

UMI0026_3

ISPI16x ISA/Linux Eval Kit Software User's Guide

Semiconductors

July 2003

User's Guide Rev. 3.0

Revision History:

Version	Date	Description	Author
3.0	June 10, 2003	<ul style="list-style-type: none">Changes related to ISPI16x Software Stack release 1.11. Highlighted in green.	Srinivas Yarra
1.2	April 30, 2003	<ul style="list-style-type: none">Changes related to software release 1.10	Srinivas Yarra
1.1	December 23, 2002	<ul style="list-style-type: none">Updated to the latest templateSection 3.2: changed “./mk_es2” command to “./mk_es3” command.	Rajaram Veerappan
1.0	September 20, 2001	First release.	Srinivas Yarra

Note: ISPI16x denotes the ISPI160 Universal Serial Bus Host Controller and any future derivatives; and the ISPI161, ISPI161A and ISPI161A1 Universal Serial Bus single-chip Host Controller and Device Controller and any future derivatives.

We welcome your feedback. Send it to wired.support@philips.com.

Philips Semiconductors - Asia Product Innovation Centre
Visit www.semiconductors.philips.com/buses/usb or www.flexiusb.com



Contents

1.	INTRODUCTION	3
2.	ARCHITECTURE	4
3.	INSTALLATION	5
3.1.	EVALUATION KIT	5
3.2.	ISPI I6X HCD SOFTWARE	5
4.	CONFIGURATION.....	6
4.1.	EVALUATION KIT	6
4.2.	ISPI I6X HCD SOFTWARE	6
5.	INSIDE THE ISPI I6X LINUX STACK	7
6.	REFERENCES.....	7

The names of actual companies and products mentioned herein may be the trademarks of their respective owners. All other names, products, and trademarks are the property of their respective owners.

Note: ISPI 16x denotes the ISPI 160 Universal Serial Bus Host Controller and any future derivatives; and the ISPI 161, ISPI 161A and ISPI 161A1 Universal Serial Bus single-chip Host Controller and Device Controller and any future derivatives.

I. Introduction

The ISPI 16x ISA/Linux evaluation kit consists of an ISPI 16x evaluation board and a PC ISA interface card. This combination allows you to plug on the eval kit to any personal computer (PC). The PC can control the ISPI 16x chip through the ISA bus (16-bit ISA mode is used). Any real-time operating system (RTOS) can be run on the PC. For this evaluation kit, it is Linux. You can develop Universal Serial Bus (USB) class drivers or device drivers on top of the Linux host stack.

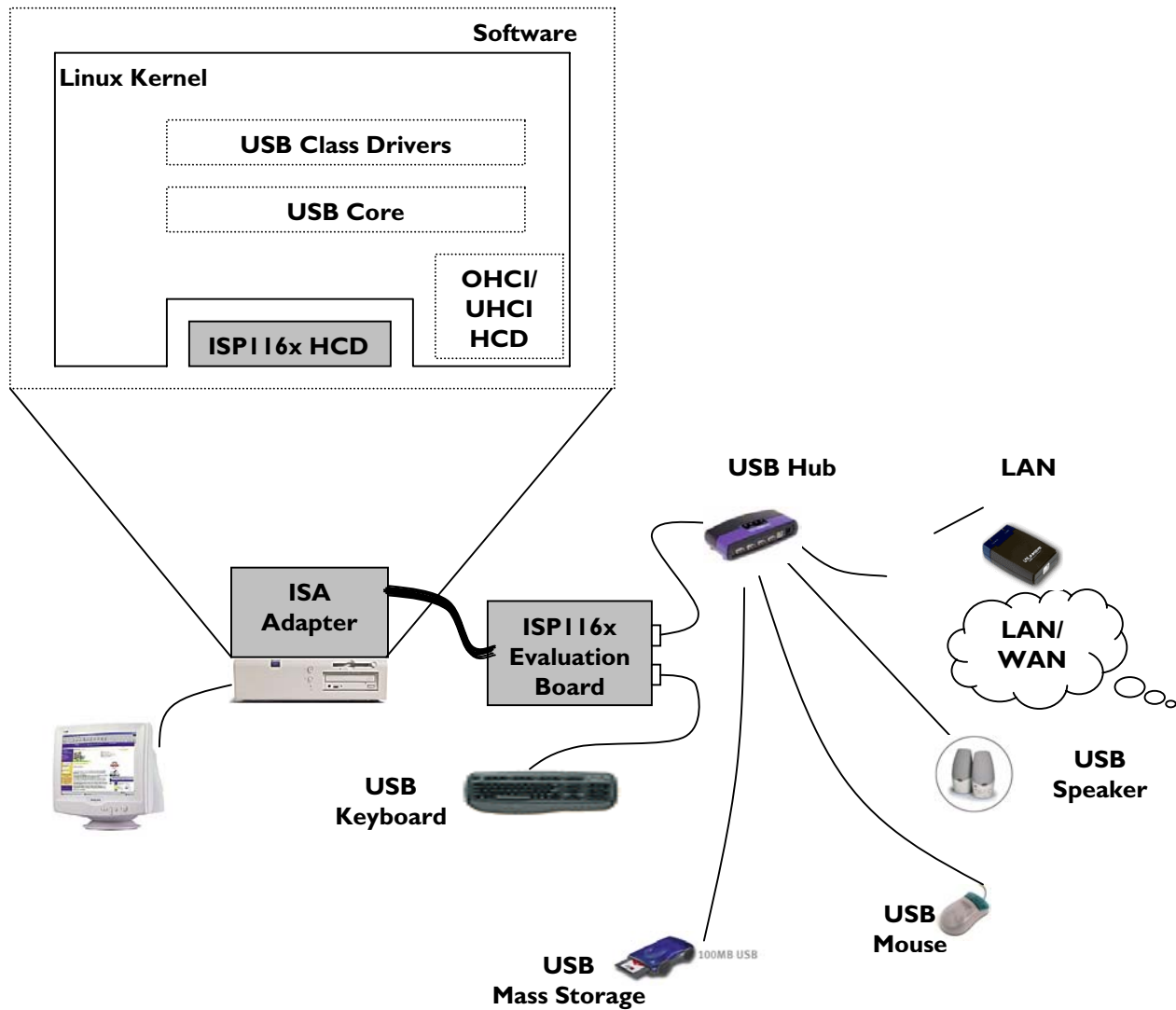
The evaluation kit comes with the Linux ISPI 16x Host Controller Driver (HCD) software. The ISPI 16x HCD works with Linux kernel version 2.4.X. You can run any device with the ISPI 16x Host Controller, provided the Linux kernel you are using supports the class driver required for the device and the class driver is enabled.

The ISPI 16x evaluation board provides an ISPI 16x expansion bus for interfacing to any reduced instruction set computer (RISC) processor development board. After completing the software development on the Linux PC, you can start porting for interfacing to the RISC processor board.

This document provides a brief introduction on the architecture, installation and configuration of the ISPI 16x PC evaluation kit HCD.

2. Architecture

The following diagram shows the architecture of the ISPI 16x evaluation kit and its driver software on Linux.



Linux kernel versions 2.4.0 and later support USB. The kernel *itself* provides the USB core, most of the class drivers and the USB HCDs. To work with a Linux PC, you just need to add another HCD software for the ISPI 16x Host Controller. You can dynamically load or unload the ISPI 16x HCD software to the Linux kernel. The interface with the USB core and the class drivers for the ISPI 16x HCD is the same and this driver can work simultaneously with other USB HCDs.

3. Installation

3.1. Evaluation Kit

1. Make sure the Linux PC is switched off.
2. Plug the PC ISA interface card into the PC (try to minimize the number of add-on cards), and connect it to the ISPI 16x evaluation board by using an IDE cable.
3. Boot the Linux PC

For details on configuration of the hardware, follow the *ISPI 16x PC Evaluation Kit User's Guide*.

3.2. ISPI 16x HCD Software

The ISPI 16x HCD software is written for and tested with Linux kernel version 2.4.20. The HCD assumes that the USB core is already present with the Linux kernel. The ISPI 16x HCD is installed using dynamic module loading. Follow the steps to install the software driver:

1. If you are not using Linux kernel version 2.4.0 or later, get a kernel version from <http://www.kernel.org/>
2. Enable the USB class drivers that you want to test while configuring the new kernel. If you are new to Linux kernel compilation and configuration, refer to *The Linux Kernel HOWTO* document available on the Linux Documentation Project web site <http://www.linuxdoc.org/>.
3. Make sure you have all the information needed to work with the required USB devices on Linux.
4. Start the PC with the configured new Linux kernel.
5. Log on to the system as root.
6. Put the ISPI 16x source code (`isp116x_linux_sw_stack.tar.gz`) to a known directory (for example, `/home`). Unzip the source code using `tar -zxvf isp116x_linux_sw_stack.tar.gz`
7. Go to the ISPI 16x source release directory `cd /home/116x/ISA/source`
8. Edit the `Makefile` for any changes needed (e.g linux kernel include directory)
9. Compile the ISPI 16x HCD and make the object file by using the `make clean all` command.
10. Load the ISPI 16x module by using the `/sbin/insmod hcd_1161.o` command.
11. Connect USB devices, such as hid, mouse, keyboard, printer, mass storage, hub, network adapter, modem, speaker (only), to the ISPI 16x evaluation board and test these devices.
12. You can unload the ISPI 16x HCD by using `/sbin/rmmod hcd_1161`.
You can load and unload the ISPI 16x HCD as long as the Linux kernel is ON.

4. Configuration

4.1. Evaluation Kit

To configure the hardware, refer to *ISPI161x PC Evaluation Kit User's Guide*.

4.2. ISPI16x HCD Software

You can customize the ISPI16x HCD software using compilation flags and predefined constants in the code. Table 4-1 provides a list of configurable parameters for the ISPI16x HCD.

Table 4-1: Configuration Parameters

Configuration Parameter	Description
NO_PTD_B5_5	This compilation flag is used for earlier versions of ISPI161 (ES1, ES2). If this flag is enabled the ISPI16x PTD header Bit 5 of Byte 5 is not used. Default not enabled
PHCI_MEM_SLAB	Use this compilation flag for allocation of chunk of memory at the time of initialization for the data structures. Default enabled.
__TRACE_LOW_LEVEL__	Use this compilation flag in the make file when low-level traces are required. This prints function entries while running the stack. Useful while debugging the ISPI16x HCD. Switch it off when you do not want these traces in the normal operation. Default not enabled
__TRACE_MID_LEVEL__	Use this compilation flag in the make file when midlevel traces are required. This prints some of the flags and internal function values while running the stack. Useful while debugging the ISPI16x HCD. Switch it off when you do not want these traces in the normal operation. Default not enabled
__TRACE_HIGH_LEVEL__	Use this compilation flag in the make file when high-level traces are required. This prints the hexadecimal values of the data going out and coming in to the driver while running the stack. Useful while debugging the ISPI16x HCD. Switch it off when you do not want these traces in the normal operation. Default not enabled
ATL_BUFF_LENGTH	This is the RAM buffer size for the ISPI16x Host Controller bulk, control and interrupt transfers. Configured to 2048 bytes. You can change this value in the <code>hcd_1161.h</code> file. This is used in addition with <code>ITL_BUFF_LENGTH</code> and the total length should be less than 4096 bytes.
ITL_BUFF_LENGTH	This is the RAM buffer size for the ISPI16x Host Controller isochronous transfers. Configured to 1024 bytes (total Isochronous ping-pong buffer size of 2048 bytes). You can change this value in the <code>hcd_1161.h</code> file. This is used in addition with <code>ATL_BUFF_LENGTH</code> and the total length should be less than 4096 bytes.
HC_IO_BASE	HCD I/O Port address base. This value depends on the jumper settings on the ISPI16x evaluation board. For jumper settings, refer to <i>ISPI16x PC Evaluation Kit User's Guide</i> .
MAX_BULK_TD_BUFF_SIZE	Specifies the maximum size of bulk transfer descriptor payload between the driver and the ISPI16x. The value of this should be less than 1023 bytes.

5. Inside the ISPI 16x Linux Stack

The following table shows the contents and their description of the ISPI 16x Linux stack release.

Table 5-1: Top-Level Directory Contents

File Name	Description
<code>116x/ISA/source/Makefile</code>	Makefile for the compilation of the ISPI 16x stack
<code>116x/ISA/source/README</code>	README file that gives details of the version changes
<code>116x/ISA/source/hcd_1161.c</code>	ISPI 16x Host Controller Driver source file
<code>116x/ISA/source/hcd_1161.h</code>	ISPI 16x Host Controller Driver header file

6. References

- *Universal Serial Bus Specification Rev. 2.0*
- *ISPI 161A1 Universal Serial Bus single-chip host and device controller datasheet*
- *ISPI 161A Universal Serial Bus single-chip host and device controller datasheet*
- *ISPI 161 Universal Serial Bus single-chip host and device controller datasheet*
- *ISPI 160 Embedded Universal Serial Bus Host Controller datasheet*
- *ISPI 161x PC Evaluation Kit User's Guide*
- *The Linux Kernel HOWTO*

Philips Semiconductors

Philips Semiconductors is a worldwide company with over 100 sales offices in more than 50 countries. For a complete up-to-date list of our sales offices please e-mail sales.addresses@www.semiconductors.philips.com. A complete list will be sent to you automatically. You can also visit our website <http://www.semiconductors.philips.com/sales/>

www.semiconductors.philips.com

© **Koninklijke Philips Electronics N.V. 2003**

All rights reserved. Reproduction in whole or in part is prohibited without the prior written consent of the copyright owner. The information presented in this document does not form part of any quotation or contract, is believed to be accurate and reliable and may be changed without notice. No liability will be accepted by the publisher for any consequence of its use. Publication thereof does not convey or imply any license under patent – or other industrial or intellectual property rights.

