keyboard

COLLABORATORS			

	TITLE : keyboard		
ACTION	*NAME*	*DATE*	*SIGNATURE*
WRITTEN BY		March 28, 2025	

REVISION HISTORY			

NUMBER	DATE	DESCRIPTION	NAME

Contents

Chapter 1

keyboard

1.1 keyboard.doc

```
CMD_CLEAR               KBD_READEVENT            KBD_RESETHANDLERDONE
CMD_RESET               KBD_READMATRIX
KBD_ADDRESETHANDER      KBD_REMRESETHANDLER
```

1.2 keyboard.device/CMD_CLEAR

```
NAME
    CMD_CLEAR - clear keyboard input buffer

FUNCTION
    Remove from the input buffer any keys transitions waiting to
    satisfy read requests.

IO REQUEST
    io_Message      mn_ReplyPort set if quick I/O is not possible
    io_Device       preset by the call to OpenDevice
    io_Command      CMD_CLEAR
    io_Flags        IOB_QUICK set if quick I/O is possible
```

1.3 keyboard.device/CMD_RESET

```
NAME
    CMD_RESET - reset the keyboard

FUNCTION
    CMD_RESET resets the keyboard device without destroying handles
    to the open device.

IO REQUEST
    io_Message      mn_ReplyPort set if quick I/O is not possible
    io_Device       preset by the call to OpenDevice
    io_Command      CMD_RESET
    io_Flags        IOB_QUICK set if quick I/O is possible
```

1.4 keyboard.device/KBD_ADDRESETHANDER

```
NAME
    KBD_ADDRESETHANDER – add a reset handler to the device

FUNCTION
    Add a function to the list of functions called to clean up
    before a hard reset:
        Handler(handlerData);
                A1

    Note that the A500 does not support this.  CTRL-Amiga-Amiga
    on an A500 does an immediate hard processor reset.

IO REQUEST
    io_Message      mn_ReplyPort set
    io_Device       preset by OpenDevice
    io_Unit         preset by OpenDevice
    io_Command      KBD_ADDRESETHANDLER
    io_Data         a pointer to an interrupt structure.
        is_Data     the handlerData pointer described above
        is_Code     the Handler function address

NOTES
    The interrupt structure is kept by the keyboard device until a
    KBD_REMRESETHANDLER command is satisfied for it.
```

1.5 keyboard.device/KBD_READEVENT

```
NAME
    KBD_READEVENT – return the next keyboard event.

FUNCTION
    Read raw keyboard events from the keyboard and put them in the
    data area of the iORequest.  If there are no pending keyboard
    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_Command      KBD_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_RAWKEY
        ie_Code
                    contains the next key up/down reports
```

```
        ie_Qualifier
                    only the shift and numeric pad bits are set
        ie_SubClass, ie_X, ie_Y, ie_TimeStamp
                    are not used, and set to zero
```

RESULTS
    This function sets the error field in the IORequest, and fills
    the IORequest with the next keyboard events (but not partial
    events).

1.6 keyboard.device/KBD_READMATRIX

NAME
    KBD_READMATRIX – read the current keyboard key matrix

FUNCTION
    This function reads the up/down state of every key in the
    key matrix.

IO REQUEST
```
    io_Message      mn_ReplyPort set if quick I/O is not possible
    io_Device       preset by the call to OpenDevice
    io_Command      KBD_READMATRIX
    io_Flags        IOB_QUICK set if quick I/O is possible
    io_Length       the size of the io_Data area in bytes: this
                    must be big enough to hold the key matrix.
    io_Data         a buffer area to fill with the key matrix:
                    an array of bytes whose component bits reflect
                    each keys state: the state of the key for
                    keycode n is at bit (n MOD 8) in byte
                    (n DIV 8) of this matrix.
```

NOTE
    For V1.2/V1.3 Kickstart, io_Length must be set to exactly 13 bytes.

RESULTS
    This function sets the error field in the IORequest, and sets
    matrix to the current key matrix.

1.7 keyboard.device/KBD_REMRESETHANDLER

NAME
    KBD_REMRESETHANDLER – remove a reset handler from the device

FUNCTION
    Remove a function previously added to the list of handler
    functions.

IO REQUEST
```
    io_Message      mn_ReplyPort set
    io_Device       preset by OpenDevice
    io_Unit         preset by OpenDevice
```

```
    io_Command        KBD_REMRESETHANDLER
    io_Data           a pointer to the handler interrupt structure.
```

1.8   keyboard.device/KBD_RESETHANDLERDONE

NAME
    KBD_RESETHANDLERDONE – indicate that reset can occur

FUNCTION
    Indicate that reset cleanup associated with the handler has
    completed.

IO REQUEST
```
    io_Message        mn_ReplyPort set
    io_Device         preset by OpenDevice
    io_Unit           preset by OpenDevice
    io_Command        KBD_RESETHANDLERDONE
    io_Data           a pointer to the handler interrupt structure.
```