

gameport

COLLABORATORS

	<i>TITLE :</i> gameport		
<i>ACTION</i>	<i>NAME</i>	<i>DATE</i>	<i>SIGNATURE</i>
WRITTEN BY		July 19, 2024	

REVISION HISTORY

NUMBER	DATE	DESCRIPTION	NAME

Contents

1	gameport	1
1.1	gameport.doc	1
1.2	gameport.device/GPD_ASKCTYPE	1
1.3	gameport.device/GPD_ASKTRIGGER	1
1.4	gameport.device/GPD_READEVENT	2
1.5	gameport.device/GPD_SETCTYPE	3
1.6	gameport.device/GPD_SETTRIGGER	4

Chapter 1

gameport

1.1 gameport.doc

```
GPD_ASKCTYPE  
GPD_ASKTRIGGER  
GPD_READEVENT  
GPD_SETCTYPE  
GPD_SETTRIGGER
```

1.2 gameport.device/GPD_ASKCTYPE

NAME

```
GPD_ASKCTYPE -- Acquire the current game port controller type
```

FUNCTION

This command identifies the type of controller at the game port, so that the signals at the port may be properly interpreted. The controller type has been set by a previous SetCType.

This command always executes immediately.

IO REQUEST

```
io_Message  mn_ReplyPort set if quick I/O is not possible  
io_Device   preset by the call to OpenDevice  
io_Unit     preset by the call to OpenDevice  
io_Command  GPD_ASKCTYPE  
io_Flags    IOB_QUICK set if quick I/O is possible  
io_Length   at least 1  
io_Data     the address of the byte variable for the  
            result
```

1.3 gameport.device/GPD_ASKTRIGGER

NAME

GPD_ASKTRIGGER -- Inquire the conditions for a game port report

FUNCTION

This command inquires what conditions must be met by a game port unit before a pending Read request will be satisfied. These conditions, called triggers, are independent -- that any one occurs is sufficient to queue a game port report to the Read queue. These conditions are set by SetTrigger.

This command always executes immediately.

IO REQUEST

io_Message mn_ReplyPort set if quick I/O is not possible
 io_Device preset by the call to OpenDevice
 io_Unit preset by the call to OpenDevice
 io_Command GPD_ASKTRIGGER
 io_Flags IOB_QUICK set if quick I/O is possible
 io_Length sizeof(gamePortTrigger)
 io_Data a structure of type GamePortTrigger, which has the following elements

- gpt_Keys -
 - GPTB_DOWNKEYS set if button down transitions trigger a report, and GPTB_UPKEYS set if button up transitions trigger a report
- gpt_Timeout -
 - a time which, if exceeded, triggers a report; measured in vertical blank units (60/sec)
- gpt_XDelta -
 - a distance in x which, if exceeded, triggers a report
- gpt_YDelta -
 - a distance in y which, if exceeded, triggers a report

1.4 gameport.device/GPD_READEVENT

NAME

GPD_READEVENT -- Return the next game port event.

FUNCTION

Read game port events from the game port and put them in the data area of the iORequest. If there are no pending game port events, this command will not be satisfied, but if there are some events, but not as many as can fill IO_LENGTH, the request will be satisfied with those currently available.

IO REQUEST

io_Message mn_ReplyPort set if quick I/O is not possible
 io_Device preset by the call to OpenDevice
 io_Unit preset by the call to OpenDevice
 io_Command GPD_READEVENT
 io_Flags IOB_QUICK set if quick I/O is possible
 io_Length the size of the io_Data area in bytes: there

are sizeof(inputEvent) bytes per input event.

io_Data a buffer area to fill with input events. The fields of the input event are:

- ie_NextEvent
links the events returned
- ie_Class
is IECLASS_RAWMOUSE
- ie_SubClass
is 0 for the left, 1 for the right game port
- ie_Code
contains any gameport button reports. No report is indicated by the value 0xff.
- ie_Qualifier
only the relative and button bits are set
- ie_X, ie_Y
the x and y values for this report, in either relative or absolute device dependent units.
- ie_TimeStamp
the delta time since the last report, given not as a standard timestamp, but as the frame count in the TV_SECS field.

RESULTS

This function sets the error field in the iORequest, and fills the iORequest with the next game port events (but not partial events).

SEE ALSO

gameport.device/SetCType, gameport.device/SetTrigger

1.5 gameport.device/GPD_SETCTYPE

NAME

GPD_SETCTYPE -- Set the current game port controller type

FUNCTION

This command sets the type of device at the game port, so that the signals at the port may be properly interpreted. The port can also be turned off, so that no reports are generated.

This command always executes immediately.

IO REQUEST

- io_Message mn_ReplyPort set if quick I/O is not possible
- io_Device preset by the call to OpenDevice
- io_Unit preset by the call to OpenDevice
- io_Command GPD_SETCTYPE
- io_Flags IOB_QUICK set if quick I/O is possible
- io_Length 1
- io_Data the address of the byte variable describing the controller type, as per the equates in the gameport include file

1.6 gameport.device/GPD_SETTRIGGER

NAME

GPD_SETTRIGGER -- Set the conditions for a game port report

FUNCTION

This command sets what conditions must be met by a game port unit before a pending Read request will be satisfied. These conditions, called triggers, are independent -- that any one occurs is sufficient to queue a game port report to the Read queue. These conditions are inquired with AskTrigger.

This command always executes immediately.

IO REQUEST

io_Message mn_ReplyPort set if quick I/O is not possible
io_Device preset by the call to OpenDevice
io_Unit preset by the call to OpenDevice
io_Command GPD_SETTRIGGER
io_Flags IOB_QUICK set if quick I/O is possible
io_Length sizeof(gamePortTrigger)
io_Data a structure of type GamePortTrigger, which has the following elements
gpt_Keys -
GPTB_DOWNKEYS set if button down transitions trigger a report, and GPTB_UPKEYS set if button up transitions trigger a report
gpt_Timeout -
a time which, if exceeded, triggers a report; measured in vertical blank units (60/sec)
gpt_XDelta -
a distance in x which, if exceeded, triggers a report
gpt_YDelta -
a distance in y which, if exceeded, triggers a report