# Universal Serial Bus

<table>
<thead>
<tr>
<th><strong>Type</strong></th>
<th>Computer Hardware Bus</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Production history</strong></td>
<td></td>
</tr>
<tr>
<td><strong>Designer</strong></td>
<td>Ajay Bhatt, Intel</td>
</tr>
<tr>
<td><strong>Designed</strong></td>
<td>January 1996</td>
</tr>
<tr>
<td><strong>Manufacturer</strong></td>
<td>Intel, Compaq, Microsoft, NEC, Digital Equipment Corporation, IBM, Nortel</td>
</tr>
<tr>
<td><strong>Superseded</strong></td>
<td>Serial port, Parallel port, Game port, Apple Desktop Bus, PS/2 connector</td>
</tr>
<tr>
<td><strong>Specifications</strong></td>
<td></td>
</tr>
<tr>
<td><strong>Length</strong></td>
<td>5 metre (maximum)</td>
</tr>
<tr>
<td><strong>Width</strong></td>
<td>11.5 mm (A-plug), 8.45 mm (B-plug),</td>
</tr>
<tr>
<td><strong>Height</strong></td>
<td>4.5 mm (A-plug), 7.78 mm (B-plug, pre-v3.0)</td>
</tr>
<tr>
<td><strong>Hot pluggable</strong></td>
<td>Yes</td>
</tr>
<tr>
<td><strong>External</strong></td>
<td>Yes</td>
</tr>
<tr>
<td><strong>Electrical</strong></td>
<td>5 volt DC</td>
</tr>
<tr>
<td></td>
<td>Max. voltage 5 volts</td>
</tr>
<tr>
<td></td>
<td>Max. current 500-900 mA @ 5 V (depending on version)</td>
</tr>
<tr>
<td><strong>Data signal</strong></td>
<td>Packet data, defined by specifications</td>
</tr>
<tr>
<td><strong>Width</strong></td>
<td>1 bit</td>
</tr>
<tr>
<td><strong>Bandwidth</strong></td>
<td>1.5-4800 Mb/s</td>
</tr>
<tr>
<td><strong>Max. devices</strong></td>
<td>127</td>
</tr>
<tr>
<td><strong>Protocol</strong></td>
<td>Serial</td>
</tr>
<tr>
<td><strong>Cable</strong></td>
<td>4 wires</td>
</tr>
<tr>
<td><strong>Pins</strong></td>
<td>4 (1 supply, 2 data, 1 ground)</td>
</tr>
<tr>
<td><strong>Connector</strong></td>
<td>Unique</td>
</tr>
</tbody>
</table>

## Pin out

The standard USB A plug (left) and B plug (right)

<table>
<thead>
<tr>
<th>Pin 1</th>
<th>Pin 2</th>
<th>Pin 3</th>
</tr>
</thead>
<tbody>
<tr>
<td>V_{CC} (+5 V)</td>
<td>Data-</td>
<td>Data+</td>
</tr>
</tbody>
</table>
Universal Serial Bus (USB) is a specification to establish communication between devices and a host controller (usually a personal computer), developed and invented by Ajay Bhatt, while working for Intel. USB has effectively replaced a variety of interfaces such as serial and parallel ports.

USB can connect computer peripherals such as mice, keyboards, digital cameras, printers, personal media players, flash drives, Network Adapters, and external hard drives. For many of those devices, USB has become the standard connection method.

USB was designed for personal computers, but it has become commonplace on other devices such as smartphones, PDAs and video game consoles, and as a power cord. As of 2008, there are about 2 billion USB devices sold per year, and approximately 6 billion total sold to date.

Unlike older connection standards such as RS-232 or Parallel port, USB connectors also supply electric power, so many devices connected by USB do not need a power source of their own.

History

The USB is a standard for peripheral devices. It began development in 1994 by a group of seven companies: Compaq, DEC, IBM, Intel, Microsoft, NEC and Nortel. USB was intended to make it fundamentally easier to connect external devices to PCs by replacing the multitude of connectors at the back of PCs, addressing the usability issues of existing interfaces, and simplifying software configuration of all devices connected to USB, as well as permitting greater bandwidths for external devices. The first silicon for USB was made by Intel in 1995.

The USB 1.0 specification was introduced in January 1996. The original USB 1.0 specification had a data transfer rate of 1.5 Mbit/s. The first widely used version of USB was 1.1, which was released in September 1998. It allowed for a 12 Mbit/s data rate for higher-speed devices such as disk drives, and a lower 1.5 Mbit/s rate for low bandwidth devices such as joysticks.

The USB 2.0 specification was released in April 2000 and was standardized by the USB-IF at the end of 2001. Hewlett-Packard, Intel, Lucent Technologies (now Alcatel-Lucent), NEC and Philips jointly led the initiative to develop a higher data transfer rate, with the resulting specification achieving 480 Mbit/s, a fortyfold increase over 12 Mbit/s for the original USB 1.1.

The USB 3.0 Specification was published in 12 November 2008. Its main goals were to increase data transfer rate (up to 5Gbps), decrease power consumption, increase power output and be backwards compatible with USB 2.0. USB 3.0 includes a new, higher speed bus called SuperSpeed in parallel with the USB 2.0 bus. For this reason, the new version is also called SuperSpeed. The first USB 3.0 equipped devices were presented in January 2010.
The System

A USB system has an asymmetric design, consisting of a host, a multitude of downstream USB ports, and multiple peripheral devices connected in a tiered-star topology. Additional USB hubs may be included in the tiers, allowing branching into a tree structure with up to five tier levels. A USB host may have multiple host controllers and each host controller may provide one or more USB ports. Up to 127 devices, including hub devices if present, may be connected to a single host controller.\[11]\[12]

USB devices are linked in series through hubs. There always exists one hub known as the root hub, which is built into the host controller.

So-called sharing hubs, which allow multiple computers to access the same peripheral device(s), also exist and work by switching access between PCs, either automatically or manually. Sharing hubs are popular in small-office environments. In network terms, they converge rather than diverge branches.

A physical USB device may consist of several logical sub-devices that are referred to as device functions. A single device may provide several functions, for example, a webcam (video device function) with a built-in microphone (audio device function). Such a device is called a compound device in which each logical device is assigned a distinctive address by the host and all logical devices are connected to a built-in hub to which the physical USB wire is connected. A host assigns one and only one device address to a function.

USB device communication is based on pipes (logical channels). A pipe is a connection from the host controller to a logical entity, found on a device, and named an endpoint. The term endpoint is occasionally incorrectly used for referring to the pipe. However, while an endpoint exists on the device permanently, a pipe is only formed when the host makes a connection to the endpoint. Therefore, when referring to a particular connection between a host and a USB device function, the term pipe should be used. A USB device can have up to 32 endpoints: 16 into the host controller and 16 out of the host controller. But, as one of the pipes is required to be of a bi-directional type (the default control pipe), and thus uses 2 endpoints, the theoretical maximum number of pipes is 31. USB devices seldom have this many endpoints.

There are two types of pipes: stream and message pipes depending on the type of data transfer.

- **isochronous transfers**: at some guaranteed data rate (often, but not necessarily, as fast as possible) but with possible data loss (e.g. realtime audio or video).
- **interrupt transfers**: devices that need guaranteed quick responses (bounded latency) (e.g. pointing devices and keyboards).
- **bulk transfers**: large sporadic transfers using all remaining available bandwidth, but with no guarantees on bandwidth or latency (e.g. file transfers).
- **control transfers**: typically used for short, simple commands to the device, and a status response, used, for example, by the bus control pipe number 0.

A stream pipe is a uni-directional pipe connected to a uni-directional endpoint that transfers data using an isochronous, interrupt, or bulk transfer. A message pipe is a bi-directional pipe connected to a bi-directional endpoint that is exclusively used for control data flow. An endpoint is built into the USB device by the manufacturer and therefore exists permanently. An endpoint of a pipe is addressable with tuple (device_address, endpoint_number) as specified in a TOKEN packet that the host sends when it wants to start a data transfer session. If the direction of the data transfer is from the host to the endpoint, an OUT packet (a specialization of a TOKEN packet) having the desired device address and endpoint number is sent by the host. If the direction of the data
transfer is from the device to the host, the host sends an IN packet instead. If the destination endpoint is a uni-directional endpoint whose manufacturer's designated direction does not match the TOKEN packet (e.g., the manufacturer's designated direction is IN while the TOKEN packet is an OUT packet), the TOKEN packet will be ignored. Otherwise, it will be accepted and the data transaction can start. A bi-directional endpoint, on the other hand, accepts both IN and OUT packets.

Endpoints are grouped into *interfaces* and each interface is associated with a single device function. An exception to this is endpoint zero, which is used for device configuration and which is not associated with any interface. A single device function composed of independently controlled interfaces is called a *composite device*. A composite device only has a single device address because the host only assigns a device address to a function.

When a USB device is first connected to a USB host, the USB device enumeration process is started. The enumeration starts by sending a reset signal to the USB device. The data rate of the USB device is determined during the reset signaling. After reset, the USB device's information is read by the host and the device is assigned a unique 7-bit address. If the device is supported by the host, the device drivers needed for communicating with the device are loaded and the device is set to a configured state. If the USB host is restarted, the enumeration process is repeated for all connected devices.

The host controller directs traffic flow to devices, so no USB device can transfer any data on the bus without an explicit request from the host controller. In USB 2.0, the host controller polls the bus for traffic, usually in a round-robin fashion. The slowest device connected to a controller sets the bandwidth of the interface. For SuperSpeed USB (defined since USB 3.0), connected devices can request service from host. Because there are two separate controllers in each USB 3.0 host, USB 3.0 devices will transmit and receive at USB 3.0 data rates regardless of USB 2.0 or earlier devices connected to that host. Operating data rates for them will be set in the legacy manner.

**Device classes**

USB 3 defines class codes used to identify a device's functionality and to load a device driver based on that functionality. This enables every device driver writer to support devices from different manufacturers that comply with a given class code.

Device classes include:[13]

<table>
<thead>
<tr>
<th>Class</th>
<th>Usage</th>
<th>Description</th>
<th>Examples</th>
</tr>
</thead>
<tbody>
<tr>
<td>00h</td>
<td>Device</td>
<td>Unspecified[14]</td>
<td>(Device class is unspecified. Interface descriptors are used for determining the required drivers.)</td>
</tr>
<tr>
<td>01h</td>
<td>Interface</td>
<td>Audio</td>
<td>Speaker, microphone, sound card, MIDI</td>
</tr>
<tr>
<td>02h</td>
<td>Both</td>
<td>Communications and CDC Control</td>
<td>Modem, Ethernet adapter, Wi-Fi adapter</td>
</tr>
<tr>
<td>03h</td>
<td>Interface</td>
<td>Human interface device (HID)</td>
<td>Keyboard, mouse, joystick</td>
</tr>
<tr>
<td>05h</td>
<td>Interface</td>
<td>Physical Interface Device (PID)</td>
<td>Force feedback joystick</td>
</tr>
<tr>
<td>06h</td>
<td>Interface</td>
<td>Image</td>
<td>Webcam, scanner</td>
</tr>
<tr>
<td>07h</td>
<td>Interface</td>
<td>Printer</td>
<td>Laser printer, inkjet printer, CNC machine</td>
</tr>
<tr>
<td>08h</td>
<td>Interface</td>
<td>Mass storage</td>
<td>USB flash drive, memory card reader, digital audio player, digital camera, external drive</td>
</tr>
<tr>
<td>Code</td>
<td>Device/Interface</td>
<td>Description</td>
<td></td>
</tr>
<tr>
<td>------</td>
<td>-----------------</td>
<td>-------------</td>
<td></td>
</tr>
<tr>
<td>09h</td>
<td>Device</td>
<td>USB hub</td>
<td></td>
</tr>
<tr>
<td>0Ah</td>
<td>Interface</td>
<td>CDC-Data</td>
<td></td>
</tr>
<tr>
<td>0Bh</td>
<td>Interface</td>
<td>Smart Card</td>
<td></td>
</tr>
<tr>
<td>0Dh</td>
<td>Interface</td>
<td>Content security</td>
<td></td>
</tr>
<tr>
<td>0Eh</td>
<td>Interface</td>
<td>Video</td>
<td></td>
</tr>
<tr>
<td>0Fh</td>
<td>Interface</td>
<td>Personal Healthcare</td>
<td></td>
</tr>
<tr>
<td>DCh</td>
<td>Both</td>
<td>Diagnostic Device</td>
<td></td>
</tr>
<tr>
<td>E0h</td>
<td>Interface</td>
<td>Wireless Controller</td>
<td></td>
</tr>
<tr>
<td>EFh</td>
<td>Both</td>
<td>Miscellaneous</td>
<td></td>
</tr>
<tr>
<td>FEh</td>
<td>Interface</td>
<td>Application-specific</td>
<td></td>
</tr>
<tr>
<td>FFh</td>
<td>Both</td>
<td>Vendor-specific</td>
<td></td>
</tr>
</tbody>
</table>

**USB mass storage**

USB implements connections to storage devices using a set of standards called the *USB mass storage device class* (referred to as MSC or UMS). This was initially intended for traditional magnetic and optical drives, but has been extended to support a wide variety of devices, particularly flash drives. This generality is because many systems can be controlled with the familiar metaphor of file manipulation within directories (the process of making a novel device look like a familiar device is also known as extension). The ability to boot a write-locked SD card with a USB adapter is particularly advantageous for maintaining the integrity and non-corruptible, pristine state of the booting medium.

Though most newer computers are capable of booting off USB mass storage devices, USB is not intended to be a primary bus for a computer's internal storage: buses such as Parallel ATA (PATA) (or IDE), Serial ATA (SATA), or SCSI fulfill that role in PC class computers. However, USB has one important advantage in that it is possible to install and remove devices without rebooting the computer (hot-swapping), making it useful for mobile peripherals, including drives of various kinds. Originally conceived and still used today for optical storage devices (CD-RW drives, DVD drives, etc.), several manufacturers offer external portable USB hard drives, or empty enclosures for disk drives, which offer performance comparable to internal drives, limited by the current number and type of attached USB devices and by the upper limit of the USB interface (in practice about 40 MB/s for USB 2.0 and potentially 400 MB/s or more[17] for USB 3.0). These external drives have typically included a "translating device" that bridges between a drive's interface (IDE, ATA, SATA, PATA, ATAPI, or even SCSI) to a USB interface port. Functionally, the drive appears to the user much like an internal drive. Other competing standards for external drive connectivity include eSATA, ExpressCard (now at version 2.0), and FireWire (IEEE 1394).

Another use for USB mass storage devices is the portable execution of software applications (such as web browsers and VoIP clients) without requiring installation on the host computer.[18][19]
Human interface devices (HIDs)

Mice and keyboards usually have USB connectors. These can be used with older computers that have PS/2 connectors with the aid of a small USB-to-PS/2 adapter. Such adaptors contain no logic circuitry: the hardware in the USB keyboard or mouse is designed to detect whether it is connected to a USB or PS/2 port, and communicate using the appropriate protocol.

Joysticks, keypads, tablets and other human-interface devices are also progressively migrating from MIDI, and PC game port connectors to USB.

Physical appearance

Pinouts of Standard, Mini, and Micro USB plugs. The USB logo is on the bottom of the two micro-USB plugs (as they are shown in this figure) but on the top of the other plugs.[20]

USB 1.x/2.0 standard pinning

<table>
<thead>
<tr>
<th>Pin</th>
<th>Name</th>
<th>Cable color</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>VBUS</td>
<td>Red</td>
<td>+5 V</td>
</tr>
<tr>
<td>2</td>
<td>D−</td>
<td>White</td>
<td>Data −</td>
</tr>
<tr>
<td>3</td>
<td>D+</td>
<td>Green</td>
<td>Data +</td>
</tr>
<tr>
<td>4</td>
<td>GND</td>
<td>Black</td>
<td>Ground</td>
</tr>
</tbody>
</table>

Micro-B USB 3.0 compatible socket

USB 2.0 connector on the side of the specification standard micro USB 3.0 connector are aligned pin-minute increase in the standard.

No.1: power (VBUS)
No.2: USB2.0 differential pair (D−)
No.3: USB2.0 differential pair (D+)
No.4: USB OTG ID for identifying lines
No.5: GND
No.6: USB3.0 signal transmission line (−)
No.7: USB3.0 signal transmission line (+)
No.8: GND
No.9: USB3.0 signal receiving line (−)
No.10: USB3.0 signal receiving line (+)
USB 1.x/2.0 Mini/Micro pinning

<table>
<thead>
<tr>
<th>Pin</th>
<th>Name</th>
<th>Cable color</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>VBUS</td>
<td>Red</td>
<td>+5 V</td>
</tr>
<tr>
<td>2</td>
<td>D−</td>
<td>White</td>
<td>Data −</td>
</tr>
<tr>
<td>3</td>
<td>D+</td>
<td>Green</td>
<td>Data +</td>
</tr>
</tbody>
</table>
| 4   | ID   | none        | permits distinction of A plug from B plug  
|     |      |             | * A plug: connected to Signal Ground  
|     |      |             | * B plug: not connected |
| 5   | GND  | Black       | Signal Ground |

Connector properties

The connectors specified by the USB committee were designed to support a number of USB’s underlying goals, and to reflect lessons learned from the menagerie of connectors which have been used in the computer industry.

Usability and "upside down" connectors

It is by design difficult to attach a USB connector incorrectly. Most connectors cannot be plugged in upside down and it is clear from kinesthetic sensation of making a connection when the plug and socket are correctly mated. However, it is not obvious whether the connector should be face up or face down, and thus it is often necessary to try the insertion both ways. The side of the connector on a USB cable or other product with the "USB Icon" (trident logo) should be "visible" to the user during the mating process. Some manufacturers do not, however, make the trident logo on USB cables easily visible or detectable by touch. Additionally, some computers such as the Mac Mini, have the ports vertically. In many Japanese computers and other devices, the trident logo on the cable or other USB device being inserted must be facing down, while in many American computers and devices the logo must be facing up. However, this is not always the case, such as with some Dell (an American company) computers.

Officially, the USB 2.0 specification states that the required USB Icon is to be "embossed" ("engraved" on the accompanying diagram) on the "topside" of the USB plug, which "provides easy user recognition and facilitates alignment during the mating process."[21] The specification also shows that the "recommended" (optional) "Manufacturer’s logo" ("engraved" on the diagram but not specified in the text) is on the opposite side of the USB Icon. The specification further states "The USB Icon is also located adjacent to each receptacle. Receptacles should be oriented to allow the Icon on the plug to be visible during the mating process." However, the specification does not consider the height of the device compared to the eye level height of the user, so the side of the cable that is "visible" when mated to a computer on a desk can depend on whether the user is standing or kneeling. Although published eight years later, the USB 3.0 specification has similar wording, stating only "USB 3.0 receptacles should be oriented to allow the Icon on the plug to be visible during the mating process."[22]

- Only moderate insertion/removal force is needed (by specification). USB cables and small USB devices are held in place by the gripping force from the receptacle (without need of the screws, clips, or thumbturns other connectors have required). The force needed to make or break a connection is modest, allowing connections to be made in awkward circumstances (i.e., behind a floor-mounted chassis, or from below) or by those with motor disabilities. This has the disadvantage of easily and unintentionally breaking connections that one has intended to
be permanent in case of cable accident (e.g., tripping, or inadvertent tugging). Conversely though, this prevents damage to the socket or the device which it is plugged into (such as pulling it off a shelf) in the case of a serious jerk.

- The standard connectors were deliberately intended to enforce the directed topology of a USB network: type A connectors on host devices that supply power and type B connectors on target devices that receive power. This prevents users from accidentally connecting two USB power supplies to each other, which could lead to dangerously high currents, circuit failures, or even fire. USB does not support cyclical networks and the standard connectors from incompatible USB devices are themselves incompatible. Unlike other communications systems (e.g. network cabling) gender changers make little sense with USB and are almost never used, though cables with 2 standard type A plugs are commonly found in North American dollar stores.

**Durability**

- The standard connectors were designed to be robust. Many previous connector designs were fragile, specifying embedded component pins or other delicate parts which proved vulnerable to bending or breakage, even with the application of modest force. The electrical contacts in a USB connector are protected by an adjacent plastic tongue, and the entire connecting assembly is usually protected by an enclosing metal sheath. Although meant to reinforce the metal connections, the construction and materials used in the plastic tongue can actually decrease the life of the connector, as some manufacturers use hard, brittle plastics that easily break when bent.

- The connector construction always ensures that the external sheath on the plug makes contact with its counterpart in the receptacle before any of the four connectors within make electrical contact. The external metallic sheath is typically connected to system ground, thus dissipating damaging static charges. This enclosure design also provides a (moderate) degree of protection from electromagnetic interference to the USB signal while it travels through the mated connector pair (the only location when the otherwise twisted data pair travels in parallel). In addition, because of the required sizes of the power and common connections, they are made after the system ground but before the data connections. This type of staged make-break timing allows for electrically safe hot-swapping, a common practice in the design of connectors in the aerospace industry.

- The newer Micro-USB receptacles are designed for up to 10,000 cycles of insertion and removal between the receptacle and plug, compared to 1500 for the standard USB and 5000 for the Mini-USB receptacle. This is accomplished by adding a locking device and by moving the leaf-spring connector from the jack to the plug, so that the most-stressed part is on the cable side of the connection. This change was made so that the connector on the less expensive cable would bear the most wear instead of the more expensive micro-USB device.
Compatibility

- The USB standard specifies relatively loose tolerances for compliant USB connectors to minimize physical incompatibilities in connectors from different vendors. To address a weakness present in some other connector standards, the USB specification also defines limits to the size of a connecting device in the area around its plug. This was done to prevent a device from blocking adjacent ports due to the size of the cable strain relief mechanism (usually molding integral with the cable outer insulation) at the connector. Compliant devices must either fit within the size restrictions or support a compliant extension cable which does.

- Two-way communication is also possible. In USB 3.0, full-duplex communications are done when using SuperSpeed (USB 3.0) transfer. In previous USB versions (i.e., 1.0 or 2.0), all communication is half-duplex and directionally controlled by the host.

In general, cables have only plugs (very few have a receptacle on one end, although extension cables with a standard A plug and jack are sold), and hosts and devices have only receptacles. Hosts almost universally have type-A receptacles, and devices one or another type-B variety. Type-A plugs mate only with type-A receptacles, and type-B with type-B; they are deliberately physically incompatible. However, an extension to USB standard specification called USB On-The-Go allows a single port to act as either a host or a device—chosen by which end of the cable plugs into the receptacle on the unit. Even after the cable is hooked up and the units are talking, the two units may "swap" ends under program control. This capability is meant for units such as PDAs in which the USB link might connect to a PC's host port as a device in one instance, yet connect as a host itself to a keyboard and mouse device in another instance.

- USB 3.0 receptacles are electrically compatible with USB 2.0 device plugs if they can physically match. Most combinations will work, but there are a few physical incompatibilities. However, only USB 3.0 Standard-A receptacles can accept USB 3.0 Standard-A device plugs.

- eSATAp (eSATA/USB) port is also compatible with USB 2.0 devices

Host Interface receptacles (USB 1.x/2.0)

The following receptacles accept the following plugs.

<table>
<thead>
<tr>
<th>Receptacle</th>
<th>Plug</th>
</tr>
</thead>
<tbody>
<tr>
<td>Standard-A</td>
<td>Yes</td>
</tr>
<tr>
<td>Standard-B</td>
<td>No</td>
</tr>
<tr>
<td>Mini-B</td>
<td>No</td>
</tr>
<tr>
<td>Micro-AB</td>
<td>No</td>
</tr>
<tr>
<td>Micro-B</td>
<td>No</td>
</tr>
</tbody>
</table>
Universal Serial Bus

Cable plugs (USB 1.x/2.0)

Cables exist with pairs of plugs as indicated in the following table.

<table>
<thead>
<tr>
<th>Plug</th>
<th>Micro-B</th>
<th>Micro-A</th>
<th>Mini-B</th>
<th>Standard-B</th>
<th>Standard-A</th>
</tr>
</thead>
<tbody>
<tr>
<td>Standard-A</td>
<td>Yes</td>
<td>NS</td>
<td>Yes</td>
<td>Yes</td>
<td>NS</td>
</tr>
<tr>
<td>Standard-B</td>
<td>No</td>
<td>NS</td>
<td>No</td>
<td>No</td>
<td></td>
</tr>
<tr>
<td>Mini-B</td>
<td>No</td>
<td>NS</td>
<td>No</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Micro-A</td>
<td>Yes</td>
<td>No</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Micro-B</td>
<td>No</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

NS: non-standard, existing for specific proprietary purposes, and not interoperable with USB-IF compliant equipment.

In addition to the above cable assemblies comprising two plugs, an "adapter" cable with a Micro-A plug and a Standard-A receptacle is compliant with USB specifications.[20] Other combinations of connectors are not compliant. However, some older devices and cables with Mini-A connectors have been certified by USB-IF. The Mini-A connector has been deprecated: there will be no new certification of assemblies using Mini-A connector.[23]

Connector types

There are several types of USB connectors, including some that have been added while the specification progressed. The original USB specification detailed Standard-A and Standard-B plugs and receptacles. The first engineering change notice to the USB 2.0 specification added Mini-B plugs and receptacles.

The data connectors in the Standard-A plug are actually recessed in the plug as compared to the outside power connectors. This permits the power to connect first which prevents data errors by allowing the device to power up first and then transfer the data. Some devices will operate in different modes depending on whether the data connection is made. This difference in connection can be exploited by inserting the connector only partially. For example, some battery-powered MP3 players switch into file transfer mode and cannot play MP3 files while a USB plug is fully inserted, but can be operated in MP3 playback mode using USB power by inserting the plug only part way so that the power slots make contact while the data slots do not. This enables those devices to be operated in MP3 playback mode while getting power from the cable.

To reliably enable a charge-only feature, modern USB accessory peripherals now include charging cables that provide power connections to the host port but no data connections, and both home and vehicle charging docks are available that supply power from a converter device and do not include a host device and data pins, allowing any capable USB device to be charged and/or operated from a standard USB cable.
USB standard connectors

Standard type A

The Standard-A type of USB plug is a flattened rectangle which inserts into a "downstream-port" receptacle on the USB host, or a hub, and carries both power and data. This plug is frequently seen on cables that are permanently attached to a device, such as one connecting a keyboard or mouse to the computer via USB connection.

USB connections eventually wear out as the connection loosens through repeated plugging and unplugging. The lifetime of a USB-A male connector is approximately 1,500 connect/disconnect cycles.[24]

There are female-female connectors.

Standard type B

A Standard-B plug—which has a square shape with bevelled exterior corners—typically plugs into an "upstream receptacle" on a device that uses a removable cable, e.g. a printer. A Type B plug delivers power in addition to carrying data. On some devices, the Type B receptacle has no data connections, being used solely for accepting power from the upstream device. This two-connector-type scheme (A/B) prevents a user from accidentally creating an electrical loop.[25]

USB Mini and Micro connectors

Various connectors have been used for smaller devices such as PDAs, mobile phones or digital cameras. These include the now-deprecated[23] (but standardized) Mini-A and the currently standard Mini-B,[26] Micro-A, and Micro-B connectors. The Mini-A and Mini-B plugs are approximately 3 by 7 mm.

The micro-USB plugs have a similar width but approximately half the thickness, enabling their integration into thinner portable devices. The micro-A connector is 6.85 by 1.8 mm with a maximum overmold size of 11.7 by 8.5 mm. The micro-B connector is 6.85 by 1.8 mm with a maximum overmold size of 10.6 by 8.5 mm.[20]

The Micro-USB connector was announced by the USB-IF on .[27] The Mini-A connector and the Mini-AB receptacle connector were deprecated on .[28] As of February 2009, many currently available devices and cables still use Mini plugs, but the newer Micro connectors are being widely adopted and as of December 2010, the Micro connectors are the most widely used. The thinner micro connectors are
intended to replace the Mini plugs in new devices including smartphones and personal digital assistants. The Micro plug design is rated for at least 10,000 connect-disconnect cycles which is significantly more than the Mini plug design.[29] The *Universal Serial Bus Micro-USB Cables and Connectors Specification* [29] details the mechanical characteristics of Micro-A plugs, Micro-AB receptacles, and Micro-B plugs and receptacles, along with a Standard-A receptacle to Micro-A plug adapter.

The cellular phone carrier group, Open Mobile Terminal Platform (OMTP) in 2007 have endorsed Micro-USB as the standard connector for data and power on mobile devices.[30] These include various types of battery chargers, allowing Micro-USB to be the single external cable link needed by some devices.

As of Micro-USB has been accepted and is being used by almost all cell phone manufacturers as the standard charging port (including HTC, Motorola, Nokia, LG, Hewlett-Packard, Samsung, Sony Ericsson, Research In Motion) in most of the world.

On, following a request from the European Commission and in close co-operation with the Commission services, major producers of mobile phones have agreed in a Memorandum of Understanding ("MoU") to harmonise chargers for data-enabled mobile phones sold in the European Union. Industry commits to provide charger compatibility on the basis of the Micro-USB connector. Consumers will be able to purchase mobile phones without a charger, thus logically reducing their cost.[31] Following a mandate from the European Commission, the European Standardisation Bodies CEN-CENELEC and ETSI have now made available the harmonised standards needed for the manufacture of data-enabled mobile phones compatible with the new common External Power Supply (EPS) based on micro-USB.[32]

In addition, on the International Telecommunication Union (ITU) has also announced that it had embraced micro-USB as the *Universal Charger Solution* its "energy-efficient one-charger-fits-all new mobile phone solution", and added: "Based on the Micro-USB interface, UCS chargers will also include a 4-star or higher efficiency rating—up to three times more energy-efficient than an unrated charger."[33]

**USB Micro-AB Socket OTG**

An OTG device is required to have one, and only one USB connector: a Micro-AB receptacle as defined in [Micro-USB1.01]. This receptacle is capable of accepting either a Micro-A plug or a Micro-B plug attached to any of the legal cables and adapters defined in [Micro-USB1.01].

The OTG device with the A-plug inserted is called the A-device and is responsible for powering the USB interface when required and by default assumes the role of host. The OTG device with the B-plug inserted is called the B-device and by default assumes the role of peripheral. An OTG device with no plug inserted defaults to acting as a B-device. If an application on the B-device requires the role of host, then the HNP protocol is used to temporarily transfer the host role to the B-device.

OTG devices attached either to a peripheral-only B-device or a standard/embedded host will have their role fixed by the cable since in these scenarios it is only possible to attach the cable one way around.
Proprietary connectors and formats

- Microsoft's original Xbox game console uses standard USB 1.1 signalling in its controllers and memory cards, but uses proprietary connectors and ports.
- IBM UltraPort uses standard USB signalling, but via a proprietary connection format.
- American Power Conversion uses USB signalling and HID device class on its uninterruptible power supplies using 10P10C connectors.

HTC manufactured Windows Mobile and Android-based Communicators which have a proprietary connector called HTC ExtUSB (Extended USB). ExtUSB combines mini-USB (with which it is backwards-compatible) with audio input as well as audio and video output in an 11-pin connector.

Nokia includes a USB connection as part of the Pop-Port connector on some older mobile phone models.
- Sony Ericsson used a proprietary connector called FastPort from 2005 to 2009.
- The second, third, and fourth generation iPod Shuffle uses a TRRS connector to carry USB, audio, or power signals.
- iriver added a fifth power pin within USB-A plugs for higher power and faster charging, used for the iriver U10 series. A mini-USB version contains a matching extra power pin for the cradle.
- Apple has shipped non-standard USB extension cables with some of their computers, for use with the included Apple USB keyboards. The extension cable's socket is keyed with a small protrusion to prevent the insertion of a standard USB plug, while the Apple USB keyboard's plug has a matching indentation. The indentation on the keyboard's plug does not interfere with insertion into a standard USB socket. Despite the keying, it is still possible to insert standard USB plugs into the extension cord. The protrusion can also be shaved off with an appropriate blade, or crushed with locking pliers.
- Apple also uses a proprietary USB 30-pin dock connector on its iPods, iPhones, and the iPad.
- HP Tablet computers use non-standard connectors to transmit the USB signals between the keyboard/mouse unit and the Computer Tablet Unit.
- PDMI (Portable Digital Media Interface) is a 30-pin docking connector for portable devices standardized by ANSI/CEA which includes USB 3.0 "SuperSpeed" and USB 2.0 "High/Standard Speed" with USB-on-the-go, as well as DisplayPort, HDMI CEC, 5 V power, and analog audio.
- Some digital cameras have their own USB connectors, like the Panasonic Lumix DMC-FT2.
- The United States Army's Land Warrior system uses standard USB signalling with 15.6 V power using a ruggedized Glenair connector.
- The ExpressCard interface includes a USB2 port as well as the express bus port.
Cable Properties

Conductor Configuration
The data cables for USB 1.x and USB 2.x use a twisted pair to reduce noise and crosstalk. They are arranged much as in the diagram below. USB 3.0 cables are more complex and employ shielding for some of the added data lines (2 pairs); a shield is added around the pair sketched.

Maximum Cable Length
The maximum length of a standard USB cable (for USB 2.0 or earlier) is 5 metres (16.4 ft).[34] The primary reason for this limit is the maximum allowed round-trip delay of about 1,500 ns. If USB host commands are unanswered by the USB device within the allowed time, the host considers the command lost. When adding USB device response time, delays from the maximum number of hubs added to the delays from connecting cables, the maximum acceptable delay per cable amounts to 26 ns.[34] The USB 2.0 specification requires cable delay to be less than 5.2 ns per meter (192,000 km/s, which is close to the maximum achievable transmission speed for standard copper cable).[35] This allows for a five meter cable. The USB 3.0 standard does not directly specify a maximum cable length, requiring only that all cables meet an electrical specification. For copper wire cabling, some calculations have suggested a maximum length of perhaps 3 m.

Power
The USB 1.x and 2.0 specifications provide a 5 V supply on a single wire from which connected USB devices may draw power. The specification provides for no more than 5.25 V and no less than 4.75 V (5 V±5%) between the positive and negative bus power lines. For USB 2.0 the voltage supplied by low-powered hub ports is 4.4 V to 5.25 V.[36]

A unit load is defined as 100 mA in USB 2.0, and was raised to 150 mA in USB 3.0. A maximum of 5 unit loads (500 mA) can be drawn from a port in USB 2.0, which was raised to 6 (900 mA) in USB 3.0. There are two types of devices: low-power and high-power. Low-power devices draw at most 1 unit load, with minimum operating voltage of 4.4 V in USB 2.0, and 4 V in USB 3.0. High-power devices draw the maximum number of unit loads supported by the standard. All devices default as low-power but the device's software may request high-power as long as the power is available on the providing bus.[37]

Some devices like high-speed external disk drives may require more than 500 mA of current[38] and therefore cannot be powered from one USB 2.0 port. Such devices usually come with Y-shaped cable that has two USB connectors to be inserted into a computer. With such a cable a device can draw power from two USB ports simultaneously.[39]

A bus-powered hub is initialized at 1 unit load and transitions to maximum unit loads after hub configuration is obtained. Any device connected to the hub will draw 1 unit load regardless of the current draw of devices connected to other ports of the hub (i.e. one device connected on a four-port hub will only draw 1 unit load despite the fact that all unit loads are being supplied to the hub).[37]

A self-powered hub will supply maximum supported unit loads to any device connected to it. A battery-powered hub may supply maximum unit loads to ports. In addition, the $V_{BUS}$ will supply 1 unit load upstream for communication if parts of the Hub are powered down.[37]

In Battery Charging Specification,[40] new powering modes are added to the USB specification. A host or hub Charging Downstream Port can supply a maximum of 1.5 A when communicating at low-bandwidth or full-bandwidth, a maximum of 900 mA when communicating at high-bandwidth, and as much current as the connector will safely handle when no communication is taking place; USB 2.0 standard-A connectors are rated at
1500 mA by default. A Dedicated Charging Port can supply a maximum of 1.8 A of current at 5.25 V. A portable device can draw up to 1.8 A from a Dedicated Charging Port. The Dedicated Charging Port shorts the D+ and D- pins with a resistance of at most 200 Ω. The short disables data transfer, but allows devices to detect the Dedicated Charging Port and allows very simple, high current chargers to be manufactured. The increased current (faster, 9 W charging) will occur once both the host/hub and devices support the new charging specification.

### Sleep and Charge

Sleep-and-charge USB ports can be used to charge electronic devices even when the computer is switched off. Normally when a computer is powered off the USB ports are powered down. This prevents phones and other devices from being able to charge unless the computer is powered on. Sleep-and-charge USB ports remain powered even when the computer is off. On laptops charging devices from the USB port when it is not being powered from AC will drain the laptop battery faster. Desktop machines need to remain plugged into AC power for Sleep-and-charge to work.\[41\]

### Mobile device charger standards

As of, all new mobile phones applying for a license in China are required to use the USB port as a power port.\[42\]\[43\] This was the first standard to use the convention of shorting D+ and D-.\[44\]

In September 2007, the Open Mobile Terminal Platform group—a forum of mobile network operators and manufacturers such as Nokia, Samsung, Motorola, Sony Ericsson and LG—announced that its members had agreed on micro-USB as the future common connector for mobile devices.\[45\]\[46\]

On, the GSM Association (GSMA) announced\[47\] that they had agreed on a standard charger for mobile phones. The standard connector to be adopted by 17 manufacturers including Nokia, Motorola and Samsung is to be the micro-USB connector (several media reports erroneously reported this as the mini-USB). The new chargers will be much more efficient than existing chargers.\[47\] Having a standard charger for all phones means that manufacturers will no longer have to supply a charger with every new phone. The basis of the GSMA's Universal Charger Solution (UCS) is the technical recommendation from OMTP and the USB-IF battery charging standard.\[48\]\[49\]\[50\]

On, this was further endorsed by the CTIA – The Wireless Association.\[51\]

In June 2009, many of the world's largest mobile phone manufacturers signed a Memorandum of Understanding (MoU), agreeing to make most data-enabled mobile phones marketed in the European Union compatible with a common External Power Supply (EPS) based on the GSMA / OMTP Universal Charging Solution.\[52\]\[53\]

On the International Telecommunication Union (ITU) announced that it had embraced the Universal Charger Solution as its "energy-efficient one-charger-fits-all new mobile phone solution", and added: "Based on the Micro-USB interface, UCS chargers will also include a 4-star or higher efficiency rating—up to three times more energy-efficient than an unrated charger."\[54\]
**Non-standard devices**

Some USB devices require more power than is permitted by the specifications for a single port. This is common for external hard and optical disc drives, and generally for devices with motors or lamps. Such devices can use an external power supply, which is allowed by the standard, or use a dual-input USB cable, one input of which is used for power and data transfer, the other solely for power, which makes the device a non-standard USB device. Some external hubs may, in practice, supply more power to USB devices than required by the specification but a standard-compliant device may not depend on this.

Some non-standard USB devices use the 5 V power supply without participating in a proper USB network which negotiates power draws with the host interface. These are usually referred to as USB decorations. The typical example is a USB-powered keyboard light; fans, mug coolers and heaters, battery chargers, miniature vacuum cleaners, and even miniature lava lamps are available. In most cases, these items contain no digital circuitry, and thus are not Standard compliant USB devices at all. This can theoretically cause problems with some computers, such as drawing too much current and hurting circuitry; prior to the Battery Charging Specification, the USB specification required that devices connect in a low-power mode (100 mA maximum) and communicate their current requirements to the host, which would then permit the device to switch into high-power mode.

In addition to limiting the total average power used by the device, the USB specification limits the inrush current (i.e., that used to charge decoupling and filter capacitors) when the device is first connected. Otherwise, connecting a device could cause problems with the host’s internal power. Also, USB devices are required to automatically enter ultra low-power suspend mode when the USB host is suspended. Nevertheless, many USB host interfaces do not cut off the power supply to USB devices when they are suspended since resuming from the suspended state would become a lot more complicated if they did.

There are also devices at the host end that do not support negotiation, such as battery packs that can power USB-powered devices; some provide power, while others pass through the data lines to a host PC. USB power adapters convert utility power and/or another power source (e.g., a car’s electrical system) to run attached devices. Some of these devices can supply up to 1 A of current. Without negotiation, the powered USB device is unable to inquire if it is allowed to draw 100 mA, 500 mA, or 1 A.

**Powered USB**

Powered USB uses standard USB signaling with the addition of extra power lines. It uses four additional pins to supply up to 6 A at either 5 V, 12 V, or 24 V (depending on keying) to peripheral devices. The wires and contacts on the USB portion have been upgraded to support higher current on the 5 V line, as well. This is commonly used in retail systems and provides enough power to operate stationary barcode scanners, printers, pin pads, signature capture devices, etc. This modification of the USB interface is proprietary and was developed by IBM, NCR, and FCI/Berg. It is essentially two connectors stacked such that the bottom connector accepts a standard USB plug and the top connector takes a power connector.
Signaling

USB supports the following signaling rates:

- **A low-speed** rate of 1.5 Mbit/s (~183 kB/s) is defined by USB 1.0. It is very similar to "full-bandwidth" operation except each bit takes 8 times as long to transmit. It is intended primarily to save cost in low-bandwidth human interface devices (HID) such as keyboards, mice, and joysticks.
- **The full-speed** rate of 12 Mbit/s (~1.43 MB/s) is the basic USB data rate defined by USB 1.1. All USB hubs support full-bandwidth.
- **A high-speed** (USB 2.0) rate of 480 Mbit/s (~57 MB/s) was introduced in 2001. All hi-speed devices are capable of falling back to full-bandwidth operation if necessary; they are backward compatible. Connectors are identical.
- **A SuperSpeed** (USB 3.0) rate of 4800 Mbit/s (~572 MB/s). The written USB 3.0 specification was released by Intel and partners in August 2008. The first USB 3 controller chips were sampled by NEC May 2009 and products using the 3.0 specification arrived beginning in January 2010. USB 3.0 connectors are generally backwards compatible, but include new wiring and full duplex operation.

USB signals are transmitted on a twisted-pair data cable with 90Ω ±15% Characteristic impedance, labeled D+ and D−. Prior to USB 3.0, these collectively use half-duplex differential signaling to reduce the effects of electromagnetic noise on longer lines. Transmitted signal levels are 0.0–0.3 volts for low and 2.8–3.6 volts for high in full-bandwidth and low-bandwidth modes, and −10–10 mV for low and 360–440 mV for high in hi-bandwidth mode. In FS mode the cable wires are not terminated, but the HS mode has termination of 45 Ω to ground, or 90 Ω differential to match the data cable impedance, reducing interference due to signal reflections. USB 3.0 introduces two additional pairs of shielded twisted wire and new, mostly interoperable contacts in USB 3.0 cables, for them. They permit the higher data rate, and full duplex operation.

A USB connection is always between a host or hub at the "A" connector end, and a device or hub's "upstream" port at the other end. Originally, this was a "B" connector, preventing erroneous loop connections, but additional upstream connectors were specified, and some cable vendors designed and sold cables which permitted erroneous connections (and potential damage to the circuitry). USB interconnections are not as fool-proof or as simple as originally intended.

The host includes 15 kΩ pull-down resistors on each data line. When no device is connected, this pulls both data lines low into the so-called “single-ended zero” state (SE0 in the USB documentation), and indicates a reset or disconnected connection.

A USB device pulls one of the data lines high with a 1.5 kΩ resistor. This overpowers one of the pull-down resistors in the host and leaves the data lines in an idle state called "J". For USB 1.x, the choice of data line indicates a device's bandwidth support; full-bandwidth devices pull D+ high, while low-bandwidth devices pull D− high.

USB data is transmitted by toggling the data lines between the J state and the opposite K state. USB encodes data using the NRZI convention; a 0 bit is transmitted by toggling the data lines from J to K or vice-versa, while a 1 bit is transmitted by leaving the data lines as-is. To ensure a minimum density of signal transitions remains in the bitstream, USB uses bit stuffing; an extra 0 bit is inserted into the data stream after any appearance of six consecutive 1 bits. Seven consecutive received 1 bits is always an error. USB 3.0 has introduced additional data transmission encodings.

A USB packet begins with an 8-bit synchronization sequence '00000001'. That is, after the initial idle state J, the data lines toggle KJKJKJKK. The final 1 bit (repeated K state) marks the end of the sync pattern and the beginning of the USB frame. For high bandwidth USB, the packet begins with a 32-bit synchronization sequence.

A USB packet's end, called EOP (end-of-packet), is indicated by the transmitter driving 2 bit times of SE0 (D+ and D− both below max) and 1 bit time of J state. After this, the transmitter ceases to drive the D+/D− lines and the aforementioned pull up resistors hold it in the J (idle) state. Sometimes skew due to hubs can add as much as one bit time before the SE0 of the end of packet. This extra bit can also result in a "bit stuff violation" if the six bits before it
in the CRC are ‘1’s. This bit should be ignored by receiver.

A USB bus is reset using a prolonged (10 to 20 milliseconds) SE0 signal.

USB 2.0 devices use a special protocol during reset, called "chirping", to negotiate the high bandwidth mode with the host/hub. A device that is HS capable first connects as an FS device (D+ pulled high), but upon receiving a USB RESET (both D+ and D− driven LOW by host for 10 to 20 ms) it pulls the D− line high, known as chirp K. This indicates to the host that the device is high bandwidth. If the host/hub is also HS capable, it chirps (returns alternating J and K states on D− and D+ lines) letting the device know that the hub will operate at high bandwidth. The device has to receive at least 3 sets of KJ chirps before it changes to high bandwidth terminations and begins high bandwidth signaling. Because USB 3.0 uses wiring separate and additional to that used by USB 2.0 and USB 1.x, such bandwidth negotiation is not required.

Clock tolerance is 480.00 Mbit/s ±500 ppm, 12.000 Mbit/s ±2500 ppm, 1.50 Mbit/s ±15000 ppm.

Though high bandwidth devices are commonly referred to as "USB 2.0" and advertised as "up to 480 Mbit/s", not all USB 2.0 devices are high bandwidth. The USB-IF certifies devices and provides licenses to use special marketing logos for either "basic bandwidth" (low and full) or high bandwidth after passing a compliance test and paying a licensing fee. All devices are tested according to the latest specification, so recently compliant low bandwidth devices are also 2.0 devices.

**USB 3**

**USB 3** released in 2008-11-12 uses tinned copper stranded AWG-28 cables with 90 ± 7 Ω impedance for its high-speed differential pairs and linear feedback shift register and 8b/10b encoding sent with a voltage of 1000 mV nominal with a 100 mV receiver threshold, the receiver uses equalization. SSC clock and 300 ppm precision is used. Packet headers are protected with CRC-16, while data payload is protected with CRC-32. Power up to 3.6 W may be used. One unit load in superspeed mode is equal to 150 mA.

**USB 3 Pinout**

<table>
<thead>
<tr>
<th>Pin</th>
<th>Color</th>
<th>Signal name ('A' connector)</th>
<th>Signal name ('B' connector)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Red</td>
<td>VBUS</td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>White</td>
<td>D−</td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>Green</td>
<td>D+</td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>Black</td>
<td>GND</td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>Blue</td>
<td>StdA_SSRX−</td>
<td>StdA_SSTX−</td>
</tr>
<tr>
<td>6</td>
<td>Yellow</td>
<td>StdA_SSRX+</td>
<td>StdA_SSTX+</td>
</tr>
<tr>
<td>7</td>
<td>Shield</td>
<td>GND_DRAIN</td>
<td></td>
</tr>
<tr>
<td>8</td>
<td>Purple</td>
<td>StdA_SSTX−</td>
<td>StdA_SSRX−</td>
</tr>
<tr>
<td>9</td>
<td>Orange</td>
<td>StdA_SSTX+</td>
<td>StdA_SSRX+</td>
</tr>
<tr>
<td>Shell</td>
<td>Shell</td>
<td>Shield</td>
<td></td>
</tr>
</tbody>
</table>

[61]
Transfer rates

USB 2.0 data rates

The theoretical maximum data rate in USB 2.0 is 480 Mbit/s (60 MB/s) per controller and is shared amongst all attached devices. Some chipset manufacturers overcome this bottleneck by providing multiple USB 2.0 controllers within the southbridge. Big performance gains can be achieved when attaching multiple high bandwidth USB devices such as disk enclosures in different controllers. The following table displays southbridge ICs that have multiple EHCI controllers.

<table>
<thead>
<tr>
<th>Vendor</th>
<th>Southbridge</th>
<th>USB 2.0 ports</th>
<th>EHCI controllers</th>
<th>Maximum data rate</th>
</tr>
</thead>
<tbody>
<tr>
<td>AMD</td>
<td>SB7x0/SP5100</td>
<td>12</td>
<td>2</td>
<td>120 MB/s</td>
</tr>
<tr>
<td>AMD</td>
<td>SB8x0</td>
<td>14</td>
<td>3</td>
<td>180 MB/s</td>
</tr>
<tr>
<td>Broadcom</td>
<td>HT1100</td>
<td>12</td>
<td>3</td>
<td>180 MB/s</td>
</tr>
<tr>
<td>Intel</td>
<td>ICH8</td>
<td>10</td>
<td>2</td>
<td>120 MB/s</td>
</tr>
<tr>
<td>Intel</td>
<td>ICH9</td>
<td>12</td>
<td>2</td>
<td>120 MB/s</td>
</tr>
<tr>
<td>Intel</td>
<td>ICH10</td>
<td>12</td>
<td>2</td>
<td>120 MB/s</td>
</tr>
<tr>
<td>Intel</td>
<td>PCH</td>
<td>8/12/14</td>
<td>2</td>
<td>120 MB/s</td>
</tr>
<tr>
<td>nVIDIA</td>
<td>ION/ION-LE</td>
<td>6</td>
<td>2</td>
<td>120 MB/s</td>
</tr>
</tbody>
</table>

Every other AMD, Broadcom, and Intel southbridge supporting USB 2.0 has only one EHCI controller. All SiS southbridge, ULi, and VIA southbridge, single chip northbridge/southbridge supporting USB 2.0 have only one EHCI controller. Also all PCI USB 2.0 ICs used for add-in cards have only one EHCI controller. In PCIe, the usual design with multiple USB ports per EHCI controller has changed with the introduction of the MosChip MCS9990 IC. MCS9990 has one EHCI controller per port so all its USB ports can operate simultaneously without any performance limitations. Cards with dual ICs have been introduced as well, with two PCI USB 2.0 ICs attached to one PCI to PCIe bridge.

Transfer speeds in practice

As of 2004, the actual throughput of USB 2.0 high bandwidth attained with a hard drive tested on a Mac was about 18 MiB/s, 30% of the maximum theoretical bulk data transfer rate of 60 MB/s (57.2 MiB/s or 480 Mbit/s). On Windows, the highest speed observed was 33 MiB/s, or 55% of the theoretical max. The drive could reach 58 MiB/s on Firewire, so the drive's speed was not a limiting factor.\[^{62}\]

According to a USB-IF chairman, "at least 10 to 15 percent of the stated peak 60 MB/s (480 Mbit/s) of Hi-Speed USB goes to overhead — the communication protocol between the card and the peripheral. Overhead is a component of all connectivity standards."\[^{63}\]

Tables illustrating the transfer limits are shown in Chapter 5 of the USB spec.

Typical high bandwidth USB devices operate at lower data rates, often about 3 MiB/s overall, sometimes up to 10–20 MiB/s. A Full Speed device (for example PIC18f2550) easily reaches a little over 1MiB/s over a bulk endpoint.

For isochronous devices like audio streams, the bandwidth is constant, and reserved exclusively for a given device. The bus bandwidth therefore only has an effect on the number of channels that can be sent at a time, not the "speed" or latency of the transmission.
Data packets

USB communication takes the form of packets. Initially, all packets are sent from the host, via the root hub and possibly more hubs, to devices. Some of those packets direct a device to send some packets in reply.

After the sync field, all packets are made of 8-bit bytes, transmitted least-significant bit first. The first byte is a packet identifier (PID) byte. The PID is actually 4 bits; the byte consists of the 4-bit PID followed by its bitwise complement. This redundancy helps detect errors. (Note also that a PID byte contains at most four consecutive 1 bits, and thus will never need bit-stuffing, even when combined with the final 1 bit in the sync byte. However, trailing 1 bits in the PID may require bit-stuffing within the first few bits of the payload.)

**USB PID bytes**

<table>
<thead>
<tr>
<th>Type</th>
<th>PID value (msb-first)</th>
<th>Transmitted byte (lsb-first)</th>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Reserved</td>
<td>0000</td>
<td>0000 1111</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Token</td>
<td>1000</td>
<td>0001 1110</td>
<td>SPLIT</td>
<td>High-bandwidth (USB 2.0) split transaction</td>
</tr>
<tr>
<td></td>
<td>0100</td>
<td>0010 1101</td>
<td>PING</td>
<td>Check if endpoint can accept data (USB 2.0)</td>
</tr>
<tr>
<td>Special</td>
<td>1100</td>
<td>0011 1100</td>
<td>PRE</td>
<td>Low-bandwidth USB preamble</td>
</tr>
<tr>
<td>Handshake</td>
<td></td>
<td></td>
<td>ERR</td>
<td>Split transaction error (USB 2.0)</td>
</tr>
<tr>
<td></td>
<td>0010</td>
<td>0110 1010</td>
<td>ACK</td>
<td>Data packet accepted</td>
</tr>
<tr>
<td></td>
<td>1010</td>
<td>1001 1010</td>
<td>NAK</td>
<td>Data packet not accepted; please retransmit</td>
</tr>
<tr>
<td></td>
<td>0110</td>
<td>0111 1000</td>
<td>NYET</td>
<td>Data not ready yet (USB 2.0)</td>
</tr>
<tr>
<td></td>
<td>1110</td>
<td>0111 1000</td>
<td>STALL</td>
<td>Transfer impossible; do error recovery</td>
</tr>
<tr>
<td>Token</td>
<td>0001</td>
<td>1000 0111</td>
<td>OUT</td>
<td>Address for host-to-device transfer</td>
</tr>
<tr>
<td></td>
<td>1001</td>
<td>1001 0110</td>
<td>IN</td>
<td>Address for device-to-host transfer</td>
</tr>
<tr>
<td></td>
<td>0101</td>
<td>1010 0101</td>
<td>SOF</td>
<td>Start of frame marker (sent each ms)</td>
</tr>
<tr>
<td></td>
<td>1101</td>
<td>1101 0100</td>
<td>SETUP</td>
<td>Address for host-to-device control transfer</td>
</tr>
<tr>
<td>Data</td>
<td>0111</td>
<td>1100 0011</td>
<td>DATA0</td>
<td>Even-numbered data packet</td>
</tr>
<tr>
<td></td>
<td>1011</td>
<td>1101 0010</td>
<td>DATA1</td>
<td>Odd-numbered data packet</td>
</tr>
<tr>
<td></td>
<td>0111</td>
<td>1110 0001</td>
<td>DATA2</td>
<td>Data packet for high-bandwidth isochronous transfer (USB 2.0)</td>
</tr>
<tr>
<td></td>
<td>1111</td>
<td>1111 0000</td>
<td>MDATA</td>
<td>Data packet for high-bandwidth isochronous transfer (USB 2.0)</td>
</tr>
</tbody>
</table>

Packets come in three basic types, each with a different format and CRC (cyclic redundancy check):

**Handshake packets**

Handshake packets consist of nothing but a PID byte, and are generally sent in response to data packets. The three basic types are **ACK**, indicating that data was successfully received, **NAK**, indicating that the data cannot be received at this time and should be retried, and **STALL**, indicating that the device has an error and will never be able to successfully transfer data until some corrective action (such as device initialization) is performed.

USB 2.0 added two additional handshake packets, **NYET** which indicates that a split transaction is not yet complete. A NYET packet is also used to tell the host that the receiver has accepted a data packet, but cannot accept any more due to buffers being full. The host will then send PING packets and will continue with data packets once the device ACK’s the PING. The other packet added was the **ERR** handshake to indicate that a split transaction failed.
The only handshake packet the USB host may generate is ACK; if it is not ready to receive data, it should not instruct a device to send any.

**Token packets**

Token packets consist of a PID byte followed by 2 payload bytes: 11 bits of address and a 5-bit CRC. Tokens are only sent by the host, never a device.

*IN* and *OUT* tokens contain a 7-bit device number and 4-bit function number (for multifunction devices) and command the device to transmit DATAx packets, or receive the following DATAx packets, respectively.

An IN token expects a response from a device. The response may be a NAK or STALL response, or a DATAx frame. In the latter case, the host issues an ACK handshake if appropriate.

An OUT token is followed immediately by a DATAx frame. The device responds with ACK, NAK, NYET, or STALL, as appropriate.

*SETUP* operates much like an OUT token, but is used for initial device setup. It is followed by an 8-byte DATA0 frame with a standardized format.

Every millisecond (12000 full-bandwidth bit times), the USB host transmits a special *SOF* (start of frame) token, containing an 11-bit incrementing frame number in place of a device address. This is used to synchronize isochronous data flows. High-bandwidth USB 2.0 devices receive 7 additional duplicate SOF tokens per frame, each introducing a 125 µs "microframe" (60000 high-bandwidth bit times each).

USB 2.0 added a *PING* token, which asks a device if it is ready to receive an OUT/DATA packet pair. The device responds with ACK, NAK, or STALL, as appropriate. This avoids the need to send the DATA packet if the device knows that it will just respond with NAK.

USB 2.0 also added a larger 3-byte *SPLIT* token with a 7-bit hub number, 12 bits of control flags, and a 5-bit CRC. This is used to perform split transactions. Rather than tie up the high-bandwidth USB bus sending data to a slower USB device, the nearest high-bandwidth capable hub receives a SPLIT token followed by one or two USB packets at high bandwidth, performs the data transfer at full or low bandwidth, and provides the response at high bandwidth when prompted by a second SPLIT token. The details are complex; see the USB specification.

**Data packets**

A data packet consists of the PID followed by 0–1023 bytes of data payload (up to 1024 in high bandwidth, at most 8 at low bandwidth), and a 16-bit CRC.

There are two basic data packets, *DATA0* and *DATA1*. They must always be preceded by an address token, and are usually followed by a handshake token from the receiver back to the transmitter. The two packet types provide the 1-bit sequence number required by Stop-and-wait ARQ. If a USB host does not receive a response (such as an ACK) for data it has transmitted, it does not know if the data was received or not; the data might have been lost in transit, or it might have been received but the handshake response was lost.

To solve this problem, the device keeps track of the type of DATAx packet it last accepted. If it receives another DATAx packet of the same type, it is acknowledged but ignored as a duplicate. Only a DATAx packet of the opposite type is actually received.

When a device is reset with a SETUP packet, it expects an 8-byte DATA0 packet next.

USB 2.0 added *DATA2* and *MDATA* packet types as well. They are used only by high-bandwidth devices doing high-bandwidth isochronous transfers which need to transfer more than 1024 bytes per 125 µs "microframe" (8192 kB/s).
PRE "packet"

Low-bandwidth devices are supported with a special PID value, PRE. This marks the beginning of a low-bandwidth packet, and is used by hubs which normally do not send full-bandwidth packets to low-bandwidth devices. Since all PID bytes include four 0 bits, they leave the bus in the full-bandwidth K state, which is the same as the low-bandwidth J state. It is followed by a brief pause during which hubs enable their low-bandwidth outputs, already idling in the J state, then a low-bandwidth packet follows, beginning with a sync sequence and PID byte, and ending with a brief period of SE0. Full-bandwidth devices other than hubs can simply ignore the PRE packet and its low-bandwidth contents, until the final SE0 indicates that a new packet follows.

Protocol analyzers

Due to the complexities of the USB protocol, USB protocol analyzers are invaluable tools to USB device developers. USB analyzers are able to capture the data on USB and display information from low-level bus states to high-level data packets and class-level information.

Comparisons with other connection methods

FireWire

USB was originally seen as a complement to FireWire (IEEE 1394), which was designed as a high-bandwidth serial bus which could efficiently interconnect peripherals such as hard disks, audio interfaces, and video equipment. USB originally operated at a far lower data rate and used much simpler hardware, and was suitable for small peripherals such as keyboards and mice.

The most significant technical differences between FireWire and USB include the following:

• USB networks use a tiered-star topology, while FireWire networks use a tree topology.
• USB 1.0, 1.1 and 2.0 use a “speak-when-spoken-to” protocol; peripherals cannot communicate with the host unless the host specifically requests communication. USB 3.0 is planned to allow for device-initiated communications towards the host (see USB 3.0 below). A FireWire device can communicate with any other node at any time, subject to network conditions.
• A USB network relies on a single host at the top of the tree to control the network. In a FireWire network, any capable node can control the network.
• USB runs with a 5 V power line, while Firewire in current implementations supplies 12 V and theoretically can supply up to 30 V.
• Standard USB hub ports can provide from the typical 500 mA [2.5 W] of current, only 100 mA from non-hub ports. USB 3.0 and USB On-The-Go supply 1800 mA [9.0 W] (for dedicated battery charging, 1500 mA [7.5 W] Full bandwidth or 900 mA [4.5 W] High Bandwidth), while FireWire can in theory supply up to 60 watts of power, although 10 to 20 watts is more typical.

These and other differences reflect the differing design goals of the two buses: USB was designed for simplicity and low cost, while FireWire was designed for high performance, particularly in time-sensitive applications such as audio and video. Although similar in theoretical maximum transfer rate, FireWire 400 is faster than USB 2.0 Hi-Bandwidth in real-use, especially in high-bandwidth use such as external hard-drives. The newer FireWire 800 standard is twice as fast as FireWire 400 and faster than USB 2.0 Hi-Bandwidth both theoretically and practically. The chipset and drivers used to implement USB and Firewire have a crucial impact on how much of the bandwidth prescribed by the specification is achieved in the real world, along with compatibility with peripherals.
Ethernet

The IEEE 802.3af Power over Ethernet (PoE) standard has a more elaborate power negotiation scheme than powered USB. It operates at 48 V DC and can supply more power (up to 12.95 W, PoE+ 25.5 W) over a cable up to 100 meters compared to USB 2.0 which provide 2.5 W with a maximum cable length of 5 meters. This has made PoE popular for VoIP telephones, security cameras, wireless access points and other networked devices within buildings. However, USB is cheaper than PoE provided that the distance is short, and power demand is low.

Ethernet standards requires electrical isolation between the networked device (computer, phone, etc.) and the network cable up to 1500 V AC or 2250 V DC for 60 seconds. USB has no such requirement as it was designed for peripherals closely associated with a host computer, and in fact it connects the peripheral and host grounds. This gives Ethernet a significant safety advantage over USB with peripherals such as cable and DSL modems connected to external wiring that can assume hazardous voltages under certain fault conditions.

Digital musical instruments

Digital musical instruments are another example of where USB is competitive for low-cost devices. However Power over Ethernet and the MIDI plug standard are preferred in high-end devices that must work with long cables. USB can cause ground loop problems in equipment because it connects the ground wires on both transceivers. By contrast, the MIDI plug standard and Ethernet have built-in isolation to 500 V or more.

eSATA / eSATAp

The eSATA connector is a more robust SATA connector, intended for connection to external hard drives and SSDs. It has a far higher transfer rate (3 Gbit/s or 6 Gbit/s, bi-directional) than USB 2.0. A device connected by eSATA appears as an ordinary SATA device, giving both full performance and full compatibility associated with internal drives.

eSATA does not supply power to external devices. This is an increasing disadvantage compared to USB. Even though USB's 2.5 W is sometimes insufficient to power external hard drives, technology is advancing and external drives gradually need less power, manifesting the eSATA disadvantage. eSATAp (power over eSATA) is a new (2009) standard that supplies sufficient power to attached devices using a new, backwards-compatible, connector. On a notebook, eSATAp can supply up to 5v to power up a 2.5” HDD/SSD. On a desktop workstation, it can supply up to 12v to power up larger devices including 3.5” HDD/SSD or 5.25” optical drives.

eSATA, like USB, supports hot plugging, although this might be limited by OS drivers and device firmware.

Version history

Prereleases

- USB 0.7: Released in November 1994.
- USB 0.8: Released in December 1994.
- USB 0.9: Released in April 1995.
- USB 0.99: Released in August 1995.
- USB 1.0 Release Candidate: Released in November 1995.
USB 1.0

• **USB 1.0**: Released in January 1996.
  Specified data rates of 1.5 Mbit/s (Low-Bandwidth) and 12 Mbit/s (Full-Bandwidth). Does not allow for extension cables or pass-through monitors (due to timing and power limitations). Few such devices actually made it to market.

• **USB 1.1**: Released in September 1998.
  Fixed problems identified in 1.0, mostly relating to hubs. Earliest revision to be widely adopted.

USB 2.0

• **USB 2.0**: Released in April 2000.
  Added higher maximum bandwidth of 480 Mbit/s [60 MB/s] (now called “Hi-Speed”). Further modifications to the USB specification have been done via Engineering Change Notices (ECN). The most important of these ECNs are included into the USB 2.0 specification package available from USB.org [73].
  - **Mini-A and Mini-B Connector ECN**: Released in October 2000.
    Specifications for Mini-A and B plug and receptacle. Also receptacle that accepts both plugs for On-The-Go. These should not be confused with Micro-B plug and receptacle.
  - **Errata as of December 2000**: Released in December 2000.
  - **Pull-up/Pull-down Resistors ECN**: Released in May 2002.
  - **Errata as of May 2002**: Released in May 2002.
  - **Interface Associations ECN**: Released in May 2003.
    New standard descriptor was added that allows multiple interfaces to be associated with a single device function.
  - **Rounded Chamfer ECN**: Released in October 2003.
    A recommended, compatible change to Mini-B plugs that results in longer lasting connectors.
  - **Unicode ECN**: Released in February 2005.
    This ECN specifies that strings are encoded using UTF-16LE. USB 2.0 did specify that Unicode is to be used but it did not specify the encoding.
  - **Inter-Chip USB Supplement**: Released in March 2006.
  - **On-The-Go Supplement 1.3**: Released in December 2006.
    USB On-The-Go makes it possible for two USB devices to communicate with each other without requiring a separate USB host. In practice, one of the USB devices acts as a host for the other device.
  - **Battery Charging Specification 1.1**: Released in March 2007 (Updated 15 Apr 2009).
    Adds support for dedicated chargers (power supplies with USB connectors), host chargers (USB hosts that can act as chargers) and the No Dead Battery provision which allows devices to temporarily draw 100 mA current after they have been attached. If a USB device is connected to dedicated charger, maximum current drawn by the device may be as high as 1.8A. (Note that this document is not distributed with USB 2.0 specification package only USB 3.0 and USB On-The-Go.)
  - **Micro-USB Cables and Connectors Specification 1.01**: Released in April 2007.
  - **Link Power Management Addendum ECN**: Released in July 2007.
    This adds a new power state between enabled and suspended states. Device in this state is not required to reduce its power consumption. However, switching between enabled and sleep states is much faster than switching between enabled and suspended states, which allows devices to sleep while idle.
USB 3.0

The USB 3.0 Promoter Group announced on 17 November 2008, that version 3.0 of the specification had been completed and had made the transition to the USB Implementers Forum (USB-IF), the managing body of USB specifications.[74] This move effectively opened the specification to hardware developers for implementation in future products. The first USB 3.0 consumer products were announced and shipped by Buffalo Technology in November 2009, while the first certified USB 3.0 consumer products were announced 5 January 2010, at the Las Vegas Consumer Electronics Show (CES), including two motherboards by ASUS and Gigabyte Technology.[75] Manufacturers of USB 3.0 host controllers includes, but are not limited to, Renesas/NEC Electronics, Fresco Logic, Asmedia, Etron, VIA Labs and Texas Instruments. As of November 2010, Renesas is the only company to have passed USB-IF certification, although motherboards for Intel's Sandy Bridge processors have been seen with Asmedia and Etron host controllers. On October 28, 2010 Hewlett-Packard released the HP Envy 17 3D featuring a Renesas USB 3.0 Host Controller several months before some of their competitors. AMD is working with Renesas to add its USB 3.0 implementation into its chipsets for its 2011 platforms. At CES2011 Toshiba unveiled a laptop called "Toshiba Qosmio X500" that included USB 3.0 and Bluetooth 3.0, and a new series of Sony VAIO laptops that will include USB 3.0. The new models in the Dell XPS series are to include USB 3.0.

Features

A new feature is the "SuperSpeed" bus, which provides a fourth transfer mode at 5.0 Gbit/s. The raw throughput is 4 Gbit/s, and the specification considers it reasonable to achieve 3.2 Gbit/s (0.4 GB/s or 400 MB/s), or more, after protocol overhead.[77] When operating in SuperSpeed mode, full-duplex signaling occurs over two differential pairs separate from the non-SuperSpeed differential pair. This results in USB 3.0 cables containing two wires for power and ground, two wires for non-SuperSpeed data, and four wires for SuperSpeed data, and a shield that was not required in previous specifications.[78] To accommodate the additional pins for SuperSpeed mode, the physical form factors for USB 3.0 plugs and receptacles have been modified from those used in previous versions. Standard-A cables have extended heads where the SuperSpeed connectors extend beyond and slightly above the legacy connectors. Similarly, the Standard-A receptacle is deeper to accept these new connectors. On the other end, the SuperSpeed Standard-B connectors are placed on top of the existing form factor. A legacy standard A-to-B cable will work as designed and will never contact any of the SuperSpeed connectors, ensuring backward compatibility. SuperSpeed standard A plugs will fit legacy A receptacles, but SuperSpeed standard B plugs will not fit into legacy standard B receptacles, so a new cable can be used to connect a new device to an old host, but not to connect a new host to an old device; for that, a legacy standard A-to-B cable will be required.[79] SuperSpeed establishes a communications pipe between the host and each device, in a host-directed protocol. In contrast, USB 2.0 broadcasts packet traffic to all devices. USB 3.0 extends the bulk transfer type in SuperSpeed with Streams. This extension allows a host and device to create and transfer multiple streams of data through a single bulk pipe. New power management features include support of idle, sleep and suspend states, as well as link-, device-, and function-level power management.

The bus power spec has been increased so that a unit load is 150 mA (+50% over minimum using USB 2.0). An unconfigured device can still draw only one unit load, but a configured device can draw up to six unit loads (900 mA, an 80% increase over USB 2.0 at a registered maximum of 500 mA). Minimum device operating voltage is dropped from 4.4 V to 4 V.

USB 3.0 does not define cable assembly lengths, except that it can be of any length as long as it meets all the requirements defined in the specification. Although electronicdesign.com estimated cables will be limited to 3 m at SuperSpeed,[80] cables which support SuperSpeed are already available up to 5 m in length.[81][82]
The technology is similar to a single channel (“1×”) of PCI Express 2.0 (5 Gbit/s). It uses 8B/10B encoding, linear feedback shift register (LFSR) scrambling for data and spread spectrum. It forces receivers to use low frequency periodic signaling (LFPS), dynamic equalization, and training sequences to ensure fast signal locking.

**Availability**

Consumer products became available in January 2010.[75] [76] To ensure compatibility between motherboards and peripherals, all USB-certified devices must be approved by the USB Implementers Forum (USB-IF). At least one complete end-to-end test system for USB 3.0 designers is on the market.[83]

On 5 January 2010, USB-IF announced the first two certified USB 3.0 motherboards, one by Asus and one by Gigabyte.[76] [84] Previous announcements included Gigabyte’s October 2009 list of seven P55 chipset USB 3.0 motherboards,[85] and an ASUS motherboard that was cancelled before production.[86]

Commercial controllers were expected to enter into volume production in the first quarter of 2010.[87] On 14 September 2009, Freecom announced a USB 3.0 external hard drive.[88] On January 4, 2010, Seagate announced a small portable HDD with PC Card targeted for laptops (or desktop with PC Card slot addition) at the CES in Las Vegas.[89] [90]

Drivers are under development for Windows 7, but support was not included with the initial release of the operating system.[91] However, drivers are available for Windows through manufacturer websites. The Linux kernel has supported USB 3.0 since version 2.6.31, which was released in September 2009.[92] [93] [94]

Intel decided not support USB 3.0 until 2011,[95] which will slow down mainstream adoption. These delays may be due to problems in the CMOS manufacturing process,[96] a focus to advance the Nehalem platform,[97] a wait to mature all the 3.0 connections standards (USB3, PCIe3, SATA3.0) before developing a new chip set,[98] [99] or a tactic by Intel to boost its new Thunderbolt interface.[100] Current AMD road maps indicate that the new southbridges released in the beginning of 2010 will not support USB 3.0.[96] Market researcher In-Stat predicts the market share of USB 3.0 will be negligible until 2011.[101]

**Related standards**

The PictBridge standard allows for interconnecting consumer imaging devices. It typically uses USB for its underlying communication layer.

The USB Implementers Forum is working on a wireless networking standard based on the USB protocol. Wireless USB is intended as a cable-replacement technology, and will use ultra-wideband wireless technology for data rates of up to 480 Mbit/s.
Further reading

- **Debugging USB 2.0 for Compliance: It's Not Just a Digital World: Agilent Technologies Application Note 1382-3**

References

[13] USB Class Codes (http://www.usb.org/developers/defined_class) at USB.org
[14] Use class information in the interface descriptors. This base class is defined to be used in device descriptors to indicate that class information should be determined from the Interface Descriptors in the device.
[17] Universal Serial Bus 3.0 Specification,4.4.11 "Efficiency"

Universal Serial Bus


[29] (zip) Universal Serial Bus Micro-USB Cables and Connectors Specification to the USB 2.0 Specification, Revision 1.01 (http://www.usb.org/developers/docs/usb_20_081810.zip). USB Implementers Forum, Inc. 2007-04-07. Retrieved 2010-11-18. "Section 1.3: Additional requirements for a more rugged connector that will have durability past 10,000 cycles and still meet the USB 2.0 specification for mechanical and electrical performance was also a consideration. The Mini-USB could not be modified and remain backward compatible to the existing connector as defined in the USB OTG specification".


External links

- USB official website (USB Implementers Forum, Inc.) (http://www.usb.org)
- USB specifications and other documents (http://www.usb.org/developers/docs/)
- Intel Universal Host Controller Interface (UHCI) (http://download.intel.com/technology/usb/UHCI11D.pdf)
- USB 3.0 specification (.zip file) (http://www.usb.org/developers/docs/usb_30_spec_020411.zip)
- (http://www.agilent.com/find/USB)
Image Sources, Licenses and Contributors


File:Male and Female USB Connectors.jpg Source: http://en.wikipedia.org/w/index.php?title=File:Male_and_Female_USB_Connectors.jpg License: GNU Free Documentation License Contributors: Original uploader was Zephyris at en.wikipedia Later version(s) were uploaded by Osama bin dipesh at en.wikipedia.


License

Creative Commons Attribution-Share Alike 3.0 Unported http://creativecommons.org/licenses/by-sa/3.0/