

”

ReadMe2Guide

COLLABORATORS

	<i>TITLE :</i>		
	"		
<i>ACTION</i>	<i>NAME</i>	<i>DATE</i>	<i>SIGNATURE</i>
WRITTEN BY	ReadMe2Guide	February 17, 2023	

REVISION HISTORY

NUMBER	DATE	DESCRIPTION	NAME

Contents

1	"	1
1.1	Main	1
1.2	Table of contents	1
1.3	1. General information	2
1.4	1.1 Warnings	3
1.5	1.2 Compatibility with v1.xx	4
1.6	2. Using GoatTracker	4
1.7	2.1 Command line options	5
1.8	2.2 Hardware support	6
1.9	2.3 Keyboard commands	7
1.10	2.3.1 General keys	7
1.11	2.3.2 Pattern edit mode	7
1.12	2.3.2.1 Protracker note-entry mode	8
1.13	2.3.2.2 DMC note-entry mode	9
1.14	2.3.3 Song edit mode	9
1.15	2.3.4 Instrument edit mode	10
1.16	2.3.5 Table edit mode	10
1.17	2.3.6 Songname edit mode	11
1.18	2.4 Mouse control	11
1.19	3. Song data	12
1.20	3.1 Orderlist data	12
1.21	3.2 Pattern data	12
1.22	3.3 Instrument data	14
1.23	3.4 Table data	15
1.24	3.4.1 Wavetable	16
1.25	3.4.2 Pulsetable	17
1.26	3.4.3 Filtertable	18
1.27	3.4.4 Speedtable	19
1.28	3.5 Playback details	20
1.29	3.6 Miscellaneous tips	21

1.30	3.6.1	Limit-based modulation steps	21
1.31	3.6.2	Precalculated note-independent hifi player portamento/vibrato	22
1.32	3.7	Multispeed tips	22
1.33	4.	Using the included utilities	23
1.34	4.1	INS2SND2.EXE	23
1.35	4.2	SNGSPLI2.EXE	23
1.36	4.3	MOD2SNG.EXE	24
1.37	4.4	BETACONV.EXE	24
1.38	5.	Using the songs outside the editor	24
1.39	5.1	Playroutine options	25
1.40	6.	File/data formats description	26
1.41	6.1	GoatTracker v2 song (.SNG) format	26
1.42	6.1.1	Song header	27
1.43	6.1.2	Song orderlists	27
1.44	6.1.3	Instruments	27
1.45	6.1.4	Tables	28
1.46	6.1.5	Patterns header	28
1.47	6.1.6	Patterns	28
1.48	6.2	GoatTracker v2 instrument (.INS) format	28
1.49	6.3	Sound effect data format	29
1.50	7.	Recompiling	30
1.51	8.	Version history	30

Chapter 1

"

1.1 Main

GoatTracker v2.71

Editor by Lasse Öörni (loorni@gmail.com)
HardSID 4U support by Téli Sándor.
Uses reSID engine by Dag Lem.
Uses reSID distortion / nonlinearity by Antti Lankila.
Uses 6510 crossassembler from Exomizer2 beta by Magnus Lind.
Uses the SDL library.
GoatTracker icon by Antonio Vera.
Command quick reference by Simon Bennett.
Patches by Stefan A. Haubenthal, Valerio Cannone and Raine M. Ekman.

Distributed under GNU General Public License
(see the file COPYING for details)

Covert BitOps homepage:
<http://covertbitops.c64.org>

GoatTracker 2 SourceForge.net page:
<http://sourceforge.net/projects/goattracker2>

1.2 Table of contents

Table of contents

- 1. General information
 - 1.1 Warnings
 - 1.2 Compatibility with v1.xx
 - 2. Using GoatTracker
 - 2.1 Command line options
 - 2.2 Hardware support
-

-
- 2.3 Keyboard commands
 - 2.3.1 General keys
 - 2.3.2 Pattern edit mode
 - 2.3.2.1 Protracker note-entry mode
 - 2.3.2.2 DMC note-entry mode
 - 2.3.3 Song edit mode
 - 2.3.4 Instrument edit mode
 - 2.3.5 Table edit mode
 - 2.3.6 Songname edit mode
 - 2.4 Mouse control

 - 3. Song data
 - 3.1 Orderlist data
 - 3.2 Pattern data
 - 3.3 Instrument data
 - 3.4 Table data
 - 3.4.1 Wavetable
 - 3.4.2 Pulsetable
 - 3.4.3 Filtertable
 - 3.4.4 Speedtable
 - 3.5 Playback details
 - 3.6 Miscellaneous tips
 - 3.6.1 Limit-based modulation steps
 - 3.6.2 Precalculated note-independent "hifi player" portamento/vibrato
 - 3.7 Multispeed tips

 - 4. Using the included utilities
 - 4.1 INS2SND2.EXE
 - 4.2 SNGSPLI2.EXE
 - 4.3 MOD2SNG.EXE
 - 4.4 BETACONV.EXE

 - 5. Using the songs outside the editor
 - 5.1 Playroutine options

 - 6. File/data formats description
 - 6.1 GoatTracker v2 song (.SNG) format
 - 6.1.1 Song header
 - 6.1.2 Song orderlists
 - 6.1.3 Instruments
 - 6.1.4 Tables
 - 6.1.5 Patterns header
 - 6.1.6 Patterns
 - 6.2 GoatTracker v2 instrument (.INS) format
 - 6.3 Sound effect data format

 - 7. Recompiling

 - 8. Version history

1.3 1. General information

- 1. General information
-
-

This program is a tracker-like C64 music editor running on Win32 or Linux platforms (using the SDL library, see <http://www.libsdl.org>)

GoatTracker v2 adds more commands and uniform step programming tables for waveform/arpeggio, pulse effects, and filter effects. It is likely much more complex to learn & master than v1.xx.

Familiarity with tracker programs in general, hexadecimal notation, and the C64's SID chip are required. Consult the C64 Programmer's Reference Guide (<http://project64.c64.org>) or AAY64 (<http://www.the-dreams.de/aay.html>) for SID chip reference.

For filesize & library compatibility reasons, precompiled binaries exist only for Win32 platform.

1.4 1.1 Warnings

1.1 Warnings

-
1. Always look at the end of this file for changes! Sometimes keyboard commands change etc.
 2. Always save your songs in the .SNG-format with F11 key if you plan to continue editing! Packed & relocated songs (PRG/BIN/SID) can not be loaded back into the editor.
 3. Even the reSID emulation is in some cases quite far from the output of a real SID. Especially if filters are in use, consider strongly testing your tune on a C64 or on a HardSID card. (Using filters has always been complicated because every SID tends to sound different.)
 4. The editor will stop playing if:
 - The song restart position is illegal (beyond end of song)
 - The sequence of orderlist commands is incorrect
 - * In a sequence of both TRANSPOSE & REPEAT, TRANSPOSE must come first
 - * The last thing before a RST-endmark should be a pattern number
 - Gateoff timer value is too high compared to song tempo.
 - The song tries to jump directly onto a table row containing a jump (FF xx), either with instrument pointers or 8XY, 9XY, AXY commands.
 - The song tries to execute pattern commands 0XY, 8XY or EXY from the wavetable.
 These all are error conditions, and trying to pack/relocate such song should give an error message.
 5. Using delayed wavetable or no wavechange (0x 00, where x is 0-F) in the first step of instrument wavetable is unsupported and may result in missing notes.
 6. When using a playroutine with unbuffered SID-writes (Standard /w unbuffered or Minimal) and encountering ADSR-bugs after packing/relocating, you can try either:
 - 1) Set pulse-startpos to nonzero value in the troublesome instruments and change the 1stFrame Wave parameter of some instrument slightly, for
-

example from \$09 to \$0B, to disable a playroutine optimization. The idea in this is to make the noteinit routine take more CPU cycles.

- 2) Use a playroutine with buffered writes to pack/relocate.
 - 3) Try hardrestart attack parameter \$F for alternative SID register write order.
7. From v2.18 onwards, wavetable left side values have changed. Delay can now be maximum 16 frames (values \$00-\$0F), and waveform register values \$00-\$0F (inaudible) have been mapped to table values \$E0-\$EF. For example, testbit + gate would be \$E9. You need to manually convert values \$08-\$0F from your old songs!
 8. From v2.2 onwards, a new table (speedtable) has been added. The portamento, vibrato and funktempo commands use it. Old songs will be converted upon loading to use the speedtable. Note the keyboard command SHIFT+RETURN to convert the old style parameters.
 9. From v2.4 onwards, pulse modulation speed has 1 bit added accuracy, so you need to double pulsespeeds when composing new songs. Old songs will be converted automatically.
 10. From v2.59 onwards, gateoff timer parameter high bit disables hardrestart and bit \$40 disables gateoff. Old songs & instruments will be converted automatically.
 11. From v2.62 onwards, realtime calculated portamento/vibrato speeds are supported. This is activated by having the high bit (\$80) set in the speedtable left side. Naturally, this means that very high portamento speeds (\geq \$8000) or vibrato speeds (\geq \$80) become unavailable, but these should not be very useful anyway.
 12. From v2.68 onwards, SID write order has been tweaked for better stability regarding badlines. This has the consequence, however, that notes with attack 0 & release 1 may ADSR-bug. In this case, try hardrestart attack parameter \$F for alternative SID register write order.

1.5 1.2 Compatibility with v1.xx

1.2 Compatibility with v1.xx

GoatTracker v2 can load v1.xx songs and instruments, but it saves only in v2 format. Some subtleties (like tricks involving instrument changes) will not play back exactly like in v1.xx.

The only major feature removal is that of the arpeggio command in v2. Everything that this command does can also be done with wavetables, and the import feature converts all arpeggio commands to corresponding wavetable programs.

1.6 2. Using GoatTracker

2. Using GoatTracker

1.7 2.1 Command line options

2.1 Command line options

Start GoatTracker V2 by typing GOATTRK2 in the command prompt, followed by the songname to be loaded at startup (optional) and any command line options you may want. For example "goattrk2 dojo.sng -sl -e1" to set 1x-speed mode and SID model 8580 and to load "dojo.sng" on startup.

```
-Axx Set hard restart ADSR parameter in hex. DEFAULT=0F00
-Bxx Set sound buffer length in milliseconds DEFAULT=100
-Cxx Use CatWeasel MK3 PCI SID (0 = off, 1 = on)
-Dxx Pattern row display (0 = decimal, 1 = hexadecimal)
-Exx Set emulated SID model (0 = 6581 1 = 8580) DEFAULT=6581
-Fxx Set custom SID clock cycles per second (0 = use PAL/NTSC default)
-Gxx Set pitch of A-4 in Hz (0 = use default frequencytable, close to 440Hz)
-Hxx Use HardSID (0 = off, 1 = HardSID ID0 2 = HardSID ID1 etc.)
-Ixx Set reSID interpolation (0 = off, 1 = on, 2 = distortion, 3 = distortion & on ←
    ) DEFAULT=off
-Kxx Note-entry mode (0 = PROTRACKER 1 = DMC) DEFAULT=PROTRK.
-Lxx SID memory location in hex. DEFAULT=D400
-Mxx Set sound mixing rate DEFAULT=44100
-Oxx Set pulse optimization/skipping (0 = off, 1 = on) DEFAULT=on
-Rxx Set realtime command optimization/skipping (0 = off, 1 = on) DEFAULT=on
-Sxx Set speed multiplier (0 for 25Hz, 1 for 1x, 2 for 2x etc.)
-Txx Set HardSID interactive mode sound buffer length in milliseconds
    DEFAULT=20, max.buffering=0
-Uxx Set HardSID playback mode sound buffer length in milliseconds
    DEFAULT=400, max.buffering=0
-Vxx Set finevibrato conversion (0 = off, 1 = on) DEFAULT=on
-Xxx Set window type (0 = window, 1 = fullscreen) DEFAULT=window
-Zxx Set random reSID write delay in cycles (0 = off) DEFAULT=off
-N    Use NTSC timing
-P    Use PAL timing (DEFAULT)
-W    Write emulated sound output to a file SIDAUDIO.RAW
-?    Show command line options
-??   Standalone online help window
```

Try the command line options if there are any problems. For example, if you experience choppy audio you can increase audio buffering with -B option. SDL seems to have trouble with some soundcards under Windows; you might want to try even a 500ms buffer, or tweak the hardware acceleration level of the soundcard (from Control Panel). Also, reSID interpolation will take remarkably more CPU time and could cause the sound and/or editing to get choppy.

Finevibrato mode is on by default. You can change this behaviour with -V option. Note that finevibrato mode only affects the conversion of old style vibrato parameters to speedtable format.

Pulse optimization -O determines if pulsetable execution will be skipped when

"

getting new notes or a new pattern. By default it is on. Be warned that turning it off will increase the rastertime requirements of packed/relocated songs greatly!

Realtime command optimization -R determines if portamento, vibrato & instrument vibrato will be skipped on tick 0 of each pattern row. By default it is on, as this is how tracker programs have operated traditionally. Turning it off makes the commands sound same regardless of tempo but will increase rastertime use.

The hard restart ADSR parameter will especially affect how rapid passages of notes will sound like. 0000 is probably too hard to be useful, except perhaps with gateoff timer value 1. 0F00 (default) is a lot softer, and 0F01 adds also a little bit of release to the gateoff phase for even softer sound. 000F makes the note start very pronounced.

Hard restart parameter with attack at maximum (F) enables the use of an alternative playroutine, where waveform is written before ADSR. This can give more reliable note triggering, especially for very fast releases 0 & 1, but may change the characteristics of the note's decay & release.

1.8 2.2 Hardware support

2.2 Hardware support

HardSID support is available with the /H option (use first HardSID = -H1, second = -H2 etc., return to emulated output = -H0). You must have the HardSID drivers installed to use this feature.

CatWeasel MK3 PCI SID support is available with -C option (-C1 to turn on).

To use the PC64 cable, you need Daniel Illgen's HardSID-DLL-Clone. Available at <http://dawork.synchronus.de/>

To enable better support of multispeeds and cycle-exact timing on HardSID (currently Win32 only), download an enhanced HardSID.dll:

For HardSID ISA/PCI cards -

http://sourceforge.net/project/showfiles.php?group_id=9266&release_id=61843

For HardSID 4U -

<http://www.hardsid.com>

Cycle-exact HardSID buffer length is separately configurable for interactive mode (-T option) & playback mode (-U option).

HardSID interactive mode:

Low latency mode for jamming and accurate GUI display. Interactive mode can be activated during song/pattern playback by enabling jamming (space key).

HardSID playback mode:

High stability mode to avoid interruptions caused by other applications/poor drivers/weaker CPU. GUI display is less accurate in this mode. Set the -U parameter to zero to achieve the maximum stability (less accurate GUI).

1.9 2.3 Keyboard commands

2.3 Keyboard commands

This program is mainly operated on keyboard. For a list of keyboard commands press F12 (online help) in the tracker or see the table below:

NOTE: SHIFT & CTRL are interchangeable in the commands. You can also use [] or () instead of < >.

1.10 2.3.1 General keys

2.3.1 General keys

F1	Play from beginning
F2	Play from current pos.
F3	Play one pattern from current pos.
F4	Stop playing & silence all sounds
F5	Go to pattern editor
F6	Go to song editor
F7	Go to instrument/table editor
F8	Go to songname editor
F9	Pack, relocate & save PRG,SID etc.
F10	Load song (Pattern/Song/Songname mode) or instrument (Instrument mode)
F11	Save song (Pattern/Song/Songname mode) or instrument (Instrument mode)
F12	Online help screen. Press F12 again to toggle context mode
SHIFT+F1	Play from beginning /w follow play
SHIFT+F2	Play from current pos. /w follow play
SHIFT+F3	Play one pattern from current pos. /w follow play
SHIFT+F4	Mute current channel
SHIFT+F5	Decrease speed multiplier
SHIFT+F6	Increase speed multiplier
SHIFT+F7	Edit hardrestart ADSR parameter
SHIFT+F8	Switch between 6581 and 8580 SID models
SHIFT+,.	Move song startposition on all channels and restart last playmode
TAB	Cycle between editing modes (forwards)
SHIFT+TAB	Cycle between editing modes (backwards)
INS	Insert row (Press INS/DEL on endmark to change pattern/song length)
DEL	Delete row
SHIFT+ESC	Optimize musicdata, or clear musicdata & set default pattern length
ESC	Exit program

1.11 2.3.2 Pattern edit mode

2.3.2 Pattern edit mode

- +	Select instrument
/ *	Select octave
< >	Select pattern

BACKQUOTE Select channel
 0-9 & A-F Enter parameters
 SPACE Switch between jam/editmode
 RETURN (also CAPSLOCK) Insert keyoff
 Enter table (when cursor is over a table-utilizing command)
 Enter instrument (when cursor is over a nonzero instrument number)
 SHIFT+RET. Insert keyon
 Convert portamento/vibrato/funktempo old style parameter to a
 speedtable entry (when cursor is over the command)
 Go into an empty table entry (when parameter is zero on a
 table-utilizing command)
 BACKSPACE Insert rest
 SHIFT+BKSP. Insert rest and clear commands
 SHIFT+SPACE Play from cursor position
 SHIFT+Q Transpose halfstep up
 SHIFT+A Transpose halfstep down
 SHIFT+W Transpose octave up
 SHIFT+S Transpose octave down
 SHIFT+O Shrink pattern (divide size by 2)
 SHIFT+P Expand pattern (multiply size by 2)
 SHIFT+J Join pattern with next pattern in orderlist
 SHIFT+K Split pattern from edit position
 SHIFT+H Calculate "hifi" left/right shifted speedtable entry for
 portamento or vibrato
 SHIFT+I Invert selection / whole pattern if no selection
 SHIFT+CRSR UP,DOWN Mark pattern
 SHIFT+CRSR LEFT,RIGHT Select pattern
 SHIFT+L Mark/unmark entire pattern
 SHIFT+M,N Choose highlighting step size
 SHIFT+X,C,V Cut,copy,paste pattern
 SHIFT+E,R Copy,paste commands
 SHIFT+Z Cycle autoadvance-mode

For pattern joining to work properly, select the pattern you want to join with the next from the orderlist first (by pressing RETURN on it). Both joining and splitting will rearrange pattern numbers (for example if you split pattern 00, pattern 01 will be created and old patterns 01-xx will be shifted forward in the pattern order)

SHIFT+SPACE will play from the pattern onwards if playback was previously stopped, but if it is pressed while in pattern play mode, it will stay in the current pattern.

There are 2 modes for note entering:

1.12 2.3.2.1 Protracker note-entry mode

2.3.2.1 Protracker note-entry mode

This is the default or activated with command line option /K0. There are two rows of a piano keyboard:

Lower octave	Higher octave
S D G H K L	2 3 5 6 7 9 0

Z X CV B NM , . Q W ER T Y UI O P

Octave (0-7) is selected with / and * keys on the numeric keypad.

In this mode there are 2 different autoadvance-modes (the mode can be seen from the color of the jam/editmode indicator)

GREEN - Advance when entering notes & command-databytes
 RED - Do not advance automatically

1.13 2.3.2.2 DMC note-entry mode

2.3.2.2 DMC note-entry mode

Activated with command line option /K1, there is one row of piano keyboard

W E T Y U O P
 A S DF G H JK L

and octave of a note (sets default octave at the same time) is changed with number keys 0-7.

In this mode there are 3 different autoadvance-modes:

GREEN - Advance when entering notes, octaves or command-databytes
 YELLOW - Advance when entering notes or command-databytes, not octaves
 RED - Do not advance automatically

1.14 2.3.3 Song edit mode

2.3.3 Song edit mode

< > Select subtune
 - + Insert Transpose down/up command (shown as -/+ in the orderlist)
 0-9 & A-F Enter pattern numbers
 SPACE Set start position for F2 key
 BACKSPACE Set end position for F2 key
 RETURN Go to pattern
 SHIFT+R Insert Repeat command (shown as "R" in the order-list)
 SHIFT+CRSR LEFT/RIGHT Mark orderlist
 SHIFT+1,2,3 Swap current channel orderlist with channel 1,2,3
 SHIFT+X,C,V Cut,copy,paste channel orderlist
 SHIFT+L Mark/unmark entire orderlist
 SHIFT+SPACE Set start position on all channels
 SHIFT+RET. Go to pattern on all channels

SHIFT+SPACE or SHIFT+BACKSPACE set the start/end mark on all channels to the same position. To clear the endmark, press BACKSPACE again on it or at/before the startmark.

1.15 2.3.4 Instrument edit mode

2.3.4 Instrument edit mode

```

-----
< >          Select instrument
- +          Select instrument
/ *          Select octave
F7           Go to table editor
0-9 & A-F    Enter parameters
SPACE        Play test note
SHIFT+SPACE  Silence test note
RETURN       Go to table position indicated by wave/pulse/filter/vibratopos.
SHIFT+N      Edit instrument name
SHIFT+S      Smart paste an instrument
SHIFT+U      Unlock/lock table scrolling
SHIFT+X,C,V  Cut,copy,paste instrument
SHIFT+DEL    (also SHIFT+BACKSPACE) Delete instrument & tabledata
SHIFT+RET.   Set tablepointer to empty position or convert old style vibrato
              parameter

```

The test note will be played on the channel you last were on in the pattern editor. To hear filtering as intended, be sure to play it on a channel that has been selected for filtering in the filter parameters.

Note that cut,copy,paste instrument do not touch the tabledata, just for the case you need it in another instrument. If you want to completely get rid of an instrument + its associated tabledata, press SHIFT+DEL.

If wave/pulse/filter/vibrato param. is zero and you press RETURN over it, you will move to the first free location in the corresponding table. If you press SHIFT+RETURN, also the instrument parameter will be set accordingly.

If vibrato parameter is nonzero and you press SHIFT+RETURN over it, it will be interpreted as an old style vibrato parameter (left nybble speed, right nybble depth) and converted to a speedtable entry.

"Smart paste" will convert instrument numbers in all patterns if you "move" an instrument by cut/pasting it.

1.16 2.3.5 Table edit mode

2.3.5 Table edit mode

```

-----
< >          Select instrument
- +          Select instrument
/ *          Select octave
BACKQUOTE    Select table
F7           Go to instrument editor
0-9 & A-F    Enter parameters
SPACE        Play test note
SHIFT+SPACE  Silence test note
RETURN       Go back to wave/pulse/filterpos. parameter

```

SHIFT+Q,A	Transpose speedtable portamento speed halfstep up/down
SHIFT+W,S	Transpose speedtable portamento speed octave up/down, or multiply/divide pulse/filterspeed by 2
SHIFT+L	Convert <limit,speed> modulation step to <time,speed>
SHIFT+N	Negate speed parameter (pulse/filtertable) or relative note (wavetable)
SHIFT+O	Optimize current table (remove unused entries)
SHIFT+R	Convert note between absolute/relative in wavetable
SHIFT+U	Unlock/lock table scrolling

If you need to insert rows in the beginning of an instrument's wave/pulse/filtertable, press SHIFT+INS instead of just INS: this way table pointers pointing to the table first row will not move.

For easier programming of negative modulation speeds in pulse/filtertables, you can first enter a positive speed positive value (\$00-\$7F) and then press SHIFT+N to negate it.

Conversion between relative and absolute notes will use the basenote C-0 if octave is 0, C-1 if octave is 1 and so on.

When table scrolling is unlocked, each table maintains its separate view position. To indicate this mode, a "U" letter is visible in the bottom right part of the screen. By default table scrolling is locked.

1.17 2.3.6 Songname edit mode

2.3.6 Songname edit mode

Use cursor UP/DOWN to move between song, author & copyright strings, and other keys to edit them.

1.18 2.4 Mouse control

2.4 Mouse control

By clicking with the left mouse button, you can select the data to edit.

By dragging with middle or right mouse button, you can mark orderlist/pattern/taledata.

Some values like octave or subtune/pattern numbers can be incremented with the left mouse button and decremented with the right.

By holding the middle/right mouse button in the status bar, a menu becomes active. Continue to hold and click with left button to select options.

In the fileselector, confirm loading or entering a directory by double-clicking. Doubleclick outside the selector to cancel.

1.19 3. Song data

3. Song data

1.20 3.1 Orderlist data

3.1 Orderlist data

A song can consist of up to 32 subtunes. For each subtune's each channel, there is an orderlist which determines in what order patterns are to be played. In addition to pattern numbers, there can be TRANSPOSE & REPEAT commands and finally there is a RST (RESTART) endmark followed by restart position. The maximum length of an orderlist is 254 pattern numbers/commands + the endmark.

TRANSPOSE is measured in halftones. Transpose up (shown as +X) can be 0-14 halftones and transpose down (shown as -X) can be 1-15. Transpose is automatically reset only when starting the song, not when looping.

A REPEAT command (shown as RX) will repeat the pattern following it 1-16 times. "Repeat 16 times" is displayed as R0.

There are some rules for orderlist command order:

- If there are both TRANSPOSE and REPEAT commands before a pattern number, TRANSPOSE must come first.
- The last thing before the RST-endmark must be a pattern number. If you need to reset transpose on song repeat, do it in the beginning of the repeat loop.

In case of wrong order, the editor will halt playback. This tells that the resulting packed/relocated song would play incorrectly.

1.21 3.2 Pattern data

3.2 Pattern data

Patterns are single-channel only for flexibility & low memory use. They contain the actual notes, instrument changes & sound commands. A pattern can have variable length, up to 128 rows. There can be 208 different patterns in a song.

The explanation of a pattern row:

```

Note name
|
| Octave
| |
| | Instrument number ($01 - $3F, or $00 for no change)
| | |
| | | Command ($0 - $F)

```



```

| | | |
| | | | Databyte
| | | | |
C-1 00 0 00

```

The highest note available in a pattern is G#7. To reach the top three notes (A-7 to B-7), you can use transpose.

In place of a normal note, there can also be one of these special "notes":

```

... Rest
--- Key off (clear gatebit mask)
+++ Key on (set gatebit mask)

```

The actual state of the gatebit will be the gatebit mask ANDed with data from the wavetable. A key on cannot set the gatebit if it was explicitly cleared at the wavetable.

Commands 1XY-4XY and FXY bear some resemblance to Soundtracker/Protracker/Fasttracker effect commands. However, they are different in some ways, so read their descriptions! Note that there is no "databyte \$00 uses the last databyte"-action in the commands.

Command 0XY: Do nothing. Databyte will always be \$00.

Command 1XY: Portamento up. XY is an index to a 16-bit speed value in the speedtable.

Command 2XY: Portamento down. XY is an index to a 16-bit speed value in the speedtable.

Command 3XY: Toneportamento. Raise or lower pitch until target note has been reached. XY is an index to a 16-bit speed value in the speedtable, or \$00 for "tie-note" effect (move pitch instantly to target note)

Command 4XY: Vibrato. XY is an index to the speed table, where left side determines how long until the direction changes (speed) and right side determines the amount of pitch change on each tick (depth).

Command 5XY: Set attack/decay register to value XY.

Command 6XY: Set sustain/release register to value XY.

Command 7XY: Set waveform register to value XY. If a wavetable is actively changing the channel's waveform at the same time, will be ineffective.

Command 8XY: Set wavetable pointer. \$00 stops wavetable execution.

Command 9XY: Set pulsetable pointer. \$00 stops pulsetable execution.

Command AXY: Set filtertable pointer. \$00 stops filtertable execution.

Command BXY: Set filter control. X is resonance and Y is channel bitmask. \$00 turns filter off and also stops filtertable execution.

Command CXY: Set filter cutoff to XY. Can be ineffective if the filtertable is active and also changing the cutoff.

Command DXY: Set mastervolume to Y, if X is \$0. If X is not \$0, value XY is copied to the timing mark location, which is playeraddress+\$3F.

Command EXY: Funktempo. XY is an index to the speedtable, tempo will alternate between left side value and right side value on subsequent pattern steps. Sets the funktempo active on all channels, but you can use the next command to override this per-channel.

Command FXY: Set tempo. Values \$03-\$7F set tempo on all channels, values \$83-\$FF only on current channel (subtract \$80 to get actual tempo). Tempos \$00-\$01 recall the funktempo values set by EXY command.

Master volume is by default the maximum (\$F), but it is only reset when loading a new song or clearing songdata in the editor, not every time playback starts. If you change mastervolume, you have to reset it manually in the beginning of your song.

If the command is not 1XY-4XY, instrument vibrato will be active.

Note that the one-shot commands 5XY-FXY allow the previous 1XY-4XY command or instrument vibrato to continue "underneath" them. In section 3.6 (hints & tips) there is an example of this.

1.22 3.3 Instrument data

3.3 Instrument data

You can use up to 63 different instruments in a song. Each instrument is defined by 9 parameters:

Attack/Decay	\$0 is fastest attack or decay, \$F is slowest
Sustain/Release	Sustain level \$0 is silent and \$F is the loudest. Release behaves like Attack & Decay (F slowest).
Wavetable Pos	Wavetable startposition. Value \$00 stops the wavetable execution and is not very useful.
Pulsetable Pos	Pulsetable startposition. Value \$00 will leave pulse execution untouched.
Filtertable Pos	Filtertable startposition. Value \$00 will leave filter execution untouched. In most cases it makes sense to have a filter-controlling instrument only on one channel at a time.
Vibrato Param	Instrument vibrato parameter. An index to the speedtable, see command 4XY.
Vibrato Delay	How many ticks until instrument vibrato starts. Value \$00

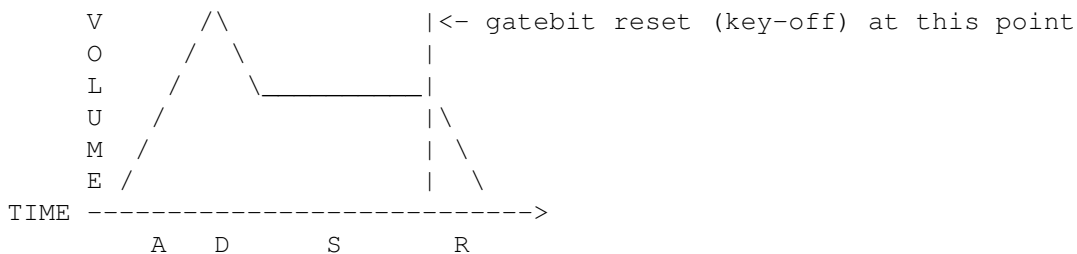
turns instrument vibrato off.

HR/Gate Timer How many ticks before note start note fetch, gateoff and hard restart happen. Can be at most tempo-1. So on tempo 4 highest acceptable value is 3. Bitvalue \$80 disables hard restart and bitvalue \$40 disables gateoff.

1stFrame Wave Waveform used on init frame of the note, usually \$09 (gate + testbit). Values \$00, \$FE and \$FF have special meaning: leave waveform unchanged and additionally set gate off (\$FE), gate on (\$FF), or gate unchanged (\$00).

In case of illegal (too high) gateoff timer values, the song playback is stopped.

ADSR settings are crucial to getting any sound at all. If all of them are zero just a very short "click" will be heard. Here is a diagram to help you visualize the Attack, Decay, Sustain & Release phases:



Some ADSR examples:

A/D 09 Will produce a sound that starts from full volume right away and fades S/R 00 to silence automatically. By increasing the Decay value, the fade will last longer.

A/D 00 A sound that goes very fast from full volume to sustain level 8. S/R 8A If you increase the Decay value, it will go to the sustain level slower. After key-off, starts fading out with speed A.

A/D CC A sound that rises slowly to maximum volume, then decays slowly to the S/R AF sustain level A and after key-off, fades out to silence very slowly.

Instrument legato works as following: When HR/Gate Timer parameter has bit \$40 set, no hard restart or gateoff will be performed. When 1st Frame Wave is also \$00, no 1st frame waveform will be set and gate flag is untouched. However wave/pulse/filterpointers and ADSR are initialized normally. You can also use this in conjunction with command 8XY to set another wavetable pointer for the note.

1.23 3.4 Table data

3.4 Table data

Tables control the execution of instruments' waveform/arpeggio changes, pulse modulation, and filter modulation. All the tables are controlled by the left side bytes, while the right side byte specifies additional parameters.

Note that you should never jump directly onto a table jump command (FF) either with instrument parameters or pattern commands 8XY, 9XY, AXY. Otherwise, results are undefined.

1.24 3.4.1 Wavetable

3.4.1 Wavetable

```

Wavetable left side:  00      Leave waveform unchanged
                    01-0F Delay this step by 1-15 frames
                    10-DF Waveform values
                    E0-EF Inaudible waveform values $00-$0F
                    F0-FE Execute command 0XY-EXY. Right side is parameter.
                    FF      Jump. Right side tells position ($00 = stop)

Wavetable right side: 00-5F Relative notes
                    60-7F Negative relative notes (lower pitch)
                    80      Keep frequency unchanged
                    81-DF Absolute notes C#0 - B-7

```

Short explanation of waveform bitvalues:

```

01 = Gatebit. When on, initiates attack/decay/sustain phase. When off,
    initiates the release phase.
02 = Synchronize. Creates weird effects using output of another channel
04 = Ring modulation. Creates weird effects using output of another channel.
    Most effective with the triangle waveform.
08 = Testbit. Silences sound and resets the oscillator.
10 = Triangle waveform.
20 = Sawtooth waveform.
40 = Pulse waveform.
80 = Noise waveform.

```

All waveforms except noise can be combined (for example triangle+pulse), but the effect will be different on 6581 and 8580 SID chips, so use caution.

The way how the channels work with synchronize/ringmod:

- When used on channel 1, channel 3's output modulates the sound.
- When used on channel 2, channel 1's output modulates the sound.
- When used on channel 3, channel 2's output modulates the sound.

Wavetable delay or no wavechange should not be used in the first step of instrument wavetable. Otherwise, missing notes may be caused. On the other hand, if you use 8XY command to jump into a wavetable program, those are allowed.

Using wavetable delay or a wavetable step with no frequency change allows realtime commands & instrument vibrato to be executed together with wavetable. Be warned that this has the potential for large rastertime usage!

You can execute pattern commands from the wavetable. Right side is the command parameter. Note that commands 0XY (do nothing), 8XY (set wavetable pointer) and EXY (funktempo) are illegal and should not be used. When executing a command, no wave/note will be changed on the same frame.

Some examples of wave tables (all examples start on table step 1)

```
01: 21 00 Sawtooth waveform on note's original pitch.
02: FF 00
```

```
01: 41 00 A flute sound with pulse on the first tick and triangle on all
02: 11 00 the rest.
03: FF 00
```

```
01: 41 01 A "koto" sound that is one halfstep higher on the first tick and
02: 40 00 on original pitch the next. Gatebit is also cleared on the second
03: FF 00 tick.
```

```
01: 81 D0 A snaredrum sound, using all absolute notes so it does not depend
02: 41 AA on which note it's played. Use pulsewidth 800 for best result.
03: 41 A4
04: 80 D4
05: 80 D1
06: FF 00
```

```
01: 81 DF A pulse sound on original pitch, preceded with a short noise (like
02: 41 00 a hi-hat or something) that has always an absolute pitch of B-7.
03: FF 00
```

```
01: 41 00 A 4-note looping arpeggio sound with pulse waveform. Note that
02: 00 04 waveform does not change in the looping part.
03: 00 07
04: 00 0C
05: 00 00
06: FF 02
```

```
01: 21 00 A delayed minor chord arpeggio with sawtooth waveform. Each step
02: 02 03 takes 3 ticks.
03: 02 07
04: 02 00
05: FF 02
```

```
01: 41 00 Use pulse first, but then switch between pulse & triangle every 5
02: 03 80 ticks while frequency remains unchanged (allowing for vibrato &
03: 11 80 slides).
04: 03 80
05: 41 80
06: FF 02
```

```
01: 81 CF Short noise in the beginning, sawtooth, after a short delay change
02: 21 00 sustain/release (via pattern command 6XY) to $2A.
03: 08 00
04: F6 2A
05: FF 00
```

1.25 3.4.2 Pulsetable

3.4.2 Pulsetable

Pulsetable left side: 01-7F Pulse modulation step. Left side indicates time and right side the speed (signed 8-bit value).
 8X-FX Set pulse width. X is the high 4 bits, right side tells the 8 low bits.
 FF Jump. Right side tells position (\$00 = stop)

Some examples of pulse tables (all examples start on table step 1)

```
01: 88 00 Set pulse value $800 (middle)
02: FF 00 Stop pulse execution
```

```
01: 80 10 Set pulse value $010 (very thin)
02: 20 40 For 32 ticks, increase pulse with speed $0040 (64)
03: 40 E0 For 64 ticks, decrease pulse with speed $FFE0 (-32)
04: 40 20 For 64 ticks, increase pulse with speed $0020 (32)
05: FF 03 Jump back to step 03 for a nice loop
```

1.26 3.4.3 Filtertable

3.4.3 Filtertable

Filtertable left side: 00 Set cutoff, indicated by right side
 01-7F Filter modulation step. Left side indicates time and right side the speed (signed 8-bit value)
 80-F0 Set filter parameters. Left side high nybble tells the passband (\$90 = lowpass, \$A0 = bandpass etc.) and right side tells resonance/channel bitmask, as in command BXY.
 FF Jump. Right side tells position (\$00 = stop)

If "Set filter parameters" is followed by "Set cutoff" directly below, both will be executed on the same frame.

Some examples of filter tables (all examples start on table step 1)

```
01: 90 F1 Set lowpass, resonance F, channel bitmask 1 (filter channel 1 only)
02: 00 40 Set cutoff to $40
03: FF 00 Stop filter execution
```

```
01: 80 00 No passband selected, resonance 0, bitmask 0 (no filtered channels)
02: FF 00 Stop filter execution
```

```
01: A0 87 Set bandpass, resonance 8, channel bitmask 7 (filter all channels)
02: 00 00 Set cutoff to $00
03: 7F 01 Increase cutoff with speed $01 for 127 ticks
04: 7F 01 Continue cutoff increase
05: 7F FF Then decrease back to starting position (speed $FF = -1)...
06: 7F FF
07: FF 03 ...and loop back to step 03
```

```
01: C0 F2 Set highpass, resonance F, channel bitmask 2 (filter channel 2)
```

```

02: 00 F0 Set cutoff to $F0
03: 90 F2 On the next frame, change to lowpass...
04: 00 50 ...and set cutoff $50
05: FF 00 Stop filter execution

```

Note that the second example could also be achieved simply with pattern command B00 (set filter control, and stop filter execution, because parameter was \$00)

1.27 3.4.4 Speedtable

3.4.4 Speedtable

The speedtable is shared by vibrato, portamento and funktempo. No jump commands exist.

For vibrato: XX YY Left side tells how long until vibrato direction changes (speed), right side is the value added to pitch each tick (depth).

For portamento: XX YY A 16-bit value added to pitch each tick. Left side is the MSB and the right side the LSB.

For funktempo: XX YY Two 8-bit tempo values that are alternated on each pattern row, starting from the left side.

For both vibrato and portamento, if XX has the high bit (\$80) set, note independent vibrato depth / portamento speed calculation is enabled, and YY specifies the divisor (higher value -> lower result and more rastertime taken).

Vibrato examples:

```

01: 03 40 Vibrato with speed $03 and depth $40. Corresponds to old vibrato
      parameter $34
01: 05 04 Speed $05 and depth $04. Slow shallow vibrato, suitable for bass
      notes. Not possible to achieve with old parameters.
01: 83 04 Speed $03, note-independent depth enabled, depth divisor 4 right-
      shifts (division by 16).

```

Portamento examples:

```

01: 00 20 Portamento with speed $0020, corresponds to old parameter $08
      (4 * $08 = $0020).
01: 01 00 Portamento with speed $0100, corresponds to old parameter $40
      (4 * $40 = $0100).
01: 4F FF Portamento with speed $4fff, loops the pitch highbyte very fast
      so that a SEUCK-like "space" effect is heard. Not possible to
      achieve with old parameters.
01: 80 01 Note-independent speed enabled, speed divisor 1 rightshift
      (division by 2).

```

Funktempo examples:

```

01: 09 06 Play tempo $09 on even pattern steps and $06 on odd steps,
      corresponds to old parameter $96.

```

01: 24 18 The same funktempo adjusted for 4x-multispeed. Not possible to achieve with old parameters.

Note that you can enter old style vibrato parameters, portamento speeds and funktempo to patterns & instruments, and when cursor is over the parameter, press SHIFT+RETURN to create a new speedtable entry corresponding to that parameter. The parameter will now be overwritten by a speedtable pointer.

1.28 3.5 Playback details

3.5 Playback details

Each pattern row is divided into as many 50Hz/60Hz "ticks" as the tempo indicates. Some ticks are reserved for special actions, and to conserve raster-time, certain realtime effects/commands (pulse, vibrato, portamento) are skipped at the same time. Let's assume a tempo of 6 and gateoff timer value 2 and look at what happens on each tick:

Tick	Actions
0	<ul style="list-style-type: none"> - Initialization of new notes (no audible sound yet) - Orderlist advance if necessary - Pulsetable execution only if no orderlist advance - Wavetable execution - "One-shot" commands 5XY-FXY
1	<ul style="list-style-type: none"> - New notes become audible - Pulsetable execution - Wavetable or realtime pattern commands 1XY-4XY
2,3	<ul style="list-style-type: none"> - Pulsetable execution - Wavetable or realtime pattern commands 1XY-4XY
4	<ul style="list-style-type: none"> - New notes fetched from the pattern - Gateoff and hard restart for new notes (2 ticks before first frame, as gateoff timer indicates) - No pulsetable execution - Wavetable or realtime pattern commands 1XY-4XY
5	<ul style="list-style-type: none"> - Pulsetable execution - Wavetable or realtime pattern commands 1XY-4XY

Filtertable is executed on each tick regardless of what the channels are doing. Wavetable is never skipped, so arpeggios/drumsounds should always play OK.

With the commandline parameter /O0 (or by clicking the "PO" text on the status bar) you can disable pulse skipping. This leads to increased rastertime use but can be good for example with tempo 3 tunes.

Likewise, with the commandline parameter /R0 (or by clicking the "RO" text on the status bar) you can disable realtime pattern command skipping on tick 0.

1.29 3.6 Miscellaneous tips

3.6 Miscellaneous tips

- Patterns will take less memory the less there are command changes. When the song is packed/relocated, for example a long vibrato or portamento command needs to be stored only once as long as the parameter stays the same on subsequent pattern rows.
- Using instrument vibrato can make 4XY commands unnecessary and save even more memory. Use realtime calculated note-independent speed (see speedtable section) if you want the vibrato to have the same magnitude regardless of the note.
- In the instrument parameters, there is no option to stop pulse execution directly. So if you have for example a sawtooth instrument, there might be an "unnecessary" pulsemodulation going on underneath and wasting rastertime. There are two ways you can stop this:
 - 1) Make a short pulseprogram like this and use it in the instrument:


```
01: 80 00 Set pulse $000
02: FF 00 Stop pulse execution
```
 - 2) Use pattern command 900 to stop pulse execution
- If you have a realtime command (vibrato/portamento), you can put "one-shot" commands 5XY-FXY inbetween and the realtime command will be unaffected. An example:

```
C-4 01000
--- 00401 Begin vibrato with speedtable index $01
--- 00401
--- 00401
--- 0065A Set sustain/release to $5A. Vibrato executes also on this step!
--- 00401 Continue vibrato normally
--- 00401
```

However, the vibrato would stop immediately if command 0 was encountered.

- Normally tempo 3 is the fastest you can use. However, by using the funktempo command you can get tempo 2. To do this, create a following speedtable entry: 02 02, and point your funktempo command to it. You also need to use gateoff timer 1 in all instruments and disable the pulse-optimization skipping.
- For very optimized songdata & player you can refrain from using any pattern commands and rely on the instruments' step-programming. Even in this case, you can set song startup default tempo with the Attack/Decay parameter of the last instrument (63), if you otherwise leave this instrument unused.

1.30 3.6.1 Limit-based modulation steps

3.6.1 Limit-based modulation steps

You can enter pulse/filter modulation steps in limit-based format, and convert

them to time-based with SHIFT+L. For example, you could enter the following in the pulsetable:

```
01: 84 00 Set initial pulse to $400
02: C0 10 Modulate pulse to $C00 with speed $0010
03: 40 10 Modulate pulse back to $400 with speed $0010
```

Now position the cursor on step 02 and press SHIFT+L twice, to convert both of the modulation steps to usual time-based steps. If necessary, the steps will be expanded to several (if the resulting time parameter is over \$7F.)

Remember that the player only understands time-based steps and you always have to convert limit-based steps before they play correctly!

1.31 3.6.2 Precalculated note-independent hifi player portamento/vibrato

3.6.2 Precalculated note-independent "hifi player" portamento/vibrato

From v2.62 onwards, GoatTracker actually supports realtime calculated note independent portamento and vibrato speeds, see speedtable section for this. However, this makes the player bigger and much slower. You can still also use the old method of manually precalculating needed note-independent speeds with the SHIFT+H function, which this section is about.

The idea is to take the frequency difference between 2 adjacent notes in the frequency table and manipulate it by bit shifting left (multiply) or right (divide).

To use, you need a portamento or vibrato command in the pattern, and the note you wish to base the calculation on on the current pattern step or before it (before it only for toneportamento) The amount of shift is determined by the command parameter. A number in the left nybble means shift left and right nybble right shift. When you press Shift+H while the cursor is on the command, a speedtable entry is created, and the parameter is replaced with a speedtable index.

Some examples:

```
C-3 00110 Take the frequency difference between C-3 & C#3 and shift it left
      1 times (multiply by 2.)

E-4 00000
... 00000
G-4 00302 Take the frequency difference between E-4 & F-4 and shift it right
      2 times (divide by 4.)
```

1.32 3.7 Multispeed tips

3.7 Multispeed tips

- When making multispeed songs, remember to multiply your tempos and also the gateoff timer values! For example, gateoff timer is normally 2, but in a 2X-speed tune the same gateoff length would be achieved with value 4.
- If you want to keep vibrato the same compared to 1X-speed, you must multiply vibrato speed by the speed multiplier, and divide vibrato depth by the multiplier. Remember also to divide portamento and pulse/filter modulation speeds as necessary.
- When importing a v1.xx multispeed song, you also need to multiply the gateoff timers.
- In the instruments, using Attack 0 might result in a very silent first row of the wavetable. You could try increasing the attack, or adding one or more rows of E9 00 (testbit+gate) to the beginning of the wavetable.

1.33 4. Using the included utilities

4. Using the included utilities

1.34 4.1 INS2SND2.EXE

4.1 INS2SND2.EXE

INS2SND2.EXE converts GoatTracker v1.xx or v2 instruments (.INS-files) into sound effects, outputting the data as source code or binary.

Usage: INS2SND <instrumentfile> <sourcecodefile> <options>

Options:

-b output in binary

-c output in CovertScript format (deprecated)

Default output is C64 assembler (DASM) style source code

Look at section 6.3 or run the program without parameters to see the limitations of the sound effect system.

1.35 4.2 SNGSPLI2.EXE

4.2 SNGSPLI2.EXE

SNGSPLI2.EXE splits the patterns of a GoatTracker v2 song into smaller pieces for memory use optimization. It is comfortable to compose with large patterns but usually more efficient memory-wise to use small patterns. Remember! Always keep the original song because a pattern-splitting song is much harder to edit further.

Usage: SNGSPLI2 <source> <destination> [target len]

For example, if a pattern is 64 rows long and the target length is 16, it will be split into 4 pieces. A pattern that falls below 2 x target length, will not be split at all.

1.36 4.3 MOD2SNG.EXE

4.3 MOD2SNG.EXE

Dedicated exclusively to T.M.R, this program converts the pattern & orderlist data of 4-channel, 31-instrument MOD-files into GoatTracker .SNG files. You must choose one channel to leave out, and obviously instruments are not converted, except for their names. You can also specify a transpose in halfsteps.

Usage: MOD2SNG <mod> <sng> [channel] [transpose]
 [channel] is the channel to leave out (1-4), default 4
 [transpose] is the halfstep transpose added to notes, default 0

1.37 4.4 BETACONV.EXE

4.4 BETACONV.EXE

Converts GT v2 beta songs to new format used by GT v2 RC1 onwards. Old format had 47 instruments max. and new has 63 max. Additionally, depending on what beta version you used, you can choose to halve vibrato depths and pulse speeds to make them play correct in the current version. Warning: the conversion is irreversible, so have backups if you overwrite the old versions.

Usage: BETACONV <source> <destination> [vibdepth] [pulse]
 [vibdepth] decides whether to halve vibdepth (1=yes 0=no), default 0
 [pulse] decides whether to halve pulse speed (1=yes 0=no), default 0

1.38 5. Using the songs outside the editor

5. Using the songs outside the editor

Press F9 in the editor to enter the packer/relocator. Choose playroutine options, startaddress, zeropage address (need 2 consecutive locations) and file format (PRG/BIN/SID), then type the filename.

If you use patterns longer than 64 rows, there is the possibility that relocation fails because of too complex patterns. Each pattern row can be 0-4 bytes packed, and the total amount of bytes per one pattern may not exceed 256.

If wave/pulse/filtertables overflow past row 255 without a jump command or there are other serious errors, you get a warning screen and have to fix the error.

The relocater optimizes & removes when saving:

- Unused patterns
- Unused instruments
- Unused table-entries
- Selfcontained (ie. no jumps to/from outside) duplicate parts of tables
- Player code not needed in the playback of the song

Look at /examples/example1.prg - example4.prg to get an idea how much these playroutines take rastertime. No promises!

The basic functions independent of playroutine options are:

Init music:

```
LDA #subtunenumber      ;Starting from 0
JSR startaddress
```

Play one frame of music:

```
JSR startaddress+3
```

1.39 5.1 Playroutine options

5.1 Playroutine options

BUFFERED SID-WRITES - Instead of writing each SID value as soon as possible, waits for the end of the channel's execution on each frame and then writes the registers in one go. Improves sound stability but takes more memory & raster-time.

SOUND EFFECT SUPPORT - Adds a sound effect engine for use in games. Implies buffered writes. To use, call in the following way:

```
LDA #<effect           ;Start address of sound effect data
LDY #>effect
LDX #channel           ;0, 7 or 14 for channels 1-3
JSR startaddress+6
```

The sound effects have a hardcoded priority system based on their start addresses. A sound higher up in memory (bigger address) will never be interrupted with a sound lower in memory.

VOLUME CHANGE SUPPORT - Adds a routine in the player's jumptable to change the master volume level. To use, call in the following way:

```
LDA #volume            ;0-15
JSR startaddress+6 ;(startaddress+9 with sound effect support)
```

Note that the master volume command (DXY) also modifies this same location, so it will clash if you also change the volume manually.

STORE AUTHOR-INFO - Writes the "AUTHOR" string from songname editor to the location range (startaddress+\$20 - startaddress+\$3F). Also enables timing mark support. A timing mark is a mastervolume command (DXY) with parameter \$10-\$FF. This parameter will be copied into startaddress+\$3F when the command is encountered in playback.

USE ZEROPAGE GHOSTREGS - Instead of writing to the SID, a memory range in the zeropage is used for holding the SID register data. This can then be copied in the main program to the SID, for example to enable the playroutine & music data to reside under the I/O area. Implies buffered writes. A reverse copy loop should be used after each call to the playroutine:

```

copy:   LDX #$18
        LDA ghostregs,x
        STA $D400,x
        DEX
        BPL copy

```

The copy loop can also be unrolled for more speed.

Note that when you use the alternative hardrestart method & playroutine (hard-restart attack parameter F) the copy loop should not be in reverse order but should be for each channel: wave, frequency, pulsewidth, and ADSR last.

If you are using both sound effect support and ghostregisters, there exists a possibility of sound errors in the case that a sound effect has finished, and the same channel starts a new pulse-waveform note without setting an initial pulsewidth value (for example, like in the beginning of Sanxion loader tune). This is because the ghostregisters are shared between music & sound effect playback, and the sound effect has already overwritten the previous pulse width.

DISABLE OPTIMIZATIONS - The relocater removes saved playroutine code based on what effects & commands the song uses, resulting in a smaller playroutine accordingly. If you encounter anomalies in the sound (such as ADSR bugs caused by unpredictable timing variation) you can try disabling the optimizations. Normally this is not necessary, but is included just in case.

1.40 6. File/data formats description

6. File/data formats description

The sections in the files appear in the sequential order in which they are described.

1.41 6.1 GoatTracker v2 song (.SNG) format

6.1 GoatTracker v2 song (.SNG) format

1.42 6.1.1 Song header

6.1.1 Song header

Offset	Size	Description
+0	4	Identification string GTS5
+4	32	Song name, padded with zeros
+36	32	Author name, padded with zeros
+68	32	Copyright string, padded with zeros
+100	byte	Number of subtunes

1.43 6.1.2 Song orderlists

6.1.2 Song orderlists

The orderlist structure repeats first for channels 1,2,3 of first subtune, then for channels 1,2,3 of second subtune etc., until all subtunes have been gone thru.

Offset	Size	Description
+0	byte	Length of this channel's orderlist n, not counting restart pos.
+1	n+1	The orderlist data: Values \$00-\$CF are pattern numbers Values \$D0-\$DF are repeat commands Values \$E0-\$FE are transpose commands Value \$FF is the RST endmark, followed by a byte that indicates the restart position

1.44 6.1.3 Instruments

6.1.3 Instruments

Offset	Size	Description
+0	byte	Amount of instruments n

Then, this structure repeats n times for each instrument. Instrument 0 (the empty instrument) is not stored.

Offset	Size	Description
+0	byte	Attack/Decay
+1	byte	Sustain/Release
+2	byte	Wavepointer
+3	byte	Pulsepointer
+4	byte	Filterpointer
+5	byte	Vibrato param. (speedtable pointer)
+6	byte	Vibrato delay
+7	byte	Gateoff timer
+8	byte	Hard restart/1st frame waveform
+9	16	Instrument name

1.45 6.1.4 Tables

6.1.4 Tables

This structure repeats for each of the 4 tables (wavetable, pulsetable, filtertable, speedtable).

Offset	Size	Description
+0	byte	Amount n of rows in the table
+1	n	Left side of the table
+1+n	n	Right side of the table

1.46 6.1.5 Patterns header

6.1.5 Patterns header

Offset	Size	Description
+0	byte	Number of patterns n

1.47 6.1.6 Patterns

6.1.6 Patterns

Repeat n times, starting from pattern number 0.

Offset	Size	Description
+0	byte	Length of pattern in rows m
+1	m*4	Groups of 4 bytes for each row of the pattern: 1st byte: Notenumber Values \$60-\$BC are the notes C-0 - G#7 Value \$BD is rest Value \$BE is keyoff Value \$BF is keyon Value \$FF is pattern end 2nd byte: Instrument number (\$00-\$3F) 3rd byte: Command (\$00-\$0F) 4th byte: Command databyte

1.48 6.2 GoatTracker v2 instrument (.INS) format

6.2 GoatTracker v2 instrument (.INS) format

Offset	Size	Description
+0	4	Identification string GTI5
+4	byte	Attack/Decay

+5	byte	Sustain/Release
+6	byte	Wavepointer
+7	byte	Pulsepointer
+8	byte	Filterpointer
+9	byte	Vibrato param. (speedtable pointer)
+10	byte	Vibrato delay
+11	byte	Gateoff timer
+12	byte	Hard restart/1st frame waveform
+13	16	Instrument name

After the instrument data, this structure repeats for each of the 4 tables (wavetable, pulsetable, filtertable, speedtable).

Offset	Size	Description
+0	byte	Amount n of rows in the table
+1	n	Left side of the table
+1+n	n	Right side of the table

Note that the tables are a partial snapshot of the current song the instrument was saved from, so upon loading they have to be relocated to the current free table locations.

1.49 6.3 Sound effect data format

6.3 Sound effect data format

Offset	Size	Description
+0	byte	Attack/Decay
+1	byte	Sustain/Release
+2	byte	Pulse width. This value has nybbles reversed from what it looks like in the editor so a middle pulse \$800 will be stored as \$08, and the sound effect routine will put this value to both \$D402 and \$D403 registers.
+3	?	Wavetable. Contains note/waveform pairs (different order than in instrument wavetable), from which the waveform can be omitted if unchanged, as the value ranges do not overlap: Value \$00 ends the sound effect Values \$01-\$81 are waveforms Values \$82-\$DF are absolute notes D-0 - B-7 Note that a note cannot be omitted to store only waveform changes!

As you can see, the sound effect format is very simplistic. When converting an instrument to a sound effect with INS2SND2, following things cause an error message:

- If the resulting sound effect is more than 128 bytes
- If the instrument's wavetable contains relative notes, absolute notes C-0 or C#0, or waveforms > 129 (\$81)

The instrument's pulsewidth modulation & filter settings will be completely discarded.

1.50 7. Recompiling

7. Recompiling

To recompile for Win32, you need the MinGW development environment, use the file src/makefile.win as makefile.

To recompile for Linux, use src/makefile.

In both cases you need the SDL development libraries in addition to the SDL runtime, see <http://www.libsdl.org>.

Compile first the utilities (datafile & dat2inc) from the src/bme directory, and place them to your path.

1.51 8. Version history

8. Version history

v2.0Beta - Original public release

v2.0

- Fixed crash in DMC-note entry mode.
- Fixed vl.xx import pulse conversion, a case where pulse startpos was lower than the pulse low limit was not converted right.
- Fixed instrument gatetimer becoming zero (can lead to tempo bugs) when cutting an instrument (SHIFT+X)
- Fixed order of SID writes to make editor & C64 playroutine behave as similarly as possible.
- Fixed removal of table-entries (when loading an instrument) in case another instrument uses them too.
- Fixed toneportamento (3XY) with wavetable delays.
- Fixed funktempo in packed/relocated songs.
- Fixed behaviour of + & - keys while editing instrument name.
- Fixed pattern editor transpose functions.
- Added SHIFT+DEL to delete an instrument + its tabledata.
- Added SHIFT+INS to insert rows on table first row without having to adjust tablepointers afterwards.
- Added SHIFT+W,S for octave transpose.
- Added SHIFT+1,2,3 to swap orderlists between channels.
- Added SHIFT+BACKSPACE to clear a whole pattern row.
- Added optimized playroutines, that will be used if all instruments have the same gateoff timer & 1st wave parameters.
- Added duplicate check for vl.xx pulse conversion.
- Added BETACONV-utility for conversion from beta format.
- Added reSID interpolation option /I.
- Added hexadecimal pattern row display option /D.
- Added SID memory location parameter /L.
- reSID interpolation is on by default.
- Changed appearance of pattern special notes (rest, keyoff/on)
- Increased amount of instruments to 63 and changed song dataformat.
- vl.xx import converts arpeggios to instruments as long as there is

- room.
- Vibrato depth changed back to same as in v1.xx.
 - Pulse modulation speeds are doubled.
 - Upon startup, songdata erase, or importing v1.xx data, gatetimer will be set to 2 * multiplier as opposed to just 2.
 - Optimized size & speed of all playroutines (initialization, checking for new note fetch, pulsetable execution).
 - Wavetable delay or no-wavechange on first step of instrument is definitely unsupported now! Protection to allow this conflicted with 3XY command.
- v2.01 - Added under-IO gamemusic playroutine.
- v2.02 - Added RETURN when instrument tableparameter (wave/pulse/filter) is 0 to get you to the first free table location. SHIFT+RETURN will additionally also set the instrument tableparameter.
- Added "set mastervolume"-jump to gamemusic routines.
- Improved sound effect handling, when an effect is interrupted by another (less silence between sounds).
- v2.03 - Added execution of realtime commands & instrument vibrato during wavetable delay.
- Added wavetable right side value \$80 to keep frequency unchanged (as a consequence, the note C-0 will cause the same effect).
- Added relocater optimizations: all unnecessary data is stripped when packing/relocating.
- Playroutines size-optimized.
- Explanation of how different gatetimer values can lead to playback going out of sync (section 3.3).
- reSID interpolation is no longer on by default :)
- v2.04 - Added a questionable 25Hz mode (/S0)
- Execution of commands on wavetable delay is completely reworked and more consistent now.
- Note C-0 is usable again.
- v2.05 - Added instrument legato feature (Hardres/1stwave parameter \$00).
- ADSR writes moved farther away from wave writes in the standard playroutine noteinit, lessening possibility of ADSR bugs.
- Song initialization and pulsetable execution speed/sizeoptimized.
- v2.06 - Packer/relocater tolerates up to 256 bytes long patterns now.
- Optimized playroutine sizes.
- v2.07 - Fixed transpose reset with F2/F3 song start (should not happen).
- Fixed varying tempo on channels with F2/F3 song start.
- Added warning screen to packer/relocater if table execution overflows.
- Added writing of PAL/NTSC and 6581/8580 flags according to PSID V2 NG specification.
- -mwindows added to linker options for no displaying of DOS prompt window (in win32 version).
- v2.08 - Added finevibrato mode (/V command line option).
- Optimized playroutine sizes & speed.
- v2.09 - Fixed max.pattern length in Clear Musicdata operation.
-

- Added Minimal playroutine.
 - First wave value \$80 also acts as a proper legato instrument now.
 - Optimized playroutine sizes & speed.
 - SID chip type & timing (PAL/NTSC) displayed on top row.
- v2.1
- Added SHIFT+N for negating pulse/filter modulation speeds.
 - Added no-funktempo & no-octave0 optimizations to playroutines (a total of 8 sub-types for each playroutine).
 - Maximum speed is 16X now.
 - Separate song/instrument/packed song directories are remembered during session.
 - Pathname is displayed in the fileselector.
 - Filters ** and *.* display all files in the fileselector.
 - Song entered on commandline will be loaded at startup.
 - Song filename currently being edited is shown in the titlebar.
 - Currently edited song filename will be used as default in the "SAVE SONG" and "SAVE MUSIC+PLAYROUTINE" dialogs.
 - Instrument name will also be used as instrument filename as default in the "SAVE INSTRUMENT" dialog.
 - UPX used for compression of win32 executables.
 - Included a short reference of waveform bits (manual only).
 - Configuration file has clearer structure.
- v2.11
- Added SHIFT+S for smart instrument paste (converts instrument numbers in all patterns, after cut/pasting an instrument).
 - Added SHIFT+L which allows to write <limit,speed> pulse/filter modulation steps and then convert them to the proper <time,speed> format.
- v2.12
- Playroutine 1 has buffered SID-writes.
 - Octave 0 is not disabled in routines with sound FX (no matter what the relocater says...)
- v2.13
- Standard playroutine now both in unbuffered and buffered flavors.
- v2.15
- Added SHIFT+I for inverting current pattern selection / whole pattern.
- v2.16
- Fixed octave selection with / * for laptop keyboards.
 - Empty patterns referenced in the orderlist will be saved when saving a song.
 - Save dialog will reappear if writing the song/instrument/executable music failed.
 - Optimized handling of "packed rests" in the playroutines.
 - Added < > for instrument selection in instrument/table edit modes.
- v2.17
- Fixed initialization of instrument vibrato.
 - Playroutines size-optimized.
- v2.18
- Wavetable left side values changed. Delay is now \$01-\$0F and inaudible waveforms (register values \$00-\$0F) have been mapped to table values \$E0-\$EF. Old songs using values \$08-\$0F have to be manually converted!
- v2.2
- Added the speedtable for more precise control of vibrato, portamento and funktempo.
 - Added SHIFT+O to optimize the speedtable (remove unused entries).
-

- Added SHIFT+R to convert between absolute/relative notes in the wavetable.
 - Added SHIFT+RETURN in pattern/instrument editor to convert old style portamento, vibrato and funktempo parameters to speedtable entries.
 - SHIFT+R will paste several effect rows if there is data in pattern copybuffer.
 - Editing during Follow Play is possible.
 - Song and instrument format modified for 4 tables. Old 3-table data will be loaded but not saved anymore.
- v2.21
- Fixed NTSC CIA timer value for SID files.
 - Shift+E will copy several effect rows if pattern has been marked.
- v2.22
- Shift+N will also negate relative wavetable notes.
 - When converting oldstyle parameters to speedtable entries with SHIFT+RETURN, the speedtable view will shift to the new entry if one was created.
- v2.23
- Fixed SHIFT+INS in wavetable (inserts \$E9 \$00 now as it should).
 - Added SHIFT+U to unlock table scrolling.
 - Added SHIFT+F5/F6 to change speed multiplier.
- v2.24
- Added SHIFT+F7 to edit the hardrestart ADSR parameter.
 - Added SHIFT+F8 to switch between 6581 and 8580 SID models.
- v2.25
- Added detail to table overflow error message in relocater.
 - Added stop of playback if a tablepointer gets pointed directly to a jump (would bug in packed/relocated tune).
 - Added relocater error message for the above case.
 - Added relocater error message for illegal song restart position.
 - Not possible to save erroneous packed/relocated songs anymore (now that there is adequate error location).
 - Added SHIFT+,. to move song startposition on all channels and restart last playmode.
- v2.26
- Fixed relocater complaining of FF parameter in speedtable.
 - To increase stability, in multispeed mode relocater does not use the "same gatetimer/1st wave" optimization.
 - Changed Shift+O to optimize the currently edited table, not just speedtable.
 - Added an option to optimize everything with the SHIFT+ESC clear function. This clears unused patterns, instruments and table-entries.
- v2.27
- Fixed optimize function with different pattern lengths.
 - Added marking & copy/pasting orderlist data like patterns & tables.
- v2.28
- Added sort of mouse control.
 - Sound reinitialization when changing multiplier or other parameters is now faster.
- v2.29Beta
- Internal reorganization.
 - New assembler-based relocater that leaves out unused portions of player code.
 - Instead of choosing a player version, one can choose combinations of player features.
 - Relocater does not save selfcontained duplicate parts of tables.
-

- Added keys 0-9 & A-Z in the fileselector to move to the corresponding part of the filelist (if found).
 - Added SHIFT+H to calculate left/right shifted "hifi" speedtable values, according to the frequency of a note.
- v2.3
- Fixed songdata optimize function.
 - Fixed relocater with no-hardrestart instrument in slot 1.
 - Fixed relocater handling of effect C (Set cutoff).
- v2.31
- Fixed effect copy/paste.
 - Added possibility to disable pulseskipping with /O command line option. Warning: causes huge rastertime increase.
 - Tempo 2 is allowed. See the tips section (3.6) for details.
- v2.32
- Maximum pattern length increased to 128. Do not complain of "pattern too complex" errors :)
 - Optimized playroutine init in case of only 1 subtune.
- v2.33
- Fixed a bug where relocater was checking pattern 208 (illegal).
 - Optimized playroutine in case there is no keyoff/keyon used.
 - Optimized playroutine in case there are no pattern effects used.
 - Optimized playroutine channel variables in case there is no transpose, repeat, wavedelay and pattern effects.
 - If not using the last instrument (63), you can control song startup default tempo by using its Attack/Decay parameter.
- v2.34
- Added relocater optimization for using 1 or 2 channels only. To enable, you must use only channels 1 or 1,2 and not be using sound effect support or zeropage ghostregs.
 - Turning off buffered writes in relocater options works now even if sound effects or zeropage ghostregs are selected (they will be turned off too).
- v2.35
- Auto-creation of speedtable entries will always generate a new entry (except for song loading), duplicate values will be optimized by the relocater.
 - Pressing SHIFT+RETURN over a speedtable-using pattern command while the parameter is zero, goes to the next empty speedtable entry and sets the command parameter to point to that entry.
- v2.4
- Added one bit of accuracy to pulse modulation speed. A new song-format (identifier GTS4) reflects this, old songs will be converted upon loading.
 - Added possibility to execute pattern commands from the wavetable.
- v2.41
- Fixed player code assembly errors when using some pattern commands only from the wavetable.
 - Fixed player code bug with command B when command A is not used.
 - Fixed SHIFT+O removing table entries pointed to from wavetable pattern commands.
- v2.42
- Fixed frequency table size determination.
 - Changed behaviour of keyoff in pattern editing changed to use releasenote() like the instrument editor.
 - Added SHIFT+RETURN when parameter is zero to get you into an empty table entry for 8, 9 and A pattern commands as well.
 - Added adjustment of table views when changing instrument number
-

- in unlocked table scrolling mode.
- Added RETURN over nonzero instrument number (in patterns) to go editing that instrument.
- v2.43
- Added duplicate pattern detection/removal to song optimize.
 - Added pattern shrink/expand (SHIFT+O,P).
 - Added pattern join/split (SHIFT+J,K).
- v2.44
- Fixed editor playroutine to not execute tick n commands, when a command has already been executed through the wavetable.
- v2.45
- Added possibility to disable realtime pattern command optimization at tick 0 (/Rxx command line option or click the "RO" text on the statusbar) for tempo-independent portamento & vibrato.
 - To make room on the status bar for the realtime optimization status indicator, finevibrato & pulse optimization indicators have been shortened to "FV" and "PO".
- v2.46
- Added transpose of speedtable portamento speeds (SHIFT+Q,A,W,S) SHIFT+W,S (octave transpose, multiply/divide by 2) also work in pulse & filtertable.
 - When editing the wavetable, RETURN & SHIFT+RETURN work for wavetable executed commands as in patterns (go to table position pointed by command, create new speedtable entries)
- v2.47
- Added experimental random delay to write of reSID registers, in order to make potential ADSR-bugging on C64 audible. The amount of delay in cycles can be adjusted with /Z command line option.
 - SHIFT+L in pulsetable creates repeated "set tempo" commands if speed is greater than 127.
- v2.48
- Fixed playroutine to use configurable baseaddress again.
- v2.49
- Added /G command line option, guard 1stwave/gatetimer parameters. When this is on (default), only silent waveforms can be entered as firstwave (only thing that makes sense) and editing gatetimer of one instrument affects all of them, as otherwise bugs in playback result easily. Use /G0 to turn off.
- v2.5
- Fixed "optimize everything" removing the tempo setting in instrument 63.
 - Fixed default tempo of multispeed tunes in packer/relocator.
- v2.51
- Fixed packing of empty patterns when not using any effects.
 - Fixed differing gatetimer value in instruments causing playback going out of sync.
 - /G command line option no longer has to keep gatetimer values the same.
 - Added high bit of gatetimer value to control whether wavetable execution starts right on the note init frame. If used, will cause significant rastertime increase.
- v2.52
- SHIFT+ENTER in orderlist view takes the next available pattern if pressed on a repeat or transpose command.
 - Added SHIFT+SPACE in pattern editor to start playback on a specific pattern row.

-
- v2.53
 - Fixed F3 to always play the currently visible patterns.
 - Added BACKSPACE in the orderlist editor to set playback end position. Only effective when starting playback from position, not beginning.
 - v2.54
 - Fixed SHIFT+F3.
 - Changed SHIFT+, . to update the pattern view.
 - Changed SHIFT+SPACE to remain in pattern playback mode, if it was active.
 - v2.55
 - Added BACKSPACE in the fileselector to go to the parent dir.
 - v2.56
 - Song filename cleared on songdata erase.
 - Added /F to set custom SID clock frequency.
 - Graphics output routines no longer compare text screen buffer to previous to find out if it should be updated.
 - v2.57
 - Added alternative hardrestart method & playroutine that is used when HR attack parameter is at maximum (FF00 or F800 for example) - this can in theory give better reliability.
 - v2.58
 - Cleanup, removal of a few questionable features:
 - * Gatetimer high bit to start wavetable immediately
 - * Guard 1stwave parameter
 - v2.59
 - Songformat version update (GTS5/GTI5)
 - Gateoff timer parameter bits control now hardrestart disable (\$80) & gateoff disable (\$40).
 - 1stFrame Wave can be any from \$01-\$8F, \$00 to leave both waveform & gateflag unchanged, or \$FE / \$FF to control gateflag but leave waveform unchanged.
 - v2.6
 - Fixed pattern default length selection display when decrementing from a length of 100 or higher.
 - Fixed mouse selection of pattern when adjusting an adjacent channel.
 - Fixed help screen instructions.
 - Changed resolution to 800x600.
 - Changed all songname rows to be displayed at the same time.
 - Changed mouse control to allow pattern column selection even when left mouse button is held down.
 - Changed speed of PageUp/PageDown scrolling to be faster.
 - Optimized graphics output.
 - BME library is no longer needed.
 - v2.61
 - Added the backquote key (top-left on keyboard) to select channel in pattern edit mode, and to select table in table edit mode. Use with SHIFT to go backwards.
 - Added SHIFT+channel number to mute channels in pattern edit mode.
 - v2.62
 - Added possibility for realtime calculated note independent (hifi) portamento & vibrato. Warning: has potential for huge rastertime increase.
 - v2.63
 - Fixed note independent portamento & vibrato to use the last note set in wavetable for calculations, instead of the last note in patterndata.
-

-
- v2.64 - Fixed paste in table (SHIFT+V) working also without SHIFT pressed.
 - v2.65 - Fixed raw keycodes over 511 interpreted as some other keys in the 0-511 range.
 - v2.66Beta - Initial cycle-exact HardSID support (Win32 only)
 - Permit running without sound.
 - v2.67 - Configurable cycle-exact HardSID buffer length (separate for interactive and playback mode, see /T and /U command line options)
 - v2.68 - Fixed set tempo -command overwriting frequencytable in 1 or 2 channel modes.
 - Fixed sound uninit crash with multicore processors (?)
 - SID register write order tweaked to resemble JCH NewPlayer 21.
 - Unbuffered playroutine optimized & modified to resemble buffered mode timing more.
 - New reSID-fp engine (with distortion & nonlinearity) from Antti Lankila integrated. Activated with /I command line option parameters 2 & 3.
 - Command quick reference by Simon Bennett included.
 - v2.69 - Fixed click bug in reSID audio output.
 - Newest reSID-fp code integrated.
 - reSID-fp filter parameters adjustable from the configuration file.
 - v2.70 - Hotfix for the new HardSID UPlay/Uno devices from Téli Sándor.
 - Fixed possible crash on some versions of the HardSID dll.
 - v2.71 - Added keycode fix patch from Valerio Cannone.
 - Added fullscreen switch patch from Raine M. Ekman (see /X option)
 - Added context mode to online help patch from Raine M. Ekman.
-