91.548 Robotics I
USB Technical Overview

David J. Meppelink
April 10, 2003

Introduction

Universal Serial Bus (USB) is a standard for connecting peripherals (such as mice, keyboards, cameras, disks) to PCs.

Marketing quotes from the USB web site[1]:

1. First, USB replaces all the different kinds of serial and parallel port connectors with one standardized plug and port combination.
2. With USB-compliant PCs and peripherals, you just plug them in and turn them on! USB makes the whole process automatic.
3. Thanks to another USB feature known as 'hot-swapping' you don't even need to shut down and restart your PC to attach or remove a peripheral. Just plug it in and go!
4. USB also lets you connect many peripherals at one time.
5. Another cool USB feature is that it distributes electrical power to many peripherals. Again, USB lets the PC automatically sense the power that's required and deliver it to the device.

This technical overview is based on the "Architectural Overview" chapter of the USB 2.0 Specification[2].

The USB Standards

The standard was developed by the USB Implementers Forum (Compaq, Hewlett-Packard, Intel, Microsoft, NEC, Philips, etc.). Obviously, it started on the PC; Mac supports it too.

There are two versions of the standard:

USB 1.1 (1995) defines two speeds: 12Mbps (full speed) and 1.5Mbps (low speed) and two types of connectors: Series A and Series B.

USB 2.0 (April 2000) defines three speeds: 480Mbps (high speed), 12Mbps and 1.5Mbps. It is completely backwards compatible with USB 1.1. The specification document is 650 pages long.

Shortly after USB 2.0 was released (October 2000), an Engineering Change Notice (ECN) added Series mini-B connectors.

[1] The USB home page is `http://www.usb.org`.

[2] The USB 2.0 specification is available for download from `http://www.usb.org/developers`.

Speed Comparisons

This table shows the throughput (in megabytes per second) of USB and other common interface standards[3]:

Connection Type	Mbytes/sec	Comments
serial port	0.013	
standard parallel port	0.115	
Original USB	1.5	(10× faster than typical IO ports)
ECP/EPP parallel port	3	
IDE	16.7	
SCSI-1	5	
SCSI-2 (Fast SCSI, Fast Narrow SCSI)	10	
Fast Wide SCSI (Wide SCSI)	20	
Ultra SCSI (SCSI-3, Fast-20, Ultra Narrow)	20	
UltraIDE	33	(typical max for PC disks)
Wide Ultra SCSI (Fast Wide 20)	40	
Ultra2 SCSI	40	
IEEE-1394 (Firewire)	50	
Hi-Speed USB	60	(2× faster than typical disks)
Wide Ultra2 SCSI	80	
Ultra3 SCSI	80	
Wide Ultra3 SCSI	160	
FC-AL Fiber Channel	400	

Bus Topology

USB devices are either (1) **hubs**, which provide additional attachment points to the USB or (2) **functions**, which provide capabilities such as digital camera, disk, keyboard, mouse.

USB uses a tiered star topology made up of hub and function nodes. A hub is at the center of each star. A wire segment is a point-to-point connection between a hub and another hub or function at the next tier.

Due to timing constraints, USB is limited to 7 tiers. The host is alone at tier 1 and there can be only one host in a USB system. Only functions are allowed at tier 7. Tiers 2 through 6 can have hubs or functions. A compound device (which has both a hub and a function) occupies two tiers, so it cannot placed in tier 7.

Electrical

USB uses a four-conductor cable: 5V power and ground (28 to 20 AWG) and two signal wires (28 AWG twisted pair).

There are three data rates: high-speed (480 Mbps), full-speed (12 Mbps), low-speed (1.5 Mbps). If a full-speed or low-speed device is attached to a host through one or more hubs, then the transfer between the host and hub(s) is always at high-speed. This minimizes bandwidth consumption by those devices.

[3] This table is adapted from the Frequently Asked Question "How does this compare to other connections used with PCs and workstations?" at `http://www.usb.org/faq`

The clock is encoded using NRZI[4] with bit stuffing[5].

The specification defines these power classes:

1. **Bus-powered hubs** draw power for the hub itself and downstream ports from the upstream port. Must draw only 100 mA at startup and 500 mA or less after configuration. Each downstream port can deliver 100 mA maximum, regardless of the draw from other ports.
2. **Self-powered hubs** draw power for the hub itself and downstream ports from an external source. But they can draw up to 100 mA in order to function when the hub is powered off. The hub can supply 100 to 500 mA to each downstream port.
3. **Low-power bus-powered functions** draw no more than 100 mA from the USB port.
4. **High-power bus-powered functions** draw no more than 100 mA at powerup and up to 500 mA after configuration. The switch to high power is controlled by software.
5. **Self-powered functions** can draw up to 100 mA in order to function when the device is powered off. All other power comes from an external source.

If devices draw too much current, the hub must report the condition to the USB software.

USB also supports power management functions so that the host can power down devices.

Mechanical

USB defines four connector types:

1. **Series A** plugs are rectangular (15.7mm × 7.5mm) and are always oriented toward the host. Series A receptacles function as outputs from hosts and/or hubs.
2. **Series B** plugs are nearly square (11.5mm × 10.5mm) and are always oriented toward the device. Series B receptacles function as inputs to hubs or devices.
3. **Series mini-B** plugs are small rectangular (6.8mm × 3mm) and are always oriented toward the device. Series mini-B receptacles function as inputs to hubs or devices.
4. **Captive cable assembly** is any non-standard connection to the device. It must use a Series A plug to connect to the host or hub.

These connectors prevent illegal wiring arrangements.

The spec discourages extension cables because they could allow the point-to-point length to exceed the electrical limit of 5 meters.

[4] Non Return to Zero Invert (NRZI): A method of encoding serial data in which ones and zeroes are represented by opposite and alternating high and low voltages where there is no return to zero (reference) voltage between encoded bits. This eliminates the need for clock pulses.

[5] Bit Stuffing: Insertion of a zero bit into a data stream to cause an electrical transition on the data wires, allowing a PLL to remain locked.

Bus Protocol

USB is a polled bus; the host controller initiates all data transfer transactions. Most transactions involve three packets:

1. The host controller sends a “token packet” that describes the type and direction of the transaction[6], device address[7], and endpoint number[8].
2. The source of the transaction sends a data packet or a “no data” packet.
3. The destination sends a handshake packet that indicates if the data transfer was successful.

Transactions between the host controller and a hub use four packets to manage the transfer to a low-speed or full-speed device.

The protocol includes separate CRCs for control and data fields of each packet. The CRC gives 100% coverage on single-bit and double-bit errors. Error recovery can be performed in hardware (retry three times) and software (client implementation specific).

The data connection between the host software and a particular endpoint is called a **pipe**. Each pipe operates independently and a single device can have more than one pipe.

A pipe supports only one of the following types of transfers at a time:

1. **Control Transfers** are used to configure a device and can be used for other device-specific purposes, including control of other pipes on the device.
2. **Bulk Data Transfers** are generated or consumed in relatively large and bursty quantities and have are not timing critical. Examples: printers and scanners.
3. **Interrupt Data Transfers** are used for timely (limited-latency) and reliable delivery of data. Examples: mouse, game pad.
4. **Isochronous Data Transfers** (also called streaming real time transfers) occupy a prenegotiated amount of USB bandwidth with a prenegotiated delivery latency. The data is continuous and real-time in creation, delivery, and consumption.

[6] The direction can be either host-to-device or device-to-host.

[7] Device Address: A seven-bit value representing the address of a device on the USB. Devices are assigned a unique device address by the USB System Software.

[8] Device Endpoint: A uniquely addressable portion of a USB device that is the source or sink of information in a communication flow between the host and device.