HollySys Programmable Logic Controller

PLC Instruction Manual

Intelligence For Excellence

HollySys Group

Beijing HollySys Intelligent Technologies Co., Ltd.

Version: 01/2025

Copyright Notice

Address: Di Sheng Middle Road, No.2,
Economic-Technological Development Area, 100176, Beijing, China
Tel: +86 010-5898 1588
Consulting Hotline: 4008-111-999
Technology Request: +86 010-58981514
Fax: +86 010-5898 1558
Web: http://www.hollysys.com
Email: PLC@hollysys.com
Sina weibo: http://weibo.com/hollysysplc

Table of Contents

Chapter 1 About This Document

PLC system of Hollysys is divided into the three platforms, including MC, LK and LE. It provides the prolific instructions for three platforms via AutoThink. Library content and uploading instructions are different via choosing different target platform. Detail contents refer to 2.1 Definition and Classification of Instructions.

There is no English version manual for HMC motion instruction library, IO operation instructions library and instruction library associated with the controller.

1.1 What's New

Version	Description	Date
V1.0	New	December 30.2016
V1.1	Update LE platform instructions	April 18. 2017
V1.2	Add MKC platform instructions	July 15. 2017
V1.3	File operation instructions are modified as follows: 1.File path "/media/flash/" is modified to "flash/" 2.File path "/media/sd/" is modified to "sd/"	December 30. 2017
V1.4	Update Chapter 3.4.1	April 20. 2018
V1.5	Add description about the input variable domain in Chapter 3.1.1 Arithmetical Operation	July 30. 2018
V1.6	Add instructions GetLE5406ModuleSlaveDiagInfo(Get LE5406 Slave Module Diagnose Information), GetLK250ModuleSlaveDiagInfo(Get LK250 Slave Module Diagnose Information), ControlLK620(Control and Read LK620 Module Information)	March 10. 2019
V1.7	Add instructions MOVE_BLOCK_EXT, DataMemCopy,DataMemSet	December 16. 2019
V1.8	sysGetEthMac (Get MAC Address) adds input parameters "solt " Add sysGetCPUModuleVernExt (Get Extended Version of CPU)	April 16. 2020
V1.9	Add instructionsGetLK631SOEAlarm(Read LK631 FRAM Alarm Info Module Information),sysGetCPUSOEDiagInfo(Get RTS SOE Diagnose Information),sysClearSOE(Clear RTS SOE Information)	September 22. 2020
V2.0	Add instruction ControlHSC(Control and Read LK620 Module Information)	December 24. 2020
V1.0	Update Considerations for Use of Instruction System	March 15. 2021
V1.1	Add instruction about LK238	December 24, 2021
V1.2	Add instruction: LK238_ModbusMaster(LK238 Modbus Master Configure Read and Write) sysReadSOE(Read RTS SOE Information) Update following instructions: ModbusTcpSlaveConfig(Modbus TCP Slave Config) sysGetCPUDiagInfo (Get Diagnosis of CPU)	December. 2023
V1.3	Add instruction: COM_FRAME_BREAK_TIME(COM PORT user-define) sysGetHTCP_CycleTime(Get Hollitcp Master Cycle Time) Update following instructions: sysGetEthMac (Get MAC Address)	March.2024

Version	Description	Date
	LK238_ModbusMaster(LK238 Modbus Master Configure Read and Write) sysReadSOE(Read RTS SOE Information) TCP_SEND(Send data via tcp protocol) TCP_RECV(Recv data via tcp protocol) TCP_CONNECT(Establish a TCP communication connection) ETH_DISCONNECT(Close the Ethernet communication connection)	
V1.4	Add instruction: MKC3167_IS_ENABLE_CHANNEL MKC3167_NORMAL_READ MKC3167_GET_ERR_CHNL_ID MKC3167_READ_FAULT_DATA	January. 2025

1.2 Purpose

This manual will help users to grasp the algorithm instructions and the using of instructions in various language environments.

1.3 Intended audience

The manual is applicable to following personnel:

- Project personnel responsible for implementing system project.
- Technologists responsible for system maintenance.

1.4 Important information

The following marks are used in this manual for identifying appropriate information:

Danger: This symbol indicates a dangerous situation that could cause death or injury if not avoided.

Electric shock: This symbol indicates a dangerous situation of electric shock if not avoided.

Warning: This symbol indicates a dangerous situation that could cause property or environment damage if not avoided.

Notes: This symbol indicates useful tips or recommendations for efficient operation.

SEE ALSO This symbol indicates information for reference.

1.5 Related documentation

HollySys Programmable Logic Control System Instruction Manual

AutoThink V3.1 User Manual_Project Configuration

LK220 Series Programmable Logic Controller System Manual

Chapter 2 Overview of Instruction System

The large-scale PLC system of HollySys provides users with abundant instructions, which can be called by the programming software AutoThink .Easy and convenient to use.

This manual will describe instructions of the MC, LK and LE in detail.

2.1 Definition and Classification of Instructions

The command and the combination of multiple commands that enables CPU to complete some operation or perform some function is referred to as the instruction in PLC, and the instruction set is referred to as the instruction system. The instruction system is a bridge for the hardware and software of PLC and the basis of the PLC program design.

AutoThink provides abundant instructions, which are divided into the conversion instruction, comparison instruction, type conversion instruction, logic operation instruction and analog application according to different functions. In order to facilitate the understanding and memory, we divide such types into six categories, including standard instruction, basic application instruction, system instruction, industry application library, field bus library (only MC1002E and MC1002R support), product extension library. product extension instruction. Three platforms all include the standard instruction, basic application instruction, system instruction, corresponding library files in industry application library and product extension library are upload according to the different target platform.

There are two implementation modes of the PLC instructions in the programming software, namely, the function and the function block. Both the function and the function block are the program organization unit of AutoThink and the pre-compiled program to perform some function. The function block can output one or more results, and each function block instance is provided with one relevant identifier (namely, the instance name). The function does not require an identifier and provides one output result only (namely, the return value of function). For the specific concept of the function and the function block, refer to POU types . The following Table indicates the function implementation method. Of which, FUN indicates the instruction is implemented by the function, and FB indicates the instruction is implemented by the function block.

- The instructions implemented by the function need not to declare anything during the use.
- The instructions implemented by the function block shall declare the instance name during the use.

Use range of instructions library in MC, LK and LE library manger are different, the follows instructions are supported by different platform, as shown in Table.

Instruction List

Instruction Type		Instruction	Instruction Description	MC	LK	LE	Note
Standard Instruction	Arithmetic Operation	ADD(FUN)	Add	√	√	√	
		SUB(FUN)	Subtract	√	√	√	
		MUL(FUN)	Multiply	√	√	√	
		DIV(FUN)	Divide	√	√	√	
		MOD(FUN)	Modulus	√	√	√	
		SIZEOF(FUN)	Variable Size	√	√	√	
		ABS(FUN)	Absolute Value	√	√	√	
		SQRT(FUN)	Square Root	√	√	√	
		LOG(FUN)	Common Logarithm	√	√	√	
		LN(FUN)	Natural Logarithm	√	√	√	
		EXP(FUN)	Exponent	√	√	√	
		SIN(FUN)	Sine	√	√	√	
		COS(FUN)	Cosine	√	√	√	
		TAN(FUN)	Tangent	√	√	√	
		ASIN(FUN)	Arc Sine	√	√	√	
		ACOS(FUN)	Arc Cosine	√	√	√	
		ATAN(FUN)	Arc Tangent	√	√	√	
		EXPT(FUN)	Power	√	√	√	
		MOVE(FUN)	Assignment	√	√	√	
	Logical Operation	AND(FUN)	And	√	√	√	
		OR(FUN)	Or	√	√	√	
		XOR(FUN)	Exclusive or	√	√	√	
		NOT(FUN)	Not	√	√	√	
	Comparison Operation	GT(FUN)	Greater Than	√	√	√	
		LT(FUN)	Less Than	√	√	√	
		GE(FUN)	Greater or Equal	√	√	√	
		LE(FUN)	Less or Equal	√	√	√	
		EQ(FUN)	Equal	√	√	√	
		NE(FUN)	Not Equal	√	√	√	
	Select Operation	SEL(FUN)	One Out of Two	√	√	√	
		MAX(FUN)	Maximum	√	√	√	
		MIN(FUN)	Minimum	√	√	√	
		MUX(FUN)	One Out of Multiply	√	√	√	
		LIMIT(FUN)	Limit	√	√	√	
	Shift Operation	SHL(FUN)	Shift Left	√	√	√	
		SHR(FUN)	Shift Right	√	√	√	
		ROL(FUN)	Ring Shift Left	√	√	√	
		ROR(FUN)	Ring Shift Right	√	√	√	
	Data Type Conversion	BOOL_TO_<TYPE>	BOOL Type Conversion	√	√	√	
		BYTE_TO_<TYPE>	BYTE Type Conversion	√	√	√	
		DATE_TO_<TYPE>	DATE Type Conversion	√	√	√	
		DINT_TO_<TYPE>	DINT Type Conversion	√	√	√	
		DT_TO_<TYPE>	DT Type Conversion	√	√	√	
		DWORD_TO_<TYPE>	DWORD Type Conversion	√	√	√	

Instruction Type		Instruction	Instruction Description	MC	LK	LE	Note
		INT_TO_<TYPE>	INT Type Conversion	√	√	√	
		LINT_TO_<TYPE>	LINT Type Conversion	√			
		LREAL_TO_<TYPE>	LREAL Type Conversion	√	√	√	
		LWORD_TO_<TYPE>	LWORD Type Conversion	√			
		REAL_TO_<TYPE>	REAL Type Conversion	√	√	√	
		SINT_TO_<TYPE>	SINT Type Conversion	√	√	√	
		STRING_TO_<TYPE>	STRING Type Conversion	√	√	√	
		TIME_TO_<TYPE>	TIME Type Conversion	√	√	√	
		TOD_TO_<TYPE>	TOD Type Conversion	√	√	√	
		UDINT_TO_<TYPE>	UDINT Type Conversion	√	√	√	
		UINT_TO_<TYPE>	UINT Type Conversion	√	√	√	
		ULINT_TO_<TYPE>	ULINT Type Conversion	√			
		USINT_TO_<TYPE>	USINT Type Conversion	√	√	√	
		WORD_TO_<TYPE>	WORD Type Conversion	√	√	√	
		TRUNC	TRUNC	√	√	√	
	Pointer Operation	ADR(FUN)	Get Address	√	√	√	
		VAL(FUN)	Get Value	√	√	√	
	Character String Operation	Left(FUN)	Get string from left	√	√		
		Right(FUN)	Get string from right	√	√		
		Concat(FUN)	Connect two strings	√	√		
		Delete(FUN)	Delete string	√	√		
		Insert(FUN)	Insert string	√	√		
		Mid(FUN)	Get string from middle	√	√		
		Replace(FUN)	Replace string	√	√		
		Find(FUN)	Find string	√	√		
		Len(FUN)	String length	√	√		
	Trigger Functions	F_TRIG(FB)	Falling Edge Detection Trigger	√	√	√	
		R_TRIG(FB)	Rising Edge Detection Trigger	√	√	√	
	Bistable Functions	RS(FB)	Reset Priority Bistable	√	√	√	
		SR(FB)	Set Priority Bistable	√	√	√	
	Counter Functions	CTD(FB)	Decrement Counter	√	√	√	
		CTU(FB)	Increment Counter	√	√	√	
		CTUD(FB)	Increment/Decrement Counter	√	√	√	
	Timer Functions	TP(FB)	Normal Timer	√	√	√	
		TON(FB)	On Delay Timer	√	√	√	
		TOF(FB)	Off Delay Timer	√	√	√	
		TONR(FB)	Retentive On Delay Timer	√	√	√	
	IEC_SFC Functions	SFCActionControlBlock(FB)	Control instruction of associated action in SFC	√	√	√	
Basic Application Instruction	BCD Conversion Functions	BCD_TO_INT(FUN)	BCD Code to Integer	√	√	√	
		INT_TO_BCD(FUN)	Integer to BCD Code	√	√	√	
	Bit_Byte Functions	UNPACK(FB)	Bit Split	√	√	√	
		EXTRACT(FUN)	Bit Extraction	√	√	√	
		PACK(FUN)	Bit Merger	√	√	√	
		PUTBIT(FUN)	Bit Assignment	√	√	√	

Instruction Type		Instruction	Instruction Description	MC	LK	LE	Note
	Mathematical Functions	DERIVATIVE(FB)	Derivative Function	√	√	√	
		INTEGRAL(FB)	Integral Function	√	√	√	
		STATISTIC_INT(FB)	Int Statistics	√	√	√	
		STATISTIC_REAL(FB)	Real Statistics	√	√	√	
		VARIANCE(FB)	Calculation of Variance	√	√	√	
	Controller Functions	PID(FB)	The PID Controller	√	√	√	
	Signal Generator Functions	BLINK(FB)	Pulse Signal Generator	√	√	√	
		GEN(FB)	Typical Signal Generator	√	√	√	
		RAND(FB)	Random Number Generator	√	√	√	
	Analog Signal Processing Functions	RAMP_INT(FB)	Int Rate Limit	√	√	√	
		RAMP_REAL(FB)	Real Rate Limit	√	√	√	
		CHARCURVE(FB)	Characteristic Curve	√	√	√	
		HYSTERESIS(FB)	Lag	√	√	√	
		LIMITALARM(FB)	Limit Alarm	√	√	√	
		HEX_ENGIN(FB)	Hexadecimal Conversion to Engineering Data	√	√	√	
		ENGIN_HEX(FB)	Engineering Data Conversion to Hexadecimal(WORD)	√	√	√	
		ENGIN_HEX2(FB)	Engineering Data Conversion to Hexadecimal(INT)	√	√	√	
	Data Block Move Functions	MOVE_BLOCK(FB)	Data Block Move	√	√	√	
		MOVE_BLOCK_EXT(FB)	Data Block Move Extension Instruction	√	√	√	
	Set Address Instruction	VALSET_BYTE(FUN)	Set contents of address			√	
		VALSET_WORD(FUN)				√	
		VALSET_DWORD(FUN)				√	
		VALSET_REAL(FUN)				√	
System Instruction	Communication Instruction	GENERATE_CRC(FB)	Cyclic Redundancy Check	√	√	√	
		ComConfig(FB)	Set Serial Port Communication Parameter	√	√		
		ComRecv(FB)	Receive Serial Port Data	√	√		
		ComSend(FB)	Send Serial Port Data	√	√		
		CANRecv(FB)	Receive CAN Data	√	√		
		CANSend(FB)	Send CAN Data	√	√		
		CANConfig(FB)	Set CAN Parameter	√	√		
		ComClearRecvFIFO(FB)	Clear Serial Port Data	√	√		
		COMM_SEND(FB)	Send Communication Data			√	
		COMM_RECEIVE(FB)	Receive Communication Data			√	
		MODBUS_MASTER(FB)	Modbus RTU Master Communication			√	
		ModbusTcpMasterConnect(FB)	Modbus TCP Master Connect			√	
		ModbusTcpMasterMsg(FB)	Modbus TCP Master Msg			√	
		ModbusTcpSlaveMsg(FB)	Modbus TCP Slave Msg			√	
		ModbusTcpSlaveConfig(FB)	Modbus TCP Slave Config			√	
	Task Operation Instruction	TaskStart(FUN)	Task Start	√	√		
		TaskStop(FUN)	Task Stop	√	√		
		TaskSuspend(FUN)	Task Suspension	√	√		
		TaskStatus(FUN)	Return Task Status	√	√		

Instruction Type		Instruction	Instruction Description	MC	LK	LE	Note
	File Operation Instruction	FileOpen(FUN)	Open File	√			
		FileClose(FUN)	Close File	√			
		FileRead(FUN)	Read File	√			
		FileWrite(FUN)	Write File	√			
		FileLseek(FUN)	Move File to Read/Write Pointer	√			
		FileListScan(FUN)	Scan Directory File	√			
		FindFirstFile(FUN)	Seach First File	√			
		FindNextFile(FUN)	Seach Next File	√			
		FindClose(FUN)	Seach End	√			
		FileRemove(FUN)	Delete File	√			
		FlashWrite(FUN)	Write Data to Flash	√			
		FlashRead(FUN)	Read Data From Flash	√			
	Time Correlation Instruction	GetTickCnt(FUN)	Get Tick Count	√	√		
		TimeDelay(FUN)	Time Delay	√	√		
		SET_RTC(FB)	Set Real Time Clock	√	√	√	
		GET_RTC(FB)	Read Real Time Clock	√	√	√	
	Interruption set instruction	SET_INTERRUPT_DI（FB）	Interruption Timer Setting			√	
		HD_TIMER0（FB）	Interruption Timer Blocks0			√	
		HD_TIMER1（FB）	Interruption Timer Blocks1			√	
		HD_TIMER2（FB）	Interruption Timer Blocks2			√	
		HD_TIMER3（FB）	Interruption Timer Blocks3			√	
		HD_TIMER4（FB）	Interruption Timer Blocks4			√	
		HD_TIMER5（FB）	Interruption Timer Blocks5			√	
	Module Information Instruction	sysGetModel(FB)	Get Module Type	√	√		
		sysGetCPUModuleVern(FB)	Get Version of CPU	√	√		
		sysGetComModuleVern(FB)	Get Version of Communication Module	√	√		
		sysGetModuleSN(FB)	Get Serial Number of Module	√	√		
		sysGetCheckTime(FB)	Get Check Time	√	√		
		sysGetEthMac(FB)	Get MAC Address	√	√		
	System State Instruction	sysGetRunTime(FB)	Get Task Running time	√	√		
		sysGetKeyState(FB)	Get Switch State of Key	√	√		
		sysGetDPMasterState(FB)	Get Current State of DP Master	√	√		
		sysGetCPULoad(FB)	Get Current Load of CPU	√	√		
	Redundancy Instruction	sysGetRedState(FB)	Get Redundant State of System	√	√		
		sysMasterSwitchToSlave(FB)	Master Switch To Slave	√	√		
		sysGetDPSlaveState(FB)	Get Diagnosis of DP Slave	√	√		
		sysGetCPUDiagInfo(FB)	Get Diagnosis of CPU	√	√		
		sysGetComModuleDiagInfo(FB)	Get Diagnosis of Communication Module	√	√		
Industry Application Library	HMC motion instruction library			√			No English version manual
	Motion control instruction	MF_M(FB)	Frequency Measurement,5128			√	
		MF_H(FB)	Frequency Measurement,5106/5107/5108/5109/5118			√	
		HSC_M(FB)	High Speed Counting,5128			√	

Instruction Type		Instruction	Instruction Description	MC	LK	LE	Note
		HSC_H(FB)	High Speed Counting,5106/5107/5108/5109/5118			√	
		HSC_L(FB)	High Speed Counting,5104/5105/5107E/5107L/5109L			√	
		PTO_H(FB)	PTO Output,5106/5108/5118			√	
		PTO_L(FB)	PTO Output,5104			√	
		MOVERELATIVE_M(FB)	Relative Move,5128			√	
		MOVERELATIVE_H(FB)	Relative Move,5106/5108/5118			√	
		MOVEABSOLUTE_M(FB)	Absolute Move,5128			√	
		MOVEABSOLUTE_H(FB)	Absolute Move, 5106/5108/5118			√	
		JOG_M(FB)	Jog,5128			√	
		JOG_H(FB)	Jog,5106/5108/5118			√	
		STOP_M(FB)	Stop,5128			√	
		STOP_H(FB)	Stop,5106/5108/5118			√	
		HOME_M(FB)	Regression of Original Point, 5128,5128			√	
		HOME_H(FB)	Regression of Original Point, 5128,5106/5108/5118			√	
		FORWARD(FB)	Repeat Pulse Output,5128			√	
		PWM(FB)	Pulse-Width Modulation,5104/5106/5108/5128/5118			√	
		DEFPOS(FB)	Define,5128			√	
		CANCEL(FB)	Cancel,5128			√	
		FASTPID(FB)	Fast PID,5128			√	
		LATCH(FB)	Latch Data,5128			√	
		MOVELINK(FB)	MoveLink,5128			√	
		CONNECT(FB)	Electronic Gear,5128			√	
Field Bus Library		Instructions associated with ETHERCAT protocol		√			No English version manual
		Instructions associated with RTEX protocol		√			No English version manual
Product Extension Instruction	DI Reversed	SetInvertDI(FUN)	Set DI Invert	√			No English version manual
		GetInvertDI(FUN)	Get DI Invert	√			
	Product Internal State Instruction	sysGetBatteryAlarm(FB)	Get Battery Alarm	√	√		
		sysGetTEMP(FB)	Get Temperature of CPU	√	√		
	Additional Module Information	GetLE5406ModuleSlaveDiagInfo(FB)	Get LE5406 Slave Module Diagnose Information		√		
		GetLK250ModuleSlaveDiagInfo(FB)	Get LK250 Slave Module Diagnose Information		√		
		GetLK250ModuleSlaveDiagInfoExt(FB)	Get LK250 Slave Module Diagnose Information		√		
		ControlLK620(FB)	Control and Read LK620 Module Information		√		
		GetLK631SOEAlarm(FB)	Read LK631 FRAM Alarm Info Module Information		√		

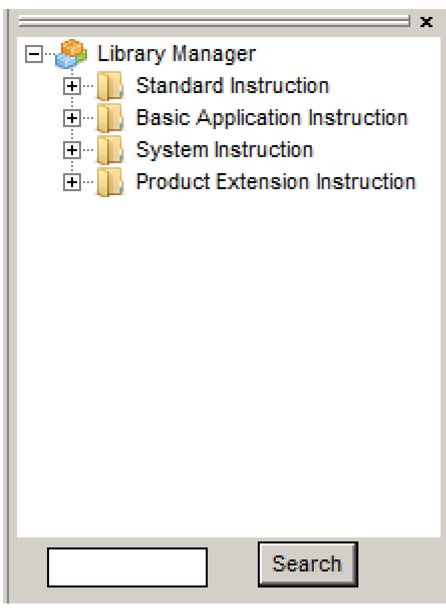
Instruction Type		Instruction	Instruction Description	MC	LK	LE	Note
		COMM_SET_CONFIG（FB）	Set LK238 Communication Instruction Param		√		
		COMM_GET_CONFIG（FB）	Read LK238 Communication Instruction Param		√		
		COMM_RW_MODBUS_MASTER（FB）	LK238 Modbus Master Read and Write Communication Data		√		
		COMM_RW_MODBUS_SLAVE（FB）	LK238 Modbus Slave Read and Write Communication Data		√		
		COMM_RW_FREEPORT（FB）	LK238 Free Protocol Read and Write Communication Data		√		
		COMM_CTL_MODULE（FB）	LK238 Module Control Function Block		√		
	SOE Instruction	sysGetCPUSOEDiagInfo(FB)	Get RTS SOE Diagnose Information		√		
		sysClearSOE(FB)	Clear RTS SOE Information		√		

2.2 Classification of Instruction Libraries

The instruction library is divided into four types according to the different functions of the instruction.

- Standard Instruction: Instructions stipulated in IEC61131-3 and general instructions in each editor.
- Basic Application Instruction: Universal instruction.
- System Instruction: Universal system interface instructions provided by OS and universal PLC system interface instructions provided by RTS.
- Product Extension Instruction: Commission product interface provided by specific products.

The Library Manager as shown in Figure.

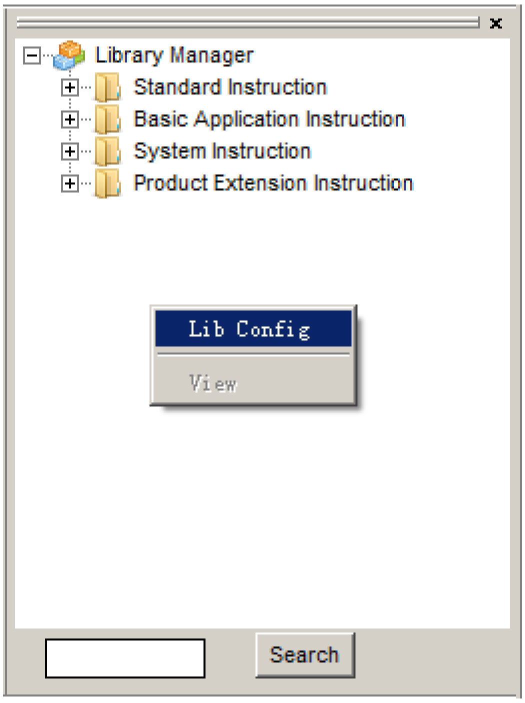


Library Manager

2.3 Addition of Instruction Library

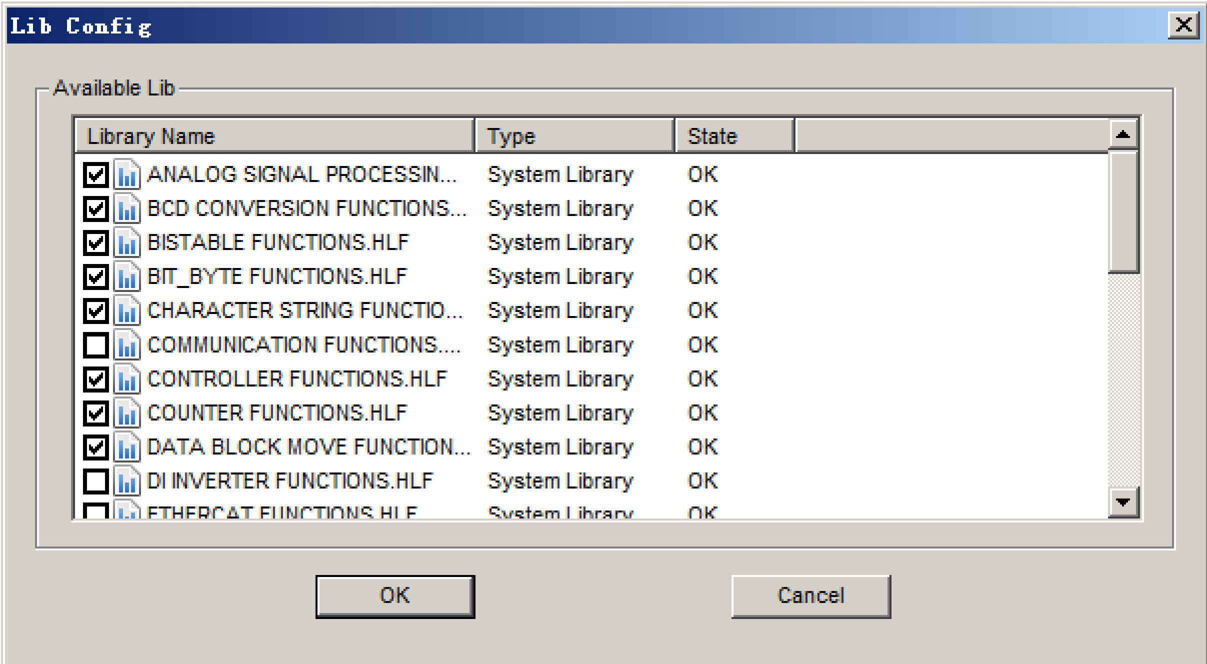
All instructions in library manager are enabled by default, which can be configured manually according to requirement.

- Library manager: Right click the blank area, and click [Lib Config].



Library Configuration Menu Item

The system will pop up the **Lib Config** dialog box as shown in figure. Check the library file to be added, and click the **OK** button, the library will be added to the Library Manager window.



Library Configuration Dialog Box

2.4 Considerations for Use of Instruction System

- The rising edge enable means that relevant function of the instruction will be performed when the enable terminal changes into the high level from the low level and always keep the high level. Furthermore, if the input parameter of the instruction changes, it will take effect only when the enable terminal changes into the high level from the low level again.
- The high level enable means that relevant function of the instruction will be performed when the enable terminal always keep the high level. Furthermore, it will take effect real time if the input parameter of the instruction changes.
- The high bit will lose if the range of the data output definition is less than the operation result when the mathematical operation instruction is used.
- If the instance name of the instruction is declared in the RETAIN area (power-down hold), all input/output variables of this instruction will occupy the RETAIN memory area. Hence, it is not suggested to declare too many instances in the RETAIN memory area, to avoid the lack of RETAIN area space.
- In LK platform, the former 4,000 bytes in the M area (namely, %MB0~%MB3999) can perform the power-down hold function. In LE platform,%MB1000~%MB1999 in the M area can perform the power-down hold function.
- The insertion of enable operators and the insertion of function blocks are two different instruction call modes in the ladder diagram (LD) programming environment. Their difference is that if the instruction

is called by the insertion of enable operators, corresponding instruction code will be scanned when the enable terminal is of the low level, and if the instruction is called by the insertion of function blocks, corresponding instruction code will take the enable terminal as one input value for the scanning regardless of the low level or the high level for the enable terminal.

- For arithmetical operations, when the inputs exceed the defined range, the output will be invalid value:
    - 1.#INF: Represents an infinity number, including the positive infinity 1.#INF and the negative infinity -1.#INF.
    - 1.#IND: Represents infinitesimal number.
    - 1.#QNAN: It is not a number.
- LE5128 electronic gear does not support reverse synchronization.

Chapter 3 Instruction

3.1 Standard Instruction

3.1.1 Arithmetical Operation

3.1.1.1 ADD(Add)

3.1.1.1.1 Function

Two (or more) variables or constants are added. The result will show overflow value when exceeds the range of the output variable. For the REAL variables, the output shows an invalid value "1.#INF" or "-1.#INF" when the result has overflowed. Please see Considerations for Use of Instruction System for the invalid value.

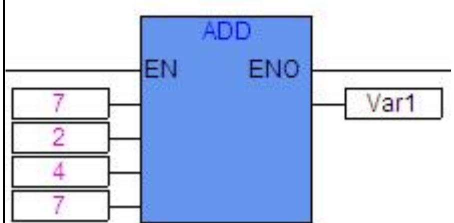
3.1.1.1.2 Input/output Data Type

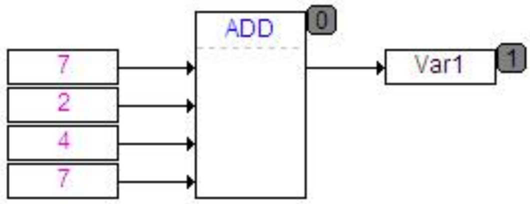
BYTE, WORD, DWORD, SINT, USINT, INT, UINT, DINT, UDINT, REAL, TIME.

3.1.1.1.3 Application Example of Instructions

Variable Definition

No.	Variable Name	Address	Variable Description	Variable Type	Initial Value	Power Fail Safeguard
0001	Var1			INT	0	FALSE

Programing language	Program
Ladder diagram (LD)	 (*The result Var1 is 20*)
Structured text (ST)	Var1:=7+2+4+7; (*The result Var1 is 20*)

Variable Definition	
Continuous Function Chart (CFC)	

ⓘ

- The TIME variable can also use the ADD function, and two TIME variables are added to obtain a new time, such as t#45s+t#50s =t#1m35s.
- The selected output data type shall be able to store the output result. Otherwise, it may cause the data error.The instruction of MUL, SUB, DIV is same with it.

3.1.1.2 SUB(Subtract)

3.1.1.2.1 Function

Two variables or constants are subtracted.

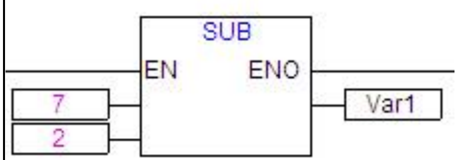
3.1.1.2.2 Input/output Data Type

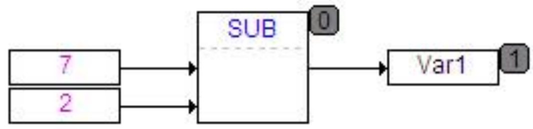
BYTE, WORD, DWORD, SINT, USINT, INT, UINT, DINT, UDINT, REAL, TIME.

3.1.1.2.3 Application Example of Instructions

Variable Definition

No.	Variable Name	Address	Variable Description	Variable Type	Initial Value	Power Fail Safeguard
0001	Var1			INT	0	FALSE

Programing language	Program
Ladder diagram (LD)	 (*The result Var1 is 5*)
Structured text (ST)	Var1:=7-2; (*The result Var1 is 5*)

Variable Definition	
Continuous Function Chart (CFC)	 (*The result Var1 is 5*)

ⓘThe TIME variable can also use the SUB function, and two TIME variables are subtracted to obtain one new time, such as t#1m35s - t#50s = t#45s. However, the time result can't be negative.

3.1.1.3 MUL(Multiply)

3.1.1.3.1 Function

Two (or more) variables or constants are multiplied.

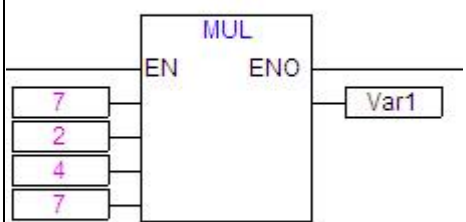
3.1.1.3.2 Input/output Data Type

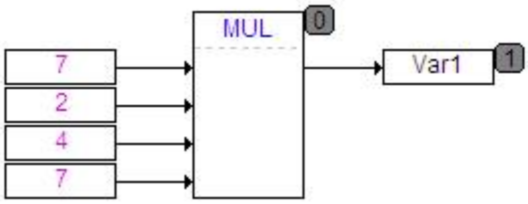
BYTE, WORD, DWORD, SINT, USINT, INT, UINT, DINT, UDINT, REAL and TIME.

3.1.1.3.3 Application Example of Instructions

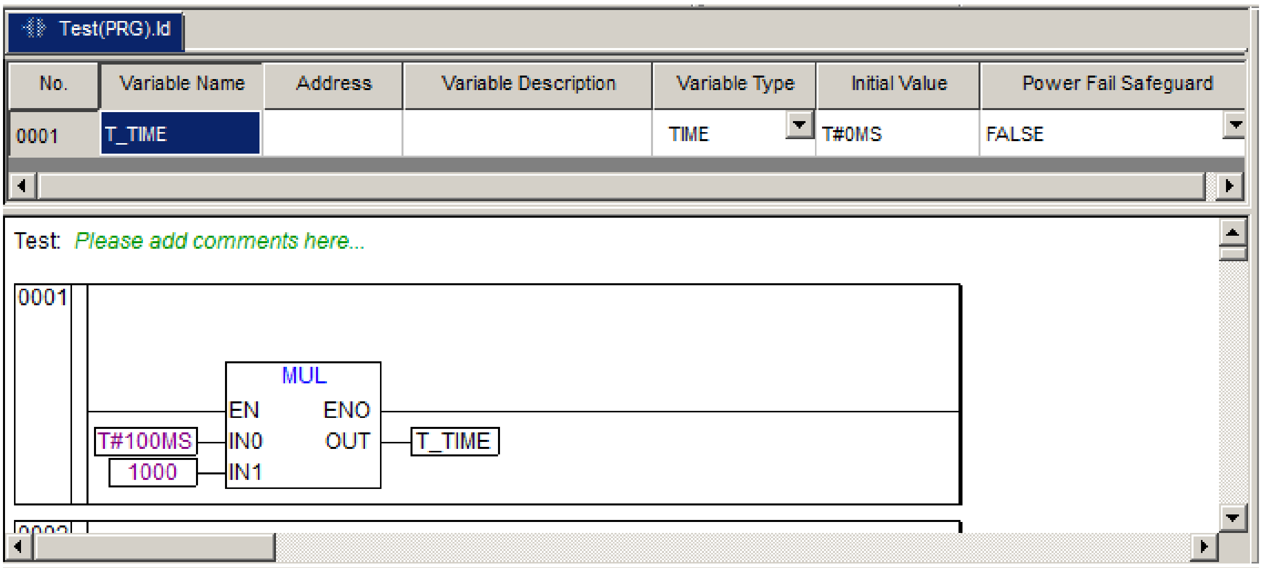
Variable Definition

No.	Variable Name	Address	Variable Description	Variable Type	Initial Value	Power Fail Safeguard
0001	Var1			INT	0	FALSE

Programing language	Program
Ladder diagram (LD)	 (*The result Var1 is 392*)
Structured text (ST)	Var1:=7*2*4*7; (*The result Var1 is 392*)

Variable Definition	
Continuous Function Chart (CFC)	 (*The result Var1 is 392*)

The TIME variable or constant can be multiplied by an integer variable or constant, and its result is the TIME variable. For example, the time constant that supports T#100MS is multiplied by the constant 1000 to get the time in AutoThink as shown in figure.



TIME Multiplication Operation

For the TIME multiplication, the compilation passes when the first operand is the TIME data only, and other operands are the integer data. The system will report the error for other cases, such as T_TIME:=1000*T#100ms.

- LD, CFC language error

  Type mismatching: The TIME type can't be converted into the WORD type.

- ST language error

  Multiplication operation: The type mismatches.

3.1.1.4 DIV(Divide)

3.1.1.4.1 Function

The variable or constant is divided. When the divisor is 0, the result is equal to the dividend.

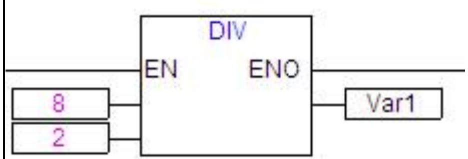
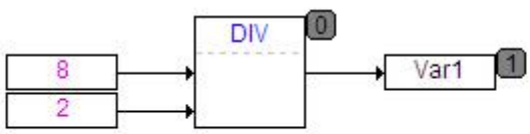
3.1.1.4.2 Input/output Data Type

BYTE, WORD, DWORD, SINT, USINT, INT, UINT, DINT, UDINT, REAL.

3.1.1.4.3 Application Example of Instructions

Variable Definition

No.	Variable Name	Address	Variable Description	Variable Type	Initial Value	Power Fail Safeguard
0001	Var1			INT	0	FALSE

Programing language	Program
Ladder diagram (LD)	 (*The result Var1 is 4*)
Structured text (ST)	Var1:=8/2; (*The result Var1 is 4*)
Continuous Function Chart (CFC)	 (*The result Var1 is 4*)

ⓘWhen the DIV instruction is used in the project, the instruction CheckDivByte, CheckDivWord, CheckDivDWord and CheckDivReal can be used to check whether the divisor is zero, to avoid the phenomenon of divisor zero.

3.1.1.5 MOD(Modulus)

3.1.1.5.1 Function

The variable or constant is divided to take the remainder, which is an integer. When the divisor is 0, the result is equal to the dividend, and the sign of result is consistent with the dividend.

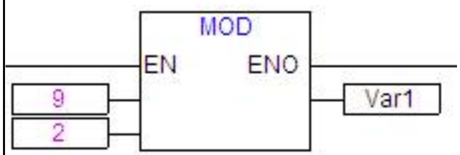
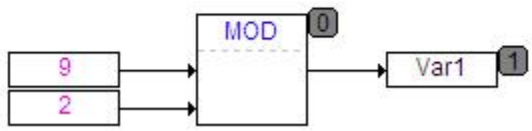
3.1.1.5.2 Input/output Data Type

BYTE, WORD, DWORD, SINT, USINT, INT, UINT, DINT, UDINT.

3.1.1.5.3 Application Example of Instructions

Variable Definition

No.	Variable Name	Address	Variable Description	Variable Type	Initial Value	Power Fail Safeguard
0001	Var1			INT	0	FALSE

Programing language	Program
Ladder diagram (LD)	 (*The result Var1 is 1*)
Structured text (ST)	Var1:=9 MOD 2; (*The result Var1 is 1*)
Continuous Function Chart (CFC)	 (*The result Var1 is 1*)

3.1.1.6 SIZEOF (Variable Size)

3.1.1.6.1 Function

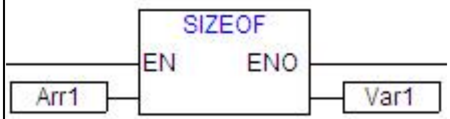
It will get the bytes required for the data type.

3.1.1.6.2 Application Example of Instructions

Variable Definition

No.	Variable Name	Address	Variable Description	Variable Type	Initial Value	Power Fail Safe
0001	Arr1			ARRAY[0..4] OF INT		FALSE
0002	Var1			INT	0	FALSE

Programing language	Program

Variable Definition	
Ladder diagram (LD)	 (*The result Var1 is 10*)
Structured text (ST)	Var1:=SIZEOF(arr1); (*The result Var1 is 10*)
Continuous Function Chart (CFC)	 (*The result Var1 is 10*)

3.1.1.7 ABS (Absolute Value)

3.1.1.7.1 Function

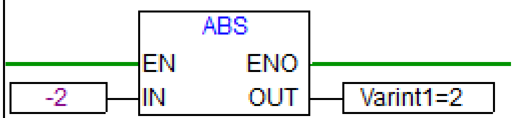
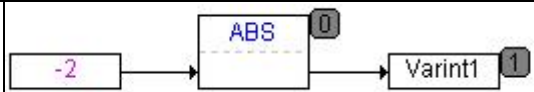
Assign the absolute value of the input data to the output variable.

3.1.1.7.2 Input/output Data Type

Input Data Type	Output Data Type
INT	INT, WORD, DWORD, DINT, UINT, REAL, LREAL
REAL	REAL
LREAL	LREAL
BYTE	INT, BYTE, WORD, DWORD, DINT, UINT, REAL, LREAL
WORD	WORD, DWORD, DINT, REAL, LREAL
DWORD	DWORD, DINT, REAL, LREAL
SINT	INT, BYTE, WORD, DWORD, DINT, UINT, REAL, LREAL
USINT	INT, BYTE, WORD, DWORD, DINT, UINT, REAL, LREAL
UINT	WORD, DWORD, DINT, UINT, REAL, LREAL
DINT	DWORD, DINT, REAL, LREAL
UDINT	DWORD, DINT, UDINT, REAL, LREAL

3.1.1.7.3 Application Example of Instructions

Variable Definition						
No.	Variable Name	Address	Variable Description	Variable Type	Initial Value	Power Fail Safeguard
0001	Varint1			INT	0	FALSE
Programing language	Program					

Variable Definition	
Ladder diagram (LD)	 (*The result Varint1 is 2*)
Structured text (ST)	Varint1:=ABS(-2); (*The result Varint1 is 2*)
Continuous Function Chart (CFC)	 (*The result Varint1 is 2*)

3.1.1.8 SQRT (Square Root)

3.1.1.8.1 Function

Get the square root value of the input data, and the input must be non- negative number.

LK platform, the output is infinitesimal number when the input is a negative number.

LE platform, the output is an invalid value when the input is a negative number. Please see Considerations for Use of Instruction System for the invalid value.

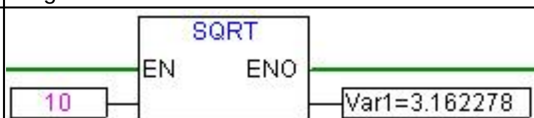
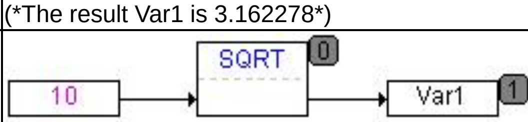
3.1.1.8.2 Input/output Data Type

BYTE, WORD, DWORD, INT, DINT, REAL, LREAL, SINT, USINT, UINT, UDINT, and the output data shall be of the REAL, LREAL types.

3.1.1.8.3 Application Example of Instructions

Variable Definition

No.	Variable Name	Address	Variable Description	Variable Type	Initial Value	Power Fail Safeguard
0001	Var1			REAL	0	FALSE

Programing language	Program
Ladder diagram (LD)	
Structured text (ST)	Var1:=SQRT(10); (*The result Var1 is 3.162278*)
Continuous Function Chart (CFC)	

Variable Definition	
	(*The result Var1 is 3.162278*)

3.1.1.9 LOG (Common Logarithm)

3.1.1.9.1 Function

Get the denary logarithm of the input data, and the input data shall be positive.

LK platform, the output is infinitesimal number when the input is less than or equal to 0.

LE platform, the output is an invalid value when the input is less than or equal to 0. Please see Considerations for Use of Instruction System for the invalid value.

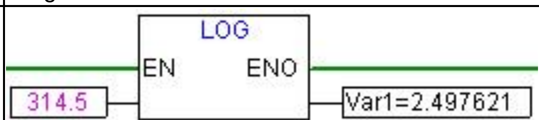
3.1.1.9.2 Input/output Data Type

It may be BYTE, WORD, DWORD, INT, DINT, REAL, LREAL, SINT, USINT, UINT, UDINT, and the output data shall be of the REAL, LREAL types.

3.1.1.9.3 Application Example of Instructions

Variable Definition

No.	Variable Name	Address	Variable Description	Variable Type	Initial Value	Power Fail Safeguard
0001	Var1			REAL	0	FALSE

Programing language	Program
Ladder diagram (LD)	
Structured text (ST)	Var1:=LOG(314.5); (*The result Var1 is 2.497621 *)
Continuous Function Chart (CFC)	 (*The result Var1 is 2.497621 *)

3.1.1.10 LN (Natural Logarithm)

3.1.1.10.1 Function

Get the natural logarithm of the input data, and the input data shall be positive.

LK platform, the output is infinitesimal number when the input is less than or equal to 0.

LE platform, the output is an invalid value when the input is less than or equal to 0. Please see Considerations for Use of Instruction System for the invalid value.

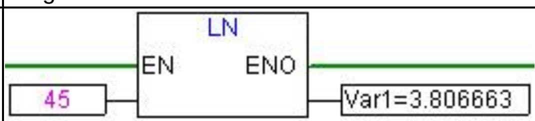
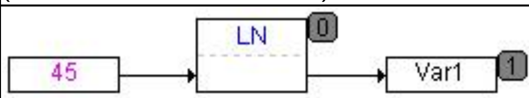
3.1.1.10.2 Input/output Data Type

It may be BYTE, WORD, DWORD, INT, DINT, REAL, LREAL , SINT, USINT, UINT, UDINT, and the output data shall be of the REAL, LREAL types.

3.1.1.10.3 Application Example of Instructions

Variable Definition

No.	Variable Name	Address	Variable Description	Variable Type	Initial Value	Power Fail Safeguard
0001	Var1			REAL	0	FALSE

Programing language	Program
Ladder diagram (LD)	
Structured text (ST)	Var1:=LN(45); (*The result Var1 is 3.806663*)
Continuous Function Chart (CFC)	 (*The result Var1 is 3.806663*)

3.1.1.11 EXP (Exponent)

3.1.1.11.1 Function

Carry out the power calculation by taking the input data as the exponent. Namely, x is the input, and y is the output.

LK platform, the output is an infinity number “1.#INF“ when the input is greater than 709.8. Output is 0 when the input is less than -745.2.

LE platform, the output is an invalid value when the input is greater than 88.7228. Please see

Considerations for Use of Instruction System for the invalid value.

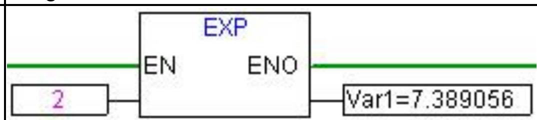
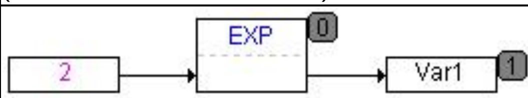
3.1.1.11.2 Input/output Data Type

It may be BYTE, WORD, DWORD, INT, DINT, REAL, LREAL,SINT, USINT, UINT, UDINT, and the output data shall be of the REAL, LREAL types.

3.1.1.11.3 Application Example of Instructions

Variable Definition

No.	Variable Name	Address	Variable Description	Variable Type	Initial Value	Power Fail Safeguard
0001	Var1			REAL	0	FALSE

Programing language	Program
Ladder diagram (LD)	
Structured text (ST)	Var1:=EXP(2); (*The result Var1 is 7.389056*)
Continuous Function Chart (CFC)	 (*The result Var1 is 7.389056*)

3.1.1.12 SIN (Sine)

3.1.1.12.1 Function

Get the sine value of the input data, and the input data is represented by the radian.

$$\text{Radian (rad)} = \text{Angle} \times \frac{\pi}{180°}$$

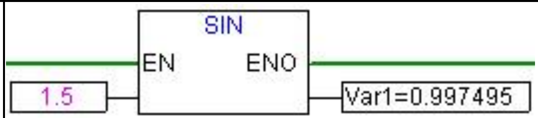
3.1.1.12.2 Input/output Data Type

The input data type may be BYTE, WORD, DWORD, INT, DINT, REAL, LREAL, SINT, USINT, UINT, UDINT, and the output data shall be of the REAL, LREAL types.

3.1.1.12.3 Application Example of Instructions

Variable Definition

No.	Variable Name	Address	Variable Description	Variable Type	Initial Value	Power Fail Safeguard
0001	Var1			REAL	0	FALSE

Programing language	Program
Ladder diagram (LD)	
Structured text (ST)	Var1:=SIN(1.5); (*The result Var1 is 0.997495 *)
Continuous Function Chart (CFC)	 (*The result Var1 is 0.997495 *)

3.1.1.13 COS (Cosine)

3.1.1.13.1 Function

Get the cosine value of the input data, and the input data is represented by the radian.

$$\text{Radian (rad)} = \text{Angle} \times \frac{\pi}{180°}$$

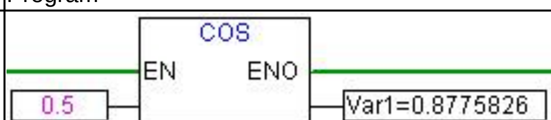
3.1.1.13.2 Input/output Data Type

The input data type may be BYTE, WORD, DWORD, INT, DINT, REAL, LREAL, SINT, USINT, UINT, UDINT, and the output data shall be of the REAL, LREAL types.

3.1.1.13.3 Application Example of Instructions

Variable Definition

No.	Variable Name	Address	Variable Description	Variable Type	Initial Value	Power Fail Safeguard
0001	Var1			REAL	0	FALSE

Programing language	Program
Ladder diagram (LD)	
Structured text (ST)	Var1:=COS(0.5); (*The result Var1 is 0.8775826 *)

Variable Definition	
Continuous Function Chart (CFC)	 (*The result Var1 is 0.8775826 *)

3.1.1.14 TAN (Tangent)

3.1.1.14.1 Function

Get the tangent value of the input data, and the input data is represented by the radian.

$$\text{Radian (rad)} = \text{Angle} \times \frac{\pi}{180°}$$

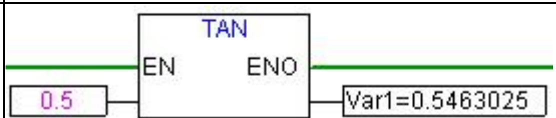
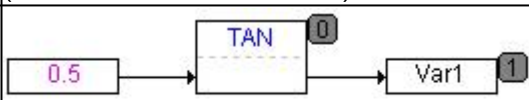
3.1.1.14.2 Input/output Data Type

The input data type may be BYTE, WORD, DWORD, INT, DINT, REAL, LREAL, SINT, USINT, UINT, UDINT, and the output data shall be of the REAL, LREAL types.

3.1.1.14.3 Application Example of Instructions

Variable Definition

No.	Variable Name	Address	Variable Description	Variable Type	Initial Value	Power Fail Safeguard
0001	Var1			REAL	0	FALSE

Programing language	Program
Ladder diagram (LD)	
Structured text (ST)	Var1:=TAN(0.5); (*The result Var1 is 0.5463025 *)
Continuous Function Chart (CFC)	 (*The result Var1 is 0.5463025 *)

3.1.1.15 ASIN (Arc Sine)

3.1.1.15.1 Function

Get the arcsine value of the input data. Input range is in [-1, 1].

LK platform, the output is infinitesimal number when the input exceeds [-1, 1].

LE platform, the output is an invalid value when the input exceeds [-1, 1]. Please see Considerations for Use of Instruction System for the invalid value.

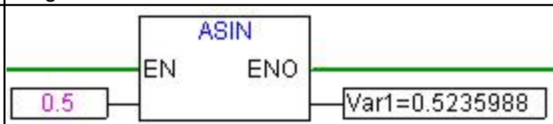
3.1.1.15.2 Input/output Data Type

The input data type may be BYTE, WORD, DWORD, INT, DINT, REAL, LREAL, SINT, USINT, UINT, UDINT, the output data shall be of the REAL, LREAL types, and the output data is represented by the radian.

3.1.1.15.3 Application Example of Instructions

Variable Definition

No.	Variable Name	Address	Variable Description	Variable Type	Initial Value	Power Fail Safeguard
0001	Var1			REAL	0	FALSE

Programing language	Program
Ladder diagram (LD)	
Structured text (ST)	Var1:=ASIN(0.5); (*The result Var1 is 0.5235988 *)
Continuous Function Chart (CFC)	 (*The result Var1 is 0.5235988 *)

3.1.1.16 ACOS (Arc Cosine)

3.1.1.16.1 Function

Get the arccosine value of the input data. Input range is in [-1, 1].

LK platform, the output is infinitesimal number when the input exceeds [-1, 1].

LE platform, the output is an invalid value when the input exceeds [-1, 1]. Please see Considerations for Use of Instruction System for the invalid value.

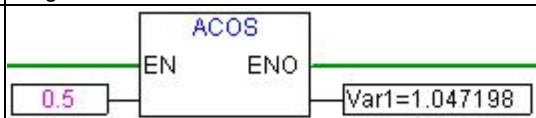
3.1.1.16.2 Input/output Data Type

The input data type may be BYTE, WORD, DWORD, INT, DINT, REAL, LREAL, SINT, USINT, UINT, UDINT, the output data shall be of the REAL, LREAL types, and the output data is represented by the radian.

3.1.1.16.3 Application Example of Instructions

Variable Definition

No.	Variable Name	Address	Variable Description	Variable Type	Initial Value	Power Fail Safeguard
0001	Var1			REAL	0	FALSE

Programing language	Program
Ladder diagram (LD)	
Structured text (ST)	Var1:=ACOS(0.5); (*The result Var1 is 1.047198 *)
Continuous Function Chart (CFC)	 (*The result Var1 is 1.047198 *)

3.1.1.17 ATAN (Arc Tangent)

3.1.1.17.1 Function

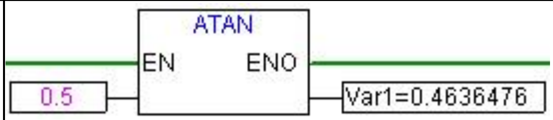
Get the arc tangent value of the input data.

3.1.1.17.2 Input/output Data Type

The input data type may be BYTE, WORD, DWORD, INT, DINT, REAL, LREAL, SINT, USINT, UINT, UDINT, the output data shall be of the REAL, LREAL types, and the output data is represented by the radian.

3.1.1.17.3 Application Example of Instructions

Variable Definition						
No.	Variable Name	Address	Variable Description	Variable Type	Initial Value	Power Fail Safeguard
0001	Var1			REAL	0	FALSE

Programing language	Program
Ladder diagram (LD)	
Structured text (ST)	Var1:=ATAN(0.5); (*The result Var1 is 0.4636476 *)
Continuous Function Chart (CFC)	 (*The result Var1 is 0.4636476 *)

3.1.1.18 EXPT (Power)

3.1.1.18.1 Function

Exponentiation operator is based on x as power base and y as power exponent. IN1 is as x, and IN2 is as y.

LK platform, if x and y are close to 0, the output is 1. If x is close to 0 and y is less than 0, the output is infinitesimal number. When x is less than 0 and y is not integer, the output is "1.#QNAN". The output is an invalid value when the overflow occurs, please see Considerations for Use of Instruction System for the invalid value.

LE platform, if x =0 and y≤0, the output is an invalid value (in simulation mode, x=0 and y=0, the output is 1). If x≠0 and y=0, the output is 1. The output is an invalid value when the overflow occurs, please see Considerations for Use of Instruction System for the invalid value.

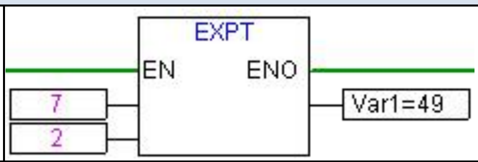
3.1.1.18.2 Input/output Data Type

The input data type may be BYTE, WORD, DWORD, INT, DINT, REAL, LREAL, SINT, USINT, UINT, UDINT, and the output data shall be of the REAL, LREAL types.

3.1.1.18.3 Application Example of Instructions

Variable Definition						
No.	Variable Name	Address	Variable Description	Variable Type	Initial Value	Power Fail Safeguard
0001	Var1			REAL	0	FALSE

Programing language	Program

Variable Definition	
Ladder diagram (LD)	
Structured text (ST)	Var1:=EXPT(7,2); (*The result Var1 is 49 *)
Continuous Function Chart (CFC)	 (*The result Var1 is 49 *)

3.1.1.19 MOVE (Assignment)

3.1.1.19.1 Function

The value of a constant or variable is assigned to another variable.

3.1.1.19.2 Input/output Data Type

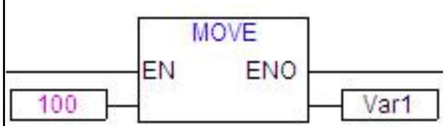
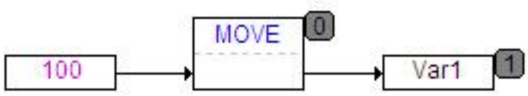
BYTE, WORD, DWORD, SINT, USINT, INT, UINT, DINT, UDINT, REAL, TIME, DT, BOOL, ARRAY, LREAL , STRING, TOD, DATE.

ⓘLE platform does not support LREAL, STRING type data.

3.1.1.19.3 Application Example of Instructions

Variable Definition

No.	Variable Name	Address	Variable Description	Variable Type	Initial Value	Power Fail Safeguard
0001	Var1			INT	0	FALSE

Programing language	Program
Ladder diagram (LD)	 (*The result Var1 is 100*)
Structured text (ST)	Var1:=100; (*The result Var1 is 100*)
Continuous Function Chart (CFC)	 (*The result Var1 is 100*)

ⓘThe character length of the right variable shall prevail for the selection when the length of two variables connected by the MOVE block is not equal during the assignment operation of the STRING variable.

3.1.2 Logic Operation

3.1.2.1 AND(And)

3.1.2.1.1 Function

The AND operation is carried out for the variable or constant.

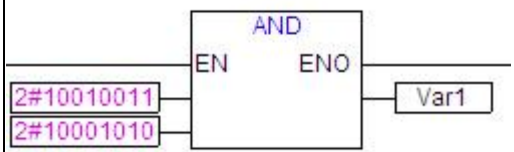
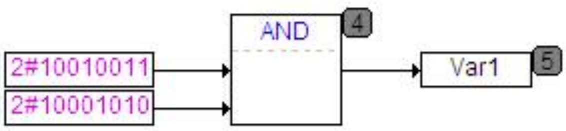
3.1.2.1.2 Input/output Data Type

BOOL, BYTE, WORD and DWORD.

3.1.2.1.3 Application Example of Instructions

Variable Definition

No.	Variable Name	Address	Variable Description	Variable Type	Initial Value	Power Fail Safeguard
0001	Var1			BYTE	0	FALSE

Programing language	Program
Ladder diagram (LD)	 (*The result Var1 is 2#10000010*)
Structured text (ST)	Var1:=2#10010011 AND 2#10001010; (*The result Var1 is 2#10000010*)
Continuous Function Chart (CFC)	 (*The result Var1 is 2#10000010*)

ⓘThe character length of the right variable shall prevail for the selection when the length of two variables connected by the MOVE block is not equal during the assignment operation of the STRING variable.

3.1.2.2 OR(Or)

3.1.2.2.1 Function

The OR operation is carried out for the variable or constant.

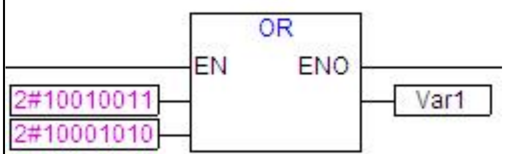
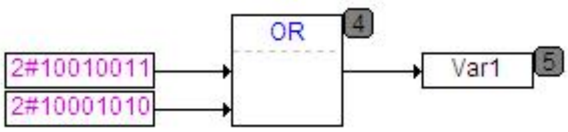
3.1.2.2.2 Input/output Data Type

BOOL, BYTE, WORD and DWORD

3.1.2.2.3 Application Example of Instructions

Variable Definition

No.	Variable Name	Address	Variable Description	Variable Type	Initial Value	Power Fail Safeguard
0001	Var1			BYTE	0	FALSE

Programing language	Program
Ladder diagram (LD)	 (*The result Var1 is 2#10011011*)
Structured text (ST)	Var1:=2#10010011 OR 2#10001010 ; (*The result Var1 is 2#10011011*)
Continuous Function Chart (CFC)	 (*The result Var1 is 2#10011011*)

3.1.2.3 XOR(Exclusive Or)

3.1.2.3.1 Function

The XOR operation is carried out for the variable or constant.

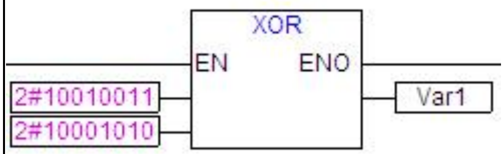
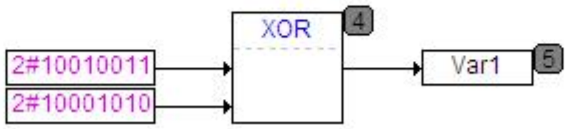
3.1.2.3.2 Input/output Data Type

BOOL, BYTE, WORD and DWORD.

3.1.2.3.3 Application Example of Instructions

Variable Definition

No.	Variable Name	Address	Variable Description	Variable Type	Initial Value	Power Fail Safeguard
0001	Var1			BYTE	0	FALSE

Programing language	Program
Ladder diagram (LD)	 (*The result Var1 is 2#00011001*)
Structured text (ST)	Var1:=2#10010011 XOR 2#10001010 ; (*The result Var1 is 2#00011001*)
Continuous Function Chart (CFC)	 (*The result Var1 is 2#00011001*)

3.1.2.4 NOT(Not)

3.1.2.4.1 Function

The NOT operation is carried out for the variable or constant in the bit-by-bit way.

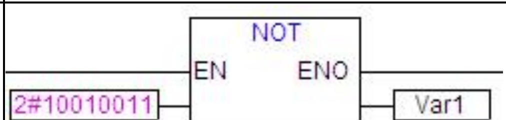
3.1.2.4.2 Input/output Data Type

BOOL, BYTE, WORD and DWORD.

3.1.2.4.3 Application Example of Instructions

Variable Definition

No.	Variable Name	Address	Variable Description	Variable Type	Initial Value	Power Fail Safeguard
0001	Var1			BYTE	0	FALSE

Programing language	Program
Ladder diagram (LD)	

Variable Definition	
	(*The result Var1 is 2#01101100*)
Structured text (ST)	Var1:= NOT 2#10010011 ; (*The result Var1 is 2#01101100*)
Continuous Function Chart (CFC)	 (*The result Var1 is 2#01101100*)

3.1.3 Comparison Operation

3.1.3.1 GT (Greater Than)

3.1.3.1.1 Function

Judge the size of two operands. It will output TRUE when the first operand is greater than the second operand. Otherwise, it will output FALSE.

3.1.3.1.2 Input/output Data Type

Input data type: BOOL, BYTE, WORD, DWORD, SINT, USINT, INT, UINT, DINT, UDINT, REAL, TIME, DATE, TOD, DT, LREAL, STRING.

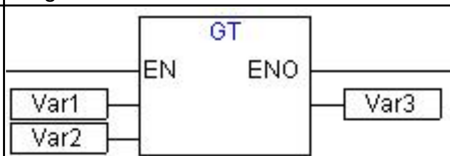
Output data type: BOOL.

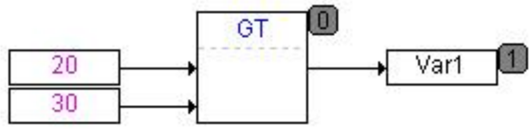
ⓘLE platform does not support LREAL, STRING type data.

3.1.3.1.3 Application Example of Instructions

Variable Definition

No.	Variable Name	Address	Variable Description	Variable Type	Initial Value	Power Fail Safeguard
0001	Var1			INT	0	FALSE
0002	Var2			INT	0	FALSE
0003	Var3			BOOL	FALSE	FALSE

Programing language	Program
Ladder diagram (LD)	 (*Result Var3 is FALSE*)
Structured text (ST)	Var3:=20>30; (*Result Var3 is FALSE*)

Variable Definition	
Continuous Function Chart (CFC)	 (*Result Var1 is FALSE*)

3.1.3.2 LT (Less Than)

3.1.3.2.1 Function

Judge the size of two operands. It will return TRUE when the first operand is less than the second operand. Otherwise, the result will be FALSE.

3.1.3.2.2 Input/output Data Type

Input data type: BOOL, BYTE, WORD, DWORD, SINT, USINT, INT, UINT, DINT, UDINT, REAL, TIME, DATE, TOD, DT, LREAL, STRING.

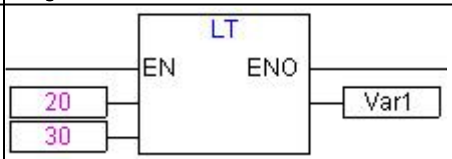
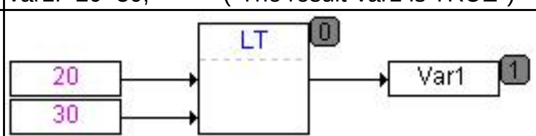
Output data type: BOOL.

ⓘLE platform does not support LREAL, STRING type data.

3.1.3.2.3 Application Example of Instructions

Variable Definition

No.	Variable Name	Address	Variable Description	Variable Type	Initial Value	Power Fail Safegua
0001	Var1			BOOL	FALSE	FALSE

Programing language	Program
Ladder diagram (LD)	 (*The result Var1 is TURE*)
Structured text (ST)	Var1:=20<30; (*The result Var1 is TRUE*)
Continuous Function Chart (CFC)	 (*The result Var1 is TURE*)

3.1.3.3 GE (Greater Than or Equal To)

3.1.3.3.1 Function

Judge the size of two operands. It will return TRUE when the first operand is greater than or equal to the second operand. Otherwise, the result will be FALSE.

3.1.3.3.2 Input/output Data Type

Input data type: BOOL, BYTE, WORD, DWORD, SINT, USINT, INT, UINT, DINT, UDINT, REAL, TIME, DATE, TOD, DT, LREAL and STRING.

Output data type: BOOL.

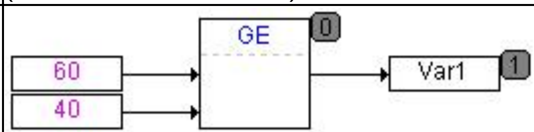
ⓘLE platform does not support LREAL, STRING type data.

3.1.3.3.3 Application Example of Instructions

<table>
<tr><td colspan="2">Variable Definition</td></tr>
<tr><td colspan="2">

No.	Variable Name	Address	Variable Description	Variable Type	Initial Value	Power Fail Safegua
0001	Var1			BOOL	FALSE	FALSE

</td></tr>
<tr><td>Programing language</td><td>Program</td></tr>
<tr><td>Ladder diagram (LD)</td><td>

(*The result Var1 is TURE*)</td></tr>
<tr><td>Structured text (ST)</td><td>Var1:=60>=40;
(*The result Var1 is TRUE*)</td></tr>
<tr><td>Continuous Function Chart (CFC)</td><td>

(*The result Var1 is TURE*)</td></tr>
</table>

3.1.3.4 LE (Less or Equal)

3.1.3.4.1 Function

Judge the size of two operands. It will return TRUE when the first operand is less than or equal to the second operand. Otherwise, the result will be FALSE.

3.1.3.4.2 Input/output Data Type

Input data type: BOOL, BYTE, WORD, DWORD, SINT, USINT, INT, UINT, DINT, UDINT, REAL, TIME, DATE, TOD, DT, LREAL, STRING.

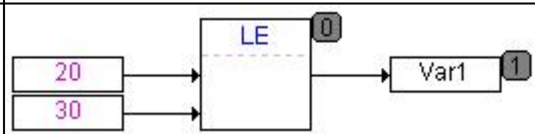
Output data type: BOOL.

ⓘLE platform does not support LREAL, STRING type data.

3.1.3.4.3 Application Example of Instructions

Variable Definition

No.	Variable Name	Address	Variable Description	Variable Type	Initial Value	Power Fail Safegua
0001	Var1			BOOL	FALSE	FALSE

Programing language	Program
Ladder diagram (LD)	 (*The result Var1 is TURE*)
Structured text (ST)	Var1:=20<=30; (*The result Var1 is TRUE*)
Continuous Function Chart (CFC)	 (*The result Var1 is TURE*)

3.1.3.5 EQ (Equal)

3.1.3.5.1 Function

Judge whether two operands are equal to each other. It will return TRUE when the first operand is equal to the second operand. Otherwise, the result will be FALSE.

3.1.3.5.2 Input/output Data Type

Input data type: BOOL, BYTE, WORD, DWORD, SINT, USINT, INT, UINT, DINT, UDINT, REAL, TIME, DATE, TOD, DT, LREAL, STRING.

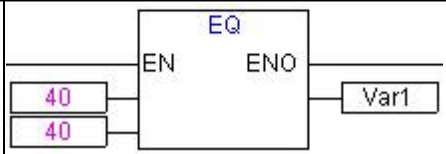
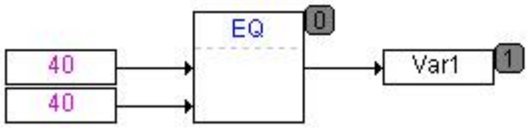
Output data type: BOOL.

ⓘLE platform does not support LREAL, STRING type data.

3.1.3.5.3 Application Example of Instructions

Variable Definition

No.	Variable Name	Address	Variable Description	Variable Type	Initial Value	Power Fail Safegua
0001	Var1			BOOL	FALSE	FALSE

Programing language	Program
Ladder diagram (LD)	 (*The result Var1 is TURE*)
Structured text (ST)	Var1:=40=40; (*The result Var1 is TRUE*)
Continuous Function Chart (CFC)	 (*The result Var1 is TURE*)

3.1.3.6 NE (Not Equal)

3.1.3.6.1 Function

Judge whether two operands are not equal to each other. It will return TRUE when the first operand is not equal to the second operand. Otherwise, the result will be FALSE.

3.1.3.6.2 Input/output Data Type

Input data type: BOOL, BYTE, WORD, DWORD, SINT, USINT, INT, UINT, DINT, UDINT, REAL, TIME, DATE, TOD, DT, LREAL, STRING.

Output data type: BOOL.

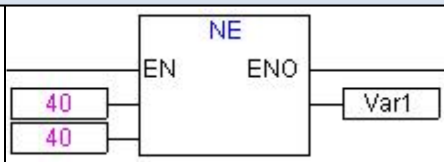
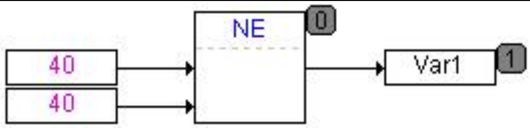
ⓘLE platform does not support LREAL, STRING type data.

3.1.3.6.3 Application Example of Instructions

Variable Definition

No.	Variable Name	Address	Variable Description	Variable Type	Initial Value	Power Fail Safegua
0001	Var1			BOOL	FALSE	FALSE

Programing language	Program

Variable Definition	
Ladder diagram (LD)	 (*The result Var1 is FALSE*)
Structured text (ST)	Var1:=40<>40; (*The result Var1 is FALSE*)
Continuous Function Chart (CFC)	 (*The result Var1 is FALSE*)

3.1.4 Select Operation

3.1.4.1 SEL (One Out of Two)

3.1.4.1.1 Function

Select one from two input data as the output by the selector switch. The output is the first input data when the selector switch is FALSE and the second input data when the selector switch is TRUE.

3.1.4.1.2 Instruction Format

OUT:= SEL(G,IN0,IN1). Of which, G is the selector switch, IN0 and IN1 are the first input data and the second input data respectively.

3.1.4.1.3 Input/output Data Type

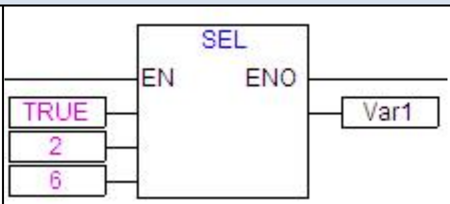
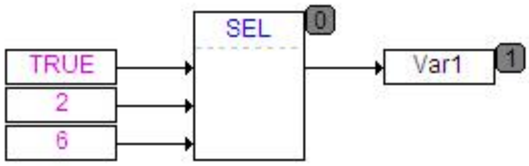
G shall be the BOOL type, IN0, IN1 and the output data may be any data type (excluding the STRING, POINTER, ARRAY type).

3.1.4.1.4 Application Example of Instructions

Variable Definition

No.	Variable Name	Address	Variable Description	Variable Type	Initial Value	Power Fail Safeguard
0001	Var1			INT	0	FALSE

Programing language	Program

Variable Definition	
Ladder diagram (LD)	 (*The result Var1 is 6 *)
Structured text (ST)	Var1:=SEL(TRUE,3,4); (*The result Var1 is 4 *)
Continuous Function Chart (CFC)	 (*The result Var1 is 6 *)

3.1.4.2 MAX (Maximum)

3.1.4.2.1 Function

Select the maximum value from two input data as the output.

3.1.4.2.2 Instruction Format

OUT:=MAX(IN0, IN1). Of which, IN0 and IN1 are the first input data and the second input data respectively, and OUT is the output data.

3.1.4.2.3 Input/output Data Type

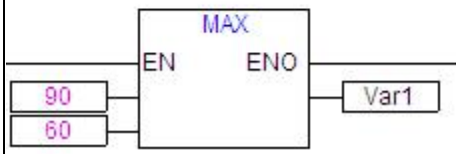
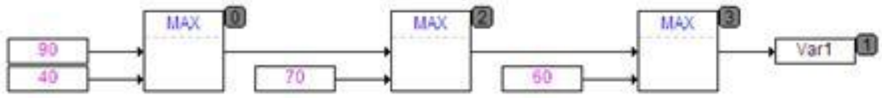
IN0, IN1 and OUT may be any data type (excluding the BOOL, STRING, POINTER, ARRAY type).

3.1.4.2.4 Application Example of Instructions

Variable Definition

No.	Variable Name	Address	Variable Description	Variable Type	Initial Value	Power Fail Safeguard
0001	Var1			INT	0	FALSE
0002	Var2			INT	0	FALSE

Programing language	Program

Variable Definition	
Ladder diagram (LD)	 (*The result Var1 is 90 *)
Structured text (ST)	Var1:=MAX(90,60); (*The result Var1 is 90 *) Var2:=MAX(40,MAX(90,60)); (*The result Var2 is 90 *)
Continuous Function Chart (CFC)	 (*The result Var1 is 90 *)

3.1.4.3 MIN (Minimum)

3.1.4.3.1 Function

Select the minimum value from two input data as the output.

3.1.4.3.2 Instruction Format

OUT:=MIN(IN0, IN1) . Of which, IN0 and IN1 are the first input data and the second input data respectively, and OUT is the output data.

3.1.4.3.3 Input/output Data Type

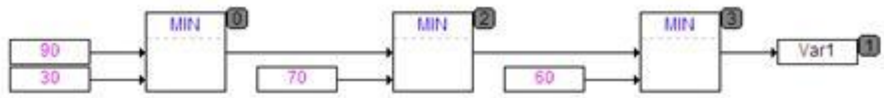
IN0, IN1 and OUT may be any data type (excluding the BOOL, STRING, POINTER, ARRAY type).

3.1.4.3.4 Application Example of Instructions

Variable Definition

No.	Variable Name	Address	Variable Description	Variable Type	Initial Value	Power Fail Safeguard
0001	Var1			INT	0	FALSE
0002	Var2			INT	0	FALSE

Programing language	Program
Ladder diagram (LD)	

Variable Definition	
	(*The result Var1 is 60 *)
Structured text (ST)	Var1:=MIN(90,30); (*The result Var1 is 30 *) Var2:=MIN(MIN(90,30),60); (*The result Var2 is 30 *)
Continuous Function Chart (CFC)	 (*The result Var1 is 30 *)

3.1.4.4 MUX (One Out of Multiple)

3.1.4.4.1 Function

Select one of multiple input data as the output by the control number.

3.1.4.4.2 Instruction Format

OUT:=MUX(K,IN0,…,INn). Of which, K is the control number, IN0,…,INn are the input data, OUT is the output result. Select the No.INk input data as the output when the control number is K.

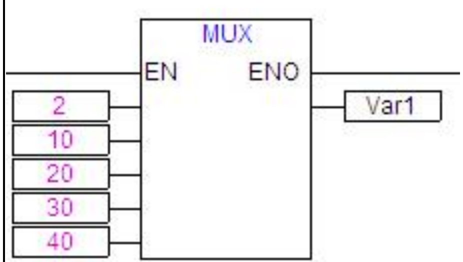
3.1.4.4.3 Input/output Data Type

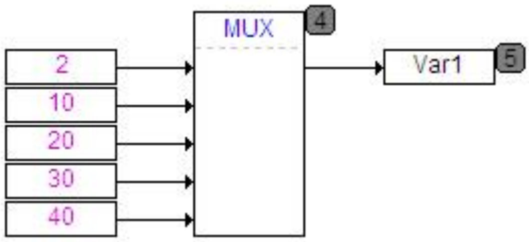
IN0,…, INn and OUT may be any data type (excluding the BOOL, STRING, POINTER, ARRAY type), and K shall be BYTE, WORD, DWORD, SINT, USINT, INT, UINT, DINT, UDINT.

3.1.4.4.4 Application Example of Instructions

Variable Definition

No.	Variable Name	Address	Variable Description	Variable Type	Initial Value	Power Fail Safeguard
0001	Var1			INT	0	FALSE

Programing language	Program
Ladder diagram (LD)	 (*2 is the control data, which corresponds to 30. Hence, it will output 30*)
Structured text (ST)	Var1:=MUX(0,30,40,50,60,70,80);

Variable Definition	
	(*The result Var1 is 30*)
Continuous Function Chart (CFC)	 (*2 is the control data, which corresponds to 30. Hence, it will output 30*)

3.1.4.5 LIMIT (Limit)

3.1.4.5.1 Function

Judge whether the input data is between the minimum value and the maximum value. If yes, take the input data as the output data for the output directly. If the input data is greater than the maximum value, take the maximum value as the output value. If the input data is less than the minimum value, take the minimum value as the output value.

3.1.4.5.2 Instruction Format

OUT :=LIMIT(Min,IN,Max).

3.1.4.5.3 Input/output Data Type

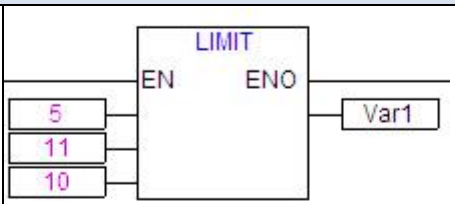
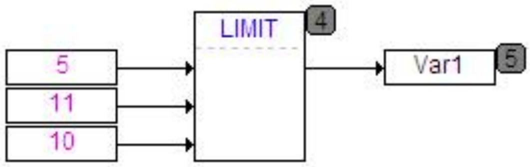
IN and OUT may be any data type (excluding the BOOL, STRING, POINTER, ARRAY type).

3.1.4.5.4 Application Example of Instructions

Variable Definition

No.	Variable Name	Address	Variable Description	Variable Type	Initial Value	Power Fail Safeguard
0001	Var1			INT	0	FALSE

Programing language	Program

Variable Definition	
Ladder diagram (LD)	 (*11 is the input data, 5 is the minimum value, 10 is the maximum value, and the result Var1 is 10*)
Structured text (ST)	Var1:=LIMIT(30,90,80); (*The result Var1 is 80 *)
Continuous Function Chart (CFC)	 (*11 is the input data, 5 is the minimum value, 10 is the maximum value, and the result Var1 is 10*)

3.1.5 Shift Operation

3.1.5.1 SHL (Shift Left)

3.1.5.1.1 Function

The bitwise shift left will be carried out for the operand, and the leftmost bit will not be processed, but the rightmost will be supplemented with zero automatically.

3.1.5.1.2 Input/output Data Type

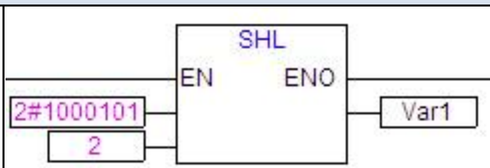
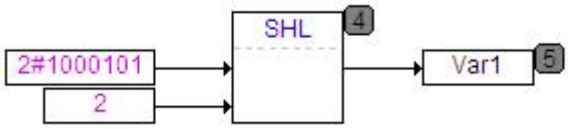
BYTE, INT, WORD, DWORD, SINT, UINT, USINT, DINT.

3.1.5.1.3 Application Example of Instructions

Variable Definition

No.	Variable Name	Address	Variable Description	Variable Type	Initial Value	Power Fail Safeguard
0001	Var1			BYTE	0	FALSE

Programing language	Program

Variable Definition	
Ladder diagram (LD)	 (*The result Var1 is 2#10100*)
Structured text (ST)	Var1:=SHL(16#45,2); (*The result Var1 is 16#14*)
Continuous Function Chart (CFC)	 (*The result Var1 is 2#10100*)

3.1.5.2 SHR (Shift Right)

3.1.5.2.1 Function

The bitwise shift right will be carried out for the operand, and the rightmost bit will not be processed, but the leftmost will be supplemented with zero automatically.

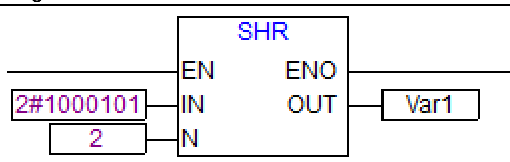
3.1.5.2.2 Input/output Data Type

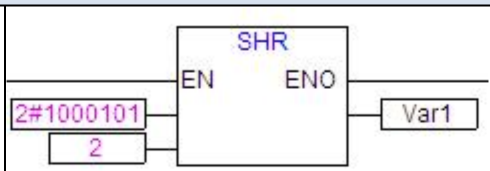
BYTE, INT, WORD, DWORD, SINT, UINT, USINT, DINT.

3.1.5.2.3 Application Example of Instructions

Variable Definition

No.	Variable Name	Address	Variable Description	Variable Type	Initial Value	Power Fail Safeguard
0001	Var1			BYTE	0	FALSE

Programing language	Program
Ladder diagram (LD)	 (*The result Var1 is 2#10001*)
Structured text (ST)	Var1:=SHR(16#45,2); (*The result Var1 is 16#11*)

Variable Definition	
Continuous Function Chart (CFC)	 (*The result Var1 is 2#10001*)

3.1.5.3 ROL (Ring Shift Left)

3.1.5.3.1 Function

The ring shift left will be carried out for the operand, and the leftmost bit will be supplemented to the lowest bit on the right directly.

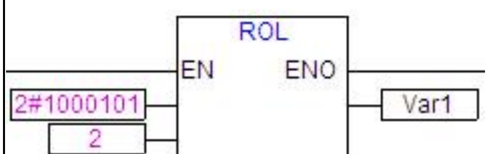
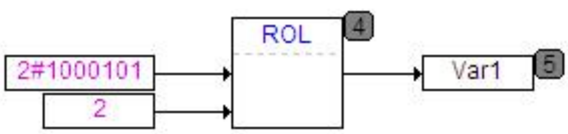
3.1.5.3.2 Input/output Data Type

BYTE, INT, WORD, DWORD, SINT, UINT, USINT, DINT.

3.1.5.3.3 Application Example of Instructions

Variable Definition

No.	Variable Name	Address	Variable Description	Variable Type	Initial Value	Power Fail Safeguard
0001	Var1			BYTE	0	FALSE

Programing language	Program
Ladder diagram (LD)	 (*The result Var1 is 2#10101*)
Structured text (ST)	Var1:=ROL(16#45,2); (*The result Var1 is 16#15*)
Continuous Function Chart (CFC)	 (*The result Var1 is 2#10101*)

3.1.5.4 ROR (Ring Shift Right)

3.1.5.4.1 Function

The ring shift right will be carried out for the operand, and the rightmost bit will be supplemented to the highest bit on the left directly.

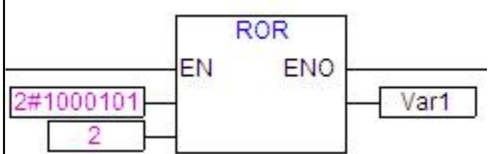
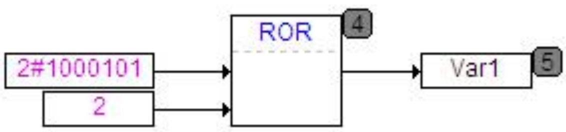
3.1.5.4.2 Input/output Data Type

BYTE, INT, WORD, DWORD, SINT, UINT ,USINT, DINT.

3.1.5.4.3 Application Example of Instructions

Variable Definition

No.	Variable Name	Address	Variable Description	Variable Type	Initial Value	Power Fail Safeguard
0001	Var1			BYTE	0	FALSE

Programing language	Program
Ladder diagram (LD)	 (*The result Var1 is 2#1010001*)
Structured text (ST)	Var1:=ROR(16#45,2); (*The result Var1 is 16#51*)
Continuous Function Chart (CFC)	 (*The result Var1 is 2#1010001*)

3.1.6 Data Type Conversion

AutoThink provides 200 data type conversion instructions, which are used to convert the various data types.

Syntax: <TYPE1> _TO_ <TYPE2>

It is forbidden to implicitly convert "larger" data types to "smaller" data types, and it is possible to lose information when changing from a larger data type to a smaller data type.

If the converted value exceeds the storage range of the target data type, the high byte is ignored. Such as converting an INT type to a BYTE type, or converting a DINT type to a WORD type.

The following Table lists all the data type conversion instructions. This Section describes the data type conversion instructions.

Data Type Conversion Instructions List

BOOL_TO_<TYPE>	BYTE_TO_<TYPE>	DATE_TO_<TYPE>	DINT_TO_<TYPE>
BOOL_TO_BYTE BOOL_TO_DATE BOOL_TO_DINT BOOL_TO_DT BOOL_TO_DWORD BOOL_TO_INT BOOL_TO_LINT BOOL_TO_LREAL BOOL_TO_LWORD BOOL_TO_REAL BOOL_TO_STRING BOOL_TO_SINT BOOL_TO_TIME BOOL_TO_TOD BOOL_TO_UDINT BOOL_TO_UINT BOOL_TO_ULINT BOOL_TO_WORD	BYTE_TO_BOOL BYTE_TO_DATE BYTE_TO_DINT BYTE_TO_DT BYTE_TO_DWORD BYTE_TO_INT BYTE_TO_LINT BYTE_TO_LREAL BYTE_TO_LWORD BYTE_TO_REAL BYTE_TO_SINT BYTE_TO_STRING BYTE_TO_TIME BYTE_TO_TOD BYTE_TO_UDINT BYTE_TO_UINT BYTE_TO_ULINT BYTE_TO_USINT BYTE_TO_WORD	DATE_TO_BOOL DATE_TO_BYTE DATE_TO_DINT DATE_TO_DWORD DATE_TO_INT DATE_TO_LINT DATE_TO_LREAL DATE_TO_LWORD DATE_TO_REAL DATE_TO_SINT DATE_TO_STRING DATE_TO_UDINT DATE_TO_UINT DATE_TO_ULINT DATE_TO_USINT DATE_TO_WORD	DINT_TO_BOOL DINT_TO_BYTE DINT_TO_DATE DINT_TO_DT DINT_TO_DWORD DINT_TO_INT DINT_TO_LINT DINT_TO_LREAL DINT_TO_LWORD DINT_TO_REAL DINT_TO_SINT DINT_TO_STRING DINT_TO_TIME DINT_TO_TOD DINT_TO_UDINT DINT_TO_UINT DINT_TO_ULINT DINT_TO_USINT DINT_TO_WORD
DT_TO_<TYPE>	DWORD_TO_<TYPE>	INT_TO_<TYPE>	LINT_TO_<TYPE>
DT_TO_BOOL DT_TO_BYTE DT_TO_DINT DT_TO_DWORD DT_TO_INT DT_TO_LINT DT_TO_LREAL DT_TO_LWORD DT_TO_REAL DT_TO_SINT DT_TO_STRING DT_TO_UDINT DT_TO_UINT DT_TO_ULINT DT_TO_USINT DT_TO_WORD	DWORD_TO_BOOL DWORD_TO_BYTE DWORD_TO_DATE DWORD_TO_DINT DWORD_TO_DT DWORD_TO_INT DWORD_TO_LINT DWORD_TO_LREAL DWORD_TO_LWORD DWORD_TO_REAL DWORD_TO_SINT DWORD_TO_STRING DWORD_TO_TIME DWORD_TO_TOD DWORD_TO_UDINT DWORD_TO_UINT DWORD_TO_ULINT DWORD_TO_USINT DWORD_TO_WORD	INT_TO_BOOL INT_TO_BYTE INT_TO_DATE INT_TO_DINT INT_TO_DT INT_TO_DWORD INT_TO_LINT INT_TO_LREAL INT_TO_LWORD INT_TO_REAL INT_TO_SINT INT_TO_STRING INT_TO_TIME INT_TO_TOD INT_TO_UDINT INT_TO_UINT INT_TO_ULINT INT_TO_USINT INT_TO_WORD	LINT_TO_BOOL LINT_TO_BYTE LINT_TO_DATE LINT_TO_DINT LINT_TO_DT LINT_TO_DWORD LINT_TO_INT LINT_TO_LREAL LINT_TO_LWORD LINT_TO_REAL LINT_TO_SINT LINT_TO_STRING LINT_TO_TIME LINT_TO_TOD LINT_TO_UDINT LINT_TO_UINT LINT_TO_ULINT LINT_TO_USINT LINT_TO_WORD
LREAL_TO_<TYPE>	LWORD_TO_<TYPE>	REAL_TO_<TYPE>	SINT_TO_<TYPE>
LREAL_TO_BOOL LREAL_TO_BYTE LREAL_TO_DATE LREAL_TO_DINT LREAL_TO_DT LREAL_TO_DWORD LREAL_TO_INT LREAL_TO_LINT LREAL_TO_LWORD LREAL_TO_REAL LREAL_TO_SINT LREAL_TO_STRING LREAL_TO_TIME LREAL_TO_TOD LREAL_TO_UDINT	LWORD_TO_BOOL LWORD_TO_BYTE LWORD_TO_DATE LWORD_TO_DINT LWORD_TO_DT LWORD_TO_DWORD LWORD_TO_INT LWORD_TO_LINT LWORD_TO_LREAL LWORD_TO_REAL LWORD_TO_SINT LWORD_TO_STRING LWORD_TO_TIME LWORD_TO_TOD LWORD_TO_UDINT	REAL_TO_BOOL REAL_TO_BYTE REAL_TO_DATE REAL_TO_DINT REAL_TO_DT REAL_TO_DWORD REAL_TO_INT REAL_TO_LINT REAL_TO_LREAL REAL_TO_LWORD REAL_TO_SINT REAL_TO_STRING REAL_TO_TIME REAL_TO_TOD REAL_TO_UDINT	SINT_TO_BOOL SINT_TO_BYTE SINT_TO_DATE SINT_TO_DINT SINT_TO_DT SINT_TO_DWORD SINT_TO_INT SINT_TO_LINT SINT_TO_LREAL SINT_TO_LWORD SINT_TO_REAL SINT_TO_STRING SINT_TO_TIME SINT_TO_TOD SINT_TO_UDINT

BOOL_TO_<TYPE>	BYTE_TO_<TYPE>	DATE_TO_<TYPE>	DINT_TO_<TYPE>
LREAL _TO_UINT LREAL _TO_ULINT LREAL _TO_USINT LREAL _TO_WORD	LWORD _TO_UINT LWORD _TO_ULINT LWORD _TO_USINT LWORD _TO_WORD	REAL _TO_UINT REAL _TO_ULINT REAL _TO_USINT REAL _TO_WORD	SINT _TO_UINT SINT _TO_ULINT SINT _TO_USINT SINT _TO_WORD
STRING_TO_<TYPE>	TIME_TO_<TYPE>	TOD_TO_<TYPE>	UDINT_TO_<TYPE>
STRING _TO_BOOL STRING _TO_BYTE STRING _TO_DATE STRING _TO_DINT STRING _TO_DT STRING _TO_DWORD STRING _TO_INT STRING _TO_LINT STRING _TO_LREAL STRING _TO_LWORD STRING _TO_REAL STRING _TO_SINT STRING _TO_TIME STRING _TO_TOD STRING _TO_UDINT STRING _TO_UINT STRING _TO_ULINT STRING _TO_USINT STRING _TO_WORD	TIME_TO_BOOL TIME_TO_BYTE TIME_TO_DINT TIME_TO_DWORD TIME_TO_INT TIME_TO_LINT TIME_TO_LREAL TIME_TO_ LWORD TIME_TO_REAL TIME_TO_SINT TIME_TO_STRING TIME_TO_UDINT TIME_TO_UINT TIME_TO_ULINT TIME_TO_USINT TIME_TO_WORD	TOD_TO_BOOL TOD_TO_BYTE TOD_TO_DINT TOD_TO_DWORD TOD_TO_INT TOD_TO_LINT TOD_TO_LREAL TOD_TO_LWORD TOD_TO_REAL TOD_TO_SINT TOD_TO_STRING TOD_TO_UDINT TOD_TO_UINT TOD_TO_ULINT TOD_TO_USINT TOD_TO_WORD	UDINT_TO_BOOL UDINT_TO_BYTE UDINT_TO_DATE UDINT_TO_DINT UDINT_TO_DT UDINT_TO_DWORD UDINT_TO_INT UDINT_TO_LINT UDINT_TO_LREAL UDINT_TO_LWORD UDINT_TO_REAL UDINT_TO_SINT UDINT_TO_ STRING UDINT_TO_TIME UDINT_TO_TOD UDINT_TO_UINT UDINT_TO_ULINT UDINT_TO_USINT UDINT_TO_WORD
UINT_TO_<TYPE>	ULINT_TO_<TYPE>	USINT_TO_<TYPE>	WORD_TO_<TYPE>
UINT_TO_BOOL UINT_TO_BYTE UINT_TO_DATE UINT_TO_DINT UINT_TO_DT UINT_TO_DWORD UINT_TO_INT UINT_TO_LINT UINT_TO_LREAL UINT_TO_LWORD UINT_TO_REAL UINT_TO_SINT UINT_TO_ STRING UINT_TO_TIME UINT_TO_TOD UINT_TO_UDINT UINT_TO_ULINT UINT_TO_USINT UINT_TO_WORD	ULINT_TO_BOOL ULINT_TO_BYTE ULINT_TO_DATE ULINT_TO_DINT ULINT_TO_DT ULINT_TO_DWORD ULINT_TO_INT ULINT_TO_LINT ULINT_TO_LREAL ULINT_TO_LWORD ULINT_TO_REAL ULINT_TO_SINT ULINT_TO_ STRING ULINT_TO_TIME ULINT_TO_TOD ULINT_TO_UDINT ULINT_TO_UINT ULINT_TO_USINT ULINT_TO_WORD	USINT_TO_BOOL USINT_TO_BYTE USINT_TO_DATE USINT_TO_DINT USINT_TO_DT USINT_TO_DWORD USINT_TO_INT USINT_TO_LINT USINT_TO_LREAL USINT_TO_LWORD USINT_TO_REAL USINT_TO_SINT USINT_TO_ STRING USINT_TO_TIME USINT_TO_TOD USINT_TO_UDINT USINT_TO_UINT USINT_TO_ULINT USINT_TO_WORD	WORD _TO_BOOL WORD _TO_BYTE WORD _TO_DATE WORD _TO_DINT WORD _TO_DT WORD _TO_DWORD WORD _TO_INT WORD _TO_LINT WORD _TO_LREAL WORD _TO_LWORD WORD _TO_REAL WORD _TO_SINT WORD _TO_ STRING WORD _TO_TIME WORD _TO_TOD WORD _TO_UDINT WORD _TO_UINT WORD _TO_ULINT WORD _TO_ USINT

3.1.6.1 BOOL Type Conversion

3.1.6.1.1 Function

Convert the BOOL type into other data types.

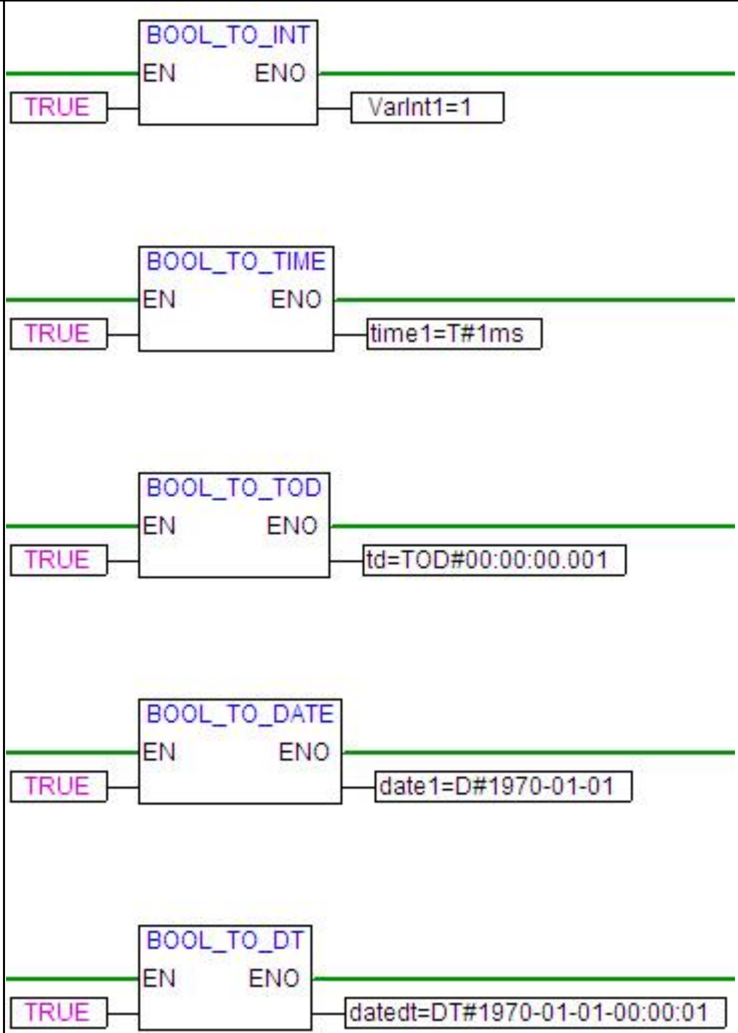
It will output 1 if the input is TRUE when the output is the numeric type and output 0 if the input is FALSE.

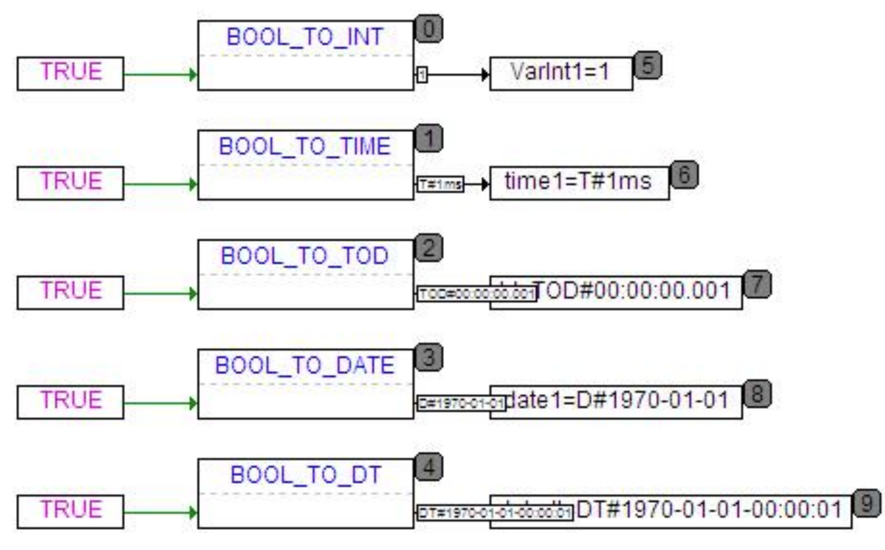
When the output is a string type, if the input is TRUE, the string 'TRUE' is output, and if the input is FALSE, the string 'FALSE' is output.

3.1.6.1.2 Application Example of Instructions

Variable Definition

No.	Variable Name	Address	Variable Description	Variable Type	Initial Value	Power Fail Safeguard
0001	VarInt1			INT	0	FALSE
0002	time1			TIME	T#0MS	FALSE
0003	td			TOD	TOD#00:00:00	FALSE
0004	date1			DATE	D#1970-01-01	FALSE
0005	datedt			DT	DT#1970-01-01-00:00:00	FALSE

<table>
<tr><td>Programing language</td><td>Program</td></tr>
<tr><td>Ladder diagram (LD)</td><td>
</td></tr>
<tr><td>Structured text (ST)</td><td>VarInt1:=BOOL_TO_INT(TRUE); (*The result is 1*)
time1:=BOOL_TO_TIME(TRUE); (*The result is T#1ms*)
td:=BOOL_TO_TOD(TRUE); (*The result is TOD#00:00:00.001*)
date1:=BOOL_TO_DATE(TRUE); (*The result is D#1970-01-01*)
datedt :=BOOL_TO_DT(TRUE); (*The result is DT#1970-01-01-00:00:01*)</td></tr>
</table>

Variable Definition	
Continuous Function Chart (CFC)	

3.1.6.2 BYTE Type Conversion

3.1.6.2.1 Function

Convert the BYTE type into other data types.

For BYTE_TO_BOOL, the output will be TRUE if the input is not equal to 0 and be FALSE if the input is equal to 0.

For BYTE_TO_TIME and BYTE_TO_TOD, the input will be converted in millisecond.

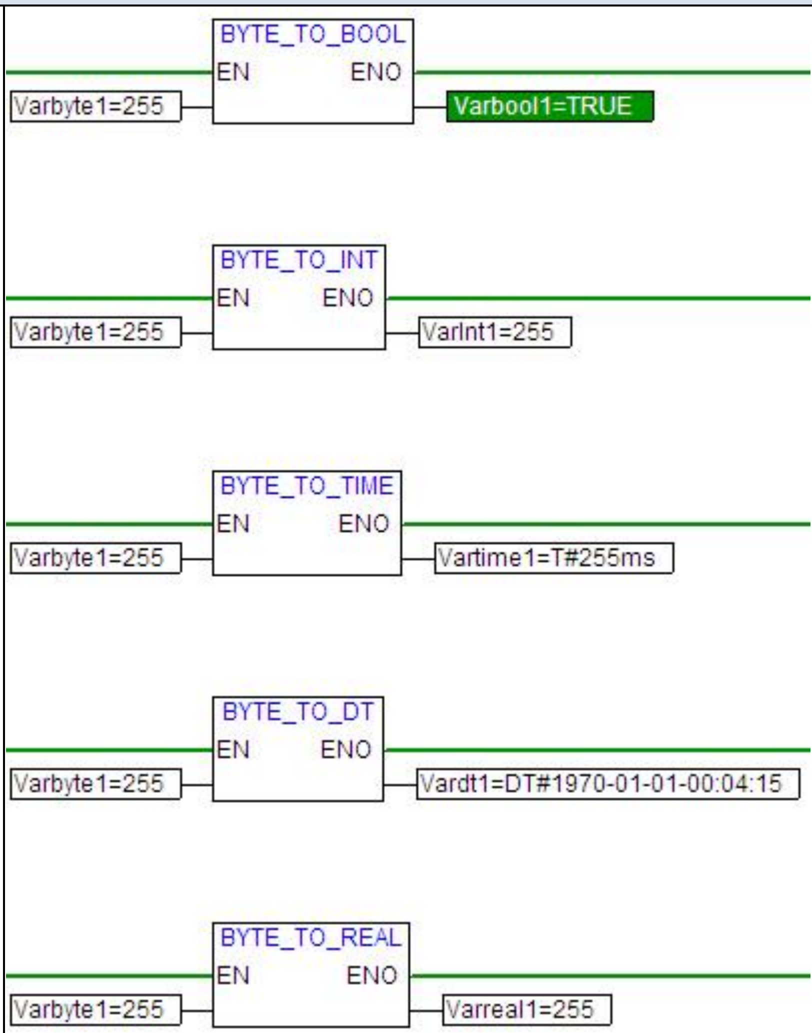
For BYTE_TO_DATE and BYTE_TO_DT, the input will be converted in second.

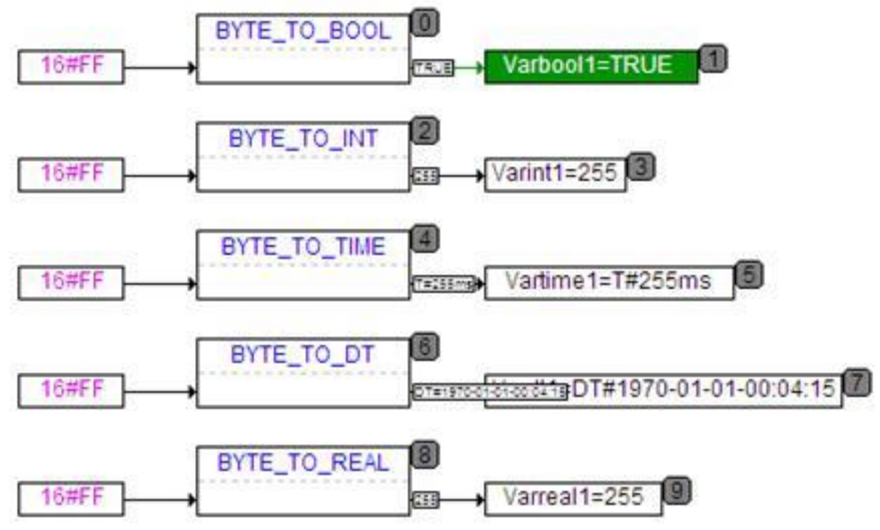
3.1.6.2.2 Application Example of Instructions

Variable Definition

No.	Variable Name	Address	Variable Description	Variable Type	Initial Value	Power Fail Safeguard
0001	Varbool1			BOOL	FALSE	FALSE
0002	Varbyte1			BYTE	0	FALSE
0003	Varint1			INT	0	FALSE
0004	Vartime1			TIME	T#0MS	FALSE
0005	Vardt1			DT	DT#1970-01-01-00:00:00	FALSE
0006	Varreal			REAL	0	FALSE

Programing language	Program

Variable Definition	
Ladder diagram (LD)	
Structured text (ST)	Varbyte1:=16#FF; (*Varbyte1 value*) Varbool1:=BYTE_TO_BOOL(Varbyte1); (*The result is TRUE *) Varint1:=BYTE_TO_INT(Varbyte1); (*The result is 16# FF *) Vartime1:=BYTE_TO_TIME(Varbyte1); (*The result is T#255ms *) Vardt1:=BYTE_TO_DT(Varbyte1); (*The result is DT#1970-01-01-00:04:15 *) Varreal1:=BYTE_TO_REAL(Varbyte1); (*The result is 255 *)

Variable Definition	
Continuous Function Chart (CFC)	

3.1.6.3 DATE Type Conversion

3.1.6.3.1 Function

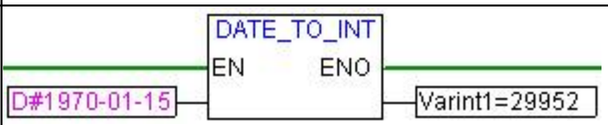
Convert the DATE type into other data types. The date is stored in second internally, which starts from January 1, 1970.

For DATE_TO_BOOL, the output will be TRUE if the input is not equal to 0 and be FALSE if the input is equal to 0.

3.1.6.3.2 Application Example of Instructions

Variable Definition

No.	Variable Name	Address	Variable Description	Variable Type	Initial Value	Power Fail Safegua
0001	Varint1			INT	0	FALSE

Programing language	Program
Ladder diagram (LD)	
Structured text (ST)	VarInt1:=DATE_TO_INT(D#1970-01-15); (*The result is 29952*) Note: If D#1970-01-15 (decimal 14×24×3600=1209600=16#127500) is saved as the INT type variable, the high bit data will lose, and the low bit data 16#7500 will be displayed only, and it is converted into the decimal 29952.

<table>
<tr><th colspan="2">Variable Definition</th></tr>
<tr><td>Continuous Function Chart (CFC)</td><td>
</td></tr>
</table>

3.1.6.4 DINT Type Conversion

3.1.6.4.1 Function

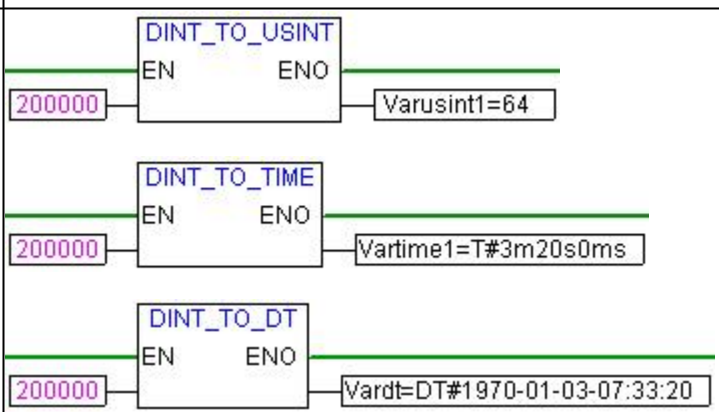
Convert the DINT type into other data types.

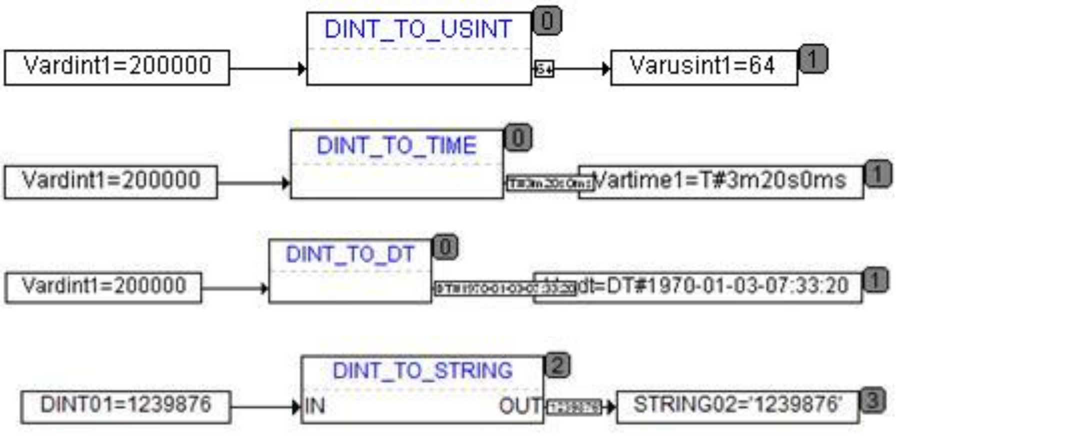
For DINT_TO_BOOL, the output will be TRUE if the input is not equal to 0 and be FALSE if the input is equal to 0.

For DINT_TO_TIME and DINT _TO_TOD, the input will be converted in millisecond.

For DINT_TO_DATE and DINT _TO_DT, the input will be converted in second.

3.1.6.4.2 Application Example of Instructions

<table>
<tr><th colspan="2">Variable Definition</th></tr>
<tr><td colspan="2">
<table>
<tr><th>No.</th><th>Variable Name</th><th>Address</th><th>Variable Description</th><th>Variable Type</th><th>Initial Value</th><th>Power Fail Safegua</th></tr>
<tr><td>0001</td><td>Vardint1</td><td></td><td></td><td>DINT</td><td>0</td><td>FALSE</td></tr>
<tr><td>0002</td><td>Varusint1</td><td></td><td></td><td>USINT</td><td>0</td><td>FALSE</td></tr>
<tr><td>0003</td><td>Vartime1</td><td></td><td></td><td>TIME</td><td>T#0MS</td><td>FALSE</td></tr>
<tr><td>0004</td><td>Vardt</td><td></td><td></td><td>DT</td><td>DT#1970-01-01-00:00:00</td><td>FALSE</td></tr>
</table>
</td></tr>
<tr><td>Programing language</td><td>Program</td></tr>
<tr><td>Ladder diagram (LD)</td><td>
</td></tr>
<tr><td>Structured text (ST)</td><td>Vardint1:=200000;
Varusint1:=DINT_TO_USINT(Vardint1); (*Result 64*)
Note: If the integer 200000 (hexadecimal 16#30D40) is saved as the USINT type variable, the high bit data will lose, and the low bit data 64 (hexadecimal 16#40) will be displayed only.
Vartime1:=DINT_TO_TIME(Vardint1); (*Result T#3m20s0ms*)
Vardt:=DINT_TO_DT(Vardint1); (*Result DT#1970-01-03-07:33:20 *)</td></tr>
</table>

Variable Definition	
Continuous Function Chart (CFC)	

3.1.6.5 DT Type Conversion

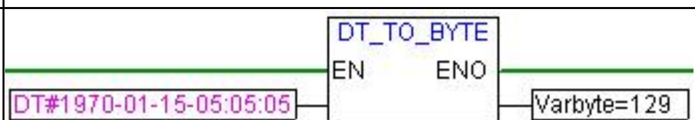
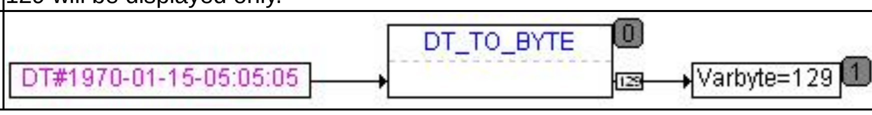
3.1.6.5.1 Function

Convert the DT type into other data types. The date is stored in second internally, which starts from January 1, 1970.

For DT_TO_BOOL, the output will be TRUE if the input is not equal to 0 and be FALSE if the input is equal to 0.

3.1.6.5.2 Application Example of Instructions

Variable Definition						
No.	Variable Name	Address	Variable Description	Variable Type	Initial Value	Power Fail Safegua
0001	Varbyte			BYTE	0	FALSE

Programing language	Program
Ladder diagram (LD)	
Structured text (ST)	Varbyte:=DT_TO_BYTE(DT#1970-01-15-05:05:05); (*Result Varbyte 为 129*) Note: DT#1970-01-15-05:05:05 is converted into seconds, which is (((14*24+5)*60+5)*60+5)=1227905=16#12BC81, and saved as the BYTE type variable, the high 24-bit data will lose, and the low 8-bit data 16#81=129 will be displayed only.
Continuous Function Chart (CFC)	

3.1.6.6 DWORD Type Conversion

3.1.6.6.1 Function

Convert the DWORD type into other data types.

For DWORD_TO_BOOL, the output will be TRUE if the input is not equal to 0 and be FALSE if the input is equal to 0.

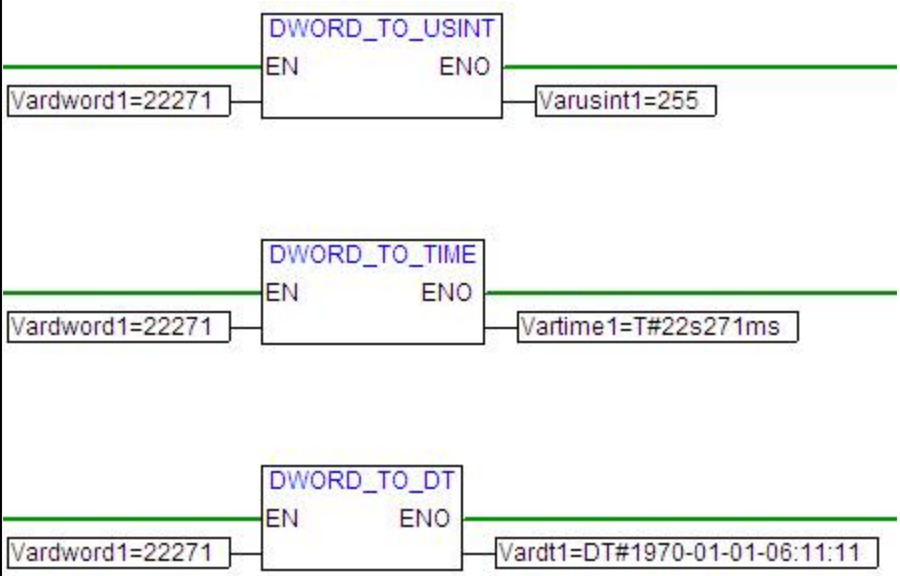
For DWORD_TO_TIME and DWORD_TO_TOD, the input will be converted in millisecond.

For DWORD_TO_DATE and DWORD_TO_DT, the input will be converted in second.

3.1.6.6.2 Application Example of Instructions

Variable Definition

No.	Variable Name	Address	Variable Description	Variable Type	Initial Value	Power Fail Safegua
0001	Varusint1			USINT	0	FALSE
0002	Vardword1			DWORD	0	FALSE
0003	Vartime1			TIME	T#0MS	FALSE
0004	Vardt1			DT	DT#1970-01-01-00:00:00	FALSE

<table>
<tr><td>Programing language</td><td>Program</td></tr>
<tr><td>Ladder diagram (LD)</td><td>
</td></tr>
<tr><td>Structured text (ST)</td><td>Varword1:=16#56FF; (*Varword1 value*)
Varusint1:=DWORD_TO_USINT(Vardword1); (*Result 255 *)
Note: If the integer 16#56FF (decimal 22271) is saved as the USINT type variable, the high bit data will lose, and the low bit data 255 (hexadecimal 16#FF) will be displayed only.
Vartime1:=DWORD_TO_TIME(Vardword1); (*Result T#22s271ms*)
Vardt1:=DWORD_TO_DT(Vardword1); (*Result DT#1970-01-01-06:11:11 *)</td></tr>
</table>

<table>
<tr><th colspan="2">Variable Definition</th></tr>
<tr><td>Continuous Function Chart (CFC)</td><td>
</td></tr>
</table>

3.1.6.7 INT Type Conversion

3.1.6.7.1 Function

Convert the INT type into other data types.

For INT_TO_BOOL, the output will be TRUE if the input is not equal to 0 and be FALSE if the input is equal to 0.

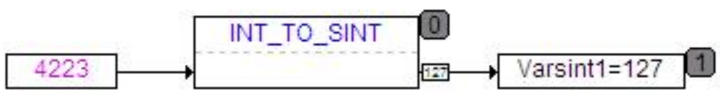
For INT_TO_TIME and INT _TO_TOD, the input will be converted in millisecond.

For INT_TO_DATE and INT _TO_DT, the input will be converted in second.

3.1.6.7.2 Application Example of Instructions

<table>
<tr><th colspan="2">Variable Definition</th></tr>
<tr><td colspan="2">

No.	Variable Name	Address	Variable Description	Variable Type	Initial Value	Power Fail Safeguard
0001	Varsint1			SINT	0	FALSE

</td></tr>
<tr><td>Programing language</td><td>Program</td></tr>
<tr><td>Ladder diagram (LD)</td><td>
</td></tr>
<tr><td>Structured text (ST)</td><td>VarSINT1 := INT_TO_SINT(4223); (*Result VarSINT1 为 127 *)
Note: If the integer 4223 (hexadecimal 16#107F) is saved as the SINT type variable, the high bit data will lose, and the low bit data 127 (hexadecimal 16#7F) will be displayed only.</td></tr>
<tr><td>Continuous Function Chart (CFC)</td><td>
</td></tr>
</table>

3.1.6.8 LINT Type Conversion

3.1.6.8.1 Function

Convert the type of long integer data to other types of data.

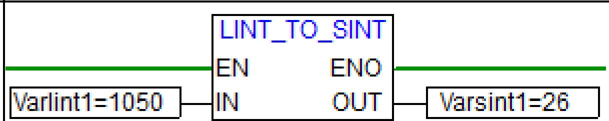
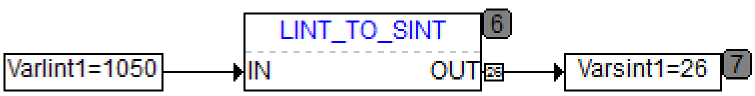
When LINT_TO_BOOL, Input is not equal to 0, the output is TRUE, when the input is equal to 0, the output is FALSE; When LINT_TO_TIME, LINT_TO_TOD, the input shall be converted by millisecond value.

When LINT_TO_DATE, LINT_TO_DT, the input shall be converted by second value.

3.1.6.8.2 Application Example of Instructions

Variable Definition

No.	Variable Name	Address	Variable Description	Variable Type	Initial Value	Power Fail Safeguard
0001	Varlint1			LINT	0	FALSE
0002	Varsint1			SINT	0	FALSE

<table>
<tr><td>Programming Language</td><td>Program</td></tr>
<tr><td>Ladder Diagram（LD）</td><td>
</td></tr>
<tr><td>Structured Text（ST）</td><td>Varsint1:=LINT_TO_SINT(Varlint1); (*The result of Varsint1 is 26*)
Description: if saving the integer 1050 (hexadecimal is 16#41A) as variable of SINT type, then the high data will be lost, and only the low data 26 (hexadecimal is 16#1A)will be showed.</td></tr>
<tr><td>Continuous Function Chart（CFC）</td><td>
</td></tr>
</table>

3.1.6.9 LREAL Type Conversion

3.1.6.9.1 Function

Convert the LREAL type into other data types.

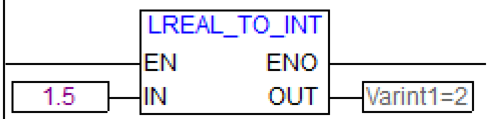
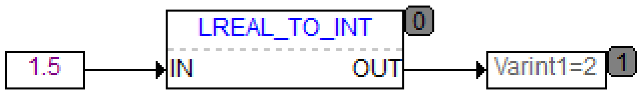
For LREAL_TO_BOOL, the output will be TRUE if the input is not equal to 0 and be FALSE if the input is equal to 0. For LREAL_TO_TIME and LREAL_TO_TOD, the input will be converted in millisecond.

For LREAL_TO_DATE and LREAL_TO_DT, the input will be converted in second.

For LREAL_TO_STRING, the maximum string length after the conversion is 16.

3.1.6.9.2 Application Example of Instructions

Variable Definition						
No.	Variable Name	Address	Variable Description	Variable Type	Initial Value	Power Fail Safeguard
0001	Varint1			INT	0	FALSE
0002	Varint2			INT	0	FALSE
0003	Varint3			INT	0	FALSE
0004	Varint4			INT	0	FALSE

Programing language	Program
Ladder diagram (LD)	
Structured text (ST)	VarINT1:= LREAL_TO_INT(1.5); (*The result VarINT1 is 2*) VarINT2:= LREAL_TO_INT(1.4); (*The result VarINT2 is 1*) VarINT3:= LREAL_TO_INT(-1.5); (*The result VarINT3 is -2*) VarINT4:= LREAL_TO_INT(-1.4); (*The result VarINT4 is -1*)
Continuous Function Chart (CFC)	

3.1.6.10 LREAL Type Conversion

3.1.6.10.1 Function

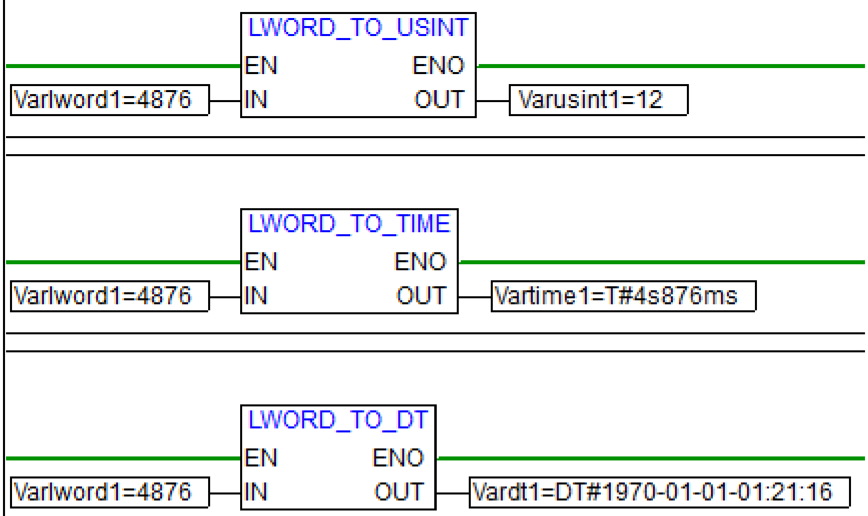
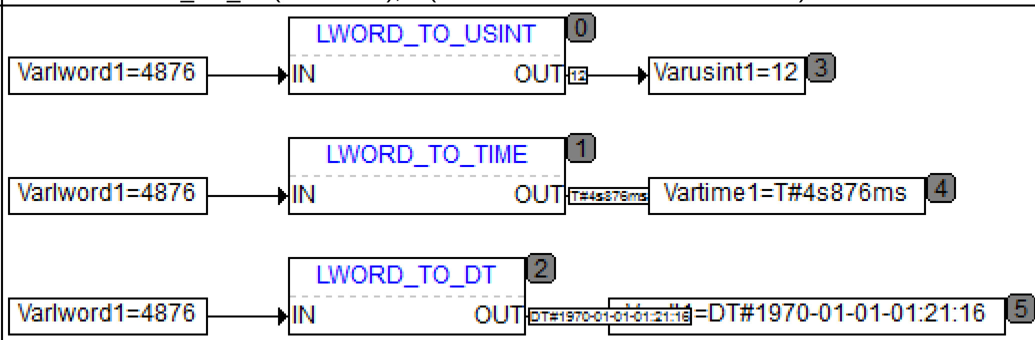
Convert the type of long word to other types of data.

When LWORD_TO_BOOL, Input is not equal to 0, the output is TRUE, when the input is equal to 0, the output is FALSE; When LWORD_TO_TIME, LWORD_TO_TOD, the input shall be transited by millisecond value.

When LWORD_TO_DATE, LWORD_TO_DT, the input shall be transited by second value.

3.1.6.10.2 Application Example of Instructions

Variable Definition						
No.	Variable Name	Address	Variable Description	Variable Type	Initial Value	Power Fail Safeguard
0001	Varlword1			LWORD	0	FALSE
0002	Varusint1			USINT	0	FALSE
0003	Vartime1			TIME	T#0MS	FALSE
0004	Vardt1			DT	DT#1970-01-01-00:00:00	FALSE

<table>
<tr><th colspan="2">Variable Definition</th></tr>
<tr><td>Programming language</td><td>Program</td></tr>
<tr><td>Ladder Diagram（LD）</td><td>
</td></tr>
<tr><td>Structured Text（ST）</td><td>Varlword1:=4876; (*Varlword1value*)
Varusint1:=LWORD_TO_USINT(Varlword1); (*Result 12*)
Description: if saving the integer 4876 (hexadecimal is 16#130C) as variable of USINT type, then the high data will be lost, and only the low data 12 (hexadecimal is 16#0C) will be showed.
Vartime1:=LWORD_TO_TIME(Varlword1); (*Result T#4s876ms*)
Vardt1:=LWORD_TO_DT(Varlword1); (*Result DT#1970-01-01-01:21:16*)</td></tr>
<tr><td>Continuous Function Chart（CFC）</td><td>
</td></tr>
</table>

3.1.6.11 REAL Type Conversion

3.1.6.11.1 Function

Convert the REAL type into other data types.

For REAL_TO_BOOL, the output will be TRUE if the input is not equal to 0 and be FALSE if the input is equal to 0. For REAL_TO_TIME and REAL_TO_TOD, the input will be converted in millisecond.

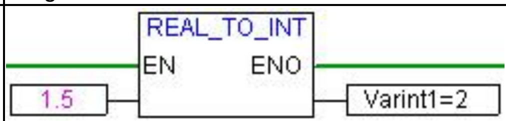
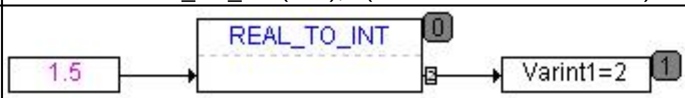
For REAL_TO_DATE and REAL_TO_DT, the input will be converted in second.

For LREAL_TO_STRING, the maximum string length after the conversion is 16.

3.1.6.11.2 Application Example of Instructions

Variable Definition

No.	Variable Name	Address	Variable Description	Variable Type	Initial Value	Power Fail Safeguard
0001	Varint1			INT	0	FALSE
0002	Varint2			INT	0	FALSE
0003	Varint3			INT	0	FALSE
0004	Varint4			INT	0	FALSE

Programing language	Program
Ladder diagram (LD)	
Structured text (ST)	VarINT1:= REAL_TO_INT(1.5); (*The result VarINT1 is 2*) VarINT2:= REAL_TO_INT(1.4); (*The result VarINT2 is 1*) VarINT3:= REAL_TO_INT(-1.5); (*The result VarINT3 is -2*) VarINT4:= REAL_TO_INT(-1.4); (*The result VarINT4 is -1*)
Continuous Function Chart (CFC)	

3.1.6.12 SINT Type Conversion

3.1.6.12.1 Function

Convert the SINT type into other data types.

For SINT_TO_BOOL, the output will be TRUE if the input is not equal to 0 and be FALSE if the input is equal to 0.

For SINT_TO_TIME and SINT _TO_TOD, the input will be converted in millisecond.

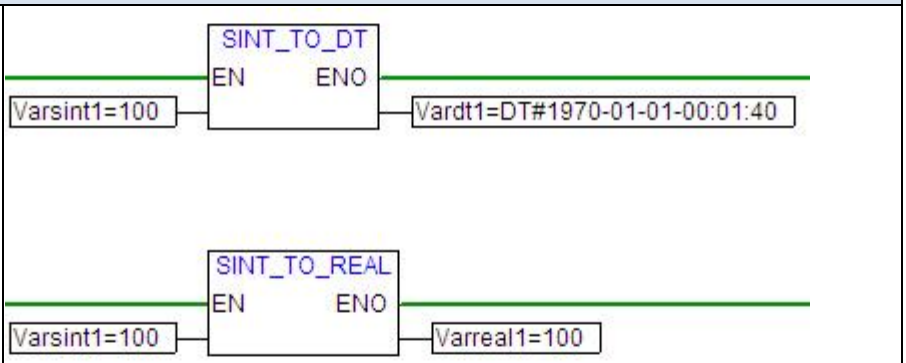
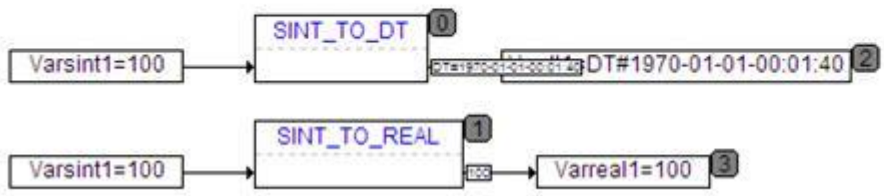
For SINT_TO_DATE and SINT _TO_DT, the input will be converted in second.

3.1.6.12.2 Application Example of Instructions

Variable Definition

No.	Variable Name	Address	Variable Description	Variable Type	Initial Value	Power Fail Safe
0001	Varsint1			SINT	0	FALSE
0002	Vardt1			DT	DT#1970-01-01...	FALSE
0003	Varreal1			REAL	0	FALSE

Programing language	Program

Variable Definition	
Ladder diagram (LD)	
Structured text (ST)	Varsint1:=100; (*Varsint1 value*) Vardt1:=SINT_TO_DT(Varsint1); (*Result DT#1970-01-01-00:01:40 *) Varreal1:=SINT_TO_REAL(Varsint1); (*The result is 100 .0*)
Continuous Function Chart (CFC)	

3.1.6.13 STRING Type Conversion

3.1.6.13.1 Function

Convert the STRING type into other data types.

FOR STRING_TO_BOOL instruction , the output will be TRUE if the input string is 'TRUE' or 'true'. Otherwise, the output will be FALSE.

3.1.6.13.2 Application Example of Instructions

Variable Definition

No.	Variable Name	Address	Variable Description	Variable Type	Initial Value	Power Fail Safegua
0001	b			TIME	T#0MS	FALSE
0002	c			BOOL	FALSE	FALSE
0003	d			DINT	0	FALSE

```
0001 b:=STRING_TO_TIME('T#127MS');
0002 c:= STRING_TO_BOOL ('TRUE');
0003 d:= STRING_TO_DINT ('124378');
0004
```

```
b = T#127ms
c = TRUE
d = 124378
```

Programing language	Program
Structured text (ST)	
Continuous Function Chart (CFC)	

3.1.6.14 TIME Type Conversion

3.1.6.14.1 Function

Convert the TIME type into other data types. The time is stored into the DWORD type in millisecond internally (start from 00:00 for the TIME_OF_DAY variable).

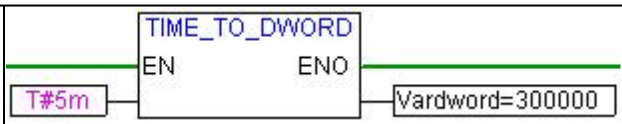
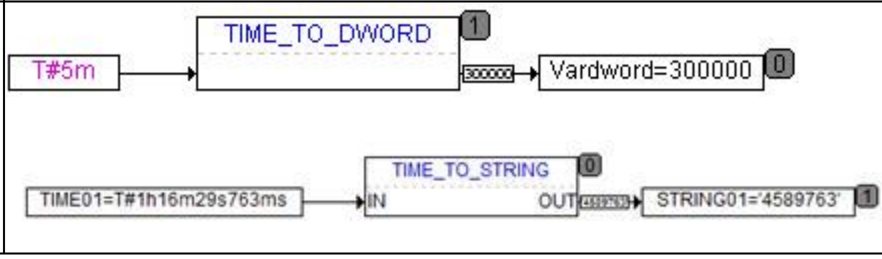
For TIME_TO_BOOL, the output will be TRUE if the input is not equal to 0 and be FALSE if the input is equal to 0.

3.1.6.14.2 Application Example of Instructions

Variable Definition

No.	Variable Name	Address	Variable Description	Variable Type	Initial Value	Power Fail Safegua
0001	Vardword			DWORD	0	FALSE

Programing language	Program

<table>
<tr><td>Ladder diagram (LD)</td><td>
</td></tr>
<tr><td>Structured text (ST)</td><td>Vardword:=TIME_TO_DWORD(T#5m);   (*The result is 300000*)</td></tr>
<tr><td>Continuous Function Chart (CFC)</td><td>
</td></tr>
</table>

3.1.6.15 TOD Type Conversion

3.1.6.15.1 Function

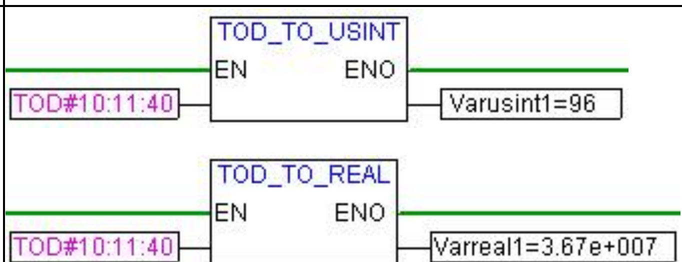
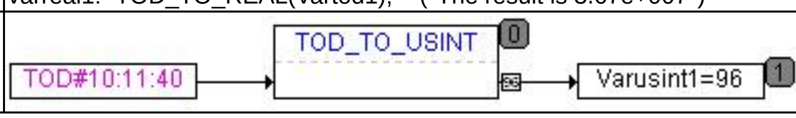
Convert the TOD type into other data types. The date is converted in millisecond internally.

For TOD_TO_BOOL, the output will be TRUE if the input is not equal to 0 and be FALSE if the input is equal to 0.

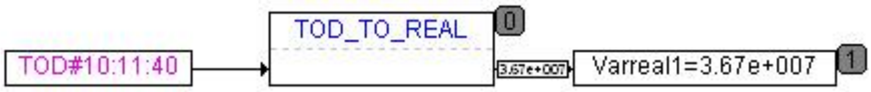
3.1.6.15.2 Application Example of Instructions

Variable Definition

No.	Variable Name	Address	Variable Description	Variable Type	Initial Value	Power Fail Safegu
0001	Vartod1			TOD	TOD#00:00:00	FALSE
0002	Varusint1			USINT	0	FALSE
0003	Vartime1			TIME	T#0MS	FALSE

<table>
<tr><td>Programing language</td><td>Program</td></tr>
<tr><td>Ladder diagram (LD)</td><td>
</td></tr>
<tr><td>Structured text (ST)</td><td>Vartod1:=TOD#10:11:40;   (*Vartod1 value*)
Varusint1:=TOD_TO_USINT(Vartod1);   (*The result is 96*)
Varreal1:=TOD_TO_REAL(Vartod1);   (*The result is 3.67e+007*)</td></tr>
<tr><td>Continuous Function Chart (CFC)</td><td>
</td></tr>
</table>

Variable Definition



3.1.6.16 UDINT Type Conversion

3.1.6.16.1 Function

Convert the UDINT type into other data types.

For UDINT_TO_BOOL, the output will be TRUE if the input is not equal to 0 and be FALSE if the input is equal to 0.

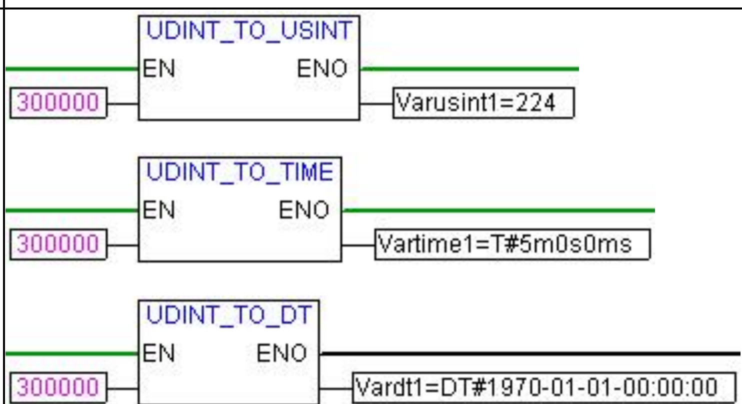
For UDINT_TO_TIME and UDINT _TO_TOD, the input will be converted in millisecond.

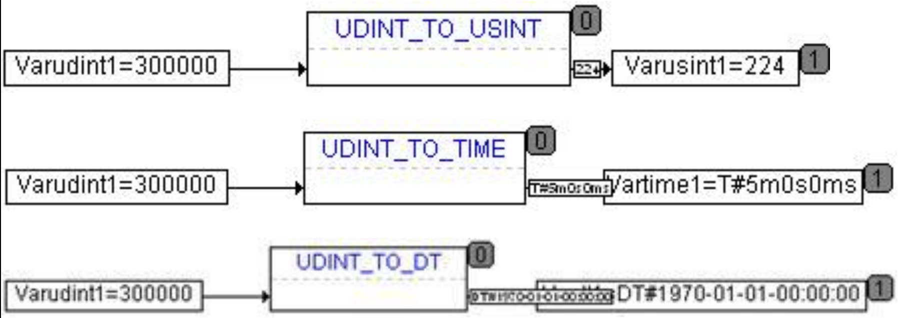
For UDINT_TO_DATE and UDINT _TO_DT, the input will be converted in second.

3.1.6.16.2 Application Example of Instructions

Variable Definition

No.	Variable Name	Address	Variable Description	Variable Type	Initial Value	Power Fail Safeguard
0001	Varudint1			UDINT	0	FALSE
0002	Varusint1			USINT	0	FALSE
0003	Vartime1			TIME	T#0MS	FALSE
0004	Vardt1			DT	DT#1970-01-01-00:00:00	FALSE

Programing language	Program
Ladder diagram (LD)	
Structured text (ST)	Varudint1:=300000; Varusint1:= UDINT_TO_USINT(Varudint1); (*Result 224*) Note: If the integer 300000 (hexadecimal 16#493E0) is saved as the USINT type variable, the high bit data will lose, and the low bit data 224 (hexadecimal 16#E0) will be displayed only. Vartime1:=UDINT_TO_TIME(Varudint1); (*Result T#5m0s0ms*) Vardt1:=UDINT_TO_DT(Varudint1); (*Result DT#1970-01-01-00:00:00 *)

Variable Definition	
Continuous Function Chart (CFC)	

3.1.6.17 UINT Type Conversion

3.1.6.17.1 Function

Convert the UINT type into other data types.

For UINT_TO_BOOL, the output will be TRUE if the input is not equal to 0 and be FALSE if the input is equal to 0.

For UINT_TO_TIME and UINT _TO_TOD, the input will be converted in millisecond.

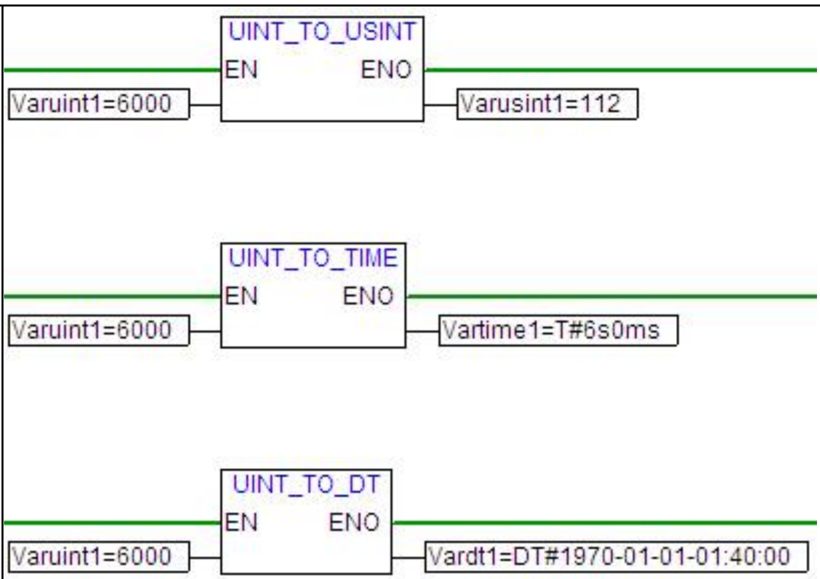
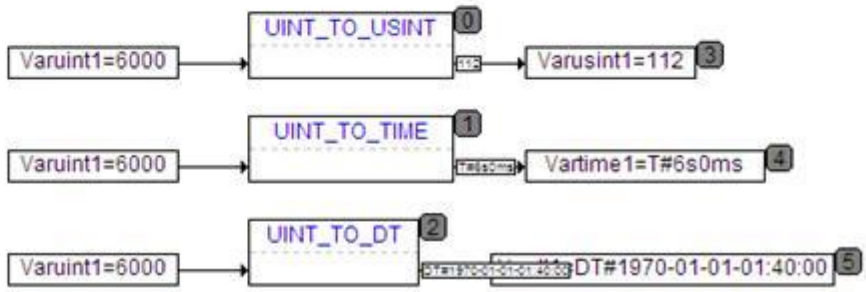
For UINT_TO_DATE and UINT _TO_DT, the input will be converted in second.

3.1.6.17.2 Application Example of Instructions

Variable Definition

No.	Variable Name	Address	Variable Description	Variable Type	Initial Value	Power Fail Safeguard
0001	Varuint1			UINT	6000	FALSE
0002	Varusint1			USINT	0	FALSE
0003	Vartime1			TIME	T#0MS	FALSE
0004	Vardt1			DT	DT#1970-01-01-00:00:00	FALSE

Programing language	Program

<table>
<tr><td>Ladder diagram (LD)</td><td>
</td></tr>
<tr><td>Structured text (ST)</td><td>Varuint1:=6000;
Varusint1:=UINT_TO_USINT(Varuint1);
Note: If the integer 6000 (hexadecimal 16#1770) is saved as the USINT type variable, the high bit data will lose, and the low bit data 112 (hexadecimal 16#70) will be displayed only.
Vartime1:=UINT_TO_TIME(Varuint1);
Vardt1:=UINT_TO_DT(Varuint1);</td></tr>
<tr><td>Continuous Function Chart (CFC)</td><td>
</td></tr>
</table>

3.1.6.18 ULINT Type Conversion

3.1.6.18.1 Function

Convert the type of unsigned long integer data to other types of data.

When ULINT_TO_BOOL, Input is not equal to 0, the output is TRUE, when the input is equal to 0, the output is FALSE.

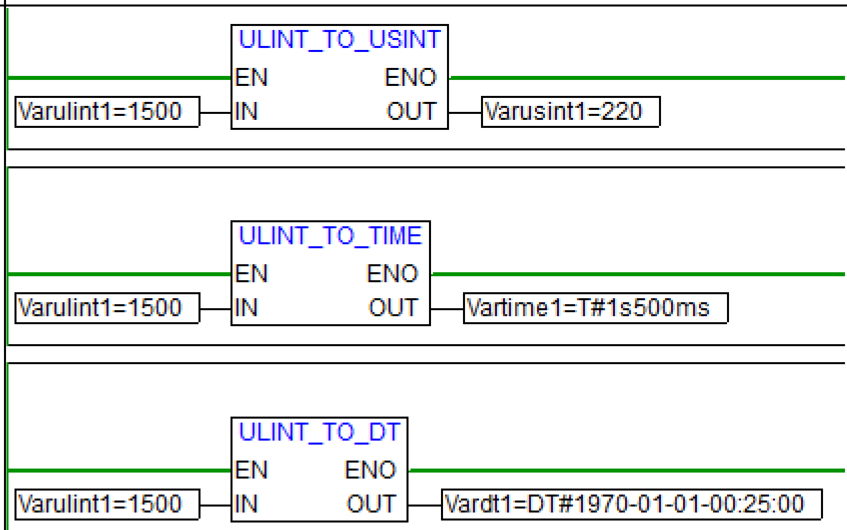
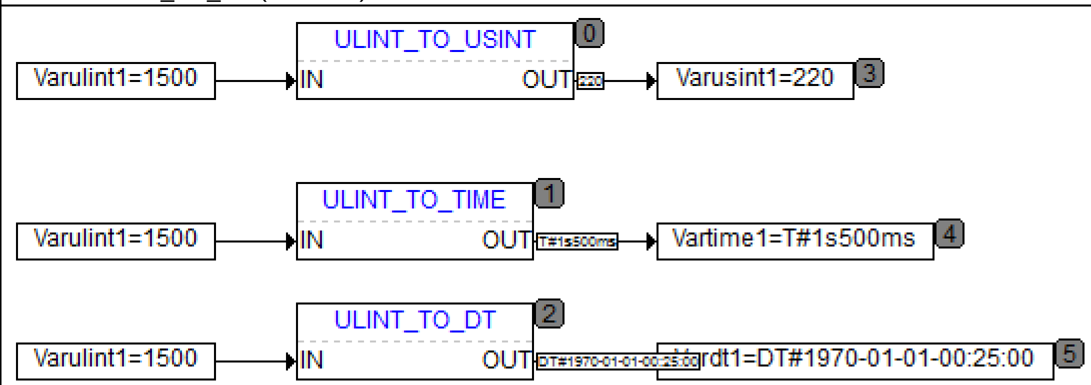
When ULINT_TO_TIME, ULINT_TO_TOD, the input shall be converted by millisecond value.

When ULINT_TO_DATE, ULINT_TO_DT, the input shall be converted by second value.

3.1.6.18.2 Application Example of Instructions

Variable Definition

No.	Variable Name	Address	Variable Description	Variable Type	Initial Value	Power Fail Safeguard
0001	Varulint1			ULINT	0	FALSE
0002	Varusint1			USINT	0	FALSE
0003	Vartime1			TIME	T#0MS	FALSE
0004	Vardt1			DT	DT#1970-01-01-00:00:00	FALSE

Programming language	Program
Ladder Diagram（LD）	
Structured Text（ST）	Varulint1:=1500. Varusint1:=ULINT_TO_USINT (Varulint1). Description: if saving the integer 1500 (hexadecimal is 16#5DC) as variable of USINT type, then the high data will be lost, and only the low data 220 (hexadecimal is 16#DC) will be showed. Vartime1:=ULINT_TO_TIME (Varulint1). Vardt1:=ULINT_TO_DT (Varulint1).
Continuous Function Chart（CFC）	

3.1.6.19 USINT Type Conversion

3.1.6.19.1 Function

Convert the USINT type into other data types.

For USINT_TO_BOOL, the output will be TRUE if the input is not equal to 0 and be FALSE if the input is equal to 0.

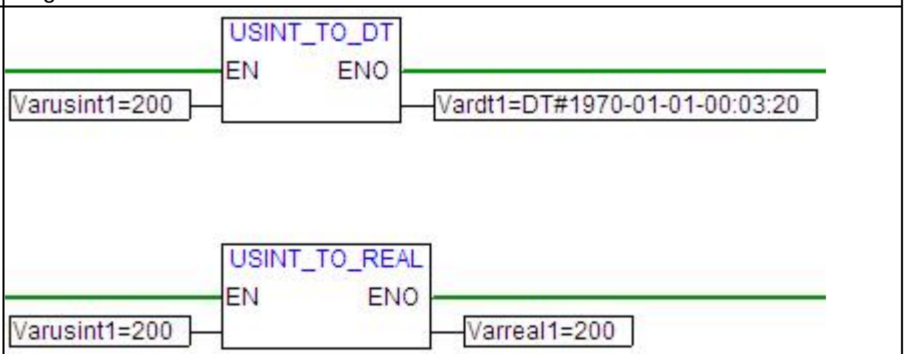
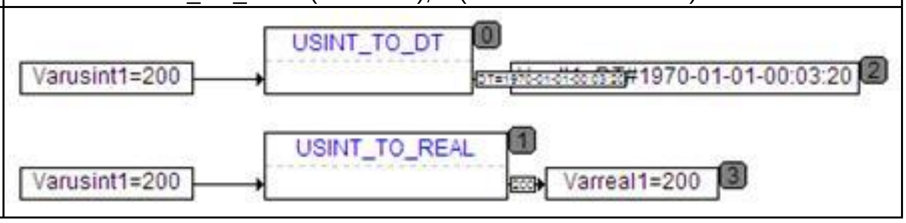
For USINT_TO_TIME and USINT _TO_TOD, the input will be converted in millisecond.

For USINT_TO_DATE and USINT _TO_DT, the input will be converted in second.

3.1.6.19.2 Application Example of Instructions

Variable Definition

No.	Variable Name	Address	Variable Description	Variable Type	Initial Value	Power Fail Safeguard
0001	Varusint1			USINT	0	FALSE
0002	Vardt1			DT	DT#1970-01-01-00:00:00	FALSE
0003	Varreal1			REAL	0	FALSE

Programing language	Program
Ladder diagram (LD)	
Structured text (ST)	Varusint1:=200; (*Varusint1 value*) Vardt1:=USINT_TO_DT(Varusint1); (*Result DT#1970-01-01-00:03:20 *) Varreal1:=USINT_TO_REAL(Varusint1); (*The result is 200 .0*)
Continuous Function Chart (CFC)	

3.1.6.20 WORD Type Conversion

3.1.6.20.1 Function

Convert the WORD type into other data types.

For WORD_TO_BOOL, the output will be TRUE if the input is not equal to 0 and be FALSE if the input is equal to 0.

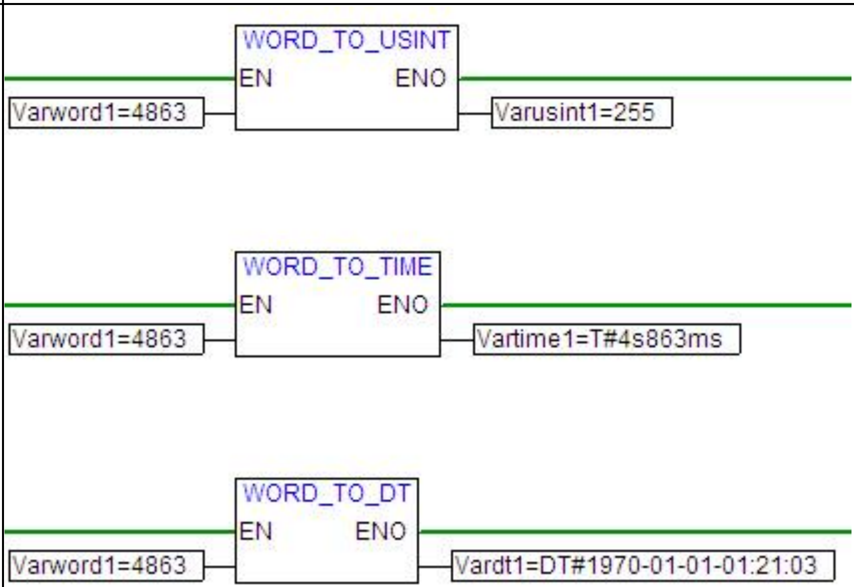
For WORD_TO_TIME and WORD_TO_TOD, the input will be converted in millisecond.

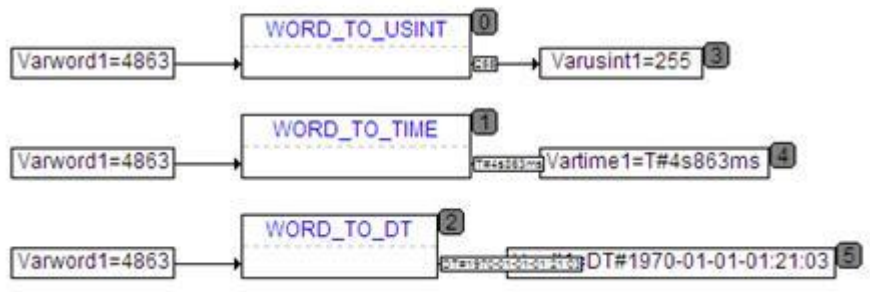
For WORD_TO_DATE and WORD_TO_DT, the input will be converted in second.

3.1.6.20.2 Application Example of Instructions

Variable Definition

No.	Variable Name	Address	Variable Description	Variable Type	Initial Value	Power Fail Safeguard
0001	Varusint1			USINT	0	FALSE
0002	Varword1			WORD	0	FALSE
0003	Vartime1			TIME	T#0MS	FALSE
0004	Vardt1			DT	DT#1970-01-...	FALSE

Programing language	Program
Ladder diagram (LD)	
Structured text (ST)	Varword1:=4863; (*Varword1 value*) Varusint1:=WORD_TO_USINT(Varword1); (*Result 255 *) Note: If the integer 4863 (hexadecimal 16#12FF) is saved as the USINT type variable, the high bit data will lose, and the low bit data 255 (hexadecimal 16#FF) will be displayed only. Vartime1:=WORD_TO_TIME(Varword1); (*Result T#4s863ms*) Vardt1:=WORD_TO_DT(Varword1); (*Result DT#1970-01-01-01:21:03 *)

Continuous Function Chart (CFC)	

3.1.6.21 TRUNC

3.1.6.21.1 Function

Get the digital integral of the floating-point number.

3.1.6.21.2 Input/output Data Type

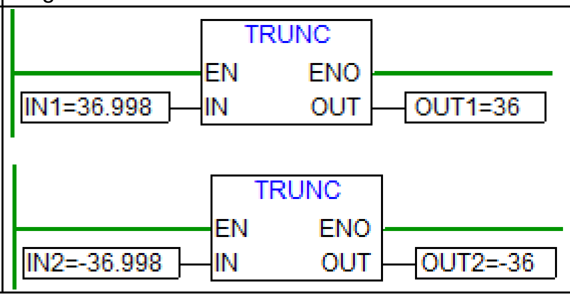
Input: REAL.

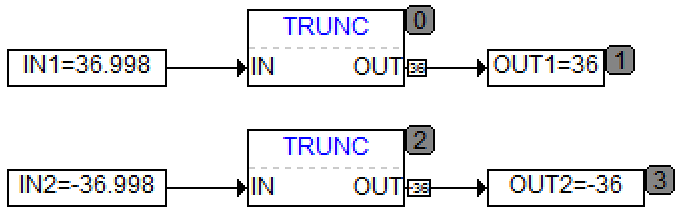
Output: DINT.

3.1.6.21.3 Application Example of Instructions

Variable Definition

No.	Variable Name	Address	Variable Description	Variable Type	Initial Value	Power Fail Safeguard
0001	IN1			REAL	0	FALSE
0002	OUT1			DINT	0	FALSE
0003	IN2			REAL	0	FALSE
0004	OUT2			DINT	0	FALSE

Programing language	Program
Ladder diagram (LD)	

Structured text (ST)	0001 OUT1:= TRUNC (36.998); 0002 OUT2:= TRUNC (-36.998); OUT1 = 36 OUT2 = -36
Continuous Function Chart (CFC)	

3.1.7 Pointer Operation

3.1.7.1 ADR (Get Address)

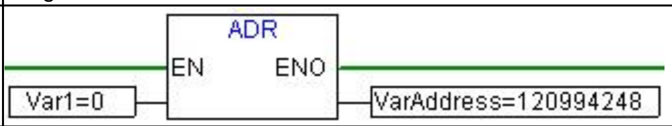
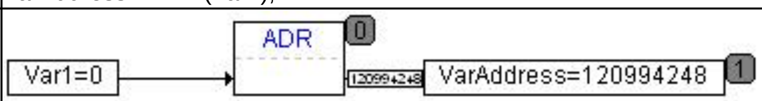
3.1.7.1.1 Function

Get the memory address of the input variable and output it. This address can be used as the pointer in the program or transferred to the function as the pointer.

3.1.7.1.2 Application Example of Instructions

Variable Definition

No.	Variable Name	Address	Variable Description	Variable Type	Initial Value	Power Fail Safeguard
0001	Var1			BYTE	0	FALSE
0002	VarAddress			POINTER TO BYTE	0	FALSE

Programing language	Program
Ladder diagram (LD)	
Structured text (ST)	VarAddress:= ADR(Var1);
Continuous Function Chart (CFC)	

- Please use ADR instruction carefully and sure that the pointer type matchs with the variable type.
- There is a risk with using this instruction, it is suggested to achieve via other ways.

3.1.7.2 VAL (Get Value)

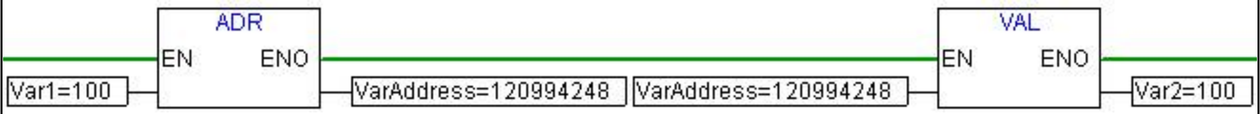
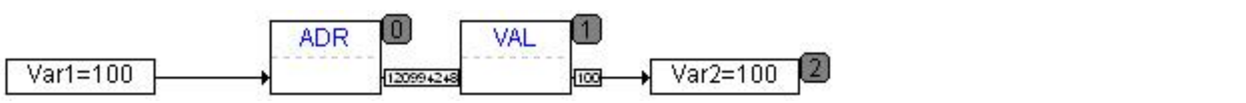
3.1.7.2.1 Function

Get the address data of the pointer.

3.1.7.2.2 Application Example of Instructions

Variable Definition

No.	Variable Name	Address	Variable Description	Variable Type	Initial Value	Power Fail Safeguard
0001	Var1			BYTE	0	FALSE
0002	Var2			BYTE	0	FALSE
0003	VarAddress			POINTER TO BYTE	0	FALSE

Programing language	Program
Ladder diagram (LD)	
Structured text (ST)	VarAddress:= ADR(Var1); Var2:= VAL(VarAddress); (*The result Var2 is 100 *)
Continuous Function Chart (CFC)	

ⓘThere is a risk with using this instruction, it is suggested to achieve via other ways.

3.1.8 Character String Operation

3.1.8.1 Left

3.1.8.1.1 Function Prototype

STRING LEFT (STRING str, INT count);

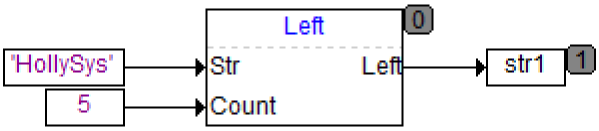
3.1.8.1.2 Function

Return the substring which is composed of the count characters in the leftmost of the specified string.

3.1.8.1.3 Parameter Description

	Data Item	Data Type	Parameter Description
Input end	Str	STRING	Input string, whose maximum length is 128
	Count	BYTE, WORD, DWORD, SINT, USINT, INT, UINT, DINT and UDINT	Length of substring to be returned
Output end	Left	STRING	Return substring (1) If str is empty, return empty. (2) If count is less than or equal to 0, return empty. (3) If the actual length of str is greater than 128, get the former 128 characters. (4) If count is greater than or equal to the length of str, return str.

3.1.8.1.4 Configuration Example



Configuration Example

The value of Str1 is ‘Holly’.

3.1.8.2 Right

3.1.8.2.1 Function Prototype

STRING RIGHT (STRING str, INT count);

3.1.8.2.2 Function

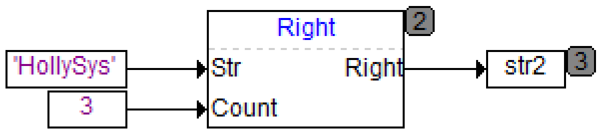
Return the substring which is composed of the count characters in the rightmost of the specified string.

3.1.8.2.3 Parameter Description

	Data Item	Data Type	Parameter Description
Input end	Str	STRING	Input string, whose maximum length is 128

	Count	BYTE, WORD, DWORD, SINT, USINT, INT, UINT, DINT and UDINT	Length of substring to be returned
Output end	Right	STRING	Return substring (1) If str is empty, return empty. (2) If count is less than or equal to 0, return empty. (3) If the actual length of str is greater than 128, get the former 128 characters. (4) If count is greater than or equal to the length of str, return str.

3.1.8.2.4 Configuration Example



Configuration Example

The value of Str2 is 'sys'.

3.1.8.3 Concat

3.1.8.3.1 Function Prototype

STRING CONCAT (STRING str1, STRING str2);

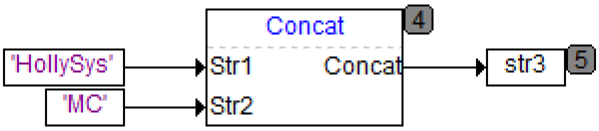
3.1.8.3.2 Function

Connect with two strings.

3.1.8.3.3 Parameter Description

	Data Item	Data Type	Parameter Description
Input end	Str1	STRING	Input string, whose maximum length is 128
	Str2	STRING	Input string, whose maximum length is 128
Output end	Concat	STRING	Return substring (1) If the length of the input string variable is greater than 128, get the former 128 characters. (2) If the length of str1 plus the length of str2 is greater than 128, return empty.

3.1.8.3.4 Configuration Example



Configuration Example

The value of str3 is ‘HollySysMC’.

3.1.8.4 Delete

3.1.8.4.1 Function Prototype

STRING DELETE (STRING str, INT len, INT pos);

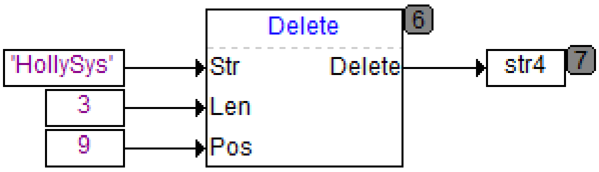
3.1.8.4.2 Function

Delete len characters from the pos character of str, and return the result.

3.1.8.4.3 Parameter Description

	Data Item	Data Type	Parameter Description
Input end	Str	STRING	Input string, whose maximum length is 128
	Len	BYTE, WORD, SINT, USINT, INT, UINT	Number of characters to be deleted
	Pos	BYTE, WORD, SINT, USINT, INT, UINT	Start positoin of strings to be deleted in the str string.
Output end	Delete	STRING	(1) If the actual length of str is greater than 128, get the former 128 characters. (2) If len is less than 0, then pos=pos+len, len=-len. (3) If pos is less than or equal to 0, then len=len+pos-1, pos=1. (4) If pos+len is greater than the length of str, then len = Length of str – pos + 1. (5) If pos is greater than the length of str or len is less than or equal to 0, return str (if its length is greater than 128, return the former 128 characters).

3.1.8.4.4 Configuration Example



Configuration Example

The value of str4 is ‘HollySys'.

3.1.8.5 Insert

3.1.8.5.1 Function Prototype

STRING INSERT (STRING str1, STRING str2, INT pos);

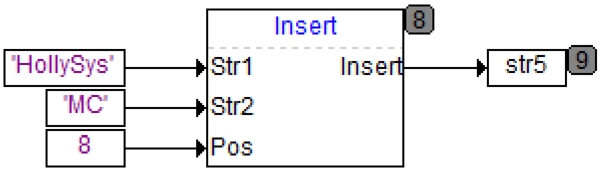
3.1.8.5.2 Function

Insert str2 after the pos character of str1, and return the result.

3.1.8.5.3 Parameter Description

	Data Item	Data Type	Parameter Description
Input end	Str1	STRING	Input string, whose maximum length is 128
	Str2	STRING	Input string, whose maximum length is 128
	Pos	BYTE, WORD, SINT, USINT, INT, UINT	Start position of string str2 to be inserted in string str1
Output end	Insert	STRING	(1) If the length of the input string variable is greater than 128, get the former 128 characters. (2) If the length of str1 plus the length of str2 is greater than 128, then return empty. (3) If pos=0, insert it before the string str1. (4) If pos=1, insert it after the first character. (5) If pos is greater than the length of str1 or less than 0, then return str1(if its length is greater than 128, return the former 128 characters).

3.1.8.5.4 Configuration Example



Configuration Example

The value of str5 is ‘HollySysMC’.

3.1.8.6 Mid

3.1.8.6.1 Function Prototype

STRING MID (STRING str, INT len, INT pos);

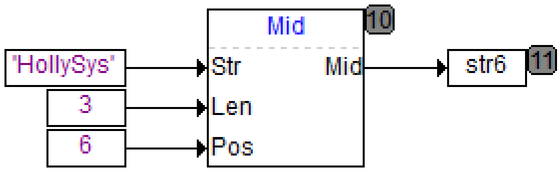
3.1.8.6.2 Function

Return the substring which is composed of continuous len characters starting with the pos character in the string.

3.1.8.6.3 Parameter Description

	Data Item	Data Type	Parameter Description
Input end	Str	STRING	Input string, whose maximum length is 128
	Len	BYTE, WORD, SINT, USINT, INT, UINT	Length of substring to be returned
	Pos	BYTE, WORD, SINT, USINT, INT, UINT	Start position of substring to be returned
Output end	Mid	STRING	Return substring (1) If str is empty, then return empty. (2) If the actual length of str is greater than 128, get the former 128 characters. (3) If pos is less than or equal to 0, then return empty. (4) If pos is greater than the length of str, then return empty. (5) If len is less than or equal to 0, then return empty. (6) If len is greater than the actual desirable characters, then return all characters after pos.

3.1.8.6.4 Configuration Example



Configuration Example

The value of str5 is ‘ys’.

3.1.8.7 Replace

3.1.8.7.1 Function Prototype

STRING REPLACE (STRING str1, STRING str2, INT len, INT pos);

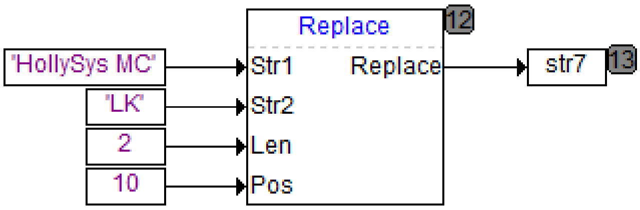
3.1.8.7.2 Function

Replace len characters starting with pos character of str1 with str2, and return the result.

3.1.8.7.3 Parameter Description

	Data Item	Data Type	Parameter Description
Input end	Str1	STRING	Input string, whose maximum length is 128
	Str2	STRING	Input string, whose maximum length is 128
	Len	BYTE, WORD, SINT, USINT, INT, UINT	Number of characters to be replaced
	Pos	BYTE, WORD, SINT, USINT, INT, UINT	Start position of characters to be replaced in string str1
Output end	Replace	STRING	(1) When the length of the input string variabel is greater than 128, get the former 128 characters. (2) If len is less than 0, then pos=pos+len, len=-len. (3) If pos is less than or equal to 0, then len=len+pos-1, pos=1. (4) If pos+len is greater than the length of str1, then len= Length of str1 - pos+1. (5) If len is greater than or equal to 0, then len characters starting with pos character of str1 will be replaced with str2. (6) If the length of str1 is greater than 128 after the replacement, intercept the former 128 characters and return them.

3.1.8.7.4 Configuration Example



Configuration Example

The value of str5 is ‘ollySys LK’.

3.1.8.8 Find

3.1.8.8.1 Function Prototype

INT FIND (STRING str1, STRING str2);

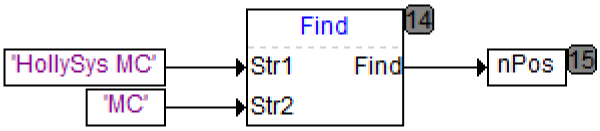
3.1.8.8.2 Function

Locate the initial character of str2 in str1 for the first time and return it.

3.1.8.8.3 Parameter Description

	Data Item	Data Type	Parameter Description
Input end	Str1	STRING	Input string, whose maximum length is 128
	Str2	STRING	Input string, whose maximum length is 128
Output end	Find	WORD, DWORD, INT, UINT, DINT, UDINT, LINT, ULINT, LWORD	(1) If str2 is not displayed in str1, then return 0. (2) If one of str1 and str2 is empty, then return 0. (3) If the length of the input string variable is greater than 128, get the former 128 characters.

3.1.8.8.4 Configuration Example



Configuration Example

The value of nPos is 10.

3.1.8.9 Len

3.1.8.9.1 Function Prototype

INT LEN (STRING str);

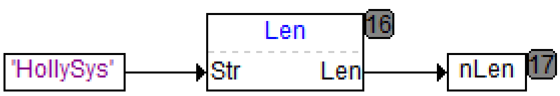
3.1.8.9.2 Function

Return the length of the input string.

3.1.8.9.3 Parameter Description

	Data Item	Data Type	Parameter Description
Input end	Str	STRING	Input string, whose maximum length is 128
Output end	Len	WORD, DWORD, INT, UINT, DINT, UDINT, LINT, ULINT, LWORD	(1) If the actual length of str is greater than 128, then get the former 128 characters. (2) Return the length of string, excluding the end of string. (3) If the input string is empty, then return 0.

3.1.8.9.4 Configuration Example



Configuration Example

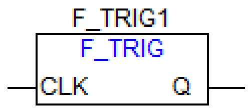
The value of nLen is 8.

3.1.9 Trigger Functions

3.1.9.1 F_TRIG (Falling Edge Detection Trigger)

3.1.9.1.1 Function

It is used to detect the falling edge.



F_TRIG Function Block

3.1.9.1.2 Logical Relation

Q := NOT CLK AND NOT M;

M := NOT CLK;

M is an intermediate variable with an initial value of TRUE, as long as CLK is TRUE, Q and M hold FALSE. CLK is FALSE, Q returns TRUE for the first time, and M is set to TRUE. Q returns FALSE when each time the instruction is called, and Q is TRUE when CLK detects a falling edge.

3.1.9.1.3 Parameter Description

Input Parameter Description

Input Parameter	Data Type	Function Description	Parameter Description	Default Value
CLK	BOOL	Input	Falling edge trigger signal	TRUE

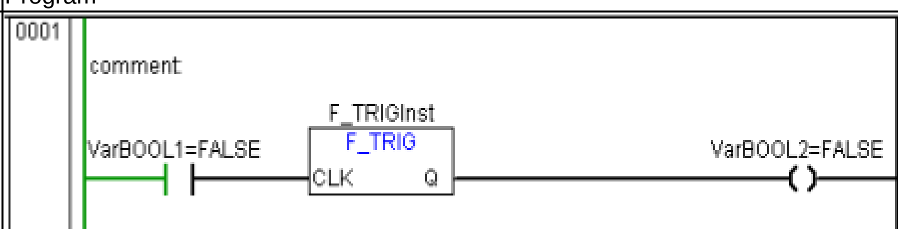
Output Parameter Description

output Parameter	Data Type	Function Description	Parameter Description	Default Value
Q	BOOL	Output	When the falling edge is detected, Q=TRUE. Once the instruction is called, Q=FALSE.	0

3.1.9.1.4 Application Example of Instructions

Variable Definition

No.	Variable Name	Address	Variable Description	Variable Type	Initial Value	Power Fail Safegua
0001	F_TRIGInst			F_TRIG		FALSE
0002	VarBOOL1			BOOL	FALSE	FALSE
0003	VarBOOL2			BOOL	FALSE	FALSE

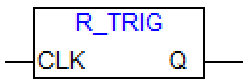
Programing language	Program
Ladder diagram (LD)	
Structured text (ST)	FTRIGInst(CLK:=VarBOOL1); VarBOOL2:=FTRIGInst.Q;

Continuous Function Chart (CFC)	

3.1.9.1.5 Function

3.1.9.2 R_TRIG (Rising Edge Detection Trigger)

It is used to detect the rising edge.



R_TRIG Function Block

3.1.9.2.1 Logical Relation

Q := CLK AND NOT M;

M := CLK;

M is an intermediate variable with an initial value of FALSE, as long as CLK is FALSE, Q and M are FALSE. Q returns FALSE when each time an instruction is called. Q returns TRUE when CLK detects a rising edge.

3.1.9.2.2 Parameter Description

3.1.9.2.2.1 Input Parameter Description

Input Parameter	Data Type	Function Description	Parameter Description	Default Value
CLK	BOOL	Input	Rising edge trigger signal	FALSE

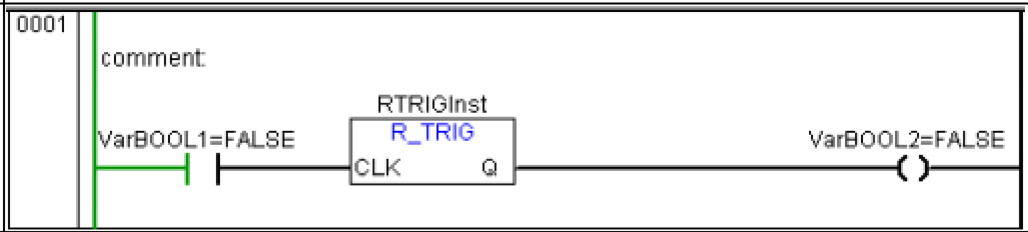
3.1.9.2.2.2 Output Parameter Description

output Parameter	Data Type	Function Description	Parameter Description	Default Value
Q	BOOL	Output	When the rising edge is detected, Q=TRUE. Once the instruction is called, Q=TURE.	0

3.1.9.2.3 Application Example of Instructions

Variable Definition

No.	Variable Name	Address	Variable Description	Variable Type	Initial Value	Power Fail Safegua
0001	R_TRIGInst			F_TRIG		FALSE
0002	VarBOOL1			BOOL	FALSE	FALSE
0003	VarBOOL2			BOOL	FALSE	FALSE

Programing language	Program
Ladder diagram (LD)	
Structured text (ST)	RTRIGInst(CLK:=VarBOOL1); VarBOOL2:=RTRIGInst.Q;
Continuous Function Chart (CFC)	

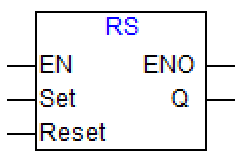
3.1.10 Bistable Functions

3.1.10.1 RS (Reset Priority Bistable)

3.1.10.1.1 Function

The reset bistable function algorithm (RS) performs the RS trigger function with the reset priority, and the SET and RESET at the input terminal and Q at the output terminal are of the BOOL type variable.

The detailed function is described as follows: Q1= RS (SET, RESET) indicates Q1= NOT RESET AND(Q1 OR SET).



RS Function Block

3.1.10.1.2 Parameter Description

Input Parameter Description

Input Parameter	Data Type	Function Description	Default Value	Power Fail Safeguard
Set	BOOL	Set	FALSE	FALSE
Reset	BOOL	Reset	FALSE	FALSE

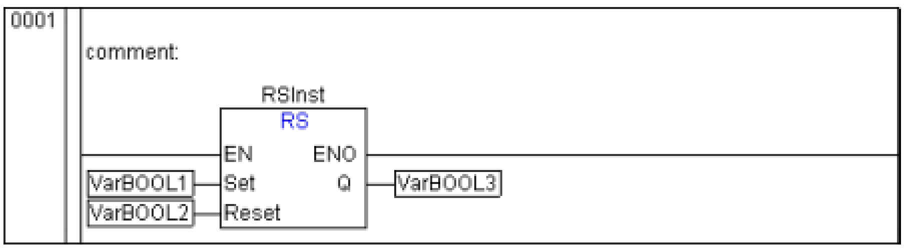
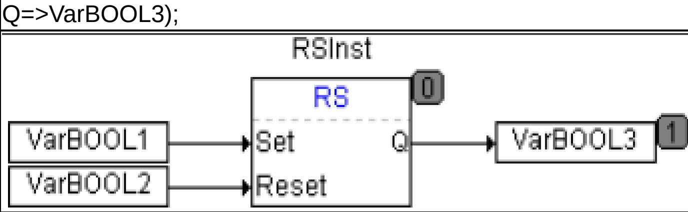
Output Parameter Description

Output Parameter	Data Type	Function Description	Default Value	Power Fail Safeguard
Q	BOOL	Output terminal	FALSE	TRUE

3.1.10.1.3 Application Example of Instructions

Variable Definition

No.	Variable Name	Address	Variable Description	Variable Type	Initial Value	Power Fail S
0001	RSInst			RS		FALSE
0002	VarBOOL1			BOOL	FALSE	FALSE
0003	VarBOOL2			BOOL	FALSE	FALSE
0004	VarBOOL3			BOOL	FALSE	FALSE

Programing language	Program
Ladder diagram (LD)	 Note: VarBOOL3＝NOT VarBOOL2 AND (VarBOOL3 OR VarBOOL1)
Structured text (ST)	RSInst(Set:=VarBOOL1, Reset:=VarBOOL2, Q=>VarBOOL3);
Continuous Function Chart (CFC)	

The truth table of the instruction is as shown in table.

Truth Table

Reset	Set	Q(n+1)
0	0	Q(n)
0	1	1

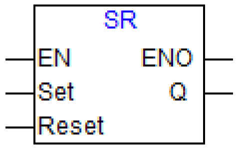
1	0	0
1	1	0

3.1.10.2 SR (Set Priority Bistable)

3.1.10.2.1 Function

The bistable function algorithm (SR) performs the RS trigger function with the set priority, and the SET and RESET at the input terminal and Q at the output terminal are of the BOOL type variable.

The detailed function is as follows: Q1 = SR (SET, RESET) indicates Q1 = (NOT RESET AND Q1) OR SET.



SR Function Block

3.1.10.2.2 Parameter Description

Input Parameter Description

Input Parameter	Data Type	Function Description	Default Value	Power Fail Safeguard
Set	BOOL	Set	FALSE	FALSE
Reset	BOOL	Reset	FALSE	FALSE

Output Parameter Description

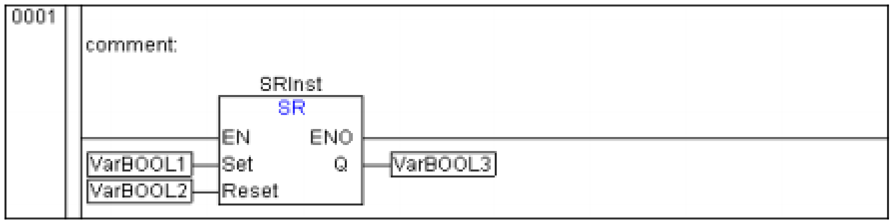
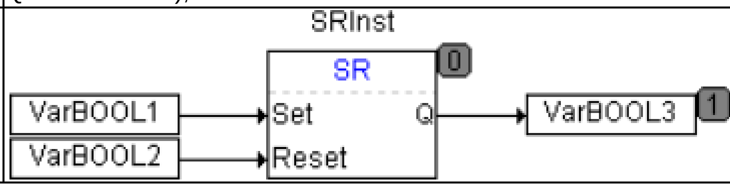
Output Parameter	Data Type	Function Description	Default Value	Power Fail Safeguard
Q	BOOL	Output terminal	FALSE	TRUE

3.1.10.2.3 Application Example of Instructions

Variable Definition

No.	Variable Name	Address	Variable Description	Variable Type	Initial Value	Power Fail Safegua
0001	SRInst			SR		FALSE
0002	VarBOOL1			BOOL	FALSE	FALSE
0003	VarBOOL2			BOOL	FALSE	FALSE
0004	VarBOOL3			BOOL	FALSE	FALSE

Programing language	Program

Ladder diagram (LD)	 Note: VarBOOL3＝(NOT VarBOOL2 AND VarBOOL3)OR VarBOOL1
Structured text (ST)	SRInst(Set:= VarBOOL1, Reset:= VarBOOL2, Q=> VarBOOL3);
Continuous Function Chart (CFC)	

The truth table of the instruction is as shown in table.

Truth Table

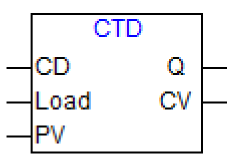
Reset	Set	Q(n+1)
0	0	Q(n)
0	1	1
1	0	0
1	1	1

3.1.11 Counter Functions

3.1.11.1 CTD (Decrement Counter)

3.1.11.1.1 Function

The decrement counter algorithm (CTD) will perform the counting decrement function when there is the rising edge for the input CD.



CTD Function Block

3.1.11.1.2 Parameter Description

Input Parameter Description

Input Parameter	Data Type	Function Description	Initial Value (Default)	Power Fail Safeguard
CD	BOOL	Rising edge detection terminal	FALSE	FALSE
Load	BOOL	The count variable CV will be initialized as PV when LOAD is TRUE.	FALSE	FALSE
PV	WORD	Set value	0	TRUE

Output Parameter Description

Output Parameter	Data Type	Function Description	Initial Value (Default)	Power Fail Safeguard
Q	BOOL	Counting flag output Q will output TRUE when CV is less than or equal to 0.	TRUE	FALSE
CV	WORD	Current count value	0	TRUE

Parameter Description of Intermediate Variable

Intermediate Parameter	Data Type	Function Description	Initial Value (Default)	Power Fail Safeguard
M	BOOL	CD detection variable	FALSE	TRUE

3.1.11.1.3 Detailed Function Description

The decrement counter algorithm (CTD) will perform the counting decrement function when there is the rising edge for the input CD. Input variable CD, Load and output variable Q are BOOL type, input variable PV and output variable CV are WORD type, described in detail as follows:

- Initialization Function

  When the initial value instruction Load is TRUE, the count variable CV will be initialized as the upper limit PV.

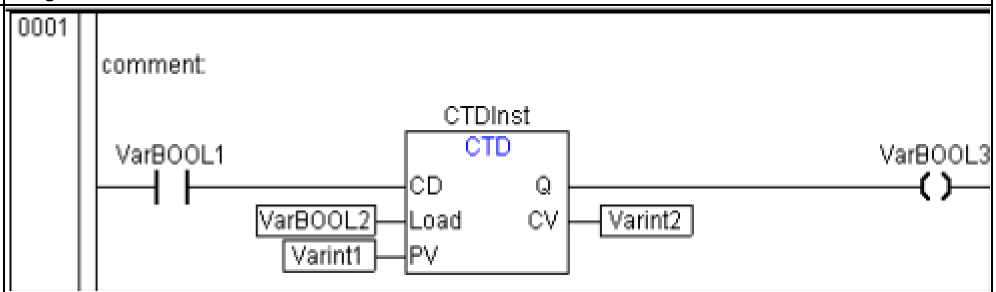
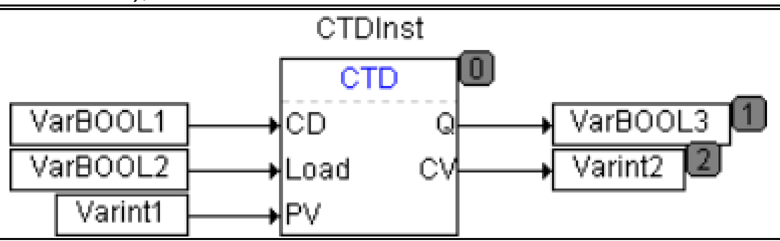
- Count Down Function

  If there is the rising edge from FALSE to TRUE for CD, we assume it is greater than 0, and the count output CV will be reduced by 1 (namely, the CV value is not less than 0). When CV is less than or equal to 0, the output Q will be set.

3.1.11.1.4 Application Example of Instructions

Variable Definition

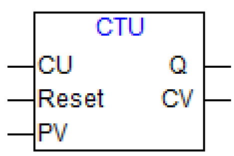
No.	Variable Name	Address	Variable Description	Variable Type	Initial Value	Power Fail Safegua
0001	CTDInst			CTD		FALSE
0002	VarBOOL1			BOOL	FALSE	FALSE
0003	VarBOOL2			BOOL	FALSE	FALSE
0004	Varint1			INT	0	FALSE
0005	VarBOOL3			BOOL	FALSE	FALSE
0006	Varint2			INT	0	FALSE

<table>
<tr><td>Programing language</td><td>Program</td></tr>
<tr><td>Ladder diagram (LD)</td><td>
</td></tr>
<tr><td>Structured text (ST)</td><td>CTDInst(
CD:=VarBOOL1,
Load:=VarBOOL2,
PV:=Varint1,
Q=>VarBOOL3,
CV=>Varint2);</td></tr>
<tr><td>Continuous Function Chart (CFC)</td><td>
</td></tr>
</table>

3.1.11.2 CTU (Increment Counter)

3.1.11.2.1 Function

The count algorithm (CTU) will perform the count function when there is the rising edge for the input CU.



CTU Function Block

3.1.11.2.2 Parameter Description

Input Parameter Description

Input Parameter	Data Type	Function Description	Initial Value (Default)	Power Fail Safeguard
CU	BOOL	Rising edge detection terminal	FALSE	FALSE
Reset	BOOL	The count variable CV will be initialized as 0 when Reset is TRUE.	FALSE	FALSE
PV	WORD	Set value	0	TRUE

Output Parameter Description

Output Parameter	Data Type	Function Description	Initial Value (Default)	Power Fail Safeguard
Q	BOOL	Q will output TRUE when the CV value is greater than or equal to PV.	TRUE	FALSE
CV	WORD	Count output	0	TRUE

Parameter Description of Intermediate Variable

Intermediate Parameter	Data Type	Function Description	Initial Value (Default)	Power Fail Safeguard
M	BOOL	CU detection variable	FALSE	TRUE

3.1.11.2.3 Detailed Function Description

The count algorithm (CTU) will perform the count function when there is the rising edge for the input CU. Input variable CU, Reset and output variable Q are BOOL type, input variable PV and output variable CV are WORD type, described in detail as follows:

- Initialization Function

  The count output CV will be initialized as 0 when Reset is TRUE.

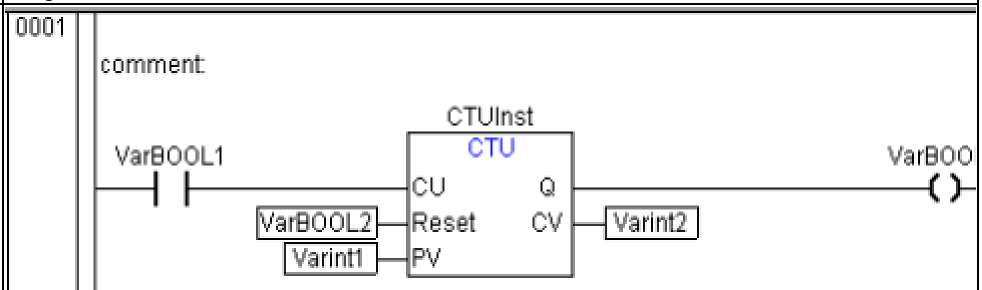
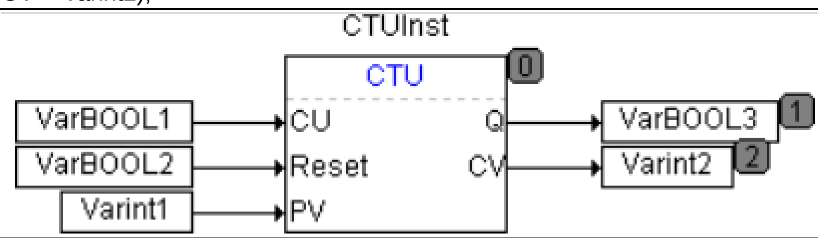
- Count Function

  The count output CV will be increased by 1 until CV is greater than or equal to the set value PV if there is the rising edge from FALSE to TRUE for CU, and the output Q will return TRUE.

3.1.11.2.4 Application Example of Instructions

Variable Definition

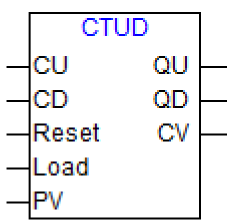
No.	Variable Name	Address	Variable Description	Variable Type	Initial Value	Power Fail Safegua
0001	CTUInst			CTU		FALSE
0002	VarBOOL1			BOOL	FALSE	FALSE
0003	VarBOOL2			BOOL	FALSE	FALSE
0004	Varint1			INT	0	FALSE
0005	VarBOOL3			BOOL	FALSE	FALSE
0006	Varint2			INT	0	FALSE

<table>
<tr><td>Programing language</td><td>Program</td></tr>
<tr><td>Ladder diagram (LD)</td><td>
</td></tr>
<tr><td>Structured text (ST)</td><td>CTUInst(
CU:=VarBOOL1,
Reset:=VarBOOL2,
PV:=Varint1,
Q=>VarBOOL3,
CV=>Varint2);</td></tr>
<tr><td>Continuous Function Chart (CFC)</td><td>
</td></tr>
</table>

3.1.11.3 CTUD (Increment/decrement Counter)

3.1.11.4 Function

Increment/decrement algorithm.



CTUD Function Block

3.1.11.5 Parameter Description

Input Parameter Description

Input Parameter	Data Type	Function Description	Initial Value (Default)	Power Fail Safeguard
CU	BOOL	Rising edge detection terminal	FALSE	FALSE
CD	BOOL	Rising edge detection terminal	FALSE	FALSE
Reset	BOOL	The count variable CV will be initialized as 0 when Reset is TRUE.	FALSE	FALSE
Load	BOOL	The count variable CV will be initialized as PV when Load is TRUE. However, CV=0 when Reset is also TRUE.	FALSE	FALSE
PV	WORD	Set value	0	TRUE

Output Parameter Description

Output Parameter	Data Type	Function Description	Initial Value (Default)	Power Fail Safeguard
QU	BOOL	QU will output TRUE when CV is equal to PV.	TRUE	FALSE
QD	BOOL	QD will output TRUE when CV is equal to 0.	TRUE	FALSE
CV	WORD	Count output	0	TRUE

Parameter Description of Intermediate Variable

Intermediate Parameter	Data Type	Function Description	Initial Value (Default)	Power Fail Safeguard
MU	BOOL	CU detection variable	FALSE	TRUE
MD	BOOL	CD detection variable	FALSE	TRUE

3.1.11.6 Detailed Function Description

Increment/decrement algorithm. Input variable CU, CD, Reset, Load and output variable QU, QD are BOOL type, input variable PV and output variable CV are WORD type, described in detail as follows:

- Initialization Function

  The count output CV will be initialized as 0 when RESET is 1. The count output CV will be initialized as PV when LOAD is 1.

- Count Function

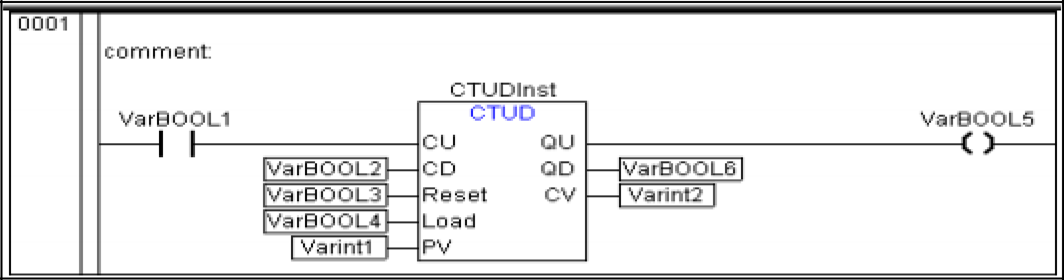
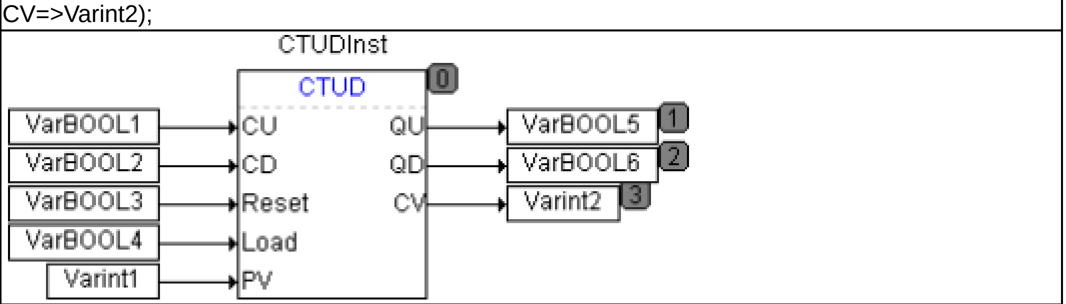
  CV will be increased by 1 if there is the rising edge from FALSE to TRUE for CU. CV will be decreased by 1 if there is the rising edge from FALSE to TRUE for CD (we assume the CV value will not be less than 0).

  - QU will return TRUE when CV is greater than or equal to PV.
  - QD will return TRUE when CV is equal to 0.

3.1.11.7 Application Example of Instructions

Variable Definition

No.	Variable Name	Address	Variable Description	Variable Type	Initial Value	Power Fail Safegua
0001	CTUDInst			CTUD		FALSE
0002	VarBOOL1			BOOL	FALSE	FALSE
0003	VarBOOL2			BOOL	FALSE	FALSE
0004	VarBOOL3			BOOL	FALSE	FALSE
0005	VarBOOL4			BOOL	FALSE	FALSE
0006	VarBOOL5			BOOL	FALSE	FALSE
0007	VarBOOL6			BOOL	FALSE	FALSE
0008	Varint1			INT	0	FALSE
0009	Varint2			INT	0	FALSE

<table>
<tr><td>Programing language</td><td>Program</td></tr>
<tr><td>Ladder diagram (LD)</td><td>
</td></tr>
<tr><td>Structured text (ST)</td><td>CTUDInst(
CU:=VarBOOL1,
CD:=VarBOOL2,
Reset:=VarBOOL3,
Load:=VarBOOL4,
PV:=Varint1,
QU=>VarBOOL5,
QD=>VarBOOL6,
CV=>Varint2);</td></tr>
<tr><td>Continuous Function Chart (CFC)</td><td>
</td></tr>
</table>

3.1.12 Timer Functions

The timer includes the Normal Timer, On Delay Timer, Off Delay Timer, Retentive On Delay Timer, which are described in follows.

SEE ALSO

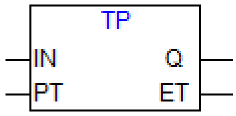
[TP (Normal Timer)]
[TON (On Delay Timer)]
[TOF (Off Delay Timer)]

TONR (Retentive On Delay Timer)

3.1.12.1 TP (Normal Timer)

3.1.12.1.1 Function

When IN changes into TRUE from FALSE, Q will be TRUE, and ET will start the timing in millisecond until ET equal to PT, and then ET holds constant. The IN value will be invalid after ET starts to count the time. When ET equal to PT, Q will be FALSE and ET will be zero if IN changes into FALSE.



TP Function Block

3.1.12.1.2 Parameter Description

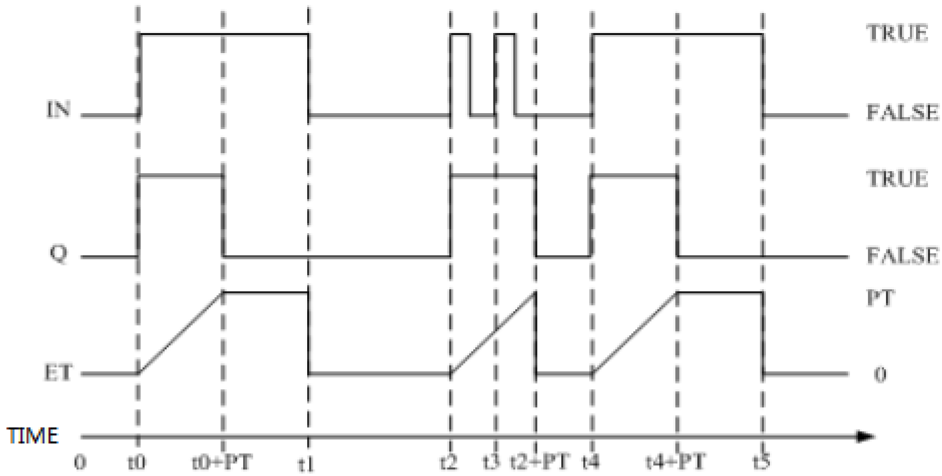
Input Parameter Description

Input Parameter	Data Type	Function Description	Parameter Description	Default Value
IN	BOOL	Timer trigger signal	The trigger timer will start the timing when IN changes ito TRUE from FALSE.	FALSE
PT	TIME	Timing value		0

Output Parameter Description

output Parameter	Data Type	Function Description	Parameter Description	Default Value
Q	BOOL	Timer output	The timer will start the operation and Q will be TRUE when IN is TRUE. IN will be invalid before ET is less than or equal to PT. Q will be FALSE when ET is equal to PT.	FALSE
ET	TIME	Current time value	ET will start the timing until ET is equal to PT when IN changes into TRUE, and then ET holds constant. On completion of the timing, ET will be equal to 0 when IN is FALSE.	0

3.1.12.1.3 Sequence Chart

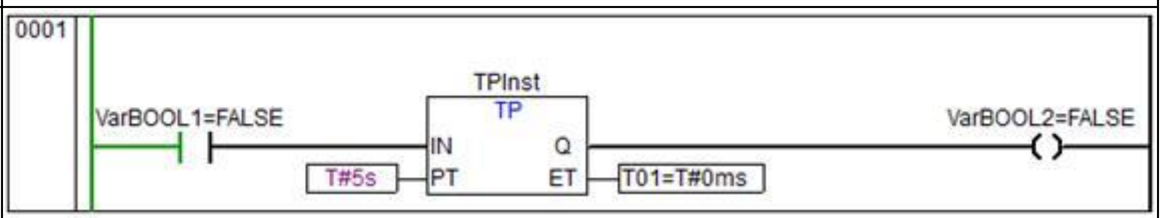
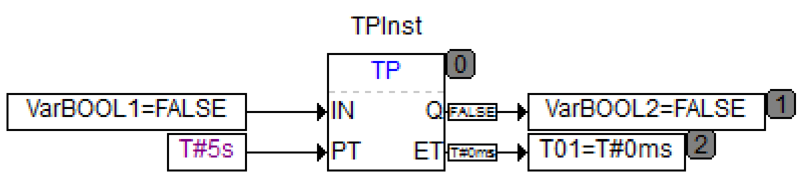


TP Sequence Chart

3.1.12.1.4 Application Example of Instructions

Variable Definition

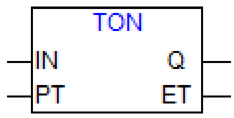
No.	Variable Name	Address	Variable Description	Variable Type	Initial Value	Power Fail Safeg
0001	TPInst			TP		FALSE
0002	VarBOOL1			BOOL	FALSE	FALSE
0003	VarBOOL2			BOOL	FALSE	FALSE
0004	T01			TIME	T#0MS	FALSE

Programing language	Program
Ladder diagram (LD)	
Structured text (ST)	TPInst(IN:=VarBOOL1, PT:=T#5S, Q=>VarBOOL2, ET=>T01);
Continuous Function Chart (CFC)	

3.1.12.2 TON (On Delay Timer)

3.1.12.3 Function

ET will start the timing in millisecond when IN changes into TRUE from FALSE. ET will hold the constant and Q will change into TRUE until ET is equal to PT. After ET starts the timing, the timer will stop the timing and ET will change into zero if IN changes into FALSE.



TON Function Block

3.1.12.4 Parameter Description

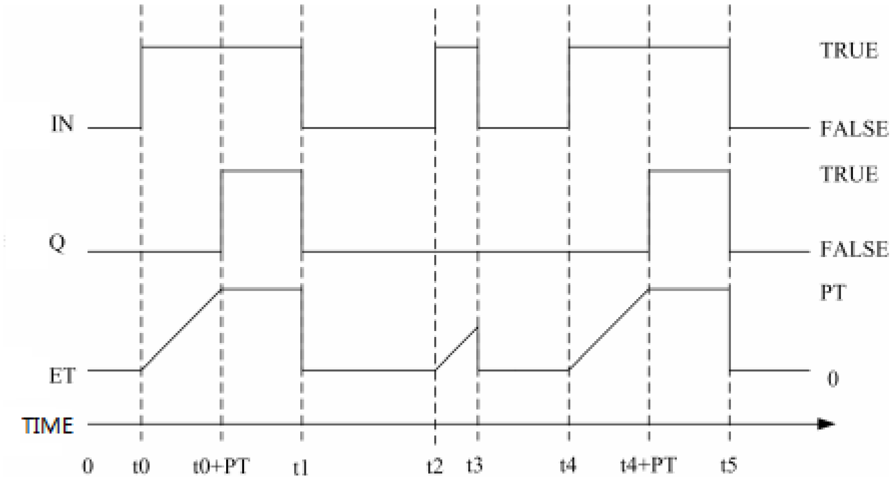
Input Parameter Description

Input Parameter	Data Type	Function Description	Parameter Description	Default Value
IN	BOOL	Timer trigger signal	The trigger will start the timing when IN changes into TRUE and stop the timing when IN is FALSE.	FALSE
PT	TIME	Timing value		0

Output Parameter Description

output Parameter	Data Type	Function Description	Parameter Description	Default Value
Q	BOOL	Timer output	Q will be FALSE and ET will be zero when IN is FALSE. The timer will start the timing when IN is TRUE, and Q will be TRUE when ET is equal to PT.	FALSE
ET	TIME	Current time value	ET will start the timing in millisecond when IN changes into TRUE until ET is equal to PT, and then ET will hold the constant. ET will be equal to zero whenever IN is FALSE.	0

3.1.12.5 Sequence Chart

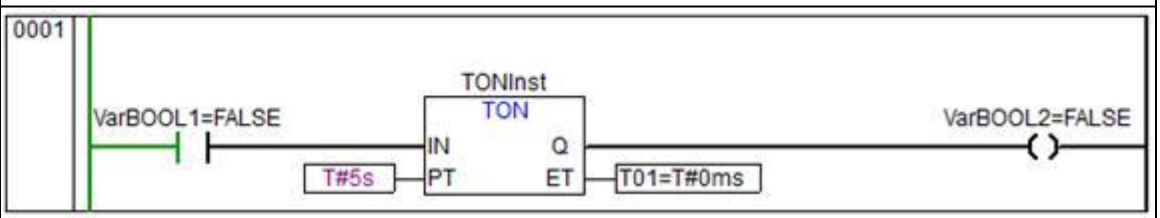
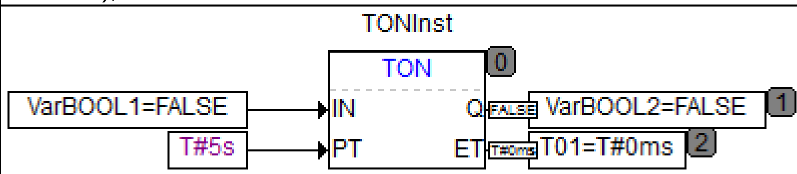


TON Sequence Chart

3.1.12.6 Application Example of Instructions

Variable Definition

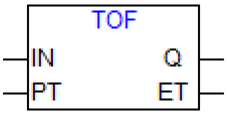
No.	Variable Name	Address	Variable Description	Variable Type	Initial Value	Power Fail Safegu
0001	TONInst			TON		FALSE
0002	VarBOOL1			BOOL	FALSE	FALSE
0003	VarBOOL2			BOOL	FALSE	FALSE
0004	T01			TIME	T#0MS	FALSE

Programing language	Program
Ladder diagram (LD)	
Structured text (ST)	TONInst(IN:=VarBOOL1, PT:=T#5S, Q=>VarBOOL2, ET=>T01);
Continuous Function Chart (CFC)	

3.1.12.7 TOF (Off Delay Timer)

3.1.12.8 Function

ET will start the timing in millisecond, Q will change into FALSE, and ET will hold the constant until ET is equal to PT when IN changes into FALSE from TRUE. After ET starts the timing, the timer will stop the timing, ET will change into zero and Q will hold the constant if IN changes into TRUE again.



TOF Function Block

3.1.12.9 Parameter Description

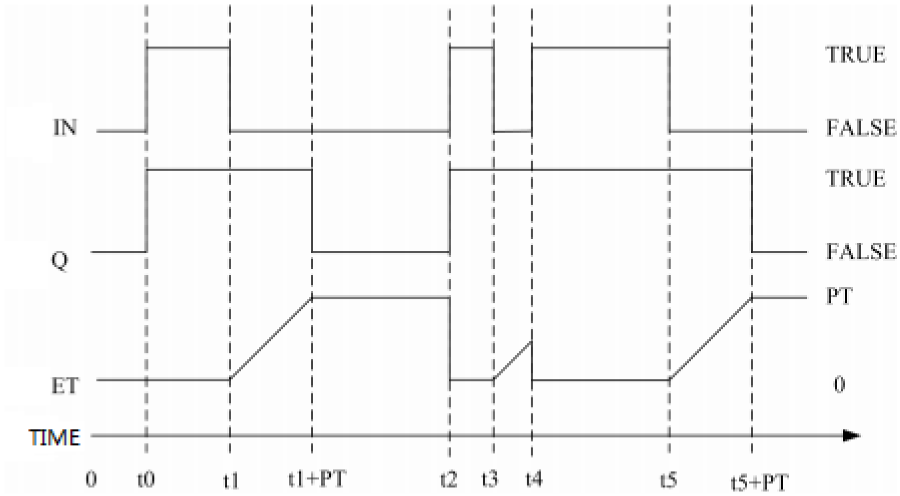
Input Parameter Description

Input Parameter	Data Type	Function Description	Parameter Description	Default Value
IN	BOOL	Timer trigger singal	ET will start the timing in millisecond until ET is equal to PT when IN changes into FALSE from TRUE, and then ET will hold the constant and Q will change into FALSE.	TRUE
PT	TIME	timing value		0

Output Parameter Description

Output Parameter	Data Type	Function Description	Parameter Description	Default Value
Q	BOOL	Timer output	Q will be FALSE when IN is FALSE and ET is equal to PT. Otherwise, Q will be TRUE.	TRUE
ET	TIME	Current time value	ET will start the timing in millisecond until ET is equal to PT when IN is FALSE, and then ET will hold the constant. ET will be equal to zero and Q will be TRUE whenever IN is TRUE.	0

3.1.12.10 Sequence Chart

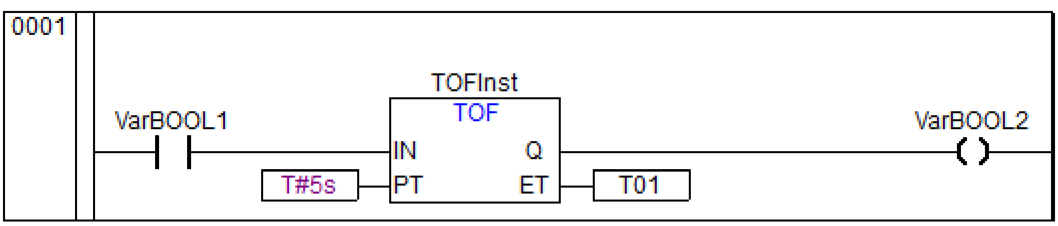


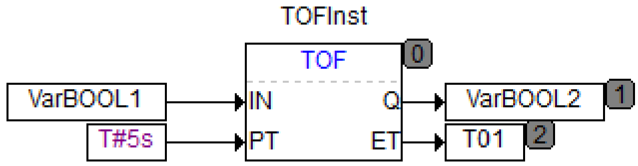
TOF Sequence Chart

3.1.12.11 Application Example of Instructions

Variable Definition

No.	Variable Name	Address	Variable Description	Variable Type	Initial Value	Power Fail Safegu
0001	TOFInst			TOF		FALSE
0002	VarBOOL1			BOOL	FALSE	FALSE
0003	VarBOOL2			BOOL	FALSE	FALSE
0004	T01			TIME	T#0MS	FALSE

Programing language	Program
Ladder diagram (LD)	
Structured text (ST)	TOFInst(IN:=VarBOOL1, PT:=T#5S, Q=>VarBOOL2, ET=>T01);
Continuous Function Chart (CFC)	

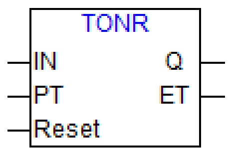
Variable Definition	
	

3.1.12.12 TONR (Retentive On Delay Timer)

3.1.12.12.1 Function

Retentive on delay timer can accumulate the ON time of the input signal.

ET will start the timing in millisecond until ET is equal to PT when IN changes into TRUE. If the external trigger causes IN changes into FALSE before the delay time is up to the set time value. At this time, ET will hold the time value before the timing stops. ET will continue the timing when IN is triggered to change into TRUE, and the output Q will be set until ET is equal to PT. If RESET is 1, all output values and intermediate variables will be reset and the timing will terminate. If RESET is FALSE, it will be allowed to perform the delay timing function.



TONR Function Block

3.1.12.12.2 Parameter Description

Input Parameter Description

Input Parameter	Data Type	Function Description	Parameter Description	Default Value
IN	BOOL	Timer trigger signal	The timer will start the timing when IN changes into TRUE, and the timer will stop the timing when IN changes into FALSE.	FALSE
PT	TIME	Timing value	The delay time value shall be set in the format of the TIME Data Type, whose unit may be d, h, m, s and ms and the minimum unit is ms.	0
Reset	BOOL	Reset input	TRUE: Reset all the output variables and middle variables with end timing. FALSE: Allow timer to execute delay timing.	FALSE

Output Parameter Description

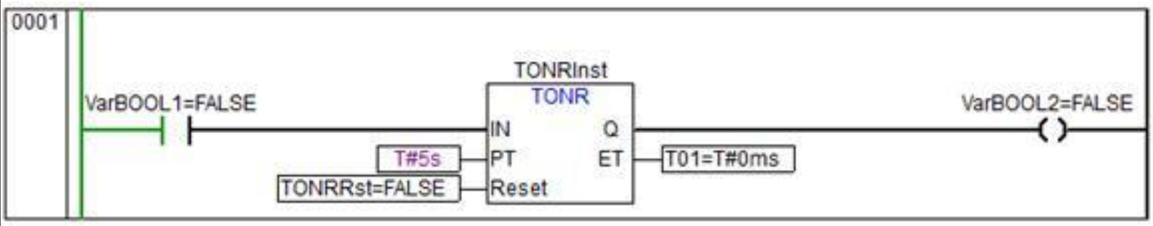
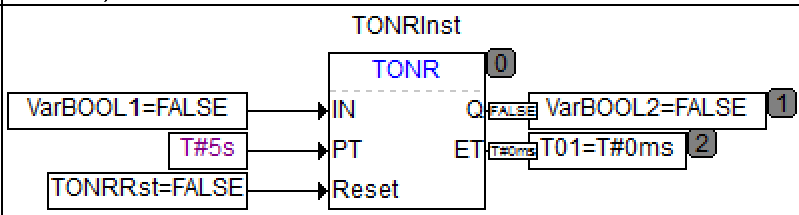
Output Parameter	Data Type	Function Description	Parameter Description	Default Value
Q	BOOL	Timer output	When Q is FALSE, it indicates ET is not up to the set value PT. When ET is equal to PT, Q will be TRUE. It will not affect the output of Q whether IN is TRUE or FALSE when the timing starts. The output of Q depends on whether ET is equal to PT and the status of the RESET terminal only.	TRUE

Output Parameter	Data Type	Function Description	Parameter Description	Default Value
ET	TIME	Current time value	ET will start the timing in millisecond until ET is equal to PT when IN changes into TRUE. ET will always output current time value until ET is equal to PT if only Q is FALSE whether IN is TRUE or FALSE when the timing starts.	0

3.1.12.12.3 Application Example of Instructions

Variable Definition

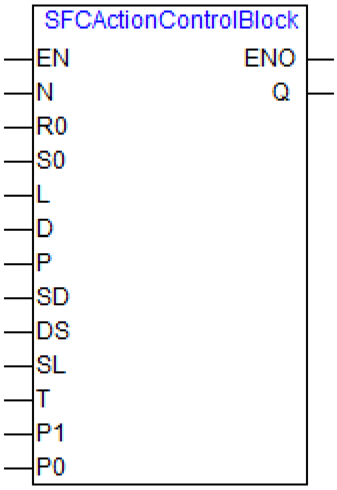
No.	Variable Name	Address	Variable Description	Variable Type	Initial Value	Power Fail Safeg
0001	TONRInst			TONR		FALSE
0002	VarBOOL1			BOOL	FALSE	FALSE
0003	VarBOOL2			BOOL	FALSE	FALSE
0004	TONRRst			BOOL	FALSE	FALSE
0005	T01			TIME	T#0MS	FALSE

Programing language	Program
Ladder diagram (LD)	
Structured text (ST)	TONRInst(IN:=VarBOOL1, PT:=T#5S, Reset:=TONRRst, Q=>VarBOOL2, ET=>T01);
Continuous Function Chart (CFC)	

3.1.13 IEC_SFC Functions

3.1.13.1 SFCActionControlBlock

This instruction is used to control the IEC step associated action in the SFC language.



IEC Action Instruction

ⓘIt is necessary to add the SFCActionControlBlock library when you use the IEC step in the SFC programming language. Otherwise, it will prompt the error during the compilation. If there is associated action for the IEC step, the system will call the SFCActionControl instruction automatically.

3.2 Basic Application Instruction

3.2.1 BCD Conversion Functions

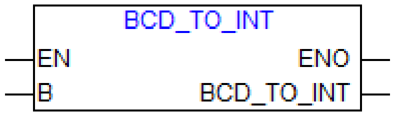
A byte of the BCD code contains an integer between 0 and 99. Each decimal bit corresponds to 4 bits, the ten digit is stored in bits 4-7, and the single digit is stored in bits 0-3. The BCD format is very similar to the hexadecimal, except that the BCD byte value is 0-99 and the hexadecimal is 0-FF.

The 4-digit binary number of the BCD code represents a decimal number, and the decimal number is represented by a string of BCD codes. For example, the decimal number 59 indicates that the BCD code is 0101 1001, but is expressed as binary 2 # 111011. Decimal 51 is converted to BCD code, 5 is represented by binary 0101,1 is represented by binary 0001, then 51 is expressed as binary 0101 0001.

3.2.1.1 BCD_TO_INT (BCD Code to Integer)

3.2.1.1.1 Function

The BCD_TO_INT algorithm requires the input value is of the BYTE type and the output value is of the INT type, to complete the BCD_TO_INT conversion.



BCD_TO_INT Function Block

3.2.1.1.2 Parameter Description

Input Parameter Description

Input Parameter	Data Type	Parameter Description	Default Value
B	BYTE	BCD code to be converted. Enter the BCD code in binary form (or decimal and hexadecimal), such as BCD code 49, expressed as 2 # 100 1001 (or corresponding to 10 # 73, 16 # 49), then enter 2 # 100 1001 (or 10 # 73, 16 # 49)	0

Output Parameter Description

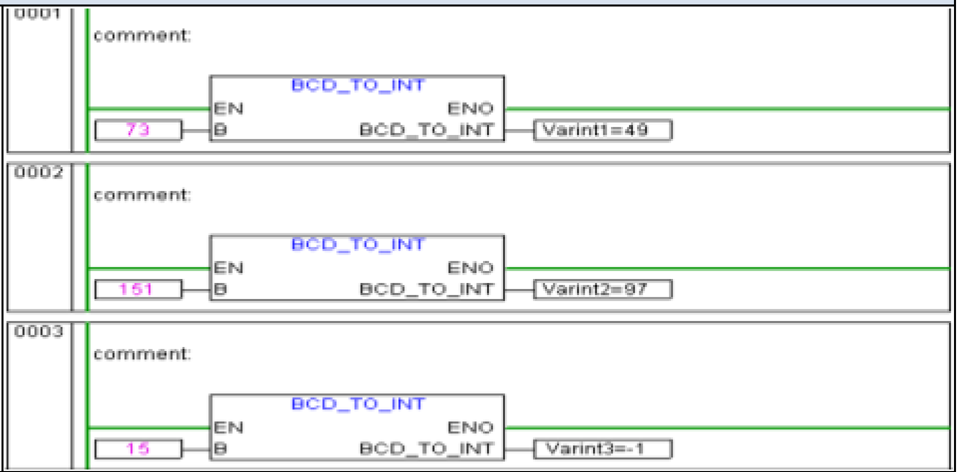
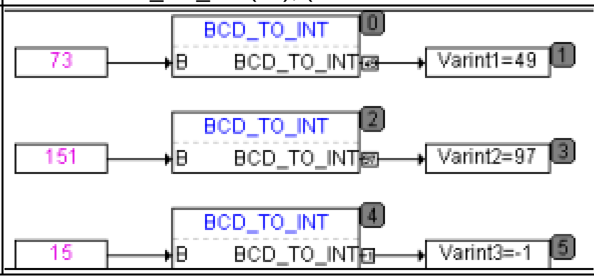
Output Parameter	Data Type	Parameter Description	Default Value
BCD_TO_INT	INT	INT value after conversion If the input byte is not the BCD code, the output is -1	0

3.2.1.1.3 Application Example of Instructions

Variable Definition

No.	Variable Name	Address	Variable Description	Variable Type	Initial Value	Power Fail Safeguard
0001	Varint1			INT	0	FALSE
0002	Varint2			INT	0	FALSE
0003	Varint3			INT	0	FALSE

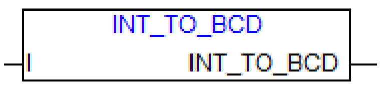
Programing language	Program

<table>
<tr><th colspan="2">Variable Definition</th></tr>
<tr><td>Ladder diagram (LD)</td><td></td></tr>
<tr><td>Structured text (ST)</td><td>Varint1:=BCD_TO_INT(73); (*The result is 49*)
Varint2:=BCD_TO_INT(151); (*The result is 97*)
Varint3:=BCD_TO_INT(15); (*The result is -1 for it is not the BCD code format *)</td></tr>
<tr><td>Continuous Function Chart (CFC)</td><td></td></tr>
</table>

3.2.1.2 INT_TO_BCD (Integer to BCD Code)

3.2.1.2.1 Function

The INT_TO_BCD algorithm requires the input value is the INT type and the output value is the BYTE type, to complete the integer to BCD code conversion. When the integer value can’t be converted to BCD code byte, the output value is 255.



INT_TO_BCD Function Block

3.2.1.2.2 Parameter Description

Input Parameter Description

Input Parameter	Data Type	Parameter Description	Default Value
I	INT	Integer data value to be converted	0

		If the integer value is 49, the 49 is entered here	

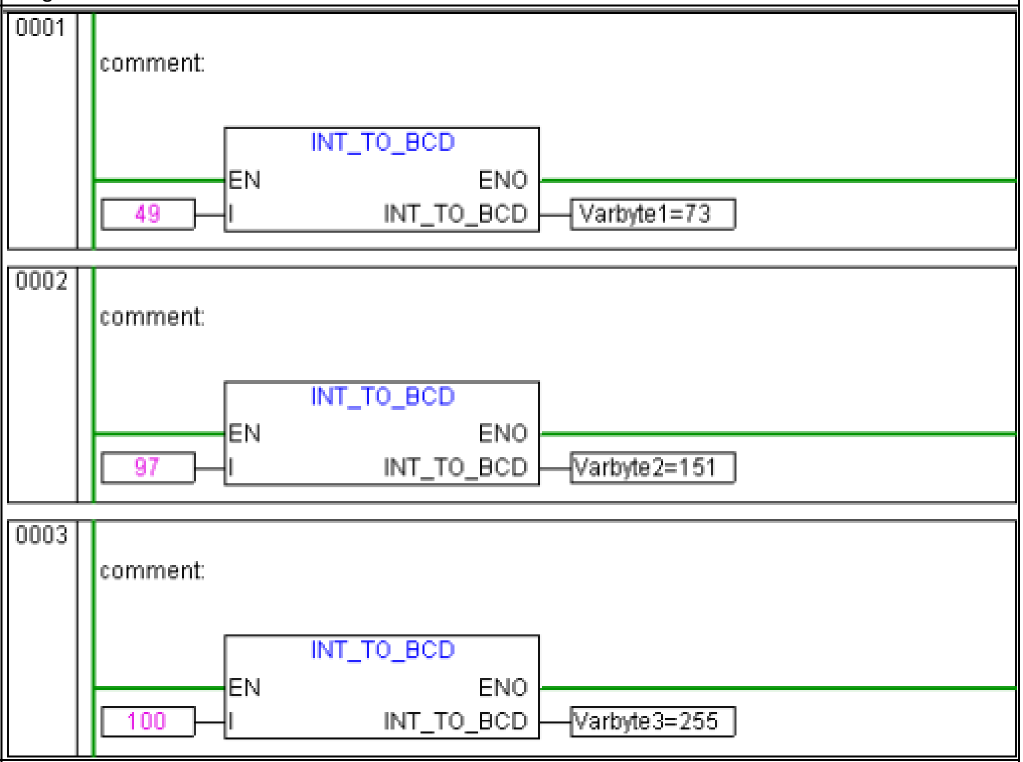
Output Parameter Description

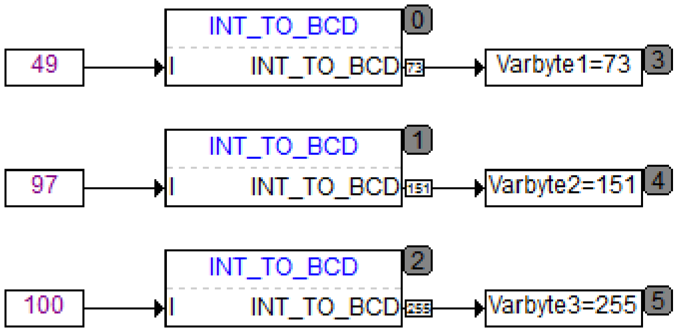
output Parameter	Data Type	Parameter Description	Default Value
INT_TO_BCD	BYTE	BCD code after conversion For example, 49 is converted to BCD for 2 # 100 1001, then output value is 2 # 1001001 (or 10 # 73, 16 # 49).	0

3.2.1.2.3 Application Example of Instructions

Variable Definition

No.	Variable Name	Address	Variable Description	Variable Type	Initial Value	Power Fail Safeguard
0001	Varbyte1			BYTE	0	FALSE
0002	Varbyte2			BYTE	0	FALSE
0003	Varbyte3			BYTE	0	FALSE

Programing language	Program
Ladder diagram (LD)	
Structured text (ST)	Varbyte1:=INT_TO_BCD(49); (*The result is 73*) Varbyte2:=INT_TO_BCD(97); (*The result is 151*) Varbyte2:=INT_TO_BCD(100); (*Error! Output: 255*)

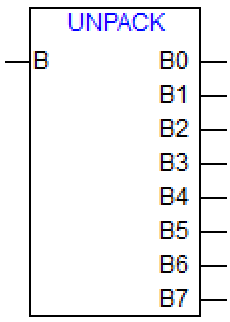
Variable Definition	
Continuous Function Chart (CFC)	

3.2.2 Bit_Byte Functions

3.2.2.1 UNPACK (Bit Split)

3.2.2.1.1 Function

Get the bit 7 ~ 0 of the input B and assign it to B7 ~ B0. It is opposite to the PACK instruction.



UNPACK Function Block

3.2.2.1.2 Parameter Description

Input Parameter Description

ID	Description	Variable Type	Initial Value (Default)	Power Fail Safeguard
B	BYTE type input variable	BYTE	0	FALSE

Output Parameter Description

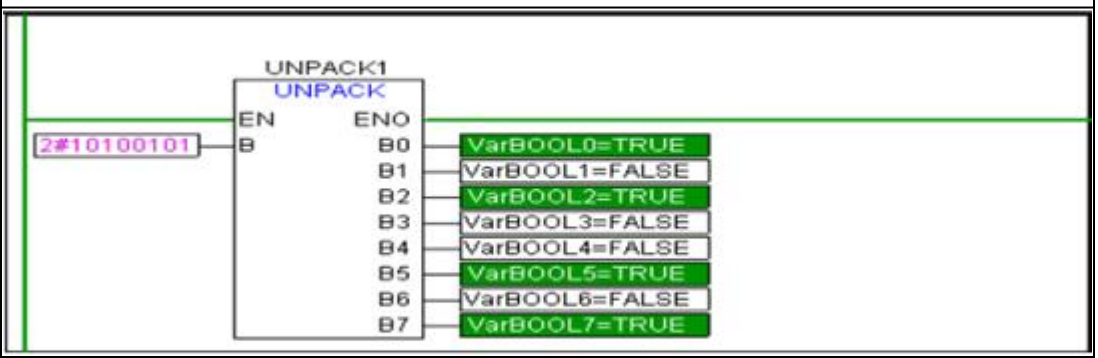
ID	Description	Variable Type	Initial Value (Default)	Power Fail Safeguard
B0	BOOL type output variable	BOOL	FALSE	FALSE
B1		BOOL	FALSE	FALSE

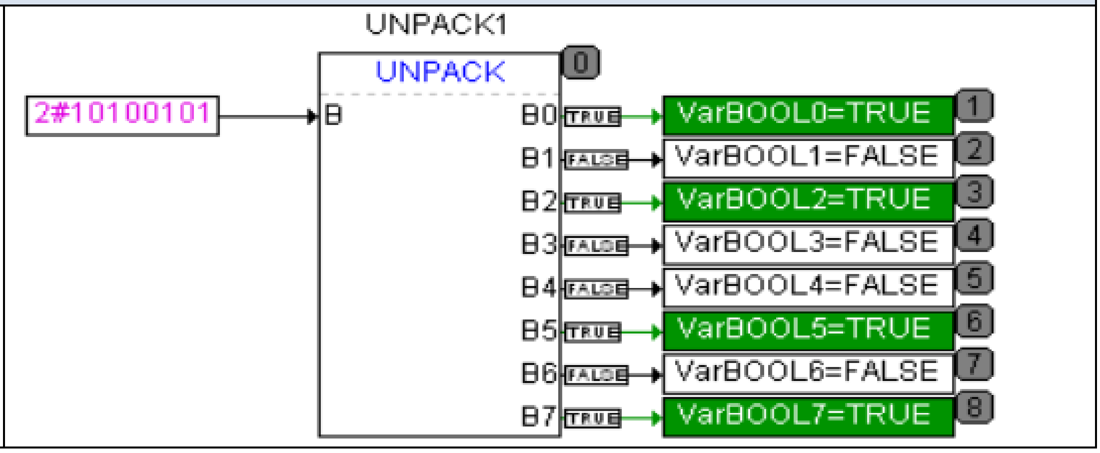
ID	Description	Variable Type	Initial Value (Default)	Power Fail Safeguard
B2		BOOL	FALSE	FALSE
B3		BOOL	FALSE	FALSE
B4		BOOL	FALSE	FALSE
B5		BOOL	FALSE	FALSE
B6		BOOL	FALSE	FALSE
B7		BOOL	FALSE	FALSE

3.2.2.1.3 Application Example of Instructions

Variable Definition

No.	Variable Name	Address	Variable Description	Variable Type	Initial Value	Power Fail Sa
0001	UNPACK1			UNPACK		FALSE
0002	VarBOOL0			BOOL	FALSE	FALSE
0003	VarBOOL1			BOOL	FALSE	FALSE
0004	VarBOOL2			BOOL	FALSE	FALSE
0005	VarBOOL3			BOOL	FALSE	FALSE
0006	VarBOOL4			BOOL	FALSE	FALSE
0007	VarBOOL5			BOOL	FALSE	FALSE
0008	VarBOOL6			BOOL	FALSE	FALSE
0009	VarBOOL7			BOOL	FALSE	FALSE

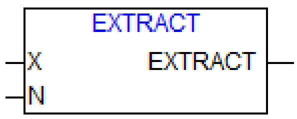
Programing language	Program
Ladder diagram (LD)	
Structured text (ST)	UNPACK1(B:=2#10101010, B0=>VarBOOL0, B1=>VarBOOL1, B2=>VarBOOL2, B3=>VarBOOL3, B4=>VarBOOL4, B5=>VarBOOL5, B6=>VarBOOL6, B7=>VarBOOL7);

Variable Definition	
Continuous Function Chart (CFC)	

3.2.2.2 EXTRACT (Bit Extraction)

3.2.2.2.1 Function

Output the specified Nth bit of the input variable X.



EXTRACT Function Block

3.2.2.2.2 Parameter Description

Input Parameter Description

Input Parameter	Data Type	Function Description	Initial Value (Default)
X	DWORD	Manipulated variable	0
N	BYTE	Position	0

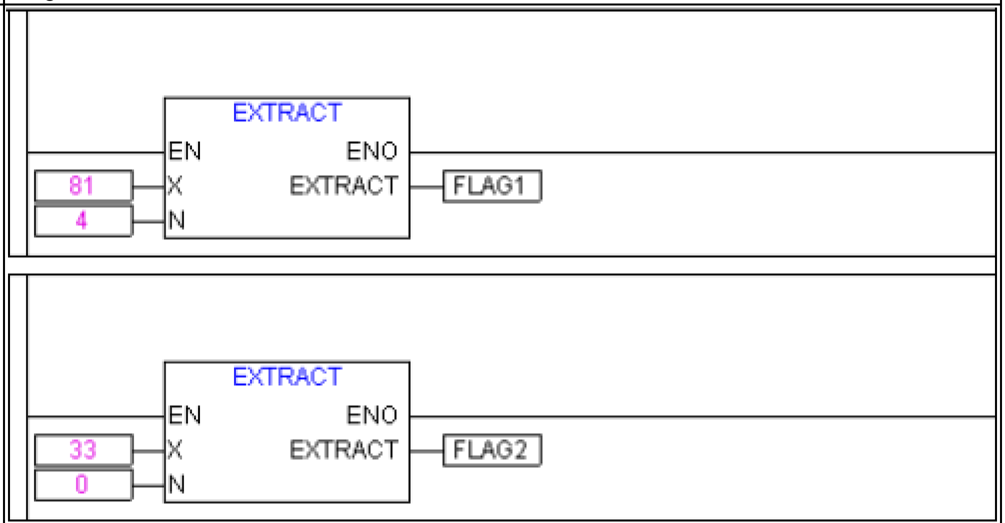
Output Parameter Description

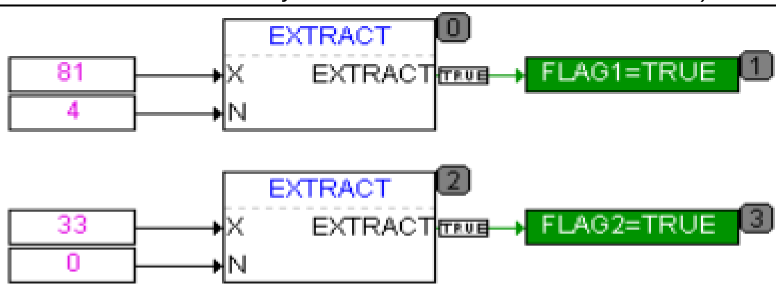
Output Parameter	Data Type	Function Description	Initial Value (Default)
EXTRACT	BOOL	Output	0

3.2.2.2.3 Application Example of Instructions

<table>
<tr><th colspan="2">Variable Definition</th></tr>
<tr><td colspan="2">

No.	Variable Name	Address	Variable Description	Variable Type	Initial Value	Power Fail Safegua
0001	FLAG1			BOOL	FALSE	FALSE
0002	FLAG2			BOOL	FALSE	FALSE

</td></tr>
<tr><td>Programing language</td><td>Program</td></tr>
<tr><td>Ladder diagram (LD)</td><td>

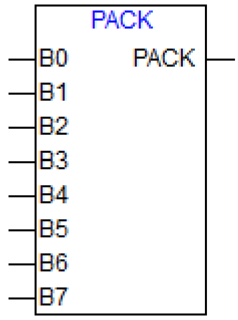
(*Result FLAG1=TRUE，FLAG2=TRUE *)</td></tr>
<tr><td>Structured text (ST)</td><td>FLAG1:= EXTRACT (81 ,4);
(*Result : TRUE, for the binary value of 81 is 1010001, the fourth bit is 1*)
FLAG2:= EXTRACT (33 ,0);
(*Result : TRUE, for the binary value of 33 is 100001, the 0th bit is 1*)</td></tr>
<tr><td>Continuous Function Chart (CFC)</td><td>

(*Result FLAG1=TRUE, FLAG2=TRUE *)</td></tr>
</table>

3.2.2.3 PACK (Bit Merger)

3.2.2.3.1 Function

Convert input B0 ~ B7 to a BYTE variable Bit0~Bit7 from low bit to high bit. It is opposite to the UNPACK instruction.



PACK Function Block

3.2.2.3.2 Parameter Description

Input Parameter Description

Input Parameter	Variable Type	Description	Initial Value (Default)
B0	BOOL	Bit0	0
B1	BOOL	Bit1	0
B2	BOOL	Bit2	0
B3	BOOL	Bit3	0
B4	BOOL	Bit4	0
B5	BOOL	Bit5	0
B6	BOOL	Bit6	0
B7	BOOL	Bit7	0

Output Parameter Description

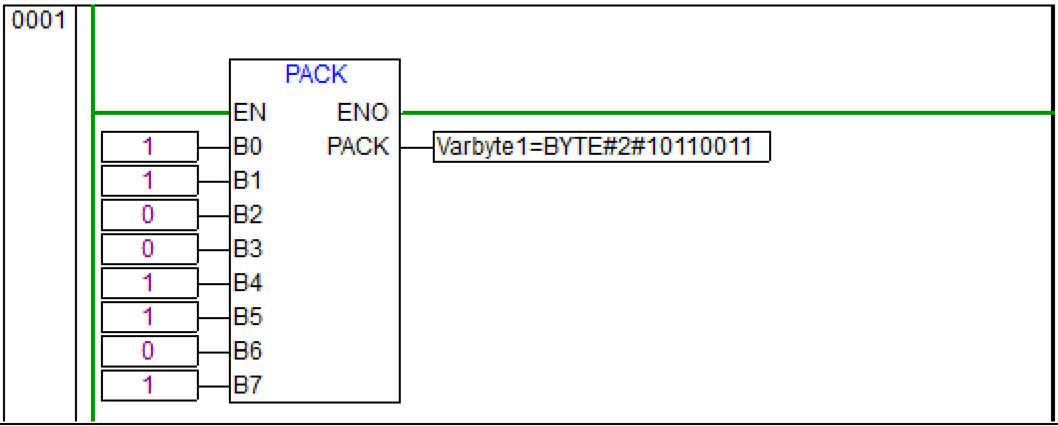
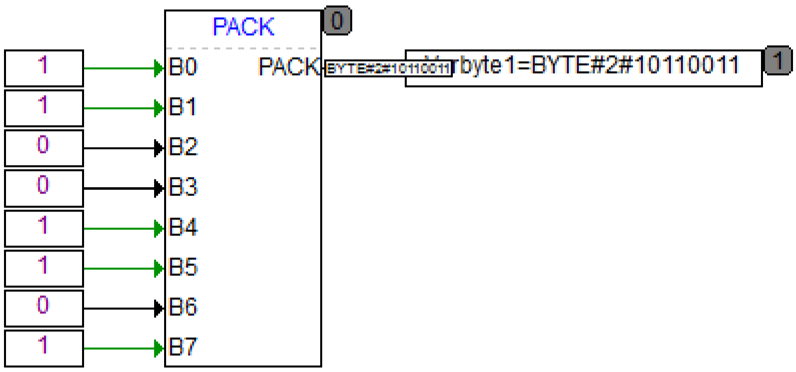
Output Parameter	Data Type	Description	Initial Value (Default)
Pack	BYTE	Output: B7~B0 (high to low)	0

3.2.2.3.3 Application Example of Instructions

Variable Definition

No.	Variable Name	Address	Variable Description	Variable Type	Initial Value	Power Fail Safeguard
0001	Varbyte1			BYTE	0	FALSE

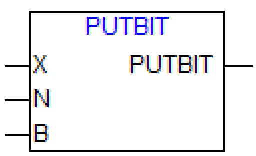
Programing language	Program

<table>
<tr><th colspan="2">Variable Definition</th></tr>
<tr><td>Ladder diagram (LD)</td><td>
</td></tr>
<tr><td>Structured text (ST)</td><td>Varbyte1:= PACK(1,1,0,0,1,1,0,1);   (*The result is 2#10110011*)</td></tr>
<tr><td>Continuous Function Chart (CFC)</td><td>
</td></tr>
</table>

3.2.2.4 PUTBIT (Bit Assignment)

3.2.2.4.1 Function

The PUTBIT will set the specified Nth bit of the input value X as the specified value B.



PUTBIT Function Block

3.2.2.4.2 Parameter Description

Input Parameter Description

Input Parameter	Data Type	Description	Initial Value (Default)
X	DWORD	Manipulated variable	0
N	BYTE	Set	0

Input Parameter	Data Type	Description	Initial Value (Default)
B	BOOL	Specified value	FALSE

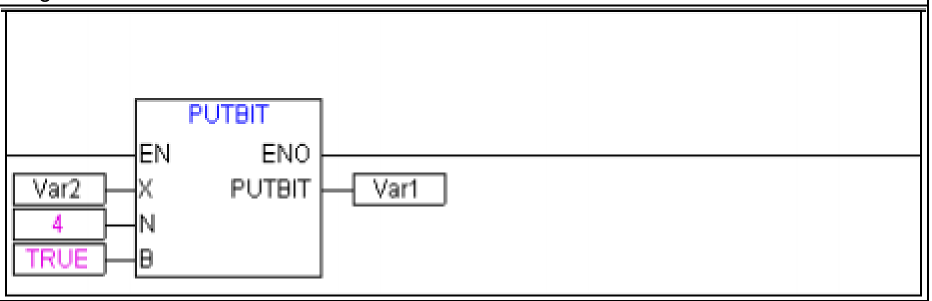
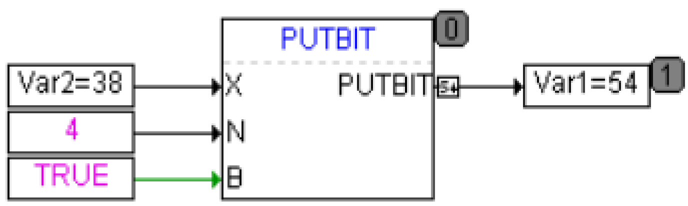
Output Parameter Description

Output Parameter	Data Type	Description	Initial Value (Default)
PutBit	DWORD	Output the variable value that completes the specified value operation	0

3.2.2.4.3 Application Example of Instructions

Variable Definition

No.	Variable Name	Address	Variable Description	Variable Type	Initial Value	Power Fail Safegua
0001	Var2			DWORD	0	FALSE
0002	Var1			DWORD	0	FALSE

<table>
<tr><td>Programing language</td><td>Program</td></tr>
<tr><td>Ladder diagram (LD)</td><td>
</td></tr>
<tr><td>Structured text (ST)</td><td>Var2:=38; (*Binary 100110*)
Var1:=PUTBIT(Var2,4,TRUE); (*Result : 54 = 2#110110*)</td></tr>
<tr><td>Continuous Function Chart (CFC)</td><td>
</td></tr>
</table>

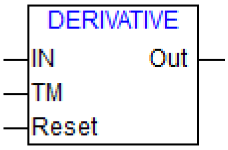
3.2.3 Mathematical Functions

3.2.3.1 DERIVATIVE (Derivative Function)

3.2.3.1.1 Function

This instruction performs a differential operation on a continuously input variable. For best results, the

DERIVATIVE only differentiates the latest four input values, reducing errors due to inaccurate input parameters.



DERIVATIVE Function Block

3.2.3.1.2 Differential iteration formula

$$\mathrm{OUT} = \frac{3 * [IN(\mathrm{k}) - IN(\mathrm{k}-3)] + IN(\mathrm{k}-1) - IN(\mathrm{k}-2)}{3 * TM(k-2) + 4 * TM(k-1) + 3 * TM(k)} * 1000$$

K-3, k-2, k-1, k are the marks of four consecutive input values.

3.2.3.1.3 Parameter Description

Input Parameter Description

Input Parameter	Data Type	Function Description	Initial Value (Default)
IN	REAL	Input variable	0
TM	DWORD	Time constant	0
Reset	BOOL	Restart signal	FALSE

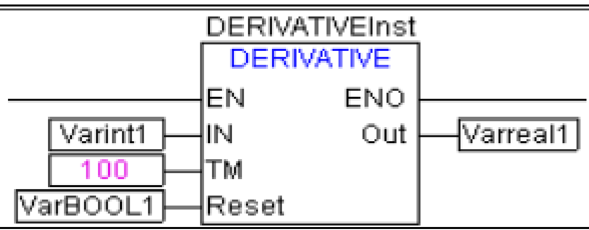
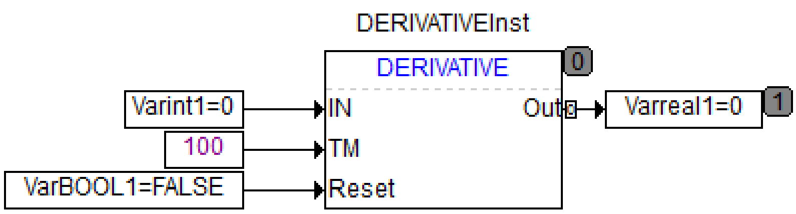
Output Parameter Description

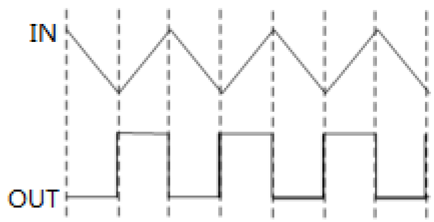
Output Parameter	Data Type	Function Description	Initial Value (Default)
OUT	REAL	Output of differential algorithm	0

3.2.3.1.4 Application Example of Instructions

Variable Definition

No.	Variable Name	Address	Variable Description	Variable Type	Initial Value	Power Fail Safegua
0001	DERIVATIVEInst			DERIVATIVE		FALSE
0002	Varint1			INT	0	FALSE
0003	VarBOOL1			BOOL	FALSE	FALSE
0004	Varreal1			REAL	0	FALSE

Programing language	Program
Ladder diagram (LD)	
Structured text (ST)	DERIVATIVEInst(IN:=Varint1, TM:=100, Reset:=VarBOOL1, Out=>Varreal1);
Continuous Function Chart (CFC)	

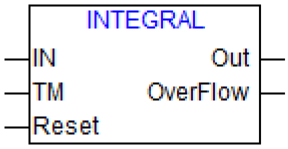


Input – Output Relation

3.2.3.2 INTEGRAL (Integral Function)

3.2.3.2.1 Function

The integral algorithm (INTEGRAL) is used to solve the input function appropriately.



INTEGRAL Function Block

3.2.3.2.2 Differential iteration formula

$$A(k) = A(k-1) + TM * IN(k-1)$$

$$B(k) = B(k-1) + TM * IN(k)$$

$$OUT(k) = \frac{A(k) + B(k)}{2}$$

k-1, k are the marks of two consecutive input values.

3.2.3.2.3 Parameter Description

Input Parameter Description

Input Parameter	Data Type	Function Description	Initial Value (Default)	Power Fail Safeguard
IN	REAL	Input variable	0	FALSE
TM	DWORD	Time constant	0	TRUE
Reset	BOOL	Restart signal	FALSE	TRUE

Output Parameter Description

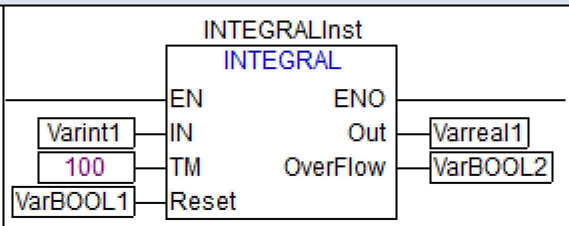
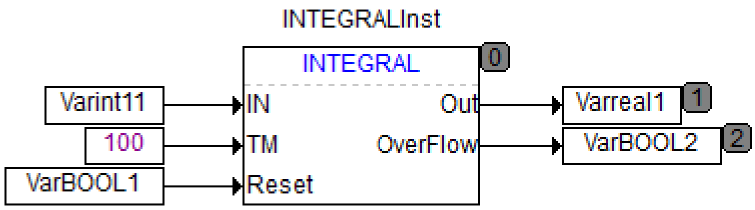
Output Parameter	Data Type	Function Description	Initial Value (Default)	Power Fail Safeguard
OUT	REAL	Approximate integral value	0	TRUE
OverFlow	BOOL	Overflow flag	FALSE	TRUE

3.2.3.2.4 Application Example of Instructions

Variable Definition

No.	Variable Name	Address	Variable Description	Variable Type	Initial Value	Power Fail Safeguard
0001	INTEGRALInst			INTEGRAL		FALSE
0002	Varint1			INT	0	FALSE
0003	VarBOOL1			BOOL	FALSE	FALSE
0004	Varreal1			REAL	0	FALSE
0005	VarBOOL2			BOOL	FALSE	FALSE

Programing language	Program

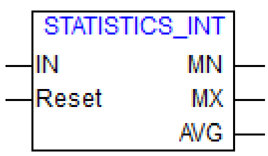
Variable Definition	
Ladder diagram (LD)	
Structured text (ST)	INTEGRALInst(IN:=Varint1, TM:=100, Reset:=VarBOOL1, Out=>Varreal1, OverFlow=> VarBOOL2);
Continuous Function Chart (CFC)	

Input – Output Relation

3.2.3.3 STATISTICS_INT (Int Statistics)

3.2.3.3.1 Function

The computational standard statistical algorithm (STATISTICS_INT) will sum up the minimum, maximum and average of all INT type input variables.



STATISTICS_INT Function Block

3.2.3.3.2 Parameter Description

Input Parameter Description

Input Parameter	Data Type	Function Description	Initial Value (Default)

IN	INT	Manipulated variable	0
Reset	BOOL	Initialization	FALSE

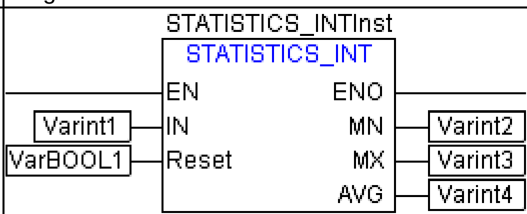
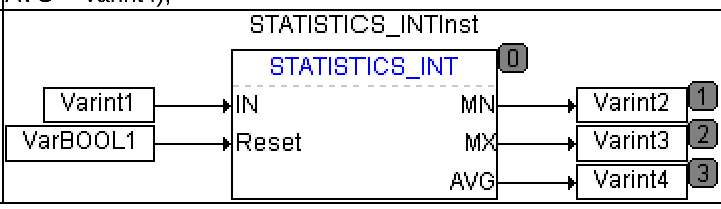
Output Parameter Description

Output Parameter	Data Type	Function Description	Initial Value (Default)
MN	INT	Minimum output	32767
MX	INT	Maximum output	-32768
AVG	INT	Average output	0

3.2.3.3.3 Application Example of Instructions

Variable Definition

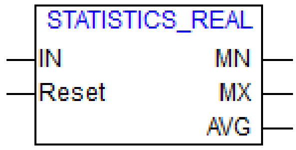
No.	Variable Name	Address	Variable Description	Variable Type	Initial Value	Power Fail Safeguard
0001	STATISTICS_INTInst			STATISTICS_INT		FALSE
0002	Varint1			INT	0	FALSE
0003	VarBOOL1			BOOL	FALSE	FALSE
0004	Varint2			INT	0	FALSE
0005	Varint3			INT	0	FALSE
0006	Varint4			INT	0	FALSE

Programing language	Program
Ladder diagram (LD)	
Structured text (ST)	STATISTICS_INTInst(IN:= Varint1, Reset:= VarBOOL1, MN=>Varint2, MX=>Varint3, AVG=>Varint4);
Continuous Function Chart (CFC)	

3.2.3.4 STATISTICS_REAL (Real Statistics)

3.2.3.4.1 Function

Sum up the maximum, minimum and average of all REAL type input variables.



STATISTICS_ REAL Function Block

3.2.3.4.2 Parameter Description

Input Parameter Description

Input Parameter	Data Type	Function Description	Initial Value (Default)
IN	REAL	Manipulated variable	0
Reset	BOOL	Initialization	FALSE

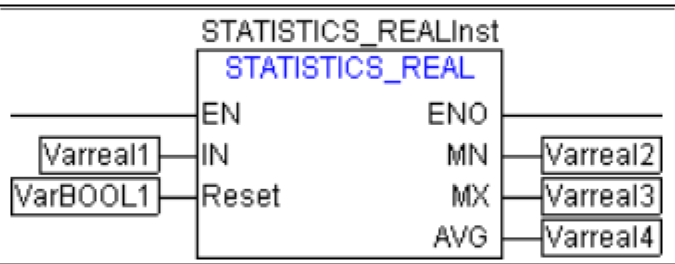
Output Parameter Description

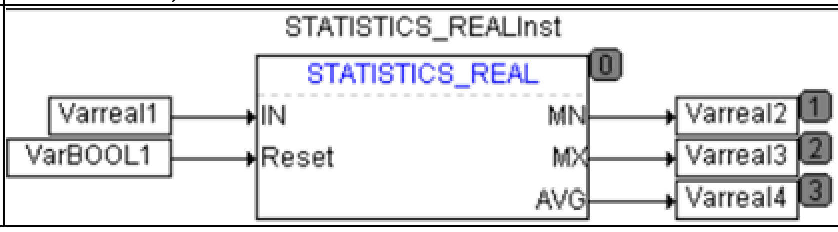
Output Parameter	Data Type	Function Description	Initial Value (Default)
MN	REAL	Minimum output	3E+38
MX	REAL	Maximum output	1E-37
AVG	REAL	Average output	0

3.2.3.4.3 Application Example of Instructions

Variable Definition

No.	Variable Name	Address	Variable Description	Variable Type	Initial Value	Power Fail Safeguard
0001	STATISTICS_REALInst			STATISTICS_REAL		FALSE
0002	Varreal1			REAL	0	FALSE
0003	VarBOOL1			BOOL	FALSE	FALSE
0004	Varreal2			REAL	0	FALSE
0005	Varreal3			REAL	0	FALSE
0006	Varreal4			REAL	0	FALSE

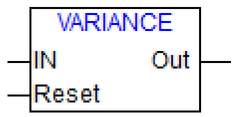
Programing language	Program
Ladder diagram (LD)	
Structured text (ST)	STATISTICS_REALInst(IN:=Varreal1, Reset:=VarBOOL1, MN=>Varreal2, MX=>Varreal3,

Variable Definition	
	AVG=>Varreal4);
Continuous Function Chart (CFC)	

3.2.3.5 VARIANCE (Calculation of Variance)

3.2.3.5.1 Function

The input variance algorithm (VARIANCE) will calculate the variance of all input variables. The standard deviation can be obtained by the square root of the square deviation.



VARIANCE Function Block

3.2.3.5.2 Parameter Description

Input Parameter Description

Input Parameter	Data Type	Function Description	Initial Value (Default)	Power Fail Safeguard
IN	REAL	Manipulated variable	0	FALSE
Reset	BOOL	Initialization	FALSE	TRUE

Output Parameter Description

Output Parameter	Data Type	Function Description	Initial Value (Default)	Power Fail Safeguard
Out	REAL	Variance output	0	FALSE

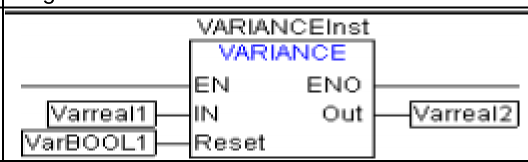
Parameter Description of Intermediate Variable

Intermediate Parameter	Data Type	Initial Value (Default)	Power Fail Safeguard
Z	INT	0	TRUE
A	REAL	0	TRUE
B	REAL	0	TRUE

3.2.3.5.3 Application Example of Instructions

Variable Definition

No.	Variable Name	Address	Variable Description	Variable Type	Initial Value	Power Fail Safeguard
0001	VARIANCEInst			VARIANCE		FALSE
0002	VarBOOL1			BOOL	FALSE	FALSE
0003	Varreal1			REAL	0	FALSE
0004	Varreal2			REAL	0	FALSE

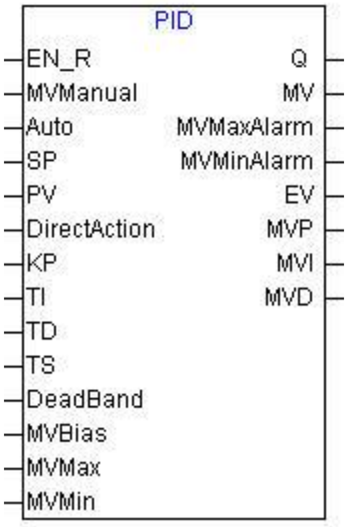
Programing language	Program
Ladder diagram (LD)	
Structured text (ST)	VARIANCEInst(IN:=Varreal1, Reset:=VarBOOL1, OUT=>Varreal2);
Continuous Function Chart (CFC)	

3.2.4 Controller Functions

3.2.4.1 PID (The PID Controller)

3.2.4.1.1 Function

The PID controller will compare the process value of the field input with the set value and carry out the PID control. The controlled instruction will be output to the field device, to respond to the control task timely and conduct the limit operation of the output value, so as to ensure the output is operated within the allowable range, and alarm the out-of-limit value timely.



PID Function Block

3.2.4.1.2 Parameter Description

Input Parameter Description

Input Parameter	Data Type	Function Description	Parameter Description	Default Value	Power Fail Safeguard
EN_R	BOOL	Enable	FALSE: Invalid TRUE: Rising edge setting, high level valid	FALSE	TRUE
MVManual	REAL	Manual input value		0	TRUE
Auto	BOOL	"Automatic" mode selection	FALSE: Manual TRUE: Automatic	FALSE	TRUE
SP	REAL	Set value		0	TRUE
PV	REAL	Process value		0	FALSE
DirectAction	BOOL	"Positive action" mode selection	FALSE: Negative action (EV=SP-PV) TRUE: Positive action (EV=PV-SP)	FALSE	TRUE
KP	REAL	Proportional gain		1	TRUE
TI	REAL	Integral time (S)	Ti>=0, it indicates there is not the integral part if the integral time is zero.	1	TRUE
TD	REAL	Derivative time (S)	Td>=0, it indicates there is not the derivative part if the derivative time is zero.	0	TRUE
TS	REAL	Operation cycle (S)	Ts>0	0	TRUE
DeadBand	REAL	Deviation dead zone	(1) It indicates there is not any dead zone if DeadBand=0. (2) It is valid if the absolute value of the deviation is greater than this value. Otherwise, the deviation is zero.	0	TRUE
MVBias	REAL	Feed forward control		0.0	TRUE
MVMax	REAL	Upper limit of output value		+100	TRUE
MVMin	REAL	Lower limit of output value		-100	TRUE

Output Parameter Description

Output Parameter	Data Type	Function Description	Parameter Description	Default Value	Power Fail Safeguard
Q	BOOL	Enable state flag	FALSE: Invalid TRUE: Rising edge setting, high level valid	FALSE	TRUE
MV	REAL	Controlled quantity Output		0	TRUE
MVMaxAlarm	BOOL	Alarm for output out of upper limit	FALSE: No over upper limit TRUE: Output has exceeded the upper limit	FALSE	TRUE
MVMinAlarm	BOOL	Alarm for output out of lower limit	FALSE: No over lower limit TRUE: Output has exceeded the lower limit	0	TRUE
EV	REAL	Deviation between set value and process value	If there is a dead zone, EV is the deviation after the dead zone	0	FALSE
MVP	REAL	Proportional component		FALSE	TRUE
MVI	REAL	Integral component		1	TRUE
MVD	REAL	Differential component		1	TRUE

3.2.4.1.3 Detailed Function Description

■ Enable

The program will operate normally when the enable terminal EN_R is TRUE. Otherwise, the program will return No Operation.

1 Judgment of Set Value

1 Judge the setting of the upper limit MVMax and the lower limit MVMin for the output value.

The lower limit of the output value shall be less than the upper limit of the output value. Otherwise, the output value will be out of the upper and lower limit flag and will be set, and the program will return No Operation.

2 Judge the setting of the integral time Ti, differential time Td and dead zone range DeadBand.

The integral time Ti, differential time Td and dead zone range DeadBand shall be the non-negative number. Otherwise, it will be operated by the initial value.

2 Calculation

3 Manual Mode

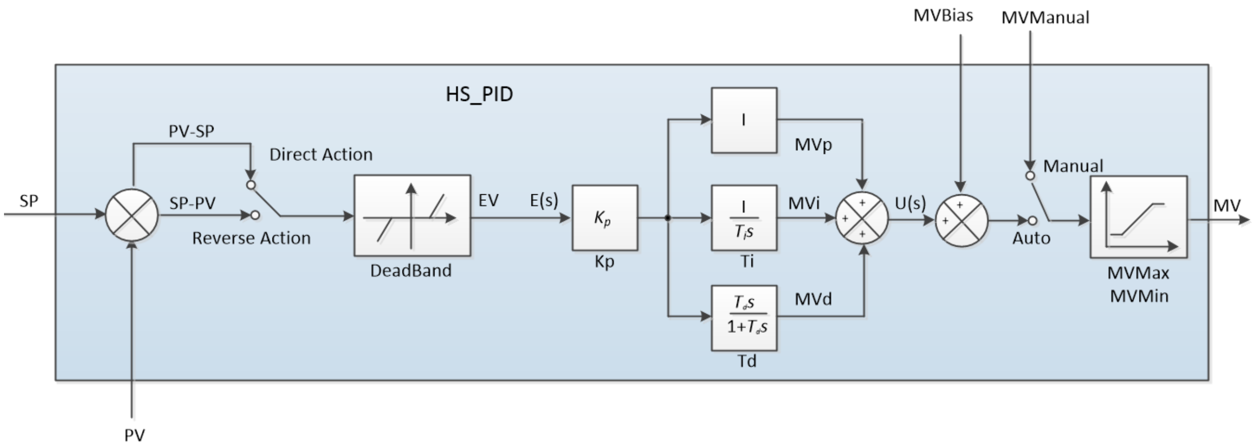
The output MV is equal to the manual input value MVManual, and SP tracks PV.

4 Automatic Mode

Calculate the deviation in accordance the positive and negative action mode DerectAction, Set valueSP, measurement value PV and dead zone range DeadBand, and then calculate the integral saturation coefficient in accordance with the output MV, upper and lower limit of the output value and the positive and negative deviation.

Calculate and control the proportional component MVp, integral component MVi, differential component MVd and control output MV of the output value in accordance with the proportional gain Kp, intergral time Ti, differential time Td and integral saturation coefficient.

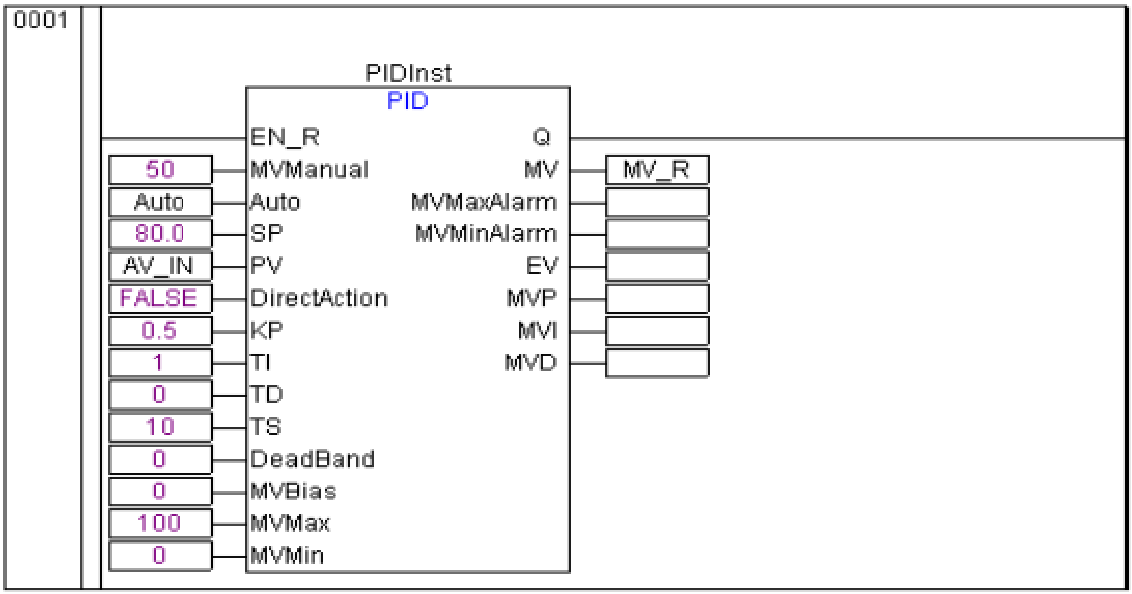
The logic structure of the PID function block is shown in figure.

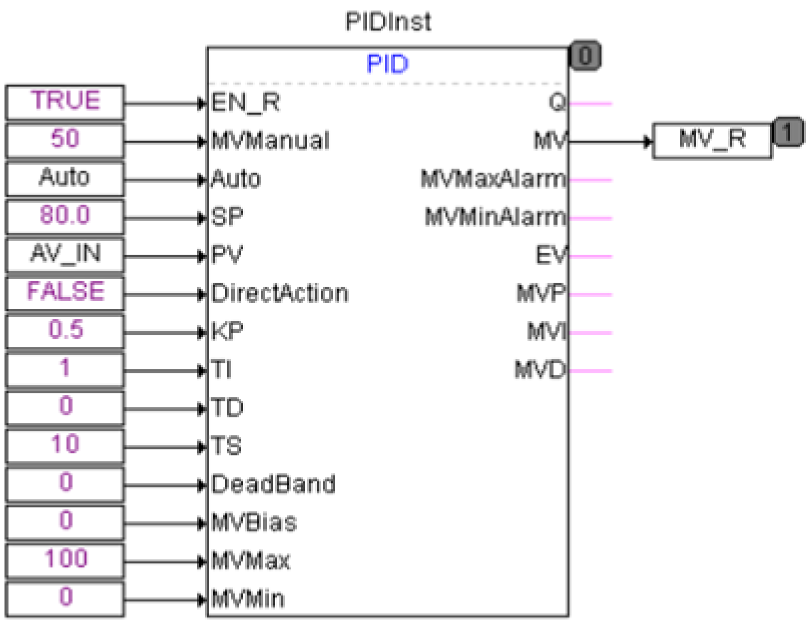


Logic Structure of PID Function Block

3.2.4.1.4 Application Example of Instructions

Variable Definition						
No.	Variable Name	Address	Variable Description	Variable Type	Initial Value	Power Fail Safegua
0001	PIDInst			PID		FALSE
0002	Auto			BOOL	FALSE	FALSE
0003	AV_IN			REAL	0	FALSE
0004	MV_R			REAL	0	FALSE
Programing language	Program					

<table>
<tr><th colspan="2">Variable Definition</th></tr>
<tr><td>Ladder diagram (LD)</td><td>PLC_PRG: Setpoint is 80, it will automatically adjust when the Auto is TRUE (it will output 50 when Auto is FALSE), reverse direction control (the output will increase when the actual value less than the setpoint), the dimension of MVMax, MVMin the same as the actual value and setpoint value ,here set to 100 and 0

</td></tr>
<tr><td>Structured text (ST)</td><td>PIDInst(
EN_R:=TRUE,
MVManual:=50,
Auto:=Auto,
SP:=80.0,
PV:=AV_IN,
DirectAction:=FALSE,
KP:=0.5,
TI:=1,
TD:=0,
TS:=10,
DeadBand:=0,
MVBias:=0,
MVMax:=100.0,
MVMin:=0,
MV=>MV_R</td></tr>
</table>

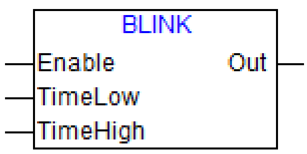
<table>
<tr><th colspan="2">Variable Definition</th></tr>
<tr><td>Continuous Function Chart (CFC)</td><td>
</td></tr>
</table>

3.2.5 Signal Generator Functions

3.2.5.1 BLINK (Pulse Signal Generator)

3.2.5.1.1 Overview

The BLINK algorithm will output the rectangular wave with the high level and the low level at the specified time.



BLINK Function Block

3.2.5.1.2 Parameter Description

Input Parameter Description

Input Parameter	Description	Variable Type	Initial Value (Default)	Power Fail Safeguard
Enable	Enable terminal	BOOL	FALSE	TRUE
TimeLow	Low level time	TIME	T#0S	TRUE
TimeHigh	High level time	TIME	T#0S	TRUE

Output Parameter Description

Output Parameter	Description	Variable Type	Initial Value (Default)	Power Fail Safeguard
Out	Output	BOOL	FALSE	TRUE

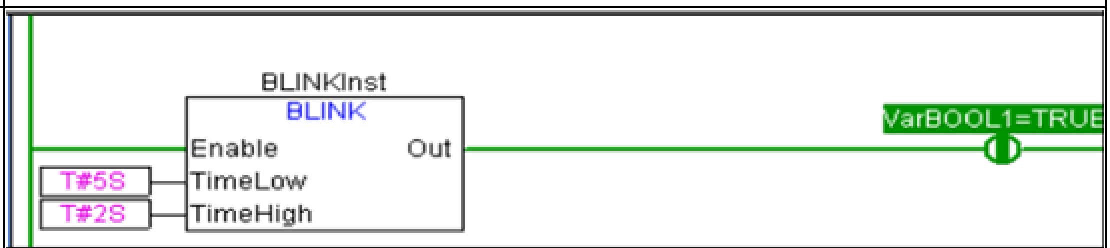
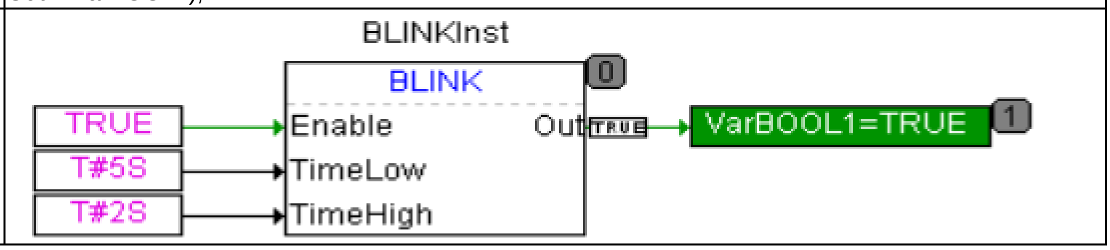
Parameter Description of Intermediate Variable

Intermediate Variable	Description	Variable Type	Initial Value (Default)	Power Fail Safeguard
CLOCK	Function block instance	TP		TRUE

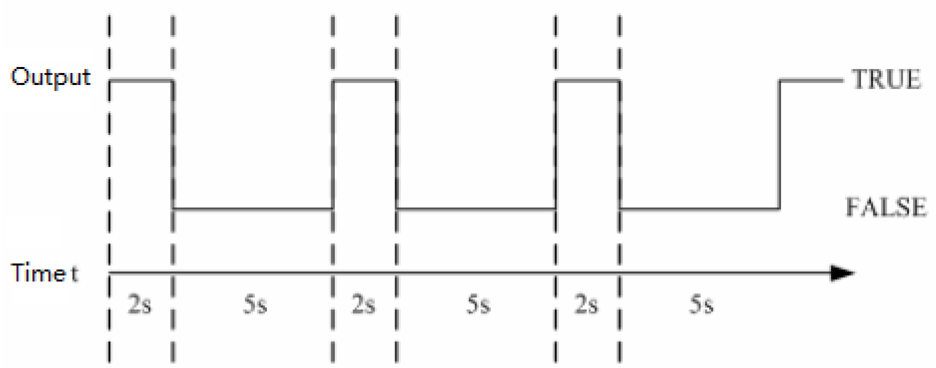
3.2.5.1.3 Application Example of Instructions

Variable Definition

No.	Variable Name	Address	Variable Description	Variable Type	Initial Value	Power Fail Safeguard
0001	BLINKInst			BLINK		FALSE
0002	VarBOOL1			BOOL	FALSE	FALSE

Programing language	Program
Ladder diagram (LD)	
Structured text (ST)	BLINKInst(Enable:=TRUE, TimeLow:=T#5S, TimeHigh:=T#2S, Out=>VarBOOL1);
Continuous Function Chart (CFC)	

When the instruction is executed, the waveform output of OUT is as shown in figure.

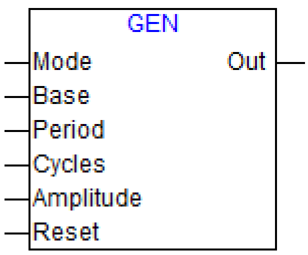


Output Waveform

3.2.5.2 GEN (Typical Signal Generator)

3.2.5.2.1 Overview

The GEN algorithm will generate the signal with the specified periodic function type, cycle and amplitude.



GEN Function Block

3.2.5.2.2 Parameter Description

Input Parameter Description

Input Parameter	Variable Type	Function Description	Parameters Description	Initial Value (Default)
Mode	GEN_MODE	Defines the type of signal to be generated	Directly input TRIANGLE, TRIANGLE_POS, SAWTOOTH_RISE, SAWTOOTH_FALL, RECTANGLE, SINUS, COSINUS in the MODE, then produce the corresponding waveform	TRIANGLE
		TRIANGLE	Triangle wave	
		TRIANGLE_POS	Triangle wave with zero point	
		SAWTOOTH_RISE	Rising sawtooth wave	
		SAWTOOTH_FALL	Descending sawtooth wave	
		RECTANGLE	Square wave	
		SINUS	Sine wave	
		COSINU	Cosine wave	
Base	BOOL	Cycle mode selection	When BASE is TRUE, the signal generator is related to the defined cycle period. When BASE is FALSE, the signal generator is related to the	FALSE

Input Parameter	Variable Type	Function Description	Parameters Description	Initial Value (Default)
			number of occurrences	
Period	TIME	Define the cycle time (valid when Base is true)		T#1S
Cycles	INT	Define the cycle time (Valid when Base is false)		1000
Amplitude	INT	The amplitude of the signal		0
Reset	BOOL	initialization	When RESET = TRUE, the signal generator is reset to zero	FALSE

Output Parameter Description

Output Parameter	Variable Type	Function Description	Initial Value (Default)
Out	INT	Waveform signal output value	0

Description of Intermediate Variable

Intermediate Variable	Description	Variable Type	Initial Value (Default)
CET		DINT	0
PER		DINT	0
COUNTER	Count variable	INT	0
CLOCK	Function block instance	TON	

ⓘThe first level of the instruction output is high and can't be selected. The enable terminal is disconnected and the output is held.

3.2.5.2.3 Detailed Function Description

- Initialization Function

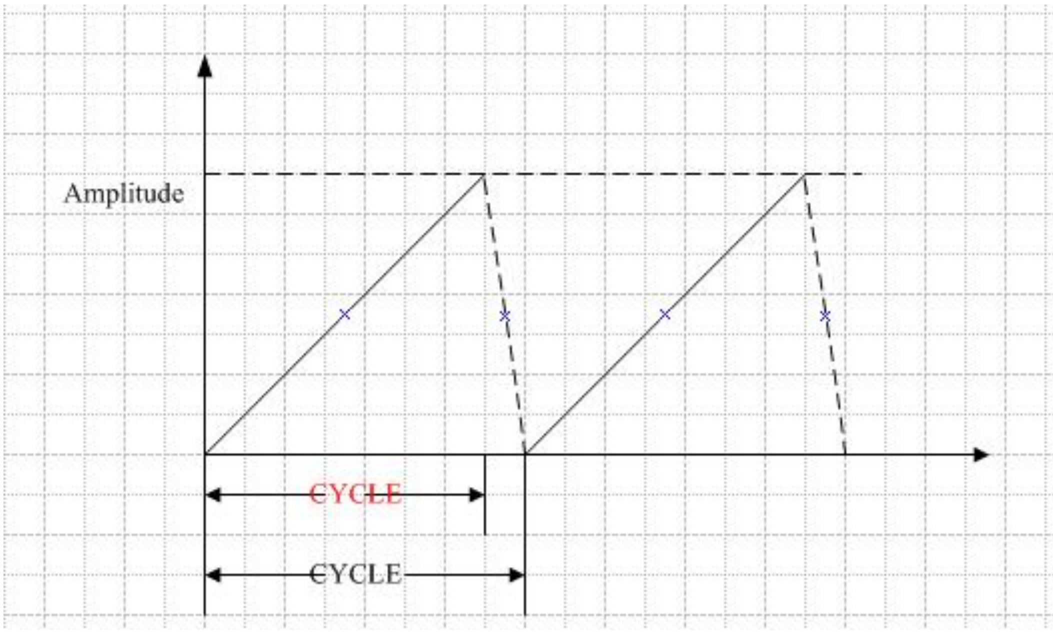
  When RESET=1, the output and relevant count variable will be cleared.

- Calculated Waveform

  When BASE=0, it will generate the MODE waveform by CYCLES and AMPLITUDE. When BASE=1, the cycle is different only, which will change into PERIOD.

3.2.5.2.4 Considerations

- When BASE=0 and the function type is SAWTOOTH_RISE or SAWTOOTH_FALL, the cycle shown by CYCLE is shown by the black font (other than red font) in the figure below (for there is not the vertical waveform in the actual signal, namely, one time point does not correspond to two values).

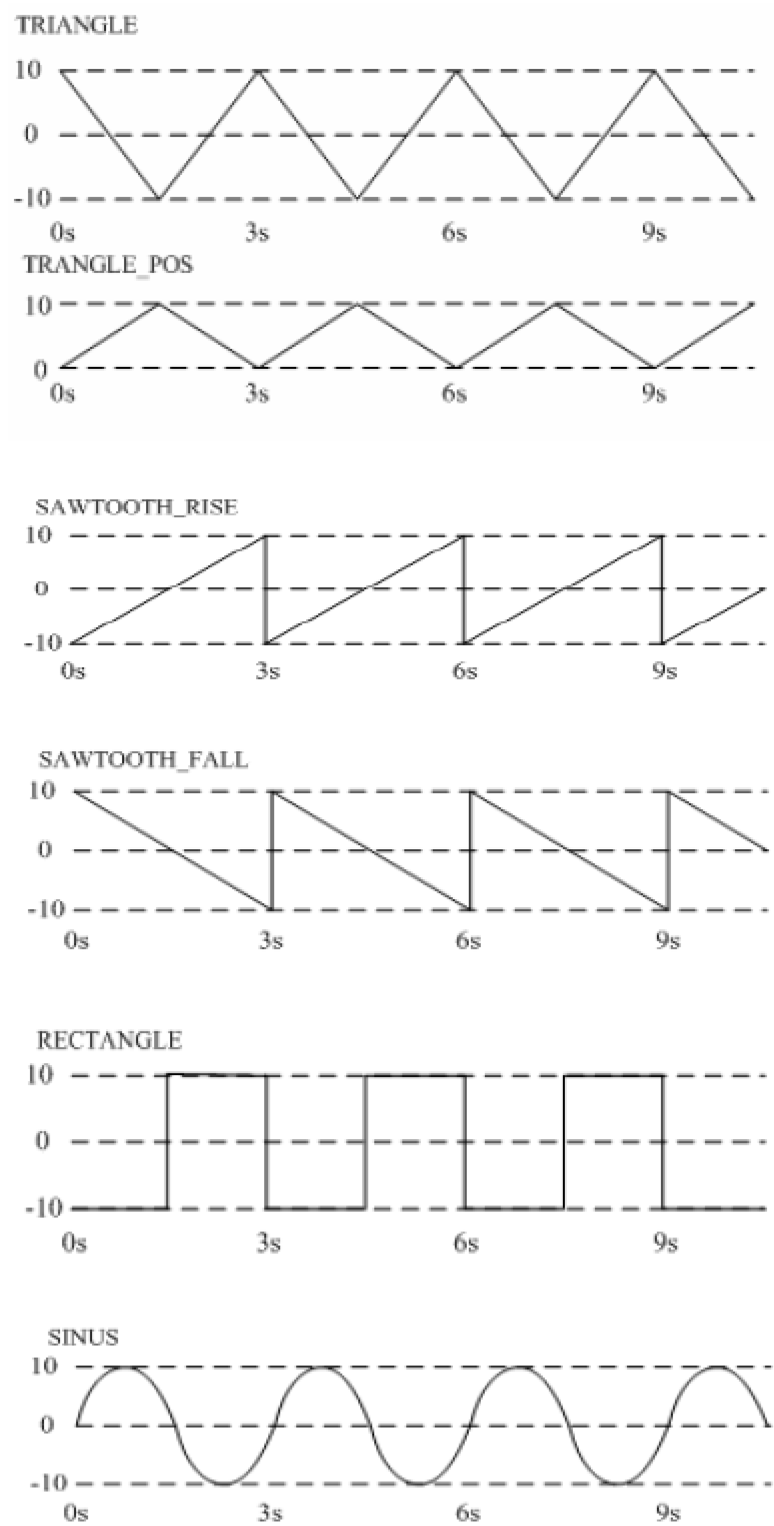


Cycle Chart

- When BASE=1 and the function type is SINUS or COSINUS, it can collect the full waveform if the ratio of the IEC operation cycle to the waveform cycle PERIOD is within 0.1.

  It may not collect the peak and valley value if the ratio is greater than 0.1, and the maximum value and the minimum value that can be collected shall be related to the ratio.

  Depending on the input in Mode, the resulting waveform is shown below.

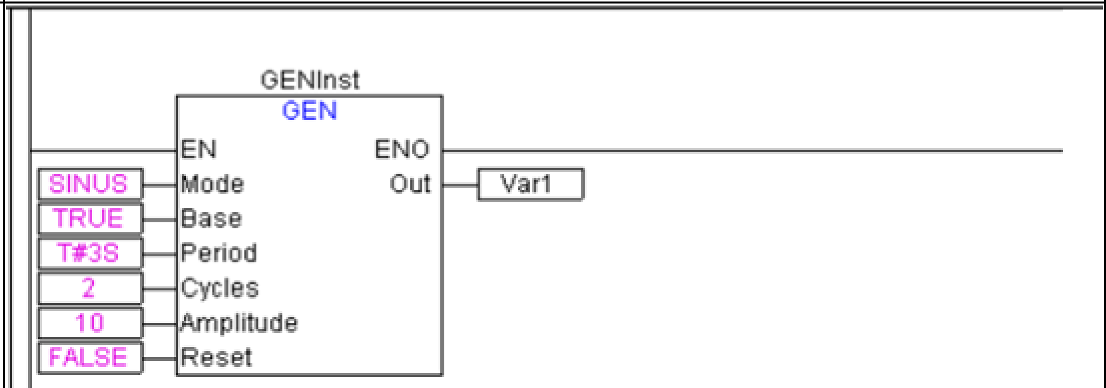
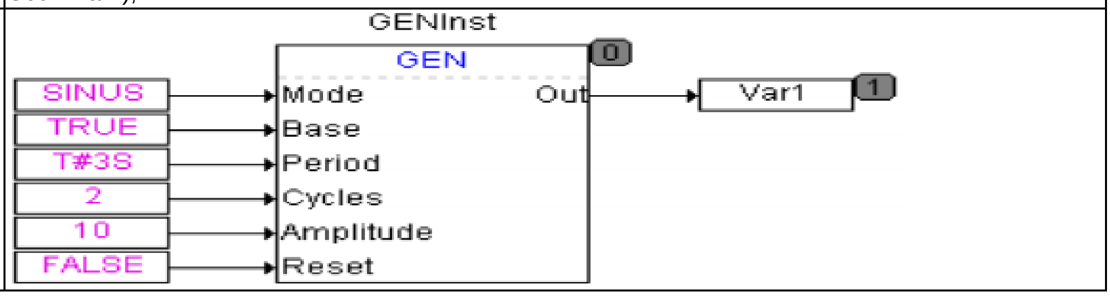

TRIANGLE
10
0
-10
0s
3s
6s
9s
TRANGLE_POS
10
0
0s
3s
6s
9s
SAWTOOTH_RISE
10
0
-10
0s
3s
6s
9s
SAWTOOTH_FALL
10
0
-10
0s
3s
6s
9s
RECTANGLE
10
0
-10
0s
3s
6s
9s
SINUS
10
0
-10
0s
3s
6s
9s

Waveforms

3.2.5.2.5 Application Example of Instructions

Variable Definition

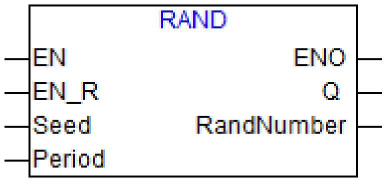
No.	Variable Name	Address	Variable Description	Variable Type	Initial Value	Power Fail Safegua
0001	GENInst			GEN		FALSE
0002	Var1			INT	0	FALSE

Programing language	Program
Ladder diagram (LD)	
Structured text (ST)	GENInst(Mode:=SINUS, Base:=TRUE, Period:=T#3S, Cycles:=2, Amplitude:=10, Reset:=FALSE, Out=>Var1);
Continuous Function Chart (CFC)	

3.2.5.3 RAND(Random Number Generator)

3.2.5.3.1 Function

Generate the random number.



RAND Function Block

3.2.5.3.2 Parameter Description

Input Parameter Description

Input Parameter	Data Type	Function Description	Parameter Description
EN_R	BOOL	Enable	0: Invalid, stop generating random number, and RandNumber holds the last generated one. 1: Rising edge enable, and high level valid
Seed	UINT	Seed	The seed is the initial value to calculate the random sequence. Value range: 0-65535 To ensure the continuity of the random number generation, the Seed value takes effect only in the following scenarios: full download, controller reset, "Cold Reset" and "Reset" in AT
Period	UDINT	Cycle to generate random number (unit: ms)	0: Generate one random number in each scanning cycle. Non 0: If it is set as 200, generate one random number every 200ms. Value range: 0~4294967295

Output Parameter Description

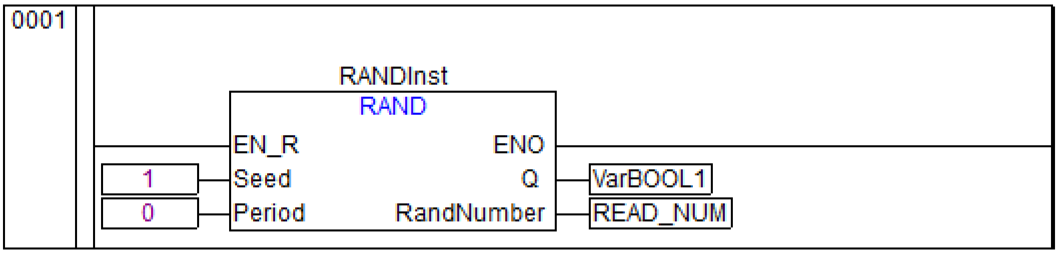
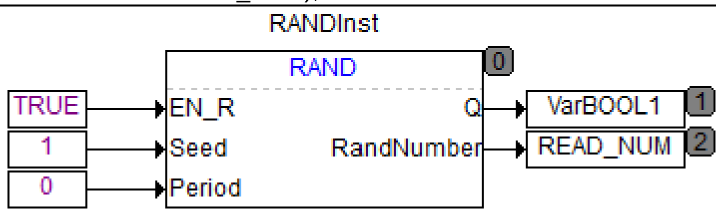
Output Parameter	Data Type	Function Description	Parameter Description
Q	BOOL	Whether the data is valid?	0: Invalid data 1: Valid data
RandNumber	UINT	Random number	0-65535

3.2.5.3.3 Application Example of Instructions

Variable Definition

No.	Variable Name	Address	Variable Description	Variable Type	Initial Value	Power Fail Safegua
0001	RANDInst			RAND		FALSE
0002	READ_NUM			UINT	0	FALSE
0003	VarBOOL1			BOOL	FALSE	FALSE

Programing language	Program

Variable Definition	
Ladder diagram (LD)	
Structured text (ST)	RANDInst(EN_R:=TRUE, Seed:=1, Period:=0, Q=>VarBOOL1, RandNumber=>READ_NUM);
Continuous Function Chart (CFC)	

Program description:

When EN_R is set, it will generate one random number every cycle.

When EN_R is reset, it will stop generating the random number, and RandNumber will hold the last generated random number.

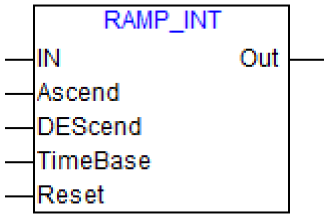
3.2.6 Analog Signal Processing Functions

3.2.6.1 RAMP_INT (Int Rate Limit)

3.2.6.1.1 Function

Limit the speed of the integer input function.

OUT is the function value of IN after the setting of the ascend and descend limit. When TIMEBASE is set as t#0s, ASCEND and DESCEND are irrelevant to the time interval. When TIMEBASE is set as other value than t#0s, ASCEND and DESCEND are the limit value of the time interval TIMEBASE.



RAMP_INT Function Block

3.2.6.1.2 Parameter Description

Input Parameter Description

Input Parameter	Data Type	Function Description	Initial Value (Default)	Power Fail Safeguard
IN	INT	Manipulated variable	0	FALSE
Ascend	INT	Maximum ascend value in time interval	0	TRUE
DEScend	INT	Maximum descend value in time interval	0	TRUE
TimeBase	TIME	Time interval	T#0S	TRUE
Reset	BOOL	Restart signal	FALSE	TRUE

Output Parameter Description

Output Parameter	Data Type	Function Description	Initial Value (Default)	Power Fail Safeguard
Out	INT	Value after ascend and descend limit	0	TRUE

Parameter Description of Intermediate Variable

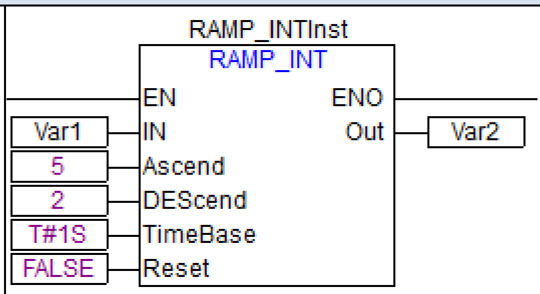
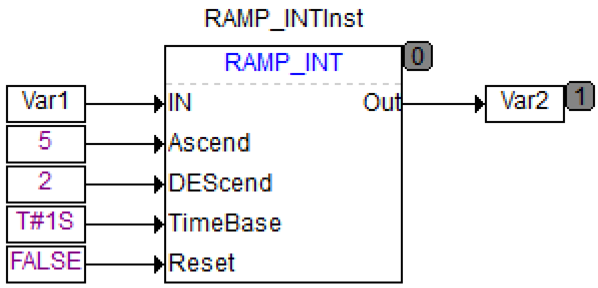
Intermediate Variable	Data Type	Function Description	Initial Value (Default)	Power Fail Safeguard
DIFF	DINT		0	FALSE
OLD_IN	INT		0	TRUE
TB	DWORD		0	FALSE
CET	DWORD		0	FALSE
CH	INT		0	FALSE
CLOCK	TON	Function block instance		TRUE

3.2.6.1.3 Application Example of Instructions

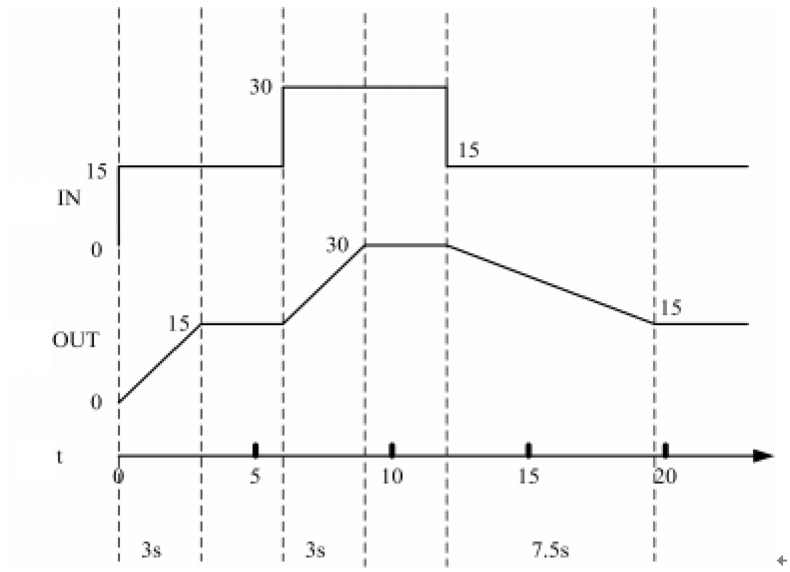
Variable Definition

No.	Variable Name	Address	Variable Description	Variable Type	Initial Value	Power Fail Safegua
0001	RAMP_INTInst			RAMP_INT		FALSE
0002	Var1			INT	0	FALSE
0003	Var2			INT	0	FALSE

Programing language	Program

Variable Definition	
Ladder diagram (LD)	
Structured text (ST)	RAMP_INTInst(IN:=Var1, Ascend:=5, DEScend:=2, TimeBase:=T#1S, Reset:=FALSE, OUT=>Var2);
Continuous Function Chart (CFC)	

The corresponding output value in above example is shown in figure according to the change of the input value IN when the instruction is executed.



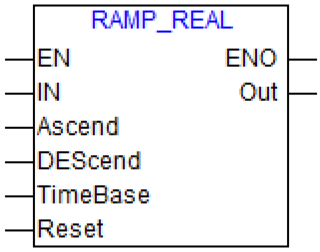
IN – OUT Relation

3.2.6.2 RAMP_REAL (Real Rate Limit)

3.2.6.2.1 Function

The RAMP_REAL instruction can limit the ascend and descend rate of the REAL input function.

OUT is the function value of IN after the setting of the ascend and descend limit. When TIMEBASE is set as t#0s, ASCEND and DESCEND are irrelevant to the time interval. When TIMEBASE is set as other value than t#0s, ASCEND and DESCEND are the limit value of the time interval TIMEBASE.



RAMP_REAL Function Block

3.2.6.2.2 Parameter Description

Input Parameter Description

Input Parameter	Data Type	Function Description	Initial Value (Default)	Power Fail Safeguard
IN	REAL	Manipulated variable	0	FALSE
Ascend	REAL	Maximum ascend value in time interval	0	TRUE
DEScend	REAL	Maximum descend value in time interval	0	TRUE
TimeBase	TIME	Time interval	T#0S	TRUE
Reset	BOOL	Restart signal	FALSE	TRUE

Output Parameter Description

Output Parameter	Data Type	Function Description	Initial Value (Default)	Power Fail Safeguard
Out	REAL	Value after ascend and descend limit	0	TRUE

Parameter Description of Intermediate Variable

Intermediate Parameter	Data Type	Function Description	Initial Value (Default)	Power Fail Safeguard
DIFF	REAL		0	FALSE
OLD_IN	REAL		0	TRUE
TB	DWORD		0	FALSE
CET	DWORD		0	FALSE
CLOCK	TON	Function block instance		TRUE

For the application example of instructions, refer to RAMP_INT (Integer Speed Limit).

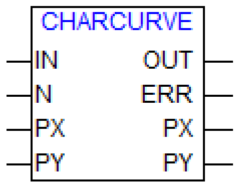
3.2.6.3 CHARCURVE (Characteristic Curve)

3.2.6.3.1 Function

IN is the DINT type, which will input the value to be operated. N is the BYTE type, which determines the points of the characteristic curve and 2=<N<=11. It will generate a characteristic line in the DINT type array PX[0..10] and PY[0..10]. The DINT type OUT is the value after the operation. The BYTE type variable ERROR will output the error message.

The point PX[0]..PX[N-1] in the array shall be stored from small to large. Otherwise, the ERR value will be 1. If IN is not between PX[0] and PX[N-1], then ERR=2, and the output includes corresponding limit value PY[0] or PY[N-1].

If N is out of the allowable value between 2 and 11, then ERR=4.



CHARCURVE Function Block

3.2.6.3.2 Parameter Description

Input Parameter Description

Input Parameter	Data Type	Function Description	Initial Value (Default)	Power Fail Safeguard
IN	DINT	Manipulated variable	0	FALSE
N	BYTE	The number of points of characteristic curve	0	TRUE

Output Parameter Description

Output Parameter	Data Type	Function Description	Initial Value (Default)	Power Fail Safeguard
OUT	DINT	Value after operation	0	TRUE
ERR	BYTE	Error message	0	FALSE

Parameter Description of Intermediate Variable

Other Variables	Data Type	Function Description	Initial Value (Default)	Power Fail Safeguard
PX	ARRAY[0..10] OF DINT	X axis point range of characteristic curve	0	TRUE
PY	ARRAY[0..10] OF DINT	Y axis point range of characteristic curve	0	TRUE

Detailed description:

Gradually display the leading function value.

IN is the DINT type, enter the value to be operated. N is the byte type, which determines the number of points of the characteristic curve and 2 = <N <= 11. Generates a feature line in the DINT type arrays PX

[0..10] and PY [0..10]. OUT is the value after operation. The byte variable ERR outputs an error message.

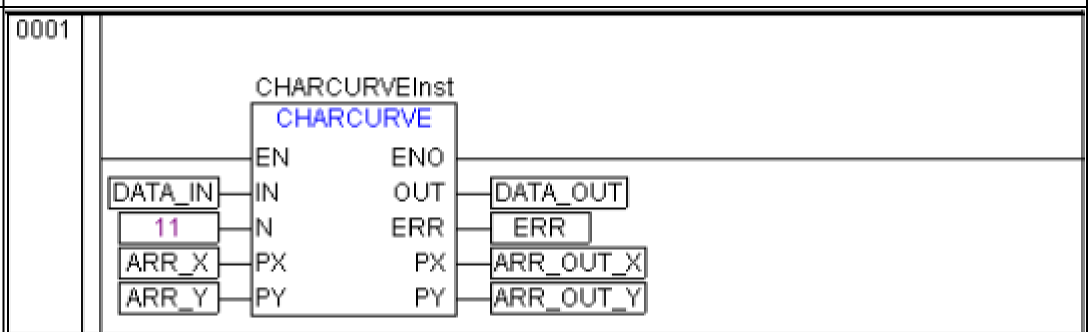
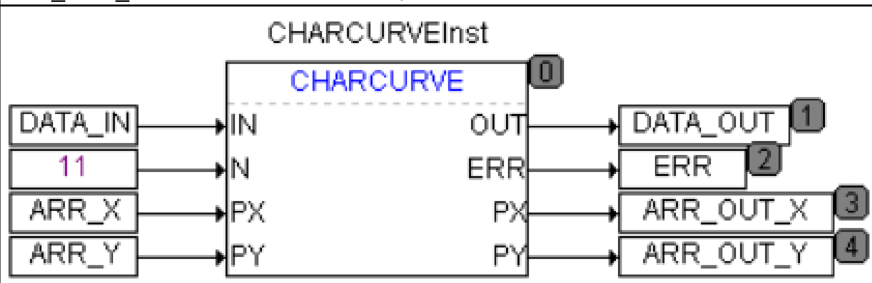
The point in the array PX [0] .. PX [N-1] must be stored from small to big, otherwise the value of ERR is 1. If IN is not between PX [0] and PX [N-1], ERR = 2, the output contains the corresponding limit value PY [0] or PY [N-1].

If N exceeds the allowable value between 2 and 11, ERR = 4.

3.2.6.3.3 Application Example of Instructions

Variable Definition

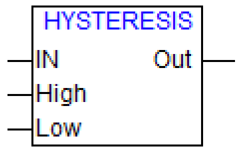
No.	Variable Name	Address	Variable Description	Variable Type	Initial Value	Power Fail Safeguai
0001	CHARCURVEInst			CHARCURVE		FALSE
0002	ERR			BYTE	0	FALSE
0003	DATA_IN			INT	0	FALSE
0004	DATA_OUT			INT	0	FALSE
0005	ARR_X			ARRAY[0..10] OF DINT		FALSE
0006	ARR_Y			ARRAY[0..10] OF DINT		FALSE
0007	ARR_OUT_X			ARRAY[0..10] OF DINT		FALSE
0008	ARR_OUT_Y			ARRAY[0..10] OF DINT		FALSE

Programing language	Program
Ladder diagram (LD)	
Structured text (ST)	CHARCURVEInst(IN:=DATA_IN, N:=11, PX:=ARR_X, PY:=ARR_Y, OUT=>DATA_OUT, ERR=>ERR); ARR_OUT_X:= CHARCURVEInst.PX; ARR_OUT_Y:= CHARCURVEInst.PY;
Continuous Function Chart (CFC)	

3.2.6.4 HYSTERESIS (Lag)

3.2.6.4.1 Function

When the input value is less than the lower limit, the output will be TRUE. When the input value is greater than the higher limit, the output will be FALSE. When the input value is between the lower limit and the higher limit, the output will hold.



HYSTERESIS Function Block

3.2.6.4.2 Parameter Description

Input Parameter Description

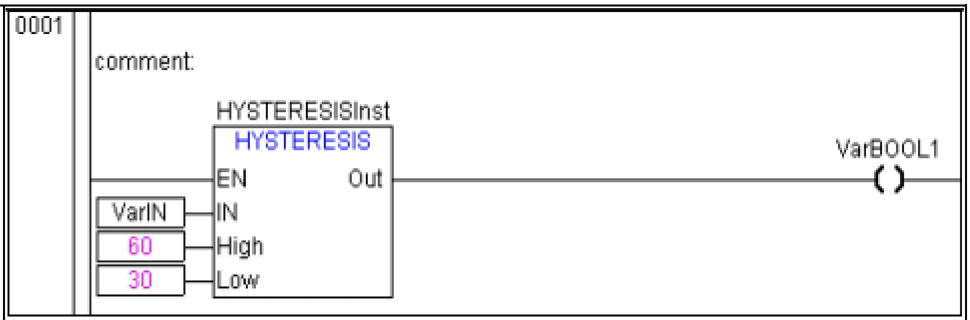
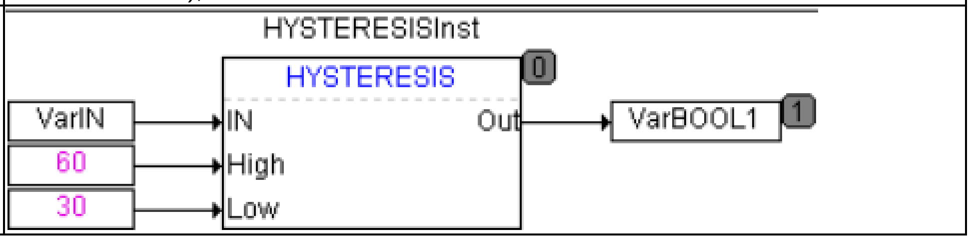
Input Parameter	Data Type	Function Description	Initial Value (Default)	Power Fail Safeguard
IN	INT	Input value	0	FALSE
Low	INT	Lower limit	0	TRUE
High	INT	Higher limit	0	TRUE

Output Parameter Description

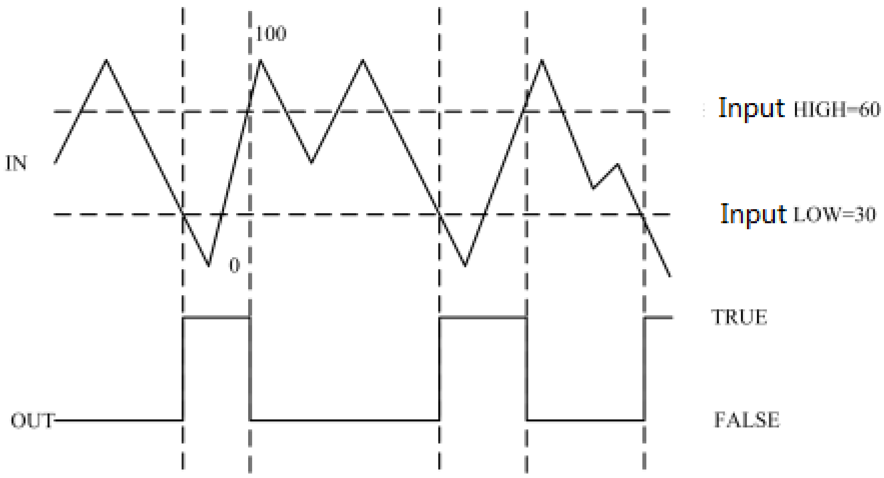
Output Parameter	Data Type	Function Description	Initial Value (Default)	Power Fail Safeguard
Out	BOOL	Output, which will be TRUE when it is less than the low threhold and will return to FALSE until it is greater than the higher limit.	FALSE	TRUE

3.2.6.4.3 Application Example of Instructions

Variable Definition						
0001	HYSTERESISInst			HEX_ENGIN		FALSE
0002	VarIN			INT	0	FALSE
0003	VarBOOL1			BOOL	FALSE	FALSE
Programing language	Program					

Ladder diagram (LD)	
Structured text (ST)	HYSTERESISInst(IN:=VarIN, High:=60, Low:=30, Out=>VarBOOL1);
Continuous function diagram (CFC)	

In the example above, the corresponding output value are shown in figure according to the change in the input value when the instruction is executed.

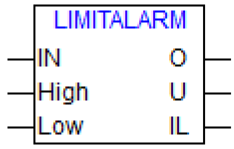


Change Relation of Input and Output

3.2.6.5 LIMITALARM (Limit Alarm)

3.2.6.5.1 Function

When the input value is less than the lower limit, the output U will be TRUE. When the input value is greater than the higher limit, the output O will be TRUE. When the input value is between the lower limit and the higher limit, the output IL will be TRUE.



LIMITALARM Function Block

3.2.6.5.2 Parameter Description

Input Parameter Description

Input Parameter	Function Description	Data Type	Initial Value (Default)	Power Fail Safeguard
IN	Input value	INT	0	FALSE
Low	Lower limit	INT	0	TRUE
High	Higher limit	INT	0	TRUE

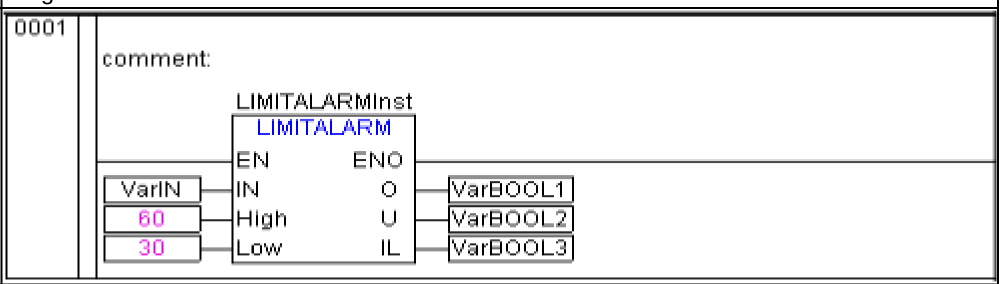
Output Parameter Description

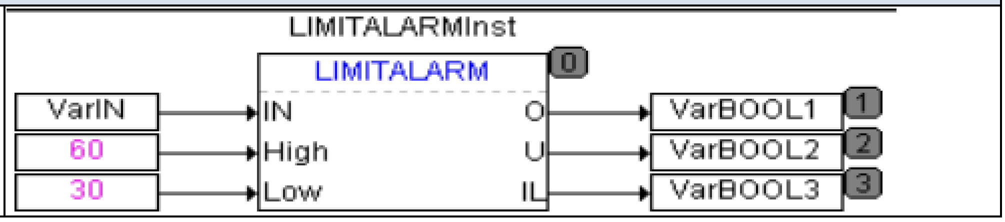
Output Parameter	Function Description	Data Type	Initial Value (Default)	Power Fail Safeguard
O	The input value is greater than the higher limit.	BOOL	FALSE	FALSE
U	The input value is less than the lower limit.	BOOL	FALSE	FALSE
IL	The input value is between the lower limit and the higher limit.	BOOL	FALSE	FALSE

3.2.6.5.3 Application Example of Instructions

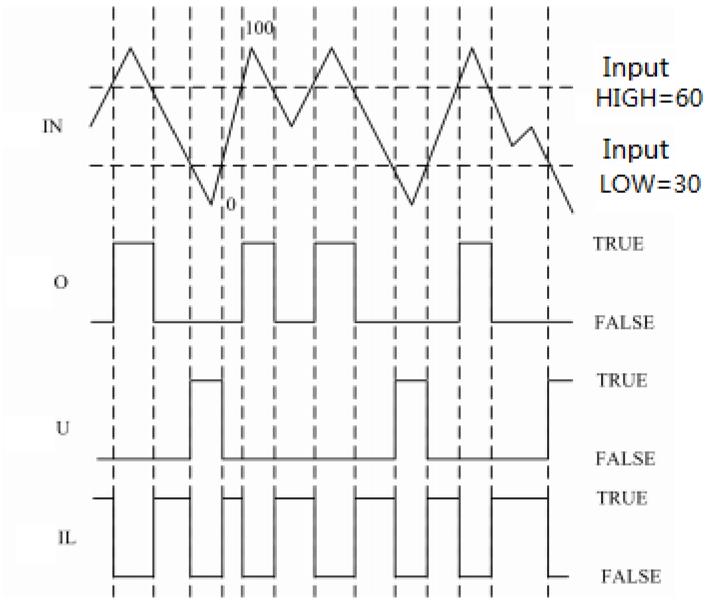
Variable Definition

No.	Variable Name	Address	Variable Description	Variable Type	Initial Value	Power Fail Safeguard
0001	VarIN			INT	0	FALSE
0002	VarBOOL1			BOOL	FALSE	FALSE
0003	VarBOOL2			BOOL	FALSE	FALSE
0004	VarBOOL3			BOOL	FALSE	FALSE

Programing language	Program
Ladder diagram (LD)	
Structured text (ST)	LIMITALARMInst(IN:=Var1, High:=60, Low:=30, O=>VarBOOL1, U=>VarBOOL2, IL=>VarBOOL3);

Variable Definition	
Continuous Function Chart (CFC)	

In the example above, the corresponding output value are shown in figure according to the change in the input value when the instruction is executed.

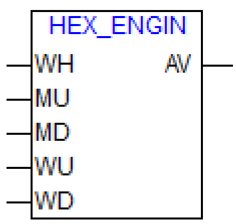


Change Relation of Input and Output

3.2.6.6 HEX_ENGIN (Hexadecimal Conversion to Engineering Data)

3.2.6.6.1 Function

The input code value WH is converted into corresponding engineering quantity AV.



HEX_ENGIN Function Block

3.2.6.6.2 Parameter Description

Input Parameter Description

Input Parameter	Description	Data Type	Initial Value (Default)	Power Fail Safeguard
WH	Input code value	DINT	0	FALSE
MU	Upper limit of engineering quantity	REAL	0	TRUE
MD	Lower limit of engineering quantity	REAL	0	TRUE
WU	Upper limit of analog input code value for PLC	DINT	0	TRUE
WD	Lower limit of analog input code value for PLC	DINT	0	TRUE

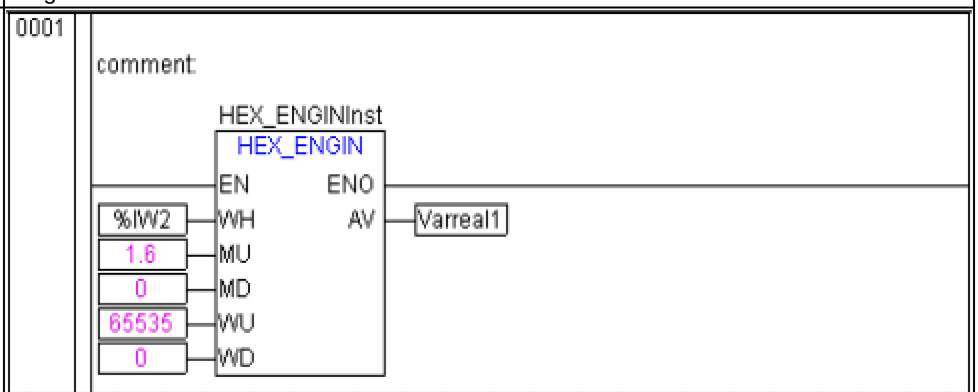
Output Parameter Description

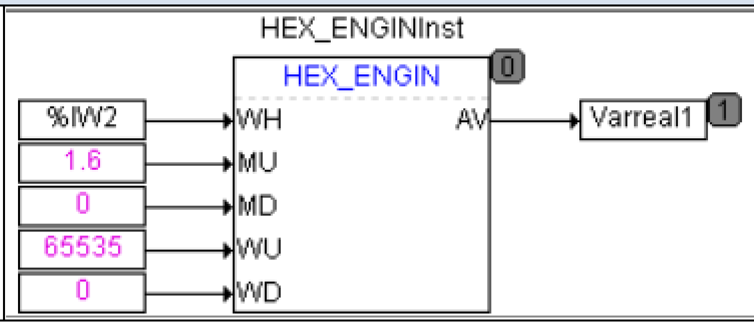
Output Parameter	Description	Data Type	Initial Value (Default)	Power Fail Safeguard
AV	Converted engineering quantity	REAL	0	FALSE

3.2.6.6.3 Application Example of Instructions

Variable Definition

No.	Variable Name	Address	Variable Description	Variable Type	Initial Value	Power Fail Safeguard
0001	HEX_ENGINInst			HEX_ENGIN		FALSE
0002	Varreal1			REAL	0	FALSE

Programing language	Program
Ladder diagram (LD)	
Structured text (ST)	HEX_ENGINInst(WH:=%IW2, MU:=1.6, MD:=0, WU:=65535, WD:=0, AV=>Varreal1);

Variable Definition	
Continuous function diagram (CFC)	

Program description:

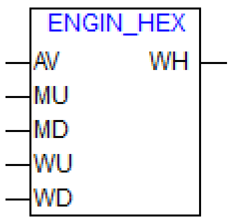
This instruction is usually used for the analog input data processing.

When EN is set, it will carry out the conversion, the input data will be %IW2, the upper limit of the engineering quantity is 1.6MPa, the lower limit of the engineering quantity is 0Mpa, and the code value range of corresponding module is 0-65535. We assume the corresponding output value is 0.799 when the value of %IW2 is 32767.

3.2.6.7 ENGIN_HEX (Engineering Data Conversion to Hexadecimal(WORD))

3.2.6.7.1 Function

The input engineering quantity is converted into corresponding code value WH.



ENGIN_HEX Function Block

3.2.6.7.2 Parameter Description

Input Parameter Description

Input Parameter	Description	Data Type	Initial Value (Default)	Power Fail Safeguard
WH	Input engineering quantity	REAL	0	FALSE
MU	Upper limit of engineering quantity	REAL	0	TRUE
MD	Lower limit of engineering quantity	REAL	0	TRUE
WU	Upper limit of analog input code value for PLC	WORD	0	TRUE

WD	Lower limit of analog input code value for PLC	WORD	0	TRUE

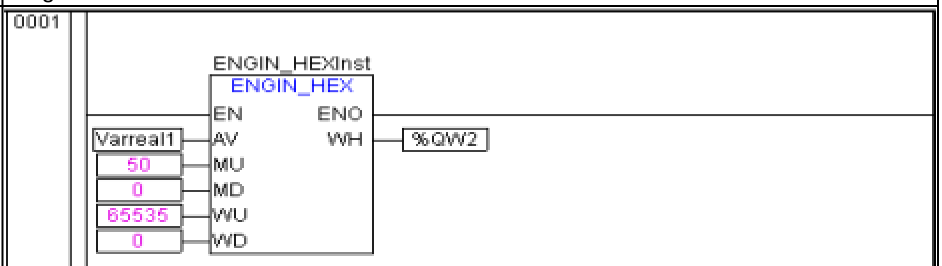
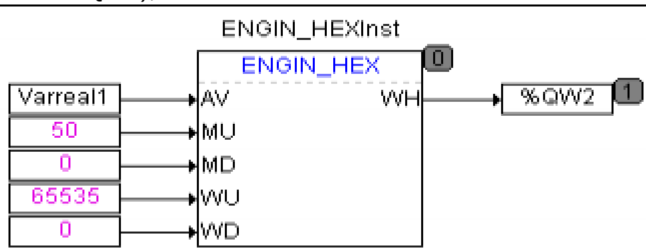
Output Parameter Description

Output Parameter	Description	Data Type	Initial Value (Default)	Power Fail Safeguard
WH	Converted code value	WORD	0	FALSE

3.2.6.7.3 Example for Use of Instructions

Variable Definition

No.	Variable Name	Address	Variable Description	Variable Type	Initial Value	Power Fail Safeguai
0001	ENGIN_HEXInst			ENGIN_HEX		FALSE
0002	Varreal1			REAL	0	FALSE

Programing language	Program
Ladder diagram (LD)	
Structured text (ST)	ENGIN_HEXInst(AV:=Varreal1, MU:=50 MD:=0, WU:=65535, WD:=0, WH=>%QW2);
Continuous function diagram (CFC)	

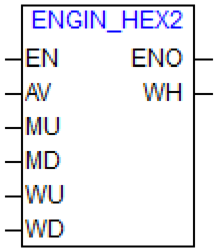
Program description:

This instruction is usually used for the analog output data processing. When EN is set, it will carry out the conversion, the actual data of the engineering quantity will be Varreal1, the upper limit of the engineering quantity is 50Hz, the lower limit of the engineering quantity is 0Hz, and the value range of corresponding module is 0-65535. When the value of Varreal1 is 25Hz, the corresponding output value will be 32768. If 32768 is assigned to %QW2, then this channel analog output value will be 25Hz.

3.2.6.8 ENGIN_HEX2 (Engineering Data Conversion to Hexadecimal(INT))

3.2.6.8.1 Function

The input engineering quantity is converted into corresponding code value WH.



ENGIN_HEX2 Function Block

3.2.6.8.2 Parameter Description

Input Parameter Description

Input Parameter	Description	Data Type	Initial Value (Default)	Power Fail Safeguard
WH	Input engineering quantity	REAL	0	FALSE
MU	Upper limit of engineering quantity	REAL	0	TRUE
MD	Lower limit of engineering quantity	REAL	0	TRUE
WU	Upper limit of analog input code value for PLC	INT	0	TRUE
WD	Lower limit of analog input code value for PLC	INT	0	TRUE

Output Parameter Description

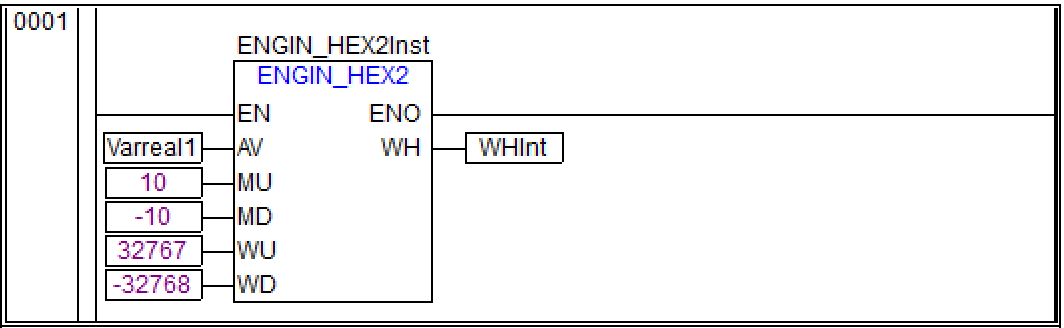
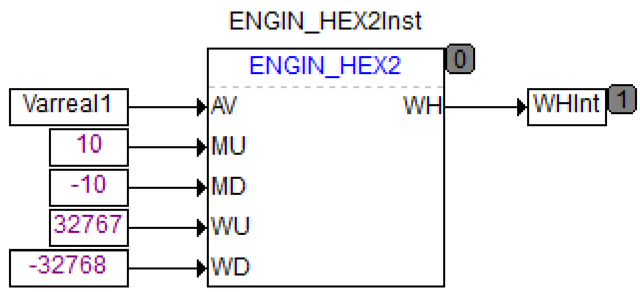
Output Parameter	Description	Data Type	Initial Value (Default)	Power Fail Safeguard
WH	Converted code value	INT	0	FALSE

3.2.6.8.3 Example for Use of Instructions

Variable Definition

No.	Variable Name	Address	Variable Description	Variable Type	Initial Value	Power Fail Safeguard	SOE Enable
0001	ENGIN_HEX2Inst			ENGIN_HEX2		FALSE	
0002	Varreal1			REAL	0	FALSE	
0003	WHInt			INT	0	FALSE	

Programing language	Program

<table>
<tr><td>Ladder diagram (LD)</td><td>
</td></tr>
<tr><td>Structured text (ST)</td><td>EENGIN_HEX2Inst(
AV:=Varreal1,
MU:=10
MD:=10,
WU:=32767,
WD:=-32768,
WH=>WHInt);</td></tr>
<tr><td>Continuous function diagram (CFC)</td><td>
</td></tr>
</table>

Program description:

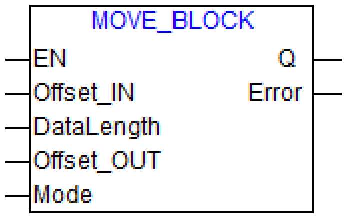
This instruction is usually used for the analog output data processing. When EN is set, it will carry out the conversion, the actual data of the engineering quantity will be Varreal1, the upper limit of the engineering quantity is 10V, the lower limit of the engineering quantity is -10V, and the value range of corresponding module is -32768-32767. When the value of Varreal1 is -5V, the corresponding output value will be -16384. If -16384 is assigned to WHInt(%QW4), then this channel analog output value will be -5V.

3.2.7 Data Block Move Functions

3.2.7.1 MOVE_BLOCK(Data Block Move)

3.2.7.1.1 Function

The data in the M area will be transferred to another location in the M area from one location in batch. It supports to transfer the data in batch by BYTE, WORD and DWORD. The maximum data transferred for one time may be up to 255 DWORD type data, and the transfer data length is determined by the transfer mode and the transfer data length jointly.



MOVE_BLOCK Function Block

3.2.7.1.2 Parameter Description

Input Parameter Description

Input Parameter	Data Type	Function Description	Parameter Description	Default Value
EN	BOOL	Enabled	0:Invalid 1:Enabled	
Offset_In	DWORD	Offset word address of data source area	Offset word address of data area	0
DataLength	BYTE	Transfer data length	0~255	0
Offset_Out	DWORD	Offset word address of data destination area	Offset word address of data area	0
Mode	BYTE	Data transfer mode	=0: Indicate the data is transferred by BYTE. =1: Indicate the data is transferred by WORD. =2: Indicate the data is transferred by DWORD.	0

Output Parameter Description

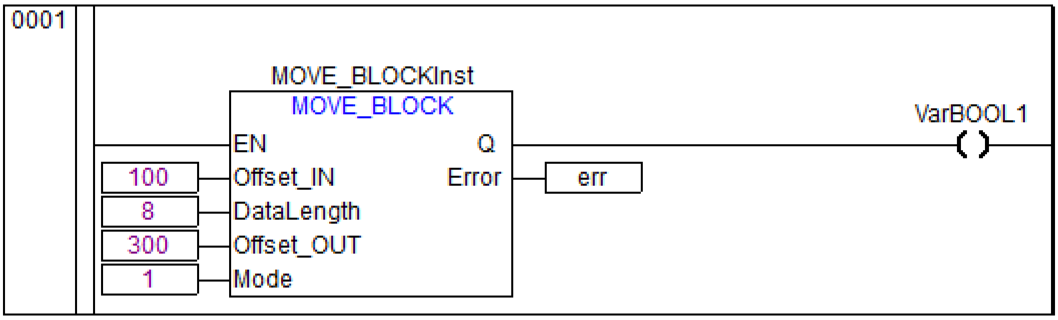
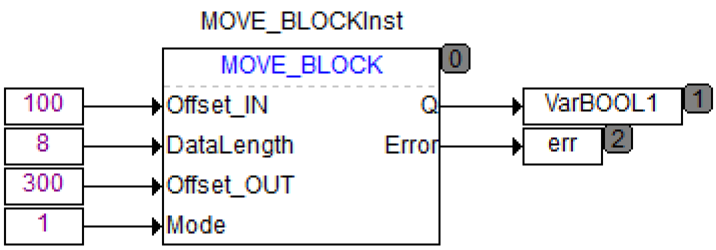
Output Parameter	Data Type	Function Description	Parameter Description
Q	BOOL	Enable output	When the data is not transferred/transferred incorrectly, Q=FALSE. When the data is transferred correctly, Q=TRUE.
Error	BYTE	Error message	=0, The data is transferred correctly. =1: Error in the offset address of the data source area (Offset_IN). =2: Error in the offset address of the data destination area (Offset_OUT). =3: Error in data transfer mode (Mode).

3.2.7.1.3 Application Example of Instructions

Variable Definition

No.	Variable Name	Address	Variable Description	Variable Type	Initial Value	Power Fail Safeguard
0001	MOVE_BLOCKInst			MOVE_BLOCK		FALSE
0002	err			BYTE	0	FALSE
0003	VarBOOL1			BOOL	FALSE	FALSE

Programing language	Program

Ladder diagram (LD)	
Structured text (ST)	MOVE_BLOCK(EN:=TRUE , Offset_IN:=100 , DataLength:=8 , Offset_OUT:=300 , Mode:=1 ,Q=>VarBOOL1 Error=> err);
Continuous Function Chart (CFC)	

Program description:

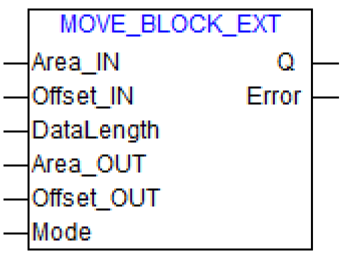
When EN is set, the data with N bytes that Offset_IN points to will be transferred to the destination address of Offset_OUT.

3.2.7.2 MOVE_BLOCK_EXT(Data Block Move Extension Instruction)

3.2.7.2.1 Function

Batch transfer the data from M/I/Q/S area to M/I/Q area.. It supports to transfer the data by BYTE, WORD and DWORD. The maximum data transferred for one time may be up to 255 DWORD type data.

and the transfer data length is determined by the transfer mode and the transfer data length jointly.



MOVE_BLOCK_EXT Function Block

3.2.7.2.2 Parameter Description

Input Parameter Description

Input Parameter	Data Type	Function Description	Parameter Description	Default Value
Area_IN	BYTE	Select source data area	Data area 0: M area 1: I area 2: Q area 3: S area	0
Offset_IN	DWORD	Offset address of source data area	Offset address of data area	0
DataLength	BYTE	Transfer data length	0~255	0
Area_OUT	BYTE	Select destination data area	Data area 0: M area 1: I area 2: Q area	0
Offset_OUT	DWORD	Offset address of destination data area	Offset address of data area	0
Mode	BYTE	Data transfer mode	=0: Indicate the data is transferred by BYTE. =1: Indicate the data is transferred by WORD. =2: Indicate the data is transferred by DWORD.	0

Output Parameter Description

Output Parameter	Data Type	Function Description	Parameter Description	Default Value
Q	BOOL	Data transfer results	When the data is not transferred/transferred incorrectly, Q=FALSE. When the data is transferred correctly, Q=TRUE.	FALSE
Error	BYTE	Error message	=0, The data is transferred correctly. =1: Error in selected source data area. =2: Error in the offset address of the source data area. =3: Error in selected destination data area. =4: Error in the offset address of thedestination data area. =5: Error in data transfer mode (Mode).	0

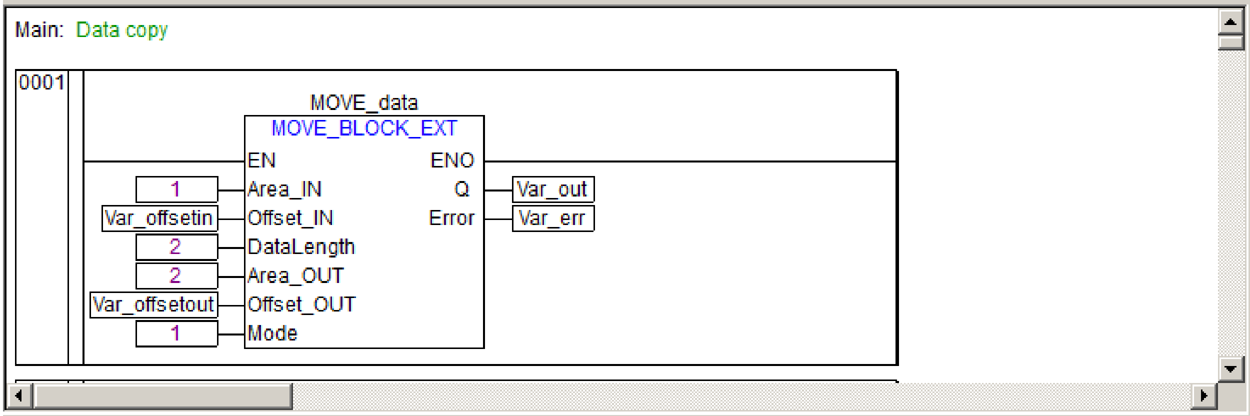
3.2.7.2.3 Application Example of Instructions

Copy the 4-byte data starting from address %Iw2 to address % QW4.

Set the data area, select WORD as transmission mode , set the initial value of the source data offset address Var_offsetin to 2, and the initial value of the destination data offset address Var_offsetout to 4. The configuration example is shown in the figure.

Main(PRG).ld

No.	Variable Name	Address	Variable Description	Variable Type	Initial Value	Power Fail Saf
0001	Var_offsetin		Offset address of source data	DWORD	2	FALSE
0002	Var_offsetout		Offset address of destination data	DWORD	4	FALSE
0003	MOVE_data		Data copy instance	MOVE_BLOCK_EXT		FALSE



Data Copy Configuration

Run the program, the data of %IW2~%IW4 is copied to %QW4~%QW6, as shown in the figure below.

LK411_1(2:LK411)

Device Configuration | Channel | Information

0_8 Channels AI

Channel Number	Channel Name	Channel Type	Online Value	Channel Address	Channel Description
1	DPIO_2_1_2_1	WORD	0	%IW0	
2	DPIO_2_1_2_2	WORD	5	%IW2	Source data
3	DPIO_2_1_2_3	WORD	4	%IW4	Source data
4	DPIO_2_1_2_4	WORD	0	%IW6	
5	DPIO_2_1_2_5	WORD	0	%IW8	
6	DPIO_2_1_2_6	WORD	0	%IW10	
7	DPIO_2_1_2_7	WORD	0	%IW12	
8	DPIO_2_1_2_8	WORD	0	%IW14	

LK411_1(2:LK411) LK510_1(3:LK510)

Device Configuration | Channel | Information

0_4 Channels AO

Channel Number	Channel Name	Channel Type	Online Value	Channel Address	Channel Description
1	DPIO_2_1_3_1	WORD	0	%QW0	
2	DPIO_2_1_3_2	WORD	0	%QW2	
3	DPIO_2_1_3_3	WORD	5	%QW4	Destination data
4	DPIO_2_1_3_4	WORD	4	%QW6	Destination data

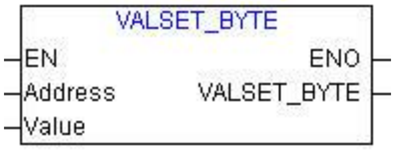
Runing Result

3.2.8 Memory Assignment Functions

3.2.8.1 VALSET_BYTE

3.2.8.1.1 Function

Modify the contents of address appointed by Address to the current value corresponding to Value.



VALSET_BYTE Function Block

ⓘThere is risk to use VALSET_BYTE instruction, during the process of using, please operate it strictly according to the requirements of instruction.

3.2.8.1.2 Parameter Description

Input Parameter Description

Input Parameter	Type of Data	Description of function	Description of Parameter Value
Address	POINTER TO BYTE	Target address	Point to the first address of definition BYTE data Range of operating data area : M, N, Q, R, special register S area %SB270 to %SB299
Value	BYTE	Set value	Set value

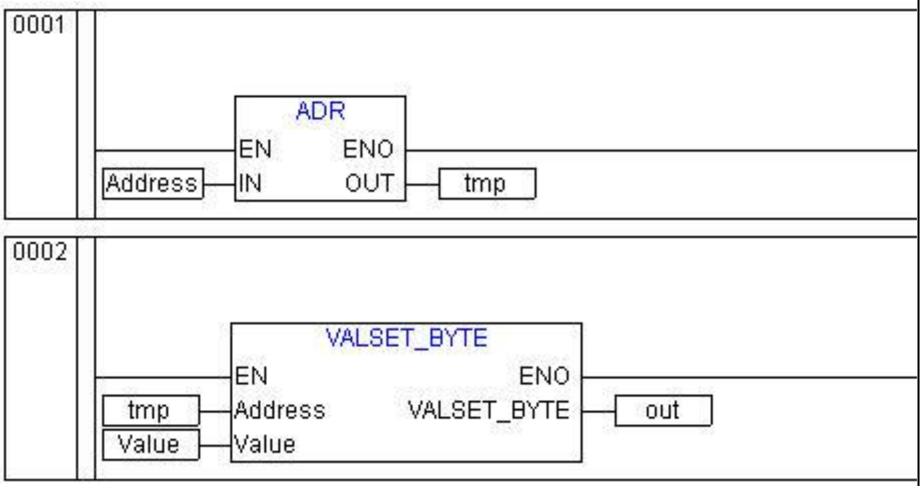
Output Parameter Description

Output Parameter	Type of Data	Description of Function	Description of Parameter Value
VALSET_BYTE	BYTE	Type of error	0: The output result is correct =1: The input parameter is wrong =2: Target segment address is I area =3: Target segment address is special register S area, the target segment address is not within the range of %SB270 to %SB299 =9: Length of copy data is wrong

3.2.8.1.3 Application Example of Instructions

Variable Definition

No.	Variable Name	Address	Variable Description	Variable Type	Initial Value	Power Fail Safeguard
0001	Address			BYTE	0	FALSE
0002	Value			BYTE	0	FALSE
0003	out			BYTE	0	FALSE
0004	tmp			POINTER TO BYTE	0	FALSE

Programming language	Program
Ladder Diagram（LD）	
Structured Text（ST）	out := VALSET_BYTE (ADR(Address),Value);

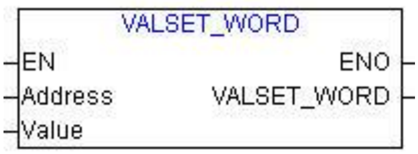
Variable Definition	
Continuous Function Chart （CFC）	

⚠When using the VALSET_BYTE instruction, if the space taken by the variable pointed by Address parameter is bigger than 1 BYTE, the instruction only can modify the contents that are less than 1 BYTE.

3.2.8.2 VALSET_WORD

3.2.8.2.1 Function

Modify the contents of the first two bytes address appointed by Address to the present value corresponding to Value.



VALSET_WORD Function Block

⚠

- There is risk to use VALSET_WORD instruction, during the process of using, please operate it strictly according to the requirements of instruction.
- When using instruction, if the space taken by the variable pointed by Address parameter is bigger than the length of the data regulated by the instruction, the instruction will modify the contents out of the address pointed by Address by accident and can make serious aftermath possibly.

3.2.8.2.2 Parameter Description

Input Parameter Description

Input Parameter	Type of Data	Description of function	Description of Parameter Value
Address	POINTER TO WORD	Target address	Point to the first address of definition WORD data Range of operating data area: M, Q, R, special register S area %SB270 to %SB299
Value	WORD	Set value	Set value

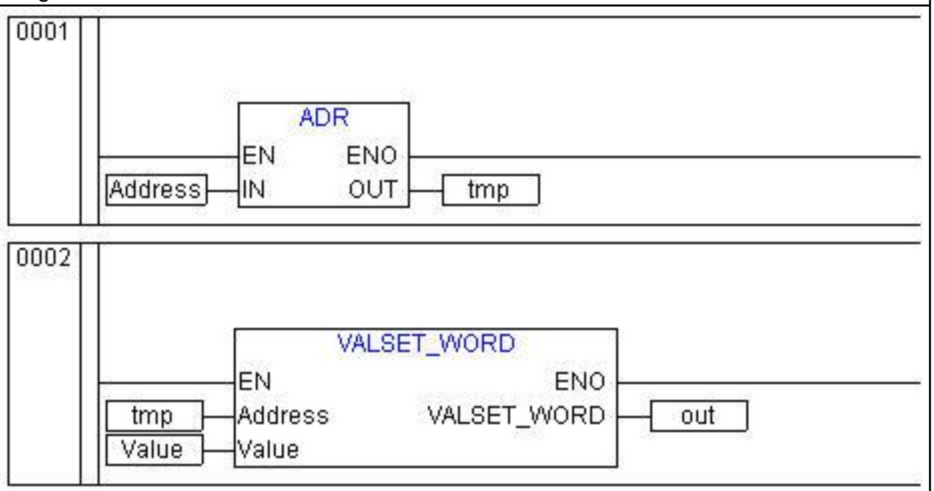
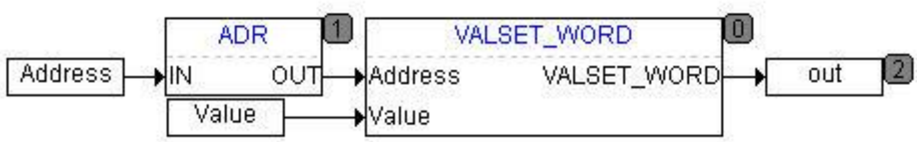
Output Parameter Description

Output Parameter	Type of Data	Description of function	Description of Parameter Value
VALSET_WORD	WORD	Type of error	=0: The output result is correct =1: The input parameter is wrong =2: The target segment address is I area =3: Target segment address is special register S area, the target segment address is not within the range of %SB270 to %SB299 =4: The target segment address is N area =5: The target segment address is M area, target segment address + length of copy database is over the range of M area =6: The target segment address is Q area, target segment address + length of copy database is over the range of Q area =8:The target segment address is R area, target segment address + length of copy database is over the range of R area =9: The length of copy data is wrong

3.2.8.2.3 Application Example of Instructions

Variable definition

No.	Variable Name	Address	Variable Description	Variable Type	Initial Value	Power Fail Safeguard
0001	Address	%MB100		WORD	0	FALSE
0002	Value			WORD	0	FALSE
0003	out			BYTE	0	FALSE
0004	tmp			POINTER TO WORD	0	FALSE

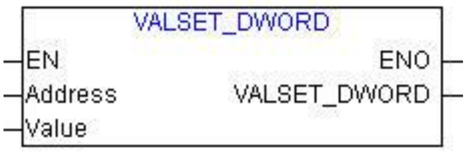
Programming language	Program
Ladder Diagram（LD）	
Structured Text（ST）	out := VALSET_WORD (ADR(Address),Value);
Continuous Function Chart（CFC）	

ⓘWhen using the VALSET_WORD instruction, if the space taken by the variable pointed by Address parameter is bigger than 1 WORD, the instruction only can modify the contents of 1 WORD minimally.

3.2.8.3 VALSET_DWORD

3.2.8.3.1 Function

Modify the contents of the first four bytes address appointed by Address to the current value corresponding to Value.



VALSET_DWORD Function Block

⚠

- There is risk to use VALSET_DWORD instruction, during the process of using, please operate it strictly according to the requirements of instruction.
- When using instruction, if the space taken by the variable pointed by Address parameter is bigger than the length of the data regulated by the instruction, the instruction will modify the contents out of the address pointed by Address by accident and can make serious aftermath possibly.

3.2.8.3.2 Instruction of Parameter

Input parameter	Type of data	Description of function	Description of parameter value
Address	POINTER TO DWORD	Target address	Point to the first address of definition DWORD data Range of operating data area : M, N, Q, R, special register S area %SB270 to %SB299
Value	DWORD	Set value	Set value

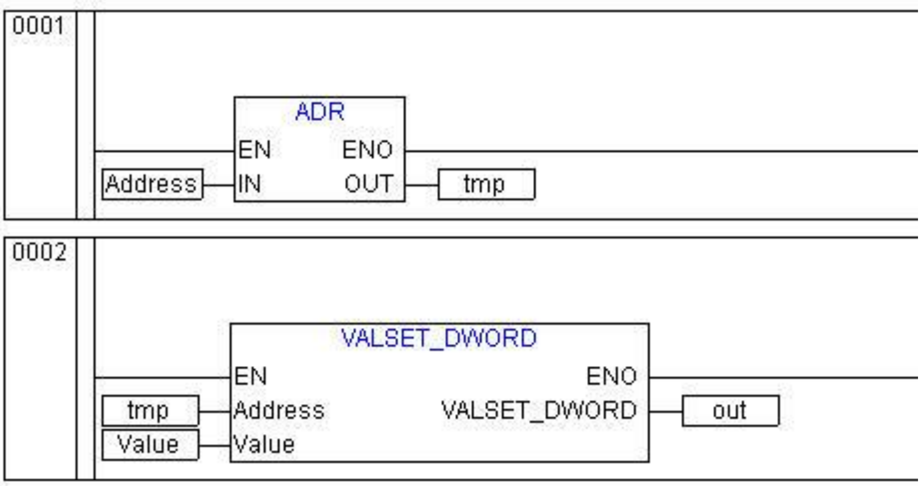
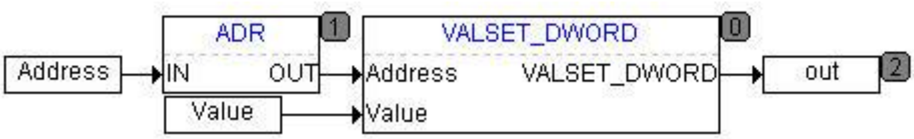
Output parameter	Type of data	Description of function	Description of parameter value
VALSET_DWORD	DWORD	Type of error	=0: The output result is correct =1: The input parameter is wrong =2: The target segment address is I area =3: Target segment address is special register S area, the target segment address is not within the range of %SB270 to %SB299 =4: The target segment address is N area =5: The target segment address is M area, target segment address + length of copy database is over the range of M area

Output parameter	Type of data	Description of function	Description of parameter value
			=6:The target segment address is Q area, target segment address + length of copy database is over the range of Q area =7: Target segment address is special register S area, the target segment address + length of copy database is not within the range of %SB270 to %SB299 =8: The target segment address is R area, target segment address + length of copy database is over the range of R area =9: The length of copy data is wrong

3.2.8.3.3 Application Example of Instructions

Variable Definition

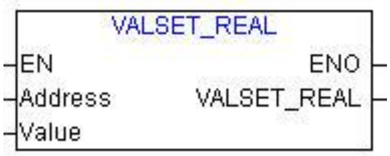
No.	Variable Name	Address	Variable Description	Variable Type	Initial Value	Power Fail Safeguard
0001	Address	%MB100		DWORD	0	FALSE
0002	Value			DWORD	0	FALSE
0003	out			BYTE	0	FALSE
0004	tmp			POINTER TO DWORD	0	FALSE

Programming language	Program
Ladder Diagram（LD）	
Structured Text（ST）	out:=VALSET_DWORD(ADR(Address),Value);
Continuous Function Chart（CFC）	

3.2.8.4 VALSET_REAL

3.2.8.4.1 Function

Modify the contents of the first four bytes address appointed by Address to the current value corresponding to Value.



VALSET_REAL Function Block

⚠

- There is risk to use VALSET_REAL instruction, during the process of using, please operate it strictly according to the requirements of instruction.
- When using instruction, if the space taken by the variable pointed by Address parameter is bigger than the length of the data regulated by the instruction, the instruction will modify the contents out of the address pointed by Address by accident and can make serious aftermath possibly.

3.2.8.4.2 Description of Parameter

Input Parameter	Type of Data	Description of Function	Description of Parameter Value
Address	POINTER TO REAL	Target address	Point to the first address of definition REAL data Range of operating data area : M, Q, R, special register S area %SB270 to %SB299
Value	REAL	Set value	Set value

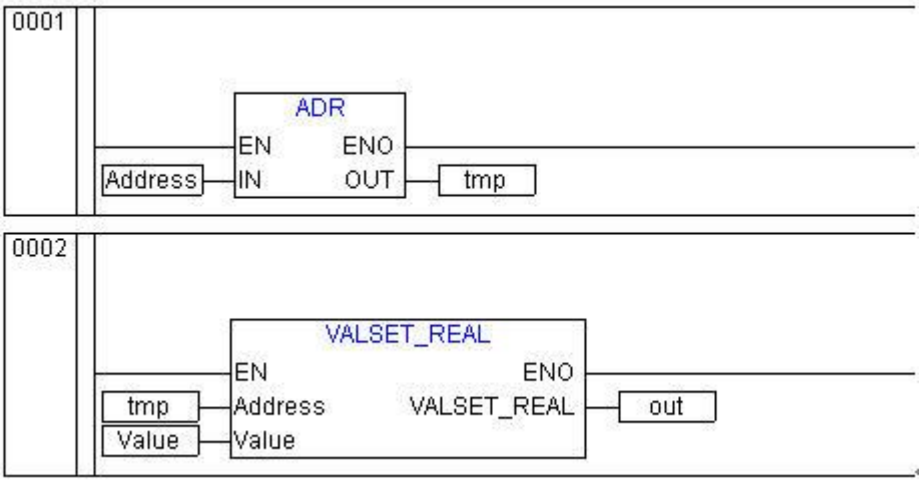
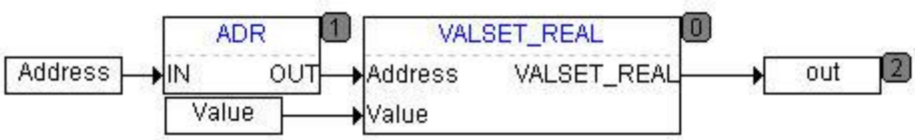
Output Parameter	Type of Data	Description of Function	Description of Parameter Value
VALSET_REAL	REAL	Type of error	=0: the output result is correct =1: the input parameter is wrong =2: the target segment address is I area =3: target segment address is special register S area, the target segment address is not within the range of %SB270 to %SB299 =4: the target segment address is N area =5: the target segment address is M area, target segment address + length of copy database is over the range of M area =6: the target segment address is Q area, target segment address + length of copy database is over the range of Q area =7: target segment address is special register S area, the target segment address + length of copy database is not within the range of %SB270 to %SB299 =8: the target segment address is R area, target segment address +

Output Parameter	Type of Data	Description of Function	Description of Parameter Value
			length of copy database is over the range of R area =9: The length of copy data is wrong

3.2.8.4.3 Application Example of Instructions

Variable Definition

No.	Variable Name	Address	Variable Description	Variable Type	Initial Value	Power Fail Safeguard
0001	Address	%MB100		REAL	0	FALSE
0002	Value			REAL	0	FALSE
0003	out			BYTE	0	FALSE
0004	tmp			POINTER TO REAL	0	FALSE

Programming language	Program
Ladder Diagram（LD）	
Structured Text（ST）	out := VALSET_REAL (ADR(Address),Value);
Continuous Function Chart（CFC）	

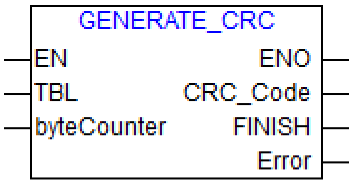
3.3 System Instruction

3.3.1 Communication Functions

3.3.1.1 GENERATE_CRC(Cyclic Redundancy Check)

3.3.1.1.1 Function

Generate 16-bits CRC checking value.



GENERATE_CRC Block

3.3.1.1.2 Parameter Description

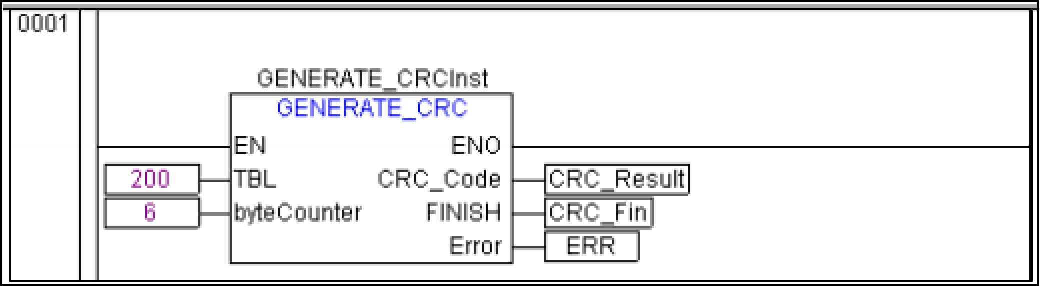
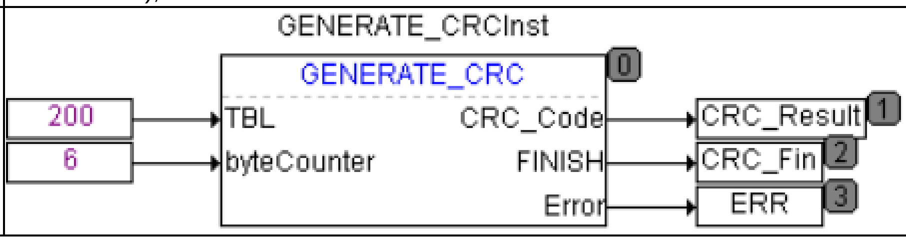
Input Parameter	Data Type	Parameter Description	Parameter Value Description
TBL	WORD	First byte address of data to be checked	Data is saved in address beginning from %MB200 if setting as 200
byteCounter	WORD	The number of bytes of data to be checked	0 is illegal value

Output Parameter	Data Type	Parameter Description	Parameter Value Description
CRC_Code	WORD	Result checked	16 bits
FINISH	BOOL	Whether completing checking	0: Checking not be finished 1: Checking is finished
Error	BYTE	Error information	0: Correct 1: The number of bytes of data to be checked byteCounter=0 2: Data to be checked exceed range of M area

3.3.1.1.3 Application Example of Instructions

Variable Definition

No.	Variable Name	Address	Variable Description	Variable Type	Initial Value	Power Fail Safeguard
0001	GENERATE_CRCInst			GENERATE_CRC		FALSE
0002	CRC_Result	%MB206		WORD	0	FALSE
0003	CRC_Fin			BOOL	FALSE	FALSE
0004	ERR			BYTE	0	FALSE

Programing language	Program
Ladder diagram (LD)	
Structured text (ST)	GENERATE_CRCInst(TBL:=200, byteCounter:=6, CRC_Code=>CRC_Result, FINISH=>CRC_Fin, Error=>ERR);
Continuous Function Chart (CFC)	

Program description:

The 6 bytes CRC check code starting from% MB200 is calculated by the CRC instruction. After that, according to the actual communication needs, the CRC check code to be sent after the other data sent out.

3.3.1.2 CANFilterEnable(Enable CAN filter parameter)

3.3.1.2.1 Function

Enable a certain set of hardware filter parameters, a maximum of 4 groups.

3.3.1.2.2 Parameter Description

Input Parameter	Type of Data	Description of Parameter Value
EN_R	BOOL	Ascending edge energy
Port	BYTE	PortID 0：CAN0

Input Parameter	Type of Data	Description of Parameter Value
Group	BYTE	Parameter group ordinal number, max 4 groups，0~3
MaskEXT	BOOL	Enable EXT entry, default TRUE
EXT	BOOL	TRUE: Extended frame; FALSE: Standard frame
MaskRTR	BOOL	Enable RTR items, default TRUE
RTR	BOOL	TRUE: Remote Frame; FALSE: Data Frame Remote Frame/Data Frame Filtering needs to be used in conjunction with other conditions and is not supported separately
MaskMsgID	BOOL	Enable MSGID entries, default TRUE
MsgID	DWORD	Message ID Standard Frame: Low 11 bit valid Extended Frame: Low 29 bit valid If the MaskEXT entry is not enabled, MSGID less than 2048 is processed in the standard frame format and greater than or equal to 2048 is processed in the extended frame format

Output Parameter	Type of Data	Description of Parameter Value
Done	BOOL	TRUE：Enable to succeed
Busy	BOOL	TRUE: Setting FALSE: Idle state
Error	BOOL	TRUE: Error generated during processing, see the corresponding error code through the Error ID entry FALSE: No errors
ErrorID	WORD	Error code 0: Provisioning succeeded 1: Wrong port number 2: Provisioning failed 3: Unsupported group ordinal

- Only MC series controllers (except MC1004L) are applicable.

3.3.1.3 TCP_CONNECT(Establish a TCP communication connection)

3.3.1.3.1 Function

Establish a TCP protocol communication connection.

Note: This function block is supported in FA-AT V3.1.10B2 and above versions, and compatible firmware version is LK220-G02,LK222-D02,LK224-D02.

- Applies to LK system,MC system and LE system LE5118-B01,LE5119-B01 and above on.LE5118, LE5119 support 5-way free port, maximum data transmission support 1K.

3.3.1.3.2 Parameter Description

	Data Item	Data Type	Parameter Description	Initial Value
Input	EN_R	BOOL	Rising edge: CPU initiates connection operation. When EN_R is FALSE, the output displays the current status of the connection	0

<table>
<tr><th></th><th>Data Item</th><th>Data Type</th><th>Parameter Description</th><th>Initial Value</th></tr>
<tr><td rowspan="8"></td><td>ConnectID</td><td>BYTE</td><td>Socket connection numbers
LK:1~16
LE:1~5
MC:1~5</td><td>0</td></tr>
<tr><td>Active</td><td>BOOL</td><td>TRUE: Create Active Client Connection
FALSE: Create passive server connection</td><td>0</td></tr>
<tr><td>IPaddr1</td><td>BYTE</td><td rowspan="4">Four byte IP address, from high to low, IPaddr1 to IPaddr4</td><td>0</td></tr>
<tr><td>IPaddr2</td><td>BYTE</td><td>0</td></tr>
<tr><td>IPaddr3</td><td>BYTE</td><td>0</td></tr>
<tr><td>IPaddr4</td><td>BYTE</td><td>0</td></tr>
<tr><td>RemotePort</td><td>WORD</td><td>The port number on the remote device. Remote port numbers range from 1 to 49151, for passive connections, use zero</td><td>0</td></tr>
<tr><td>LocalPort</td><td>WORD</td><td>The port number on the local device. The local port number range is from 1 to 49151, but there are some limitations. Please refer to the LocalPort parameter of Common_Parameters_of_Ethernet_Free_Protocol_Library_Instruction_the for details</td><td>0</td></tr>
<tr><td rowspan="9">Output</td><td>Done</td><td>BOOL</td><td>If the connection is active and ready to send or receive</td><td>0</td></tr>
<tr><td>Busy</td><td>BOOL</td><td>If the connection process is still in progress</td><td>0</td></tr>
<tr><td>Error</td><td>BOOL</td><td>If the connection is not available. ErrorID displays one of the error codes used to indicate an existing problem</td><td>0</td></tr>
<tr><td>ErrorID</td><td>WORD</td><td>If the Error is set to output, the ErrorID output will display an error code
If the Busy or Done is set to output, the ErrorID is zero (no error)
For more information, please refer to the
Error_Codes_of_Ethernet_Free_Protocol_Communication_Instruction_</td><td>0</td></tr>
<tr><td>LinkIPaddr1</td><td>BYTE</td><td rowspan="4">Four octets of the IP address of the connected remote device. LinkIPAddr1 is the most significant byte of the IP address, and LinkIPAddr4 is the least significant byte of the IP address</td><td rowspan="5">default hiding</td></tr>
<tr><td>LinkIPaddr2</td><td>BYTE</td></tr>
<tr><td>LinkIPaddr3</td><td>BYTE</td></tr>
<tr><td>LinkIPaddr4</td><td>BYTE</td></tr>
<tr><td>LinkPort</td><td>WORD</td><td>Connected remote device port number</td></tr>
</table>

The connection operation is asynchronous and may require several scans to complete. When the connection operation is pending, the Busy output value is TRUE. When the CPU completes the operation, the instruction is set to Done or Error output. If an error occurs, the ErrorID output will display an error code.

TCP_CONNECT must not be changed when the instruction is in a busy state. The CPU needs to rely on this to understand that this is the continuation of the call that initiates the connection process.

You assign a Connection ID (ConnectID) input to the connection, and then use this ConnectID to reference the connection when sending, receiving, or disconnecting.

The Active input bit determines whether this is an active connection (Active set to TRUE) or a passive connection (Active set to FALSE).

When the Active input is set to TRUE, the CPU creates an active connection (client), and the CPU attempts to contact and create a connection to the specified IP address and remote port number (RemotePort). The CPU opens the Local Port to receive messages from remote devices.

When Active input is set to FALSE, the CPU creates a passive (server) connection. In this case, the CPU opens the requested local port (LocalPort) and accepts connection requests from remote devices. If you want to accept connection requests from any remote IP address, you should set the IP address to 0.0.0.0. If the IP address is not zero, the CPU only accepts connection requests from the specified IP address. For passive connections, the CPU ignores the remote port number (RemotePort), which can be set to zero.

You can call TCP_ CONNECT instruction at any time determines the current state of the connection. Add EN_ R input set to FALSE and provide a valid Connection ID (ConnectID), TCP_ CONNECT returns the following:

- Busy, if the connection process is still in progress.
- Done, if the connection is active and ready to be sent or received.
- Error, if the connection is not available. ErrorID displays one of the error codes used to indicate an existing problem.
- LinkIPaddr1, output the first bit of the current connected device IP address
- LinkIPaddr2, output the second bit of the current connected device IP address
- LinkIPaddr3, output the 3rd bit of the current connected device IP address
- LinkIPaddr4, output the 4th bit of the current connected device IP address
- LinkPort, output the current connected device port number

Please note:

1. Active connection may take up to 30 seconds to determine whether the remote device is allowed to connect. Passive connection displays Busy status until the remote device attempts to connect to the CPU.

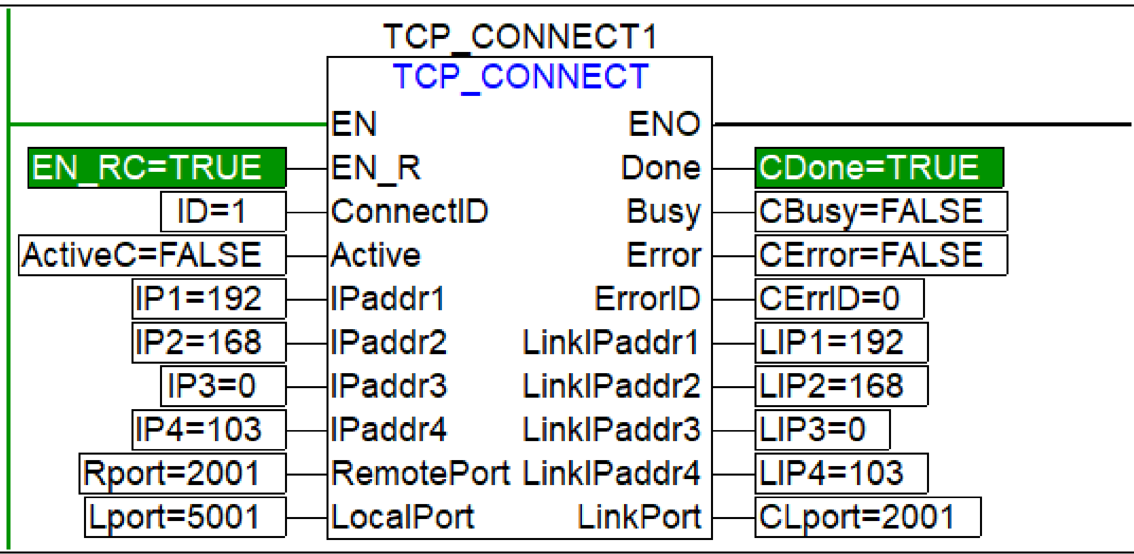
2. Actively connect to EN_ R remains at high power level, and if the remote device disconnects from the device, the CPU automatically attempts to reconnect to the remote device

3. After the connection is closed, the CPU will not automatically attempt to reconnect to the device, both active and passive connections.

3.3.1.3.3 Application Example of Instructions

Variable Definition

No.	Variable Name	Address	Variable Description	Variable Type	Online Value	Power Fail Safeguard
0001	TCP_CONNECT1			TCP_CONNECT		FALSE
0002	EN_RC			BOOL	TRUE	FALSE
0003	ID			BYTE	1	FALSE
0004	ActiveC			BOOL	FALSE	FALSE
0005	IP1			BYTE	192	FALSE
0006	IP2			BYTE	168	FALSE
0007	IP3			BYTE	0	FALSE
0008	IP4			BYTE	103	FALSE
0009	Report			WORD	2001	FALSE
0010	Lport			WORD	5001	FALSE
0011	CDone			BOOL	TRUE	FALSE
0012	CBusy			BOOL	FALSE	FALSE
0013	CError			BOOL	FALSE	FALSE
0014	CErrID			WORD	0	FALSE
0015	LIP1			BYTE	192	FALSE
0016	LIP2			BYTE	168	FALSE
0017	LIP3			BYTE	0	FALSE
0018	LIP4			BYTE	103	FALSE
0019	CLport			WORD	2001	FALSE

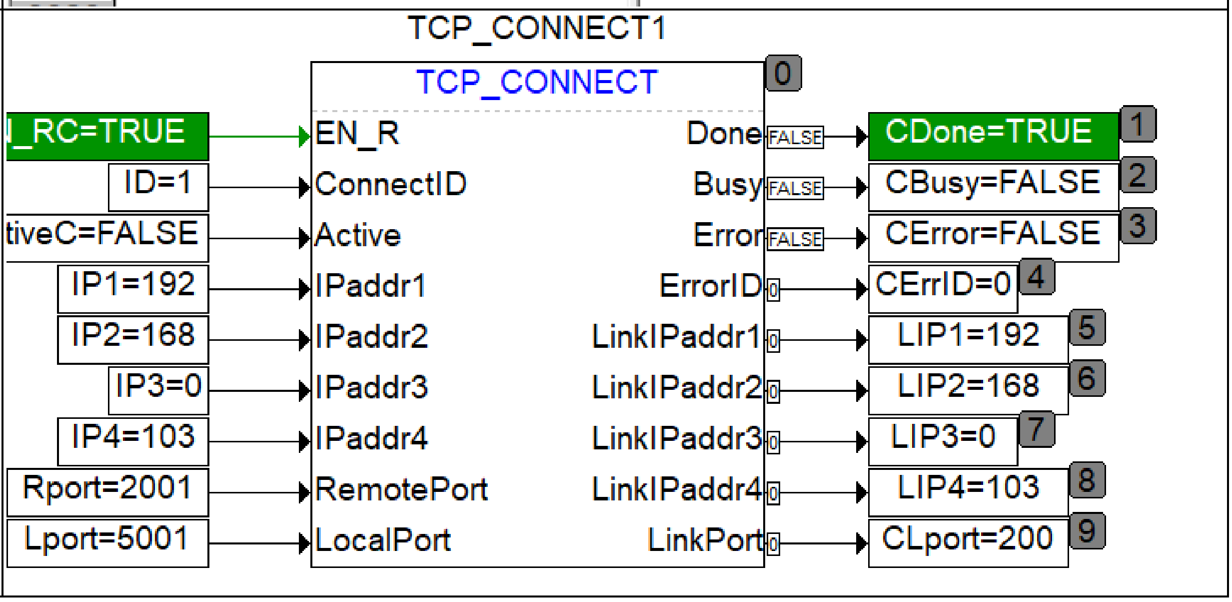
Programing language	Program
Ladder diagram (LD)	

Structured text（ST）

```
TCP_CONNECT1(
EN_R:=EN_RC,
ConnectID:=ID,
Active:=ActiveC,
IPaddr1:=IP1,
IPaddr2:=IP2,
IPaddr3:=IP3,
IPaddr4:=IP4,
RemotePort:=Rport,
LocalPort:=Lport,
Done=>CDone,
Busy=>CBusy,
Error=>CError,
ErrorID=>CErrID,
LinkIPaddr1=>LIP1,
LinkIPaddr2=>LIP2,
LinkIPaddr3=>LIP3,
LinkIPaddr4=>LIP4,
LinkPort=>CLport);
```

```
TCP_CONNECT1
EN_RC = TRUE
ID = 1
ActiveC = FALSE
IP1 = 192
IP2 = 168
IP3 = 0
IP4 = 103
Rport = 2001
Lport = 5001
CDone = TRUE
CBusy = FALSE
CError = FALSE
CErrID = 0
LIP1 = 192
LIP2 = 168
LIP3 = 0
LIP4 = 103
CLport = 2001
```

Continuous Function Chart （CFC）



3.3.1.4 TCP_SEND(Send data via tcp protocol)

3.3.1.4.1 Function

Use TCP protocol for Ethernet data transmission.

Note: This function block is supported in FA-AT V3.1.10B2 and above versions, and compatible firmware version is LK220-G02,LK222-D02,LK224-D02.

- Applies to LK system,MC system and LE system LE5118-B01,LE5119-B01 and above on.LE5118, LE5119 support 5-way free port, maximum data transmission support 1K.

3.3.1.4.2 Parameter Description

	Data Item	Data Type	Parameter Description	Initial Value
Input	EN_R	BOOL	Rising edge: CPU starts sending operation	0
	ConnectID	BYTE	ConnectID is the number of the connection used for this send operation. Using your ConnectID selected for the TCP_CONNECT operation	0
	DataLen	WORD	Request to send data length LK:1-8K LE:64K(LE5118/LE5119 is 1K) MC:64K	0
	DataAddr	POINTER TO BYTE	DataAddr is a pointer to the location where the data is stored for sending	0
Output	Done	BOOL	When the send operation is completed and there are no errors, the Done is set to output	0
	Busy	BOOL	When the send operation is in progress, the Busy is set to output	0
	Error	BOOL	When the send operation is completed but an error occurs, the Error is set to output	0
	ErrorID	WORD	If the Error is set to output, the ErrorID output will display an error code. If the instruction is set to Busy or Done output, the ErrorID is zero (no error) For more information, please refer to the Error_Codes_of_Ethernet_Free_Protocol_Communication_Instruction_	0
	SentLen	WORD	SendLen is the actual number of bytes sent. SendLen is only valid when the instruction is set to Done or Error output. If the instruction is set to Done output, the entire message information is sent	0

When the following situations occur, The TCP_SEND instruction initiates the operation of sending a specified number of bytes:

- The program calls the instruction through the EN_R is set to TRUE .
- The connection is not used to perform other send operations.

The EN_ R input uses a rising edge trigger so that the instruction does not initiate unexpected send operations. When TCP_SEND is in a busy state, the program ignores EN_ R input. The Done, Busy, Error, and output bytes of ErrorID displays the TCP_ SEND status or each calling .

After the send operation is completed, the instruction displays Done or Error status after calling TCP_SEND once, TCP_ SEND responds with an error code (error description information: no pending action),(if called by setting EN-R input to FALSE). If the EN_R input is set to TRUE, the program will initiate another send operation. The following describes the relationship between input and output parameters

- EN_R is set to TRUE to start the message sending operation. Busy is set to TRUE.
- Message sending completed. Done is set, Busy is cleared to zero.
- EN_ R is FALSE, no message sending operation is in progress.
- EN_ R is set to TRUE again, thus starting another message sending operation. Busy is set to

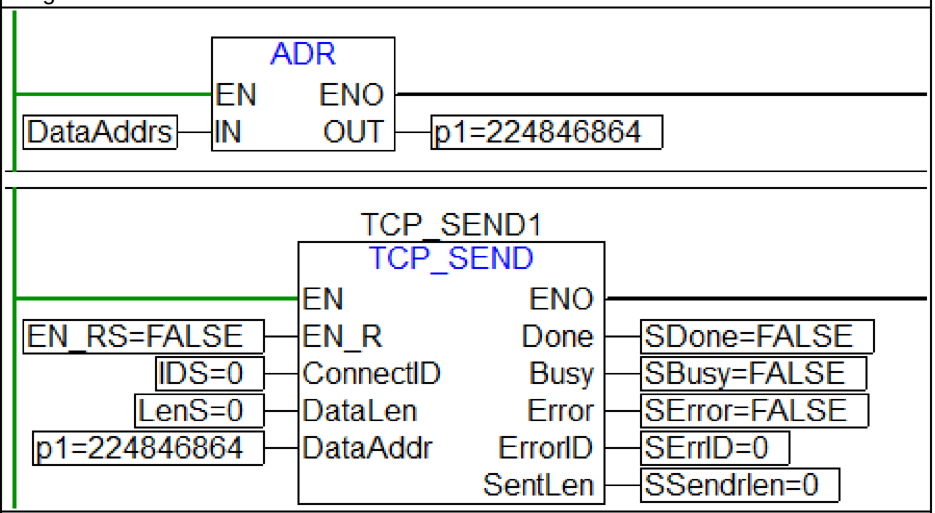
TRUE.

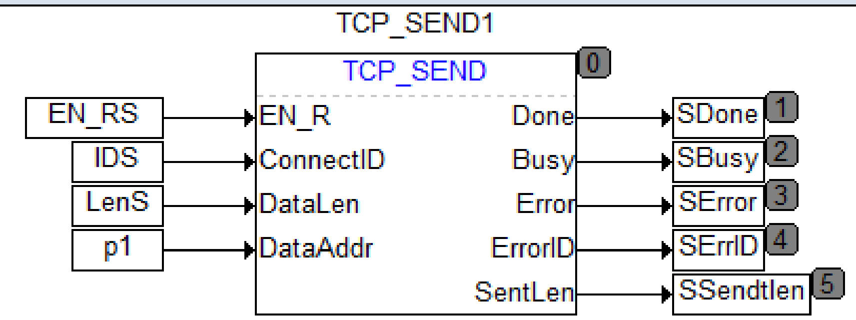
- Message sending completed. Done is set and Busy is reset within one scan cycle.

3.3.1.4.3 Application Example of Instructions

Variable Definition

No.	Variable Name	Address	Variable Description	Variable Type	Online Value	Power Fail Safeguard
0001	DataAddrs			ARRAY[0..1024] OF BOOL		FALSE
0002	p1			POINTER TO BYTE	224846864	FALSE
0003	TCP_SEND1			TCP_SEND		FALSE
0004	EN_RS			BOOL	FALSE	FALSE
0005	IDS			BYTE	0	FALSE
0006	LenS			WORD	0	FALSE
0007	SDone			BOOL	FALSE	FALSE
0008	SBusy			BOOL	FALSE	FALSE
0009	SError			BOOL	FALSE	FALSE
0010	SErrID			WORD	0	FALSE
0011	SSendrlen			WORD	0	FALSE

Programing language	Program
Ladder diagram（LD）	
Structured text（ST）	TCP_SEND1(EN_R:=EN_RS, ConnectID:=IDS, DataLen:=LenS, DataAddr:=p1, Done=>SDone, Busy=>Sbusy, Error=>SError, ErrorID=>SErrID, SentLen=>SSendtlen);

Variable Definition	
Continuous Function Chart (CFC)	

3.3.1.5 TCP_RECV(Recv data via tcp protocol)

3.3.1.5.1 Function

Use TCP protocol for Ethernet data reception.

Note: This function block is supported in FA-AT V3.1.10B2 and above versions, and compatible firmware version is LK220-G02,LK222-D02,LK224-D02.

- Applies to LK system,MC system and LE system LE5118-B01,LE5119-B01 and above on.LE5118, LE5119 support 5-way free port, maximum data transmission support 1K.

3.3.1.5.2 Parameter Description

	Data Item	Data Type	Parameter Description	Initial Value
Input	IN	BOOL	Continuous reception when high level is enable	0
	ConnectID	BYTE	ConnectID is the number of the connection used for this send operation. Using your ConnectID selected for the TCP_CONNECT operation	0
	MaxLen	WORD	MaxLen is the maximum number of bytes to receive (for example, the size of the buffer in DataPtr (1 to 8K))	0
	DataAddr	POINTER TO BYTE	DataAddr is a pointer to the receiving data storage location	0
Output	Done	BOOL	When the receive operation is completed and there are no errors, Done is set to output. Done is set to output, the RcvdLen output is valid	0
	Busy	BOOL	When the receive operation is in progress,Busy is set to output	0
	Error	BOOL	When the receive operation is completed but an error occurs, Error is set to output	0
	ErrorID	WORD	If the Error is set to output, the ErrorID output will display an error code. If the instruction is set to Busy or Done output, the ErrorID is zero (no error) For more information, please refer to the Error_Codes_of_Ethernet_Free_Protocol_Communication_Instruction_	0
	RcvdLen	WORD	RcvdLen is the actual number of bytes received. RcvdLen is only valid when the Done or Error is set to output. If the Done is set to output, the instruction receives the entire message. If the Error is set to output, the message exceeds the buffer size (MaxLen) and is truncated	0

The TCP_ RECV instruction has an IN (enable) input and is triggered by a high level. After the TCP_RECV instruction is executed firstly, the status bit shows that the instruction is in a busy state. TCP_RECV will display a busy state until the CPU receives data through the specified connection.

After the CPU receives messages through the specified connection, When using the TCP_RECV command at next time, the following tasks are executed:

- Copy the message data to the program's data area (DataAddr).
- Set the RcvdLen output to the number of bytes received.
- Set the Done output, clear the Busy and Error outputs, and set the ErrorID output byte value to zero.

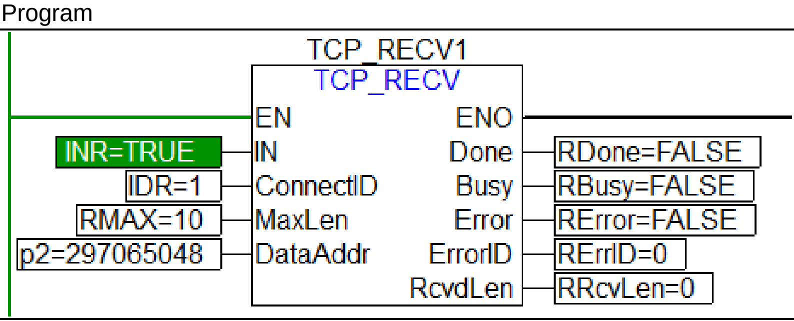
When using the TCP protocol, The TCP_RECV instruction returns all bytes received by the CPU through the specified connectionthe after last calling TCP_RECV instruction. The program must call TCP_RECV instruction enough frequently , because TCP acts as a "flow" protocol. There is no message description (no start or end tag) in the TCP protocol. Therefore, the CPU does not know when the message starts or ends.

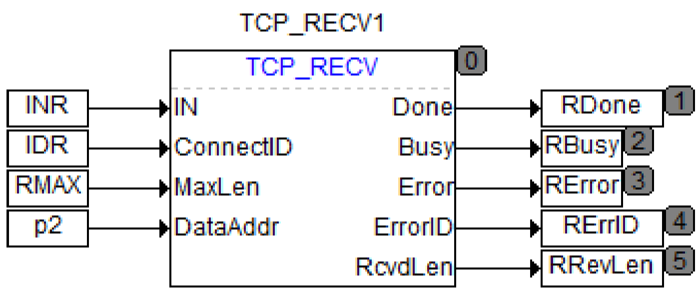
For example, let's assume that there is a TCP client that continuously sends 10 10 10 byte messages to the CPU, and during this time, the program does not call TCP_ RECV instruction. The program calls TCP_RECV instruction after the CPU accepts all 10 messages. The TCP_RECV instruction returns this data as a receive message of 100 bytes. The program is responsible for calling TCP_ RECV frequently enough to receive each message as it was sent.

3.3.1.5.3 Application Example of Instructions

Variable Definition

No.	Variable Name	Address	Variable Description	Variable Type	Online Value	Power Fail Safeguard
0001	TCP_RECV1			TCP_RECV		FALSE
0002	INR			BOOL	TRUE	FALSE
0003	IDR			BYTE	1	FALSE
0004	RMAX			WORD	10	FALSE
0005	p2			POINTER TO BYTE	297065048	FALSE
0006	RDone			BOOL	FALSE	FALSE
0007	RBusy			BOOL	FALSE	FALSE
0008	RError			BOOL	FALSE	FALSE
0009	RErrID			WORD	0	FALSE
0010	RRcvLen			WORD	0	FALSE

Programing language	Program
Ladder diagram （LD）	
Structured text （ST）	TCP_RECV1(IN:=INR, ConnectID:=IDR, MaxLen:=RMax,

<table>
<tr><th colspan="2">Variable Definition</th></tr>
<tr><td></td><td>DataAddr:=p2,
Done=>RDone,
Busy=>Rbusy,
Error=>RError,
ErrorID=>RErrID,
RcvdLen=>RRevLen);</td></tr>
<tr><td>Continuous Function Chart（CFC）</td><td>
</td></tr>
</table>

3.3.1.6 ETH_DISCONNECT(Close the Ethernet communication connection)

3.3.1.6.1 Function

ETH_ The DISCONNECT command is used to terminate an existing communication connection.

Note: This function block is supported in FA-AT V3.1.10B2 and above versions, and compatible firmware version is LK220-G02.

- Applies to LK system,MC system and LE system LE5118-B01,LE5119-B01 and above on.LE5118,LE5119 support 5-way free port, maximum data transmission support 1K.

3.3.1.6.2 Parameter Description

	Data Item	Data Type	Parameter Description	Initial Value
Input	EN_R	BOOL	Rising edge triggering to close connection	0
	ConnectID	BYTE	The CPU uses the Connection ID to identify the connection to terminate (defined during the connection process)	0
Output	Done	BOOL	When the disconnection operation is completed and there are no errors, Done is set to output	0
	Busy	BOOL	When the disconnection is in progress, Busy is set to output	0
	Error	BOOL	When the disconnection operation is completed but an error occurs, Error is set to output	0
	ErrorID	WORD	If the Error is set, the ErrorID output will display an error code. If the Busy or Done is set to output, the ErrorID is zero (no error) For more information, please refer to Error_Codes_of_Ethernet_Free_Protocol_Communication_Instruction_	0

3.3.1.6.3 Common Parameters of Ethernet Free Protocol Library Instruction

Common Parameters	Parameter Description
EN_R	Input is used to initiate the operation. Triggered by level. EN_R is connected to the library instruction by the rising edge command so that the operation can only be initiated once. When the instruction is Busy, the program will ignore EN_ R input
Active	Active input is used to specify whether the connection instruction creates an active client connection (Active=TRUE) or a passive server connection (Active=FALSE). In active connection, the local CPU initiates communication to the remote device. In passive connection, the local CPU waits for the remote device to initiate communication
Done	When the operation is completed and there are no errors, the OUC instruction is set to Done output. If the instruction is set to Done output, Busy, Error, and ErrorID outputs are zero. Only when Done output is set, other outputs (such as the number of bytes received) are valid
Busy	Busy output indicates that an operation is in progress. Initiate the operation when EN_ R is set to TRUE , the OUC instruction is set to Busy output. For all subsequent calls to the instruction, the Busy output remains set until the operation is completed
Error	The Error output indicates that the operation is completed with errors. If the OUC instruction is set to Error output, the Done and Busy outputs will be set to FALSE. If the instruction is set to Error output, the ErrorID output will indicate the cause of the error. If the Error output is set, all other outputs are invalid
ConnectID	The ConnectID number is the identifier of the connection. Through TCP_CONNECT creates a connection, it creates a ConnectID that can be selected from any value in the range of 1 to 5. Each connection must have a unique ConnectID. The program uses ConnectID to specify the connections required for subsequent send, receive, and disconnect operations
IPaddr1~IPaddr4	These are the four bytes of the remote device IP address. IPaddr1 is the most significant byte of the IP address, while IPaddr4 is the least significant byte of the IP address. For example, for IP address 192.168.2.15, set the following values: IPaddr1=192 IPaddr2=168 IPaddr3=2 IPaddr4=15 The IP address cannot be the following value: 0.0.0.0 (for active connections) Any broadcast IP address (e.g. 255.255.255.255) Any multicast address The IP address of the local CPU IP address 0.0.0.0 can be used for passive connections. By selecting IP address 0.0.0.0, the CPU accepts connections from any remote IP address. If a non zero IP address is selected for passive connections, the CPU will only accept connections from the specified address
RemotePort	The port number on the remote device. The port number can be used for TCP and UDP protocols to route messages within the device The rules for remote port numbers are as follows: The valid port number range is 1 to 49151 The recommended port number range is 2000 to 49151 For passive connections, the CPU ignores the remote port number (which can be set to zero)
LocalPort	It is the port number on the local CPU. The port number can be used for TCP and UDP protocols to route messages within the device. For all passive connections, the local port number must be unique The rules for local port numbers are as follows: The valid port number range is 1 to 49151 Cannot use port numbers 20, 21, 25, 80, 102, 135, 161, 162, 443, 1200, and 34962 to 34964. These ports have specific purposes The recommended port number range is 2000 to 49151 For passive connections, the local port number must be unique (not duplicated)

3.3.1.6.4 Error Codes of Ethernet Free Protocol Communication Instruction

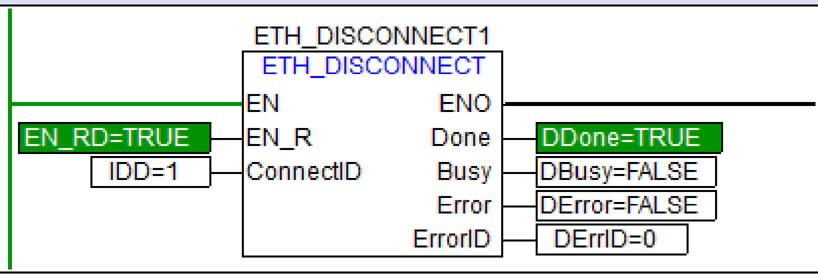
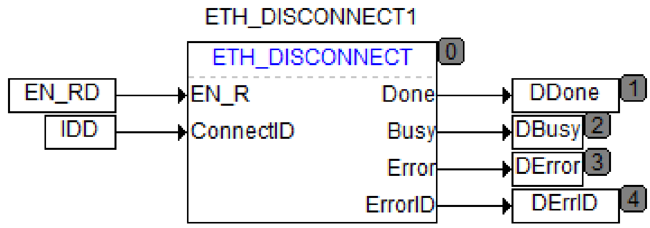
Error code	Error message description
0	No errors
1	Input ConnectID error
2	Illegal input of IP address
3	The remote port number entered is illegal
4	The local port number entered is illegal
5	When using TCP_CONNECT command to connect, the input parameters change
6	Attempting to use TCP_CONNECT command creates an active connection, where the input information of this connection is duplicated with that of other connections
7	Attempting to use TCP_ The CONNECT command creates a passive connection, where the input information of this connection is duplicated with that of other connections
8	Connection closed
9	Connection non TCP type
10	Connection establishment failed
11	Connection object rejected connection
12	Received a connection request with a non set IP address
13	Ethernet protocol link not configured with Ethernet free protocol function
14	Incorrect input data length or address information when sending or receiving data using TCP or UDP free protocols
15	TCP_ SEND failed to send data
16	TCP_ RECV failed to receive data
17	Connection non UDP type
18	Using ETH_ DISCONNECT command closes a closed link
19	UDP_ CONNECT input information is duplicated with the connection information of other channels
20	Attempting to use UDP_CONNECT for connecting, the input parameters change
21	Error occurred during TCP_CONNECT setting socket options
22	TCP_ CONNECT binding port process failed
23	TCP_ CONNECT link listening listen process failed
24	Error occurred during UDP_CONNECT setting socket options
25	UDP_ CONNECT binding port process failed
26	Multiple tasks are performing the same connection operation simultaneously

3.3.1.6.5 Application Example of Instructions

Variable Definition

No.	Variable Name	Address	Variable Description	Variable Type	Online Value	Power Fail Safeguard
0001	ETH_DISCONNECT1			ETH_DISCONNECT		FALSE
0002	EN_RD			BOOL	FALSE	FALSE
0003	IDD			BYTE	1	FALSE
0004	DDone			BOOL	FALSE	FALSE
0005	DBusy			BOOL	FALSE	FALSE
0006	DError			BOOL	FALSE	FALSE
0007	DErrID			WORD	0	FALSE

Programing language	Program

Variable Definition	
Ladder diagram （LD）	
Structured text （ST）	ETH_DISCONNECT(EN_R:=EN_RD, ConnectID:=IDD, Done=>DDone, Busy=>Dbusy, Error=>DError, ErrorID=>DErrID);
Continuous Function Chart （CFC）	

3.3.1.7 ComConfig(Set Serial Port Communication Parameter)

3.3.1.7.1 Function

The function currently supports configuring commucation parameter for each channel in online.

3.3.1.7.2 Parameter Description

	Data Item	Data Type	Parameter Description		Initial Value
			MC Platform	LE Platform	
Input	EN_R	BOOL	Enabled in rising edge	Enabled in rising edge	0
	Port	USINT	0: UART1, currently support 232 serial port 1: UART2, currently support 422 / 485 serial port 2: UART3, currently support 485 serial port 3: UART4, currently support 485 serial port	0: Circular interface in CPU body 1: Terminal communication port in CPU body 2: Communication port of extended function board For the port number of serial port expansion module, please refer to hardware configuration >LE5400 serial port configuration parameter Port	0
	ComType	USINT	Communication type 0: RS485 1: RS422 2: RS232	Communication type 0: RS485 1: RS232	0
	Parity	USINT	Check mode 0: no checking	Check mode 0: no checking	0

	Data Item	Data Type	Parameter Description		Initial Value
			MC Platform	LE Platform	
			1: Even parity 2: Odd parity	1: Even parity 2: Odd parity	
	DataWidth	USINT	Data width（Support 8 bit） 8: Data width is 8 bit	Data width,Supports 8/7 bit	8
	Stop	USINT	Stop bit（Support 1, 2） 1: Stop bit is 1 2: Stop bit is 2	Stop bit（Support 1, 2） 1: Stop bit is 1 2: Stop bit is 2	1
	BaudRate	UDINT	Baud rate 9600: support baud rate is 9,600 19200: support baud rate is 19,200 38400: support baud rate is 38,400 57600: support baud rate is 57,600 115200: support baud rate is 115,200 COM1 port does not support the following baud rate, but other COM port support: 500K: baud rate is 500,000 1M: baud rate is 1,000,000 1.125M: baud rate is 1,125,000 1.25M: baud rate is 1,250,000 1.5M: baud rate is 1,500,000	Baud rate, Supports 9600/19200/38400/57600/115200	38400
Output	Q	BOOL	Output The Q is TURE, It can only show the command is executed in current task cycle	Output The Q is TURE, It can only show the command is executed	0
	Err	USINT	0: No error 1: ComType error 2:Port error 4: Parity error 5: DataWidth error 6: Stop error 7: BaudRate error	0: No error 1: Port error 2: ComType error 3: BaudRate error 4: DataWidth error 5: Parity error 6: Stop error 7: Controller internal error	0

ⓘLK220 module does not support this instruction.

3.3.1.8 ComRecv(Receive Serial Port Data)

3.3.1.8.1 Function

Receive the data form serial port.

3.3.1.8.2 Parameter Description

	Data Item	Data Type	Parameter Description	Initial Value (Default)
Input	En_R	BOOL	Active high	0
	DataLenth	UDINT	Data length to receive (Unit: Byte)	0
	DataOffset	UDINT	Offset of M area where data address located (Unit: Byte)	0
	Port	USINT	0: UART1, currently support 232 serial port 1: UART2, currently support 422/485 serial port 2: UART3, currently support 485 serial port 3: UART4, currently support 485 serial port	0
Output	Q	BOOL	Output, the instruction is finished in current task cycle if Q is effectiv	0
	Err	USINT	Error 0: No error 1: Configuration of input parameters is error 2: Free protocol not be configured for current port	0
	RecvDataBytes	UDINT	The number of bytes received actually	0

ⓘLK220 module does not support this instruction.

3.3.1.9 ComSend(Send Serial Port Data)

3.3.1.9.1 Function

Send the data form serial port.

3.3.1.9.2 Parameter Description

	Data Item	Data Type	Parameter Description	Initial Value (Default)
Input	En_R	BOOL	Enabled in rising edge	0
	DataLenth	UDINT	Data length to send (Unit: Byte)	0
	DataOffset	UDINT	Offset of M area where data address located (Unit: Byte)	0
	Port	USINT	0: UART1, currently support 232 serial port 1: UART2, currently support 422/485 serial port 2: UART3, currently support 485 serial port 3: UART4, currently support 485 serial port	0
Output	Q	BOOL	Output	0
	Err	USINT	Error 0: No error 1: Configuration of input parameters is error 2: Free protocol not be configured for current port	0
	SendDataBytes	UDINT	The number of bytes sent actually	0

ⓘLK220 module does not support this instruction.

3.3.1.10 ComClearRecvFIFO(Clear Serial Port Data)

3.3.1.10.1 Function

Clear received data of serial port.

3.3.1.10.2 Parameter Description

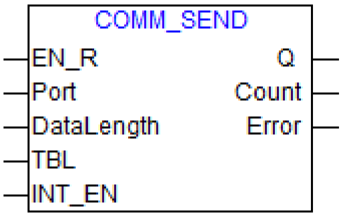
<table>
<tr><th rowspan="2"></th><th rowspan="2">Data Item</th><th rowspan="2">Data Type</th><th colspan="2">Parameter Description</th><th rowspan="2">Initial Value (Default)</th></tr>
<tr><th>MC Platform</th><th>LE Platform</th></tr>
<tr><td rowspan="2">Input</td><td>En_R</td><td>BOOL</td><td>Enabled in rising edge</td><td>Enabled in rising edge</td><td>0</td></tr>
<tr><td>Port</td><td>BYTE</td><td>0: UART1, currently support 232 serial port
1: UART2, currently support 422 / 485 serial port
2: UART3, currently support 485 serial port
3: UART4, currently support 485 serial port</td><td>Only supported by LE5400, please refer to hardware configuration >LE5400 serial port configuration parameter Port</td><td>0</td></tr>
<tr><td rowspan="2">Output</td><td>Q</td><td>BOOL</td><td>Output</td><td>Output</td><td>0</td></tr>
<tr><td>Err</td><td>BYTE(LE Platform)
USINT(MC Platform)</td><td>Error</td><td>Error
0: No error
1: Port configuration error
14: Backplane communication failure or clear failure</td><td>0</td></tr>
</table>

ⓘLK220 module does not support this instruction.

3.3.1.11 COMM_SEND (Free Protocol Communication Data Transmission)

3.3.1.11.1 Function

Provide parameter configuration of serial port free-protocol communication data sending end and showing the user port of sending wrong information.



COMM_SEND Function Block

3.3.1.11.2 Description of Parameter

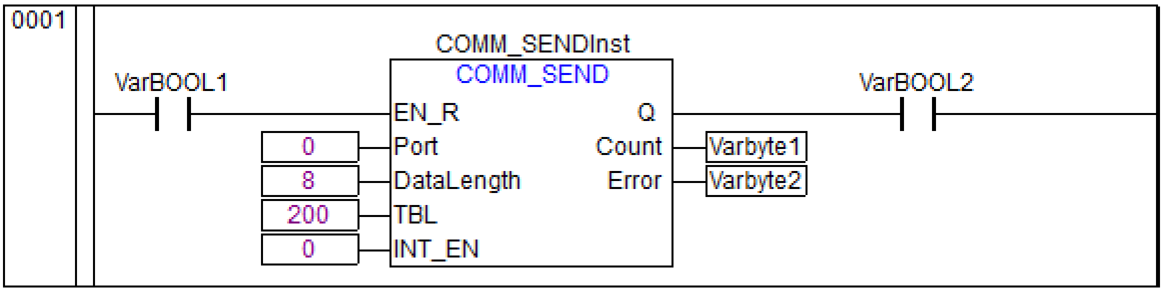
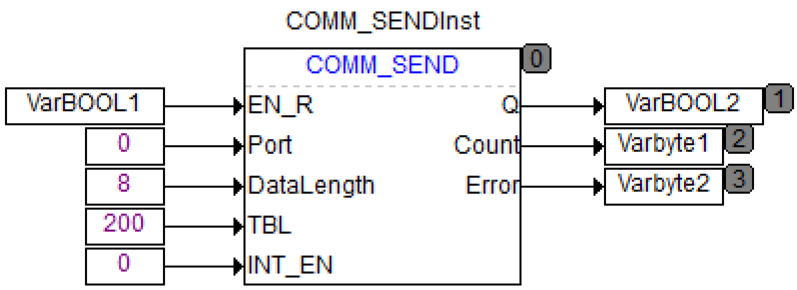
Input Parameter	Type of Data	Description of Function	Description of Parameter Value	Default Value
EN_R	BOOL	Enable	0: Invalid 1: Rising edge enable	0
Port	BYTE	Choice of communication port	0: PS/2 of ontology of CPU 1: Sub-communication port of ontology end of CPU (14 points are not supported) 2: Communication port of extension function board (14 points are not supported) 3: The first communication extension module communication port 4~N: The second to the Nth communication extension module communication port	0
DataLength	BYTE	Sending number of Bytes	Sending number of Bytes, 0~255 bytes	255
TBL	WORD	Saving address of data	Saving address of first byte pointed to data of M area such as 200, showing the space from %MB200 saving address	0
INT_EN	BOOL	Interruption enable input	0: Doesn't break off after sending data 1: Break off after sending data The name of stopping program is set through stopping set interface	0

Output Parameter	Type of Data	Description of Function	Description of Parameter Value
Q	BOOL	Mark of finishing sending	The first scanning circle after finishing sending is 1
Count	BYTE	Number of bytes of actual sending	Number of bytes of actual sending
Error	BYTE	Error information	Error=0: Correct Error=1: Fault set of communication port (Port) Error=2: Data length is wrong Error=3: Cross-border of saving address of data Error=4: Fail to get the space pointer of user Error=5: Many function blocks enable a communication port at the same time Error=8: Communication port occupied (Please check whether the port has been used in other protocols or communicates with AT) Error=14: Backplane communication failure(Only reported by LE5400)

3.3.1.11.3 Application Example of Instructions

Variable Definition

No.	Variable Name	Address	Variable Description	Variable Type	Initial Value	Power Fail Safeguard
0001	COMM_SENDInst			COMM_SEND		FALSE
0002	VarBOOL1			BOOL	FALSE	FALSE
0003	Varbyte1			BYTE	0	FALSE
0004	VarBOOL2			BOOL	FALSE	FALSE
0005	Varbyte2			BYTE	0	FALSE

Programing language	Program
Ladder diagram (LD)	
Structured text (ST)	COMM_SENDInst(EN_R:=VarBOOL1, Port:=0, DataLength:=8, TBL:=200, INT_EN:=0, Q=>VarBOOL2, Count=>Varbyte1, Error=>Varbyte2);
Continuous Function Chart (CFC)	

Program description:

The communication port is set to the circular interface of the CPU body via the Port parameter.

When EN_R is set and held, the 8 bytes starting from % MB200 are sent in consecutive (that is,% MB200-% MB207), Q is equal to TRUE in one cycle after transmission, and then becomes FALSE.

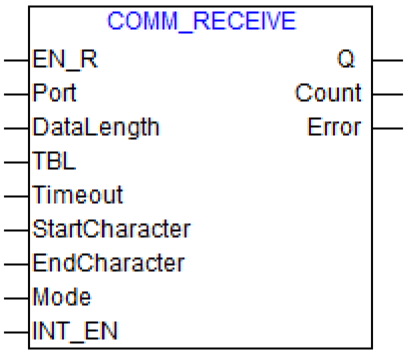
When EN_R is reset, no transmission is performed.

EN_R each time changes from FALSE to TRUE (set), the data is sent once.

3.3.1.12 COMM_RECEIVE(Free Protocol Communication Data Reception)

3.3.1.12.1 Function

Provide parameter configuration of serial port free-protocol communication data receiving end and showing the user port of receiving wrong information.



COMM_ RECEIVE Function Block

3.3.1.12.2 Description of Parameter

Input Parameter	Type of Data	Description of Function	Description of Parameter Value	Default Value
EN_R	BOOL	Enable	0: Invalid 1: Rising edge enable	0
Port	BYTE	Choice of communication port	0: PS/2 of ontology of CPU 1: Sub-communication port of ontology end of CPU (14 points are not supported) 2: Communication port of extension function board (14 points are not supported) 3: The first communication extension module communication port 4~N: The second to the Nth communication extension module communication port	0
Data Length	BYTE	Receiving number of Bytes	Receiving number of bytes, 0~255 bytes, set 0 is variable-length receiving mode	255
TBL	WORD	Saving address of data	Saving address of first byte pointed to data of M area, such as 200, showing the space from %MB200 saving address	0
Timeout	WORD	Timeout of frame （ms）	Calculating from starting the receiving process, and doesn’t receive integral database within the regulated time, and terminate receiving process. The minimum value is 50ms, the minimum time unit is scheduling time of system.	0
StartCharacter	WORD	Beginning bytes	When beginning to use beginning characters, only receiving the beginning characters shows the beginning of frame, otherwise it shall be threw away.	0
EndCharacter	WORD	End bytes	When beginning to use end characters, only receiving the end character shows the end of a frame of data, or continue to receive until buffer reaches (255 bytes)	0

Mode	BYTE	Parameter set of controlling mode	See details in Table_1_Detailed_Instruction_of_Mode_Parameters	0
INT_EN	BOOL	Interruption enable input	0: Doesn't break off after receiving data 1: Break off after receiving data The name of stopping program is set through stopping set interface	0

Output Parameter	Type of Data	Description of Function	Description of Parameter Value
Q	BOOL	Mark of finishing receiving	The first scanning circle after finishing receiving is 1
Count	BYTE	Number of bytes of actual receiving	Number of bytes of actual receiving
Error	BYTE	Error information	Error=0: correct Error=1: Fault set of communication port (Port) Error=2: Cross-border of saving address of data (TBL) Error=3: Lack of end character Error=4: Timeout (Timeout) is set too small Error=5: Fail to get the pointer of user's space Error=6: Timeout Error=7: Multiple function blocks enable a communication port at the same time Error=8: Communication port occupied (Please check whether the port has been used in other protocols or communicates with AT) Error=14: Backplane communication failure(Only reported by LE5400)

Table 1 Detailed Instruction of Mode Parameters

<table>
<tr><th colspan="6">Mode（BYTE）</th></tr>
<tr><td>7 6 5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td></tr>
<tr><td>Undefined</td><td>3.5cha enable control</td><td>Control of length of data of beginning and end character</td><td>Control end character</td><td>Control start character</td><td>Control timeout</td></tr>
<tr><td colspan="6">Control Timeout</td></tr>
<tr><td colspan="6">0: Forbidden timeout control
1: Begin timeout control</td></tr>
<tr><td colspan="6">Start Character Control</td></tr>
<tr><td colspan="6">0: Forbidden beginning character control
1: Begin start character control</td></tr>
<tr><td colspan="6">End character control</td></tr>
<tr><td colspan="6">0: Forbidden end character control
1: Begin end character control</td></tr>
<tr><td colspan="6">Length control of beginning and end characters data</td></tr>
<tr><td colspan="6">0: Start and end character are words
1: Start and end character are bytes</td></tr>
<tr><td colspan="6">3.5char enable control</td></tr>
<tr><td colspan="6">0: Forbidden 3.5char frame control
1: Begin 3.5char frame control</td></tr>
</table>

Data Length, Start Character and End Character Used in Conjunction

DataLength	StartCharacter	EndCharacter	Receiving Process of Character
0	enable	enable	When receiving start character show the beginning of frame, or the data is threw away, only receiving the end character shows the end of frame, or continue to receive until buffer full (255 bytes)

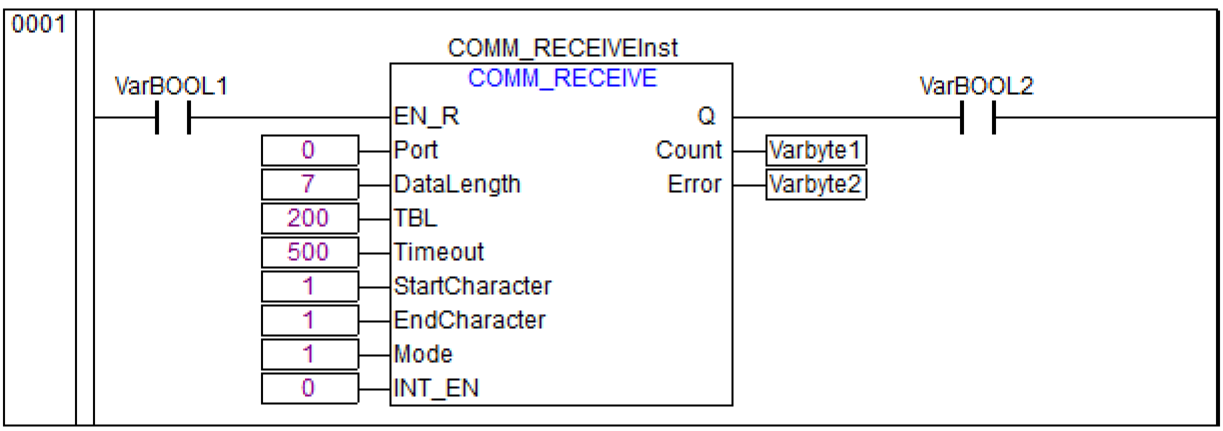
0	forbidden	enable	Receiving the end character shows the end of frame, or continue to receive until buffer full (255 bytes)
0	enable	forbidden	Error, lack of end character
0	forbidden	forbidden	Error, lack of end character
Not 0	enable	forbidden	Continue to receive after receiving start character, and place in order until reaching DataLength characters (including start character), then end the receiving process
Not 0	enable	enable	Continue to receive after receiving start character until reaching DataLength characters (including start character) or receiving end character, then end the receiving process
Not 0	forbidden	enable	Begin to receive when the instruction is valid until reaching DataLength characters (including start character) or receiving end character, then end the receiving process
Not 0	forbidden	forbidden	Begin to receive when the instruction is valid until reaching DataLength characters, then end the receiving process

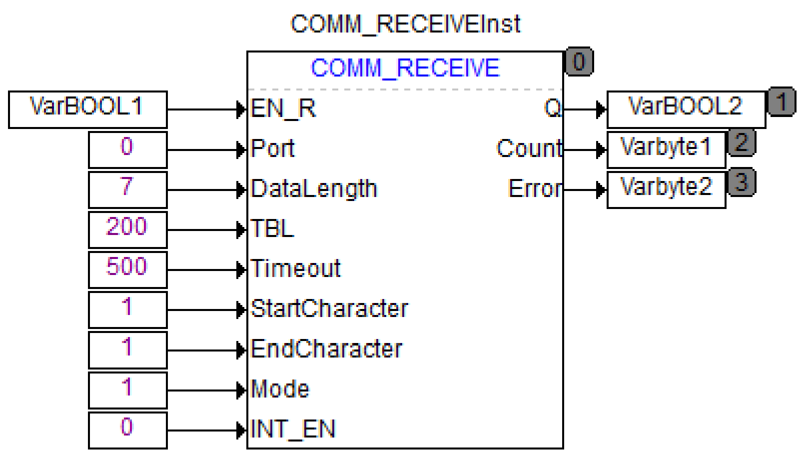
ⓘPay attention to big and little endian when using start /end character to control the functions.

3.3.1.12.3 Application Example of Instructions

Variable Definition

No.	Variable Name	Address	Variable Description	Variable Type	Initial Value	Power Fail Safeguard
0001	COMM_RECEIVEInst			COMM_RECEIVE		FALSE
0002	VarBOOL1			BOOL	FALSE	FALSE
0003	Varbyte1			BYTE	0	FALSE
0004	VarBOOL2			BOOL	FALSE	FALSE
0005	Varbyte2			BYTE	0	FALSE

Programing language	Program
Ladder diagram (LD)	
Structured text (ST)	COMM_RECEIVEInst(EN_R:=VarBOOL1,

	Port:=0, DataLength:=7, TBL:=200, Timeout:=500, StartCharacter:=1, EndCharacter:=1, Mode:=1, INT_EN:=0, Q=>VarBOOL2, Count=>VarByte1, Error=>VarByte2);
Continuous Function Chart (CFC)	

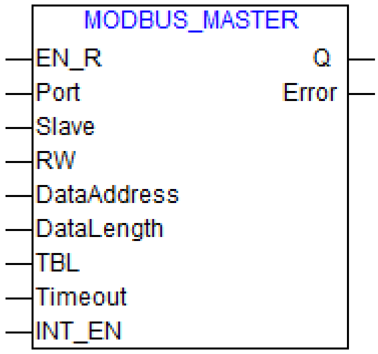
Program description:

In the free protocol communication, the Q is equal to FALSE, when the first scan cycle is ready to receive 7 bytes of data and which were saved to% MB200-% MB206, after receiving successfully Q is equal to TRUE, so EN_R is equal to FALSE. No data is received in the next scan cycle, Q is equal to FALSE, so EN_R is equal to TRUE. And then ready to receive 7 bytes of data in the next scan cycle and which were saved to% MB200-% MB206, so the data is received cyclically.

3.3.1.13 MODBUS_MASTER (Modbus RTU Master Communication)

3.3.1.13.1 Function

Provide the user port which is used to complete read and write data of master station for achieving serial port Modbus protocol.



MODBUS_MASTER Function Block

3.3.1.13.2 Description of Parameter

Input Parameter	Type of Data	Description of Function	Description of Parameter Value	Default Value
EN_R	BOOL	Enable	0: Invalid 1: Rising edge enable	0
Port	BYTE	Choice of communication port	0: PS/2 of ontology of CPU 1: Sub-communication port of ontology end of CPU (14 points are not supported) 2: Communication port of extension function board (14 points are not supported) 3: Communication port in first communication extension module 4~N: Communication port in the second to the Nth communication extension module	0
Slave	BYTE	Address of slave station	Modbus slave station address,1~247	1
RW	BOOL	Choice of read/ write	0: Read data 1: Write data	0
DataAddress	DWORD	Address of slave station for saving data	See details in Table Detail_Description_of_DataAddress_Parameter	1
DataLength	BYTE	Data length	1~100, for total bit number transported for digital input /digital output. Total channels transported for analog input / analog out	1
TBL	WORD	Address of first word of master station for saving data	Points to the first byte address in the M area. Such as 200, showing the saving address is the space from %MW200. If it is read instruction, the back data value shall be saved in the data area. For example, the data of 3050 address of No.3 slave station is 1000, then %MW200 is saved in 1000. If it is writing instruction, then the writing instruction shall be saved in the data area. For example, the writing data of 3050 address of No.3 slave station is 500, then %MW200 is saved in 500.	0
Timeout	WORD	Timeout set（ms）	The receiving process begins, and the correct response frame from slave station is not received within the predetermined time, and the receiving process is terminated. Minimum time unit is 50ms	0
INT_EN	BOOL	Interruption enable	0: Forbidden interruption 1: Interruption enable There will be an interruption separately when the communication port finishes sending the information and receiving information. The interruption 0 and interruption 1 of communication port in the	0

Input Parameter	Type of Data	Description of Function	Description of Parameter Value	Default Value
			interruption list.	

Output Parameter	Type of Data	Description of Function	Description of Parameter Value
Q	BOOL	Mark of finishing receiving	The first scanning circle after finishing receiving is 1
Error	BYTE	Error information	Error=0: Correct Error=1: Communication port（Port）set error Error=2: Timeout（Timeout）set too small Error=3: Station address（Slave）error Error=4: Address of slave station for saving data（DataAddress）error Error=5: Data length（DataLength）error Error=6: Address of master station for saving data（TBL）cross-border Error=7: Function code error Error=8: Fail to get pointer of user's space Error=9: Timeout Error=10: CRC verify error Error=11: Abnormal responding Error=12: Communication port occupied (Please check whether the port has been used in other protocols or communicates with AT) Error=13: Many function blocks enable a communication port at the same time Error=14: Force abnormal responding

Detail Description of DataAddress Parameter

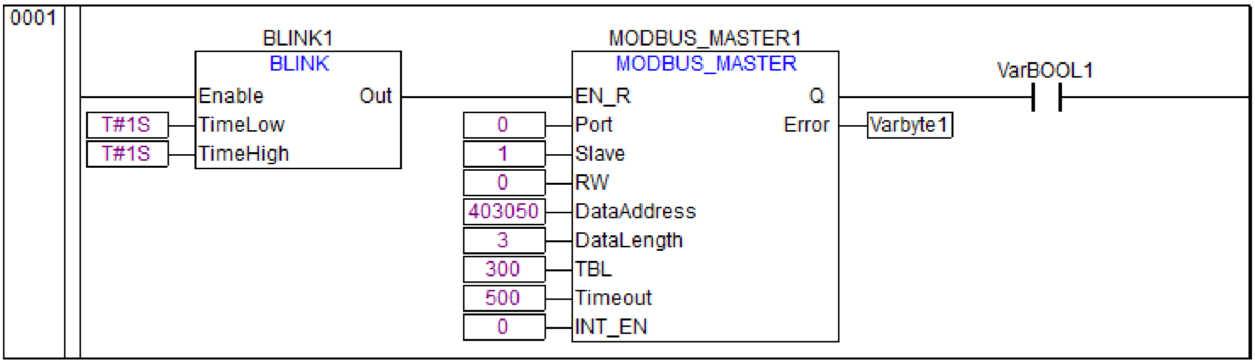
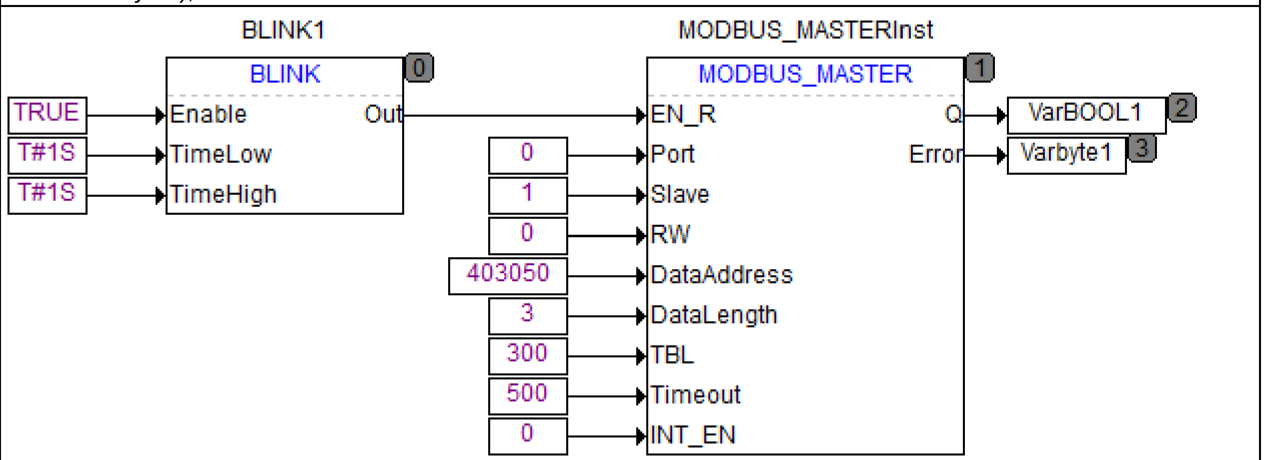
Description of Parameter	
Highest bit	Low five bits
0: Digital-out	Original address of Modbus
1: Digital-in	
3: Analog-in	
4: Analog-out	
The address of slave station adopts filling method of address of standard Modbus RTU drive, that is adopting six digits (decimal), the sixth digit only can choose 0,1,3,4, separately represents different data types of Modbus, low five represents address of register (range of address 0~65, 535). For example, 000001 represents digital output whose Modbus address is 0001; 403050 represents analog output whose Modbus address is 3050.	

3.3.1.13.3 Application Example of Instructions

Variable Definition

No.	Variable Name	Address	Variable Description	Variable Type	Initial Value	Power Fail Safeguard
0001	BLINK1			BLINK		FALSE
0002	MODBUS_MASTERInst			MODBUS_MASTER		FALSE
0003	VarBOOL1			BOOL	FALSE	FALSE
0004	Varbyte1			BYTE	0	FALSE

Programing language	Program

<table>
<tr><th colspan="2">Variable Definition</th></tr>
<tr><td>Ladder diagram (LD)</td><td>
</td></tr>
<tr><td>Structured text (ST)</td><td>BLINK1(
Enable:=TRUE,
TimeLow:=T#1S,
TimeHigh:=T#1S,
Out=>MODBUS_MASTERInst.EN_R);
MODBUS_MASTERInst(
EN_R:= BLINK1.Out,
Port:=0,
Slave:=1,
RW:=0,
DataAddress:=403050
DataLength:=3,
TBL:=300,
Timeout:=500,
INT_EN:=0
Q=>VarBOOL1,
Error=>Varbyte1);</td></tr>
<tr><td>Continuous Function Chart (CFC)</td><td>
</td></tr>
</table>

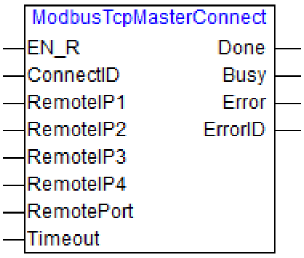
Program description:

The BLINK pulse generator is used to make the square wave pulse so that the MODBUS RTU master function block reads the 3-way analog outputs register starting from 3050 address of No.1 slave station once every 2 seconds and 6 bytes received (each analog for 2 bytes) is stored in the address starting from% MW300, the communication timeout time is 500ms.

3.3.1.14 ModbusTcpMasterConnect(Modbus TCP Master Connect)

3.3.1.14.1 Function

When LE5118/LE5119 module is used as Modbus master station to communicate with slave station, You can configure this function block to establish Modbus TCP communication connection.



ModbusTcpMasterConnect Function Block

3.3.1.14.2 Description of Parameter

Input Parameter	Type of Data	Description of Function	Description of Parameter Value	Default Value
EN_R	BOOL	Enable function block	1.The connection is established at the rising edge. After the connection request is send, the modification of the input parameters is no longer effective 2. Reset and disconnect when the signal is low	FALSE
ConnectID	BYTE	Socket Connection number	It supports 10 connections, and the number value is 1~10. Value cannot be repeated	0
RemoteIP1	BYTE	Slave IP address	Highest bit of address	0
RemoteIP2	BYTE	Slave IP address	\	0
RemoteIP3	BYTE	Slave IP address	\	0
RemoteIP4	BYTE	Slave IP address	lowest bit of address	0
RemotePort	WORD	Slave port number	Set the parameter according to the port number setting of the slave station equipment, the value range is 1~49151	0
Timeout	WORD	Connection timeout	After the connection waiting timeout, the connection will not be send. The setting value is ≥ 50ms	0

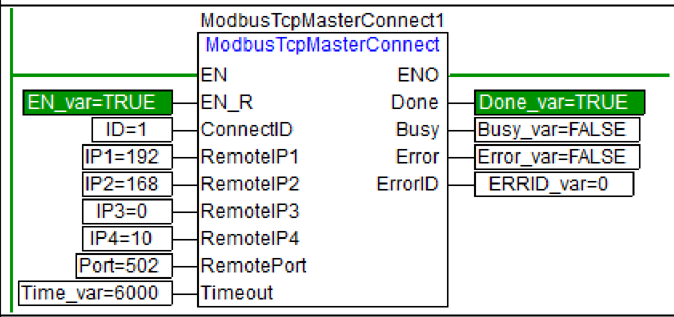
Output Parameter	Type of Data	Description of Function	Description of Parameter Value
Done	BOOL	Connection completion flag	TRUE: connection succeeded FALSE: no connection established
Busy	BOOL	Connecting	TRUE: establishing connection FALSE: connection succeeded or failed
Error	BOOL	Connection failed	TRUE: failed to establish connection FALSE: the connection is successful or in progress
ErrorID	BYTE	Connection failure error code	=0: No error =1: Connectid is illegal. Please check whether the connectid configuration value is 1~10 =2: The slave IP address is illegal. Please check whether the IP address is consistent with the slave address =3: The port number of the slave station is illegal. Please check

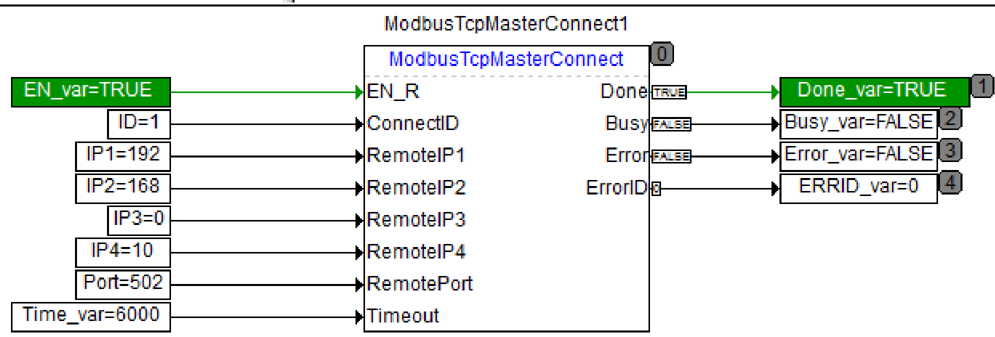
			whether the port number is consistent with that of the slave station =4: Illegal timeout, please check the value ≥ 50ms =5: Connectid conflict, please check whether the connectid configuration value has been used =6: Connection timed out. Please check whether the slave station is abnormal or not online =7: The connection is interrupted. Please check whether the slave station is abnormal or not online =8: Connection failed, please check whether the slave station is abnormal or offline =9: System error, that is an internal problem in the module, please contact technical support

3.3.1.14.3 Application Example of Instructions

Variable Definition

No.	Variable Name	Address	/ariable Descriptior	Variable Type	Online Value	Power Fail Safeguard
0001	EN_var			BOOL	TRUE	FALSE
0002	ID			BYTE	1	FALSE
0003	IP1			BYTE	192	FALSE
0004	IP2			BYTE	168	FALSE
0005	IP3			BYTE	0	FALSE
0006	IP4			BYTE	10	FALSE
0007	Port			WORD	502	FALSE
0008	Time_var			WORD	6000	FALSE
0009	Done_var			BOOL	TRUE	FALSE
0010	Busy_var			BOOL	FALSE	FALSE
0011	Error_var			BOOL	FALSE	FALSE
0012	ERRID_var			BYTE	0	FALSE
0013	ModbusTcpMasterConnect1			MODBUSTCPMASTERCONNECT		TRUE

<table>
<tr><th>Programing language</th><th>Program</th></tr>
<tr><td>Ladder diagram (LD)</td><td>
</td></tr>
<tr><td>Structured text (ST)</td><td>ModbusTcpMasterConnect1(
EN_R:=EN_var,
ConnectID:=ID,
RemoteIP1:=IP1,
RemoteIP2:=IP2,
RemoteIP3:=IP3,
RemoteIP4:=IP4,
RemotePort:=Port ,
Timeout:=Time_var,
Done=>Done_var,
Busy=>Busy_var,
Error=>Error_var,
ErrorID=>ERRID_var);

ModbusTcpMasterConnect1
EN_var = TRUE
ID = 1
IP1 = 192
IP2 = 168
IP3 = 0
IP4 = 10
Port = 502
Time_var = 6000
Done_var = TRUE
Busy_var = FALSE
Error_var = FALSE
ERRID_var = 0</td></tr>
<tr><td>Continuous Function Chart (CFC)</td><td>
</td></tr>
</table>

Program description:

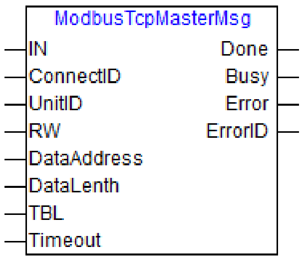
LE5119 as a Modbus master station establishes a communication connection with the slave station equipment. The IP address of the slave device is 192.168.0.10, and the port number of the slave device is 502. During configuration, we set the connection ID to 1 (value range 1~10), as long as it is not repeated with other connection ID. The timeout is set to 6000ms.

During the communication process, if there is no data transmission for a long time (the specific time is different for each slave station), the slave station will automatically disconnect from the master station.

3.3.1.15 ModbusTcpMasterMsg(Modbus TCP Master Msg)

3.3.1.15.1 Function

When LE5118/LE5119 module as Modbus master station communicates with slave station, communication parameters and data are configured through this function block.



ModbusTcpMasterMsg Function Block

3.3.1.15.2 Description of Parameter

Input Parameter	Type of Data	Description of Function	Description of Parameter Value	Default Value
IN	BOOL	Enable function block	When the rising edge is enabled, the high level is effective, that is, each enable must maintain the high level until the end of command execution (success or failure); If the enable pin becomes FALSE before the instruction execution is completed, the instruction terminates	0
ConnectID	BYTE	Socket connection number	It supports 10 connections, and the number value is 1~10. During configuration, it is consistent with the connectid value of the modbustcpmasterconnect function block	
UnitID	BYTE	Slave Unit ID	Values: 1~247, 255	
RW	BOOL	Read / write selection	0: read data 1: Write data	0
DataAddress	DWORD	Address where data is stored in the slave station	See tableDetail_Description_of_DataAddress_Parameter for details	1
DataLength	BYTE	Data length	1~100, for input / output, it is the total number of bits to be transmitted. For analog in / analog out, the total number of channels to be transmitted	1
TBL	WORD	First address of data stored in master station	Address of the first byte pointing to the data store Analog data: For example, 200 indicates that the storage address is a space starting from %MW200. If it is a read instruction, the read data value is stored	0

Input Parameter	Type of Data	Description of Function	Description of Parameter Value	Default Value
			in this data area. For example, if the data at 3050 address of slave station 3 is 1000, then %MW200 stores 1000. If it is a write instruction, put the data value to be written into this data area. For example, write 500 to the 3050 address of slave station 3, and%MW200 stores 500 Switching value data: For example, 100 indicates that the storage address is a space starting from %MX100.0. If it is a read instruction, the read data value is stored in this data area. For example, if the data of 3050 address of slave station 3 is 1, then %MX100.0 stores 1. If it is a write instruction, put the data value to be written into this data area. For example, write 0 to the 3050 address of slave station 3, and %MX100.0 stores 0	
Timeout	WORD	Master request response timeout(ms)	Starting from the start of the receiving process, if the correct slave response frame is not received within the specified time, the receiving process is aborted. The minimum time is 50 ms	0

Output Parameter	Type of Data	Description of Function	Description of Parameter Value
Done	BOOL	Communication success flag	TRUE: this communication is successful FALSE: this communication fails or is waiting for a reply
Busy	BOOL	Responding	TRUE: waiting for a slave reply FALSE: end of communication
Error	BOOL	Communication failure	TRUE: this communication failed FALSE: this communication succeeds or waits for the response from the slave station
ErrorID	BYTE	Communication failure error code	=0: no error =1: Illegal connectid, check whether the connectid configuration value is 1~10 =2: Illegal unit ID, please check whether the ID value is 1~247, 255 =3: The address where the slave station stores data (dataaddress) is illegal,please check whether the data address of the slave meets the requirements of "Detail Description of DataAddress Parameter" in the table =4: Illegal datalength, please check whether the data length is 1~100 =5: The address (TBL) where the master station stores data is out of range, please check the address of the main station ⩾ 16K =6: Illegal timeout, please check the value ⩾ 50ms =7: Address + length out of range, please check the address + length ⩾ 16K =8: Connectid conflict, please check whether the connectid configuration value has been used =9: Failed to send data =10: Failed to receive data =11: Timeout =12: Slave station response frame error (mbap header verification failed) =13: Slave station response data length error =14: Slave station reply address error (write instruction) =15: Slave station response data error (write single coil / register instruction) =16: Disconnected, please check whether the slave station is abnormal or not online =17: System error, module internal problems, please contact technical

Output Parameter	Type of Data	Description of Function	Description of Parameter Value
			support =129: function codes not supported by the slave station, please keep consistent with the configuration of the slave station =130: address not supported by the slave, please keep consistent with the configuration of the slave station =131: data not supported by the slave, please keep consistent with the configuration of the slave station =132: slave station other type error, please keep consistent with the configuration of the slave station

Detail Description of DataAddress Parameter

Description of Parameter	
Highest bit	Low five bits
0: Digital-out 1: Digital-in 3: Analog-in 4: Analog-out	Original address of Modbus
The address of slave station adopts filling method of address of standard Modbus RTU drive, that is adopting six digits (decimal), the sixth digit only can choose 0,1,3,4, separately represents different data types of Modbus, low five represents address of register (range of address 0~65, 535). For example, 000001 represents digital output whose Modbus address is 0001; 403050 represents analog output whose Modbus address is 3050.	

3.3.1.15.3 Application Example of Instructions

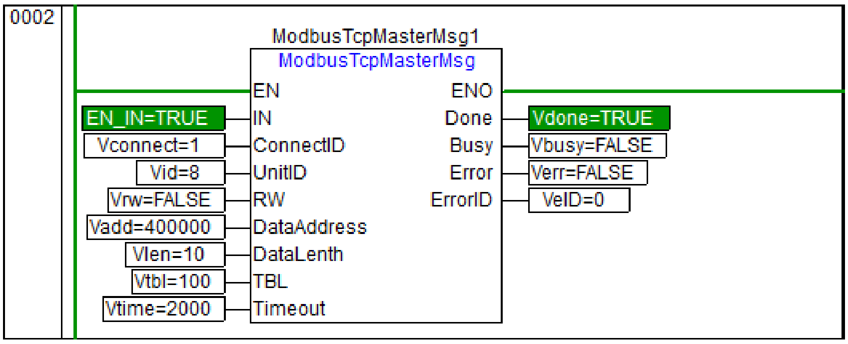
Variable Definition

No.	Variable Name	Address	Variable Description	Variable Type	Initial Value	Power Fail Safeguard
0014	ModbusTcpMasterMsg1			MODBUSTCPMASTERMSG		FALSE
0015	EN_IN			BOOL	FALSE	FALSE
0016	Vconnect			BYTE	1	FALSE
0017	Vid			BYTE	8	FALSE
0018	Vrw			BOOL	FALSE	FALSE
0019	Vadd			DWORD	400000	FALSE
0020	Vlen			BYTE	10	FALSE
0021	Vtbl			WORD	100	FALSE
0022	Vtime			WORD	2000	FALSE
0023	Vdone			BOOL	FALSE	FALSE
0024	Vbusy			BOOL	FALSE	FALSE
0025	Verr			BOOL	FALSE	FALSE
0026	VeID			BYTE	0	FALSE
0027	p1	%MW100		ARRAY[0..20] OF WORD		FALSE
0028	ModbusTcpSlaveConfig1			MODBUSTCPSLAVECONFIG		FALSE

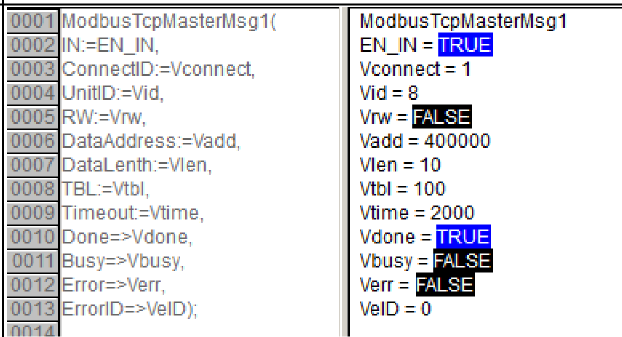
Programing language	Program

Variable Definition

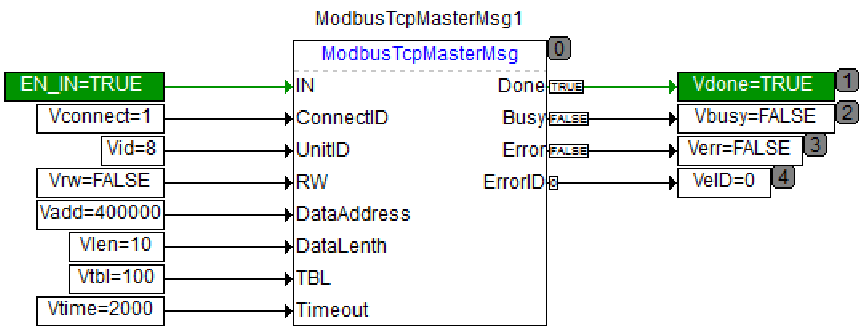
Ladder diagram (LD)



Structured text (ST)



Continuous Function Chart (CFC)

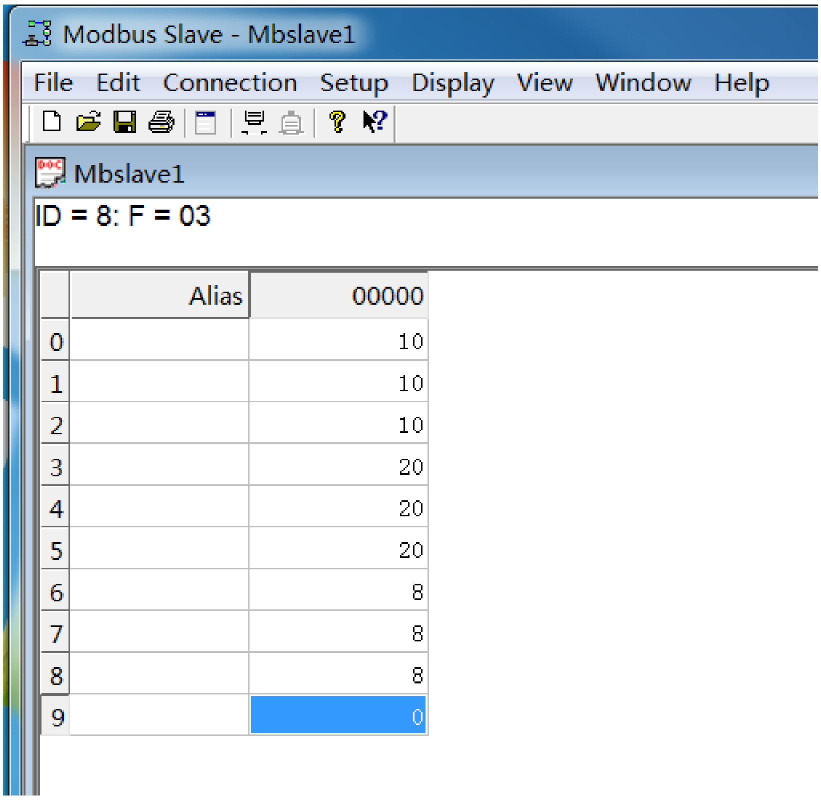


Program description:

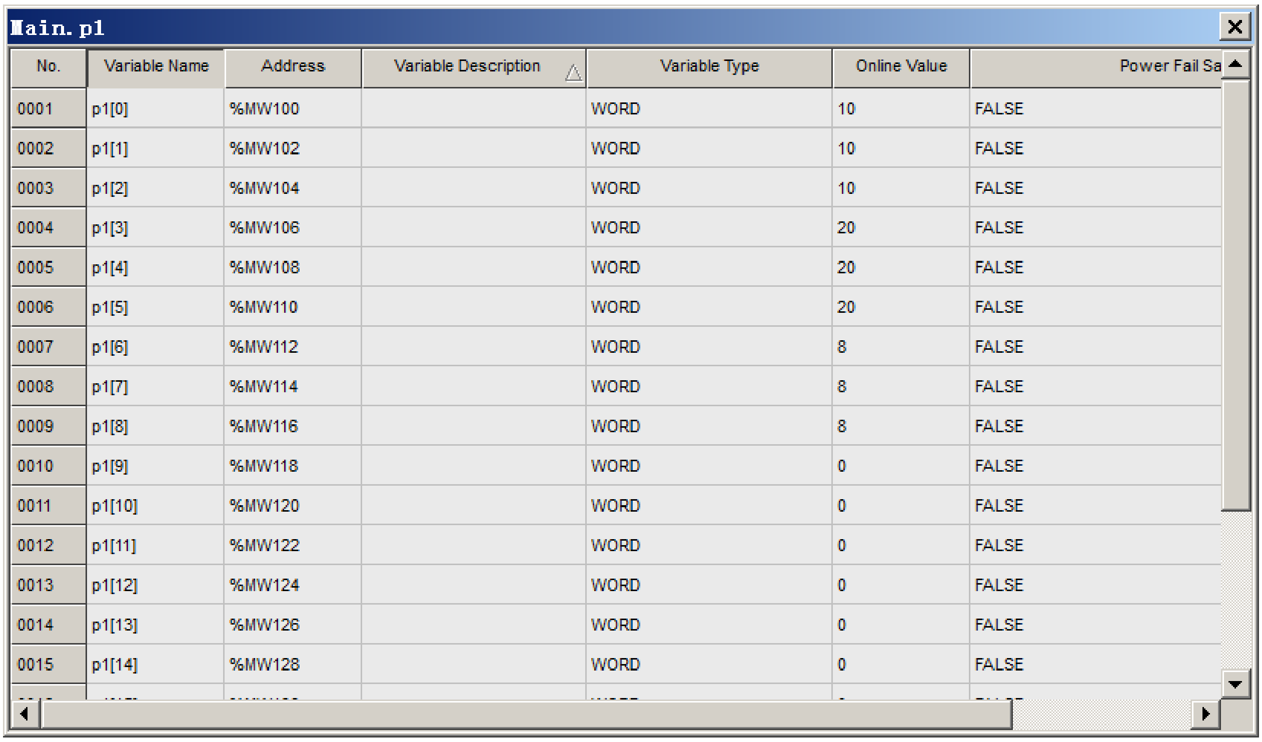
After LE5119 is connected with the slave equipment as the Modbus master station, when the communication enable signal IN is TRUE, the read-write operation can be performed.

In the above example, the master reads the "read holding register" data of the slave (function code 0x03), and the data address of the slave is 0. Read 10 data and store them in the storage area starting from the master station %MW100. During configuration, the ConnectID is consistent with the value set in the ModbusTcpMasterConnect function block.

The following is the data storage diagram of the slave station and the data storage diagram read by the master station.



Data stored in slave station



Main. p1

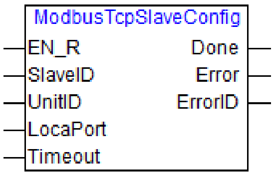
No.	Variable Name	Address	Variable Description	Variable Type	Online Value	Power Fail Sa
0001	p1[0]	%MW100		WORD	10	FALSE
0002	p1[1]	%MW102		WORD	10	FALSE
0003	p1[2]	%MW104		WORD	10	FALSE
0004	p1[3]	%MW106		WORD	20	FALSE
0005	p1[4]	%MW108		WORD	20	FALSE
0006	p1[5]	%MW110		WORD	20	FALSE
0007	p1[6]	%MW112		WORD	8	FALSE
0008	p1[7]	%MW114		WORD	8	FALSE
0009	p1[8]	%MW116		WORD	8	FALSE
0010	p1[9]	%MW118		WORD	0	FALSE
0011	p1[10]	%MW120		WORD	0	FALSE
0012	p1[11]	%MW122		WORD	0	FALSE
0013	p1[12]	%MW124		WORD	0	FALSE
0014	p1[13]	%MW126		WORD	0	FALSE
0015	p1[14]	%MW128		WORD	0	FALSE

Data Read to master

3.3.1.16 ModbusTcpSlaveConfig(Modbus TCP Slave Config)

3.3.1.16.1 Function

When LE5118/LE5119 module is accessed by the master station as a Modbus slave station, the communication parameters of the slave station are configured through this function block. Each call can support 5-way master station to access at the same time. When not called, the function block parameters will work with the default settings.



ModbusTcpSlaveConfig Function Block

3.3.1.16.2 Description of Parameter

Input Parameter	Type of Data	Description of Function	Description of Parameter Value	Default Value
EN_R	BOOL	Enable function block	Parameter configuration on rising edge	FALSE
SlaveID	BYTE	Slave ID	The parameter value is fixed value 1	0
UnitID	BYTE	Unit ID	Value : 1~247 and 255	0
LocalPort	WORD	Local port number	Value: 1~49151	0
Timeout	DWORD	Timeout(ms)	The minimum access interval of the third-party master station. If it exceeds this interval, it will actively disconnect from the third-party master station (when it is 0, the slave station will not actively disconnect)	0

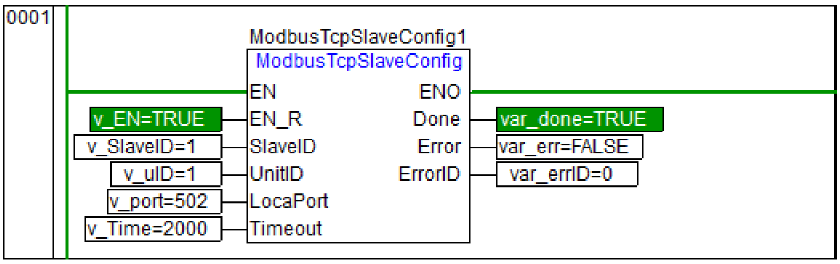
Note: When ModbusTcpSlaveConfig Function Block is not called, the system uses the default configuration SlaveID=1, UnitID=1, LocalPort=502, Timeout=2000ms as salve communication parameters.

Output Parameter	Type of Data	Description of Function	Description of Parameter Value
Done	BOOL	Configuration completion flag	TRUE: configuration succeeded FALSE: configuration failed
Error	BOOL	Error flag	TRUE: configuration failed FALSE:Not enabled or configured successfully
ErrorID	BYTE	Error code	=0: no error =1: Illegal slaveid, please check whether the slaveid is 1 =2: Illegal local unit ID , please check whether the ID is: 1~247, 255 =3: Illegal local port number, please check whether the port configuration is: 1~49151 =4: Illegal IP address,Illegal addresses: 0.0.0.0, 255.255.255.255, 127.0.0.1 =5: Parameter configuration error, please reconfigure =6: System error, please contact technical support

3.3.1.17 Application Example of Instructions

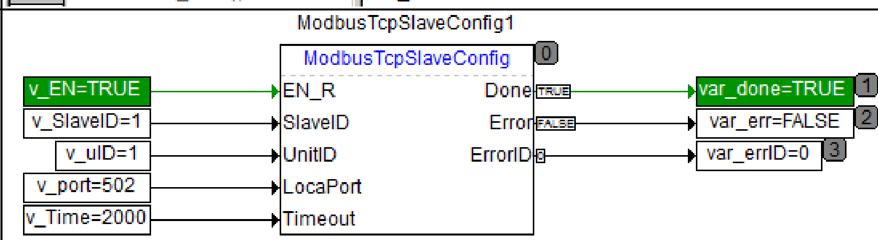
Variable Definition

No.	Variable Name	Address	Variable Description	Variable Type	Online Value	Power Fail Safeguard
0007	ModbusTcpSlaveConfig1			MODBUSTCPSLAVECONFIG		FALSE
0008	v_EN			BOOL	TRUE	FALSE
0009	v_SlaveID			BYTE	1	FALSE
0010	v_uID			BYTE	1	FALSE
0011	v_port			WORD	502	FALSE
0012	v_Time			DWORD	2000	FALSE
0013	var_done			BOOL	TRUE	FALSE
0014	var_err			BOOL	FALSE	FALSE
0015	var_errID			BYTE	0	FALSE
0016	p1	%MW0		ARRAY[0..10] OF WORD		FALSE
0017	p3	%MX100.0		ARRAY[0..10] OF BOOL		FALSE

<table>
<tr><td>Programing language</td><td>Program</td></tr>
<tr><td>Ladder diagram (LD)</td><td>
</td></tr>
<tr><td>Structured text (ST)</td><td>

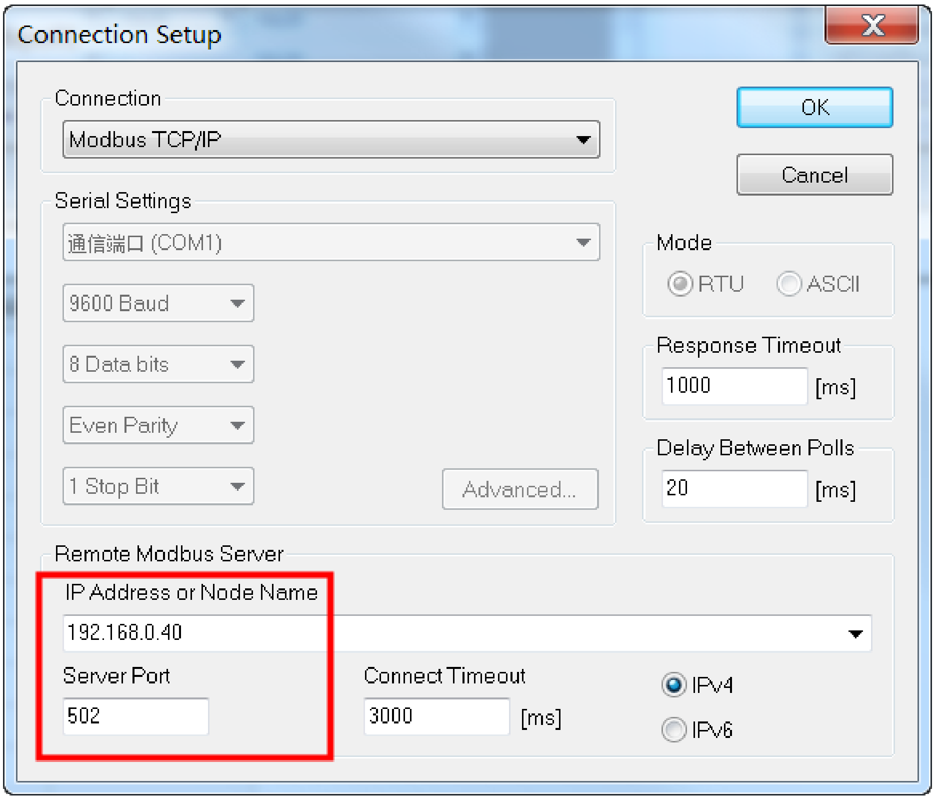
```
0001 ModbusTcpSlaveConfig1(
0002 EN_R:=v_EN,
0003 SlaveID:=v_SlaveID,
0004 UnitID:=v_uID,
0005 LocaPort:=v_port,
0006 Timeout:=v_Time,
0007 Done=>var_done,
0008 Error=>var_err,
0009 ErrorID=>var_errID);
```

```
ModbusTcpSlaveConfig1
v_EN = TRUE
v_SlaveID = 1
v_uID = 1
v_port = 502
v_Time = 2000
var_done = TRUE
var_err = FALSE
var_errID = 0
```

</td></tr>
<tr><td>Continuous Function Chart (CFC)</td><td>
</td></tr>
</table>

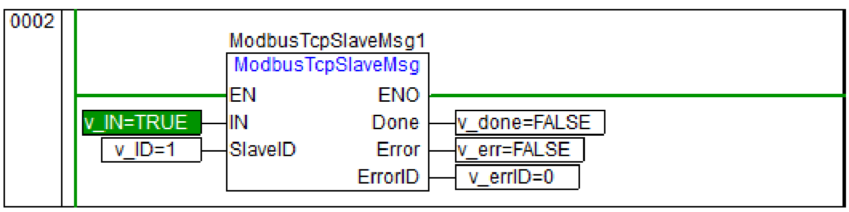
Procedure description:

LE5119 establishes communication connection with master station as Modbus slave station. The SlaveID is set to a fixed value of 1, and the UnitID is set to 1. This parameter is consistent with it in the master station. The slave device has an IP address of 192.168.0.40 and a port number of 502. The IP address and port number need to be configured in the master station, as shown in Figure. When enable signal EN_ R is TRUE, establish a communication connection with the master station. The master station can read and write data at any time.



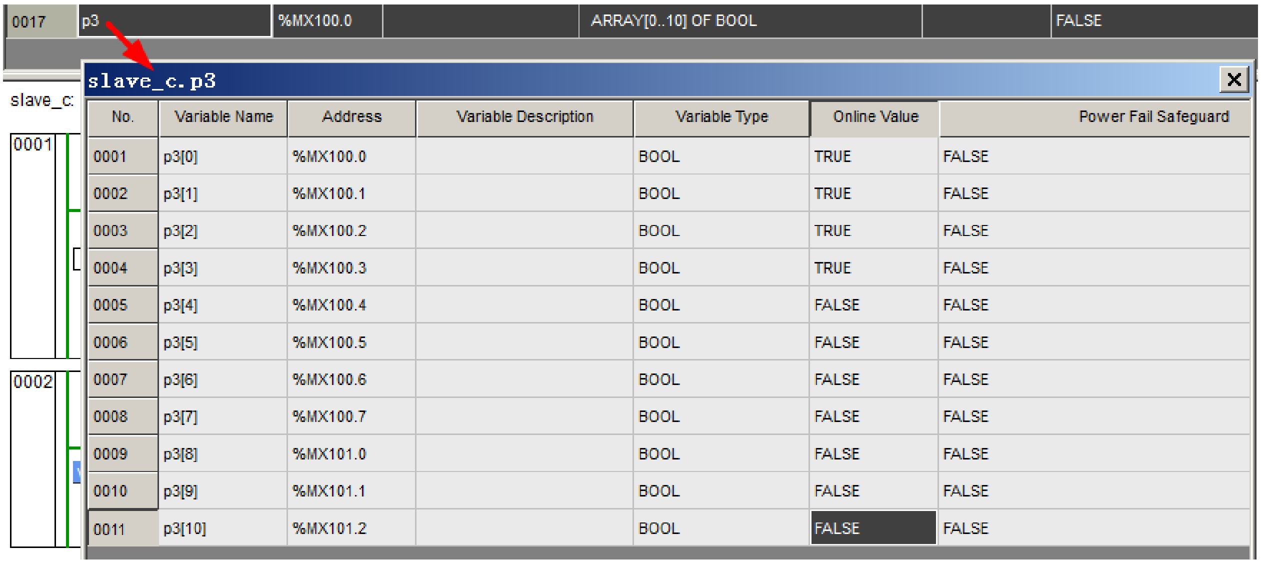
Configure the Slave IP and Port Number in the Master

After the master and slave establish communication connection, start the read / write operation through the function block ModbusTcpSlaveMsg. As shown in Figure, when the communication enable signal IN is TRUE, the master station starts reading and writing data.



Example of Triggering a Read / Write Operation

In the above example, the master station reads the "coil status" data of the slave station (function code 0x01), and the slave station data is stored in the array starting from %MX100.0, as shown in Figure .



0017 p3 %MX100.0 ARRAY[0..10] OF BOOL FALSE

slave_c.p3

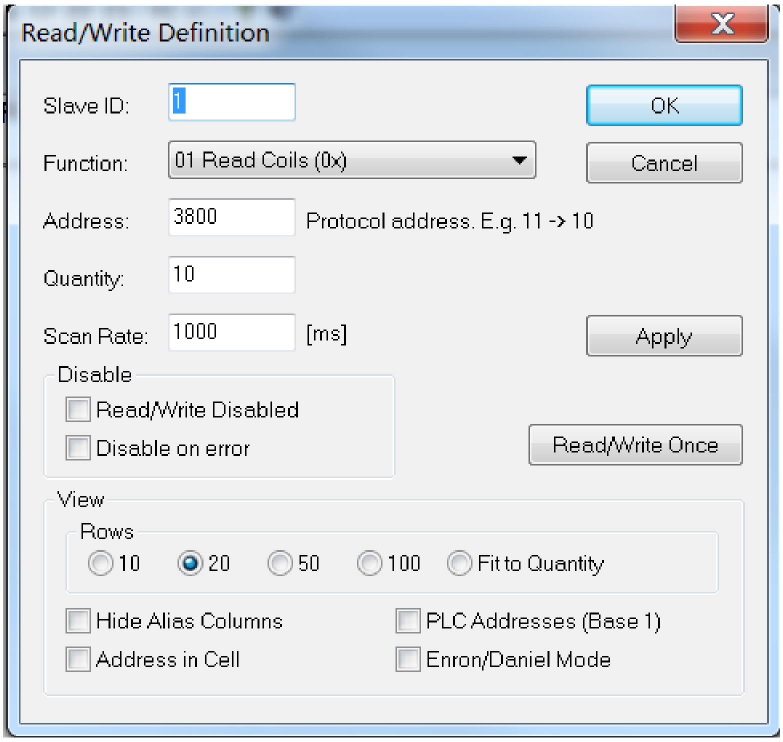
No.	Variable Name	Address	Variable Description	Variable Type	Online Value	Power Fail Safeguard
0001	p3[0]	%MX100.0		BOOL	TRUE	FALSE
0002	p3[1]	%MX100.1		BOOL	TRUE	FALSE
0003	p3[2]	%MX100.2		BOOL	TRUE	FALSE
0004	p3[3]	%MX100.3		BOOL	TRUE	FALSE
0005	p3[4]	%MX100.4		BOOL	FALSE	FALSE
0006	p3[5]	%MX100.5		BOOL	FALSE	FALSE
0007	p3[6]	%MX100.6		BOOL	FALSE	FALSE
0008	p3[7]	%MX100.7		BOOL	FALSE	FALSE
0009	p3[8]	%MX101.0		BOOL	FALSE	FALSE
0010	p3[9]	%MX101.1		BOOL	FALSE	FALSE
0011	p3[10]	%MX101.2		BOOL	FALSE	FALSE

Slave Data

Mapping Relations between the Data Areas and MODBUS TCP Protocol

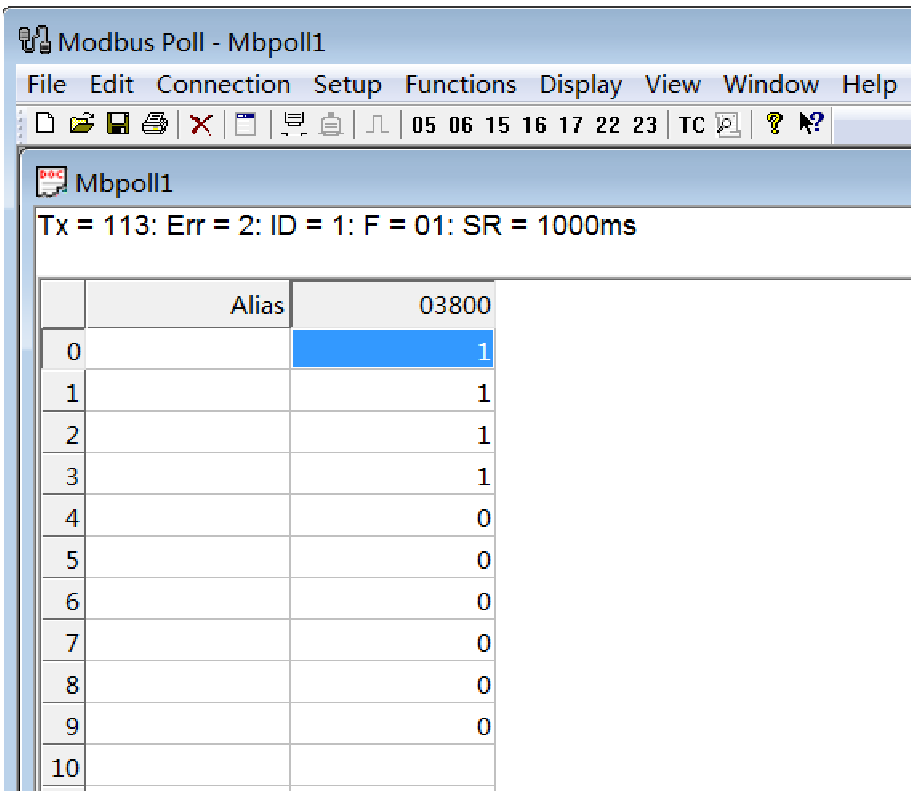
Data area		Type	Address range	MODBUS address	Mapping Relation	X (register type)selection
I area	%IX	BOOL	%IX0.0,… %IX0.7 %IX1.0,… %IX1.7 …%IX374.7	X0000,…X0007 X0008,…X0015 …X2999	IXm.n: m*8+n	Read-only, X selects 1
	%IW	WORD	%IW0, %IW2,…%IW5998	X0000, X0001,…X2999	IWm: m/2	Read-only, X selects 3
Q area	%QX	BOOL	%QX0.0,…%QX0.7 %QX1.0,…%QX1.7 …%QX374.7	X0000,…X0007 X0008,…X0015 …X2999	QXm.n: m*8+n	Read and Write, X selects 0
	%QW	WORD	%QW0, %QW2,…%QW5998	X0000, X0001,…X2999	QWm: m/2	Read and Write, X selects 4
M area	%MX	BOOL	%MX0.0,…%MX0.7 %MX1.0,…%MX1.7 …%MX7816.7	X3000,…X3007 X3008,…X3015 …X65,535	MXm.n: m*8+n+3000	Read-only, X selects 1 Read and Write, X selects 0
	%MW	WORD	%MW0, %MW2,…%MW125070	X3000,X3001,…X65,535	MWm: m/2+3000	Read-only, X selectes 3 Read and Write, X selects 4

Configure the communication data in the master station, as shown in follow Figure. The slave ID number is consistent with the UnitID of the ModbusTcpSlaveConfig function block. According to the mapping relationship corresponding of M area in above table, configure the starting address of the read / write slave station in the master station as: 100 × 8+3000=3800。 Read the 10 data starting from this address and store it in the master station.



Configure Read / Write Data

When the communication enable signal IN of ModbusTcpSlaveMsg function block is TRUE, the master station reads 10 bool data starting from station %MX100.0, as shown in the following figure.

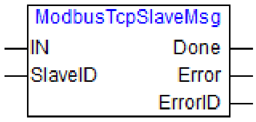


Data Read by Master Station

3.3.1.18 ModbusTcpSlaveMsg(Modbus TCP Slave Msg)

3.3.1.18.1 Function

LE5118/LE5119 module is accessed by the master station as a Modbus slave station, the communication data is configured through this function block.



ModbusTcpSlaveMsg Function Block

3.3.1.18.2 Description of Parameter

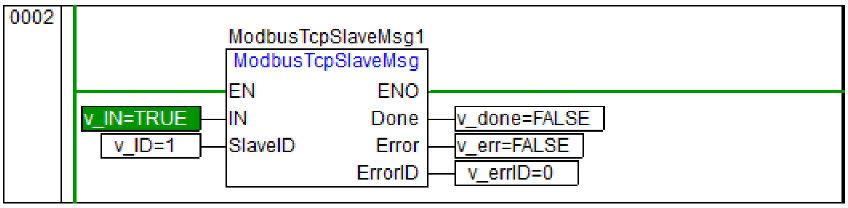
Input Parameter	Type of Data	Description of Function	Description of Parameter Value	Default Value
IN	BOOL	Enable function block	TRUE: enter monitoring state and respond to the master station connection request and read /write request FALSE: actively disconnection from the third party master station	FALSE
SlaveID	BYTE	Slave ID	The parameter value is fixed value 1	0

Output Parameter	Type of Data	Description of Function	Description of Parameter Value
Done	BOOL	Completion flag	Process one instruction in the current IEC cycle
Error	BOOL	Communication error	\
ErrorID	BYTE	Error code	=0: no error =1: Illegal slaveid, please check whether the slaveid is 1 =2: Slaveid conflict , please check that the slaveid value has been used =3: System error.this is module internal problem, please contact technical support =4: Illegal unit ID,please check whether the master and slave UnitIDS are consistent =5: Illegal length =6: Illegal data =7: Failed to receive data =8: Illegal data frame =9: Failed to send data =10: Disconnected,please check whether the slave service is abnormal

3.3.1.18.3 Application Example of Instructions

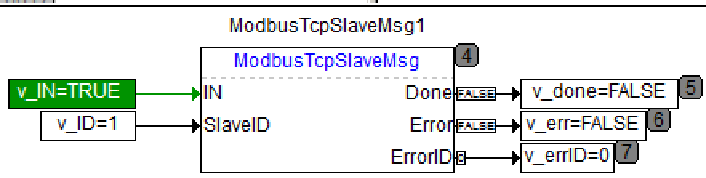
Variable Definition

No.	Variable Name	Address	Variable Description	Variable Type	Online Value	Power Fail Safeguard
0001	v_IN			BOOL	TRUE	FALSE
0002	v_ID			BYTE	1	FALSE
0003	v_done			BOOL	FALSE	FALSE
0004	v_err			BOOL	FALSE	FALSE
0005	v_errID			BYTE	0	FALSE
0006	ModbusTcpSlaveMsg1			MODBUSTCPSLAVEMSG		FALSE

<table>
<tr><td>Programing language</td><td>Program</td></tr>
<tr><td>Ladder diagram (LD)</td><td>
</td></tr>
<tr><td>Structured text (ST)</td><td>

```
ModbusTcpSlaveMsg1(
IN:=v_IN ,
SlaveID:=v_ID,
Done=>v_done,
Error=>v_err,
ErrorID=>v_errID);
```

```
ModbusTcpSlaveMsg1
v_IN = TRUE
v_ID = 1
v_done = FALSE
v_err = FALSE
v_errID = 0
```

</td></tr>
<tr><td>Continuous Function Chart (CFC)</td><td>
</td></tr>
</table>

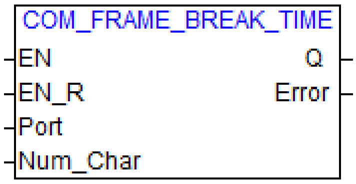
Procedure description:

After LE5119 is connected with the master station as aModbus slave station, when the communication enable signal IN is TRUE, the master station starts reading and writing data. For detailed examples, refer to the instruction example Application_Example_of_Instructions .

3.3.1.19 COM_FRAME_BREAK_TIME(COM PORT user-define)

3.3.1.19.1 Function

Implementation broken frame intervals between Modbus RTU and free port data reception,Support custom modbus RTU communication data for setting.



COM_FRAME_BREAK_TIME Function Block

- Applies to LE5107E-F01,LE5118-B01and LE5119-B01 and above on LE system.

3.3.1.19.2 Description of Parameter

Input Parameter	Type of Data	Description of Function	Description of Parameter Value	Default Value
EN_R	BOOL	Enable	0: Invalid 1: Rising edge enable	0
Port	BYTE	Communication port selection	LE： 0：Circular interface to the CPU 1：Terminal communication port to the CPU（Not supported at 14 points） 2：Extended function board communication port（Not supported at 14 points）	0
Num_Char	BYTE	Custom broken frame bytes	Custom break frame selection：0~255 0~3:Select port default 3.5char break frame time 4~255：Broken frame time set to the corresponding byte.	0

Output Parameter	Type of Data	Description of Function	Description of Parameter Value	Default Value
Q	BOOL	Completion flag	First scan cycle after received is 1.	0
Error	BYTE	Error information	Error=0：Right Error =1：Communication port（Port） Setup Error. Error =2：LE5601 is not configured when port 2 configuration 485 frame-breaks is selected. Error =3：BaudRate Error.	0

Protocol stack formula：Broken frame time (ms) = (bytes of broken frames * 1000 * Single byte number)/baud rate

Single byte number = Check bit + Stop bit + Data bit + Start bit.

Method of verification:

1 Check bit: no check: 0 bits; odd/even check: 1;

2 Stop bits: 0,1,2;

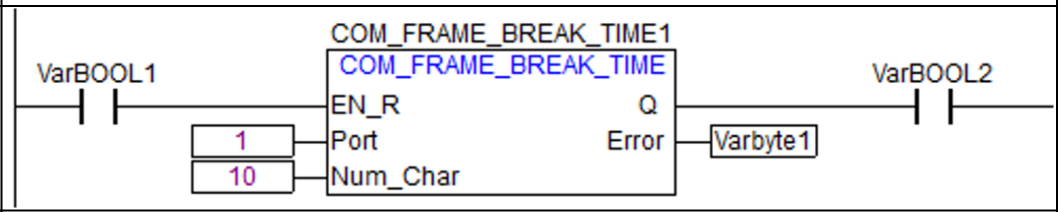
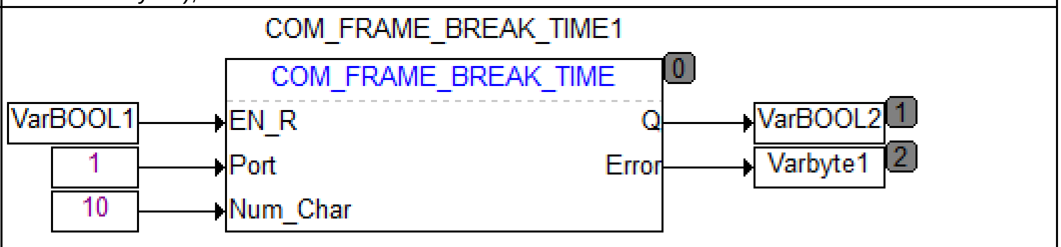
3 Data bits: 7,8;

4 Starting bit: 1.

3.3.1.19.3 Application Example of Instructions

Variable Definition

No.	Variable Name	Address	Variable Description	Variable Type	Online Value	Power Fail Safeguard
0001	COM_FRAME_BREAK_TIME1			COM_FRAME_BREAK_TIME		FALSE
0002	VarBOOL1			BOOL	TRUE	FALSE
0003	VarBOOL2			BOOL	FALSE	FALSE
0004	Varbyte			BYTE	0	FALSE

Programing language	Program
Ladder diagram （LD）	
Structured text （ST）	COM_FRAME_BREAK_TIME1(EN_R:=VarBOOL1, Port1:=1, Num_Char:=10, Q=>VarBOOL2, Error=>Varbyte1);
Continuous Function Chart （CFC）	

3.3.2 Task Operation Instruction

3.3.2.1 TaskStart(Task Start)

3.3.2.1.1 Function Prototype

DINT TaskStart (UDINT uiTaskID);

3.3.2.1.2 Function

Run the specified task

3.3.2.1.3 Parameter Description

	Data Item	Data Type	Parameter Description
Input end	uiTaskID	UDINT	Task ID
Output end	TaskStart	DINT	Success to return 0, or failure to return -1

ⓘLK220 module does not support this instruction.

3.3.2.2 TaskStop(Task Stop)

3.3.2.2.1 Function Prototype

DINT TaskStop (UDINT uiTaskID);

3.3.2.2.2 Function

stop the specified task.

3.3.2.2.3 Parameter Description

	Data Item	Data Type	Parameter Description
Input end	uiTaskID	UDINT	Task ID
Output end	TaskStop	DINT	Success to return 0, or failure to return -1

ⓘLK220 module does not support this instruction.

3.3.2.3 TaskSuspend(Task Suspension)

3.3.2.3.1 Function Prototype

DINT TaskSuspend (UDINT uiTaskID);

3.3.2.3.2 Function

Suspend the specified task.

3.3.2.3.3 Parameter Description

	Data Item	Data Type	Parameter Description
Input end	uiTaskID	UDINT	Task ID
Output end	TaskSuspend	DINT	Success to return 0, or failure to return -1

ⓘLK220 module does not support this instruction.

3.3.2.4 TaskStatus(Return Task Status)

3.3.2.4.1 Function Prototype

DINT TaskStatus (UDINT uiTaskID);

3.3.2.4.2 Function

Return current state of task.

3.3.2.4.3 Parameter Description

	Data Item	Data Type	Parameter Description
Input end	uiTaskID	UDINT	Task ID
Output end	TaskStatus	DINT	Task state: Running 1 Suspending 2 Stoping 3 Failure -1

ⓘLK220 module does not support this instruction.

3.3.3 File Operation Instruction

ⓘ

- File Operation Instruction is supported only in MC platform.
- "/media/sd/" and "/media/flash/" path are only applicable to firmware versions before V1.1.0_R. path in firmware version V1.1.0_R and later versions are “sd/” and “flash/”.
- Before opening (FileOpen), finding (FindFirstFile, FindNextFile), scanning (FileListScan) and removing (FileRemove) files in the "sd/" directory, please ensure that the SD card has been inserted into

the slot correctly.

3.3.3.1 FileOpen(Open File)

3.3.3.1.1 Function Prototype

DINT FileOpen (STRING Name, UDINT uiAccess);

3.3.3.1.2 Function

Open the file according to the manner which is set by user.

3.3.3.1.3 Parameter Description

	Data Item	Data Type	Parameter Description
Input end	Name	STRING(63)	The file name supports maximum 63 characters, and only supports the full path way sd/ is directory of SD flash/ is directory of flash
	uiAccess	UDINT	File operation modes 0: Read and write (open the file in read- write mode by default) 1: Read only (open the file in a read-only mode) 2: Write only (open the file in a write-only mode) 8: Create (create and open the file) 16: Clear (clear the contents of the file and open the file) 32: Add (position the cursor to the end of the file and open the file) Be careful： 1) The cursor is located in the file head in other modes except Add mode 2) The write mode will covers the content after the cursor 3) The cursor can be moved through the FileLseek instruction
Output end	FileOpen	DINT	Success: return the ID of file visit Failure: return -1

- FileClose is performed after the write-read operation is executed by FileOpen, otherwise resources can not be released.
- Naming rules of file is limited by file system, SD adopts the FAT32 file system, then naming rules the same as FAT32 file system. File naming of flash is limited by JFFS2 file system, then naming rules the same as JFFS2 file system.
- LK220 module does not support this instruction.

3.3.3.2 FileClose(Close File)

3.3.3.2.1 Function Prototype

DINT FileClose (DINT iFd);

3.3.3.2.2 Function

Close the file.

3.3.3.2.3 Parameter Description

	Data Item	Data Type	Parameter Description
Input end	iFd	DINT	FileOpen returns the file ID
Output end	FileClose	DINT	Success: return 0 Failure: return -1

ⓘLK220 module does not support this instruction.

3.3.3.3 FileRead(Read File)

3.3.3.3.1 Function Prototype

DINT FileRead (DINT iFd, POINTER TO USINT pcBuf, UDINT uiSize);

3.3.3.3.2 Function

Read the file.

3.3.3.3.3 Parameter Description

	Data Item	Data Type	Parameter Description
Input end	iFd	DINT	FileOpen returns the file ID
	pcBuf	POINTER TO USINT	Read memory address of data
	uiSize	UDINT	Read file size (byte)
Output end	FileRead	DINT	Success: return actual number of bytes for read the file Failure: return -1

ⓘ

- In the process of reading and writing files via FileRead / FileWrite, such as return -1, ensuring no error to input parameters, you need to close and open the file again to continue reading and writing.

- LK220 module does not support this instruction.

3.3.3.4 FileWrite(Write File)

3.3.3.4.1 Function Prototype

DINT FileWrite (DINT iFd, POINTER TO USINT pcBuf, UDINT uiSize);

3.3.3.4.2 Function

Write file.

3.3.3.4.3 Parameter Description

	Data Item	Data Type	Parameter Description
Input end	iFd	DINT	FileOpen returns the file ID
	pcBuf	POINTER TO USINT	Write memory address of data
	uiSize	UDINT	Write file size (byte)
Output end	FileWrite	DINT	Success: return size of data successfully written Failure: return -1

- In the process of reading and writing files via FileRead / FileWrite, such as return -1, ensuring no error to input parameters, you need to close and open the file again to continue reading and writing.
- LK220 module does not support this instruction.

3.3.3.5 FileLseek(Move File to Read/Write Pointer)

3.3.3.5.1 Function Prototype

DINT FileLseek (DINT iFd, DINT iOffset, UDINT uiFrom);

3.3.3.5.2 Function

Move the file read and write pointers.

3.3.3.5.3 Parameter Description

	Data Item	Data Type	Parameter Description
Input end	iFd	DINT	FileOpen returns the file ID
	iOffset	DINT	Move the file pointer offset
	uiFrom	UDINT	0: Locate to the file header 1: Locate to the current position of file 2: Locate to the end of the file
Output end	FileLseek	DINT	Success: return to current read-write position Failure:return -1

ⓘLK220 module does not support this instruction.

3.3.3.6 FileListScan(Scan Directory File)

3.3.3.6.1 Function Prototype

SINT FileListScan (STRING PathName, POINTER TO USINT pFileNameList, SINT cFileNum, SINT cFileNameSize);

3.3.3.6.2 Function

Scan all files contained in the specified path.

3.3.3.6.3 Parameter Description

	Data Item	Data Type	Parameter Description
Input end	PathName	STRING(63)	Directory name to be scanned supports maximum 63 characters, and only supports the full path way sd/ is directory of SD flash/ is directory of flash
	pFileNameList	POINTER TO USINT	Address for string array of pointer. Return list of file names Scanned
	cFileNum	SINT	Number of file to be scanned, Currently it supports up to 32 files
	cFileNameLen	SINT	File name string '\ 0' end (including the maximum path length: the maximum length of 63 characters + terminator character '\ 0', a total of 64 characters)
Output end	FileListScan	SINT	Success: return number of file scanned Failure: return -1

ⓘLK220 module does not support this instruction.

3.3.3.7 FindFirstFile(Seach First File)

3.3.3.7.1 Function Prototype

DINT FindFirstFile (STRING PathName, STRING FileName);

3.3.3.7.2 Function

Open the file handle directory traversal, and get the first name of the file under the directory.

3.3.3.7.3 Parameter Description

	Data Item	Data Type	Parameter Description
Input end	Str1	STRING	The directory name to be traversed supports maximum 63 characters, and only supports the full path way sd/ is directory of SD flash/ is directory of flash
	Str2	STRING	Return file name traversed (excluding path , maximum 63 characters is supported)
Output end	Find	INT	Success : return file handle scanned Failure: return -1

ⓘLK220 module does not support this instruction.

3.3.3.8 FindNextFile(Seach Next File)

3.3.3.8.1 Function Prototype

SINT FindNextFile (DINT iFileFindHandle, STRING FileName);

3.3.3.8.2 Function

Traversal directory to get next file name in specified directory.

3.3.3.8.3 Parameter Description

	Data Item	Data Type	Parameter Description
Input end	iFileFindHandle	DINT	File handle traversed (return value of FindFirstFile)
	FileName	STRING(63)	Return file name traversed (excluding path , maximum 63 characters is supported)
Output end	FindNextFile	SINT	Success: return file handle scanned Failure: return -1

ⓘLK220 module does not support this instruction.

3.3.3.9 FindClose(Seach End)

3.3.3.9.1 Function Prototype

SINT FindClose (DINT iFileFindHandle);

3.3.3.9.2 Function

Close the file handle traversed.

3.3.3.9.3 Parameter Description

	Data Item	Data Type	Parameter Description
Input end	iFileFindHandle	DINT	file handle traversed (return value of FindFirstFile)
Output end	FindClose	SINT	Success: return 0 Failure: return -1

ⓘLK220 module does not support this instruction.

3.3.3.10 FileRemove(Delete File)

3.3.3.10.1 Function Prototype

DINT FileRemove (STRING pStrFileName);

3.3.3.10.2 Function

Delete file.

3.3.3.10.3 Parameter Description

	Data Item	Data Type	Parameter Description
Input end	Name	STRING(63)	The file name supports maximum 63 characters, and only supports the full path way sd/ is directory of SD flash/ is directory of flash
Output end	FileRemove	DINT	Success: return 0 Failure: return -1

ⓘLK220 module does not support this instruction.

3.3.3.11 FlashWrite(Write Data to Flash)

3.3.3.11.1 Function Prototype

DINT FlashWrite (DINT iFileID, POINTER TO USINT pDataAddr);

3.3.3.11.2 Function

File ID ranging from 0 to 15, corresponding to 16 file on the Flash. By calling this function the user can complete that data of specified address 1k bytes is stored in the corresponding file in Flash.

3.3.3.11.3 Parameter Description

	Data Item	Data Type	Parameter Description
Input end	iFileID	DINT	iFileID
	pDataAddr	POINTER TO USINT	Power down protection zone address
Output end	Flashwrite	DINT	Success: 0 Failure: -1

ⓘLK220 module does not support this instruction.

3.3.3.12 FlashRead(Read Data From Flash)

3.3.3.12.1 Function Prototype

DINT FlashRead (DINT iFileID, POINTER TO USINT pDataAddr);

3.3.3.12.2 Function

File ID ranging from 0 to 15, corresponding to 16 file on the Flash. By calling this function the user can complete that the specified file data on the Flash is restored to data area.

3.3.3.12.3 Parameter Description

	Data Item	Data Type	Parameter Description
Input end	iFileID	DINT	iFileID

	pDataAddr	POINTER TO USINT	Power down protection zone address
Output end	FlashRead	DINT	Success: 0 Failure: -1

ⓘLK220 module does not support this instruction.

3.3.4 Time Correlation Instruction

3.3.4.1 GetTickCnt (Get Tick Count)

3.3.4.1.1 Function Prototype

UDINT GetTickCnt (void);

3.3.4.1.2 Function

Get the time of the controller since it boots (unit: μs, time frame: 0 - 2^{32}μs).

3.3.4.1.3 Parameter Description

Return Value	Type	Meaning
GetTickCnt	UDINT	The time since the controller boots, which is counted from zero after it overflows (us)

3.3.4.2 TimeDelay (Time Delay)

3.3.4.2.1 Function Prototype

DINT TimeDelay (UDINT uiTimeDelay);

3.3.4.2.2 Function

After it is called, this function will return after it executes uiTimeDelay milliseconds, so as to achieve the effect of time delay.

3.3.4.2.3 Parameter Description

Input Parameter	Type	Parameter Description

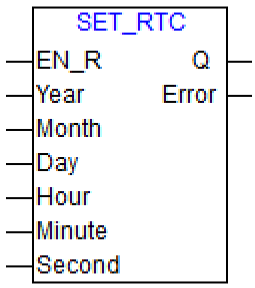
uiTimeDelay	UDINT	Delay time (unit: ms)

Output Parameter	Type	Parameter Description
TimeDelay	DINT	0: The time delay is executed successfully. 1: The time delay is failed to be executed.

3.3.4.3 SET_RTC(Set Real Time Clock)

3.3.4.3.1 Function

Set the user specified time to the PLC real time clock.



SET_RTC Function Block

3.3.4.3.2 Parameter Description

Input Parameter	Data Type	Function Description	Parameter Description	Default Value
EN_R	BOOL	Enable	0: Invalid 1: Rising edge valid	0
Year	WORD	Year	LE/MKC platform: 2000~2099 MC platform: MC1000 series 1971 ~ 2036, MC2000 series, MC1002R, MC1002E 2000 ~ 2036 LK platform: V2.1.0 before 1971 ~ 2036, V2.1.0 ~V2.8.0 is 2000 ~ 2036, V2.8.1 and later is 2000~2066.	2011
Month	BYTE	Month	1-12	1
Day	BYTE	Day	1-31	1
Hour	BYTE	Hour	0-23	0
Minute	BYTE	Minute	0-59	0
Second	BYTE	Second	0-59	0

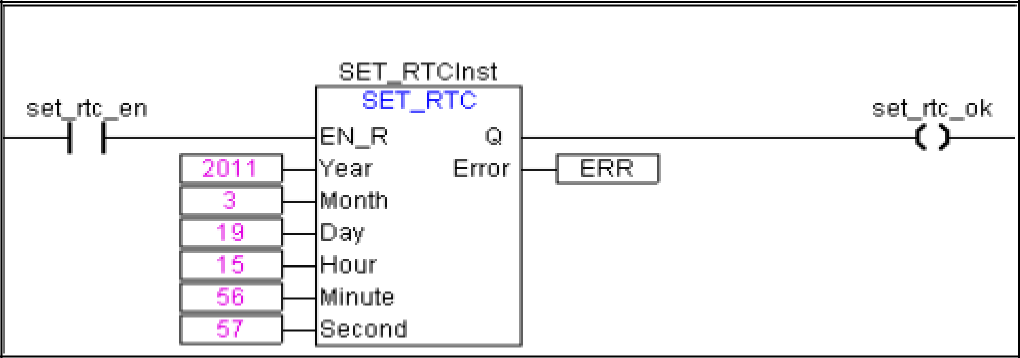
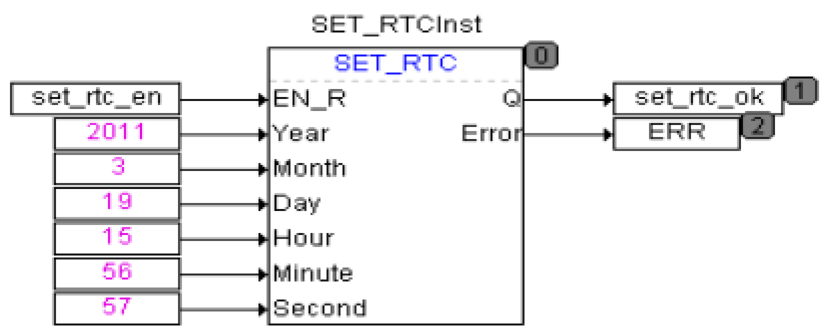
Output Parameter	Data Type	Function Description	Parameter Description
Q	BOOL	Setting completed	0: Not completed 1: Completed

Error	BYTE	Parameter setting error	=0: No error in setting RTC time. =Non 0: indicate error in setting RTC time =1: The setting year is out of the specified range =2: The setting month is not reasonable =3: The setting day is not reasonable =4: The setting hour is not reasonable =5: The setting minute is not reasonable =6: The setting second is not reasonable =200: Module is repeatedly defined and called

3.3.4.3.3 Application Example of Instructions

Variable Definition

No.	Variable Name	Address	Variable Description	Variable Type	Initial Value	Power Fail S
0001	set_rtc_en			BOOL	FALSE	FALSE
0002	set_rtc_ck			BOOL	FALSE	FALSE
0003	ERR			BYTE	0	FALSE
0004	SET_RTCInst			SET_RTC		FALSE

Programing language	Program
Ladder diagram (LD)	
Structured text (ST)	SET_RTCInst(EN_R:=set_rtc_en , Year:=2011 , Month:=3 , Day:= 19, Hour:= 15, Minute:= 56, Second:=57 , Q=>set_rtc_ok Error=>ERR);
Continuous Function Chart (CFC)	

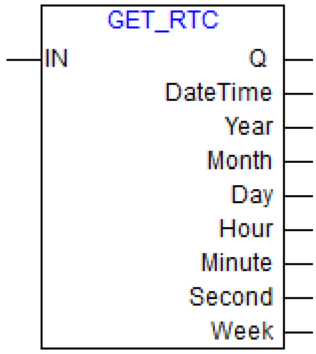
Program description:

When EN_R is set, the date/time as shown in the figure will be set to the real time clock of the PLC hardware. Current real time clock is 15:56:57 on March 19, 2011.

3.3.4.4 GET_RTC(Read Real Time Clock)

3.3.4.4.1 Function

This function block will read the time of the real time clock for the PLC hardware. The time is Eastern 8 District time.



GET_RTC Function Block

3.3.4.4.2 Parameter Description

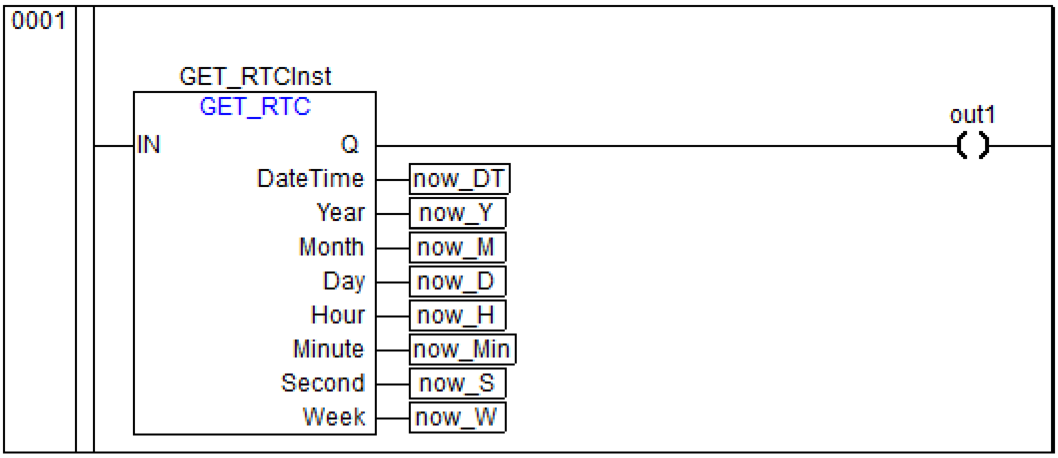
Input Parameter	Data Type	Function Description	Parameter Description	Default Value
IN	BOOL	Enable	0: Invalid 1: Valid	0

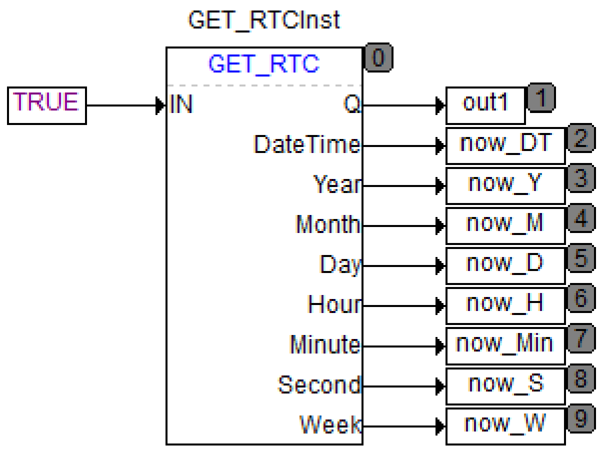
Output Parameter	Data Type	Function Description	Parameter Description	Default Value
Q	BOOL	Whether data is readout	0: Not read out data. 1: Read out data correctly.	FALSE
DateTime	DT	Date/Time	LE platform: DT#2000-01-01-00:00:00~ DT#2099-12-31-23:59:59 MC platform: DT#1971-01-01-00:00:00~ DT#2036-12-31-23:59:59 LK platform: DT#2000-01-01-08:00:00~ DT#2036-12-31-23:59:59	DT#1970-01-01-00:00:00
Year	WORD	Year	LE/MKC platform: 2000~2099 MC platform: MC1000 series 1971 ~ 2036, MC2000 series, MC1002R, MC1002E 2000 ~ 2036 LK platform: V2.1.0 before 1971 ~ 2036, V2.1.0 ~V2.8.0 is 2000 ~ 2036, V2.8.1 and later is 2000~2066.	0
Month	BYTE	Month	1-12	0
Day	BYTE	Day	1-31	0
Hour	BYTE	Hour	0-23	0
Minute	BYTE	Minute	0-59	0
Second	BYTE	Second	0-59	0
Week	BYTE	Week	1-7	0

3.3.4.4.3 Application Example of Instructions

Variable Definition

No.	Variable Name	Address	Variable Description	Variable Type	Initial Value	Power Fail Safeguard
0001	now_DT			DT	DT#1970-01-01-00:00:00	FALSE
0002	now_Y			WORD	0	FALSE
0003	now_M			BYTE	0	FALSE
0004	now_D			BYTE	0	FALSE
0005	now_H			BYTE	0	FALSE
0006	now_Min			BYTE	0	FALSE
0007	now_S			BYTE	0	FALSE
0008	now_W			BYTE	0	FALSE
0009	GET_RTCInst			GET_RTC		FALSE
0010	out1			BOOL	FALSE	FALSE

<table>
<tr><td>Programing language</td><td>Program</td></tr>
<tr><td>Ladder diagram (LD)</td><td>
</td></tr>
<tr><td>Structured text (ST)</td><td>GET_RTC(
IN:=TRUE ,
Q=>out1 ,
DateTime=>now_DT ,
Year=>now_Y ,
Month=>now_M ,
Day=>now_D ,
Hour=>now_H ,
Minute=>now_Min ,
Second=>now_S ,
Week=>now_W);</td></tr>
</table>

Variable Definition	
Continuous function diagram (CFC)	

Program description:

When IN is set, DateTime will output the date/time of the PLC real time clock in the DT data format, output Year in the WORD data format, output Month, Date, Hour, Minute, Second and Week in the BYTE data format. Q is equal to TRUE, and current real time clock is 2012-03-19-15:56:57, Monday.

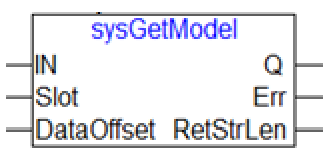
When EN is reset, Q will be equal to FALSE, and the DateTime, Year, Month, Date, Hour, Minute, Second and Week will hold the last output value constantly.

3.3.5 Module Information Instruction

3.3.5.1 SysGetModel (Get Model Type)

3.3.5.1.1 Function

Get the model of modules.



sysGetModel Instruction

3.3.5.1.2 Parameter Description

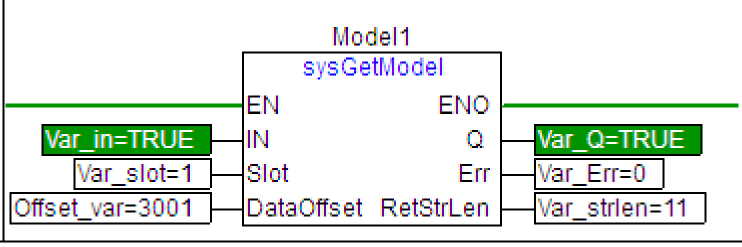
Input Parameter	Type	Parameter Description
IN	BOOL	High level enable (default value: FALSE)
Slot	BYTE	Slot number (default value: 1) The slot number of the master control module is fixed to 1. The slot number of other modules is ascended to the right according to the location related to the master control module. Input the slot number of the version information module to be obtained during the use.
DataOffset	DWORD	M zone offset (the model of modules is stored in the specified offset location in the string format) (default value:0, range of valid value from 0~1048575)

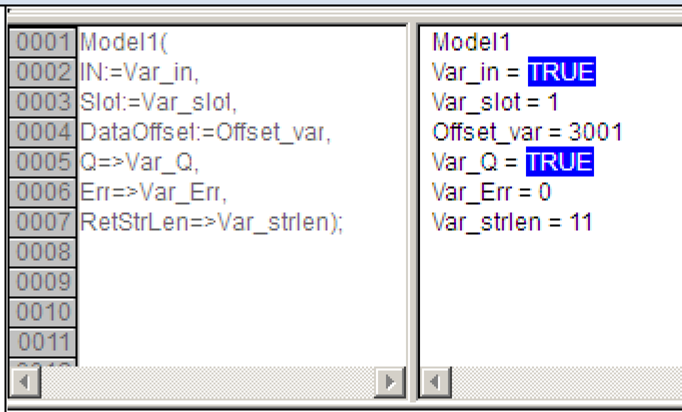
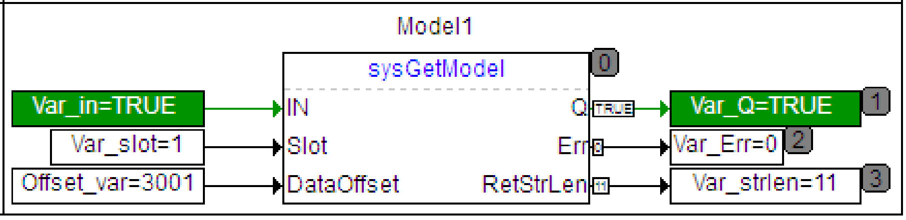
Output Parameter	Type	Parameter Description
Q	BOOL	Output (default value: FALSE)
Err	BYTE	Error (default value: 0) 1: Slot number is error 2: DataOffset configuration is error 128: There is no module in corresponding slot 129: The string returned exceed the scope of M area 130: Information returned is error 131: Slot module type is unknown
RetStrLen	BYTE	String length for model of modules (default value: 0)

3.3.5.1.3 Application Example of Instructions

Variable Definition

No.	Variable Name	Address	Variable Description	Variable Type	Online Value	Power Fail Safegua
0001	Model1			SYSGETMODEL		FALSE
0002	Var_in			BOOL	TRUE	FALSE
0003	Var_slot			BYTE	1	FALSE
0004	Offset_var			DWORD	3001	FALSE
0005	Var_Q			BOOL	TRUE	FALSE
0006	Var_Err			BYTE	0	FALSE
0007	Var_strlen			BYTE	11	FALSE
0008	address_var	%MB3001		STRING(80)	'Model: LK220'	FALSE

Programing language	Program
Ladder diagram (LD)	

Variable Definition	
Structured text (ST)	
Continuous Function Chart (CFC)	

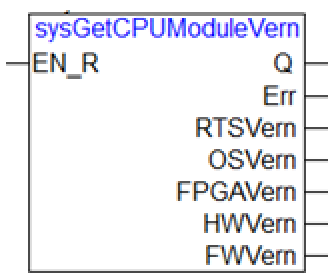
Program description:

User need to define a string variable which used to store the model read, such as variable address_var in instance is associated to address in M area. The value of parameter DataOffset quote the address in M area, such as value of Offset_var is set to 3001.

3.3.5.2 SysGetCPUModuleVern (Get Version of CPU)

3.3.5.2.1 Function

Get firmware version information of controller, which is displayed in decimal mode.



sysGetCPUModuleVern Instruction

3.3.5.2.2 Parameter Description

Input Parameter	Type	Parameter Description
EN_R	BOOL	Enable FALSE: Invalid (default value) TRUE: Rising edge is valid

Output Parameter	Type	Parameter Description
Q	BOOL	Output (FALSE, by default)
Err	BYTE	Error (0, by default) 128: Get version is failed
RTSVern	DWORD	RTS firmware version High 16 bits are RTS version of fast kernel HEROS, corresponding to the RTOS_ RTS version Vx. y.z in Assistant tool Low 16 bits are RTS version of slow kernel LINUX, corresponding to the LINUX_RTS version Vx. y.z in Assistant tool Among them, the high 5 bits represent x, the middle 6 bits represent y, and the last 5 bits represent z
OSVern	DWORD	OS version number High 16 bits are OS version of fast core HEROS, corresponding to the HEROS version V x. y.z in Assistant tool Low 16 bits are OS version of slow core LINUX, corresponding to the LINUX version V x. y.z in Assistant tool Among them, the high 5 bits represent x, the middle 6 bits represent y, and the last 5 bits represent z
FPGAVern	WORD	FPGA version, corresponding to the FPGA version V x. y.z in Assistant tool 16 bits, the high 5 bits represent x, the middle 6 bits represent y, and the last 5 bits represent z
HWVern	WORD	Hardware version number The return value is 1 ~ 26, which corresponds to A~ Z
FWVern	WORD	Module firmware version, corresponding to the SW version V x. y.z in Assistant tool 16 bits, the high 5 bits represent x, the middle 6 bits represent y, and the last 5 bits represent z

3.3.5.2.3 Application Example of Instructions

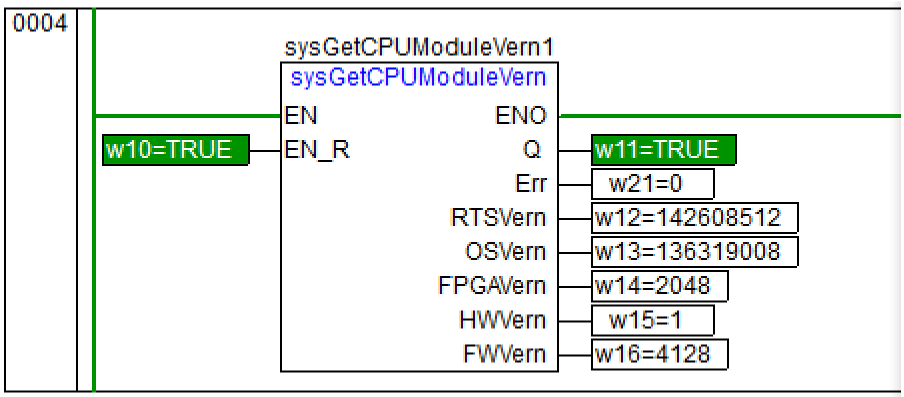
Get controller version information through the sysgetcpumodulevern instruction and assistant tool, as shown in the following figure.

Take RTS version information as an example.

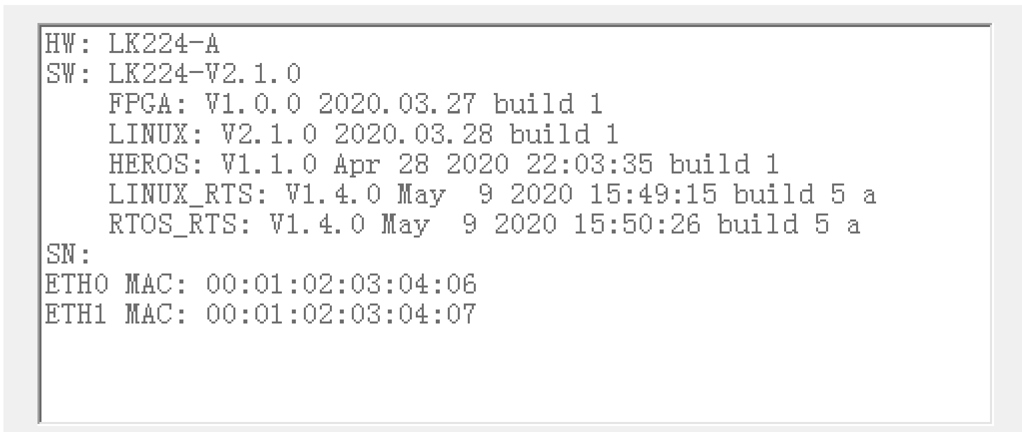
RTSVern outputs 142608512 (0x08800880), in which the high 16 bits represent the RTS version of fast core and the low 16 bits represent the RTS version of slow core.

The high 16 bits 0x0880 corresponds to the binary 0000 1000 1000 0000b, the high 5 bits value is 1, the middle 6 bits value is 4, and the last 5 bits value is 0, the version is V1.4.0, corresponding to RTOS_RTS: V1.4.0 in line 7 of assistant tool.

The low 16 bits 0x0880 corresponds to the binary 0000 1000 1000 0000b, the high 5 bits value is 1, the middle 6 bits value is 4, and the last 5 bits value is 0, the version is V1.4.0, corresponding to LINUX_RTS: V1.4.0 in line 6 of assistant tool.



Get Controller Version by Instruction

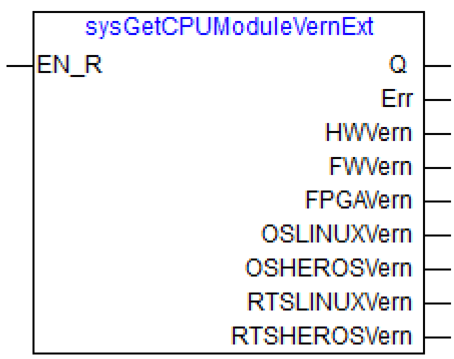


Get Controller Version by Assistant Tool

3.3.5.3 sysGetCPUModuleVernExt (Get Extended Version of CPU)

3.3.5.3.1 Function

Get firmware version information of controller, which is displayed in string format. It is the same as the "sysGetCPUModuleVern" instruction, only the version information is displayed in different format.



sysGetCPUModuleVernExt Instruction

3.3.5.3.2 Parameter Description

Input Parameter	Type	Parameter Description
EN_R	BOOL	Enable FALSE: Invalid (default value) TRUE: Rising edge is valid

Output Parameter	Type	Parameter Description
Q	BOOL	Output (FALSE, by default)
Err	BYTE	Error (0, by default) 128: Get version is failed
HWVern	STRING	Hardware version, corresponding to HW version information in Assistant tool Return value is 1-26, corresponding to A-Z, display format 'A' ~ 'Z'
FWVern	STRING	Module firmware version, corresponding to SW version V x.y.z in Assistant tool
FPGAVern	STRING	FPGA version, corresponding to FPGA version V x.y.z in Assistant tool
OSLINUXVern	STRING	OS version of LINUX, corresponding to LINUX version V x.y.z in Assistant tool
OSHEROSVern	STRING	OS version of HEROS, corresponding to HEROS version V x.y.z in Assistant tool
RTSLINUXVern	STRING	RTS version of LINUX, corresponding to LINUX_RTS version V x.y.z in Assistant tool
RTSHEROSVern	STRING	RTS version of HEROS, corresponding to RTOS_RTS version V x.y.z in Assistant tool

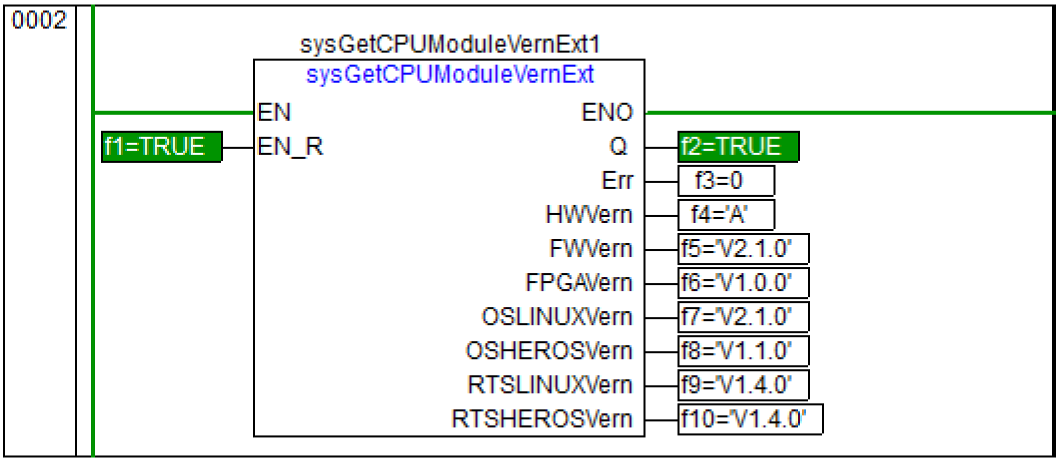
3.3.5.3.3 Application Example of Instructions

Get controller version information through the sysgetcpumodulevernExt instruction and assistant tool, as shown in the following figure.

Take HW and FPGA version information as an example.

HWVern outputs string 'A', corresponding to 'A' in the first line of assistant tool HW: LK224-A, indicating that the hardware version is A.

FPGAVern outputs string V1.0.0, corresponding to the third line of assistant tool FPGA: V1.0.0, indicating that the FPGA version is V1.0.0.



Get Controller Version by Instruction

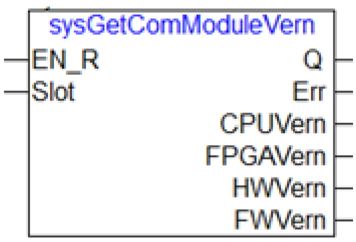
```
HW: LK224-A
SW: LK224-V2.1.0
    FPGA: V1.0.0 2020.03.27 build 1
    LINUX: V2.1.0 2020.03.28 build 1
    HEROS: V1.1.0 Apr 28 2020 22:03:35 build 1
    LINUX_RTS: V1.4.0 May  9 2020 15:49:15 build 5 a
    RTOS_RTS: V1.4.0 May  9 2020 15:50:26 build 5 a
SN:
ETH0 MAC: 00:01:02:03:04:06
ETH1 MAC: 00:01:02:03:04:07
```

Get Controller Version by Assistant Tool

3.3.5.4 SysGetComModuleVern (Get Version of Communication Module)

3.3.5.4.1 Function

Get the firmware version information of the communication module.



sysGetComModuleVern Instruction

3.3.5.4.2 Parameter Description

Input Parameter	Type	Parameter Description
EN_R	BOOL	Enable FALSE: Invalid (default value) TRUE: Rising edge is valid
Slot	BYTE	Slot number (2, by default, valid value is 2~5) The slot number of the master control module is fixed to 1 The slot number of other modules is ascended to the right according to the location related to the master control module. Input the slot number of the version information module to be obtained during the use

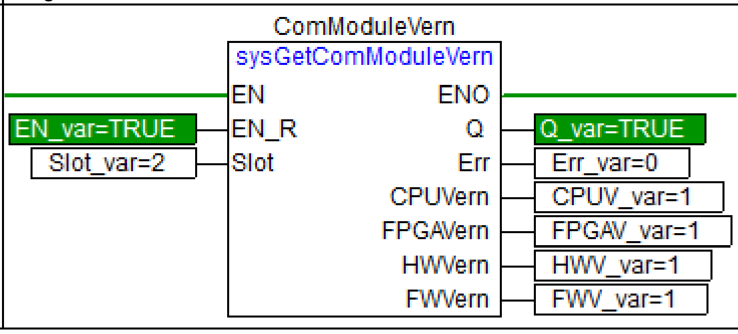
Output Parameter	Type	Parameter Description

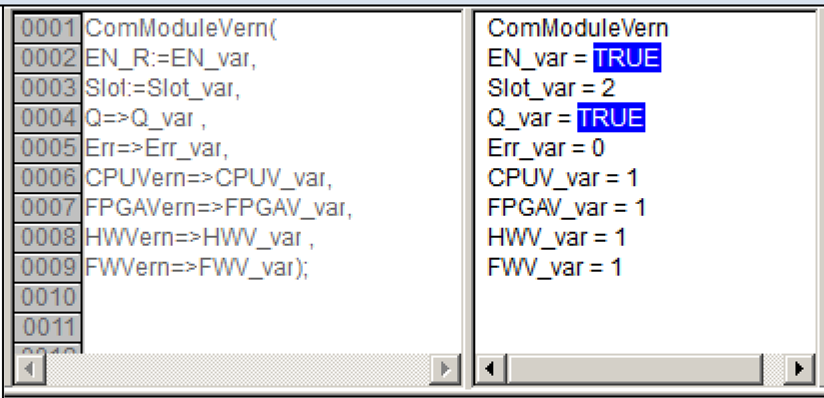
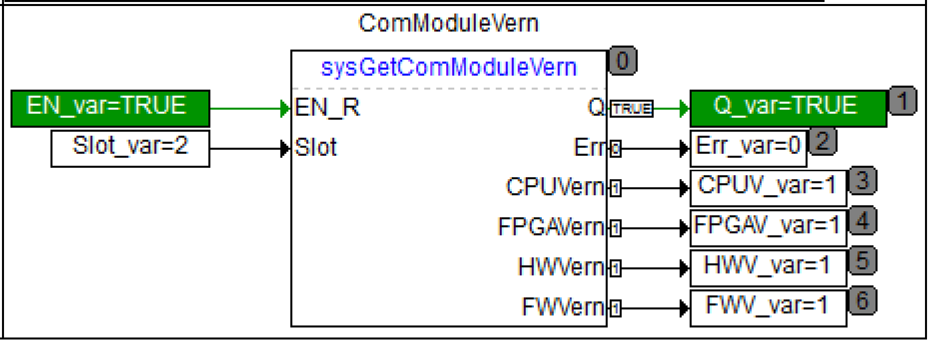
Q	BOOL	Output (FALSE, by default)
Err	BYTE	Error (0, by default) 0: Correct 1: Slot configuration is error 128: There is no module in corresponding slot 129: Get failed 200: Duplicate definition module and call
CPUVern	WORD	CPU firmware version number
FPGAVern	WORD	FPGA version number
HWVern	WORD	Hardware version number The return value is 1~26, which corresponds to A ~ Z
FWVern	WORD	Module firmware version number The shown version number is 01 in LK240-A01

3.3.5.4.3 Application Example of Instructions

Variable Definition

No.	Variable Name	Address	Variable Description	Variable Type	Online Value	Power Fail Safeguard
0001	EN_var			BOOL	TRUE	FALSE
0002	Slot_var			BYTE	1	FALSE
0003	Q_var			BOOL	TRUE	FALSE
0004	Err_var			BYTE	0	FALSE
0005	CPUV_var			WORD	1	FALSE
0006	FPGAV_var			WORD	1	FALSE
0007	HWV_var			WORD	1	FALSE
0008	FWV_var			WORD	1	FALSE
0009	ComModuleVern			SYSGETCOMMODULEVERN		FALSE

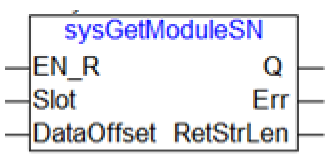
Programing language	Program
Ladder diagram (LD)	

Variable Definition	
Structured text (ST)	
Continuous Function Chart (CFC)	

3.3.5.5 SysGetModuleSN (Get Serial Number of Modules)

3.3.5.5.1 Function

Get the serial number of modules.



sysGetModuleSN Instruction

3.3.5.5.2 Parameter Description

Input Parameter	Type	Parameter Description
EN_R	BOOL	Enable FALSE: Invalid (default value) TRUE: Rising edge is valid
Slot	BYTE	Slot number (1, by default,valid value from 1~5) The slot number of the master control module is fixed to 1 The slot number of other modules is ascended to the right according to the location related to the master control module Input the slot number of the version information module to be obtained during the use
DataOffset	DWORD	M zone offset (the serial number is stored in the specified offset location in the string format)

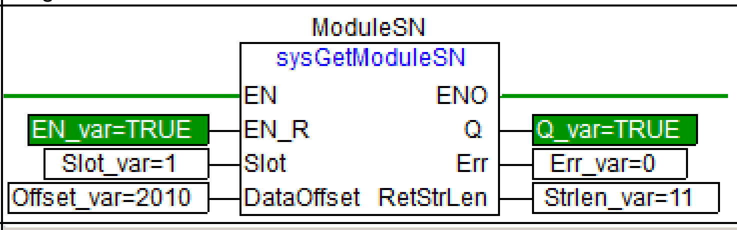
Input Parameter	Type	Parameter Description
		(default value:0, range of valid value from 0~1,048,575)

Output Parameter	Type	Parameter Description
Q	BOOL	Output (FALSE, by default)
Err	BYTE	Error (0, by default) 0: Correct 1: Slot number is error 2: DataOffset configuration is error 128: There is no module in corresponding slot 129: The string returned exceed the scope of M area 130: Information returned is error or the correct data is not obtained temporarily in reading (it will be 0 after reading successfully) 200: Duplicate definition module and call (MC does not support)
RetStrLen	BYTE	String length for serial number of modules (0, by default), up support for 18-bit ASCII string, beyond the content is cut off automatically

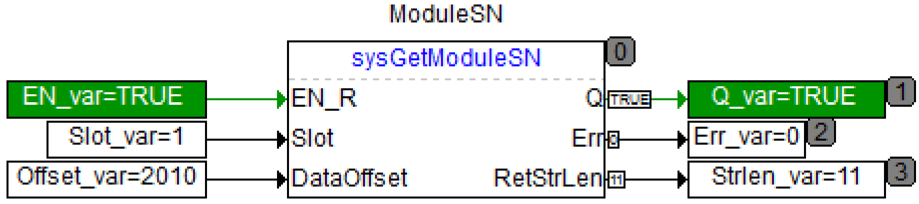
3.3.5.5.3 Application Example of Instructions

Variable Definition

No.	Variable Name	Address	Variable Description	Variable Type	Online Value	Power Fail Safeguard
0001	EN_var			BOOL	TRUE	FALSE
0002	Slot_var			BYTE	1	FALSE
0003	Offset_var			DWORD	2010	FALSE
0004	Q_var			BOOL	TRUE	FALSE
0005	Err_var			BYTE	0	FALSE
0006	Strlen_var			BYTE	11	FALSE
0007	ModuleSN			SYSGETMODULESN		FALSE
0008	address_var	%MB2010		STRING(80)	'01234567891'	FALSE

<table>
<tr><td>Programing language</td><td>Program</td></tr>
<tr><td>Ladder diagram (LD)</td><td>
</td></tr>
<tr><td>Structured text (ST)</td><td>ModuleSN(
EN_R:=EN_var,
Slot:=Slot_var,
DataOffset:=Offset_var,
Q=>Q_var,
Err=>Err_var,
RetStrLen=>Strlen_var);

ModuleSN
EN_var = TRUE
Slot_var = 1
Offset_var = 2010
Q_var = TRUE
Err_var = 0
Strlen_var = 11</td></tr>
</table>

Variable Definition	
Continuous Function Chart (CFC)	

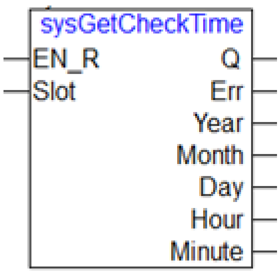
Program description:

User need to define a string variable which used to store the model SN, such as variable address_var in instance is associated to address in M area. The value of parameter DataOffset quote the address in M area, such as value of Offset_var is set to 2010.

3.3.5.6 SysGetCheckTime (Get Check Time)

3.3.5.6.1 Function

Get the check time.



sysGetCheckTime Instruction

3.3.5.6.2 Parameter Description

Input Parameter	Type	Parameter Description
EN_R	BOOL	Enable FALSE: Invalid (default value) TRUE: Rising edge is valid
Slot	BYTE	Slot number (1, by default, valid value from 1~5) The slot number of the master control module is fixed to 1 The slot number of other modules is ascended to the right according to the location related to the master control module. Input the slot number of the version information module to be obtained during the use

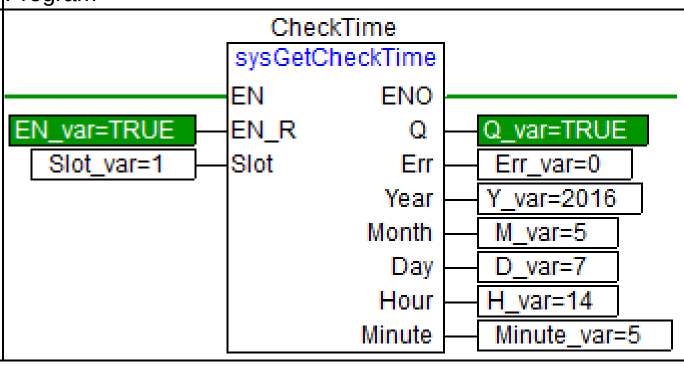
Output Parameter	Type	Parameter Description
Q	BOOL	Output (default value: FALSE)
Err	BYTE	Error (default value: 0) 0: Correct 1: Slot configuration is error 128: There is no module in corresponding slot

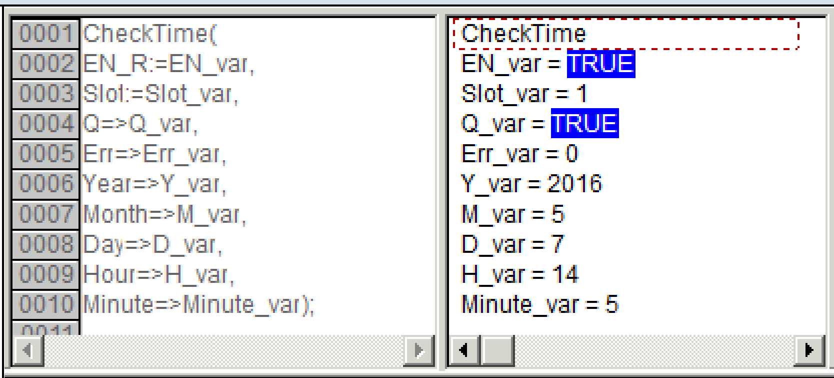
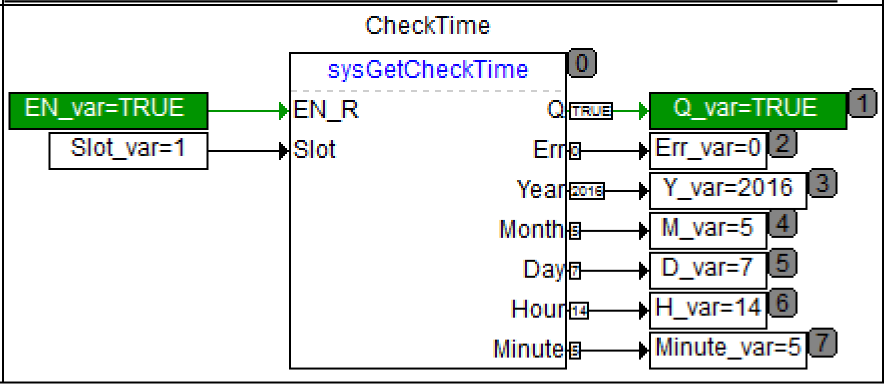
		129: Get failed 130: Year error(2015~2099) 131: Month error 132: Day error 133: Hour error 134: Minute error 200: Duplicate definition module and call
Year	WORD	Year (default value: 0)
Month	BYTE	Month (default value: 0)
Day	BYTE	Day (default value: 0)
Hour	BYTE	Hour (default value: 0)
Minute	BYTE	Minute (default value: 0)

3.3.5.6.3 Application Example of Instructions

Variable Definition

No.	Variable Name	Address	Variable Description	Variable Type	Online Value	Power Fail Safeguard
0001	EN_var			BOOL	TRUE	FALSE
0002	Slot_var			BYTE	1	FALSE
0003	Q_var			BOOL	TRUE	FALSE
0004	Err_var			BYTE	0	FALSE
0005	Y_var			WORD	2016	FALSE
0006	M_var			BYTE	5	FALSE
0007	D_var			BYTE	7	FALSE
0008	H_var			BYTE	14	FALSE
0009	Minute_var			BYTE	5	FALSE
0010	CheckTime			SYSGETCHECKTIME		FALSE

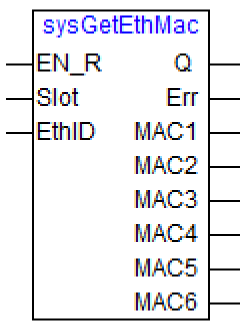
Programing language	Program
Ladder diagram (LD)	

Variable Definition	
Structured text (ST)	
Continuous Function Chart (CFC)	

3.3.5.7 SysGetEthMac (Get MAC Address)

3.3.5.7.1 Function

Get the Mac address.



sysGetEthMac Instruction

3.3.5.7.2 Parameter Description

Input Parameter	Type	Parameter Description
EN_R	BOOL	Enable

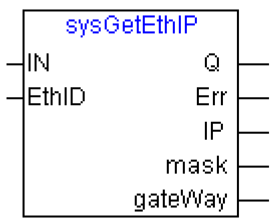
		FALSE: Invalid (by default) TRUE: Rising edge is valid
Slot	BYTE	1:Local controller 2~5: CP module in 2~5 slots
EthID	BYTE	ID number of network interface card (0, by default, valid value is 0-1) 0: First network interface card 1: Second network interface card

Output Parameter	**Type**	**Parameter Description**
Q	BOOL	Output (FALSE, by default)
Err	BYTE	Error (0, by default) 1: EthID configuration is error 128: Get mac address failed 200: Duplicate definition module and call
MAC1	BYTE	First part of MAC address
MAC2	BYTE	Second part of MAC address
MAC3	BYTE	Third part of MAC address
MAC4	BYTE	Fourth part of MAC address
MAC5	BYTE	Fifth part of MAC address
MAC6	BYTE	Sixth part of MAC address

3.3.5.8 sysGetEthIP (Get IP Information)

3.3.5.8.1 Function Description

Obtain the IP address information of the controller.



sysGetEthIP

- Applies to LE5118-B01,LE5119-B01 and above on.

3.3.5.8.2 Parameter Description

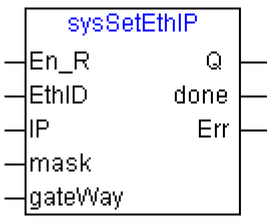
Input Parameter	**Type**	**Parameter Description**
IN	BOOL	Active high
EthID	BYTE	Network card number, LE5118/LE5119 only has one network card, defaults to 0

Output Parameter	Type	Parameter Description
Q	BOOL	Read completion flag FALSE: Setting not completed TRUE: Setting completed
Err	BYTE	Error code 0: Read successfully 1: Network card number error 2: Error in obtaining IP information
IP	DWORD	IP address, 32-bit binary, with each byte representing one bit of the IP. High bytes store the first bit of the IP, such as 192.168.0.250, high bytes store 192, and low bytes store 250
mask	DWORD	Subnet mask, stored in the same format as IP
gateWay	DWORD	Gateway, storage format is the same as IP

3.3.5.9 sysSetEthIP (Set IP Information)

3.3.5.9.1 Function Description

Configure IP addresses for the controller.



sysSetEthIP

- Applies to LE5118-B01,LE5119-B01 and above on.

3.3.5.9.2 Parameter Description

Input Parameter	Type	Parameter Description
EN_R	BOOL	Rising edge effective, rising edge enable configuration
EthID	BYTE	Network card number, LE5118/LE5119 only has one network card, defaults to 0
IP	DWORD	IP address, 32-bit binary, with each byte representing one bit of the IP. High bytes store the first bit of the IP, such as 192.168.0.250, high bytes store 192, and low bytes store 250
mask	DWORD	Subnet mask, stored in the same format as IP
gateWay	DWORD	Gateway, storage format is the same as IP

Output Parameter	Type	Parameter Description
Q	BOOL	Set completion flag TRUE: Setting completed FALSE: Setting not completed
done	BOOL	Set success flag TRUE: Set successfully

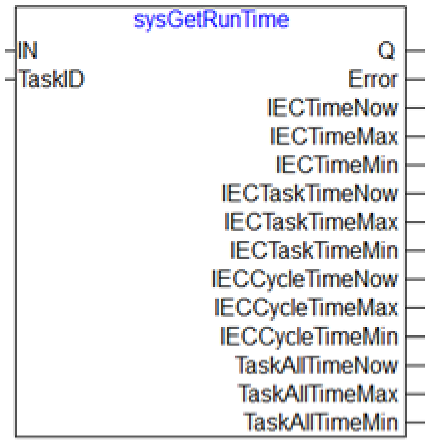
		FALSE: Setting failed or function block not triggered
Err	BOOL	Error code 0: Successfully set, 1: The network card number is incorrect, 2: Error in obtaining IP information 3: Incoming IP check error 4: IP write failure

3.3.6 System State Instruction

3.3.6.1 sysGetRunTime (Get Task Running Time)

3.3.6.1.1 Function

Get the task running time of the controller.



sysGetRunTime Instruction

3.3.6.1.2 Parameter Description

Input Parameter	Type	Parameter Description
IN	BOOL	High level enable (FALSE, by default)
TaskID	BYTE	IEC task ID ,1~8 LK220 supports one IEC task only, default ID is 1, the nine time-data of forward three groups return 0 if ID from 2 to 8

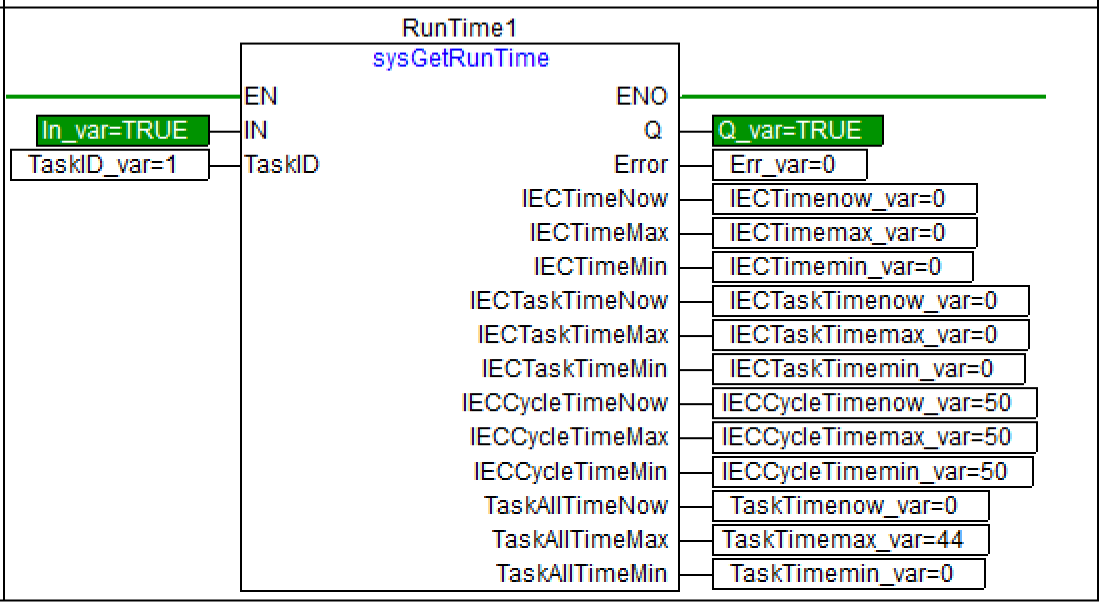
Output Parameter	Type	Parameter Description
Q	BOOL	Output (FALSE, by default)
Error	BYTE	Error (0, by default) 1: TaskID configuration is error
IECTimeNow	DWORD	Current run time of current configuration logic (0, by default)
IECTimeMax	DWORD	Maximum run time of current configuration logic (0, by default)
IECTimeMin	DWORD	Minimum run time of current configuration logic (0, by default)

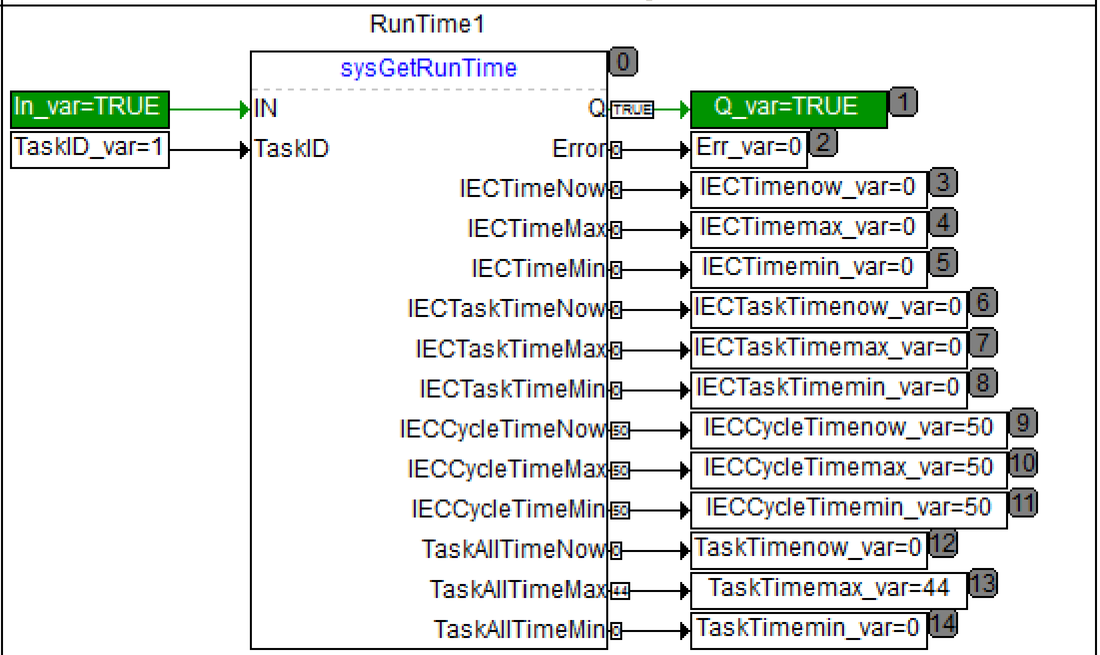
IECTaskTimeNow	DWORD	Current run time of current IEC task (0, by default)
IECTaskTimeMax	DWORD	Maximum run time of current IEC task (0, by default)
IECTaskTimeMin	DWORD	Minimum run time of current IEC task (0, by default)
IECCycleTimeNow	DWORD	Current run time of current IEC task cycle (0, by default)
IECCycleTimeMax	DWORD	Maximum run time of current IEC task cycle (0, by default)
IECCycleTimeMin	DWORD	Minimum run time of current IEC task cycle (0, by default)
TaskAllTimeNow	DWORD	Current run time of current task (0, by default)
TaskAllTimeMax	DWORD	Maximum run time of current task (0, by default)
TaskAllTimeMin	DWORD	Minimum run time of current task (0, by default)

3.3.6.1.3 Application Example of Instructions

Variable Definition

No.	Variable Name	Address	Variable Description	Variable Type	Online Value	Power Fail Safeguard
0001	In_var			BOOL	TRUE	FALSE
0002	TaskID_var			BYTE	1	FALSE
0003	Q_var			BOOL	TRUE	FALSE
0004	Err_var			BYTE	0	FALSE
0005	IECTimenow_var			DWORD	0	FALSE
0006	IECTimemax_var			DWORD	0	FALSE
0007	IECTimemin_var			DWORD	0	FALSE
0008	IECTaskTimenow_var			DWORD	0	FALSE
0009	IECTaskTimemax_var			DWORD	0	FALSE
0010	IECTaskTimemin_var			DWORD	0	FALSE
0011	IECCycleTimenow_var			DWORD	50	FALSE
0012	IECCycleTimemax_var			DWORD	50	FALSE
0013	IECCycleTimemin_var			DWORD	50	FALSE
0014	TaskTimenow_var			DWORD	0	FALSE
0015	TaskTimemax_var			DWORD	44	FALSE
0016	TaskTimemin_var			DWORD	0	FALSE
0017	RunTime1			SYSGETRUNTIME		FALSE

Programing language	Program
Ladder diagram (LD)	

Variable Definition		
Structured text (ST)	RunTime1(IN:=In_var , TaskID:=TaskID_var, Q=>Q_var, Error=>Err_var, IECTimeNow=>IECTimenow_var, IECTimeMax=>IECTimemax_var, IECTimeMin=>IECTimemin_var, IECTaskTimeNow=>IECTaskTimenow_var, IECTaskTimeMax=>IECTaskTimemax_var, IECTaskTimeMin=>IECTaskTimemin_var, IECCycleTimeNow=>IECCycleTimenow_var, IECCycleTimeMax=>IECCycleTimemax_var, IECCycleTimeMin=>IECCycleTimemin_var, TaskAllTimeNow=>TaskTimenow_var, TaskAllTimeMax=>TaskTimemax_var, TaskAllTimeMin=>TaskTimemin_var);	RunTime1 In_var = TRUE TaskID_var = 1 Q_var = TRUE Err_var = 0 IECTimenow_var = 0 IECTimemax_var = 0 IECTimemin_var = 0 IECTaskTimenow_var = 0 IECTaskTimemax_var = 0 IECTaskTimemin_var = 0 IECCycleTimenow_var = 50 IECCycleTimemax_var = 50 IECCycleTimemin_var = 50 TaskTimenow_var = 0 TaskTimemax_var = 44 TaskTimemin_var = 0
Continuous Function Chart (CFC)		

3.3.6.2 sysGetRunTimeInfo (Get Task Running Time information)

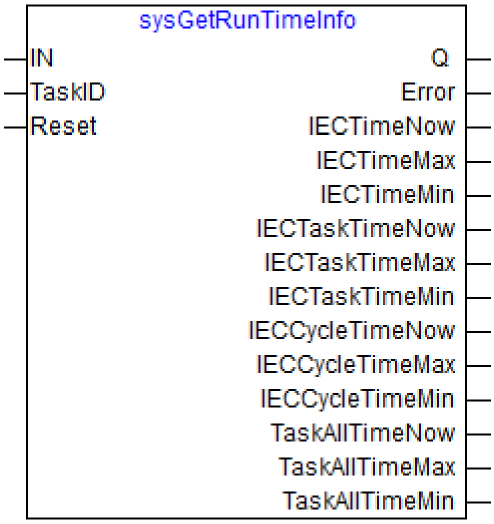
The instruction is supported in V3.1.5B3 and above version.

3.3.6.2.1 Function

Get the task running time information of the controller.

It is recommended that the user reset the instruction before use.

Operations in AutoThink such as downloading and cold reset may lead to longer task running time.



sysGetRunTimeInfo Instruction

3.3.6.2.2 Parameter Description

Input Parameter	Type	Parameter Description
IN	BOOL	High level enable (FALSE, by default)
TaskID	BYTE	IEC task ID,1~8 LK220 supports one IEC task only, default ID is 1, the nine time-data of forward three groups return 0 if ID from 2 to 8
Reset	BOOL	Clear the each time variable of the output variable to zero

Output Parameter	Type	Parameter Description
Q	BOOL	Output (FALSE, by default)
Error	BYTE	Error (0, by default) 1: TaskID configuration is error
IECTimeNow	DWORD	Current run time of current configuration logic (0, by default)
IECTimeMax	DWORD	Maximum run time of current configuration logic (0, by default)
IECTimeMin	DWORD	Minimum run time of current configuration logic (0, by default)
IECTaskTimeNow	DWORD	Current IO refresh run time (0, by default)
IECTaskTimeMax	DWORD	Current IO refresh maximum run time (0, by default)
IECTaskTimeMin	DWORD	Current IO refresh minimum run time (0, by default)
IECCycleTimeNow	DWORD	The current set task cycle (0, by default)
IECCycleTimeMax	DWORD	The current set task cycle (0, by default)
IECCycleTimeMin	DWORD	The current set task cycle (0, by default)
TaskAllTimeNow	DWORD	Current run time of current task (0, by default)
TaskAllTimeMax	DWORD	Maximum run time of current task (0, by default)
TaskAllTimeMin	DWORD	Minimum run time of current task (0, by default)

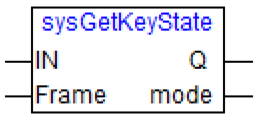
3.3.6.2.3 Application Example of Instructions

Please see Application Example of Instructions of sysGetRunTime (Get Task Running Time).

3.3.6.3 SysGetKeyState (Get Switch State of Key)

3.3.6.3.1 Function

Get the key switch state of the CPU module for the master frame.



sysGetKeyState Instruction

3.3.6.3.2 Parameter Description

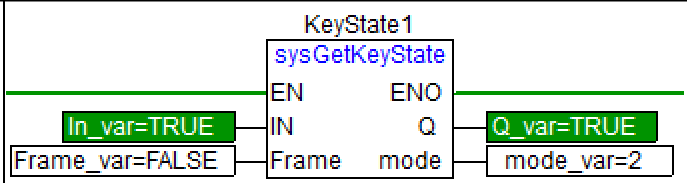
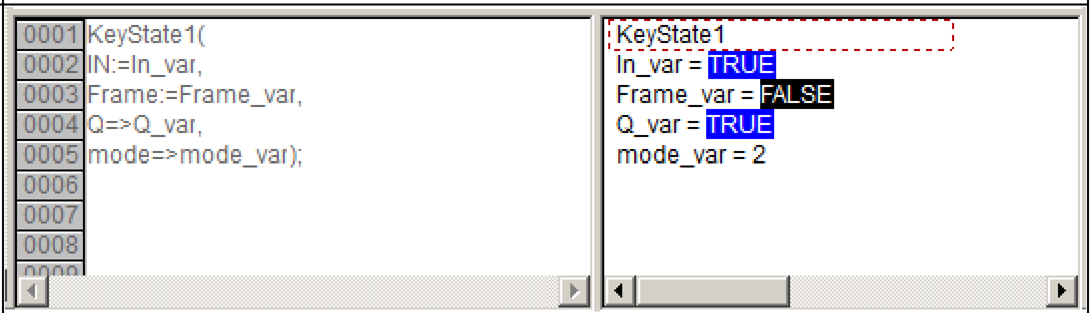
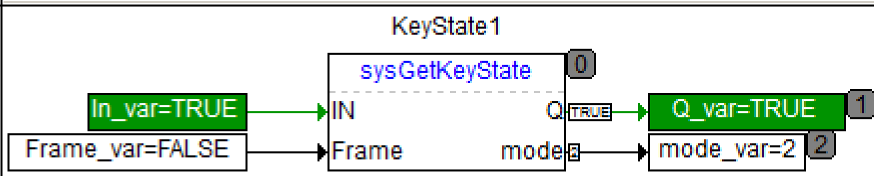
Input Parameter	Type	Parameter Description
IN	BOOL	High level enable (FALSE, by default)
Frame	BOOL	Select frame (FALSE, by default) TRUE:Slave frame FALSE:Master frame

Output Parameter	Type	Parameter Description
Q	BOOL	Output (FALSE, by default)
Mode	BYTE	Key switch state (0, by default) 1: RUN 2: REM 3: PRG

3.3.6.3.3 Application Example of Instructions

Variable Definition

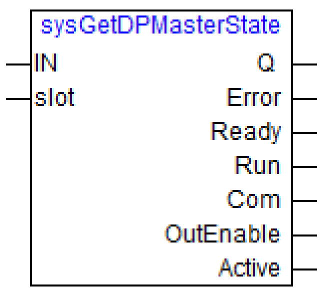
No.	Variable Name	Address	Variable Description	Variable Type	Online Value	Power Fail Safeguard
0001	In_var			BOOL	TRUE	FALSE
0002	Frame_var			BOOL	FALSE	FALSE
0003	Q_var			BOOL	TRUE	FALSE
0004	mode_var			BYTE	2	FALSE
0005	KeyState1			SYSGETKEYSTATE		FALSE

Variable Definition	
Programing language	Program
Ladder diagram (LD)	
Structured text (ST)	
Continuous Function Chart (CFC)	

3.3.6.4 SysGetDPMasterState (Get Current State of DP Master)

3.3.6.4.1 Function

Get the diagnosis information of the DP master station.



sysGetDPMasterState Instruction

3.3.6.4.2 Parameter Description

Input Parameter	Type	Parameter Description
IN	BOOL	High level enable (FALSE, by default)

Input Parameter	Type	Parameter Description
Slot	BYTE	Slot number (2, by default, valid value from 2~5) The slot number of the master control module is fixed to 1 The slot number of other modules is ascended to the right according to the location related to the master control module Input the slot number of the version information module to be obtained during the use

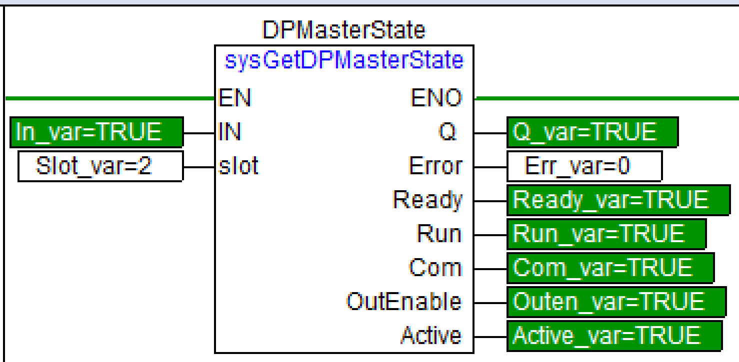
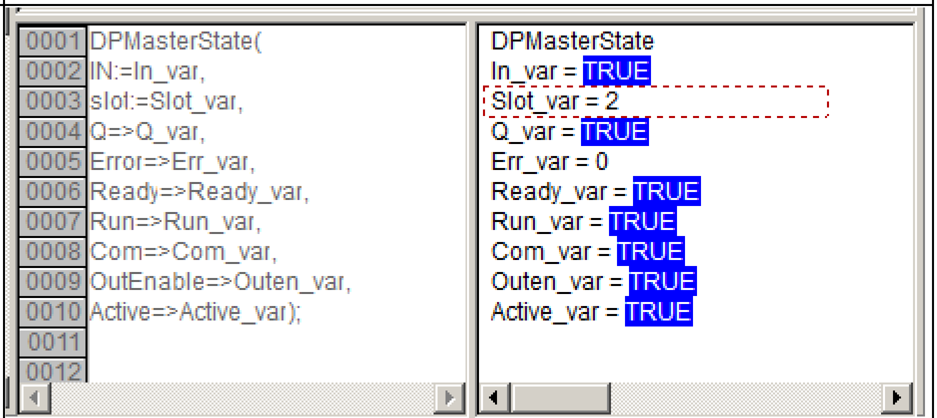
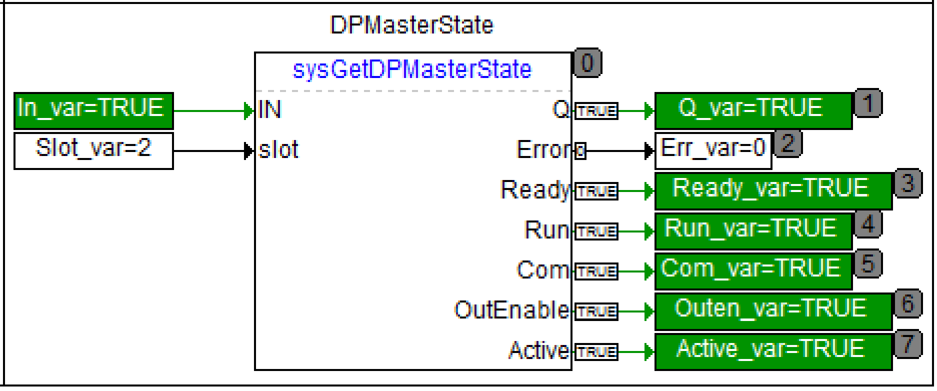
Output Parameter	Type	Parameter Description
Q	BOOL	Output (FALSE, by default)
Error	BYTE	Error (0, by default) 0: Correct 1: Slot configuration is error 128: The DP master station is not detected in corresponding slot
Ready	BOOL	Identification about whether the master station is ready FALSE: The initialization of the master station is not completed TRUE: The initialization of the master station is completed
Run	BOOL	Identification about whether the master station runs FALSE: The master station doesn’t receive the downloading data successfully TRUE: The master station receives the downloading data successfully, and starts to communicate with the slave station
Com	BOOL	Identification about whether the master station establishes the data exchange with the slave station FALSE: Don’t establish the data exchange with the slave station TRUE: Establish the data exchange with a slave station at least
OutEnalbe	BOOL	Identification about master station output enable FALSE: Output not enabled TRUE: Output enabled
Active	BOOL	Identification about master station active state FALSE: Hot standby state TRUE: Active state

3.3.6.4.3 Application Example of Instructions

Variable Definition

No.	Variable Name	Address	Variable Description	Variable Type	Online Value	Power Fail Safeguar
0001	In_var			BOOL	TRUE	FALSE
0002	Slot_var			BYTE	2	FALSE
0003	Q_var			BOOL	TRUE	FALSE
0004	Err_var			BYTE	0	FALSE
0005	Ready_var			BOOL	TRUE	FALSE
0006	Run_var			BOOL	TRUE	FALSE
0007	Com_var			BOOL	TRUE	FALSE
0008	Outen_var			BOOL	TRUE	FALSE
0009	Active_var			BOOL	TRUE	FALSE
0010	DPMasterState			SYSGETDPMASTERSTATE		FALSE

Programing language	Program

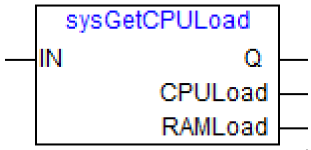
<table>
<tr><th colspan="2">Variable Definition</th></tr>
<tr><td>Ladder diagram (LD)</td><td></td></tr>
<tr><td>Structured text (ST)</td><td></td></tr>
<tr><td>Continuous Function Chart (CFC)</td><td></td></tr>
</table>

3.3.6.5 SysGetCPULoad (Get Current Load of CPU)

3.3.6.5.1 Function

Get the CPU utilization ratio of the controller.

Operations in AutoThink such as downloading and cold reset may lead to higher CPU utilization.



sysGetCPULoad Instruction

3.3.6.5.2 Parameter Description

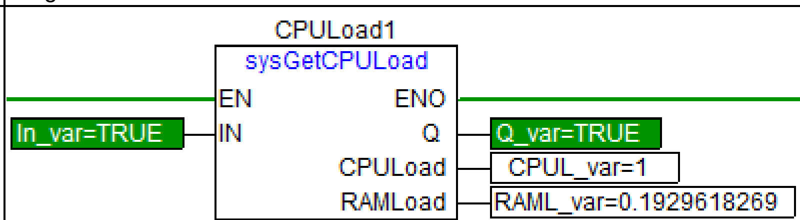
Input Parameter	Type	Parameter Description
IN	BOOL	High level enable (FALSE, by default)

Output Parameter	Type	Parameter Description
Q	BOOL	Output (FALSE, by default)
CPU1Load	REAL	Utilization ratio of CPU (0, by default)
CPU2Load	REAL	Utilization ratio of memory (0, by default)

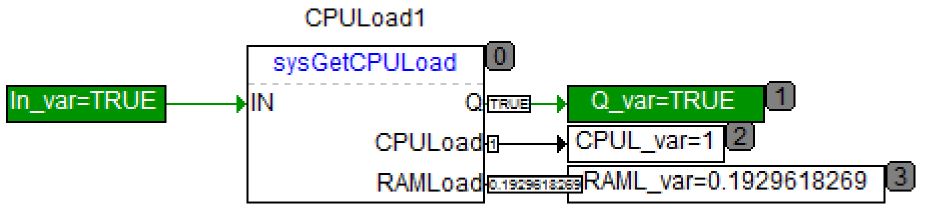
3.3.6.5.3 Application Example of Instructions

Variable Definition

No.	Variable Name	Address	Variable Description	Variable Type	Initial Value	Power Fail Safeguard
0001	In_var			BOOL	TRUE	FALSE
0002	Q_var			BOOL	TRUE	FALSE
0003	CPUL_var			REAL	1	FALSE
0004	RAML_var			REAL	0.1929618269	FALSE
0005	CPULoad1			SYSGETCPULOAD		FALSE

<table>
<tr><td>Programing language</td><td>Program</td></tr>
<tr><td>Ladder diagram (LD)</td><td>
</td></tr>
<tr><td>Structured text (ST)</td><td>0001 CPULoad1(
0002 IN:=In_var,
0003 Q=>Q_var,
0004 CPULoad=>CPUL_var,
0005 RAMLoad=>RAML_var);
0006
0007

CPULoad1
In_var = TRUE
Q_var = TRUE
CPUL_var = 1
RAML_var = 0.1929618269</td></tr>
</table>

Variable Definition	
Continuous Function Chart (CFC)	

3.3.6.6 sysGetHTCP_CycleTime(Get Hollitcp Master Cycle Time)

3.3.6.6.1 Function

Get the cycle time of the IO below the HolliTCP master

3.3.6.6.2 Description of Parameter

Input Parameter	Type of Data	Description of Function
IN	BOOL	High level energy

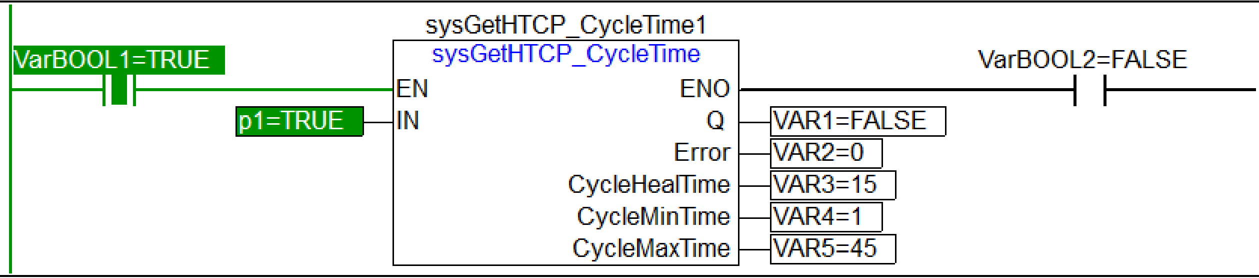
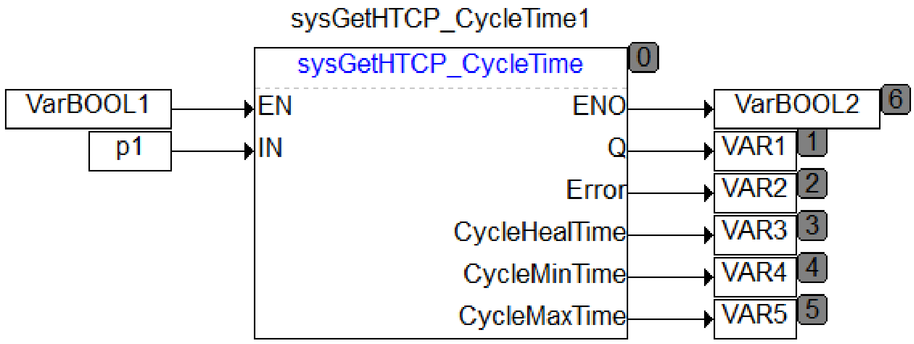
Output Parameter	Type of Data	Description of Function
Q	BOOL	Output FALSE: Output is completion TRUE: Output is not completion
Error	BYTE	Error code 0: Success 128:Hollitcp master cycle not output
CycleHealTime	DWORD	HolliTcp master cycle heal time
CycleMinTime	DWORD	HolliTcp master cycle min time
CycleMaxTime	DWORD	HolliTcpmaster cycle max time

3.3.6.6.3 Application Example of Instructions

Variable Definition

No.	Variable Name	Address	Variable Description	Variable Type	Online Value	Power Fail Safeguard	SOE Enable
0001	sysGetHTCP_CycleTime1			SYSGETHTCP_CYCLETIME		FALSE	
0002	VarBOOL1			BOOL	TRUE	FALSE	
0003	VarBOOL2			BOOL	FALSE	FALSE	
0004	p1			BOOL	TRUE	FALSE	
0005	VAR1			BOOL	FALSE	FALSE	
0006	VAR2			BYTE	0	FALSE	
0007	VAR3			DWORD	15	FALSE	
0008	VAR4			DWORD	1	FALSE	
0009	VAR5			DWORD	45	FALSE	

Programing	Program

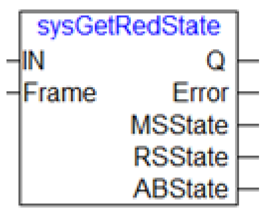
Variable Definition	
language	
Ladder diagram (LD)	
Structured text (ST)	sysGetHTCP_CycleTime1(IN:=p1, Q=>VAR1, Error=>VAR2, CycleHealTime=>VAR3, CycleMinTime=>VAR4, CycleMaxTime=>VAR5);
Continuous Function Chart (CFC)	

3.3.7 Redundancy Instruction

3.3.7.1 SysGetRedState (Get Redundant State of System)

3.3.7.1.1 Function

Get the redundancy state information of the master and slave frames.



sysGetRedState Instruction

3.3.7.1.2 Parameter Description

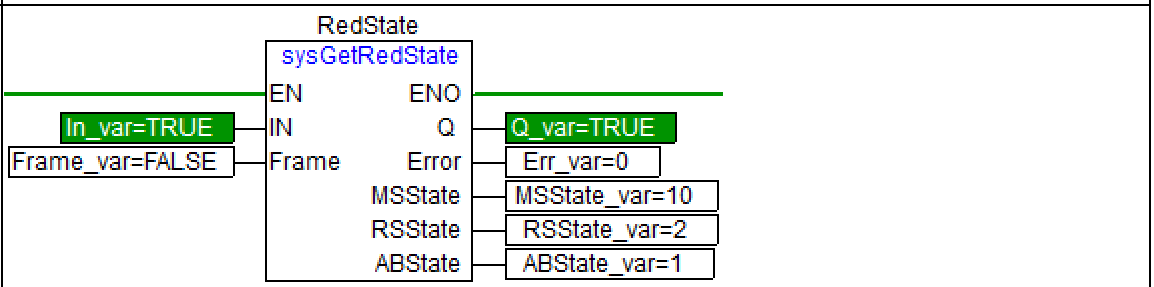
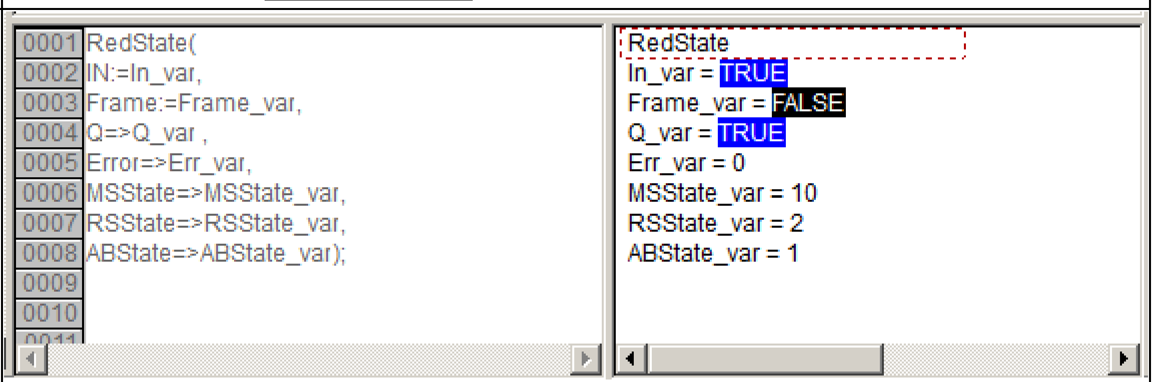
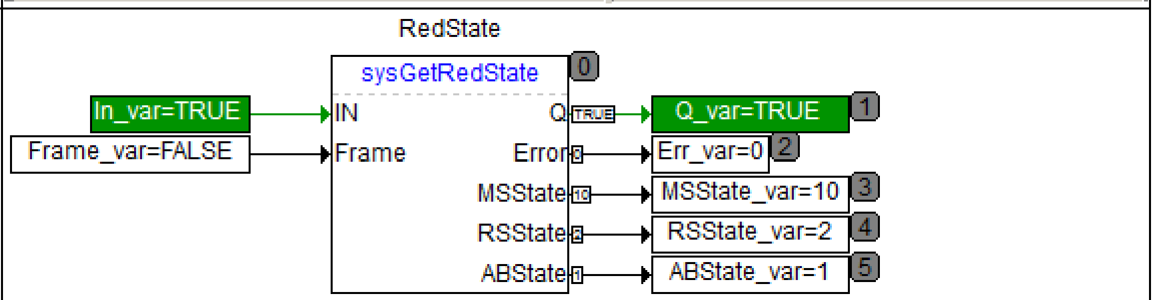
Input Parameter	Type	Parameter Description
IN	BOOL	High level enable (FALSE, by default)
Frame	BOOL	Identification about master and slave frames FALSE: Master frame TRUE: Slave frame

Output Parameter	Type	Parameter Description
Q	BOOL	Output (FALSE, by default)
Error	BYTE	Error (0, by default) 128: Current CPU is redundancy abnormal state (corresponding to 0-6 and 11 in states of master and slave frames) 129: Key switch sate of master CPU and slave CPU is inconsistent 130: The task running time exceeds the set task cycle in the dual computer system 131: Redundancy link is abnormal between the master and slave
MSState	BYTE	State of master and slave frames (0, by default) 0: Unknown state 1: InitialSlave stateframe 2: Hardware Ready stateMaster frame 3: Dual frames Ready state 4: Standalone frame Ready state 5: Failure state 6. Error state 7: Engineering redundancy 8: Engineering verification 9: Slave frame 10: Master frame 11: Unqualified master frame
RSState	BYTE	Standalone and dual frames state (0, by default) 0: Unknown state 1: Standalone frame 2: Dual frames
ABState	BYTE	AB frames state (0, by default) 0: Unknown state 1: A frame 2: B frame

3.3.7.1.3 Application Example of Instructions

Variable Definition

No.	Variable Name	Address	Variable Description	Variable Type	Online Value	Power Fail Safegu
0001	In_var			BOOL	TRUE	FALSE
0002	Frame_var			BOOL	FALSE	FALSE
0003	Q_var			BOOL	TRUE	FALSE
0004	Err_var			BYTE	0	FALSE
0005	MSState_var			BYTE	10	FALSE
0006	RSState_var			BYTE	2	FALSE
0007	ABState_var			BYTE	1	FALSE
0008	RedState			SYSGETREDSTATE		FALSE

<table>
<tr><td>Programing language</td><td>Program</td></tr>
<tr><td>Ladder diagram (LD)</td><td>
</td></tr>
<tr><td>Structured text (ST)</td><td>
</td></tr>
<tr><td>Continuous Function Chart (CFC)</td><td>
</td></tr>
</table>

3.3.7.2 SysMasterSwitchToSlave (Master Switch to Slave)

3.3.7.2.1 Function

Switch the master frame to the slave frame.

sysMasterSwitchToSlave Instruction

3.3.7.2.2 Parameter Description

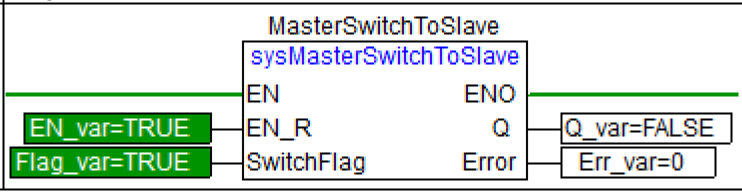
Input Parameter	Type	Parameter Description
EN_R	BOOL	Enable FALSE: Invalid (by default) TRUE: Rising edge effective
SwitchFlag	BOOL	FALSE: Not allowed to switch TRUE: Allowed to switch

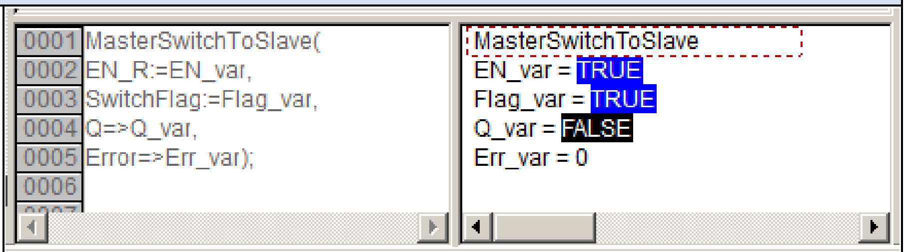
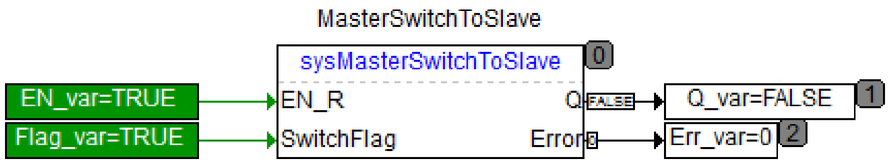
Output Parameter	Type	Parameter Description
Q	BOOL	Output (FALSE, by default)
Error	BYTE	Error (0, by default) 0: Correct 128: Standalone system 129: The master/slave synchronization is not completed 130: The key swtich of the slave frame is in the PRG state

3.3.7.2.3 Application Example of Instructions

Variable Definition

序号	变量名	直接地址	变量说明	变量类型	在线值	掉电保护
0001	EN_var			BOOL	TRUE	FALSE
0002	Flag_var			BOOL	TRUE	FALSE
0003	Q_var			BOOL	FALSE	FALSE
0004	Err_var			BYTE	0	FALSE
0005	MasterSwitchToSlave			SYSMASTERSWITCHTOSLAVE		FALSE

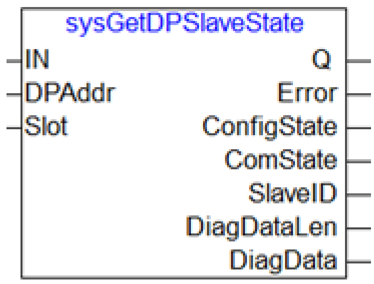
Programing language	Program
Ladder diagram (LD)	

Variable Definition	
Structured text (ST)	
Continuous Function Chart (CFC)	

3.3.7.3 SysGetDPSlaveState (Get Diagnosis of DP Slave)

3.3.7.3.1 Function

Get the diagnosis information of the DP slave station.



sysGetDPSlaveState Instruction

3.3.7.3.2 Parameter Description

Input Parameter	Type	Parameter Description
IN	BOOL	High level enable (FALSE, by default)
DPAddr	BYTE	Slave station address (default value is 0, valid value from 2~125)
Slot	BYTE	Slot that DP master card is located

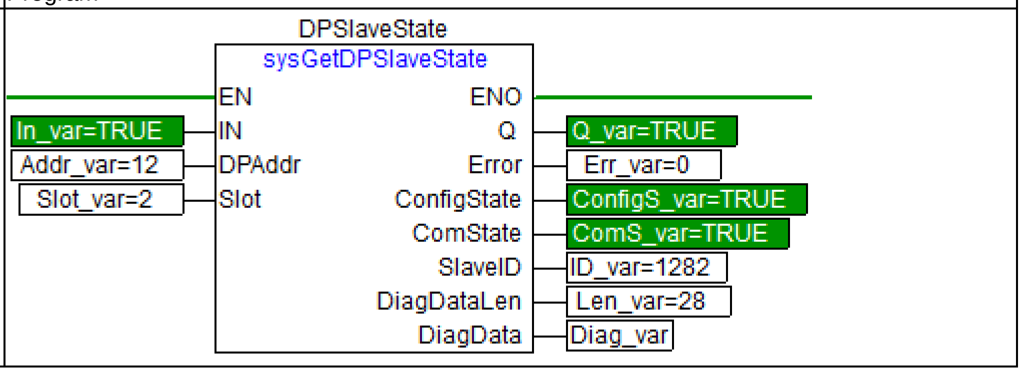
Output Parameter	Type	Parameter Description
Q	BOOL	Output (FALSE, by default)
Error	BYTE	Error (0, by default) 0: Correct 1:Slot number is error 2: Slave address configuration is error 128: There is no DP master card or slot number does not match

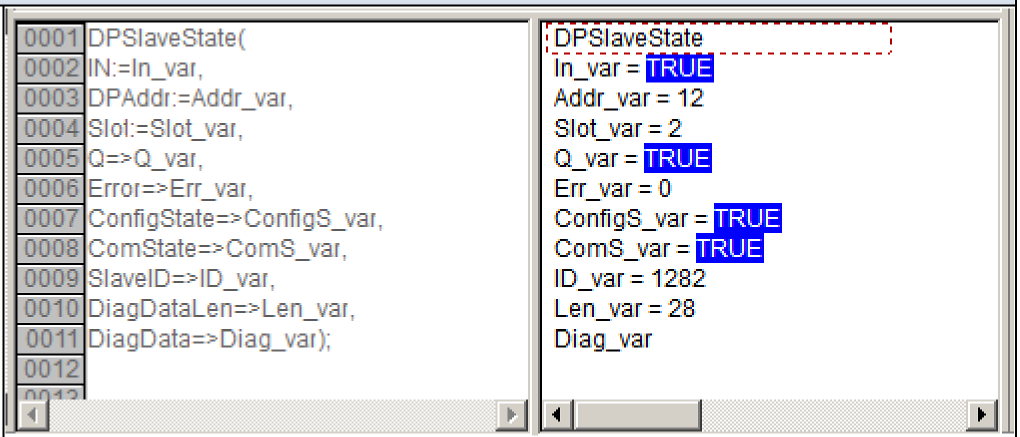
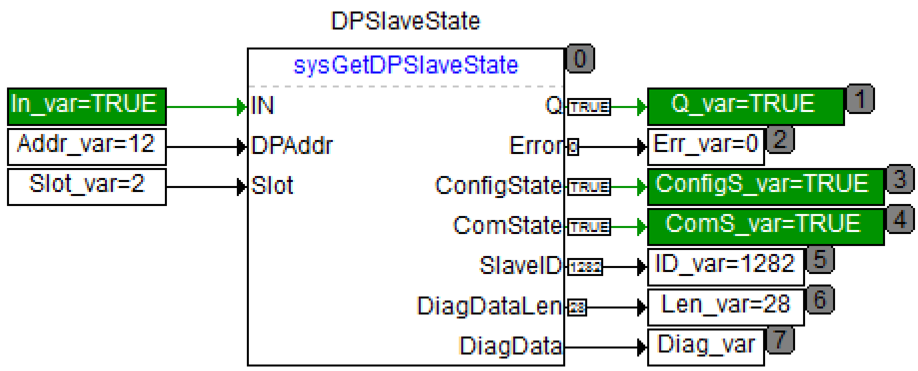
Output Parameter	Type	Parameter Description
ConfigState	BOOL	Whether configuration identification for Slave station TRUE: Configured FALSE: not be Configured
ComState	BOOL	State identification for Slave station communication TRUE: Normal FALSE: Abnormal
SlaveID	WORD	Slave ID (0, by default) Note: ID range for LK Slave station mosule is 0x0500 ~ 0x05FF
DiagDataLen	BYTE	Diagnostic information length of Slave station
DiagData	ARRAY[0…51] OF BYTE	Diagnostic information, reference to diagnosis in each module of manual *LK Series Programmable Logic Controller System Manual* DiagData[50]: Fault status of slave station DP network bit0: Fault status of DP1 network, 1 is fault, 0 is normal bit1: Fault status of DP2 network, 1 is fault, 0 is normal

3.3.7.3.3 Application Example of Instructions

Variable Definition

No.	Variable Name	Address	Variable Description	Variable Type	Online Value	Power Fail Safeguard
0001	In_var			BOOL	TRUE	FALSE
0002	Addr_var			BYTE	12	FALSE
0003	Slot_var			BYTE	2	FALSE
0004	Q_var			BOOL	TRUE	FALSE
0005	Err_var			BYTE	0	FALSE
0006	ConfigS_var			BOOL	TRUE	FALSE
0007	ComS_var			BOOL	TRUE	FALSE
0008	ID_var			WORD	1282	FALSE
0009	Len_var			BYTE	28	FALSE
0010	Diag_var			ARRAY[0..51] OF BYTE		FALSE
0011	DPSlaveState			SYSGETDPSLAVESTATE		FALSE

Programing language	Program
Ladder diagram (LD)	

<table>
<tr><th colspan="2">Variable Definition</th></tr>
<tr><td>Structured text (ST)</td><td>
</td></tr>
<tr><td>Continuous Function Chart (CFC)</td><td>
</td></tr>
</table>

Program description:

Example for getting the diagnosis information of LK411 module whose slave address is 12.

Double-click the array variable Diag_var to view the diagnosis information, as shown in figure.

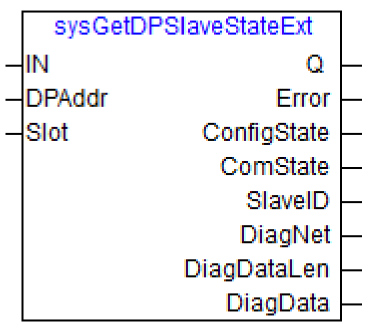
No.	Variable Name	Address	Variable Description	Variable Type	Online Value	Power Fail Safeguard
0001	Diag_var[0]			BYTE	2	FALSE
0002	Diag_var[1]			BYTE	0	FALSE
0003	Diag_var[2]			BYTE	66	FALSE
0004	Diag_var[3]			BYTE	1	FALSE
0005	Diag_var[4]			BYTE	128	FALSE
0006	Diag_var[5]			BYTE	64	FALSE
0007	Diag_var[6]			BYTE	160	FALSE
0008	Diag_var[7]			BYTE	128	FALSE
0009	Diag_var[8]			BYTE	65	FALSE
0010	Diag_var[9]			BYTE	160	FALSE
0011	Diag_var[10]			BYTE	128	FALSE
0012	Diag_var[11]			BYTE	66	FALSE
0013	Diag_var[12]			BYTE	160	FALSE
0014	Diag_var[13]			BYTE	128	FALSE
0015	Diag_var[14]			BYTE	67	FALSE
0016	Diag_var[15]			BYTE	160	FALSE

View the Diagnosis information

3.3.7.4 sysGetDPSlaveStateExt (Get Diagnosis of DP Slave)

3.3.7.4.1 Function

Get the diagnosis information of DP slave station, which is suitable for the case of large diagnosis data bytes. When data length of diagnosis information exceeds 52 Bytes, it is recommended to use this function block.



sysGetDPSlaveStateExt Instruction

3.3.7.4.2 Parameter Description

Input Parameter	Type	Parameter Description
IN	BOOL	High level enable (FALSE, by default)
DPAddr	BYTE	Slave station address (default value is 0, valid value from 2~125)
Slot	BYTE	Slot that DP master card is located

Output Parameter	Type	Parameter Description
Q	BOOL	Output (FALSE, by default)
Error	BYTE	Error (0, by default) 0: Correct 1:Slot number is error 2: Slave address configuration is error 128: There is no DP master card or slot number does not match
ConfigState	BOOL	Whether configuration identification for Slave station TRUE: Configured FALSE: not be Configured
ComState	BOOL	State identification for Slave station communication TRUE: Normal FALSE: Abnormal
SlaveID	WORD	Slave ID (0, by default) Note: ID range for LK Slave station mosule is 0x0500 ~ 0x05FF
DiagNet	BYTE	Fault status of slave station DP network bit0: Fault status of DP network bit1: Fault status of DP network
DiagDataLen	BYTE	Diagnostic information length of slave station
DiagData	ARRAY[0…237] OF BYTE	Slave diagnostic information, reference to diagnosis in each module

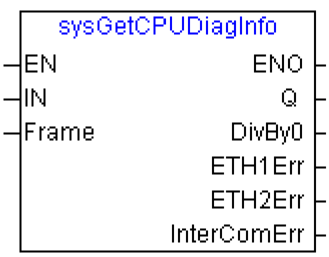
3.3.7.4.3 Application Example of Instructions

Please see sysGetDPSlaveState (Get Diagnosis of DP Slave).

3.3.7.5 SysGetCPUDiagInfo (Get Diagnosis of CPU)

3.3.7.5.1 Function

Get the diagnosis information of the master control module.



sysGetCPUDiagInfo Instruction

3.3.7.5.2 Parameter Description

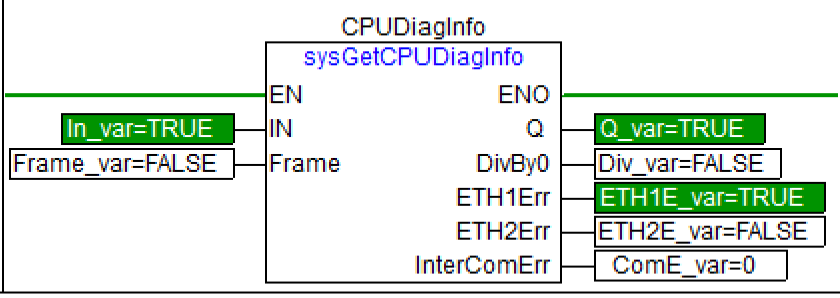
Input Parameter	Type	Parameter Description
IN	BOOL	High level enable (FALSE, by default)
Frame	BOOL	Select frame to be get, FALSE is current frame (master), TRUE is opposite frame (slave)

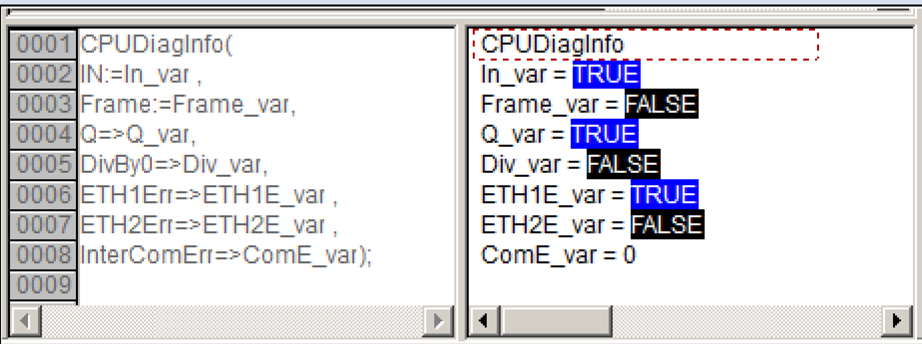
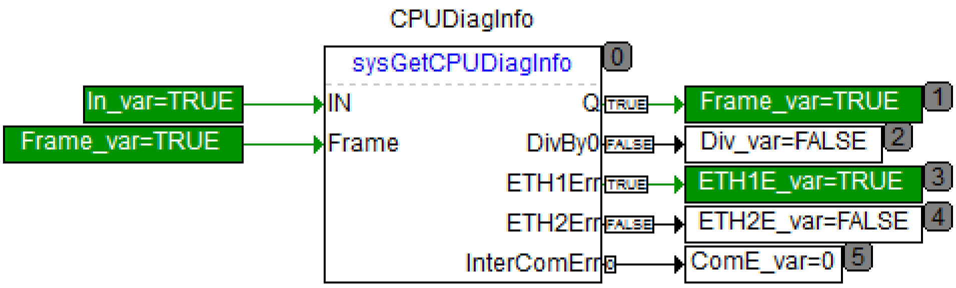
Output Parameter	Type	Parameter Description
Q	BOOL	Output (FALSE, by default)
DivBy0	BOOL	FALSE as default value. When it attempts to divide by zero, DivBy0 is set to TRUE. It is necessary to check the logic of the project to ensure no dividing by zero operation, the value will turn to FALSE after downloading the project.
ETH1Err	BOOL	Fault information of first Ethernet, this value is TRUE when the network cable is disconnected (FALSE, by default)
ETH2Err	BOOL	Fault information of second Ethernet, this value is TRUE when the network cable is disconnected (FALSE, by default)
InterComErr	BYTE	Fault information of internal communication link, the corresponding bit is 1 when the different channels fails (0, by default) bit0 = 1 : Internal communication link in slot 2 failure bit1 = 1 : Internal communication link in slot 3 failure bit2 = 1 : Internal communication link in slot 4 failure (6 slot backboard r reserved) bit3 = 1 : Internal communication link in slot 5 failure (6 slot backboard r reserved) bit4 = 1 :Slot mismatch bit5~bit7 : reserved

3.3.7.5.3 Application Example of Instructions

Variable Definition

No.	Variable Name	Address	Variable Description	Variable Type	Online Value	Power Fail Safeguard
0001	In_var			BOOL	TRUE	FALSE
0002	Frame_var			BOOL	FALSE	FALSE
0003	Q_var			BOOL	TRUE	FALSE
0004	Div_var			BOOL	FALSE	FALSE
0005	ETH1E_var			BOOL	TRUE	FALSE
0006	ETH2E_var			BOOL	FALSE	FALSE
0007	ComE_var			BYTE	0	FALSE
0008	CPUDiagInfo			SYSGETCPUDIAGINFO		FALSE

Programing language	Program
Ladder diagram (LD)	

<table>
<tr><th colspan="2">Variable Definition</th></tr>
<tr><td>Structured text (ST)</td><td>
</td></tr>
<tr><td>Continuous Function Chart (CFC)</td><td>
</td></tr>
</table>

Program description:

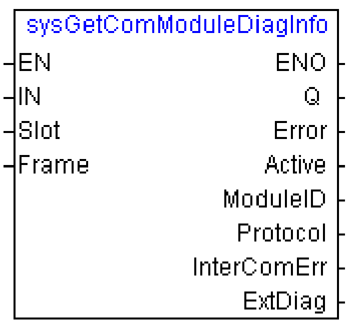
Example for getting the diagnosis information of CPU module in master frame, this moment, the first Ethernet is broken.

3.3.7.6 SysGetComModuleDiagInfo (Get Diagnosis of Communication Module)

3.3.7.6.1 Function

Get the diagnosis information of the communication module.

Note: This function block can get the diagnosis information of controller, LK249 and LK240 module. You can double click the LK246/LK241 module tree node to view diagnostic information in the [Diagnostic Information] tab. The [diagnostic information] tab is displayed after the project is compiled.



sysGetComModuleDiagInfo Instruction

3.3.7.6.2 Parameter Description

Input Parameter	Type	Parameter Description
IN	BOOL	High level enable (FALSE, by default)
Slot	BYTE	Slot number (2, by default, valid value from 2-5) The slot number of the master control module is fixed to 1. The slot number of other modules is ascended to the right according to the location related to the master control module. Input the slot number of the version information module to be obtained during the use.
Frame	BOOL	Select frame to be get, FALSE is current frame (master), TRUE is opposite frame (slave)

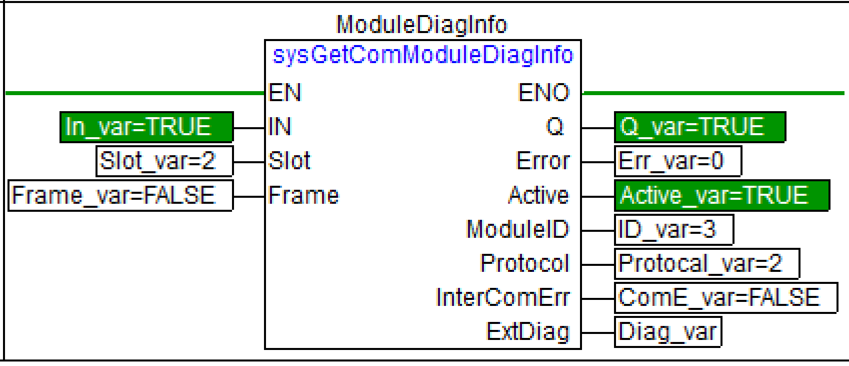
Output Parameter	Type	Parameter Description
Q	BOOL	Output (FALSE, by default)
Error	BYTE	Error (0, by default) 0: Correct 1: Slot configuration is error, the actual configuration address does not match the installation slot 128: There is no module on the slot of local frame 129: There is no module on the slot of opposite frame 130 The slot module does not support this function
Active	BOOL	Working state of communication module (0, by default) FALSE: Offline TRUE: Online
ModuleID	BYTE	Module ID number 1: LK220, LK222, LK224 (reserved) 2: LK240 3: LK249 4: LK247 5: LK248 6: LK245
Protocol	BYTE	Communication protocol ID number 1: Internal redundancy protocol 2: DP protocol 3: CANOpen protocol 4: EtherCAT protocol 5: PowerLink protocol 6: ModbusRTU protocol 7: Free port protocol 8: ModbusTCP protocol
InterComErr	BOOL	Fault information of internal communication link, this value is TRUE when the fault occurs (0, by default).
ExtDiag	ARRAY[0…31] OF BYTE	Extended diagnosis information LK240: The first byte represents the first optical fiber is broken (the corresponding value is 1). The second byte represents the second optical fiber is broken (the corresponding value is 1). The third byte represents the A/B switch conflict occurs (the corresponding value is 1). The fourth byte represents clear the flag of watchdog reset result (the corresponding value is 1). Other bytes are reserved currently. The fifth byte is reserved. The sixth byte represents the bit error rate of the first optical fiber (0~100). The seventh byte represents the bit error rate of the second optical fiber

Output Parameter	Type	Parameter Description
		(0~100). Other bytes are reserved currently.
		LK249: The first byte represents the first 485 is broken (the corresponding value is 1). The second byte represents the second 485 is broken (the corresponding value is 1). The third byte are reserved currently. The fourth byte represents clear the flag for watchdog reset result (the corresponding value is 1). Other bytes are reserved currently.
		LK245: The first byte represents the first Ethernet is broken (the corresponding value is 1). The second byte represents the second Ethernet is broken (the corresponding value is 1). Other bytes are reserved currently.

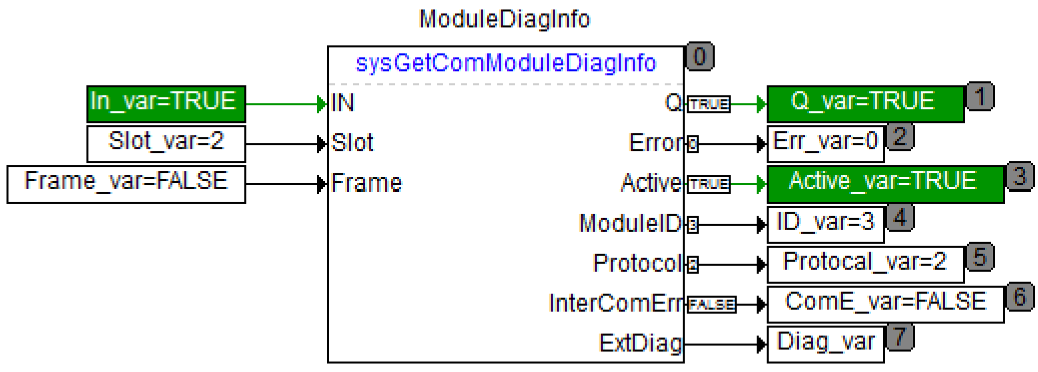
3.3.7.6.3 Application Example of Instructions

Variable Definition

No.	Variable Name	Address	Variable Description	Variable Type	Online Value	Power Fail Safeguard
0001	In_var			BOOL	TRUE	FALSE
0002	Slot_var			BYTE	2	FALSE
0003	Frame_var			BOOL	FALSE	FALSE
0004	Q_var			BOOL	TRUE	FALSE
0005	Err_var			BYTE	0	FALSE
0006	Active_var			BOOL	TRUE	FALSE
0007	ID_var			BYTE	3	FALSE
0008	Protocal_var			BYTE	2	FALSE
0009	ComE_var			BOOL	FALSE	FALSE
0010	Diag_var			ARRAY[0..31] OF BYTE		FALSE
0011	ModuleDiagInfo			SYSGETCOMMODULEDIAGINFO		FALSE

Programing language	Program
Ladder diagram (LD)	

<table>
<tr><th colspan="2">Variable Definition</th></tr>
<tr><td>Structured text (ST)</td><td>
ModuleDiagInfo(
IN:=In_var,
Slot:=Slot_var,
Frame:=Frame_var ,
Q=>Q_var,
Error=>Err_var,
Active=>Active_var,
ModuleID=>ID_var,
Protocol=>Protocal_var,
InterComErr=>ComE_var,
ExtDiag=>Diag_var);

ModuleDiagInfo
In_var = TRUE
Slot_var = 2
Frame_var = FALSE
Q_var = TRUE
Err_var = 0
Active_var = TRUE
ID_var = 3
Protocal_var = 2
ComE_var = FALSE
Diag_var
</td></tr>
<tr><td>Continuous Function Chart (CFC)</td><td>


</td></tr>
</table>

Program description:

Example for getting the diagnosis information of LK249 module in master frame.

Double-click the array variable Diag_var to view the diagnosis information, as shown in Figure. The online value of first byte is 1 ,and it indicates the first DP link is default.

Comdiag1.Diag_var

No.	Variable Name	Address	Variable Description	Variable Type	Online Value	Power Fail Safeguard
0001	Diag_var[0]			BYTE	1	FALSE
0002	Diag_var[1]			BYTE	0	FALSE
0003	Diag_var[2]			BYTE	0	FALSE
0004	Diag_var[3]			BYTE	0	FALSE
0005	Diag_var[4]			BYTE	0	FALSE
0006	Diag_var[5]			BYTE	0	FALSE
0007	Diag_var[6]			BYTE	0	FALSE
0008	Diag_var[7]			BYTE	0	FALSE
0009	Diag_var[8]			BYTE	0	FALSE
0010	Diag_var[9]			BYTE	0	FALSE
0011	Diag_var[10]			BYTE	0	FALSE
0012	Diag_var[11]			BYTE	0	FALSE
0013	Diag_var[12]			BYTE	0	FALSE
0014	Diag_var[13]			BYTE	0	FALSE
0015	Diag_var[14]			BYTE	0	FALSE
0016	Diag_var[15]			BYTE	0	FALSE

View the Diagnosis Information

3.3.8 Memory Operation Instruction

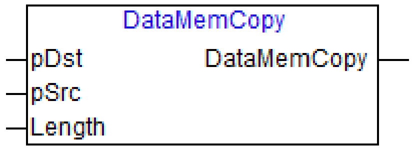
3.3.8.1 DataMemCopy(Copy Data from Source Address to Destination Address)

3.3.8.1.1 Function

In bytes, copy data from source address to destination address. the data from any area can be selected as a source data, and S area data cannot be seleted as a destination data.

When using the instruction, the memory address of the source data and the destination data should be obtained first. When a condition is satisfied, copy a length of data from the source memory address to the destination memory address.

Note that the data length must be less than or equal to the size of the source data and destination data indicated by the pointer.



DataMemCopy Function

3.3.8.1.2 Parameter Description

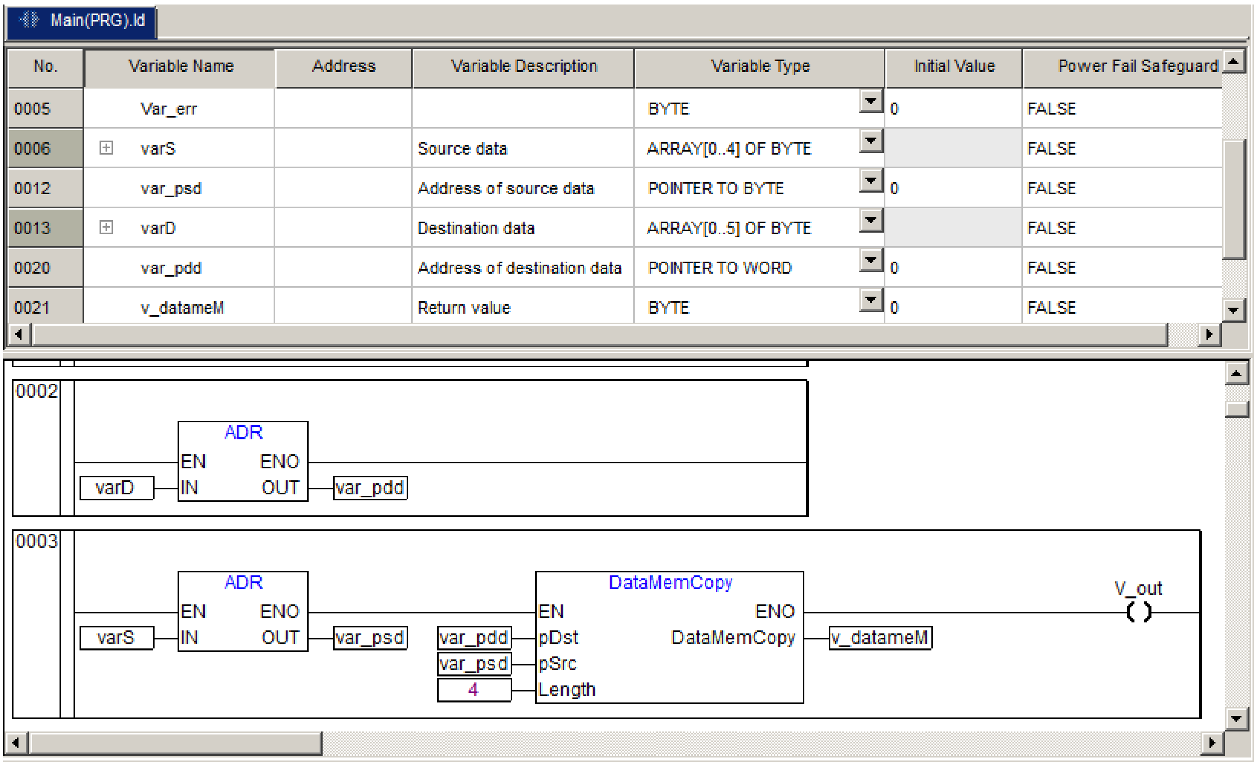
Input Parameter	Data Type	Function Description	Parameter Description	Default Value
pDst	POINTER TO BYTE	Source data address	Valid address of data area	0
pSrc	POINTER TO BYTE	Destination data address	Valid address of data area	0
Length	DWORD	Data length	Unit is 1 byte	0

Output Parameter	Data Type	Function Description	Parameter Description
DataMemCopy	BYTE	Error message	0: execution succeeded 1: source data address is error 2: destination data address is error

3.3.8.1.3 Application Example of Instructions

Copy the four bytes data from the array varS to the array varD.

Set the pointer variables var_psd and var_pdd to obtain the memory addresses of source data and destination data respectively. Set the data length to 4 bytes. The configuration example is shown in the figure.

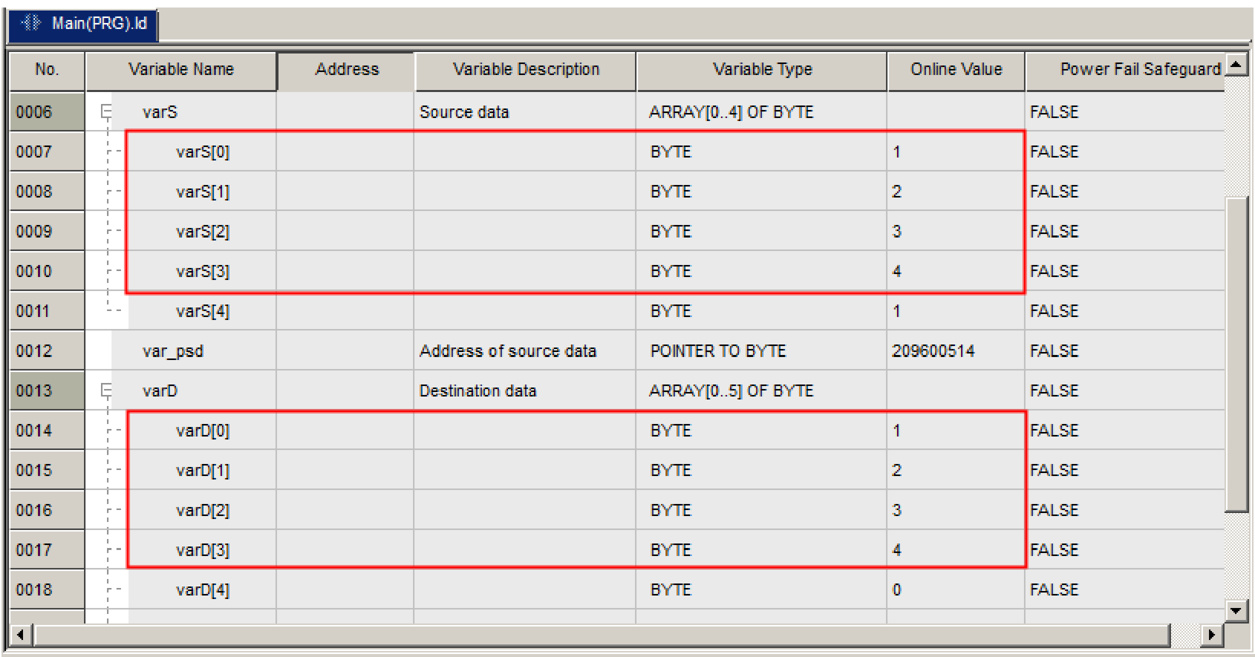


Main(PRG).ld

No.	Variable Name	Address	Variable Description	Variable Type	Initial Value	Power Fail Safeguard
0005	Var_err			BYTE	0	FALSE
0006	varS		Source data	ARRAY[0..4] OF BYTE		FALSE
0012	var_psd		Address of source data	POINTER TO BYTE	0	FALSE
0013	varD		Destination data	ARRAY[0..5] OF BYTE		FALSE
0020	var_pdd		Address of destination data	POINTER TO WORD	0	FALSE
0021	v_datameM		Return value	BYTE	0	FALSE

DataMemCopy Configuration

Run the program. The array elements varS [0] ~ varS [3] are copied to the array elements varD [0] ~ varD [3], as shown in the following figure.



Main(PRG).ld

No.	Variable Name	Address	Variable Description	Variable Type	Online Value	Power Fail Safeguard
0006	varS		Source data	ARRAY[0..4] OF BYTE		FALSE
0007	varS[0]			BYTE	1	FALSE
0008	varS[1]			BYTE	2	FALSE
0009	varS[2]			BYTE	3	FALSE
0010	varS[3]			BYTE	4	FALSE
0011	varS[4]			BYTE	1	FALSE
0012	var_psd		Address of source data	POINTER TO BYTE	209600514	FALSE
0013	varD		Destination data	ARRAY[0..5] OF BYTE		FALSE
0014	varD[0]			BYTE	1	FALSE
0015	varD[1]			BYTE	2	FALSE
0016	varD[2]			BYTE	3	FALSE
0017	varD[3]			BYTE	4	FALSE
0018	varD[4]			BYTE	0	FALSE

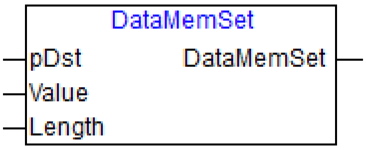
Runing Result

3.3.8.2 DataMemSet(Set Data to Destination Address)

3.3.8.2.1 Function

Write a data value to the specified address.S area data cannot be seleted as a destination data.

Note that the data length must be less than or equal to the size of the destination data indicated by the pointer.



DataMemSet Function

3.3.8.2.2 Parameter Description

Input Parameter	Data Type	Function Description	Parameter Description	Default Value
pDst	POINTER TO BYTE	Destination data address	Valid address of data area	0
Value	BYTE	Data value	Set data value	0
Length	DWORD	Data length	Unit is 1 byte	0

Output Parameter	Data Type	Function Description	Parameter Description
DataMemSet	BYTE	Error message	0: execution succeeded 1: destination data address is error

3.3.8.2.3 Application Example of Instructions

Before copying the memory data, please reset the destination address data to the set value of 6.

Set the pointer variables var_psd and var_pdd to obtain the memory addresses of source data and destination data respectively.

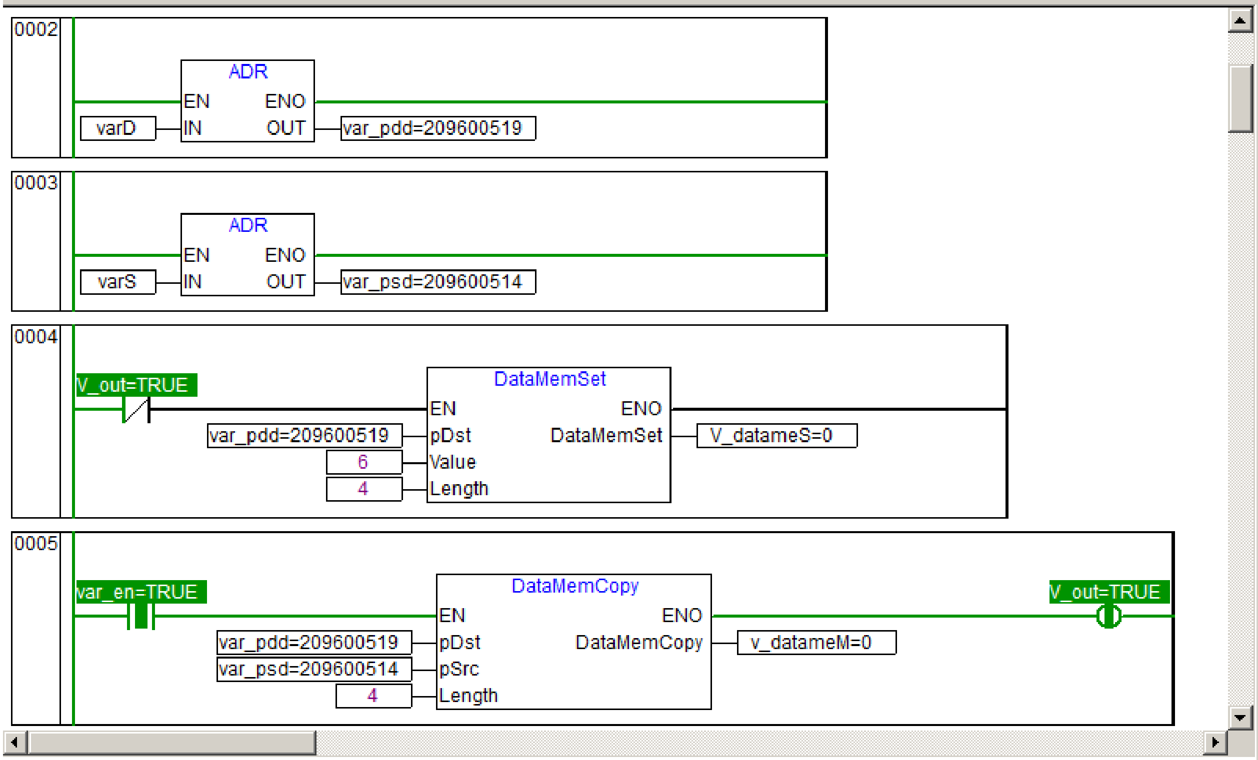
The ENO of the DataMemCopy instruction is reversed as the enable signal of the DataMemSet instruction. Before the memory copying, first reset the destination data to the set value.

When the enable signal of DataMemCopy is TRUE, the instruction is executed, and the source data is copied to the destination address

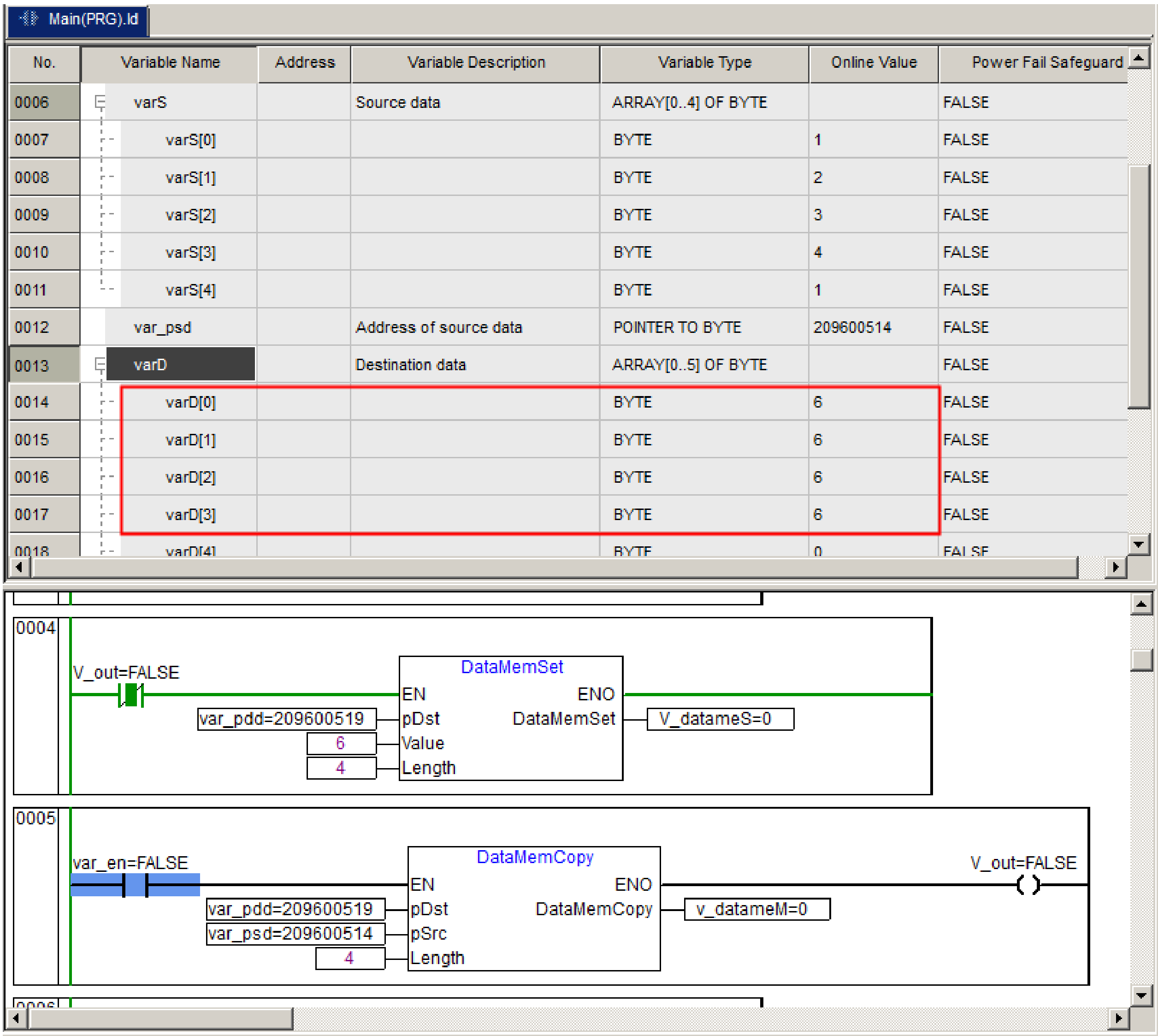
.

Main(PRG).ld

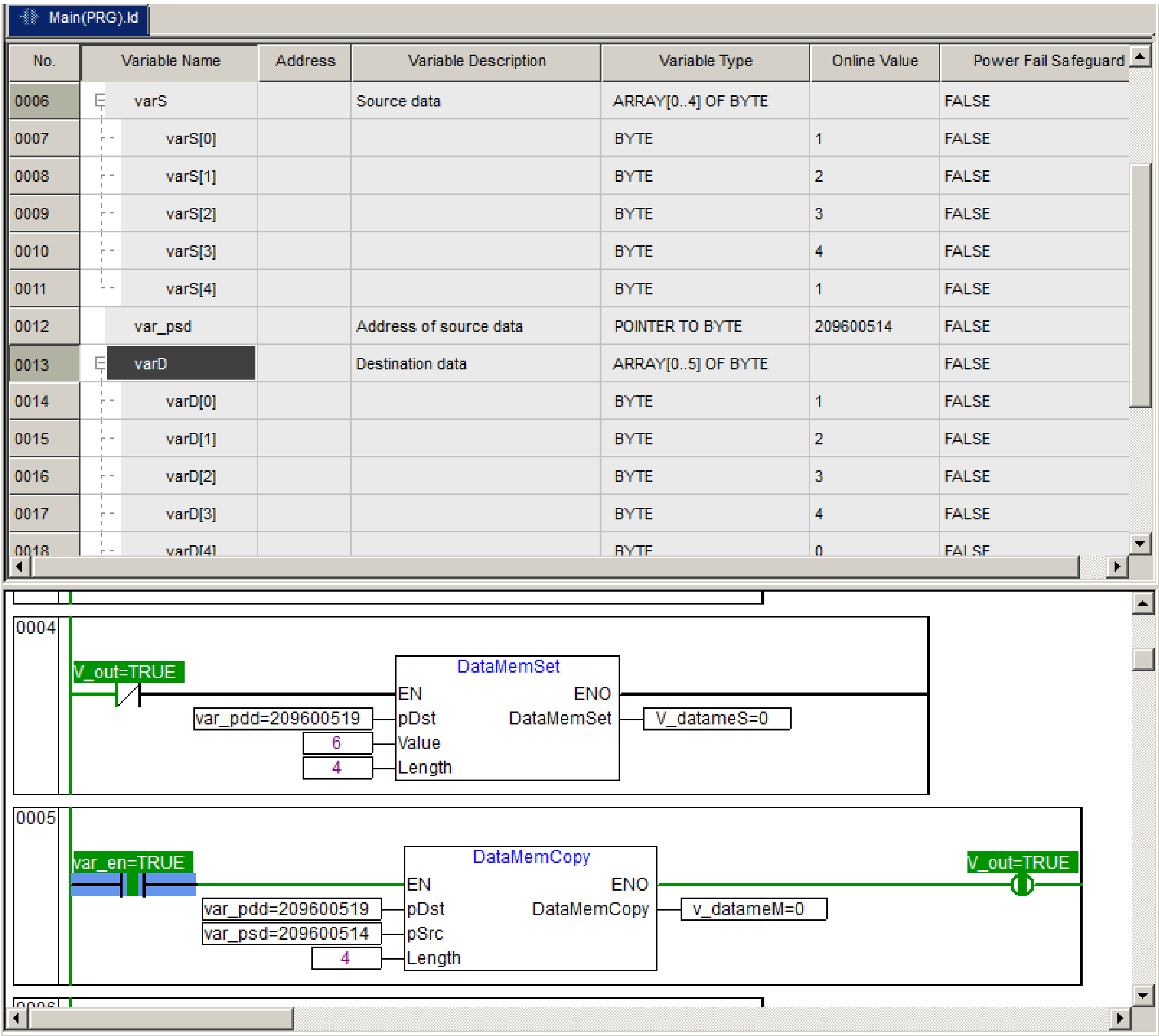
No.	Variable Name	Address	Variable Description	Variable Type	Online Value	Power Fail Safeguard
0006	varS		Source data	ARRAY[0..4] OF BYTE		FALSE
0012	var_psd		Address of source data	POINTER TO BYTE	209600514	FALSE
0013	varD		Destination data	ARRAY[0..5] OF BYTE		FALSE
0020	var_pdd		Address of destination data	POINTER TO WORD	209600519	FALSE
0021	v_datameM		Veturn value	BYTE	0	FALSE
0022	V_datameS			BYTE	0	FALSE
0023	V_out		Enable signal of DataMemSet	BOOL	TRUE	FALSE
0024	var_en		Enable signal of DataMemCopy	BOOL	TRUE	FALSE



DataMemSet Configuration



Write Value to Destination Address


Main(PRG).ld

No.	Variable Name	Address	Variable Description	Variable Type	Online Value	Power Fail Safeguard
0006	varS		Source data	ARRAY[0..4] OF BYTE		FALSE
0007	varS[0]			BYTE	1	FALSE
0008	varS[1]			BYTE	2	FALSE
0009	varS[2]			BYTE	3	FALSE
0010	varS[3]			BYTE	4	FALSE
0011	varS[4]			BYTE	1	FALSE
0012	var_psd		Address of source data	POINTER TO BYTE	209600514	FALSE
0013	varD		Destination data	ARRAY[0..5] OF BYTE		FALSE
0014	varD[0]			BYTE	1	FALSE
0015	varD[1]			BYTE	2	FALSE
0016	varD[2]			BYTE	3	FALSE
0017	varD[3]			BYTE	4	FALSE
0018	varD[4]			BYTE	0	FALSE

Copy Data to Destination Address

3.3.9 Interrupt Setting Functions

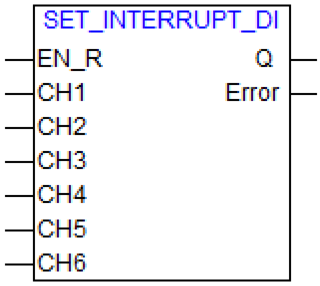
3.3.9.1 SET_INTERRUPT_DI(Interruption Setting)

3.3.9.1.1 Module Support

14 points (LE5104, LE5105) support CH1,CH2. 24 points (LE5106, LE5106M, LE5107) and 40 points (LE5108, LE5109, LE5118) support total 6 channels.

3.3.9.1.2 Function

Set quick external interruption parameter instruction which can set external interruption for one or multiple channels, but it is only available for 6 channels; default is 0, that is external interruption forbidden.



SET_INTERRUPT_DI Function Bock

3.3.9.1.3 Description of Parameter

Input Parameter	Type of Data	Description of Function	Description of Parameter Value
EN_R	BOOL	Enable	0→1 Rising edge enable
CH1	BYTE	Set external interruption parameter of first channel	Channel 1: 0 interruption forbidden; 1 rising edge triggered interruption; 2 falling edge triggered interruption; 3 falling edge and rising edge triggered interruption
CH2	BYTE	Set external interruption parameter of second channel	Channel 2: 0 interruption forbidden; 1 rising edge triggered interruption; 2 falling edge triggered interruption; 3 falling edge and rising edge triggered interruption
CH3	BYTE	Set external interruption parameter of third channel	Channel 3: 0 interruption forbidden; 1 rising edge triggered interruption; 2 falling edge triggered interruption; 3 falling edge and rising edge triggered interruption
CH4	BYTE	Set external interruption parameter of fourth channel	Channel 4: 0 interruption forbidden; 1 rising edge triggered interruption; 2 falling edge triggered interruption; 3 falling edge and rising edge triggered interruption
CH5	BYTE	Set external interruption parameter of fifth channel	Channel 5:0 interruption forbidden; 1 rising edge triggered interruption; 2 falling edge triggered interruption; 3 falling edge and rising edge triggered interruption
CH6	BYTE	Set external interruption parameter of sixth channel	Channel 6: 0 interruption forbidden; 1 rising edge triggered interruption; 2 falling edge triggered interruption; 3 falling edge and rising edge triggered interruption

Output Parameter	Type of Data	Description of Function	Value
Q	BOOL	Mark of finishing	0: Set parameter failure 1: Set parameter successfully
Error	BYTE	Error information	0: No error information 1: Configuration parameter is error 2: Call many function blocks 8: Resources are occupied (for example: DI channels have been configured as pulse catch or high speed count)

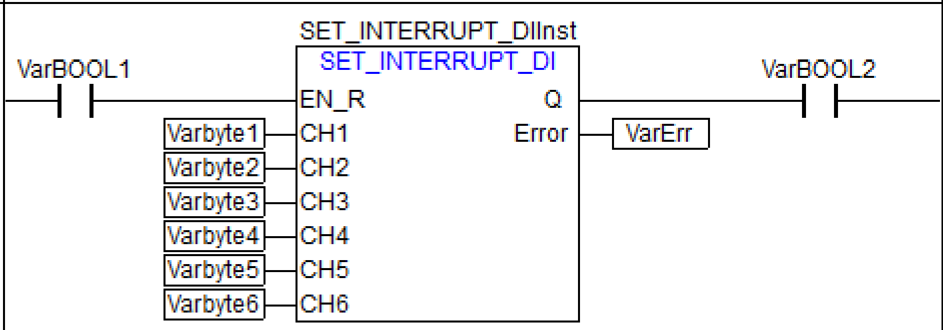
ⓘ

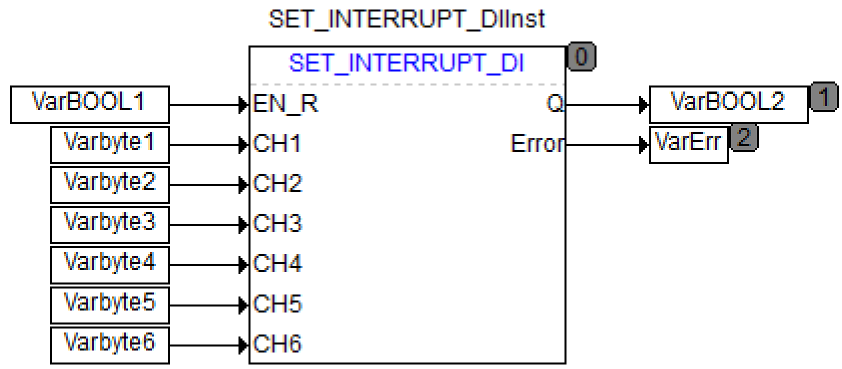
- Set external interruption function end CH1 in first channel: parameter value range is 0~3; when setting CH1=0, function block forbids external interruption; when setting CH1=1, after enabling function block, input pulse has the change of rising edge, then an external interruption is triggered; when setting CH1=2, after enabling function block, input pulse has the change of falling edge, then an external interruption is triggered; when setting CH1=3, after enabling function block, input pulse has the change of rising edge or falling edge, then an external interruption is triggered. Other channels settings and so on.
- 14 points, the quick external interruption channel is arranged in I0.0~I0.1; 24 points, the quick external interruption channel is arranged in I0.0~I0.5; 40 points, the front four channels of quick external interruption channel is arranged in I0.0~I0.3, and last two channels are arranged inI1.4~1.5.

3.3.9.1.4 Application Example of Instructions

Variable Definition

No.	Variable Name	Address	Variable Description	Variable Type	Initial Value	Power Fail Safeguard
0001	SET_INTERRUPT_DIInst			SET_INTERRUPT_DI		FALSE
0002	Varbyte1			BYTE	0	FALSE
0003	Varbyte2			BYTE	0	FALSE
0004	Varbyte3			BYTE	0	FALSE
0005	Varbyte4			BYTE	0	FALSE
0006	Varbyte5			BYTE	0	FALSE
0007	Varbyte6			BYTE	0	FALSE
0008	VarErr			BYTE	0	FALSE
0009	VarBOOL1			BOOL	FALSE	FALSE
0010	VarBOOL2			BOOL	FALSE	FALSE

Programing language	Program
Ladder diagram (LD)	
Structured text (ST)	SET_INTERRUPT_DIInst(EN_R:=VarBOOL1, CH1:=Varbyte1, CH2:=Varbyte2, CH3:=Varbyte3, CH4:=Varbyte4 CH5:=Varbyte5, CH6:=Varbyte6,

Variable Definition	
	Q=>VarBOOL2, Error=> VarErr);
Continuous function diagram (CFC)	

Program Description:

When EN_R is set and held, Q is equal to TRUE, and the corresponding input channel triggers the corresponding fast external interrupt when each time it reaches a rising edge.

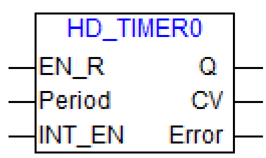
When EN_R is reset, the corresponding input channel stops receiving the interrupt pulse and Q is equal to FALSE.

3.3.9.2 HD_TIMER0 (Interruption Timer Blocks0)

3.3.9.2.1 Module Supporting

14 points (LE5104, LE5105) support front four interruption timers. 24 points (LE5106, LE5106M, LE5107) and 40 points (LE5108, LE5109, LE5118, LE5119) support total 6 channels.

3.3.9.2.2 Function



HD_TIMER0 Function Block

3.3.9.2.3 Description of Parameter

Input Parameter	Type of Data	Description of Function	Description of Parameter Value
EN_R	BOOL	Enable	0: Invalid 1: Rising edge enable
Period	DWORD	Set time（us）	500 us~800,000 us can be set

Input Parameter	Type of Data	Description of Function	Description of Parameter Value
INT_EN	BOOL	Interruption enable	0: When set time is up, there is no interruption 1: When set time is up, there is interruption

Output Parameter	Type of Data	Description of Function	Description of Parameter Value
Q	BOOL	Whether finishing set or not	0: No setting 1: Finish setting and preserving
CV	DWORD	Passed time（us）	0~800,000 us
Error	BYTE	Error information	=0: No error information =1: Parameter setting is error =2: Hardware timer has no any operation =3: Call multiple same function blocks

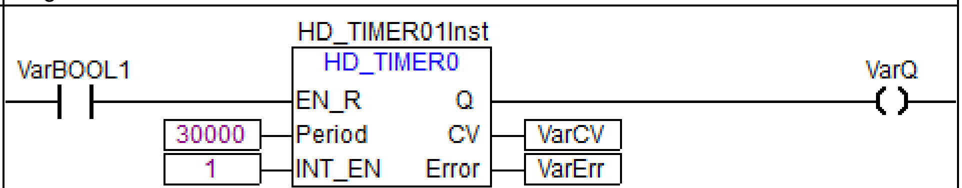
ⓘThere are 6 hardware timers, name is HD_TIMER0~5, each hardware timer corresponding to 1 channel timer interruption.

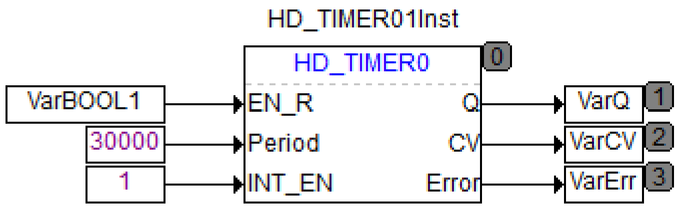
3.3.9.2.4 Application Example of Instructions

HD_TIMER1~HD_TIMER5 are similar to HD_TIMER0.

Variable Definition

No.	Variable Name	Address	Variable Description	Variable Type	Initial Value	Power Fail Safeguard
0001	HD_TIMER01Inst			HD_TIMER0		FALSE
0002	VarBOOL1			BOOL	FALSE	FALSE
0003	VarCV			DWORD	0	FALSE
0004	VarErr			BYTE	0	FALSE
0005	VarQ			BOOL	FALSE	FALSE

Programing language	Program
Ladder diagram (LD)	
Structured text (ST)	HD_TIMER0Inst(EN_R:=VarBOOL1, Period:=30000, INT_EN:=1, Q=>VarQ, CV=>VarCV, Error=> VarErr);

Variable Definition	
Continuous function diagram (CFC)	

Program Description:

EN_R is set and held (rising edge), Q is equal to 1, since INT_EN is equal to 1, then the HD_TIMER0 interrupt event is triggered every 30000us, and the HD_TIMER0 interrupt event is called by the system event in AutoThink. CV displays the current elapsed time value, the unit is us.

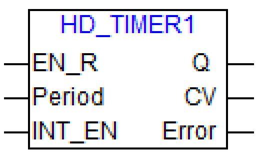
When EN is reset, the HD_TIMER0 interrupt event is stopped and Q is equal to 0.

3.3.9.3 HD_TIMER1(Interruption Timer Blocks1)

3.3.9.3.1 Module Supporting

14 points (LE5104, LE5105) support front four interruption timers. 24 points (LE5106, LE5106M, LE5107) and 40 points (LE5108, LE5109, LE5118, LE5119) support total 6 channels.

3.3.9.3.2 Function



HD_TIMER1 Function Bock

3.3.9.3.3 Description of Parameter

Input Parameter	Type of Data	Description of Function	Description of Parameter Value
EN_R	BOOL	Enable	0: Invalid 1: Rising edge enable
Period	DWORD	Set time（us）	500 us~800,000 us can be set
INT_EN	BOOL	Interruption enable	0: When set time is up, there is no interruption 1: When set time is up, there is interruption

Output Parameter	Type of Data	Description of Function	Description of Parameter Value
Q	BOOL	Whether finishing set or not	0: No setting 1: Finish setting and preserving
CV	DWORD	Passed time（us）	0~800,000 us

Output Parameter	Type of Data	Description of Function	Description of Parameter Value
Error	BYTE	Error information	=0: No error information =1: Parameter setting is error =2: Hardware timer has no any operation =3: Call multiple same function blocks

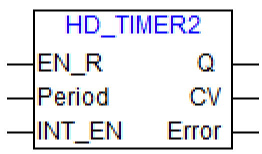
ⓘThere are 6 hardware timers, name is HD_TIMER0~5, each hardware timer corresponding to 1 channel timer interruption.

3.3.9.4 HD_TIMER2 (Interruption Timer Blocks2)

3.3.9.4.1 Module Supporting

14 points (LE5104, LE5105) support front four interruption timers. 24 points (LE5106, LE5106M, LE5107) and 40 points (LE5108, LE5109, LE5118, LE5119) support total 6 channels.

3.3.9.4.2 Function



HD_TIMER2 Function Block

3.3.9.4.3 Description of Parameter

Input Parameter	Type of Data	Description of Function	Description of Parameter Value
EN_R	BOOL	Enable	0: Invalid 1: Rising edge enable
Period	DWORD	Set time（us）	500 us~800,000 us can be set
INT_EN	BOOL	Interruption enable	0: When set time is up, there is no interruption 1: When set time is up, there is interruption

Output Parameter	Type of Data	Description of Function	Description of Parameter Value
Q	BOOL	Whether finishing set or not	0: No setting 1: Finish setting and preserving
CV	DWORD	Passed time（us）	0~800,000 us
Error	BYTE	Error information	=0: No error information =1: Parameter setting is error =2: Hardware timer has no any operation =3: Call multiple same function blocks

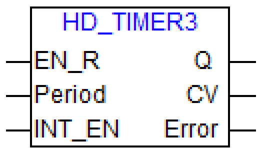
ⓘThere are 6 hardware timers, name is HD_TIMER0~5, each hardware timer corresponding to 1 channel timer interruption.

3.3.9.5 HD_TIMER3 (Interruption Timer Blocks3)

3.3.9.5.1 Module Supporting

14 points (LE5104, LE5105) support front four interruption timers. 24 points (LE5106, LE5106M, LE5107) and 40 points (LE5108, LE5109, LE5118, LE5119) support total 6 channels.

3.3.9.5.2 Function



HD_TIMER3 Function Block

3.3.9.5.3 Description of Parameter

Input Parameter	Type of Data	Description of Function	Description of Parameter Value
EN_R	BOOL	Enable	0: Invalid 1: Rising edge enable
Period	DWORD	Set time（us）	500 us~800,000 us can be set
INT_EN	BOOL	Interruption enable	0: When set time is up, there is no interruption 1: When set time is up, there is interruption

Output Parameter	Type of Data	Description of Function	Description of Parameter Value
Q	BOOL	Whether finishing set or not	0: No setting 1: Finish setting and preserving
CV	DWORD	Passed time（us）	0~800,000 us
Error	BYTE	Error information	=0: No error information =1: Parameter setting is error =2: Hardware timer has no any operation =3: Call multiple same function blocks

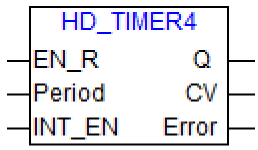
ⓘThere are 6 hardware timers, name is HD_TIMER0~5, each hardware timer corresponding to 1 channel timer interruption.

3.3.9.6 HD_TIMER4 (Interruption Timer Blocks4)

3.3.9.6.1 Module Supporting

14 points (LE5104, LE5105) support front four interruption timers. 24 points (LE5106, LE5106M, LE5107) and 40 points (LE5108, LE5109, LE5118, LE5119) support total 6 channels.

3.3.9.6.2 Function



HD_TIMER4 Function Block

3.3.9.6.3 Description of Parameter

Input Parameter	Type of Data	Description of Function	Description of Parameter Value
EN_R	BOOL	Enable	0: Invalid 1: Rising edge enable
Period	DWORD	Set time（us）	500 us~800,000 us can be set
INT_EN	BOOL	Interruption enable	0: When set time is up, there is no interruption 1: When set time is up, there is interruption

Output Parameter	Type of Data	Description of Function	Description of Parameter Value
Q	BOOL	Whether finishing set or not	0: No setting 1: Finish setting and preserving
CV	DWORD	Passed time（us）	0~800,000 us
Error	BYTE	Error information	=0: No error information =1: Parameter setting is error =2: Hardware timer has no any operation =3: Call multiple same function blocks

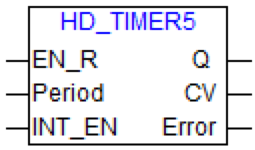
ⓘThere are 6 hardware timers, name is HD_TIMER0~5, each hardware timer corresponding to 1 channel timer interruption.

3.3.9.7 HD_TIMER5 (Interruption Timer Blocks5)

3.3.9.7.1 Module Supporting

14 points (LE5104, LE5105) support front four interruption timers. 24 points (LE5106, LE5106M, LE5107) and 40 points (LE5108, LE5109, LE5118, LE5119) support total 6 channels.

3.3.9.7.2 Function



HD_TIMER5 Function Block

3.3.9.7.3 Description of Parameter

Input Parameter	Type of Data	Description of Function	Description of Parameter Value
EN_R	BOOL	Enable	0: Invalid 1: Rising edge enable
Period	DWORD	Set time（us）	500 us~800,000 us can be set
INT_EN	BOOL	Interruption enable	0: When set time is up, there is no interruption 1: When set time is up, there is interruption

Output Parameter	Type of Data	Description of Function	Description of Parameter Value
Q	BOOL	Whether finishing set or not	0: No setting 1: Finish setting and preserving
CV	DWORD	Passed time（us）	0~800,000 us
Error	BYTE	Error information	=0: No error information =1: Parameter setting is error =2: Hardware timer has no any operation =3: Call multiple same function blocks

ⓘThere are 6 hardware timers, name is HD_TIMER0~5, each hardware timer corresponding to 1 channel timer interruption.

3.4 Product Extension Instruction

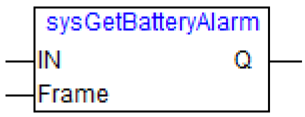
3.4.1 Product Internal State Instruction

3.4.1.1 SysGetBatteryAlarm (Get Battery Alarm)

3.4.1.1.1 Function

Get the battery power information of the master control module.

When the standby controller is in slave status ,the function block outputs valid value.



sysGetBatteryAlarm Instruction

3.4.1.1.2 Parameter Description

Input Parameter	Type	Parameter Description
IN	BOOL	High level enable (FALSE, by default)
Frame	BOOL	Select frame (FALSE, by default) TRUE:Slave frame FALSE:Master frame

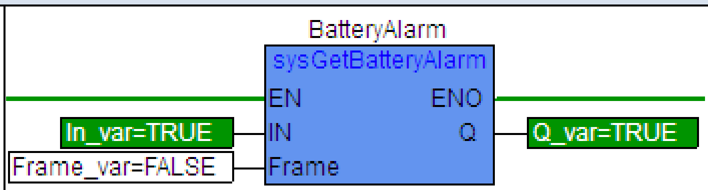
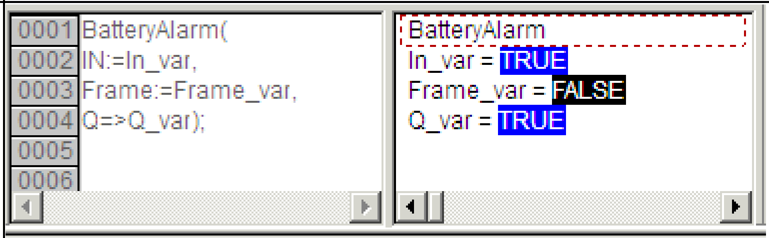
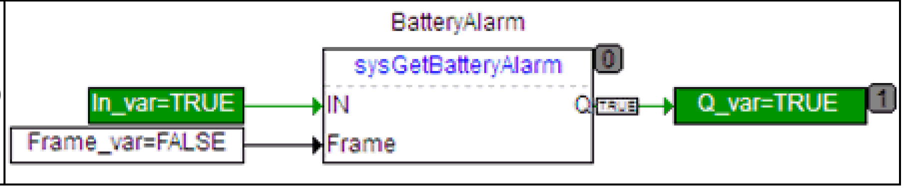
Output Parameter	Type	Parameter Description
Q	BOOL	Battery power TURE: No .batteryor low battery FALSE: The battery power is normal.

3.4.1.1.3 Application Example of Instructions

Variable Definition

No.	Variable Name	Address	Variable Description	Variable Type	Online Value	Power Fail Safeguard
0001	In_var			BOOL	TRUE	FALSE
0002	Frame_var			BOOL	FALSE	FALSE
0003	Q_var			BOOL	TRUE	FALSE
0004	BatteryAlarm			SYSGETBATTERYALARM		FALSE

Programing language	Program

Variable Definition	
Ladder diagram (LD)	
Structured text (ST)	
Continuous Function Chart (CFC)	

Program description:

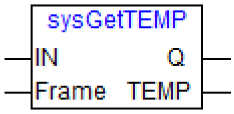
Low battery in the master frame trigger the Q as TRUE, then you need to replace the battery in CPU module.

3.4.1.2 SysGetTEMP (Get Temperature of CPU)

3.4.1.2.1 Function

Get the temperature information of the master control module.

When the standby controller is in slave status ,the function block outputs valid value.



sysGetTEMP Instruction

3.4.1.2.2 Parameter Description

Input Parameter	Type	Parameter Description
IN	BOOL	High level enable (FALSE, by default)
Frame	BOOL	Select frame (FALSE, by default) TRUE:Slave frame

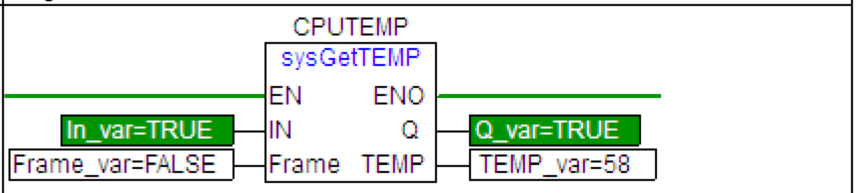
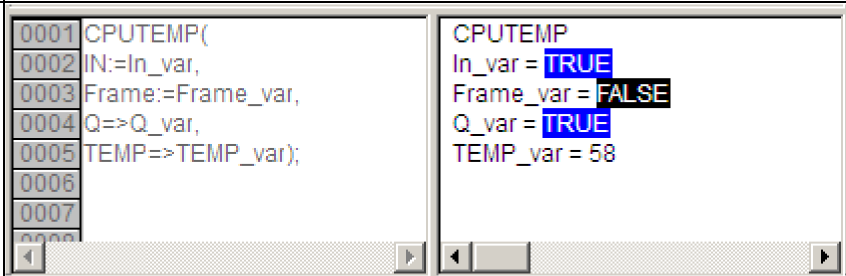
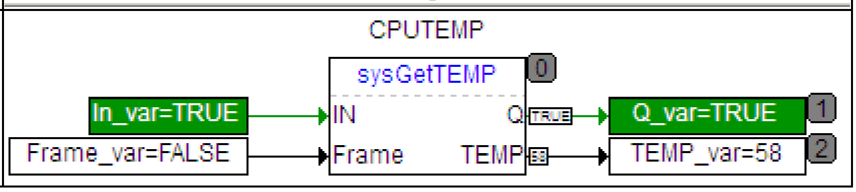
Input Parameter	Type	Parameter Description
		FALSE:Master frame

Output Parameter	Type	Parameter Description
Q	BOOL	Output (FALSE, by default)
TEMP	SINT	Temperatuer value with the unit celsius (0, by default)

3.4.1.2.3 Application Example of Instructions

Variable Definition

No.	Variable Name	Address	Variable Description	Variable Type	Online Value	Power Fail Safeguard
0001	In_var			BOOL	TRUE	FALSE
0002	Frame_var			BOOL	FALSE	FALSE
0003	Q_var			BOOL	TRUE	FALSE
0004	TEMP_var			SINT	58	FALSE
0005	CPUTEMP			SYSGETTEMP		FALSE

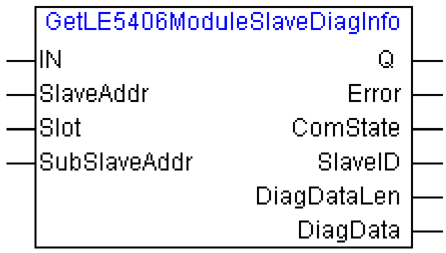
Programing language	Program
Ladder diagram (LD)	
Structured text (ST)	
Continuous Function Chart (CFC)	

3.4.2 Additional Module Information

3.4.2.1 GetLE5406ModuleSlaveDiagInfo(Get LE5406 Slave Module Diagnose Information)

3.4.2.1.1 Function

Obtain the diagnosis information of LE IO slave station under the LE5406 gateway module.



GetLE5406ModuleSlaveDiagInfo

3.4.2.1.2 Parameter Description

Input Parameter	Type	Parameter Description
IN	BOOL	High level enable (FALSE, by default)
SlaveAddr	BYTE	LE5406 address, valid value from 2~125
Slot	BYTE	Slot of DP master module
SubSlaveAddr	BYTE	LE IO slave address

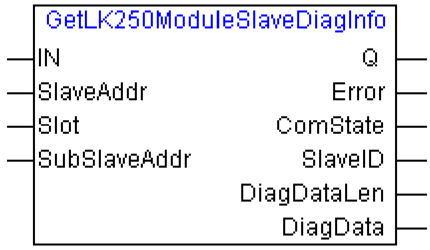
Output Parameter	Type	Parameter Description
Q	BOOL	Output enable
Error	BYTE	Error: 0: Correct 1: Slot number is error 2: Slave address of gateway module is error 3: Slave address of IO module is error 4: Gateway module is offline 128:No DP master station or slot error 255: Configured module does not match the actual model
ComState	BOOL	LE IO slave online status: TRUE: Module is online FALSE: Module is offline
SlaveID	LEIOTYPE	ID of LE IO slave station
DiagDataLen	BYTE	Diagnosis information length of slave station
DiagData	ARRAY OF BYTE	LE IO slave diagnosis information, see LE Special register area (S)

ⓘWhen the module is online (The output parameters Q, Error, ComState are normal), LE IO slave station diagnosis information is valid.

3.4.2.2 GetLK250ModuleSlaveDiagInfo(Get LK250 Slave Module Diagnose Information)

3.4.2.2.1 Function

Obtain the diagnosis information of IO slave station under the LK250 gateway module. AutoThink V3.1.8 and later versions recommend using GetLK250ModuleSlaveDiagInfoExt(Get LK250 Slave Module Diagnose Information) function block to obtain diagnostic information.



GetLK250ModuleSlaveDiagInfo

3.4.2.2.2 Parameter Description

Input Parameter	Type	Parameter Description
IN	BOOL	High level enable (FALSE, by default)
SlaveAddr	BYTE	LK250 address, valid value from 2~125
Slot	BYTE	Slot of LK249 module
SubSlaveAddr	BYTE	Slave station address of IO module connected by LK250

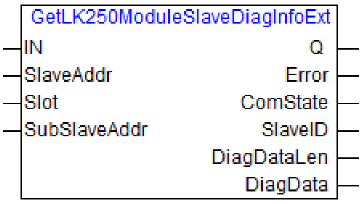
Output Parameter	Type	Parameter Description
Q	BOOL	Output enable
Error	BYTE	Error: 0: Correct 1: Slot number is error 2: Slave address of gateway module is error 3: Slave address of IO module is error 4: Gateway module is offline 128: No DP master station or slot error 255: Configured module does not match the actual model
ComState	BOOL	Online status of DPslave station: TRUE: Module is online FALSE: Module is offline
SlaveID	DPMODULETYPE	ID of slave station connected by LK250
DiagDataLen	BYTE	Diagnosis information length of slave station
DiagData	ARRAY OF BYTE	DP slave station diagnosis information, see diagnosis of each module

ⓘWhen the module is online (The output parameters Q, Error, ComState are normal), IO slave station diagnosis information is valid.

3.4.2.3 GetLK250ModuleSlaveDiagInfoExt(Get LK250 Slave Module Diagnose Information)

3.4.2.3.1 Function

Obtain the diagnosis information of IO slave station under the LK250 gateway module. AutoThink V3.1.8 and later versions recommend using this function block to obtain diagnostic information.



GetLK250ModuleSlaveDiagInfoExt

3.4.2.3.2 Parameter Description

Input Parameter	Type	Parameter Description
IN	BOOL	High level enable (FALSE, by default)
SlaveAddr	BYTE	LK250 address, valid value from 2~125
Slot	BYTE	Slot of LK249 module
SubSlaveAddr	BYTE	Slave station address of IO module connected by LK250

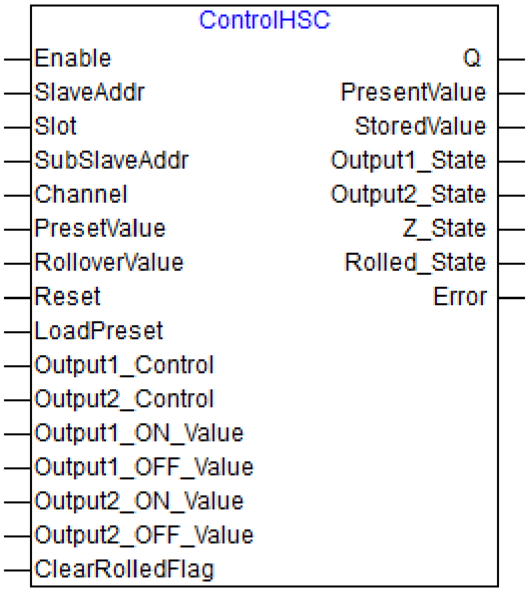
Output Parameter	Type	Parameter Description
Q	BOOL	Output enable
Error	BYTE	Error: 0: Correct 1: Slot number is error 2: Slave address of gateway module is error 3: Slave address of IO module is error 4: Gateway module is offline 128: No DP master station or slot error 255: Configured module does not match the actual model
ComState	BOOL	Online status of DPslave station: TRUE: Module is online FALSE: Module is offline
SlaveID	DPMODULETYPE_N	ID of slave station connected by LK250
DiagDataLen	BYTE	Diagnosis information length of slave station
DiagData	ARRAY OF BYTE	DP slave station diagnosis information, see diagnosis of each module

ⓘWhen the module is online (The output parameters Q, Error, ComState are normal), IO slave station diagnosis information is valid.

3.4.2.4 ControlHSC(Control and Read LK620 Module Information)

3.4.2.4.1 Function

Set the parameters and read the operation status of LK620 module.



ControlHSC

3.4.2.4.2 Parameter Description

Input Parameter	Type	Function Description	Parameter Description
Enable	BOOL	Enable intput	0: invalid 1: valid
SlaveAddr	BYTE	Station address	Slave address of LK620, 2~125, default value 2
Slot	BYTE	Slot number	Slot number of communication module (LK249/LK241) , 2~5, default value 2
SubSlaveAddr	BYTE	Slave station address	LK235/LK271 device address, 1~239, default value 1 SubSlaveAddr is set, when communication module is LK241. SubSlaveAddr remains default value 1, when communication module is LK249
Channel	BYTE	Channel	=1:channel1(counter1) =2:channel2(counter2)
PresetValue	DWORD	Preset value	Counts from preset value, which must be less than the rollover value. Otherwise, there will be a number error. The range of preset values is 0-4, 294, 967, 295.
RolloverValue	DWORD	Rollover value	In the count mode, upper limit range is 1-4, 294, 967, 295. In the counting process, when the counting value is equal to rollover value-1, the counter returns to zero and starts counting again.
Reset	BOOL	Reset counter	0: invalid 0→ 1: Rising edge, counter reset and count from 0 1: Two output channels output OFF. Set output not according to the configuration

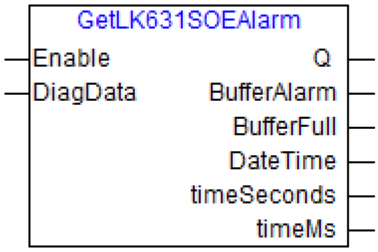
Input Parameter	Type	Function Description	Parameter Description
LoadPreset	BOOL	Control single of preset value	0: invalid 1: Rising edge is valid. The counter loads the preset value and counts from the preset value.
Output1_Control	BYTE	Output control signal	Used to control the output state of the counter, modify current output value of output 1 : = 0x00, output value according to counting result = 0x02, force OUT1 value to OFF = 0x03, force OUT1 value to ON
Output2_Control	BYTE	Output control signal	Used to control the output state of the counter, modify current output value of output 2 : = 0x00, output value according to counting result = 0x02, force OUT1 value to OFF = 0x03, force OUT1 value to ON
Output1_ON_Value	DWORD	Output ON signal	Output ON trigger value (0-4,294,967,295)
Output1_OFF_Value	DWORD	Output OFF signal	Output OFF trigger value (0-4,294,967,295)
Output2_ON_Value	DWORD	Output ON signal	Output ON trigger value (0-4,294,967,295)
Output2_OFF_Value	DWORD	Output OFF signal	Output OFF trigger value (0-4,294,967,295)
ClearRolledFlag	BOOL	Clear signal of rollover identification	Clear the rollover identification

Output Parameter	Type	Function Description	Parameter Description
Q	BOOL	Output flag	0: Output not completed 1: Output completed
PresentValue	DWORD	Current count value	Store the current count value of the selected counter
StoredValue	DWORD	Store count value	Current count value of the selected counter
Output1_State	BOOL	Output 1 state	1:ON (channel is ON), the current count reaches Output1_ON_Value 0:OFF (channel is OFF, stop output), the current count reaches Output1_OFF_Value
Output2_State	BOOL	Output 2 state	1:ON (channel is ON), the current count reaches Output2_ON_Value 0:OFF (channel is OFF, stop output), the current count reaches Output2_OFF_Value
Z_State	BOOL	Z state	0: Low level 1: High level
Rolled_State	BOOL	Rollover identification	Whether the counter reaches the reversed value and reverse 0: not reversed 1: Reversed
Error	DWORD	Error information	= 0, no errors bit1 = 1, Slot is not in range 2 ~ 5 bit2= 1, Module ID is not in range 3 ~ 4 bit3 = 1, LK235/LK271 station address is not in range 1 ~ 239 (LK241) bit4 = 1, Input IO slave not in range 2 ~ 125 (LK249/ LK235/LK271) bit5 = 1. This station is not configured to LK620 module. bit6 = 1, Channel value error bit7 = 1, Preset value error bit8 = 1, Output control error bit9 = 1, Output trigger value error

3.4.2.5 GetLK631SOEAlarm(Read LK631 FRAM Alarm Info Module Information)

3.4.2.5.1 Function

Get SOE events alarm information of LK631 module, including alarm information of 70% SOE events, 100% SOE events nd event recording time information.



GetLK631SOEAlarm

3.4.2.5.2 Parameter Description

Input Parameter	Type	Parameter Description
Enable	BOOL	Enable intput 0: invalid 1: valid
DiagData	ARRAY[0..237] OF BYTE	Diagnosis data. When the module reports diagnosis data, a SOE event is recorded When the module is configured in the Profibus-DP, this parameter is associated with the output pin "DiagData" of the sysGetDPSlaveStateExt instruction When the module is configured in the POWERLINK, you need to calculate sum of the first address of the diagnostic variable plus 6, and assign the sum to "diagdata"

Output Parameter	Type	Parameter Description
Q	BOOL	Flag of Output state 0: Output not completed 1: Output completed
BufferAlarm	BOOL	70% alarm flag When SOE events of LK631 module exceeds 70% maximum storage, set alarm flag to 1. The maximum storage of SOE is 3072
BufferFull	BOOL	100% alarm flag When SOE events of LK631 module exceeds 100% maximum storage, set alarm flag to 1. The maximum storage of SOE is 3072
DataTime	DT	Recording time of SOE alarm event
timeSeconds	DWORD	The second time corresponding to the event recording time of minutes and above
timeMs	WORD	The millisecond time corresponding to the event recording time less than minutes

3.4.2.6 COMM_SET_CONFIG(Set LK238 Communication Instruction Param)

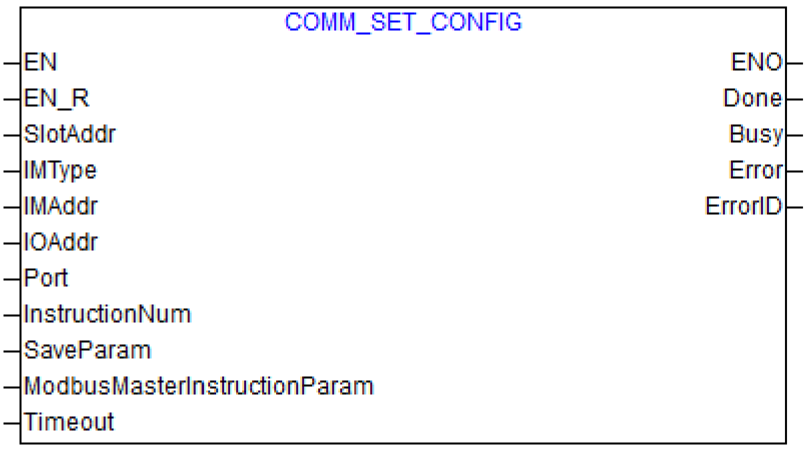
3.4.2.6.1 Function description

Set the communication instruction parameters of the LK238 module, and set up to 16 instructions at a time. The instruction has a corresponding example project, which can be directly imported for use.

When importing algorithm examples, you can import them in the following two ways:

- In the menu bar, click [Project] - [Import Sample Project]. For details, see import sample project.
- In the library manager, right-click the function block and select the [Import Sample Project] command. For details, see LK238 serial port communication module.

The imported example project needs to be called in the program or under the task to be used normally.



COMM_SET_CONFIG

3.4.2.6.2 Parameter description

Input Parameter	Type	Parameter Description
EN_R	BOOL	Rising edge execution
SlotAddr	BYTE	Slot number, the effective value is 2~5
IMType	BYTE	Type of interface module used in the backplane where the IO module is located: 1: The interface module type is LK231 or LK232. At this time, the IMAddr parameter is meaningless 2: The interface module type is LK235/LK271 3: The interface module type is LK250 4: The interface module type is LK234/LK235M. At this time, the SlotAddr parameter is meaningless
IMAddr	DWORD	The interface module address used by the backplane where the IO module is located When the interface module is LK231 or LK232,

Input Parameter	Type	Parameter Description
		the IMAddr value is meaningless and does not need to be set When the interface module is LK235/LK271, IMAddr is consistent with the setting value of the module address dial switch When the interface module is LK250, IMAddr is consistent with the setting value of the expansion backplane dial switch When the interface module is LK234/LK235M, IMAddr is the hexadecimal value of IP address of LK234. For example, the IP address is 192.168.1.250, and the IMAddr parameter is 16#C0A801FA If the IMAddr configuration is wrong, the ErrorID will report 7
IOAddr	BYTE	DP slave address of the backplane where the IO module is located (2 ~ 125)
Port	BYTE	Port number (1~4)
InstructionNum	BYTE	Number of configuration instructions (1~16)
SaveParam	BOOL	Whether the issued command parameter is maintained after power failure
ModbusMasterInstructionParam	ARRAY[0..15] OF MASTERINSTRUCTIONPARAM	Modbus master command parameter structure When there is a duplicate instruction ID in the Modbus master instruction parameter structure, the instruction parameter of the last instruction ID in the structure member is valid.
Timeout	WORD	Execution timeout time (unit: 1ms)(0 ~ 65535), when it is set to 0, the default value of 2000 is adopted; this parameter indicates how long an error is reported after the communication between the controller and the serial port module is abnormal, which has nothing to do with whether the actual communication times out

Output Parameter	Type	Parameter Description
Done	BOOL	Whether the command has been executed
Busy	BOOL	Whether the command is being executed
Error	BYTE	An error is occurred when setting communication parameters
ErrorID	WORD	Error code =0: Correct =1: IMType parameter error =2: SlotAddr specified slot range error =3: SlotAddr the communication board type of the specified slot is not supported =4: IO module DP slave address specified by IOAddr is incorrect =5: The interface module address specified by IMAddr is incorrect =7: The specified address module type of IOAddr is wrong (this station is not configured with LK238 module, please check whether the IMType, SlotAddr, IMAddr and IOAddr parameters are configured correctly) =8: I / O byte length error (please check the I / O module on the module configuration page) =9: Port parameter configuration error =10: Instructions number configured by InstructionNum is error =11: The configuration instruction parameter length exceeds the output byte length (please check

Output Parameter	Type	Parameter Description
		the output module on the module configuration page) =12: ModbusMasterInstructionParam parameter configuration is error =13: The user parameter of this port is not configured as a Modbus master =15: Abnormal communication between controller and serial port module

MasterInstructionParam Structure member

Member variables	Type	Parameter Description
InstructionID	BYTE	Instruction ID number (1~32)
InstructionStatus	BYTE	Command status (0: invalid 1: valid)
StationAddress	BYTE	Slave address (1~247)
FunctionCode	BYTE	0x01 (read coil), 0x02 (read discrete input), 0x03 (read hold register), 0x04 (read input register), 0x05 (write single coil), 0x06 (write single register), 0x0f (write multiple coils), 0x10 (write multiple registers)
OffsetAddress	WORD	Read and write offset address
DataLength	WORD	Data length for reading and writing switch/analog Switch:1~1600 Analog:1~100
Timeout	WORD	Execution timeout (unit: 1ms), When it is set to 0, the default value is 1000
TriggerMode	BYTE	Trigger mode 0: periodic trigger , when the parameter configuration is successful and the protocol port is enabled, MODBUS starts communication according to the polling cycle set by period 1: event trigger, when the parameter configuration is successful and the protocol port is enabled, through rising edge of EN_R of COMM_RW_MODBUS_MASTER (LK238 Modbus Master Read and Write Communication Data) triggers Modbus communication once
Period	BYTE	Polling cycle time (unit: 10ms), when set to 0, the default polling mode is dead pocket type

3.4.2.7 COMM_GET_CONFIG(Read LK238 Communication Instruction Param)

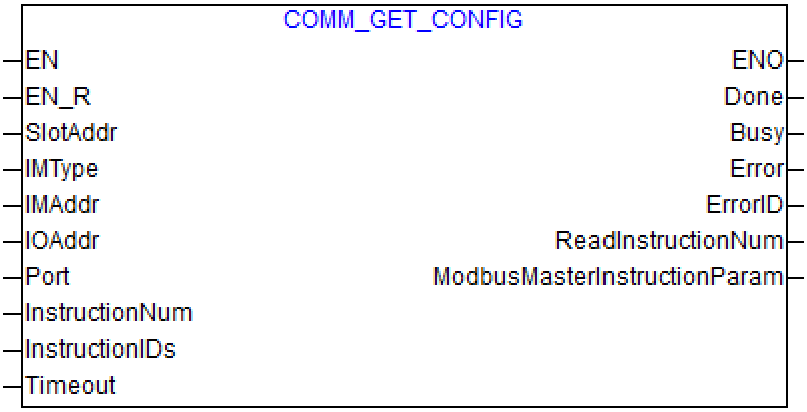
3.4.2.7.1 Function description

Get the communication command parameters of the LK238 module, and get up to 16 commands at a time.The instruction has a corresponding example project, which can be directly imported for use.

When importing algorithm examples, you can import them in the following two ways:

- In the menu bar, click [Project] - [Import Sample Project]. For details, see import sample project.
- In the library manager, right-click the function block and select the [Import Sample Project] command. For details, see LK238 serial port communication module.

The imported example project needs to be called in the program or under the task to be used normally.



COMM_GET_CONFIG

3.4.2.7.2 Parameter description

Input Parameter	Type	Parameter Description
EN_R	BOOL	Rising edge execution
SlotAddr	BYTE	Slot number, the effective value is 2~5
IMType	BYTE	Type of interface module used in the backplane where the IO module is located: 1: The interface module type is LK231 or LK232. At this time, the IMAddr parameter is meaningless 2: The interface module type is LK235/LK271 3: The interface module type is LK250 4: The interface module type is LK234/LK235M. At this time, the SlotAddr parameter is meaningless
IMAddr	DWORD	The interface module address used by the backplane where the IO module is located When the interface module is LK231 or LK232, the IMAddr value is meaningless and does not need to be set When the interface module is LK235/LK271, IMAddr is consistent with the setting value of the module address dial switch When the interface module is LK250, IMAddr is consistent with the setting value of the expansion backplane dial switch When the interface module is LK234/LK235M, IMAddr is the hexadecimal value of IP address of LK234. For example, the IP address is 192.168.1.250, and the IMAddr parameter is 16#C0A801FA If the IMAddr configuration is wrong, the ErrorID will report 7
IOAddr	BYTE	DP slave address of the backplane where the IO module is located (2 ~ 125)
Port	BYTE	Port number (1~4)
InstructionNum	BYTE	Number of configuration instructions (1~16)
InstructionIDs	ARRAY[0..15] OF BYTE	Instruction ID number (1~32)
Timeout	WORD	Execution timeout time (unit: 1ms))(0 ~ 65535), when it is set to 0, the default value of 2000 is adopted; this parameter indicates how long to report error after the communication between the controller and the serial port module is abnormal, which has nothing to do with whether the actual communication times out

Output Parameter	Type	Parameter Description
Done	BOOL	Reading communication parameters is

Output Parameter	Type	Parameter Description
		completed
Busy	BOOL	Reading parameter configuration in progress
Error	BOOL	An error occurred while reading communication parameters
ErrorID	WORD	Error code =0: Correct =1: IMType parameter error =2: SlotAddr specified slot range error =3: SlotAddr the communication board type of the specified slot is not supported =4: IO module DP slave address specified by IOAddr is incorrect =5: The interface module address specified by IMAddr is incorrect =7: The specified address module type of IOAddr is wrong (this station is not configured with LK238 module, please check whether the IMType, SlotAddr, IMAddr and IOAddr parameters are configured correctly) =8: I / O byte length error (please check the I / O module on the module configuration page) =9: Port parameter configuration error =10: Instructions number configured by InstructionNum is error =11: The configuration instruction parameter length exceeds the input byte length (please check the input module on the module configuration page) =12: InstructionIDs parameter configuration is error =13: The user parameter of this port is not configured as a Modbus master =15: Abnormal communication between controller and serial port module
ReadInstructionNum	BYTE	Read the total number of configuration
ModbusMasterInstructionParam	ARRAY[0..15] OF MASTERINSTRUCTIONPARAM	Modbus master command parameter structure

MasterInstructionParam Structure member

Member variables	Type	Parameter Description
InstructionID	BYTE	Instruction ID number (1~32)
InstructionStatus	BYTE	Command status (0: invalid 1: valid)
StationAddress	BYTE	Slave address (1~247)
FunctionCode	BYTE	0x01 (read coil), 0x02 (read discrete input), 0x03 (read hold register), 0x04 (read input register), 0x05 (write single coil), 0x06 (write single register), 0x0f (write multiple coils), 0x10 (write multiple registers)
OffsetAddress	WORD	Read and write offset address
DataLength	WORD	Data length for reading and writing switch/analog Switch:1~1600 Analog:1~100
Timeout	WORD	Execution timeout (unit: 1ms)
TriggerMode	BYTE	Trigger mode (0: cycle trigger, 1: event trigger)
Period	BYTE	Polling cycle time (unit: 10ms)

3.4.2.8 COMM_RW_MODBUS_MASTER(LK238 Modbus Master Read and Write Communication Data)

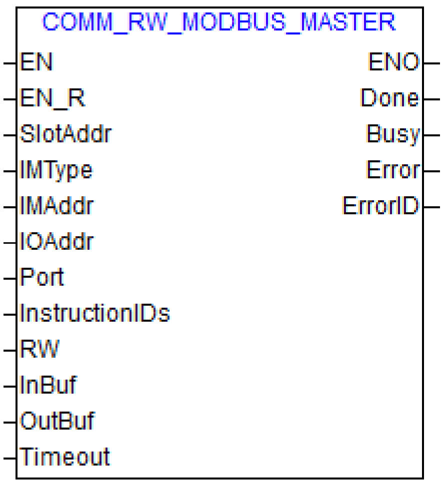
3.4.2.8.1 Function description

Read or write LK238 Modbus master station data.The instruction has a corresponding example project, which can be directly imported for use.

When importing algorithm examples, you can import them in the following two ways:

- In the menu bar, click [Project] - [Import Sample Project]. For details, see import sample project.
- In the library manager, right-click the function block and select the [Import Sample Project] command. For details, see LK238 serial port communication module.

The imported example project needs to be called in the program or under the task to be used normally.



COMM_RW_MODBUS_MASTER

3.4.2.8.2 Parameter description

Input Parameter	Type	Parameter Description
EN_R	BOOL	Rising edge execution
SlotAddr	BYTE	Slot number, the effective value is 2~5
IMType	BYTE	Type of interface module used in the backplane where the IO module is located: 1: The interface module type is LK231 or LK232. At this time, the IMAddr parameter is meaningless 2: The interface module type is LK235/LK271 3: The interface module type is LK250 4: The interface module type is LK234/LK235M. At this time, the SlotAddr parameter is meaningless
IMAddr	DWORD	The interface module address used by the backplane where the IO module is located When the interface module is LK231 or LK232, the IMAddr value is meaningless and does not need to be set

Input Parameter	Type	Parameter Description
		When the interface module is LK235/LK271, IMAddr is consistent with the setting value of the module address dial switch When the interface module is LK250, IMAddr is consistent with the setting value of the expansion backplane dial switch When the interface module is LK234/LK235M, IMAddr is the hexadecimal value of IP address of LK234. For example, the IP address is 192.168.1.250, and the IMAddr parameter is 16#C0A801FA If the IMAddr configuration is wrong, the ErrorID will report 7
IOAddr	BYTE	DP slave address of the backplane where the IO module is located (2 ~ 125)
Port	BYTE	Port number (1~4)
InstructionIDs	BYTE	Instruction ID number (1~32)
RW	BOOL	Read and write direction (read data when RW is 0, write data when 1)
InBuf	POINTER TO BYTE	Input data buffer, the size of the buffer pointed to must be ≥ 224 bytes
OutBuf	POINTER TO BYTE	Output data buffer, the size of the buffer pointed to must be ≥ 224 bytes
Timeout	WORD	Execution timeout time (unit: 1ms))(0 ~ 65535), when it is set to 0, the default value of 2000 is adopted; this parameter indicates how long to report error after the communication between the controller and the serial port module is abnormal, which has nothing to do with whether the actual communication times out

Output Parameter	Type	Parameter Description
Done	BOOL	Reading and writing master station data completed
Busy	BOOL	Reading and writing master data
Error	BOOL	An error occurred while reading and writing master data
ErrorID	WORD	Function block error code =0: Correct =1: IMType parameter error =2: SlotAddr specified slot range error =3: SlotAddr the communication board type of the specified slot is not supported =4: IO module DP slave address specified by IOAddr is incorrect =5: The interface module address specified by IMAddr is incorrect =7: The specified address module type of IOAddr is wrong (this station is not configured with LK238 module, please check whether the IMType, SlotAddr, IMAddr and IOAddr parameters are configured correctly) =8: I / O byte length error (please check the I / O module on the module configuration page) =9: Port parameter configuration error =10: InstructionIDs parameter configuration error =15: Abnormal communication between controller and serial port module Master station error message =1001: Data frame error =1002: Timeout =1004: The current instruction is invalid =1005: The port is not working (When Com Status in the user parameter page is disable) =1006: Port protocol mismatch =1128+1: Function code not supported by the slave =1128+2: Data address overflow =1128+3: Data range overflow =1128+4: Slave station equipment failure =1128+6: The slave device is busy

3.4.2.9 COMM_RW_MODBUS_SLAVE(LK238 Modbus Slave Read and Write Communication Data)

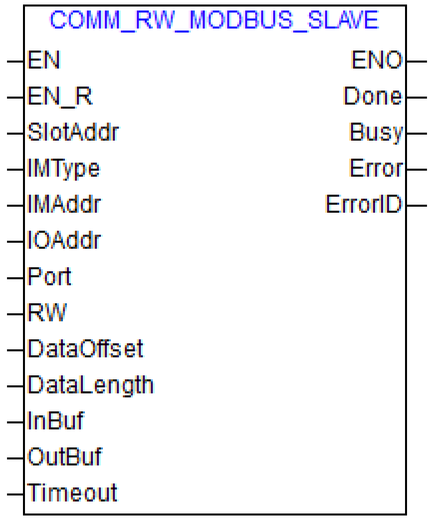
3.4.2.9.1 Function description

Read or write LK238 Modbus slave data.The instruction has a corresponding example project, which can be directly imported for use.

When importing algorithm examples, you can import them in the following two ways:

- In the menu bar, click [Project] - [Import Sample Project]. For details, see import sample project.
- In the library manager, right-click the function block and select the [Import Sample Project] command. For details, see LK238 serial port communication module.

The imported example project needs to be called in the program or under the task to be used normally.



COMM_RW_MODBUS_SLAVE

3.4.2.9.2 Parameter description

Input Parameter	Type	Parameter Description
EN_R	BOOL	Rising edge execution
SlotAddr	BYTE	Slot number, the effective value is 2~5
IMType	BYTE	Type of interface module used in the backplane where the IO module is located: 1: The interface module type is LK231 or LK232. At this time, the IMAddr parameter is meaningless 2: The interface module type is LK235/LK271 3: The interface module type is LK250 4: The interface module type is LK234/LK235M. At this time, the SlotAddr parameter is meaningless
IMAddr	DWORD	The interface module address used by the backplane where the IO module is located When the interface module is LK231 or LK232, the IMAddr value is meaningless and

Input Parameter	Type	Parameter Description
		does not need to be set When the interface module is LK235/LK271, IMAddr is consistent with the setting value of the module address dial switch When the interface module is LK250, IMAddr is consistent with the setting value of the expansion backplane dial switch When the interface module is LK234/LK235M, IMAddr is the hexadecimal value of IP address of LK234. For example, the IP address is 192.168.1.250, and the IMAddr parameter is 16#C0A801FA If the IMAddr configuration is wrong, the ErrorID will report 7
IOAddr	BYTE	DP slave address of the backplane where the IO module is located (2 ~ 125)
Port	BYTE	Port number (1~4)
RW	BOOL	Read and write direction (read data when RW is 0, write data when 1)
DataOffset	WORD	Read / write data offset address
DataLength	BYTE	Read data length
InBuf	POINTER TO BYTE	Input data buffer, the size of the buffer pointed to must be ≥ 224 bytes
OutBuf	POINTER TO BYTE	Output data buffer, the size of the buffer pointed to must be ≥ 224 bytes
Timeout	WORD	Execution timeout time (unit: 1ms))(0 ~ 65535), when it is set to 0, the default value of 2000 is adopted; this parameter indicates how long to report error after the communication between the controller and the serial port module is abnormal, which has nothing to do with whether the actual communication times out

Output Parameter	Type	Parameter Description
Done	BOOL	Reading and writing slave station data completed
Busy	BOOL	Reading and writing slave data
Error	BOOL	An error occurred while reading and writing slave data
ErrorID	WORD	Function block error code =0: Correct =1: IMType parameter error =2: SlotAddr specified slot range error =3: SlotAddr the communication board type of the specified slot is not supported =4: IO module DP slave address specified by IOAddr is incorrect =5: The interface module address specified by IMAddr is incorrect =7: The specified address module type of IOAddr is wrong (this station is not configured with LK238 module, please check whether the IMType, SlotAddr, IMAddr and IOAddr parameters are configured correctly) =8: I / O byte length error (please check the I / O module on the module configuration page) =9: Port parameter configuration error =10: The read data length exceeds the input byte length (please check the input module on the module configuration page) =11: The write data length exceeds the output byte length (please check the output module on the module configuration page) =12: The set value of read / write data length is too small (0) =13: The length of read and write data exceeds 220 bytes =14: The offset plus length is greater than 60 * 1024 =15: Abnormal communication between controller and serial port module Slave station error message =1001: CRC check error =1005: The port is not working (When Com Status in the user parameter page is disable) =1006: Port protocol mismatch =1128+2: Data address overflow =1128+3: Data range overflow

3.4.2.10 COMM_RW_FREEPORT(LK238 Free Protocol Read and Write Communication Data)

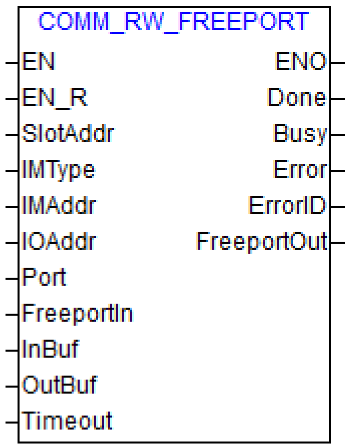
3.4.2.10.1 Function description

Read or write LK238 free port data.The instruction has a corresponding example project, which can be directly imported for use.

When importing algorithm examples, you can import them in the following two ways:

- In the menu bar, click [Project] - [Import Sample Project]. For details, see import sample project.
- In the library manager, right-click the function block and select the [Import Sample Project] command. For details, see LK238 serial port communication module.

The imported example project needs to be called in the program or under the task to be used normally.



COMM_RW_FREEPORT

3.4.2.10.2 Parameter description

Input Parameter	Type	Parameter Description
EN_R	BOOL	Rising edge execution
SlotAddr	BYTE	Slot number, the effective value is 2~5
IMType	BYTE	Type of interface module used in the backplane where the IO module is located: 1: The interface module type is LK231 or LK232. At this time, the IMAddr parameter is meaningless 2: The interface module type is LK235/LK271 3: The interface module type is LK250 4: The interface module type is LK234/LK235M. At this time, the SlotAddr parameter is meaningless
IMAddr	DWORD	The interface module address used by the backplane where the IO module is located

Input Parameter	Type	Parameter Description
		When the interface module is LK231 or LK232, the IMAddr value is meaningless and does not need to be set When the interface module is LK235/LK271, IMAddr is consistent with the setting value of the module address dial switch When the interface module is LK250, IMAddr is consistent with the setting value of the expansion backplane dial switch When the interface module is LK234/LK235M, IMAddr is the hexadecimal value of IP address of LK234. For example, the IP address is 192.168.1.250, and the IMAddr parameter is 16#C0A801FA If the IMAddr configuration is wrong, the ErrorID will report 7
IOAddr	BYTE	DP slave address of the backplane where the IO module is located (2 ~ 125)
Port	BYTE	Port number (1~4)
FreeportIn	FREEPORTINPARAM	Free port protocol input data
InBuf	POINTER TO BYTE	Input data buffer, the size of the buffer pointed to must be ≥ 224 bytes
OutBuf	POINTER TO BYTE	Output data buffer, the size of the buffer pointed to must be ≥ 224 bytes
Timeout	WORD	Execution timeout time (unit: 1ms))(0 ~ 65535), when it is set to 0, the default value of 2000 is adopted; this parameter indicates how long to report error after the communication between the controller and the serial port module is abnormal, which has nothing to do with whether the actual communication times out

Output Parameter	Type	Parameter Description
Done	BOOL	Reading and writing free port data completed
Busy	BOOL	Reading and writing data of free port
Error	BOOL	An error occurred while reading and writing data
ErrorID	WORD	Function block error code =0: Correct =1: IMType parameter error =2: SlotAddr specified slot range error =3: SlotAddr the communication board type of the specified slot is not supported =4: IO module DP slave address specified by IOAddr is incorrect =5: The interface module address specified by IMAddr is incorrect =7: The specified address module type of IOAddr is wrong (this station is not configured with LK238 module, please check whether the IMType, SlotAddr, IMAddr and IOAddr parameters are configured correctly) =8: I / O byte length error (please check the I / O module on the module configuration page) =9: Port parameter configuration error =10: The length of the send data exceeds the output length by bytes (the length of the send data includes the length of 10 bytes of the data header) =11: The received data length exceeds the input length by bytes (the received data length includes the 7-byte length of the data header) =12: Abnormal send length setting (equal to 0 or greater than 210) =13: Abnormal reception length setting (greater than 210) =14: The port does not work (When Com Status in the user parameter page is disable; in the COMM_CTL_MODULE function block, the corresponding port in the ComEnable parameter is not enabled) =15: Abnormal communication between controller and serial port module =16: Port protocol mismatch =17: Both receive enable and send enable are set to 0 (at least one of receive and send needs to be enabled in receive-send mode)
FreeportOut	FREEPORTOUTPARAM	Free port protocol output data

FreeportInParam Structure member

Member variables	Type	Parameter Description
SendEn	BOOL	Send enable(high level enable)
SendLength	BYTE	Set sending length(1~210)
SendTimeout	BYTE	Set sending timeout time (1~255, unit: 10ms)
RecvEn	BOOL	Receive enable(configed as receiving-sending mode in user parameter,RecvEn is enabled by high level. configed as only-receiving mode in user parameter,RecvEn is enabled by rising edge, receive data during the high level.)
RecvLen	BYTE	Set receiving length(0 ~ 210) (when the start character and end character are enabled, the reception length can be set to 0)
StartChar	BYTE	Set start character(1~255)
EndChar	BYTE	Set ending character(1~255)
RecvTimeout	BYTE	Set the receiving timeout time (1`255, unit: 10ms)
RecvMode	BYTE	Receiving mode bit0: timeout time enable bit1: ignore bit2: end character enable bit3: start character enable bit4-7: ignore
AckID	BYTE	The master station confirms the package number

FreeportOutParam Structure member

Member variables	Type	Parameter Description
SendQ	BOOL	Sending end
SendErr	BYTE	Send status =0: correct =2: Timeout
RecvQ	BOOL	Reception end
RecvCount	BYTE	Bytes received
RecvErr	BYTE	End state =0: correct =2: The start condition is missing (enable the start character but set the start character to 0; enable the end character but not enable the start character, and receive the length is 0) =3: The end condition is missing (enable the end character but set the end character to 0; enable the start character but not enable the end character, and receive the length is 0) =4: The timeout setting is too small =5: Timeout =6: End character not found
PackID	BYTE	LK238 returns command number

3.4.2.11 COMM_CTL_MODULE(LK238 Module Control Function Block)

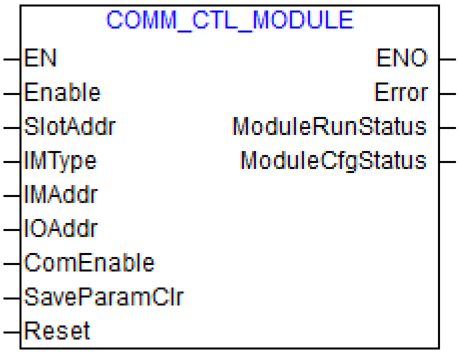
3.4.2.11.1 Function Description

This function block can enable four serial ports, empty the MODBUS master instruction, reset interactive interface.The instruction has a corresponding example project, which can be directly imported for use.

When importing algorithm examples, you can import them in the following two ways:

- In the menu bar, click [Project] - [Import Sample Project]. For details, see import sample project.
- In the library manager, right-click the function block and select the [Import Sample Project] command. For details, see LK238 serial port communication module.

The imported example project needs to be called in the program or under the task to be used normally.



COMM_CTL_MODULE

3.4.2.11.2 Parameter description

Input Parameter	Type	Parameter Description
Enable	BOOL	High level enable
SlotAddr	BYTE	Slot number, the effective value is 2~5
IMType	BYTE	Type of interface module used in the backplane where the IO module is located: 1: The interface module type is LK231 or LK232. At this time, the IMAddr parameter is meaningless 2: The interface module type is LK235/LK271 3: The interface module type is LK250 4: The interface module type is LK234/LK235M. At this time, the SlotAddr parameter is meaningless
IMAddr	DWORD	The interface module address used by the backplane where the IO module is located When the interface module is LK231 or LK232, the IMAddr value is meaningless and does not need to be set When the interface module is LK235/LK271, IMAddr is consistent with the setting value of the module address dial switch When the interface module is LK250, IMAddr is consistent with the setting value of the expansion backplane dial switch When the interface module is LK234/LK235M, IMAddr is the hexadecimal value of IP address of LK234. For example, the IP address is 192.168.1.250, and the IMAddr parameter is 16#C0A801FA If the IMAddr configuration is wrong, the ErrorID will report 7
IOAddr	BYTE	DP slave address of the backplane where the IO module is located (2 ~ 125)
ComEnable	MODULECONTROLPARAM	Working enable bits of COM1 ~ COM4
SaveParamClr	BOOL	Clear power down holding
Reset	BOOL	Reset control

Output Parameter	Type	Parameter Description
Error	BYTE	Function block error code =0: Correct =1: IMType parameter error =2: SlotAddr specified slot range error =3: SlotAddr the communication board type of the specified slot is not supported =4: IO module DP slave address specified by IOAddr is incorrect =5: The interface module address specified by IMAddr is incorrect =7: The specified address module type of IOAddr is wrong (this station is not configured with LK238 module, please check whether the IMType, SlotAddr, IMAddr and IOAddr parameters are configured correctly)
ModuleRunStatus	MODULERUNSTATUSPARAM	Module working status
ModuleCfgStatus	MODULECFGSTATUSPARAM	Module configuration status

ModuleControlParam Structure member

Member variables	Type	Parameter Description
COM1_EN	BOOL	COM1 work enabled
COM2_EN	BOOL	COM2 work enabled
COM3_EN	BOOL	COM3 work enabled
COM4_EN	BOOL	COM4 work enabled

ModuleRunStatusParam Structure member

Member variables	Type	Parameter Description
ComRunStatus1	BOOL	COM1 working status
ComRunStatus2	BOOL	COM2 working status
ComRunStatus3	BOOL	COM3 working status
ComRunStatus4	BOOL	COM4 working status

ModuleCfgStatusParam Structure member

Member variables	Type	Parameter Description
ComCfgStatus1	BOOL	COM1 configuration status
ComCfgStatus2	BOOL	COM2 configuration status
ComCfgStatus3	BOOL	COM3 configuration status
ComCfgStatus4	BOOL	COM4 configuration status

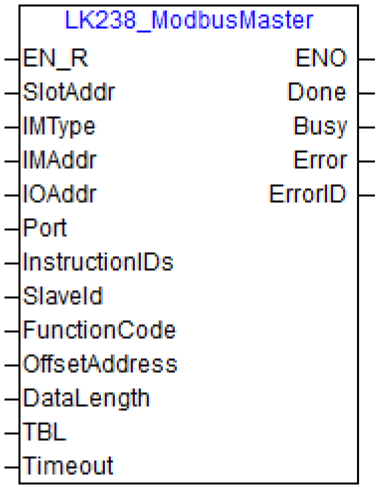
3.4.2.12 LK238_ModbusMaster(LK238 Modbus Master Configure Read and Write)

3.4.2.12.1 Function Description

Read or write LK238 Modbus master data through this function block. When EN_ When R detects the rising edge, read or write data once. Read/write data is stored in the data cache area pointed to by the

TBL pointer, and the cache size definition must be ≥ 224 bytes.

Note: This function block is supported in FA-AT V3.1.10B1Patch1 and above versions.



sysReadSOE

3.4.2.12.2 Parameter Description

Input Parameter	Type	Parameter Description
EN_R	BOOL	Rising edge execution
SlotAddr	BYTE	Slot number, the effective value is 2~5
IMType	BYTE	The type of interface module used on the backplane where the IO module is located: 1: The interface module type is LK231 or LK232, and in this case, the IMAddr parameter is meaningless 2: The interface module type is LK235/LK271 3: The interface module type is LK250 4: The interface module type is LK234/LK235M, and in this case, the SlotAddr parameter is meaningless
IMAddr	DWORD	Address of the interface module used on the backplane where the IO module is located: When the interface module is LK231 or LK232, the IMAddr value is meaningless and does not need to be set When the interface module is LK235/LK271, the IMAddr and module address dial switch settings remain consistent When the interface module is LK250, the IMAddr and expansion backplane dial switch settings remain consistent When the interface module is LK234/LK235M, IMAddr is the result of converting the four segments of the module's IP address into hexadecimal and merging them, for example, the IP address 192.168.1.250, and the IMAddr parameter is 16 # C0A801FA If IMAddr configuration error occurs, ErrorID will report 7
IOAddr	BYTE	LK238 module slave address (2-125)
Port	BYTE	Port numbers (1-4)
InstructionIDs	BYTE	Instruction ID numbers (1-32) cannot be duplicated under the same Port
SlaveId	BYTE	Slave address (1-247)
FunctionCode	BYTE	Function code 0x01 (read coil), 0x02 (read discrete input), 0x03 (read hold register), 0x04 (read input register), 0x05 (write single coil), 0x06 (write single register), 0x0F (write multiple coils), 0x10 (write multiple registers) When setting, write the corresponding function code decimal value. For example, 0x01 (read coil) function code with a set value of 1, and 0x0F (write multiple coils) function

Input Parameter	Type	Parameter Description
		code with a set value of 15
OffsetAddress	WORD	Read/write offset address, set range from 0 to 65535
DataLength	WORD	When setting, the data length+offsetAddress should be ⩽ 65536. For example, if the address is set to 65535, the data length must be set to 1 Switching value: 1-1600 Analog quantity: 1-100
TBL	POINTER TO BYTE	The main station stores data addresses, and the buffer size pointed to must be ⩾ 224 Bytes
Timeout	WORD	The main station stores data addresses, and the buffer size pointed to must be ⩾ 224 Bytes When setting ,the timeout ＞Tran_Byte/(Baud rate/(8*1000))ms, Tran_Byte is the total length of input/output bytes and Baud rate is the set Modbus communication baud rate

Output Parameter	Type	Parameter Description
Done	BOOL	Completion of reading and writing master station data
Busy	BOOL	Reading and writing master data in progress
Error	BOOL	An error occurred while reading and writing data from the main station
ErrorID	WORD	Function block error code =0: No errors =1: IMType protocol type error =2: SlotAddr specified slot range error =3: The communication board type specified in SlotAddr slot is not supported =4: The IO module DP slave address specified by IOAddr is incorrect =5: IMAddr specified interface module address error =7: The IOAddr specified address module type is incorrect (this station is not configured with an LK238 module, please check if the IMType, SlotAddr, IMAddr, and IOAddr parameter configurations are correct) =8: Input/output byte length error (please check the module configuration page for input/output modules) =9: Port parameter configuration error =10: Illegal slave address (1-247) =11: Illegal access address (OffsetAddress+DataLength exceeds 0x10000) =12: Illegal function code (01 02 03 04 05 06 15 16) =13: The user parameter configuration for this port is not a Modbus master station =14: Illegal data length (switching value: 1-1600 analog value: 1-100) =15: Abnormal communication between controller and serial module =16: InstructionIDs parameter configuration error Master station error message =1001: Data frame error =1002:Timeout =1005: The port is not working(when port Com Status is Disable in the user parameter page) =1128+1: Function code not supported by the slave station =1128+2: Data address overflow =1128+3: Data Range Overflow =1128+4: Slave equipment failure =1128+6: Slave device busy

3.4.2.12.3 Application Example of Instructions

Example explanation:

LK238 reads slave data through COM1 port. The slave station ID number is 1, and the data of 100 coils with offset address starting from 0 is read.

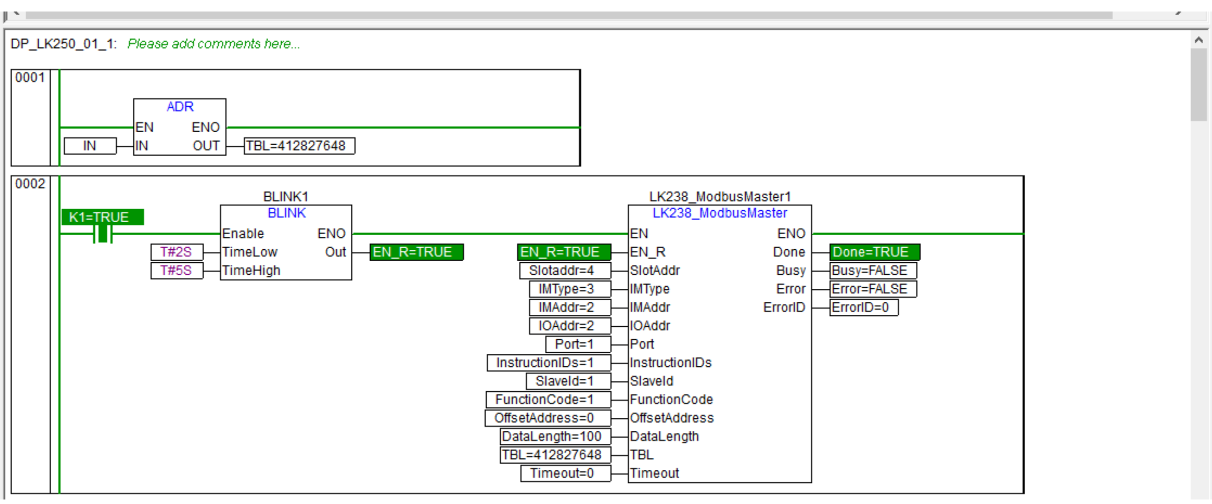
DP_LK250_01_1(PRG).ld

No.	Variable Name	Address	Variable Description	Variable Type	Online Value	Power Fail Safeguard	SO
0001	IN			ARRAY[0..223] OF BYTE		FALSE	
0002	TBL			POINTER TO BYTE	412827648	FALSE	
0003	K1			BOOL	TRUE	FALSE	
0004	BLINK1			BLINK		FALSE	
0005	LK238_ModbusMaster1			LK238_MODBUSMASTER		FALSE	
0006	EN_R			BOOL	FALSE	FALSE	
0007	Slotaddr			BYTE	4	FALSE	
0008	IMType			BYTE	3	FALSE	
0009	IMAddr			DWORD	2	FALSE	
0010	IOAddr			BYTE	2	FALSE	
0011	Port			BYTE	1	FALSE	
0012	InstructionIDs			BYTE	1	FALSE	
0013	SlaveId			BYTE	1	FALSE	
0014	FunctionCode			BYTE	1	FALSE	
0015	OffsetAddress			WORD	0	FALSE	
0016	DataLength			WORD	100	FALSE	
0017	Timeout			WORD	0	FALSE	
0018	Done			BOOL	FALSE	FALSE	
0019	Busy			BOOL	FALSE	FALSE	
0020	Error			BOOL	FALSE	FALSE	
0021	ErrorID			WORD	0	FALSE	

Variable Definition

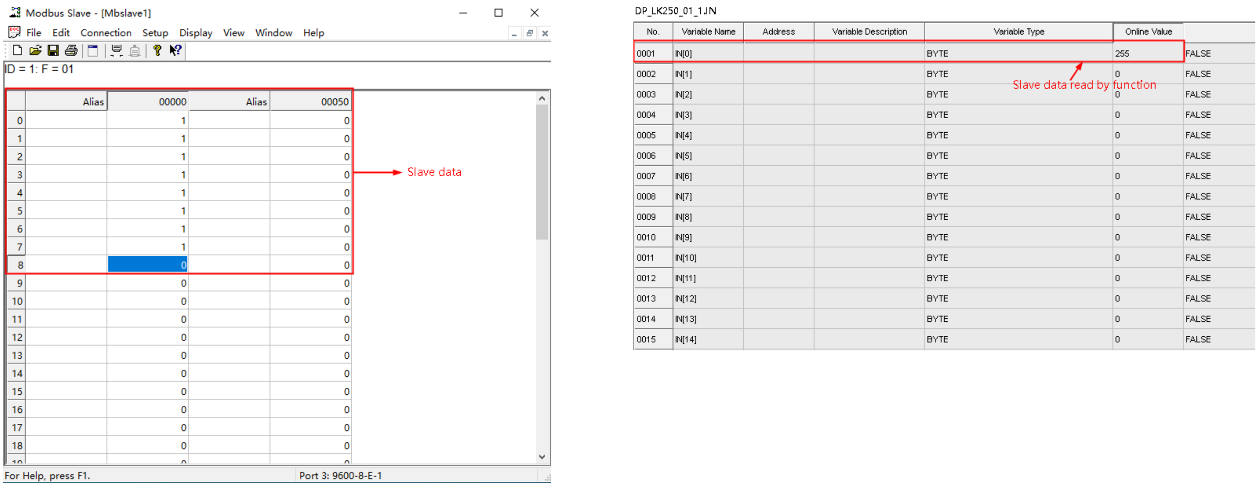
The program obtains the address of the array IN through the ADR function block, and the TBL pointer points to the array address.

When K1 is true, trigger a 2ms pulse signal, LK238_ The ModbusMaster function block reads the slave data and stores it in the array IN.



Program Example

The offset address 0-7 data from the station is 1, and it is read into the array IN [0] variable with a value of 255.



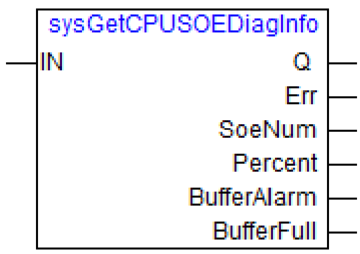
Schematic Diagram for Reading Slave Station Data

3.4.3 SOE Instruction

3.4.3.1 sysGetCPUSOEDiagInfo(Get RTS SOE Diagnose Information)

3.4.3.1.1 Function

Get the diagnosis information of SOE events from storage area.



sysGetCPUSOEDiagInfo

3.4.3.1.2 Parameter Description

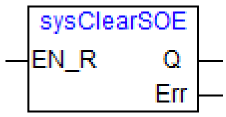
Input Parameter	Type	Parameter Description
IN	BOOL	Enable input, level signal 0: invalid 1: valid

Output Parameter	Type	Parameter Description
Q	BOOL	Flag of Output state 0: Output not completed 1: Output completed
Err	BYTE	Error value 0: Get SOE diagnosis information successfully
SoeNum	WORD	The number of current SOE stored in the controller
Percent	REAL	The percentage of current SOE and the maximum number of SOE, the maximum storage of SOE is 6000
BufferAlarm	BOOL	When SOE exceeds 70% of the maximum storage, set alarm flag to 1
BufferFull	BOOL	When SOE exceeds 100% of the maximum storage, set alarm flag to 1

3.4.3.2 sysClearSOE(Clear RTS SOE Information)

3.4.3.2.1 Function

Clear SOE events stored in controller and SOE module.



sysClearSOE

3.4.3.2.2 Parameter Description

Input Parameter	Type	Parameter Description
EN_R	BOOL	Enable signal with rising edge

Output Parameter	Type	Parameter Description
Q	BOOL	Flag of Output state 0: Output not completed 1: Output completed
Err	BYTE	Error value 0: Clear SOE information successfully

3.4.3.3 sysReadSOE(Read RTS SOE Information)

3.4.3.3.1 Function Description

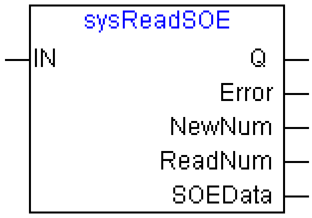
Read SOE data from the SOE module, with a maximum of 50 entries read at a time. When IN is the rising edge, a data read is triggered. The read data is stored in the SOEData structure array.

The data that supports SOE event includes the DI channel data of the LK631 SOE module and the BOOL type variable with SOE enabled. The LK631 module needs to set SOE of DI channel function in the "User

Parameters" in order to enable SOE function.

Function block can only be called once.

Note: This function block is supported in FA-AT V3.1.10B1Patch1 and above versions, and compatible firmware version is LK220-G02.



sysReadSOE

3.4.3.3.2 Parameter Description

Input Parameter	Type	Parameter Description
IN	BOOL	Rising edge enable

Output Parameter	Type	Parameter Description
Q	BOOL	Output TRUE: Instruction execution successful FALSE: Instruction execution failed
Error	BYTE	Error code 0: Correct 1: Internal SOE queue error 2: Configuration SOE data receive buffer error 3: No new SOE event data generated 200:Function block repeated calling
NewNum	WORD	Number of unread SOE events since last read
ReadNum	WORD	The number of SOE events read this time, with no more than 50 events read each time
SOEData	ARRAY [0 49] of STSOEDATA	A cache area for storing 50 SOE events, with a maximum of 50 SOE events The definition of structural entity STSOEDATA is detailed in the table below

The following table provides a detailed definition of SOE data:

Member variables	Type	Meaning	Byte	
			Serial Number	Description
Seconds	DWORD	The second time of SOE event occurrence	Byte1	Lowest Byte of Time (seconds)
			Byte2	Time (seconds) low byte
			Byte3	Time (seconds) high byte
			Byte4	Time (seconds) highest byte
T1	WORD	The second time of SOE event occurrence	Byte1	Time (hundreds of microseconds) low byte
			Byte2	Time (hundreds of microseconds) high byte

Member variables	Type	Meaning	Byte	
			Serial Number	Description
ulOffset	DWORD	The offset address of the data area where SOE events occur is currently fixed as a bit offset. For example, if the channel address of the data is% IX4.6, the offset address is: 4 * 8+6=38 bytes	Byte1	Offset Address Lowest Byte
			Byte2	Offset address low byte
			Byte3	Offset address high byte
			Byte4	Offset address highest byte
ChannelData	BYTE	Channel information of SOE event occurrence	Byte1	Bit0~Bit3: Data area index, indicating which data area SOE point belongs to, for example: 0 represents M area; 1 represents I area; 3 represents the R area; 4 represents N area Bit4: Significant bit. 1 indicates valid, 0 indicates possibly invalid. For example, when the timing is invalid, the significant bit of the recorded SOE event is 0, indicating that the time of the recorded data may be inaccurate Bit5: Offset address type, where 0 represents bit offset and 1 represents byte offset Bit6~Bit7: The current value of the channel, which is 0 or 1

When viewing SOE data information, the data area can be determined by the Bit0~Bit3 values of the ChannelData variable, and then the offset address can be determined by ulOffset to locate the SOE variable.

- For channel data, the channel address can be directly calculated by offset to find the corresponding SOE channel variable.

  As shown in the following figure, the current value of the ChannelData variable is 17, corresponding to a binary value of 00010001. The values of Bit0~Bit3 are 1, which determines the data from I area. Bit4 is 1, the current SOE data is valid. Bit6~Bit7 values are 0, then the current value is 0 (representing a jump from 1 to 0).

PRG_SOE.SOEData3

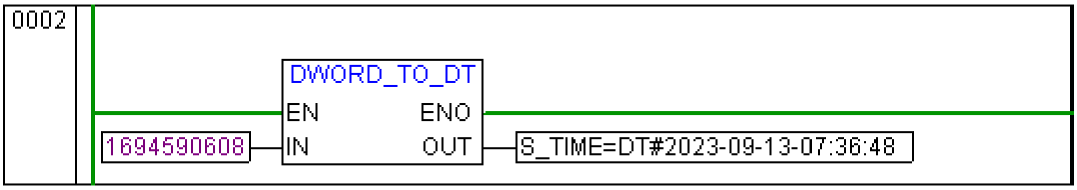
No.	Variable Name	Address	ariable Descriptio	Variable Type	Online Value	Power Fail Safeguard
0001	SOEData3[0]			STSOEDATA		FALSE
0002	SOEData3[1]			STSOEDATA		FALSE
0003						
0004						
0005						
0006						
0007						
0008						

PRG_SOE.SOEData3[0]

No.	Variable Name	Address	Variable Description	Variable Type	Online Value	Power Fail Saf
0001	Seconds		The second time of SOE event occurrence	DWORD	1694590608	FALSE
0002	T1		Hundred microseconds of time of SOE event o...	WORD	6416	FALSE
0003	ulOffset		Data area offset during SOE events	DWORD	9931	FALSE
0004	ChannelData		Channel information of SOE event occurrence	BYTE	17	FALSE

SOE Data Information_Channel Data

As shown in the figure below, the second time resolution is: 2023-09-13-07: 36:48. The hundred microseconds time T1 in the above figure is 6416, so the time of SOE event occurrence is: 2023-09-13-07:40:23 (641.6 ms)



Second Time Resolution to DT

The ulOffset offset address is 9931, and the calculated channel address is:% IX1241.3. Find the address % IX1241.3 for the whole project and locate it to the corresponding channel address, as shown in the following figure.

LK631_284(37:LK631)

Device Configuration | Channel | Diagnostic Information | Information

0_14 Channels Digital Input Module

Channel Number	Channel Name	Channel Type	Channel Address
1	DPIO_261_1_37_1	BOOL	%IX1240.0
2	DPIO_261_1_37_2	BOOL	%IX1240.1
3	DPIO_261_1_37_3	BOOL	%IX1240.2
4	DPIO_261_1_37_4	BOOL	%IX1240.3
5	DPIO_261_1_37_5	BOOL	%IX1240.4
6	DPIO_261_1_37_6	BOOL	%IX1240.5
7	DPIO_261_1_37_7	BOOL	%IX1240.6
8	DPIO_261_1_37_8	BOOL	%IX1240.7
9	DPIO_261_1_37_9	BOOL	%IX1241.0
10	DPIO_261_1_37_10	BOOL	%IX1241.1
11	DPIO_261_1_37_11	BOOL	%IX1241.2
12	DPIO_261_1_37_12	BOOL	%IX1241.3
13	DPIO_261_1_37_13	BOOL	%IX1241.4
14	DPIO_261_1_37_14	BOOL	%IX1241.5

Channel Data

The SOE event information is as follows: DPIO_ 261_ 1_ 37_12 channel of LK631 module occurs a jump from 1 to 0 at 2023-09-13-07:40:23（641.6 ms）.

SOE Event Information

Jump Time	POU Name/Module Node Name	Variable Name	Value	SOE Significant Bit
2023-09-13-07:40:23（641.6 ms)	LK631_284(37:LK631)	DPIO_261_1_37_12	from 1 to 0	Bit4 is 1, the current SOE data is valid

■ For BOOL type variables in PRG, it is necessary to export the variable table, find the corresponding data area in the variable table, and find the BOOL type variable corresponding to the ulOffset offset address.

As shown in the following figure, the current value of the ChannelData variable is 52, corresponding to a binary value of 00110100. The values of Bit0~Bit3 are 4, which determines the data from N area. Bit4 is 1, the current SOE data is valid. Bit6~Bit7 values are 0, then the current value is 0 (representing a jump from 1 to 0).

PRG_SOE.SOEData3

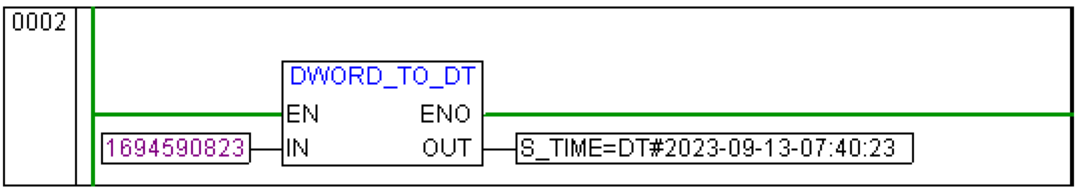
No.	Variable Name	Address	ariable Descriptio	Variable Type	Online Value	Power Fail Safeguard
0001	SOEData3[0]			STSOEDATA		FALSE
0002	SOEData3[1]			STSOEDATA		FALSE
0003						
0004						
0005						
0006						
0007						
0008						

PRG_SOE.SOEData3[0]

No.	Variable Name	Address	Variable Description	Variable Type	Online Value	Power Fail Saf
0001	Seconds		The second time of SOE event occurrence	DWORD	1694590823	FALSE
0002	T1		Hundred microseconds of time of SOE event o...	WORD	7110	FALSE
0003	ulOffset		Data area offset during SOE events	DWORD	1894	FALSE
0004	ChannelData		Channel information of SOE event occurrence	BYTE	52	FALSE

SOE Data Information_Channel Data

As shown in the figure below, the second time resolution is: 2023-09-13-07:40:23. The hundred microseconds time T1 in the above figure is 7110, so the time of SOE event occurrence is: 2023-09-13-07:40:23（711.0 ms）.



Second Time Resolution to DT

The ulOffset offset address is 1894. Find the corresponding variable with the offset address 1894 in the N area of the variable table, as shown in the following figure.

ProjectName: TEST						
DateTime: 2023-09-15 11-02-22						
VarName	GroupName	Comment	Type	DataArea	Offset	Size
RCP_MOVE_BLOCK.Error	DATA BLOCK MOV	Error message	BYTE	G	14	1
p5	Main		BOOL	G	20	1
p4	Main		BOOL	G	21	1
p7	Main		BOOL	G	22	1
g1	GV_Group		BOOL	G	16	1
g2	GV_Group		BOOL	G	17	1
SOE_BOOL10	PRG_SOE		BOOL	G	1894	1
Percent1	PRG_SOE	SOE AREA	REAL	G	18028	4
SOE_BOOL11	PRG_SOE		BOOL	G	18032	1
SOE_BOOL12	PRG_SOE		BOOL	G	18033	1
SOE_BOOL13	PRG_SOE		BOOL	G	18034	1
SOE_BOOL14	PRG_SOE		BOOL	G	18035	

Channel Data

The SOE event information is as follows: Variable SOE_BOOL10 in PRG_SOE program occurs a jump from 1 to 0 at 2023-09-13-07:40:23（711.0 ms）.

SOE Event Information

Jump Time	POU Name/Module Node Name	Variable Name	Value	SOE Significant Bit
2023-09-13-07:40:23（711.0 ms）	PRG_SOE	SOE_BOOL10	from 1 to 0	Bit4 is 1, the current SOE data is valid

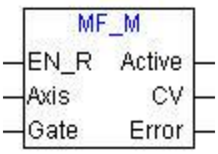
3.5 Industry Application Instruction

3.5.1 Motion Control (LE) Function

3.5.1.1 MF_M (Frequency Measurement, 5128)

3.5.1.1.1 Description of Function

Measure the pulse frequency of input channel appointed by Axis.



MF_M Function Block

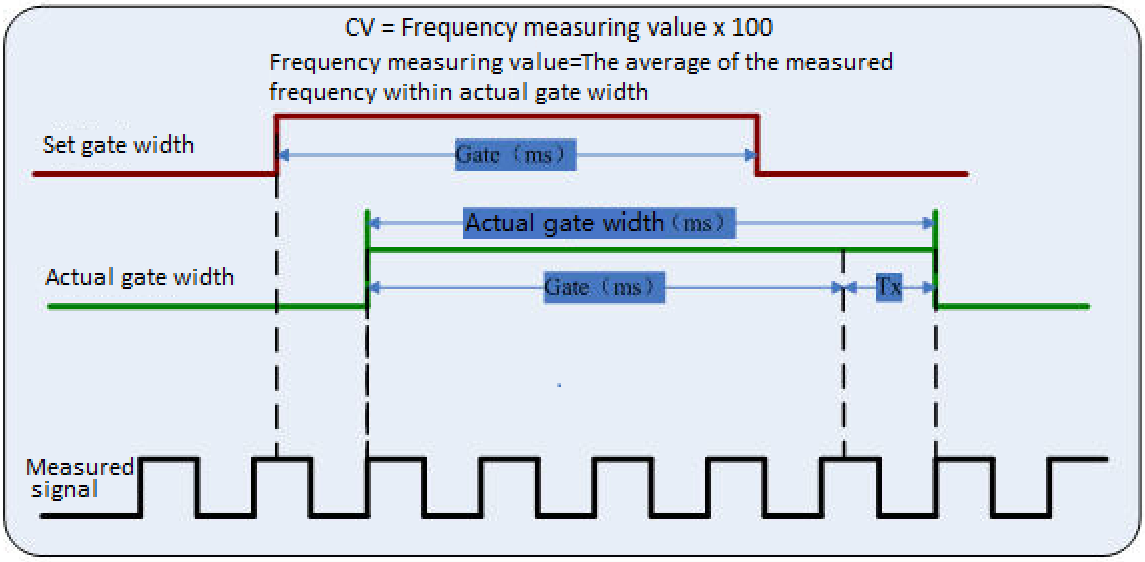
3.5.1.1.2 Parameter Description

Input Parameter	Type of Data	Description of Function	Description of Parameter Value
EN_R	BOOL	Enable	0→1 Rising edge enable, high level keep
Axis	BYTE	Choice of input axis	=1,2,3,4 respectively represents the input axis: axis1, axis2, axis3, axis4
Gate	WORD	Gate width of frequency detected	Set range is 1~100 (ms) If more than 100, taking 100 as set value If less than 1, taking 1 as setting value

Output Parameter	Type of Data	Description of Function	Description of Parameter Value
Active	BOOL	Status mark of measuring frequency	0: Stop measuring frequency 1: In the process of measuring frequency
CV	DWORD	Current frequency value	1000～10,000,000
Error	BYTE	Error information	= 0: Correct = 1: This channel is occupied = 2: Configured axis error

3.5.1.1.3 Instruction Description

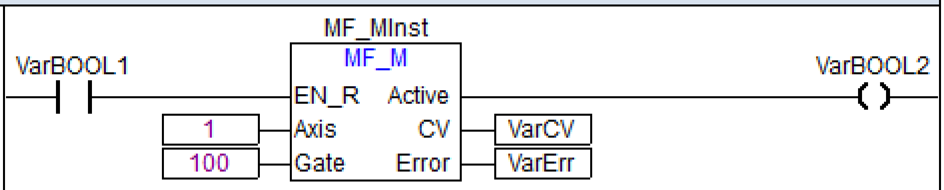
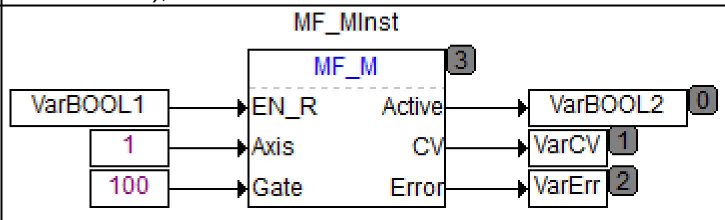
- When Function Block enable signal is valid, measuring pulse frequency in real time; if the enable signal is invalid, stopping measure, and CV shows 0.
- If within the setting gate width and don't test an integral pulse, then enlarge gate width automatically until testing next edge. If can't test next edge within 100ms range, then doesn't test any more, CV shows 0.
- For the pulse input of the same channel, high speed counting instruction and frequency measurement instruction are permitted to use at the same time.
- CV value= input pulse frequency x 100.



Signal Relationship

3.5.1.1.4 Application Example of Instructions

Variable Definition						
No.	Variable Name	Address	Variable Description	Variable Type	Initial Value	Power Fail Safeguard
0001	MF_MInst			MF_M		FALSE
0002	VarCV			DWORD	0	FALSE
0003	VarErr			BYTE	0	FALSE
0004	VarBOOL1			BOOL	FALSE	FALSE
0005	VarBOOL2			BOOL	FALSE	FALSE
Programing language	Program					

<table>
<tr><th colspan="2">Variable Definition</th></tr>
<tr><td>Ladder diagram (LD)</td><td>
</td></tr>
<tr><td>Structured text (ST)</td><td>MF_MInst(
EN_R:=VarBOOL1,
Axis:=1,
Gate:=100,
Active=> VarBOOL2,
CV=>VarCV,
Error=>VarErr);</td></tr>
<tr><td>Continuous function diagram (CFC)</td><td>
</td></tr>
</table>

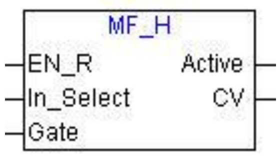
Program description:

When EN_R is set and held, the axis 1 channel receives the pulse and the CV shows the pulse frequency value of the current pulse received.

3.5.1.2 MF_H (Frequency Measurement,5106/5107/5108/5109/5118)

3.5.1.2.1 Description of Function

Measuring the pulse frequency input in the channel pointed by In_Select, the valid measuring range 10 Hz~100 kHz.



MF_H Function Block

3.5.1.2.2 Description of Parameter

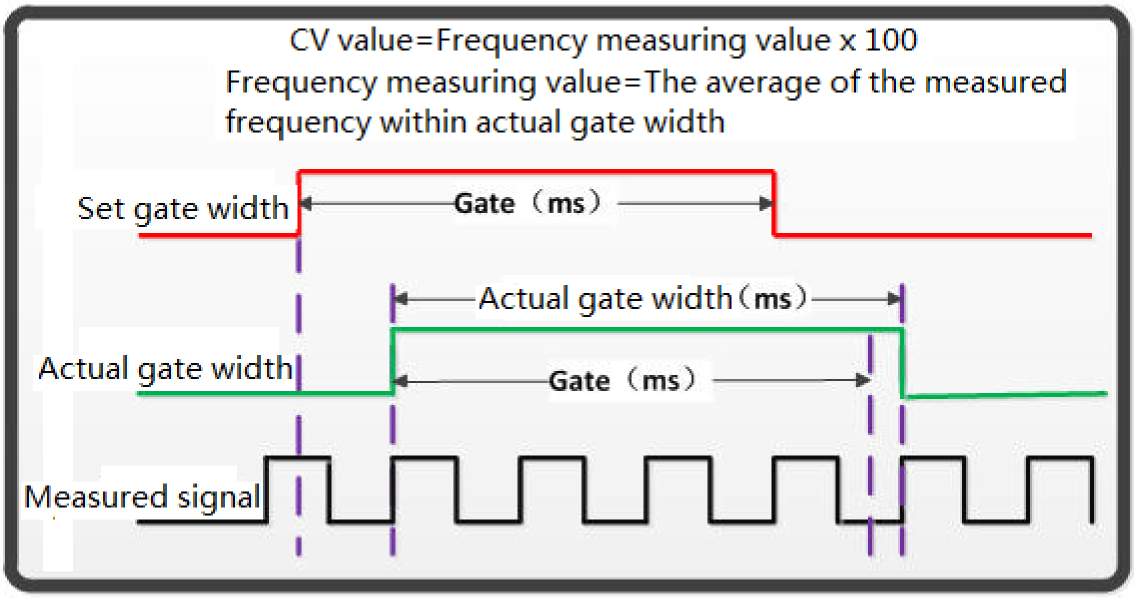
Input Parameter	Type of Data	Description of Function	Description of Parameter Value
EN_R	BOOL	Enable	0→1 Rising edge enable, high level keep
In_Select	BYTE	Choice of input channel (if the module only supports 4 channel for frequency testing, then choose the front 4 channels.)	=1: I0.0 is pulse input end =2: I0.1 is pulse input end =3: I0.2 is pulse input end

Input Parameter	Type of Data	Description of Function	Description of Parameter Value
			=4: I0.3 is pulse input end =5: I1.4 is pulse input end =6: I1.5 is pulse input end =7: I1.6 is pulse input end =8: I1.7 is pulse input end
Gate	WORD	Gate width of frequency testing	Set range is 1~2000 (ms) If more than 2000, taking 2000 as set value If less than 1, taking 1 as setting value

Output Parameter	Type of Data	Description of Function	Description of Parameter Value
Active	BOOL	Status mark of measuring frequency	0: Stop measuring frequency 1: In the process of measuring frequency
CV	DWORD	Current frequency value	50～10,000,000, (resolution ratio: 0.01 Hz), there is no sense if the data is less than 1000.

3.5.1.2.3 Instruction Description

- When Function Block enable signal is valid, testing pulse frequency in real time; if the enable signal is invalid, stopping test, and CV shows 0.
- If within the setting gate width, and don't test an integral pulse, then enlarge gate width automatically until testing next edge. If can't test next edge within 2000ms range, then doesn't test any more, CV shows 0.
- For the pulse input of the same channel, high speed counting instruction and frequency testing instruction are permitted to use at the same time.
- CV value= frequency measuring value x 100.



Signal Relationship

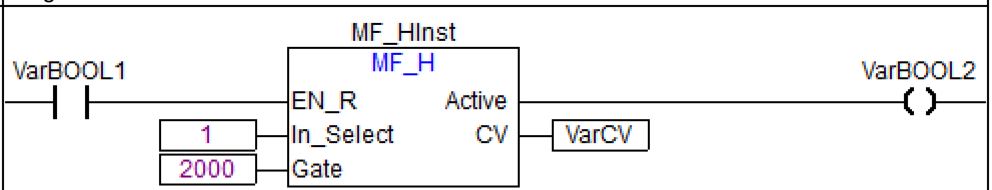
Number of Channels of frequency Measure Supported by Module

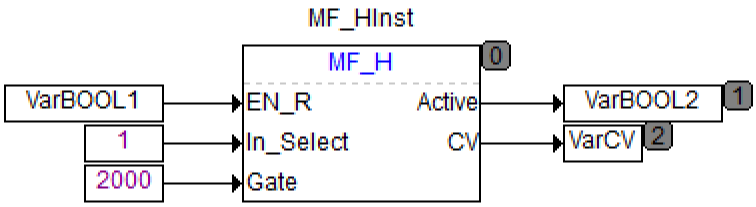
Module Type	LE5106	LE5107	LE5108	LE5109	LE5128
Number of MF channels	4	8	4	8	8

3.5.1.2.4 Application Example of Instructions

Variable Definition

No.	Variable Name	Address	Variable Description	Variable Type	Initial Value	Power Fail Safeguard
0001	MF_HInst			MF_H		FALSE
0002	VarCV			DWORD	0	FALSE
0003	VarBOOL1			BOOL	FALSE	FALSE
0004	VarBOOL2			BOOL	FALSE	FALSE

Programing language	Program
Ladder diagram (LD)	
Structured text (ST)	MF_HInst(EN_R:=VarBOOL1, In_Select:=1, Gate:=2000, Active=> VarBOOL2, CV=>VarCV);

Variable Definition	
Continuous function diagram (CFC)	

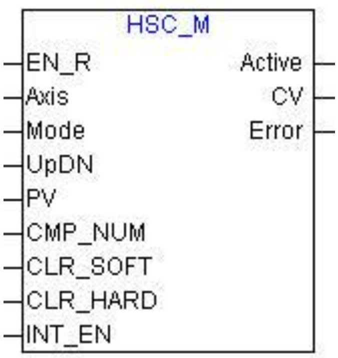
Program description:

When EN_R is set and held, the I0.0 channel receives a 50Hz pulse, CV shows the pulse frequency value of the current pulse received, VarCV = 4999 (about 50Hz).

3.5.1.3 HSC_M (High Speed Counting, 5128)

3.5.1.3.1 Description of Function

High-speed counting function block can achieve duplex count of A,B or increase or decrease count of one-edge of A, B .Users can set up to 21 different counting target value, and choose whether to start interruption after reaching set value or not. When A and B phase counting, one can receive the difference between the range of 90°±60°. The counting value range shall be -1,073,741,824～1,073,741,823.



HSC_M Function Block

3.5.1.3.2 Description of Parameter

Input Parameter	Type of Data	Description of Function	Instruction of Parameter Value
EN_R	BOOL	Enable	0→1: Rising edge enable, level preserved
Axis	BYTE	Input axis selection	Single-phase (bit5=0): =1, 2, 3, 4 Separately shows the input axis is: axis 1, axis 2, axis 3, axis 4 Double-phase (bit5=1): =1, 3 Separately shows the input axis is : axis 1, axis 3 If input 2 still corresponding to axis 1 input, if input 4 still corresponding to axis 3 input
Mode	BYTE	Mode selection	See details in Table Instruction_of_Mode_Parameter

Input Parameter	Type of Data	Description of Function	Instruction of Parameter Value
UpDN	BOOL	When single-phase counting, counting direction control	0: Count upward 1: Count downward
PV	ARRAY[21] OF DINT	Set counting value	-1,073,741,824～1,073,741,823 ARRAY[0]～ARRAY[20]
CMP_NUM	BYTE	The number of comparing	1~21
CLR_SOFT	BOOL	Software clearing	Set the Mode (bit2=0) for software clearing (if bit4=0: rising edge clearing, bit4=1, falling edge clearing) to achieve clearing of CV value.
CLR_HARD	BYTE	Hardware clearing	Set the Mode (bit2=1) for hardware clearing (if bit4=0:rising edge clearing, bit4=1, falling edge clearing) to achieve clearing of CV value. 0: Invalid; 1~16: Separately show clearing channel of hardware is I0.0~I1.7
INT_EN	BOOL	Interruption enable	0: Doesn't interrupt after reaching count 1: Interrupt after reaching count

Output Parameter	Type of Data	Description of Function	Instruction of Parameter Value
Active	BOOL	Mark of reaching count	0: Count doesn't reach set value 1: Count reaches set value
CV	DINT	Current count value	-1,073,741,824～1,073,741,823
Error	BYTE	Error information	0: Correct =1: Input channel is taken up =2: Channel selection is error =3: Compare number setting is error =4: PV setting is error =5: Mode setting is error =6: Hardware clearing channel setting is error =7: Set count value error（over 31 bits valid values）

Instruction of Mode Parameter

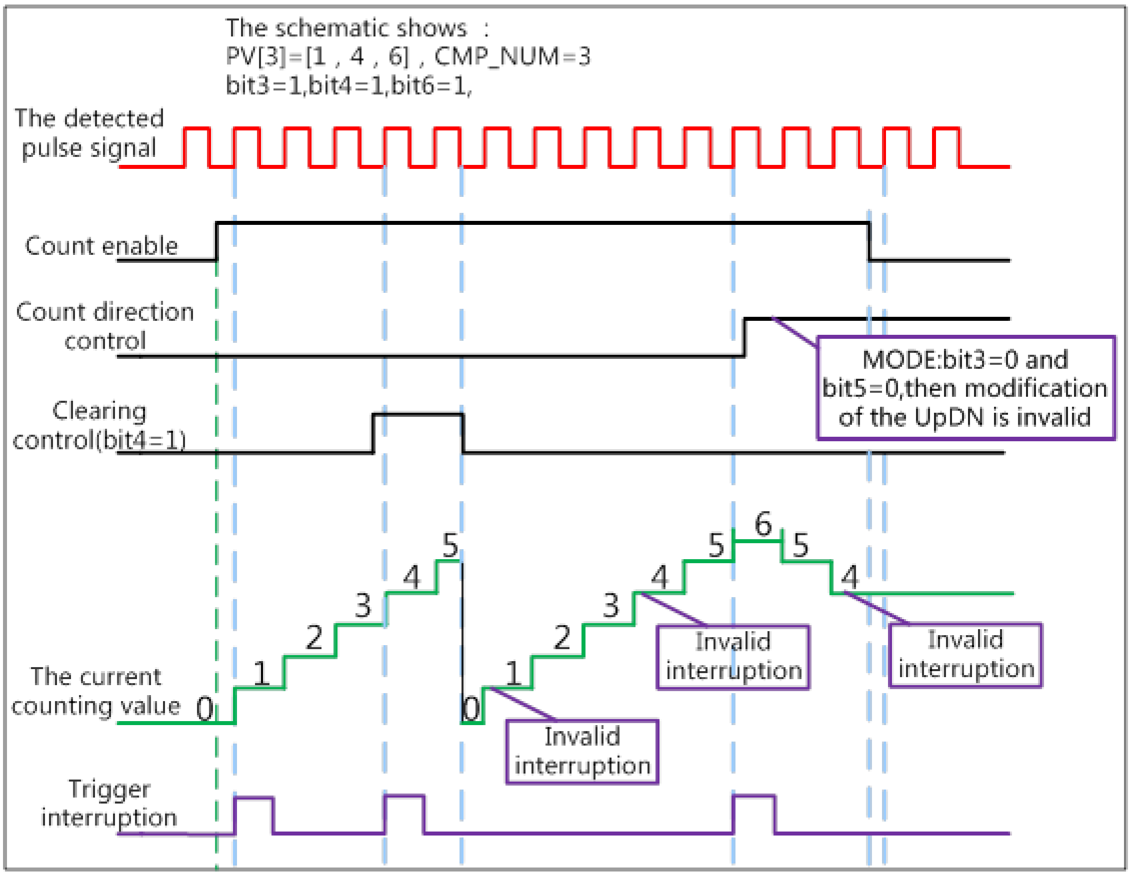
Serial No.	Overview of Function	Description
Bit7	-	Reserve
Bit6	Count continuity selection	0: Count to PV value then stop counting 1: Count to PV value then continue to count
Bit5	One and double-phase count selection	0: Single-phase count 1: Double-phase count
Bit4	Valid clearing edge selection	0: Rising edge clearing 1: Falling clearing
Bit3	Counting direction selection	0: Software control, parameter UpDN controls counting direction 1: Hardware control, pointed DI channel (I0.4, I0.5, I0.6, I0.7) control counting direction
Bit2	Clearing method selection	0: Software clearing, parameter CLR_SOFT control clearing 1: Hardware clearing, parameter CLR_HARD pointed DI channel controlling clearing
Bit1～Bit0	Frequency doubling mode selection	00: Double (one) phase 1 time frequency count 01: Double (one) phase 2 times frequency count 10: Double phase 4 times frequency count, invalid when single-phase 11: Both single and double phase are invalid

(1) Instruction of Clearing

Users can choose hardware clearing, software clearing and valid clearing level jump (rising edge valid, falling valid); if choosing hardware clearing, channel can be pointed by user. Total clearing when clearing (equal to definition current input point is 0 point).

(2) Description of Relationship between HSC Instruction and Internal Axis Location

- When connecting outer encoder, regarding high-speed pulse input as input axis of encoder, and need to record location of encoder at real time, on the moment of beginning, begin the HSC instruction and operate it all the time.
- There is no relation between the counting value of HSC and parameter of axis location recorded in special register, the two values independent from each other. The special register begins to record location after beginning HSC instruction, the counting direction of special register has relation with counting direction of HSC of current moment, frequency doubling mode only has relation with first HSC frequency doubling parameter, and set or clearing by DEFPOS or HOME instruction. HSC counting value has relation with HSC instruction set.

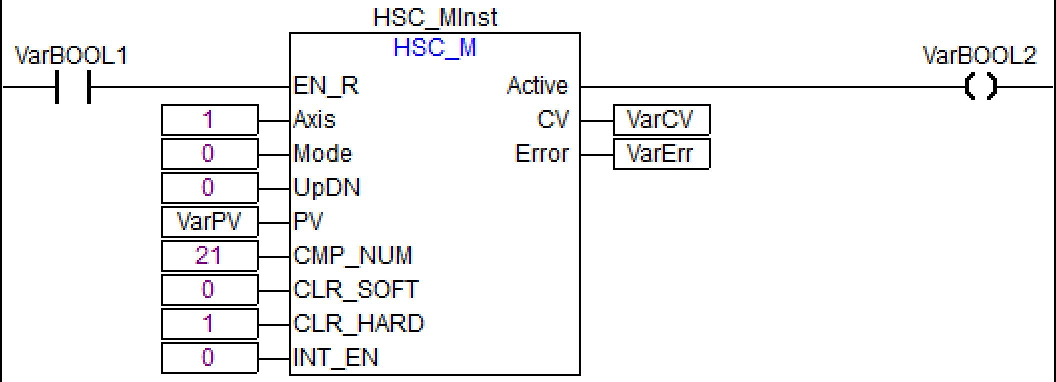
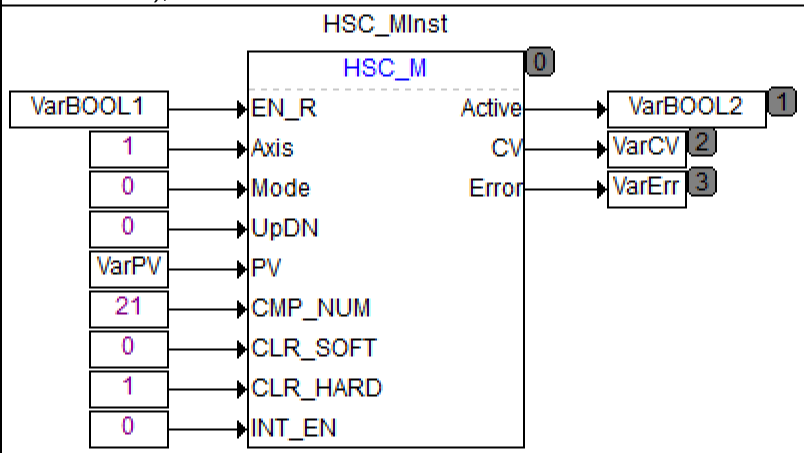


Parameter Relation Schematic

3.5.1.3.3 Application Example of Instructions

Variable Definition

No.	Variable Name	Address	Variable Description	Variable Type	Initial Value	Power Fail Safeguard
0001	HSC_MInst			HSC_M		FALSE
0002	VarBOOL1			BOOL	FALSE	FALSE
0003	VarBOOL2			BOOL	FALSE	FALSE
0004	VarPV			ARRAY[0..20] OF DINT		FALSE
0005	VarCV			DINT	0	FALSE
0006	VarErr			BYTE	0	FALSE

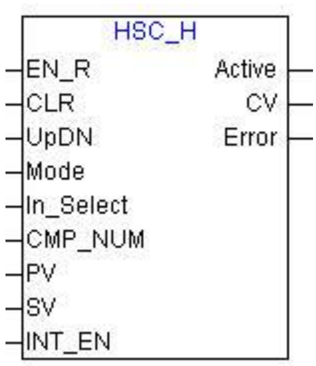
Programing language	Program
Ladder diagram (LD)	
Structured text (ST)	HSC_MInst(EN_R:=VarBOOL1, Axis:=1, Mode:=0, UpDN:=0, PV:=VarPV, CMP_NUM:=21, CLR_SOFT:=0, CLR_HARD:=1, INT_EN:=0, Active=>VarBOOL2, CV=>VarCV, Error=>VarErr);
Continuous function diagram (CFC)	

3.5.1.4 HSC_H (High Speed Counting,5106/5107/5108/5109/5118)

In the different CPU modules, HSC instructions are different.

3.5.1.4.1 Description of function

High-speed counting function block can achieve dual-phase count of A, B or increase or decrease count of single-phase of A, B. Users can set up to 21 different counting target value, and can choose whether to start interruption after reaching the set value. When A and B phase counting, one can receive the phase difference between the range of 90°±60°. The counting range shall be in -2,147,483,648～2,147,483,647.



HSC_H Function Block

3.5.1.4.2 Description of Parameter

Input Parameter	Type of Data	Description of Function	Description of Parameter Value
EN_R	BOOL	Enable	0→1:Rising edge enable, level preserve
CLR	BOOL	Clear	0→1: Rising edge is cleared
UpDN	BOOL	Counting direction control	0: Counting upwards 1: Counting downwards
Mode	BYTE	Choose the mode	See details in Table Detailed_Instruction_of_Mode_Parameter
In_Select	BYTE	Choose input point	See details in Note_1
CMP_NUM	BYTE	The number of Comparing	1~21
PV	ARRAY[21] OF DINT	Set counting value	-2,147,483,648～2,147,483,647 ARRAY[0]～ARRAY[20]
SV	DINT	CV value changed after triggering	-2,147,483,648~2,147,483,647
INT_EN	BOOL	Interruption enable	0: Don't interrupt after reaching count 1: Interrupt after reaching count

Output Parameter	Type of Data	Description of Function	Description of Parameter Value
Active	BOOL	Mark of reaching count	0: Counting doesn't reach set value 1: Counting reaches set value
CV	DINT	Current counting value	-2,147,483,648~2,147,483,647

Output Parameter	Type of Data	Description of Function	Description of Parameter Value
Error	BYTE	Error information	=0: Correct =1: The input channel is taken up =2: Channel choice error =3: Compare number set error =4: SV, PV set error =5: Mode set error

Detailed Instruction of Mode Parameter

Serial No.	Function Overview	Description
Bit7～Bit4	-	Reserved
Bit3	Choice of counting direction	0: Software control, the value of function block parameter UpDN controls counting direction 1: Hardware control, pointing DI channel (I0.4~I0.7, I2.0~I2.3) value to control counting direction
Bit2	Choice of clearing method	0: Rising edge of Reset achieve clearing value 1: Rising edge of according DI channel (I1.0~I1.3, I2.4~I2.7) to achieve counting cleared
Bit1～Bit0	Choice of frequency doubling mode	00: Double (Single) phase counting 1 times frequency 01: Double (Single) phase counting 2 times frequency 10: Double (Single) phase counting 4 times frequency for LE5118 controller, Invalid for single or double phase for other controller 11: Invalid for single or double phase

Note 1: Choice of input channel and corresponding channels instruction

=1: 1st single-phase counting

=2: 2nd single-phase counting

=3: 3rd single-phase counting

=4: 4th single-phase counting

=5: 5th single-phase counting

=6: 6th single-phase counting

=7: 7th single-phase counting

=8: 8th single-phase counting

=21: 1st double-phase counting

=22: 2nd double-phase counting

=23: 3rd double-phase counting

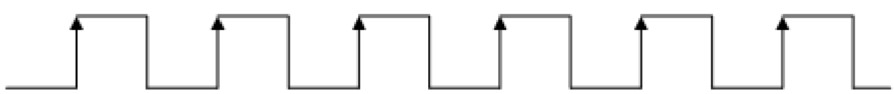
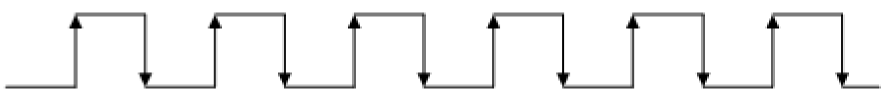
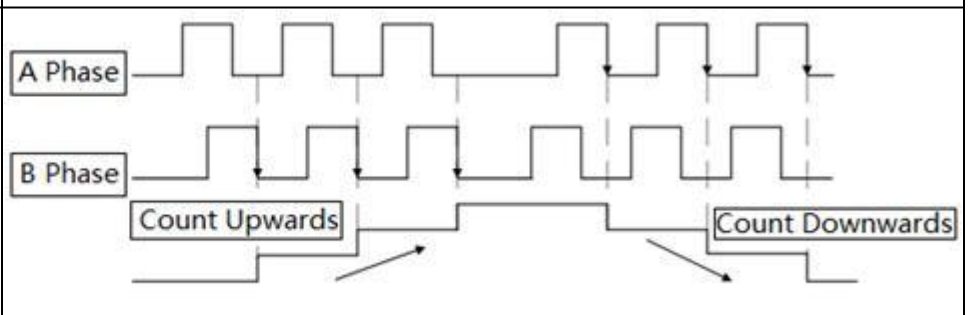
=24: 4th double-phase counting

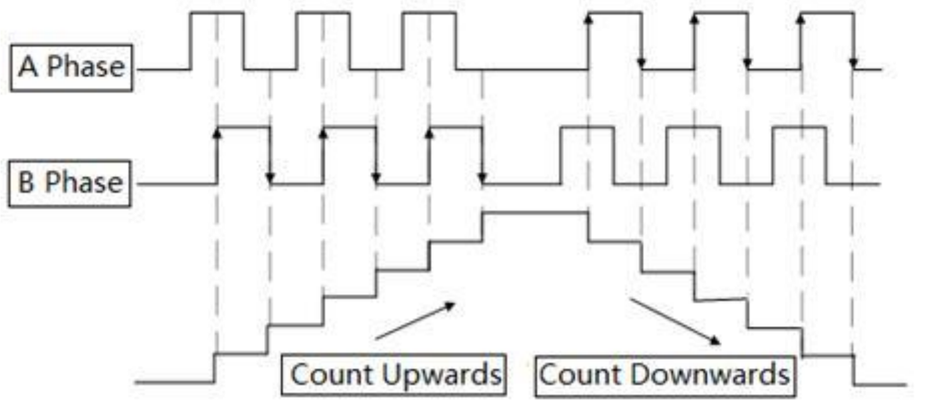
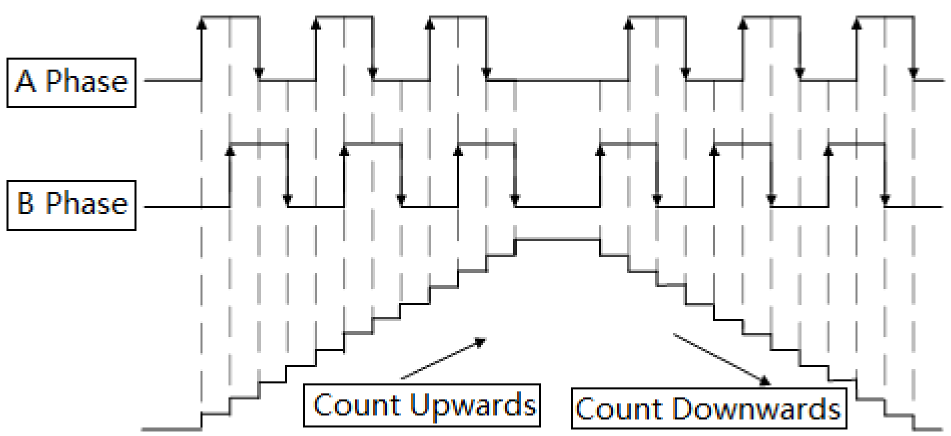
Instruction of Input Channel

High-speed Counting Channel	Single-phase Counting/Double-phase Counting	Corresponding Direction Channel	Direction Control (Only for Single-phase Counter)	Corresponding Clearing Channel	Clearing Control
I0.0	1st way single-phase/1st	I0.4	Direction control	I1.0	1st single-phase

High-speed Counting Channel	Single-phase Counting/Double-phase Counting	Corresponding Direction Channel	Direction Control (Only for Single-phase Counter)	Corresponding Clearing Channel	Clearing Control
	way double-phase A phase		of single-phase of 1st way		clearing/1st way double-phase clearing
I0.1	2nd way single-phase/1st way double-phase B phase	I0.5	Direction control of single-phase of 2nd way	I1.1	2nd way single-phase clearing
I0.2	3rd way single-phase/2nd way double-phase A phase	I0.6	Direction control of single-phase of 3rd way	I1.2	3rd single-phase clearing/2th way double-phase clearing
I0.3	4th way single-phase/2nd way double-phase B phase	I0.7	Direction control of single-phase of 4th way	I1.3	4th way single-phase clearing
I1.4	5th way single-phase/3rd way double-phase A phase	I2.0	Direction control of single-phase of 5th way	I2.4	5th single-phase clearing/3rd way double-phase clearing
I1.5	6th way single-phase/3rd way double-phase B phase	I2.1	Direction control of single-phase of 6th way	I2.5	6th way single-phase clearing
I1.6	7th way single-phase/4th way double-phase A phase	I2.2	Direction control of single-phase of 7th way	I2.6	7th single-phase clearing/4th way double-phase clearing
I1.7	8th way single-phase/4th way double-phase B phase	I2.3	Direction control of single-phase of 8th way	I2.7	8th way single-phase clearing

3.5.1.4.3 High-speed Counting Mode Schematic

Counting Mode		Counting Waveform
Single-phase counting	1 (one time frequency)	
	2 (two times frequency)	
Double-phase counting	1 (one time frequency)	

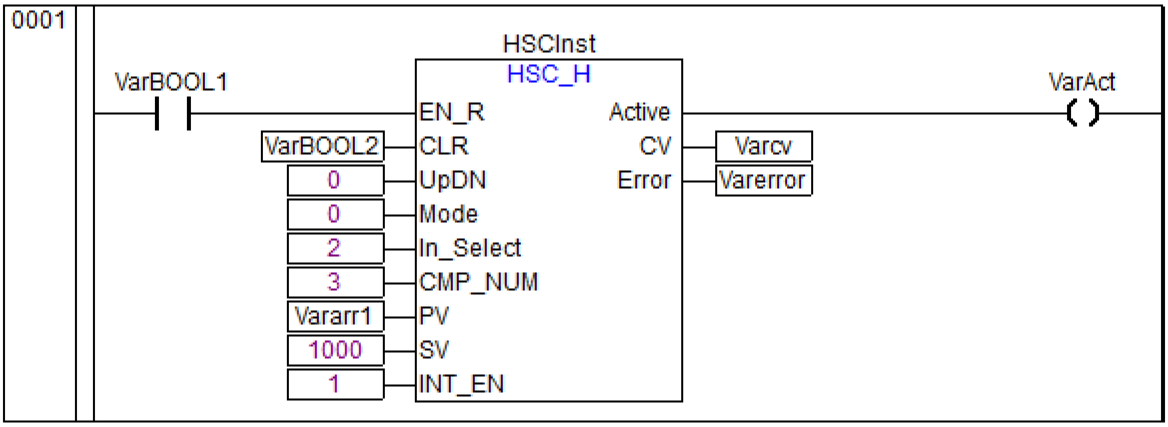
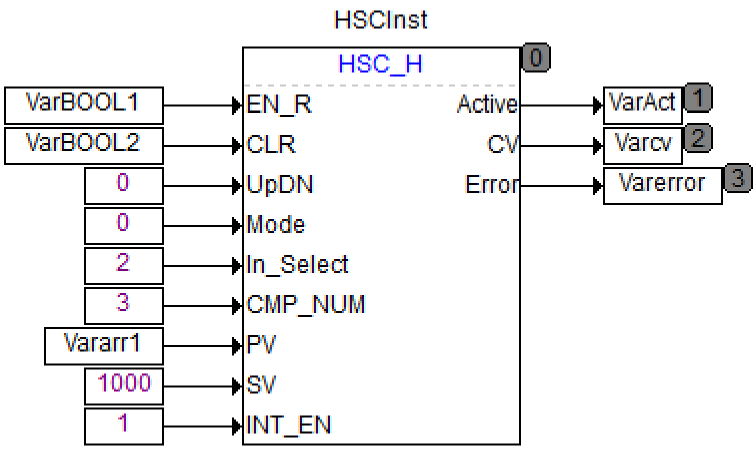
Counting Mode		Counting Waveform
	2 (two times frequency)	
	4 (four times frequency)	

3.5.1.4.4 Application Example of Instructions

Variable Definition

No.	Variable Name	Address	Variable Description	Variable Type	Initial Value	Power Fail Safeguard
0001	HSCInst			HSC_H		FALSE
0002	VarBOOL1			BOOL	FALSE	FALSE
0003	VarBOOL2			BOOL	FALSE	FALSE
0004	Vararr1			ARRAY[0..20] OF DINT		FALSE
0005	VarAct			BOOL	FALSE	FALSE
0006	Varcv			DINT	0	FALSE
0007	Varerror			BYTE	0	FALSE

Programming Language	Program

Variable Definition	
Ladder Diagram （LD）	
Structured Text（ST）	HSCInst(EN_R:=VarBOOL1,CLR:=VarBOOL2,UpDN:=0,Mode:=0,In_Select:=2, CMP_NUM:=3,PV:=Vararr1,SV:=1000,INT_EN:=1,Active=>VarAct,CV=>Varcv,Error=>Varerror);
Continuous Function Chart （CFC）	

3.5.1.4.5 Program Instruction

When EN_R is set and preserved, the second way single-phase counter counting upwards, and producing 3 interruptions during the counting, counting from 1000. When CLR is 1, the counting value reset.

ⓘ

- When high-speed counter enable end is invalid, stop high-speed counting immediately, CV resets when re-touching off the high-speed counter.
- When clearing end CLR changes from 0 to 1, that is rising edge comes, CV value becomes 0, if there is still pulse input, CV value begins to accumulate from 0.
- SV set value is valid after high-speed counting enable, if set value improper, corresponding value of error refer to the instruction of Error.
- Counting direction control function end UpDN only makes sense when one-phase counting.
- When still counting upwards if CV values equal to 2,147,483,647, CV value becomes -2,147,483,648,

similarly, when continuing to count downwards even high-speed counting instruction current CV value equals to -2,147,483,648, CV value becomes 2,147,483,647.

- One-phase counting channel supports the highest counting frequency is 100,000 Hz, double-phase counting channel supports the highest counting frequency is 50,000 Hz. Set the data in counting value PV data shall meet the condition: when counting upwards, PV[i+1]>PV[i]; when counting downwards, PV[i+1]<PV[i], or can't count normally.

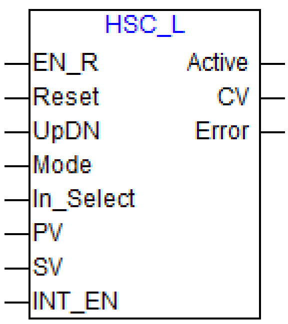
- Each one-phase high-speed counter takes up a high-speed input channel, double-phase (A/B) high-speed count takes up two high-speed input channels. If using double-phase high speed counter, then the two corresponding high-speed channel one-phase counter can't work, and the process value, target value and counting mark of double-phase (A/B) high-speed count shall be saved in special register of corresponding first high-speed input channel.

3.5.1.5 HSC_L(High Speed Counting,5104/5105/5107E/5107L/5109L)

In different CPU modes, HSC instructions are different.

3.5.1.5.1 Description of Function

High-speed counting function block can achieve duplex count of A, B or increase or decrease count of one-edge of A, B. When the counting value reaches set value, then can choose whether to start interruption. When A and B phase counting, one can receive the difference of pulse input count between the range of 90°±60°.



HSC_L Function Block

3.5.1.5.2 Description of Parameter

Input Parameter	Type of Data	Description of Function	Instruction of Parameter Value
EN_R	BOOL	Enable	0→1: Rising edge enable, level preserved
Reset	BOOL	Clearing	0→1: Rising edge clearing
UpDN	BOOL	Counting direction	0: Count upward

Input Parameter	Type of Data	Description of Function	Instruction of Parameter Value
		control	1: Count downward
In_Select	BYTE	Choice of input point	=1: I0.0 is pulse input end (double-phase count A phase), I0.2 is clearing end, I0.4 is counting direction control (double-phase count B phase) =2: I0.1 is pulse input end, I0.3 is clearing end, I0.5 is counting direction control
Mode	BYTE	Choice of mode	See details in Table
PV	DINT	Set counting value	-2,147,483,648~2,147,483,647
SV	DINT	Changing value after triggering	-2,147,483,648~2,147,483,647
INT_EN	BOOL	Interruption enable	0: Doesn't interrupt after reaching count 1: Interrupt after reaching count

Output Parameter	Type of Data	Description of Function	Instruction of Parameter Value
Active	BOOL	Mark of reaching the count	0: Count doesn't reach set value 1: Count reaches set value
CV	DINT	Current counting value	-2,147,483,648~2,147,483,647
Error	BYTE	Error information	=0: Correct =1: In_Select parameter configuration error =2: Mode parameter configuration error =3: Related parameter of hardware =4: Related parameter of hardware =5: Related parameter of hardware =6: Related parameter of hardware =7: Related parameter of hardware =8: Resources are occupied (for example: DI channels have been configured as pulse catch or fast external interruption)

Detailed Instruction of Mode Parameter

Serial No.	Overview of Function	Description
Bit7～Bit5	-	Reserve
Bit4	Choice of counting direction	0: Software control, the value of function block parameter UpDN controls counting direction 1: Hardware control, pointing DI channel (I0.4, I0.5) value to control counting direction.
Bit3	Choice of clearing method	0: Rising edge of Reset achieve value clearing 1: Rising edge of corresponding DI channel (I0.2,I0.3) to achieve counting value clearing
Bit2～Bit1	Choice of frequency doubling mode	00: Single-phase counting 1 time frequency, double-phase counting 2 times frequency 01: Single-phase counting 2 times frequency, double-phase counting 4 times frequency 10, 11: Both single and double phase are invalid.
Bit0	Choice of one or double phase counting	0: Single-phase counting 1: Double -phase counting

3.5.1.5.3 Description of Instruction

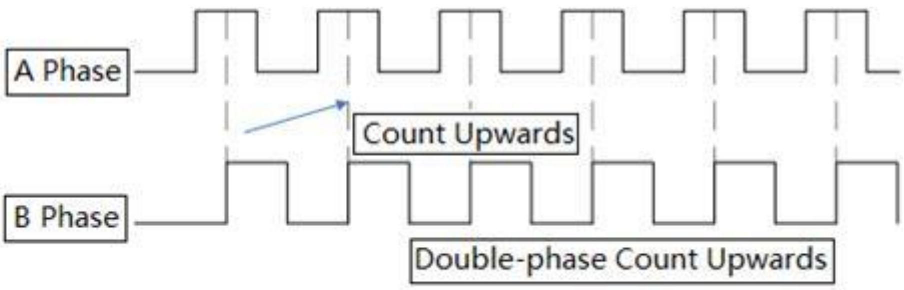
Clearing end Reset: when Reset becomes from False to True, CV value becomes 0, if there is still pulse input, CV value continues to accumulate from 0, clearing end can be outer input point or variable.

Input point choosing function end In_Select: In_Select has 2 values. When In_Select=1, one can choose single-phase or double phase count through setting bit0 of mode; when In_Select=2, there shall only be single-phase count, the bit0 of mode only can be 0.

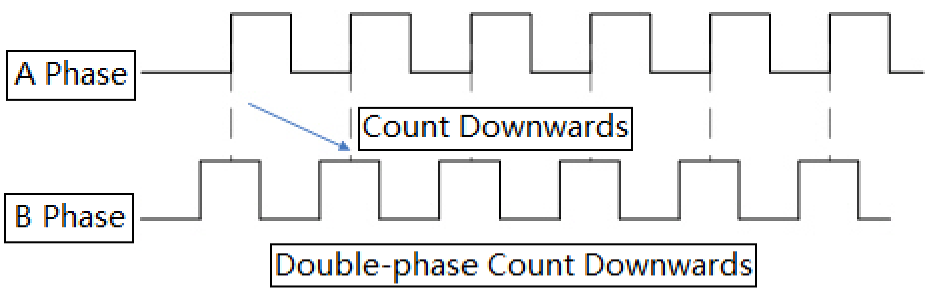
Variable set	**0**	**Upward count**
	1	Downward count
Outer input point	Low level	Upward count
	High level	Downward count

When single-phase counting, the counting direction shall be decided by UpDN input variable or outer input point. When the UpDN is not bind with outer input point (that is bit4 of mode =0), the counting direction shall be decided by UpDN end input variable; When UpDN chooses to bind with outer input point (that is bit4 of mode =1), counting direction shall be decided by the high and low level of binding input end.

When double -phase counting, UpDN function end has no sense, the counting direction shall be decided by A-phase and B-phase imput. When A-phase is over B-phase, counting direction is upwards counting, when B-phase is over A-phase, counting direction is downwards counting.



(a)

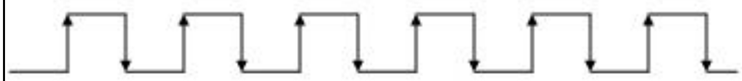
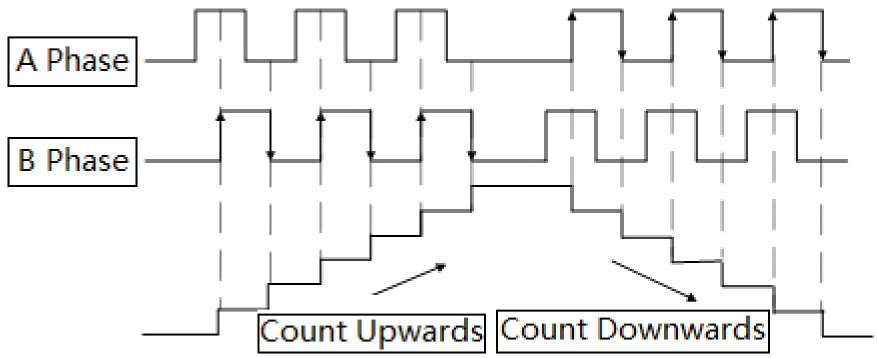
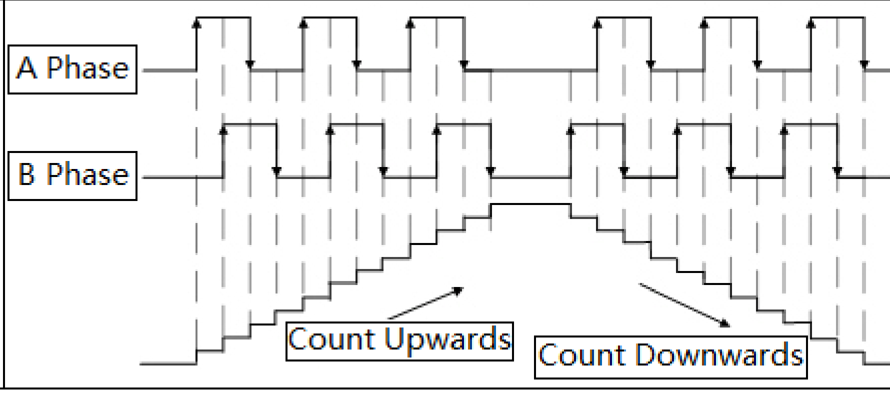


(b)

Double-direction Counting Schematic

Frequency multiplication function end Mode counting as shown in follows table(↑↓ shows the appearance of counting conduct).

Counting Mode Schematic

Counting Mode		Counting Conduct Waveform
Single-phase count	One time frequency	
	Two times frequency	
Double-phase count	Two times frequency	
	Four times frequency	

When still counting upwards if CV values equal to 2,147,483,647, CV value becomes -2,147,483,648; similarly, when still counting downwards if high-speed counting instruction current CV value equals to -2,147,483,648, CV value becomes 2,147,483,647.

High-speed counter is triggered by level, when the front energy line is broken, stop high-speed count and reset high-speed counter, CV clears zero when re-triggering the high-speed counter.

3.5.1.5.4 Special Instruction

Each single-phase high-speed counter takes up a high-speed input channel, and each double-phase high-speed count takes up two high-speed input channels. If using double-phase high speed counter, then the two corresponding high-speed channel single-phase counter can’t work.

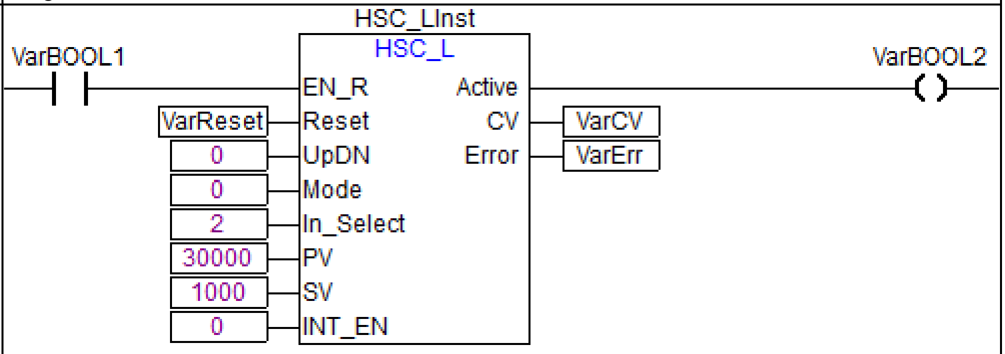
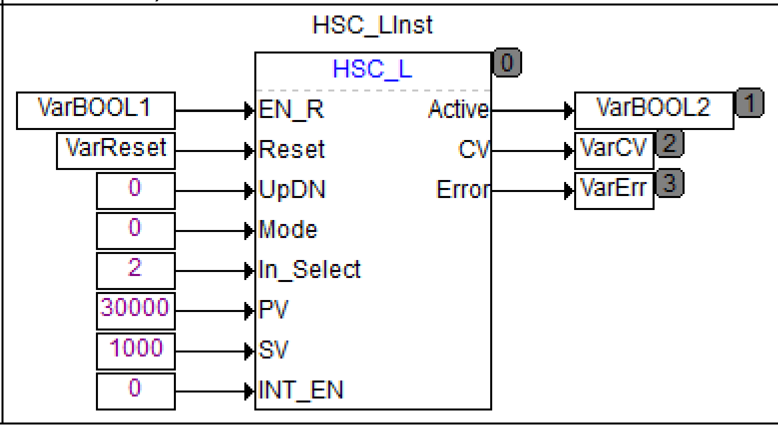
If one channel is taken up by HSC function block, then other HSC function block can’t use the channel, or there will be some counting errors.

HSC function block and channel filtering parameter (configuration in AutoThink interface) takes up much CPU hardware resources; and will make heavy burden when they are opened together, and there may be counting errors, so do not enable them together during the use.

3.5.1.5.5 Application Example of Instructions

Variable Definition

No.	Variable Name	Address	Variable Description	Variable Type	Initial Value	Power Fail Safeguard
0001	HSC_LInst			HSC_L		FALSE
0002	VarBOOL1			BOOL	FALSE	FALSE
0003	VarBOOL2			BOOL	FALSE	FALSE
0004	VarReset			BOOL	FALSE	FALSE
0005	VarCV			DINT	0	FALSE
0006	VarErr			BYTE	0	FALSE

<table>
<tr><td>Programing language</td><td>Program</td></tr>
<tr><td>Ladder diagram (LD)</td><td>
</td></tr>
<tr><td>Structured text (ST)</td><td>HSC_LInst(
EN_R:=VarBOOL1,
Reset:=VarReset,
UpDN:=0,
Mode:=0,
In_Select:=2,
PV:=30000,
SV:=1000,
INT_EN:=0,
Active=> VarBOOL2,
CV=>VarCV,
Error=>VarErr);</td></tr>
<tr><td>Continuous function diagram (CFC)</td><td>
</td></tr>
</table>

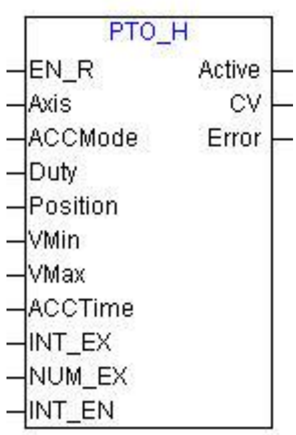
Program description:Program Instruction:

When EN_R is set and held, the 2nd single-phase counter counts up to 30000 pulses and the initial count starts at 1000. When Reset is 1, the count value is reset.

3.5.1.6 PTO_H (PTO Output, 5106/5108/5118)

3.5.1.6.1 Description of Function

Output the pulse with frequency pointed by VMin, VMax, ACCTime, duty ratio pointed by Duty and number set by Position from the port pointed by Axis.



PTO_H Function Block

3.5.1.6.2 Parameter Description

Input Parameter	Type of Data	Description of Function	Instruction of Parameter Value
EN_R	BOOL	Enable	0→1:Rising edge enable, high level keep
Axis	BYTE	Output channel selection	=1: Input axis 1, output pulse end Q0.0 =2: Input axis 2, output pulse end Q0.1 =3: Input axis 3, output pulse end Q0.2 =4: Input axis 4, output pulse end Q0.3
ACCMode	BOOL	Acceleration and deceleration mode selection	0: Trapezoidal curve 1: S curve
Duty	BOOL	Duty ratio	0: 50% 1: 25%
Position	DWORD	Number of output pulse set	1～4,294,967,295
VMin	DWORD	Preliminary frequency	10 Hz～100,000 Hz
VMax	DWORD	Highest frequency	10 Hz～100,000 Hz
ACCTime	WORD	Acceleration and deceleration time	0～32,767 ms (0 means average speed)
INT_EX	BYTE	Interruption input point selection	See details in Note
NUM_EX	DWORD	Re-send number after interruption input	0～4,294,967,295
INT_EN	BOOL	Interruption enable	0: Doesn’t interrupt after finishing output of pulse 1: Interruption occurs after finishing output of pulse

Output Parameter	Type of Data	Description of Function	Instruction of Parameter Value
Active	BOOL	Mark of sending pulse	0: Stop sending 1: On the process of sending
CV	DWORD	Number of sent pulse	1～4,294,967,295
Error	BYTE	Error information	=0: Correct =1: The channel is taken up =2: Configuration output axis error =3: Frequency output error

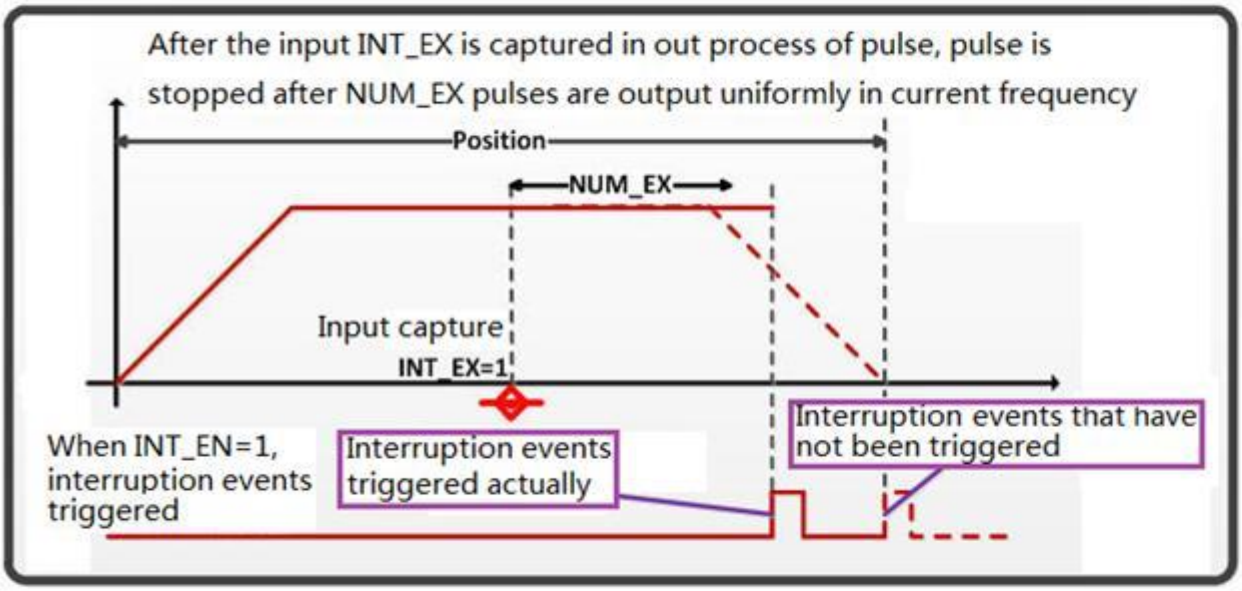
Note: instruction of interruption input of channels

- Trigger selection: Outer input point.
- Trigger condition: Rising edge trigger, high level doesn't keep.
- INT_EX parameter corresponding meaning: 0: not choose; 1~24: I0.0～I2.7.
- Description of process: The current enable status only can support one valid trigger, The output immediately executes the number of pulses specified by NUM_EX when the rising edge is triggered.
- Interruption enable: If interruption enable (NT_EN=1), then respond the corresponding interruption events after finishing re-sending number of pulse.

3.5.1.6.3 Special Instruction

When one of the following situations occurs, output the pulse with frequency pointed by VMax, Duty ratio pointed by Duty and number set by Position at the same speed.

- Acceleration and deceleration time ACCTime is 0.
- Preliminary frequency VMin are equal to the highest frequency VMax.
- Number of output pulse Position is less than 6.
- The output change of PTO_H function will not change the status of special register value.

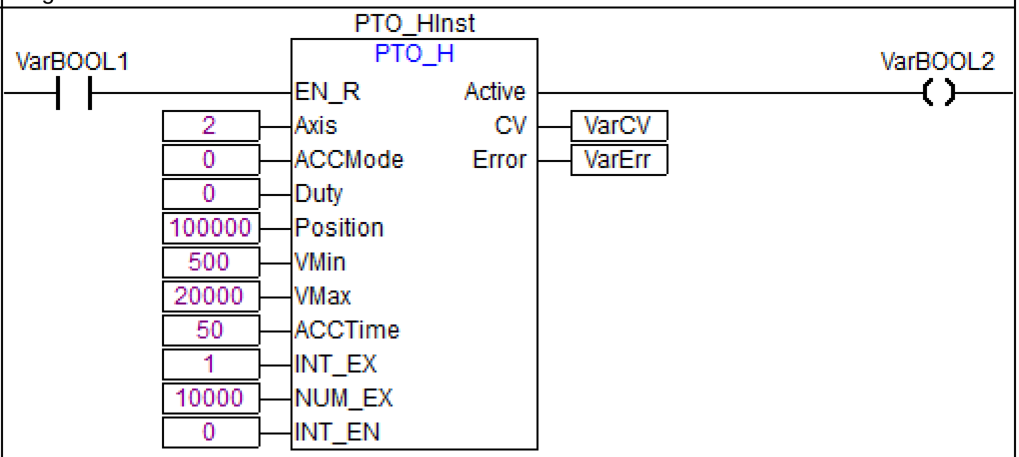


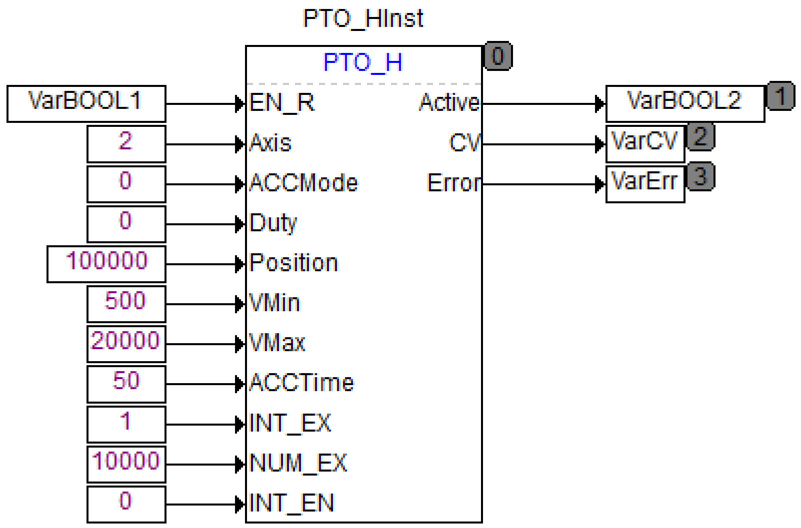
Process of Pulse Output

3.5.1.6.4 Application Example of Instructions

Variable Definition

No.	Variable Name	Address	Variable Description	Variable Type	Initial Value	Power Fail Safeguard
0001	PTO_HInst			PTO_H		FALSE
0002	VarBOOL1			BOOL	FALSE	FALSE
0003	VarBOOL2			BOOL	FALSE	FALSE
0004	VarCV			DWORD	0	FALSE
0005	VarErr			BYTE	0	FALSE

Programing language	Program
Ladder diagram (LD)	
Structured text (ST)	PTO_HInst(EN_R:=VarBOOL1, Axis:=2, ACCMode:=0, Duty:=0,

Variable Definition	
	Position:=100000, VMin:=500, VMax:=20000, ACCTime:=50, INT_EX:=1, NUM_EX:=10000, INT_EN:=0, Active=> VarBOOL2, CV=> VarCV, Error=> VarErr);
Continuous function diagram (CFC)	

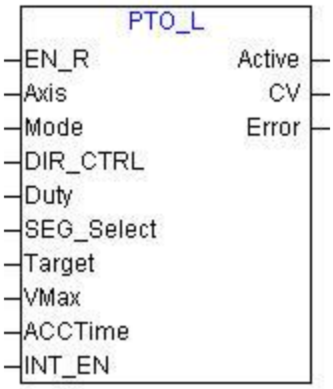
Program Instruction:

When EN_R is set and held, the axis 2 (Q0.1) sends 100000 pulses with a duty cycle of 50, accelerates from 500 Hz to 20000 Hz in the trapezoidal acceleration mode, and the acceleration time is 50 ms. If I0.0 inputs pulse, generate an interrupt, and then send 10,000 pulses, the end. CV shows the number of pulses sent.

3.5.1.7 PTO_L (PTO Output, 5104)

3.5.1.7.1 Description of Function

After setting number of output pulse and frequency, output pulse whose number and frequency has been set by specific switching output point.



PTO_L Function Block

3.5.1.7.2 Description of Parameter

Input Parameter	Type of Data	Description of Function	Instruction of Parameter Value
EN_R	BOOL	Enable	0→1 Rising edge enable, level keep
Axis	BYTE	Output axis selection	=1: Axis 1 output pulse (including direction axis) =2: Axis 2 output pulse (including direction axis)
Mode	BYTE	Action mode	Action mode and output logistic relation 1: Type of increment PLS+SIGN (forward OFF/backward ON) 2: Type of increment PLS+SIGN (forward ON/backward OFF)
DIR_CTRL	BOOL	Whether output direction is associated with input end configuration	0: Not associated 1: Associated
Duty	BOOL	Duty ratio	0: 50% 1: 25%
SEG_Select	BYTE	Segments selection	SEG_SELECT=0: 10 segments 1: 30 segments 2: 60 segments 3: 100 segments
Target	DINT	Target location set	-2,147,483,648~2,147,483,647, the least absolute value of number of pulse shall be not less than 20
VMax	DWORD	The highest frequency	10 Hz~20,000 Hz
ACCTime	WORD	Acceleration and deceleration time	0~32,767 ms (0 shows uniform)
INT_EN	BOOL	Interruption enable	0: Doesn't interrupt after finishing output of pulse 1: Interruption occurs after finishing output of pulse

Output Parameter	Type of Data	Description of Function	Instruction of Parameter Value
Active	BOOL	Mark of sending pulse	0: Stop sending (finish sending pulse) 1: On the process of sending
CV	DINT	Number of sent pulse	-2,147,483,648~2,147,483,647
Error	BYTE	Error information	=0: Correct =1: The channel is taken up by PWM, directly quit function block =2: Configuration output axis is error, directly quit function block

Output Parameter	Type of Data	Description of Function	Instruction of Parameter Value
			=3: Configuration action mode is error, directly quit function block =4: Configuration acceleration segment is error, directly quit function block =5: Configuration output pulse number is error, directly quit function block =6: Configuration output frequency is error, directly quit function block =7: Configuration acceleration and deceleration time is error, directly quit function block

ⓘThe number of segments indicates the number of frequency jumps that theoretically need to pass when the output frequency is accelerated to the set frequency value. Selecting a larger number of segments will make the acceleration and deceleration process smoother, but may cause the acceleration time error to be larger.

3.5.1.7.3 Description of Instruction

- The instruction of pulse output control has direction end, there are 2 motion control modes; description: when enable signal becomes ON from OFF, output pulse from pointed channel.

Output Axis Selection	Output Channel of Pulse	Pulse Output Direction Channel
1	Q0.0	Q0.2
2	Q0.1	Q0.3

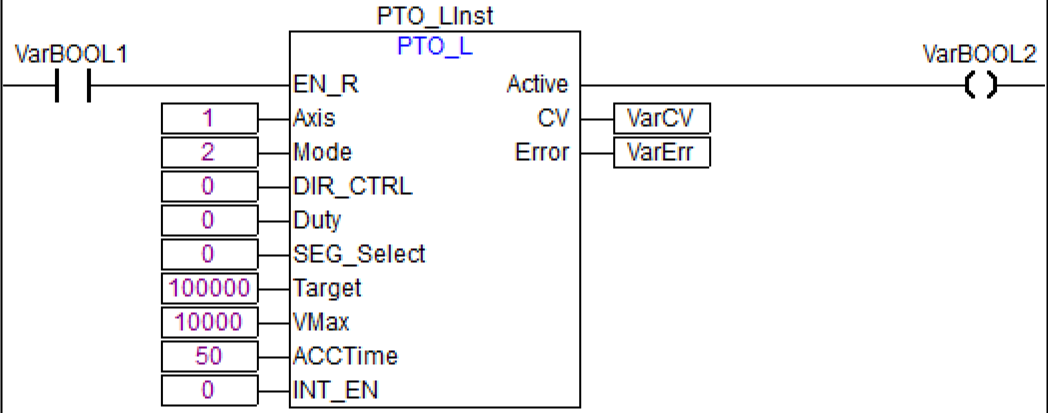
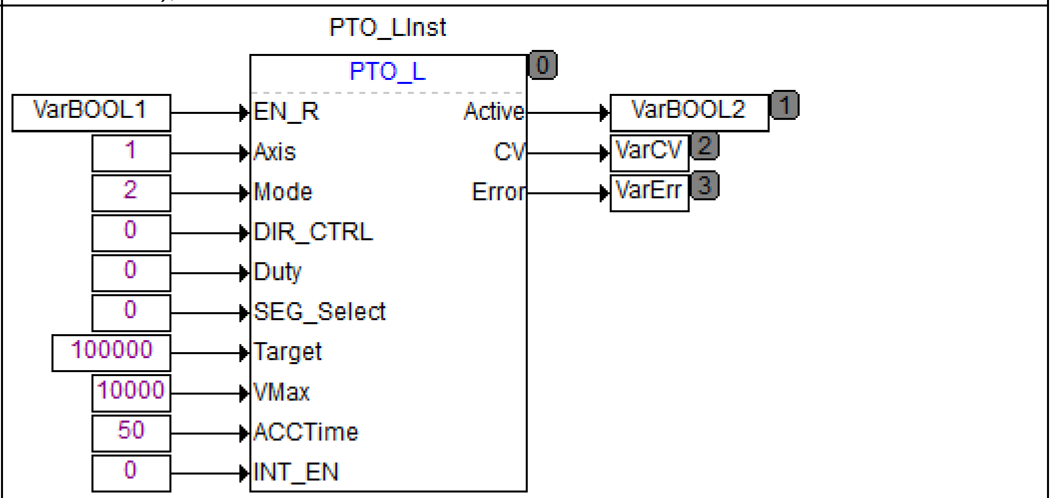
- Increment type of action mode and sending method of pulse; increment type relative value control the output of pulse with number set by target value.

Selection Method Target Value	PLS+SIGN Forward : OFF Backward: ON	PLS+SIGN Forward : ON Backward: OFF	Process Value
Positive number	Pulse input: ON Direction output : OFF	Pulse input: ON Direction output : ON	Progressive increase
Negative number	Pulse input: ON Direction output : ON	Pulse input: ON Direction output : OFF	Progressive decrease

3.5.1.7.4 Application Example of Instructions

Variable Definition

No.	Variable Name	Address	Variable Description	Variable Type	Initial Value	Power Fail Safeguard
0001	PTO_LInst			PTO_L		FALSE
0002	VarBOOL1			BOOL	FALSE	FALSE
0003	VarBOOL2			BOOL	FALSE	FALSE
0004	VarCV			DINT	0	FALSE
0005	VarErr			BYTE	0	FALSE

<table>
<tr><th colspan="2">Variable Definition</th></tr>
<tr><td>Programing language</td><td>Program</td></tr>
<tr><td>Ladder diagram (LD)</td><td>
</td></tr>
<tr><td>Structured text (ST)</td><td>PTO_LInst(
EN_R:=VarBOOL1,
Axis:=1,
Mode:=2,
DIR_CTRL:=0,
Duty:=0,
SEG_Select:=0,
Target:=100000,
VMax:=10000,
ACCTime:=50,
INT_EN:=0,
Active=> VarBOOL2,
CV=> VarCV,
Error=> VarErr);</td></tr>
<tr><td>Continuous function diagram (CFC)</td><td>
</td></tr>
</table>

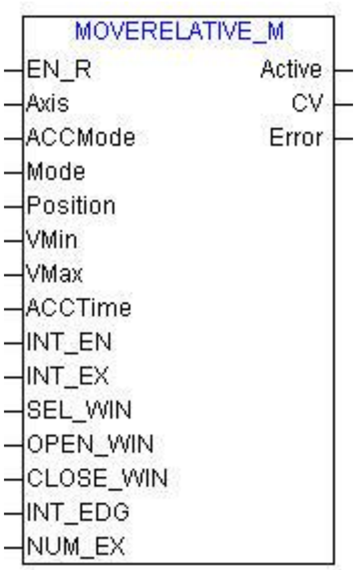
Program Instruction:

When EN_R is set and held, axis 1 (Q0.0) sends a pulse with a duty cycle of 50 (Q0.2 for controlling the rotation direction of the motor) in increments of PLS + SIGN (forward ON / reverse OFF). Send a total of 100,000, the maximum output frequency is 10000Hz, acceleration and deceleration time is 50ms, CV shows the number of pulses sent.

3.5.1.8 MOVERELATIVE_M (Relative Move, 5128)

3.5.1.8.1 Description of Function

Used for controlling electric machine for positioning, taking original point as reference point to output pulse whose number and frequency has been set to reach positioning function. The article doesn't support to modify MOVERELATIVE parameter value during operation and doesn't support triggering the matching interruption of process value in process during output of pulse.



MOVERELATIVE_M Function Block

3.5.1.8.2 Description of Parameter

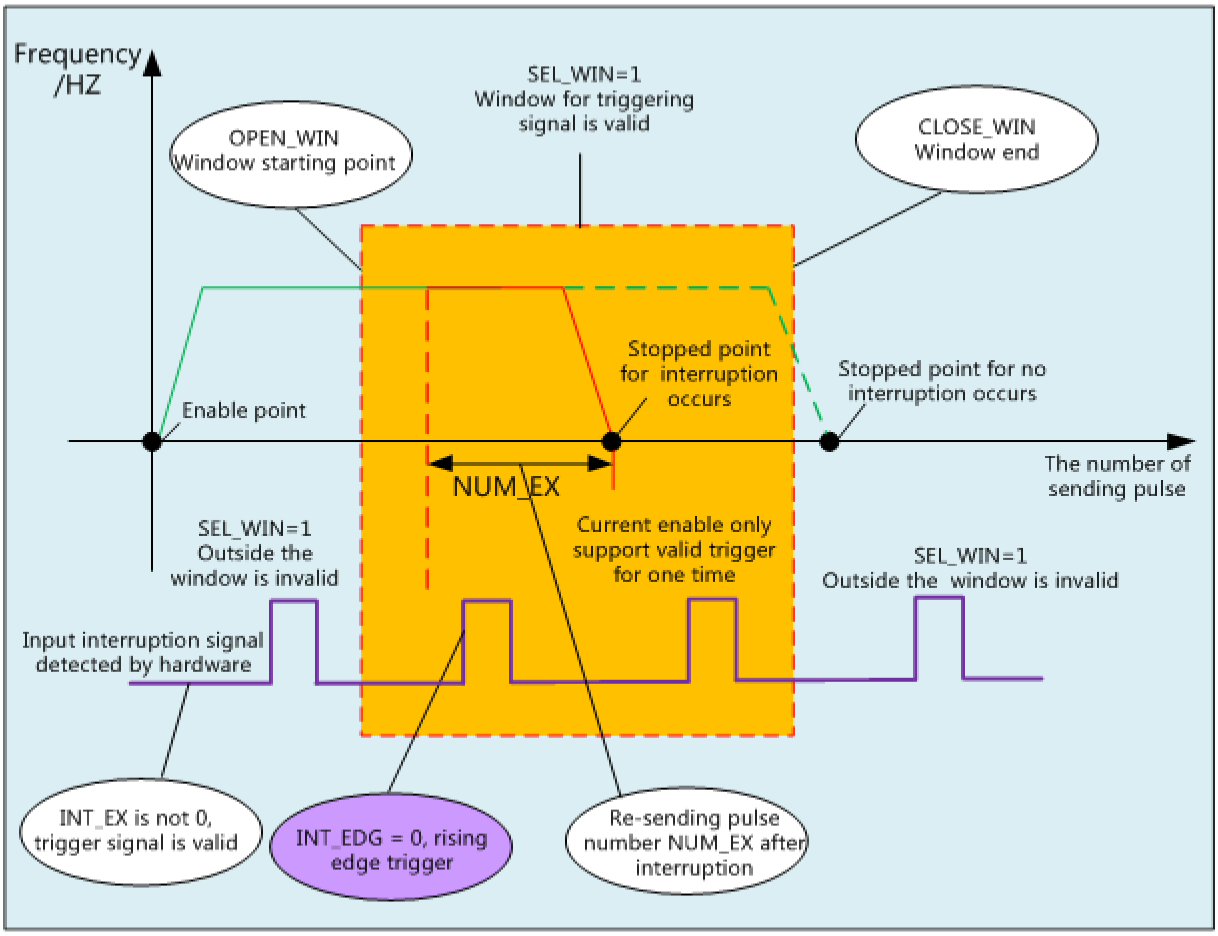
Input Parameter	**Type of Data**	**Description of Function**	**Instruction of Parameter Value**
EN_R	BOOL	Enable	0→1 Rising edge enable, high level keep
Axis	BYTE	Output axis selection	PLS+SIGN mode =5 : Output axis 5, pulse output end Q0.0, pulse direction Q0.4 =6: Output axis 6, pulse output end Q0.1, pulse direction Q0.5 =7: Output axis 7, pulse output end Q0.2, pulse direction Q0.6 =8: Output axis 8, pulse output end Q0.3, pulse direction Q0.7 CW/CCW mode =5, 7 Separately show output is : axis5, axis 7 If output 6 response to axis 5 output, if output 8 responses to axis 7 output
ACCMode	BOOL	Acceleration and deceleration mode selection	0: Trapezoidal curve 1: S curve
Mode	BOOL	Pulse mode control	0: PLS+SIGN mode 1: CW/CCW mode
Position	DINT	Relative position set	-1,073,741,824～1,073,741,823

<table>
<tr><th>Input Parameter</th><th>Type of Data</th><th>Description of Function</th><th colspan="2">Instruction of Parameter Value</th></tr>
<tr><td>VMin</td><td>DWORD</td><td>Preliminary frequency</td><td colspan="2">10 Hz~100,000 Hz</td></tr>
<tr><td>VMax</td><td>DWORD</td><td>The highest frequency</td><td colspan="2">10 Hz~100,000 Hz</td></tr>
<tr><td>ACCTime</td><td>WORD</td><td>Acceleration and deceleration time</td><td colspan="2">0~32,767 ms (0 shows uniform)</td></tr>
<tr><td>INT_EN</td><td>BOOL</td><td>Whether Interruption enable after finishing output</td><td colspan="2">0: Doesn’t interrupt after finishing output of pulse
1: Interruption occurs after finishing output of pulse</td></tr>
<tr><td>INT_EX</td><td>BYTE</td><td>Interruption input point selection during output process</td><td colspan="2">0～16 refer to Interruption_Input_Instruction_of_Various_Axis_INT_EX , if the parameter is 0, then the follows parameters related with interruption are invalid.</td></tr>
<tr><td>SEL_WIN</td><td>BYTE</td><td>Window selection</td><td>Constraint the validity of parameter of INT_EX hardware
0 : Not use window
1 : Valid within window
2 : Valid out of window</td><td rowspan="5">When INT_EX=0, the array is valid
See the function schematic in operation examples.</td></tr>
<tr><td>OPEN_WIN</td><td>DINT</td><td>Relative value of window 1</td><td>-1,073,741,824～1,073,741,823</td></tr>
<tr><td>CLOSE_WIN</td><td>DINT</td><td>Relative value of window 2</td><td>-1,073,741,824～1,073,741,823</td></tr>
<tr><td>INT_EDG</td><td>BOOL</td><td>INT_EX interruption jump edge selection</td><td>0: Rising edge trigger
1: Falling edge trigger</td></tr>
<tr><td>NUM_EX</td><td>DWORD</td><td>Send number after stopping input</td><td>0～2,147,483,647</td></tr>
</table>

Output Parameter	Type of Data	Description of Function	Instruction of Parameter Value
Active	BOOL	Mark of sending pulse	0 : Stop sending 1 : On the process of sending
CV	DINT	Number of sent pulse	-1,073,741,824～1,073,741,823
Error	BYTE	Error information	=0: Correct =1: The channel is taken up =2: Configuration axis is error =3: Frequency output is error =4: Process interruption input point selection is error =5: Setting parameter for relative position is out of range =6: Acceleration and deceleration time parameter setting error

3.5.1.8.3 Operating Examples

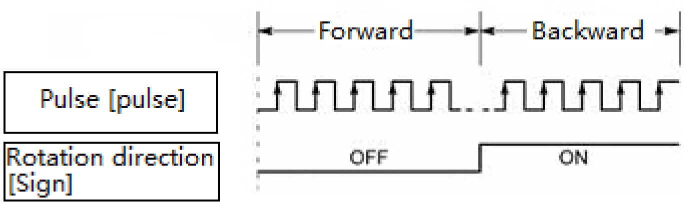
Choosing external interruption to trigger during the process of output, the schematic of parameter related to INT_EX instruction is as shown in figure.



Schematic Diagram of Parameter Related to INT_EX Instruction

3.5.1.8.4 Explanation

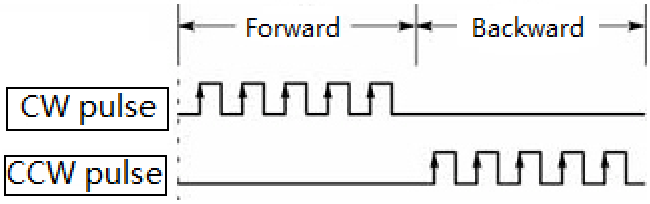
1. PLS+SIGN output mode



PLS+SIGN Output Mode

Speed points pulse to output, rotation direction point use ON/OFF to point, under the mode, rotation direction OFF response to forward of electric machine.

2. CW/CCW output mode



CW/CCW Output Mode

Under the effect of the method, forward and backward shall be controlled by double pulse of pulse.

3.5.1.8.5 Example

Current position is 5000, as target value +7000, operating pulse output instruction;

Pulse + direction: output 7000 pulse from pulse output axis, direction output low level.

CW/CCW: CW axis output 7000 pulse, CCW output low level.

The current location is 12000 after the above two methods.

3.5.1.8.6 Interruption Input Instruction of Various Axis INT_EX

- Trigger selection: hardware input trigger, trigger point pointed by users. 0: not choose; 1~16: I0.0~I1.7.
- Trigger condition: rising edge or falling edge trigger are available, high level doesn’t keep.
- Process description: current enable status only can support one-time valid trigger, and still output number of pulse pointed by NUM_EX according to set parameter.

3.5.1.8.7 Special instruction

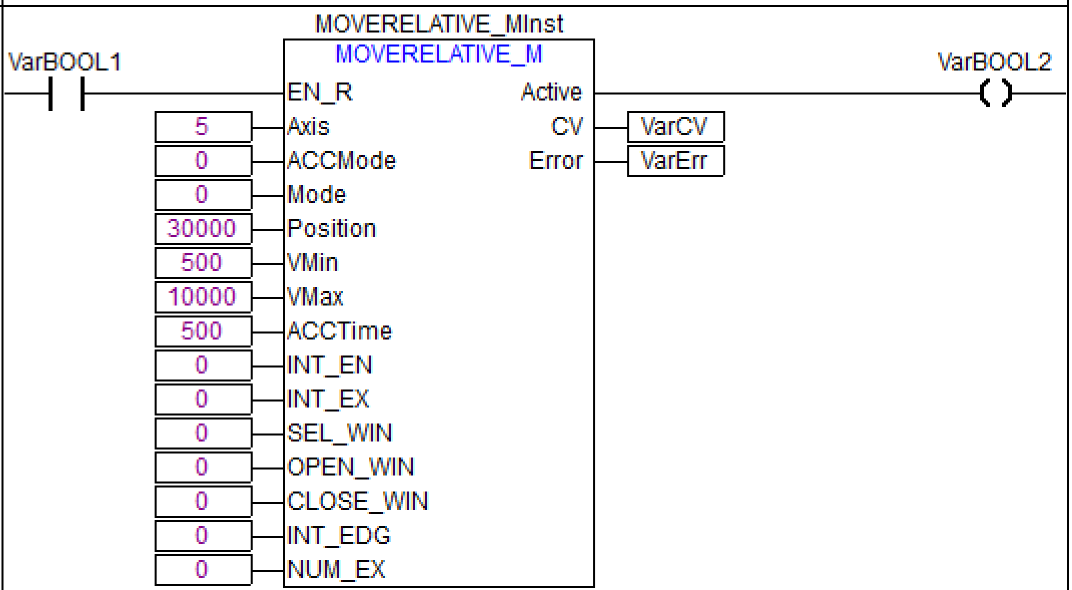
Output pointed number of pulse at same speed by VMax when there is one of following cases.

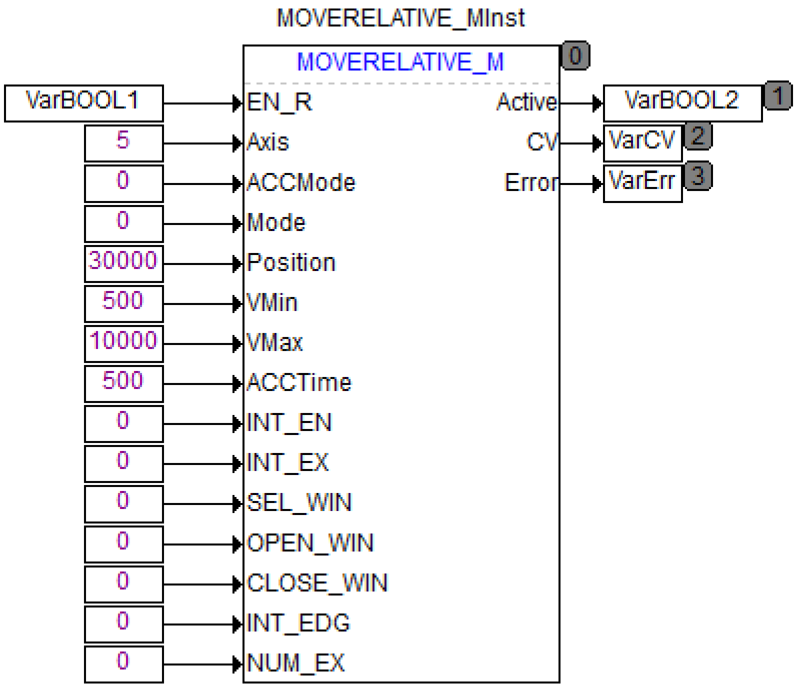
- Acceleration and deceleration time ACCTime is 0.
- Preliminary frequency VMin equal to the highest frequency VMax.
- Number of output pulse Position is less than 6.

3.5.1.8.8 Application Example of Instructions

Variable Definition

No.	Variable Name	Address	Variable Description	Variable Type	Initial Value	Power Fail Safeguard
0001	MOVERELATIVE_MInst			MOVERELATIVE_M		FALSE
0002	VarBOOL1			BOOL	FALSE	FALSE
0003	VarBOOL2			BOOL	FALSE	FALSE
0004	VarCV			DINT	0	FALSE
0005	VarErr			BYTE	0	FALSE

Programing language	Program
Ladder diagram (LD)	
Structured text (ST)	MOVERELATIVE_MInst(EN_R:=VarBOOL1, Axis:=5, ACCMode:=0, Mode:=0, Position:=30000, VMin:=500, VMax:=10000, ACCTime:=500, INT_EN:=0, INT_EX:=0, SEL_WIN:=0, OPEN_WIN:=0, CLOSE_WIN:=0, INT_EDG:=0, NUM_EX:=0, Active=> VarBOOL2, CV=> VarCV, Error=> VarErr);

<table>
<tr><th colspan="2">Variable Definition</th></tr>
<tr><td>Continuous function diagram (CFC)</td><td>
</td></tr>
</table>

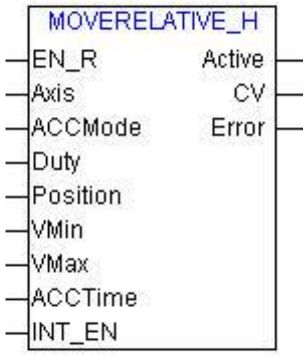
Program Instruction:

When the EN_R is set and held, the axes 5 (Q0.0 and Q0.4) send a pulse with a duty cycle of 50 in the double pulse mode, sending a total of 30,000, the initial frequency is 500 Hz, the maximum output frequency is 10000 Hz, acceleration and deceleration time is 500ms, trapezoidal acceleration / deceleration mode, CV shows the number of sent pulses, no interrupt generated.

3.5.1.9 MOVERELATIVE_H (Relative Move, 5106/5108/5118)

3.5.1.9.1 Description of Function

Used for controlling electric machine for positioning, taking original point as reference point to output pulse whose number and frequency has been set to reach positioning function.



MOVERELATIVE_H Function Block

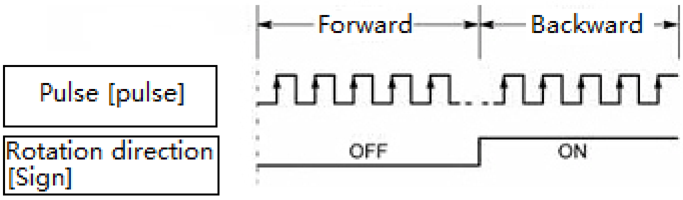
3.5.1.9.2 Description of Parameter

Input Parameter	Type of Data	Description of Function	Instruction of Parameter Value
EN_R	BOOL	Enable	0→1 Rising edge enable, high level keep
Axis	BYTE	Output channel selection	Pulse + direction mode 1 : Output axis 1, pulse output end Q0.0, pulse direction Q0.4 2 : Output axis 2, pulse output end Q0.1, pulse direction Q0.5 3 : Output axis 3, pulse output end Q0.2, pulse direction Q0.6 4 : Output axis 4, pulse output end Q0.3, pulse direction Q0.7 CW/CCW mode 5 : Output axis 1, pulse output end (CW) Q0.0, pulse output end (CCW) Q0.1 6 : Output axis 2, pulse output end (CW) Q0.2, pulse output end (CCW) Q0.3
ACCMode	BOOL	Acceleration mode selection	0: Trapezoidal curve 1: S curve
Duty	BOOL	Duty ratio	0: 50% 1: 25%
Position	DINT	Number of output of pulse set	-2,147,483,648~2,147,483,647
VMin	DWORD	Preliminary frequency	10 Hz~100,000 Hz
VMax	DWORD	Highest frequency	10 Hz~100,000 Hz
ACCTime	WORD	Acceleration and deceleration time	0~32,767 ms (0 shows uniform)
INT_EN	BOOL	Interruption enable	0: Doesn't interrupt after finishing output of pulse 1: Interruption occurs after finishing output of pulse

Output Parameter	Type of Data	Description of Function	Instruction of Parameter Value
Active	BOOL	Mark of sending pulse	0: Stop sending 1: On the process of sending
CV	DINT	Number of sent pulse	-2,147,483,648~2,147,483,647
Error	BYTE	Error information	=0: Correct =1: The channel is taken up =2: Configuration output axis is error =3: Frequency output is error

3.5.1.9.3 Description of Instruction

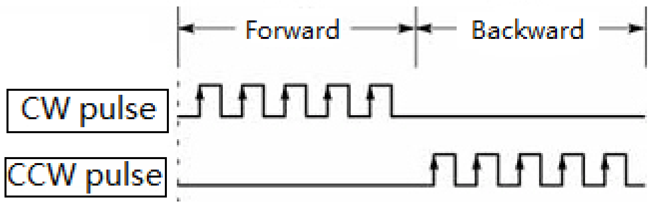
1. PLS+SIGN output mode



PLS+SIGN Output Mode

Speed points pulse to output, rotation direction use ON/OFF to point, under the mode, rotation direction OFF response to forward of electric machine.

2. CW/CCW output mode



CW/CCW Output Mode

Under the effect of the method, forward and backward shall be controlled by double pulse of pulse.

3.5.1.9.4 Example

Current position is 5000, as target value +7000, operating pulse output instruction:

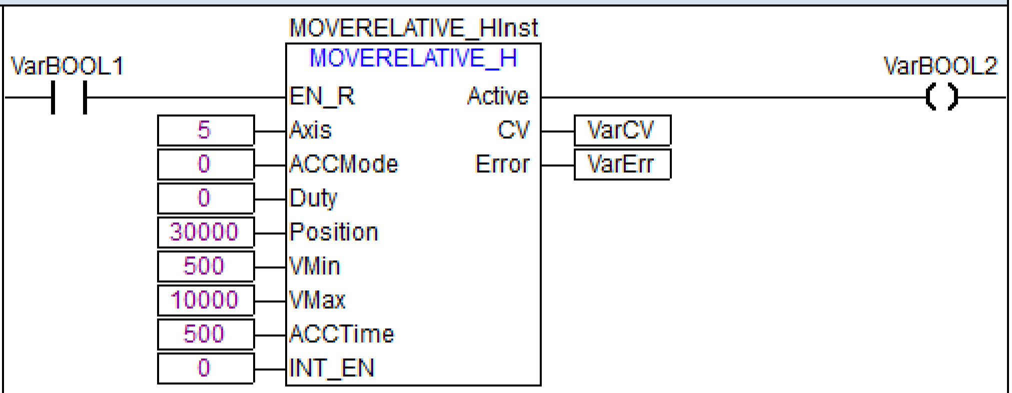
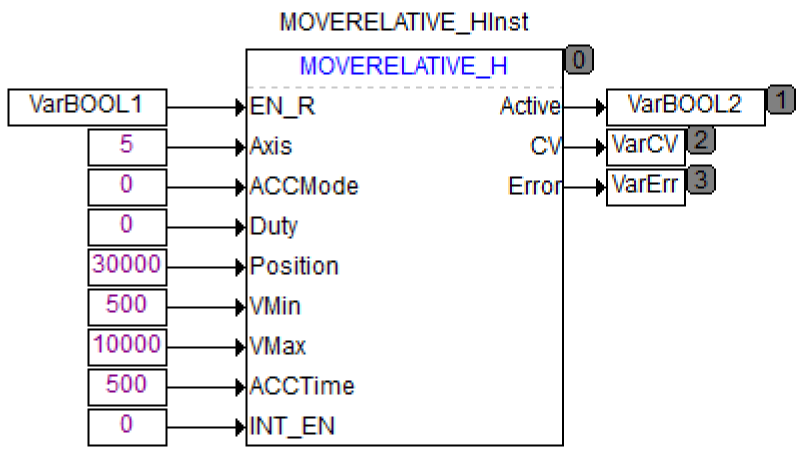
- Pulse + direction: output 7000 pulse from pulse output axis, direction output low level.
- CW/CCW: CW axis output 7000 pulse, CCW output low level.
- The current location is 12000 after the above two methods.

3.5.1.9.5 Application Example of Instructions

Variable Definition

No.	Variable Name	Address	Variable Description	Variable Type	Initial Value	Power Fail Safeguard
0001	MOVERELATIVE_HInst			MOVERELATIVE_H		FALSE
0002	VarBOOL1			BOOL	FALSE	FALSE
0003	VarBOOL2			BOOL	FALSE	FALSE
0004	VarCV			DINT	0	FALSE
0005	VarErr			BYTE	0	FALSE

Programing language	Program

<table>
<tr><th colspan="2">Variable Definition</th></tr>
<tr><td>Ladder diagram (LD)</td><td>
</td></tr>
<tr><td>Structured text (ST)</td><td>MOVERELATIVE_HInst(
EN_R:=VarBOOL1,
Axis:=5,
ACCMode:=0,
Duty:=0,
Position:=30000,
VMin:=500,
VMax:=10000,
ACCTime:=500,
INT_EN:=0,
Active=> VarBOOL2,
CV=> VarCV,
Error=> VarErr);</td></tr>
<tr><td>Continuous function diagram (CFC)</td><td>
</td></tr>
</table>

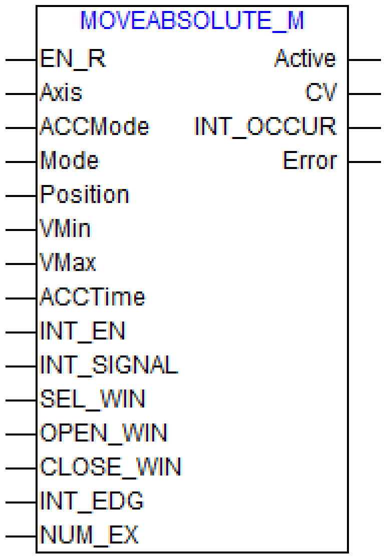
Program Instruction:

When the EN_R is set and held, the axes 1 (Q0.0 and Q0.1) send a pulse with a duty cycle of 50 in the double pulse mode, sending a total of 30,000, the initial frequency is 500 Hz, the maximum output frequency is 10000 Hz, acceleration and deceleration time is 500ms, trapezoidal acceleration / deceleration mode, CV shows the number of sent pulses.

3.5.1.10 MOVEABSOLUTE_M (Absolute Move,5128)

3.5.1.10.1 Description of Function

Used for controlling electric machine for positioning, taking original point as reference point to output pulse whose number and frequency has been set to reach positioning function. And support the interruption function.



MOVEABSOLUTE_M Function Block

3.5.1.10.2 Description of Parameter

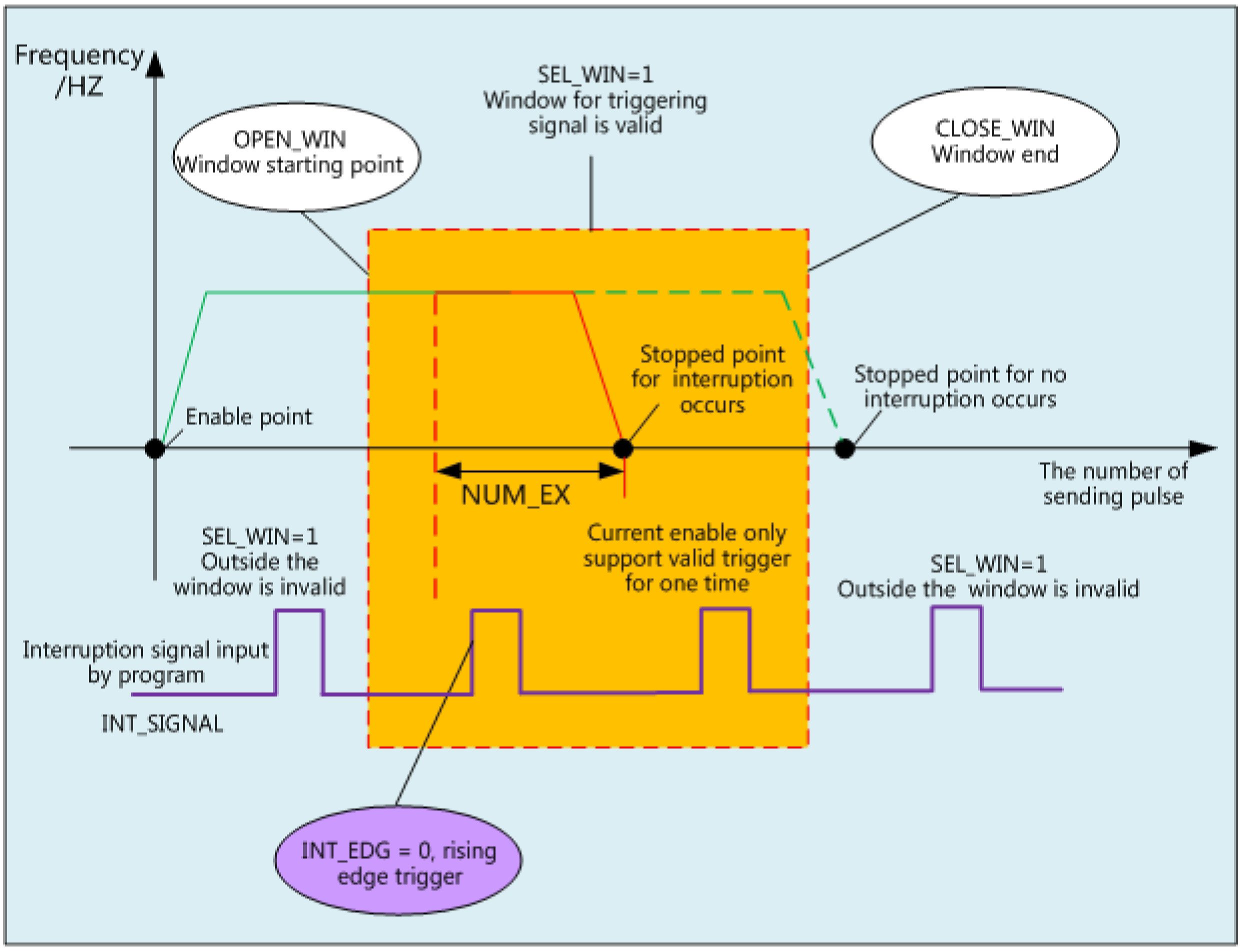
Input Parameter	Type of Data	Description of Function	Instruction of Parameter Value
EN_R	BOOL	Enable	0→1 Rising edge enable, high level keep
Axis	BYTE	Output axis selection	PLS+SIGN mode =5 : Output axis 5, pulse output end Q0.0, pulse direction Q0.4 =6: Output axis 6, pulse output end Q0.1, pulse direction Q0.5 =7: Output axis 7, pulse output end Q0.2, pulse direction Q0.6 =8: Output axis 8, pulse output end Q0.3, pulse direction Q0.7 CW/CCW mode =5, 7 Separately show output is : axis5, axis 7 If output 6 response to axis 5 output, if output 8 responses to axis 7 output
ACCMode	BOOL	Acceleration and deceleration mode selection	0: Trapezoidal curve 1: S curve
Mode	BOOL	Pulse mode control	0: PLS+SIGN mode 1: CW/CCW mode
Position	DINT	Absolute positioning value set	-1,073,741,824～1,073,741,823

Input Parameter	Type of Data	Description of Function	Instruction of Parameter Value
VMin	DWORD	Preliminary frequency	10 Hz~100,000 Hz
VMax	DWORD	The highest frequency	10 Hz~100,000 Hz
ACCTime	WORD	Acceleration and deceleration time	0~32,767 ms (0 shows uniform)
INT_EN	BOOL	Interruption enable	0: Doesn't interrupt after finishing output of pulse 1: Interruption occurs after finishing output of pulse
INT_SIGNAL	BOOL	Interruption input signal	
SEL_WIN	BYTE	Window selection	Constraint the validity of parameter of INT_EX hardware 0 : Not use window 1 : Inside the window is valid 2 : Outside the window is valid
OPEN_WIN	DINT	Widow 1 absolute value	-1,073,741,824～1,073,741,823
CLOSE_WIN	DINT	Widow 2 absolute value	-1,073,741,824～1,073,741,823
INT_EDG	BOOL	Level jump edge selection	INT_SIGNAL interruption level jump edge selection 0: Rising edge trigger 1: Falling edge trigger
NUM_EX	DINT	Target stopped positioning after interruption trigger	-1,073,741,824～1,073,741,823

Output Parameter	Type of Data	Description of Function	Instruction of Parameter Value
Active	BOOL	Mark of sending pulse	0: Stop sending 1 : On the process of sending
CV	DINT	Number of sent pulse	-1,073,741,824～1,073,741,823
INT_OCCUR	BOOL	Mark of validity of stop operation	0: The interrupt does not take effect 1: The interrupt takes effect
Error	BYTE	Error information	=0: Correct =1: The channel is occupied =2: Configured axis error =3: Frequency output error =4: Acceleration and deceleration time parameter setting error =5 : Setting parameter for absolute position is out of range =6: When the interruption signal reaches, the current location equals to interruption stopped target location =7: When the interruption signal reaches, the current location has crossed the interruption stopped target location

3.5.1.10.3 Examples of Operation

When the program interrupts signal is triggered during the process of output, operating the schematic diagram as shown in Figure.



Schematic Diagram of Parameter Related to INT_SIGNAL Instruction

The function block supports interrupt stop function; interruption signal is received through inputting parameter INT_SIGNAL. The edge trigger method of interruption input signal INT_SIGNAL is pointed by parameter INT_EDG.

When interruption input signal meets following conditions at the same time, then the trigger interruption operates:

- After function block re-enable and hasn't operated valid interruption, that INT_OCCUR parameter is FALSE.
- Before the axis doesn't reach the absolute positioning target location, INT_SIGNAL produces valid jump edge.
- When interruption signal reaches, the current location of axis meets the window selection condition. For example, when selecting SEL_WIN=1, the current location of axis shall be in the window range pointed by OPEN_WIN and CLOSE_WIN.
- The current location of axis doesn't jump over the interrupt target stopped location. When the axis moves forward, interrupt target stop location NUM_EX> current absolute location of axis; or if the axis moves backward, interrupt target stop location NUM_EX<current absolute location of axis.

ⓘ

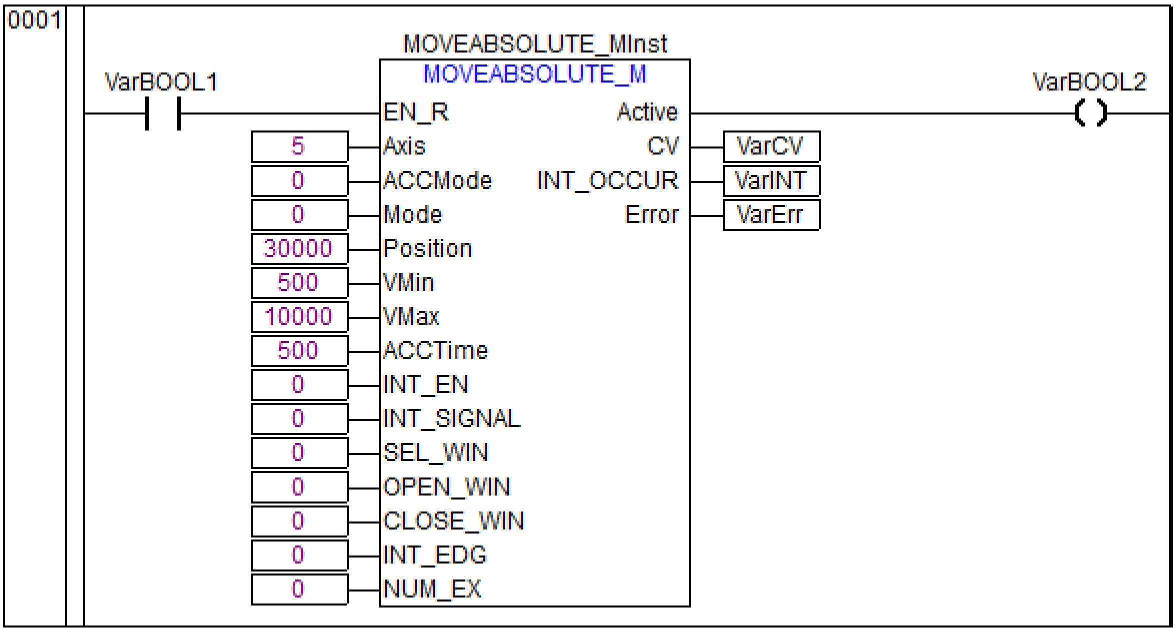
- If not using HOME instruction to set original point, then default the location before the first time using any one of MOVERELATIVE, MOVEABSOLUTE, JOG or HOME instruction to control the movement of electric machine. PWM instruction is not included, if using the two instructions to make the electric machines move before using the above function block, then the preliminary original point will change.
- The original point referred by the absolute position is the same original point as that set by HOME instruction.
- Absolute positioning function block is similar to relative positioning function block, only the selections of reference point are different, when absolutely positioning, the location point and window always take the original point as reference point, and the relative positioning always takes the current coordinate point as reference point. If current location is 5000, as target value +7000, operating pulse output instruction.
- Pulse + Direction: Output 2000 pulse from pulse output axis, direction outputs low level.
- CW/CCW: CW axis outputs 2000 pulse, CCW outputs low level.

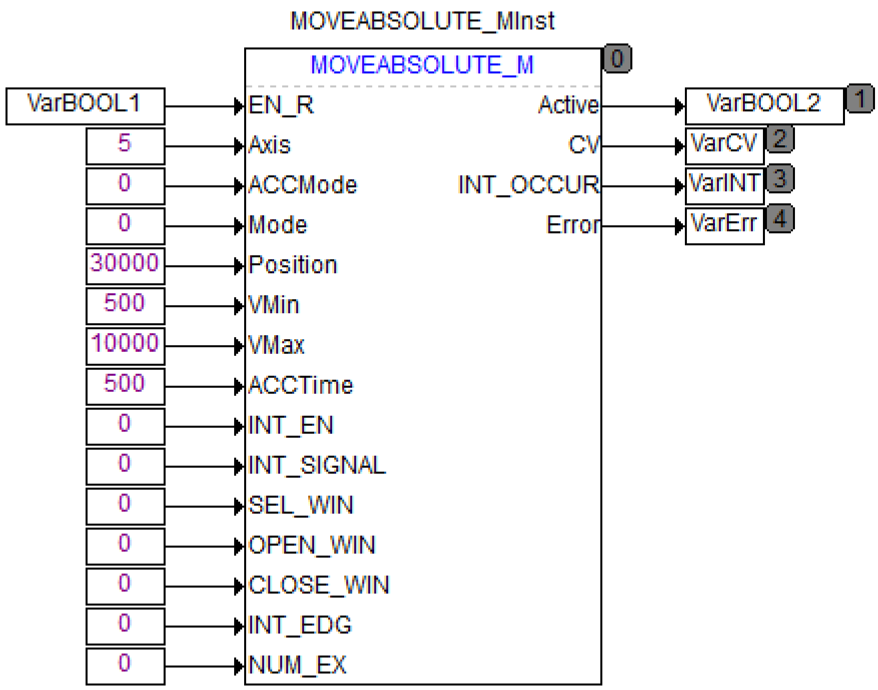
3.5.1.10.4 Application Example of Instructions

Variable Definition

No.	Variable Name	Address	Variable Description	Variable Type	Initial Value	Power Fail Safeguard
0001	MOVEABSOLUTE_MInst			MOVEABSOLUTE_M		FALSE
0002	VarBOOL1			BOOL	FALSE	FALSE
0003	VarBOOL2			BOOL	FALSE	FALSE
0004	VarCV			DINT	0	FALSE
0005	VarINT			BOOL	FALSE	FALSE
0006	VarErr			BYTE	0	FALSE

Programing language	Program

Variable Definition	
Ladder diagram (LD)	
Structured text (ST)	MOVEABSOLUTE_MInst(EN_R:=VarBOOL1, Axis:=5, ACCMode:=0, Mode:=0, Position:=30000, VMin:=500, VMax:=10000, ACCTime:=500, INT_EN:=0, INT_SIGNAL:=0, SEL_WIN:=0, OPEN_WIN:=0, CLOSE_WIN:=0, INT_EDG:=0, NUM_EX:=0, Active=>VarBOOL2, CV=> VarCV, INT_OCCUR=>VarINT, Error=>VarErr);

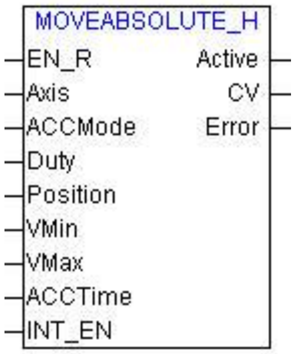
Variable Definition	
Continuous function diagram (CFC)	

When the EN_R is set and held, the axes 5 (Q0.0 and Q0.1) send a pulse with a duty cycle of 50 in the double pulse mode, moving to position for 30000 pulses (the number of pulses is calculated from the origin), the initial frequency is 500 Hz, the maximum output frequency is 10000 Hz, acceleration and deceleration time is 500ms, trapezoidal acceleration / deceleration mode, CV shows the number of sent pulses.

3.5.1.11 MOVEABSOLUTE_H (Absolute Move, 5106/5108/5118)

3.5.1.11.1 Description of Function

Used for controlling electric machine for positioning, taking original point as reference point to output pulse whose number and frequency has been set to reach positioning function.



MOVEABSOLUTE_H Function Block

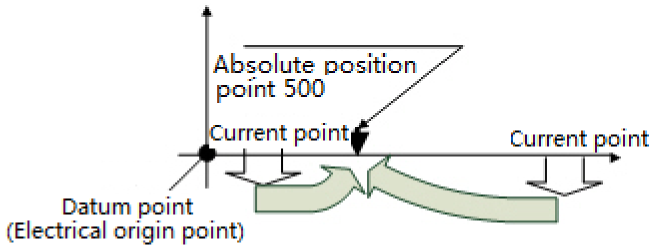
3.5.1.11.2 Description of Parameter

Input Parameter	Type of Data	Description of Function	Instruction of Parameter Value
EN_R	BOOL	Enable	0→1 Rising edge enable, high level keep
Axis	BYTE	Output channel selection	Pulse + direction mode 1 : Output axis 1, pulse output end Q0.0, pulse direction Q0.4 2 : Output axis 2, pulse output end Q0.1, pulse direction Q0.5 3 : Output axis 3, pulse output end Q0.2, pulse direction Q0.6 4 : Output axis 4, pulse output end Q0.3, pulse direction Q0.7 CW/CCW mode 5 : Output axis 1, pulse output end (CW) Q0.0, pulse output end (CCW) Q0.1 6 : Output axis 2, pulse output end (CW) Q0.2, pulse output end (CCW) Q0.3
ACCMode	BOOL	Acceleration mode selection	0: Trapezoidal curve 1: S curve
Duty	BOOL	Duty ratio	0: 50% 1: 25%
Position	DINT	Number of output pulse set	-2,147,483,648~2,147,483,647
VMin	DWORD	Preliminary frequency	10 Hz~100,000 Hz
VMax	DWORD	The highest frequency	10 Hz~100,000 Hz
ACCTime	WORD	Acceleration and deceleration time	0~32,767 ms (0 shows uniform)
INT_EN	BOOL	Interruption enable	0: Doesn’t interrupt after finishing output of pulse 1: Interruption occurs after finishing output of pulse

Output Parameter	Type of Data	Description of Function	Instruction of Parameter Value
Active	BOOL	Mark of sending pulse	0 : Stop sending 1 : On the process of sending
CV	DINT	Number of sent pulse	-2,147,483,648~2,147,483,647
Error	BYTE	Error information	=0: Correct =1: The channel is taken up =2: Configuration output axis is error =3: Frequency output is error

3.5.1.11.3 Description of Instruction

Absolute positioning function block is similar to relative positioning function block, only the reference point and calling times are different, the absolute positioning always takes original point as reference point, and relative positioning always takes current coordinate point as the reference point.



Absolute Positioning Schematic Diagram

3.5.1.11.4 Examples

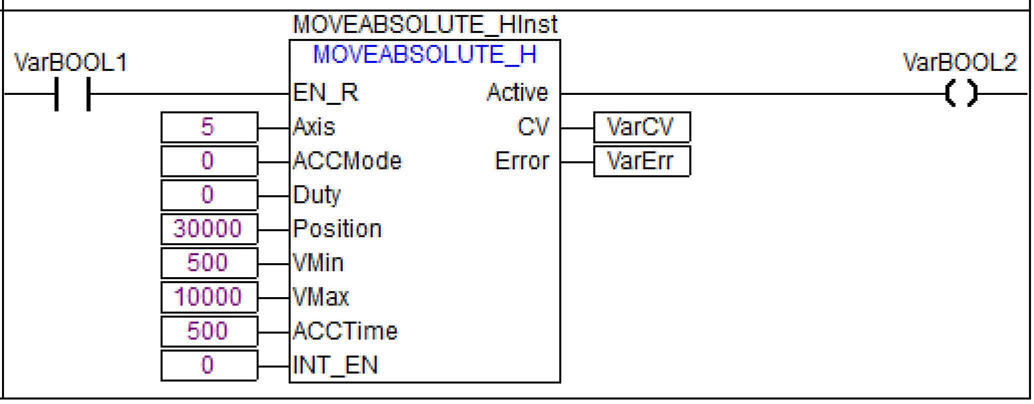
Current position is 5000, as target value +7000, operating pulse output instruction:

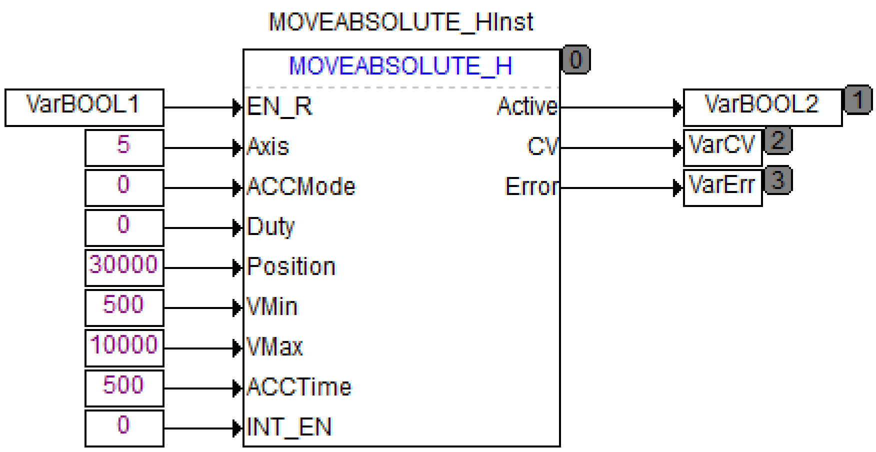
- Pulse + direction: output 2000 pulse from pulse output axis, direction output low level.
- CW⁄CCW: CW axis output 2000 pulse, CCW output low level
- The current location is 7000 after the above two methods.

3.5.1.11.5 Application Example of Instructions

Variable Definition

No.	Variable Name	Address	Variable Description	Variable Type	Initial Value	Power Fail Safeguard
0001	MOVEABSOLUTE_HInst			MOVEABSOLUTE_H		FALSE
0002	VarBOOL1			BOOL	FALSE	FALSE
0003	VarBOOL2			BOOL	FALSE	FALSE
0004	VarCV			DINT	0	FALSE
0005	VarErr			BYTE	0	FALSE

Programing language	Program
Ladder diagram (LD)	

Variable Definition	
Structured text (ST)	MOVEABSOLUTE_HInst(EN_R:=VarBOOL1, Axis:=5, ACCMode:=0, Duty:=0, Position:=30000, VMin:=500, VMax:=10000, ACCTime:=500, INT_EN:=0, Active=> VarBOOL2, CV=> VarCV, Error=> VarErr);
Continuous function diagram (CFC)	

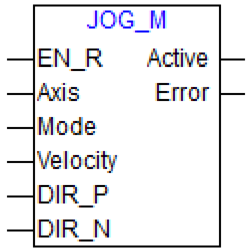
Program Instruction:

When the EN_R is set and held, the axes 1 (Q0.0 and Q0.1) send a pulse with a duty cycle of 50 in the double pulse mode, moving to position for 30000 pulses (the number of pulses is calculated from the origin), the initial frequency is 500 Hz, the maximum output frequency is 10000 Hz, acceleration and deceleration time is 500ms, trapezoidal acceleration / deceleration mode, CV shows the number of sent pulses.

3.5.1.12 JOG_M (Jog,5128)

3.5.1.12.1 Description of Function

Jog control of high speed output point. When the pointed forward switch is ON, the motor rotates forwards, when the backward switch is ON, the motor rotates backwards. If the forward and backward switches are ON or OFF at the same time, the motor will stop.



JOG_M Function Block

3.5.1.12.2 Description of Parameters

Input Parameter	Type of Data	Description of Function	Instruction of Parameter Value
EN_R	BOOL	Enable	0→1 Rising edge enable, high level is valid
Mode	BOOL	Mode control	0 : PLS+SIGN Mode 1: CW/CCW Mode
Axis	BYTE	Output axis selection	PLS+SIGN mode =5 : Output axis 5, pulse output end Q0.0, pulse direction Q0.4 =6: Output axis 6, pulse output end Q0.1, pulse direction Q0.5 =7: Output axis 7, pulse output end Q0.2, pulse direction Q0.6 =8: Output axis 8, pulse output end Q0.3, pulse direction Q0.7 CW/CCW mode =5: Axis 5 =7: Axis 7 Separately show output is : axis5, axis 7 If output 6 response to axis 5 output, if output 8 responses to axis 7 output
Velocity	DWORD	Output frequency	10 Hz~100,000 Hz
DIR_P	BOOL	Forward output control end	If forward output control end is ON, forward pulse output or CW pulse output
DIR_N	BOOL	Backward output control end	If backward output control end is ON, backward pulse output or CW pulse output

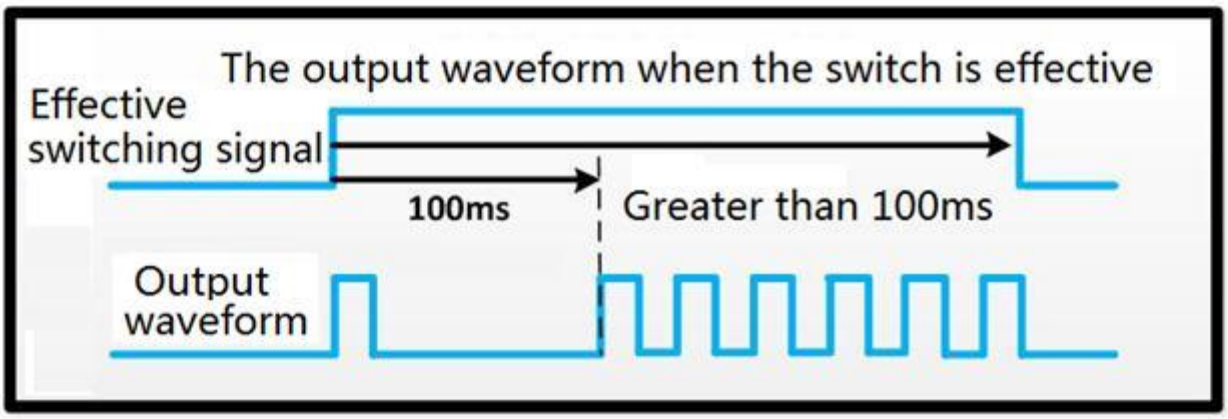
Output Parameter	Type of Data	Description of Function	Instruction of Parameter Value
Active	BOOL	Mark of sending pulse	0: Stop sending pulse 1: On the process of sending pulse
Error	BYTE	Error information	=0: Correct =1: The channel is taken up =2: Configuration axis is error =3: Frequency output is error

ⓘOverview of operation: when DIR-P, DIR-N enable signal is connected, producing a forward or backward pulse corresponding to the minimum instruction unit (Velocity). If the enable time is over 100ms, outputting pulse constantly.

3.5.1.12.3 Examples of Instruction

No.	Variable Name	Address	Variable Description	Variable Type	Initial Value	Power Fail Safeguard
0001	VarBOOL1			BOOL	FALSE	FALSE
0002	VarBOOL2			BOOL	FALSE	FALSE
0003	VarErr			BYTE	0	FALSE
0004	JOG_MInst			JOG_M		FALSE

（a）



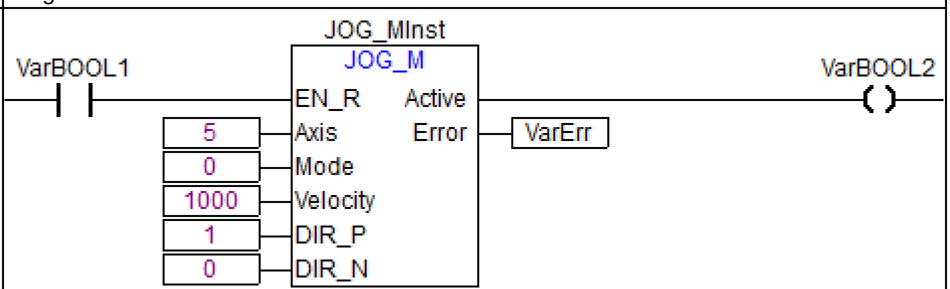
（b）

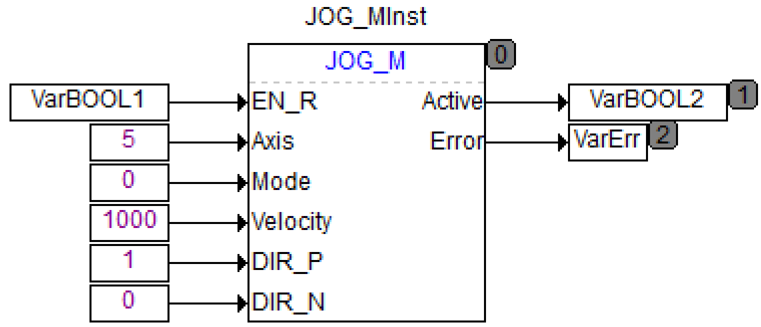
Schematic Diagram of Jog Instruction

3.5.1.12.4 Application Example of Instructions

Variable Definition

No.	Variable Name	Address	Variable Description	Variable Type	Initial Value	Power Fail Safeguard
0001	VarBOOL1			BOOL	FALSE	FALSE
0002	VarBOOL2			BOOL	FALSE	FALSE
0003	VarErr			BYTE	0	FALSE
0004	JOG_MInst			JOG_M		FALSE

Programing language	Program
Ladder diagram (LD)	
Structured text (ST)	JOG_MInst(EN_R:=VarBOOL1,

Variable Definition	
	Axis:=5, Mode:=0, Velocity:=1000, DIR_P:=1, DIR_N:=0, Active=> VarBOOL2, Error=> VarErr);
Continuous function diagram (CFC)	

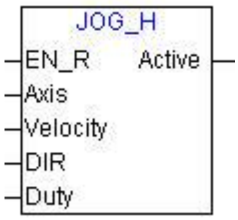
Program Instruction:

When EN_R is set and held, axes 1 (Q0.0 and Q0.4) send a pulse with a duty cycle of 50 in the double pulse mode, moving according to points and the output frequency is 1000Hz. If the EN_R hold time is ≥0.1 seconds, the pulse will continue to be sent until EN_R becomes FALSE.

3.5.1.13 JOG_H (Jog,5106/5108/5118)

3.5.1.13.1 Description of Function

Jog control of output point at high speed.



JOG_H Function Block

3.5.1.13.2 Description of Parameter

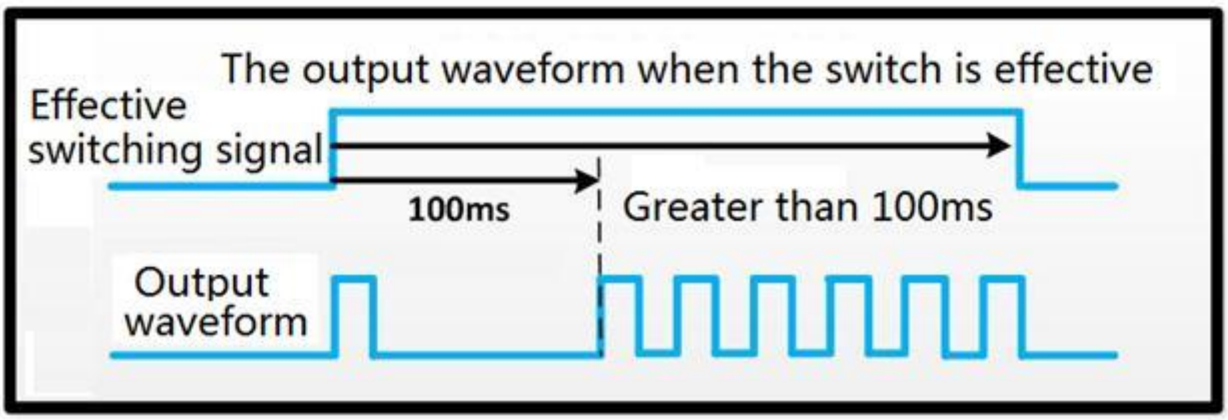
Input Parameter	Type of Data	Description of Function	Instruction of Parameter Value
EN_R	BOOL	Enable	0→1 Rising edge enable, level keep
Axis	BYTE	Output axis selection	Pulse + direction mode 1 : Output axis 1, pulse output end Q0.0, pulse direction Q0.4 2 : Output axis 2, pulse output end Q0.1, pulse direction Q0.5 3 : Output axis 3, pulse output end Q0.2, pulse direction Q0.6 4 : Output axis 4, pulse output end Q0.3, pulse direction Q0.7 CW/CCW mode

Input Parameter	Type of Data	Description of Function	Instruction of Parameter Value
			5 : Output axis 1, pulse output end (CW) Q0.0, pulse output end (CCW) Q0.1 6 : Output axis 2, pulse output end (CW) Q0.2, pulse output end (CCW) Q0.3
Velocity	DWORD	Output frequency	10 Hz~100,000 Hz
DIR	BOOL	Direction mode	Action mode and output logistic relationship 1 : Direction output ON, CCW direction 0 : Direction output OFF, CW direction
Duty	BOOL	Duty ratio	0: 50% 1: 25%

Output Parameter	Type of Data	Description of Function	Instruction of Parameter Value
Active	BOOL	Mark of sending pulse	0 : Stop sending pulse 1: On the process of sending pulse

- Parameter DIR can be set by users to control jog direction.
- Overview of operation: when the input enable signal is connected, producing a forward or backward pulse corresponding to the minimum instruction unit. If the enable time is over 0.1s, output pulse constantly.

3.5.1.13.3 Example of Instructions

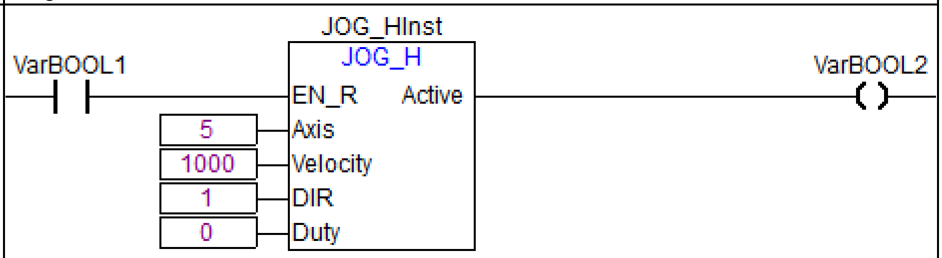
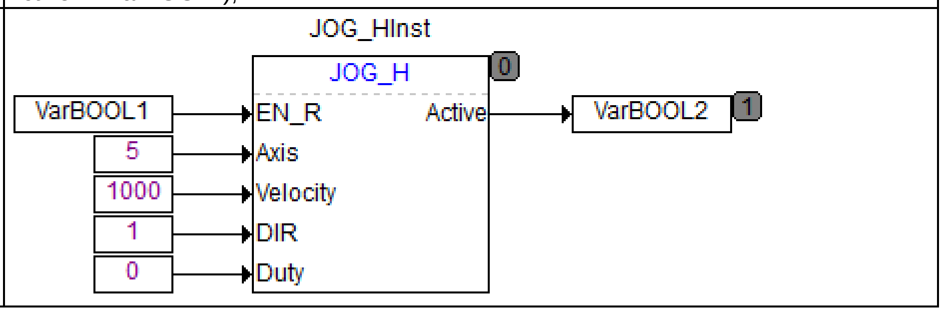


Schematic Diagram of Jog Instruction

3.5.1.13.4 Application Example of Instructions

Variable Definition

No.	Variable Name	Address	Variable Description	Variable Type	Initial Value	Power Fail Safeguard
0001	VarBOOL1			BOOL	FALSE	FALSE
0002	VarBOOL2			BOOL	FALSE	FALSE
0003	JOG_HInst			JOG_H		FALSE

Programing language	Program
Ladder diagram (LD)	
Structured text (ST)	JOG_HInst(EN_R:=VarBOOL1, Axis:=5, Velocity:=1000, DIR:=1, Duty:=0, Active=> VarBOOL2);
Continuous function diagram (CFC)	

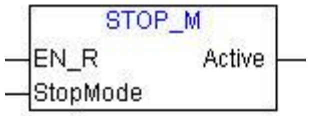
Program Instruction:

When EN_R is set and held, axes 1 (Q0.0 and Q0.1) send a pulse with a duty cycle of 50 in the double pulse mode, moving according to points and the output frequency is 1000Hz. If the EN_R hold time is ≥0.1 seconds, the pulse will continue to be sent until EN_R becomes FALSE.

3.5.1.14 STOP_M (Stop, 5128)

3.5.1.14.1 Description of Function

Instruction cancels the current motion of the whole axis and stop.



STOP_M Function Block

3.5.1.14.2 Description of Parameter

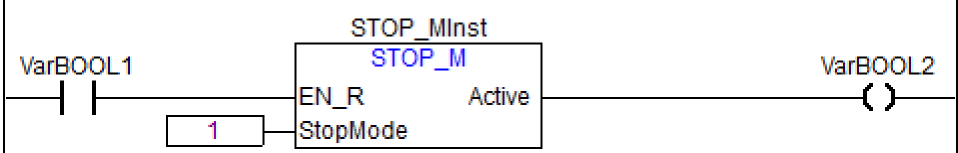
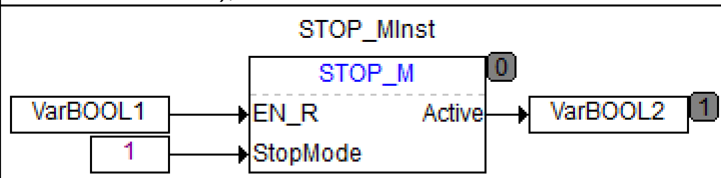
Input Parameter	Type of Data	Description of Function	Instruction of Parameter Value
EN_R	BOOL	Enable	0→1 Rising edge enable, high level is valid
StopMode	BOOL	Stop mode	=0: Achieve stop according to the set deceleration rate or the relative deceleration ratio of deceleration time =1: Stop output immediately

Output Parameter	Type of Data	Description of Function	Instruction of Parameter Value
Active	BOOL	Mark of finishing stop	0: On the process of stop 1: All of the stop are finished

3.5.1.14.3 Application Example of Instructions

Variable Definition

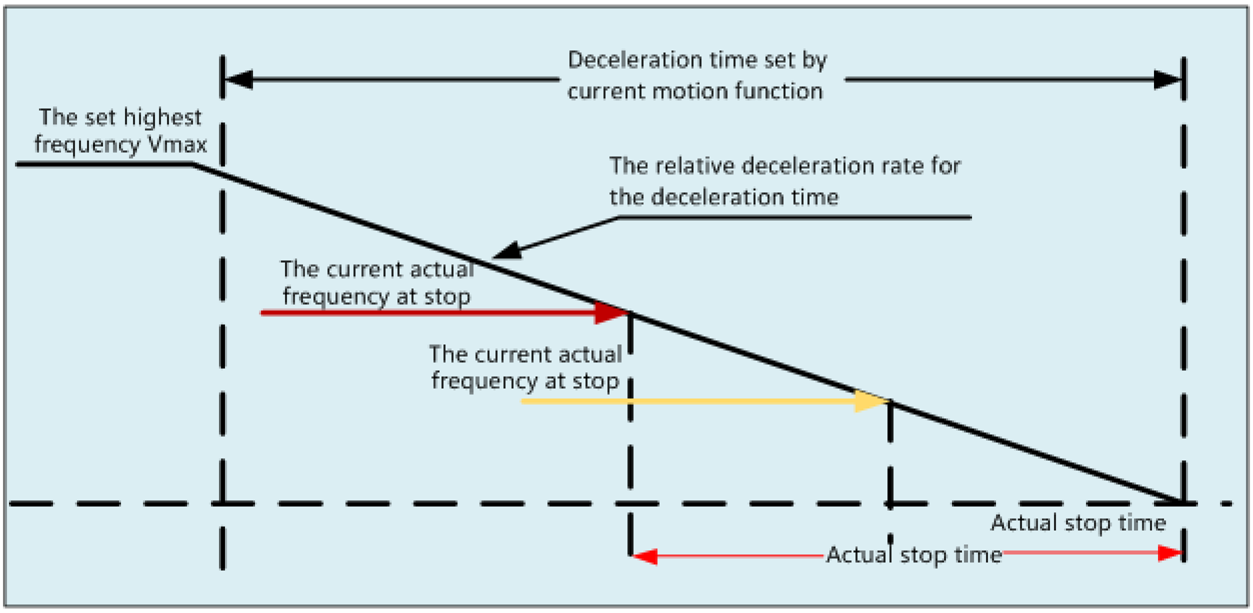
No.	Variable Name	Address	Variable Description	Variable Type	Initial Value	Power Fail Safeguard
0001	VarBOOL1			BOOL	FALSE	FALSE
0002	VarBOOL2			BOOL	FALSE	FALSE
0003	STOP_MInst			STOP_M		FALSE

Programing language	Program
Ladder diagram (LD)	
Structured text (ST)	STOP_MInst(EN_R:=VarBOOL1, StopMode:=1, Active=> VarBOOL2);
Continuous function diagram (CFC)	

3.5.1.14.4 Special Instruction

Achieve the stop of two function block STOP_M and CANCEL when the deceleration is stopping

(StopMode=0) according to the set deceleration rate and speed rate (the relative deceleration rate for the deceleration time). As shown in Figure.

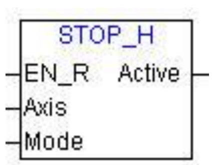


Schematic Diagram of Deceleration Stop

3.5.1.15 STOP_H (Stop, 5106/5108/5118)

3.5.1.15.1 Description of Function

Achieve the immediately stop of pulse.



STOP_H Function Block

3.5.1.15.2 Description of Parameter

Input Parameter	Type of Data	Description of Function	Instruction of Parameter Value
EN_R	BOOL	Enable	0→1 Rising edge enable, level keep
Mode	BOOL	Mode selection	0 : Not bind deviation counter reset output end 1 : Bind deviation counter reset output end
Axis	BYTE	Output axis selection	PLS + SIGN mode 1 : Output axis 1, pulse Q0.0, direction Q0.4, deviation counting clearing Q1.0 2 : Output axis 2, pulse Q0.1, direction Q0.5, deviation counting

Input Parameter	Type of Data	Description of Function	Instruction of Parameter Value
			clearing Q1.1 3 : Output axis 3, pulse Q0.2, direction Q0.6, deviation counting clearing Q1.2 4 : Output axis 4, pulse Q0.3, direction Q0.7, deviation counting clearing Q1.3 CW/CCW mode =5 : Output axis 1, pulse output end (CW) Q0.0, pulse output end (CCW) Q0.1 =6 : Output axis 2, pulse output end (CW) Q0.2, pulse output end (CCW) Q0.3

Output Parameter	Type of Data	Description of Function	Instruction of Parameter Value
Active	BOOL	Mark of sending pulse	1: Stop sending pulse 0: On the process of sending pulse

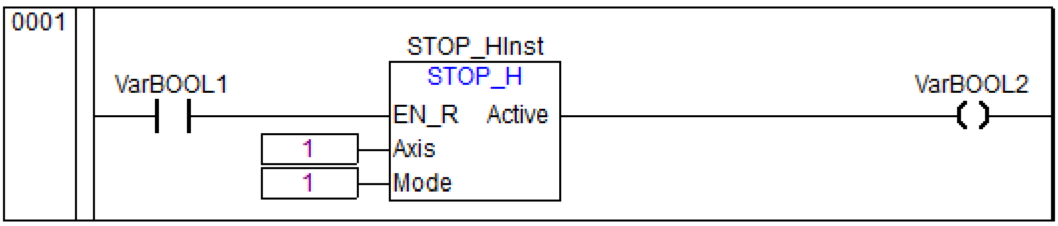
3.5.1.15.3 Application Example of Instructions

Variable Definition

No.	Variable Name	Address	Variable Description	Variable Type	Initial Value	Power Fail Safeguard
0001	VarBOOL1			BOOL	FALSE	FALSE
0002	VarBOOL2			BOOL	FALSE	FALSE
0003	STOP_HInst			STOP_H		FALSE

Programing language	Program

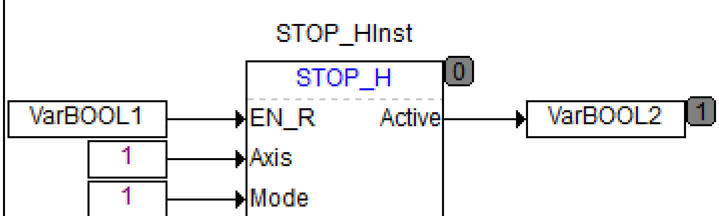
Ladder diagram (LD)



Structured text (ST)

```
STOP_HInst(
EN_R:=VarBOOL1,
Axis:=1,
Mode:=1,
Active=> VarBOOL2);
```

Continuous function diagram (CFC)



Program Instruction:

When EN_R is set and held, the motion of axis 1 (Q0.0, send pulse in pulse + direction mode) is stopped and the deviation counter (Q1.0) is reset.

3.5.1.15.4 Description of instruction

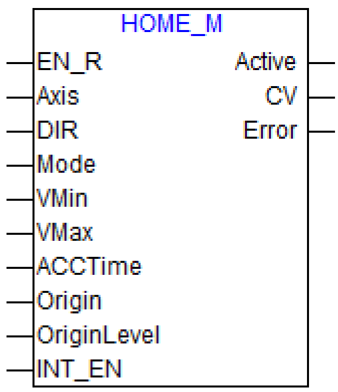
Example: if Mode is set as 1, and bind with deviation counter reset output end. If under the case that input parameter Axis is set as 1, after the function block enable, pulse output end Q0.0 can stop sending pulse, and corresponding deviation counter output end Q1.0 output signal that makes the deviation counter of servo electric machine clearing.

If Mode is set as 0, then the deviation counter set end doesn’t output the signal, which is the stop instruction and deviation counter output end don’t have any contact at this time.

3.5.1.16 HOME_M (Regression of Original Point, 5128)

3.5.1.17 Description of Function

Control corresponding high-speed output channel to make the motor move from the current location to set point.



HOME_M Function Block

3.5.1.18 Description of Parameter

Input Parameter	Type of Data	Description of Function	Instruction of Parameter Value
EN_R	BOOL	Enable	0→1 Rising edge enable, high level keep
Axis	BYTE	Output axis selection	PLS+SIGN mode =5 : Output axis 5, pulse output end Q0.0, pulse direction Q0.4 =6: Output axis 6, pulse output end Q0.1, pulse direction Q0.5 =7: Output axis 7, pulse output end Q0.2, pulse direction Q0.6 =8: Output axis 8, pulse output end Q0.3, pulse direction Q0.7 CW/CCW mode =5, 7

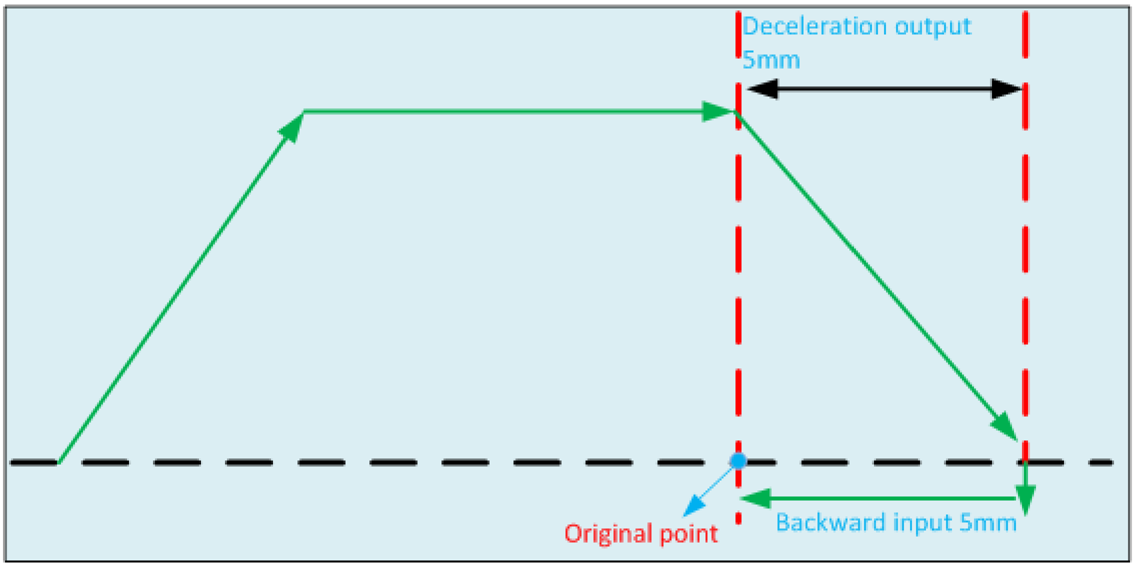
Input Parameter	Type of Data	Description of Function	Instruction of Parameter Value
			Separately show output is : axis5, axis 7 If output 6 response to axis 5 output, if output 8 responses to axis 7 output
DIR	BOOL	Original point search direction	0: CW direction, forward pulse output 1: CCW direction, backward pulse output
Mode	BYTE	Mode	Bit0 status 0: Rising edge of original point signal is valid 1: Falling edge of original point signal is valid Bit1 status 0 : Return to original point mode 1 : Doesn’t return to original point mode Bit2 status 0: PLG+SIGN 1: CW/CCW
VMin	DWORD	Speed of crawl	10 Hz～100,000 Hz
VMax	DWORD	Regression speed	10 Hz～100,000 Hz
ACCTime	WORD	Acceleration and deceleration time	0～32,767 ms (0 shows uniform)
Origin	BYTE	Original point input point selection	1~16: I0.0~I1.7
OriginLevel	BOOL	original point valid level	0: Low level is valid 1: High level is valid
INT_EN	BOOL	Interruption enable	0: Doesn’t interrupt after completing back to zero 1: Interruption occurs after completing back to zero

Output Parameter	Type of Data	Description of Function	Instruction of Parameter Value
Active	BOOL	mark of original point search completion	1: On the process of regression 0: Doesn’t return to original point and regression completion
CV	DINT	number of having sent pulse	-1,073,741,824～1,073,741,823 Before finding the origin, the CV value is the number of the total transmitted pulses After finding the origin, the CV value is the number of transmitted pulses after finding the origin
Error	BYTE	Error information	=0: Correct =1: The channel is taken up =2: The configuration output axis error =3: Frequency output error =4: Original point input point selection error =5: Acceleration and deceleration time parameter setting error

3.5.1.19 Description of instruction

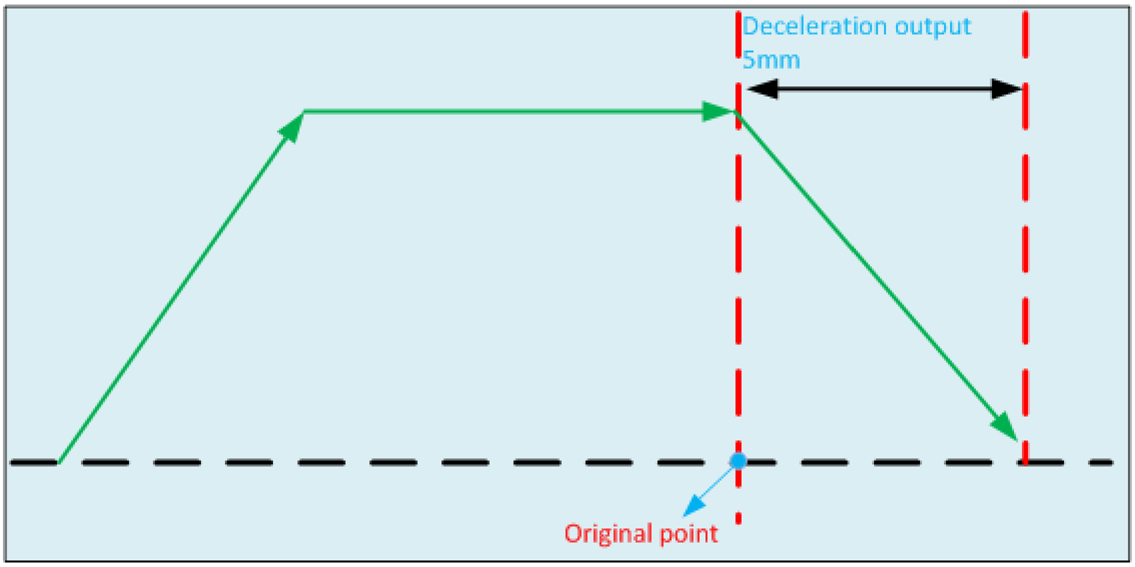
Please refer to Description_of_instruction.

3.5.1.20 Example of Mode of Original Point Search Returning to Original Point



Example of Mode of Original Point Search Returning to Original Point

3.5.1.21 Example of Mode of Original Point Search Not Returning to Original Point

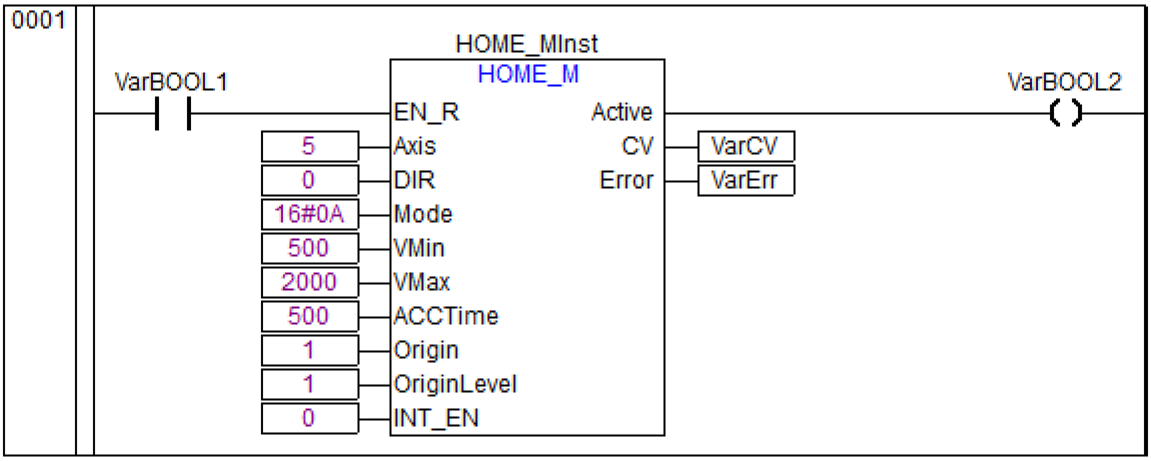


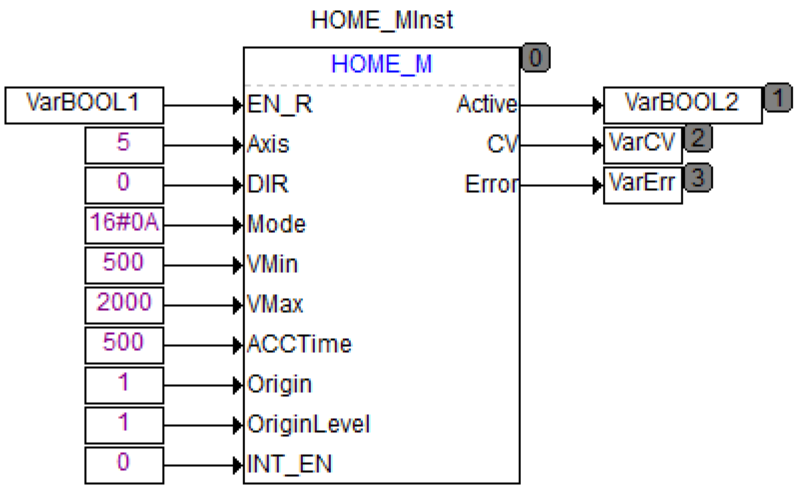
Example of Mode of Original Point Search Not Returning to Original Point

3.5.1.22 Application Example of Instructions

Variable Definition

No.	Variable Name	Address	Variable Description	Variable Type	Initial Value	Power Fail Safeguard
0001	VarBOOL1			BOOL	FALSE	FALSE
0002	VarBOOL2			BOOL	FALSE	FALSE
0003	VarCV			DINT	0	FALSE
0004	VarErr			BYTE	0	FALSE
0005	HOME_MInst			HOME_M		FALSE

Programing language	Program
Ladder diagram (LD)	
Structured text (ST)	HOME_MInst(EN_R:=VarBOOL1, Axis:=5, DIR:=0, Mode:=16#0A, VMin:=500, VMax:=2000, ACCTime:=500, Origin:=1, OriginLevel:=1 INT_EN:=0, Active=> VarBOOL2, CV=>VarCV, Error=> VarErr);

Variable Definition	
Continuous function diagram (CFC)	

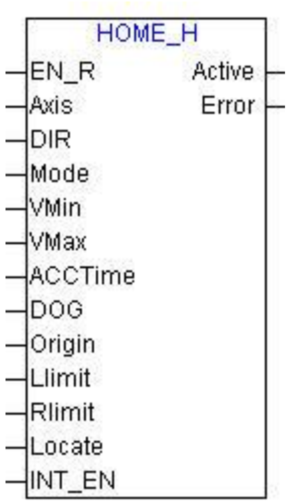
Program Instruction:

When EN_R is set and held, the axis 5 of the servomotor performs the origin search in the positive direction according to the pulse + direction mode. The search mode follows the work mode 2, method 2 of original point test and Invert mode 1 (refer to HOME_H (Regression of Original Point,5106/5108) for details) , Which crawling speed is 500Hz, the return speed is 2000Hz, acceleration and deceleration time is 500ms. The origin pulse input is provided by I0.0 and no interruption.

3.5.1.23 HOME_H (Regression of Original Point,5106/5108/5118)

3.5.1.23.1 Description of Function

Controlling relative high-speed output channel to make the electric machine move from the current location to mechanical original point or set the current point as original point of electric gas through positioning signal.



HOME_H Function Block

3.5.1.23.2 Description of Parameter

Input Parameter	Type of Data	Description of Function	Instruction of Parameter Value
EN_R	BOOL	Enable	0→1 Rising edge enable, high level keep
Axis	BYTE	Output channel selection	PLS + SIGN mode 1: Output axis 1, pulse Q0.0, direction Q0.4, deviation counting clearing Q1.0 2: Output axis 2, pulse Q0.1, direction Q0.5, deviation counting clearing Q1.1 3: Output axis 3, pulse Q0.2, direction Q0.6, deviation counting clearing Q1.2 4: Output axis 4, pulse Q0.3, direction Q0.7, deviation counting clearing Q1.3 CW/CCW mode =5: Output axis 1, pulse output end (CW) Q0.0, pulse output end (CCW) Q0.1 =6 : Output axis 2, pulse output end (CW) Q0.2, pulse output end (CCW) Q0.3
DIR	BOOL	Search direction of original point	0: CW direction 1: CCW direction
Mode	BYTE	Mode	Refer to Detailed_Instruction_of_Mode_Parameter
VMin	DWORD	Crawl speed	10 Hz～100,000 Hz
VMax	DWORD	Regression speed	10 Hz～100,000 Hz
ACCTime	WORD	Acceleration and deceleration time	0～32,767 ms(o shows uniform)
DOG	BYTE	Near-point input point selection	0: Null 1~24: I0.0~I2.7
Origin	BYTE	Original point input point selection	1~24: I0.0~I2.7
L_Limit	BYTE	Left limit input point selection	0: Null 1~24: I0.0~I2.7
R_Limit	BYTE	Right limit input point selection	0: Null 1~24: I0.0~I2.7
Locate	BYTE	Positioning input point selection	0: Null 1~24: I0.0~I2.7
INT_EN	BOOL	Interruption enable	0: Doesn't interrupt after finishing output of pulse 1: Interruption occurs after finishing output of pulse

Output Parameter	Type of Data	Description of Function	Instruction of Parameter Value
Active	BOOL	Mark of completion of regression of original point	1: On the process of regression 0: Complete regression
Error	BYTE	Error information	=0: Correct =1: The channel is taken up =2: The configuration output axis error =3: Frequency output error =4: Mode set error

3.5.1.23.3 Description of instruction

Detailed Instruction of Mode Parameter

Serial No.	Overview of Function	Description
Bit7	-	Reservation
Bit6～Bit5	Action mode	00: Original point search 01: Original point positioning 10: Original point reset 11: Invalid
Bit4～Bit3	Work mode	00: Work mode 1 (stepper motor) 01: Work mode 2 (servo motor) 10: Work mode 3 (servo motor) 11: Invalid
Bit2～Bit1	Method of original point test	00: Method 1 of original point test 01: Method 2 of original point test 10: Method 3 of original point test 11: Invalid
Bit0	Invert acts	0: Invert mode 1 1: Invert mode 2

■ Action mode select

- Original point searching: when operating original point search, output pulse makes the motor act actually, using the input signal near the original point and original point to decide the original point, and the original point that is tested is the mechanical original point of the system.
- Original point positioning: when operating original point positioning, setting the current value of the motor as original point of coordinate, that is original point of electric. If only operating original point search and no original point positioning, the original point of electric of the system coincides with the mechanical original point.
- Original point reset: when operating original point reset, after operating original point search or original point positioning, that is under the situation that there exists mechanical original point or motor original point, the motor can move to the set motor original point according to the current location.

ⓘ
Even under the case that the original point is not set (original point search and original point positioning are not operated), the motor can work, but it can have following phenomena:

- Original point reset and absolute positioning instruction are not available.
- Relative positioning instruction can be used normally.
- When regression speed of original point search <= crawl speed of original point search, the original point search is not operated.

■ Work mode selection

Work mode is the parameter of input signal when deciding original search. According to the deviation counter reset output signal and positioning interruption input, there will be following three modes.

Working Mode

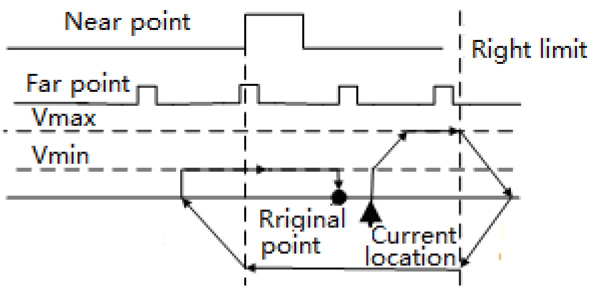
Working Mode	**Input and Output Signal**		
	Original input signal	Deviation counter reset	Positioning completion input
1	Testing original point input signal rising edge, and confirm the original point	Disuse	Disuse
2		When original point is tested, 20~30ms (duration of high level)	
3			After original point test, until the positioning completion of driver enters, the original point search acts come to end.

Above three modes shall be set corresponding work mode for actual use according to the used driver and usage.

Stepper motor driver always use work mode 1.

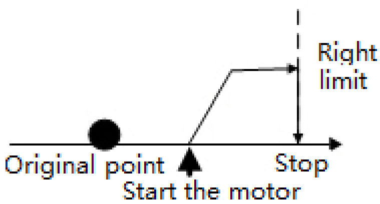
When using servo motor driver (or the stepper motor driver with positioning completion signal), if one hopes that it can shorten the operating time and doesn't care about the accuracy of the positioning and use work mode 2 (the mode doesn't use positioning completion signal near the servo driver); if one hopes to improve the accuracy of positioning and use work mode 3 (using the positioning completion signal of servo driver).

- Invert action: that is during the process of original point search, the motor's movement keeps away from original point signal.
  - □ Invert mode 1: if the motor tests the limit switch firstly, the motor operates backward to continue to test original point after the motor stops until testing it. The schematic diagram is as shown in Figure.



Invert Mode 1

  - □ Invert mode 2: if the motor tests the limit switch, the motor immediately stops original point search. The schematic diagram is as shown in Figure.

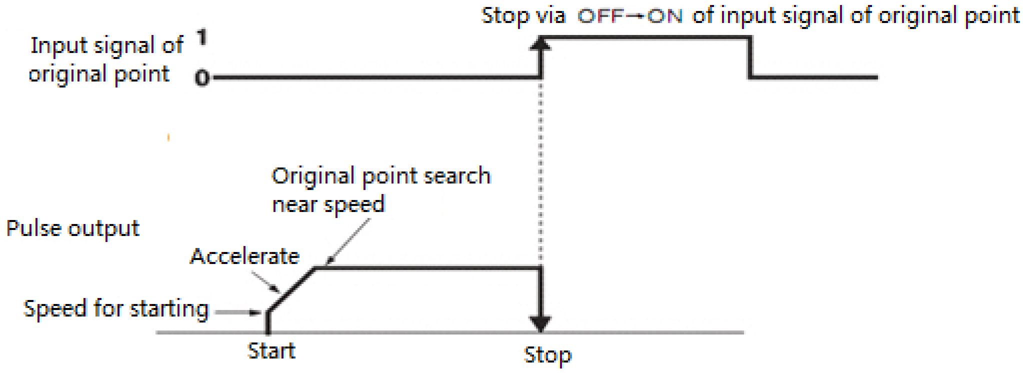


Invert Mode 2

- Method of original point test :
  - □ Method 1 of original point test (not use near original point signal), detailed steps are as follows :

After the reset of original point is started, going forward constantly to the original point regression direction at regression speed VMax.

When the inducted point tests the original point input signal, the pulse output shall stop immediately.



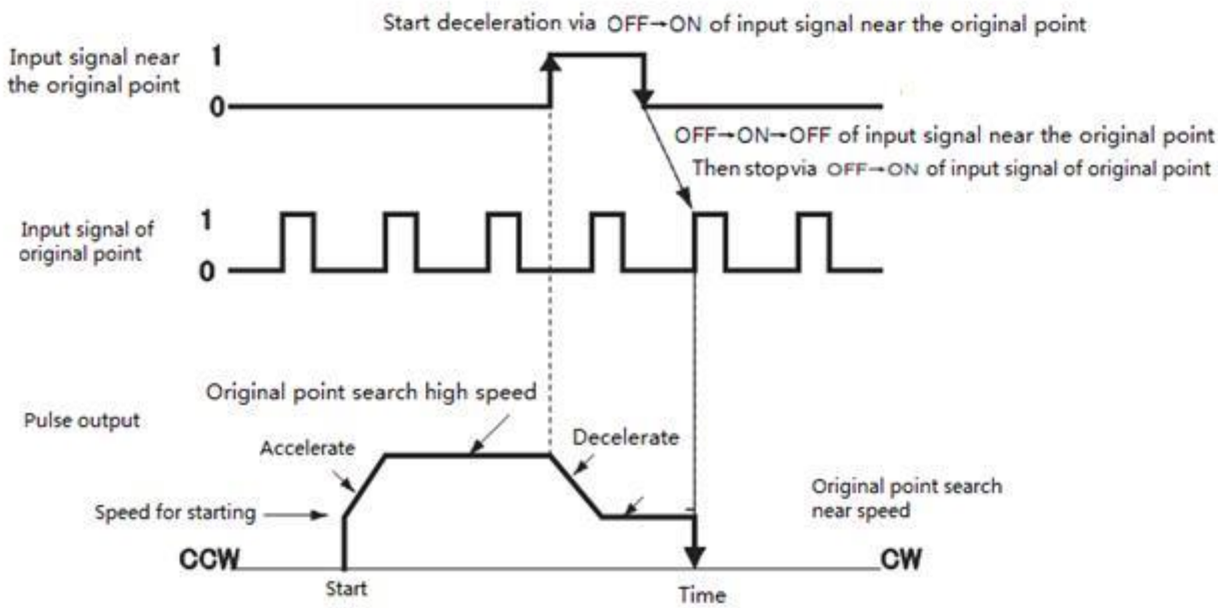
Method 1 of Original Point Test

□ Method 2 of original point test (not use near original point signal), detailed steps are as follows :

After the reset of original point is started, going forward constantly to the original point regression direction at regression speed VMax.

When meeting near-point induction switch DOG sensor, making deceleration action by deceleration slope, and decelerating to reach the crawl speed (front is set as VMin), then go forward constantly.

After the inducted point I leaves near-point induction switch DOG sensor, when testing the original point signal, the pulse output shall be stopped immediately, and finish the action of original point search.



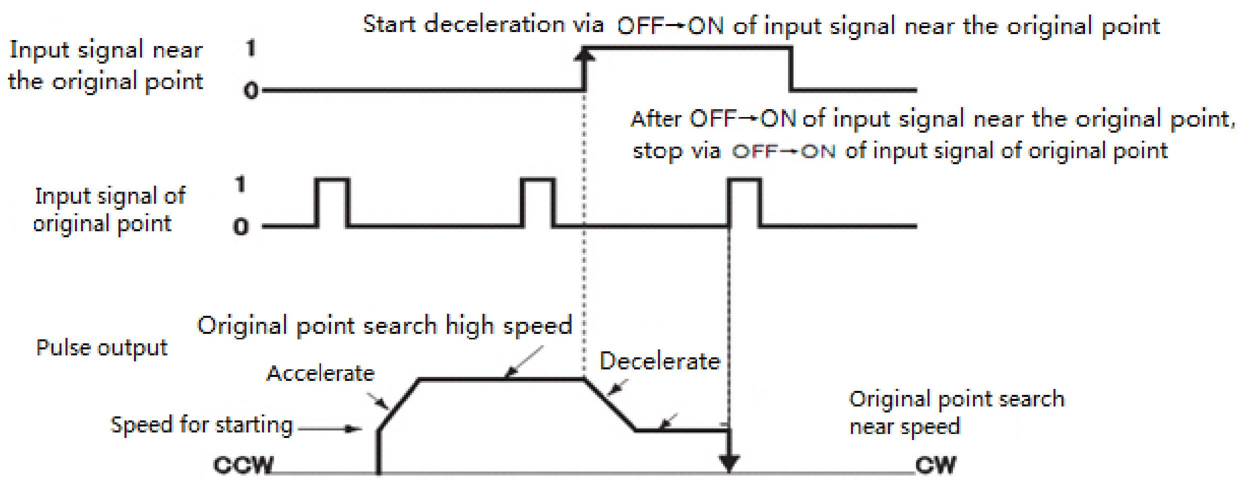
Method 2 of Original Point Test

□ Method 3 of original point test (not use near original point signal), detailed steps are as follows :

After the reset of original point is started, going forward constantly to the original point regression direction at regression speed VMax.

When meeting near-point induction switch DOG sensor, making deceleration action by deceleration slope, and decelerating to reach the crawl speed (front is set as VMin), then go forward constantly.

When testing the original point signal, the pulse output shall be stopped immediately, and finish the action of original point search.



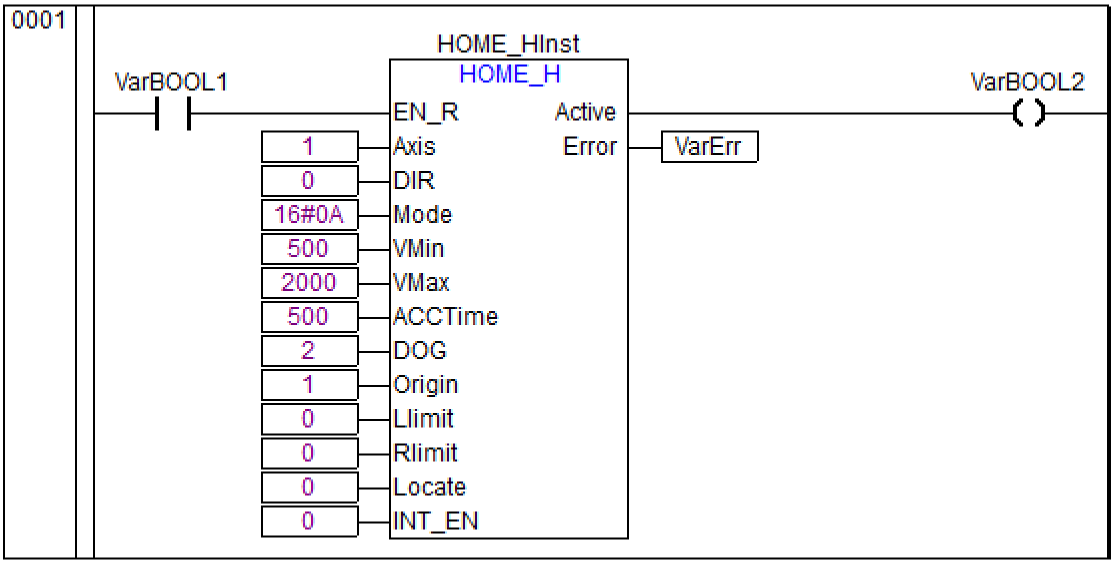
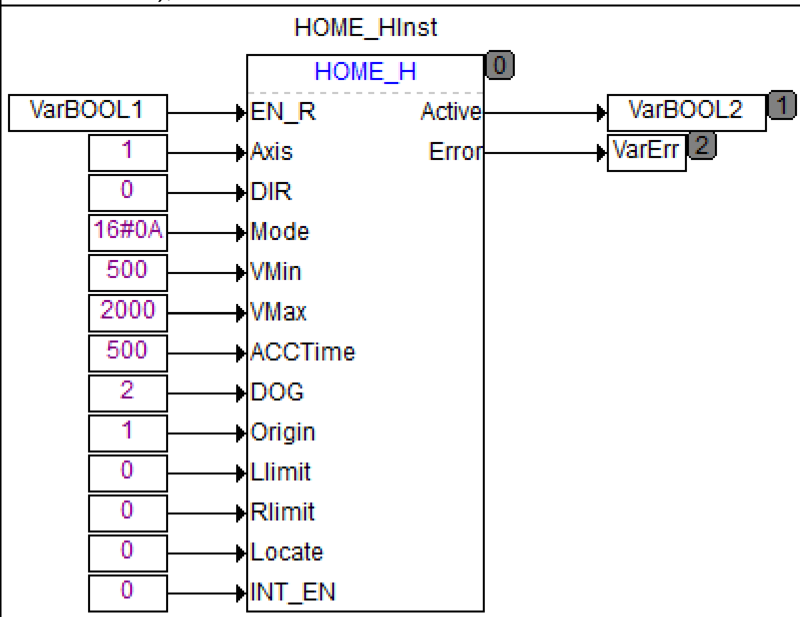
Method 3 of Original Point Test

3.5.1.23.4 Application Example of Instructions

Variable Definition

No.	Variable Name	Address	Variable Description	Variable Type	Initial Value	Power Fail Safeguard
0001	VarBOOL1			BOOL	FALSE	FALSE
0002	VarBOOL2			BOOL	FALSE	FALSE
0003	HOME_HInst			HOME_H		FALSE
0004	VarErr			BYTE	0	FALSE

Programing language	Program

<table>
<tr><th colspan="2">Variable Definition</th></tr>
<tr><td>Ladder diagram (LD)</td><td>
</td></tr>
<tr><td>Structured text (ST)</td><td>HOME_HInst(
EN_R:=VarBOOL1,
Axis:=1,
DIR:=0,
Mode:=16#0A,
VMin:=500,
VMax:=2000,
ACCTime:=500,
DOG:=2,
Origin:=1,
Llimit:=0,
Rlimit:=0,
Locate:=0,
INT_EN:=0,
Active=> VarBOOL2,
Error=> VarErr);</td></tr>
<tr><td>Continuous function diagram (CFC)</td><td>
</td></tr>
</table>

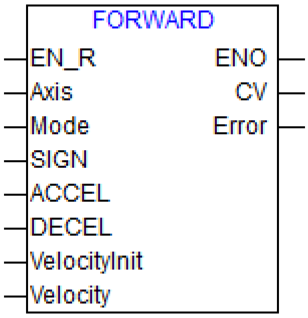
Program Instruction:

When EN_R is set and held, the axis 1(Q0.0) of the servomotor performs the origin search in the positive direction according to the pulse + direction mode. The search mode follows the work mode 2, method 2 of original point test and Invert mode 1 (refer to the related content in this Chapter for details) , Which crawling speed is 500Hz, the return speed is 2000Hz, acceleration and deceleration time is 500ms. The near-origin pulse input is provided by I0.1 and the origin pulse input is provided by I0.0.

3.5.1.24 FORWARD (Repeat Pulse Output,5128)

3.5.1.24.1 Description of Function

When FORWARD instruction enables, high-speed pule output will always output according to the value set by pulse until CANCEL or STOP instruction can make the pulse output stop. When starting the mode enable first, default minimum frequency 10Hz. The output frequency Velocity, ACCEL, DECEL of function block supports online modification, other parameters don’t support online modification.



FORWARD Function Block

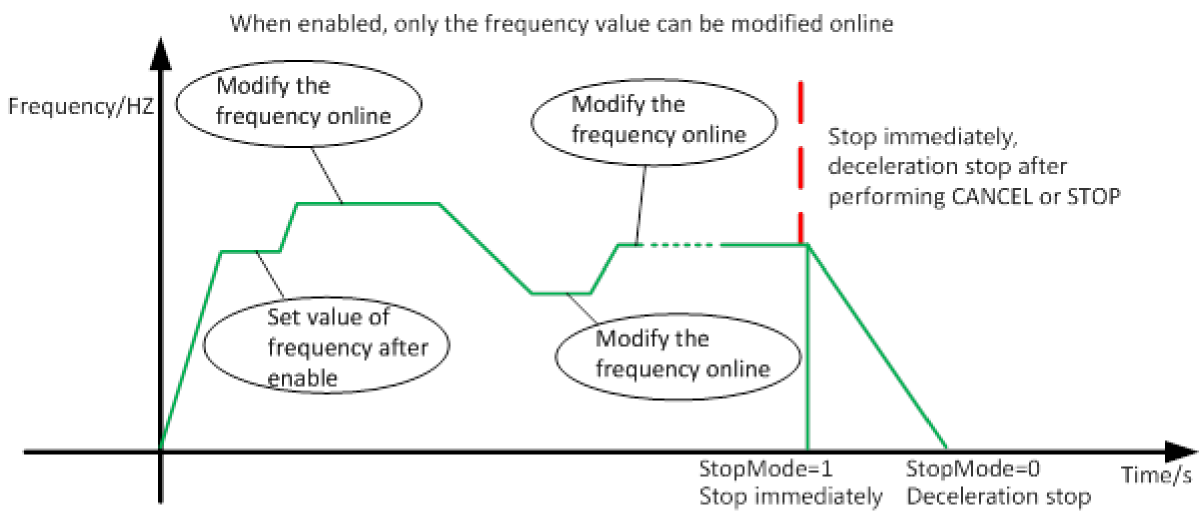
3.5.1.24.2 Description of Parameter

Input Parameter	Type of Data	Description of Function	Instruction of Parameter Value
EN_R	BOOL	Enable	0→1 Rising edge enable, high level keep
Axis	BYTE	Output axis selection	PLS+SIGN mode =5 : Output axis 5, pulse output end Q0.0, pulse direction Q0.4 =6: Output axis 6, pulse output end Q0.1, pulse direction Q0.5 =7: Output axis 7, pulse output end Q0.2, pulse direction Q0.6 =8: Output axis 8, pulse output end Q0.3, pulse direction Q0.7 CW/CCW mode =5, 7 Separately show output axis5 (Q0.0, Q0.1), axis 7 (Q0.2, Q0.3) If output 6 response to axis 5 output, if output 8 responses to axis 7 output
Mode	BOOL	Pulse mode control	0: PLS+SIGN Mode, 1: CW/CCW Mode
SIGN	BOOL	Direction	0: forward, 1: backward
ACCEL	DWORD	Acceleration	0~5,000,000 Hz/s,This parameter represents the acceleration after the first start and the acceleration of modified output frequency every time (0 represents 10Hz uniform speed)
DECEL	DWORD	Deceleration	0~5,000,000 Hz/s, This parameter represents the deceleration after the first start and the deceleration of modified output frequency every time (0 represents 10Hz uniform speed)

Input Parameter	Type of Data	Description of Function	Instruction of Parameter Value
VelocityInit	DWORD	Initial output frequency	10 Hz～100,000 Hz (This parameter means the initial frequency after the first time start)
Velocity	DWORD	Output frequency	10 Hz～100,000 Hz

Output Parameter	Type of Data	Description of Function	Instruction of Parameter Value
CV	DINT	Number of pulse that have been sent	-1,073,741,824～1,073,741,823
Error	BYTE	Error information	=0: Correct =1: The output axis is occupied =2: Configured output axis error =3: Configured output frequency error =4: Acceleration and deceleration setting error =5: Initial frequency range error =6: The initial frequency is greater than the target frequency

3.5.1.24.3 Description of Instruction



Online Modified Frequency Value

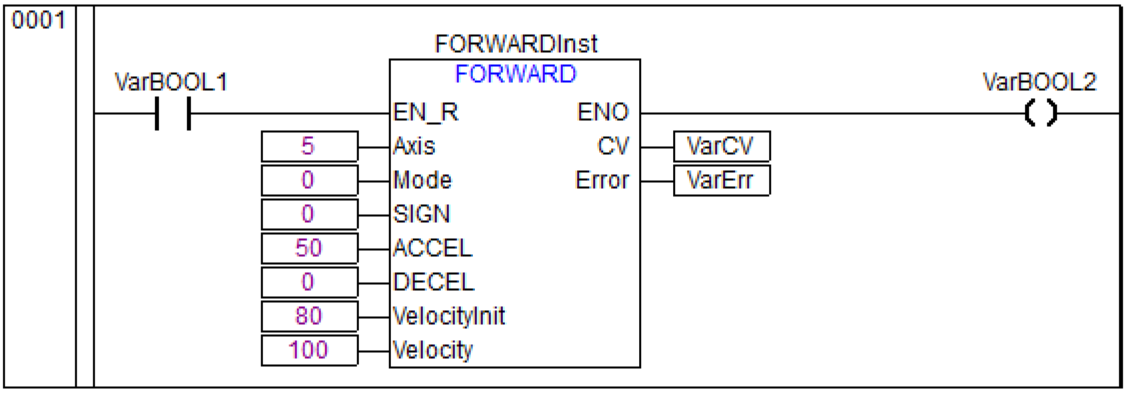
3.5.1.24.4 Application Example of Instructions

Variable Definition

No.	Variable Name	Address	Variable Description	Variable Type	Initial Value	Power Fail Safeguard
0001	VarBOOL1			BOOL	FALSE	FALSE
0002	VarBOOL2			BOOL	FALSE	FALSE
0003	VarCV			DINT	0	FALSE
0004	VarErr			BYTE	0	FALSE
0005	FORWARDInst			FORWARD		FALSE

Programing language	Program

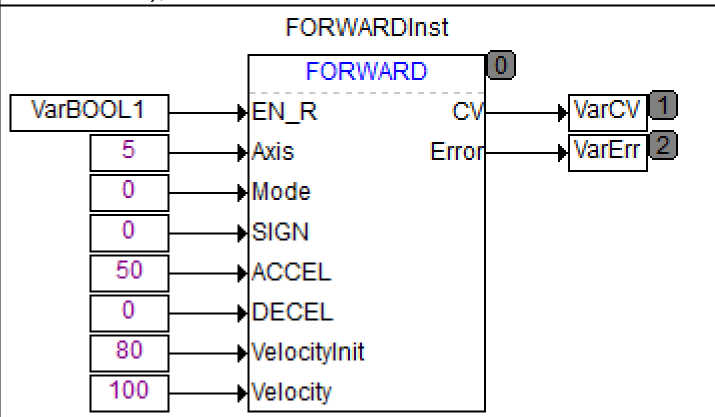
Ladder diagram (LD)



Structured text (ST)

```
FORWARDInst(
EN_R:=VarBOOL1,
Axis:=5,
Mode:=0,
SIGN:=0,
ACCEL:=50,
DECEL:=0,
VelocityInit:=80,
Velocity:=100,
CV=>VarCV,
Error=>VarErr);
```

Continuous function diagram (CFC)



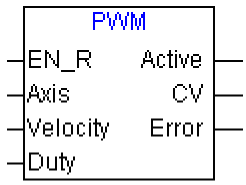
Program Instruction:

When EN_R is set and held, the axis 5 is output according to frequency of 100 Hz and acceleration of 50 Hz / s.

3.5.1.25 PWM (Pulse-Width Modulation, 5104/5106/5108/5128/5118)

3.5.1.25.1 Description of Function

Output the pulse whose frequency pointed by Velocity and duty ratio pointed by Duty from the end pointed by Axis.



PWM Function Block

3.5.1.25.2 Description of Parameter

Input Parameter	Type of Data	Description of Function	Instruction of Parameter Value
EN_R	BOOL	Enable	0→1 Rising edge enable, high level keep
Axis	BYTE	Output axis selection	LE5104/LE5106/LE5108/LE5118 =1: Output axis 1, pulse output end Q0.0 =2: Output axis 2, pulse output end Q0.1 =3: Output axis 3, pulse output end Q0.2 =4: Output axis 4, pulse output end Q0.3 5128 =5, 6, 7, 8 Separately show output axis is : axis 5, axis 6, axis 7, axis 8, =5: Output axis 5, pulse output end Q0.0 =6: Output axis 6, pulse output end Q0.1 =7: Output axis 7, pulse output end Q0.2 =8: Output axis 8, pulse output end Q0.3
Velocity	DWORD	Output frequency	10 Hz～100,000 Hz Can be modified online
Duty	WORD	Duty ratio	Duty ratio input value :0~1000 (corresponding 0~100.0) Duty ratio resolution ratio: When 10 Hz≤output frequency≤1000 Hz: 1/1000 When 1000 Hz<output frequency≤10,000 Hz:1/100 When 10,000 Hz<output frequency≤100,000 Hz: 1/10 Can be modified online

Output Parameter	Type of Data	Description of Function	Instruction of Parameter Value
Active	BOOL	Mark of sending pulse	0: Stop sending 1:On the process of sending
CV	DWORD	Number of pulse that have been sent	0~2,147,483,647
Error	BYTE	Error information	=0: Correct =1: The channel is taken up =2: The configuration axis error =3: Frequency output error =4: Duty ratio set error

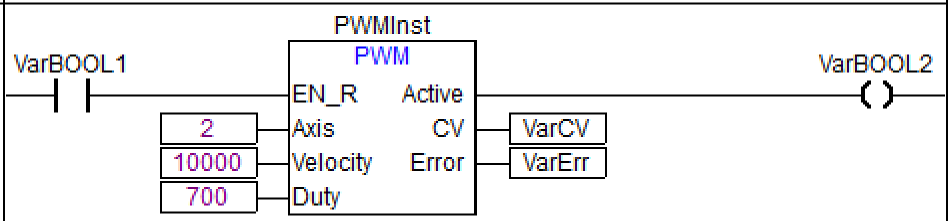
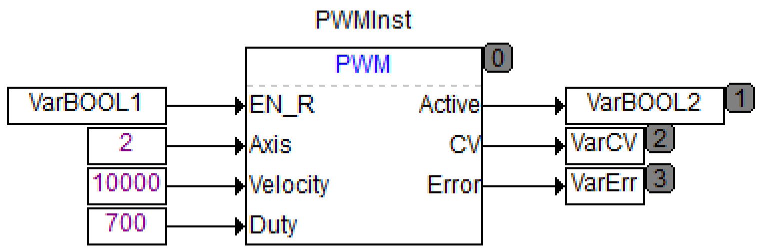
3.5.1.25.3 Description of instruction

- When function block enable, output frequency and duty ratio can be modified online.
- When duty ratio is set to 0, it is low level all the time; when it is set to 1000, it is high level all the time.
- PWM doesn’t influence axis absolute position output of the special register in S area.

3.5.1.25.4 Application Example of Instructions

Variable Definition

No.	Variable Name	Address	Variable Description	Variable Type	Initial Value	Power Fail Safeguard
0001	PWMInst			PWM		FALSE
0002	VarBOOL1			BOOL	FALSE	FALSE
0003	VarBOOL2			BOOL	FALSE	FALSE
0004	VarCV			DWORD	0	FALSE
0005	VarErr			BYTE	0	FALSE

<table>
<tr><td>Programing language</td><td>Program</td></tr>
<tr><td>Ladder diagram (LD)</td><td>
</td></tr>
<tr><td>Structured text (ST)</td><td>PWMInst(
EN_R:=VarBOOL1,
Axis := 2,
Velocity :=10000,
Duty :=700,
Active =>VarBOOL2,
CV =>VarCV,
Error=> VarErr);</td></tr>
<tr><td>Continuous function diagram (CFC)</td><td>
</td></tr>
</table>

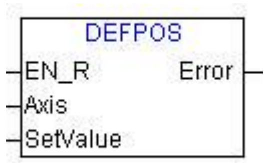
Program Instruction:

When EN_R is set and held, the 2nd high-speed pulse output sends pulses in 10 kHz frequency with a duty cycle of 70.0.

3.5.1.26 DEFPOS (Define,5128)

3.5.1.26.1 Description of Function

DEFPOS is used to define the current location of pointed axis, and record the current location of axis in the special register area to change the location of original point.



DEFPOS Function Block

3.5.1.26.2 Description of Parameter

Input Parameter	Type of Data	Description of Function	Instruction of Parameter Value
EN_R	BOOL	Enable	0→1:Rising edge enable, high level is valid
Axis	BYTE	Axis number selection	=1,2,3,4,5,6,7,8 separately shows pointed axis : axis 1 to axis 8
SetValue	DINT	Absolute location set value	-1,073,741,824～1,073,741,823

Output Parameter	Type of Data	Description of Function	Instruction of Parameter Value
Error	BYTE	Error information	=0: Correct =1: The channel is taken up =2: Configuration axis error

3.5.1.26.3 Description of Instruction

When input and output location changes, the below special register records the absolute location of corresponding axis at real time. Except PWM, the whole enable work of high-speed output instruction can change the absolute location value of corresponding register of output axis.

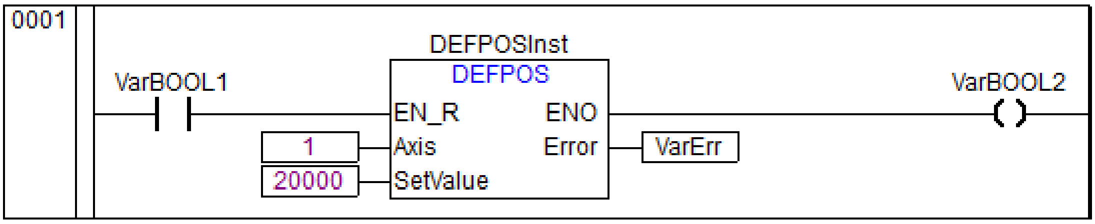
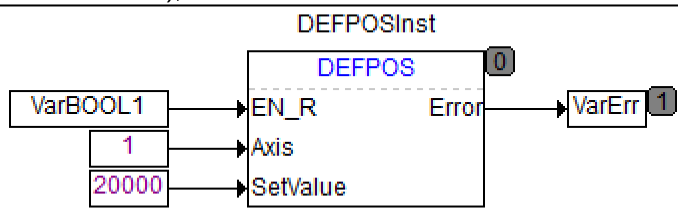
Axis Number	1	2	3	4	5	6	7	8
Special register	%SD108	%SD112	%SD116	%SD120	%SD140	%SD144	%SD148	%SD152

When axis1 and axis 3 count for AB phase, the corresponding counting value of axis2, and axis 4 do not accumulate data. When axis 5 and axis 7 are CW/CCW pulse output mode, the corresponding counting value of axis 6 and axis 8 do not accumulate data.

3.5.1.26.4 Application Example of Instructions

Variable Definition

No.	Variable Name	Address	Variable Description	Variable Type	Initial Value	Power Fail Safeguard
0001	VarBOOL1			BOOL	FALSE	FALSE
0002	VarBOOL2			BOOL	FALSE	FALSE
0003	DEFPOSInst			DEFPOS		FALSE
0004	VarErr			BYTE	0	FALSE

<table>
<tr><td>Programing language</td><td>Program</td></tr>
<tr><td>Ladder diagram (LD)</td><td>
</td></tr>
<tr><td>Structured text (ST)</td><td>DEFPOSInst(
EN_R:=VarBOOL1,
Axis:=1,
SetValue:=20000,
Error=>VarERR);</td></tr>
<tr><td>Continuous function diagram (CFC)</td><td>
</td></tr>
</table>

Program Instruction:

When EN_R is set and held, the setpoint of absolute position is assigned to the corresponding special register as the current absolute position.

3.5.1.27 CANCEL (Cancel,5128)

3.5.1.27.1 Description of Function

Instruction cancels any connect relationship between current pointed axis and other axis, and stop.

CANCEL Function Block

3.5.1.27.2 Description of Parameter

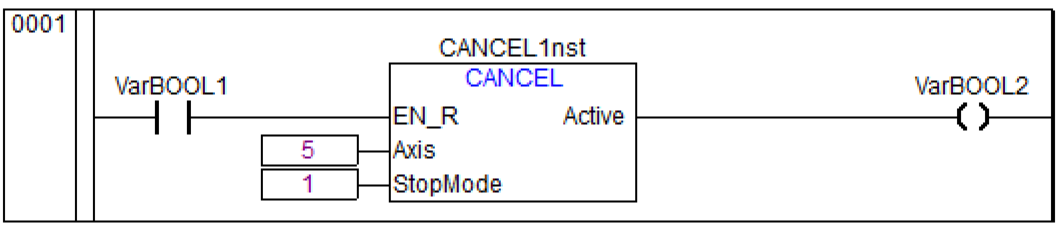
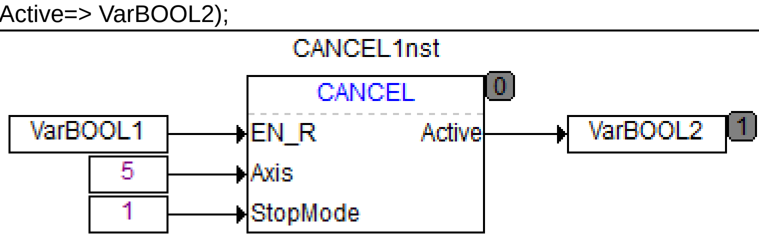
Input Parameter	Type of Data	Description of Function	Instruction of Parameter Value
EN_R	BOOL	Enable	0→1: Rising edge enable, high level is valid
Axis	BYTE	Stop axis selection	=5,6,7,8 Current axis : PLS+SIGN mode Separately shows stop axis : axis 5, axis 6, axis 7, axis 8 Current axis : CW/CCW mode 5, 6 show stop axis 5 (Q0.0, Q0.1); 7, 8 shows stop axis 7 (Q0.2, Q0.3)
StopMode	BOOL	Stop mode	=0: Achieve stop according to the set deceleration rate or deceleration rate related to deceleration time =1: Immediately stop output

Output Parameter	Type of Data	Description of Function	Instruction of Parameter Value
Active	BOOL	Mark of stop completion	=1: Stop completion =0: On the process of stop

3.5.1.27.3 Application Example of Instructions

Variable Definition

No.	Variable Name	Address	Variable Description	Variable Type	Initial Value	Power Fail Safeguard
0001	VarBOOL1			BOOL	FALSE	FALSE
0002	VarBOOL2			BOOL	FALSE	FALSE
0003	CANCEL1nst			CANCEL		FALSE

Programing language	Program
Ladder diagram (LD)	
Structured text (ST)	CANCELInst(EN_R:=VarBOOL1, Axis:=5, StopMode:=1, Active=> VarBOOL2);
Continuous function diagram (CFC)	

Program Instruction:

When EN_R is set and held, axis 5 stops output immediately.

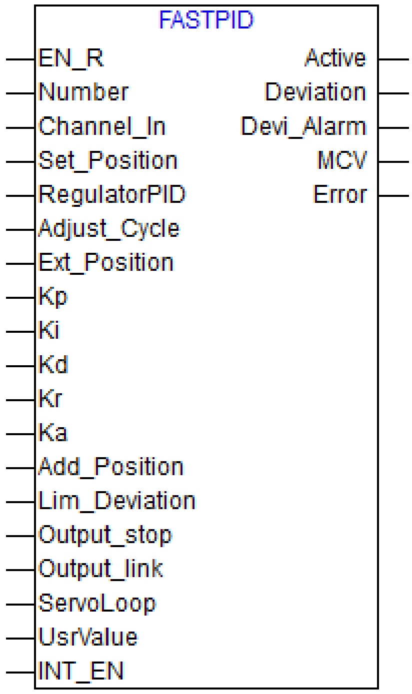
3.5.1.27.4 Special Instruction

When the two function block STOP_M, CANCEL stop deceleration (StopMode=0), they achieve stop according to set deceleration rate and speed rate (deceleration time equals to deceleration rate).

3.5.1.28 FASTPID (Fast PID, 5128)

3.5.1.28.1 Description of Function

Comparing the process value returned from site and set target value, adjusting and controlling PID, and output the adjustment value to facilities on site to reach the task of calling to adjustment in time, and operate the amplitude limit for the output value to ensure the output works within the permit range.



FASTPID Function Block

3.5.1.28.2 Description of Parameter

Input Parameter	Type of Data	Description of Function	Instruction of Parameter Value
EN_R	BOOL	Enable	0→1: Rising edge enable, high level is valid
Number	BYTE	Current PID function block No.	=1,2,3,4 Separately shows : 1st, 2nd, 3rd, 4th way analogy quantity output

Input Parameter	Type of Data	Description of Function	Instruction of Parameter Value
Channel_In	BYTE	Feedback input channel (Note: the mode only supports positive analogy quantity input)	=1,2,3,4 Separately shows : 1st, 2nd, 3rd, 4th way digital quantity input =5,6 Separately shows : 1st, 2nd, way analogy quantity input =7 Separately shows : outer algorithm input
Set_Position	DINT	Set target value	Data range-integer of -1,073,741,824～1,073,741,823
RegulatorPID	BYTE	Superposition of PID No.	0 shows no superposition; X shows superposition of Xth PID adjustment value, X is one of 1,2,3,4
Adjust_Cycle	BYTE	PID adjustment period	=1~50 ms can be set Adjustment error 0.1ms. The adjustment period is : Channel input time +PID calculation time +AO output time
Ext_Position	DINT	Feedback comes from outer algorithm	Data range-integer of -1,073,741,824～1,073,741,823
Kp	REAL	Proportionality coefficient	0～63.999
Ki	REAL	Integral coefficient	0～63.999
Kd	REAL	Differential coefficient	0～63.999
Kr	REAL	Feedforward coefficient	0～63.999
Ka	REAL	Superposition coefficient	1/4095～4095
Add_Position	DINT	PID adjustment value extra increment	Data range-integer of -1,073,741,824～1,073,741,823
Lim_Deviation	DWORD	Deviation value amplitude limit	0~2,147,483,647
Output_stop	BOOL	Whether stop output if deviation is too big	0: Disconnection ; 1: Connection
Output_link	BOOL	Whether output to the operation office	0: Connection ; 1: Disconnection
ServoLoop	BOOL	Closed-loop output switch	0: PID output open-loop, MCV output sum of UsrValue and Add_Position 1: PID output closed-loop, MCV calculates output value according to feedback quantity
UsrValue	WORD	Output value pointed by users	Input value of users, when ServoLoop=0, it is valid
INT_EN	BOOL	Whether to stop if deviation quantity is too big	0:Not trigger stop ; 1: Trigger stop

Output Parameter	Type of Data	Description of Function	Instruction of Parameter Value
Active	BOOL	Mark of successful set of location value	0:Location set failure; 1: Location set success
Deviation	DINT	Deviation value	Integral in the data range
Devi_Alarm	BOOL	Alarm if deviation is too big	0: Deviation value normal; 1: Deviation value over set value
MCV	DWORD	Controlling quantity output	0～65,535
Error	BYTE	Error information	=0: Correct =1: Mark of the channel being taken up =2: Function Block No. error =3: Feedback input channel error =4: PID adjustment period set error =5: PID adjustment coefficient set error =6: Superposition coefficient Ka set error

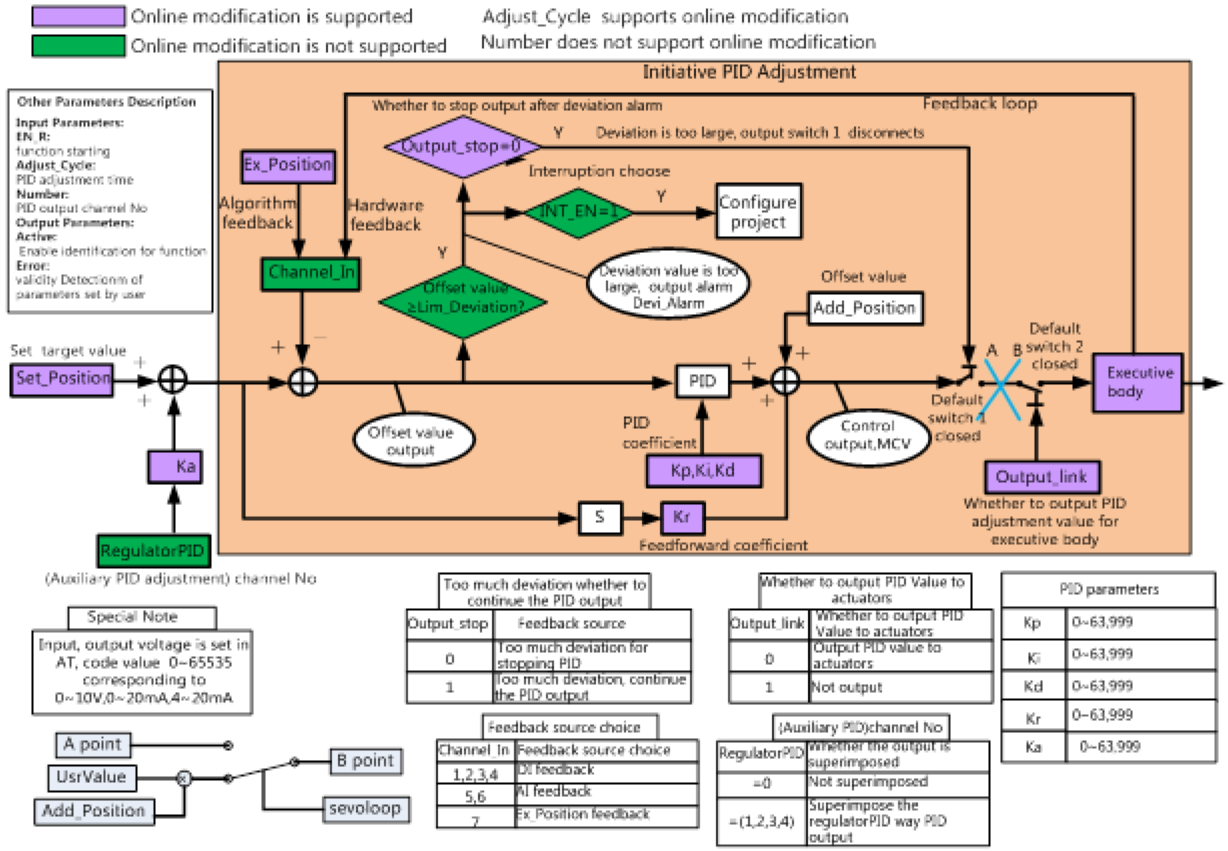
Output Parameter	Type of Data	Description of Function	Instruction of Parameter Value
			=7: Analogy quantity channel is forbidden =8: 31 bits input data set error =9: Superposition PID No. set error

3.5.1.28.3 Description of Instruction

The function supports the calculation function that can be set by adjustment period. The number of PID is 4, each PID can singly set whether to start or not.

- The adjustment period of PID shall calculate the whole started PID for a time within a adjustment period, the accuracy and difference of adjustment period shall be within 0.1ms. For example : set to start the 3 PID, adjustment period is 1ms, then calculate the three PID for a time within 1ms.
- PID parameter can be adjusted online at real time. The adjusted parameter can be saved in engineering documents automatically.
- PID algorithm index :
  - Multiple PID can set the same target parameter at the same time. Multiple PID function independent from each other;
  - There are two physics origins of PID feedback: one comes from analog quantity input, the other comes from high-speed pulse counting channel.
  - PID feedforward adjustment algorithm : variable of value set by target multiply by feedforward coefficient , and make final superposition output of PID algorithm results;
  - Within the set adjustment period, superposition of extra offset and PID algorithm results;
  - Within set adjustment period, superposition output of two PID algorithm results;
  - Too big deviation of PID trigger stop, after too big deviation of PID trigger stop, defaulting stop PID adjustment output, one can choose not to stop and only call for police for deviation.
- When ServoLoop=0, PID doesn’t take part in calculation, output value is UsrValue+ Add_Position, and deal with amplitude limit. When the result is more than 65,535, outout 65,535; when the result is less than 0, output is 0.
- Output_link control the output of PID, when Output_link=0, PID only make calculation, and show the result on MCV. When Output_link=1, beside PID calculation shows the result on MCV, and output from AO.
- Four ways PID corresponding to four ways AO output, AO channel is bind with PID channel No. AO output renew a time per calculation period.
- When operating PID, the AO data in Q area doesn’t change, AO output is not in accordance with output in Q area.
- When firstly selecting feedback signal, one shall use high-speed counting function. High-speed counting CV value as feedback of PID calculation to take part in calculation.

PID function frame is as shown in Figure.



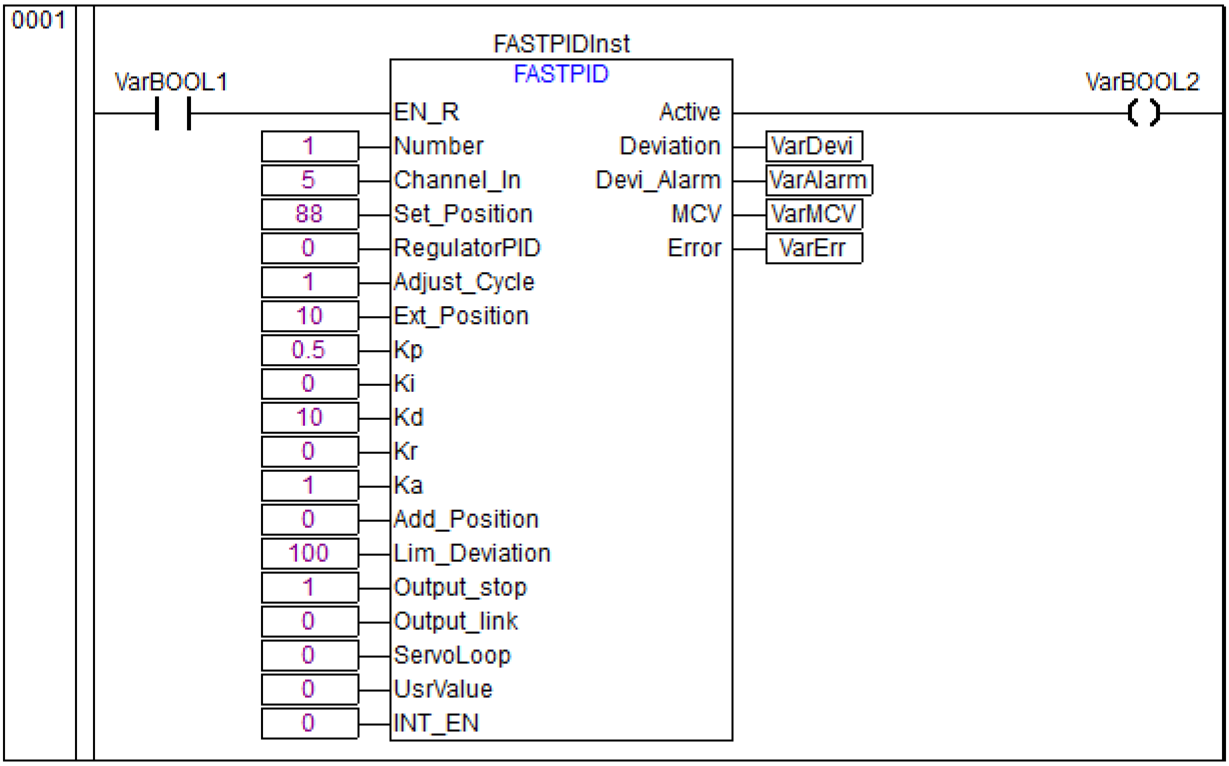
PID Function Frame Diagram

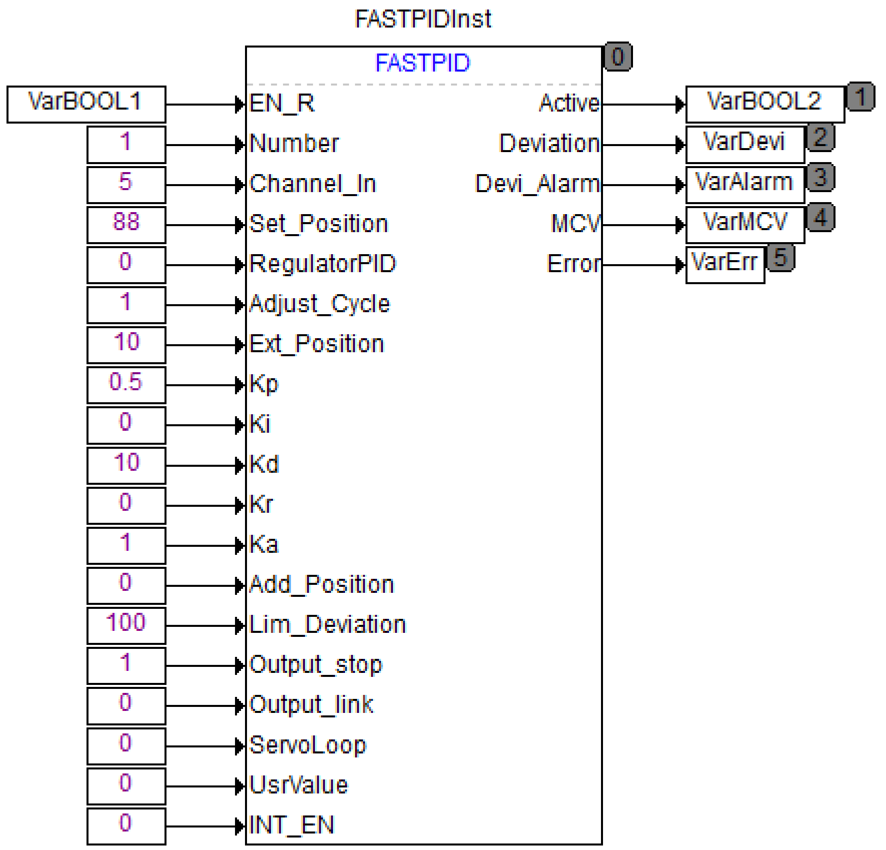
3.5.1.28.4 Application Example of Instructions

Variable Definition

No.	Variable Name	Address	Variable Description	Variable Type	Initial Value	Power Fail Safeguard
0001	VarBOOL1			BOOL	FALSE	FALSE
0002	VarBOOL2			BOOL	FALSE	FALSE
0003	VarDevi			DINT	0	FALSE
0004	VarAlarm			BOOL	FALSE	FALSE
0005	VarMCV			DWORD	0	FALSE
0006	VarErr			BYTE	0	FALSE
0007	FASTPIDInst			FASTPID		FALSE

Programing language	Program

Variable Definition	
Ladder diagram (LD)	
Structured text (ST)	FASTPIDInst(EN_R:=VarBOOL1, Number:=1, Channel_In:=5, Set_Position:=88, RegulatorPID:=0, Adjust_Cycle:=1, Ext_Position:=10, Kp:=0.5, Ki:=0, Kd:=10, Kr:=0, Ka:=1, Add_Position:=0, Lim_Deviation:=100, Output_stop:=1, Output_link:=0, ServoLoop:=0, UsrValue:=0, INT_EN:=0, Active=>VarBOOL2, Deviation=>VarDevi, Devi_Alarm=>VarAlarm, MCV=>VarMCV, Error=> VarErr);

<table>
<tr><th colspan="2">Variable Definition</th></tr>
<tr><td>Continuous function diagram (CFC)</td><td>
</td></tr>
</table>

3.5.1.29 LATCH (Latch Data, 5128)

3.5.1.29.1 Description of Function

During starting the function, receiving rising edge (or Falling edge) of switch quantity signal, getting the total pulse from reference zero point to that are sent currently (or receive). It can support 4 Latch function block enable at most and the same trigger point can latch location of different axis.

LATCH Function Block

3.5.1.29.2 Description of Parameter

Input Parameter	Type of Data	Description of Function	Instruction of Parameter Value
EN_R	BOOL	Enable	0→1: Rising edge enable, high level is valid
Number	BYTE	LATCH No.	1,2,3,4
Axis	BYTE	Pointed latch axis	=1,2,3,4,5,6,7,8
INT_SIG	BYTE	Latch point input selection	=1,2,3,4,5,6 show I0.0～I0.5
SEL_WIN	BYTE	Window selection	Constrain validity of INT_SIG hardware parameter 0: Not use window 1: Valid within window 2: Valid outside the window
OPEN_WIN	DINT	Window 1	-1,073,741,824～1,073,741,823
CLOSE_WIN	DINT	Window 2	-1,073,741,824～1,073,741,823
SIG_EDG	BOOL	INT_SIG valid edge selection	0: Rising edge valid 1: Falling edge valid

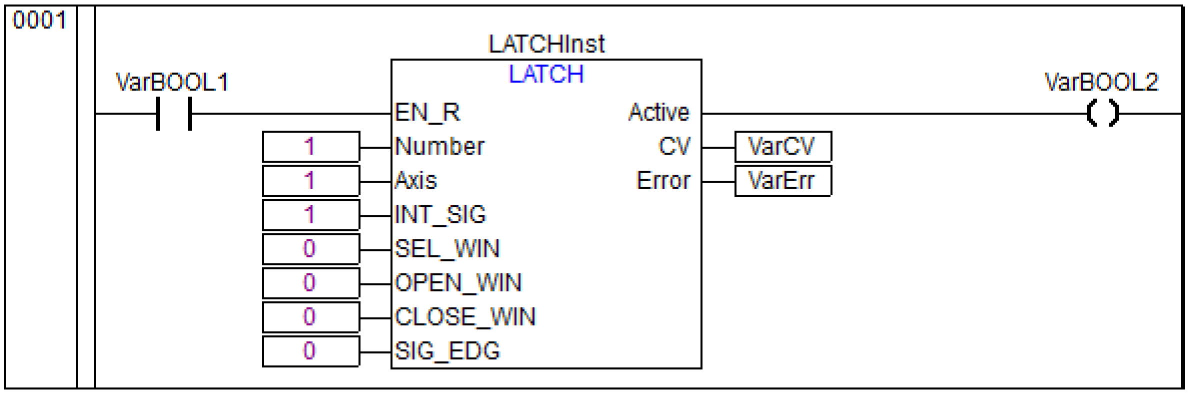
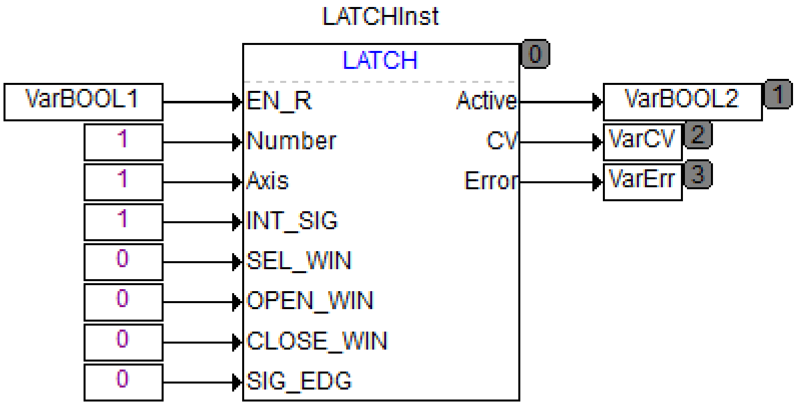
Output Parameter	Type of Data	Description of Function	Instruction of Parameter Value
Active	BOOL	Mark of successful set	0: Doesn't set successfully or complete catching 1: Succeed in setting catching parameter
CV	DINT	Data latch value	-1,073,741,824～1,073,741,823
Error	BYTE	Error information	=0: Correct =1:The channel is taken up =2: Latch axis selection error =3: Interruption input point selection error =4: LATCH No. set error =5: 31 bits data set error

3.5.1.29.3 Application Example of Instructions

Variable Definition

No.	Variable Name	Address	Variable Description	Variable Type	Initial Value	Power Fail Safeguard
0001	LATCHInst			LATCH		FALSE
0002	VarBOOL1			BOOL	FALSE	FALSE
0003	VarBOOL2			BOOL	FALSE	FALSE
0004	VarCV			DINT	0	FALSE
0005	VarErr			BYTE	0	FALSE

Programing language	Program

Variable Definition	
Ladder diagram (LD)	
Structured text (ST)	LATCHInst(EN_R:=VarBOOL1, Number:=1, Axis:=1, INT_SIG:=1, SEL_WIN:=0, OPEN_WIN:=0, CLOSE_WIN:=0, SIG_EDG:=0, Active=> VarBOOL2, CV=>VarCV, Error=> VarErr);
Continuous function diagram (CFC)	

Program Instruction:

When EN_R is set and held, the total number of pulses is output when the rising edge is enabled by IX0.0 latch input.

3.5.1.29.4 Description of Instruction

- Requirement of precision: the difference between the moment that switch quantity signal INT_SIG triggers rising edge (or Falling edge) and the pulse moment of latch axis is less than 10 microseconds.
- Set latch relationship between switch quantity signal and axis.
- When function block enables, if it triggers switch quantity signal for many times, it is subject to the first valid trigger switch quantity signal.
- The interruption input channel can’t correspond to the same hardware channel with the latch axis. If

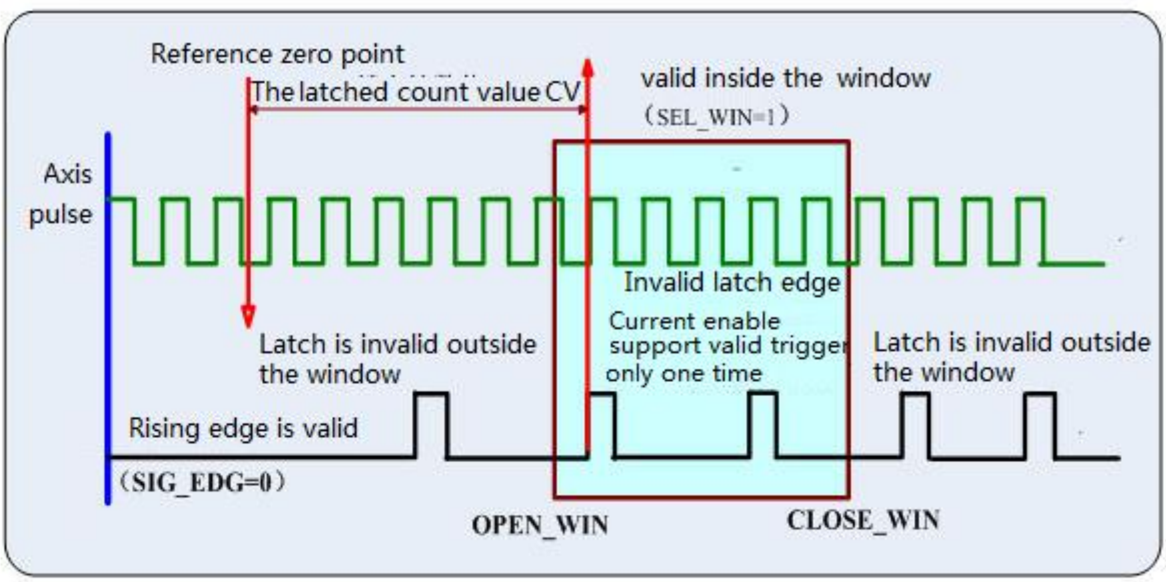
Axis=1 (I0.0), INT_SIG=1 (I0.0) error.

For example: as show in follows figure .

SEL_WIN=1: INT_SIG signal is valid within window.

SIG_EDG=0: the rising edge of INT_SIG signal is valid.

Description: under the current enable, the rising edge of first INT_SIG within the window is valid.

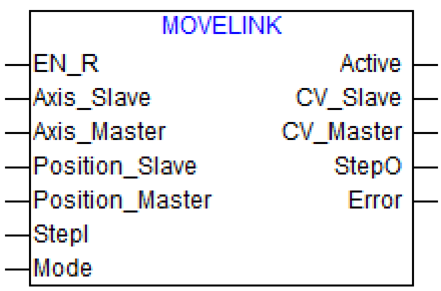


Examples of Instruction

3.5.1.30 MOVELINK (MoveLink, 5128)

3.5.1.30.1 Description of Function

The function build a segmentation linear proportion relationship between axis 1 (following axis) and axis 2 (driving axis) so that slave axis moves following the master axis.



MOVELINK Function Block

3.5.1.30.2 Description of Parameter

Input Parameter	Type of Data	Description of Function	Instruction of Parameter Value
EN_R	BOOL	Connect enable	0→1: Rising edge enable, high level is valid
Axis_Slave	BYTE	Pointed following axis	=5 : Output axis 5, pulse output end Q0.0, pulse direction Q0.4 =6: Output axis 6, pulse output end Q0.1, pulse direction Q0.5 =7: Output axis 7, pulse output end Q0.2, pulse direction Q0.6 =8: Output axis 8, pulse output end Q0.3, pulse direction Q0.7
Axis_Master	BYTE	Pointed driving axis	=1,2,3,4,5,6,7,8 Separately shows set value : axis1, axis 2...axis 7, axis 8
Position_Slave	ARRAY[50] OF DINT	Following axis location set value	-1,073,741,824～1,073,741,823
Position_Master	ARRAY[50] OF DINT	Main axis location set value	-1,073,741,824～1,073,741,823
StepI	BYTE	The maximum valid steps of following axis	1～50
Mode	BOOL	Operation method	0: Single run 1: Cycle run

Output Parameter	Type of Data	Description of Function	Instruction of Parameter Value
Active	BOOL	Mark of successful set	0: Set failure 1: Set success
CV_Slave	DINT	Current location value of following axis	-1,073,741,824～1,073,741,823
CV_Master	DINT	Current location value of main axis	-1,073,741,824～1,073,741,823
StepO	BYTE	Current steps of following axis	1～50
Error	BYTE	Error information	=0: Correct =1: Mark of output channel is taken up =2: Pointed following axis error =3: Pointed main axis error =4: Valid steps set error =5: Set main and following axis are same =6: Main axis data within valid steps is zero =7: Data set of main axis and following axis within valid steps error

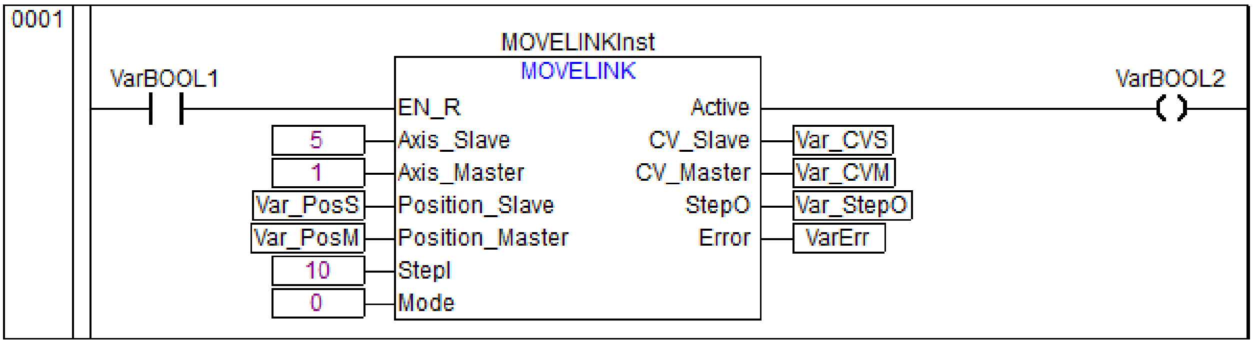
3.5.1.30.3 Application Example of Instructions

Variable Definition

No.	Variable Name	Address	Variable Description	Variable Type	Initial Value	Power Fail Safeguard
0001	Var_PosS			ARRAY[0..49] OF DINT		FALSE
0002	Var_PosM			ARRAY[0..49] OF DINT		FALSE
0003	Var_CVS			DINT	0	FALSE
0004	Var_CVM			DINT	0	FALSE
0005	Var_StepO			BYTE	0	FALSE
0006	VarErr			BYTE	0	FALSE
0007	MOVELINKInst			MOVELINK		FALSE
0008	VarBOOL2			BOOL	FALSE	FALSE
0009	VarBOOL1			BOOL	FALSE	FALSE

Programing language	Program

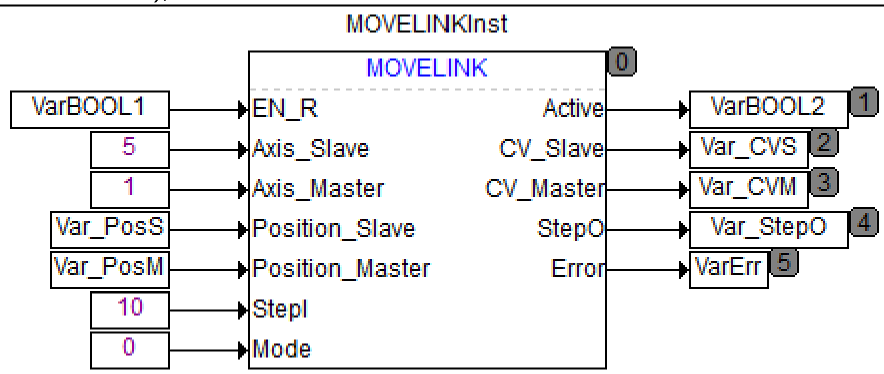
Ladder diagram (LD)



Structured text (ST)

```
MOVELINKInst(
EN_R:= VarBOOL1,
Axis_Slave:=5,
Axis_Master:=1,
Position_Slave:=Var_PosS,
Position_Master:=Var_PosM,
StepI:=10,
Mode:=0,
Active=> VarBOOL2,
CV_Slave=>Var_CVS,
CV_Master=>Var_CVM,
StepO=> Var_StepO,
Error=> VarErr);
```

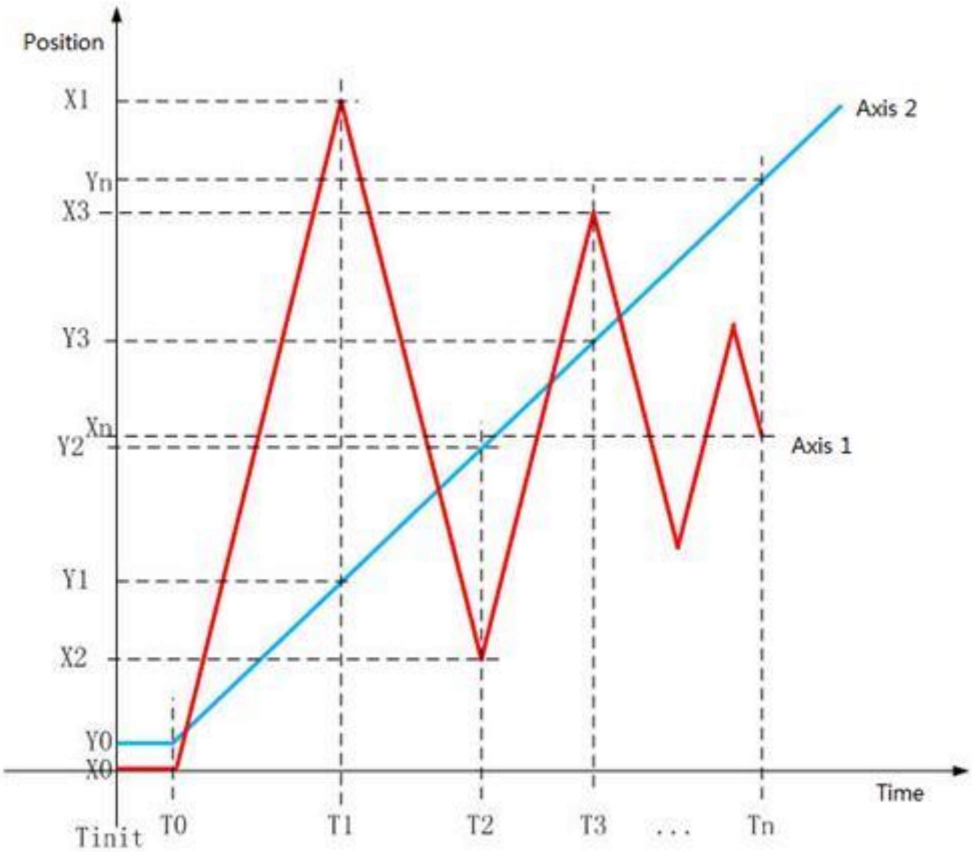
Continuous function diagram (CFC)



3.5.1.30.4 Description of Instruction

Through the function, user cam re-set the current location value of following axis (axis 1), and can re-set current value of driving axis (axis 2). Achieve the flexible change of movement control axis reference zero point.

Example: show as the following Figure.



Location Connection Diagram of Winding Machine

- Tinit moment, through instruction, building the connection between axis 1 and axis 2, axis begins to follow axis 2. But the axis 2 is in state of rest at this moment, so the axis is in state of rest too.
- T0 moment: axis 2 begins to move. Axis 1 also moves with axis 2 according to the reserved relationship.
- T1 moment, axis 2 moves to Y1 location, axis 1 shall move to X1 location.
- T2 moment, axis 2 moves to Y2 location, axis 1 shall move to X2 location.
- T3 moment, axis 2 moves to Y3 location, axis 1 shall move to X3 location.
- Tn moment, axis 2 moves to Yn location, axis 1 shall move to Xn location.

2-d table describing the movement relationship of two axis.

The value of X1~Xn, Y1~Yn shall be filled by users, and can be adjusted online at real time, or assignment via other algorithm.

In the below 2-d table, the value of Xn, Yn set by users are the increment of Xn-1, Yn-1.

If setting x1=10, x2=20, then shows:

The first step: axis 1 moves forward 10 pulse from the current location,

The second step: axis 1 moves backward 20 pulse from the current location.

The maximum of n is 50.

Step	Axis 1	Axis 2
1	X1	Y1
2	X2	Y2
3	X3	Y3
...	...	...
n	Xn	Yn

The greatest valid steps in Table. For example: step=2, the valid step in Table is front 2 steps. Steps=3, the valid step in Table is front 3 steps.

- Mode: operation method. Providing two operation methods: single run, cycle run.
  - Single run: operate to step and stop automatically with canceling the connection of the two axes automatically.
  - Cycle run: operate to n, and return to the first line of table to operate from the beginning. And only can stop movement until meeting ‘canceling connection’ instruction.

3.5.1.31 CONNECT (Electronic Gear,5128)

3.5.1.31.1 Description of Function

The target location of current axis connects with the measuring location defined by connection axis through electric gear.

CONNECT Function Block

3.5.1.31.2 Description of Parameter

Input Parameter	Type of Data	Description of Function	Instruction of Parameter Value
EN_R	BOOL	Enable	0→1: Rising edge enable, high level is valid
Axis_slave	BYTE	Pointed following	=5 : Output axis 5, pulse output end Q0.0, pulse direction Q0.4

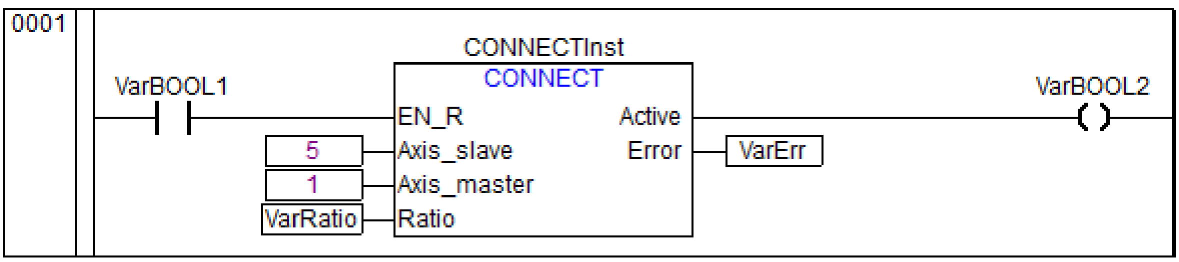
Input Parameter	Type of Data	Description of Function	Instruction of Parameter Value
		axis	=6: Output axis 6, pulse output end Q0.1, pulse direction Q0.5 =7: Output axis 7, pulse output end Q0.2, pulse direction Q0.6 =8: Output axis 8, pulse output end Q0.3, pulse direction Q0.7
Axis_master	BYTE	Pointed driving axis	=1, 2, 3, 4, 5, 6, 7, 8 Separately shows set value : axis1, axis 2...axis 7, axis 8
Ratio	REAL	Set ratio	The ratio of electric gear, ratio can be changed at any time, there is no need to cancel the past ratio. Range of value (-1000.000~1000.000)

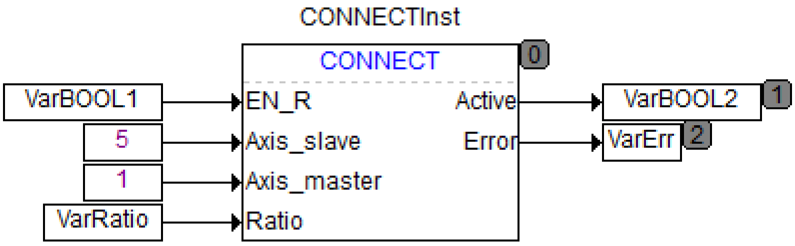
Output Parameter	Type of Data	Description of Function	Instruction of Parameter Value
Active	BOOL	Mark of successful set	0: Set failure 1: Set success
Error	BYTE	Error information	=0: Correct =1: Mark of following axis is taken up =2: Pointed following axis error =3: Pointed main axis error =4: Set ratio error =5: The master and slave axis set are the same

3.5.1.31.3 Application Example of Instructions

Variable Definition

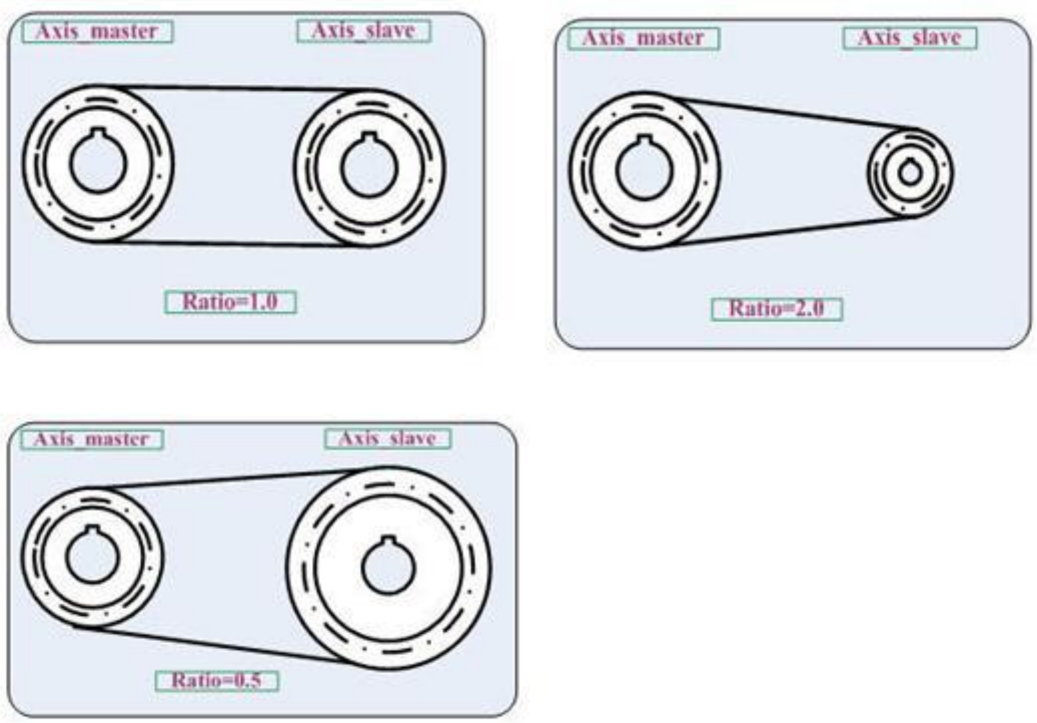
No.	Variable Name	Address	Variable Description	Variable Type	Initial Value	Power Fail Safeguard
0001	VarRatio			REAL	0	FALSE
0002	VarErr			BYTE	0	FALSE
0003	VarBOOL2			BOOL	FALSE	FALSE
0004	VarBOOL1			BOOL	FALSE	FALSE
0005	CONNECTInst			CONNECT		FALSE

Programing language	Program
Ladder diagram (LD)	
Structured text (ST)	CONNECTInst(EN_R:= VarBOOL1, Axis_slave:=5, Axis_master:=1, Ratio:=VarRatio, Active=>VarBOOL2, Error=>VarErr);

Variable Definition	
Continuous function diagram (CFC)	

3.5.1.31.4 Description of Instruction

Axis_master is the detected axis, Axis_master is output axis after detected rotating speed is multiplied by the gear ratio.



Ratio Relationship of Detected Axis and Output Axis

3.5.2 Fault Record Wave

3.5.2.1 MKC3167_IS_ENABLE_CHANNEL (Is or not enable IO board upload and ISSule channel ID)

3.5.2.1.1 Function

Implement the real-time data upload enable function to transmit the real-time data from the specified channels collected by the protection IO to the CPU module.

3.5.2.1.2 Description of Parameter

The current channel data acquisition and waveform recording function supports high-voltage side (KJG) scenarios and low-voltage dual circuit (BXJ2) scenarios. The specific correspondence between channel numbers and hardware channels is shown in the following table:

Input Parameter	Type of Data	Explanation of Parameter Values	Initial Value
EN_R	BOOL	Enable, rising edge effective	FALSE
ChannelID	BYTE	Channel number: 1-16, where 1 represents the first channel, 2 represents the second channel, and so on, up to a total of 16 channels.	0
IsEnable	BOOL	Enable IO board data reporting TRUE: enabled FALSE: disabled	FALSE
Output Parameter	**Type of Data**	**Explanation of Parameter Values**	**Initial Value**
Q	BOOL	Enable function block output, set to TRUE upon completion	FALSE
Error	BYTE	Error code 0: operation completed 1: channel ID Input Error	0

Correspondence between channel numbers and hardware channels

Hardware Channel	ChannelID	High-voltage Side (KJG)	Low-voltage Single Circuit (BXJ2)	Low-voltage Dual Circuit (BXJ2)
L/N	12	Voltage Usys	\	\
A1	11	\	Voltage	Voltage of the first circuit
A2	13	\	\	Voltage of the second circuit
A3	14	\	\	\
A4	1	\	\	\
A5	3	Current A	\	\
A6	5	Current C	\	\
A7	7	\	Current A	Current of the first circuit A
A8	8	\	Current C	Current of the first circuit C
A9	15	\	\	Current of the second circuit A
A10	16	\	\	Current of the second circuit C
RS	9	\	Leakage resistance RS	Leakage resistance RS

3.5.2.2 MKC3167_NORMAL_READ (Normal data read)

3.5.2.2.1 Function

To implement the function of reading real-time data from a specified channel and copying the real-time data from the protection IO uploaded to the CPU module's channel to a designated address.

- Precautions: : the MKC3167_NORMAL_READ function block can only be called once in the same configuration project, and it reads data from only one channel each time.

3.5.2.2.2 Description of Parameter

Input Parameter	Type of Data	Explanation of Parameter Values	Initial Value
In	BOOL	Enable, active high	FALSE
ChannelID	BYTE	The channel numbers for which real-time data needs to be acquired are 1-16, where 1 represents the first channel, 2 represents the second channel, and so on, up to a total of 16 channels.	0
DataLen	WORD	The data length, or address space size, should not exceed 120 sample points.	0
Address	POINTER TO WORD	The data storage address.	0
Output Parameter	**Type of Data**	**Explanation of Parameter Values**	**Initial Value**
Done	BOOL	10:49Execution Result:Set to 1 if the execution is successful.	FALSE
Busy	BOOL	TRUE: busy; FALSE: free	FALSE
Error	BYTE	Error code 0:operation completed ; 1: channel ID input error when reading; 2: illegal data length ; 3: illegal read address ; 4:channel upload function not enabled ;	0
RealDataLen	WORD	Data Length: Actual copied data length, which must not exceed DataLen.	0

3.5.2.3 MKC3167_GET_ERR_CHNL_ID (Get Error Channel ID)

3.5.2.3.1 Function

Report the faulty channel ID.

- Precautions: the current channel fault recording function is only supported for high-voltage side (KJG) scenarios and low-voltage side dual-circuit (BXJ2) scenarios. For the specific correspondence between the obtained fault channel numbers and the module hardware channel numbers, please refer to the table of channel numbers and hardware channel correspondences.

3.5.2.3.2 Description of Parameter

Input Parameter	Type of Data	Explanation of Parameter Values	Initial Value
EN_R	BOOL	Enable or not, rising edge effective.	FALSE
Output Parameter	**Type of Data**	**Explanation of Parameter Values**	**Initial Value**
Q	BOOL	Execution result, set to TRUE if successful.	FALSE
Error	BYTE	Error Code: 0: operation completed 1: no faulty channel	0
ChannelID	BYTE	Fault Channel Number, channel numbers: 1-16, 1 represents the first channel, 2 represents the second channel, and so on, up to a total of 16 channels.	0

3.5.2.4 MKC3167_READ_FAULT_DATA (Fault data read)

3.5.2.4.1 Function

Retrieve the fault channel data and upload the fault channel data obtained from the protective IO to the CPU module.

3.5.2.4.2 Description of Parameter

Input Parameter	Type of Data	Explanation of Parameter Values	Initial Value
In	BOOL	Enable, active high	FALSE
ChannelID	BYTE	The channel numbers for which fault data needs to be retrieved are 1-16. Here, 1 represents the first channel, 2 represents the second channel, and so on, up to a total of 16 channels.	0
DataLen	WORD	Data Length, Address Space Size (user-specified length, maximum 240 sample points, cannot be less than 192 sample points).	0
Address	POINTER TO WORD	Data Storage Address	0
Output Parameter	**Type of Data**	**Explanation of Parameter Values**	**Initial Value**
Done	BOOL	Execution Result: TRUE for Success: Execution successful, data read successfully. FALSE for Failure: Execution failed.	FALSE
Busy	BOOL	Is data being read: TRUE：busy；FALSE：free	FALSE
	BYTE	Error Code:	0

Error		0: operation completed 1: no fault channel 2: illegal input channel ID 3: illegal address 4: illegal input data length 5: fault data read error	
RealDataLen	WORD	Data Length: Actual copied data length, which must not exceed DataLen.	0

Beijing HollySys Intelligent Technologies Co., Ltd..

Di Sheng Middle Road, No.2

Economic-Technological Development Area

100176 Beijing, China

Tel: 010-5898 1588

Hotline: 4008111999

Fax: 010-5898 1558

http://www.hollysys.com