

STarKos 1.2 Manual

- By **Targhan/Arkos**
- Released in **May 2009**

General description

STarKos is a conventional soundchip tracker, in the sense that writing musics is done the same way as existing trackers on CPC or Atari ST. The new features mainly comes from sound editing.

Some features to begin :

- 256 positions available.
- Each position is composed of 3 tracks, one par channel.
- Manages 512 tracks, each one being reusable several times in different positions OR in the same position, why not. Memory saving !
- A track can have a height from 0 to 127.
- Each track can be be transposed independently.
- The whole song can be transposed.
- Each position has its own height, from 0 to 127, independently of the height of the tracks.
- 255 instruments free to use.
- Each PSG registers can be modified while creating instruments.
- Each track/instrument is compacted in memory, this one is reorganized after every modification, so you've got plenty of memory for you !
- Frequency replay of 13, 25, 50, 100, 150 or 300 hz.
- Digidrums are supported : it is possible to enter 'SPL' keys along with a sample number, that can be read through the external player while playing the song. However, you can't load/hear a sample inside the editor.

General handling

After having typed RUN"STK", appears the STarKos main menu.

Most of the pages composing Starkos follows the same 3 parts design : a menu on top-left, some data on top-right, and the middle and bottom of the screen being taken by the song information.

You can go from a part to another by pressing **TAB**, and **ESC** to come back to the first one.

With the cursor keys, move this one, use **SPACE** or **RETURN** to select.

Every number upon which you move can be modified, either by typing a new number (ESC to cancel), or by increasing/decreasing it through left/right combined with SHIFT and/or CONTROL. Each number possesses a min/max limit, and every of your entries is corrected according to this interval.

Note that some numbers are signed, so you will have to use '-' if you want to enter a negative value by typing it. Finally, let's precise that all numbers use the hexadecimal notation, I won't insult you by explaining.

You can access most of time to every screen composing **STarKos** with these keys :

- CTRL+F1 = TRACKS LINKER (first page)
- CTRL+F2 = PATTERNS EDITOR
- CTRL+F3 = INSTRUMENTS EDITOR

- F9/F6 = Increase/Decrease octave

lexicon

Even though Starkos works as most trackers, some variations exist. A little vocabulary lesson is essential :

TRACK

The TRACKS represent the basis of your compositions. All the notes are entered there. Each track has a height from 0 to 127, and you've got 512 of them. Each track is identified by its number.

PATTERN

A PATTERN is composed of 3 tracks, each played on one channel of the PSG.

It's impossible to play a track alone, it must be included to a pattern. You can use up to 256 patterns, each having a height from 0 to 127. If the tracks included are higher, then they'll be truncated **INSIDE THIS PATTERN ONLY**, which means that the 'overflowed' data aren't erased from the memory at all, they're not used in this pattern, that's all. So it's possible to use half of a track in a pattern, and the whole in another. On the opposite, a pattern can be higher than the tracks inside. In this case, there's no problem, you'll find empty notes at the end of the tracks.

Note that it's possible to use the same track 2 or 3 times in the same pattern.

The fact that the tracks are independent of the patterns allows a lot of memory saving. I encourage you to use a track as much as you can along your song.

POSITIONS

POSITIONS represent the way PATTERNS are played.

Consider position like the link of a chain. To each link corresponds a single pattern. When you start playing your song, the position 0 is played. When the pattern to which it's linked is over, then the POSITION 1 is played, and so on... 256 positions are allowed, but your songs will be really shorter than that usually. The Position system allows the musician to play the same pattern several times without having to copy it.

INSTRUMENTS

You know or guess what INSTRUMENTS are, so I won't dwell on that. You have 255 instruments, from 1 to 256.

SPECIAL TRACKS

SPECIAL TRACKS are new. They allow to set digidrums, and speed changes. Each pattern has a Special Track, on top of the 3 normal tracks. You have 256 special tracks available.

The various screens

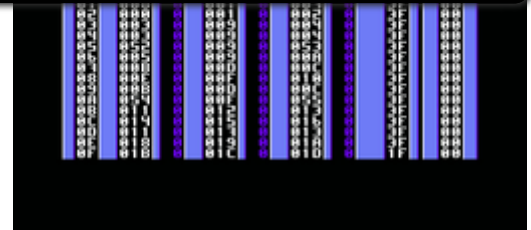
Once **STarKos** is loaded, the first screen to appear is:

The tracks linker

Here are the explanations of the different columns you can see:

NOPOS

Position number. The position 0 is the first one played of your song.



TRACK1/2/3

Tells which track is played in this position, and on which sound channel.

TRANSP1/2/3

Tells the level of transposition attributed to the track of each channel. In most case, it equals 0.

HEIGHT

Height of the pattern, from 0 to &7f, thus a height of 1 to 128 lines.

SPECIAL TRACK

Number of the special track used for this position.

As you probably noticed, some NOPOS are white, others blue. Know that only white POS are going to be played, because only them are included to the interval of the played positions, from 0 to ENDPOS. To modify this value, 2 ways : Press TAB till you're in the upper right part and modify the value.

You can also use the keyboard shortcuts in the middle part :

- RETURN to define ENDPOS
- SPACE to define LOOPTO, according to the cursor position, obviously.

Note that each move refreshes NOWPOS. Moreover, know that LOOPTO is always equal or inferior to ENDPOS, which is logical.

But let's go back to the upper left menu. We'll see later about 'DISC' and 'MISC', but by selecting 'SONG INFO' (or by pressing TAB twice), you go to the upper-right menu. Here's the description of the various columns :

SONG POS

It's the current position. It's the pattern linked to this position that's going to be edited (when going to the pattern editor). It's also from here that will be played the song if you select 'PLAY SONG'.

END POS

Last position used in the song. Defines the end of the song, in other words.

LOOPTO

When the music is over, it loops to the position designed by LOOPTO. Generally, LOOPTO=0 to make the song restart from the beginning.

INSTR

Speed of the song when played from the beginnning through PLAY SONG.

REP FRQ

Replay Frequency. Various possibilities : 13, 25, 50, 100, 150, 300hz. Enough to entertain you for centuries. Beware, highest frequencies considerably slow down the software. 50Hz is generally used. Warning! Modifying the REP FRQ also changes the speed of the song! The player was first intended to adjust the speed accordingly, but some people argued against that, because it wouldn't have allowed unusual speeds.

OCTAVE

Used when playing a sound, or entering a note in a track.

FREEMEM

Free memory. STarkOs manages memory dynamically and optimizes data as much as possible. When this value equals 0, then the screen becomes red and the last action is cancelled. It's greatly advisable NEVER to save your song if this happen.

Disc operations

This menu allows you to load/save your song/instruments.

All files generated with STarkOs are saved in ASCII format. Disc accesses are therefore slower, but it's a necessity, as the data aren't written linearly in the memory. It brings a second problem : the AMSDOS errors managment of such file is a bit limited ; it doesn't allow an accurate detection of what error can happen while loading/saving. However, when an error occurs, it's possible to RETRY or CANCEL.

STARKOS filenames have reserved extension :

- SKS for SONGS.
- SKI for INSTRUMENTS.

Here's what you can find in the DISC menu :

GETDIR

Read the disc directory and gives back the remaining kilobytes, as well as the numbers of SONGS/INSTRS on the disc. Loading a file is disabled if there's nothing to load, which is logical.

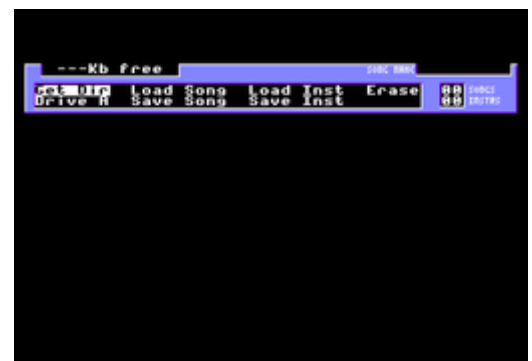
DRIVE A/B

Allows you to switch between your drives, if you're lucky enough to have two of course.

LOAD SONG

Load a song, chosen in a little list. WARNING ! When the loading begins, the previous song in memory is erased !

SAVE SONG



Load an instrument. Once in memory, you must choose where to put it among the 255 slots.

Note that a badly loaded instrument is automatically erased in memory.

SAVE INST

Save a selected instrument. The filename is the instrument name.

ERASE

Erase a file to select in a list. White filenames are the ones created with STarkOs, for you not to erase a wrong file.

Press ESC to quit this menu.

A little remark : a loaded music is already completely optimized, so it's useless to optimize it once loaded.

Miscellaneous

This menu has various options, about the software and your song.

OPTIMIZE

Optimize the song. If you feel a bit cramped, try this. Warning, as all the data is crushed, the software might slow down a bit.

OPT LEV

Optimization Level.

- **MAX** : as soon as a pattern/instrument is created or modified, only the needed room is allocated. This state saves every possible byte of memory, but slows down the software, proportionally to the song size.
- **NORMAL** : When a pattern/instrument becomes smaller, the memory is not reorganised. This mode should be enough for 99.99% of the musics.

CLEAR

Clear the Patterns, or the Instruments, or both ! WARNING, this can't be undone, so be careful !

TRANSP

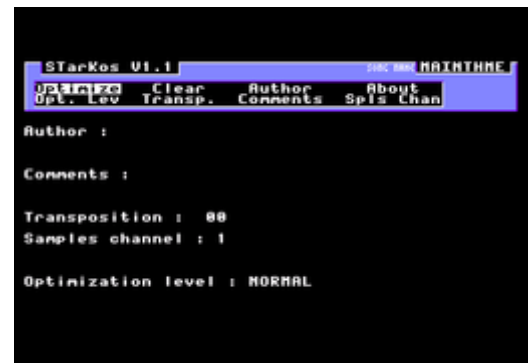
Defines the transposition level of the WHOLE song. EVERYTHING is transposed, but your tracks aren't modified. Even the notes you enter with the keyboard are transposed ! As this 'effect' doesn't modify your song in any way finally, don't hesitate to use it as many times as you want to choose the right reference note.

AUTHOR

Allows to sign your music. Saved in the SKS file, but not in the generated file !

COMMENTS

Allow you to write a little comment of 32 characters. Same remark as above.



SPLS CHAN

Defines the channel where the samples entered in the Special Tracks are going to be played. This option is only esthetical for the moment, as no sample can be heard under the editor. However, this value can be retrieved with the external player : so it's possible to play a tune with samples according to its author's will.

HIGHLIGHT

New since 1.2, allows to highlight every X lines, where X goes from 3 to 8. 0 disable the Highlighting.

COLORS

Allows to change the palette.

Press ESC to get out of this menu.

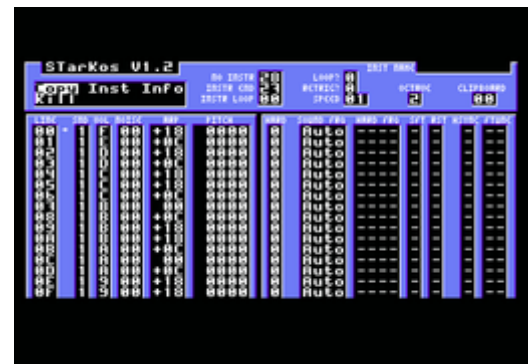
Instruments editor

Here's the first really interesting part of this notice.

You're here to create your instruments. The upper-left menu allows you to copy or destroy any instrument.

A STarkOs instrument is built on a values list, of 256 lines max strong. These values are detailed just below.

Know that you've 2 possiblity to create your sounds. The first one, the most complex but also the most powerful, consists in using the list of lines. The second one is graphical, like in the BSC's sountrakker. You can flip from one to the the other by pressing ENTER.



But for now, let's stick to the first one. In the upper right menu, you can find this :

NO INSTR

Number of the current instrument, the one that's being edited.

INSTR END

Instrument height. Corresponds in fact to the last line used.

INSTR LOOP

Number of the line where the instrument is going to loop when having reached its end. Obviously **INSTR LOOP** must be inferior or equal to **INSTR END**.

LOOP?

Boolean telling whether the sound must loop or not. **INSTR LOOP** is useless if **LOOP?**=0. A sound that doesn't loop doesn't make any sound at the end.

RETRIG?

Defines the time (a multiple of the Replay Frequency) between playing two lines of the instrument. '0' means that a line is played every 50Hz (if your Replay Frequency is 50), 1 = each 100Hz... Very useful to play slow and long sounds without having to copy the lines several times. Save both time and memory.

CLIPBOARD

You can't access it directly. The value indicates where is the pointer inside the clipboard. It allows to copy lines of instruments in memory, to paste them elsewhere later.

By pressing **TAB**, you go inside the instrument edition itself. The cursor management works exactly as in the PATTERNS LINKER.

- **RETURN/SPACE** to define END/LOOP.
- **Control+SPACE** to switch LOOP?. Note that when defining LOOP, LOOP? is automatically set to 1.

Let's have a look at these columns :

SND

Tell to the PSG to create a sound wave if set to 1. Set to 0, you can't create any melody with conventional sounds. But you can still create HARD envelope and add NOISE.

VOL

Gives the volume, from 0 to &F. The volume is cancelled if a HARD sound is triggered. Likewise, a non-HARD sound with a volume to 0 cancels all the line, as nothing can be heard.

NOISE

White noise value, from 0 to &1F. Mainly used for drums, percussions and explosions. Not linked to SND, but needs a volume greater than 0, or a HARD sound.

ARP

Arpeggio. A signed value (from -#5F to #F added to the basic noise (given by your tracks, or your keyboard). Type 0,3,7 on 3 lines to get a minor chord.

PITCH

Value added to the final frequency. Usually used to create pitch bends or vibratos. Value between -#FFF and #FFF

HARD

Tells if a HARD envelope is used. 0 if no, 1,2,3 or 4 if yes. 1 corresponds to a sawtooth curve, 3 to its symmetry. 2 to a 'mountain' curve, 4 to its symmetry. Don't forget that the PSG can only manage one hard envelope, so giving a hard sound to more than one track at the same time generally results in harsh sounds (but some of you may find something interesting out of that for sure).

SOUND FRQ

Sound Frequency. In AUTO mode (value =0), the sound frequency depends on the key played by your track or your keyboard. By giving a value (from 1 to #FFF), you force the frequency on your own initiative. This can be used to create special effects or to add effect/drums before the rest of the sound.

SHIFT becomes useless and is cancelled.

SHIFT

This one is interesting. Give the report between HARD FREQ and SOUND FREQ (in Auto mode or not). Follow this nice equation : $\text{HARD FRQ} = \text{SND FRQ} / (2 \text{ to the power of 'SHIFT'})$. Value from 0 to 7. Generally used : 4 or 5 for cool basses.

INITWAVE

Allows to force the Register 13 to 1. When this happens, the HARD envelope is restarted from the beginning. Warning, setting a INITWAVE=1 in a loop produces a very disgracious sound.

HARDSYNC

An idea from ST Trackers. Allows to perfectly synchronize a normal envelope with a HARD envelope (if HARD set to 1, of course). Indeed, when the previous formula is used, appears an approximation between the two curves, which brings a 'mwwaaaiiii' sound so distinctive of hard sounds :).

When HARDSYNC is set to 1, the player corrects the sound frequency to get an EXACT multiple of the HARD freq. No more desync ! The problem is, the approximation applies this time to BOTH curves, which produces some off-key notes. This effect is really only useful with low sounds, highest being completely wrong so much the frequency are truncated.

FINETUNE

When HARDSYNC is activated, so is FINETUNE. Allows to generate a shift between the SOUND FREQ and the HARD FREQ. Now you can precisely master your 'mwwaiiii', how lucky you are :). Value between &00 and &FF.

As you can notice, the 'NO POS' column has white (only one when the instrument is empty) and blue values. Only the white ones are included to the interval of the sound to play. To enlarge this interval, increase the END parameter.

Insert/Delete functions exist, through **F0** and **'.'**. They apply on an entire line. But it's also possible to insert/delete a value in a single column only, but pressing CONTROL with F0 and **'.'**.

Be careful, this function doesn't take all the instrument in account, but only the interval defined by your cursor and the END of the instrument, contrary to the 'raw' insertion that shifts everything from the cursor.

As soon as you leave an instrument (by changing to another page or instrument), this one is automatically optimized in memory ! This means that some values will be modified all by themselves, and that the lines not included in the playing interval will be destroyed.

Alternate instruments editor

As I said above, there's a second sound editor, graphical but less powerful. It will however be useful to quickly create simple sounds composed of volume and noise curves.

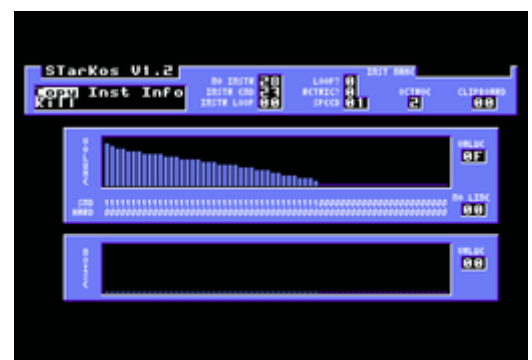
Press **ENTER** to switch between the 2 editors.

Oh, how beautiful it is ! By pressing **TAB**, you switch between the 2 graphics.

The first one concerns the volume, the second the noise.

Some explanation :

The columns can be light or dark blue, along with a halftone (or not) :



on the sound), whether HARD is set to 1, which cancel all volume anyway. To clear this halftone, check these two flags with the 'line' editor (another way, very simple, see below).

- No halftone : No probleme, there's no HARD flag, so the column is going to be played, unless it's dark blue.

For simplicity's sake, as soon as you modify a column, SND is automatically set to 1 (only if HARD=0), so you don't need the 'line' editor to create simple sounds.

Halftone/dark blue columns will be eliminated when optimizing, because they're of no consequence in the sound !

Use keyboard shortcuts to set END, LOOPTO and LOOP? without having to go to the upper-right menu.

To finish, note that it's possible to insert columns with F0 and '.', but that corresponds to a column-insertion in the line editor ! Thus, only the volume/noise curves are shifted. As before, only the interval defined by your cursor and END is shifted.

Now you've got a wonderful sound, let's learn how to play it !

Playing a sound

It is possible to play the current sound in each top-left menu of the main pages. To do this, you just have to use your keyboard, exactly like in the BSC's Sountrakker.

```
" # $ % & '
Q W E R T Y ... --> Current octave +1

S D F G H J
Z X C V B N ... --> Current octave
```

It's also possible to play the current sound in the lower part of the instrument editor, by using the keyboard while pressing CONTROL. Warning, changes made on the sound data while playing it DOESN'T apply on what it heard ! You have to retrigger it to hear the changes.

Now you're ready to fight among the Follins brothers and Madmax, let's see how to create patterns...

Patterns editor

The upper-left has nothing outstanding :

- **PLAYPATT** to play the current pattern.
- **PLAYSONG** to play the song from the current position. Note that if you play your song from the position 0, SPEED takes the value of BEG SPD, which is logical.

By pressing **TAB** you go to the lower part, the pattern edition itself. You can find the 3 tracks, and the special track, smaller. Here's the description of the column composing each normal track :

KEY

key and octave to play. Example : C-1, D#4.

INSTR

Instrument to play, from 1 to #ff. Always go with the KEY.



PITCH

Allows to make portamentos up/down. The value is signed, from -#7F to #7F.

- **INSTR** and **KEY** are linked : you can't have one without the other.
- **VOLUME** is always associated with a **KEY**, however you can put it where's there's no note.
- **PITCH** is completely independent, you can put it where you want, KEY or not.

You can place a '**RST**', a reset, by pressing **CLR**. This **RST** cuts the sound of the canal. Warning, what remains on the line will be erased when saving the track in memory.

You can also put your digidrums in 'normal' tracks by pressing |. The number of the sample is defined by the current instrument, from 1 to &1f. Warning, the rest of the line will be ignored.

The special track allows you to set Speed change, and to put Digidrums (for the moment, these one aren't played). Just put a S for Speed or D for Digidrum, followed by the value you want.

Note that a little raster in at the bottom of the screen shows you when a digidrum starts IN THIS FRAME. So if you start more than one in a frame (by using a faster than 50hz song), you will still see one raster only !

So, you'll ask, why giving two possibility to set digidrums ? Well, because the behaviour is different according to where you put them. A digidrum put in normal tracks cuts the sound of that channel, as a normal sound would. But a digidrum set in a special track will cut the sound of a channel (designed by SPC CHAN in MISC menu) only during the exact duration of the sample. At its end, the sound of the channel is recovered.

Other cool stuff, two little switches :

- **CYCLE** allows you to go from an extremity of the pattern to the other when going beyond. The '\ ' keys switch the switch, and a little logo shows if it's on or off.
- **BEYOND**, switched by SHIFT + '\ ', allows to see what's 'beyond' the limit of the pattern. Know that if your pattern is #7f height, and that you bring it back to #1f, what's now hidden is still in memory, and can be seen by activating the BEYOND switch. When on, the NO LINE becomes white, indicating that you're still in limit of the pattern as you defined it. If they're blue, then you're beyond the limit, and can reach the physical limit of the pattern, #7F.

CLIPBOARDS

Let's talk about the clipboards. You've got 2 clipboards only for the pattern edition. They work exactly like the Soundtrækker's, with one option added.

The first clipboard is used to save a whole track. Note that if you copy ('copy' in the copy/cut/paste sense) a TRACK, you won't be able to paste it into a SPECIAL TRACK, and the opposite.

- **SHIFT+1** : Copy a track in the clipboard
- **SHIFT+2** : Cut a track in the clipboard
- **SHIFT+3** : Paste a track

Something new : if you use CONTROL instead of SHIFT while pasting, you'll paste notes only, and not emptiness ! So it's like a OR paste mode. Very useful to mix two tracks into one.

The second clipboard allows you to take notes from a track, one by one, to stack them up in memory and paste them one by one where you want.

- **SHIFT+7** : Copy a note in memory
- **SHIFT+8** : Paste a note in a track
- **SHIFT+0** : Comes back at the beginning of the clipboard

What's more, coming back at the beginning of the clipboard sets a limit at the last item put in the clipboard recently. This limit can be redefined by copying others notes inside the clipboard.

Ok, I think we've talked about everything. Wasn't it great ?

Keyboard shortcuts

General

TAB	CONTROL+F3
Go from one part of the screen to another	Goto INSTRUMENTS EDITOR
ESC	F7/F4
Come back to the upper left menu	Increase/Decrease current position
SPACE/RETURN/COPY	F8/F5
Select	Increase/Decrease current instrument
SHIFT and/or CONTROL + left/right	F9/F6
Increase/decrease value	Increase/Decrease octave
CONTROL+F1	1/2/3
Goto TRACKS LINKER	Cut/Activate channel 1/2/3 (only in Patterns Editor)
CONTROL+F2	SHIFT/CONTROL + Select PLAYSONG
Goto PATTERNS EDITOR	Play the song from the beginning.

Patterns linker

SPACE	F0/.
Define LOOP TO	Insert (copy current line)/Delete a line
CONTROL + SPACE	CLR
Switch LOOP?	Switch between STEP 0/1
RETURN	[
Define ENDPOS	Switch STEP 0/1
U	Q
Find a new empty and unused track	Invert the sign of a transposition

SPACE

Define LOOP TO

CONTROL + SPACE

Switch LOOP?

RETURN

Define ENDPOS

ENTER

Goto GRAPHIC INSTRUMENT EDITOR

]

Switch STEP 0/1

A

Invert value sign

^

Fill the rest of the column with what's above cursor
(USEFUL !)**DEL**

Set the current value to 0

DEL + SHIFT/CONTROL

Clear line

COPY

Copy the current value to the next line (USEFUL !)

F0/.

Insert/Delete a line, new line = previous line

SHIFT + F0/.

Insert/Delete a line, new line filled with 0

CONTROL + F0/.Insert/Delete a column (interval = cursor-
>ENDLINE)**SHIFT + 7**Copy current line into clipboard (and inc clipboard
position)**SHIFT + 8**Paste clipboard to track (and decrement
clipboard position)**SHIFT + 0**

Goto clipboard position 0

ESC

Stop sounds

KeyboardPlay a sound (Press CONTROL if you're in middle
part)**ESC + SHIFT**

Stop sounds if cursor is in the middle part

Alternate/Graphic instrument editor

SPACE

Define LOOP TO

CONTROL + SPACE

Switch LOOP?

RETURN**DEL**

Set the column value to 0

SHIFT+Up/Down

Inc/dec all the interval

F0/.

Copy current column to the next one

Pattern editor

\	SHIFT + 1
Switch Cycle flag	Copy track
\ + SHIFT	SHIFT + 2
Switch Beyond Flag	Cut track
CLR	SHIFT + 3
Reset	Paste track
^	CONTROL + 3
Sample	Paste track in OR mode
CONTROL+I	SHIFT + 7
Invert pitch sign	Copy current line into clipboard (and inc clipboard position)
] (+SHIFT)	SHIFT + 8
Increase (Decrease) STEP	Paste clipboard to track (and decrement clipboard position)
ENTER	SHIFT + 0
Set the current instrument to the instrument pointed by cursor	Goto clipboard position 0
SHIFT + Q/W/E/R...	Copy
Goto line 0,&10,&20,&30...	Read and play line
SHIFT + F7	SHIFT + ESC
Increase TRACK of current channel	Cut sounds
SHIFT + F4	F0/.
Decrease TRACK of current channel	Insert/Delete line
DEL	
Clear note+instr+vol OR vol alone OR pitch alone	
DEL + SHIFT	
Clear note+instr+vol+pitch	

Generating a STarKos player

Once you have finished a song, perhaps you want to use it in your own productions. After having converted a SKS file into a BINary file (using the Song Generator, look at GENSONG.ENG), it is now possible to play the song with one of the players available.

Run the file called "GP", or if you connected the right ROM, type |GP.

The Player Generator Main Menu

After the program is loaded, a menu appears where you can select which type of player you want to generate.

The differents types of player

STarKos gives you the choice between 3 players. Why three ? Well, it all depends on what you want to do :

ASM Player

If you intend to use your song in a demo, without the system, then this one is for you. This is the most optimised version of the StarKos players.

Demomakers/Fanzmakers will be only interested by this one. The ASM Player doesn't save any registers, modifies all of them (including the auxiliary, but not SP), so the system won't like it (strange behaviour/crash).

BASIC Player

This one is made if you want to use your song in a Basic program, or an Asm code that still uses System. The only difference between the BASIC and the ASM Player is that the BASIC Player saves some registers so that Basic doesn't crash after playing the song.

INTERRUPTION Player

This one is for people who want to put a song and stop thinking about calling it. Basic users are the main users of this player, but it can also fit to any ASM program that doesn't kill the system and the interruptions. A simple CALL and the song begins !

Please note, if you use ASM or BASIC players, that it's up to YOU to call the song at the right moments (once per VBL, twice... up to six times for 300hz songs). I guess it's possible to get a 300hz song in BASIC, but it's somewhat useless ; use INTERRUPTION Player instead). INTERRUPTION Player automatically detects the song frequency. In every case, a 300hz interruption is cast (using system vectors), but only a few of interruptions are used if using a less than 300hz song.

Generating a player

Once you've chosen the player you need, you will have to give the name of the saved player (STKPLBAS/STKPLASM/STKPLINT for example), then the address in memory where you want it to be located. The player is then saved.



First, load the player at the address you gave in GENPLAY. In basic, this could be :

```
Memory &1fff:load"PLAYER",&2000
```

Then, load the song, where you compiled it in GENSONG.

```
load"SONG",&3000
```

ALWAYS PROVIDE THE ADDRESSES WHEN LOADING A SONG OR A PLAYER. This is needed because the saved start address (in the AmsDOS header) aren't (and can't be) right. However, execution address is good.

To start playing the song, it varies according to the type of player used:

BASIC Player

Three CALLs are available :

- CALL PLAYER,ADSONG : Initialize the song.
- CALL PLAYER+3 : Play song.
- CALL PLAYER+6 : Stop song.

Initializing the song **MUST** be done once before playing the song for the first time. **DON'T FORGET** the Address of your song (ADSONG) ! It's needed for the player to know where the song is. In our example, you would have to write :

- CALL &2000,&3000 to initialize the song.
- CALL &2003 to play the song
- CALL &2006 to stop the song.

You can also re-initialize the song if you want it to restart. Note that the BASIC player can also be called in any ASM program, it's just slightly slower than the ASM player. To give ADSONG to the player when initializing, just set DE with this value (like below).

ASM player

Well, it works nearly the same way.

- CALL ADPLAYER to initialize the song, with DE pointing on the song.
- CALL ADPLAYER+3 to play the song
- CALL ADPLAYER+6 to initialize the song.

In our example, this would do :

- LD DE,&3000:CALL &2000 to initialize
- CALL &2003 to play
- CALL &2006 to stop.

Be sure that you don't need the system, or it's going to crash. All registers are corrupt, too.

INTERRUPTION player

It's even more simple :

- CALL PLAYER,ADSONG : Initialize and start the song.
- CALL PLAYER+3 : Stop the song.

PLAYERS.ENG.

Compiling an SKS song

Once you've finished a song under STarkOs, perhaps you want to use it in your own productions. This is a very simple procedure. Just run the GS (GENSONG) file (or |GS if you've connect the right ROM) and follow the instruction below.

This program will convert a SKS file into one to eight BINARY file(s). These files can then be used, along with the STarkOs player (see PLAYERS.ENG, and the GP prog), in any of your program, whether it's coded in Basic or Assembler.

So, you've probably understood that it's most the same as the BSC's Soundtrakker, except that the song you generate is NOT in the same file as the player. This has no convenience, and only advantages : you can use ONE player for a multitude of songs ! And you can put them everywhere in memory (well, there're some restrictions if you code in basic). Don't forget the 2 modules that allow you to relocate both song and player... That's what makes STarkOs the most flexible music editor available.

Why a compiler?

Because !! I think it's a terrible mistake to prevent the compiling of a song, like the AMC. Why ? Because most of the times, a song contains unused instruments, patterns, and the format used under an editor follows most of the time more 'opened' rules. The format used by the Starkos player is more optimized. Every unused data is cut, as well as data that will never be reached, at the end of patterns for example.

The MUSIC GENERATOR main menu

Once GS is loaded, a menu appears :

- Show all files on disc
- Show SKS files on disc

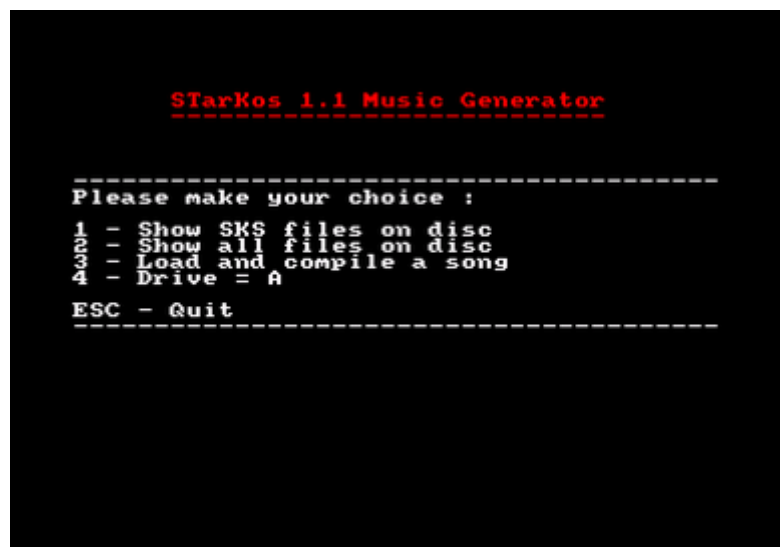
These two options are pretty self explanatory, so I won't tell you what they were made for..

- Load and compile a song

This option is of course the main one. See below.

- Drive A

Allows you to switch between Drive A and B.



Load and compile an SKS song

First of all, is asked the name of the SKS file. Then you'll have to enter the address of the song, from 0 to #ffff. After that, the screen turns to black and the CPC's going to work, the longer the song is the longer you'll have to wait, but hey, who cares.

For technical reason, the screen is turned off because it's used as a buffer. If an error disc happens, the screen turns to red and you will have to press 'R' to Retry or 'C' to Cancel.

files of #8800 bytes max (34kb) only, the rest of the memory being taken by the compiler, the buffers and the system. If your song is bigger than that, then a second file is created.

The screen shows 2 options : Automatic or Custom size. What does it mean ?

Well, it's very simple. **Automatic** means that the files are as big as they can be. If you song size is #A000, then the first file size will be #8800, and the second #1800 (#8800+#1800=#A000). Most users will use the Automatic size.

Custom allows you to define the size of each file. You can save up to 8 files, with a size from 1 to #8800 (34kb). What is the use of this ? Well, it can be useful if the song is big and you need to load it in several parts. You can want to first have a multitude of files, to be linked later before playing it.

This brings me to an important point :

Even though you can create several files, THE FILES MUST BE LINKED ONE AFTER THE OTHER ! So it's not possible for the moment to cut the songs in various parts in the memory. The song data must be put in a LINEAR way. However, you can open a bank and put a part of the song inside, even if the other part get over it. Example : A song could start from #6000 to #7fff of bank #c4, and continue from #8000 to #a000 in the 'normal' memory. Just remember that the player must have access to any data of the song, so don't forget opening the bank if needed.

Once the choice is done, Automatic or Custom (and the sizes given), a SECOND pass begins. The compiler creates indeed the song in 2 pass, like most assembler in fact. Just wait the same time as previously ! However, this time the files are saved on disk. When it's done, a message indicated you whether the song has been successfully saved or not.

It's over !

The files created have the same name as the SKS song, but the extension is different. For a song called TRY.SKS, the output files will be:

```
TRY.BIN
TRY.BI1
TRY.BI2
...
TRY.BI7
```

The files generated don't have the right START address if you look at their header. So when you load them, you MUST precise their address. LOAD"TRY.BIN" will NOT work. But LOAD"TRY.BIN",&4000 will work.

That's all. Now that you have a song compiled, just use GP, that generates a player. Please read GENPLAY.ENG for more info !

The End

That's it!

STarKos features a lot of keyboard shortcuts. You should print the KEYS file on the disc (or keep your browser open [here](#)) to learn them easily. It's much more convenient to quickly access/do stuff with a shortcut than navigating through all the parameters/windows.

I hope this tool will please you because it wasn't easy to make, even if it looks so when over..

Bugs report, remarks, suggestions, here we go :

- targhan@cpcscene.com
- julien.nevo@netcourrier.com

You can write me here:

Last remarks

Here're some points I want to talk about briefly :

- Contrary to what most people think, it is NOT possible to convert ALL the ST musics to CPC, even if they use conventional sounds. The PSG in the ST is clocked at 2MHz, against 1 for the CPC. In order get the same sound frequency, you must divide all periods by 2. Therefore some approximation appears, i.e. a desynchronisation between the tone and hardware soundwaves. This is what happens in the MEGADIST song included in this kit. As Madram suggested, you could correct that by doubling the sound frequency (50hz -> 100hz) and 'surrounding' the desired values with the 2 nearest values, and switching between them. Good luck...

About the tool in itself :

- **STarkKos** is slower than most trackers, that's because the organisation is far more complex. I allow 255 instruments, 512 tracks, 256 patterns and special tracks... It's more than any trackers on CPC or ST. If I had managed my memory like BSC and PRODATRON did, I would never had enough memory. Everything is allocated dynamically. It's quite complex to manage and takes a bit of machine time.
- The pattern display is slow. That's not because the code is not optimized, but because I used the routines I already had i.e. they're quite powerful, honestly fast, but that's not enough for such display. I would need specialized routines for that but...
- But I can't afford that for the moment : I don't have enough memory ! **STarkKos** is 24000 lines long, for 30KB max of compiled code !
- And last but not least : I MADE STARKOS FOR THE CPC SCENE, SO TELL ME WHAT YOU THINK ABOUT IT !!!

Ok, that's all, folks ! See you soon !

Targhan/Arkos

documentations/software/starkos/manual.en.txt · Last modified: 2009/08/10 13:17 by grim