√	√	√

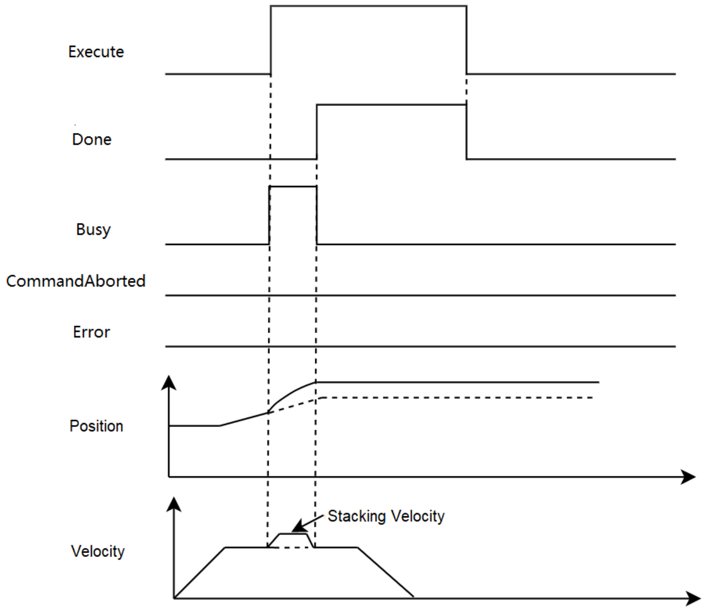
Function Description

On the rising edge of Execute (start), this command superimpose a relative positioning motion on the basis of the original control mode of the axis according to the input parameters. The displacement of the superimposed motion is specified by Position, and the stacking velocity is specified by Velocity.

Processing Methods for Axis in Different Control Modes

1. Single-Axis Positioning Related Commands

If this command is called separately, the system reports an error for this command. If called during the operation of the positioning commands (for absolute positioning or phase positioning), this command performs the superimposed motion. This command will change the trajectory of the original command.



During the execution of this command, if you trigger other commands that enable the axis to handle the CSP mode, such as MC_MoveAbsolute, this command will be interrupted. This command can be stopped by the MC_Stop, MC_Halt, and MC_ImmediateStop commands.

During the validity period of the MC_Halt command, the superimposed motion command cannot be executed.

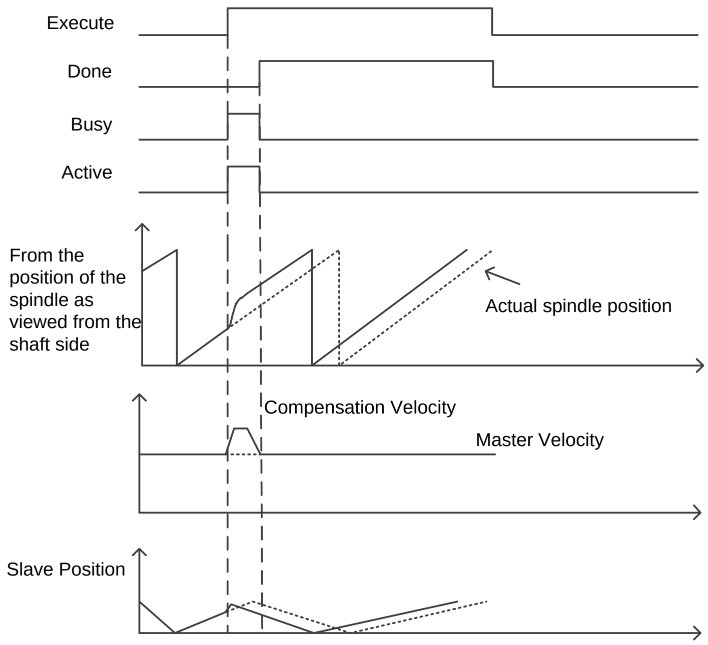
During the execution of this command, if you execute non-CSP motion related commands such as MC_Home, the system reports an error for this command.

2. Axis Group Related Commands

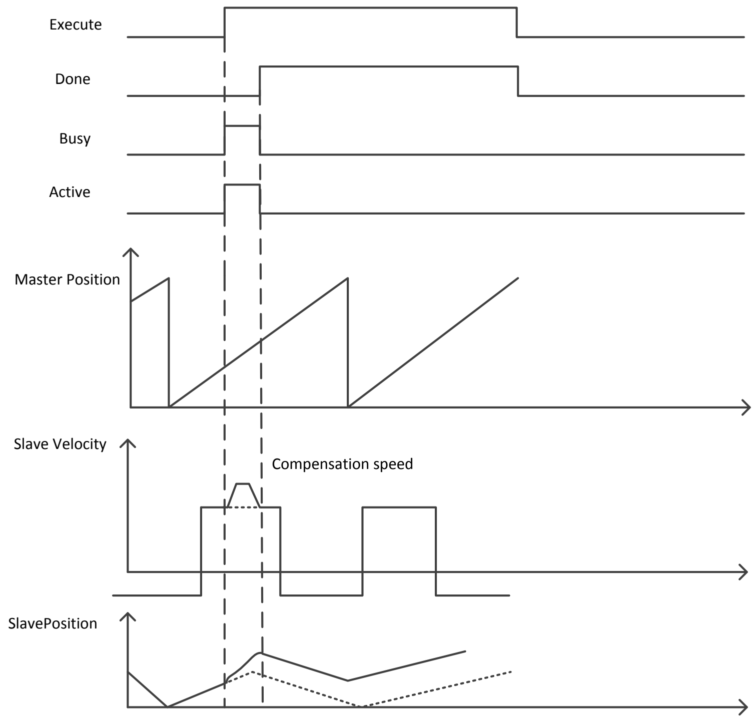
When the axis call this command during the control of the axis group related commands, this command reports an error and does not perform the superimposed motion action.

3. Cam and Gear Related Commands

The operation of the master axis follows the rules of single axis and axis group related commands.



If called when the slave axis follows gears and cams, this command performs the superimposed motion action.



The MC_GearOut and MC_CamOut commands can terminate the execution of this command.

Resetting This Command

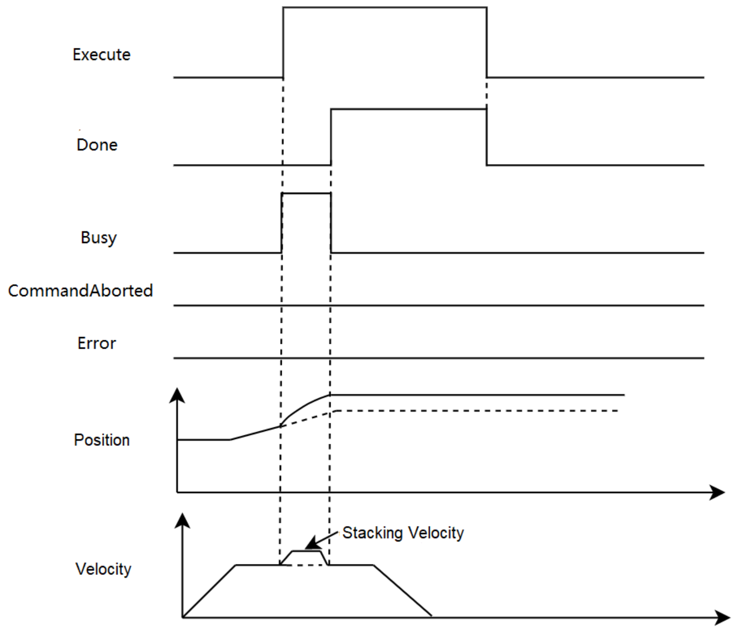
During the validity period of the Busy signal of the MC_MoveSuperImposed command, if the MC_MoveSuperImposed command is triggered again, it will re-plan with new target parameters according to the current motion position, velocity, etc.

Multiple Starts of This Command

During the validity period of the Busy signal of the MC_MoveSuperImposed command, if the second MC_MoveSuperImposed command is triggered, the system reports an error for the second command.

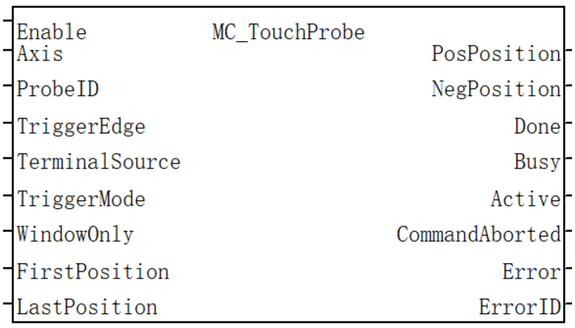
Timing diagram

During the execution of the relative motion command, the MC_MoveSuperImposed command is triggered.



3.21.22 MC_TouchProbe

Graphic Block



16-Bit command	-					
32-Bit command	MC_TouchProbe: Probe					
Operand	**Name**	**Description**	**Nullable**	**Default value**	**Range**	**Data Type**
S1	Axis	Axis name/axis ID	No	-	-	WORD
S2	ProbeID	Probe ID 0: probe 1 1: probe 2	No	0	0~1	INT
S3	TriggerEdge	Edge type 0: only rising edge triggered 1: only falling edge triggered	Yes	0	0~1	INT
S4	Terminal Source	Probe signal source 0: DI terminal 1: encoder Z signal	Yes	0	0~1	INT
S5	Trigger Mode	Trigger type 0: single trigger 1: continuous trigger	Yes	0	0~1	INT

16-Bit command	-					
32-Bit command	MC_TouchProbe: Probe					
Operand	**Name**	**Description**	**Nullable**	**Default value**	**Range**	**Data Type**
S6	Window Only	Probe window settings 0: The window function is turned off, and the probe signal can be detected in all position ranges 1: The window function is turned on, and the probe signal cab be detected only when the current position is ⩾ FirstPosition and ⩽ LastPosition	Yes	OFF	ON/OFF	BOOL
S7	FirstPosition	Start position of probe window	Yes	0	Positive/ negative/0	REAL
S8	LastPosition	End position of probe window	Yes	0	Positive/ negative/0	REAL
D1	PosPosition	Rising edge latch position	Yes	0	Positive/ negative/0	REAL
D2	NegPosition	Falling edge latch position	Yes	0	Positive/ negative/0	REAL
D3	Done	Execution completion flag	Yes	OFF	ON/OFF	BOOL
D4	Busy	Ongoing execution flag	Yes	OFF	ON/OFF	BOOL
D5	Active	Execution validity flag	Yes	OFF	ON/OFF	BOOL
D6	Command Aborted	Execution interrupt flag	Yes	OFF	ON/OFF	BOOL
D7	Error	Command error flag	Yes	OFF	ON/OFF	BOOL
D8	ErrorID	Error code, which displays error information	Yes	0	Positive/0	WORD

Operand	Const	Y	M	S	D	R	Custom Variable
S1	√	-	-	-	-	-	-
S2	√	-	-	-	√	√	
S3	√	-	-	-	√	√	√
S4	√	-	-	-	√	√	√
S5	√	-	-	-	√	√	√
S6		-	√	√	-	-	√
S7	√	-	-	-	√	√	√
S8	√	-	-	-	√	√	√
D1	-	-	-	-	√	√	√
D2	-	-	-	-	√	√	√
D3	-	√	√	√	-	-	√
D4	-	√	√	√	-	-	√
D5	-	√	√	√	-	-	√
D6	-	√	√	√	-	-	√
D7	-	√	√	√	-	-	√
D8	-	-	-	-	√	√	√

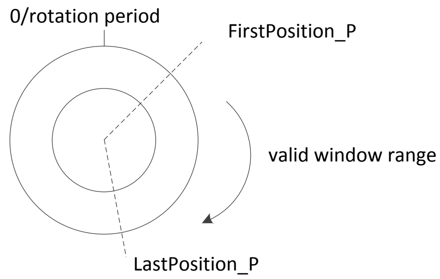
Function Description

1. This command does not support the virtual axis mode, the rising edge updates function block parameters, and the low-level module is invalid.
2. Currently, only the EtherCAT bus axis mode is supported, and the servo driver should be configured with the corresponding PDO.

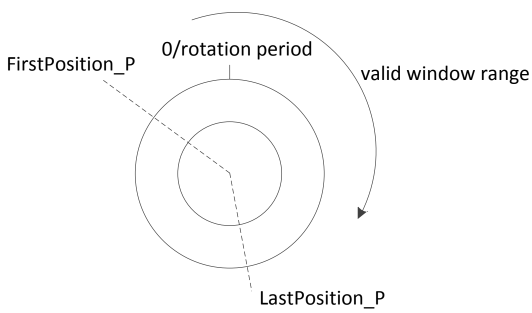
PDO	Description
0x60b8	Probe control word (required)
0x60b9	Probe state word (optional)
0x60ba	Rising edge latch of probe 1, encoder axis latch (optional)
0x60bb	Falling edge latch of probe 1 (optional)
0x60bc	Rising edge latch of probe 2, encoder axis latch (optional)
0x60bd	Falling edge latch of probe 2 (optional)

3. The rising edge latch value is stored in Position, while the falling edge latch value is stored in NegPosition.
4. In case of TriggerEdge=0, only the rising edge trigger (including DI rising edge trigger and encoder axis Z signal trigger) is supported; in case of TriggerEdge=1, only the DI signal falling edge trigger is supported.
5. In case of TerminalSource=0, the DI signal trigger is selected; in case of TerminalSource=1, the encoder axis Z signal trigger is selected.
6. In case of WindowOnly=TRUE, the window function is enabled, and the probe signal can be detected on when the current position is within the window (the window size is determined by the parameters FirstPosition and LastPosition).
7. In the linear mode, the window range is greater than or equal to FirstPosition but less than or equal to LastPosition.
8. In circular mode, the command first uses FirstPosition and LastPosition to calculate the modulus over the cycle to obtain the range positions FirstPosition_P and LastPosition_P within one cycle.

(1) In case of FirstPosition_p<LastPosition_p, the valid window range is shown in the figure below.



(2) In case of FirstPosition_p>LastPosition_p, the valid window range is shown in the figure below.



9. In case of TriggerMode=0, the single trigger mode is selected, and the probe detection ends when a single latch occurs; in case TriggerMode=1, the continuous trigger mode is selected, and every successful trigger outputs the Done signal for one cycle.

10. The latch value triggered by the rising edge and encoder axis Z signal is output at the Position end, while the latch value triggered by the falling edge is output at the NegPosition end.

Resetting This Command

If this command is triggered again during the validity period of its Busy signal, it reads the probe value according to the new probe configuration parameters.

Multiple Starts of This Command

When multiple commands call the same probe channel (probe channels 1 and 2 for selection) of the same axis, if the next command is triggered during the Busy signal validity period of the previous command, the next command will take effect, and the previous command will be interrupted and invalidated.

If two commands call the same axis but different probe channels, they do not affect each other.

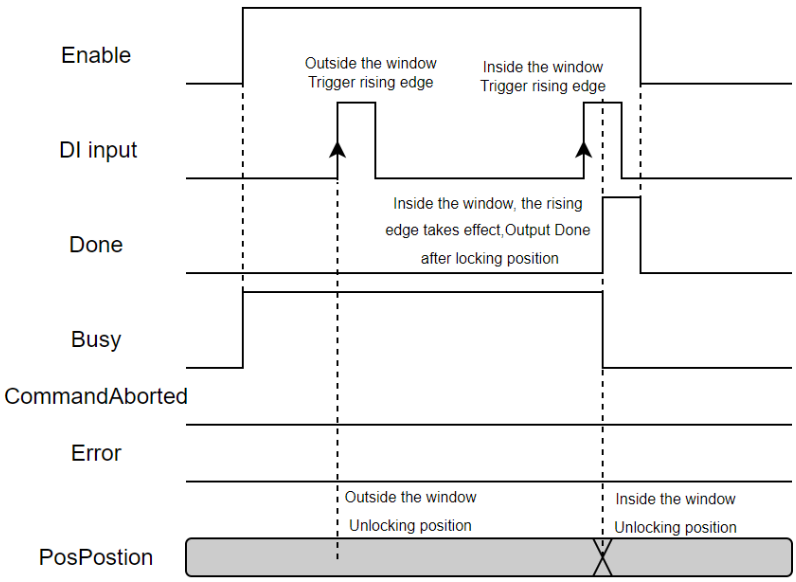
If multiple commands use the same probe channel of the same axis, the last triggered MC_MoveFeed command will interrupt the running MC_TouchProbe command, while the first triggered MC_MoveFeed command will trigger the MC_TouchProbe command to report an error (the probe channel has been occupied by the interrupt fixed length function).

Timing diagram

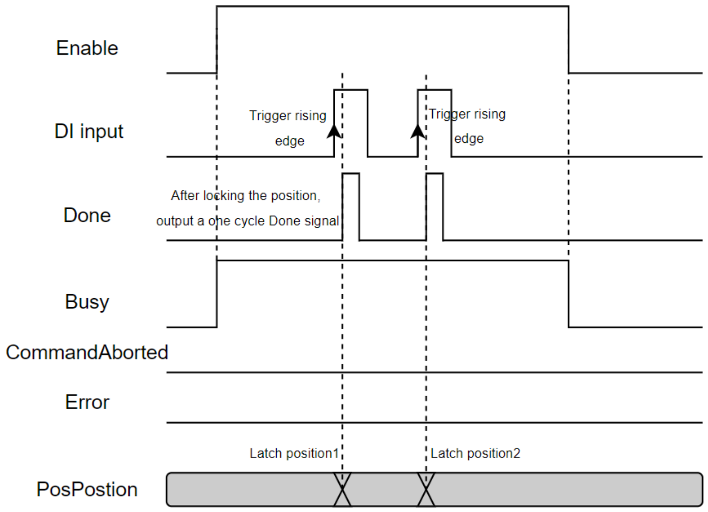
There are two sources for probe signals, one is the rising edge of the encoder Z signal, the other is the rising or falling edge of the switch input. By default, probe channel 1 is triggered by switching value input 1 (DI1), while probe channel 2 is triggered by switching value input 2 (DI2).

Taking the rising edge of switching value input 1 as the signal source as an example, the timing of the probe is shown below.

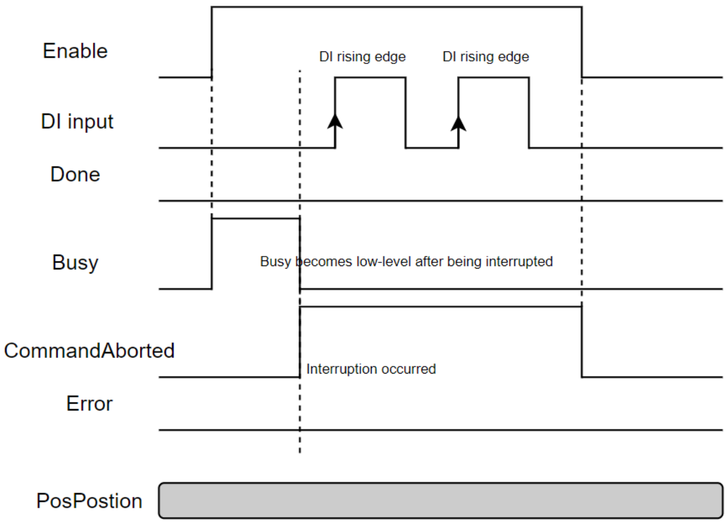
- The rising edge of probe 1 is valid, the DI terminal is triggered in the single trigger mode, and the window function WindowOnly is TRUE.



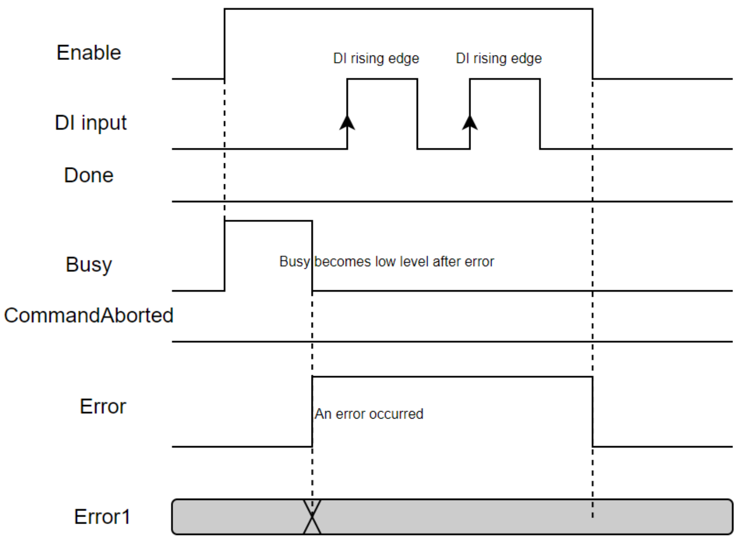
- The rising edge of probe 1 is valid, the DI terminal is triggered in the continuous trigger mode, and the window function WindowOnly is FALSE.



- Probe 1 is interrupted by a probe-related command, and the window function WindowOnly is FALSE.

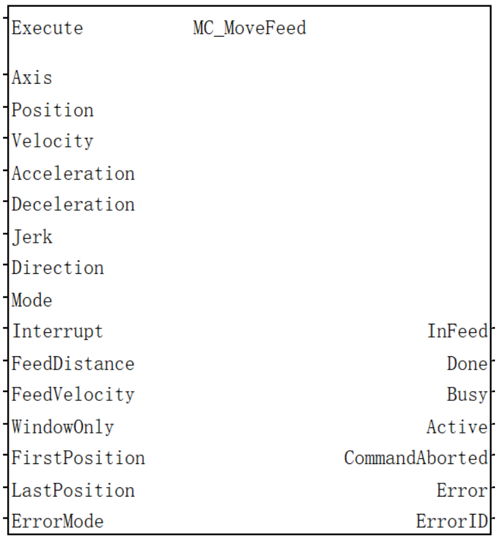


- A fault occurs in probe 1.



3.21.23 MC_MoveFeed

Graphic Block



16-Bit command	-					
32-Bit command	MC_MoveFeed: Interrupt Fixed Length					
Operand	**Name**	**Description**	**Nullable**	**Default value**	**Range**	**Data Type**
S1	Axis	Axis name/axis ID	No	-	-	WORD
S2	Position	Target position	Yes	0	Positive/negative/0	REAL
S3	Velocity	Target velocity	Yes	0	Positive number	REAL
S4	Acceleration	Acceleration	Yes	0	Positive number	REAL
S5	Deceleration	Deceleration	Yes	0	Positive number	REAL
S6	Jerk	Jerk value 0: T-type acceleration and deceleration >0: S-type acceleration and deceleration	Yes	0	Positive/0	REAL
S7	Direction	Direction	Yes	0	In case of Mode=0 and circular mode: 0: positive; 1: negative 2: minimum distance 3: current In case of Mode=2: 0: positive; 1-3: negative	INT
S8	Mode	Mode 0: absolute positioning mode 1: relative positioning mode 2: velocity mode	Yes	0	0~2	INT
S9	Interrupt	Interrupt source selection	Yes	0	0~1	INT

16-Bit command	-					
32-Bit command	MC_MoveFeed: Interrupt Fixed Length					
Operand	**Name**	**Description**	**Nullable**	**Default value**	**Range**	**Data Type**
		0: probe 1; 1: probe 2				
S10	FeedDistance	Displacement after reaching interrupt source Positive: After the interrupt source is reached, the axis runs for the distance specified by FeedDistance in the direction of original motion Negative: After the interrupt source is reached, the axis slows down to zero first, and then runs for the distance specified by FeedDistance in the opposite direction of the original motion	Yes	0	Positive/negative/0	REAL
S11	FeedVelocity	Target velocity after reaching interruption	Yes	0	Positive number	REAL
S12	WindowOnly	Enable interrupt source window 0: Disable window detection function 1: Enable window detection function	Yes	OFF	ON/OFF	BOOL
S13	FirstPosition	Start position of interrupt source window	Yes	0	Positive/negative/0	REAL
S14	LastPosition	End position of interrupt source window	Yes	0	Positive/negative/0	REAL
S15	ErrorMode	Fault mode OFF: If the probe signal has not yet arrived after the position specified by Position is reached, the command does not report an error and	Yes	OFF	ON/OFF	BOOL

16-Bit command	-					
32-Bit command	MC_MoveFeed: Interrupt Fixed Length					
Operand	**Name**	**Description**	**Nullable**	**Default value**	**Range**	**Data Type**
		continues to wait for the probe signal ON: If the probe signal has not yet arrived after the position specified by Position is reached, the function block reports an error				
D1	InFeed	Interrupt signal validity	Yes	OFF	ON/OFF	BOOL
D2	Done	Execution completion flag	Yes	OFF	ON/OFF	BOOL
D3	Busy	Ongoing execution flag	Yes	OFF	ON/OFF	BOOL
D4	Active	Execution validity flag	Yes	OFF	ON/OFF	BOOL
D5	CommandAborted	Execution interrupt flag	Yes	OFF	ON/OFF	BOOL
D6	Error	Command error flag	Yes	OFF	ON/OFF	BOOL
D7	ErrorID	Error code, which displays error information	Yes	0	Positive/0	WORD

Operand	Const	Y	M	S	D	R	Custom Variable
S1	√	-	-	-	-	-	-
S2	√	-	-	-	√	√	√
S3	√	-	-	-	√	√	√
S4	√	-	-	-	√	√	√
S5	√	-	-	-	√	√	√
S6	√	-	-	-	√	√	√
S7	√	-	-	-	√	√	√
S8	√	-	-	-	√	√	√
S9	√	-	-	-	√	√	√
S10	√	-	-	-	√	√	√
S11	√	-	-	-	√	√	√
S12	-	-	√	√			√
S13	√	-	-	-	√	√	√
S14	√	-	-	-	√	√	√
S15	√	-	-	-	√	√	√
D1	-	√	√	√	-	-	√
D2	-	√	√	√	-	-	√
D3	-	√	√	√	-	-	√
D4	-	√	√	√	-	-	√

Operand	Const	Y	M	S	D	R	Custom Variable
D5	-	√	√	√	-	-	√
D6	-	√	√	√	-	-	√
D7	-	-	-	-	√	√	√

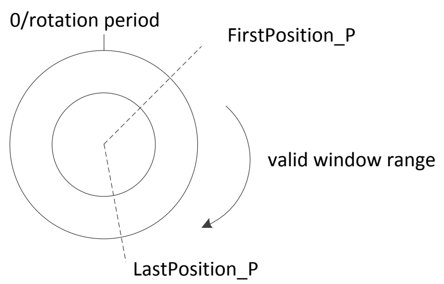
Function Description

1. This command does not support the virtual axis mode, the rising edge updates function block parameters, and pulling down the module does not interrupt the existing motion.
2. Currently, only the EtherCAT bus axis mode is supported, and the servo driver should be configured with the corresponding PDO.

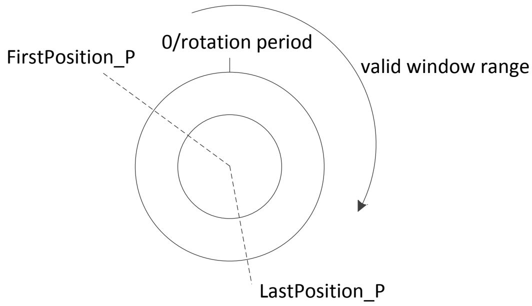
PDO	Description
0x60b8	Probe control word (required)
0x60b9	Probe state word (optional)
0x60ba	Rising edge latch of probe 1, encoder axis latch (optional)
0x60bc	Rising edge latch of probe 2, encoder axis latch (optional)

3. Before the interrupt signal arrives, the axis moves according to the Mode parameter settings. In case of Mode=0, the axis moves in the absolute mode; in case of Mode=1, the axis moves in the relative mode; in case of Mode=2, the axis moves in the velocity mode.
4. Direction can take effect in the following two modes: Mode=0, where the axis is set to circular mode, and its motion mode is consistent with the description in MC_MoveAbsolute; Mode=2, where Direction=0 means clockwise and Direction=1 means counterclockwise.
5. FeedDistance represents the distance traveled after the interrupt signal arrives, where a positive number indicates that the axis moves for the specified distance in the direction of the original motion after the interrupt source arrives, while a negative number indicates that the axis first slows down zero and then moves the specified distance in the opposite direction of the original motion after the interrupt source arrives;
6. In case of WindowOnly=TRUE, the window function is enabled, and the probe signal can be detected on when the current position is within the window (the window size is determined by the parameters FirstPosition and LastPosition);
7. In the linear mode, the window range is greater than or equal to FirstPosition but less than or equal to LastPosition;
8. In circular mode, the command first uses FirstPosition and LastPosition to calculate the modulus over the cycle to obtain the range positions FirstPosition_P and LastPosition_P within one cycle;

(1) In case of FirstPosition_p<LastPosition_p, the valid window range is shown in the figure below.



(2) In case of FirstPosition_p>LastPosition_p, the valid window range is shown in the figure below.



9. When Mode=0 or Mode=1 is configured, if the probe signal has not arrived after the distance is completed, the system takes the corresponding action depending on the value of ErrorMode: in case of ErrorMode=FALSE, it keeps the Busy state and continues to wait for the probe signal; in case of ErrorMode=TRUE, it reports an error.
10. Starting the interrupt fixed length command interrupts probe commands that occupy the same probe channel of the same axis. But the probe command cannot interrupt the interrupt fixed length command (if the channel is occupied, the probe command reports an error, the interrupt fixed length command runs normally).

Resetting This Command

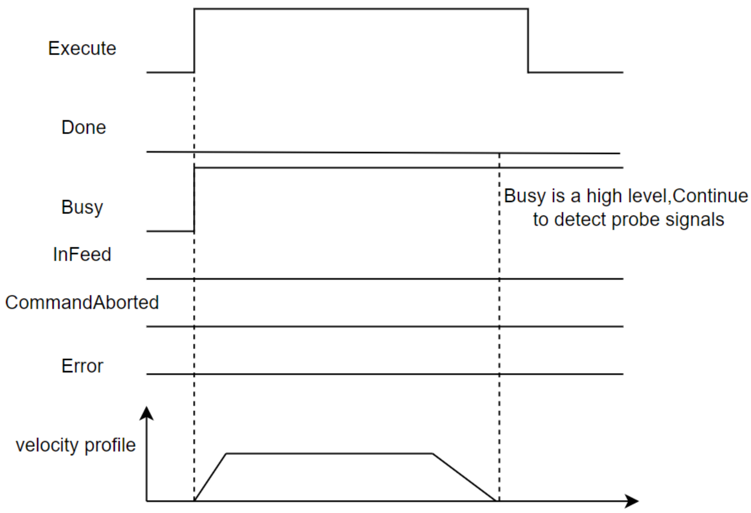
If this command is triggered again during the validity period of its Busy signal, it re-plans the motion according to the new configuration parameters.

Multiple Starts of This Command

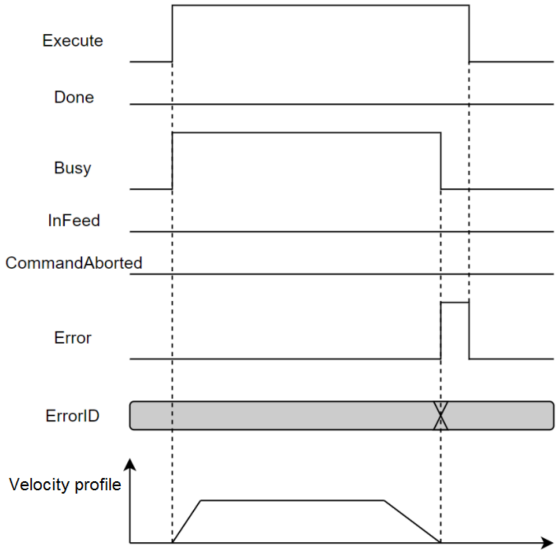
When multiple commands call the same axis, if the next command is triggered during the Busy signal validity period of the previous command, the next command will take effect, and the previous command will be interrupted and invalidated.

Timing diagram

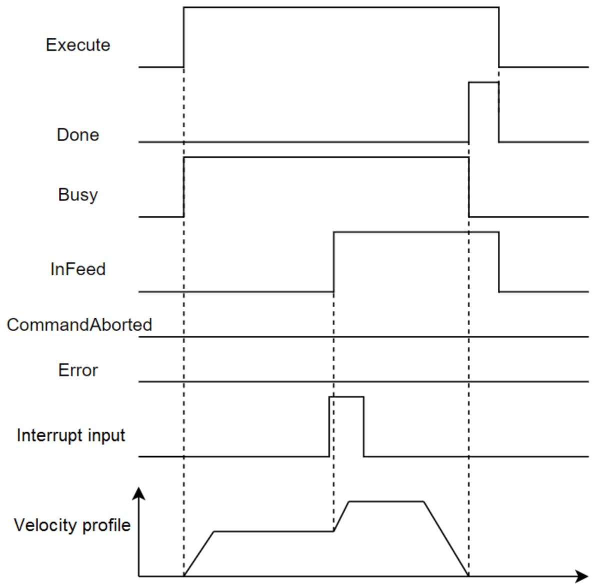
- When the relative positioning and absolute positioning modes are selected, the motion ends without triggering an interrupt signal, and ErrorMode is OFF.



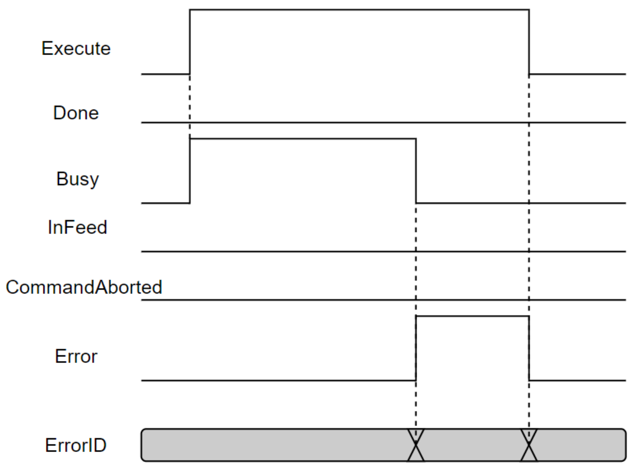
- When the relative positioning and absolute positioning modes are selected, the motion ends without triggering an interrupt signal, and ErrorMode is ON.



- When the relative positioning, absolute positioning, and velocity modes are selected, an interrupt signal arrives during the motion.

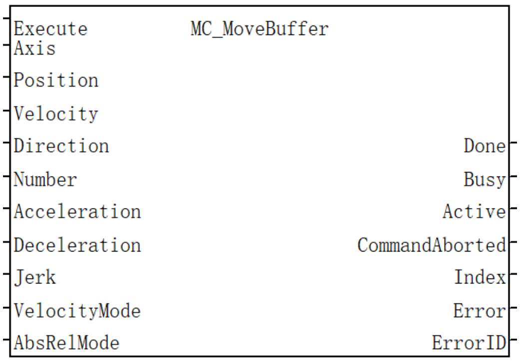


- An error occurs during the command execution.



3.21.24 MC_MoveBuffer

Graphic Block



16-Bit command	-					
32-Bit command	MC_MoveBuffer Multi-Segment Positioning					
Operand	**Name**	**Description**	**Nullable**	**Default value**	**Range**	**Data Type**
S1	Axis	Axis name/axis ID	No	-	-	WORD
S2	Position	Target position	No	0	Positive/ negative/0	REAL [16]
S3	Velocity	Target velocity	No	100	Positive number	REAL [16]
S4	Direction	Direction (valid in absolute mode) 0: positive (velocity ＞ 0) 1: negative (velocity ＜ 0) 2: minimum distance 3: current	No	-	0~3	INT[16]
S5	Number	Number of buffer queues	No	-	1-16	INT
S6	Acceleration	Acceleration	Yes	1000	Positive number	REAL
S7	Deceleration	Deceleration	Yes	1000	Positive number	REAL
S8	Jerk	Jerk value 0: T-type acceleration and deceleration >0: S-type acceleration and deceleration	Yes	0	Positive/0	REAL
S9	VelocityMode	Velocity switching mode 0: Slows down to 0 and starts the next segment 1: Maintains the current velocity and starts the next segment	Yes	0	0~1	INT
S10	AbsRelMode	Positioning mode 0: absolute positioning 1: relative positioning	Yes	0	0~1	INT
D1	Done	Command execution completion	Yes	OFF	ON/OFF	BOOL
D2	Busy	Ongoing execution flag	Yes	OFF	ON/OFF	BOOL
D3	Active	Execution validity flag	Yes	OFF	ON/OFF	BOOL
D4	Command Aborted	Execution interrupt flag	Yes	OFF	ON/OFF	BOOL
D5	Index	Segment being executed	Yes	0	0~15	INT
D6	Error	Error flag	Yes	OFF	ON/OFF	BOOL
D7	ErrorID	Error code	Yes	0	-	WORD

Operand	Const	Y	M	S	D	R	Custom Variable
S1	√	-	-	-			
S2	√	-	-	-	√	√	√
S3	√	-	-	-	√	√	√
S4	√	-	-	-	√	√	√
S5	√	-	-	-	√	√	√
S6	√	-	-	-	√	√	√
S7	√	-	-	-	√	√	√
S8	√	-	-	-	√	√	√
S9	√	-	-	-	√	√	√
S10	√	-	-	-	√	√	√
D1	-	√	√	√	-	-	√
D2	-	√	√	√	-	-	√
D3	-	√	√	√	-	-	√
D4	-	√	√	√	-	-	√
D5	-	√	√	√	-	-	√
D6	-	-	-	-	√	√	√

Function Description

This command is used for the multi-segment positioning function the bus servo axis or local pulse axis, and is valid to the rising edge.

After the rising edge of the command is triggered, the current input parameters are latched, and the axis run the absolute positioning (AbsRelMode==0) or relative positioning (AbsRelMode==1) function in the buffer mode according to the configuration of the AbsRelMode mode. This command supports caching up to 16 segments of positions.

- Position: The target position, which is of array type and supports up to 16 levels. It is used to set the absolute target position of the axis in the absolute mode or the relative target position of the axis in the relative mode.
- Velocity: The target velocity, which is of array type and supports up to 16 levels. It is used to set the target velocity.
- Direction: The target direction of the rotary axis in the absolute positioning mode. It has the same meaning as Direction in the MC_MoveAbsolute command.
- Number: The valid data length in the cache queue, which ranges between 1 and 16. Exceeding this range will result in an error.
- VelocityMode: The velocity switching mode, which is used to command the velocity transition mode between two target positions. VelocityMode=0 means decelerating to 0 and then starting the next positioning trajectory; VelocityMode=1 means that the transition velocity between the two target positions is the target velocity of the previous command (note that VelocityMode=1 is temporarily invalid in the current version).
- AbsRelMode: The positioning mode, where AbsRelMode=0 indicates that the current command is in the absolute positioning mode, while AbsRelMode=1 indicates that the current command is in the relative positioning mode.

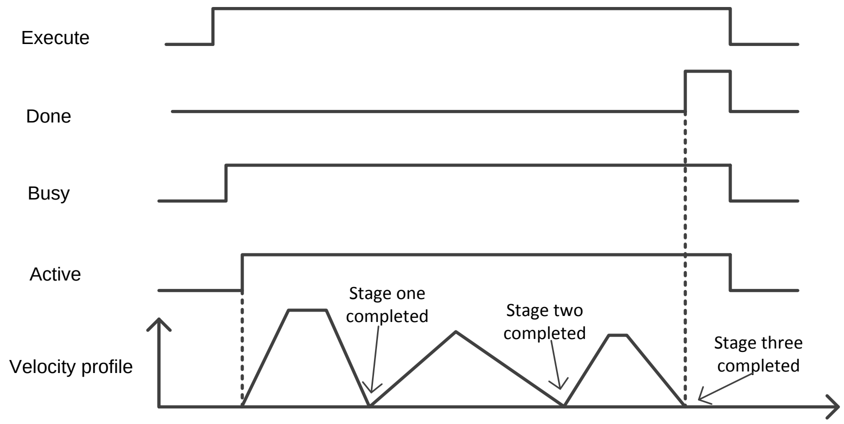
Interrupt

During the execution of this command, the axis is in the DiscreteMotion state, and other commands that allow the axis to be in the DiscreteMotion state or meet the state switching of the PLCopen state machine can interrupt this command. When this command is interrupted, the CommandAborted signal output is valid.

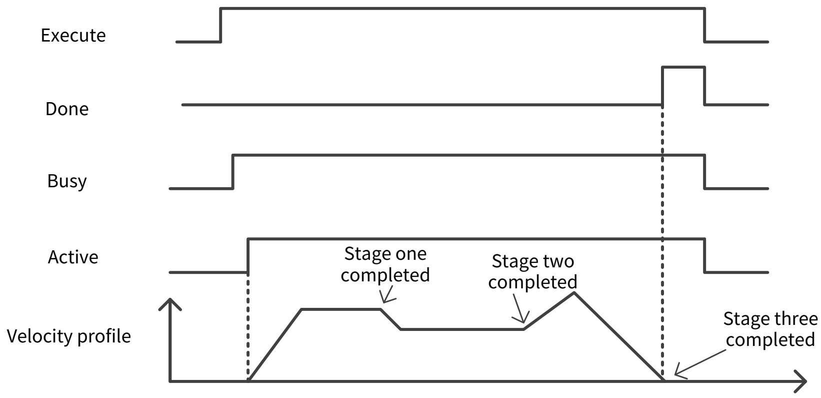
- In case of Execute=ON and invalid Done signal, when the soft and hard limits are triggered, the CommandAborted signal output is valid.
- In case of Execute=ON and invalid Done signal, when an error occurs in the servo, the CommandAborted signal output is valid.
- In case of Execute=ON and invalid Done signal, when disable is triggered, the CommandAborted signal output is valid.

Timing diagram

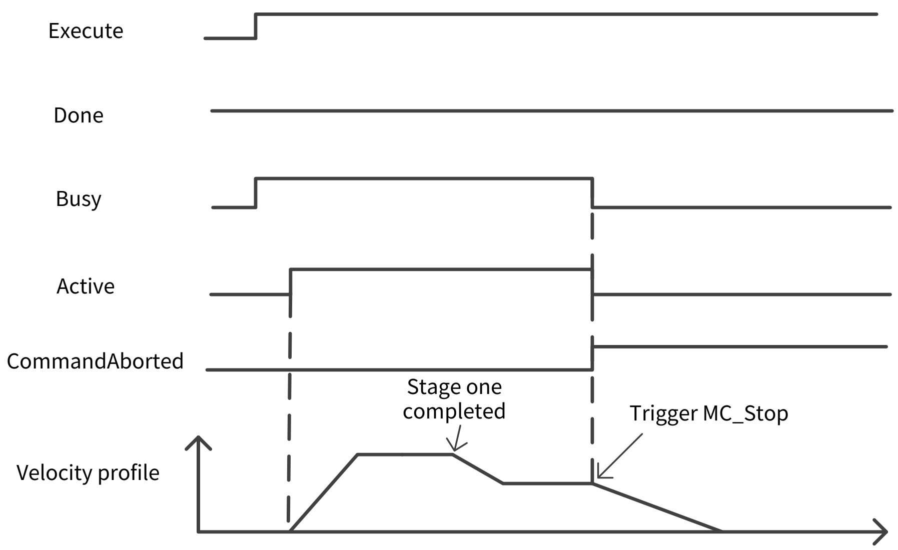
The 3-segment buffer is set, with VelocityMode=0



The 3-segment buffer is set, with VelocityMode=1

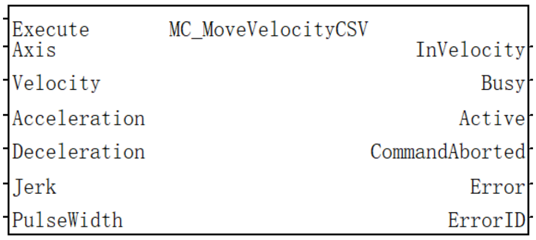


The 3-segment buffer is set, but interrupted by the MC_Stop command during its execution



3.21.25 MC_MoveVelocityCSV

Graphic Block



16-Bit command	-					
32-Bit command	MC_MoveVelocityCSV: Velocity					
Operand	**Name**	**Description**	**Nullable**	**Default value**	**Range**	**Data Type**
S1	Axis	Axis name/axis ID	No	-	-	WORD
S2	Velocity	Target velocity	Yes	100	Positive/negative/0	REAL
S3	Acceleration	Acceleration	Yes	1000	Positive number	REAL
S4	Deceleration	Deceleration	Yes	1000	Positive number	REAL
S5	Jerk	Jerk value 0: T-type acceleration and deceleration >0: S-type acceleration and deceleration	Yes	0	Positive/0	REAL
S6	PulseWidth	Pulse width, in units of 0.01%	Yes	5000	1-9999	INT
D1	InVelocity	Reach target velocity	Yes	OFF	ON/OFF	BOOL
D2	Busy	Ongoing execution flag	Yes	OFF	ON/OFF	BOOL
D3	Active	Execution validity flag	Yes	OFF	ON/OFF	BOOL
D4	CommandAborted	Execution interrupt flag	Yes	OFF	ON/OFF	BOOL
D5	Error	Error flag	Yes	OFF	ON/OFF	BOOL
D6	ErrorID	Error code	Yes	0	-	WORD

Operand	**Const**	**Y**	**M**	**S**	**D**	**R**	**Custom Variable**
S1	√	-	-	-	-	-	
S2	√	-	-	-	√	√	√
S3	√	-	-	-	√	√	√
S4	√	-	-	-	√	√	√

Operand	Const	Y	M	S	D	R	Custom Variable
S5	√	-	-	-	√	√	√
S6	√	-	-	-	√	√	√
D1	-	√	√	√	-	-	√
D2	-	√	√	√	-	-	√
D3	-	√	√	√	-	-	√
D4	-	√	√	√	-	-	√
D5	-	√	√	√	-	-	√
D6	-	-	-	-	√	√	√

Function Description

1. The MC_MoveVelocityCSV command is used to control the bus servo axis or local pulse axis to achieve synchronous velocity control, and is valid to the rising edge.
2. The state machine of the running axis is in the ContinuousMotion mode.
3. If you are using a bus servo axis, it is necessary to configure the relevant PDO object dictionaries 0x6060, 0x6061, 0x60FF, 0x6083, and 0x6084. In the bus mode, this command and other motion commands such as MC_MoveAbsolute and MC_Stop can interrupt each other.
4. To use pulse axis, you need to configure "Output Device", and select a pulse mode under "Output Mode" in "Mode Setting" in the axis configuration, among which modifications can be made to the PulseWidth parameter in "pulse + direction" and "forward-reverse pulse train", while modifications to the PulseWidth parameter are invalid in " orthogonal coding pulse".

Resetting This Command

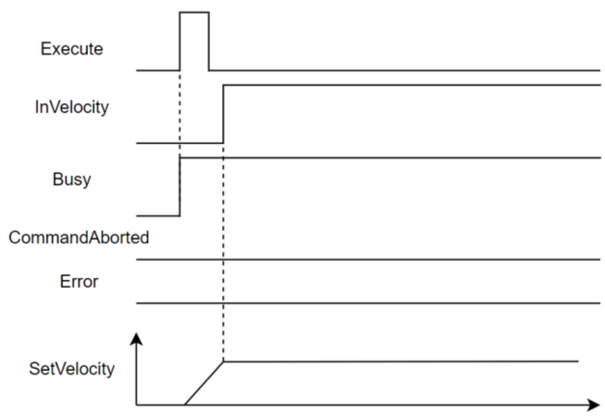
During the validity period of the Busy signal of the MC_MoveVelocityCSV, if the MC_MoveVelocity command is triggered again, it will re-plan with new target parameters according to the current motion position, velocity, etc.

Multiple Starts of This Command

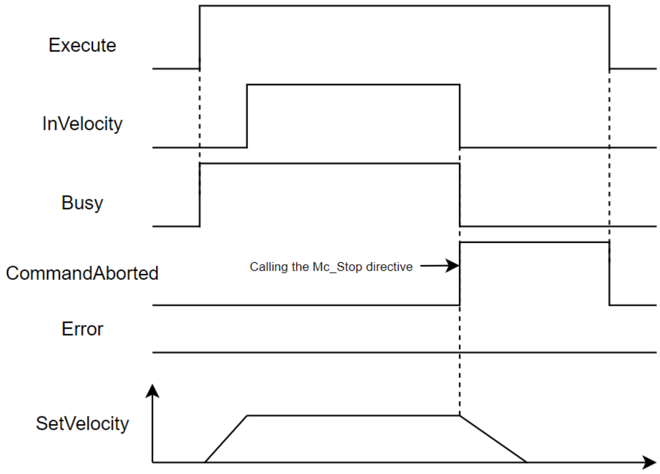
During the validity period of the Busy signal of the MC_MoveVelocity command, if the second MC_MoveVelocity command is triggered, the second command will re-plan with new target parameters according to the current motion position, velocity, etc., while the first command will be interrupted and invalidated.

Timing diagram

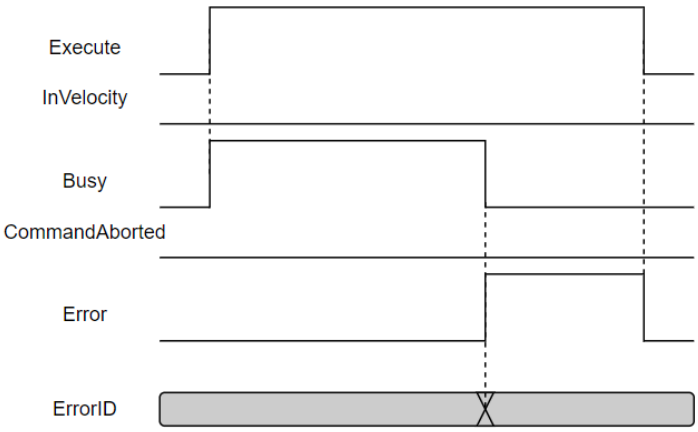
In the StandStill state, the axis calls this command to perform the continuous motion with a T-typed curve.



During running, the axis is interrupted by the MC_Stop command.

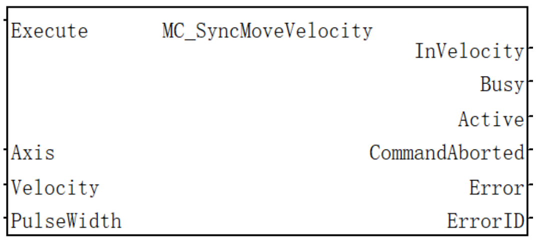


During the axis acceleration, the driver experiences a fault.



3.21.26 MC_MoveSyncVelocity

Graphic Block



16-Bit command	-					
32-Bit command	MC_SyncMoveVelocity: Velocity					
Operand	**Name**	**Description**	**Nullable**	**Default value**	**Range**	**Data Type**
S1	Axis	Axis name/axis ID	No	-	-	WORD
S2	Velocity	Target velocity	Yes	100	Positive number	REAL
S3	PulseWidth	Pulse width, in units of 0.01%	Yes	5000	1-9999	INT
D1	InVelocity	Reach target velocity	Yes	OFF	ON/OFF	BOOL
D2	Busy	Ongoing execution flag	Yes	OFF	ON/OFF	BOOL
D3	Active	Execution validity flag	Yes	OFF	ON/OFF	BOOL

16-Bit command	-					
32-Bit command	MC_SyncMoveVelocity: Velocity					
Operand	**Name**	**Description**	**Nullable**	**Default value**	**Range**	**Data Type**
D4	CommandAborted	Execution interrupt flag	Yes	OFF	ON/OFF	BOOL
D5	Error	Error flag	Yes	OFF	ON/OFF	BOOL
D6	ErrorID	Error code	Yes	0	-	WORD

Operand	Const	Y	M	S	D	R	Custom Variable
S1	√	-	-	-	-	-	
S2	√	-	-	-	√	√	√
S3	√	-	-	-	√	√	√
D1	-	√	√	√	-	-	√
D2	-	√	√	√	-	-	√
D3	-	√	√	√	-	-	√
D4	-	√	√	√	-	-	√
D5	-	√	√	√	-	-	√
D6	-	-	-	-	√	√	√

Function Description

1. The MC_MoveSyncVelocity command is used to control the bus servo axis or local pulse axis to achieve synchronous velocity control, and is valid at high levels.
2. When using the bus axis, the PDO object dictionary should be configured with 0x6061, 0x6061, and 0x60FF, and this command and other motion commands such as MC_MoveAbsolute and MC_Stop can interrupt each other.
3. Calling this command enables the velocity to change at the maximum servo acceleration or deceleration.
4. The state machine of the running axis is in the ContinuousMotion mode.
5. To use pulse axis, you need to configure "Output Device", and select a pulse mode under "Output Mode" in "Mode Setting" in the axis configuration, among which modifications can be made to the PulseWidth parameter in "pulse + direction" and "forward-reverse pulse train", while modifications to the PulseWidth parameter are invalid in " orthogonal coding pulse".

Resetting This Command

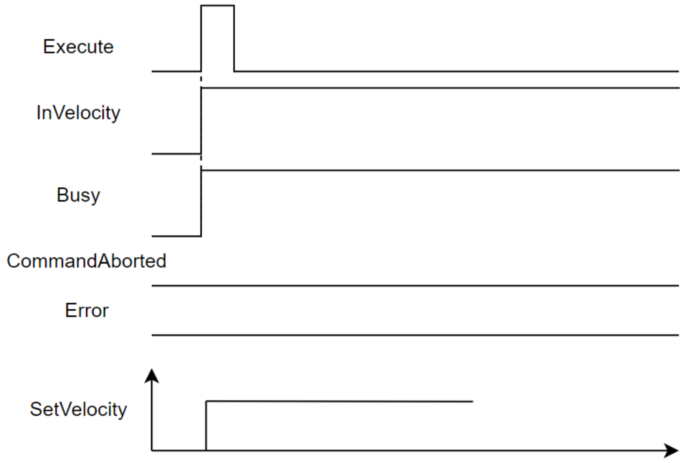
During the validity period of the Busy signal of the MC_MoveSyncVelocity command, if the MC_MoveSyncVelocity command is triggered again, the axis will change to the new target velocity at the maximum servo acceleration or deceleration according to the current motion position, velocity, etc.

Multiple Starts of This Command

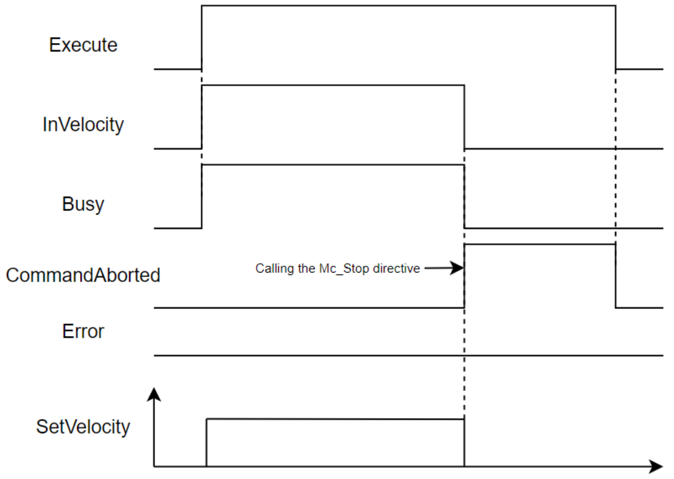
During the validity period of the Busy signal of the MC_MoveSyncVelocity command, if the second MC_MoveSyncVelocity command is triggered, the second command will change to the new target velocity at the maximum servo acceleration or deceleration according to the current motion position, velocity, etc., while the first command will be interrupted and invalidated.

Timing diagram

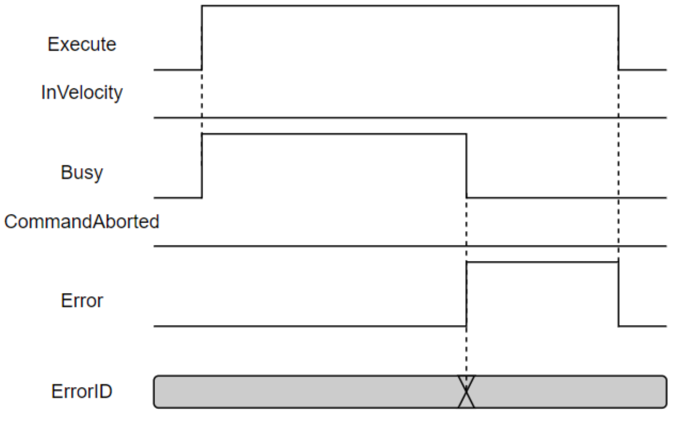
In the StandStill state, the axis calls this command to perform the continuous motion.



During running, the axis is interrupted by the MC_Stop command.

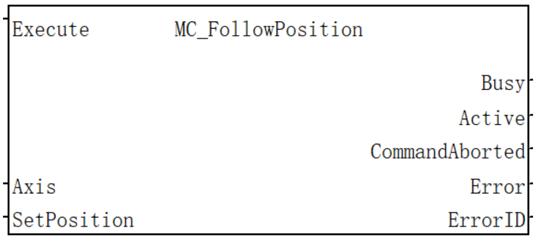


During the axis acceleration, the driver experiences a fault.



3.21.27 MC_FollowPosition

Graphic Block



16-Bit command	-					
32-Bit command	MC_FollowPosition: Synchronized Position Based on CSP mode					
Operand	**Name**	**Description**	**Nullable**	**Default value**	**Range**	**Data Type**
S1	Axis	Axis name/axis ID	No	-	-	WORD
S2	SetPosition	Target position	No	-	Positive/negative/0	REAL
D1	Busy	Ongoing execution flag	Yes	OFF	ON/OFF	BOOL
D2	Active	Execution validity flag	Yes	OFF	ON/OFF	BOOL
D3	CommandAborted	Execution interrupt flag	Yes	OFF	ON/OFF	BOOL
D4	Error	Command error flag	Yes	OFF	ON/OFF	BOOL
D5	ErrorID	Error code	Yes	0	-	WORD

Operand	Const	Y	M	S	D	R	Custom Variable
S1	√	-	-	-	-	-	-
S2	√				√	√	√
D1	-	√	√	√	-	-	√
D2	-	√	√	√	-	-	√
D3	-	√	√	√	-	-	√
D5	-	√	√	√	-	-	√
D6	-	-	-	-	√	√	√

Function Description

This command is applicable to the EtherCAT bus axis and local encoder axis, and is used to achieve periodic sending of user-level target position commands. This command has no acceleration or deceleration planning process for its data sending and therefore directly sends the position increments of the first and second cycles to the servo. The data sending of this command is affected by the scan cycle of the user program, so it is recommended to configure the user program to have a constant scan cycle.

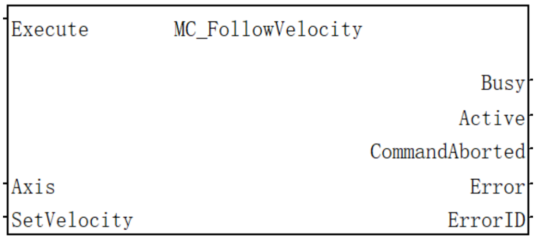
During the execution of this command, the axis is in the DiscreteMotion state.

Interruption

During its execution, this command, it can be interrupted other motion-related commands.

3.21.28 MC_FollowVelocity

Graphic Block



16-Bit command	-					
32-Bit command	MC_FollowVelocity: Synchronized Velocity Based on CSP mode					
Operand	**Name**	**Description**	**Nullable**	**Default value**	**Range**	**Data Type**
S1	Axis	Axis name/axis ID	No	-	-	WORD
S2	SetVelocity	Target velocity	No	-	Positive/negative/0	REAL
D1	Busy	Ongoing execution flag	Yes	OFF	ON/OFF	BOOL
D2	Active	Execution validity flag	Yes	OFF	ON/OFF	BOOL
D3	CommandAborted	Execution interrupt flag	Yes	OFF	ON/OFF	BOOL
D4	Error	Command error flag	Yes	OFF	ON/OFF	BOOL
D5	ErrorID	Error code	Yes	0	-	WORD

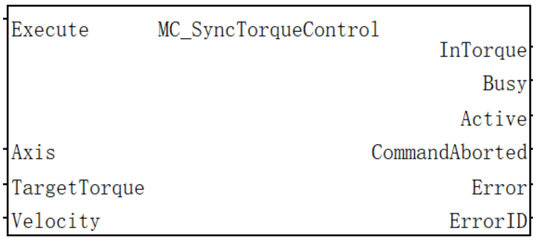
Operand	**Const**	**Y**	**M**	**S**	**D**	**R**	**Custom Variable**
S1	√	-	-	-	-	-	-
S2	√				√	√	√
D1	-	√	√	√	-	-	√
D2	-	√	√	√	-	-	√
D3	-	√	√	√	-	-	√
D5	-	√	√	√	-	-	√
D6	-	-	-	-	√	√	√

Function Description

1. This command is applicable to the EtherCAT bus axis and local encoder axis, and is used to achieve the running of axes at a periodically synchronized velocity in the CSP mode at the user level. This command has no acceleration or deceleration planning process for its data sending and therefore directly sends the data to the servo according to the position increment calculated from the current target velocity. The data sending of this command is affected by the scan cycle of the user program, so it is recommended to configure the user program to have a constant scan cycle.
2. During the execution of this command, the axis is in the ContinuousMotion state.
3. During its execution, this command, it can be interrupted other motion-related commands.

3.21.29 MC_SyncTorqueControl

Graphic Block



16-Bit command	-					
32-Bit command	MC_SyncTorqueControl: Synchronized Torque					
Operand	**Name**	**Description**	**Nullable**	**Default value**	**Range**	**Data Type**
S1	Axis	Axis name/axis ID	No	-	-	WORD
S2	TargetTorque	Target toque, in units of 0.1%. If the target torque is set to 100, it corresponds to 10% of the actual torque	No	0	Positive/ negative/0	INT
S3	Velocity	Limited velocity, whose unit is defined by the user	No	0	Positive/0	REAL
D1	InTorque	Torque reaching signal, which indicates that feedback torque reaches the range of ± 5% of the target torque. If the target torque is 100 and corresponds to 10% of the actual torque and the feedback torque is between 9.5% and 10.5%, the InTorque signal is set to TRUE, otherwise it is set to FALSE	Yes	OFF	ON/OFF	BOOL
D2	Busy	Ongoing execution flag	Yes	OFF	ON/OFF	BOOL
D3	Active	Execution validity flag	Yes	OFF	ON/OFF	BOOL
D4	CommandAborted	Execution interrupt flag	Yes	OFF	ON/OFF	BOOL
D5	Error	Error flag	Yes	OFF	ON/OFF	BOOL
D6	ErrorID	Error code	Yes	0	-	WORD

Operand	Const	Y	M	S	D	R	Custom Variable
S1	√	-	-	-			
S2	√	-	-	-	√	√	√
S3	√	-	-	-	√	√	√
D1	-	√	√	√	-	-	√
D2	-	√	√	√	-	-	√
D3	-	√	√	√	-	-	√
D4	-	√	√	√	-	-	√
D5	-	√	√	√	-	-	√
D6	-	-	-	-	√	√	√

Function Description

1. The MC_SyncTorqueControl command is used to control the bus servo axis (not supporting the virtual axis) to achieve synchronized torque control, is valid to the rising edge, and supports modifying the target torque and maximum velocity limit value through repeated rising edges.
2. TargetTorque represents the target torque in units of 0.1%, assuming a target torque value is 100, it corresponds to 10% of the actual motor torque. Velocity represents the maximum velocity limit, whose unit is defined by the user; it is valid when the axis mapping is 0x607F and invalid when there is no mapping.
3. To use this function block, the EtherCAT control unit type parameter of the servo driver P4.25 should be set to the manufacturer unit, otherwise it will not run.
4. During the startup phase, the torque ramp can be adjusted by setting the torque RAMP time parameter of servo driver P0.68. For example, P0.68 can be set to 100ms rather than too large a value.
5. During the startup phase, if the torque reaching signal generated by the step response of the motor recognizes that the invalid torque is reached, the InTorque output is FALSE.
6. This command supports the mutual interruption between functional blocks according to the PLCopen state machine standards.
7. After calling this command, you can call the MC_Stop, MC_Halt and MC_ImmediateStop commands to stop the axis from running. After the stop is completed, the mode switches to position mode 8, which can be viewed through the current mode parameters of the servo driver R0.32.
8. Running this command switches the servo mode to torque mode 10.

Resetting This Command

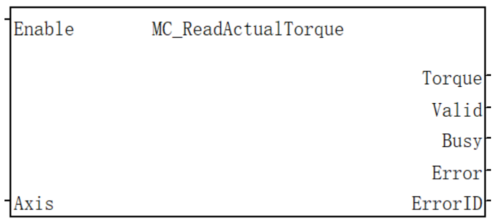
During the validity period of the Busy signal of the MC_SyncTorqueControl command, if the MC_SyncTorqueControl command is triggered again, the running parameters for the second trigger shall prevail.

Multiple Starts of This Command

During the validity period of the Busy signal of the MC_SyncTorqueControl command, if the second MC_SyncTorqueControl command is triggered, the parameters of the second command will be run, while the first command will be interrupted and invalidated.

3.21.30 MC_ReadActualTorque

Graphic Block



16-Bit command	-					
32-Bit command	MC_ReadActualTorque: Read Current Torque					
Operand	Name	Description	Nullable	Default value	Range	Data Type
S1	Axis	Axis name/axis ID	No	-	-	WORD
D1	Torque	Current torque	Yes	0	Positive/negative/0	INT
D2	Valid	Valid flag	Yes	OFF	ON/OFF	BOOL
D3	Busy	Ongoing execution flag	Yes	OFF	ON/OFF	BOOL
D4	Error	Command error flag	Yes	OFF	ON/OFF	BOOL
D5	ErrorID	Error code	Yes	0	-	WORD

Operand	Const	Y	M	S	D	R	Custom Variable
S1	√	-	-	-	-	-	
D1					√	√	√
D2	-	√	√	√	-	-	√
D3	-	√	√	√	-	-	√
D4	-	√	√	√	-	-	√
D5	-	-	-	-	√	√	√

Function Description

1. The MC_ReadActualTorque command is used to read the feedback torque of the bus axis, and is valid at high levels.
2. This command does not support the virtual axis mode nor the pulse axis.
3. This command has no interrupt flag and therefor multiple commands can run simultaneously.

3.21.31 Error Codes of Single Axis Commands

Main error code	Secondary error code	Error level	Possible cause	Solution
0x11(17) Operation control fault	0x1(1)	Fault	The current axis ID is not within the valid range	Check whether the axis ID parameter settings are reasonable
	0x2(2)	Fault	The current function block ID is not within the valid range	Check whether the function block ID parameter settings of the host computer are reasonable
	0x3(3)	Warning	The current function block cannot be started due to the unreasonable PLCopen state	Check whether the current axis state meets the PLCopen state machine switching process when the current command is triggered
	0x4(4)	Warning	Axis configuration failed	Check whether the axis configuration is successful
	0x5(5)	Warning	The address of the PDO parameter DigitalIput is NULL	Check whether the parameter is mapped in the slave station IO mapping. Check whether the parameter exist in the XML version of the servo slave station
	0x6(6)	Fault	Current axis/servo error	The axis/servo is faulty, and the error can be cleared by calling the MC_Reset command or restarting the MC_Power command
	0x7(7)	Warning	The current axis is not enabled and therefore in the Disabled state	Switch the axis to the Standstill state by calling the MC_Power command
	0x8(8)	Fault	The positive hard limit of the axis is triggered	Call the reset instruction to switch the axis state from ErrorStop state to Standstill state
	0x9(9)	Fault	The negative hard limit of the axis is triggered	Call the reset instruction to switch the axis state from ErrorStop state to Standstill state
	0xA(10)	Fault	The positive soft limit of the axis is triggered	Call the reset instruction to switch the axis state from ErrorStop state to

Main error code	Secondary error code	Error level	Possible cause	Solution
				Standstill state
	0xB(11)	Fault	The negative soft limit of the axis is triggered	Call the reset instruction to switch the axis state from ErrorStop state to Standstill state
	0xC(12)	Warning	The pulse axis has not selected any output device	Check whether the pulse axis has selected an output device
	0xD(13)	Warning	The bus axis has not selected any output device	Check whether the bus axis has selected an output device
	0xE(14)	Warning	The current command does not support repeated calls	The current command does not support repeated calls to the function block, so avoid this situation manually
	0xF(15)	Warning	Axis type setting error	Check whether the axis type matches the command type
	0x10(16)	Warning	The PDO address of the bus axis is NULL	Do not use axis control commands to map and send PDO parameters from the I/O mapping of the slave device description file Check whether if the following PDO parameters are configured: ControlWord(16#6040) TargetPosition(16#607A) ModeOfOperation(16#6060) StatusWord(16#6041) PositionActualValue(16#6064) SpeedActualValue(16#606C) DigitalInputs(16#60FD)
	0x11(17)	Warning	Positive hard limit ID configuration failed	Check whether the current pulse axis input and output points are reused
	0x12(18)	Warning	Negative hard limit ID configuration failed	Check whether the current pulse axis input and output points are reused
	0x13(19)	Warning	Probe ID1 configuration failed	Check whether the current pulse axis input and output points are reused
	0x14(20)	Warning	Probe ID2 configuration failed	Check whether the current pulse axis input and output points are reused
	0x15(21)	Warning	Servo error ID configuration failed	Check whether the current pulse axis input and output points are reused
	0x16(22)	Warning	Home signal ID configuration failed	Check whether the current pulse axis input and output points are reused
	0x17(23)	Warning	Z signal ID configuration failed	Check whether the current pulse axis input and output points are reused
	0x18(24)	Warning	Axis enable ID configuration failed	Check whether the current pulse axis input and output points are reused
	0x19(25)	Warning	Failed to clear the servo error ID configuration	Check whether the current pulse axis input and output points are reused
	0x1A(26)	Warning	The axis address is NULL	Check whether the axis configuration is successful
	0x1B(27)	Warning	Bus axis enable failed	If bus axis enable timed out, check whether the EtherCAT communication and feedback state words are normal
	0x1C(28)	Fault	The bus axis has not entered	Check whether the EtherCAT

Main error code	Secondary error code	Error level	Possible cause	Solution
			the OP state	communication has entered the OP state
	0x1D(29)	Warning	The current function block execution is invalid	The current command function is not yet open and is invalid for use
	0x1E(30)	Warning	The current axis communication timed out	Check whether the EtherCAT communication has entered the OP state Check whether the EtherCAT communication return value is normal
	0x1F(31)	Warning	Under the current axis configuration, the EtherCAT synchronization cycle cannot be less than 1 ms	Check whether the setting of the synchronization cycle of the EtherCAT master station is less than 1ms (in case of mixed use of bus axis and pulse axis, the EtherCAT synchronization cycle cannot be less than 1 ms)
	0x20(32)	Warning	The PLC does not run	Check whether the PLC dial switch is set to Stop
	0x21(33)	Warning	The axis triggered a soft-limit deceleration and stop	The current axis is in the process of the soft-limit deceleration and stopping, and the execution of the current triggerd command is invalid
	0x22(34)	Warning	The address of the current command parameter is NULL	If the address of the current command parameter is NULL, provide an input variable or contact the INVT technical personnel
	0x23(35)	Fault	During the pulse axis motion, the pulse frequency of the current interpolation cycle is greater than or equal to 200K	The maximum running frequency of the pulse axis must not exceed 200K, so it is recommended to reduce the running velocity
	0x24(36)	Warning	The pulse axis FPGA cache reached the limit value	This is only a prompt
	0x25(37)	Fault	The PDO data address in EtherCAT is NULL	Check whether the EtherCAT communication is normal
	0x26(38)	Fault	The current servo axis is not on-line	Check whether the EtherCAT communication is normal; Check whether the current servo axis is connected to the network cable
	0x27(39)	Warning	The current axis communication failed	If the EtherCAT communication failed during the operation, check the state of the EtherCAT communication
	0x28(40)	Warning	The value of the PDO parameter StatusWord is 0	Check whether the EtherCAT communication is normal
	0x29(41)	Warning	The address of the PDO parameter ErrorCode is NULL	Check whether the EtherCAT communication is normal Check whether the PDO parameter is configured
	0x2A(42)	Warning	The current axis does not support torque control	Check the axis type configuration, as torque control only supports the bus

Main error code	Secondary error code	Error level	Possible cause	Solution
				axis
	0x65(101)	Warning	The enable command state is abnormal	If the enable command state is abnormal, contact the INVT technical personnel
	0x66(102)	Warning	The reset command state is abnormal	If the reset command state is abnormal, contact the INVT technical personnel
	0x67(103)	Warning	Reset timed out	If the axis reset timed out, check whether the EtherCAT communication is normal
	0x68(104)	Warning	The current axis state does not support the superimposed motion command	If the current axis state does not support the superimposed motion command, refer to the specific commands for using the command
	0x69(105)	Warning	Input parameter error	The command input parameter is not within the valid range
	0x6A(106)	Warning	The system report an error about the repeated calls of the MC_Stop command	Please check whether the same axis is called more than once
	0x6B(107)	Warning	The system report an error about the repeated calls of the MC_ImmediateStop command	Please check whether the same axis is called more than once
	0x6C(108)	Fault	The input parameter of the MC_Stop command is not within the valid range	Please check whether the command parameters are within the valid range, and call the MC_Reset command to reset the axis state
	0x6D(109)	Fault	The input parameter of the MC_Halt command is not within the valid range	Please check whether the command parameters are within the valid range, and call the MC_Reset command to reset the axis state
	0x6E(110)	Warning	The input parameter of the MC_SetOverride command is not within the valid range	Please check whether the command parameters are within the valid range, and call the MC_Reset command to reset the axis state
	0x6F(111)	Fault	The input parameter of the MC_MoveVelocity command is not within the valid range	Please check whether the command parameters are within the valid range, and call the MC_Reset command to reset the axis state
	0x70(112)	Fault	The input parameter of the MC_MoveRelative command is not within the valid range	Please check whether the command parameters are within the valid range, and call the MC_Reset command to reset the axis state
	0x71(113)	Fault	The input parameter of the MC_MC_MoveAbsoulte command is not within the valid range	Please check whether the command parameters are within the valid range, and call the MC_Reset command to reset the axis state
	0x72(114)	Fault	The input parameter of the MC_Jog command is not within the valid range	Please check whether the command parameters are within the valid range, and call the MC_Reset command to

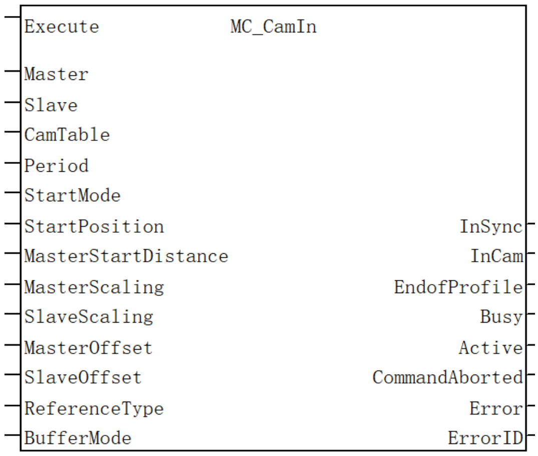
Main error code	Secondary error code	Error level	Possible cause	Solution
				reset the axis state
	0x73(115)	Fault	The input parameter of the MC_Inch command is not within the valid range	Please check whether the command parameters are within the valid range, and call the MC_Reset command to reset the axis state
	0x74(116)	Fault	The input parameter of the MC_Home command is not within the valid range	Please check whether the command parameters are within the valid range, and call the MC_Reset command to reset the axis state
	0x75(117)	Warning	The input parameter of the MC_SetPosition command is not within the valid range	Please check whether the command parameters are within the valid range, and call the MC_Reset command to reset the axis state
	0x76(118)	Warning	It is invalid to trigger the MC_SetOverride command in the current axis state	The current axis is in the process of reversing, and the velocity regulation does not take effect
	0x77(119)	Warning	The current axis is in the operation process of the axis group	Run the axis after the axis group operation is completed
	0x78(120)	Warning	The axis is not in the Standstill state	Before triggering the current command, switch the axis to the StandStill state
	0x79(121)	Warning	Resetting by the MC_Reset command is invalid	The current axis state is not ErrorStop, so resetting is invalid
	0x7A(122)	Warning	The interpolation cycle value settings are invalid	Check the EtherCAT synchronization cycle settings
	0x7B(123)	Warning	It is invalid to trigger the MC_Stop command	Check whether the current axis state can trigger the instruction
	0x7C(124)	Warning	It is invalid to trigger the MC_Halt command	Check whether the current axis state can trigger the instruction
	0x7D(125)	Warning	It is invalid to trigger the MC_ImmediateStop command	Check whether the current axis state can trigger the instruction
	0x7E(126)	Warning	The input parameter of the MC_TouchProbe command is not within the valid range	Please check whether the command parameters are within the valid range, and call the MC_Reset command to reset the axis state
	0x7F(127)	Warning	The input parameter of the MC_MoveSuperImosed command is not within the valid range	Please check whether the command parameters are within the valid range, and call the MC_Reset command to reset the axis state
	0x80(128)	Warning	The MC_Home command has been called repeatedly	Check whether the home function block has been called repeatedly on the same axis
	0x81(129)	Fault	The input parameter of the MC_MoveFeed command is not within the valid range	Please check whether the command parameters are within the valid range, and call the MC_Reset command to reset the axis state
	0x82(130)	Warning	The probe channel used is not configured	Check whether the PDO data in the "Process Data" section of the

Main error code	Secondary error code	Error level	Possible cause	Solution
				configuration interface for the servo axis on the host computer has been added (Possible mappings: 0x60B8, 0x60B9, 0x60BA, 0x60BB, 0x60BC, and 0x60BD)
	0x83(131)	Warning	When the interrupt fixed length function is used with Mode=0 or Mode=1, the probe signal has not arrives after the first distance is traveled.	Check whether the probe signal is triggered normally.
	0x84(132)	Warning	When the probe function is triggered, the probe channel used has already been occupied by the interrupt fixed length function.	Check whether the channel is incompatible.
	0x85(133)	Warning	The axis configuration index parameter is not within the valid range	Check whether the axis configuration index parameter is within the valid range
	0x86(134)	Warning	The axis parameter input by the MC_SetAxisConfigPara command is not within the valid range	Check whether the axis setting parameter is within the valid range
	0x87(135)	Warning	The input parameter of the MC_MoveBuffer command is not within the valid range	Please check whether the command parameters are within the valid range, and call the MC_Reset command to reset the axis state
	0x88(136)	Warning	The input parameter of the MC_SyncMoveVelocity command is not within the valid range	Please check whether the command parameters are within the valid range, and call the MC_Reset command to reset the axis state
	0x89(137)	Warning	The input parameter of the MC_MoveVelocityCSV command is not within the valid range	Please check whether the command parameters are within the valid range, and call the MC_Reset command to reset the axis state
	0x8A(138)	Warning	The input parameter of the MC_SyncTorqueControl command is not within the valid range	Please check whether the command parameters are within the valid range, and call the MC_Reset command to reset the axis state
	0x8B(139)	Warning	PDO data used is not configured	The process data 0x6060 and 0x6061 in the servo configuration of the host computer are not configured
	0x8C(140)	Warning	PDO data used is not configured	The process data 0x606C is not configured in the servo configuration of the host computer
	0x8D(141)	Warning	PDO data used is not configured	The process data 0x60FF is not configured in the servo configuration of the host computer
	0x8E(142)	Warning	PDO data used is not configured	The process data 0x6071 and 0x607F in the servo configuration of the host

Main error code	Secondary error code	Error level	Possible cause	Solution
				computer are not configured
	0x8F(143)	Warning	PDO data used is not configured	The process data 0x6083 and 0x6084 in the servo configuration of the host computer are not configured
	0x90(144)	Warning	The current axis state does not support the single-axis velocity regulation command	Check whether the current axis state meet the requirements of the velocity regulation function

3.21.32 MC_CamIn

Graphic Block



16-Bit command	-					
32-Bit command	MC_CamIn: Electronic Cam Entry					
Operand	**Name**	**Description**	**Nullable**	**Default value**	**Range**	**Data Type**
S1	Master	Master axis name/axis ID It can be chosen from bus servo axis, local pulse axis, and local encoder axis	No	-	0~39	WORD
S2	Slave	Slave axis name/axis ID It can be chosen from bus servo axis and local pulse axis	No	-	0~39	WORD
S3	CamTable	Cam table name	No	-	0~15	WORD
S4	Period	Repeated mode 0: executed periodically 1: executed for only one cycle	Yes	0	0~1	INT
S5	StartMode	Start mode 0: absolute mode 1: relative mode 2: immediate start	Yes	2	0~2	INT
S6	StartPosition	Start position of cam table	Yes	0	Positive/0	REAL
S7	MasterStartDistance	Master axis tracking	Yes	0	Positive/0	REAL

16-Bit command	-					
32-Bit command	MC_CamIn: Electronic Cam Entry					
Operand	**Name**	**Description**	**Nullable**	**Default value**	**Range**	**Data Type**
		distance				
S8	MasterScaling	Master axis scale coefficient	Yes	1	Positive number	REAL
S9	SlaveScaling	Slave axis scale coefficient	Yes	1	Positive number	REAL
S10	MasterOffset	Master axis offset	Yes	0	Positive/ negative/0	REAL
S11	SlaveOffset	Slave axis offset	Yes	0	Positive/ negative/0	REAL
S12	ReferenceType	Master axis position source 0: command position for previous cycle 1: command position for current cycle 2: feedback position for current cycle	Yes	1	0~2	INT
S13	BufferMode	Buffer mode 0: Wait for the completion of the previous cam cycle; others: reserved	Yes	0	0	INT
D1	InSync	Cam synchronization flag	Yes	OFF	ON/OFF	BOOL
D2	InCam	Cam table engagement flag	Yes	OFF	ON/OFF	BOOL
D3	EndOfProfile	Cam cycle completion	Yes	OFF	ON/OFF	BOOL
D4	Busy	Executing	Yes	OFF	ON/OFF	BOOL
D5	Active	Execution validity flag	Yes	OFF	ON/OFF	BOOL
D6	CommandAborted	Execution interruption	Yes	OFF	ON/OFF	BOOL
D7	Error	Error flag	Yes	OFF	ON/OFF	BOOL
D8	ErrorID	Error code	Yes	0	-	WORD

Operand	Const	Y	M	S	D	R	Custom Variable
S1	√	-	-	-	-	-	-
S2	√	-	-	-	-	-	-
S3	√	-	-	-	-	-	√
S4	√	-	-	-	√	√	√
S5	√	-	-	-	√	√	√
S6	√	-	-	-	√	√	√
S7	√	-	-	-	√	√	√
S8	√	-	-	-	√	√	√
S9	√	-	-	-	√	√	√
S10	√	-	-	-	√	√	√
S11	√	-	-	-	√	√	√
S12	√	-	-	-	√	√	√
S13	√	-	-	-	√	√	√

Operand	Const	Y	M	S	D	R	Custom Variable
D1	-	√	√	√	-	-	√
D2	-	√	√	√	-	-	√
D3	-	√	√	√	-	-	√
D4	-	√	√	√	-	-	√
D5	-	√	√	√	-	-	√
D6	-	√	√	√	-	-	√
D7	-	√	√	√	-	-	√
D8	-	-	-	-	√	√	√

Command start condition

This command can be started in any state of master axis stop, position control, velocity control, and synchronization control.

This command can be started when the slave axis is in any of the StandStill, Discrete Motion, Continuous Motion, and Synchronized Motion states.

Function Description

The module is used to implement the electronic cam entry function.

Relative Cam Table

The phase and displacement of the cam table are specified by the relative quantities starting from 0.0. In each EtherCAT cycle, the CamIn function block calculates the slave axis displacement corresponding to the master axis phase based on the selected cam curve type.

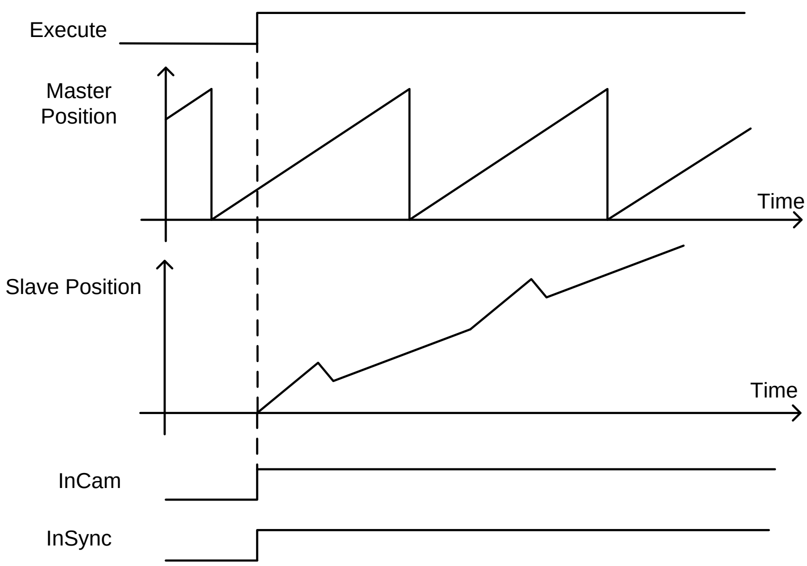
Soft Limit Function

When the position of the slave axis exceeds the software limit in the shaft configuration interface during the cam action, an exception will occur, causing the slave axis to decelerate and stop.

At the Beginning of Cam Action

1. When StartMode is set to 2, which means immediate start.

After the function block Execute is set to TRUE, the command immediately performs the cam action, the current position of the master axis is the phase zero point of the cam, and the current position of the slave axis is the displacement zero point of the cam. At this time, any value set for the function block parameters StartPosition and MasterStartDistance is invalid.



2. When StartMode is set to absolution position or relative position.

After the function block Execute is set to TRUE, the command waits for the master axis to reach StartPosition (the start position of the cam table) to execute the start point of the cam table, and the output variable InCam is TRUE.

The phase and displacement of the cam table are specified by the relative quantities from the zero start point. Therefore, the absolute positions of each axis at each phase are the relative values with the absolute positions of each axis in the cam table as the start points. For example, the camshaft is shown in the following figure, where StartPosition (the start point of the cam table) 50, the absolute position of the master axis is the phase of the cam table plus the value of StartPosition, and the absolute position of the slave axis is the displacement of the cam table plus the absolute position of the slave axis at the start point of the cam table.

Phase	Shift
0	0
80	30
120	50
240	20
360	0

StartPosition = 50

Master axis	Slave axis
50	0 + absolute position of slave axis at start point of cam table
130	30 + absolute position of slave axis at start point of cam table
170	50 + absolute position of slave axis at start point of cam table
290	20 + absolute position of slave axis at start point of cam table
50	0 + absolute position of slave axis at start point of cam table

In addition, when MasterStartDistance (the master axis tracking distance) is passed, the slave axis cam action starts, and InSync outputs TRUE.

The cam table is set as follows:

Phase	Shift
0	0
80	120
120	80
360	140

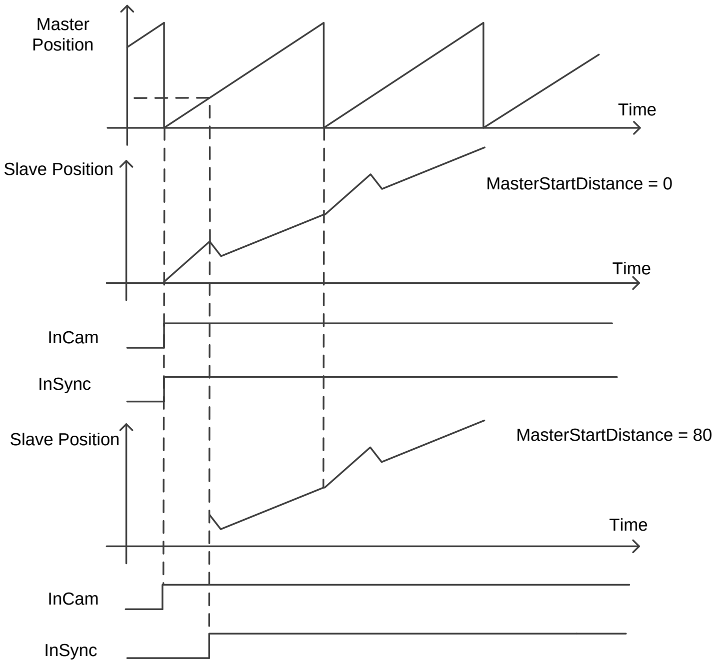
The conditions for the cam action start are listed as follows:

Input variable	Condition 1	Condition 2
StartMode (specified start position method)	Relative position	Relative position
StartPosition (start position of cam table)	0	0
MasterStartDistance (master axis tracking distance)	0	80

Under condition 1, when the master axis passes through 0, the output variables InCam (in cam action) and InSync (in synchronization) are both output as TRUE, and the slave axis starts the cam action.

Under condition 2, when the master axis passes through 0, the output variable InCam (in cam action) is ouput as TRUE; when the master axis passes through 80, the output variable InSync (in synchronization) is output as TRUE, and the slave axis starts the cam action.

Note that under condition 2, the slave axis has a rapid acceleration process when starting the cam action halfway through the cam table.



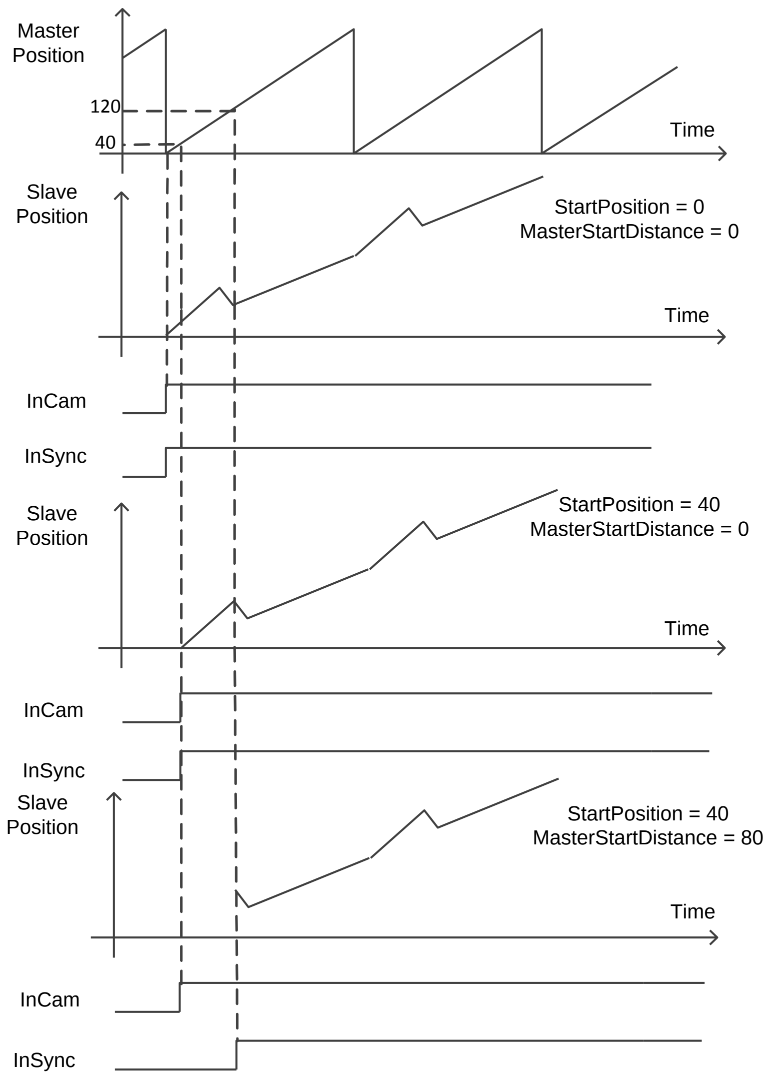
The starting conditions are modified as follows for the above cam table:

Input variable	Condition 1	Condition 2	Condition 3
StartMode	Relative position	Relative position	Relative position
StartPosition	0	40	40
MasterStartDistance	0	0	80

Under condition 1, when the master axis passes through 0, the output variables InCam (in cam action) and InSync (in synchronization) are both output as TRUE, and the slave axis starts the cam action.

Under condition 2, when the master axis passes through 40 specified by StartPosition (start position of cam table), the output variables InCam (in cam action) and InSync (in synchronization) are both output as TRUE, and the slave axis starts the cam action.

Under condition 3, when the master axis passes through 40 specified by StartPosition (start position of cam table), the output variable InCam (in cam action) is ouput as TRUE; when the master axis passes through 120, the output variable InSync (in synchronization) is output as TRUE, and the slave axis starts the cam action.



By using StartMode (specified start position method), you can also decide whether to process the specified values of StartPosition and MasterStartDistance as absolute positions or relative positions.

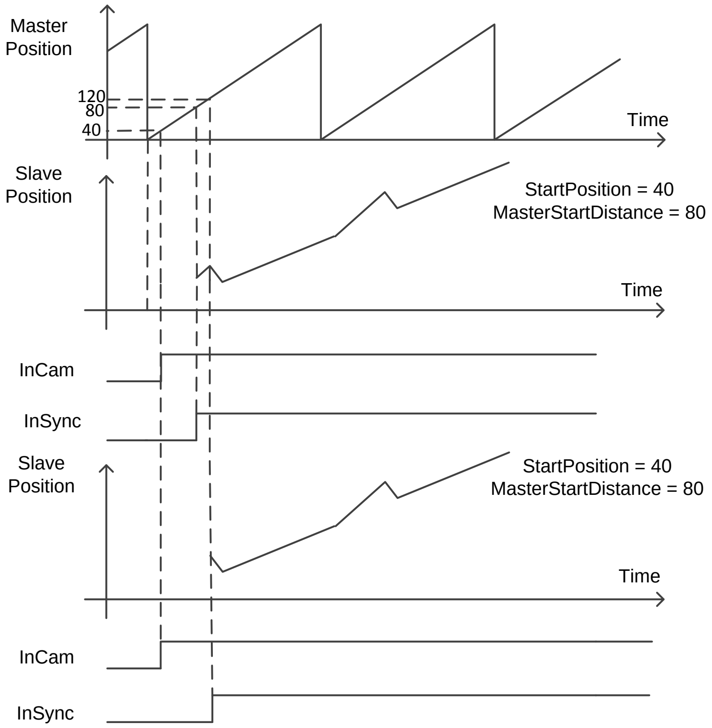
The starting conditions of the cam table are set as follows:

Input variable	Condition 1	Condition 2
StartMode	Absolute position	Relative position
StartPosition	40	40
MasterStartDistance	80	80

Under conditions 1 and 2, when the master axis passes through 40, the output variable InCam is output as TRUE. Under condition 1, since StartMode is specified as an absolute position, when the master axis passes through 80, the output variable InSync is output as TRUE, and the slave axis starts the cam action.

Especially note that under condition 1, if the current axis is a linear axis and the current position is non-zero, StartPosition can be set to an integer multiple of the end point of the cam table to avoid the sharp acceleration or deceleration when the slave starts the cam action. Under condition 1, MasterStartDistance needs to be ahead of the StartPosition position in the master axis direction.

Under condition 2, since StartMode is specified as a relative position, when the master axis passes through 120 (=40+80), the output variable InSync becomes TRUE, and the slave axis starts the cam action.

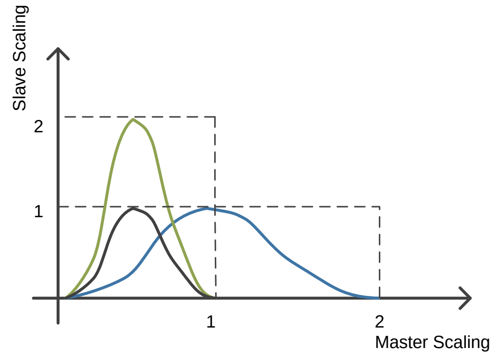


At the End of Cam Action

To end the cam action halfway, you can use the MC_CamOut function block stop command.

Scaling Coefficients of Master and Slave Axes

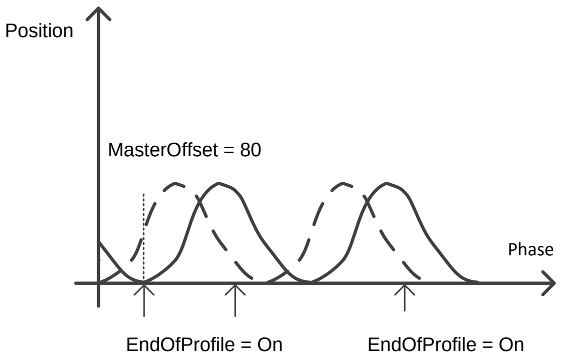
For the specified cam table, the master axis phase and slave axis displacement can be scaled according to the specified ratios.



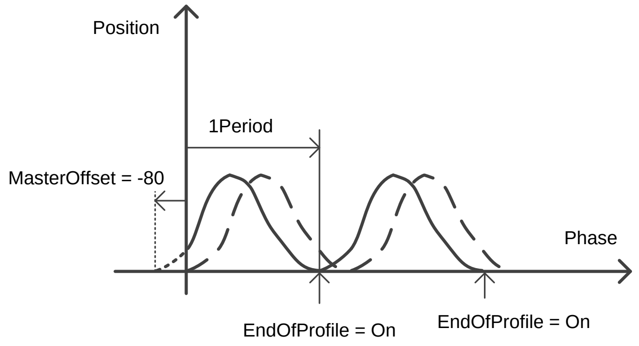
Offsets of Master and Slave Axes

For the specified cam table, the master axis phase and slave shaft displacement can be moved according to the offsets.

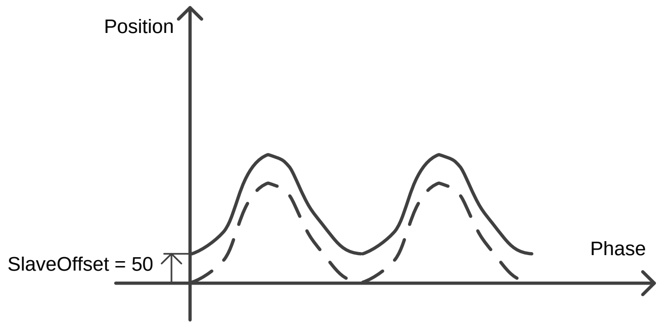
- In case of MasterOffset > 0:



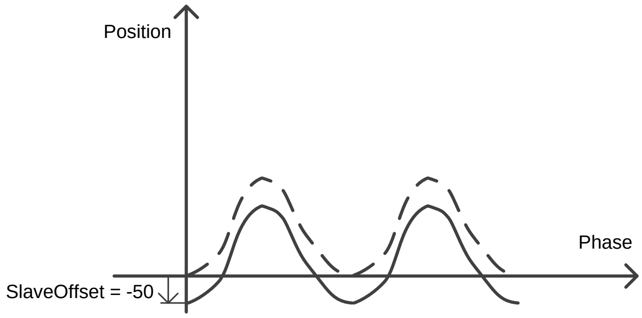
- In case of MasterOffset < 0:



- In case of SlaveOffset > 0:



- In case of SlaveOffset < 0:



Position Type Selection

ReferenceType is used to set the source of position data for the master axis.

1. When the master axis is a local encoder axis, this parameter setting is invalid and will always be the feedback position for this cycle.
2. When the master axis is set to the bus servo axis and local pulse axis, the following three modes can be set: command position for previous cycle, command position for current cycle, and feedback position for this cycle.

Resetting This Command

During the validity period of the Busy signal of the MC_CamIn command, if this command is triggered again and the axis has been in the cam engagement process, parameters StartPosition and MasterStartDistance will be invalid, while parameters Periodic, MasterScaling, SlaveScaling, and ReferenceType will take effect in the next cam cycle.

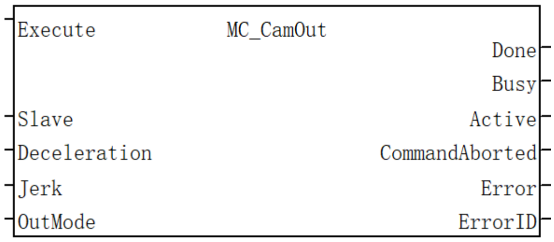
Multiple Starts of This Command

During the validity period of the Busy signal of the MC_CamIn command, if the second MC_CamIn command is triggered and the same slave axis is used, the Busy signal of the second command will be valid, but the

Active signal will be invalid. After a cam cycle ends, the first command is interrupted, and the Active output of the second command is valid. Parameters StartPosition and MasterStartDistance are invalid, but Parameters Periodic, MasterScaling, SlaveScaling, and ReferenceType still take effect as per the parameters of the second command.

3.21.33 MC_CamOut

Graphic Block



16-Bit command	-					
32-Bit command	MC_CamOut: Electronic Cam Exit					
Operand	**Name**	**Description**	**Nullable**	**Default value**	**Range**	**Data Type**
S1	Slave	Axis name/axis ID	No	-	0~39	WORD
S2	Deceleration	Deceleration	Yes	1000	Positive number	REAL
S3	Jerk	Acceleration and deceleration 0: T-type acceleration and deceleration >0: S-type acceleration and deceleration	Yes	0	Positive number	REAL
S4	OutMode	Synchronization mode cancellation selection 0: deceleration-based stop 1: immediate stop after completion of current cam cycle	Yes	0	0~1	INT
D1	Done	Execution completion	Yes	OFF	ON/OFF	BOOL
D2	Busy	Executing	Yes	OFF	ON/OFF	BOOL
D3	Active	Execution validity	Yes	OFF	ON/OFF	BOOL
D4	CommandAborted	Execution interruption	Yes	OFF	ON/OFF	BOOL
D5	Error	Fault	Yes	OFF	ON/OFF	BOOL
D6	ErrorID	Error code	Yes	0	-	WORD

Operand	**Const**	**Y**	**M**	**S**	**D**	**R**	**Custom Variable**
S1	√						
S2	√				√	√	√
S3	√				√	√	√
S4	√				√	√	√
D1		√	√	√			√
D2		√	√	√			√
D3		√	√	√			√
D4		√	√	√			√
D5		√	√	√			√
D6					√	√	√

Function Description

1. The cam action of the axis is cancelled by using this command.
2. When Execute (start) is set to ON, the MC_CamIn command is interrupted, and the interrupt flag bit is valid.
3. If OutMode is set to 0, the slave axis performs the deceleration action is executed based on Deceleration (deceleration); after the slave axis decelerates to 0, the Done output is valid; the slave axis is in the ContinuousMotion state before it stops moving.
4. If OutMode is set to 1, the slave axis stops immediately after completing the cam action for the current cycle; before the cam action ends, the slave axis is in synchronous motion mode.
5. When you enable this command on an axis that has not performed the cam action yet, an exception will occur.

Repeated Triggering

When the MC_CamOut command is triggered repeatedly on the rising edge, the stop mode runs according to the following rules:

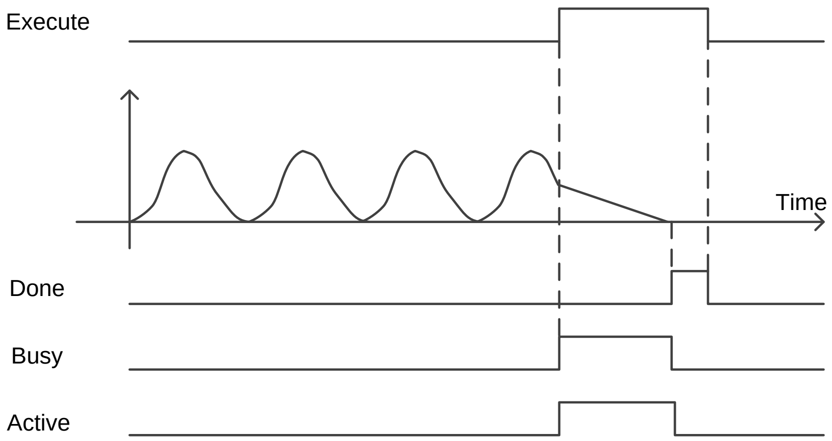
Initial Stop Mode	New Selected Mode	Execution Result
Deceleration-based stop	Immediate stop after completion of current cam cycle	The axis decelerates to completely stop and then enters the Standstall state
Deceleration-based stop	Deceleration-based stop	The axis stops as per the new deceleration
Immediate stop after completion of current cam cycle	Deceleration-based stop	The axis switches to the stop through deceleration mode
Immediate stop after completion of current cam cycle	Immediate stop after completion of current cam cycle	The axis immediately stop after completing the current cam cycle

Multiple Starts of This Command

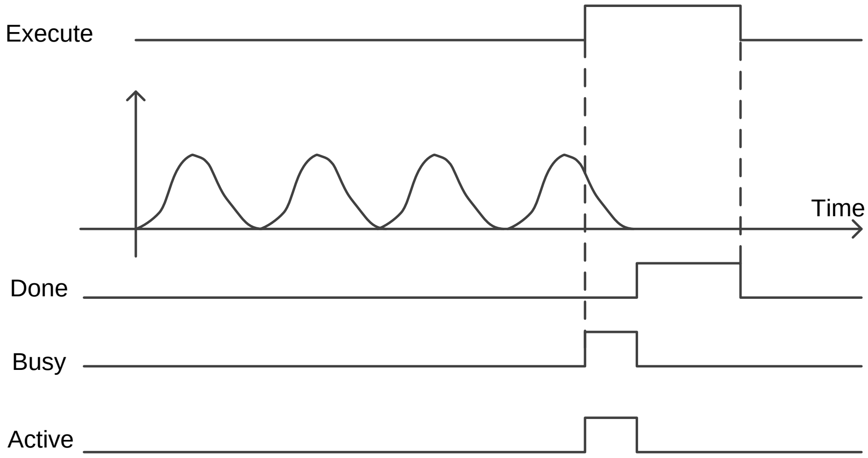
During the validity period of the Busy signal of the MC_CamOut command, if the second MC_CamOut command is triggered and the same slave axis is used, the Busy signal of the second command will be valid, the Active signal will be valid, and the deceleration will take effect according to the parameter setting of the second command, causing the first command to be interrupted and CommandAborted to be valid.

Timing diagram

- Stop through deceleration.

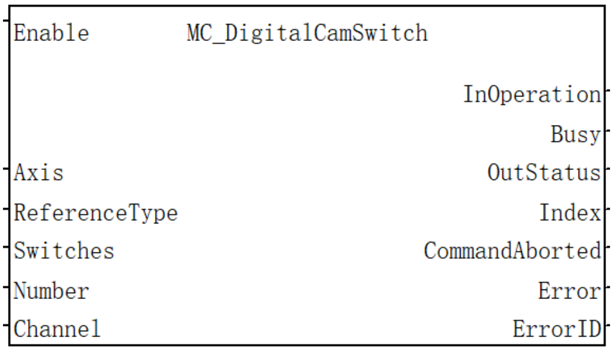


- Immediate stop after completion of current cam cycle.



3.21.34 MC_DigitalCamSwitch

Graphic Block



<table>
<tr><th>16-Bit command</th><td colspan="6">-</td></tr>
<tr><th>32-Bit command</th><td colspan="6">MC_DigitalCamSwitch: Tappet Control</td></tr>
<tr><th>Operand</th><th>Name</th><th>Description</th><th>Nullable</th><th>Default value</th><th>Range</th><th>Data Type</th></tr>
<tr><td>S1</td><td>Axis</td><td>Axis name/axis ID</td><td>No</td><td>-</td><td>0~39</td><td>WORD</td></tr>
<tr><td>S2</td><td>ReferenceType</td><td>Position Type Selection
0: command position for previous cycle
1: command position for current cycle
2: feedback position for current cycle
3: Corresponding master axis phase with axis used as cam slave axis</td><td>No</td><td>-</td><td>0~3</td><td>REAL</td></tr>
<tr><td>S3</td><td>Switches</td><td>Switch</td><td>No</td><td>-</td><td>-</td><td>_stru_MC_DIGITAL_SWITCH[32]</td></tr>
<tr><td>S4</td><td>Number</td><td>Qty</td><td>No</td><td>-</td><td>1-32</td><td>INT</td></tr>
<tr><td>S5</td><td>Channel</td><td>Tappet terminal selection
0~15 represent physical terminal
16~31 represent virtual terminal</td><td>No</td><td>-</td><td>0~31</td><td>INT</td></tr>
</table>

16-Bit command	-					
32-Bit command	MC_DigitalCamSwitch: Tappet Control					
Operand	**Name**	**Description**	**Nullable**	**Default value**	**Range**	**Data Type**
D1	InOperation	Tappet in operation	Yes	OFF	ON/OFF	BOOL
D2	Busy	Command under execution	Yes	OFF	ON/OFF	BOOL
D3	OutStatus	Output status	Yes	OFF	ON/OFF	BOOL
D4	Index	Index; comparison point being executed	Yes	0	0~31	INT
D5	Command Aborted	Execution interruption	Yes	OFF	ON/OFF	BOOL
D6	Error	Fault	Yes	OFF	ON/OFF	BOOL
D7	ErrorID	Error code	Yes	0	-	WORD

Operand	Const	Y	M	S	D	R	Custom Variable
S1	√	-	-	-			
S2	√	-	-	-	√	√	√
S3		-	-	-	√	√	√
S4	√	-	-	-	√	√	√
S5	√	-	-	-	√	√	√
D1	-	√	√	√	-	-	√
D2	-	√	√	√	-	-	√
D3	-	√	√	√	-	-	√
D4	-	-	-	-	√	√	√
D5	-	√	√	√	-	-	√
D6	-	√	√	√	-	-	√
D7	-	-	-	-	√	√	√

Function Description

1. This command is used in combination with the cam to achieve the tappet function.
2. ReferenceType is used for position type selection, supporting the command position for the previous cycle (0), the command position for the current cycle (1), the feedback position for the current cycle (2), and the master axis phase when the axis is specified as a cam slave by the current command (3).
3. Switches is a configuration parameter setting the output point of the tappet, its variable is a structure array with a length of 32, and its structure data type is _stru_MC_DIGITAL_SWITCH.

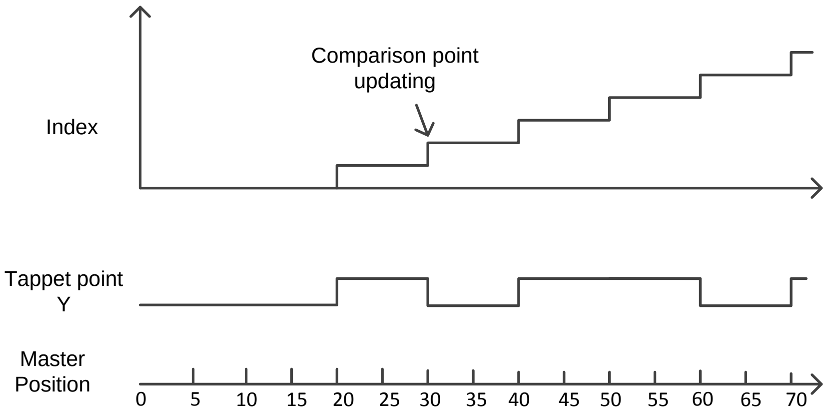
Variable	Data Type	Description
fPosition	REAL	Absolute position during ON validity period
iPosAction	INT	Positive running switch action 0: no action; 1: ON; 2: OFF; 3: inversion
iNegAction	INT	Negative running switch action 0: no action; 1: ON; 2: OFF; 3: inversion

After the tappet command is activated, the command determines the closest tappet point to the current position of the axis internally. After the axis reaches this point, it immediately outputs the tappet action according to Action. Note that fPosition needs to be set to ascending order, otherwise the command reports an error.

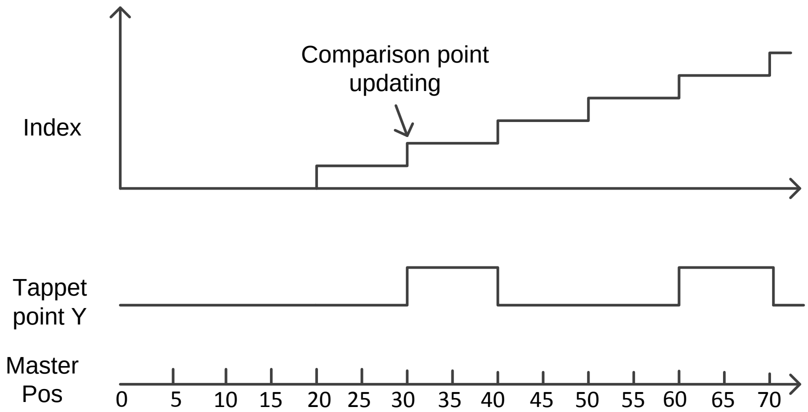
During the positive operation of the command, the tappet sets the tappet output terminal according to iPosAction. During the negative operation of the command, the tappet will set the tappet output terminal according to iNegAction.

No.	fPosition	iPosAction	iNegAction
1	20	1	2
2	30	2	3
3	40	3	1
4	50	0	0
5	60	2	2
6	70	1	3

In case of positive master axis running:



In case of negative master axis running:



OutStatus can be used to monitor the output state of the tappet.

Channel is used to select the tappet terminals, where 0~15 are the digital output terminals of the CPU body and correspond to Y00~Y07 and Y10~Y17, and 16~31 are virtual tappets that only occupy the number of tappets but will not output to the physical hardware terminals. Channel does not support setting a tappet terminal through an expansion module.

On the rising edge of Enable, this command latches its left-side input parameters, and starts executing the tappet output comparison function.

Note that during the Enable=ON period, modifying the left-side input parameters of the command is invalid, and the entire process of the tappet action requires that ON keep valid.

On the falling edge of Enable, the command stops the tappet comparison output function and interrupts the tappet terminal that outputs ON.

Repeated Triggering

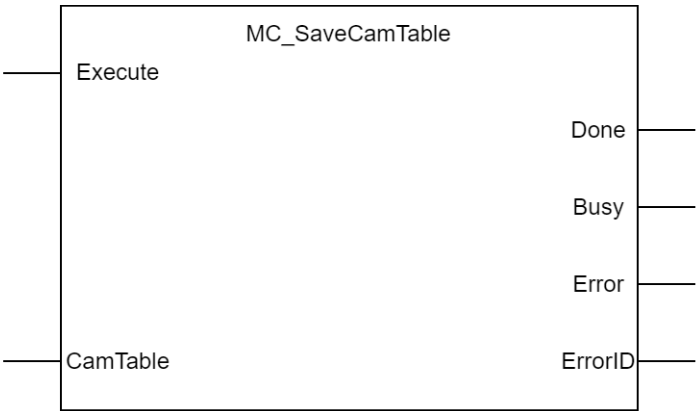
This command is an Enable control command and has no involvement with repeated triggering. It is valid at high levels and invalid at low levels.

Multiple Calls

If two identical MC_DigitalCamSwitch commands have the same Channel setting value, the first command is triggered first. If the second command is triggered during the validity period of Busy signal of the first command, the first command is interrupted, and the tappet point is output through the control of the second tappet command.

3.21.35 MC_SaveCamTable

Graphic Block



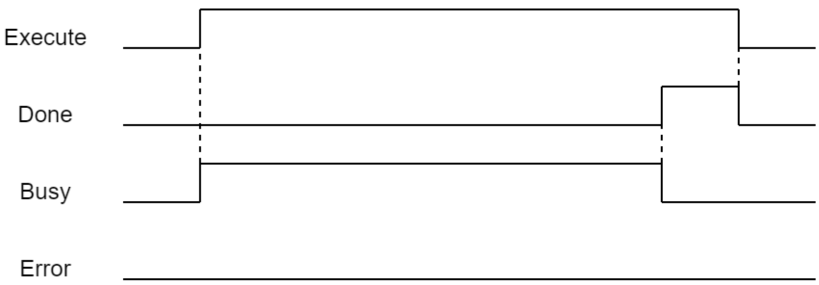
16-Bit command	-					
32-Bit command	MC_SaveCamTable: Save Cam Table					
Operand	Name	Description	Nullable	Default value	Range	Data Type
S1	CamTable	Cam table	No	-	0~15	WORD
D1	Done	Completion	Yes	-	ON/OFF	BOOL
D2	Busy	Executing	Yes		ON/OFF	BOOL
D3	Error	Fault	Yes		ON/OFF	BOOL
D4	ErrorID	Error code	Yes		-	WORD

Function Description

On the rising edge of Execute, this command saves the cam table specified by CamTable to the non-volatile memory.

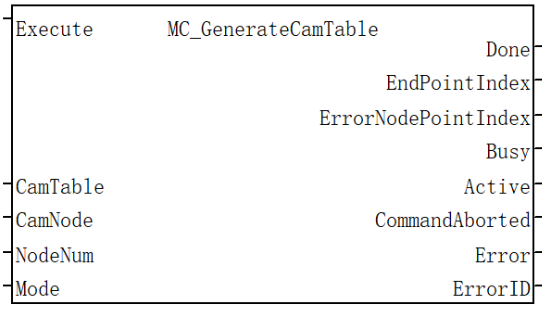
During the execution of this command, the control power cannot be turned off, otherwise data saving failure and data loss will be caused.

Timing diagram



3.21.36 MC_GenerateCamTable

Graphic Block



16-Bit command	-					
32-Bit command	MC_GenerateCamTable: Update Cam Table Command					
Operand	**Name**	**Description**	**Nullable**	**Default value**	**Range**	**Data Type**
S1	CamTable	Cam table name/cam table ID	No	-	0~15	WORD
S2	CamNode	Cam node array When set to empty, it means to use the original cam node array	No	-		_stru_CAM_NODE
S3	NodeNum	The number of cam nodes When set to empty, it means to use the original number of cam nodes	Yes	0	2~361	INT
S4	Mode	Validity mode 0: valid in the next cam cycle Other: reserved	Yes	0	0	INT
S5	Done	Execution completion	Yes	OFF	ON/OFF	BOOL
S6	EndPointIndex	End point index	Yes	0	0~360	REAL
S7	ErrorEndPointIndex	Error node number	Yes	0	0~360	REAL
D2	Busy	Executing	Yes	OFF	ON/OFF	BOOL
D3	Active	Execution validity	Yes	OFF	ON/OFF	BOOL
D4	CommandAborted	Execution interruption	Yes	OFF	ON/OFF	BOOL
D5	Error	Error flag	Yes	OFF	ON/OFF	BOOL
D6	ErrorID	Error code	Yes	0	-	WORD

Function Description

This command is activated through the rising edge of Execute, calculates the cam data based on the values of input codes CamNode and NodeNum, and updates the data to the cam table specified in CamTable, which takes effect in the next cam cycle.

CamNode Variable Functions

The parameter CamNode is used to specify whether to use a new user-defined cam node array. When this parameter is empty, it represents the original node array of the cam table specified by CamTable. When it is not empty, it means to use the cam node array specified by CamNode.

1. When CamNode is empty

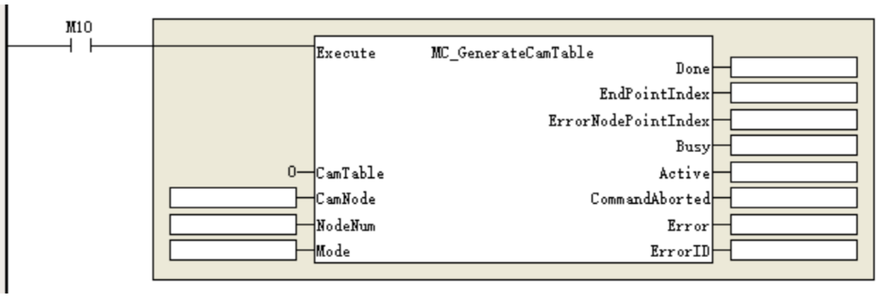
The values in the cam node array in the cam table can be modified through the PLC program, take effect through the MC_GenerateCamTable command, and get executed according to the new cam nodes at the beginning of the next cam cycle.

Phase	Shift
0	0
50	40
90	60
130	30
200	0

Only modify keypoints

Phase	Shift
0	0
50	40
100	70
130	30
200	0

Examples are as follows:



2. When CamNode is not empty

Create a new cam node array in PLC program, and copy the values in the cam node array to the cam table through MC_GenerateCamTable, which are executed in the next cam cycle.

Cam Table A Before Replacement

Phase	Shift
0	0
50	40
90	60
130	30
200	0
0	0

Cam Table A After Replacement

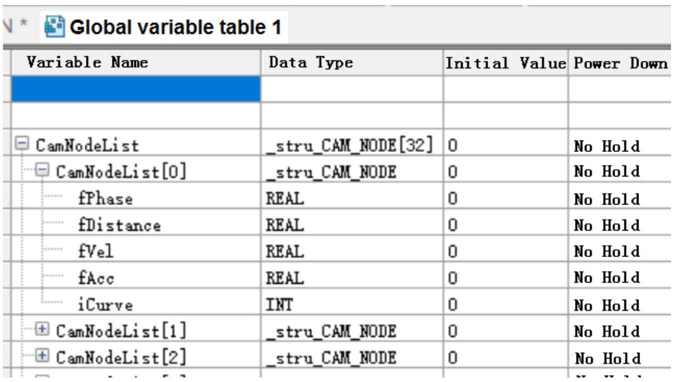
Phase	Shift
0	0
30	40
70	80
100	120
240	200
360	0

Through MC_Generate CamTable copy

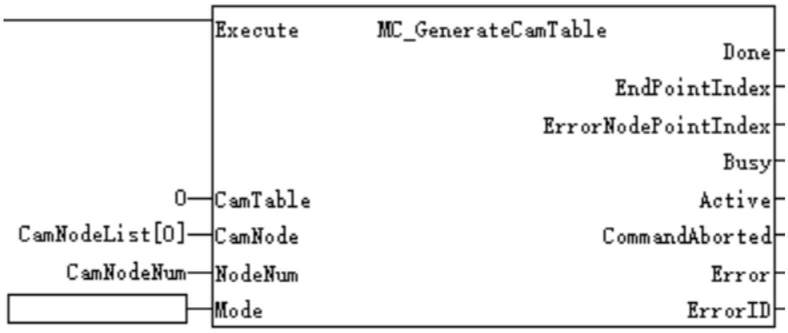
Cam Node Array Created by PLC Program

Phase	Shift
0	0
30	40
70	80
100	120
240	200
360	0

Program Example


Global variable table 1

Variable Name	Data Type	Initial Value	Power Down
CamNodeList	_stru_CAM_NODE[32]	0	No Hold
CamNodeList[0]	_stru_CAM_NODE	0	No Hold
fPhase	REAL	0	No Hold
fDistance	REAL	0	No Hold
fVel	REAL	0	No Hold
fAcc	REAL	0	No Hold
iCurve	INT	0	No Hold
CamNodeList[1]	_stru_CAM_NODE	0	No Hold
CamNodeList[2]	_stru_CAM_NODE	0	No Hold



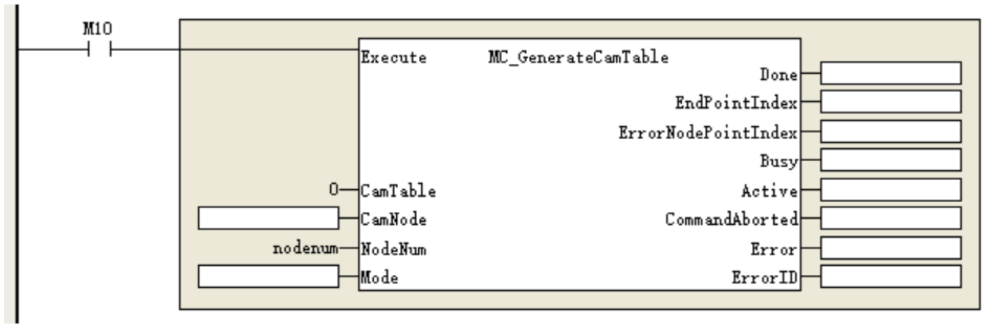
The above CamNodeNum is used to specify the number of nodes in the cam node array in the cam table created by the PLC program.

NodeNum Variable Functions

The parameter NodeNum is used to represent the number of nodes in the new generated cam table. When this parameter is empty, it indicates that the number of nodes in the cam table remains unchanged; if it is not empty, the value specified by NodeNum is used.

You can use this parameter to modify the number of cam table keypoints, make the number valid through the MC_GenerateCamTable command, and execute the program according to the new cam node array from the next cam cycle.

Program Example



Parameter Rationality Check

Before calling this command, you should first check the rationality of the cam keypoint data:

1. The phase and displacement of the first point must be 0, otherwise the system reports an error.
2. The number of nodes cannot exceed 361, otherwise the system reports an error.
3. The number of nodes must be 2 at least, otherwise the system reports an error.
4. Phases must be arranged in ascending order, otherwise the system reports an error.
5. The phase difference between two adjacent master axes must be greater than 0.0001, otherwise the system reports an error.
6. The curve type of the node can only be set to quintic curve (1) or straight line (0), otherwise the system reports an error.

Rules of Velocity Ratio Adjustment

When you call this command, if the velocity ratio of the cam keypoint is not set reasonably, the ratio will be adjusted automatically. See below for the modification rules:

1. When the current segment is a straight line, the velocity ratio will be automatically calculated and adjusted according to the formula.

For example, if A1 and A2 form a straight line, the calculated velocity ratio will be written into the A2 keypoint. If the coordinates of the A1 point are (x_1,y_1) and the coordinates of the A2 point are (x_2,y_2), the velocity ratio of the line A1~A2 is:

$$\frac{y_2 - y_1}{x_2 - x_1}$$

2. When the quintic curve immediately follows a straight line, adjustment should be done to ensure the continuity of the velocity ratio between the quintic curve and the straight line at the junction point and to prevent step jumps.

For example, points A1 and A2 form a quintic curve, and points A2 and A3 form a straight line. First, the velocity ratio of the straight line A2~A3 is calculated, and the calculation result is written into A3; then, the velocity ratio of the quintic curve is adjusted, and the adjustment result is written into A2 to keep the speed ratios of A2 and A3 consistent.

3. If a quintic curve immediately follows another quintic curve, no adjustment needs to be made.
4. If a straight line immediately follows another straight line, it is necessary to calculate the joint velocity of each segment separately, and a sudden change in the joint velocity ratio is allowed at this time.

For example, A1~A2 form the first straight line segment, and A2~A3 form the second straight line segment. First, the velocity ratio of the first straight line segment is calculated and written into A2. Then, the joint velocity of the second straight line segment is calculated and written into A3. At this time, it is allowed that a sudden velocity change caused by the joint velocity inconsistency between the first and second straight line segments.

Multiple Starts of This Command

For this command, after the Done signal is set to ON, you can trigger the current command by repeating the rising edge or trigger a new MC_GenerateCamTable command to take effect, otherwise the system reports an error.

3.21.37 MC_GetCamTableDistance

Graphic Block

```
Execute       MC_GetCamTableDistance
CamTable                                   Done
StartPoint                             Distance
EndPoint                                  Error
Phase                                   ErrorID
```

16-Bit command	-					
32-Bit command	MC_GetCamTableDistance: Get Cam Table Displacement					
Operand	Name	Description	Nullable	Default value	Range	Data Type
S1	CamTable	Cam table name/cam table ID	No	-	0~3	REAL
S2	StartPoint	Start point	No	-	-	_stru_CAM_NODE
S3	EndPoint	End point	No	-	-	_stru_CAM_NODE
S4	Phase	Master axis phase	No	-	Positive/ negative/0	REAL
D1	Done	Execution completion flag	Yes	OFF	ON/OFF	BOOL
D2	Distance	Solved displacement	Yes	0	Positive/ negative/0	REAL
D3	Error	Fault	Yes	OFF	ON/OFF	BOOL
D4	ErrorID	Error code	Yes	0	-	WORD

Operand	Const	Y	M	S	D	R	Custom Variable
S1	√	-	-	-			
S2	-	-	-	-	√	√	√
S3	-	-	-	-	√	√	√
S4	√	-	-	-	√	√	√
D1	-	√	√	√	-	-	√
D2	-	-	-	-	√	√	√
D3	-	√	√	√	-	-	√
D4	-	-	-	-	√	√	√

Function Description

This command can calculate the corresponding slave axis displacement (Distance) according to the master axis phase (Phase) between two cam keypoints.

Description of _stru_CAM_NODE Structure Members:

fPhase: Master axis phase

fDistance: Slave axis position corresponding to master axis phase

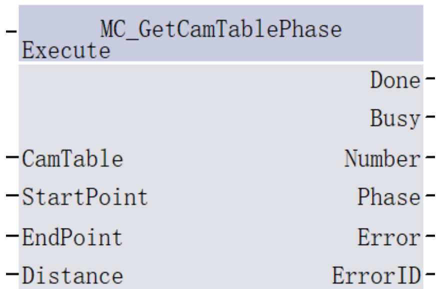
fVel: master-slave axis velocity ratio

fAcc: master-slave axis acceleration ratio

iCurve: Curve type, 0: straight line; 1: quintic curve

3.21.38 MC_GetCamTablePhase

Graphic Block



16-Bit command	-					
32-Bit command	MC_GetCamTablePhase: Get Cam Table					
Operand	Name	Description	Nullable	Default value	Range	Data Type
S1	CamTable	Cam table name/cam table ID	No	-	0~15	REAL
S2	StartPoint	Start point	No	-	-	_stru_CAM_NODE
S3	EndPoint	End point	No	-	-	_stru_CAM_NODE
S4	Distance	Slave axis displacement	No	-	Positive/ negative/0	REAL
D1	Done	Execution completion flag	Yes	OFF	ON/OFF	BOOL
D2	Busy	Executing	Yes	OFF	ON/OFF	BOOL
D3	Number	Number of corresponding phases -1: Mariad solution	Yes	0	Positive/ negative/0	INT

16-Bit command	-					
32-Bit command	MC_GetCamTablePhase: Get Cam Table					
Operand	**Name**	**Description**	**Nullable**	**Default value**	**Range**	**Data Type**
		0: None >0: Actual number				
D4	Phase	Solved phase valure	Yes	0	Positive/ 0	REAL[6]
D5	Error	Error flag	Yes	OFF	ON/OFF	BOOL
D6	ErrorID	Error code	Yes	0	-	WORD

Operand	Const	Y	M	S	D	R	Custom Variable
S1	√	-	-	-			
S2	-	-	-	-	-	-	√
S3	-	-	-	-	-	-	√
S4	√	-	-	-	√	√	√
D1	-	√	√	√	-	-	√
D2	-	√	√	√	-	-	√
D3	-	-	-	-	√	√	√
D4	-	-	-	-	-	-	√
D5	-	√	√	√	-	-	√
D6	-	-	-	-	√	√	√

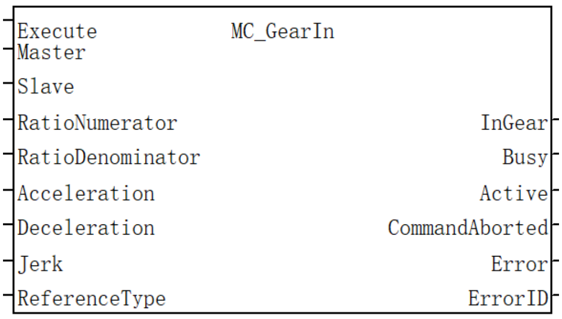
Function Description

This command can calculate the corresponding master axis phase (Phase) according to the slave axis displacement (Distance) between two cam keypoints.

- If the cam curve is a straight line and parallel to the X-axis, the Distance given in the specifications is on the straight line, and the command output parameter Number outputs -1, and Phase [0] outputs the abscissa of the starting point.
- If the cam curve is a quintic curve, there may be multiple solutions. Thecommand output parameter Number represents the solved number of Phase, and the specific solution values are stored in the Phase array.
- If there is no solution, the output parameter Number is equal to 0, and the function block reports an error message stating Distance is not within the range of the cam curve.

3.21.39 MC_GearIn

Graphic Block



16-Bit command	-					
32-Bit command	MC_GearIn: Electronic Gear Entry					
Operand	Name	Description	Nullable	Default value	Range	Data Type
S1	Master	Master axis name/axis ID It can be chosen from bus servo axis, local pulse axis, and local encoder axis	No	-	0~39	WORD
S2	Slave	Slave axis name/axis ID It can be chosen from bus servo axis and local pulse axis	No	-	0~39	WORD
S3	RatioNumerator	Numerator of the gear ratio	Yes	1	Positive/ negative/0	DINT
S4	RatioDenominator	Denominator of the gear ratio	Yes	1	Positive number	DWORD
S5	Acceleration	Acceleration	Yes	1000	Positive number	REAL
S6	Deceleration	Deceleration	Yes	1000	Positive number	REAL
S7	Jerk	Jerk value	Yes	10000	Positive number	REAL
S8	ReferenceType	Master axis position source 0: command position for previous cycle 1: command position for current cycle 2: feedback position for current cycle	Yes	1	0~2	INT
D1	InGear	Reach specified gear ratio	Yes	OFF	ON/OFF	BOOL
D2	Busy	Executing	Yes	OFF	ON/OFF	BOOL
D3	Active	Execution validity	Yes	OFF	ON/OFF	BOOL
D4	CommandAborted	Execution interruption	Yes	OFF	ON/OFF	BOOL
D5	Error	Error flag	Yes	OFF	ON/OFF	BOOL
D6	ErrorID	Error code	Yes	0	-	WORD

Operand	Const	Y	M	S	D	R	Custom Variable
S1	√	-	-	-	-	-	-
S2	√	-	-	-	-	-	-
S3	√	-	-	-	√	√	√
S4	√	-	-	-	√	√	√
S5	√	-	-	-	√	√	√
S6	√	-	-	-	√	√	√
S7	√	-	-	-	√	√	√
S8	√	-	-	-	√	√	√
D1	-	√	√	√	-	-	√
D2	-	√	√	√	-	-	√
D3	-	√	√	√	-	-	√
D4	-	√	√	√	-	-	√

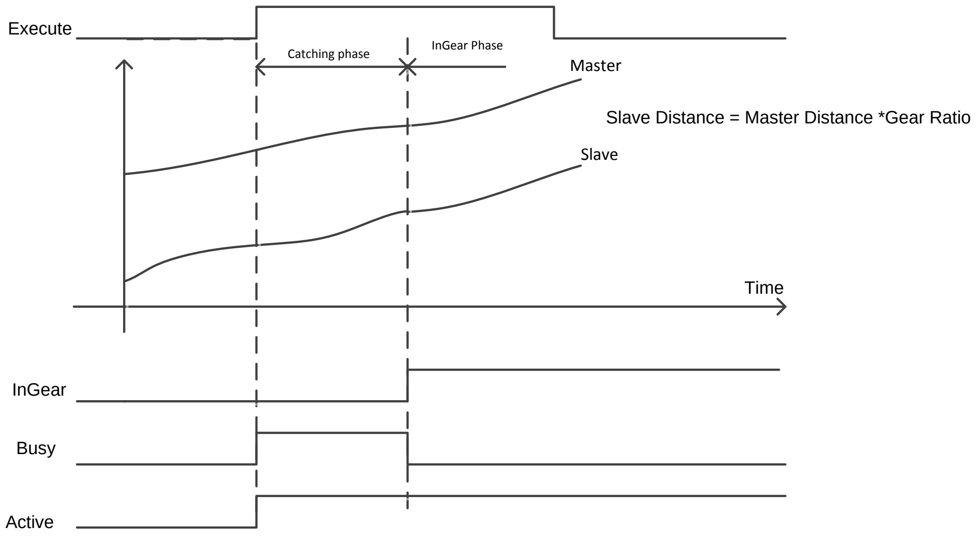
Operand	Const	Y	M	S	D	R	Custom Variable
D5	-	√	√	√	-	-	√
D6	-	-	-	-	√	√	√

Function Description

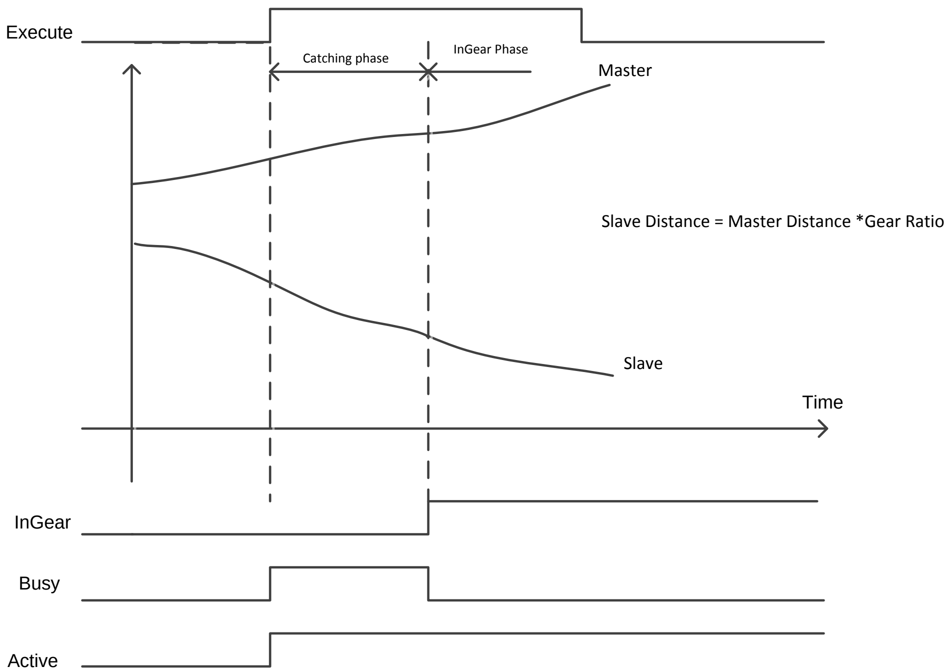
After the action is started, the speed obtained by multiplying the master axis speed by the gear ratio is taken as the target speed, and the acceleration and deceleration operation is carried out on the slave axis.

Before reaching the target position, the phase is called Catching Phase; after reaching the target position, the phase is called InGear Phase.

If the gear ratio is positive, the slave axis moves in the same direction as the master axis.



If the gear ratio is negative, the slave axis moves in the direction opposite to the master axis.



Resetting This Command

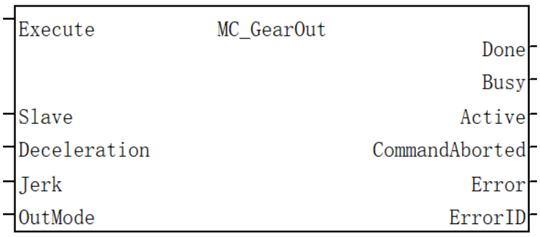
If triggered again during the validity period of the Busy signal of the MC_GearIn command, this command will re-calculate the target velocity of the slave axis based on the numerator and denominator of the gear ratio, and the slave axis will perform a follow-up action based on the calculation result and determine whether InGear is set based on the calculation result.

Multiple Starts of This Command

During the validity period of the Busy signal of the MC_GearIn command, if the second command is triggered, the Busy signal of the second command will be valid, the first command will be interrupted, the second command will re-calculate the target velocity of the slave axis based on the numerator and denominator of the gear ratio, and the slave axis will perform a follow-up action based on the calculation result and determine whether InGear is set based on the calculation result.

3.21.40 MC_GearOut

Graphic Block



16-Bit command	-					
32-Bit command	MC_GearOut: Electronic Gear Exit					
Operand	**Name**	**Description**	**Nullable**	**Default value**	**Range**	**Data Type**
S1	Slave	Axis name/axis ID	No	-	0~39	WORD
S2	Deceleration	Deceleration	Yes	1000	Positive number	REAL
S3	Jerk	Acceleration and deceleration 0: T-type acceleration and deceleration >0: S-type acceleration and deceleration	Yes	0	Positive number	REAL
S4	OutMode	Synchronization mode cancellation selection 0: deceleration-based stop 1: immediate stop	Yes	0	0~1	INT
D1	Done	Execution completion	Yes	OFF	ON/OFF	BOOL
D2	Busy	Executing	Yes	OFF	ON/OFF	BOOL
D3	Active	Execution validity	Yes	OFF	ON/OFF	BOOL
D4	CommandAborted	Execution interruption	Yes	OFF	ON/OFF	BOOL
D5	Error	Fault	Yes	OFF	ON/OFF	BOOL
D6	ErrorID	Error code	Yes	0	-	WORD

Operand	Const	Y	M	S	D	R	Custom Variable
S1	√	-	-	-	-	-	-
S2	√	-	-	-	√	√	√
S3	√	-	-	-	√	√	√
S4	√	-	-	-	√	√	√
D1	-	√	√	√	-	-	√
D2	-	√	√	√	-	-	√
D3	-	√	√	√	-	-	√
D4	-	√	√	√	-	-	√
D5	-	√	√	√	-	-	√
D6	-	-	-	-	√	√	√

Function Description

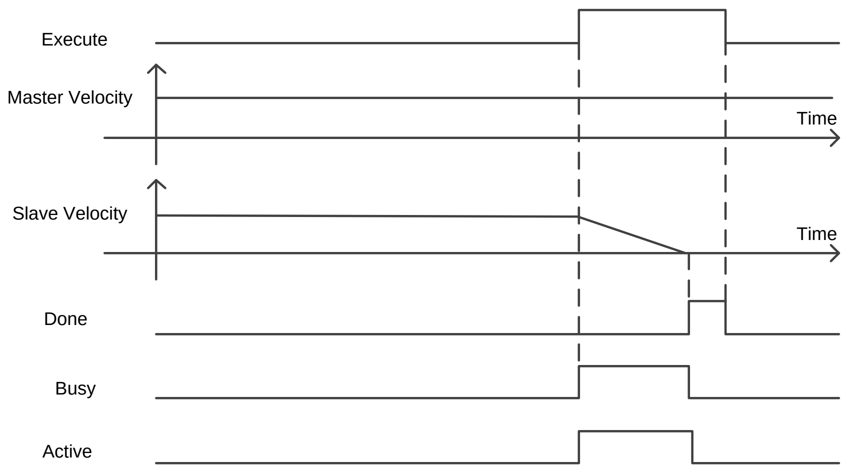
This function block is used to implement the electronic gear exit function. When the gear exits, the slave axis slows down and stops as per the specified deceleration. If Execute is set to TRUE, the slave axis slows down to "0" according to Deceleration (deceleration); if the slave axis command velocity changes to "0", Done becomes TRUE.

Multiple Starts of This Command

During the validity period of the Busy signal of the MC_GearOut command, if the second command is triggered, the Busy signal of the second command is valid, the first command is interrupted, and the axis decelerates to 0 according to the deceleration of the second command.

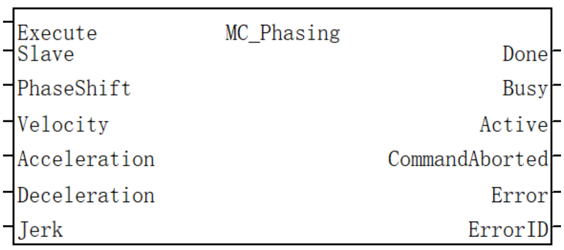
Timing diagram

Timing diagram for stop with deceleration



3.21.41 MC_Phasing

Graphic Block



16-Bit command	-					
32-Bit command	MC_Phasing: Phase Offset					
Operand	**Name**	**Description**	**Nullable**	**Default value**	**Range**	**Data Type**
S1	Slave	Slave axis name/axis ID	No	-	0~39	WORD
S2	PhaseShift	Phase offset	No	-	Positive/ negative/0	REAL
S3	Velocity	Speed	Yes	100	Positive number	REAL
S4	Acceleration	Acceleration	Yes	1000	Positive number	REAL
S5	Deceleration	Deceleration	Yes	1000	Positive number	REAL
S6	Jerk	Jerk value 0: T-type acceleration and deceleration >0: S-type acceleration and deceleration	Yes	0	Positive number	REAL
D1	Done	Phase offset completion	Yes	OFF	ON/OFF	BOOL
D2	Busy	Executing	Yes	OFF	ON/OFF	BOOL
D3	Active	Execution validity	Yes	OFF	ON/OFF	BOOL
D4	CommandAborted	Execution interruption	Yes	OFF	ON/OFF	BOOL
D5	Error	Error flag	Yes	OFF	ON/OFF	BOOL
D6	ErrorID	Error code	Yes	0	-	WORD

Operand	Const	Y	M	S	D	R	Custom Variable
S1	√	-	-	-	-	-	-
S2	√	-	-	-	√	√	√
S3	√	-	-	-	√	√	√
S4	√	-	-	-	√	√	√
S5	√	-	-	-	√	√	√
S6	√	-	-	-	√	√	√
D1	-	√	√	√	-	-	√
D2	-	√	√	√	-	-	√
D3	-	√	√	√	-	-	√
D4	-	√	√	√	-	-	√
D5	-	√	√	√	-	-	√
D6	-	√	√	√	-	-	√

Function Description

This command is only applicable to the MC_CamIn (cam action start) and MC_GearIn (gear action start) commands.

If activated in the master-slave axis synchronization control, this command compensates the master axis phase according to on the set PhaseShift (phase compensation), Velocity (velocity), Acceleration (acceleration), and Deceleration (deceleration).

1. For cooperating with the cam motion, this command is only called after calling the CamIn command. In case of InSync==OFF for the CamIn command, the phase compensation command is in a buffered state, only the Busy signal is valid, and the Active signal output is invalid. In case of InSync==ON for the

MC_CamIn command, the cam is fully engaged, the Active signal output of the phase compensation command is valid, and the phase compensation action starts.

2. For cooperating with the gear motion, this command is only called after calling the GearIn command. First, the MC_GearIn command is triggered to establish a gear relationship between master and slave axes; after the slave axis is in the SyncMotion state, the MC_Phasing command is triggered, and the execution of the phase compensation action starts.

During the execution, the set position (feedback position) of the master axis remains unchanged, and the value compensated for the set position (feedback position) with simply a relative amount by the MC_Phasing is the "master axis phase". The slave axis is synchronized with the compensated "master axis phase".

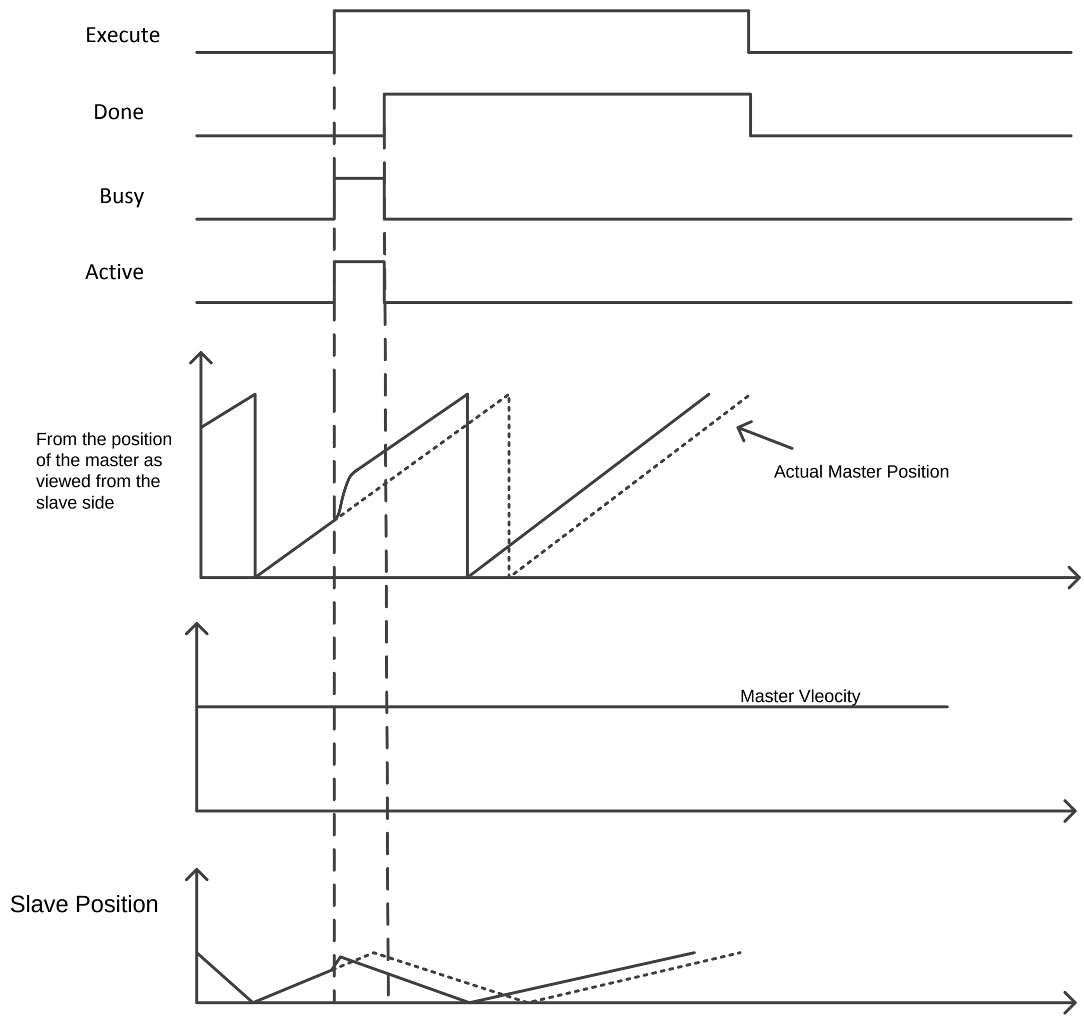
When PhaseShift (the phase compensation) is reached, Done (the execution completion flag) becomes ON.

When the synchronization control command in execution is completed, the compensation ends. When you execute the synchronization control command again, the previous compensation is not affected.

Multiple Starts of This Command

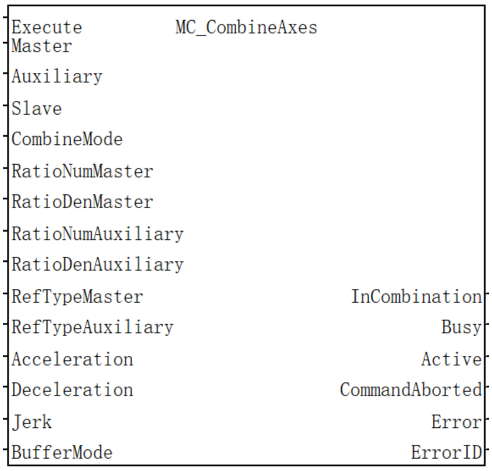
During the validity period of the Busy signal of the MC_Phasing command, if the second command is triggered, the Busy signal of the second command is valid, the first command is interrupted, and the slave axis performs the acceleration or deceleration action according to the parameters of the second command to keep the specified phase difference with the master axis.

Timing diagram



3.21.42 MC_CombineAxes

Graphic Block



16-Bit command	-					
32-Bit command	MC_GearIn: Electronic Gear Entry					
Operand	**Name**	**Description**	**Nullable**	**Default value**	**Range**	**Data Type**
S1	Master	Master axis name/axis ID It can be chosen from bus servo axis, local pulse axis, and local encoder axis	No	-	0~39	WORD
S2	Auxiliary	Auxiliary axis name/axis ID It can be chosen from bus servo axis, local pulse axis, and local encoder axis	No	-	0~39	WORD
S3	Slave	Slave axis name/axis ID It can be chosen from bus servo axis and local pulse axis	No	-	0~39	DINT
S4	CombineMode	Addition-subtraction selection 0: addition 1: subtraction	No	-	0~1	INT
S5	RatioNumMaster	Gear ratio numerator of master axis	Yes	1	Positive number	DINT
S6	RatioDenMaster	Gear ratio denominator of master axis	Yes	1	Positive number	DWORD
S7	RatioNumAuxiliary	Gear ratio numerator of auxiliary axis	Yes	1	Positive number	DINT
S8	RatioDenAuxiliary	Gear ratio denominator of auxiliary axis	Yes	1	0~2	DWORD
S9	RefTypeMaster	Selection of master axis position type 0: command position for previous cycle 1: command position for current cycle 2: feedback position for current cycle	Yes	1		INT

16-Bit command	-					
32-Bit command	MC_GearIn: Electronic Gear Entry					
Operand	**Name**	**Description**	**Nullable**	**Default value**	**Range**	**Data Type**
S10	RefTypeAuxiliary	Selection of auxiliary axis position type 0: command position for previous cycle 1: command position for current cycle 2: feedback position for current cycle	Yes	1		INT
S11	Acceleration	Acceleration (reserved)	Yes	1000		REAL
S12	Deceleration	Deceleration (reserved)	Yes	1000		REAL
S13	Jerk	Jerk (reserved)	Yes	0		REAL
S14	BufferMode	Buffer mode (reserved)	Yes	0		INT
D1	InCombination	Ongoing acceleration-deceleration operation	Yes	OFF	ON/OFF	BOOL
D2	Busy	Executing	Yes	OFF	ON/OFF	BOOL
D3	Active	Execution validity	Yes	OFF	ON/OFF	BOOL
D4	CommandAborted	Execution interruption	Yes	OFF	ON/OFF	BOOL
D5	Error	Error flag	Yes	OFF	ON/OFF	BOOL
D6	ErrorID	Error code	Yes	0	-	WORD

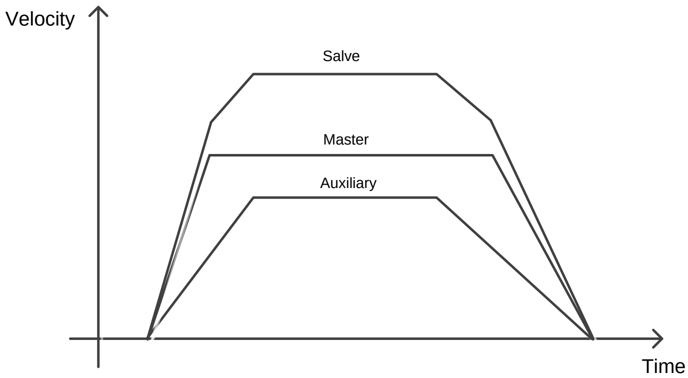
Operand	Const	Y	M	S	D	R	Custom Variable
S1	√	-	-	-	-	-	-
S2	√	-	-	-	-	-	-
S3	√	-	-	-	-	-	-
S4	√	-	-	-	√	√	√
S5	√	-	-	-	√	√	√
S6	√	-	-	-	√	√	√
S7	√	-	-	-	√	√	√
S8	√	-	-	-	√	√	√
S9	√	-	-	-	√	√	√
S10	√	-	-	-	√	√	√
S11	√	-	-	-	√	√	√
S12	√	-	-	-	√	√	√
S13	√	-	-	-	√	√	√
S14	√	-	-	-	√	√	√
D1	-	√	√	√	-	-	√
D2	-	√	√	√	-	-	√
D3	-	√	√	√	-	-	√
D4	-	√	√	√	-	-	√
D5	-	√	√	√	-	-	√
D6	-	-	-	-	√	√	√

Function Description

1. This command outputs the value obtained by addition or subtraction of the positions of the main and auxiliary axes as the position of the slave axis.
2. There are two ways for this command to execute combination: addition or subtraction, which can be set through the input parameter CombineMode.

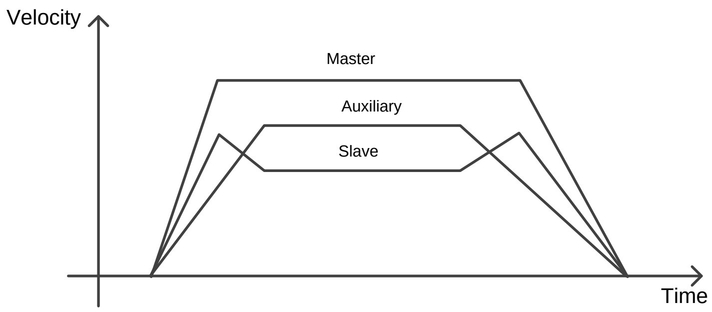
➢ When CombineMode is set to 0:

Increment of slave axis position = increment of master axis position * RatioNumMaster/RatioDenMaster + increment of auxiliary axis position * RatioNumAuxiliary/RatioDenAuxiliary



➢ When CombineMode is set to 1:

Increment of slave axis position = increment of master axis position * RatioNumMaster/RatioDenMaster - increment of auxiliary axis position * RatioNumAuxiliary/RatioDenAuxiliary



The numerator and denominator of the gear ratio between the master and auxiliary axes are set as the position increment adjustment factors for these two axes.

To end the master-slave axis relationship of this command, this can be done by exiting the engagement between the master and slave axes through commands such as MC_GearOut and MC_Halt.

Resetting This Command

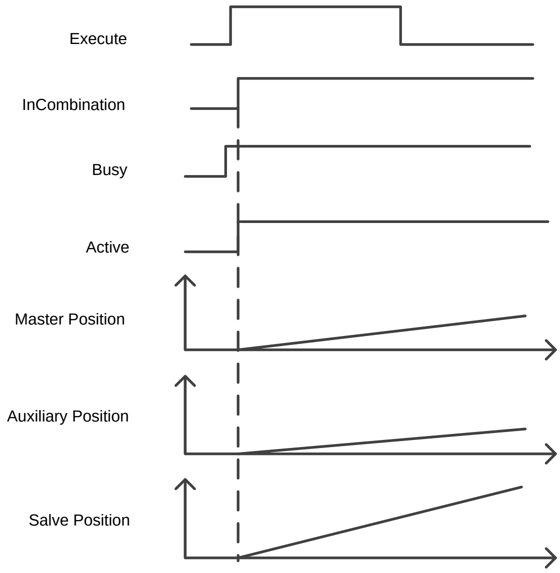
If triggered again during the validity period of the Busy signal of the MC_CombineAxes command, this command will re-latch the current input parameters, and the slave axis will perform a follow-up action based on the calculation result and determine whether InCombination is set based on the calculation result.

Multiple Calls

During the validity period of the Busy signal of the MC_CombineAxes command, if the second command is triggered, the Busy signal of the second command is valid, the first command is interrupted, and the CommandAborted output ON of the first command is valid.

Timing diagram

CombineMode is set to addition.



3.21.43 Error Codes of Master and Slave Axis Commands

Main error code	Secondary error code	Error level	Possible cause	Solution
0x11 (17) Operation control fault	0xC9(201)	Warning	The master and slave axes use the same axis ID	Check whether the master and slave axes are the same
	0xCA(202)	Fault	Input parameter error of MC_GearOut function block	Check whether the input parameters of GearOut are within the constraint range of the command parameter list
	0xCB(203)	Warning	It is invalid to trigger the MC_GearOut function block	Check whether the slave axis is in the gear action Check whether the slave axis is in the gear disengagement action
	0xCC(204)	Fault	Input parameter error of MC_GearIn function block	Check whether the input parameters of MC_GearIn are within the constraint range of the command parameter list
	0xCD(205)	Warning	The current command to run the master axis does not meet the requirements	Check whether the master axis state meets the requirements With the MC_Phasing command running, check whether the current axis is in the process of cam or gear operation
	0xCE(206)	Warning	The master axis has not reached the target velocity	Check whether the current master axis has reached the target velocity
	0xCF(207)	Fault	Input parameter error of MC_CamOut function block	Check whether the input parameters of MC_CamOut are within the constraint range of the command parameter list
	0xD0(208)	Warning	It is invalid to trigger the MC_CamOut command	Check whether the slave axis is in the cam action Check whether the slave axis is in the cam disengagement action
	0xD1(209)	Fault	Input parameter error of MC_CamIn function block	Check whether the input parameters of MC_CamIn are within the constraint range of the command parameter list

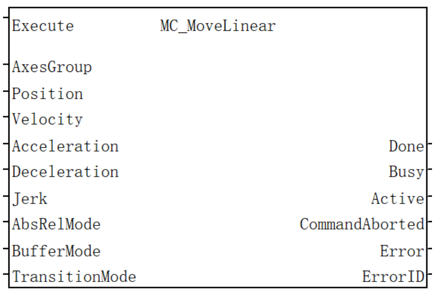
Main error code	Secondary error code	Error level	Possible cause	Solution
	0xD2(210)	Warning	The current CamTable ID is not within the valid range	Check whether the CamTable ID is within the constraint range of the command parameter list
	0xD3(211)	Warning	Setting error of StartPosition or MasterStartDistance in MC_CamIn command	Check whether MasterStart Distance and Start Position are in the current master axis running direction in the absolute position mode
	0xD4(212)	Warning	When the MC_CamIn command is in the absolute position mode, StartPosition ahead of Master Start Distance	Check whether StartPosition is ahead of MasterStartDistance in the absolute position mode
	0xD5(213)	Fault	The input parameters of the MC_Phasing command are not within the valid range	Check whether the input parameters of MC_Phasing are within the constraint range of the command parameter list
	0xD6(214)	Warning	Reserved	-
	0xD7(215)	Warning	Reserved	-
	0xD8((216)	Warning	Reserved	-
	0xD9(217)	Warning	Reserved	-
	0xDA(218)	Warning	Reserved	-
	0xDB(219)	Warning	Reserved	-
	0xDC(220)	Warning	Reserved	-
	0xDD(221)	Warning	Reserved	-
	0xDE(222)	Warning	Reserved	-
	0xDF(223)	Warning	Reserved	-
	0xE0(224)	Warning	Reserved	-
	0xE1(225)	Warning	Master axis phase setting error	Check whether the master axis phases of two adjacent keypoints are less than or equal to 0.001 in the user-defined cam table of the MC_GenerateCamTable command
	0xE2(226)	Warning	The start point of the cam table cannot be set as a non-zero parameter	Check whether the positions of the master and slave axes at the start point of the cam are set to non-zero in the user-defined cam table of the MC_GenerateCamTable command
	0xE3(227)	Warning	The current NodeNum parameter cannot be set to 0	Check whether the MC_NodeNum parameter is set to 0 in the current mode in the GenerateCamTable command
	0xE4(228)	Warning	The current NodeNum parameter is not within the valid range	Check whether the MC_NodeNum parameter is set within the constraint range of the command parameter list in the current mode in the GenerateCamTable command
	0xE5(229)	Warning	Curve type setting error in cam table	Check whether the cam curve type settings are within the constraint range of the command parameters list. They only support 0 (which represents straight lines) and 1 (which represents quintic curves)
	0xE6(230)	Warning	The cam table is empty	Check whether the cam table is configured

Main error code	Secondary error code	Error level	Possible cause	Solution
	0xE7(231)	Warning	Encoder master axis enable failed	Check whether the counting command ENC_Counter is enabled when using the encoder master axis
	0xE8(232)	Warning	The length of the user-defined cam table is not within the valid range	Check that the length of the user-defined cam table array must be 32 in the MC_GenerateCamTable command
	0xE9(233)	Warning	The the user-defined tappet switch is not within the valid range	Check that the length of the user-defined switch array must be 32 in the MC_DigitalCamSwitch command
	0xEA(234)	Warning	The ReferenceType parameter settings are not within the valid range	Check whether ReferenceType parameter settings are within the valid range for the current command
	0xEB(235)	Warning	The Channel parameter settings are not within the valid range	Check whether Channel parameter settings are within the valid range for the current command
	0xEC(236)	Warning	The Number parameter settings are not within the valid range	Check whether Number parameter settings are within the valid range for the current command
	0xED(237)	Warning	The address of the Switches parameter is NULL	Check whether the Switches parameter has a given variable in the current command
	0xEE(238)	Warning	Positions are not arranged in ascending order in the tappet switch	Check whether Position in the Switches parameter is set to ascending order in the current command. If not, modify it.
	0xEF(239)	Warning	The current axis state does not support the use of the tappet command	Check whether the axis is in the home state
	0xF0(240)	Warning	The Action settings are not within the valid range for the tappet switch	Check whether Action in the Switches parameter is within the valid range for the current command
	0xF1(241)	Warning	The current Channel is already in use	Check if there is any reuse of Channel
	0xF2(242)	Warning	The Position settings in the tappet switch exceeds the rotation axis modulus cycle	Check whether Position in the Switch parameter exceeds the rotation cycle value in the rotation axis mode for the current command
	0xF3(243)	Fault	The input parameters of the MC_CombineAxes command are not within the valid range	Check whether the command parameters are within the valid range, and call the MC_Reset command to reset the axis state
	0xF4(244)	Warning	Phase of the MC_GetCamTableDistance command is not within the valid range between the start and end points	Check whether the input parameter Phase of this command is within the valid range between the start and end points
	0xF5(245)	Warning	The CurveType parameter settings are not within the valid range	Check whether CurveType parameter settings are within the valid range for the current command
	0xF6(246)	Warning	The phases of the start and end points for the MC_GetCamTableDistance command is not arranged in	Check if the phase difference between the start and end points for this command is less than 0.001 Check whether Phase in CamTable is

Main error code	Secondary error code	Error level	Possible cause	Solution
			ascending order	in ascending order
	0xF7(247)	Warning	The current master axis has entered the ErrorStop state, and the function block has stopped running	Check the reason why the master axis has entered the ErrorStop state
	0xF8(248)	Warning	Multiple cam table save commands are used on the same axis	Check whether multiple cam table save commands are used on the same axis in the user program
	0xF9 (249)	Warning	The cam table update command was not completed and the cam table save command was called instead	Check whether the user program has not completed the cam table update command and has called the cam table save command instead
	0xFA (250)	Warning	The MC_GetCamTablePhase command output parameter Phase arrey length was not equal to 6	Check whether the MC_GetCamTablePhase command output parameter Phase arrey length is equal to 6
	0xFB (251)	Warning	The acceleration settings at the starting and ending points of MC_GetCamTablePhase command were abnormal	Check the acceleration settings at the starting and ending points of MC_GetCamTablePhase command
	0xFC (252)	Warning	The MC_GetCamTablePhase command inout parameter Distance was not within the valid range of the cam table	Check the MC_GetCamTablePhase command Distance settings
	0xFD (253)	Warning	The MC_GetCamTablePhase command start point and end point references were abnormal	Please contact us
	0xFE (254)	Warning	The MC_GearInPos commandinput parameter was not within the valid range	Check whether the command input parameter is within the valid range

3.21.44 MC_MoveLinear

Graphic Block



16-Bit command	-					
32-Bit command	MoveLinear: Linear Interpolation					
Operand	**Name**	**Description**	**Nullable**	**Default value**	**Range**	**Data Type**
S1	AxesGroup	Axis group number	No	-	0~7	INT
S2	Position	Positive/negative/0	No	-	Array (positive/ negative/0)	REAL [4]

16-Bit command	-					
32-Bit command	MoveLinear: Linear Interpolation					
Operand	**Name**	**Description**	**Nullable**	**Default value**	**Range**	**Data Type**
S3	Velocity	Positive number, which is the maximum absolute value of the velocity	Yes	100	Positive number	REAL
S4	Acceleration	Positive number, which is the maximum absolute value of the acceleration or deceleration	Yes	1000	Positive number	REAL
S5	Jerk	Jerk=0: T-type acceleration and deceleration Jerk>0: S-type acceleration and deceleration; the larger the positive number, the poorer the S-type acceleration and deceleration effect	Yes	0	Positive/0	REAL
S6	AbsRelMode	0: absolute mode 1: relative mode	Yes	0	0~1	INT
D1	Done	Execution completion flag	Yes	OFF	ON/OFF	BOOL
D2	Busy	Ongoing execution flag	Yes	OFF	ON/OFF	BOOL
D3	Active	Execution validity flag	Yes	OFF	ON/OFF	BOOL
D4	CommandAborted	Execution interrupt flag	Yes	OFF	ON/OFF	BOOL
D5	Error	Command error flag	Yes	OFF	ON/OFF	BOOL
D6	ErrorID	Error code, which displays error information	Yes	0	Positive/0	WORD

Operand	**Soft Elements**						
	Const	**D**	**R**	**Y**	**M**	**S**	**Custom Variable**
S1	√			-	-	-	-
S2		√	√	-	-	-	-
S3	√	√	√	-	-	-	√
S4	√	√	√	-	-	-	√
S5	√	√	√	-	-	-	√
S6	√	√	√	-	-	-	√
D1	-	-	-	√	√	√	√
D2	-	-	-	√	√	√	√
D3	-	-	-	√	√	√	√
D4	-	-	-	√	√	√	√
D5	-	-	-	√	√	√	√
D6	-	√	√	-	-	-	√

Function Description

1. The rising edge triggers parameters to take effect, and this module is used to implement linear interpolation function, which support two-axis, three-axis, and four-axis linear interpolations.
2. When there is a single-axis motion in the axis group, the axis group command cannot be started. Therefore, you should ensure that: before starting the command, all axes in the axis group are in the enabled state; after starting, the axis group state switches from 6 (GroupStandStill) to 8 (GroupSynchronizedMotion). The axis group states are shown in the table below:

Axis Group Status	Status Code	State instruction
SingleDisabled	1	There is an axis in the disable state in the axis group
SingleStop	2	There is an axis in the single axis stop state in the axis group
SingleHoming	3	There is an axis in the homing state in the axis group
SingleMotion	4	There is an axis in the single axis or master-slave axes motion state in the axis group
GroupErrorStop	5	There is an axis in the error state in the axis group
GroupStandStill	6	All axes in the axis group are enabled
GroupStopping	7	Axis Group Stop or Axis Group Immediate Stop is called
GroupSynchronizeMotion	8	The function block of the axis group is pulled up and successfully enters this state

3. This command provide absolute and relative motion modes, and sets the current coordinate positions to (px, py) and the parameter target positions to (posx, posy). In case of AbsRelMode=0, it plans the target positions to (posx, posy); in case of AbsRelMode=1, it plans the target positions to (px+posx, py+posy).
4. The number of axes involved in linear interpolation is determined by the number of axes configured in the "Basic Settings" section of "Axis Group Settings" in the host computer.
5. This command and commands MC_MoveCircular2D and MC_GroupHalt are allowed to interrupt each other, and using the MC_GroupSetOverride axis group velocity control function can adjust the velocity of interpolation operation online.

Resetting This Command

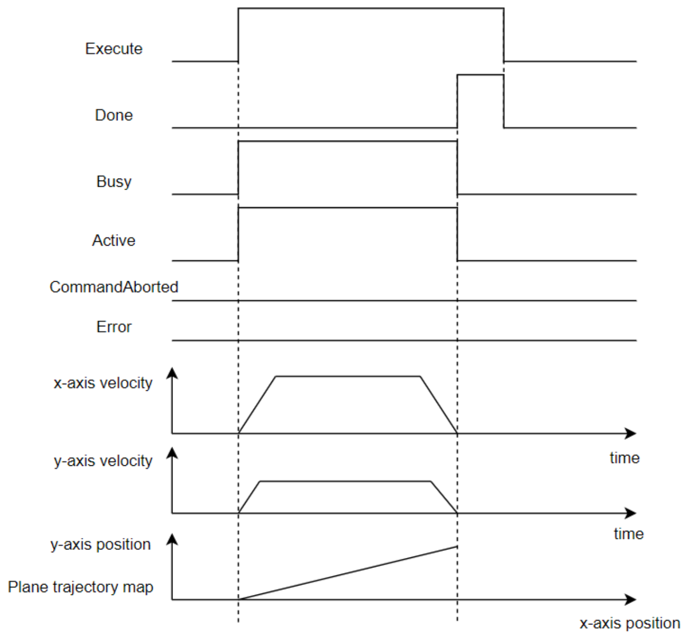
If triggered again during the validity period of its Busy signal, this command re-plans with new target parameters according to the current motion position, velocity, etc.

Multiple Starts of This Command

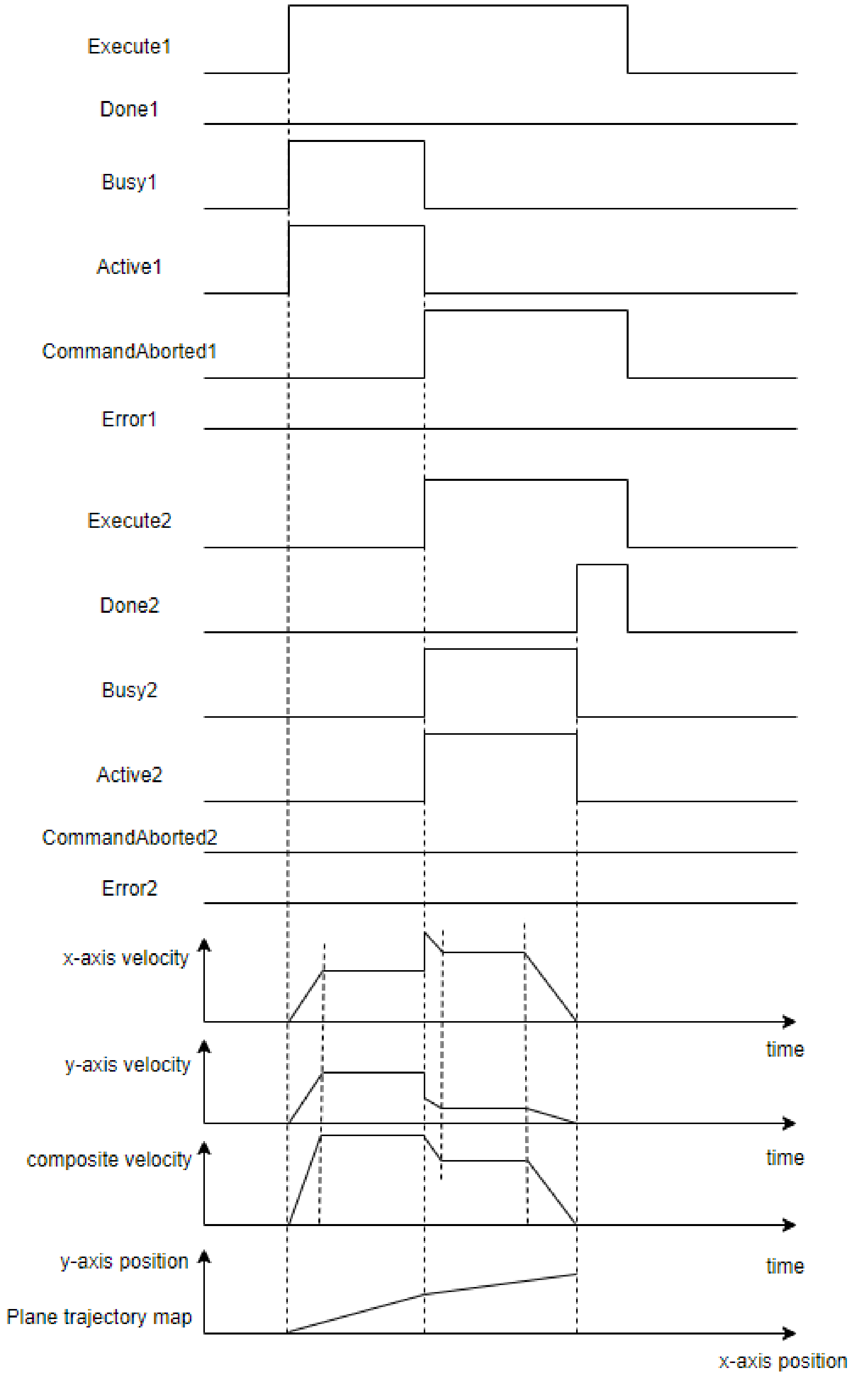
When multiple commands call the same axis group, if the next command is triggered during the Busy signal validity period of the previous command, the next command will take effect and re-plans with new target parameters according to the current motion position, velocity, etc., and the previous command will be interrupted and invalidated.

Timing diagram

A single command is called to perform the linear interpolation on the X-axis and Y-axis planes. Note that the Position parameter is a REAL type array with a length of 4. Here D0~D7 are used as input element for this parameter to obtain D0=200, D2=80, D4=0, and D6=0.

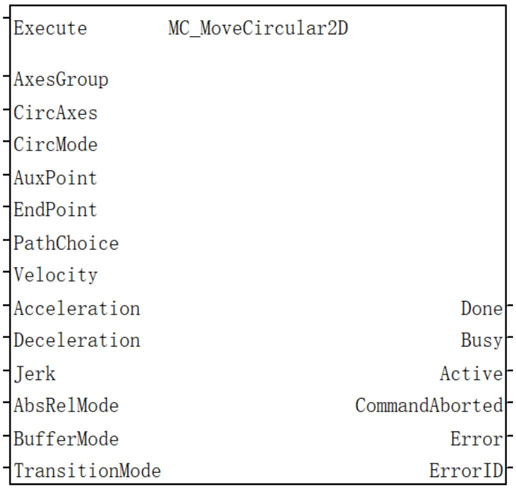


For interrupting commands of the same type, two commands are called. During the operation of the first command, the second command is pulled up to modify the target composite velocity and target position. Note that a slight jump occurs in the single-axis velocity during the interrupt cycle. In this example, you can see the starting position (x, y)=(40,0), the interruption position (x, y)=(100,30), and the new target position (x, y)=(160,45). During the interrupt cycle, the x-axis velocity jump becomes larger, the y-axis velocity jump becomes smaller, and the composite velocity of the axis group remains unchanged. Afterwards, the velocity of the axis group gradually changes to the new target value.



3.21.45 MC_MoveCircular2D

Graphic Block



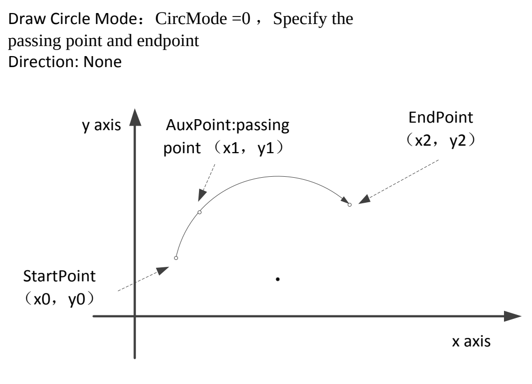
16-Bit command	-					
32-Bit command	MC_MoveCircular2D: Plane Arc Interpolation					
Operand	**Name**	**Description**	**Nullable**	**Default value**	**Range**	**Data Type**
S1	AxesGroup	Axis group number	No	-	0~7	WORD
S2	CircAxes	0: x-y plane 1: y-z plane 2: x-z plane	Yes	0	0~2	INT
S3	CircMode	0: _mcBorder, specified as the pass point 1: _mcCenter, specified as the circle center 2: _mcRadius, specified as the radius	Yes	0	0~2	INT
S4	AuxPoint	CircMode=0: coordinates of pass midpoint CircMode=1: coordinates of circle center CircMode=2: coordinates of radius (only AuxPoint [0] is assigned; note that positive numbers select the superior arc, while negative numbers select the inferior arc) Superior arc: an arc with a center angle greater than 180°, that is, the arc longer than a semicircle	No	-	Array (positive/ negative/0)	REAL [2]
S5	EndPoint	End point of arc	No	-	Array (positive/ negative/0)	REAL [2]
S6	Velocity	Positive number, which is the maximum absolute value of the velocity	Yes	100	Positive number	REAL

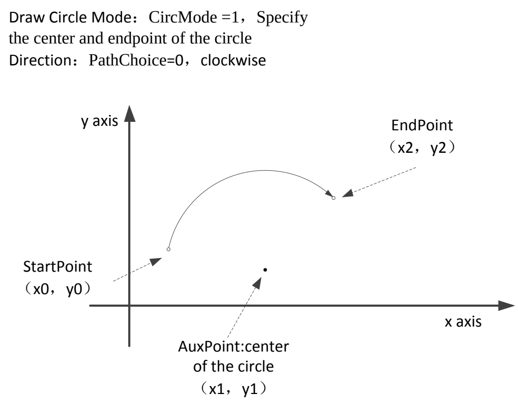
16-Bit command	-					
32-Bit command	MC_MoveCircular2D: Plane Arc Interpolation					
Operand	**Name**	**Description**	**Nullable**	**Default value**	**Range**	**Data Type**
S7	Acceleration	Positive number, which is the maximum absolute value of the acceleration or deceleration	Yes	1000	Positive number	REAL
S8	PathChoice	It is used as a supplementary condition 0: clockwise 1: counterclockwise	Yes	0	0~1	INT
S9	Jerk	Jerk=0: T-type acceleration and deceleration; Jerk>0: S-type acceleration and deceleration; the larger the positive number, the poorer the S-type acceleration and deceleration effect	Yes	0	Positive/0	REAL
S10	AbsRelMode	0: absolute mode 1: relative mode	Yes	0	0~1	INT
D1	Done	Execution completion flag	Yes	OFF	ON/OFF	BOOL
D2	Busy	Ongoing execution flag	Yes	OFF	ON/OFF	BOOL
D3	Active	Execution validity flag	Yes	OFF	ON/OFF	BOOL
D4	Command Aborted	Execution interrupt flag	Yes	OFF	ON/OFF	BOOL
D5	Error	Command error flag	Yes	OFF	ON/OFF	BOOL
D6	ErrorID	Error code, which displays error information	Yes	0	Positive/0	WORD

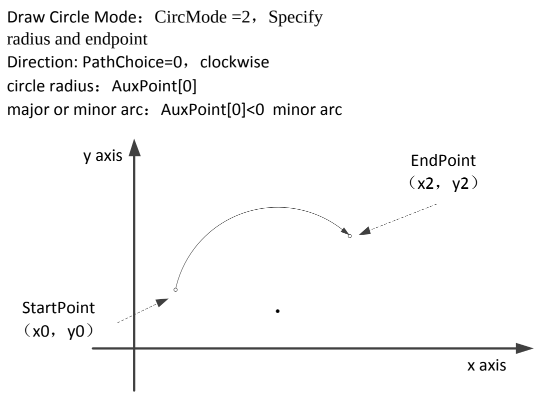
Operand	Soft Elements					
	Const	**D**	**R**	**Y**	**M**	**S**	**Custom Variable**
S1	√	-	-	-	-	-	-
S2	√	√	√	-	-	-	√
S3	√	√	√	-	-	-	√
S4	-	√	√	-	-	-	-
S5	-	√	√	-	-	-	-
S6	√	√	√	-	-	-	√
S7	√	√	√	-	-	-	√
S8	√	√	√	-	-	-	√
S9	√	√	√	-	-	-	√
S10	√	√	√	-	-	-	√
D1	-	-	-	√	√	√	√
D2	-	-	-	√	√	√	√
D3	-	-	-	√	√	√	√
D4	-	-	-	√	√	√	√
D5	-	-	-	√	√	√	√
D6	-	√	√	-	-	-	√

Function Description

1. The rising edge triggers parameters to take effect, and this module is used to implement arc interpolation function. Users can specify two axes within the axis group to participate in arc interpolation through CircAxes.
2. When there is a single-axis motion in the axis group, the axis group command cannot be started. Therefore, you should ensure that: before starting the command, all axes in the axis group are in the enabled state; after starting, the axis group state switches from 6 (GroupStandStill) to 8 (GroupSynchronizedMotion).
3. This command provide absolute and relative motion modes, and sets the current coordinate positions to (px, py) and the parameter target positions to (posx, posy). In case of AbsRelMode=0, it plans the target positions to (posx, posy); in case of AbsRelMode=1, it plans the target positions to (px+posx, py+posy).
4. The arc interpolation function supports three circle drawing modes, which are selected through the parameter CircMode. Note that the start point is the current position of the PLC, and the three modes are CircMode=0 (known start point, midpoint, and end point), CircMode=1 (known start point, circle center coordinates, and endpoint), and CircMode=2 (known start point, radius, and endpoint). In case of CircMode=2, the radius is set by AuxPoint [0], which selects the superior arc when set to a positive number or the inferior arc when set to a negative number.
5. This command and commands MC_MoveLinear and MC_ GroupHalt are allowed to interrupt each other, and using the MC_GroupSetOverride axis group velocity control function can adjust the velocity of interpolation operation online.
6. In addition to the CircMode=0 mode, the direction of circle drawing is determined by the parameter PathChoice supplementarily. See below for the diagrams of drawing circles through CircMode=0, CircMode=1, and CircMode=2.







Resetting This Command

If triggered again during the validity period of its Busy signal, this command re-plans with new target parameters according to the current motion position, velocity, etc.

Multiple Starts of This Command

When multiple commands call the same axis group, if the next command is triggered during the Busy signal validity period of the previous command, the next command will take effect and re-plans with new target parameters according to the current motion position, velocity, etc., and the previous command will be interrupted and invalidated.

Timing diagram

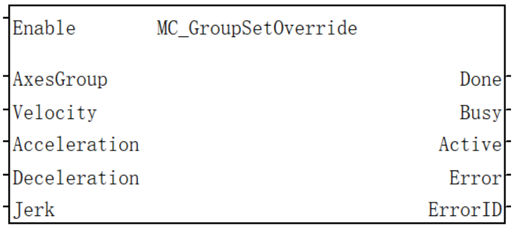
For the timing diagram of arcual interpolation, see the timing diagram of linear interpolation in the previous section.

3.21.46 MC_MoveEllipse

Reserved

3.21.47 MC_GroupSetOverride

Graphic Block



16-Bit command	-					
32-Bit command	MC_GroupSetOverRide: Axis Group Velocity Regulation					
Operand	**Name**	**Description**	**Nullable**	**Default value**	**Range**	**Data Type**
S1	AxesGroup	Axis group number	No	-	0~7	INT
S2	Velocity	Positive number, which is the maximum absolute value of the velocity	Yes	100	Positive number	REAL
S3	Acceleration	Positive number, which is the maximum absolute value of the acceleration or deceleration	Yes	1000	Positive number	REAL
S4	Jerk	Jerk=0: T-type acceleration and deceleration Jerk>0: S-type acceleration and deceleration; the larger the positive number, the poorer the S-type acceleration and deceleration effect	Yes	0	Positive/0	REAL
D1	Done	Execution completion flag	Yes	OFF	ON/OFF	BOOL
D2	Busy	Ongoing execution flag	Yes	OFF	ON/OFF	BOOL
D3	Active	Execution validity flag	Yes	OFF	ON/OFF	BOOL
D4	CommandAborted	Execution interrupt flag	Yes	OFF	ON/OFF	BOOL
D5	Error	Command error flag	Yes	OFF	ON/OFF	BOOL
D6	ErrorID	Error code, which displays error information	Yes	0	Positive/0	WORD

Operand	Soft Elements						
	Const	D	R	Y	M	S	Custom Variable
S1	√	-	-	-	-	-	-
S2	√	√	√	-	-	-	√
S3	√	√	√	-	-	-	√
S4	√	√	√	-	-	-	√
D1	-	-	-	√	√	√	√
D2	-	-	-	√	√	√	√
D3	-	-	-	√	√	√	√
D4	-	-	-	√	√	√	√
D5	-	-	-	√	√	√	√
D6	-	√	√	-	-	-	√

Function Description

1. The high-level parameters take effect and are used to achieve the online variable velocity (composite velocity) processing of the axis group motion module.
2. The velocity regulation function does not change the axis group state machines.
3. The modified command parameters are user input values, which are constrained by the maximum values in the axis group configuration.
4. This command allows the parameters to be pulled up when the axis group state is 6 (GroupStandStill) and also allows the online modified parameters to take effect. After pulling down the velocity control function, interpolation restores the original interpolation velocity.

Resetting This Command

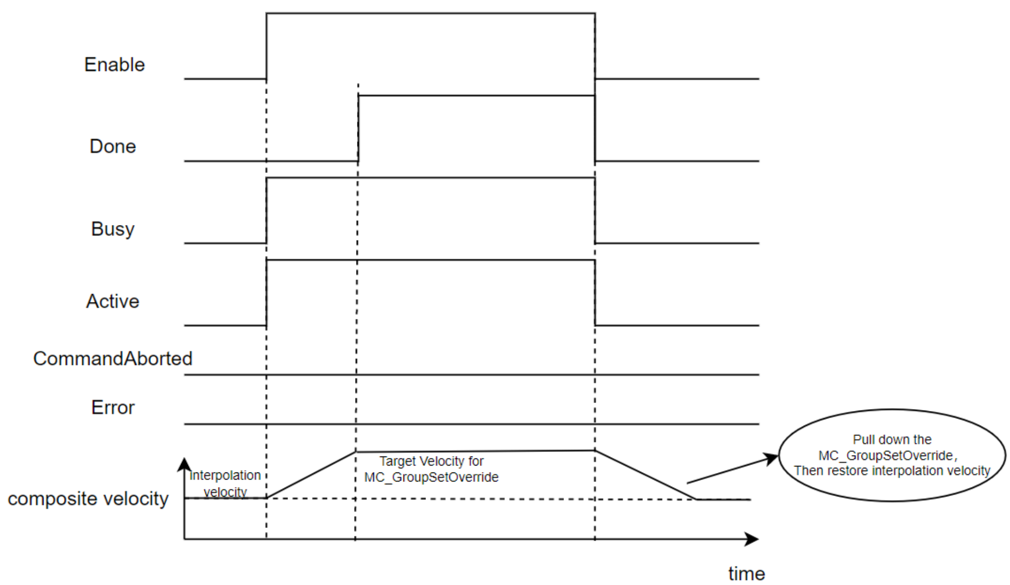
As an Enable-type command, this command becomes invalid when pulled down, is valid at high levels, and can be modified online.

Multiple Starts of This Command

When multiple commands call the same axis group, if the next command is triggered during the Busy signal validity period of the previous command, the next command will report error code 15, which means that multiple starts are not supported.

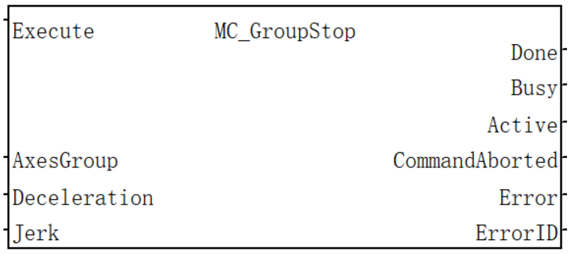
Timing diagram

A single command is called to trigger the velocity regulation command during the interpolation operation.



3.21.48 MC_GroupStop

Graphic Block



16-Bit command	-					
32-Bit command	MC_GroupStop: Axis Group Stop					
Operand	**Name**	**Description**	**Nullable**	**Default value**	**Range**	**Data Type**
S1	AxesGroup	Axis group number	No	-	0~7	INT
S2	Deceleration	Positive number, which is the maximum absolute value of the deceleration	Yes	1000	Positive number	REAL
S3	Jerk	Jerk=0: T-type acceleration and deceleration; Jerk>0: S-type acceleration and deceleration; the larger the positive number, the poorer the S-type acceleration and deceleration effect	Yes	0	Positive/0	REAL
D1	Done	Execution completion flag	Yes	OFF	ON/OFF	BOOL
D2	Busy	Ongoing execution flag	Yes	OFF	ON/OFF	BOOL
D3	Active	Execution validity flag	Yes	OFF	ON/OFF	BOOL
D4	CommandAborted	Execution interrupt flag	Yes	OFF	ON/OFF	BOOL
D5	Error	Command error flag	Yes	OFF	ON/OFF	BOOL
D6	ErrorID	Error code, which displays error information	Yes	0	Positive/0	WORD

Operand	**Soft Elements**						
	Const	**D**	**R**	**Y**	**M**	**S**	**Custom Variable**
S1	√	-	-	-	-	-	-
S2	√	√	√	-	-	-	√
S3	√	√	√	-	-	-	√
D1	-	-	-	√	√	√	√
D2	-	-	-	√	√	√	√
D3	-	-	-	√	√	√	√
D4	-	-	-	√	√	√	√
D5	-	-	-	√	√	√	√
D6	-	√	√	-	-	-	√

Function Description

1. The rising edge triggers parameters to take effect, and this module is used to stop the existing motion of the axis group (except for MC_GroupImmediateStop).

2. When there is a single-axis motion in the axis group, the axis group command cannot be started. Therefore, you should ensure that: before starting the command, the axis group is in state 6 (GroupStandStill) or state 8 (GroupSynchronizedMotion); after starting, the axis group state is 7 (GroupStopping).
3. After GroupStop stops, GroupStop must be pulled down to switch the axis group state back to 6 (GroupStandStill), so that new interpolation action can be performed.
4. When the deceleration settings are unreasonable, "Axis Group Fault Deceleration" in the axis group configuration interface will be used first. If the deceleration is still unreasonable, the axis will stop immediately.
5. This command and commands MC_MoveCircular2D and MC_GroupHalt are allowed to interrupt each other, and using the MC_GroupSetOverride axis group velocity control function can adjust the velocity of interpolation operation online.

Resetting This Command

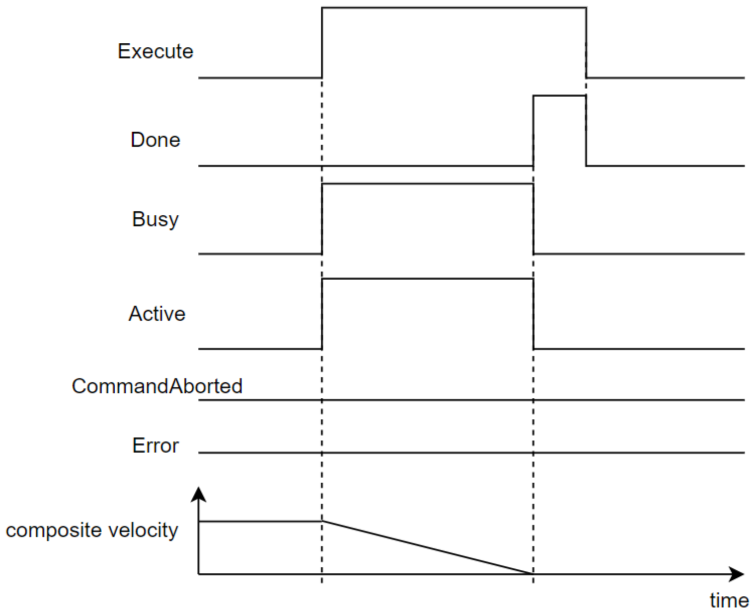
If triggered again during the validity period of its Busy signal, this command re-plans with new target parameters according to the current motion position, velocity, etc.

Multiple Starts of This Command

When multiple commands call the same axis group, if the next command is triggered during the Busy signal validity period of the previous command, the previous command still runs normally, but the next command can not take effect and reports error code 309, which means that multiple starts are not supported.

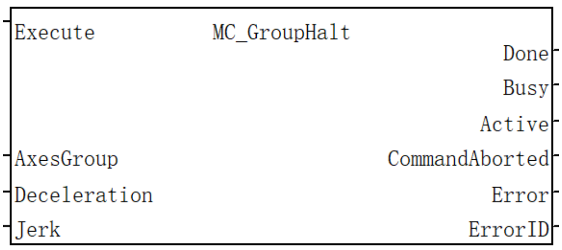
Timing diagram

A single command is called to perform deceleration-based stop.



3.21.49 MC_GroupHalt

Graphic Block



16-Bit command	-					
32-Bit command	MC_GroupHalt: Axis Group Halt					
Operand	**Name**	**Description**	**Nullable**	**Default value**	**Range**	**Data Type**
S1	AxesGroup	Axis group number	No	-	0~7	INT
S2	Deceleration	Positive number, which is the maximum absolute value of the deceleration	Yes	1000	Positive number	REAL
S3	Jerk	Jerk=0: T-type acceleration and deceleration Jerk>0: S-type acceleration and deceleration; the larger the positive number, the poorer the S-type acceleration and deceleration effect	Yes	0	Positive/0	REAL
D1	Done	Execution completion flag	Yes	OFF	ON/OFF	BOOL
D2	Busy	Ongoing execution flag	Yes	OFF	ON/OFF	BOOL
D3	Active	Execution validity flag	Yes	OFF	ON/OFF	BOOL
D4	CommandAborted	Execution interrupt flag	Yes	OFF	ON/OFF	BOOL
D5	Error	Command error flag	Yes	OFF	ON/OFF	BOOL
D6	ErrorID	Error code, which displays error information	Yes	0	Positive/0	WORD

Operand	**Soft Elements**					
	Const	**D**	**R**	**Y**	**M**	**S**	**Custom Variable**
S1	√						
S2	√	√	√				√
S3	√	√	√				√
D1				√	√	√	√
D2				√	√	√	√
D3				√	√	√	√
D4				√	√	√	√
D5				√	√	√	√
D6		√	√				√

Function Description

1. The rising edge triggers parameters to take effect, and this module is used to halt the existing motion of the axis group (except for MC_GroupImmediateStop and MC_GroupStop).
2. When there is a single-axis motion in the axis group, the axis group command cannot be started. Therefore, you should ensure that: before starting the command, the axis group is in state 6 (GroupStandStill) or state 8 (GroupSynchronizedMotion); after starting, the axis group state is 8 (GroupSynchronizedMotion); after stopping, the axis group state is 6 (GroupStandStill).
3. This command and commands MC_MoveCircular2D and MC_GroupLinear are allowed to interrupt each other. This command does not support calling the velocity regulation module for velocity regulation during deceleration.

Resetting This Command

If triggered again during the validity period of its Busy signal, this command re-plans with new target parameters according to the current motion position, velocity, etc.

Multiple Starts of This Command

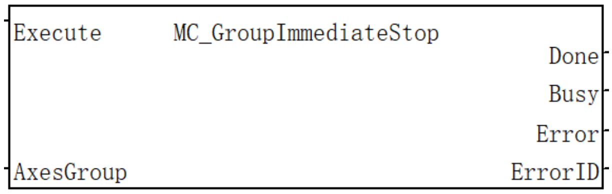
When multiple commands call the same axis group, if the next command is triggered during the Busy signal validity period of the previous command, the next command will take effect and re-plans with new target parameters according to the current motion position, velocity, etc., and the previous command will be interrupted and invalidated.

Timing diagram

See the timing diagram for MC_GroupStop.

3.21.50 MC_GroupImmediateStop

Graphic Block



16-Bit command	MC_GroupImmediateStop: Immediate Axis Group Stop					
32-Bit command	-					
Operand	**Name**	**Description**	**Nullable**	**Default value**	**Range**	**Data Type**
S1	AxesGroup	Axis group number	No	-	0~7	INT
D1	Done	Execution completion flag	Yes	OFF	ON/OFF	BOOL
D2	Busy	Ongoing execution flag	Yes	OFF	ON/OFF	BOOL
D3	Error	Command error flag	Yes	OFF	ON/OFF	BOOL
D4	ErrorID	Error code, which displays error information	Yes	0	Positive/0	WORD

Operand	Soft Elements						
	Const	D	R	Y	M	S	Custom Variable
S1	√	-	-	-	-	-	-
D1	-	-	-	√	√	√	√
D2	-	-	-	√	√	√	√
D3	-	-	-	√	√	√	√
D4	-	√	√	-	-	-	√

Function Description

1. The rising edge triggers parameters to take effect, and this module is used to immediately stop the existing motion of the axis group. This command has the highest interrupt priority.
2. When there is a single-axis motion in the axis group, the axis group command cannot be started. Therefore, you should ensure that: before starting the command, the axis group is in state 6 (GroupStandStill) or state 8 (GroupSynchronizedMotion); after starting, the axis group state is 7 (GroupStopping).
3. After GroupImmediateStop stops, GroupImmediateStop must be pulled down to switch the axis group state back to 6 (GroupStandStill), so that new interpolation action can be performed.

Resetting This Command

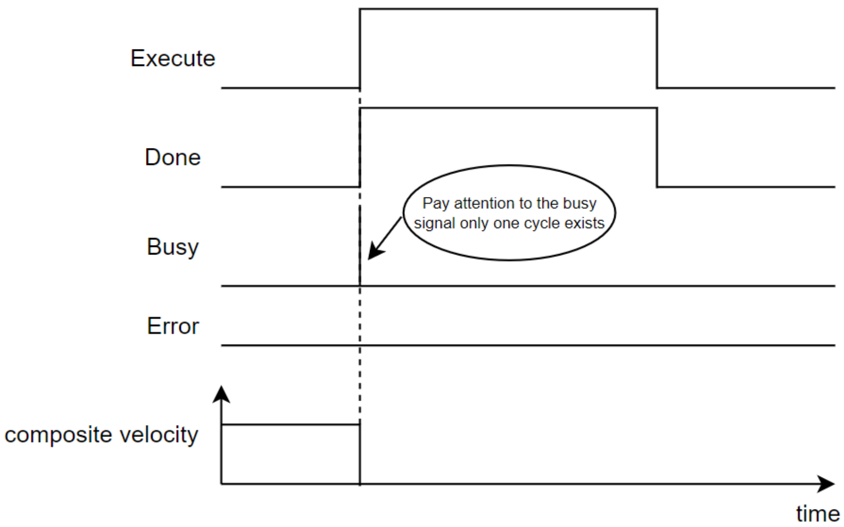
After a successful pull-up, if you pull up this command again, it still outputs Done.

Multiple Starts of This Command

When multiple commands call the same axis group, if the next command is triggered during the Busy signal validity period of the previous command, the previous command outputs Done, but the next command can not take effect and reports error code 307, which means that multiple starts are not supported.

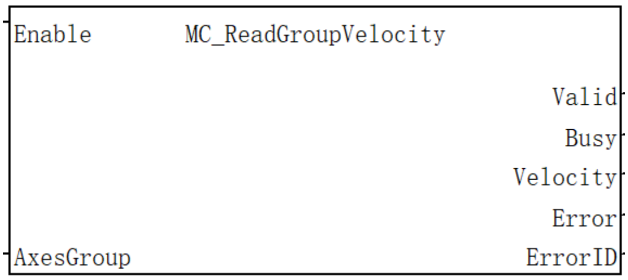
Timing diagram

A single command is called.



3.21.51 MC_ReadGroupVelocity

Graphic Block



16-Bit command	-					
32-Bit command	MC_ReadGroupVelocity: Read Composite Axis Group Velocity					
Operand	**Name**	**Description**	**Nullable**	**Default value**	**Range**	**Data Type**
S1	AxesGroup	Axis group number	No	-	0~7	INT
D1	Valid	Execution validity flag	Yes	OFF	ON/OFF	BOOL
D2	Busy	Ongoing execution flag	Yes	OFF	ON/OFF	BOOL
D3	Velocity	Axis group command velocity	Yes	0	Positive/negative/0	REAL
D4	Error	Command error flag	Yes	OFF	ON/OFF	BOOL
D5	ErrorID	Error code, which displays error information	Yes	0	Positive/0	WORD

Operand	Soft Elements						
	Const	D	R	Y	M	S	Custom Variable
S1	√	-	-	-	-	-	-
D1	-	-	-	√	√	√	√
D2	-	-	-	√	√	√	√
D3	-	√	√	-	-	-	√
D4	-	-	-	√	√	√	√
D5	-	√	√	-	-	-	√

Function Description

This command is valid at high levels, and the module is used to read the composite velocity of the specified axis group.

Resetting This Command

This command is valid at high levels, and it becomes invalid when pulled up or pulled down.

Multiple Starts of This Command

When multiple commands are run to call the same axis group, they don’ t affect each other.

3.21.52 Fault Codes of Axis Group Commands

Main error code	Secondary error code	Error level	Possible cause	Solution
0x11 (17) Operation control fault	0x12D(301)	Fault	Input parameter error of function block	In plane arc interpolation mode 2, if the distance between the start and end points is greater than twice the radius, check and correct the parameters
	0x12E(302)	Fault	Axis group ID settings exceeds the range	Check and correct the axis group ID
	0x12F(303)	Fault	Two or more identical axis IDs are configured in the axis group	Check and correct the duplicated axis IDs in the axis group configuration interface
	0x130(304)	Fault	The distance from the start end to the circle center is not equal to that from the end point to the circle center in the plane arc function block	In plane arc interpolation mode 1, check and modify the distance from the start point to the circle center and that from the and end point to the circle center
	0x131(305)	Fault	The start point, circle center, and end point are on the same straight line in the plane arc function block	In plane arc interpolation mode 0, ensure that the start point, auxiliary point, and end point are on the same straight line
	0x132(306)	Fault	The calculated circle center position is not unique in the plane arc function block	In plane arc interpolation mode 2, ensure that the start point is equal to the end point
	0x133(307)	Fault	In the GroupImmediateStop module, the same axis group can only call this function block once, and the second function block starts reporting an error	For the same axis group, the second immediate axis group stop module reports error
	0x134(308)	Fault	The axis group is in the GroupImmediate Stopping	Pull down the MC_GroupImmediateStop module

Main error code	Secondary error code	Error level	Possible cause	Solution
			state	first, and then pull up the MC_GourpStop module
	0x135(309)	Fault	In the GroupStop module, the same axis group can only call this function block once, and the second function block starts reporting an error	For the same axis group, the second MC_GroupStop module reports error when pulled up
	0x136(310)	Fault	The configured velocity parameters are not within a reasonable range	Check the corresponding parameters
	0x137(311)	Fault	The configured acceleration parameters are not within a reasonable range	Check the corresponding parameters
	0x138(312)	Fault	The configured deceleration parameters are not within a reasonable range	Check the corresponding parameters
	0x139(313)	Fault	The configured Jerk parameters are not within a reasonable range	Check the corresponding parameters
	0x13A(314)	Fault	The configured AbsRelMode parameters are not within a reasonable range	Check the corresponding parameters
	0x13B(315)	Fault	Interpolation is not allowed as there a single axis is in the rotation mode in the axis group	De-select the rotation mode option in the single axis configuration interface
	0x13C(316)	Fault	Interpolation is not allowed as there a single axis is in the debugging mode in the axis group	De-select the debugging mode option in the single axis configuration interface
	0x13D(317)	Fault	The radius parameter is not allowed to be zero	Check the corresponding parameters
	0x13E(318)	Fault	The parameter CircAxes is not within the allowed range	Check the corresponding parameters
	0x13F(319)	Fault	The parameter CircMode is not within the allowed range	Check the corresponding parameters
	0x140(320)	Fault	The parameter PathChoice is not within the allowed range	Check the corresponding parameters
	0x141(321)	Fault	The array parameters passed in by the host computer are incorrect	Enable host computer error protection
	0x142(322)	Fault	It is not allowed to modify the parameter CircAxes during the operation of arc interpolation	During the operation of arc interpolation, it is not allowed to modify the parameter CircAxes
	0x143(323)	Fault	The current state does not allow axis group velocity regulation	The current state does not allow axis group velocity regulation, including moderate axis group deceleration
	0x144(324)	Fault	An unconfigured axis group number has been used	Configure an axis group number for the used axis group in the "Axis Group Settings" list on the host computer

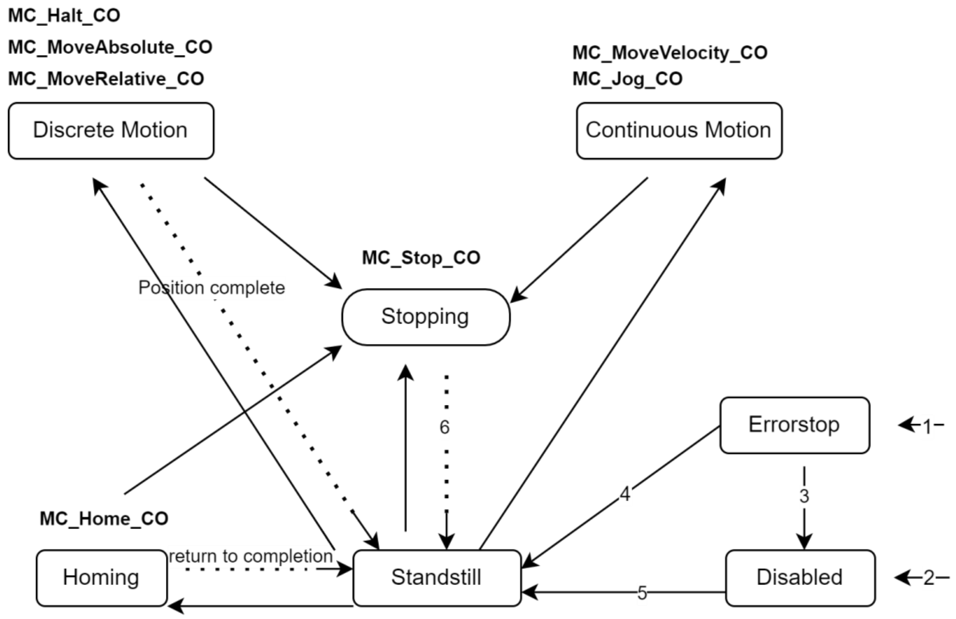
Main error code	Secondary error code	Error level	Possible cause	Solution
	0x145(325)	Fault	There is a pulse axis velocity exceeding 200kHz	There is a pulse axis velocity exceeding 200kHz
	0x146(326)	Fault	Two axis groups use the same axis, so that when one axis group is in motion state, the other axis group cannot enter the motion state.	Modify the reused axis, or run two axis groups at different times.

3.22 MC Axis Control (CANopen)

3.22.1 Command Table

Command	Name
MC_Power_CO	Communication control servo axis enable/disable
MC_Reset_CO	Communication control servo axis reset
MC_ReadStatus_CO	Read axis state by communication control
MC_ReadActualVelocity_CO	Read actual axis velocity by communication control
MC_ReadActualPosition_CO	Read actual axis position by communication control
MC_Halt_CO	Communication control servo axis halt
MC_Stop_CO	Communication control servo axis stop
MC_MoveVelocity_CO	Velocity operation mode of communication control axis
MC_MoveRelative_CO	Relative positioning of communication control axis
MC_MoveAbsolute_CO	Absolute positioning of communication control axis
MC_Home_CO	Communication control axis home
MC_Jog_CO	Communication control axis jog
MC_ReadAcceleration_CO	Read axis acceleration by communication control
MC_ReadDeceleration_CO	Read axis deceleration by communication control
MC_ReadDlStatus_CO	Read axis DI output state by communication control
CO_ReadSDO	Read SDO by communication control
CO_WriteSDO	Write SDO by communication control

3.22.2 Axis State Machines



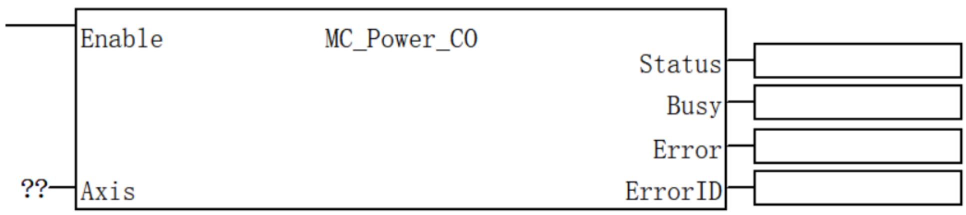
State Machine Description

State	Function Description
Disabled	Disable
ErrorStop	Stop due to fault
Standstill	Enabled state
Homing	Home
Stopping	Stop
Discrete Motion	Discretely move
Continuous Motion	Continuously move
Disabled	Disable

Conversion	Conversion Condition
1	When the fault detection logic of the axis detects a fault
2	When there is no fault with the axis and the energy flow of MC_Power is OFF
3	When MC_Reset is called to reset axis failure and MC_Power energy flow is OFF
4	When MC_Reset is called to reset axis failure and MC_Power energy flow is ON
5	When the energy flow of MC_Power is ON and the output flag Status is ON
6	When MC_Stop.Done is ON and the energy flow of the graphic block is OFF

3.22.3 MC_Power_CO

Graphic Block



16-Bit command	MC_Power_CO: Axis Enable/Disable					
32-Bit command	-					
Operand	**Name**	**Description**	**Nullable**	**Default value**	**Supported element**	**Data Type**
S1	Axis	Axis name/axis ID	No	-	Const	WORD
D1	Status	Axis status	Yes	OFF	Y, M, S	BOOL
D2	Busy	Busy flag	Yes	OFF	Y, M, S	BOOL
D3	Error	Command error flag	Yes	OFF	Y, M, S	BOOL
D4	ErrorID	Error code	Yes	0	D, R	WORD

Function Description

Axis ID: specifies the number of the axis to be controlled; range: 1-30.

Axis status: the actual state output of the axis, where ON indicates that the axis is enabled, while OFF indicates that the axis is disabled.

Error code: Refer to "Command Error Code Definitions".

The MC_Power_CO command writes the corresponding control word (6040h) according to the read status word (6041h) to enable the axis.

The writing correspondence between the status word (6041h) and the control word (6040h) is shown in the table below, where x represents any value (for status word) or remains unchanged (for control word):

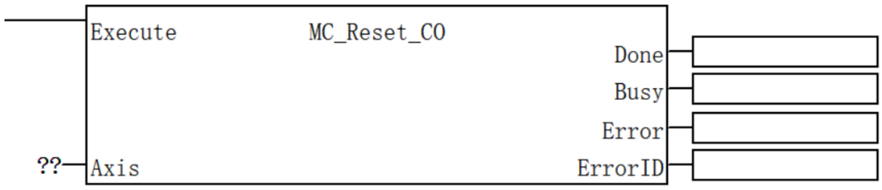
<table>
<tr><th>Energy flow state</th><th colspan="2">State word (6041h)</th><th colspan="2">Control word (6040h)</th></tr>
<tr><td rowspan="7">ON</td><td>Not ready to switch on</td><td>xxxx xxxx x0xx 0000$_b$</td><td rowspan="2">Shutdown</td><td rowspan="2">0000 0000 0000 0110$_b$</td></tr>
<tr><td>Switch on disabled</td><td>xxxx xxxx x1xx 0000$_b$</td></tr>
<tr><td>Ready to switch on</td><td>xxxx xxxx x01x 0001$_b$</td><td>Switch on</td><td>0000 0000 0000 0111$_b$</td></tr>
<tr><td>Switched on</td><td>xxxx xxxx x01x 0011$_b$</td><td>Switch on + enable operation</td><td>0000 0000 0000 1111$_b$</td></tr>
<tr><td>Fault reaction active</td><td>xxxx xxxx x0xx 1111$_b$</td><td rowspan="2">-</td><td rowspan="2">xxxx xx00 xx00 xxxx$_b$</td></tr>
<tr><td>Fault</td><td>xxxx xxxx x0xx 1000$_b$</td></tr>
<tr><td colspan="2">Others</td><td>-</td><td>xxxx xxxx xxxx xxxx$_b$</td></tr>
<tr><td rowspan="4">OFF</td><td>Ready to switch on</td><td>xxxx xxxx x01x 0001$_b$</td><td rowspan="3">Disable voltage</td><td rowspan="3">0000 0000 0000 0000$_b$</td></tr>
<tr><td>Switched on</td><td>xxxx xxxx x01x 0011$_b$</td></tr>
<tr><td>Operation enabled</td><td>xxxx xxxx x01x 0111$_b$</td></tr>
<tr><td colspan="2">Others</td><td>-</td><td>xxxx xx00 xx00 xxxx$_b$</td></tr>
</table>

Precautions

When called, up to 512 copies of this commands are supported.

3.22.4 MC_Reset_CO

Graphic Block



16-Bit command	MC_Reset_CO: Communication Control Servo Axis Reset					
32-Bit command	-					
Operand	**Name**	**Description**	**Nullable**	**Default value**	**Supported element**	**Data Type**
S1	Axis	Axis name/axis ID	No	-	Const	WORD
D1	Done	Completion	Yes	OFF	Y, M, S	BOOL
D2	Busy	Busy flag	Yes	OFF	Y, M, S	BOOL
D3	Error	Command error flag	Yes	OFF	Y, M, S	BOOL
D4	ErrorID	Error code	Yes	0	D, R	WORD

Function Description

This command is used to reset faults of the CANopen bus axis, causing the axis to enter the "Ready" or "Disabled" state.

Axis number: specifies the number of the axis to be controlled; range: 1-30.

Done: completes the reset operation and outputs the result.

Error code: Refer to "Command Error Code Definitions".

The MC_Reset_CO command writes the corresponding control word (6040h) according to the read status word (6041h) to reset the axis fault.

The writing correspondence between the status word (6041h) and the control word (6040h) is shown in the table below, where x represents any value (for status word) or remains unchanged (for control word):

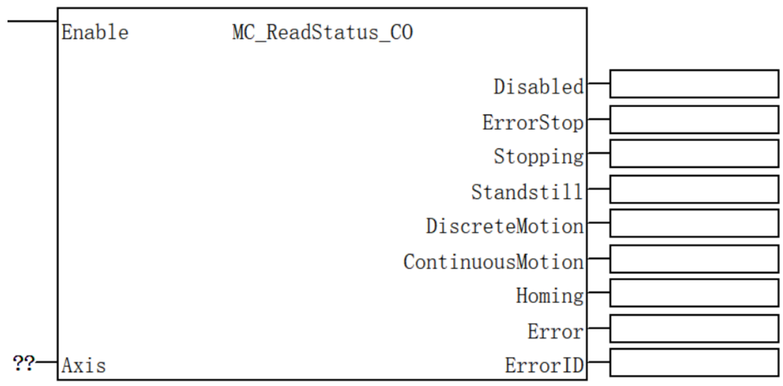
Energy flow state	State word (6041h)		Control word fault reset (6040h.bit7)
ON	Switch on disabled	xxxx xxxx x1xx 0000b	0
	Operation enabled	xxxx xxxx x01x 0111b	-
	Fault	xxxx xxxx x0xx 1000b	1
	-	Others	x
↑	-	xxxx xxxx xxxx xxxxb	0
OFF	-	xxxx xxxx xxxx xxxxb	x

Precautions

When called, up to 512 copies of this commands are supported.

3.22.5 MC_ReadStatus_CO

Graphic Block



16-Bit command	MC_ReadStatus_CO: Read Axis Status					
32-Bit command	-					
Operand	**Name**	**Description**	**Nullable**	**Default value**	**Supported element**	**Data Type**
S1	Axis	Axis name/axis ID	No	-	Const	WORD
D1	Disabled	Disabled flag	Yes	OFF	Y, M, S	BOOL
D2	ErrorStop	Fault message	Yes	OFF	Y, M, S	BOOL
D3	Stopping	Stop	Yes	OFF	Y, M, S	BOOL
D4	Standstill	Ready	Yes	OFF	Y, M, S	BOOL
D5	DiscreteMotion	Discretely move	Yes	OFF	Y, M, S	BOOL
D6	ContinuousMotion	Continuously move	Yes	OFF	Y, M, S	BOOL
D7	Homing	Home	Yes	OFF	Y, M, S	BOOL
D8	Error	Command error flag	Yes	OFF	Y, M, S	BOOL
D9	ErrorID	Error code	Yes	0	D, R	WORD

Function Description

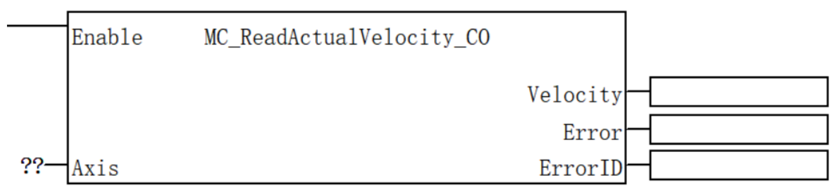
1. This command is used to read the states of the PLCOpen state machine, as well as the accelerating and decelerating states, of the axis, and is valid at high levels.
2. This command has no interrupt flag and therefor multiple commands can run simultaneously.

Precautions

When called, up to 2048 copies of this commands are supported.

3.22.6 MC_ReadActualVelocity_CO

Graphic Block



16-Bit command	MC_ReadActualVelocity_CO: Read Actual Axis Velocity					
32-Bit command	-					
Operand	**Name**	**Description**	**Nullable**	**Default value**	**Supported element**	**Data Type**
S1	Axis	Axis name/axis ID	No	-	Const	WORD
D1	Velocity	Current actual velocity	Yes	OFF	D, R	REAL
D2	Error	Command error flag	Yes	OFF	Y, M, S	BOOL
D3	ErrorID	Error code	Yes	0	D, R	WORD

Function Description

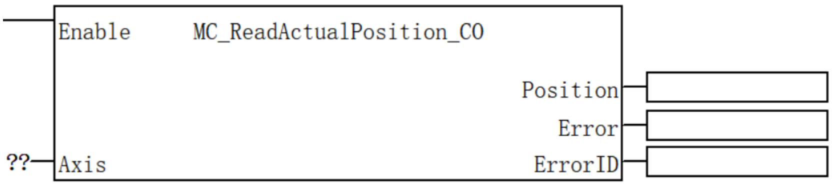
1. MC_ReadActualVelocity_CO command is used to read the actual running velocity of the axis, and is valid at high levels.
2. This command has no interrupt flag and therefor multiple commands can run simultaneously.

Precautions

When called, up to 512 copies of this commands are supported.

3.22.7 MC_ReadActualPosition_CO

Graphic Block



16-Bit command	MC_ReadActualPosition_CO: Read Actual Axis Position					
32-Bit command	-					
Operand	**Name**	**Description**	**Nullable**	**Default value**	**Supported element**	**Data Type**
S1	Axis	Axis name/axis ID	No	-	Const	WORD
D1	Position	Current actual position	Yes	OFF	D, R	REAL
D2	Error	Command error flag	Yes	OFF	Y, M, S	BOOL
D3	ErrorID	Error code	Yes	0	D, R	WORD

Function Description

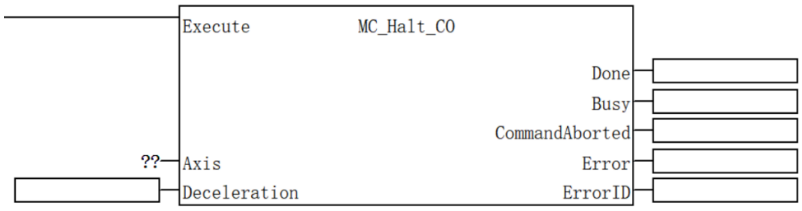
1. MC_ReadActualPosition_CO command is used to read the axis command position or axis feedback position, and is valid at high levels.
2. When the axis is a local pulse axis, the command output parameter Position is actually the command position.
3. This command has no interrupt flag and therefor multiple commands can run simultaneously.

Precautions

When called, up to 512 copies of this commands are supported.

3.22.8 MC_Halt_CO

Graphic Block



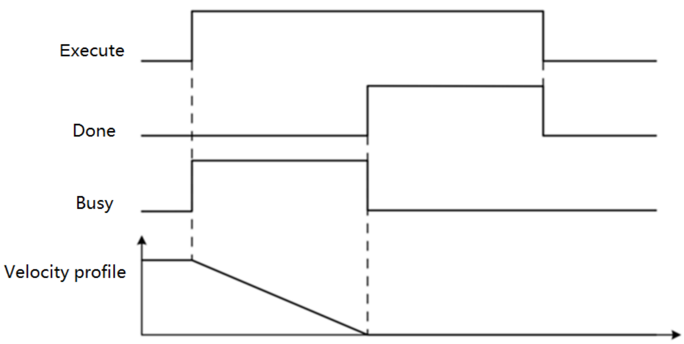
16-Bit command	MC_Halt_CO: Control Axis Halt					
32-Bit command	-					
Operand	**Name**	**Description**	**Nullable**	**Default value**	**Supported element**	**Data Type**
S1	Axis	Axis name/axis ID	No	-	Const	WORD
S2	Deceleration	Deceleration	Yes	OFF	Const, D, R	REAL
D1	Done	Completion	Yes	OFF	Y, M, S	BOOL
D2	Busy	Busy flag	Yes	OFF	Y, M, S	BOOL
D3	CommandAborted	Execution interrupt flag	Yes	OFF	Y, M, S	BOOL
D4	Error	Command error flag	Yes	OFF	Y, M, S	BOOL
D5	ErrorID	Error code	Yes	0	D, R	WORD

Function Description

1. This command is used to control the CANOpen bus axis to terminate the current motion and afterwards be able to respond to other commands that cause the axis to move.
2. The MC_Halt_CO command can be interrupted by the MC_MoveAbsolute_CO, MC_MoveRelative_CO, MC_MoveVelocity_CO, and MC_ Jog_CO commands.

Step	Action/Condition	Description
1	6040h.bit4 = 0	The control word triggers the motion to stop The target velocity is zeroed
	6040h.bit5 = 0	
	6040h.bit6 = 0	
	6040h.bit8 = 1	
	60FFh = 0	
2	606Ch = 0	Wait for the completion of stop
	6061h = 3 and 6041h.bit13 = 1	
	6061h != 3 and 6041h.bit10 = 1	
3	6060h = 1	Switch to the position mode

Timing diagram

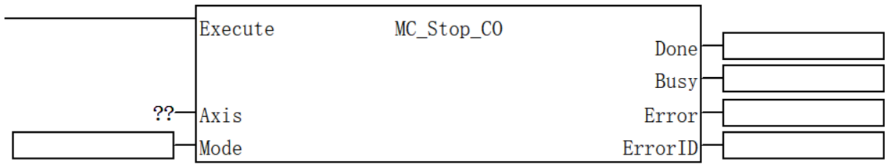


Precautions

When called, up to 512 copies of this commands are supported.

3.22.9 MC_Stop_CO

Graphic Block



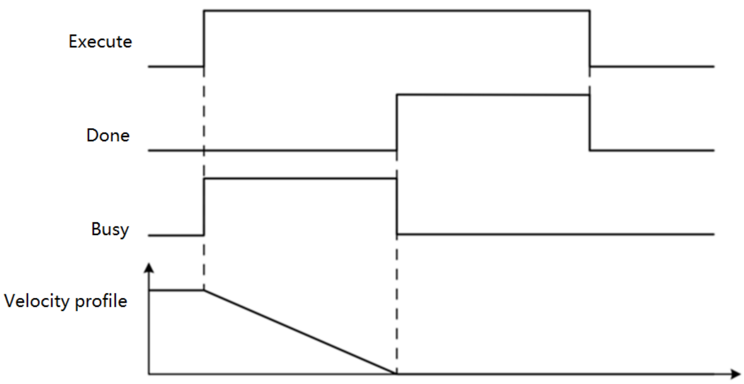
16-Bit command	MC_Stop_CO: Control Axis Stop					
32-Bit command	-					
Operand	**Name**	**Description**	**Nullable**	**Default value**	**Supported element**	**Data Type**
S1	Axis	Axis name/axis ID	No	-	Const	WORD
S2	Mode	Stop mode	Yes	OFF	Const, Y, M, S	BOOL
D1	Done	Completion	Yes	OFF	Y, M, S	BOOL
D2	Busy	Busy flag	Yes	OFF	Y, M, S	BOOL
D3	Error	Command error flag	Yes	OFF	Y, M, S	BOOL
D4	ErrorID	Error code	Yes	0	D, R	WORD

Function Description

This command is used to control the CANOpen bus axis to terminate the current motion, enter the "Stop" state, and no longer respond to any commands that cause the axis to move.

Step	Action/Condition	Description
1	6040h.bit4 = 0	The control word triggers the motion to stop The target velocity is zeroed
	6040h.bit5 = 0	
	6040h.bit6 = 0	
	6040h.bit8 = 1	
	60FFh = 0	
2	606Ch = 0	Wait for the completion of stop
	6061h = 3 and 6041h.bit13 = 1	
	6061h != 3 and 6041h.bit10 = 1	
3	6060h = 1	Switch to the position mode

Timing diagram

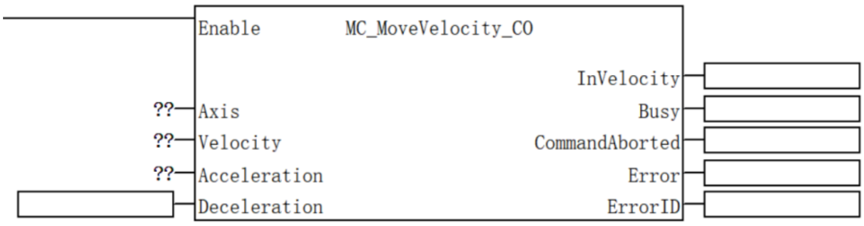


Precautions

When called, up to 512 copies of this commands are supported.

3.22.10 MC_MoveVelocity_CO

Graphic Block



16-Bit command	MC_MoveVelocity_CO: Control Axis Velocity Motion					
32-Bit command	-					
Operand	**Name**	**Description**	**Nullable**	**Default value**	**Supported element**	**Data Type**
S1	Axis	Axis name/axis ID	No	-	Const	WORD
S2	Velocity	Speed	No	OFF	Const, D, R	REAL
S3	Acceleration	Acceleration	No	OFF	Const, D, R	REAL
S4	Deceleration	Deceleration	Yes	OFF	Const, D, R	REAL
D1	InVelocity	Speed reached	Yes	OFF	Y, M, S	BOOL
D2	Busy	Busy flag	Yes	OFF	Y, M, S	BOOL
D3	CommandAborted	Execution interrupt flag	Yes	OFF	Y, M, S	BOOL
D4	Error	Command error flag	Yes	OFF	Y, M, S	BOOL
D5	ErrorID	Error code	Yes	0	D, R	WORD

Function Description

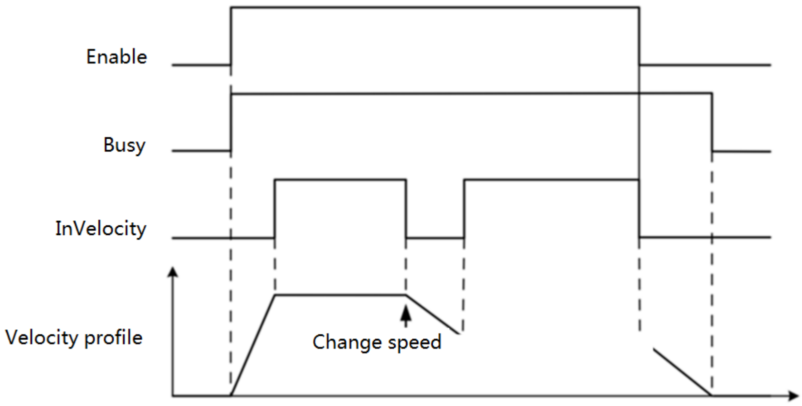
1. This command controls the CANOpen bus axis to specify the velocity motion.
2. When the specified velocity (Velocity) is greater than 0, the axis moves forward; when it is less than 0, the axis moves backward.
3. This command supports modifying speed parameters during runtime and allows them to take effect in real time. If the deceleration (Deceleration) is not specified (that is, the deceleration parameter is empty), it is equal to the specified acceleration by default.

Precautions

When called, up to 512 copies of this commands are supported.

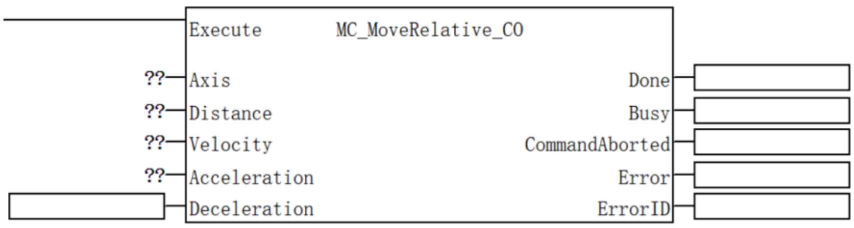
Step	Action/Condition	Description
1	6040h.bit8 = 0 Reset the Halt bit of the control word	Reset the Halt bit of the control word
2	6083h = acceleration	Write the acceleration
3	6084h = deceleration	Write the deceleration
4	6060h = 3	Switch to the velocity mode
5	6061h = 3	Wait for the completion of velocity mode switching
6	60FFh = target velocity	Set target velocity
	6041h.bit10 = 1	Reach target velocity
	60FFh < 0 and 6041h.bit11 = 1 and 60FDh.bit0 = 1: 60FFh = 0	When the negative motion encounters the negative limit, the motion ends
	607Ah > 0 and 6041h.bit11 = 1 and 60FDh.bit1 = 1: 60FFh = 0	When the positive motion encounters the positive limit, the motion ends
	60FFh = 0	When the energy flow of the command is invalid, the motion ends

Timing diagram



3.22.11 MC_MoveRelative_CO

Graphic Block



16-Bit command	MC_MoveRelative_CO: Relative Positioning of Control Axis					
32-Bit command	-					
Operand	**Name**	**Description**	**Nullable**	**Default value**	**Supported element**	**Data Type**
S1	Axis	Axis name/axis ID	No	-	Const	WORD
S2	Distance	Target distance	No	OFF	Const, D, R	REAL
S3	Velocity	Maximum velocity	No	OFF	Const, D, R	REAL
S4	Acceleration	Acceleration	No	OFF	Const, D, R	REAL
S5	Deceleration	Deceleration	Yes	OFF	Const, D, R	REAL
D1	Done	Completion	Yes	OFF	Y, M, S	BOOL
D2	Busy	Busy flag	Yes	OFF	Y, M, S	BOOL
D3	CommandAborted	Execution interrupt flag	Yes	OFF	Y, M, S	BOOL
D4	Error	Command error flag	Yes	OFF	Y, M, S	BOOL
D5	ErrorID	Error code	Yes	0	D, R	WORD

Function Description

This command achieves the relative positioning function of the CANopen bus axis, and controls the axis to move for a specified distance from the current position. If the deceleration (Deceleration) is not specified (that is, the deceleration parameter is empty), it is equal to the specified acceleration by default.

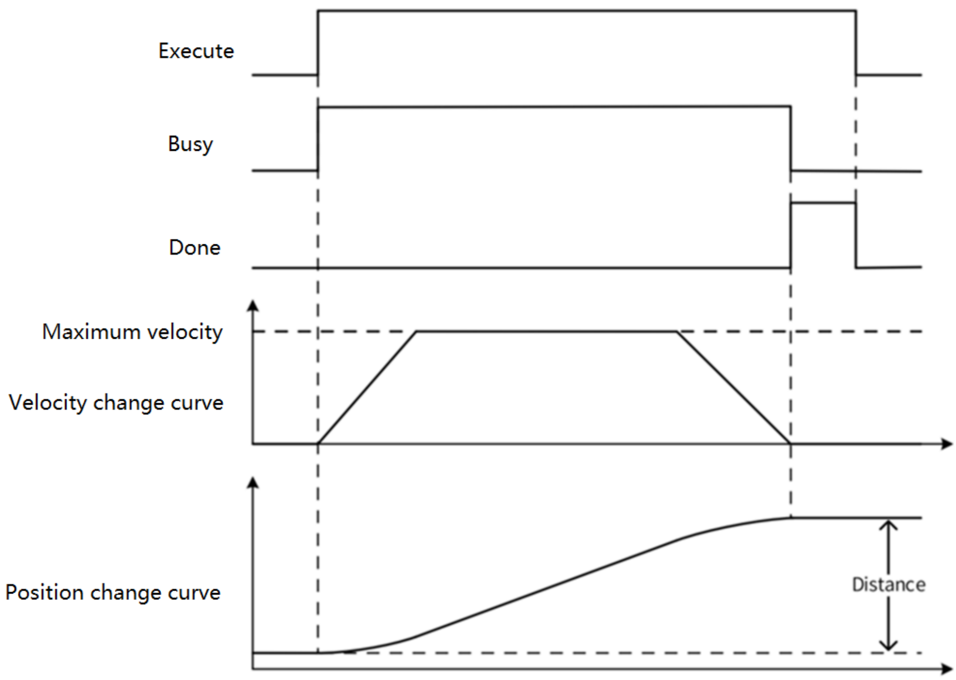
Precautions

When called, up to 512 copies of this commands are supported.

Step	Action/Condition	Description
1	6060h = 1	Switch to the position mode
2	6061h = 1	Wait for the completion of position mode switching

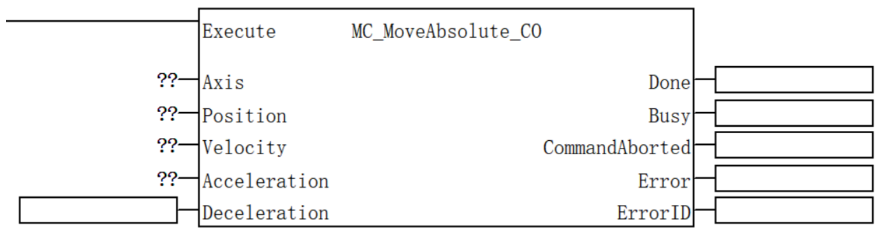
Step	Action/Condition	Description
3	6040h.bit5 = m 6040h.bit6 = 1 6040h.bit8 = 0 6040h.bit9 = 0	The control word writes the corresponding mode. In case of cache mode (parameter number: 1000) = 0, m =1; otherwise, m = 0.
4	607Ah = position 6081h = velocity	Write the (absolute) target position and positioning velocity
5	6083h = acceleration	Write the acceleration
6	6084h = deceleration	Write the deceleration
7	6040h.bit4 = 1	Trigger positioning
8	6041h.bit12 = 1	Wait for the start of positioning
9	6040h.bit4 = 0	Reset the positioning trigger
10	607Ah < 6064h and 6041h.bit11 = 1 and 60FDh.bit0 = 1	When the negative motion encounters the negative limit, positioning ends
	607Ah > 6064h and 6041h.bit11 = 1 and 60FDh.bit1 = 1	When the positive motion encounters the positive limit, positioning ends
	6041h.bit10 = 1 and 6041h.bit12 = 0	When the target position is reached, positioning is completed

Timing diagram



3.22.12 MC_MoveAbsolute_CO

Graphic Block



16-Bit command	MC_MoveAbsolute_CO: Absolute Positioning of Control Axis					
32-Bit command	-					
Operand	Name	Description	Nullable	Default value	Supported element	Data Type
S1	Axis	Axis name/axis ID	No	-	Const	WORD
S2	Position	Target position	No	OFF	Const, D, R	REAL
S3	Velocity	Maximum velocity	No	OFF	Const, D, R	REAL
S4	Acceleration	Acceleration	No	OFF	Const, D, R	REAL
S5	Deceleration	Deceleration	Yes	OFF	Const, D, R	REAL
D1	Done	Completion	Yes	OFF	Y, M, S	BOOL
D2	Busy	Busy flag	Yes	OFF	Y, M, S	BOOL
D3	CommandAborted	Execution interrupt flag	Yes	OFF	Y, M, S	BOOL
D4	Error	Command error flag	Yes	OFF	Y, M, S	BOOL
D5	ErrorID	Error code	Yes	0	D, R	WORD

Function Description

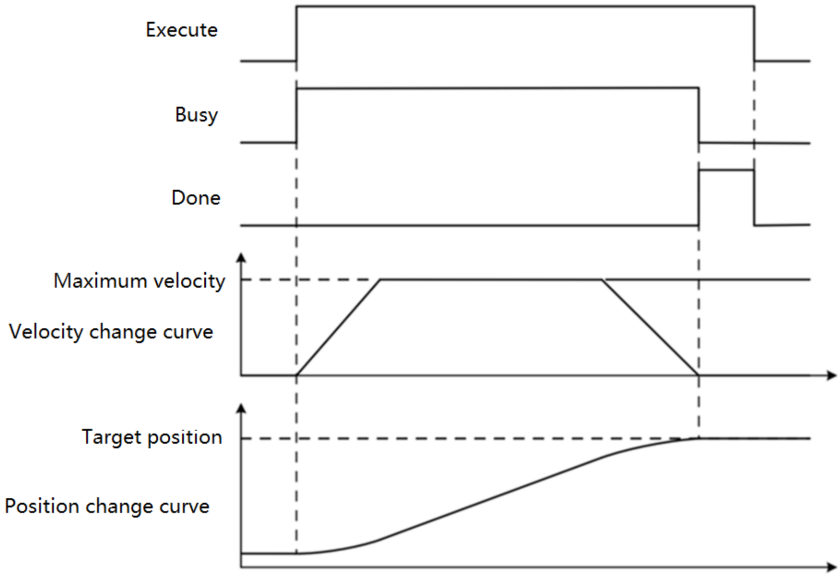
This command achieves the absolute positioning function of the CANopen bus axis, and controls the axis to move to the specified position. If the deceleration (Deceleration) is not specified (that is, the deceleration parameter is empty), it is equal to the specified acceleration by default.

Precautions

When called, up to 512 copies of this commands are supported.

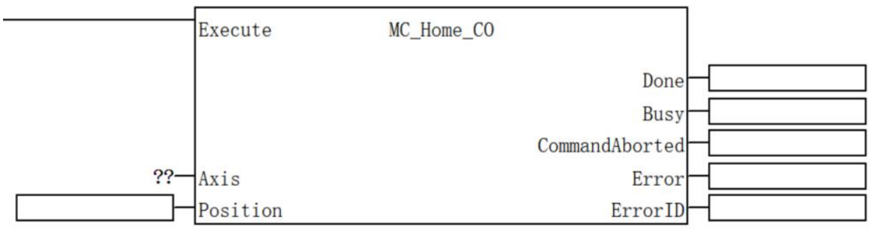
Step	Action/Condition	Description
1	6060h = 1	Switch to the position mode
2	6061h = 1	Wait for the completion of position mode switching
3	6040h.bit5 = m	The control word writes the corresponding mode In case of cache mode (parameter number: 1000) = 0, m =1; otherwise, m = 0.
	6040h.bit6 = 0	
	6040h.bit8 = 0	
	6040h.bit9 = 0	
4	607Ah = position	Write the (absolute) target position and positioning velocity
	6081h = velocity	
5	6083h = acceleration	Write the acceleration
6	6084h = deceleration	Write the deceleration
7	6040h.bit4 = 1	Trigger positioning
8	6041h.bit12 = 1	Wait for the start of positioning
9	6040h.bit4 = 0	Reset the positioning trigger
10	607Ah < 6064h and 6041h.bit11 = 1 and 60FDh.bit0 = 1	When the negative motion encounters the negative limit, positioning ends
	607Ah > 6064h and 6041h.bit11 = 1 and 60FDh.bit1 = 1	When the positive motion encounters the positive limit, positioning ends
	6041h.bit10 = 1 and 6041h.bit12 = 0	When the target position is reached, positioning is completed

Timing diagram



3.22.13 MC_Home_CO

Graphic Block



16-Bit command	MC_Home_CO: Communication Control Axis Home					
32-Bit command	-					
Operand	**Name**	**Description**	**Nullable**	**Default value**	**Supported element**	**Data Type**
S1	Axis	Axis name/axis ID	No	-	Const	WORD
S2	Position	Target position after homing	Yes	OFF	Const, D, R	REAL
D1	Done	Completion	Yes	OFF	Y, M, S	BOOL
D2	Busy	Busy flag	Yes	OFF	Y, M, S	BOOL
D3	CommandAborted	Execution interrupt flag	Yes	OFF	Y, M, S	BOOL
D4	Error	Command error flag	Yes	OFF	Y, M, S	BOOL
D5	ErrorID	Error code	Yes	0	D, R	WORD

Function Description

This command is used to achieve the homing of the CANopen bus axis. The homing mode and velocity should be set in the CANopen configuration interface. For various homing modes, see the manuals related to the servo/motor drivers.

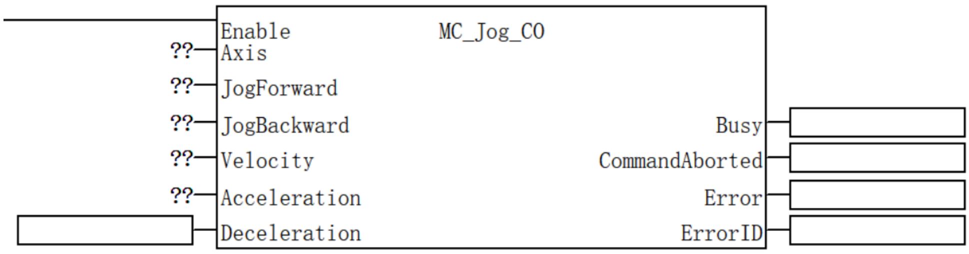
Precautions

When called, up to 512 copies of this commands are supported.

Step	Action/Condition	Description
1	6060h = 6	The control word triggers the motion to stop, and the target velocity is zeroed
2	6061h = 6	Wait for the completion of home mode switching
3	607Ch = origin offset	Set the origin offset
4	6040h.bit4 = 1	Start homing
5	6041h.bit10 = 1 and 6041h.bit13 = 1	Homing failed
	6041h.bit10 = 1 and 6041h.bit12 = 1	Homing done

3.22.14 MC_Jog_CO

Graphic Block



16-Bit command	MC_Jog_CO: Control Axis Jog					
32-Bit command	-					
Operand	Name	Description	Nullable	Default value	Supported element	Data Type
S1	Axis	Axis name/axis ID	No	-	Const	WORD
S2	JogForward	Positive motion, where the rising edge is valid	No	OFF	Y, M, S	BOOL
S3	JogBackward	Positive motion, where the rising edge is valid	No	OFF	Y, M, S	BOOL
S4	Velocity	Target velocity	No	OFF	Const, D, R	REAL
S5	Acceleration	Acceleration	No	OFF	Const, D, R	REAL
S6	Deceleration	Deceleration	Yes	OFF	Const, D, R	REAL
D1	Busy	Busy flag	Yes	OFF	Y, M, S	BOOL
D2	CommandAborted	Execution interrupt flag	Yes	OFF	Y, M, S	BOOL
D3	Error	Command error flag	Yes	OFF	Y, M, S	BOOL
D4	ErrorID	Error code	Yes	0	D, R	WORD

Function Description

This command is used to achieve the jog function of the CANopen bus axis. When JogForward is valid, the axis moves forward at the velocity specified by Velocity; when JogBackward is valid, the axis moves backward at the velocity specified by Velocity. If JogForward and JogBackward are valid at the same time, the axis stops moving.

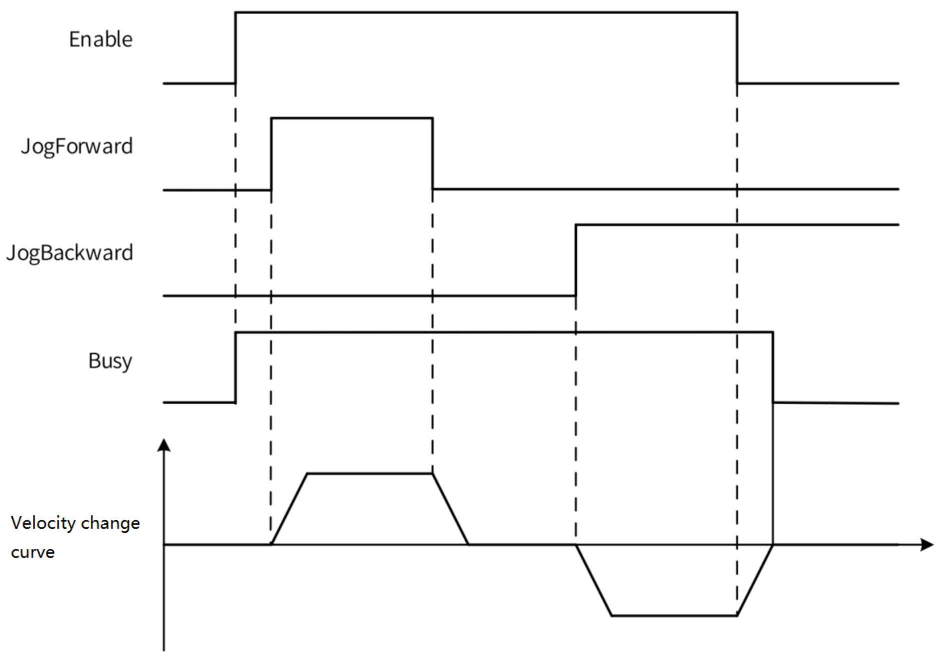
Precautions

When called, up to 512 copies of this commands are supported.

Step	Action/Condition	Description
1	6040h.bit8 = 0	Reset the Halt bit of the control word
2	6083h = acceleration/deceleration	Write the acceleration

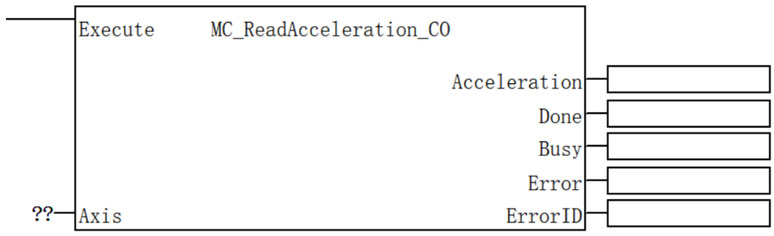
Step	Action/Condition	Description
3	6084h = acceleration/deceleration	Write the deceleration
4	6060h = 3	Switch to the velocity mode
5	6061h = 3	Wait for the completion of velocity mode switching
6	Forward jog: 60FFh = target velocity Backward jog: 60FFh = - target velocity Others: 60FFh = 0	Forward jog and backword jog
	60FFh < 0 and 6041h.bit11 = 1 and 60FDh.bit0 = 1: 60FFh = 0	When the negative motion encounters the negative limit, jog ends
	607Ah > 6040h and 6041h.bit11 = 1 and 60FDh. bit1 = 1: 60FFh = 0	When the positive motion encounters the positive limit, jog ends
	60FFh = 0	When the energy flow of the command is invalid, jog ends

Timing diagram



3.22.15 MC_ReadAcceleration_CO

Graphic Block



16-Bit command	MC_ReadAcceleration_CO: Read Axis Acceleration					
32-Bit command	-					
Operand	**Name**	**Description**	**Nullable**	**Default value**	**Supported element**	**Data Type**
S1	Axis	Axis name/axis ID	No	-	Const	WORD
D1	Acceleration	Present ACC speed	Yes	OFF	D, R	REAL
D2	Done	Completion	Yes	OFF	Y, M, S	BOOL
D3	Busy	Busy flag	Yes	OFF	Y, M, S	BOOL
D4	Error	Command error flag	Yes	OFF	Y, M, S	BOOL
D5	ErrorID	Error code	Yes	0	D, R	WORD

Function Description

This command is used to read the current acceleration of the CANopen bus axis.

Axis number: specifies the number of the axis to be read; range: 1-32.

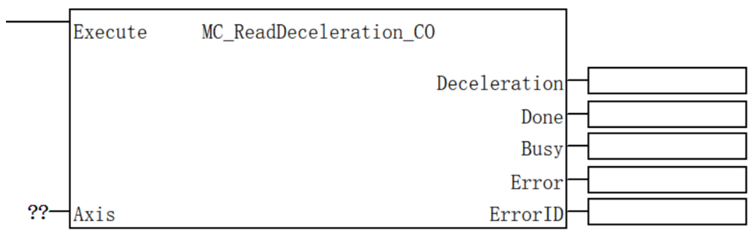
Acceleration: The current acceleration of the axis, which is a 32-bit floating-point number.

Precautions

When called, up to 512 copies of this commands are supported.

3.22.16 MC_ReadDeceleration_CO

Graphic Block



16-Bit command	MC_ReadDeceleration_CO: Read Axis Deceleration					
32-Bit command	-					
Operand	**Name**	**Description**	**Nullable**	**Default value**	**Supported element**	**Data Type**
S1	Axis	Axis name/axis ID	No	-	Const	WORD
D1	Deceleration	Current deceleration	Yes	OFF	D, R	REAL
D2	Done	Completion	Yes	OFF	Y, M, S	BOOL
D3	Busy	Busy flag	Yes	OFF	Y, M, S	BOOL
D4	Error	Command fault flag	Yes	OFF	Y, M, S	BOOL
D5	ErrorID	Error code	Yes	0	D, R	WORD

Function Description

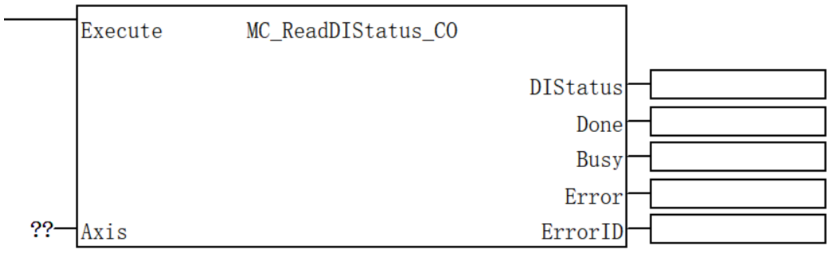
1. This command is used to read the current deceleration of the CANopen bus axis.
2. Axis: specifies the number of the axis to be read; range: 1-30.
3. Deceleration: The current deceleration of the axis, which is a 32-bit floating-point number.

Precautions

When called, up to 512 copies of this commands are supported.

3.22.17 MC_ReadDIStatus_CO

Graphic Block



16-Bit command	MC_ReadDIStatus_CO: Read Axis DI Output State					
32-Bit command	-					
Operand	**Name**	**Description**	**Nullable**	**Default value**	**Supported element**	**Data Type**
S1	Axis	Axis name/axis ID	No	-	Const	WORD
D1	DIStatus	Current DI output state	Yes	OFF	D, R	DWORD
D2	Done	Completion	Yes	OFF	Y, M, S	BOOL
D3	Busy	Busy flag	Yes	OFF	Y, M, S	BOOL
D4	Error	Command error flag	Yes	OFF	Y, M, S	BOOL
D5	ErrorID	Error code	Yes	0	D, R	WORD

Function Description

DI input state

[31:16]: manufacturer-defined; [15:3]: reserved; [1]: forward limit, where 0 means invalid and 1 means valid; [0]: backward limit, where 0 means invalid and 1 means valid.

Precautions

When called, up to 512 copies of this commands are supported.

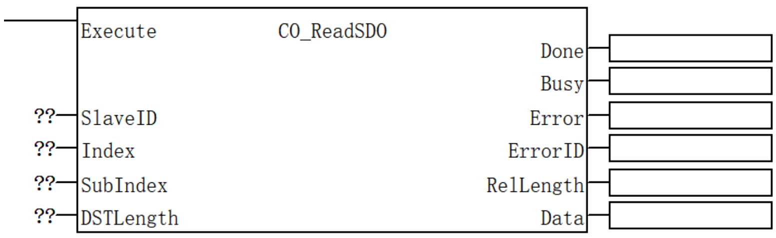
3.23 Communication (CANopen)

3.23.1 Command Table

Command	Name
ReadSDO_CO	CANopen read SDO
WriteSDO_CO	CANopen write SDO

3.23.2 ReadSDO_CO

Graphic Block



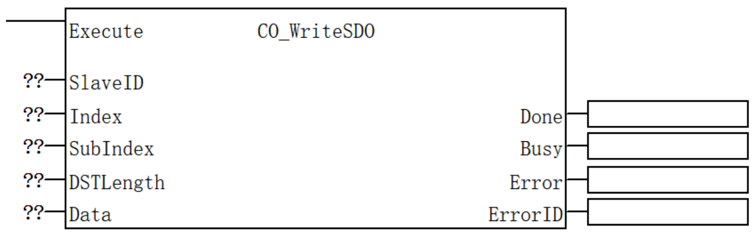
16-Bit command	ReadSDO_CO: Axis Enable/Disable					
32-Bit command	-					
Operand	**Name**	**Description**	**Nullable**	**Default value**	**Supported element**	**Data Type**
S1	SlaveID	Slave ID	No	-	Const	WORD
S2	Index	Index	No	OFF	Const, D, R	WORD
S3	SubIndex	Sub-index	No	OFF	Const, D, R	WORD
S4	DSTLength	Target length	No	OFF	Const, D, R	WORD
D1	Done	Completion	Yes	OFF	Y, M, S	BOOL
D2	Busy	Busy flag	Yes	OFF	Y, M, S	BOOL
D3	Error	Error flag	Yes	OFF	Y, M, S	BOOL
D4	ErrorID	Error code	Yes	OFF	D, R	WORD
D5	RelLength	Real data length	Yes	OFF	D, R	IN
D6	Data	Read data	Yes	0	D, R	DINT

Function Description

This command is used to read SDO, and is valid to the rising edge.

3.23.3 WriteSDO_CO

Graphic Block



16-Bit command	WriteSDO_CO: Axis Enable/Disable					
32-Bit command	-					
Operand	**Name**	**Description**	**Nullable**	**Default value**	**Supported element**	**Data Type**
S1	SlaveID	Slave ID	No	-	Const	WORD
S2	Index	Index	No	OFF	Const, D, R	WORD
S3	SubIndex	Sub-index	No	OFF	Const, D, R	WORD
S4	DSTLength	Target length	No	OFF	Const, D, R	WORD
S5	Data	Write data	No	OFF	Const, D, R	DINT
D1	Done	Completion	Yes	OFF	Y, M, S	BOOL
D2	Busy	Busy flag	Yes	OFF	Y, M, S	BOOL
D3	Error	Command error flag	Yes	OFF	Y, M, S	BOOL
D4	ErrorID	Error code	Yes	0	D, R	WORD

Function Description

This command is used to write SDO, and is valid to the rising edge.

3.24 ENC Axis Control (Pulse Output)

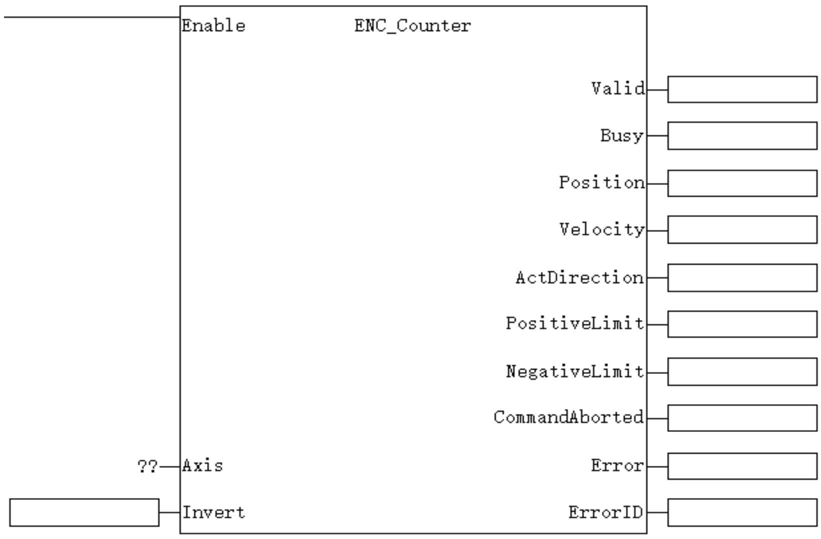
3.24.1 Command Table

Command Category	Name	Function
Local encoder axis	ENC_Counter	Encoder enable (high-speed counter)
	ENC_Reset	Encoder reset
	ENC_Preset	Encoder preset
	ENC_TouchProbe	Encoder probe
	ENC_Compare	Single-point comparison of encoder
	ENC_ArrayCompare	Unidimensional array comparison of encoder
	ENC_StepCompare	Unidimensional step size comparison of encoder
	ENC_ResetCompare	Encoder reset comparator
	ENC_SetUnit	Set axis gear ratio
	ENC_SetLineRotationMode	Set axis operation mode

3.24.2 ENC_Counter

This command (ENC_Counter) controls the counting enable (high-speed counter) of the encoder axis.

Graphic Block



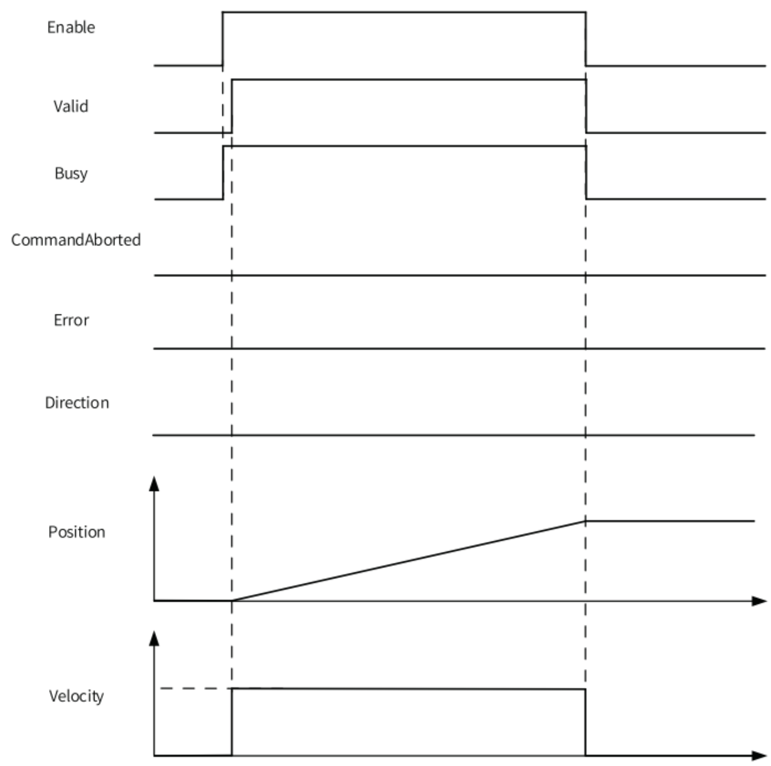
16-Bit command	-						
32-Bit command	ENC_Counter: Encoder Enable (High-Speed Counter)						
Operand	**Name**	**Description**	**Supported element**	**Nullable**	**Default value**	**Range**	**Data Type**
S1	Axis	Axis number	-	No	-	-	-
S2	Invert	Counter direction	Const; D; R; custom variable	Yes	0	0~1	INT
D1	Valid	Validity status	M; S; Y; custom variable	Yes	OFF	ON, OFF	BOOL
D2	Busy	Executing	M; S; Y; custom variable	Yes	OFF	ON, OFF	BOOL
D3	Position	Current position	D; R; custom variable	Yes	0	Negative, positive	REAL

16-Bit command	-						
32-Bit command	ENC_Counter: Encoder Enable (High-Speed Counter)						
Operand	**Name**	**Description**	**Supported element**	**Nullable**	**Default value**	**Range**	**Data Type**
D4	Velocity	Current velocity	D; R; custom variable	Yes	0	Negative, positive	REAL
D5	ActDirection	Counting direction	M; S; Y; custom variable	Yes	OFF	ON, OFF	BOOL
D6	PositiveLimit	Positive limit state in linear mode	M; S; Y; custom variable	Yes	OFF	ON, OFF	BOOL
D7	Negative Limit	Negative limit state in linear mode	M; S; Y; custom variable	Yes	OFF	ON, OFF	BOOL
D8	Command Aborted	Execution interrupt	M; S; Y; custom variable	Yes	OFF	ON, OFF	BOOL
D9	Error	Error state	M; S; Y; custom variable	Yes	OFF	ON, OFF	BOOL
D10	ErrorID	Error code	D; R; custom variable	Yes	0	0~65535	INT

Function Description

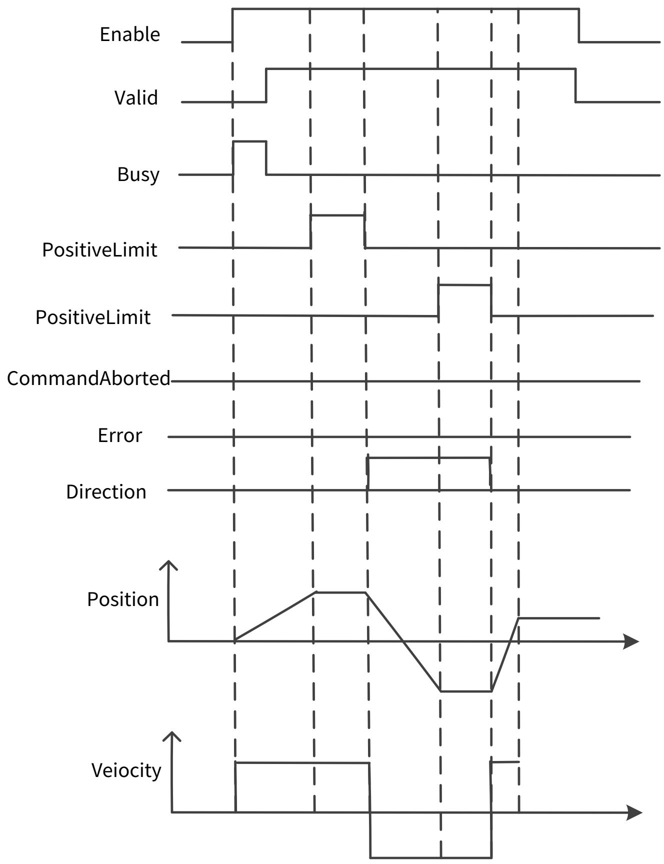
1. In case of the command input Enable=ON, Busy=ON and Valid=ON are set, and the encoder axis starts counting.
2. In case of the command input Enable=OFF, Busy=OFF and Valid=OFF are set, and the encoder axis stops counting.

Note: When Enable is on the rising edge, the current input parameters are valid; when Enable is in the constant ON state, it is invalid to modify the input parameters in the graphic block being executed.



In the linear mode, if the software limit is enabled, after the counting value reaches the limit value, the counter stops counting, and the limit signal output is valid; after the pulse input inverses, the limit signal resets, and the counter performs inverse counting.

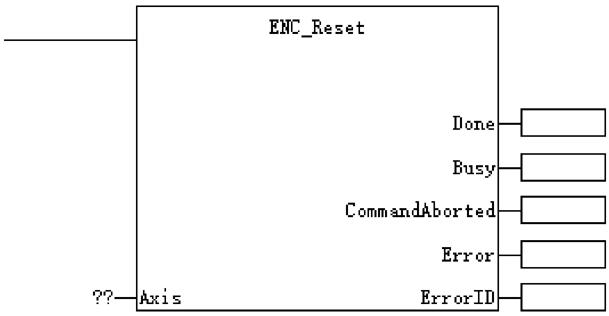
Timing diagram



3.24.3 ENC_Reset

This command is used to reset faults of the bus encoder axis.
ENC_Reset - encoder reset.

Graphic Block



16-Bit command	ENC_Reset: Encoder Reset						
32-Bit command	-						
Operand	**Name**	**Description**	**Supported element**	**Nullable**	**Default value**	**Range**	**Data Type**
S1	Axis	Axis number	-	No	-	-	-
D1	Done	Completion flag	M; S; Y; custom variable	Yes	OFF	ON, OFF	BOOL

16-Bit command	ENC_Reset: Encoder Reset						
32-Bit command	-						
Operand	Name	Description	Supported element	Nullable	Default value	Range	Data Type
D2	Busy	Executing	M; S; Y; custom variable	Yes	OFF	ON, OFF	BOOL
D3	Command Aborted	Execution interrupt	M; S; Y; custom variable	Yes	OFF	ON, OFF	BOOL
D4	Error	Error state	M; S; Y; custom variable	Yes	OFF	ON, OFF	BOOL
D5	ErrorID	Error code	Const; D; R; custom variable	Yes	0	0~65535	INT

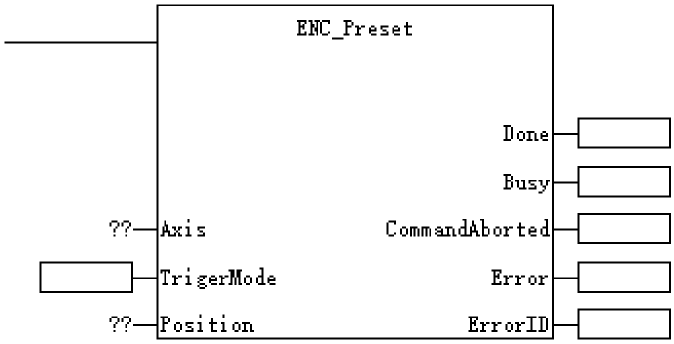
Function Description

If the command input Enable is on the rising edge, the corresponding counter continues to count after being reset once

3.24.4 ENC_Preset

Encoder Preset - encoder preset.

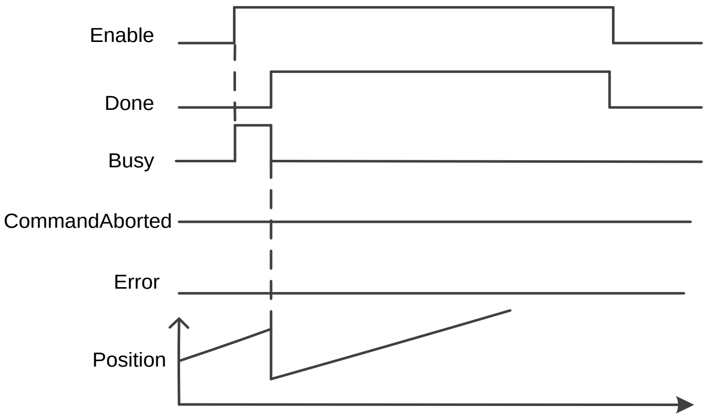
Graphic Block



16-Bit command	-						
32-Bit command	Encoder Preset: Encoder Preset						
Operand	Name	Description	Supported element	Nullable	Default value	Range	Data Type
S1	Axis	Axis number	-	No	-	-	-
D1	Done	Completion flag	M; S; Y	Yes	OFF	ON, OFF	BOOL
D2	Busy	Executing	M; S; Y	Yes	OFF	ON, OFF	BOOL
D3	CommandAborted	Execution interrupt	M; S; Y	Yes	OFF	ON, OFF	BOOL
D4	Error	Error state	M; S; Y	Yes	OFF	ON, OFF	BOOL
D5	ErrorID	Error code	Const; D; R	Yes	0	0~65535	INT

Function Description

1. In case of TrigerMode=0, Enable inputs a high level to complete the encoder position settings.

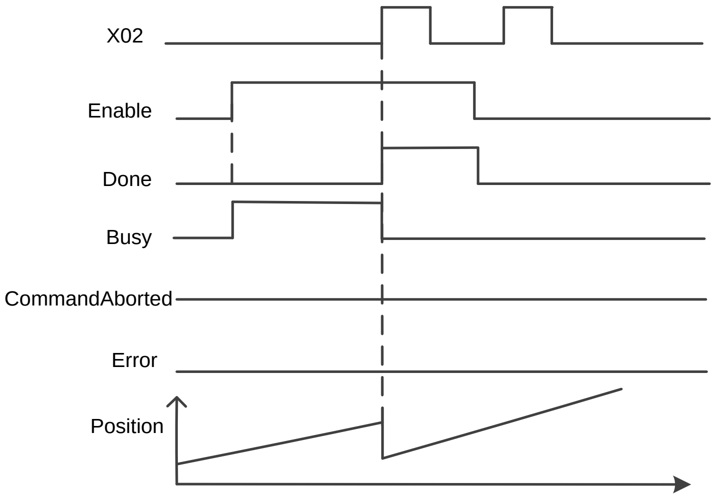


2. In case of TrigerMode=1~3, it is necessary to configure relevant parameters in the corresponding control axis and trigger IO externally.

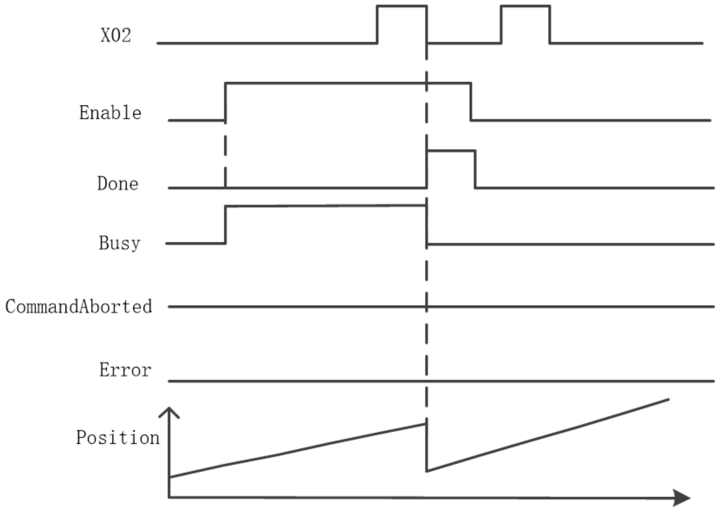
Note: When Enable is on the rising edge, the current input parameters are valid; when Enable is in the constant ON state, it is invalid to modify the input parameters in the graphic block being executed.

Timing diagram

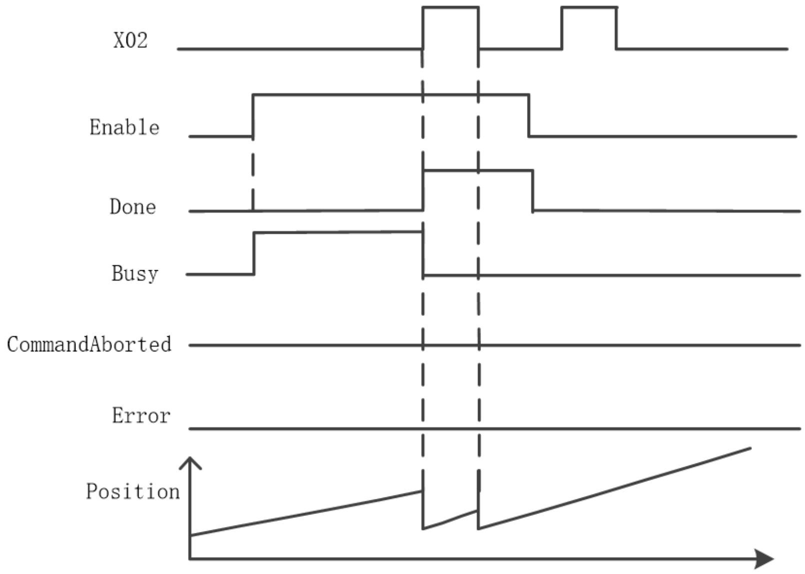
- The rising edge is valid with TrigerMode=1.



- The falling edge is valid with TrigerMode=2.



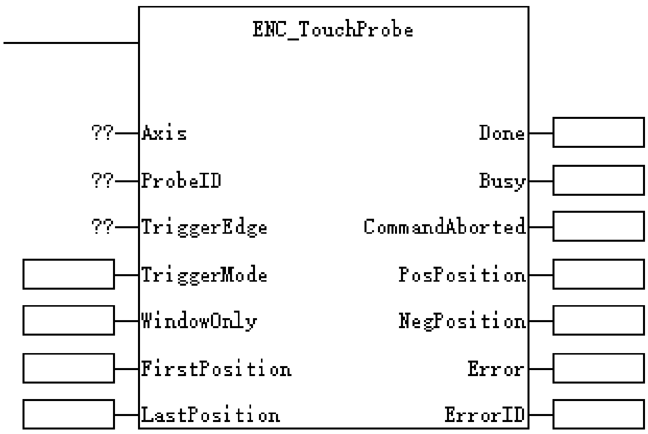
- Both rising and falling edges are valid with TrigerMode=3



3.24.5 ENC_TouchProbe

ENC_TouchProbe - encoder probe

Graphic Block



16-Bit command	-

32-Bit command	ENC_TouchProbe: Encoder Probe						
Operand	**Name**	**Description**	**Supported element**	**Nullable**	**Default value**	**Range**	**Data Type**
S1	Axis	Axis number	-	No	-	-	-
S2	ProbeID	Probe ID 0: probe 1 1: probe 2	Const; D; R; custom variable	No	-	0~1	INT
S3	Trigger Edge	Edge type 0: only falling edge triggered 1: only falling edge triggered 2: both rising and falling edges triggered	Const; D; R; custom variable	No	-	0~2	INT
S4	Trigger Mode	Trigger type 0: single trigger 1: continuous trigger	Const; D; R; custom variable	Yes	0	0~1	INT
S5	Window Only	Enable probe window 0: disables the probe window function 1: enables the probe window function	X; M; S; Y; custom variable	Yes	OFF	ON, OFF	BOOL
S6	First Position	Start position of probe window	Const; D; R; custom variable	Yes	0	Positive/ negative/0	REAL
S7	Last Position	End position of probe serial port	Const; D; R; custom variable	Yes		Not equal to FirstPositon	REAL
D1	Done	Completion flag	M; S; Y; custom variable	Yes	OFF	ON, OFF	BOOL
D2	Busy	Executing	M; S; Y; custom variable	Yes	OFF	ON, OFF	BOOL
D3	Command Aborted	Execution interrupt	M; S; Y	Yes	OFF	ON, OFF	BOOL
D4	PosPostion	Rising edge latch position	D; R	Yes	0	Positive/ negative/0	REAL
D5	NegPositon	Falling edge latch position	D; R	Yes	0	Positive/ negative/0	REAL
D6	Error	Error state	M; S; Y	Yes	OFF	ON, OFF	BOOL
D7	ErrorID	Error code	D; R	Yes	0	0~65535	INT

Function Description

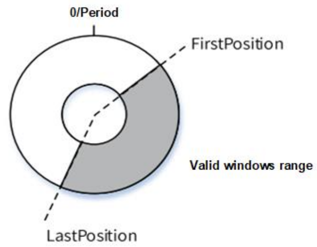
This command starts the probe function module and requires the relevant hardware configuration to be activated.

In case of Enable=ON, when the command detects that the probe input specified by ProbeID is valid and meets the probe detection conditions, the function block latches the current position of the axis. In case of WindowOnly=OFF, the window detection function is invalid. As long as the probe input signal is valid, the position of the axis during the validity period of the probe signal can be latched. In case of WindowOnly=ON, the window detection function is valid.

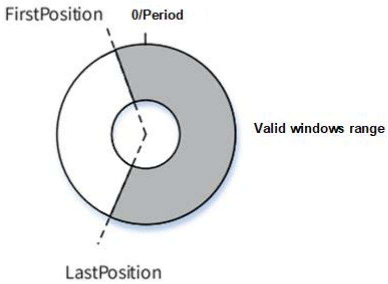
Note: When Enable is on the rising edge, the current input parameters are valid; when Enable is in the constant ON state, it is invalid to modify the input parameters in the graphic block being executed.

In the linear mode, this command detects the probe signal only when the current position of the axis is within the range specified by FirstPosition and LastPosition.

In the circular mode, in case of FirstPosition < LastPosition, the valid window range is shown in the figure below.



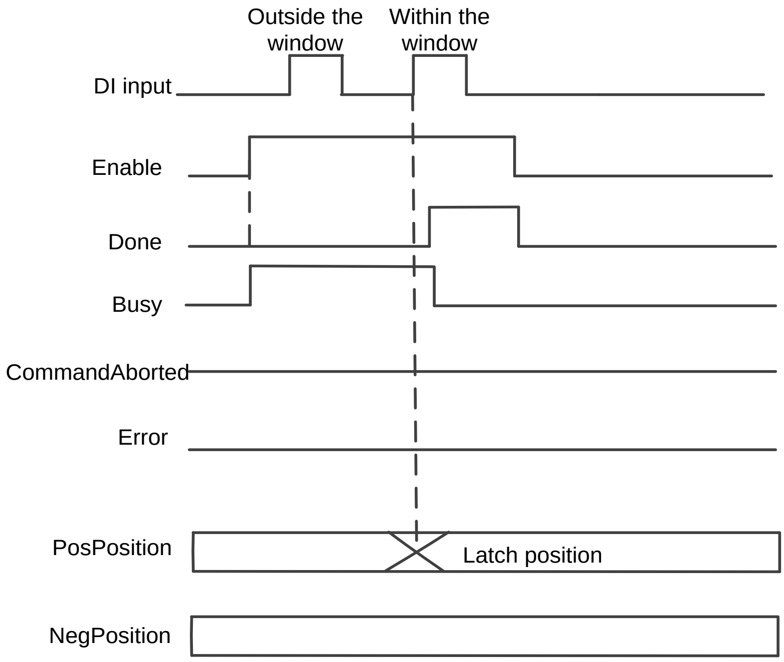
In case of FirstPosition > LastPosition, the valid window range is shown in the figure below.



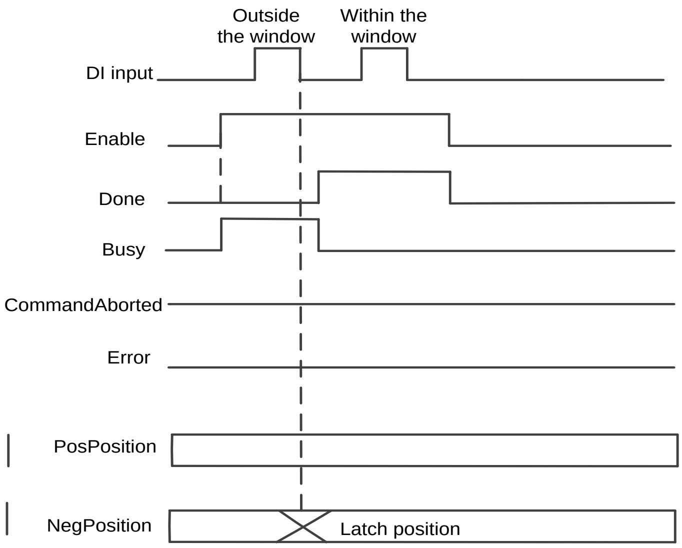
This command can the rising and falling edges of the probe signal separately or simultaneously. When detecting the rising (falling) edge only, the command writes the value detected on the rising (falling) edge to PosPosition (NegPosition); at this time, a detection cycle is completed to set the Done signal. If the rising and falling edges are detected simultaneously, after Enable of the command is valid, the command writes the position to the PosPosition/NegPosition as soon as it has detected the rising/falling edge; at this time, a complete detection cycle is done to output the Done signal, and there is no requirement for the input order of the rising and falling edges.

Timing diagram

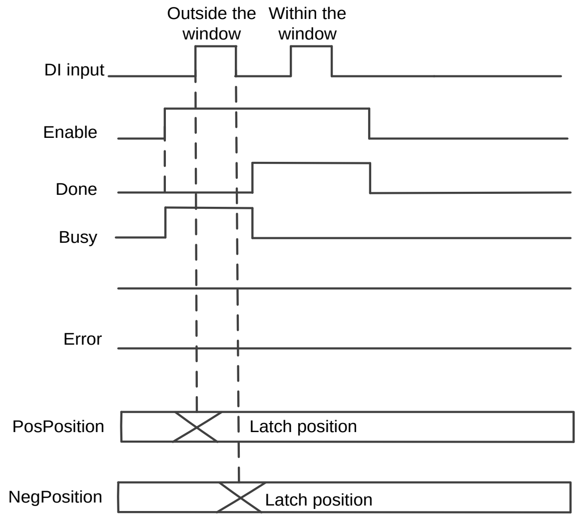
- For probe 1, the rising edge is valid, the single trigger mode is applied, and the window function is valid.



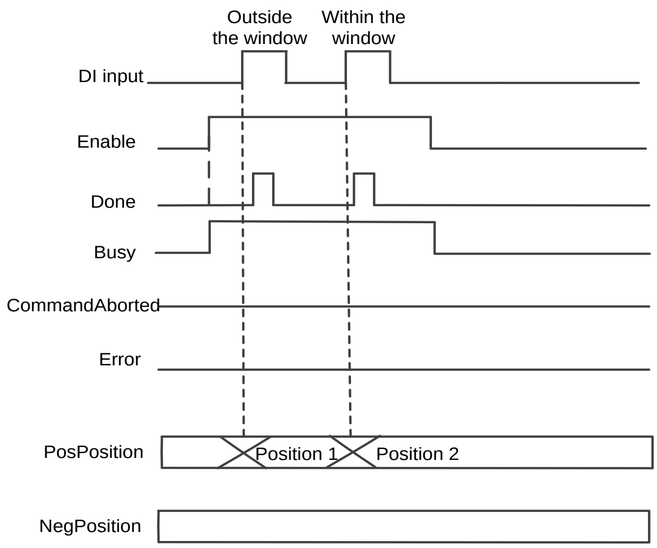
- For probe 1, the falling edge is valid, the DI terminal is triggered in the single trigger mode, and the window function is invalid.



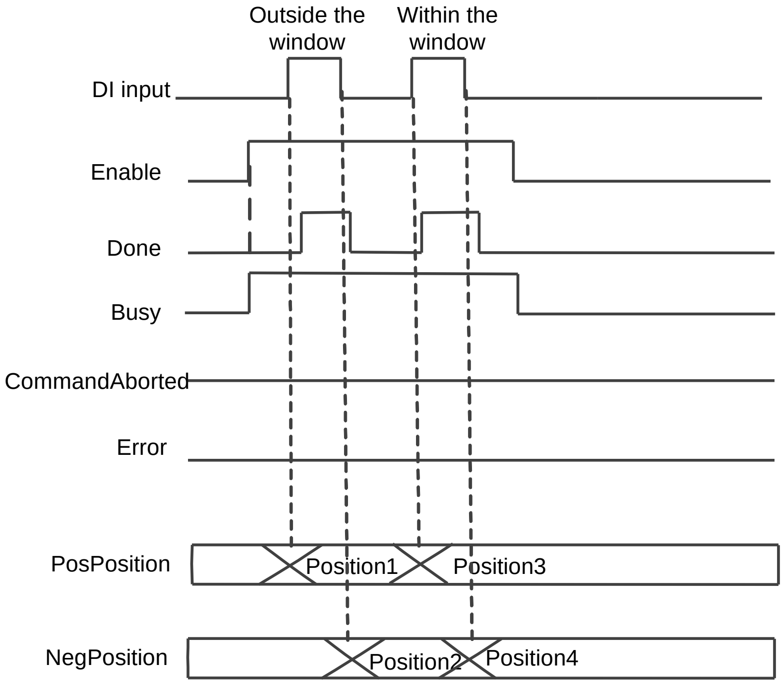
- For probe 1, both the rising and falling edges are valid, the DI terminal is triggered in the single trigger mode, and the window function is invalid.



- For probe 1, the rising edge is valid, the DI terminal is triggered in the continuous trigger mode, and the window function is invalid.



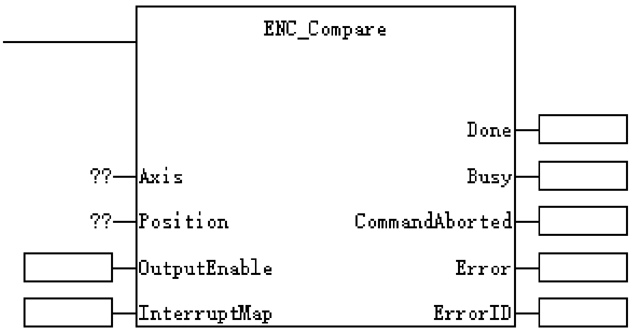
- For probe 1, both the rising and falling edges are valid, the DI terminal is triggered in the continuous trigger mode, Done generates a valid signal for one cycle after both the rising and falling edges of DI are valid, and the window function is invalid.



3.24.6 ENC_Compare

ENC_Compare - single-point comparison output

Graphic Block



16-Bit command	-						
32-Bit command	ENC_Compare: Continuous Execution						
Operand	**Name**	**Description**	**Supported element**	**Nullable**	**Default value**	**Range**	**Data Type**
S1	Axis	Axis number	-	No	-	-	-
S2	Position	Comparison position	Const; D; R; custom variable	No	-	-	REAL
S3	OutPutEnable	Hardware output enable, where ON: Enable; OFF: Disable	X; M; S; Y; custom variable	Yes	OFF	ON, OFF	BOOL
S4	InterruptMap	Interrupt number, where 0: does not associate comparison interrupt; 1-16:	Const; D; R; custom variable	Yes	0	0~16	INT

16-Bit command	-						
32-Bit command	ENC_Compare: Continuous Execution						
Operand	**Name**	**Description**	**Supported element**	**Nullable**	**Default value**	**Range**	**Data Type**
		associates comparison interrupt;					
D1	Done	Completion flag	M; S; Y; custom variable	Yes	OFF	ON, OFF	BOOL
D2	Busy	Executing	M; S; Y; custom variable	Yes	OFF	ON, OFF	BOOL
D3	Command Aborted	Execution interrupt	M; S; Y; custom variable	Yes	OFF	ON, OFF	BOOL
D4	Error	Error state	M; S; Y; custom variable	Yes	OFF	ON, OFF	BOOL
D5	ErrorID	Error code	D; R; custom variable	Yes	0	0~65535	INT

Function Description

In the "Compare Output Set" interface of the local pulse axis, check "Compare Output Enable", select the comparison output terminal, and choose whether to output pulses by time or by unit.

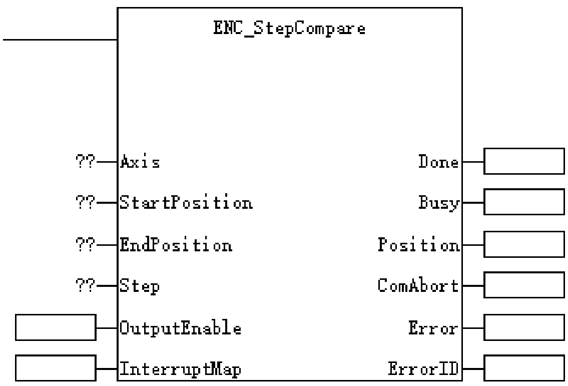
If OutputEnable is set to 1, the set comparison output terminal generates a comparison output signal; if it is set to 0, no comparison output signal is generated. InterruptMap is used to associate comparison interrupt subroutines. When it is set to 0, interrupt subroutines are not associated; when it is set to 1-16, interrupt subroutines are associated.

Note: When Enable is on the rising edge, the current input parameters are valid; when Enable is in the constant ON state, it is invalid to modify the input parameters in the graphic block being executed.

3.24.7 ENC_StepCompare

ENC_StepCompare - unidimensional step size comparison of encoder.

Graphic Block



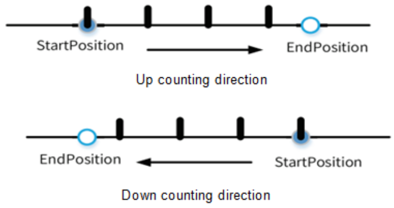
16-Bit command	-						
32-Bit command	ENC_StepCompare: Unidimensional Step Size Comparison of Encoder						
Operand	**Name**	**Description**	**Supported element**	**Nullable**	**Default value**	**Range**	**Data Type**
S1	Axis	Axis number	-	No	-	-	-
S2	StartPosition	Starting comparison set	Const; D; R	No	-	-	REAL
S3	EndPosition	Ending comparison set	Const; D; R	No	-	-	REAL
S4	Step	Step length	Const; D; R	No	-	-	REAL
S5	OutPutEnable	Hardware output enable, where ON: Enable; OFF: Disable	X; M; S; Y	Yes	OFF	ON, OFF	BOOL
S6	InterruptMap	Interrupt number, where 0: does not associate comparison interrupt 1-16: associates comparison interrupt	Const; D; R	Yes	0	0~16	INT
D1	Done	Completion flag	M; S; Y	Yes	OFF	ON, OFF	BOOL
D2	Busy	Executing	M; S; Y	Yes	OFF	ON, OFF	BOOL
D3	position	Output position	D; R	Yes	0	-	REAL
D4	Command Aborted	Execution interruption	M; S; Y	Yes	OFF	ON, OFF	BOOL
D5	Error	Error state	M; S; Y	Yes	OFF	ON, OFF	BOOL
D6	ErrorID	Error code	D; R	Yes	0	0~65535	INT

Function Description

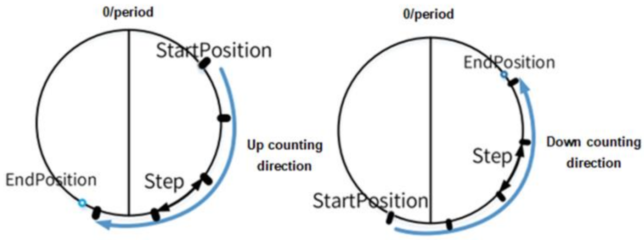
In the "Compare Output Set" interface of the local pulse axis, check " Compare Output Enable", select the comparison output terminal, and choose whether to output pulses by time or by unit.

If OutputEnable is set to 1, the set comparison output terminal generates a comparison output signal; if it is set to 0, no comparison output signal is generated. InterruptMap is used to associate comparison interrupt subroutines. When it is set to 0, interrupt subroutines are not associated; when it is set to 1-16, interrupt subroutines are associated.

Comparison point settings: In this command, StartPosition is used to set the start position of the comparison point, and EndPosition is used to set the end position of the comparison point. In the linear mode, the comparison point settings follow the rules below: when the value of StartPosition is less than the value of EndPosition, Step should be set to a positive number, which represents the additive count comparison method; when the value of StartPosition is greater than the value of EndPosition, Step should be set to a negative number, which represents the subtractive count comparison method.



In the circular mode, the comparison point settings follow the rules below: when the value of StartPosition is less than the value of EndPosition, Step should be set to a positive number, which represents the additive count comparison method; when the value of StartPosition is greater than the value of EndPosition, Step should be set to a negative number, which represents the subtractive count comparison method.

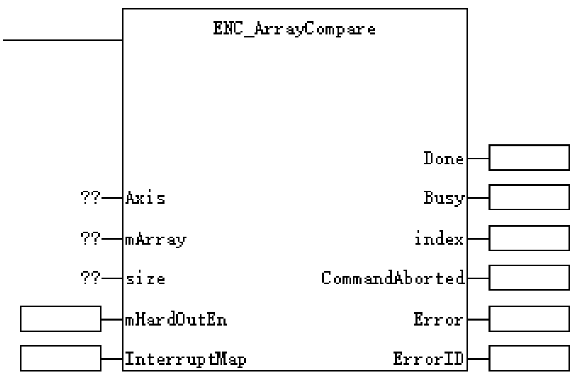


Note: When Enable is on the rising edge, the current input parameters are valid; when Enable is in the constant ON state, it is invalid to modify the input parameters in the graphic block being executed.

3.24.8 ENC_ArrayCompare

ENC_ArrayCompare - unidimensional array comparison of encoder.

Graphic Block



16-Bit command	-						
32-Bit command	ENC_ArrayCompare: Unidimensional Array Comparison of Encoder						
Operand	Name	Description	Supported element	Nullable	Default value	Range	Data Type
S1	Axis	Axis number	-	No	-	-	-
S2	mArray	Starting comparison position	Const; D; R; custom variable	No	-	-	REAL
S3	Size	Length	Const; D; R; custom variable	No	-	1~100	REAL
S4	OutPut Enable	Hardware output enable, where ON: Enable; OFF: Disable	X; M; S; Y; Custom variable	Yes	OFF	ON, OFF	BOOL
S5	InterruptMap	Interrupt number, where	Const; D; R; custom variable	Yes	0	0~16	INT

16-Bit command	-						
32-Bit command	ENC_ArrayCompare: Unidimensional Array Comparison of Encoder						
Operand	**Name**	**Description**	**Supported element**	**Nullable**	**Default value**	**Range**	**Data Type**
		0: does not associate comparison interrupt 1-16: associates comparison interrupt					
D1	Done	Completion flag	M; S; Y; Custom variable	Yes	OFF	ON, OFF	BOOL
D2	Busy	Executing	M; S; Y; Custom variable	Yes	OFF	ON, OFF	BOOL
D3	Index	Next comparison position	D; R; custom variable	Yes	0	0~100	INT
D4	Command Aborted	Execution interrupt	M; S; Y; Custom variable	Yes	OFF	ON, OFF	BOOL
D5	Error	Error state	M; S; Y; Custom variable	Yes	OFF	ON, OFF	BOOL
D6	ErrorID	Error code	D; R; Custom variable	Yes	0	0~65535	INT

Function Description

1. System Parameter Settings:

In the "Compare Output Set" interface of the local pulse axis, check "Compare Output Enable", select the comparison output terminal, and choose whether to output pulses by time or by unit.

2. Graphic Block Settings:

If OutputEnable is set to 1, the set comparison output terminal generates a comparison output signal; if OutputEnable is set to 0, no comparison output signal is generated. InterruptMap is used to associate comparison interrupt subroutines. When it is set to 0, interrupt subroutines are not associated; when it is set to 1-16, interrupt subroutines are associated.

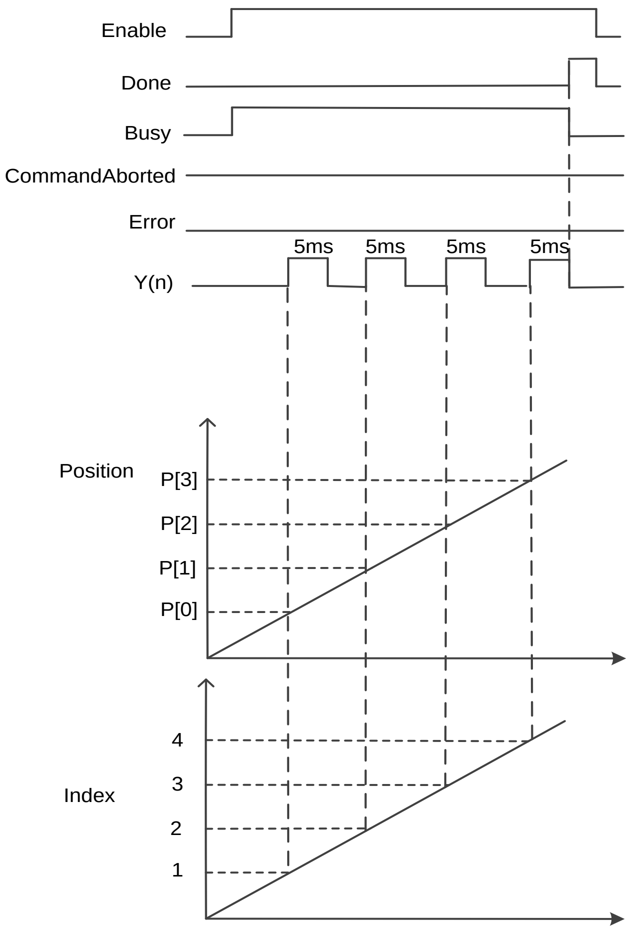
3. Comparison Settings:

After the encoder axis reaches the comparison point, the digital output terminal changes to a high level, and the duration of the high level is determined by the parameters set in the background configuration interface. The number of comparison points is specified by Size, and the comparison point array is specified by Array. Index represents the next array coordinate point to be compared for output, and the array content must be incremented or decremented sequentially.

Note: When Enable is on the rising edge, the current input parameters are valid; when Enable is in the constant ON state, it is invalid to modify the input parameters in the graphic block being executed.

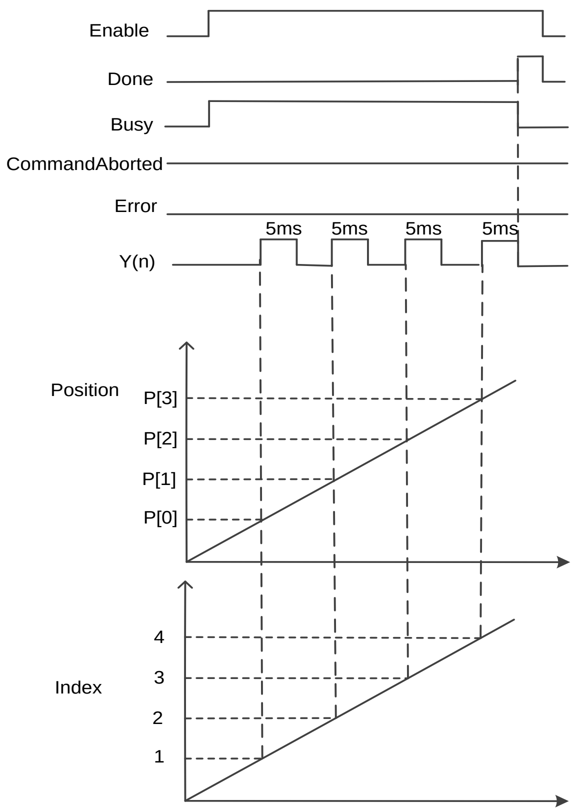
Timing diagram

- The comparison point group P[4] is set, the output time at each comparison point is 5 ms, and the command starts running after Enable is enabled.



Note: If point Y is configured as a comparison output point, normal output control will be invalid.

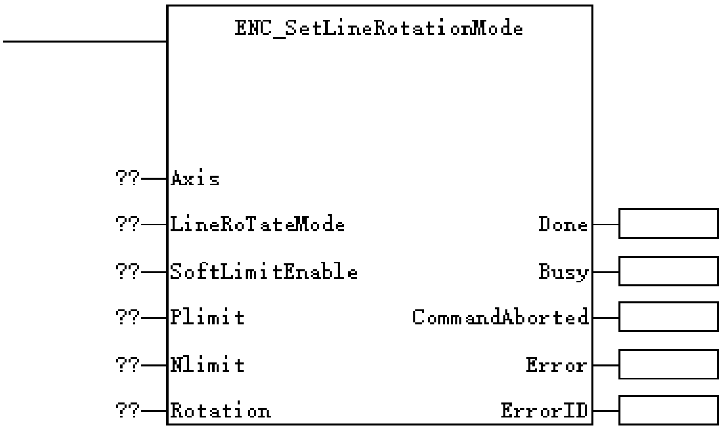
- The comparison point group P[4] is set, the output time at each comparison point lasts for 2 pulses, and the command starts running after Enable is enabled.



3.24.9 ENC_SetLineRotationMode

ENC_SetLineRotationMode - set axis operation mode

Graphic Block



16-Bit command	-						
32-Bit command	ENC_SetLineRotationMode: Set Axis Operation Mode						
Operand	**Name**	**Description**	**Supported element**	**Nullable**	**Default value**	**Range**	**Data Type**
S1	Axis	Axis number	-	No	-	-	-
S2	LineRo TateMode	Mode selection 0: linear mode; 1: periodic mode	Const; D; R; custom variable	No	-	0~1	INT
S3	SoftLimitEnable	Limit function	X; M; S; Y; custom variable	No	-	ON, OFF	BOOL
S4	Plimit	Positive limit value	Const; D; R; custom variable	No	-	-	REAL
S5	Nlimit	Negative limit value	Const; D; R; custom variable	Yes	-	-	REAL
S6	Rotation	Periodic value	Const; D; R; custom variable	Yes	-	-	REAL
D1	Done	Completion flag	M; S; Y; custom variable	Yes	OFF	ON, OFF	BOOL
D2	Busy	Executing	M; S; Y; custom variable	Yes	OFF	ON, OFF	BOOL
D4	Command Aborted	Execution interrupt	M; S; Y; custom variable	Yes	OFF	ON, OFF	BOOL
D5	Error	Error state	M; S; Y; custom variable	Yes	OFF	ON, OFF	BOOL
D6	ErrorID	Error code	D; R; custom variable	Yes	0	0~65535	INT

Function Description

1. This command enables the PLC to reconfigure the linear rotation mode of the local encoder axis after powering on, downloading programs, or executing a RUN/STOP operation and before enabling the counting of the local encoder axis.
2. In case of LineRotateMode=0, the local encoder axis is in linear mode. In the linear mode, SoftLimitEnable=OFF indicates disabling the limit; SoftLimitEnable=ON indicates enabling the limit, where PLimit represents the positive limit value and NLimit represents the negative limit value.

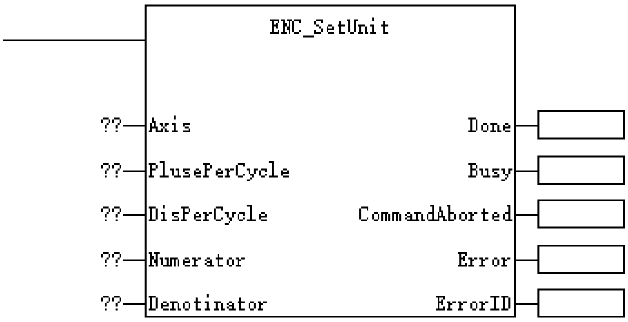
3. In case of LineRotateMode=1, the local encoder axis is in rotary mode. At this point, Rotation represents the value of the rotation cycle. When the encoder axis counts positively, the counting value cycles from 0 to the cycle value; when the encoder axis counts negatively, the counting value cycles from the cycle value to 0.

Note: When Enable is on the rising edge, the current input parameters are valid; when Enable is in the constant ON state, it is invalid to modify the input parameters in the graphic block being executed.

3.24.10 ENC_SetUnit

ENC_SetUnit - set axis gear ratio

Graphic Block



16-Bit command	-						
32-Bit command	ENC_SetUnit: Continuous Execution						
Operand	**Name**	**Description**	**Supported element**	**Nullable**	**Default value**	**Range**	**Data Type**
S1	Axis	Axis number	-	No	-	-	-
S2	PlusePerCycle	The number of pulses per revolution of the encoder	Const; D; R; custom variable	No	-	Positive number	Dword
S3	DisPerCycle	The distance per revolution of the workbench	Const; D; R; custom variable	No	-	0.01~ 9999999	REAL
S4	Numerator	Numerator of the gear ratio	Const; D; R; custom variable	No	-	Positive number	Dword
S5	Denotinator	Denominator of the gear ratio	Const; D; R; custom variable	Yes	-	Positive number	Dword
D1	Done	Completion flag	M; S; Y; custom variable	Yes	OFF	ON, OFF	BOOL
D2	Busy	Executing	M; S; Y; custom variable	Yes	OFF	ON, OFF	BOOL
D4	CommandAborted	Execution interrupt	M; S; Y; custom variable	Yes	OFF	ON, OFF	BOOL
D5	Error	Error state	M; S; Y; custom variable	Yes	OFF	ON, OFF	BOOL
D6	ErrorID	Error code	M; S; Y; custom variable	Yes	0	0~65535	INT

Note: When Enable is on the rising edge, the current input parameters are valid; when Enable is in the constant ON state, it is invalid to modify the input parameters in the graphic block being executed.

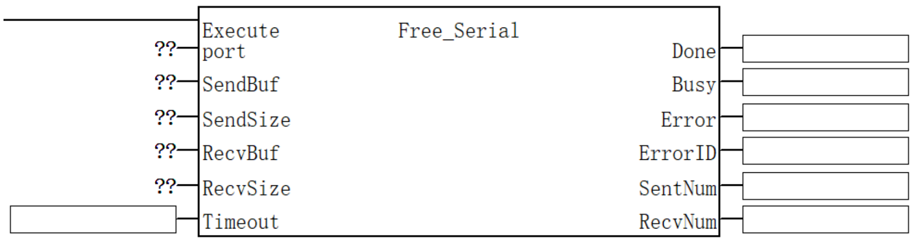
3.25 Communication Commands

3.25.1 Command Table

Command Category	Communication protocol	Name	Function
Communication Protocol Command	Free Protocol for Serial Ports	Free_Serial	Sending and receiving under free protocol for serial ports
	Free Protocol for TCP/IP	TCP_Server	Server socket creation
		TCP_Accept	Reception of client connection request by server
		TCP_Client	Client socket creation
		TCP_Send	Send TCP data
		TCP_Recv	Receive TCP data
		TCP_Close	Close TCP socket
	Free Protocol for UDP/IP	UDP_Peer	UDP socket creation
		UDP_Send	Send UDP data
		UDP_Recv	Receive UDP data
	EtherCAT	ECAT_ReadParameter_CoE	Read SDO parameter from slave station
		ECAT_WriteParameter_CoE	Write SDO parameter to slave station
		ECAT_RestartMaster_CoE	Restart EtherCAT master station

3.25.2 Free_Seral Commands

Graphic Block



16-Bit command	Free_Seral: Sending and Receiving Under Free Protocol for Serial Ports					
32-Bit command	-					
Operand	**Name**	**Description**	**Nullable**	**Default value**	**Range**	**Data Type**
S1	port	Port number	No	-	1~3	WORD
S2	SendBuf	Sending buffer	No	-	-	INT
S3	SendSize	The number of bytes sent	No	-	-	INT
S4	RecvBuf	Receiving buffer	No	-	-	INT
S5	RecvSize	The number of bytes received	No	-	-	INT
S6	Timeout	Receiving timeout time	Yes	1000	-	WORD
D1	Done	Completion flag	Yes	OFF	ON, OFF	BOOL
D2	Busy	Ongoing execution flag	Yes	OFF	ON, OFF	BOOL
D3	Error	Error flag	Yes	OFF	ON, OFF	BOOL
D4	ErrorID	Error code	Yes	0	-	WORD
D5	SentNum	The number of bytes already sent	Yes	0	-	WORD
D6	RecvNum	The number of bytes already received	Yes	-	-	WORD

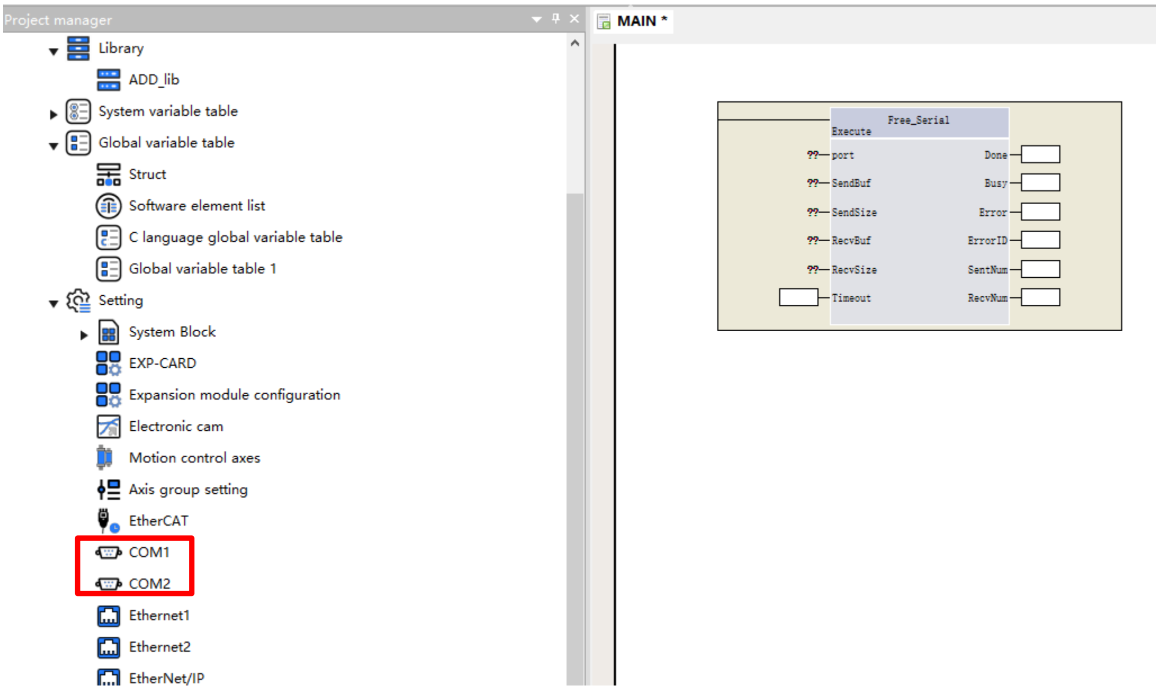
Operand	Const	Y	M	S	D	R
S1	√	-	-	-		
S2		-	-	-	√	√
S3	√	-	-	-	√	√
S4		-	-	-	√	√
S5	√	-	-	-	√	√
S6	√	-	-	-	√	√
D1	-	√	√	√	-	-
D2	-	√	√	√	-	-
D3	-	√	√	√	-	-
D4	-	-	-	-	√	√
D5	-	-	-	-	√	√
D6	-	-	-	-	√	√

Function Description

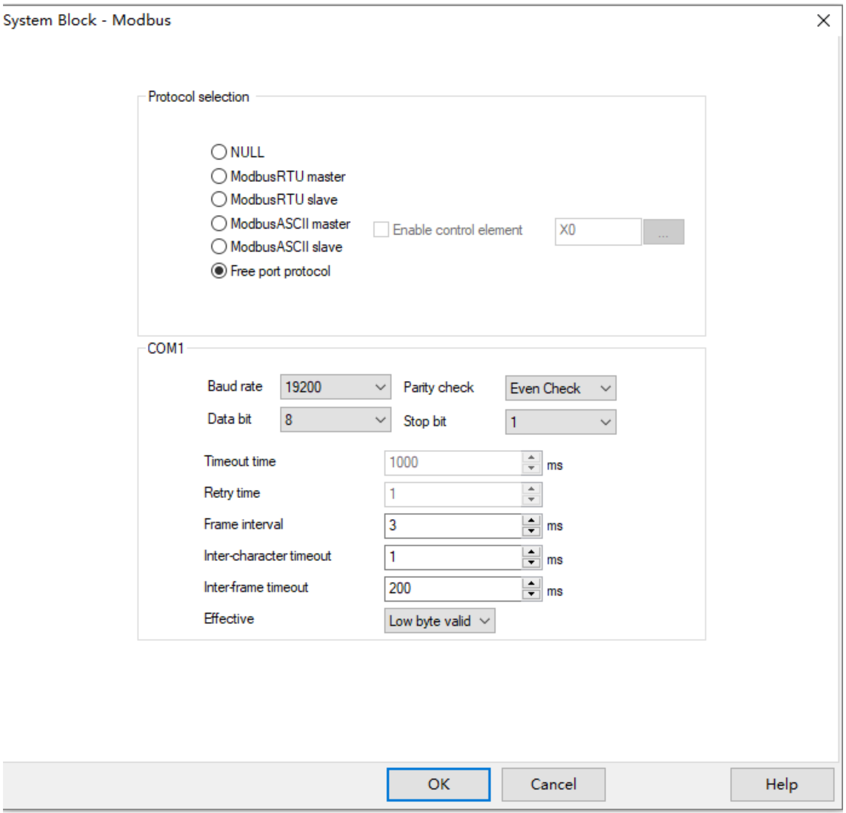
This command implements the sending and receiving of the data under the free protocol for serial ports. After the rising edge of the command is triggered, the command sends data of the specified length (SendSize) in the specified sending buffer (SendBuf) through the specified port (Port). After sending, the command receives data of the specified length (RecvSize) and places it in the specified receiving buffer (RecvBuf).

Instructions for System Block Configuration

1. Before using a free protocol command, it is necessary to configure the corresponding COM port in the system block. First, double-click the COM port in the system block configuration.



2. Select "Free port protocol" in the pop-up box.



Instructions for System Block Parameters

1、 "Baud rate", "Parity check", "Data bit", and "Stop bit" can be set as needed, just like the peer device.

2、 For the free port protocol, there is no need to configure "Timeout time" and "Retry time".

3、 "Frame interval": The sending interval (in ms) between every two frames of data.

4、 "Inter-frame timeout ": A value, which causes the time for receiving two received bytes to be discarded when exceeded.

5、 "Effective": "Low byte valid": When transmitting or receiving data, manipulate the low byte of a word element, if two bytes are to be transmitted, then transmit the low byte of two word elements. "High and low byte valid": When transmitting or receiving data, manipulate the high and low byte of a word element, if two bytes are to be transmitted, then transmit the high byte and low byte of a word element.

Instructions for Command Parameters

S1: port number. COM1 corresponds to port1, and so on.

S2: sending buffer. When SendSize is 0, this parameter is invalid, and the data from the specified element is sent.

S3: the number of bytes sent. The data of specified bytes is sent. When the port only receives data, SendSize is set to 0.

S4: receiving buffer. When SendSize is 0, this parameter is invalid, and the received data is placed in the specified element. When the port only sends data, RecvSize is set to 0.

S5: the number of bytes received. The data of specified bytes is received. When the port only sends data, RecvSize is set to 0.

When both SendSize and RecvSize are 0, this command is invalid.

When neither SendSize nor RecvSize is 0, the port sends data and receives data within the timeout time.

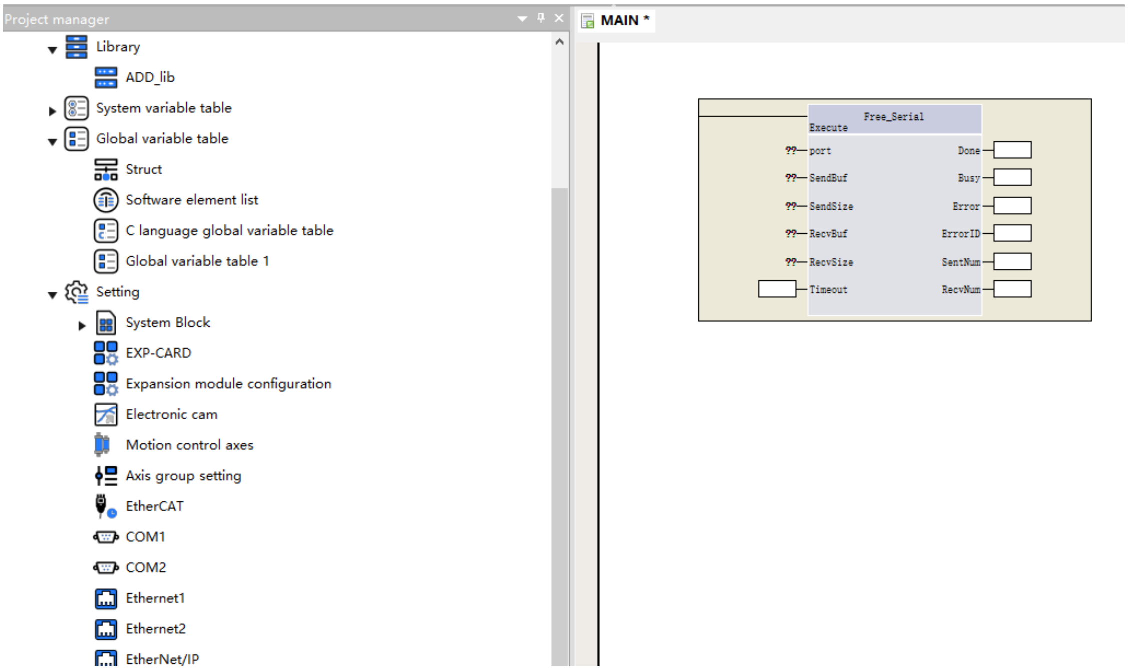
Precautions

When called, up to 512 copies of this commands are supported.

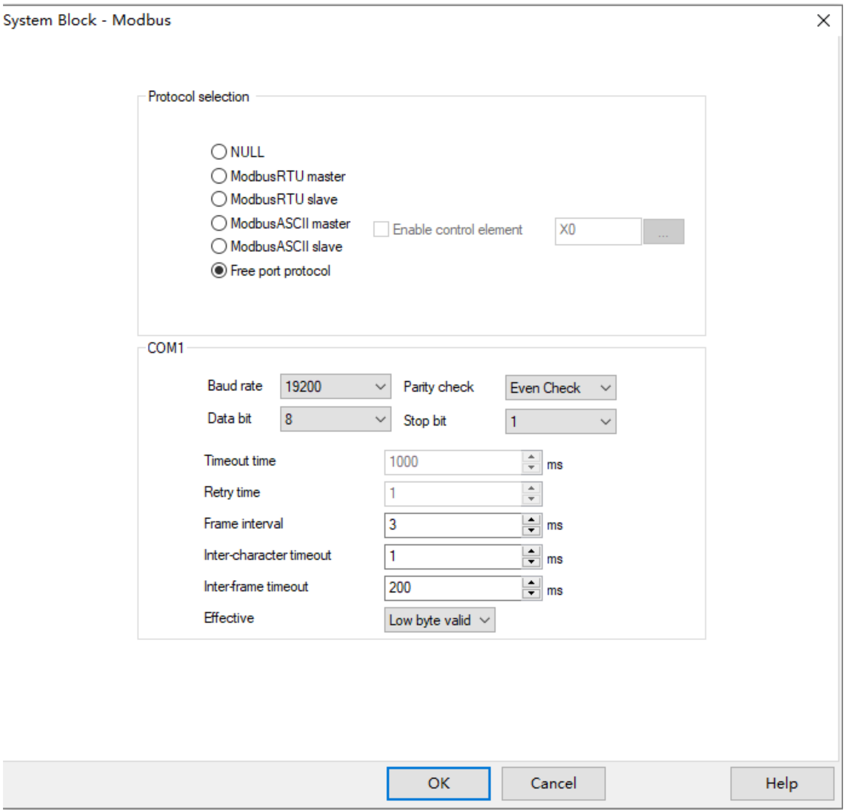
Application Example

The COM1 serial port is used to send 5 bytes of data from D0~D4 (taking the low bytes of the word elements), and then receive 5 bytes of data to store it in D10~D14.

1. Firstly, double click on "COM1" in the system block.



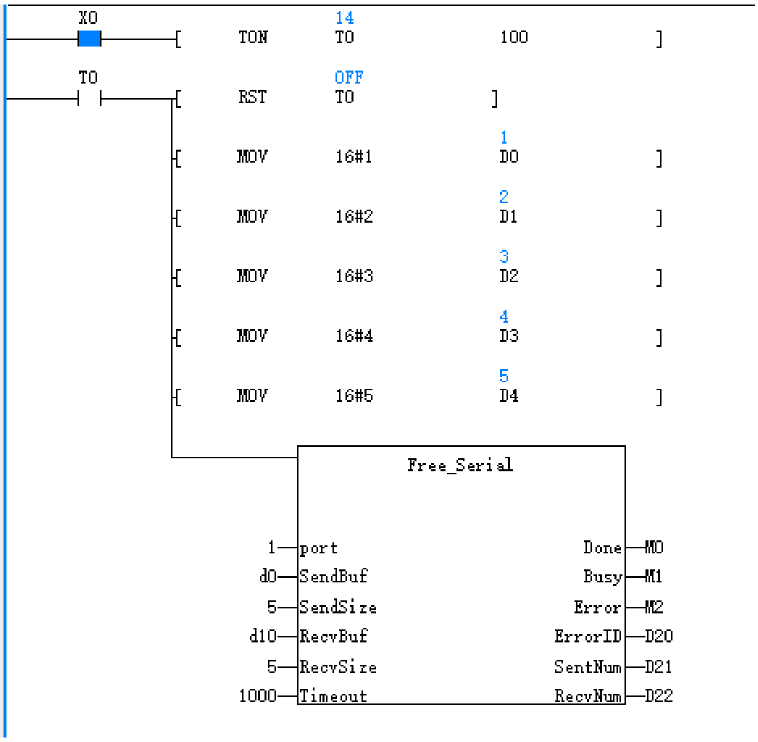
2. In the system block settings, set communication port 1 as a free port, and then set "Baud rate", "Parity check", "Data bit", "Stop bit", etc.



3. Write the data to be sent to the sending buffer. In this example, the data in the following table is sent.

D0	D1	D2	D3	D4
0x01	0x02	0x03	0x04	0x05

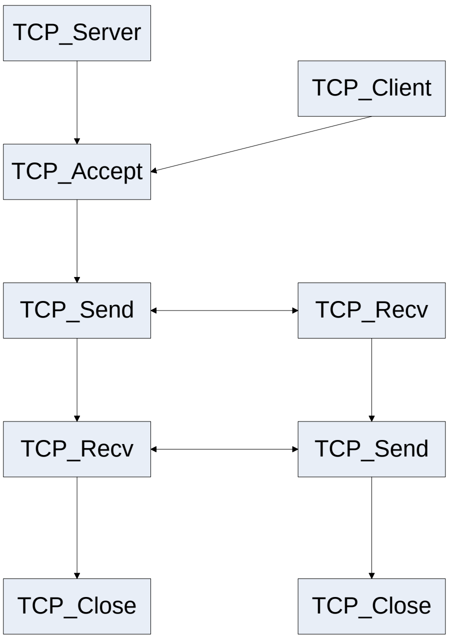
4. Write a ladder diagram program to send and receive data.



5. In the ladder diagram program, taking every 100 ms as an interval, use COM1 to take the low bytes of D0~D4, send 5 bytes of data, and wait for 1000 ms to receive 5 bytes of data and place them in D10~D14.

3.25.3 TCP communication

TCP is a connection-oriented full duplex communication, where each TCP connection can only have two endpoints and can only be made in a point-to-point manner. TCP provides reliable delivery services. The data transmitted through a TCP connection is error-free, non-lost, and non-duplicated, and arrives in sequence; the software framework for TCP communication is shown in the following figure.



Precautions

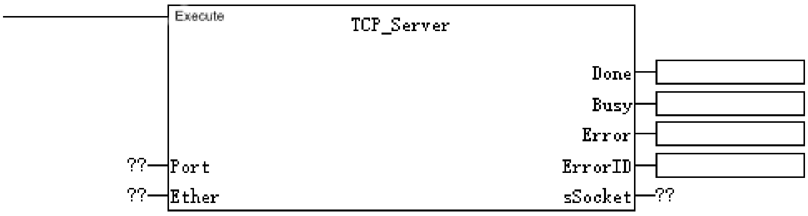
The TCP client and server currently support a single connection, and the number of data in the data transmission and reception function block cannot exceed 16.

Due to functional limitations, sSocket is temporarily a DINT type array with a size of 10. The meanings of each member of the array are shown in the table below:

Member	Meaning
sSocket[0]	Bit0: validity state, which indicates the socket validity (client) Bit1: connection state, which indicates whether the socket is connected Bit2: listening state Other Bits: reserved
sSocket[1]	Socket ID
sSocket[2]	Low 16 bits: free protocol type, where 1 indicates TCP and 2 indicates UDP High 16 bits: the number of connected devices (server side)
sSocket[3]	Peer IP address
sSocket[4]	Socket
sSocket[5]	Local port number
sSocket[6]	Reserved
sSocket[7]	Reserved
sSocket[8]	Reserved
sSocket[9]	Reserved

3.25.4 TCP_Server Commands

Graphic Block



Command Table	Function block form			Applicable model		TS600 series
16-Bit command	TCP_Server: Server Socket Creation					
32-Bit command	-					
Operand name	Name	Description	Supported element	Element Z indexing	Nullable	Data Type
S1	Port	Port number	Const; D; R	No	No	WORD
S2	Ether	Network interface number	Const	No	No	WORD
D1	Done	Completion flag	M; S; Dx.y	No	No	BOOL
D2	Busy	Ongoing execution flag	M; S; Dx.y	No	No	BOOL
D3	Error	Function block error flag	M; S; Dx.y	No	No	BOOL
D4	ErrorID	Error code	D; R	No	No	WORD
D5	sSocket	Output socket	D; R	No	No	DINT

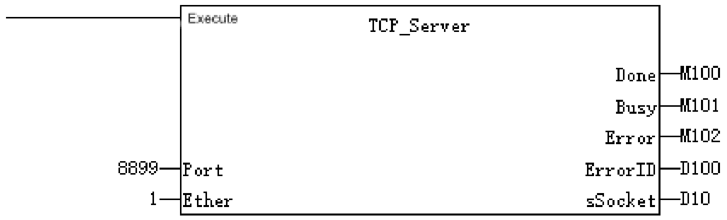
Function Description

1. For a server, this command is mainly used to create a socket for the server and return the corresponding creation state and creation value.
2. This command is executed on the rising edge.

Precautions

After a successful connection, if the socket is not closed, it is prohibited to trigger the function block again.

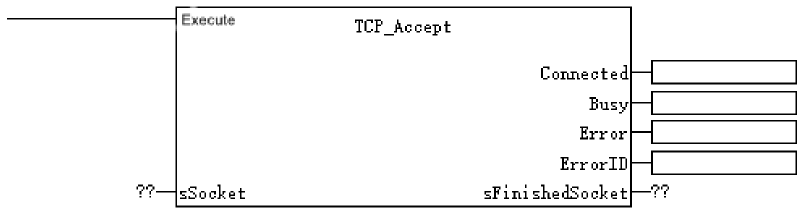
Application Example



3.25.5 TCP_Accept Commands

The socket at the socket communication server side of TCP receives client requests to create connection sockets.

Graphic Block

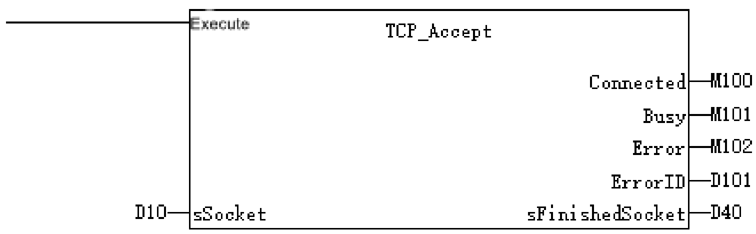


Command Table	Function block form			**Applicable model**		TS600 series
16-Bit command	TCP_Accept: Reception of Client Connection Request by Server					
32-Bit command	-					
Operand name	**Name**	**Description**	**Supported element**	**Element Z indexing**	**Nullable**	**Data Type**
S1	sSocket	Server-side socket	D; R	No	No	WORD
D1	Connected	Connection made or not	M; S; Dx.y	No	Yes	BOOL
D2	Busy	Ongoing execution flag	M; S; Dx.y	No	Yes	BOOL
D3	Error	Function block error flag	M; S; Dx.y	No	Yes	BOOL
D4	ErrorID	Error code	D; R	No	Yes	WORD
D5	sFinishedSocket	Socket already connected	D; R	No	No	WORD

Function Description

1. For a server, this command is mainly used to create a socket for the client and return the successfully connected socket.
2. This command is valid at high levels.

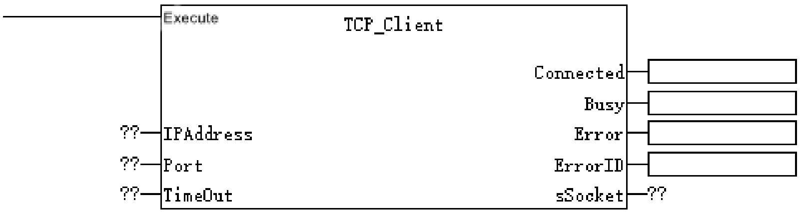
Application Example



3.25.6 TCP_Client Commands

The PLC uses this command to create a TCP server and input the server port number and network interface serial number.

Graphic Block



Command Table	Function block form			Applicable model		TS600 series
16-Bit command	TCP_Client: Client Socket Creation					
32-Bit command	-					
Operand name	Name	Description	Supported element	Element Z indexing	Nullable	Data Type
S1	IPAddress	Server IP address	Const; D; R	No	No	DWORD
S2	Port	Port number	Const; D; R	No	No	WORD
S3	TimeOut	Timeout time	Const; D; R	No	No	WORD
D1	Connected	Connection made or not	M; S; Dx.y	No	Yes	BOOL
D2	Busy	Ongoing execution flag	M; S; Dx.y	No	Yes	BOOL
D3	Error	Function block error flag	M; S; Dx.y	No	Yes	BOOL
D4	ErrorID	Error code	D; R	No	Yes	WORD
D5	sSocket	Output socket	D; R	No	No	DINT

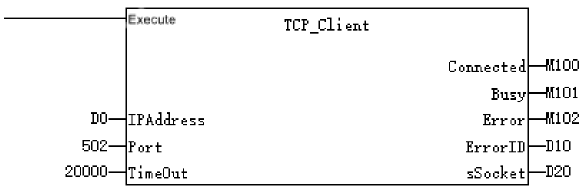
Function Description

1. For a client, this command is mainly used to create a socket for the client and return the corresponding creation state and creation value.
2. This command is executed on the rising edge.

Precautions

After a successful connection, if the socket is not closed, it is prohibited to trigger the function block again.

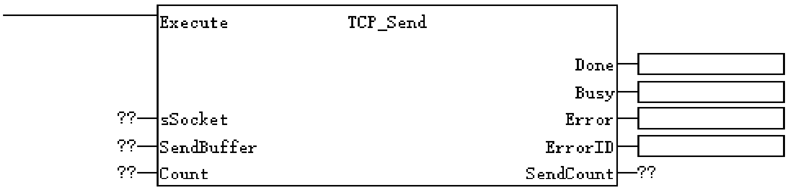
Application Example



3.25.7 TCP_Send Commands

TCP uses this command to send data for socket communication.

Graphic Block



Command Table	Function block form			Applicable model		TS600 series
16-Bit command	TCP_Send: Send TCP Data					
32-Bit command	-					
Operand name	Name	Description	Supported element	Element Z indexing	Nullable	Data Type
S1	sSocket	Input socket	D; R	No	No	DINT
S2	SendBuffer	Sending data storage array	D; R	No	No	INT
S3	Count	The number of data sent (it should be less than or equal to the number of members in the data storage array)	Const; D; R	No	No	WORD
D1	Done	Completion signal flag	M; S; Dx.y	No	Yes	BOOL
D2	Busy	Ongoing execution flag	M; S; Dx.y	No	Yes	BOOL
D3	Error	Function block error flag	M; S; Dx.y	No	Yes	BOOL
D4	ErrorID	Error code	D; R	No	Yes	WORD
D5	SendCount	The number of data sent (if it is less than the number of data input, sending cannot be enabled again)	D; R	No	No	WORD

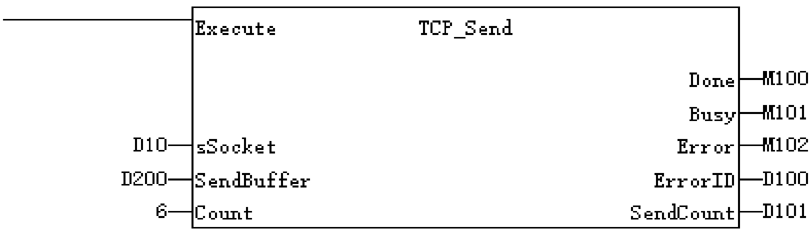
Function Description

1. This command is used as a function to send socket data.
2. This command is executed on the rising edge.

Precautions

The number of data sent cannot exceed the capacity of the sending data storage area.

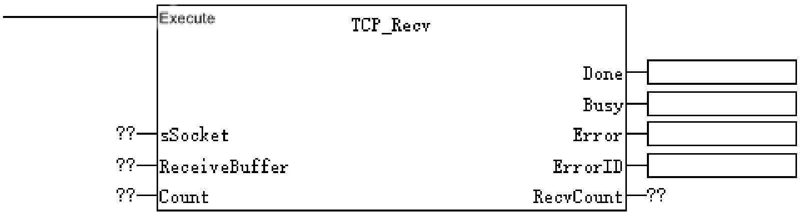
Application Example



3.25.8 TCP_Recv Commands

TCP uses this command to receive data for socket communication.

Graphic Block



Command Table	Function block form			Applicable model		TS600 series
16-Bit command	TCP_Recv: Receive TCP Data					
32-Bit command	-					
Operand name	**Name**	**Description**	**Supported element**	**Element Z indexing**	**Nullable**	**Data Type**
S1	sSocket	Input socket	D; R	No	No	DINT
S2	ReceiveBuffer	Receiving data storage array	D; R	No	No	INT
S3	Count	The number of data received (it should be less than or equal to the number of members in the data storage array)	Const; D; R	No	No	WORD
D1	Done	Completion signal flag	M; S; Dx.y	No	Yes	BOOL
D2	Busy	Ongoing execution flag	M; S; Dx.y	No	Yes	BOOL
D3	Error	Function block error flag	M; S; Dx.y	No	Yes	BOOL
D4	ErrorID	Error code	D; R	No	Yes	WORD
D5	RecvCount	The number of stored received data	D; R	No	No	WORD

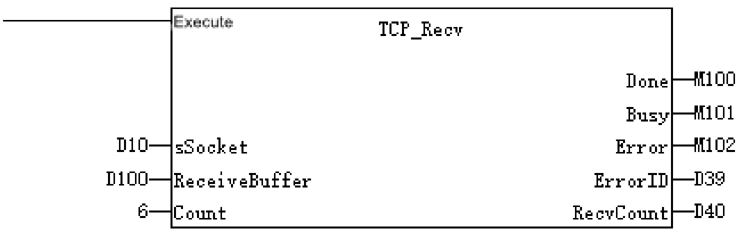
Function Description

1. This command is used as a function to receive socket data.
2. This command is executed on the rising edge.

Precautions

The number of data received cannot exceed the capacity of the receiving array.

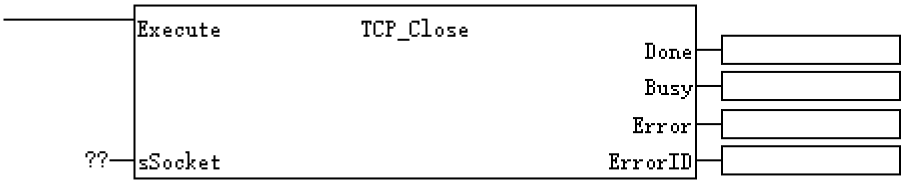
Application Example



3.25.9 TCP_Close Commands

This command to used to close the socket communication of the TCP connection.

Graphic Block



Command Table	Function block form			**Applicable model**		TS600 series
16-Bit command	TCP_Close: Close TCP socket					
32-Bit command	-					
Operand name	**Name**	**Description**	**Supported element**	**Element Z indexing**	**Nullable**	**Data Type**
S1	sSocket	Input socket	D; R	No	No	DINT
D1	Done	Completion flag	M; S; Dx.y	No	Yes	BOOL
D2	Busy	Ongoing execution flag	M; S; Dx.y	No	Yes	BOOL
D3	Error	Function block error flag	M; S; Dx.y	No	Yes	BOOL
D4	ErrorID	Error code	D; R	No	Yes	WORD

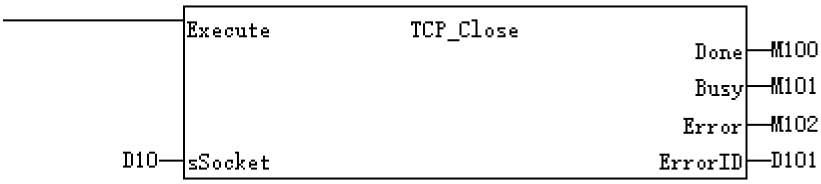
Function Description

1. This command closes the TCP socket.
2. This command is executed on the rising edge.

Precautions

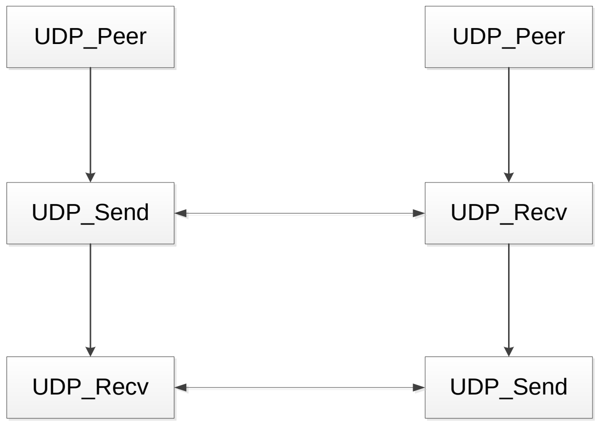
After a successful connection, if the socket is not closed, it is prohibited to trigger the function block again.

Application Example



3.25.10 UDP communication

UDP (User Datagram Protocol) is a packet-oriented connectionless communication, which is characterized by no congestion control, low latency during data transmission, and high data transmission efficiency. Therefore, it is suitable for applications that don' t require high reliability. The main framework for UDP communication is shown below:



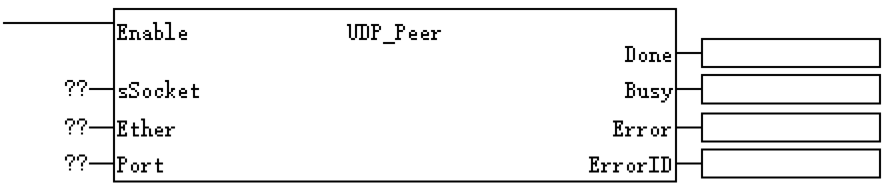
Precautions

UDP currently support a single connection, and the number of data in the data transmission and reception function block cannot exceed 16.

3.25.11 UDP_Peer Commands

This command mainly functions to return the description word of the connected UDP socket and bind it to the local port number.

Graphic Block



Command Table	Function block form			Applicable model		TS600 series
16-Bit command	UDP_Peer: UDP Socket Creation					
32-Bit command	-					
Operand name	**Name**	**Description**	**Supported element**	**Element Z indexing**	**Nullable**	**Data Type**
S1	sSocket	Socket	D; R	No	No	DINT
S2	Ether	Network interface number	Const	No	No	WORD
S3	Port	Port number	Const; D; R	No	No	WORD
D1	Done	Completion flag	M; S; Dx.y	No	Yes	BOOL
D2	Busy	Ongoing execution flag	M; S; Dx.y	No	Yes	BOOL
D3	Error	Function block error flag	M; S; Dx.y	No	Yes	BOOL
D4	ErrorID	Error code	D; R	No	Yes	WORD

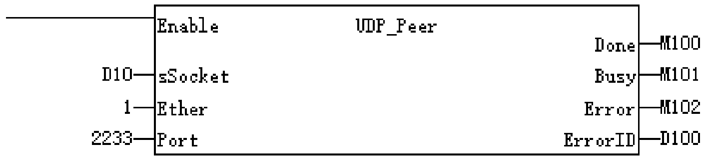
Function Description

1. For a client, this command is mainly used to create a socket for the client and return the corresponding creation state and creation value.
2. This command is executed on the rising edge.

Precautions

After a successful connection, if the socket is not closed, it is prohibited to trigger the function block again.

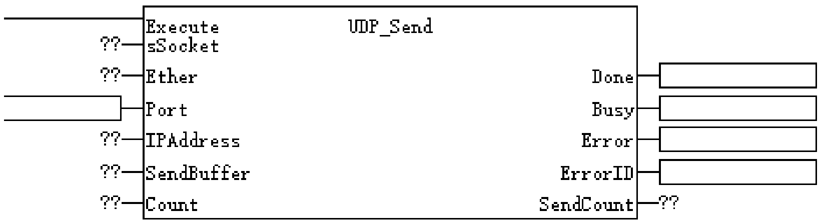
Application Example



3.25.12 UDP_Send Commands

This command mainly functions to send data to the target IP address and port.

Graphic Block



Command Table	Function block form			**Applicable model**		TS600 series
16-Bit command	UDP_Send: Send UDP Data					
32-Bit command	-					
Operand name	**Name**	**Description**	**Supported element**	**Element Z indexing**	**Nullable**	**Data Type**
S1	sSocket	Input socket	D; R	No	No	DINT
S2	Ether	Network interface number	Const	No	No	WORD
S3	Port	Port number	Const; D; R	No	Yes	WORD
S4	IPAddress	Peer IP address	Const; D; R	No	No	DWORD
S5	SendBuffer	Sending data storage array	D; R	No	No	INT
S6	Count	The number of data sent (it should be less than or equal to the number of members in the data storage array)	Const; D; R	No	No	WORD
D1	Done	Completion signal flag	M; S; Dx.y	No	Yes	BOOL
D2	Busy	Ongoing execution flag	M; S; Dx.y	No	Yes	BOOL
D3	Error	Function block error flag	M; S; Dx.y	No	Yes	BOOL
D4	ErrorID	Error code	D; R	No	Yes	WORD
D5	SendCount	The number of data sent (if it is less than the number of data input, sending cannot be enabled again)	D; R	No	No	WORD

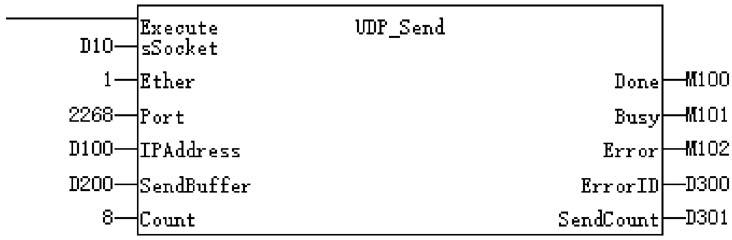
Function Description

1. This command is used as a function to send UDP data.
2. This command is executed on the rising edge.

Precautions

The number of data sent cannot exceed the capacity of the sending data storage area.

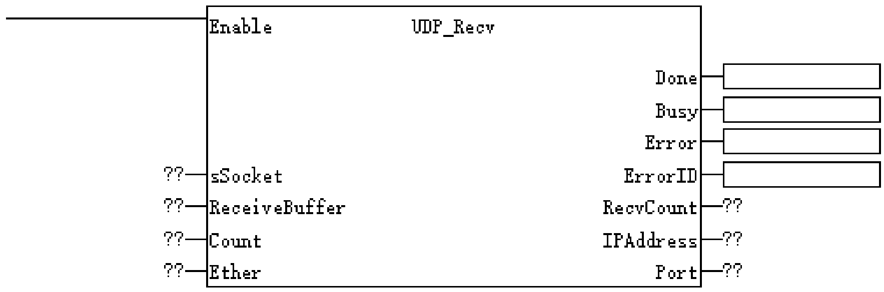
Application Example



3.25.13 UDP_Receive Commands

This command mainly functions to receive the data send from a remote end

Graphic Block



Command Table	Function block form			Applicable model		TS600 series
16-Bit command	UDP_Receive: Receive UDP Data					
32-Bit command	-					
Operand name	Name	Description	Supported element	Element Z indexing	Nullable	Data Type
S1	sSocket	Input socket	D; R	No	No	DINT
S2	ReceiveBuffer	Receiving data storage array	D; R	No	No	INT
S3	Count	The number of data received (it should be less than or equal to the number of members in the data storage array)	Const; D; R	No	No	WORD
S4	Ether	Network interface number	Const	No	No	WORD
S5	Port	Port number	Const; D; R	No	Yes	WORD
S6	IPAddress	Peer IP address	Const; D; R	No	No	DWORD
D1	Done	Completion signal flag	M; S; Dx.y	No	Yes	BOOL
D2	Busy	Ongoing execution flag	M; S; Dx.y	No	Yes	BOOL
D3	Error	Function block error flag	M; S; Dx.y	No	Yes	BOOL
D4	ErrorID	Error code	D; R	No	Yes	WORD
D5	RecvCount	The number of stored received data	D; R	No	No	WORD

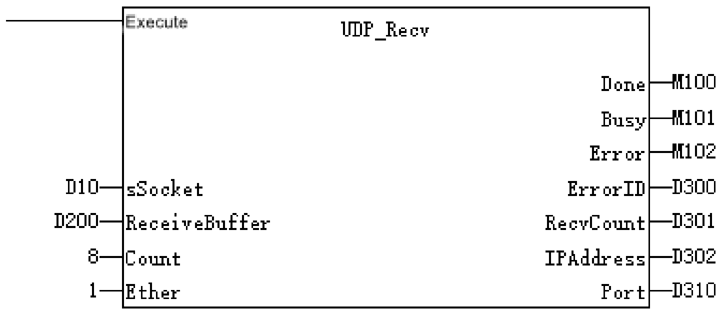
Function Description

1. This command is used as a function to receive UDP data.
2. This command is executed on the rising edge.

Precautions

The number of data received cannot exceed the capacity of the receiving array.

Application Example

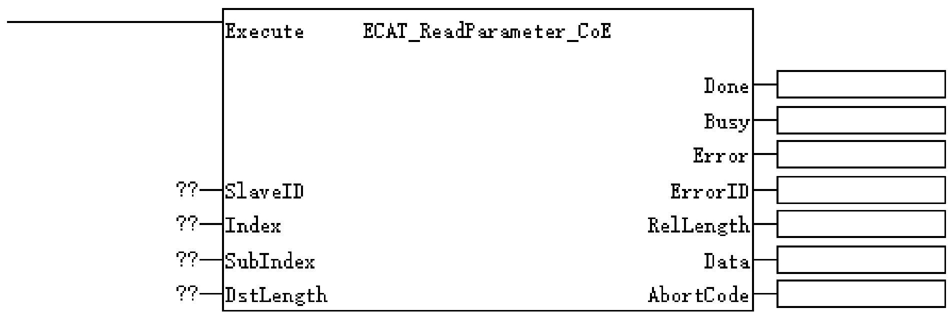


3.25.14 EtherCAT communication

3.25.15 ECAT_ReadParameter_CoE

It reads SDO parameters from the slave station.

Graphic Block



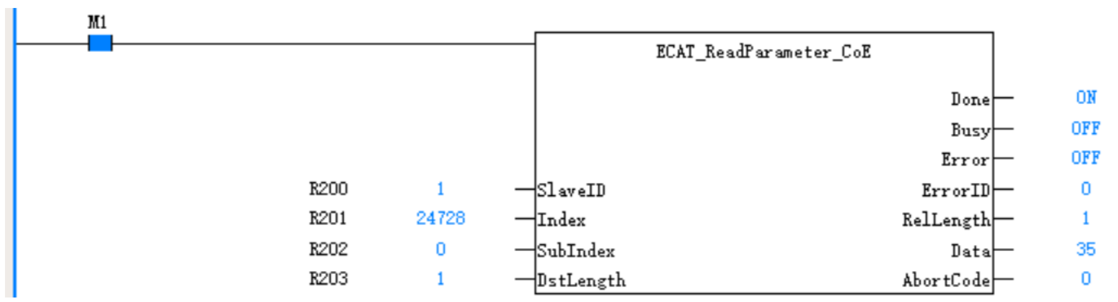
16-Bit command	-					
32-Bit command	ECAT_ReadParameter_CoE					
Operand	Name	Description	Nullable	Default value	Range	Data Type
S1	SlaveID	Only the configuration address of the slave station can be entered	No	0	0~71	INT
S2	Index	Index	No	0	Positive number	INT
S3	SubIndex	Sub-index	No	0	Positive number	INT
S4	DSTLength	Target data length	No	0	1, 2, and 4	INT
D1	Done	Completion signal flag	Yes	OFF	ON, OFF	BOOL
D2	Busy	Ongoing execution flag	Yes	OFF	ON, OFF	BOOL
D3	Error	Function block error flag	Yes	OFF	ON, OFF	BOOL
D4	ErrorID	Error code	Yes	0	-	INT
D5	RelLength	The actual length read, in bytes	No	0	1, 2, and 4	INT
D6	Data	Data read	No	0	-	DINT
D7	AbortCode	AbortCode generated when reading slave station failed	No	0	-	DINT

Operand	Const	Y	M	S	D	R	Custom Variable
S1	√	-	-	-	-	-	-
S2	√	-	-	-	√	√	√
S3	√	-	-	-	√	√	√
S4	√	-	-	-	√	√	√
D1	-	√	√	√	√	√	√
D2	-	√	√	√	√	√	√
D3	-	√	√	√	√	√	√
D4	-	-	-	-	√	√	√
D5	-	-	-	-	√	√	√
D6	-	-	-	-	√	√	√
D7	-	-	-	-	√	√	√

Function Description

1. This command is used to read the object dictionary of the EtherCAT slave station, and is valid to the rising edge.
2. SlaveID is used to specify the configuration address of the EtherCAT slave station.
3. On the rising edge of Execute, the command latches the left-side input parameters and triggers the reading of the object dictionaries specified by Index and SubIndex.
4. DstLength is used to specify the length of the object dictionary to be read in bytes.
5. After successful reading, the Done signal is valid, Dstate is used to display the read value, and RelLength is used to display the actual length of the object dictionary read. In case of failed reading, the Error output is valid, and AbortCode and ErrorID work together to determine the cause of the failure.
6. In this command, the Data parameter is a DINT type parameter, which occupies 4 bytes of space. When the object dictionary read is SINT or INT, the result read is placed in the low 8 or 16 bits of the Data parameter, and then the unused high 24 or 16 bits are padded with 0. For example, when reading -8 of SINT and INT types, the actual stored data of Data are 0x000000f8 and 0x0000fff8, respectively.

Application Example



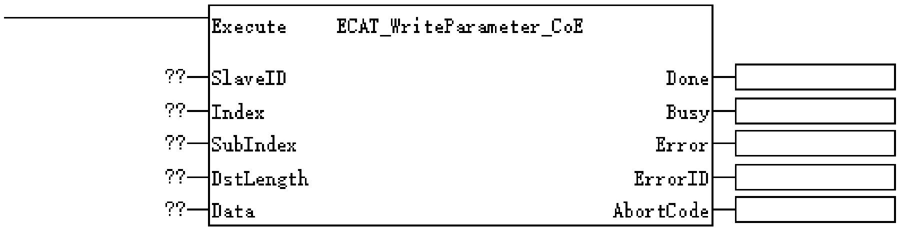
Error code

Error code	Possible cause	Solution
1	Slave station ID out of range	Check whether the slave station ID is out of range
2	Master station configuration failed	Check whether the EtherCAT communication is estalished
3	SDO communication failed	Check whether the SDO parameters are correct
4	Slave station disabled	Check whether the slave station is disabled

3.25.16 ECAT_WriteParameter_CoE

This command is used to write the SDO parameters of slave station.

Graphic Block



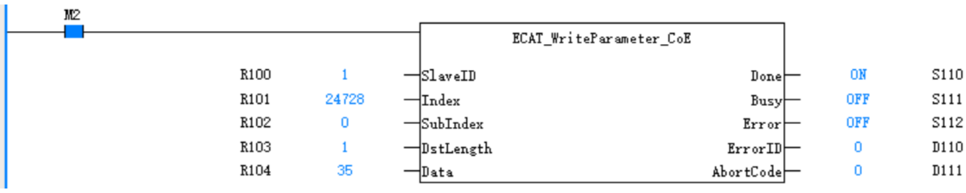
16-Bit command	-					
32-Bit command	ECAT_WriteParameter_CoE					
Operand	**Name**	**Description**	**Nullable**	**Default value**	**Range**	**Data Type**
S1	SlaveID	Only the configuration address of the slave station can be entered	No	0	0~71	INT
S2	Index	Index	No	0	Positive number	INT
S3	SubIndex	Sub-index	No	0	Positive number	INT
S4	DSTLength	Length of the data written	No	0	1/2/4	INT
S5	Data	Data written	No	0	-	DINT
D1	Done	Completion signal flag	Yes	OFF	ON/OFF	BOOL
D2	Busy	Ongoing execution flag	Yes	OFF	ON/OFF	BOOL
D3	Error	Function block error flag	Yes	OFF	ON/OFF	BOOL
D4	ErrorID	Error code	Yes	0	-	INT
D5	AbortCode	AbortCode generated when reading slave station failed	No	0	-	DINT

Operand	Const	Y	M	S	D	R	Custom Variable
S1	√	-	-	-	-	-	-
S2	√	-	-	-	√	√	√
S3	√	-	-	-	√	√	√
S4	√	-	-	-	√	√	√
S5	√	-	-	-	√	√	√
D1	-	√	√	√	√	√	√
D2	-	√	√	√	√	√	√
D3	-	√	√	√	√	√	√
D4	-	-	-	-	√	√	√
D5	-	-	-	-	√	√	√

Function Description

1. This command is used to write the object dictionary of the EtherCAT slave station, and is valid to the rising edge.
2. SlaveID is used to specify the configuration address of the EtherCAT slave station.
3. On the rising edge of Execute, the command latches the left-side input parameters and writes the data from Data to the object dictionaries specified by Index and SubIndex.
4. DstLength is used to specify the length of the object dictionary to be written in bytes.
5. After successful writing, the Done signal is valid. In case of failed writing, the Error output is valid, and AbortCode and ErrorID work together to determine the cause of the failure.

Application Example



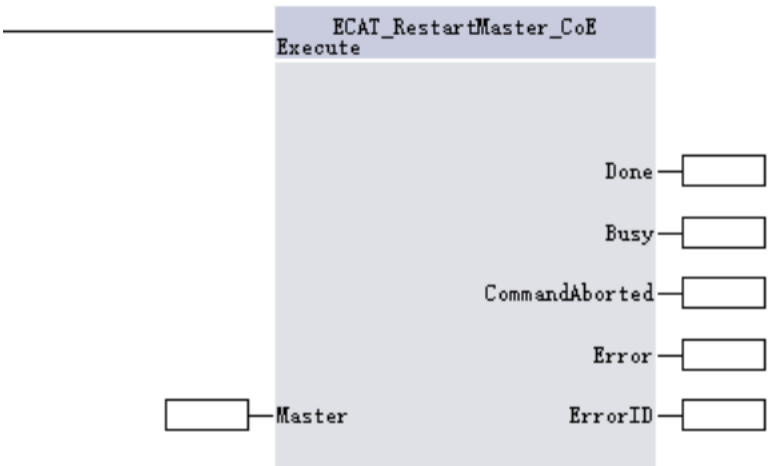
Error code

Error code	Possible cause	Solution
1	Slave station ID out of range	Check whether the slave station ID is out of range
2	Master station configuration failed	Check whether the EtherCAT communication is estalished
3	SDO communication failed	Check whether the SDO parameters are correct
4	Slave station disabled	Check whether the slave station is disabled

3.25.17 ECAT_ RestartMaster_CoE

This command is used to restart the EtherCAT master station.

Graphic Block



16-Bit command	ECAT_RestartMaster_CoE					
32-Bit command						
Operand	Name	Description	Nullable	Default value	Range	Data Type
S1	Master	EtherCAT master station	Yes	-	-	-
D1	Done	Completion signal flag	Yes	OFF	ON, OFF	BOOL
D2	Busy	Ongoing execution flag	Yes	OFF	ON, OFF	BOOL
D3	CommandAborted	Execution abortion	Yes	OFF	ON, OFF	BOOL
D4	Error	Function block error flag	Yes	OFF	ON, OFF	BOOL
D5	ErrorID	Error code	Yes	0	-	INT

Operand	Const	Y	M	S	D	R	Custom Variable
S1	√	-	-	-	√	√	√
D1	-	√	√	√	-	-	√
D2	-	√	√	√	-	-	√
D3	-	√	√	√	-	-	√

Operand	Const	Y	M	S	D	R	Custom Variable
D4	-	√	√	√	-	-	√
D5	-	-	-	-	√	√	√

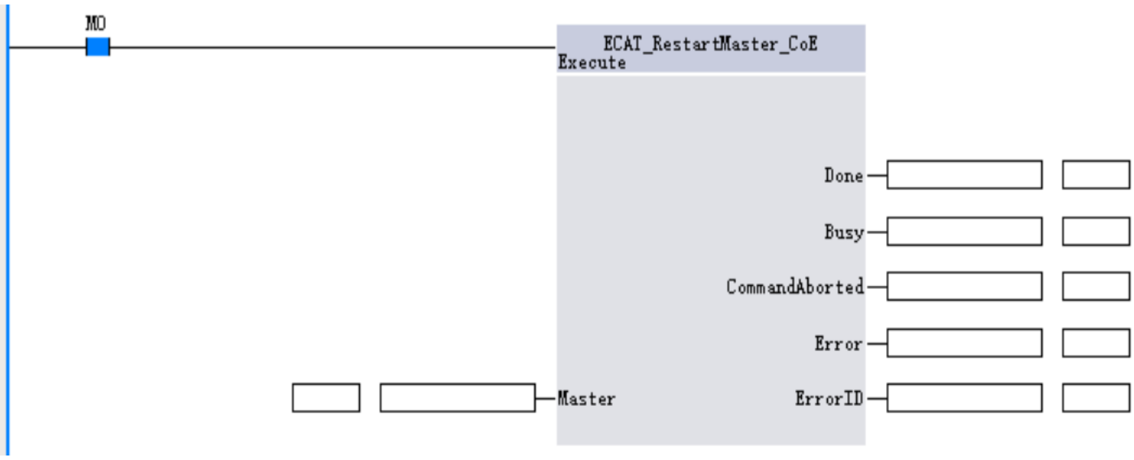
Function Description

Calling this command restarts the EtherCAT bus.

Precautions

This command does not allow multiple triggers. If the second command is triggered during the execution of the first command, the second command will not be executed, and the first command will continue to be executed.

Application Example



3.26 Real-time Clock Command

3.26.1 Command Table

Command Category	Name	Function
Real-time Clock Command	TRD	Real-time clock read
	TWR	Real-time clock write
	TADD	Clock addition operation
	TSUB	Clock subtraction operation
	HOUR	Hour meter
	DCMP=	Date comparison equal to
	DCMP>	Date comparison greater than
	DCMP<	Date comparison less than
	DCMP<>	Date comparison not equal to
	DCMP⩾	Date comparison greater than or equal to
	DCMP<=	Date comparison less than or equal to
	TCMP=	Time comparison equal to
	TCMP>	Time comparison greater than
	TCMP<	Time comparison less than
	TCMP<>	Time comparison not equal to
	TCMP⩾	Time comparison greater than or equal to
	TCMP<=	Time comparison less than or equal to
	HTO*S	Conversion from hours, minutes, or seconds to word/doubleword second data
	*STOH	Conversion from word/doubleword second data to hours, minutes, or seconds

3.26.2 TRD: Real-Time Clock Read

Command table	TRD (D)	Applicable model	TS600 series
16-Bit command	TRD: Real-Time Clock Read		
32-Bit command	-		

Operand	Type	Bit			Word		Indexing	Constant
		X, Y, M, LM, T, C, S	Dx.y	Custom bit variable	D, R, V, Z, T, C	Custom word variable		
D	WORD/ Array*7	-	-	-	√[1]	√	√	

Remark:

[1]Only the D, V, and R elements are supported.

Operand Description

D: The destination operand, which reads the starting unit that stores the system time, occupying seven consecutive units starting from the unit specified by D.

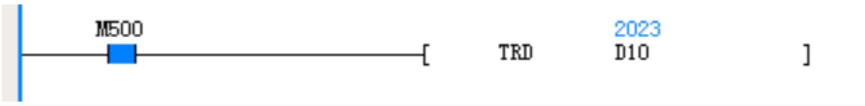
Function Description

1. When this command is driven, the time in the system is read and stored in the storage unit specified by D.
2. The addresses starting from D store year, month, day, hour, minute, second, and week, respectively.

Precautions

When there is a clock setting error in the system, TRD time reading is unsuccessful.

Application Example



		Element Name	Data Type	Display Format	Current Value
1	...	D10	WORD	Decimal	2023
2	...	D11	WORD	Decimal	4
3	...	D12	WORD	Decimal	22
4	...	D13	WORD	Decimal	15
5	...	D14	WORD	Decimal	14
6	...	D15	WORD	Decimal	37
7	...	D16	WORD	Decimal	3

In case of M500=ON, the system time is sent to the 7 units starting from D10. See below for the process.

	System variables	Item	Clock Data		Element	Item
Special Data Registers for Real-Time Clock	mYear	Year	2000~2099	→	D10	Year
	mMouth	Month	1~12	→	D11	Month
	mDay	Day	1~31	→	D12	Day
	mHour	Hour	0~23	→	D13	Hour
	mMinute	Minute	0~59	→	D14	Minute
	mSecond	Second	0~59	→	D15	Second
	mWeekday	Week	0~6	→	D16	Week

3.26.3 TWR: Real-Time Clock Write

Command table		TWR (S)			Applicable model	TS600 series		
16-Bit command		TWR: Real-time clock write						
32-Bit command		-						
Operand	Type	Bit			Word		Indexing	Constant
		X, Y, M, LM, T, C, S	Dx.y	Custom bit variable	D, R, V, Z, T, C	Custom word variable		
S	WORD/ Array*7	-	-	-	√[1]	√	√	

Remark:

[1]Only the D, V, and R elements are supported.

Operand Description

S: The source operand, which indicates the starting address of the soft element that writes the system time.

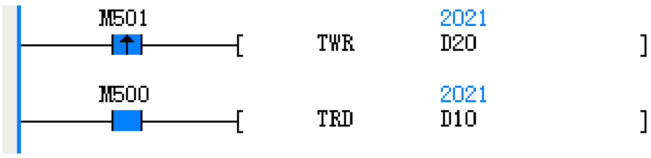
Function Description

When the system time differs from the actual time, the TWR command can be used to change the system time.

Precautions

- The written time data must meet the requirements of the Gregorian calendar, otherwise a clock reading-writing error will be reported, and this command will not be executed.
- It is recommended to execute this command by using the edge trigger.

Application Example



		Element Name	Data Type	Display Format	Current Value
1	...	D10	WORD	Decimal	2021
2	...	D11	WORD	Decimal	5
3	...	D12	WORD	Decimal	3
4	...	D13	WORD	Decimal	14
5	...	D14	WORD	Decimal	52
6	...	D15	WORD	Decimal	29
7	...	D16	WORD	Decimal	1
8	...		WORD	Decimal	
9	...	D20	WORD	Decimal	2021
10	...	D21	WORD	Decimal	5
11	...	D22	WORD	Decimal	
12	...	D23	WORD	Decimal	14
13	...	D24	WORD	Decimal	52
14	...	D25	WORD	Decimal	29
15	...	D26	WORD	Decimal	2
16	...		WORD	Decimal	
17	...	_sDateTime	_stru_DATE	Decimal	
18		_sDateTime.Second	INT	Decimal	29
19		_sDateTime.Minute	INT	Decimal	52
20		_sDateTime.Hour	INT	Decimal	14
21		_sDateTime.Day	INT	Decimal	3
22		_sDateTime.Month	INT	Decimal	5
23		_sDateTime.Year	INT	Decimal	2021
24		_sDateTime.WeekDay	INT	Decimal	1
25		_sDateTime.YearDay	INT	Decimal	0
26		_sDateTime.Timestamp	DINT	Decimal	1620053549

When M501 is ON, the time data of the 7 units starting from D20 is written into the system time, and the time data is updated to the system variable _sDateTime. The process is shown below.

	Element	Item	Clock Data		System variables	Item
Data for Clock Settings	D10	Year	2000~2099	————→	mYear	Year
	D11	Month	1~12	————→	mMouth	Month
	D12	Day	1~31	————→	mDay	Day
	D13	Hour	0~23	————→	mHour	Hour
	D14	Minute	0~59	————→	mMinute	Minute
	D15	Second	0~59	————→	mSecond	Second
	D16	Week	0~6		mWeekday	Week

3.26.4 TADD: Clock Addition Operation

Command table	TADD (S1) (S2) (D)			Applicable model	TS600 series		
16-Bit command	TADD: Clock Addition Operation						
32-Bit command	-						

Operand	Type	Bit			Word		Indexing	Constant
		X, Y, M, LM, T, C, S	Dx.y	Custom bit variable	D, R, V, Z, T, C	Custom word variable		
S1	WORD/ Array*3	-	-	-	√[1]	√	√	-
S2	WORD/ Array*3	-	-	-	√[1]	√	√	-
D	WORD/ Array*3	-	-	-	√[1]	√	√	-

Remark:

[1]Only the D, V, and R elements are supported.

Operand Description

S1: The source operand, which indicates clock data 1.

S2: The source operand, which indicates clock data 2.

D: The destination operand, which indicates the time result storage unit

Function Description

When the is command is driven, the data processed by the time addition operation is stored in the 3 storage units referred to by D. Depending on the processed result, the carry flag SM20 and the zero flag SM18 may be affected.

Precautions

- The time data involved in the calculation should conform to the following time formats:
  - The set range of "hour": 0~23
  - The set range of "minute": 0~59
  - The set range of "second": 0~59
- When any data does not meet the time format, the system prompts a command operand error, and this command is not executed.

Application Example

		Element Name	Data Type	Display Format	Current Value
1	...	D10	WORD	Decimal	23
2	...	D11	WORD	Decimal	59
3	...	D12	WORD	Decimal	59
4	...		WORD	Decimal	
5	...	D20	WORD	Decimal	23
6	...	D21	WORD	Decimal	58
7	...	D22	WORD	Decimal	58
8	...		WORD	Decimal	
9	...	D30	WORD	Decimal	23
10	...	D31	WORD	Decimal	58
11	...	D32	WORD	Decimal	57
12	...		WORD	Decimal	
13	...	SM18	BOOL	Binary	OFF
14	...	SM20	BOOL	Binary	ON

When M502 is ON, the 3 storage units starting from D10 are added to the 3 storage units starting from D20, and the processed results are stored in the 3 storage units starting from D30.

The carry flag (SM20) is set to ON, and the zero flag (SM18) is set to OFF. See below for the process.

S1			S2			D	
D10	23 hours	+	D20	23 hours	=	D30	23 hours
D11	59 minutes		D21	58 minutes		D31	58 minutes
D12	59 seconds		D22	58 seconds		D32	57 seconds

3.26.5 TSUB: Clock Subtraction Operation

Command table	TSUB (S1) (S2) (D)			Applicable model	TS600 series		
16-Bit command	TSUB: Clock Subtraction Operation						
32-Bit command	-						
Operand	Type	Bit			Word		Indexing	Constant
		X, Y, M, LM, T, C, S	Dx.y	Custom bit variable	D, R, V, Z, T, C	Custom word variable		
S1	WORD/ Array*3	-	-	-	√[1]	√	√	-
S2	WORD/ Array*3	-	-	-	√[1]	√	√	-
D	WORD/ Array*3	-	-	-	√[1]	√	√	-

Remark:

[1]Only the D, V, and R elements are supported.

Operand Description

S1: The source operand, which indicates clock data 1.

S2: The source operand, which indicates clock data 2.

D: The destination operand, which indicates the time result storage unit

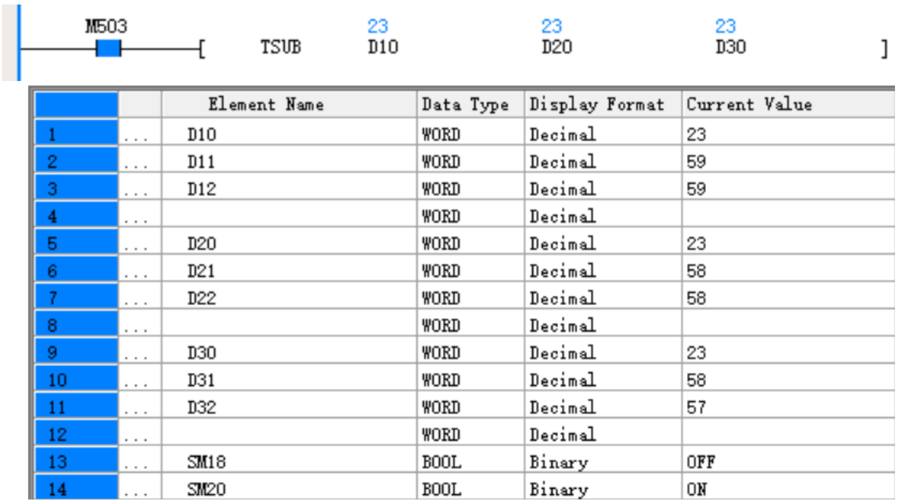
Function Description

When the is command is driven, the data processed by the time subtraction operation is stored in the 3 storage units referred to by D. Depending on the processed result, the carry flag SM20 and the zero flag SM18 may be affected.

Precautions

- The time data involved in the calculation should conform to the following time formats:
  - ◆ The set range of "hour": 0~23
  - ◆ The set range of "minute": 0~59
  - ◆ The set range of "second": 0~59
- When any data does not meet the time format, the system prompts a command operand error, and this command is not executed.

Application Example



		Element Name	Data Type	Display Format	Current Value
1	...	D10	WORD	Decimal	23
2	...	D11	WORD	Decimal	59
3	...	D12	WORD	Decimal	59
4	...		WORD	Decimal	
5	...	D20	WORD	Decimal	23
6	...	D21	WORD	Decimal	58
7	...	D22	WORD	Decimal	58
8	...		WORD	Decimal	
9	...	D30	WORD	Decimal	23
10	...	D31	WORD	Decimal	58
11	...	D32	WORD	Decimal	57
12	...		WORD	Decimal	
13	...	SM18	BOOL	Binary	OFF
14	...	SM20	BOOL	Binary	ON

When M503 is ON, the 3 storage units starting from D20 are subtracted from the 3 storage units starting from D10, and the processed results are stored in the 3 storage units starting from D30.

The borrow flag bit (SM19) is set to ON, and the zero flag bit (SM20) is set to OFF. See below for the process.

S1			**S2**			**D**	
D10	23 hours	-	D20	23 hours	=	D30	23 hours
D11	59 minutes		D21	59 minutes		D31	59 minutes
D12	58 seconds		D22	59 seconds		D32	59 seconds

3.26.6 HOUR: Hour Meter Commands

Command table		HOUR (S) (D1) (D2)				Applicable model	TS600 series		
16-Bit command		Hour: Hour Meter							
32-Bit command		-							
Operand	Type	Bit			Word			Indexing	Constant
		X, Y, M, LM, T, C, S	Dx.y	Custom bit variable	D, R, V, Z, T, C	Custom word variable			
S	WORD	-	-	-	√	√		√	√
D1	WORD/ Array*2	-	-	-	√[1]	√		√	
D2	BOOL	√[2]	-	√	-	-		-	-

Remark:

[1]Only the D, V, and R elements are supported.

[2]The X element is not supported.

Operand Description

S: The source operand, which indicates the hour comparison data. The data value ranges between 0 and 32767.

D1: Destination operand, which indicates a time storage unit, where the data unit of D1 the stores hour data, and the data unit of D1+1 stores the second data.

D2: Destination operand, which indicates an alarm output address. When the data of D1 is greater than or equal to the data specified by S, the alarm point becomes ON output.

Function Description

This command is used to record the accumulated time when the energy flow is ON. When the set time is reached, the corresponding alarm point output of this command is valid.

Precautions

- The set value of D1 ranges between 1 and 32767, in hours. D1+1 represents the current time value less than 1 hour, which ranges between 0 and 3599, in seconds.
- To still use the current data after cutting off the power supply of the PLC, specify D1 as the holding soft element unit against power outage. If an ordinary soft element is used, the current data will be cleared when the power supply of the PLC is cut off or when the RUN → STOP operation is performed.
- Even if the alarm output D2 is ON, the hour meter can still continue counting.
- In this command, hour is 16-bit integer data. When the hour data is greater than 32767, counting starts from 0 again.
- Up to 128 HOUR commands are supported.

Application Example

		Element Name	Data Type	Display Format	Current Value
1	...	D100	WORD	Decimal	1000
2	...	D200	WORD	Decimal	0
3	...	D201	WORD	Decimal	182

When M504 is ON, HOUR performs time accumulation on the input contact.

When the accumulated time of the ON state of M1 is greater than or equal to 1000, M10 is in the ON state.

3.26.7 DCMP (=, <, >, <>, >=, <=): Date Comparison Commands

Command table	DCMP (S1) (S2) (D)	Applicable model	TS600 series
16-Bit command	DCMP=: Date Comparison Equal To		
32-Bit command	-		
16-Bit command	DCMP>: Date Comparison Greater Than		
32-Bit command	-		
16-Bit command	DCMP<: Date Comparison Less Than		
32-Bit command	-		
16-Bit command	DCMP<>: Date Comparison Not Equal To		
32-Bit command	-		
16-Bit command	DCMP>=: Date Comparison Greater Than or Equal To		
32-Bit command	-		
16-Bit command	DCMP<=: Date Comparison Less Than or Equal To		
32-Bit command	-		

Operand	Type	Bit			Word		Indexing	Constant
		X, Y, M, LM, T, C, S	Dx.y	Custom bit variable	D, R, V, Z, T, C	Custom word variable		
S1	INT, Array*3	-	-	-	√[1]	√	√	-
S2	INT, Array*3	-	-	-	√[1]	√	√	-
D	BOOL	√[2]	√	√	-	-	-	-

Remark:

[1]Only the D, V, and R elements are supported.

[2]The X element is not supported.

Operand Description

S1: The source operand, which indicates date comparison data 1, occupying the first 3 characters of the specified unit in S1. The data in these 3 units must conform to the Gregorian format, otherwise the system will report an operand error.

S2: The source operand, which indicates date comparison data 2, occupying the first 3 characters of the specified unit in S2. The data in these 3 units must conform to the Gregorian format, otherwise the system will report an operand error.

D: The destination operand, which indicates the comparison the output state. If the data meets the comparison conditions, D is set to ON; otherwise it is set to OFF.

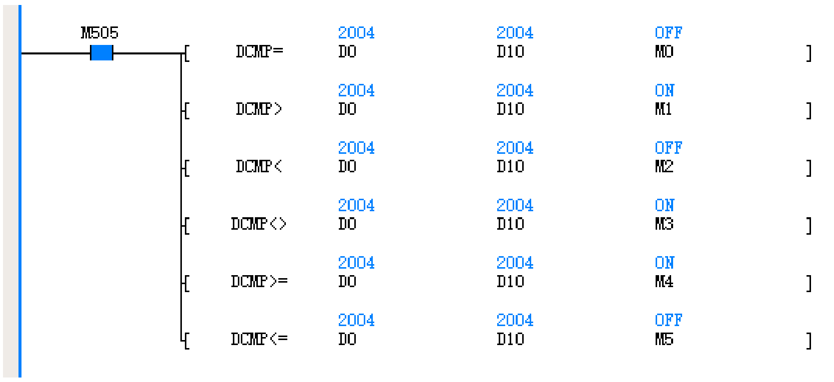
Function Description

When driven, this command performs a comparison between the date data respectively starting from the S1 and S2 units and assigns the comparison result to D.

Precautions

The date data starting from S1 and S2 must comply with the Gregorian calendar, otherwise an operand error (such as 2004-9-31 or 2003-2-29) will be reported, and this command will not be executed.

Application Example



		Element Name	Data Type	Display Format	Current Value
1	...	D0	WORD	Decimal	2004
2	...	D1	WORD	Decimal	10
3	...	D2	WORD	Decimal	25
4	...		WORD	Decimal	
5	...	D10	WORD	Decimal	2004
6	...	D11	WORD	Decimal	10
7	...	D12	WORD	Decimal	24

The command performs a BIN comparison between the date data respectively starting from the D0 and D10 units and assigns the comparison result to the destination data (M0, etc.).

3.26.8 TCMP (=, <, >, <>, >=, <=): Time Comparison Commands

Command table	TCMP (S1) (S2) (D)	Applicable model	TS600 series
16-Bit command	TCMP=: Time Comparison Equal To		
32-Bit command	-		
16-Bit command	TCMP>: Time Comparison Greater Than		
32-Bit command	-		
16-Bit command	TCMP<: Time Comparison Less Than		
32-Bit command	-		
16-Bit command	TCMP<>: Time Comparison Not Equal To		
32-Bit command	-		
16-Bit command	TCMP>=: Time Comparison Greater Than or Equal To		
32-Bit command	-		
16-Bit command	TCMP<=: Time Comparison Less Than or Equal To		
32-Bit command	-		

Operand	Type	Bit			Word		Indexing	Constant
		X, Y, M, LM, T, C, S	Dx.y	Custom bit variable	D, R, V, Z, T, C	Custom word variable		
S1	INT, Array*3	-	-	-	√[1]	√	√	-
S2	INT, Array*3	-	-	-	√[1]	√	√	-
D	BOOL	√[2]	√	√	-	-	-	-

Remark:

[1]Only the D, V, and R elements are supported.

[2]The X element is not supported.

Operand Description

S1: The source operand, which indicates time comparison data 1, occupying the first 3 characters of the specified unit in S1. The data in these 3 units must conform to the 24-hour time format, otherwise the system will report an operand error.

S2: The source operand, which indicates time comparison data 2, occupying the first 3 characters of the specified unit in S2. The data in these 3 units must conform to the 24-hour time format, otherwise the system will report an operand error.

D: The destination operand, which indicates the comparison the output state. If the data meets the comparison conditions, D is set to ON; otherwise it is set to OFF.

Function Description

When driven, this command performs a comparison between the time data respectively starting from the S1 and S2 units and assigns the comparison result to D.

Precautions

The time data starting from S1 and S2 must comply with the 24-hour system, otherwise an operand error (such as 24-10-31 or 13-59-60) will be reported, and this command will not be executed.

Application Example

		Element Name	Data Type	Display Format	Current Value
1	...	D0	WORD	Decimal	20
2	...	D1	WORD	Decimal	31
3	...	D2	WORD	Decimal	1
4	...		WORD	Decimal	
5	...	D10	WORD	Decimal	20
6	...	D11	WORD	Decimal	30
7	...	D12	WORD	Decimal	59

The command performs a comparison between the time data respectively starting from the D0 and D10 units and assigns the comparison result to the destination data (M0, etc.).

3.26.9 HTO*S: Commands for Conversion from Hours, Minutes, or Seconds to Word/Doubleword Second Data

Command table	HTOS (S) (D)			Applicable model	TS600 series		
16-Bit command	HTOS: Conversion from Hours, Minutes, or Seconds to Word Second Data						
32-Bit command	HTODS: Conversion from Hours, Minutes, or Seconds to Doubleword Second Data						

Operand	Type	Bit			Word		Indexing	Constant
		X, Y, M, LM, T, C, S	Dx.y	Custom bit variable	D, R, V, Z, T, C	Custom word variable		
S	WORD, Array*3	-	-	-	√[1]	√	√	-
D	WORD/ DWORD	-	-	-	√[2]	√	√	-

Remark:

[1]The Z element is not supported.

[2]The Z element is not supported; for the 32-bit command HTODS, the Z, T, and C elements are not supported.

Operand Description

S: The source operand, which indicates the starting number of the soft element storing the time data before conversion.

D: The destination operand, which indicates the number of the soft element storing the time data after conversion.

Function Description

1. HTOS command: When driven, this command converts the time data (hours, minutes, seconds) of [S, S+1, S+2] into seconds, and stores the result in D.

- Hour range: 0~18
- Minute range: 0~59
- Second range: 0~59

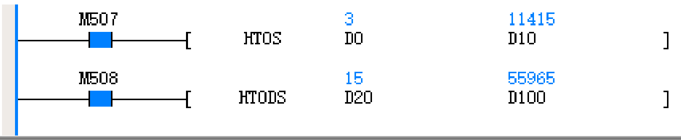
2. DHTOS command: When driven, this command converts the time data (hours, minutes, and seconds) of [S, S+1, S+2] into seconds, and stores the result in [D, D+1].

- Hour range: 0~32767
- Minute range: 0~59
- Second range: 0~59

Precautions

- If the operands (hour, minute, and second) of the HTOS/DHTOS command exceeds their respective ranges, the system will report an operand error, and this command will not be executed.
- If the HTOS conversion result is greater than 65535, the system will report an operand error, and this command will not be executed.

Application Example



		Element Name	Data Type	Display Format	Current Value
1	...	D0	WORD	Decimal	3
2	...	D1	WORD	Decimal	10
3	...	D2	WORD	Decimal	15
4	...		WORD	Decimal	
5	...	D10	WORD	Decimal	11415
6	...		WORD	Decimal	
7	...	D20	WORD	Decimal	15
8	...	D21	WORD	Decimal	32
9	...	D22	WORD	Decimal	45
10	...		WORD	Decimal	
11	...	D100	WORD	Decimal	55965

In case of M507=ON, this command converts the time data of the hours, minutes, and seconds starting from D0 unit to word seconds, and stores the results in D10. In case of D0=3, D1=10, and D2=15, D10=11415 is obtained.

In case of M508=ON, this command converts the time data of the hours, minutes, and seconds starting from D20 unit to doubleword seconds, and stores the results in (D100, D101). In case of D20=15, D21=32, and D22=45, (D100, D101)=55965 is obtained.

3.26.10 *STOH: Commands for Conversion from Word/Doubleword Second Data to Hours, Minutes, or Seconds

Command table	STOH (S) (D)			Applicable model	TS600 series		
16-Bit command	STOH: Conversion from Word Second Data to Hours, Minutes, or Seconds						
32-Bit command	DSTOH Conversion from Doubleword Second Data to Hours, Minutes, or Seconds						

Operand	Type	Bit			Word		Indexing	Constant
		X, Y, M, LM, T, C, S	Dx.y	Custom bit variable	D, R, V, Z, T, C	Custom word variable		
S	WORD, / DWORD	-	-	-	√[1]	√	√	-
D	WORD, Array*3	-	-	-	√[2]	√	√	-

Remark:

[1]The Z element is not supported; for the 32-bit command HTODS, the Z, T, and C elements are not supported.

[2]The Z element is not supported.

Operand Description

S: The source operand, which indicates the starting number of the soft element storing the time data before conversion.

D: The destination operand, which indicates the number of the soft element storing the time data after conversion.

Function Description

1. STOH command: When driven, this command converts the second data of S into hours, minutes, and seconds, and stores the results in [D, D+1, D+2].

- The value range of S is 0~65535
- Hour range: 0~18
- Minute range: 0~59
- Second range: 0~59

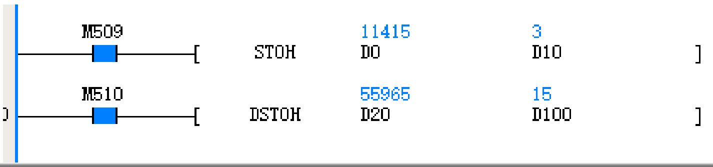
2. DSTOH command: When driven, this command converts the second data of [S, S+1] into hours, minutes, and seconds, and stores the results (hours, minutes, and seconds) in [D, D+1, D+2].

- The value range of [S, S+1] is 0~235929599
- Hour range: 0~32767
- Minute range: 0~59
- Second range: 0~59

Precautions

- If the hours, minutes, and seconds after conversion by the HTOS/DHTOS command exceeds their respective ranges, the system will report an operand error, and this command will not be executed.
- If the second data to be converted by the STOH/DSTOH command has exceeded the upper limit (235929599), the system will report an operand error.

Application Example



		Element Name	Data Type	Display Format	Current Value
1	...	D0	WORD	Decimal	11415
2	...		WORD	Decimal	
3	...	D10	WORD	Decimal	3
4	...	D11	WORD	Decimal	10
5	...	D12	WORD	Decimal	15
6	...		WORD	Decimal	
7	...	D20	WORD	Decimal	55965
8	...		WORD	Decimal	
9	...	D100	WORD	Decimal	15
10	...	D101	WORD	Decimal	32
11	...	D102	WORD	Decimal	45

In case of M509=ON, this command converts the word second data in D0 into hours, minutes, and seconds, and then stores the results in the 3 units starting from D10. In case of D0=11415, D10=3, D11=10, and D12=15 are obtained.

In case of M510=ON, the command converts the second data in (D20, D21) into hours, minutes, and seconds, and then stores the results in the 3 units starting from D100. In case of D20=55965, D100=15, D101=32, and D102=45 are obtained.

3.27 Control Calculation Command

3.27.1 Command Table

Command Category	Name	Function
Control Calculation Command	PID	PID Function Commands
	RAMP	Ramp signal output
	HACKLE	Hackled wave signal output
	TRIANGLE	Triangular wave signal output
	MSC	Multi-station control

3.27.2 PID: PID Control Commands

Command table		PID (S1) (S2) (S3) (S4) (D)			Applicable model	TS600 series		
16-Bit command		PID: PID Control						
32-Bit command		-						
Operand	Type	Bit			Word		Indexing	Constant
		X, Y, M, LM, T, C, S	Dx.y	Custom bit variable	D, R, V, Z, T, C	Custom word variable		
S1	INT	-	-	-	√[1]	√	√	-
S2	INT	-	-	-	√[1]	√	√	-
S3	INT	-	-	-	√[1]	√	√	√
S4	INT/ Array*number	-	-	-	√[1]	√	√	-
D	INT	-	-	-	√[1]	√	√	-

Remark:

[1] Only the D and R elements are supported.

Operand Description

S1: The source operand, which indicates the set target value in units of 0.1 (set unit).

S2: The source operand, which indicates the current measurement value in units of 0.1 (set unit).

S3: The source operand, which indicates th PID mode option that currently supports 0~1.

S4: The source operand, which indicates a PID parameter.

D: The destination operand, which indicates an analog output percentage in units of 0.1%.

Mode 0: Positional PID Mode

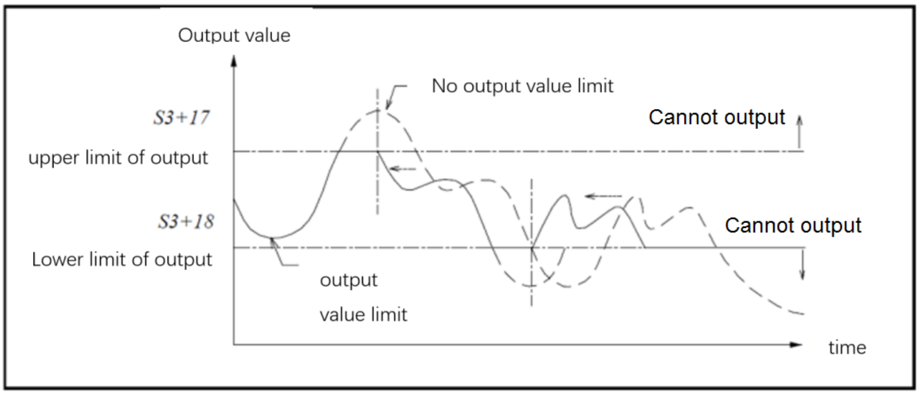
S4: The setting of parameter required for PID operation.

Address	Name	Setting range	Meaning
S4	Proportional gain (KP)	0.0~32767.0 (%)	Proportional gain (0 by default), which is input as an floating-point number by 2 bytes
S4+2	Integral time (TI)	0.0～32767.0 (s)	Integral time (0 by default), which is input as an floating-point number with 2 bytes
S4+4	Differential time (TD)	0.0～32767.0 (s)	Differential time (0 by default), which is input as an floating-point number with 2 bytes
S4+6	Sampling time (TS)	1~32767 (ms)	It is set to range between 1 and 32767, which requires that the PID operation cycle should be greater than the PLC program scan cycle (100 by default)

Address	Name	Setting range	Meaning
S4+7	Operation mode	0~1	0: backward PID (cooling); 1: forward PID (heating)
S4+8	Deadband width	0~32767	0: invalid (default); non 0: the deviation is considered as 0 if it is less than this value
S4+9	Integral separation width	0~32767	When the deviation exceeds this value, there is no integration effect; if this value is 0 (0 by default), integral separation will not be used.
S4+10	Upper limit of input	-32768~32767 (0.1 as input unit)	Maximum value of input
S4+11	Lower limit of input	-32768~32767 (0.1 as input unit)	Minimum value of input
S4+12	Upper limit of output	-32768~32767 (0.1 as output unit)	Maximum value of output
S4+13	Lower limit of output	-32768~32767 (0.1 as output unit)	Minimum value of output
S4+14	Feedforward input	-32768~32767	It is used to compensate for PID input values

Function Description

1. When the energy flow is valid and reaches the sampling time, this command performs PID operation. Proportional gain, integration time, differentiation time, sampling time, operation mode, upper/lower limit of input, and upper/lower limit of output are mandatory options. The upper and lower limits of the input and output are set according to the actual situation. For example, in a temperature control scenario, the upper and lower limits of input are the allowable upper and lower limits of the temperature, and the upper and lower limits of the output are the upper and lower limits of the output of the control relay (220V/0V).
2. The PID commands can be executed multiple times simultaneously (up to 128 commands at the same time), but it is important to note that the numbers and regions of the soft elements used by the S1, S2, S3, S4, and D operands used in the operation should not be duplicated or overwritten, otherwise unpredictable errors will occur in the PID commands.
3. PID commands can be used in timed interrupt subroutines, general subroutines, and main programs.
4. The differential gain works for mitigating drastic changes in output values.
5. The deviation deadband can be used to avoid excessively frequent control and eliminate oscillations caused by frequent actions.
6. The integral separation prevents system overshoots or even oscillations during program startup or significant increase or decrease of target values.
7. The upper and lower limits of the output suppress excessively large integral terms in PID control. See the figure below for the output value.



8. The basic operation formula of PID command:

$$u(k) = Kp * e(k) + Ki * T * \sum e(i) + (Kd/T) * [e(k) - e(k-1)]$$

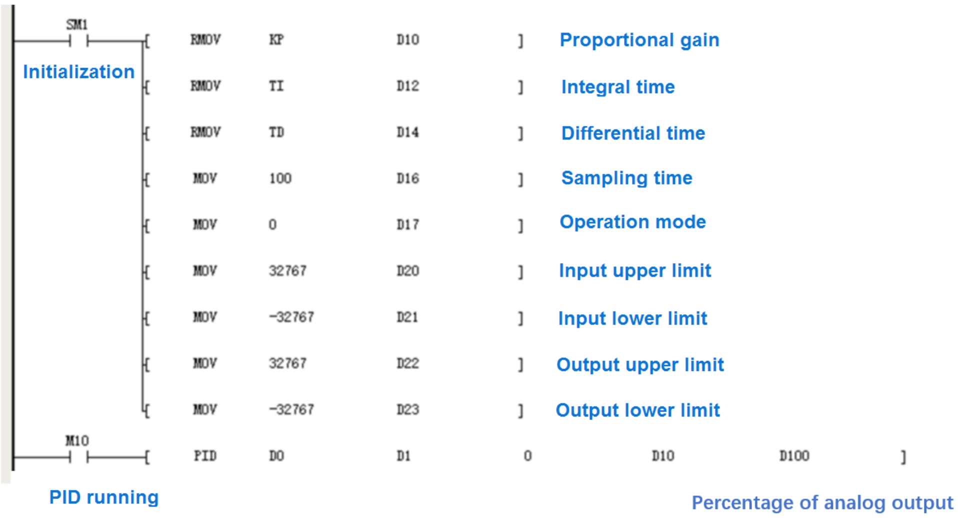
See the table below for the symbol description:

Symbol	Description	Symbol	Description
u(k)	Current output value	e(k-1)	Deviation value from the previous moment
e(k)	Current deviation	T	Sampling period
Σ e(i)	Current cumulative integration	Kp	Proportional gain
Sv(k)	Current set value	Ki = Kp / Ti	Integral gain
Pv(k)	Current feedback value	Kd = Kp * Td	Differential gain

Precautions

- For D, please specify the data register region to be held during the non-shutdown period. When specifying the data register region to be held during the shutdown period, always make sure to reset it to 0 (LD SM0 MOV 0 D****) at the time of first run.
- The positional PID command needs to occupy 15 data registers starting from S4.
- The maximum error for the sampling time TS is [- (1 scan cycle + 1ms), + (1 scan cycle)]. When TS is small, it can affect the PID effect. It is recommended to use the PID command in the timing interrupt for best results.
- When the upper and lower limits of PID input/output are set to be effective, if the upper limit value is smaller than the lower limit value, the system will swap the upper and lower limit values and issue an error warning, and the command will continue to run.
- When the set operand of PID is not within the valid range, the system reports operand error and the PID command operation is not carried out.
- When the sampling time is less than or equal to 1 scan cycle, the default sampling time is 1 scan cycle. If data overflow occurs during the calculation process, a warning will appear, and the PID command calculation will continue to execute.
- Before executing the PID command, it is necessary to initialize each operand. If the operands remain unchanged during the runtime, the control operand unit will not be overridden by other programs, and the initialization program can be executed only once.
- If the PID input exceeds the upper/lower limit of the input range, the PID output will be 0.
- In the same project, only 128 PID commands can be called.

Application Example



When the main module starts running the first scan cycle, it initializes the PID configuration parameters. After that, the PID operands are no longer initialized in the subsequent scan cycles. The PID parameters are input starting from D10. When M10=ON, the current measurement value is read from an external A/D module (in actual applications, this can be done in other ways) and filled into the measurement value unit. Execute PID calculation at this point, and D100 outputs analog percentage.

Mode 1: Temperature control self-tuning PID mode

S4: The setting of parameter required for PID operation.

Address	Name	Setting range	Meaning
S4	Proportional gain (Kp)	1.0~32767.0(%)	Proportional gain, two-byte floating point input
S4+2	Integral time (TI)	0.0~32767.0(s)	Integral time, 0 = no integral processing, two-byte floating point input
S4+4	Differential time (TD)	0.0~32767.0(s)	Differential time, 0 = no derivative processing, two-byte floating point input
S4+6	Sampling time (TS)	1~32767 (ms)	PID calculation cycle should be greater than the PLC program scan cycle and less than the relay control cycle (default 100)
S4+7	Relay on/off cycle	1~32767 (ms)	The on-off control cycle time of the relay (default 2000)
S4+8	Upper temperature limit	32768~32767(0.1°C)	The maximum input temperature value, which is greater than the input temperature lower limit (default 1000)
S4+9	Temperature lower limit	32768~32767(0.1°C)	The minimum input temperature value, which is less than the input temperature upper limit (default -50)
S4+10	Function mode	0~3	0: default PID (default); 1: self-tuning PID; 2: Manual PID; 3: ON/OFF mode
S4+11	Self-tuning configuration parameters	-	bit0: 0: Limit cycle; 1: ascending curve bit1-bit7: reserved
S4+12	Autotuning coefficient	1-10	The larger the parameter, the faster the adjustment, but the overshoot increases, default is 5
S4+13	Stop heating function	0~1	0: heating (default); 1: stop heating
S4+14	-	-	-
S4+15	DO	0~1	DO output status 1: heating on; 0: heating off
S4+16	Tuning proportional gain	1~32767(s)	Proportional gain obtained by tuning
S4+18	Tuning integral time	0~32767(s)	Integral time obtained by tuning
S4+20	Tuning derivative time	0~32767(s)	Derivative time obtained by tuning

Function Description

1. Temperature control PID command can be effectively executed when the power flow is available.
2. The temperature control PID command needs to occupy 21 data registers starting from S4. If self-tuning PID is not used, tuning proportional gain, integral time, derivative time, sampling time, control cycle, temperature upper/lower limit, and function mode are required fields. If self-tuning PID is used, sampling time, control cycle, temperature upper/lower limit, function mode, self-tuning configuration parameters, and self-tuning coefficients are required fields.

3. Temperature control PID command control modes include: default manual PID control mode, self-tuning PID control mode, manual PID control mode, ON/OFF control mode.
4. Default manual PID control mode: It is the default mode of the command, in which a set of default PID initial parameters are given internally, and you just need to set the target temperature, real-time temperature, temperature upper and lower limits, and the control cycle time. The command will have a certain degree of overshooting for the first temperature rise, and then the temperature will be stabilized at the target temperature after a few minutes. The mode time is relatively fast.
5. Self-tuning temperature control PID control mode:

Limit cycle method>: This method is relatively stable in tuning, but the tuning period is longer. When the user selects this control mode, they need to set self-tuning configuration parameter bit0=0 and enable the power flow. The command will automatically tune a set of suitable control parameters based on the control object during the heating process. After the self-tuning mode is completed, the system will output the self-tuning parameters and automatically adjust them. The next time it is used, it needs to be switched to manual mode and input the self-tuning parameters for control. There may be a small temperature overshoot, but the error will be within ±1°C after stabilization. The initial tuning time is relatively long, and the self-tuning effect is best in a normal temperature environment. It is suitable for occasions with high temperature requirements. The self-tuning sampling time can be set to 100ms. If using relay control, the control cycle can be set to 2000ms. The example curve for self-tuning temperature of 60°C and target temperature of 90°C is as follows.

The condition for self-tuning is that the internal self-tuning temperature of the system is 60°C.

A: When the target temperature is greater than the actual temperature and the target temperature is above 90°C, use (target temperature - 30°C) for self-tuning.

B: When the target temperature is greater than the actual temperature and the target temperature is above 60°C but below 90°C, use 60°C for self-tuning.

C: When the target temperature is greater than the actual temperature and the target temperature is below 60°C, use target temperature for self-tuning.

Note: The system will not self-tune if the actual temperature is higher than the target temperature or higher than the self-tuning temperature. Self-tuning can only be performed when the actual temperature is lower than both the target temperature and the self-tuning temperature.

Ascending Curve Method: This tuning method has a certain probability of failure, but it has a shorter tuning period. When the user selects this control mode, they need to set the self-tuning configuration parameter bit0=1 and enable the power flow. To use this mode for self-tuning, the temperature should be set to at least 60°C higher than the current temperature. If self-tuning fails, the self-tuning coefficient can be reduced or the set temperature can be increased. This command will automatically tune a set of suitable control parameters for the heating process of the controlled object. After the self-tuning mode is completed, the system will output the self-tuning parameters and automatically adjust them. The next time it is used, it needs to be switched to manual mode and the self-tuning parameters need to be entered for control. There may be a small temperature overshoot, but the error will be within ±1°C after stabilization. The initial tuning time is relatively long, and the self-tuning effect is best in a normal temperature environment. It is suitable for occasions with high temperature requirements. The self-tuning sampling time can be set to 100ms. If using relay control, the control cycle can be set to 2000ms.

6. Manual PID control mode: This control mode requires the user to manually enter the PID parameters, proportional gain, integral time constant and differential time constant, and the PID output is adjusted according to the adjustment equation. The PID adjustment equation is as follows:

$$u(k) = K_p\left[e(k) + \frac{T}{T_i}\sum_{j=0}^{k} e(j) + \frac{T_d}{T}[e(k) - e(k-1)]\right]$$

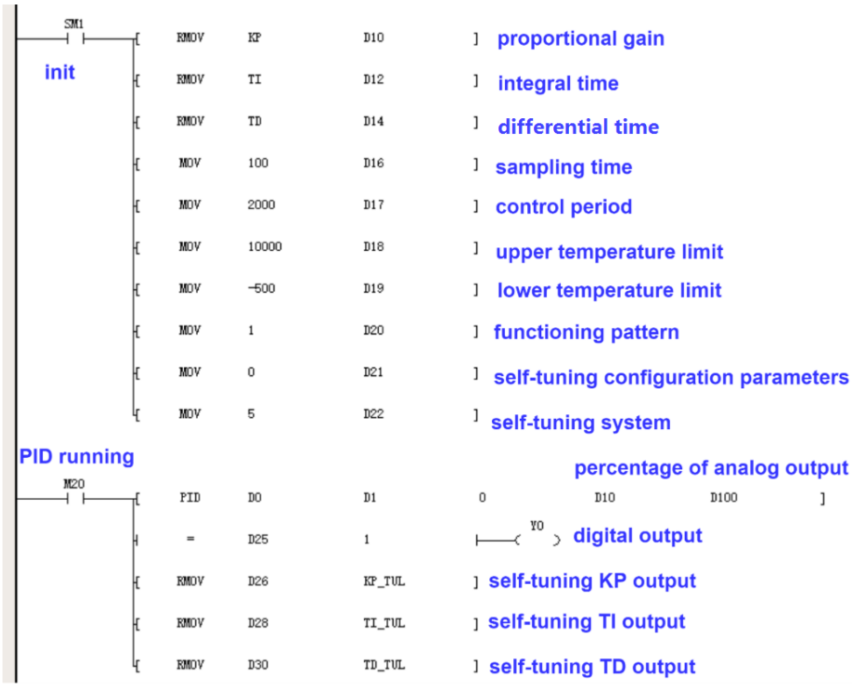
In the equation, e(k) represents the error at the current time, e(k-1) represents the error at the previous time, Kp represents the proportional gain, Ti represents the integral time, Td represents the derivative time, and T represents the sampling period.

7. ON/OFF control mode: This control mode is a On/Off control. When the temperature exceeds the target temperature, the relay is disconnected. When the temperature is below the target temperature, the relay is opened at a certain control cycle ratio. After reaching stability, there will still be some errors in this control mode, so it is suitable for situations where high accuracy is not required.
8. Before executing the above control modes, the user needs to confirm whether the sampling period, control cycle, temperature upper and lower limits, and control mode are suitable for the specific heating requirements of the controlled object. If they are not suitable, please modify them before heating. The sampling period should be set according to the heating object, generally around 100 ms. The temperature system changes slowly, so the scanning period should not be too small. Considering the service life of the relay, the control cycle should be greater than or equal to 2000 ms.

Precautions

- The temperature control PID command needs to set the control parameter of the software element to the starting address number of 21 data registers.
- Due to the slow temperature changes, the temperature sampling time should not be too small. When using a small time value, it will affect the effectiveness of the temperature control PID, it should not be too large either, and it should be less than the relay switch cycle time.
- Before the first execution of the temperature control PID command, each operand needs to be initialized. If the operands change during the running process, the underlying data of the temperature control PID will be updated in the next cycle.
- During the self-tuning mode, if the power flow is disconnected or switched to other functional modes, the self-tuning parameters will be reset. After reconnecting the power flow or switching back to the self-tuning mode, the PID parameters will be self-tuned again.

Application Example



During the initialization process, configure the PID parameters starting from D10. When M20=ON, perform PID calculation, and D100 outputs the analog percentage. The analog signal here is suitable for a 220V solid-state relay.

3.27.3 RAMP: Ramp signal output command

Command table	RAMP (S1) (S2) (D1) (S3) (D2)	Applicable model	TS600 series
16-Bit command	RAMP signal output command		
32-Bit command	-		

Operand	Type	Bit			Word		Indexing	Constant
		X, Y, M, LM, T, C, S	Dx.y	Custom bit variable	D, R, V, Z, T, C	Custom word variable		
S1	INT	-	-	-	√	√	√	√
S2	INT	-	-	-	√	√	√	√
D1	INT	-	-	-	√[1]	√	√	
S3	INT	-	-	-	√	√	√	√
D2	BOOL	√[2]	-	-	-	-	-	-

Remark:

[1]Only the D, V, and R elements are supported.

[2]The X element is not supported.

Operand Description

S1: Source operand, starting value.

S2: Source operand, ending value.

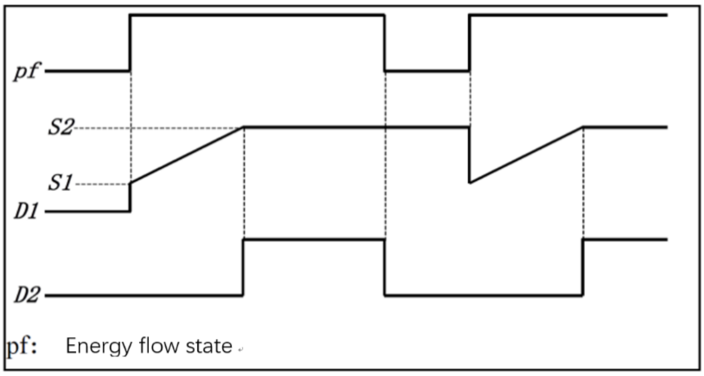
D1: Destination operand, output current value.

S3: Source operand, step number (S3>0, otherwise, an operand error is reported and the operation is not executed).

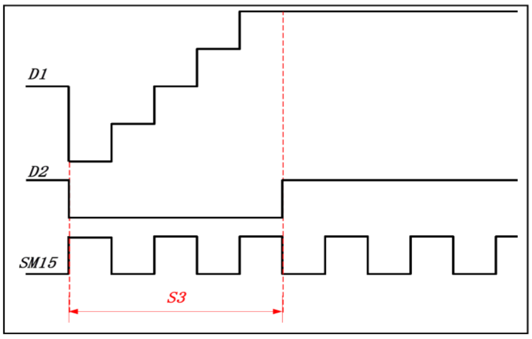
D2: Destination operand, output status.

Function Description

1. When the power flow has a rising edge and remains ON, perform linear interpolation every scanning cycle to determine the increment and current output value. After reaching S2, the output value (D1) remains in the current state, and the output status is set to ON.
2. If the power flow has a falling edge, the output status (D2) is set to OFF, and the output value (D1) remains in the current state until the power flow has a rising edge again. Then, the output value (D1) is initialized to the value of S1, and the next ramp calculation is continued, as shown in the following figure.



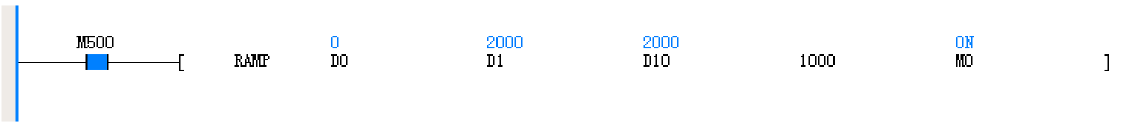
The execution process of the ramp command is decomposed as shown in the following figure (S3=5).



Precautions

- If this command is executed in the normal main cycle, to ensure linear interpolation of the output, the program execution needs to be set to a fixed scanning mode.
- If the step number is less than 0, the system will report an operand error and the command will not be executed.
- Users can convert data into analog waveforms through external special modules.
- The command will only generate one set of ramp data each time a rising edge occurs.
- When S1=S2, D1=S2 and D2=ON.
- The total number of RAMP, HACKLE, and TRIANGLE commands in the program should not exceed 100.

Application Example



When M500 is valid, interpolate from 0 to 2000 for 1000 operands and display the output result in D10. When the output value D10=endpoint value and D1=2000, set the output status M0 to ON. If a falling edge occurs in the input flow, set the output status M0 to OFF, keep the current data of output value D10 until the next rising edge occurs, and set the output value D10=initial value D0=0, then start a new ramp process.

3.27.4 HACKLE: Sawtooth Wave Signal Output Command

Command table	HACKLE (S1) (S2) (D1) (S3) (D2)	Applicable model	TS600 series
16-Bit command	HACKLE Sawtooth Wave Signal Output		
32-Bit command	-		

Operand	Type	Bit			Word		Indexing	Constant
		X, Y, M, LM, T, C, S	Dx.y	Custom bit variable	D, R, V, Z, T, C	Custom word variable		
S1	INT	-	-	-	√	√	√	√
S2	INT	-	-	-	√	√	√	√
D1	INT	-	-	-	√[1]	√	√	-
S3	INT	-	-	-	√	√	√	√
D2	BOOL	√[2]	-	-	-	-	-	-

Remark:

[1]Only the D, V, and R elements are supported.

[2]The X element is not supported.

Operand Description

S1: Source operand, starting value.

S2: Source operand, ending value.

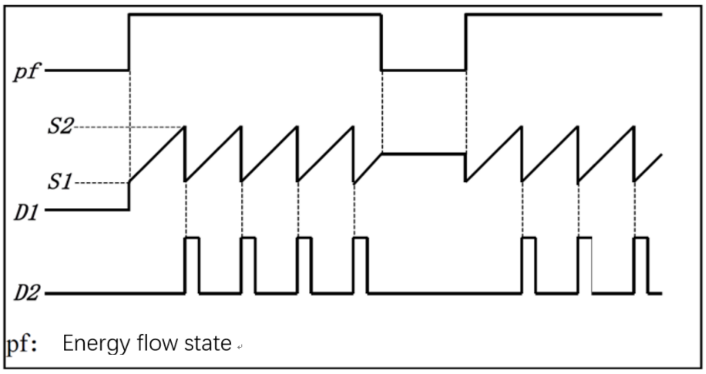
D1: Destination operand, output current value.

S3: Source operand, step number (S3>0, otherwise, an operand error is reported and the operation is not executed).

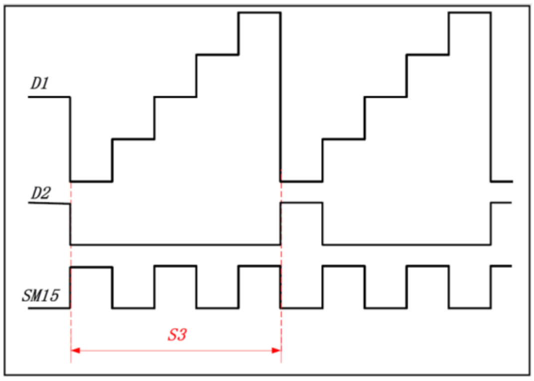
D2: Destination operand, output status.

Function Description

1. Perform linear interpolation every scanning cycle when power flow is effective. When the output value reaches S2, initialize it to the value of S1 and set the status output bit (D2) to ON. If the power flow continues to be ON in the next scanning cycle, set the status output bit (D2) to OFF and continue to generate the next sawtooth wave.
2. During the generation process of the sawtooth wave function, if there is a falling edge in the power flow, the output status (D2) is set to OFF, and the output value (D1) remains in the current state until the power flow appears rising edge again. At this point, the output value (D1) is initialized to the value of S1, and the next sawtooth wave operation continues. See the figure below.



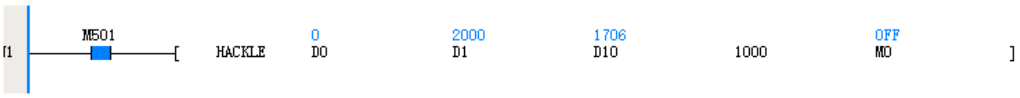
The execution process of the sawtooth wave command is decomposed as shown in the following figure (S3=5).



Precautions

- If this command is executed in the normal main cycle, to ensure linear interpolation of the output, the program execution needs to be set to a fixed scanning mode.
- If the step number is less than 0, the system will report an operand error and the command will not be executed.
- Users can convert data into analog waveforms through external special modules.
- As long as the power flow is valid, the command will generate a series of continuous sawtooth wave data.
- When S1=S2, D1=S2 and D2=ON (no count pulse is generated).
- The total number of RAMP, HACKLE, and TRIANGLE commands in the program should not exceed 100.

Application Example



When M501 is valid, interpolate from 0 to 2000 for 1000 operands and display the output result in D10. When the output value D10=endpoint value and D1=2000, set the output status M0 to ON. In the next scanning cycle, if X0 remains ON, the output value D10 is initialized to the initial value D0=0, and at the same time, the output status M0 is set to OFF, starting the next sawtooth wave generation process.

If there is a falling edge in the power flow during the operation, set the output status M0 to OFF, and the output value D10 remains in the current data until the power flow appears rising edge again. At this point, the output value D10 is initialized to the initial value D0=0, and a new sawtooth wave generation process starts again.

3.27.5 TRIANGLE: Triangle wave signal output command

Command table	TRIANGLE (S1) (S2) (D1) (S3) (D2)	Applicable model	TS600 series
16-Bit command	HACKLE Sawtooth Wave Signal Output		
32-Bit command	-		

Operand	Type	Bit			Word		Indexing	Constant
		X, Y, M, LM, T, C, S	Dx.y	Custom bit variable	D, R, V, Z, T, C	Custom word variable		
S1	INT	-	-	-	√	√	√	√
S2	INT	-	-	-	√	√	√	√
D1	INT	-	-	-	√[1]	√	√	-
S3	INT	-	-	-	√	√	√	√
D2	BOOL	√[2]	-	-	-	-	-	-

Remark:

[1]Only the D, V, and R elements are supported.

[2]The X element is not supported.

Operand Description

S1: Source operand, starting value.

S2: Source operand, ending value.

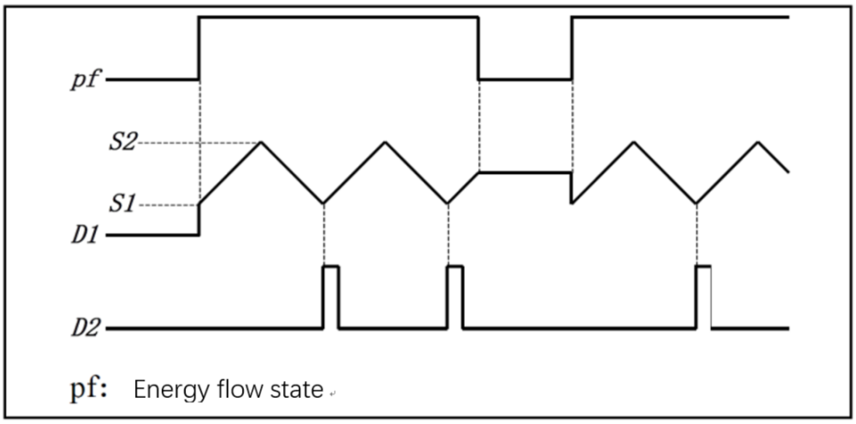
D1: Destination operand, output current value.

S3: Source operand, step number (S3>0, otherwise, an operand error is reported and the operation is not executed).

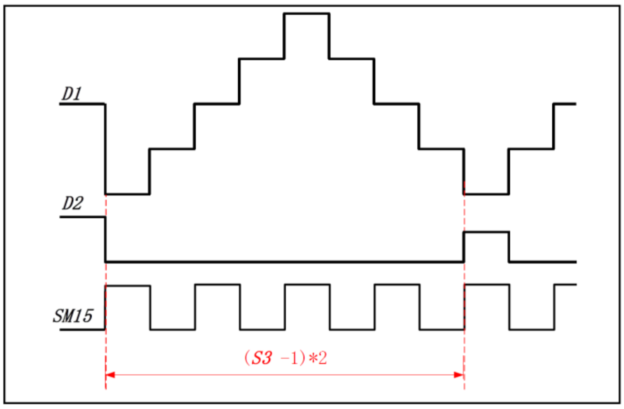
D2: Destination operand, output status.

Function Description

Perform linear interpolation every scanning cycle when power flow is effective. When the output value reaches S2, the first half ramp of the triangular wave is completed, and the increment direction of the output value is changed to continue generating the second half ramp. When the output value (D1) reaches the S1 value again, the status output bit (D2) is turned ON. If the power flow continues to be ON in the next scanning cycle, set the status output bit (D2) to OFF and continue to generate the next triangle wave. During the generation process of the triangular wave function, if there is a falling edge in the power flow, the output status (D2) is set to OFF, and the output value (D1) remains in the current state until the power flow appears rising edge again. At this point, the output value (D1) is initialized to the value of S1, and a new triangular wave generation process starts again, as shown in the following figure.



The execution process of the triangle wave command is decomposed as shown in the following figure (S3=5).

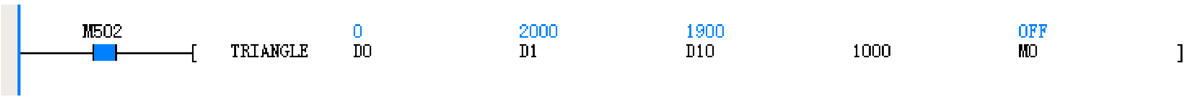


Precautions

- If this command is executed in the normal main cycle, to ensure linear interpolation of the output, the program execution needs to be set to a fixed scanning mode.
- If the step number is less than 0, the system will report an operand error and the command will not be executed.
- Users can convert data into analog waveforms through external special modules.

- As long as the power flow is valid, the command will generate a series of continuous triangle wave data.
- When S1=S2, D1=S2 and D2=ON (no count pulse is generated).
- The cycle of the triangle wave = (S3-1) x 2.
- The total number of RAMP, HACKLE, and TRIANGLE commands in the program should not exceed 100.

Application Example



When M502=ON, interpolate from 0 to 2000 for 1000 operands and display the output result in D10. When the output value D10=ending value D1=2000, the half wave of the triangular wave is completed. Afterwards, interpolate 1000 times from 2000 to 0. When the output value D1=initial value D0, the complete triangular wave is generated, and the output status M0 is set to ON. In the next scan cycle, if X0 is kept ON, the output status M0 is set OFF and the next triangle wave process starts.

If there is a falling edge during the operation, set the output status M0 to OFF, and the output value M10 remains in the current data until the power flow appears rising edge again. At this point, the output value D10 is initialized to the initial value D0=0, and a triangle wave generation process starts again.

3.27.6 MSC: Multi-station control command

Command table	MSC (S1) (S2) (S3) (S4) (D5) (D6)	Applicable model	TS600 series
16-Bit command	MSC Multi-station control		
32-Bit command	-		

Operand	Type	Bit			Word		Indexing	Constant
		X, Y, M, LM, T, C, S	Dx.y	Custom bit variable	D, R, V, Z, T, C	Custom word variable		
S1	BOOL	√[1]	-	-	-	-	-	-
S2	DWORD	-	-	-	√[2]	√	-	-
S3	WORD	-	-	-	√[2]	√	-	-
S4	DWORD	-	-	-	√[2]	√	-	-
S5	DWORD	-	-	-	√[2]	√	-	-
S6	WORD	-	-	-	√[2]	√	-	-
D7	BOOL	√[3]	-	√	-	√	-	-
D8	DWORD	-	-	-	√[2]	√	-	-
D9	DWORD	-	-	-	√[2]	√	-	-

Remark:

[1]Only X elements are supported.

[2]Only D, R elements are supported.

[3]Only Y, M elements are supported.

Operand Description

S1: To initiate an command at the trigger input point, choose an external interrupt input point or a general input point. The command is triggered on both rising and falling edges, capturing the value of S2.

S2: Compare the data source.

S3: Occupy two consecutive 16-bit registers continuously, used for setting the station count n (1~100) and the workpiece count m (1~50). It is recommended to use non-volatile holding registers.

S4, S5: These represent the entry and exit values of workpieces, occupying n consecutive 32-bit registers (double words). Each parameter utilizes two consecutive registers, and it is advisable to employ non-volatile holding registers to preserve data in case of power loss. When the entry comparison value for a particular station exceeds the exit comparison value (S4 > S5), it indicates that the comparison action for that station is not executed. The specific register address allocation is as follows:

Parameter Name	Station 1	Station 2	...	Station n
Station Entry Comparison Value	S4	S4+2	...	S4+(n-1)*2
Station Exit Comparison Value:	S5	S5+2	...	S5+(n-1)*2

S6: Length of workpieces to be filtered, ranging from 0 to 1000.

D7: Occupies n consecutive coils (corresponding to the number of stations). Only Y and M coils can be specified for output. Used to determine whether the corresponding workpiece has entered or left the station. During command execution, each station will use the sequential index to determine if the corresponding workpiece has entered or left the station based on the configured comparison values. When the real-time count value of the corresponding workpiece is ≥ the entry comparison value, the corresponding output point is set to ON. When the real-time count value of the corresponding workpiece is ≥ the exit comparison value, the corresponding output point is set to OFF.

Assuming the trigger input point S1 is X0 and the comparison output component D6 is Y1, the following explains the roles of S3 and S3+1: the entry value and exit value of the workpiece are relative to the station position with respect to X0. For example, if the entry value for Station 1 is set to 1000 and the exit value is set to 1200, it means that when the workpiece passes through position X0 after the rising edge, after counting 1000 high-speed values, it enters Station 1, and at this point, Y1 is set to ON. It waits for the falling edge of the trigger input point X0 and then, after counting 1200 high-speed values, leaves Station 1, and at this point, Y1 is set to OFF.

Note: If the trigger input point X0 experiences multiple rising and falling edges and the high-speed count value has not reached 1000, the underlying system remembers a maximum of 50 workpieces for each station.

D8: Number of completed workpieces.

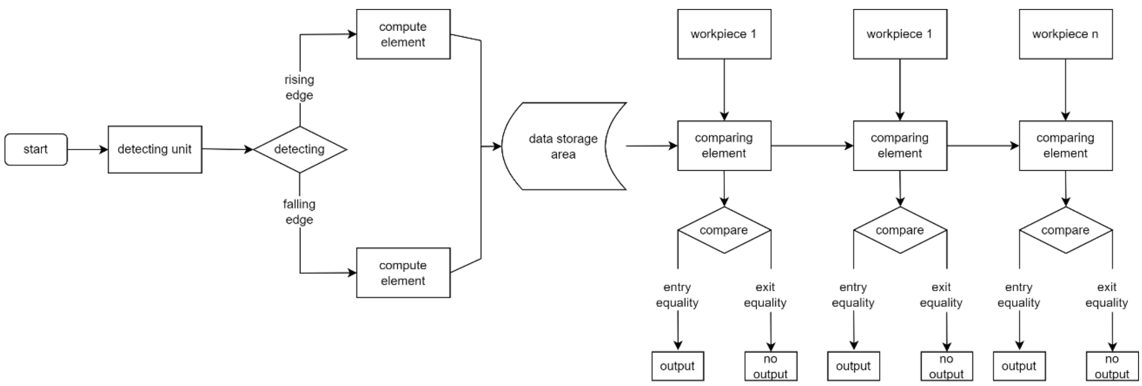
D9: Current length of the workpiece.

Precautions

- The MSC command can be invoked up to a maximum of 128 times.

Function Description

The basic flow of the command is illustrated in the following diagram.

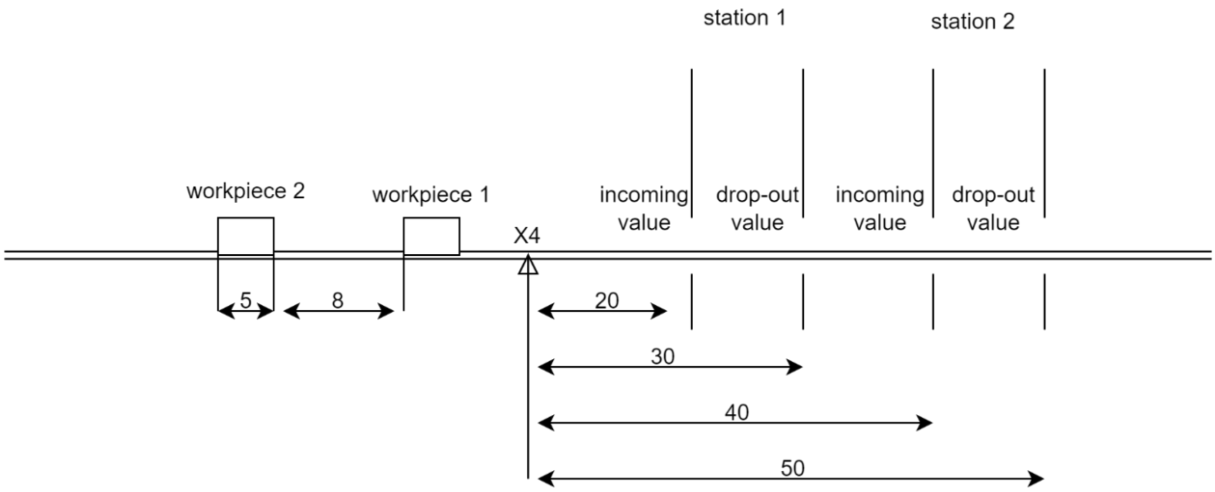


Precautions

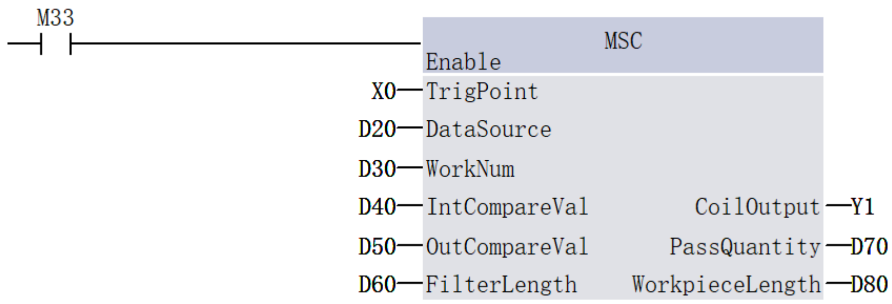
- Upon the disconnection and subsequent reconnection of the prerequisites for the MSC, all values within the D6 storage area will be cleared, and the output will be set to OFF.

Application Example

Assuming there are two workpieces that need to undergo processing at two stations, with the trigger input signal being X0 and the encoder signal input point being D20. The width of each workpiece is 5, and the distance between workpieces is 9.



The program is as follows:



Input Soft Elements:

Soft Element Address	Function Description
X0	Trigger input point = X0
D20	Comparative data source = D20
D30	Set number of workstations = 2
D31	Set number of workpieces = 2
D40	Station 1 preset entry value = 20
D42	Station 2 preset entry value = 40
D50	Station 1 preset exit value = 30
D52	Station 2 preset exit value = 50
D60	Filter workpiece length = 0

Output Results:

Assuming the high-speed count value when workpiece 1 triggers X0 rising edge is 11, the entry and exit comparison values for each station are as follows:

Parameter Name	Station 1	Station 2
Workpiece 1 Entry Comparison Value	31	51
Workpiece 1 Exit Comparison Value	46	66
Workpiece 2 Entry Comparison Value	45	65
Workpiece 2 Exit Comparison Value	60	80

3.28 Verification Command

3.28.1 Command Table

Command Category	Name	Function
Verification Command	CCITT	CCITT checksum calculation
	CRC16	CRC16 checksum calculation
	LRC	LRC checksum calculation
	CCD	CCD checksum calculation

3.28.2 CCITT: CCITT Checksum Calculation Command

Command table	CCITT (S) (n) (D)			Applicable model	TS600 series		
16-Bit command	CCITT Checksum Calculation						
32-Bit command	-						

Operand	Type	Bit			Word		Indexing	Constant
		X, Y, M, LM, T, C, S	Dx.y	Custom bit variable	D, R, V, Z, T, C	Custom word variable		
S	WORD, Array*n	-	-	-	√[1]	√	√	-
n	INT	-	-	-	√	√	√	√
D	WORD	-	-	-	√[1]	√	√	-

Remark:

[1]Only the D, V, and R elements are supported.

Operand Description

S: The operand of the source, representing the starting address of the data to be verified.

n: The quantity of data to be verified, in bytes, where $0 \leqslant n \leqslant 256$.

D: The operand of the destination, representing the address of the verification result software element.

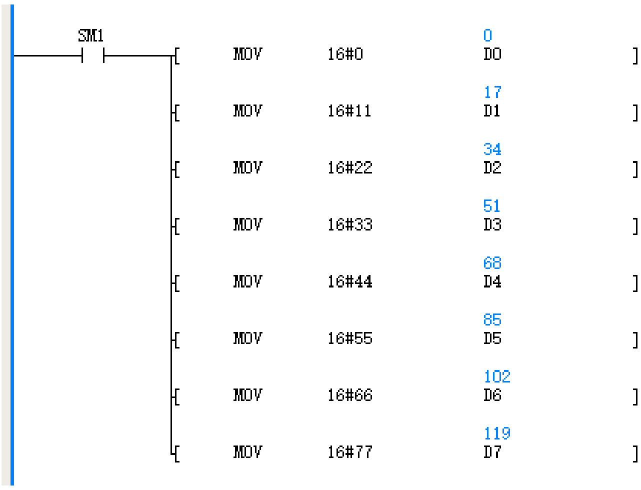
Function Description

1. Perform CCITT checksum operation on n bytes of data starting from the initial unit (S), and assign the result to unit D.
2. The polynomial for the CCITT checksum algorithm is: $X^{16} + X^{12} + X^5 + 1$.

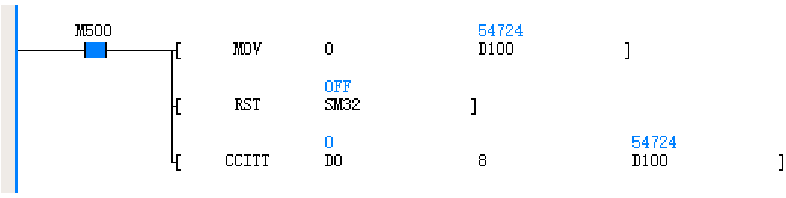
Precautions

- If the verification quantity n is less than 0 or greater than 256, the system reports an operand error, and the command is not executed.
- Each time the command is executed, the system substitutes the content of D before execution into the operation, so D must be initialized before execution.
- When SM32 is in the OFF state, the high 8 bits and low 8 bits of the verification data starting from unit S2 participate in the CCITT checksum operation together, with 16 bits as the unit.
- When SM32 is in the ON state, the low 8 bits of the verification data starting from unit S2 participate in the CCITT checksum operation with 8 bits as the unit.

Application Example

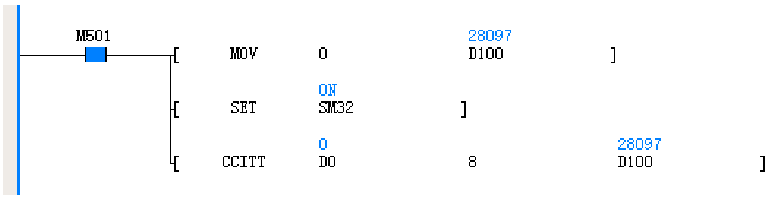


- In the case of SM32 = OFF, perform a 16-bit mode checksum.



If M501 = ON, perform CCITT checksum operation on the 8 bytes of data starting from D0 (both high byte and low byte participate in the operation), and store the checksum in D100 = 54724.

- In the case of SM32 = ON, perform a 8-bit mode checksum.



If M502 = ON, perform CCITT checksum operation on the 8 bytes of data starting from D0 (only the low byte participates in the operation, and the high byte does not participate), and store the checksum in D100 = 28097.

3.28.3 CRC16: CRC16 Checksum Calculation Command

Command table	CRC16 (S) (n) (D)	Applicable model	TS600 series
16-Bit command	CRC16 CRC16 Checksum Calculation		
32-Bit command	-		

Operand	Type	Bit			Word		Indexing	Constant
		X, Y, M, LM, T, C, S	Dx.y	Custom bit variable	D, R, V, Z, T, C	Custom word variable		
S	WORD, Array*n	-	-	-	√[1]	√	√	-
n	INT	-	-	-	√	√	√	√
D	WORD	-	-	-	√[1]	√	√	-

Remark:

[1]Only the D, V, and R elements are supported.

Operand Description

S: The operand of the source, representing the starting address of the data to be verified.

n: The quantity of data to be verified, in bytes, where 0 ⩽ n ⩽ 256.

D: The operand of the destination, representing the address of the verification result software element.

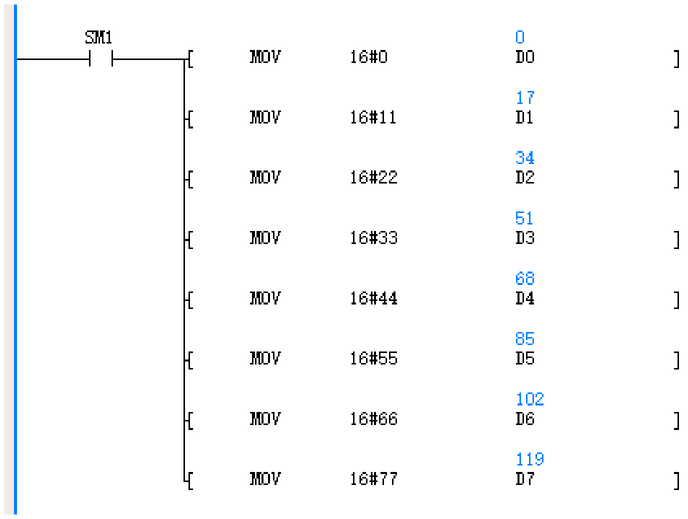
Function Description

1. Perform CRC16 checksum operation on n bytes of data starting from the initial unit (S), and assign the result to unit D.
2. The polynomial for the CRC16 checksum algorithm is: $X^{16} + X^{12} + X^{5} + 1$.

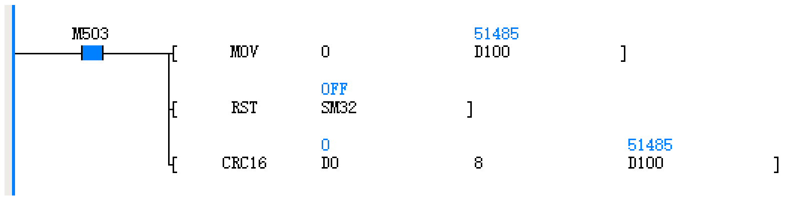
Precautions

- If the verification quantity n is less than 0 or greater than 256, the system reports an operand error, and the command is not executed.
- Each time the command is executed, the system substitutes the content of D before execution into the operation, so D must be initialized before execution.
- If using standard Modbus CRC for verification, initialize the D element (checksum) with the value 16#FFFF. Additionally, the high and low bytes (high 8 bits, low 8 bits) need to be swapped.
- When SM32 is in the OFF state, perform the CRC16 checksum operation with 16 bits as the unit, with the high 8 bits and low 8 bits of the verification data starting from unit S2 participating together.
- When SM32 is in the ON state, perform the CRC16 checksum operation with 8 bits as the unit, with the low 8 bits of the verification data starting from unit S2 participating.

Application Example

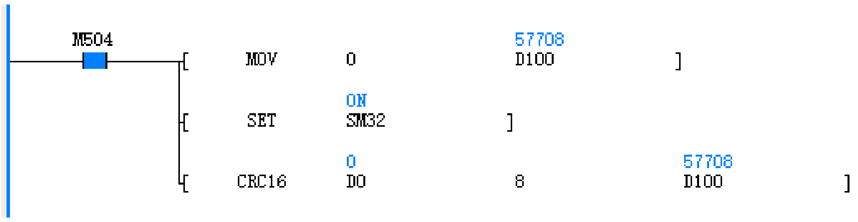


- In the case of SM32 = OFF, perform a 16-bit mode checksum.



If M503 = ON, perform CCITT checksum operation on the 8 bytes of data starting from D0 (both high byte and low byte participate in the operation), and store the checksum in D100 = 51485.

- In the case of SM32 = ON, perform a 8-bit mode checksum.



If M504 = ON, perform CCITT checksum operation on the 8 bytes of data starting from D0 (only the low byte participates in the operation, and the high byte does not participate), and store the checksum in D100 = 57708.

3.28.4 LRC: LRC16 Checksum Calculation Command

Command table	LRC (S) (n) (D)			Applicable model	TS600 series			
16-Bit command	LRC (LRC16 Checksum Calculation)							
32-Bit command	-							
Operand	Type	Bit			Word		Indexing	Constant
		X, Y, M, LM, T, C, S	Dx.y	Custom bit variable	D, R, V, Z, T, C	Custom word variable		
S	WORD, Array*n	-	-	-	√[1]	√	√	-
n	INT	-	-	-	√	√	√	√
D	WORD	-	-	-	√[1]	√	√	-

Remark:

[1]Only the D, V, and R elements are supported.

Operand Description

S: The operand of the source, representing the starting address of the data to be verified.

n: The quantity of data to be verified, in bytes, where 0 ⩽ n ⩽ 256.

D: The operand of the destination, representing the address of the verification result software element.

Function Description

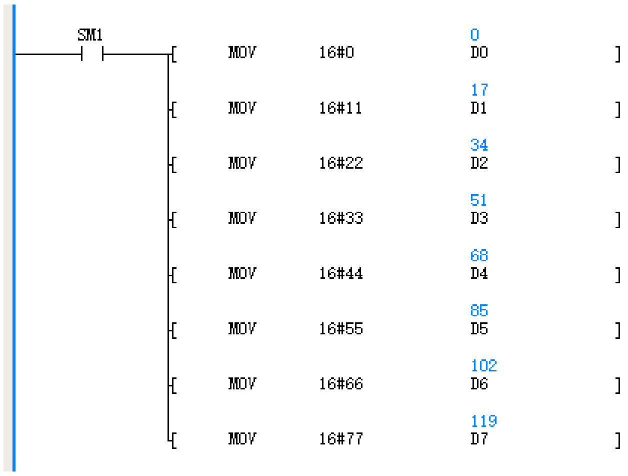
Perform LRC checksum operation on n bytes of data starting from the initial unit (S), and assign the result to unit D.

Precautions

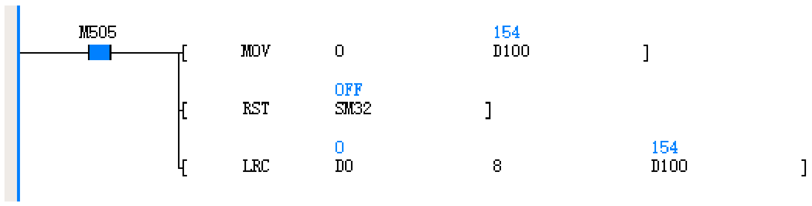
- If the verification quantity n is less than 0 or greater than 256, the system reports an operand error, and the command is not executed.
- Each time the command is executed, the system substitutes the content of D before execution into the operation, so D must be initialized before execution.
- When SM32 is in the OFF state, perform LRC checksum operation with 16 bits as the unit. The high 8 bits and low 8 bits of the verification data starting from unit S2 participate together in the LRC checksum operation.
- When SM32 is in the ON state, perform LRC checksum operation with 8 bits as the unit. The low 8 bits of the verification data starting from unit S2 participate in the LRC checksum operation.

Application Example

- Init

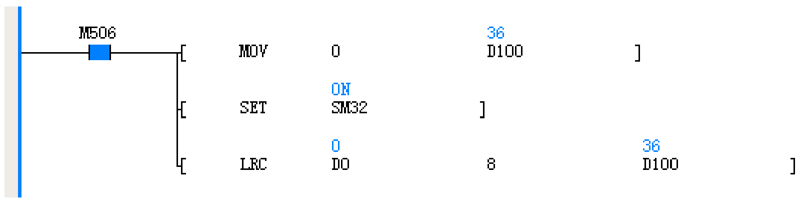


- In the case of SM32 = OFF, perform a 16-bit mode checksum.



If M505 = ON, perform CCITT checksum operation on the 8 bytes of data starting from D0 (both high byte and low byte participate in the operation), and store the checksum in D100 = 154.

- In the case of SM32 = ON, perform a 8-bit mode checksum.



If M506 = ON, perform CCITT checksum operation on the 8 bytes of data starting from D0 (only the low byte participates in the operation, and the high byte does not participate), and store the checksum in D100 = 36.

3.28.5 CCD: CCD Checksum Calculation Commands

Command table	CCD (S) (D1) (D2) (n)				Applicable model	TS600 series		
16-Bit command	CCD: CCD Checksum Calculation							
32-Bit command	-							
Operand	Type	Bit			Word		Indexing	Constant
		X, Y, M, LM, T, C, S	Dx.y	Custom bit variable	D, R, V, Z, T, C	Custom word variable		
S	WORD, Array*n	-	-	-	√[1]	√	-	-
D1	WORD	-	-	-	√[1]	√	-	-
D2	WORD	-	-	-	√[1]	√	-	-
n	WORD	-	-	-	√	√	√	√

Remark:

[1]Only the D, V, and R elements are supported.

Operand Description

S: Source operand, representing the starting address of the variable to be used in the checksum calculation.

D1: Destination operand 1, which stores the checksum.

D2: Destination operand 2, which stores the result of a termwise XOR logical operation.

n: Number of bytes for checksum, where $0 \leqslant n \leqslant 256$.

Function Description

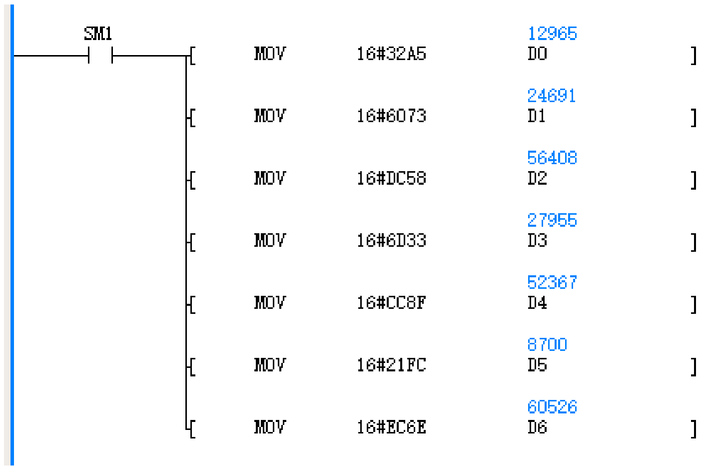
1. Perform two types of checksum calculations on n data elements starting from the initial unit (S).
2. Store the result of the direct addition summation operation in D1.
3. Store the result of the bitwise XOR operation in D2.
4. This operation is commonly used for communication data correctness and integrity verification.

Precautions

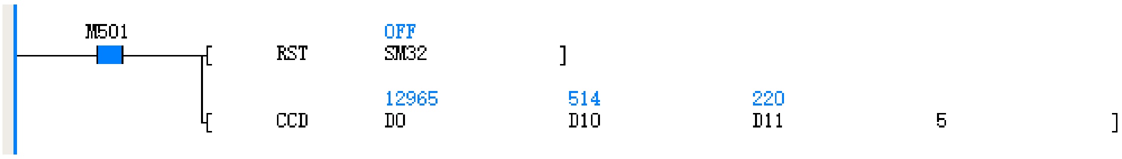
- If the verification quantity n is less than 0 or greater than 256, the system reports an operand error, and the command is not executed.
- When SM32 is in the OFF state, perform CCD checksum operation with 16 bits as the unit. The high 8 bits and low 8 bits of the verification data starting from unit S2 participate together in the CCD checksum operation.
- When SM32 is in the ON state, the low 8 bits of the verification data starting from unit S2 participate in the CCD checksum operation with 8 bits as the unit.

Application Example

- Init

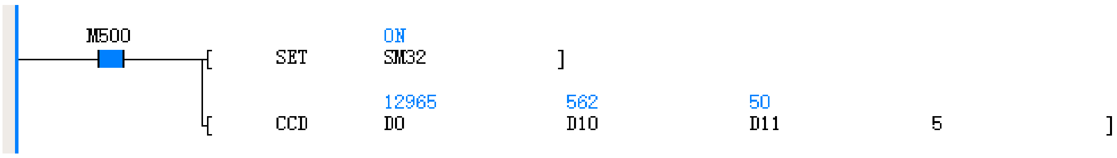


- In the case of SM32 = OFF, perform a 16-bit mode checksum.



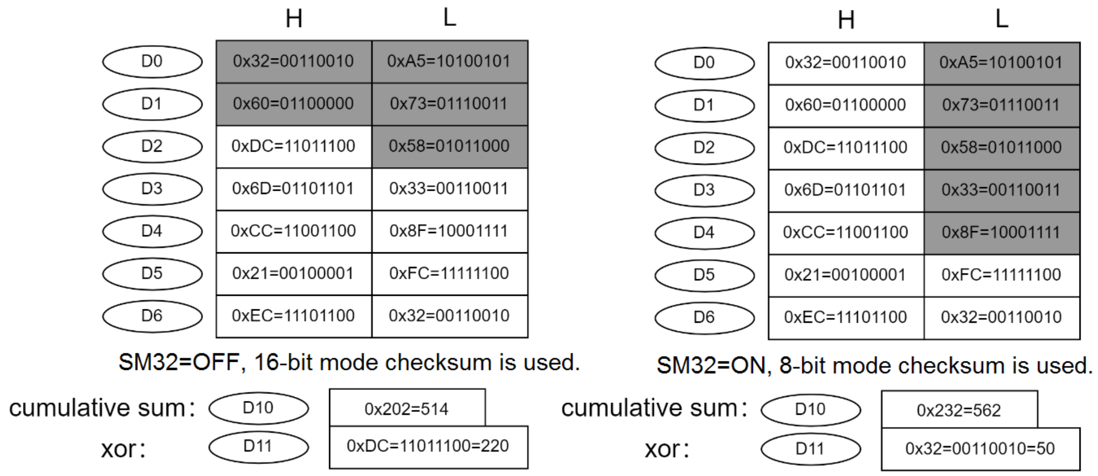
When M501 is set to ON, perform a CCD checksum operation on the 5-byte data starting from D0 (involving both high and low bytes in the calculation). Store the checksum result in D10, and store the XOR result in D11.

- In the case of SM32 = ON, perform a 8-bit mode checksum.



When M502 is set to ON, perform a CCD checksum operation on the 5-byte data starting from D0 (only involving the low byte in the calculation, excluding the high byte). Store the checksum result in D10, and store the XOR result in D11.

Refer to the specific diagram below for details.



3.29 Other Commands

3.29.1 Command Table

Command Category	Name	Function
Other Commands	RND	Generate random number
	DUTY	Generate timed pulse
	REF	I/O immediate refresh

3.29.2 RND: Generate Random Number Command

Command table	RND (D)			Applicable model	TS600 series			
16-Bit command	The RND instruction generates a random number.							
32-Bit command	-							
Operand	Type	Bit			Word		Indexing	Constant
		X, Y, M, LM, T, C, S	Dx.y	Custom bit variable	D, R, V, Z, T, C	Custom word variable		
D	WORD/ DWORD	-	-	-	√[1]	√	√	-

Remark:

[1]The V element is not supported.

Operand Description

D: Starting address of the soft element to store the random number.

Function Description

Upon execution of the command, a pseudorandom number in the range of 0 to 65535 is generated, and its value is stored in the D unit as a random number. If the generated random number is 0, the zero flag (SM18) is set.

Application Example

When M500 is set to ON, a random number is generated and stored in D0, where D0 is set to 16214.

3.29.3 DUTY: Generate Duty Cycle Pulse Command

Command table	DUTY (S1) (S2) (D1) (D2)	Applicable model	TS600 series
16-Bit command	The DUTY instruction generates a duty cycle pulse.		
32-Bit command	-		

Operand	Type	Bit			Word		Indexing	Constant
		X, Y, M, LM, T, C, S	Dx.y	Custom bit variable	D, R, V, Z, T, C	Custom word variable		
S1	WORD	-	-	-	√[1]	√	√	√
S2	WORD	-	-	-	√[1]	√	√	√
D1	BOOL	√[2]	√	√	-	-	-	-
D2	WORD	-	-	-	√	√	√	-

Remark:

[1]The V and Z elements are not supported.

[2]The X element is not supported.

Operand Description

S1: Number of scans in the ON state.

S2: Number of scans in the OFF state.

D1: Destination address for the continuous timed output.

D2: Number of pulses in the timed output.

Function Description

1. The timed pulse output unit D undergoes a transition in the ON state for S1 scans, followed by the OFF state for S2 scans.
2. Multiple DUTY commands should not utilize the same target address for timed clock outputs.

Precautions

- The action begins on the rising edge of the command. The current flow persists even if interrupted, and it halts only on STOP or power failure.
- Up to 128 SORTC commands are supported.

Application Example

When M500 is set to ON, M0 undergoes ON state for 10,000 scans, followed by an OFF state for the next 10,000 scans. Simultaneously, the count of scan occurrences is stored in D0.

3.29.4 REF: Immediate I/O Refresh Command

Command table	DUTY (D) (S)	Applicable Models	TS600 series
16-Bit command	DUTY: I/O Immediate Refresh		
32-Bit command	-		

Operand	Type	Bit			Word		Indexing	Constant
		X, Y, M, LM, T, C, S	Dx.y	Custom bit variable	D, R, V, Z, T, C	Custom word variable		
D	BOOL	√[1]	-	-	-	-	-	-
S	INT	-	-	-	-	-	-	√

Remark:

[1]Only X and Y elements are supported.

Operand Description

D: Starting X/Y element to be refreshed.

S: Number of ports to be refreshed.

Function Description

Generally, the inputs and outputs of a PLC are executed after the user program concludes. During the computation, if there is a need to read the latest input status or update the output status immediately, this command can be employed.

Precautions

- The indices of input ports (X_n, Y_n) should be multiples of 8.
- The quantity of ports to be refreshed should also be multiples of 8.
- Between FOR-NEXT commands, REF is commonly used for immediate processing.
- During the execution of interrupt processing involving input and output actions, in the interrupt subroutine, use the REF command to refresh inputs and outputs, obtain the latest input information, and promptly output computation results.
- For relay-type output points, consider the response time of the output points.

Application Example

When M0 is set to ON, the states of Y0 to Y7 are immediately output, unaffected by the scanning cycle.

4 Appendix

4.1 System Variables

4.1.1 Overview

System variables are registers used to represent and modify PLC runtime status information, such as device model, version number, serial ports, Ethernet, CAN communication, etc.

4.1.2 List of System Variables

Table 4-1 Overview of System Variables

Category of system variables	Description
_SYS_CAN	Information related to CAN communication, such as node number, baud rate, slave on-line status, etc
_SYS_COM	Serial communication related information, such as node number, baud rate, slave node online status, etc.
_SYS_ECAT	EtherCAT master and slave node status information
_SYS_ETHERNET	Ethernet communication information, e.g., IP, MAC, online status, error diagnostics.
_SYS_INFO	PLC system information, e.g., SN number, firmware version, RTC clock, module diagnostics, system logs.

4.1.3 _SYS_CAN CAN interface running information

Table 4-2 _sCAN Interface Information

Name	Data Type	Description	R/W type
_sCAN.BaudRate	INT	Baud rate (kbps)	R
_sCAN.LoadRate	INT	Load rate (%)	R
_sCAN.RxPerSec	INT	Frames received per second (FPS)	R
_sCAN.TxPerSec	INT	Frames transmitted per second (FPS)	R
_sCAN.RxErrCnt	INT	Receive error counter	R
_sCAN.TxErrCnt	INT	Transmit error counter	R
_sCAN.Protocol	INT	Communication protocol	R

Table 4-3 _sCANOpen Interface Information

Name	Data Type	Description	R/W type
_sCANOpen.NodeID	INT	Node ID	R
_sCANOpen.NodeState	INT	Node status, 1 for online and 0 for offline	R
_sCANOpen.sEmcy	_stru_CANOpen_EMCY	Emergency	R
_sCANOpen.sDebug	_stru_CANOpen_DEBUG	Commissioning interface	R
_sCANOpen.sCfgErr	_stru_CANOpen_CFG_ERR	Configuration error message	R
_sCANOpen.sEmcy.NodeID	INT	Emergency Node ID	R
_sCANOpen.sEmcy.ErrorCode	INT	Emergency error code	R
_sCANOpen.sEmcy.RegAndMsErrField	INT	Error register	R

Name	Data Type	Description	R/W type
		pre-manufacturer custom error message area	
Note: CANOpen.sDebug.NodeID	INT	Debug node ID	R
_sCANOpen.sDebug.State	INT	Debug status	R
_sCANOpen.sDebug.Index	INT	Debug primary index	R
_sCANOpen.sDebug.SubIndexAndSize	INT	Debug sub-index and data size	R
_sCANOpen.sDebug.Data	INT	Debug data or error code	R
_sCANOpen.sCfgErr.NodeID	INT	Configuration error message node ID	R
_sCANOpen.sCfgErr.ConfigIndex	INT	Configuration number	R
_sCANOpen.sCfgErr.ErrorCode	DINT	Error code	R

4.1.4 _SYS_COM Serial Port Operation Information

Table 4-4 _sCOMx Serial Port Information

Name	Data Type	Description	R/W type
_sCOMx.BaudRate	DINT	Baud rate	R
_sCOMx.DataBits	INT	Data bit	R
_sCOMx.Parity	INT	Check bit	R
_sCOMx.StopBits	INT	Stop bit	R
_sCOMx.Interface	INT	Physical interface	R
_sCOMx.Protocol	INT	Communication protocol	R

Note: Here _sCOMx represents _sCOM1_485, _sCOM2_485, and _sCOM3_232.

Table 4-5 Free Port Protocol Information of _sFreex Serial Ports

Name	Data Type	Description	R/W type
_sFreex.Sent	DINT	The number of bytes sent by the COM port	R
_sFreex.Received	DINT	The number of bytes received by the COM port	R
_sFreex.Timeout	DINT	Maximum timeout time (ms)	R
_sFreex.SendLen	INT	The number of bytes sent	R
_sFreex.SendBuf	INT[256]	Transmit data buffer	R
_sFreex.RecvBuf	INT[256]	Receive data buffer	R
_sFreex.RecvLen	INT	The number of bytes at a single time	R
_sFreex.Enable	BOOL	Enabled state	R
_sFreex.Activate	BOOL	Activated status	R
_sFreex.Busy	BOOL	Busy state	R
_sFreex.Done	BOOL	Completion flag	R
_sFreex.Error	BOOL	Error flag	R

Note: Here x represents a serial port number which ranges between 1 and 3.

Table 4-6 Modbus RTU/ASCII Master Station Information of _sMbMstx Serial Ports

Name	Data Type	Description	R/W type
_sMbMstx.AddrNum	INT	Number of nodes	R
_sMbMstx.TimeOut	INT	Maximum timeout time (ms)	R
_sMbMstx.ResponseTime	INT	Response time (ms)	R
_sMbMstx.Connected	BOOL	Number of connections	R
_sMbMstx.Enable	BOOL	Enabled state	R

Name	Data Type	Description	R/W type
_sMbMstx.Activate	BOOL	Activated status	R
_sMbMstx.Busy	BOOL	Busy state	R
_sMbMstx.Done	BOOL	Port Modbus communication completion flag bit	R
_sMbMstx.Error	BOOL	Communication error flag bit for port Modbus	R
_sMbMstx.ErrSlID	BOOL[256]	Communication error flag bit for corresponding slave station Modbus	R

Note: Here x represents a serial port number which ranges between 1 and 3.

Table 4-7 Information of Slave Station Connected to Modbus RTU/ASCII Master Station _sMbMstMsgx Serial Ports

Name	Data Type	Description	R/W type
_sMbMstMsgx.DisableSlv	BOOL	Slave node disabled or not	R
_sMbMstMsgx.IsSlvDisable	BOOL[256]	Slave disability flag	R

Note: Here x represents a serial port number which ranges between 1 and 3.

Table 4-8 Modbus RTU/ASCII Master Station Information of _sMbSlvx Serial Ports

Name	Data Type	Description	R/W type
_sMbSlvx.SlvID	INT	Node number	R
_sMbSlvx. Enable	BOOL	Enabled state	R
_sMbSlvx. Activate	BOOL	Activated status	R
_sMbSlvx. Busy	BOOL	Busy state	R
_sMbSlvx. Done	BOOL	Port Modbus communication completion flag bit	R
_sMbSlvx. Error	BOOL	Communication error flag bit for port Modbus	R

Note: Here x represents a serial port number which ranges between 1 and 3.

Table 4-9 _sNNBusx N: N Protocol Information

Name	Data Type	Description	R/W type
_NNBusx.SlvId	INT	Node number	R
_NNBusx.Delay	INT	N: N additional delay	R
_NNBusx.RetryTimes	INT	Retry times	R
_NNBusx.Mode	INT	N: N network refresh mode	R
_NNBusx.Period	DINT	N: Polling cycle of N communication	R
_NNBusx.Error	DINT	Communication error flag, where bits 0~31 respectively represent the error flag bits of stations with station numbers 0~31. A value of 1 represents an error, while a value of 0 represents no error	R

Note: Here x represents a serial port number which ranges between 1 and 3.

4.1.5 _SYS_ECAT EtherCAT running status information

Table 4-10 EtherCAT Master Station State Information

Name	Data Type	Description	R/W type
_sECATMst.MasterRunState	BOOL	Flag bit of main station operation state, ON: run; OFF: stop	R
_sECATMst.LinkState	BOOL	Physical connection state of master station ON: normal; OFF: network cable disconnected	R
_sECATMst.HeartBeat	BOOL	EtherCAT real-time task heartbeat	R
_sECATMst.BlockHeartBeat	BOOL	EtherCAT non-real-time task heartbeat	R

Name	Data Type	Description	R/W type
_sECATMst.MaxCycleTime	DINT	Maximum cycle time, in μs	R
_sECATMst.MinCycleTime	DINT	Minimum cycle time, in μs	R
_sECATMst.CycleTime	DINT	Cycle time, in μs	R
_sECATMst.MaxExeTime	DINT	Maximum execution time, in μs	R
_sECATMst.MinExeTime	DINT	Minimum execution time, in μs	R
_sECATMst.ExeTime	DINT	Execution time, in μs	R
_sECATMst.Tx_frames	DINT	Total frames sent	R
_sECATMst.Rx_frames	DINT	Total frames received	R
_sECATMst.Tx_frame_rates	DINT	Frame rate at which the data is transmitted, frames/second	R
_sECATMst.Rx_frame_rates	DINT	Frame rate at which the data is received, frames/second	R
_sECATMst.Tx_bytes_rate	DINT	The speed at which the byte is transmitted, bytes/second	R
_sECATMst.Rx_bytes_rate	DINT	The speed at which the byte is received, bytes/second	R
_sECATMst.Loss_rate	DINT	Lost EtherCAT data frame, in frames	R
_sECATMst.ResetTime	BOOL	Reset execution time and cycle time	R/W
_sECATMst.StartMaster	BOOL	Start the master	R/W
_sECATMst.StopMaster	BOOL	Stop the master	R/W
_sECATMst.ClearFrameCounter	BOOL	Reset transmit and receive data frame counter	R/W
_sECATMst.DisableMaster	BOOL	Disable Master Enable	R/W
_sECATMst.SlavesState	INT	Online status of all slaves 0: some slave stations are not online; 1: all slave stations are online	R
_sECATMst.FirstErrorSlave	INT	First faulty slave	R
_sECATMst.LibVersion	DINT	EtherCAT library version	R
_sECATMst.MstVersion	DINT	EtherCAT master version	R
_sECATMst.DriveVersion	DINT	EtherCAT NIC driver version	R
_sECATMst.Tx_error_cnt	DINT	EtherCAT transmit error count	R
_sECATMst.Rx_timeout_cnt	DINT	EtherCAT receive frame timeout count	R
_sECATMst.Tx_corrupt_cnt	DINT	EtherCAT receive invalid frame count	R
_sECATMst.Tx_unmach_cnt	DINT	EtherCAT receive unmatched frame count	R
_sECATMst.RxPDOLength	DINT	EtherCAT total receive PDOs	R
_sECATMst.TxPDOLength	DINT	EtherCAT total transmit PDOs	R
_sECATMst.ConfigureState	DINT	EtherCAT configuration status	R
_sECATMst.Delay	DINT	EtherCAT synchronizer	R
_sECATMst.SlvLinkState	INT	Connection status of all slave	R
_sECATMst.DisableState	INT	Master disability state	R

Table 4-11 EtherCAT Slave Station State Information

Name	Data Type	Description	R/W type
_sECATSlv[x].Unused	BOOL	System retention	R
_sECATSlv[x].SlaveRunState	BOOL	Slave running status ON: run; OFF: stop	R
_sECATSlv[x].SetAliasState	BOOL	Written site alias state of slave station, where ON indicates busy; OFF indicates idle or settings completed	R
_sECATSlv[x].SetAliasError	BOOL	Failed to write alias to slave	R
_sECATSlv[x].MatchState	BOOL	Slave type mismatch	R

Name	Data Type	Description	R/W type
_sECATSlv[x].ConfigError	BOOL	Slave configuration error	R
_sECATSlv[x].SetAlias	BOOL	Set slave alias, rising edge is valid	R/W
_sECATSlv[x].DisableEnable	BOOL	Disable slave enable	R/W
_sECATSlv[x].ALState	INT	EtherCAT state machine status	R
_sECATSlv[x].ALCode	INT	Fault code	R
_sECATSlv[x].ActAlias	INT	Actual node alias	R
_sECATSlv[x].TarAlias	INT	Target alias to write	R/W
_sECATSlv[x].StationAddress	INT	Actual node name	R
_sECATSlv[x].SlaveRingPos	INT	Configuration address	R
_sECATSlv[x].SDOErrorCode	INT	Startup parameter configuration error count	R
_sECATSlv[x].CfgErrorCode	DINT	Configuration error code	R
_sECATSlv[x].DisableState	INT	Configuration state	R

Note: Here x represents an ECAT slave station number which ranges between 0 and 71.

4.1.6 _SYS_ETHERNET Ethernet Information

Table 4-12 _sENETx Network Port Information

Name	Data Type	Description	R/W type
_sENETx.MAC	INT[3]	Physical address	R
_sENETx.IP	DINT	Native IP address	R/W
_sENETx.NetMask	DINT	Subnet mask	R/W
_sENETx.GateWay	DINT	Gateway	R/W

Note:

- The IP, MAC, and other information of the local machine can be monitored in the variable table.
- Here x represents an Ethernet number which ranges between 1 and 2.

Table 4-13 _sMbTcpMstx[i] Modbus RTU/ASCII Master Station Information

Name	Data Type	Description	R/W type
_sMbTcpMstx[i].SlvIP	DINT	IP address of connected slave station	R
_sMbTcpMstx[i]. SlvPort	DINT	Port number of connected slave station	R
_sMbTcpMstx[i].Timeout	INT	Connection timeout time (ms)	R
_sMbTcpMstx[i].ResponseTime	INT	Response time (ms)	R
_sMbTcpMstx[i].Connected	BOOL	Connection flag	R
_sMbTcpMstx[i].Enable	BOOL	Enabled state	R
_sMbTcpMstx[i].Activate	BOOL	Activated status	R
_sMbTcpMstx[i].Busy	BOOL	Busy state	R
_sMbTcpMstx[i].Done	BOOL	Communication completion flag bit for Modbus	R
_sMbTcpMstx[i].Error	BOOL	Communication error flag bit for Modbus	R

Note: Here x represents an Ethernet number which ranges between 1 and 2, and i represent a ModbusTCP slave station number which ranges between 0 and 63.

Table 4-14 Information of Slave Station Connected to _sMbTcpMstMsgx[i] ModbusTCP Master Station

Name	Data Type	Description	R/W type
_sMbTcpMstMsgx[i].MstIP	DINT	IP address of master station	R
_sMbTcpMstMsgx[i].MstPort	DINT	Port number of master station	R
_sMbTcpMstMsgx[i].DisableSlv	BOOL	Slave node disabled or not	R
_sMbTcpMstMsgx[i].IsSlvDisable	BOOL[256]	Slave disability flag	R

Note: Here x represents an Ethernet number which ranges between 1 and 2, and i represent a ModbusTCP slave station number which ranges between 0 and 63.

Table 4-15 _sMbTcpSlvx Modbus TCP Slave Station Information

Name	Data Type	Description	R/W type
_sMbTcpSlvx.Connections	INT	Number of connections	R
_sMbTcpSlvx.MstIP	DINT	Master IP address table	R
_sMbTcpSlvx.MstPort	DINT	Master port number table	R
_sMbTcpSlvx.SlvIP	DINT	Slave node IP address.	R
_sMbTcpSlvx.SlvPort	DINT	Slave node port number.	R
_sMbTcpSlvx. SlvID	INT	Slave node ID	R
_sMbTcpSlvx.Connected	BOOL	Connection flag of corresponding node	R
_sMbTcpSlvx.Enable	BOOL	Enabled state	R
_sMbTcpSlvx.Error	BOOL	Communication error flag bit	R
_sMbTcpSlvx.ErrIP	DINT	IP address of master node with error	R
_sMbTcpSlvx.ErrPort	DINT	Port number of master node with error	R

Note: Here x represents an Ethernet number which ranges between 1 and 2.

4.1.7 _SYS_INFO PLC Running Information

Table 4-16 DevInfo: Device Information

Name	Data Type	Description	R/W type
_sDevInfo.Device	INT	Device Model ID	R
_sDevInfo.Vender	INT	Manufacturer ID	R
_sDevInfo.HWVersion	DINT	Hardware version	R
_sDevInfo.SWVersion	DINT	Software version	R
_sDevInfo.FPGAVersion	DINT	FPGA version	R
_sDevInfo.BattVolt	DINT	Battery voltage	R

Get PLC production device information.

Table 4-17 OSM: System Monitor

Name	Data Type	Description	R/W type
_sOSM.CPU	INT	CPU usage rate	R
_sOSM.Memory	INT	Memory usage	R

Get CPU and memory utilization and diagnose CPU performance.

Table 4-18 Program: User Program Information

Name	Data Type	Description	R/W type
_sProgram.TotalSize	DINT	Total program capacity	R
_sProgram.UsedSize	DINT	Used program capacity	R
_sProgram.CurRunTime	DINT	Current program runtime (μs)	R
_sProgram.MinRunTime	DINT	Minimum program runtime (μs)	R
_sProgram.MaxRunTime	DINT	Maximum program runtime (μs)	R
_sProgram.AveRunTime	DINT	Avergae program runtime (μs)	R
_sProgram.ConstScanTime	DINT	Constant scan time (μs)	R
_sProgram.WDT	DINT	Watch dog reset time (s)	R
_sProgram.Reset	BOOL	Reset cycle time	R/W

Obtain the execution cycle time of programs and tasks, so as to judge the complexity of program execution logic.

Table 4-19 CurErrLst: Error Message List

Name	Data Type	Description	R/W type
_sCurErrLst.Quantity	DINT	Current error quantity	R
_sCurErrLst.sErrInfo	_stru_ERR_INFO[42]	Current error message list	R
_sCurErrLst.sErrInfo.SubErrorCode	INT	Sub-error code	R
_sCurErrLst.sErrInfo.MainErrorCode	INT	Main error code	R
_sCurErrLst.sErrInfo.TimStamp	DINT	Time stamp	R

The error log information of PLC is recorded.

Table 4-20 RTC Clock

Name	Data Type	Description	R/W type
_sDataTime.Second	INT	Second	R
_sDataTime.Minute	INT	Minute	R
_sDataTime.Hour	INT	Hour	R
_sDataTime.Day	INT	Day	R
_sDataTime.Month	INT	Month	R
_sDataTime.Year	INT	Year	R
_sDataTime.WeekDay	INT	Week	R
_sDataTime.YearDay	INT	Days	R
_sDataTime.Timestamp	DINT	Total seconds	R

Get the RTC clock.

Table 4-21 UsrIntCtl: Interrupt Enable Control

Name	Data Type	Description	R/W type
_sUsrIntCtl[x]	_stru_USR_INT_CTL[67]	Interrupt enable control	-
_sUsrIntCtl[x].Enable	BOOL	Enable control bit	R
_sUsrIntCtl[x].IntID	INT	Interrupt program ID	R

Note: Here x represents an interrupt number which ranges between 0 and 66.

Table 4-22 ExtModule: Extension Module System Variable Related Information

Name	Data Type	Description	R/W type
_sExtModule.CfgNum	INT	User-configured module number	R
_sExtModule.ActNum	INT	Actually mounted module number	R
_sExtModule.ResAlign	-	Reserved for byte alignment	-
_sExtModule.ExtSlot	_stru_EXT_SLOT[16]	-	-
_sExtModule.ExtSlot.CfgType	INT	Type of module configured by user	R
_sExtModule.ExtSlot.ActType	INT	Type of module auctually mounted	R
_sExtModule.ExtSlot.Error	BOOL	Error state	R
_sExtModule.ExtSlot.Disable	BOOL	Module disable	R
_sExtModule.ExtSlot.ResAlign	-	Reserved for byte alignment	-
_sExtModule.ExtSlot.SWVersion	DINT	Software version	R
_sExtModule.ExtSlot.LGVersion	DINT	Logic device version	R

Table 4-23 ExtCard: Expansion Card Related Information

Name	Data Type	Description	R/W type
_sExtCard.CfgType	INT	Type of module configured by user	R
_sExtCard.ActType	INT	Type of module auctually mounted	R
_sExtCard.SWVersion	DINT	Software version	R
_sExtCard.LGVersion	DINT	Logic device version	R

Name	Data Type	Description	R/W type
_sExtCard.Error	BOOL	Error state	R
_sExtCard.Disable	BOOL	Module disable	R

Table 4-24 AlmInfo Alarm Information and Control Bits

Name	Data Type	Description	R/W type
_sAlmInfo.Enable	BOOL	Alarm enabled	R/W
_sAlmInfo.ActFlg	BOOL	S900 ~ S999 alarm action flag	R
_sAlmInfo.MinNum	INT	S900~S999 minimum alarm action element number	R

Table 4-25 SM System Variables

Name	Data Type	Description	R/W type
SM0	BOOL	Running monitoring bit	R
SM1	BOOL	Initial running pulse bit	R
SM2	BOOL	Energization flag bit	R
SM3	BOOL	Error flag bit	R
SM10	BOOL	Clock oscillation with a cycle of 10ms	R
SM11	BOOL	Clock oscillation with a cycle of 100ms	R
SM12	BOOL	Clock oscillation with a cycle of 1s	R
SM13	BOOL	Clock oscillation with a cycle of 1min	R
SM14	BOOL	Clock oscillation with a cycle of 1hour	R
SM15	BOOL	Scan cycle oscillation bit	R
SM18	BOOL	Operation zero flag	R
SM19	BOOL	Operation borrow flag	R
SM20	BOOL	Operation carry flag	R
SM22	BOOL	Bit set for command execution error	R
SM23	BOOL	Bit set for overflow of command element number subscript	R
SM24	BOOL	Bit set for illegal command parameter	R
SM30	BOOL	Flag for multi-cycle command completion	R
SM31	BOOL	Flag for BINDA command output character	R/W
SM32	BOOL	Flag for processing mode of ATI/ITA/ASC/CCITT/CRC16/LRC/CCD command bit	R/W
SM33	BOOL	SORTR/SORTC command descending sort enable	R/W
SM34	BOOL	Bit for data format settings of SMOV command	R/W
SM35	BOOL	Flag for all comparison results of BKCMP command matrices being 1	R

4.2 Error Codes

4.2.1 Error Code Classification

Table 4-26 Error Code Classifications

Device Category	Device type	Module type	Main error code (HEX)	Sub-error Code Range
CPU	System-related	Hardware failure	0001	0001~FFFF
		System failure	0002	0001~FFFF
		Program failure	0003	0001~FFFF
		Reserved fault	0004~0007	0001~FFFF
	System component-related	Clock system component failure	0008	0001~FFFF
		IP system component failure	0009	0001~FFFF
		Reserved fault	000A~000F	0001~FFFF

Device Category	Device type	Module type	Main error code (HEX)	Sub-error Code Range
	Functional component-related	Codesys motion control failure	0010	0001~FFFF
		Autonomous motion control failure	0011	0001~FFFF
		High speed input failure	0012	0001~FFFF
		CANopen axis control failure	0013	0001~FFFF
		Reserved fault	0014~0017	0001~FFFF
	Process library	Reserved fault	0018~002F	0001~FFFF
Backplane bus	Backplane bus-related	CPU IO failure	0030	0001~FFFF
		Digital quantity failure	0031	0001~FFFF
		Analog quantity failure	0032	0001~FFFF
		Fault of temperature measuring module	0033	0001~FFFF
		Encoder input failure	0034	0001~FFFF
		Reserved fault	0035~003F	0001~FFFF
Fieldbus	Serial port-related	Modbus RTU/ASCII Master 1	0040	0001~FFFF
		Modbus RTU/ASCII Master 2	0041	0001~FFFF
		Modbus RTU/ASCII Master 3	0042	0001~FFFF
		Modbus RTU/ASCII Slave 1	0043	0001~FFFF
		Modbus RTU/ASCII Slave 2	0044	0001~FFFF
		Modbus RTU/ASCII Slave 3	0045	0001~FFFF
		Serial freeport 1	0046	0001~FFFF
		Serial freeport 2	0047	0001~FFFF
		Serial freeport 3	0048	0001~FFFF
		Reserved fault	0049~004F	0001~FFFF
	CAN-related	CANopen	0050	0001~FFFF
		CANnet	0051	0001~FFFF
		Reserved fault	0052~0057	0001~FFFF
	Profibus	Profibus DP	0058	0001~FFFF
		Reserved fault	0059~005F	0001~FFFF
	Reserved	Reserved fault	005A~6F	0001~FFFF
Industrial Ethernet	Profinet-related	Profinet	0070	0001~FFFF
		Reserved fault	0071~007F	0001~FFFF
	Ethernet/IP-related	Ethernet/IP	0080	0001~FFFF
		Reserved fault	0081~008F	0001~FFFF
	EtherCAT-related	EtherCAT	0090	0001~FFFF
		ET-Digital quantity	0091	0001~FFFF
		ET-Analog quantity	0092	0001~FFFF
		ET-Temperature measuring module	0093	0001~FFFF
		ET-Encoder input	0094	0001~FFFF
		Reserved fault	0095~009F	0001~FFFF
	Modbus TCP-related	Modbus TCP Master(Ethernet1)	00A0	0001~FFFF
		Modbus TCP Master(Ethernet2)	00A1	0001~FFFF
		Modbus TCP Slave(Ethernet1)	00A2	0001~FFFF
		Modbus TCP Slave(Ethernet2)	00A3	0001~FFFF
		Reserved fault	00A4~00AF	0001~FFFF
	TCP-related	TCP	00B0	0001~FFFF
		Reserved fault	00B1~00B7	0001~FFFF
	UDP-related	UDP	00B8	0001~FFFF
		Reserved fault	00B9~00BF	0001~FFFF
	OPCUA	Reserved fault	00C0	0001~FFFF
	Reserved	Reserved fault	00C1–EF	0001~FFFF

Device Category	Device type	Module type	Main error code (HEX)	Sub-error Code Range
Expansion card	IoT card	4G expansion card	00F0	0001~FFFF
	Reserved	Reserved fault	00F1~00F3	0001~FFFF
Other	Other	Reserved fault	00F4~00FF	0001~FFFF

4.2.2 Error Code List

Table 4-27 Error Code Details

Hex main error code (corresponding decimal)	Hex error subcode (corresponding decimal)	Meaning of error	Solution	Error level
0x0001(1)	0x0001(1)	Button cell is not installed or battery voltage is too low	Check the battery	General error
0x0001(1)	0x0002(2)	Device supply voltage is too low (less than 19V)	Check the power supply	General error
0x0002(2)	0x0001(1)	Hardware initialization error	Check whether the peripheral device works normally and whether the driver is loaded successfully	Serious error
0x0002(2)	0x0002(2)	Failed to open GPIO	Check whether the driver and hardware work properly	Serious error
0x0002(2)	0x0003(3)	Failed to write GPIO	Check whether the driver and hardware work properly	Serious error
0x0002(2)	0x0004(4)	Failed to read GPIO	Check whether the driver and hardware work properly	Serious error
0x0002(2)	0x0005(5)	Failed to open FPGA FMC	Check whether the driver and hardware work properly	Serious error
0x0002(2)	0x0006(6)	SPI operation failed	Check whether the driver and hardware work properly	Serious error
0x0002(2)	0x0007(7)	Failed to update FPGA firmware read signal	Check whether the driver and hardware work properly	Serious error
0x0002(2)	0x0008(8)	Failed to read FPGA firmware file	Check whether the file exists or is corrupted	Serious error
0x0002(2)	0x0009(9)	Failed to open I2C device	Check whether the driver and hardware work properly	Serious error
0x0002(2)	0x000A(10)	Failed to write to I2C device	Check whether the driver and hardware work properly	Serious error
0x0002(2)	0x000B(11)	Failed to read I2C device	Check whether the driver and hardware work properly	Serious error
0x0002(2)	0x000C(12)	Failed to write FMC device	Check whether the driver or FPGA is working properly	Serious error
0x0002(2)	0x000D(13)	Failed to read FMC device	Check whether the driver or FPGA is working properly	Serious error
0x0002(2)	0x000E(14)	Failed to open USB device	Check whether the driver and hardware work properly	Serious error
0x0002(2)	0x000F(15)	Failed to create USB epoll	Check whether the system is working properly	Serious error
0x0002(2)	0x0010(16)	Programming port TCP initialization failed	Check whether the driver and hardware work properly	Serious error
0x0002(2)	0x0011(17)	Failed to create	Check whether the driver and	Serious

Hex main error code (corresponding decimal)	Hex error subcode (corresponding decimal)	Meaning of error	Solution	Error level
		programming port TCP epoll	hardware work properly	error
0x0002(2)	0x0012(18)	Element and variable forced setting failed	Check whether the element type and address are correct	Serious error
0x0002(2)	0x0013(19)	Failed to open configuration file	Check whether the configuration file exists or is damaged	Serious error
0x0002(2)	0x0014(20)	Power-down keeping configuration parsing failed	Check whether the profile data is correct	General error
0x0002(2)	0x0015(21)	Failed to stop output configuration parsing	Check whether the profile data is correct	General error
0x0002(2)	0x0016(22)	Watchdog configuration parsing failed	Check whether the profile data is correct	General error
0x0002(2)	0x0017(23)	Constant scan time configuration parsing failed	Check whether the profile data is correct	General error
0x0002(2)	0x0018(24)	Power-down wait time configuration parsing failed	Check whether the profile data is correct	General error
0x0002(2)	0x0019(25)	Digital filter parameter configuration parsing failed	Check whether the profile data is correct	General error
0x0002(2)	0x001A(26)	Advanced control configuration parsing failed	Check whether the profile data is correct	General error
0x0002(2)	0x001B(27)	External input run configuration parsing failed	Check whether the profile data is correct	General error
0x0002(2)	0x001C(28)	Serial port 1 configuration parsing failed	Check whether the profile data is correct	General error
0x0002(2)	0x001D(29)	Serial port 2 configuration parsing failed	Check whether the profile data is correct	General error
0x0002(2)	0x001E(30)	Serial port 232 configuration parsing failed	Check whether the profile data is correct	General error
0x0002(2)	0x001F(31)	Modbus RTU Master 1 configuration parsing failed	Check whether the profile data is correct	General error
0x0002(2)	0x0020(32)	Modbus RTU Master 2 configuration parsing failed	Check whether the profile data is correct	General error
0x0002(2)	0x0021(33)	Ebus serial port 1 configuration parsing failed	Check whether the profile data is correct	General error
0x0002(2)	0x0022(34)	Ebus serial port 2 configuration parsing failed	Check whether the profile data is correct	General error
0x0002(2)	0x0023(35)	Expansion module configuration parsing failed	Check whether the profile data is correct	General error
0x0002(2)	0x0024(36)	Interrupt configuration code configuration parsing failed	Check whether the profile data is correct	General error
0x0002(2)	0x0025(37)	Network port 1 configuration parsing failed	Check whether the profile data is correct	General error
0x0002(2)	0x0026(38)	Network port 2 configuration parsing failed	Check whether the profile data is correct	General error
0x0002(2)	0x0027(39)	Modbus TCP Master 1 configuration parsing failed	Check whether the profile data is correct	General error
0x0002(2)	0x0028(40)	Modbus TCP Master 2 configuration parsing failed	Check whether the profile data is correct	General error

Hex main error code (corresponding decimal)	Hex error subcode (corresponding decimal)	Meaning of error	Solution	Error level
0x0002(2)	0x0029(41)	CANOpen configuration parsing failed	Check whether the profile data is correct	General error
0x0002(2)	0x002A(42)	Ethercat configuration parsing failed	Check whether the profile data is correct	General error
0x0002(2)	0x002B(43)	Fieldbus pulse axis configuration parsing failed	Check whether the profile data is correct	General error
0x0002(2)	0x002C(44)	Encoder axis configuration parsing failed	Check whether the profile data is correct	General error
0x0002(2)	0x002D(45)	Fieldbus pulse axis configuration parsing failed	Check whether the profile data is correct	General error
0x0002(2)	0x002E(46)	Encoder axis configuration parsing failed	Check whether the profile data is correct	General error
0x0002(2)	0x002F(47)	Axis group configuration parsing failed	Check whether the profile data is correct	General error
0x0002(2)	0x0030(48)	CAM table configuration parsing failed	Check whether the profile data is correct	General error
0x0002(2)	0x0031(49)	Axis type configuration parsing failed	Check whether the profile data is correct	General error
0x0002(2)	0x0032(50)	Fieldbus servo axis configuration parsing failed	Check whether the profile data is correct	General error
0x0002(2)	0x0033(51)	Local pulse axis configuration parsing failed	Check whether the profile data is correct	General error
0x0002(2)	0x003F(63)	Failed to allocate profile memory	Check system free memory	Serious error
0x0002(2)	0x0040(64)	Configuration parsing overrun failed	Check whether the profile data is correct	General error
0x0002(2)	0x0041(65)	Failed to start application due to undervoltage	Check whether the power supply voltage is normal	Serious error
0x0002(2)	0x0042(66)	Power failure detected	Check whether the power supply voltage is normal	Serious error
0x0002(2)	0x0043(67)	Failed to open power-down keeping file	Check whether the power-down keeping file or file directory exists	Serious error
0x0002(2)	0x0044(68)	Failed to get power-down keeping file size	Check whether the power-down keeping file is damaged	Serious error
0x0002(2)	0x0045(69)	Failed to map the power-down keeping file	Check the file size and whether the system is normal	Serious error
0x0002(2)	0x0046(70)	Failed to release the power-down keeping file mapping	Check the file size and whether the system is normal	Serious error
0x0002(2)	0x0047(71)	Error in detecting power-down keeping file header	Check whether the power-down keeping file is damaged	Serious error
0x0002(2)	0x0048(72)	Error in detecting power-down keeping file length	Check whether the power-down keeping file is damaged	Serious error
0x0002(2)	0x0049(73)	Error in detecting power-down keeping file tail	Check whether the power-down keeping file is	Serious error

Hex main error code (corresponding decimal)	Hex error subcode (corresponding decimal)	Meaning of error	Solution	Error level
			damaged	
0x0002(2)	0x004A(74)	Error in detecting power-down keeping file CRC	Check whether the power-down keeping file is damaged	Serious error
0x0002(2)	0x0050(80)	Watchdog timeout	Check whether the user program is running correctly	Warning
0x0002(2)	0x0051(81)	Error message mismatch	Check whether the error message is filled in correctly	Warning
0x0002(2)	0x0052(82)	Failed to open RND file	Check whether the system is working properly	Serious error
0x0002(2)	0x0053(83)	Failed to create thread	Check whether the system is working properly	Serious error
0x0002(2)	0x0054(84)	Failed to open instruction library	Check whether the system is working properly	Serious error
0x0002(2)	0x0055(85)	Failed to open user program	Check whether the system is working properly	Serious error
0x0002(2)	0x0056(86)	Device model does not match fieldbus axis number	Check device model and user program fieldbus number	Serious error
0x0003(3)	0x0030(48)	Illegal instruction system parameter	Please recompile and download the user program	General error
0x0003(3)	0x0031(49)	Parameter is out of limit address range	Check whether the parameter variable data length of the instruction is out of range	General error
0x0003(3)	0x0032(50)	Illegal instruction user parameters	Check whether the parameters of the instruction are in the wrong order or the size is set incorrectly	General error
0x0003(3)	0x0033(51)	Wrong PID sampling time	Stop PID operation and check whether the parameters are set correctly	General error
0x0003(3)	0x0034(52)	Wrong PID filter constant	Stop PID operation and check whether the parameters are set correctly	General error
0x0003(3)	0x0035(53)	Wrong PID proportional gain	Stop PID operation and check whether the parameters are set correctly	General error
0x0003(3)	0x0036(54)	Wrong PID integration time	Stop PID operation and check whether the parameters are set correctly	General error
0x0003(3)	0x0037(55)	Wrong PID differential gain	Stop PID operation and check whether the parameters are set correctly	General error
0x0003(3)	0x0038(56)	Wrong PID differential time	Stop PID operation and check whether the parameters are set correctly	General error
0x0003(3)	0x0039(57)	Wrong PID manual output PID value	Stop PID operation and check whether the parameters are set correctly	General error

Hex main error code (corresponding decimal)	Hex error subcode (corresponding decimal)	Meaning of error	Solution	Error level
0x0003(3)	0x003A(58)	The PID setting target value exceeds the upper/lower limit of the setting value	Stop PID operation and check whether the parameters are set correctly	General error
0x0003(3)	0x003B(59)	TPID temperature catastrophe error	Stop PID operation and check whether the parameters are set correctly	General error
0x0003(3)	0x003C(60)	TPID temperature out of range error	Stop PID operation and check whether the parameters are set correctly	General error
0x0003(3)	0x003D(61)	TPID mode is not supported	Check whether the setting mode is correct	General error
0x0003(3)	0x003E(62)	The setting of TPID control period is unreasonable	Check whether the control period is larger than the PID sampling time	General error
0x0003(3)	0x003F(63)	TPID mode auto-tuning failed	Try to lower the set temperature value and rerun the program for self-tuning. If self-tuning is not possible for a long time, please confirm whether the control device and sensors are abnormal	General error
0x0003(3)	0x0050(80)	Illegal ASCII code conversion value	Confirm whether the ASCII code to be converted conforms to the ASCII code specification	General error
0x0003(3)	0x0051(81)	Clock chip read-write error	If the host computer clock can be read normally, recompile and download the program. If the clock of the host computer cannot be read, check whether the hardware is damaged or the battery is exhausted	General error
0x0003(3)	0x0052(82)	Stack definition error	Check whether the stack data is normal	General error
0x0003(3)	0x0053(83)	The divisor in the division operation is 0	Check whether the divisor in the division operation is 0	General error
0x0003(3)	0x0054(84)	String instruction or data error	Check whether the string instruction or string data is illegal	General error
0x0003(3)	0x0055(85)	Override between source and target operands	Check for overlap between source and target operands	General error
0x0003(3)	0x0080(128)	Invalid upper and lower limit setting range	Check whether the lower limit is greater than the upper limit, and exchange the upper/lower limit operation in this case	Warning
0x0003(3)	0x0081(129)	PID measurements out of range	-	Warning

Hex main error code (corresponding decimal)	Hex error subcode (corresponding decimal)	Meaning of error	Solution	Error level
0x0003(3)	0x0082(130)	PID deviation out of range	The calculated value of PID deviation exceeds-32768 ~ 32767	Warning
0x0003(3)	0x0083(131)	PID proportional term out of range	The calculated value of PID proportional term exceeds-32768 ~ 32767	Warning
0x0003(3)	0x0084(132)	PID integral term out of range	The calculated value of PID integral term exceeds-32768 ~ 32767	Warning
0x0003(3)	0x0085(133)	PID differential term out of range	The calculated value of PID differential term exceeds-32768 ~ 32767	Warning
0x0003(3)	0x0086(134)	PID operation result out of range	PID calculation result exceeds-32768 ~ 32767	Warning
0x0003(3)	0x0087(135)	Instruction fetches ID number incorrectly	Check the compilation ID of the host computer	Warning
0x0008(8)	0x0001(1)	Failed to open RTC device	Check whether the driver and hardware work properly	Serious error
0x0008(8)	0x0002(2)	Failed to write RTC device	Check whether the driver and hardware work properly	Serious error
0x0008(8)	0x0003(3)	Failed to read RTC device	Check whether the driver and hardware work properly	Serious error
0x0008(8)	0x0004(4)	Failed to read the real time of the system	Check whether the system works normally in real time	General error
0x0008(8)	0x0005(5)	Failed to read RTC time of FPGA	Check whether FPGA and RTC work properly	Warning
0x0008(8)	0x0006(6)	Failure to write RTC time of FPGA	Check whether FPGA and RTC work properly	Warning
0x0009(9)	0x0001(1)	The IP segments of IP1 and IP2 repeat error	Misplace the network segments of IP1 and IP2	General error
0x0009(9)	0x0011(17)	Read: IP1 module--Open file to report error	Check whether the network driver is running normally	General error
0x0009(9)	0x0012(18)	Read: IP1 module--Unable to get IP information	Check whether the network driver is running normally	General error
0x0009(9)	0x0013(19)	Write: IP1 module--IP address configuration error	Check whether the IP segment data is valid data (0~255)	General error
0x0009(9)	0x0014(20)	Write: IP1 module--Mask configuration error	Check whether the mask data is valid data (0~255)	General error
0x0009(9)	0x0015(21)	Write: IP1 module--Gateway configuration error	Check whether the gateway data is valid data (0~255)	General error
0x0009(9)	0x0016(22)	Write: IP1 module--USB network segment repeat error	Misalign the IP1 segment with the USB segment (TM700--192.168.3.x)	General error
0x0009(9)	0x0017(23)	Write: IP1 module--IP and gateway not in the same network error	Configure the IP segment and gateway in the same network	General error
0x0009(9)	0x0021(33)	Read: IP2 module--Open file to report error	Check whether the network driver is running normally	General error

Hex main error code (corresponding decimal)	Hex error subcode (corresponding decimal)	Meaning of error	Solution	Error level
0x0009(9)	0x0022(34)	Read: IP2 module--Unable to get IP information	Check whether the network driver is running normally	General error
0x0009(9)	0x0023(35)	Write: IP2 module--IP address error	Check whether the IP segment data is valid data (0~255)	General error
0x0009(9)	0x0024(36)	Write: IP2 module--Mask error	Check whether the mask data is valid data (0~255)	General error
0x0009(9)	0x0025(37)	Write: IP2 module--Gateway error	Check whether the gateway data is valid data (0~255)	General error
0x0009(9)	0x0026(38)	Write: IP2 module--USB network segment repeat error	Misalign the IP2 segment with the USB segment (TM700--192.168.3.x)	General error
0x0009(9)	0x0027(39)	Write: IP2 module--IP and gateway not in the same network error	Configure the IP segment and gateway in the same network	General error
0x0011(17)	0x0001(1)	The current axis ID is not within the valid range	Check whether the axis ID parameter settings are reasonable	General error
0x0011(17)	0x0002(2)	The current function block ID is not within the valid range	Check whether the function block ID parameter settings of the host computer are reasonable	General error
0x0011(17)	0x0003(3)	The current function block cannot be started due to the unreasonable PLCopen state	Check whether the current axis state meets the PLCopen state machine switching process when the current command is triggered	Warning
0x0011(17)	0x0004(4)	Axis configuration failed	Check whether the axis configuration is successful	Warning
0x0011(17)	0x0005(5)	The address of the PDO parameter DigitalIput is NULL	1, Detect whether the parameter is mapped in the slave IO mapping; 2. Check whether the PDO parameter exists in the XML version of the servo slave	Warning
0x0011(17)	0x0006(6)	Current axis/servo error	The axis/servo is faulty, and the error can be cleared by calling the MC_Reset command or restarting the MC_Power command	General error
0x0011(17)	0x0007(7)	The current axis is not enabled and therefore in the Disabled state	Switch the axis to the Standstill state by calling the MC_Power command	Warning
0x0011(17)	0x0008(8)	The positive hard limit of the axis is triggered	Call the reset instruction to switch the axis state from ErrorStop state to Standstill state	General error

Hex main error code (corresponding decimal)	Hex error subcode (corresponding decimal)	Meaning of error	Solution	Error level
0x0011(17)	0x0009(9)	The negative hard limit of the axis is triggered	Call the reset instruction to switch the axis state from ErrorStop state to Standstill state	General error
0x0011(17)	0x000A(10)	The positive soft limit of the axis is triggered	Call the reset instruction to switch the axis state from ErrorStop state to Standstill state	General error
0x0011(17)	0x000B(11)	The negative soft limit of the axis is triggered	Call the reset instruction to switch the axis state from ErrorStop state to Standstill state	General error
0x0011(17)	0x000C(12)	The pulse axis has not selected any output device	Check whether the pulse axis has selected an output device	Warning
0x0011(17)	0x000D(13)	The bus axis has not selected any output device	Check whether the bus axis has selected an output device	Warning
0x0011(17)	0x000E(14)	The current command does not support repeated calls	The current command does not support repeated calls to the function block, so avoid this situation manually	Warning
0x0011(17)	0x000F(15)	Axis type setting error	Check whether the axis type matches the command type	Warning
0x0011(17)	0x0010(16)	The fieldbus base PDO address is NULL	It is not recommended to map PDO parameters in slave device description file I/O mapping	Warning
0x0011(17)	0x0011(17)	Positive hard limit ID configuration failed	Check whether the current pulse axis input and output points are reused	Warning
0x0011(17)	0x0012(18)	Negative hard limit ID configuration failed	Check whether the current pulse axis input and output points are reused	Warning
0x0011(17)	0x0013(19)	Probe ID1 configuration failed	Check whether the current pulse axis input and output points are reused	Warning
0x0011(17)	0x0014(20)	Probe ID2 configuration failed	Check whether the current pulse axis input and output points are reused	Warning
0x0011(17)	0x0015(21)	Servo error ID configuration failed	Check whether the current pulse axis input and output points are reused	Warning
0x0011(17)	0x0016(22)	Home signal ID configuration failed	Check whether the current pulse axis input and output points are reused	Warning
0x0011(17)	0x0017(23)	Z signal ID configuration failed	Check whether the current pulse axis input and output points are reused	Warning
0x0011(17)	0x0018(24)	Axis enable ID configuration failed	Check whether the current pulse axis input and output	Warning

Hex main error code (corresponding decimal)	Hex error subcode (corresponding decimal)	Meaning of error	Solution	Error level
			points are reused	
0x0011(17)	0x0019(25)	Failed to clear the servo error ID configuration	Check whether the current pulse axis input and output points are reused	Warning
0x0011(17)	0x001A(26)	The axis address is NULL	Check whether the axis configuration is successful	Warning
0x0011(17)	0x001B(27)	Bus axis enable failed	If bus axis enable timed out, check whether the EtherCAT communication and feedback state words are normal	Warning
0x0011(17)	0x001C(28)	The bus axis has not entered the OP state	Check whether the EtherCAT communication has entered the OP state	General error
0x0011(17)	0x001D(29)	The current function block execution is invalid	The current command function is not yet open and is invalid for use	Warning
0x0011(17)	0x001E(30)	The current axis communication timed out	1. Check whether EtherCAT communication is in OP state; 2. Check whether EtherCAT communication return value is normal	Warning
0x0011(17)	0x001F(31)	Under the current axis configuration, the EtherCAT synchronization cycle cannot be less than 1 ms	Check whether the setting of the synchronization cycle of the EtherCAT master station is less than 1ms (in case of mixed use of bus axis and pulse axis, the EtherCAT synchronization cycle cannot be less than 1 ms)	Warning
0x0011(17)	0x0020(32)	The PLC does not run	Check whether the PLC dial switch is set to Stop	Warning
0x0011(17)	0x0021(33)	The axis triggered a soft-limit deceleration and stop	The current axis is in the process of the soft-limit deceleration and stopping, and the execution of the current triggerd command is invalid	Warning
0x0011(17)	0x0022(34)	The address of the current command parameter is NULL	If the address of the current command parameter is NULL, provide an input variable or contact the INVT technical personnel	Warning
0x0011(17)	0x0023(35)	During the pulse axis motion, the pulse frequency of the current interpolation cycle is greater than or equal to 200K	The maximum running frequency of the pulse axis must not exceed 200K, so it is recommended to reduce the running velocity	General error
0x0011(17)	0x0024(36)	The pulse axis FPGA cache	This is only a prompt	Warning

Hex main error code (corresponding decimal)	Hex error subcode (corresponding decimal)	Meaning of error	Solution	Error level
		reached the limit value		
0x0011(17)	0x0025(37)	The PDO data address in EtherCAT is NULL	Check whether the EtherCAT communication is normal	General error
0x0011(17)	0x0026(38)	The current servo axis is not on-line	Check whether EtherCAT communication is normal; Check whether the current servo axis is plugged into the network cable connection	General error
0x0011(17)	0x0027(39)	The current axis communication failed	If the EtherCAT communication failed during the operation, check the state of the EtherCAT communication	Warning
0x0011(17)	0x0028(40)	The value of the PDO parameter StatusWord is 0	Check whether the EtherCAT communication is normal	Warning
0x0011(17)	0x0029(41)	The address of the PDO parameter ErrorCode is NULL	1. Check whether EtherCAT communication is normal. 2. Check whether the PDO parameter is configured	Warning
0x0011(17)	0x002A(42)	The current axis does not support torque control	Check the axis type configuration, as torque control only supports the bus axis	Warning
0x0011(17)	0x0065(101)	The enable command state is abnormal	If the enable command state is abnormal, contact the INVT technical personnel	Warning
0x0011(17)	0x0066(102)	The reset command state is abnormal	If the reset command state is abnormal, contact the INVT technical personnel	Warning
0x0011(17)	0x0067(103)	Reset timed out	If the axis reset timed out, check whether the EtherCAT communication is normal	Warning
0x0011(17)	0x0068(104)	The current axis state does not support the superimposed motion command	If the current axis state does not support the superimposed motion command, refer to the specific commands for using the command	Warning
0x0011(17)	0x0069(105)	Input parameter error	The command input parameter is not within the valid range	Warning
0x0011(17)	0x006A(106)	The system report an error about the repeated calls of the MC_Stop command	Please check whether the same axis is called more than once	Warning
0x0011(17)	0x006B(107)	The system report an error about the repeated calls of the MC_ImmediateStop command	Please check whether the same axis is called more than once	Warning

Hex main error code (corresponding decimal)	Hex error subcode (corresponding decimal)	Meaning of error	Solution	Error level
0x0011(17)	0x006C(108)	The input parameter of the MC_Stop command is not within the valid range	Please check whether the command parameters are within the valid range, and call the MC_Reset command to reset the axis state	General error
0x0011(17)	0x006D(109)	The input parameter of the MC_Halt command is not within the valid range	Please check whether the command parameters are within the valid range, and call the MC_Reset command to reset the axis state	General error
0x0011(17)	0x006E(110)	The input parameter of the MC_SetOverride command is not within the valid range	Please check whether the command parameters are within the valid range, and call the MC_Reset command to reset the axis state	Warning
0x0011(17)	0x006F(111)	The input parameter of the MC_MoveVelocity command is not within the valid range	Please check whether the command parameters are within the valid range, and call the MC_Reset command to reset the axis state	General error
0x0011(17)	0x0070(112)	The input parameter of the MC_MoveRelative command is not within the valid range	Please check whether the command parameters are within the valid range, and call the MC_Reset command to reset the axis state	General error
0x0011(17)	0x0071(113)	The input parameter of the MC_MC_MoveAbsoulte command is not within the valid range	Please check whether the command parameters are within the valid range, and call the MC_Reset command to reset the axis state	General error
0x0011(17)	0x0072(114)	The input parameter of the MC_Jog command is not within the valid range	Please check whether the command parameters are within the valid range, and call the MC_Reset command to reset the axis state	General error
0x0011(17)	0x0073(115)	The input parameter of the MC_Inch command is not within the valid range	Please check whether the command parameters are within the valid range, and call the MC_Reset command to reset the axis state	General error
0x0011(17)	0x0074(116)	The input parameter of the MC_Home command is not within the valid range	Please check whether the command parameters are within the valid range, and call the MC_Reset command to reset the axis state	General error
0x0011(17)	0x0075(117)	The input parameter of the MC_SetPosition command is not within the valid range	Please check whether the command parameters are within the valid range, and call the MC_Reset command to reset the axis state	Warning

Hex main error code (corresponding decimal)	Hex error subcode (corresponding decimal)	Meaning of error	Solution	Error level
0x0011(17)	0x0076(118)	It is invalid to trigger the MC_SetOverride command in the current axis state	The current axis is in the process of reversing, and the velocity regulation does not take effect	Warning
0x0011(17)	0x0077(119)	The current axis is in the operation process of the axis group	Run the axis after the axis group operation is completed	Warning
0x0011(17)	0x0078(120)	The axis is not in the Standstill state	Before triggering the current command, switch the axis to the StandStill state	Warning
0x0011(17)	0x0079(121)	Resetting by the MC_Reset command is invalid	The current axis state is not ErrorStop, so resetting is invalid	Warning
0x0011(17)	0x007A(122)	The interpolation cycle value settings are invalid	Check the EtherCAT synchronization cycle settings	Warning
0x0011(17)	0x007B(123)	It is invalid to trigger the MC_Stop command	Check whether the current axis state can trigger the instruction	Warning
0x0011(17)	0x007C(124)	It is invalid to trigger the MC_Halt command	Check whether the current axis state can trigger the instruction	Warning
0x0011(17)	0x007D(125)	It is invalid to trigger the MC_ImmediateStop command	Check whether the current axis state can trigger the instruction	Warning
0x0011(17)	0x007E(126)	The input parameter of the MC_TouchProbe command is not within the valid range	Please check whether the command parameters are within the valid range, and call the MC_Reset command to reset the axis state	Warning
0x0011(17)	0x007F(127)	The input parameter of the MC_MoveSuperImosed command is not within the valid range	Please check whether the command parameters are within the valid range, and call the MC_Reset command to reset the axis state	Warning
0x0011(17)	0x0080(128)	The MC_Home command has been called repeatedly	Check whether the home function block has been called repeatedly on the same axis	Warning
0x0011(17)	0x0081(129)	The input parameter of the MC_MoveFeed command is not within the valid range	Please check whether the command parameters are within the valid range, and call the MC_Reset command to reset the axis state	General error
0x0011(17)	0x0082(130)	The probe channel used is not configured	Check whether the PDO data in the "Process Data" section of the configuration interface for the servo axis on the host computer has been added	Warning

Hex main error code (corresponding decimal)	Hex error subcode (corresponding decimal)	Meaning of error	Solution	Error level
			(Possible mappings: 0x60B8, 0x60B9, 0x60BA, 0x60BB, 0x60BC, and 0x60BD)	
0x0011(17)	0x0083(131)	When the interrupt fixed length function is used with Mode=0 or Mode=1, the probe signal has not arrives after the first distance is traveled.	Check whether the probe signal is triggered normally.	Warning
0x0011(17)	0x0084(132)	When the probe function is triggered, the probe channel used has already been occupied by the interrupt fixed length function.	Check whether the channel is incompatible.	Warning
0x0011(17)	0x0085(133)	The axis configuration index parameter is not within the valid range	Check whether the axis configuration index parameter is within the valid range	Warning
0x0011(17)	0x0086(134)	The axis parameter input by the MC_SetAxisConfigPara command is not within the valid range	Check whether the axis setting parameter is within the valid range	Warning
0x0011(17)	0x0087(135)	The input parameter of the MC_MoveBuffer command is not within the valid range	Please check whether the command parameters are within the valid range, and call the MC_Reset command to reset the axis state	Warning
0x0011(17)	0x0088(136)	The input parameter of the MC_SyncMoveVelocity command is not within the valid range	Please check whether the command parameters are within the valid range, and call the MC_Reset command to reset the axis state	Warning
0x0011(17)	0x0089(137)	The input parameter of the MC_MoveVelocityCSV command is not within the valid range	Please check whether the command parameters are within the valid range, and call the MC_Reset command to reset the axis state	Warning
0x0011(17)	0x008A(138)	The input parameter of the MC_SyncTorqueControl command is not within the valid range	Please check whether the command parameters are within the valid range, and call the MC_Reset command to reset the axis state	Warning
0x0011(17)	0x008B(139)	PDO data used is not configured	The process data 0x6060 and 0x6061 in the servo configuration of the host computer are not configured	Warning
0x0011(17)	0x008C(140)	PDO data used is not configured	The process data 0x606C is not configured in the servo configuration of the host	Warning

Hex main error code (corresponding decimal)	Hex error subcode (corresponding decimal)	Meaning of error	Solution	Error level
			computer	
0x0011(17)	0x008D(141)	PDO data used is not configured	The process data 0x60FF is not configured in the servo configuration of the host computer	Warning
0x0011(17)	0x008E(142)	PDO data used is not configured	The process data 0x6071 and 0x607F in the servo configuration of the host computer are not configured	Warning
0x0011(17)	0x008F(143)	PDO data used is not configured	The process data 0x6083 and 0x6084 in the servo configuration of the host computer are not configured	Warning
0x0011(17)	0x0090(144)	The current axis state does not support the single-axis velocity regulation command	Check whether the current axis state meet the requirements of the velocity regulation function	Warning
0x0011(17)	0x0091(145)	Probe does not support virtual axis or pulse axis	Check the current axis type and configure it as a bus axis	Warning
0x0011(17)	0x00C9(201)	The master and slave axes use the same axis ID	Check whether the master and slave axes are the same	Warning
0x0011(17)	0x00CA(202)	Input parameter error of MC_ GearOut function block	Check whether the input parameters of GearOut are within the constraint range of the command parameter list	General error
0x0011(17)	0x00CB(203)	It is invalid to trigger the MC_ GearOut function block	Check whether the slave axis is in gear action, and check whether the slave axis is in gear engagement action	Warning
0x0011(17)	0x00CC(204)	Input parameter error of MC_GearIn function block	Check whether the input parameters of MC_GearIn are within the constraint range of the command parameter list	General error
0x0011(17)	0x00CD(205)	The current command to run the master axis does not meet the requirements	Check whether the master axis state meets the requirements; Run the MC_Phasing instruction to check whether the current axis is in cam or gear action	Warning
0x0011(17)	0x00CE(206)	The master axis has not reached the target velocity	Check whether the current master axis has reached the target velocity	Warning
0x0011(17)	0x00CF(207)	Input parameter error of MC_CamOut function block	Check whether the input parameters of MC_CamOut are within the constraint range of the command parameter list	General error
0x0011(17)	0x00D0(208)	It is invalid to trigger the MC_CamOut command	Check whether the slave axis is in cam action, and check	Warning

Hex main error code (corresponding decimal)	Hex error subcode (corresponding decimal)	Meaning of error	Solution	Error level
			whether the slave axis is in cam engagement action	
0x0011(17)	0x00D1(209)	Input parameter error of MC_CamIn function block	Check whether the input parameters of MC_CamIn are within the constraint range of the command parameter list	General error
0x0011(17)	0x00D2(210)	The current CamTable ID is not within the valid range	Check whether the CamTable ID is within the constraint range of the command parameter list	Warning
0x0011(17)	0x00D3(211)	Setting error of StartPosition or MasterStartDistance in MC_CamIn command	Check whether MasterStartDistance and StartPosition are in the current master axis running direction in absolute position mode	Warning
0x0011(17)	0x00D4(212)	The MC_CamIn instruction is in absolute position mode, with StartPosition ahead of MasterStartDistance	Check whether StartPosition is ahead of MasterStartDistance in the absolute position mode	Warning
0x0011(17)	0x00D5(213)	The input parameters of the MC_Phasing command are not within the valid range	Check whether the input parameters of MC_Phasing are within the constraint range of the command parameter list	General error
0x0011(17)	0x00E1(225)	Master axis phase setting error	Check whether the master axis phases of two adjacent keypoints are less than or equal to 0.001 in the user-defined cam table of the MC_GenerateCamTable command	Warning
0x0011(17)	0x00E2(226)	The start point of the cam table cannot be set as a non-zero parameter	Check whether the positions of the master and slave axes at the start point of the cam are set to non-zero in the user-defined cam table of the MC_GenerateCamTable command	Warning
0x0011(17)	0x00E3(227)	The current NodeNum parameter cannot be set to 0	Check whether the MC_NodeNum parameter is set to 0 in the current mode in the GenerateCamTable command	Warning
0x0011(17)	0x00E4(228)	The current NodeNum parameter is not within the valid range	Check whether the MC_NodeNum parameter is set within the constraint range of the command parameter list in the current mode in the	Warning

Hex main error code (corresponding decimal)	Hex error subcode (corresponding decimal)	Meaning of error	Solution	Error level
			GenerateCamTable command	
0x0011(17)	0x00E5(229)	Curve type setting error in cam table	Check whether the cam curve type settings are within the constraint range of the command parameters list. They only support 0 (which represents straight lines) and 1 (which represents quintic curves)	Warning
0x0011(17)	0x00E6(230)	The cam table is empty	Check whether the cam table is configured	Warning
0x0011(17)	0x00E7(231)	Encoder master axis enable failed	Check whether the counting command ENC_Counter is enabled when using the encoder master axis	Warning
0x0011(17)	0x00E8(232)	The length of the user-defined cam table is not within the valid range	Check that the length of the user-defined cam table array must be 32 in the MC_GenerateCamTable command	Warning
0x0011(17)	0x00E9(233)	The the user-defined tappet switch is not within the valid range	Check that the length of the user-defined switch array must be 32 in the MC_DigitalCamSwitch command	Warning
0x0011(17)	0x00EA(234)	The ReferenceType parameter settings are not within the valid range	Check whether ReferenceType parameter settings are within the valid range for the current command	Warning
0x0011(17)	0x00EB(235)	The Channel parameter settings are not within the valid range	Check whether Channel parameter settings are within the valid range for the current command	Warning
0x0011(17)	0x00EC(236)	The Number parameter settings are not within the valid range	Check whether Number parameter settings are within the valid range for the current command	Warning
0x0011(17)	0x00ED(237)	The address of the Switches parameter is NULL	Check whether the Switches parameter has a given variable in the current command	Warning
0x0011(17)	0x00EE(238)	Positions are not arranged in ascending order in the tappet switch	Check whether Position in the Switches parameter is set to ascending order in the current command. If not, modify it.	Warning
0x0011(17)	0x00EF(239)	The current axis state does	Check whether the axis is in	Warning

Hex main error code (corresponding decimal)	Hex error subcode (corresponding decimal)	Meaning of error	Solution	Error level
		not support the use of the tappet command	the home state	
0x0011(17)	0x00F0(240)	The Action settings are not within the valid range for the tappet switch	Check whether Action in the Switches parameter is within the valid range for the current command	Warning
0x0011(17)	0x00F1(241)	The current Channel is already in use	Check if there is any reuse of Channel	Warning
0x0011(17)	0x00F2(242)	The Position settings in the tappet switch exceeds the rotation axis modulus cycle	Check whether Position in the Switch parameter exceeds the rotation cycle value in the rotation axis mode for the current command	Warning
0x0011(17)	0x00F3(243)	The input parameters of the MC_CombineAxes command are not within the valid range	Please check whether the command parameters are within the valid range, and call the MC_Reset command to reset the axis state	General error
0x0011(17)	0x00F4(244)	Phase of the MC_GetCamTableDistance command is not within the valid range between the start and end points	Check whether the input parameter Phase of this command is within the valid range between the start and end points	Warning
0x0011(17)	0x00F5(245)	The CurveType parameter settings are not within the valid range	Check whether CurveType parameter settings are within the valid range for the current command	Warning
0x0011(17)	0x00F6(246)	The phases of the start and end points for the MC_GetCamTableDistance command is not arranged in ascending order	1, Check whether the phase difference between the starting point and the ending point of the instruction is less than 0.001; 2. Check whether Phase in CamTable is ascending	Warning
0x0011(17)	0x00F7(247)	The current master axis has entered the ErrorStop state, and the function block has stopped running	Check the reason why the master axis has entered the ErrorStop state	Warning
0x0011(17)	0x00F8(248)	Multiple cam table save commands are used on the same axis	Check whether multiple cam table save commands are used on the same axis in the user program	Warning
0x0011(17)	0x00F9(249)	The cam table update command was not completed and the cam table save command was called instead	Check whether the user program has not completed the cam table update command and has called the cam table save command instead	Warning
0x0011(17)	0x00FA(250)	The MC_GetCamTablePhase	Check whether the	Warning

Hex main error code (corresponding decimal)	Hex error subcode (corresponding decimal)	Meaning of error	Solution	Error level
		command output parameter Phase arrey length was not equal to 6	MC_GetCamTablePhase command output parameter Phase arrey length is equal to 6	
0x0011(17)	0x00FB(251)	The acceleration settings at the starting and ending points of MC_GetCamTablePhase command were abnormal	Check the acceleration settings at the starting and ending points of MC_GetCamTablePhase command	Warning
0x0011(17)	0x00FC(252)	The MC_GetCamTablePhase command inout parameter Distance was not within the valid range of the cam table	Check the MC_GetCamTablePhase command Distance settings	Warning
0x0011(17)	0x00FD(253)	The MC_GetCamTablePhase command start point and end point references were abnormal	Please contact us	Warning
0x0011(17)	0x00FE(254)	The MC_GearInPos commandinput parameter was not within the valid range	Check whether the command input parameter is within the valid range	Warning
0x0011(17)	0x012D(301)	Input parameter error of function block	In plane arc interpolation mode 2, if the distance between the start and end points is greater than twice the radius, check and correct the parameters	General error
0x0011(17)	0x012E(302)	Axis group ID settings exceeds the range	Check and correct the axis group ID	General error
0x0011(17)	0x012F(303)	Two or more identical axis IDs are configured in the axis group	Check and correct the duplicated axis IDs in the axis group configuration interface	General error
0x0011(17)	0x0130(304)	The distance from the start end to the circle center is not equal to that from the end point to the circle center in the plane arc function block	In plane arc interpolation mode 1, check and modify the distance from the start point to the circle center and that from the and end point to the circle center	General error
0x0011(17)	0x0131(305)	The start point, circle center, and end point are on the same straight line in the plane arc function block	In plane arc interpolation mode 0, ensure that the start point, auxiliary point, and end point are on the same straight line	General error
0x0011(17)	0x0132(306)	The calculated circle center position is not unique in the plane arc function block	In plane arc interpolation mode 2, ensure that the start point is equal to the end point	General error
0x0011(17)	0x0133(307)	In the GroupImmediateStop module, the same axis group	For the same axis group, the second immediate axis group	General error

Hex main error code (corresponding decimal)	Hex error subcode (corresponding decimal)	Meaning of error	Solution	Error level
		can only call this function block once, and the second function block starts reporting an error	stop module reports error	
0x0011(17)	0x0134(308)	Axis group is in GroupImmediateStopping state	Pull down the MC_GroupImmediateStop module first, and then pull up the MC_GourpStop module	General error
0x0011(17)	0x0135(309)	In the GroupStop module, the same axis group can only call this function block once, and the second function block starts reporting an error	For the same axis group, the second MC_GroupStop module reports error when pulled up	General error
0x0011(17)	0x0136(310)	The configured velocity parameters are not within a reasonable range	Check the corresponding parameters	General error
0x0011(17)	0x0137(311)	The configured acceleration parameters are not within a reasonable range	Check the corresponding parameters	General error
0x0011(17)	0x0138(312)	The configured deceleration parameters are not within a reasonable range	Check the corresponding parameters	General error
0x0011(17)	0x0139(313)	The configured Jerk parameters are not within a reasonable range	Check the corresponding parameters	General error
0x0011(17)	0x013A(314)	The configured AbsRelMode parameters are not within a reasonable range	Check the corresponding parameters	General error
0x0011(17)	0x013B(315)	Interpolation is not allowed as there a single axis is in the rotation mode in the axis group	De-select the rotation mode option in the single axis configuration interface	General error
0x0011(17)	0x013C(316)	Interpolation is not allowed as there a single axis is in the debugging mode in the axis group	De-select the debugging mode option in the single axis configuration interface	General error
0x0011(17)	0x013D(317)	The radius parameter is not allowed to be zero	Check the corresponding parameters	General error
0x0011(17)	0x013E(318)	The parameter CircAxes is not within the allowed range	Check the corresponding parameters	General error
0x0011(17)	0x013F(319)	The parameter CircMode is not within the allowed range	Check the corresponding parameters	General error
0x0011(17)	0x0140(320)	The parameter PathChoice is not within the allowed range	Check the corresponding parameters	General error
0x0011(17)	0x0141(321)	The array parameters passed in by the host computer are incorrect	Enable host computer error protection	General error
0x0011(17)	0x0142(322)	It is not allowed to modify	Interrupt the arc	General

Hex main error code (corresponding decimal)	Hex error subcode (corresponding decimal)	Meaning of error	Solution	Error level
		the parameter CircAxes during the operation of arc interpolation	interpolation first, and then modify the parameter CircAxes	error
0x0011(17)	0x0143(323)	The current state does not allow axis group velocity regulation	The current state does not allow axis group velocity regulation, including moderate axis group deceleration	General error
0x0011(17)	0x0144(324)	An unconfigured axis group number has been used	Configure an axis group number for the used axis group in the "Axis Group Settings" list on the host computer	General error
0x0011(17)	0x0145(325)	There is a pulse axis velocity exceeding 200kHz	There is a pulse axis velocity exceeding 200kHz	General error
0x0011(17)	0x0146(326)	Two axis groups use the same axis, so that when one axis group is in motion state, the other axis group cannot enter the motion state.	Modify the reused axis, or run two axis groups at different times.	General error
0x0012(18)	0x0101(257)	System configuration-Code header frame configuration error	Update the corresponding version of the host computer or contact a technician	General error
0x0012(18)	0x0102(258)	System configuration-Code length configuration error	Update the corresponding version of the host computer or contact a technician	General error
0x0012(18)	0x0103(259)	System configuration-Code module type configuration error	Update the corresponding version of the host computer or contact a technician	General error
0x0012(18)	0x0104(260)	System configuration-Code module length configuration error	Update the corresponding version of the host computer or contact a technician	General error
0x0012(18)	0x0105(261)	System Configuration-Parameter module type configuration error	Update the corresponding version of the host computer or contact a technician	General error
0x0012(18)	0x0106(262)	System Configuration-Parameter module length configuration error	Update the corresponding version of the host computer or contact a technician	General error
0x0012(18)	0x0201(513)	Counter reset module-Axis ID exceeds maximum value	Configure the correct axis ID	Warning
0x0012(18)	0x0202(514)	Counter reset module-This axis does not belong to encoder axis	Configure the correct axis ID	Warning
0x0012(18)	0x0203(515)	Counter reset module-Axis number not configured	Configure the correct axis ID	Warning
0x0012(18)	0x0301(769)	Comparator reset module-Axis ID exceeds	Configure the correct axis ID	Warning

Hex main error code (corresponding decimal)	Hex error subcode (corresponding decimal)	Meaning of error	Solution	Error level
		maximum value		
0x0012(18)	0x0302(770)	Counter reset module-This axis does not belong to encoder axis	Configure the correct axis ID	Warning
0x0012(18)	0x0303(771)	Counter reset module-Axis number not configured	Configure the correct axis ID	Warning
0x0012(18)	0x0401(1025)	Preset module-Axis ID exceeds maximum value	Configure the correct axis ID	Warning
0x0012(18)	0x0402(1026)	Preset module-This axis does not belong to encoder axis	Configure the correct axis ID	Warning
0x0012(18)	0x0403(1027)	Preset module-No axis number configured	Configure the correct axis ID	Warning
0x0012(18)	0x0404(1028)	Preset module-TrigerMode trigger mode parameter exception	Configure the correct TrigerMode range parameter 0~3	Warning
0x0012(18)	0x0405(1029)	Preset module-Preset position out of range	Configure the preset positions within the range	Warning
0x0012(18)	0x0406(1030)	Preset module-No preset position configured	Configure the preset positions	Warning
0x0012(18)	0x0501(1281)	Counter module-Axis ID exceeds maximum value	Configure the correct axis ID	Warning
0x0012(18)	0x0502(1282)	Counter module-This axis does not belong to encoder axis	Configure the correct axis ID	Warning
0x0012(18)	0x0503(1283)	Counter module-Axis number not configured	Configure the correct axis ID	Warning
0x0012(18)	0x0601(1537)	Comparator module-Axis ID exceeds maximum value	Configure the correct axis ID	Warning
0x0012(18)	0x0602(1538)	Comparator module-This axis does not belong to encoder axis	Configure the correct axis ID	Warning
0x0012(18)	0x0603(1539)	Comparator module-Axis number not configured	Configure the correct axis ID	Warning
0x0012(18)	0x0604(1540)	Comparator module-Comparison position out of limit	Configure the comparison value within the range	Warning
0x0012(18)	0x0605(1541)	Comparator module-No comparison position configured	Configure the comparison value	Warning
0x0012(18)	0x0606(1542)	Comparator module-System does not enable hardware comparison output for this axis	Enable hardware comparison output in the axis setting interface	Warning
0x0012(18)	0x0607(1543)	Comparator module-Interrupt number out of range	Configure interrupt numbers 0~16 within the range	Warning
0x0012(18)	0x0608(1544)	Comparator module-Undefined interrupt	Generate the corresponding interrupt functions in the	Warning

Hex main error code (corresponding decimal)	Hex error subcode (corresponding decimal)	Meaning of error	Solution	Error level
		function	program	
0x0012(18)	0x0701(1793)	One-dimensional step length comparison module-Axis ID exceeds maximum value	Configure the correct axis ID	Warning
0x0012(18)	0x0702(1794)	One-dimensional step length comparison module-This axis does not belong to encoder axis	Configure the correct axis ID	Warning
0x0012(18)	0x0703(1795)	One-dimensional step comparison module-Axis number not configured	Configure the correct axis ID	Warning
0x0012(18)	0x0704(1796)	One-dimensional step length comparison module-Starting position out of range	Configure the position value within the range	Warning
0x0012(18)	0x0705(1797)	One-dimensional step length comparison module-End position out of range	Configure the position value within the range	Warning
0x0012(18)	0x0706(1798)	One-dimensional step length comparison module-Single step position out of range	Configure the position value within the range	Warning
0x0012(18)	0x0707(1799)	One-dimensional step length comparison module-Starting position not configured	Configure the position value	Warning
0x0012(18)	0x0708(1800)	One-dimensional step comparison module-End position not configured	Configure the position value	Warning
0x0012(18)	0x0709(1801)	One-dimensional step length comparison module-Single step position not configured	Configure the position value	Warning
0x0012(18)	0x070A(1802)	One-dimensional step comparison module-System does not enable hardware comparison output for this axis	Enable hardware comparison output in the axis setting interface	Warning
0x0012(18)	0x070B(1803)	One-dimensional step length comparison module-Interrupt number out of range	Configure interrupt numbers 0~16 within the range	Warning
0x0012(18)	0x070C(1804)	One-dimensional step length comparison module-Undefined interrupt function	Generate the corresponding interrupt functions in the program	Warning
0x0012(18)	0x070D(1805)	Linear mode-The starting position is less than the ending position, and the single step position is positive	The starting position is less than the ending position, and the single step position is positive	Warning
0x0012(18)	0x070E(1806)	Linear mode-The starting position is greater than the	The starting position is greater than the ending	Warning

Hex main error code (corresponding decimal)	Hex error subcode (corresponding decimal)	Meaning of error	Solution	Error level
		ending position, and the single-step position is negative	position, and the single-step position is negative	
0x0012(18)	0x0801(2049)	Linear array comparison module-Axis ID exceeds maximum value	Configure the correct axis ID number	Warning
0x0012(18)	0x0802(2050)	Linear array comparison module-This axis does not belong to encoder axis	Configure the correct axis ID number	Warning
0x0012(18)	0x0803(2051)	Linear array comparison module-Axis number not configured	Configure the correct axis ID number	Warning
0x0012(18)	0x0804(2052)	Linear array comparison module-Array not configured	Configure the array position	Warning
0x0012(18)	0x0805(2053)	Linear array comparison module-Array size greater than 100	Set the array length less than 100	Warning
0x0012(18)	0x0806(2054)	Linear array comparison module-Array size out of array range	Set the array length within the array range	Warning
0x0012(18)	0x0807(2055)	Linear array comparison module-Array size not configured	Configure the array size	Warning
0x0012(18)	0x0808(2056)	Linear array comparison module-System does not enable hardware comparison output for this axis	Enable hardware comparison output in the axis setting interface	Warning
0x0012(18)	0x0809(2057)	Linear array comparison module-Interrupt number out of range	Configure interrupt numbers 0~16 within the range	Warning
0x0012(18)	0x080A(2058)	Linear array comparison module-Undefined interrupt function	Generate the corresponding interrupt functions in the program	Warning
0x0012(18)	0x080B(2059)	Array size ≤ 0	Set the array length greater than 0	Warning
0x0012(18)	0x0901(2305)	Probe module-Axis ID exceeds maximum value	Configure the correct axis ID number	Warning
0x0012(18)	0x0902(2306)	Probe module-This axis does not belong to encoder axis	Configure the correct axis ID number	Warning
0x0012(18)	0x0903(2307)	Probe module-No axis number configured	Configure the correct axis ID number	Warning
0x0012(18)	0x0904(2308)	Probe module-Probe number parameter error (0 ~ 1)	The probe number range is 0 ~ 1	Warning
0x0012(18)	0x0905(2309)	Probe module-No probe number configured	Configure the probe number	Warning
0x0012(18)	0x0906(2310)	Probe module-Edge parameter error (0 ~ 2)	Edge parameter range is 0~2	Warning

Hex main error code (corresponding decimal)	Hex error subcode (corresponding decimal)	Meaning of error	Solution	Error level
0x0012(18)	0x0907(2311)	Probe module-Edge parameters not configured	Configure edge parameters	Warning
0x0012(18)	0x0908(2312)	Probe module-Trigger mode parameter error (0 ~ 1)	Mode parameter range is 0~1	Warning
0x0012(18)	0x0909(2313)	Probe module-Window start parameter out of range	Configure the position parameter within the range	Warning
0x0012(18)	0x090A(2314)	Probe module-Window end parameter out of range	Configure the position parameter within the range	Warning
0x0012(18)	0x090B(2315)	Probe module-Probe not enabled	Enable the probe in the axis setting interface	Warning
0x0012(18)	0x090C(2316)	In linear mode, the initial value is greater than or equal to the end value	In linear mode, set the initial value smaller than the end value	Warning
0x0012(18)	0x0A01(2561)	Linear mode configuration-Axis ID exceeds maximum value	Configure the correct axis ID number	Warning
0x0012(18)	0x0A02(2562)	Linear mode configuration-This axis does not belong to encoder axis	Configure the correct axis ID number	Warning
0x0012(18)	0x0A03(2563)	Linear mode configuration-No axis number configured	Configure the correct axis ID number	Warning
0x0012(18)	0x0A04(2564)	Linear mode configuration-Mode selection parameter error (0 ~ 1)	Mode selection parameter range is 0~1	Warning
0x0012(18)	0x0A05(2565)	Linear mode configuration-Mode selection parameter not configured	Configure the mode selection parameter	Warning
0x0012(18)	0x0A06(2566)	Linear mode configuration-Software limit enable parameter not configured	Configure the software limit parameter	Warning
0x0012(18)	0x0A07(2567)	Linear mode configuration-Positive limit configuration out of range value	Configure the position value within the range	Warning
0x0012(18)	0x0A08(2568)	Linear mode configuration-Negative limit position out of range value	Configure the position value within the range	Warning
0x0012(18)	0x0A09(2569)	Linear mode configuration-Period value position out of range value	Configure the position value within the range	Warning
0x0012(18)	0x0A0A(2570)	Linear mode configuration-Positive limit parameter not configured	Configure the positive limit parameter	Warning
0x0012(18)	0x0A0B(2571)	Linear mode configuration-Negative limit	Configure the negative limit parameter	Warning

Hex main error code (corresponding decimal)	Hex error subcode (corresponding decimal)	Meaning of error	Solution	Error level
		parameter not configured		
0x0012(18)	0x0A0C(2572)	Linear mode configuration-Period value parameter not configured	Configure the period valud	Warning
0x0012(18)	0x0A0D(2573)	The period value is zero	Set the period value to a non-zero value	Warning
0x0012(18)	0x0A0E(2574)	Negative limit is greater than positive limit	Set the positive limit greater than the negative limit	Warning
0x0012(18)	0x0B01(2817)	Gear ratio mode configuration-Axis ID exceeds maximum value	Configure the correct axis ID number	Warning
0x0012(18)	0x0B02(2818)	Gear ratio mode configuration-This axis does not belong to encoder axis	Configure the correct axis ID number	Warning
0x0012(18)	0x0B03(2819)	Gear ratio mode configuration-No axis number configured	Configure the correct axis ID number	Warning
0x0012(18)	0x0B04(2820)	Gear ratio mode configuration-Parameters for one revolution of encoder not configured	Configure the parameters for one revolution of encoder	Warning
0x0012(18)	0x0B05(2821)	Gear ratio mode configuration-Parameters for one revolution of worktable not configured	Configure the parameters for one revolution of worktable	Warning
0x0012(18)	0x0B06(2822)	Gear ratio mode configuration-Maximum value of gear ratio numerator parameter not configured	Configure the gear ratio numerator parameter	Warning
0x0012(18)	0x0B07(2823)	Gear ratio mode configuration-Gear ratio denominator parameter not configured	Configure the gear ratio denominator parameter	Warning
0x0013(19)	0x0001(1)	Axis number out of range	Check the axis number setting	Warning
0x0013(19)	0x0002(2)	Axis number does not exist in CANopen configuration or PDO configuration error	Check the CANopen configuration	Warning
0x0013(19)	0x0003(3)	Absolute position instruction speed is less than or equal to zero	Check the speed parameter in the command	Warning
0x0013(19)	0x0004(4)	Relative position instruction speed is less than or equal to zero	Check the speed parameter in the command	Warning
0x0013(19)	0x0005(5)	In velocity mode instruction the speed is less than or equal to zero	Check the speed parameter in the command	Warning
0x0013(19)	0x0006(6)	The jog instruction speed is	Check the speed parameter	Warning

Hex main error code (corresponding decimal)	Hex error subcode (corresponding decimal)	Meaning of error	Solution	Error level
		less than or equal to zero	in the command	
0x0013(19)	0x0007(7)	Absolute position instruction deceleration is less than or equal to zero	Check the acceleration and deceleration parameters in the command	Warning
0x0013(19)	0x0008(8)	Relative position instruction deceleration is less than or equal to zero	Check the deceleration parameters in the command	Warning
0x0013(19)	0x0009(9)	In velocity mode instruction the deceleration is less than or equal to zero	Check the deceleration parameters in the command	Warning
0x0013(19)	0x000A(10)	The jog instruction deceleration is less than or equal to zero	Check the deceleration parameters in the command	Warning
0x0013(19)	0x000B(11)	Homing failed	Check for disconnected wires	Warning
0x0013(19)	0x000C(12)	Homing timeout	Check the CANopen configuration PDO setting	Warning
0x0013(19)	0x000D(13)	The axis is not enabled, and the current instruction cannot be executed	Enable the CANopen axis	Warning
0x0013(19)	0x000E(14)	Not in "Fault Stop State", the reset axis error instruction cannot be executed	Check the status of the axis	Warning
0x0013(19)	0x000F(15)	The axis is in the "Stop" state, and the current instruction cannot be executed	Check the status of the axis	Warning
0x0013(19)	0x0010(16)	The axis is homing, and the current instruction cannot be executed	Check the status of the axis	Warning
0x0013(19)	0x0011(17)	The axis is moving continuously, and the current instruction cannot be executed	Check the status of the axis	Warning
0x0013(19)	0x0012(18)	The axis is positioning, and the current instruction cannot be executed	Check the status of the axis	Warning
0x0013(19)	0x0013(19)	The axis is in the "Fault Stop" state, and the current instruction cannot be executed	Check the status of the axis	Warning
0x0013(19)	0x0014(20)	Axis enable timeout	Check whether the CANopen configuration is correct	Warning
0x0013(19)	0x0015(21)	CANopen is not configured	Check whether the CANopen configuration is correct	Warning
0x0013(19)	0x0016(22)	Fault reset timeout	Check if the circuit connection is normal	Warning
0x0013(19)	0x0017(23)	SDO write timeout	Check if the circuit connection is normal	Warning

Hex main error code (corresponding decimal)	Hex error subcode (corresponding decimal)	Meaning of error	Solution	Error level
0x0013(19)	0x0018(24)	SDO read timeout	Check if the circuit connection is normal	Warning
0x0013(19)	0x0019(25)	SDO instruction error	Check if the circuit connection is normal	Warning
0x0013(19)	0x001A(26)	Software limit reached in axis operation	Check the software limit setting	Warning
0x0013(19)	0x001B(27)	Axis absolute positioning failure	Check if the circuit connection is normal	Warning
0x0013(19)	0x001C(28)	Axis relative positioning failure	Check if the circuit connection is normal	Warning
0x0013(19)	0x001D(29)	Homing speed set incorrectly	Check whether the command speed parameter setting is correct	Warning
0x0013(19)	0x001E(30)	Axis halt instruction execution timeout	Check if the circuit connection is normal	Warning
0x0013(19)	0x001F(31)	Homing approach speed is set incorrectly	Check if the setting for homing approach speed parameter in the CANopen configuration is correct	Warning
0x0013(19)	0x0020(32)	Homing acceleration set incorrectly	Check if the setting for homing acceleration parameter in the CANopen configuration is correct	Warning
0x0013(19)	0x0021(33)	Speed operation instruction execution failure	Check if the circuit connection is normal	Warning
0x0013(19)	0x0022(34)	Jog instruction execution failure	Check if the circuit connection is normal	Warning
0x0013(19)	0x0023(35)	Enable instruction is not allowed to be called on the same axis	Check if the Power command is called twice for the same axis number	Warning
0x0030(48)	0x0001(1)	Module configuration fault	Check whether the module network configuration corresponds to the physical configuration	General error
0x0030(48)	0x0002(2)	Incorrect module parameter setting	Check the module parameter setting	Warning
0x0031(49)	0x0001(1)	Digital input module configuration fault	Check whether the network configuration corresponds to the physical configuration of the module	General error
0x0031(49)	0x0002(2)	Digital input module parameter configuration failure	Check module parameter configuration	Warning
0x0031(49)	0x2001(8193)	Digital output module configuration failure	Check whether the network configuration corresponds to the physical configuration of the module	General error
0x0031(49)	0x2002(8194)	Digital output module parameter configuration	Check module parameter configuration	Warning

Hex main error code (corresponding decimal)	Hex error subcode (corresponding decimal)	Meaning of error	Solution	Error level
		failure		
0x0031(49)	0x2003(8195)	Digital output module output port power supply failure	Check the module output port power supply	Warning
0x0031(49)	0x2004(8196)	Digital output module output failure	Check whether the module output port load exceeds the specification range	Warning
0x0032(50)	0x0001(1)	Analog input module configuration failure	Check whether the network configuration corresponds to the physical configuration of the module	General error
0x0032(50)	0x0012(18)	Analog input channel 0 parameter configuration failure	Check channel 0 parameter configuration	Warning
0x0032(50)	0x0015(21)	Analog input channel 0 signal source open circuit fault	Check the physical connection of Channel 0 signal source	Warning
0x0032(50)	0x0016(22)	Analog input channel 0 sampling signal out of limit fault	Check whether the channel 0 sampling signal exceeds the chip limit	Warning
0x0032(50)	0x0017(23)	Analog input channel 0 sampling signal exceeds upper limit fault	Check whether the channel 0 sampling signal exceeds the upper range	Warning
0x0032(50)	0x0018(24)	Analog input channel 0 sampling signal exceeds the lower range	Check whether the channel 0 sampling signal exceeds the lower range	Warning
0x0032(50)	0x0022(34)	Analog input channel 1 parameter configuration failure	Check channel 1 parameter configuration	Warning
0x0032(50)	0x0025(37)	Analog input channel 1 signal source open circuit fault	Check the physical connection of Channel 1 signal source	Warning
0x0032(50)	0x0026(38)	Analog input channel 1 sampling signal out of limit fault	Check whether the channel 1 sampling signal exceeds the chip limit	Warning
0x0032(50)	0x0027(39)	Analog input channel 1 sampling signal exceeds upper limit fault	Check whether the channel 1 sampling signal exceeds the upper range	Warning
0x0032(50)	0x0028(40)	Analog input channel 1 sampling signal exceeds the lower range	Check whether the channel 1 sampling signal exceeds the lower range	Warning
0x0032(50)	0x0032(50)	Analog input channel 2 parameter configuration failure	Check channel 2 parameter configuration	Warning
0x0032(50)	0x0035(53)	Analog input channel 2 signal source open circuit fault	Check the physical connection of Channel 2 signal source	Warning
0x0032(50)	0x0036(54)	Analog input channel 2 sampling signal out of limit	Check whether the channel 2 sampling signal exceeds the	Warning

Hex main error code (corresponding decimal)	Hex error subcode (corresponding decimal)	Meaning of error	Solution	Error level
		fault	chip limit	
0x0032(50)	0x0037(55)	Analog input channel 2 sampling signal exceeds upper limit fault	Check whether the channel 2 sampling signal exceeds the upper range	Warning
0x0032(50)	0x0038(56)	Analog input channel 2 sampling signal exceeds the lower range	Check whether the channel 2 sampling signal exceeds the lower range	Warning
0x0032(50)	0x0042(66)	Analog input channel 3 parameter configuration failure	Check channel 3 parameter configuration	Warning
0x0032(50)	0x0045(69)	Analog input channel 3 signal source open circuit fault	Check the physical connection of Channel 3 signal source	Warning
0x0032(50)	0x0046(70)	Analog input channel 3 sampling signal out of limit fault	Check whether the channel 3 sampling signal exceeds the chip limit	Warning
0x0032(50)	0x0047(71)	Analog input channel 3 sampling signal exceeds upper limit fault	Check whether the channel 3 sampling signal exceeds the upper range	Warning
0x0032(50)	0x0048(72)	Analog input channel 3 sampling signal exceeds the lower range	Check whether the channel 3 sampling signal exceeds the lower range	Warning
0x0032(50)	0x2001(8193)	Analog output module configuration fault	Check whether the network configuration corresponds to the physical configuration of the module	General error
0x0032(50)	0x2003(8195)	Analog output module output port power supply failure	Check the module output port power supply	Warning
0x0032(50)	0x2012(8210)	Analog output channel 0 parameter configuration failure	Check channel 0 parameter configuration	Warning
0x0032(50)	0x2014(8212)	Analog output channel 0 output fault	Check channel 0 output for short/open circuit	Warning
0x0032(50)	0x2022(8226)	Analog output channel 1 parameter configuration failure	Check channel 1 parameter configuration	Warning
0x0032(50)	0x2024(8228)	Analog output channel 1 output fault	Check channel 1 output for short/open circuit	Warning
0x0032(50)	0x2032(8242)	Analog output channel 2 parameter configuration failure	Check channel 2 parameter configuration	Warning
0x0032(50)	0x2034(8244)	Analog output channel 2 output fault	Check channel 2 output for short/open circuit	Warning
0x0032(50)	0x2042(8258)	Analog output channel 3 parameter configuration failure	Check channel 3 parameter configuration	Warning
0x0032(50)	0x2044(8260)	Analog output channel 3 output fault	Check channel 3 output for short/open circuit	Warning

Hex main error code (corresponding decimal)	Hex error subcode (corresponding decimal)	Meaning of error	Solution	Error level
0x0033(51)	0x0001(1)	Temperature sampling module configuration failure	Check whether the network configuration corresponds to the physical configuration of the module	General error
0x0033(51)	0x0012(18)	Temperature sampling channel 0 parameter configuration failure	Check channel 0 parameter configuration	Warning
0x0033(51)	0x0015(21)	Temperature sampling channel 0 signal source open circuit fault	Check the physical connection of Channel 0 signal source	Warning
0x0033(51)	0x0017(23)	Temperature sampling channel 0 sampling signal exceeds the upper range	Check whether the channel 0 sampling signal exceeds the upper range	Warning
0x0033(51)	0x0018(24)	Temperature sampling channel 0 sampling signal exceeds the lower range	Check whether the channel 0 sampling signal exceeds the lower range	Warning
0x0033(51)	0x0022(34)	Temperature sampling channel 1 parameter configuration failure	Check channel 1 parameter configuration	Warning
0x0033(51)	0x0025(37)	Temperature sampling channel 1 signal source open circuit fault	Check the physical connection of Channel 1 signal source	Warning
0x0033(51)	0x0027(39)	Temperature sampling channel 1 sampling signal exceeds the upper range	Check whether the channel 1 sampling signal exceeds the upper range	Warning
0x0033(51)	0x0028(40)	Temperature sampling channel 1 sampling signal exceeds the lower range	Check whether the channel 1 sampling signal exceeds the lower range	Warning
0x0033(51)	0x0032(50)	Temperature sampling channel 2 parameter configuration failure	Check channel 2 parameter configuration	Warning
0x0033(51)	0x0035(53)	Temperature sampling channel 2 signal source open circuit fault	Check the physical connection of Channel 2 signal source	Warning
0x0033(51)	0x0037(55)	Temperature sampling channel 2 sampling signal exceeds the upper range	Check whether the channel 2 sampling signal exceeds the upper range	Warning
0x0033(51)	0x0038(56)	Temperature sampling channel 2 sampling signal exceeds the lower range	Check whether the channel 2 sampling signal exceeds the lower range	Warning
0x0033(51)	0x0042(66)	Temperature sampling channel 3 parameter configuration failure	Check channel 3 parameter configuration	Warning
0x0033(51)	0x0045(69)	Temperature sampling channel 3 signal source open circuit fault	Check the physical connection of Channel 3 signal source	Warning
0x0033(51)	0x0047(71)	Temperature sampling channel 3 sampling signal exceeds the upper range	Check whether the channel 3 sampling signal exceeds the upper range	Warning

Hex main error code (corresponding decimal)	Hex error subcode (corresponding decimal)	Meaning of error	Solution	Error level
0x0033(51)	0x0048(72)	Temperature sampling channel 3 sampling signal exceeds the lower range	Check whether the channel 3 sampling signal exceeds the lower range	Warning
0x0040(64)	0x0001(1)	Standard Modbus error, exception code 01, illegal function code	Check whether the configuration of function code accessed by master connected with PLC is legal	Warning
0x0040(64)	0x0002(2)	Standard Modbus error, exception code 02, illegal register address	Check whether the address configuration accessed by the master connected with PLC is legal	Warning
0x0040(64)	0x0003(3)	Standard Modbus error, exception code 03, data number error	Check whether the number configuration accessed by the master connected with PLC is legal	Warning
0x0040(64)	0x0004(4)	Standard Modbus error, exception code 04, slave device failure	Check whether the master connected with PLC is configured correctly	Warning
0x0040(64)	0x0005(5)	Communication timeout, the communication time exceeds the maximum communication time set by the user	Check whether the serial connection is normal	Warning
0x0040(64)	0x0007(7)	The communication connection is disconnected	Check whether the line connection is normal	Warning
0x0040(64)	0x0008(8)	The received data frame does not conform to the Modbus protocol	Check whether the baud rate, data bit and parity bit are configured correctly	Warning
0x0040(64)	0x0009(9)	CRC/LRC check error	Check whether the baud rate, data bit and parity bit are configured correctly	Warning
0x0040(64)	0x000A(10)	Element address overflow (the amount of data received or sent exceeds the storage space of the element)	Check the element address	Warning
0x0040(64)	0x000B(11)	The length of data received does not conform to the protocol or the number of elements exceeds the maximum limit specified by the function code	Check the master connected with PLC	Warning
0x0040(64)	0x000C(12)	The received slave address does not match the requested slave address	Check the slave connected with PLC	Warning
0x0040(64)	0x000D(13)	The received function code does not match the requested function code	Check the slave connected with PLC	Warning
0x0040(64)	0x000E(14)	Instruction execution failed	Check the parameter configuration of the host	Warning

Hex main error code (corresponding decimal)	Hex error subcode (corresponding decimal)	Meaning of error	Solution	Error level
			computer; Re-download the program	
0x0041(65)	0x0001(1)	Standard Modbus error, exception code 01, illegal function code	Check whether the configuration of function code accessed by master connected with PLC is legal	Warning
0x0041(65)	0x0002(2)	Standard Modbus error, exception code 02, illegal register address	Check whether the address configuration accessed by the master connected with PLC is legal	Warning
0x0041(65)	0x0003(3)	Standard Modbus error, exception code 03, data number error	Check whether the number configuration accessed by the master connected with PLC is legal	Warning
0x0041(65)	0x0004(4)	Standard Modbus error, exception code 04, slave device failure	Check whether the master connected with PLC is configured correctly	Warning
0x0041(65)	0x0005(5)	Communication timeout, the communication time exceeds the maximum communication time set by the user	Check whether the serial connection is normal	Warning
0x0041(65)	0x0007(7)	The communication connection is disconnected	Check whether the line connection is normal	Warning
0x0041(65)	0x0008(8)	The received data frame does not conform to the Modbus protocol	Check whether the baud rate, data bit and parity bit are configured correctly	Warning
0x0041(65)	0x0009(9)	CRC/LRC check error	Check whether the baud rate, data bit and parity bit are configured correctly	Warning
0x0041(65)	0x000A(10)	Element address overflow (the amount of data received or sent exceeds the storage space of the element)	Check the element address	Warning
0x0041(65)	0x000B(11)	The length of data received does not conform to the protocol or the number of elements exceeds the maximum limit specified by the function code	Check the master connected with PLC	Warning
0x0041(65)	0x000C(12)	The received slave address does not match the requested slave address	Check the slave connected with PLC	Warning
0x0041(65)	0x000D(13)	The received function code does not match the requested function code	Check the slave connected with PLC	Warning
0x0041(65)	0x000E(14)	Instruction execution failed	Check the parameter configuration of the host computer; Re-download the	Warning

Hex main error code (corresponding decimal)	Hex error subcode (corresponding decimal)	Meaning of error	Solution	Error level
			program	
0x0042(66)	0x0001(1)	Standard Modbus error, exception code 01, illegal function code	Check whether the configuration of function code accessed by master connected with PLC is legal	Warning
0x0042(66)	0x0002(2)	Standard Modbus error, exception code 02, illegal register address	Check whether the address configuration accessed by the master connected with PLC is legal	Warning
0x0042(66)	0x0003(3)	Standard Modbus error, exception code 03, data number error	Check whether the number configuration accessed by the master connected with PLC is legal	Warning
0x0042(66)	0x0004(4)	Standard Modbus error, exception code 04, slave device failure	Check whether the master connected with PLC is configured correctly	Warning
0x0042(66)	0x0005(5)	Communication timeout, the communication time exceeds the maximum communication time set by the user	Check whether the serial connection is normal	Warning
0x0042(66)	0x0007(7)	The communication connection is disconnected	Check whether the line connection is normal	Warning
0x0042(66)	0x0008(8)	The received data frame does not conform to the Modbus protocol	Check whether the baud rate, data bit and parity bit are configured correctly	Warning
0x0042(66)	0x0009(9)	CRC/LRC check error	Check whether the baud rate, data bit and parity bit are configured correctly	Warning
0x0042(66)	0x000A(10)	Element address overflow (the amount of data received or sent exceeds the storage space of the element)	Check the element address	Warning
0x0042(66)	0x000B(11)	The length of data received does not conform to the protocol or the number of elements exceeds the maximum limit specified by the function code	Check the master connected with PLC	Warning
0x0042(66)	0x000C(12)	The received slave address does not match the requested slave address	Check the slave connected with PLC	Warning
0x0042(66)	0x000D(13)	The received function code does not match the requested function code	Check the slave connected with PLC	Warning
0x0042(66)	0x000E(14)	Instruction execution failed	Check the parameter configuration of the host computer; Re-download the program	Warning

Hex main error code (corresponding decimal)	Hex error subcode (corresponding decimal)	Meaning of error	Solution	Error level
0x0043(67)	0x0001(1)	Standard Modbus error, exception code 01, illegal function code	Check whether the configuration of function code accessed by master connected with PLC is legal	Warning
0x0043(67)	0x0002(2)	Standard Modbus error, exception code 02, illegal register address	Check whether the address configuration accessed by the master connected with PLC is legal	Warning
0x0043(67)	0x0003(3)	Standard Modbus error, exception code 03, data number error	Check whether the number configuration accessed by the master connected with PLC is legal	Warning
0x0043(67)	0x0004(4)	Standard Modbus error, exception code 04, slave device failure	Check whether the master connected with PLC is configured correctly	Warning
0x0043(67)	0x0005(5)	Communication timeout, the communication time exceeds the maximum communication time set by the user	Check whether the serial connection is normal	Warning
0x0043(67)	0x0007(7)	The communication connection is disconnected	Check whether the line connection is normal	Warning
0x0043(67)	0x0008(8)	The received data frame does not conform to the Modbus protocol	Check whether the baud rate, data bit and parity bit are configured correctly	Warning
0x0043(67)	0x0009(9)	CRC/LRC check error	Check whether the baud rate, data bit and parity bit are configured correctly	Warning
0x0043(67)	0x000A(10)	Element address overflow (the amount of data received or sent exceeds the storage space of the element)	Check the element address	Warning
0x0043(67)	0x000B(11)	The length of data received does not conform to the protocol or the number of elements exceeds the maximum limit specified by the function code	Check the master connected with PLC	Warning
0x0043(67)	0x000C(12)	The received slave address does not match the requested slave address	Check the slave connected with PLC	Warning
0x0043(67)	0x000D(13)	The received function code does not match the requested function code	Check the slave connected with PLC	Warning
0x0043(67)	0x000E(14)	Instruction execution failed	Check the parameter configuration of the host computer; Re-download the program	Warning
0x0044(68)	0x0001(1)	Standard Modbus error,	Check whether the	Warning

Hex main error code (corresponding decimal)	Hex error subcode (corresponding decimal)	Meaning of error	Solution	Error level
		exception code 01, illegal function code	configuration of function code accessed by master connected with PLC is legal	
0x0044(68)	0x0002(2)	Standard Modbus error, exception code 02, illegal register address	Check whether the address configuration accessed by the master connected with PLC is legal	Warning
0x0044(68)	0x0003(3)	Standard Modbus error, exception code 03, data number error	Check whether the number configuration accessed by the master connected with PLC is legal	Warning
0x0044(68)	0x0004(4)	Standard Modbus error, exception code 04, slave device failure	Check whether the master connected with PLC is configured correctly	Warning
0x0044(68)	0x0005(5)	Communication timeout, the communication time exceeds the maximum communication time set by the user	Check whether the serial connection is normal	Warning
0x0044(68)	0x0007(7)	The communication connection is disconnected	Check whether the line connection is normal	Warning
0x0044(68)	0x0008(8)	The received data frame does not conform to the Modbus protocol	Check whether the baud rate, data bit and parity bit are configured correctly	Warning
0x0044(68)	0x0009(9)	CRC/LRC check error	Check whether the baud rate, data bit and parity bit are configured correctly	Warning
0x0044(68)	0x000A(10)	Element address overflow (the amount of data received or sent exceeds the storage space of the element)	Check the element address	Warning
0x0044(68)	0x000B(11)	The length of data received does not conform to the protocol or the number of elements exceeds the maximum limit specified by the function code	Check the master connected with PLC	Warning
0x0044(68)	0x000C(12)	The received slave address does not match the requested slave address	Check the slave connected with PLC	Warning
0x0044(68)	0x000D(13)	The received function code does not match the requested function code	Check the slave connected with PLC	Warning
0x0044(68)	0x000E(14)	Instruction execution failed	Check the parameter configuration of the host computer; Re-download the program	Warning
0x0045(69)	0x0001(1)	Standard Modbus error, exception code 01, illegal	Check whether the configuration of function	Warning

Hex main error code (corresponding decimal)	Hex error subcode (corresponding decimal)	Meaning of error	Solution	Error level
		function code	code accessed by master connected with PLC is legal	
0x0045(69)	0x0002(2)	Standard Modbus error, exception code 02, illegal register address	Check whether the address configuration accessed by the master connected with PLC is legal	Warning
0x0045(69)	0x0003(3)	Standard Modbus error, exception code 03, data number error	Check whether the number configuration accessed by the master connected with PLC is legal	Warning
0x0045(69)	0x0004(4)	Standard Modbus error, exception code 04, slave device failure	Check whether the master connected with PLC is configured correctly	Warning
0x0045(69)	0x0005(5)	Communication timeout, the communication time exceeds the maximum communication time set by the user	Check whether the serial connection is normal	Warning
0x0045(69)	0x0007(7)	The communication connection is disconnected	Check whether the line connection is normal	Warning
0x0045(69)	0x0008(8)	The received data frame does not conform to the Modbus protocol	Check whether the baud rate, data bit and parity bit are configured correctly	Warning
0x0045(69)	0x0009(9)	CRC/LRC check error	Check whether the baud rate, data bit and parity bit are configured correctly	Warning
0x0045(69)	0x000A(10)	Element address overflow (the amount of data received or sent exceeds the storage space of the element)	Check the element address	Warning
0x0045(69)	0x000B(11)	The length of data received does not conform to the protocol or the number of elements exceeds the maximum limit specified by the function code	Check the master connected with PLC	Warning
0x0045(69)	0x000C(12)	The received slave address does not match the requested slave address	Check the slave connected with PLC	Warning
0x0045(69)	0x000D(13)	The received function code does not match the requested function code	Check the slave connected with PLC	Warning
0x0045(69)	0x000E(14)	Instruction execution failed	Check the parameter configuration of the host computer; Re-download the program	Warning
0x0046(70)	0x0001(1)	COM port parameters not configured	Choose the serial freeport in the host computer and configure the parameters of	Warning

Hex main error code (corresponding decimal)	Hex error subcode (corresponding decimal)	Meaning of error	Solution	Error level
			the serial port	
0x0046(70)	0x0002(2)	Transmit length or receive length setting error	Check the transmit length or receive length	Warning
0x0046(70)	0x0003(3)	Element address overflow (the amount of data received or sent exceeds the storage space of the element)	Check the receive or transmit buffer element address and the transmit or receive length of instruction parameters	Warning
0x0046(70)	0x0004(4)	Port setting error	Check whether the Port setting in the instruction is correct	Warning
0x0046(70)	0x0005(5)	Transmit instruction execution failed	Retry	Warning
0x0046(70)	0x0006(6)	Receive instruction execution failed	Retry	Warning
0x0046(70)	0x0007(7)	Incomplete received data	Check the length of data transmitted by the sender	Warning
0x0046(70)	0x0008(8)	Receive data timeout	Check whether the serial line connection is normal	Warning
0x0046(70)	0x0009(9)	Instruction execution failed	Retry	Warning
0x0047(71)	0x0001(1)	COM port parameters not configured	Choose the serial freeport in the host computer and configure the parameters of the serial port	Warning
0x0047(71)	0x0002(2)	Transmit length or receive length setting error	Check the transmit length or receive length	Warning
0x0047(71)	0x0003(3)	Element address overflow (the amount of data received or sent exceeds the storage space of the element)	Check the receive or transmit buffer element address and the transmit or receive length of instruction parameters	Warning
0x0047(71)	0x0004(4)	Port setting error	Check whether the Port setting in the instruction is correct	Warning
0x0047(71)	0x0005(5)	Transmit instruction execution failed	Retry	Warning
0x0047(71)	0x0006(6)	Receive instruction execution failed	Retry	Warning
0x0047(71)	0x0007(7)	Incomplete received data	Check the length of data transmitted by the sender	Warning
0x0047(71)	0x0008(8)	Receive data timeout	Check whether the serial line connection is normal	Warning
0x0047(71)	0x0009(9)	Instruction execution failed	Retry	Warning
0x0048(72)	0x0001(1)	COM port parameters not configured	Choose the serial freeport in the host computer and configure the parameters of the serial port	Warning
0x0048(72)	0x0002(2)	Transmit length or receive length setting error	Check the transmit length or receive length	Warning

Hex main error code (corresponding decimal)	Hex error subcode (corresponding decimal)	Meaning of error	Solution	Error level
0x0048(72)	0x0003(3)	Element address overflow (the amount of data received or sent exceeds the storage space of the element)	Check the receive or transmit buffer element address and the transmit or receive length of instruction parameters	Warning
0x0048(72)	0x0004(4)	Port setting error	Check whether the Port setting in the instruction is correct	Warning
0x0048(72)	0x0005(5)	Transmit instruction execution failed	Retry	Warning
0x0048(72)	0x0006(6)	Receive instruction execution failed	Retry	Warning
0x0048(72)	0x0007(7)	Incomplete received data	Check the length of data transmitted by the sender	Warning
0x0048(72)	0x0008(8)	Receive data timeout	Check whether the serial line connection is normal	Warning
0x0048(72)	0x0009(9)	Instruction execution failed	Retry	Warning
0x0050(80)	0x0001(1)	CANopen communication error	Please check whether the CAN network line connection is normal, ensure that there is no reverse connection, short connection or open circuit between CANH and CANL, and check whether the terminal resistance is connected correctly, and whether the baud rate of CAN communication matches.	General error
0x0050(80)	0x0002(2)	CANopen configuration error	Check whether the host computer configuration matches the actual situation	General error
0x0050(80)	0x0003(3)	CANopen load rate is too high	Detect whether too many PDOs are configured, and there are devices on the fieldbus that transmit CAN messages autonomously, such as CAN analyzers or multiple CANopen masters. This situation may lead to poor communication status, data loss and other problems.	Warning
0x0080(128)	0x1020(4128)	Wrong number of module matches	Update the version of the host computer or contact a technician	Serious error
0x0080(128)	0x1040(4160)	Connection 0 module length matching exception	Update the version of the host computer or contact a technician	Serious error
0x0080(128)	0x1041(4161)	Connection 1 module length matching exception	Update the version of the host computer or contact a technician	Serious error

Hex main error code (corresponding decimal)	Hex error subcode (corresponding decimal)	Meaning of error	Solution	Error level
0x0080(128)	0x1042(4162)	Connection 2 module length matching exception	Update the version of the host computer or contact a technician	Serious error
0x0080(128)	0x1043(4163)	Connection 3 module length matching exception	Update the version of the host computer or contact a technician	Serious error
0x0080(128)	0x1044(4164)	Connection 4 module length matching exception	Update the version of the host computer or contact a technician	Serious error
0x0080(128)	0x1045(4165)	Connection 5 module length matching exception	Update the version of the host computer or contact a technician	Serious error
0x0080(128)	0x1046(4166)	Connection 6 module length matching exception	Update the version of the host computer or contact a technician	Serious error
0x0080(128)	0x1047(4167)	Connection 7 module length matching exception	Update the version of the host computer or contact a technician	Serious error
0x0080(128)	0x1048(4168)	Connection 8 module length matching exception	Update the version of the host computer or contact a technician	Serious error
0x0080(128)	0x1049(4169)	Connection 9 module length matching exception	Update the version of the host computer or contact a technician	Serious error
0x0080(128)	0x104A(4170)	Connection 10 module length matching exception	Update the version of the host computer or contact a technician	Serious error
0x0080(128)	0x104B(4171)	Connection 11 module length matching exception	Update the version of the host computer or contact a technician	Serious error
0x0080(128)	0x104C(4172)	Connection 12 module length matching exception	Update the version of the host computer or contact a technician	Serious error
0x0080(128)	0x104D(4173)	Connection 13 module length matching exception	Update the version of the host computer or contact a technician	Serious error
0x0080(128)	0x104E(4174)	Connection 14 module length matching exception	Update the version of the host computer or contact a technician	Serious error
0x0080(128)	0x104F(4175)	Connection 15 module length matching exception	Update the version of the host computer or contact a technician	Serious error
0x0080(128)	0x1060(4192)	Connection 0 module input length exception	Update the version of the host computer or contact a technician	Serious error
0x0080(128)	0x1061(4193)	Connection 1 module input length exception	Update the version of the host computer or contact a technician	Serious error
0x0080(128)	0x1062(4194)	Connection 2 module input	Update the version of the	Serious

Hex main error code (corresponding decimal)	Hex error subcode (corresponding decimal)	Meaning of error	Solution	Error level
		length exception	host computer or contact a technician	error
0x0080(128)	0x1063(4195)	Connection 3 module input length exception	Update the version of the host computer or contact a technician	Serious error
0x0080(128)	0x1064(4196)	Connection 4 module input length exception	Update the version of the host computer or contact a technician	Serious error
0x0080(128)	0x1065(4197)	Connection 5 module input length exception	Update the version of the host computer or contact a technician	Serious error
0x0080(128)	0x1066(4198)	Connection 6 module input length exception	Update the version of the host computer or contact a technician	Serious error
0x0080(128)	0x1067(4199)	Connection 7 module input length exception	Update the version of the host computer or contact a technician	Serious error
0x0080(128)	0x1068(4200)	Connection 8 module input length exception	Update the version of the host computer or contact a technician	Serious error
0x0080(128)	0x1069(4201)	Connection 9 module input length exception	Update the version of the host computer or contact a technician	Serious error
0x0080(128)	0x106A(4202)	Connection 10 module input length exception	Update the version of the host computer or contact a technician	Serious error
0x0080(128)	0x106B(4203)	Connection 11 module input length exception	Update the version of the host computer or contact a technician	Serious error
0x0080(128)	0x106C(4204)	Connection 12 module input length exception	Update the version of the host computer or contact a technician	Serious error
0x0080(128)	0x106D(4205)	Connection 13 module input length exception	Update the version of the host computer or contact a technician	Serious error
0x0080(128)	0x106E(4206)	Connection 14 module input length exception	Update the version of the host computer or contact a technician	Serious error
0x0080(128)	0x106F(4207)	Connection 15 module input length exception	Update the version of the host computer or contact a technician	Serious error
0x0080(128)	0x1080(4224)	Connection 0 module output length exception	Update the version of the host computer or contact a technician	Serious error
0x0080(128)	0x1081(4225)	Connection 1 module output length exception	Update the version of the host computer or contact a technician	Serious error
0x0080(128)	0x1082(4226)	Connection 2 module output length exception	Update the version of the host computer or contact a	Serious error

Hex main error code (corresponding decimal)	Hex error subcode (corresponding decimal)	Meaning of error	Solution	Error level
			technician	
0x0080(128)	0x1083(4227)	Connection 3 module output length exception	Update the version of the host computer or contact a technician	Serious error
0x0080(128)	0x1084(4228)	Connection 4 module output length exception	Update the version of the host computer or contact a technician	Serious error
0x0080(128)	0x1085(4229)	Connection 5 module output length exception	Update the version of the host computer or contact a technician	Serious error
0x0080(128)	0x1086(4230)	Connection 6 module output length exception	Update the version of the host computer or contact a technician	Serious error
0x0080(128)	0x1087(4231)	Connection 7 module output length exception	Update the version of the host computer or contact a technician	Serious error
0x0080(128)	0x1088(4232)	Connection 8 module output length exception	Update the version of the host computer or contact a technician	Serious error
0x0080(128)	0x1089(4233)	Connection 9 module output length exception	Update the version of the host computer or contact a technician	Serious error
0x0080(128)	0x108A(4234)	Connection 10 module output length exception	Update the version of the host computer or contact a technician	Serious error
0x0080(128)	0x108B(4235)	Connection 11 module output length exception	Update the version of the host computer or contact a technician	Serious error
0x0080(128)	0x108C(4236)	Connection 12 module output length exception	Update the version of the host computer or contact a technician	Serious error
0x0080(128)	0x108D(4237)	Connection 13 module output length exception	Update the version of the host computer or contact a technician	Serious error
0x0080(128)	0x108E(4238)	Connection 14 module output length exception	Update the version of the host computer or contact a technician	Serious error
0x0080(128)	0x108F(4239)	Connection 15 module output length exception	Update the version of the host computer or contact a technician	Serious error
0x0080(128)	0x10A0(4256)	Connection 0 element matching exception	Update the version of the host computer or contact a technician	Serious error
0x0080(128)	0x10A1(4257)	Connection 1 element matching exception	Update the version of the host computer or contact a technician	Serious error
0x0080(128)	0x10A2(4258)	Connection 2 element matching exception	Update the version of the host computer or contact a technician	Serious error

Hex main error code (corresponding decimal)	Hex error subcode (corresponding decimal)	Meaning of error	Solution	Error level
0x0080(128)	0x10A3(4259)	Connection 3 element matching exception	Update the version of the host computer or contact a technician	Serious error
0x0080(128)	0x10A4(4260)	Connection 4 element matching exception	Update the version of the host computer or contact a technician	Serious error
0x0080(128)	0x10A5(4261)	Connection 5 element matching exception	Update the version of the host computer or contact a technician	Serious error
0x0080(128)	0x10A6(4262)	Connection 6 element matching exception	Update the version of the host computer or contact a technician	Serious error
0x0080(128)	0x10A7(4263)	Connection 7 element matching exception	Update the version of the host computer or contact a technician	Serious error
0x0080(128)	0x10A8(4264)	Connection 8 element matching exception	Update the version of the host computer or contact a technician	Serious error
0x0080(128)	0x10A9(4265)	Connection 9 element matching exception	Update the version of the host computer or contact a technician	Serious error
0x0080(128)	0x10AA(4266)	Connection 10 element matching exception	Update the version of the host computer or contact a technician	Serious error
0x0080(128)	0x10AB(4267)	Connection 11 element matching exception	Update the version of the host computer or contact a technician	Serious error
0x0080(128)	0x10AC(4268)	Connection 12 element matching exception	Update the version of the host computer or contact a technician	Serious error
0x0080(128)	0x10AD(4269)	Connection 13 element matching exception	Update the version of the host computer or contact a technician	Serious error
0x0080(128)	0x10AE(4270)	Connection 14 element matching exception	Update the version of the host computer or contact a technician	Serious error
0x0080(128)	0x10AF(4271)	Connection 15 element matching exception	Update the version of the host computer or contact a technician	Serious error
0x0080(128)	0x3000(12288)	Connection 0 connection not established	Check wires, connection port, connection mode, and attribute ID	Warning
0x0080(128)	0x3001(12289)	Connection 1 connection not established	Check wires, connection port, connection mode, and attribute ID	Warning
0x0080(128)	0x3002(12290)	Connection 2 connection not established	Check wires, connection port, connection mode, and attribute ID	Warning
0x0080(128)	0x3003(12291)	Connection 3 connection not	Check wires, connection port,	Warning

Hex main error code (corresponding decimal)	Hex error subcode (corresponding decimal)	Meaning of error	Solution	Error level
		established	connection mode, and attribute ID	
0x0080(128)	0x3004(12292)	Connection 4 connection not established	Check wires, connection port, connection mode, and attribute ID	Warning
0x0080(128)	0x3005(12293)	Connection 5 connection not established	Check wires, connection port, connection mode, and attribute ID	Warning
0x0080(128)	0x3006(12294)	Connection 6 connection not established	Check wires, connection port, connection mode, and attribute ID	Warning
0x0080(128)	0x3007(12295)	Connection 7 connection not established	Check wires, connection port, connection mode, and attribute ID	Warning
0x0080(128)	0x3008(12296)	Connection 8 connection not established	Check wires, connection port, connection mode, and attribute ID	Warning
0x0080(128)	0x3009(12297)	Connection 9 connection not established	Check wires, connection port, connection mode, and attribute ID	Warning
0x0080(128)	0x300A(12298)	Connection 10 connection not established	Check wires, connection port, connection mode, and attribute ID	Warning
0x0080(128)	0x300B(12299)	Connection 11 connection not established	Check wires, connection port, connection mode, and attribute ID	Warning
0x0080(128)	0x300C(12300)	Connection 12 connection not established	Check wires, connection port, connection mode, and attribute ID	Warning
0x0080(128)	0x300D(12301)	Connection 13 connection not established	Check wires, connection port, connection mode, and attribute ID	Warning
0x0080(128)	0x300E(12302)	Connection 14 connection not established	Check wires, connection port, connection mode, and attribute ID	Warning
0x0080(128)	0x300F(12303)	Connection 15 connection not established	Check wires, connection port, connection mode, and attribute ID	Warning
0x0080(128)	0x3020(12320)	Connection 0 path error	Check the configuration path	Warning
0x0080(128)	0x3021(12321)	Connection 1 path error	Check the configuration path	Warning
0x0080(128)	0x3022(12322)	Connection 2 path error	Check the configuration path	Warning
0x0080(128)	0x3023(12323)	Connection 3 path error	Check the configuration path	Warning
0x0080(128)	0x3024(12324)	Connection 4 path error	Check the configuration path	Warning
0x0080(128)	0x3025(12325)	Connection 5 path error	Check the configuration path	Warning
0x0080(128)	0x3026(12326)	Connection 6 path error	Check the configuration path	Warning
0x0080(128)	0x3027(12327)	Connection 7 path error	Check the configuration path	Warning

Hex main error code (corresponding decimal)	Hex error subcode (corresponding decimal)	Meaning of error	Solution	Error level
0x0080(128)	0x3028(12328)	Connection 8 path error	Check the configuration path	Warning
0x0080(128)	0x3029(12329)	Connection 9 path error	Check the configuration path	Warning
0x0080(128)	0x302A(12330)	Connection 10 path error	Check the configuration path	Warning
0x0080(128)	0x302B(12331)	Connection 11 path error	Check the configuration path	Warning
0x0080(128)	0x302C(12332)	Connection 12 path error	Check the configuration path	Warning
0x0080(128)	0x302D(12333)	Connection 13 path error	Check the configuration path	Warning
0x0080(128)	0x302E(12334)	Connection 14 path error	Check the configuration path	Warning
0x0080(128)	0x302F(12335)	Connection 15 path error	Check the configuration path	Warning
0x0080(128)	0x3040(12352)	Connection 0 transmission data size mismatch	Check the size of matching data between the sender and receiver	Warning
0x0080(128)	0x3041(12353)	Connection 1 transmission data size mismatch	Check the size of matching data between the sender and receiver	Warning
0x0080(128)	0x3042(12354)	Connection 2 transmission data size mismatch	Check the size of matching data between the sender and receiver	Warning
0x0080(128)	0x3043(12355)	Connection 3 transmission data size mismatch	Check the size of matching data between the sender and receiver	Warning
0x0080(128)	0x3044(12356)	Connection 4 transmission data size mismatch	Check the size of matching data between the sender and receiver	Warning
0x0080(128)	0x3045(12357)	Connection 5 transmission data size mismatch	Check the size of matching data between the sender and receiver	Warning
0x0080(128)	0x3046(12358)	Connection 6 transmission data size mismatch	Check the size of matching data between the sender and receiver	Warning
0x0080(128)	0x3047(12359)	Connection 7 transmission data size mismatch	Check the size of matching data between the sender and receiver	Warning
0x0080(128)	0x3048(12360)	Connection 8 transmission data size mismatch	Check the size of matching data between the sender and receiver	Warning
0x0080(128)	0x3049(12361)	Connection 9 transmission data size mismatch	Check the size of matching data between the sender and receiver	Warning
0x0080(128)	0x304A(12362)	Connection 10 transmission data size mismatch	Check the size of matching data between the sender and receiver	Warning
0x0080(128)	0x304B(12363)	Connection 11 transmission data size mismatch	Check the size of matching data between the sender and receiver	Warning
0x0080(128)	0x304C(12364)	Connection 12 transmission data size mismatch	Check the size of matching data between the sender and receiver	Warning

Hex main error code (corresponding decimal)	Hex error subcode (corresponding decimal)	Meaning of error	Solution	Error level
0x0080(128)	0x304D(12365)	Connection 13 transmission data size mismatch	Check the size of matching data between the sender and receiver	Warning
0x0080(128)	0x304E(12366)	Connection 14 transmission data size mismatch	Check the size of matching data between the sender and receiver	Warning
0x0080(128)	0x304F(12367)	Connection 15 transmission data size mismatch	Check the size of matching data between the sender and receiver	Warning
0x0080(128)	0x3060(12384)	Connection 0 other errors	Contact technicians	Warning
0x0080(128)	0x3061(12385)	Connection 1 other errors	Contact technicians	Warning
0x0080(128)	0x3062(12386)	Connection 2 other errors	Contact technicians	Warning
0x0080(128)	0x3063(12387)	Connection 3 other errors	Contact technicians	Warning
0x0080(128)	0x3064(12388)	Connection 4 other errors	Contact technicians	Warning
0x0080(128)	0x3065(12389)	Connection 5 other errors	Contact technicians	Warning
0x0080(128)	0x3066(12390)	Connection 6 other errors	Contact technicians	Warning
0x0080(128)	0x3067(12391)	Connection 7 other errors	Contact technicians	Warning
0x0080(128)	0x3068(12392)	Connection 8 other errors	Contact technicians	Warning
0x0080(128)	0x3069(12393)	Connection 9 other errors	Contact technicians	Warning
0x0080(128)	0x306A(12394)	Connection 10 other errors	Contact technicians	Warning
0x0080(128)	0x306B(12395)	Connection 11 other errors	Contact technicians	Warning
0x0080(128)	0x306C(12396)	Connection 12 other errors	Contact technicians	Warning
0x0080(128)	0x306D(12397)	Connection 13 other errors	Contact technicians	Warning
0x0080(128)	0x306E(12398)	Connection 14 other errors	Contact technicians	Warning
0x0080(128)	0x306F(12399)	Connection 15 other errors	Contact technicians	Warning
0x0080(128)	0x30A0(12448)	Connection 0 communication timeout	Check wires, connection port, connection mode, and attribute ID	Warning
0x0080(128)	0x30A1(12449)	Connection 1 communication timeout	Check wires, connection port, connection mode, and attribute ID	Warning
0x0080(128)	0x30A2(12450)	Connection 2 communication timeout	Check wires, connection port, connection mode, and attribute ID	Warning
0x0080(128)	0x30A3(12451)	Connection 3 communication timeout	Check wires, connection port, connection mode, and attribute ID	Warning
0x0080(128)	0x30A4(12452)	Connection 4 communication timeout	Check wires, connection port, connection mode, and attribute ID	Warning
0x0080(128)	0x30A5(12453)	Connection 5 communication timeout	Check wires, connection port, connection mode, and attribute ID	Warning

Hex main error code (corresponding decimal)	Hex error subcode (corresponding decimal)	Meaning of error	Solution	Error level
0x0080(128)	0x30A6(12454)	Connection 6 communication timeout	Check wires, connection port, connection mode, and attribute ID	Warning
0x0080(128)	0x30A7(12455)	Connection 7 communication timeout	Check wires, connection port, connection mode, and attribute ID	Warning
0x0080(128)	0x30A8(12456)	Connection 8 communication timeout	Check wires, connection port, connection mode, and attribute ID	Warning
0x0080(128)	0x30A9(12457)	Connection 9 communication timeout	Check wires, connection port, connection mode, and attribute ID	Warning
0x0080(128)	0x30AA(12458)	Connection 10 communication timeout	Check wires, connection port, connection mode, and attribute ID	Warning
0x0080(128)	0x30AB(12459)	Connection 11 communication timeout	Check wires, connection port, connection mode, and attribute ID	Warning
0x0080(128)	0x30AC(12460)	Connection 12 communication timeout	Check wires, connection port, connection mode, and attribute ID	Warning
0x0080(128)	0x30AD(12461)	Connection 13 communication timeout	Check wires, connection port, connection mode, and attribute ID	Warning
0x0080(128)	0x30AE(12462)	Connection 14 communication timeout	Check wires, connection port, connection mode, and attribute ID	Warning
0x0080(128)	0x30AF(12463)	Connection 15 communication timeout	Check wires, connection port, connection mode, and attribute ID	Warning
0x0080(128)	0x4020(16416)	Network configuration exception	Contact technicians	Serious error
0x0080(128)	0x4040(16448)	Network initialization exception	Contact technicians	Serious error
0x0080(128)	0x4060(16480)	Thread attribute initialization failed	Contact technicians	Serious error
0x0080(128)	0x4080(16512)	Thread request stack failure	Contact technicians	Serious error
0x0080(128)	0x40A0(16544)	Thread setting scheduling policy failed	Contact technicians	Serious error
0x0080(128)	0x40C0(16576)	Thread priority setting failed	Contact technicians	Serious error
0x0080(128)	0x40E0(16608)	Failed to set parent thread inheritance policy	Contact technicians	Serious error
0x0080(128)	0x4100(16640)	Failed to create thread	Contact technicians	Serious error
0x0090(144)	0x0001(1)	Failed to apply for master	Check whether the card software matches the background version; Restart	General error

Hex main error code (corresponding decimal)	Hex error subcode (corresponding decimal)	Meaning of error	Solution	Error level
			PLC	
0x0090(144)	0x0002(2)	Wrong master version	Check whether the single board software matches the background version	General error
0x0090(144)	0x0003(3)	The number of PDO entries sent exceeds the maximum limit	Check whether the number of PDO entries sent exceeds the maximum limit	General error
0x0090(144)	0x0004(4)	The number of PDO configuration objects sent exceeds the maximum limit	Check whether the number of PDO configuration objects sent exceeds the maximum limit	General error
0x0090(144)	0x0005(5)	The number of PDO entries received exceeds the maximum limit	Check whether the number of PDO entries received exceeds the maximum limit	General error
0x0090(144)	0x0006(6)	The number of PDO configuration objects received exceeds the maximum limit	Check whether the number of PDO configuration objects received exceeds the maximum limit	General error
0x0090(144)	0x0007(7)	The number of startup parameters exceeds the maximum limit	Check whether the number of startup parameters exceeds the maximum limit	General error
0x0090(144)	0x0008(8)	The number of servos exceeds the maximum limit	Check whether the number of servos configured exceeds the maximum limit	General error
0x0090(144)	0x0009(9)	The number of slaves exceeds the maximum limit	Check whether the number of slaves configured exceeds the maximum limit	General error
0x0090(144)	0x000A(10)	Wrong configuration type	Reserved	General error
0x0090(144)	0x000B(11)	Configured number does not match actual number of connections	Check whether the actual number of connected slaves is less than the configured number of slaves	General error
0x0090(144)	0x000C(12)	DC mode is not supported by slaves	Reserved	General error
0x0090(144)	0x000D(13)	Wrong Slave type	Check whether the devices in the configuration match the actual connected devices	General error
0x0090(144)	0x000E(14)	The number of mapped slaves exceeds the set value	Check whether the actual number of connected slaves is greater than the configured number of slaves	General error
0x0090(144)	0x000F(15)	Mapping slave transmit PDO communication exception	Reserved	General error
0x0090(144)	0x0010(16)	Mapping slave receive PDO communication exception	Reserved	General error
0x0090(144)	0x0011(17)	Slave PDO offline	Check whether the network among slaves is disconnected; Check whether	General error

Hex main error code (corresponding decimal)	Hex error subcode (corresponding decimal)	Meaning of error	Solution	Error level
			the slaves are powered off	
0x0090(144)	0x0012(18)	Failed to initialize slave parameters	Contact the original manufacturer	General error
0x0090(144)	0x0013(19)	Network connection failure	Check whether the slaves are connected; Check whether all slaves are powered off	General error
0x0090(144)	0x0014(20)	Unable to identify the number of slaves	Reserved	General error
0x0090(144)	0x0015(21)	Aperiodic communication timeout	Reserved	Warning
0x0090(144)	0x0016(22)	Failed to apply for master	Contact the original manufacturer	Serious error
0x0090(144)	0x0017(23)	Illegal IO mapping	Reserved	Serious error
0x00A0(160)	0x0001(1)	Standard Modbus error, exception code 01, illegal function code	Check whether the configuration of function code accessed by master connected with PLC is legal	Warning
0x00A0(160)	0x0002(2)	Standard Modbus error, exception code 02, illegal register address	Check whether the address configuration accessed by the master connected with PLC is legal	Warning
0x00A0(160)	0x0003(3)	Standard Modbus error, exception code 03, data number error	Check whether the number configuration accessed by the master connected with PLC is legal	Warning
0x00A0(160)	0x0004(4)	Standard Modbus error, exception code 04, slave device failure	Check whether the master connected with PLC is configured correctly	Warning
0x00A0(160)	0x0005(5)	Communication timeout, the communication time exceeds the maximum communication time set by the user	Check whether the serial connection is normal	Warning
0x00A0(160)	0x0006(6)	Modbus TCP master-slave connection timeout	Check whether the connection of network cable is normal, and whether the ip and port number are set correctly	Warning
0x00A0(160)	0x0007(7)	The communication connection is disconnected	Check whether the line connection is normal	Warning
0x00A0(160)	0x0008(8)	The received data frame does not conform to the Modbus protocol	Check whether the baud rate, data bit and parity bit are configured correctly	Warning
0x00A0(160)	0x0009(9)	CRC/LRC check error	Check whether the baud rate, data bit and parity bit are configured correctly	Warning

Hex main error code (corresponding decimal)	Hex error subcode (corresponding decimal)	Meaning of error	Solution	Error level
0x00A0(160)	0x000A(10)	Element address overflow (the amount of data received or sent exceeds the storage space of the element)	Check the element address	Warning
0x00A0(160)	0x000B(11)	The length of data received does not conform to the protocol or the number of elements exceeds the maximum limit specified by the function code	Check the master connected with PLC	Warning
0x00A0(160)	0x000C(12)	The received slave address does not match the requested slave address	Check the slave connected with PLC	Warning
0x00A0(160)	0x000D(13)	The received function code does not match the requested function code	Check the slave connected with PLC	Warning
0x00A0(160)	0x000E(14)	Instruction execution failed	Check the parameter configuration of the host computer; Re-download the program	Warning
0x00A1(161)	0x0001(1)	Standard Modbus error, exception code 01, illegal function code	Check whether the configuration of function code accessed by master connected with PLC is legal	Warning
0x00A1(161)	0x0002(2)	Standard Modbus error, exception code 02, illegal register address	Check whether the address configuration accessed by the master connected with PLC is legal	Warning
0x00A1(161)	0x0003(3)	Standard Modbus error, exception code 03, data number error	Check whether the number configuration accessed by the master connected with PLC is legal	Warning
0x00A1(161)	0x0004(4)	Standard Modbus error, exception code 04, slave device failure	Check whether the master connected with PLC is configured correctly	Warning
0x00A1(161)	0x0005(5)	Communication timeout, the communication time exceeds the maximum communication time set by the user	Check whether the serial connection is normal	Warning
0x00A1(161)	0x0006(6)	Modbus TCP master-slave connection timeout	Check whether the connection of network cable is normal, and whether the ip and port number are set correctly	Warning
0x00A1(161)	0x0007(7)	The communication connection is disconnected	Check whether the line connection is normal	Warning
0x00A1(161)	0x0008(8)	The received data frame does not conform to the	Check whether the baud rate, data bit and parity bit are	Warning

Hex main error code (corresponding decimal)	Hex error subcode (corresponding decimal)	Meaning of error	Solution	Error level
		Modbus protocol	configured correctly	
0x00A1(161)	0x0009(9)	CRC/LRC check error	Check whether the baud rate, data bit and parity bit are configured correctly	Warning
0x00A1(161)	0x000A(10)	Element address overflow (the amount of data received or sent exceeds the storage space of the element)	Check the element address	Warning
0x00A1(161)	0x000B(11)	The length of data received does not conform to the protocol or the number of elements exceeds the maximum limit specified by the function code	Check the master connected with PLC	Warning
0x00A1(161)	0x000C(12)	The received slave address does not match the requested slave address	Check the slave connected with PLC	Warning
0x00A1(161)	0x000D(13)	The received function code does not match the requested function code	Check the slave connected with PLC	Warning
0x00A1(161)	0x000E(14)	Instruction execution failed	Check the parameter configuration of the host computer; Re-download the program	Warning
0x00A2(162)	0x0001(1)	Standard Modbus error, exception code 01, illegal function code	Check whether the configuration of function code accessed by master connected with PLC is legal	Warning
0x00A2(162)	0x0002(2)	Standard Modbus error, exception code 02, illegal register address	Check whether the address configuration accessed by the master connected with PLC is legal	Warning
0x00A2(162)	0x0003(3)	Standard Modbus error, exception code 03, data number error	Check whether the number configuration accessed by the master connected with PLC is legal	Warning
0x00A2(162)	0x0004(4)	Standard Modbus error, exception code 04, slave device failure	Check whether the master connected with PLC is configured correctly	Warning
0x00A2(162)	0x0005(5)	Communication timeout, the communication time exceeds the maximum communication time set by the user	Check whether the serial connection is normal	Warning
0x00A2(162)	0x0006(6)	Modbus TCP master-slave connection timeout	Check whether the connection of network cable is normal, and whether the ip and port number are set correctly	Warning

Hex main error code (corresponding decimal)	Hex error subcode (corresponding decimal)	Meaning of error	Solution	Error level
0x00A2(162)	0x0007(7)	The communication connection is disconnected	Check whether the line connection is normal	Warning
0x00A2(162)	0x0008(8)	The received data frame does not conform to the Modbus protocol	Check whether the baud rate, data bit and parity bit are configured correctly	Warning
0x00A2(162)	0x0009(9)	CRC/LRC check error	Check whether the baud rate, data bit and parity bit are configured correctly	Warning
0x00A2(162)	0x000A(10)	Element address overflow (the amount of data received or sent exceeds the storage space of the element)	Check the element address	Warning
0x00A2(162)	0x000B(11)	The length of data received does not conform to the protocol or the number of elements exceeds the maximum limit specified by the function code	Check the master connected with PLC	Warning
0x00A2(162)	0x000C(12)	The received slave address does not match the requested slave address	Check the slave connected with PLC	Warning
0x00A2(162)	0x000D(13)	The received function code does not match the requested function code	Check the slave connected with PLC	Warning
0x00A2(162)	0x000E(14)	Instruction execution failed	Check the parameter configuration of the host computer; Re-download the program	Warning
0x00A3(163)	0x0001(1)	Standard Modbus error, exception code 01, illegal function code	Check whether the configuration of function code accessed by master connected with PLC is legal	Warning
0x00A3(163)	0x0002(2)	Standard Modbus error, exception code 02, illegal register address	Check whether the address configuration accessed by the master connected with PLC is legal	Warning
0x00A3(163)	0x0003(3)	Standard Modbus error, exception code 03, data number error	Check whether the number configuration accessed by the master connected with PLC is legal	Warning
0x00A3(163)	0x0004(4)	Standard Modbus error, exception code 04, slave device failure	Check whether the master connected with PLC is configured correctly	Warning
0x00A3(163)	0x0005(5)	Communication timeout, the communication time exceeds the maximum communication time set by the user	Check whether the serial connection is normal	Warning
0x00A3(163)	0x0006(6)	Modbus TCP master-slave	Check whether the	Warning

Hex main error code (corresponding decimal)	Hex error subcode (corresponding decimal)	Meaning of error	Solution	Error level
		connection timeout	connection of network cable is normal, and whether the ip and port number are set correctly	
0x00A3(163)	0x0007(7)	The communication connection is disconnected	Check whether the line connection is normal	Warning
0x00A3(163)	0x0008(8)	The received data frame does not conform to the Modbus protocol	Check whether the baud rate, data bit and parity bit are configured correctly	Warning
0x00A3(163)	0x0009(9)	CRC/LRC check error	Check whether the baud rate, data bit and parity bit are configured correctly	Warning
0x00A3(163)	0x000A(10)	Element address overflow (the amount of data received or sent exceeds the storage space of the element)	Check the element address	Warning
0x00A3(163)	0x000B(11)	The length of data received does not conform to the protocol or the number of elements exceeds the maximum limit specified by the function code	Check the master connected with PLC	Warning
0x00A3(163)	0x000C(12)	The received slave address does not match the requested slave address	Check the slave connected with PLC	Warning
0x00A3(163)	0x000D(13)	The received function code does not match the requested function code	Check the slave connected with PLC	Warning
0x00A3(163)	0x000E(14)	Instruction execution failed	Check the parameter configuration of the host computer; Re-download the program	Warning
0x00B0(176)	0x0001(1)	Client connection failed	Check whether the server side is turned on; Check whether the IP address set by the client is the IP address of the server; Check whether the network cable connection is loose	General error
0x00B0(176)	0x0002(2)	Instruction parameter setting error	Check whether the data quantity setting value is less than or equal to 0	General error
0x00B0(176)	0x0003(3)	Instruction parameter element number setting error	The amount of data sent or received exceeds the capacity of the data transmitting or receiving area	General error
0x00B0(176)	0x0004(4)	Server listening failed	Server socket not created, recreate server socket	General error
0x00B0(176)	0x0005(5)	Transmit instruction	Check the network	General

Hex main error code (corresponding decimal)	Hex error subcode (corresponding decimal)	Meaning of error	Solution	Error level
		execution failed	connection	error
0x00B0(176)	0x0006(6)	Receive instruction execution failed	Check the network connection	General error
0x00B8(184)	0x0002(2)	Instruction parameter setting error	Check whether the data quantity setting value is less than or equal to 0	General error
0x00B8(184)	0x0003(3)	Instruction parameter element number setting error	The amount of data sent or received exceeds the capacity of the data transmitting or receiving area	General error
0x00B8(184)	0x0005(5)	Transmit instruction execution failed	Check the network connection	General error
0x00B8(184)	0x0006(6)	Receive instruction execution failed	Check the network connection	General error
0x00F0(240)	0x0001(1)	The system version is too low to start the IoT card	Update system firmware version	Serious error
0x00F0(240)	0x0002(2)	Serious error in starting the IoT module	Check whether the driver and hardware work properly	Serious error
0x00F0(240)	0x0003(3)	Abnormal signal strength	Check whether the driver and hardware work properly	Warning
0x00F0(240)	0x0004(4)	No port or port read-write error	Check whether the driver and hardware work properly	Warning
0x00F0(240)	0x0005(5)	Dial activation failed	Check whether the driver and hardware work properly	Warning
0x00F0(240)	0x0006(6)	No sim card inserted	Check whether the sim card is installed correctly	Warning
0x00F0(240)	0x0007(7)	The sim card has no data flow, apn error, etc.	Change another sim card	Warning

Note: COM1 and COM2 belongs to the RS-485 interface, while COM3 belongs to the RS-232 interface.

Your Trusted Industry Automation Solution Provider

Shenzhen INVT Electric Co., Ltd.
Address: INVT Guangming Technology Building, Songbai Road, Matian,
Guangming District, Shenzhen, China
INVT Power Electronics (Suzhou) Co., Ltd.
Address: No. 1 Kunlun Mountain Road, Science & Technology Town,
Gaoxin District, Suzhou, Jiangsu, China
Website: www.invt.com

INVT mobile website   INVT e-manual

66001-01290

202403 (V1.0)