07-4 Basic instruction

last modified by Joey

on 2022/06/09 15:07

Table of Contents

Data conversion instruction

BCD/BIN → BCD

BCD(P)

Convert the BIN data of the device specified in (s) to BCD, and store it in the device specified in (d).

The calculation of the CPU module uses BIN (binary number) data for processing, which is used to display values in a 7-segment display equipped with a BCD decoder.

-[BCD (s) (d)]

Content, range and data type

Parameter	Content	Range	Data type	Data type (label)
(s)	BIN data or start device storing BIN data	0 to 9999	Signed BIN16	ANY16
(d)	Start device for storing BCD data	-	BCD 4 digits	ANY16

Device used

Instruction	Parameter	Devices											Offset modification	Pulse extension
		KnX	KnY	KnM	KnS	T	C	D	R	SD	K	H	[D]	XXP
BCD	Parameter 1	●	●	●	●	●	●	●	●	●	●	●	●	●
	Parameter 2		●	●	●	●	●	●	●	●	●	●	●	●

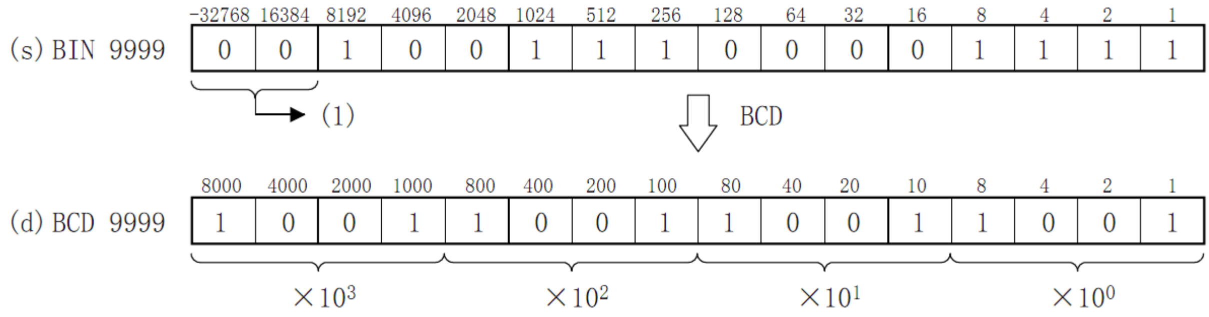
Features

The BIN 16-bit data (0 to 9999) of the device specified in (s) is converted to BCD 4-bit data and stored in the device specified in (d).

The data specified in (s) can be converted within the range of 0 to 9999 (BCD).

When the data specified in (s) or (d) is digit specification, the conditions are as shown in the table below.

(1): Must be set to 0.



The data specified in (s) can be converted in the range of K0 to K9999 by BCD (decimal number).

When the data specified in (s) or (d) is digit specification, the conditions are as shown in the table below.

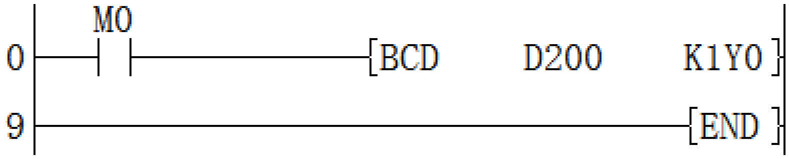
(d)	Digits	Data range
K1Y0	1-bit	0 to 9
K2Y0	2-bit	00 to 99
K3Y0	3-bit	000 to 999
K4Y0	4-bit	0000 to 9999

#**Note:** The four arithmetic operations (+-×÷), increment, decrement instructions and other operations in the CPU module are all performed by BIN (binary number). Therefore, when sending BCD (decimal) digital switch information to the CPU module, please use the BIN(P) command (BCD→BIN conversion transfer command). In addition, when outputting to the 7-segment display of BCD (decimal number), please use the BCD(P) command (BIN→BCD conversion transmission).

Error code

Error code	Content
4084H	The data input in the application instruction (s) exceeds the specified range
4085H	The output result of the read application instruction (s) exceeds the device range
4086H	The output result of the write application instruction (d) exceeds the device range

Example



When M0 is set, the BIN value of D200 is converted into BCD and stored in K1Y0.

BIN/4-bit BCD → BIN

BIN(P)

Convert the BCD data of the device specified in (s) to BIN and store it in the device specified in (d).

Similar to the digital switch, it converts the value set in BCD (decimal number) to BIN (binary number) that can be operated by the CPU module and is used for reading.

-[BIN (s) (d)]

Content, range and data type

Parameter	Content	Range	Data type	Data type (label)
(s)	BCD data or start device storing BIN data	0 to 9999	BCD 4 digits	ANY16
(d)	Start device for storing BIN data	-	Signed BIN16	ANY16

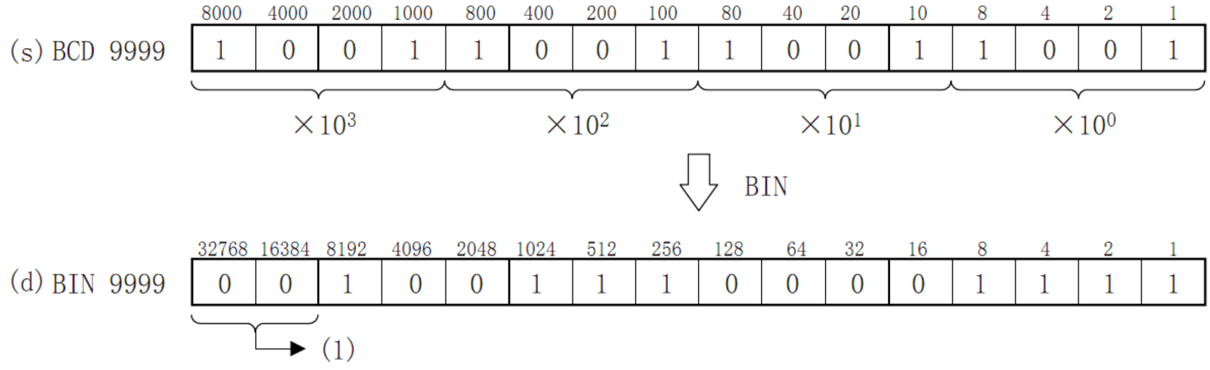
Device used

Instruction	Parameter	Devices											Offset modification	Pulse extension
		KnX	KnY	KnM	KnS	T	C	D	R	SD	K	H	[D]	XXP
BIN	Parameter 1	●	●	●	●	●	●	●	●	●	●	●	●	●
	Parameter 2		●	●	●	●	●	●	●	●			●	●

Features

The BCD 4-bit data (0 to 9999) of the device specified in (s) is converted into BIN 16-bit data and stored in the device specified in (d).

(1): Must become 0.



(1)： 必须变为0。

The data specified in (s) can be converted within the range of 0 to 9999 (BCD).

When the data specified in (s) or (d) is digit specification, the conditions are as shown in the table below.

(d)	Digits	Data range
K1X0	1-bit	0 to 9
K2X0	2-bit	00 to 99
K3X0	3-bit	000 to 999
K4X0	4-bit	0000 to 9999

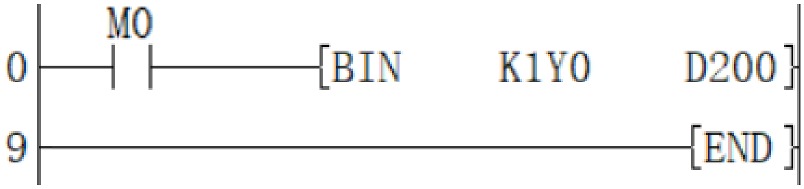
#**Note:** The calculations in the CPU module such as the four arithmetic operations (+-×÷), increment and decrement instructions are all performed by BIN (binary number). Therefore, when sending BCD (decimal) digital switch information to the CPU module, please use the BIN(P) command (BCD→BIN conversion transfer command).

In addition, when outputting to the 7-segment display of BCD (decimal number), please use the BCD(P) command (BIN→BCD conversion transmission).

Error code

Error code	Content
4084H	The data input in the application instruction (s) exceeds the specified range
4085H	The output result of the read application instruction (s) exceeds the device range
4086H	The output result of the write application instruction (d) exceeds the device range

Example



When M0 is set, the BCD value of K1Y0 is converted into BIN and stored in D200.

DBIN/8-bit BCD → BIN

DBIN(P)

Convert the BCD data of the device specified in (s) to BIN and store it in the device specified in (d).

Similar to the digital switch, it converts the value set in BCD (decimal number) to BIN (binary number) that can be operated by the CPU module and is used for reading.

-[DBIN (s) (d)]

Content, range and data type

Parameter	Content	Range	Data type	Data type (label)
(s)	BCD data or start device storing BIN data	0 to 99999999	BCD 8 digits	ANY32
(d)	Start device for storing BIN data	-	Signed BIN32	ANY32

Device used

Instruction	Parameter	Devices													Offset modification	Pulse extension
		KnX	KnY	KnM	KnS	T	C	D	R	SD	LC	HSC	K	H	[D]	XXP
DBIN	Parameter 1	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●
	Parameter 2		●	●	●	●	●	●	●	●					●	●

Features

The BCD 8-bit data (0 to 99999999) of the device specified in (s) is converted to BIN 32-bit data and stored in the device specified in (d).

(1): Must become 0.

The data specified in (s) can be converted within the range of 0 to 99999999 (BCD).

When the data specified in (s) or (d) is digit specification, the conditions are as shown in the table below.

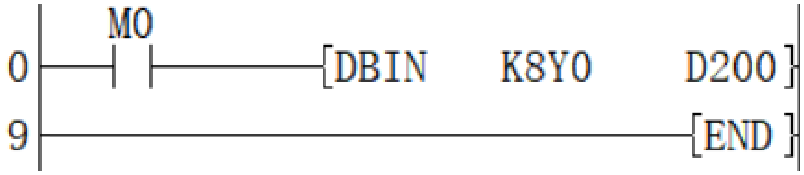
(d)	**Bit**	**data range**
K1X0	1-bit	0 to 9
K2X0	2-bit	00 to 99
K3X0	3-bit	000 to 999
K4X0	4-bit	0000 to 9999
K5X0	5-bit	00000 to 99999
K6X0	6-bit	000000 to 999999
K7X0	7-bit	0000000 to 9999999
K8X0	8-bit	00000000 to 99999999

#**Note:** The calculations in the CPU module such as the four arithmetic operations (+-×÷), increment and decrement instructions are all performed by BIN (binary number). Therefore, when sending BCD (decimal) digital switch information to the CPU module, please use the BIN(P) command (BCD→BIN conversion transfer command). In addition, when outputting to the 7-segment display of BCD (decimal number), please use the BCD(P) command (BIN→BCD conversion transmission).

Error code

Error code	Content
4084H	The data input in the application instruction (s) exceeds the specified range
4085H	The output result of the read application instruction (s) exceeds the device range
4086H	The output result of the write application instruction (d) exceeds the device range

Example



When M0 is set, the BCD value of K8Y0 is converted into BIN and stored in D200.

FLT/BIN integer → binary floating point number

FLT(P)

An instruction to convert a BIN 16-bit integer value into a binary floating point number (real number).

-[FLT (s) (d)]

Content, range and data type

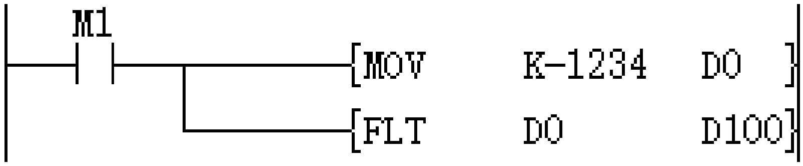
Parameter	Content	Range	Data type	Data type (label)
(s)	The data register number that saves the BIN integer value	-	Signed BIN 16 bit	ANY16
(d)	The data register number that saves the binary floating-point number (real number)	-	Single precision real number	ANYREAL_32

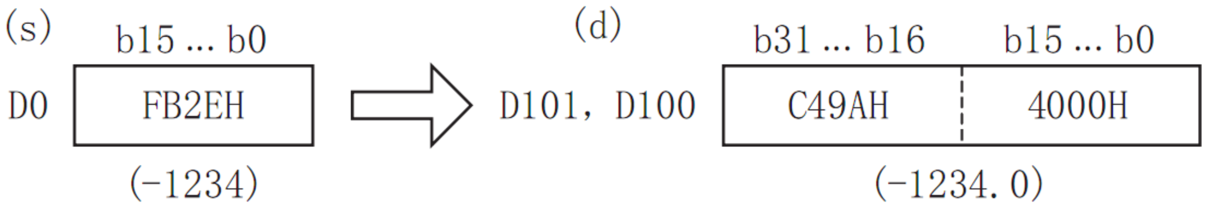
Device used

Instruction	Parameter	Devices													Offset modification	Pulse extension
		KnX	KnY	KnM	KnS	T	C	D	R	SD	LC	HSC	K	H	[D]	XXP
FLT	Parameter 1	●	●	●	●	●	●	●	●	●			●	●	●	●
	Parameter 2					●	●	●	●	●	●	●			●	●

Features

The signed 16-bit data specified in (s) is converted into a binary floating point data and stored in (d)+1, (d).





#Note: In each binary floating point number (real number) operation instruction, the specified K and H values will be automatically converted into a binary floating point number (real number), so there is no need to use the FLT instruction for conversion.

The inverse conversion instruction of this instruction is INT (convert a binary floating point value into a BIN integer).

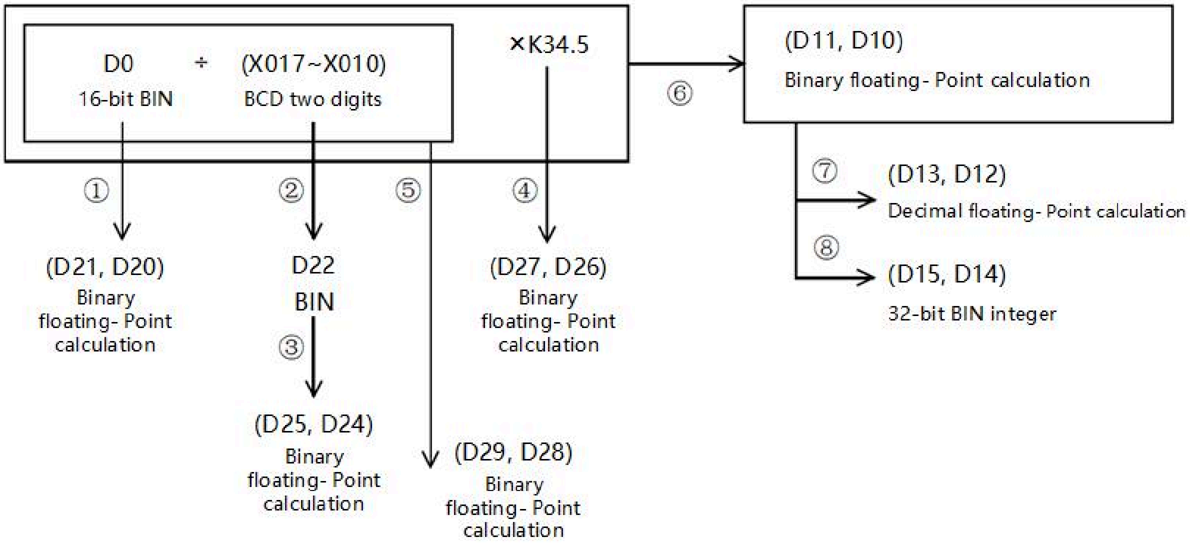
Error code

Error code	Content
4085H	When the device specified in the read application instruction (s) exceeds the corresponding device range
4086H	When the device specified in the write application instruction (d) exceeds the corresponding device range

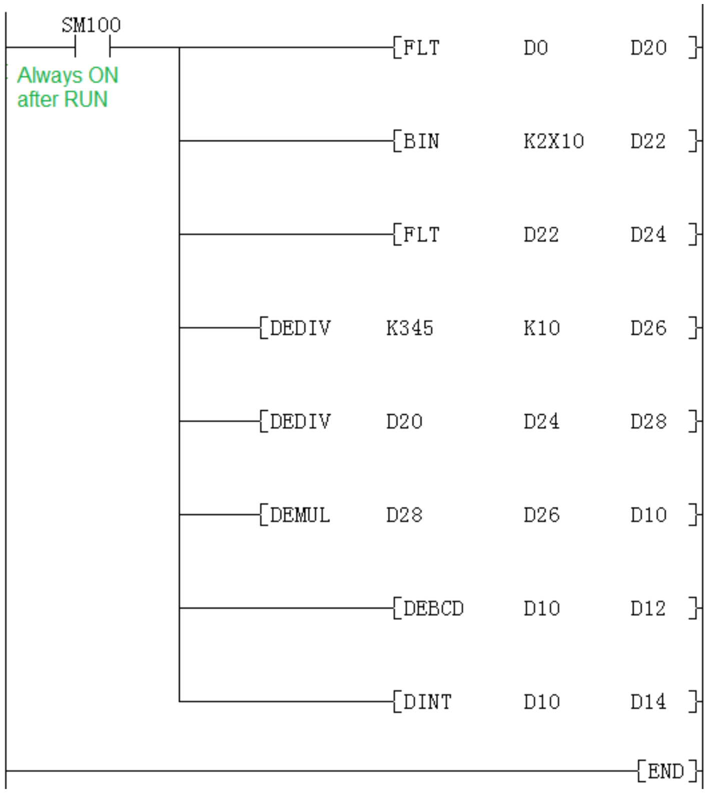
Example

Four arithmetic using binary floating point operations

(1) Calculation example


D0
16-bit BIN
÷
(X017~X010)
BCD two digits
×K34.5
(D11, D10)
Binary floating- Point calculation
⑥
①
②
⑤
④
(D21, D20)
Binary floating- Point calculation
D22
BIN
③
(D25, D24)
Binary floating- Point calculation
(D27, D26)
Binary floating- Point calculation
(D29, D28)
Binary floating- Point calculation
⑦
(D13, D12)
Decimal floating- Point calculation
⑧
(D15, D14)
32-bit BIN integer

(2) Sequence control program



(D0) → (D21, D20)

BIN binary floating point operations

(X17 to X10) → (D22)

BCD BIN

(D22) → (D25, D24)

BIN binary floating point operations

K345÷K10 → (D27, D26)

Binary floating point operations

(D29,D28)×(D27,D26) → (D11,D10)

Binary floating-point number multiplication

(D21,D20)÷(D25,D24) → (D29,D28)

Binary floating-point number division operation →Binary floating-point number operation

(D11,D10) → (D13,D12)

Binary floating-point calculations →

Decimal floating-point calculations monitoring

(D11,D10) → (D13,D12)

Binary floating point operations 32-bit BIN integer

DFLT/BIN integer → binary floating point number

DFLT(P)

An instruction to convert a BIN 32-bit integer value into a binary floating point number (real number).

-[DFLT (s) (d)]

Content, range and data type

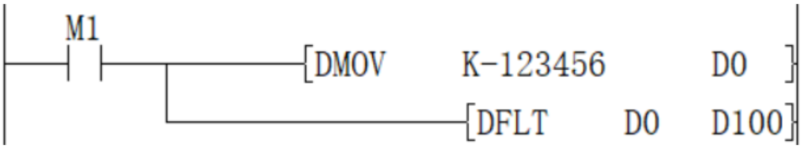
Parameter	Content	Range	Data type	Data type (label)
(s)	The data register number that saves the BIN32 integer value	-	Signed BIN 32 bit	ANY32
(d)	The data register number that saves the binary floating-point number (real number)	-	Single precision real number	ANYREAL_32

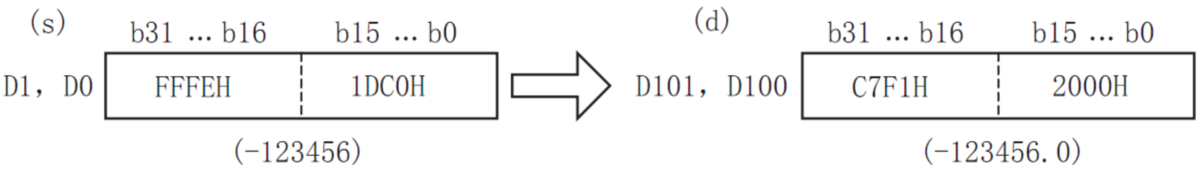
Device used

Instruction	Parameter	Devices			
		KnX	KnY	KnM	KnS
DFLT	Parameter 1	●	●	●	●
	Parameter 2				

Features

Convert the signed BIN 32-bit data specified in (s) to binary floating point data and store them in (d)+1, (d).



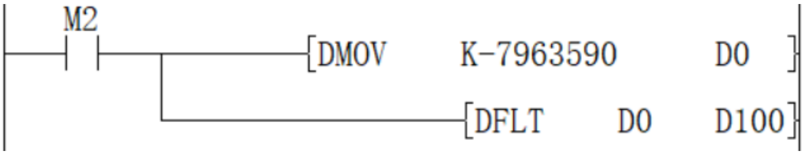


#Note: In each binary floating-point number (real number) operation instruction, the specified K and H values are automatically converted into a binary floating-point number (real number), so there is no need to use the DFLT instruction for conversion. The inverse conversion instruction of this instruction is INT (convert a binary floating point value into a BIN integer).

Error code

Error code	Content
4085H	When the device specified in the read application instruction (s) exceeds the corresponding device range
4086H	When the device specified in the write application instruction (d) exceeds the corresponding device range

Example



When M2=ON, convert the BIN 32-bit integer -7963590 in [D1, D0] into a single-precision floating point number -7963590.0 and store it in the [D101, D100] device.

VAL/ String → BIN 16-bit data conversion

VAL(P)

After converting the character string stored in the device number specified in (s) and later into BIN 16-bit data, store the number of digits in (d1) and store the BIN data in (d2).

[VAL (s) (d1) (d2)]

Content, range and data type

Parameter	Content	Range	Data type	Data type (label)
(s)	The character string converted to BIN data or the start device that stores the character string	-	String	ANYSTRING_SINGLE
(d1)	The start device that stores the number of digits of converted BIN data	-	Signed BIN 16 bit	ANY16_S_ARRAY
(d2)	Start device for storing converted BIN data	-	Signed BIN 16 bit	ANY16_S

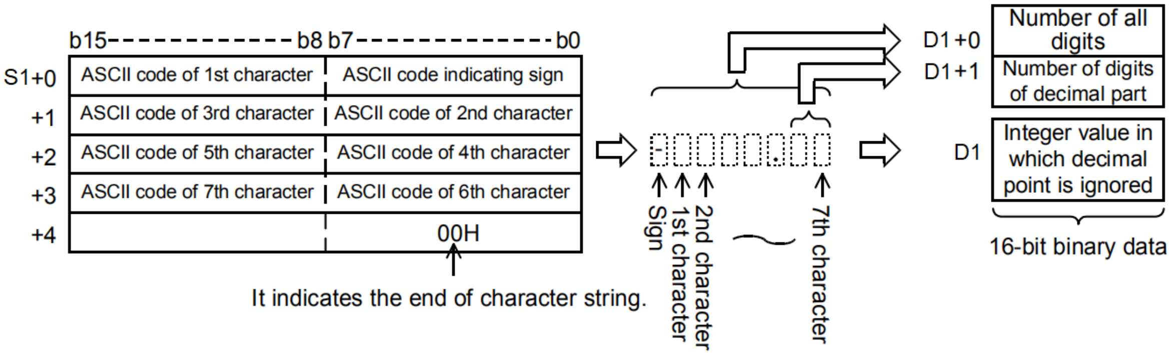
Device used

Instruction	Parameter	Devices								Offset modification	Pulse extension
		KnY	KnM	KnS	T	C	D	R	SD	[D]	XXP
VAL	Parameter 1				●	●	●	●	●	●	●
	Parameter 2	●	●	●	●	●	●	●	●	●	●
	Parameter 3	●	●	●	●	●	●	●	●	●	●

Features

After converting the character string stored in the device number specified in (s) and later into BIN 16-bit data, store the number of digits in (d1) and store the BIN data in (d2). In the conversion from character string to BIN, the data from the device number specified in (s) to the device number storing 00H is treated as a character string.

The total number of digits stored in (d1) stores the number of all characters (including signs and decimal points) representing the value. The number of decimal places stored in (d1)+1 stores the number of characters representing the decimal part after 2EH(.). For the BIN 16-bit data stored in (d2), the character string ignoring the decimal point is converted into a BIN value and stored.



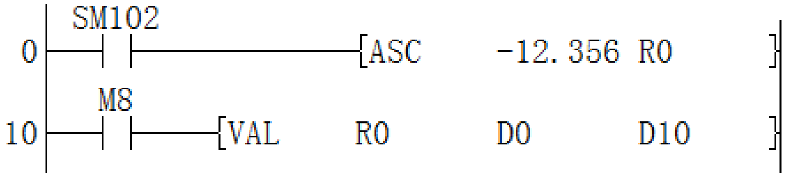
	b15 - - - b8	b7 - - - b0
S1+0	ASCII code of 1st character	ASCII code indicating sign
+1	ASCII code of 3rd character	ASCII code of 2nd character
+2	ASCII code of 5th character	ASCII code of 4th character
+3	ASCII code of 7th character	ASCII code of 6th character
+4		00H

It indicates the end of character string.
Sign
1st character
2nd character
7th character
D1+0
Number of all digits
D1+1
Number of digits of decimal part
D1
Integer value in which decimal point is ignored
16-bit binary data

Error code

Error code	Content
4082H	The character string specified by (s) could not be converted into a numeric value For example: The first character is not a negative sign or a space, space appears in the middle of the number, decimal point appears Except for the first character, there are non-character and decimal Signs For example, 3.4000 is 34000 after removing the decimal point, which is out of range.
4085H	(s) read address exceeds the device range
408AH	When the character number of character string the specified in (s) is other than 2 to 8.
408BH	The maximum range of the device is read when (s) taking character string, but 00H is not found as the end
4086H	When using offset, the offset address of (d) exceeds the device range

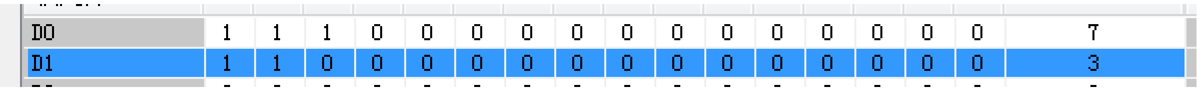
Example



The result obtained above:

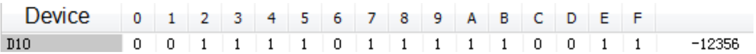
D0 corresponds to str length is 7.

D1 corresponds to a decimal point length of 3.



D0	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	7
D1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	3

D10 corresponds to -12356 ignoring the decimal point.



Device	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F	
D10	0	0	1	1	1	1	0	1	1	1	1	1	0	0	1	1	-12356

DVAL/String → BIN32-bit data conversion

DVAL(P)

After converting the character string stored in the device number specified in (s) into BIN 32-bit data, store the number of digits in (d1) and store the BIN data in (d2).

-[DVAL (s) (d1) (d2)]

Content, range and data type

Parameter	Content	Range	Data type	Data type (label)
(s)	The character string converted to BIN data or the start device that stores the character string	-	String	ANYSTRING_SINGLE
(d1)	The start device that stores the number of digits of converted BIN data	-	Signed BIN 16 bit	ANY16_S_ARRAY
(d2)	Start device for storing converted BIN data	-	Signed BIN 32 bit	ANY32_S

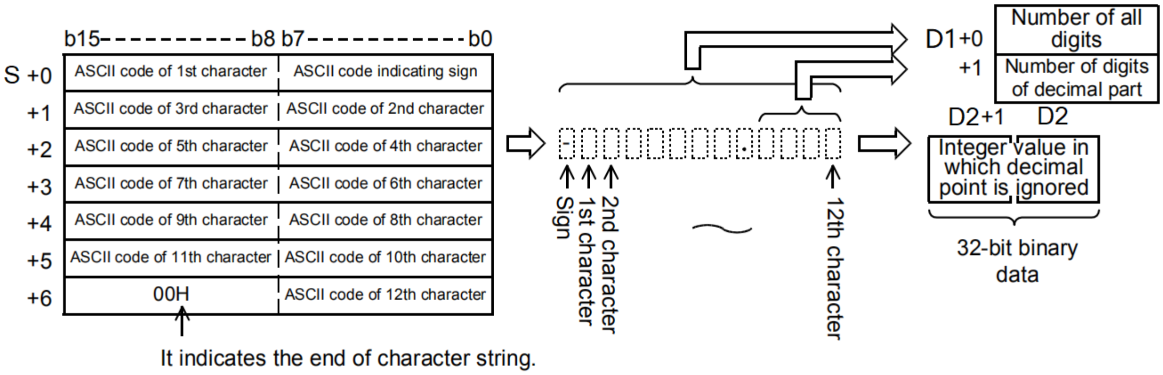
Device used

Instruction	Parameter	Devices										Offset modification	Pulse extension
		KnY	KnM	KnS	T	C	D	R	SD	LC	HSC	[D]	XXP
DVAL	Parameter 1				●	●	●	●	●			●	●
	Parameter 2	●	●	●	●	●	●	●	●			●	●
	Parameter 3	●	●	●	●	●	●	●	●	●	●	●	●

Features

After converting the character string stored in the device number specified in (s) into BIN 32-bit data, store the number of digits in (d1) and store the BIN data in (d2). In the conversion from character string to BIN, the data from the device number specified in (s) to the device number storing 00H is treated as a character string.

The total number of digits stored in (d1) stores the number of all characters (including signs and decimal points) representing the value. The number of decimal places stored in (d1)+1 stores the number of characters representing the decimal part after 2EH(.). For the BIN 32-bit data stored in (d2), the character string ignoring the decimal point is converted into a BIN value and stored.



	b15----------b8	b7-----------b0
S +0	ASCII code of 1st character	ASCII code indicating sign
+1	ASCII code of 3rd character	ASCII code of 2nd character
+2	ASCII code of 5th character	ASCII code of 4th character
+3	ASCII code of 7th character	ASCII code of 6th character
+4	ASCII code of 9th character	ASCII code of 8th character
+5	ASCII code of 11th character	ASCII code of 10th character
+6	00H	ASCII code of 12th character

It indicates the end of character string.

Error code

Error code	Content
4082H	The character string specified by (s) could not be converted into a numeric value.

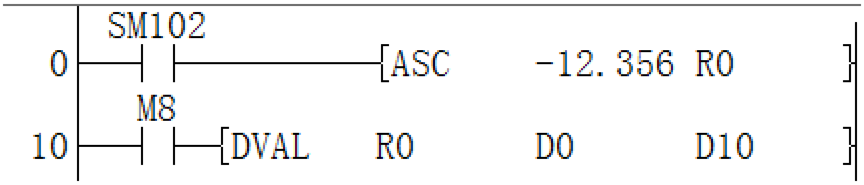
For example:The first character is not a negative sign or a space, space appears in the middle of the number, decimal po

Except for the first character, there are non-character and decimal Signs

For example, 3.000000000 is 3000000000 after removing the decimal point, which is out of range.

4085H	(s) read address exceeds the device range
408AH	When the character number of character string the specified in (s) is other than 2 to 13.
408BH	The maximum range of the device is read when (d1) and (d2) taking character string, but 00H is not found as the end
4086H	When using offset, the offset address of (d) exceeds the device range

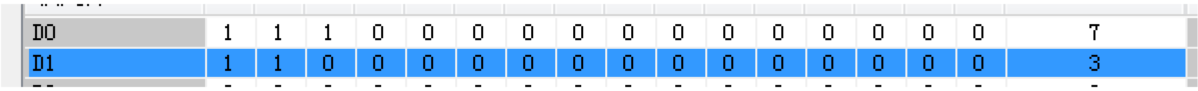
Example



The result obtained above

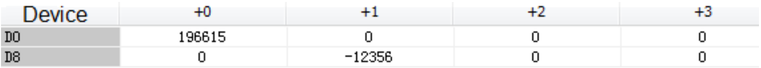
D0 corresponds to str length is 7.

D1 corresponds to a decimal point length of 3.



D0	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	7
D1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	3

D10 corresponds to -12356 ignoring the decimal point



Device	+0	+1	+2	+3
D0	196615	0	0	0
D8	0	-12356	0	0

ASCI/HEX code data →ASCII conversion

ASCI(P)

After the n characters (bits) in the HEX code data specified in (S) are converted into ASCII codes, they are stored after the device number specified in (D).

-[ASCI (S) (D) (N)]

Content, range and data type

Parameter	Content	Range	Data type	Data type (label)
(S)	The start number of the device storing the HEX code to be converted	-	BIN16 bit	ANY16
(D)	The start number of the device storing the converted ASCII code	-	String	ANYSTRING_SINGLE
(N)	The number of characters (digits) of the HEX code to be converted	1to256	BIN16 bit	ANY16_U

Device used

Instruction	Parameter	Devices												Offset modification	Pulse extension
		KnX	KnY	KnM	KnS	T	C	D	R	SD	K	H	E	[D]	XXP
ASCI	Parameter 1	●	●	●	●	●	●	●	●	●	●	●		●	●
	Parameter 2		●	●	●	●	●	●	●	●				●	●

Parameter 3	●	●	●	●	●			●

Features

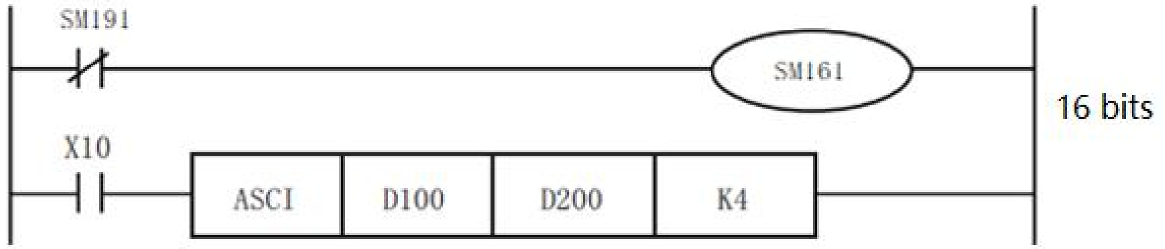
The number of characters (bits) specified by (N) in the HEX code data specified in (S) is converted into ASCII code and stored in the device number specified in (D) or later.

ASCI(P) instruction uses 16-bit mode and 8-bit mode when converting. For the operation of each mode, please refer to the following content.

(1) 16-bit conversion mode (when SM8161=OFF)

Convert the digits of the HEX code after the device specified in (S) into ASCII, and transfer to the upper and lower 8 bits (bytes) of the device specified in (D). When using in 16-bit conversion mode, SM161 should always be turned OFF.

In the case of the following program, perform the conversion as shown below.



Devices after (s): D100=OABCH, D101=1234H, D102=5678H

Specify the number of bits (characters) and the conversion result

(N)	K1	K2	K3	K4	K5	K6	K7	K8	K9
(D)									
Under D200	C	B	C	0	4	3	2	1	8
D200 on		C	B	C	0	4	3	2	1
Under D201			C	B	C	0	4	3	2
D201 on				C	B	C	0	4	3
Under D202					C	B	C	0	4
D202 on		Unchanged				C	B	C	0
Under D203							C	B	C
D203 on								C	B
Under D204									C

Bit structure in the case of =K4

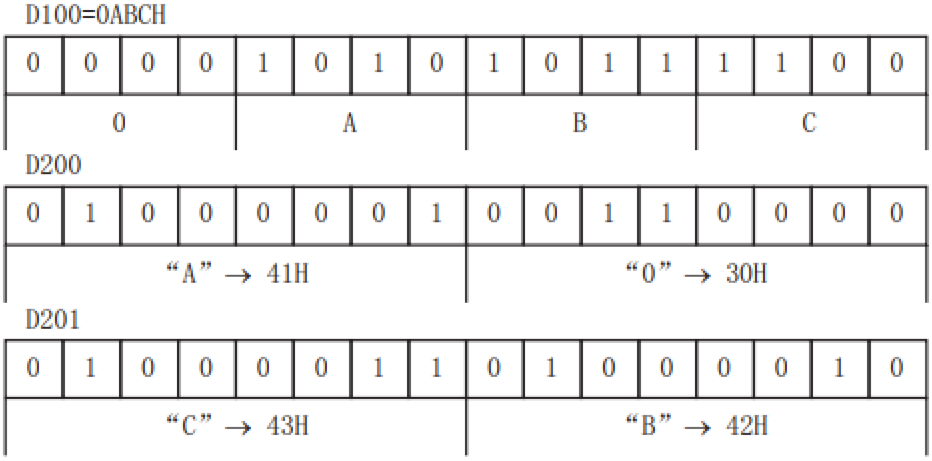
ASCII code

"0"=30H "1"=32H "5"=35H

"A"=41H "2"=32H "6"=36H

"B"=42H "3"=33H "7"=37H

"C"=43H "4"=34H "8"=38H



D100=0ABCH

0	0	0	0	1	0	1	0	1	0	1	1	1	1	0	0
0				A				B				C			

D200

0	1	0	0	0	0	0	1	0	0	1	1	0	0	0	0
"A" → 41H								"0" → 30H							

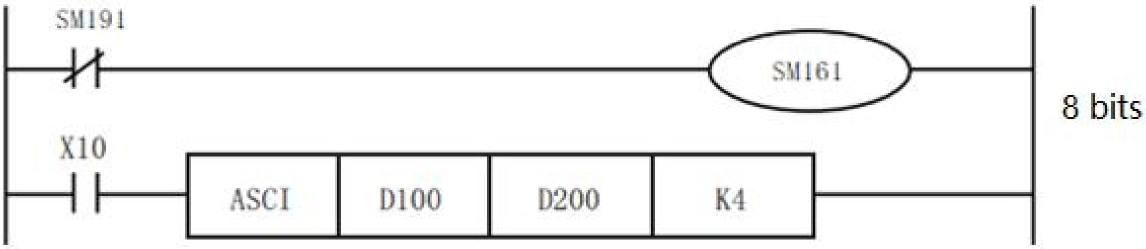
D201

0	1	0	0	0	0	1	1	0	1	0	0	0	0	1	0
"C" → 43H								"B" → 42H							

(2) 8-bit conversion mode (when SM161=ON)

Convert the digits of the HEX code after the device specified in (S) into ASCII, and transfer to the lower 8 bits (bytes) of the device specified in (D). When using in 8-bit conversion mode, SM161 should always be set to ON for use.

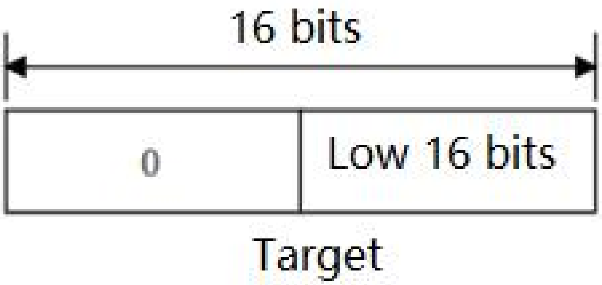
In the case of the following program, perform the conversion as shown below.



Devices after (s1): D100=OABCH, D101=1234H, D102=5678H

If SM161 is set to ON, it will become 8-bit mode,

Perform conversion processing as shown below.



(N) (D)	K1	K2	K3	K4	K5	K6	K7	K8	K9
D200	C	B	C	0	4	3	2	1	8

D201	C	B	C	0	4	3	2	1
D202		C	B	C	0	4	3	2
D203			C	B	C	0	4	3
D204				C	B	C	0	4
D205	Unchanged				C	B	C	0
D206						C	B	C
D207							C	B
D208								C

Bit structure in the case of (N)=K2

ASCII

"0"=30H "1"=31H "5"=35H

"A"=41H "2"=32H "6"=36H

"B"=42H "3"=33H "7"=37H

"C"=43H "4"=34H "8"=38H

D100=0ABCH

0	0	0	0	1	0	1	0	1	0	1	1	1	1	0	0
0				A				B				C			

D200=B ASCII =42H

0	0	0	0	0	0	0	0	0	1	0	0	0	0	1	0
								4				2			

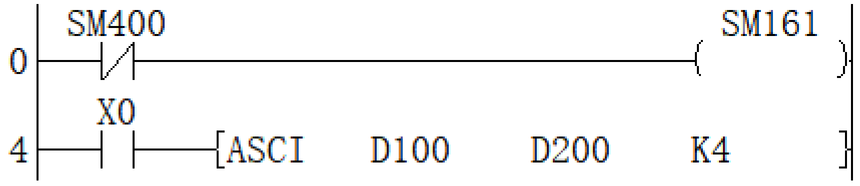
D201=C ASCII =34H

0	0	0	0	0	0	0	0	0	1	0	0	0	0	1	1
								4				3			

Error code

Error code	**Content**
4085H	When the specified device range is read to exceed the corresponding device range
4086H	When the specified device range is written to exceed the corresponding device range
4084H	When the value specified in (N) exceeds the range of 1 to 256

Example



• 16-bit conversion mode (when SM161=OFF)

Convert the digits of the HEX code after the device specified in d100 into ASCII, and transfer to the upper and lower 8 bits (bytes) of the device specified in d200. When using in 16-bit conversion mode, SM161 should always be turned OFF.

HEX/ASCII → HEX code data conversion

HEX(P)

After the device number specified in (s), the ASCII data stored in the number of characters specified in is converted to HEX code, and then stored in the device number specified in (d) or later.

-[HEX (S) (D) (N)]

Content, range and data type

Parameter	Content	Range	Data type	Data type (label)
(S)	The start device that stores the ASCII data converted to HEX code	-	String	ANYSTRING_SINGLE
(D)	The start device that stores converted HEX code	-	BIN16 bit	ANY16
(N)	Number of characters (bytes) of converted ASCII data	1 to 256	BIN16 bit	ANY16_U

Device used

Instruction	Parameter	Devices											Offset modification	Pulse extension
		KnX	KnY	KnM	KnS	T	C	D	R	SD	K	H	[D]	XXP
HEX	Parameter 1	●	●	●	●	●	●	●	●	●	●	●	●	●
	Parameter 2		●	●	●	●	●	●	●	●			●	●
	Parameter 3							●	●	●	●	●		●

Features

• After the device number specified in (S), the ASCII data stored in the number of characters specified in (N) is converted to HEX code, and then stored in the device number specified in (D) or later. The HEX(P) instruction uses 16-bit conversion mode and 8-bit conversion mode when converting. For the operation of each mode, please refer to the following content.

(1) 16-bit conversion mode (when SM161=OFF)

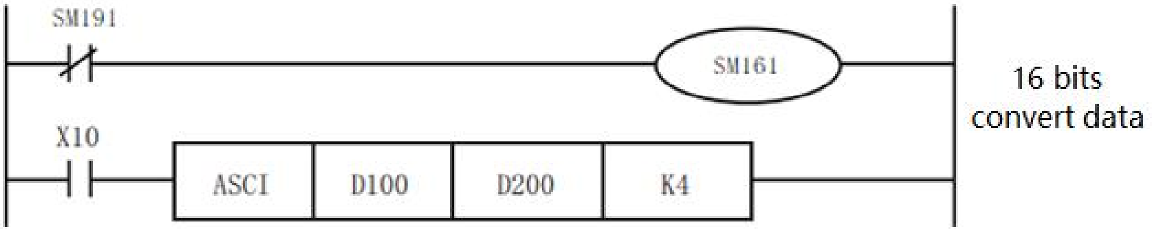
After converting the ASCII data stored in the upper and lower 8 digits (bytes) of the device specified in (S) into HEX code, it transmits every 4 digits to the device specified in (D). The number of characters to be converted is specified in (N).

SM161 is shared with ASC, ASCI, BCC, CCD and CRC instructions. When using in 16-bit conversion mode, please always set SM161 to OFF.

SM161 is cleared when RUN→STOP.

In addition, it is necessary to store the ASCII data in the 16-bit conversion mode in the upper 8 bits of the device specified in (S).

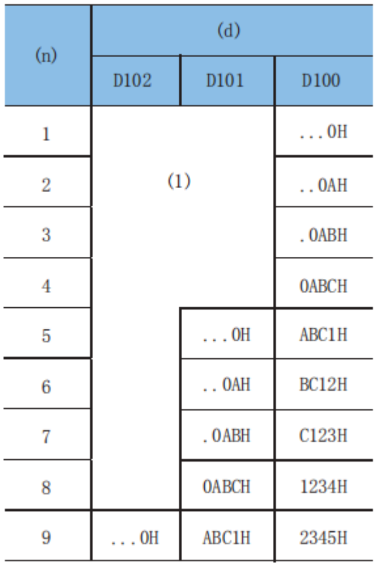
In the following program, the conversion will be performed in the following manner.



Transform the source data

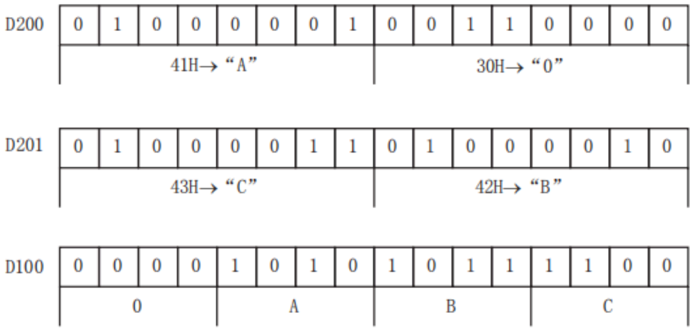
(S)	ASCII data	HEX conversion
Under D200	30H	0
D200 on	41H	A
Under D201	42H	B
D201 on	43H	C
Under D202	31H	1
D202 on	32H	2
Under D203	33H	3
D203 on	34H	4
Under D204	35H	5

Bit structure in the case of (N)=K4



(n)	(d)		
	D102	D101	D100
1	(1)		...0H
2			..0AH
3			.0ABH
4			0ABCH
5		...0H	ABC1H
6		..0AH	BC12H
7		.0ABH	C123H
8		0ABCH	1234H
9	...0H	ABC1H	2345H

(1)：不变化



D200	0	1	0	0	0	0	0	1	0	0	1	1	0	0	0	0
	41H→"A"								30H→"0"							
D201	0	1	0	0	0	0	1	1	0	1	0	0	0	0	1	0
	43H→"C"								42H→"B"							
D100	0	0	0	0	1	0	1	0	1	0	1	1	1	1	0	0
	0				A				B				C			

The number of characters specified and the conversion result becomes 0.

(N)		1	2	3	4	5	6	7	8	9
(D)	**D102**	Unchanged								...OH
	D101					...OH	..OAH	.OABH	OABCH	ABC1H

D100	...OH	..OAH	.OABH	OABCH	ABC1H	BC12H	C123H	1234H	2345H

(2) 8-bit conversion mode (when SM161=ON)

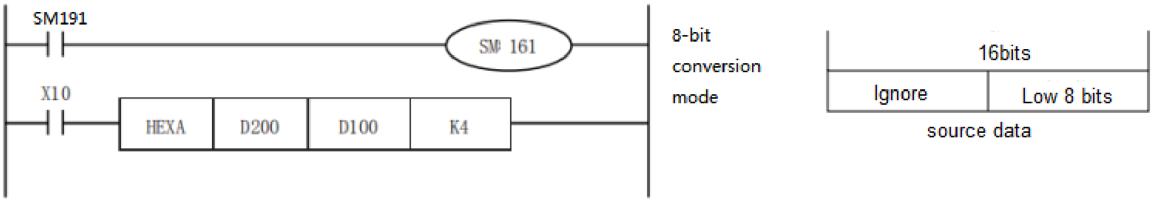
After converting the ASCII data stored in the lower 8 digits of the device specified in (S) into HEX code, it will be transmitted to the device specified in (D) every 4 digits.

The number of characters to be converted is specified in (N).

SM161 is shared with ASC, ASCI, BCC, CCD and CRC instructions. When using in 8-bit conversion mode, please always turn on SM161.

SM161 is cleared when RUN→STOP.

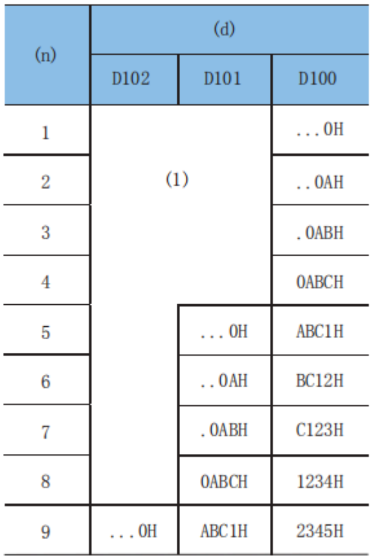
In the following program, the conversion will be performed in the following manner.



Transform the source data

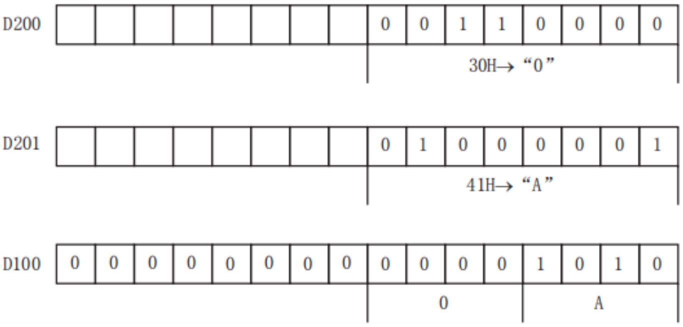
(S)	ASCII data	HEX conversion
D200	30H	0
D201	41H	A
D202	42H	B
D203	43H	C
D204	31H	1
D205	32H	2
D206	33H	3
D207	34H	4
D208	35H	5

Bit structure in the case of (N)=K2



(n)	(d)		
	D102	D101	D100
1	(1)		...0H
2			..0AH
3			.0ABH
4			0ABCH
5		...0H	ABC1H
6		..0AH	BC12H
7		.0ABH	C123H
8		0ABCH	1234H
9	...0H	ABC1H	2345H

(1)：不变化



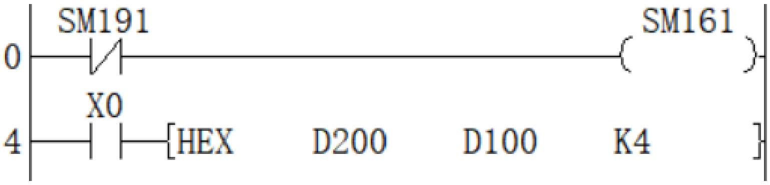
The number of characters specified and the conversion result becomes 0.

(N)		1	2	3	4	5	6	7	8	9
(D)	**D102**	Unchanged								...OH
	D101					...OH	..OAH	.OABH	OABCH	ABC1H
	D100	...OH	..OAH	.OABH	OABCH	ABC1H	BC12H	C123H	1234H	2345H

Error code

Error code	**Content**
4084H	When the value specified in (N) exceeds the range.
	When ASCII codes other than 30H to 39H and 41H to 46H are set in (s).
4085H	When the specified device range is read to exceed the corresponding device range
4086H	When the specified device range is written to exceed the corresponding device range

Example



After converting the ASCII data stored in the upper and lower 8 digits (bytes) of the device specified in (S) into HEX code, it transmits every 4 digits to the device specified in (D). The number of characters to be converted is specified in (N).

SM161 is shared with ASC, ASCI, BCC, CCD and CRC instructions. When using in 16-bit conversion mode, please always set SM161 to OFF.

CCD/Check code

CCD(P)

Calculate the horizontal parity value and the sum check value of the error checking method used in communication and the like. In addition to these error checking methods, there are CRC (Cyclic

Redundancy Check). To calculate the CRC value, use the CRC(P) command.

-[CCD (S) (D) (N)]

Content, range and data type

Parameter	Content	Range	Data type	Data type (label)
(S)	The start number of object device	-	BIN16 bit	ANY16
(D)	The start number of the storage destination device of the calculated data	-	BIN16 bit	ANY16_ARRAY (number of elements: 2)
(N)	Number of data	1 to 256	BIN16 bit	ANY16_U

Device used

Instruction	Parameter	Devices											Offset modification	Pulse extension
		KnX	KnY	KnM	KnS	T	C	D	R	SD	K	H	[D]	XXP
CCD	Parameter 1	●	●	●	●	●	●	●	●	●			●	●
	Parameter 2		●	●	●	●	●	●	●	●			●	●
	Parameter 3							●	●	●	●	●		●

Features

Calculate the addition data and horizontal parity data of the data stored in (S) to (S)+(N)-1, and store the addition data in (D), horizontal parity

The data is stored in (D)+1. The modes used by this instruction in calculation are 16-bit mode and 8-bit mode. For the operation of each mode, please refer to the following content.

(1) 16-bit conversion mode (when SM161=OFF)

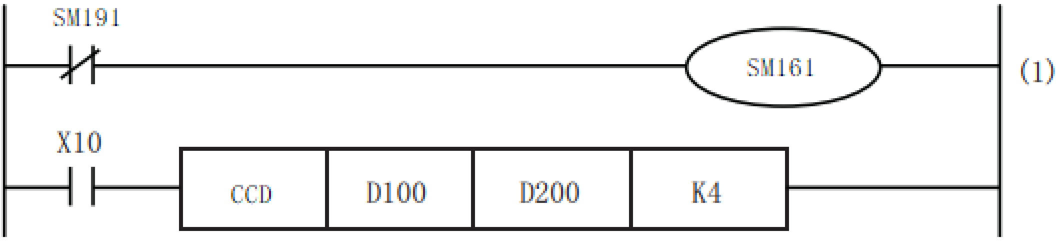
Regarding the data at point (N) starting with (S), the addition data and horizontal parity data of the high and low 8-bit data are stored in the Devicess (D) and (D)+1.

SM161 is shared with ASC, ASCI, BCC, CCD and CRC instructions. When using in 16 bits, always set to OFF for use.

SM161 is cleared when RUN→STOP.

In the case of the following program, perform the conversion as shown below.

16-bit conversion mode



If the number of 1 is odd, the horizontal parity is 1

If the number of 1 is even, the horizontal parity is 0

1091 at BCD

Horizontal parity

D0	0	0	0	0	0	1	0	0	0	1	0	0	0	0	1	1	⇐ BCD时为1091
D1	0	0	0	0	0	0	0	0	1	0	0	0	0	1	0	1	⇐ 水平奇偶校验

(2) 8-bit conversion mode (when SM161=ON)

Regarding (S) as the starting point (N) data (lower 8 bits only), its addition data and horizontal parity data are stored in the devices (D) and (D)+1.

SM161 is shared with ASC, ASCI, BCC, CCD and CRC instructions. If it is used in 8 bits, it should always be set to ON for use.

SM161 is cleared when RUN→STOP.

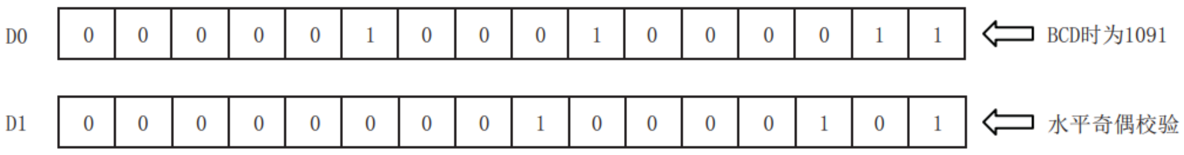
In the case of the following program, perform the conversion as shown below.

If the number of 1 is odd, the horizontal parity is 1

If the number of 1 is even, the horizontal parity is 0

1091 at BCD

Horizontal parity

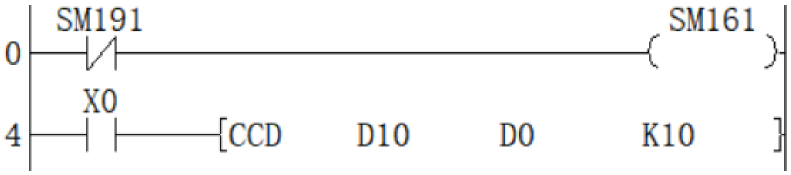


D0	0	0	0	0	0	1	0	0	0	1	0	0	0	0	1	1	BCD时为1091
D1	0	0	0	0	0	0	0	0	1	0	0	0	0	1	0	1	水平奇偶校验

Error code

Error code	Content
4084H	When the value specified in (N) exceed the range of 1 to 256.
4085H	When the specified device range is read to exceed the corresponding device range
4086H	When the specified device range is written to exceed the corresponding device range

Example



Regarding D10 as the initial 10-point data, the addition data and horizontal parity data of the high and low 8-bit data are stored in the Devicess of D0 and D0+1.

SM161 is shared with ASC, ASCI, BCC, CCD and CRC instructions. When using in 16 bits, always set to OFF for use.

GBIN/Gray code → BIN 16-bit data conversion

GBIN(P)

Convert the BIN 16-bit Gray code data stored in the device specified in (s) into BIN 16-bit data, and store it in the device specified in (d).

-[GBIN (s) (d)]

Content, range and data type

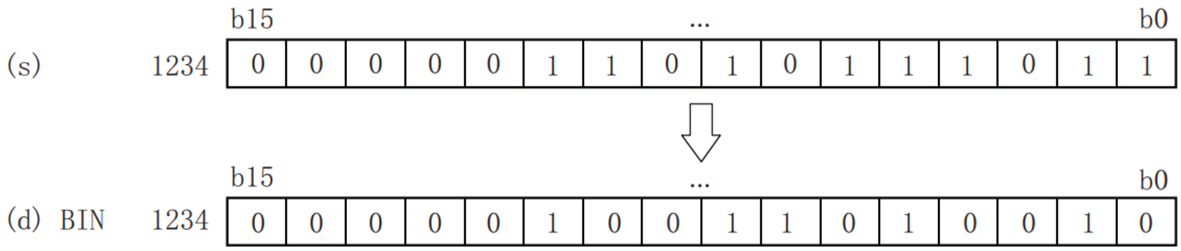
Parameter	Content	Range	Data type	Data type (label)
(s)	Gray code data or the start device that stores Gray code	0 to 32767	BIN16 bit	ANY16_S
(d)	The start device that stores the converted BIN data	-	BIN16 bit	ANY16_S

Device used

Instruction	Parameter	Devices											Offset modification	Pulse extension
		KnX	KnY	KnM	KnS	T	C	D	R	SD	K	H	[D]	XXP
GBIN	Parameter 1	●	●	●	●	●	●	●	●	●	●	●	●	●
	Parameter 2		●	●	●	●	●	●	●	●			●	●

Features

Convert the BIN 16-bit Gray code data stored in the device specified in (s) into BIN 16-bit data, and store it in the device specified in (d).

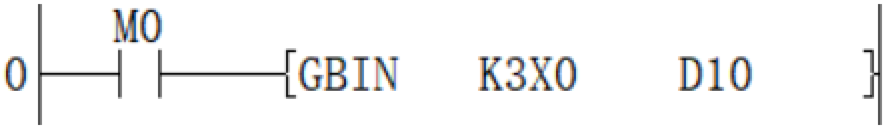


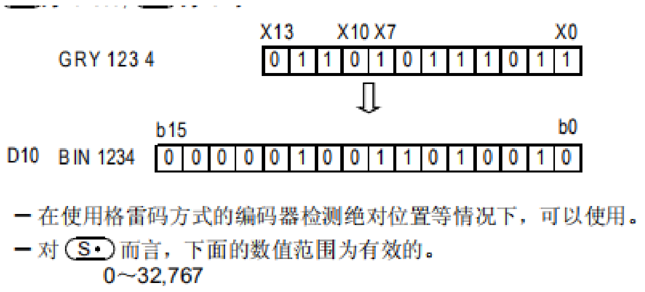
GRY→BIN Mathematical Algorithm: Starting from the second bit from the left, XOR each bit with the decoded value of the left bit as the decoded value of the bit (the leftmost bit remains unchanged).

Error code

Error code	Content
4084H	When the value specified in (s) exceeds the range
4085H	When the specified device range is read to exceed the range of the corresponding device
4086H	When the specified device range is written to exceed the range of the corresponding device

Example





一在使用格雷码方式的编码器检测绝对位置等情况下，可以使用。

一对 (S•) 而言，下面的数值范围为有效的。

It could be used when the encoder of Gray code method is used to detect the absolute position.

For S, the numerical are valid in the range of 0 to 32767.

DGBIN/Gray code → BIN32-bit data conversion

DGBIN(P)

Convert the BIN32-bit Gray code data stored in the device specified in (s) to BIN 32-bit data and store it in the device specified in (d).

Content, range and data type

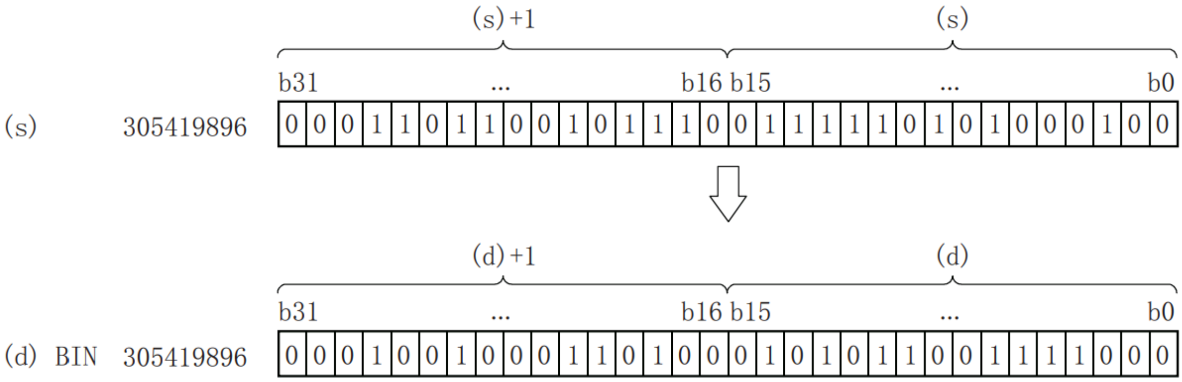
Parameter	Content	Range	Data type	Data type (label)
(s)	Gray code data or the start device that stores Gray code	0 to 2147483647	BIN32 bit	ANY32_S
(d)	The start device that stores converted BIN data	-	BIN32 bit	ANY32_S

Device used

Instruction	Parameter	Devices													Offset modification	Pulse extension
		KnX	KnY	KnM	KnS	T	C	D	R	SD	LC	HSC	K	H	[D]	XXP
DGBIN	Parameter 1	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●
	Parameter 2		●	●	●	●	●	●	●	●	●	●			●	●

Features

Convert the BIN32-bit Gray code data stored in the device specified in (s) into BIN 32-bit data, and store it in the device specified in (d).



(s)+1: high 16 bits

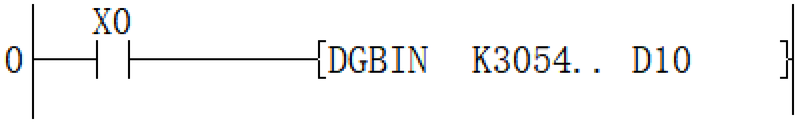
(s): low 16 bits

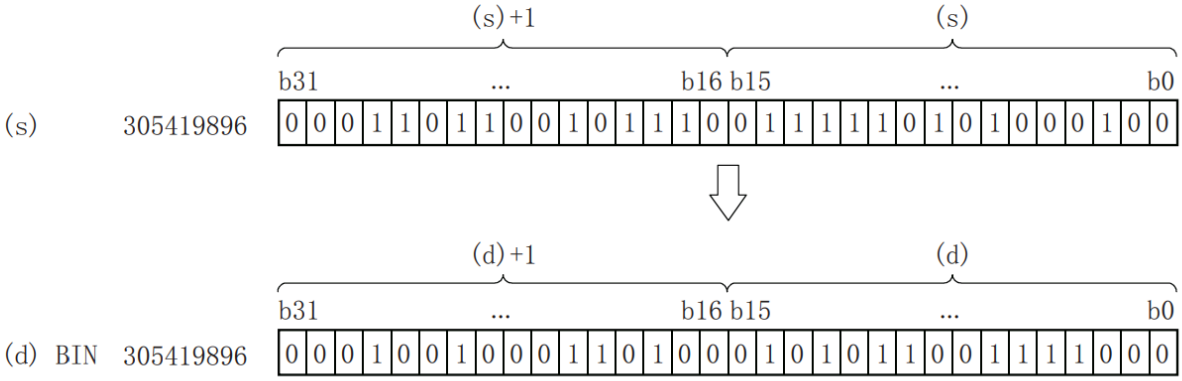
GRY→BIN Mathematical Algorithm: Starting from the second bit from the left, XOR each bit with the decoded value of the left bit as the decoded value of the bit (the leftmost bit remains unchanged).

Error code

Error code	Content
4084H	When the value specified in (s) exceeds the range
4085H	When the specified device range is read to exceed the corresponding device range
4086H	When the specified device range is written to exceed the corresponding device range

Example





(s) 305419896: 0 0 0 1 1 0 1 1 0 0 1 0 1 1 1 0 0 1 1 1 1 1 0 1 0 1 0 0 0 1 0 0

(d) BIN 305419896: 0 0 0 1 0 0 1 0 0 0 1 1 0 1 0 0 0 1 0 1 0 1 1 0 0 1 1 1 1 0 0 0

GRY/BIN 16-bit data → Gray code conversion

GRY(P)

After converting the BIN 16-bit data of the device specified in (s) to BIN 16-bit Gray code data, it is stored in the device specified in (d).

-[GRY (s) (d)]

Content, range and data type

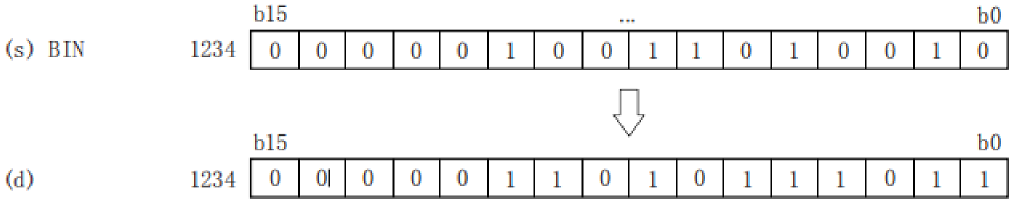
Parameter	Content	Range	Data type	Data type (label)
(s)	BIN data or the start device that stores BIN data	0 to 32767	BIN16 bit	ANY16_S
(d)	The start device that stores the converted Gray code	-	BIN16 bit	ANY16_S

Device used

Instruction	Parameter	Devices											Offset modification	Pulse extension
		KnX	KnY	KnM	KnS	T	C	D	R	SD	K	H	[D]	XXP
GRY	Parameter 1	●	●	●	●	●	●	●	●	●	●	●	●	●
	Parameter 2		●	●	●	●	●	●	●	●			●	●

Features

Convert the BIN 16-bit data specified in (s) into BIN 16-bit Gray code, and store it in the device specified in (d).



(s) BIN 1234: 0 0 0 0 0 1 0 0 1 1 0 1 0 0 1 0

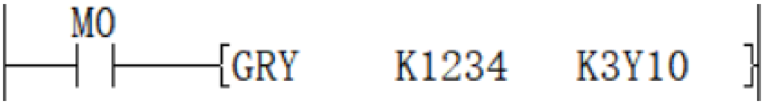
(d) 1234: 0 0| 0 0 0 1 1 0 1 0 1 1 1 0 1 1

BIN→GRY Mathematical Algorithm: Starting from the rightmost bit, XOR each bit with the left bit as the value corresponding to the GRY bit, and the leftmost bit remains unchanged (equivalent to 0 on the left) .

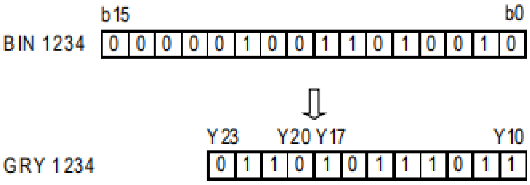
Error code

Error code	Content
4084H	When the value specified in (s) exceeds the range
4085H	When the specified device range is read to exceed the corresponding device range
4086H	When the specified device range is written to exceed the corresponding device range

Example



As shown in the above Circuit program:



一对 S· 而言，下面的数值范围为有效的。
0~32,767

For S, the range of 0 to 32767 is valid.

DGRY/BIN 32-bit data → Gray code conversion

DGRY(P)

After converting the BIN 16-bit data of the device specified in (s) to BIN 16-bit Gray code data, it is stored in the device specified in (d).

-[GRY (s) (d)]

Content, range and data type

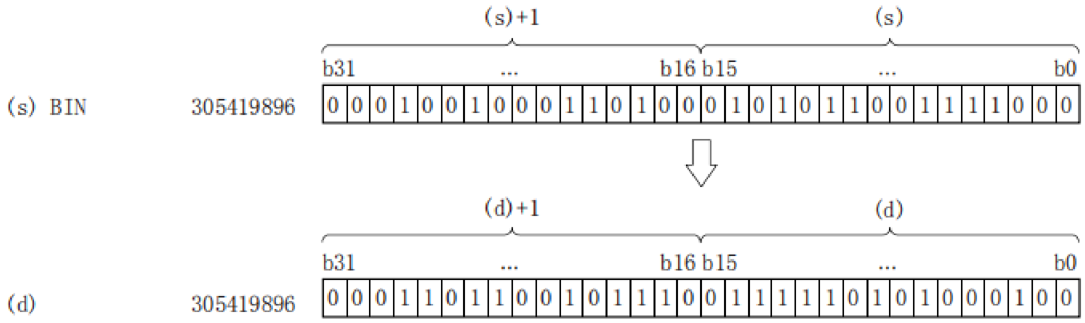
Parameter	Content	Range	Data type	Data type (label)
(s)	BIN data or the start device that stores BIN data	0 to 2147483647	BIN32 bit	ANY32_S
(d)	The start device that stores the converted Gray code	-	BIN32 bit	ANY32_S

Device used

Instruction	Parameter	Devices													Offset modification	Pulse extension
		KnX	KnY	KnM	KnS	T	C	D	R	SD	LC	HSC	K	H	[D]	XXP
DGRY	Parameter 1	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●
	Parameter 2		●	●	●	●	●	●	●	●	●	●			●	●

Features

Convert the BIN32-bit data specified in (s) into BIN32-bit Gray code and store it in the device specified in (d)



(s)+1: high 16 bits

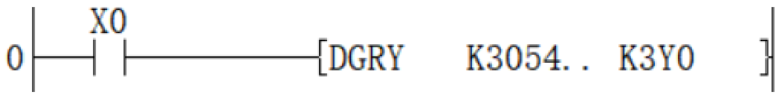
(s): low 16 bits

BIN→GRY Mathematical Algorithm: Starting from the rightmost bit, XOR each bit with the left bit as the value corresponding to the GRY bit, and the leftmost bit remains unchanged (equivalent to 0 on the left) .

Error code

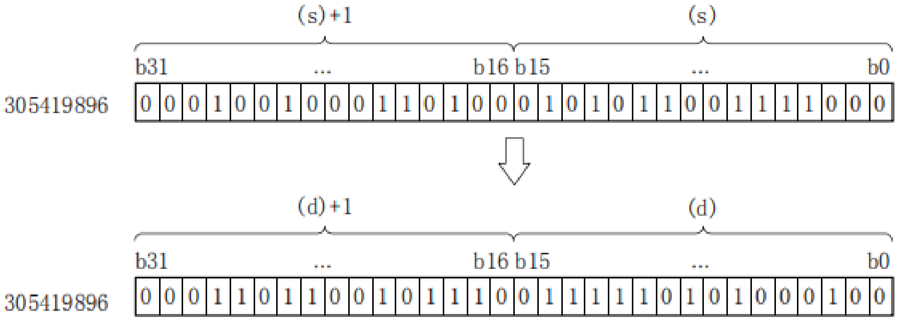
Error code	Content
4084H	When the value specified in (s) exceeds the range
4085H	When the specified device range is read to exceed the corresponding device range
4086H	When the specified device range is written to exceed the corresponding device range

Example



As shown in the above Circuit program:

[illegible]



DPRUN/Otal digit transmission (32-bit data)

DPRUN(P)

After processing the device numbers of (s) and (d) with specified digits as octal numbers, transfer the data.

-[PRUN (s) (d)]

Content, range and data type

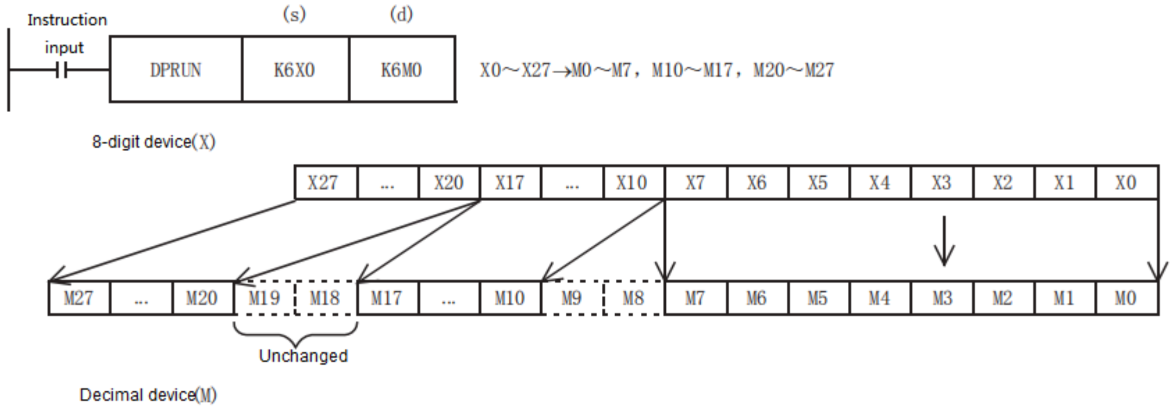
Parameter	Content	Range	Data type	Data type (label)
(s)	Digit specification*1	-	BIN32 bit	ANY32
(d)	Transfer destination device number*1	-	BIN32 bit	ANY32

Device used

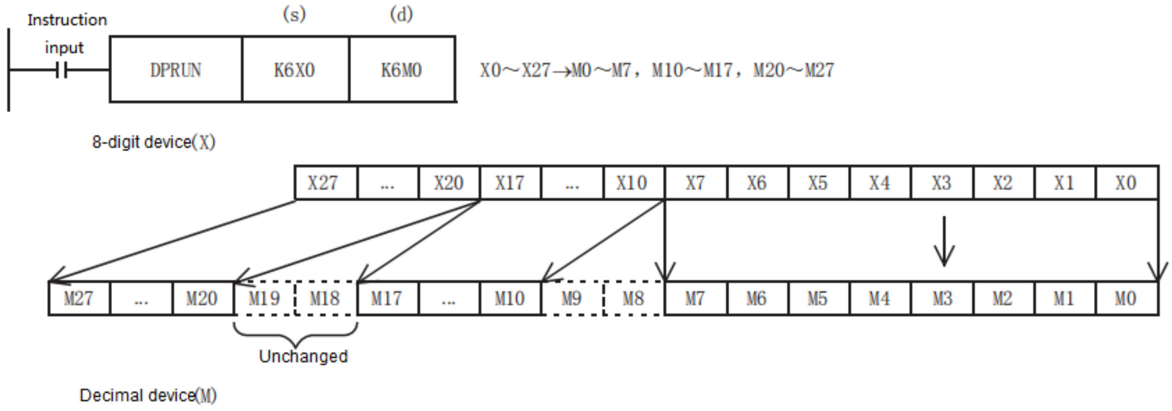
Instruction	Parameter	Devices			Offset modification	Pulse extension
		KnX	KnY	KnM	[D]	XXP
DPRUN	Parameter 1	●		●	●	●
	Parameter 2		●	●	●	●

Features

- Octal digit device to decimal digit device



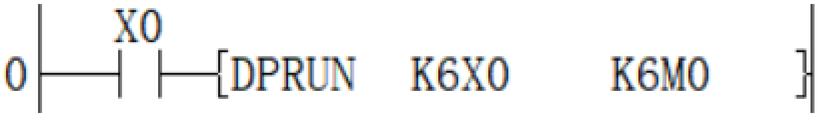
- Decimal digit device → octal digit device



Error code

Error code	Content
4085H	When the specified device range is read to exceed the corresponding device range
4086H	When the specified device range is written to exceed the corresponding device range

Example



As shown in the above Circuit program:

X0 to X27 take the value of octal digits and pass them to the Devices corresponding to M.

Floating point instructions

DACOS/Single precision real number COS-1 operation

DACOS(P)

After calculating the COS $^{-1}$ (arc cosine) value of the angle specified in (s), the calculation result is stored in the device number specified in (d).

-[DACOS (s) (d)]

Content, range and data type

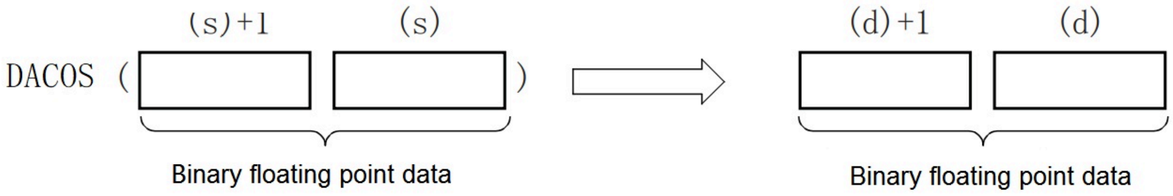
Parameter	Content	Range	Data type	Data type (label)
(s)	The angle data for COS $^{-1}$ (arc cosine) calculation or the start device number that stores the angle data	0, 2^{-126} ≤|(s)|<1	Single precision real number	ANYREAL_32
(d)	The start device number that stores operation result	0 to π	Single precision real number	ANYREAL_32

Device used

Instruction	Parameter	Devices								Offset modification	Pulse extension
		T	C	D	R	SD	LC	HSC	E	[D]	XXP
DACOS	Parameter 1	●	●	●	●	●	●	●	●	●	●
	Parameter 2	●	●	●	●	●	●	●		●	●

Features

After calculating the COS $^{-1}$ (arc cosine) value of the angle specified in (s), the calculation result is stored in the device number specified in (d).



The COS value specified in (s) can be set within the range of -1.0 to 1.0.

The angle (calculation result) stored in (d) stores the value from 0 to π in radians.

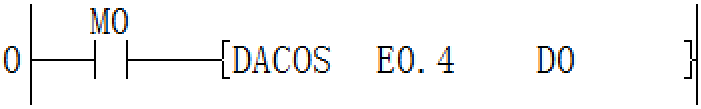
Related device are as follows:

Devices	Name	Content	
		Condition	**Operation**
SM153	Zero	The operation result is zero	The zero flag (SM153) turns ON.
SM152	Borrow	The absolutevalue of operation result<2 $^{-126}$	The value of (d) becomes the minimum value of 32-bit real numbers (2 $^{-126}$), and the borrow flag (SM152) turns ON.

Error code

Error code	Content
4085H	The write address in (s) exceeds the device range
4086H	The write address in (d) exceeds the device range
4084H	When the content of the device specified by (s) is an irregular number, a non-number, ±∞ and exceeds -1.0 to 1.0

Example



Calculate the arc cosine value of 0.4 and the result is 1.159279.

D0 1.159279

DASIN/Single precision real number SIN^{-1} operation

DASIN(P)

After calculating the SIN -1 (arc sine) value of the angle specified in (s), the calculation result is stored in the device number specified in (d).

-[DASIN (s) (d)]

Content, range and data type

Parameter	Content	Range	Data type	Data type (label)
(s)	The angle data for SIN $^{-1}$ (arcsine) calculation or the start device number that stores the angle data	0, 2^{-126} ≤|(s)|<1	Single precision real number	ANYREAL_32
(d)	The start device number that stores operation result	-π/2 to π/2	Single precision real number	ANYREAL_32

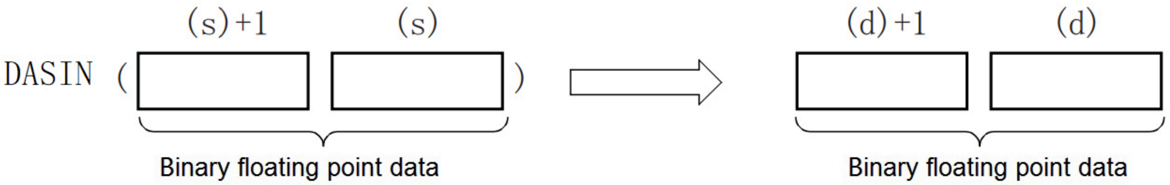
Device used

Instruction	Parameter	Devices								Offset modification	Pulse extension
		T	C	D	R	SD	LC	HSC	E	[D]	XXP

DASIN	Parameter 1	●	●	●	●	●	●	●	●	●	●
	Parameter 2	●	●	●	●	●	●	●		●	●

Features

• After calculating the SIN-1 (arc sine) value of the angle specified in (s), the calculation result is stored in the device number specified in (d).



The SIN^{-1} value specified in (s) can be set within the range of -1.0 to 1.0.

The angle (calculation result) stored in (d) is stored in the unit of radians (-π/2) to (π/2).

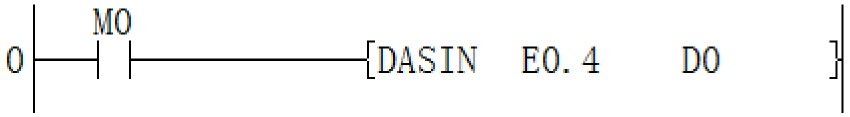
• The related devices are as follows.

Devices	Name	Content	
		Condition	Operation
SM153	Zero	The operation result is zero	The zero flag (SM153) turns ON.
SM152	Borrow	The absolute value of operation result<2^{-126}	The value of (d) becomes the minimum value of 32-bit real numbers (2^{-126}), and the borrow flag (SM152) turns ON.

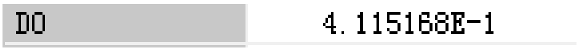
Error code

Error code	Content
4085H	The write address in (s) exceeds the device range
4086H	The write address in (d) exceeds the device range
4084H	When the content of the device specified by (s) is an irregular number, a non-number, ±∞ and exceeds -1.0 to 1.0

Example



Calculate the arc sine of 0.4 and the result is 0.4115168.



```
D0          4.115168E-1
```

DATAN/Single precision real number TAN^{-1} operation

DATAN(P)

After calculating the TAN -1 (arctangent) value of the angle specified in (s), the calculation result is stored in the device number specified in (d).

-[DATAN (s) (d)]

Content, range and data type

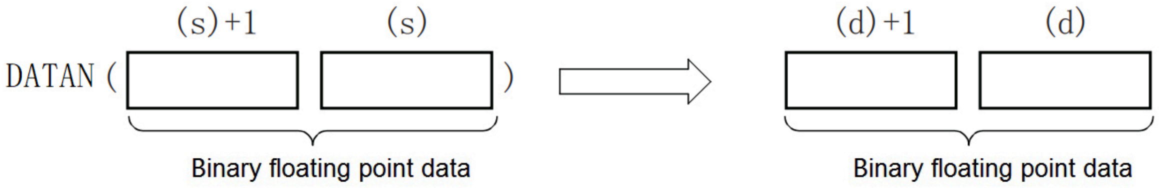
Parameter	Content	Range	Data type	Data type (label)
(s)	The angle data for TAN $^{-1}$ (arctangent) calculation or the start device number that stores the angle data	$0, 2^{-126} \le \|(s)\| < 2^{128}$	Single precision real number	ANYREAL_32
(d)	The start device number that stores operation result	-π/2 to π/2	Single precision real number	ANYREAL_32

Device use

Instruction	Parameter	Devices								Offset modification	Pulse extension
		T	C	D	R	SD	LC	HSC	E	[D]	XXP
DATAN	Parameter 1	●	●	●	●	●	●	●	●	●	●
	Parameter 2	●	●	●	●	●	●	●		●	●

Features

Calculate the TAN -1 ((arctangent) value of the angle specified in (s), and store the calculation result in the device number specified in (d).



The angle (calculation result) stored in (d) is stored in the unit of radians (-π/2) to (π/2).

- The related devices are as follows.

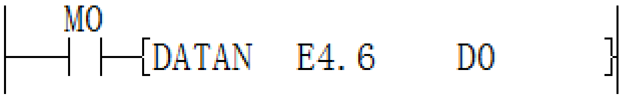
Devices	Name	Content	
		Condition	Operation
SM153	Zero	The operation result is zero	The zero flag (SM153) turns ON.
SM152	Borrow	The absolute value of operation result<2 $^{-126}$	The value of (d) becomes the minimum value of 32-bit real numbers (2^{-126}), and the borrow flag (SM152) turns ON.

Error code

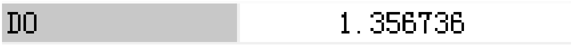
Error code	Content
4085H	The write address in (s) exceeds the device range

4086H	The write address in (d) exceeds the device range
4084H	When the content of the device specified by (s) is an irregular number, a non-number and ±∞

Example



Calculate the arctangent value of 4.6 and the result is 1.356736



D0	1.356736

DCOS/Single precision real number COS operation

DCOS (P)

After calculating the COS (cosine) value of the angle specified in (s), the calculation result is stored in the device number specified in (d).

-[DCOS (s) (d)]

Content, range and data type

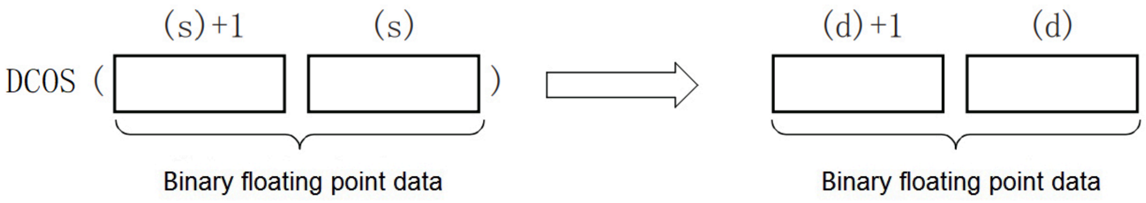
Parameter	Content	Range	Data type	Data type (label)
(s)	The angle data for COS (cosine) calculation or the start device number that stores the angle data	0, 2^{-126} ≤|(s)|< 2^{128}	Single precision real number	ANYREAL_32
(d)	The start device number that stores operation result	-	Single precision real number	ANYREAL_32

Device used

Instruction	Parameter	Devices								Offset modification	Pulse extension
		T	C	D	R	SD	LC	HSC	E	[D]	XXP
DCOS	Parameter 1	●	●	●	●	●	●	●	●	●	●
	Parameter 2	●	●	●	●	●	●	●		●	●

Features

After calculating the COS (cosine) value of the angle specified in (s), store the calculation result in the device number specified in (d).



For the angle specified in (s), set it in radians (angle×π÷180).

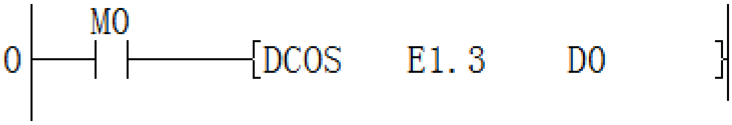
- The related devices are as follows.

Devices	Name	Content	
		Condition	**Operation**
SM153	zero	The operation result is zero	The zero flag (SM153) turns ON.
SM152	Borrow	The absolute value of operation result<2^{-126}	The value of (d) becomes the minimum value of 32-bit real numbers (2^{-126}), and the borrow flag (SM152) turns ON.

Error code

Error code	Content
4085H	The write address in (s) exceeds the device range
4086H	The write address in (d) exceeds the device range
4084H	When the content of the device specified by (s) is an irregular number, a non-number and ±∞

Example



Calculate the cosine value of 1.3 and the result is 2.674989E-1

D0 2.674989E-1

DCOSH/Single precision real number COSH operation

DCOSH(P)

After calculating the DCOSH (hyperbolic cosine) value of the angle specified in (s), the calculation result is stored in the device number specified in (d).

-[DCOSH (s) (d)]

Content, range and data type

Parameter	Content	Range	Data type	Data type (label)
(s)	The angle data for DCOSH (hyperbolic cosine) calculation or the device start number that stores the angle data	0, 2^{-126} ≤|(s)|< 2^{128}	Single precision real number	ANYREAL_32
(d)	The start device number that stores operation result	-	Single precision real number	ANYREAL_32

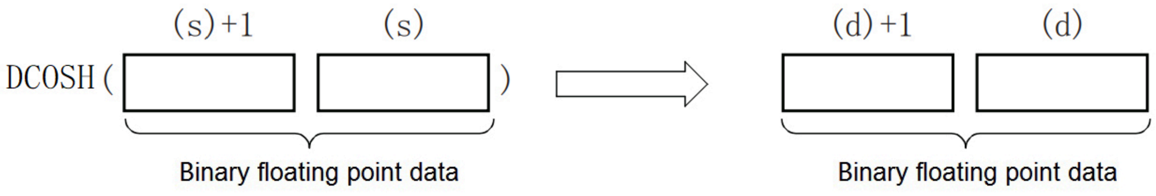
Device used

Instruction	Parameter	Devices								Offset modification	Pulse extension
		T	C	D	R	SD	LC	HSC	E	[D]	XXP
DCOSH	Parameter 1	●	●	●	●	●	●	●	●	●	●

Parameter 2	●	●	●	●	●	●	●		●	●

Features

• After calculating the DCOSH (hyperbolic cosine) value of the angle specified in (s), the calculation result is stored in the device number specified in (d).



This instruction is to take the COSH value of a binary floating point number. The calculation formula is cosh value=(e s +e -s)/2.

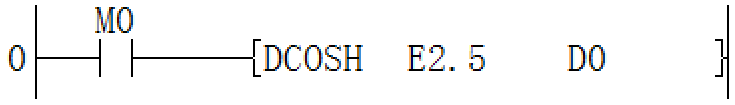
• The related devices are as follows.

Devices	Name	Content	
		Condition	Operation
SM151	carry	The absolute value of the operation result>2^{128}	The value of (d) becomes the maximum value of 32-bit real numbers (2^{128}), and the carry flag (SM151) turns on.

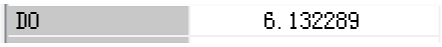
Error code

Error code	Content
4085H	The write address in (s) exceeds the device range
4086H	The write address in (d) exceeds the device range
4084H	When the content of the device specified by (s) is an irregular number, a non-number and ±∞

Example



Calculate the hyperbolic cosine value of 2.5, and the result is 6.132289



```
D0        6.132289
```

DSIN/Single precision real number SIN operation

DSIN(P)

After calculating the SIN (sine) value of the angle specified in (s), the calculation result is stored in the device number specified in (d).

-[DSIN (s) (d)]

Content, range and data type

Parameter	Content	Range	Data type	Data type (label)

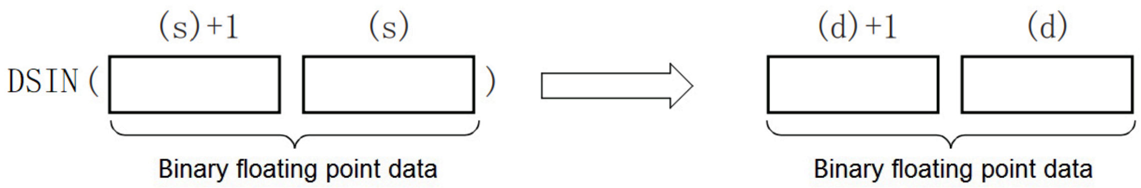
(s)	The angle data for SIN (sine) calculation or the device start number that stores the angle data	$0, 2^{-126} \leq \lvert(s)\rvert < 2^{128}$	Single precision real number	ANYREAL_32
(d)	The start device number that stores operation result	-	Single precision real number	ANYREAL_32

Device used

Instruction	Parameter	Devices								Offset modification	Pulse extension
		T	C	D	R	SD	LC	HSC	E	[D]	XXP
DSIN	Parameter 1	●	●	●	●	●	●	●	●	●	●
	Parameter 2	●	●	●	●	●	●	●		●	●

Features

After calculating the SIN (sine) value of the angle specified in (s), the calculation result is stored in the device number specified in (d).



For the angle specified in (s), set it in radians (angle×π÷180).

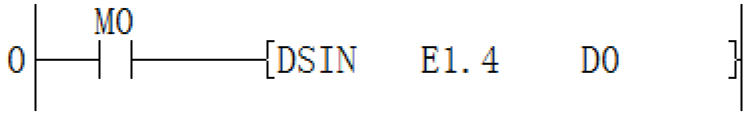
- The related devices are as follows.

Devices	Name	Content	
		Condition	Operation
SM153	Zero	The operation result is zero	The zero flag (SM153) turns ON.
SM152	Borrow	The absolute value of operation result <2 -126	The value of (d) becomes the minimum value of a 32-bit real number (2^{-126}), and the borrow flag (SM152) turns on.

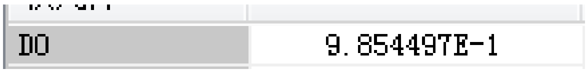
Error code

Error code	Content
4085H	The write address in (s) exceeds the device range
4086H	The write address in (d) exceeds the device range
4084H	When the content of the device specified by (s) is an irregular number, a non-number and ±∞

Example



Calculate the sine of 1.4 and the result is 0.9854497



D0 9.854497E-1

DSINH/Single precision real number SINH operation

DSINH(P)

After calculating the SINH (hyperbolic sine) value of the angle specified in (s), the calculation result is stored in the device number specified in (d).

-[DSINH (s) (d)]

Content, range and data type

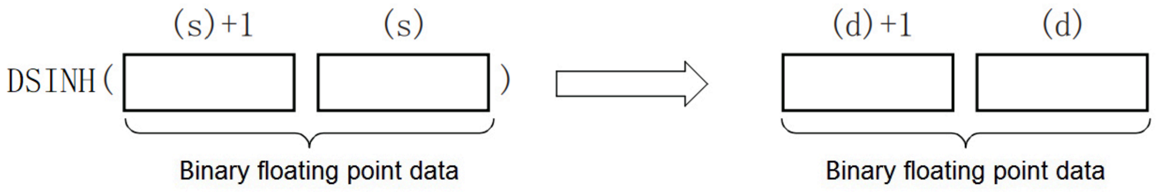
Parameter	Content	Range	Data type	Data type (label)
(s)	The angle data for SINH (hyperbolic sine) calculation or the device start number that stores the angle data	0, 2^{-126} ≤|(s)|< 2^{128}	Single precision real number	ANYREAL_32
(d)	The start device number that stores operation result	-	Single precision real number	ANYREAL_32

Device used

Instruction	Parameter	Devices								Offset modification	Pulse extension
		T	C	D	R	SD	LC	HSC	E	[D]	XXP
DSINH	Parameter 1	●	●	●	●	●	●	●	●	●	●
	Parameter 2	●	●	●	●	●	●	●		●	●

Features

After calculating the SINH (hyperbolic sine) value of the angle specified in (s), the calculation result is stored in the device number specified in (d).



The instruction is to take the SINH value from a binary floating point number. The calculation formula is sinh value = (e s -e -s)/2.

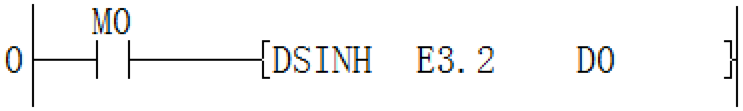
The related devices are shown below.

Devices	Name	Content	
		Condition	Operation
SM153	Zero	The operation result is zero	The zero flag (SM153) turns ON.
SM152	Borrow	The absolute value of operation result <2^{-126}	The value of (d) becomes the minimum value of a 32-bit real number (2^{-126}), and the borrow flag (SM152) turns on.
SM151	Carry	The absolute value of operation result > 2^{128}	The value of (d) becomes the maximum value of 32-bit real numbers (2^{128}), and the carry flag (SM151) turns on.

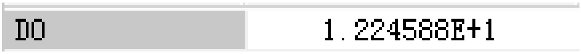
Error code

Error code	Content
4085H	The write address in (s) exceeds the device range
4086H	The write address in (d) exceeds the device range
4084H	When the content of the device specified by (s) is an irregular number, a non-number and ±∞

Example



Calculate the hyperbolic sine value of 3.2 and the result is 12.24588


D0 1.224588E+1

DTAN/Single precision real number TAN operation

DTAN(P)

After calculating the TAN (tangent) value of the angle specified in (s), the calculation result is stored in the device number specified in (d).

-[DTAN (s) (d)]

Content, range and data type

Parameter	Content	Range	Data type	Data type (label)
(s)	The angle data for TAN (tangent) calculation or the device start number that stores the angle data	0, 2^{-126} ≤|(s)|< 2^{128}	Single precision real number	ANYREAL_32
(d)	The start device number that stores operation result	-	Single precision real number	ANYREAL_32

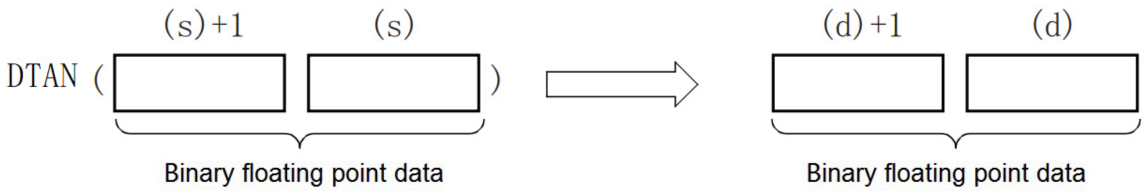
Device used

Instruction	Parameter	Devices								Offset modification	Pulse extension
		T	C	D	R	SD	LC	HSC	E	[D]	XXP

DTAN	Parameter 1	●	●	●	●	●	●	●	●	●	●
	Parameter 2	●	●	●	●	●	●	●		●	●

Features

After calculating the TAN (tangent) value of the angle specified in (s), the calculation result is stored in the device number specified in (d).



For the angle specified in (s), set it in radians (angle×π÷180).

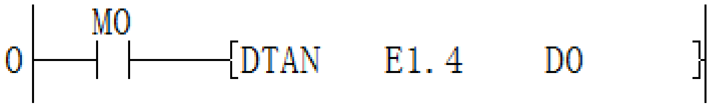
The related devices are shown below.

Devices	Name	Content	
		Condition	Operation
SM153	Zero	The operation result is zero	The zero flag (SM153) turns ON.
SM152	Borrow	The absolute value of operation result <2^{-126}	The value of (d) becomes the minimum value of a 32-bit real number (2^{-126}), and the borrow flag (SM152) turns on.
SM151	Carry	The absolute value of operation result> 2^{128}	The value of (d) becomes the maximum value of 32-bit real numbers (2^{128}), and the carry flag (SM151) turns on.

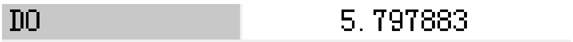
Error code

Error code	Content
4085H	The write address in (s) exceeds the device range
4086H	The write address in (d) exceeds the device range
4084H	When the content of the device specified by (s) is an irregular number, a non-number and ±∞

Example



Calculate the tangent of 1.4 and the result is 5.797883



D0	5.797883

DATANH/Single precision real number TANH operation

DTANH(P)

After calculating the DTANH (hyperbolic tangent) value of the angle specified in (s), the calculation result is stored in the device number specified in (d).

-[DTANH (s) (d)]

Content, range and data type

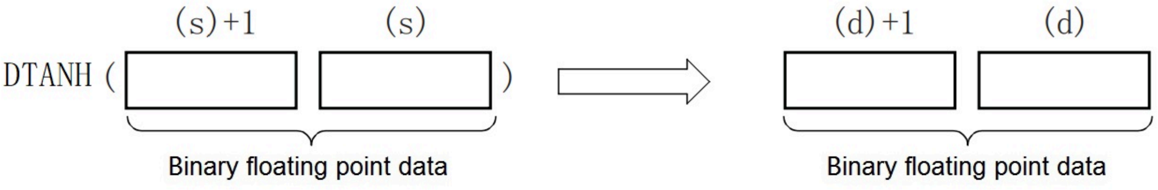
Parameter	Content	Range	Data type	Data type (label)
(s)	The angle data for DTANH (hyperbolic tangent) calculation or the device start number that stores the angle data	0, 2^{-126} ≤|(s)|< 2^{128}	Single precision real number	ANYREAL_32
(d)	The start device number that stores operation result	-	Single precision real number	ANYREAL_32

Device used

Instruction	Parameter	Devices								Offset modification	Pulse extension
		T	C	D	R	SD	LC	HSC	E	[D]	XXP
DTANH	Parameter 1	●	●	●	●	●	●	●	●	●	●
	Parameter 2	●	●	●	●	●	●	●		●	●

Features

After calculating the DTANH (hyperbolic tangent) value of the angle specified in (s), the calculation result is stored in the device number specified in (d).



The instruction is to take the TANH value of a binary floating point number. The calculation formula is tanh value=(e^{s} -e^{-s})/(e^{s} +e^{-s}).

The related devices are shown below.

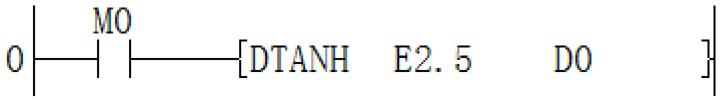
Device	Name	Content	
		Condition	Operation
SM153	Zero	The operation result is zero	The zero flag (SM153) turns ON.
SM152	Borrow	The absolute value of operation result <2^{-126}	The value of (d) becomes the minimum value of a 32-bit real number (2^{-126}), and the borrow flag (SM152) turns on.

Error code

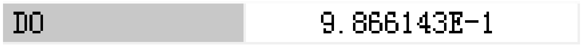
Error code	Content

4085H	The write address in (s) exceeds the device range
4086H	The write address in (d) exceeds the device range
4084H	When the content of the device specified by (s) is an irregular number, a non-number and ±∞

Example



Calculate the hyperbolic tangent of 2.5, and the result is 0.9866143



D0 9.866143E-1

DDEG/Single precision real number radian → angle conversion

DDEG(P)

Convert the size unit of the angle from the radian unit specified in (s) to the degree unit (DEG. unit), and store it in the device number specified in (d).

-[DDEG (s) (d)]

Content, range and data type

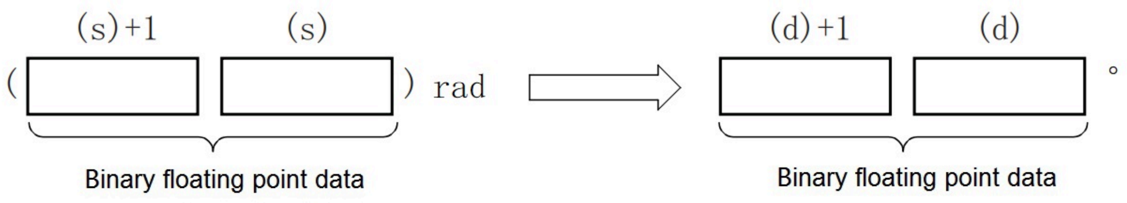
Parameter	Content	Range	Data type	Data type (label)
(s)	The radian angle that converts the degree unit or the device start number that stores the radian angle	0, 2^{-126} ≤|(s)|< 2^{128}	Single precision real number	ANYREAL_32
(d)	The device start number that stores the value converted in degrees	-π/2 to π/2	Single precision real number	ANYREAL_32

Device used

Instruction	Parameter	Devices								Offset modification	Pulse extension
		T	C	D	R	SD	LC	HSC	E	[D]	XXP
DDEG	Parameter 1	●	●	●	●	●	●	●	●	●	●
	Parameter 2	●	●	●	●	●	●	●		●	●

Features

The angle size unit is converted from the radian unit specified in (s) to the degree unit (DEG. unit), and then stored in the device number specified in (d).



The conversion from degree unit to radian unit is performed as follows.

Radian unit = degree unit * 180/π

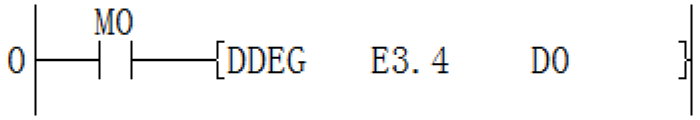
• The related devices are as follows.

Devices	Name	Content	
		Condition	**Operation**
SM153	Zero	The operation result of is zero (when the mantissa part is zero)	The zero flag (SM153) turns ON.
SM151	Carry	The absolute value of the operation result>2^{128}	The value of (d) becomes the maximum value of 32-bit real numbers (2^{128}), and the carry flag (SM151) turns on.

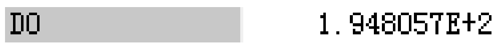
Error code

Error code	Content
4085H	The write address in (s) exceeds the device range
4086H	The write address in (d) exceeds the device range
4084H	When the content of the device specified by (s) is an irregular number, a non-number and ±∞

Example



The result is 194.8057



```
D0        1.948057E+2
```

DRAD/Single precision real number conversion angle → radian conversion

DRAD(P)

The angle size unit is converted from the degree unit (DEG. unit) specified in (s) to the radian unit and stored in the device number specified in (d).

-[DRAD (s) (d)]

Content, range and data type

Parameter	Content	Range	Data type	Data type (label)

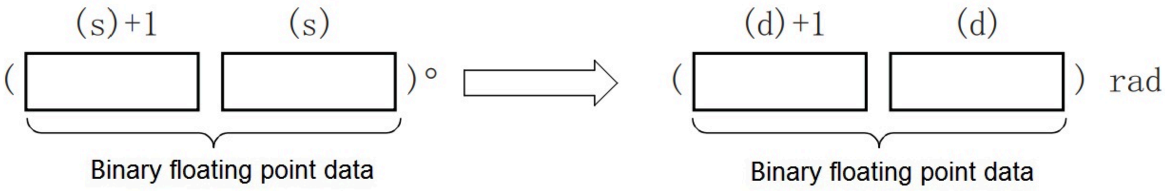
(s)	The radian angle that converts the degree unit or the device start number that stores the angle	$0, 2^{-126} \leq \|(s)\| < 2^{128}$	Single precision real number	ANYREAL_32
(d)	The device start number that stores the value converted in degrees	-	Single precision real number	ANYREAL_32

Device used

Instruction	Parameter	Devices								Offset modification	Pulseextension
		T	C	D	R	SD	LC	HSC	E	[D]	XXP
DRAD	Parameter 1	●	●	●	●	●	●	●	●	●	●
	Parameter 2	●	●	●	●	●	●	●		●	

Features

The angle size unit is converted from the degree unit (DEG. unit) specified in (s) to the radian unit and stored in the device number specified in (d).



Degree unit → radian unit

The conversion is performed as follows.

Radian unit = degree unit*π/180

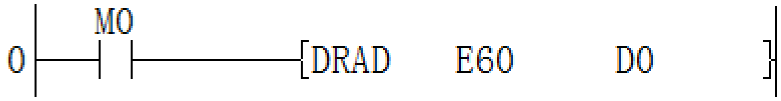
- The related devices are as follows.

Devices	Name	Content	
		Condition	Operation
SM153	Zero	The operation result is zero	The zero flag (SM153) turns ON.
SM152	Borrow	The absolute value of operation result $<2^{-126}$	The value of (d) becomes the minimum value of a 32-bit real number (2^{-126}), and the borrow flag (SM152) turns on.

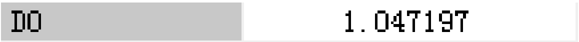
Error code

Error code	Content
4085H	The write address in (s) exceeds the device range
4086H	The write address in (d) exceeds the device range
4084H	When the content of the device specified by (s) is an irregular number, a non-number and ±∞

Example



The result is 1.047197


D0 1.047197

DEADD/Single precision real number addition operation

DEADD(P)

Add the binary floating point data specified in (s1) and the binary floating point data specified in (s2), and store the result in the device specified in (d).

-[DEADD (s1) (s2) (d)]

Content, range and data type

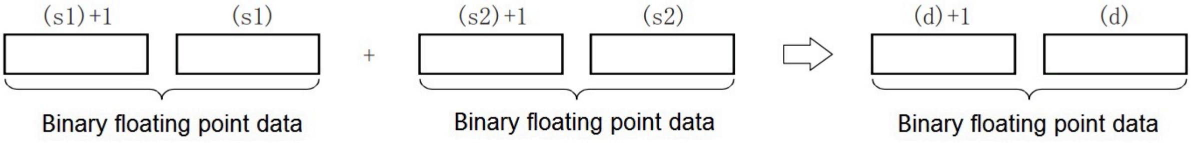
Parameter	Content	Range	Data type	Data type (label)
(s1)	The added data or the device start number that stores the added data	0, 2^{-126} ≤|(s)|< 2 ^{128}	Single precision real number	ANYREAL_32
(s2)	Addition data or the device start number that stores the addition data	0, 2^{-126} ≤|(s)|< 2^{128}	Single precision real number	ANYREAL_32
(d)	the device start number that stores the operation result	-	Single precision real number	ANYREAL_32

Device used

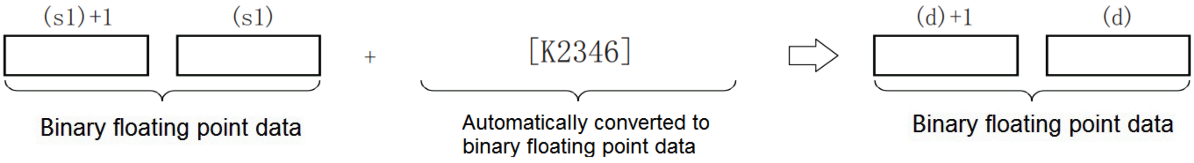
Instruction	Parameter	Devices										Offset modification	Pulse extension
		T	C	D	R	SD	LC	HSC	K	H	E	[D]	XXP
DEADD	Parameter 1	●	●	●	●	●	●	●	●	●	●	●	●
	Parameter 2	●	●	●	●	●	●	●	●	●	●	●	●
	Parameter 3	●	●	●	●	●	●	●				●	●

Features

Add the binary floating point data specified in (s1) and the binary floating point data specified in (s2), and store the result of the addition in the device specified in (d).


(s1)+1
(s1)
+
(s2)+1
(s2)
(d)+1
(d)
Binary floating point data
Binary floating point data
Binary floating point data

When constants (K, H) to (s1), (s2) are specified, the value is automatically converted to a binary floating point data.


(s1)+1
(s1)
Binary floating point data
+
[K2346]
Automatically converted to binary floating point data
(d)+1
(d)
Binary floating point data

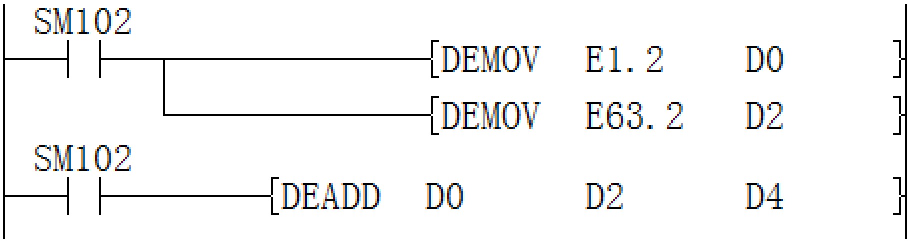
- The related devices are as follows.

Devices	Name	Content	
		Condition	Operation
SM153	Zero	The operation result is zero	The zero flag (SM153) turns ON.
SM152	Borrow	The absolute value of operation result <2^{-126}	The value of (d) becomes the minimum value of a 32-bit real number (2^{-126}), and the borrow flag (SM152) turns on.
SM151	Carry	The absolute value of operation result> 2^{128}	The value of (d) becomes the maximum value of 32-bit real numbers (2^{128}), and the carry flag (SM151) turns on.

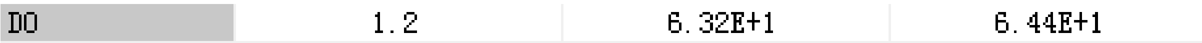
Error code

Error code	Content
4085H	The write address in (s1) and (s2) exceed the device range
4086H	The write address in (d) exceeds the device range
4084H	When the content of the device specified by (s1) and (s2) is an irregular number, a non-number and ±∞

Example



The result is 1.2 + 63.2 = 64.4



D0	1.2	6.32E+1	6.44E+1

DESUB/Single precision real number subtraction operation

DESUB(P)

Subtract the binary floating point data specified in (s1) and the binary floating point data specified in (s2), and store the result in the device specified in (d).

-[DESUB (s1) (s2) (d)]

Content, range and data type

Parameter	Content	Range	Data type	Data type (label)

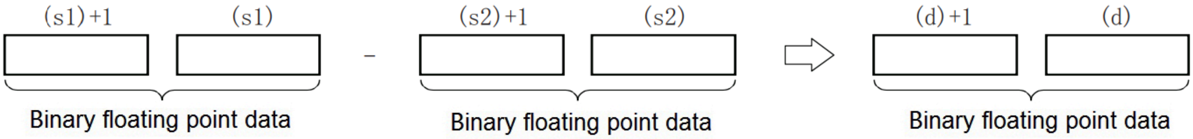
(s1)	The subtracted data or the device start number that stores the subtracted data	$0, 2^{-126} \leq \lvert(s)\rvert < 2^{128}$	Single precision real number	ANYREAL_32
(s2)	subtract data or the device start number that stores the subtracted data	$0, 2^{-126} \leq \lvert(s)\rvert < 2^{128}$	Single precision real number	ANYREAL_32
(d)	the device start number that stores the operation result	-	Single precision real number	ANYREAL_32

Device used

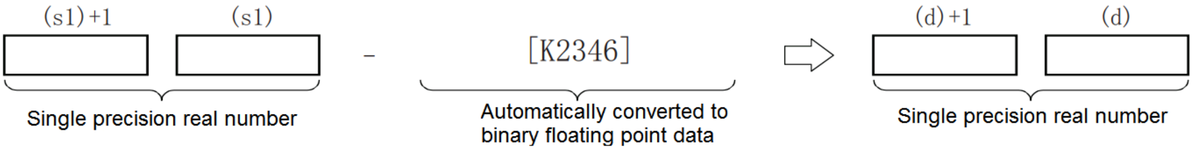
Instruction	Parameter	Devices										Offset modification	Pulse extension
		T	C	D	R	SD	LC	HSC	K	H	E	[D]	XXP
DESUB	Parameter 1	●	●	●	●	●	●	●	●	●	●	●	●
	Parameter 2	●	●	●	●	●	●	●	●	●	●	●	●
	Parameter 3	●	●	●	●	●	●	●				●	●

Features

• Subtract the binary floating point data specified in (s1) and the binary floating point data specified in (s2), and store the subtraction result in the device specified in (d).


(s1)+1
(s1)
Binary floating point data
-
(s2)+1
(s2)
Binary floating point data
(d)+1
(d)
Binary floating point data

When constants (K, H) to (s1), (s2) are specified, the value is automatically converted to a binary floating point data.


(s1)+1
(s1)
Single precision real number
-
[K2346]
Automatically converted to binary floating point data
(d)+1
(d)
Single precision real number

• The related devices are as follows.

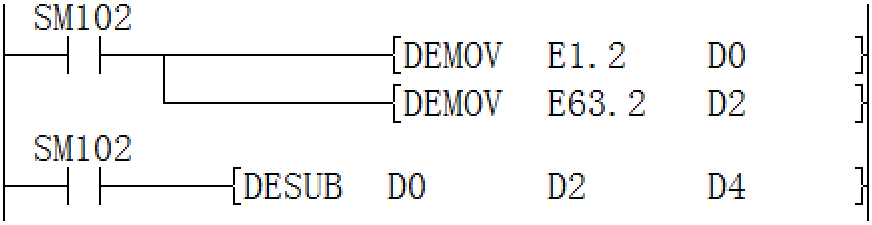
Devices	Name	Content	
		Condition	Operation
SM153	Zero	The operation result is zero	The zero flag (SM153) turns ON.
SM152	Borrow	The absolute value of operation result <2^{-126}	The value of (d) becomes the minimum value of a 32-bit real number (2^{-126}), and the borrow flag (SM152) turns on.

SM151	Carry	The absolute value of operation result> 2^{128}	The value of (d) becomes the maximum value of 32-bit real numbers (2^{128}), and the carry flag (SM151) turns on.

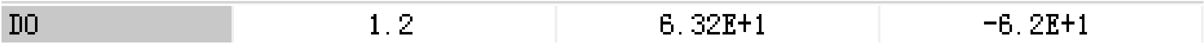
Error code

Error code	Content
4085H	The write address in (s1) and (s2) exceeds the device range
4086H	The write address in (d) exceeds the device range
4084H	When the content of the device specified by (s1) and (s2) is an irregular number, a non-number and ±∞

Example



The calculation result is 1.2-63.2 = -62



D0	1.2	6.32E+1	-6.2E+1

DEMUL/Single precision real number multiplication operation

DEMUL(P)

Multiply the binary floating point data specified in (s1) and the binary floating point data specified in (s2), and store the result in the device specified in (d).

-[DEMUL (s1) (s2) (d)]

Content, range and data type

Parameter	Content	Range	Data type	Data type (label)
(s1)	The sum data or the device start number that stores the sum data	0, 2^{-126} ≤|(s)|< 2^{128}	Single precision real number	ANYREAL_32
(s2)	The sum operation data or the device start number that stores the addition data	0, 2^{-126} ≤|(s)|< 2^{128}	Single precision real number	ANYREAL_32
(d)	the device start number that stores the operation result	-	Single precision real number	ANYREAL_32

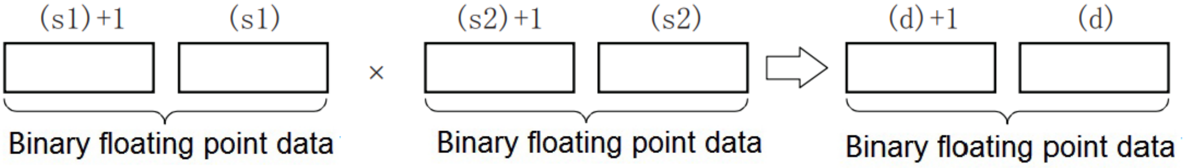
Device used

Instruction	Parameter	Devices										Offset modification	Pulse extension
		T	C	D	R	SD	LC	HSC	K	H	E	[D]	XXP
DEMUL	Parameter 1	●	●	●	●	●	●	●	●	●	●	●	●
	Parameter 2	●	●	●	●	●	●	●	●	●	●	●	●

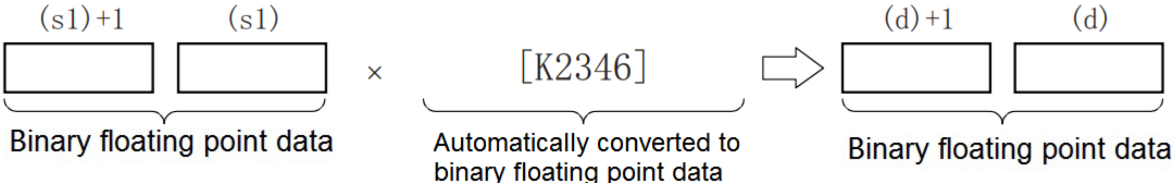
Parameter 3	●	●	●	●	●	●	●				●	●

Features

Multiply the binary floating point data specified in (s1) and the binary floating point data specified in (s2), and store the multiplication result in the device specified in (d).


(s1)+1
(s1)
×
(s2)+1
(s2)
(d)+1
(d)
Binary floating point data
Binary floating point data
Binary floating point data

When constants (K, H) to (s1), (s2) are specified, the value is automatically converted to a binary floating point data.


(s1)+1
(s1)
×
[K2346]
(d)+1
(d)
Binary floating point data
Automatically converted to binary floating point data
Binary floating point data

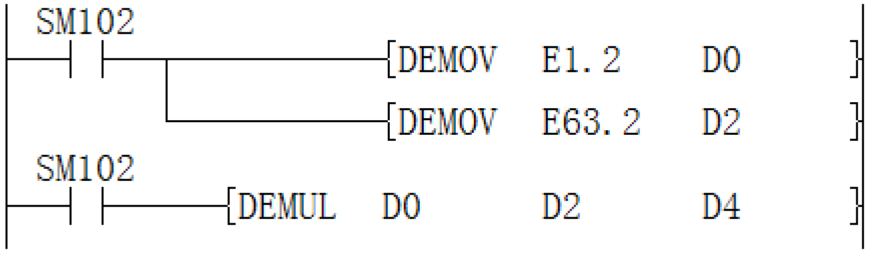
- The related devices are as follows.

Devices	Name	Content	
		Condition	Operation
SM153	Zero	The operation result is zero	The zero flag (SM153) turns ON.
SM152	Borrow	The absolute value of operation result <2^{-126}	The value of (d) becomes the minimum value of a 32-bit real number (2^{-126}), and the borrow flag (SM152) turns on.
SM151	Carry	The absolute value of operation result> 2^{128}	The value of (d) becomes the maximum value of 32-bit real numbers (2^{128}), and the carry flag (SM151) turns on.

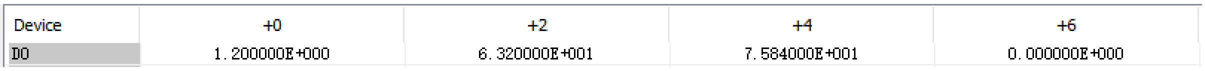
Error code

Error code	Content
4085H	The write address in (s1) and (s2) exceeds the device range
4086H	The write address in (d) exceeds the device range
4084H	When the content of the device specified by (s1) and (s2) is an irregular number, a non-number and ±∞

Example



The calculated result: 1.2*63.2 = 75.84



Device	+0	+2	+4	+6
D0	1.200000E+000	6.320000E+001	7.584000E+001	0.000000E+000

DEDIV/Single precision real number division operation

DEDIV(P)

Divide the binary floating point data specified in (s1) and the binary floating point data specified in (s2), and store the result in the device specified in (d).

-[DEDIV (s1) (s2) (d)]

Content, range and data type

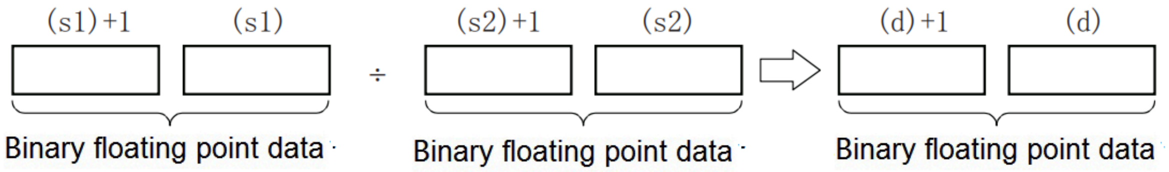
Parameter	Content	Range	Data type	Data type (label)
(s1)	The divided data or the device start number that stores the devided data	$0, 2^{-126} \leq \|(s)\| < 2^{128}$	Single precision real number	ANYREAL_32
(s2)	Division operation data or the device start number that stores the division operation data	$0, 2^{-126} \leq \|(s)\| < 2^{128}$	Single precision real number	ANYREAL_32
(d)	the device start number that stores the operation result	-	Single precision real number	ANYREAL_32

Device used

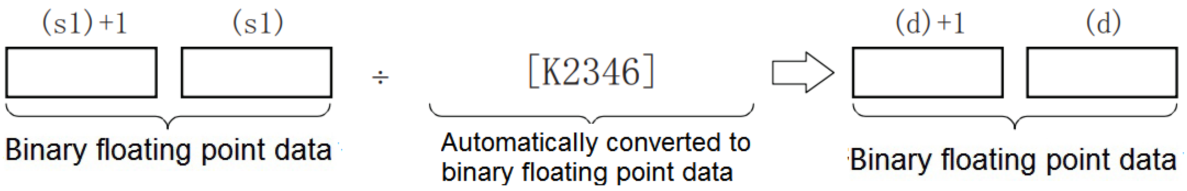
Instruction	Parameter	Devices										Offset modification	Pulse extension
		T	C	D	R	SD	LC	HSC	K	H	E	[D]	XXP
DEDIV	Parameter 1	●	●	●	●	●	●	●	●	●	●	●	●
	Parameter 2	●	●	●	●	●	●	●	●	●	●	●	●
	Parameter 3	●	●	●	●	●	●	●				●	●

Features

Divide the binary floating point data specified in (s1) and the binary floating point data specified in (s2), and store the result of the division in the device specified in (d).


(s1)+1
(s1)
÷
(s2)+1
(s2)
(d)+1
(d)
Binary floating point data
Binary floating point data
Binary floating point data

When constants (K, H) to (s1), (s2) are specified, the value is automatically converted to a binary floating point data.


(s1)+1
(s1)
÷
[K2346]
(d)+1
(d)
Binary floating point data
Automatically converted to binary floating point data
Binary floating point data

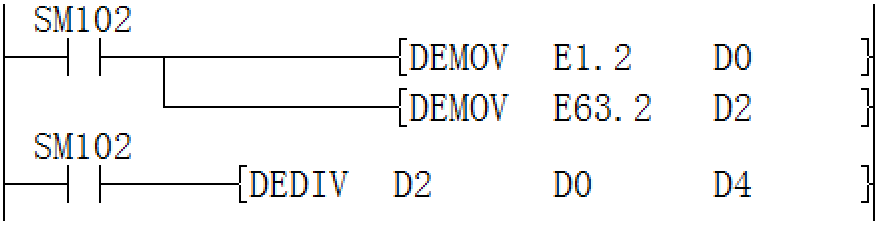
- The related devices are as follows.

Devices	Name	Content	
		Condition	Operating
SM153	Zero	The operation result is zero	The zero flag (SM153) turns ON.
SM152	Borrow	The absolute value of operation result <2^{-126}	The value of (d) becomes the minimum value of a 32-bit real number (2^{-126}), and the borrow flag (SM152) turns on.
SM151	Carry	The absolute value of operation result> 2^{128}	The value of (d) becomes the maximum value of 32-bit real numbers (2^{128}), and the carry flag (SM151) turns on.

Error code

Error code	Content
4085H	The write address in (s1) and (s2) exceeds the device range
4086H	The write address in (d) exceeds the device range
4084H	When the content of the device specified by (s1) and (s2) is an irregular number, a non-number and ±∞
4080H	(s2) value is 0

Example



Get the calculation result: 63.2 / 1.2 = 52.66666667

D0	1.2	6.32E+1	5.266666E+1

DEMOV/Single precision real data transmission

DEMOV(P)

Transfer the binary floating point data data stored in the device specified in (s) to the device specified in (d).

-[DEMOV (s) (d)]

Content, range and data type

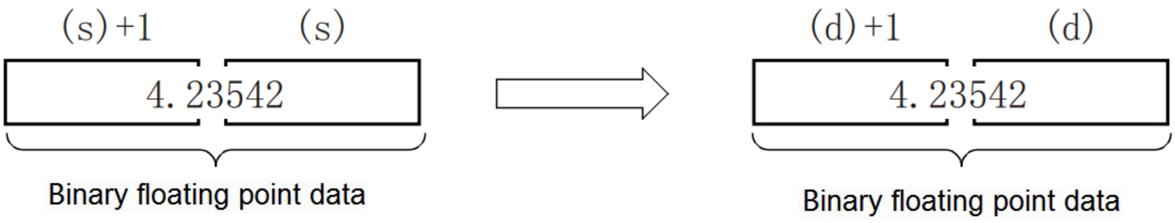
Parameter	Content	Range	Data type	Data type (label)
(s)	The transmitted data or the device that stores the transmitted data	0, 2^{-126} ≤|(s)|< 2^{128}	Single precision real number	ANYREAL_32
(d)	The device number that stores the transmit destination data	-	Single precision real number	ANYREAL_32

Device used

Instruction	Parameter	Devices								Offset modification	Pulse extension
		T	C	D	R	SD	LC	HSC	E	[D]	XXP
DEMOV	Parameter 1	●	●	●	●	●	●	●	●	●	●
	Parameter 2	●	●	●	●	●	●	●		●	●

Features

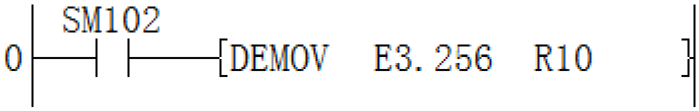
Transfer the binary floating point data data stored in the device specified in (s) to the device specified in (d).



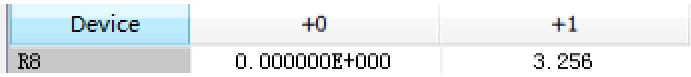
Error code

Error code	Content
4085H	(s) read address exceeds the device range
4086H	(d) write address exceeds the device range

Example



Assign 3.265 to R10



Device	+0	+1
R8	0.000000E+000	3.256

DEBCD/Binary floating point → decimal floating point conversion

DEBCD(P)

After converting the binary floating point specified in (s) into a decimal floating point, it is stored in the device specified in (d).

-[DEBCD (s) (d)]

Content, range and data type

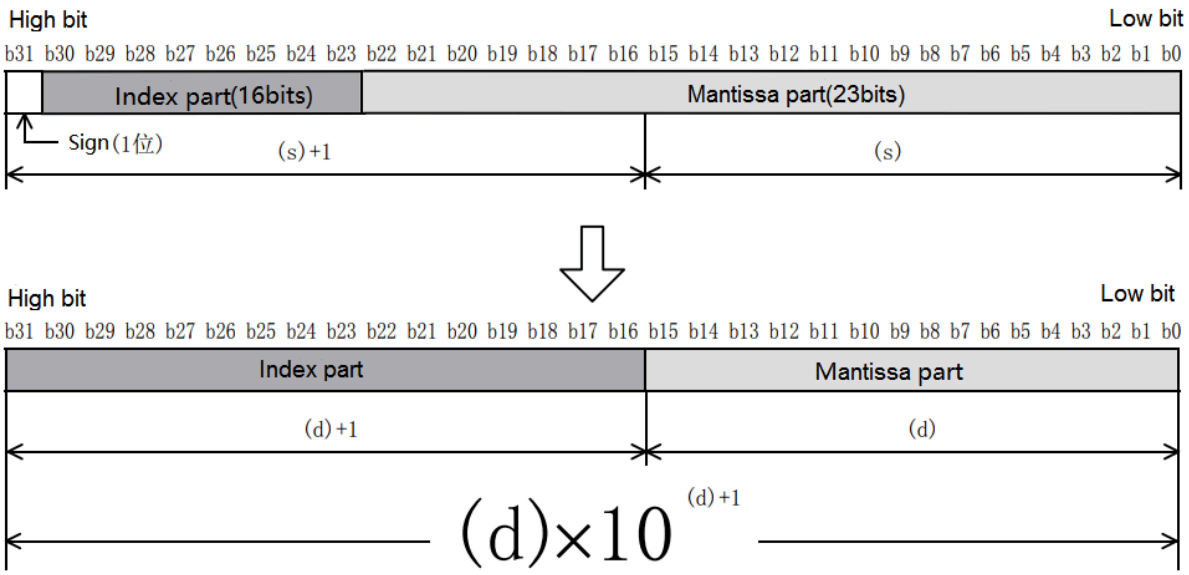
Parameter	Content	Range	Data type	Data type (label)
(s)	The device number that stores the binary floating point data	$0, 2^{-126} \leq \|(s)\| < 2^{128}$	Single precision real number	ANYREAL_32
(d)	The device number that stores the converted decimal floating point data	-	Real number	ANY32

Device used

Instruction	Parameter	Devices								Offset modification	Pulse extension
		T	C	D	R	SD	LC	HSC	E	[D]	XXP
DEBCD	Parameter 1	●	●	●	●	●	●	●	●	●	●
	Parameter 2	●	●	●	●	●	●	●		●	●

Features

After converting the binary floating point specified in (s) into a decimal floating point, it is stored in the device specified in (d).


High bit
Low bit
b31 b30 b29 b28 b27 b26 b25 b24 b23 b22 b21 b20 b19 b18 b17 b16 b15 b14 b13 b12 b11 b10 b9 b8 b7 b6 b5 b4 b3 b2 b1 b0
Index part(16bits)
Mantissa part(23bits)
Sign(1位)
(s)+1
(s)
High bit
Low bit
b31 b30 b29 b28 b27 b26 b25 b24 b23 b22 b21 b20 b19 b18 b17 b16 b15 b14 b13 b12 b11 b10 b9 b8 b7 b6 b5 b4 b3 b2 b1 b0
Index part
Mantissa part
(d)+1
(d)
(d)×10 (d)+1

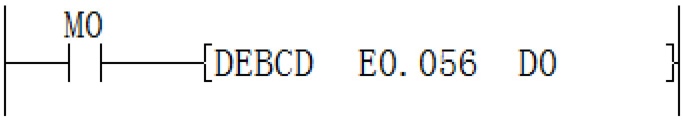
#Note

All floating-point operations are performed in binary floating-point. However, the binary floating point is a difficult-to-understand value (special monitoring method), so by converting it into a decimal floating point operation, it is convenient for peripheral equipment to monitor and so on.

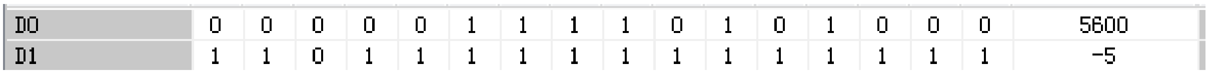
Error code

Error code	Content
4085H	The write address in (s) exceeds the device range
4086H	The write address in (d) exceeds the device range
4084H	When the content of the device specified by (s) is an irregular number, a non-number and ±∞

Example



Get the result: 5600 ×10 -5



D0	0	0	0	0	0	1	1	1	1	0	1	0	1	0	0	0	5600
D1	1	1	0	1	1	1	1	1	1	1	1	1	1	1	1	1	-5

DEBIN/Decimal floating point → binary floating point conversion

DEBIN(P)

Convert the decimal floating point specified in (s) to binary floating point and store it in the device specified in (d).

-[DEBIN (s) (d)]

Content, range and data type

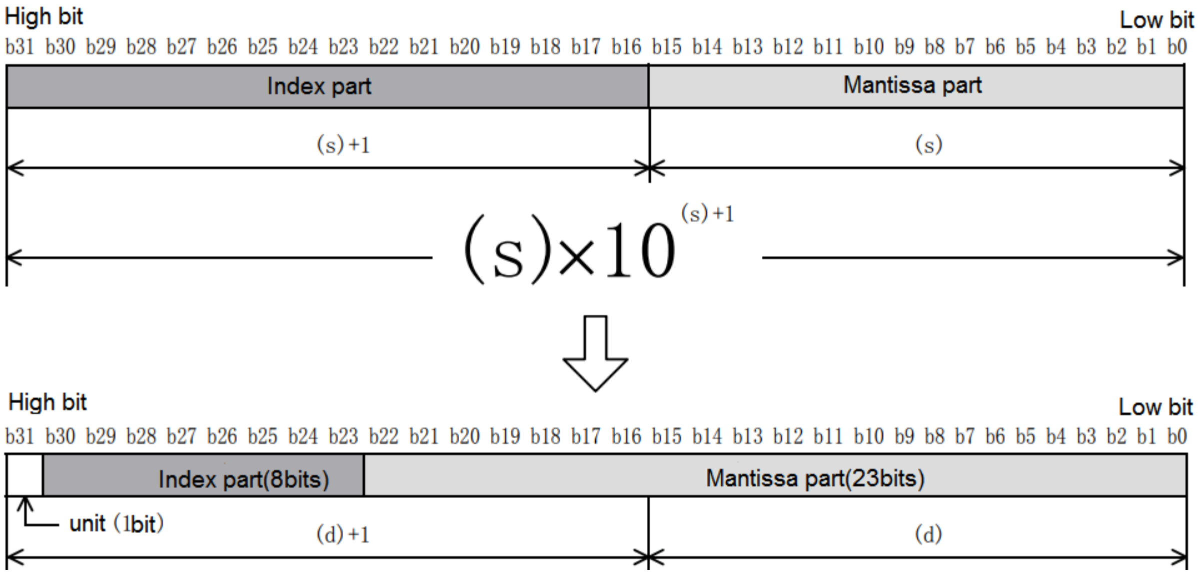
Parameter	Content	Range	Data type	Data type (label)
(s)	The device number that stores the decimal floating point data	-	Real	ANY32
(d)	The device number that stores the converted binary floating point data	-	Single precision real number	ANYREAL_32

Device used

Instruction	Parameter	Devices							Offset modification	Pulse extension
		T	C	D	R	SD	LC	HSC	[D]	XXP
DEBIN	Parameter 1	●	●	●	●	●	●	●	●	●
	Parameter 2	●	●	●	●	●	●	●	●	●

Features

Convert the decimal floating point specified in (s) to binary floating point and store it in the device specified in (d).


High bit
Low bit
b31 b30 b29 b28 b27 b26 b25 b24 b23 b22 b21 b20 b19 b18 b17 b16 b15 b14 b13 b12 b11 b10 b9 b8 b7 b6 b5 b4 b3 b2 b1 b0
Index part
Mantissa part
(s)+1
(s)
(s)×10^(s)+1
High bit
Low bit
b31 b30 b29 b28 b27 b26 b25 b24 b23 b22 b21 b20 b19 b18 b17 b16 b15 b14 b13 b12 b11 b10 b9 b8 b7 b6 b5 b4 b3 b2 b1 b0
Index part(8bits)
Mantissa part(23bits)
unit (1bit)
(d)+1
(d)

- The related devices are as follows.

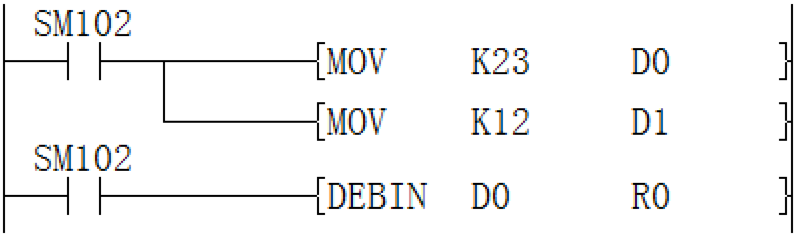
Devices	Name	Content	
		Condition	Operation
SM153	Zero	The operation result is zero	The zero flag (SM153) turns ON.
SM152	Borrow	The absolute value of operation result $<2^{-126}$	The value of (d) becomes the minimum value of a 32-bit real number (2^{-126}), and the borrow flag (SM152) turns on.

SM151	Carry	The absolute value of operation result> 2^{128}	The value of (d) becomes the maximum value of 32-bit real numbers (2^{128}), and the carry flag (SM151) turns on.

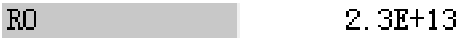
Error code

Error code	Content
4085H	(s) read address exceeds the device range
4086H	(d) write address exceeds the device range

Example



The result after conversion:


R0 2.3E+13

DENEG/Single precision real number sign inversion

DENEG(P)

After inverting the sign of the single precision real number of the device specified in (d), it is stored in the device specified in (d).

-[DEBEG (d)]

Content, range and data type

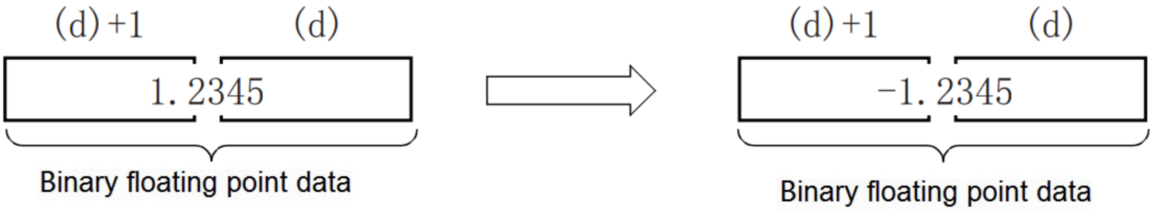
Parameter	Content	Range	Data type	Data type (label)
(d)	The device start number that stores the sign-inverted binary floating point data	-	Single precision real number	ANYREAL_32

Device used

Instruction	Parameter	Devices							Offset modification	Pulse extension
		T	C	D	R	SD	LC	HSC	[D]	XXP
DENEG	Parameter 1	●	●	●	●	●	●	●	●	●

Features

The sign of the binary floating point data of the device specified in (d) is inverted and stored in the device specified in (d).

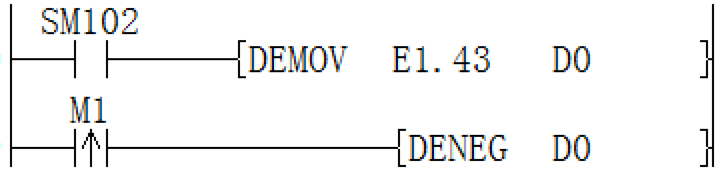


Used when inverting positive and negative signs.

Error code

Error code	Content
4086H	The write address in (d) exceeds the device range

Example



It becomes -1.43 after conversion

D0 -1.43

DECMP/Single precision real number comparison

DECMP(P)

Compare two data (binary floating point data), and output their large, small, and consistent results to the bit device (3 points).

-[DECMP (s1) (s2) (d)]

Content, range and data type

Parameter	Content	Range	Data type	Data type (label)
(s1)	Comparison data or the device number that stores the comparison data	$0, 2^{-126} \leq \|(s)\| < 2^{128}$	Single precision real number	ANYREAL_32
(s2)	Comparison data or the device number that stores the comparison data	$0, 2^{-126} \leq \|(s)\| < 2^{128}$	Single precision real number	ANYREAL_32
(d)	The start bit device number that outputs the comparison result (occupies 3 points)	-	Bit	ANYBIT_ARRAY

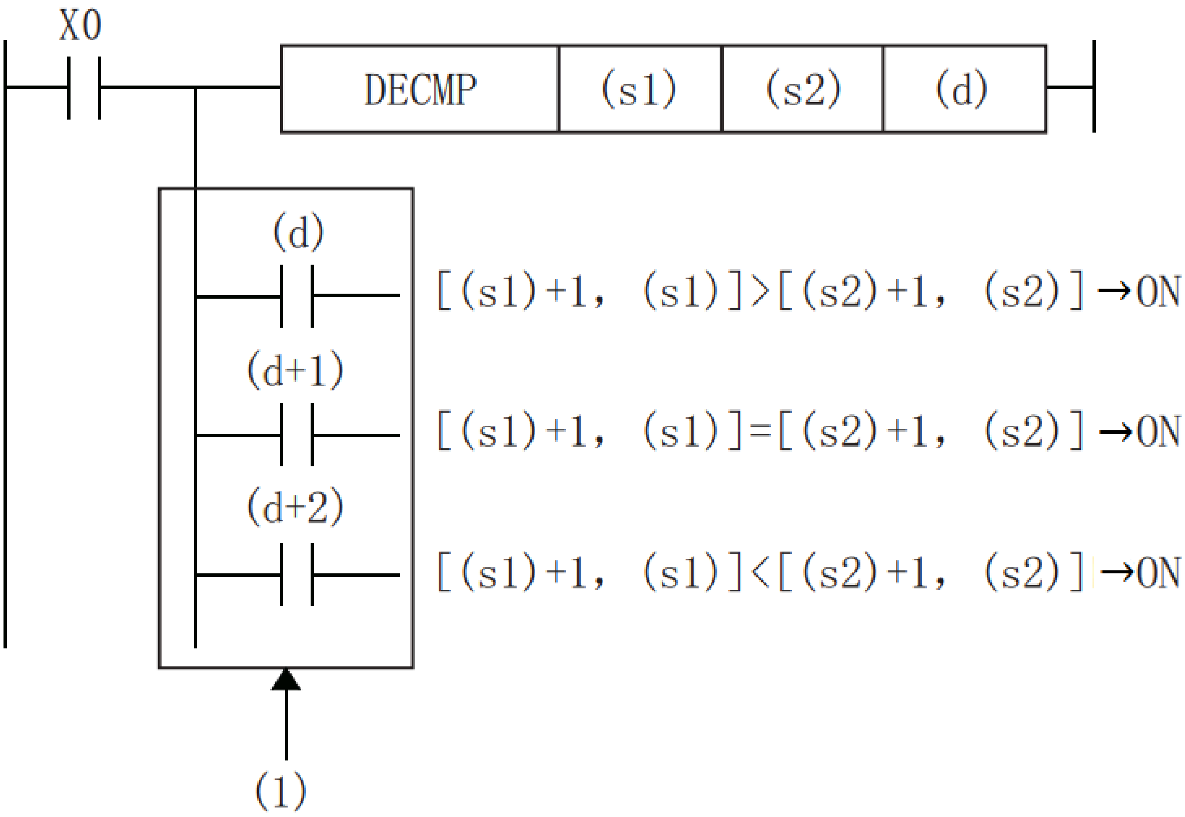
Device used

Instruction	Parameter	Devices															Offset modification	Pulse extension
		Y	M	S	SM	D.b	T	C	D	R	SD	LC	HSC	K	H	E	[D]	XXP

DECMP	Parameter 1						●	●	●	●	●	●	●	●	●	●	●	●
	Parameter 2						●	●	●	●	●	●	●	●	●	●	●	●
	Parameter 3	●	●	●	●	●											●	●

Features

Compare the comparison value (s1) and the comparison source (s2) as a floating point comparison. According to the result of small, consistent, and large, one of (d), (d)+1, (d)+2 turns ON.


X0
DECMP
(s1)
(s2)
(d)
(d)
[(s1)+1, (s1)]>[(s2)+1, (s2)]→ON
(d+1)
[(s1)+1, (s1)]=[(s2)+1, (s2)]→ON
(d+2)
[(s1)+1, (s1)]<[(s2)+1, (s2)]→ON
(1)

(1): Even if the command input is turned OFF and the DECMP command is not executed, (d) to (d)+2 will keep the state before X0 is turned OFF.

When the constant (K, H) to the device specified in (s1), (s2) is specified, the value BIN→binary floating point data conversion is processed automatically

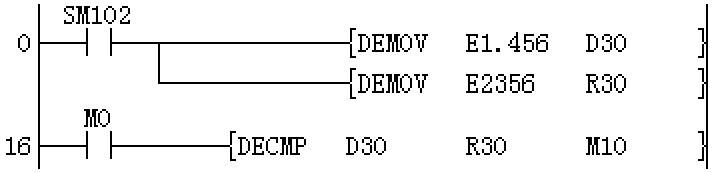
#**Note:** The device specified in (d) occupies 3 points [(d), (d)+1, (d)+2]. Please be careful not to overlap with devices used for other purposes.

Error code

Error code	**Content**

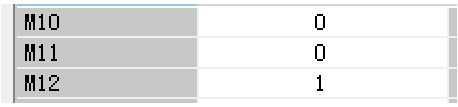
4085H	The write address in (s1) or (s2) exceeds the device range
4086H	The write address in (d) exceeds the device range
4084H	When the content of the device specified by (s) or (s2) is an irregular number, a non-number and ±∞

Example



```
     SM102
0  ──┤ ├──┬──────────────[DEMOV   E1.456   D30    ]
          └──────────────[DEMOV   E2356    R30    ]
      M0
16 ──┤ ├─────────[DECMP   D30      R30      M10    ]
```

Since the floating point number in R30 is greater than the floating point number in D30, M12 turns ON.



M10	0
M11	0
M12	1

DEZCP/Binary floating point bandwidth comparison

DEZCP(P)

Compare the comparison range and data (binary floating point) of high and low 2 points, and output the result of its large, small, and bandwidth to the bit device (3 points).

-[DEZCP (s1) (s2) (s3) (d)]

Content, range and data type

Parameter	Content	Range	Data type	Data type (label)
(s1)	Comparison data or the device number that stores the comparison data	0, 2^{-126} ≤|(s)|< 2^{128}	Single precision real number	ANYREAL_32
(s2)	Comparison data or the device number that stores the comparison data	0, 2^{-126} ≤|(s)|< 2^{128}	Single precision real number	ANYREAL_32
(s3)	Comparison data or the device number that stores the comparison data	0, 2^{-126} ≤|(s)|< 2^{128}	Single precision real number	ANYREAL_32
(d)	The start bit device number that outputs the comparison result (occupies 3 points)	-	Bit	ANYBIT_ARRAY

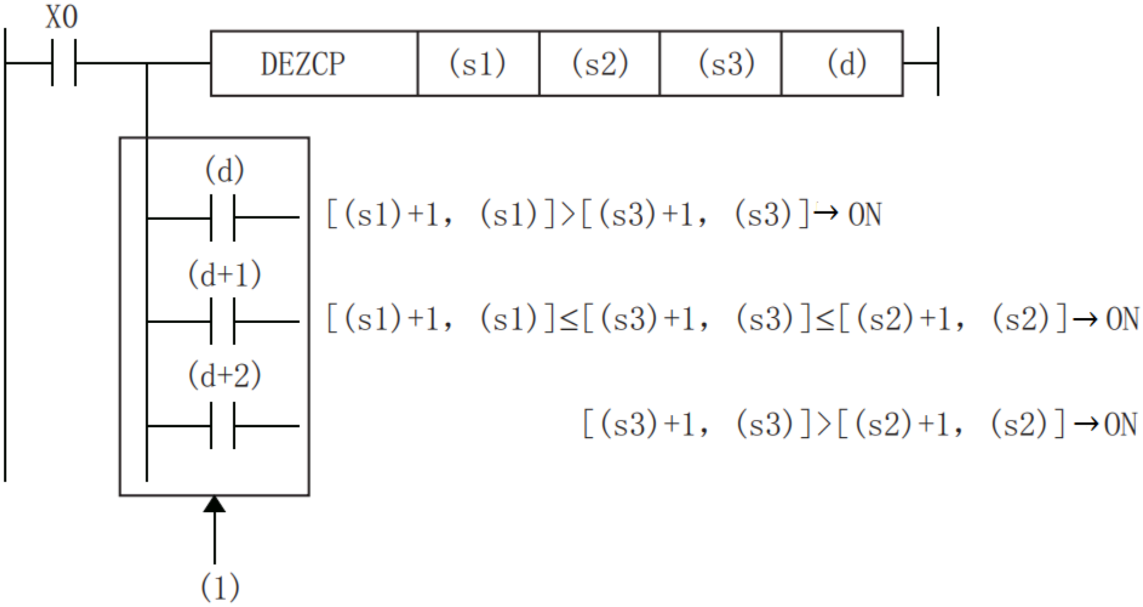
Device used

Instruction	Parameter	Devices															Offset modification	Pulse extension
		Y	M	S	SM	D.b	T	C	D	R	SD	LC	HSC	K	H	E	[D]	XXP
DEZCP	Parameter 1						●	●	●	●	●	●	●	●	●	●	●	●
	Parameter 2						●	●	●	●	●	●	●	●	●	●	●	●
	Parameter 3						●	●	●	●	●	●	●	●	●	●	●	●

Parameter 4	●	●	●	●	●											●	●

Features

Compare the comparison value (s1), (s2) and the comparison source (s3) as a floating point comparison, according to its small, range, and large result, one of (d), (d)+1, (d)+2 The bit turns ON.


X0
DEZCP
(s1)
(s2)
(s3)
(d)
(d)
[(s1)+1, (s1)]>[(s3)+1, (s3)]→ON
(d+1)
[(s1)+1, (s1)]≤[(s3)+1, (s3)]≤[(s2)+1, (s2)]→ON
(d+2)
[(s3)+1, (s3)]>[(s2)+1, (s2)]→ON
(1)

(1): Even if the instruction input is turned OFF and DEZCP instruction is not executed, (d) to (d)+2 will keep the state before X0 is turned OFF.

When the constant (K, H) to the device specified in (s1), (s2), (s3) is specified, the value is automatically converted from BIN to binary floating point for processing.

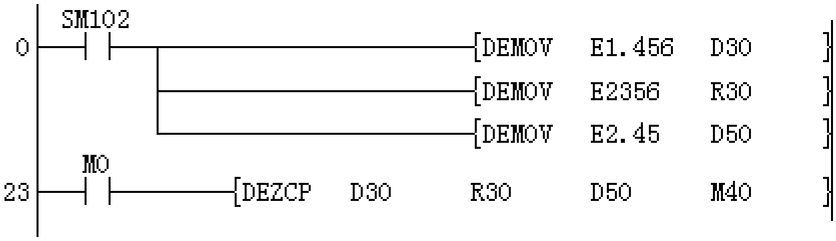
#**Note:** The device specified in (d) occupies 3 points [(d), (d)+1, (d)+2]. Please be careful not to overlap with devices used for other purposes.

Please set the size relationship of the comparison data as [(s1)+1,(s1)]≤[(s2)+1,(s2)]. In the case of [(s1)+1,(s1)]>[(s2)+1,(s2)], it is regarded as the value of [(s2)+1,(s2)] and [(s1)+1, (s1)] Same for comparison.

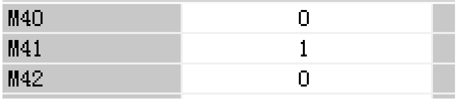
Error code

Error code	Content
4085H	The write address in (s1), (s2) and (s3) exceeds the device range
4086H	The write address in (d) exceeds the device range
4084H	When the content of the device specified by (s1), (s2) and (s3) is an irregular number, a non-number and ±∞

Example



Since 2.45 is greater than 1.456 and 2.45 is less than 2356, M41 is set to ON



M40	0
M41	1
M42	0

DESQR/Single precision real square root

DESQR(P)

After the square root of the value specified in (s) is calculated, the calculation result is stored in the device specified in (d).

-[DESQR (s) (d)]

Content, range and data type

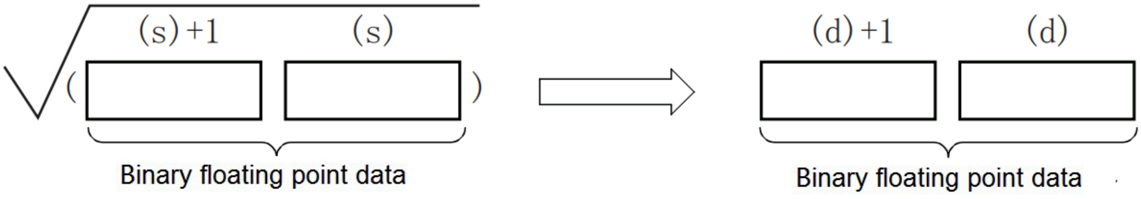
Parameter	Content	Range	Data type	Data type (label)
(s)	The data for square root operation or the device start number that stores the data	$0, 2^{-126} \leq \|(s)\| < 2^{128}$	Single precision real number	ANYREAL_32
(d)	The device start number stores operation result	-	Single precision real number	ANYREAL_32

Device used

Instruction	Parameter	Devices										Offset modification	Pulse extension
		T	C	D	R	SD	LC	HSC	K	H	E	[D]	XXP
DESQR	Parameter 1	●	●	●	●	●	●	●	●	●	●	●	●
	Parameter 2	●	●	●	●	●	●	●				●	●

Features

• After the square root of the value specified in (s) is calculated, the calculation result is stored in the device number specified in (d).



The value specified in (s) can only be set to a positive number. (Cannot perform operations with negative numbers.)

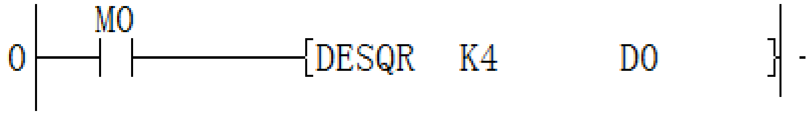
- The related devices are as follows.

Devices	Name	Content	
		Condition	Operation
SM153	Zero	The operation result is zero	The zero flag (SM153) turns ON.

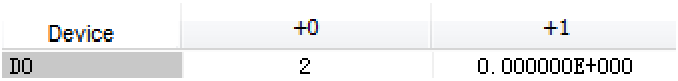
Error code

Error code	Content
4085H	The write address in (s) exceeds the device range
4086H	The write address in (d) exceeds the device range
4084H	When the content of the device specified by (s) is an irregular number, a non-number and ±∞

Example



Get the result: D0 is a floating point number 2



Device	+0	+1
D0	2	0.000000E+000

DESTR/Single precision real number → string conversion

DESTR(P)

Convert the binary floating point data data stored in the device specified in (s1) into a character string according to the display specification stored after the device number specified in (s2), and store it in the device number specified in (d) or later .

-[DESTR (s1) (s2) (d)]

Content, range and data type

Parameter	Content
(s1)	Converted single precision real number data or the device start number that stores the data
(s2)	Display the specified device start number that stores the converted value. The device specified in (s1) is used as the start, and (s2)+2 is used
(d)	Start number of the device storing the converted character string

Device used

Instruction	Parameter	Devices												Offset modification	Pulse extension
		KnX	KnY	KnM	KnS	T	C	D	R	SD	LC	HSC	E	[D]	XXP

DESTR	Parameter 1					●	●	●	●	●	●	●	●	●	●
	Parameter 2	●	●	●	●	●	●	●	●	●				●	●
	Parameter 3		●	●	●	●	●	●	●	●				●	●

Features

Convert the binary floating point data data stored in the device specified in (s1) into a character string according to the display specification stored after the device number specified in (s2), and store it in the device number specified in (d) after. You can also directly specify the real number to (s1).

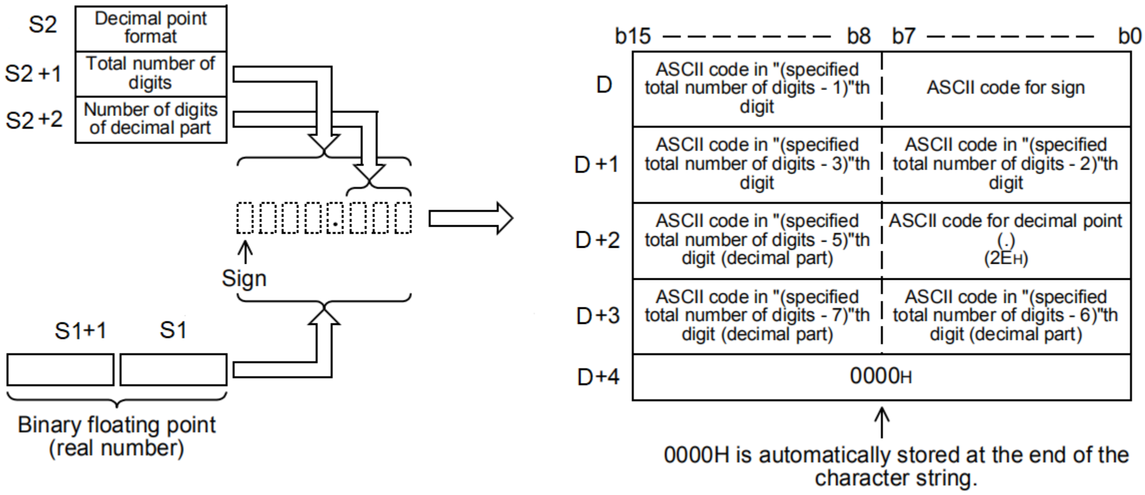
- The converted data differs according to the display specification specified in (s2).

Unit	Features
(s2)	0: Decimal point form 1: Exponential form
(s2)+1	All digits (total number of strings). Range: 2 to 24
(s2)+2	The number of decimal digits. Range: 0 to 7

The range in the above table will change the value range according to the conversion form and other information used

Decimal form

If 0 is specified in (s2), it will be in decimal form.


S2
Decimal point format
S2 +1
Total number of digits
S2 +2
Number of digits of decimal part
Sign
S1+1
S1
Binary floating point (real number)
b15
b8
b7
b0
D
ASCII code in "(specified total number of digits - 1)"th digit
ASCII code for sign
D +1
ASCII code in "(specified total number of digits - 3)"th digit
ASCII code in "(specified total number of digits - 2)"th digit
D +2
ASCII code in "(specified total number of digits - 5)"th digit (decimal part)
ASCII code for decimal point (.) (2EH)
D +3
ASCII code in "(specified total number of digits - 7)"th digit (decimal part)
ASCII code in "(specified total number of digits - 6)"th digit (decimal part)
D +4
0000H
0000H is automatically stored at the end of the character string.

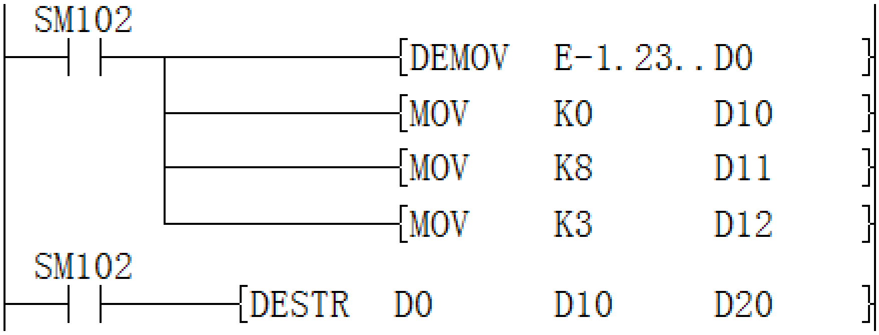
Corresponding digit range in decimal form:

Unit	Features
(s2)	0: Decimal point form
(s2)+1	All digits (total number of strings). Range: 2 to 24. When (s2)+2 is not 0: digits ≥ (number of decimal places + 3).
(s2)+2	The number of decimal places. Range 0 to 7,

When (s2)+2 is not 0: digits ≥ (number of decimal places + 3).

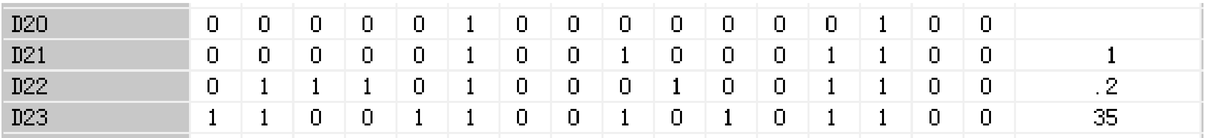
1.Example: The total number of digits is 8, the number of decimal places is 3, and when -1.235 is specified, (d) will be stored in the following way.

When displaying character strings, display character strings in normal order from left to right for convenience.



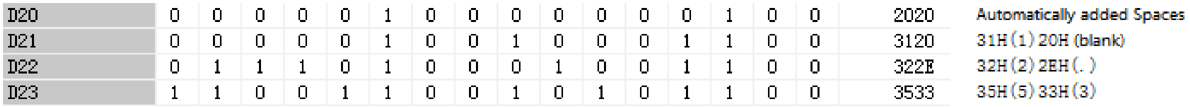
```
SM102
─┤ ├──┬──────────[DEMOV  E-1.23..D0        ]
      ├──────────[MOV    K0        D10       ]
      ├──────────[MOV    K8        D11       ]
      └──────────[MOV    K3        D12       ]
SM102
─┤ ├──────[DESTR  D0     D10       D20       ]
```

Converted string



D20	0	0	0	0	0	1	0	0	0	0	0	0	0	1	0	0	
D21	0	0	0	0	0	1	0	0	1	0	0	0	1	1	0	0	1
D22	0	1	1	1	0	1	0	0	0	1	0	0	1	1	0	0	.2
D23	1	1	0	0	1	1	0	0	1	0	1	0	1	1	0	0	35

The corresponding ASCII code is:



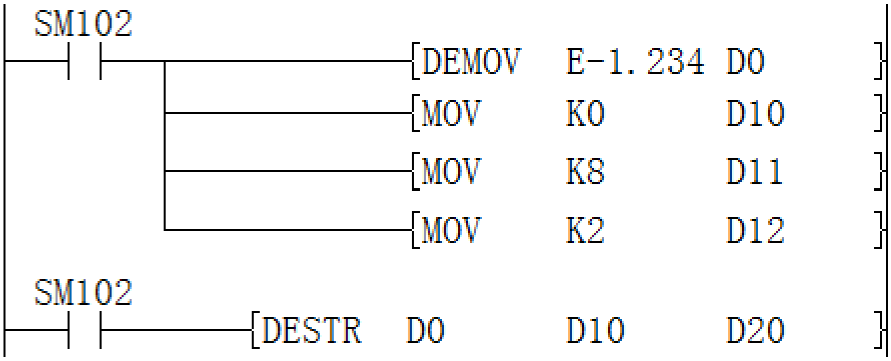
D20	0	0	0	0	0	1	0	0	0	0	0	0	0	1	0	0	2020	Automatically added Spaces
D21	0	0	0	0	0	1	0	0	1	0	0	0	1	1	0	0	3120	31H(1)20H (blank)
D22	0	1	1	1	0	1	0	0	0	1	0	0	1	1	0	0	322E	32H(2)2EH(.)
D23	1	1	0	0	1	1	0	0	1	0	1	0	1	1	0	0	3533	35H(5)33H(3)

The first one is the sign bit. In the sign, when binary floating point data data is positive, 20H (blank) is stored, and when it is negative, 2DH⛔ is stored.

If the actual number of digits is less than all digits during conversion, 20H (blank) will be added between the sign and the first number

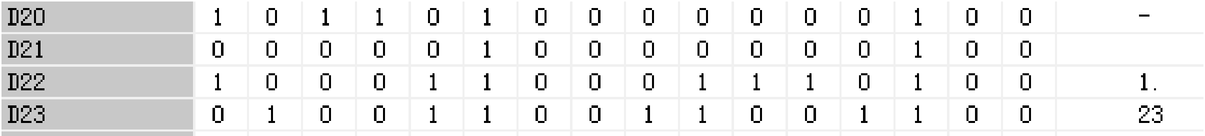
If the decimal part of the binary floating point data data cannot be accommodated in the decimal part, the lower decimal part will be rounded off.

2. Example: The total number of digits is 8, the number of decimal places is 2, and when -1.234 is specified, (d) will be stored in the following way.



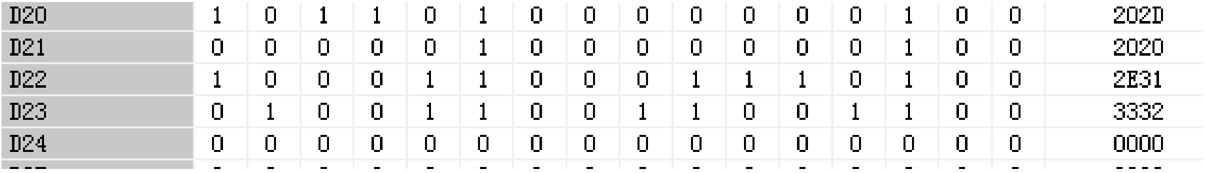
```
SM102
─┤ ├──┬──────────[DEMOV  E-1.234  D0        ]
      ├──────────[MOV    K0       D10       ]
      ├──────────[MOV    K8       D11       ]
      └──────────[MOV    K2       D12       ]
SM102
─┤ ├──────[DESTR  D0     D10      D20       ]
```

The converted string:



D20	1	0	1	1	0	1	0	0	0	0	0	0	0	1	0	0	-
D21	0	0	0	0	0	1	0	0	0	0	0	0	0	1	0	0	
D22	1	0	0	0	1	1	0	0	0	1	1	1	0	1	0	0	1.
D23	0	1	0	0	1	1	0	0	1	1	0	0	1	1	0	0	23

The corresponding ASCII code is:

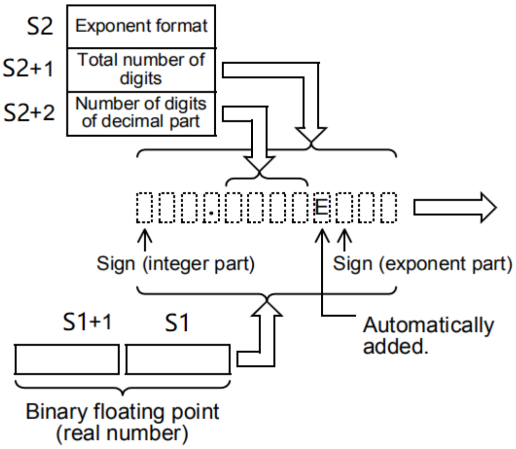


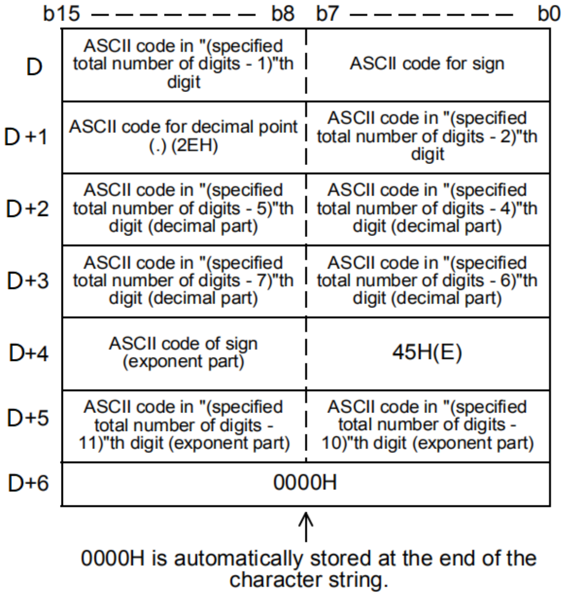
D20	1	0	1	1	0	1	0	0	0	0	0	0	0	1	0	0	202D
D21	0	0	0	0	0	1	0	0	0	0	0	0	0	1	0	0	2020
D22	1	0	0	0	1	1	0	0	0	1	1	1	0	1	0	0	2E31
D23	0	1	0	0	1	1	0	0	1	1	0	0	1	1	0	0	3332
D24	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0000

In the above example: the low byte of D20 stores the negative sign 2DH⛔. Then due to insufficient number of digits, the high byte of D20 and D21 are both 20H (blank). Finally, D22 to D23 store numeric characters 1.23

Exponential form

When 1 is specified in (s2), it will be in exponential format.





	b15 ——— b8	b7 ——— b0
D	ASCII code in "(specified total number of digits - 1)"th digit	ASCII code for sign
D+1	ASCII code for decimal point (.) (2EH)	ASCII code in "(specified total number of digits - 2)"th digit
D+2	ASCII code in "(specified total number of digits - 5)"th digit (decimal part)	ASCII code in "(specified total number of digits - 4)"th digit (decimal part)
D+3	ASCII code in "(specified total number of digits - 7)"th digit (decimal part)	ASCII code in "(specified total number of digits - 6)"th digit (decimal part)
D+4	ASCII code of sign (exponent part)	45H(E)
D+5	ASCII code in "(specified total number of digits - 11)"th digit (exponent part)	ASCII code in "(specified total number of digits - 10)"th digit (exponent part)
D+6	0000H	

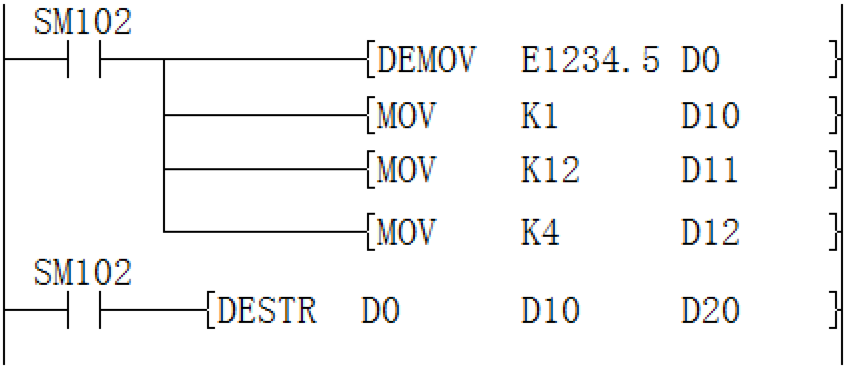
0000H is automatically stored at the end of the character string.

The corresponding digit range in exponential form:

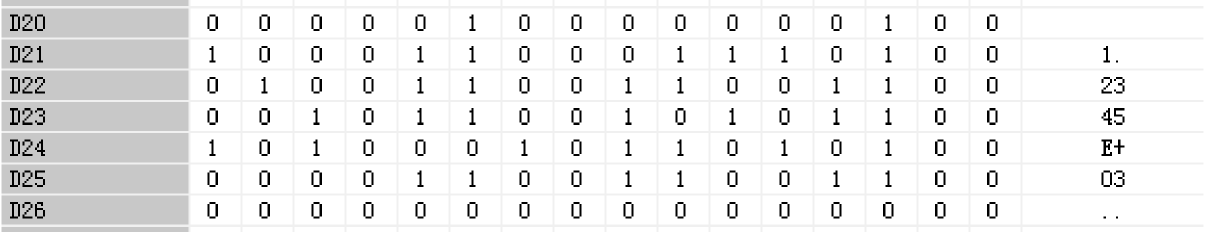
Unit	Features
(s2)	1: Exponential form
(s2)+1	All digits (total number of strings). Range: 2 to 24. (s2)+2 when non-zero: digits ≥ (number of decimal places + 7)

(s2)+2	The number of decimal places. Range 0 to 7 (s2)+2 when non-zero: digits ≥ (number of decimal places + 7)

For example 3, all digits are 12, decimal place is 4, and 1234.5 is specified, (d) and later will be stored as follows.

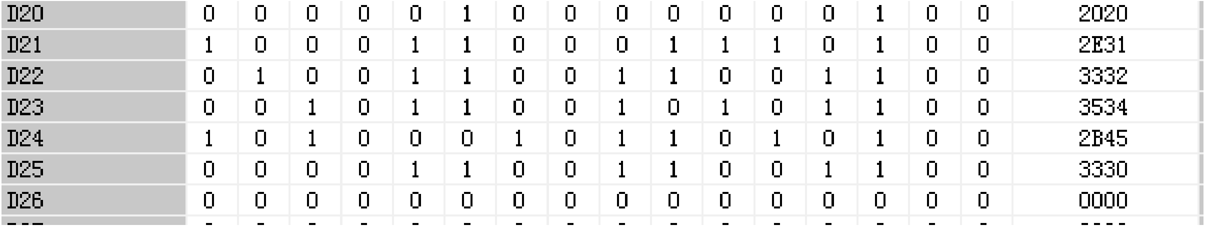


The converted string:



D20	0	0	0	0	0	1	0	0	0	0	0	0	0	1	0	0	
D21	1	0	0	0	1	1	0	0	0	1	1	1	0	1	0	0	1.
D22	0	1	0	0	1	1	0	0	1	1	0	0	1	1	0	0	23
D23	0	0	1	0	1	1	0	0	1	0	1	0	1	1	0	0	45
D24	1	0	1	0	0	0	1	0	1	1	0	1	0	1	0	0	E+
D25	0	0	0	0	1	1	0	0	1	1	0	0	1	1	0	0	03
D26	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	. .

The corresponding ASCII code is:



D20	0	0	0	0	0	1	0	0	0	0	0	0	0	1	0	0	2020
D21	1	0	0	0	1	1	0	0	0	1	1	1	0	1	0	0	2E31
D22	0	1	0	0	1	1	0	0	1	1	0	0	1	1	0	0	3332
D23	0	0	1	0	1	1	0	0	1	0	1	0	1	1	0	0	3534
D24	1	0	1	0	0	0	1	0	1	1	0	1	0	1	0	0	2B45
D25	0	0	0	0	1	1	0	0	1	1	0	0	1	1	0	0	3330
D26	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0000

In the sign of the integer part, when the binary floating point data data is positive, 20H (blank) is stored, and when it is negative, 2DH⊖ is stored.

The integer part is fixed to 1 digit. 20H (blank) is stored between the integer part and the Sign.

If the decimal part of the binary floating point data data cannot be accommodated in the decimal part, the lower decimal part will be rounded off.

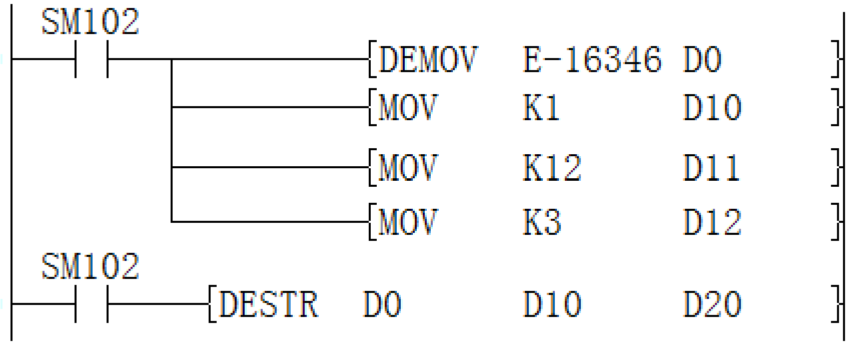
When the number of decimal places is set to other than 0, 2EH(.) is automatically stored in the number of specified decimal places+1 digit. When the decimal place is 0, 2EH(.) is not stored.

In the sign of the exponent, 2BH⊕ is stored when the exponent is positive, and 2DH⊖ is stored when it is negative.

The exponent is fixed to 2 digits. When the exponent part is a 1-digit number, 30H(0) is stored between the signs of the exponent part.

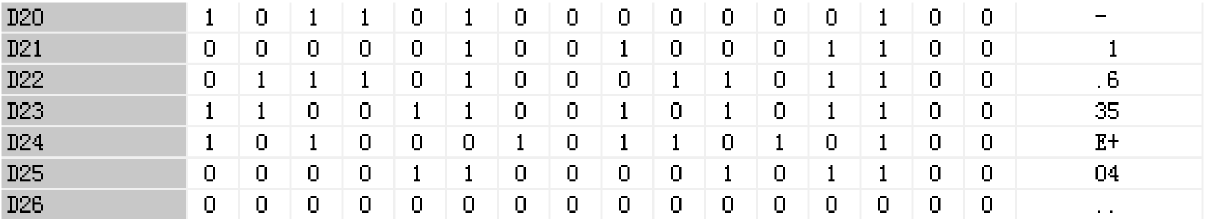
00H is automatically stored at the end of the converted character string.

Example 4: All digits are 12, decimal places are 3, and -16346 is specified, (d) will be stored in the following way.



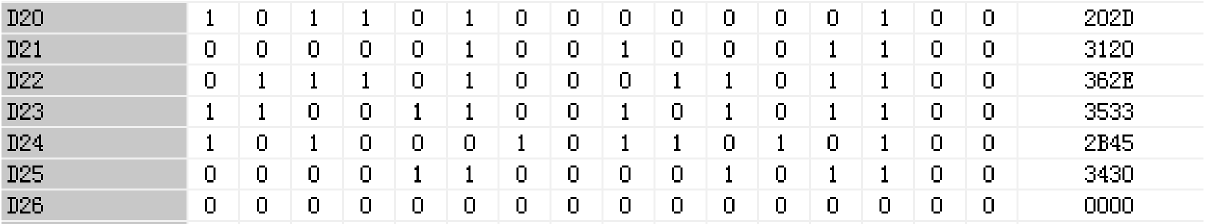
```
SM102
─┤ ├─┬─[DEMOV  E-16346  D0   ]
     ├─[MOV    K1       D10  ]
     ├─[MOV    K12      D11  ]
     └─[MOV    K3       D12  ]
SM102
─┤ ├───[DESTR  D0   D10   D20 ]
```

The converted string:



D20	1	0	1	1	0	1	0	0	0	0	0	0	0	1	0	0	-
D21	0	0	0	0	0	1	0	0	1	0	0	0	1	1	0	0	1
D22	0	1	1	1	0	1	0	0	0	1	1	0	1	1	0	0	.6
D23	1	1	0	0	1	1	0	0	1	0	1	0	1	1	0	0	35
D24	1	0	1	0	0	0	1	0	1	1	0	1	0	1	0	0	E+
D25	0	0	0	0	1	1	0	0	0	0	1	0	1	1	0	0	04
D26	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	..

The corresponding ASCII code is:



D20	1	0	1	1	0	1	0	0	0	0	0	0	0	1	0	0	202D
D21	0	0	0	0	0	1	0	0	1	0	0	0	1	1	0	0	3120
D22	0	1	1	1	0	1	0	0	0	1	1	0	1	1	0	0	362E
D23	1	1	0	0	1	1	0	0	1	0	1	0	1	1	0	0	3533
D24	1	0	1	0	0	0	1	0	1	1	0	1	0	1	0	0	2B45
D25	0	0	0	0	1	1	0	0	0	0	1	0	1	1	0	0	3430
D26	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0000

As in the above example:

The low byte of D20 stores the negative sign 2DH.

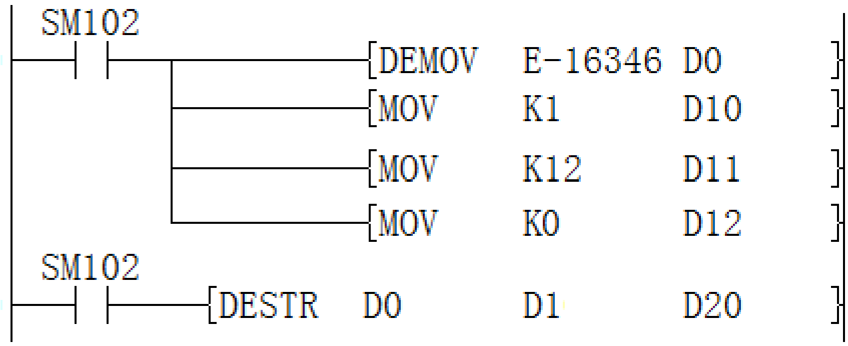
Then due to insufficient number of digits, the high byte of D20 and the low bit of D21 are both 20H (blank).

16346 becomes the string 1.635E+04, in which the last digit "6" of 16346 is rounded.

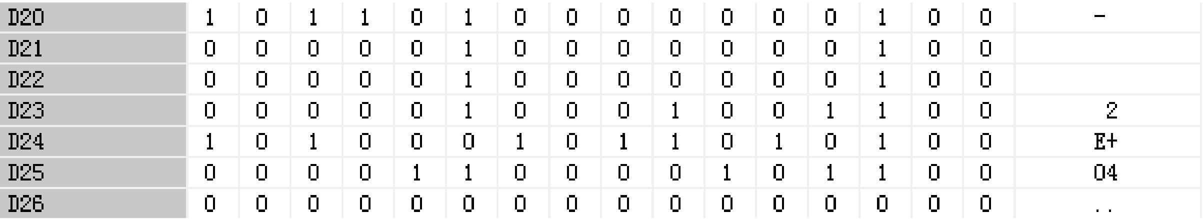
The exponent part is 34H(4) with only one bit, then add 30H(0) between the Signs 2DH and 34H(4).

Finally D26 automatically stores 00H

Example 5: All digits are 12, and the number of decimal places is 0. If -16346 is specified, (d) will be stored as follows.

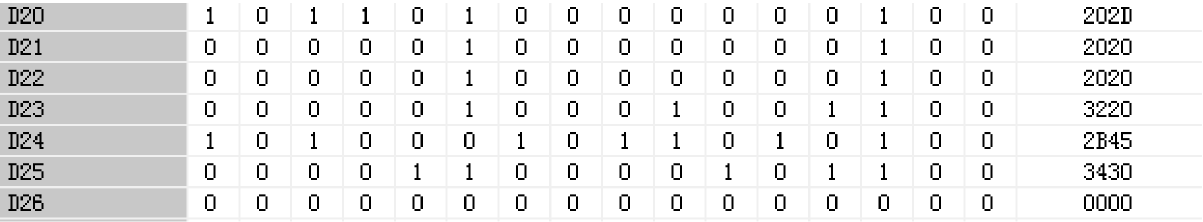


The converted string:



D20	1	0	1	1	0	1	0	0	0	0	0	0	0	1	0	0	-
D21	0	0	0	0	0	1	0	0	0	0	0	0	0	1	0	0	
D22	0	0	0	0	0	1	0	0	0	0	0	0	0	1	0	0	
D23	0	0	0	0	0	1	0	0	0	1	0	0	1	1	0	0	2
D24	1	0	1	0	0	0	1	0	1	1	0	1	0	1	0	0	E+
D25	0	0	0	0	1	1	0	0	0	0	1	0	1	1	0	0	04
D26	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	. .

The corresponding ASCII code is:



D20	1	0	1	1	0	1	0	0	0	0	0	0	0	1	0	0	202D
D21	0	0	0	0	0	1	0	0	0	0	0	0	0	1	0	0	2020
D22	0	0	0	0	0	1	0	0	0	0	0	0	0	1	0	0	2020
D23	0	0	0	0	0	1	0	0	0	1	0	0	1	1	0	0	3220
D24	1	0	1	0	0	0	1	0	1	1	0	1	0	1	0	0	2B45
D25	0	0	0	0	1	1	0	0	0	0	1	0	1	1	0	0	3430
D26	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0000

This example mainly shows that if the decimal place is set to 0, the decimal point 2EH(.) will be automatically omitted.

#**Note:** When the binary floating point data is converted, the more digits, the lower the accuracy of the digits, the worse the accuracy of the digits, and the conversion value may be inaccurate due to the progress.

Error code

Error code	Content
4085H	The read address of (s1) and (s2) exceeds the device range
4086H	The write address of (d) exceeds the device range
4084H	When the content of the specified device (s1) and (s2) is an irregular number, a non-number, or ±∞
	When the format specified in (s2) is other than 0 or 1
	When all the digits specified in (s1) +1 exceeds the value of 24
	When the number of decimal places specified in (s2) +2 exceeds the range of 0 to 7
	In the decimal form, when (s2) is 0. 1. When the number of decimal places is 0: [(s2)+1]<2 2. When the number of decimal places is other than 0: [(s2)+1]<(number of decimal places+3)
	In the exponential form, when (s2) is 0. 1. When the number of decimal places is 0: [(s2)+1]<6 2. When the number of decimal places is other than 0: [(s2)+1]<(number of decimal places+7)

DEVAL/String → single precision real number conversion

DEVAL(P)

The character string stored in the device number specified in (s) and later is converted to a binary floating point data, and then stored in the device specified in (d).

-[DEVAL (s) (d)]

Content, range and data type

Parameter	Content	Range	Data type	Data type (label)
(s)	String data for single precision real number or the device start number that stores the string data	-	String	ANYSTRING_SINGLE
(d)	The device start number that stores the converted single precision real number	-	Single precision real number	ANYREAL_32

Device used

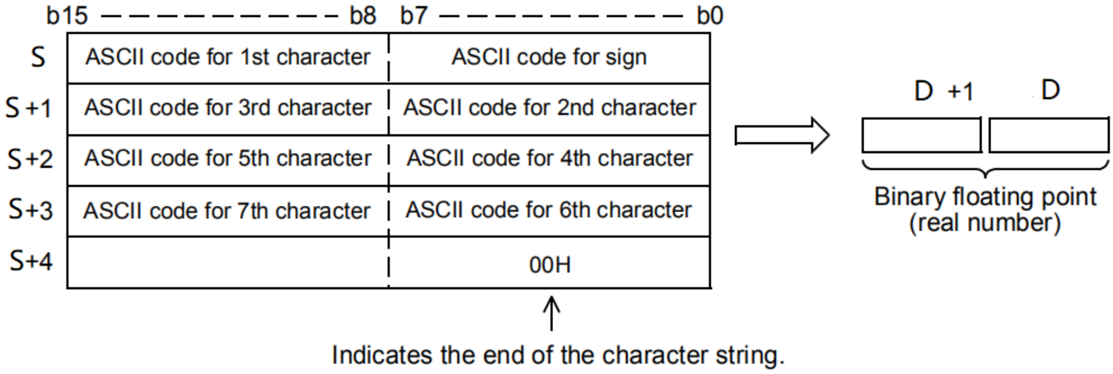
Instruction	Parameter	Devices											Offset modification	Pulse extension
		KnX	KnY	KnM	KnS	T	C	D	R	SD	LC	HSC	[D]	XXP
DEVAL	Parameter 1	●	●	●	●	●	●	●	●	●			●	●
	Parameter 2					●	●	●	●	●	●	●	●	●

Features

The character string stored in the device number specified in (s) and later is converted to a binary floating point data, and then stored in the device specified in (d).

Whether the specified string is in decimal form or exponential form, it can be converted to a binary floating point data.

Up to 24 characters can be set for the string. 20H (blank) and 30H (0) in the character string are also counted as 1 character.



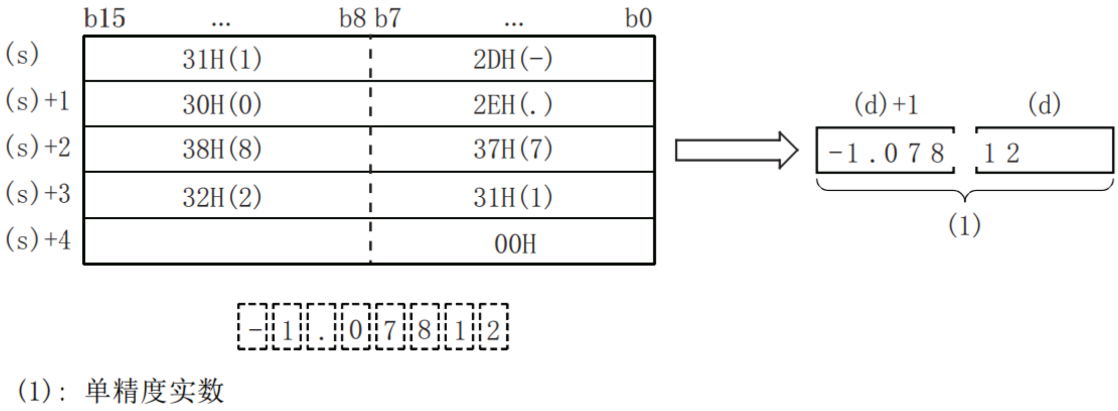
	b15 ——————— b8	b7 ——————— b0
S	ASCII code for 1st character	ASCII code for sign
S +1	ASCII code for 3rd character	ASCII code for 2nd character
S+2	ASCII code for 5th character	ASCII code for 4th character
S+3	ASCII code for 7th character	ASCII code for 6th character
S+4		00H

Indicates the end of the character string.

(1) Decimal form

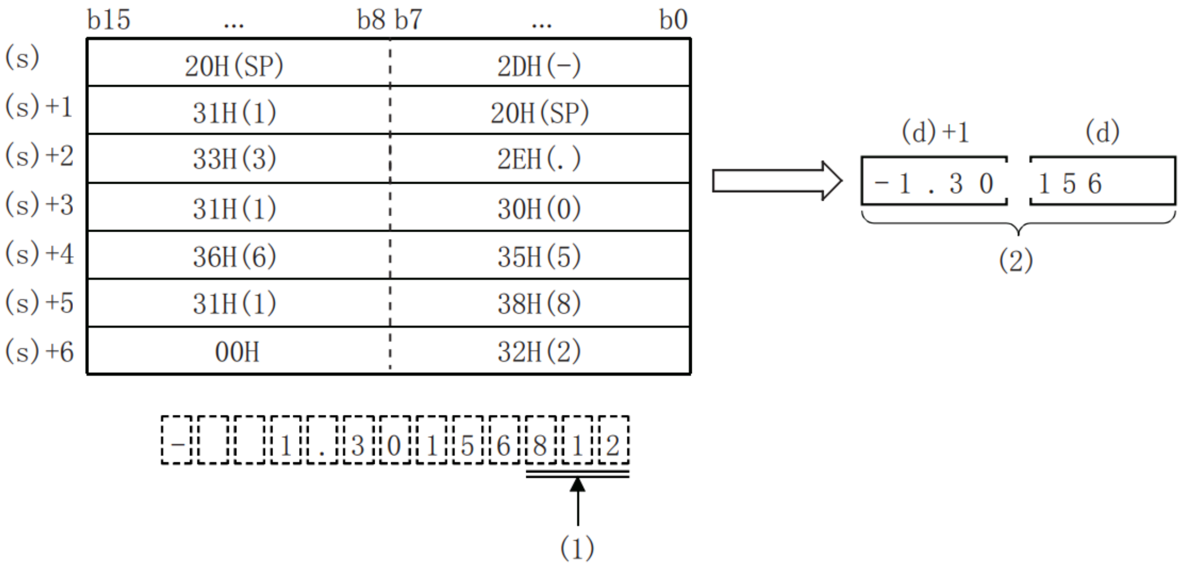
1) When the character string specified in (s) is in decimal format, the following is the case.

(1): single-precision real number



2) In the character string specified in (s), for the character string to be converted to a binary floating point data, the 6 digits after the sign, decimal point, and exponent are valid, and the 7th digit and later will be discarded during conversion.

(1): discarded
(2): single-precision real number



	b15 … b8	b7 … b0
(s)	20H(SP)	2DH(−)
(s)+1	31H(1)	20H(SP)
(s)+2	33H(3)	2EH(.)
(s)+3	31H(1)	30H(0)
(s)+4	36H(6)	35H(5)
(s)+5	31H(1)	38H(8)
(s)+6	00H	32H(2)

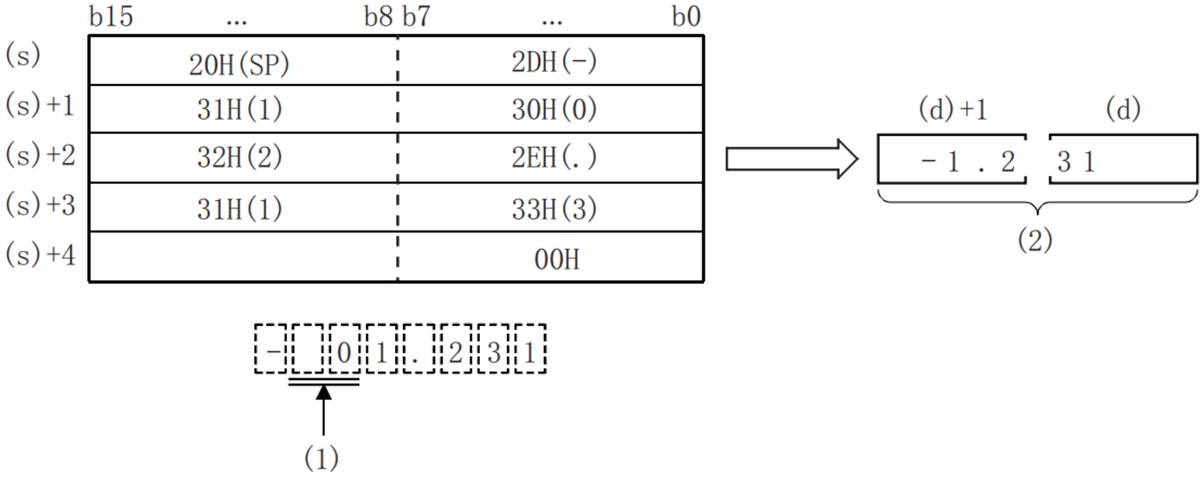
(1)： 被舍去。
(2)： 单精度实数

When the sign is specified as 2BH ⊕ or omitted in the decimal point format, it will be converted as a positive value. In addition, when the sign is specified as 2DH ⊖, it will be converted as a negative value.

3) If there are 20H (blank) or 30H (0) in the character string specified in (s) other than the first 0, 20H and 30H will be ignored during conversion.

(1): ignore

(2): single-precision real number



	b15 … b8	b7 … b0
(s)	20H(SP)	2DH(−)
(s)+1	31H(1)	30H(0)
(s)+2	32H(2)	2EH(.)
(s)+3	31H(1)	33H(3)
(s)+4		00H

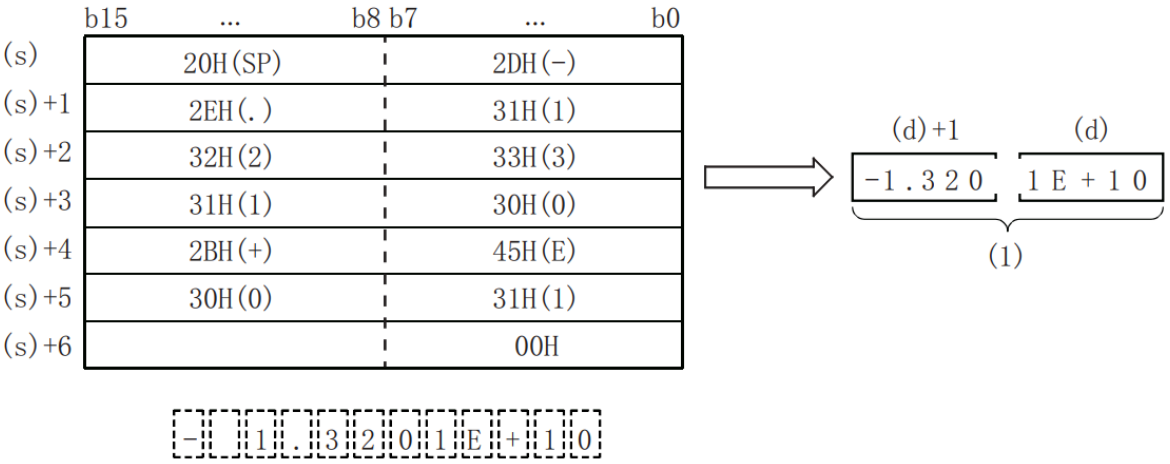
(1)： 忽略。

(2)： 单精度实数

(2) In the case of exponential form

1) When the character string specified in (s) is in exponential form, it is executed as follows.

(1): single-precision real number



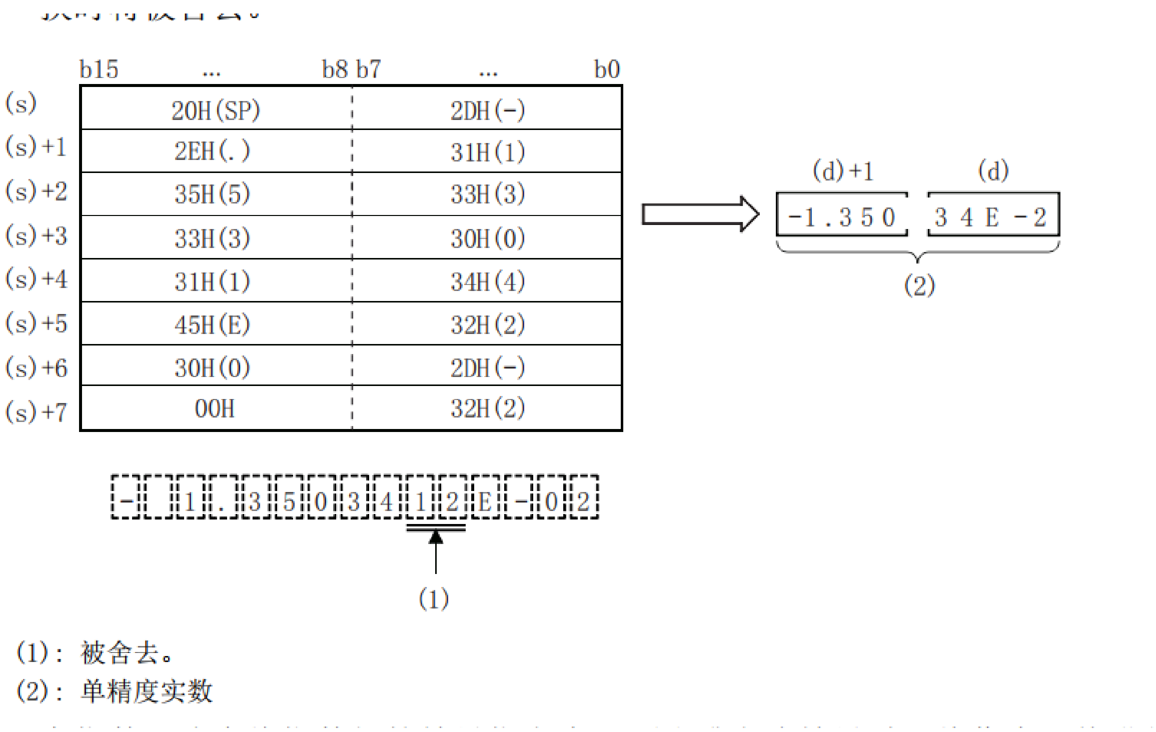
	b15 … b8	b7 … b0
(s)	20H(SP)	2DH(−)
(s)+1	2EH(.)	31H(1)
(s)+2	32H(2)	33H(3)
(s)+3	31H(1)	30H(0)
(s)+4	2BH(+)	45H(E)
(s)+5	30H(0)	31H(1)
(s)+6		00H

(1)： 单精度实数

2) Among the character strings specified in (s), for the character string to be converted to a binary floating point data, the 6 digits after the sign, decimal point, and exponent are valid, and the 7th digit and later will be discarded during conversion.

(1): discarded

(2): single-precision real number



换时将被舍去。

	b15 … b8	b7 … b0
(s)	20H(SP)	2DH(-)
(s)+1	2EH(.)	31H(1)
(s)+2	35H(5)	33H(3)
(s)+3	33H(3)	30H(0)
(s)+4	31H(1)	34H(4)
(s)+5	45H(E)	32H(2)
(s)+6	30H(0)	2DH(-)
(s)+7	00H	32H(2)

- 1 . 3 5 0 3 4 1 2 E - 0 2

(1)：被舍去。
(2)：单精度实数

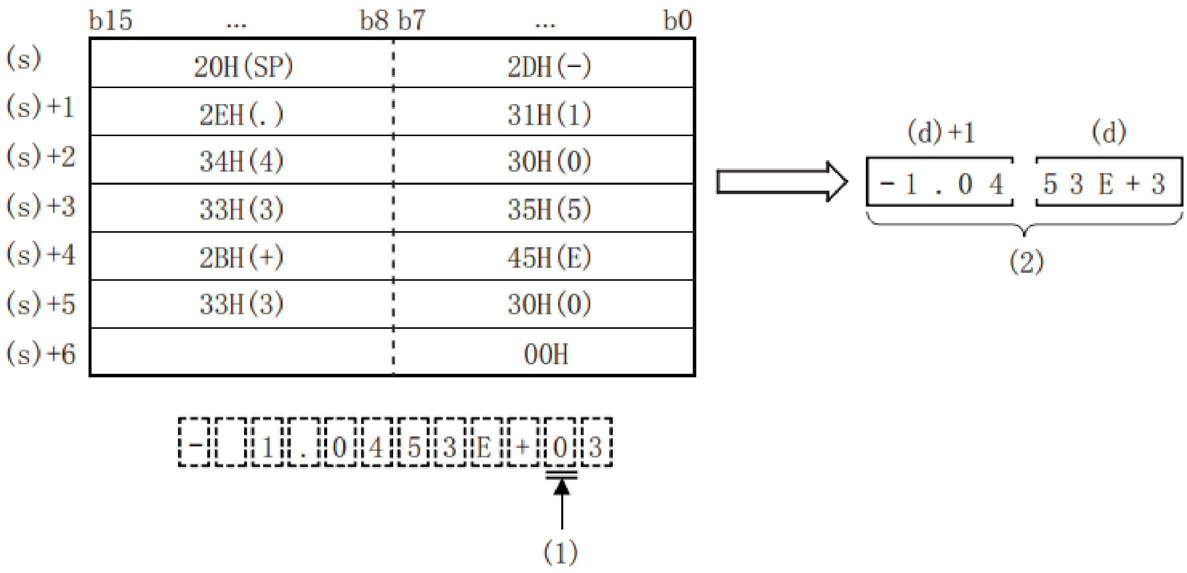
If the sign of the exponent part is specified as 2BH⊕ or omitted in the exponential form, it will be converted as a positive value. When the sign of the exponent is specified as 2DH⊖, it will be converted as a negative value.

3) If there is 20H (blank) or 30H (0) in the character string specified in (s) other than the first 0, 20H and 30H will be ignored during conversion.

In the exponential character string, if 30H (0) is stored between "E" and the value, 30H will be ignored during conversion.

(1): Ignore.

(2): Single precision real number.



	b15 … b8	b7 … b0
(s)	20H(SP)	2DH(−)
(s)+1	2EH(.)	31H(1)
(s)+2	34H(4)	30H(0)
(s)+3	33H(3)	35H(5)
(s)+4	2BH(+)	45H(E)
(s)+5	33H(3)	30H(0)
(s)+6		00H

(1)：忽略。
(2)：单精度实数

The related devices are shown below.

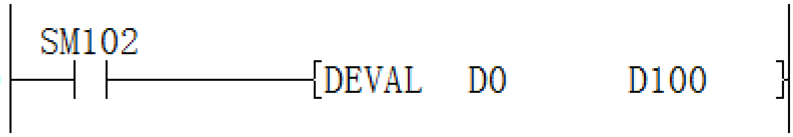
Devices	Name	Content	
		Condition	Operating
SM153	Zero	The operation result is zero	The zero flag (SM153) turns ON.
SM152	Borrow	The absolute value of operation result $<2^{-126}$	The value of (d) becomes the minimum value of a 32-bit real number (2^{-126}), and the borrow flag (SM152) turns on.
SM151	Carry	The absolute value of operation result$> 2^{128}$	The value of (d) becomes the maximum value of 32-bit real numbers (2^{128}), and the carry flag (SM151) turns on.

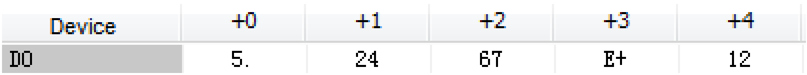
Error code

Error code	Content
4085H	The read address of (s) exceeds the device range
4086H	The write address of (d) exceeds the device range
408AH	The string is not read by (s), or the string length exceeds 24
408BH	When (s) reading a character string, the maximum range of the device is read, but 00H is not found and the end
4084H	When there are characters other than 2BH ⊕, 2DH ⊖, 20H (space), 2EH (.), 45H (E), 65H(e), and 30 H(0) to 39H (9) in the string specified in (s) When there are two or more 2EH(.) characters in the character string specified in (s). When there are characters other than 45H(E), 2BH ⊕, 2DH ⊖, and 30 H(0) to 39H (9) in the exponent part specified

in (s), or if there are multiple exponent parts, or exponent In some cases, 2BH ⊕ or 2DH ⊖ occurred twice or more. 2BH ⊕ or 2DH ⊖ appears twice or more before the first digit of the string specified in (s).

Example

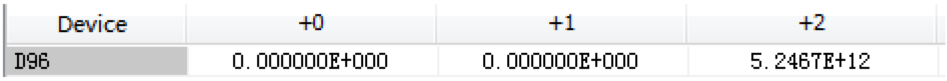




Device	+0	+1	+2	+3	+4
D0	5.	24	67	E+	12

The stored character string of D0 is: 5.2467E+12

The resulting floating point number is: 5.2467E+12



Device	+0	+1	+2
D96	0.000000E+000	0.000000E+000	5.2467E+12

DEXP/Single precision real number exponential operation

DEXP(P)

After performing the exponential calculation of the value specified in (s), the calculation result is stored in the device specified in (d).

-[DEXP (s) (d)]

Content, range and data type

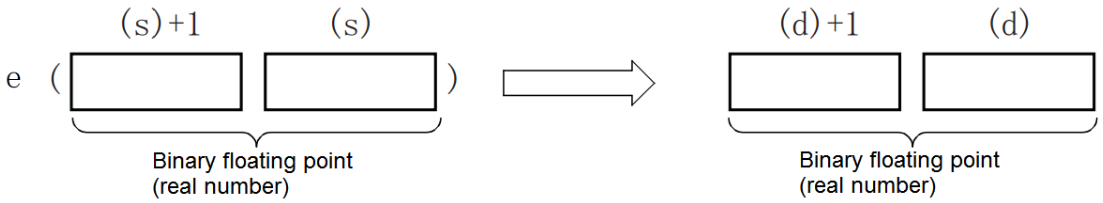
Parameter	Content	Range	Data type	Data type (label)
(s)	Data for exponential calculation or the device start number that stores the data	0, 2^{-126} ≤|(s)|< 2^{128}	Single precision real number	ANYREAL_32
(d)	The device start number that stores the operation result	-	Single precision real number	ANYREAL_32

Device used

Instruction	Parameter	Devices								Offset modification	Pulse extension
		T	C	D	R	SD	LC	HSC	E	[D]	XXP
DEXP	Parameter 1	●	●	●	●	●	●	●	●	●	●
	Parameter 2	●	●	●	●	●	●	●		●	●

Features

After performing the exponential calculation of the value specified in (s), the calculation result is stored in the device number specified in (d).


(s)+1
(s)
e
(d)+1
(d)
Binary floating point (real number)
Binary floating point (real number)

In exponential calculation, the base (e) is calculated as "2.71828".

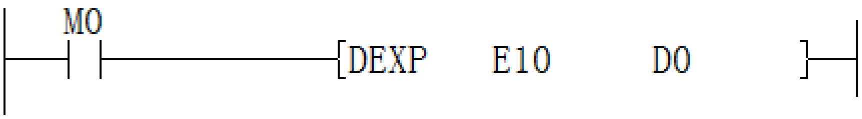
The related devices are shown below.

Devices	Name	Content	
		Condition	**Operating**
SM153	Zero	The operation result is zero	The zero flag (SM153) turns ON.
SM152	Borrow	The absolute value of operation result <2^{-126}	The value of (d) becomes the minimum value of a 32-bit real number (2^{-126}), and the borrow flag (SM152) turns on.
SM151	Carry	The absolute value of operation result> 2^{128}	The value of (d) becomes the maximum value of 32-bit real numbers (2^{128}), and the carry flag (SM151) turns on.

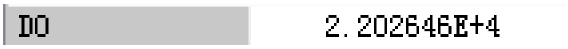
Error code

Error code	Content
4085H	The write address in (s) exceeds the device range
4086H	The write address in (d) exceeds the device range
4084H	When the content of the device specified by (s) is an irregular number, a non-number and ±∞

Example



Calculate the result:



```
D0        2.202646E+4
```

INT/Single precision real number → signed BIN 16-bit data

INT(P)

Convert the specified single precision real number into signed BIN 16-bit data.

-[INT (s) (d)]

Content, range and data type

Parameter	Content	Range	Data type	Data type (label)
(s)	Single precision real number or the start	-32768 to 32767	Single precision real number	ANYREAL_32

	device storing single precision real number			
(d)	Signed device for storing BIN data	-	BIN16 bit	ANY16_S

Device used

Instruction	Parameter	Devices							Offset modification	Pulse extension
		T	C	D	R	SD	LC	HSC	[D]	XXP
INT	Parameter 1	●	●	●	●	●	●	●	●	●
	Parameter 2	●	●	●	●	●			●	●

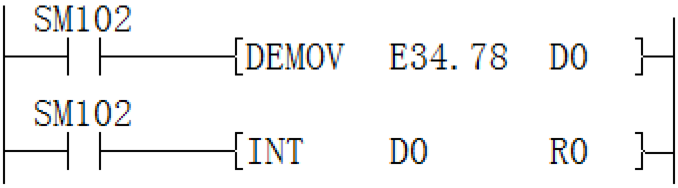
Features

- Convert the single precision real number specified in (s) into signed BIN 16-bit data and store it in the device specified in (d).
- The converted data will be rounded to the first digit below the decimal point of the single precision real number specified in (s).
- When setting the input value with the engineering tool, rounding errors may occur.
- The related devices are as follows.

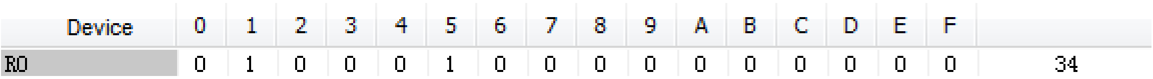
Devices	Name	Content	
		Condition	Operation
SM153	Zero	The operation result is zero	The zero flag (SM153) turns ON.
SM152	Borrow	Decimal places are rounded off when converting	During conversion ((s) – (d))> (2 $^{-126}$) ,borrow (SM152) turns ON
SM151	Carry	Conversion result is out of range	The value of (s) is out of the range -32768 to 32767 or the value of (s) is less than the minimum value of 32-bit real numbers (2 $^{-126}$), and the carry flag (SM151) turns on.

Error code

Error code	Content
4085H	The write address in (s) exceeds the device range
4086H	The write address in (d) exceeds the device range
4084H	When the content of the device specified by (s) is an irregular number, a non-number and ±∞

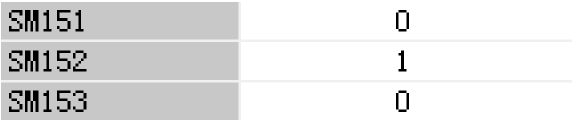


Get the conversion result:



Device	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F	
R0	0	1	0	0	0	1	0	0	0	0	0	0	0	0	0	0	34

And the borrow means turn ON



SM151	0
SM152	1
SM153	0

DINT/Single precision real number→ signed BIN 32-bit data

DINT(P)

Convert the specified single precision real number into signed BIN 32-bit data.

-[DINT (s) (d)]

Content, range and data type

Parameter	Content	Range	Data type	Data type (label)
(s)	Single precision real number or the start device storing single precision real number	-2147483648 to 2147483647	Single precision real number	ANYREAL_32
(d)	The start device storing BIN data	-	Signed BIN 32 bit	ANY32_S

Device used

Instruction	Parameter	Devices							Offset modification	Pulse extension
		T	C	D	R	SD	LC	HSC	[D]	XXP
DINT	Parameter 1	●	●	●	●	●	●	●	●	●
	Parameter 2	●	●	●	●	●	●	●	●	●

Features

• Convert the binary floating point data specified in (s) into signed BIN 32-bit data and store it in the device specified in (d).

• The converted data will be rounded to the first digit below the decimal point of the binary floating point data specified in (s).

• When setting the input value with the engineering tool, rounding errors may occur.

• The related devices are as follows.

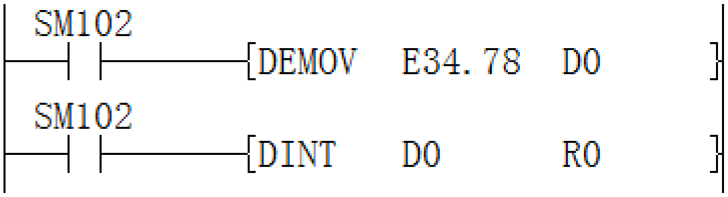
Devices	Name	Content	
		Condition	Operating
SM153	Zero	The operation result is zero	The zero flag (SM153) turns ON.
SM152	Borrow	The absolute value of operation result $<2^{-126}$	The value of (d) becomes the minimum value of a 32-bit real number (2^{-126}), and the borrow flag (SM152) turns on.
SM151	Carry	The absolute value of operation result$> 2^{128}$	The value of (d) becomes the maximum value of 32-bit real

numbers (2^{128}), and the carry flag (SM151) turns on.

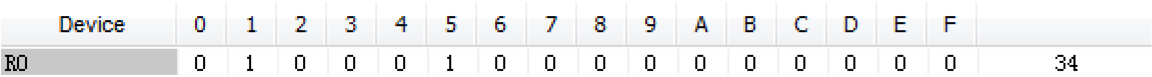
Error code

Error code	Content
4085H	The write address in (s) exceeds the device range
4086H	The write address in (d) exceeds the device range
4084H	When the content of the device specified by (s) is an irregular number, a non-number and ±∞

Example

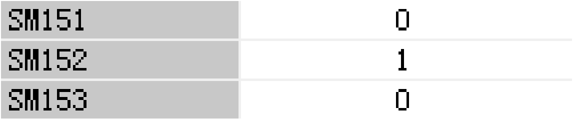


Get the conversion result:



Device	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F	
R0	0	1	0	0	0	1	0	0	0	0	0	0	0	0	0	0	34

And the borrow means turn ON



SM151	0
SM152	1
SM153	0

DLOG10/Single precision real number common logarithmic operation

DLOG10(P)

Calculate the common logarithm (base 10 logarithm) of the value specified in (s), and store the result of the operation in the device specified in (d).

-[DLOG10 (s) (d)]

Content, range and data type

Parameter	Content	Range	Data type	Data type (label)
(s)	Data for common logarithmic operations or the device start number storing the data	0, 2^{-126} ≤|(s)|< 2^{128}	Single precision real number	ANYREAL_32
(d)	The device start number storing operation result	-	Single precision real number	ANYREAL_32

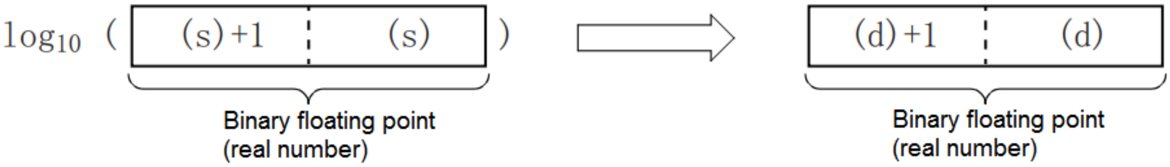
Device used

Instruction	Parameter	Devices								Offset modification	Pulse extension
		T	C	D	R	SD	LC	HSC	E	[D]	XXP
DLOG10	Parameter 1	●	●	●	●	●	●	●	●	●	●

Parameter 2	●	●	●	●	●	●	●		●	●

Features

Calculate the common logarithm (base 10 logarithm) of the value specified in (s), and store the result of the calculation in the device number specified in (d).


log10
(s)+1
(s)
(d)+1
(d)
Binary floating point (real number)
Binary floating point (real number)

The value specified in (s) can only be set to a positive number. (Cannot perform operations with negative numbers.)

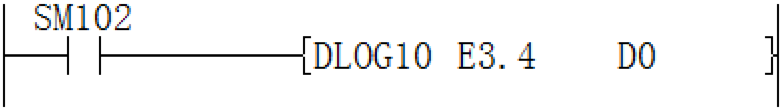
- The related devices are as follows.

Devices	Name	Content	
		Condition	**Operating**
SM153	Zero	The operation result is zero	The zero flag (SM153) turns ON.
SM152	Borrow	The absolute value of operation result $<2^{-126}$	The value of (d) becomes the minimum value of a 32-bit real number (2^{-126}), and the borrow flag (SM152) turns on.
SM151	Carry	The absolute value of operation result> 2^{128}	The value of (d) becomes the maximum value of 32-bit real numbers (2^{128}), and the carry flag (SM151) turns on.

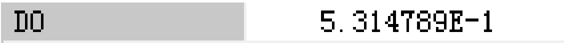
Error code

Error code	Content
4085H	The write address in (s) exceeds the device range
4086H	The write address in (d) exceeds the device range
4084H	When the content of the device specified by (s) is an irregular number, a non-number and ±∞

Example



Get calculation results



D0 5.314789E-1

DLOGE/Single precision real number natural logarithm operation

DLOGE(P)

After calculating the logarithm when the natural logarithm e of the value specified in (s) is the base, store the calculation result in the device specified in (d).

-[DLOGE (s) (d)]

Content, range and data type

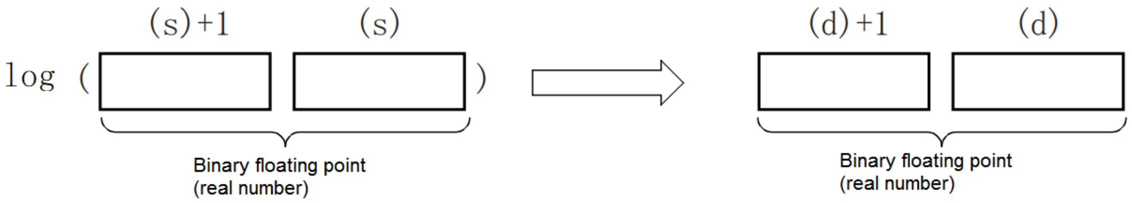
Parameter	Content	Range	Data type	Data type (label)
(s)	Data for logarithm operation or the device start number storing the data	0, $2^{-126} \leq \lvert(s)\rvert < 2^{128}$	Single precision real number	ANYREAL_32
(d)	the device start number storing operation result	-	Single precision real number	ANYREAL_32

Device used

Instruction	Parameter	Devices								Offset modification	Pulse extension
		T	C	D	R	SD	LC	HSC	E	[D]	XXP
DLOGE	Parameter 1	●	●	●	●	●	●	●	●	●	●
	Parameter 2	●	●	●	●	●	●	●		●	●

Features

• After calculating the logarithm when the natural logarithm e of the value specified in (s) is the base, store the result of the calculation in the device number specified in (d).



• The value specified in (s) can only be set to a positive number. (Cannot perform operations with negative numbers.)

• The related devices are as follows.

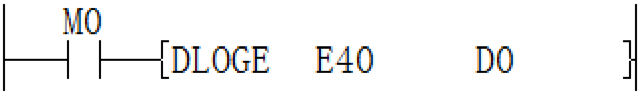
Devices	Name	Content	
		Condition	Operating
SM153	Zero	The operation result is zero	The zero flag (SM153) turns ON.
SM152	Borrow	The absolute value of operation result $<2^{-126}$	The value of (d) becomes the minimum value of a 32-bit real number (2^{-126}), and the borrow flag (SM152) turns on.
SM151	Carry	The absolute value of operation result$> 2^{128}$	The value of (d) becomes the maximum value of 32-bit real numbers (2^{128}), and the carry flag (SM151) turns on.

Error code

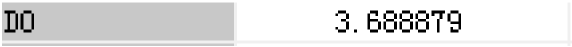
Error code	Content
4085H	The write address in (s) exceeds the device range
4086H	The write address in (d) exceeds the device range

4084H	When the content of the device specified by (s) is an irregular number, a non-number and ±∞

Example



The result is as below:



D0	3.688879