

STOP

POWER

IT

2.1.1

USER MANUAL

TABLE OF CONTENTS

Background	3
The manual	3
Hardware compatibility	3
Input	4
Keyboard commands for sequencer, order list and sound editor.....	4
Keyboard commands for sequencer and order list.....	5
Keyboard commands in sequencer	6
Order list commands	7
Sound/instrument editor keyboard commands	8
Load menu commands	9
Saving & dumping tunes.....	12
The sound editor	12
Waveforms and wavetable programs	15
Waveform commands	16
Vibrato program	20
Pulse program	21
Filter program.....	22
Arpeggio program.....	23
Tempo program.....	24
Initial volume	24
The sequencer	25
Sequencer FX + note combinations (for channels 1-3)	28
Sequencer FX + note combinations (for channel 4)	29
Sequencer / order list screen.....	30
Tricks to make a tune use less memory	31
Finalizing a tune.....	32
The player flags	33
Using the music in a demo/game.....	35
Known bug issues	36
Dictionary	36
Memory overview	37
PC/emulator key mapping	39
Note tables.....	40

BACKGROUND

SDI is a music tracker system for the Commodore 64, and is built on ideas from JCH/Vibrants editor, Olav Morkrid/Panoramic "Digitalizer" editor and Geir Tjelta/Shape/Moz(ic)art SID Systems. And some others.

The name **SID Duzz' It** was invented while watching a terrible TV commercial from America. It was a screwdriver that really did it!

THE MANUAL

This manual is written by Henrik Mortensen. It consists of text from the official docs included with SDI 2.1.6, and notes and corrections based on both newer versions, my own experiences and typos from the docs. The learning curve can be steep, so I made this manual for my own ease of use, and hope that others can benefit from it as well. I am not affiliated with Glenn Rune Gallefoss or Geir Tjelta, just a regular user of this great tracker, wanting to make the documentation a little easier on the eyes :)

This document is subject to change if new versions are released in the future.

HARDWARE COMPATIBILITY

Commodore 64 and C128 in C64 mode. And it should work on all disk drives/devices that can be connected to these computers using device number #6 - #30. Only the drives and cartridges listed below have been tested:

Disk drives (physical and emulated)

- 1541,1541-II and clones (Oceanic ++)
- 1570,1571 and 1581
- CMD FD 2000,CMD FD 4000 and CMD HD.
- CMD Ramlink and IDE64.
- 1541 Ultimate
- SD2iec
- Netdrive (TFR)

Cartridges

- The Final Cartridge III: Works with fastload and fastsave on 1541(II) and 1571. You will have to use the KILL command before loading/saving from a 1581 or a CMD drive.
- Action Replay 6: Works with fastload and fastsave on 1541(II), 1571 and 1581.
- Epyx Fastload: Works with fastload and fastsave on 1541(II) and 1571.
- Retro Replay: Works with fastload and fastsave on 1541(II), 1571, 1581. I know CMD FD 4000work with fastload, but im not sure how reliable fast save is.
- The Final Replay with Netdrive.

NOTE: This editor has NO UNDO function! Save your work often to avoid pain!

INPUT

All parameters you change and values you enter in sound editor, sequencer and order list are hexadecimal. Heximal input keys are **0123456789ABCDEF**.

KEYBOARD COMMANDS FOR SEQUENCER, ORDER LIST AND SOUND EDITOR:

+	Fast forward play
G 1-4	Track on/off
F1	Play from mark (mark made with F2)
F2	Set play mark
F3	Stop/Continue play
F6	Show/hide order list
F7/F8	Select octave (individual octaves for each track)
=	Play current line (press again to stop)
Z	Play song from current line
RUN/STOP	Sound Editor (press again to exit)
/	Toggle Sequencer & Sound editor edit on same screen
SHIFT L	Load music

⌘ S	Dump music
⌘ +/-	Select subtune
?	Set speed calls
⌘ Q	Quantize (0/2) ¹ for recording notes
CRSR	Left/Right/Up/Down

¹ When Quantize is 0 you will record notes on every line. When Quantize is 2 you will record notes on even lines.

KEYBOARD COMMANDS FOR SEQUENCER AND ORDER LIST

F5	Toggle between sequence and order list
⌘ *	Set speed channels ¹ (place cursor on track)
H	Hunt for next UNUSED sequence ² and replace the current one with it
⌘ H	Hunt for next UNUSED sequence, and insert it after the current one
S/L	TAB jump left/right
V	Toggle View player counters / Full screen sequencer
SHIFT CLR/HOME	To start of song
SHIFT S	Save Music
SHIFT A	Increase cursor jump
⌘ A	Decrease cursor jump
SHIFT C	Copy sequence (whole sequence)
SHIFT V	Paste into sequence (whole sequence)
SHIFT D	Double sequence length (also clones data within it)
⌘ C	Clone sequence ³ to the next UNUSED one and insert it after the current sequence
⌘ F	Fill sequence with empty lines (from cursor position)
⌘ I	Insert line in the order list with same value as the current one
⌘ K	Kill sequence (set to UNUSED) / Make a sequence ready for use
M	Set start mark (grey background is marked area)

SHIFT M	Set end mark
⌘ M	Copy marked area ⁴ to cursor position
: / [Set track transpose
; /]	Set sequence number

¹ **Active speed channels** are highlighted with light grey on the top of the screen.

² **Hunt unused sequence** will search through sequencer memory for the next available sequence. It will not scan track data to check if that sequence is already present as an **UNUSED** sequence.

³ **Clone sequence** makes a copy of the current sequence and paste it into a new sequence number on the next track line.

⁴ The **copy marked area** function in the sequencer works as a "EOR" copy. Copy some sequence data with some empty lines into a sequence, now move the cursor down one line and copy again - and you'll see...

When you use the mark function (**M** and **SHIFT+M**), you will notice that the marked area turns grey. This area can be copied anywhere within another sequence or inside the same sequence.

KEYBOARD COMMANDS IN SEQUENCER

F4	Edit/Synth/Record mode (Grey/Blue/Red)
INST/DEL	Delete a line
SHIFT INST/DEL	Insert a line
CLR/HOME	Cycle Sequence top/middle/bottom
SPACE	Delete down (depends on cursor jump)
SHIFT SPACE	Delete up (depends on cursor jump)
SHIFT X	Narrow sequence from cursor position (remove every 2nd line)
⌘ X	Expand sequence from cursor position (place cursor on a odd line)
⌘ RTYU	Set sequencer mark
⌘ 5678	Jump to sequencer mark
, / .	Jump down/up 16 lines
> / <	Increase/decrease notes from cursor position
⌘ RETURN	Play music from top of current sequence
SHIFT RETURN	Play and loop music from top of current sequence
RETURN	Play song from current line (same as Z)

N	Turn note to tie/normal
SHIFT N	All marked notes turned to normal
◀ N	All marked notes turned to tie
G	Gate on (GAT in sequence)
SHIFT G	Gate off (GAT in sequence)

ORDER LIST COMMANDS

INST/DEL	Delete an order list line
SHIFT INST/DEL	Insert an order list line
>/<	Increase/decrease transpose values or order list values
RETURN	Set loop mark for current channel
SHIFT RETURN	Set stop mark for current channel
SHIFT 1-3	Swap tracks (including all subtunes). Music must be off.

Marking and copying tracks/order list

Don't copy within the marked area and don't copy to a position above the marked area within the same track. You can copy track data to any of the other channels and to subtunes. The marked area in the order list is of course only displayed if the order list is turned on (**FG**). You can only copy tracks when the music is turned off.

Swapping tracks

When swapping tracks, you should be located at the very beginning of the first tune. All track data is swapped, including sub tunes. You can only swap tracks when the music is turned off.

When entering notes in the sequencer, you have 2 octaves present at all times:

	1	2		4	5	6		8	9		+	-	£
CT RL	Q	W	E	R	T	Y	U	I	O	P	e	*	↑

Hold down **SHIFT** to get tie/attack notes. When you press **F1** to play the song, your **F4** mode (grey/blue/red) is saved, and when you press **F3** to stop/pause the song, your saved **F4** mode is restored.

To use the sequence divide or multiply function (**SHIFT+X** and **C+X**) you have to place the cursor on a odd line number (01,03,05..etc). Fill up a \$1F length sequence with a few notes and you will understand what this is used for.

Channel marking (**C+RTYU**) is for quick jump to different positions in the song. Note that the marked positions can easily be corrupted by inserting track lines in front of the marked position.

When you press **SHIFT+RETURN** to enable the "play and loop" function, you will see that an up-arrow (**↑**) is inserted into the clock timer on screen. Using this function while editing short length sequences in grey mode is pretty neat.

Setting gate off with **SHIFT+G** in the sequence will make the instrument start the release cycle using its original release value. Setting gate on with **G** in the sequence will make the instrument start its original attack, decay and sustain cycles.

SOUND/INSTRUMENT EDITOR KEYBOARD COMMANDS

N	Name sound/instrument (RETURN to exit)
+/-	Select sound
SHIFT +/-	Select sound with focus on displaying its program
>/<	Select arpeggio number with focus on the arpeggio data
SHIFT A	Arpeggio program table
SHIFT F	Filter program table
SHIFT I	Initial volume and Filter channels / Filter speed table
SHIFT P	Pulse program table

SHIFT S	Sound setup table
SHIFT T	Tempo program table
SHIFT V	Vibrato program table
SHIFT W	Waveform program table
M	Mark current instrument for copy
SHIFT M	Paste instrument copied with M
RETURN	Put current program line into sound setup. Only for waveform, pulse, filter and vibrato. (RETURN in tempo table will set default tempo for this song).
SHIFT RETURN	Delete program from sound setup. Only for waveform, pulse, filter and vibrato.
,/.	Jump 4 lines up/down
CLR/HOME	Go to current sound's program line (if there is one). (For arpeggio it will display current arpeggio at top).
SHIFT CLR/HOME	Go to top of program line table
INST/DEL	Delete a program line (not for sound setup table)
SHIFT INST/DEL	Insert a program line (not for sound setup table)

You cannot press **SHIFT+S** to save the tune when you are in sound editor mode. Instruments \$20-\$2F are only available through the arpeggio program. You should only use these instruments if you run out of \$00-\$1F instruments. And as explained, they can only be used as arpeggios. The currently selected instrument is used in the blue synth mode.

LOAD MENU COMMANDS

SPACE	Read a new directory into memory
SHIFT SPACE	Go to DOS command screen
*	Display files A-Z
SHIFT *	Display files Z-A
A-Z	Set display path
CTRL 1-0	Select disk drive 11,12,13,14,15,16,17,8,9 and 10
CRSR	Select music

RETURN	Load music
SHIFT RETURN	When loading music with SHIFT held down, it will load order list and sequencer data only, and leave instruments intact
,/. 	Jump 8 files up/down
CLRHOME	Top of directory
SHIFT CLR/HOME	Bottom of directory
RUN/STOP	Exit

The load menu will immediately access the device you started the editor from, but only the first time you enter it. The load routine has disk error checks in case there are any problems loading a file.

If you try to access a drive that is turned off you will get a message saying **DISK STATUS ERROR**. The same message will pop up if you try to read a dir and you have removed the disk.

Clearing memory or loading track + sequencer data can be done by pressing **SHIFT+RETURN**: If you load the file "**clear memory...**" this way, then sequencer and order list data will be cleared - and sound memory will stay intact. If you load any other file this way, then sound memory will stay intact and memory for order list + sequencer will be replaced with the file you selected.

The load menu will only display SDI files. All SDI 2.X files are tagged with 'up arrow' (↑) at the beginning of the filename.

The directory displayer can only handle 128 SDI files. A 1541 disk can only handle 144 files, while other drives (1581,CMD,IDE64) can handle much more.

DOS commands

The DOS command screen can be used to handle files and directories. Anything you do here is at your own risk, so be careful...

CRSR	Move cursor around.
CLR/HOME	Go to start of command line
SHIFT CLR/HOME	Clear command line
RETURN	Send disk command
RUN/STOP	Exit to load menu

Wildcards are **?** and *****, and the following commands works on all drives:

\$	Display directory
\$a*	Display files starting with the letter 'a'
\$??a*	Display files where the 3rd letter is 'a'
S0 : FILENAME	Scratch file
R0 : NEWNAME=OLDNAME	Rename file
N0 : DISKNAME , ID	Format disk (ID must be 2 letters for 1541 format)
V	Validate disk
I	Initialize drive
UI	Reset drive (returns DOS version number)

The following commands works on CMD drives:

MD : DIRNAME	Make directory
RD : DIRNAME	Remove directory
CD : DIRNAME	Go to directory
CD : /	Go down one directory level

The following commands works on the IDE64:

MD : DIRNAME	Make directory
RD : DIRNAME	Remove directory
CD : DIRNAME	Change directory
/DIRNAME	Change directory
/	Go down one dir level + autodisplay dir
CD : . .	Go down one dir level
/ . .	Go down one dir level (root dir?) + autodisplay dir
T-RA	Display date and time

SAVING & DUMPING TUNES

Saving

The filename can only be 15 characters long - the first character is reserved for the SDI filename tag (†) and it is not displayed on the screen. Save routine has disk error check in case there are problems saving a file. A bytepacker that will pack memory \$3000-\$D000 and \$E000-\$EE00 is used inside the editor - that's why the files are small. Everything you change will be saved, including marking tags and marked sound. File size of an empty file is approximately 5 blocks. File size for a tune using all sequences and tracks is approximately 60 blocks.

Dumping

The filename can only be 15 characters long. The first character is reserved for the SDI dumpname tag " " (a single space), and it is not displayed on the screen. Dump routine has disk error check. The dumper converts your tune(s) into a Turbo Assembler sequential file.

Note: There is a 256 byte limit on each sequence, so sequences with far too much data may cause the dumper to stall!

The dumper will only dump sequences that are in use inside the sequencer/order list, and it will not dump sequences inside a channel that is OFF. All you have to do is make sure the music sounds alright before dumping, and remember that only channels that are turned ON will be dumped, including any subtunes you have. File size of a dumped file will range from 5 to 120 blocks. Before you dump a file, you should go through your instruments and check that they are ok. In most cases where a dumped tune sounds weird, it is caused by the user having program pointers to zero filled lines - especially waveform / vibrato / filter / pulse programs.

THE SOUND EDITOR

This is a basic sound setup example:

```
05 WAVEFORM PRG
08 ATTACK/DECAY
7D SUSTA/RELEASE
20 GATE TIMEOUT
08 VIBRATO PRG
02 PULSE PRG
01 FILTER PRG
1F BAND/RESONANS
00 DETUNE HI
00 DETUNE LO
```

05 WAVEFORM PRG	Waveform program. This points to a line in the waveform table, which you can access this by pressing SHIFT+W .
08 ATTACK/DECAY	Attack and decay values for the instrument
7D SUSTA/RELEASE	Sustain and release values for the instrument
20 GATE TIMEOUT	Gate timeout/hard restart. Let you specify for how long the player shall wait before setting release. <ul style="list-style-type: none"> • 00,20,40,60,80,A0,C0,E0 : No timeout • 01-1F : Normal hard restart and gate timeout • 21-3F : Hard restart 2 and gate timeout • 41-5F : Hard restart 3 and gate timeout • 61-7F : Hard restart 4 and gate timeout • 81-9F : Soft restart 1 and gate timeout • A1-BF : Soft restart 2 and gate timeout • C1-DF : Soft restart 3 and gate timeout • E1-FF : Soft restart 4 and gate timeout
08 VIBRATO PRG	Vibrato program. This points to a line in the vibrato table, which you can access by pressing SHIFT+V . <ul style="list-style-type: none"> • 00 : No vibrato • 01-55 : Vibrato program
02 PULSE PRG	Pulse program. This points to a line in the pulse table, which you can access by pressing SHIFT+P . <ul style="list-style-type: none"> • 00 : No pulse • 01-40 : Pulse program • 41-80 : Pulse program with infinite sweep • 8x : where x is a number between 1-F. This value is stored directly in the SID register for Pulse High.
01 FILTER PRG	Filter program. This points to a line in the filter table, which you can access by pressing SHIFT+F . <ul style="list-style-type: none"> • 00 : No filter • 01-40 : Filter program • 41-80 : Filter sweep mode 1 • 81-C0 : Filter sweep infinite mode 2 • C1-FF : Filter sweep mode 3
1F BAND/RESONANS	Filter band and resonance for the instrument. <ul style="list-style-type: none"> • 00 : No filter

00 DETUNE HI	Detune high frequency. <ul style="list-style-type: none"> • 00 : High frequency finetuning off • 01-7F : Finetune high frequency upwards • 80-FF : Finetune high frequency downwards
00 DETUNE LO	Detune low frequency. <ul style="list-style-type: none"> • 00 : Low frequency finetuning off • 01-7F : Finetune low frequency upwards • 80-FF : Finetune low frequency downwards

Drums examples

00 SNARE DRUM	WAVES	01 BASS DRUM	WAVES
00 WAVEFORM PRG	00:09 00	07 WAVEFORM PRG	07:09 00
08 ATTACK/DECAY	01:81 CE	08 ATTACK/DECAY	08:81 CE
88 SUSTA/RELEASE	02:41 AC	86 SUSTA/RELEASE	09:41 A6
22 GATE TIMEOUT	03:41 AA	22 GATE TIMEOUT	0A:41 A2
00 VIBRATO PRG	04:41 A7	00 VIBRATO PRG	0B:41 9E
88 PULSE PRG	05:81 DE	88 PULSE PRG	0C:FF 0B
00 FILTER PRG	06:FF 05	00 FILTER PRG	
00 BAND/RESONANS		00 BAND/RESONANS	
00 DETUNE HI		00 DETUNE HI	
00 DETUNE LO		00 DETUNE LO	

02 BASS AND BASSD1	WAVES	PULSE ;Using pulse sweep
00 WAVEFORM PRG	00:09 00	01:F7 88 08 82
08 ATTACK/DECAY	0E:81 CE	02:81 2e 10 42
86 SUSTA/RELEASE	0F:41 A6	
20 GATE TIMEOUT	10:41 A2	
00 VIBRATO PRG	11:41 00	
01 PULSE PRG	12:FF 11	
00 FILTER PRG		
00 BAND/RESONANS		
00 DETUNE HI		
00 DETUNE LO		
03 BASS AND BASSD2	WAVES	PULSE ;Using pulse hold
00 WAVEFORM PRG	00:09 00	01:08 00 03 82
08 ATTACK/DECAY	0E:81 CE	02:81 2e 10 42
86 SUSTA/RELEASE	0F:41 A6	
20 GATE TIMEOUT	10:41 A2	
00 VIBRATO PRG	11:41 00	
01 PULSE PRG	12:FF 11	
00 FILTER PRG		
00 BAND/RESONANS		
00 DETUNE HI		
00 DETUNE LO		

These examples are all using fixed notes in the waveform table, but you can of course use soft notes to create drums as well.

WAVEFORMS AND WAVETABLE PROGRAMS

These are the four standard waveforms available in the SID chip:

	Triangle waveform (10)		Pulse waveform (40) (pulse value must be set)
	Sawtooth waveform (20)		Noise waveform (80)

Waveform program

C1 C2 C3 00:41 00 01:41 03 02:41 CA 03:FF 00	C1 displays program line position. C2 displays waveforms and waveform commands. C3 displays soft/fixed note values and 2nd part of the waveform commands. Possible note values for C3 : 00-5E Soft notes, added to note+track transpose 60-7F Soft notes, subtracted from note+track transpose 80-DE Fixed notes, overrides note+track tranpose
---	---

These waveforms sounds the same on both 6581 and 8580 SID chip. They can be turned on and off by adding the gate bit:

- **00** Gate off
- **01** Gate on

The waveforms and gate bits are added up, so in other words - a sawtooth waveform with gate on is **21**, and a sawtooth waveform with gate off is **20**.

You also have the possibilty to add sync and ring modulation:

- **02** Sync bit gate off
- **03** Sync bit gate on
- **04** Ring modulation gate off
- **05** Ring modulation gate on

Adding ring modulation gate on to Triangle waveform therefore becomes **15** (**\$10 + \$05**).

Combined waveforms (with gate off)

- **30** Triangle+Sawtooth waveform. Works best on new SIDs.
- **50** Pulse+Triangle waveform. Works with both old and new SIDs.
- **60** Pulse+Sawtooth waveform. May work differently on old SID revisions.
- **70** Pulse+Sawtooth+Triangle waveform. Very silent on old SIDs.

These may sound differently depending on the SID chip you have.

Arpeggio Waveforms

When creating arpeggios with the arpeggio program you must to add **\$80** to the waveform you want to use, like shown below:

- **91** Triangle waveform (same as **11**. Add **80**, and it becomes **91**.)
- **A1** Sawtooth waveform (same as **21**. Add **80**, and it becomes **A1**.)
- **B1** Triangle+Sawtooth waveform (same as **31**. Add **80**, and it becomes **B1**.)
- **C1** Pulse waveform (same as **41**. Add **80**, and it becomes **C1**.)
- **D1** Pulse+Triangle (same as **51**. Add **80**, and it becomes **D1**.)
- **E1** Pulse+Sawtooth (same as **61**. Add **80**, and it becomes **E1**.)

WAVEFORM COMMANDS

In the waveform column (**C2**) you enter the command, and in the note column (**C3**) you enter the parameter.

FF - Jump command

:FF xx	Jumps to program line position xx (xx can be any value between 00-FF)
---------------	---

FE - Delay command

:FE xx	Delay the following waveform for xx frames. (xx can be any value between 00-FF)
---------------	---

FD - ADSR command

:FD xx	xx tells the music routine how many frames to wait before setting gate off (xx can be any number between 00-FF). xx = 00 or 80 = no frame delay = no gate off will happen. xx = 81-FF is the same as 01-7F above, but when gate is turned off, it can't be turned back on again.
:AD SR	AD = Attack/Decay value / SR = Sustain/Release value. Note that all parameters in this command are set on the same frame.

Example:

```
00:09 00
01:41 00
02:FE 0F
03:41 00
04:FD 10 ;wait 10 frames before setting gate off
05:90 89 ;set attack 9, decay 0, sustain 8, release 9
06:40 00
07:FF 01
```

FB - Multipulse command

Switches between two pulse programs. The instrument must have a pointer to a pulse program from the sound setup.

:FB P2	P2 = second pulse program pointer
:0x yy	x = 0 : start with P2 pointer. x = 1 : start with sound setup pulse pointer. yy = switch speed

Example:

```
00:09 00
01:FB 02 ;select pulse program 02
02:01 07 ;start with this pulse pointer, then goto P2
03:41 00
04:FF 03
```

FA - Repeat command

Tells the player to repeat the following **FF** jump **XX** times. When the last **FF** jump is executed the player automatically jumps to the line below the **FF** command. Must be used together with the jump command.

:FA xx	xx = 01-FF
---------------	-------------------

Example:

```
00:FA 08 ;repeat 8 times
01:09 00
02:81 CE
03:41 A7
04:41 A5
05:41 A2
06:FF 01
07:21 00 ;ends here after the repeats
08:FF 07
```

The **FA** command can be used together with all the other waveform commands.

F0-F7 - \$D415 Filter command

This is a 1 byte command. The value you enter (**F0-F7**) is the lower 3 bits of the filter cutoff. It is stored directly into the lowpass filter register.

:Fx yy	Set lowpass filter cutoff value x (yy are not used)
---------------	---

EE - Pulse init

:EE lh	Write low high pulse value to SID registers (\$D402/\$D403) and player registers (pulselo/pulsehi)
---------------	---

ED - Pulse subtract:

:ED xx	Subtract pulse with value xx (You must have set the initial value first)
---------------	--

EC - Pulse addition

:EC xx	Add pulse with value xx (You must have set the initial value first)
---------------	---

EB - Pulse write

:EB lh	Write low high pulse value to SID pulse registers only
---------------	---

Example using the above pulse commands:

```
00:41 00
01:EE 0F ;init pulse
02:41 00
03:ED 22 ;subtract 22 on pulse
04:44 00
05:FF 03
```

When programming the pulse this way you don't need the pulse program and can disable the pulseroutine (**rem_pu = 1**) in the player flags. This may save you some rastertime.

E2-E7 - Noise trick

This is used to get a none-random (metallic) noise waveform, but has to be programmed properly in the waveform program.

:Ex yy	Write Ex to waveform register (x must be a value between 2 and 7) (yy is unused)
---------------	---

Example:

```
00:09 00 ;hard restart
01:E5 00 ;write value 5 to waveform register
02:09 00 ;another hard restart
03:81 C8 ;noise with gate on (fixed note C-6)
04:FF 03 ;loop to 03
```

VIBRATO PROGRAM

C1 C2 C3 C4 01:10 00 00 02:FF 03 32	C1 displays the table position C2 displays the delay value, detune command(s) and infinite loop command (FF). Delay values range from 01-FD : 00 = detuning and continue 01-FD = delay value FE = detuning and hold FF = infinite loop on vibrato C3 displays vibrato width: 00-7F = going up, then down 80-FF = going down, then up C4 displays vibrato speed OR detune high byte
--	---

Crazy comet ex: **00:08 00 00 ;wait 8 frames**
 01:FF 10 BE ;crazy comet loop

Using values greater than **80** in column 4 will produce what we call the Crazy Comet loop.

Detuning example: **01:FE DL DH**

Detuning followed by vibrato example:

01:00 d1 dh
 02:FF 03 32

d1 = Low value of frequency detuning (Depends on **dh**)

dh = High value of frequency detuning (**00-7F** = finetune upwards / **FF-80** = finetune downwards)

Most likely you will only need to change the **d1** value and leave **dh** to zero. You can call a vibrato program from the FX column in the sequencer with **\$21-\$3F**.

PULSE PROGRAM:

Pulse sweep

<p>C1 C2 C3 C4 C5 01:F7 88 08 82 02:01 2E 30 42</p>	<p>C1 displays the table position C2 displays PulseLow / PulseHigh starting value C3 displays PulseHigh sweeping value C4 displays sweep speed C5 displays sweep mode OR sweep jump. C5 column commands are:</p> <p style="padding-left: 40px;">00,40,80,C0 Sweep until reaching end value, then the sweep will stop. No jumping to other pulse program line will occur. The result of the pulse sweep all depends on the values used in C1 and C2.</p> <p style="padding-left: 40px;">0x-3F Sweep till end value then cut to the C2 value. x indicates which program line to cut to when reaching the end.</p> <p style="padding-left: 40px;">4x-7F If x points to the same program line then the sweep will be continuous between the two values in C3. If x points to a different program line the sweep will first go all the way between the two values in C3, then it will sweep to the new program line value in C3.</p> <p style="padding-left: 40px;">8x-BF Sweep till end value then cut to the C2 value. Behaves the same way as 0x-3F but the sweep is reversed.</p> <p style="padding-left: 40px;">Cx-FF Continuous sweep between the 2 values in C3. Behaves the same way as 4x-7F, but the sweep is reversed.</p>
--	--

Pulse hold

Pulse hold is another method of creating interesting pulse sweeps:

<p>C1 C2 C3 C4 C5 01:08 00 18 82 02:01 00 18 81</p>	<p>C1 displays the table position C2 is the initial pulse value. If C3 is zero, then C4 will be used as a delay value. The player will decrease this value down to zero, then perform the jump value in C5.</p>
--	---

FILTER PROGRAM

This program works just like the pulse program, with two exceptions:

1. The low/high byte starting values in column **C2** have switched places to high/low.
2. Filter program doesn't have the pulse hold routine. Instead of the pulse hold routine, the filter routine has special filter frame routine. This routine can set filter cutoff high, band, resonance and a frame delay counter of 1 or 2.

Filter frame Example: **C1 C2 C3 C4 C5**
01:4F 00 2F 82 ;filter frame with 1 as delay
02:22 00 64 03 ;filter frame with 2 as delay
03:11 15 02 43 ;normal filter sweep (continuous)

When column **C3** is zero, the values of **C2**, **C4** and **C5** is used in the filter frame routine. Let's take a closer look at the example lines:

Filter frame with 1 as delay

4F	4F as filter cutoff high (\$D416)
00	ZERO means this line is treated as a filter frame
2F	Band is set to \$20 and resonance is set to \$0F
82	8x is a 1 frame delay before jumping to line 02

Filter frame with 2 as delay

22	22 as filter cutoff high (\$D416)
00	ZERO means this line is treated as a filter frame
64	Band is set to \$60 and resonance is set to \$04
03	0x is a 2 frame delay before jumping to line 03

Normal filter sweep (continuous)

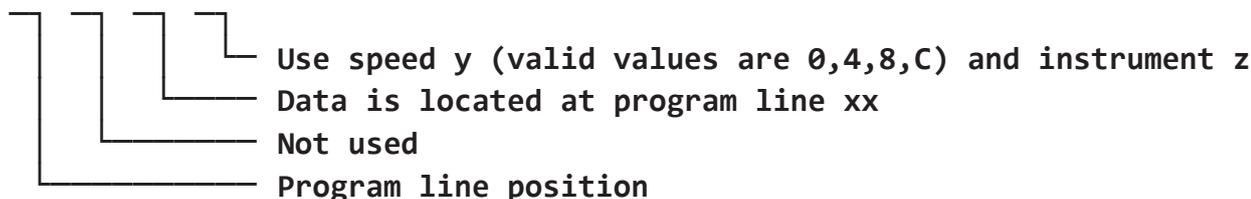
11	11 as filter cutoff high (\$D416)
15	Sweep between \$10 and \$50
02	Sweep speed 02
43	Loop back to program line 3 (+\$40 = continuous sweep)

ARPEGGIO PROGRAM

Editor's note: The official docs doesn't explain arpeggio programming very good (no offense, Glenn & Geir), and it can be rather confusing. Instead, I will show how to use jump tables, which will give you a better overview of your chords.

The principle of jumptables is that you make shortcuts at the beginning of the arpeggio table (which you will reference with **40-6F** in the sequencer), and have them jump to the actual programs a little further down the table:

C1 C2 C3 C4
00:00 xx yz



Example of a jumptable with minor/major chords

C1 C2 C3 C4 00:00 05 43	** the shortcuts at the beginning of the arp table ** C1 Program line position. This (00) will be arpeggio program 40 . C2 Not used C3 This arpeggio's data is located at line 05 C4 Use speed 4 and instrument 03
01:00 0A 28	C1 Program line position. This (01) will be arpeggio program 41 . C2 Not used C3 This arpeggio's data is located at line 0A C4 Use speed 2 and instrument 08
...	
...	
...	
05:00 00 00	** the chords referenced ** C1 Major chord: no transpose
06:04 00 00	C1 Major chord: transpose 4 semitones up from root
07:87 00 00	C1 Major chord: transpose 7 semitones up from root and loop (+\$80)
...	
0A:00 00 00	C1 Minor chord: no transpose
0B:03 00 00	C1 Minor chord: transpose 3 semitones up from root
0C:87 00 00	C1 Minor chord: transpose 7 semitones up from root and loop (+\$80)

Remember to add **\$80** to your waveforms in the wave program to be able to use them for arpeggio...

TEMPO PROGRAM

C1 C2 C3 01:02 00 02:83 00	C1 displays the program line position & tempo program number C2 displays the the tempo values: 01-7F = tempo value 81-FF = value greater than 80 indicates a loop C3 displays the program line lookup pointer
---	---

To set default tempo program press **RETURN** on the lookup pointer you want to use. Notice that the **TP** value in the down-right corner of the screen will change. You can also call these tempo programs with **\$40-\$6F** from track 4 in the sequencer.

INITIAL VOLUME

If you want to change the volume or make the volume fade in you can access the INVOL menu from the sound editor by pressing **SHIFT+I**.

C1 C2 C3 00:14 00 01:0F 35 02:F4 00	C1 displays song number C2 displays volume/fadein for the song C3 displays filter channel and filter speed
--	---

Example: **00:14 00 ;song 00 fadein 1, initial volume 4**
 01:0F 35 ;song 01 max volume, no fadein
 ;force filter channels 1+2, filter delay 5
 02:F4 00 ;song 02 fadein F, initial volume 4

The high nibble (1st byte) of **C2** is used to set fadein. Valid numbers are **1-F** where **1** is the fastest fadein (**0** = no fadein).

The low nibble (2nd byte) of **C2** is the starting volume of the song. Valid numbers are **0-F** where **0** = no volume and **F** = max volume.

Using maximum volume together with a fadein value will not produce any fadein. The fadein routine will always fadeup to maximum volume. Default volume for all tunes when starting editor from scratch is **\$0F**.

Filter settings

You do not need to type anything in **C3** to make use of filter in the sound editor. The high nibble (1st byte) of **C3** is used to force filter on a channel when a tune is played from the beginning. To make use of this function you need a filter instrument in one of the other channels.

- **0** Filter force off

Force a single channel:

- **1** Force channel 1
- **2** Force channel 2
- **4** Force channel 3

Force multiple channels:

- **3** Force channels 1+2
- **5** Force channels 1+3
- **6** Force channels 2+3
- **7** Force channels 1+2+3

The low nibble (2nd byte) of **C3** controls filter speed delay. The default value here is **0**, which is the fastest speed delay. This value is printed to screen just below "**FILTE**" and it is used to delay the filter speed inside filter program. Each song can have its own filter speed delay.

Valid values are **0** to **F**.

At the moment its not possible to change this value from inside the tune.

Press **CLR/HOME** to go to current song's volume number.

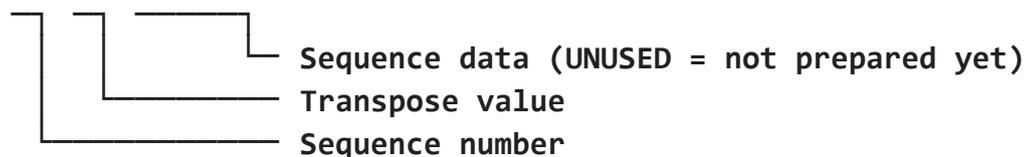
THE SEQUENCER

The sequencer looks pretty much like this when you start up SDI:

```
OFF 00 00 OFF 01 00 OFF 02 00 OFF 7F 00
A000UNUSED A001UNUSED A002UNUSED A003UNUSED
J000          STOP          J000          J000
```

There are 4 channels available, so let's break that up into a bit more detail for a single channel:

A0 00 UNUSED



Jxxx / STOP



Valid sequence numbers are **00-7F**, and valid transpose values are:

- **80-9F** Transpose down
- **A0** No transpose
- **A1-BF** Transpose up

The jump pointer (**J000**) at the bottom is a 2 byte value. You can set the jump pointer in the tracker by pressing **RETURN**. This will make the tune jump back to a specific location when it reaches this pointer.

The stop tag (**STOP**) triggers the music player to stop the music channel. You can set the stop tag by pressing **SHIFT+RETURN** in the tracker.

The sequencer consists of 4 channels. Channels 1-3 are audio channels while channel 4 is used for main transpose/tempo changes and filter effects.

All sequences are marked as "**UNUSED**" when you start the editor. To make a sequence usable you have to press **C+K** and "**UNUSED**" changes to "-- ---".

An easy way to prepare a sequence, is pressing **SHIFT+INST/DEL** a couple of times, and then tap **SHIFT+D** to double it until you have the desired length. You can then adjust it to your liking by using (**SHIFT+**) **INST/DEL**. Each time you press **SHIFT+D**, the sequence will double in length, like **03, 07, 0F, 1F** (decimal **04, 08, 16, 32**) and so on and so forth.

There are 3 different edit modes in the sequencer, and you can cycle between them by pressing F4. The edit bar will change color between Grey, Blue and Red.

1. Grey mode (normal mode):

You can enter notes while the music is playing or when the music is turned off.

2. Blue mode (synth mode):

You can enter notes with sound output, but you cannot play the music at the same time. To select a new sound, you can type in the sound number or arpeggio number in the FX column, or you can select an instrument from the sound editor (**+/-**).

3. Red mode (record mode):

Red mode is available ONLY when music is playing. Try pressing **F4**, and you'll see that the edit bar turns red and the sequencer is scrolling all the sequences. If the channel you are editing is turned ON you can record your notes at this point. If the channel you are editing is turned OFF you can play along with the music. You can't edit the FX columns while in record mode.

Special 1:

Pressing **F3** to pause the music returns you to grey or blue mode, and the edit bar will be at the position where you stopped your music.

Special 2:

Pressing **F4** to switch mode returns you to the position you had before you pressed **F1** to play the music.

Special 3:

When you have the split screen up (sequencer/sound edit) you will not be able to record your notes. All keyboard inputs to the sequencer is ignored (This is only for record mode - Blue mode and Grey mode will behave as normal). This way you can safely edit your sounds while being in red mode.

SEQUENCER FX + NOTE COMBINATIONS (for channels 1-3)

Each sequence has two columns - the first is the FX column where you control what kind effects you want to use, and the second is the note column.

-- ---	Empty line
06 ---	[00-1F] Set sound number 06 and no note
10 C-4	[00-1F] Set sound number 10 and note C-4
06 [C-4]	[00-1F] Add sound number to waveform and tie note C-4
2E C-4	[21-3F] Set glide value 2E and note C-4
2E [C-4]	[21-3F] Set glide value 2E and tie note C-4
22 ---	[21-3F] Set vibrato program 02
40 ---	[40-6F] Set arpeggio 00 and no note
44 C-4	[40-6F] Set arpeggio 04 and note C-4
6F [C-4]	[40-6F] Set arpeggio 2F and tie note C-4
74 ---	[70-7F] Set release 4 (cannot have a note combination here)
74 C-4	[70-7F] Set sustain 4 and note C-4
7A [C-4]	[70-7F] Set attack A and tie note C-4
-- C-4	[C#0-A#7] Notes (played with current sustain value - if set)
-- [C-4]	[C#0-A#7] Tie notes
-- GAT	[C-0] Set gate on for current sound
-- [GAT]	[C-0] Set gate off using current sound release value
70 C-4	[70] Restore current sound original ADSR values with note C-4

Tie notes are displayed with a brown background color inside the editor. The note for the attack function is displayed as a tie note, but it is a note that restarts all programs the instrument is using.

SEQUENCER FX + NOTE COMBINATIONS (for channel 4)

-- ---	Empty line
06 ---	[00-1F] Set tempo to 06 and no transpose
04 C#0	[01-1F] [C-0 to A#7] Set tempo to 04 and transpose 1
-- D-0	Set transpose 2
41 ---	[40-60] Look up tempo program 01
44 GAT	[40-60] Look up tempo program 04 and transpose 0
70 ---	[70] Filter control back to main filter channel
71 ---	[71-7F] Force filter output
21 ---	[21-3F] Force filter program 01
63 ---	[61-67] Force filter band 03

This channel controls main speed and main transpose for all channels. It's not necessary to use this channel, but you can create some strange and interesting tunes by changing the speed and transpose values here. The only problem here is when setting a too high transpose value, it could make the editor crash. Keep the transpose between **GAT** and **C-2** just to be safe...

Starting the first sequence line here with "-- **GAT**" is a very good choice.

00 as tempo doesn't work - it will stop the music.

**** Critical information ****

Make sure channel 1-3 has initialized a filter program before starting playing around with filter effects here. Channel 4 filter control works like a layer over Channel 1-3. But it is of vital importance that you have initialized a filter program with an instrument in channel 1-3.

If you haven't, the tune might not init properly, and/or channels will be muted. What most likely happens when you get this muted bug is that band/resonance is set to **00**.

SEQUENCER / ORDER LIST SCREEN

TOP OF SCREEN:

1A	1B	1C	2A	2B	2C	3A	3B	3C	4A	4B	4C
000	03	7F	000	06	3F	000	07	3F	OFF	03	7F
1D	1E		2D	2E		3D	3E		4D	4E	
A00302			D#1A00606			D#2A00708			D#7A00302		D#1
A009--			---A005--			---A002--			---J000--		---
A009--			---A005--			---A002--			---03	--	---
9E09--			---9E05--			---9E02--			---04	--	---
9E09--			---9E05--			---9E02--			---05	--	---
A209--			---A205--			---A202--			---06	--	---
A209--			---A205--			---A202--			---07	--	---
STOP--			---STOP--			---STOP--			---08	--	---

- 1A** Track 1 current Position [000]
- 1B** Track 1 current sequence number [03]
- 1C** Track 1 current sequence length [7F]
- 1D** Track 1 current Transpose [A0] for ...
- 1E** ... sequence number [03]
- 2B** Track 2 current Position [000]
- 2B** Track 2 current sequence number [06]
- 2C** Track 2 current sequence length [3F]
- 2D** Track 2 current Transpose [A0] for ...
- 2E** ... sequence number [06]

- 3A** Track 3 current Position [000]
- 3B** Track 3 current sequence number [07]
- 3C** Track 3 current sequence length [3F]
- 3D** Track 3 current Transpose [A0] for ...
- 3E** ... sequence number [07]
- 4A** Track 4 is turned OFF. [OFF]
- 4B** Track 4 current sequence number [03]
- 4C** Track 4 current sequence length [7F]
- 4D** Track 4 current transpose [A0] for ...
- 4E** ... sequence number [03]

BOTTOM OF SCREEN:

INSTRUMT	WAVEFORM	PULSELOW	SEQUENCE	01
00 00 00	00 00 00	00 00 00	00 00 00	00 00
ARPEGGIO	FILTE	3F	PULSHIGH	TRANSPOSE
FF FF FF	1 F1	567 0	0 0 0	00 00 00 00
01:59	00	"FILENAME	"	00/02 1 2 5

Most of the information shown here is taken from the player when it is in play mode (F1).

FILTE 3F is Band/Volume. The 3 numbers below "**FILTE**" are:
1 : Filter delay speed
F1 : Resonance/Filter voices
567 : **56** = Filter cutoff high (**\$D416**) and **7** = Filer cutoff low (**\$D415**)

SEQUENCE 01 is the cursor jump value (**SHIFT+A/C+A**).

01:59 is the Time of Day Clock - followed by ...

00 ... which tells the song number (**C +/-**)

FILENAME should be pretty self-explanatory :) Is is followed by ...

00/02 ... which is the tempo program pointer/current tempo data.

1 is the multispeed calls (**SHIFT+?**).

2 is the quantize mode (**C+Q**).

5 is the transpose value while editing (**F7/F8**)

Channels has individual transpose values here.

TRICKS TO MAKE A TUNE USE LESS MEMORY

The music player itself is around \$0900 bytes long +/- the effects you want to use. You may be able to make it use less memory by using these techniques:

1. Instead of repeating the same notes inside a sequence you can make the tracker repeat the sequence. That will minimize the file size a lot, especially for the finalized tune.
2. Only set the instrument you want to use when it is necessary. You can save 1-2 bytes per note when finalizing the tune by doing this. The same goes for arpeggio and sustain/release values. Attack HAVE to be set each time.
3. Each tie note takes 2 bytes in the finalized (dumped) tune. One byte for the duration and one for the note, while normal notes only needs a new duration when the duration value is different from the last. You will save a lot of memory by making a instrument that emulates tie notes. Make a instrument using gate time out **E0** and set the pulse program to infinite sweep. This will sound just a like a tie note.
4. Use the arpeggio routine instead of the waveform table to create arpeggios.

FINALIZING A TUNE

Now that you have made your üb3r1337 SID smash hit, it's time to slap it together into a usable file - and that's all done in Turbo Assembler. There are two versions to choose from:

- **BMTASS FAST/9000**

This assembler uses a fastload/fastsave routine that is only compatible with 1541/1571 disk drives. But entering files with **+** and **E/W** is done using normal load routines.

- **SDI TASS /9000**

This is a modified version of Soci's TASS supporting all device numbers. There are no fastload routines, but it works well on IDE64, SD2IEC and all other drives.

The source files are also available as sequence files on a separate disk, and you can easily convert them to be used with **64tass.exe**.

You will also need a machine language monitor to save the tune as a PRG file. In this example, I'm using Action Replay's monitor, but except from the syntax, the process should be similar on most others as well.

1. Dump your tune in SDI by pressing **C+S**
2. Load the assembler you want to use, and start it with **SYS4096*9**.
3. Type **+L** to load either:
 - **S.SDI21-N50** (the singlespeed player routine) *or*
 - **S.SDI21-SPD50** (the multispeed player routine)
4. Once loaded, press **F8** to go to the bottom of the player code (just below the **rts** opcode).
5. Type **+E** to load and append the dumped music data to the end of the player routine. **Remember to add a SPACE at the beginning of the filename!** When it is loaded, you may want to take a quick look to see if everything is alright. Move your cursor to the **W** label and scroll down to the **AD** label. A lot of zeros following each other may indicate that you have a program pointing nowhere. Head back to the editor and make sure all your filter-, vibrato-, pulse- and arpeggio programs are ok.
6. Adjust flags at the top of the player routine as necessary (explained in the next chapter).
7. Type **+3** to assemble, and press **S** to preview your tune.
8. If everything sounds as it should, exit the player with **C**. Type **+3** again, but do not preview this time. Instead, **take note of the end address**.
9. Press SPACE, and go to BASIC by typing **+1**, and enter the monitor by typing **MON**.

Now you can save your tune by typing:

```
s"finalmusic",8,1000,xxxx + 1
```

- where **8** is the drive number, **1000** is the start address, and **xxxx** is the end address PLUS ONE BYTE! Yes, you have to add 1 byte. If your tune ends at **\$1E15**, write **\$1E16** instead. If it ends at **\$1DAF**, write **\$1DB0**, and so on...

10. You now have a "finalmusic" PRG file, which you can use in another project or rip in programs like SIDedit to get a .SID file.

THE PLAYER FLAGS

These are the player flags at the top of the source. Almost all effects are set to be ignored (**1**) by default. Turning them back on will make the player take more rastertime and memory, so only enable the ones your tune requires.

Here's a brief explanation on each effect:

sid = \$d400	The SID chip have a few mirror banks. Try setting this value to \$D5C0 and you'll still hear music. For a C128 only banks \$D400-\$D500 will work.
mzero = \$fe	The player is using \$FE and \$FF as zeropage.
rem_4ch = 1	1 = Ignore 4th channel. If you made a tune using 4 channels you have to set this parameter to zero.
rem_det = 0	Ignore detuning (data at Z8 and Z9 can be removed).
rem_gout = 0	Ignore gate timeout.
rem_lwf = 0	1 = Ignore 1st byte of waveform. If you want to save raster time and memory you can set this parameter to 1. Warning! sound output might be different...

The **E2-FE** commands are used inside the waveform table:

rem_wfd = 1	1 = Ignore waveform delay (FE command). If you are using the waveform delay command you have to set this parameter to zero.
rem_adsr = 1	1 = Ignore ADSR (FD command). If you are using the ADSR command you have to set this parameter to zero.
rem_mp = 1	1 = Ignore multipulse (FB command). If you are using the multipulse command you have to set this parameter to zero.

<code>rem_wfr = 1</code>	1 = Ignore waveform repeat (FA command). If you are using the waveform repeat command you have to set this parameter to zero.
<code>rem_wf0 = 1</code>	ignore \$D415. \$F0-\$F7 in waveform table
<code>rem_puw = 1</code>	1 = Ignore (EB-EE command) waveform pulse
<code>rem_pu = 1</code>	1 = Ignore pulse routine
<code>rem_we2 = 1</code>	1 = Ignore E2-E7 waveform noise
<code>rem_arp = 0</code>	1 = Ignore arpeggio routine
<code>rek_fi = 0</code>	1 = Ignore filter routine
<code>rem_fspd = 0</code>	1 = Ignore filter speed
<code>rem_glid = 0</code>	1 = Ignore glide routine
<code>rem_vib = 0</code>	1 = Ignore vibrato routine
<code>rem_cc = 1</code>	1 = Ignore Crazy Comet vibrato effect
<code>rem_fad = 1</code>	1 = Ignore fadeout routine
<code>rem_gat = 1</code>	1 = Ignore GAT/FLG command
<code>rem_f20 = 1</code>	1 = Ignore sequence command 20 XX
<code>rem_wfo = 1</code>	1 = Ignore waveform ORA command in sequencer
<code>rem_voff = 1</code>	1 = Ignore voice on/off. Setting this parameter to 1 removes the code that makes it possible to toggle voices on and off while listening.
<code>rem_trk1 = 1</code>	1 = Maximum \$FF bytes per track. It is possible for each track to be \$07FF bytes long. In most cases you won't exceed the \$0100 limit. You have to test or count track bytes.
<code>rem_tp = 0</code>	1 = Ignore tempo program. If only a single tempo is used you can save cycles by ignoring the tempo program. Instead insert the tempo value you want to use in offset "s".

The following effects are for the frame player only:

<code>rem_opt = 0</code>	1 = Optional speed channels. If you have multispeed on all channels leave this parameter at 0 . If you made a speed tune that only made use of one or two speed channels you can set this parameter to 1 . In addition you have to modify the parameter below:
--------------------------	---

<code>spdchan = %00000111</code>	<p><code>%00000001</code> = channel 1 <code>%00000010</code> = channel 2 <code>%00000100</code> = channel 3 <code>%00000011</code> = channel 1 and 2 <code>%00000101</code> = channel 1 and 3 <code>%00000110</code> = channel 2 and 3 <code>%00000111</code> = all channels (set <code>rem_opt = 0</code>)</p>
<code>speed = 4</code>	<p>This is how many speeds you have made the tune in. You can enter a value between 2 and 15 here. This value has nothing to do with the player. It's for the music displayer which you can start with +3.</p>
<code>system = 1</code>	<p>1 = PAL / 0 = NTSC. This has to do with the music displayer. US machines have to select zero for NTSC mode.</p>

USING THE MUSIC IN A DEMO/GAME

You are allowed to use the music player in a game free of charge, but you will have to credit us for writing the music player. Simply state that the music was written in Sid Duzz'it, or you can credit us with our real names.

You are allowed to modify the player source.

Now let's say you assembled the tune to \$1000:

```

ldx #00 - $1f
jmp $1000      ;Init call for the tune you want to play.
jmp $1003      ;Main play call for the tune.
lda #$00-$7f
jmp $1006      ;Fadeout call for the tune
jmp $1009      ;Speedplay call for the tune.

```

Correct raster setup for a speed tune on a PAL machine (PAL = 312 scanlines):
raster = 312/speed (**speed** = how many speeds the tune was made in)

Correct raster setup for a speed tune on a NTSC machine (NTSC = 262 scanlines):
raster = 262/speed (**speed** = how many speeds the tune was made in)

You will need one Main call to the player, and the remaining calls must go to the speedplay call.

If speed = 4:

```
irq:    lda #0
        cmp $d012
        bne *-3
        jsr $1003      ;main call - updates tracks, sequences and sounds
        lda #raster
        cmp $d012
        bne *-3
        jsr $1009      ;sound update call
        lda #raster*2
        cmp $d012
        bne *-3
        jsr $1009      ;sound update call
        lda #raster*3
        cmp $d012
        bne *-3
        jsr $1009      ;sound update call
        rti
```

Remember that the **\$D012-\$D011** timing is critical to get the same sound output as in the editor.

KNOWN BUG ISSUES

1. It is possible to make the player bug by filling up a **\$7F** sequence completely with tie notes (that means a note on every line, including a instrument change or a glide). Doing this will make a dumped sequence larger than 256 bytes, which is the limit.
Only solution is to split up the sequence.

DICTIONARY

Arpeggio	A method to emulate chords in chip music by playing the chord notes in quick succession.
Channels	Tracks
Glide	Glide from note X to note Y with a specified speed. There are two different note possibilities - one with hard restart glide and one with tie glide.
Instrument	Sound

Line number	All sound programs or sequences have a line number displayed on the left to make it easy understand where you are located.
Order list	The order in which the various sequences will be played
Program line #	(In sound edit) The number displayed before ":" (ie. 01:).
Sequence	A group of simultaneously played tracks that represents a full section of the song.
Sequencer	The 4-channel editor where you enter notes, effects, etc.
Sound	Instrument
Tie note	A note that doesn't restart the different sound setup programs, but takes over the characteristics of a previously played note.
Tie glide	A tie note using the glide function.
Track line	Each track line consists of 2 parameters - the first is transpose, and the second is its sequence number.
Tracker	Sequence transpose and sequence variables for all 4 channels.
Tracks	Channels

MEMORY OVERVIEW

0100-017F	Stack
0180-0200	Sequence lengths
0200-0229	DOS command buffer
0229-	Filename
02A7-0300	Data tables
0340-0400	Sprites
0400-07E8	Screen
0800-2EE0	Editor part 1
2F00-3000	Data buffer
3000-3800	Track 1

3800-4000	Track 2
4000-4800	Track 3
4800-5000	Track 4
5000-D000	Sequences
D000-D810	Directory memory (Max 128 SDI files in a directory. The rest will be ignored.)
D810-E000	Editor part 2
E000-E100	Waveform program table
E100-E200	Waveform program note table
E200-E300	Pulse program table
E300-E400	Arpeggio data
E400-E500	Arpeggio program table
E500-E600	Vibrato program table
E600-E700	Filter program table
E700-E8E0	Sound setup
E700-E730	Waveform program pointer
E730-E760	Attack/Decay
E760-E790	Sustain/Release
E790-E7C0	Gate timeout
E7C0-E7F0	Vibrato program pointer
E7F0-E820	Pulse program pointer
E820-E850	Filter program pointer
E850-E880	Filter band/resonance
E880-E8B0	Detune high
E8B0-E8E0	Detune low
E8E0-E970	Future expansion
E970-E980	File info (speed calls/speed channels)
E980-EA00	Tempo data
EA00-ED00	Sound names
ED00-ED20	Default tempo lookup table for each tune
ED20-ED40	Channels ON lookup table for each tune
ED40-ED70	Tempo program table

ED80-EDC0	Marked channel positions
EDC0-EDE0	Invol Volume setup
EDE0-EE00	Invol Filter setup
EE00-EEC0	Note frequency table
EEC0-FFE6	Player/Editor part 3

PC/EMULATOR KEY MAPPING

<i>C64 KEYS</i>	<i>PC KEYS</i>
RESTORE	Page Up
CTRL	Tab
RUN/STOP	CapsLock / Escape
RETURN	Enter
◀	CTRL
←	§ (The key above the Tab key)
=	’ (The key above right Shift and to the left of the Enter key)
/	/_ (Can also be the Underscore key)
?	?_ (Can be the Shift + Underscore key)
CRSR	Arrow keys
*	^"}
Clr/home	Home
Note A#?	Insert
Note B-?	Delete

NOTE TABLES

FIXED NOTE TABLE:

OCTAVE 0	OCTAVE 1	OCTAVE 2	OCTAVE 3	OCTAVE 4	OCTAVE 5	OCTAVE 6	OCTAVE 7
## : RES							
80 : C-0	8C : C-1	98 : C-2	A4 : C-3	B0 : C-4	BC : C-5	C8 : C-6	D4 : C-7
81 : C#0	8D : C#1	99 : C#2	A5 : C#3	B1 : C#4	BD : C#5	C9 : C#6	D5 : C#7
82 : D-0	8E : D-1	9A : D-2	A6 : D-3	B2 : D-4	BE : D-5	CA : D-6	D6 : D-7
83 : D#0	8F : D#1	9B : D#2	A7 : D#3	B3 : D#4	BF : D#5	CB : D#6	D7 : D#7
84 : E-0	90 : E-1	9C : E-2	A8 : E-3	B4 : E-4	C0 : E-5	CC : E-6	D8 : E-7
85 : F-0	91 : F-1	9D : F-2	A9 : F-3	B5 : F-4	C1 : F-5	CD : F-6	D9 : F-7
86 : F#0	92 : F#1	9E : F#2	AA : F#3	B6 : F#4	C2 : F#5	CE : F#6	DA : F#7
87 : G-0	93 : G-1	9F : G-2	AB : G-3	B7 : G-4	C3 : G-5	CF : G-6	DB : G-7
88 : G#0	94 : G#1	A0 : G#2	AC : G#3	B8 : G#4	C4 : G#5	D0 : G#6	DC : G#7
89 : A-0	95 : A-1	A1 : A-2	AD : A-3	B9 : A-4	C5 : A-5	D1 : A-6	DD : A-7
8A : A#0	96 : A#1	A2 : A#2	AE : A#3	BA : A#4	C6 : A#5	D2 : A#6	DE : A#7
8B : B-0	97 : B-1	A3 : B-2	AF : B-3	BB : B-4	C7 : B-5	D3 : B-6	DF : B-7

NOTE TABLE UP with C-0 as base note:

OCTAVE 0	OCTAVE 1	OCTAVE 2	OCTAVE 3	OCTAVE 4	OCTAVE 5	OCTAVE 6	OCTAVE 7
## : RES							
00 : C-0	0C : C-1	18 : C-2	24 : C-3	30 : C-4	3C : C-5	48 : C-6	54 : C-7
01 : C#0	0D : C#1	19 : C#2	25 : C#3	31 : C#4	3D : C#5	49 : C#6	55 : C#7
02 : D-0	0E : D-1	1A : D-2	26 : D-3	32 : D-4	3E : D-5	4A : D-6	56 : D-7
03 : D#0	0F : D#1	1B : D#2	27 : D#3	33 : D#4	3F : D#5	4B : D#6	57 : D#7
04 : E-0	10 : E-1	1C : E-2	28 : E-3	34 : E-4	40 : E-5	4C : E-6	58 : E-7
05 : F-0	11 : F-1	1D : F-2	29 : F-3	35 : F-4	41 : F-5	4D : F-6	59 : F-7
06 : F#0	12 : F#1	1E : F#2	2A : F#3	36 : F#4	42 : F#5	4E : F#6	5A : F#7
07 : G-0	13 : G-1	1F : G-2	2B : G-3	37 : G-4	43 : G-5	4F : G-6	5B : G-7
08 : G#0	14 : G#1	20 : G#2	2C : G#3	38 : G#4	44 : G#5	50 : G#6	5C : G#7
09 : A-0	15 : A-1	21 : A-2	2D : A-3	39 : A-4	45 : A-5	51 : A-6	5D : A-7
0A : A#0	16 : A#1	22 : A#2	2E : A#3	3A : A#4	46 : A#5	52 : A#6	5E : A#7
0B : B-0	17 : B-1	23 : B-2	2F : B-3	3B : B-4	47 : B-5	53 : B-6	5F : B-7

NOTE TABLE DOWN with C-3 as base note:

OCTAVE 0	OCTAVE 1	OCTAVE 2
## : RES	## : RES	## : RES
	68 : C-1	74 : C-2
	69 : C#1	75 : C#2
	6A : D-1	76 : D-2
	6B : D#1	77 : D#2
60 : E-0	6C : E-1	78 : E-2
61 : F-0	6D : F-1	79 : F-2
62 : F#0	6E : F#1	7A : F#2
63 : G-0	6F : G-1	7B : G-2
64 : G#0	70 : G#1	7C : G#2
65 : A-0	71 : A-1	7D : A-2
66 : A#0	72 : A#1	7E : A#2
67 : B-0	73 : B-1	7F : B-2