



870 Maude Avenue  
Sunnyvale, California 94085  
Tel (408) 774-9060  
Fax (408) 774-2169

E-mail: [www.plxtech.com/contacts](http://www.plxtech.com/contacts)  
Internet: [www.plxtech.com/netchip](http://www.plxtech.com/netchip)

## PLX USB Development Kit

PLX Technology's USB development kit (PCI-RDK) is a complete development solution helping engineers to create USB 2.0 device firmware, host drivers and applications.

The primary goal is to provide the engineer with a stable hardware and software platform for developing firmware for PLX's USB peripheral controllers such as the NET2272 and NET2280. In many cases, firmware, drivers and applications can be fully developed and tested before the developer's hardware is complete. This means quicker development time and faster time to market.

With the PCI-RDK installed, the Windows PC becomes a USB firmware development platform. The Windows PC is essentially "transformed" into a USB 2.0 device with extensive debug and analysis capabilities. Emulation of almost any USB device can be accomplished with the PCI-RDK. It is not a simulation.

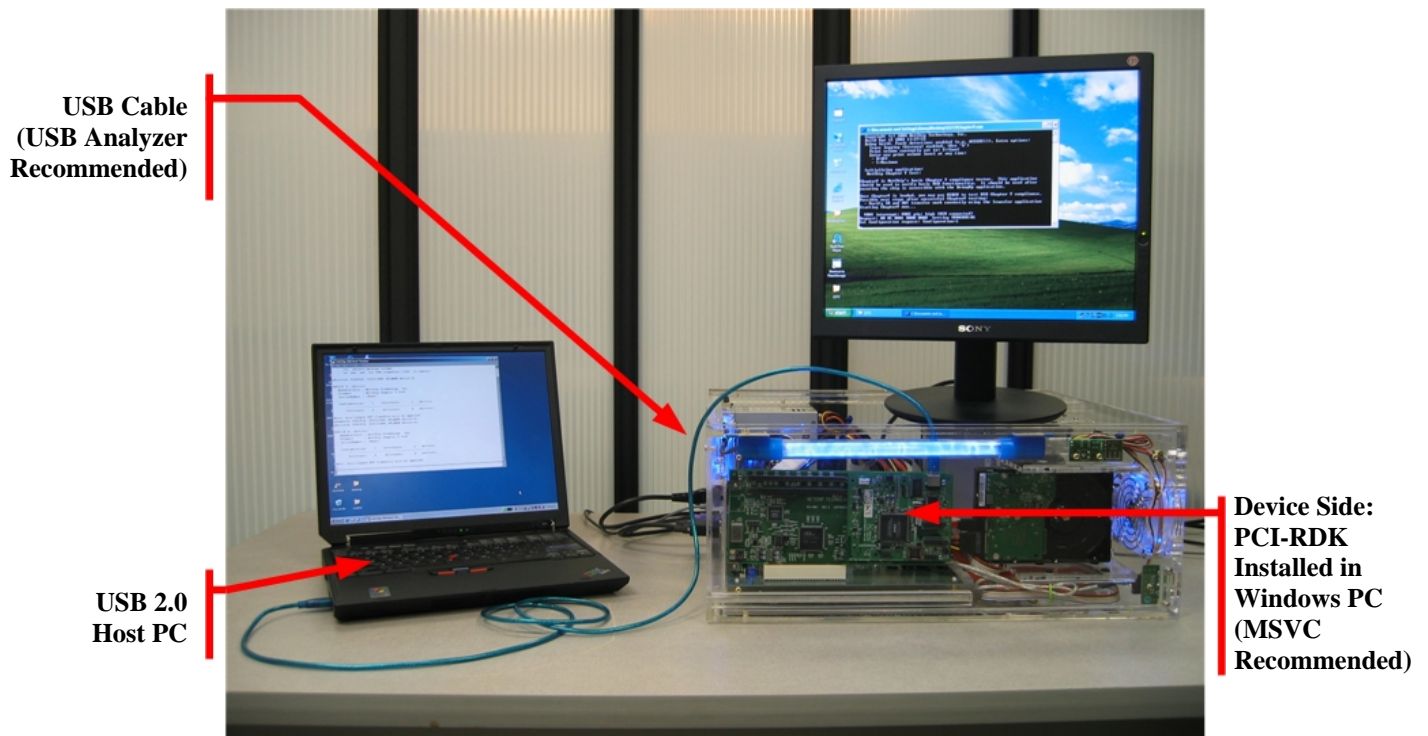


Figure 1. Typical PCI-RDK development environment

## With the PCI-RDK it is possible to:

- Build a working model on the PCI-RDK:
  - Firmware is fully portable
  - Prove firmware before device hardware is fully operational
  - Trouble-free hardware, so developer can focus on firmware
  - Conceptual experimentation is easy
- Run pre-compiled firmware examples to experiment with USB transfers:
  - Firmware runs on a Windows PC as a standard ‘console application.’
  - Special driver allows direct access to registers; abstracts interrupts, etc.
  - Firmware developers can use popular tools (e.g. MSVC) to recompile, test, and customize included firmware to build working device models.
- Simplify software development using included Firmware API (Application Programming Interface):
  - Chip, interrupt, and other details are hidden within API.
  - All applications are chip independent
  - Utilize a standard set of functions that allow full control of USB
- Simplify software debugging with trace logging, PLX’s built-in firmware analyzer:
  - Analyze trace logs on unexpected events and eliminate guesswork.
  - Works on PCI-RDK, portable to target platform
  - E-mail trace logs to PLX for expert advice
- Become familiar with PLX’s USB peripheral chip register set using provided register monitor software. (see also USB Software Support)

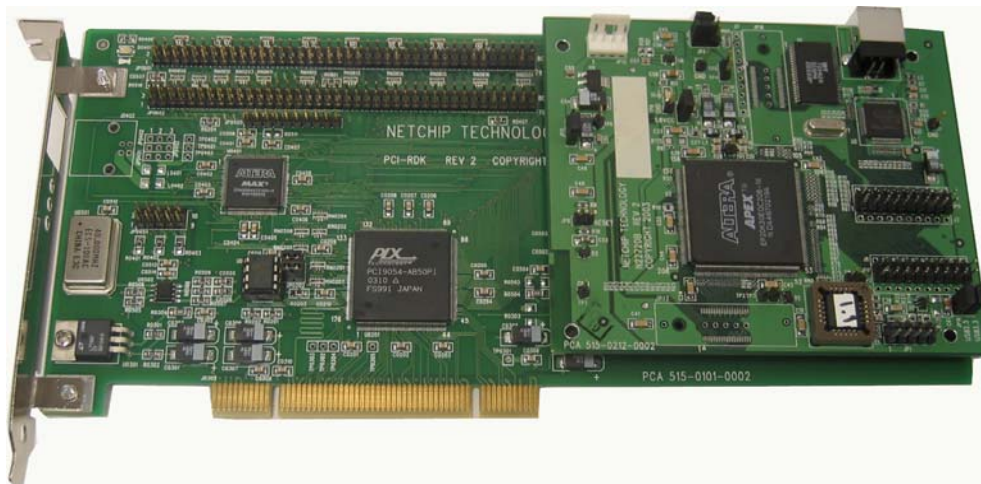


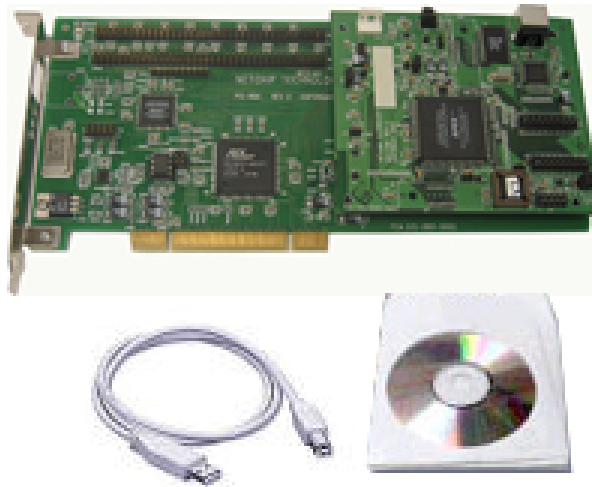
Figure 2. PCI-RDK development board

## What is included in the PCI-RDK kit?

The kit includes operational hardware, software and source code for development of USB devices based on PLX's family of USB peripheral controllers.

### PCI-RDK package contents:

- PCI board
- Device development card (mounted to PCI board)
- CD-ROM with PLX software suite, tools, and documentation
- Hi-Speed USB 2.0 Compliant cable



**Figure 3. Complete PCI-RDK development kit**

## PCI board: What's on it?

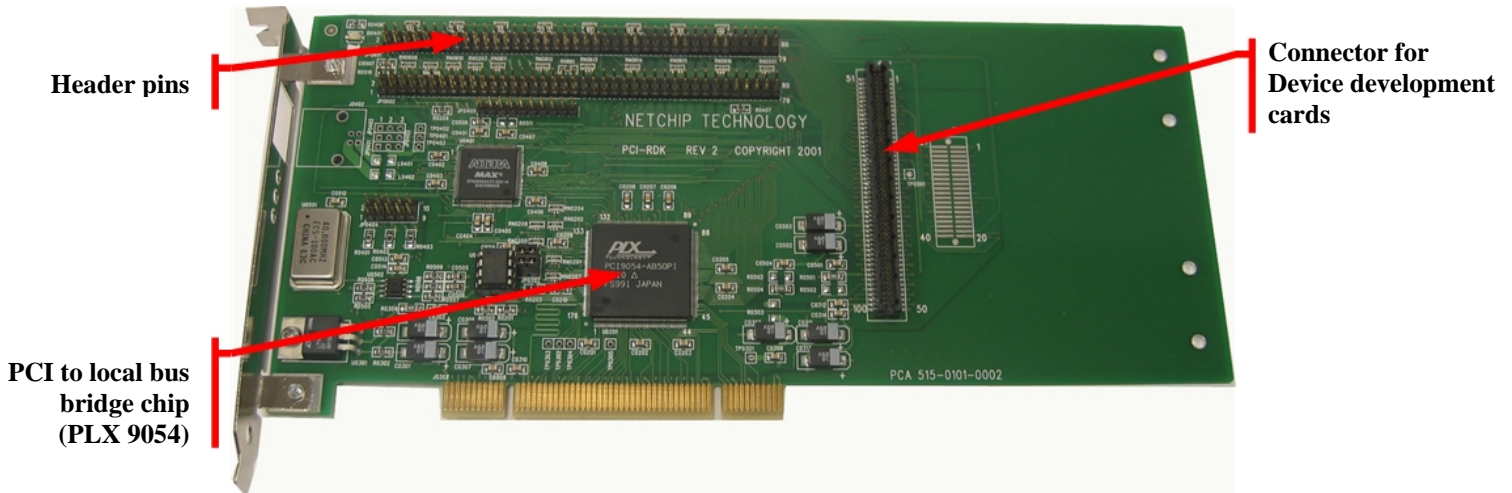


Figure 4. PCI-RDK Development board (without development card)

The PCI board interfaces the device development cards to the PCI bus. PCI board component highlights:

- PCI to local bus bridge chip (PLX 9054)
- Connector for the device development card
- Header pins (so that all local bus device signals can be probed)

## Device development card: What's on it?

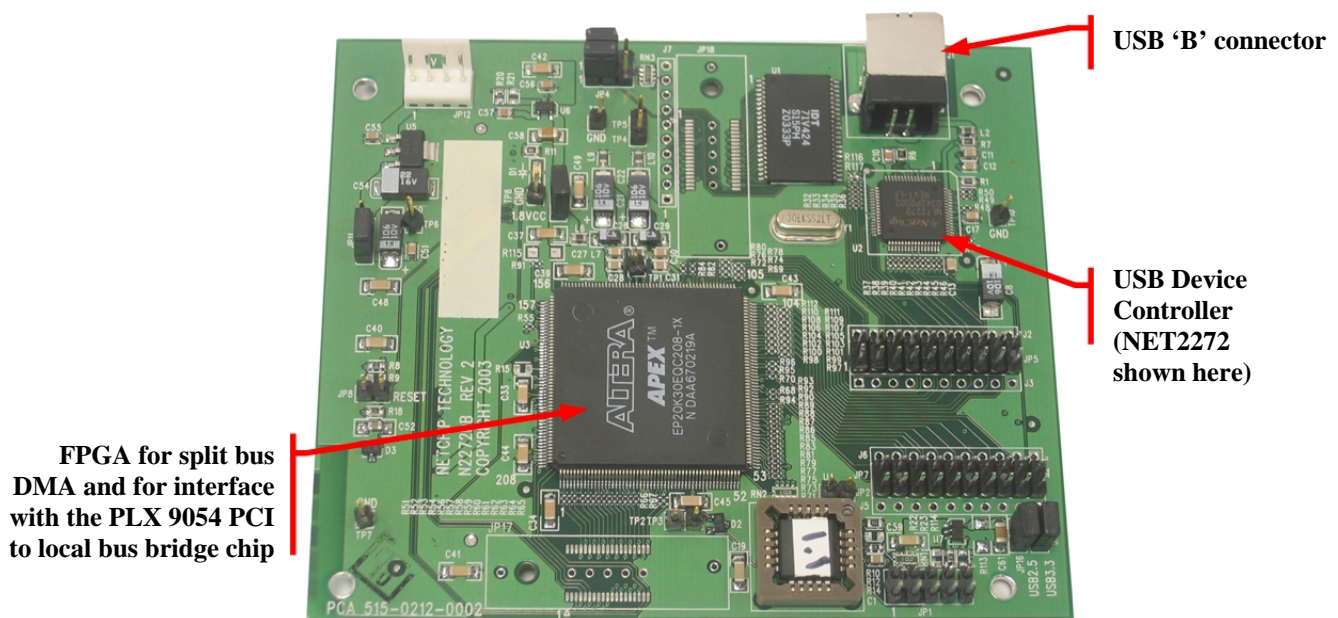


Figure 5. Device development card

## CD-ROM: What's on it?

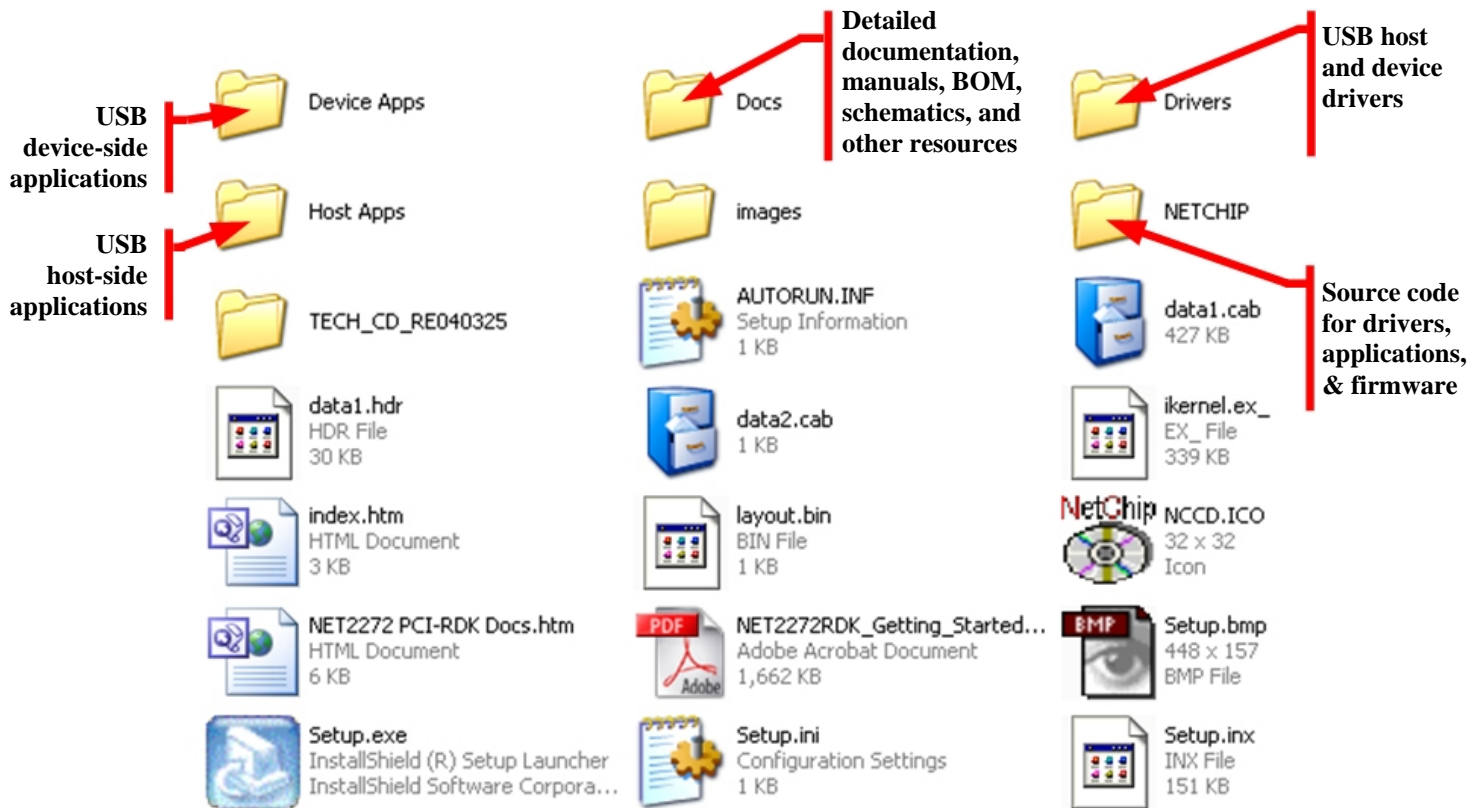


Figure 6. Actual CD-ROM contents of PCI-RDK

## USB Software Support

### Device Firmware Support

PLX offers an extensive library of example device projects. At the core of these projects is the Firmware API. The Firmware API is a chip-independent, self contained set of structures and function calls that encapsulate and abstract the USB device controller chip and its programming. The firmware clients, such as Loopback or MassStorage (described below), can be applied to either the NET2272 or the NET2280 by simply utilizing the appropriate core firmware for the selected chip. Furthermore, once the core firmware has been ported to the target platform, porting any of the applications in the device project library is simple and straightforward. To aid in the whole process, all sources are ANSI C compliant. And, extensive debugging support, including firmware trace logging, has been built into the firmware.

The Firmware API handles all the difficult aspects of USB programming while allowing the client full control of the device. Endpoints can be set to any type or direction and transfers can be of any size. Standard control requests on Endpoint 0 and USB Specification Chapter 9 compliance are handled by the core firmware but standard requests can be filtered and processed by the client if needed. Performance-tuned transfers are driven by programmed I/O or DMA. The core firmware handles all chip interrupts due to transfer activity or USB bus state changes.

## Prototype Test and Development Devices

Use the following device projects to prove the functionality and reliability of the target platform's board design and layout. Each project iteratively adds a small amount of complexity while marking a significant development milestone.

- **Bringup:** Use Bringup on new a new platform *before* connecting it to a USB host. Bringup verifies chip registers, interrupts, and some basic chip operations; it prints helpful messages when subsystems are not working as expected. When Bringup is connected, it prints out the interrupts it detects, but Bringup does not handle them. Once the platform passes all Bringup tests reliably, it is time to apply the first application for connecting to a USB host, **Chapter 9**.
- **Chapter 9:** PLX's first-time-connection application, Chapter 9 passes USB Chapter 9 compliance (but not much more). Chapter 9 is an excellent module to run on a new target platform when connecting to a USB host for the first time. (Note: *All* PLX application modules pass Chapter 9 compliance; the difference is that Chapter 9 is minimal, and doesn't do anything *except* pass Chapter 9 compliance.)
- **Transfer:** After applying Chapter 9 to verify basic Endpoint Zero exchanges, enumeration and general USB capabilities, Transfer should be used to test data transfers on data endpoints. Transfer is simple and flexible. It is non-specific, meaning it can be tailored to meet virtually any requirement. Transfer is the foundation for all PLX applications.
- **Loopback:** Use Loopback to "stress test" a device platform. Loopback endlessly loops back transfers from OUT endpoints to IN endpoints. Transfer size and content is specified by the host; both can be completely arbitrary. PLX provides a companion host-side Windows loopback application specifically designed for use with loopback firmware.

## Class Devices

Many peripherals fall under specific categories in regards to features and functionality. For instance, printers all provide the same basic function and generally talk to PCs in a similar manner. When the demand for a certain category is large enough, device and OS manufacturers collaborate to standardize the protocol of such devices over USB. This allows device manufacturers to concentrate on designing and building gadgets and gizmos while OS manufacturers focus on writing the drivers for the device class. PLX offers the following device class examples that can be ported to your platform and extended to your requirements. All of these projects have drivers that come standard in Windows XP (unless otherwise stated). See [http://www.usb.org/developers/devclass\\_docs](http://www.usb.org/developers/devclass_docs) for the specification of each class.

- **MassStorage:** MassStorage emulates a USB key drive utilizing the USB Mass Storage Class. With MassStorage running on the device PC, the USB Host will immediately recognize the device when connected and load the MassStorage class driver. The MassStorage project simply uses a block of system memory to emulate the storage medium. Read and write commands that come from the host are performed on this block of memory.

- **Audio:** Audio implements a USB speaker capable of outputting 16-bit, 44.1 kHz audio on a single channel. The USB host streams any sound to the device. If a sound card is present on the device PC, the audio will be outputted to that sound card. Audio uses a single isochronous endpoint to receive the audio stream.
- **Video:** Video mimics a USB web cam. The Video class driver was released with Windows XP SP2. Video can stream a series of jpg files to create an animation.
- **Printer:** Printer implements a generic text-only printer. Simple text files can be printed to the device. All text data is sent to the console of the Printer project.
- **Remote NDIS:** Remote NDIS, or RNDIS, is a sub-class of the USB Communications Device Class. It provides a standardized protocol for networking over USB. The USB host recognizes PLX's RNDIS example as an RNDIS device. However, since the example is a console application, it has no means to route network packets it receives from the Host nor the ability to register itself as a network interface on the device PC.
- **Still Image Class:** Still Image Class, which is also known as Picture Transfer Protocol, applies to imaging devices such as digital cameras and scanners. When connected, this example automatically pops up the Windows "Scanner/Camera Wizard" which allows images to be downloaded from the device.
- **Keyboard:** Keyboard emulates a standard USB Keyboard. Keys typed to Keyboard's console are sent to the host as key presses.
- **Mouse:** Mouse emulates a standard USB Mouse. Use the arrow keys in Mouse's console to move the mouse pointer on the host.

## ***USB Host Software Support***

Not all USB devices fit within a certain USB class. Or some may require functionality that the closest matching class cannot provide. For customers that need to implement a proprietary solution, PLX provides a generic driver and application set.

- **NcBulk:** NcBulk is a general purpose, WDM, USB client driver. It provides a standard file I/O interface that allows user applications to initiate transfers. In most cases, NcBulk can be used without modification. In some instances, developers may choose to simply incorporate NcBulk's USB specific portions to their drivers.
- **SimpleHost:** SimpleHost provides a source example for interfacing to NcBulk. The complexity of its graphical user interface was minimized in order to focus on the USB aspects of the application. SimpleHost utilizes an API that is similar to the FirmwareAPI from the device side. Through the Host API, SimpleHost demonstrates:
  - How to detect and handle device connection and removal
  - How to initiate USB transfers
  - How to cancel transfers

- **WinMON:** WinMON is a test utility used for testing and development. With WinMon’s flexible console interface, arbitrary length transfers can be performed on any endpoint, transfer buffers can be peeked and poked, and almost any sequence of USB traffic can be generated.
- **LoopHost:** LoopHost is Loopback’s host side companion. LoopHost can increment or randomize its transfer size at each iteration. The data that LoopHost sends can be a random buffer, an incrementing pattern, or a blank buffer. Overnight tests with LoopHost ensure solid reliability throughout the system on both the host and device.

## Software Architectural Layer Model

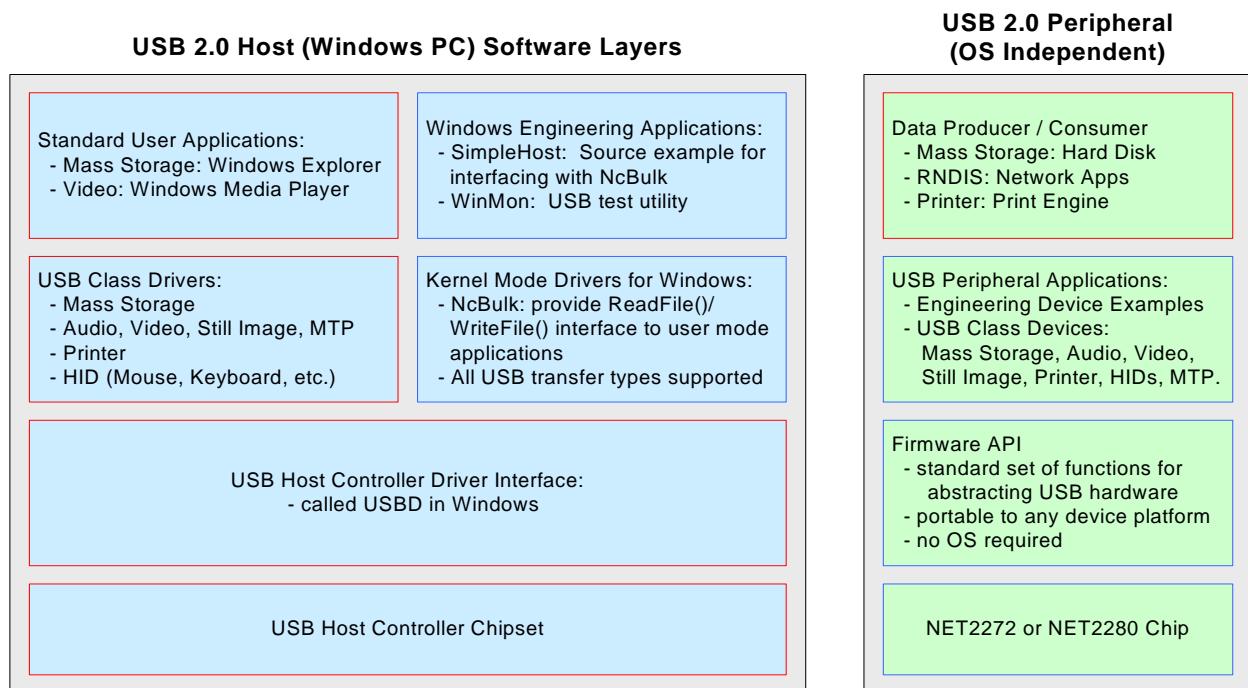


Figure 7. Architectural software layers

## Using the USB 2.0 PCI-RDK in a development environment

### How can I get the most out of the PCI-RDK?

Here is one suggested sequence for getting the most out of the PCI-RDK:

1. Use the PCI-RDK to familiarize yourself with the PLX USB device controller.
2. Run precompiled firmware demos with USB host monitor applications. (Class device demos do not require USB host monitor applications)
3. Recompile firmware, and run under debugger (MSVC).
4. Explore firmware using MSVC breakpoints and single step features
5. Customize firmware to create a USB device interface that matches your specifications
6. Port your working PCI-RDK device model to your own hardware

### ***Additional Notes:***

1. Fast engineering response to technical questions: One business day
2. Customers can receive from PLX a complete schematic and layout verification of the USB portion of their design.
3. Customers can also receive USB-IF Logo Certification compliance assistance. (Contact PLX Sales for more information.)
4. The PCI-RDK development board is designed only for development purposes. It is NOT suited for bandwidth performance measurement or comparisons.
5. PLX porting guides are included with the PCI-RDK to assist in the porting from the development environment to the target platforms.

### ***Questions?***

#### **For Technical Support or Inquiries:**

If you have any technical questions or concerns please feel free to contact us at:

[www.plxtech.com/support](http://www.plxtech.com/support)

#### **Ordering and Sales information:**

If you like to know how to purchase the PCI-RDK development kit please contact us at:

[ww.plxtech.com/contacts](http://www.plxtech.com/contacts)