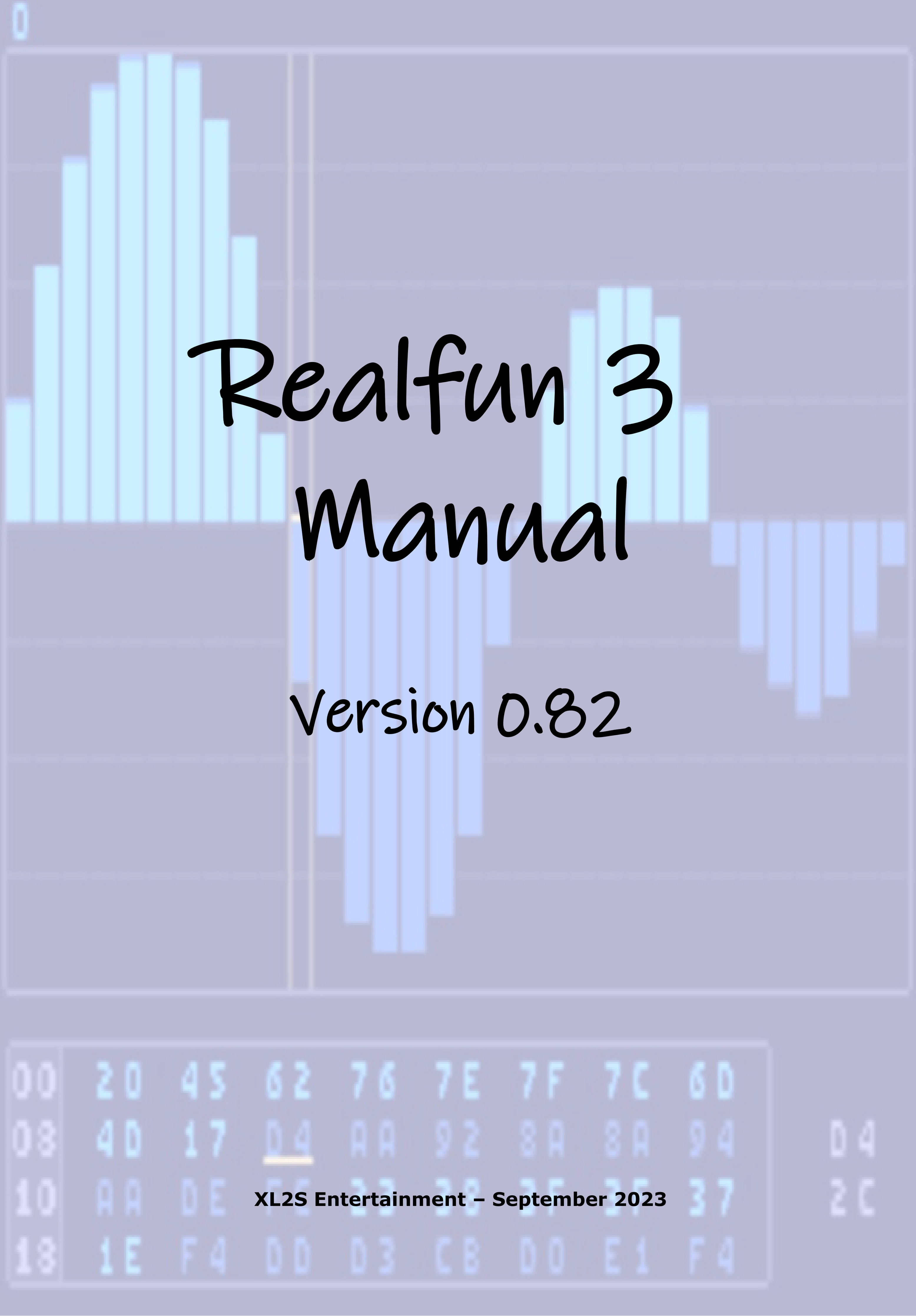


Realfun 3 Manual

Version 0.82

XL2S Entertainment – September 2023

Getting started

Realfun 3 is a music tracker under development for the SCC+, PSG and SN sound chip.

This document will provide an overview of the current features and functionalities.

Current version: v0.82

The software is in principle intended for emulator (openMSX), however should in theory work on a real MSX system, with the following minimum system requirements:

- MSX turboR (256 kb RAM)
- GFX9000 or video9000
- MegaFlashROM SCC+ (with SCC+ and PSG at port 010h) in slot 1
- SN chip (at port 049h)

In openMSX (version 16.0), simply select a turboR machine & video9000 & MegaFlashROM SCC+ & SN76489.
The volumes of the sound chips may need to be balanced a bit with openMSX Catapult. A volume balance in the tracker itself is not implemented.

Note that this product is still under development, nonetheless all features mentioned in this manual should work as advertised.

New in this version:

- Swap channel (cursor with block) using grph + s

Some known bugs or perks:

- interrupted playback (skipped notes) during editing (it will recover shortly after)
- changes to a list, during playback, will only filter through after a re-trigger event
- no undo functionalities
- copy functions of multiple patterns length will get noticeable slower (especially during playback – music might hang for a while).
- do not try to get sub steps in the tempo channel, it is not intended to be possible.
- **BUG** - at some point the test mode will not register all key inputs anymore... very annoying (but also muted channels will be muted in the test mode, check that first before hitting the keys harder and harder ;)
- **BUG** – extending a list (O,M, or V) in the list editor sometimes produces dirt on the end, by experience this is not harmful, just clean up (delete) manually when you get it.
- possibly still some other things ;)

In this document it is assumed the user will already have some basic understanding of the SCC and PSG.

For more details on the SCC you can have a peak at Trilo Tracker (http://trilobyte-msx.com/TriloTracker/index.php/Main_Page) -> manual and examples song from Gryzor87.
For more detailed info on the AY can have a peak at Arkos tracker 2 (http://www.julien-nevo.com/arkostracker/) tutorials -> understanding the AY.

Get the ISR SCC samples / voice encoder by ARTRAG here:
https://github.com/cornelisser/TriloTracker/files/6185847/tt_voicenc_scc_2021_03_23.zip

and for further details see MRC forum:
https://www.msx.org/forum/msx-talk/development/voice-synthesis-on-isr?page=31

Have fun!

Pattern editor

Pat = current pattern number

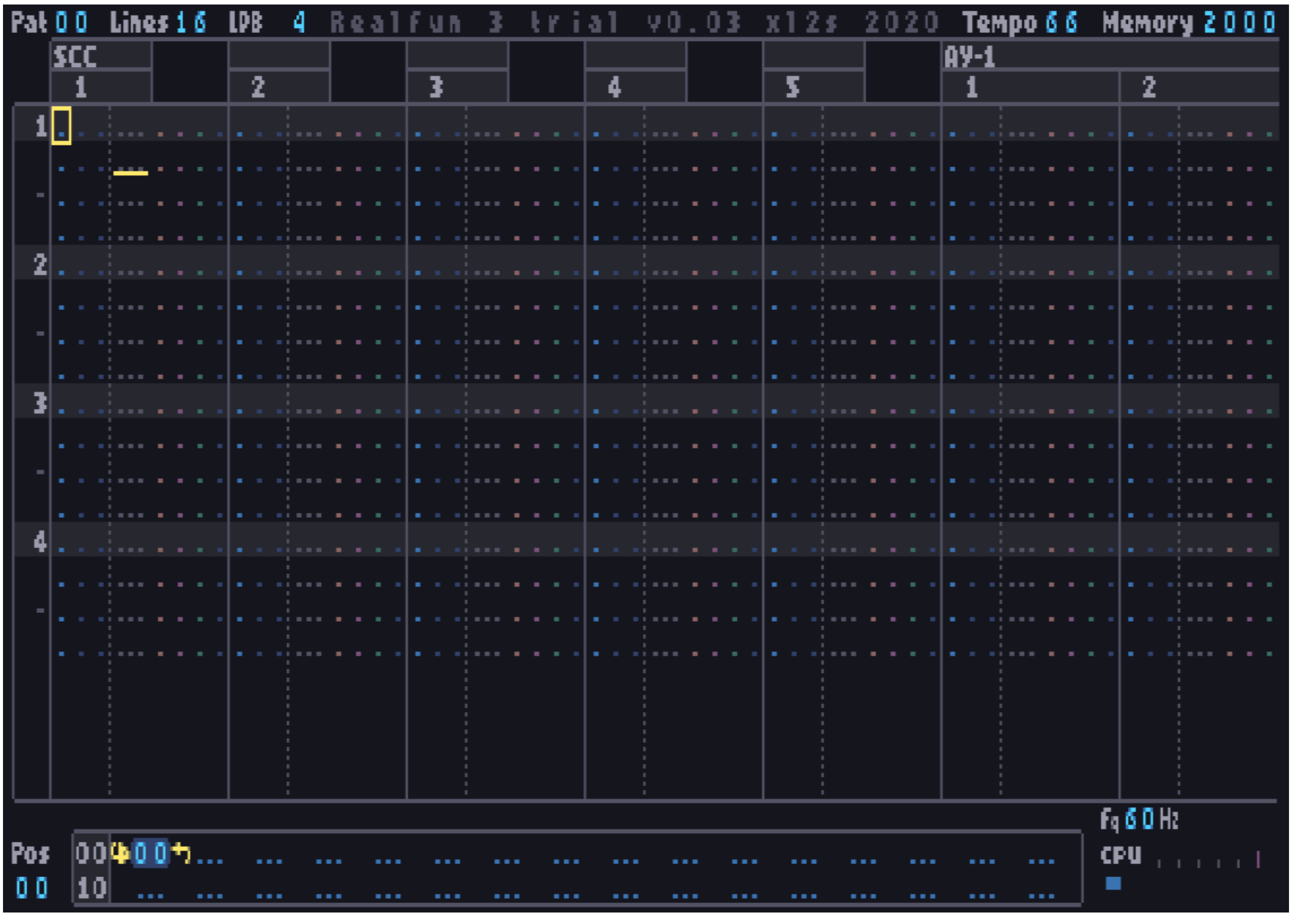
Lines = number of lines for current pattern

LPB = lines per beat highlight for current pattern

Song Title (32 characters)

Start tempo

song buffer
free memory



Pat 00 Lines 16 LPB 4 Realfun 3 trial v0.03 x12s 2020 Tempo 66 Memory 2000

Pos 00

Fq 60 Hz

Position editor with start position, loop position and end position, see position editor for further details.

Playback frequency
& CPU use indicator

Pattern editor displays from left to right the following sound chips and channels:

SCC	5 x tone (mega cart SCC+)
AY-1	3 x mix/tone + 1 x noise freq. + 1 x envelope freq. (mega cart PSG – 010h)
SN	3 x tone + 1 x noise/mix control
AY-2	3 x mix/tone + 1 x noise freq. + 1 x envelope freq. (MSX onboard PSG – 0a0h)
Tmp	1 x tempo control

Max number of patterns 256, numbered hexadecimal 00 to FF.

Each pattern can have a length from 1 to 64 lines (decimal) and for each, a beat hi-light can be specified from 1 to 16 (decimal). The after beat is automatically indicated with the " - " symbol (or double pair if no line halfway the beat exists).

Adjusting the length or LPB of multiple patterns at once is possible by using the block functions.

Note: When reducing pattern length, the song list data in a pattern is not altered or deleted even so in case it falls outside the new length. Hence, after adjusting pattern length by *mistake the song data will not be lost (and will also be saved along), however it will not be played or visible.*

Pattern editor Keys overview:

(for specific column, effect, note, etc input see specific support sections)

cursor keys	move cursor (left or right + shift double speed)
F1	start song playback
F2	switch to sample editor
F3	switch to list editor
F4	switch to SCC wave / list editor
F5	disk menu
esc	stop song playback
tab	switch to position editor
select	toggle “show ornament ” mode
	note: this allows ornament input “behind” pitch bend effects, but also enables note input (C-0 to B-8) in the PSG envelope frequency column in stead of a 16 bit number.
ctrl + select	toggle “hide sub steps” display mode
del	delete entry below cursor
shift + del	delete complete step – this also deletes a sub-step
ret	repeats last input for a specific column
[	repeats last input – 1
]	repeats last input + 1
	note: currently only works for the NOMEV, Nf, Ef columns, “wave & mix” columns are not supported.
= or n	toggle note attack (gray note = attack, pink note = no attack)
ins	insert empty step, following steps in pattern are rolled down
bs	delete empty step, following steps are rolled up.
	note: only if a step goes beyond line 64 it will be deleted (if pattern length is less than steps may not be visible but are still there. Steps “below” current pattern may be pulled back in using [bs])
shift + ins	insert sub-step above current line
	note: to enter a sub-step below the last line the cursor should be place on the empty line below.
shift + t	adjust sub-step timing (01 to 99 - decimal)
	note: multiple sub-steps, nicely spaced, can be inserted by using a block function
<	transpose notes in channel down by 1 semitone
>	transpose notes in channel up by 1 semitone
ctrl + <	transpose notes in channel down by 1 octave
ctrl + >	transpose notes in channel up by 1 octave
{	decrease volumes in channel by 1
}	increase volumes in channel by 1
crtl + left right	walk through patterns with step size 1
crtl + up down	walk through patterns with step size 4
ctrl + p	go to pattern (input: 00-FF)
ctrl + l	specify current pattern number of lines (input: 0 to 64)
ctrl + b	specify current pattern lines per beat hi-light (input: 0 to 16)
ctrl + n	specify song title (input: 32 characters long)
ctrl + t	specify start tempo (input: 11-FF, see time for further details)
ctrl + f	toggle playback frequency 50 Hz / 60 Hz (= equalizer on/off)
	note: this setting is saved along with the song!

Pattern editor – Channel muting (on/off)

ctrl + 1-8 with cursor positioned in SCC or AY-1 channel: 1-5 => SCC, 6-8 => AY-1
with cursor positioned in SN or AY-2 channel: 1-4 => SN, 6-8 => AY-2

Pattern editor – Block functions:

q set upper left corner of block
w set lower right corner of block

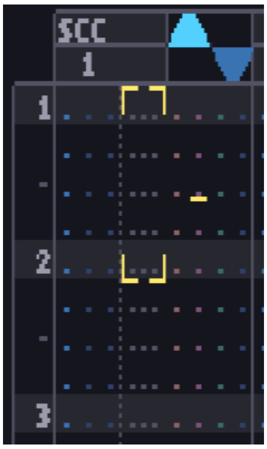
ctrl + c copy block to "clip board"
ctrl + ret paste block from "clip board" to song buffer with overlay
ctrl + shift + ret paste block from "clip board" to song buffer with clear

note 1: adjusting pattern lengths, load or save song, will invalidate the clipboard; see icon in upper left corner, if valid clip board data is present a blue square is shown.
note 2: paste with clear will also clear steps that are incompatible, care should be taken copying data between different channel types (especially Ef, Nf & SN 4)

grph + , transpose notes in block down by 1 semitone
grph + . transpose notes in block up by 1 semitone
grph + < transpose notes in block down by 1 octave
grph + > transpose notes in block up by 1 octave
grph + [decrease volumes in block by 1 (*alternative left alt + 9*)
grph +] increase volumes in block by 1 (*alternative left alt + 0*)

grph + del delete content of block
grph + del + shift delete complete steps and all sub-steps covered by block

grph + ins adds specified number of sub steps between start step to end step of block, as per following example:



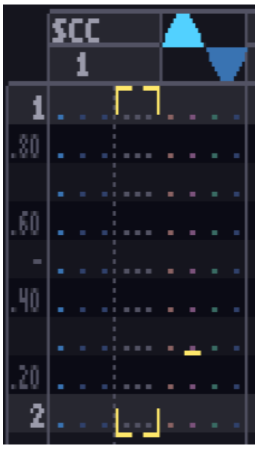
1. Select beat 1 up to and including beat 2, as per picture on the left.

2. press [grph]+[ins]

3. enter "5" in the popup followed by [ret].

4. result is as per picture on the right, and it is now possible to place five equally* spaced notes at the place of four.

* *tempo permitted round off to nearest 1/60 or 1/50 of a second.*



Note:
1. this operation works only on one channel at a time and covering only one pattern, but sub-steps can be copied. Also, entering a new note on a empty spot in another channel will create a sub-step for that channel with the same timing.
2. at the end of a pattern the end of the block can be set on the empty line below last line in order to have a smooth transition to the 1st step of the next pattern.

grph + l set pattern number of lines for all patterns from block start pattern to block end pattern. (input: 0 to 64 in popup)

grph + b set pattern lines per beat for all patterns from block start pattern to block end pattern. (input: 0 to 16 in popup)

grph + s swap channel under cursor with channel under block (1 channel only) for all patterns from block start pattern to block end pattern. Do not use this for channels between different sound chips (compatibility issue ;)

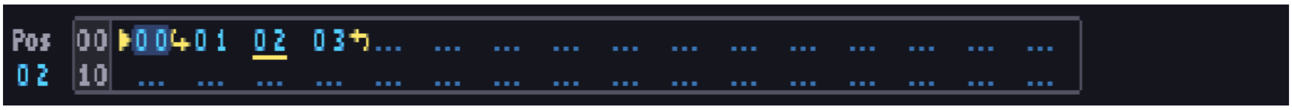
Position editor

Pos = cursor position in position editor or current position during playback (blue highlight)

Play start position

Loop position

End position



Patterns are ordered in the position editor. Up to 256 positions can be specified.

<u>Position editor keys input overview:</u>

F1	start song playback
esc	stop song playback
tab	switch to back pattern editor
cursor keys	move around in position table
n	add pattern with a sequential number
0-F	number input
ins	insert pattern number (defaults to 00)
del	delete pattern number
bs	- not implemented yet -
s	set start position
,	set loop position
.	set end position

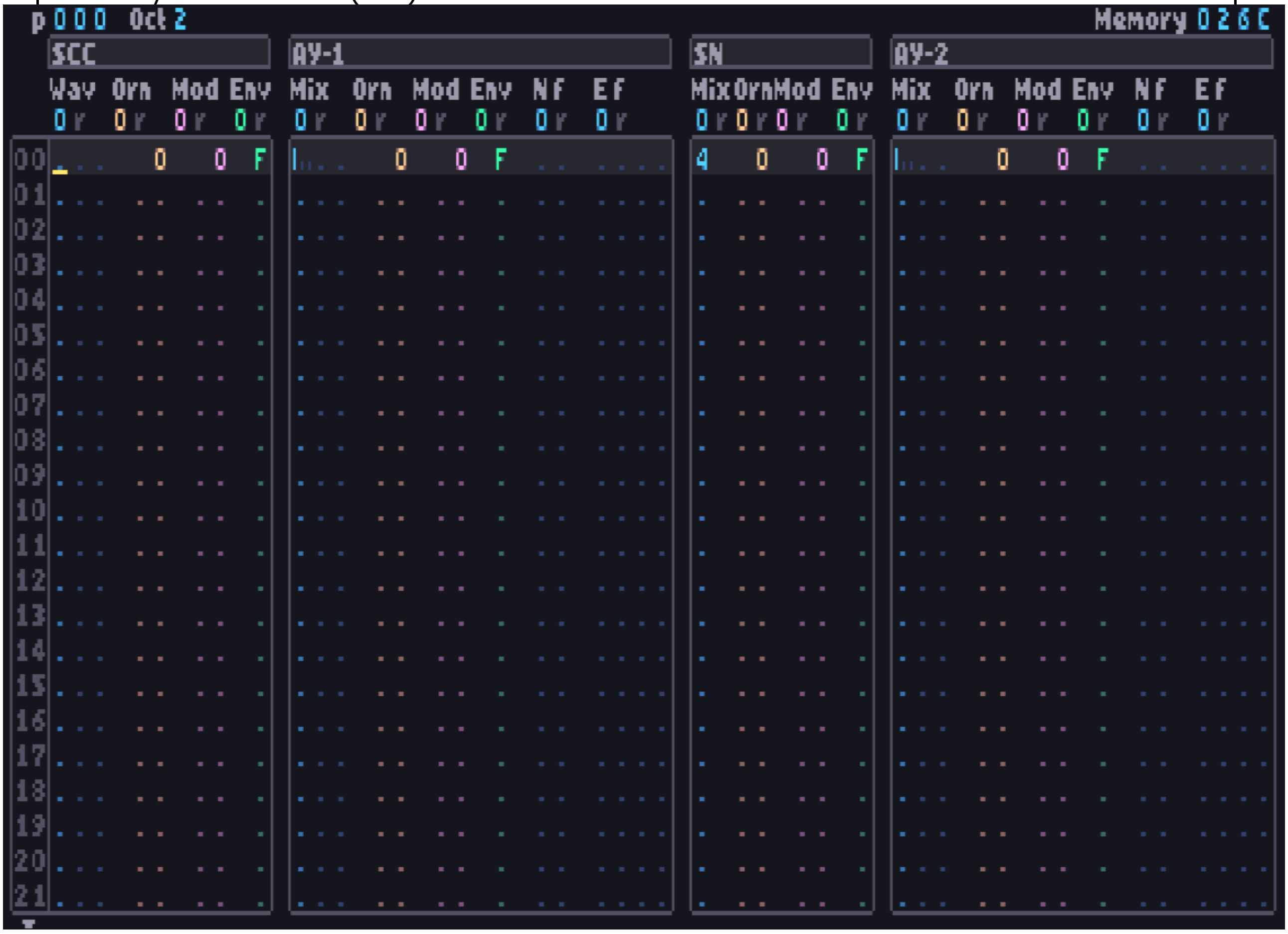
note: end position cannot be set in front of loop position and loop position can not set after end position

List editor

SCC parameter for sound test (<fx0>, <wave1>, <wave 2>)

Keyboard octave (0-7)

Free list data memory



Column ID	Description
Wav	SCC wave list
Orn	ornament list
Mod	modulation list
Env	volume envelope list
Mix (AY)	PSG mix control list
Nf	PSG noise frequency list
Ef	PSG hardware (amplitude modulation) envelope frequency list
Mix (SN)	SN noise / tone mix list

Each sound chip can have up to 16 list of each type. Number of current list is indicated on the top, 0 to F (hexadecimal). Next to the list number the “list trigger” status is shown, where “r” denotes re-triggered at note attack event and “-” not re-triggered at note attacker event.

The maximum length for each list is 0 to 39 lines for Wav, AY Mix and AY Ef lists, and 0 to 94 lines for the other.

It should be noted that only the wave list has a list run timer, all other lists run at full speed (1/60 s or 1/50 s per step). Depending the list type empty steps can be used or otherwise the same value should be repeated in order to 'slow down' the list.

List editor general keys overview:

F1	-
F2	switch to sample editor
F3	(switch to list editor)
F4	switch to SCC wave / list editor
F5	disk menu
esc	return to pattern editor

cursor keys:	move cursor
ctrl + left/right	walk through lists

0 – F	input value
r	toggle current list “trigger” status
l	set loop position
-	toggle sign
yen (or \)	toggle “update base” mode – only for Nf and Ef

ins	insert new line (as long a max length is not reached)
del	delete or default entry underneath cursor
bs	delete current line (roll up)

ctrl + c	copy list to “clip board”	(this will not effect other clip boards)
ctrl + ret	paste list from “clip board”	

Note 1: lists types of source and destination must be the same.
Note 2: load or save will invalidate the clipboard

shift + F1	lower keyboard octave (or [4] in test mode)
shift + F2	raise keyboard octave (or [8] in test mode)

space	toggle test mode; see test mode section (Note: currently only SCC supported)

Common lists:

Orn	Ornament list	input range: -64 to 63 (decimal) (offset in semitones)
Mod	Modulation list	input range: -63 to 63 (decimal) (offset on period, Tp)
Env	Volume Envelope list	input range: 0 to F (hexadecimal) with or without sign

Note: if sign is set the volume will be reduced by 1 on every pass. This decrement is applied immediately, hence -F will be the same as volume E.

Sound chip Specific lists:

SCC wave list	input: similar to pattern input, see SCC section

AY noise freq list	input: -1F to 1F (hexadecimal) (with or without base update)
AY envelope freq list	input:-FFFF to FFFF (hexadecimal) (with or without base update)

Note 1: regular working of these AY list is as follows: the value specified in this list is added to the base value specified in the pattern. With the yen (or \) key the update base mode is toggled, this is visualized with a gold + or – sign, then the new value will overwrite the previous base value. This can be used for frequency slides.
Note 2: empty spaces in the list are permitted.

AY tone/noise/hardware mix list	input: similar to pattern input, see AY section

SN noise/tone mix list	input: see SN section

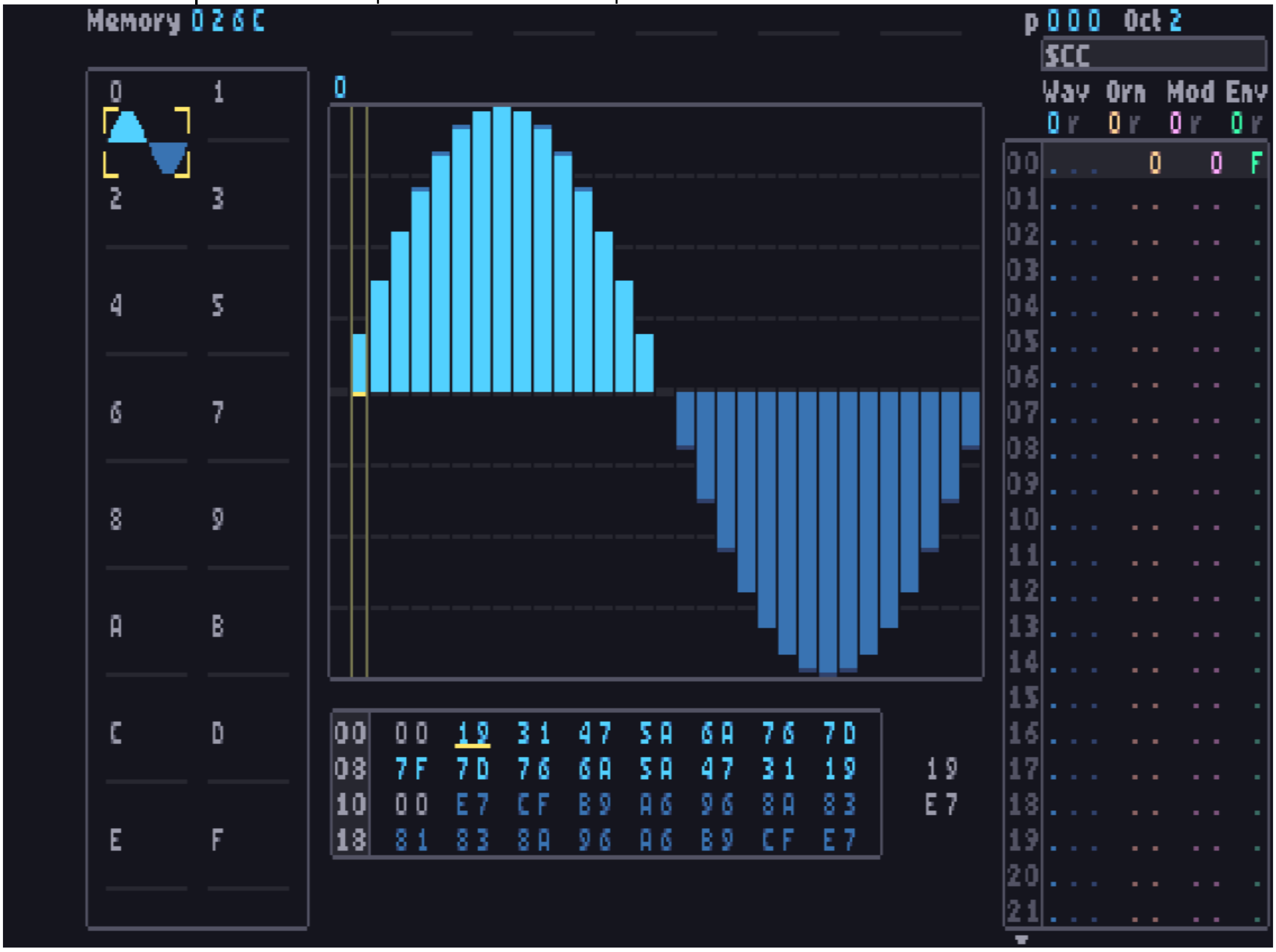
SCC Wave editor

Free list data memory

SCC wave edit field

on top: channel wave from previews & current edit wave number

(see list editor)



```
00  00 19 31 47 5A 6A 76 7D
08  7F 7D 76 6A 5A 47 31 19     19
10  00 E7 CF B9 A6 96 8A 83     E7
18  81 83 8A 96 A6 B9 CF E7
```

Current song SCC waves

SCC wave in hexadecimal values (32x)

current value
negated value

A song can have up to 16 wave forms, 0 to F.

Wave form F is a special case, which can be overwritten by the song during playback, see wave effects for further details.

Note: using this functionality it is for instance also possible to generate a new waveform, without manually editing. For example: using a multiplier operator with factor 2 a double octave sine wave can be created from above sine wave.

For loading, saving or exchange of voices an SCC voice bank can be opened up with [F4] or [select]. For details see voice bank section.

Wave editor keys overview:

(for list editor specific key inputs see list editor section)

F1	-
F2	switch to sample editor
F3	switch to list editor
F4	open voice bank
F5	disk menu
esc	return to pattern editor
tab	switch between SCC list editor and SCC wave editor (this screen)

space	toggle test mode; see test mode section

ctrl + cursor keys	walk through wave forms (0 to F) on	
left/right	walk through the wave (0 to 31)	(wrap around)
left/right + shift	make selection	(move without shift to break selection)
up/down	adjust value step size 4	(single or selection)
up/down + shift	adjust value step size 1	(single or selection)

0-F	direct value input (hexadecimal)

del	value to zero (00)
ret	repeat last value
,	roll wave left
.	roll wave right
;	roll wave “up”
/	roll wave “down”

s	smooth single value	(weighted average of 5 columns)
n	negate value	(single or selection)
k	copy value from left column	
l	copy value from right column	
[	follow trend from two columns on the left	
]	follow trend from two columns on the right	
x	copy negated value from cursor position + half wave	
z	point mirror value about wave center	

ctrl + ;	scale wave amplitude up	(full wave or selection)
ctrl + /	scale wave amplitude down	(full wave or selection)
ctrl + s	smooth wave	(full wave or selection)

ctrl + n	negate full wave
ctrl + m	mirror full wave

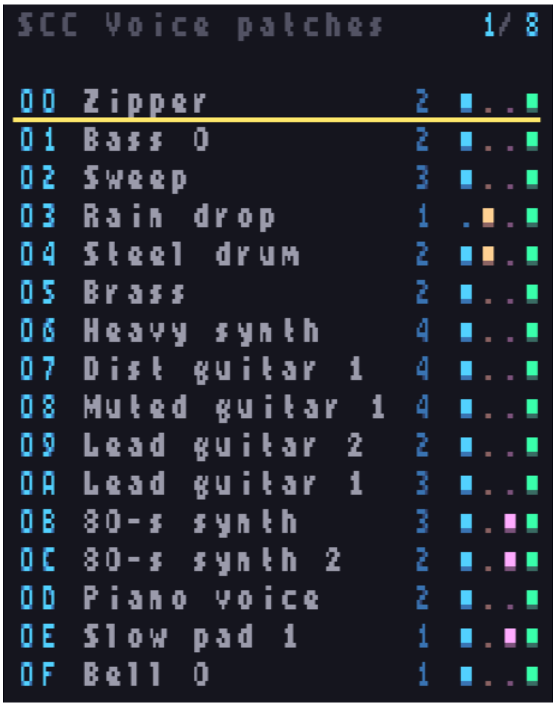
ctrl + c	copy current wave to clipboard	(this will not effect other clip boards)
ctrl + ret	copy clipboard to current wave	

ctrl + o	generate basic sine wave

SCC voice bank

An example voice bank is shown below:

voice bank page no.



```
SCC Voice patches      1/ 8

00 Zipper              2 ■..■
01 Bass 0              2 ■..■
02 Sweep               3 ■..■
03 Rain drop           1 .■.■
04 Steel drum          2 ■■.■
05 Brass               2 ■..■
06 Heavy synth         4 ■..■
07 Dist guitar 1       4 ■..■
08 Muted guitar 1      4 ■..■
09 Lead guitar 2       2 ■..■
0A Lead guitar 1       3 ■..■
0B 80-s synth          3 ■.■■
0C 80-s synth 2        2 ■.■■
0D Piano voice         2 ■..■
0E Slow pad 1          1 ■.■■
0F Bell 0              1 ■..■
```

voice no.
voice name (14 chars)

indicators for lists used
(wave, ornament, modulation and envelope)
number of wave forms used

A voice bank can hold up to 128 voices (16 per page). Each voice can have up to four (4) basic wave forms, a wave effect list, ornament list, modulation list and a envelope list.

A voice can be tried out by standing on a voice and then simple use the “piano keys” to play it.

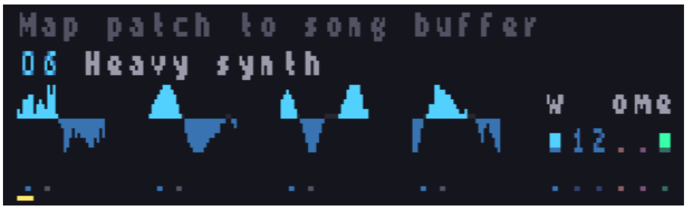
It should be noted that the voice bank is only a storage medium, voices cannot be used directly in the song but need to be patched. This is a manual procedure! Patching the individual components of a voice from or into a voice bank is addressed in more detail below.

<u>Keys overview:</u>

esc, tab	return to wave editor
cursor keys:	move cursor
ctrl + up/down	move voice entry up or down; i.e. allows for reordering of voices
“piano keys” : (see voice test section)	sound test of voice
ins	insert / store new voice
space	select voice
shift + F1	lower keyboard octave (or [4])
shift + F2	raise keyboard octave (or [8])

Select voice from voice bank

When selecting a voice with [space] the following patch window pops up. The window will show the used wave forms of a voice and indication on which lists are used (flags).



```
Map patch to song buffer
06 Heavy synth
                                        W  ome
                                        ■12..■
```

voice no. and name:

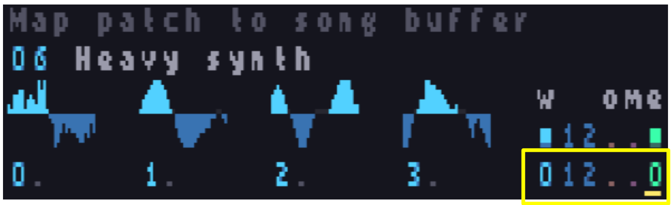
voice wave forms 0-3:

patch (to) row:

List id
(wave, orn, mod & env)

List used flags

Now, to patch this to the song buffer wave forms and lists the destination number can be specified on the patch row, for example:



```
Map patch to song buffer
06 Heavy synth
                                        W  ome
                                        ■12..■
0.      1.      2.      3.              012..0
```

voice no. and name:

voice wave forms 0-3:

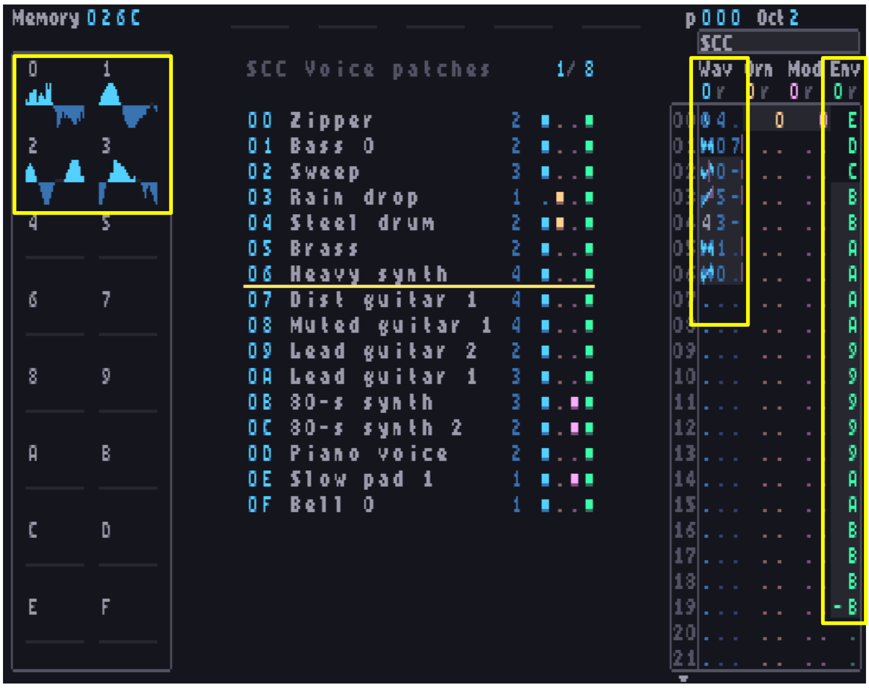
patch (to) row:

List id
(wave, orn, mod & env)

List used flags

When all numbers are specified, it should be noted that the numbers in the yellow box (above) is a preview on the how the voice can be reproduced in the pattern editor (so note it down ;).

Subsequently pressing [ret] will patch the voice wave forms and lists to the song buffer:



```
SCC Voice patches        1/ 8

00 Zipper              2 ■..■
01 Bass 0              2 ■..■
02 Sweep               3 ■..■
03 Rain drop           1 .■.■
04 Steel drum          2 ■■.■
05 Brass               2 ■..■
06 Heavy synth         4 ■..■
07 Dist guitar 1       4 ■..■
08 Muted guitar 1      4 ■..■
09 Lead guitar 2       2 ■..■
0A Lead guitar 1       3 ■..■
0B 80-s synth          3 ■.■■
0C 80-s synth 2        2 ■.■■
0D Piano voice         2 ■..■
0E Slow pad 1          1 ■.■■
0F Bell 0              1 ■..■
```

It could happen that some wave forms are reused or simply duplicate. In that case the same destination waves may be used.

There is also an option to prevent the wave form to be copied by specifying [n] at the specific wave form on the patch row. This allows a voice to be patched using other / existing song buffer wave forms; which may lead to the discovery of new interesting sounds.

After the voices are patched to the song buffer they can be customized to suit your specific needs for the song (e.g. adjust envelope (ADSR), wave update timers etc.)

Keys overview:

cursor keys: move cursor (left right)

0-F enter wave form or list value
n toggle copy or no copy
del delete wave form or list value

esc cancel
ret confirm / patch voice

ctrl + n change voice name

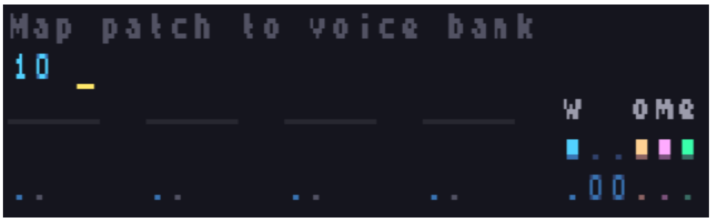
Store voice in voice bank

In order to store a new voice in the voice bank, go to the desired entry (empty voice or overwrite existing) and press [ins]. The following patch window pops up.

voice no. and name:

voice wave forms 0-3:

patch (to) row:



List id
(wave, orn, mod & env)

List used flags

First the voice name should be specified.

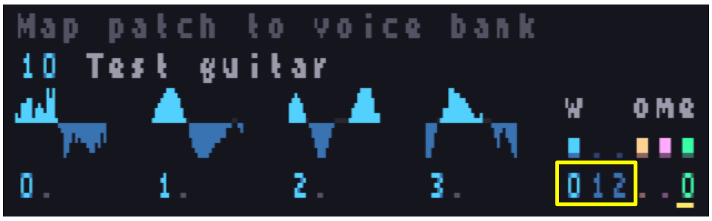
After that the wave forms and lists that make up the voice can be specified on the patch row *(also be aware of wave forms that may be used in the wave list!)*. Lists and wave forms not used should be left blank.

Finally the <wave1> and <wave2> parameters should be specified, see yellow box below, which is simply the SCC specific part of the step in the pattern editor (if not relevant leave it 00).

voice no. and name:

voice wave forms 0-3:

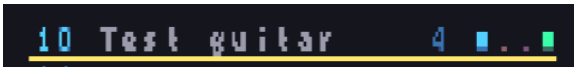
patch (to) row:



List id
(wave, orn, mod & env)

List used flags

Subsequently pressing [ret] will patch the voice wave forms and lists into the voice bank:



Keys overview:

cursor keys: move cursor (left right)

0-F enter wave form or list value
del delete wave form or list value

esc cancel
ret confirm / patch voice

ctrl + n change voice name

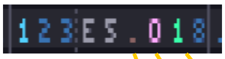
SCC voice test mode

The SCC voice test mode can be entered from the wave or list editor by pressing [space]. When active the word "Test" will appear in the upper right corner of at the top middle of the screen.

Once activated the "piano keys" on the keyboard may be used to play the voice.

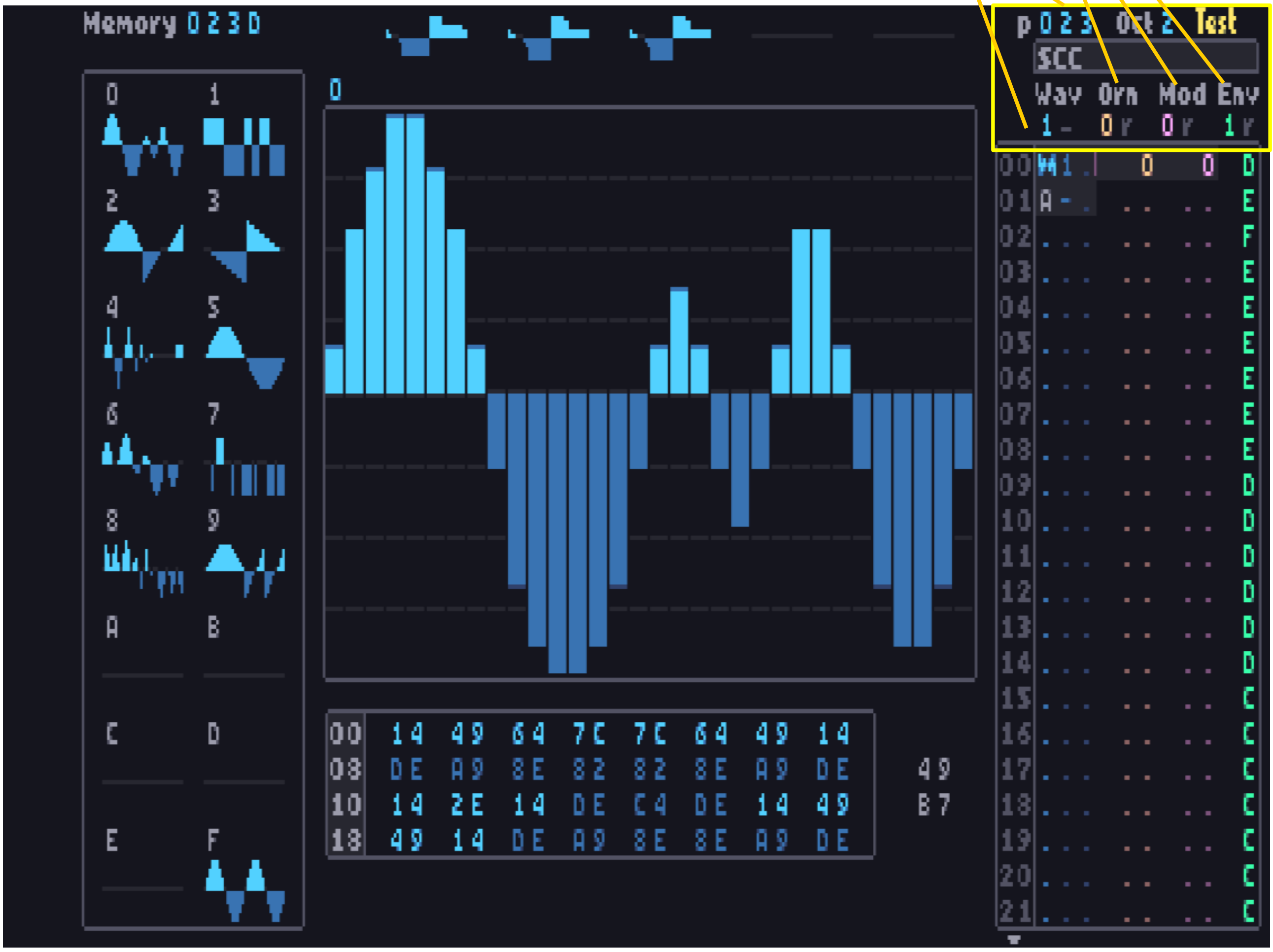
For the test voice the active lists (wave, ornament, modulation and envelope) and the SCC parameter (<fx0>, <wave1> and <wave 2>) settings will be used; the later can be entered by pressing [ctrl] + [p] in the list editor (but not when test mode is active).

Voice from pattern editor:



```
123E5.018
```

Testing the same voice in the test mode:

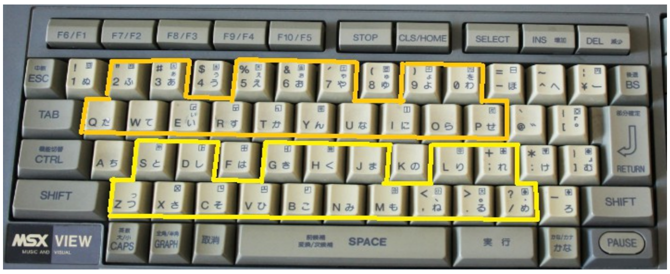


```
00  14 49 64 7C 7C 64 49 14
08  DE A9 8E 82 82 8E A9 DE    49
10  14 2E 14 DE C4 DE 14 49    B7
18  49 14 DE A9 8E 8E A9 DE
```

```
00 W1 .    0   0  D
01 A- .    ..  .. E
02 ...     ..  .. F
03 ...     ..  .. E
04 ...     ..  .. E
05 ...     ..  .. E
06 ...     ..  .. E
07 ...     ..  .. E
08 ...     ..  .. E
09 ...     ..  .. D
10 ...     ..  .. D
11 ...     ..  .. D
12 ...     ..  .. D
13 ...     ..  .. D
14 ...     ..  .. D
15 ...     ..  .. C
16 ...     ..  .. C
17 ...     ..  .. C
18 ...     ..  .. C
19 ...     ..  .. C
20 ...     ..  .. C
21 ...     ..  .. C
```

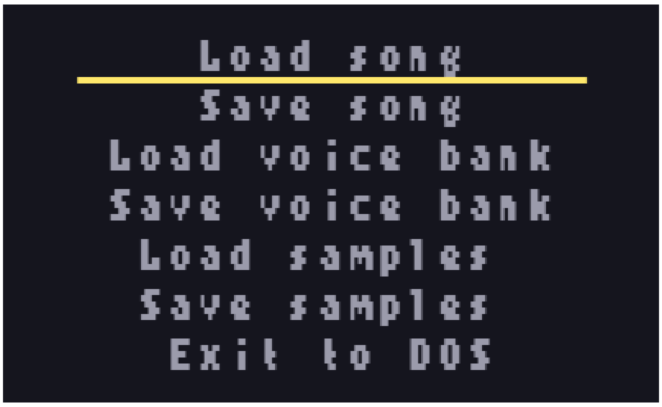
Note that for editing and tuning voices it may be easier and more effective to compose a small loop and let it play whilst editing the voice, instead of switching back and forth all the time.

Keys overview:



esc, tab, space	exit test mode
"piano keys" (see right):	sound test of voice
shift + F1	lower keyboard octave(or [4])
shift + F2	raise keyboard octave (or [8])

Disk menu

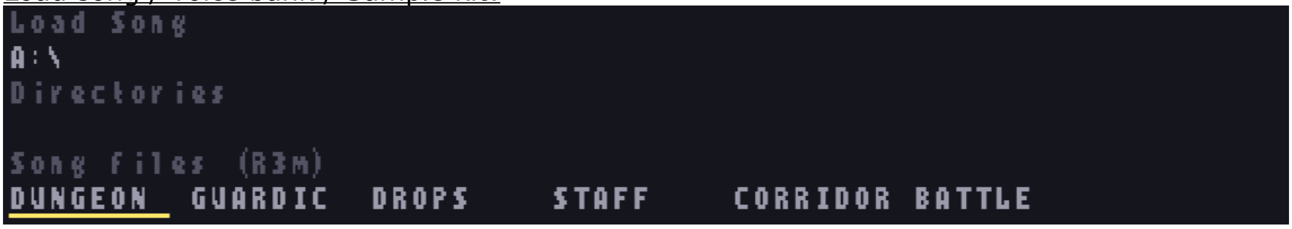


Currently 7 options available: load / save song, load / save voice bank, load / save sample kit and exit to DOS. A brief overview is given below.
It should be noted the error support is fairly limited, and disk functionalities have not been tested on real MSX, so there is no guarantee it will work if an error occurs. Also changing drives is currently not implemented.

Keys:

cursor keys	select option
ret or space	choose option
esc	return to pattern editor

Load song / voice bank / Sample kit:



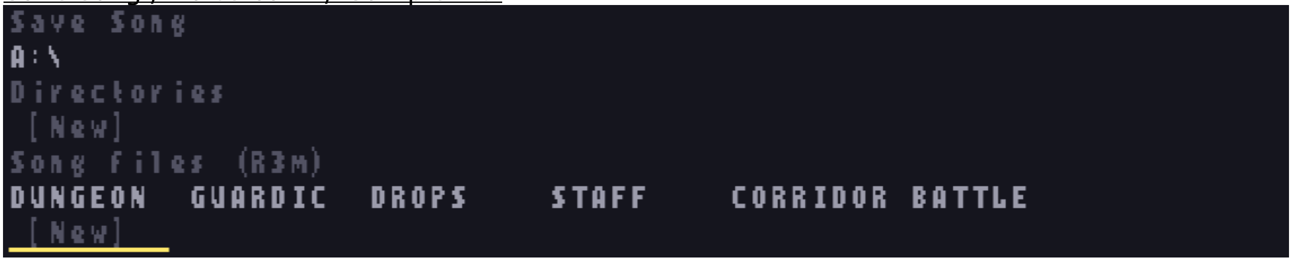
```
Load Song
A:\
Directories

Song files (R3m)
DUNGEON  GUARDIC  DROPS    STAFF    CORRIDOR BATTLE
```

Keys:

cursor keys	select file or directory
ret or space	load file or enter directory
F5	refresh
esc	return to pattern editor

Save song / voice bank / Sample kit:



```
Save Song
A:\
Directories
 [New]
Song files (R3m)
DUNGEON  GUARDIC  DROPS    STAFF    CORRIDOR BATTLE
 [New]
```

Keys:

cursor keys	select file, directory, [new] file or [new] directory
ret or space	save file over exiting file, enter directory or create new directory or save new song / voice bank file
F5	refresh
esc	return to pattern editor

If [New] is selected a 8 character long file name or directory name should be specified.

Exit to DOS:

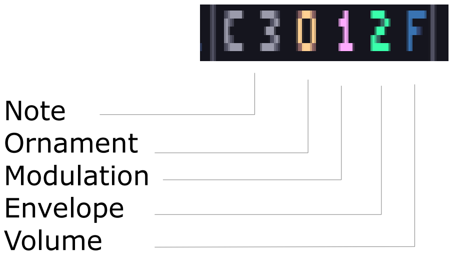
speaks for itself ;)
note: when using video9000 in openMSX screen switching will be automatically.

General step layout

All tone channels for the three sounds chips will have a chip specific part and a generic part, where the latter is also denoted as NOMEV. This generic part is described here below, and input and behavior is the same for all. The chip specific parts are described in the subsequent, chip specific sections.

Furthermore, on the far right of the pattern, there you can find the tempo control channel. A description on this channel is also given here below.

Generic step layout



Note column

Notes can be inserted in the note column in the range from C-0 to B-8.

> Note: first type in the note (A to G) followed by the octave (0 to 8). To type # notes press the note key twice before specifying the the octave (for example press [a] [a] [4] for A#4).

It should be noted that the valid range for SCC and AY tone notes is A-0 to B-8, except for hardware envelope tones where C-0 may be specified. On the SN the applicable range is even more restricted from A-2 to B-8, with exception of periodic noise tones which can go all the down way to C-0.

When placing the cursor on the note, the “note attack” can be toggled with the [=] or [n] keys. Notes with “attack” are displayed in gray, without are pink.

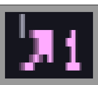
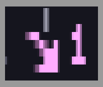
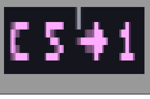
A note off event can be inserted with the [o] key.

Ornament column

In this column a ornament list, 0 to F, can be specified. Ornament can be used to create ornaments, fast (note) slides and arpeggio.

Modulation column

On the modulation column a modulation list can be specified with 0-F, or a pitch bend effect can be inserted, where:

Key	Symbol	Description
i		Pitch bend up
o		Pitch bend down
p		Portamento (pitch to note) only possible in combination with a “no attack” note *note: when inserted a portamento the note attack is automatically updated*

Timer: 0 to 6 (gray number)   Increment: 0 to F (pink number)

Note that the pitch bend symbol will be place above the ornament column. With [select] the pitch bend effects can be turned off and the hidden ornaments can be uncovered and edited.

The pink number in the modulation column is the increment /decrement value, which can be specified 0 to F. With 0 the pitch will be hold.

The small number in gray, above the symbol, is the pitch bend timer. This timer can be adjusted by simply placing the cursor on the symbol (in the ornament column) and specifying the new value with 0 to 6, fast to slow respectively. Timer 1 will be set by default, and for most cases the velocity can be controlled adequately by changing the increment. For higher frequency ranges an increment of 1 might still be too fast, for this the timers 2 to 6 are available. Timer 0 is mainly intended for pitching in the lower frequency ranges (< octave 2).

Pitch bend effect are only stopped if there is a note event, note off event or, in case of portamento, the target pitch is reached.

During the pitch bend the modulation list may be changed, but note this will only come into effect once the pitch bend effect is finished or stopped.

Envelope column

In this columns a volume envelop list, 0 to F, can be specified. Note that this effect simulates the attack, decay and/or sustain part of an instrument.

Volume column

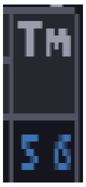
In this column the channel volume, 0 to F, is specified.

General remarks on NOMEV:

1. *It should be noted that Ornament-, Modulation- and Envelope lists are always active, and cannot be switched off. It is therefore highly recommended to reserve lists no. "0" as a no effect list for ornaments and modulations specifically.*
2. *Note events, except "no attack" notes, will generate an attack event for the channel. This event will re-triggers all the lists in that channel; unless the trigger setting of the list is set to no re-trig.*
3. *However, if a list is specified, this list is always (re-)triggered. Hence, if for example an envelope list is specified along with a "no attack" note, the list is re-triggered anyway and will sound just the same as an "attack" note.*

Tempo control column

Tempo can be specified for odd steps and even steps independently.



Tempo odd steps (1st, 3rd, 5th, etc.)

Tempo even steps (2nd , 4th, etc.)

Tempo is specified by the the amount of ticks (units of 1/60 s) until the next line is played. Note that this does not include sub lines which will be played in between two regular lines (*permitted all sub line notes can be played within the time frame, if it does not fit all following notes will be delayed)*. The input range is 1-F (hexadecimal) each, tempo 0 is not permitted.

The start time setting is identical, this can be set by [crtl]+[t] in the pattern editor.

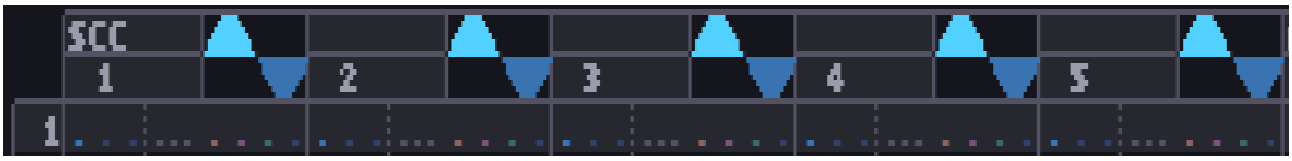
Signals

The time column can also be used for "signals". These can be inputted with [s] key followed by a hexadecimal value 0-F. This signal/value can later be read & used by the user in his own program to sync certain events to the music (e.g. timing of demo cut scenes, or characters start dancing around if a certain part of the song is playing etc. etc).

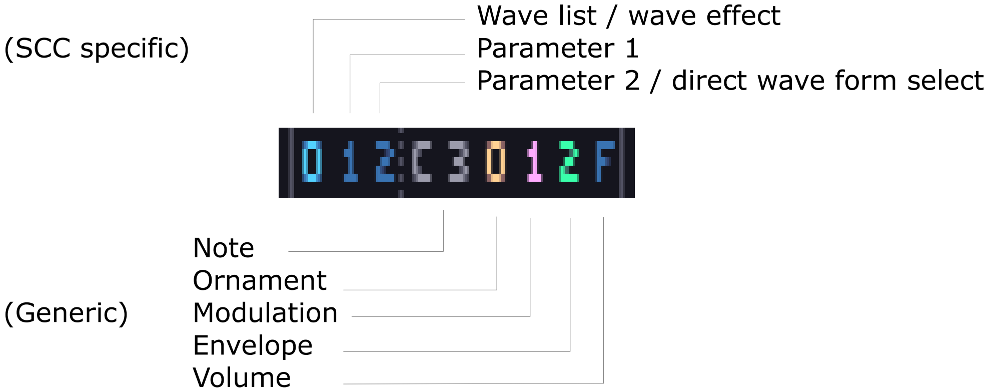
SCC SUPPORT

The SCC-I sound chip has five channels, of which each can have it's own (32 bytes) programmable wave form. In the pattern editor, the current playing wave form for each channel is shown at the top. These are updated in real time during playback, which give a nice visualization on the various wave morphing effects that can be applied.

The SCC channels are first in the editor on the left, as per figure below:

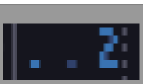
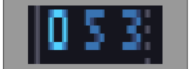
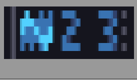


The step layout is identical for all five channels, as below:



Note that the generic part is described in the previous section.

With the first 3 columns in the SCC step, wave forms can be selected and or wave effects can be applied. For this there are three options:

Option	Step	Description
1		Direct wave select: first field is empty, second field is <wave 1> select (optional), third is <wave 2> select and SET => wave is send to SCC. In this case wave form '2' is set.
2		Use wave list: first field specifies the wave list no. (0 to F), second field is <wave 1> select (optional), second field is <wave 2> select (optional). Wave list is activated and wave updates to the SCC will be as defined in the wave list until list is completed or stopped.
3		Direct wave effect: first field specifies the wave effect (each effect is represented by a symbol), second field is effect <parameter 1> (optional), second field is effect <parameter 2> (optional).

Wave effects are in typically limited to one instance only. But no worries, for more complex effects, a wave list (option 2) can be used. In a wave list, multiple effects can be specified in sequence and/or in a loop.

The input for a wave list and direct wave effect input in the pattern editor, is very similar. There are a few small differences, which is mainly in the handling and updating process of the internal parameters. This is explained below but will probably become clear in the examples later on.

Internal parameters

Each SCC channel has the following internal parameters, which are used in various effects and/or for the wave list:

<timer>	List timer (0-8); determines the wave list run speed.
<wave 1>	Wave form 1 parameter (0-F; index to a wave form)
<wave 2>	Wave form 2 parameter (0-F; index to a wave form)
<fx0>	Wave fx parameter (0-F)
<offset>	Wave 'roll' offset parameter (0-31)
<buffer>	Internal wave form buffer

It is important to know that effects will take their input from these internal parameters, but not all can be set along with the effect. For example the “interference” effect uses <wave 1>, <wave 2> and <offset>, but only <offset> can be specified along with the effect. This means that <wave 1> and <wave 2> need to be specified in advance. Or, as a keen observer may have already noticed, <wave 1> and <wave 2> can be specified along with a list.

If an effect has an internal parameters as input, by leaving it blank the effect will use the internal parameter directly. This is the same for the wave list as for in the pattern.

The difference comes into play when a new input is specified. For an effect given in the pattern, the internal parameters are updated accordingly. This is handy for consecutive effect input as not all parameter have to be specified at each instance. In the wave list however, the internal parameters are <u>not</u> updated and the specified value will only be used for that instance only. This is done so that a wave list could be used in combination with different wave forms.

The exception to the above are the “-” (use <buffer> wave form) and “s” (set wave form) operators, these need to be (re-)specified every instance.

The fx parameter <fx0>, is intended to control certain effects in a list from outside the list. This gives more flexibility, but also potentially lead to a lot more wave data. For the editor this is not a problem, but later on in the re-player it is possible there will be some restrictions on total wave data.

The <offset> parameter, is used for “interference” which is a roll offset (or phase offset) for <wave 1> in 32 steps.

The <buffer> is the internal wave form buffer. After each effect the <buffer> is updated with the latest result. The <buffer> is by default send directly to the SCC. However, it is also possible to stack and place effects in sequence by using the “suppress” feature. This can be toggled by standing in the 3rd column and pressing [p]. When set it is shown by a red line on the right. When this is feature is specified the <buffer> is not send to the SCC and the result can be used in another effect as input, using the “-” operator.
This feature is available in both the wave list as well as in the pattern. But, it should be noted that in the list editor, after an “suppress”, the next effect is also intermediately executed. Whereas in the pattern, the next event is simply executed when the next step is being 'played'.

One more thing: the amount of effects that can be 'stacked' in the list this way is limited by the list length and by the CPU!

Wave effects

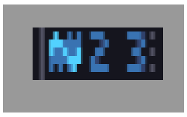
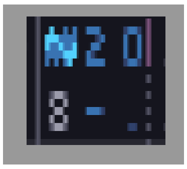
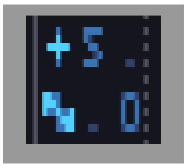
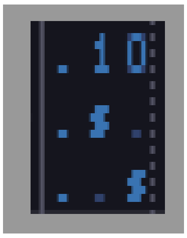
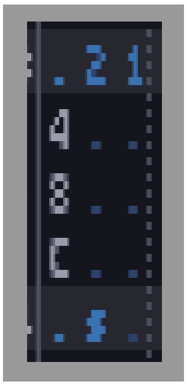
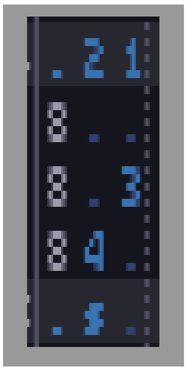
An overview of all the wave effects and input is summarized in the table below.

It is best to experiment for yourselves to get a feeling what each effect does. Also it is suggested to have a look the worked examples overleaf and the demo songs.

Key	Symbol	Description	Parameter 1	Parameter 2	Notes
x and/ or 0-F	0 F (0 to F)	Wave mix mix level value 1 to F specifies the amount of <wave 1> mixed with <wave 2>. If 0 then level is specified by <fx0>	Wave 1: 0-F / – / .. .. => wave selected from <wave 1>	Wave 2: 0-F / – / .. .. => wave selected from <wave 2>	3
g		Gain (upward / downward compression)	Factor: 0-F / .. 0 = no gain, 1-7 upward, F-8 downward (-1 to -8) .. => factor by <fx0>	Wave: 0-F / – / .. .. => wave selected from <wave 2>	3
m or 2x m		Modulate mode 1: amplitude of wave 2 ("modulator") specifies the offset in wave 1 ("carrier") mode 2: see FM modulation section	Wave 1: 0-F / .. .. => wave selected from <fx0>	Wave 2: 0-F / – / .. .. => wave selected from <wave 2>	3
j		Squeeze "squeeze" wave form towards center	Factor: 0-F / .. 0 = no squeeze	Wave: 0-F / – / .. .. => wave selected from <wave 2>	3
k		Stretch "stretch" wave form outwards	Factor: 0-F / .. 0 = no stretch	Wave: 0-F / – / .. .. => wave selected from <wave 2>	3
y		Multiplier factors the "frequency" of the wave	Factor: 0-F / .. 0 = ½ low, 8 = ½ high 1 to 7 = normal, F to 9 = mirror (-1 to -7) .. => factor by <fx0>	Wave: 0-F / – / .. .. => wave selected from <wave 2>	3
u		Bit depth (or "bit crusher" effect) Parameter "F" is used for "clear <buffer>" , could be useful on occasion.	Factor: 0-F / .. 0 to 7 = 8 bit to 1 bit 8 to B = 1 bit "threshold" C to E = do not use .. => factor by <fx0>	Wave: 0-F / – / .. .. => wave is selected from <wave 2>	3
i or 2x i		Interference mode 1:<wave 1> is rolled by <offset> and mixed with <wave 2> (50%:50%) mode 2: <buffer> is used with itself	Increment: 0-F / .. 0 to 7 = 0 to 7 F to 8 = -1 to -8 .. => increment = 1	Offset: 0-F / .. if specified (0-F) <offset> will be updated, if .. then not	1
h or 2x h		Interference with buffer mix <wave 1> is rolled by <offset> and mixed with <wave 2> and buffer as per mode: 1: 50%<wave 1>+ 25%<wave 2>+ 25%<buffer> 2: 25%<wave 1>+ 25%<wave 2>+ 50%<buffer>	Increment: 0-F / .. 0 to 7 = 0 to 7 F to 8 = -1 to -8 .. => increment = 1	Offset: 0-F / .. if specified (0-F) <offset> will be updated, if .. then not	1
s		Sync wave form link wave form updates to updates from other channel (note: ~4. on channel 5 will simulate SCC shared wave form)	Channel number: 0-5 0 = break sync 1 to 5 = set sync	(**SET** Wave: 0-F / s)	
t		Wave List run timer sets <timer>	Timer: 0 to 8 .. => <timer> = 1	(**SET** Wave: 0-F / s)	
-	-	Set wave parameters	Wave1: 0-F / – / .. <wave 1>	Wave2: 0-F / – / .. <wave 2>	
+ or ;	+	Set fx parameter	Set fx param value: 0-F sets <fx0>	Wave: 0-F / – / .. sets <wave 2>	
o		Set offset update "roll" <offset>	Increment: 0-F / .. 0 to 7 = 0 to 7 F to 8 = -1 to -8 .. => increment = 1	Offset: 0-F / .. .. => <offset> = 0	1
v		Save buffer to wave from F (parameter behavior further the same as set wave parameters)	Wave1: 0-F / – / .. <wave 1>	Wave2: 0-F / – / .. <wave 2>	
z		Stop list or jump Jump to list in wave list when Parameter 1 is set, ".." for Stop list	(**SET**) Wave: 0-F / s / .. <wave 1> (**JUMP**) List no. 0-F	**SET** Wave: 0-F / s / .. <wave 2> (**JUMP**) .. (input ignored)	2
..	.	NO EFFECT (direct wave select)	(**SET**) Wave: 0-F / s / .. <wave 1>	**SET** Wave: 0-F / s / .. <wave 2>	2

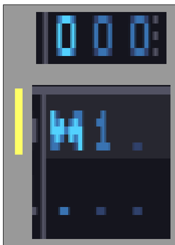
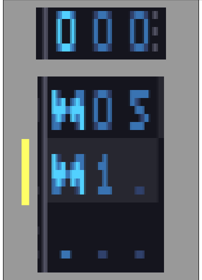
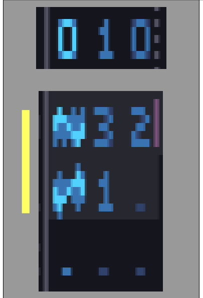
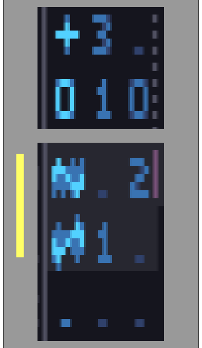
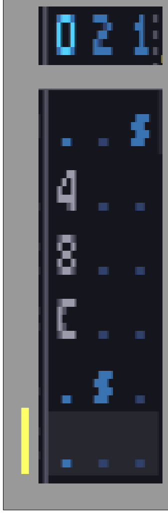
1 the increment is applied immediately, therefore the total "roll" <offset> will be offset x 2 + increment
2 "s" operator will SET the wave form from either <wave 1> or <wave 2> respectively.
3 "-" operator will select <buffer> wave form as input wave
4 ".." denotes empty field

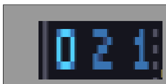
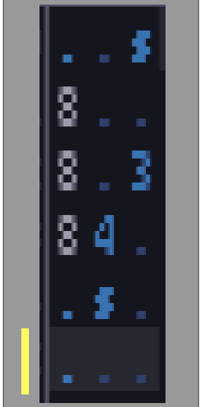
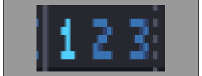
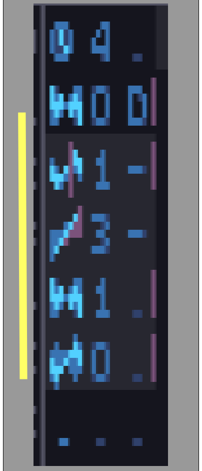
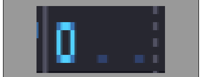
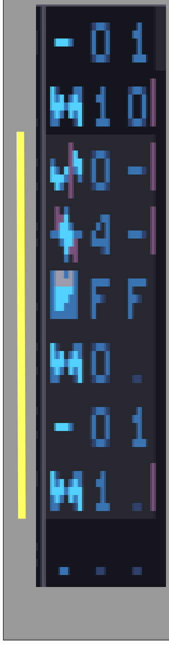
Direct wave effects examples (specified in the pattern)

	1. Multiply wave form “3” frequency by 2 and SET <fx0> = 2, <wave 2> = 3
	1. Multiply wave form “0” frequency by 2 and SUPPRESS <fx0> = 2, <wave 2> = 0 2. Mix 8/16th of <buffer> with 8/16th of wave form “0” and SET Note that <parameter 2> field is left blank, hence wave form “0” is taken from <wave 2> parameter, previously set at the multiplier effect.
	1. Set <fx0> parameter to “5” <fx0> = 5 2. Adjust bit depth of wave form “0” with <fx0> as input and SET Note that the <parameter 1> field is left blank, hence <fx0> will be used.
	1. Select “1” for <wave 1> and select “0” for <wave 2> and SET “0” <wave 1> = 1, <wave 2> = 0 2. Set wave form from <wave 1> => wave form “1” SET 3. Set wave form from <wave 2> => wave form “0” SET
	1. Select “2” for <wave 1> and select “1” for <wave 2> and SET “1” <wave 1> = 2, <wave 2> = 1 2. mix 4/16th of wave form “2” with 12/16th of wave form “1” and SET 3. mix 8/16th of wave form “2” with 8/16th of wave form “1” and SET 4. mix 12/16th of wave form “2” with 4/16th of wave form “1” and SET 5. Set wave form from <wave 1> => wave form “2” SET
	1. Select “2” for <wave 1> and select “1” for <wave 2> and SET “1” <wave 1> = 2, <wave 2> = 1 2. Mix 8/16th of wave form “2” with 8/16th of wave form “1” and SET 3. Mix 8/16th of wave form “2” with 8/16th of wave form “3” and SET <wave 1> = 2, <wave 2> = 3 4. Mix 8/16th of wave form “4” with 8/16th of wave form “3” and SET <wave 1> = 4, <wave 2> = 3 5. Set wave form from <wave 1> => wave form “4” SET

Wave list effect examples

Note: loop parts marked with yellow line for clarity

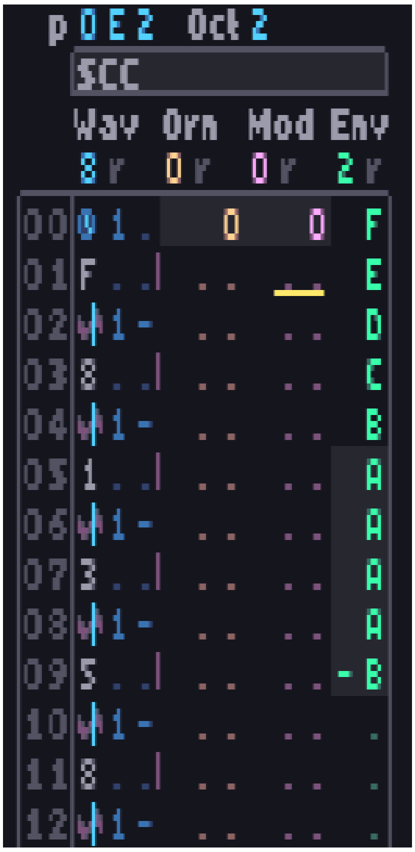
	Step: set wave list 0, select wave form “0” in both <wave 1> and <wave 2> <wave 1> = 0, <wave 2> = 0 Wave List 0: 1. Increment <offset> by “1”; Generate interference of wave form “0” with wave form “0” and SET 2. Loop to 1.
	Step: set wave list 0, select wave form “0” in both <wave 1> and <wave 2> <wave 1> = 0, <wave 2> = 0 Wave List 0: 1. Set <offset> parameter to 5 x 2 + 0 = 5 Generate interference of wave form “0” with wave form “0” and SET 2. Increment <offset> by “1”; Generate interference of wave form “0” with wave form “0” and SET 3. Loop to 2. *Note that every time the list is (re-)triggered the <offset> will be reset to 5.*
	Step: set wave list 0, select wave form “1” in <wave 1> and “0” in <wave 2> <wave 1> = 1, <wave 2> = 0, <fx0> = ? Wave List 0: 1. Multiply wave form “2” frequency by 3 and SUPPRESS <wave 1> = 1, <wave 2> = 0, <fx0> = ? ; note that internal parameters are not updated! 2. Increment <offset> by “1”; Generate interference of wave form “1” and wave form “0” and mix with <buffer> and SET 3. Loop to 1.
	Step: 1. Set <fx0> parameter to 3 <wave 1> = ?, <wave 2> = ?, <fx0> = 3 2. set wave list 0, select wave form “1” in <wave 1> and “0” in <wave 2> <wave 1> = 1, <wave 2> = 0, <fx0> = 3 Wave List 0: Note that this list is very much the same the previous example. Now, instead of a specified parameter value, the field multiplier is left blank and thus <fx0> will be used. The <fx0> parameter can be modified from the pattern, using the “set fx parameter” event, and the multiplier effect is controlled from outside the list. *Note of caution: these types of wave lists can potentially generate lots of wave data!*
	Step: set wave list 0, select wave form “2” in <wave 1> and “1” in <wave 2> <wave 1> = 2, <wave 2> = 1 Wave List 0: 1. Set wave form from <wave 2> => wave form “1” SET 2. mix 4/16th of wave form “2” with 12/16th of wave form “1” and SET 3. mix 8/16th of wave form “2” with 8/16th of wave form “1” and SET 4. mix 12/16th of wave form “2” with 4/16th of wave form “1” and SET 5. Set wave form from <wave 1> => wave form “2” SET 6. Loop to 6. (empty row)

 	Step: set wave list 0, select wave form “2” in <wave 1> and “1” in <wave 2> <wave 1> = 2, <wave 2> = 1 Wave List 0: 1. Set wave form from <wave 2> => wave form “1” SET <wave 1> = 2, <wave 2> = 1 2. Mix 8/16th of wave form “2” with 8/16th of wave form “1” and SET 3. Mix 8/16th of wave form “2” with 8/16th of wave form “3” and SET <wave 1> = 2, <wave 2> = 1; note that internal parameter is not updated! 4. Mix 8/16th of wave form “4” with 8/16th of wave form “1” and SET <wave 1> = 2, <wave 2> = 1; note that internal parameter is not updated! 5. Set wave form from <wave 1> => wave form “4” SET 6. Loop to 6. (empty row) *Note how the behavior is different compared to the similar “Direct wave effects” example*
 	Step: set wave list 1, select wave form “2” in <wave 1> and “3” in <wave 2> <wave 1> = 2, <wave 2> = 3 Wave list 1: 1. Set <timer> to 4, go to next 2. Set <offset> parameter to D x 2 + 0 = 1A; Generate interference of wave form “2” and wave form “3” and SUPPRESS 3. Modulate wave form “1” with wave form from <buffer> and SUPPRESS 4. Apply gain (factor 3) over <buffer> wave form and **SET** 5. Increment <offset> by “1”; Generate interference of wave form “2” with wave form “3” and SUPPRESS 6. No increment on <offset>; Generate interference of wave form “2” and wave form “3” and mix with <buffer> and SUPPRESS 7. Loop to 3. *Note that for a new note attack the <offset> is reset which gives the instrument an specific attack sound / accent.*
  - 0 1 1 0 0 - 4 - F F 0 . - 0 1 1	Step: set wave list 1, select wave form “2” in <wave 1> and “3” in <wave 2> <wave 1> = ?, <wave 2> = ? Wave list 0: 1. Select wave form “0” in <wave 1> and “1” in <wave 2> and go to next <wave 1> = 0, <wave 2> = 1 2. Set <offset> parameter to 0 x 2 + 0 = 1; Generate interference of wave form “0” and wave form “1” and SUPPRESS 3. Modulate wave form “0” with wave form from <buffer> and SUPPRESS 4. Apply squeeze (factor 4) over <buffer> wave form and SUPPRESS 5. Store wave form form <buffer> in to F and select wave form F in both <wave 1> and <wave 2> <wave 1> = F, <wave 2> = F 6. No increment on <offset>; Generate interference of wave form “F” and wave form “F” and **SET** 7. Select wave form “0” in <wave 1> and “1” in <wave 2> and go to next <wave 1> = 0, <wave 2> = 1 8. Increment <offset> by “1”; Generate interference of wave form “0” with wave form “1” and SUPPRESS 9. Loop to 3. *Note that in the pattern no parameters need to be specified since these are specified in the list.*

Wave list FM modulation

Zero wave
<wave 1> ="E"

Double sine
<wave 2> = "2"



```
  p 0 E 2  Oct 2
   SCC
   Way  Orn  Mod Env
   8 r  0 r  0 r  2 r
00 0 1 .      0    0  F
01 F . .|   . .  . .  E
02 w 1 -    . .  . .  D
03 8 . .|   . .  . .  C
04 w 1 -    . .  . .  B
05 1 . .|   . .  . .  A
06 w 1 -    . .  . .  A
07 3 . .|   . .  . .  A
08 w 1 -    . .  . .  A
09 5 . .|   . .  . . -B
10 w 1 -    . .  . .  .
11 8 . .|   . .  . .  .
12 w 1 -    . .  . .  .
```

Single sine

It is possible to simulate FM voices (to some extent) by using the mode 2 wave modulation effect.

With this mode the modulator wave form is used to modulate the carrier form. It is important to understand that if the modulator wave form is fully zero (i.e. all 32 byte values zero) then no modulation occurs and the output is simply the carrier wave form (just like in FM).
To simulate FM voice (and ADSR) the "total level" of the modulator and strength of the modulation can be controlled using the mix event prior using the wave modulation effect. The total level of the carrier (and voice) is simply controlled with the volume envelope.
A basic 2-op instrument wave list example is shown here on the left. In this example it is to be noted that wave form "1" is a simple sine, "2" is a double sine and "E" is a zero wave form. You can see from the wave list that the total level on the modulator starts at zero ("F") attacks to full in two steps ("8" and then "1") and then decays to 8 in 3 steps.

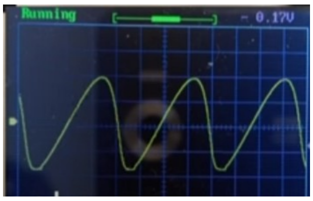
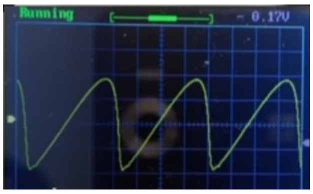
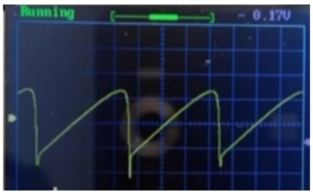
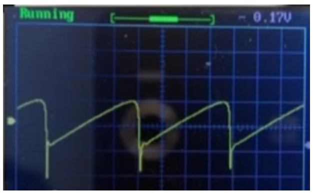
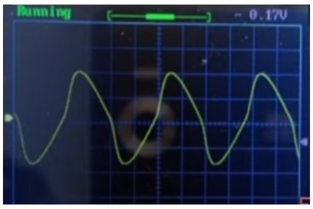
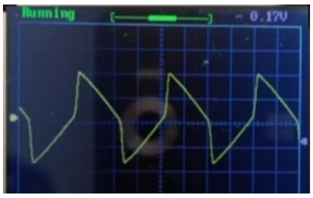
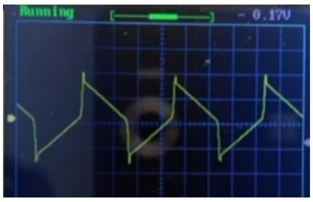
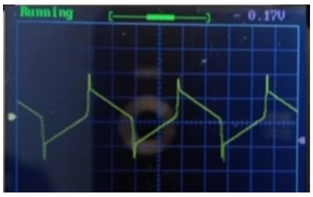
Of course you are not limited to a fixed ADSR pattern as in case of OPL sound chips.

Also use of this feature is not limited to only sine waves. It works with any wave form. In order to simulate feedback this is essentially the way to do it...

For fm like sounds the 40 lines in the wave list is not always enough. Therefore a jump to list (see 'z' in effects table) feature is added, allowing to continue in a other list. At a note re-trigger realfun jumps automatically back to the first list.

Feedback

High feedback values with total levels resulting in chaos (i.e. noise) is not possible. To simulate feed back to some degree a feed back wave form must be defined. To give an idea what a sine with feedback looks like, have a look at the following figures.

Saw tooth (or regular) feedback, increasing left to right.			
			
Square wave pulse feedback, increasing left to right.			
			

Note: double, triple etc sine wave forms for the FM synthesis can be generated with scc instrument testing / editing using the "multiplier" and "save buffer to waveform F" effects.

SCC ISR SAMPLE PLAYBACK

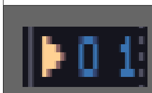
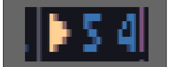
With this feature it is possible to play back up to 5 voices simultaneously on the SCC. Per song it is possible to play 16 different samples, with a total size of 128 kb. The samples are stored in a separate sample kit file (.R3W). This kit can be loaded (or saved) via the disk menu. Loading of individual samples, adapting, editing or saving will not be possible. Instructions on how to make your own sample kit are given in the next section.
The ISR sample editor (also discussed lateron) can be used to set looping position and do some small tweaking if required.

The sample playback input is given in the SCC specific part, where:

Key	Symbol	Description	Parameter 1	Parameter 2	Notes
p		Play sample note: frequency mode can be toggled with 'p' in the parameter 2 column.	Sample no.: 0-F / .. .. => defaults to sample 0	Timer: 1-8 / .. & Mode select (with 'p') .. => defaults to timer 1	-

Samples can be played as any other instrument and thus the generic part works exactly as is described in the previous section. Samples are also re-triggered with a “note on” event.

There are two frequency control modes (which can be toggled with pressing 'p' in the parameter 2 column), these are described below:

Symbol	Mode	Description
	Regular (indicated with dashed gray line)	This mode plays the sample with the pitch embedded in the sample data. The note C-4 will playback the sample at sampled frequency. For every other note the pitch is recalculated / shifted from this C-4 note to the target note.
	Direct pitch (indicated with red line)	This mode ignores the embedded sample pitch and plays the frequency of the note specified. This gives a sort of auto tune effect :) on voice samples, but is obviously intended for instrument type samples (i.e. not dissimilar to the regular wave list type scc voices).

Note that the volume (and ADSR) may be embedded in the sample data, it good to have an envelope that is max volume “F” constantly.

Unlike with (older/mod) sample tackers, changing the pitch of the sample will not change the speed of the sample. Samples cannot be sped up, however they can be slowed down using the <timer> specified in the parameter 2 column; with 1 as default speed and 8 as slowest. This is the same timer as used for the wave list and it can also be changed midway a sample using the set timer event. Using higher timer values will slow down the sample playback, but note this may not work that well for all samples.

SCC ISR SAMPLE KIT

Samples for Realfun 3 are similar to TT samples, hence the TT sample files (.sam) which come out of ARTRAG's voice encoder can be used almost directly.

As the only available load option in Realfun 3 is loading a sample kit (and not for individual samples) the samples need to bundled in a sample kit file first. This file can however quite easily be created using an assembler.

The following example is for sjasm assembler:

```
  org 0h

  byte "RF3"                                         ; Realfun 3 header

  byte ((samples_end - sample0)/10000h)*128 + 0      ; version number & size (most
                                                     ; significant bit in bit 7)
  word (samples_end - sample0)&0ffffh                ; wavekit size (lower 16 bit)

  word sample0 /2                 ; samples jump table, 16 words long always!
  word sample1 /2                 ; addresses divided by 2 !!!
  word sample2 /2                 ; fill up with 0 for unused sample sloths
  word sample3 /2
  word sample4 /2
  word sample5 /2
  word sample6 /2
  word sample7 /2
  word 0 ;sample8 /2
  word 0 ;sample9 /2
  word 0 ;samplea /2
  word 0 ;sampleb /2
  word 0 ;samplec /2
  word 0 ;sampled /2
  word 0 ;samplee /2
  word 0 ;samplef /2

  phase 0h

sample0
  incbin TTsample01.sam,6        ; skip first 6 bytes of TT header
sample1
  incbin TTsample02.sam,6
sample2
  incbin TTsample03.sam,6
sample3
  incbin TTsample04.sam,6
sample4
  incbin TTsample05.sam,6
sample5
  incbin TTsample06.sam,6
sample6
  incbin TTsample07.sam,6
sample7
  incbin TTsample08.sam,6

; ... add up to 16 samples

samples_end

  dephase

  end
```

Save the assembled file using the - .R3W - extension and you are good to go!

Maximum number of samples in a kit is 16. The maximum total sample data can be 128kb. It is possible to have a single sample of 128kb in the editor, however for for practical reasons (in case of in-game replayer) a single sample should not normal not exceed 8 or 16 kb.

ISR Sample format

The sample is simply a number of bin data followed by an end / loop marker. Each bin contains 1 word with period data followed by 32 bytes for the scc waveform. The data length of a bin is thus 34 bytes. The end / loop marker is two words (4 bytes) long.

Bin:

```
      word 001ABh               ; period        (2 bytes)
      byte 00,01,...,30,31      ; waveform data (32 bytes)
```

End markers:

```
      word 0ffffh, 0            ; end sample
      word 0bfffh, -(n*34)      ; loop sample, where n = number of bins in loop
```

With knowing this it is possible to shorten/trim samples or add a loop already manualy when assembling the wave kit, for example:

```
sample7
   ; TT sample file size: 11910  bytes
   ; sample length in bins = (11910 - 10) / 34 = 350

  incbin TTsample08.sam,6,340*34  ;trim last 10 bins of sample
  word -1,0                       ;set new end marker
```

or:

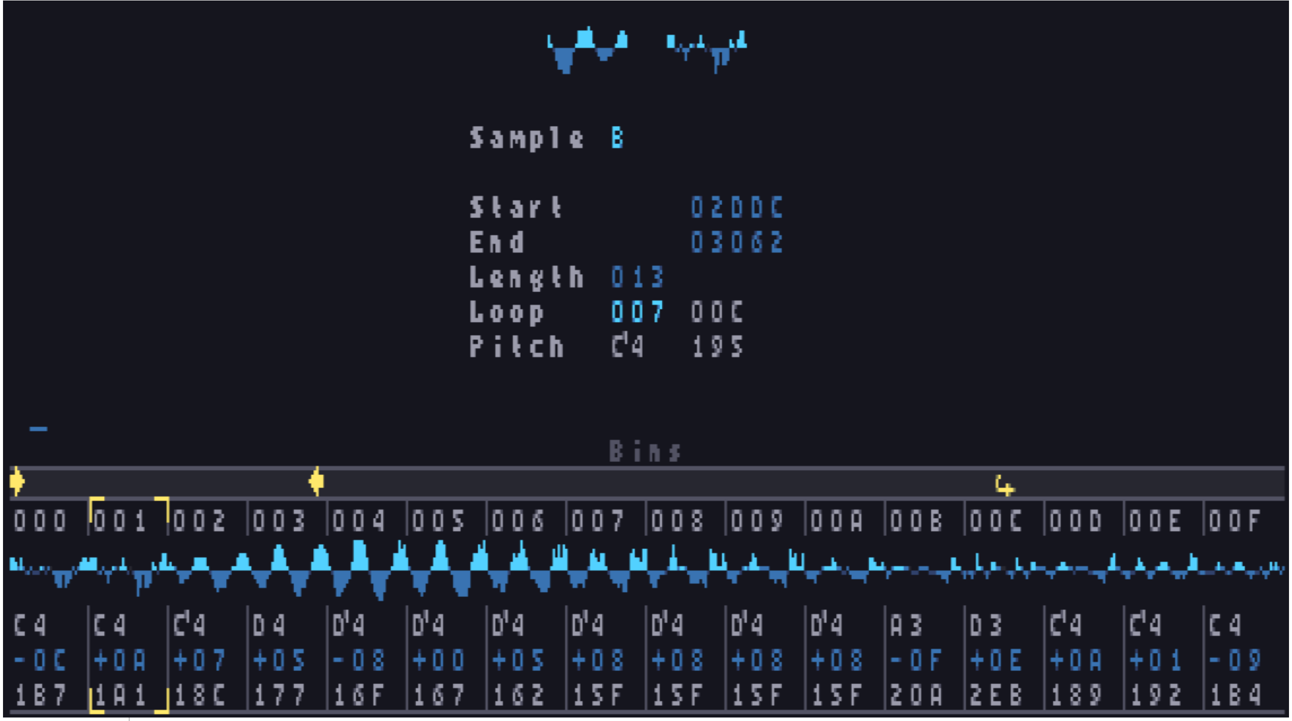
```
sample7
   ; sample file size: 11910  bytes
   ; sample length in bins = (11910 - 10) / 34 = 350

  incbin TTsample08.sam,6+(2*34),(340-2)*34   ; trim first 2 and last 10 bins of sample
  word 0bfffh,-(140*34)                       ; set loop to loop over last 140 bins
```

SCC ISR Sample editor

Channel 1 waveform

Bin cursor waveform



Sample B

Start 0200C

End 03062

Length 013

Loop 007 00C

Pitch C'4 195

Bins

000	001	002	003	004	005	006	007	008	009	00A	00B	00C	00D	00E	00F
C4	C4	C'4	D4	D'4	D'4	D'4	D'4	D'4	D'4	D'4	A3	D3	C'4	C'4	C4
-0C	+0A	+07	+05	-08	+00	+05	+08	+08	+08	+08	-0F	+0E	+0A	+01	-09
1B7	1A1	18C	177	16F	167	162	15F	15F	15F	15F	20A	2EB	189	192	1B4

Bin cursor

Start bin (block)

End bin (block)

Loop position

The sample info from top down is: Sample number, start address, end address, sample length in bins, loop number of bins in loop (blue) and loop bin number (grey) and last the average sample pitch in notes and associated period value.

The bin information from top down is: Bin number, waveform (thumb nail), bin pitch in notes, detune value and last the bin period.

Note it is only possible to access the sample editor when a wave kit is loaded. All values in the editor are hexadecimal numbers.

With [ctrl] + [b] a temporary backup of the sample kit can be made before editing. With [ctrl] + [z] this backup is restored. Note that this temporary backup will however be destroyed when using disk functions or copy functions in the pattern editor. A blue marker, shown left above the bins, indicates if a temporary wave kit back up is available.

Apart from some direct editing options available, wave forms can also be copied to the clip board and pasted in the wave editor for editing. After editing the wavefrom can be copied back from the wave editor into the sample bin similarly.

General notes to keep in mind that can improve sample quality are:

- changes between waveforms should be as graduate as possible, hence peaks and valleys of consecutive waveforms should align as good as possible (use roll to align)
- octave jumps within the waveform should be avoided (use j, k commands and a little smooth, or define transition wavforms using the wave editor)

To get an idea what can be done with the sample editor see following video:

https://youtu.be/nE8XaWoequw

Sample editor Keys overview:

F1	-	
F2	-	
F3	-	
F4	switch to SCC wave / list editor	
F5	disk menu	
esc	back to pattern editor	
ctrl + up / down	change sample number (0-F)	
left / right	step through bins (step size 1)	
left / right + shift	step through bins (step size 8)	
tab	cycle through/ go to bin in order: loop bin > last bin > first bin	
c	copy pitch to clipboard	
ret	paste pitch	
ctrl + c	copy waveform to clipboard	
ctrl + ret	paste waveform	
ctrl + l	set loop position	
ctrl + o	remove loop position	
ctrl + b	make temporary backup of the entire wavekit	
ctrl + z	restore wavekit from temporary backup buffer	
ctrl + f	change base frequency to C4, where the period under the bin cursor will be taken as the reference base that will be shifted to C4 and the rest of the sample bin periods will be transposed accordingly. (if the sample base is more than 1 octave off it may be better to change the bin octave first, i.e. bring it as close to C4 as possible, apply the command and change octave back afterwards. Example: base is A2, change it to A3 first, apply command, and lower bin octave now C4 to C3)	
q	set start bin of block	
w	set end bin of block	
3	convert 3 wavelengths in waveform into 2	
5	convert 5 wavelengths in waveform into 4	

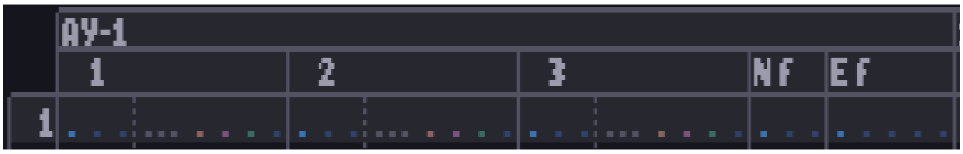
The following commands below are applied to a single bin or for the block selection when holding down [grph]:

up / down	Increase/decrease bin pitch	
up / down + shift	Increase/decrease bin pitch up to next note	
,	roll wave form left	
.	roll wave form right	
j	interleave two halves odd even of waveform and halve period	
k	separate odd even to two halves of waveform and double period	
s	soft smooth wave (smooth is useful after [j] command but do not overdo this as this will muffle the sound; smooth also reduces amplitude, note this can be boosted up again using the scale wave amplitude command.)	
ctrl + ;	scale wave amplitude up	(use grph i.s.o. ctrl for use on selection)
ctrl + /	scale wave amplitude down	(use grph i.s.o. ctrl for use on selection)

AY SUPPORT

The AY-8910 sound chip has 3 tone channels, one noise frequency channel and one hardware envelope (or amplitude modulation) frequency channel. Two AY'S are supported, AY-1 is the MegaFlashRomSCC+ AY addressed at 010h, AY-2 is the MSX build in AY addressed at 0a0h.

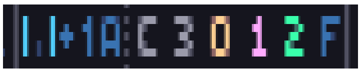
The AY-1 channels can be found after the SCC and the AY-2 after the SN sound chip, where the setup is as shown below:



The step layout is identical for both AY chips, shown below:

AY tone step:

(AY specific)
Mix list / direct mix control
Tone to Envelope frequency link ratio
Hardware envelope shape select



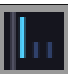
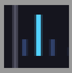
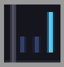
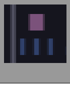
(Generic)
Note
Ornament
Modulation
Envelope
Volume

Note that the generic part is described in the general step layout section.

Mix list can be specified in the 1st column, from 0 to F. Instead of a mix list also direct mix control can be specified in the pattern. The input is described below; note that the input is in principle the same for the mix list.

Mix control

The mix between tone, noise and amplitude modulation (AM) can be directly set in the pattern, where:

Key	Symbol	Description
t		Toggle enable/disable tone
n		Toggle enable/disable noise
m		Toggle enable/disable AM
2x t,n,m		All off

Note that 8 combinations can be made.

When a direct mix event is specified an active mix list is stopped.

Link ratio

The link ratio between tone and AM (hardware envelope) frequency can be specified in the 2nd column. Also, the frequency synchronization mode can be selected.

In the first mode, which can be recognized by the blue numbers/ symbols, the tone frequency is determined first and the corresponding AM frequency is calculated from the tone frequency. In the second mode, recognized by the red numbers and symbols, the AM frequency is determined first and the corresponding tone frequency is calculated from the AM frequency.

This second mode is also know as "sync" notes from Arkos tracker (or its predecessor). It has the advantage that tone an the AM frequencies are better matched and will result in a still tone, but it false easily in high octave ranges.

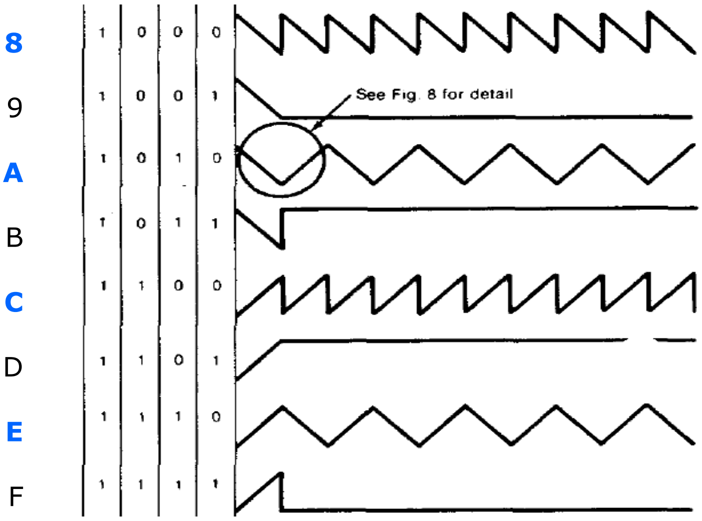
Note: with negative link ratios the tone will be often quiet to silent depending timings and phasing between the AM and tone pulse.

Link ratio and sync mode can be specified as follows:

Key	**Symbol**	**Description**
0 to 7	-3 +4 MODE 1 -3 +4 MODE 2	Set link ratio, or octave shift with +0 as center, between tone and AM modulation frequency. The value indicates how much the tone octave is shifted compared to the AM octave. (thus +2 means tone is 2 octaves higher than the AM) Negative values -1 to -3 are mapped under [7] to [5] respectively.
8	0 MODE 1 0 MODE 2	No link, AM frequency is either set in another channel or taken from the AM column (i.e. Ef)
m		Toggle sync mode when standing on the symbol.

Hardware Envelope shape select

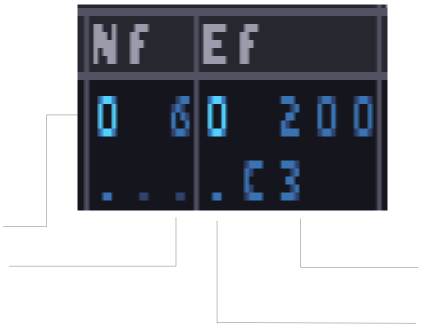
The hardware envelope shape can be specified in the 3rd column, with 8 to F. The value and the envelope shape is show here below.



It should be noted that blue hi-lighted are the most practical ones.

By default when a shape is set the AM will not be re-triggered. Re-trigger can be forced, which can be specified (toggled) by pressing the [t] key. If set a red line will be displayed on the right.

AY noise & envelope frequency

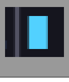


Noise freq. List
Noise frequency

Amplitude Modulation frequency
- value input – right aligned
- note input – left aligned
AM freq. List

List select

The noise and AM frequency list is specified in the first column, where:

Key	Symbol	Description
0 to F		Specify list number
z		Stop list

A noise or AM frequency list is automatically re-triggered when a new (“base”) frequency is specified unless list no re-trig is specified in the list.

Noise frequency can be specified with a hexadecimal value, from 0 to 1F.

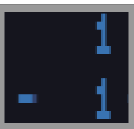
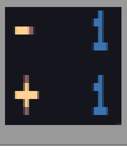
AM frequency can be specified with a hexadecimal value, from 0 to FFFF, or as a note. To insert notes the input mode can be switched with [select].

Note that this is also the same mode to hide pitch bend effects. In this mode the cursor color will change to orange instead of the default yellow. Also, it is not possible to copy notes from regular channels into the Ef channel or vice versa and block functions like transpose are not implemented either.

With the use of the Ef channel it is in principle possible to play 4 different notes at the same time of the AY... whether it sounds good is another thing ;). Perhaps it is interesting for die-hard AY chip tuners out there.

The Lists

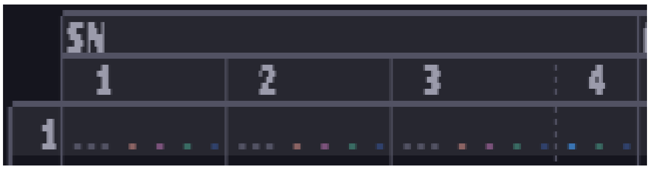
The lists for noise and AM frequency are similar. Both have two additive modes, without or with base update, where:

Key	Symbol	Description
0 to F and -		Input relative offset to base value. Toggle sign with [-] key
\ or (yen)		Toggle update base mode with [yen] or [\], result will be stored as new base. For AM frequency note that a negative value means an increase in frequency and positive value a decrease.

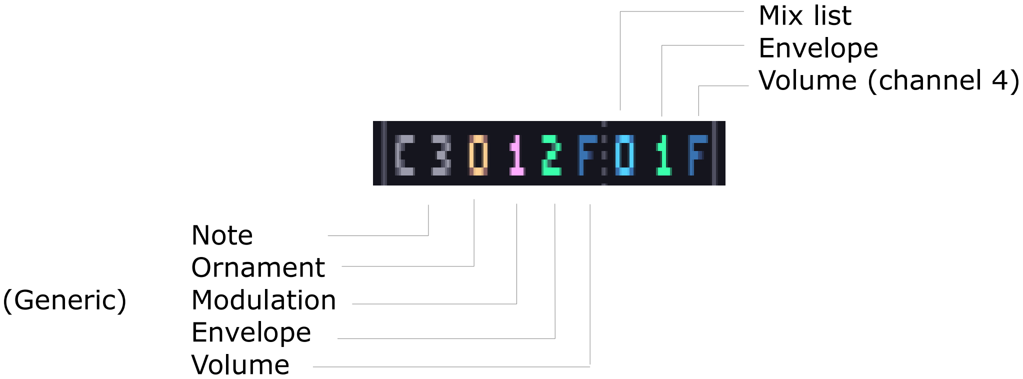
For applicable input ranges see list editor.

SN SUPPORT

The SN76489 sound chip is shear elegance in its simplicity, at least compared to the AY. The SN chip has four channels; setup as shown below:



Channels 1 and 2 only have the generic part, step layout for channel 3 & 4 is as follows:



Channels 1 and 2 are very basic and both can only generate a tone (square wave), similar to the AY, except note that the tone range is limited from A-2 to B-8.

The channel 3 & 4 duo has more capabilities, where the following combinations are featured:

1. Tone on channel 3 (same as channel 1 & 2) and noise (one out of three frequencies) on channel 4, both with independent volume
2. Tone on channel 3 (same as channel 1 & 2) and "periodic noise" tone (one out of three frequencies) on channel 4, both with independent volume
3. Periodic noise tone on channel 3 only, with tone range now from C-0 to B-8.
   Note: good for bass lines and sounds kinda like AY hardware envelope, perhaps not quite as nice, but with volume control (just go easy on volume changes, i.e. do not take too big steps too quickly)!
4. White noise on channel 3 only, with a broader range of noise frequencies, perfect for noise slides

When either option 3 or 4 of the above is selected, the volume of the sound is controlled with channel 3 volume, channel 4 volume will be ignored.
Note: technically all noise is played over channel 4, but the frequency is controlled with channel 3. For convenience the volume of channel 3 is mapped to channel 4 in order to keep the frequency and volume together in the same channel.

The setting of the two channels is controlled with the mix list, specified in channel 4. In the mix list a mix value of 0 to 7 can be specified, where:

Mix value	Description	Remark
0 =	periodic noise N/512	tone "A#4" on channel 4
1 =	periodic noise N/1024	tone "A#3" on channel 4
2 =	periodic noise N/2048	tone "A#2" on channel 4
3 =	periodic noise via tone gen # 3 output	tone "C-0 to B-8" on channel 3
4 =	white noise N/512	high frequency noise on channel 4
5 =	white noise N/1024	medium frequency noise on channel 4
6 =	white noise N/2048	low frequency noise on channel 4
7 =	white noise via tone gen # 3 output	noise "C-0 to B-8" on channel 3

Note: drum sound effects may require some action on channel 3 and 4 both.