USBSID-Pico software manual

Author: LouD - generated on 2025-10-29

Table of Contents

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!

Supported platforms

In active development
Linux: Vice, JSidplay2, SidplayFp, RetroDebugger, SidBerry, USB Midi, WebUSB & ASID (in webbrowser) SID Play
Windows: Vice, JSidplay2, SidBerry, USB Midi, WebUSB & ASID (in webbrowser) SID Play
Android: USB Midi, WebUSB & ASID (in webbrowser) SID Play
Amiga: PlaySID library

Drivers

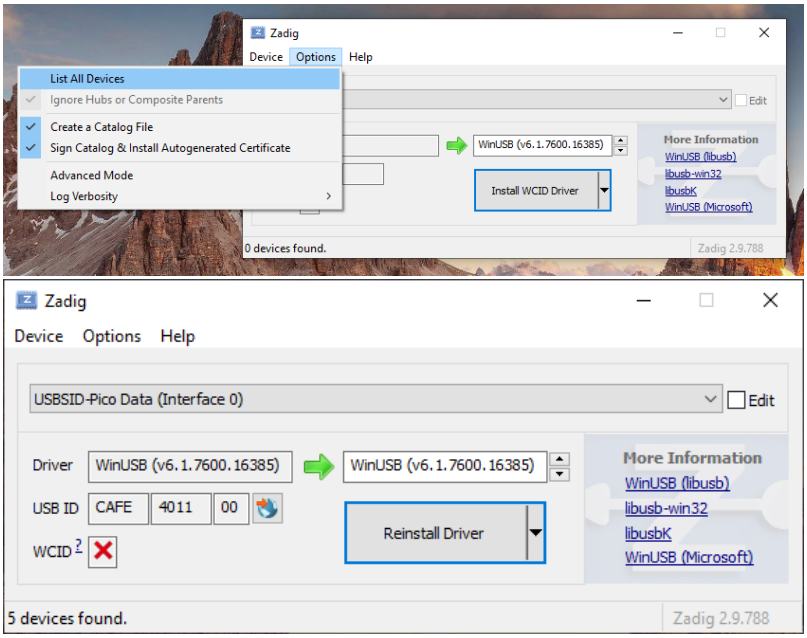
Linux Udev rules

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
```

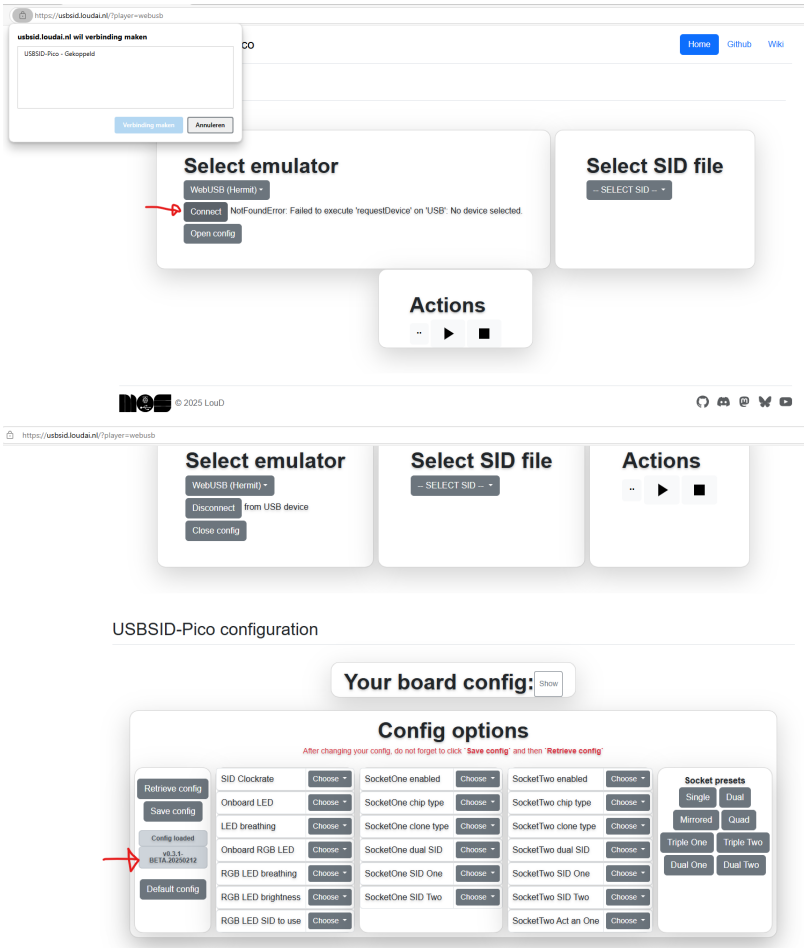
Windows driver

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 USBSID config tool website



USBSID-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`

WARNING	Switching on readmode and then switching it off again without restarting the PCB and Vice *will* result in distorted audio

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.
Below is an excerpt from the Vice manual with information about each command-line option

```
-usreadmode <0 or 1>
    Enable USBSID-Pico read mode. When enabled, this mode allows for reading from SID
registers that provides a means to configure an FPGASID that is seated on USBSID-Pico.
This mode is not suited for regular audio use as it disables cycle exact writing &
digiplay. Default is (SidUSBSIDReadMode=0) for off, (SidUSBSIDReadMode=1) for on.

-usaudiomode <0 or 1>
    Set USBSID-Pico PCB (v1.3) audio mode to Stereo or Mono. When enabled the audio of
SID socket 1 will be output over the left audio channel and the audio of SID socket 2
over the right channel of USBSID-Pico's audio out. Default is (SidUSBSIDAudioMode=0)
for Mono, (SidUSBSIDAudioMode=1) for Stereo.

-usdiffsize <number divisable by 8>
    Specifies the minimal difference in bytes to use between the head and tail of the
drivers ringbuffer, this allows for some audio quality tuning on slower systems. In
most cases the default value of 64 bytes (also the size of a USB packet) is the right
setting for best audio quality. (SidUSBSIDDiffSize=32-256).

-usbuffsize <number divisable by 8>
    Specifies the size in bytes to use as ringbuffer to store temporary data when
sending SID writes to USBSID-Pico, this allows for some audio quality tuning on slower
systems. In most cases the default value of 8192 bytes is the right setting for best
audio quality. (SidUSBSIDBufferSize=512-16384).
```

x64sc GTK

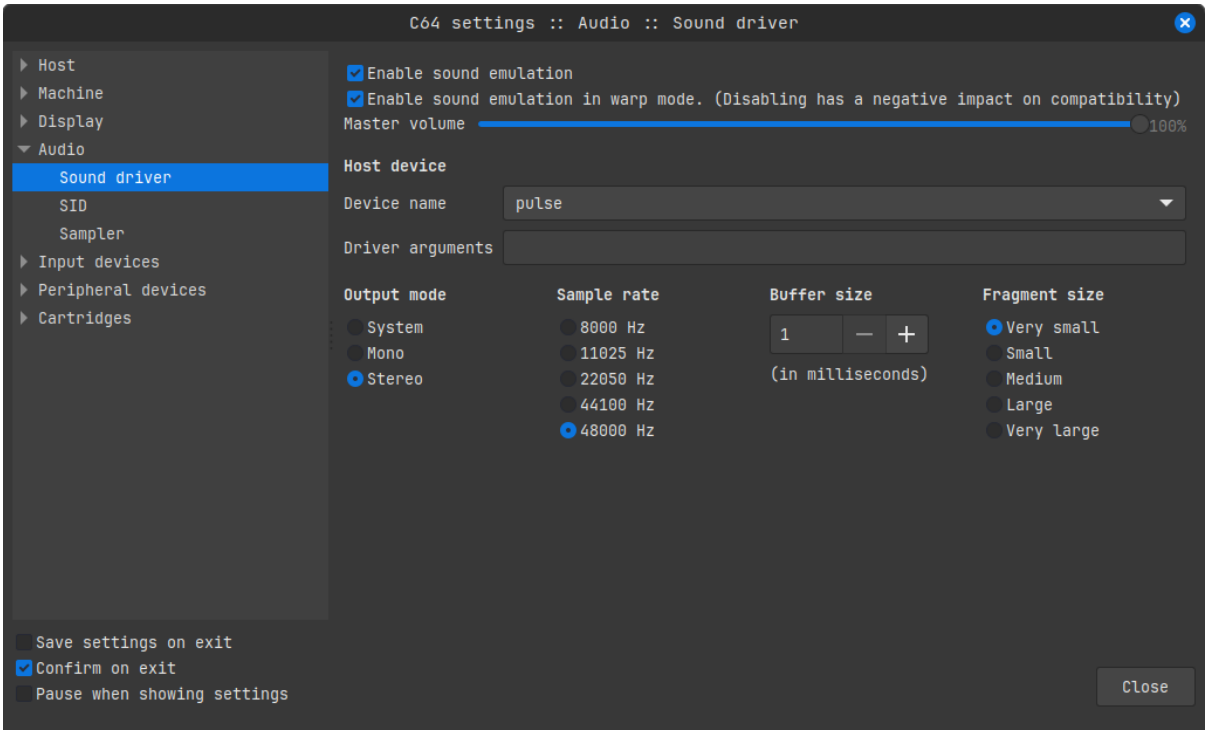
ALT+O or click Preferences/Settings to open the settings menu.

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.

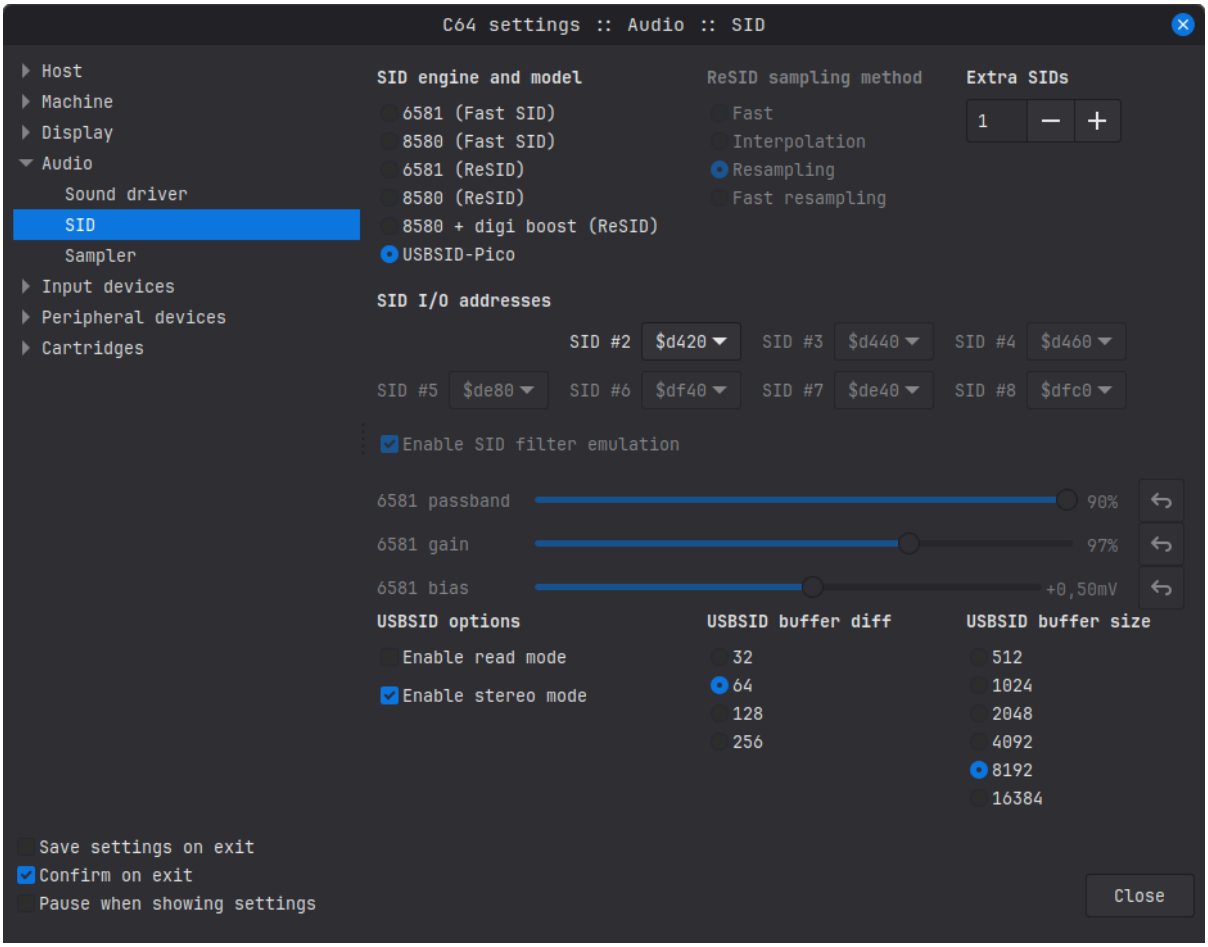
These 3 settings can make or break the audio quality.



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 the one SID then you need to select the amount of `Extra SIDs` and set their I/O addresses.


C64 settings :: Audio :: SID
Host
Machine
Display
Audio
Sound driver
SID
Sampler
Input devices
Peripheral devices
Cartridges
SID engine and model
6581 (Fast SID)
8580 (Fast SID)
6581 (ReSID)
8580 (ReSID)
8580 + digi boost (ReSID)
USBSID-Pico
ReSID sampling method
Fast
Interpolation
Resampling
Fast resampling
Extra SIDs
1
SID I/O addresses
SID #2 $d420
SID #3 $d440
SID #4 $d460
SID #5 $de80
SID #6 $df40
SID #7 $de40
SID #8 $dfc0
Enable SID filter emulation
6581 passband 90%
6581 gain 97%
6581 bias +0,50mV
USBSID options
Enable read mode
Enable stereo mode
USBSID buffer diff
32
64
128
256
USBSID buffer size
512
1024
2048
4092
8192
16384
Save settings on exit
Confirm on exit
Pause when showing settings
Close

x64sc SDL2

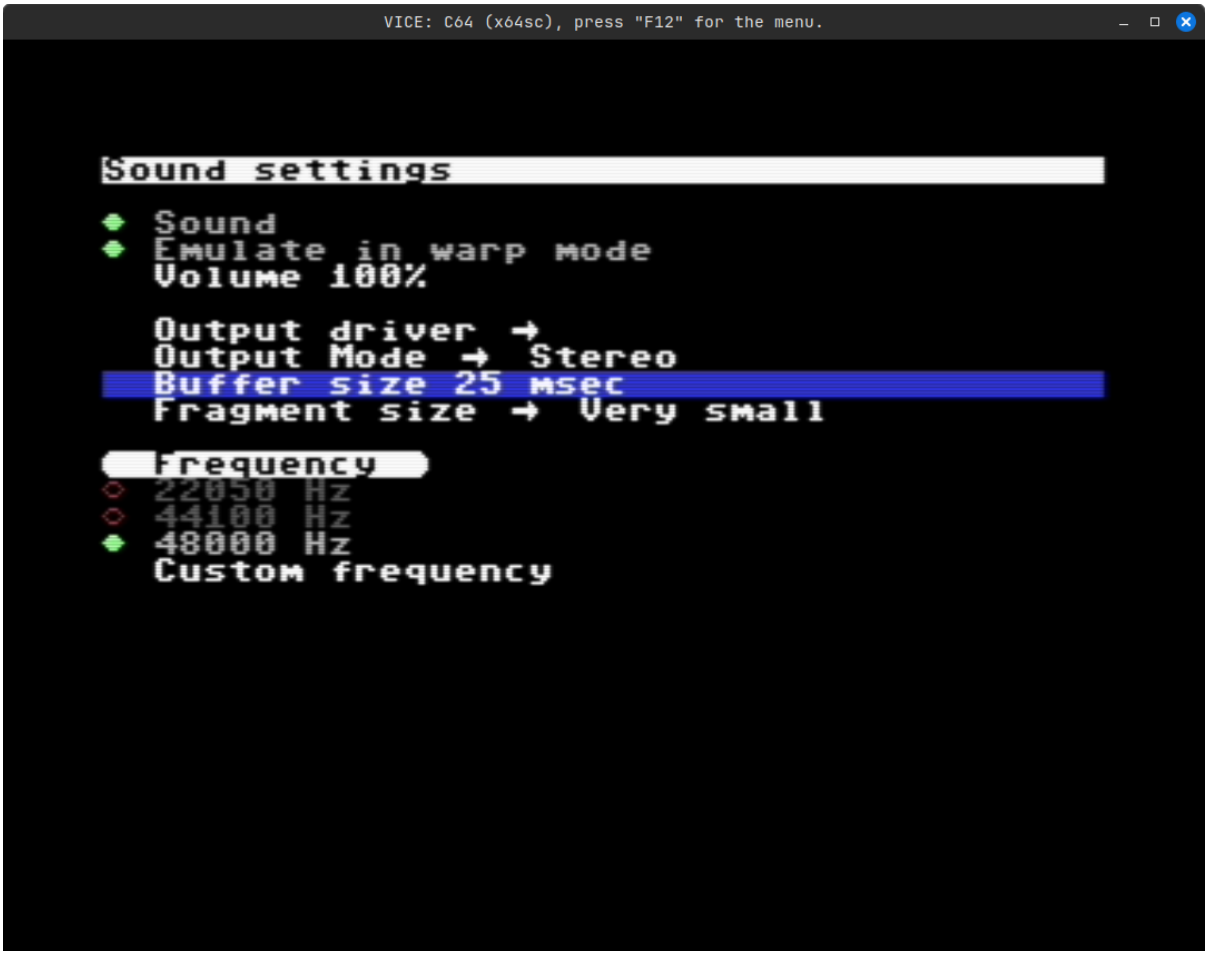
Press F12 (Linux/Windows) / F10 (MacOs) to open the settings menu.

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

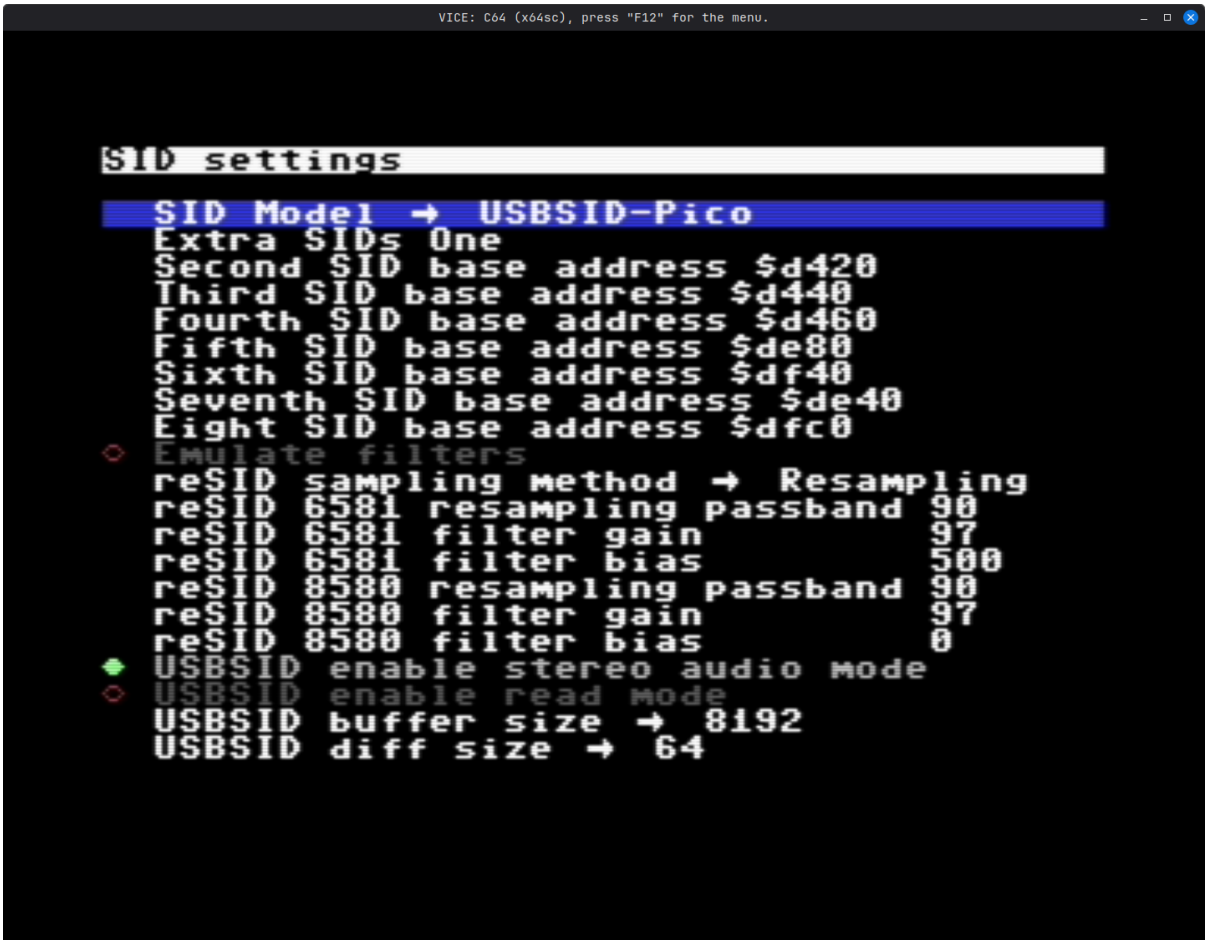
`Sound settings` ⇒ `Buffer size` ⇒ Select the milliseconds you want to use

`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

vsid GTK

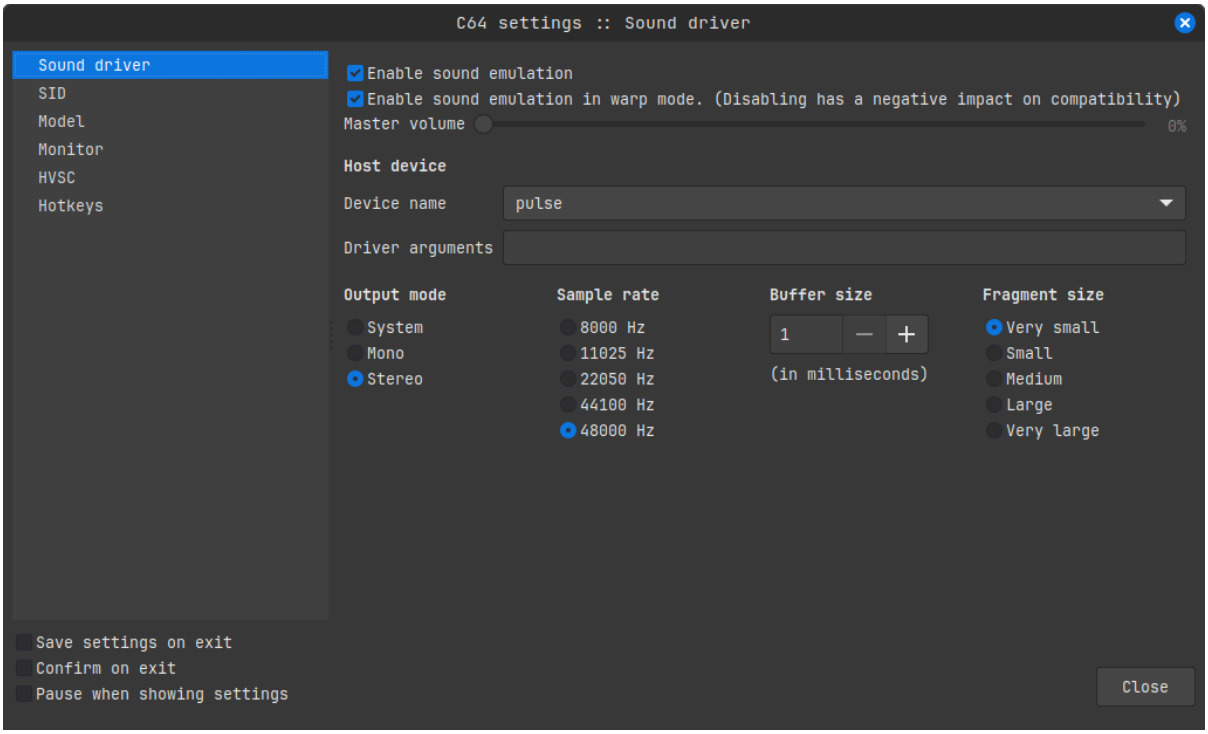
ALT+O or click Preferences/Settings to open the settings menu.

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.

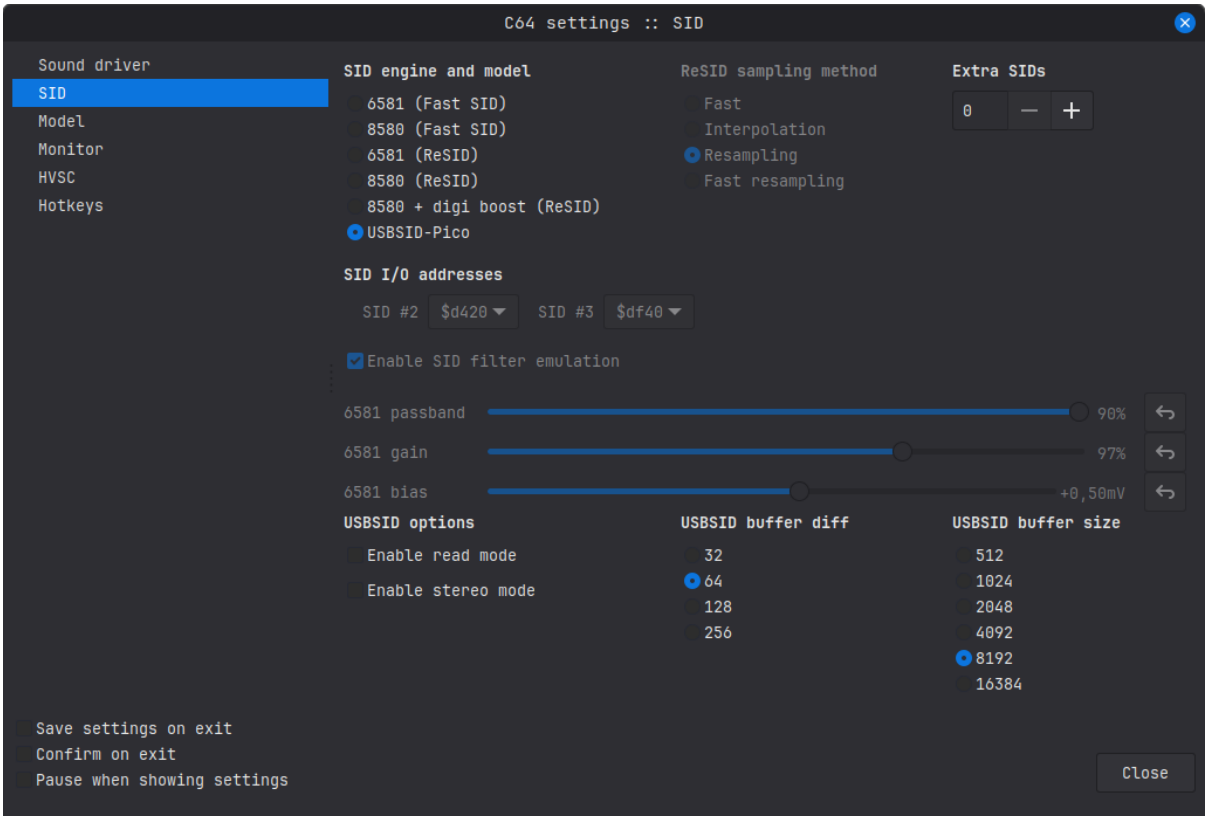
These 3 settings can make or break the audio quality.



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 the one SID then you need to select the amount of `Extra SIDs` and set their I/O addresses.


C64 settings :: SID
Sound driver
SID
Model
Monitor
HVSC
Hotkeys
SID engine and model
6581 (Fast SID)
8580 (Fast SID)
6581 (ReSID)
8580 (ReSID)
8580 + digi boost (ReSID)
USBSID-Pico
ReSID sampling method
Fast
Interpolation
Resampling
Fast resampling
Extra SIDs
0
SID I/O addresses
SID #2 $d420
SID #3 $df40
Enable SID filter emulation
6581 passband 90%
6581 gain 97%
6581 bias +0,50mV
USBSID options
Enable read mode
Enable stereo mode
USBSID buffer diff
32
64
128
256
USBSID buffer size
512
1024
2048
4092
8192
16384
Save settings on exit
Confirm on exit
Pause when showing settings
Close

vsid SDL2

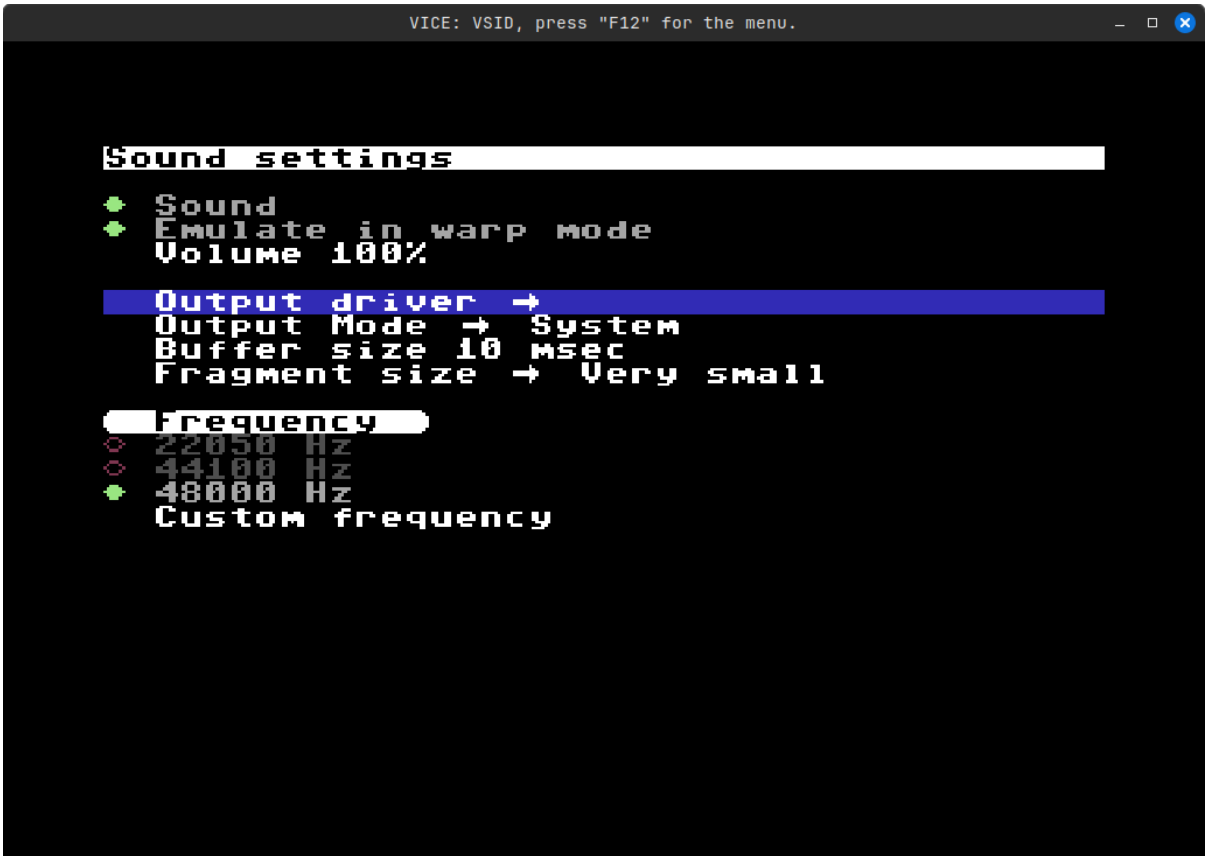
Press F12 (Linux/Windows) / F10 (MacOs) to open the settings menu.

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

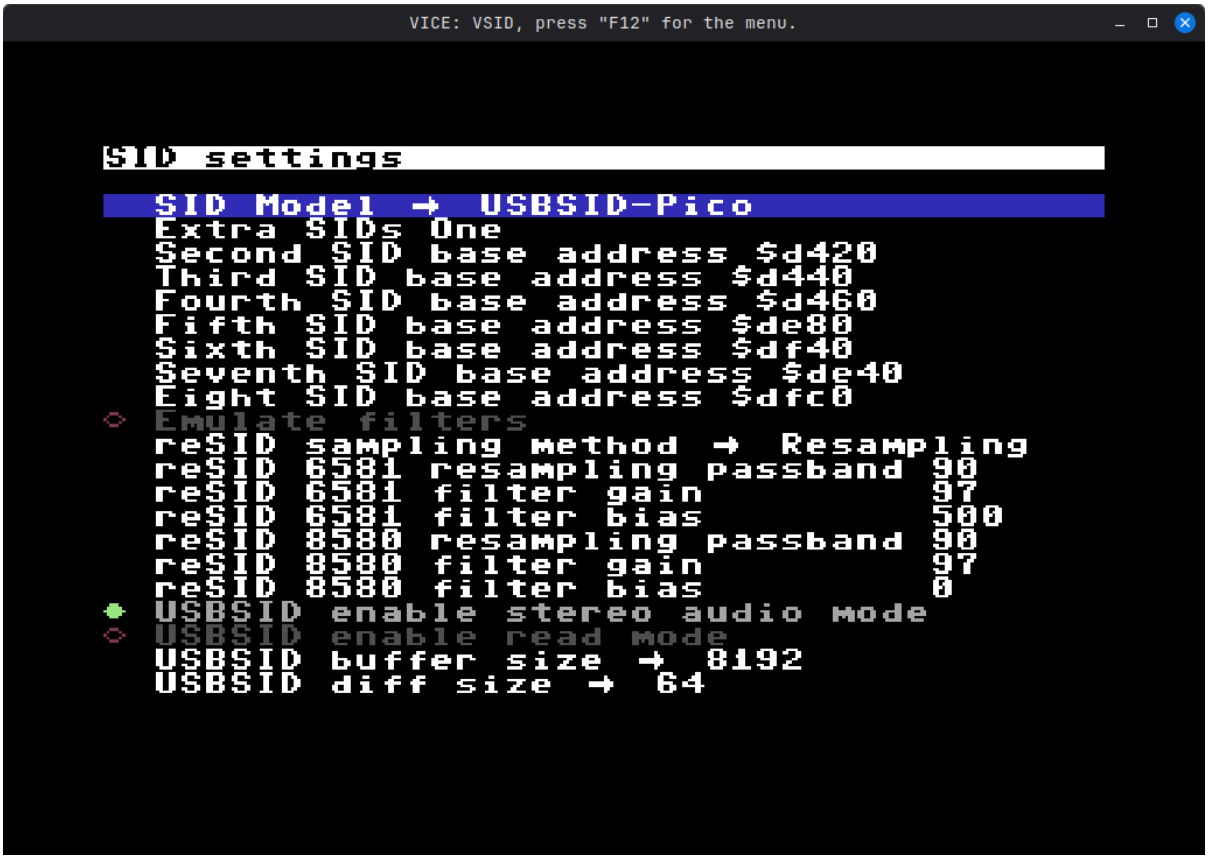
`Sound settings` ⇒ `Buffer size` ⇒ Select the milliseconds you want to use

`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`

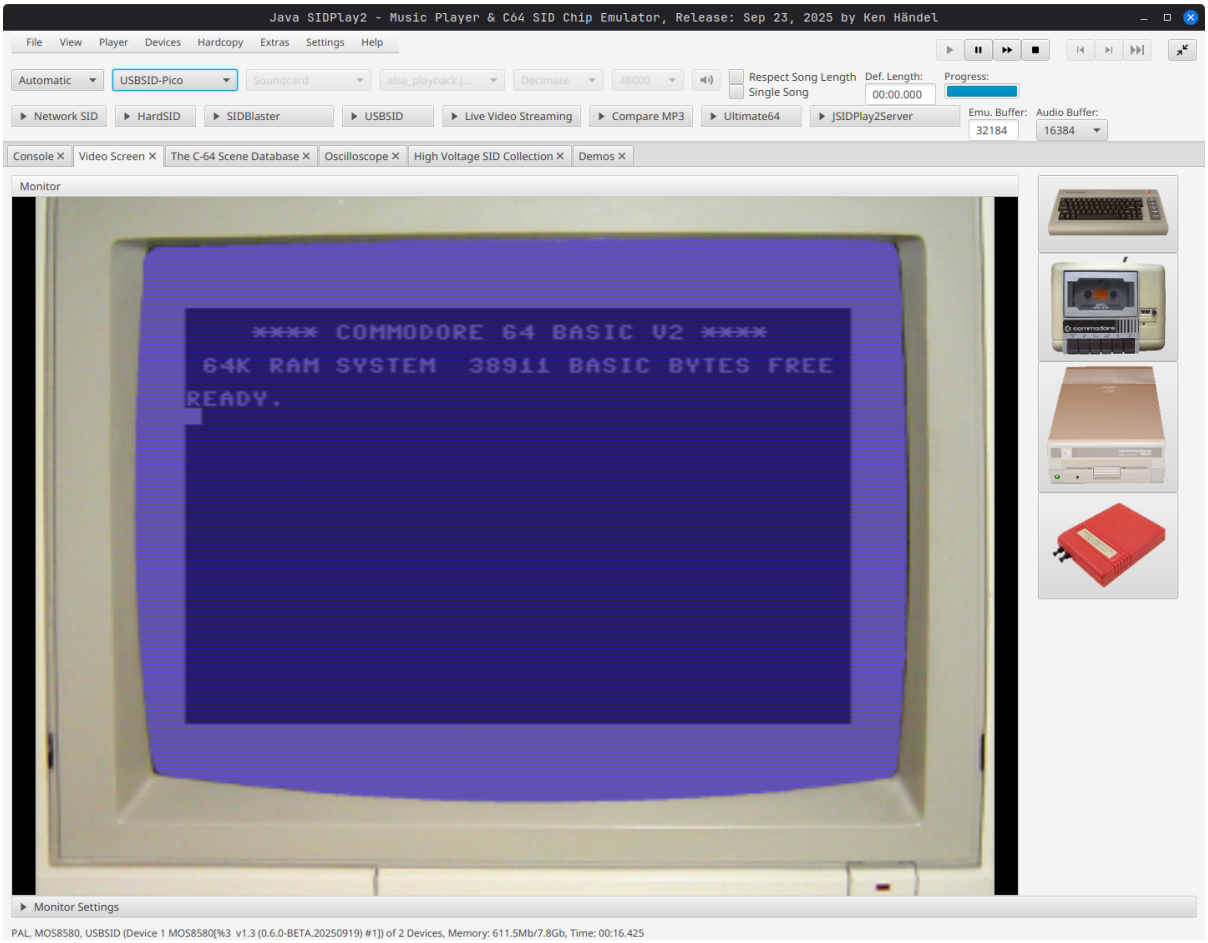
`SID settings` ⇒ `Extra SIDs` ⇒ select the amount of extra SID's and set their addresses

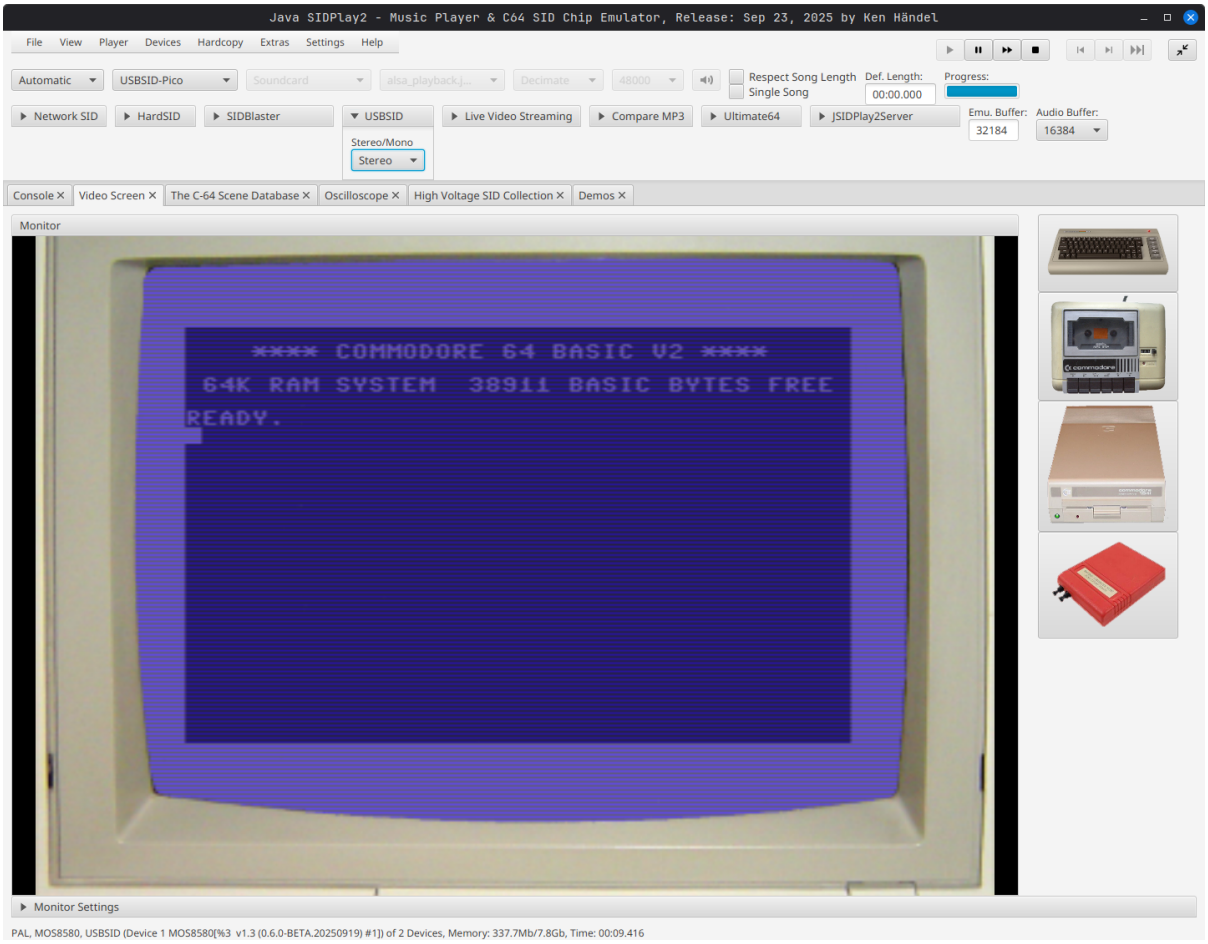
jsidplay2

This section covers the use of USBSID-Pico with `jsidplay2`

Settings

After starting JSidplay2 you can choose USBSID-Pico from the dropdown menu that says emulation and set the board to mono or stereo using the settings dropdown





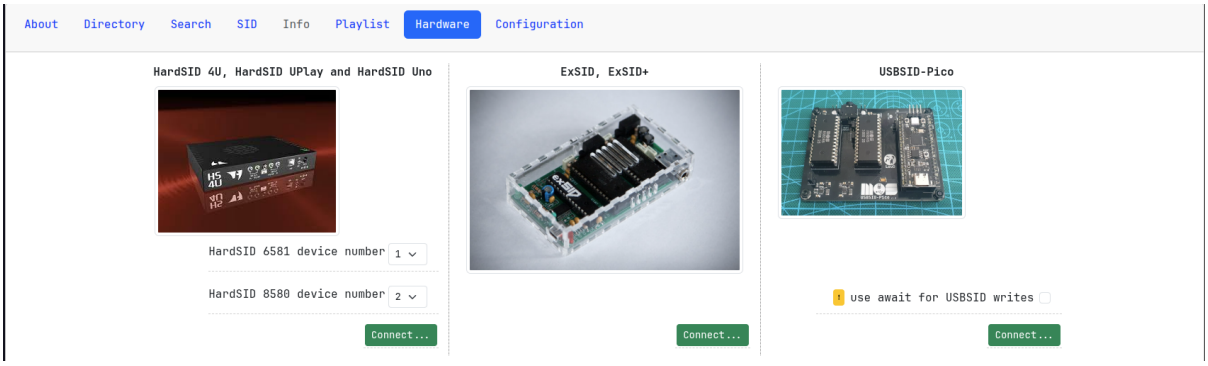
c64jukebox

`c64jukebox` website that runs on `jsidplay2` available at haendel.ddns.net:8443/static/c64jukebox.vue

To enable USBSID-Pico in a WebUSB supporting browser (like Chrome) click on the `Hardware` tab and then click `Connect` in the USBSID-Pico box. After connecting search for any SID file in the search tab and click the play button.

WARNING

Because this play method is not perfected yet there will be delays when selecting a follow up tune and pressing stop or pause. This can be circumvented enabling `use await for USBSID write` on the Hardware tab, but this will severely impact playing tunes that rely on timing

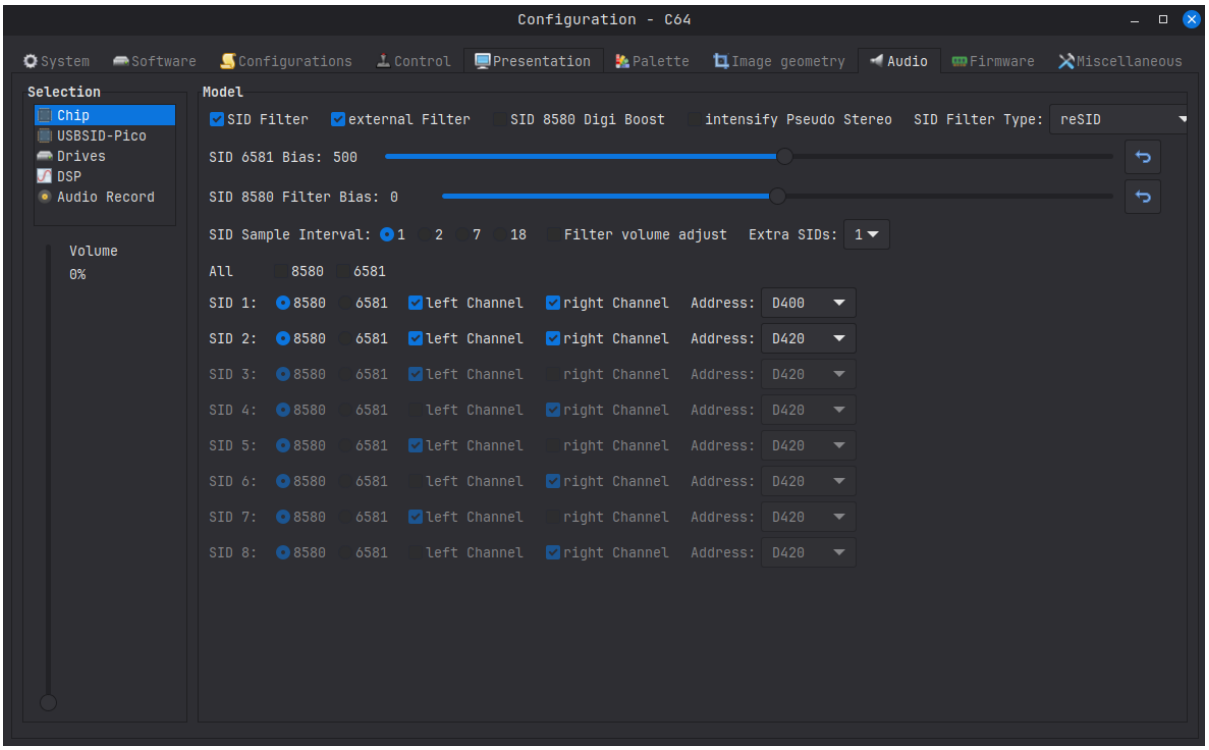


Denise

This section covers the use of USBSID-Pico with the C64 emulator of `Denise`

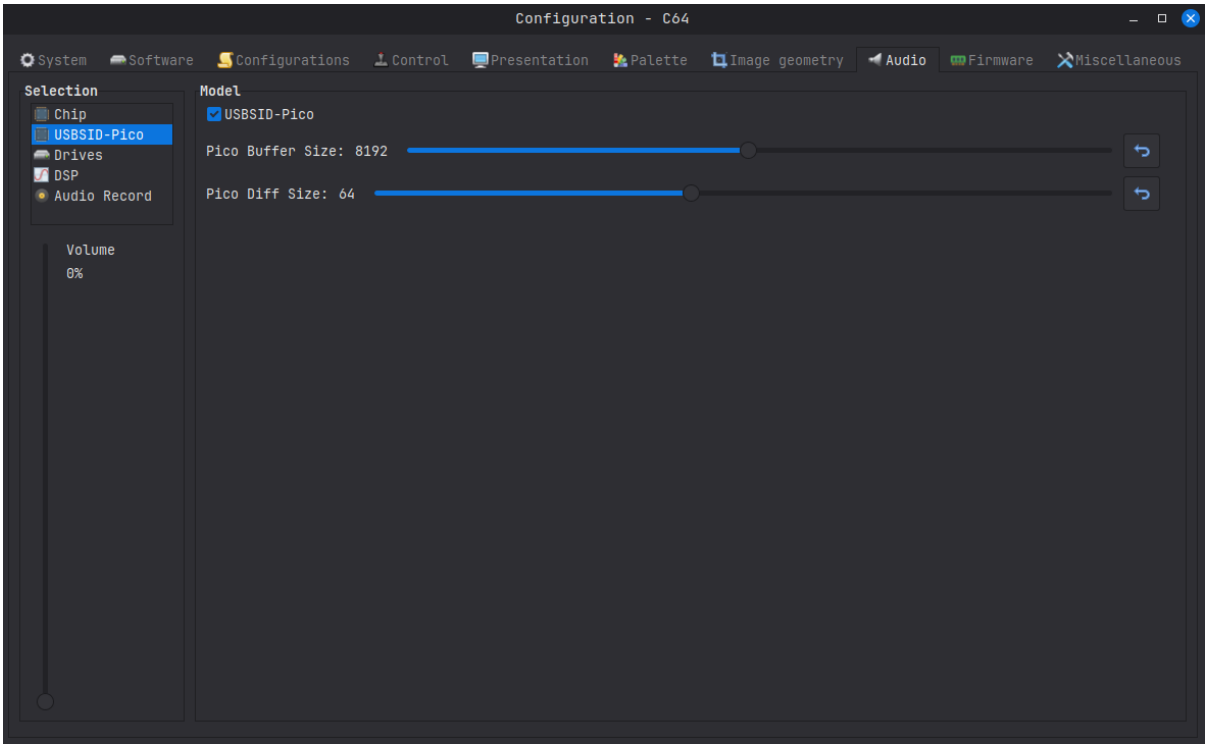
USBSID settings

Open the C64 settings menu and go to the audio tab. Under this tab on the left side you will see an `USBSID-Pico` option
Here you can enable or disable the use of USBSID-Pico. If you have any audio issue you can try to increase the buffer and diff sizes.



SID settings

Open the C64 settings menu and go to the audio tab. Under this tab on the left side you will see a `Chip` option
Here you can configure the amount of SID's you have on the board. Be sure to try the different settings for better output.


Configuration - C64
System
Software
Configurations
Control
Presentation
Palette
Image geometry
Audio
Firmware
Miscellaneous
Selection
Chip
USBSID-Pico
Drives
DSP
Audio Record
Volume
0%
Model
USBSID-Pico
Pico Buffer Size: 8192
Pico Diff Size: 64

Retrodebugger

t.b.d.

sidplayfp

t.b.d.

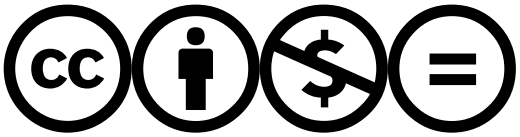
License

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 for more information about this license please read the LICENSE document in the root of the USBSID-Pico repository.

USBSID-Pico v1.3 PCB Hardware License ~ Creative Commons

USBSID-Pico v1.3 PCB © 2025 by LouD is licensed under
Creative Commons Attribution-NonCommercial-NoDerivatives 4.0 International



USBSID-Pico v1.0 PCB Hardware License ~ Creative Commons

USBSID-Pico v1.0 PCB © 2024 by LouD is licensed under
Creative Commons Attribution-ShareAlike 4.0 International

Author: LouD - generated on 2025-10-29