USB EXPLAINED: AN INTRODUCTION TO USB AND ITS FUTURE

STEPHEN OLSEN, SOFTWARE ENGINEER, MENTOR GRAPHICS
COLIN WALLS, EMBEDDED SOFTWARE TECHNOLOGIST, MENTOR GRAPHICS

WHITEPAPER

Mentor Graphics®

EMBEDDED SOFTWARE

www.mentor.com

USB HISTORY

The Universal Serial Bus (USB) was introduced to the world of PC computing to solve the growing problem of how to connect peripherals to a PC. The world of desktop computers needed a way, not only to physically connect devices, but also allow them to be identified and categorized to enable a simple ease of use and method of discovery.

Prior to USB, many people overloaded the use of their serial and parallel ports to support more than one peripheral on that port. Stephen's story:

"I had a printer, two dongles, and a scanner on a single parallel port. It was so complex that booting with the scanner selected meant that the dongle was not recognized. An A/B switch was added to make all of the components work together. This meant booting the computer with the switch set correctly in order to access that particular device and its accompanying software. Reboots were necessary to run the different sets of software and their respective peripherals. It was very frustrating and not very user friendly."

In 1996, the 1.0 USB specification was released to solve this type of problem, not only with serial and parallel ports, but with keyboards, mice, and many other low/full-speed devices. USB 1.0 offered a single bus at either 12 Mb/s bandwidth (full speed) or 1.5 Mb/s (low speed) shared between up to 127 peripherals, all with different needs. It is superseded by the USB 1.1 specification released in September 1998.

USB is a host-centric tree topology where all devices are hooked to the host via a set of hubs. Each device supports up to 32 endpoints: 16 in and 16 out. Each endpoint is defined as either: control, bulk, interrupt, or isochronous.

The basic types of transactions are as follows:

- **Control.** Used to configure and enumerate a peripheral, allowing the host software to interrogate the peripheral to find out what it is and how it can use it. Every peripheral needs to support this traffic type.
- **Bulk.** Reliable data transport that guarantees sequential reception of data. Retransmits are used to ensure the data is received correctly. Depending on the current bus utilization, the throughput with bulk transfers will vary. Typical uses of bulk transport are Disk-On-Keys and printers.
- **Interrupt.** Meant for low bandwidth transfers with periodic transport needs. The host pre-schedules an interval at which the peripheral is queried. Unlike bulk, this transport type is guaranteed allocation on the bus for transport. The typical use is keyboards, mice, track-balls, and game controllers.
- **Isochronous**. Guaranteed bandwidth, but not guaranteed reception. This method is typically associated with video cameras and audio speakers. It is used to insure that the data always has a path. During enumeration and configuration, the host determines if a configuration setting has enough bandwidth to talk with a certain peripheral. If sufficient bandwidth is not available, it rejects the configuration selection in favor of less bandwidth consumption.

All peripherals support a single control endpoint and any number of bulk, interrupt, and isochronous endpoints to fit their application up to the maximum of 32 endpoints.

The USB Implementers Forum saw that adoption was good, but with the competition of FireWire, many high-speed peripherals were bypassing USB, and therefore, USB needed to grow with the increased demand for bandwidth. The USB 2.0 specification released in April of 2000, added the key feature of high-speed devices, cranking up the USB bandwidth to 480Mb/s or 40 times full-speed USB bus speed. The USB 2.0 specification maintained backward compatibility with all USB 1.1 compliant host and peripherals as interoperability was essential. Disk-On-Keys were considered a "killer app" for USB 2.0.

Besides providing a universal serial bus on which all peripherals talk, the USB Implementers Forum also adopted a class approach to device drivers. No longer do vendors of peripherals need to provide both a host and a peripheral driver, they can provide their functionality as part of a particular class of devices. Then the host will provide a set of class drivers for use with a set of peripherals.

One of the classes is the human interface device (HID) class. It is well adopted by providers of mice, keyboards, joysticks, and trackballs. Most of these devices do not ship with a host driver as they just use the functionality provided in the host for the HID class.

In addition to the HID class, there is mass storage class (MSC), printer, video, audio, communications, hub, and device firmware upgrade (DFU). Where the needs are unique, the vendor can provide a vendor-specific driver for use with a particular host operating system. Each class is as complex or as simple as needed to accomplish the basic set of peripherals. For instance, the communications class is made up of a set of communications paradigms including modem and Ethernet subclasses. Each subclass represents a set of peripherals that have specific needs, and are subsets of the entire communications class.

USB support for embedded designs, facilitating the implementation of both peripheral and host functionality, is available from a number of manufacturers, including Mentor Graphics.

USB ON-THE-GO

There was still a missing element in USB 2.0: The ability for peripherals to talk to peripherals. For example, the ability to sync a cellular phone contact database with a PDA's contact database. Therefore, in 2001, the On-The-Go (OTG) supplement addressed this concern. The OTG supplement to the USB 2.0 specification allows USB peripherals to assume the role of host to another USB peripheral. In keeping with the plug-and-play nature of USB, the implementers decided to allow the host and peripheral to swap roles without removing the cable. This is called the Host Negotiation Protocol (HNP), a complex signaling algorithm designed specifically to address the consumer's need for operations to always succeed, if possible. For low power consumption, USB was allowed to completely be powered down, and a peripheral could wake up the host by issuing a Session Request Protocol (SRP).

Embedded considerations for using USB in a design are split between need to implement functionality as a peripheral device, a host, or both as in OTG.

There was much enthusiasm and optimism about OTG when it first appeared and many manufacturers, including Mentor Graphics, provided support for embedded developers to implement this new technology. However, take up was disappointing and there are few OTG-enabled devices on the market today.

WIRELESS USB

With the increasing use and popularity of wireless networking, it seemed obvious that USB should embrace such technology. So, in 2005, the wireless USB specification 1.0 was released. Wireless USB is specified to run at 480Mbps at 3 meters, and 110Mb/s at 10 meters.

However, regardless of there being much excitement about Wireless USB and pledges of support from major manufacturers, it never took off. Although technically, Wireless USB had many advantages over Bluetooth and Wi-Fi – its main "competitors" – these other technologies were already well entrenched and there was no room for a newcomer.

USB 3.0

The standards committees have taken an approach of "more of the same, but better and faster" with the specification of USB 3.0 – SuperSpeed USB. There is a clear recognition that the volumes of data being handled by modern consumer devices – still and moving images and audio – is increasing dramatically and USB needs to grow to accommodate their connectivity requirements.

The key enhancement is significantly greater speed – just under 5Gb/s, which is a 10X improvement over USB 2.0. As battery life is critical to portable equipment, device power management capabilities are defined in the USB 3.0 specification. Recognizing the popularity of devices that derive their power from the bus (or are simply charged that way) USB 3.0 includes the possibility to deliver more power via the bus.

Unfortunately, the cost of these enhancements is a change in connectors to accommodate additional wiring. Devices adhering to this spec are just coming on to the market and support for embedded developers implementing USB 3.0 is beginning to appear.

Resources:

You can find more detailed information and specifications located at the following websites:

USB: http://www.usb.org

Mentor Graphics/Mentor Embedded USB support: http://www.mentor.com/embedded

For the latest product information, call us or visit: www.mentor.com

Corporate Headquarters
Mentor Graphics Corporation
8005 SW Boeckman Road
Wilsonville, OR 97070-7777
Phone: 503.685.7000
Fax: 503.685.1204

Sales and Product Information
Phone: 800.547.3000

Silicon Valley
Mentor Graphics Corporation
1001 Ridder Park Drive
San Jose, California 95131 USA
Phone: 408.436.1500
Fax: 408.436.1501

North American Support Center
Phone: 800.547.4303

Europe
Mentor Graphics
Deutschland GmbH
Arnulfstrasse 201
80634 Munich
Germany
Phone: +49.89.57096.0
Fax: +49.89.57096.400

Pacific Rim
Mentor Graphics (Taiwan)
Room 1001, 10F
International Trade Building
No. 333, Section 1, Keelung Road
Taipei, Taiwan, ROC
Phone: 886.2.87252000
Fax: 886.2.27576027

Japan
Mentor Graphics Japan Co., Ltd.
Gotenyama Garden
7-35, Kita-Shinagawa 4-chome
Shinagawa-Ku, Tokyo 140-0001
Japan
Phone: +81.3.5488.3033
Fax: +81.3.5488.3004