

DatasheetDirect.com

Your dedicated source for free downloadable datasheets.

- Over one million datasheets
- Optimized search function
- Rapid quote option
- Free unlimited downloads

Visit www.datasheetdirect.com to get your free datasheets.

This datasheet has been downloaded by <http://www.datasheetdirect.com/>

**USB - DAC
configuration editor
version 2.00**



USER MANUAL

USB - DAC
configuration editor
version 2.00

Report No.: RNB-C/40/98XII - 1231

Sedat Serper

**Product Development – Design & Application, Consumer Systems Nijmegen
The Netherlands**

Date : December 19th, 1998

Preface

This paper describes the USB - DAC Configuration Editor software consisting of a detailed user-manual, which describes the possibilities of this program. This tool is specially created for the UDA1321x /N101 and UDA1331H /N101 USB - DAC IC's.

The UDA1321x and UDA1331H are USB compliant audio DAC's. Both IC's are capable of replacing the whole sound-card in today's (1998- 1999) PC systems. This, enabling manufacturers of speaker systems to integrate the USB - DAC IC into their own product which is plug-and-play compatible, opening a whole new market.

This software package runs only on Windows 95 and 98™ based operating systems.

CONTENTS

1 USB – DAC configuration Editor v2.00

- 1.1 The software package
- 1.2 Recommendations
- 1.3 Installing and starting the program
- 1.4 The software features
- 1.5 Using the program
 - 1.5.1 General
 - 1.5.2 Loading a configuration
 - 1.5.3 Editing a configuration
 - 1.5.4 Saving a configuration
 - 1.5.5 Scrolling through results
 - 1.5.6 Help command
 - 1.5.7 Reading and Writing a configuration into an I²C - EEPROM
- 1.6 Closing the program

2 Future plans

- 2.1 File formats
- 2.2 More features

3 Revision history

- 3.1 Version 1.00, first release
- 3.2 Version 1.01, second release
- 3.3 Version 1.02, third release
- 3.4 Version 1.03, fourth release
- 3.5 Version 1.04, fifth release
- 3.6 Version 2.00, sixth release

1 USB – DAC configuration editor v2.00

1.1 The software package

The software package consists of a total of three setup disks. The first disk contains the executable setup file named as SETUP.EXE.

1.2 Recommendations

A mouse is not strictly necessary but advisable, since Windows © and programs within Windows © are mostly controlled by mouse.

1.3 Installing and starting the program

If you installed this program and same version before, please uninstall it first before you continue. You can use the *Add/Remove Programs* button in the menu *Settings - Control Panel*.

Be sure that you have nothing important saved in the setup-folder / directory ! You should remove them yourself before you continue with *Uninstall*, otherwise the uninstall procedure will not be able to remove the directory where you installed the editor.

Press on the button  and select the *Run* option.

This will pop-up a new window in which you can browse and select any program to start. In this case, select the file : SETUP.EXE from the first disk.

This executable file will install the files to a user-defined directory. After the installation is completed, please run the editor by selecting it in the menu *Programs*. You will be prompted to make a choice if you would like to add the shortcut to this program on your desktop. Follow the instructions.

Notice the icon at the upper left corner on your desktop after the installation of the icon has completed. This icon is the shortcut to the executable. Doubleclick on the icon  to start the program.

1.4 The software features

This version of *Configuration Editor* supports the following features:

- 2 file formats are supported.
- Easy string editing by means of a separate menu in the same style of the *USB-CODEC configuration editor*.
- Previous viewing of address values in form of a summary.
- Default option included.
- Topic search for easy access of the addresses.
- Help for program-use included, added extra features such as schematic(s).
- 32-bit DLL files.
- Official Windows setup with automated *Uninstall* capability.
- This version is equipped with the possibility to read and write your I²C- EEPROM via a *Single Master Controller* (see chapter 1.5.7 for more information).

1.5 Using the program

1.5.1 General

After starting the program the startup screen will appear, as shown below:



Fig. 1

You can start either by the *Editor* by selecting the *Edit Configuration* button, which loads the defaults configuration or select the *Load Configuration* button to load one of the supported file formats : *.mem or *.adr files. This chapter is for the first time equipped with the possibility to read from and write to the I²C - EEPROM via a *Single Master Controller*, so you could also start load the data of the I²C - EEPROM and start editing the respective data. The next chapter describes how you can load a file and edit it.

1.5.2 Loading a configuration

Select the *Load Configuration* button to initiate the window for loading files. You can use also your keyboard by pressing on the TAB key until the dashed marker appears on the button you want en strike the space bar to activate it (for more information about disabled buttons and functions of them, please read chapter 2.2).

Once the *Load File* window is on your screen (like illustrated in **Fig. 2**), you can browse freely through the available drives and directories to select any configuration file.

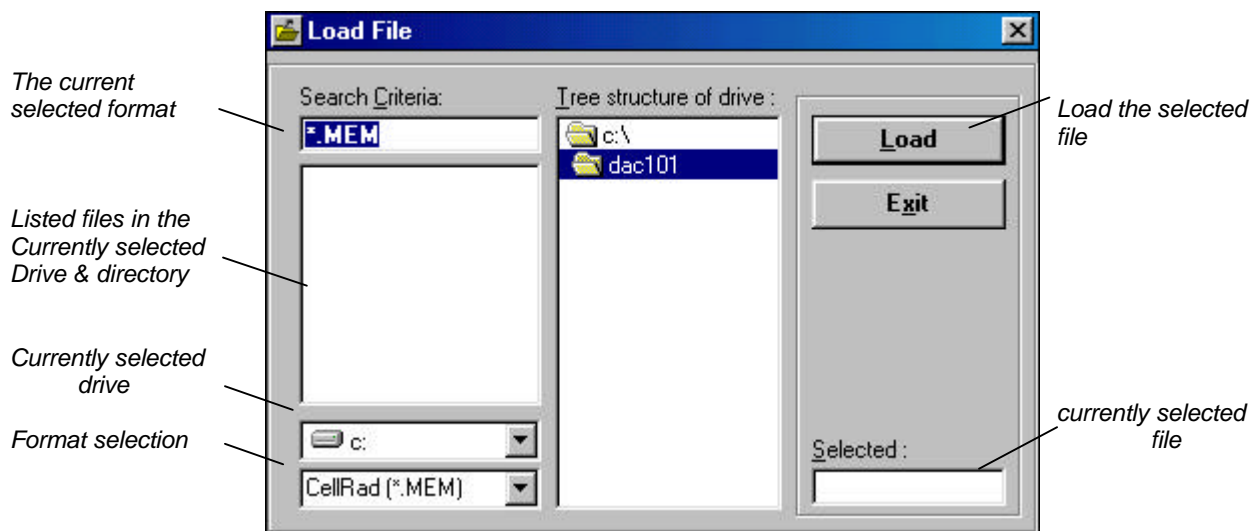


Fig. 2

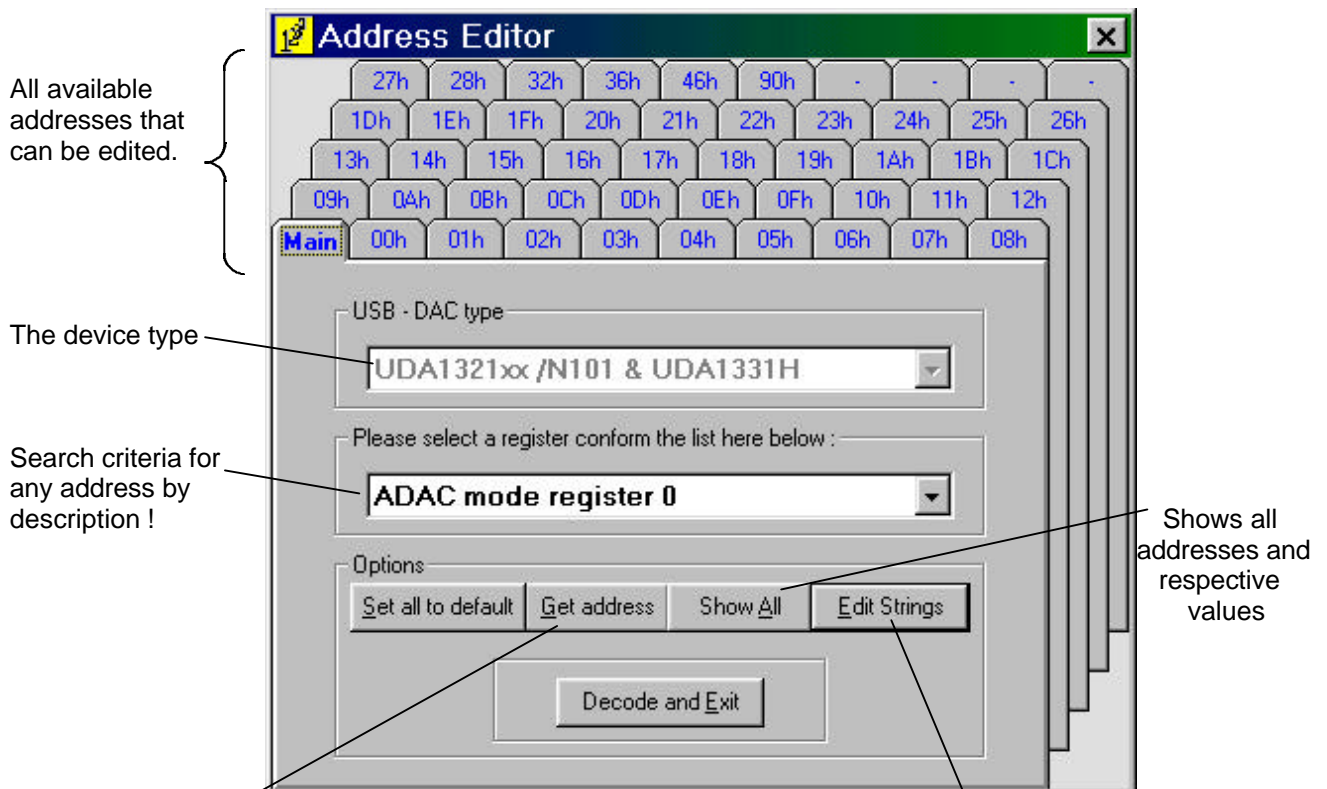
Depending on the format you have selected, only the respective extensioned files will appear on the listed file sub-window. If you need to look for files with other formats, press on the corresponding spin-button control (see **Fig. 2**) to display the list of available formats and select the one you want.

If you finally found your file, select this and press on the *Load* button to load it and decode the contents. This version is equipped with a procedure which checks the data consistency / validation. If the data in the file is erroneous / invalid, the program tells the user which error has occurred and selects the default values after the warning.

1.5.3 Editing a configuration

It is not necessary to load a file to be able to start the editing section with the *Edit Configuration* command. Once this button is pressed and released, the program will automatically detect whether you have loaded a file (file or directly from I²C-EEPROM) or not and will decode and represent the data if loaded or set its default values for all the addresses and show them instead.

The edit window looks similar as the one illustrated here below:



The topic that is selected in the Search criteria box can be called with this button.

Fig. 3

To edit the available strings

The first input box is to select the USB – DAC type of device. As you can see, this section is disabled and at this point/ stage only one device type is supported (UDA1321H & UDA1331H). If necessary, it can easily be expanded in the future.

It is rather difficult to know at which address what kind of settings are available. Therefore, we have included a search topic by means of the description of the address (select criteria). Once you have selected a register description (in the example above : **ADAC mode register 0**), you can call this TAB field by pressing on the button *Get Address*.

As you can see, if you want to edit the strings, you can do this by calling a separate menu with the button *Edit Strings*. The same method is used in the *USB-CODEC configuration editor*.

If you are finished editing you can either take a brief look at the results by pressing the button *Show All* or directly exit the editing window with *Decode and Exit*. More details will be given in chapter 1.5.4.

1.5.4 Saving a configuration

If you want to save your configuration file then select the *Save Configuration* button. This will pop-up a new window as shown below:

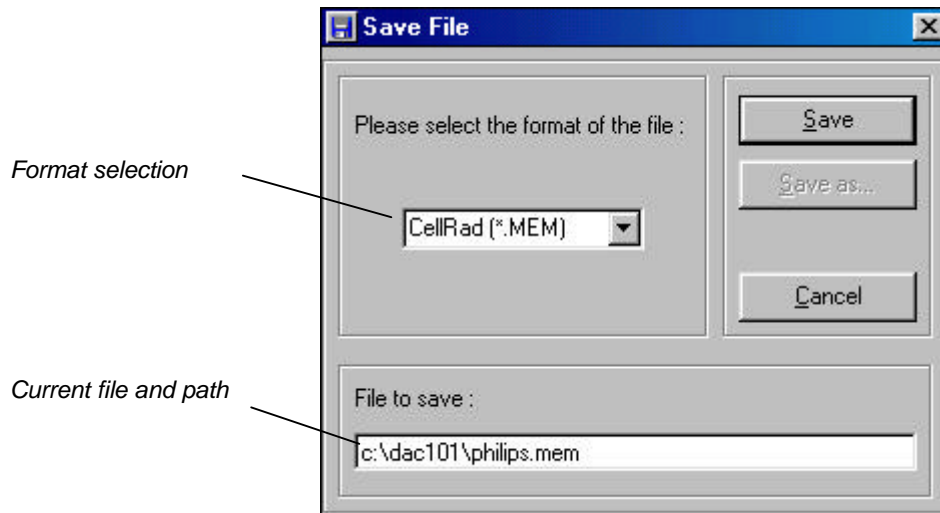


Fig. 4

Before you save your file with the *Save* command, you may change the format of the destination file if this is necessary.

If you prefer another name for the file after all, simply click on the text-box and edit the file name or path to your own wishes. As you can see, the command *Save as...* is disabled, but will be active in the coming versions of the configuration editor.

1.5.5 Scrolling through results

You don't really have to save your file to *look* what the results are in hexadecimal values with an I²C – programmer. The command *Show All* in the address editor can do this for you. Once activated you will see a similar window like presented here below:

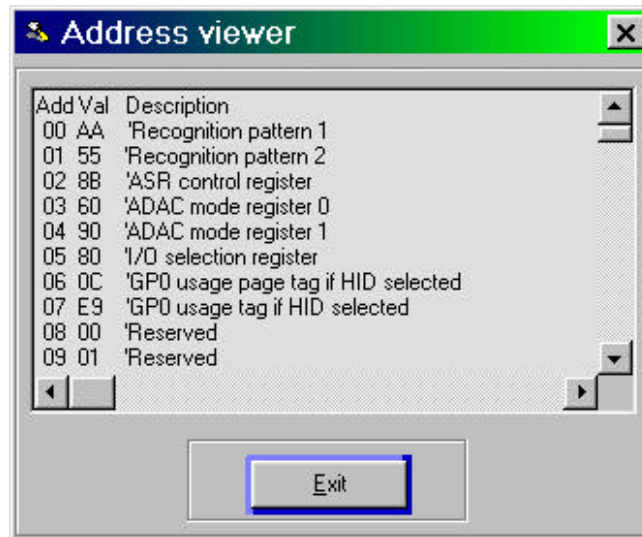



Fig. 5

Every line (except the first one) represents an address with its value and a short description of the address regarded. You can scroll through all the addresses by moving the vertical and horizontal scroll bars in the desired direction.

This format is the exact format of *.ADR files listed in the available format box in the Load and Save command windows. Furthermore, it is really advisable NOT to alter the format, since this will cause unpredictable results when loaded and decoded by this program. There is not (yet) included any error checking by any means. But you could change the hexadecimal values if needed, this should be no problem.

NOTE:



The button  on the main screen (see page 6) has the same function as the Show All command in the editor window.

1.5.6 Help command

To complete the program, a small but hopefully effective *help* option is included. In general sence, it gives you the same information as described in this user manual and some extras.

The main window of *Help* looks as illustrated below:

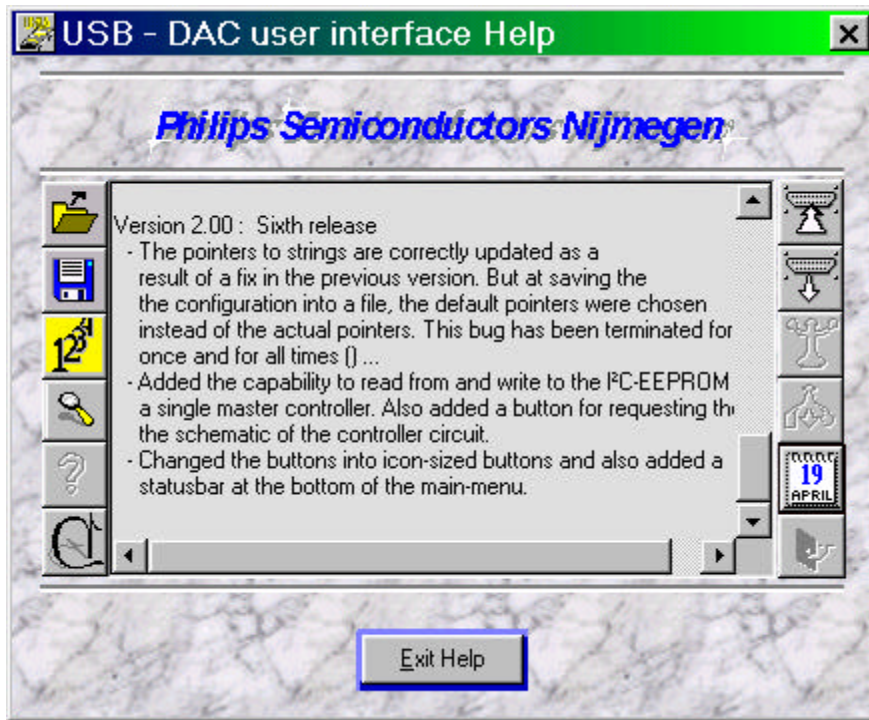


Fig. 6

You have 8 topics and 1 control button at your service. The 8 topic buttons give you each a description of the function when pressed in the main menu.

Use the vertical and horizontal scroll bars to browse through the text.

1.5.7 Reading and Writing a configuration into an I²C-EEPROM

This version of the editor is equipped with the capability of *Reading* and *Writing* an I²C-EEPROM. Both functions need a *Single Master Controller* to be able to setup the I²C -communication. A schematic of this controller is available via the Help menu and also in the main-menu (*Programming -> Via SingleMC -> SingleMC schematic*).

The buttons of the respective functions :



To read the I²C-EEPROM



To write an I²C-EEPROM

Reading an I²C-EEPROM has the risk of reading invalid / incorrect data which can cause run-time errors in the editor and causing it to terminate. Therefore, we have introduced some important checks on the data to ensure that the data is valid and editable. Each read-session is therefore finished with a small summary of these checks as shown in figure 7.

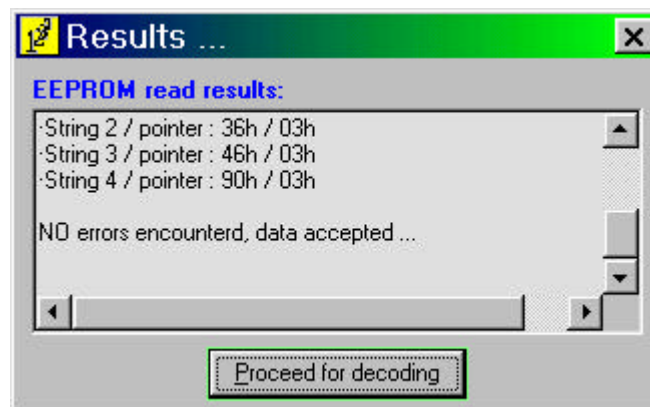



Fig. 7

The check-procedure will be extended and in more detail in the near future. A very quick indication that the data is valid and that the read-process was successful, is the surrounding color of the one-and-only button in the same window (in figure 7. named as *Proceed for decoding*). If it is red, either the data-read-process was interrupted for the displayed reason, or the data is invalid. Green, indicates that everything was o.k. and that you can proceed for editing the data ("proceed for decoding").

The read-processing is rather quick (2..3 seconds) in comparison with the write-procedure. Writing of any configuration is time-consuming and can take up to 15 seconds. This is partly of the method used in the DOS program. It is processing data byte-per-byte and every cycle is programmed for a delay of 10 ms. Some I²C-EEPROM's have a much shorter write-cycle delay, but to ensure that every I²C-EEPROM can be written without worrying about cyclus-timing, we have set this value on default at 10ms.

By the way, there are Windows-based DLL files to setup the I²C-communication via a *Single Master Controller*, but these are still under test. Ultimately, this will replace the DOS-based programs once available ...

1.6 Closing the program

To close / exit this program, use the button: 

It is important to know that you will NOT be prompted if you want to save any changes (if they are made) when you select the *Exit program* button. So be carefull, although changing any configuration to your own wishes is a matter of minutes ...

This will be changed in one of the next versions of this program, a small dialog window will then appear when these sort of situations should occur.

2 Future plans

2.1 File formats

Like mentioned before, two file-formats are included which are supported by this program. The first one are files with the extension *.MEM, created and handled by an I²C programmer called *CellRad* ©. The second one is an ASCII formatted file so that you can read this file in every file-editor and get the information you want from it. It is strongly recommend **NOT** to alter the format of the file. If the format is changed you can expect unpredictable results, even program termination (NON-FATAL).

This version is equipped with data-validation checking for the first time, as was announced that it would be implemented in the future in the previous versions of this program. The same procedure at loading data from an I²C-EEPROM is used for this purpose, which will make the program more reliable towards erroneous data.

2.2 More features

You probably recognized the disabled buttons on the main screen (see page 6). Both buttons are preserved for future implementation of USB-I²C programming. These buttons will be able to program your USB-DAC in a direct fashion via the USB – bus ! This means that you don't need any other I²C programmer at all and you use only one tool to configure your USB – DAC ! At the moment we have made the first steps towards user-friendly programming capabilities via the method of the *Single Master Controller*. We hope to implement the USB-I²C programming capability in the next versions.



Let's make things better.

3 Revision history

3.1 Version 1.00, first release

Since this is the first release, no revisions are made.

If you have suggestions, found bugs or have advises please don't hesitate to contact me.
My e-mail account : sedat.serper@nym.sc.philips.com

Many thanks in regard.

3.2 Version 1.01, second release

There have been one bug fixed and one extra option is included in the Help menu:

Bug report:

Bug :

If no file has been loaded and the user activates the button **Edit Config**, the default values will be loaded automatically by the program. Exactly three addresses were not correct in their corresponding value:

Address 07h should be E9h instead of E0h
Address 0Bh should be EAh instead of E0h
Address 0Fh should be E2h instead of E5h

Result : Fixed.

Extra Options:

To inform you of the revisions that have been made, there has been also a button created in the Help menu. This will give the user an impression of the development of this program. Please read chapter 1.5.6 for details.

3.3 Version 1.02, third release

There have been two bugs fixed:

Bug report 1:

Bug :

The decoding procedure failed to write the code 03h when the string lengths were varied relative to the default values of the editor. This causes Windows to show a *Product string* which is not correct when the USB-device is installed for the first time.

Result : Fixed.

Bug report 2:

Bug :

The selected format in the combo-boxes of the *Load* and *Save* menus were erasable with any keyboard input. This resulted in a run-time error if the *Save* or *Load* command was activated.

Result : Fixed.

3.4 Version 1.03, third release

- No significant changes has been made to the editor, only installation procedures has been changed from a partly DOS environment to a 'pure' Windows environment.
- Furthermore, the whole program is transformed to a 32-bit environment by means of Visual Basic v5.0
- And there is an official Windows setup instead of a batch file. This creates also the possibility to easily uninstall the editor if needed.

3.5 Version 1.04, fifth release

- The editing of the strings are simplified by grouping them in a seperate menu / window. This menu can be accessed by the button 'Edit Strings'. The pointers to strings were not correctly updated when the string lengths were changed. Corrected this bug.
- The interface for editing of the VID and PID is conform the *USB-CODEC configuration editor's* interface.

3.6 Version 2.00, sixth release

- The pointers to strings are correctly updated as a result of a fix in the previous version. But at saving the the configuration into a file, the default pointers were chosen instead of the actual pointers. This bug has been terminated for once and for all times () ...
- Added the capability to read from and write to the I²C-EEPROM via a single master controller. Also added a button for requesting the schematic of the controller circuit.
- Changed the buttons into icon-sized buttons and also added a statusbar at the bottom of the main-menu.