

Assignment on USB Interfacing

Ayon Roy
201714018

ROYSHOUHAG@GMAIL.COM

Universal Serial Bus (USB) is an industry standard that establishes specifications for cables and connectors and protocols for connection, communication and power supply between computers, peripheral devices and other computers. Released in 1996, the USB standard is currently maintained by the USB Implementers Forum (USB-IF). There have been four generations of USB specifications: USB 1.x, USB 2.0, USB 3.x and USB4.

1. Overview

USB was designed to standardize the connection of peripherals to personal computers, both to communicate with and to supply electric power. It has largely replaced interfaces such as serial ports and parallel ports, and has become commonplace on a wide range of devices.

USB connectors have been increasingly replacing other types for battery chargers of portable devices.

Examples of peripherals that are connected via USB include keyboards, pointing devices, digital still and video cameras, printers, portable media players, disk drives and network adapters.

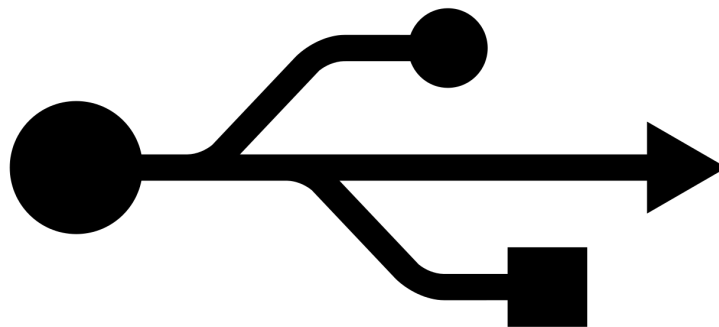


Figure: The basic USB trident logo

1.1 Objectives

The Universal Serial Bus was developed to simplify and improve the interface between personal computers and peripheral devices, when compared with previously existing standard or ad-hoc proprietary interfaces.

From the computer user's perspective, the USB interface improved ease of use in several ways. The USB interface is self-configuring, so the user need not adjust settings on the device and interface for speed or data format, or configure interrupts, input/output addresses, or direct memory access channels. USB connectors are standardized at the host, so any peripheral can use most available receptacle. USB takes full advantage of the additional processing power that can be economically put into peripheral devices so that they can manage themselves; USB devices often do not have user-adjustable interface settings. The USB interface is "hot pluggable", meaning devices can be exchanged without rebooting the host computer. Small devices can be powered directly from the USB interface, displacing extra power supply cables. Because use of the USB logos is only permitted after compliance testing, the user can have confidence that a USB device will work as expected without extensive interaction with settings and configuration; the USB interface defines protocols for recovery from common errors, improving reliability over previous interfaces. Installation of a device relying on the USB standard requires minimal operator action. When a device is plugged into a port on a running personal computer system, it is either entirely automatically configured using existing device drivers, or the system prompts the user to locate a driver which is then installed and configured automatically.

For hardware manufacturers and software developers, the USB standard eliminates the requirement to develop proprietary interfaces to new peripherals. The wide range of transfer speeds available from a USB interface suits devices ranging from keyboards and mice up to streaming video interfaces. A USB interface can be designed to provide the best available latency for time-critical functions, or can be set up to do background transfers of bulk data with little impact on system resources. The USB interface is generalized with no signal lines dedicated to only one function of one device.

2. USB Versions

2.1 USB 1.x

Released in January 1996, USB 1.0 specified data rates of 1.5 Mbit/s (Low Bandwidth or Low Speed) and 12 Mbit/s (Full Speed). It did not allow for extension cables or pass-through monitors, due to timing and power limitations. Few USB devices made it to the market until USB 1.1 was released in August 1998. USB 1.1 was the earliest revision that was widely adopted and led to what Microsoft designated the "Legacy-free PC".

Neither USB 1.0 nor 1.1 specified a design for any connector smaller than the standard type A or type B. Though many designs for a miniaturised type B connector

appeared on many peripherals, conformity to the USB 1.x standard was hampered by treating peripherals that had miniature connectors as though they had a tethered connection (that is: no plug or receptacle at the peripheral end). There was no known miniature type A connector until USB 2.0 (revision 1.01) introduced one.

2.2 USB 2.0

USB 2.0 was released in April 2000, adding a higher maximum signaling rate of 480 Mbit/s (60 MB/s) named High Speed or High Bandwidth, in addition to the USB 1.x Full Speed signaling rate of 12 Mbit/s.

Modifications to the USB specification have been made via Engineering Change Notices (ECN). The most important of these ECNs are included into the USB 2.0 specification package available from USB.org.

Features like: On-The-Go Supplement 1.3 USB On-The-Go makes it possible for two USB devices to communicate with each other without requiring a separate USB host; Link Power Management Addendum ECN which adds a sleep power state.

2.3 USB 3.x

The USB 3.0 specification was released on 12 November 2008, with its management transferring from USB 3.0 Promoter Group to the USB Implementers Forum (USB-IF), and announced on 17 November 2008 at the SuperSpeed USB Developers Conference.

USB 3.0 adds a SuperSpeed transfer mode, with associated backward compatible plugs, receptacles, and cables. SuperSpeed plugs and receptacles are identified with a distinct logo and blue inserts in standard format receptacles.

The SuperSpeed bus provides for a transfer mode at a nominal rate of 5.0 Gbit/s, in addition to the three existing transfer modes. Its efficiency is dependent on a number of factors including physical symbol encoding and link level overhead. At a 5 Gbit/s signaling rate with 8b/10b encoding, each byte needs 10 bits to be transmitted, so the raw throughput is 500 MB/s. When flow control, packet framing and protocol overhead are considered, it is realistic for 400 MB/s (3.2 Gbit/s) or more to be delivered to an application.[18](4–19) Communication is full-duplex in SuperSpeed transfer mode; earlier modes are half-duplex, arbitrated by the host.

Low-power and high-power devices remain operational with this standard, but devices using SuperSpeed can take advantage of increased available current of between 150 mA and 900 mA, respectively.

USB 3.1, released in July 2013 has two variants. The first one preserves USB 3.0's SuperSpeed transfer mode and is labeled USB 3.1 Gen 1, and the second version introduces a new SuperSpeed+ transfer mode under the label of USB 3.1 Gen 2. SuperSpeed+ doubles the maximum data signaling rate to 10 Gbit/s, while reducing line encoding overhead to just 3 percentage by changing the encoding scheme to

128b/132b. The number of connectors used for USB 3.1 has also been reduced to two: USB-A and USB-C.

USB 3.2, released in September 2017, preserves existing USB 3.1 SuperSpeed and SuperSpeed+ data modes but introduces two new SuperSpeed+ transfer modes over the USB-C connector with data rates of 10 and 20 Gbit/s (1.25 and 2.5 GB/s). The increase in bandwidth is a result of multi-lane operation over existing wires that were intended for flip-flop capabilities of the USB-C connector. USB 3.2 is also the first version to use the USB-C connector as the sole connector.

USB 3.0 also introduced the UASP protocol which provides generally faster transfer speeds than the BOT (Bulk-Only-Transfer) protocol.

2.4 USB4

The USB4 specification was released on 29 August 2019 by USB Implementers Forum.

USB4 is based on the Thunderbolt 3 protocol specification. It supports 40 Gbit/s throughput, is compatible with Thunderbolt 3, and backwards compatible with USB 3.2 and USB 2.0. The architecture defines a method to share a single high-speed link with multiple end device types dynamically that best serves the transfer of data by type and application.

The USB4 specification states that following technologies shall be supported by USB4.

3. System Architecture

A USB system consists of a host with one or more downstream ports, and multiple peripherals, forming a tiered-star topology. Additional USB hubs may be included, allowing up to five tiers. A USB host may have multiple controllers, each with one or more ports. Up to 127 devices may be connected to a single host controller. USB devices are linked in series through hubs. The hub built into the host controller is called the root hub.

A USB device may consist of several logical sub-devices that are referred to as device functions. A composite device may provide several functions, for example, a webcam (video device function) with a built-in microphone (audio device function). An alternative to this is a compound device, in which the host assigns each logical device a distinct address and all logical devices connect to a built-in hub that connects to the physical USB cable.

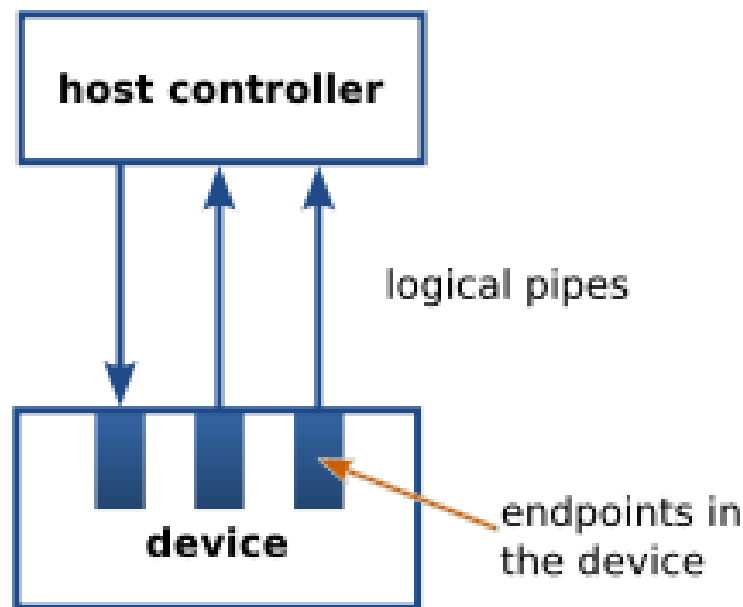


Figure: USB endpoints reside on the connected device: the channels to the host are referred to as pipes

USB device communication is based on pipes (logical channels). A pipe is a connection from the host controller to a logical entity within a device, called an endpoint. Because pipes correspond to endpoints, the terms are sometimes used interchangeably. Each USB device can have up to 32 endpoints (16 in and 16 out), though it is rare to have so many. Endpoints are defined and numbered by the device during initialization (the period after physical connection called "enumeration") and so are relatively permanent, whereas pipes may be opened and closed.

When a host starts a data transfer, it sends a TOKEN packet containing an endpoint specified with a tuple of device address, endpoint number. If the transfer is from the host to the endpoint, the host sends an OUT packet a specialization of a TOKEN packet with the desired device address and endpoint number. If 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 is ignored. Otherwise, it is 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 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 throughput of each USB port is determined by the slower speed of either the USB port or the USB device connected to the port.

High-speed USB 2.0 hubs contain devices called transaction translators that convert between high-speed USB 2.0 buses and full and low speed buses. There may be one translator per hub or per port.

Because there are two separate controllers in each USB 3.0 host, USB 3.0 devices 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 earlier devices are set in the legacy manner.

4. Transactions

The basic transactions of USB are:

- OUT transaction
- IN transaction
- SETUP transaction
- Control transfer exchange

5. Protocol layer

During USB communication, data is transmitted as 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.

6. Limitations

USB cables are limited in length, as the standard was meant to connect to peripherals on the same table-top, not between rooms or between buildings. However, a USB port can be connected to a gateway that accesses distant devices. USB has a strict

"tree" topology and "master-slave" protocol for addressing peripheral devices; peripheral devices cannot interact with one another except via the host, and two hosts cannot communicate over their USB ports directly. Some extension to this limitation is possible through USB On-The-Go. A host cannot "broadcast" signals to all peripherals at once—each must be addressed individually. Some very high speed peripheral devices require sustained speeds not available in the USB standard. While converters exist between certain "legacy" interfaces and USB, they may not provide full implementation of the legacy hardware; for example, a USB to parallel port converter may work well with a printer, but not with a scanner that requires bi-directional use of the data pins.

For a product developer, use of USB requires implementation of a complex protocol and implies an "intelligent" controller in the peripheral device. Developers of USB devices intended for public sale generally must obtain a USB ID which requires a fee paid to the Implementers' Forum. Developers of products that use the USB specification must sign an agreement with Implementer's Forum. Use of the USB logos on the product require annual fees and membership in the organization.

7. References

- <https://en.wikipedia.org/wiki/USB>
- <https://www.keil.com/pack/doc/CMSIS/Driver/html>
- <https://www.researchgate.net/publication/256981030>