

USB Embedded Host Performance

Yingbo Hu
Micro Digital Inc

According to the USB specification, USB full speed is 12 Mbps (Mega bits per second) and USB high speed is 480 Mbps. Hence, many people think that they can get 1.5 MBps (Mega Bytes per second) or 60 MBps, respectively, for things such as file transfers and data streaming. They are surprised to learn that real-life embedded USB performance numbers are generally far less than these theoretical limits. So, what are reasonable expectations for USB data transfer rates in embedded systems?

First, it should be recognized that 12 Mbps and 480 Mbps are raw bus speeds and the USB protocol, itself, reduces these numbers. For example, each millisecond, the USB host must send an SOF (Start of Frame) packet; for each data packet there is a PID (Packet Identifier) token; and, most of the time, an ACK/NAK packet is needed per data packet. Within the data packet, there is additional overhead for the packet header and the CRC (Cyclic Redundancy Code). The USB specification lists the theoretical data payload for each transfer type (control, bulk, interrupt and ISOC) vs. low, full and high speed. An example is shown in Figure 1.

Data Payload (KB)	Max Bandwidth (KBps)	Frame Bandwidth per Transfer	Max Transfers	Bytes Remaining	Bytes/Frame Useful Data
1	107	1%	107	2	107
2	200	1%	100	0	200
4	352	1%	88	4	351
8	568	1%	71	9	568
16	816	2%	51	21	816
32	1056	3%	33	15	1056
64	1216	5%	19	37	1216

Table 1: Transfer Rates for Full Speed Bulk Transfer Mode

As shown above, the theoretical maximum data transfer speed for full-speed bulk transfer mode (which is used for most thumb drives) is less than 1.25 MBps (10 Mbps) vs. the raw bus speed of 12 Mbps, and this requires using 64 KB data packets (which require 64 KB buffers.) For smaller packets, maximum performance is considerably less.

The USB class specification for a particular class of peripherals adds even more overhead. For example, the mass storage class uses a SCSI (Small Computer Standard Interface) command block to transfer data. For each data transfer, it is first necessary to transfer a 31-byte SCSI command block, then the real data payload, followed by a 13-byte SCSI status block. For 8KB data blocks, this adds 0.5% overhead. For 512

byte data blocks, it adds 8.6% overhead! Hence, the amount of RAM available for data blocks can have a significant impact upon USB performance.

Performance is also limited by USB devices, themselves; most are not fast enough to accept data at high speed. For example, the fastest USB hard disk we have seen achieves 25 MBps read and 20 MBps write speeds (vs. 60 MBps raw bus speed). Because thumb drives use NAND flash, extra time is necessary for wear-leveling and garbage collection. (As Murphy's Law would have it, these will occur at the worst possible times.) For example, a low-end 2 MBps flash disk may sometimes need up to 2 seconds to finish one 8 KB data write transfer because it became necessary to do and garbage collection and erase a new block to write the data. This can get worse for larger capacity thumb drives. Thus, if you are trying to data stream at 2 MBps into a thumb drive, you are going to have trouble. Hence, you need to know the inherent limitations of the USB peripherals you are using.

Ignoring peripheral limitations, if you have a 500 MHz processor, very fast and large RAM buffers, and a speed-optimized USB controller, you might come close to the theoretical maximum bandwidth, after deducting the overheads listed above. If this does not accurately describe your embedded system, read on. A typical 80 MHz embedded microprocessor is almost 40 times slower than the 3 GHz Pentium processor that we are accustomed to in a PC, and the same software runs 40 times slower on it. There is not much that can be done about this. USB, like TCP/IP, was not designed to be efficient for embedded microprocessor usage. The biggest USB overhead for typical embedded systems is introduced by the microprocessor, itself, followed by the USB controller, the RAM, and the processor bus speed. Many microprocessors are deficient in one or more of these areas. Therefore, careful attention to all four is necessary if high USB performance is to be achieved. Interrupt latency (which may be increased by other software) is also important.

Most USB host controllers require special data structures for data transfer; for example: the PTD (Philips Transfer Descriptor) for NXP ISPxxx and the Transfer Descriptor for OHCI. Consequently, the USB host controller driver must generate the required data structure before sending the packet to the USB controller. When the transfer is done, the driver must check the results of the data transfer. Hence, the USB controller, as well as the USB protocol, may impose significant overhead on each data packet transfer. Of course, it is also important whether DMA is being used to facilitate RAM to USB controller data transfers. Obviously, the speed of the processor bus and the RAM, itself, impact this.

For specific performance information, see our smxUSBH data sheets. More information is provided in the smxUSBH manual. These provide real performance data measured on real platforms. Data is provided for different microprocessors, host controllers, and USB devices. Together, this information will give you an idea what to expect for your own product. Bear in mind that no other applications were running when these measurements were made. Hence, higher priority applications running simultaneously would probably reduce USB performance further.

The following table shows some measured raw data read and write performance examples (*raw* means without any file system overhead). The device driver reads or writes 4KB of data at a time from or to the USB flash disk, for an overall transfer of 20MB. No other file operations occur.

Host Controller	Raw Reading KBps	Raw Writing KBps
EHCI (NEC)	12684	8320
OHCI (NEC)	891	832
UHCI (VIA)	639	611
ISP116x (NXP/Philips)	352	334
ISP1362 (NXP/Philips)	621	493
ISP176x (NXP/Philips)	7425	3214

The hardware environment for this testing is: Celeron 300MHz CPU; 32MB 100M SDRAM; PC motherboard; Host Controller connects to System by 33MHz PCI bus. Flash Disk is Lexar JumpDrive USB 2.0 512MB. The EHCI and ISP176x are high speed controllers; the rest are full speed controllers.

Table 2: Measured Raw Data Performances

Contact me if you have USB performance questions.
 Yingbo Hu, yingbohu@smxrtos.com