**Intel
University
Press**

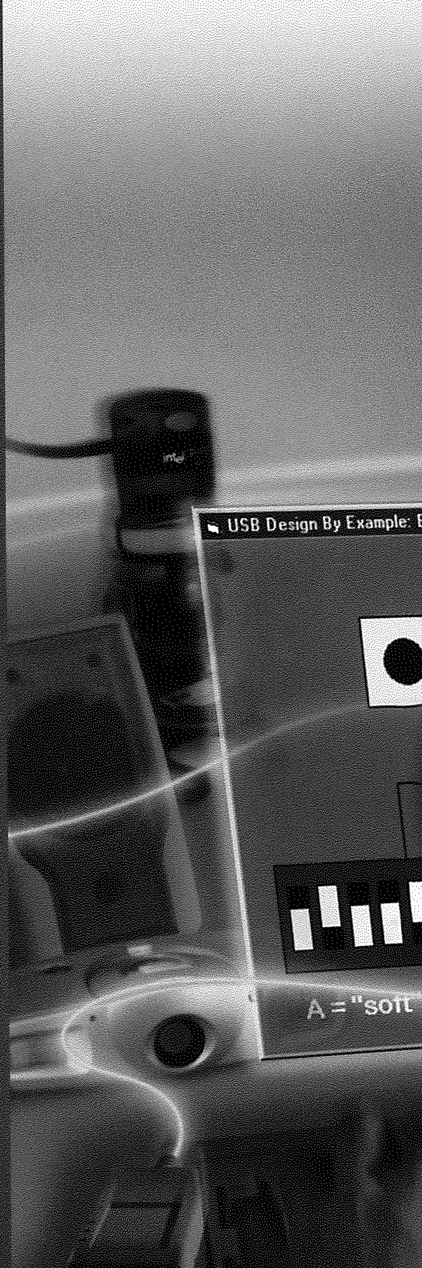
*The PC Platform
Designers' Choice*

intel

USB Desi

A Practical Guide

John Hyde



USB Design by

A Practical Guide to Buildin

John Hyde

Wiley Computer Publish

WILEY
John Wiley & Sons, Inc
New York • Chichester • Weinheim • Brisban

Publisher: Robert Ipsen (Wiley), Rich Bowles (Intel)
Acquisitions Editor: Cary Sullivan
Editor: Marcia Petty
Assistant Editor: David Spencer
Managing Editor: Frank Grazioli
New Media, Associate Editor: Mike Sosa
Text Design & Composition: Marianne Phelps
Graphic Art: Jerry Heideman (illustrations), Dan Mandish (cover)

Copyright © 1999 Intel Corporation. All rights reserved.

Published by John Wiley & Sons, Inc. Published simultaneously in Canada.

No part of this publication may be reproduced, stored in a retrieval system or transmitted in any form or by any means, electronic, mechanical, photocopying, recording, scanning or otherwise, except as permitted under Sections 107 or 108 of the 1976 United States Copyright Act, without either the prior written permission of the Publisher, or authorization through payment of the appropriate per-copy fee to the Copyright Clearance Center, 222 Rosewood Drive, Danvers, MA 01923, (978) 750-8400, fax (978) 750-4744. Requests to the Publisher for permission should be addressed to the Permissions Department, John Wiley & Sons, Inc., 605 Third Avenue, New York, NY 10158-0012, (212) 850-6011, fax (212) 850-6008, E-Mail: PERMREQ @ WILEY.COM.

This publication is designed to provide accurate and authoritative information in regard to the subject matter covered. It is sold with the understanding that the publisher is not engaged in professional services. If professional advice or other expert assistance is required, the services of a competent professional person should be sought.

All information provided in this publication is provided "as is" with no warranties, express or implied, including but not limited to any implied warranty of merchantability, fitness for a particular purpose, or non-infringement of intellectual property rights. Any information provided related to future Intel products and plans is preliminary and subject to change at any time without notice.

Intel Corporation may have patents or pending patent applications, trademarks, copyrights, or other intellectual property rights that relate to the presented subject matter. The furnishing of this publication does not provide any license, express or implied, by estoppel or otherwise, to any such patents, trademarks, copyrights, or other intellectual property rights.

Designations used by companies to distinguish their products are often claimed as trademarks. In all instances where John Wiley & Sons, Inc., is aware of a claim, the product names appear in initial capital or all capital letters. Readers, however, should contact the appropriate companies for more complete information regarding trademarks and registration.

This book is printed on acid-free paper. ∞

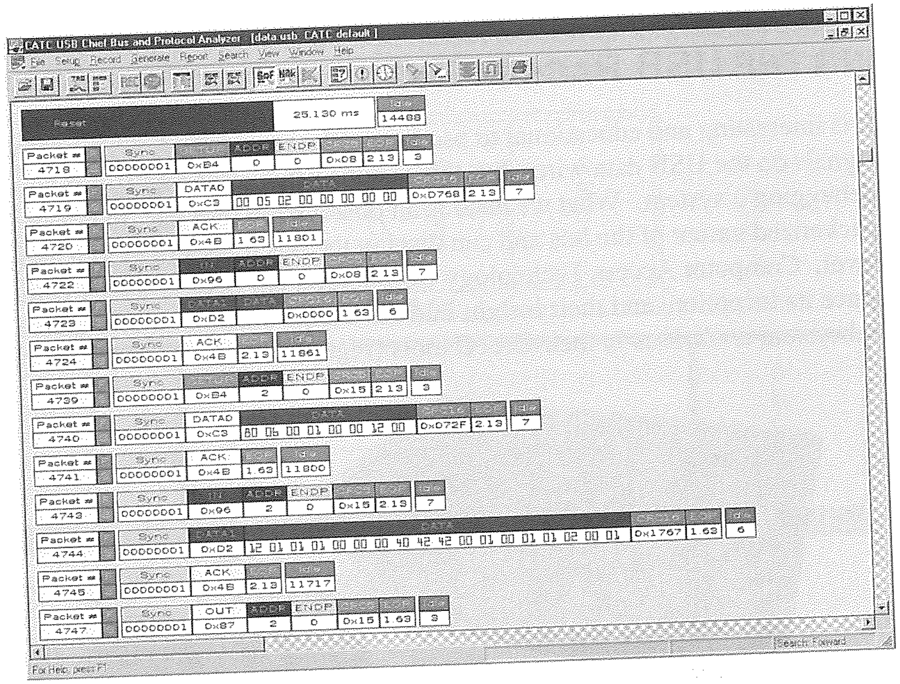
Library of Congress Cataloging in Publication Data:

Hyde, John, 1952–
    USB design by example : a practical guide to building IO devices / John Hyde.
        p.   cm.
    Includes index.
    ISBN 0-471-37048-7
    1. USB (Computer bus) 2. Computer input-output equipment—Design and construction. I. Title.
TK7895.B87H93 1999
004.6'4—dc21   99-35865
                                                                CIP

Printed in the United States of America

10 9 8 7 6 5 4 3 2

DOCKET ALARM Find authenticated court documents without watermarks at docketalarm.com.



```
Reset                 25.130 ms   14488
Packet # 4718   Sync 00000001   0xB4   ADDR 0   ENDP 0   0x08   2.13   3
Packet # 4719   Sync 00000001   DATA0 0xC3   00 05 02 00 00 00 00 00   0xD768   2.13   7
Packet # 4720   Sync 00000001   ACK 0x4B   1.63   11801
Packet # 4722   Sync 00000001   0x96   ADDR 0   ENDP 0   0x08   2.13   7
Packet # 4723   Sync 00000001   0xD2      0x0000   1.63   6
Packet # 4724   Sync 00000001   ACK 0x4B   2.13   11861
Packet # 4739   Sync 00000001   0xB4   ADDR 2   ENDP 0   0x15   2.13   3
Packet # 4740   Sync 00000001   DATA0 0xC3   80 06 00 01 00 00 12 00   0x072F   2.13   7
Packet # 4741   Sync 00000001   ACK 0x4B   1.63   11800
Packet # 4743   Sync 00000001   0x96   ADDR 2   ENDP 0   0x15   2.13   7
Packet # 4744   Sync 00000001   0xD2   12 01 01 01 00 00 00 40 42 42 00 01 00 01 01 02 00 01   0x1767   1.63   6
Packet # 4745   Sync 00000001   ACK 0x4B   2.13   11717
Packet # 4747   Sync 00000001   OUT 0x87   ADDR 2   ENDP 0   0x15   1.63   3
```

Figure 2-17. USB packets displayed as packets

All of the CATC tools capture bus activity for later analysis. The use of color to display different packet types and the grouping of building-block packets into transactions allow the designer to quickly interpret what has happened on the bus. The simpler CATC tools capture all bus activity while the more elaborate tools have programmable capture and programmable triggers. Being able to isolate packets sent to a specific device or triggering after, for example, device 42 has received 58 DATA0 packets, aids the debugging of more sophisticated USB devices. The high-end CATC tools can also be used to generate bus traffic for device reliability testing and failure analysis.

HAPTER SUMMARY

This chapter provided insight into the signals on the bus, the fundamental packetized nature of the bus, and the transactions used to exchange data on the bus. The PC host uses a defined set of requests to control all of the devices attached to the bus, and these devices need to respond in a defined manner. Bus observation, or "sniffer" tools as they are called, are available to monitor and analyze these low-level bus signals.

THE ENUME

Let us assume that the PC host meets all of the
Chapter 1, is running a USB-aware operating s
port. This port could be on the PC host itself (
an external hub. Now we have a new USB I/C
running system. What actually happens betwe
to deliver the many USB features?

After understanding what the PC host is doing
general I/O device. All devices describe thems
tables. We start by looking inside the simplest
discussion to cover the general case. We then
requirements for a device.

There are many "chicken-vs.-egg" situations ir
some technical discussions to keep the flow of

VICE DETECTION

Figure 3-1 shows details about the USB cable. The cable has four wires: two power wires for Vcc and Gnd and two signal wires for D+ and D–. The cable end that attaches to the hub has a Series A connector, and the cable end that attaches to the new device is either connected directly (no connector) or has a Series B connector. Both connectors have longer power and ground connector pins to ensure that the device has good voltages before signals are applied.

The hub socket supplies Vcc and Gnd. The current limiter will initially prevent more than 100 mA from being drawn, even instantaneously, from the hub. If excess current is drawn, then the hub informs the host software of this error (see "Enumeration steps," step 5), an error message is displayed on the PC screen, and the device is **not** configured.

Because we haven't plugged in the I/O device yet, it is in the **unattached** state.

In Figure 3-1, note the two biasing resistors in the hub; they ensure that D+ and D– are low when no device is plugged in. There is a single biasing resistor on the device that is attached to either D+ or D–. When the USB cable is plugged in, the biasing resistor causes D+ or D– to rise above ground, and this changed voltage difference is recognized by the hub. We have detected a cable being plugged in! By convention, if the device-biasing resistor is connected to D+, we are informing the hub that this device is full speed (12 Mbps), while a biasing resistor on D– indicates a low speed (1.5 Mbps) device. Simple and effective!

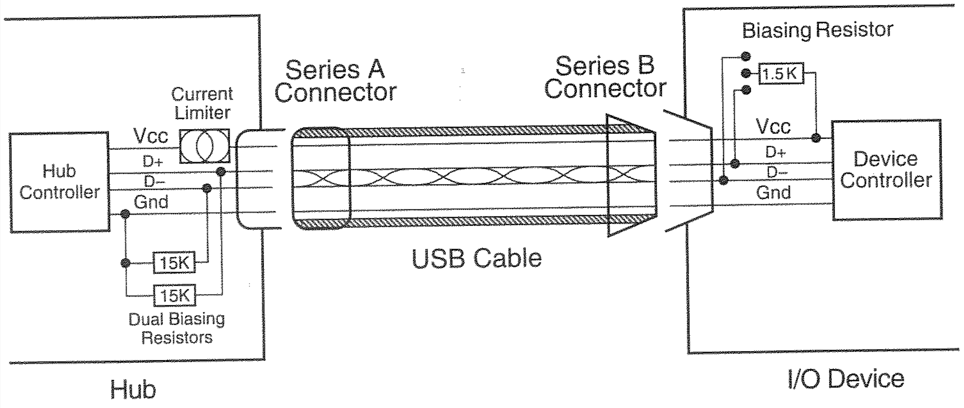


Figure 3-1. USB cable connection details

The I/O device is now in the **attached** state, a
configured and operational, the device moves

The hub updates a STATUS_CHANGE regis
device and then waits to be told what to do.

The PC host controls the enumeration phase a
requests to two devices. The hub that identifi
receive many requests for action, and the new
receive requests. If there are any other hubs b
they will not take part in this process. They w
bus, because that is one of their roles, but bec
by the PC host software during this process, t

The PC host software regularly polls all conn
In most cases a hub has nothing to report so w
this time the hub responds with the STATUS_
port has had a change in status—the PC host-b
begun!

ENUMERATION STEPS

In the following description the PC host is init
ToHub: prefix if the addressed device is the h
addressed device is the newly attached I/O dev

1. **ToHub:Get_Port_Status**: Host discover
2. **ToHub:Clear_Port_Feature(C_PORT_**
in the STATUS_CHANGE register that st
3. **ToHub:Set_Port_Feature(PORT_RESE**
a reset to the I/O device. The hub maintai
10 milliseconds. It then updates the RESE
PORT_CHANGE register and enables the
PORT_ENABLE bit in the PORT_STATU
register update causes an update to the ST
PC host will notice this on its next schedul
4. **ToHub:Get_Port_Status**: The PC host d

DOCKET ALARM

Explore Litigation Insights

Docket Alarm provides insights to develop a more informed litigation strategy and the peace of mind of knowing you're on top of things.

Real-Time Litigation Alerts

Keep your litigation team up-to-date with **real-time alerts** and advanced team management tools built for the enterprise, all while greatly reducing PACER spend.

Our comprehensive service means we can handle Federal, State, and Administrative courts across the country.

Advanced Docket Research

With over 230 million records, Docket Alarm's cloud-native docket research platform finds what other services can't. Coverage includes Federal, State, plus PTAB, TTAB, ITC and NLRB decisions, all in one place.

Identify arguments that have been successful in the past with full text, pinpoint searching. Link to case law cited within any court document via Fastcase.

Analytics At Your Fingertips

Learn what happened the last time a particular judge, opposing counsel or company faced cases similar to yours.

Advanced out-of-the-box PTAB and TTAB analytics are always at your fingertips.

API

Docket Alarm offers a powerful API (application programming interface) to developers that want to integrate case filings into their apps.

LAW FIRMS

Build custom dashboards for your attorneys and clients with live data direct from the court.

Automate many repetitive legal tasks like conflict checks, document management, and marketing.

FINANCIAL INSTITUTIONS

Litigation and bankruptcy checks for companies and debtors.

E-DISCOVERY AND LEGAL VENDORS

Sync your system to PACER to automate legal marketing.

WHAT WILL YOU BUILD? | **sales@docketalarm.com** | **1-866-77-FASTCASE**

fastcase®
Smarter legal research.