



SD Logger Script Language Reference Format

Version 2.03 March 3, 2020

■■Condition■■

Line

The maximum number of characters per line is 127 characters, excluding the line feed character.

Statements must be written one per line. It cannot be described over multiple lines.

The maximum number of lines of the script excluding comment statements is as follows.

Item	SD logger	SD logger2	SD logger3	SD logger4	Unit
Max lines	256	256	512	512	line

Data

The total number of bytes of data and text that can be written in data statements, WAIT DATA statements, and LOG statements is as follows.

Item	SD logger	SD logger2	SD logger3	SD logger4	Unit
Total No. of bytes	512	512	1024	1024	byte

Loop

Loops can be nested and written up to eight times.

Process

Up to eight processes can be written in the script, including the main process.

■■Data statement■■

Statements with "/" or ":" at the beginning of a line are interpreted as data statements, and the data described is sent as is.

Text data sentence

/<data>

Data statements with a "/" at the beginning of the line are interpreted as if the data was described in text. From the character following "/" to before the line feed code is sent, including spaces and tab codes. No line feed code is sent.

Binary data statement

:<data>

A data statement with a ":" at the beginning of a line is interpreted as if the data is written in hexadecimal.

Hexadecimal characters can be uppercase or lowercase. Spaces and tabs are interpreted as delimiters and are not sent on their own. If there is no delimiter, it will be interpreted with two digits.

■■Control statement■■

Statement preceded by "#" are control statement. There are the following types of control statement.

●LOOP statement

#LOOP *<count>* | **EVER**

Repeat *<count>* times or infinite times until the corresponding END statement.
If "EVER" is specified, "0" is specified, or the parameter is omitted, repeats indefinitely.

<i><count></i>	0~60000
----------------------	---------

●END statement

#END

Indicates the end of a block such as the corresponding LOOP statement.
If the blocks are nested, it has the effect of terminating the innermost block.

●WAIT TIME 文

#WAIT TIME *<time>*
<time> := *<number>* *<unit>*
<unit> := **MS** | **S** | **M**

Wait for the specified time. Specify the wait time in *<time>*. *<time>* is expressed with the unit *<unit>*. If *<number>* is omitted, it is assumed to be "1". If *<unit>* is omitted, it is assumed to be "S". If *<time>* is omitted, 1 second is assumed. If you specify a time of zero, do not wait at all.
That is, it is the same as writing nothing.

<i><unit></i>	Unit	Range
MS	millisecond	0MS~60000MS
S	second	0S~60000S
M	minute	0M~999M

The wait time of the WAIT TIME statement includes the error associated with the time accuracy of the hardware.

Please refer to "Instruction Manual" for the time accuracy of the hardware.

Only the WAIT statement and the FCHANGE statement have time lapse in the script language.

Other statements are assumed to execute with zero execution time.

●WAIT DATA statement

#WAIT DATA *<data>*

Wait for the specified data to be received.
The way of writing data is the same as the data statement.

●WAIT BYTE statement

#WAIT BYTE <byte>

Waits for the specified number of bytes of data to be received.

If <byte> is omitted, it is assumed to be "1".

If you specify zero in <byte>, nothing will wait. That is, it is the same as writing nothing.

<byte>	0~60000
--------	---------

●WAIT CTS statement

#WAIT CTSON

#WAIT CTSOFF

Wait for the CTS signal to be turned ON or OFF.

The CTS signal is a signal input to the 8th pin of the D-Sub 9-pin connector and is recognized as ON at a high level.

●WAIT EX statement

#WAIT EX1ON

#WAIT EX1OFF

#WAIT EX2ON

#WAIT EX2OFF

#WAIT EX3ON

#WAIT EX3OFF

#WAIT EX4ON

#WAIT EX4OFF

Wait for the extension signal to turn ON or OFF.

When the input of the extension signal is low level, it is recognized as ON.

The extension signals correspond to the pins shown in the table below according to the model.

【SD logger 4】

Extension signal	CN7
EX1	Pin 2
EX2	Pin 3
EX3	Pin 4
EX4	Pin 5

【SD logger 3】

Extension signal	CN6
EX1	It can not be used.
EX2	It can not be used.
EX3	Pin 3
EX4	Pin 2

●LOG statement

#LOG <text>

Writes the specified <text> to the card.

<text> can include the format specification characters shown in the writing.

@c	Log counter. Number of times this log was taken. It starts from 0 when started, and is incremented by 1 each time this log is taken. When it exceeds 4294967295, it returns to 0.
@Y	Year. Two-digit ASCII character. 【SD loggers 3 and 4 only】
@M	Month. Two-digit ASCII character. 【SD loggers 3 and 4 only】
@D	Day. Two-digit ASCII character. 【SD loggers 3 and 4 only】
@h	Hour. Two-digit ASCII character (24 hour clock). 【SD loggers 3 and 4 only】
@m	Minutes. Two-digit ASCII character. 【SD loggers 3 and 4 only】
@s	Seconds. Two-digit ASCII character. 【SD loggers 3 and 4 only】
@@	The ASCII character "@" itself.
@r	Return code. (0xD)
@n	Line feed code. (0xA)

※The maximum number of characters that can be written in <text> is as follows, together with data in WAIT DATA statements and data statements.

Item	SD logger	SD logger2	SD logger3	SD logger4	Unit
Total No. of bytes	512	512	1024	1024	byte

※The maximum number of characters that can be written in one LOG statement is 127 characters.

【適応】

Item	SD logger	SD logger2	SD logger3	SD logger4	
@Y, @M, @D, @h, @m, @s	×	×	○	○	

●RTS statement

#RTS ON

#RTS OFF

Turn the RTS signal ON or OFF.

The RTS signal is output to pin 7 of the D-Sub 9-pin connector, and outputs a high level when turned ON.

●FCHANGE statement

#FCHANGE

Closes the currently logged log file and starts logging to a new logging file.

The name of the new logging file will be one step further according to the naming convention. Execution of this control statement may take some time.

●PAUSE statement

#PAUSE

Stops logging (recording on the card) from the next data received. When the power is turned on, it is in the logging state.

【Adaptation】

Item	SD logger	SD logger2 SD logger3	SD logger4 Before V392X	SD logger4 After V934X	
#PAUSE statement	×	×	×	○	

●RESUME statement

#RESUME

Logging (recording on the card) resumes from the next data received.

【Adaptation】

Item	SD logger	SD logger2 SD logger3	SD logger4 Before V392X	SD logger4 After V934X	
#RESUME statement	×	×	×	○	

●PROCESS statement

#PROCESS

When the script starts, all PROCESS statements start executing at the same time.

The process stops execution when it finds the next PROCESS statement.

It is not necessary to write a PROCESS statement in the first process of the script, but it is OK to do so.

●NOP statement

#NOP

Do nothing.

It can be used to prevent WAIT DATA statements from being linked.

■ Configuration statement ■

● f: ENCODE statement

#f:ENCODE <code>

When the specified <code> is received, another same code is added to the logging data (encoding function).

In <code>, write one character to be encoded (escape character). Any specification after the second character is ignored.

If ":" is written first, it is interpreted as a one- or two-digit hexadecimal character string thereafter.

To specify "/" as an escape character, write "/:".

To specify "/" as an escape character, write "//".

:	The following character is interpreted as a two-digit or one-digit hex string.
/	This character is ignored, and the next character is valid.

● f: STOPBITS statement

#f:STOPBITS <n>

Specify the stop bit for serial communication.

The stop bit is "1" by default, and is changed to "2" stop bit only when "2" is specified for <n>. If a value other than "2" is specified for <n>, it is interpreted as "1" stop bit.

● f: EX statement

#f:EX<n> <f>

<n> := 1 | 2 | 3 | 4

<f> := IN

Set the function of the extension signal.

The function of the extension signal is currently only the input signal (= IN).

The extension signal is the input signal (= IN) by default.

● f: OMIT statement

#f:OMIT <data>

Register a code that is not logged even if it is received.

Registered codes are not logged when received.

The order of <data> is meaningless and is compared simply for each byte.

<data> is written in the same way as data statements. You can use "/" to write text and ":" to write hex.

The code that can be registered in <data> is up to 10 bytes in total.

Even if the code registered in the f: OMIT statement is specified in the f: ENCODE statement, the encoding function does not work (OMIT priority).

[Adaptation] Applicable to SD loggers 3 and 4.

●f:LFEXT statement

#f:LFEXT <ext>

Specify the extension of the log file. Log files have the extension specified here.
The characters that can be used for the extension are shown in the following table. If lowercase is specified, it is converted to uppercase. Up to three characters can be specified.
If not specified here, the log file extension is "LOG" (default value).

	2	3	4	5	6	7
0		0	@	P	`	p
1	!	1	A	Q	a	q
2	"	2	B	R	b	r
3	#	3	C	S	c	s
4	\$	4	D	T	d	t
5	%	5	E	U	e	u
6	&	6	F	V	f	v
7	'	7	G	W	g	w
8	(8	H	X	h	x
9)	9	I	Y	i	y
A	*	:	J	Z	j	z
B	+	;	K	[k	{
C	,	<	L	¥	l	
D	-	=	M]	m	}
E	.	>	N	^	n	~
F	/	?	O	_	o	

	Characters that can be used
	Characters that cannot be used

■■Comment statement■■

Statements preceded by ";" are comment statements. A free comment can be written after ";" in the comment statement.

■ ■ Concatenation of WAIT DATA statements ■ ■

WAIT DATA statements written on multiple lines are concatenated into one.

Concatenated WAIT DATA statements are considered the same as one WAIT DATA statement, and proceed to the next instruction of the WAIT DATA statement only if the data written in them is received consecutively in that order without gaps .

In a single WAIT DATA statement that is not linked, the statement advances for each data item indicated in each WAIT DATA statement, so even if any data is received in the middle, the script proceeds.

Use a NOP statement to avoid concatenation of WAIT DATA statements that are written continuously on multiple lines.

WAIT DATA statements are concatenated even if a comment statement is inserted in the middle.

The following shows examples of when WAIT DATA statements are linked and when they are not linked.

Script	received data	Action
#WAIT DATA /ABC #WAIT DATA /XYZ	ABCXYZ	Proceed to next instruction
	ABC123XYZ	Do not proceed to next instruction
	ABC123456	Do not proceed to next instruction
#WAIT DATA /ABC #NOP #WAIT DATA /XYZ	ABCXYZ	Proceed to next instruction
	ABC123XYZ	Proceed to next instruction
	ABC123456	Advance only the first row

■Revision history■

2020/03/03	2.03	Describes adaptation to SD logger 4 V934X or later.
2020/02/21	2.02	Corrected the error that the log start / stop function was set to LSTART / LSTOP. Rather, PAUSE / RESUME. Deleted description of CF logger.
2020/01/29	2.01	Changed the model name of SD Logger 4 Advanced Edition.
2019/06/05	2.0	Added WAIT CLOCK statement and LSTART / LSTOP statement. (Compatible with SD Logger 4 Advanced Edition) Added description of SD logger 3 and SD logger 4 for external input.
2015/04/21	1.9	Added f: LFEXT statement. Added SD to document name.
2014/09/11	1.8	Added f: OMIT statement. Add date and time to LOG statement and format. (Compatible with SD logger 3 <DT-MC03, DT-MCK3>)
2013/06/27	1.7	Added WAIT TIME statement about time accuracy. Added FCHANGE sentence to the explanation of sentence with time lapse.
2009/10/21	1.6	Added file exchange (FCHANGE) statement.
2008/07/03	1.5	Added external input wait function (WAIT CTSON statement, etc.).
2008/02/22	1.4	Added f: STOPBITS statement.
2007/04/25	1.3	Added RTS statement.
2006/07/12	1.2	Added f: ENCODE statement. Added LOG statement.
2005/03/24	1.1	Added concatenation of NOP statement and WAIT DATA statement.