JR TRACKER

F256 JR

https://github.com/econtrerasd/...

Table of content

REQUIREMENTS

JR Tracker requires the following to operate:

1. Foenix F256 Jr or Foenix F256k
2. Keyboard to operate.
3. SD card or IEC Drive if you want to save your creations.

You can get the required files to use the Tile Map Editor from:

<< Pending GitHub Link >>

You only require one file to execute the program:

- JRTRACKER.BAS

The complete program is written in BASIC (SuperBASIC)

DESCRIPTION

JR Tracker is a simple Tracker that drives the SN76489 chips (referred as PSG from now on for simplicity) in the Foenix F256 series of Computers created by Stefany Allaire, check her website for more information.

https://c256foenix.com/

A pair of PSGs are implemented within the FPGA since they are very simple sound generators, easy to understand and easy for the CPU to drive, requiring very low supervision from the CPU, thus making them ideal for a simple Tracker.

The Motivations to create this program were:

1. Create a First Native Tracker on the platform, I must confess I had never made a tracker, so starting with the simplest sound chip allowed me to concentrate on the tracker features rather than the sound chip complexity.
2. Explore & understand how to use the PSG to produce sound.
3. Implement a routine in BASIC that gets called through an event to play the music.

The Program is operated exclusively with the keyboard and uses the 320x240 graphical screen mode, mixed with Text.

UNDERSTANDING THE PSG

The PSG chip supports 3 square wave channels, and an additional noise channel, useful for playing Rhythm sounds.

On the F256 the range of playable notes on the PSG goes from A2 to B7, mostly due to the relation between the PSG and the clock signal (all tones produced are derived from the CPU clock), here's an excerpt from the F256jr Reference manual that explains it better:

The frequency generated depends upon the system clock driving the chip and the 'n' number provided in the frequency register of the PSG.

The relationship between these numbers is:

$$f = C/32*n$$

where f is the frequency produced, C is the system clock, and n is the number provided in the register.

Expressed in a different way, the 'n' value we need to produce a given the frequency can be computed as:

$$n = C/32*f$$

For the F256 the system clock is 3570 Hz (3.57 MHz) thus:

$$n = 111,552/f$$

Note: The constant used is a bit different from the one suggested by the F256jr Reference manual (111563) but the one I use its divisible by 32, and it seems to my ear that it produces better sound by doing a better note frequency quantization (yeah, it's my theory & I'm sticking to it!)

The **n** value expected by the PSG chip must be a 10-bit value, knowing this restriction we can demonstrate why the lowest note supported is A2, which has a frequency of *110.0 MHz*

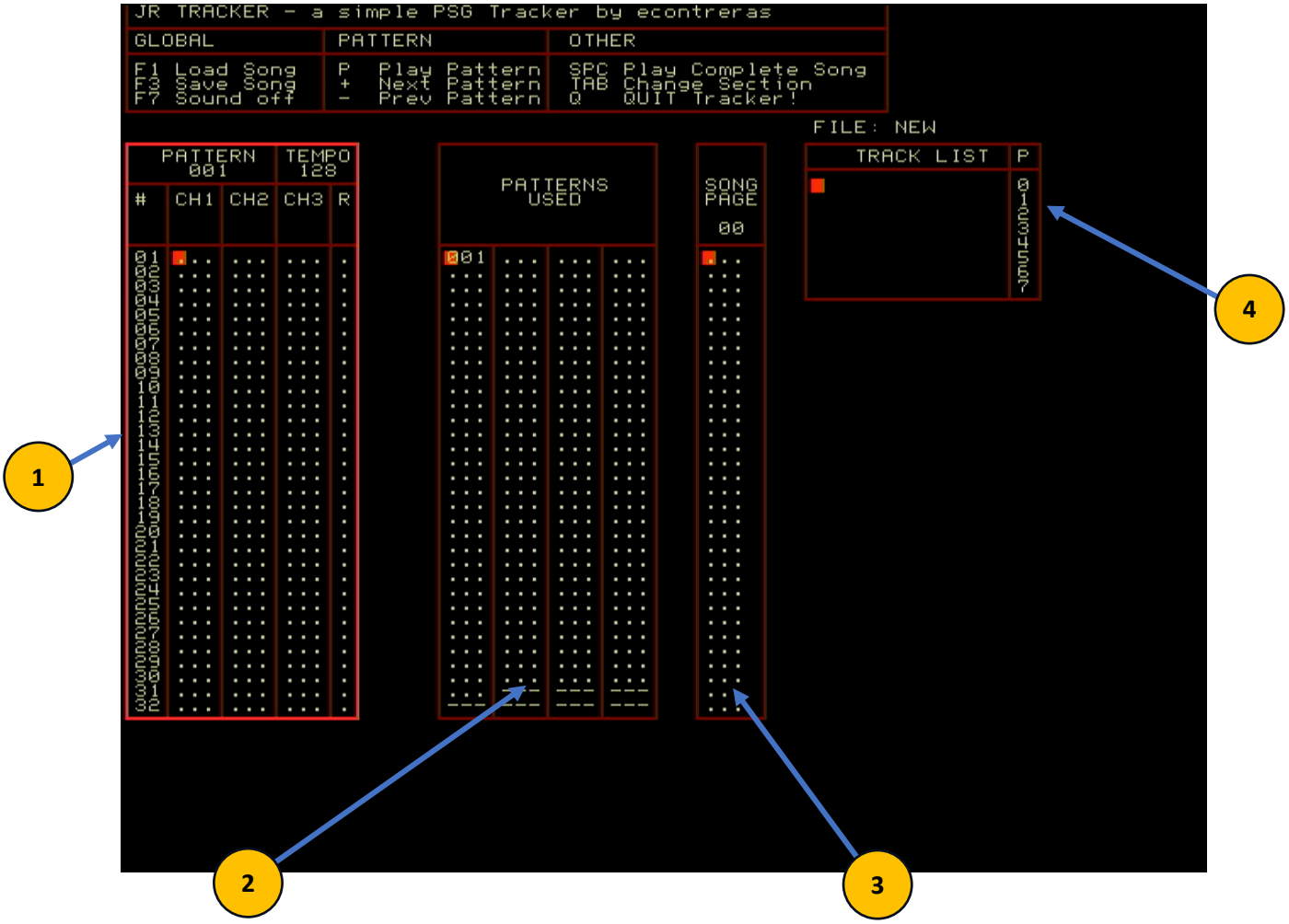
Let's do the math:

$$n = 111,552/110 = 1014$$

1014 is the maximum note value that fits in a 10-bit value (which can represent values between 0-1023), the next lower note: F#2 at 103.8 MHz, produces a value of 1074 which is outside the 10-bit range.

THE TRACKER INTERFACE

The Tracker divides the screen in various sections:



1. Pattern Edit Section
2. Pattern Manipulation Section
3. Song Sequence
4. Track List in Memory

The program operation will be discussed as we review each one of the sections of the screen.

MOVING BETWEEN THE SECTIONS

To move between the different sections of the tracker you can use the {**TAB**} and {**SHIFT TAB**} key combinations to move left & right between the sections.

For those daring souls that have a F256 JR, it has an option to connect a c64 keyboard, which unfortunately lacks the {**TAB**} key, so to support this group of users an alternate set of keys is provided. The {**<**} and {**>**} keys will be equivalent to using the TAB keys.

To highlight which section you are on, the outer border of the current section is drawn in a different color, this way you can easily identify the active section.

PATTERN EDITOR SECTION

A pattern consists of 32 steps, each roughly representing a time slice of 1/16th of a note, so the equivalent of 2 whole musical measures (8/4) can fit in one pattern.

The first 3 columns (channels) of the pattern are mapped to the melodic voices of the PSG and can play notes, this allows the song to play up to three simultaneous notes for melody or chords.

The pattern Editor allows you to enter the notes in these channels by using the following musical notation:

E # 3

Note	*Sharp (optional)*	*Octave*

This means that you can use the following keys to enter notes:

Key(s)	Element
{**a**}...{**g**}	Notes
{**#**} | {*****}	Sharp symbol
{**2**}...{**7**}	Octave

The fourth column in the Pattern controls rhythm, designated by "R", which allows the noise channel to play short noise sounds to simulate percussion sounds.

While the noise channel can produce noise in user selectable frequencies it does so at the expense of using the 3rd melodic channel to set the frequency and basically sacrificing this melodic channel whenever a beat sounds. To avoid this, by design, the rhythm channel will only use the preset noise frequencies built-in on the PSG, limiting the rhythm to using only 3 different sounds, the upside is that these beats can be used at the same time as the 3rd melodic channel.

The Rhythm column thus can only use the following values:

Value	Sound
{**0**}	Acute Beat
{**1**}	Mid Beat
{**2**}	Mid-Low Beat

Note Off/Flow Control Commands

Key	Command	Explanation
{**o**}	Note Off	A note is continuously played until a "note off" event (OFF) is encountered in the pattern to silence it. A "note off" command helps separate one sound from another, else the notes sound "tied"
{**z**}	End of Pattern	If you don't want to use all 32 steps in a pattern you can place an "end of pattern" (EOP) command that will force the pattern to end at this spot. While playing a song this will make the song jump to the next pattern in the song sequence.

Action Controls in the Pattern Edit Section

These key combinations allow you to move within the pattern, move to a different pattern, erase notes, play test the current pattern or mute the sound.

Key	Action	Explanation
{**CRSR KEYS**}	Move in Pattern	Pressing the cursor keys lets you move within the pattern to select other channels (horizontal moving) or other steps (vertical movement).
{**+**},{**-**}	Switch	Let's you move to the next or previous pattern,

	Patterns	without switching to the Pattern section, very useful while editing your music.
{**DEL**}	Erase Note	Deletes the note under the cursor.
{**P**}	Play Pattern	Plays the notes in the current Pattern.
{**F7**}	Sound off	While testing your pattern with "play pattern" if your pattern doesn't end in a "note off" command you can restore peace & quiet by pressing this key.
{**T**}	Tempo	Allows you to set the TEMPO of the song from 0 (slow) to 255 (fast), be aware that the TEMPO is a global setting for the whole song.

PATTERN MANIPULATION SECTION

This section shows you all the patterns that currently have data, there are 121 patterns available for usage (This is a design limitation to keep the maximum memory requirements for storing songs and pattern sequence to 16k).

Some actions can be executed over the patterns in this section, please verify the action table below.

Action Controls in the Pattern Edit Section

Key	Action	Explanation
{**CRSR KEYS**}	Move between patterns	By using the cursor Keys, you can quickly move between patterns.
{**ENTER**}	Edit Pattern	Pressing enter over a Pattern will show the Pattern data in the Pattern Edit Section
{**DEL**}	Clear Pattern	Pressing the delete key will ask if you really want to clear all data from the selected pattern, pressing {**y**} will clear the data.
{**s**}	Append to Song Sequence	Will add the current pattern at the end of the Song Sequence in the Song Page section.
{**r**}	Replace actual sequence	Will replace the selected step in the song sequence section with the current pattern

SONG SEQUENCE SECTION

The Song sequence section allows you to select the patterns that will play in your song, you can order and repeat patterns as you like, combining them as you see fit. Each song page has space for 32 Patterns, but you can create songs that use more than 32 patterns, in this case the song will occupy multiple "song pages".

There are only a few actions available while you are in this section, please review the table below.

Action Controls in the Song Sequence Section

Key	Action	Explanation
{**CRSR KEYS**}	Move through the song patterns	By using the cursor Keys, you can move between the patterns forming the song.
{**DEL**}	Delete Last Pattern from sequence	Pressing the delete key will delete the last pattern of the song
{ **[** } | { **]** }	Move to another page	You can use these keys to move to the next previous Page, useful if your song uses more than one page

TRACK LIST SECTION

The ".trk" file format used for the tracker allows you to have up to 8 songs in the file. The Track List Section shows then names of these 8 songs contained in memory as well as the initial "song page" where they begin. Empty (or unnamed) song tracks show the £ character as a name.

If you are creating a new song and wish to give a name to your creation, move to this section and start typing a name for the song, please mind that only 14 characters are allowed, so be creative!

In some cases, you might want to adjust the initial page of a song, for example if you have your first song using page 0 and 1, you will need to adjust the initial page for the next song to 2. You can do this by moving your cursor the number at the right of the song name and pressing {**0**} – {**7**} to select the correct page.

Action Controls in the Song Sequence Section

Key	Action	Explanation
{**CRSR KEYS**}	Move through songs & initial page	By using the cursor Keys, you can move between the different songs and their initial page.
{**DEL**}	Delete song name & enter name edit mode	Pressing the delete key will delete the song name and enter edit mode to input a new name, press {**ENTER**} to exit edit mode.
{**ENTER**}	Select song	Pressing Enter selects the current song as the active song and changes the song page to the initial song page of the selected entry

GLOBAL CONTROLS

There are a few Keys that work no matter on what section you are positioned, to execute actions, here's the list of such controls.

Global Action Controls

Key	Action	Explanation
{**F1**}	Load a song file	Load a tracker file (".trk") Consider that you don't need to type the extension.
{**F3**}	Save song file	Save all songs in memory in a tracker file (".trk") for later usage in the tracker. Consider that you don't need to type the extension.
{**SPC**}	Play song	Plays the current song from the patterns in the "song sequence" section
{**q**}	Quit program	End your tracking session

TRACKER FILE FORMAT

The Format for Tracker files is custom designed, although somewhat inspired by PSGTracker*, but modified to fit up to 8 songs into 16Kb of data.

*https://www.youtube.com/watch?v=thbTjPRdDjk
*https://mus.msx.click/index.php?title=PSGTRACKER_PSG_FILE_FORMAT

HEADER - 8 Bytes ***Address: 0-7***

Byte	Field Name	Explanation
0-2	ID	Contains ASC code for "PSG" letters
3	Sub Version	Contains 1, which is the current subversion
4	Patterns Pages	Total number of patterns Pages – fixed to 8
5	Pattern Size	Number of steps in Pattern – fixed to 32
6	Patterns Used	Patterns Used in File
7	Number of Songs	Number of songs in file

SONG INDEX – 128 bytes (16 Bytes per entry, 8 Entries) ***Address: 8-136***

Byte	Field Name	Explanation
0	Tempo	Tempo of the song (0-255)
1	Song Page	Initial Song Page of the song
2-15	Song Name	14 letter Song Name

FUTURE EXPANSION – 118 Bytes Reserved ***Address: 137-255***

Song Pages – 256 Bytes (32 Bytes each, 8 Entries) ***Address: 256-511***

Byte	Field Name	Explanation
0	Pattern	Pattern number in song, 0=empty pattern, 1-124 are valid patterns to play

PATTERN DATA – up to 16384 Bytes (128 bytes x 124) ***Address: 512-16384***

Each pattern consists of 32 rows of the following structure:

Byte	Field Name	Explanation
0	Channel 1 Data	Contains Values for notes or Note/Flow command
1	Channel 2 Data	Contains Values for notes or Note/Flow command
2	Channel 3 Data	Contains Values for notes or Note/Flow command
3	Noise Channel Data	Contains values from 0-2 for rhythm element

NOTE VALUES / COMMAND CODES

The following values are used in patterns to represent corresponding notes & commands.

C 1	1	**C #1**	2	**D 1**	3	**D#1**	4	**E 1**	5	**F 1**	6
F#1	7	**G 1**	8	**G#1**	9	**A 1**	10	**A#1**	11	**B 1**	12
C 2	13	**C #2**	14	**D 2**	15	**D#2**	16	**E 2**	17	**F 2**	18
F#2	18	**G 2**	20	**G#2**	21	**A 2**	22	**A#2**	23	**B 2**	24
C 3	25	**C #3**	26	**D 3**	27	**D#3**	28	**E 3**	29	**F 3**	30
F#3	31	**G 3**	32	**G#3**	33	**A 3**	34	**A#3**	35	**B 3**	36
C 4	37	**C #4**	38	**D 4**	39	**D#4**	40	**E 4**	41	**F 4**	42
F#4	43	**G 4**	44	**G#4**	45	**A 4**	46	**A#4**	47	**B 4**	48
C 5	49	**C #5**	50	**D 5**	51	**D#5**	52	**E 5**	53	**F 5**	54
F#5	55	**G 5**	56	**G#5**	57	**A 5**	58	**A#5**	59	**B 5**	60
C 6	61	**C #6**	62	**D 6**	63	**D#6**	64	**E 6**	65	**F 6**	66
F#6	67	**G 6**	68	**G#6**	69	**A 6**	70	**A#6**	71	**B 6**	72
C 7	73	**C #7**	74	**D 7**	75	**D#7**	76	**E 7**	77	**F 7**	78
F#7	79	**G 7**	80	**G#7**	81	**A 7**	82	**A#7**	83	**B 7**	84
OFF	85	**EOP**	125								

Shaded notes are defined but are not playable by the PSG.

LIMITATIONS, KNOWN ISSUES

TBD.

OTHER UTILITY PROGRAMS

TBD.

FUTURE FEATURES & IDEAS

Create a new version of the tracker to play SID Music

CONTACT INFORMATION

Hopefully you'll enjoy using this program as much as I enjoyed creating it, if you have any comment or suggestion feel free to contact me: ***Ernesto Contreras***

Email	eacontrerasd@gmail.com
Discord Server	https://discord.com/channels/691915291721990194/1008139105386889346 username: ***econtrerasd***