TECHNICAL REPORT

USB Audio Playback Peripheral
(USB-APP) UDA1331H

APPLICATION NOTE 1

V1.2

DML97001

USB Audio Playback Peripheral (USB-APP) UDA1331H

APPLICATION NOTE 1
v1.2

Author:

S. Serper
Philips Semiconductors, Inc.
BLITS, Digital Media Laboratory
Sunnyvale, CA
USA

Keywords:

Universal Serial Bus (USB)
Digital-Analog Converter (DAC)
UDA1321
UDA1331H
USB-DAC configuration editor
UniCoDes © 1999-2000
Windows™ 98
iMac™

Date : June 14th, 2000

Abstract

This application note describes the installation and application of the Philips Semiconductors USB Audio Playback Peripheral, UDA1331H for the Universal Serial Bus (USB). Specific information is given about hardware and software requirements necessary to make the USB-Audio function work in a USB-enabled computer system.

Notice: *Windows™ 95, Windows ™ 98 and Windows™ NT are registered trademarks of Microsoft ® Corporation.*

USB compliant conform USB specifications v1.0

Purchase of Philips I^2C components conveys a license under the I^2C patent to use the components in the I^2C system, provided the system conforms to the I^2C specifications defined by Philips.

Table of Contents

1. Introduction

1.1 General Information

This application note describes the installation and use of the Philips Semiconductors USB-APP UDA1331H evaluation board. It furthermore explains the application of the USB-APP in an environment as a USB-Audio Device. A detailed Application note is available for explanation of the detailed aspects of the device.

The UDA1331H is essentially the same product as the UDA1321 (USB-DAC) but using an external (E)PROM for it's firmware instead of integrated firmware. This allows for easy upgrades and optimizations of the system during the time that the Windows™98 operating system is being changed. The ongoing firmware development for the UDA1331H will result into a several final firmware versions for use with the UDA1331H chip. Windows™98 compatible firmware will be integrated in the UDA1321 for a single chip solution. The UDA1321 will become available in SO28 package (UDA1321T), SDIP32 package (UDA1321PS), and QFP64 package (UDA1321H). The QFP64 versions of the UDA1331H and UDA1321H are pin compatible.

This application note will explain the Universal Serial Bus only in a few brief words (see section 1.2). If you want (or need) any further information or specific details we refer you to the USB Specification, to the USB Audio Device Class Description and other documents. You will find details in Chapter 4 - References.

The application of the USB-APP, UDA1331H requires a USB enabled PC running offical releases of Windows™98. This is USB specification related and must be checked before evaluation can start. Updates of the firmware will be available through the standard distribution channels.

NOTE :

The UDA1331H can not be used with other or earlier Windows operating systems. These operating systems do not provide the drivers and system support, USB Audio Class and Streaming Class drivers necessary to operate audio devices attached to the Universal Serial Bus.

To achieve full system functionality it is necessary that the PC hardware, the PC operating system, the installed driver software and the µC firmware of the chip work flawlessly together.

Legal usage of the Philips USB-AUDIO DACs:

It is very important that you are aware that use of one of the four internal maps is legal if you are the legal owner of the internal map. In any other case, you may **NOT** use any of the internal maps, other than for test purposes ! Otherwise you can and will be prosecuted to the maximum extent possible under law and you can count on severe civil and criminal penalties. So, if you are not a legal owner of one of the internal maps, you must implement your own configuration by means of an external I²C-EEPROM and program your companies VID as explained in application note 2 of the UDA1331H. The internal maps are dedicated to either Philips products or customers of Philips who have a legal agreement / contract to use one of the internal maps.

1.2 Universal Serial Bus Overview

The Universal Serial Bus (USB) is a “medium speed” serial bus designed to be part of every computer system (i.e. desktop as well as laptop, home as well as business …). USB is a royalty free bus that was invented for PC’s with basically two important goals in mind :

- To allow installation of additional hardware without opening the computer case
- To allow easy and intuitive installation and configuration of additional Hardware without additional user installable software (Plug‘n’Play).

The maximum speed of USB is 12 Mbit/s. This allows for a maximum “bulk” data rate in the order of 1 MByte/s. This is obviously plenty for peripheral devices such as keyboards, mice etc. that only generate data in small amounts “once in a while”. More than that, it is perfectly suited for Digital Audio Applications. Good quality Digital Audio requires data rates of less than 200 kByte/s (full-blown CD-Audio e.g. at 48kHz, 16 bit Stereo requires only about 13% of the bus bandwidth).

USB has the topology of a “Tiered Star”. The one (and only) host in the system is connected to the “Root Hub” (usually found right on the PC’s motherboard) and is responsible for all data traffic. This means that all data interactions are initiated by the host. The host polls all devices to find out which devices want to receive or transmit data. All other devices need to wait to be polled, they can not initiate data traffic by themselves.
The Root Hub can have one or more so called “Downstream Ports” (since they are facing away from the Host). In the simplest form the USB has only one other USB device connected directly to one of the Root Hub’s downstream ports.

Figure 1-1 below gives the simplest USB topology possible. It contains a ‘Host Processor’ , the ‘Root Hub’ and a single USB device connected to one of the Root Hub’s ports. The USB-Device could in this case e.g. be the UDA1331H evaluation board.

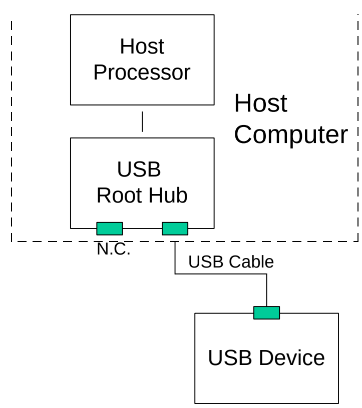


Figure 1-1 *Simplest possible USB Topology*

The ‘Host Processor’ is responsible to control all USB ‘Data Traffic’. It can send e.g. digital audio data on the bus to a specific USB audio device which is capable of playing this audio data and receiving control data to control e.g. the volume settings.

An additional USB Hub device is only needed to build more complex topologies, it doesn’t add real functionality to a USB environment.

Figure 1-2 shows a typical, more complex USB topology :

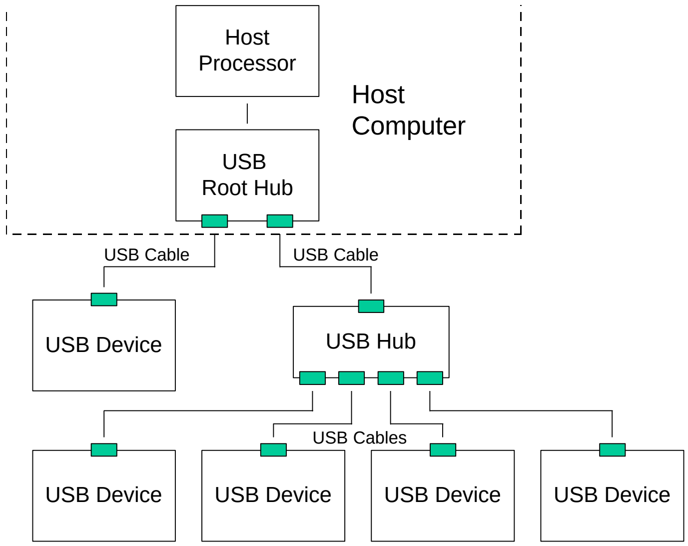


Figure 1-2 *Typical, more complex USB Topology*

There is a whole variety of USB devices that will make their appearance in the market in the near future. The most popular applications are

- Keyboards
- Mice
- Joysticks
- Monitor
- Audio / Speakers
- Printers
- Scanners
- Video Conferencing Cameras
- Modems / Phones

Given these applications a typical "real life" application could look like the example shown in Figure 1-3.

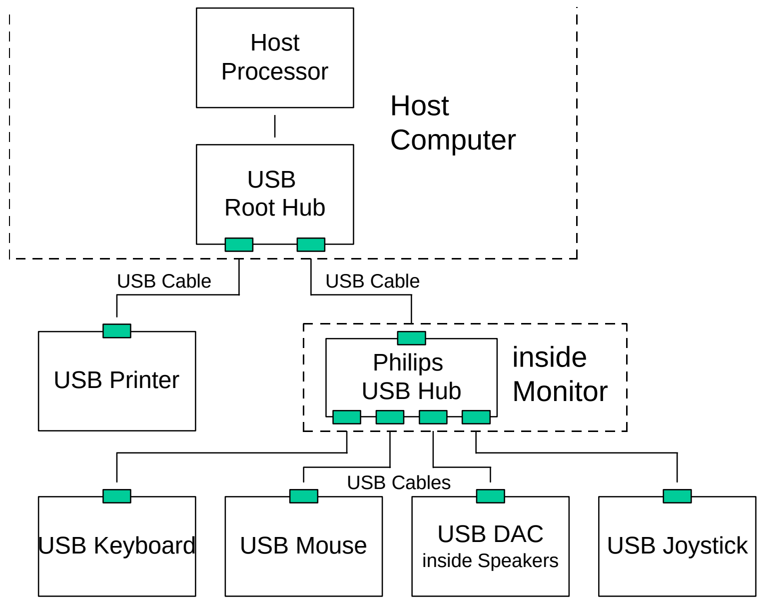


Figure 1-3 *Complex "Real Life" USB Topology*

The maximum number of devices that can be connected to the USB is limited to 127 (using 7 address bits. Remark : Since the Root Hub already counts as one device the number of external devices on the bus is actually limited to 126). Given the fact that all these devices would have to be *outside* of the computer case this number should be sufficient for most computer systems.

One of the interesting features of USB is that it allows for devices to be either

- Self powered or
- Bus powered

Self powered devices have some kind of connection to an external power source, e.g. to a DC power supply or a mains power outlet. They draw all their current from this external power source whereas the USB connection delivers only regular data and control information.

Bus powered devices on the other hand do not have any connections to external power sources. They draw all the necessary current through the USB cables from the hub that they are connected to. However, the maximum current that a single device can draw is limited to 500 mA (thus the total power consumption is limited to 2.5 W). This is of course only possible if the hub is self powered and allows attached devices to pull the mentioned current. To ensure that the mentioned limits are not exceeded a device has to start up in a "low power mode" with a maximum of 100 mA. It then negotiates whether the hub can provide the necessary current or not. It is in consequence possible that a desired bus topology is not possible because of certain power / current constraints of the various devices involved. It's within the responsibility of the operating system to notify a user of these conflicts.

There are basically 4 different “Data Types”. These are :

- Isochronous
- Control
- Interrupt
- Bulk

Isochronous (“Equally spaced in time“) Data Transfers are used for devices where data delivery to or from the host is absolutely time critical. These are e.g. Audio or Video Conferencing devices where the data is basically “useless” when not delivered in time. Devices that need isochronous data delivery need to negotiate a guaranteed time slot from the host. Once granted the specific time slot will be reserved whether actual data needs to be transmitted or not.

Control Data is usually not as time critical as audio or video data. For that reason control data is being transmitted “asynchronously”, meaning that it will be transmitted “on demand only”. Control data does not have a guaranteed time slot and can thus experience a delay.

Interrupt data are actually not really interrupts. Since all data traffic has to be initiated by the host devices don’t have a chance to announce that they need service. Fact is that the host needs to poll the different devices that can generate interrupt data to find out if they need service.

Bulk data is the transfer of a large amount of data. This could e.g. be the data transferred to a printer in order to print a postscript or bitmap file. This kind of data transmission is not time-critical and will not be handled using “isochronous” transfer. It will actually be used to “stuff” whatever is still open during a certain transmission time slot. This means that if the bus is already heavily loaded with isochronous data transfers like audio or video it can take a very long time to print a single page. If there is no load at all on the bus data rates of about 1 MByte/s can be achieved.

A specific feature of USB devices is that they can actually consist of more than one “real devices”. They are for that reason called “compound devices”. The USB-APP is a good example since it e.g. hosts an audio device as well as a Human Interface Device (HID) that is used to control the volume, bass and treble. All these specific devices need to negotiate their own special kind of data traffic and their own time slot with the host.

Every single device on the USB can have one or more so-called “Endpoints”. Endpoints are the real “data sources and sinks”. Every device must have at least ONE endpoint (using the internal number ID=0) which is used to communicate with the host. Endpoint zero contains the descriptors that explain the type of device and the data transfer needs to the host. On top of that, a device can have up to 15 more endpoints that communicate using different types of data transfer. Please note that the actual maximum number of endpoints in a system is 127 x 16 = 2032.

1.3 Definitions, Acronyms and Abbreviations

The following explains a number of terms and acronyms that are used in this document.

ADAC	Asynchronous Digital-to-Analog Converter
ASR	Audio Sample Redistributor
DAC	Digital to Analog Converter
Descriptor	Registers inside the chip containing descriptive information about the device
Device Class	Collection of devices that share certain properties.
GPIO	General Purpose Input / Output
EEPROM	Electrical Erasable Programmable Read Only Memory
Endpoint	Actual final data sink or source in a device. Every device has at least one endpoint.
Hub	A USB device that has one upstream connector (towards the Host) and a number of downstream ports (usually between 4 and 7). A Hub Device enhances the number of USB devices (including other hubs) that can be connected to the USB. It does not add any further functionality to the bus.
Memphis	Microsoft's internal code name for Windows™ 98 during the development cycle.
Port	USB connection on a Hub Device, enabling one USB Device (possibly a Hub) to be connected.
USB	Universal Serial Bus
USB-APP	All parts of the implementation of the USB-APP including the hardware and the software/firmware part
Windows™ 98	Microsoft® Operating System, upgrade to Windows™ 95, scheduled for release in Q2/1998, beta versions have been available to developers since about 7/1997

See additionally the Terms, definitions, acronyms and abbreviations section as given in the USB specification version 1.0 November 1995 for those terms not mentioned in this section.

2. Installing the USB-APP Evaluation Board

To ensure that the USB-APP Evaluation Board is perfectly working a number of things have to be checked and possibly adjusted :

- The hardware of the PC you intend to use as a host computer system,
- The version of the operating system that your host computer system is running,
- The version of the firmware inside the (E)PROM on the evaluation board.

As pointed out earlier USB is a new and emerging technology that is continuously developing so that the compatibility between versions is not guaranteed. Continuous efforts of all parties involved will guarantee that this problem is solved when the final version of Microsoft Windows98 is released to the public.

The following paragraphs give details about the installation procedure.

2.1 PC Hardware Requirements

The PC used to install the UDA1331H evaluation board needs to be based on a motherboard that has USB support and on which the USB circuitry is fully assembled. These motherboards first appeared in the second half of 1996 and are now basically standard. Accordingly chances are that you will find an assembled and enabled USB port on a new computer with a 133 MHz Pentium Processor or better. It is possible that the USB option on the motherboard is assembled but still needs the connectors to the outside to be hooked up (non ATX-type motherboards). These should be available for purchase in computer stores.
Most Desktop Systems have two USB ports in the back, newer laptops usually offer one USB port.

2.2 PC Software Requirements

It is necessary that the PC, to run the USB hardware, has the Microsoft® Windows™98 operating system installed. This may be either the First Edition (FE), Gold, or Second Edition (SE).
Download USB tools form the USB-IF forum to be able to evaluate your test setup. For URL, see chapter References.

2.3 Board Installation

Provided it has been checked that the PC operating system and the chips firmware are compatible the installation is very simple and intuitive. After all this is one of the key ideas behind the USB concept.

There are so called “A” and “B” type USB connectors. Every USB cable must have an type A connector on one side and a type B connector on the other. Type “A” connectors are used for “Downstream Ports” (facing away from the host computer - to be found e.g. on the back of a USB equipped desktop computer) whereas type “B” connectors are used on “Upstream Ports” (facing towards the host computer).

Plug the type B connector of a USB cable into the USB port connector on the evaluation board. In case of doubt, just try it, there is only one way to do it. Now connect the other side of the USB cable (the type A connector) into the USB port on the back of your USB enabled computer. It is not necessary to connect any power source to the USB-APP since the device can be bus-powered (meaning that all necessary supplies are provided through the USB cable - please check the jumper setting on the board for correct setting). If you already have other USB devices connected to your system and the usual two USB connectors on your desktop system (one on laptops) are already taken you can connect the USB-APP into a downstream port of any hub device connected to the system.

As soon as the USB-APP is connected to the system the host is being notified of the existence of a new device. The operating system interrogates the new device and makes an attempt to install the necessary software from its internal data base. The user will see a number of message boxes that announce the fact that new hardware has been found and that the software for it is being installed.

The evaluation board can operate in bus-powered and self-powered mode. When bus-powered, no external power supply is needed and power from the PC is used. In self-powered mode a DC power supply can be used for supplying 3.3V or 5V/3.3V directly to the board. The UDA1331H can be used in bus-powered mode (handy for test purposes), but since the USB-DAC and USB-APP do both not meet the suspend current budget as specified with the USB specification, you will not be able to pass the USB-IF tests during Plugfests with such a product. You must therefore use the USB-DAC and USB-APP as self-powered devices.

A headphone can be connected directly to the analog output of the evaluation board but the output can also be connected to a line input of a standard loudspeaker system.

Remark : You will probably need to have the Windows™ 98 CD in the CD-ROM drive since the operating system will look for the latest drivers in that location. Make sure to have the CD handy.

Since the USB-APP is a Composite Device - consisting of an Audio Device as well as a HID control device - the installation will happen in stages announcing first the Philips USB System itself followed by a Composite Device, then an Audio Device and finally a HID device. Once all these installations have been successfully completed, the USB-APP should be ready to play audio. An easy way to verify this is to play e.g. a wave file through the USB devices (using the “Media Player” application that comes with the operating system). The music should be playing through the speakers attached to the USB-APP evaluation board now.

2.4 In case of problems

The following are a number of common problems that have been observed. As pointed out earlier they are in most cases related to incompatibilities between different versions of the Windows™ 98 operating system, the USB-APP drivers and the firmware belonging to the UDA1331H.

One specific check that is generally advisable is to check the entries in the “Control Panel -> System-> System Properties->General “ section of the Windows™ operating system registry. The different USB devices need to be listed there and should have no remarks (such as exclamation marks or red tags). In case of the USB-APP you need to find at least one entry for the USB Root Hub, one entry for the USB Audio Device and one entry for the Human Interface Device (which is part of the audio device and used for functions such as volume, bass and treble control).

If you can’t find the mentioned devices or they are tagged as not being fully functional you need to reinstall the USB-APP or investigate further details (see below).

2.4.1 “USB Device not responding”

This message appears on the screen during the interrogation process when the operating system has detected a new device on the USB but is not able to properly receive a description of the device. Since the OS doesn’t know who the new client is, the necessary software can and will not be installed. There are various ways you can try to solve this :

- Verify once more the Operating System version number and the USB-APP firmware version number. Make sure that they are compatible. Contact the Philips Semiconductors support if you don’t have enough information or can’t match the versions.
- Make sure that you have the correct drivers from the Window™ 98. If you had an earlier version of the driver installed you will have to explicitly specify that you want newer drivers to be used. You can do so by checking the driver version in the “Control Panel -> System-> System Properties->General “ section and then ask to “Update Driver”. You will have to specify the location on the latest version of the Window ™ 98 CD.
- Reboot the host computer system and try again to attach the USB-APP once the OS is up and running.
- Leave the USB device attached and reboot the host computer system.
- Change the bus topology - if you have been using an extra hub eliminate it. If you didn’t use an extra hub try and insert a hub into the system now.
- Try and use another port on the host computer system or on the hub (if you are using one). A working port on a hub is frequently indicated through a lit LED.

2.4.2 Device is recognized but there is no sound

If the device has been recognized but there is no sound, it is possible that your system audio mixer is at a very low volume or muted. You can check that by accessing the “Programs-> Accessories->Multimedia->Volume Control” application on the “Start” Menu. Make sure that the “Wave” Device is not muted and at a reasonable level.

Should this not help it is possible that you still have an old ‘legacy” sound card in your system (which might not be hooked up to speakers or an amplifier anymore). Please refer to the following section 2.4.3 how to handle this case.

2.4.3 Sound is still coming from the ‘old’ sound system

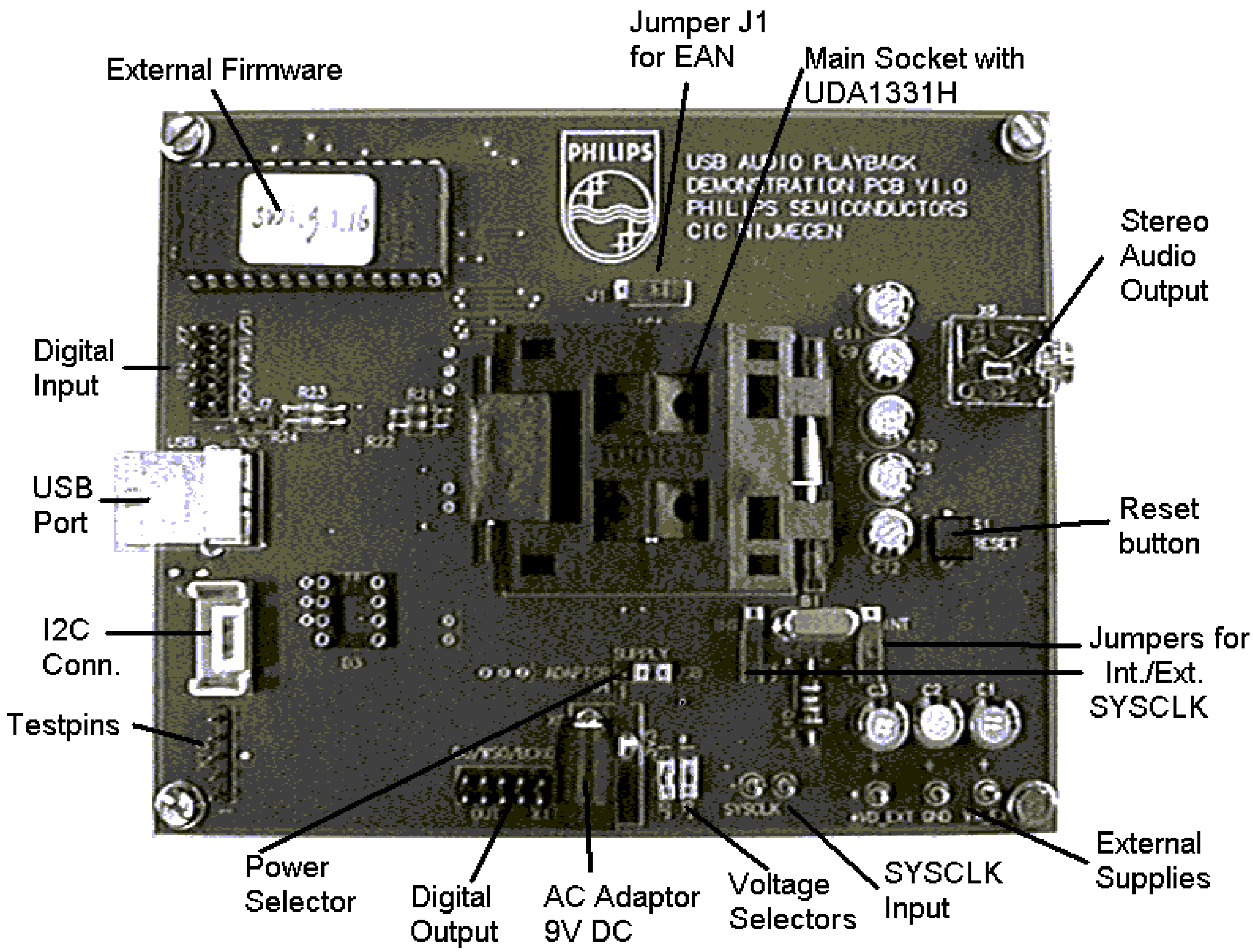
If you installed the USB-APP evaluation board on a computer system that already had a sound card (e.g. Soundblaster or similar) installed it is possible that the sound still plays through the old channels instead of the new USB Audio system. It is in this case necessary to change the routing inside the Windows ™ operating system. This can be done in the “Settings->Control Panel -> Multimedia” section under the Audio tab in the “Preferred Device” slot . Choose the USB Audio System as your playback device.

The other (more radical) possibility is to delete the “legacy” audio device from the computer system. This is again done in the “Control Panel -> System -> System Properties -> Device Manager” dialog box. Select the device(s) by clicking the right mouse button on them and choose “Delete”. You will be prompted whether you really want to permanently delete the specific device from the system. Boldly answer “Yes”.

Once all the “legacy” audio devices are deleted try again if the USB Audio Device now plays your audio. If this is still not the case you might need to reboot the system. When the system is powered down it is a good time to remove the “legacy” hardware from your system. Should you decide not to remove the hardware and your hardware is Plug’n’Play compatible your system will detect the “legacy” sound card again and will prompt you to install drivers for it. In this case it is necessary that you choose the “Don’t install a driver” option. Otherwise the device(s) will be installed again.

3. Hardware Description

The following shows a descriptive Photograph of the USB-APP (USB-DAC) evaluation board :



3.1 Input and Output Connectors

The UDA1331H evaluation Board has the following connectors :

1	USB Type B Input Connector	Connects the boards upstream port to a downstream port on the bus (e.g. a hub device)
2	Mini Stereo Jack for Analog Audio Output	Connects the stereo analog output signal to an stereo audio amplifier
3	DC Voltage Adaptor	External DC power supply (3.3V)

3.2 Jumpers, Headers and Switches

1	J1	Jumper Ext./ Int. ROM	Selects whether the device boots from internal ROM (UDA1321H) or from the external EPROM (UDA1331H)
2	J2	Jumper Ext./ Int. System clock	Selects whether the device is running from the on-board crystal or of an external system clock by disconnecting the crystal
	J3	same as J2	Other side of the crystal oscillator
3	J4	Jumper Ext. / Int. Power	Selects whether device is self- or bus-powered
4	X1	10-pin Header	Provides Digital Audio (I2S) Output signals to interface e.g. to an external DSP
5	X7	10-pin Header	Provides Digital Audio (I2S) Input signals to interface e.g. to an external DSP
6	X8	4-pin Header	I2C signal SDA and SCL
7	X10	5-pin Header	Test signals SHTCB, RTCB, TC
8		2-pin Header	To feed in an external SYSCLK signal (48MHz)
9		3 pins	To feed in 5.0 Volt or 3.3 Volts from an external power supply
10	S1	Reset Switch	“Hard” Resets the USB-DAC

3.3 Chip pin out

The following table describes the pins of the QFP64, the SO28 and the SDIP32 versions of the UDA1321 and the QFP64 version of the UDA1331H :

Name	Pin QFP64	Pin SO28	Pin SDIP32	I/O	Pin Description
GP2/DO	10	1	1	I/O	General Purpose Pin/ Data Output to external DSP
GP3/WSO	13	2	2	I/O	General Purpose Pin/ Word Select Output to external DSP
GP4/BCKO	14	3	3	I/O	General Purpose Pin/ Bitclock Output to external DSP
SHTCB	15	4	4	I	Shift clock TCB (active high, digital)
NC	-	-	5	-	Not connected
D-	17	5	6	I/O	Negative Data signal of the differential bus
D+	20	6	7	I/O	Positive Data signal of the differential bus
VDDI	25	7	8	-	Digital supply for digital core
VSSI	29	8	9	-	Digital ground for core
VSSE	30	9	10	-	Digital ground for I/O pads
VDDE	32	10	11	-	Digital supply for I/O pads
NC	-	-	12	-	Not connected
VSSX	36	11	13	-	Ground for crystal oscillator
XTAL1	37	12	14	I	crystal oscillator input (analog)
XTAL2	38	13	15	O	crystal oscillator output (analog)
VDDX	39	14	16	-	Digital supply for crystal oscillator
NC	-	-	17	-	Not connected
VREF	42	15	18	I	Reference voltage output pin
VSSA	44	16	19	-	Analog Ground
VDDA	45	17	20	-	Analog Supply
VOUTR	46	18	21	O	Voltage output right channel (analog)
VSSO	49	19	22	-	Ground for Operational amplifiers
VDDO	51	20	23	-	Supply for Operational amplifiers
VOUTL	53	21	24	O	Voltage output right channel (analog)
TC	55	22	25	I	Test control pin (active high, digital)
RTCB	61	23	26	I	Asynchronous reset TCB (active high, digital)

GP0/BCKI	64	24	27	I/O	General Purpose Pin/ Bitclock Input from external DSP
NC	-	-	28	-	Not connected
GP5/WSI	2	25	29	I/O	General Purpose Pin/ Word Select Input from external DSP
SCL	3	26	30	I/O	I2C Clock Line
SDA	4	27	31	I/O	I2C Data Line
GP1/DI	7	28	32	I/O	General Purpose Pin/ Data Input from external DSP

Pins only found on the QFP64 package (UDA1331)

Name	Pin QFP64	I/O	Pin Description
P0.0	56	**I/O**	µC Data Port 0, Pin 0
P0.1	57	**I/O**	µC Data Port 0, Pin 1
P0.2	58	**I/O**	µC Data Port 0, Pin 2
P0.3	59	**I/O**	µC Data Port 0, Pin 3
P0.4	60	**I/O**	µC Data Port 0, Pin 4
P0.5	62	**I/O**	µC Data Port 0, Pin 5
P0.6	63	**I/O**	µC Data Port 0, Pin 6
P0.7	5	**I/O**	µC Data Port 0, Pin 7
P2.0	11	**I/O**	µC Data Port 2, Pin 0
P2.1	12	**I/O**	µC Data Port 2, Pin 1
P2.2	18	**I/O**	µC Data Port 2, Pin 2
P2.3	19	**I/O**	µC Data Port 2, Pin 3
P2.4	21	**I/O**	µC Data Port 2, Pin 4
P2.5	22	**I/O**	µC Data Port 2, Pin 5
P2.6	23	**I/O**	µC Data Port 2, Pin 6
P2.7	24	**I/O**	µC Data Port 2, Pin 7
/EA	6	**I/O**	External access (active low)
/PSEN	8	**I/O**	Program Store Enable (active low)
ALE	9	**I/O**	Address Latch Enable (active high)

3.4 Schematics

See the application diagram delivered with the UDA1331H evaluation board(s).

3.5 List of Active Components

3.5.1 Integrated Circuits

Name	Description	Remark
IC1	USB-APP or USB-DAC QFP64 package	UDA1331H or UDA1321H
D1	Latch for configuration EPROM	74HCT1373D
D2	EPROM for external µController code	suggested EPROM's are EEPM27128 or AM27C256-70
D3	Socket for external configuration EEPROM	(not stuffed)
D4	Voltage Regulator 5V/3.3V	LK115033 or ZR 78 L033

3.5.2 Transistors

Name	Value	Description	Remark
V1		Startup circuitry transistor	pnp transistor
V2		Startup circuitry transistor	npn transistor

3.6 List of Passive Components

The following table gives a list of all active and passive components used on the UDA1331H evaluation board. Please note that the components marked with an asterisk (*) are for test purposes only and will not be required in a final application.

3.6.1 Resistors

Name	Value	Description	Remark
R1	47 Ω	System clock input series resistor	
R2	100 Ω	Output resistance Right	
R3	100 Ω	Output resistance Left	
R4	10k	Resistor to ground (DC-path)	
R5	10k	Resistor to ground (DC-path)	
R6	10k	Pull-up resistor for I2C SCL line	
R7	10k	Pull-up resistor for I2C SDA line	
R8	4.7k	Pull-up resistor for EAN	
R9	1.5k	D+ pull-up resistor for full speed devices (part of Start-up circuitry)	
R10	0 Ω	Pull-down for pin SHTCB	wire to GND
R11	0 Ω	Pull-down for pin TC	wire to GND
R12	0 Ω	Pull-down for pin RTCB	wire to GND
R13	22 Ω	Serial resistor for current limitation to D-	
R14	22 Ω	Serial resistor for current limitation to D+	
R15	1 Ω	DC supply filtering for pin VDDA	
R16	1 Ω	DC supply filtering for pin VDDI	
R17	1 Ω	DC supply filtering for pin VDDE	
R18	1 Ω	DC supply filtering for pin VDDO	
R19	1 Ω	DC supply filtering for pin VDDX	
R20	1 Ω	DC supply filtering for VDD EEPROM	
R21	22k	Startup Circuitry	
R22	22k	Startup Circuitry	
R23	22k	Startup Circuitry	
R24	22k	Startup Circuitry	

3.6.2 Capacitors

Name	Value	Description	Remark
C1	100µF	DC Filtering/ Smoothing capacitor for VA	16Volt
C2	100µF	DC Filtering/ Smoothing capacitor for VC	16Volt
C3	100µF	DC Filtering/ Smoothing capacitor for VD	16Volt
C4	22pF	HF filtering on D-	63 Volt
C5	22pF	HF filtering on D+	63 Volt
C6	12pF	Oscillator capacitor	63 Volt
C7	10nF	Oscillator capacitor	63 Volt
C8	47µF	DC Supply filtering capacitor for pin VDDA	16 Volt
C9	47µF	DC Supply filtering capacitor for pin VDDO	16 Volt
C10	47µF	Output AC coupling Right	16 Volt
C11	47µF	Output AC coupling Left	16 Volt
C12	47µF	DC Filter capacitor for Reference voltage	16 Volt
C13	4.7pF	Oscillator capacitor	50 Volt
C14	100nF	HF supply filtering capacitor for pin VDDA	63 Volt
C15	100nF	HF supply filtering capacitor for pin VDDI	63 Volt
C16	100nF	DC supply filtering capacitor for pin VDDI	63 Volt
C17	100nF	DC supply filtering capacitor for pin VDDE	63 Volt
C18	100nF	DC supply filtering capacitor for pin VDDE	63 Volt
C19	100nF	DC supply filtering capacitor for pin VDDO	63 Volt
C20	100nF	DC supply filtering capacitor for pin VDDX	63 Volt
C21	100nF	DC supply filtering capacitor for pin VDDX	63 Volt
C22	100nF	DC Filter capacitor for Reference voltage	63 Volt
C23	100nF	DC Filter capacitor for VDD EEPROM	63 Volt
C24	100nF	DC Supply voltage filter capacitor	63 Volt
C25	100nF	DC Supply voltage filter capacitor	63 Volt
C26	10nF	USB Supply filtering	50 Volt
C27	10nF	USB Ground filtering	50 Volt
C28	100nF	HF filtering capacitor	50 Volt
C29	100nF	HF filtering capacitor	50 Volt

3.6.3 Inductors

Name	Value	Description	Remark
L10	1.5µH	Oscillator inductor	
L11		Ferrite bead for FCC rule compliance	BLM32A07
L12		Ferrite bead for FCC rule compliance	BLM32A07
L13		Ferrite bead for FCC rule compliance	BLM32A07
L14		Ferrite bead for FCC rule compliance	BLM32A07
L15		Ferrite bead for FCC rule compliance	BLM32A07
L16		Ferrite bead for FCC rule compliance	BLM32A07

3.6.4 Crystal Oscillators

Name	Value	Description	Remark
B1	48MHz	3rd Overtone Crystal for Oscillator Circuit	

4. References

More information on the different topics can be found in the following publications :

[1] UDA1331H specification

[2] USB-Specification version 1.0 November 1995

[3] USB Device Class Definition for Audio Devices Version 0.9, draft 1 May ’97

[4] USB Human interface Device Class specification Version 1.00 draft #4, Dec ’96

[5] USB Human Interface Device Class Usage Table 0.9 June 11 1997

The versions mentioned above are the current versions at the time this document was printed and are NOT relevant for the UDA1331H.

A lot of the above and additional useful information can further be found on the following Internet web site that is fully dedicated to USB support and information :

http://www.usb.org

Check the Philips USB-AUDIO website frequently for updates on tools, application notes and other important information at:

http://www.semiconductors.philips.com/usb/products/audio

5. In case of Problems

Should you still experience problems in operating the Philips Semiconductors UDA1331H demonstration board we recommend that you get into contact with your local Philips Semiconductors Field Applications Engineer. The local Philips FAE will certainly be able to help you to get your system up and running.