+++Installing a SIDKick pico

Pay attention to *correctly orient and insert* the RPi Pico and the SKpico (see backside of PCB for markings) into the SID-socket of your C64 or C128. Note that in a C128D you might need to remove one support bolt of the power supply to fit the SKpico.

You can choose to emulate a single SID only. If you want to use a second SID or FM sound you need to connect additional cables to get the signals to the SKpico as they are not available at the SID socket:

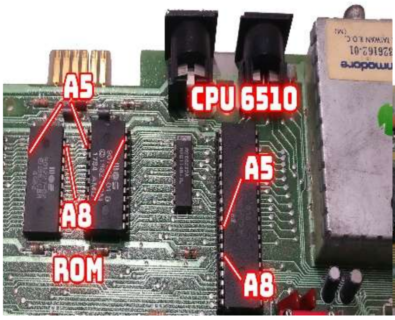
Installing additional cables in C64

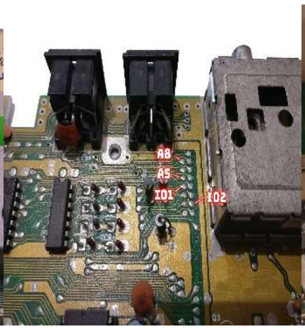
SKpico pin	C64 (see images for alternative locations)
A5/A6	CPU Pin 12 (for $d420) or 13 (required for FM)
A8/IO	CPU Pin 15 (required for $d500) OR expansion port pin 7 or 10 for $de00/$df00 addresses OR external chip select-signal (FM requires $df00)

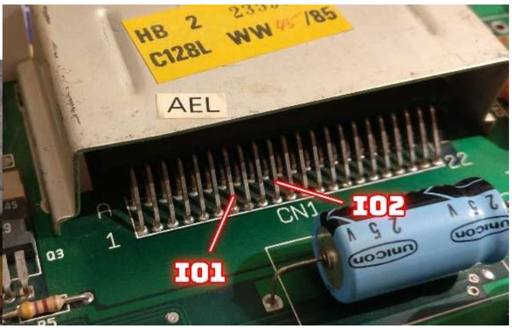
Installing additional cables in C128

SKpico pin	C128
A5/A6	CPU Pin 12 (for $d420) or 13 (required for FM)
A8/IO	$d500 is not supported for the C128 connect to expansion port pin 7 or 10 for $de00/$df00 OR external chip select-signal (FM requires $df00)

The photographs show various (other) locations where these signals can be tapped, e.g. A5 to A8 and IOx are conveniently available on some mainboards (see photo of ASSY 250469), A5/A6/A8 at the ROM-chips (not shown on the photos: on the 250469 at the kernal ROM 251913 at pin 5, 4, and 29, for example).





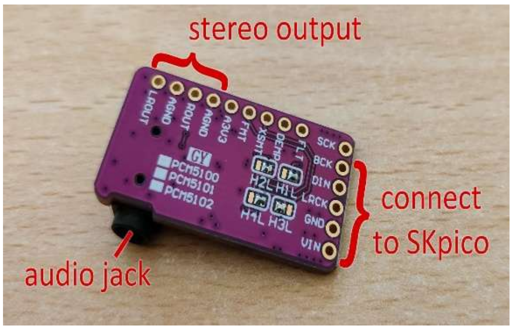


The built-in configuration tool autodetects and displays which cables have been (properly) installed and shows only possible SID-addresses.

Hint: before attaching/detaching signal cables, set the 2nd-SID/FM address to $d400 in the config tool.

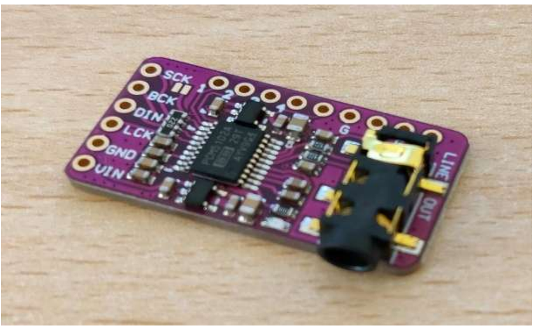
Audio Output

Interface board without DAC



The PWM sound is output via the SID-socket. If you use the optional DAC-module you can directly get the stereo audio signal from the audio jack or connectors on the PCB.

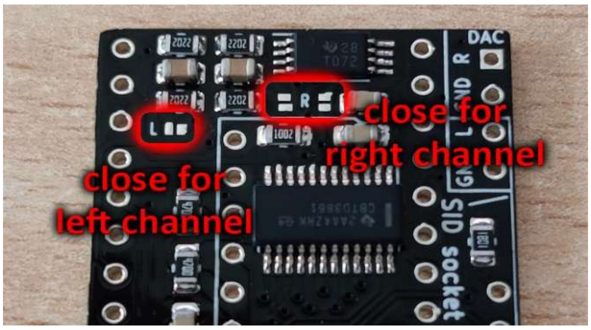
Hint: You can also connect the DAC-output to the video-audio-connector of the C64/C128. Pin 3 is the standard audio output, pin 7 is often used for the second audio channel. If you do so: use the firmware which *does not output via PWM*.



Note, which of the two output options is active depends on the firmware that you use (there are options for PWM-only and PWM+DAC simultaneously). PWM quality is slightly better when the DAC output is not enabled.

The order of pins to connect the DAC to the SKpico is easy to determine: the order is given once you match Vin which corresponds to Vcc at the SKpico and GND to GND.

Interface board with onboard DAC

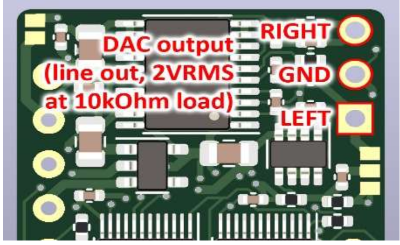


Hint: For best quality connect the DAC-output on the SKpico-PCB to the audio/video-socket (pin 3 and 7, sometimes 5 is used for the right channel) or to any other audio jack/socket.

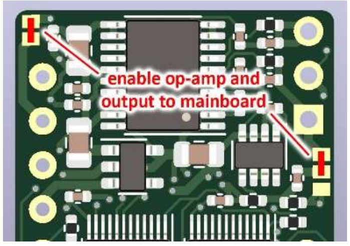
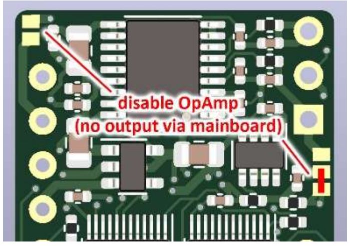
To output the sound via the SID-socket you need to close solder jumpers! The one marked with "L" outputs the left channel through pin 27 of the SID-socket (the standard audio output). Note that this way, audio goes through the circuitry on the board and quality suffers a bit. Optionally you can also close the two solder jumpers marked "R" to output the right channel to pin 26 (normally this pin is the "external in"!) -- this signal will be available on pin 5 on the audio/video socket, however, the circuitry dampens the signal significantly and this option only really makes sense on modified mainboards (e.g. connecting this pin to the video-socket).

In short: ideally connect the DAC-outputs directly to the audio/video-socket (then DO NOT close the solder jumpers), otherwise route the left channel through the mainboard by closing solder jumper "L".

SKpico2040DAC



There are two sound routing options: using the stereo-line out connector, and via the C64/C128 mainboard. Enabling line out and routing via mainboard work simultaneously (use jumper settings as for mainboard below).

output	jumpers
Routing the sound via the mainboard (set the two solder jumpers as indicated in red)	
using only the line out (for external amplifiers or connecting to the video-audio-socket) In this case, you can disable the operational amplifier as shown on the right.	

Connecting additional address lines works as for the other variants, the connectors are the two at the short side of the PCB.

Powering the SKpico

The SKpico is powered from the C64/C128-mainboard, DO NOT power from USB.

Firmware Uploading (interface board)

The SKpico-PCBs do not need to be programmed in any way. Only the RPi Pico needs to be flashed with the pre-built binaries (available in the release package).

The procedure is simple: press and hold the 'Boot'-button on the RPi pico, then connect to your PC. When it shows up as USB-drive you can simply copy the firmware (.uf2) to this drive (no need to keep pressing the boot button).

IMPORTANT: DO NOT connect the RPi Pico to USB while it is plugged into yourC64/C128! (but it's no problem if the Pico is soldered to the SKpico-PCB)**Sidekick64-/RAD Expansion Unit-InterOp**

If you're using Sidekick64 or RAD Expansion Unit then you should update to their latest firmware.

Configuration-Tool

The built-in configuration tool allows you to choose the emulated SID-types (or SID + FM), digi boost settings, volume and panning and is hopefully mostly self-explanatory.

The mouse/paddle-settings ("POT X/Y") deserve a bit of explanation: as one goal was to keep the interfacing circuitry simple, you might need to adjust some settings for the SKpico to work (best) with your mouse or paddles. Once you move the cursor to the potentiometer settings, a preview of the values and movement is shown. You can now tune the configuration:

- if a mouse moves only horizontally or not at all then choose the "level" option (by pressing 'V'). This option does not work with paddles!
- if your mouse shows some more weird jumps, try the "outlier" option (key 'O'). There are two intensities: normal and aggressive outlier removal.
- the "trigger"-option (key 'T') implements an alternative reading of mouse values (pressing 'T' multiple times selects different voltage thresholds); note, trigger does sometimes not work on Pico clones -- read "troubleshooting" below on how to fix this.
- additional filters can reduce the inherent jittering of mice/paddles on the C64/C128 further: "median" is a simple yet good outlier rejection for remaining jitter. "smooth" uses a exponential weighted average (it comes in versions for paddles and mice).

NOTE: potentiometer filtering modes do not work with two paddles/mice used simultaneously.

If you choose 'reSID+digi detect' as emulation option, then the SKpico uses heuristics to detect modern digi playing techniques (such as that used in Vicious Sid) which yield improved quality compared to the (extended) reSID 0.16 emulation. These techniques, when detected successfully, are emulated with special code paths. The heuristics are based on the findings by Jürgen Wothke used in WebSid.

To avoid bus conflicts when you use cartridges operating in the IO1/2 address spaces, make sure you do not use the IO1/2 addresses for the SKpico as well. The configuration tool tries to detect cartridges and prints a warning message.

Adding PRGs

You can add PRGs to the firmware which can then be started from the configuration tool (via F7) or directly from Basic with SYS 54333,0 (for the first PRG), SYS 54333,1 (2nd PRG) etc.

To add PRGs either use

- *skpicopatch* (it's in the release package), with the respective firmware as parameter. This adds all PRGs listed in *prg.lst* to the firmware and writes it to *SKpicoPRG.uf2*, or
- add PRGs in the configuration tool from a (disk) drive 8.