“Maple Bus 1.0”
Function Type Specifications

FT_6 : Keyboard

Revision 0.90

Issued by
Sega Enterprises, Inc.
CS Development and Production Dep't, Second Division

Revision history

0.75	1998/04/19	
0.78	1998/05/25	Contents of Free Device Status revised, and unifying information added.
		Added description of the characters that can be used for Fixed Device Status and Free Device Status.
0.80	1998/06/10	Chart for explaining Function key features partly revised.
		84-, 85-, 87-, 88-key added to recommended operating system.
		Keyboard language setting contents changed.
		Keyboard type (87-key, 88-key) added.
		Wraparound contents partly changed.
		Caution added to explanation of the handling of the henka-key's key code.
0.82	1998/08/20	Key code tables divided into US, UK and JPN tables.
0.90	1999/04/30	Display location of diagram numbers unified under diagrams.
		Keyboard items extensively revised.
		Left Win, right Win, and application key designations changed to special keys S1, S2, and S3.

Contents

1. Keyboard Function Conditions

1.1 Keyboard Function Definition

The Keyboard function is an input type man/machine interface serving for input and editing of characters and numerals, and for control of screen operations. It provides

(1) External shape is not prescribed, but operation must be compatible with every product.
(2) The minimum function complement must be provided.
(3) The Keyboard function must comply to the Maple Bus 1.0 Standard Specifications.

1.2 Function Outline

The Keyboard function elements are listed below.

- Alphanumeric keys :A - Z, 0 - 9
- Symbol keys :- ^ \ @ [ ] : ; . / \ ! " # $ % & ' ( ) ~ | ` { + * } < > ? _ ,
- Function keys :F1 - F24
- Editing keys :Tab, Backspace, Space, CapsLock, Insert, Delete, Home, End, PageUp, PageDown, PrintScreen, ScrollLock, Pause, Enter, Esc, Hankaku/Zenkaku, Muhenkan, Zenkouho Henkan (jikouho), Katakana Hiragana, Stop, Again, Undo, Cut, Copy, Paste, Find, Alternate, Ease, Cancel, Clear, Prior, Return, Separator, Out, Open, Clear/Again, CrSel/Props, ExSel, SysReq, Power, Mute, Volume Up, Volume Down
- Cursor keys : ← ↑ → ↓
- Control keys :Ctr, Alt, Shift
- Numeric keypad :0 - 9, NumLock, Enter . / + - * , =
- Special keys :S1,S2,S3
- Error code :No operation, rollover error, POST Fail, undefined error

*Composition of elements may vary according to language.

The minimum required elements for the Keyboard function are as follows.

- Alphanumeric keys :A - Z, 0 - 9
- Symbol keys : - ^ \ @ [ ] : ; . / \ ! " # $ % & ' ( ) ~ | ` { + * } < > ? _ ,
- Function keys :F1 - F6
- Editing keys :Tab, Backspace, Space, CapsLock, Enter, Esc
- Cursor keys :← ↑ → ↓
- Control keys :Ctr, Alt, Shift
- Error code :Rollover error

Unless these elements are provided, a function cannot be defined as Keyboard function.

1.3 Configuration Details

This section gives a detailed description of Keyboard function elements.

(1) Alpha keys, symbol keys: A - Z, - ^ \ @ [ ] : ; . / \ ! " # $ % & ' ( ) = ~ | ` { + * } < > ? _ ,
These are digital keys that take only two states: pressed/not pressed (ON/OFF). When pressed, the corresponding key scan code is output. The corresponding letter (A - Z), numeral (0 - 9), or symbol is displayed on the screen. To input the symbol printed i the top left corner of the key (for example ` { + * } etc.), the Shift key must be held down while pressing the key.

(2) Function keys: F1 – F24
These are digital keys that take only two states: pressed/not pressed (ON/OFF). When pressed, the corresponding key scan code is output and a function assigned to the key (such as displaying help) is executed.

(3) Editing keys: Tab, Backspace, Space, CapsLock, Insert, Delete, Home, End, PageUp, PageDown, PrintScreen, ScrollLock, Pause, Enter, Esc, NumLock, Hankaku/Zenkaku, Muhenkan, Zenkouho Henkan (jikouho), Katakana Hiragana, etc.

These are digital keys that take only two states: pressed/not pressed (ON/OFF). When pressed, the corresponding key scan code is output. These keys perform auxiliary functions when creating a document with the alphanumeric keys and symbol keys. Some repreentative functions are listed below in Table 1.3.1.

Editing key	Function Example
Space	Inserts a blank (space). Also uses as conversion.
Tab	Moves the cursor to a position defined by the word processing software.
CapsLock	Fixes the alpha keys A - Z to output only capitals (same as when Shift key is pressed).
Insert	Toggles the word processing software between overwrite mode and insert mode.
Delete	Deletes the character at the cursor without changing the cursor position.
Home	Moves the cursor to the home position.
End	Moves the cursor to the end of the line.
PageUp	Scrolls one screen up in word processing software.
PageDown	Scrolls one screen down in word processing software.
PrintScreen	Saves the current display contents as an image in the buffer.
ScrollLock	Serves as locking key, for example to keep the cursor position while scrolling.
Enter	Confirms an input or inserts a line feed/carriage return.
Esc	Cancels a command or other input before it has been confirmed.
NumLock	Toggles the numeric keypad ON and OFF.
BackSpace	Deletes the character before the cursor.
Hankaku/Zenkaku	Switches between zenkaku (dual-byte) and hankaku (single-byte) characters in Japanese input. Also activates the kanji mode.
Muhenkan	Enters characters without conversion in Japanese input.
Zenkouho Henkan (jikouho)	Displays a list of henkan (conversion) candidates for kana-to-kanji conversion.
Katakana Hiragana	Toggles between katakana and hiragana in Japanese input for word processing software etc.

Fig. 1-1 Editing key function example

(4) Cursor keys: ← ↑ → ↓

These are digital keys that take only two states: pressed/not pressed (ON/OFF). When pressed, the corresponding key scan code is output. The keys serve to move the cursor in the indicated direction.

(5) Control keys: Ctr, Alt, Shift

These are digital keys that take only two states: pressed/not pressed (ON/OFF). When pressed, the corresponding key scan code is output. Some function examples are listed below in Table 1.3.2.

Control key	Function example
Ctrl	Used in conjunction with other keys for shortcuts etc.
Alt	Used mainly as a menu key. Also used in conjunction with other keys for shortcuts etc.
Shift	Used to switch between lower case and upper case and to input symbols

Fig. 1-2 Control key function example

(6) Numeric keypad: 0 - 9, NumLock, Enter . / + - *

These are digital keys that take only two states: pressed/not pressed (ON/OFF). When pressed, the corresponding key scan code is output. The keys serve for numeric input, equation input, etc.

(7) Special keys: S1,S2,S3

These are digital keys that take only two states, pressed or not pressed (ON/OFF). When pressed (ON), the corresponding key code is output. Used to display the operating system Start menu, etc.

(8) Error code

This function is provided to deal with keyboard errors. It is not associated with a physical key. When seven or more keys are pressed simultaneously or when a sneak current has occurred, an error code is returned to the host.

1.4 Recommended Configurations

To ensure compatibility among Keyboard function products, the following configurations are recommended.

(1) 84-key keyboard	USA keyboard with no ten keypad.
(2) 85-key keyboard	UK keyboard with no ten keypad.
(3) 87-key keyboard	84-key keyboard + Special keys (S1, S2, S3)
(4) 88-key keyboard	85-key keyboard + Special keys (S1, S2, S3)
(5) 89-key keyboard	Japanese keyboard with no ten key pad.
(6) 92-key keyboard	89-key keyboard + Special keys (S1, S2, S3)
(7) 101-key keyboard	USA AT keyboard
(8) 102-key keyboard	UK AT keyboard
(9) 104-key keyboard	101-key keyboard + Special keys (S1, S2, S3)
(10) 105-key keyboard	102-key keyboard + Special keys (S1, S2, S3)
(11) 106-key keyboard	Japanese AT keyboard
(12) 109-key keyboard	106-key keyboard + Special keys (S1, S2, S3)

2. Keyboard Function Operation

This section explains the operation of the Keyboard function as a device, and as a keyboard. The keyboard is controlled according to these functions.

2.1 Device Operation

(1) Initialization

Sets the used registers, RAM, etc. to the initial state.
MIE is reset and all flags and TRBF are cleared.
MIE is set to the receive condition.

(2) Downstream Timeout

The device (except for Device Kill) constantly detects the downstream timeout condition of the Maple Bus. When there is no state transition on either SDCKA or SDCKB for 1 ms or more, timeout is detected. Detection starts when the start pattern is received(RXB = '1'), and ends when the end pattern is received. (RFB = '1').
When timeout is detected, the following processing steps are carried out.

Processing steps	Device
1	Reset
2	Wait for [Device Request]
3	Normal operation

Fig. 2-1 Timeout processing sequence

(3) Upstream Timeout

When sending data to the host (upstream), an error in MIE or the Maple Bus may result the inability to normally recover from the send condition. In such a case, the host will detect a timeout and carry out a hardware reset, but the device will not accept he reset pattern because it is still in the send state.
Therefore the device is made to calculate the estimated send time from the data size and enter into timeout processing (MIE reset) when that period is exceeded.

2.2 Keyboard Operation

(1) Key scan

All key readout data (key scan codes) must be constantly updated, so that any data request from the host can be processed with quick response. Latest key scan data are therefore kept in memory, ready to be returned to the host immediately when needed. Allkeys are scanned simultaneously, with no priority.

(2) LEDs

The keyboard LEDs can be controlled either by the host or by the Keyboard function. When controlled by the host, the ON condition is maintained until an OFF command is received. When reset is performed, all LEDs are turned OFF.

(3) Optimization conditions

a) Simultaneous key press patterns must be detected.
b) Maximum number of simultaneous key-press readout is six.
c) When seven or more keys are pressed simultaneously, the Keyboard function generates an error.
d) When two or more keys are pressed and a sneak current has occurred, the Keyboard function generates an error.

(4) Restrictions

Key repeat (continuous output of a key scan code when a key is held down) cannot be set at the Keyboard function. When key repeat is required, the repeat rate and initial repeat delay must be set by the host in software.

(5) Default state

All LEDs are out.

3. Device ID

The device ID corresponds to the Maple Bus 1.0 Standard Specifications. The table below shows the memory image on the host.

3.1 Maple Bus 1.0 Device ID Configuration

The configuration uses 16 bytes (128 bit).

bit	7	6	5	4	3	2	1	0
1st Data	FT_{31}	FT_{30}	FT_{29}	FT_{28}	FT_{27}	FT_{26}	FT_{25}	FT_{24}
2nd Data	FT_{23}	FT_{22}	FT_{21}	FT_{20}	FT_{19}	FT_{18}	FT_{17}	FT_{16}
3rd Data	FT_{15}	FT_{14}	FT_{13}	FT_{12}	FT_{11}	FT_{10}	FT_9	FT_8
4th Data	FT_7	FT_6	FT_5	FT_4	FT_3	FT_2	FT_1	FT_0
5th Data	$FD1_{31}$	$FD1_{30}$	$FD1_{29}$	$FD1_{28}$	$FD1_{27}$	$FD1_{26}$	$FD1_{25}$	$FD1_{24}$
6th Data	$FD1_{23}$	$FD1_{22}$	$FD1_{21}$	$FD1_{20}$	$FD1_{19}$	$FD1_{18}$	$FD1_{17}$	$FD1_{16}$
7th Data	$FD1_{15}$	$FD1_{14}$	$FD1_{13}$	$FD1_{12}$	$FD1_{11}$	$FD1_{10}$	$FD1_9$	$FD1_8$
8th Data	$FD1_7$	$FD1_6$	$FD1_5$	$FD1_4$	$FD1_3$	$FD1_2$	$FD1_1$	$FD1_0$
9th Data	$FD2_{31}$	$FD2_{30}$	$FD2_{29}$	$FD2_{28}$	$FD2_{27}$	$FD2_{26}$	$FD2_{25}$	$FD2_{24}$
10th Data	$FD2_{23}$	$FD2_{22}$	$FD2_{21}$	$FD2_{20}$	$FD2_{19}$	$FD2_{18}$	$FD2_{17}$	$FD2_{16}$
11th Data	$FD2_{15}$	$FD2_{14}$	$FD2_{13}$	$FD2_{12}$	$FD2_{11}$	$FD2_{10}$	$FD2_9$	$FD2_8$
12th Data	$FD2_7$	$FD2_6$	$FD2_5$	$FD2_4$	$FD2_3$	$FD2_2$	$FD2_1$	$FD2_0$
13th Data	$FD3_{31}$	$FD3_{30}$	$FD3_{29}$	$FD3_{28}$	$FD3_{27}$	$FD3_{26}$	$FD3_{25}$	$FD3_{24}$
14th Data	$FD3_{23}$	$FD3_{22}$	$FD3_{21}$	$FD3_{20}$	$FD3_{19}$	$FD3_{18}$	$FD3_{17}$	$FD3_{16}$
15th Data	$FD3_{15}$	$FD3_{14}$	$FD3_{13}$	$FD3_{12}$	$FD3_{11}$	$FD3_{10}$	$FD3_9$	$FD3_8$
16th Data	$FD3_7$	$FD3_6$	$FD3_5$	$FD3_4$	$FD3_3$	$FD3_2$	$FD3_1$	$FD3_0$

Fig. 3-1 Device ID

FT : Peripheral function type
FD1 : Function definition block for 1st function
FD2 : Function definition block for 2nd function
FD3 : Function definition block for 3rd function

(1) FT_0 - FT_{31}: Function type
Indicates the function type implemented by the peripheral. There are a total of 32 function types.

(2) FD_{31} - FD_0: Function definition block
These blocks define the various elements that make up a function.

3.2 Function Type

The function type (FT) is indicated in the device ID. FT for the keyboard function is FT_6 = '1'.

bit	7	6	5	4	3	2	1	0
1st Data	FT_{31}	FT_{30}	FT_{29}	FT_{28}	FT_{27}	FT_{26}	FT_{25}	FT_{24}
2nd Data	FT_{23}	FT_{22}	FT_{21}	FT_{20}	FT_{19}	FT_{18}	FT_{17}	FT_{16}
3rd Data	FT_{15}	FT_{14}	FT_{13}	FT_{12}	FT_{11}	FT_{10}	FT_9	FT_8
4th Data	FT_7	1	FT_5	FT_4	FT_3	FT_2	FT_1	FT_0

Fig. 3-2 Keyboard function type

For example, a peripheral device which incorporates only the Keyboard function will have the function type FT = '00-00-00-40h'.
A device incorporating also other functions will have the bit corresponding to the other function types set to '1'.

3.3 Function Definition Block

The function definition block in the device ID is shown below. For the keyboard, it specifies the implemented function configuration.

bit	7	6	5	4	3	2	1	0
1st Data	KL_7	KL_6	KL_5	KL_4	KL_3	KL_2	KL_1	KL_0
2nd Data	KT_7	KT_6	KT_5	KT_4	KT_3	KT_2	KT_1	KT_0
3rd Data	LD_7	LD_6	LD_5	LD_4	LD_3	LD_2	LD_1	LD_0
4th Data	LDON	FD_6	FD_5	FD_4	FD_3	FD_2	FD_1	FD_0

Fig. 3-3 Keyboard function definition block

KL : Keyboard language 1 byte
KT : Keyboard type 1 byte
LD : LED type 1 byte
LDON : LED control 1 bit
FD : Reserved

(1) KL: Keyboard language

Keyboard language	KL
*1	00h
Japan	01h
America (US)	02h
England (UK)	03h
Germany	04h
France	05h
Italy	06h
Spain	07h
Sweden	08h
Switzerland	09h
Holland	0Ah
Portugal	0Bh
Latin America	0Ch
Canadian French	0Dh
Russia	0Eh
China	0Fh
Korea	10h
Reserved	11h～FFh

Fig. 3-4 keyboard language setting

*1: Prohibited setting

Example: The Japanese keyboard is KL = '01h'.

Note: For new language types, KL = '11h - FFh' will be used.

(2) KT: Keyboard type

Keyboard type	KT
*2	00h
89-key	01h
92-key	02h
101-key	03h
102-key	04h
104-key	05h
105-key	06h
106-key	07h
109-key	08h
87-key	09h
88-key	0Ah
Reserved	0Bh～80h
Reserved for unusual types	81h～FFh

Fig. 3-5 Keyboard type setting

*2: Prohibited setting

Example: The 89-key keyboard is KT = '01h'.

Note: The reserved setting is for other keyboards that are available on the market (such as a 128-key keyboard) and for keyboards to be developed in house. For these, KT = '0Bh - 80h' is used. Within this definition, new and unusual keys can be accommodated by sing the reserved setting KT = '81h - FFh'.

(3) LD: LED type

LD	Mapping	Yes	No
0	Num Lock	1	0
1	Caps Lock	1	0
2	Scroll Lock	1	0
3	Reserved	1	0
4	Reserved	1	0
5	Kana	1	0
6	Power	1	0
7	Shift	1	0

Fig. 3-6 LED type setting

(4) LDON: LED control

Determines whether the keyboard LED ON/OFF state is controlled by the host or the Keyboard function.

LED ON/OFF control	LDON
Host	0
Keyboard function	1

Fig. 3-7 LDON setting

When set to be controlled by the Keyboard function, the host disregards the LD setting.

(5) FD: Reserved

Fixed to '0'.

4. Data Format

This section describes the Keyboard function data format.
The notation uses the memory image on the host.

4.1 Write Format

The format for writing Keyboard function data is shown below. When the host sends write data with "Set Condition", the keyboard returns a "Device Reply" for normal end or an error code for abnormal end.

bit	7	6	5	4	3	2	1	0
1st Data	LD_7	LD_6	LD_5	LD_4	LD_3	LD_2	LD_1	LD_0
2nd Data	$W1_7$	$W1_6$	$W1_5$	$W1_4$	$W1_3$	$W1_2$	$W1_1$	$W1_0$
3rd Data	$W2_7$	$W2_6$	$W2_5$	$W2_4$	$W2_3$	$W2_2$	$W2_1$	$W2_0$
4th Data	$W3_7$	$W3_6$	$W3_5$	$W3_4$	$W3_3$	$W3_2$	$W3_1$	$W3_0$

Fig. 4-1 Write format

Write format description

1st : LD LED setting
2nd : W1 Reserved = 00h
3rd : W2 Reserved = 00h
4th : W3 Reserved = 00h

LED setting

LD	Mapping	Lit	Out
0	Num Lock	1	0
1	Caps Lock	1	0
2	Scroll Lock	1	0
3	Reserved	1	0
4	Reserved	1	0
5	Kana	1	0
6	Power	1	0
7	Shift	1	0

Fig. 4-2 LED information bit

When set to ON (lit), the state is maintained until set to OFF (out).

4.2 Read Format

The format for reading data from the Keyboard function is shown below. When the host sends a "Get Condition", the keyboard returns the "Data Transfer".
The data format size is 8 bytes.

bit	7	6	5	4	3	2	1	0
1st Data	M_7	M_6	M_5	M_4	M_3	M_2	M_1	M_0
2nd Data	LD_7	LD_6	LD_5	LD_4	LD_3	LD_2	LD_1	LD_0
3rd Data	$KC1_7$	$KC1_6$	$KC1_5$	$KC1_4$	$KC1_3$	$KC1_2$	$KC1_1$	$KC1_0$
4th Data	$KC2_7$	$KC2_6$	$KC2_5$	$KC2_4$	$KC2_3$	$KC2_2$	$KC2_1$	$KC2_0$
5th Data	$KC3_7$	$KC3_6$	$KC3_5$	$KC3_4$	$KC3_3$	$KC3_2$	$KC3_1$	$KC3_0$
6th Data	$KC4_7$	$KC4_6$	$KC4_5$	$KC4_4$	$KC4_3$	$KC4_2$	$KC4_1$	$KC4_0$
7th Data	$KC5_7$	$KC5_6$	$KC5_5$	$KC5_4$	$KC5_3$	$KC5_2$	$KC5_1$	$KC5_0$
8th Data	$KC6_7$	$KC6_6$	$KC6_5$	$KC6_4$	$KC6_3$	$KC6_2$	$KC6_1$	$KC6_0$

Fig. 4-3 Read format

Read format description

1st : Change key bit
2nd : LED information
3rd : Key scan code array #1
4th : Key scan code array #2
5th : Key scan code array #3
6th : Key scan code array #4
7th : Key scan code array #5
8th : Key scan code array #6

Change key bit

M	Mapping	Down	Up
0	Left Control	1	0
1	Left Shift	1	0
2	Left Alt	1	0
3	Left GUI	1	0
4	Right Control	1	0
5	Right Shift	1	0
6	Right Alt	1	0
7	S2	1	0

Fig. 4-4 Change key bit

Example: [Control] + [Alt]

Normal '00000000' (00h)
Left [Alt] down '00000100' (04h)
Right [Control] down '00010100' (14h)

LED information

LD	Mapping	Lit	Out
0	Num Lock	1	0
1	Caps Lock	1	0
2	Scroll Lock	1	0
3	Reserved	1	0
4	Reserved	1	0
5	Kana	1	0
6	Power	1	0
7	Shift	1	0

Fig. 4-5 LED information bit

Key scan code array

When a key on the keyboard is pressed, the key scan code (see Fig. 4.3) is placed in the array. The array holds up to six key scan codes. When seven or more keys are pressed, a rollover error is generated.

Key status	Change key	Array #1	Array #2	Array #3	Array #4	Array #5	Array #6	Description
Normal	00h	00h	00h	00h	00h	00h	00h	
Right [Alt] down	40h	00h	00h	00h	00h	00h	00h	Changed key only is updated
[A] down	40h	04h	00h	00h	00h	00h	00h	Array is left-aligned
[H] down	40h	04h	0Bh	00h	00h	00h	00h	
[I] down	40h	04h	0Bh	0Ch	00h	00h	00h	
[M] down	40h	04h	0Bh	0Ch	10h	00h	00h	
Left [Shift] down	42h	04h	0Bh	0Ch	10h	00h	00h	Changed key only is updated
[O] down	42h	04h	0Bh	0Ch	10h	12h	00h	
[T] down	42h	04h	0Bh	0Ch	10h	12h	17h	
[Q] down	42h	01h	01h	01h	01h	01h	01h	Seventh key was pressed. Rollover error generated.
[A] up	42h	0Bh	0Ch	10h	12h	17h	00h	Array is left-aligned
[H] + [M] up	42h	0Ch	12h	17h	00h	00h	00h	
Right [Alt] up	02h	0Ch	12h	17h	00h	00h	00h	Changed key only is updated
[I] up	02h	12h	17h	00h	00h	00h	00h	
[O] up + [Y] down	02h	17h	1Ch	00h	00h	00h	00h	Array is aligned with priority to "up"
[T] up	02h	1Ch	00h	00h	00h	00h	00h	
Left [Shift] up	00h	1Ch	00h	00h	00h	00h	00h	Changed key only is updated
[Y] up	00h	00h	00h	00h	00h	00h	00h	No key pressed

Fig. 4-6 Operation example

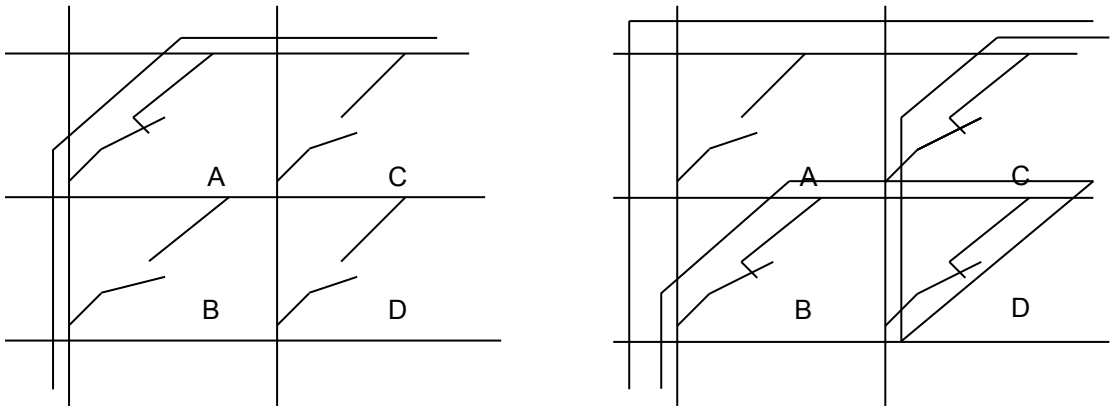
Sneak current phenomenon

Depending on the key position and timing, pressing three or more keys can lead to a condition where the keyboard circuitry cannot respond accurately, producing a false key scan code output. This is called sneak current. When it occurs, the keyboard producs a rollover error.

Key status	Change key	Array #1	Array #2	Array #3	Array #4	Array #5	Array #6	Description
Normal	00h	00h	00h	00h	00h	00h	00h	
Right [Alt] down	40h	00h	00h	00h	00h	00h	00h	Changed key only is updated
[A] down	40h	04h	00h	00h	00h	00h	00h	Array is left-aligned
[Z] down	40h	04h	1Dh	00h	00h	00h	00h	
[P] down	40h	04~~01~~h	1D~~01~~h	01h	01h	01h	01h	Sneak current occurs. Rollover error is generated.

Fig. 4-7 Sneak current condition

The sneak current condition is illustrated below.



[A] key ON [B], [C], [D] key ON

Fig. 4-8 Sneak current principle

When three adjacent keys are pressed as shown in Fig. 4.2.6, the current flow becomes as indicated by the arrows, causing detection of a non-existent keypress (key [A]).

4.3 Key Scan Codes (US, UK JP)

Mapping differs according to language. For details, refer to “Maple Bus 1.0”, Peripheral Hardware Specifications for each keyboard.

Key scan code		Key mapping and usage	Keyboard type							
Decimal	Hex		89 JP	92 JP	101 US	102 UK & others	104 US	105 UK & others	106 JP	109 JP
0	00h	No operation[9]	0	0	0	0	0	0	0	0
1	01h	Rollover error [9]	0	0	0	0	0	0	0	0
2	02h	POST Fail [9]	0	0	0	0	0	0	0	0
3	03h	Undefined error [6]	0	0	0	0	0	0	0	0
4	04h	Keyboard [a], [A][4]	0	0	0	0	0	0	0	0
5	05h	Keyboard [b], [B]	0	0	0	0	0	0	0	0
6	06h	Keyboard [c], [C][4]	0	0	0	0	0	0	0	0
7	07h	Keyboard [d], [D]	0	0	0	0	0	0	0	0
8	08h	Keyboard [e], [E]	0	0	0	0	0	0	0	0
9	09h	Keyboard [f], [F]	0	0	0	0	0	0	0	0
10	0Ah	Keyboard [g], [G]	0	0	0	0	0	0	0	0
11	0Bh	Keyboard [h], [H]	0	0	0	0	0	0	0	0
12	0Ch	Keyboard [i], [I]	0	0	0	0	0	0	0	0
13	0Dh	Keyboard [j], [J]	0	0	0	0	0	0	0	0
14	0Eh	Keyboard [k], [K]	0	0	0	0	0	0	0	0
15	0Fh	Keyboard [l], [L]	0	0	0	0	0	0	0	0
16	10h	Keyboard [m], [M][4]	0	0	0	0	0	0	0	0
17	11h	Keyboard [n], [N]	0	0	0	0	0	0	0	0
18	12h	Keyboard [o], [O][4]	0	0	0	0	0	0	0	0
19	13h	Keyboard [p], [P][4]	0	0	0	0	0	0	0	0
20	14h	Keyboard [q], [Q][4]	0	0	0	0	0	0	0	0
21	15h	Keyboard [r], [R]	0	0	0	0	0	0	0	0
22	16h	Keyboard [s], [S][4]	0	0	0	0	0	0	0	0
23	17h	Keyboard [t], [T]	0	0	0	0	0	0	0	0
24	18h	Keyboard [u], [U]	0	0	0	0	0	0	0	0
25	19h	Keyboard [v], [V]	0	0	0	0	0	0	0	0
26	1Ah	Keyboard [w], [W][4]	0	0	0	0	0	0	0	0
27	1Bh	Keyboard [x], [X][4]	0	0	0	0	0	0	0	0
28	1Ch	Keyboard [y], [Y] [4]	0	0	0	0	0	0	0	0
29	1Dh	Keyboard [z], [Z][4]	0	0	0	0	0	0	0	0
30	1Eh	Keyboard [1], [!][4]	0	0	0	0	0	0	0	0
31	1Fh	Keyboard [2], [@][4]	0	0	0	0	0	0	0	0
32	20h	Keyboard [3], [#][4]	0	0	0	0	0	0	0	0
33	21h	Keyboard [4], [$][4]	0	0	0	0	0	0	0	0
34	22h	Keyboard [5], [%][4]	0	0	0	0	0	0	0	0
35	23h	Keyboard [6], [^][4]	0	0	0	0	0	0	0	0

Fig. 4-9 Key scan codes (1)

Key scan code		Key mapping and usage	Keyboard type							
Decimal	Hex		89 JP	92 JP	101 US	102 UK & others	104 US	105 UK & others	106 JP	109 JP
36	24h	Keyboard [7], [&][4]	0	0	0	0	0	0	0	0
37	25h	Keyboard [8], [*][4]	0	0	0	0	0	0	0	0
38	26h	Keyboard [9], [(][4]	0	0	0	0	0	0	0	0
39	27h	Keyboard [0], [)][4]	0	0	0	0	0	0	0	0
40	28h	Keyboard [Return] (Enter)[5]	0	0	0	0	0	0	0	0
41	29h	Keyboard [Esc]	0	0	0	0	0	0	0	0
42	2Ah	Keyboard [Delete] (Backspace)[13]	0	0	0	0	0	0	0	0
43	2Bh	Keyboard [Tab]	0	0	0	0	0	0	0	0
44	2Ch	Keyboard [Spacebar]	0	0	0	0	0	0	0	0
45	2Dh	Keyboard [-], [_][4]	0	0	0	0	0	0	0	0
46	2Eh	Keyboard [=], [+] [4]	0	0	0	0	0	0	0	0
47	2Fh	Keyboard [[], [{][4]	0	0	0	0	0	0	0	0
48	30h	Keyboard []], [}][4]	0	0	0	0	0	0	0	0
49	31h	Keyboard [\], [|]			0		0			
50	32h	Keyboard Non-US #, [~] [2]	0	0	0	0	0	0	0	0
51	33h	Keyboard [;], [:] [4]	0	0	0	0	0	0	0	0
52	34h	Keyboard['], ["][4]	0	0	0	0	0	0	0	0
53	35h	Keyboard Grave Accent, Tilde[4]	0	0	0	0	0	0	0	0
54	36h	Keyboard [,], [<][4]	0	0	0	0	0	0	0	0
55	37h	Keyboard[.], [>][4]	0	0	0	0	0	0	0	0
56	38h	Keyboard [/], [?][4]	0	0	0	0	0	0	0	0
57	39h	Keyboard [Caps Lock][11]	0	0	0	0	0	0	0	0
58	3Ah	Keyboard [F1]	0	0	0	0	0	0	0	0
59	3Bh	Keyboard [F2]	0	0	0	0	0	0	0	0
60	3Ch	Keyboard [F3]	0	0	0	0	0	0	0	0
61	3Dh	Keyboard [F4]	0	0	0	0	0	0	0	0
62	3Eh	Keyboard [F5]	0	0	0	0	0	0	0	0
63	3Fh	Keyboard [F6]	0	0	0	0	0	0	0	0
64	40h	Keyboard [F7]	0	0	0	0	0	0	0	0
65	41h	Keyboard [F8]	0	0	0	0	0	0	0	0
66	42h	Keyboard [F9]	0	0	0	0	0	0	0	0
67	43h	Keyboard [F10]	0	0	0	0	0	0	0	0
68	44h	Keyboard [F11]	0	0	0	0	0	0	0	0
69	45h	Keyboard [F12]	0	0	0	0	0	0	0	0
70	46h	Keyboard [Print Screen][1]	0	0	0	0	0	0	0	0
71	47h	Keyboard [Scroll Lock][11]	0	0	0	0	0	0	0	0
72	48h	Keyboard [Pause][1]	0	0	0	0	0	0	0	0
73	49h	Keyboard [Insert][1]	0	0	0	0	0	0	0	0
74	4Ah	Keyboard [Home][1]	0	0	0	0	0	0	0	0

Fig. 4-10 Key scan codes (2)

Key scan code		Key mapping and usage	Keyboard type							
Decimal	Hex		89 JP	92 JP	101 US	102 UK & others	104 US	105 UK & others	106 JP	109 JP
75	4Bh	Keyboard [Page Up][1]	0	0	0	0	0	0	0	0
76	4Ch	Keyboard [Delete Forward][1,14]	0	0	0	0	0	0	0	0
77	4Dh	Keyboard [End][1]	0	0	0	0	0	0	0	0
78	4Eh	Keyboard [Page Down][1]	0	0	0	0	0	0	0	0
79	4Fh	Keyboard [→][1]	0	0	0	0	0	0	0	0
80	50h	Keyboard [←][1]	0	0	0	0	0	0	0	0
81	51h	Keyboard [↓][1]	0	0	0	0	0	0	0	0
82	52h	Keyboard [↑][1]	0	0	0	0	0	0	0	0
83	53h	Numeric keypad [Num Lock], [Clear][11]			0	0	0	0	0	0
84	54h	Numeric keypad [/] 1[1]			0	0	0	0	0	0
85	55h	Numeric keypad [*]			0	0	0	0	0	0
86	56h	Numeric keypad [-]			0	0	0	0	0	0
87	57h	Numeric keypad [+]			0	0	0	0	0	0
88	58h	Numeric keypad [Enter][5]			0	0	0	0	0	0
89	59h	Numeric keypad [1], [End]			0	0	0	0	0	0
90	5Ah	Numeric keypad [2], [↓]			0	0	0	0	0	0
91	5Bh	Numeric keypad [3], [Page Down]			0	0	0	0	0	0
92	5Ch	Numeric keypad [4], [←]			0	0	0	0	0	0
93	5Dh	Numeric keypad [5]			0	0	0	0	0	0
94	5Eh	Numeric keypad [6], [→]			0	0	0	0	0	0
95	5Fh	Numeric keypad [7], [Home]			0	0	0	0	0	0
96	60h	Numeric keypad [8], [↑]			0	0	0	0	0	0
97	61h	Numeric keypad [9], [Page Up]			0	0	0	0	0	0
98	62h	Numeric keypad [0], [Insert]			0	0	0	0	0	0
99	63h	Numeric keypad [.], [Delete]			0	0	0	0	0	0
100	64h	Keyboard Non-US #, [|][2]				0		0	0	0
101	65h	Keyboard [S3][10]		0			0	0		0
102	66h	Keyboard [Power][9]								
103	67h	Numeric keypad [=]								
104	68h	Keyboard [F13]								
105	69h	Keyboard [F14]								
106	6Ah	Keyboard [F15]								
107	6Bh	Keyboard [F16]								
108	6Ch	Keyboard [F17]								
109	6Dh	Keyboard [F18]								
110	6Eh	Keyboard [F19]								
111	6Fh	Keyboard [F20]								
112	70h	Keyboard [F21]								
113	71h	Keyboard [F22]								
114	72h	Keyboard [F23]								

Fig. 4-11 Key scan codes (3)

Key scan code		Key mapping and usage	Keyboard type							
Decimal	Hex		89 JP	92 JP	101 US	102 UK & others	104 US	105 UK & others	106 JP	109 JP
115	73h	Keyboard [F24]								
116	74h	Keyboard [Execute]								
117	75h	Keyboard [Help]								
118	76h	Keyboard [Menu]								
119	77h	Keyboard [Select]								
120	78h	Keyboard [Stop]								
121	79h	Keyboard [Again]								
122	7Ah	Keyboard [Undo]								
123	7Bh	Keyboard [Cut]								
124	7Ch	Keyboard [Copy]								
125	7Dh	Keyboard [Paste]								
126	7Eh	Keyboard [Find]								
127	7Fh	Keyboard [Mute]								
128	80h	Keyboard [Volume Up]								
129	81h	Keyboard [Volume Down]								
130	82h	Keyboard [Caps Lock] fixed[12]								
131	83h	Keyboard [Num Lock] fixed [12]								
132	84h	Keyboard [Scroll Lock] fixed [12]								
133	85h	Numeric keypad Comma [27]								
134	86h	Numeric keypad Equal sign [29]								
135	87h	International Keyboard 1[15,28]	0	0					0	0
136	88h	International Keyboard 2[16]	0	0					0	0
137	89h	International Keyboard 3[17]	0	0					0	0
138	8Ah	International Keyboard 4[18]	0	0					0	0
139	8Bh	International Keyboard 5[19]	0	0					0	0
140	8Ch	International Keyboard 6[20]								
141	8Dh	International Keyboard 7[21]								
142	8Eh	International Keyboard 8[22]								
143	8Fh	International Keyboard 9[22]								
144	90h	Keyboard by language 1[25]								
145	91h	Keyboard by language 2[26]								
146	92h	Keyboard by language 3								
147	93h	Keyboard by language 4								
148	94h	Keyboard by language 5								
149	95h	Keyboard by language 6[8]								
150	96h	Keyboard by language 7[8]								
151	97h	Keyboard by language 8[8]								
152	98h	Keyboard by language 9[8]								

Fig. 4-12 Key scan codes (4)

Key scan code		Key mapping and usage	Keyboard type							
Decimal	Hex		89 JP	92 JP	101 US	102 UK & others	104 US	105 UK & others	106 JP	109 JP
153	99h	Keyboard [Alternate Erase] [7]								
154	9Ah	Keyboard [Sys Req/Attention] [1]								
155	9Bh	Keyboard [Cancel]								
156	9Ch	Keyboard [Clear]								
157	9Dh	Keyboard [Prior]								
158	9Eh	Keyboard [Return]								
159	9Fh	Keyboard [Separator]								
160	A0h	Keyboard [Out]								
161	A1h	Keyboard [Oper]								
162	A2h	Keyboard [Clear/Again]								
163	A3h	Keyboard [CrSel/Props]								
164	A4h	Keyboard [ExSel]								
165	A5h	Reserved								
166	A6h	Reserved								
167	A7h	Reserved								
168	A8h	Reserved								
169	A9h	Reserved								
170	AAh	Reserved								
171	ABh	Reserved								
172	ACh	Reserved								
173	ADh	Reserved								
174	AEh	Reserved								
175	AFh	Reserved								
176	B0h	Reserved								
177	B1h	Reserved								
178	B2h	Reserved								
179	B3h	Reserved								
180	B4h	Reserved								
181	B5h	Reserved								
182	B6h	Reserved								
183	B7h	Reserved								
184	B8h	Reserved								
185	B9h	Reserved								
186	BAh	Reserved								
187	BBh	Reserved								
188	BCh	Reserved								
189	BDh	Reserved								
190	BEh	Reserved								

Fig. 4-13 Key scan codes (5)

Key scan code		Key mapping and usage	Keyboard type							
Decimal	Hex		89 JP	92 JP	101 US	102 UK & others	104 US	105 UK & others	106 JP	109 JP
191	BFh	Reserved								
192	C0h	Reserved								
193	C1h	Reserved								
194	C2h	Reserved								
195	C3h	Reserved								
196	C4h	Reserved								
197	C5h	Reserved								
198	C6h	Reserved								
199	C7h	Reserved								
200	C8h	Reserved								
201	C9h	Reserved								
202	CAh	Reserved								
203	CBh	Reserved								
204	CCh	Reserved								
205	CDh	Reserved								
206	CEh	Reserved								
207	CFh	Reserved								
208	D0h	Reserved								
209	D1h	Reserved								
210	D2h	Reserved								
211	D3h	Reserved								
212	D4h	Reserved								
213	D5h	Reserved								
214	D6h	Reserved								
215	D7h	Reserved								
216	D8h	Reserved								
217	D9h	Reserved								
218	DAh	Reserved								
219	DBh	Reserved								
220	DCh	Reserved								
221	DDh	Reserved								
222	DEh	Reserved								
223	DFh	Reserved								
224	E0h	Keyboard Left [Control] *3	0	0	0	0	0	0	0	0
225	E1h	Keyboard Right [Shift] *3	0	0	0	0	0	0	0	0
226	E2h	Keyboard Left [Alt] *3	0	0	0	0	0	0	0	0
227	E3h	Keyboard Left [S1][10,23] *3		0			0	0		0
228	E4h	Keyboard Right [Control] *3	0	0	0	0	0	0	0	0
229	E5h	Keyboard Right [Shift] *3	0	0	0	0	0	0	0	0
230	E6h	Keyboard Right [Alt] *3	0	0	0	0	0	0	0	0
231	E7h	Keyboard Right [S3][10,24] *3		0			0	0		0

Fig. 4-14 Key scan codes (6)

Key scan code		Key mapping and usage	Keyboard type							
Decimal	Hex		89 JP	92 JP	101 US	102 UK & others	104 US	105 UK & others	106 JP	109 JP
232	E8h	Reserved								
233	E9h	Reserved								
234	EAh	Reserved								
235	EBh	Reserved								
236	ECh	Reserved								
237	EDh	Reserved								
238	EEh	Reserved								
239	EFh	Reserved								
240	F0h	Reserved								
241	F1h	Reserved								
242	F2h	Reserved								
243	F3h	Reserved								
244	F4h	Reserved								
245	F5h	Reserved								
246	F6h	Reserved								
247	F7h	Reserved								
248	F8h	Reserved								
249	F9h	Reserved								
250	FAh	Reserved								
251	FBh	Reserved								
252	FCh	Reserved								
253	FDh	Reserved								
254	FEh	Reserved								
255	FFh	Reserved								

Fig. 4-15 Key scan codes (7)

#3 These keys are treated as 2nd data henka-key bits in the read format. The key code is not returned for the 3rd to 8th data of the read format.

Notes:

1. Key scan codes do not change also when [Control], [Alt], [Shift], or [Num Lock] is pressed. Pressing these keys does not generate an additional key scan code.

2. Mapping differs depending on the keyboard language. Normally, the following keys are mapped.
US:\| Belg:µ` £ FrCa:<}> Dan:'* Dutch:<> Fren:*µ Ger:#' Ital:ù§ LatAm:}`] Nor:,* Span:}Ç Swed:,* Swiss:$£ UK:#~

3. Mapping differs according to keyboard language. Normally, the following keys are mapped.
Belg:<\> Frca: 《º》 Dan:<\> Dutch:]|[Fren:<> Ger:<|> Ital:<> LatAm:<> Nor:<> Span:<> Swed:<|> Swiss:<\> UK:\| Brazil:\|

4. Mapping for other language performed at the host.

5. The keyboard Enter key and the numeric keypad Enter key generate a different key scan code.

6. Ordinarily located near the left [Shift] key on 102 Key Keyboards.

7. Example, the Erase-Eaze™ key

8. Reserved for language-specific functions, such as FEP, IME, etc.

9. Reserved for standard keyboard state or keyboard error. Available as keyboard array, but does not correspond to physical key being pressed.

10. Special key 3. The key top may change depending on keyboard specifications.

11. Functions as non-fixed key.

12. Functions as fixed key. Use as a toggle button to support older equipment is possible, but use as a non-fixed key is recommended.

13. Moves cursor one position back and erases characters one at a time.

14. Erases characters without moving cursor position.

15-20. Used on Japanese keyboards.

21.1 Changes between the 1-byte and 2-byte modes.

22.Undefined. Reserved for FEPs of other languages.

23. Special key 1. Key top may change depending on keyboard specifications.

24. Special key 2. Key top may change depending on keyboard specifications.

25. Key for switching between Hangul and English. Used on the Korean keyboard.

26. Hanja change key. Used on the Korean keyboard.

27. Used as the period key [.] on the Brazilian keyboard. This may be obvious, but please note the precise key mapping for each language under the system software.

28.

29. Used with the AS/400 keyboard.

	AT-109	NEC PC-98
15.	_ | \ ろ	— ろ
16.	カタカナ ひら がな	かな
17.	| 「 ¥ —	| ¥ —
18.	前候補 変換	XFER
19.	無変換	NFER
20.		,

5. Supported Commands

Within the framework of commands defined by the Maple Bus 1.0 Standard Specifications, the commands supported by the Keyboard function are listed below. When the Keyboard function receives a command other than one of those listed, it returns an error.
This section also describes the operation of the Keyboard function in response to the command.

5.1 Control Commands

5.1.1 Device Request

Issue privilege	:Host
Command code	:01h
Data size	:00h
Data area	:None
Expected return value	:[Device Status]
Description	:This command requests the [Device Status] information from the Keyboard function specified as transfer target AP. The command is also used for connection check of the various ports. After initialization, the Keyboard function does not respond to any command before this command is received.
Operation sequence	:(1) Receive command. (2) Generate transfer target AP and return to host.

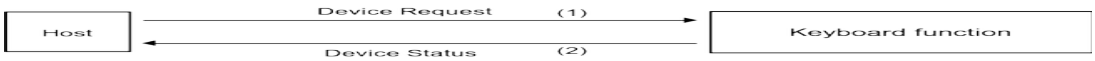


Fig. 5-1 Device Request

5.1.2 All Status Request

Issue privilege	:Host
Command code	:02h
Data size	:00h
Data area	:None
Expected return value	: [Device All Status]
Description	:This command requests the entire device status information (Fixed Device Status and Free Device Status) from the Keyboard function specified as transfer target AP.

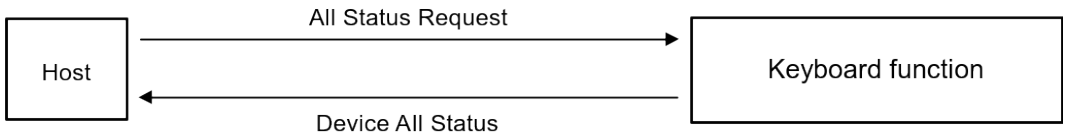


Fig. 5-2 All Status Request

5.1.3 Device Reset

Issue privilege	:Host
Command code	:03h
Data size	:00h
Data area	:None
Expected return value	: [Device Reply]
Description	:This command initializes the Keyboard function specified as transfer target AP.
Operation sequence	:(1) Return [Device Reply]. (2) Initialize.

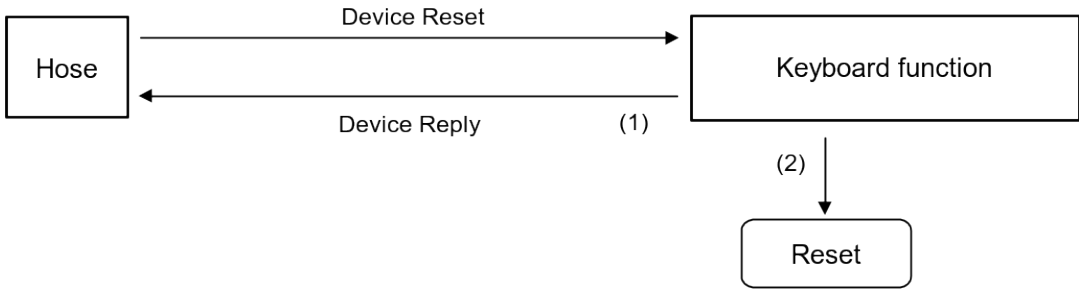


Fig. 5-3 Device Reset

5.1.4 Device Kill

Issue privilege	:Host
Command code	:04h
Data size	:00h
Data area	:None
Expected return value	: [Device Reply]
Description	:This command disables operation of the Keyboard function specified as transfer target AP. The function goes into standby current consumption mode and does not accept any further commands. To reactivate the function, a hardware rest or power off/on cycle is required.
Operation sequence	:(1) Return [Device Reply]. (2) Stop operation.

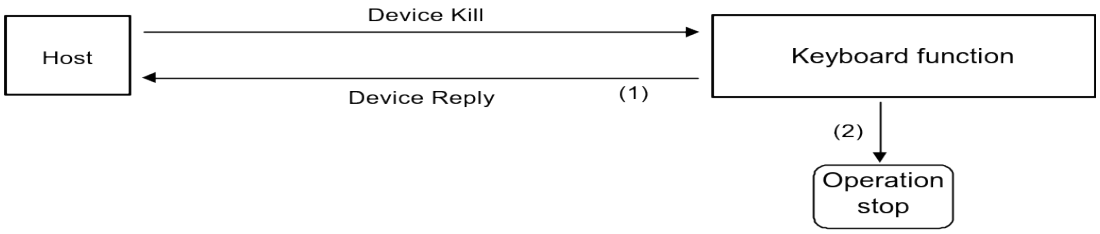


Fig. 5-4 Device Kill

5.1.5 Device Status (Note: In Maple Bus Rev. 0.81 data sizes will be changed.)

Issue privilege	:Peripheral device	
Command code	:05h	
Data size	:1Ch (28)	
Data area	:Device ID:	16 bytes
	Country code:	1 byte
	Product name:	30 bytes
	License:	60 bytes
	Standby current consumption:	2 bytes
	Maximum current consumption:	2 bytes
Description	:Returns Fixed Device Status data in response to [Device Request] from host.	

5.1.6 Device All Status (Note: In Maple Bus Rev. 0.81 data sizes will be changed.)

Issue privilege	:Peripheral device	
Command code	:06h	
Data size	:1Ch +(n/4)	
Data area	:Fixed Device Status:	112 bytes
	Device ID:	16 bytes
	Country code:	1 byte
	Product name:	30 bytes
	License:	60 bytes
	Standby current consumption:	2 bytes
	Maximum current consumption:	2 bytes
	Free Device Status: (n x 4) bytes	
Description	:Returns Fixed Device Status and Free Device Status data in response to [All Status Request] from host.	

5.1.7 Device Reply

Issue privilege	:Peripheral device
Command code	:07h
Data size	:00h
Data area	:None
Description	:Used as reply.

5.1.8 Get Condition

Issue privilege :Host
Command code :09h
Data size :01h
Data area :Function type :4 bytes
Specifies keyboard (00-00-00-40h)
Expected return value :[Data Transfer]
Description :Requests physical status of Keyboard function (button, key, lever status). Used when reading keyboard data.

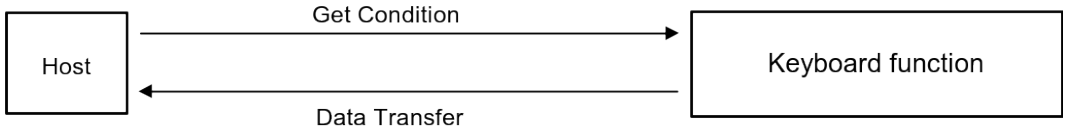


Fig. 5-5 Get Condition

Data Address	Data	Setting example	Setting example description
+0000	Command code	09h	Specifies [Get Condition]
+0001	Transfer target AP	20h	Specifies port A device
+0002	Transfer source AP	00h	Send from port A
+0003	Data size	01h	Data size is 4 bytes
+0004	Function type	00h	Function type is specified as "keyboard".
+0005		00h	
+0006		00h	
+0007		40h	

Fig. 5-6 Host [Get Condition] send data (memory image)

5.1.9 Data Transfer

Issue privilege :Keyboard function
Command code :08h
Data size :03h
Data area :Function type :4 bytes
Specifies keyboard (00-00-00-40h)
Read format :8 bytes
Expected return value :None
Description :Used when a [Get Condition] command is received from the host and the keyboard returns read format information.

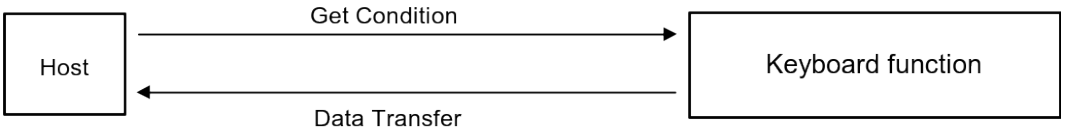


Fig. 5-7 Data Transfer

Data Address	Data	Setting example	Setting example description
+0000	Command code	08h	Specifies [Data Transfer]
+0001	Transfer target AP	00h	Specifies port A
+0002	Transfer source AP	20h	No expansion device
+0003	Data size	03h	Data size is 12 bytes
+0004	Function type	00h	Function type is specified
+0005		00h	as "keyboard".
+0006		00h	
+0007		40h	
+0008	Read format	00h	
+0009		00h	
+000A		00h	
+000B		00h	
+000C		00h	
+000D		00h	
+000E		00h	
+000F		00h	

Fig. 5-8 Host [Data Transfer] send data (memory image)

5.1.10 Set Condition

Issue privilege	:Host	
Command code	:0Eh	
Data size	:02h	
Data area	:Function type	:4 bytes
		Specifies keyboard (00-00-00-40h)
	Write format:	:4 bytes
Expected return value	:[Device Reply]	
Description	:Specified according to the physical configuration of the Keyboard function. Controls the ON/OFF status of LEDs on the keyboard.	

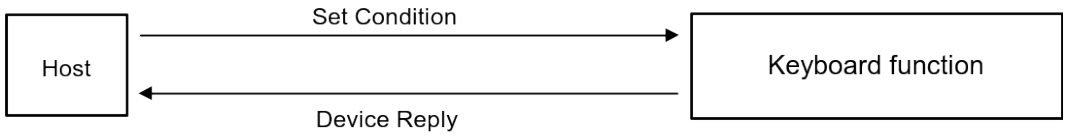


Fig. 5-9 Set Condition

Data Address	Data	Setting example	Setting example description
+0000	Command code	0Eh	Specifies [Set Condition]
+0001	Transfer target AP	00h	Specifies port A
+0002	Transfer source AP	20h	No expansion device
+0003	Data size	02h	Data size is 8 bytes
+0004	Function type	00h	Function type is specified
+0005		00h	as "keyboard".
+0006		00h	
+0007		40h	
+0008	Write format	00h	LED setting
+0009		00h	
+000A		00h	
+000B		00h	

Fig. 5-10 Host [Set Condition] send data (memory image)

5.2 Error Commands

The Keyboard function supports the following three error commands.

5.2.1 Function Type Unknown

Issue privilege	:Peripheral device
Command code	:FEh
Data size	:00h
Data area	:None
Description	:Returned when a function type other than "keyboard" is specified in received data.
Possible causes	:(1) Wrong function type specification (2) Wrong data content (3) Corrupted Device ID (4) Communication error has corrupted data
Remedy	:(1) Specify correct function type (2) Specify data correctly (3) Use [Device Request] once more to obtain Device ID (4) Resend (up to 3 times, afterwards same processing as for timeout)

5.2.2 Command Unknown

Issue privilege	:Peripheral device
Command code	:FDh
Data size	:00h
Data area	:None
Description	:Returned when a command not supported by the Keyboard function was received.
Possible causes	:(1) Wrong command specification (2) Wrong data content (3) Corrupted Device ID (4) Communication error has corrupted data
Remedy	:(1) Specify correct command (2) Specify data correctly (3) Use [Device Request] once more to obtain Device ID (4) Resend (up to 3 times, afterwards same processing as for timeout)

5.2.3 Transmit Again

Issue privilege	:Host, peripheral device
Command code	:FCh
Data size	:00h
Data area	:None
Description	:Returned when an error was detected in received data and data should be sent once more. However, because the keyboard constantly updates its data, the resent keyboard read data may not be the same as the previously sent data.
Possible causes	:(1) Parity error has occurred (2) Data overflow (3) Communication error has corrupted data (4) Other cause
Remedy	:Resend (up to 3 times, afterwards same processing as for timeout)

6. Keyboard Function Information

This section describes device-specific information (device status). The device status must be stored in such a way that device status data cannot be changed or erased.

6.1 Type

Fixed Device Status

This refers to 112 bytes of device status information data with a fixed format, comprising required information. Correct connection and operation are only assured if all items are properly recorded.

Free Device Status

This refers to a maximum of 908~~912~~ bytes of device-specific status information that can be allocated freely.

6.2 Fixed Device Status

The Fixed Device Status area must include all the items listed below.

(1) Device ID

Size :16 bytes

Description :Specifies the function definition. For details, refer to chapter 3.

(2) Country specification

Size :1 byte

Description :Specifies the destination (marketing) region.

bit	7	6	5	4	3	2	1	0
data	DES_7	DES_6	DES_5	DES_4	DES_3	DES_2	DES_1	DES_0

Fig. 6-1 Country specification bit

Area	Setting bit
North America	DES_0='1'
Japan	DES_1='1'
Asia except Japan	DES_2='1'
Europe	DES_3='1'
Reserved area 1	DES_4='1'
Reserved area 2	DES_5='1'
Reserved area 3	DES_6='1'
Reserved area 4	DES_7='1'

Fig. 6-2 Country specification bit

Worldwide: DES = '11111111' = FFh

The country specification and the keyboard language are not necessarily identical.

(3) Connection method

Size :1 byte

Description :For device:

Indicates orientation of expansion socket for connection of expansion device.

For expansion device:

Indicates orientation of connector for connection to device.

bit	7	6	5	4	3	2	1	0
data	$SD4_1$	$SD4_0$	$SD3_1$	$SD3_0$	$SD2_1$	$SD2_0$	$SD1_1$	$SD1_0$

Fig. 6-3 Connection method bit configuration

- Device

Specifies expansion socket orientation.

Non-installed sockets are indicated by '00'.

Orientation	SDx_1	SDx_0
Top	0	0
Bottom	0	1
Left	1	0
Right	1	1

Fig. 6-4 Socket orientation setting

X = 1 - 4: socket number

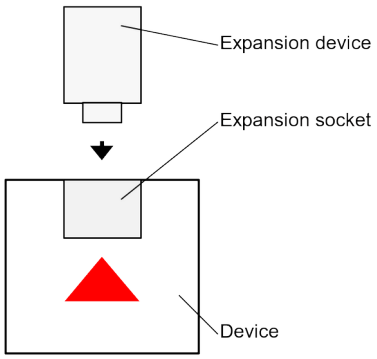


Fig. 6-5 Socket orientation (top)

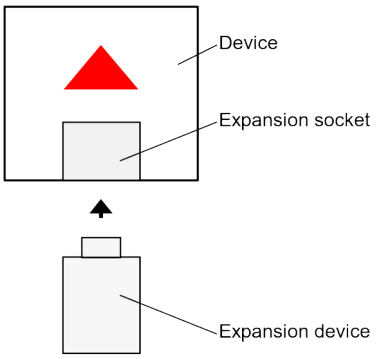


Fig. 6-6 Socket orientation (bottom)

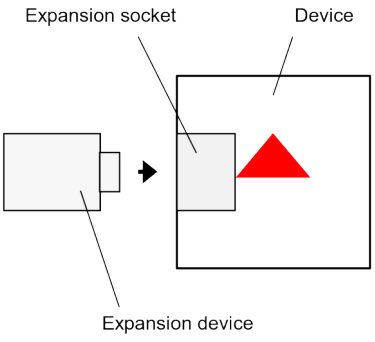


Fig. 6-7 Socket orientation (left)

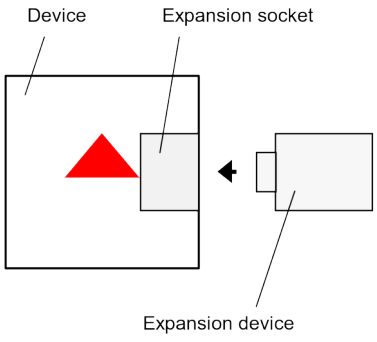


Fig. 6-8 Socket orientation (right)

- Expansion device

Specifies connector orientation.

Bit	7	6	5	4	3	2	1	0
Top facing	0	0	0	0	0	0	0	0
Bottom facing	0	0	0	0	0	0	1	0
Left facing	0	0	0	0	0	1	0	0
Right facing	0	0	0	0	1	0	0	0

Fig. 6-9 Connector orientation setting

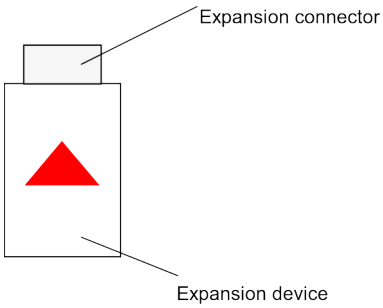


Fig. 6-10 Connector orientation (top)

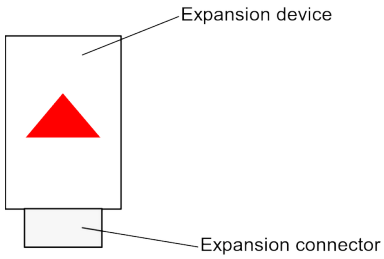


Fig. 6-11 Connector orientation (bottom)

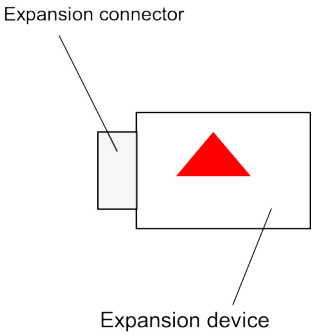


Fig. 6-12 Connector orientation (left)

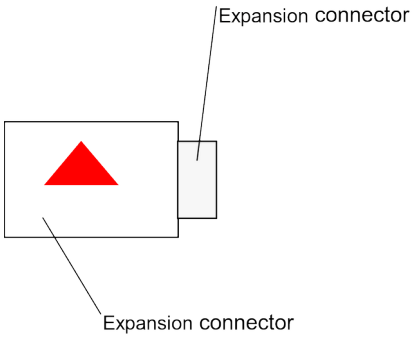


Fig. 6-13 Connector orientation (right)

(4) Model name

Size :30 bytes

Description :ASCII string describing the model
Remaining slots to be padded with spaces (20h).
The product name must be registered in advance.

(5) License

Size :60 bytes

Description :ASCII string. Normally "Produced By or Under License From SEGA ENTERPRISES,LTD."
Remaining slots to be padded with spaces (20h).

(6) Standby current consumption

Size :2 bytes

Description :Indicates the current consumption of the unit in paused condition (During minimum power consumption), in 0.1 mA units (hexadecimal notation). For example, 10.5 mA would be 00- 69h.

(7) Maximum current consumption

Size :2 bytes

Description :Indicates the maximum current consumption of the unit, in 0.1 mA units (hexadecimal notation). For example, 127.9 mA would be 04-FFh.

6.3 Free Device Status

The Free Device Status area can include information about developers, designers, and programmers or any other information. The host can obtain this information by issuing the "All Device Request" command. If it is to be used by an application, the data ordering sequence must be taken into consideration.

Unless there are special reasons for not doing so, the following 40 bytes of information should first be entered.

Versions (ROM, logic, etc.) :13 bytes E.g., “Version 1.000”
Recommended: The version orthography as shown in the example, three digits following the decimal point, and the first release is 1.000.

Release date :10 bytes E.g., “1998/05/11”
Recommended: Year in four digits, month in two digits, and day in two digits.

IC parts number :14 bytes E.g., “315-6125-AB ”
Recommended: Enter space (20h) in the empty area.

Each item should be separated by “,”.

Example: “Version 1.000,1998/05/11,315-6125-AB ,”

7. Usable Characters

The characters that can be used for Fixed Device Status and Device Status are described here.

The characters indicated in the following ASCII code diagram can be used.
Use of the items marked (P.) (= Prohibited) in the diagram is prohibited becomes the control code and characters differ depending on the country.
Use of characters beyond '80h' of the ASCII code is also prohibited.

Character code		Lower 4 bits															
		0h	1h	2h	3h	4h	5h	6h	7h	8h	9h	Ah	Bh	Ch	Dh	Eh	Fh
Upper 4 bits	0h	(P.)	(P.)	(P.)	(P.)	(P.)	(P.)	(P.)	(P.)	(P.)	(P.)	(P.)	(P.)	(P.)	(P.)	(P.)	(P.)
	1h	(P.)	(P.)	(P.)	(P.)	(P.)	(P.)	(P.)	(P.)	(P.)	(P.)	(P.)	(P.)	(P.)	(P.)	(P.)	(P.)
	2h	Space	!	"	(P.)	(P.)	%	&	'	(	)	*	+	,	-	.	/
	3h	0	1	2	3	4	5	6	7	8	9	:	;	<	=	>	?
	4h	(P.)	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
	5h	P	Q	R	S	T	U	V	W	X	Y	Z	(P.)	(P.)	(P.)	(P.)	_
	6h	(P.)	A	b	c	d	e	f	g	h	i	j	k	l	m	n	o
	7h	p	Q	r	s	t	u	v	w	x	Y	z	(P.)	(P.)	(P.)	(P.)	(P.)

Fig. 7-1 ASCII code table

Example: A is '41h' and g is '67h'.

8. Remarks

Contents subject to major or minor changes until release of final (distribution) version (Rev. 1.0).