

SID-Wizard

User Manual

by Hermit (Mihaly Horvath)

Contents

1. What is SID-Wizard?.....	3
2. Features of SID-Wizard.....	3
2.1 Significant implemented features.....	3
2.2 Planned and desired features.....	4
.....	4
3. A little about SID and sound-synthesis.....	5
3.1 Principle of sound synthesis.....	5
3.2 Overview of structure and sound generation of SID.....	5
.....	6
3.3 Registers of SID.....	7
1. User interface.....	8
1.0 User interface overview.....	8
1.1 Displays.....	8
1.2 Pattern editor window.....	8
1.3 Orderlist.....	9
1.4 Instrument-editor window.....	9
1.5 Chord-table.....	10
1.6 Tempo-program table.....	10
.....	10
2. Keyboard layout and usage summary.....	11
.....	11
2.1 General purpose and globally usable keys.....	12
2.2 Pattern-editor related keys.....	13
2.4 Instrument-editor related keys.....	15
2.5 Chord-table / Tempo-table related keys.....	16
1. Instrument-parameters.....	17
1.1 Main instrument-settings.....	17
1.2 Waveform-arpeggio-detune table.....	17
.....	18
1.3 Pulsewidth-program table.....	18
1.4 Filter-program table.....	18
.....	18
2. Pattern Effects.....	19
2.1 Note-column effects.....	19
2.2 Instrument-column effects.....	19
2.3 Effect-column SMALL effects.....	19
2.4 Effect-column BIG effects.....	20
3. Orderlist-Effects.....	21
0. Overview of menu and file-browser.....	22
.....	22
1. Load/Save Tune (workfile) or Instrument.....	23
2. Initialize/Clear/Wipe Tune or Instrument.....	23
3. SID-Maker (SID-Wizard's Packer/Relocator/Exporter).....	24
.....	25
4. SWM-converter.....	25
5. Application note to the player routine (to embed in programs).....	26
Closing words.....	27

I. Introduction

1. What is SID-Wizard?

Hey folks, I'm Hermit! If you wonder what SID-Wizard is, I have a short answer: yet another tracker application for the famous Commodore 64 personal computer of the eighties. More specifically, a music editor for its sound chip called SID (Sound Interface Device). As you may know, there is still an evolving culture and technical background behind this machine - that's why demo musicians are still welcome to join the scene and compose wonderful SID tunes. SID-Wizard is an open-source project from the very 1st release, and everyone is welcome to help in further developments of my extensively commented source-code.

My aim from the beginning was to create a comprehensive native C64 tool for SID music creation, as there are so many editors around which are good at some aspects but on the other hand many of them lack very basic features. (For instance, saving/loading individual instruments, multi-speed support, detuning, jamming, keyboard-tracking, etc.)

However there are several trackers to mention which contain almost all important features, and which I got much inspiration from. My personal favourites - and possibly the most featureful, in order of appearance - are: Goattracker (cross-platform with emulated SID sound), SDI, X-SID, SID-Factory, JCH Editor, DMC, Hardtrack Composer. I made some comparison charts about these editors, and I figured out that I need to code a native C64 based editor, which should contain everything essential, and even some more inventive/innovative features. More or less I found the way to do so, and coupled the different features under common hood to achieve simplicity, ease of use despite the amount of new functions. Let's see in a nutshell in the next section what can be done with SID-Wizard....

2. Features of SID-Wizard

2.1 Significant implemented features

- *Around 60 instruments, ~100 of 256 byte long patterns and 16 of subtunes in total*, may vary in later SID-Wizard versions
- *Optimized rastertime (around \$18, below \$20) and small code size* to let more room for musical data (editor code and graphics are ~12kbyte together, player is ~2kbyte)
- *Virtually no zeropage usage*, so the music routine won't interfere with other parts of your program it's called from
- *3 x 25 rows pattern-editing perspective* to keep as much as possible musical data in screen-focus.
- *Goattracker-like pattern-editing concept and keyboard control* to ease migration from Goattracker.
- *Everything on one screen* (except menu) to ease composition workflow. Extensive use of border graphics to display data.
- *Polyphonic jamming mode with dovetailing* (each played note cycles through voice 1..3) - good for sounds with long release
- *Realtime compression of pattern-data* - zeroes don't appear in memory, therefore more memory is left for informative data (similar to XM-patterndata compression, but more effective).
- 'Jump to other subtune' command in orderlists/sequences. Tune can be written in subtune-chunks then be put together.
- *All instruments have independent tables and names* - no need for table-pointers, instruments can be loaded/saved independently
- *All instruments have own Hard-restart ADSR*, Hard-restart timing, and 1st-waveform (sexy restart) settings

- *Per-row based detuning and keyboard-tracking* in arpeggio, pulsewidth, filter instrument-tables, easier to do flexible, lively sounds
- *Up to 8x (400Hz) multi-framespeed* not affecting speed values and tempo (idea taken from X-SID) multispeed calls eat less rastertime than single-speed call
- *Advanced partial copy/paste and auto-advance amount, page-up/page-down* in patterns - fast navigation and editing
- *Independent track-tempos / funktempos / tempo-programs* and scrolling is possible (like Goattracker)
- *4 vibrato types* to simulate impression expressions of various instruments like violin/guitar or Rob Hubbard / Jeroen Tel tunes.
- *3..4 simultaneous pattern effects are possible at the same pattern row* (in instrument and even in note column there can be some effects)
- *Calculated vibrato and slide/portamento speed* to achieve evenly audible vibrato amplitudes through octaves - arpeggio and vibrato can be mixed (to a degree, depending on arpeggio speed)
- *Oscilloscope and Pulsewidth/Cutoff-frequency displayer bars* to keep track of the sound-variations and waveforms visually
- *Gate-off event triggered pointers* in instrument-tables (instrument's release phase can be controlled)
- *Basic editor settings are saved into workfile* (like framespeed, auto-advance amount, etc.) - templates are possible
- *There is concept and separate table for chords* that can be called from arpeggio-table of instruments or from pattern-FX (SDI has similar solution in place of instruments.) No need to create new instruments for each chord!
- Compression of empty places in note-column when saving tune (called '**packed rest**' in Goattracker's player source code, I call them 'packed NOPs')
- Save/export tune directly in SID format, or even in runnable PRG format (beside native, simple PRG and BIN codes)
- **Export/import to/from XM/MID** format with 'SWMconvert' to ease cooperation with x86 applications
- MIDI-keyboard support (monophonic) - dedicated easy-to-build interface-hardware called HerMIDI.

*There is a **SID-Wizard-light/medium** (and corresponding **SID-Maker-light/medium**) versions too, which lack some non-essential features like zeropage-saving, calculated vibrato, chords, pw/filter-reset-OFF, 11bit filtersweep, vibrato-types, hardrestart-types, arpeggio speed, gateoff-programs, octave-shift, transpose orderlist-FX, track/note delay, and subtune-jumping.... The **disabled features are greyed out in the editors**. (Medium version has almost everything but player-code is \$100 less than full version.) The purpose for these light/medium-versions is, of course, to **reduce player-code size and rastertime-consumption**. You can further switch particular features on/off (in source/settings.cfg file) before compilation to customize the player to your special needs. At the moment there's no option to tweak code-optimizations by users after compilation.*

2.2 Planned and desired features

- MMC64/IDE64/64HDD, etc. compatibility extension (to switch directory, etc.)
- Unused playercode-wiping optimization (Like in Goattracker does with post-compilation of its player)
- User-configurable keyboard layout (now only configurable in source-code before compilation)
- Mouse controller hardware/software extensions
- **Configurable colour scheme** (like in DMC) (now configurable in source-code before compilation)
- Undo/Redo function
- ...Whatever comes out as a feature-request. I count on your feedbacks or contribution...

3. A little about SID and sound-synthesis

This part is intended for readers of the manual who are new to sound-synthesis. Others can jump this whole part over.

3.1 Principle of sound synthesis

There are basically two types of analog frequency synthesis: Additive and Subtractive. Additive synthesis tries to simulate the sound of instruments by adding a limited number of harmonics to the base frequency of a musical note. Subtractive method is based upon oscillators which can generate some kinds of waveforms (sinus, triangle, sawtooth, pulse/square, noise, etc.) with inherently rich spectral harmonic content and they can be mixed and filtered to be shaped to a desired timbre.

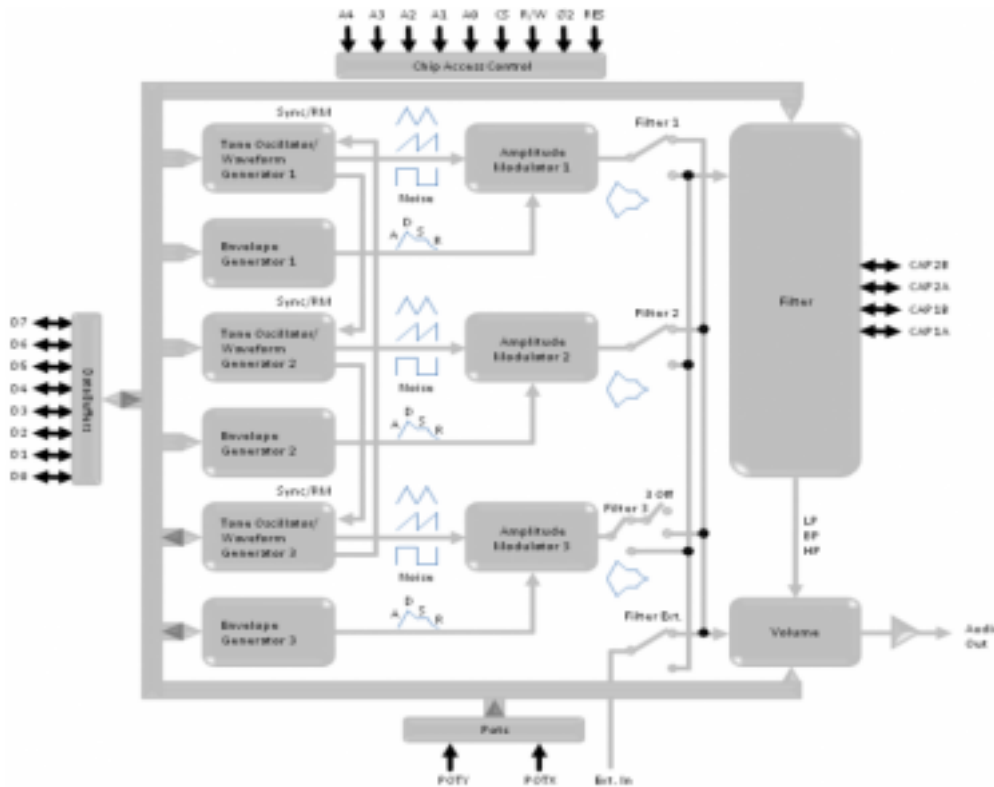
We can alter many other parameters of the sound, the most important is to control its volume-envelope in order to be able to model live instruments to a degree. Possibilities are endless compared to sample-based synthesis, only the constraints of hardware and software limit the number of variations a little bit. I'm not going to tell more about the sound synthesis at this point, many readings can be found over the cyberspace.

3.2 Overview of structure and sound generation of SID

The MOS technology SID 6581 or 8580 sound chip in C64 is a subtractive sound-synthesis chip designed by Bob Yannes in 1982. It is one of the most important reasons of the C64's popularity. There are many differences between different revisions and types of SID chips produced over the years, maybe due to the shortened development time, and the specification of the chip was somewhat unfinished.

The most significant change/improvement has appeared in the 8580 new SID model over the original 6581 SID, but many consider the 6581 better. 8580 has more balanced filter-cutoff frequency control curve and less-distorted resonance, better mixed-waveform support, and maybe more predictable ADSR envelope-handling. It's still a mystery in some aspects what the chip exactly does inside, however there were thorough reverse-engineering projects recently both in software and hardware emulation areas (e.g Resid dll, 1541 Ultimate VHDL code for SID). Even Bob Yannes himself doesn't keep tracking of the long lifetime of SID and what people make for it.

Let's see a block-diagram of SID on next page for better understanding of its structure and behaviour:



The chip has only 3 channels' polyphony (btw. Yannes's initial design ideas were far more advanced, if he had more time & chip area, there could possibly be as much as 32 channels). We have to deal with 3 voices, but throughout the years I learned (after coding my 9 polyphony 3SID tracker) that it is quite enough for a clever musician. What's more, these limits has an effect on a composer: we have to do something really cool in order to keep the listener's interest. (That's why we love remixing and performing SID tunes with my SIDrip Alliance bandmates.)

There's one limit what I personally consider a real limiting factor: there's only one filter to shape the plain waveforms. Luckily this filter can be switched on/off for all tracks individually. Designing the filter usage needs careful planning when composing music, especially if a music editor doesn't track filter utilization of the instruments automatically. (In SID-Wizard the filter is handled automatically on tracks, only tracks with filtered instruments are filtered, and normally the latest filtered instrument controls the filter's cutoff-frequency).

One advantage however is that each channels' pulse waveform's duty cycle can be changed smoothly. That feature can be utilized to create cool effects by sweeping it with slow/fast pace. (This feature is missing in many VSTi and Synthesizer modules.)

The SID is not a perfect device when it comes to timing. ADSR envelope sometimes stucks and fails to deliver exact volume-envelope, or to generate the sound at all. There's a workaround to this that's called hard-restart. It can stabilize SID's response to fast paced ADSR note-triggering sequences by resetting some registers 1-2 frames (20..40ms) before a sound actually starts (gate-on).

I don't want to repeat the whole specification chart of SID 6581, so if further/deeper info is needed, don't be afraid checking it out online. A good information summary is:

http://en.wikipedia.org/wiki/MOS_Technology_SID

and <http://noname.c64.org/csdb/release/?id=57722>

For the sake of a quasi-completeness, in the next subsection (next page) we'll collect the controllable options (registers) of SID. Most of the time it's needed for coders, but the knowledge can be handy for musicians too sometimes to tweak things a bit more consciously in the editor.

3.3 Registers of SID

The SID chip can be controlled through its registers, there are virtually 32 of them, most of them are write-only, some of them are read-only, some are not existing physically. (The model how the computer program controls SID is called 'memory-mapped' I/O.) The chip by default is routed to \$D400 (54272) memory area by a PLA in C64, therefore the registers can be written by storing bytes to \$D400..\$D41F area from music routine. (Additional SID chips can be routed inside \$D420..\$D800 area by external chip-selector circuitry, though it's not a standard and widespread solution.)

Register-address	R/W	Function	Remarks
\$D400, \$D407, \$D40E	W	Channel 1..3 note-pitch frequency low-byte	No frequency-table built into SID, pitches are controlled by code
\$D401, \$D408, \$D40F	W	Channel 1..3 note-pitch frequency high-byte	
\$D402, \$D409, \$D410	W	Channel 1..3 square-waveform pulse-width low-byte	Most of the times these spice the pulse-waveform to sound fine.
\$D403, \$D40A, \$D411	W	Channel 1..3 square-waveform pulsewidth high-nibble	
\$D404, \$D40B, \$D412	W	Channel 1..3 Waveform and sound-control nibbles	There are triangle/sawtooth/square/noise and combined waveforms.
\$D405, \$D40C, \$D413	W	Channel 1..3 envelope-generator Attack/Decay nibbles	
\$D406, \$D40D, \$D414	W	Channel 1..3 envelope-generator Sustain/Release nibbles	
\$D415	W	Filter cutoff-frequency least significant 3 bits	
\$D416	W	Filter cutoff-frequency most significant byte (8 bits)	Unfortunately this control is common for all channels.
\$D417	W	Filter resonance-amount and channel-filter-switch nibbles	Resonance is common but filter-switches are present for each channel.
\$D418	W	Filter band-mode and Main Volume nibbles	High-pass/Band-pass/Low-pass filter modes, can be combined.
\$D419	R	Paddle Potmeter X value	
\$D41A	R	Paddle Potmeter Y value	
\$D41B	R	Channel 3 oscillator output	Geir Tjelta has an echo generator routine that utilizes this feature afaik.
\$D41C	R	Channel 3 envelope-generator output	Used (beside \$D41B) in SID-Wizard to display the Oscilloscope.
\$D41D	-	not used	
\$D41E	-	not used	
\$D41F	-	not used	

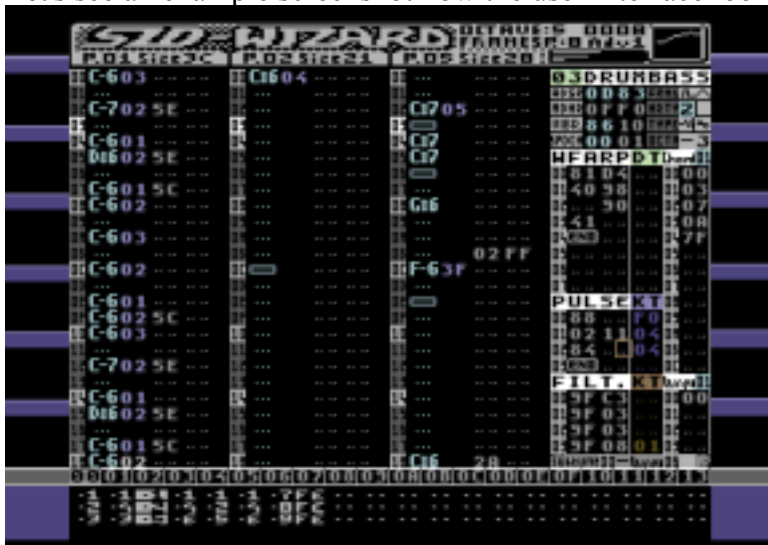
II. Usage of SID-Wizard

1. User interface

1.0 User interface overview

As it is common to trackers, this tool is based on character-matrix mode with some tiny graphical parts. That means you cannot use mouse or sliders/knobs but you have to input almost every data by hexadecimal numbers (0..9, A..F), and navigating between 'windows'/'panels' using your keyboard.

Let's see an example screenshot how the user interface looks like:



We'll be a bit redundant by telling the keyboard shortcut a bit more verbosely in this section, than later in some tables the keyboard controls will be summarized again.

1.1 Displays

At the very top in the upper border there is an informative part of the interface. Here you can see from left to right the logo, the number and size of selected patterns, framespeed, jamming octave, autoadvance-mode/amount, playback-time, pulsewidths' / filter-frequency's meters, oscilloscope, etc.

1.2 Pattern editor window

The largest part of the screen is occupied by the pattern-editor window, where you can see and edit patterns corresponding to the 3 tracks/channels that the SID routine can play. Each track has 4 columns. First column is for the notes/pitch control and note-on/note-off (plus several other) effects. Second column's main function is to select instruments or 'legato', or to contain some small pattern effects either. The 3rd and 4th columns are explicitly designed for pattern-effects, but the 4th column is only needed when we use the 'BIG' effects, and in that case it contains effect-value.

To go to pattern-editor window you can press F5 key anytime. You can easily navigate between the three tracks by cursor keys and 'Control' keys (tab key in Vice emulator), and the other obvious keys like Slash/Shift+Slash, Page-Up/Page-Down, Home, etc... If cursor is

around an instrument number and you press Return, you get to the corresponding instrument's editor panel. The length/size of the patterns can freely be increased/decreased by pressing Insert/Delete keys in the last row of the patterns (showing the '-end-' signal).

If you change the 'auto-advance' amount with Shift+A (increase) or Shift+Z (decrease), the cursor jumps the specified amount of rows after you type a note in note-column.

If you toggle the 'binding mode' by pressing Shift+B, you can scroll each track/pattern independently or otherwise you can move/scroll them together (as long as they're long enough to be scrolled).

You may check out the 'keyboard layout' section for complete summary of keys in pattern-editor window.

1.3 Orderlist

The very bottom of the screen in the lower border introduces the Orderlist window. This is where you can control the whole arrangement of the tune by putting consequent pattern numbers after each other. The currently played Orderlist positions of the song appear in inverse. It helps you to follow where you are in the timeline of your music.

To get to Orderlist, press F6 (Shift+F5) key anytime. The navigation keys are obvious here, individual track-sequence lengths can be increased/decreased by pressing Insert/Delete keys on the last 'FF' or 'FE' values. Pressing 'Return'/'Enter' over a pattern-number in Orderlist will lead you back to pattern-editor window and selects the pattern for corresponding track. Pressing Shift+Return will change all the 3 patterns corresponding to the current Orderlist cursor-position. Pressing C+=Return selects the patterns that are played at the moment (useful with different track-speeds or lengths).

Value 'FE' means the end of playback for the given track, 'FF' jumps back to a position in pattern-sequence given by the number following it. (As value 00 is reserved for 'no process', you can't use pattern 00, the lowest pattern-value is '01'.) Be aware that these are hexadecimal numbers, so for example, after '09' not '10' but '0A' comes.

*There are some effects that can be controlled from Orderlist like transposing the musical key or changing volume/tempo...

1.4 Instrument-editor window

The right quarter of the screen is the instrument-editor panel. All the parameters of the selected instrument can be seen here. You can select an instrument by (Shift)+Plus/Minus keys. At the top the instrument's number and name can be seen and edited, below it the main parameters can be edited either by hexa-keys (0..9..F) or by pressing Return on some switch-type parameters.

To get into instrument panel, press F7 key anytime. The adjustable parameters include the selected instrument's default ADSR envelope, Hard-restart ADSR and HardRestart-type settings, Vibrato-type/-frequency/-amplitude/-delay, default chord and default arpeggio-table speed, and octave-transpose amount. (If the arp./chord-speed value is > \$40, pulsewidth-table can be multispeed, if >\$80, filtertable can be multispeed for the instrument as well...)

The 3 tables below are: Waveform-arpeggio table (which contains a detuning-table too to create fancy choir effects), the Pulsewidth-program table to control duty-cycle/pulsewidth of square waveforms over time, and the Filter-program table to assign filter to the selected instrument. You can cycle between the main instrument settings and these 3 tables with Control key (TAB in Vice emulator).

In the instrument editor window you can navigate with the usual navigation keys in a standard manner. To edit the name of the selected instrument, press Shift+N key-combination and enter the name, then press Return to store, or press Escape/Stop key to cancel.

1.5 Chord-table

To the right of instrument tables the thin table contains the selected chord's pitch-list. Chords are essentially arpeggios in SID-Wizard, but they don't depend on instruments and can be called from/for any of the instruments. The main point of using chords is to reduce the number/redundancy of creating more instruments of the same type just because we want them to sound as different chords on a channel.

Press C= + F5 anytime to get to Chordtable. The usage is very simple, you can select a chord with +/- keys here, or Shift+K/L keys anywhere. In the column you type the (hexadecimal 2's complement) relative pitches counted in halftones in the order you want them to roll over time. Insert/Delete keys over '7F' or '7E' can expand/shrink size/complexity of the chord. '7F' means looping the chord over, '7E' means returning to the arpeggio table the chord was called from and continuing waveform-arpeggio-table execution with its next row.

1.6 Tempo-program table

The table in the bottom-right corner shows the selected tempo-program. For extreme pattern timing/swing/funktempo you can describe complex tempo-programs here. If the tempo-program is selected later from pattern-FX, all of its rows correspond to the timing of consequent rows of the pattern the tempo-program is applied to. Usually we don't need complex tempo-programs, just funktempo, but it might be useful for frequently appearing special rhythms...

Press C+=F7 (Control+F7) anytime to get to Tempo-program table. To select a tempo-program, you can press Shift+T/Y anytime or +/- in tempo-program table. Editing is similar to that of chord-table, except there are no numbers allowed more than \$80 (hexa) value. The tempo-program automatically rolls over after the last value, there's no need for special jump-commands.

2. Keyboard layout and usage summary

Commodore 64 US keyboard layout



US international keyboard layout:



VICE emulator C64 mapped (US positional) keyboard layout:



2.1 General purpose and globally usable keys

Keys on C64	US int. Keys in VICE emulator	Category	Functions of the pressed keys
F5	F5	Navigation	Go to Pattern-editor window.
Shift+F5	F6	Navigation	Go to Orderlist.
F7	F7	Navigation	Go to Instrument-editor panel.
C= + F5	Control + F5 or F6	Navigation	Go to Chord-table.
C= + F7	Control+F7	Navigation	Go to Tempoprogram-table.
Shift+F7	F8	Navigation	Go to disk menu.
CURSOR-down/right and Shift	Cursor up/down/left/right	Navigation	Move the cursor step-by-step up/down/left/right where possible.
CONTROL or Shift +CONTROL	TAB or Shift+TAB	Navigation	Cycle through tracks or instrument-tables.
/ or Shift+/, UP-arrow* or RESTORE	/ or Shift+ / , Page-Down*/Page-Up	Navigation	Move cursor in 4 or 8 steps up/down/left/right depending on window.
HOME	Home	Navigation	Move cursor to start-position on screen, then absolute start-position in table/pattern.
RETURN	Enter	Navigation	Car return to beginning of row, or go to pattern (s) or instrument under cursor.
F1 or C= + F1	F1 or Control+F1	Playback	Play the tune from start / follow-play tune from start.
Shift+F1 or C= + Shift + F1	F2 or Control+F2	Playback	Play the tune from marker position(s) / follow-play tune from marker position(s).
F3 or C= + F3	F3 or Control+F3	Playback	Play the selected patterns / follow-play the selected patterns.
Shift + F3 or STOP	F4 or ESC	Playback	Toggle pause/continue the playback of tune/patterns.
Shift+SPACE	Shift+SPACE	Playback	Play the selected patterns from cursor-position.
Shift + A/Z	Shift + A/Z	Setting	Increase/Decrease auto-advance (stepping) amount after typing notes.
Shift + D	Shift + D	Setting	Toggle monophonic+legato or polyphonic +dovetailing jamming mode.
Shift + F/G	Shift + F/G	Setting	Decrease/increase framespeed (1x singlespeed...8x multispeed).
Shift + H/J	Shift+H/J	Setting	Decrease/increase step-highlighting aid frequency in pattern-editor window.
Shift + B	Shift + B	Setting	Toggle pattern/track-binding on/off. Patterns scroll together/independently.
C= + T	Control + T	Setting	Toggle global normal tempo / funktempo mode for subtune.
Shift + colon/dot , Shift + : or ;	Shift + colon/dot , Shift + ; or '	Setting	Increase/decrease global normal tempo / 2nd funktempo for subtune.
C= + colon / dot	Control + colon / dot	Selection	Select subtune (Orderlist). Patterns are shared between subtunes.

C= + 1...8	Control + 1...8	Selection	Select octave 1...8 to edit or jam musical notes.
Shift+PLUS or Shift +MINUS	Shift+PLUS or Shift +MINUS	Selection	Select actual instrument for jamming / editing.
Shift + T/Y	Shift + T/Y	Selection	Select (decrease/increase number of) tempo-program.
Shift + K/L	Shift + K/L	Selection	Select (decrease/increase number of) Chord to edit.
SPACE	SPACE	Editing	Toggle Keyboard note-jamming or note-editing in pattern-editor window.
Shift + N	Shift + N	Editing	Rename selected instrument. Max. 8 characters, Esc/Stop aborts the renaming.
DEL/Pound or Shift +DEL	Delete/Backspace or Insert	Editing	Delete/insert in cursor position or increase/decrease pattern/table size in end-positions.

*In Linux the VICE build seems to have up-arrow key associated to Page-Down, while the default would be 'Del'-key (which is good, because therefore in Linux the Page-Down (up-arrow) is in correct place, below Page-Up, which is associated to Restore C64 key.) In Linux the 'Del'-key works as Backspace which is better selection IMO.

*Note that in follow-play modes the pattern-editor keys are inactive to prevent accidental editing while playing the tune.

2.2 Pattern-editor related keys

Keys on C64	US int. Keys in VICE emulator	Category	Functions of pressed keys
CONTROL or Shift +CONTROL	TAB or Shift+TAB	Navigation	Cycle through tracks forwards/backwards.
RETURN or Shift +RETURN	Enter or Shift+Enter	Navigation	Go to next row, place Note-on/off, or select instrument under cursor.
Z,X,C,V,B,N,M, colon, dot	Z,X,C,V,B,N,M, colon, dot	Edit/Jam	Lower octave white piano-keys (C major diatonic scale whole-tones). Legato at jamming is possible.
Q,W,E,R,T,Y,U,I,O,P	Q,W,E,R,T,Y,U,I,O,P	Edit/Jam	Upper octave white piano-keys (C major diatonic scale whole-tones).
S,D, G,H,J, L , 2,3, 5,6,7, 9,0	S,D, G,H,J, L , 2,3, 5,6,7, 9,0	Edit/Jam	Lower and upper octave black piano keys (halftones).
C= + 1...8	Control + 1...8	Edit/Jam	Select octave 1...8 where editing or jamming musical notes happens.
A, 1	A, 1	Editing	Empty-note (Delete note in note-column pattern-position without moving the rest.)
1...9, A...F hexa keys	1...9, A...F hexa keys	Editing	In instrument/effect columns type value, in note-column set vibrato-amplitude or note.
DEL/Pound or Shift +DEL	Delete/Backspace or Insert	Editing	Delete/insert in cursor position or increase/decrease pattern-size in pattern-end position.
C= + DEL	Control + Delete/ Backspace	Editing	Delete instrument and effect columns in actual pattern-row in actual track.
C= + Shift + DEL	Control + Shift + Del./ Backspace	Editing	Delete the entire pattern-row (note and instrument and effect) in the actual track.
Shift + Q/W	Shift + Q/W	Editing	Transpose notes up/down by half-notes in actual pattern after cursor-position.
C= + Q/W	Control + Q/W	Editing	Transpose notes up/down by octaves in actual pattern after cursor-position.
Shift + R or C= + R	Shift + R or Control + R	Editing	Place ring-modulation effect on/off into note column at actual cursor position.
Shift + P	Shift + P	Editing	Place auto-portamento effect into note column at actual cursor position.
Shift + S or C= + S	Shift + S or Control + S	Editing	Place sync-bit on/off effect into note column at actual cursor position.
Shift + V	Shift + V	Editing	Place vibrato-effect into note column at actual cursor position. Amplitude editable.
C= + X	Control + X	Editing	Cut/delete pattern content from cursor position to end of pattern and copy to pattern-buffer/ clipboard.
C= + C	Control + C	Editing	Copy pattern content from cursor position to end of pattern to pattern-buffer/clipboard.
Shift + C	Shift + C	Editing	Limit the range of data copied (or cut) into buffer to cursor position. End of selection to copy.
C= + V	Control + V	Editing	Paste pattern-clipboard content from cursor position till end of pattern.

2.3 Orderlist-editor related keys

Keys on C64	US int. Keys in VICE emulator	Category	Functions of pressed keys
RETURN or Shift+RETURN	Enter or Shift+Enter	Navigation	Selects the pattern(s) under the cursor position in Orderlist and jumps to actual track in Pattern-editor.
C= + RETURN	Control + Enter	Navigation	Selects the patterns that are actually played and jumps to actual track in the Pattern-editor.
Shift+SPACE	Shift + Space	Playback	Sets Orderlist playstart-marker (for F2-playing) to cursor-position for all tracks. Position number gets inverted.
C= + SPACE	Control + SPACE	Playback	Sets Orderlist playstart-markers to individual played position for all tracks. Position number of 1st track gets inverted.
1...9, A...F hexa keys	1...9, A...F hexa keys	Editing	To type hexa pattern-numbers and effect/jump numbers into Orderlist sequences of tracks.
DEL/Pound or Shift+DEL	Delete/Backspace or Insert	Editing	Delete/insert in cursor position or increase/decrease Orderlist sequence-size in sequence-end/loop position.

2.4 Instrument-editor related keys

Keys on C64	US int. Keys in VICE emulator	Category	Functions of pressed keys
CONTROL or Shift+CONTROL	TAB or Shift+TAB	Navigation	Cycle through main instrument-setting panel and instrument-tables.
RETURN or Shift+RETURN	Enter or Shift+Enter	Navigation	'Car return' to beginning of next row in tables (or toggle/cycle some main instrument-parameters.)
+ / -	+ / -	Selection	Select actual instrument. The same in pattern/orderlist/instrument windows. Shift is needed in other windows.
RETURN	Enter	Selection	Toggle/cycle some of the instrument's main parameters like HR-type, Vibrato-type, octave-shift sign....
1...9, A...F hexa keys and F..Z	1...9, A...F hexa keys	Editing	To type hexa values into instrument-data fields, or type the name of the instrument (some signals allowed too.)
Shift + N	Shift + N	Editing	Rename selected instrument. Max. 8 characters, Esc/Stop aborts the renaming.
DEL/Pound or Shift+DEL	Delete/Backspace or Insert	Editing	Delete/insert in cursor position. Increases/decreases table-size.
C= + C	Control + C	Editing	Copy the selected instrument entirely to instrument-buffer/clipboard.
C= + V	Control + V	Editing	Paste instrument-clipboard content to the selected instrument. The entire instrument will be overwritten.
Shift + Space	Shift + Space	Editing	Set/clear gate-off index to current table-row under cursor

--	--	--	--

2.5 Chord-table / Tempo-table related keys

Keys on C64	US int. Keys in VICE emulator	Category	Functions of pressed keys
CONTROL or Shift +CONTROL	TAB or Shift+TAB	Navigation	Go to pattern editor or go to instrument-editor.
RETURN or Shift +RETURN	Enter or Shift+Enter	Navigation	'Car return' to beginning of next row in tables (or toggle/cycle some main instrument-parameters.)
+ / -	+ / -	Selection	Select actual chord/tempo-program. Shift+K/L or Shift+T/Y is needed in other windows.
Shift+PLUS or Shift +MINUS	Shift+PLUS or Shift +MINUS	Selection	Select actual instrument for jamming / editing.
1...9, A...F hexa keys	1...9, A...F hexa keys	Editing	To type hexa values and chord loop/return numbers into chordtable/tempoprogram-table.
DEL/Pound or Shift +DEL	Delete/Backspace or Insert	Editing	Delete/insert in cursor position. Increases/ decreases table-size.

III. A guide to hexa parameter values

1. Instrument-parameters

1.1 Main instrument-settings

- **ADSR** - Attack/Decay/Sustain/Release envelope-generator values
- **ADHR** - The ADSR value for hard restart (get loaded to ADSR registers 1-2 frames before a new note is triggered).
- **1st frame** - If switched on, a 'sexy' start is performed on frame 1 of instrument. (Test-bit and gate is set.)
- **Hard-restart timer** - 0..2 - the amount of frames for hard-restart before note-triggering
- **Hard-restart type** - If Gate-bit is also to be reset at hard-restart (Adds 1-2 frames of gap between consequent notes.)
- **Vibrato amplitude, frequency and delay / increment-speed** (the latter in case of increasing vibrato-type)
- **Vibrato-type** - can be 'incremental' ('violin' style), delayed 'up-down', delayed 'upwards', delayed 'downwards'
- **Instrument-table & chord-table execution speed** - especially useful for multispeed to slow down WF-table execution if needed. (If the value is > \$40, pulsewidth-table can be multispeed, if >\$80, filtertable can be multispeed for the instrument as well...)
- **Default Chord** - this is linked to the instrument by default, but a pattern-FX can change the chord as well
- **Octave shift** - to incorporate bass and high pitched sounds easier - no need to jump between octaves back & forth

1.2 Waveform-arpeggio-detune table

- **WF-COLUMN:**
 - **\$00..\$0F** Repeat only arpeggio/detune-columns for 1..16 frames (no Waveform-change)
 - **\$10..\$FD** Simple WAVEFORM/CONTROL register value setting
 - **\$FE** Jump to a table-position (position in 2nd column, if >=\$40, it jumps to itself)
 - **\$FF** End of the table, table execution ends here
- **ARP-COLUMN:**
 - **\$00** Zero pitch-shifting (prime, '0' interval) - regain original note-pitch
 - **\$01..\$5F** Relative pitch-shift upwards in halftones (positive interval)
 - **\$7F** Jump to default or pattern-FX set Chord
 - **\$80** No process (useful when don't want to touch pitch and detune, only waveform)
 - **\$81..\$DF** Set absolute pitch (frequencies are identical to C-1..A-8 notes)
 - **\$E0..\$FF** Relative pitch-shift downwards in negative direction (negative interval)
- **DETUNE-COLUMN:**
 - **\$00..\$FF** Set fine-detuning amount

1.3 Pulsewidth-program table

- **\$8x..\$Fx xx** set square-wave Pulsewidth hi-nibble and low-byte
 - **\$00..\$7F xx** 0..127 times adding/subtracting signed xx value (0...\$7F / \$80...\$ff)
 - **\$FE** Jump to a position in the table (position in 2nd column, can jump to itself)
 - **\$FF** End of the table, table execution ends here
- **The 3rd (last) column sets Keyboard-tracking (KT).** That technique is known in Korg synthesizers (MS20, 707, etc.) and makes the pulsewidth dependent on note-pitch. The sounds get more lively and varying by this setting. The bigger the value, the bigger the difference in timbre of adjacent note-pitches is.

1.4 Filter-program table

Filter-switch for channels: SID has only one filter, common for the 3 tracks. Every channel gets filtered whose instrument has something in its filter-table's 1st row, when the note starts. The instrument which has \$00 there, is only filtered but doesn't control the filter, while if there's nonzero in the 1st row, the instrument controls the filter (band, resonance and cutoff-frequency). Always the latest instrument takes over the control over the common filter-band/resonance/frequency (like with polyphonic jamming in JCH's editor)...

- **\$8r..\$Fr xx** Case 1: Set filter-parameters on: bit6..4:Filter-band, bit3..0 (nibble 2):Resonance, XX:Cutoff frequency hi-byte (for example: 9F 38 means setting low-pass filter with \$F (15) resonance and cutoff-frequency high byte at \$38)
 - **\$00..\$7F xx** Case 2: repeat adding/subtracting signed XX on every frame 0..127 times (the sweep utilizes full 11bit of the filter-cutoff frequency register in SID.)
 - **\$FE** Jump to a position in the table (position in 2nd column, can jump to itself)
 - **\$FF** End of the table, table execution ends here
- **The 3rd column is Keyboard-Tracking strength of cutoff-frequency.** The cutoff frequency gets pitch-dependent with this setting. This effect works in the same way as in pulsewidth-table, and produces more lively sounds which can open in timbre / resonate always near the frequency of the musical tone thus giving impression of live instruments.

2. Pattern Effects

2.1 Note-column effects

Note column can hold some pitch/gating related effects, these are displayed graphically in note-column and can be placed into the pattern by keyboard combinations described before. Almost all of them are simple switches (Sync ON/OFF, Ring-modulation ON/OFF, tone-portamento ON), except vibrato, which can have a one-nibble (0..F) amplitude value.

2.2 Instrument-column effects

Instrument column can hold some Small effects where the 1st nibble is the effect-type, the 2nd nibble is the corresponding value (0..15). These Small-FX numbers are almost identical to Small-FX and Big-FX numbers of effect-column, and as such, can be remembered easier.

- \$01-\$3E Select an instrument for the actual note, which stays selected until new comes
- \$3F Tied note (true legato, instrument doesn't restart just note-pitch changes)
- \$40-\$4F **Waveform (reg.4) nibble adjusting - any subsequent WF-table waveform change overrides it.**
- \$50-\$5F **Sustain nibble adjusting of ADSR (Small-FX)**
- \$60-\$6F **Release (SID reg.6) nibble adjusting of ADSR (Small-FX)**
- \$70-\$7F **Select Chord (overriding the default) for the instrument. (No need to create instrument for every chord. Instead, create an instrument, call chord-table with \$7f from ARP-table, then select the chord with this pattern-FX.)**

2.3 Effect-column SMALL effects

Small effects' 1st nibble is their type of operation, the 2nd nibble is the corresponding effect-value/amount. Small-FX numbers \$4..\$7 are identical in function to Big-FX numbers of the same range, (what more, the hexa effect-numbers are related to SID-registers and effect-names) to support easier memorization.

- \$20-\$2F Attack nibble adjusting of ADSR (Small-FX)
- \$30-\$3F Decay nibble adjusting of ADSR (Small-FX)
- \$40-\$4F **Waveform (reg.4) nibble adjusting - any subsequent WF-table waveform change overrides it.**
- \$50-\$5F **Sustain nibble adjusting of ADSR (Small-FX)**
- \$60-\$6F **Release (SID reg.6) nibble adjusting of ADSR (Small-FX)**
- \$70-\$7F **Select Chord (overriding the default) for the instrument. (No need to create instrument for every chord. Instead, create an instrument, call a chord with \$7f from ARP-table, then select chord.)**
- \$80-\$8F Vibrato Amplitude adjustment - The frequency stays intact.
- \$90-\$9F Vibrato Frequency adjustment - Amplitude stays intact.
- \$A0-\$AF Adjust Main volume (low nibble of \$d418)
- \$B0-\$BF **Band-nibble of filter-register (LOW/MID/HI/3OFF)**
- \$C0-\$CF **Chord-speed adjustment (arpeggio-speed in case of explicit arpeggio)**

- \$D0-\$DF **Detune** actual note with given amount
- \$E0-\$EF **Enable/disable** Test-bit/Ring-bit/Sync-bit/Gate-bit (Waveform-table can override)
- \$F0-\$FF **Filter resonance** (strength) nibble setting (Subsequent filter-table command can override it.)

2.4 Effect-column **BIG** effects

Big effects has their effect-type in effect-column and their 1 byte (\$0..\$FF or -\$7f..+\$80) value is in the last column of a pattern. (\$4...\$7 effects are identical to Small effect-types of the same range to aid human memorization.)

- \$01 - Pitch Slide UP (same FX-number as Goattracker's)
- \$02 - Pitch Slide DOWN (same FX-number as Goattracker's)
- \$03 - Tone-portamento (with given speed) (same FX-number as Goattracker's)
- **\$04 - Waveform-Control (\$d404, etc.) register setting (or above \$f0: goes to WF-ARP table-position)**
- **\$05 - ATTACK/DECAY (SID-register 5) byte adjustment**
- **\$06 - SUSTAIN/RELEASE (SID-register 6) byte adjustment**
- **\$07 - Select Chord (overriding the default) for the instrument.**
- \$08 - Vibrato Amplitude and Frequency adjustment (overrides instrument's default vibrato)
- \$09 - Select vibrato-type (possible values: \$00, \$10, \$20, \$30)
- \$0A - Adjust Pulsewidth-program table-index for the actual instrument (go to table-position)
- \$0B - Branch to Filter-program table-index for the actual instrument
- \$0C - Chord-speed adjustment (arpeggio-speed in case of explicit arpeggio)
- \$0D - Detune actual note with given amount
- \$0E - Simple pulsewidth setting for pulse-waveform (instr. PW-table changes can override it)
- \$0F - Filter cutoff-frequency high-byte setting (instr. Filter-table changes can override it)
- \$10 - Set Main (subtune's basic) single-tempo
- \$11 - Set Main (subtune's) funktempo. 1st (even rows) and 2nd (odd rows) tempo, in left/right nibbles.
- \$12 - Set Main tempo-program for whole subtune (any speeds can be given for each rows)
- \$13 - Set Track's individual single-tempo
- \$14 - Set Track's funktempo (even and odd rows' in left/right nibbles)
- \$15 - Set Track's individual tempo-program
- \$16..\$1C - Kept for later expansions....
- \$1D - Delay track by \$00..\$ff (20ms) frames
- \$1E - Delay only the actual note by \$00..\$ff frames. Protection needed to avoid too large.
- \$1F - Kept for later expansions....

3. Orderlist-Effects

- \$00..\$7F The values kept for normal pattern-numbers in the orderlist
- \$80..\$8F Transpose key down (e.g.: \$8f=half-note down ; \$8e=whole note down...)
- \$90 Switch off transposing (set original key)
- \$91..\$9F Transpose key up (e.g.: \$91=half-note up ; \$92=whole note up....)
- \$A0..\$AF Set main volume (\$0..\$f)
- \$B0..\$FD Set track-tempo on the corresponding track (\$00..\$5d)
- **\$FE..\$FF are reserved to control sequence-flow. Playbacks stops at \$FE and loops when reaching \$FF position. If a position number below \$80 is written after \$FF, it loops to that given position in actual subtune. If a number above \$80 is after \$FF in any tracks, the playback jumps to the given (that number minus \$80) subtune!** (useful for composing music in smaller chunks for e.g. demo-parts)

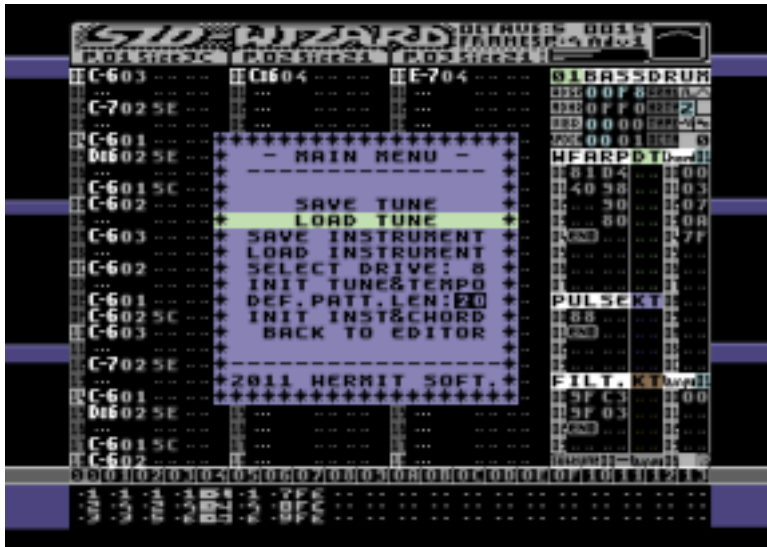
**Note: An Orderlist-effect shouldn't be right before a \$FF loop-signal. If this happens, the loop command will be ignored to prevent freezing that could be caused by jumping rapidly between orderlist-FX and looping.*

IV. Menu and file operations

0. Overview of menu and file-browser

You can select **SID-Wizard Menu** by pressing **F8** as mentioned in the previous chapters. Then you can choose an option with the obvious **cursor up/down keys** then by pressing **Return**. The only exception is the menu point to select default pattern-length, where you may use **+/- keys** to select default pattern-length for the tune.

Some keyboard functions (related to playback and instrument-selection) will still be functional while you can select an action from the menu. (Pattern-playback display won't be refreshed while menu is displayed in front of it... The latest used menu point will be remembered even after file-operations or exiting from menu.)



Let's discuss the behaviour of the different menu points.

1. Load/Save Tune (workfile) or Instrument

If you select 'Save' or 'Load' options, they'll **invoke the PC-like file-dialog where you can type desired filename directly or select (and further edit) it from the disk directory.**

(When you want to load or save instrument, make sure the **instrument (subject of load/save operation) was selected in the editor/menu beforehand with +/- keys.** Disk drive 8..11 can be selected (cycled) with Return/Enter key in the 'select drive' menupoint beforehand.)

When disk-directory is read, filenames are filtered and distributed in 3 columns, and you can observe diskname and free disk space (in kbytes) at the bottom of the screen. Note that **only 12 character-long filenames are allowed.** That is due to the **automatically attached '.SWM' or '.SWI' extension** ('M' stands for 'module', 'I' stands for instrument), **which SID-Wizard uses to filter the filenames when displaying the disk-directory. The default disk-drive could be selected in advance from Menu** (device 8...11 alias hexa 8,9,A,B). In the upper-right corner of the file-dialog you can see the editor's version number, if you want to load a newer/other module-version (possibly) coming in the future, you'll be warned, because the module's format-version should always match the editor's or exporter's version.

You can use these keys in file-dialog

Keys on C64	US int. Keys in VICE emulator	Functions of pressed keys
F1	F1	Re-read disk-directory, go to its beginning
F3 / UP-arrow	F3 or Page-Down	Page down in directory (if bigger than one-screen)
Control	TAB	Cycle between filename-typer / authorinfo-typer / filename-selector fields. (Author-info is only editable when saving tune.)
Cursor Up/Down/Left/Right	Cursor Up/Down/Left/Right	Obviously navigating cursor in typer / file-selector fields
Return	Enter	Perform disk-operation (or leave author-info editor). You'll notified if problems occur (file exists/file not found).
F7 / STOP	F7/F8/Esc	Exit from file-dialog back to editor without performing any file operation.

The .SWM format is an optimized/compressed own format of SID-Wizard, not compatible with any other systems. But must stay compatible through further SID-Wizard versions. (It contains 'SWM1' filetype&version string to notify editor if its version mismatches.)

The .SWI format is a copy of one selected instrument (without the empty/unused spaces)...

2. Initialize/Clear/Wipe Tune or Instrument

You can clear/initialize subtunes (patterns, tempos) or instruments/chords independently from two different menupoints. Before clearing tune, you might have to use +/- keys on **'default pattern-length' menupoint to give direction for SID-Wizard what pattern-length to use by default** when initializing the tune. The default pattern-length will be saved as information in the .SWM workfile, so when you load the tune back later, the empty patterns will automatically be sized to that value (but size can be set afterwards as well.)

3. SID-Maker (SID-Wizard's Packer/Relocator/Exporter)



This is a **separate executable** application file on the SID-Wizard disk, because relocation could hardly be achieved to memory areas which SID-Wizard editor takes for code. I guess this won't be such a big problem considering that this step will possibly be the last step after tune gets ready to be released. (The device will be, which SID-Maker was loaded from.) SID-Maker can generate **4 kinds of file outputs at the moment: 3 native C64 formats ('.C64.PRG', '.BIN.SEQ' and runnable '.EXE.PRG'), and '.SID.SEQ' (SID) format** which enables cross-platform playback, if transferred with a cross-platform file-copier like Star Commander or Total Commander D64 plugin. (The .PRG and .SEQ are standard extensions appended by CBM-DOS. If you need a SID, you have to rename the '.sid.seq' extension of the exported tune to '.sid' - this might depend on your copier tool.)

-If you export SID format and you used a ':' in the author-information, the SID-header will automatically be split in this fashion: The content before the ':' will be the author-name, the content after the ':' will be the tune-title.

-Version of modules is checked in SID-Maker as well as in SID-Wizard, SWM module-version needs to match SID-Maker software-version.

-In case you have 'SAVE ERROR' (with CBM-DOS error code) you might not have compatible Disk Drive like 1541, so the exporter cannot write SEQ files. If you want a compatible mode anyway, you can still perform a standard compatible 'SAVE' process by exporting in the (1st) C64 PRG format without modifying relocation address.

If you set relocation address for PRG or SID format, the exporter will switch to compatible SAVE mode automatically in case of error; but that will involve the exported tune's load-address to be forced to the default \$1000 (or \$0F82 in case of SID) - however the code itself gets relocated. If this is the case, you should load the relocated tune manually by forcing the desired load-address, when you include the PRG in a program. (In case of SID you might delete the 2 byte load-address with a hex-editor to make the SID compatible.)

4. SWM-converter

I created a PC (Win/Linux) executable for the bundle that can convert SWM format to/from XM or MID format in command-line. (A GUI frontend is planned for the future, now you have to type the names of the input/output module-files by hand...)

The usage and syntax is easy, and described by SWMconvert executable if you run it without command-line parameters or with '-help' parameter. (Good to have it in system-directory...)

Note that, there are some effects that cannot be transferred between the different formats, e.g. XM doesn't support tempo-programs or different pattern-lengths; and SWM can't contain more than 3 channels. Only channels/tracks 1..3 of XM or MID are converted to the SWM at this time, so you should edit the XM/MID file beforehand. These 3 track should even be monophonic for better conversion, because only some notes of polyphonic tracks get converted otherwise. On the other hand, the SWM chords are expanded to XM/MIDI tracks when converted.

If an XM pattern is too long, it will be cropped to the maximum allowable length in SWM...

MIDI-to-SWM conversion processes the time signature in MIDI files. If the MIDI-file doesn't contain the time-signature, default will be 4/4. The converter tries to reuse as much patterns as possible when there are repeated parts in the MIDI.

SWM-to-MIDI conversion handles track-based tempo-changes (even from orderlist). The 3 SWM tracks are converted directly to 3 MIDI-tracks (MIDI format 1). However the instruments are spread in different channels. Your MIDI-sequencer can possibly separate them... (e.g. Muse could do it, while SEQ24 handled it as 3 tracks only...)

```
=====
SID-Wizard SWM module v1.0 converter
=====
Converts to/from XM or MIDI formats,
between .prg & .P00, and S00 to SID.
Usage syntax is pretty simple:
SWMconvert <inputfile> [outputfile]
-----
(Formats will depend on extensions.)
If you don't specify an output-file,
name of the input file will be used
as output-filename. Existing files
will be overwritten automatically.
-----
2012 Hermit Soft. (Mihaly Horvath)
=====
```

5. Application note to the player routine (to embed in programs)

The player saves and restores the 2 internally used zeropage bytes (by default \$fe and \$ff). Therefore inclusion is easy in any programs as **virtually no zeropage is affected outside the routine**. The player routine's required **maximal rastertime is around \$20** raster-rows, slightly depending on the number of effects and table-commands used simultaneously...if less than 3 tracks are utilized in the music, rastertime-usage of the player routine decreases significantly.

The **initializer routine's caller address is the same as the base-address/load-address (e.g. \$1000)** which requires a subtune-number present in Accumulator (as usual with other routines too).

The **single-speed playing-address is the init-address plus 3 (e.g \$1003)**. The multi-speed playing is done in the same way as XSID's (not as Goatracker's), so the **multi-speed routine's calls are to be performed at init-address plus 6 (e.g. \$1006)** beside the single-speed routine, on different rasterlines of course.

To **change volume**, put the desired volume (0..F) into the Accumulator and call volume-setter routine at init-address plus 9 (e.g. \$1009).

V. The HerMIDI interface (a MIDI-input hardware extension)

If you want to control SID-Wizard with a MIDI-keyboard or controller, here's my approach to the solution:

I developed a MIDI interface for the C64 which can easily be built by anyone who wan't a fast and cheap reproducible solution. (Yes, I have a MSSIAH, but it's interface isn't open-source and cannot be used by other programs afaik...)

You don't have to worry about hard-to-find parts, like Userport-connector, and don't even have to occupy the Expansion-port. **HerMIDI is to be connected to Serial port** at the back of the C64 (or 1541 drive in daisy-chain), and **behaves similarly to a usual drive**.

When C64 (or other IEC/Serial compatible product) **calls device 15 with TALK-command**, HerMIDI knows it's time to serve the C64 with MIDI data coming from its input, and **its LED lights up** as a sign it's ready for C64's requests.

C64 can request data anytime by pulling DATA line low, and HerMIDI puts one byte through the Serial port asynchronously in around 6..8 rastertimes. At this time only 1 byte can be sent and this might include a Note/Note-off or other message like e.g. instrument-selection. In HerMIDI's building/programming manual these messages are explained in details.

If you have a fastloader routine that uses ATN line for transfer HerMIDI should be unplugged or switched off to prevent data-interferences (it receives its 5V power through the MIDI cable).

The source code for the on-board PIC12F509 and the schematics are included in SID-Wizard source-directory under 'HerMIDI' folder.

In advance, this is how HerMIDI looks like when built and assembled:



Closing words

The program has been going through extensive testing but it's still possible to get hang-ups occasionally if e.g. there's a problem with an external device like floppy-drive or disk. If this is the case, you can restart SID-Wizard safely by resetting the machine and typing **SYS2061** launcher (BASIC) instruction at the C64 prompt...

I hope You'll enjoy using this music editor and my long-term work is useful.
(I advise to use it on real C64 hardware to get true SID sound, and reduce air-pollution. :)
Wish you the best of inspiration and musical ideas...

Thanks to the people who contributed to the project:
Necropolo (Péter Nagy-Miklós) by testing SID-Wizard,
Unreal (Csaba Kémeri) who designed graphics of the box
Dóra Kőrösi for correcting my 'here-and-there' grammar issues

Mihaly Horvath alias Hermit,
Hungary
2012 AD

hmtmade@gmail.com
<http://hermit.netne.net>