## NAME

**usb** - Universal Serial Bus

## SYNOPSIS

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

```
device usb
```

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

```
usb_load="YES"
```

## USERLAND PROGRAMMING

USB functions can be accessed from userland through the libusb library. See libusb(3) for more information.

## DESCRIPTION

FreeBSD provides machine-independent bus support and drivers for USB devices in host and device side mode.

The **usb** driver has three layers:

USB Controller (Bus)

USB Device

USB Driver

The controller attaches to a physical bus like pci(4). The USB bus attaches to the controller, and the root hub attaches to the controller. Any devices attached to the bus will attach to the root hub or another hub attached to the USB bus.

The **uhub** device will always be present as it is needed for the root hub.

## INTRODUCTION TO USB

The USB is a system where external devices can be connected to a PC. The most common USB speeds are:

Low Speed (1.5 MBit/sec)

Full Speed (12 MBit/sec)

High Speed (480 MBit/sec)

SuperSpeed (5 GBit/sec)

Each USB has a USB controller that is the master of the bus. The physical communication is simplex which means the host controller only communicates with one USB device at a time.

There can be up to 127 devices connected to an USB HUB tree. The addresses are assigned dynamically by the host when each device is attached to the bus.

Within each device there can be up to 16 endpoints. Each endpoint is individually addressed and the addresses are static. Each of these endpoints will communicate in one of four different modes: *control*, *isochronous*, *bulk*, or *interrupt*. A device always has at least one endpoint. This endpoint has address 0 and is a control endpoint and is used to give commands to and extract basic data, such as descriptors, from the device. Each endpoint, except the control endpoint, is unidirectional.

The endpoints in a device are grouped into interfaces. An interface is a logical unit within a device, e.g., a compound device with both a keyboard and a trackball, would present one interface for each. An interface can sometimes be set into different modes, called alternate settings, which affects how it operates. Different alternate settings can have different endpoints within it.

A device may operate in different configurations. Depending on the configuration, the device may present different sets of endpoints and interfaces.

The bus enumeration of the USB bus proceeds in several steps:

1. Any interface specific driver can attach to the device.

2. If none is found, generic interface class drivers can attach.

## SYSCTL VARIABLES

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

*hw.usb.debug*

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

## SEE ALSO

The USB specifications can be found at:

*https://www.usb.org/documents*

libusb(3), aue(4), axe(4), axge(4), cue(4), ehci(4), kue(4), mos(4), ohci(4), pci(4), rue(4), ucom(4), udav(4), uhci(4), uhid(4), ukbd(4), ulpt(4), umass(4), ums(4), uplcom(4), urio(4), uvscom(4), xhci(4) usbconfig(8), usbdi(9)

## STANDARDS

The **usb** module complies with the USB 3.0 standard.

## HISTORY

The **usb** module has been inspired by the NetBSD USB stack initially written by Lennart Augustsson. The **usb** module was written by Hans Petter Selasky <*hselasky@FreeBSD.org*>.