TECHNICAL NOTE

USB-AUDIO CODEC UDA1325PS

EXTENDED
EVALUATION BOARD
AND
APPLICATION DOCUMENTATION
Version 1.5

DML98022

TECHNICAL NOTE

USB-AUDIO CODEC
UDA1325PS

EXTENDED
EVALUATION BOARD
AND
APPLICATION DOCUMENTATION
VERSION 1.5

Author:

Sedat 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 is an extension to accompany technical note DML98021 that describes the installation and application of the Philips Semiconductors USB-Digital to Analog Converter and Analog to Digital Convertor solution UDA1325PS for the Universal Serial Bus (USB). Whereas document DML98021 focused on basic information about the Universal Serial Bus itself and how to install and use the UDA1325PS evaluation board, this document gives lots of IC related details for those customers that want to fully understand and customize the USB-CODEC UDA1325PS for use in their own specific application.
This document does not include the basic information given in DML98021. For information on how to install the Philips Semiconductors USB-CODEC on your system please refer to the mentioned document. All information within this document refer to both the N103 and N106 silicon/firmware releases.

Notice: Windows™ 95, Windows ™ 98 and Windows™ NT are registered trademarks of Microsoft ® Corporation. Memphis is the code name used by Microsoft® for the Windows™ 98 development project.

USB compliant conform USB specifications

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 explains details about the specific and extended features of the Philips Semiconductors USB-CODEC UDA1325PS / N106 release. It is intended for customers that want to fully understand and customize the USB-CODEC for use in their own specific application.

This application note will not explain the Universal Serial Bus as such or the installation of the UDA1325PS evaluation board. You can find this information in the principal application note *DML98021* of Philips Semiconductors and in the documentation that is noted/referred in Chapter 8 "References".

The application of the USB-CODEC UDA1325PS requires a USB enabled PC running Windows 98.

NOTES:

The UDA1325PS operates under Windows 98. It can not be used with other (not USB streaming supporting) Windows operating systems. These operating systems do not provide the drivers and system support necessary to operate devices attached to the Universal Serial Bus.

The Operating System of the *iMac*™ platform is also introducing support for USB-AUDIO CODECs and support for this platform is under development by our facilities too.

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

We strongly advise not to use earlier builds. You may try earlier builds but it is unrealistic for us to guarantee that this will work without problems.

In any case, constant and quick efforts are being made to guarantee that the USB-CODEC will run without problems with any official Windows 98 release. The firmware used with the USB-CODEC is based upon an internal µC80C51 core. It will be flexible/configurable enough to cover as many USB audio features as possible. The current evaluation systems allows the User to:

- select between 4 different integrated ROM configurations for their own USB-CODEC application
- plug in an additional I²C-EEPROM that contains a custom configuration. The customer is capable of editing it's own configuration map by means of the software tool *UniCoDes.*

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 (& PID) as will be explained in the following chapters. 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.

2. System Startup

At Startup/Power-on, the USB-CODEC completes a boot cycle during which all its internal registers are loaded and all pins and settings are configured. There are a total of 5 different configurations that a user can choose from at startup time. Four of these configurations are built into the UDA1325PS firmware. They can be used without any change **IF YOU ARE THE LEGAL OWNER** of the internal map. You are not allowed to use any of the internal maps, if you are not the legal owner of the map. In that case, you must use an external configuration ! If only the name descriptor needs to be changed this can be done by changing some system files. This has the advantage that there is no need for an I²C-EEPROM (see section 4.1 of this application note). If you don't want to do that or you want another configuration, you can use the fifth configuration. At startup this fifth configuration can be loaded using the common I²C Bus protocol from an external 1024 or -EEPROM (if present). The USB-CODEC has a number of General Purpose Input / Output pins that can serve various different purposes. These will be described in detail in the following chapters. For now it is only necessary to notice that two of the GPIO pins (GP3 and GP4) are used to select one of the four internal configurations (in case there is no dedicated I²C EEPROM present).

The basic boot sequence of the USB-CODEC is shown in Figure 2-1:

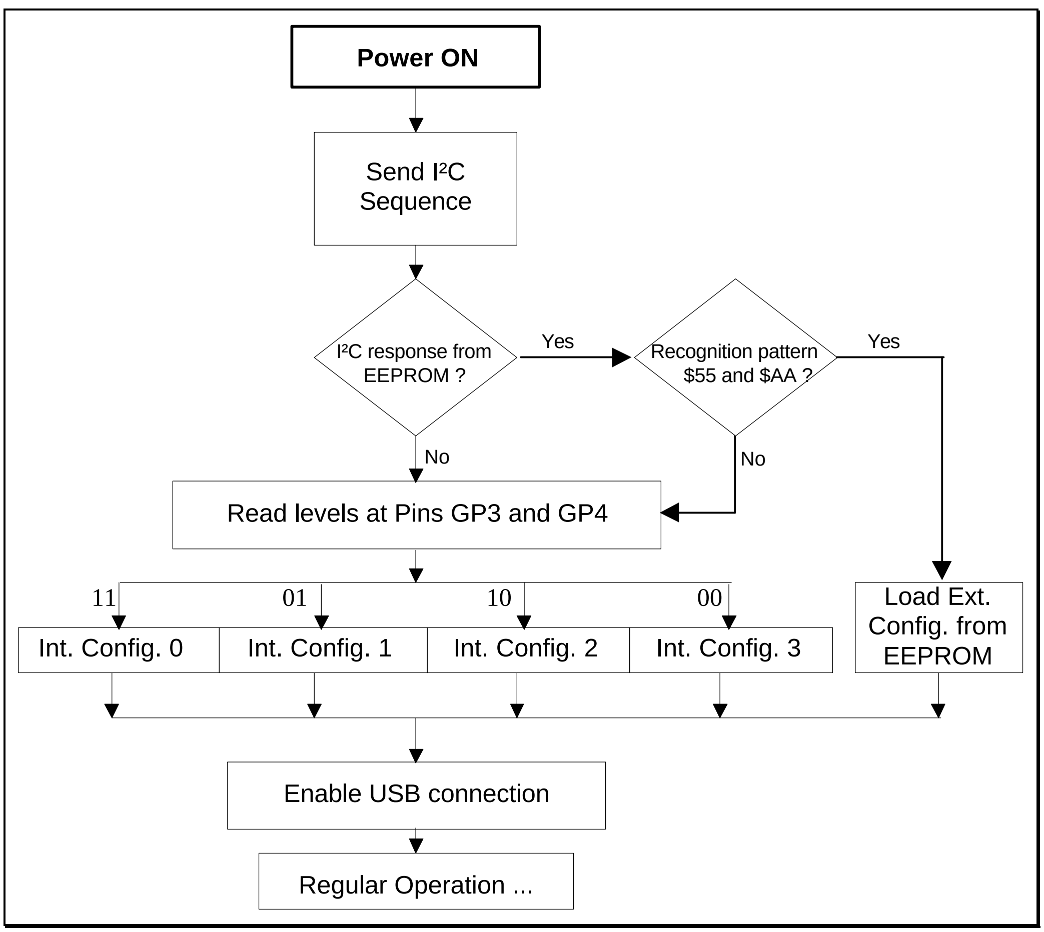


Figure 2-1 Boot Sequence Flow Chart

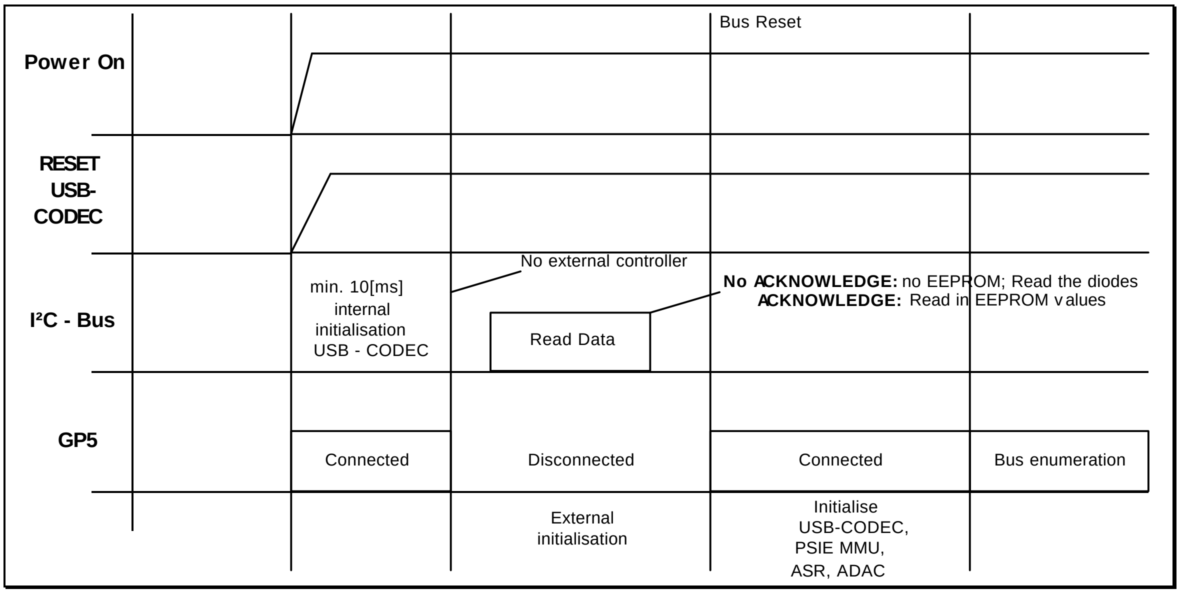


Figure 2-2 Basic boot-sequence of the USB - CODEC

The USB-CODEC can only be used as a self-powered device as defined in the USB Specification. The transitions from/to the states 'power- -down' and 'suspended' will be handled at the hardware level. After the hardware power-up phase, the firmware initializes the USB-CODEC components. Initialization is done in a specific order:

To load data from an external I²C-EEPROM, at initialization time, the internal µController will send a read request to I²C slave address A0h. It will attempt to read the device's first two byte locations and compare these bytes 5h AAh". If a match occurs, it assumes that this I²C device is an EEPROM and that it is dedicated to the USB-CODEC. It will then read the stored configuration data. The exact structure of this EEPROM configuration data, is given in the datasheet. For more details about this configuration map, please read chapter 3.

The circuitry connected to the General Purpose Input/Output Pins serves a number of different purposes and will be described in more detail later in this chapter. For now (or for the simplest case) it will be assumed that they are simple pull-up/ pull-down resistors that determine the level. A detailed explanation will follow in chapter 2.

2.1 General Purpose I/O Pins and I/O Expander Pins

The USB-CODEC has a total of 6 General Purpose Input / Output (GPIO) pins which can be configured and used for different functions such as I^2S interfacing to an external DSP, Volume Up and Down, Tone Control or Mute, Selecting terminal outputs (in this case MIC or LINE).

The GPIO pins of the USB-CODEC µController can be used for:

- Interface a maximum of 5 user definable resources.
- Support I^2S interface lines.
- Selection of input terminals.
- Support a ‘USB-CODEC configuration’ map selection.
- Disconnect/Connect for bus-powered operation.

There are basically three different configurations for the GPIO pins possible that can be described as:

- No digital I/O communication.
- 4-pin digital I/O communication.
- 6-pin digital I/O communication.

Regarding the recording-path there are 4 different topologies supported on the current *application board* and *UniCoDes* provided by Philips Semiconductors:

- Analog input, No selector unit
- Analog input, With selector unit
- Digital input (I^2S MUX input), No selector unit
- Analog input (sample frequency pre-defined by an extra ADC-crystal), No selector unit

2.1.1 No Digital I/O configurations

This port configuration can be chosen via the configuration map at start-up of the UDA1325PS. The following table gives an overview of the different functions assigned to the respective pins for the “No Digital I/O” communication mode:

Table 2-1 No Digital I/O communication

Pin	GP I/O
GP5	Connect / Disconnect (2)
GP4	HID input 2 (1) (5)
GP3	HID input 1 (1) (5)
GP2	Selector output
GP1	Mute or standby output (3) (4)
GP0	Interrupt input

Pin	I/O Expander (6)
P0	HID input 3 (1) (5)
P1	HID input 4 (1) (5)
P2	HID input 5 (1) (5)
P3	HID input 6 (1) (5)
P4	Output pin 1
P5	Output pin 2
P6	Output pin 3
P7	Mute or standby output (3) (4)

Notes:
(1) All input pins must be defined at a certain voltage level by means of pull-up resistors or pull-down resistors.
(2) Connect / Disconnect: Holds the USB disconnected as long as the initialization is not finished.
(3) Standby is toggled if the mute is active for 2 minutes programmable time and no ISO data on USB bus.
(4) Mute is toggled if the isochronous (ISO) data flow is interrupted.
(5) For selection of HID application see HID report descriptors in *UniCoDes*.
(6) For the I²C expander, the PCF8574P Remote 8 bit I/O expander for I²C-bus can be used.

2.1.1.1 Functions within the “No Digital I/O configurations” - mode

<u>* Disconnect/Connect (GP5) functionality:</u>
Pin GP5 is used to avoid malfunction during initialization phase. It is **very important** to use this functionality within your application. While initializing, the Universal Serial Bus remains disconnected and the configuration selection diodes or the contents of the I²C EEPROM are read in. As soon as the µController is ready for regular operation, the Universal Serial Bus gets connected and the configuration selection diodes are switched off. In the time-scale we will see the following happening:

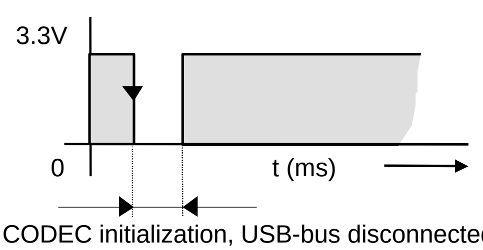


Figure 2-3 Connect/ Disconnect phase after power-up.

<u>IMPORTANT:</u>
Recently, we have introduced a new Disconnect/Connect circuit to improve the enumeration behavior of the USB-CODEC’s. Please contact your local FAE or check the USB-AUDIO website to download the report **REP0021.PDF** which explains why and how to implement the newly introduced circuit.

* HID inputs 1& 2:

As you will see, both HID inputs are implemented on the application board. The respective switches are defined as S2 and S3. Depending on the definition of each switch, theoretically you should be able to use these switches for different purposes. These functions are defined in the *Universal Serial Bus HID Usage Tables* document.
In version 1.0 of this document -page 60-, you will see many functions defined that are supported by Win98.
In this case, only Audio related functions are interesting for us. Therefore, page 62 has some interesting functions like Volume control, mute, bass etc. that you could define or test.
UniCoDes enables you to set different functions with the command / button *Descriptors*. This will pop-up a new window in which you must select the button *HID Report Descriptors*. Now you have three TAB fields to choose from. Select the second tabfield titled as *HID input functionality*. Default, the values are set to Volume controls for HID input 1 & 2. See figure below.

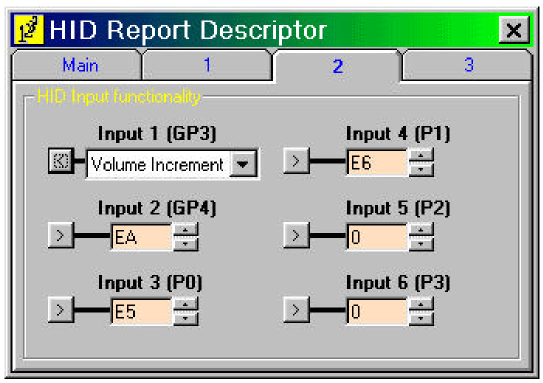


Fig. 2-4 HID functions definitions

* Selector outputs:

In case you are using topology 2, you have the ability to control the selector with GP2. This pin is defined at two places:

- Address 0Bh defines the logical level of GP2.
- Address 13h defines the preferred Terminal to use.

The table below gives the possible combinations and results of each configuration:

Selector Preferred state	**Selector output level**	**Results**
Terminal 1	Normal	PGA settings for Terminal 1 takes effect with GP2 = 0V
Terminal 1	Inverted	PGA settings for Terminal 1 takes effect with GP2 = Vdd
Terminal 2	Normal	PGA settings for Terminal 2 takes effect with GP2 = Vdd
Terminal 2	Inverted	PGA settings for Terminal 2 takes effect with GP2 = 0V

Table 2-2 Possible combinations for definition of GP2

* Interrupt input:

The firmware uses two different methods for detecting changes on the expander inputs. The first one applies serial polling in which the inputs are scanned every 20ms. This frequency is defined within the firmware and is not settable. The other method is by implementing hardware interrupt. Everytime that the expander detects changes at the input lines, it will produce an interrupt signal at the respective pin (13) of the expander. The firmware will be triggered by this via the GP0 interrupt input and will update all respective registers.

* Mute & Standby outputs:

GP1 represents the Mute or Standby output depending on the definition in address 0Bh. In case you think it is better or need to define the respective output on the expander, select address 0Bh and select the respective option-entry. The timing between the different states are settable in the addresses 11h and 12h. If T1 = 'Time between mute and play' and T2 = 'Time between mute and standby' then we can visualize the states as following:

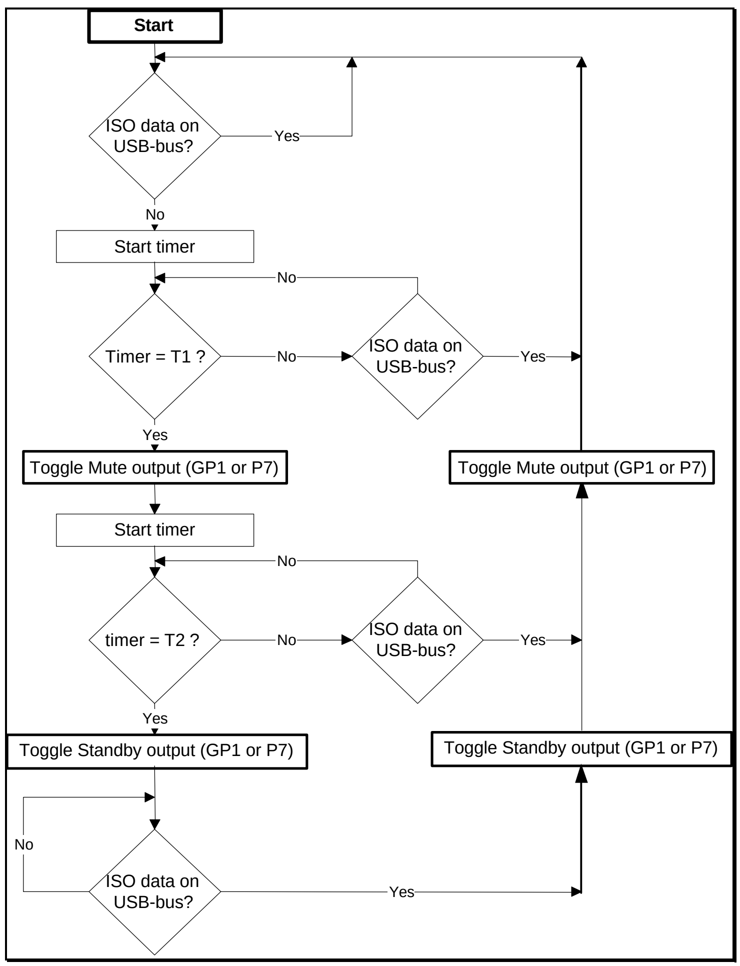


Figure 2-5 Flow-chart of the Standby and Mute functions

* If I/O-expander used, HID inputs 3,4,5 & 6:

The functions of these inputs can be defined in *UniCoDes* too. For this you will have to select tabfield 2 of the *HID Report Descriptor* menu (see fig 2-4) and set the corresponding value of the function you had in mind.

* If I/O-expander used, output pins 1, 2 & 3:

The three output pins are also programmable in *UniCoDes*. For this you have the following addresses available:

- Address 0Bh for defining the polarity of the output pins (see fig 2-6, LEFT).
- Address 0Ch, 0Dh and 0Eh for defining the function for each pin. As you can see the functions are restricted to 2 possible functions. These are Mute and the DBB functions (see fig 2-6, RIGHT)

The result is that when one of these two functions is toggled, the “state machine” or software routine in the firmware will toggle the corresponding pins on the expander.

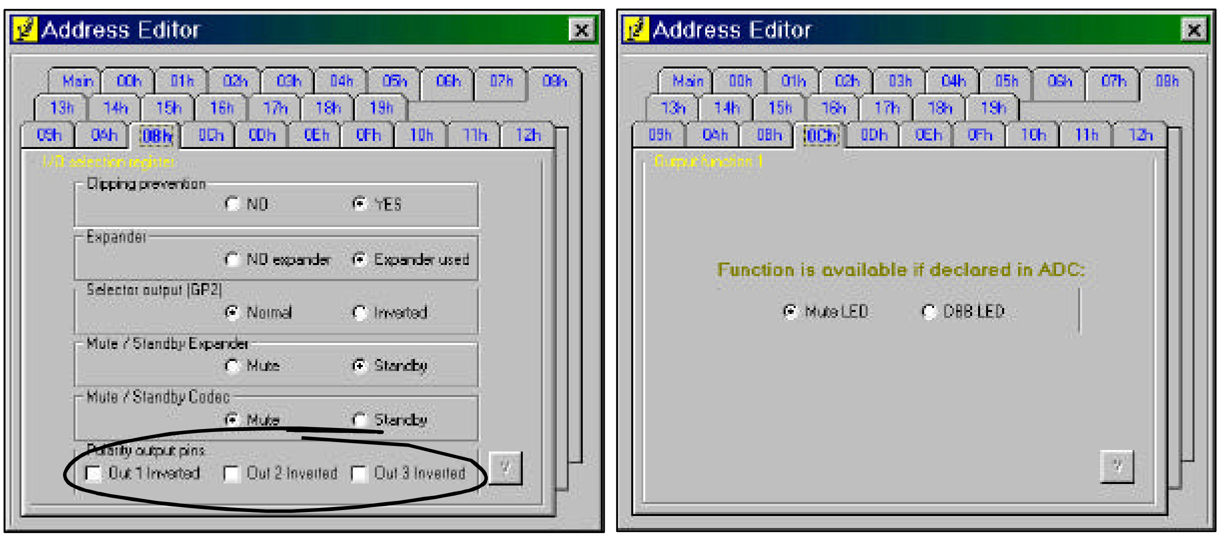


Figure 2-6 Output definition related addresses

* If I/O-expander used, Mute & Standby outputs:

If prefferred you can define either the Standby or Mute output on the expander. This can and may vary on each application.

2.1.2 Four-pins I/O configurations

The pin-assigments in the 4-pins I/O configuration are listed in the table below:

Table 2-3 4-pins Digital I/O communication

Pin	GP I/O
GP5	Connect / Disconnect (2)
GP4	BCK output
GP3	WS output
GP2	DATA output
GP1	DATA input
GP0	Interrupt input

Pin	I/O Expander
P0	HID input 3 (5)
P1	HID input 2 (5)
P2	HID input 5 (5)
P3	HID input 6 (5)
P4	Output pin 1
P5	Output pin 2
P6	Selector output
P7	Mute or standby output (3) (4)

Notes:
(1) All input pins must be defined at a certain voltage level by means of pull-up resistors or pull-down resistors.
(2) Connect / Disconnect: Holds the USB disconnected as long as the initialization is not finished.
(3) Standby is Switched on (output becomes LOW) if the mute is active for 2 minutes programmable time.
(4) Mute is switched on (output becomes LOW) if the isochronous data flow is interrupted.
(5) For selection of HID application see HID report descriptors in *UniCoDes*.
(6) For the I²C expander, the PCF8574P Remote 8 bit I/O expander for I²C-bus can be used.

2.1.2.1 Functions within the “Four-pins I/O configurations” – mode

The functions of GP5 and GP0 on the USB-CODEC have not changed in comparison with the ’No Digital I/O’ mode. On the expander, only P6 has changed functionality as you can see in table 2-1.
Therefore, we will focus on the GP-pins that have changed functionality and newly introduced. These are GP4, GP3, GP2 and GP1.

<u>* I²S output pins (GP4, GP3 and GP2) and I²S input pin (GP1):</u>

These lines can be used for digital processing via a compatible DSP. Consequently, the DSP can perform the neseccary calculations and feedback the data into the I²S input line (GP1).
To get an idea, see figure 2-6.

<u>* HID input pins (P0 … P3), Output pins (P4 and P5), Selector output (P6) and Mute/Standby output (P7):</u>

The functions are still the same as described in the previous chapters for the ‘No Digital I/O mode’.

2.1.3 Six-pins I/O configurations

The pin-assigments in the 6-pins I/O configuration are listed in the table below:

Table 2-4 6-pins Digital I/O communication

Pin	GP I/O
GP5	WS input
GP4	BCK output
GP3	WS output
GP2	DATA output
GP1	DATA input
GP0	BCK input

Pin	I/O Expander
P0	Connect / Disconnect (2)
P1	HID input 4 (5)
P2	HID input 5 (5)
P3	HID input 6 (5)
P4	Output pin 1
P5	Output pin 2
P6	Selector output
P7	Mute or standby output (3) (4)

Notes:
(1) All input pins must be defined at a certain voltage level by means of pull-up resistors or pull-down resistors.
(2) Connect / Disconnect: Holds the USB disconnected as long as the initialization is not finished.
(3) Standby is Switched on (output becomes LOW) if the mute is active for 2 minutes programmable time.
(4) Mute is switched on (output becomes LOW) if the isochronous data flow is interrupted.
(5) For selection of HID application see HID report descriptors in *UniCoDes*.
(6) For the I²C expander, the PCF8574P Remote 8 bit I/O expander for I²C-bus can be used.

As we can see in the table above, we have shifted the Disconnect/Connect function from GP5 to the expander pin P0. Please read chapter 7 how to implement 6-pins Digital I/O.The main difference with the previous I/O configurations, is the fact that the I²S output is totally seperated from the I²S input. This mode opens the possibility to implement a DSP unit between these two general purpose pins. This way you can manipulate the I²S data and send it back to the ADAC of the UDA1325PS like illustrated in figure 2-7:

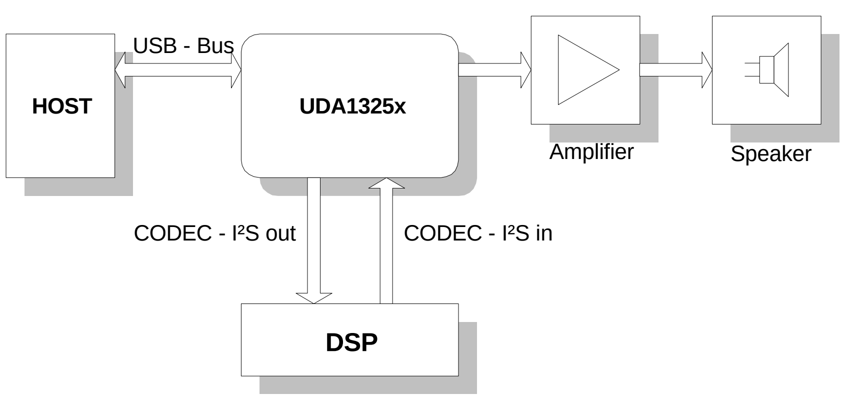


Figure 2-7 Example for purpose of the 4-pins & 6-pins I/O configuration.

2.1.3.1 Functions within the "Six-pins I/O configurations" – mode

* I²S related GP pins (GP0..GP5):

Instead of only the I²S DATA input pin, we have also the other two I²S pins available in the form of GP5 (WS input) and GP0 (BCK input).

* All expander pins :

The functions are still the same as described in chapters 2.1.1 and 2.1.1.1 regarding 'No Digital I/O mode'.

2.1.4 Diode Selection

The schematic in fig. 2-8 gives the circuit for selecting a configuration via diodes in combination with two HID-input switches. If you want to implement the Connect / Disconnect functionality into your application (which we strongly recommend), you can use the schematic presented in the latest application board schematic or see the respective datasheet (UDA1325.PDF) on page 44 to start with.

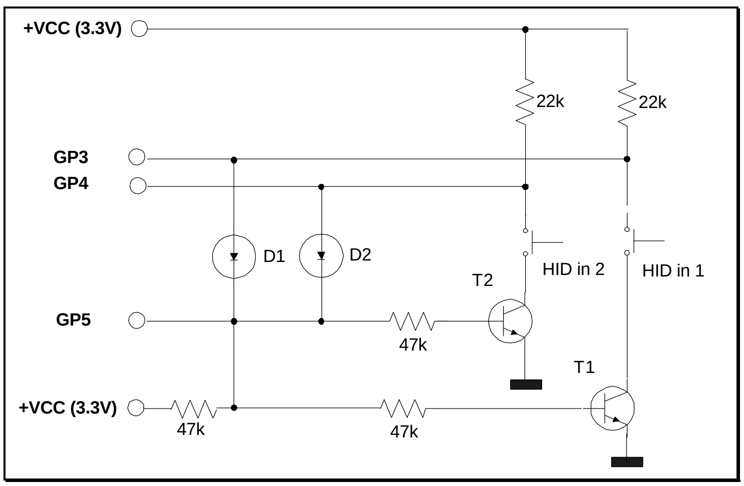


Figure 2-8 Circuit for selecting a configuration map in combination with 2 HID inputs.

At initialization phase (GP5 low), both transistors are not conducting and it is not possible to read in a wrong diode value by (accidently or by purpose) pressing one or two of the external function (HID) buttons. After initialization the transistors are conducting. Including an optional additional diode requires another additional transistor:

Configuration map Number	Diode #	Comment	level during initialization	
			GP3	GP4
0	--	No diodes or transistors needed	high	high
1	D1	Transistor T1 needed	low	high
2	D2	Transistor T2 needed	high	low
3	D1 and D2	Both transistors needed	low	low

Table 2-5 Diode Selections

Note:

If no HID is used in the specific option or a wrong selection doesn't matter, the transistors are not needed.

In the general USB concept the 'USB-Host' needs to be notified of all activity in it's USB topology. This includes the use of the GPIO pins. For this purpose the USB HID specifications were defined. The HID Entities describe to the 'Host' the type of inputs/outputs that can be controlled.

There are two HID concepts /phases defined in the HID entity description. Phase one defines that 'all' control parameter changes need to be sent as requests to the Host first. Upon reception it will take action (phase two) and send a control change request command when e.g. a button was pushed.

Accordingly the action flow after pressing a button is:

1. the Device µController reports to the Host that a push button has been pressed and what function it represents,
2. the Host will initiate the correct request via the correct Class (Audio Device Class or HID), according to the pressed push button.

The Device µController will translate this request into a physical action by e.g. controlling the Volume level.

If you have decided not to use the HID inputs, then the schematic can be reduced as illustrated in fig. 2-9.

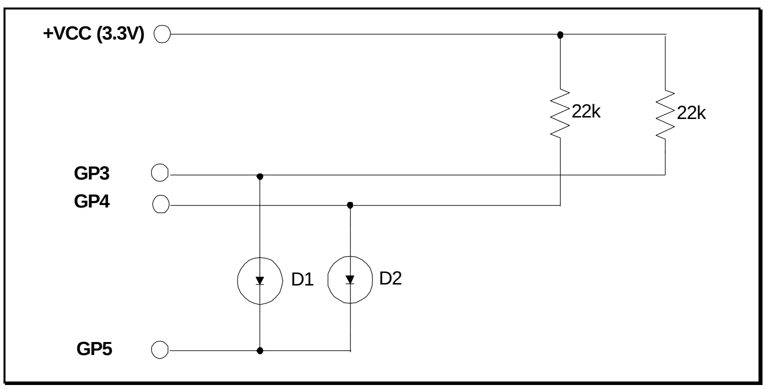


Figure 2-9 Diode Selections for selecting configuration with NO HID inputs switches.

In such case you will also have to remove all HID related data of your configuration map. This can be realized with just a few clicks of your mouse. Go to address 0Fh of the editor and select the respective button(s) :

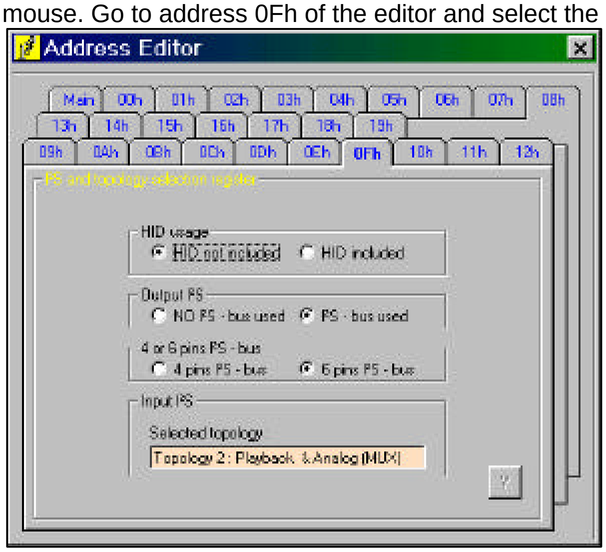


Figure 2-10 Neseccary adjustments for "NO-HID option" within the configuration map.

2.2 Additional External Circuitry

2.2.1 Oscillator Circuit

The USB-CODEC runs at 48MHz. An external signal can be applied to the XTAL input (pin 25). If the USB-CODEC is used with a crystal, the circuit described in figure 2-12 should be used.

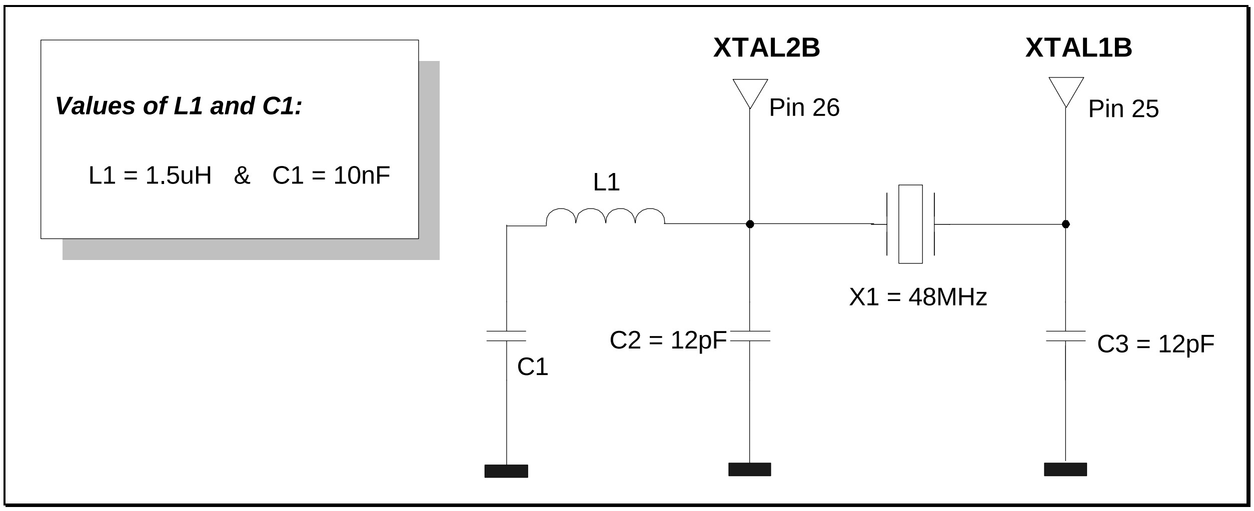


Figure 2-11 Crystal circuitry with 48MHz filtering components

The crystal specifications (summarized):

* Frequency stability in a temperature range of :
  - 55 Deg. Celcius to 105 Deg. Celcius => 50ppm typ.
  - 10 Deg. Celcius to 60 Deg. Celcius => 10ppm typ.
  - Resonance impedance = 40..45 Ohm
  - Static capacitance (between electrodes) = 5pF typ.
  - Load capacitance = 8pF typ.

* Third overtone crystal.

It is important that the capacitors C2 and C3 have the value as mentioned in figure 2-11. These values are beside defining crystal frequency output also important for the UDA1325PS internal hardware. There are no special requirements regarding to the deviation of these capacitors and therefore standard cap's can be used (maximum of 25% spread).

There are no special requirements for the inductor.

2.2.2 Coils at the input of the USB-bus

To meet the EMC compatibility rules,standardized by the ISO orginization, it is in some cases vital to use the coils as shown in the schematic at the input of the USB-bus. Though, removing the coils should not affect the functionality of the UDA1325PS in any way. It also depends on your PCB design if such countermeasures does have any benefit or not. If you plan to use this component it is important that you use a 4-in one package coil and not 4 physically separated coils. This is to meet the specifications given in the datasheet.
The isolators in the 4-in one packaged coil are designed to minimize the cross-over signals between the coils which is hard to accomplish if you use 4 separate coils (depending on the placement of these coils).
Furthermore, we discovered that the TDK ZJYS – 4 is causing enumeration problems with USB-cables longer then approx. 2 meters. It seems that there is significant difference with the one which should be inserted :
the ***TDK ZJY – M4A***. Application boards of version 1.4 which are send until 1st of March '99, have the ***TDK ZJYS – 4*** which should be replaced by either shortcircuits or the ***TDK ZJY – M4A***.

If you are interested in optimum protection of the USB-datalines against ESD on your application and PC peripherals which are also attached to the Universal Serial Bus, we recommend you to use ESD protection diodes in your application. There are several manufacturers of Surge Protection Diode Arrays which can be used for this purpose. Some of them also provide a zener diode to protect the Vbus power supply of USB. All these counter measures depend on which market (industry or private-sector) your product / application is focused on and how sensitive your application is / should be towards ESD.

2.2.3 Coils at the Vdd lines of the UDA325PS

It depends on the PCB design of your application and market whether you need / want adequate countermeasures to reduce EMI (and of course your budget). Fig. 2-12 shows how you can decrease EMI.

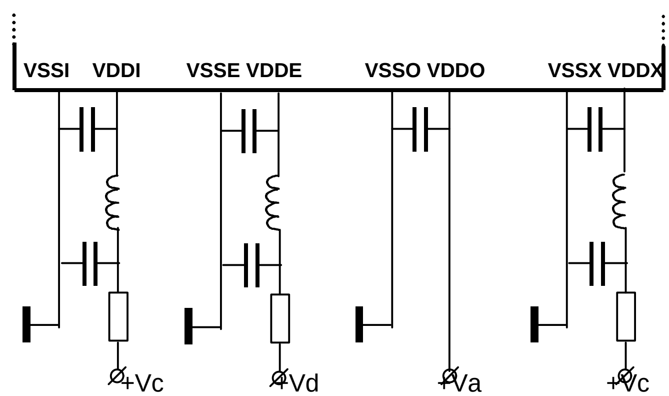


Figure 2-12 EMI-countermeasure related compoments directly at the individual supply lines of the UDA1325PS

In fig 2-12, all SMD resistors should be dimensioned as 1 ohm, SMD ceramic cap's as 100nF and SMD EMI CHIP ferrite beads as BLM32A07. In all attempts to reduce the influence of EMC, it is important to follow the generally applicable rules for designing PCB's. One of the most important of these rules is to keep all copper transmission paths as short as possible. This to reduce the antenna-effect of these lines. Please read your EMC related documents for more information. These extra decoupling and filtering components in combination with your specific application will help to get an EMC compatible design.

2.2.4 I/O Expander test-board

Depending on the communication method you have chosen (*No Digital I/O, 4-pins or 6-pins communication*), you will have to decide whether you need an expander or not. The table given in the datasheet of the USB-CODEC on page 28 of the *UDA1335.PDF **(**!)* datasheet, you can see the summary of all pin configurations implemented. If you should decide to add an expander, we recommend you to begin with a test board which can be very helpfull for you during the development of your own application. The schematic shown here below, is the schematic which provides you all neseccary controls and visual status LED's.

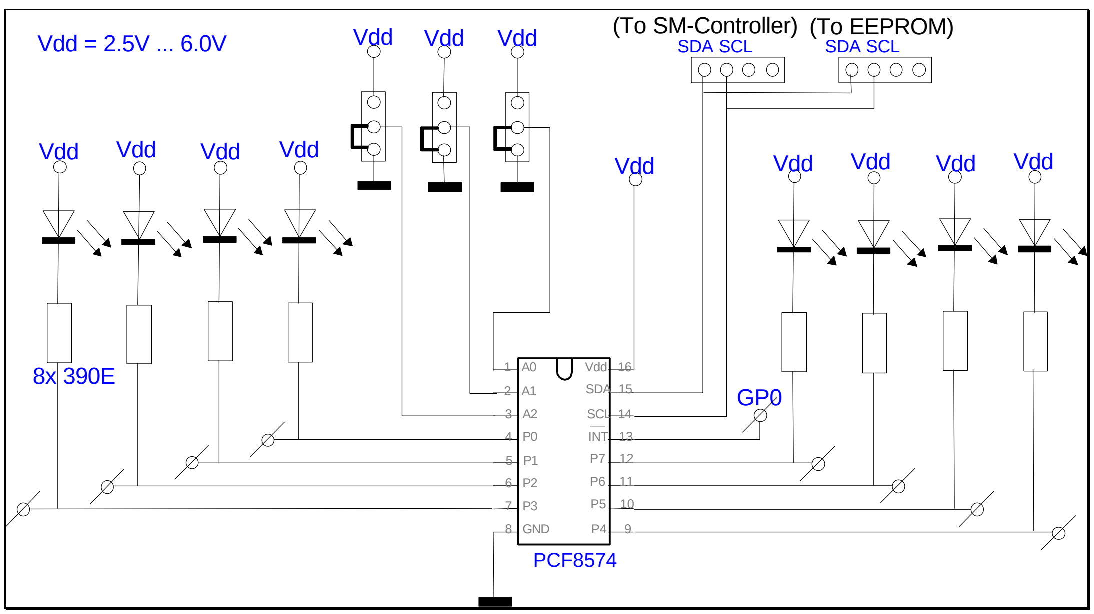


Figure 2-13 Expander circuitry with LED indicators and address selection

2.2.5 Bilateral Selector / Switch (HEF 4066 or 74LV4066)

We have provided a relative standard selector unit by means of a 4-in-1 packaged bilateral analog switch CMOS IC, the 74LV4066 (USB-CODEC application board v1.4 and higher). If you look at the datasheet of the 74LV4066 in any related CMOS databook, you will see that the S/N - ratio is relative low in comparison with the performance of the USB-CODEC specified.

Since the application PCB provided by us is for demonstration purposes only, we decided to use this relative low cost alternative. If your application is meant for the high-end market and you also plan to use the bilateral analog switches, we recommend you to make your choice depending on parameters like relative high S/N ratio, low series impedance and low crosstalk behaviour. But it is possible that power dissipation is more important for your application and your selection is based on this and other parameters.

To save you time and money, we have summarized some possible types of high-end bilateral switch types in table 2-6 here below.

Type	Manufacturer	Internet homepage
74LV4066D	Philips Semiconductors	www.philips.com
TL06X family	Texas Instruments	www.ti.com
M74HC4316	SGS-Thomson	www.st.com
ADG201A	Analog Devices	www.analog.com

Table 2-6 Alternative bilateral switches.

We don't know what your application specifications are and therefore it is up to you to decide and test the best possible solution. The selection is made pure by datasheet and therefore 100% guarantees for correct functionality with the USB-CODEC can NOT be given. It is obvious that you will have to test and determine what is possible or not.

See also chapter FAQ for more information regarding the selector unit.

2.2.6 Single Master Controller board

UniCoDes has the ability to program a configuration via the parallel port of your PC. In order to do this it needs a *Single Master Controller*. If you have no other means / tools to program your I²C-EEPROM, this method can be handy. The schematic for this controller is shown here below.

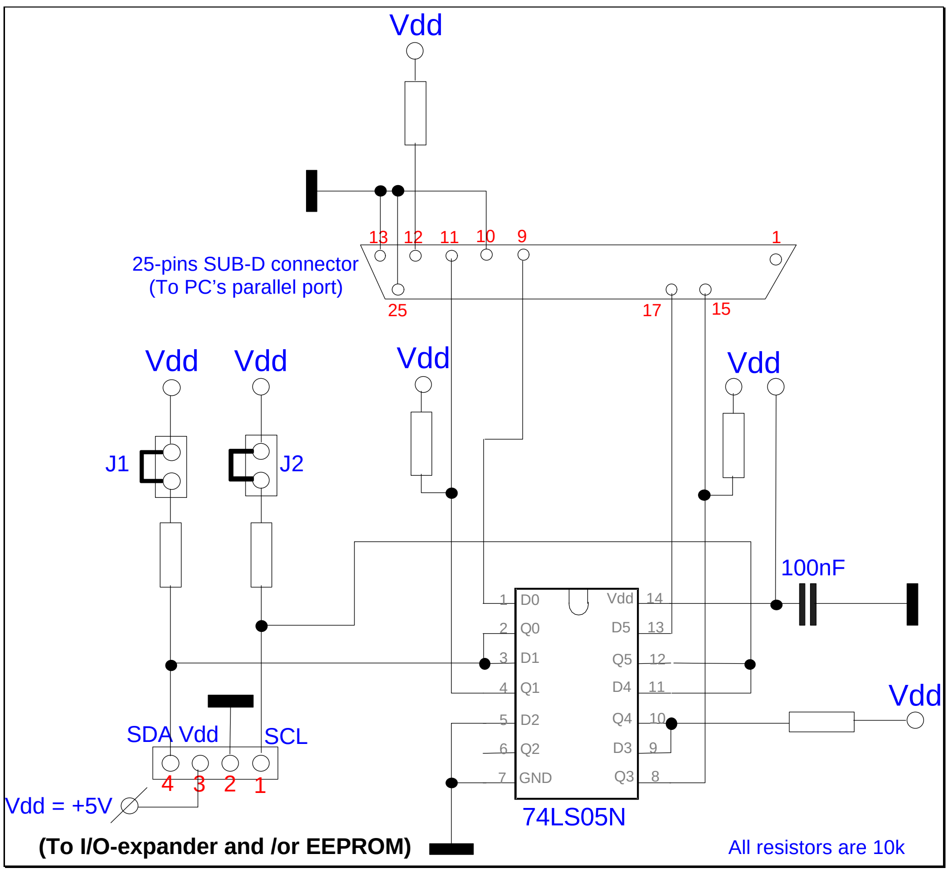


Figure 2-14 Single Master Controller circuitry

The 25-pins SUB D-connector has to be connected to your PC's parallel port. The jumpers J1 and J2 are just in case the I²C lines are already fitted with pull-ups on other boards. You can remove and disable the resitors if needed by simply removing the respective jumpers.

2.2.7 Suspend Mode Implementations

There is an important issue regarding the selection of which power source you want to use for your final application / product. 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-“low”-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 all power consuming items / components on the board. This enables us to reach approx. 560uA @ 3.30V (Vdd) at suspend-mode with the entire board. So we do not comply to the spec with the N103 & N106 silicon. An additional problem is the fact that the UDA1325PS / N103 & 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 and/or impedance) connected to the crystal is critical for the oscillating characteristics. But, since the UDA1325**H** / N103 & N106 has a digital CLK output pin (pin 27), we can use this output which can serve as a trigger. Bus-powered UDA1325x applications are under development and we plan to add many additional features on firmware level as well as full compliancy regarding suspend-mode implementation ! This new silicon & firmware release of the USB-AUDIO CODEC is named as the N104 release.

So, the suspend-mode circuitry as proposed on the current application board can be removed. This means that you can remove V9, V10, V11, R4, R35, J13 and J16. All supply lines labled as *VASW* and *VDSW* should be named as respectively *VA* and *VD* and connected directly to these power supply lines. Finally you should connect the LED_SW lines to ground.

2.2.8 Default Jumper Settings

The application board should be delivered to you with the following jumper / switch settings:

Jumper	Status	Function
J1	**Inserted**	To select one of the input terminals (MIC-IN or LINE-IN)
J2	**Inserted**	To enable *Connect/ Disconnect functionality* and to enable *internal map selection*.
J3	**Pin 1- 2**	For selecting I²C device address.
J4	**Pin 1- 2**	To set the *Write Control mode*. See datasheets of 24C08 and/or 24C16.
J5	**Pin 1- 2**	See J3.
J6	**Inserted**	To read the *Interrupt output* status pin of the I/O expander (if inserted).
J7	**Pin 1- 2**	Jumper should select the +3.3V stabilisator / regulator instead of +5V USB-bus (depends on your application).
J8	**Inserted**	To select the power supply of the Universal Serial Bus ($V_{bus\,=}$ 5V / 0.5A).
J9	**Inserted**	To enable the analog units of the UDA1325PS.
J10	**Removed**	To select one of the 4 internal maps. See also table 7-2 in chapter 7.
J11	**Removed**	To select one of the 4 internal maps. See also table 7-2 in chapter 7.
J12	**Pin 2- 3**	In this modus, the external OSCAD crystal is selected instead of the function generator (topology 4).
J13	**Pin 2- 3**	To enable the external peripherals (only needed for suspend-mode tests).
J14	**Pin 1- 2**	See J3.
J15	**Pin 2- 3**	To enable the I/O expander LED’s.
J16	**Inserted**	To enable the status LED’s (LED1, LED2 and LED3).

Table 2-7 Jumper Settings

2.3 Currently implemented hardware configurations

As in chapter 2.1 already mentioned, there are 4 different topologies. These are also available by configuration map via *UniCoDes*. We will briefly discuss the possibilities within these hardware configurations / topologies.

- *Topology 1: Analog input, No selector unit*

- This topology supports no selector definitions, by default it selects Terminal 1.
- It supports analog input recording, with 11 sample frequencies defined by an analog PLL.
- Playback is in analog format.

- *Topology 2: Analog input, With selector unit*

- This topology supports a selector unit for the recording channel.
- It supports analog input recording, with 11 sample frequencies defined by an analog PLL.
  By default it selects Terminal 1. Though Terminal 2 is also accessable.
- Playback is in analog format.

- *Topology 3: Digital input (I²S MUX input), No selector unit*

- This topology supports no selector definitions.
- It supports digital input recording via the *MUX IN* terminal / header.
  The recording sample frequency is defined by the WS-input signal.
- Playback is in analog format.

- *Topology 4: Analog input (pre-defined ADC-X-tal), No selector unit*

- This topology supports no selector definitions, by default it selects Terminal 1.
- It supports analog input recording with one fixed sample frequency defined by the X-tal.
- Playback is in analog format.

If you have a different design (firmware and/or hardware) in mind, you should also read chapter 3.5.

3. Configuration Data Structure

This chapter describes some interesting features of the USB-CODEC in combination with the possibilities of Win98.

3.1 Working with Dynamic Bass Boost (DBB)

As you can read from the datasheet, address 14h is the register which contains the value for the DBB value.
If this value is equal to 00h = 0 [dB], bass boost will not be reported to the USB host by the device. You will not see any DBB control box if this is the case. Should you enter any value not equal to 00h, the bass boost will be reported to the host and you will see a check-box in the *Speaker Control* - *Advanced* menu of Win98.
Without any detection you could create clipping of the output signal. The USB-CODEC can detect this sort of situations and reduce the *bass* and or *treble* to prevent clipping. This way we create a *dynamic bass boost* control.

Important note:
If you activate the *bass boost* and your *volume slide-bar* is at it's maximum in combination with *clipping prevention mode* ON, then this will result in no audible effect when the slide-bar of the *bass* or *treble* is activated ! To be able to hear the effect of the *bass* or *treble* you must regulate the *master volume* to a medium level.

3.2 Working with Default Volume

It is possible that you desire a specific volume at startup of the USB-CODEC. This is an option which the USB-CODEC does support by means of address 15h. The value in this address will be loaded the first time Windows starts up and this wil be applied to the volume slide-bar in the *Speaker Control* menu.
Every time you re-plug the USB-CODEC application board, the value in the respective address will be ignored and overruled by the most recent value of Windows (set by the user earlier).

3.3 Working with Clipping Prevention and Distortion Level

If clipping prevention of the audio signal is required during playback, you can set the respective option in address 0Bh. When the sum of the *Volume*, the *Bass Boost* and the *Bass* or the *Volume* and the *Treble* is above clipping prevention level, the actual *Bass* or *Treble* boost is reduced so that the result is the programmed distortion. If the *Volume* decreases again, the selected *Bass* and/or *Treble* setting will be restored.
There is the possibility to accept a specific amount of distortion if this should be in your interest. You can set the distortion level in address 0Ch in steps of 1 dB. In mathematical / programming representation we can write the following:

```
Repeat
  If ( Volume + Bass + DBB >= DistortionLevel ) Then
    Decrease Bass
  End if
  If ( Volume + Treble >= DistortionLevel ) Then
    Decrease Treble
  End if
Until ( Volume + Bass + DBB < DistortionLevel ) and ( Volume + Treble < DistortionLevel )
```

3.4 Product ID (idProduct) and Vendor ID (idVendor)

The Philips USB-CODEC provides a way to customers to define their specific Vendor ID and Product ID if this is desired. In that case, the customer first has to contact *Microsoft* to make an official entry in the product and drivers listing of *Microsoft*. The result will be that the values entered in the respective addresses of your configuration map will be read and recognized by Win98 the first time that the USB-CODEC is connected to the USB-host & Win98. This will activate Win98 to set the defined VID and PID in the systems registers and load the neseccary drivers to operate the USB-device properly. See chapter 7 how to program your specific VID and PID and corresponding strings into the configuration I²C-EEPROM.

3.5 Other possible hardware configurations

In case that you wish to develop your own hardware vs. firmware, there is a tool-package in development just for this purpose. More information can be requested by your nearest FAE. They will inform you and if available supply you with this tool or otherwise forward you to the respective Philips engineer who is responsible for the development and/or support of this tool.

4. Customizing your design

4.1 Handling Device Configurations

As has been explained in the previous chapter, there are in principle 5 different configurations that can be used to configure the UDA1325PS for use within a USB Audio System:

- 4 internal configurations that can be selected via external circuitry
- 1 external configuration in an I²C-EEPROM that is downloaded via I²C protocol

The external configuration basically leaves all possibilities open for specific settings but it involves the use of an external I²C-EEPROM of which the contents can be generated with *UniCoDes* (see next chapter).

4.2 Configuring the USB-CODEC

Philips Semiconductors provides a program called the *Universal Configuration Designer,* shortened to *UniCoDes*, that greatly simplifies the generation & editing of a configuration data set.

Let's make things universal !

4.3 The USB-I²C communication driver

This driver allows the control of any additional I²C devices through the USB communication path.
A software driver is available from Philips Semiconductors (currently only for Windows 98) that offers a simple API for software packages that need to setup or control these additional devices using the industry standard I²C bus. See chapter FAQ how to get this driver.

5. USB Contact addresses

Basically all relevant information regarding USB is available on the Web under address

http://www.usb.org/developers/

You can e.g. request a new manufacturer's or product ID to use in your USB product, you can find recent news and you can of course download the latest USB specifications.
To stay updated, it is furthermore important to look into the webpages of Philips for the latest datasheet, application notes and software tools etc. You can contact the following address for this:

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

6. Glossary

The following explains a number of terms that are used in this document:

Term	Explanation
ADAC	Asynchronous Digital Audio Converter.
ASR	Audio Sample Rate Redistributor.
DAC	Digital to Analog Converter.
CODEC	CODer – DECoder
Descriptors	Set of registers (hard- or software) that describes the device capabilities to the USB host.
Device Class	Class of Devices that share certain features e.g. Audio Device Class, Human Interface Device Class …).
EEPROM	Electrically Erasable Programmable Read Only Memory.
Endpoint	Actual data sink/source in a device. A device can have several endpoints.
GPIO	General Purpose Input / Output.
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.
MMU	Memory Management Unit.
Port	USB connection on a Hub Device, enabling one USB Device (possibly another Hub) to be connected.
PSIE	Philips Serial Interface Engine.
UDAO	Universal Digital Audio Output.
USB	Universal Serial Bus, max. speed is 12 Mbit/sec.
Windows™ 98	Microsoft ® Operating System, upgrade to Windows™ 95.

Table 6-1 Glossary listing

7. FREQUENTLY ASKED QUESTIONS

* In case of Problems

Should you still experience problems in defining the most appropriate application for 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 or will get you into contact with a USB expert within Philips Semiconductor.

* Frequently asked questions listing

We have provided a listing of frequently asked questions how to setup a specific configuration or solve a problem. Some of them might be interesting for you and maybe could answer a question you had in mind. The listing here below gives a summarize of the questions which will be answered in the folllowing pages:

Page	Question
30	**How can I setup 6-pins I^2S configuration ?**
31	**How can I setup the USB-CODEC to test Volume control and Audio Mute?**
32	**How can I select an internal configuration map?**
33	**How can I set I^2S recording (Topology 3) ?**
34	**How can I set OSCAD recording (Topology 4) ?**
35	**How can I control the selector ?**
35	**How can I control the selector within Win98 ?**
35	**How can I control the PGA within Win98 ?**
37	**I would like to change the HID functionality of the GP HID-input pins …**
37	**HID input and/or output functionality does not work ?**
38	**I don't want to use a selector unit …**
38	**I don't want to use an expander …**
38	**What are the contents of the internal mappings used in the latest firmware ?**
38	**I can't hear the effect of bass or treble …**
39	***(Dynamic) Bass Boost* in the advanced menu of the speaker-controls …**
39	**I am using an electret microphone, the gain (PGA) is insufficient …**
40	**Where can I get the latest versin of *UniCoDes* ?**
40	**How can I implement USB-IIC communication into my application software ?**
40	**What Audio software tool(s) does Philips Semiconductors recommend ?**
41	**If I insert an EEPROM and restart Win98, the USB-CODEC is NOT recognized ?**
41	**I would like to set the mute on active at start-up of Win98 …**
41	**Why does the value of C14 has to be <2uF instead of 47uF ?**
42	**What is the difference between the N103 and N106 silicon/firmware releases ?**
42	**What is the correct supply voltage for Vdde of the USB-CODEC ?**

Table 7-1 *Frequently asked questions* summarize.

*** Question : How can I setup a 6-pins I^2S configuration ?**

*** Answer/solution :**

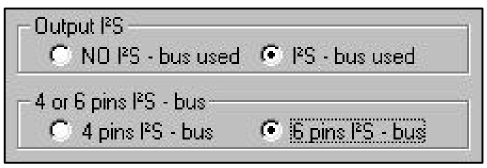
Follow the steps below to setup the UDA1325PS QFP64 PCB for 6-pins I^2S configuration.

Step 1. If *UniCoDes* is installed you can directly continue with Step 2.
Otherwise, please download the software tool from the USB-AUDIO webpage and install it on your Win98 PC.

Step 2. From the File-menu, select the command "New Design" and select the USB-CODEC /N106 release. Select the desired topology, for this example we recommend "Topology 2".

Step 3. From the File-menu, select the command "Edit via Palett". This will pop-up the configuration editor.

Step 4. Select address 0Fh and activate the following items:



Step 5. Go to the Design-menu and select "Device Specific Configuration".You will see that address 0Fh has the value E2h.

Step 6. You may decide to save your design into a file before continuing.

Step 7. Program the hexadecimal values to your I^2C EEPROM on the CODEC board with your personel I^2C programmer or use the embedded programmer within *UniCoDes*. See the Help-menu for additional hardware that you need to use the embedded programmers.

See also chapter 2.2.6 for more information.

Step 8. In this configuration mode, you will also need the expander unit in order to provide the Connect / Disconnect functionality. This expander unit with some addional circuitry is shown on page 20, on version 1.4 of the demoboard the expander unit is already available. Now, remove jumper J2 and J1 and then connect pin nr. 4 (P0) of the expander to the intersection of R11 and R12. Connect the I^2S datalines (SW, BCK, D) together with a flat-cable. Double-check that the pins are correctly connected !!

Step 9. Now, to read in the newly programmed configuration, you will have to press on the RESET button (S1) on board of the application board. Wait until enumeration has finished (a few seconds until mouse pointer is showing idle-processing).

Step 10. Play a WAV-file to check your setup. You should hear sound during playback.

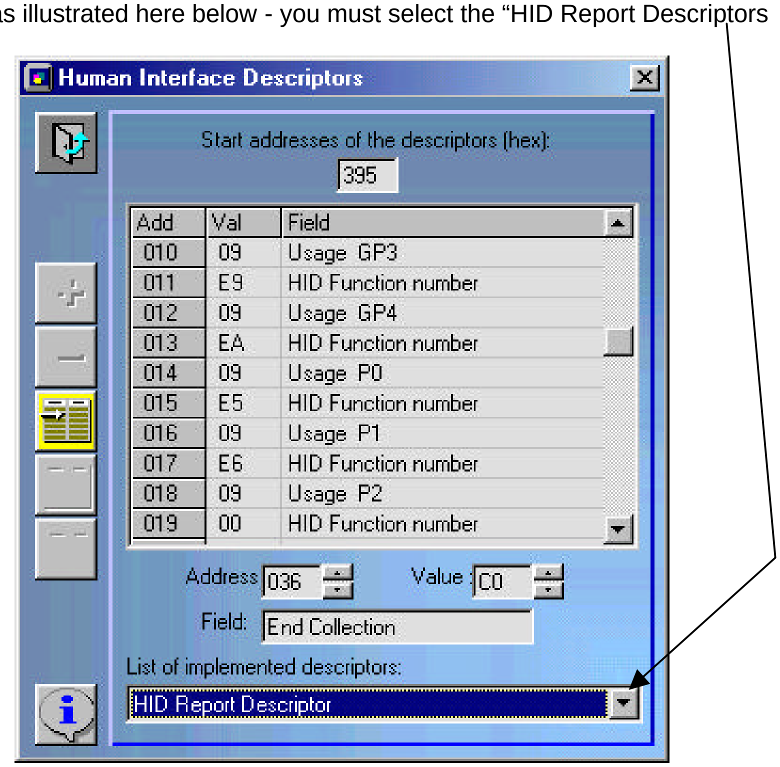
*** Question** : **How can I setup the USB-CODEC to test Volume control and Audio Mute?**

*** Answer/solution** :

Step 1. If *UniCoDes* is installed you can directly continue with Step 2.
Otherwise, please download the software tool from the USB-AUDIO webpage and install it on your Win98 PC.

Step 2. From the File-menu, select the command "New Design" and select the USB-CODEC /N106 release. Select the desired topology, for this example we recommend "Topology 2".

Step 3. From the Design-menu, select the command "Human Interface Descriptors". Once you are prompted to the new menu - as illustrated here below - you must select the "HID Report Descriptors



Step 4. As you can see, HID Input 1 is defined as *Volume Up*, HID input 2 as *Volume Down* and HID input 3 as *Mute control*. Remember for future applications that you can change the functions if needed. Exit this menu and program the configuration into your I²C EEPROM.

Step 5. Unfortunately, the first official release of Win98 FE had a bug which results in NO HID control via our USB-APRP device. The USB-DAC doesn't have this problem, it is only in combination with the USB-APRP. Within Win98 SE, this problem is solved.

*** Question : How can I select an internal configuration map?**

*** Answer/solution :**

Step 1. If you plug-in your USB-cable to the USB-CODEC board, by default, the configuration inputs GP3 and GP4 are both on a high level (pull-up's). The corresponding internal map will be loaded if there is no external I²C EEPROM inserted with it's first two byte addresses programmed as 55h and AAh.

Step 2. To select any other internal map, setup a circuit like illustrated in figure 2-8 and/or 2-9.

Step 3. Connecting a diode to one of the two general purpose pins, GP3 and/or GP4, will select a low level on start-up. See datasheet or the latest FRS document for the contents of the internal map that corresponds with the selected digital levels (HIGH or LOW). We have summarized the contents of the internal maps of the latest firmware (N106 = SW1.1.1.19) in the table below:

Int. Config	***Company***	***Diodes to insert***	***Feautures***
0	Philips Sound & Vision	No diodes	Topology 1 (PGA=27dB)
1	Philips Semiconductors	Only V4	Topology 1
2	Harman JBL	Only V5	Topology 1 (4-pins)
3	Creative	Both diodes	Topology 1

Table 7-2 Internal maps of the latest firmware release (SW1.1.1.19 ⇨ COD11119.HEX)

*** Question : How can I setup I²S recording (Topology 3)?**

*** Answer/solution :**

Step 1. From the File-menu within *UniCoDes,* select the command "New Design" and select topology 3 to start with. See also chapter 2.3 for more information about this topology.

Step 2. If you have an I²S source which is outputting a different sample frequency than 48kHz at the WS line, then adjust address 17h within the *Device Specific Configuration* map according your source. This action will also automatically calculate the *wMaxPackageSize* to determine the maximum bandwidth of the recording & playback channel.

Step 3. Program the values of topology 3 into your I²C-EEPROM.

Step 4. Connect the I²S source on the "MUX IN"-labled input socket and be 100% sure that all the lines are connected. Especially with flat-cables a mistake is very easily made…
Also, after connecting the I²S lines, measure the levels with a scope !

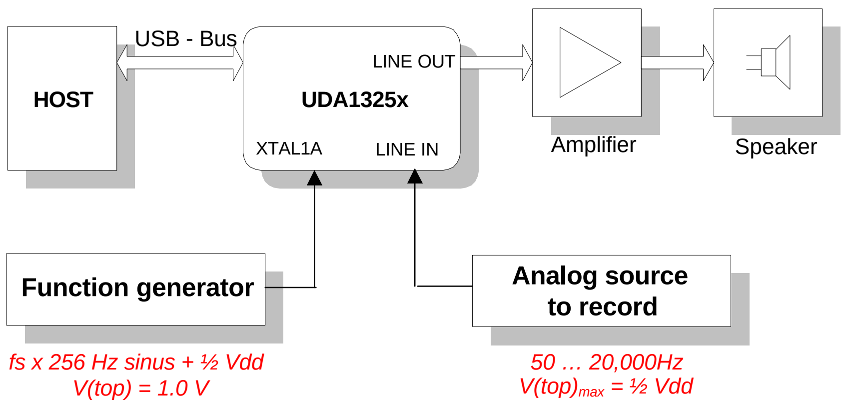
Step 5. Start recording with your specific recording software tool with the correct recording sample frequency.

*** Question : How can I setup OSCAD recording (Topology 4)?**

*** Answer/solution :**

Step 1. From the File-menu within *UniCoDes,* select the command "New Design" and select the USB-CODEC /N106 vs. topology 4 to start with. See also chapter 2.3 for more information about this topology.

Step 2. Now connect a crystal or function-generator which is outputting a sinus of *fs x 256* Hz. The setup should be as following:



If you have a crystal or clockgenerator-source which is outputting a different sample frequency than 48kHz x 256, then adjust address 17h according your generator / crystal. This action will also automatically calculate the *wMaxPackageSize* to determine the maximum bandwidth which will be used for recording and playback. If you are using a function generator, set the top value of the sinus to 1 V. In general, it should give the same form and amplitude as the *48 MHz* crystal. This means that you also need an offset of *½ Vdd* to this clock signal. Only the frequency should / could be different. Don't forget to remove the capacitor C6 !

Step 3. Now, program the values of topology 4 into your I²C EEPROM.

Step 4. Start the recording and don't forget to set the right sample frequency in your recording program !

*** Question : How can I control the selector ?**

*** Answer/solution :**

Step 1. First take a look into chapter 2.1.1.1 and examine table 2-2.

Step 2. The best thing to do here, is to program these different values and see what happens. You can measure the logic level of the selector via GP2. This pin is also directly connected to jumper J1 and gives the possibility to connect a Volt-meter to these pins.

Step 3. To determine the effect of the PGA settings (address 06h and 07h) of the respective terminals, you should change the values in the extreme ranges (max and min). This way you can also determine that the correct PGA is implemented into your own application.

*** Question : How can I control the PGA within Win98 ?**

*** Answer/solution :**

You will see that there is no way to control the PGA within Win98 by means of any control button. This is because we do not support this feature within the N103 and N106 firmware releases. At the time being when this note was written, the N104 was under development. The N104 release includes the possibility to control the PGA via the a slide-bar within the *Recording Properties* or speaker-menu of Win98.

*** Question : How can I control the selector within Win98 ?**

*** Answer/solution :**

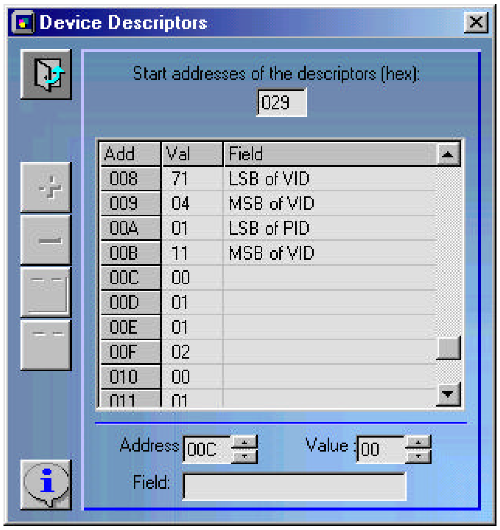
The same can be said about the selection of input channel/terminal. The N104 release supports a *Selector Feature Unit* in which it displays a check-box within the speaker-menu of Win98 to make a selection between one of the two input terminals. So, the N103 & N106 releases do not support this feature.

*** Question** **: How can I set a different VID or PID and corresponding strings ?**

*** Answer/solution** **:**

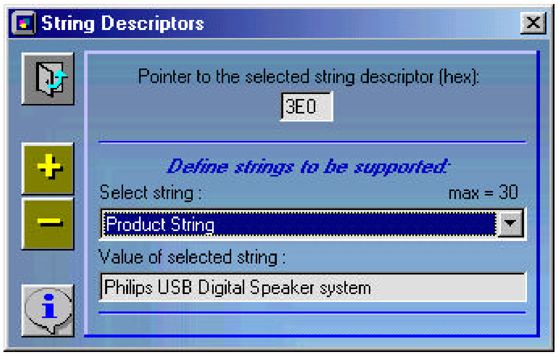
Step 1. From the File-menu within *UniCoDes,* select the command "New Design" and select the USB-CODEC /N106 vs. topology 2 to start with. See also chapter 2.3 for more information about this topology.

Step 2. From the Design-menu, select the command "Device Descriptors" :



Step 3. Change the values according your company's VID and the ID of the product. Close the menu when finished.

Step 4. From the Design-menu, select the command "String Descriptors" :



Step 5. Set the respective string(s) to your own wishes.

Step 6. Once you are finished editing, you can program it to the EEPROM. After reset of the USB-CODEC, you will see that the *product string* is shown at *recognition for the first time*. Win98 will try to install the needed drivers and after this procedure these strings will not be shown again at re-plug. Though, the strings can be requested at any time via specific application-programs like *USBCHECK.EXE* from *Microsoft/Intel.*

*** Question : I would like to change the HID functionality of the GP HID-input pins …**

*** Answer/solution :**

Step 1. Read chapter 2.1.1.1, section HID inputs (page 10, fig 2-4).

*** Question : HID input and/or output functionality does not work ?**

*** Answer/solution :**

Step 1. Read chapter 2.1.1.1, section HID inputs (page 10, fig 2-4).

Step 2. Now read question/answer on page 31, *Step 6.*

*** Question : I don't want to use a selector unit …**

*** Answer/solution :**

Step 1. Select topology 1. This topology is exactly the same regarding configuration and all the descriptors as topology 2, except that there is no support (output) for a selector. On default, the output of GP2 is always at low level. Therefore Terminal 1 = LINE INPUT will be selected.

*** Question : I don't want to use an expander …**

*** Answer/solution :**

Step 1. Select any topology you want.

Step 2. Go to address 0Bh and select "No Expander". This will ensure that the expander will not be scanned or controlled by the USB-CODEC.

*** Question : What are the contents of the internal mappings used in the latest firmware ?**

*** Answer/solution** : See page 32, table 7-2.

*** Question : I can't hear the effect of bass or treble …**

*** Answer/solution :**

If the clipping level is active and at the same time your speaker volume is set to *max.*, the clipping mode will prevent any distortion by reducing automatically the total gain to a distortion-free level. This means that if you change the levels of bass or treble, you will not be able to hear any effect of this action. You will have to lower the volume to e.g. medium level to hear the effects of *bass, bass boost (if enabled in advanced menu)* and/or or *treble*.

*** Question** **: I don’t want (Dynamic) *Bass Boost* in the advanced menu of the speaker-controls …**

*** Answer/solution** **:**

Step 1. Select any topology you want.

Step 2. Go to address 14h and set the value to 0 dB.

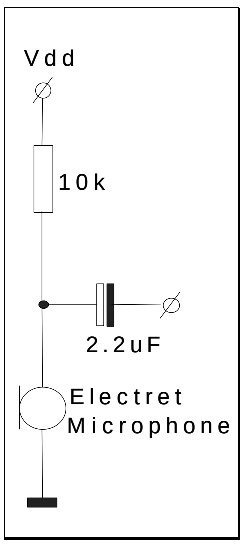
Step 3. After programming the configuration and resetting the USB-CODEC, this will ensure that the button will not be shown in the speaker and advanced menu.

*** Question** **: I am using an electret microphone, the gain (PGA) is insufficient …**

*** Answer/solution** **:**

The fact that the max. gain of the PGA is 27 dB is a problem for a standard electret microphone. There are basically two solutions here.

- The first one is by designing an external amplifier into your application.
  In case that you did not know, electret microphones need a DC offset (biasing) to be able to work. You can do this by applying a capacitor and resistor as shown here below:



There are also electret microphones which have these two components integrated into the housing. The choice which type to use / support depends on costs or maybe other issues.

- The third one is by amplifying the recorded sound by means of dedicated software application. *Cool Edit**, for instance, is capable of amplifying the sound up to 300% of the original.

** Cool Edit is a program created by the company Syntrillium. See page 40 for more information.*

*** Question : Where can I get the latest version of UniCoDes ?**

*** Answer/solution :**

All documentation and tools regarding our USB-CODEC & USB-DAC products can be downloaded from our USB-AUDIO website. See chapter 5 – USB contact addresses for more information.

*** Question : How can I implement USB-IIC communication into my software application ?**

*** Answer/solution :**

Philips Sunnyvale (USA) has a driver developed for USB-IIC communication purposes. This driver can be downloaded from the USB-AUDIO webpage. The source provides you with software application examples how to use the driver in question.

*** Question : What Audio software tool(s) does Philips Semiconductors recommend ?**

*** Answer/solution :**

At our laboratory / development environment we use the program *Cool Edit* from the company *Syntrillium*.
You can download the demo-version from the following site:

http://www.syntrillium.com

You will see that there are many possibilities within the context of AUDIO functionality with this program and the most important feautures, like sample frequency and 8/16 bits and stereo/mono selections, are also settable.

*** Question : If I insert an EEPROM and restart Win98, the USB-CODEC is NOT recognized?**

*** Answer/solution :**

This behaviour could occur when an I²C-EEPROM with valid data is inserted during restart of Win98 or power-up of the PC and there is also an I²C Single Master Controller connected. The reason for this behaviour is probably because of the restart-procedure of Win98 which disables the parallel port at the critical moments when the I²C data is read from the EEPROM. When you check the status of the dataline (SDA) with a scope-prober, you will discover that the SDA line is held low during this enumeration attempt. This will result in no device recognition.

The solution is simple. You will have to remove the connection between I²C Single Master Controller and USB-CODEC during power-on of the PC or restart of Win98:

There are actually two kind of restart procedures which is important to know. Both methods are selected with the normal *Restart* procedure in Win98, the difference here is when restart is activated with the ***Shift*** key pressed until the DOS-prompt "*Windows is now restarting…"* is displayed. In this situation, the USB-CODEC will enumerate without any problem, but in the other case (no ***Shift*** key pressed) the USB-CODEC will not be recognized.

*** Question : I would like to set the mute on active at start-up of Win98 …**

*** Answer/solution :**

For this, you will have to select address 09h and press on the respective button to activate this option:

Though, like the *Start-up default Volume* level (address 15h), the mute setting will be ignored by Win98 and set according the value set by the user before. The reason for implementing both functions anyway, is the possibility that other operating systems do not support the same functionality as Win98 does. In such case, the USB-CODEC can offer the functionality instead.

*** Question : Why does the value of C14 has to be <2uF instead of 47uF ?**

*** Answer/solution :**

<u>IMPORTANT : READ CHAPTER 2.1.1.1</u>

*** Question : What is the difference between the N103 and N106 silicon/firmware releases ?**

*** Answer/solution :**

The N106 release is exactly the same as the N103 release in terms of functionality, the only difference is the upgrade in the USB-library firmware in the N106 to improve enumeration behavior with OHCI based PC systems.

*** Question : What is the correct supply voltage for Vdde of the USB-CODEC ?**

*** Answer/solution :**

In the datasheets of the UDA1325 USB-CODEC from -10 th of May 1999 - it is mentioned that the Vdde (Supply voltage periphery) should be 5V +/- 0.25V. It should be stated that the minumum voltage may be 3.0V as long as Vdde is NOT lower than Vddi (Digital core supply) or the analog supplies.

8. References

Title	Notes
UDA1325.PDF	Datasheet of the device UDA1325PS.
DML98021.PDF	General application note for using the UDA1325PS.
DML98022.PDF	Detailed application note for programming and use of the UDA1325PS.
SCHEM.PDF	Schematic of the UDA1325PS evaluation board.
BOARD1.PDF	Components on the board (top view).
BOARD2.PDF	Components on the board (bottom view).
COD1119.HEX	The firmware version that is implemented into the N103 silicon release.
COD11119.HEX	The firmware version that is implemented into the N106 silicon release.
REP0021.PDF	Report on new proposal for Diconnect/Connect circuit to improve enumeration behavior of the USB-APRP & USB-CODEC.