Converting single-precision real number to character string
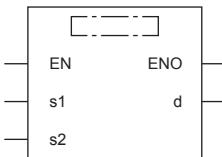
ESTR(P)/DESTR(P)

FX5S　**FX5UJ**　**FX5U**　**FX5UC**

These instructions convert the single-precision real number data stored in the device specified by (s1) into a character string according to the display specification stored in the device specified by (s2) and later, and store the string in the device specified by (d) and later.

The ESTR(P) instructions can also be used as DESTR(P).

Ladder diagram	Structured text
┌─┈─┐ (s1) (s2) (d)	ENO:=ESTR(EN,s1,s2,d); ENO:=ESTRP(EN,s1,s2,d);

FBD/LD
┌─┈─┐ EN   ENO s1   d s2

Setting data

■Descriptions, ranges, and data types

Operand	Description	Range	Data type	Data type (label)
(s1)	Single-precision real number data to be converted or the start number of the device where data is stored	$0, 2^{-126}<|(s1)|<2^{128}$	Single-precision real number	ANYREAL_32
(s2)	Head device number storing the display specification of a numeric value to be converted The device specified in (s2) and following 2 devices are used.	—	16-bit signed binary	ANY16_ARRAY (Number of elements: 3)
(d)	Head device number for storing the converted data	—	Character string	ANYSTRING_SINGLE
EN	Execution condition	—	Bit	BOOL
ENO	Execution result	—	Bit	BOOL
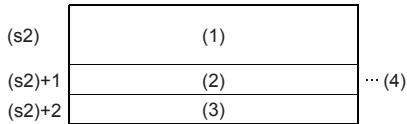
■Applicable devices

Operand	Bit	Word			Double word		Indirect specification	Constant			Others
	X, Y, M, L, SM, F, B, SB, S	T, ST, C, D, W, SD, SW, R	U□\G□	Z	LC	LZ		K, H	E	$	
(s1)	—	○	○	—	○	—	○	—	○	—	—
(s2)	○	○	○	—	—	—	○	—	—	—	—
(d)	—	○[*1]	—	—	—	—	○	—	—	—	—

*1　T, ST, and C cannot be used.

Processing details

- These instructions convert the single-precision real number data stored in the device specified by (s1) into a character string according to the display specification stored in the device specified by (s2) and later, and store the string in the device specified by (d) and later. A real number can be directly specified as (s1).
- The data after conversion varies depending on the display specification stored in (s2).

(s2)	(1)	
(s2)+1	(2)	···(4)
(s2)+2	(3)	

(1): 0: Decimal part format, 1: Exponent format
(2): Total number of digits
(3): Number of digits of decimal part
(4): 2 to 24 can be set.
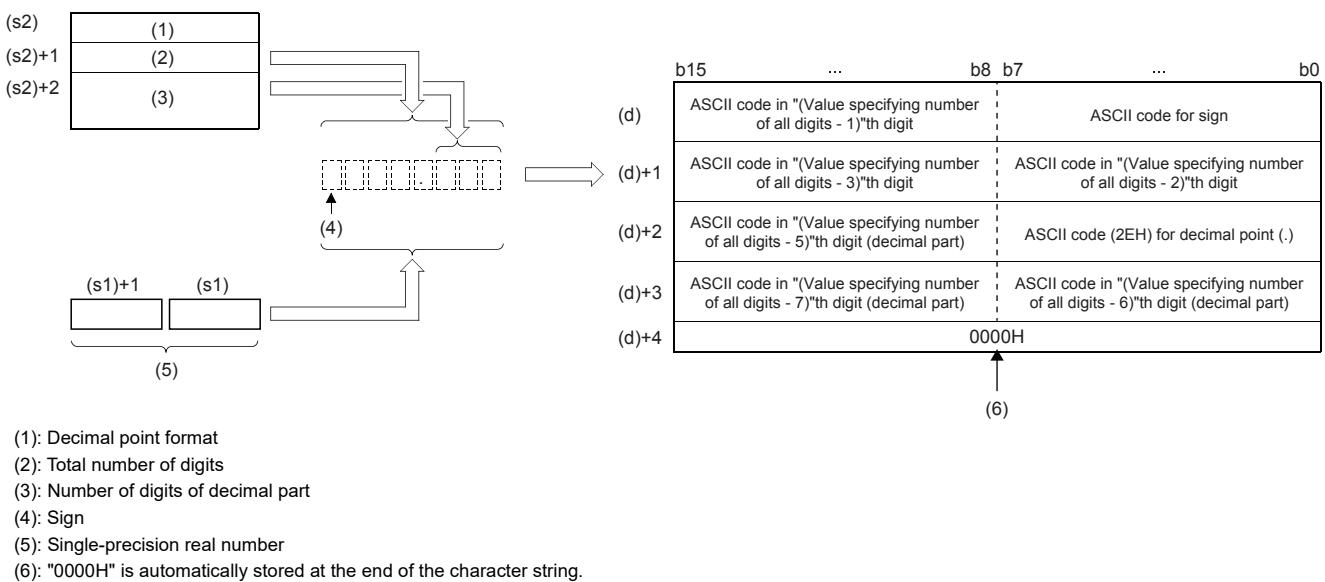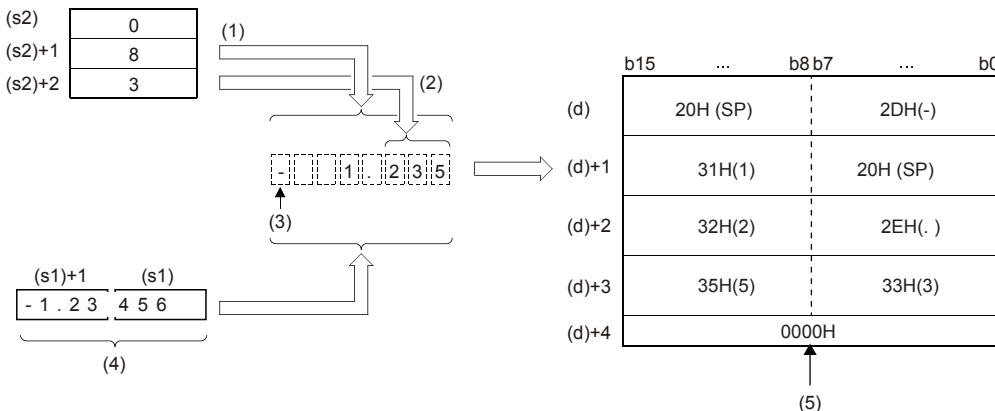
■Decimal point format

- When 0 is specified in (s2), the decimal point format is applied.
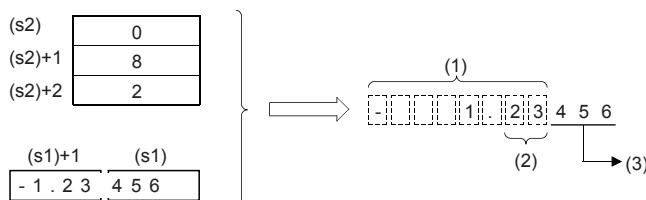
(1): Decimal point format
(2): Total number of digits
(3): Number of digits of decimal part
(4): Sign
(5): Single-precision real number
(6): "0000H" is automatically stored at the end of the character string.

- When the number of decimal part digits is 0, the number of digits that can be specified by (s2)+1 is "the number of digits (24 at a maximum) $\geq$ 2". For other than 0, the number of digits that can be specified by (s2)+1 is "the number of digits (24 at a maximum) $\geq$ (the number of decimal point digits + 3)".
- The number of digits in the decimal part that can be specified by (s2)+2 is 0 to 7. Note that the number of digits in the decimal part must be smaller than the total number of digits minus 3.
- For example, when the total number of digits is "8", the number of digits of the decimal part is "3", and "-1.23456" is specified, data is stored in (d) and later as shown below:

(1): Total number of digits
(2): Number of digits of decimal part
(3): Sign
(4): Single-precision real number
(5): "0000H" is automatically stored at the end of the character string.
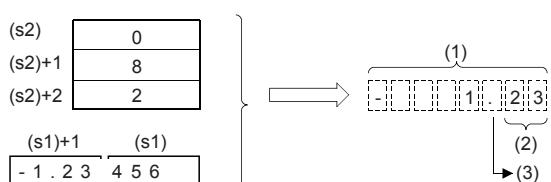
- The converted character string data are stored in the device areas specified by (d) and later as shown below.
  - As sign data, "20H" (space) is stored if the single-precision real number is positive, and "2DH" (-) is stored if the data is negative.
  - If the decimal part of the single-precision real number data cannot be accommodated in the number of digits of the decimal part, lower digits of the decimal part are rounded off.
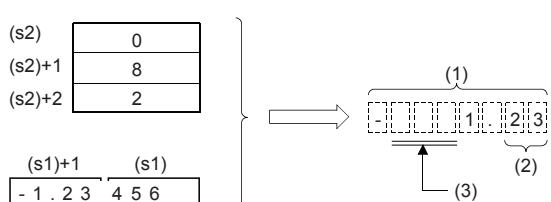
(1): Total number of digits (2): Number of digits of decimal part (3): These digits are rounded off.

  - If the number of digits in the decimal part is set to other than 0, "2EH" (.) is automatically stored at the position before the specified number of digits. If the number of digits in the decimal part is 0, "2EH" (.) is not stored.

(1): Total number of digits
(2): Number of digits of decimal part
(3): Added automatically

 - When the total number of digits subtracted by the digits for sign, decimal point, and decimal part is larger than the integer part of the single-precision real number data, "20H (space)" is stored between the sign and the integer part.
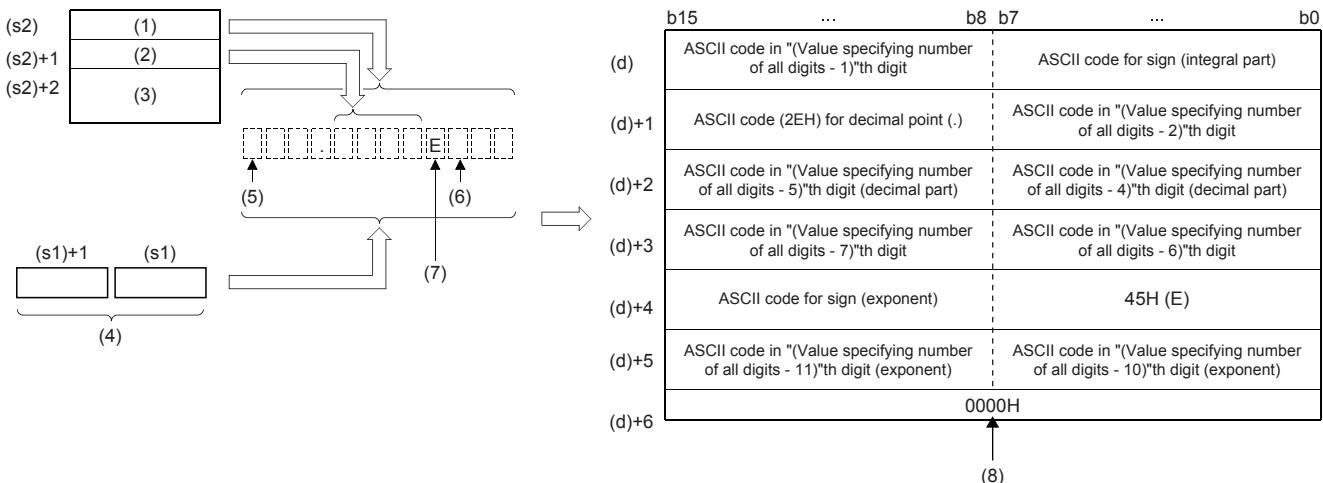
(1): Total number of digits
(2): Number of digits of decimal part
(3): "20H (SP)" is stored.
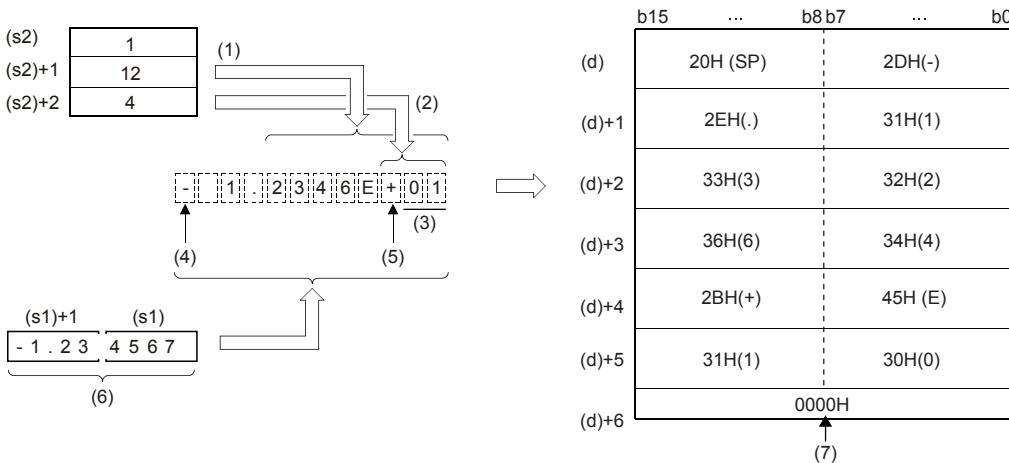
  - The value "00H" is automatically stored at the end of the converted character string.

■Exponent format

• When 1 is specified in (s2), the exponent format is applied.
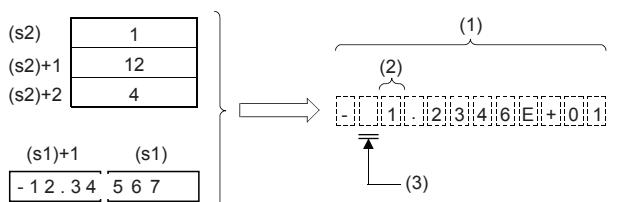
(1): Exponent format
(2): Total number of digits
(3): Number of digits of decimal part
(4): Single-precision real number
(5): Sign (integer part)
(6): Sign (exponent part)
(7): Added automatically
(8): "0000H" is automatically stored at the end of the character string.

• When the number of decimal part digits is 0, the number of digits that can be specified by (s2)+1 is "the number of digits (24 at a maximum) ≥ 6". For other than 0, the number of digits that can be specified by (s2)+1 is "the number of digits (24 at a maximum) > (the number of decimal point digits + 7)".

• The number of digits in the decimal part that can be specified by (s2)+2 is 0 to 7. Note that the number of digits in the decimal part must be equal to or smaller than the total number of digits minus 7.

• For example, when the total number of digits is "12", the number of digits of the decimal part is "4", and "-12.34567" is specified, data is stored in (d) and later as shown below:

(1): Total number of digits
(2): Number of digits of decimal part
(3): Fixed to 2 digit
(4): Sign (integer part)
(5): Sign (exponent part)
(6): Single-precision real number
(7): "0000H" is automatically stored at the end of the character string.

- The converted character string data are stored in the device areas specified by (d) and later as shown below.
  - As sign data of the integral part, "20H" (space) is stored if the single-precision real number is positive, and "2DH" (-) is stored if the data is negative.
  - The integer part is fixed to 1 digit. "20H (space)" is stored between the integer part and the sign.

```
(s2)      1                                (1)
(s2)+1    12                      (2)
(s2)+2    4                      ⌒
                    ⇒     - | 1 | 2 3 4 6 E + 0 1
(s1)+1    (s1)                   ̄
  - 1 2 . 3 4  5 6 7            ↑
                                └─ (3)
```
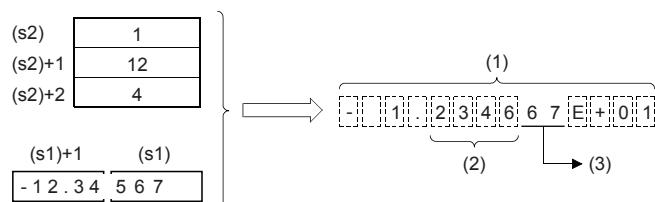
(1): Total number of digits
(2): Fixed to 1 digit
(3): "20H (SP)" is stored.

- If the decimal part of the single-precision real number data cannot be accommodated in the number of digits of the decimal part, lower digits of the decimal part are rounded.

```
(s2)      1                                (1)
(s2)+1    12                    ⌒⌒⌒⌒⌒⌒⌒⌒⌒⌒⌒
(s2)+2    4
                    ⇒     - | 1 | 2 3 4 6  6  7 E + 0 1
(s1)+1    (s1)                      ⌣⌣⌣⌣      ↓
  - 1 2 . 3 4  5 6 7               (2)       (3)
```
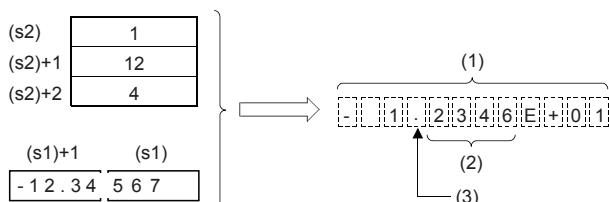
(1): Total number of digits
(2): Number of digits of decimal part
(3): These values are rounded down.

- If the number of digits in the decimal part is set to other than 0, "2EH" (.) is automatically stored at the position before the specified number of digits. If the number of digits in the decimal part is 0, "2EH" (.) is not stored.

```
(s2)      1                                (1)
(s2)+1    12                    ⌒⌒⌒⌒⌒⌒⌒⌒⌒⌒⌒
(s2)+2    4
                    ⇒     - | 1 | . 2 3 4 6 E + 0 1
(s1)+1    (s1)                  ↑   ⌣⌣⌣⌣
  - 1 2 . 3 4  5 6 7           │    (2)
                              (3)
```
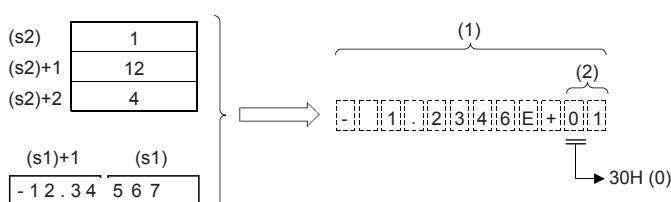
(1): Total number of digits
(2): Number of digits of decimal part
(3): Added automatically

- For the sign of the exponent part, "2BH (+)" is stored when the exponent is positive, and "2DH (-)" is stored when the exponent is negative.
- The exponent part is fixed to 2 digits. When the exponent part is 1 digit, "30H (0)" is stored after the sign of the exponent part.

```
(s2)      1                                (1)
(s2)+1    12                              (2)
(s2)+2    4                              ⌒
                    ⇒     - | 1 | 2 3 4 6 E + 0 1
(s1)+1    (s1)                             ̄
  - 1 2 . 3 4  5 6 7                       ↓
                                        30H (0)
```
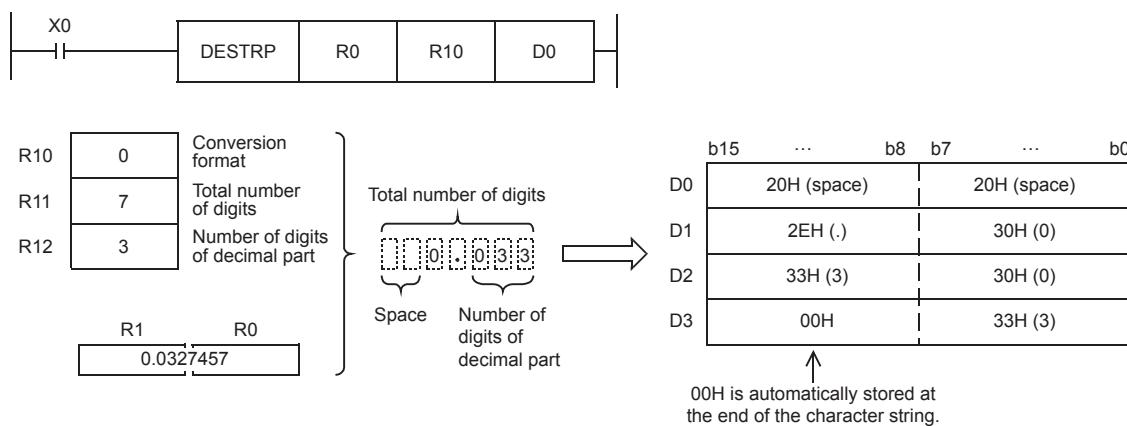
(1): Total number of digits
(2): Fixed to 2 digit
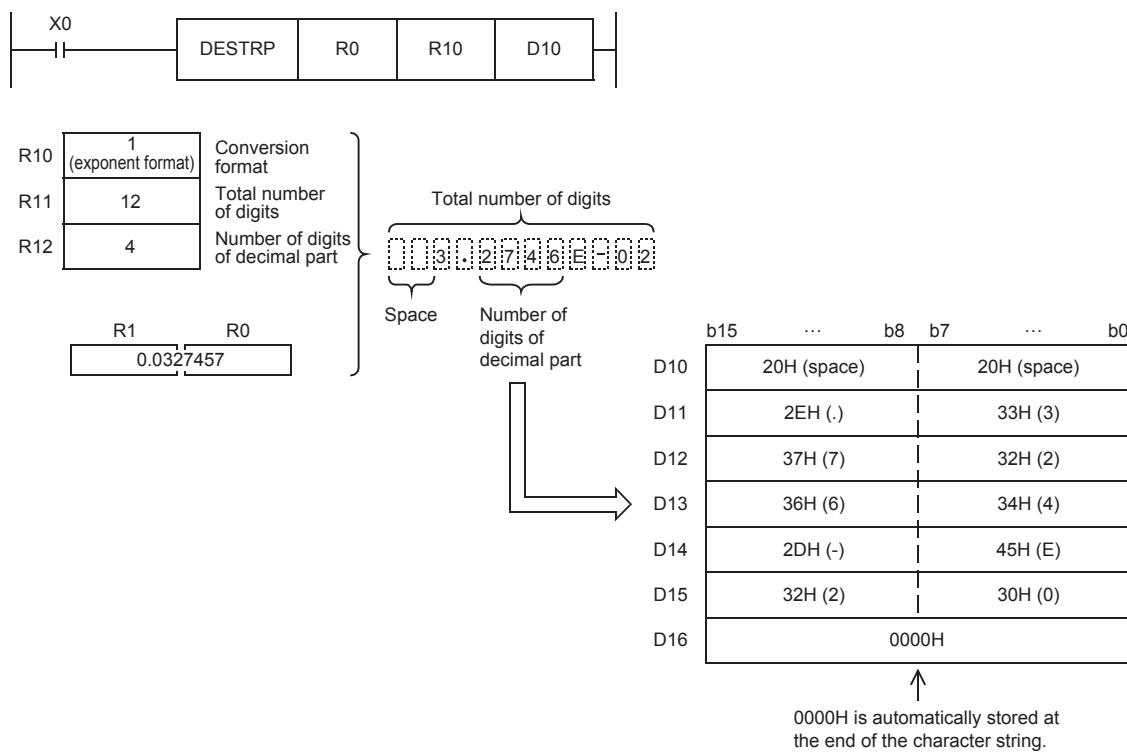
- The value "00H" is automatically stored at the end of the converted character string.

Program example

In the program example shown below, the contents (single-precision real number data) of R0 and R1 are converted according to the contents specified by R10 to R12, and then stored to D0 and later when X0 turns ON.

```
    X0
────┤├────┌────────┬──────┬──────┬──────┐
         │ DESTRP │  R0  │ R10  │  D0  │
         └────────┴──────┴──────┴──────┘
```

R10	0	Conversion format
R11	7	Total number of digits
R12	3	Number of digits of decimal part

Total number of digits

⌐ 0 . 0 3 3 ¬

Space   Number of digits of decimal part

	b15 ··· b8	b7 ··· b0
D0	20H (space)	20H (space)
D1	2EH (.)	30H (0)
D2	33H (3)	30H (0)
D3	00H	33H (3)

R1	R0
0.0327457	

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

In the program shown below, the contents (single-precision real number data) of R0 and R1 are converted according to the contents specified by R10 to R12, and then stored to D10 and later when X0 turns ON.

```
    X0
────┤├────┌────────┬──────┬──────┬──────┐
         │ DESTRP │  R0  │ R10  │ D10  │
         └────────┴──────┴──────┴──────┘
```

R10	1 (exponent format)	Conversion format
R11	12	Total number of digits
R12	4	Number of digits of decimal part

Total number of digits

⌐ 3 . 2 7 4 6 E - 0 2 ¬

Space   Number of digits of decimal part

R1	R0
0.0327457	

	b15 ··· b8	b7 ··· b0
D10	20H (space)	20H (space)
D11	2EH (.)	33H (3)
D12	37H (7)	32H (2)
D13	36H (6)	34H (4)
D14	2DH (-)	45H (E)
D15	32H (2)	30H (0)
D16	0000H	

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

Operation error

Error code (SD0/SD8067)	Description
2820H	The device specified by (s2) exceeds the corresponding device range.
3401H	The number of total digits specified by (s1)+1 exceeds 24.
	The format specified by (s2) is any value other than "0" or "1".
	The total number of digits specified by (s2)+1 is not within the following range in the decimal point format. When the number of digits of the decimal part is "0": Total number of digits ≥ 2 When the number of digits of the decimal part is any value other than "0": Total number of digits ≥ (Number of digits of decimal part + 3)
	The total number of digits specified by (s2)+1 is not within the following range in the exponent format. When the number of digits of the decimal part is "0": Total number of digits ≥ 6 When the number of digits of the decimal part is any value other than "0": Total number of digits ≥ (Number of digits of decimal part + +7)
	The number of digits of the decimal part specified by (s2)+2 is not within the following range. In the decimal part format ≤ (Total number of digits - 3) In the exponent format ≤ (Total number of digits - 7)
	When the conversion result exceeds the specified total number of digits
3402H	(s1) is not within the following range $0, \pm 2^{-126} \leq (s1) < \pm 2^{128}$
	The specified device value is denormalized number, NaN (not a number), or $\pm\infty$.
3405H	The number of digits of the decimal part specified by (s2)+2 is not within the following range. 0 to 7
3406H	The device areas that store the character string specified by (d) exceed the corresponding device range.

8