Challenges of Migrating to Wireless USB

ABSTRACT

As manufacturers migrate products from wired to wireless USB, they will encounter new challenges created by a radically-changed communication protocol and power consumption issues. Although the USB interface has enjoyed great success, the Certified Wireless USB standard faces competition from Bluetooth and Wi-Fi. Satisfying demanding consumers means anything less than faultless performance might jeopardize market acceptance of new mobile devices. We believe that manufacturers have only one chance to get their Wireless USB products right. This article describes these issues and illustrates how developers can use purpose-built protocol analyzers to address them.

Introduction

It is clear that the Certified Wireless USB from the USB-IF version 1.0 standard has to deliver superior benefits to consumers in order to become an accepted interface, especially in the face of competition from Bluetooth and Wi-Fi (see the sidebar entitled ***The Interface Wars*** for more information on competition among interface standards). Technically, Wireless USB offers high data rates and a superior power management scheme that will help it prevail over other more power-hungry interface technologies.

But most experts think that the two measures by which consumers will judge the Wireless USB interface, either consciously or unconsciously, are customer experience and overall performance. The ideal customer experience would be that devices work straight out of the box with no setup or configuration, no fussing about with cables, and seamless operation. Important performance parameters include a high transfer speed with low error rate, the ability to reliably connect at a reasonable distance regardless of device positioning and room furnishings, and zero negative impact on battery life.

And if this all wasn't enough, manufacturers need to act quickly to minimize their time-to-market. Although device wire adapters and dongles are expected first, consumer devices with integrated Wireless USB are predicted to appear as soon as the end of 2006.

Deep Inside the Wireless USB Protocol

Wireless USB is based on WiMedia Ultrawideband (UWB) and takes advantage of many of its capabilities. The fundamental element of the Wireless USB protocol is the Micro-scheduled Management Command (MMC). MMCs are used to help devices discover information about a Wireless USB cluster, notify their intentions, manage power, and schedule data transmissions efficiently to attain very high throughputs.

MMCs are UWB control frames composed of Information Elements (IE). Naturally, there are several different IEs, each for a different purpose. For example, the Host Information Element provides information that assists a device in connecting to a specific cluster.

With USB, Start of Frame (SOF) packets are regularly sent to devices and are used for synchronization. They also ensure that devices will not enter a power-down mode. Wireless USB does not use SOF packets, but rather transmits MMCs to all devices from time to time (See Figure 1).

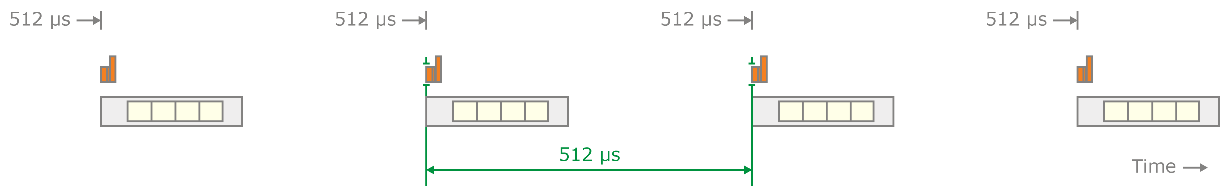

Figure 1 - Periodic Micro-scheduled Management Command Packets

Shown above in orange are four Micro-scheduled Management Commands (MMCs). All figures in this article with a time base are drawn to scale; in this example, MMCs are sent every 512 microseconds.

An important Information Element is the Channel Time Allocation (CTA) IE. The CTA IE contains a transmission schedule which tells devices when to transmit or receive data. A CTA IE can be of one of three kinds: Device Transmit (DT), Device Receive (DR) and Device Notification Time Slot (DNTS).

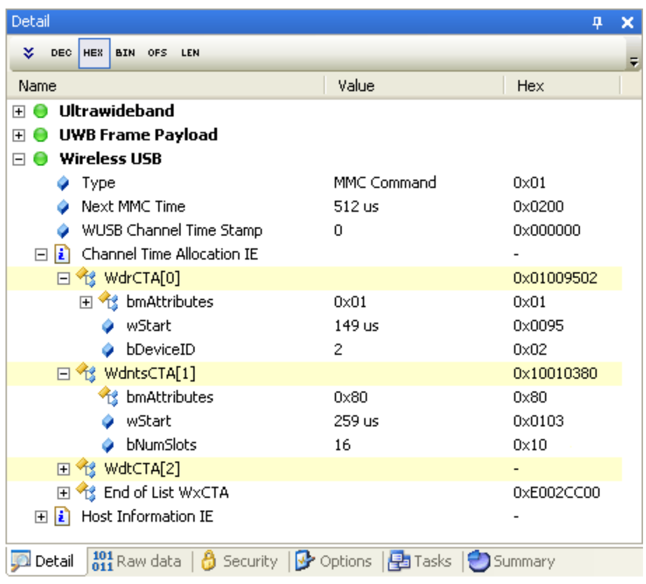

Figure 2 - The Channel Time Allocation Information Element in Action

This screen capture from an Ellisys Wireless USB Explorer 300 protocol analyzer shows the decoded packets of a Device Receive CTA directed to the device with ID 2 followed by a Device Notification Time Slot CTA, followed in turn by a Device Transmit CTA. The same information is presented below on a timing diagram (Figure 6).

In wired USB, data transmissions were executed using IN and OUT token packets. Named with respect to the host, an IN is used by the host to read data from a device while an OUT writes data from the host to a device. So, a Wireless USB DT CTA is the equivalent of a USB token IN; a Wireless USB DR CTA is the equivalent of a USB token OUT.

The DNTS CTA is more similar to wired USB electrical states. When a wired USB device would like to notify its presence to the host, it switches in a resistor on one of the two data lines to create state change. As electrical states are not possible on a wireless medium, devices need some means to initiate a data transmission. Device Notification Time Slots enable devices to send notifications when they require, for example, to notify their intention to connect.

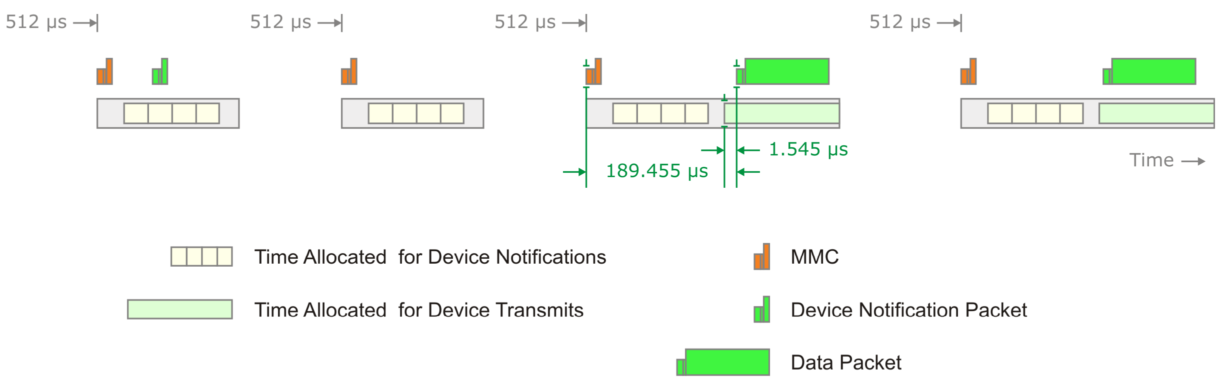

Figure 3 - Device Notification Time Slots

> Several MMCs were sent in this example timing diagram. The DNTS allow any device to send notifications. Two DTs were scheduled. Large data packets were sent by the wireless USB device. A Device Notification was sent by the device after the first MMC.

Such conceptual parallels permit us to construct helpful correlations between USB and Wireless USB packets. Some are listed below in Figure 4. Please note that these are not exact equivalents but they do provide a good starting point for someone familiar with USB. It is also important to note that directions were host-centric with wired USB (OUT means that the host transmits) and are now device-centric in Wireless USB (DR means that the device receives).

Wired USB Protocol Elements	Wireless USB Equivalent
Start of Frame (SOF)	Micro-scheduled Management Command (MMC)
IN Token	Device Transmit (DT CTA)
OUT Token	Device Receive (DR CTA)
Data flow control & electrical state control	Device Notification Time Slot (DNTS CTA)

Figure 4 - USB Transactions and Their Wireless USB Equivalents

Since the protocol itself is so different, USB experts will not find much that they recognize in Wireless USB. Happily, leading USB analysis software is clever enough to automatically decode Ultrawideband traffic and display its equivalent USB traffic by using the concepts described above. Developers can thus:

- Quickly understand the relationship between Ultrawideband traffic and Wireless USB traffic;
- Focus on low-level detail by viewing UWB traffic for items such as timing and error rates; and,
- Save time when testing high-level transactions by viewing equivalent USB traffic.

Figure 5 shows the same traffic in both Ultrawideband and Wireless USB decode views.

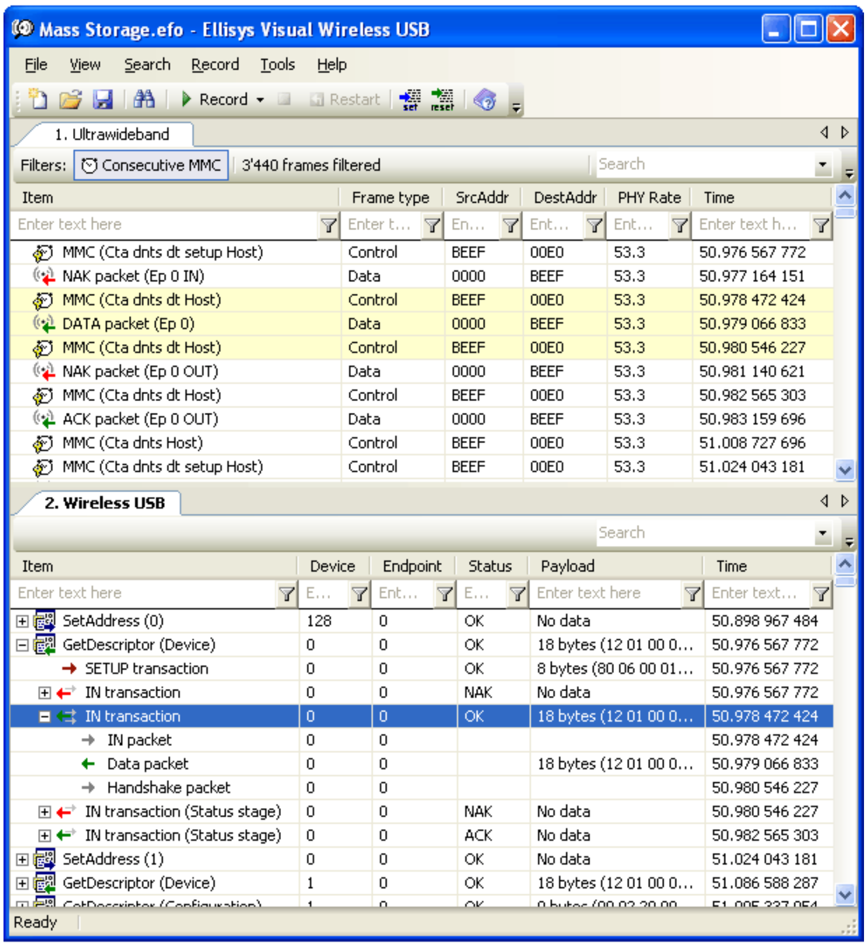

Figure 5 - Ultrawideband and Wireless USB Decode Views

> The above screen shows a GetDescriptor standard request in the Wireless USB view and the related UWB frames.

Power Management

New in Wireless USB is a sophisticated power management scheme designed to maximize battery life. Of course, this power management scheme is totally transparent to users of wireless USB devices. We will describe the power management process briefly and examine the resulting protocol traffic.

Powering up a Wireless USB device initiates a search for a host. The device scans each radio channel for MMC frames. When a valid MMC is detected, the device stays on that channel and looks for a Host Information Element. If the host is trusted by the device, the device waits for a DNTS and sends a DN_Connect notification. The host begins an authentication process after receiving this notification. If successful, the host schedules data transmissions using a Channel Time Allocation (CTA) IE.

The device examines further MMCs to check if it is scheduled for a data transmission. If none is scheduled, it can sleep (reduce its power consumption) until the next MMC by powering down the radio and any unused processing elements. If a traffic exchange is scheduled for the device, a timer is programmed to switch the radio on and thus “wake up” at the right time to execute the requested data transmission.

The first data exchange is similar to the classic USB enumeration. The host requests the device's descriptors to discover the device's capabilities and then loads appropriate drivers. Wireless USB was designed to use the same drivers as USB 2.0 without modification.

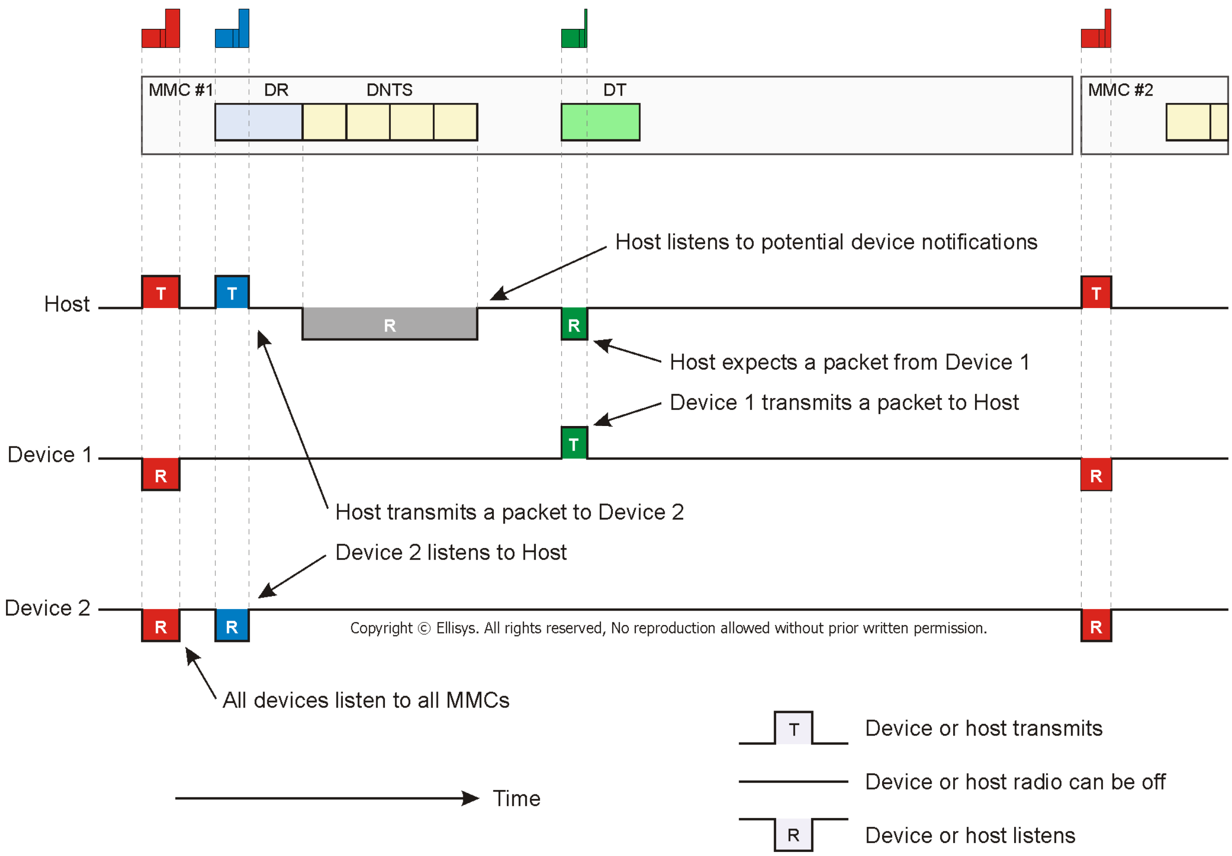

Figure 6 - Power Management Protocol Exchange

As shown in the state traces of Device 1 and Device 2, Wireless USB device radios are only powered on sometimes, reducing average power consumption and hence maximizing battery life.

Figure 6 illustrates the protocol transactions behind power management for two devices, Device 1 and Device 2. The horizontal axis represents a timebase. MMC #1 includes three CTAs. The first CTA schedules a DR from Device 2. The second CTA describes a DNTS not used by any device. The last CTA schedules a DT to Device 1.

Each device processes its CTA and then switches on its radio on as instructed. The resulting power state changes for each device are shown in their respective traces. Both devices should turn their radios on in order to receive the next MMC, MMC #2.

Considering the whole process, we can see that a device can be woken up only when MMC frames are supposed to be transmitted and when a given DT or DR is scheduled for that specific device. The rest of the time the radio can be off. Because the radio is only on for short periods of time, the device's average power consumption is lowered, saving power.

Cutting the Wire Means More Errors

The star topology of Wireless USB is conceptually simple while the topology of wired USB device connections can be a chaotic hierarchy with multiple links of differing speeds. Chip designers and software developers will soon realize that simplifying the life of end users complicates theirs!

Simpler topology aside, the wireless link is more prone to data errors than a shielded cable. Figure 7 illustrates how the expected error rate increases with both distance and packet length at a fixed nominal data rate of 110 Mbps. Although error rates fall dramatically at distances under 4 meters, packet lengths will increase as Wireless USB is used in digital camcorders, MP3 players and other devices that exchange large volumes of data.

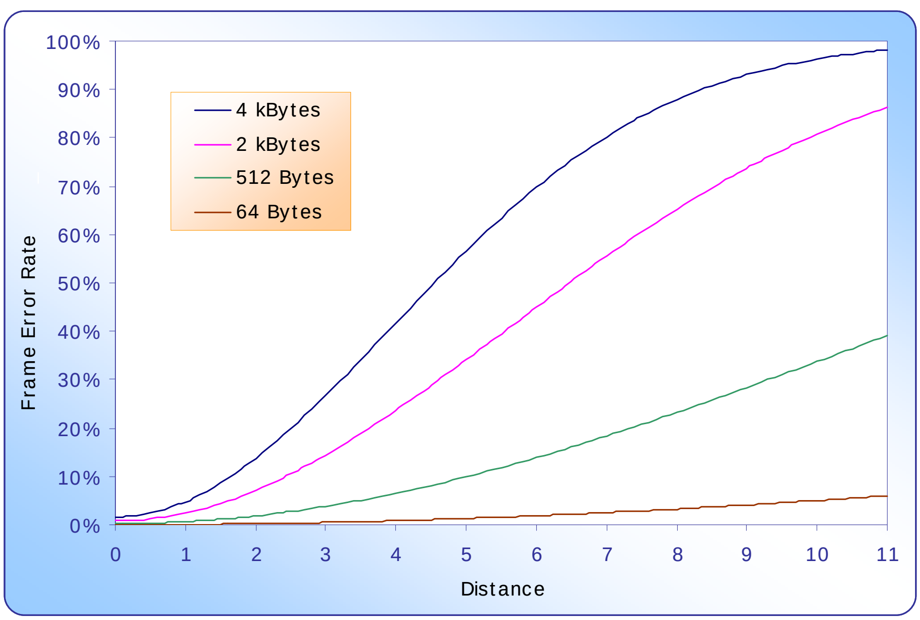

Figure 7 - Effect of Distance on Frame Error Rate

> Frame error rates are much higher for Wireless USB than USB and increases with both packet length and distance. As shown above, our calculations indicate that about 50% of packets with a length of 2 kBytes, and 80% of packets with a length of 4 kBytes, will have errors at a distance of 7 meters.

A well-designed Wireless USB host should be able to dynamically adapt to ensure a superior user experience. For example, it can transmit packets at a lower nominal data rate to ensure better reception by a device located further away. A host could also request that the device send packets at a specific data rate to minimize errors and subsequent retransmissions. Minimizing retransmissions will, of course, maximize battery life. Hosts can choose an appropriate data rate based on the frame error rate (FER).

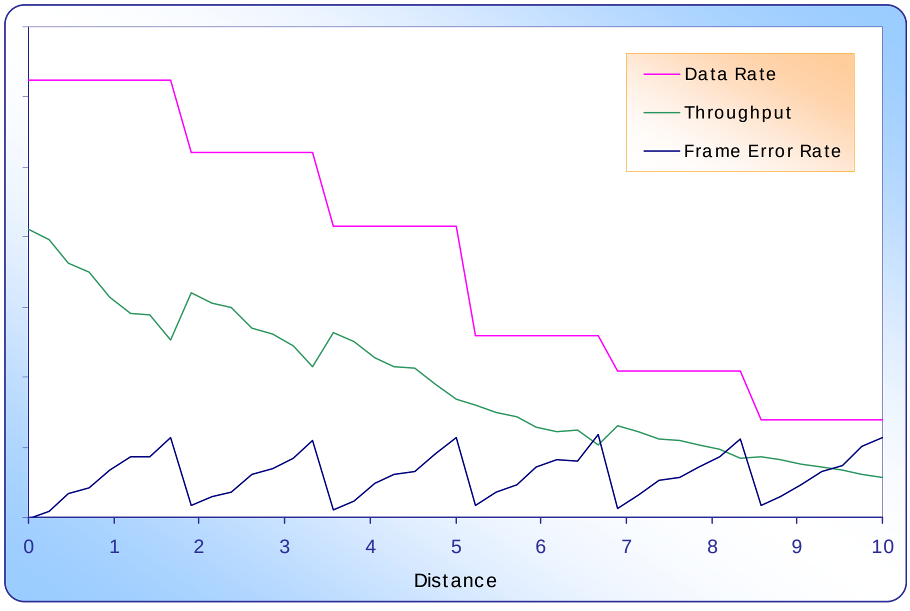

Figure 8 - Dynamic Data Rate Adjustment

> Decreasing the data transmission rate in a situation with a high frame error rate optimizes throughput.

Figure 8 shows the anticipated result of a walking user carrying a Wireless USB device. At a short distance from the host, the transmission speed is high and frame error rate is low. As he walks away from the host, he reaches a distance of 1.5 meters and the error rate increases. When the error rate passes a threshold determined by the host, the host dynamically lowers the data rate. The relative FER should decrease at this lower speed and so throughput increases. This cycle repeats as the user continues to walk further away from the host.

Long battery life will be an important product attribute of Wireless USB devices that rely on internal batteries. Users of current wireless devices that use Wi-Fi or Bluetooth quickly learn that power consumption is proportional to the amount of time that the radio is on. This radio time varies with both traffic and overhead. While traffic is under user control, overhead is not and includes initialization, flow control, error checking, data error recovery, and protocol error recovery.

Implications for Developers

Learning a new protocol is never easy, but is especially difficult in the course of a fast-tracked R&D project. Connected between a computer and a peripheral, a wired or wireless USB protocol analyzer records all data and control information so that developers can see exactly what and when data was transmitted. The raw bits, bytes and packets are decoded into comprehensive protocol exchanges. This higher level of abstraction is much easier to understand and thus faster to work with. Traffic can be searched or filtered to identify the cause of errors. Once correct operation is verified, developers can use a protocol analyzer to fine-tune performance.

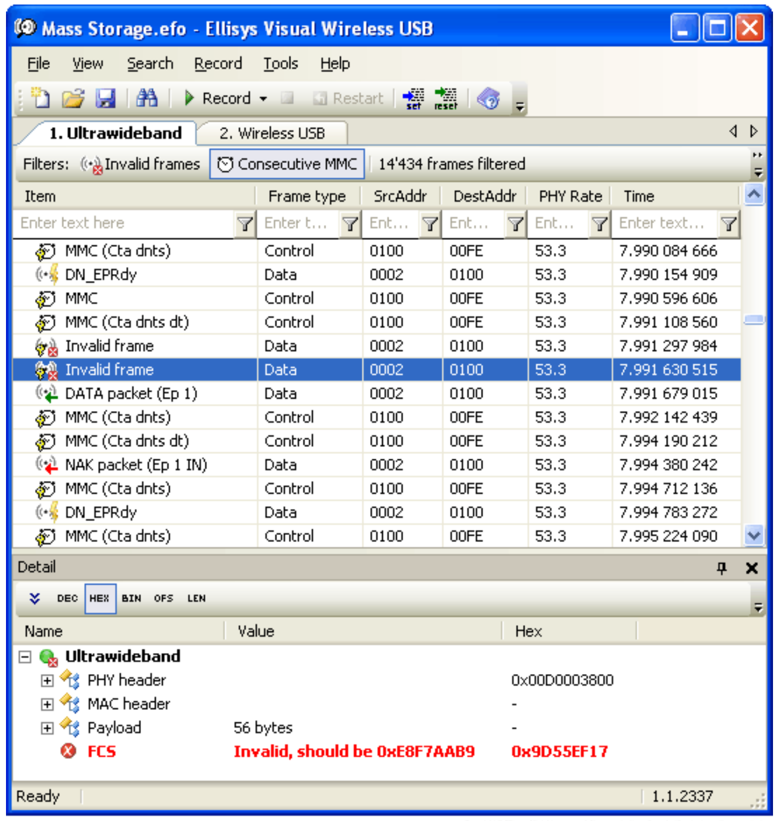

Figure 9 - An Automatically-Detected Error

> Leading protocol analyzers automatically detect and highlight errors, helping developers reduce compatibility risks.

Developers can use a frame generator to increase robustness by deliberately creating errors and ensuring that the device under test recovers properly. A frame generator, as for example the Ellisys UWB Generator 320, can play back previously recorded frames or generate specific frames on demand. It helps finding corner-case errors and ensuring proper error recovery mechanism.

Wireless USB protocol analyzers use an antenna to capture traffic and an external PC to decode the captured traffic. See Figure 10 for the setup of a Wireless USB protocol analyzer.

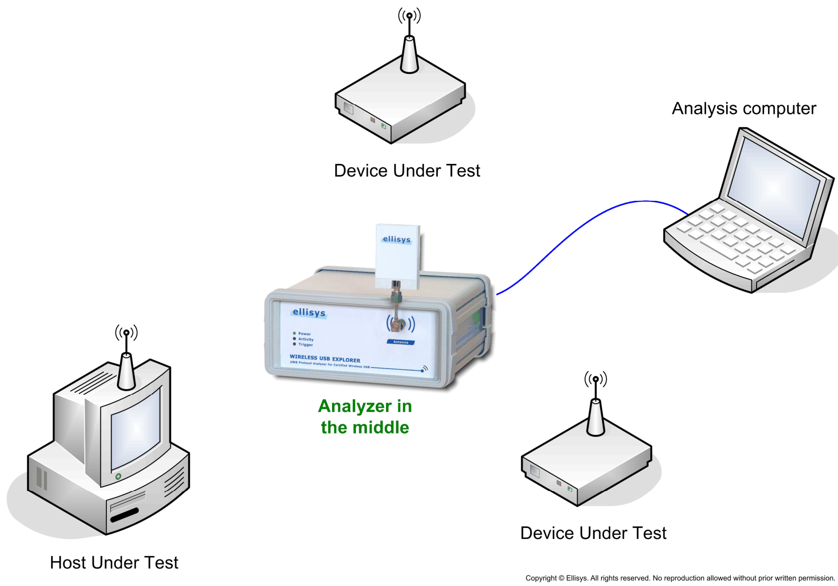

Figure 10 - Test Setup of a Wireless USB Protocol Analyzer

Shown above is an Ellisys Wireless USB 300 Explorer with the omnidirectional antenna located an equal distance from each Wireless USB device. This antenna placement provides the lowest error rates and hence best analysis performance. The analysis computer is connected with the protocol analyzer via a cable and may be located at the user's convenience.

Leading Wireless USB protocol analyzers perform extensive protocol verification to help debug interoperability issues. For example, USB Classes are high-level protocols that provide services to applications to simplify working with certain device types; classes exist for such categories as Human Interface Devices (keyboards and mice), Still Image devices (digital cameras), and Mass Storage devices (flash disks, DVD writers). Class decodes simplify working with USB classes by decoding high-level protocol requests rather than the lower-level protocol exchanges (See Figure 11).

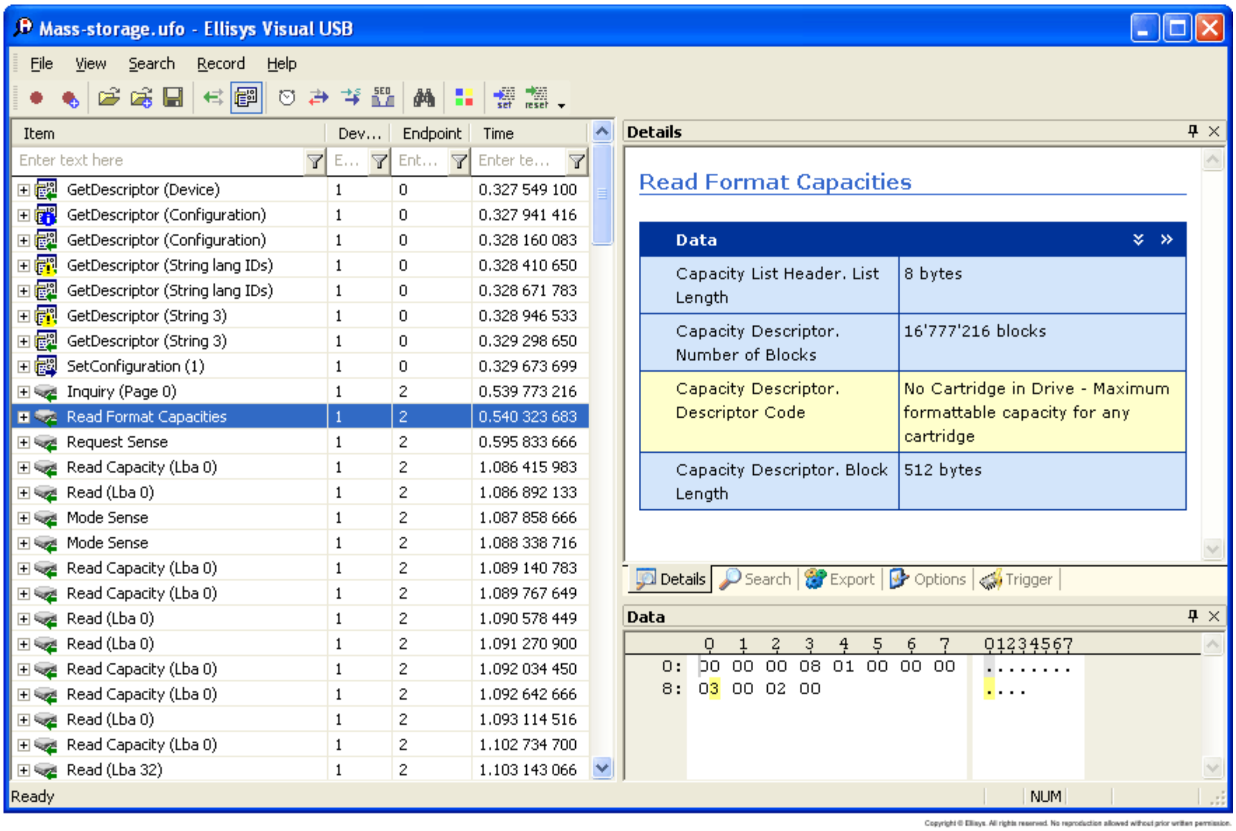

Figure 11 - USB Class Decodes

> Class decodes are productivity-boosting interpretations of higher-level protocol requests such as the "Inquiry" request for a Mass Storage device shown above. The alternative to using class decodes is to interpret the payload data contained in a sequence of multiple IN and OUT protocol exchanges.

Conclusion

As discussed above, Wireless USB is a technically-superior interface technology but its advantages are the result of significantly greater complexity when compared to wired USB. Given today's demanding customers and competing interface technologies, we believe that manufacturers can only ensure market acceptance by delivering robust Wireless USB performance in their initial products.

The quality of a Wireless USB implementation will depend on the ability to successfully balance high throughput and power. After achieving basic operation with early prototypes, these are key success factors which developers will need to address to fine-tune product performance. For example, it is entirely conceivable that a poor Wireless USB implementation will repeatedly retransmit as a result of data errors, slowing down its throughput and wasting valuable battery power.

Protocol analyzers can help ease developers' transition to Wireless USB by automatically detecting and highlighting errors for developers during initial prototyping stages, and providing productivity-enhancing high-level decode views to ensure that performance tradeoffs have been successfully implemented.

SIDEBAR: The Interface Wars

The Universal Serial Bus (USB), first introduced in 1994, quickly replaced RS-232 as the serial interface of choice for computer peripherals due its much higher speed. While RS-232 had a maximum speed of 115 kbps, USB 1.1 has a Low Speed mode of 1.5 Mbps and a Full Speed mode of 12 Mbps and USB 2.0 supports a new High Speed mode operating at 480 Mbps. USB also offers the capacity to connect multiple peripherals to a single physical port via hubs and can supply power to peripheral devices. Today, the ubiquitous USB interface dominates the serial interface market and is found in virtually all peripheral product categories, including PCs, keyboards, mice, printers, scanners, PDAs, MP3 digital music players, digital cameras, handheld GPS devices, mobile telephones and external storage devices.

The Certified Wireless USB from the USB-IF version 1.0 standard, first released in May 2005, has the goal of building on the success of the USB interface by replacing interconnecting cables with a wireless connection between peripherals and a PC. The link will operate at speeds specified as 480 Mbps at 3 meters and 110 Mbps at 10 meters. Backwards compatibility with USB will be achieved with “device wire adapters”, sometimes called “wireless hubs”. The concept is that existing USB devices may continue to be used by connecting them with a cable to a device wire adapter. In turn, the device wire adapter communicates with a Wireless USB-equipped PC. “Host wire adapters" will perform the reverse function, plugging into a PC's USB port to provide wireless connectivity to new Wireless USB devices.

Bluetooth is a competing wireless technology for short-range wireless personal connectivity and is now used in high-end mobile telephones to connect with wireless headsets and for dialing from PDAs. With a range of 1 meter for Class 1 radios, 10 meters for Class 2 radios and a data rate of 3 Mbps for Version 2.0, Bluetooth has not been considered as a serious substitute for WUSB's two-orders-of-magnitude faster data rate at similar distances. However, in March 2006 the Bluetooth SIG announced its selection of the WiMedia Alliance multiband orthogonal frequency division multiplexing (MB-OFDM) version of ultra-wideband (UWB) to create a new high-speed version of Bluetooth which might rival Wireless USB.

A third wireless technology, IEEE's 802.11 Wi-Fi, was designed for networking rather than device interconnection. However, the prevalence of commercial hotspots and sales of millions of residential Wi-Fi gateways – low-cost devices that combine an edge router, firewall, Ethernet switch and Wi-Fi radio to share a high-speed Internet connection – have prompted some manufacturers to equip products such as PDAs with Wi-Fi. This allows users not just to check e-mail and synchronize calendars, but also transfer digital photos and download MP3 music files. These latter functions start to cross into the type of functionality that Wireless USB was designed to provide.

Product Information

With the Wireless USB Explorer 300 Ellisys introduce a protocol analysis solution that captures, analyses and verifies traffic transmitted over-the-air in both the WiMedia ultrawideband and Certified Wireless USB protocols. Analysis and display software enables developers to choose a dedicated window to focus on the protocol layers of their choice. They show transfers, transactions and packets simultaneously on screen, giving them split second understanding of bus activity. Developers can use powerful search functionality to filter out unwanted data, quickly locate essential information they need to speed up their development work and accelerate time to market.

The UWB Generator 320 is the world's first frame generator for WiMedia Ultrawideband and Certified Wireless USB protocols. It helps verify product and component reliability by generating reproducible traffic, timing and error scenarios. Containing a specialized processor designed specifically for WiMedia and Certified Wireless USB protocols, the UWB Generator 320 enables you to emulate Wireless USB hosts and devices as well as various WiMedia equipments. Based on the same proven hardware as Ellisys' Wireless USB Explorer 300 protocol analyzer, the UWB Generator 320 provides a powerful new test methodology to maximize the quality of your protocol implementation and surpass your Ultrawideband R&D challenges.

The USB Explorer 200 Professional Edition is the only protocol analyzer currently on the market that decodes all USB-defined classes. USB class decoding enables embedded software, drivers and applications developers to view USB protocol subsystem layers easily, enabling them to instantly improve quality and lower production costs of their USB peripherals.

Find out more about Ellisys products at www.ellisys.com.

About Ellisys

Ellisys is a Test and Measurement company committed to the design and timely introduction of advanced protocol analysis solutions for USB devices, Wireless USB and Ultrawideband. Developers have been using Ellisys' USB products and solutions for more than five years with great success. With the consumer electronics market moving toward wireless technology Ellisys proves once again its commitment to the developers' community. By providing WiMedia and Certified Wireless USB early adopters with the right tools at the right time Ellisys enables these promising markets to grow in a secure and confident manner, and ensures a rapid and wide acceptance of these technologies. For more information, please visit www.ellisys.com.