

Overview

There are an overwhelming number of IO standards that exist in the world and often times it is difficult to determine if a given standard is best suited for a particular application. This application note discusses the benefits and limitations of using QuickUSB to add a USB interface to a given design.

Benefits of Using QuickUSB

The Power of USB

USB is one of the most popular computer interfaces in the world with over a billion USB peripherals sold each year. Choosing to use USB in a design means that nearly any computer can interface to your hardware without anything except a USB cable. Additionally, USB 2.0 can transfer data at an astounding theoretical rate of 480 Mbit/s, as well as provide as much as 500 mA @ 5 V of power to alleviate the need to design separate power supplies in some designs.

Proven design with Multi-Platform Support

Using QuickUSB to implement your USB 2.0 interface means that with little to no knowledge of the USB 2.0 Specification you may create a design that will work without needing to write any firmware or driver software. Additionally, QuickUSB is fully supported on Windows, Linux, and Mac. Using the platform-independent QuickUSB Software API, you can create end-user software that will run on all supported platforms without needing to know the intimate details of each platform.

Using the proven design of QuickUSB means that you can be sure that your hardware will work the first time and that you will not spend countless hours trying to debug complicated USB issues. This allows you to focus on what is most important in your design instead of the data interface.

Additional IO Ports

QuickUSB is a lot more than a USB port—it offers a number of additional features that mean less design and less hardware in your final product. The additional IO Ports includes: two RS-232 ports, an I2C port (100 or 400 kHz), a SPI port with 10 slave-select lines, and a number of general-purpose IO pins. QuickUSB also contains 2 KB of EEPROM for general use and is capable of configuring Altera and Xilinx FPGAs, as well as many serial configuration devices such as Altera's EPCS devices.

Limitations of QuickUSB

USB Peripheral

QuickUSB is a USB peripheral. This means that QuickUSB must be connected to a USB Host, such as a PC. Furthermore, QuickUSB is only supported on 32-bit and 64-bit versions of Windows (XP, Vista, and 7), Linux (Kernel v2.6.25 or later), and Mac OS X (v10.6 or later). If your design needs to act as a USB Host or plug into a host running an unsupported operating system then QuickUSB is not suited for your design.

Data Throughput

The theoretical maximum data rate of USB is 480 Mbit/s but, in actuality, the data rates you achieve are far less. This is due to overhead of the USB protocol, performance of the USB driver stack, hardware

implementation, and software latencies in the user application, operating system, and driver stack. Typically, a well-designed QuickUSB application can offer 20 to 35 MB/s of sustained data transfer. If your design requires data rates of over 35 MB/s then QuickUSB is not suited for your application. Additionally, if your design requires data rates that really exercise the full power of QuickUSB (25 MB/s or more) then you must take extra care in your design to make sure that you can achieve such data rates. Even designs that only require low data rates must be designed properly.

Data rates are a large function of request size—that is, you get higher data rates when you request reads or writes for large amounts of data as opposed to small amounts. Data rates also depend on the IO Model you are using with QuickUSB, software implementation, and, in some cases, the depth of data buffers. Please consult the additional QuickUSB Application Notes on how to more accurately determine the expected data throughput for your application to determine if QuickUSB can meet your needs.

USB Device Classes

QuickUSB only enumerates as a custom USB device and does not support any of the other USB Device Classes such as HID, Video, Audio, Mass Storage, etc. This means that if your application requires your hardware to enumerate as a webcam, audio device, mass storage device, or any other class of USB devices then QuickUSB is not suited for your application. It is also important to note that the two RS-232 ports on QuickUSB do not map to native RS-232 COM ports on a computer and are only accessible through the QuickUSB Software API. This means that QuickUSB may not be used to create USB-to-serial adapters whose RS-232 are accessible as COM ports.

USB Endpoints

The USB 2.0 Specification defines four types of endpoints that a device may implement: bulk, isochronous, interrupt, and control. QuickUSB enumerates with one control endpoint, one bulk IN endpoint, and one bulk OUT endpoint. The control endpoint is used for control transfers, such as I2C, RS-232, SPI, and GPIO. The bulk endpoints are used for high-speed data transfer. If your design requires isochronous or interrupt endpoints then QuickUSB is not suited for your design.

Custom Firmware and Driver

QuickUSB is designed to be easy and does so by hiding the complexity of USB in proprietary firmware, drivers, and software library. Therefore, if your application requires features that are not built into the current QuickUSB firmware, drivers, or software library then QuickUSB may be unsuited for your design. Bitwise Systems does, however, provide contract services so if QuickUSB is a match for your design except for a specific feature requiring changes in the QuickUSB firmware, driver, or library please contact support@quickusb.com for an RFQ on your specific application requirements.