NAME

hidraw - Raw Access to HID devices

SYNOPSIS

To compile this driver into the kernel, place the following line in your kernel configuration file:

```
device hidraw
device hid
device hidbus
```

Alternatively, to load the driver as a module at boot time, place the following line in loader.conf(5):

```
hidraw_load="YES"
```

DESCRIPTION

The **hidraw** driver provides a raw interface to Human Interface Devices (HIDs). The reports are sent to and received from the device unmodified.

The device handles 2 sets of ioctl(2) calls:

FreeBSD uhid(4) -compatible calls:

HIDRAW_GET_REPORT_ID (*int*)

Get the report identifier used by this HID report.

HIDRAW_GET_REPORT_DESC (*struct hidraw_gen_descriptor*)

Get the HID report descriptor. Copies a maximum of *hgd_maxlen* bytes of the report descriptor data into the memory specified by *hgd_data*. Upon return *hgd_actlen* is set to the number of bytes copied. Using this descriptor the exact layout and meaning of data to/from the device can be found. The report descriptor is delivered without any processing.

```
struct hidraw_gen_descriptor {
        void   *hgd_data;
        uint16_t hgd_maxlen;
        uint16_t hgd_actlen;
        uint8_t     hgd_report_type;
        ...
};
```

HIDRAW_SET_IMMED (*int*)

Sets the device in a mode where each read(2) will return the current value of the input report. Normally a read(2) will only return the data that the device reports on its interrupt pipe. This call may fail if the device does not support this feature.

HIDRAW_GET_REPORT (*struct hidraw_gen_descriptor*)

Get a report from the device without waiting for data on the interrupt pipe. Copies a maximum of *hgd_maxlen* bytes of the report data into the memory specified by *hgd_data*. Upon return *hgd_actlen* is set to the number of bytes copied. The *hgd_report_type* field indicates which report is requested. It should be HID_INPUT_REPORT, HID_OUTPUT_REPORT, or HID_FEATURE_REPORT. On a device which uses numbered reports, the first byte of the returned data is the report number. The report data begins from the second byte. For devices which do not use numbered reports, the report data begins at the first byte. This call may fail if the device does not support this feature.

HIDRAW_SET_REPORT (*struct hidraw_gen_descriptor*)

Set a report in the device. The *hgd_report_type* field indicates which report is to be set. It should be HID_INPUT_REPORT, HID_OUTPUT_REPORT, or HID_FEATURE_REPORT. The value of the report is specified by the *hgd_data* and the *hgd_maxlen* fields. On a device which uses numbered reports, the first byte of data to be sent is the report number. The report data begins from the second byte. For devices which do not use numbered reports, the report data begins at the first byte. This call may fail if the device does not support this feature.

HIDRAW_GET_DEVICEINFO (*struct hidraw_device_info*)

Returns information about the device, like vendor ID and product ID. This call will not issue any hardware transfers.

Linux **hidraw** -compatible calls:

HIDIOCGRDESCSIZE (*int*)

Get the HID report descriptor size. Returns the size of the device report descriptor to use with HIDIOCGRDESC ioctl(2).

HIDIOCGRDESC (*struct hidraw_report_descriptor*)

Get the HID report descriptor. Copies a maximum of *size* bytes of the report descriptor data into the memory specified by *value*.

```
struct hidraw_report_descriptor {
        uint32_t  size;
        uint8_t           value[HID_MAX_DESCRIPTOR_SIZE];
};
```

HIDIOCGRAWINFO (*struct hidraw_devinfo*)

Get structure containing the bus type, the vendor ID (VID), and product ID (PID) of the device. The bus type can be one of: BUS_USB or BUS_I2C which are defined in dev/evdev/input.h.

```
struct hidraw_devinfo {
        uint32_t  bustype;
        int16_t         vendor;
        int16_t         product;
};
```

HIDIOCGRAWNAME(len) (*char[] buf*)

Get device description. Copies a maximum of *len* bytes of the device description previously set with device_set_desc(9) procedure into the memory specified by *buf*.

HIDIOCGRAWPHYS(len) (*char[] buf*)

Get the newbus path to the device. Copies a maximum of *len* bytes of the newbus device path into the memory specified by *buf*.

HIDIOCGFEATURE(len) (*void[] buf*)

Get a feature report from the device. Copies a maximum of *len* bytes of the feature report data into the memory specified by *buf*. The first byte of the supplied buffer should be set to the report number of the requested report. For devices which do not use numbered reports, set the first byte to 0. The report will be returned starting at the first byte of the buffer (ie: the report number is not returned). This call may fail if the device does not support this feature.

HIDIOCSFEATURE(len) (*void[] buf*)

Set a feature Report in the device. The value of the report is specified by the *buf* and the *len* parameters. Set the first byte of the supplied buffer to the report number. For devices which do not use numbered reports, set the first byte to 0. The report data begins in the second byte. Make sure to set len accordingly, to one more than the length of the report (to account for the report number). This call may fail if the device does not support this feature.

Use read(2) to get data from the device. Data should be read in chunks of the size prescribed by the report descriptor. On a device which uses numbered reports, the first byte of the returned data is the report number. The report data begins from the second byte. For devices which do not use numbered reports, the report data begins at the first byte.

Use write(2) to send data to the device. Data should be written in chunks of the size prescribed by the report descriptor. The first byte of the buffer passed to write(2) should be set to the report number. If the device does not use numbered reports, there are 2 operation modes: **hidraw** mode and uhid(4) mode.

In the **hidraw** mode, the first byte should be set to 0 and the report data itself should begin at the second byte. In the uhid(4) mode, the report data should begin at the first byte. The modes can be switched with issuing one of HIDIOCGRDESC or HID_GET_REPORT_DESC ioctl(2) accordingly. Default mode is **hidraw**.

SYSCTL VARIABLES

The following variables are available as both sysctl(8) variables and loader(8) tunables:

hw.hid.hidraw.debug

Debug output level, where 0 is debugging disabled and larger values increase debug message verbosity. Default is 0.

FILES

/dev/hidraw?

SEE ALSO

usbhidctl(1), hid(4), hidbus(4), uhid(4)

HISTORY

The uhid(4) driver appeared in NetBSD 1.4. **hidraw** protocol support was added in FreeBSD 13 by Vladimir Kondratyev <*wulf@FreeBSD.org*>. This manual page was adopted from NetBSD by Tom Rhodes <*trhodes@FreeBSD.org*> in April 2002.