

USBSID-Pico software manual

Author: LouD - generated on 2025-06-28

Table of Contents

Supported platforms	2
Drivers	2
Linux Udev rules	2
Windows driver	2
Test and configure	3
Supported software	4
Deepsid	4
SidBerry	5
Vice	6
Command-line arguments	6
vsid GTK	7
vsid SDL2	8
x64sc GTK	10
x64sc SDL2	12
Retrodebugger	14
jsidplay2	14
c64jukebox	14
sidplayfp	14
Disclaimer	14
License	14
Software License ~ GNUv2	14
Hardware License ~ Creative Commons	14

Supported platforms

In active development

Linux: Vice, RetroDebugger, SidBerry, SidplayFp, JSidplay2, USB Midi, ASID & WebUSB (in webbrowser) SID Play

Windows: Vice, SidBerry, USB Midi, ASID & WebUSB (in webbrowser) SID Play

Android: USB Midi, ASID & WebUSB (in webbrowser) SID Play

Amiga: PlaySID library

Drivers

Linux Udev rules

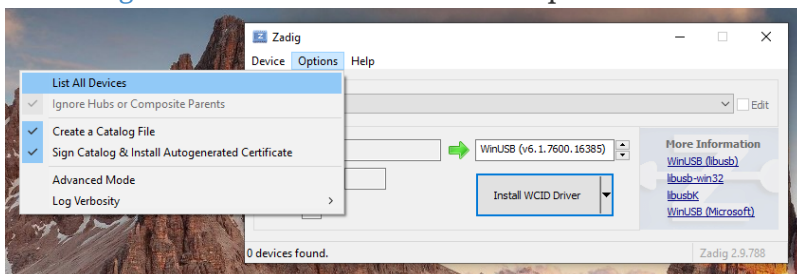
In the [examples/udev](#) directory you can find the [udev rules](#) that I use on Linux.

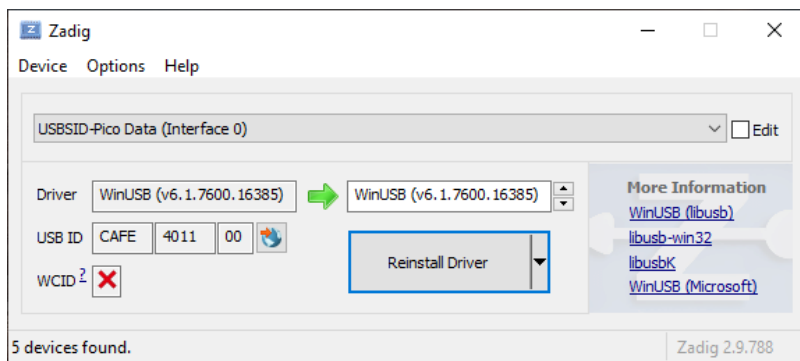
This purely an example file that you can use and change to your own needs. Steps required for this to work

```
# Check if you are in the plugdev group with the groups command
groups
# If not, add yourself to the plugdev group, then log out and in
sudo usermod -aG plugdev $USER
# Copy the udev ules file to the correct directory
sudo cp 69-usbsid-permissions.rules /etc/udev/rules.d
# Now reload the udev rules
udevadm control --reload-rules && udevadm trigger
# Not working? Try reloading the service
sudo systemctl restart udev
```

Windows driver

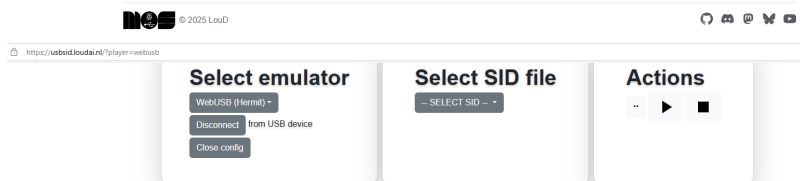
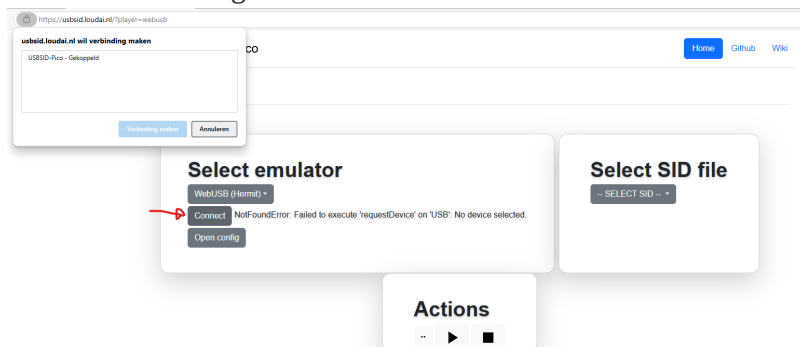
Use [Zadig](#) to install the correct driver. Replace the driver for **USBSID-Pico Data** with WinUSB.



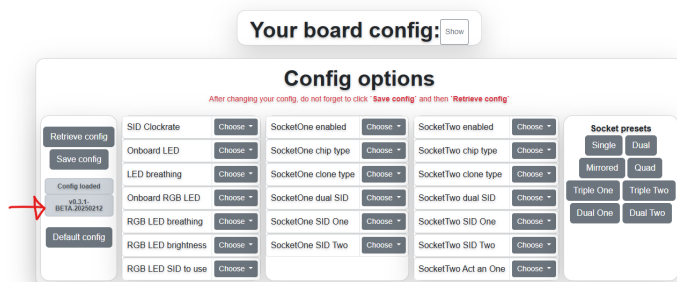


Test and configure

After setting up your linux udev rules or Windows driver check, configure and test your board on the [USBID](https://usbhid.loudai.nl/) config tool website



USBID-Pico configuration



For more information on how to configure your board, please refer to the [Firmware manual](#)

Supported software

Each section covers the basic use of USBSID-Pico with the software under that section. For more in depth information please refer to the repective website and or github repo of the software.

Deepsid

On the [deepsid](#) website you can select **WebUSB (Hermit)** or **ASID (MIDI)** for playing PSID files using a supported browser.

WebUSB: After selecting **WebUSB (Hermit)** you need to click the **CONNECT** button and select USBSID-Pico in the pop-up window. Due to the autoplay feature of Deepsid, you might have to press **STOP** a couple of times before the **CONNECT** button will work.

ASID: After selecting **ASID (MIDI)** you need to select USBSID-Pico in the dropdown box next to the line that says **MIDI port for ASID**. It might require refreshing the page a couple of times for the box to be filled with midi devices.

Supported browsers:

- **Firefox** supports ASID.
- **Chrome** supports both WebUSB and ASID.
- **Edge** supports both WebUSB and ASID.

SidBerry

[SidBerry](#) supports PSID files only and can play SID files on USBSID-Pico via CDC (default) or on ASID supporting players via command line arguments.

General Linux usage:

```
# to play sidfile.sid on USBSID-Pico
usbsidberry path/to/sidfile.sid
# to detect and print available midi devices
usbsidberry -midi
# to play sidfile.sid via ASID on USBSID-Pico where `1` is the respective port of the
ASID device
usbsidberry -asid 1 path/to/sidfile.sid
```

Basic Windows usage:

```
# to play sidfile.sid on USBSID-Pico
usbsidberry.exe path/to/sidfile.sid
# to detect and print available midi devices
usbsidberry.exe -midi
# to play sidfile.sid via ASID on USBSID-Pico where `1` is the respective port of the
ASID device
usbsidberry.exe -asid 1 path/to/sidfile.sid
```

Vice

This section covers the use of USBSID-Pico with Vice under Linux, Windows and MacOS for the GTK and SDL2 versions of `vsid` and `x64sc`

Command-line arguments

Command-line arguments override any previously configured items in your `vicerc` config file.

All three OS support the same command-line arguments for both `vsid` and `x64sc`.

However, the sound device argument is OS specific and `vsid` supports a maximum of 3 SID chips

Note that if you define extra SID's and their addresses, these addresses need to point to the required address for the SID tune or Software you want to use. The Vice USBSID-Pico driver will automatically convert each write to the in firmware configured SID chip.

```
# vsid with usbsid enabled
vsid -sidenginemodel usbsid
# Define extra SID chips and their respective addresses in de Commodore 64 memory
space
vsid -sidenginemodel usbsid -sidextra 2 -sid2address 0xD420 -sid3address 0xD440

# vsid with linux Alsa driver and usbsid enabled
vsid -sounddev alsa -sidenginemodel usbsid
```

vsid GTK

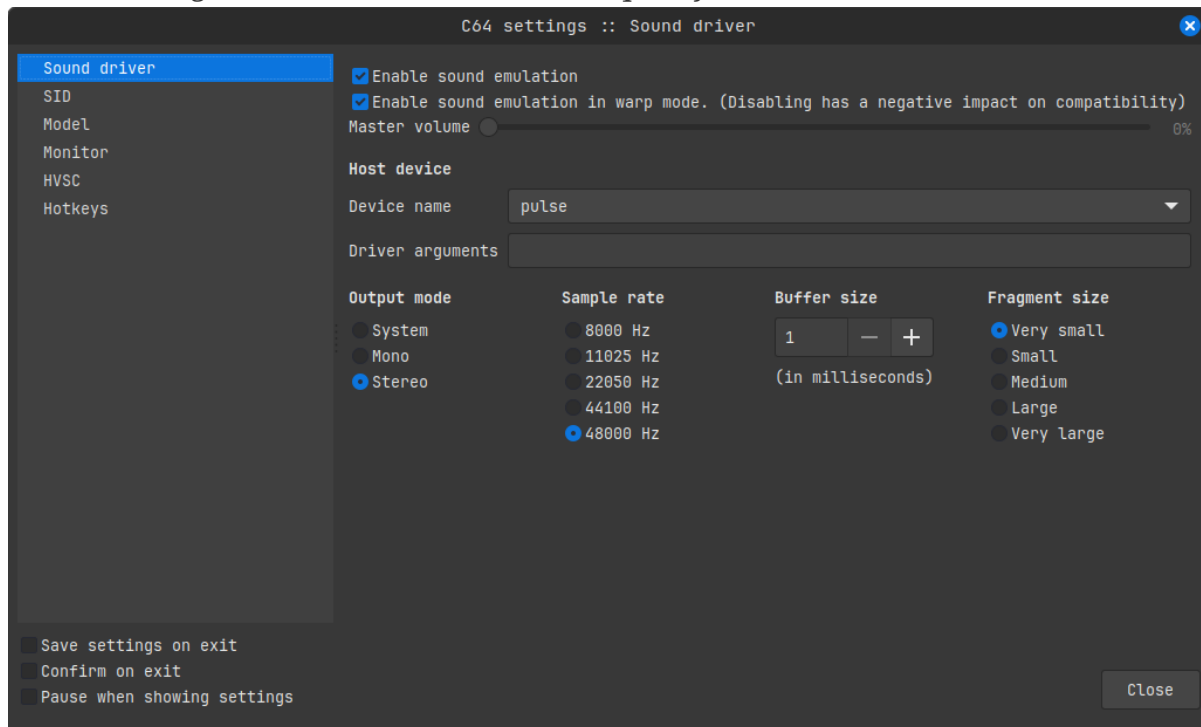
ALT+O or click Preferences/Settings to open the settings menu.

Sound configuration

The sound configuration will require tuning for the computer you use it on.

In the screenshot below you can see that for this system I use the **pulse** audio driver and have the **Buffer size** set to 1 millisecond and **Fragment size** set to very small.

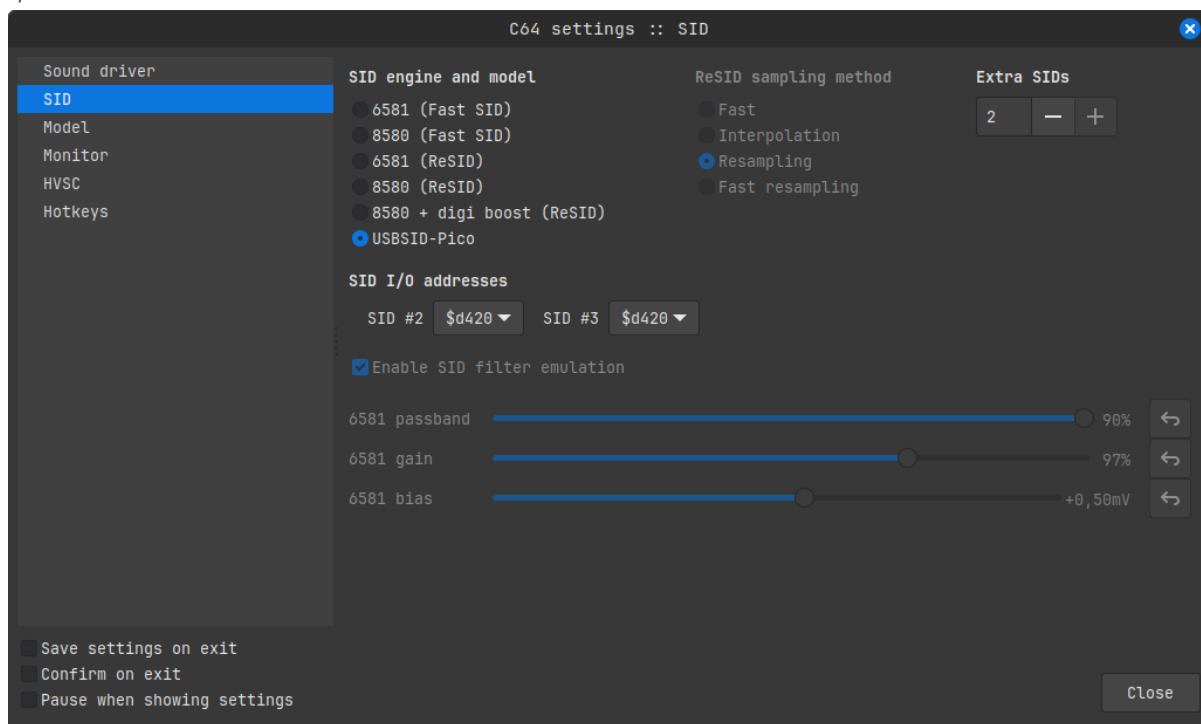
These 3 settings can make or break the audio quality.



SID configuration

For any audio to come out of the board you *need* to select USBSID-Pico as SID engine.

If you want to use more than one SID then you need to select the amount of **Extra SIDs** and set their I/O addresses.



vsid SDL2

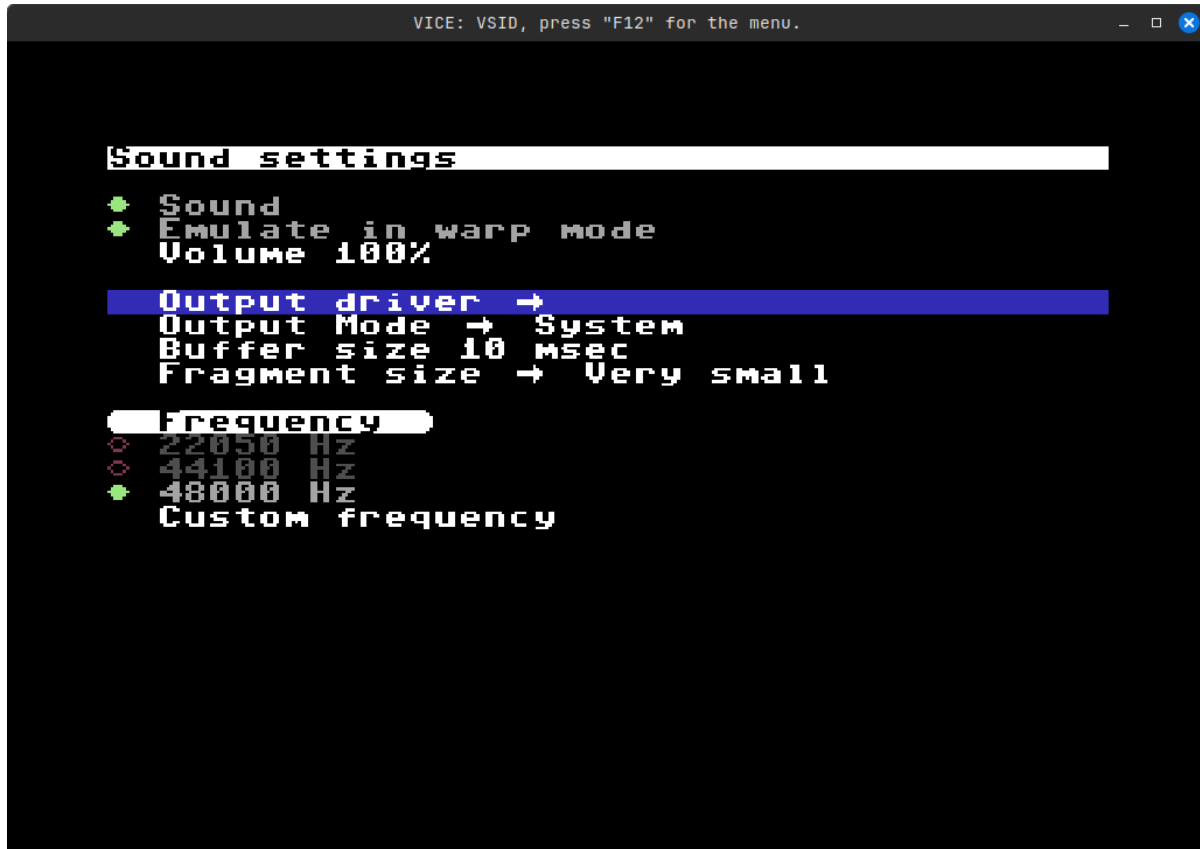
F12 to open the settings menu.

Sound configuration

The sound configuration will require tuning for the computer you use it on.

In the screenshot below you can see that for this system I use the **pulse** audio driver and have the **Buffer size** set to 10 milliseconds and **Fragment size** set to very small.

These 3 settings can make or break the audio quality.



To get to these settings follow the following settings path:

Sound settings ⇒ Output driver ⇒ Select the driver you want to use

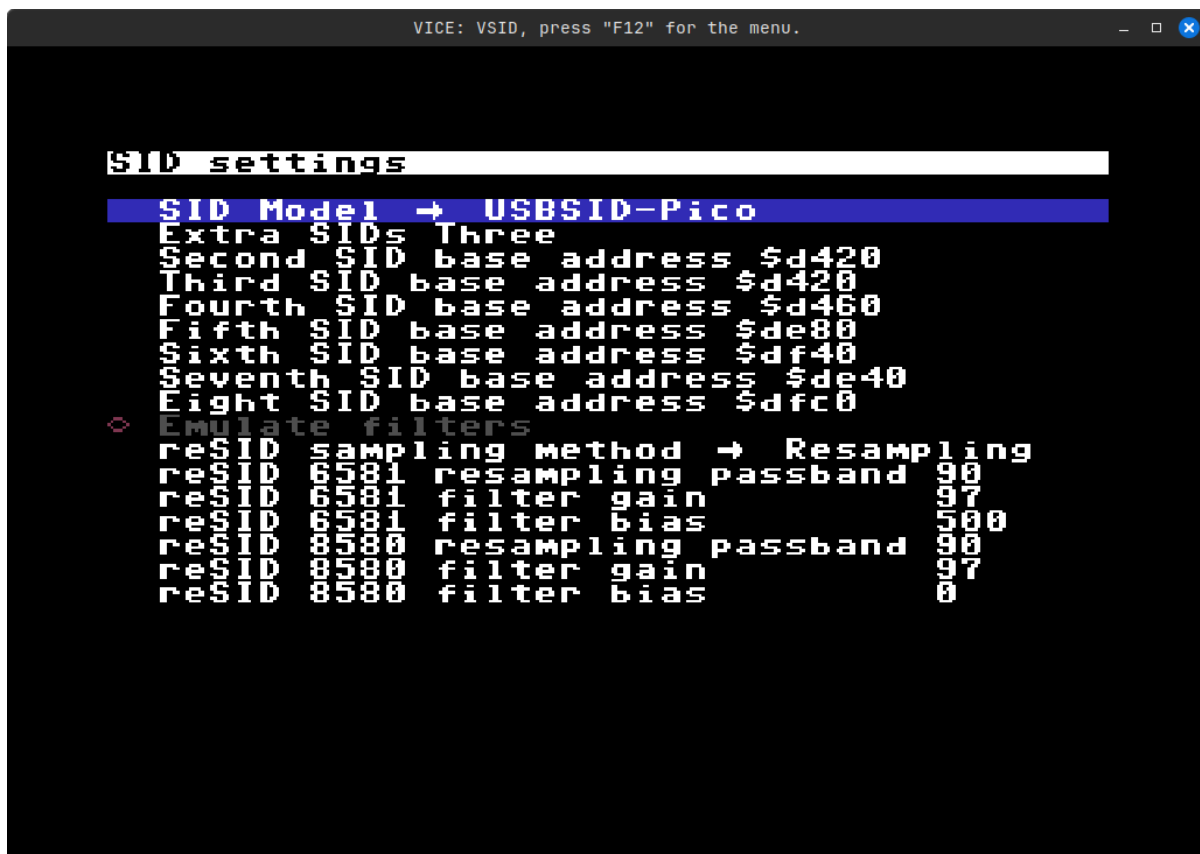
Sound settings ⇒ Buffer size ⇒ Select the milliseconds you want to use

Sound settings ⇒ Fragment size ⇒ Select the size you want to use

SID configuration

For any audio to come out of the board you *need* to select USBSID-Pico as SID engine.

If you want to use more the one SID then you need to select the amount of **Extra SIDs** and set their I/O addresses.



To get to

these settings follow the following settings path:

SID settings ⇒ SID model ⇒ USBSID-Pico

SID settings ⇒ Extra SIDs ⇒ select the amount of extra SID's and set their addresses

x64sc GTK

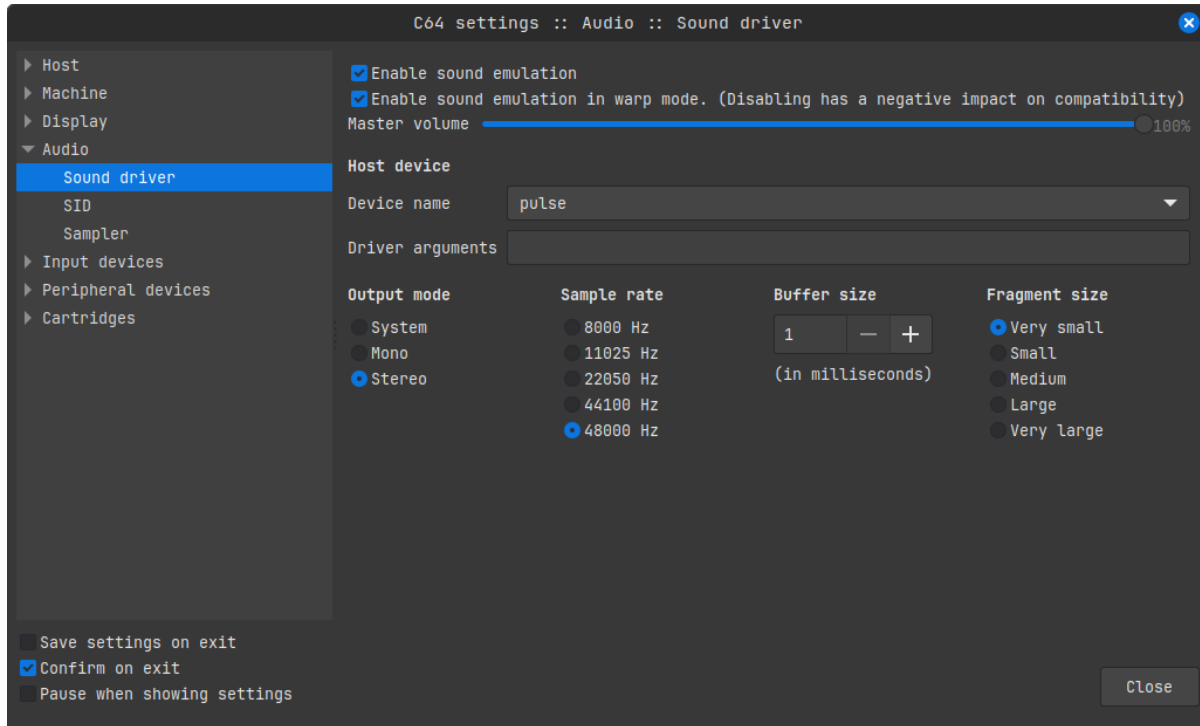
ALT+O or click Preferences/Settings to open the settings menu.

Sound configuration

The sound configuration will require tuning for the computer you use it on.

In the screenshot below you can see that for this system I use the **pulse** audio driver and have the **Buffer size** set to 1 millisecond and **Fragment size** set to very small.

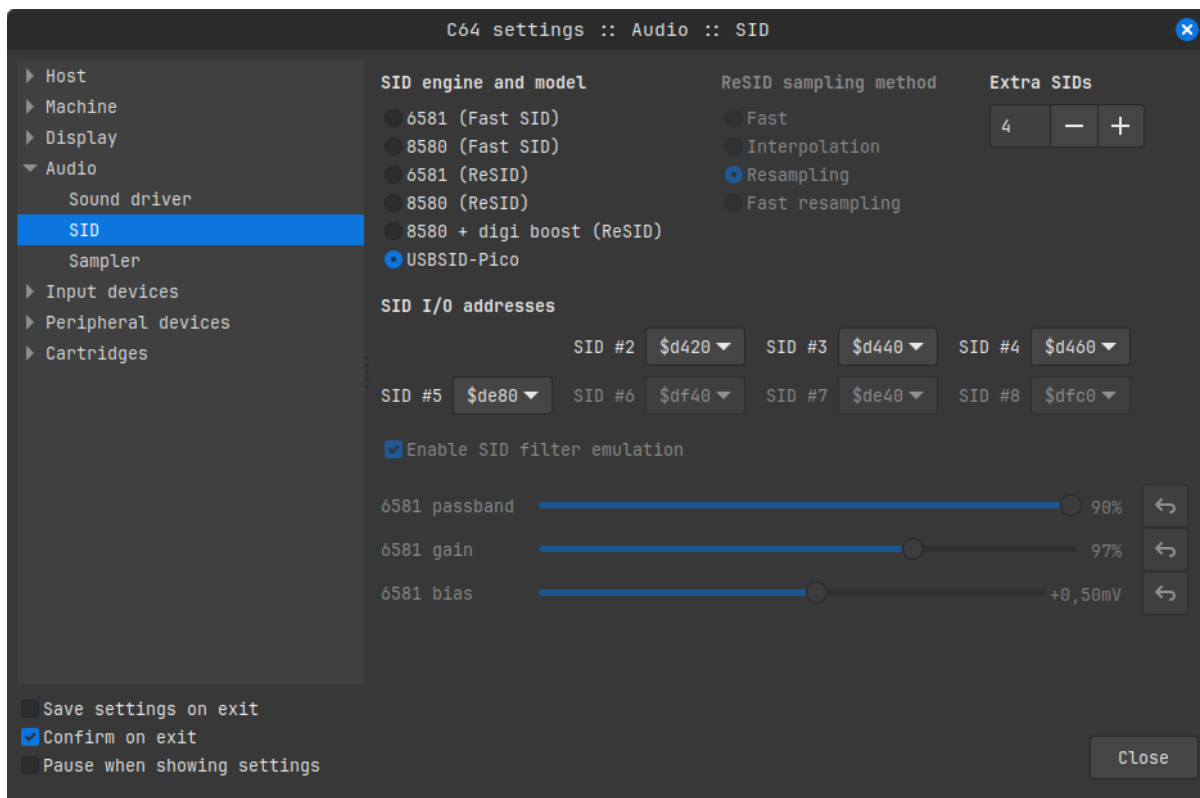
These 3 settings can make or break the audio quality.



SID configuration

For any audio to come out of the board you *need* to select USBSID-Pico as SID engine.

If you want to use more than one SID then you need to select the amount of **Extra SIDs** and set their I/O addresses.



x64sc SDL2

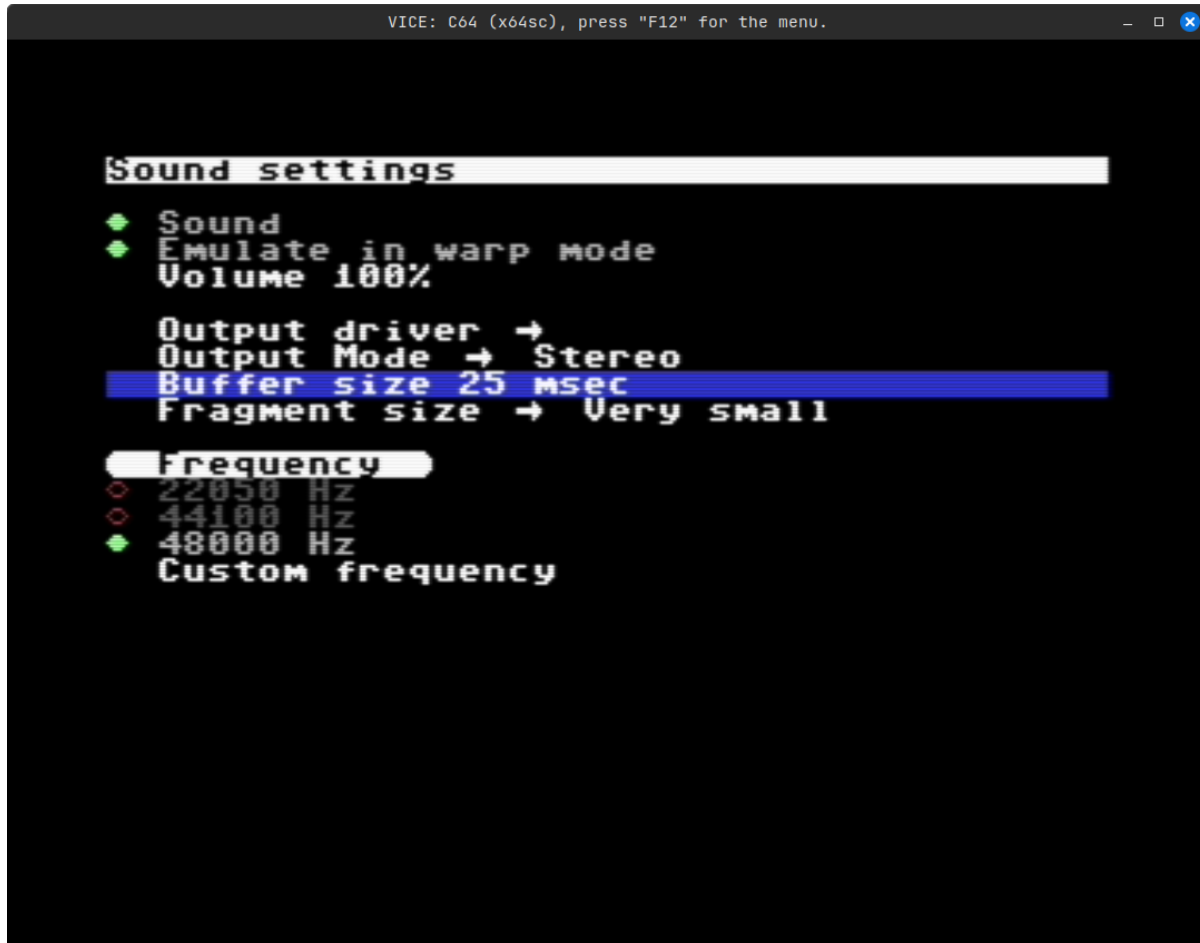
F12 to open the settings menu.

Sound configuration

The sound configuration will require tuning for the computer you use it on.

In the screenshot below you can see that for this system I use the **pulse** audio driver and have the **Buffer size** set to 25 milliseconds and **Fragment size** set to very small.

These 3 settings can make or break the audio quality.



To get to these settings follow the following settings path:

Sound settings ⇒ Output driver ⇒ Select the driver you want to use

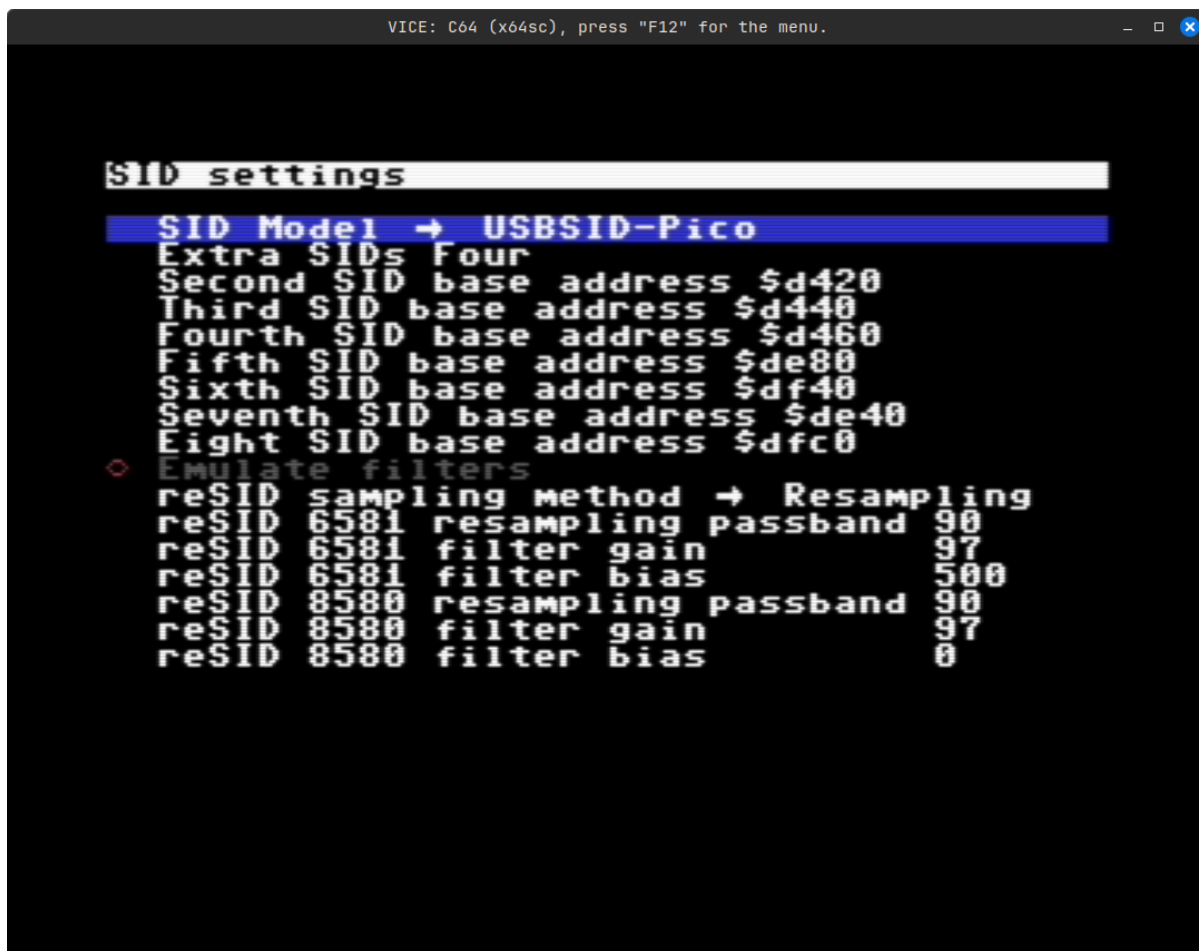
Sound settings ⇒ Buffer size ⇒ Select the milliseconds you want to use

Sound settings ⇒ Fragment size ⇒ Select the size you want to use

SID configuration

For any audio to come out of the board you *need* to select USBSID-Pico as SID engine.

If you want to use more the one SID then you need to select the amount of **Extra SIDs** and set their I/O addresses.



To get to these settings follow the following settings path:

Machine settings ⇒ Model settings ⇒ SID settings ⇒ SID model ⇒ USBSID-Pico

Machine settings ⇒ Model settings ⇒ SID settings ⇒ Extra SIDs ⇒ select the amount of extra SID's and set their addresses

Retrodebugger

t.b.d.

jsidplay2

This section covers the use of USBSID-Pico with [c64jukebox](#) website that runs on [jsidplay2](#)

c64jukebox

t.b.d.

sidplayfp

t.b.d.

Disclaimer

I do this stuff in my free time for my enjoyment. Since I like to share my joy in creating this with everyone I try my best to provide a working PCB and Firmware. I am in no way an electronics engineer and can give ***no guarantees*** that this stuff does not break or damage your hardware, computer, phone, or whatever you try to hook it up to. Be sure to take great care when inserting any real MOS SID chips into the board. While everything has been tested with real chips, this is in no way a guarantee that nothing could go wrong. Use of this board and firmware at your own risk! I am in no way responsible for your damaged hardware. That being said, have fun!

License

Software License ~ GNUv2

All code written by me in this repository is licensed under the terms of the GNU General Public License as published by the Free Software Foundation, version 2. Any code in this repository that is not written by me automatically falls under the licensing conditions by the authors of said code as mentioned in the source code header. For more information about this license please read the LICENSE document in the root of the USBSID-Pico repository.

Hardware License ~ Creative Commons

USBSID-Pico PCB by [LouD](#) is licensed under
[Creative Commons Attribution-NonCommercial-NoDerivatives 4.0 International](#)
[cc][by][nc][nd]

Author: LouD - generated on 2025-06-28