



PHILIPS

Philips Semiconductors

Connectivity

June 2002

UMI0010-01

Kenobi2

(Audio Play + Printer Demo Software for the ISP116x)

Rev 1.07

Akifumi OKANO
+81-3-3740-4668
akifumi.okano@philips.com
Computing Segment/BL-Connectivity
Philips Semiconductors, Tokyo

Note: In this document, ISP116x denotes the ISP1161, the ISP1161A and the ISP1160.

Revision History:

Rev.	Date	Descriptions	Author
0.96	Nov 21, 2001	First draft	OKANO, Akifumi
0.97	Nov 30, 2001	Updates for correction	OKANO, Akifumi
1.00	Nov 30, 2001	Official release, formalized by APIC	Tee Chee Yen
1.01	Dec 05, 2001	Code revision history updated for version1.1i	OKANO, Akifumi
1.02	Dec 13, 2001	Code revision history updated for version1.1j	OKANO, Akifumi
1.03	Jan 23, 2002	PCI board support part added for version1.2	OKANO, Akifumi
1.04	Feb 26, 2002	Manual for PCI PnP support executable version 1.3	OKANO, Akifumi
1.05	Apr 01, 2002	Minor fix for manual	OKANO, Akifumi
1.07	Apr 04, 2002	Typo corrected	OKANO, Akifumi

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



PHILIPS

Kenobi2

Rev. 1.07

This is a legal agreement between you (either an individual or an entity) and Philips Semiconductors. By accepting this product, you indicate your agreement to the disclaimer specified as follows:

DISCLAIMER

PRODUCT IS DEEMED ACCEPTED BY RECIPIENT. THE PRODUCT IS PROVIDED "AS IS" WITHOUT WARRANTY OF ANY KIND. TO THE MAXIMUM EXTENT PERMITTED BY APPLICABLE LAW, PHILIPS SEMICONDUCTORS FURTHER DISCLAIMS ALL WARRANTIES, INCLUDING WITHOUT LIMITATION ANY IMPLIED WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE, AND NON-INFRINGEMENT. THE ENTIRE RISK ARISING OUT OF THE USE OR PERFORMANCE OF THE PRODUCT AND DOCUMENTATION REMAINS WITH THE RECIPIENT. TO THE MAXIMUM EXTENT PERMITTED BY APPLICABLE LAW, IN NO EVENT SHALL PHILIPS SEMICONDUCTORS OR ITS SUPPLIERS BE LIABLE FOR ANY CONSEQUENTIAL, INCIDENTAL, DIRECT, INDIRECT, SPECIAL, PUNITIVE, OR OTHER DAMAGES WHATSOEVER (INCLUDING, WITHOUT LIMITATION, DAMAGES FOR LOSS OF BUSINESS PROFITS, BUSINESS INTERRUPTION, LOSS OF BUSINESS INFORMATION, OR OTHER PECUNIARY LOSS) ARISING OUT OF THIS AGREEMENT OR THE USE OF OR INABILITY TO USE THE PRODUCT, EVEN IF PHILIPS SEMICONDUCTORS HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

CONTENTS

1. INTRODUCTION	5
2. ENVIRONMENT SETUP	5
2.1. HARDWARE SETUP.....	6
2.1.1. <i>ISP116x Evaluation Kit</i>	6
2.1.2. <i>USB Speakers</i>	6
2.1.3. <i>Printer</i>	6
2.1.4. <i>Other Equipment</i>	6
2.2. SOFTWARE SETUP.....	7
2.2.1. <i>Installing Kenobi2</i>	7
2.2.2. <i>Preparing the Audio Data</i>	7
2.2.3. <i>Preparing the Printer Data</i>	7
2.2.4. <i>Launching the Program</i>	8
2.2.5. <i>Command-Line Switches</i>	9
2.3. OPERATION.....	11
2.3.1. <i>Plugging-In and Plugging-Out USB Devices</i>	11
2.3.2. <i>Playing Audio</i>	12
2.3.3. <i>Operating a Printer</i>	13
2.3.4. <i>Register Dump</i>	13
2.3.5. <i>Port-Device Information</i>	13
2.3.6. <i>Environment Parameter View</i>	13
2.3.7. <i>Online Help</i>	13
2.3.8. <i>Quit</i>	13
3. SOURCE CODE	14
3.1. REBUILDING THE CODE.....	14
3.2. MODULE STRUCTURE.....	14
3.3. ATL BUFFER CONTROL MECHANISM.....	15
3.4. IITL BUFFER CONTROL MECHANISM.....	16
4. LIMITATIONS	17
5. REFERENCES	18

FIGURES

Figure 2-1: Software Model.....	5
Figure 2-2: Directory View of Kenobi2	7
Figure 2-3: Kenobi2 Started.....	8
Figure 2-4: ISP1161 Not Found.....	9
Figure 2-5: No Status Line	9
Figure 2-6: Music List File Sample.....	10
Figure 2-7: Enumeration Complete	11
Figure 2-8: Music Play Started	12
Figure 2-9: Music Selection Menu List.....	12
Figure 2-10: Operating Printer.....	13
Figure 3-1: Module Structure	14
Figure 3-2: ATL Buffer Control Mechanism.....	16
Figure 3-3: ITL Buffer Control Mechanism	17

Microsoft and Windows are either registered trademarks or trademarks of Microsoft Corp. in the United States and/or other countries. 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.

Kenobi2**Rev. 1.07**

Note: In this document, ISP116x denotes the ISP1161, the ISP1161A and the ISP1160.

1. Introduction

Kenobi2 is a program that can be used to play audio (music) with the ISP116x. Kenobi2 can drive a printer too.

Kenobi2 is a program for the DOS environment to show the isochronous data transfer capability of the ISP116x. This program works on IBM-compatible PC with Microsoft® Windows® 95 or Windows 98 operating system that allows the DOS program to directly access the hardware.

To play this software, you must have the ISP116x ISA or PCI evaluation kit.

Kenobi2 plays audio using USB speakers. This program has been tested with USB speaker system based on Philips Semiconductors UDA1321 USB audio codec.

Kenobi2 is distributed as an executable binary code with the source code. No restriction is applied on the distribution and modification of the code. However, use and change the code at your own risk.

2. Environment Setup

Figure 2-1 shows the Kenobi2 software model.

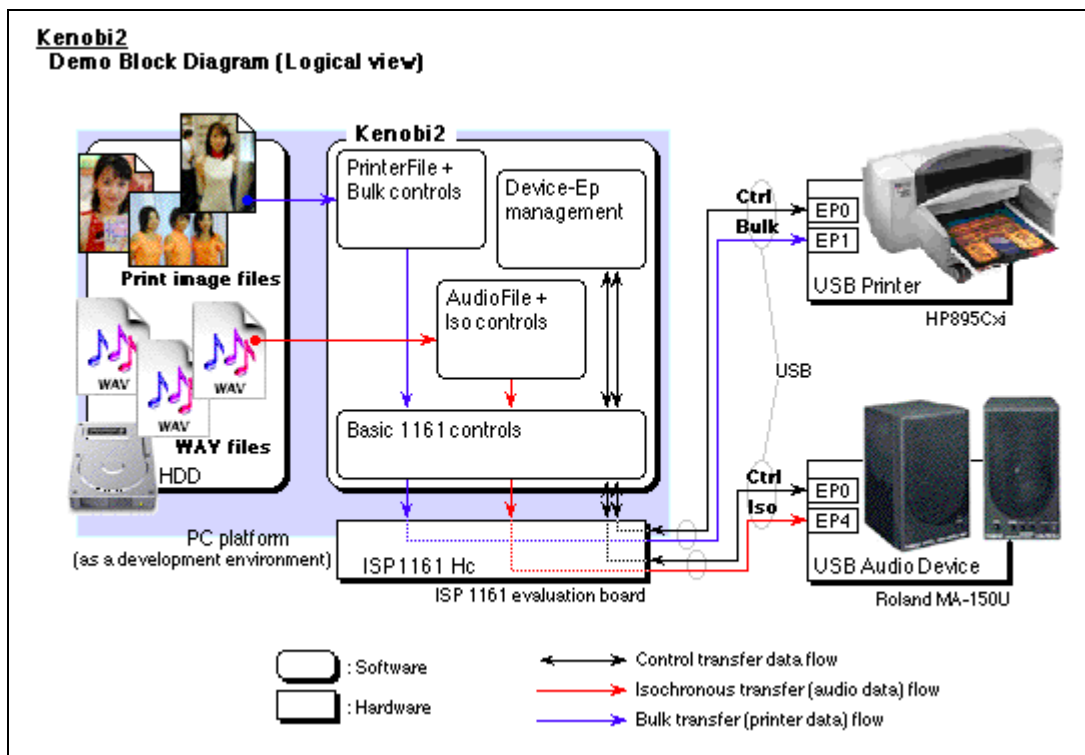


Figure 2-1: Software Model

Kenobi2

Rev. 1.07

2.1. Hardware Setup

2.1.1. ISP116x Evaluation Kit

You must have Windows 95 or Windows 98 operating system to play Kenobi2. These 16-bit OS family allows direct hardware access from the DOS environment without any I/O port accessing drivers.

Important: You **must not** install any drivers for the ISP116x evaluation board. The Kenobi2 program directly accesses the evaluation hardware. Installing any driver software may cause problems because some of these drivers may unexpectedly overwrite the hardware settings. When the first time you boot the PC after installing the PCI evaluation board, Windows Plug and Play (PnP) feature may prompt you to install a driver for the PCI bridge. Ignore this prompt.

To play Kenobi2, you must have the ISP116x ISA or PCI evaluation kit. To get more details on the evaluation kit, refer to *ISP1161 PC Eval Kit User's Guide* if you are using the ISA evaluation board or refer to *ISP1161x PCI Evaluation Board User's Guide* if you are using the PCI evaluation board.

You may need disk space on the PC to save the audio data. Kenobi2 uses CD equivalent linear-PCM audio data (16-bit/sample, 2-channel stereo, 44.1 kHz sampling frequency) that takes disk space of about 10 MB/min.

2.1.2. USB Speakers

You need USB speakers or headphones for the audio output. It converts the USB data to air vibration for your ears.

The audio has been tested using Roland MA-150U and Philips 350/00. Both these systems use the Philips USB audio codec chip (UDA1321).

It is recommended that you use USB speaker system with analog volume control because Kenobi2 does not support HID class, see Section 4. This means that the USB host side does not support sound amplitude control.

2.1.3. Printer

The operation test has been done using the following printers:

- HP® 895Cxi
- Epson® CL-700
- Canon® BJ F850.

These printers work fine for raster image printing. You may use any other equivalent printer. Among these printers, the HP printer is the only printer that works for text data printing. If you want to print a text file, choose a text file in the print command prompt instead of the raster image command file. You must add a file suffix for the text file. This is because Kenobi2 adds a suffix to the file name if it does not have one. Kenobi2 automatically selects the suffix name by detecting the vendor ID code that can be obtained in the USB enumeration process.

2.1.4. Other Equipment

Other USB devices can be useful for the demo. You can illustrate that the ISP116x can enumerate another USB device during the audio play. The USB device connection or disconnection on the port demonstrates that the control transfer can be done with no audio degradation.

Kenobi2

Rev. 1.07

2.2. Software Setup

2.2.1. Installing Kenobi2

Just copy Kenobi2.exe to any directory.

You must have the audio data as a file named “default.wav” in the same directory as that of Kenobi2.exe. However, it is not necessary to have the “default.wav” file in the same directory if you always use the “-a” or “-l” option switch. Details on “default.wav” can be found in Section 2.2.2, and details on command-line switches can be found in Section 2.2.5.

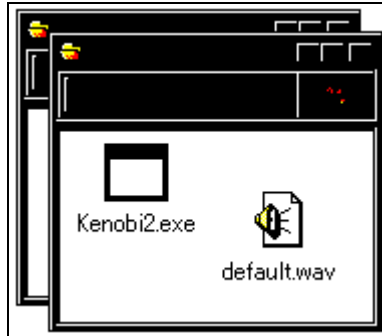


Figure 2-2: Directory View of Kenobi2

2.2.2. Preparing the Audio Data

Before launching Kenobi2, you must prepare the audio file. You can create it by using the CD ripping software. You can find this software on the Internet or a CD-R writing software package. The file must have data in the same format as an audio CD. Using the ripping software, save the file in the “.wav” format. The internal format must be in linear-PCM, 16-bit/sample, 2-channel stereo, 44.1 kHz sampling frequency. Kenobi2 does not process the audio data. The audio data flows down to endpoint 4 in the USB audio device through the isochronous pipe.

If you want to make your audio file as the default file, name it “default.wav” and put it in the same directory as the Kenobi2 executable. The default file is used if you are not using any command-switch option for audio files when launching Kenobi2.

The file name and the directory name must be in the classic DOS format. You can use only eight characters for the file name and three characters for the file extension.

2.2.3. Preparing the Printer Data

Kenobi2 does not do anything to the data content of the print image. It just transfers data from the file to an endpoint of the printer. The printer must receive data in the printer language. For example, in the case of the HP 895Cxi printer, the printer language is HP-PCL. Therefore, you must convert the image source to the command data. You can do this by using the printer driver on the Windows operating system as follows:

1. Open the picture file with any graphics software in Windows.
2. Choose **Print** from the **File** menu.
3. In the printer driver dialog box, select the check box that will enable you to print the picture to a file (a printer image file).
4. Save the file in the classic DOS format.

Kenobi2

Rev. 1.07**2.2.4. Launching the Program**

You can launch Kenobi2 in either the Windows or DOS environment. To launch Kenobi2 in the Windows environment, double-click the **Kenobi2.exe**¹ icon. To launch it in the DOS environment, type **Kenobi2** at the command prompt. If you are launching Kenobi2 in the DOS environment, you may need to be in the Kenobi2.exe directory.

```

KENOB12 - KENOB12
*****
***          Kenobi2          ***
***   version 1.3 (Feb 25 2002)   ***
***          Philips Semiconductors, Tokyo          ***
*****
***   build : Feb 25 2002  20:12:25   ***
***   123456789ABCDEF                ***

>>> ChipID = 0x6122  :  ISP1161A ES4 <<<

Hc ISA base address = 0x0290 (on ISA)
Hc IRQ               = 3
operation started...
* sof=0x00010642 uis=0x00 is=0x04 p1=0x00000100 p2=0x00000100 t(sec)=0000122

```

Figure 2-3: Kenobi2 Started

If you have both the PCI and ISA bus boards on your PC, Kenobi2 will detect the PCI board and start to operate the ISP116x on PCI.

If it does not work when using PCI

If it does not work with PCI, make sure that the board is connected properly and try to run the program again. Also, make sure that no driver is installed for the evaluation board.

If it does not work when using ISA

When using ISA, if you get the message:

- “1161 is not present”: this may be because of incorrect I/O address; see Figure 2-4
- or
- “no status line²”: this may be because of incorrect IRQ number, see Figure 2-5.

You must use command-line switches when launching Kenobi2. The “-b” and “-h” switches set these parameters. These switches are explained in detail in Section 2.2.5.

¹ In this document, items that you click or type are indicated in bold.

² A status line is a line that starts with the ‘!’ character and periodically reports the internal status of the frame count, interrupt cause and root hub register values, and the timer value.


```

*****
****              Kenobi2              ****
****      version 1.3 (Feb 25 2002)      ****
****              ****                  ****
****      Philips Semiconductors, Tokyo  ****
****              ****                  ****
****      build : Feb 25 2002  20:12:25  ****
****      123456789ABCDEF                ****
*****

>>> ChipID = 0xFFFF :   This may not ISP1161.
*** ABORTED! ***
1161 is not present!
* sof=0x00000000 uis=0xFFFF is=0xFFFF p1=0xFFFFFFFF p2=0xFFFFFFFF t(sec)=0000000
D:¥pr j¥Kenobi2>

```

Figure 2-4: ISP1161 Not Found

```

*****
****              Kenobi2              ****
****      version 1.3 (Feb 25 2002)      ****
****              ****                  ****
****      Philips Semiconductors, Tokyo  ****
****              ****                  ****
****      build : Feb 25 2002  20:12:25  ****
****      123456789ABCDEF                ****
*****

>>> ChipID = 0x6122 :   ISP1161A ES4 <<<
Hc ISA base address = 0x0290 (on ISA)
Hc IRQ               = 10
operation started...
-

```

Figure 2-5: No Status Line

2.2.5. Command-Line Switches

There are some command-line switches in Kenobi2 that can set program parameters at the time of the program launch. All these switches take only one parameter after the switch identifier. Sample scenarios on the use of these command-line switches are given on the following page.

Command-Line Switch Description

- b** Defines the **base address** of the ISP116x in the ISA I/O space.
A hexadecimal value must follow this switch. If this address is incorrect, Kenobi2 will report “1161 is not present” (see Figure 2-4).

Kenobi2**Rev. 1.07**

Default = 290 (hexadecimal)

For the PCI board, this setting is ignored.

-h Defines the **IRQ number** for the Host Controller (INT1 interrupt signal).

A decimal value must follow this switch. If this number is not set correctly, you may not get the ISP116x status line on the screen and the 1 ms periodic pulse cannot be found on the INT1 signal line (check with an oscilloscope by probing on INT1), see Figure 2-5.

Default = 10 (decimal)

For the PCI board, this setting is ignored.

-a Defines the **audio (music) file** path and the **file** name.

It defines the audio file to play. The path must be written in the UNIX path format, i.e., with path delimiter (/). A linear-PCM coded “.wav” file or raw data audio file (no “.wav” header data file) can be used. Kenobi2 will detect whether the file is in the “.wav” or raw data format.

-l Defines the **music list file** path and the **file** name.

You can define audio files by using the “music list file”. The “music list file” is a text file that contains a list of the audio file path and the file name. Each item is placed in a new line. The list will display as a menu when you start to play the audio enabling you to select a file to play. The path must be written in the UNIX path format, i.e., with path delimiters.

```
# music list file '#' and '!' character in the
# line head can be used as comment line marker.
!
! Also the empty line will be ignored.

../music/allnight.wav
../music/morethan.wav
../music/dig_it.wav

e:/k_debug/testwave/sin.wav
e:/k_debug/testwave/sq.wav
e:/k_debug/testwave/ramp.wav
e:/k_debug/testwave/ramp2.wav
../music/i_will.wav
../music/yozora.wav
```

Figure 2-6: Music List File Sample

-p Defines the default path of the **print file**.

You can set a default path for print image files. The printer operation command (see Section 2.3.3) prompts you to enter the file path and the file name for the source of printing image. The “-p” switch enables you to set the default path from the DOS command line.

Here are some sample scenarios:

- The following expression sets the base address of the ISP116x to 300H and the IRQ number of the Host Controller to 3. The music file is placed in the directory “../music” and its name is “dig_it.wav”.

Kenobi2 -b 300 -h 3 -a ../music/dig_it.wav¹

¹ Press the **Enter** key after typing the command.

Kenobi2**Rev. 1.07**

- The following expression sets the base address of the ISP116x to 290H and the IRQ number of the Host Controller to 10. The music list file is placed in the program directory and is named “song.lst”. The default path of the print file is set to “c:/print/image/”.

Kenobi2 -b 290 -h 10 -l song.lst -p c:/print/image/¹

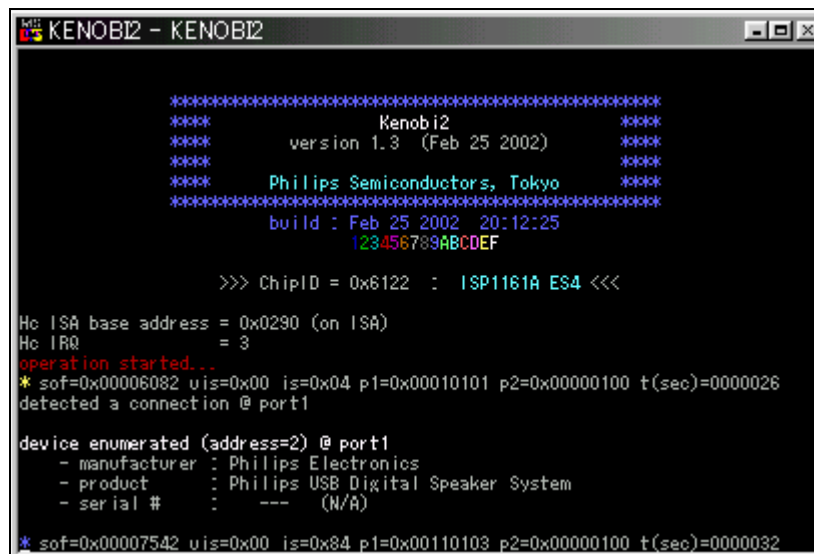
These switches can be used in the DOS environment. However, if you want to launch Kenobi2 in the Windows environment by double clicking on it, you will need to write a batch file or edit-rebuild the source code to change the default value. You can have more options by editing-rebuilding the code. For more information, refer to the Kenobi2 code.

2.3. Operation

2.3.1. Plugging-In and Plugging-Out USB Devices

After launching Kenobi2 and when the program starts running, you can connect a device to any of the downstream ports. It emulates the USB device enumeration.

When the device enumeration is complete, string device descriptors are displayed on the screen with assigned USB device addresses (see Figure 2-7). The USB device address of the first enumerated device is 2. The address of the second enumerated device is 3. If two devices are connected on both ports at the same time, enumeration will start from port 1. Kenobi2 does not support hub. Therefore, the maximum number of devices that can be connected to Kenobi2 is two.



```

KNOB12 - KNOB12

*****
***          Kenobi2          ***
***   version 1.3 (Feb 25 2002)   ***
***   Philips Semiconductors, Tokyo   ***
*****
build : Feb 25 2002  20:12:25
      123456789ABCDEF

>>> ChipID = 0x6122 : ISP1161A ES4 <<<

He ISA base address = 0x0290 (on ISA)
He IRQ      = 3
operation started...
* sof=0x00006082 uis=0x00 is=0x04 p1=0x00010101 p2=0x00000100 t(sec)=0000026
detected a connection @ port1

device enumerated (address=2) @ port1
- manufacturer : Philips Electronics
- product      : Philips USB Digital Speaker System
- serial #     : --- (N/A)

* sof=0x00007542 uis=0x00 is=0x84 p1=0x00110103 p2=0x00000100 t(sec)=0000032

```

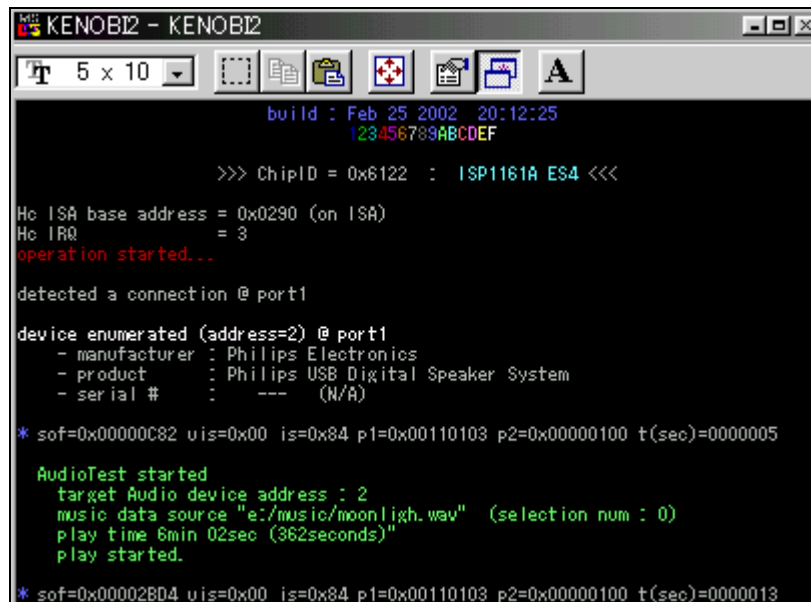
Figure 2-7: Enumeration Complete

Addresses are recycled. At enumeration, the smaller available address number is used for the address. The smallest USB address for a device is 2. This is because 1 is reserved for future use in Kenobi2.

¹ Press the **Enter** key after typing the command.

2.3.2. Playing Audio

To play audio, press the **a** key on the keyboard. Kenobi2 will search for the USB audio device on the ISP116x port. The audio device is identified by the interface class code (0x01) that can be found in the interface descriptor. If there are two audio devices, Kenobi2 will choose the device with the smaller USB address. If you have defined multiple audio files using the “-l” option switch at the program launch, the “music file list” as given in Figure 2-9 is displayed.



```

KENOB12 - KENOB12
build : Feb 25 2002 20:12:25
123456789ABCDEF

>>> ChipID = 0x6122 : ISP1161A E64 <<<

Hc ISA base address = 0x0290 (on ISA)
Hc IRQ = 3
operation started...

detected a connection @ port1

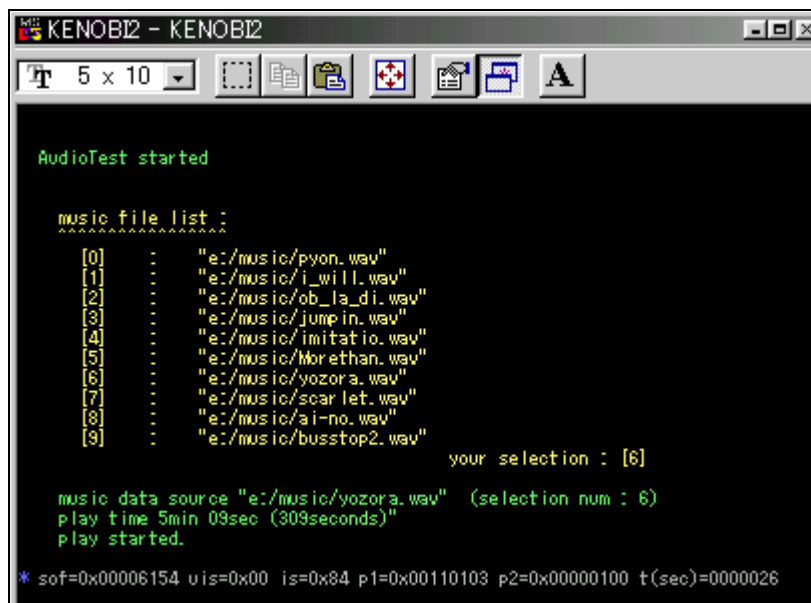
device enumerated (address=2) @ port1
- manufacturer : Philips Electronics
- product : Philips USB Digital Speaker System
- serial # : --- (N/A)

* sof=0x00000C82 uis=0x00 is=0x84 p1=0x00110103 p2=0x00000100 t(sec)=0000005

AudioTest started
target Audio device address : 2
music data source "e:/music/moonligh.wav" (selection num : 0)
play time 6min 02sec (362seconds)"
play started.

* sof=0x00002B04 uis=0x00 is=0x84 p1=0x00110103 p2=0x00000100 t(sec)=0000013
  
```

Figure 2-8: Music Play Started



```

KENOB12 - KENOB12

AudioTest started

music file list :
.....
[0] : "e:/music/pyon.wav"
[1] : "e:/music/i_will.wav"
[2] : "e:/music/ob_la_di.wav"
[3] : "e:/music/jumpin.wav"
[4] : "e:/music/imitatio.wav"
[5] : "e:/music/Morethan.wav"
[6] : "e:/music/yozora.wav"
[7] : "e:/music/scar let.wav"
[8] : "e:/music/ai-no.wav"
[9] : "e:/music/busstop2.wav"

your selection : [6]

music data source "e:/music/yozora.wav" (selection num : 6)
play time 5min 09sec (309seconds)"
play started.

* sof=0x00006154 uis=0x00 is=0x84 p1=0x00110103 p2=0x00000100 t(sec)=0000026
  
```

Figure 2-9: Music Selection Menu List

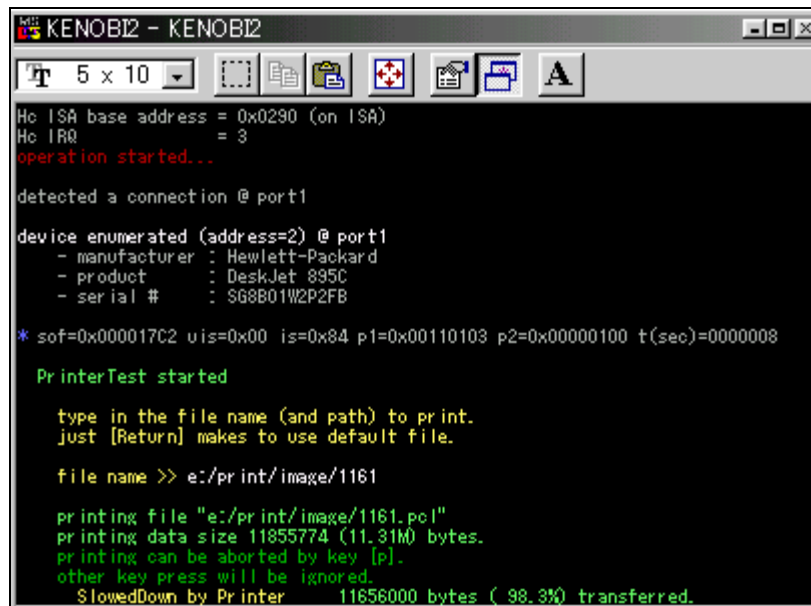
Kenobi2
Rev. 1.07

2.3.3. Operating a Printer

To operate a printer, press the **p** key on the keyboard. Kenobi2 will search the USB printer device on the ISP116x port. The printer is identified by the interface class code (0x07) that can be found in the interface descriptor. If there are two printer devices, Kenobi2 will choose the device that has the smaller USB address.

After the printer is found, Kenobi2 will prompt you to point to a printer image file. Type the file path and the file name, and then press the **Enter** key.

The printer process works as a **modal process**. This means the printer process works in the printer mode. The printer mode does not accept any user interaction except printer abort command. The printer mode can be aborted by pressing the **p** key. Moreover, enumeration processing will not be started during the printing process.



```

KNOB12 - KNOB12
He ISA base address = 0x0290 (on ISA)
He IRQ = 3
operation started...
detected a connection @ port1
device enumerated (address=2) @ port1
- manufacturer : Hewlett-Packard
- product : DeskJet 895C
- serial # : SG8B01W2P2FB
* sof=0x000017C2 uis=0x00 is=0x84 p1=0x00110103 p2=0x00000100 t(sec)=0000008
Pr interTest started
type in the file name (and path) to print.
just [Return] makes to use default file.
file name >> e:/print/image/1161
printing file "e:/print/image/1161.pcl"
printing data size 11855774 (11.31M) bytes.
printing can be aborted by key [p].
other key press will be ignored.
SlowedDown by Pr inter 11856000 bytes ( 98.3%) transferred.
  
```

Figure 2-10: Operating Printer

2.3.4. Register Dump

Press the **d** key on the keyboard to show the register dump in the ISP116x. The contents of all readable registers will be displayed on the screen.

2.3.5. Port-Device Information

Pressing the **i** key displays the port-device information.

2.3.6. Environment Parameter View

Press the **e** key to display setting status of Kenobi2 command-line parameters.

2.3.7. Online Help

To display the online help, press the **h** or **?** key. A list of operation commands and command-option switches will be displayed on the screen.

2.3.8. Quit

To quit, press the **q** or **Esc** key. To confirm, Kenobi2 will prompt you to press the **y** key. Pressing any key other than **y** at confirmation will cancel the quit.

3. Source Code

3.1. Rebuilding the Code

The Kenobi2 code is free. No restriction is applied on its distribution and modification. This code is written from scratch. Some portion of the code is also referenced to free code.

For code building, use Borland® Turbo C++ 3.0 because there are some development environment dependencies in the source code. Interrupt service routines (ISRs) definitions are in the Turbo C manner, and the DMA memory allocation and file access use Turbo C's library calls. Create a folder called "Kenobi2.prj" and copy all the ".c" files to the folder. If you are using the project file, edit the appropriate parameters, such as, the directory in which Turbo C is installed, and include the file directory, library directory, and so on.

3.2. Module Structure

Figure 3-1 shows the module structure of the source code. Kenobi2 is a monolithic program. However, the modules are separated in layers inside the source code. The bottom block in the figure is the ISP116x hardware, and the block above it is the PC hardware. Kenobi2 accesses the ISP116x through the PC hardware. Kenobi2 accesses the ISP116x register via the ISA or PCI bus by using the I/O port read/write calls. The interrupt controller on the PC handles ISR and DMA tasks with the DMA controller on PC.

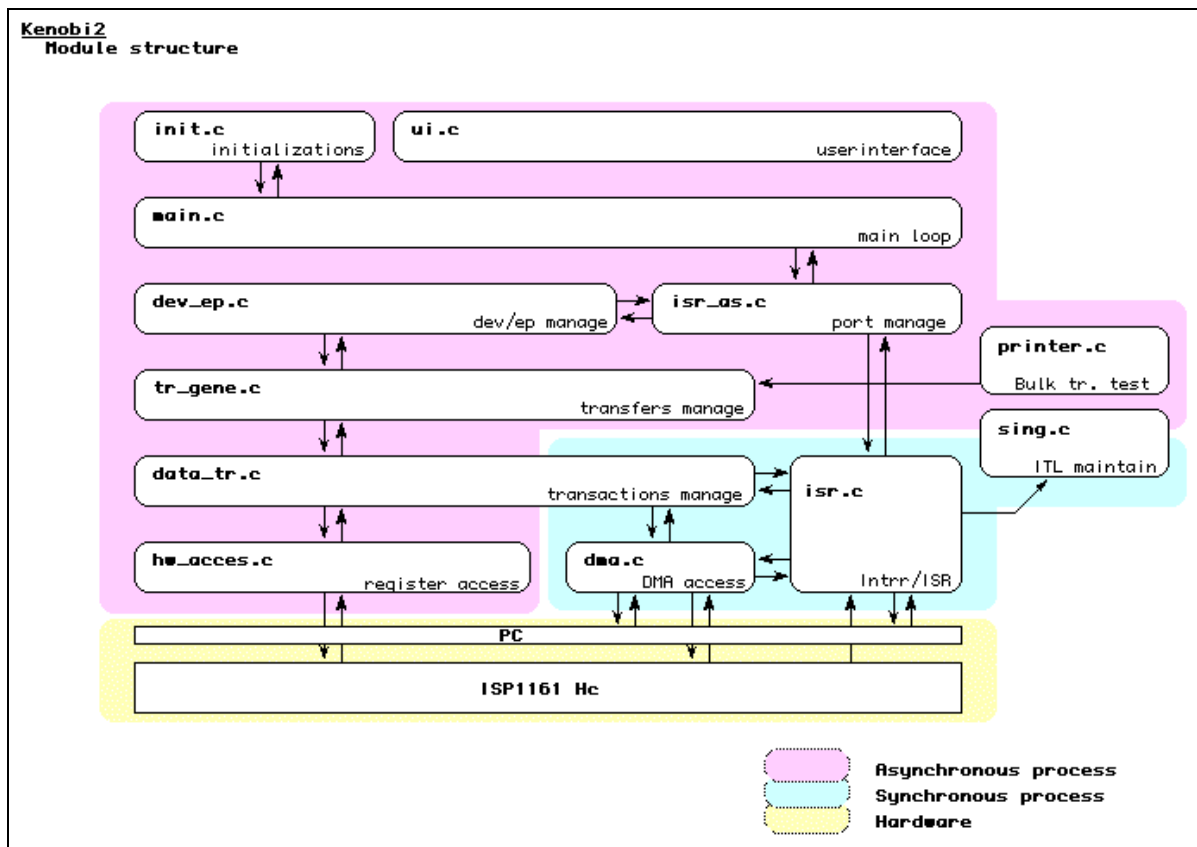


Figure 3-1: Module Structure

Kenobi2**Rev. 1.07**

Software modules can be categorized into two types of processes:

- Synchronous process: These are synchronized with SOF/ITL, ATL and ALLEOT interrupts (these will be executed in the context of interrupt service routine (ISR)).
- Asynchronous process: These modules have tasks that are not required to serve the process in real time. Table 3-1 explains functionality of these modules.

Table 3-1: Module Functionality

Module Name	Category	Functionality
data_tr.c	synchronous and asynchronous	transaction management low-level ATL controls using PIO
dev_ep.c	asynchronous	device and endpoint management
hw_acces.c	asynchronous	ISP116x register access abstraction
init.c	asynchronous	initializations
isr.c	synchronous	interrupt service routines (ISRs) ISR installer/restorer
isr_as.c	asynchronous	port maintenance (InterruptServiceRoutine_ASynchronous)
main.c	asynchronous	main function program main loop
printer.c	asynchronous	printer-driving process
sing.c	synchronous and asynchronous	audio-driving process
tr_gene.c	asynchronous	transfer management
ui.c	asynchronous	user interface

3.3. ATL Buffer Control Mechanism

The ATL buffer management mechanisms is designed for control, bulk and interrupt (using ATL) transfers. A PTD in the ATL buffer makes one or more transactions. Figure 3-2 shows how Kenobi2 does the transaction. You can find the code inside the “raw_transaction()” function in the “tr_gene.c” file.

The current implementation of Kenobi2 can place only one PTD in one ATL image. To keep the ATL control logic simple, no scheduling mechanism is implemented.

To generate a transaction, a data structure “ATLD” is generated. It includes a pointer to the ATL data image and some flags to manage the ATL buffer. An ATLD can be configured as a member of the link list to make a sequence of transfer.

The “raw_transaction()”function in the “tr_gene.c” file makes an ATLD, sets its pointer to ATLD_base and enables it. All the control functions called from “raw_transaction()” are found in the “data_tr.c” file. The “data_tr.c” file has asynchronous and synchronous functions. The synchronous function call will be called by ISR in isr.c. Figure 3-2 shows how these functions work with interrupt services.

The upper right corner in Figure 3-2: shows an ATLD link list image. If there is only one ATLD, its next_ATLD_ptr will hold NULL. In the middle of Figure 3-2 are flags and time sequence.

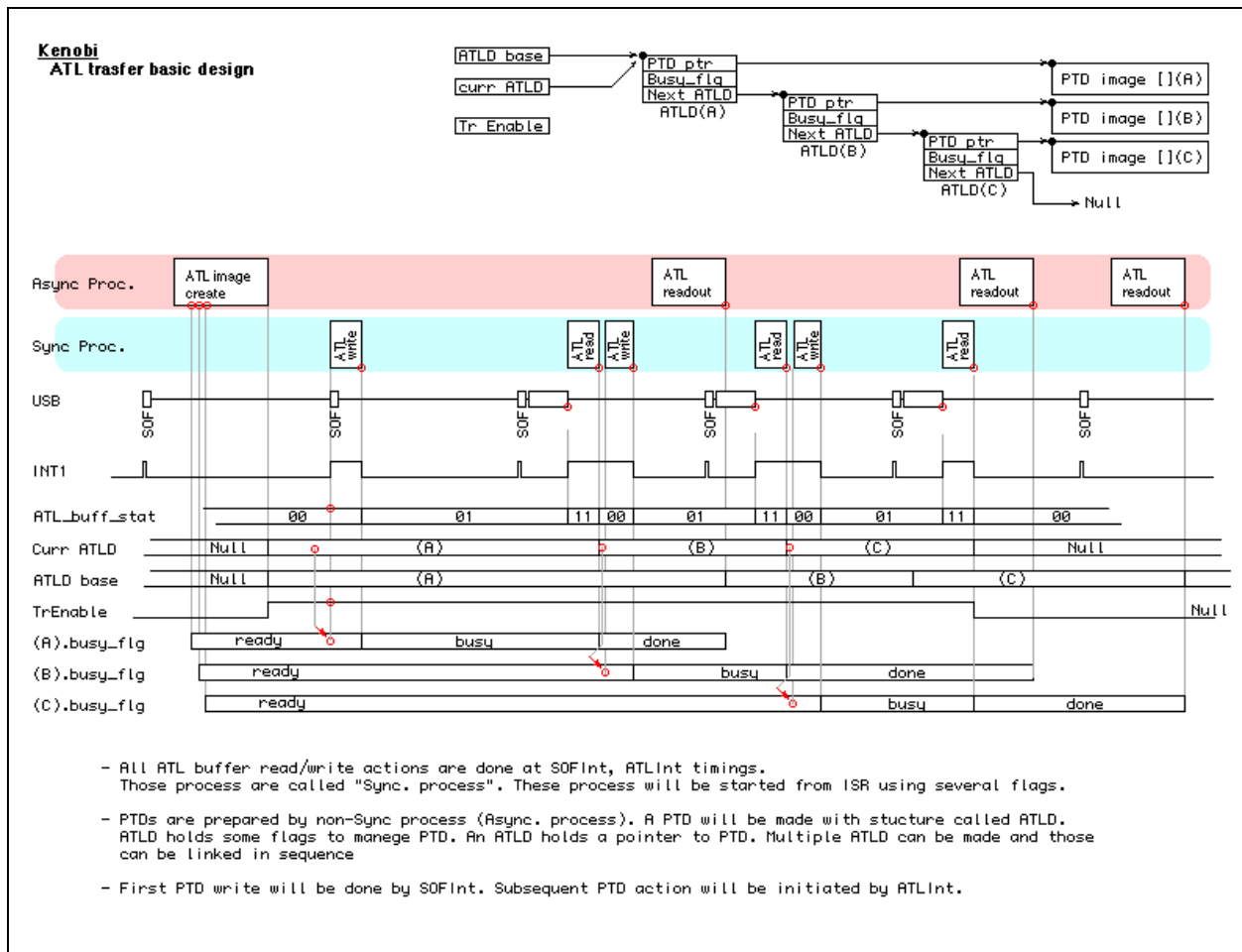


Figure 3-2: ATL Buffer Control Mechanism

3.4. ITL Buffer Control Mechanism

The ITL buffer management mechanism is very simple. To ensure the buffer maintenance service in real-time, all processes are done as synchronous process, except initialization and start/stop processes.

Kenobi2 uses a minimum of 1764 x 2 bytes buffer in a program. Its function is the same as that of the cache for ISP116x's ITL buffers.

Figure 3-3 shows the buffer filling timings. Red lines indicate the timing for the application buffer filling from the file, and blue lines indicate timing for the ITL filling.

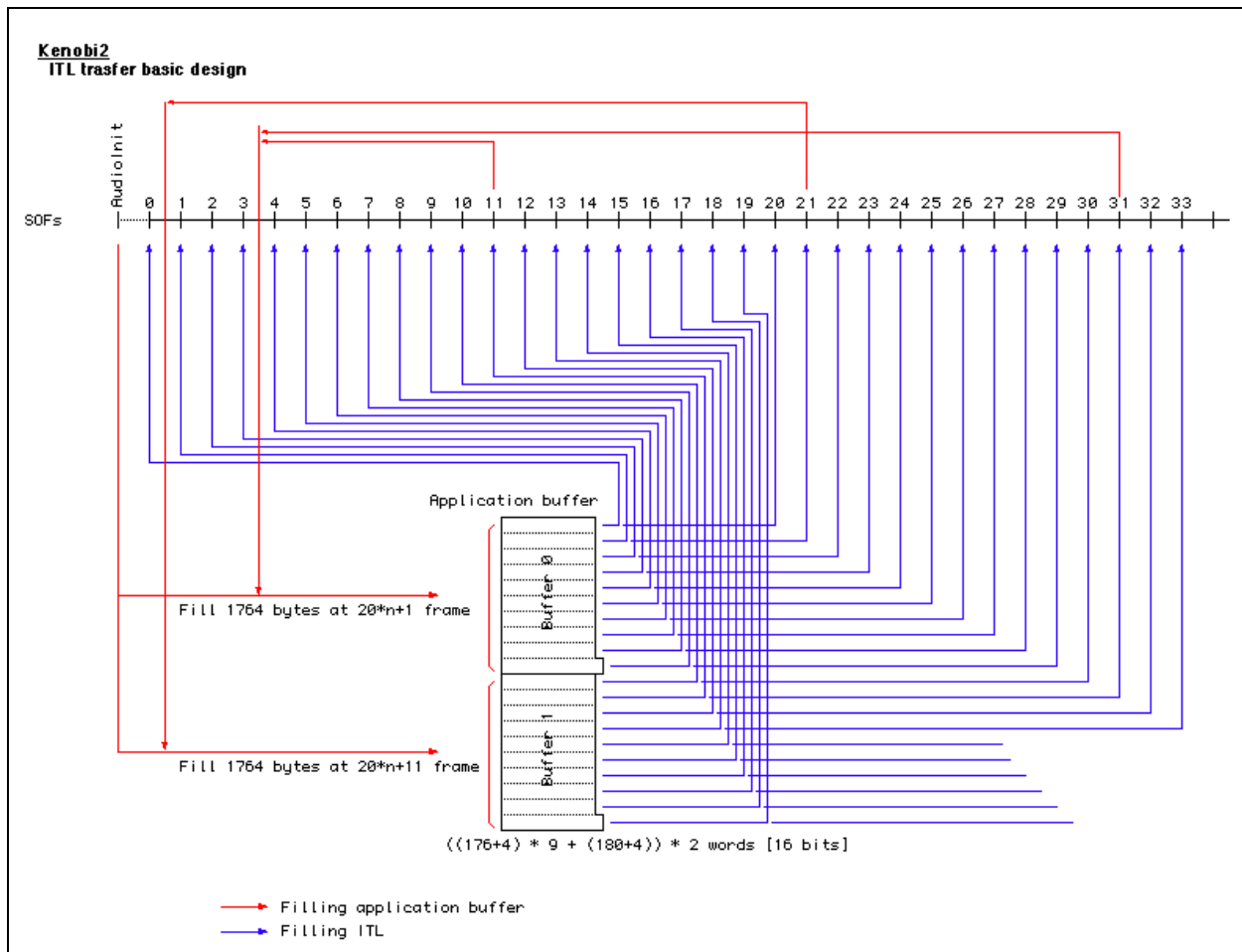
Kenobi2**Rev. 1.07**

Figure 3-3: ITL Buffer Control Mechanism

4. Limitations

- Audio playback is corrupted by interrupt service delays. Kenobi2 is a DOS program. Therefore, it is very difficult to guarantee the response time for interrupts. Sometime you will notice long interrupt service delays because the operating system is handling other tasks. For example, it will happen with simultaneous printer operation. The file (disk) access may cause frequent and long interrupt service delays.

If the delay is very short, the USB audio speaker system conceals the noise.

- Kenobi2 assumes that the audio file and the printer device are present when the program is launched and therefore, starts searching for them.
- Does not provide hub support. No USB topology management mechanism is implemented.
- No scheduling mechanism is implemented. No USB driver architecture design is used. This is because Kenobi2 is a monolithic program for just checking the functionality of the ISP116x. It just emulates driver/ClassDriver protocols.

5. References

- *ISP116x Full-speed Universal Serial Bus single-chip host and device controller datasheet*
- *Universal Serial Bus Specification Rev. 2.0 (full speed section)*
- *ISP1161x PCI Evaluation Board User's Guide*
- *ISP1161 PC Eval Kit User's Guide.*