

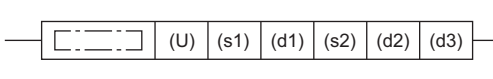
8.7 File Operation Instructions

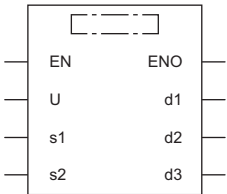
Reading data from the specified file

SP.FREAD

FX5S**FX5UJ****FX5U****FX5UC**

This instruction reads device data from the specified file in an SD memory card.

Ladder	ST
	ENO:=SP_FREAD(EN,U,s1,s2,d1,d2,d3);

FBD/LD


Setting data

■Description, range, data type

Operand	Description	Range	Data type	Data type (label)
(U)	Dummy	■FX5S CPU module U1 ■FX5UJ CPU module U1 to U8 ■FX5U/FX5UC CPU module U1 to U10	Device name ^{*2}	ANY16
(s1)	Drive specification	2 (fixed) ^{*1}	16-bit signed binary	ANY16
(d1)	Start device where the control data is stored	☞ Page 487 Control data (d1)	Word	ANY16_ARRAY (Number of elements: 8)
(s2)	Start device where the file name is stored	☞ Page 488 File name (s2)	Unicode string	ANYSTRING_DOUBLE
(d2)	Start device for storing the data that has been read	☞ Page 489 Read data (d2)	Word	ANY16 ^{*3}
(d3)	Bit device that turns on upon completion of the processing (In the case of an error completion, the device specified by (d3)+1 also turns on.)	—	Bit	ANYBIT_ARRAY (Number of elements: 2)
EN	Execution condition	—	Bit	BOOL
ENO	Execution result	—	Bit	BOOL

*1 Only drive 2 (for the SD memory card) can be set.

*2 Specification with a label is not allowed.

*3 When specifying setting data by using a label, define an array to secure enough operation area and specify an element of the array label.

■Applicable devices

Operand	Bit	Word			Double word		Indirect specification	Constant			Others (U)
	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	\$	
(U)	—	—	—	—	—	—	—	—	—	—	○
(s1)	○	○	—	—	—	—	○	○	—	—	—
(d1)	—	○	—	—	—	—	○	—	—	—	—
(s2)	—	○	—	—	—	—	○	—	—	○	—
(d2)	○*1	○	—	—	—	—	○	—	—	—	—
(d3)	○*2	○*3	—	—	—	—	—	—	—	—	—


*1 When the bit device digit is specified in (d2), only multiples of 16 (0, 16, 32, 64...) can be specified as the device number. Only K4 can be specified as the number of digits.


*2 S cannot be used.

*3 T, ST, and C cannot be used.

Only bit specification of word device is applicable.

■Control data (d1)

For the combination of processing units, refer to  Page 490 Combination of processing units .

Operand: (d1)				
Device	Item	Description	Setting range	Set by
+0	Execution/completion type	Specify the execution type. ■00**H: Reading binary data • 0000H: 16-bit binary data • 0001H: 32-bit binary data ■01**H: Reading data after converted to CSV format • 0100H: Decimal (16-bit data) • 0110H: Decimal (32-bit data) • 0120H: Hexadecimal (16-bit data) • 0121H: Hexadecimal (32-bit data) • 0130H: String (ASCII data) • 0140H: Floating point real number (single-precision real number)	0000H 0001H 0100H 0110H 0120H 0121H 0130H 0140H	User
+1	Completion status	The completion status is stored upon completion of the instruction. • 0000H: Completed successfully • Other than 0000H: Completed with an error (error code) ( Page 579 Error codes generated for file operation instructions)	—	System
+2	Number of read-target data	Specify the number of data to be read. The processing unit and setting range vary depending on the execution/completion type in (d1)+0 and data type specification in (d1)+7. ■When "Reading binary data" is specified by (d1)+0 • When 16-bit binary data is specified: In units of words (1 to 65535)*1*2*3 • When 32-bit binary data is specified: In units of double words (1 to 32767) ■When "Reading data after converted to CSV format" is specified by (d1)+0 • When decimal (16-bit data) is specified: Number of elements (1 to 65535)*1*2*3 • When decimal (32-bit data) is specified: Number of elements (1 to 32767) • When hexadecimal (16-bit data) is specified: Number of elements (1 to 65535)*3 • When hexadecimal (32-bit data) is specified: Number of elements (1 to 32767) • When a string (ASCII data) is specified: Number of elements (1 to 1023) • When a floating point real number (single-precision real number) is specified: Number of elements (1 to 32767)	1 to 65535	User
+3	Maximum number of read data	■When "0130H: String (ASCII data)" is specified by (d1)+0 • Total size of the characters in the element (in units of words) ■When data other than "0130H: String (ASCII data)" is specified by (d1)+0 • Fixed to 0	0, 1 to 65535	User
+4 +5	File position	■When "Reading binary data" is specified by (d1)+0 • 00000000H: From the beginning of the file • 00000001H to FFFFFFFEH: From the specified position (The data unit is determined by (d1)+7.) • FFFFFFFFH: Cannot be specified. ■When "Reading data after converted to CSV format" is specified by (d1)+0 • 00000000H: From the beginning of the file • 00000001H to FFFFFFFEH: From the specified row • FFFFFFFFH: From the position where the previous reading ends	00000000H to FFFFFFFFH	User

Operand: (d1)				
Device	Item	Description	Setting range	Set by
+6	Number of columns	<p>When "Reading binary data" is specified by (d1)+0, set 0.</p> <p>When "Reading data after converted to CSV format" is specified by (d1)+0, set the number of read-target columns.</p> <ul style="list-style-type: none"> • 0: No specification of number of columns. In the CSV format, the data from the first column to the last column are read ignoring line separators. (The number of read-target data is read as one row.) • Other than 0: Number of specified columns 	0000H to FFFFH (0 to 65535)	User
+7	Data type specification	<ul style="list-style-type: none"> • 0: Word • 1: Even number of bytes^{*1} • 2: Unit of the data type specified by (d1)+0 • 3: Odd number of bytes^{*1*2} <p>"0: Word", "1: Even number of bytes", and "3: Odd number of bytes" can be specified only when "0000H: 16-bit binary data" or "0100H: Decimal (16-bit data)" is specified by (d1)+0.</p>	0, 1, 2, 3	User

*1 When "1: Even number of bytes" or "3: Odd number of bytes" is specified by (d1)+7 (Data type specification), the setting range of (d1)+2 (Number of read-target data) is 1 to 32767.

*2 Even when "3: Odd number of bytes" is specified by (d1)+7 (Data type specification), specify the number in units of words. Add one byte to the odd number of bytes to be read and set the number in (d1)+2 (Number of read-target data).

*3 Up to 32767 points (when the file register (R) is used) when only devices are used.

■File name (s2)

Operand: (s2)				
Device	Item	Description	Setting range	Set by
+0 to +□	File name character string	<p>Specify the folder path where the files are stored and the file name.</p> <ul style="list-style-type: none"> • The folder path and file name (including an extension) must be within 253 characters in total. • The folder path must be within 244 characters. (Delimiters are not included.) • The number of folder path hierarchies must be within 10 levels. • When omitting an extension in the file name, omit the ". (period)" as well. • The file name must be within 60 characters (a period and extension excluded). If 61 or more characters are used, the extension is ignored and replaced with ".BIN" or ".CSV". • Do not specify a half-width space at the end of the character string or just before each delimiter. <div style="text-align: center;"> <p>"(3)folder1/(2)user1/user1.(4)csv"</p> </div> <p>(1): Up to 253 characters (2): Use "/" or "\" as delimiters for the folder path and file. (3): Can be omitted. When it is omitted, (1) is up to 252 characters. (4): If the extension is omitted, ".BIN" or ".CSV" will be automatically added by the system.</p>	Unicode string	User

■Read data (d2)

For the combination of processing units, refer to  Page 490 Combination of processing units.

Operand: (d2)				
Device	Item	Description	Setting range	Set by
+0	Number of data actually read	The number of data actually read is set. The data unit is determined by (d1)+7 (Data type specification).	—	System
+1 to +□	Read data	The read data is stored.	—	System

Processing details

- This instruction reads data from the specified file. Set the execution/completion type in the control data to specify the file read-target format.
- The read target is the data in the SD memory card only.
- The processing completion bit device (d3) automatically turns on at the execution of the END instruction in the scan in which the completion of processing of the SP.FREAD instruction is detected. The bit device (d3) turns off at the execution of the END instruction in the next scan. If the processing completion bit device (d3) is ON, it will automatically turn OFF when the SP.FREAD instruction is executed.
- If the SP.FREAD instruction completes with an error, the error completion device (d3)+1 turns on or off in synchronization with (d3).
- SM753 (File being accessed) turns on while the SP.FREAD instruction is being executed. While SM753 is on, the SP.FREAD instruction cannot be executed. (If the instruction is executed, no processing is performed.)
- If an error is detected during the execution of the instruction, Processing Complete (d3), Error Completion (d3)+1, and SM753 do not turn on.
- When "Odd number of bytes" is specified by (d1)+7, the read data is stored as shown below. (In the shaded area shown below, the data before execution of the instruction is kept.)

	b15	...	b8 b7	...	b0
(d2)					
(d2)+1					
(d2)+2					
(d2)+3					
(d2)+4					
(d2)+5					

■Combination of processing units

Specify data in (d1)+2, (d1)+4, (d1)+5, and (d2)+0 depending on the combination of (d1)+0 and (d1)+7.

Execution/completion type (d1)+0		Data type specification (d1)+7	Processing unit and setting range		
			Number of read- target data (d1)+2	File position (d1)+4, (d1)+5	Number of data actually read (d2)+0
Reading binary data	0000H: 16-bit binary data	0: Word	Word (1 to 65535) ^{*1}	Word (00000000H to 7FFFFFFFH)	Word
		1: Even number of bytes	Word (1 to 32767)	Byte (00000000H to FFFFFFFEH)	Byte
		2: Unit of the data type specified by the execution/completion type	Word (1 to 65535) ^{*1}	Word (00000000H to 7FFFFFFFH)	Word
		3: Odd number of bytes	Word (1 to 32767) ^{*2}	Byte (00000000H to FFFFFFFEH)	Byte
	0001H: 32-bit binary data	0: Word 1: Even number of bytes	(Cannot be specified)		
		2: Unit of the data type specified by the execution/completion type	Double word (1 to 32767)	Double word (00000000H to 3FFFFFFFH)	Double word
		3: Odd number of bytes	(Cannot be specified)		

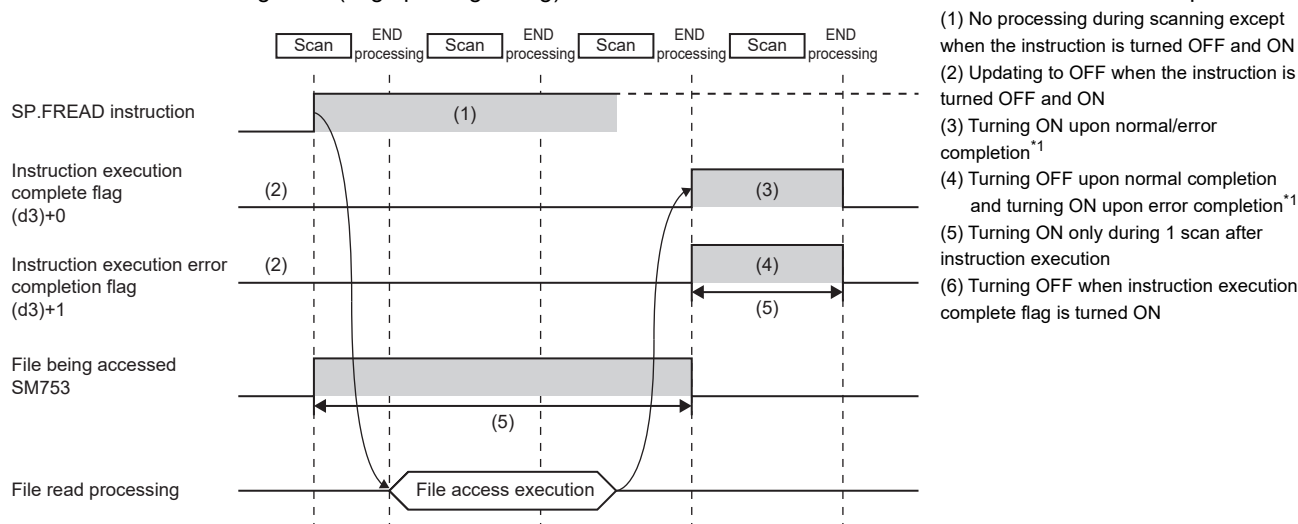
Execution/completion type (d1)+0		Data type specification (d1)+7	Processing unit and setting range		
			Number of read- target data (d1)+2	File position (d1)+4, (d1)+5	Number of data actually read (d2)+0
Reading data after converted to CSV format	0100H: Decimal (16- bit data)	0: Word	Number of elements (1 to 65535) ^{*1}	Number of lines	Word
		1: Even number of bytes	Number of elements (1 to 32767)	Number of lines	Byte
		2: Unit of the data type specified by the execution/completion type	Number of elements (1 to 65535) ^{*1}	Number of lines	Word
		3: Odd number of bytes	Number of elements (1 to 32767) ^{*2}	Number of lines	Byte
	0110H: Decimal (signed 32-bit data)	0: Word 1: Even number of bytes	(Cannot be specified)		
		2: Unit of the data type specified by the execution/completion type	Number of elements (1 to 32767)	Number of lines	Double word
		3: Odd number of bytes	(Cannot be specified)		
	0120H: Hexadecimal (16-bit data)	0: Word 1: Even number of bytes	(Cannot be specified)		
		2: Unit of the data type specified by the execution/completion type	Number of elements (1 to 65535)	Number of lines	Word
		3: Odd number of bytes	(Cannot be specified)		
	0121H: Hexadecimal (32-bit data)	0: Word 1: Even number of bytes	(Cannot be specified)		
		2: Unit of the data type specified by the execution/completion type	Number of elements (1 to 32767)	Number of lines	Double word
		3: Odd number of bytes	(Cannot be specified)		
	0130H: String (ASCII data)	0: Word 1: Even number of bytes	(Cannot be specified)		
		2: Unit of the data type specified by the execution/completion type	Number of elements (1 to 1023)	Number of lines	Number of elements
		3: Odd number of bytes	(Cannot be specified)		
	0140H: Floating point real number (single- precision real number)	0: Word 1: Even number of bytes	(Cannot be specified)		
		2: Unit of the data type specified by the execution/completion type	Number of elements (1 to 32767)	Number of lines	Double word
		3: Odd number of bytes	(Cannot be specified)		

*1 Up to 32767 points (when the file register (R) is used) when only devices are used.

*2 Specify the number of read-target data in word units adding 1 to the number of odd byte data to be read.

■Timing chart

Below is shown the timing chart (flag updating timing) from the execution of the SP.FREAD instruction to the completion.



*1 The complete flag is not turned ON when an error is detected during instruction execution.

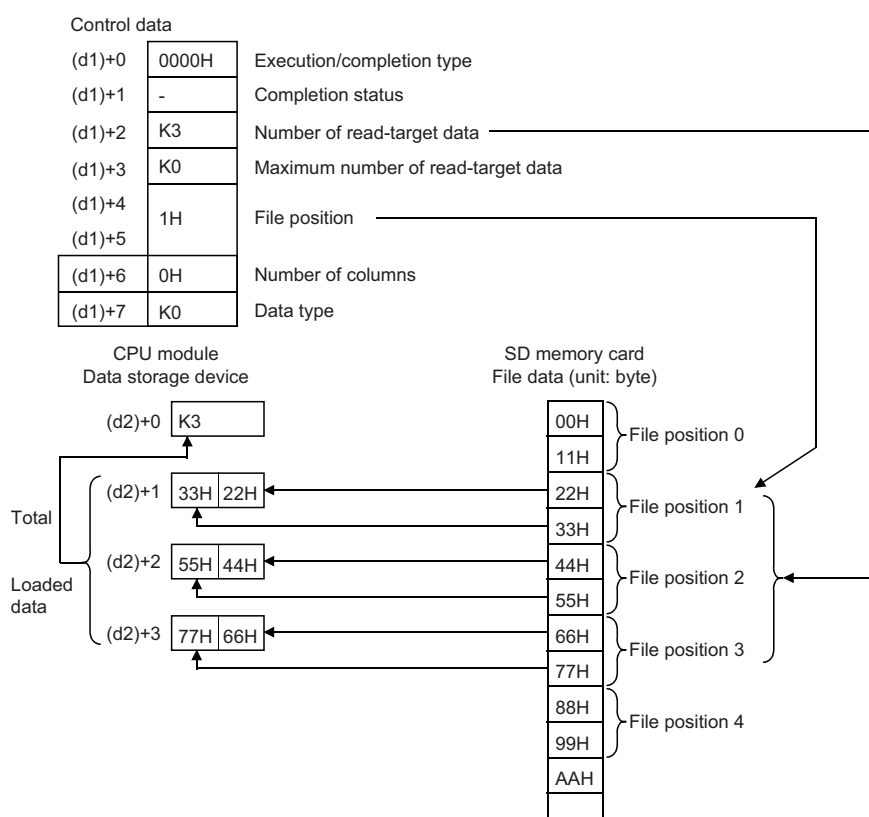
■When reading binary data

- If the extension of the target file is omitted, the extension will be ".BIN".
- If the specified file does not exist, an error (8002H) occurs.
- If the position is specified exceeding the existing file size, 0 point of data is read and the processing completes successfully.

The following figure shows an example of reading binary data.

Ex.

Example of reading binary data (16-bit binary data)



■When reading data after converted to CSV format

- Elements in the CSV format file (cells in Excel[®]) are read by each row and stored in the device.
- If the extension of the target file is omitted, the extension will be ".CSV".
- If the specified file does not exist, an error (8002H) occurs.
- If the position is specified exceeding the existing file size, 0 point of data is read and the processing completes successfully.
- The number of data specified by (d1)+2 is read from the beginning of the file. If the last data in the file is read before reaching to the number specified, only the available number of data is read.
- When the number of columns is set to 0, the data are read by ignoring the rows (CR+LF, only CR, only LF) in the CSV format file.
- The character code of the specified file is read as a shift JIS code when the specified execution/completion type is decimal, hexadecimal or single-precision real number.
- The string data in the CSV file and the value stored in the device after it is read are determined by the execution/completion type.

Execution/ completion type	Data type specification	Data (one element) in the CSV file	Value stored in the device	Remarks
0100H: Decimal (16-bit data)	0: Word 2: Unit of the data type specified by (d1)+0	-32768 to -1	-32768 to -1 (32768 to 65535)	The value should be within the range of -32768 to -1 as signed 16-bit data and within the range of 32768 to 65535 as unsigned 16-bit data. The same value is stored in the device.
		0 to 32767	0 to 32767	—
		32768 to 65535	-32768 to -1 (32768 to 65535)	The value should be within the range of -32768 to -1 as signed 16-bit data and within the range of 32768 to 65535 as unsigned 16-bit data. The same value is stored in the device.
		• Numeric values other than above • String containing alphabets and symbols	0	Filled with 0 since it cannot be converted.
	1: Even number of bytes 3: Odd number of bytes	-128 to -1	-128 to -1 (128 to 255)	The value should be within the range of -128 to -1 as signed 8-bit data and within the range of 128 to 255 as unsigned 8-bit data. The same value is stored in the device.
		0 to 127	0 to 127	—
		128 to 255	-128 to -1 (128 to 255)	The value should be within the range of -128 to -1 as signed 8-bit data and within the range of 128 to 255 as unsigned 8-bit data. The same value is stored in the device.
		• Numeric values other than above • String containing alphabets and symbols	0	Filled with 0 since it cannot be converted.
0110H: Decimal (32-bit data)	—	-2147483648 to -1	-2147483648 to -1 (2147483648 to 4294967295)	The value should be within the range of -2147483648 to -1 as signed 32-bit data and within the range of 2147483648 to 4294967295 as unsigned 32-bit data. The same value is stored in the device.
		0 to 2147483647	0 to 2147483647	—
		2147483648 to 4294967295	-2147483648 to -1 (2147483648 to 4294967295)	The value should be within the range of -2147483648 to -1 as signed 32-bit data and within the range of 2147483648 to 4294967295 as unsigned 32-bit data. The same value is stored in the device.
		• Numeric values other than above • String containing alphabets and symbols	0	Filled with 0 since it cannot be converted.

Execution/ completion type	Data type specification	Data (one element) in the CSV file	Value stored in the device	Remarks
0120H: Hexadecimal (16-bit data)	—	0H to FFFFH	0H to FFFFH	—
		<ul style="list-style-type: none"> Numeric values other than above String containing alphabets other than A to F and symbols 	0000H	Filled with 0 since it cannot be converted.
0121H: Hexadecimal (32-bit data)	—	0H to FFFFFFFFH	0H to FFFFFFFFH	—
		<ul style="list-style-type: none"> Numeric values other than above String containing alphabets other than A to F and symbols 	00000000H	Filled with 0 since it cannot be converted.
0130H: String (ASCII data)	—	String (up to 1999 characters)	String (up to 1999 characters)	When the number of bytes of the string in the CSV file is odd, NULL (00H) is added to the end of the string. When the number of bytes of the string in the CSV file is even, 0000H is stored in the next one word. When the string in the CSV file contains 00H, it is ignored.
		String (2000 characters or more)		If the number of characters in one element exceeds 1999, characters until the 1999th character are read as one element. The 2000th character and after are not read and the next element is read.
0140H: Floating point real number (single-precision real number)	—	Values within the range of: $-2^{128} < \text{data} \leq -2^{-126}$, 0, $2^{-126} \leq \text{data} < 2^{128}$	As given on the left	The value is converted in either decimal point or exponential format.
		Numeric values other than above	0	Filled with 0 since it cannot be converted.

Ex.

When "Reading data after converted to CSV format" (String (ASCII data)) is specified

[Data stored in CSV format]

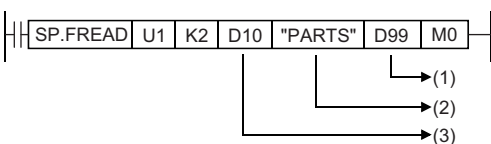
PARTS.CSV

No.	,	Name	,	Value1	,	Value2	CR	LF
AA_0001	,	Prts_A	,	100	,	200	CR	LF
BB_0002	,	Prts_B	,	300	,	400	CR	LF

Enclosed values as shown to the left are read.

(Total of six elements, three columns from two lines of PARTS.CSV, are read.)

[Data to be loaded to the device]



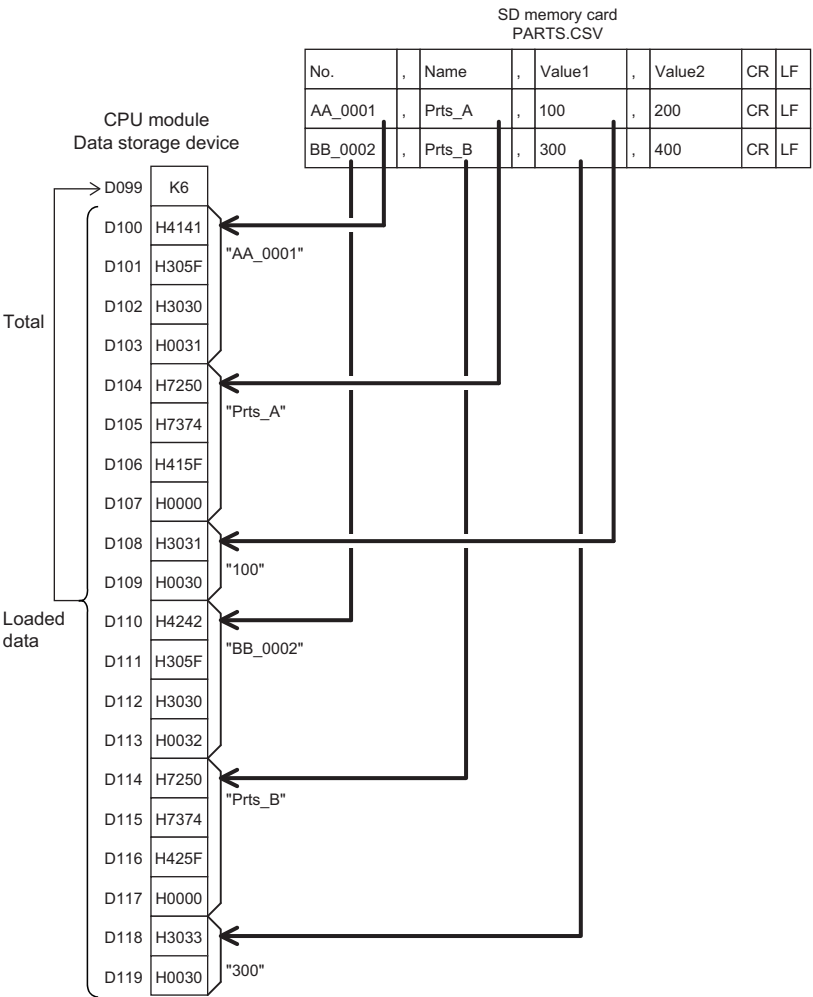
- (1) Start device for storing the data that has been read
- (2) File name
- (3) Control data

[Control data]

D10	H0130
D11	H0000
D12	K6
D13	K100
D14	2H
D15	3H
D16	3H
D17	K2

- D10: Execution/completion type
- D11: Completion status
- D12: Number of read-target data
- D13: Maximum number of read data
- D14, D15: File position
- D16: Number of columns
- D17: Data type specification

[Loaded data]



D99: Number of data actually read
D100 to D103: String in the 1st column of the 2nd line
D104 to D107: String in the 2nd column of the 2nd line
D108, D109: String in the 3rd column of the 2nd line
D110 to D113: String in the 1st column of the 3rd line
D114 to D117: String in the 2nd column of the 3rd line
D118, D119: String in the 3rd column of the 3rd line



When String (ASCII data) is specified, it is necessary to set the total size (in word units) for the maximum number of read data ((d1)+3).

Example: When a string with 100 100-character elements is read from a CSV file, the size is as shown below.

- (100 (character) + 2 (NULL)) × 100 (element) = 10200 byte = 5100 word

In the above case, an area for 5101 words must be secured for the file name (s2).

Ex.

When "Reading data after converted to CSV format" (Decimal (16-bit data)) is specified and the number of columns is set to 0

[Data created in Excel]

	A	B	C
1	Main/sub item		Measured value
2	Length	1	3
3	Temperature	-21	
4			



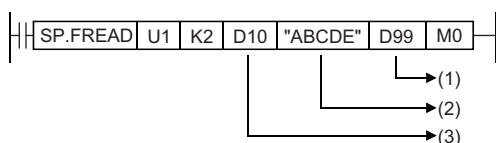
[Data saved in CSV format]

ABCD.CSV

Main/sub item	,	,	Measured value	CR	LF
Length	,	1	,	3	CR LF
Temperature	,	-21	,		CR LF



[Data to be loaded to the device]



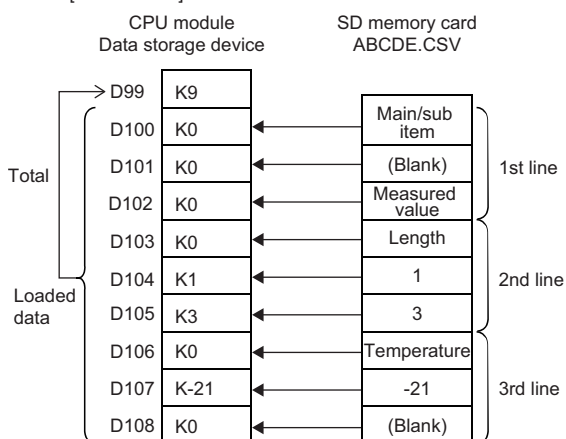
- (1) Start device for storing the data that has been read
- (2) File name
- (3) Control data

[Control data]

D10	0100H
D11	-
D12	K9
D13	K0
D14	0H
D15	0H
D16	0H
D17	K0

- D10: Execution/completion type
- D11: Completion status
- D12: Number of read-target data
- D13: Maximum number of read data
- D14, D15: File position
- D16: Number of columns
- D17: Data type specification

[Loaded data]



D99: Number of data actually read

D100: Since "Main/sub item" is not a numerical value, the conversion data, 0, is stored.

D101: Since " " is not a numerical value, the conversion data, 0, is stored.

D102: Since "Measured value" is not a numerical value, the conversion data, 0, is stored.

D103: Since "Length" is not a numerical value, the conversion data, 0, is stored.

D104: Since "1" is a numerical value, it is converted to a binary value.

D105: Since "3" is a numerical value, it is converted to a binary value.

D106: Since "Temperature" is not a numerical value, the conversion data, 0, is stored.

D107: Since "-21" is a numerical value, it is converted to a binary value.

D108: Since " " is not a numerical value, the conversion data, 0, is stored.

- When the number of columns differs in each row, the data are also read by ignoring the rows.

Ex.

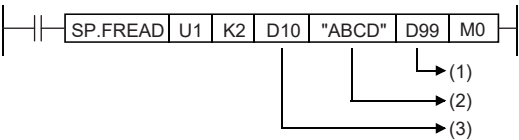
When the number of columns differs in each row

[Data saved in CSV format]
ABCD.CSV

Main/sub item	,	,	Measured value	,	Excess	CR	LF
Length	CR	LF					
Temperature	,	-21	,	CR	LF		



[Data to be loaded to the device]



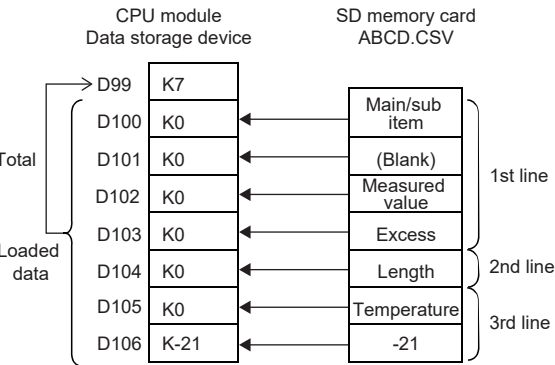
- (1) Start device for storing the data that has been read
- (2) File name
- (3) Control data

[Control data]

D10	0100H
D11	-
D12	K7
D13	K0
D14	0H
D15	0H
D16	0H
D17	K0

- D10: Execution/completion type
- D11: Completion status
- D12: Number of read-target data
- D13: Maximum number of read data
- D14, D15: File position
- D16: Number of columns
- D17: Data type specification

[Loaded data]



- D99: Number of data actually read
- D100: Since "Main/sub item" is not a numerical value, the conversion data, 0, is stored.
- D101: Since " " is not a numerical value, the conversion data, 0, is stored.
- D102: Since "Measured value" is not a numerical value, the conversion data, 0, is stored.
- D103: Since "Excess" is not a numerical value, the conversion data, 0, is stored.
- D104: Since "Length" is not a numerical value, the conversion data, 0, is stored.
- D105: Since "Temperature" is not a numerical value, the conversion data, 0, is stored.
- D106: Since "-21" is a numerical value, it is converted to a binary value.

Point

This type of file in which the number of columns vary with individual rows cannot be created by Excel. It is created when the CSV file is modified by a user.

- When the specified number of columns is set to a value other than 0, a CSV format file is read as the table with the specified number of columns. The elements outside the specified number of columns are ignored.

Ex.

When "Reading data after converted to CSV format" (Decimal (16-bit data)) is specified and the number of columns is set to a value other than 0 ((d1)+6 is set to 2)

[Data created in Excel]

	A	B	C
1	Main/sub item		Measured value
2	Length	1	3
3	Temperature	-21	
4			



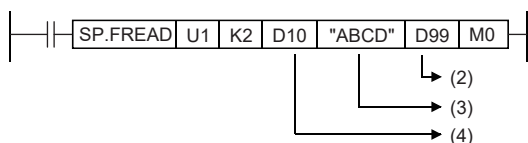
[Data saved in CSV format]
ABCD.CSV

Main/sub item	,	,	Measured value	CR	LF
Length	,	1	,	3	CR LF
Temperature	,	-21	,		CR LF

(1)



[Data to be loaded to the device]



(1) Columns out of the specified range are ignored.

(2) Start device for storing the data that has been read

(3) File name

(4) Control data

[Control data]

D10	0100H
D11	-
D12	K6
D13	K0
D14	0H
D15	0H
D16	2H
D17	K0

D10: Execution/completion type

D11: Completion status

D12: Number of read-target data

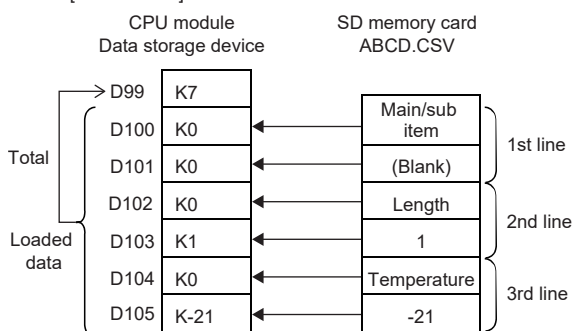
D13: Maximum number of read data

D14, D15: File position

D16: Number of columns

D17: Data type specification

[Loaded data]



D99: Number of data actually read

D100: Since "Main/sub item" is not a numerical value, the conversion data, 0, is stored.

D101: Since " " is not a numerical value, the conversion data, 0, is stored.

D102: Since "Length" is not a numerical value, the conversion data, 0, is stored.

D103: Since "1" is a numerical value, it is converted to a binary value.

D104: Since "Temperature" is not a numerical value, the conversion data, 0, is stored.

D105: Since "-21" is a numerical value, it is converted to a binary value.

- When the number of columns differs in each row, the elements outside the specified number of columns are ignored and 0 is added to the cells where no element exists.

Ex.

When the number of columns differs in each row

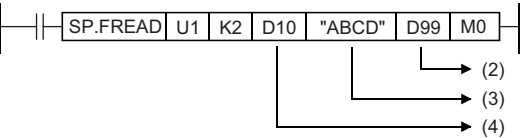
[Data saved in CSV format]
ABCD.CSV

Main/sub item	,	,	Measured value	,	Excess	CR	LF
Length	CR	LF					
Temperature	,	-21		CR	LF		

(1) Columns out of the specified range are ignored.



[Data to be read into devices]



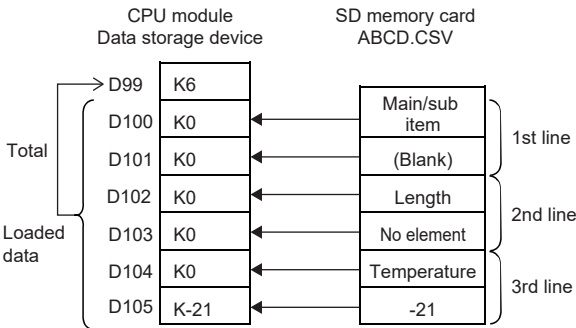
- (2) Start device for storing the data that has been read
- (3) File name
- (4) Control data

[Control data]

D10	0100H
D11	-
D12	K6
D13	K0
D14	0H
D15	0H
D16	2H
D17	K0

- D10: Execution/completion type
- D11: Completion status
- D12: Number of read-target data
- D13: Maximum number of read data
- D14, D15: File position
- D16: Number of columns
- D17: Data type specification

[Loaded data]

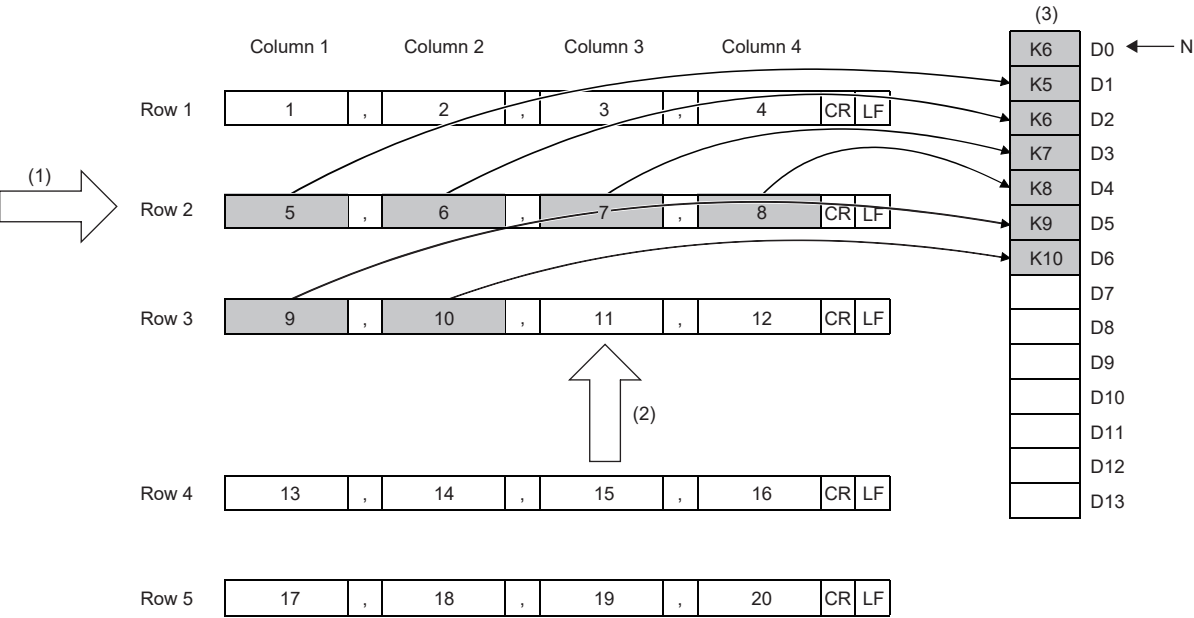


- D99: Number of data actually read
- D100: Since "Main/sub item" is not a numerical value, the conversion data, 0, is stored.
- D101: Since " " is not a numerical value, the conversion data, 0, is stored.
- D102: Since "Length" is not a numerical value, the conversion data, 0, is stored.
- D103: Since no element exists, the conversion data, 0, is added.
- D104: Since "Temperature" is not a numerical value, the conversion data, 0, is stored.
- D105: Since "-21" is a numerical value, it is converted to a binary value.

- When "Reading data after converted to CSV format" is specified, data can be divided and read.

[Specify the row to start reading]

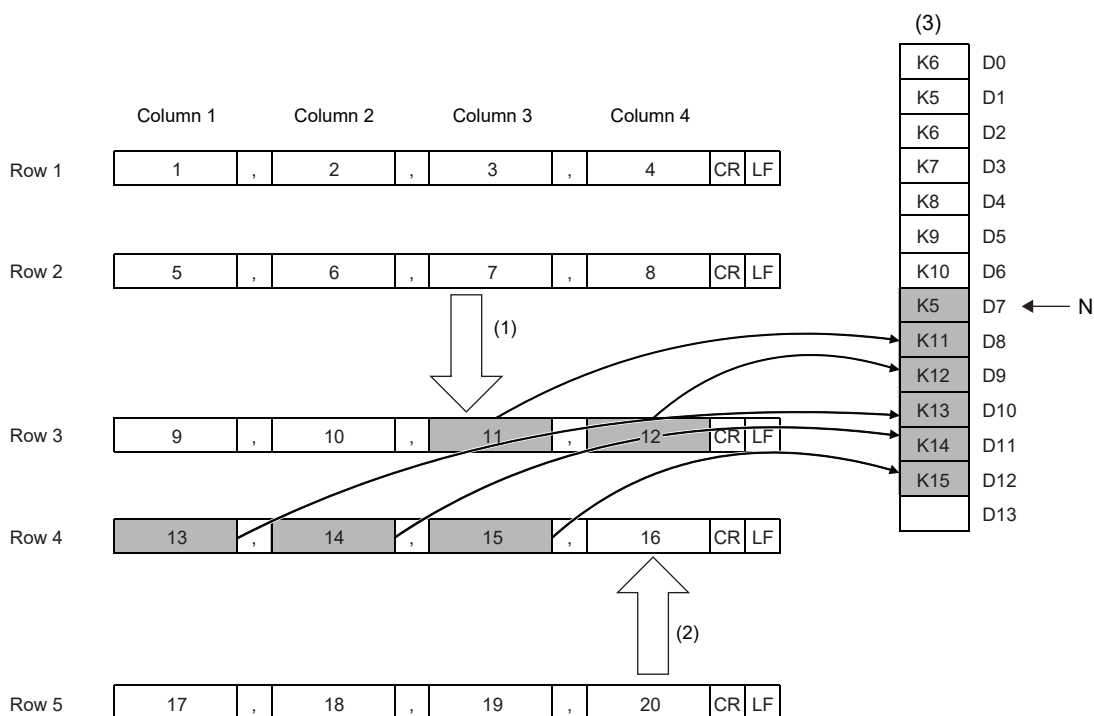
- Execution/completion type: Reading data after converted to CSV format (Decimal (16-bit data))
- Number of read-target data: K6
- File position: 2H
- Number of columns specification: 4H
- Data type specification: Words
- Read start device: D0
- Number of data actually read: 6H



- (1) Starting row
- (2) Next starting position
- (3) Data in the device (loaded data)
- N: Number of data

[Continuing from the last read position]

- Execution/completion type: Reading data after converted to CSV format (Decimal (16-bit data))
- Number of read-target data: K5
- File position: FFFFFFFFH (continuing from the position where the previous reading ends)
- Number of columns specification: 4H
- Data type specification: Words
- Read start device: D7
- Number of data actually read: 5H



(1) Starting row

(2) Next starting position

(3) Data in the device (loaded data)

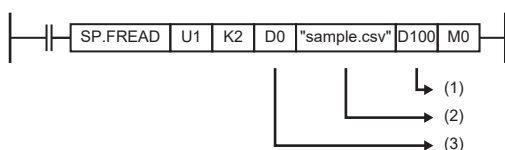
N: Number of data

Point

- When reading data from the position where the previous reading ends, specify the same values for "Execution/End type", "Number of columns", and "Data type specification". If not, data cannot be added correctly from the position where the previous reading ends.
- While reading data from the position where the previous reading ends, if the SP.FREAD instruction with different settings or the SP.FWRITE instruction is executed, data cannot be added correctly from the position where the previous reading ends.
- If another file is specified when data is read continuously from the last read position, the data cannot be normally read from the last read position.
- If data is read continuously from the last read position at the time of initial file reading (for example after the power is turned on the CPU module), the data will be read from the beginning of the file position.
- When reading data after converted to CSV format (string (ASCII data)) has been specified, if the number of read data exceeds the maximum number of read data, the file position will be initialized to the beginning of file even if the reading from the last read position is continued.

Ex.**When out-of-range value or string data is read**

[Data to be read into devices]



(1): Start device for storing the data that has been read

• D100: Number of data actually read

• D101 to D120: Read data

(2): File name

• "sample.csv"

(3): Control data

[Control data]

D0	0100H
D1	-
D2	K20
D3	K0
D4	
D5	K2
D6	K5
D7	K2

D0: Execution/completion type

D1: Completion status

D2: Number of read-target data

D3: Maximum number of read data

D4, D5: File position

D6: Number of columns

D7: Data type specification

[SD memory card]

sample.csv

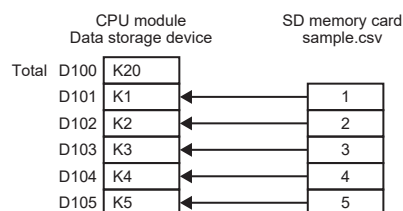
	A	B	C	D	E	F
1	1	2	3	4	5	6
2	1	2	3	4	5	6
3	A	B	C	D	E	F
4	-32769	-32768	65535	65536	32768	-32769
5	A	B	C	D	1.11E+16	F
6	G	H	I	J	2.21E+16	L

The data in the five columns from the second row are transferred to D101 to D120.

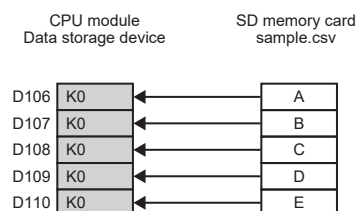


[Loaded data]

■sample.csv 2nd line

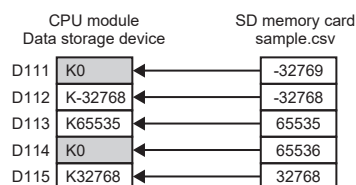


■sample.csv 3rd line

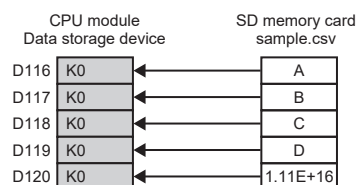


0 is stored in D106 to D110 because all are strings.

■sample.csv 4th line



■sample.csv 5th line



0 is stored in D111 and D114 because -32769 and 65536 are out of the range. 0 is stored in D116 to D120 because all are strings or out of the range.



If values that cannot be expressed in word units in decimal (values out of the range of -32768 to 65535, strings, etc.) are set in the elements of the CSV file, all are converted to 0 and stored in the corresponding devices. Even if 0 is stored due to out-of-range values, they are included in the number of data actually read.

Program example

An example of a program for reading binary data/CSV data from the SD memory card by using the SP.FREAD instruction is shown below.

■Reading binary data

When X0 is turned ON, 100 data starting from the 16th word from the beginning of the "sample.bin" file stored in the SD memory card are stored in D101 to D200.

[Program operation]

1. Control data is created during RUN.
2. The drive contact of X0 is held in M0. When the drive contact is turned ON, the instruction execution complete flag and instruction error completion flag are initialized.
3. The SP.FREAD instruction is executed.*¹
4. Since the instruction execution complete flag and instruction error completion flag are ON only during 1 scan, they are held in the M150 and M151 devices to identify the normal/abnormal completion.

*¹ The instruction is executed after confirming that the following special devices are OFF to prevent simultaneous execution of another file operation instruction.

- SM606 (Memory card disable request)
- SM753 (File being accessed)

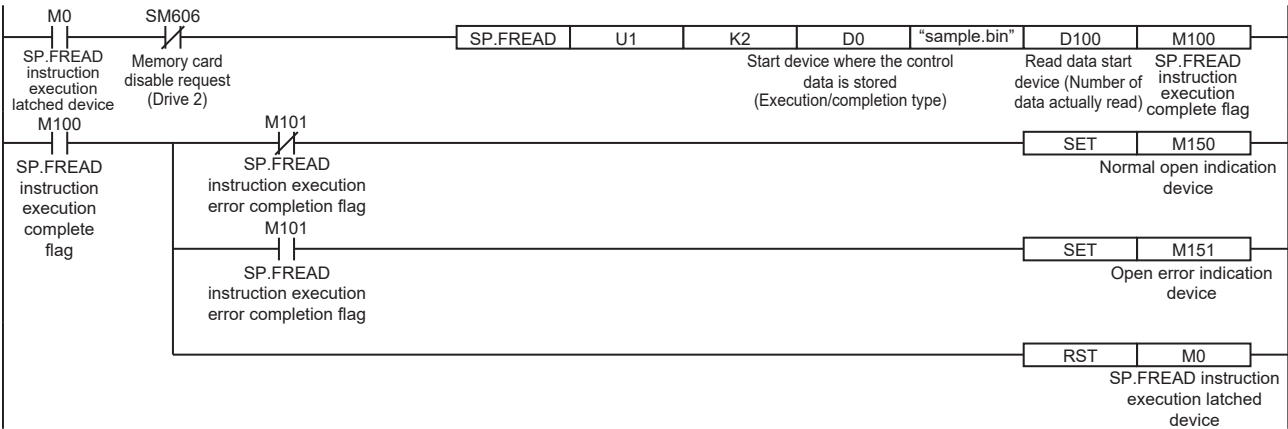
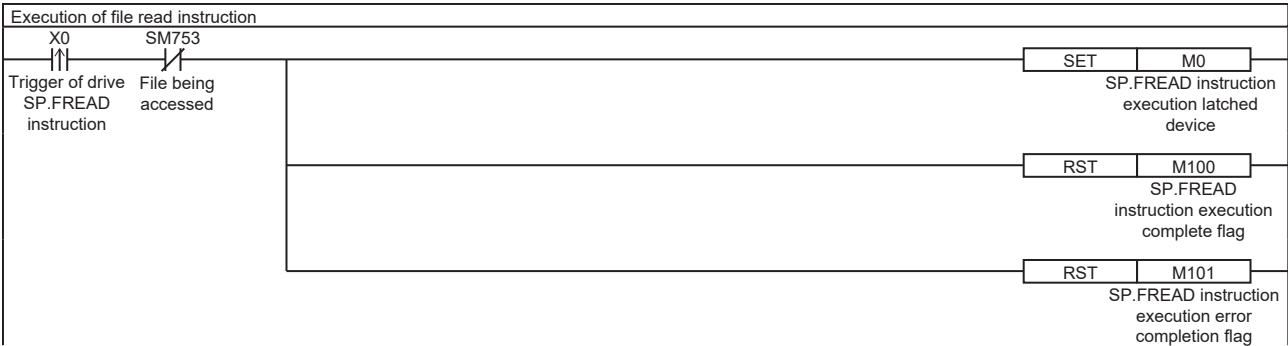
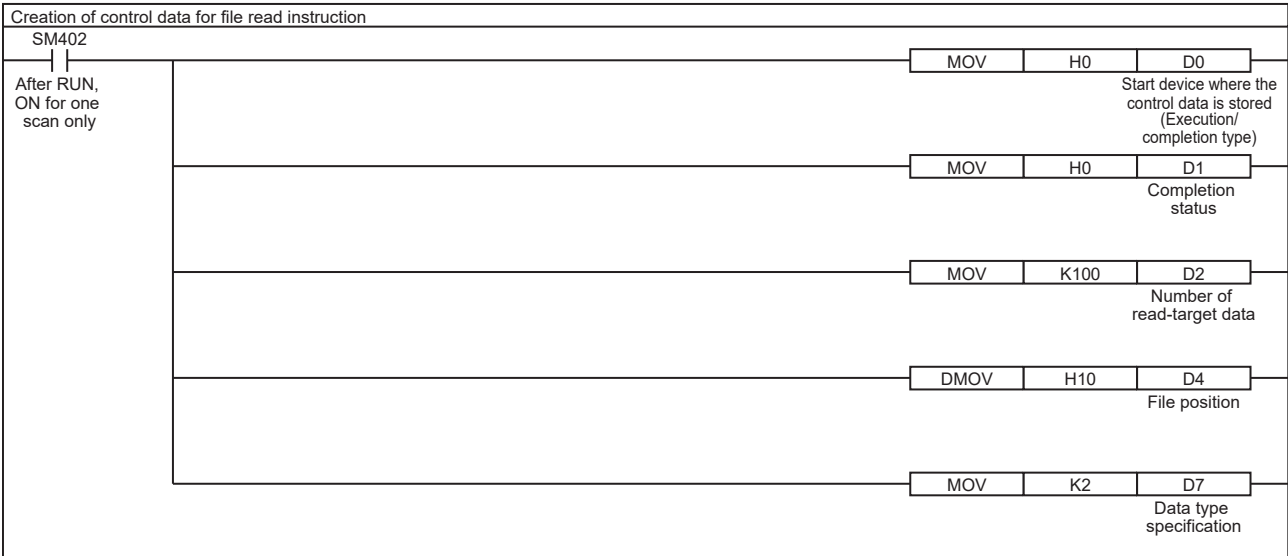
[Devices used]

Device	Description
X0	Trigger of drive SP.FREAD instruction
D0	Start device where the control data is stored <ul style="list-style-type: none"> • D0: Execution/completion type • D1: Completion status • D2: Number of read-target data • D4, D5: File position • D7: Data type specification
D100	Start device for storing the data that has been read <ul style="list-style-type: none"> • D100: Number of data actually read • D101, D200: Read data
M0	SP.FREAD instruction execution latched device
M100	SP.FREAD instruction execution complete flag
M101	SP.FREAD instruction execution error completion flag
M150	Normal open indication device
M151	Open error indication device

[SP.FREAD instruction operand setting]

Operand	Description	Set value
(U)	Dummy	U1
(s1)	Drive specification	K2 (SD memory card)
(d1)	Start device where the control data is stored	D0: 0H (16-bit binary data) D1: 0H (Completed successfully) D2: K100 (100 words are read.) D4, D5: 10H (Data after the 16th word from the beginning are read.) D7: K2 (Unit of data type specified in D0)
(s2)	Start device where the file name is stored	"sample.bin"
(d2)	Start device for storing the data that has been read	D100
(d3)	Bit device that turns on upon completion of the processing	M100: Execution complete flag M101: Execution error completion flag

[Ladder program]



[ST program]

```
//(1) Creation of control data for file read instruction
IF SM402 THEN
D0 := H0; //Execution/completion type (Control data start device)
D1 := H0; //Completion status
D2 := 100; //Number of read-target data
D4:UD := H10; //File position
D7 := 2; //Data type specification
END_IF;

//(2) Processing to start up the drive contact (X0)
IF LDP(TRUE, X0) THEN
//Checking that the file being accessed flag is OFF
IF (SM753 <> TRUE) THEN
SET(TRUE, M0); //Holds drive contact
RST(TRUE, M100); //Initialize instruction execution complete flag
RST(TRUE, M101); //Initialize instruction execution error complete flag
END_IF;
END_IF;

//(3) Execution of file read instruction
IF M0 THEN
//Checking that the memory card disable request is OFF
IF (SM606 <> TRUE) THEN
//EN = TRUE (Enable Input, always execute)
//U = U1 (Dummy)
//S1 = 2 (Drive specification, 2 fixed)
//S2 = "sample.bin"(Start device where the file name is stored)
//D1 = D0 (Start device where the control data is stored)
//D2 = D100 (Start device for storing the data that has been read)
//D3 = M100 (Bit device that turns on upon completion of the processing)
SP_FREAD(TRUE, U1, 2, "sample.bin", D0, D100, M100);
END_IF;
END_IF;

//(4) Checking the instruction execution complete flag
IF M100 THEN
SET((M101 <> TRUE), M150); //Holds instruction execution complete flag
SET(M101, M151); //Holds instruction execution error complete flag
RST(TRUE, M0); //Releasing the drive contact
END_IF;
```

[Execution result]

When the target binary data is read by using the program example, the results are as shown below.

[Read target (sample.bin)]

sample.bin	
ADDRESS	00 01 02 03 04 05 06 07 08 09 0A 0B 0C 0D 0E 0F 0123456789ABCDEF
00000000	5A 5A 5A 5A 5A 5A 5A 5A 5A 5A 5A 5A 5A 5A 5A 5A ZZZZZZZZZZZZZZZZ
00000010	5A 5A 5A 5A 5A 5A 5A 5A 5A 5A 5A 5A 5A 5A 5A 5A ZZZZZZZZZZZZZZZZ
00000020	E9 03 EA 03 EB 03 EC 03 ED 03 EE 03 EF 03 F0 03
00000030	F1 03 F2 03 F3 03 F4 03 F5 03 F6 03 F7 03 F8 03
00000040	F9 03 FA 03 FB 03 FC 03 FD 03 FE 03 FF 03 00 04
00000050	01 04 02 04 03 04 04 04 05 04 06 04 07 04 08 04
00000060	09 04 0A 04 0B 04 0C 04 0D 04 0E 04 0F 04 10 04
00000070	11 04 12 04 13 04 14 04 15 04 16 04 17 04 18 04
00000080	19 04 1A 04 1B 04 1C 04 1D 04 1E 04 1F 04 20 04
00000090	21 04 22 04 23 04 24 04 25 04 26 04 27 04 28 04 ! " # \$ % & ' (.
000000A0	29 04 2A 04 2B 04 2C 04 2D 04 2E 04 2F 04 30 04) * + , - . / : 0.
000000B0	31 04 32 04 33 04 34 04 35 04 36 04 37 04 38 04 1. 2. 3. 4. 5. 6. 7. 8.
000000C0	39 04 3A 04 3B 04 3C 04 3D 04 3E 04 3F 04 40 04 9. : ; . < = > . ? . @.
000000D0	41 04 42 04 43 04 44 04 45 04 46 04 47 04 48 04 A. B. C. D. E. F. G. H.
000000E0	49 04 4A 04 4B 04 4C 04 I. J. K. L.

[Read result]*1

Device Name	F	E	D	C	B	A	9	8	7	6	5	4	3	2	1	0	Current Value	String
D99	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0000	..
D100	0	0	0	0	0	0	0	0	0	1	1	0	0	1	0	0	0064	..
D101	0	0	0	0	0	0	1	1	1	1	1	0	1	0	0	1	03E9	064
D102	0	0	0	0	0	0	1	1	1	1	1	0	1	0	1	0	03EA	..
D103	0	0	0	0	0	0	1	1	1	1	1	0	1	0	1	1	03EB	..
D104	0	0	0	0	0	0	1	1	1	1	1	0	1	1	0	0	03EC	..
D105	0	0	0	0	0	0	1	1	1	1	1	0	1	1	0	1	03ED	..
(2)																		
D196	0	0	0	0	0	1	0	0	0	1	0	0	1	0	0	0	0048	..
D197	0	0	0	0	0	1	0	0	0	1	0	0	1	0	0	1	0049	..
D198	0	0	0	0	0	1	0	0	0	1	0	0	1	0	1	0	004A	..
D199	0	0	0	0	0	1	0	0	0	1	0	0	1	0	1	1	044B	..
D200	0	0	0	0	0	1	0	0	0	1	0	0	1	1	0	0	044C	..
D201	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0000	..
D202	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0000	..

- (1) 100 words from 03E9 to 044C are described. In the program example, the data is read starting from the 16th word, and so 03E9 (boxed) is stored in D101.
- (2) In the program example, 100 words of data are read, and so the number of data actually read is stored in D100, and the read data are stored in D101 to D200.

*1 Device/Buffer Memory Batch Monitor (hexadecimal display) of engineering tool

■Reading data after converted to CSV format

When X0 is turned ON, 20 words of data in the five columns in the second to fifth rows in the “sample.csv” file stored in the SD memory card are stored in D101 to D120.

[Program operation]

1. Control data is created during RUN.
2. The drive contact of X0 is held in M0. When the drive contact is turned ON, the instruction execution complete flag and instruction error completion flag are initialized.
3. The SP.FREAD instruction is executed.*¹
4. Since the instruction execution complete flag and instruction error completion flag are ON only during 1 scan, they are held in the M150 and M151 devices to identify the normal/abnormal completion.

*¹ The instruction is executed after confirming that the following special devices are OFF to prevent simultaneous execution of another file operation instruction.

- SM606 (Memory card disable request)
- SM753 (File being accessed)

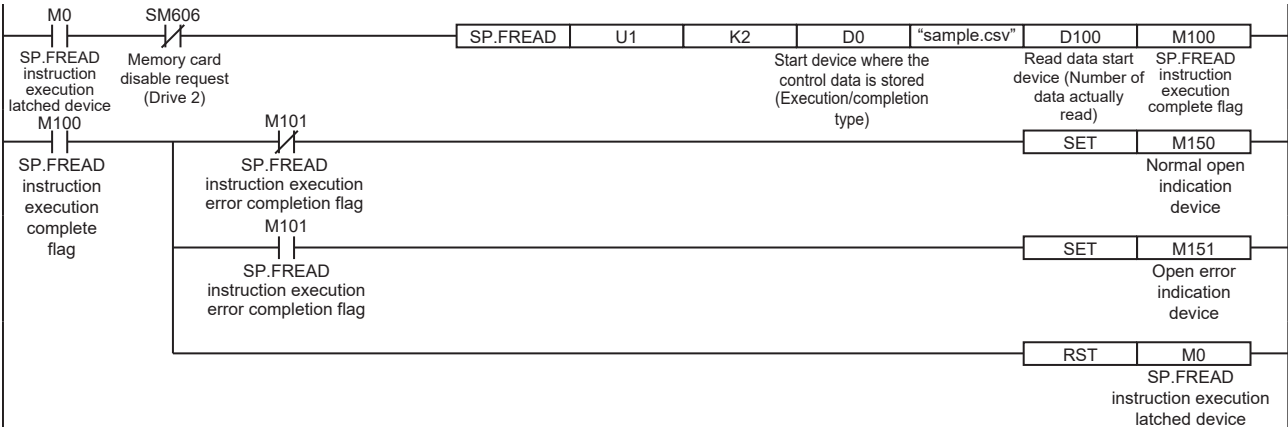
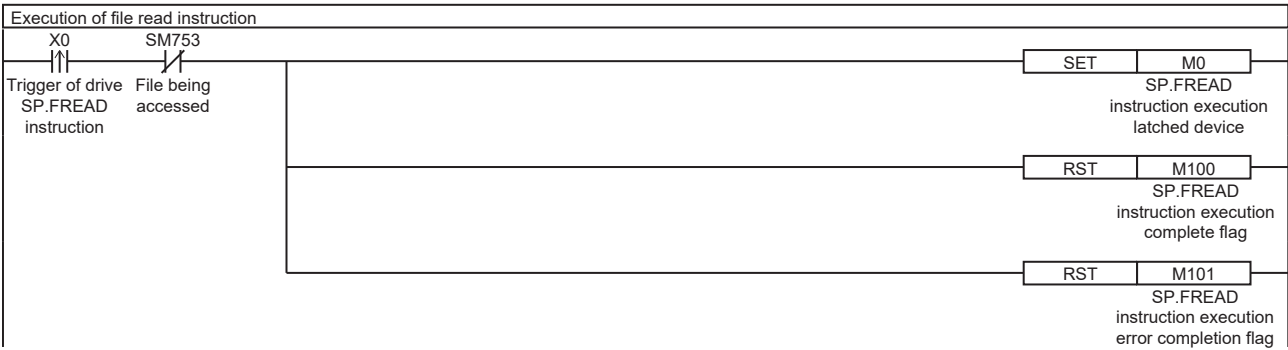
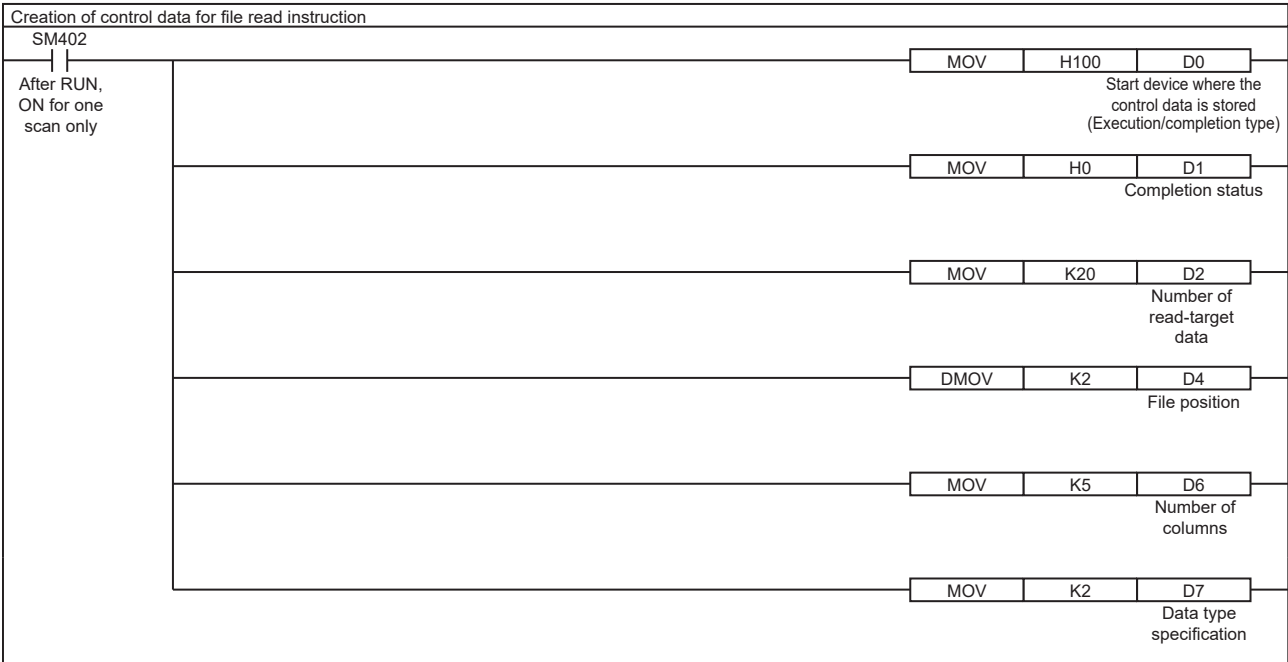
[Devices used]

Device	Description
X0	Trigger of drive SP.FREAD instruction
D0	Start device where the control data is stored <ul style="list-style-type: none"> • D0: Execution/completion type • D1: Completion status • D2: Number of read-target data • D4,D5: File position • D6: Number of columns • D7: Data type specification
D100	Start device for storing the data that has been read <ul style="list-style-type: none"> • D100: Number of data actually read • D101 to D120: Read data
M0	SP.FREAD instruction execution latched device
M100	SP.FREAD instruction execution complete flag
M101	SP.FREAD instruction execution error completion flag
M150	Normal open indication device
M151	Open error indication device

[SP.FREAD instruction operand setting]

Operand	Description	Set value
(U)	Dummy	U1
(s1)	Drive specification	K2 (SD memory card)
(d1)	Start device where the control data is stored	D0: 0H (16-bit binary data) D1: 0H (Completed successfully) D2: K20 (20 words are read.) D4,D5: K2 (Reading from the second row) D6: K5 (Reading the data in 5 columns) D7: K2 (Unit of data type specified in D0)
(s2)	Start device where the file name is stored	“sample.csv”
(d2)	Start device for storing the data that has been read	D100
(d3)	Bit device that turns on upon completion of the processing	M100: Execution complete flag M101: Execution error completion flag

[Ladder program]



[ST program]

```
//(1) Creation of control data for file read instruction
IF SM402 THEN
D0 := H100; //Execution/completion type (Control data start device)
D1 := H0; //Completion status
D2 := 20; //Number of read-target data
D4:UD := 2; //File position
D6 := 5; //Number of columns
D7 := 2; //Data type specification
END_IF;

//(2) Processing to start up the drive contact (X0)
IF LDP(TRUE, X0) THEN
// Checking that the file being accessed flag is OFF
IF (SM753 <> TRUE) THEN
SET(TRUE, M0); //Holds drive contact
RST(TRUE, M100); //Initialize instruction execution complete flag
RST(TRUE, M101); //Initialize instruction execution error complete flag
END_IF;
END_IF;

//(3) Execution of file read instruction
IF M0 THEN
//Checking that the memory card disable request is OFF
IF (SM606 <> TRUE) THEN
//EN = TRUE (Enable Input, always execute)
//U = U1 (Dummy)
//S1 = 2 (Drive specification, 2 fixed)
//S2 = "sample.csv" (Start device where the file name is stored)
//D1 = D0 (Start device where the control data is stored)
//D2 = D100 (Start device for storing the data that has been read)
//D3 = M100 (Bit device that turns on upon completion of the processing)
SP_FREAD(TRUE, U1, 2, "sample.csv", D0, D100, M100);
END_IF;
END_IF;

//(4) Checking the instruction execution complete flag
IF M100 THEN
SET((M101 <> TRUE), M150); //Holds instruction execution complete flag
SET(M101, M151); //Holds instruction execution error complete flag
RST(TRUE, M0); //Releasing the drive contact
END_IF;
```


[[Execution result]]

When the target CSV data is read by using the program example, the results are as shown below.

[Read target (sample.bin)]

	A	B	C	D	E	F
1	1	2	3	4	5	6
2	1	2	3	4	5	(1) 6
3	A	B	C	D	E	F
4	-32769	-32768	65535	65536	32768	-32769
5	G	H	I	J	1.11E+16	L
6	M	N	O	P	2.22E+16	R

[Read result]*1

Device Name	F	E	D	C	B	A	9	8	7	6	5	4	3	2	1	0	Current Value	String
D100	0	0	0	0	0	0	0	0	0	0	0	1	0	1	0	0	20	
D101	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	
D102	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	2	
D103	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	3	
D104	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	4	
D105	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	5	
D106	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
D107	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
D108	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
D109	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	(2)
D110	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
D111	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
D112	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	-32768	
D113	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	-1	□□
D114	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
D115	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	-32768	
D116	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
D117	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
D118	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
D119	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
D120	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	

- (1) In the program example, five columns in each row from the second row are read, and so 1 (boxed) is stored in D101.
- (2) In the program example, 20 words of data are read, and so the number of data actually read is stored in D100, and the read data are stored in D101 to D120. For values/strings that cannot be read in the execution/completion type format, the stored device values of the read data are 0. Even if the device values are 0, they are included in the number of read data.

*1 Device/Buffer Memory Batch Monitor (decimal display) of engineering tool

Precautions

- The SP.FREAD instruction cannot be executed in user interrupt programs. An error (3582H) will occur.
- The SP.FREAD instruction cannot be executed while SM606 (Memory card disable request) is ON. When SM606 is turned ON during execution of the instruction, the program will terminate abnormally. (The data that has been read before SM606 is turned ON will be stored in the devices.)
- When reading multiple elements at reading data after converted to CSV format, secure devices sufficient for the total size of the elements in the read data area before executing the instruction. Since read data is stored from (d2)+1, the number of words required to be secured as (d2) is ((total number of words for each element) + 1) words.
- The files in the system folder (\$MELPRJ\$) cannot be handled by the SP.FREAD instruction. If an access to any file in the system folder (\$MELPRJ\$) is requested, a calculation error (3405H) will occur.
- If the storage device exceeds the end of the device range, an error (2820H) will occur.
- The SP.FREAD instruction cannot be executed simultaneously with the SP.DEVST instruction, the SP.FTPPUT instruction, and the SP.FTPGET.
- Do not disconnect the power or remove the SD memory card during execution of the SP.FREAD instruction. (The file may be damaged, or an error may occur.)

Point

When 16-bit data (0000H, 0100H or 0120H) has been specified for Execution/completion type (d1)+0, the setting range of the number of read-target data (d1)+2 is as shown below.

- Up to 64512 points when only label is used.
- Up to 32767 points (when the file register (R) is used) when only device is used.

Operation error

Error code (SD0/SD8067)	Description
2820H	The storage device of the control data (d1) exceeds the end of the device range.
	The bit label digits specified in (s2) and (d2) are unacceptable settings (the number of digits is not K4).
	The size of the read data exceeds the read device size.
3405H	The drive specified by (s1) is not the one for the SD memory card.
	Any value that is set in the device specified by (d1) and later as control data is out of the range.
	The file name string specified by (s2) cannot be read.
	<ul style="list-style-type: none">• The number of characters of the string in the file name specified exceeds the range.• An inhibited value is set.• The specified file name string ends with a delimiter.• The system folder (\$MELPRJ\$) directly under the route folder is specified.• A half-width period is specified at the end of the specified file name string or immediately before each delimiter.
3427H	An invalid combination of (d1)+0 (Execution/completion type) and (d1)+7 (Data type specification) is specified.
3582H	The SP.FREAD instruction is executed in an interrupt program.

If the SP.FREAD instruction completes with an error, an error code is stored in the device specified by (d1)+1. (Note that an error code is not stored if the instruction results in an operation error.)

For the error code stored in (d1)+1, refer to the following.

 Page 579 Error codes generated for file operation instructions