# “Maple Bus 1.0”
# Function Type Specifications

# $FT_0$ : Controller Function

Revision 0.90

Produced by:
CS Hardware DIV.2
SEGA Enterprises Ltd.

Revisions:

| | | |
|---|---|---|
| 0.80 | January 20 1998 | First distribution |
| 0.81 | January 30 1998 | When the standard controller buttons were altered, it was ascertained that the existing device ID (function definition block) could not be used for definitions. Accordingly, the function definition block was changed. |
| 0.90 | May 20 1998 | Fixed the LM-Bus reset undefined duration.<br>Described buttons and key data not to be used.<br>Added connection direction information to the Free Device Status.<br>Other corrections. |

* For items affected by the latest revisions and previous revisions, added items are set off by , and deleted items are set off by .

CONTENTS

# 1 CONTROLLER FUNCTION CONDITIONS

## 1.1 Controller function definitions

The controller function should be able to control all application software and indicate the input form of the man-machine interface.

(1) There are no restrictions on external view, but there should be compatibility with control of any product.

(2) It is an absolute condition that it is equipped with the minimum function constituent elements.

(3) It must conform to the “Maple Bus 1.0” Standard Specifications.

## 1.2 Function elements

Controller function elements are as follows:

- Digital direction keys A : Ra,La,Da,Ua
- Digital direction keys B : Rb,Lb,Db,Ub
- Digital buttons : A,B,C,D,X,Y,Z,Start
- Analog keys : A3,A4,A5,A6
- Analog levers : A1,A2

It is an absolute condition that the controller is equipped with the following elements.

- Digital direction keys A : Ra,La,Da,Ua
- Digital buttons : A,B,Start

If these elements are missing, it cannot be regarded as a controller.

Accordingly, the controller can have the following features.

a) Have two systems for 4-direction (8-direction) operation.

b) Buttons can be freely organized within the range of seven buttons.

c) The analog system control has up to six axes.

d) Two analog X-Y axes systems.

## 1.3 Detailed description of constituent elements

Each of the controller function elements is describe in detail in the following.

(1) Digital direction keys A : Ra,La,Da,Ua

These are 2-value press/release (= ON/OFF) digital type keys (buttons).

Ra and La, Da and Ua form counterparts. The straight line (X-axis) on which Ra and La are placed at the respective endpoints intersects with the other straight line (Y-axis) on which the Da and Ua are placed at the respective endpoints. The keys (buttons) are arranged on the X-Y surface composed by these straight lines. The way the keys (buttons) are arranged and the directions of movement are as follows: Ra is on the right side, right direction, La is on the left side, left direction, Da is at the bottom, downward direction and toward the viewer, Ua is at the top, upward direction and away from the viewer.

The values are press= '0', release= '1'.

The controller must not generate the key data of more than 3 keys (buttons) at the same time.

These direction keys are primarily for left-hand operation.

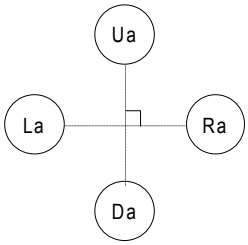


Fig. 1.1 Button layout of digital cross keys A

(2) Digital direction keys B : Rb,Lb,Db,Ub

These are 2-value press/release (= ON/OFF) digital type keys (buttons).

Rb and Lb, Db and Ub form counterparts. The straight line (X-axis) on which Rb and Lb are placed at the respective endpoints intersects with the other straight line (Y-axis) on which the Db and Ub are placed at the respective endpoints. The keys (buttons) are arranged on the X-Y surface composed by these straight lines.

The way the keys (buttons) are arranged and the directions of movement are as follows: Rb is on the right side, right direction, Lb is on the left side, left direction, Db is at the botton, downward direction and toward the viewer, Ub is at the top, upward direction and away from the viewer.

These direction keys are primarily for right-hand operation.

The values are: press = '0', release = '1'.

The controller must not generate the key data of more than 3 keys (buttons) at the same time.

* The principle of use should be so that the digital direction keys A have priority in use while the digital direction keys B act as additional direction keys.

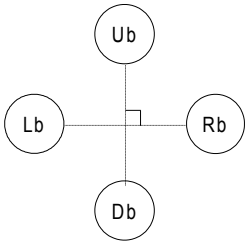


Fig. 1.2 Button layout of digital cross keys B

(3) Digital button : A,B,C,D,X,Y,Z,Start

These are 2-value press/release (= ON/OFF) digital type keys (buttons).

The button arrangement is optional.

The controller must be able to detect whether multiple keys (buttons) are ON at the same time.

The values are: press = '0', release = '1'.

(4) Analog keys : A3,A4,A5,A6

These are analog type keys where the value detected in accordance with the distance the key is moved from its initial position changes linearly.

The value at the key's initial position is 80h, and the value changes in 01h units from the minimum value 00h to the maximum value FFh.

In relation to the key position, the direction in which the value decreases is the minus direction, and the

direction in which the value increases is the plus direction.

The key should be able to move in these two directions.

When the load applied to move the key is released, the centering of the key should be performed automatically so that the key returns to the initial position.

In general, A3 and A4 indicate the Xa-axis and Ya-axis, and A5 and A6 indicate the Xb-axis and Yb-axis. The (Xa,Ya) system is operated with the left hand while the (Xb,Yb) system is operated with the right hand.

Use of only one axis at a time should also be possible.

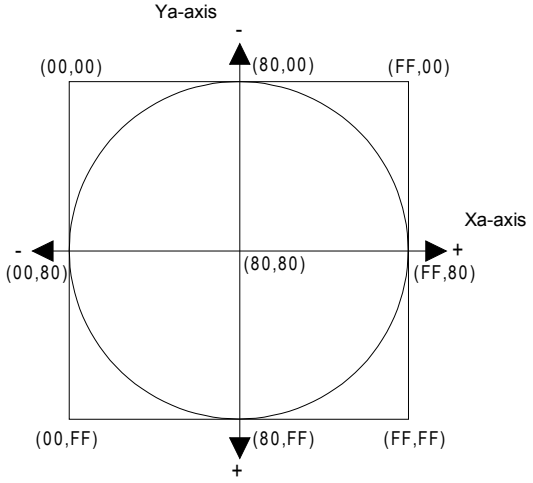


Fig. 1.3 Data range of analog key

(5) Analog levers : A1,A2

These are analog type levers where the value detected in accordance with the distance the lever is moved from its initial position changes linearly.

The value at the lever's initial position is 00h, and the value changes in 01h units from the minimum value 00h to the maximum value FFh.

In relation to the lever position, the direction in which the value decreases is the minus direction, and the direction in which the value increases is the plus direction.

The lever should be able to move in these two directions but from the initial position it should only be able to move in one direction.

When the load applied to move the lever is released, the lever should automatically return to its initial position.

In general, A1 indicates the R-axis and A2 indicates the L-axis. The R-axis is operated with the right hand and the L-axis with the left hand.

Use of only one axis at a time should also be possible.

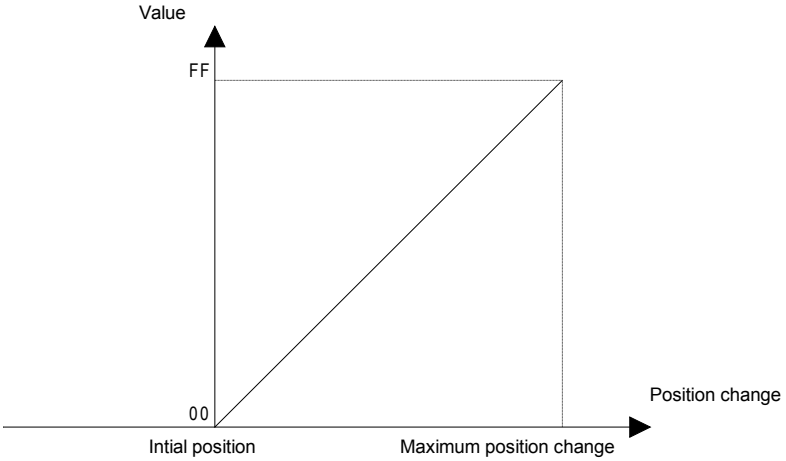


Fig. 1.4 Data range of analog lever

## 1.4 Recommended operation systems

To maintain operation system compatibility between products with controller functions, recommended operation systems are described in the following.

There is a chance that the configuration of the digital buttons may change.

| | |
|---|---|
| (1) Controller with minimum constituents (Essential operation system) | Digital direction keys A, digital buttons (A,B,Start) |
| (2) Controller with standard constituents | Digital direction keys A, digital buttons (A,B,X,Y,Strat)<br>Analog X-Y axis (A3, A4), analog levers (A1, A2) |
| (3) W cross-key controller | Digital direction keys A, digital direction keys B<br>Digital button (A, B, X, Y Start) |
| (4) W analog controller | Analog X-Y axis A (A3, A4), analog X-Y axis B (A5, A6)<br>Analog levers (A1, A2), digital buttons (A, B, C, Start) |
| (5) JOY-STICK | Digital direction keys A, digital buttons (A, B, C, D, X, Y, Z, Start) |
| (6) Analog JOY-STICK | Digital direction keys A, digital buttons (A, B, C, X, Y, Z, Start)<br>Analog X-Y axis (A3, A4), analog levers (A1, A2) |
| (7) W Analog JOY-STICK | Digital direction keys A, digital direction keys B<br>Analog X-Y axis A (A3, A4), analog X-Y axis B (A5, A6)<br>Digital buttons (A, B, C, X, Y, Z, Start) |
| (8) Handle controller | Analog X axis A (A3), analog levers (A1, A2)<br>Digital direction key A, digital buttons (A, B, Strat) |
| (9) W JOY-STICK | Digital direction keys A, Digital direction keys B<br>Digital buttons (A, B, C, X, Y, Z, Start) |

# 2 CONTROLLER FUNCTION OPERATION

The following describes the controller function operation in the case of operation as device and operation as controller.

The controller should be manufactured in accordance with the operation of these functions.

## 2.1 Operation as device

(1) Initialization

Initializes the registers, RAM, etc. to be used.

Resets MIE and clears all flags and TRBF.

MIE is set to the receive condition.

The LM-Bus is completely disconnected and is set so that data does not flow to expansion devices.

The ID0, ID1 of the LM-Bus are set as the reverse of the Bus No.'s setting values.

(2) LM-Bus control

The LM-Bus remains as it is until it receives a [Device Request].

When it receives the first [Device Request] after initialization, it reverses the ID0, ID1 of the LM-Bus and returns them to the set values.

At the same, it also requests reset of expansion devices.

If it detects a '1' on ID2 for each LM-Bus, the expansion device is connected, so it connects that LM-Bus to the Maple-Bus.

It reads the current status of the connected LM-Bus and reflects the [Device Status], which is the reply to the [Device Request] in the AP.

If a '0' is detected on the ID2 of each LM-Bus, the expansion device is not physically connected, so that LM-Bus is disconnected from the Maple-Bus.

This check of ID2 is conducted every time a command is received from the host.

(3) Down Stream Time Out

The device is always detecting the Maple-Bus' Down Stream Time Out (except during [Device Kill]).

A Time Out is when there is no change for more than 1 ms in either the SDCKA or SDCKB.

Detection begins when the start pattern is received (RXB = '1') and continues until the end when the end pattern is received (RFB = '1').

In case of Time Out, the following processing starts.

| Process-ing order | Device | Expansion Device |
|---|---|---|
| 1 | Resets | Normal operation |
| 2 | There is a chance that ID0,1 will become indefinite during resetting. (the period is within 5ms ~~300us~~ (provisional)) | Normal operation<br>ID0, ID1 are not reversed, so they don't initialize. |
| 3 | Reversed from the ID0, ID1 from before initialization. (Initialization) | ID0, ID1 are reversed so they are initialized. (LM-Bus reset) |
| 4 | Waits for [Device Request]. | Resets (Initializes.) |
| 5 | Normal operation | Waits for [Device Request]. |
| 6 | Normal operation | Normal operation |

Fig. 2.1 Time Out processing sequence

(4) Up Stream Time Out

If an abnormal condition occurs in the MIE or the Maple-Bus when sending from a device to the host

(Up Stream) it may not be able to get out of the transmission condition.

In this case, the host will detect a Time Out and perform a hard reset. However, because the device is transmitting, it will not receive the reset pattern.

Therefore, the expected time of transmission completion is calculated by the device which sent the data and if the transmission condition continues for longer than that time it resets the MIE (Time Out processing).

## 2.2 Operation as controller

(1) Key scan

It is a requirement that the data (key data) of the digital keys (buttons) and analog keys are always being updated.

There should always be a good response to data requests from the host.

Accordingly, key scan should always be performed and held key data should always be retained in the newest condition.

When there is a request from the host, the retained data should be in the condition to be sent.

Since there is no order of priority for key scan, all the keys (buttons) should be concurrently readable.

(2) A/D conversion

In a controller using analog data, A/D conversion should be performed to convert the analog quantities to digital quantities.

The A/D conversion accurracy is not a question but there should be 8 bits per axis for the key data.

The maximum total time for A/D conversion of all analog keys should be less than 1 msec which is the Time Out time .

(3) Optimization, conditions

e) For the cross keys, no more than 3 keys (buttons) must be ON at the same time (key data must not be generated).

f) The cross keys U and D, R and L must not be On at the same time (key data must not be generated).

g) The simultaneous ON statuses of multiple digital buttons must detectable.

h) When two or more keys (buttons) are simultaneously pressed, keys (buttons) that are not pressed must not come ON (key data must not be generated).

i) When the analog key is at the center position (initial position), the key data value must not fluctuate while the key is not touched.

j) When the analog lever is at the initial position, the key data value must not fluctuate while the lever is not touched.

k) When the analog key or analog lever is at the maximum changed positions, the key data value must be the maximum value or the minimum value.

l) No matter where the analog key is located inside the operation range, the key must return to the center position when the key is released.

(4) Initialization

Indicates the key data value after power is supplied (reset processing).

Digital direction keys and digital buttons have absolute values that will depend on the key (button) condition at that time.

Analog keys and analog levers will be as follows depending on the device.

1) The key (lever) position at that time becomes the initial position.

2) For keys and levers with decided absolute initial positions, the key (lever) position at that time becomes the key data.

3) Undefined until the key (lever) comes to the decided position.

# 3 Device ID

In accordance with the device ID definition in the “Maple Bus 1.0” Standard Specifications.

The notation is that of the host’s memory image.

## 3.1 Configuration of the “Maple Bus 1.0” device ID

The device ID consists of 16 bytes (128 bits).

| 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 Configuration of device ID

FT : Designates type of function that the peripheral is equipped with.

FD1 : Designates the function definition block of the first function.

FD2 : Designates the function definition block of the second function.

FD3 : Designates the function definition block of the third function.

(1) $FT_0$-$FT_{31}$ : Function type

Designates the function that the peripheral is equipped with.

There are 32 function types altogether.

(2) $FD1_{31}$-$FD1_0$ : Function definition block

This is for the block defining the individual elements making up the function.

## 3.2 Function types

The function types (FT) within the device ID are as follows.

The controller function is $FT_0$='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 | $FT_7$ | $FT_6$ | $FT_5$ | $FT_4$ | $FT_3$ | $FT_2$ | $FT_1$ | 1 |

Fig. 3.2 Controller function types

## 3.3 Function definition block

Indicates the function definition block (FD) within the device ID.

The controller definition block specifies the constituent elements of the function to be used.

The definition method sets the division of the data format's read format.

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| 1st Data | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 2nd Data | 0 | 0 | $RB_{21}$ | $RB_{20}$ | $RB_{19}$ | $RB_{18}$ | $RB_{17}$ | $RB_{16}$ |
| 3rd Data | $RB_{15}$ | $RB_{14}$ | $RB_{13}$ | $RB_{12}$ | $RB_{11}$ | $RB_{10}$ | $RB_9$ | $RB_8$ |
| 4th Data | $RB_7$ | $RB_6$ | $RB_5$ | $RB_4$ | $RB_3$ | $RB_2$ | $RB_1$ | $RB_0$ |

Fig. 3.3 Controller function definition block

RB: Read format division block

'0' indicates fixed 0.

Read format blocks are divided as shown in the figure below.

Details on the read format are explained in the next chapter.

In correspondence with Fig. 3-3, set the bits to '1' for the blocks to be used, and '0' for the blocks not used.

Unused blocks are ignored by the Function Libraries.

| bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| 1st Data | $RB_7$ | $RB_6$ | $RB_5$ | $RB_4$ | $RB_3$ | $RB_2$ | $RB_1$ | $RB_0$ |
| 2nd Data | $RB_{15}$ | $RB_{14}$ | $RB_{13}$ | $RB_{12}$ | $RB_{11}$ | $RB_{10}$ | $RB_9$ | $RB_8$ |
| 3rd Data | $RB_{16}$ | | | | | | | |
| 4th Data | $RB_{17}$ | | | | | | | |
| 5th Data | $RB_{18}$ | | | | | | | |
| 6th Data | $RB_{19}$ | | | | | | | |
| 7th Data | $RB_{20}$ | | | | | | | |
| 8th Data | $RB_{21}$ | | | | | | | |

Fig. 3.4 Block division for data format

# 4 DATA FORMATS

The controller function data formats are explained in the following.

The notation is that of the host's memory image.

## 4.1 Read format

This is the key data format when the controller function data are read.

When the host transmits Get Condition, the controller returns data according to the data format. The command is Data Transfer.

The data format size is 8 bytes.

The key data block to be used from these key data is defined by the function definition block.

| bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| 1st Data | Ra | La | Da | Ua | Start | A | B | C |
| 2nd Data | Rb | Lb | Db | Ub | D | X | Y | Z |
| 3rd Data | $A1_7$ | $A1_6$ | $A1_5$ | $A1_4$ | $A1_3$ | $A1_2$ | $A1_1$ | $A1_0$ |
| 4th Data | $A2_7$ | $A2_6$ | $A2_5$ | $A2_4$ | $A2_3$ | $A2_2$ | $A2_1$ | $A2_0$ |
| 5th Data | $A3_7$ | $A3_6$ | $A3_5$ | $A3_4$ | $A3_3$ | $A3_2$ | $A3_1$ | $A3_0$ |
| 6th Data | $A4_7$ | $A4_6$ | $A4_5$ | $A4_4$ | $A4_3$ | $A4_2$ | $A4_1$ | $A4_0$ |
| 7th Data | $A5_7$ | $A5_6$ | $A5_5$ | $A5_4$ | $A5_3$ | $A5_2$ | $A5_1$ | $A5_0$ |
| 8th Data | $A6_7$ | $A6_6$ | $A6_5$ | $A6_4$ | $A6_3$ | $A6_2$ | $A6_1$ | $A6_0$ |

Fig. 4.1 Read format

Key data explanation

1st : Digital button data. (ON = '0', OFF ='1')

2nd : Digital button data. (ON = '0', OFF = '1')

3rd : Analog axis 1 (A1) data.

4th : Analog axis 2 (A2) data.

5th : Analog axis 3 (A3) data.

6th : Analog axis 4 (A4) data.

7th : Analog axis 5 (A5) data.

8th : Analog axis 6 (A6) data.

The data of an unused button is fixed at '1'.

The data of an unused analog axis become the origin of the axis.

## 4.2 Write format

There is no write format for writing data to the controller functions.

Controller functions are read only.

# 5 SUPPORTED COMMANDS

The commands supported by the controller function among the commands specified in the “Maple Bus 1.0” Standard Specifications are described here.

If other commands than the following are sent, the controller function will return an error.

The concrete operations of the controller function for each command are also described.

## 5.1 Control command

Device Request

| | |
|---|---|
| Issuing right | : Host |
| Command code | : 01h |
| Data size | : 00h |
| Data field | : none |
| Expected return value | : [Device Status] |
| Description | : Requests controller function of destination AP to return Device Status. |
| | This command is also used to check port connection. |
| | Following initialization of the function and until this command is sent, the controller function does not respond to other commands. |
| | The LM-Bus is also disconnected for the expansion device so that the expansion device operation is stopped. |
| Order of operation | (1) Receives command. |
| | (2) Checks connection of each expansion device. |
| | (3) If connected, connects to corresponding LM-Bus. |
| | (4) Based on this result, the AP of originating device is created and returned to the host. |

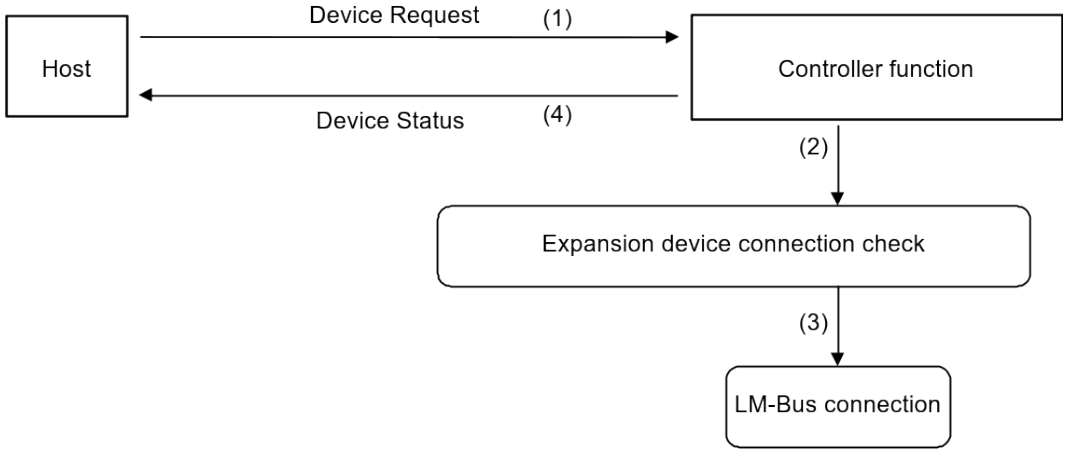


Fig. 5.1 Device Request

All Status Request

Issuing right : Host

Command code : 02h

Data size : 00h

Data field : none

Expected return value : [Device All Status]

Description : Requests controller function of destination AP to return all device statuses (both Fixed Device Status and Free Device Status).

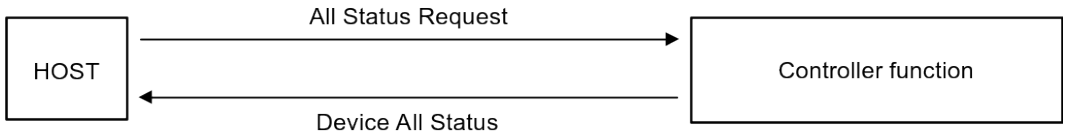


Fig. 5.2 All Status Request

Device Reset

Issuing right : Host

Command code : 03h

Data size : 00h

Data field : none

Expected return value : [Device Reply]

Description : Controller functions specified by the destination AP can be initialized.

Order of operation : (1) [Device Reply] returned.

(2) Initialization

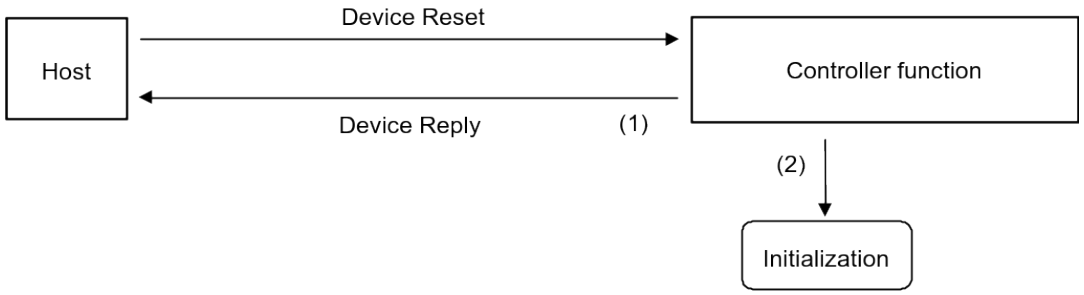


Fig. 5.3 Device Reset

Device Kill

Issuing right : Host

Command code : 04h

Data size : 00h

Data field : none

Expected return value : [Device Reply]

Description : Operation is recognized for the controller function specified by destination AP.

The function stands by in standby power consumption mode (minimum power consumption) and no commands can be received.

To start operation, a hard reset must be performed or the power should be turned off and then operation should be started again.

When the controller function which is a device receives the Device Kill command, the ID0,1 of the LM-Bus are reversed, expansion device is reset and the LM-Bus is disconnected.

Order of operation : (1) LM-Bus is reset and disconnected from the Maple-Bus.

(2) [Device Reply] returned.

(3) Operation terminated.

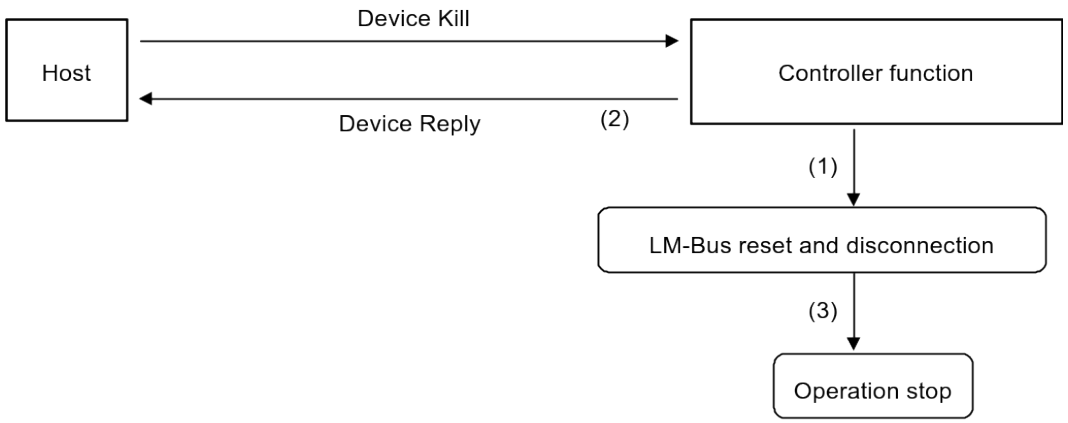


Fig. 5.4 Device Kill

Device Status

Issuing right : Controller function

Command code : 05h

Data size : 1Ch (28)

Data field : Device ID : 16 bytes

Destination code : 1 byte

Connection direction : 1 byte

Product name : 30 bytes

License : 60 bytes

Standby current consumption : 2 bytes

Maximum current consumption : 2 bytes

Description : In response to the Device Request from the host, Fixed Device Status data is returned.

Device All Status

| | | | |
|---|---|---|---|
| Issuing right | : Controller function | | |
| Command code | : 06h | | |
| Data size | : 1Ch+(n/4) | | |
| Data field | : Fixed Device Status | | : 112 bytes |
| | | Device ID | : 16 bytes |
| | | Destination code | : 1 byte |
| | | Connection direction | : 1 byte |
| | | Product name | : 30 byte |
| | | License | : 60byte |
| | | Standby current consumption | : 2 bytes |
| | | Maximum current consumption | : 2 bytes |
| | | Free Device Status | : n byte |
| Description | : In response to the All Status Request from the host, both Fixed Device Status and Free Device Status are returned. | | |

Device Reply

| | |
|---|---|
| Issuing right | : Controller function |
| Command code | : 07h |
| Data size | : 00h |
| Data field | : none |
| Description | : Used as response. |

Get Condition

Issuing right : Host

Command code : 09h

Data size : 01h

Data field : Function type : 4 bytes

Specifies the controller (00-00-00-01h).

Expected return value : [Data Transfer]

Description : This command requests the physical status (buttons, keys, lever statuses) of the controller function.

Used when the controller data should be read.

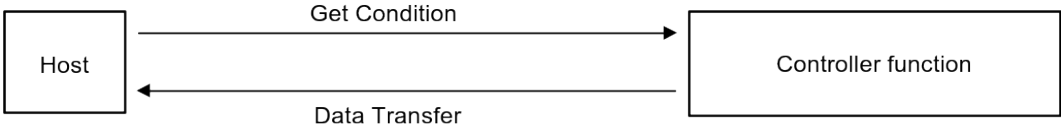


Fig. 5.5 Get Condition

| Data Address | Data | Setting example | Description |
|---|---|---|---|
| +0000 | Command code | 09h | Specifies Get Condition. |
| +0001 | Destination AP | 20h | Specifies device of port A. |
| +0002 | Origin AP | 00h | Send from Port A. |
| +0003 | Data size | 01h | Data size is 4 bytes. |
| +0004 | Function type | 00h | The function type specifies the controller. |
| +0005 | | 00h | |
| +0006 | | 00h | |
| +0007 | | 01h | |

Fig. 5.6 Data transmission from host (memory image)

Data Transfer

Issuing right : Controller function

Command code : 08h

Data size : 03h

Data field : Function type : 4 bytes

Specifies the controller (00-00-00-01h).

Reset format : 8 bytes

Expected return value : none

Description : This command is used to receive Get Condition from the host and when the controller's read format should be returned.

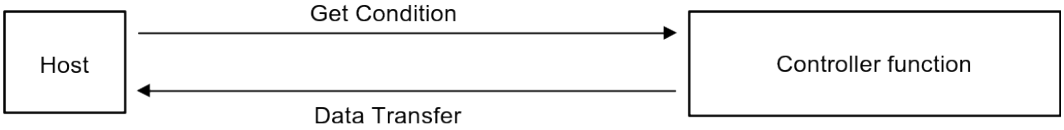


Fig. 5.7 Data Transfer

| Data Address | Data | Setting example | Description |
|---|---|---|---|
| +0000 | Command code | 08h | Specifies Data Transfer. |
| +0001 | Destination AP | 00h | Specifies device of port A. |
| +0002 | Origin AP | 20h | No expansion device |
| +0003 | Data size | 03h | Data size is 12 bytes. |
| +0004 | Function type | 00h | The function type specifies |
| +0005 | | 00h | the controller. |
| +0006 | | 00h | |
| +0007 | | 01h | |
| +0008 | Read format | FFh | According to the controller format, |
| +0009 | | FFh | the controller's data is stored. The |
| +000a | | 00h | blocks to be used have been |
| +000b | | 00h | declared already by the device ID. |
| +000c | | 80h | |
| +000d | | 80h | |
| +000e | | 80h | |
| +000f | | 80h | |

Fig. 5.8 Data transmission to host (host's memory image)

## 5.2 Error Commands

The controller function supports only 3 types of error commands.

Function Type Unknown

Issuing right : Controller function

Command code : FEh

Data size : 00h

Data field : none

Description : This command is returned when the sent function type is other than a controller.

Possible causes : (1) Mistaken specification of function type
(2) Data written incorrectly.
(3) Data of device ID jumbled.
(4) Data became jumbled during communication.

Remedies : (1) Specify function type correctly.
(2) Write data correctly.
(3) Resend Device Request to obtain device ID.
(4) Try sending again (maximum of three times; subsequent tries are processed as Time Out).

Command Unknown

Issuing right : Controller function

Command code : FDh

Data size : 00h

Data field : none

Description : This command is returned when the sent command is not supported by the controller function.

Possible causes : (1) Mistaken specification of function type
(2) Data written incorrectly.
(3) Data of device ID jumbled.
(4) Data became jumbled during communication.

Remedies : (1) Specify function type correctly.
(2) Write data correctly.
(3) Resend Device Request to obtain device ID.
(4) Try sending again (maximum of three times; subsequent tries are processed as Time Out).

Transmit Again

| | |
|---|---|
| Issuing right | : Host, controller function |
| Command code | : FCh |
| Data size | : 00h |
| Data field | : none |
| Description | : This command is used to request that the data be transmitted again when the data contained some kind of error.<br>However, since the controller is always updating the data, it is not always that the returned contents are identical with the contents of the controller's read data when the error occurred. |
| Possible causes | : (1) Parity error was generated.<br>(2) Data overflowed.<br>(3) Data became jumbled during communication.<br>(4) Others |
| Remedy | : Try sending again (maximum of three times; subsequent tries are processed as Time Out). |

# 6 CONTROLLER FUNCTION INFORMATION

This chapter explains information about specific devices (device statuses).

To prevent device statuses from being rewritten or erased, the data is recorded as is.

## 6.1 Types

Fixed Device Status

This is a set form of device status, consisting of 112 bytes in all, that must be designated.

Unless all items are designated, operation and connection are not guaranteed.

Free Device Status

The individual devices can use this status freely. It consists of 908 ~~912~~ bytes.

## 6.2 Fixed Device Status

The Fixed Device Status has the following items, all of which must be designated.

(1) Device ID

Capacity : 16 bytes

Description : Indicates the controller function's device ID.

If there is another function besides the controller, this must also be specified. (Maximum 2 types)

(2) Destination

Capacity : 1 byte

Description : Indicates product destination (sales 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 Configuration of setting bits for destination

| Destination | Setting bit |
|---|---|
| North American region | $DES_0$='1' |
| Japan region | $DES_1$='1' |
| Asia region | $DES_2$='1' |
| European region | $DES_3$='1' |
| Reserved region 1 | $DES_4$='1' |
| Reserved region 2 | $DES_5$='1' |
| Reserved region 3 | $DES_6$='1' |
| Reserved region 4 | $DES_7$='1' |

Fig. 6.2 Setting bits for destination

Worldwide : DES='11111111'=FFh

Except for special device types, the controller function is generally the same worldwide.

(3) Connection direction

Capacity : 1 byte

Description : Information differs for devices and expansion devices.

Device :

Indicates the direction of the expansion socket for expansion devices.

For details, refer to “Maple Bus 1.0” Standard Specifications.

(4)~~(3)~~ Product name Size : 30 ~~1~~ bytes

Description : Designates the product name in English or romaji. Either full-width or half-width characters can be used.

A space code (20h) is inserted for unused space.

(5)~~(4)~~ License

Size : 60 bytes

Description : Designates the license.

Generally, it designates “Produced By or Under License From SEGA ENTERPRISES,LTD.”

A space code (20h) is inserted for unused space.

(6)~~(5)~~ Standby current consumption

Size : 2 bytes

Description : Standby current consumption for temporary stop, in units of 0.1 mA, is designated in hexadecimal notation.

For example, 10.5 mA is designated by 00-69h.

(7)~~(6)~~ Maximum current consumption

Size : 2 bytes

Description : Maximum current consumption, in units of 0.1 mA, is designated in hexadecimal notation.

For example, 127.9 mA is designated by 04-FFh.

## 6.3 Free Device Status

The Free Device Status area is available for product planners, developers, designers and programmers to enter any information they wish. The host obtains this status by the All Device Request.

For details, see “Maple Bus 1.0” Standard Specifications.

# 7 AFTERWORD

Until the official version (Rev. 1.0) is distributed, contents will be modified to a small or large extent.