Introduction

This book is for developers who design and program devices that use the Universal Serial Bus (USB) interface. My goal is to introduce you to USB and to help you get your devices up and communicating as quickly and easily as possible.

The USB interface is versatile enough for a wide range of peripheral devices. Standard peripherals that use USB include mice, keyboards, drives, printers, and audio/video devices. USB is also suitable for data-acquisition units, control systems, and other devices with specialized functions, including one-of-a-kind designs.

To develop a device with a USB interface, you need to know something about how the interface works, what tasks your device firmware must perform to communicate on the bus, and what class drivers and other support are available on the host computers that your device will attach to. The right choices of device hardware, device class, and development tools and techniques can go a long way in avoiding snags and simplifying what needs to be done.

If you're involved with designing USB devices, writing the firmware that resides inside USB devices, or writing applications that communicate with USB devices, this book will help you along the way.

What's Inside

These are some of questions the book answers:

- *How do USB devices communicate?* The USB interface can seem daunting at first. The USB 2.0 specification is over 600 pages, not counting the class specifications and other supplementary documents. This book doesn't attempt to restate everything in the specifications. Instead, the focus is on what you'll need to know to enable your devices to communicate efficiently and reliably.
- *How can I decide if my device should use a USB interface?* USB isn't the best choice for every application. Find out whether your design should use USB or another interface. The chances are good that you will choose USB, however, and if so, you'll learn how to decide which of USB's three speeds and four transfer types are appropriate for your application.
- *What controller chip should my device use?* Every USB device must contain an intelligent controller to implement the USB interface. Dozens of manufacturers offer controller chips with differing architectures and abilities. This book includes descriptions of popular chips and tips to help you select a controller based on your project's needs and your background and preferences.
- *How do applications communicate with USB devices?* PC applications access a USB device by communicating with the device driver the operating system has assigned to the device. Some devices can use class drivers that are included with Windows. Others devices require custom drivers. This book will introduce you to the classes and will help you determine if a defined class is appropriate for your device. If your device requires a custom driver, you'll learn what's involved in writing a driver, what tools

can help speed up the process, and options for obtaining drivers from other sources. Example code shows how to detect and communicate with devices in Visual Basic .NET and Visual C++ .NET applications.

- *What firmware does my device need to support USB communications?* Learn how to write device firmware that enables your device to respond to received requests and exchange other data on the bus.
- *How do I decide whether my device can use bus power or needs its own supply?* Many USB devices can be powered entirely from the bus. Find out whether your device can use bus power. Learn how to ensure that your device meets USB's requirement to limit the use of bus current when the host computer suspends the bus.
- *Can I connect other USB peripherals to my device?* Find out how to use USB On-The-Go to enable your device to act as a limited-capability host that can access other USB peripherals.
- *How can I ensure that my device will communicate without problems?* At the device, writing bugfree firmware requires understanding what your device must do to meet the requirements of the USB specifications. At the host computer, Windows must have the information needed to identify the device and locate a driver to communicate with the device. This book has tips, example code, and information about debugging software and hardware to help with these tasks.

To understand the material in the book, it's helpful to have basic knowledge in a few areas. I assume you have some experience with digital logic, application programming for PCs and writing embedded code for peripherals. You don't have to know anything at all about USB.

What's New in the Third Edition?

Since the publication of *USB Complete Second Edition*, much has happened in the world of USB. Additions to the USB specifications include many updated and expanded device-class specifications and the USB On-The-Go supplement. Many new device-controller chips have been released. New tools for debugging and compliance testing are available. Support for USB device classes under Windows has improved. And Microsoft's .NET Framework has become a popular platform for developing host applications.

These developments prompted me to write *USB Complete Third Edition*. The material is revised and updated from start to finish to reflect these and other developments related to USB hardware and programming.

More Information, Updates, and Corrections

To find out more about developing USB devices and the software that communicates with them, I invite you to visit my USB Central page at Lakeview Research's Web site (*www.Lvr.com*). You'll find code examples and links to articles, products, tools, and other information related to developing USB devices. If you have a suggestion, code, or other information that you'd like me to post or link to, let me know at *jan@Lvr.com*.

Corrections and updates will also be available at *www.Lvr.com*. If you find an error, please let me know and I'll post it.

Acknowledgements

USB is too big a topic to write about without help. I have many people to thank.

I owe a big thanks to my technical reviewers, who provided feedback that has greatly improved the book. (With that said, every error in this book is mine and mine alone.)

Thanks first to Paul E. Berg, MCCI Vice President, Architecture and USB-IF Device Working Group Chair. Thanks also to David Goll of the USB-IF's Video Device Working Group, Lucio DiJasio and Rawin Rojvanit of Microchip Technology, John Hyde of *usb-by-example.com,* Geert Knapen of the USB-IF's Audio Device Working Group, Walter Oney of Walter Oney Software, and Marc Reinig of System Solutions.

Others I want to thank for their support are Glenn M. Roberts of Cypress Semiconductor, Fred Dart and Keith Dingwall of FTDI Chip, Wendy Dee of Keil Software, Michael DeVault of DeVaSys Embedded Systems, Alan Lowne of Saelig Company Inc., Laurent Guinnard of Ellisys, Rich Moran of RPM Systems Corporation, and Bob Nathan of NCR Corporation.

For help and support with the previous editions that this edition builds on, thanks to Joshua Buergel, Gary Crowell, Dave Dowler, Mike Fahrion, John M. Goodman, Lane Hauck, David James, Christer Johansson, Kosta Koeman, Jon Lueker, Brad Markisohn, Amar Rajan, Robert Severson, Craig R. Smith, and Dave Wright.

I hope you find the book useful. Comments invited!

Jan Axelson
jan@Lvr.com