LKS Safety Control System Instruction Manual

Intelligence For Excellence

HollySys Group

Beijing HollySys Intelligent Technologies Co., Ltd.

Version: 01/2025

LKS Safety Control System

Instruction Manual

Version: V1.4

Jaunary, 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
Fax: +86 010-5898 1558
Web: http://www.hollysys.com
Email: PLC@hollysys.com
Sina weibo: http://weibo.com/hollysysplc

Contents

Chapter 1 About Manual

1.1 Document Update

Version	Date	Description
V1.0	2017.08.30	New
V1.1	2018.02.02	Updated
V1.2	2018.11.30	Add the Chapter 2.3.2 Data Split and Merge Functions
V1.0	June 14.2019	Update Copyright
V1.1	March 11.2022	Update logo
V1.2	August 2023	Descriptive modification
V1.3	August 2024	Add BYTE, WORD, SINT, INT to REAL, LREAL data type conversion description
V1.4	January. 2025	An increase in CPU load as indicated by sysGetCPULoad (which retrieves the controller load) suggests that the controller's CPU load should be kept below 80%.

1.2 Purpose

This manual gives a comprehensive, detailed introduction and examples of each algorithm block, to help you better understand the information of each block, so that the correct selection and use of.

1.3 Intended Audience

LKS programmers, operators and system maintenance personnel.

The engineers who use this manual should have the automatic control knowledge.

1.4 Document Conventions

1.4.1 Menu

Menu commands, the sub commands in the menu, the tag page name in this manual with the note said, such as [file], [Edit] menu.

1.4.2 Mouse

Press: press the left mouse button once and keep.

Click: Press the left mouse button once and release.

Right-click: Press the right mouse button once and release.

Double click: quickly press the left mouse button two times and release.

Drag: Press and hold the left mouse button while moving the mouse.

1.4.3 Keyboard

The keys on the keyboard in the manual with bold style, such as **Enter**.

1.4.4 Online Help

The online help that is integrated into the software is a supplement to this manual. The purpose of providing online help is to provide detailed support when using the software.

The help system is integrated into the software through a number of interfaces.

- Press the **F1** key or **help** in the dialog box to accurately locate the relevant help instructions.
- You can also use the menu bar [help] - [content] call the help of the current software.

As the manual and online help some of the contents of the scheduling order is different, so the use of online help, you can use the search function to find the full text.

1.4.5 Important Information

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

- Danger, indicating that the operation may lead to the potential threats of physical damage or personal injury or death.

- Electrical shock, indicating that the operation may lead to the potential threat of electric shock.

- Warning, indicating that the operation may lead to the potential threats of failure or damage to software and hardware equipment.

- Important icon, identifies important information about the operations or functions need to understand.

- Operation icon, indicating the open or operation method of the object.

Interpretation of the contents of the operation Icon.

The location of the operation: Mouse command + Operation sequence.

- Menu bar: click [Insert]-[Input element],
- Tool bar:  ,
- Shortcut: **Ctrl** + **I**,
- **POU**: right click the empty area, click [input element].

As shown in the first operation method above, in the menu bar, click [Insert] menu and the sub menu [Input element] command.

SEE ALSO

- Reference icon, indicating the page or content to be referenced for getting to know the function.

1.5 Related Documentation

Safety FA-AutoThink User Manual

LKS Safety Control System Product Manual

LKS Safety Control System Instruction Manual

Chapter 2 Library Management

2.1 Instructions

The following instructions are not supported for use.

Instruction	Library
sysGetDPConfirmState	HS_DPDIAG.HLF
sysGetDPSlaveComState	
DIProc	HS_IOPROC.HLF

2.2 Basic Operation

2.2.1 Arithmetical Operation

2.2.1.1 ADD(Add)

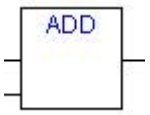


Figure 1 Add

1. Parameter Description

Variable	Description	Data Type
Input	Input Variable	DWORD, DINT, REAL, TIME, LREAL
Output	Output Variable	DWORD, DINT, REAL, TIME, LREAL

2. Description

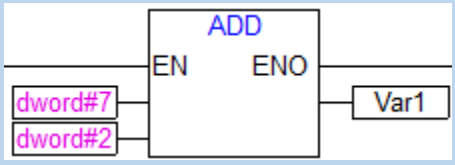
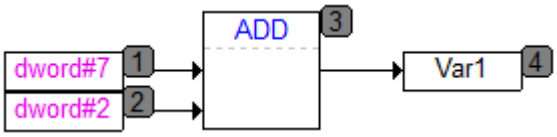
This function block completes addition to input variables.

3. Application Example of Instructions

Variable Definition

No.	Name	Description	Data Type	Initial Value	Area
0001	Var1		DWORD	0	G Area

Programing language	Program

Variable Definition	
Ladder diagram (LD)	
Function Block Diagram(FBD)	

2.2.1.2 SUB(Subtract)

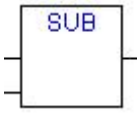


Figure 2 Subtract

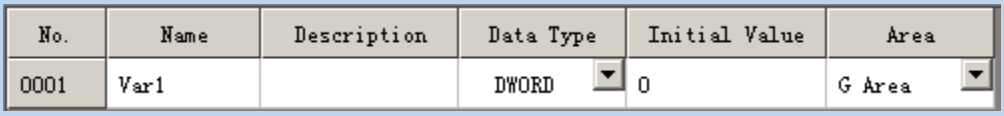
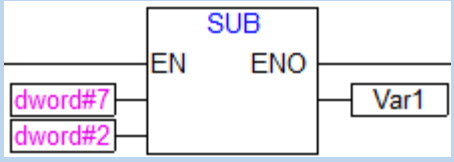
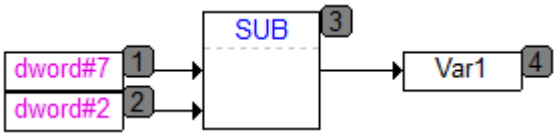
1. Parameter Description

Variable	Description	Data Type
Input	Input Variable	DWORD, DINT, REAL, TIME, LREAL
Output	Output Variable	DWORD, DINT, REAL, TIME, LREAL

2. Description

This function block completes subtraction to input variables.

3. Application Example of Instructions

Variable Definition	
	
Programing language	Program
Ladder diagram (LD)	
Function Block Diagram(FBD)	

2.2.1.3 MUL(Multiply)

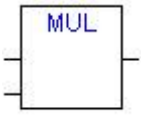


Figure 3 Multiply

1. Parameter Description

Variable	Description	Data Type
Input	Input Variable	DWORD, DINT, REAL, LREAL
Output	Output Variable	DWORD, DINT, REAL, LREAL

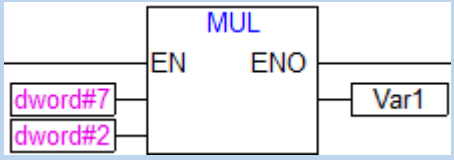
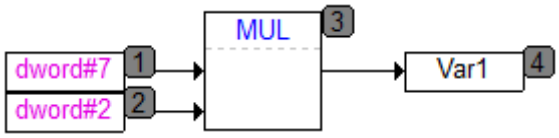
2. Description

This function block completes multiplication to input variables.

3. Application Example of Instructions

Variable Definition

No.	Name	Description	Data Type	Initial Value	Area
0001	Var1		DWORD	0	G Area

Programing language	Program
Ladder diagram (LD)	
Function Block Diagram(FBD)	

2.2.1.4 DIV(Divide)

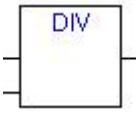


Figure 4 Divide

1. Parameter Description

Variable	Description	Data Type
Input	Input Variable	DWORD, DINT, REAL, LREAL

Variable	Description	Data Type
Output	Output Variable	DWORD, DINT, REAL, LREAL

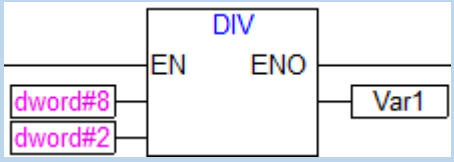
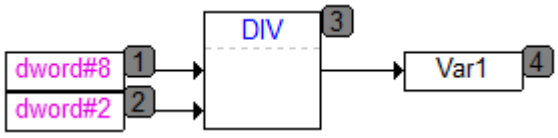
2. Description

This function block completes division to input variables.

3. Application Example of Instructions

Variable Definition

No.	Name	Description	Data Type	Initial Value	Area
0001	Var1		DWORD	0	G Area

Programing language	Program
Ladder diagram (LD)	
Function Block Diagram(FBD)	

2.2.1.5 MOD(Modulus)

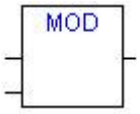


Figure 5 Modulus

1. Parameter Description

Variable	Description	Data Type
Input	Input Variable	DWORD, DINT
Output	Output Variable	DWORD, DINT

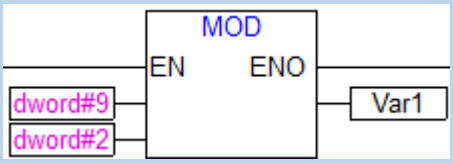
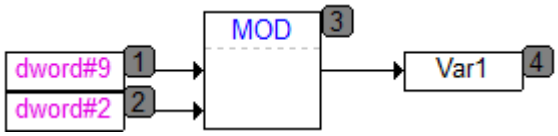
2. Description

The result of this function will be the remainder of the division.

3. Application Example of Instructions

Variable Definition

No.	Name	Description	Data Type	Initial Value	Area
0001	Var1		DWORD	0	G Area

Variable Definition	
Programing language	Program
Ladder diagram (LD)	
Function Block Diagram(FBD)	

2.2.1.6 ADD_WORD(Add)

This function block completes addition to input variables of WORD type.

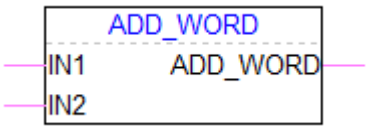


Figure 6 Add

1. Parameter Description

Input	Data Type
IN1	WORD
IN2	WORD
Output	
ADD_WORD	WORD

2. Application Example of Instructions

Please refer to 2.2.1.1 ADD(Add).

2.2.1.7 SUB_WORD(Subtract)

This function block completes subtraction to input variables of WORD type.

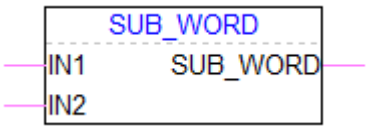


Figure 7 Subtract

1. Parameter Description

Input	Data Type
IN1	WORD
IN2	WORD
Output	
SUB_WORD	WORD

2. Application Example of Instructions

Please refer to 2.2.1.2 SUB(Subtract).

2.2.1.8 MUL_WORD(Multiply)

This function block completes multiplication to input variables of WORD type.

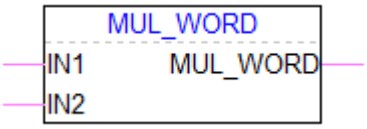


Figure 8 Multiply

1. Parameter Description

Input	Data Type
IN1	WORD
IN2	WORD
Output	
MUL_WORD	WORD

2. Application Example of Instructions

Please refer to 2.2.1.3 MUL(Multiply).

2.2.1.9 DIV_WORD(Divide)

This function block completes division to input variables of WORD type.

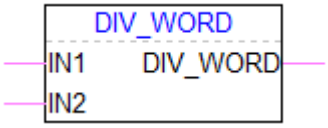


Figure 9 Divide

1. Parameter Description

Input	Data Type
IN1	WORD
IN2	WORD
Output	
DIV_WORD	WORD

2. Application Example of Instructions

Please refer to 2.2.1.4 DIV(Divide).

2.2.1.10 MOD_WORD(Modulus)

The result of this function will be the remainder of the division, and only WORD type for inputs.

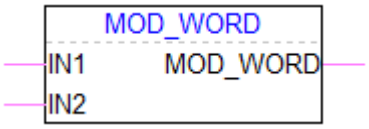


Figure 10 Modulus

1. Parameter Description

Input	Data Type
IN1	WORD
IN2	WORD
Output	
MOD_WORD	WORD

2. Application Example of Instructions

Please refer to 2.2.1.5 MOD(Modulus).

2.2.1.11 ADD_INT(Add)

This function block completes addition to input variables of INT type.

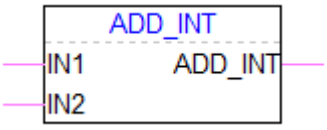


Figure 11 Add

1. Parameter Description

Input	Data Type
IN1	INT
IN2	INT
Output	
ADD_INT	INT

2. Application Example of Instructions

Please refer to 2.2.1.1 ADD(Add).

2.2.1.12 SUB_INT(Subtract)

This function block completes subtraction to input variables of INT type.

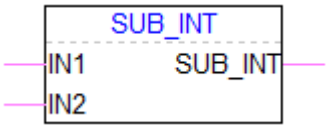


Figure 12 Subtract

1. Parameter Description

Input	Data Type
IN1	INT
IN2	INT
Output	
SUB_INT	INT

2. Application Example of Instructions

Please refer to 2.2.1.2SUB(Subtract).

2.2.1.13 MUL_INT(Multiply)

This function block completes multiplication to input variables of INT type.

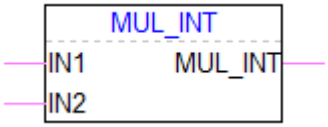


Figure 13 Multiply

1. Parameter Description

Input	Data Type
IN1	INT
IN2	INT
Output	
MUL_INT	INT

2. Application Example of Instructions

Please refer to 2.2.1.3 MUL(Multiply).

2.2.1.14 DIV_INT(Divide)

This function block completes division to input variables of INT type.

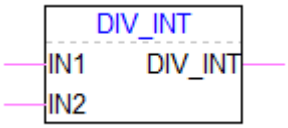


Figure 14 Divide

1. Parameter Description

Input	Data Type
IN1	INT
IN2	INT
Output	
DIV_INT	INT

2. Application Example of Instructions

Please refer to 2.2.1.4 DIV(Divide).

2.2.1.15 MOD_INT(Modulus)

The result of this function will be the remainder of the division, and only INT type for inputs.

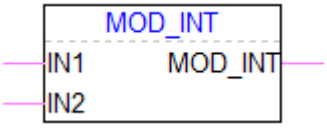


Figure 15 Modulus

1. Parameter Description

Input	Data Type
IN1	INT
IN2	INT
Output	
MOD_INT	INT

2. Application Example of Instructions

Please refer to 2.2.1.5 MOD(Modulus).

2.2.1.16 DIV_DWORD(Divide)

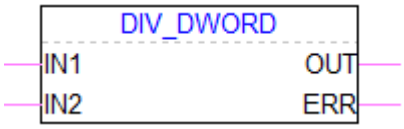


Figure 16 Division Operation

1. Parameter Description

Input	Data Type	Description	Initial Value	Supplement
IN1	DWORD	Dividend	0	
IN2	DWORD	Divisor	0	Divisor cannot be 0.
Output				
OUT	DWORD	Result after division calculation.	0	
ERR	BOOL	If ERR=0, it means function block works normally; if ERR=1, it means divisor=0, and the OUT will keep the last calculation result.	0	

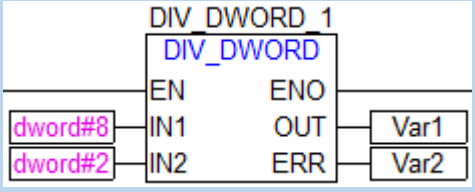
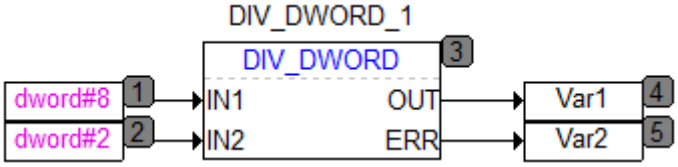
2. Description

When IN2≠0, carry out division operation calculation to input variable and output it; if IN2=0, output will keep the last calculation result, and ERR pin shall be set.

3. Application Example of Instructions

Variable Definition

No.	Name	Description	Data Type	Initial Value	Area
0001	DIV_DWORD_1		DIV_DWORD		G Area
0002	Var1		DWORD	0	G Area
0003	Var2		BOOL	FALSE	G Area

Programing language	Program
Ladder diagram (LD)	
Function Block Diagram(FBD)	

2.2.1.17 DIV_DINT(Divide)

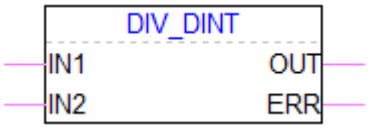


Figure 17 Division Operation

1. Parameter Description

Input	Data Type	Description	Initial Value	Supplement
IN1	DINT	Dividend	0	
IN2	DINT	Divisor	0	Divisor can not be 0.
Output				
OUT	DINT	Result after division calculation.	0	
ERR	BOOL	If ERR=0, it means function block works normally; if ERR=1, it means divisor=0, and the OUT will keep the last calculation result.	0	

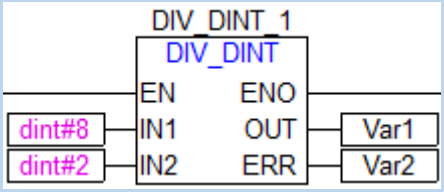
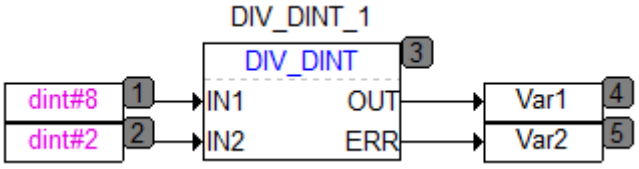
2. Description

When IN2≠0, carry out division operation calculation to input variable and output it; if IN2=0, output will keep the last calculation result, and ERR pin shall be set.

3. Application Example of Instructions

Variable Definition

No.	Name	Description	Data Type	Initial Value	Area
0001	DIV_DINT_1		DIV_DINT		G Area
0002	Var1		DINT	0	G Area
0003	Var2		BOOL	FALSE	G Area

Programing language	Program
Ladder diagram (LD)	
Function Block Diagram(FBD)	

2.2.1.18 DIV_REAL(Divide)

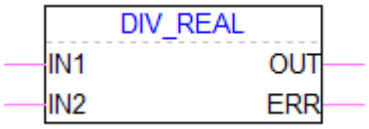


Figure 18 Division Operation

1. Parameter Description

Input	Data Type	Description	Initial Value	Supplement
IN1	REAL	Dividend	0	
IN2	REAL	Divisor	0	Divisor can not be 0.
Output				
OUT	REAL	Result after division calculation	0	
ERR	BOOL	If ERR=0, it means function block works normally; if ERR=1, it means divisor=0,, and the OUT will keep the last calculation result.	0	

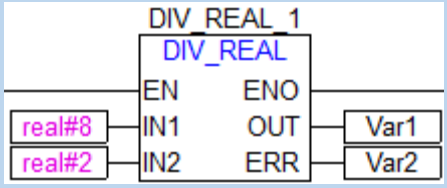
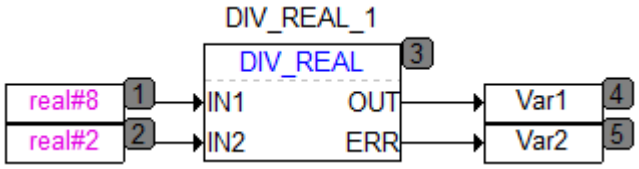
2. Description

When IN2≠0, carry out division operation calculation to input variable and output it; if IN2=0, output will keep the last calculation result, and ERR pin shall be set.

3. Application Example of Instructions

Variable Definition

No.	Name	Description	Data Type	Initial Value	Area
0001	DIV_REAL_1		DIV_REAL		G Area
0002	Var1		LREAL	0	G Area
0003	Var2		BOOL	FALSE	G Area

Programing language	Program
Ladder diagram (LD)	
Function Block Diagram(FBD)	

2.2.1.19 DIV_LREAL(Divide)

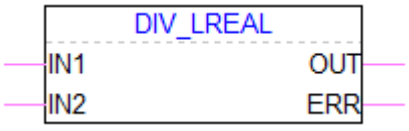


Figure 19 Division operation

1. Parameter Description

Input	Data Type	Description	Initial Value	Supplement
IN1	LREAL	Dividend	0	
IN2	LREAL	Divisor	0	Divisor can not be 0.
Output				
OUT	LREAL	Result after division calculation.	0	

Input	Data Type	Description	Initial Value	Supplement
ERR	BOOL	If ERR=0, it means function block works normally; if ERR=1, it means divisor=0, and the OUT will keep the last calculation result.	0	

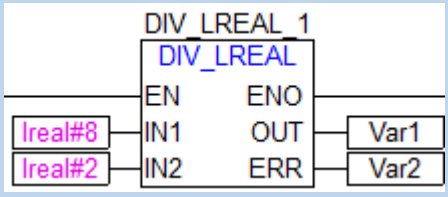
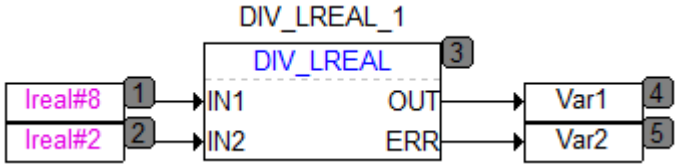
2. Description

When IN2≠0, carry out division operation calculation to input variable and output it; if IN2=0, output will keep the last calculation result, and ERR pin shall be set.

3. Application Example of Instructions

Variable Definition

No.	Name	Description	Data Type	Initial Value	Area
0001	DIV_LREAL_1		DIV_LREAL		G Area
0002	Var1		LREAL	0	G Area
0003	Var2		BOOL	FALSE	G Area

Programing language	Program
Ladder diagram (LD)	
Function Block Diagram(FBD)	

2.2.1.20 ABS(Absolute Value)

Figure 20 Absolute Value

1. Parameter Description

Input	Description	Data Type	Remarks
Input	Input variable	LREAL	
Output			
Output	Result of ABS calculation	LREAL	

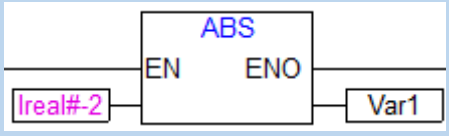
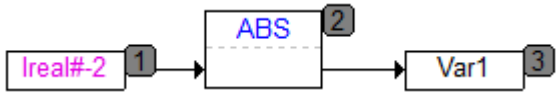
2. Description

Result of ABS calculation.

3. Application Example of Instructions

Variable Definition

No.	Name	Description	Data Type	Initial Value	Area
0001	Var1		LREAL	0	G Area

Programing language	Program
Ladder diagram (LD)	
Function Block Diagram(FBD)	

2.2.1.21 SQRT(Square Root)

Figure 21 Square Root

1. Parameter Description

Input	Description	Data Type	Remarks
Input	Input variable	LREAL	
Output			
Output	Result after SQRT calculation	LREAL	

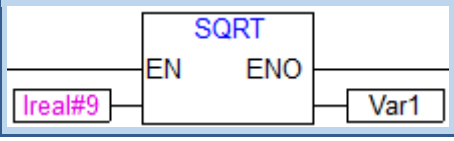
2. Description

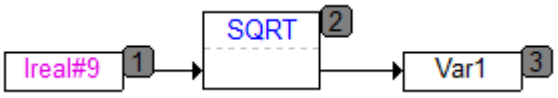
Result after SQRT calculation.

3. Application Example of Instructions

Variable Definition

No.	Name	Description	Data Type	Initial Value	Area
0001	Var1		LREAL	0	G Area

Programing language	Program
Ladder diagram (LD)	

Variable Definition	
Function Block Diagram(FBD)	

2.2.1.22 LOG(Common Logarithm)

Figure 22 Logarithm of a Number in Base 10

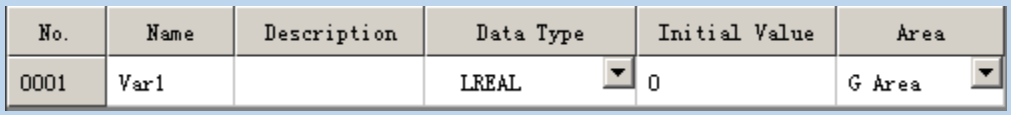
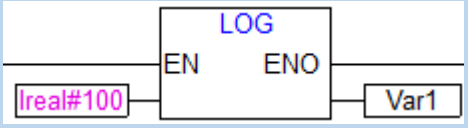
1. Parameter Description

Input	Description	Data Type	Remarks
Input	Input variable	LREAL	
Output			
Output	Result after LOG calculation	LREAL	

2. Description

Result after LOG calculation.

3. Application Example of Instructions

Variable Definition	
	
Programing language	Program
Ladder diagram (LD)	
Function Block Diagram(FBD)	

2.2.1.23 LN(Natural Logarithm)

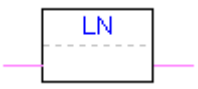


Figure 23 Natural Logarithm

1. Parameter Description

Input	Description	Data Type	Remarks
Input	Input variable	LREAL	

Input	Description	Data Type	Remarks
Output			
Output	Result after LN calculation	LREAL	

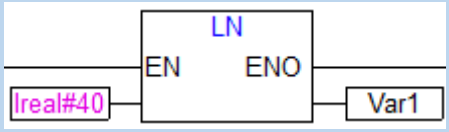
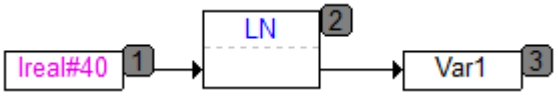
2. Description

Result after LN calculation.

3. Application Example of Instructions

Variable Definition

No.	Name	Description	Data Type	Initial Value	Area
0001	Var1		LREAL	0	G Area

Programing language	Program
Ladder diagram (LD)	
Function Block Diagram(FBD)	

2.2.1.24 EXP(Exponent)

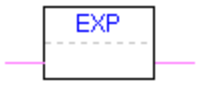


Figure 24 Exponential Function Operation

1. Parameter Description

Input	Description	Data Type	Remarks
Input	Input variable	LREAL	
Output			
Output	Result after exponential operation	LREAL	

2. Description

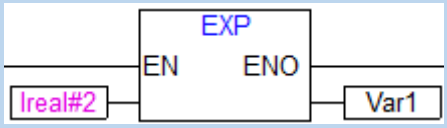
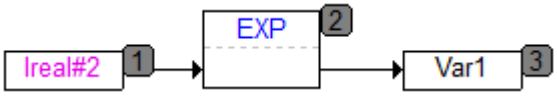
The output value of the input variables e after the second side.

3. Application Example of Instructions

Variable Definition

No.	Name	Description	Data Type	Initial Value	Area
0001	Var1		LREAL	0	G Area

Programing language	Program

Variable Definition	
Ladder diagram (LD)	
Function Block Diagram(FBD)	

2.2.1.25 SIN(Sine)

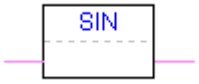


Figure 25 Sine

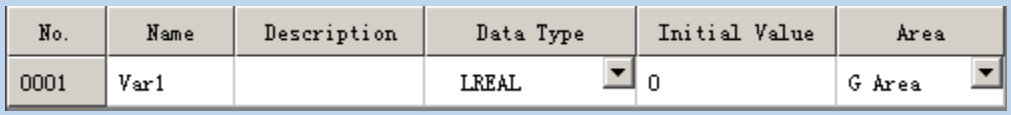
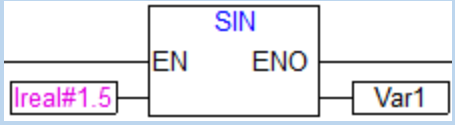
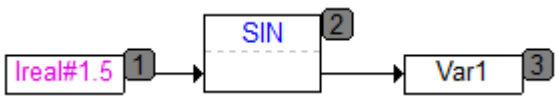
1. Parameter Description

Input	Description	Data Type	Remarks
Input	Input variable	LREAL	Radian value The maxed valid LREAL input is the data expressed by 52 bits in binary, not the default 64 bits.
Output			
Output	Result after exponential operation	LREAL	

2. Description

Result after SIN calculation, the input is radian value.

3. Application Example of Instructions

Variable Definition	
	
Programing language	Program
Ladder diagram (LD)	
Function Block Diagram(FBD)	

2.2.1.26 COS(Cosine)

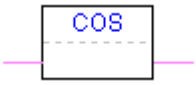


Figure 26 Cosine

1. Parameter Description

Input	Description	Data Type	Remarks
Input	Input variable	LREAL	Radian value The maxed valid LREAL input is the data expressed by 52 bits in binary, not the default 64 bits.
Output			
Output	Result after COS calculation.	LREAL	

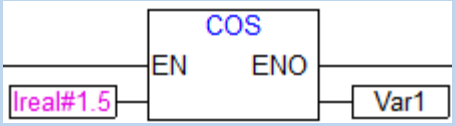
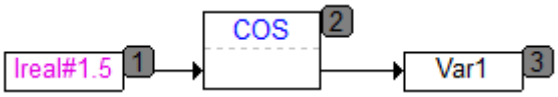
2. Description

Result after COS calculation, input is radian value.

3. Application Example of Instructions

Variable Definition

No.	Name	Description	Data Type	Initial Value	Area
0001	Var1		LREAL	0	G Area

Programing language	Program
Ladder diagram (LD)	
Function Block Diagram(FBD)	

2.2.1.27 TAN(Tangent)

Figure 27 Tangent

1. Parameter Description

Input	Description	Data Type	Remarks
Input	Input variable	LREAL	Radian value The maxed valid LREAL input is the data expressed by 52 bits in binary, not the default 64 bits.

Input	Description	Data Type	Remarks
Output			
Output	Result after TAN calculation	LREAL	

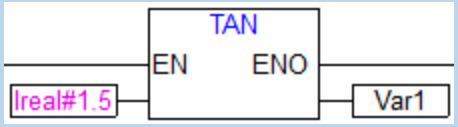
2. Description

Result after TAN calculation, input is radian value.

3. Application Example of Instructions

Variable Definition

No.	Name	Description	Data Type	Initial Value	Area
0001	Var1		LREAL	0	G Area

Programing language	Program
Ladder diagram (LD)	
Function Block Diagram(FBD)	

2.2.1.28 ASIN(Arc Sine)

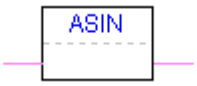


Figure 28 Arc Sine

1. Parameter Description

Input	Description	Data Type	Remarks
Input	Input variable	LREAL	
Output			
Output	Result after ASIN calculation	LREAL	Radian value

2. Description

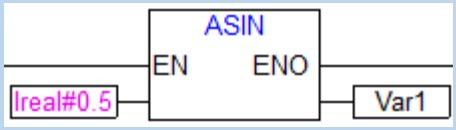
Result of ASIN calculation.

3. Application Example of Instructions

Variable Definition

No.	Name	Description	Data Type	Initial Value	Area
0001	Var1		LREAL	0	G Area

Programing language	Program

Variable Definition	
Ladder diagram (LD)	
Function Block Diagram(FBD)	

2.2.1.29 ACOS(Arc Cosine)

Figure 29 Arc cosine

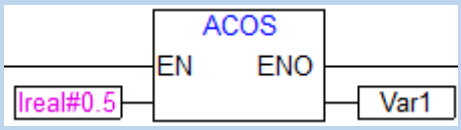
1. Parameter Description

Input	Description	Data Type	Remarks
Input	Input variable	LREAL	
Output			
Output	Result after ACOS calculation	LREAL	Radian value

2. Description

Result after ACOS calculation.

3. Application Example of Instructions

Variable Definition	
	
Programing language	Program
Ladder diagram (LD)	
Function Block Diagram(FBD)	

2.2.1.30 ATAN(Arc Tangent)

Figure 30 Arc Tangent

1. Parameter Description

Input	Description	Data Type	Remarks
Input	Input variable	LREAL	
Output			
Output	Result after ATAN calculation	LREAL	Radian value

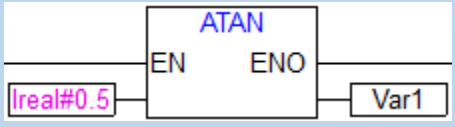
2. Description

Result after ATAN calculation.

3. Application Example of Instructions

Variable Definition

No.	Name	Description	Data Type	Initial Value	Area
0001	Var1		LREAL	0	G Area

Programing language	Program
Ladder diagram (LD)	
Function Block Diagram(FBD)	

2.2.1.31 EXPT(Power)

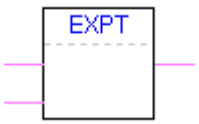


Figure 31 Exponentiation of Variable with Another Variable

1. Parameter Description

Input	Description	Data Type	Remarks
Input 1	Input variable 1 (Y)	LREAL	
Input 2	Input variable 2 (n)	LREAL	
Output			
Output	Result of EXPT calculation.	LREAL	

2. Description

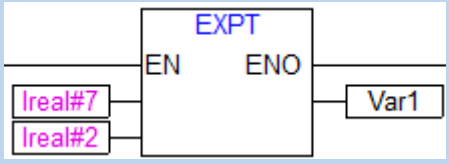
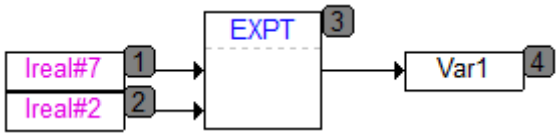
Output the result of Y^n.

- It shall comply with arithmetical operation rule when assign input 1 and input 2.

3. Application Example of Instructions

Variable Definition

No.	Name	Description	Data Type	Initial Value	Area
0001	Var1		LREAL	0	G Area

Programing language	Program
Ladder diagram (LD)	
Function Block Diagram(FBD)	

2.2.1.32 SQRT_R(Single-precision Square Root)

Figure 32 Square Root

1. Parameter Description

Input	Data Type	Description	Initial Value	Supplement
IN	REAL	Input variable	0	It is a real number, which is≥0.
Output				
OUT	REAL	Result after square root calculation.	0	
ERR	BOOL	If ERR=0, it means function block works normally; if ERR=1, it means the input<0, and the OUT will keep the last calculation result.	0	

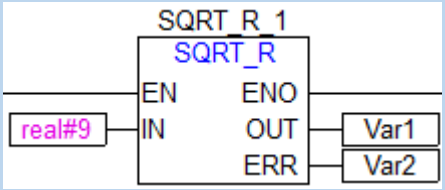
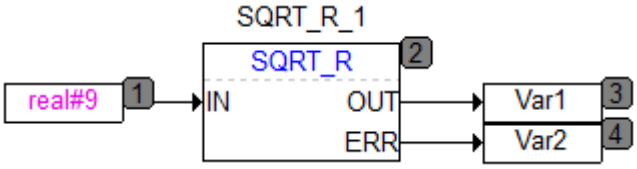
2. Description

If IN≥0, carry out square root calculation to input variable and output it; if IN<0, output will keep the last calculation result, and ERR pin shall be set.

3. Application Example of Instructions

Variable Definition

No.	Name	Description	Data Type	Initial Value	Area
0001	SQRT_R_1		SQRT_R		G Area
0002	Var1		REAL	0	G Area
0003	Var2		BOOL	FALSE	G Area

Programing language	Program
Ladder diagram (LD)	
Function Block Diagram(FBD)	

2.2.1.33 LOG_R(Single-precision Common Logarithm)

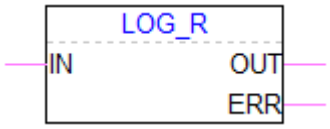


Figure 33 Logarithm of a Number in Base 10

1. Parameter Description

Input	Data Type	Description	Initial Value	Supplement
IN	REAL	Input variable	0	It is a positive real number.
Output				
OUT	REAL	Result after logarithm of a number in base 10 calculation.	0	

Input	Data Type	Description	Initial Value	Supplement
ERR	BOOL	If ERR=0, it means function block works normally; if ERR=1, it means the input≤0, and the OUT will keep the last calculation result.	0	

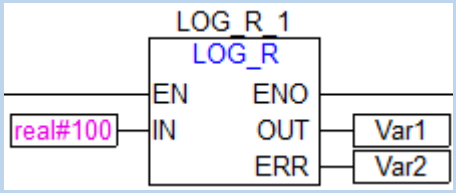
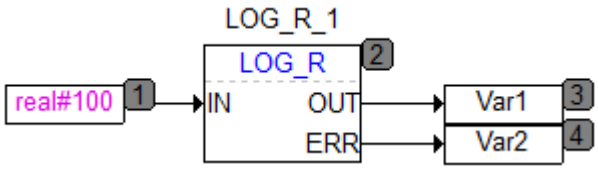
2. Description

If IN>0, carry out logarithm of a number in base 10 calculation to input variable and output it; if IN≤0, output will keep the last calculation result, and ERR pin shall be set.

3. Application Example of Instructions

Variable Definition

No.	Name	Description	Data Type	Initial Value	Area
0001	LOG_R_1		LOG_R		G Area
0002	Var1		REAL	0	G Area
0003	Var2		BOOL	FALSE	G Area

Programing language	Program
Ladder diagram (LD)	
Function Block Diagram(FBD)	

2.2.1.34 LN_R(Single-precision Natural Logarithm)

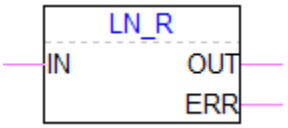


Figure 34 Natural Logarithm

1. Parameter Description

Input	Data Type	Description	Initial Value	Supplement
IN	REAL	Input variable	0	It is a real number, which is>0.
Output				
OUT	REAL	Result after natural logarithm calculation.	0	
ERR	BOOL	If ERR=0, it means function block works normally; if ERR=1, it means the input≤0, and the OUT will keep the last calculation result.	0	

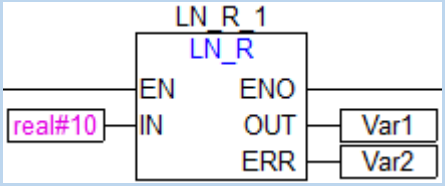
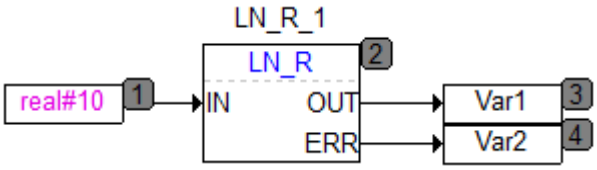
2. Description

If IN>0, carry out natural logarithm calculation to input variable and output it; if IN≤0, output will keep the last calculation result, and ERR pin shall be set.

3. Application Example of Instructions

Variable Definition

No.	Name	Description	Data Type	Initial Value	Area
0001	LN_R_1		LN_R		G Area
0002	Var1		REAL	0	G Area
0003	Var2		BOOL	FALSE	G Area

Programing language	Program
Ladder diagram (LD)	
Function Block Diagram(FBD)	

2.2.1.35 EXP_R(Single-precision Exponent)

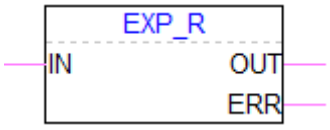


Figure 35 Exponential Function

1. Parameter Description

Input	Data Type	Description	Initial Value	Supplement
IN	REAL	Input variable	0	It is a real number, and IN≤88.
Output				
OUT	REAL	Result after exponential calculation.	0	
ERR	BOOL	If ERR=0, it means function block works normally; If ERR=1, it means the input>88, and the OUT will keep the last calculation result.	0	

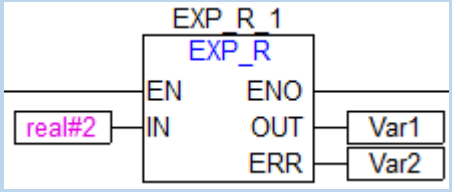
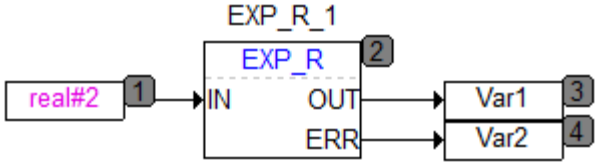
2. Description

If IN≤88, carry out exponential function calculation to input variable and output it; if IN>88, output will keep the last calculation result, and ERR pin shall be set.

3. Application Example of Instructions

Variable Definition

No.	Name	Description	Data Type	Initial Value	Area
0001	EXP_R_1		EXP_R		G Area
0002	Var1		REAL	0	G Area
0003	Var2		BOOL	FALSE	G Area

Programing language	Program
Ladder diagram (LD)	
Function Block Diagram(FBD)	

2.2.1.36 ASIN_R(Single-precision Arc Sine)

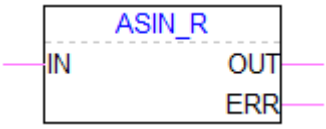


Figure 36 ASIN_R

1. Parameter Description

Input	Data Type	Description	Initial Value	Supplement
IN	REAL	Input variable	0	The range is [-1, 1].
Output				
OUT	REAL	Result after arc sine calculation.	0	
ERR	BOOL	If ERR=0, it means function block works normally; if ERR=1, it means the input out the range of [-1, 1], and the OUT will keep the last	0	

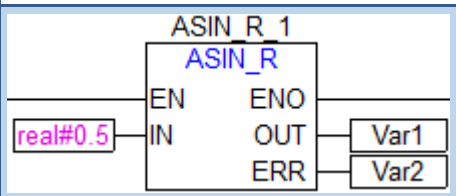
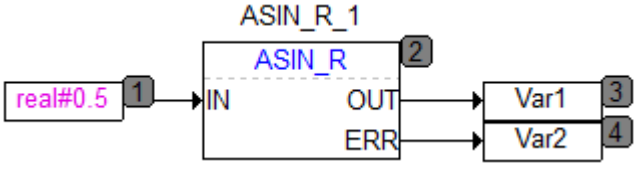
2. Description

If -1≤IN≤1, carry out arc sine calculation to input variable and output it; if IN>1 or IN<-1, output will keep the last calculation result, and ERR pin shall be set.

3. Application Example of Instructions

Variable Definition

No.	Name	Description	Data Type	Initial Value	Area
0001	ASIN_R_1		ASIN_R		G Area
0002	Var1		REAL	0	G Area
0003	Var2		BOOL	FALSE	G Area

Programing language	Program
Ladder diagram (LD)	
Function Block Diagram(FBD)	

2.2.1.37 ACOS_R(Single-precision Arc Cosine)

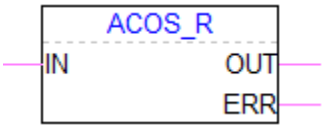


Figure 37 ACOS_R

1. Parameter Description

Input	Data Type	Description	Initial Value	Supplement
IN	REAL	Input variable	0	The range is [-1, 1].
Output				
OUT	REAL	Result after arc cosine calculation.	0	
ERR	BOOL	If ERR=0, it means function block works normally; if ERR=1, it means the input out the range of [-1, 1], and the OUT will keep the last calculation result.	0	

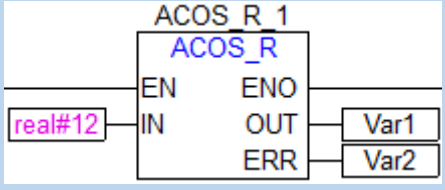
2. Description

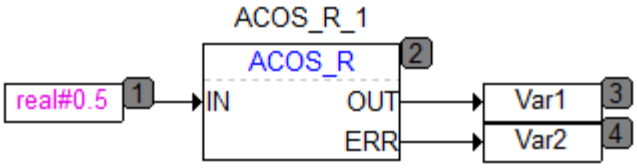
If -1≤IN≤1, carry out arc cosine calculation to input variable and output it; if IN>1 or IN<-1, output will keep the last calculation result, and ERR pin shall be set.

3. Application Example of Instructions

Variable Definition

No.	Name	Description	Data Type	Initial Value	Area
0001	ACOS_R_1		ACOS_R		G Area
0002	Var1		REAL	0	G Area
0003	Var2		BOOL	FALSE	G Area

Programing language	Program
Ladder diagram (LD)	

Variable Definition	
Function Block Diagram(FBD)	

2.2.1.38 EXPT_R(Single-precision Power)

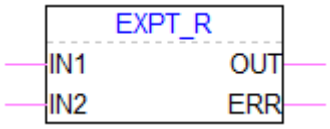


Figure 38 Exponentiation of Variable with Another Variable

1. Parameter Description

Input	Data Type	Description	Initial Value	Supplement
IN1	REAL	Input1	0	The range is [0, +∞).
IN2	REAL	Input2	0	
Output				
OUT	REAL	Result after exponentiation of variable with another variable calculation.	0	
ERR	BYTE	If ERR=0, it means function block works normally; if ERR=1, it means IN1 is out of the range, and the OUT will keep the last calculation result; if ERR=2, it means IN2 is less than or equal to zero when IN1=0, and the OUT will keep the last calculation result; if ERR=3, it means ln|ln（IN1）|+ln|ln2|>4.466, here the OUT will keep the last calculation result.	0	

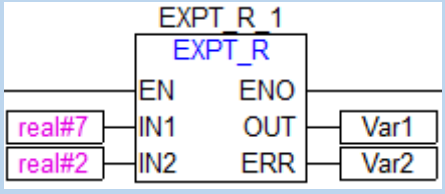
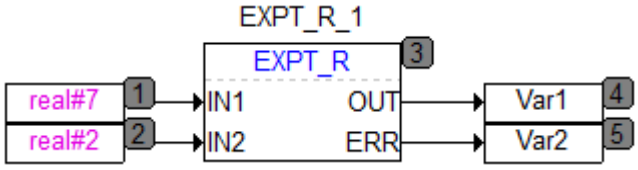
2. Description

When IN1 and IN 2 is in correct range, output the result of $IN1^{IN\,2}$; if IN 1 or IN2 does not in correct range, ERR output is not 0 and keep the last calculation result.

3. Application Example of Instructions

Variable Definition

No.	Name	Description	Data Type	Initial Value	Area
0001	EXPT_R_1		EXPT_R		G Area
0002	Var1		REAL	0	G Area
0003	Var2		BYTE	0	G Area

Programing language	Program
Ladder diagram (LD)	
Function Block Diagram(FBD)	

2.2.1.39 SQRT_LR(Double-precision Square Root)

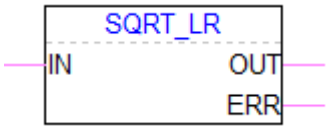


Figure 39 Square Root

1. Parameter Description

Input	Data Type	Description	Initial Value	Supplement
IN	REAL	Input variable	0	It is a real number, which is≥0.
Output				
OUT	REAL	Result after square root calculation.	0	
ERR	BOOL	If ERR=0, it means function block works normally; if ERR=1, it means the input<0, and the OUT will keep the last calculation result.	0	

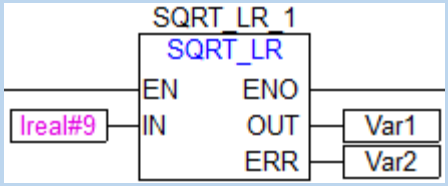
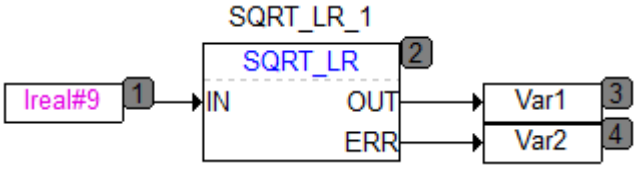
2. Description

If IN≥0, carry out square root calculation to input variable and output it; if IN<0, output will keep the last calculation result, and ERR pin shall be set.

3. Application Example of Instructions

Variable Definition

No.	Name	Description	Data Type	Initial Value	Area
0001	SQRT_LR_1		SQRT_LR		G Area
0002	Var1		LREAL	0	G Area
0003	Var2		BOOL	FALSE	G Area

Programing language	Program
Ladder diagram (LD)	
Function Block Diagram(FBD)	

2.2.1.40 LOG_LR(Double-precision Common Logarithm)

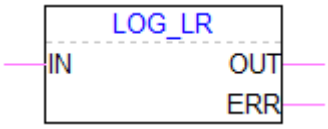


Figure 40 Logarithm of a Number in Base 10

1. Parameter Description

Input	Data Type	Description	Initial Value	Supplement
IN	LREAL	Input variable	0	It is a real number, which is≥0.
Output				
OUT	LREAL	Result after logarithm of number in base 10 calculation.	0	

Input	Data Type	Description	Initial Value	Supplement
ERR	BOOL	If ERR=0, it means function block works normally; if ERR=1, it means the input≤0, and the OUT will keep the last calculation result.	0	

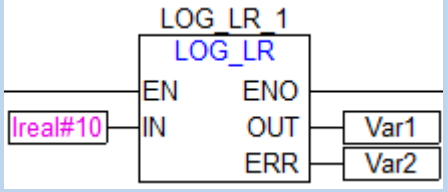
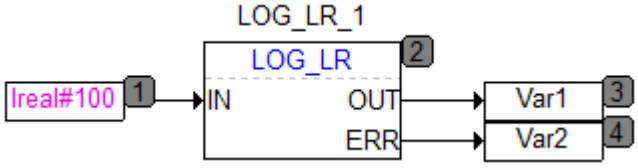
2. Description

If IN>0, carry out logarithm of a number in base 10 calculation to input variable and output it; if IN⩽0, output will keep the last calculation result, and ERR pin shall be set.

3. Application Example of Instructions

Variable Definition

No.	Name	Description	Data Type	Initial Value	Area
0001	LOG_LR_1		LOG_LR		G Area
0002	Var1		LREAL	0	G Area
0003	Var2		BOOL	FALSE	G Area

Programing language	Program
Ladder diagram (LD)	
Function Block Diagram(FBD)	

2.2.1.41 LN_LR(Double-precision Natural Logarithm)

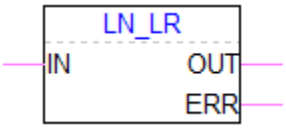


Figure 41 Natural Logarithm

1. Parameter Description

Input	Data Type	Description	Initial Value	Supplement
IN	LREAL	Input variable	0	It is a real number, which is≥0.
Output				
OUT	LREAL	Result after natural logarithm calculation.	0	
ERR	BOOL	If ERR=0, it means function block works normally; If ERR=1, it means the input≤0, and the OUT will keep the last calculation result.	0	

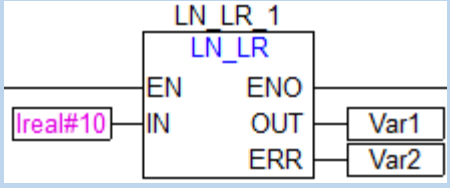
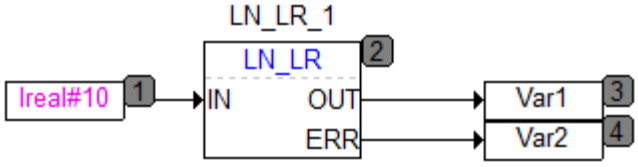
2. Description

If IN≥0, carry out natural logarithm calculation to input variable and output it; if IN<0, output will keep the last calculation result, and ERR pin shall be set.

3. Application Example of Instructions

Variable Definition

No.	Name	Description	Data Type	Initial Value	Area
0001	LN_LR_1		LN_LR		G Area
0002	Var1		LREAL	0	G Area
0003	Var2		BOOL	FALSE	G Area

Programing language	Program
Ladder diagram (LD)	
Function Block Diagram(FBD)	

2.2.1.42 EXP_LR(Double-precision Exponent)

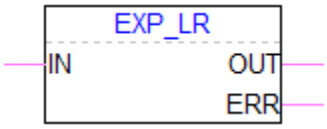


Figure 42 Exponential Function

1. Parameter Description

Input	Data Type	Description	Initial Value	Supplement
IN	LREAL	Input variable	0	It is a real number, which is≤709.
Output				
OUT	LREAL	Result after exponential calculation.	0	
ERR	BOOL	If ERR=0, it means function block works normally; If ERR=1, it means the input>709, and the OUT will keep the last calculation result.	0	

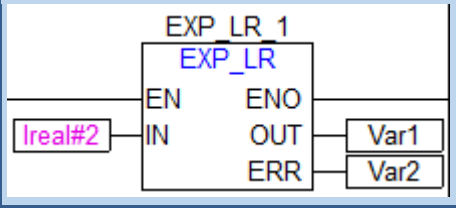
2. Description

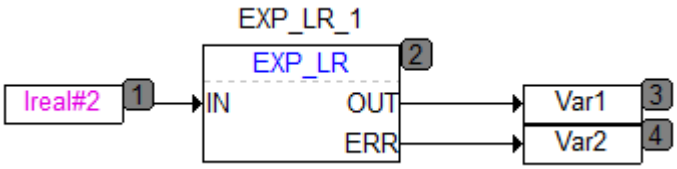
If IN≤709, carry out exponential function calculation to input variable and output the result; if IN>709, output will keep the last calculation result, and ERR pin shall be set.

3. Application Example of Instructions

Variable Definition

No.	Name	Description	Data Type	Initial Value	Area
0001	EXP_LR_1		EXP_LR		G Area
0002	Var1		LREAL	0	G Area
0003	Var2		BOOL	FALSE	G Area

Programing language	Program
Ladder diagram (LD)	

Variable Definition	
Function Block Diagram(FBD)	

2.2.1.43 ASIN_LR(Double-precision Arc Sine)

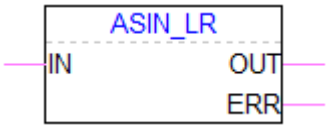


Figure 43 Arc Sine

1. Parameter Description

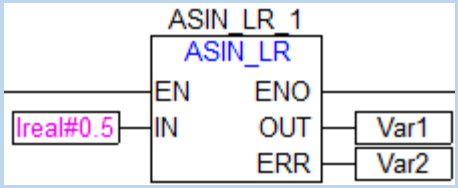
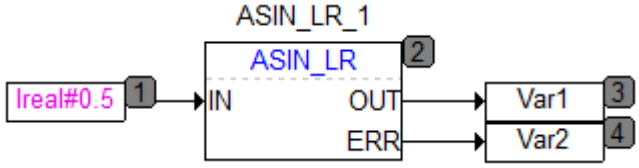
Input	Data Type	Description	Initial Value	Supplement
IN	LREAL	Input variable	0	The range is [-1, 1].
Output				
OUT	LREAL	Result after arc sine calculation.	0	
ERR	BOOL	If ERR=0, it means function block works normally; If ERR=1, it means the input out the range of [-1, 1], and the OUT will keep the last calculation result.	0	

2. Description

If -1≤IN≤1, carry out arc sine calculation to input variable and output the result; if IN>1 or IN<-1, output will keep the last calculation result, and ERR pin shall be set.

3. Application Example of Instructions

Variable Definition					
No.	Name	Description	Data Type	Initial Value	Area
0001	ASIN_LR_1		ASIN_LR		G Area
0002	Var1		LREAL	0	G Area
0003	Var2		BOOL	FALSE	G Area
Programing language	Program				

Variable Definition	
Ladder diagram (LD)	
Function Block Diagram(FBD)	

2.2.1.44 ACOS_LR(Double-precision Arc Cosine)

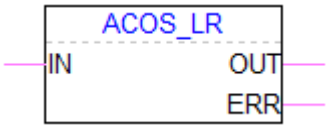


Figure 44 Arc Cosine

1. Parameter Description

Input	Data Type	Description	Initial Value	Supplement
IN	LREAL	Input variable	0	The range is [-1, 1].
Output				
OUT	LREAL	Result after arc cosine calculation.	0	
ERR	BOOL	If ERR=0, it means function block works normally; if ERR=1, it means the input out the range of [-1, 1], and the OUT will keep the last calculation result.	0	

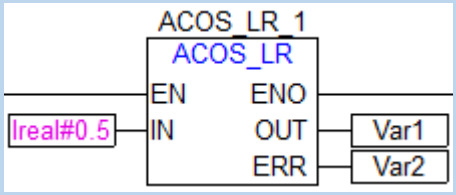
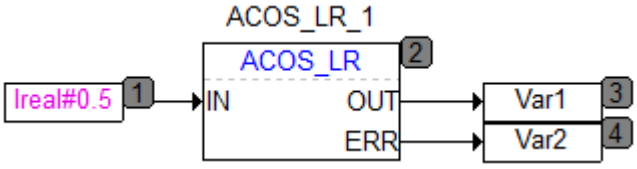
2. Description

If -1≤IN≤1, carry out arc cosine calculation to input variable and output it; if IN>1 or IN<-1, output will keep the last calculation result, and ERR pin shall be set.

3. Application Example of Instructions

Variable Definition

No.	Name	Description	Data Type	Initial Value	Area
0001	ACOS_LR_1		ACOS_LR		G Area
0002	Var1		LREAL	0	G Area
0003	Var2		BOOL	FALSE	G Area

Programing language	Program
Ladder diagram (LD)	
Function Block Diagram(FBD)	

2.2.1.45 EXPT_LR(Double-precision Power)

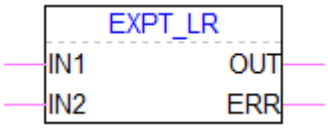


Figure 45 Exponentiation of Variable with Another Variable

1. Parameter Description

Input	Data Type	Description	Initial Value	Supplement
IN1	LREAL	Input variable	0	The range is [0, +∞)
IN2	LREAL	Input variable 2	0	
Output				
OUT	LREAL	Result after exponentiation of variable with another variable calculation.	0	
ERR	BYTE	If ERR=0, it means function block works normally; If ERR=1, it means IN1 is out of the range, and the OUT will keep the last calculation result; If ERR=2, it means IN2 is less than or equal to zero when IN1=0, and the	0	

Input	Data Type	Description	Initial Value	Supplement
		OUT will keep the last calculation result; If ERR=3, it means ln|ln（IN1）|+ln|ln2|>6.562, here the OUT will keep the last calculation result.		

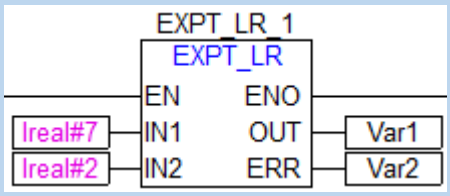
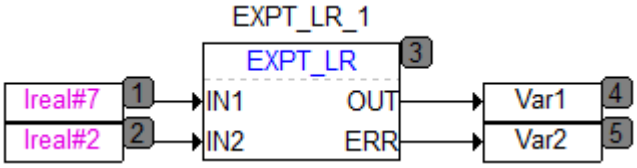
2. Description

When IN1 and IN 2 is in correct range, output the result of IN 1 IN 2; if IN 1 or IN2 does not in correct range, ERR output is not 0 and keep the last calculation result.

3. Application Example of Instructions

Variable Definition

No.	Name	Description	Data Type	Initial Value	Area
0001	EXPT_LR_1		EXPT_LR		G Area
0002	Var1		LREAL	0	G Area
0003	Var2		BYTE	0	G Area

Programing language	Program
Ladder diagram (LD)	
Function Block Diagram(FBD)	

2.2.1.46 MOVE(Assignment)

Figure 46 Assignment

1. Parameter Description

Variable	Description	Data Type
Input	Input Variable	BOOL, BYTE, WORD, DWORD, SINT, INT, DINT, REAL, LREAL, TIME, TOD, DT, DATE
Output	Output Variable	BOOL, BYTE, WORD, DWORD, SINT, INT, DINT, REAL, LREAL, TIME, TOD, DT, DATE

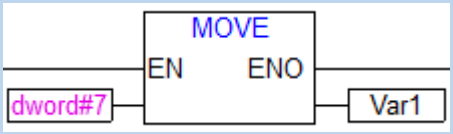
2. Function

The variable is assigned to another variable of matching type in order to two variable values are equal.

3. Application Example of Instructions

Variable Definition

No.	Name	Description	Data Type	Initial Value	Area
0001	Var1		DWORD	0	G Area

Programing language	Program
Ladder diagram (LD)	

2.2.2 Logic Operation

2.2.2.1 AND(And)

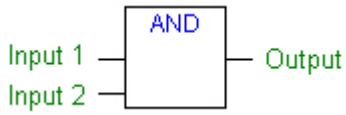


Figure 47 AND

1. Parameter Description

Variable	Description	Data Type
Input1	Input Variable	BOOL, BYTE, WORD, DWORD
Input2	Input Variable	BOOL, BYTE, WORD, DWORD
…		
Output	Output Variable	BOOL, BYTE, WORD, DWORD

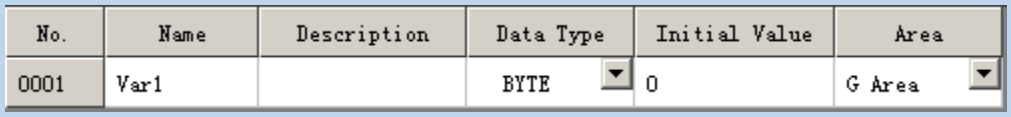
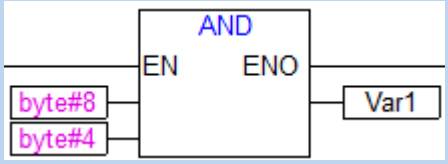
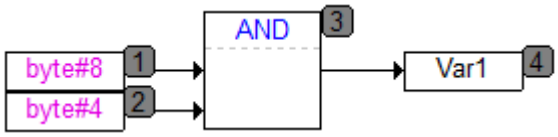
Its Truth Table is as follows.

Input1	Input2	Output
0	0	0
0	1	0
1	0	0
1	1	1

2. Description

This function block is to complete the bitwise AND of bit operands.

3. Application Example of Instructions

Variable Definition	
	
Programing language	Program
Ladder diagram (LD)	
Function Block Diagram(FBD)	

2.2.2.2 OR(Or)

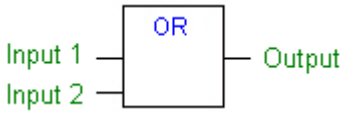


Figure 48 Or

1. Parameter Description

Variable	Description	Data Type
Input1	Input Variable	BOOL, BYTE, WORD, DWORD
Input2	Input Variable	BOOL, BYTE, WORD, DWORD
…		
Output	Output Variable	BOOL, BYTE, WORD, DWORD

Its Truth Table is as follows.

Input1	Input2	Output
0	0	0
0	1	1
1	0	1

Input1	Input2	Output
1	1	1

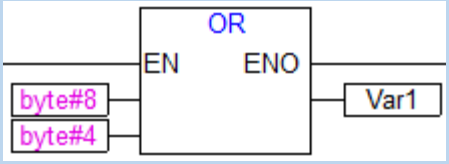
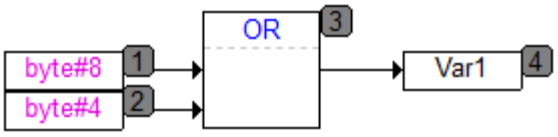
2. Description

This function block is to complete the bitwise OR of bit operands.

3. Application Example of Instructions

Variable Definition

No.	Name	Description	Data Type	Initial Value	Area
0001	Var1		BYTE	0	G Area

Programing language	Program
Ladder diagram (LD)	
Function Block Diagram(FBD)	

2.2.2.3 XOR(Exclusive Or)

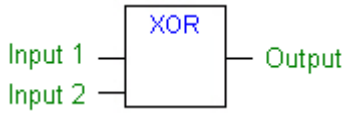


Figure 49 Exclusive Or

1. Parameter Description

Variable	Description	Data Type
Input1	Input Variable	BOOL, BYTE, WORD, DWORD
Input2	Input Variable	BOOL, BYTE, WORD, DWORD
Output	Output Variable	BOOL, BYTE, WORD, DWORD

Its Truth Table is as follows.

Input1	Input2	Output
0	0	0
0	1	1

Input1	Input2	Output
1	0	1
1	1	0

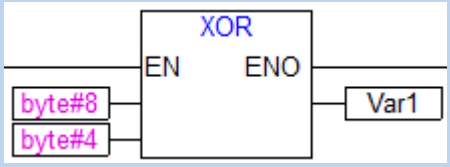
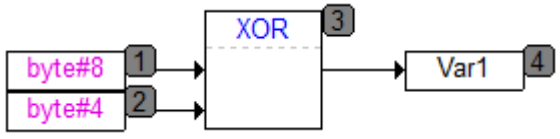
2. Description

This function block is to complete the bitwise XOR of bit operands.

3. Application Example of Instructions

Variable Definition

No.	Name	Description	Data Type	Initial Value	Area
0001	Var1		BYTE	0	G Area

Programing language	Program
Ladder diagram (LD)	
Function Block Diagram(FBD)	

2.2.2.4 NOT(Not)

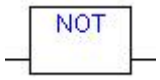


Figure 50 Not

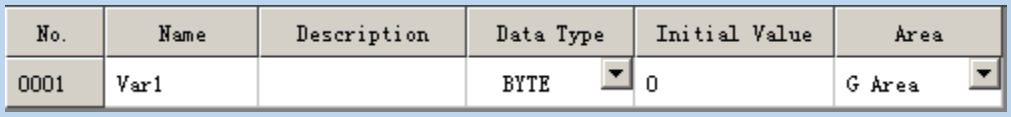
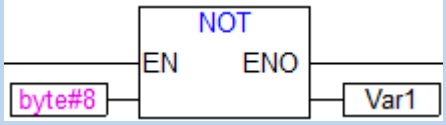
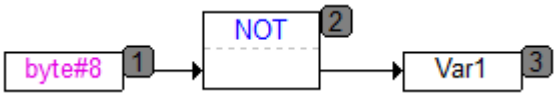
1. Parameter Description

Variable	Description	Data Type
Input	Input Variable	BOOL, BYTE, WORD, DWORD
Output	Output Variable	BOOL, BYTE, WORD, DWORD

2. Description

This function block is to complete the bitwise NOT of bit operands.

3. Application Example of Instructions

Variable Definition	
	
Programing language	Program
Ladder diagram (LD)	
Function Block Diagram(FBD)	

2.2.2.5 AND_FB(And)

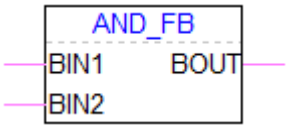


Figure 51 AND

1. Parameter Description

Variable	Description	Data Type
BIN1	Input 1	BOOL
BIN2	Input 2	BOOL
Output		
BOUT	Output	BOOL

Its Truth Table is as follows.

Input1	Input2	Output
0	0	0
0	1	0
1	0	0
1	1	1

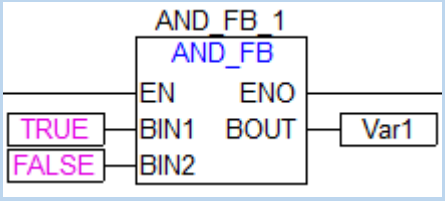
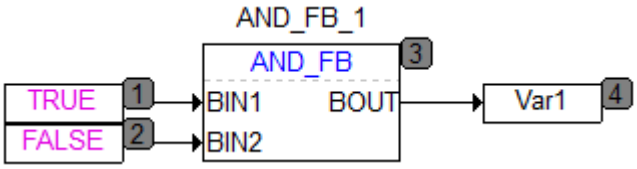
2. Description

This function block is to complete the AND operands of BOOL variables.

3. Application Example of Instructions

Variable Definition

No.	Name	Description	Data Type	Initial Value	Area
0001	AND_FB_1		AND_FB		G Area
0002	Var1		BOOL	FALSE	G Area

Programing language	Program
Ladder diagram (LD)	
Function Block Diagram(FBD)	

2.2.2.6 OR_FB(Or)

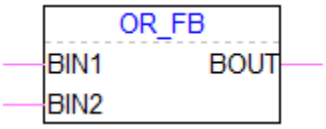


Figure 52 Or

1. Parameter Description

Variable	Description	Data Type
BIN1	Input 1	BOOL
BIN2	Input 2	BOOL
Output		
BOUT	Output	BOOL

Its Truth Table is as follows.

Input1	Input2	Output
0	0	0
0	1	1
1	0	1

Input1	Input2	Output
1	1	1

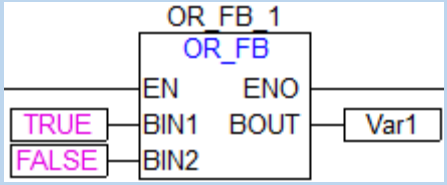
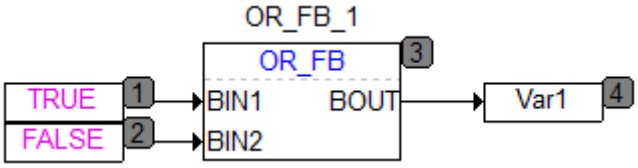
2. Description

This function block is to complete the OR operands of BOOL variables.

3. Application Example of Instructions

Variable Definition

No.	Name	Description	Data Type	Initial Value	Area
0001	OR_FB_1		OR_FB		G Area
0002	Var1		BOOL	FALSE	G Area

Programing language	Program
Ladder diagram (LD)	
Function Block Diagram(FBD)	

2.2.2.7 NOT_FB(Not)

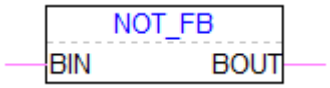


Figure 53 Not

1. Parameter Description

Variable	Description	Data Type
BIN	Input	BOOL
Output		
BOUT	Output	BOOL

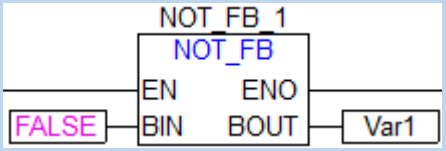
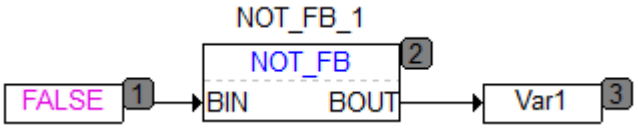
2. Description

This function block is to complete the NOT operands of BOOL variables.

3. Application Example of Instructions

Variable Definition

No.	Name	Description	Data Type	Initial Value	Area
0001	NOT_FB_1		NOT_FB		G Area
0002	Var1		BOOL	FALSE	G Area

Programing language	Program
Ladder diagram (LD)	
Function Block Diagram(FBD)	

2.2.2.8 AND3(And)

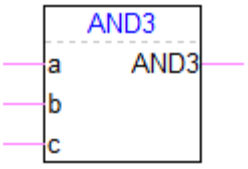


Figure 54 Take And

1. Parameter Description

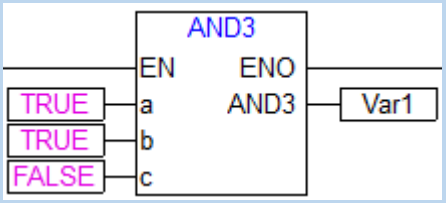
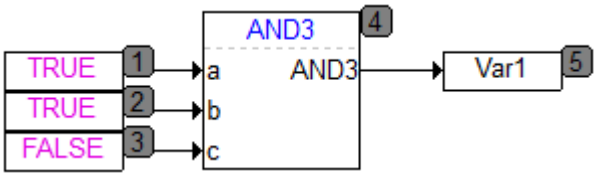
Variable	Data Type	Description	Initial Value
a	BOOL	Variable a	0
b	BOOL	Variable b	0
c	BOOL	Variable c	0

2. Application Example of Instructions

Variable Definition

No.	Name	Description	Data Type	Initial Value	Area
0001	Var1		BOOL	FALSE	G Area

Programing language	Program

Variable Definition	
Ladder diagram (LD)	
Function Block Diagram(FBD)	

2.2.2.9 OR3(Or)

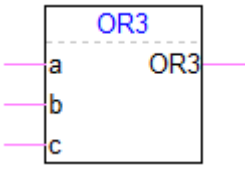
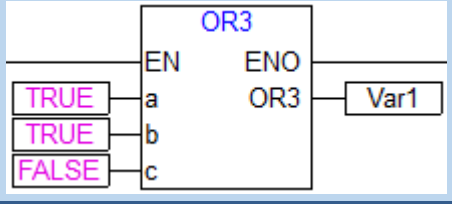


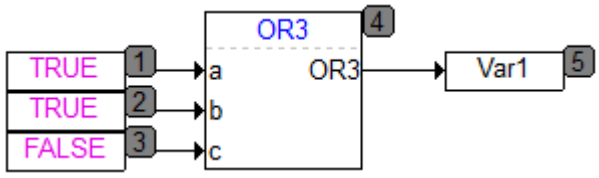
Figure 55 Take Or

1. Parameter Description

Variable	Data Type	Description	Initial Value
a	BOOL	Variable a	0
b	BOOL	Variable b	0
c	BOOL	Variable c	0

2. Application Example of Instructions

Variable Definition					
No.	Name	Description	Data Type	Initial Value	Area
0001	Var1		BOOL	FALSE	G Area
Programing language	Program				
Ladder diagram (LD)					

Variable Definition	
Function Block Diagram(FBD)	

2.2.3 Comparison Operation

2.2.3.1 GT(Greater Than)

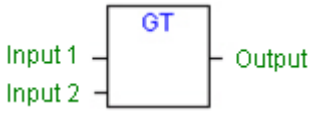


Figure 56 Greater Than

1. Parameter Description

Input	Description	Data Type
Input1	Input Variable	BYTE, WORD, DWORD, SINT, INT, DINT, REAL, LREAL, TIME, DATE, TOD, DT
Input2	Input Variable	BYTE, WORD, DWORD, SINT, INT, DINT, REAL, LREAL, TIME, DATE, TOD, DT
Output		
Output	Output Variable	BOOL

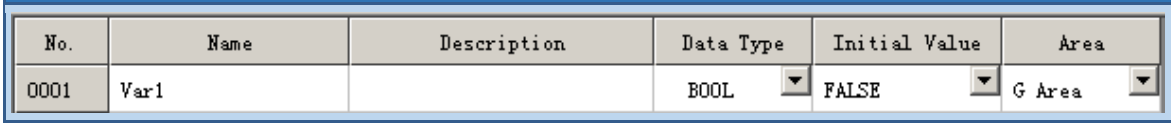
- The data types of the two inputs of compare operations must be the same.

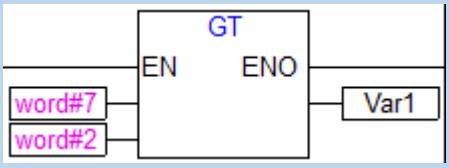
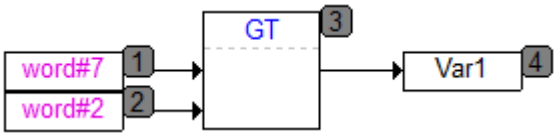
2. Description

When input1 is greater than input2, output is TRUE;

When input1 is less than or equal to input2, output is FALSE.

3. Application Example of Instructions

Variable Definition	
	
Programing language	Program

Variable Definition	
Ladder diagram (LD)	
Function Block Diagram(FBD)	

2.2.3.2 LT(Less Than)

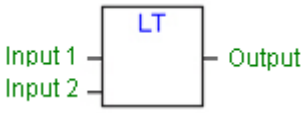


Figure 57 Less Than

1. Parameter Description

Input	Description	Data Type
Input1	Input	BYTE, WORD, DWORD, SINT, INT, DINT, REAL, LREAL, TIME, DATE, TOD, DT
Input2	Input	BYTE, WORD, DWORD, SINT, INT, DINT, REAL, LREAL, TIME, DATE, TOD, DT
Output		
Output	Output	BOOL

- The data types of the two inputs of compare operations must be the same.

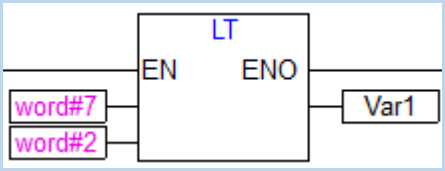
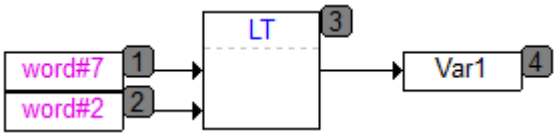
2. Description

When input1 is less than input2, output is TRUE;

When input1 is greater than or equal Input2, output is FALSE.

3. Application Example of Instructions

Variable Definition					
No.	Name	Description	Data Type	Initial Value	Area
0001	Var1		BOOL	FALSE	G Area
Programing language	Program				

Variable Definition	
Ladder diagram (LD)	
Function Block Diagram(FBD)	

2.2.3.3 GE(Greater or Equal)

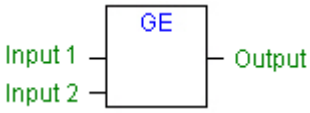


Figure 58 Greater or Equal

1. Parameter Description

Input	Description	Data Type
Input1	Input	BYTE, WORD, DWORD, SINT, INT, DINT, REAL, LREAL, TIME, DATE, TOD, DT
Input2	Input	BYTE, WORD, DWORD, SINT, INT, DINT, REAL, LREAL, TIME, DATE, TOD, DT
Output		
Output	Output	BOOL

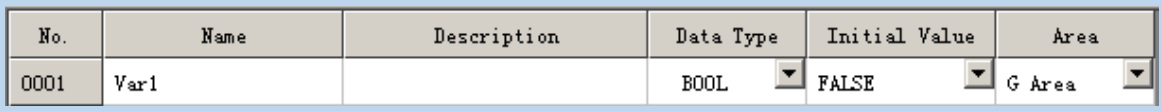
- The data types of the two inputs of compare operations must be the same.

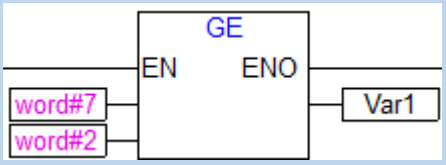
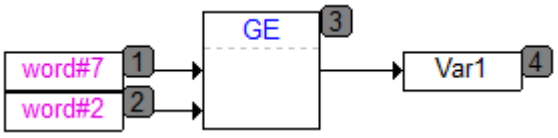
2. Description

When input1 is greater than or equal to input2, Output is TRUE;

When input1 is less than input2, output is FALSE.

3. Application Example of Instructions

Variable Definition	
	
Programing language	Program

Variable Definition	
Ladder diagram (LD)	
Function Block Diagram(FBD)	

2.2.3.4 LE(Less or Equal)

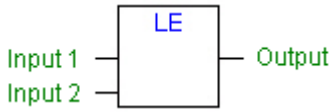


Figure 59 Less or Equal

1. Parameter Description

Input	Description	Data Type
Input1	Input	BYTE, WORD, DWORD, SINT, INT, DINT, REAL, LREAL, TIME, DATE, TOD, DT
Input2	Input	BYTE, WORD, DWORD, SINT, INT, DINT, REAL, LREAL, TIME, DATE, TOD, DT
Output		
Output	Output	BOOL

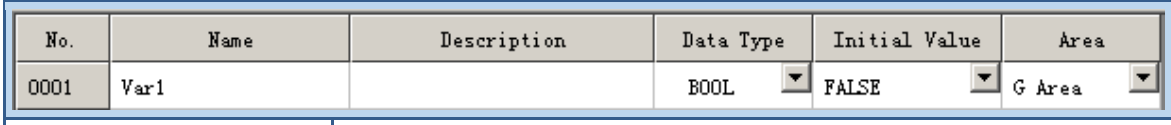
- The data types of the two inputs of compare operations must be the same.

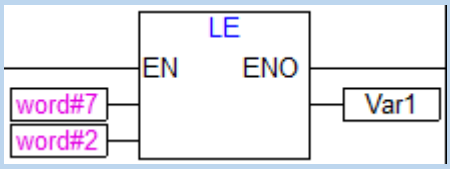
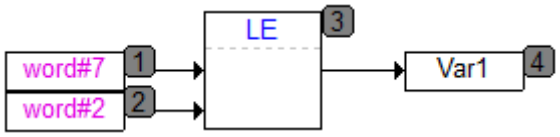
2. Description

When input1 is less than or equal to input2, output is TRUE;

When input1 is greater than input2, output is FALSE.

3. Application Example of Instructions

Variable Definition	
	
Programing language	Program

Variable Definition	
Ladder diagram (LD)	
Function Block Diagram(FBD)	

2.2.3.5 EQ(Equal)

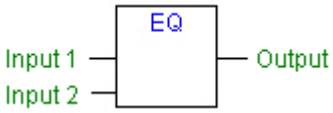


Figure 60 Equal

1. Parameter Description

Input	Description	Data Type
Input1	Input	BOOL, BYTE, WORD, DWORD, SINT, INT, DINT, REAL, LREAL, TIME, DATE, TOD, DT
Input2	Input	BOOL, BYTE, WORD, DWORD, SINT, INT, DINT, REAL, LREAL, TIME, DATE, TOD, DT
Output		
Output	Output	BOOL

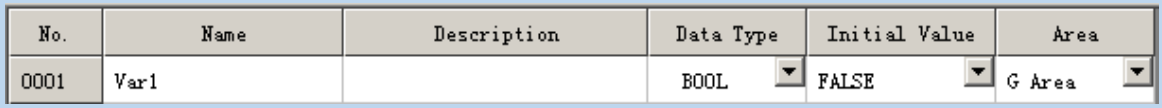
- The data types of the two inputs of compare operations must be the same.

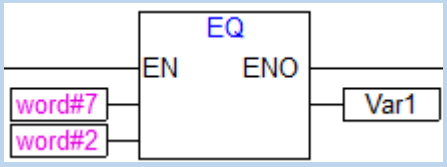
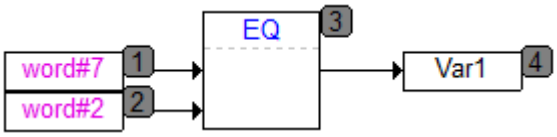
2. Description

When input1 is equal to input2, output is TRUE;

When input1 is not equal to Input2, output is FALSE.

3. Application Example of Instructions

Variable Definition	
	
Programing language	Program

Variable Definition	
Ladder diagram (LD)	
Function Block Diagram(FBD)	

2.2.3.6 NE(Not Equal)

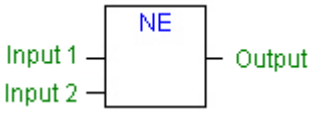


Figure 61 Not Equal

1. Parameter Description

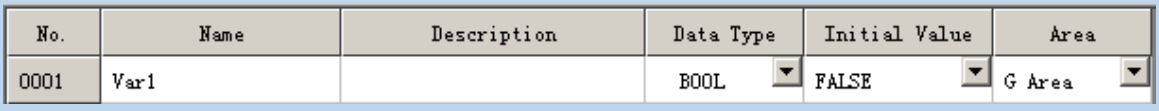
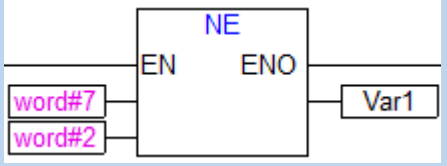
Input	Description	Data Type
Input1	Input	BOOL, BYTE, WORD, DWORD, SINT, INT, DINT, REAL, LREAL, TIME, DATE, TOD, DT
Input2	Input	BOOL, BYTE, WORD, DWORD, SINT, INT, DINT, REAL, LREAL, TIME, DATE, TOD, DT
Output		
Output	Output	BOOL

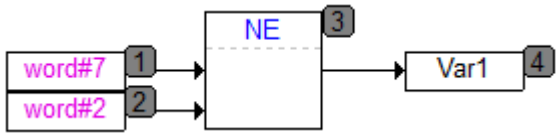
2. Description

When input1 is not equal to input2, output is TRUE;

When input1 is equal to Input2, output is FALSE.

3. Application Example of Instructions

Variable Definition	
	
Programing language	Program
Ladder diagram (LD)	

Variable Definition	
Function Block Diagram(FBD)	

2.2.3.7 GT_DINT_FB(Greater Than)

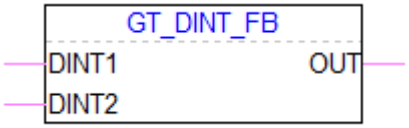


Figure 62 Greater Than

1. Parameter Description

Input	Description	Data Type
DINT1	Input 1	DINT
DINT2	Input 2	DINT
Output		
OUT	Output	BOOL

2. Description

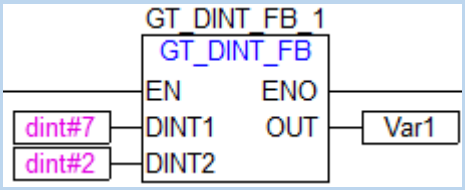
When input1 is greater than input2, output is TRUE;

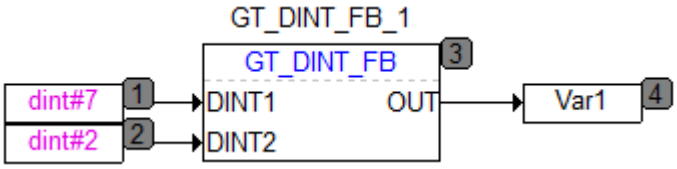
When input1 is less than or equal to input2, output is FALSE;

3. Application Example of Instructions

Variable Definition

No.	Name	Description	Data Type	Initial Value	Area
0001	GT_DINT_FB_1		GT_DINT_FB		G Area
0002	Var1		BOOL	FALSE	G Area

Programing language	Program
Ladder diagram (LD)	

Variable Definition	
Function Block Diagram(FBD)	

2.2.3.8 LT_DINT_FB(Less Than)

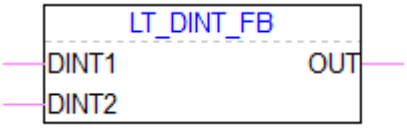


Figure 63 Less Than

1. Parameter Description

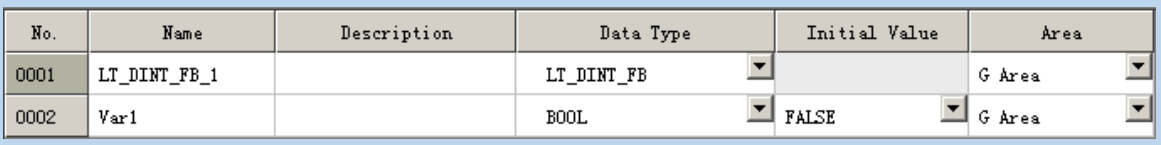
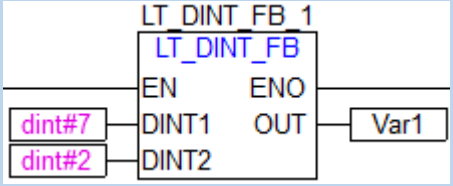
Input	Description	Data Type
DINT1	Input1	DINT
DINT2	Input2	DINT
Output		
OUT	Output	BOOL

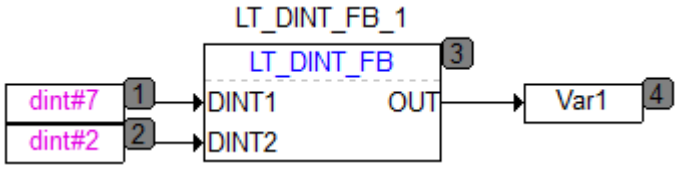
2. Description

When input1 is less than input2, output is TRUE;

When input1 is greater than or equal Input2, output is FALSE.

3. Application Example of Instructions

Variable Definition	
	
Programing language	Program
Ladder diagram (LD)	

Variable Definition	
Function Block Diagram(FBD)	

2.2.3.9 GE_DINT_FB(Greater or Equal)

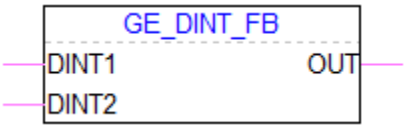


Figure 64 Greater and Equal

1. Parameter Description

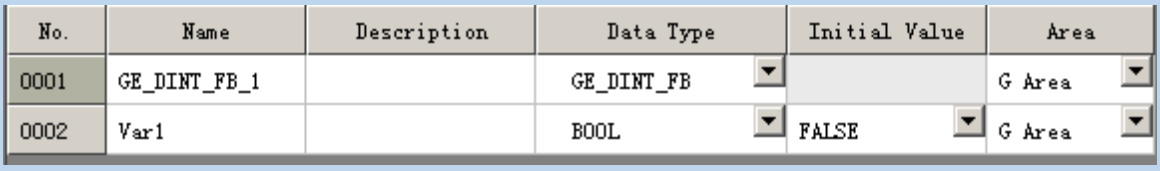
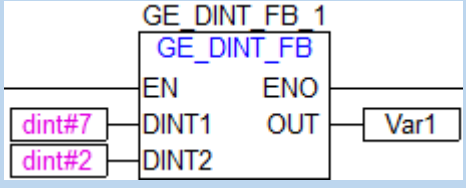
Input	Description	Data Type
DINT1	Input1	DINT
DINT2	Input2	DINT
Output		
OUT	Output	BOOL

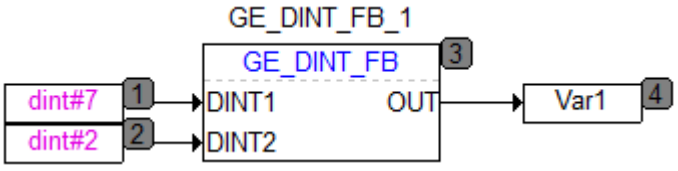
2. Description

When input1 is greater than or equal to input2, Output is TRUE;

When input1 is less than input2, output is FALSE.

3. Application Example of Instructions

Variable Definition	
	
Programing language	Program
Ladder diagram (LD)	

Variable Definition	
Function Block Diagram(FBD)	

2.2.3.10 LE_DINT_FB(Less or Equal)

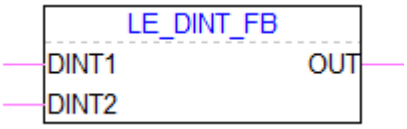


Figure 65 Less and Equal

1. Parameter Description

Input	Description	Data Type
DINT1	Input1	DINT
DINT2	Input2	DINT
Output		
OUT	Output	BOOL

2. Description

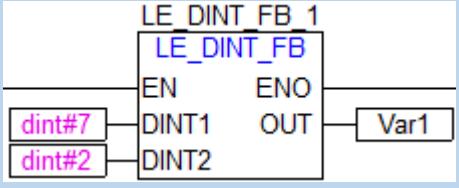
When input1 is equal to input2, output is TRUE;

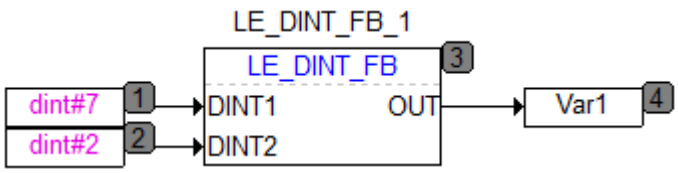
When input1 is not equal to Input2, output is FALSE.

3. Application Example of Instructions

Variable Definition

No.	Name	Description	Data Type	Initial Value	Area
0001	LE_DINT_FB_1		LE_DINT_FB		G Area
0002	Var1		BOOL	FALSE	G Area

Programing language	Program
Ladder diagram (LD)	

Variable Definition	
Function Block Diagram(FBD)	

2.2.3.11 EQ_DINT_FB(Equal)

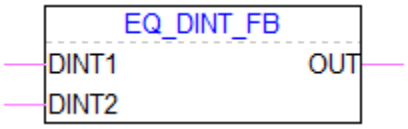


Figure 66 Equal

1. Parameter Description

Input	Description	Data Type
DINT1	Input1	DINT
DINT2	Input2	DINT
Output		
OUT	Output	BOOL

2. Description

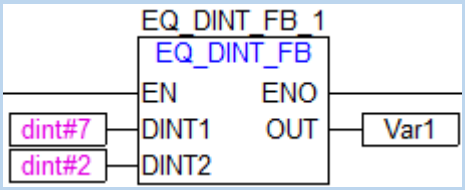
When input1 is equal to input2, output is TRUE;

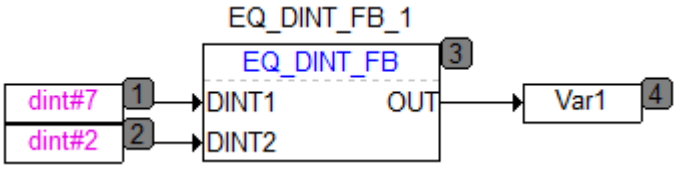
When input1 is not equal to Input2, output is FALSE.

3. Application Example of Instructions

Variable Definition

No.	Name	Description	Data Type	Initial Value	Area
0001	EQ_DINT_FB_1		EQ_DINT_FB		G Area
0002	Var1		BOOL	FALSE	G Area

Programing language	Program
Ladder diagram (LD)	

Variable Definition	
Function Block Diagram(FBD)	

2.2.3.12 GT_REAL_FB(Greater Than)

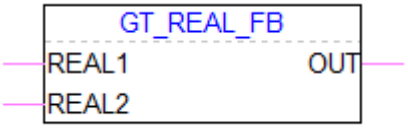


Figure 67 Greater than

1. Parameter Description

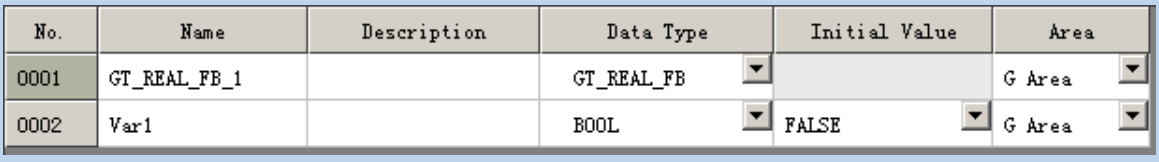
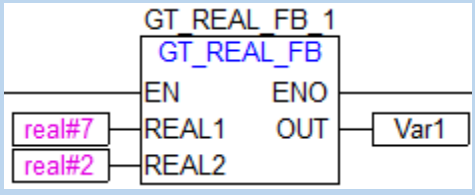
Input	Description	Data Type
REAL1	Input1	REAL
REAL2	Input2	REAL
Output		
OUT	Output	BOOL

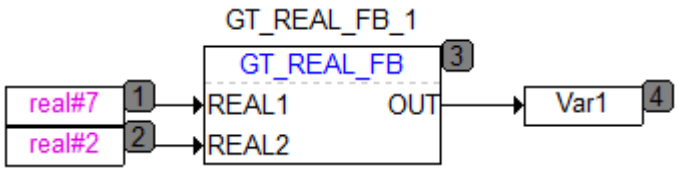
2. Description

When input1 is greater than input2, output is TRUE;

When input1 is less than or equal to input2, output is FALSE.

3. Application Example of Instructions

Variable Definition	
	
Programing language	Program
Ladder diagram (LD)	

Variable Definition	
Function Block Diagram(FBD)	

2.2.3.13 LT_REAL_FB(Less Than)

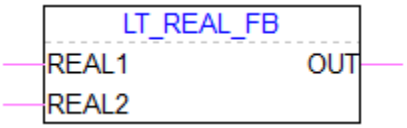


Figure 68 Less than

1. Parameter Description

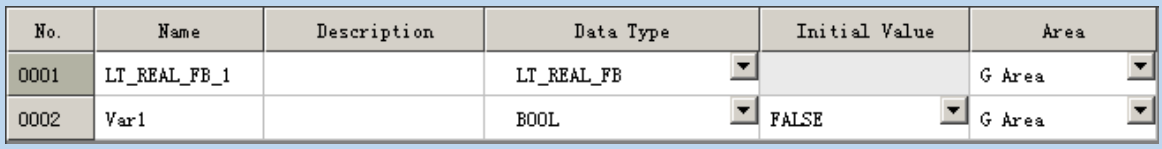
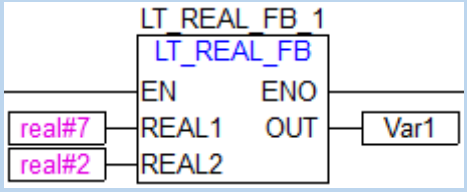
Input	Description	Data Type
REAL1	Input1	REAL
REAL2	Input2	REAL
Output		
OUT	Output	BOOL

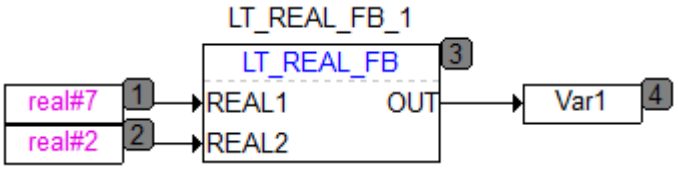
2. Description

When input1 is less than input2, output is TRUE;

When input1 is greater than or equal input2, output is FALSE.

3. Application Example of Instructions

Variable Definition	
	
Programing language	Program
Ladder diagram (LD)	

Variable Definition	
Function Block Diagram(FBD)	

2.2.3.14 GE_REAL_FB(Greater or Equal)

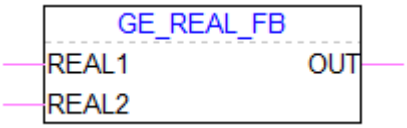


Figure 69 Greater and Equal

1. Parameter Description

Input	Description	Data Type
REAL1	Input1	REAL
REAL2	Input2	REAL
Output		
OUT	Output	BOOL

2. Description

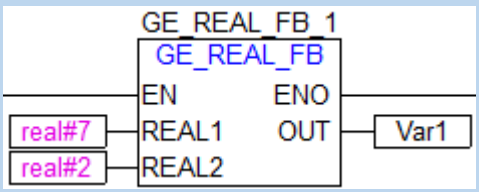
When input1 is greater than or equal to input2, Output is TRUE;

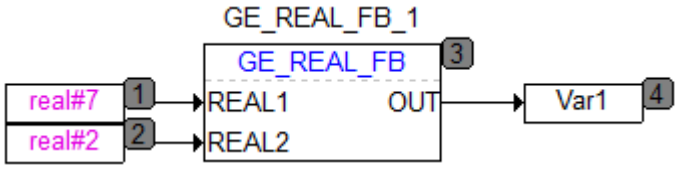
When input1 is less than input2, output is FALSE.

3. Application Example of Instructions

Variable Definition	

No.	Name	Description	Data Type	Initial Value	Area
0001	GE_REAL_FB_1		GE_REAL_FB		G Area
0002	Var1		BOOL	FALSE	G Area

Programing language	Program
Ladder diagram (LD)	

Variable Definition	
Function Block Diagram(FBD)	

2.2.3.15 LE_REAL_FB(Less or Equal)

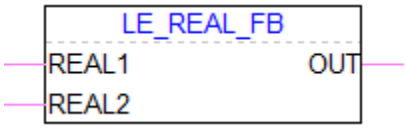


Figure 70 Less and Equal

1. Parameter Description

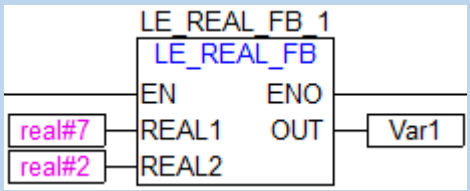
Input	Description	Data Type
REAL1	Input1	REAL
REAL2	Input2	REAL
Output		
OUT	Output	BOOL

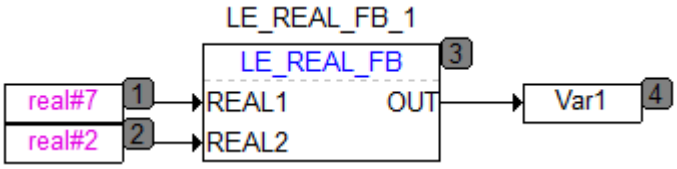
2. Description

When input1 is equal to input2, output is TRUE;

When input1 is not equal to input2, output is FALSE.

3. Application Example of Instructions

Variable Definition					
No.	Name	Description	Data Type	Initial Value	Area
0001	LE_REAL_FB_1		LE_REAL_FB		G Area
0002	Var1		BOOL	FALSE	G Area
Programing language	Program				
Ladder diagram (LD)					

Variable Definition	
Function Block Diagram(FBD)	

2.2.3.16 EQ_REAL_FB(Equal)

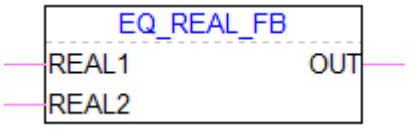


Figure 71 Equal

1. Parameter Description

Input	Description	Data Type
REAL1	Input1	REAL
REAL2	Input2	REAL
Output		
OUT	Output	BOOL

2. Description

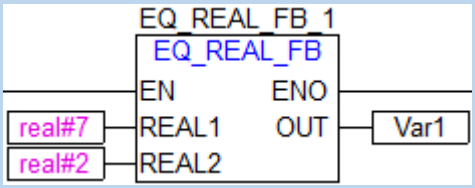
When input1 is equal to input2, output is TRUE;

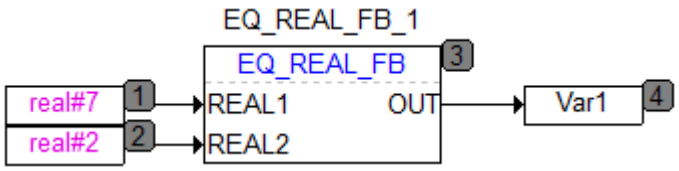
When input1 is not equal to input2, output is FALSE.

3. Application Example of Instructions

Variable Definition

No.	Name	Description	Data Type	Initial Value	Area
0001	EQ_REAL_FB_1		EQ_REAL_FB		G Area
0002	Var1		BOOL	FALSE	G Area

Programing language	Program
Ladder diagram (LD)	

Variable Definition	
Function Block Diagram(FBD)	

2.2.4 Select Operation

2.2.4.1 SEL(One Out of Two)

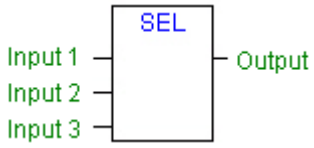


Figure 72 One Out of Two

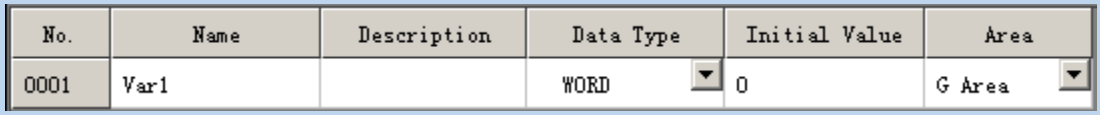
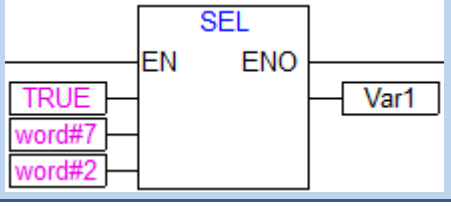
1. Parameter Description

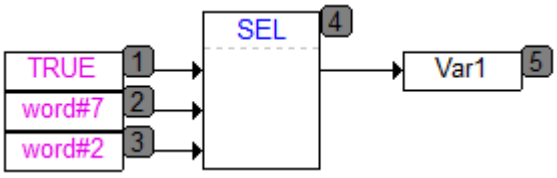
Input	Description	Data Type
Input1	Choose Variable	BOOL
Input2	Selected Variable	Arbitrary
Input3	Selected Variable	Arbitrary
Output		
Output	When Input1 is FALSE Input2 as Output; when Input1 is TRUE Input3 as Output.	Arbitrary, but in operation need to ensure input and output of type match.

2. Description

When Input 1 is FALSE, it will output input2; When Input 1 is TRUE, it will output input3.

3. Application Example of Instructions

Variable Definition	
	
Programing language	Program
Ladder diagram (LD)	

Variable Definition	
Function Block Diagram(FBD)	

2.2.4.2 MAX(Maximum)

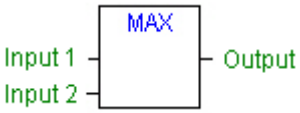


Figure 73 Maximum

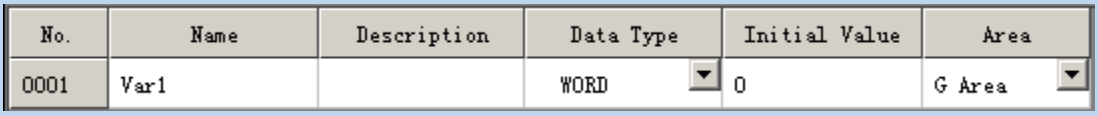
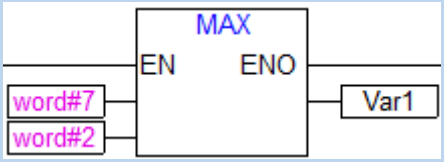
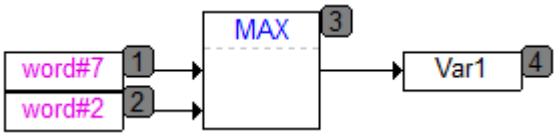
1. Parameter Description

Input	Description	Data Type
Input1	Input Variable	Arbitrary, except BOOL
Input2	Input Variable	Arbitrary, except BOOL
Output		
Output	Output Variable	Arbitrary, but in operation need to ensure input type.

2. Description

This function block output the maximum of two inputs.

3. Application Example of Instructions

Variable Definition	
	
Programing language	Program
Ladder diagram (LD)	
Function Block Diagram(FBD)	

2.2.4.3 MIN(Minimum)

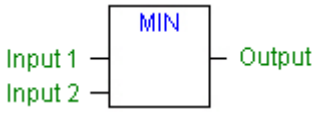


Figure 74 Minimum

1. Parameter Description

Input	Description	Data Type
Input1	Input	Arbitrary, except BOOL
Input2	Input	Arbitrary, except BOOL
Output		
Output	Output Variable	Arbitrary, but in operation need to ensure input type.

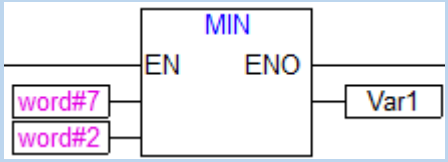
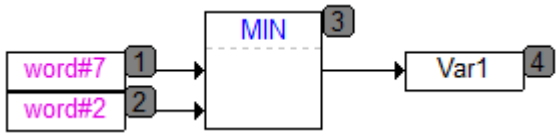
2. Description

This function block output the minimum of two inputs.

3. Application Example of Instructions

Variable Definition

No.	Name	Description	Data Type	Initial Value	Area
0001	Var1		WORD	0	G Area

Programing language	Program
Ladder diagram (LD)	
Function Block Diagram(FBD)	

2.2.5 Shift Operation

2.2.5.1 SHL(Shift Left)

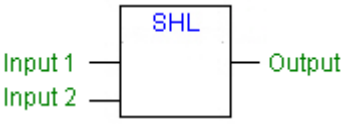


Figure 75 Shift Left

1. Parameter Description

Input	Description	Data Type
Input1	Operated variable	DWORD, DINT
Input2	Shifting Digit	BYTE
Output		
Output	Result	DWORD, DINT

2. Description

Input variable is shifted left in accordance with prescribed bits, and the right vacancy is filled 0.

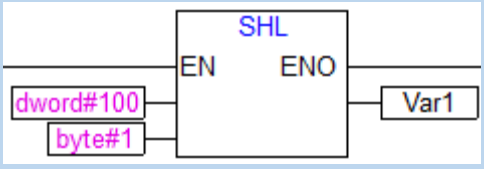
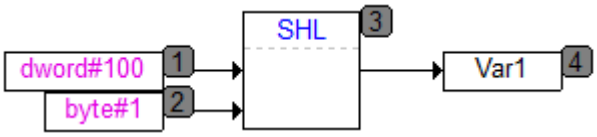
3. Example

4 DWORD type, becomes 8, after one bit moving to the left.

4. Application Example of Instructions

Variable Definition

No.	Name	Description	Data Type	Initial Value	Area
0001	Var1		DWORD	0	G Area

Programing language	Program
Ladder diagram (LD)	
Function Block Diagram(FBD)	

2.2.5.2 SHR(Shift Right)

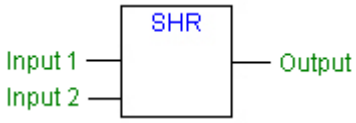


Figure 76 Shift Right

1. Parameter Description

Input	Description	Data Type
Input1	Operated variable	DWORD, DINT

Input	Description	Data Type
Input2	Shifting digit	BYTE
Output		
Output	Result	DWORD, DINT

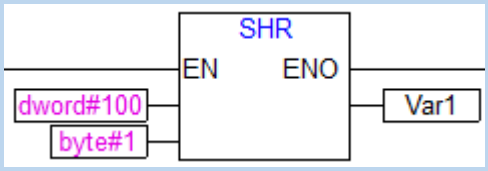
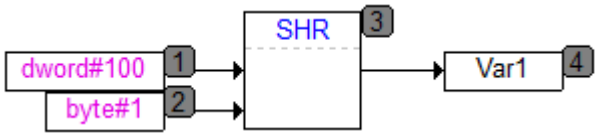
2. Description

Input variable is shifted right in accordance with prescribed bits, and the left vacancy is filled 0.

3. Application Example of Instructions

Variable Definition

No.	Name	Description	Data Type	Initial Value	Area
0001	Var1		DWORD	0	G Area

Programing language	Program
Ladder diagram (LD)	
Function Block Diagram(FBD)	

2.2.5.3 ROL(Ring Shift Left)

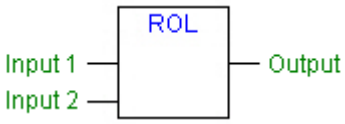


Figure 77 Ring Shift Left

1. Parameter Description

Input	Description	Data Type
Input1	Operated variable	DWORD, DINT
Input2	Shifting digit	BYTE
Output		
Output	Result	DWORD, DINT

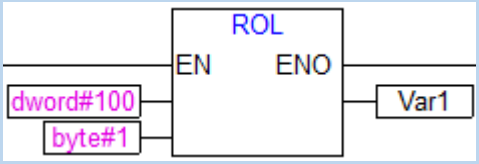
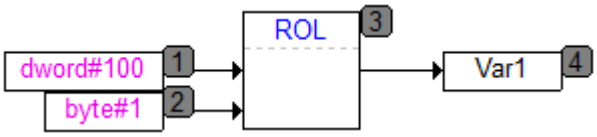
2. Description

Input variable is shifted left in accordance with prescribed bits, and the left bit is inserted right.

3. Application Example of Instructions

Variable Definition

No.	Name	Description	Data Type	Initial Value	Area
0001	Var1		DWORD	0	G Area

Programing language	Program
Ladder diagram (LD)	
Function Block Diagram(FBD)	

2.2.5.4 ROR(Ring Shift Right)

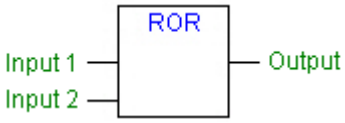


Figure 78 Ring Shift Right

1. Parameter Description

Input	Description	Data Type
Input1	Operated variable	DWORD, DINT
Input2	Shifting digit	BYTE
Output		
Output	Result	DWORD, DINT

2. Description

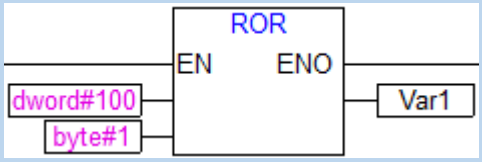
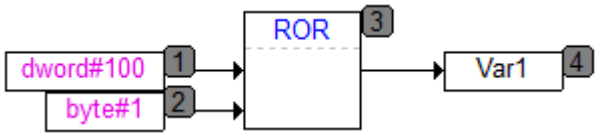
Input variable is shifted right in accordance with prescribed bits, and the right bit is inserted left.

3. Application Example of Instructions

Variable Definition

No.	Name	Description	Data Type	Initial Value	Area
0001	Var1		DWORD	0	G Area

Programing language	Program

Variable Definition	
Ladder diagram (LD)	
Function Block Diagram(FBD)	

2.2.6 Data Type Conversion

2.2.6.1 BYTE_TO(BYTE Type Conversion)

1. Function

Convert BYTE type to other types.

2. Parameter Description

Input	Description	Data Type
Input	Input variable	BYTE
Output		
Output	Output variable	DATE, DINT, DT, DWORD, INT, SINT, TIME, TOD, WORD,REAL,LREAL

3. Description

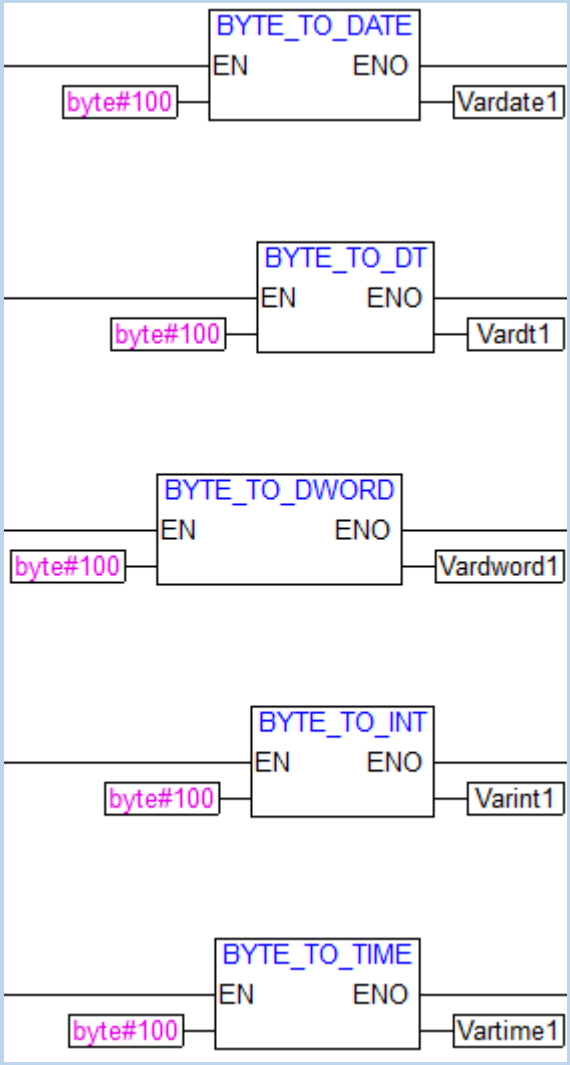
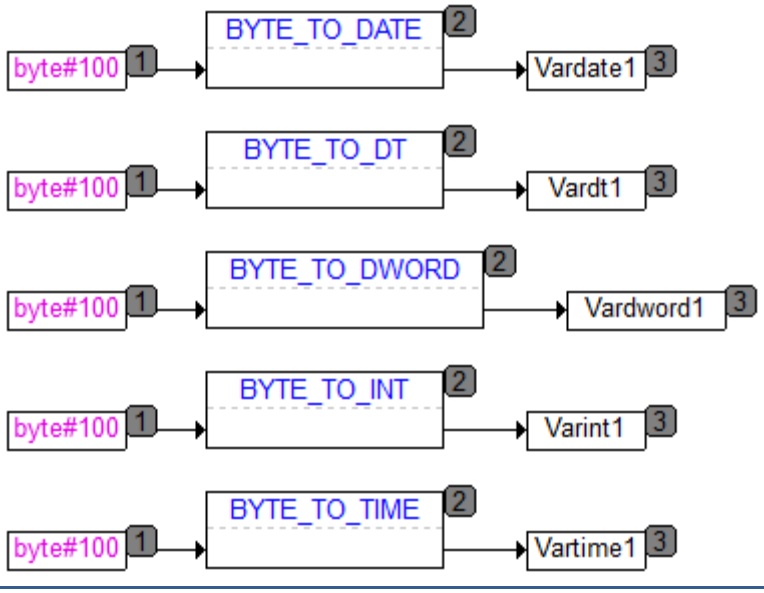
From byte type convert to other types (BYTE_TO_xxx),xxx means other data Type, it completes the conversion of byte type convert to other types.

4. Application Example of Instructions

Variable Definition

No.	Name	Description	Data Type	Initial Value	Area
0001	Vardate1		DATE	D#1970-01-01	G Area
0002	Vardt1		DT	DT#1970-01-01-00:00:00	G Area
0003	Vardword1		DWORD	0	G Area
0004	Varint1		INT	0	G Area
0005	Vartime1		TIME	T#0MS	G Area

Programing language	Program

Variable Definition	
Ladder diagram (LD)	
Function Block Diagram(FBD)	

2.2.6.2 WORD_TO(WORD Type Conversion)

1. Function

Convert WORD type to other types.

2. Parameter Description

Input	Description	Data Type
Input	Input Variable	WORD
Output		
Output	Output Variable	BYTE, DATE, DINT, DT, DWORD, INT, SINT, TIME, TOD, REAL, LREAL

3. Description

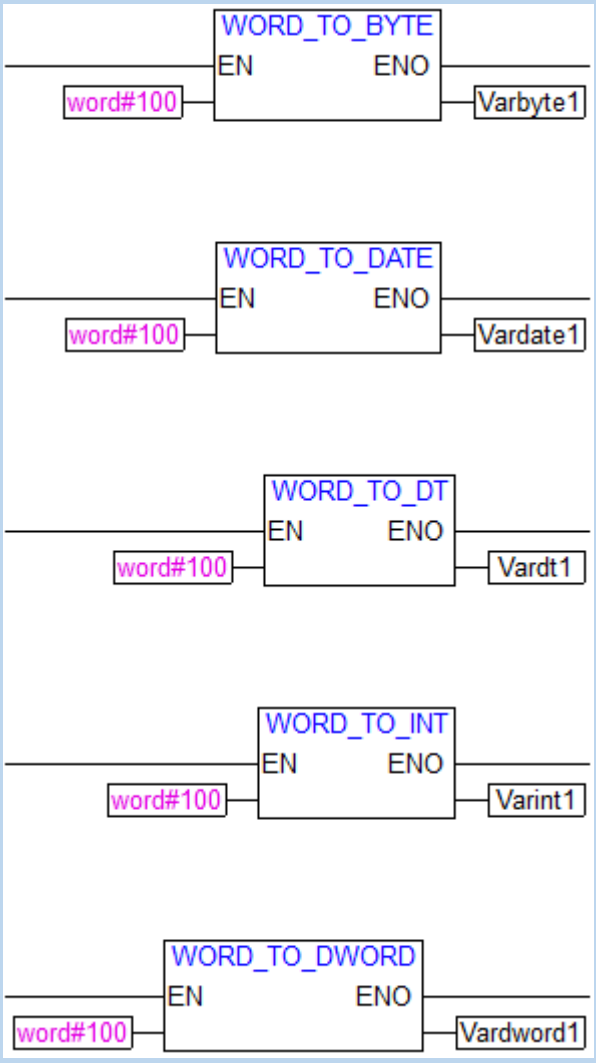
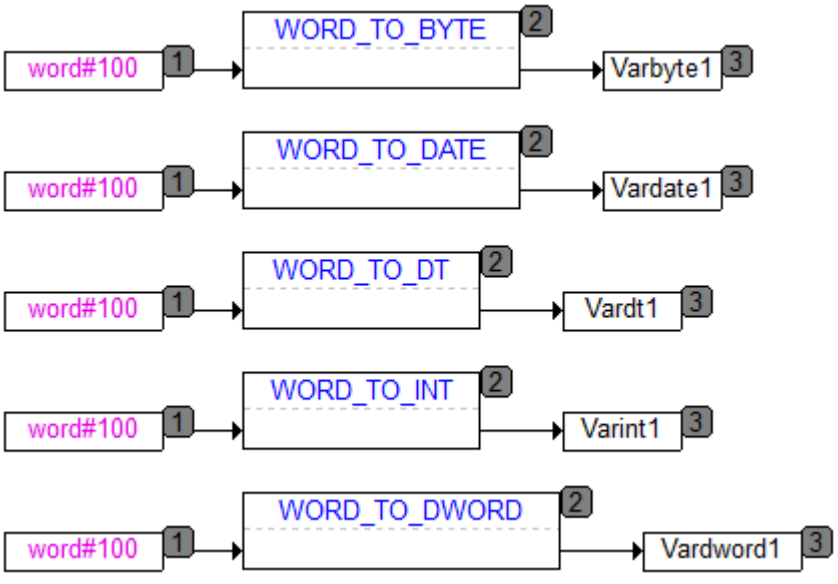
From WORD convert to other types (WORD_TO_xxx), xxx means other data type, it completes the conversion of WORD type convert to other types.

4. Application Example of Instructions

Variable Definition

No.	Name	Description	Data Type	Initial Value	Area
0001	Vardate1		DATE	D#1970-01-01	G Area
0002	Vardt1		DT	DT#1970-01-01-00:00:00	G Area
0003	Vardword1		DWORD	0	G Area
0004	Varint1		INT	0	G Area
0005	Varbyte1		BYTE	0	G Area

Programing language	Program

Variable Definition	
Ladder diagram (LD)	
Function Block Diagram(FBD)	

2.2.6.3 DWORD_TO(DWORD Type Conversion)

1. Function

Convert DWORD type to other types.

2. Parameter Description

Input	Description	Data Type
Input	Input Variable	DWORD
Output		
Output	Output Variable	BYTE, DATE, DINT, DT, INT, LREAL, REAL, SINT, TIME, TOD, WORD

3. Description

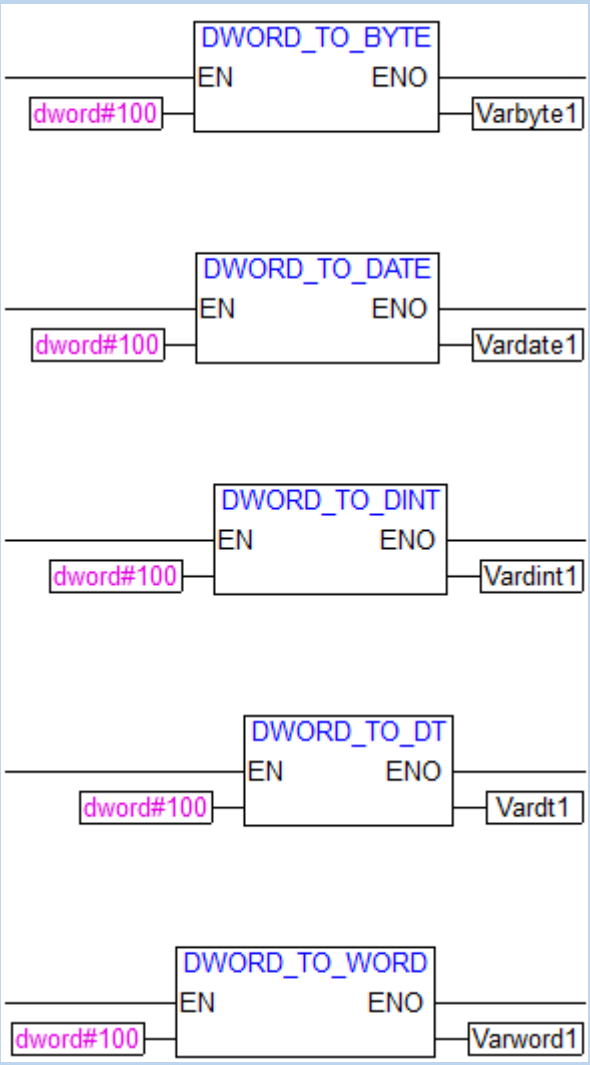
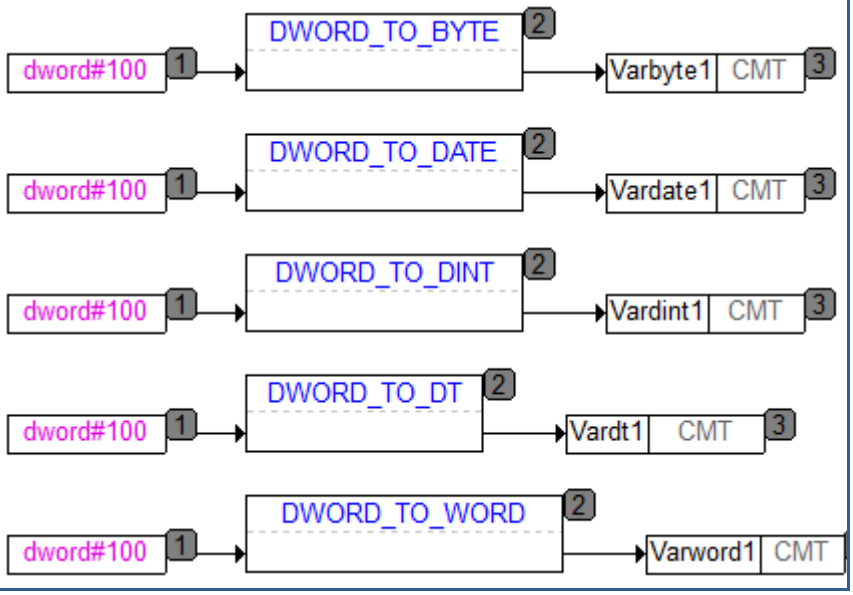
From DWORD convert to other types (DWORD_TO_xxx),xxx means other data type, it completes the conversion of DWORD type converts to other types.

4. Application Example of Instructions

Variable Definition

No.	Name	Description	Data Type	Initial Value	Area
0001	Vardate1		DATE	D#1970-01-01	G Area
0002	Vardt1		DT	DT#1970-01-01-00:00:00	G Area
0003	Varbyte1		BYTE	0	G Area
0004	Vardint1		DINT	0	G Area
0005	Varword1		WORD	0	G Area

Programing language	Program

Variable Definition	
Ladder diagram (LD)	
Function Block Diagram(FBD)	

2.2.6.4 SINT_TO(SINT Type Conversion)

1. Function

Convert SINT type to other types.

2. Parameter Description

Input	Description	Data Type
Input	Input Variable	SINT
Output		
Output	Output Variable	BYTE, DATE, DINT, DT, DWORD, INT, TIME, TOD, WORD, REAL, LREAL

3. Description

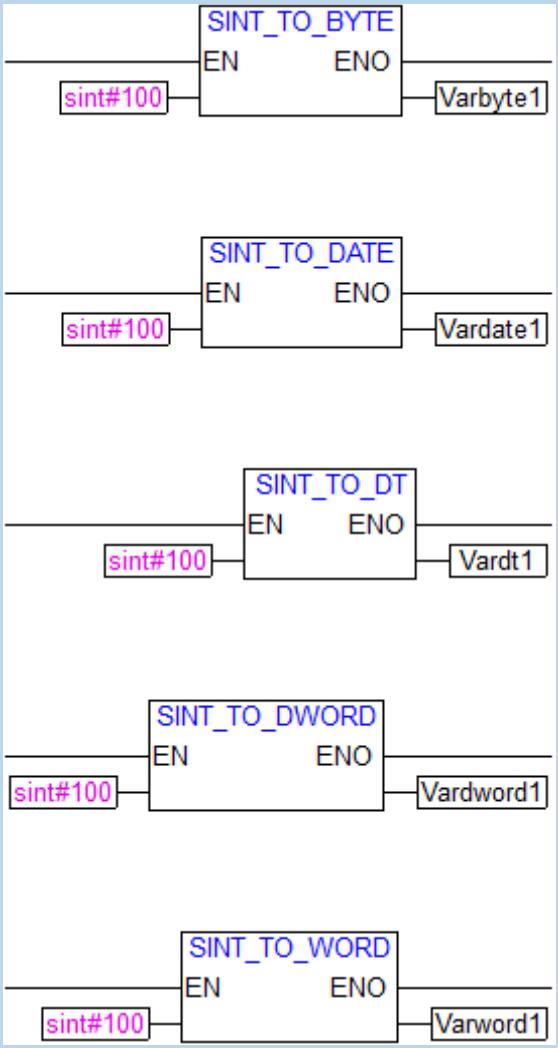
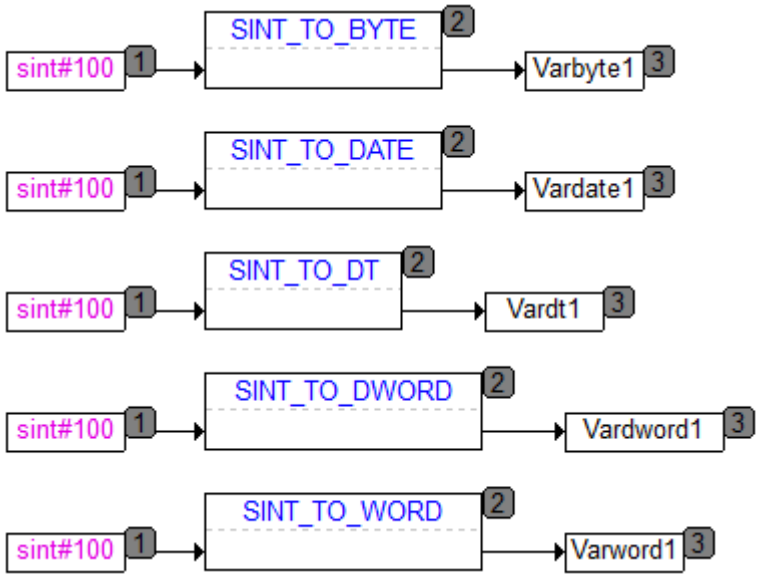
From SINT convert to other types (SINT_TO_xxx), xxx means other data type, it completes the conversion of SINT type convert to other types.

4. Application Example of Instructions

Variable Definition

No.	Name	Description	Data Type	Initial Value	Area
0001	Vardate1		DATE	D#1970-01-01	G Area
0002	Vardt1		DT	DT#1970-01-01-00:00:00	G Area
0003	Varbyte1		BYTE	0	G Area
0004	Varword1		WORD	0	G Area
0005	Vardword1		DWORD	0	G Area

Programing language	Program

Variable Definition	
Ladder diagram (LD)	
Function Block Diagram(FBD)	

2.2.6.5 INT_TO(INT Type Conversion)

1. Function

Convert INT type to other types.

2. Parameter Description

Input	Description	Data Type
Input	Input Variable	INT
Output		
Output	Output Variable	BYTE, DATE, DINT, DT, DWORD, SINT, TIME, TOD, WORD, REAL, LREAL

3. Description

From INT convert to other types (INT_TO xxx),xxx means other data type, the completed function seems like following:

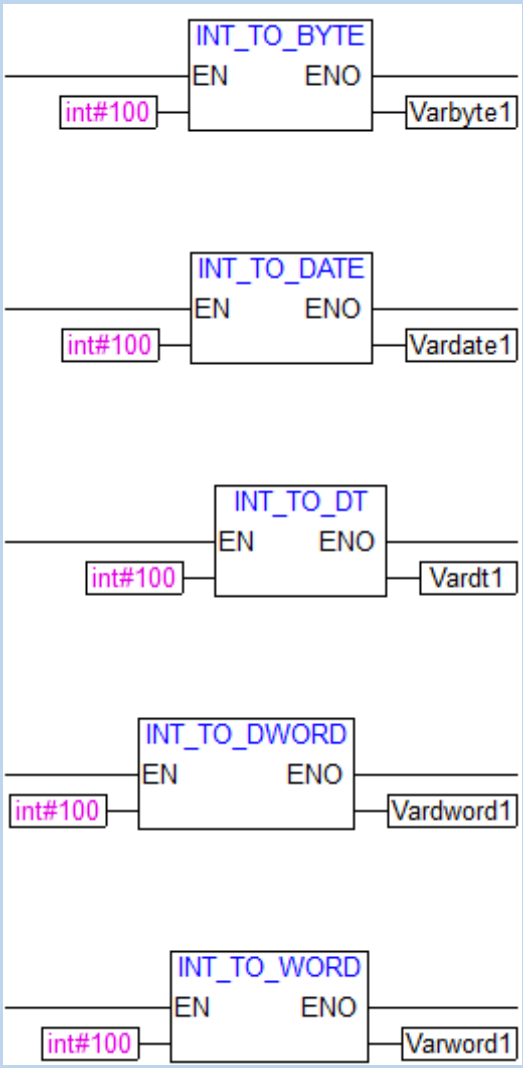
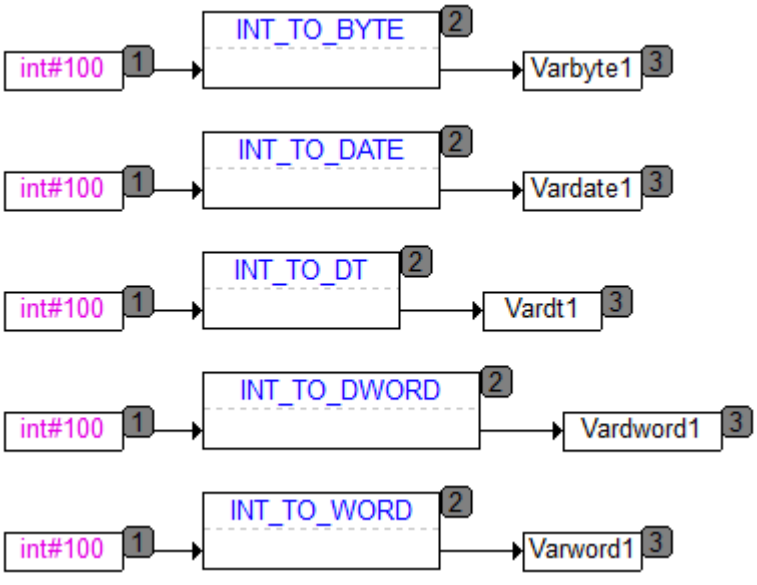
- It completes the conversion of INT type convert to other types.
- When converted value exceeds limit, it will ignore the first byte of the number;

4. Application Example of Instructions

Variable Definition

No.	Name	Description	Data Type	Initial Value	Area
0001	Vardate1		DATE	D#1970-01-01	G Area
0002	Vardt1		DT	DT#1970-01-01-00:00:00	G Area
0003	Varbyte1		BYTE	0	G Area
0004	Varword1		WORD	0	G Area
0005	Vardword1		DWORD	0	G Area

Programing language	Program

Variable Definition	
Ladder diagram (LD)	
Function Block Diagram(FBD)	

2.2.6.6 DINT_TO(DINT Type Conversion)

1. Function

Convert DINT type to other types.

2. Parameter Description

Input	Description	Data Type
Input	Input Variable	DINT
Output		
Output	Output Variable	BYTE, DATE, DT, DWORD, INT, LREAL, REAL, SINT, TIME, TOD, WORD

3. Description

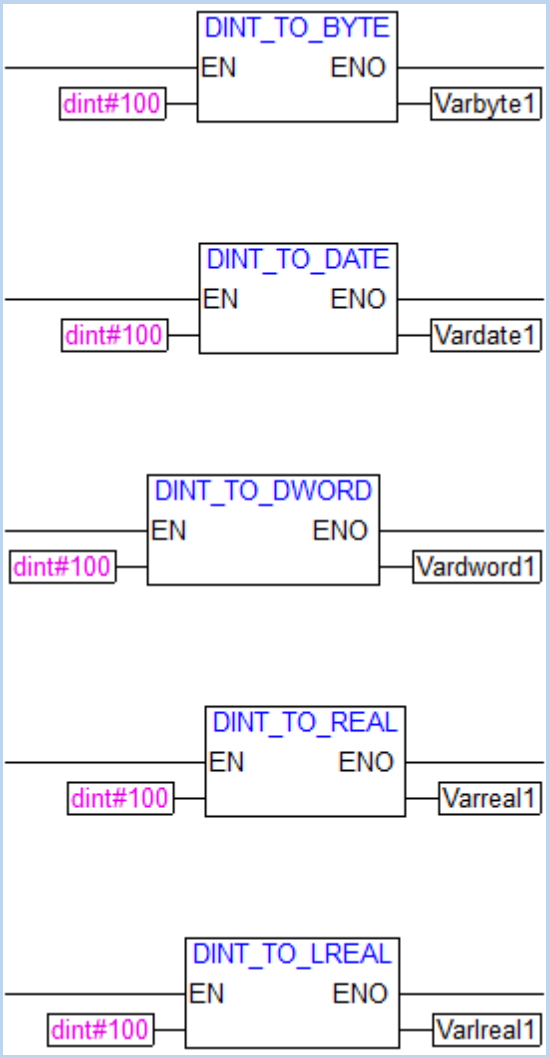
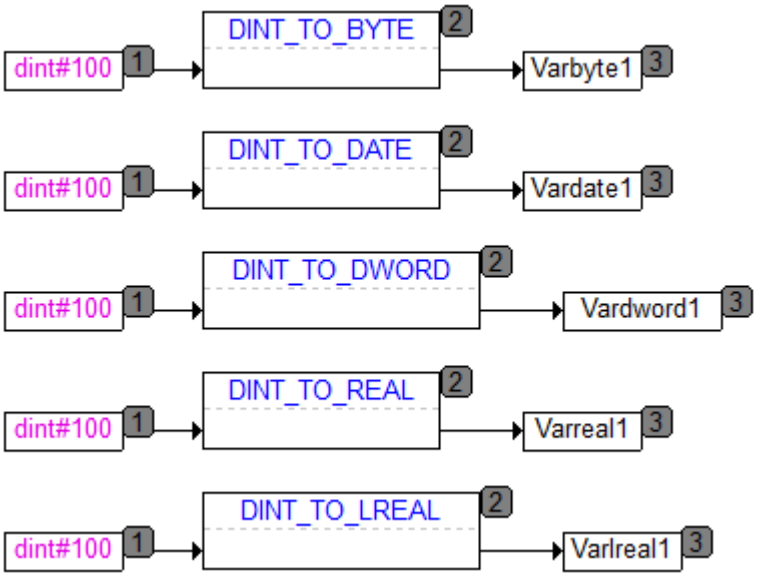
From DINT convert to other types (DINT _TO_xxx), xxx means other data type, it completes the conversion of DINT type converts to other types.

4. Application Example of Instructions

Variable Definition

No.	Name	Description	Data Type	Initial Value	Area
0001	Vardate1		DATE	D#1970-01-01	G Area
0002	Varbyte1		BYTE	0	G Area
0003	Vardword1		DWORD	0	G Area
0004	Varreal1		REAL	0	G Area
0005	Varlreal1		LREAL	0	G Area

Programing language	Program

Variable Definition	
Ladder diagram (LD)	
Function Block Diagram(FBD)	

2.2.6.7 REAL/LREAL_TO (REAL/LREAL Type Conversion)

1. Function

Convert REAL/LREAL type to other types.

2. Parameter Description

Input	Description	Data Type
Input	Input Variable	REAL/LREAL
Output		
Output	Output Variable	BYTE, DATE, DT, DWORD, INT, SINT, TIME, TOD, WORD, DINT; Another REAL input may convert LREAL, LREAL input also may convert REAL.

3. Description

REAL/LREAL convert to other type algorithms (REAL_TO/LREAL_TO_xxx), the value is equal to integer value with disregarding the decimals, then it converts to a new variable type, xxx means other data type.

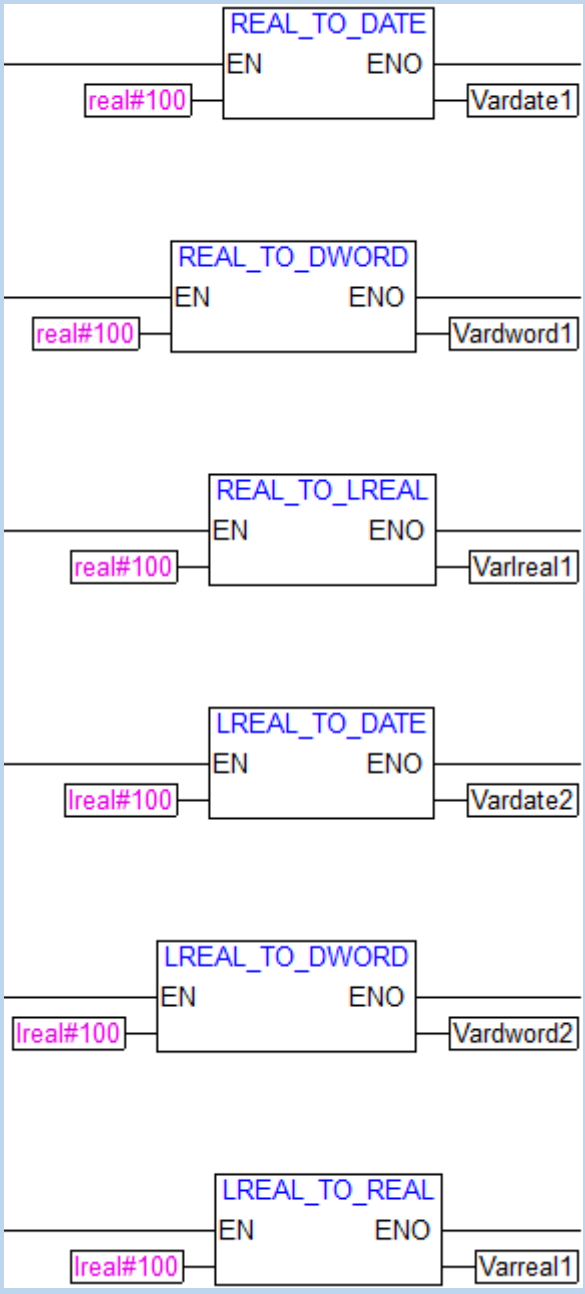
- When the data type from complex to simple data type, there may be loss of information.

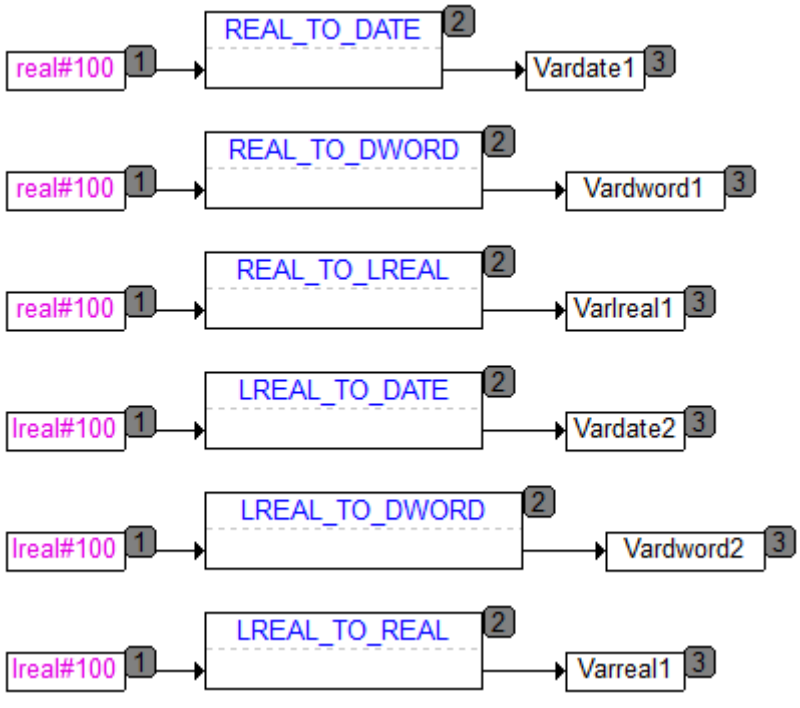
4. Application Example of Instructions

Variable Definition

No.	Name	Description	Data Type	Initial Value	Area
0001	Vardate1		DATE	D#1970-01-01	G Area
0002	Vardword1		DWORD	0	G Area
0003	Varreal1		REAL	0	G Area
0004	Varlreal1		LREAL	0	G Area
0005	Vardate2		DATE	D#1970-01-01	G Area
0006	Vardword2		DWORD	0	G Area

Programing language	Program

Variable Definition	
Ladder diagram (LD)	

Variable Definition	
Function Block Diagram(FBD)	

2.2.6.8 TIME_TO(TIME Type Conversion)

1. Function

Convert TIME type to other types.

2. Parameter Description

Input	Description	Data Type
Input	Input Variable	TIME
Output		
Output	Output Variable	BYTE, DINT, DWORD, INT, REAL, LREAL, SINT, WORD

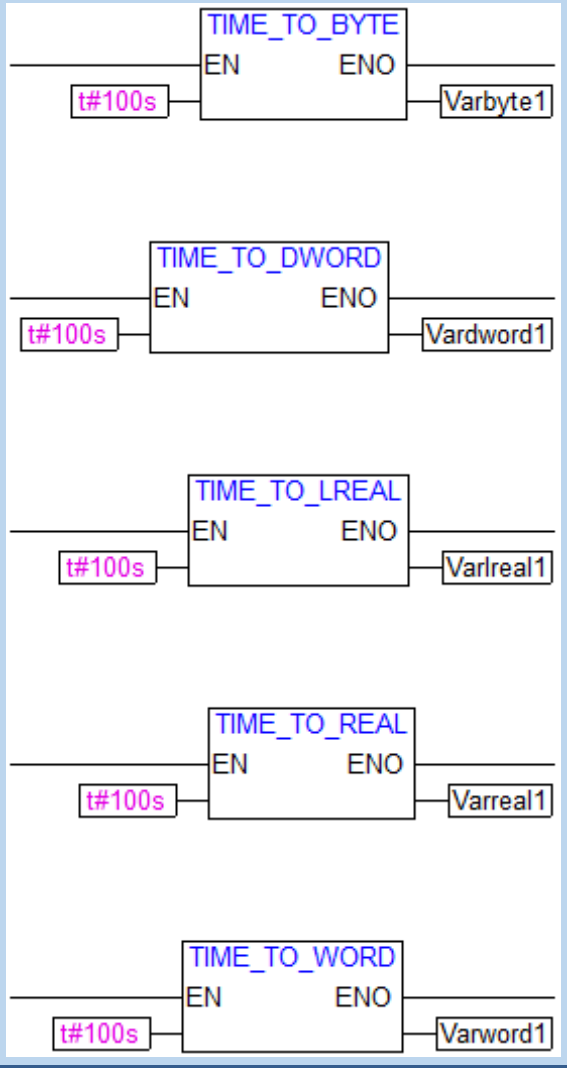
3. Description

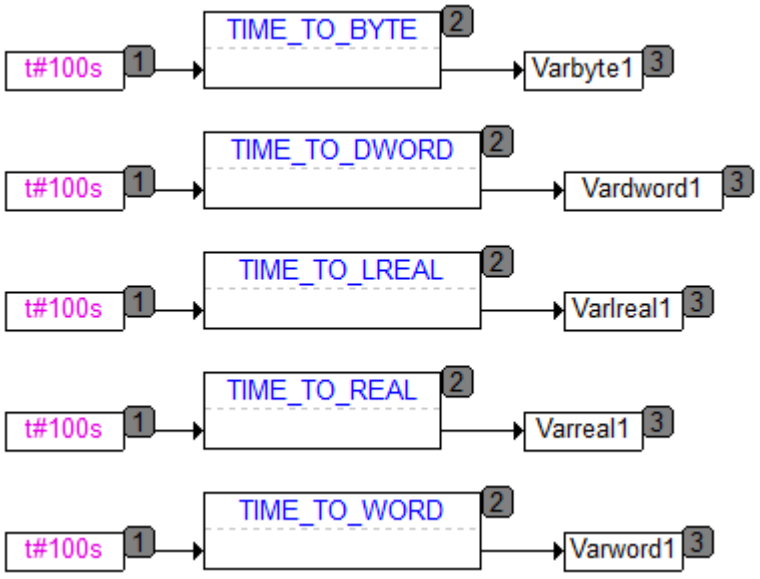
TIME convert to other type algorithms (TIME_TO_xxx), xxx means other data type, it completes the conversion of TIME type convert to other types.

4. Application Example of Instructions

Variable Definition

No.	Name	Description	Data Type	Initial Value	Area
0001	Varbyte1		BYTE	0	G Area
0002	Vardword1		DWORD	0	G Area
0003	Varlreal1		LREAL	0	G Area
0004	Varreal1		REAL	0	G Area
0005	Varword1		WORD	0	G Area

Programing language	Program
Ladder diagram (LD)	

Variable Definition	
Function Block Diagram(FBD)	

2.2.6.9 DATE_TO(DATE Type Conversion)

1. Function

Convert DATE type to other types.

2. Parameter Description

Input	Description	Data Type
Input	Input Variable	DATE
Output		
Output	Output Variable	BYTE, DINT, DWORD, INT, REAL, LREAL, SINT, WORD

3. Description

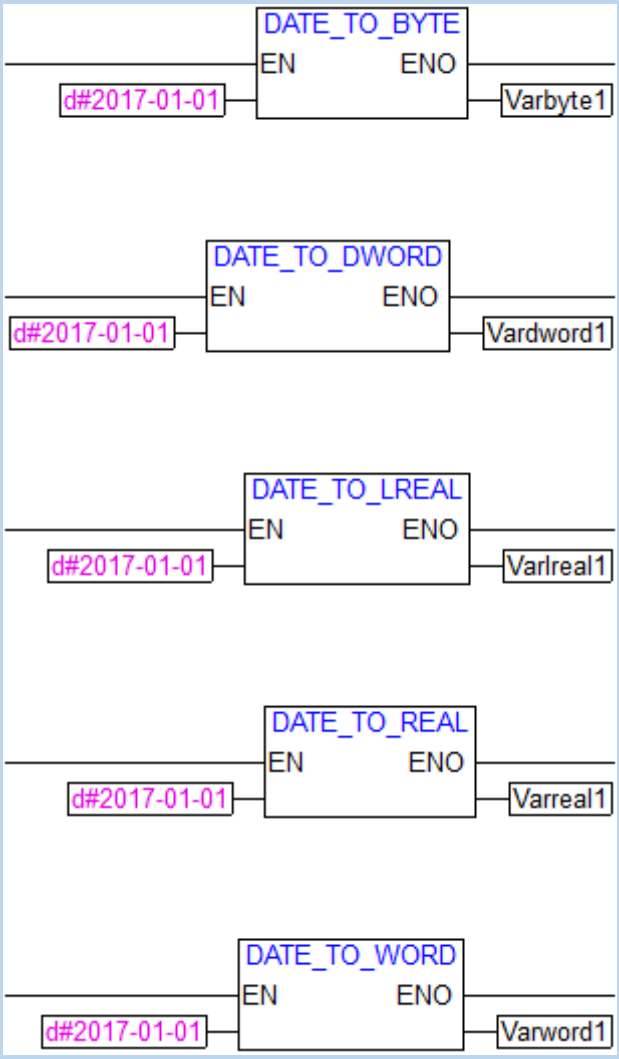
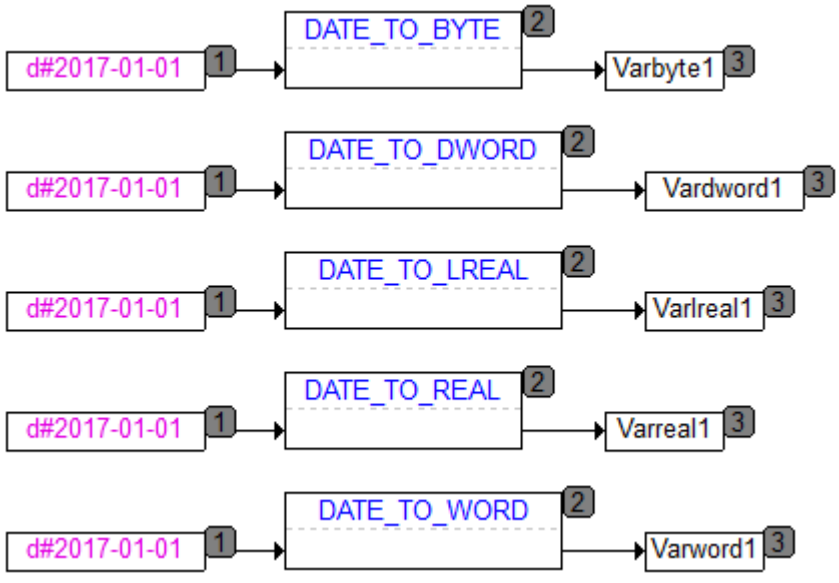
DATE convert to other types (DATE_TO_xxx), xxx means other data type, it completes the conversion of DATE type convert to other types.

4. Application Example of Instructions

Variable Definition

No.	Name	Description	Data Type	Initial Value	Area
0001	Varbyte1		BYTE	0	G Area
0002	Vardword1		DWORD	0	G Area
0003	Varlreal1		LREAL	0	G Area
0004	Varreal1		REAL	0	G Area
0005	Varword1		WORD	0	G Area

Programing language	Program

Variable Definition	
Ladder diagram (LD)	
Function Block Diagram(FBD)	

2.2.6.10 DT_TO(DT Type Conversion)

1. Function

Convert DT type to other types.

2. Parameter Description

Input	Description	Data Type
Input	Input Variable	DT
Output		
Output	Output Variable	BYTE, DINT, DWORD, INT, REAL, LREAL, SINT, WORD

3. Description

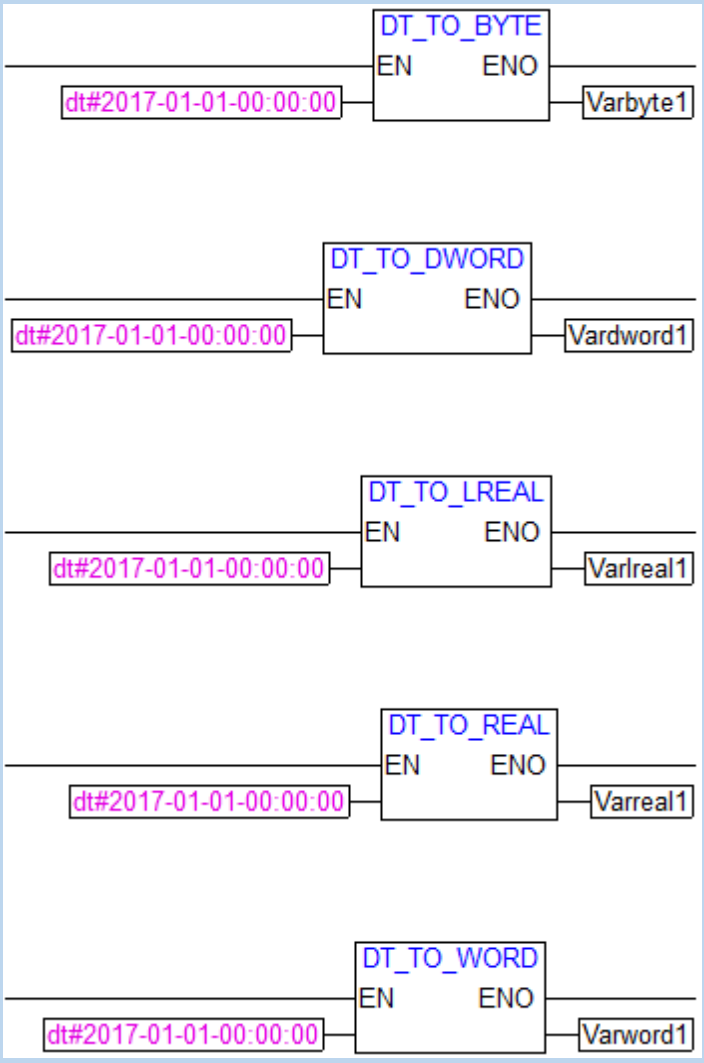
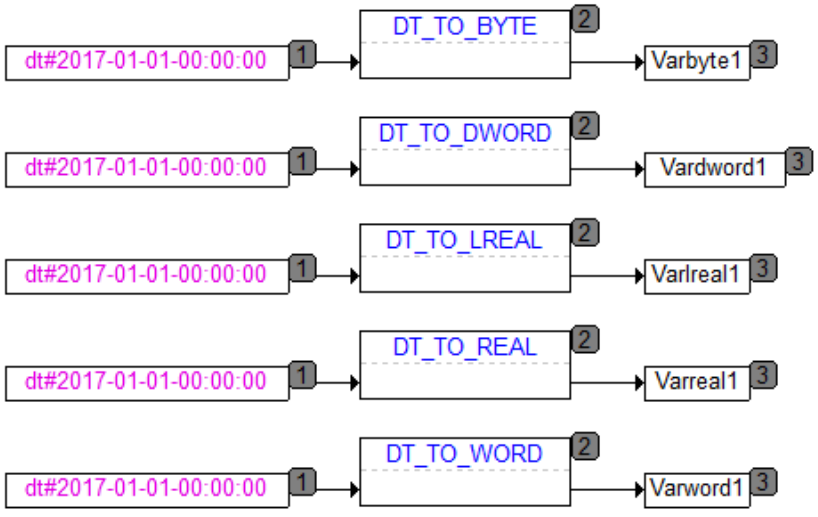
DT converts to other types (DT_TO_xxx), xxx means other data type, it completes the conversion of DT type convert to other types.

4. Application Example of Instructions

Variable Definition

No.	Name	Description	Data Type	Initial Value	Area
0001	Varbyte1		BYTE	0	G Area
0002	Vardword1		DWORD	0	G Area
0003	Varlreal1		LREAL	0	G Area
0004	Varreal1		REAL	0	G Area
0005	Varword1		WORD	0	G Area

Programing language	Program

Variable Definition	
Ladder diagram (LD)	
Function Block Diagram(FBD)	

2.2.6.11 TOD_TO(TOD Type Conversion)

1. Function

Convert TOD type to other types.

2. Parameter Description

Input	Description	Data Type
Input	Input Variable	TOD
Output		
Output	Output Variable	BYTE, DINT, DWORD, INT, REAL, LREAL, SINT, WORD

3. Description

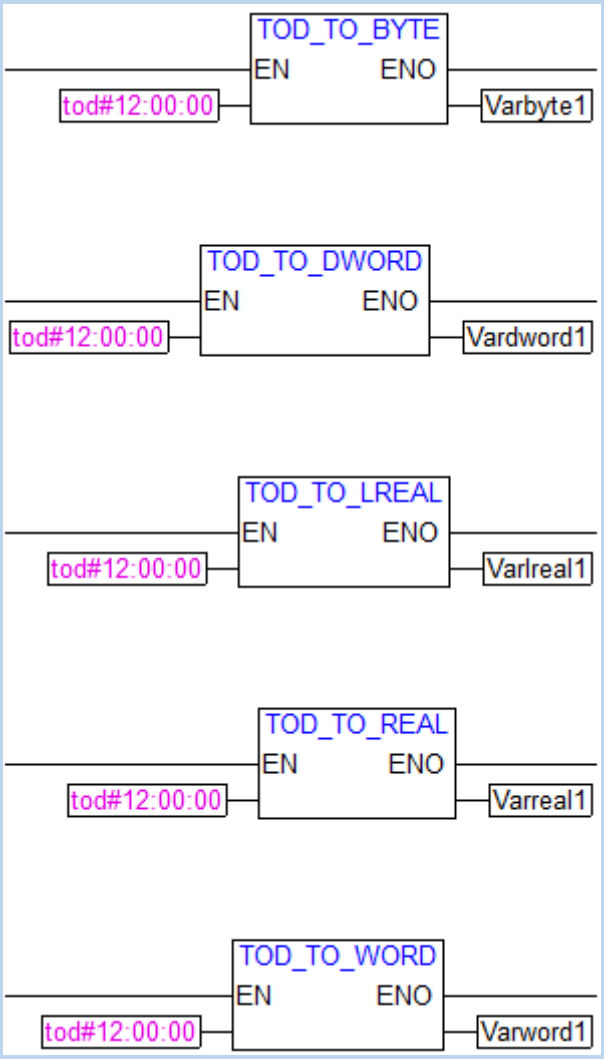
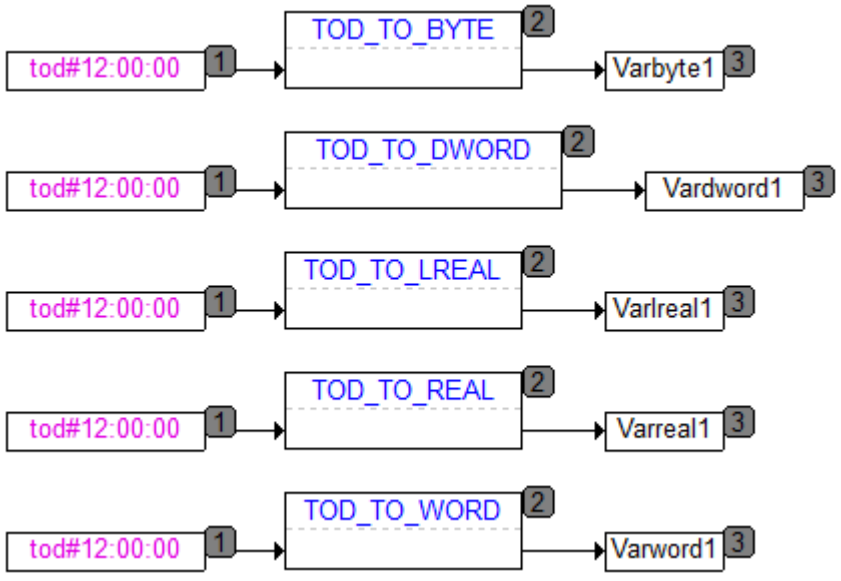
TOD convert to other type algorithms (TOD_TO_xxx), xxx means other data type, it completes the conversion of TOD type convert to other types.

4. Application Example of Instructions

Variable Definition

No.	Name	Description	Data Type	Initial Value	Area
0001	Varbyte1		BYTE	0	G Area
0002	Vardword1		DWORD	0	G Area
0003	Varlreal1		LREAL	0	G Area
0004	Varreal1		REAL	0	G Area
0005	Varword1		WORD	0	G Area

Programing language	Program

Variable Definition	
Ladder diagram (LD)	
Function Block Diagram(FBD)	

2.2.7 Edge Trigger

2.2.7.1 R_TRIG(Rising Edge Detection Trigger)

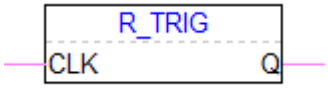


Figure 79 Rising Edge Delection Trigger

1. Parameter Description

Input	Data Type	Description	Initial Value
CLK	BOOL	Input	0
Output			
Q	BOOL	Output	0

2. Description

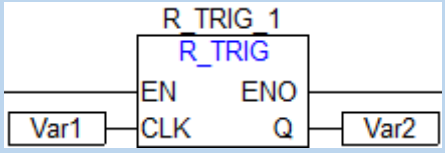
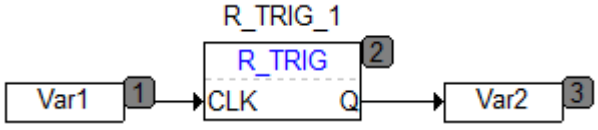
R_TRIG can realize the function of testing input rising edge (CLK). The input CLK and output Q are BOOL variables, the functions are described as follows in details.

When input variable (CLK) has rising edge from FALSE to TRUE, Q will output TRUE for one cycle, and output FALSE in other times.

3. Application Example of Instructions

Variable Definition

No.	Name	Description	Data Type	Initial Value	Area
0001	R_TRIG_1		R_TRIG		G Area
0002	Var1		BOOL	FALSE	G Area
0003	Var2		BOOL	FALSE	G Area

Programing language	Program
Ladder diagram (LD)	
Function Block Diagram(FBD)	

2.2.7.2 F_TRIG(Falling Edge Detection Trigger)

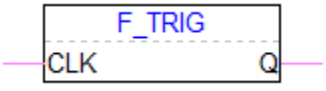


Figure 80 F_TRIG

1. Parameter Description

Input	Data Type	Description	Initial Value
CLK	BOOL	Input	0
Output			
Q	BOOL	Output	0

2. Description

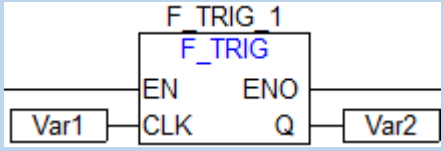
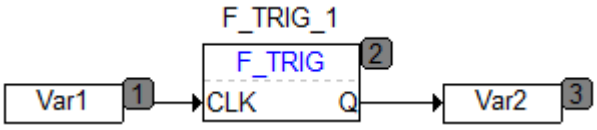
F_TRIG can realize the function of testing input falling edge (CLK). The input (CLK) and output (Q) are BOOL variables, the functions are described as follows in details.

When input variable (CLK) has falling edge from TRUE to FALSE, Q will output TRUE for one cycle, and output FALSE in other times.

3. Application Example of Instructions

Variable Definition

No.	Name	Description	Data Type	Initial Value	Area
0001	F_TRIG_1		F_TRIG		G Area
0002	Var1		BOOL	FALSE	G Area
0003	Var2		BOOL	FALSE	G Area

Programing language	Program
Ladder diagram (LD)	
Function Block Diagram(FBD)	

2.2.8 Bistable Trigger

2.2.8.1 SR(Set-Priority Bistable Trigger)

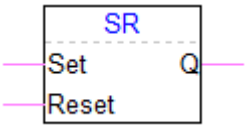


Figure 81 Set-Priority Bistable Trigger

1. Parameter Description

Input	Data Type	Description	Initial Value
Set	BOOL	Setting input	0
Reset	BOOL	Reset input	0
Output			

Input	Data Type	Description	Initial Value
Q	BOOL	Output	0

Truth table:

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

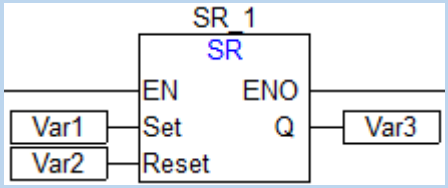
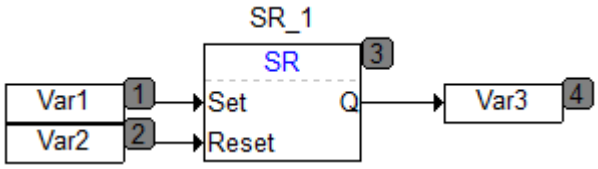
2. Description

The input (Set, Reset) and output (Q) are BOOL variable, it can realize the functions as follows: Q = (NOT Reset AND Q) OR Set.

3. Application Example of Instructions

Variable Definition

No.	Name	Description	Data Type	Initial Value	Area
0001	SR_1		SR		G Area
0002	Var1		BOOL	FALSE	G Area
0003	Var2		BOOL	FALSE	G Area
0004	Var3		BOOL	FALSE	G Area

Programing language	Program
Ladder diagram (LD)	
Function Block Diagram(FBD)	

2.2.8.2 RS(Reset-Priority Bistable Trigger)

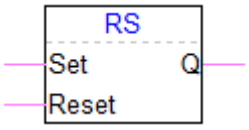


Figure 82 Reset-Priority Bistable Trigger

1. Parameter Description

Input	Data Type	Description	Initial Value
Set	BOOL	Setting input	0
Reset	BOOL	Reset input	0
Output			
Q	BOOL	Output	0

Truth table:

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

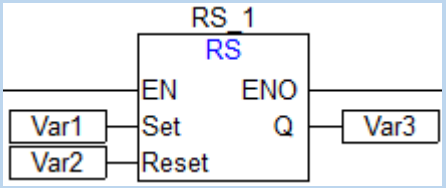
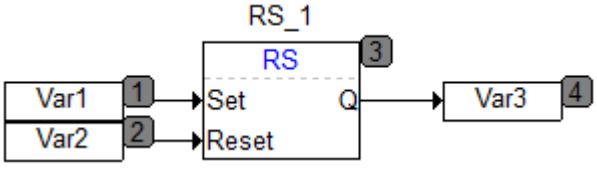
2. Description

The input (Set, Reset) and output (Q) of RS function block are BOOL variables. It can realize the functions as follows: Q= NOT Reset AND (Q OR Set).

3. Application Example of Instructions

Variable Definition

No.	Name	Description	Data Type	Initial Value	Area
0001	RS_1		RS		G Area
0002	Var1		BOOL	FALSE	G Area
0003	Var2		BOOL	FALSE	G Area
0004	Var3		BOOL	FALSE	G Area

Programing language	Program
Ladder diagram (LD)	
Function Block Diagram(FBD)	

2.2.9 Counter

2.2.9.1 CTD(Decrement Counter)

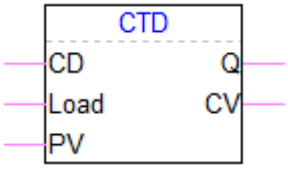


Figure 83 Decrement Counter

1. Parameter Description

Input	Data Type	Description	Initial Value
CD	BOOL	Rising edge testing pin	0
Load	BOOL	If Load is TRUE, CV=PV.	0
PV	WORD	Setting value	0
Output			
Q	BOOL	If CV≤0, Q output TRUE.	1
CV	WORD	Counter output	0

2. Description

CTD can realize the function of counter value decrease when input (CD) has rising edge. The input (CD, Load) and output (Q) are BOOL variables; input (PV) and output (CV) are WORD variable, which is described as follows in details.

- Initialized function

  If Load is TRUE, counter variable (CV) will be initialized to upper limit (PV).

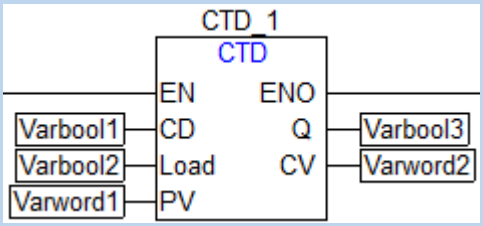
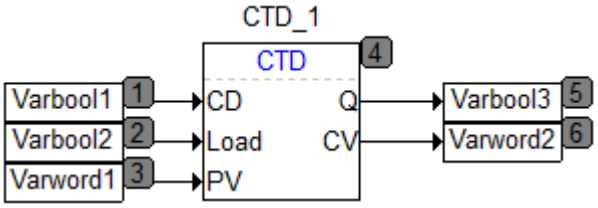
- Decrement counter function

  If CD has rising edge from FALSE to TRUE, counter (CV) will decrease 1. When CV⩽0, Q output 'TRUE'.

3. Application Example of Instructions

Variable Definition

No.	Name	Description	Data Type	Initial Value	Area
0001	CTD_1		CTD		G Area
0002	Varbool1		BOOL	FALSE	G Area
0003	Varbool2		BOOL	FALSE	G Area
0004	Varword1		WORD	0	G Area
0005	Varbool3		BOOL	FALSE	G Area
0006	Varword2		WORD	0	G Area

Variable Definition	
Programing language	Program
Ladder diagram (LD)	
Function Block Diagram(FBD)	

2.2.9.2 CTU(Increment Counter)

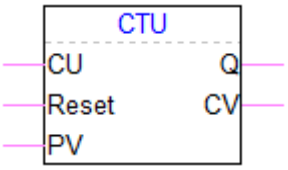


Figure 84 Increment Counter

1. Parameter Description

Input	Data Type	Description	Initial Value	Supplement
CU	BOOL	Rising edge testing pin	0	
Reset	BOOL	The variable CV is initialized for 0 when Reset is TRUE	0	
PV	WORD	Setting value	0	
Output				
Q	BOOL	When CV value is greater than or equal to PV, Q output TRUE.	0	
CV	WORD	Counter output	0	

2. Description

CTU can realize the function of counter increase when input (CU) has rising edge. The input (CU, Reset) and output (Q) are BOOL variables; input (PV) and output (CV) are WORD variables, which are described as follows in details.

- Initialized function

  If Reset is TRUE, the counter output CV=0.

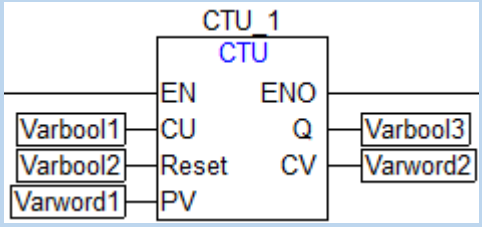
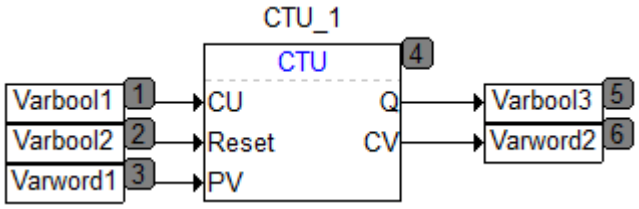
■ Counter function

If CU has rising edge from 'FALSE' to 'TRUE', the output of counter (CV) will increase 1 until CV ⩾PV, the output Q will return to TRUE.

3. Application Example of Instructions

Variable Definition

No.	Name	Description	Data Type	Initial Value	Area
0001	CTU_1		CTU		G Area
0002	Varbool1		BOOL	FALSE	G Area
0003	Varbool2		BOOL	FALSE	G Area
0004	Varword1		WORD	0	G Area
0005	Varbool3		BOOL	FALSE	G Area
0006	Varword2		WORD	0	G Area

Programing language	Program
Ladder diagram (LD)	
Function Block Diagram(FBD)	

2.2.9.3 CTUD(Increment/Decrement Counter)

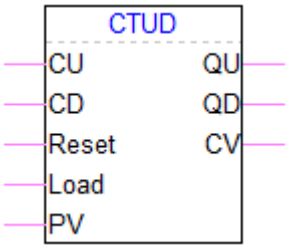


Figure 85 Increment/Decrement Counter

1. Parameter Description

Input	Data Type	Description	Initial Value
CU	BOOL	Rising edge testing pin	0

Input	Data Type	Description	Initial Value
CD	BOOL	Rising edge testing pin	0
Reset	BOOL	If Reset is TRUE, counter variable CV=0.	0
Load	BOOL	If Load is TRUE, counter variable CV=PV; if Reset is TRUE, CV=0.	0
PV	WORD	Setting value	0
Output			
QU	BOOL	If CV=PV, QU output TRUE.	1
QD	BOOL	If CV=0, QD output TRUE	1
CV	WORD	Counter output	0

2. Description

For CTUD the inputs such as CU, CD, Reset, Load and outputs QU, QD are BOOL variables; input PV and output CV are WORD variables, which are described as follows in details.

- Initialized function

  If Reset=1, the counter output CV=0; if Load=1, the counter output CV=PV.

- Counter function

  If CU has rising edge from 'FALSE' to 'TRUE', CV will increase 1. If CD has rising edge from 'FALSE' to 'TRUE', CV will decrease 1 (assume CV≥0).

  If CV≥PV, QU will return TRUE.

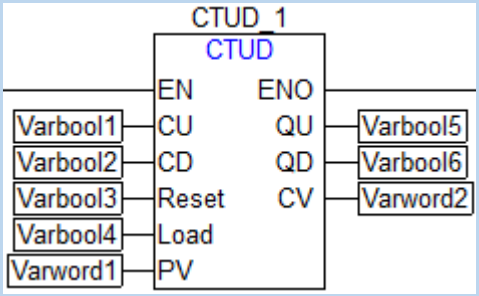
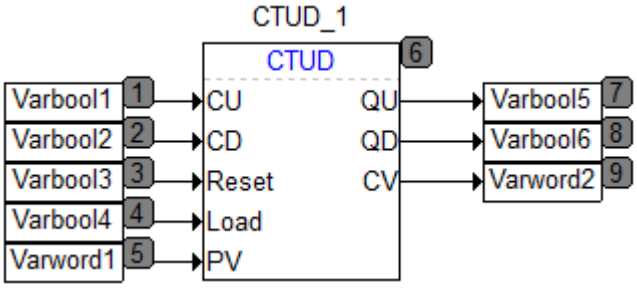
  If CV=0, QD will return TRUE.

3. Application Example of Instructions

Variable Definition

No.	Name	Description	Data Type	Initial Value	Area
0001	CTUD_1		CTUD		G Area
0002	Varbool1		BOOL	FALSE	G Area
0003	Varbool2		BOOL	FALSE	G Area
0004	Varbool3		BOOL	FALSE	G Area
0005	Varbool4		BOOL	FALSE	G Area
0006	Varword1		WORD	0	G Area
0007	Varbool5		BOOL	FALSE	G Area
0008	Varbool6		BOOL	FALSE	G Area
0009	Varword2		WORD	0	G Area

Programing language	Program

Variable Definition	
Ladder diagram (LD)	
Function Block Diagram(FBD)	

2.2.10 Timer

2.2.10.1 TON(On Delay Timer)

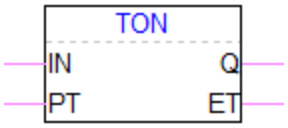


Figure 86 On Delay Timer

1. Parameter Description

Input	Data Type	Description	Initial Value
IN	BOOL	Input	0
PT	TIME	Delay time	T#0S
Output			
Q	BOOL	Output	0
ET	TIME	Timing	T#0S

- When this function block is used in the LD program, attention to EN pin :

  When the EN pin is changed from TRUE to FALSE, its internal timer continues to time. When EN is changed to TRUE again, the time accumulative value of EN during FALSE is effective, so the output may appear step change.

2. Description

TON can realize timer on delay command. The input (IN) and outputs (Q) are BOOL variables; the input (PT) and output (ET) are TIME variables, the functions are described as follows in details.

It will output the timer on delay command according to delay time (PT). If the IN is FALSE, the output is FALSE. When there is falling edge for input (IN), ET start to counting. If IN does not shift in PT, then when ET≥PT, Q will be TRUE and ET=PT. If in PT cycle, the IN changes to FALSE again, then Q is FALSE and ET will clear to 0.

- Sequence diagram

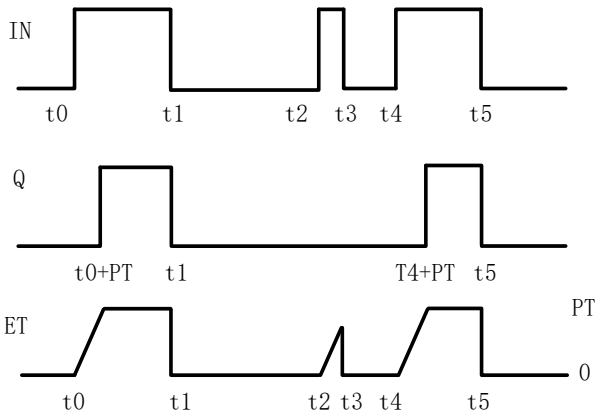
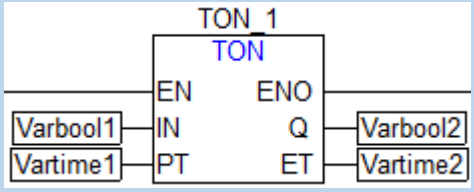
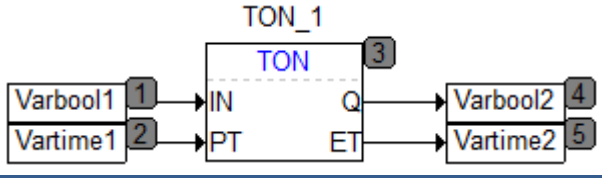


Figure 87 TON Sequence Diagram

3. Application Example of Instructions

Variable Definition

No.	Name	Description	Data Type	Initial Value	Area
0001	TON_1		TON		G Area
0002	Varbool1		BOOL	FALSE	G Area
0003	Vartime1		TIME	T#0MS	G Area
0004	Varbool2		BOOL	FALSE	G Area
0005	Vartime2		TIME	T#0MS	G Area

Programing language	Program
Ladder diagram (LD)	
Function Block Diagram(FBD)	

2.2.10.2 TONR(Retentive On Delay Timer)

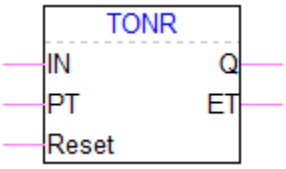


Figure 88 On Delay Timer

1. Parameter Description

Input	Data Type	Description	Initial Value
IN	BOOL	Input	0
PT	TIME	Delay time	T#0MS
Reset	BOOL	Reset end	FALSE
Output			
Q	BOOL	Output	0
ET	TIME	Timing	T#0MS

- When this function block is used in the LD program, attention to EN pin :

  When the EN pin is changed from TRUE to FALSE, its internal timer continues to time. When EN is changed to TRUE again, the time accumulative value of EN during FALSE is effective, so the output may appear step change.

2. Description

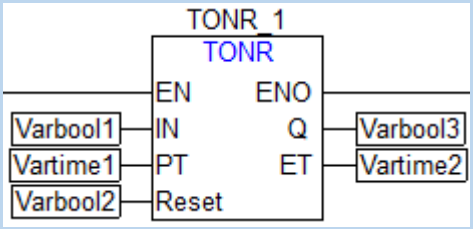
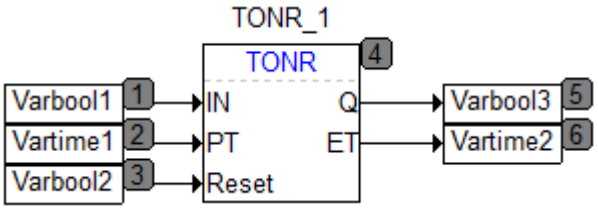
ET will start the timing in millisecond until ET is equal to PT when IN changes into TRUE. If the external trigger causes IN changes into FALSE before the delay time is up to the set time value. At this time, ET will hold the time value before the timing stops. ET will continue the timing when IN is 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.

3. Application Example of Instructions

Variable Definition

No.	Name	Description	Data Type	Initial Value	Area
0001	TONR_1		TONR		G Area
0002	Varbool1		BOOL	FALSE	G Area
0003	Vartime1		TIME	T#0MS	G Area
0004	Varbool2		BOOL	FALSE	G Area
0005	Varbool3		BOOL	FALSE	G Area
0006	Vartime2		TIME	T#0MS	G Area

Programing language	Program

Variable Definition	
Ladder diagram (LD)	
Function Block Diagram(FBD)	

2.2.10.3 TOF(Off Delay Timer)

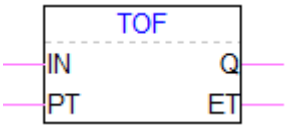


Figure 89 Off Delay Timer

1. Parameter Description

Input	Data Type	Description	Initial Value
IN	BOOL	Input	0
PT	TIME	Delay time	T#0S
Output			
Q	BOOL	Output	0
ET	TIME	Timing	T#0S

- When this function block is used in the LD program, attention to EN pin :

  When the EN pin is changed from TRUE to FALSE, its internal timer continues to time. When EN is changed to TRUE again, the time accumulative value of EN during FALSE is effective, so the output may appear step change.

2. Description

TOF can output turn-off command delay. The input (IN) and outputs (Q) are BOOL variables; the input (PT) and output (ET) are TIME variables, which are described as follows in details.

It will output the turn-off command delay according to delay time (PT). If the IN is TRUE, the output is TRUE. When there is falling edge for input (IN), ET start to counting. If IN does not shift in PT, then when ET≥PT, Q will turn to FALSE, and ET=PT. If in PT cycle, the IN changes to TRUE again, then Q is TRUE and ET will clear to 0.

- Sequence diagram

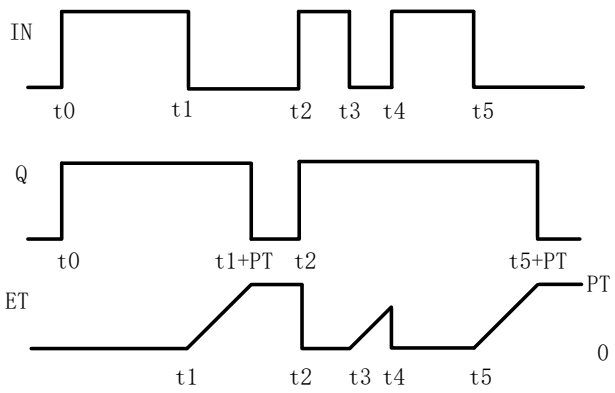
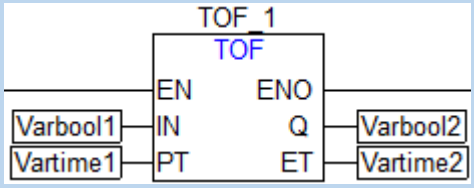
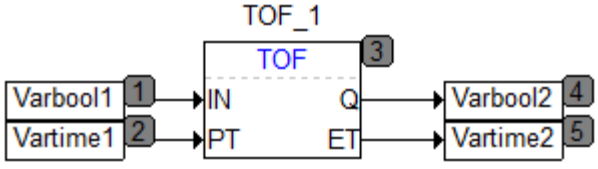


Figure 90 TOF Sequence Diagram

3. Application Example of Instructions

Variable Definition

No.	Name	Description	Data Type	Initial Value	Area
0001	TOF_1		TOF		G Area
0002	Varbool1		BOOL	FALSE	G Area
0003	Vartime1		TIME	T#0MS	G Area
0004	Varbool2		BOOL	FALSE	G Area
0005	Vartime2		TIME	T#0MS	G Area

Programing language	Program
Ladder diagram (LD)	
Function Block Diagram(FBD)	

2.2.10.4 TP(Normal Timer)

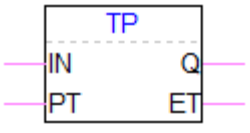


Figure 91 Normal Timer

1. Parameter Description

Input	Data Type	Description	Initial Value
IN	BOOL	Input	0
PT	TIME	Delay time	T#0S
Output			
Q	BOOL	Output	0
ET	TIME	Timing	T#0S

- When this function block is used in the LD program, attention to EN pin :

  When the EN pin is changed from TRUE to FALSE, its internal timer continues to time. When EN is changed to TRUE again, the time accumulative value of EN during FALSE is effective, so the output may appear step change.

2. Description

TP can realize timer function. The input (IN) and outputs (Q) are BOOL variables; the input (PT) and output (ET) are TIME variables, the functions are described as follows in details.

If IN is FALSE, Q will be FALSE, ET equal to 0. If IN change to TRUE, Q will be TRUE and ET start to time. If in PT cycle, the IN is invalid, then when ET>=PT, Q will be FALSE; after timing when IN is FALSE, ET=0.

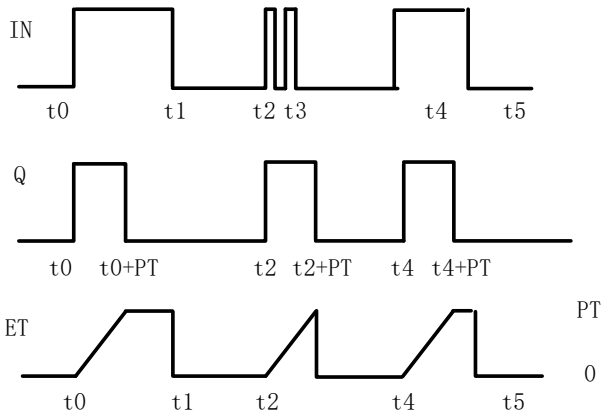


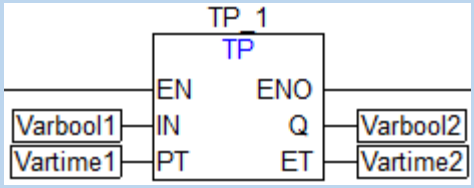
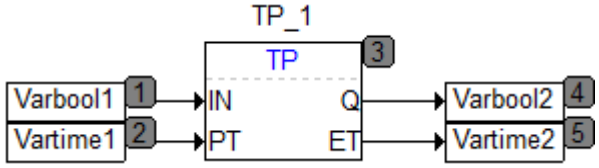
Figure 92 TP Sequence Diagram

3. Application Example of Instructions

Variable Definition

No.	Name	Description	Data Type	Initial Value	Area
0001	TP_1		TP		G Area
0002	Varbool1		BOOL	FALSE	G Area
0003	Vartime1		TIME	T#0MS	G Area
0004	Varbool2		BOOL	FALSE	G Area
0005	Vartime2		TIME	T#0MS	G Area

Programing language	Program

Variable Definition	
Ladder diagram (LD)	
Function Block Diagram(FBD)	

2.2.10.5 RTC(Set Time Counter)

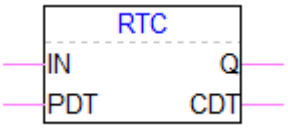


Figure 93 RTC

1. Parameter Description

Input	Data Type	Description	Initial Value
IN	BOOL	Enable	0
PDT	DT	Setting time	DT#1970-01-01-00:00:00
Output			
Q	BOOL	Valid symbol	0
CDT	DT	When IN is FALSE, the output is 1970-01-01-00-00：00：00; when IN is TRUE, the output will be counted up in seconds based on PDT.	DT#1970-01-01-00:00:00

- When this function block is used in the LD program, attention to EN pin :

  When the EN pin is changed from TRUE to FALSE, its internal timer continues to time. When EN is changed to TRUE again, the time accumulative value of EN during FALSE is effective, so the output may appear step change.

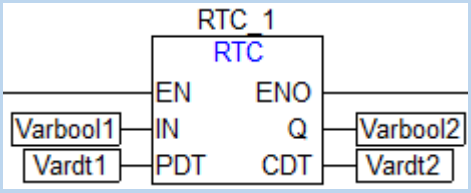
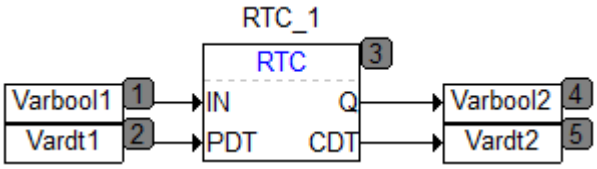
2. Description

RTC will return to real-time clock in preset time. If IN is TRUE, CDT is counted up in seconds based on PDT; If IN is FALSE, CDT is return to initialized value DT#1970-01-01-00:00:00.

3. Application Example of Instructions

Variable Definition

No.	Name	Description	Data Type	Initial Value	Area
0001	RTC_1		RTC		G Area
0002	Varbool1		BOOL	FALSE	G Area
0003	Vardt1		DT	DT#1970-01-01-00:00:00	G Area
0004	Varbool2		BOOL	FALSE	G Area
0005	Vardt2		DT	DT#1970-01-01-00:00:00	G Area

Programing language	Program
Ladder diagram (LD)	
Function Block Diagram(FBD)	

2.3 Basic Application Instruction

2.3.1 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.

2.3.1.1 BCD_TO_INT (BCD Code to Integer)

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.

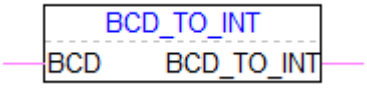


Figure 94 BCD_TO_INT Function Block

2. Parameter Description

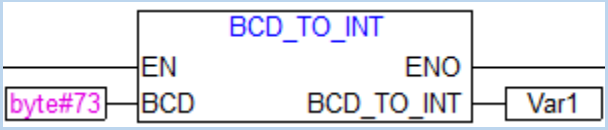
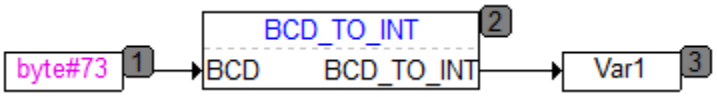
Input Parameter	Data Type	Parameter Description	Default Value
BCD	BYTE	BCD code to be converted.	0

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. Application Example of Instructions

Variable Definition

No.	Name	Description	Data Type	Initial Value	Area
0001	Var1		INT	0	G Area

Programing language	Program
Ladder diagram (LD)	
Function Block Diagram(FBD)	

2.3.1.2 INT_TO_BCD (Integer to BCD Code)

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.

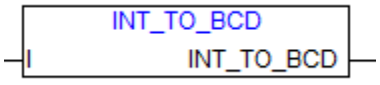


Figure 95 INT_TO_BCD Function Block

2. Parameter Description

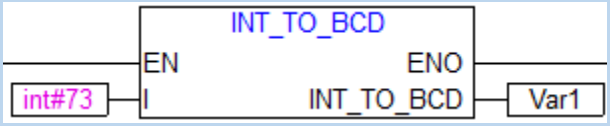
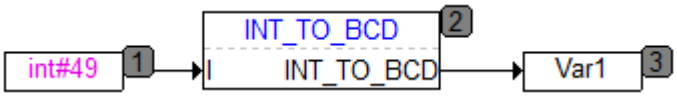
Input Parameter	Data Type	Parameter Description	Default Value
I	INT	Integer data value to be converted If the integer value is 49, the 49 is entered here	0

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. Application Example of Instructions

Variable Definition

No.	Name	Description	Data Type	Initial Value	Area
0001	Var1		BYTE	0	G Area

Programing language	Program
Ladder diagram (LD)	
Function Block Diagram(FBD)	

2.3.2 Data Split and Merge Functions

2.3.2.1 UNPACK (Bit Split)

1. Function

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

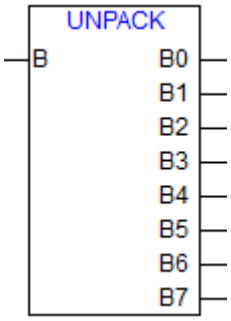


Figure 96 UNPACK Function Block

2. Parameter Description

Input Parameter	Description	Variable Type	Initial Value (Default)
B	BYTE type input variable	BYTE	0

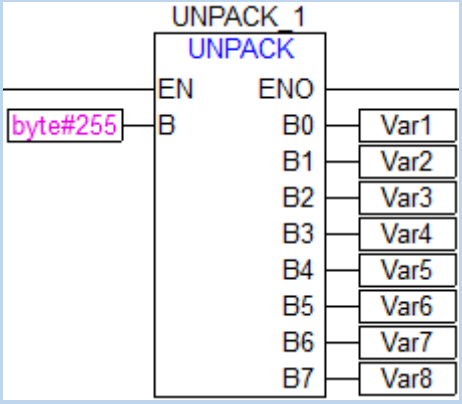
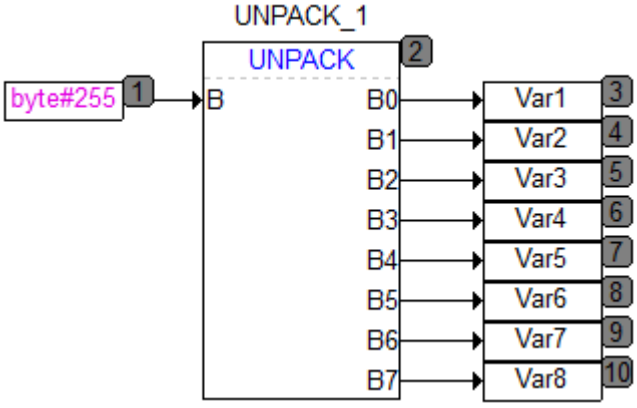
Output Parameter	Description	Variable Type	Initial Value (Default)
B0	BOOL type output variable	BOOL	FALSE
B1		BOOL	FALSE
B2		BOOL	FALSE
B3		BOOL	FALSE
B4		BOOL	FALSE
B5		BOOL	FALSE

Output Parameter	Description	Variable Type	Initial Value (Default)
B6		BOOL	FALSE
B7		BOOL	FALSE

3. Application Example of Instructions

Variable Definition

No.	Name	Description	Data Type	Initial Value	Area
0001	UNPACK_1		UNPACK		G Area
0002	Var1		BOOL	FALSE	G Area
0003	Var2		BOOL	FALSE	G Area
0004	Var3		BOOL	FALSE	G Area
0005	Var4		BOOL	FALSE	G Area
0006	Var5		BOOL	FALSE	G Area
0007	Var6		BOOL	FALSE	G Area
0008	Var7		BOOL	FALSE	G Area
0009	Var8		BOOL	FALSE	G Area

Programing language	Program
Ladder diagram (LD)	
Function Block Diagram(FBD)	

2.3.2.2 EXTRACT (Bit Extraction)

1. Function

Output the specified Nth bit of the input variable X.

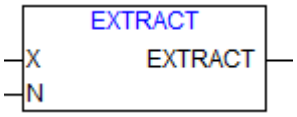


Figure 97 EXTRACT Function Block

2. Parameter Description

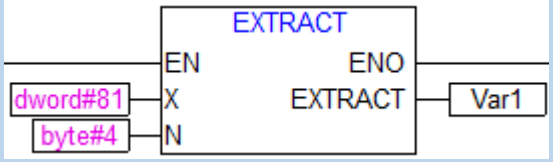
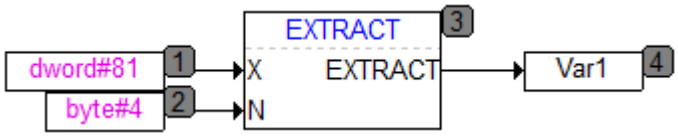
Input Parameter	Data Type	Function Description	Initial Value (Default)
X	DWORD	Manipulated variable	0
N	BYTE	Position	0

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

3. Application Example of Instructions

Variable Definition

No.	Name	Description	Data Type	Initial Value	Area
0001	Var1		BOOL	FALSE	G Area

Programing language	Program
Ladder diagram (LD)	
Function Block Diagram(FBD)	

2.3.2.3 PACK (Bit Merger)

1. Function

Combine with B0~B7 to form one BYTE variable from the 0th bit to the 7th bit. It is opposite to the UNPACK instruction.

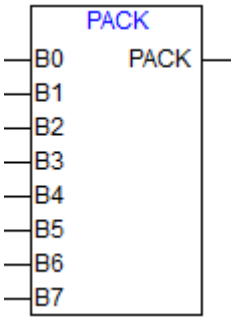


Figure 98 PACK Function Block

2. Parameter Description

Input Parameter	Description	Variable Type	Initial Value (Default)
B0	BOOL type input variable	BOOL	0
B1		BOOL	0
B2		BOOL	0
B3		BOOL	0
B4		BOOL	0
B5		BOOL	0
B6		BOOL	0
B7		BOOL	0

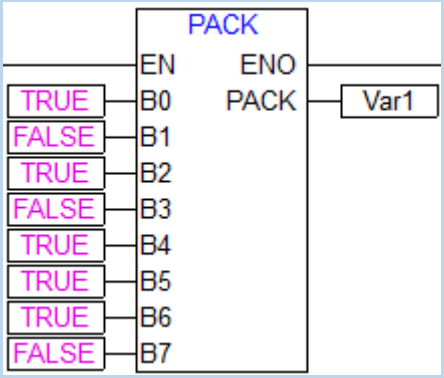
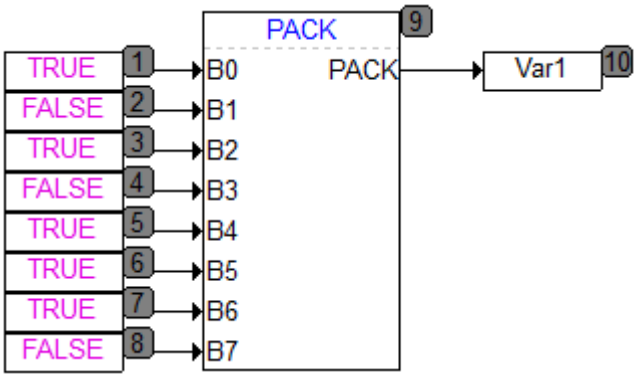
Output Parameter	Data Type	Function Description	Initial Value (Default)
Pack	BYTE	BYTE output	0

3. Application Example of Instructions

Variable Definition

No.	Name	Description	Data Type	Initial Value	Area
0001	Var1		BYTE	0	G Area

Programing language	Program

Variable Definition	
Ladder diagram (LD)	
Function Block Diagram(FBD)	

2.3.2.4 PUTBIT (Bit Assignment)

1. Function

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

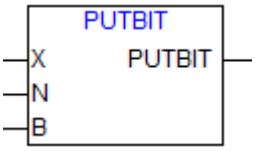


Figure 99 PUTBIT Function Block

2. Parameter Description

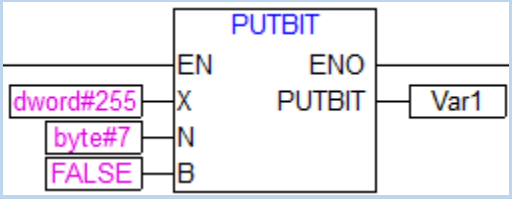
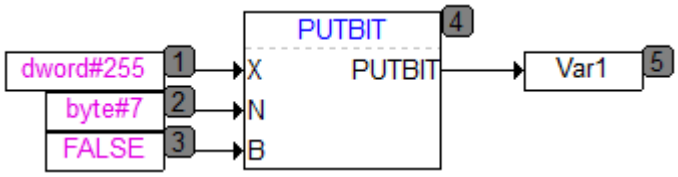
Input Parameter	Data Type	Description	Initial Value (Default)
X	DWORD	Manipulated variable	0
N	BYTE	Set	0
B	BOOL	Specified value	FALSE

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

3. Application Example of Instructions

Variable Definition

No.	Name	Description	Data Type	Initial Value	Area
0001	Var1		DWORD	0	G Area

Programing language	Program
Ladder diagram (LD)	
Function Block Diagram(FBD)	

2.3.2.5 DWORD_UNPACK_WORD (DWORD split WORD)

1. Function

Split a DWORD variable into two WORD variables according to the storage value in memory.

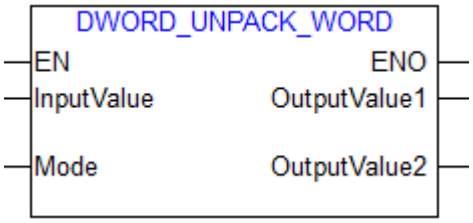


Figure 100 DWORD_UNPACK_WORD Function Block

2. Parameter Description

Input Parameter	Data Type	Parameter Description
InputValue	DWORD	Initial data to be split
Mode	BOOL	FALSE: high 16 bits and low 16 bits remain unchanged. TRUE: exchange the high 16 bits and low 16 bits.

Output Parameter	Data Type	Parameter Description
OutputValue1	WORD	When Mode is FALSE, the value corresponds to the high 16-bit byte of the Initial data. When Mode is TRUE, the value corresponds to the low 16-bit byte of the Initial data.
OutputValue2	WORD	When Mode is FALSE, the value corresponds to the low 16-bit byte of the Initial data. When Mode is TRUE, the value corresponds to the high 16-bit byte of the Initial data.

2.3.2.6 DINT_UNPACK_WORD (DINT split WORD)

1. Function

Split a DINT variable into two WORD variables according to the storage value in memory.

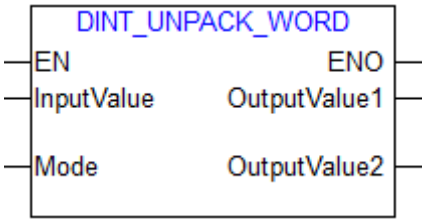


Figure 101 DINT_UNPACK_WORD Function Block

2. Parameter Description

Input Parameter	Data Type	Parameter Description
InputValue	DINT	Initial data to be split
Mode	BOOL	FALSE: high 16 bits and low 16 bits remain unchanged. TRUE: exchange the high 16 bits and low 16 bits.

Output Parameter	Data Type	Parameter Description
OutputValue1	WORD	When Mode is FALSE, the value corresponds to the high 16-bit byte of the Initial data. When Mode is TRUE, the value corresponds to the low 16-bit byte of the Initial data.
OutputValue2	WORD	When Mode is FALSE, the value corresponds to the low 16-bit byte of the Initial data. When Mode is TRUE, the value corresponds to the high 16-bit byte of the Initial data.

2.3.2.7 REAL_UNPACK_WORD (REAL split WORD)

1. Function

Split a REAL variable into two WORD variables according to the storage value in memory.

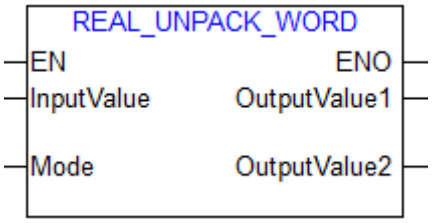


Figure 102 REAL_UNPACK_WORD Function Block

2. Parameter Description

Input Parameter	Data Type	Parameter Description
InputValue	REAL	Initial data to be split
Mode	BOOL	FALSE: high 16 bits and low 16 bits remain unchanged. TRUE: exchange the high 16 bits and low 16 bits.

Output Parameter	Data Type	Parameter Description
OutputValue1	WORD	When Mode is FALSE, the value corresponds to the high 16-bit byte of the Initial data. When Mode is TRUE, the value corresponds to the low 16-bit byte of the Initial data.
OutputValue2	WORD	When Mode is FALSE, the value corresponds to the low 16-bit byte of the Initial data. When Mode is TRUE, the value corresponds to the high 16-bit byte of the Initial data.

2.3.2.8 WORD_PACK_DWORD (WORD merge DWORD)

1. Function

Merge two WORD variables into a DWORD variable according to the storage value in memory.

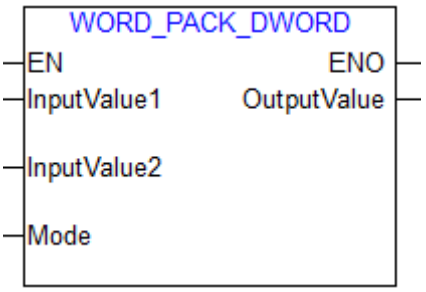


Figure 103 WORD_PACK_DWORD Function Block

2. Parameter Description

Input Parameter	Data Type	Parameter Description
InputValue1	WORD	Initial data to be merged
InputValue2	WORD	Initial data to be merged
Mode	BOOL	FALSE: InputValue1 is merged to high 16 bits and InputValue2 is merged to low 16 bits. TRUE: InputValue1 is merged to low 16 bits and InputValue2 is merged to high 16 bits.

Output Parameter	Data Type	Parameter Description
OutputValue	DWORD	Merged variable according to the Mode.

2.3.2.9 WORD_PACK_ DINT (WORD merge DINT)

1. Function

Merge two WORD variables into a DINT variable according to the storage value in memory.

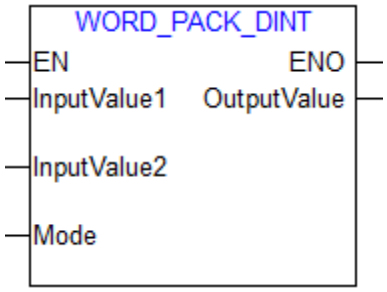


Figure 104 WORD_PACK_ DINT Function Block

2. Parameter Description

Input Parameter	Data Type	Parameter Description
InputValue1	WORD	Initial data to be merged
InputValue2	WORD	Initial data to be merged
Mode	BOOL	FALSE: InputValue1 is merged to high 16 bits and InputValue2 is merged to low 16 bits. TRUE: InputValue1 is merged to low 16 bits and InputValue2 is merged to high 16 bits.

Output Parameter	Data Type	Parameter Description
OutputValue	DINT	Merged variable according to the Mode.

2.3.2.10 WORD_PACK_ REAL (WORD merge REAL)

1. Function

Merge two WORD variables into a REAL variable according to the storage value in memory.

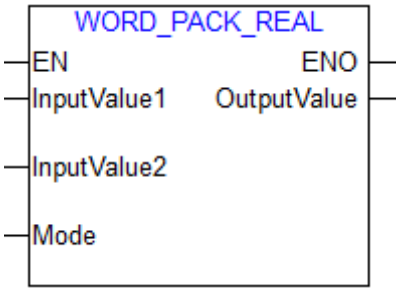


Figure 105 WORD_PACK_ REAL Function Block

2. Parameter Description

Input Parameter	Data Type	Parameter Description
InputValue1	WORD	Initial data to be merged
InputValue2	WORD	Initial data to be merged
Mode	BOOL	FALSE: InputValue1 is merged to high 16 bits and InputValue2 is merged to low 16 bits. TRUE: InputValue1 is merged to low 16 bits and InputValue2 is merged to high 16 bits.

Output Parameter	Data Type	Parameter Description
OutputValue	REAL	Merged variable according to the Mode.

2.3.3 Mathematical Functions

2.3.3.1 DERIVATIVE (Derivative Function)

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.

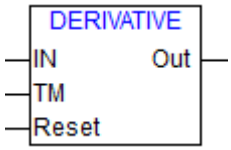


Figure 106 DERIVATIVE Function Block

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. 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

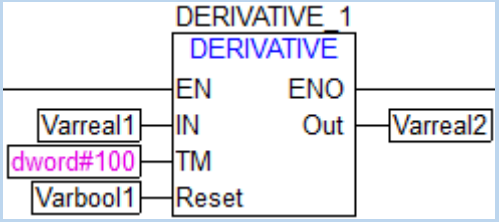
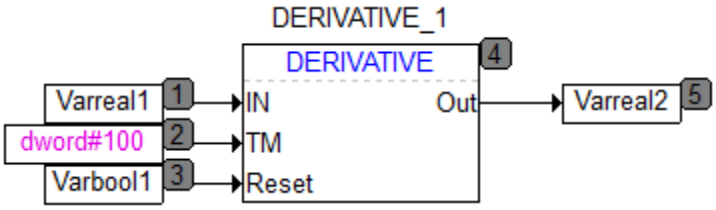
Output Parameter	Data Type	Function Description	Initial Value (Default)
OUT	REAL	Output of differential algorithm	0

4. Application Example of Instructions

Variable Definition

No.	Name	Description	Data Type	Initial Value	Area
0001	DERIVATIVE_1		DERIVATIVE		G Area
0002	Varreal1		REAL	0	G Area
0003	Varbool1		BOOL	FALSE	G Area
0004	Varreal2		REAL	0	G Area

Programing language	Program

Variable Definition	
Ladder diagram (LD)	
Function Block Diagram(FBD)	

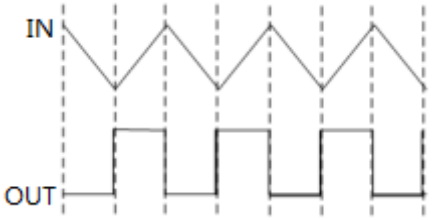


Figure 107 Input – Output Relation

2.3.3.2 INTEGRAL (Integral Function)

1. Function

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

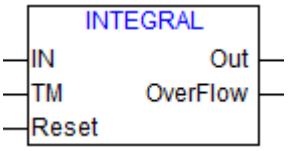


Figure 108 INTEGRAL Function Block

2. Integral iteration formula

OUT(k)=OUT(k-1)+TM*IN(k);

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

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

4. Application Example of Instructions

Variable Definition

No.	Name	Description	Data Type	Initial Value	Area
0001	INTEGRAL_1		INTEGRAL		G Area
0002	Varreal1		REAL	0	G Area
0003	Varbool1		BOOL	FALSE	G Area
0004	Varreal2		REAL	0	G Area
0005	Varbool2		BOOL	FALSE	G Area

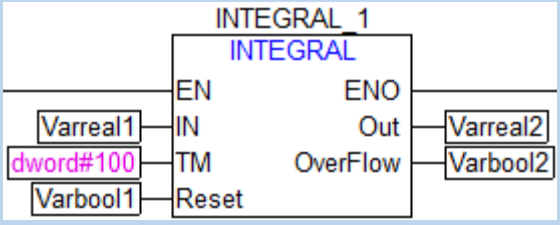
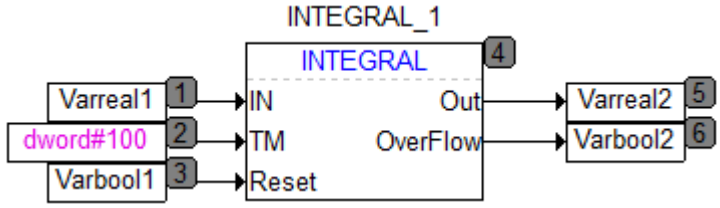
Programing language	Program
Ladder diagram (LD)	
Function Block Diagram(FBD)	

Figure 109 Input – Output Relation

2.3.3.3 STATISTICS_INT (Int Statistics)

1. Function

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

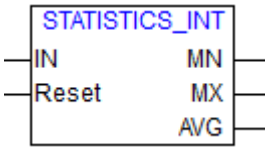


Figure 110 STATISTICS_INT Function Block

2. Parameter Description

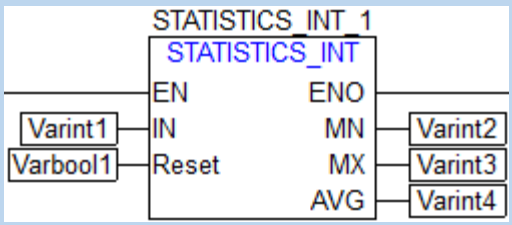
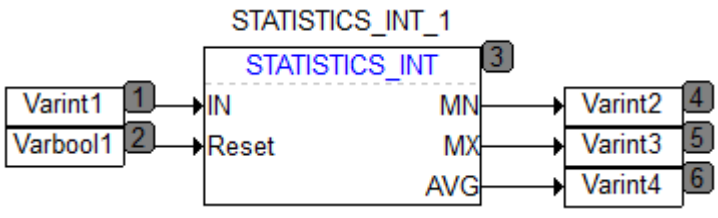
Input Parameter	Data Type	Function Description	Initial Value (Default)	Power Fail Safeguard
IN	INT	Manipulated variable	0	FALSE
Reset	BOOL	Initialization	FALSE	TRUE

Output Parameter	Data Type	Function Description	Initial Value (Default)	Power Fail Safeguard
MN	INT	Minimum output	32,767	TRUE
MX	INT	Maximum output	-32,768	TRUE
AVG	INT	Average output	0	FALSE

3. Application Example of Instructions

Variable Definition

No.	Name	Description	Data Type	Initial Value	Area
0001	STATISTICS_INT_1		STATISTICS_INT		G Area
0002	Varint1		INT	0	G Area
0003	Varbool1		BOOL	FALSE	G Area
0004	Varint2		INT	0	G Area
0005	Varint3		INT	0	G Area
0006	Varint4		INT	0	G Area

Programing language	Program
Ladder diagram (LD)	
Function Block Diagram(FBD)	

2.3.3.4 STATISTICS_REAL (Real Statistics)

1. Function

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

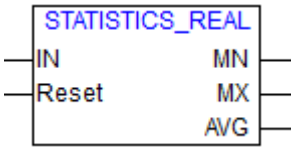


Figure 111 STATISTICS_ REAL Function Block

2. Parameter Description

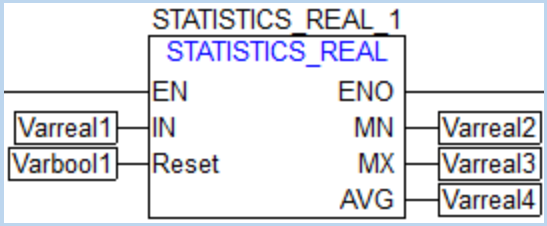
Input Parameter	Data Type	Function Description	Initial Value (Default)	Power Fail Safeguard
IN	REAL	Manipulated variable	0	FALSE
Reset	BOOL	Initialization	FALSE	TRUE

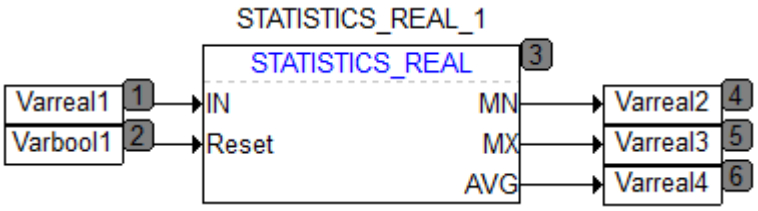
Output Parameter	Data Type	Function Description	Initial Value (Default)	Power Fail Safeguard
MN	REAL	Minimum output	3.4E+38	TRUE
MX	REAL	Maximum output	- 3.4E+38	TRUE
AVG	REAL	Average output	0	TRUE

3. Application Example of Instructions

Variable Definition

No.	Name	Description	Data Type	Initial Value	Area
0001	STATISTICS_REAL_1		STATISTICS_REAL		G Area
0002	Varreal1		REAL	0	G Area
0003	Varbool1		BOOL	FALSE	G Area
0004	Varreal2		REAL	0	G Area
0005	Varreal3		REAL	0	G Area
0006	Varreal4		REAL	0	G Area

Programing language	Program
Ladder diagram (LD)	

Variable Definition	
Function Block Diagram(FBD)	

2.3.3.5 VARIANCE (Calculation of Variance)

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.

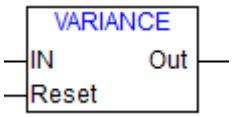


Figure 112 VARIANCE Function Block

2. 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

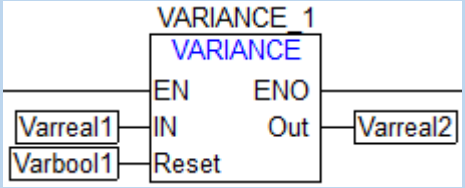
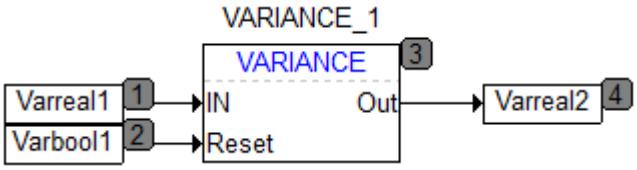
Output Parameter	Data Type	Function Description	Initial Value (Default)	Power Fail Safeguard
Out	REAL	Variance output	0	FALSE

Intermediate Variable	Data Type	Initial Value (Default)	Power Fail Safeguard
Z	DWORD	0	TRUE
A	LREAL	0	TRUE
B	LREAL	0	TRUE

3. Application Example of Instructions

Variable Definition

No.	Name	Description	Data Type	Initial Value	Area
0001	VARIANCE_1		VARIANCE		G Area
0002	Varreal1		REAL	0	G Area
0003	Varbool1		BOOL	FALSE	G Area
0004	Varreal2		REAL	0	G Area

Variable Definition	
Programing language	Program
Ladder diagram (LD)	
Function Block Diagram(FBD)	

2.3.4 Controller Functions

2.3.4.1 PID (The PID Controller)

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.

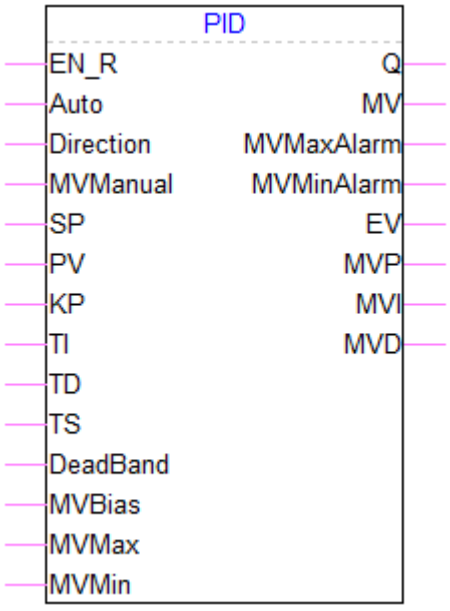


Figure 113 PID Function Block

2. Parameter Description

Input Parameter	Data Type	Function Description	Parameter Description	Initial Value (Default)	Power Fail Safeguard
EN_R	BOOL	Enable	FALSE: Invalid TRUE: Rising edge setting, high level valid	FALSE	TRUE
Auto	BOOL	“Automatic” mode selection	FALSE: Manual TRUE: Automatic	FALSE	TRUE
Direction	BOOL	“Positive action” mode selection	FALSE: Negative action (EV=SP-PV) TRUE: Positive action (EV=PV-SP)	FALSE	TRUE
MVManual	REAL	Manual input value		0	TRUE
SP	REAL	Set value		0	TRUE
PV	REAL	Process value		0	FALSE
KP	REAL	Proportional gain		1.0	TRUE
TI	REAL	Integral time (S)	Ti>=0, it indicates there is not the integral part if the integral time is zero.	0	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	0.1	TRUE
DeadBand	REAL	Deviation dead zone	(1) DeadBand>=0 (2) It indicates there is not any dead zone if DeadBand=0 (3) 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		100.0	TRUE
MVMin	REAL	Lower limit of output value		-100.0	TRUE

Output Parameter	Data Type	Function Description	Parameter Description	Initial Value (Default)	Power Fail Safeguard
Q	BOOL	Enable state flag	FALSE: Invalid	FALSE	FALSE

Output Parameter	Data Type	Function Description	Parameter Description	Initial Value (Default)	Power Fail Safeguard
			TRUE: Rising edge setting, high level valid		
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	FALSE
MVMinAlarm	BOOL	Alarm for output out of lower limit	FALSE: No over lower limit TRUE: Output has exceeded the lower limit	FALSE	FALSE
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		0	FALSE
MVI	REAL	Integral component		0	FALSE
MVD	REAL	Differential component		0	FALSE

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.

- Judgment of Set Value

  - 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.

  - 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.

- Calculation

  - Manual Mode

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

  - 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 114.

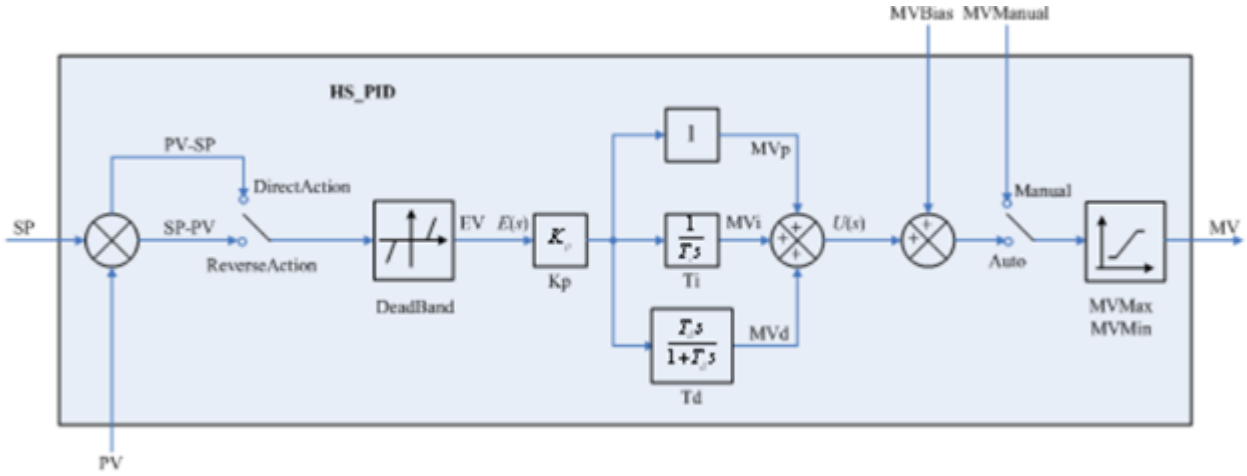


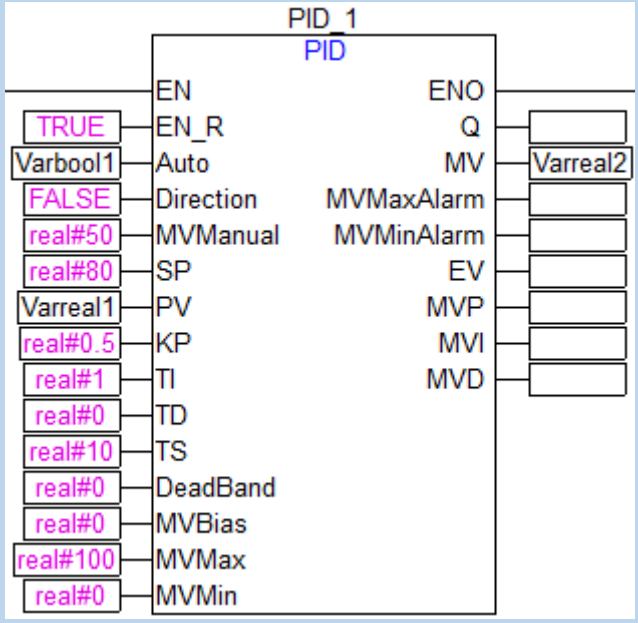
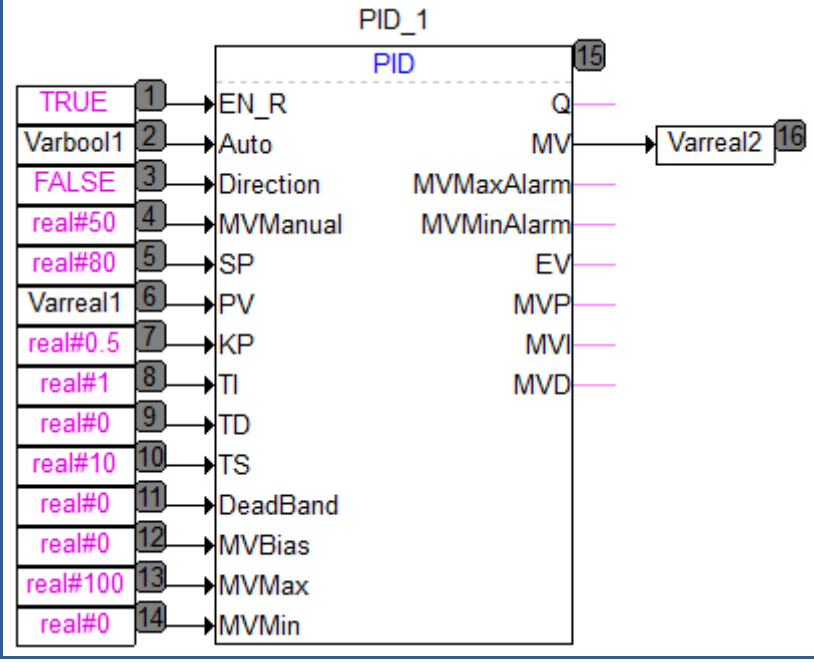
Figure 114 Logic Structure of PID Function Block

4. Application Example of Instructions

Variable Definition

No.	Name	Description	Data Type	Initial Value	Area
0001	PID_1		PID		G Area
0002	Varbool1		BOOL	FALSE	G Area
0003	Varreal1		REAL	0	G Area
0004	Varreal2		REAL	0	G Area

Programing language	Program

Variable Definition	
Ladder diagram (LD)	
Function Block Diagram(FBD)	

2.3.5 Signal Generator Functions

2.3.5.1 BLINK (Pulse Signal Generator)

1. Overview

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

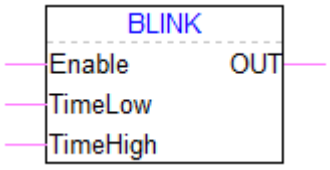


Figure 115 BLINK Function Block

2. 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	Variable Type	Initial Value (Default)	Power Fail Safeguard
OUT	Output	BOOL	FALSE	TRUE

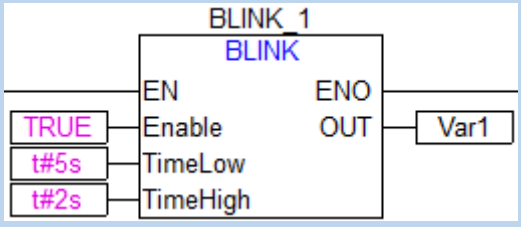
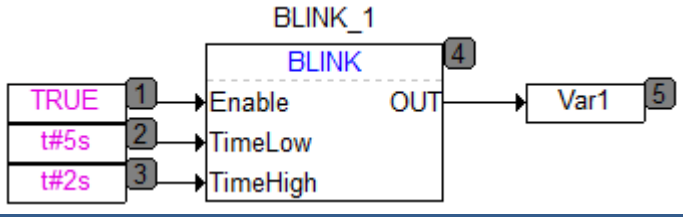
- When this function block is used in the LD program, attention to EN pin :

  When the EN pin is changed from TRUE to FALSE, its internal timer continues to time. When EN is changed to TRUE again, the time accumulative value of EN during FALSE is effective, so the output may appear step change.

3. Application Example of Instructions

Variable Definition

No.	Name	Description	Data Type	Initial Value	Area
0001	BLINK_1		BLINK		G Area
0002	Var1		BOOL	FALSE	G Area

Programing language	Program
Ladder diagram (LD)	
Function Block Diagram(FBD)	

When the instruction is executed, the waveform output of OUT is as shown in Figure 116.

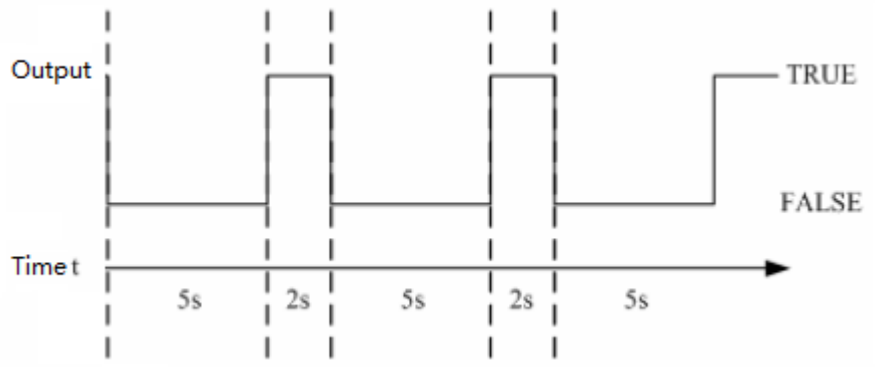


Figure 116 Output Waveform

2.3.5.2 GEN (Typical Signal Generator)

1. Overview

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

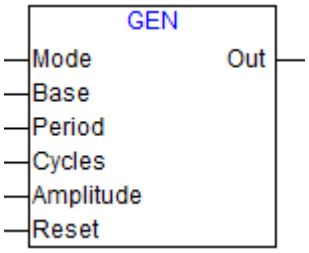


Figure 117 GEN Function Block

2. Parameter Description

Input Parameter	Variable Type	Function Description	Parameters Description	Initial Value (Default)	Power Fail Safeguard
Mode	BYTE	Defines the type of signal to be generated	Directly input value in the MODE, then produce the corresponding waveform	0	TRUE
		0: TRIANGLE	Triangle wave		
		1: TRIANGLE_POS	Triangle wave with zero point		
		2: SAWTOOTH_RISE	Rising sawtooth wave		
		3: SAWTOOTH_FALL	Descending sawtooth wave		
		4: RECTANGLE	Square wave		
		5: SINUS	Sine wave		
		6: COSINU	Cosine wave		
Base	BOOL	Cycle mode selection	When BASE is TRUE, the signal generator is related to the defined cycle period.	FALSE	TRUE

Input Parameter	Variable Type	Function Description	Parameters Description	Initial Value (Default)	Power Fail Safeguard
			When BASE is FALSE, the signal generator cycle is the number of current task cycles		
Period	TIME	Define the cycle time (valid when Base is true)		T#0MS	TRUE
Cycles	DWORD	Define the cycle time (Valid when Base is false)		0	TRUE
Amplitude	INT	The amplitude of the signal		0	TRUE
Reset	BOOL	Initialization	When RESET = TRUE, the signal generator is reset to zero	FALSE	TRUE

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

- When this function block is used in the LD program, attention to EN pin :

  When the EN pin is changed from TRUE to FALSE, its internal timer continues to time. When EN is changed to TRUE again, the time accumulative value of EN during FALSE is effective, so the output may appear step change.

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.

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 118 below (for there is not the vertical waveform in the actual signal, namely, one time point does not correspond to two values).

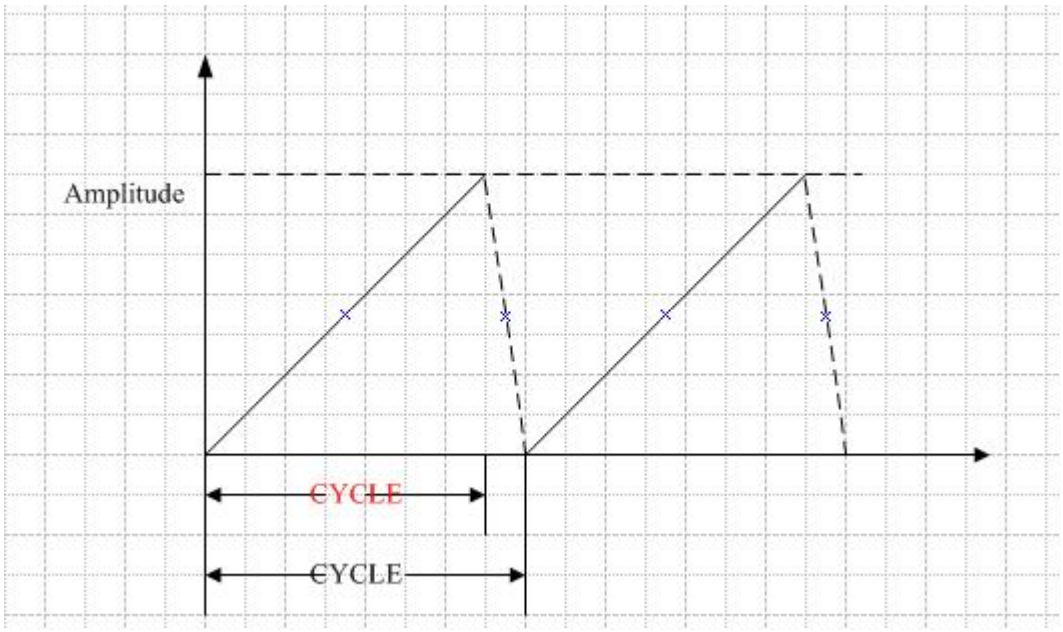


Figure 118 Cycle Chart

- 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 in Figure 119.

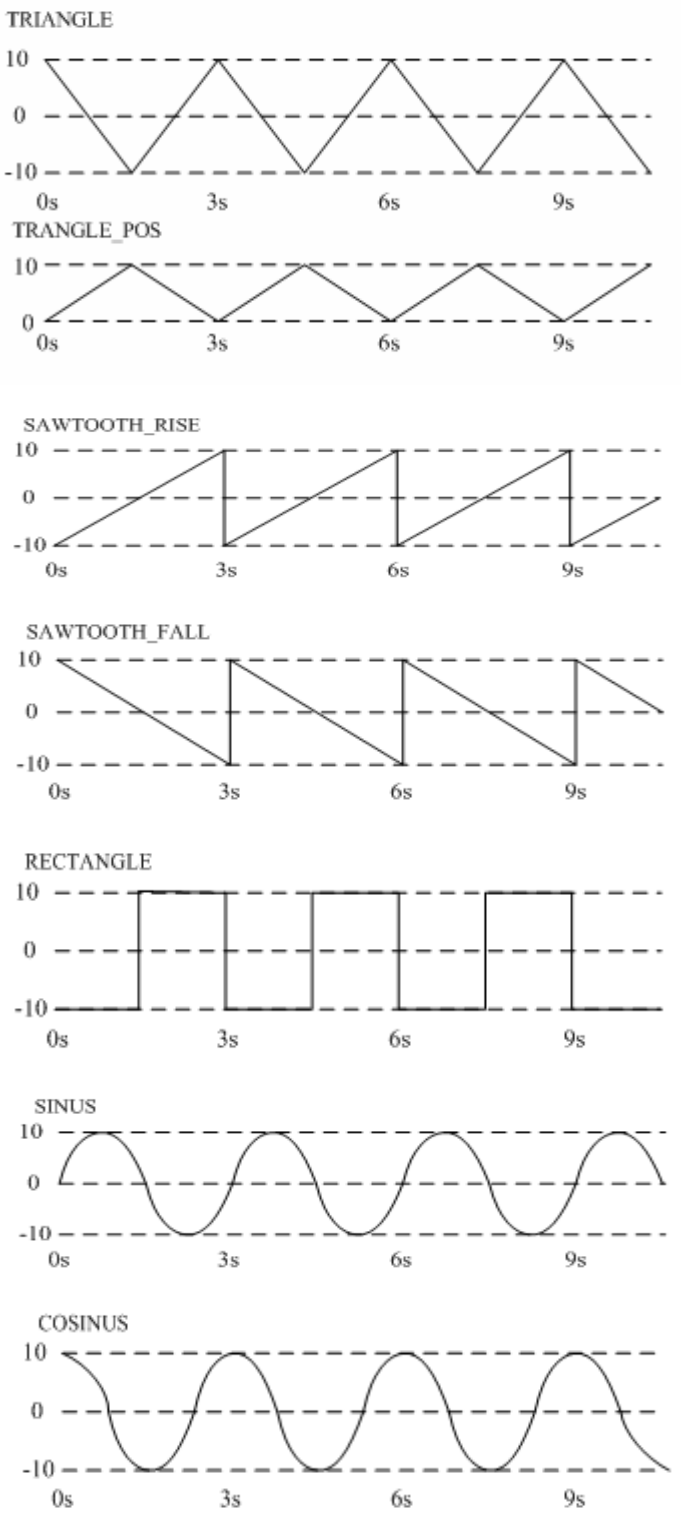
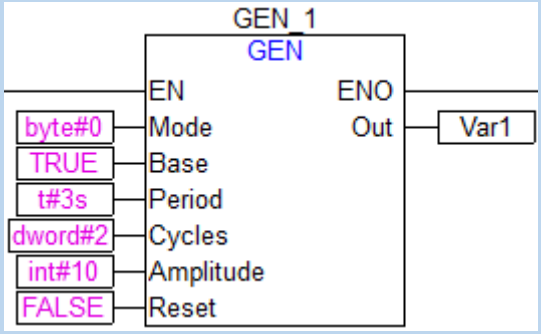
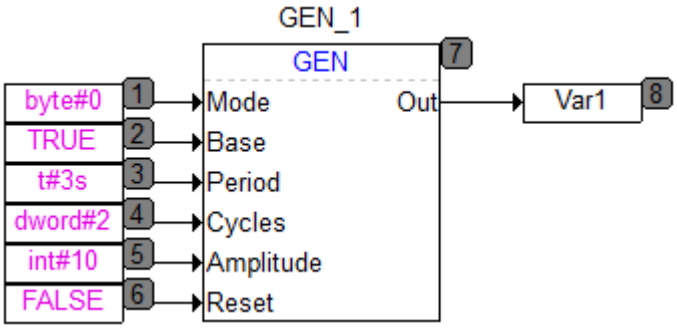


Figure 119 Waveforms

5. Application Example of Instructions

Variable Definition

No.	Name	Description	Data Type	Initial Value	Area
0001	GEN_1		GEN		G Area
0002	Var1		INT	0	G Area

Programing language	Program
Ladder diagram (LD)	
Function Block Diagram(FBD)	

2.3.5.3 RAND (Random Number Generator)

1. Function

Generate the random number.

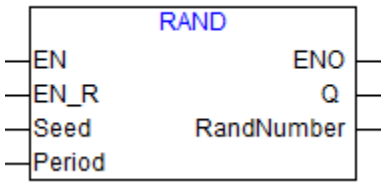


Figure 120 RAND Function Block

2. Parameter Description

Input Parameter	Data Type	Function Description	Parameter Description	Initial Value (Default)	Power Fail Safeguard
EN_R	BOOL	Enable	0: Invalid, stop generating random number, and RandNumber holds the last generated one.	FALSE	FALSE

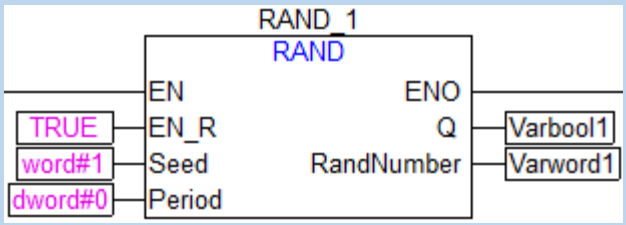
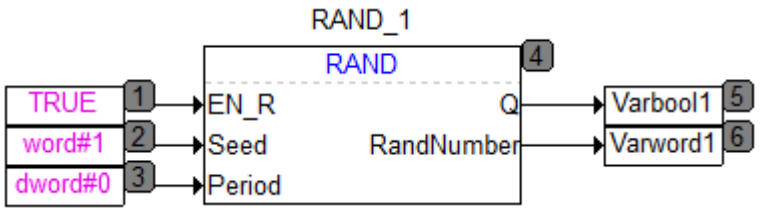
Input Parameter	Data Type	Function Description	Parameter Description	Initial Value (Default)	Power Fail Safeguard
			1: Rising edge enable, and high level valid		
Seed	WORD	Seed	The seed is the initial value to calculate the random sequence. Value range: 0-65,535	0	FALSE
Period	DWORD	Cycle to generate 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 200 ms. Value range: 0~4,294,967,295	0	FALSE

Output Parameter	Data Type	Function Description	Parameter Description	Initial Value (Default)	Power Fail Safeguard
Q	BOOL	Whether the data is valid?	0: Invalid data 1: Valid data	FALSE	FALSE
RandNumber	WORD	Random number	0-65,535	0	FALSE

3. Application Example of Instructions

Variable Definition

No.	Name	Description	Data Type	Initial Value	Area
0001	RAND_1		RAND		G Area
0002	Varbool1		BOOL	FALSE	G Area
0003	Varword1		WORD	0	G Area

Programing language	Program
Ladder diagram (LD)	
Function Block Diagram(FBD)	

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.

2.3.6 Analog Signal Processing Functions

2.3.6.1 RAMP_INT (Int Rate Limit)

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.

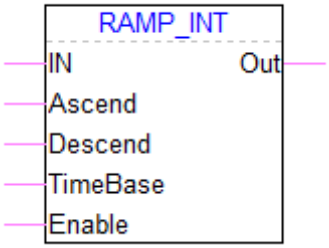


Figure 121 RAMP_INT Function Block

2. 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#1S	TRUE
Enable	BOOL	Rate limit enable	TRUE	FALSE

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

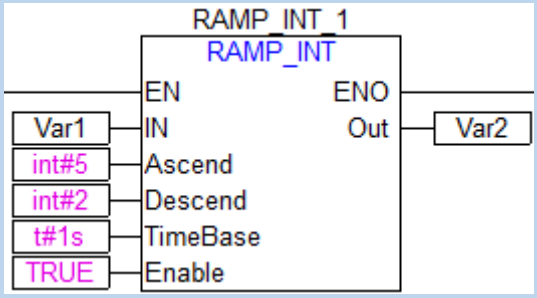
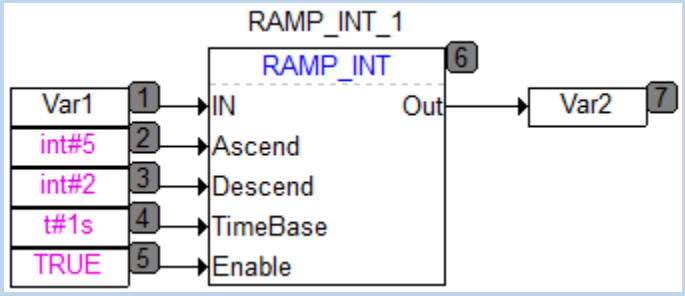
- When this function block is used in the LD program, attention to EN pin :

  When the EN pin is changed from TRUE to FALSE, its internal timer continues to time. When EN is changed to TRUE again, the time accumulative value of EN during FALSE is effective, so the output may appear step change.

3. Application Example of Instructions

Variable Definition

No.	Name	Description	Data Type	Initial Value	Area
0001	RAMP_INT_1		RAMP_INT		G Area
0002	Var1		INT	0	G Area
0003	Var2		INT	0	G Area

Programing language	Program
Ladder diagram (LD)	
Function Block Diagram(FBD)	

The corresponding output value in above example is shown in Figure 122 according to the change of the input value IN when the instruction is executed.

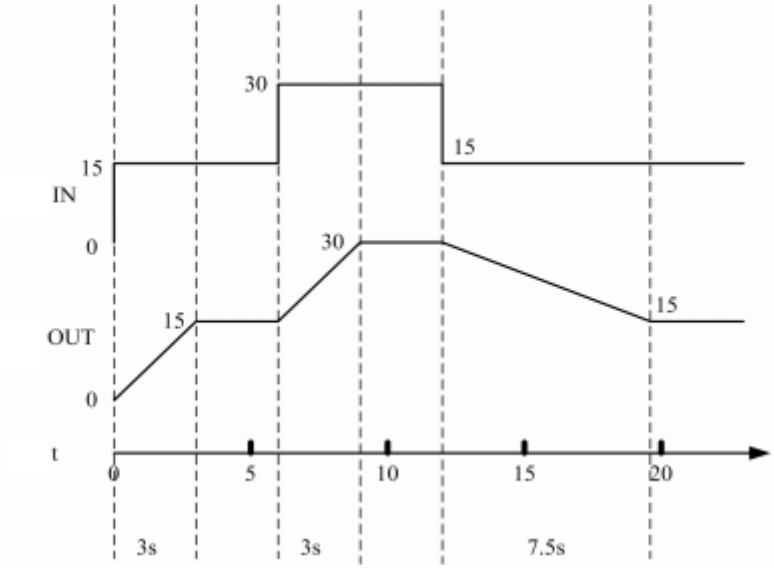


Figure 122 IN – OUT Relation

2.3.6.2 RAMP_REAL (Real Rate Limit)

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.

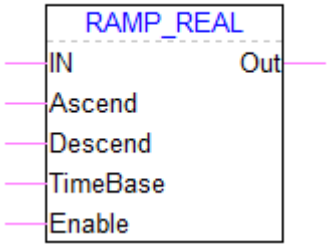


Figure 123 RAMP_REAL Function Block

2. 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#1S	TRUE
Enable	BOOL	Rate limit enable	TRUE	TRUE

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

- When this function block is used in the LD program, attention to EN pin :

  When the EN pin is changed from TRUE to FALSE, its internal timer continues to time. When EN is changed to TRUE again, the time accumulative value of EN during FALSE is effective, so the output may appear step change.

3. Application Example of Instructions

For the application example of instructions, refer to chapter 2.3.6.1 RAMP_INT (Int Rate Limit) Instruction.

2.3.6.3 CHARCURVE (Characteristic Curve)

1. Function

IN is the INT 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 INT type array PX[0..10] and PY[0..10]. The INT type OUT is the value after the operation. The BYTE type variable ERR 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<PX[0], ERR=2; IN>PX[N-1], ERR=3, 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.

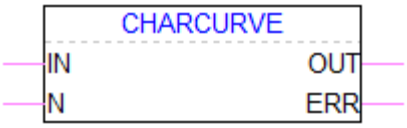


Figure 124 CHARCURVE Function Block

2. Parameter Description

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

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

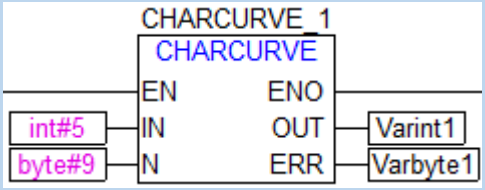
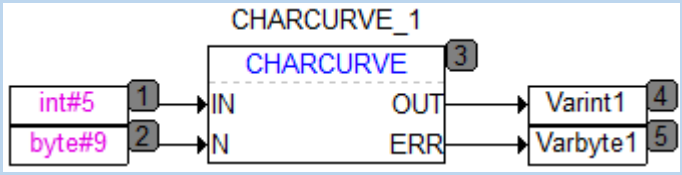
Other Variables	Data Type	Function Description	Initial Value (Default)	Power Fail Safeguard
PX	ARRAY[0..10] OF INT	X axis point range of characteristic curve	0	TRUE
PY	ARRAY[0..10] OF INT	Y axis point range of characteristic curve	0	TRUE

3. Application Example of Instructions

Variable Definition

No.	Name	Description	Data Type	Initial Value	Area
0001	CHARCURVE_1		CHARCURVE		G Area
0002	Varint1		INT	0	G Area
0003	Varbyte1		BYTE	0	G Area

Programing language	Program

Variable Definition	
Ladder diagram (LD)	
Function Block Diagram(FBD)	

2.3.6.4 HYSTERESIS (Lag)

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.

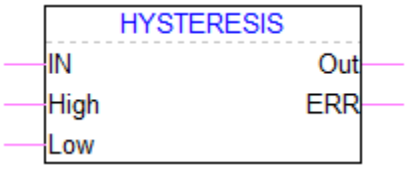


Figure 125 HYSTERESIS Function Block

2. Parameter Description

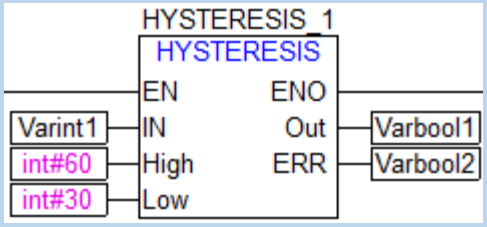
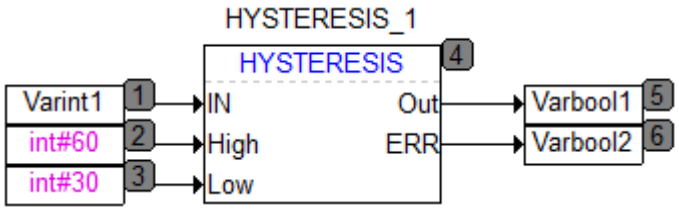
Input Parameter	Data Type	Function Description	Initial Value (Default)	Power Fail Safeguard
IN	INT	Input value	0	FALSE
Low	INT	Lower limit	0	TRUE
High	INT	Higher limit	0	TRUE

Output Parameter	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
ERR	BOOL	ERR will alarm when higher limit and lower limit of input are opposite each other	FALSE	FALSE

3. Application Example of Instructions

Variable Definition

No.	Name	Description	Data Type	Initial Value	Area
0001	HYSTERESIS_1		HYSTERESIS		G Area
0002	Varint1		INT	0	G Area
0003	Varbool1		BOOL	FALSE	G Area
0004	Varbool2		BOOL	FALSE	G Area

Programing language	Program
Ladder diagram (LD)	
Function Block Diagram(FBD)	

In the example above, the corresponding output value are shown in Figure 126 according to the change in the input value when the instruction is executed.

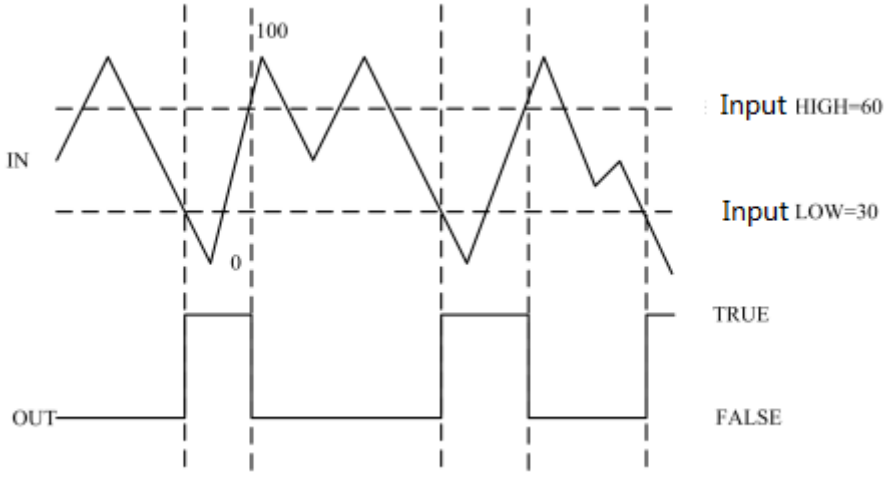


Figure 126 Change Relation of Input and Output

2.3.6.5 LIMITALARM (Limit Alarm)

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.

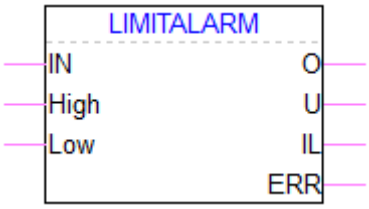


Figure 127 LIMITALARM Function Block

2. 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

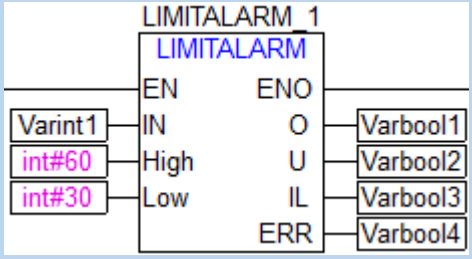
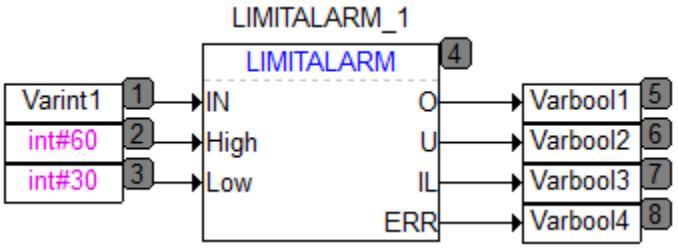
Output Parameter	Function Description	Data Type	Initial Value (Default)	Power Fail Safeguard
O	The input value is greater than the higher limit.	BOOL	FALSE	FALSE
U	The input value is less than the lower limit.	BOOL	FALSE	FALSE
IL	The input value is between the lower limit and the higher limit.	BOOL	FALSE	FALSE
ERR	ERR will alarm when higher limit and lower limit of input are opposite each other	BOOL	FALSE	FALSE

3. Application Example of Instructions

Variable Definition

No.	Name	Description	Data Type	Initial Value	Area
0001	LIMITALARM_1		LIMITALARM		G Area
0002	Varint1		INT	0	G Area
0003	Varbool1		BOOL	FALSE	G Area
0004	Varbool2		BOOL	FALSE	G Area
0005	Varbool3		BOOL	FALSE	G Area
0006	Varbool4		BOOL	FALSE	G Area

Programing language	Program

Variable Definition	
Ladder diagram (LD)	
Function Block Diagram(FBD)	

In the example above, the corresponding output value are shown in Figure 128 according to the change in the input value when the instruction is executed.

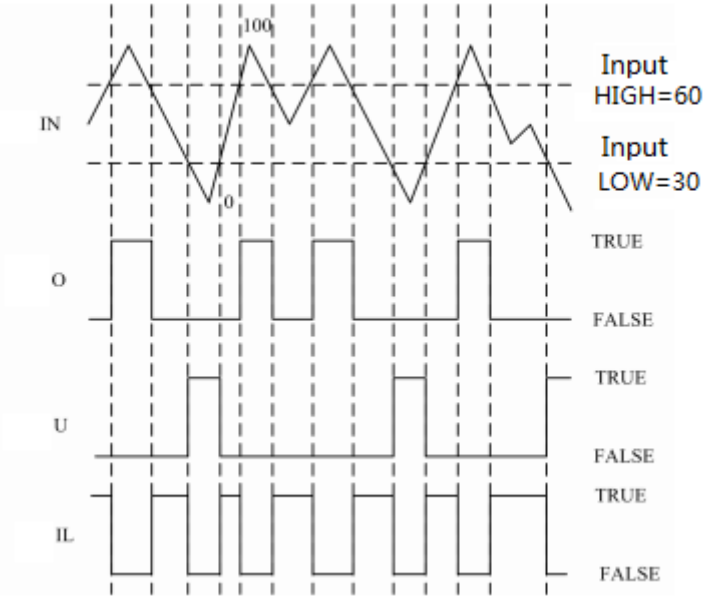


Figure 128 Change Relation of Input and Output

2.3.6.6 HEX_ENGIN (Hexadecimal Conversion to Engineering Data)

1. Function

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

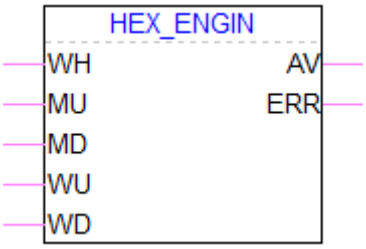


Figure 129 HEX_ENGIN Function Block

2. Parameter Description

Input Parameter	Description	Data Type	Initial Value (Default)	Power Fail Safeguard
WH	Input code value	WORD	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

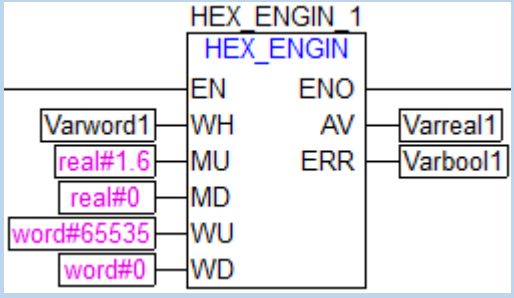
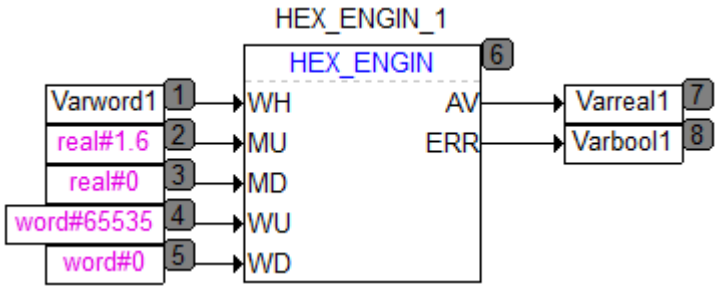
Output Parameter	Description	Data Type	Initial Value (Default)	Power Fail Safeguard
AV	Converted engineering quantity	REAL	0	FALSE
ERR	ERR will alarm when upper limit and lower limit of engineering quantity are opposite each other	BOOL	FALSE	FALSE

3. Application Example of Instructions

Variable Definition

No.	Name	Description	Data Type	Initial Value	Area
0001	HEX_ENGIN_1		HEX_ENGIN		G Area
0002	Varword1		WORD	0	G Area
0003	Varreal1		REAL	0	G Area
0004	Varbool1		BOOL	FALSE	G Area

Programing language	Program

Variable Definition	
Ladder diagram (LD)	
Function Block Diagram(FBD)	

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 Varword1, the upper limit of the engineering quantity is 1.6 Mpa, the lower limit of the engineering quantity is 0 Mpa, and the code value range of corresponding module is 0-65,535. We assume the corresponding output value is 0.799 when the value of Varword1 is 32,767.

2.3.6.7 ENGIN_HEX (Engineering Data Conversion to Hexadecimal)

1. Function

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

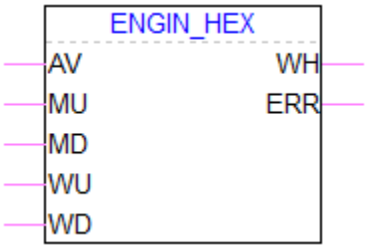


Figure 130 ENGIN_HEX Function Block

2. 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

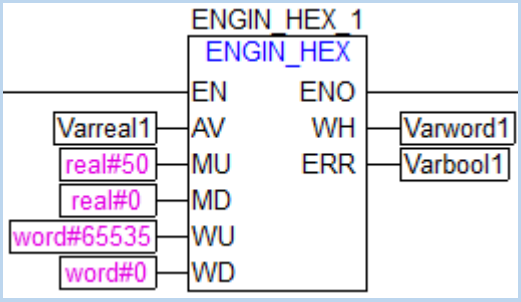
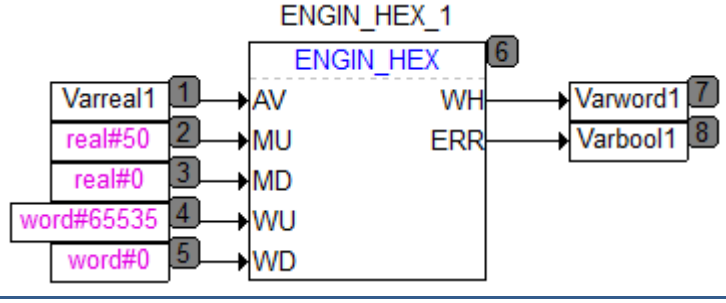
Input Parameter	Description	Data Type	Initial Value (Default)	Power Fail Safeguard
WU	Upper limit of analog input code value for PLC	WORD	0	TRUE
WD	Lower limit of analog input code value for PLC	WORD	0	TRUE

Output Parameter	Description	Data Type	Initial Value (Default)	Power Fail Safeguard
WH	Converted code value	WORD	0	FALSE
ERR	ERR will alarm when upper limit and lower limit of engineering quantity are opposite each other	BOOL	FALSE	FALSE

3. Example for Use of Instructions

Variable Definition

No.	Name	Description	Data Type	Initial Value	Area
0001	ENGIN_HEX_1		ENGIN_HEX		G Area
0002	Varreal1		REAL	0	G Area
0003	Varword1		WORD	0	G Area
0004	Varbool1		BOOL	FALSE	G Area

Programing language	Program
Ladder diagram (LD)	
Function Block Diagram(FBD)	

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 50 Hz, the lower limit of the engineering quantity is 0 Hz, and the value range

of corresponding module is 0-65,535. When the value of Varreal1 is 25 Hz, the corresponding Varword1 will output 32,768.

2.4 System Instruction

2.4.1 Time Correlation Instruction

2.4.1.1 GetTickCnt (Get Tick Count)

1. Function

Get the time of the controller since it boots (unit: μs, time frame: 0 - 2^{32}μs). if the time exceeds upper limit, it will start counting from 0.

2. Parameter Description

Return Value	Type	Meaning
GetTickCnt	DWORD	The time since the controller boots, which is counted from zero after it overflows (us)

2.4.1.2 TimeDelay (Time Delay)

1. Function

Delay IEC time.

2. Parameter Description

Input Parameter	Type	Parameter Description
uiTimeDelay	DWORD	Delay time (unit: ms), default value is 0.

Output Parameter	Type	Parameter Description
TimeDelay	DINT	0: The time delay is executed successfully. -1: The time delay is failed to be executed.

2.4.1.3 SET_RTC (Set Real Time Clock)

1. Function

Set start time of the PLC real time clock. Only set RTC of master controller and RTC of slave controller is synchronized from master controller.

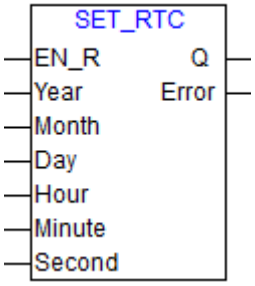


Figure 131 SET_RTC Function Block

2. Parameter Description

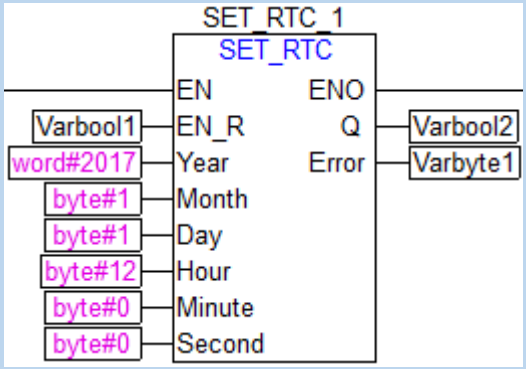
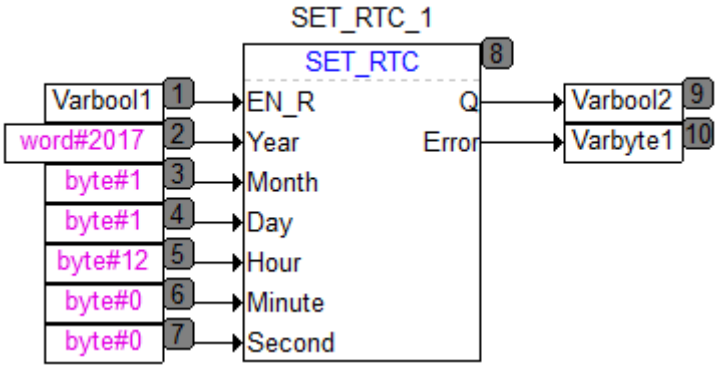
Input Parameter	Data Type	Function Description	Parameter Description	Default Value
EN_R	BOOL	Enable	0: Invalid 1: Rising edge valid	0
Year	WORD	Year	1971~2036	2011
Month	BYTE	Month	1~12	1
Day	BYTE	Day	1~31	1
Hour	BYTE	Hour	0~23	0
Minute	BYTE	Minute	0~59	0
Second	BYTE	Second	0~59	0

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. Application Example of Instructions

Variable Definition

No.	Name	Description	Data Type	Initial Value	Area
0001	SET_RTC_1		SET_RTC		G Area
0002	Varbool1		BOOL	FALSE	G Area
0003	Varbool2		BOOL	FALSE	G Area
0004	Varbyte1		BYTE	0	G Area

Programing language	Program
Ladder diagram (LD)	
Function Block Diagram(FBD)	

2.4.1.4 GET_RTC (Read Real Time Clock)

1. Function

This function block will read the time of the real time clock for the PLC.

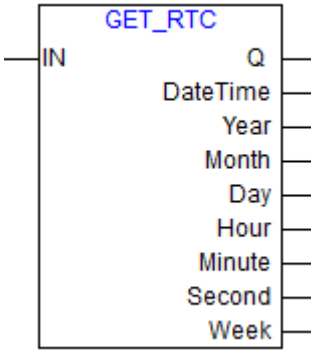


Figure 132 GET_RTC Function Block

2. Parameter Description

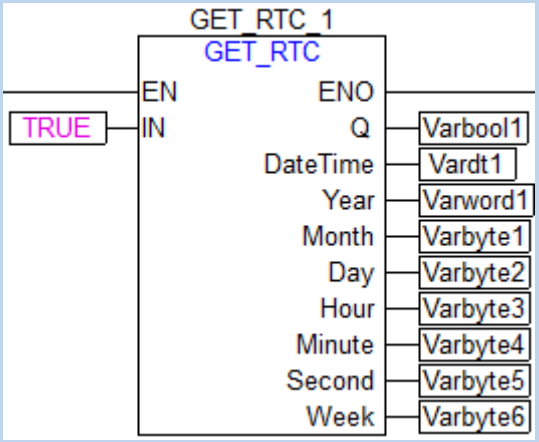
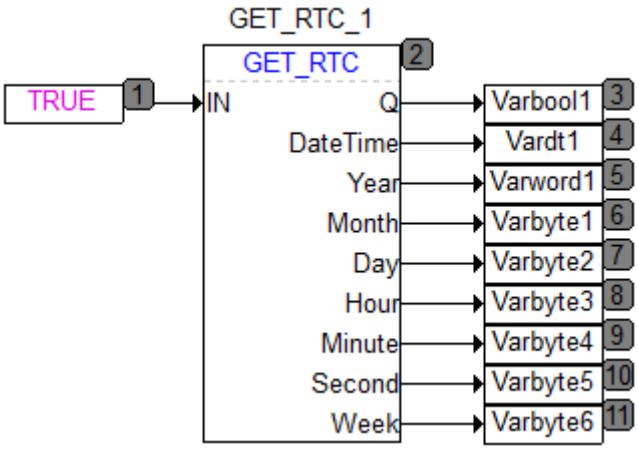
Input Parameter	Data Type	Function Description	Parameter Description	Default Value
IN	BOOL	Enable	0: Invalid 1: Valid	0

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	DT#1971-01-01-00:00:00~ DT#2036-12-31-23:59:59	DT#1970-01-01-00:00:00
Year	WORD	Year	1971~2036	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. Application Example of Instructions

Variable Definition

No.	Name	Description	Data Type	Initial Value	Area
0001	GET_RTC_1		GET_RTC		G Area
0002	Varbool1		BOOL	FALSE	G Area
0003	Vardt1		DT	DT#1970-01-01-00:00:00	G Area
0004	Varword1		WORD	0	G Area
0005	Varbyte1		BYTE	0	G Area
0006	Varbyte2		BYTE	0	G Area
0007	Varbyte3		BYTE	0	G Area
0008	Varbyte4		BYTE	0	G Area
0009	Varbyte5		BYTE	0	G Area
0010	Varbyte6		BYTE	0	G Area

Programing language	Program
Ladder diagram (LD)	
Function Block Diagram(FBD)	

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.

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.

2.4.2 Module Information Instruction

2.4.2.1 sysGetCPUModuleVern (Get Version of CPU)

1. Function

Get firmware version information of controller.

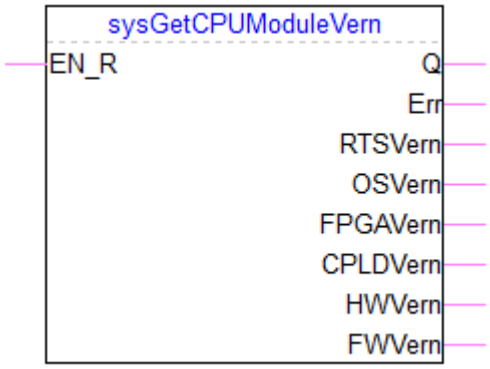


Figure 133 sysGetCPUModuleVern Instruction

2. Parameter Description

Input Parameter	Type	Parameter Description
EN_R	BOOL	Enable FALSE: Invalid (default value) TRUE: Rising edge is valid

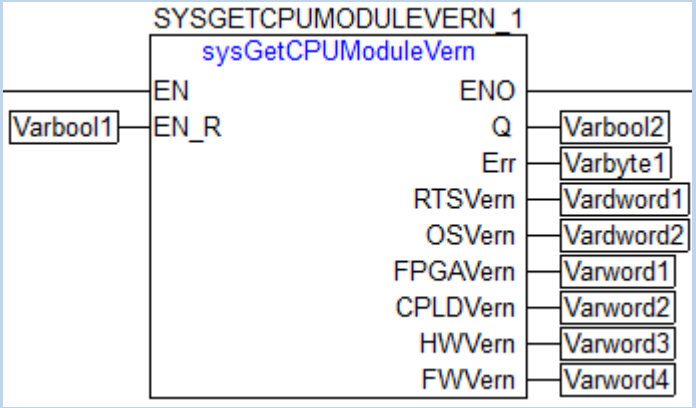
Output Parameter	Type	Parameter Description
Q	BOOL	Output (FALSE, by default) TRUE: Instruction execution successful FALSE: Instruction execution failed
Err	BYTE	Error (0, by default) 128: Get version is failed
RTSVern	DWORD	RTS firmware version build number High 16-bit is kernel 1 RTS version build number, low 16-bit is kernel 0 RTS version build number
OSVern	DWORD	OS version build number High 16-bit is OS version build number of kernel 1 , low 16-bit is OS version build number of kernel 0
FPGAVern	WORD	FPGA version build number

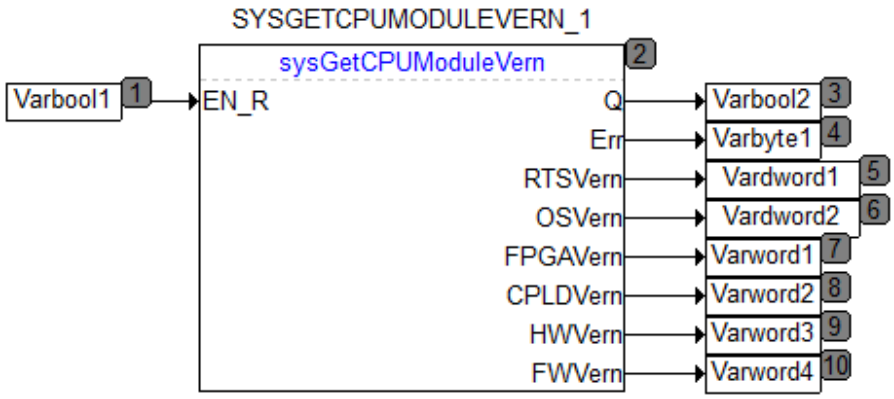
Output Parameter	Type	Parameter Description
CPLDVern	WORD	CPLD version build number
HWVern	WORD	Hardware version number The return value is 1~26, which corresponds to A~Z
FWVern	WORD	Firmware version number The shown version number is 01 in LK220S-A01

3. Application Example of Instructions

Variable Definition

No.	Name	Description	Data Type	Initial Value	Area
0001	SYSGETCPUMODULEVERN_1		SYSGETCPUMODULEVERN		G Area
0002	Varbool1		BOOL	FALSE	G Area
0003	Varbool2		BOOL	FALSE	G Area
0004	Varbyte1		BYTE	0	G Area
0005	Vardword1		DWORD	0	G Area
0006	Vardword2		DWORD	0	G Area
0007	Varword1		WORD	0	G Area
0008	Varword2		WORD	0	G Area
0009	Varword3		WORD	0	G Area
0010	Varword4		WORD	0	G Area

Programing language	Program
Ladder diagram (LD)	

Variable Definition	
Function Block Diagram(FBD)	

2.4.2.2 sysGetComModuleVern (Get Version of Communication Module)

1. Function

Get the firmware version information of the communication module.

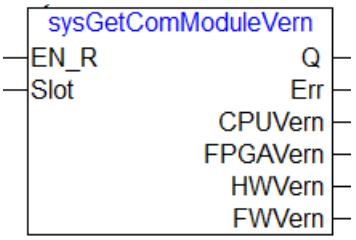


Figure 134 sysGetComModuleVern Instruction

2. Parameter Description

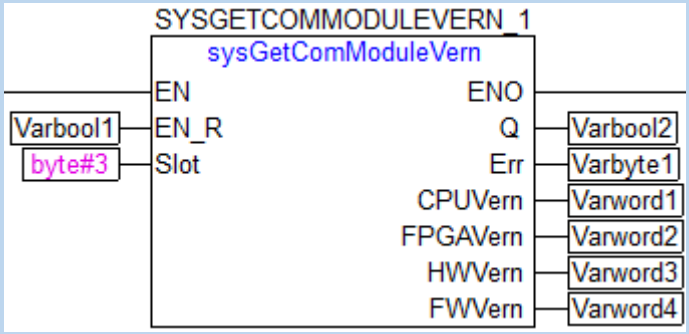
Input Parameter	Type	Parameter Description
EN_R	BOOL	Enable FALSE: Invalid (default value) TRUE: Rising edge is valid
Slot	BYTE	Slot number (2, by default, valid value is 2~5) The slot number of the master control module is fixed to 1 The slot number of other modules is ascended to the right according to the location related to the master control module Input the slot number of the version information module to be obtained during the use

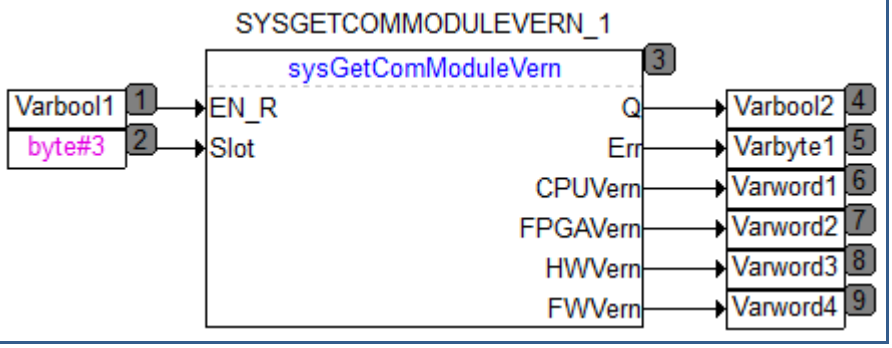
Output Parameter	Type	Parameter Description
Q	BOOL	Output (FALSE, by default) TRUE: Instruction execution successful FALSE: Instruction execution failed
Err	BYTE	Error (0, by default) 0: Correct 1: Slot configuration is error 128: There is no module in corresponding slot 129: Get failed 200: Duplicate definition module and call
CPUVern	WORD	CPU firmware version build number
FPGAVern	WORD	FPGA version build number
HWVern	WORD	Hardware version number The return value is 1~26, which corresponds to A~Z
FWVern	WORD	Firmware version number The shown version number is 01 in LK240S-A01

3. Application Example of Instructions

Variable Definition

No.	Name	Description	Data Type	Initial Value	Area
0001	SYSGETCOMMODULEVERN_1		SYSGETCOMMODULEVERN		G Area
0002	Varbool1		BOOL	FALSE	G Area
0003	Varbool2		BOOL	FALSE	G Area
0004	Varbyte1		BYTE	0	G Area
0005	Varword1		WORD	0	G Area
0006	Varword2		WORD	0	G Area
0007	Varword3		WORD	0	G Area
0008	Varword4		WORD	0	G Area

Programing language	Program
Ladder diagram (LD)	

Variable Definition	
Function Block Diagram(FBD)	

2.4.2.3 sysGetCheckTime (Get Check Time)

1. Function

Get the check time.

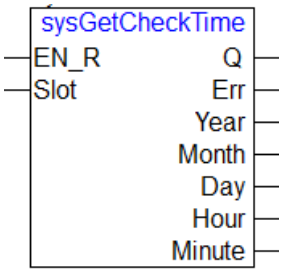


Figure 135 sysGetCheckTime Instruction

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.

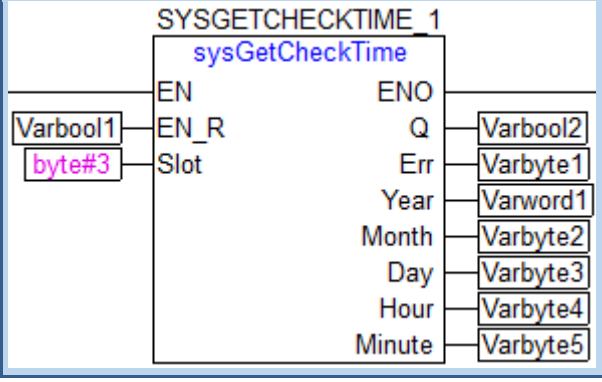
Output Parameter	Type	Parameter Description
Q	BOOL	Output (default value: FALSE) TRUE: Instruction execution successful FALSE: Instruction execution failed
Err	BYTE	Error (default value: 0) 0: Correct 1: Slot configuration is error

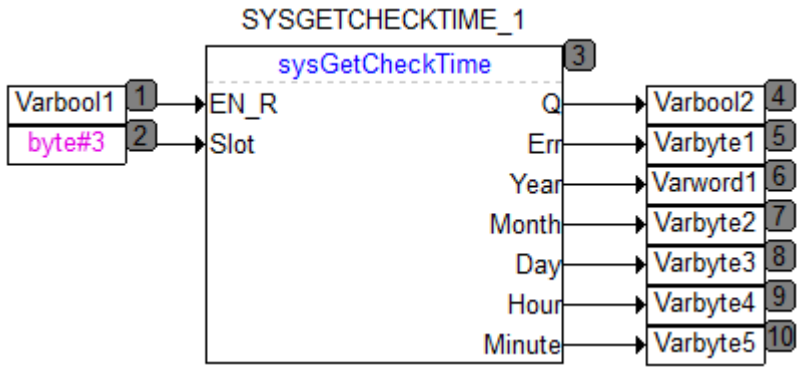
Output Parameter	Type	Parameter Description
		128: There is no module in corresponding slot 129: Get failed 130: Year error(2015~2099) 131: Month error 132: Day error 133: Hour error 134: Minute error 200: Duplicate definition module and call
Year	WORD	Year (default value: 0)
Month	BYTE	Month (default value: 0)
Day	BYTE	Day (default value: 0)
Hour	BYTE	Hour (default value: 0)
Minute	BYTE	Minute (default value: 0)

3. Application Example of Instructions

Variable Definition

No.	Name	Description	Data Type	Initial Value	Area
0001	SYSGETCHECKTIME_1		SYSGETCHECKTIME		G Area
0002	Varbool1		BOOL	FALSE	G Area
0003	Varbool2		BOOL	FALSE	G Area
0004	Varbyte1		BYTE	0	G Area
0005	Varword1		WORD	0	G Area
0006	Varbyte2		BYTE	0	G Area
0007	Varbyte3		BYTE	0	G Area
0008	Varbyte4		BYTE	0	G Area
0009	Varbyte5		BYTE	0	G Area

Programing language	Program
Ladder diagram (LD)	

Variable Definition	
Function Block Diagram(FBD)	

2.4.2.4 sysGetEthMac (Get MAC Address)

1. Function

Get the Mac address.

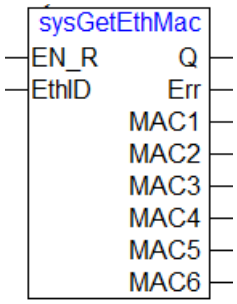


Figure 136 sysGetEthMac Instruction

2. Parameter Description

Input Parameter	Type	Parameter Description
EN_R	BOOL	Enable FALSE: Invalid (by default) TRUE: Rising edge is valid
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

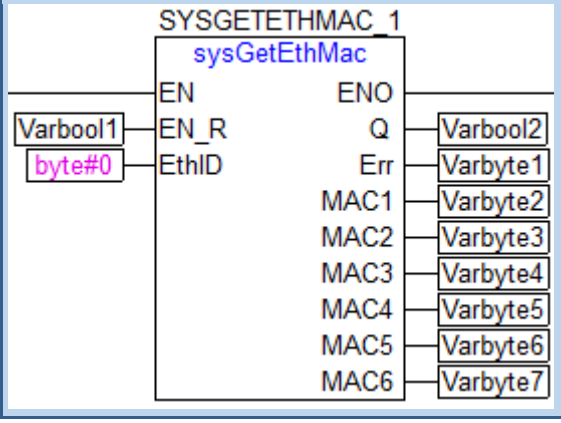
Output Parameter	Type	Parameter Description
Q	BOOL	Output (FALSE, by default) TRUE: Instruction execution successful FALSE: Instruction execution failed
Err	BYTE	Error (0, by default) 1: EthID configuration is error

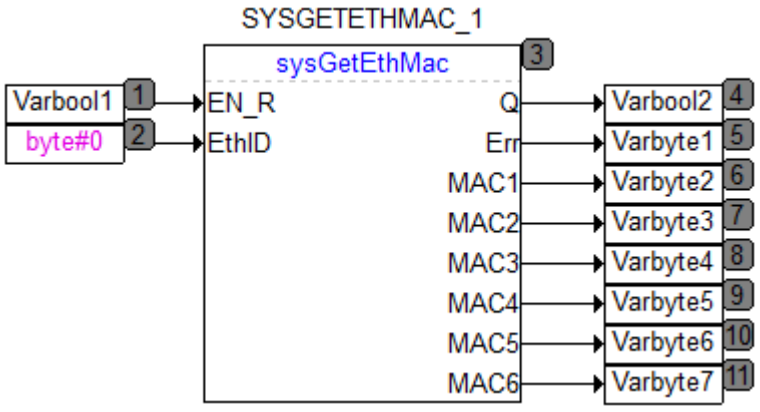
Output Parameter	Type	Parameter Description
		128: Get mac address failed 200: Duplicate definition module and call
MAC1	BYTE	First part of MAC address
MAC2	BYTE	Second part of MAC address
MAC3	BYTE	Third part of MAC address
MAC4	BYTE	Fourth part of MAC address
MAC5	BYTE	Fifth part of MAC address
MAC6	BYTE	Sixth part of MAC address

3. Application Example of Instructions

Variable Definition

No.	Name	Description	Data Type	Initial Value	Area
0001	SYSGETETHMAC_1		SYSGETETHMAC		G Area
0002	Varbool1		BOOL	FALSE	G Area
0003	Varbool2		BOOL	FALSE	G Area
0004	Varbyte1		BYTE	0	G Area
0005	Varbyte2		BYTE	0	G Area
0006	Varbyte3		BYTE	0	G Area
0007	Varbyte4		BYTE	0	G Area
0008	Varbyte5		BYTE	0	G Area
0009	Varbyte6		BYTE	0	G Area
0010	Varbyte7		BYTE	0	G Area

Programing language	Program
Ladder diagram (LD)	

Variable Definition	
Function Block Diagram(FBD)	

Program description:

The got MAC address of first Ethernet card.

2.4.2.5 sysGetBatteryAlarm (Get Battery Alarm)

1. Function

Get the battery power information of the master control module.

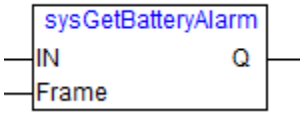


Figure 137 sysGetBatteryAlarm Instruction

2. Parameter Description

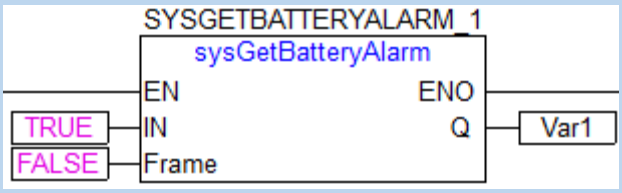
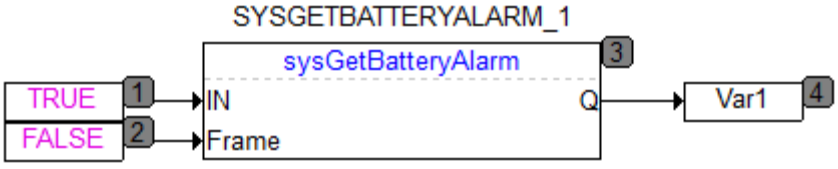
Input Parameter	Type	Parameter Description
IN	BOOL	High level enable (FALSE, by default)
Frame	BOOL	Select frame (FALSE, by default) TRUE: Slave frame FALSE: Master frame

Output Parameter	Type	Parameter Description
Q	BOOL	Battery power TURE: No battery or low battery. FALSE: The battery power is normal.

3. Application Example of Instructions

Variable Definition

No.	Name	Description	Data Type	Initial Value	Area
0001	SYSGETBATTERYALARM_1		SYSGETBATTERYALARM		G Area
0002	Var1		BOOL	FALSE	G Area

Programing language	Program
Ladder diagram (LD)	
Function Block Diagram(FBD)	

2.4.2.6 sysGetTEMP (Get Temperature of CPU)

1. Function

Get the temperature information of the master control module.

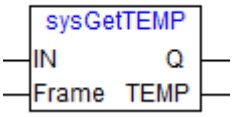


Figure 138 sysGetTEMP Instruction

2. Parameter Description

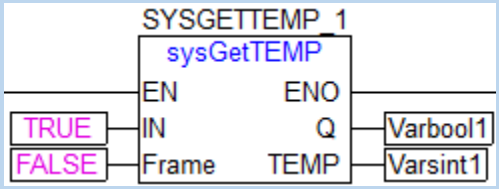
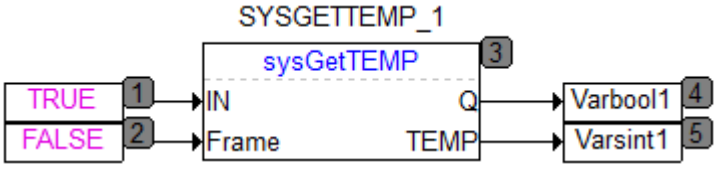
Input Parameter	Type	Parameter Description
IN	BOOL	High level enable (FALSE, by default)
Frame	BOOL	Select frame (FALSE, by default) TRUE: Slave frame FALSE: Master frame

Output Parameter	Type	Parameter Description
Q	BOOL	Output (FALSE, by default) TRUE: Instruction execution successful FALSE: Instruction execution failed
TEMP	SINT	Temperatuer value with the unit celsius (0, by default)

3. Application Example of Instructions

Variable Definition

No.	Name	Description	Data Type	Initial Value	Area
0001	SYSGETTEMP_1		SYSGETTEMP		G Area
0002	Varbool1		BOOL	FALSE	G Area
0003	Varsint1		SINT	0	G Area

Programing language	Program
Ladder diagram (LD)	
Function Block Diagram(FBD)	

2.4.2.7 sysGetModel (Get Model Type)

1. Function

Get the model of modules.

Figure 139 sysGetModel Instruction

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

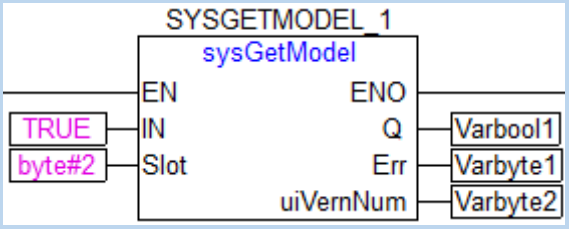
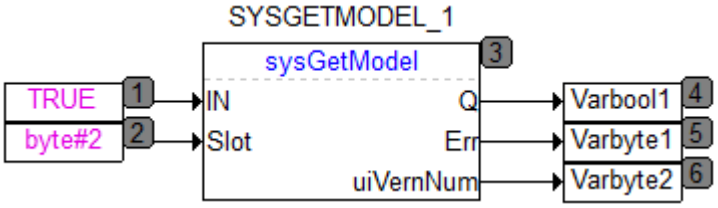
Output Parameter	Type	Parameter Description
Q	BOOL	Output (default value: FALSE) TRUE: Instruction execution successful FALSE: Instruction execution failed

Output Parameter	Type	Parameter Description
Err	BYTE	Error (default value: 0) 1: Slot number is error 128: There is no module in corresponding slot
uiVernNum	BYTE	Model ID of modules (default value: 0) 1: LK220S 2: LK249S 3: LK240S

3. Application Example of Instructions

Variable Definition

No.	Name	Description	Data Type	Initial Value	Area
0001	SYSGETMODEL_1		SYSGETMODEL		G Area
0002	Varbool1		BOOL	FALSE	G Area
0003	Varbyte1		BYTE	0	G Area
0004	Varbyte2		BYTE	0	G Area

Programing language	Program
Ladder diagram (LD)	
Function Block Diagram(FBD)	

2.4.2.8 sysGetModuleSN (Get Serial Number of Module)

1. Function

Get the serial number of modules.

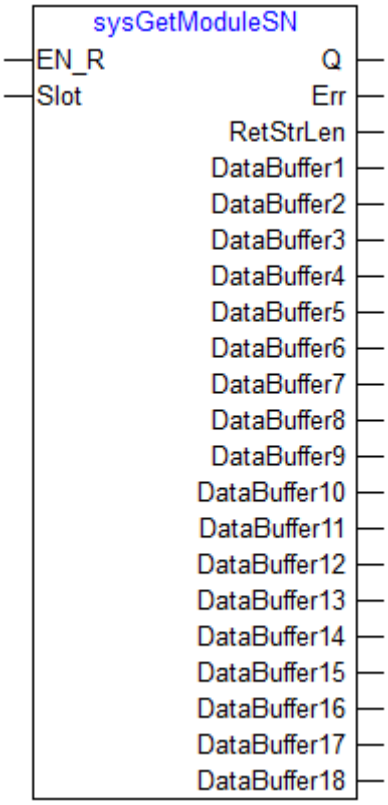


Figure 140 sysGetModuleSN Instruction

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

Output Parameter	Type	Parameter Description
Q	BOOL	Output (FALSE, by default) TRUE: Instruction execution successful FALSE: Instruction execution failed
Err	BYTE	Error (0, by default) 0: Correct 1: Slot number is error 128: There is no module in corresponding slot

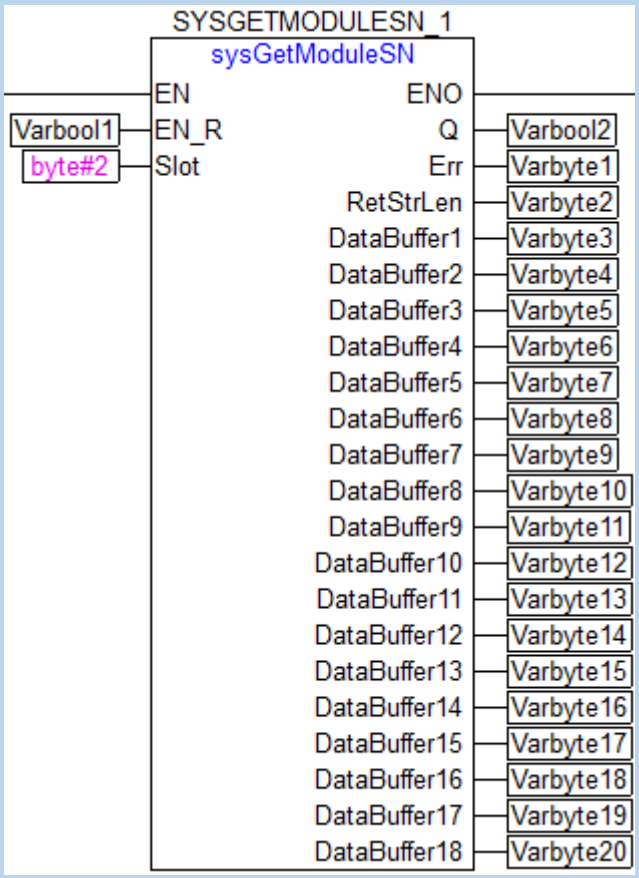
Output Parameter	Type	Parameter Description
		130: Information returned is error or the correct data is not obtained temporarily in reading (it will be 0 after reading successfully) 200: Duplicate definition module and call
RetStrLen	BYTE	String length for serial number of modules (0, by default), up support for 18-bit ASCII string, beyond the content is cut off automatically
uiDataBuffer[1]~ uiDataBuffer[18]	BYTE	Module sequence number, 18 bytes, a byte represents a ASCII code

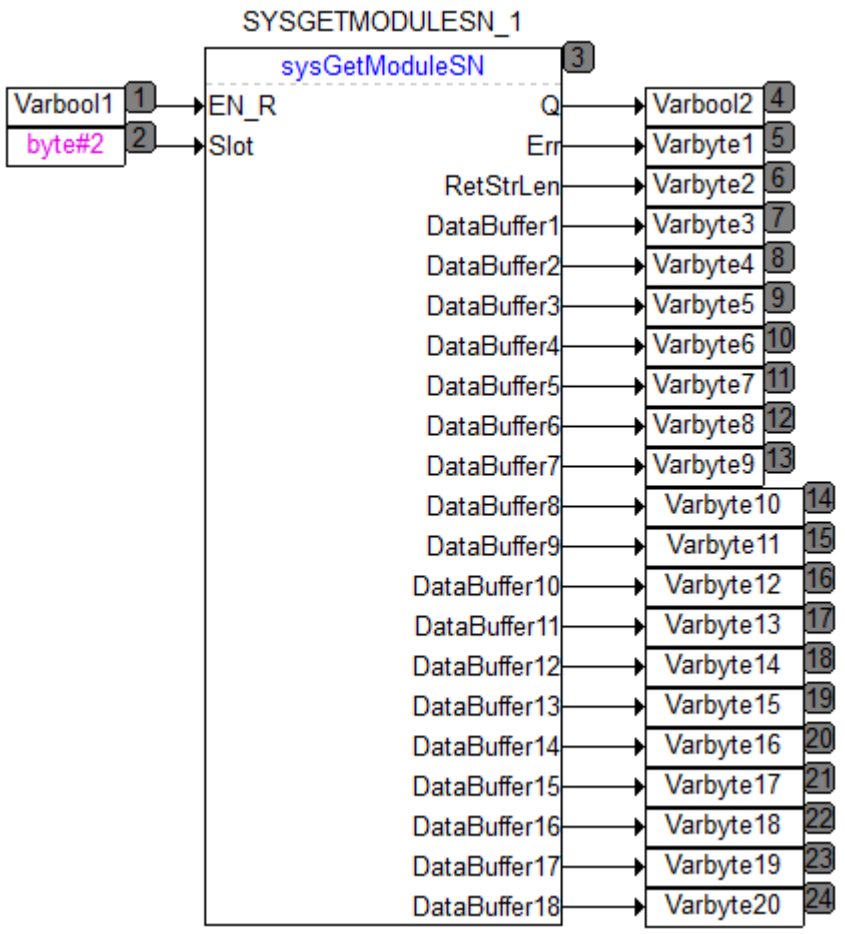
3. Application Example of Instructions

Variable Definition

No.	Name	Description	Data Type	Initial Value	Area
0001	SYSGETMODULESN_1		SYSGETMODULESN		G Area
0002	Varbool1		BOOL	FALSE	G Area
0003	Varbool2		BOOL	FALSE	G Area
0004	Varbyte1		BYTE	0	G Area
0005	Varbyte2		BYTE	0	G Area
0006	Varbyte3		BYTE	0	G Area
0007	Varbyte4		BYTE	0	G Area
0008	Varbyte5		BYTE	0	G Area
0009	Varbyte6		BYTE	0	G Area
0010	Varbyte7		BYTE	0	G Area
0011	Varbyte8		BYTE	0	G Area
0012	Varbyte9		BYTE	0	G Area
0013	Varbyte10		BYTE	0	G Area
0014	Varbyte11		BYTE	0	G Area
0015	Varbyte12		BYTE	0	G Area
0016	Varbyte13		BYTE	0	G Area
0017	Varbyte14		BYTE	0	G Area
0018	Varbyte15		BYTE	0	G Area
0019	Varbyte16		BYTE	0	G Area
0020	Varbyte17		BYTE	0	G Area
0021	Varbyte18		BYTE	0	G Area
0022	Varbyte19		BYTE	0	G Area
0023	Varbyte20		BYTE	0	G Area

Programing language	Program

Variable Definition	
Ladder diagram (LD)	

Variable Definition	
Function Block Diagram(FBD)	

2.4.3 System State Instruction

2.4.3.1 sysGetRunTime (Get Task Running Time)

1. Function

Get the task running time of the controller.

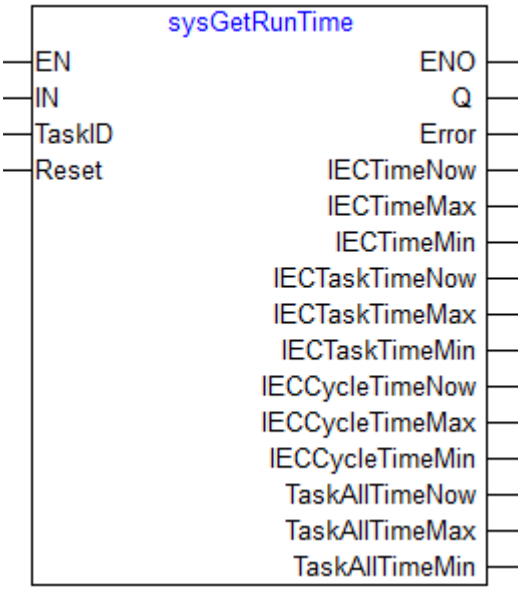


Figure 141 sysGetRunTime Instruction

2. Parameter Description

Input Parameter	Type	Parameter Description
IN	BOOL	High level enable (FALSE, by default)
TaskID	BYTE	IEC task ID,1~8 LK220S 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 output parameter, high level enable (FALSE, by default)

Output Parameter	Type	Parameter Description
Q	BOOL	Output (FALSE, by default) TRUE: Instruction execution successful FALSE: Instruction execution failed
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)

Output Parameter	Type	Parameter Description
IECTimeMin	DWORD	Minimum run time of current configuration logic (0, by default)
IECTaskTimeNow	DWORD	Current run time of current configuration logic task (0, by default)
IECTaskTimeMax	DWORD	Maximum run time of current configuration logic task (0, by default)
IECTaskTimeMin	DWORD	Minimum run time of current configuration logic task (0, by default)
IECCycleTimeNow	DWORD	Current run time of current configuration logic task cycle (0, by default)
IECCycleTimeMax	DWORD	Maximum run time of current configuration logic task cycle (0, by default)
IECCycleTimeMin	DWORD	Minimum run time of current configuration logic 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)

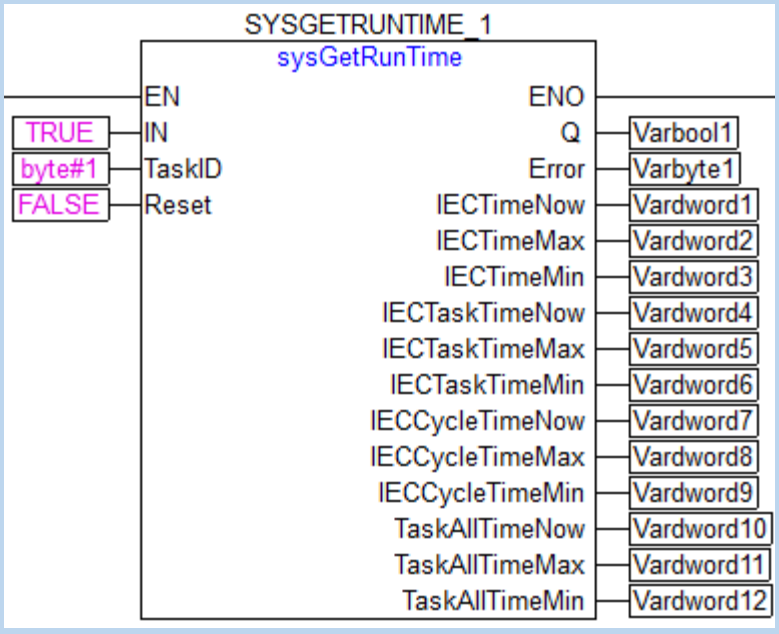
- Detailed description of parameters
  - IECTime: The time taken to execute all of the POU in the current project.
  - IECTaskTime: The execution time of the configuration logic task, including logical task scheduling and self-checking time, excluding redundant time.
  - IECCycleTime: Configuration logic task execution cycle.
  - TaskALLTime: All execution times of current logical tasks, including logical task scheduling, self-checking time, and redundant time.

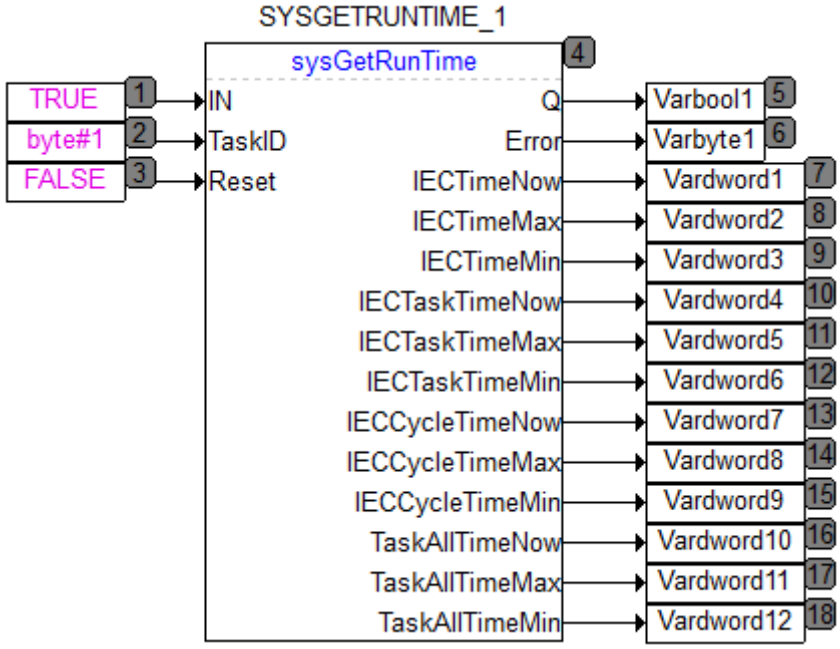
- When external operations occur, such as changing the state of the dial switch, AT reset, AT cold reset, AT download, etc., it is recommended that the user set the Reset pin to TRUE to reset the output value. After the data is stable, obtain the latest time data.

3. Application Example of Instructions

Variable Definition

No.	Name	Description	Data Type	Initial Value	Area
0001	SYSGETRUNTIME_1		SYSGETRUNTIME		G Area
0002	Varbool1		BOOL	FALSE	G Area
0003	Varbyte1		BYTE	0	G Area
0004	Vardword1		DWORD	0	G Area
0005	Vardword2		DWORD	0	G Area
0006	Vardword3		DWORD	0	G Area
0007	Vardword4		DWORD	0	G Area
0008	Vardword5		DWORD	0	G Area
0009	Vardword6		DWORD	0	G Area
0010	Vardword7		DWORD	0	G Area
0011	Vardword8		DWORD	0	G Area
0012	Vardword9		DWORD	0	G Area
0013	Vardword10		DWORD	0	G Area
0014	Vardword11		DWORD	0	G Area
0015	Vardword12		DWORD	0	G Area

Programing language	Program
Ladder diagram (LD)	

Variable Definition	
Function Block Diagram(FBD)	

2.4.3.2 sysGetKeyState (Get Switch State of Key)

1. Function

Get the key switch state of the CPU module.

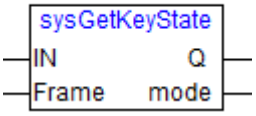


Figure 142 sysGetKeyState Instruction

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

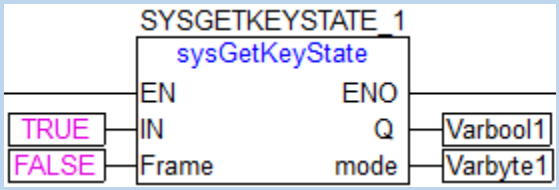
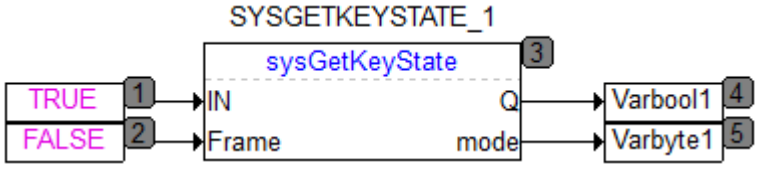
Output Parameter	Type	Parameter Description
Q	BOOL	Output (FALSE, by default) TRUE: Instruction execution successful FALSE: Instruction execution failed

Output Parameter	Type	Parameter Description
Mode	BYTE	Key switch state (0, by default) 1: RUN, run mode 2: REM, remote mode 3: PRG, programming mode

3. Application Example of Instructions

Variable Definition

No.	Name	Description	Data Type	Initial Value	Area
0001	SYSGETKEYSTATE_1		SYSGETKEYSTATE		G Area
0002	Varbool1		BOOL	FALSE	G Area
0003	Varbyte1		BYTE	0	G Area

Programing language	Program
Ladder diagram (LD)	
Function Block Diagram(FBD)	

2.4.3.3 sysGetDPMasterState (Get Current State of DP Master)

1. Function

Get the diagnosis information of the DP master.

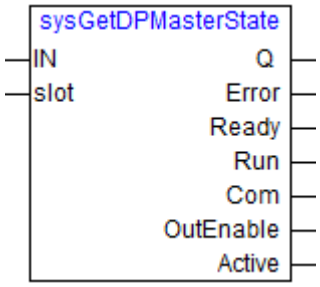


Figure 143 sysGetDPMasterState Instruction

2. Parameter Description

Input Parameter	Type	Parameter Description
IN	BOOL	High level enable (FALSE, by default)

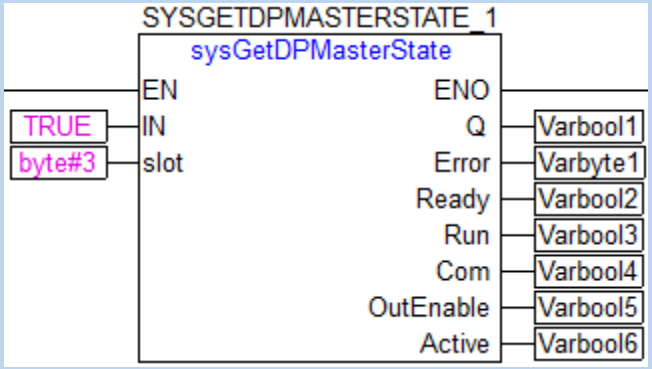
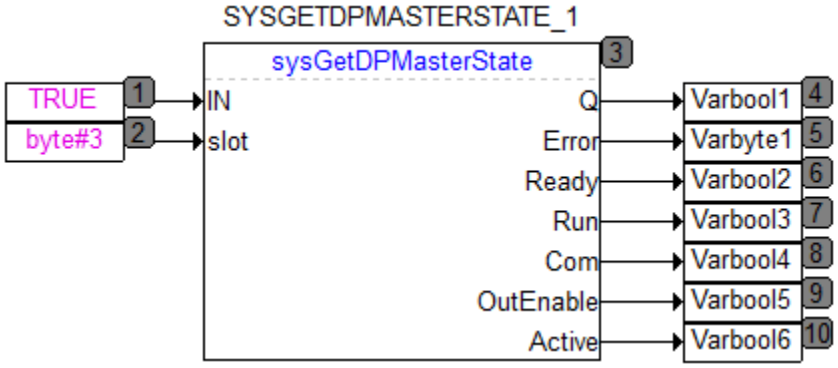
Input Parameter	Type	Parameter Description
Slot	BYTE	Slot number (2, by default, valid value from 2~5) The slot number of the master control module is fixed to 1 The slot number of other modules is ascended to the right according to the location related to the master control module Input the slot number of the version information module to be obtained during the use

Output Parameter	Type	Parameter Description
Q	BOOL	Output (FALSE, by default) TRUE: Instruction execution successful FALSE: Instruction execution failed
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 initialization of the master station 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. Application Example of Instructions

Variable Definition

No.	Name	Description	Data Type	Initial Value	Area
0001	SYSGETDPMASTERSTATE_1		SYSGETDPMASTERSTATE		G Area
0002	Varbool1		BOOL	FALSE	G Area
0003	Varbyte1		BYTE	0	G Area
0004	Varbool2		BOOL	FALSE	G Area
0005	Varbool3		BOOL	FALSE	G Area
0006	Varbool4		BOOL	FALSE	G Area
0007	Varbool5		BOOL	FALSE	G Area
0008	Varbool6		BOOL	FALSE	G Area

Programing language	Program
Ladder diagram (LD)	
Function Block Diagram(FBD)	

2.4.3.4 sysGetCPULoad (Get Current Load of CPU)

1. Function

Get the CPU and memory utilization of the controller.

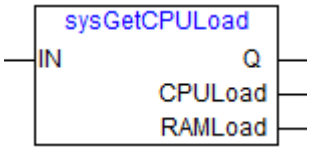


Figure 144 sysGetCPULoad Instruction

2. Parameter Description

Input Parameter	Type	Parameter Description
IN	BOOL	High level enable (FALSE, by default)

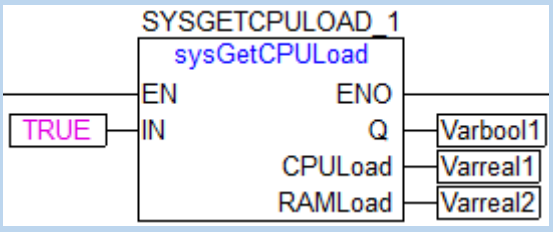
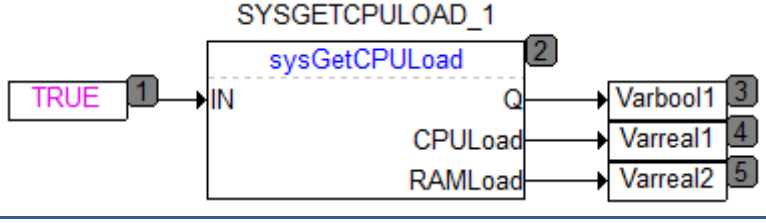
Output Parameter	Type	Parameter Description
Q	BOOL	Output (FALSE, by default) TRUE: Instruction execution successful FALSE: Instruction execution failed
CPULoad	REAL	Utilization ratio of CPU (0, by default), unit:%
RAMLoad	REAL	Utilization ratio of memory(0, by default), unit:%

- CPU load: The time taken for DP data input to DP data output divided by the IEC task execution cycle. If the load exceeds 100%, it indicates that the actual required task cycle is longer than the set value, and the task cycle should be increased. It is recommended to keep the controller's CPU load below 80%.
- RAM load: Usage ratio of data area, engineering data size / total size of data area.

3. Application Example of Instructions

Variable Definition

No.	Name	Description	Data Type	Initial Value	Area
0001	SYSGETCPULOAD_1		SYSGETCPULOAD		G Area
0002	Varbool1		BOOL	FALSE	G Area
0003	Varreal1		REAL	0	G Area
0004	Varreal2		REAL	0	G Area

Programing language	Program
Ladder diagram (LD)	
Function Block Diagram(FBD)	

2.4.3.5 sysGetRedState (Get Redundant State of System)

1. Function

Get the redundancy state information of the master and slave frames.

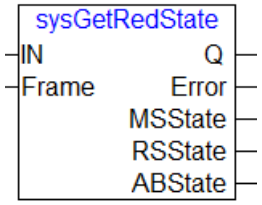


Figure 145 sysGetRedState Instruction

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

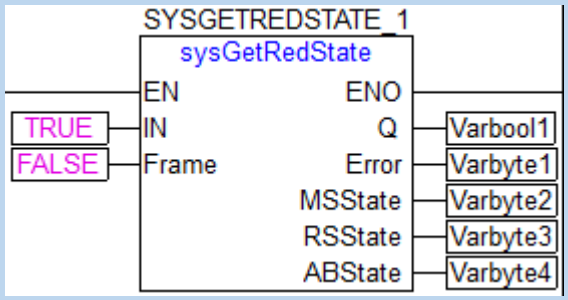
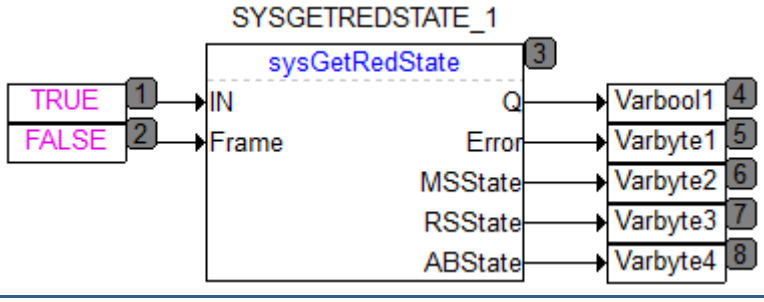
Output Parameter	Type	Parameter Description
Q	BOOL	Output (FALSE, by default) TRUE: Instruction execution successful FALSE: Instruction execution failed
Error	BYTE	Error (0, by default) 128: Current CPU is redundancy abnormal state (corresponding to 0~6 and 11 in states of master and slave frames) 129: Key switch sate of master CPU and slave CPU is inconsistent 130: Data is out of sync between the master and slave
MSState	BYTE	State of master and slave frames (0, by default) 0: Unknown state 1: Initial state 2: Hardware Ready state 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

Output Parameter	Type	Parameter Description
RSState	BYTE	Standalone and dual frames state (0, by default) 0: Unknown state 1: Standalone frame 2: Dual frames
ABState	BYTE	ABsystems state (0, by default) 0: Unknown state 1: A system 2: B system

3. Application Example of Instructions

Variable Definition

No.	Name	Description	Data Type	Initial Value	Area
0001	SYSGETREDSTATE_1		SYSGETREDSTATE		G Area
0002	Varbool1		BOOL	FALSE	G Area
0003	Varbyte1		BYTE	0	G Area
0004	Varbyte2		BYTE	0	G Area
0005	Varbyte3		BYTE	0	G Area
0006	Varbyte4		BYTE	0	G Area

Programing language	Program
Ladder diagram (LD)	
Function Block Diagram(FBD)	

2.4.3.6 sysMasterSwitchToSlave (Master Switch to Slave)

1. Function

Switch the master frame to the slave frame.

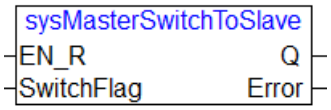


Figure 146 sysMasterSwitchToSlave Instruction

2. Parameter Description

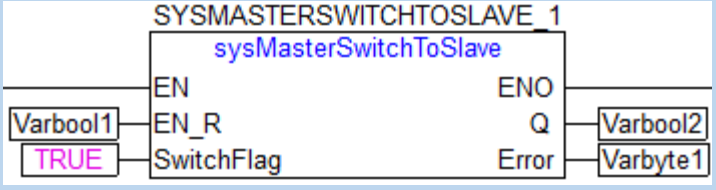
Input Parameter	Type	Parameter Description
EN_R	BOOL	Enable FALSE: Invalid (by default) TRUE: Rising edge effective
SwitchFlag	BOOL	FALSE: Not allowed to switch TRUE: Allowed to switch

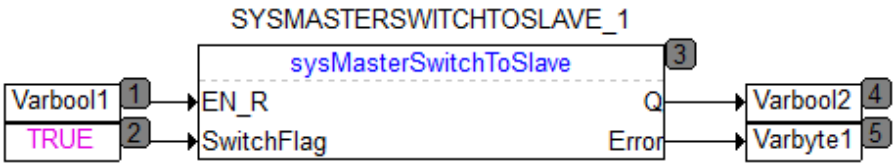
Output Parameter	Type	Parameter Description
Q	BOOL	Output (FALSE, by default) TRUE: Instruction execution successful FALSE: Instruction execution failed
Error	BYTE	Error (0, by default) 0: Correct 128: Standalone system 129: The master/slave synchronization is not completed 130: The key swtich of the slave frame is in the PRG state

3. Application Example of Instructions

Variable Definition

No.	Name	Description	Data Type	Initial Value	Area
0001	SYSMASTERSWITCHTOSLAVE_1		SYSMASTERSWITCHTOSLAVE		G Area
0002	Varbool1		BOOL	FALSE	G Area
0003	Varbool2		BOOL	FALSE	G Area
0004	Varbyte1		BYTE	0	G Area

Programing language	Program
Ladder diagram (LD)	

Variable Definition	
Function Block Diagram(FBD)	

2.4.3.7 sysGetDPSlaveState (Get Diagnosis of DP Slave)

1. Function

Get the diagnosis information of the DP slave station.

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) TRUE: Instruction execution successful FALSE: Instruction execution failed
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 LKS Slave station module is 0x0500~0x05FF
DiagDataLen	BYTE	Diagnostic information length of Slave station
DiagData1~ DiagData54	BYTE	Diagnostic information (Different meanings to each module)

Output Parameter	Type	Parameter Description
		Please see diagnosis information of each IO in *LKS Safety Control System Product Manual* DiagData51: Fault status of transceiver of slave station AB network bit0: Fault status of A network transceiver bit1: Fault status of B network transceiver

2.4.3.8 sysGetCPUDiagInfo (Get Diagnosis of CPU)

1. Function

Get the diagnosis information of the master control module.

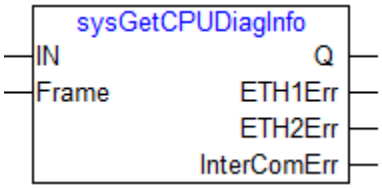


Figure 147 sysGetCPUDiagInfo Instruction

2. Parameter Description

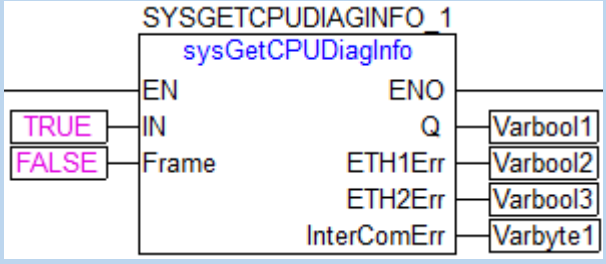
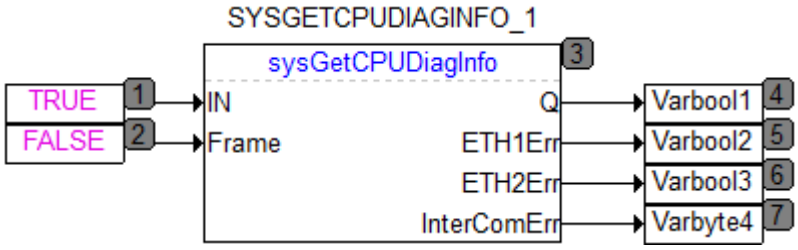
Input Parameter	Type	Parameter Description
IN	BOOL	High level enable (FALSE, by default)
Frame	BOOL	Select frame (FALSE, by default) TRUE: Slave frame FALSE: Master frame

Output Parameter	Type	Parameter Description
Q	BOOL	Output (FALSE, by default) TRUE: Instruction execution successful FALSE: Instruction execution failed
ETH1Err	BOOL	Fault information of first Ethernet, this vlaue is TRUE when the network cable is disconnected (FALSE, by default)
ETH2Err	BOOL	Fault information of second Ethernet, this vlaue 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 slots 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~bit7 : reserved

3. Application Example of Instructions

Variable Definition

No.	Name	Description	Data Type	Initial Value	Area
0001	SYSGETCPULOAD_1		SYSGETCPULOAD		G Area
0002	Varbool1		BOOL	FALSE	G Area
0003	Varreal1		REAL	0	G Area
0004	Varreal2		REAL	0	G Area

Programing language	Program
Ladder diagram (LD)	
Function Block Diagram(FBD)	

Program description:

Example for getting the diagnosis information of CPU module in master frame.

2.4.3.9 sysGetComModuleDiagInfo (Get Diagnosis of Communication Module)

1. Function

Get the diagnosis information of the communication module.

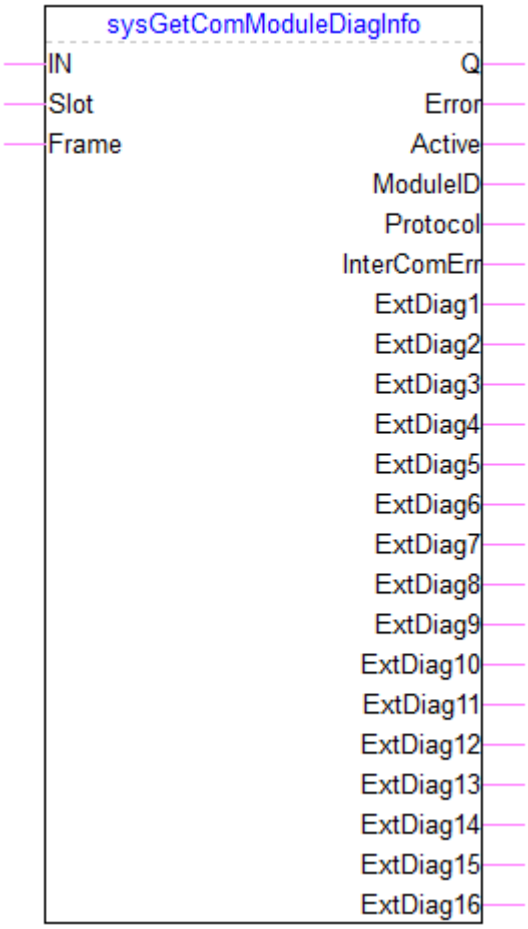


Figure 148 sysGetComModuleDiagInfo Instruction

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 (FALSE, by default) TRUE: Slave frame FALSE: Master frame

Output Parameter	Type	Parameter Description
Q	BOOL	Output (FALSE, by default) TRUE: Instruction execution successful FALSE: Instruction execution failed
Error	BYTE	Error (0, by default) 0: Correct 1: Slot configuration is error 128: There is no module on the slot of local frame 129: There is no module on the slot of opposite frame
Active	BOOL	Working state of communication module (0, by default) FALSE: Offline TRUE: Online
ModuleID	BYTE	Module ID number 1 :LK220 (reserved) 2: LK240 3: LK249 4: LK247 5: LK248 6: LK220S (reserved) 7: LK240S 8: LK249S
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
InterComErr	BOOL	Fault information of internal communication link, this value is TRUE when the fault occurs (0, by default).
ExtDiag1	BYTE	Extended diagnosis information LK240S: The first byte represents the first optical fiber is broken (the corresponding value is 1). LK249S: The first byte represents DP1 is broken (the corresponding value is 1).
ExtDiag2	BYTE	Extended diagnosis information LK240S: The second byte represents the second optical fiber is broken (the corresponding value is 1). LK249S: The second byte represents DP2 is broken (the corresponding value is 1).

Output Parameter	Type	Parameter Description
ExtDiag3	BYTE	Extended diagnosis information LK240S: The third byte represents the A/B switch conflict occurs (the corresponding value is 1). LK249S: The third byte is reserved.
ExtDiag4	BYTE	Extended diagnosis information LK240S: The fourth byte represents clear the flag of watchdog reset (the corresponding value is 1). LK249S: The fourth byte represents clear the flag of watchdog reset (the corresponding value is 1).
ExtDiag5	BYTE	Extended diagnosis information LK240S: The fifth byte is reserved. LK249S: The fifth byte is reserved.
ExtDiag6	BYTE	Extended diagnosis information LK240S: The sixth byte represents the bit error rate of the first optical fiber (0~100). LK249S: The sixth byte is reserved.
ExtDiag7	BYTE	Extended diagnosis information LK240S: The seventh byte represents the bit error rate of the second optical fiber (0~100). LK249S: The seventh byte is reserved.
ExtDiag8~16	BYTE	Reserved

2.4.3.10 sysSetProfiOareq(Safety slave station Profisafe confirmation)

1. Function

Profisafe confirmation for safety slave station.

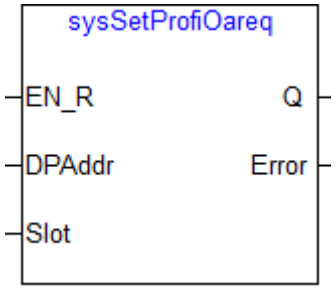


Figure 149 sysSetProfiOareq Instruction

2. Parameter Description

Input Parameter	Type	Parameter Description
EN_R	BOOL	Rising edge enable (FALSE, by default)
DPAddr	BYTE	Slave station address (default value is 2, 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) TRUE: Instruction execution successful FALSE: Instruction execution failed
Error	BYTE	Error (0, by default) 0: Correct 1: Slot number is error 2: Safety slave station does not need to confirm 3: Safety slave station confirmation failed

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