TECHNICAL REPORT

USB AUDIO-CODEC UDA1325PS
EVALUATION BOARD
AND
APPLICATION DOCUMENTATION
VERSION 1.3

DML98021

TECHNICAL REPORT

USB AUDIO-CODEC UDA1325PS
EVALUATION BOARD
AND
APPLICATION DOCUMENTATION
VERSION 1.3

Author:

S. Serper
Philips Semiconductors, Inc.
BLITS, Digital Media Lab (DML)
Sunnyvale, CA
U.S.A

Keywords:

Universal Serial Bus (USB)
UDA1325PS
Analog-Digital Converter
Digital-Analog Converter
USB-CODEC configuration editor
UniCoDes © 1999-2000
Windows™ 98
iMac™

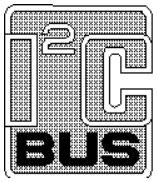
Date : June 15th, 2000

Abstract

This application note describes the installation and application of the Philips Semiconductors USB-CODEC solution UDA1325PS for the Universal Serial Bus (USB). Specific information is given about hardware and software requirements necessary to make the USB-CODEC work in a USB-enabled computer system.
DML98022 is the extended application note which describes the usage & programming capabilities of the UDA1325PS application board in more detail.

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

USB compliant conform USB specifications



Purchase of Philips I²C components conveys a license under the I²C patent to use the components in the I²C system, provided the system conforms to the I²C specifications defined by Philips.

© Philips Electronics N.V. 2000
All rights are 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 nor imply any license under patent- or other industrial or intellectual property rights.

Table of Contents

1. Introduction

1.1 General Information

This application note describes the installation and use of the Philips Semiconductors USB-UDA1325PS evaluation board. It furthermore explains the application of the USB-CODEC in an environment as a USB-Audio Device.

The type number UDA1325PS refers to the ROM-less package version of the UDA1335H. The UDA1335H is used with an external (E)PROM that holds it's firmware. This allows for easy upgrades and optimizations of the system during the time that the Windows™98 operating system is not officially released. The ongoing firmware development for the UDA1335H has resulted in a second generation firmware version for the UDA1325PS chip (SW1.1.1.19). This last version of the firmware is integrated into the UDA1325PS (and UDA1325H QFP64) /N106, and sold as a single chip solution without the external (E)PROM.

While a lot of the information in this document will be applicable to the UDA1335H too, for simplicity reasons in this document reference will only be made to the UDA1325PS. For detailed information in regard of the UDA1335H, please get the latest application notes (*DML98011.PDF* and *DML98012.PDF*) from your FAE or read chapter References how to download the files from the Philips internet-website.

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 please refer to the USB Specification, to the USB Audio Device Class Description and other relevant documents. You will find details about these documents in Chapter 5 - References.

The application of the USB-CODEC UDA1325PS requires a USB enabled PC running Windows 98. Beta versions and 'Release Candidates' of this operating system are being distributed in the hardware developer's community. The version to use is related to the UDA1325PS / N106 firmware releases and Philips can not give any guarantee that a specific firmware release will work with the pre-release operating systems version you are using. It is highly advisable to check whether for a version match before evaluation starts and to stay updated with the latest releases of Win98.

NOTE: The UDA1325PS, as most other USB solutions, cannot be used with other or earlier Windows operating systems. These operating systems do not provide the drivers and system support, USB Audio Class, Streaming Class and HID Class drivers necessary to operate audio devices attached to the Universal Serial Bus. Currently, the *iMac*$^{(TM)}$ is also developing drivers for the USB-AUDIO device class and their latest release (OS9) does support our USB-CODEC too. We will inform you later on further developments for this platform !

Legal usage of the Philips USB-AUDIO CODEC:

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 UDA1325PS. 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 Roo -Device could in this case e.g. be the UDA1325PS evaluation / demo board.

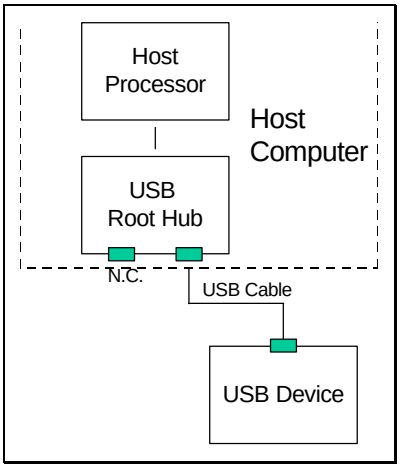


Figure 1-1 *Simplest possible USB Topology*

The ‘Host Processor’ is responsible to control all USB ‘Data Traffic’. It can send, for example, 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, for example, 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:

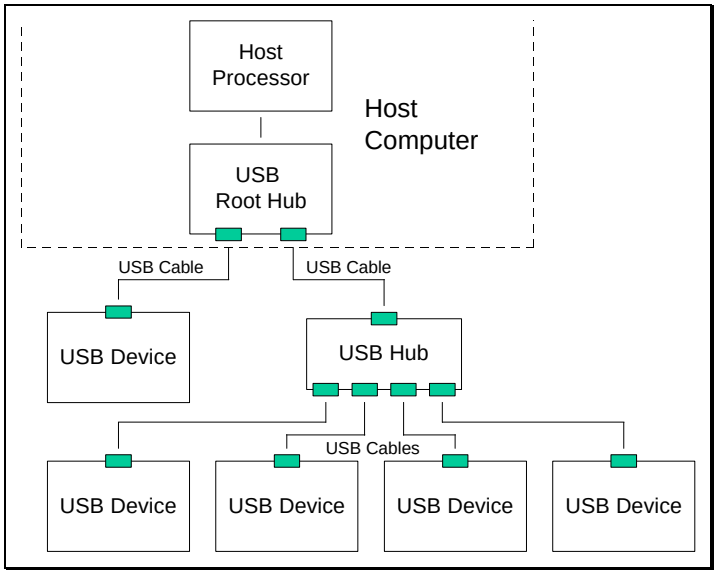


Figure 1-2 *Typical, more complex USB Topology*

There are 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
- Monitors
- 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.

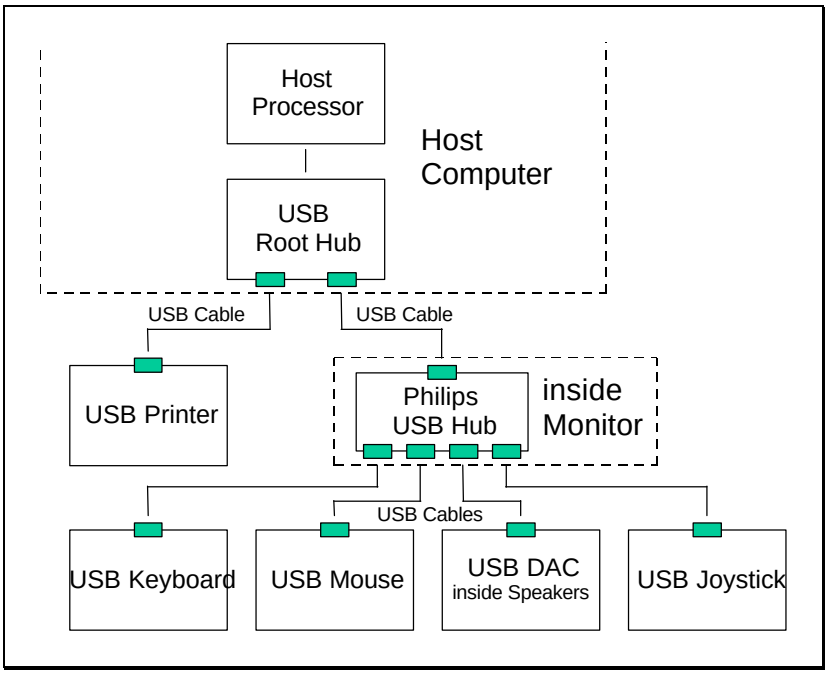


Figure 1-3 *Complex “Real Life” USB Topology*

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 to which they are connected. However, the maximum current that a single device can draw is limited to 500mA @ Vbus=5V (thus the total power consumption is limited to 2.5 W). This is, of course, is 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 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. The UDA1325PS application board provides the possibility to select whether you want to use an external power supply or the USB-power via jumper J8.

But there is an important issue regarding the selection of which power source you want to use for your application. If you want to use the USB-power, you will have to report this in your configuration descriptor via an I²C – EEPROM. Beside descriptor report, your whole application must comply to the suspend-mode specification. This specification forces all USB-powered devices to be able to run at a maximum of 500uA when this mode is activated by the USB-host.The current application board (v1.4) of the UDA1325PS has circuitry which enables you to switch off some power consuming items / components on the board. This enables us to reach approx. 570uA @ 3.30V(Vdd) at suspend-mode with the entire board. There are two problems here. The first one is obvious: we can not comply to the 500uA spec. The second one is the fact that the UDA1325PS / N106 has no way of a trigger available to use to switch off the power consuming parts on the application board. The crystal clock will be switched off when the suspend-mode is activated and it is actually a good indicator, but it is not possible to connect other circuitry on the XTAL1B and XTAL2B oscillator pins. The influence of the external circuitry (capacitance) connected to the crystal is critical for the oscillating characteristics. But, since the UDA1325H / N106 has a digital CLK output pin (pin 27), we can use this output which can serve as a trigger. Because of these two reasons, we do not promote the UDA1325x / N106 devices as bus-powered and we advise you to be considerate and **not** to use the N106 as bus-powered either.

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

- **I**sochronous
- **C**ontrol
- **I**nterrupt
- **B**ulk

Isochronous (“Equally spaced in time“) Data Transfers are used for devices where data delivery to or from the host is absolutely time critical. These include 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 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. The polling mechanism itself can consequently be described as “isochronous” whereas the interrupt data itself will only be transmitted “on demand”, meaning “asynchronously”.

Bulk data is the transfer of a large amount of data. This could 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 “fill” an opening in a time slot during a certain transmission. 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 consist of more than one “real device”. They are for that -CODEC is a good example since it can host an audio device as well as a Human Interface Device (HID) 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 Audio Converter.
CODEC	CODer – DECoder.
ASR	Audio Sample rate Redistributor .
DAC	Digital to Analog Converter.
Descriptor	Group of 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 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 bus. 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-CODEC	All parts of the implementation of the USB-APRP including the hardware and the software/firmware.
Windows™ 98	Microsoft ® Operating System, upgrade to Windows™ 95, released in June 1998.

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-CODEC Evaluation Board

To ensure that the USB-CODEC Evaluation Board is 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 chip 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 / successor versions 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 UDA1325PS 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 running the USB hardware has the Microsoft ® Windows™98 operating system installed. It is thus necessary to look carefully which version is installed and whether it complies with the firmware in the UDA1325PS. There should be no problems if you have received a demo board from Philips Semiconductors after beginning of May 1998. The complete type number of the UDA1325PS IC with firmware version 1.1.1.19 should be UDA1325PS / N106.

2.3 Board Installation

Once it has been checked that the PC operating system and the chip's 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 a 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-CODEC since the device is 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), you can connect the USB-CODEC into a downstream port of any hub device connected to the system.

As soon as the USB-CODEC is connected to the system the host is 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 database. 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 and in standalone modes. When bus powered, no external power supply is needed, and power from the PC is used. In standalone mode, a 3.3V stabilized DC power supply must be used.

A headphone can be directly connected to the evaluation board but it can also be connected to a standard loudspeaker system. In this case it is necessary to connect the "Line out" stereo jack on the evaluation board to the amplifier inputs of a stereo system or a pair of powered speakers.

Remark: You will probably need to have your Windows™ 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-CODEC 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 a USB Audio Device and finally a HID device. Once all these installations have been successfully completed the USB_CODEC should be ready to play audio. An easy way to verify this is to play 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-CODEC evaluation board.

2.4 In case of problems

The following are a number of common problems that have been observed. Experience shows that they are in most cases related to incompatibilities between different versions of the Windows™ 98 operating system, the USB-CODEC drivers and the firmware built into the USB-CODEC or wrong mixer volume settings.

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-CODEC, you need to find at least one entry under *System Devices* for the USB Root Hub. Furthermore there should be two entries in the *Sound, Video and Game Controller* section for the USB Audio Device and 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-CODEC or investigate further details (see below).

2.4.1 “USB Device not responding” or “Unknown Device”

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-CODEC 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 latest drivers from the Window™ 98 CD. 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 after the fact 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 where these new drivers can be found (this will generally be the Windows™ 98 CD.
- Reboot the host computer system and try again to attach the USB-CODEC 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.
- Unplug and re-plug the device. Make sure to give the operating system enough time to recognize the new device.

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” and the “Speaker” Devices are 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 paragraph (2.4.3) how to handle this case.

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

If you installed the USB-CODEC evaluation board on a computer system that already had a sound card (e.g. a Soundblaster™ type system or similar) installed, it is possible that the sound still plays through the old channels instead of the new USB 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. -> 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” (you can always reinstall it later if you have to).

Once all the “legacy” audio devices are deleted, try again to see 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.

2.4.4 Recording doesn’t work or the sound quality of recorded data is bad

To evaluate the recording capabilities of the device it is necessary to connect a suitable device to the device’s input. This could for example be the analog stereo output of a CD player (LINE).

Another possibility for no input signal during recording is that the USB-CODEC has not been selected as the preferred input device. This can be changed in the “Settings -> Control Panel -> Multimedia” section under the Audio tab in the “Preferred Device” slot. Choose the USB Audio System as your recording device. Some recording tools, like Cool Edit, provides there own audio-device selection routine. Be sure that the correct device is selected during recording.

If you are using a recording program other than the ‘standard’ Windows 98 programs (e.g. Syntrillium CoolEdit or SoundForge) you might as well have to change the settings inside the program rather than in the Control Panel. This can usually be done in the ‘Options’ menu of the software package. Look around for a possibility to adjust the hardware settings and select the USB device in there.

If you have a recording signal from the USB-CODEC but the sound quality is really bad (e.g. a lot of noise with some feeble sound in the background consider the following:

- Try to connect your source to the other input jack.
- Change the recording format in your recording software. A setting of 8kHz, 8-bit , mono will certainly disappoint you if you are hoping for CD quality. We recommend to start with 44.1 kHz, 16 bit , stereo. This is equivalent to CD quality and should give you a good representation of whatever your input source may be.

2.4.5 Digital CD-AUDIO Playback (Redbook protocol)

In some cases, you could desire to playback a CD-AUDIO via the digital channels (IDE-bus) instead of analog output of your CD-ROM player. In such case, complete the following steps:

- Go to *Multimedia Properties* and select the CD Music tab-field.

- If you enable the one and only check-box in that tab-field, you can use the digital playback feauture of your CD-ROM player. This will ensure that the audio-data is transferred to the IDE-interface. The “Redbook” protocol converts this data to the USB-bus and ultimatly, you will be able to playback the CD-AUDIO via the USB-CODEC.
- On some PC systems, the check-box is disabled. This is probably because of the fact that your CD-ROM is not Redbook-compatible.
- In other cases, the sound during playback is combined with a plopping or interrupted sound. This is not a problem of your USB-CODEC, but more likely the CD-ROM’s inability to keep up with the speed of the protocol. Microsoft has developed a small program for testing the compatibility of your CD-ROM player, called **RBTEST.EXE**. You should be able to download this program from Microsoft’s web-page.

2.4.6 Known bugs on the current evaluation board

Here is a list of known bugs on the corresponding version of evalution / demo boards:

Board version 1.4 :

- C14 should be < 2 uF instead of 47uF (see elco. at VrefDA pin)
- The Disconnect/Connect circuit has been revised, the new proposed circuit is **not** implemented on v1.4 of the demoboard. Ask your local FAE for the report **REP0021.PDF** or check the USB-AUDIO website for this file to utilize the latest circuit in your own USB-CODEC design. Read the report to understand why & how to use the new proposed solution.

3. Hardware Description

The following shows a descriptive scan of the USB-CODEC evaluation board (top-view):

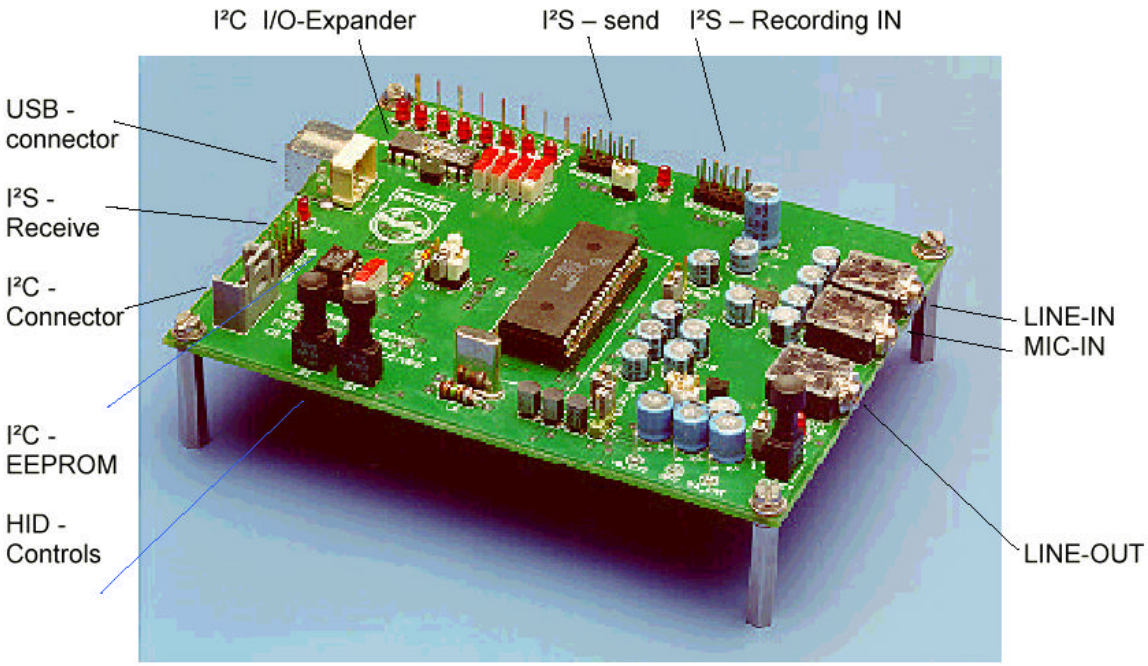


Figure 3-1 : The evaluation / application board of the UDA1325PS.

The schematic of the board is either delivered by means of a print-out together with the board or otherwise ask the respective file from your local FAE or download it from the Philips USB-AUDIO internet-website. The default jumper settings should be as displayed in the respective schematic.

Application note 2 (DML98022.PDF) describes the functions of these jumpers in more detail.

3.1 Input and Output Connectors

The UDA1325PS Evaluation Board has the following connectors:

1	X4	USB Type B Input Connector	Connects the boards upstream port to a downstream port on the bus (e.g. a hub device)
2	X7	Mini Stereo Jack for Analog Audio Output	Connects the right and left analog output signals to an audio amplifier
3	X3	Mini Stereo Jack for Analog Audio LINE Input	Connects the right and left analog input signals of a sound source to the system recording path
4	X5	Mini Stereo Jack for Analog Audio MIC Input	Connects the right and left analog input signals of a sound source to the system recording path
5	X8	I²C connector	Connects the I²C lines of the CODEC with Single Master Controller.

3.2 Jumpers, Headers and Switches

1	J1	Jumper for Selector	Selects GP-pin controlled selection of terminal or default (Line IN) if jumper is removed.
2	J2	Jumper Startup circuit	Selects whether the startup-circuit is active or not. Escpecially if 6-pins communication is configured, this jumper should be removed.
3	J3, J5 & J14	Jumpers for I/O expander address selection	Should be connected to ground (default).
4	J4	Jumper for Multiple write-mode	Only if ST24C16 is used, it is possible to enable set the multiple writing mode of the EEPROM. If not used set to GND
5	J6	Jumper for selecting Interrupt-input	Accoring the selected I/O communication, you will have to remove (6-pins) or insert (No I/O and 4-pins) the jumper.
6	J7	Jumper 5V or 3.3V	Selects whether VD+ is on 5V or 3.3V.
7	J8	Jumper Ext. / Int. Power	Selects whether device is self- or bus-powered
8	J9	Jumper Ext. / Int. Power	Selects whether device is self- or bus-powered
9	X1	10-pin Header	Provides Digital Audio (I²S) Output signals to interface e.g. to an external DSP
10	X2	10-pin Header	Provides Digital Audio (I²S) Input signals to interface e.g. to an external DSP or I²S source.
11	X4	4-pin Header	Provides USB communication with USB - host.
12	X6	10-pin Header	Provides Digital Audio (I²S) Input signals to interface e.g. to an external DSP
13	X8	4-pin Header	I²C signal SDA and SCL

14	X9	5-pin Header	Test signals SHTCB, RTCB, TC
15	J10 & J11	Diode selectors	For selecting an internal configuration map.
16	J12	Clock selector	For selecting the clock source. Default set to crystal input, if needed a generator can be applied instead with the jumper connected to the GEN-pin.
17	J13	Suspend mode simulator	With this jumper you can simulate the Suspend mode functionality.
18	J15	LED power control	For control of the LED's. The power consumption at nominal status can be reduced with this switch if needed.
19	J16	Jumper for LED power control	Switch for disabling or enabling the the status LED's.

3.3 Chip pinout

The following table describes the pins of the UDA1325PS / N10x SDIP42 version:

Name	Pin SDIP42	I/O	Pin Description
DA	1	I	data input (digital)
WS	2	I	word select input (digital)
BCK	3	I	bit clock input (digital)
GP2/DO	4	I/O	general purpose pin 2 or data output
GP3/WSO	5	I/O	General purpose pin 3 word select output
GP4/BCKO	6	I/O	General purpose pin 4 bit clock output
SHTCB	7	I	shift clock of the test control block (active high)
D-	8	I/O	Negative data line of the differential data bus
D+	9	I/O	Positive data line of the differential data bus
V_{DDI}	10	-	digital supply voltage for core
V_{SSI}	11	-	digital ground for core
V_{SSE}	12	-	digital ground for I/O pads
V_{DDE}	13	-	digital supply voltage for I/O pads
GP1/DI	14	I/O	General purpose pin 1 or data input
GP5/WSI	15	I/O	General purpose pin 5 or word select input
GP0/BCKI	16	I/O	General purpose pin 0 or bit clock input
SCL	17	I/O	serial clock line I²C-bus
SDA	18	I/O	serial data line I²C-bus

V_{SSX}	19	-	crystal oscillator ground (48MHz)
XTAL1b	20	I	crystal input (analog; 48MHz)
XTAL2b	21	O	crystal output (analog; 48MHz)
V_{DDX}	22	-	supply crystal oscillator (48MHz)
V_{DDO}	23	-	Supply voltage for OPAMP
V_{SSO}	24	-	OPAMP ground
VOUTL	25	O	Voltage output left channel
TC	26	I	test control input (active HIGH)
RTCB	27	I	asynchronious reset input of the test control block
VOUTR	28	O	voltage output right channel
V_{DDA1}	29	-	analog supply voltage 1
V_{SSA1}	30	-	analog ground 1
$V_{ref(DA)}$	31	O	reference voltage output DAC
$V_{ref(AD)}$	32	O	reference voltage output ADC
V_{DDA2}	33	-	analog supply voltage 2
VINL	34	I	input signal left channel PGA
V_{SSA2}	35	-	analog ground 2
VINR	36	I	input signal right channel PGA
VRN	37	I	negative reference input voltage ADC
VRP	38	I	positive reference input voltage ADC
V_{DDA3}	39	-	supply voltage for crystal oscillator and analog PLL
XTAL2a	40	O	crystal output (analog; ADC)
XTAL1a	41	I	crystal input (analog; ADC)
V_{SSA3}	42	-	crystal oscillator and analog PLL ground

3.4 Schematics

Please see chapter References for URL of the Philips internet-website for downloading the latest schematic of the demo-board.

3.5 List of Active Components

3.5.1 Integrated Circuits

Name	Description	Remark
IC1	USB-CODEC UDA1325PS SDIP42 package	
D1	74LV4066 bilateral switch	Not for HIGH-END audio.
D2	Socket for external EEPROM	(not stuffed)
D3	TDA3663T Voltage Regulator	3.3V, $I_{quiescent}$ = 17uA, unloaded
A2	PCF8574 I²C - I/O Expander	

3.5.2 Transistors & LED's

Name	Type	Description	Remark
V1		LED2, status GP2	
V2	BC848B	Selector transistor	SMD
V3		LED1, status GP1	
V4		Diode matrix, GP3	
V5		Diode matrix, GP4	
V6	BC848B	Startup circuitry transistor	SMD
V7	BC848B	Startup circuitry transistor	SMD
V8	BC858B	Startup circuitry transistor	SMD
V9	BC557B	Switch-logic for suspend mode functionality	
V10	BC557B	Switch-logic for suspend mode functionality	
V11	BC547B	Switch-logic for suspend mode functionality	
V12		LED3, Stabilisator power status	
V13		Status of I/O expander pin 5	
V14		Status of I/O expander pin 4	
V15		Status of I/O expander pin 6	
V16		Status of I/O expander pin 7	
V17		Status of I/O expander pin 9	
V18		Status of I/O expander pin 10	
V19		Status of I/O expander pin 11	
V20		Status of I/O expander pin 12	

3.6 List of Passive Components

The following table gives a list of all active and passive components used on the UDA1325PS 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	1k5	Startup Circuitry	
R2	68k	Startup Circuitry	
R3	68k	Startup Circuitry	
R4	100 Ω	Base resistor for V9 and V10	
R5	10k	Output resistor	
R6	10k	Pull-up resistor for I²C SCL line	
R7	10k	Pull-up resistor for I²C SDA line	
R8	10k	Output resistor	
R9	22 Ω	Serial resistor for input impedance and rise/fall time control	
R10	22 Ω	Serial resistor for input impedance and rise/fall time control	
R11	100k	Base resistor for V6	
R12	100k	Base resistor for V7	
R13	100k	Pull-up resistor for GP4	
R14	100k	Pull-up resistor for GP3	
R15	68k	Base resistor for V8	
R16	68k	Base resistor for V8	
R17	68k	Base resistor for V2	
R18	22k	Offset resistor LEFT IN	
R19	22k	Offset resistor RIGHT IN	
R20	22k	Collector resistor for V2 and pull-up for E0 and E3 input	
R21	100k	Base resistor for V6 and V7	
R22	22k	Offset resistor RIGHT IN	
R23	22k	Offset resistor LEFT IN	
R24	22k	Offset resistor RIGHT IN	
R25	22k	Offset resistor LEFT IN	
R26	22k	Pull-up resistor for GP0	
R27	820 Ω	Current limitation for V14	
R28	820 Ω	Current limitation for V13	

R29	820 Ω	Current limitation for V15	
R30	820 Ω	Current limitation for V16	
R31	820 Ω	Current limitation for V17	
R32	820 Ω	Current limitation for V18	
R33	820 Ω	Current limitation for V19	
R34	820 Ω	Current limitation for V20	
R35	22k	Base resistor V11	
R36	1k	Current limitation for LED V3	
R37	1k	Current limitation for LED V1	
R38	1k	½Vdd resistor network	
R39	1k	½Vdd resistor network	
R40	1k	Current limitation for LED V12	Power status

3.6.2 Capacitors

Name	Value	Description	Remark
C1*	100µF	DC Smoothing capacitor	16 Volt
C2*	100µF	DC Smoothing capacitor	16 Volt
C3*	100µF	DC Smoothing capacitor	16 Volt
C4	22pF	to adjust rise and fall time & impedance matching	63 Volt
C5	22pF	to adjust rise and fall time & impedance matching	63 Volt
C6	12pF	Filter capacitor for 3rd overtone crystal	63 Volt
C7	47µF	DC Smoothing capacitor	16 Volt
C8	47µF	DC Smoothing capacitor	16 Volt
C9	47µF	DC Smoothing capacitor	16 Volt
C10	47µF	Serie capacitor for DC filtering	16 Volt
C11	47µF	Serie capacitor for DC filtering	16 Volt
C12	47µF	Serie capacitor for DC filtering	16 Volt
C13	47µF	DC Smoothing capacitor	16 Volt
C14	47µF	DC Smoothing capacitor	16 Volt
C15	47µF	Serie capacitor for DC filtering	16 Volt
C16	47µF	Serie capacitor for DC filtering	16 Volt
C17	47µF	Serie capacitor for DC filtering	16 Volt
C18	47µF	Serie capacitor for DC filtering	16 Volt

C19	47µF	Serie capacitor for DC filtering	16 Volt
C20	12pF	Filter capacitor for 3rd overtone crystal	63 Volt
C21	18pF	Filter capacitor for 3rd overtone crystal	63 Volt
C22	18pF	Filter capacitor for 3rd overtone crystal	63 Volt
C23	100µF	DC Smoothing capacitor	16 Volt
C24	100nF	HF supply filtering capacitor	63 Volt
C25	100nF	HF supply filtering capacitor	63 Volt
C26	100nF	HF supply filtering capacitor	63 Volt
C27	100nF	HF supply filtering capacitor	63 Volt
C28	10nF	Capacitor for 3rd overtone filtering of 48MHz crystal	50 Volt
C29	100nF	HF supply filtering capacitor	63 Volt
C30	100nF	HF supply filtering capacitor	63 Volt
C31	100nF	HF supply filtering capacitor	63 Volt
C32	100nF	HF supply filtering capacitor	63 Volt
C33	100nF	HF filtering capacitor for VREFAD	63 Volt
C34	100nF	HF filtering capacitor for VREFDA	63 Volt
C35	100nF	HF supply filtering capacitor for EEPROM	63 Volt
C36	10nF	HF supply filtering capacitor for USB-power	50 Volt
C37	10nF	USB-cable decoupling capacitor	50 Volt
C38	100nF	HF filtering capacitor for USB-power if used	50 Volt
C39	100nF	HF supply filtering capacitor for voltage regulator	50 Volt
C40	100nF	HF supply filtering capacitor for selector IC	50 Volt
C41	100nF	HF supply filtering capacitor for ½Vdd resistor network	50 Volt
C42	100nF	HF supply filtering capacitor for I/O expander	50 Volt

3.6.3 Inductors

Name	Value	Description	Remark
L1*		4 inductors in one 8-pins package, EMI – countermeasures.	TDK ZJY-M4A
L2	1.5µH	Axial Choke / Oscillator inductor	
L3*		Ferrite bead for FCC rule compliance	BLM32A07
L4*		Ferrite bead for FCC rule compliance	BLM32A07
L5*		Ferrite bead for FCC rule compliance	BLM32A07

3.6.4 Crystal Oscillators

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

3.7 Removable components

Since this application board is intended to give as much as possible functions and modes on 1 board, it is certainly possible that you will be able to remove components which are of no interest for your own application / product. The list here below gives an overview of which components can be removed under the given condition:

Name	Value	Condition for removal
D2	*EEPROM*	If you are one of the legal owners of the internal maps, you do not need to program an external configuration for your application. In that case, you may remove the EEPROM.
V4, V5, J10, J11.	Diode Diode jumper jumper	You should remove these items, if you are not a legal & intended owner of on of the internal configuration maps ! In that case, you must program your own configuration within an I2C EEPROM.
S1	Reset Switch	The reset switch is not for customers endproduct. It is meant for debugging purposes only and should be removed.
V9, V10, V11, R4, R35, J13	Transistor Transistor Transistor Resistor Resistor Jumper	Remove these items, because these are meant for debugging and testing the suspend-mode capability. Since the SDIP42 version has no CLK output pin (which the UDA1325H QFP64 does), there is no trigger available to trigger the external circuitry when the suspend-mode is activated. You can connect all **+VASW** to +VA lines and all **+VDSW** to the +VD lines. And finally connect the **LED_SW** lines to ground (0V).
C21, C22, J12	Capacitor Capacitor Jumper	If you are not planning to use topology 4 (analog IN, with extra OSCAD crsytal at XTAL2A and XTAL1A), you can remove these items and the lines may be floating.
A2	I/O expander	If you are not intended to use any I/O expander functionality, you can remove all items attached to this IC. This means, all LED's V13 .. V20, all series resistors R27 .. R34, switch J15, jumpers J3 / J5 / J6 / J14 and finally capacitor C42.
X6	I²S input connector	If you don't plan to use topology 3 (Digital Input), you may remove this connector.
X1, X2	I/O Connector	If you don't plan to use the I²S lines, you may remove these two connectors.
L3, L4, L5	Coils	Should be inserted for EMC compliance. It is possible that your application even demands more counter measures to reduce EMI and influences from outside peripherals because of your PCB design method and/or environment. In such case, we recommend to use extra coils close to the individual supply pins of the USB-CODEC as presented on the application board/schematic of the UDA1335H. See chapter References how to get the board-schematic of the UDA1335H.
L1	Coils	Please read application note 2, chapter 2.2.2.

4. References

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

[1] UDA1325PS specification

[2] USB-Specification version 1.0 and version 1.1

[3] USB Device Class Definition for Audio Devices

[4] USB Human interface Device Class specification

[5] USB Human Interface Device Class Usage Table

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

For downloading the latest documents, software tools and more, please regularly check our USB-AUDIO website:

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

5. In case of Problems

Should you still experience problems in operating the Philips Semiconductors UDA1325PS demonstration board, we recommend that you get into contact with your local Philips Semiconductors Field Applications Engineer. The local Philips FAE will be able to help you to get your system up and running.