

Converting shift JIS character string to Unicode (with byte order mark)

SJIS2WSB(P)

FX5S FX5UJ FX5U FX5UC

These instructions convert a shift JIS character string to a Unicode character string, and adds a byte order mark to the start of the converted data.

Ladder

ST

ENO:=SJIS2WSB(EN,s,d);
ENO:=SJIS2WSBP(EN,s,d);

FBD/LD

Setting data

■Description, range, data type

Operand	Description	Range	Data type	Data type (label)
(s)	Shift JIS string (up to 255 characters) to be converted or start device containing the shift JIS string	—	String	ANYSTRING_SINGLE
(d)	Start device for storing the converted Unicode character string	—	Unicode string	ANYSTRING_DOUBLE
EN	Execution condition	—	Bit	BOOL
ENO	Execution result	—	Bit	BOOL

■Applicable devices

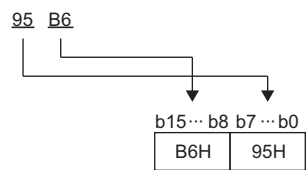
Operand	Bit	Word			Double word		Indirect specification	Constant			Others
	X, Y, M, L, SM, F, B, SB, S	T, ST, C, D, W, SD, SW, R	U□\G□	Z	LC	LZ		K, H	E	\$	
(s)	—	○	—	—	—	—	○	—	—	○	—
(d)	—	○	—	—	—	—	○	—	—	—	—

Processing details

- These instructions convert the shift JIS character string in the device specified by (s) to the Unicode character string, add a byte order mark to the start of the converted data, and store it in the device specified by (d).
- Specify the shift JIS string in (s) in big endian.

Ex.

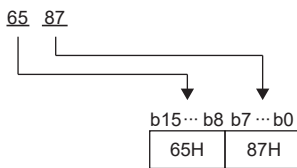
A shift JIS string “95B6H” is specified with “B695H”.



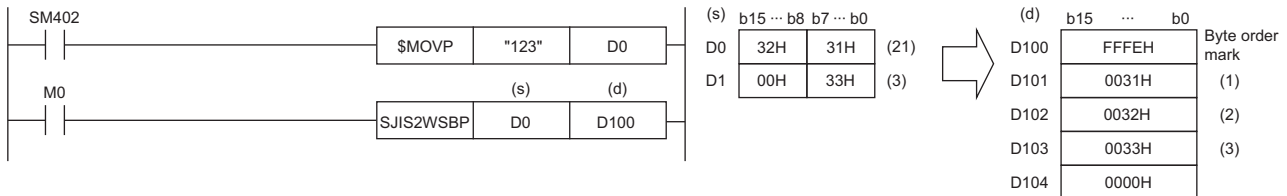
- The Unicode string in (d) is stored in little endian.

Ex.

A Unicode string "6587H" is specified.



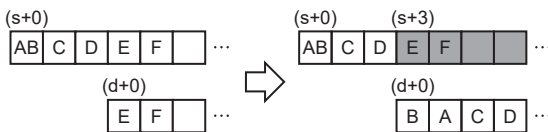
- The following figure shows the operation for converting shift JIS to Unicode.



Precautions

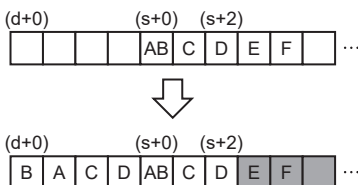
- When the data length of the ASCII (Shift JIS) string specified in (s) is 0, only the byte order mark and one character (one word) of termination NULL are written to (d)+0.
- If the ASCII (Shift JIS) string specified in (s) contains a character code that cannot be converted, the string to the character causing the error including the byte order mark is written to the device No. specified in (d).
- If the device ranges specified in (s) and (d) overlap, the overlap may be detected in advance. The operation in each case of overlap of the device ranges is shown below.

[When the overlap is detected in advance]



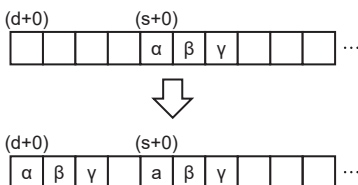
If the overlap of the device ranges is detected in advance, the string to the overlapped point is converted and written to (d).
Since the overlap of the device ranges in (s+3) is revealed in advance, the string in (s+0) to (s+2) is converted and written to (d+0) to (d+3).
An overlap error (2821H) occurs.

[When the overlap cannot be detected in advance (error completion)]



Since the device ranges overlap when the string in (s+2) is converted, the next destination to store is not found.
An overlap error (2821H) occurs.

[When the overlap cannot be detected in advance (normal completion)]



As the result of string conversion, the device ranges do not overlap, and the program terminates normally.

Operation error

Error code (SD0/SD8067)	Description
2820H	0000H does not exist between setting areas in the device/label memory after the device No. specified in (s).
2821H	The ranges of data in the devices specified by (s) and (d) are overlapping.
3401H	The range of data in the device specified by (s) includes a character code that cannot be converted.
3405H	The character string in the device specified by (s) exceeds 512 characters.*1
3406H	All converted shift Unicode strings cannot be stored to the number of points to the last No. in each setting area of the relevant device/label memory after the device No. specified in (d).

*1 A two-byte character such as a kanji character represented in shift JIS code should be counted 2.