Microbe v1.3 Reference manual

Contents

1 Overview

Microbe is an all-in one electronic music studio for PalmOS© Handhelds. It takes advantage of the computing power of the ARM-powered devices and of the advanced sound manipulation features offered by the OS 5.0 to generate live, cool, synthetic or sampled sounds. You are a DJ and want a tool to generate fill-ins, drum patterns, cool synth sounds or sampled jingles? You like composing music while commuting? You think that the end of laptop music is coming soon - an omen to a new era of palmtop musicians? **Microbe** is made for you. Enjoy!

2 System requirements and Installation

Microbe runs on any ARM-powered device running PalmOS© 5.0, and supporting the standard PalmSource Sound APIs. It has been especially tested (and developped) on the Tungsten T Series. High resolution is not required, however, **Microbe** will look hotter and better on a hi-res screen. The support for Sony Cliés handheld is done through the installation of a hack emulating PalmOS© 5.0 Sound API. **Get it, it is free**!

Microbe is made of a main application database and an auxiliary database containing sound samples. This file is not required to run **Microbe**, however, if it is not present, you will have to find your way with the 15 built-in TR-808 / TR-909 samples, and it won't be that fun.

Microbe can load samples in .WAV files stored on VFS volumes. However, the presence of a VFS slot / card is not required.

After a several minutes of use, the demo version of **Microbe** stops playing sound - and doesn't allow you to change the name of your save files. Too bad... If you think that it is not fair, please register!

> The built-in speaker is... well... erhh... lousy? Don't make any conclusion on **Microbe** before having heard it using earphones, or, better, connected your Device to a stereo system or mixing desk - the sound quality is then rather good. Note also that on various devices, the built-in speaker only plays the left (or right) channel, so the pan control might behave as a volume control!

3 Architecture
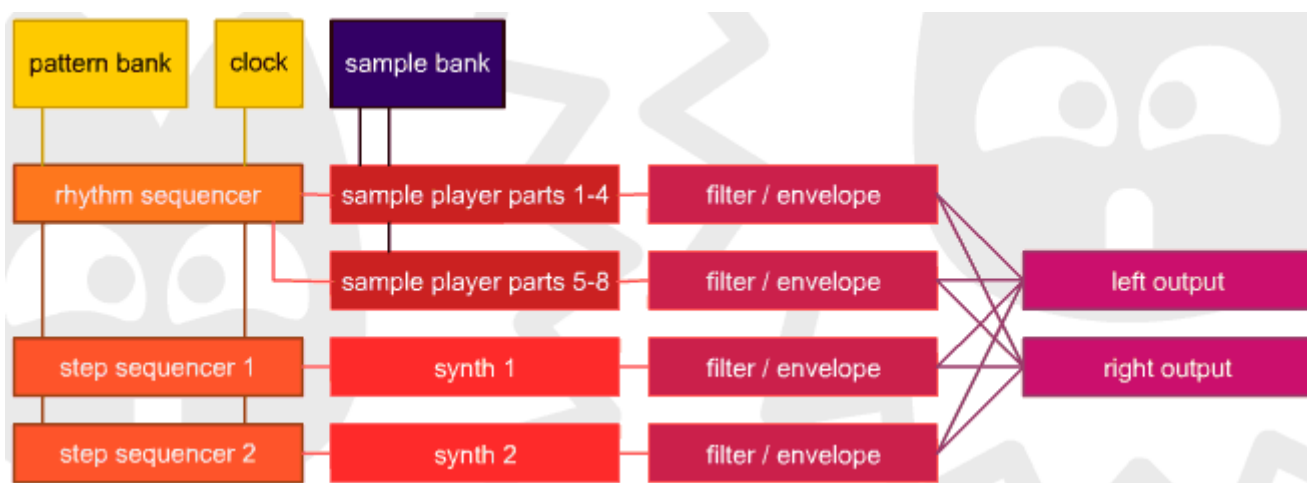
If you are used to music gear, synthesizers, and software such as FruityLoops or Rebirth-338, you will understand this diagram quite well and you can skip this section. Otherwise, read this section carefully.

There are 10 sound sources in **Microbe**. Two of them are analog bass synthesizers emulators. The eight other sound sources are simple sample players. Samples are loaded in memory from the built-in **Microbe** sample database, **Microbe SamplePacks**, or wav files stored on VFS volumes! Sample packs are standard Palm databases files containing dozens of samples - much more handy than .wav files on VFS! The sampled sources are gathered in two groups: the sample players 1-4 are mainly used for drums and are therefore referered as **drums**, the sample players 5-8 are mainly used for special effects and are refered as **extras**. Thus, we have finally 4 voices: two for the synthesizers, one for the drums, one for the extras. For each of these tracks, it is possible to control filter parameters, volume and pan. The result is mixed to the stereo output.

Now, you wonder: How do I play these instruments? Well, we won't make you hit the screen in rhythm, because it would be tedious and since you have a computer in your hand, let me introduce you to the... **sequencers**! Both the synth sections are controlled by two distinct step by step sequencers. In a step sequencer, notes are to be entered one by one, for the time-span of one bar. The sample player sections are controlled by a rhythm sequencer. The principle of such a sequencer is simple: the bar is divided into 16 boxes, and you tell whether an instrument of the kit has to be played for each box. Vintage drum machines such as the Roland TR-808 or TR-606 use such an approach, and you'll realize soon that it is very intuitive indeed.

As we have just seen, what you program in the sequencers is short: one bar - just a few hundredth seconds. Such a small chunk of music is called a pattern. You can create up to 32 patterns with **Microbe**, and loop or chain them.

Finally, the heart of the system: the clock, which generates rhythmic pulses. You can control its rate by setting a BPM value (beats per minutes).

4 User Interface

Command bar

These 3 icons allows you to perform some of the most important operations.

- The first **Microbe** icon displays a menu with all the possible tasks, like opening, saving a setup, loading the samples, etc. You can as well display the menu using the *Menu* silk key.

- The second **Play** icon starts playing. Note that it turns then into a **Stop** icon, that will... stop the playing.

- The last **Chaining** icon deals with the way the patterns are edited. Refer to the next section for more information.

Pattern bank

This block lets you select a pattern to work with. **Microbe** stores thirty two patterns, in four groups referenced by letters A, B, C, D. The eight selectors on the left let you chose a pattern within the group selected with the four selectors on the right.

When you start playing a chain of patterns or a song, two cases are possible: either you want the selected pattern to be changed as the chain is played (for example, 1, 2, 3, 4, 1, 2, 3, 4: the coloured square will move along with the different bars played, to always reflect what is presently played) ; or you just want the selected pattern not to change, for example if you want to edit it. To switch between one or the other of these behaviours, use the **Chaining** icon:

Edition lock is **on**: the current pattern is fixed and you are free to edit it.

Edition lock is **off**: the current pattern reflects what is played.

Tempo indicator

The tempo in Beats Per Minute (BPM) is selected here. A **tap tempo** feature is implemented: just click on the numbers to quickly adjust the tempo. An additional option allows you to select the time reference used for the tap. By default, one tap = one beat, but you can as well use one tap = 2 beats to accurately adjust very fast tempos.

Mute

This bar allows you to mute individually one of the eight sample players. When the selector is disabled, the corresponding track will be muted.

Track parameters
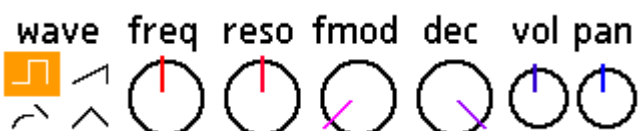
This is where some interesting things happen! These 7 controls allow you to alter and play with the sound from the 4 main tracks. To change the value of a controller, tap on one of the knobs, hold the pen down and move it upward or downward to inrease or decrese the value ; then release when you are done.

|	
wave	This control selects the oscillator **waveform** for a synthesizer track. It has no effect for one of the rhythmic / effects tracks. The possible choices are (from bottom left to right down):
	• Square wave
	• Sawtooth wave
	• Wicked sawtooth wave
	• Triangle
freq	This control modifies the filter **cutoff** frequency. Turn the knob clockwise for a brighter, sharper sound.
reso	This control modifies the **resonance** of the filter. Turn the knob clockwise to enhance the harmonics in the neighbourhood of the cutoff frequency. A very high value can lead to auto-oscillation. Watch out!
fmod	This control modifies the **modulation amount** of the envelope on the filter cutoff.
dec	This control selects the envelope **decay time**. Note that this has no effect if no modulation is applied to the filter cutoff. The filter envelope is retriggered for every step in the rhythmic section, and for every non-slided step in the synthesizer sections.
vol	This control modifies the **volume** of the track.
pan	This control modifies the position of the track in the **stereo** image.

Selectors

part ▼ synth 1   **sq** ▼ sy1   ▼ bd

The screen of a Palm Device is small, too small to display a telephone switchboard on it! Thus, the parameters of only one track can be modified at a time with the controllers of the **Track Parameters** section, only one rythmic part can be edited at a time, and only one synthesizer can be displayed.

	The first popup list lets you chose which track you want to tweak among the 4 tracks of **Microbe**.

The second popup list lets you chose which of the two step by step note sequencers you want to program.

The last popup lists the different **parts** of the rhythm sequencer that can be edited. Well, the two last items, are not really *parts*:

- 'acc' lets you modify the accentuation. All the steps on which the accentuation is set will sound louder.

- 'cut' lets you define the end of a pattern. For example, if 'cut' is set at the 13th step, only the 12 first steps of the patterns will be played. It is obviously useless to add more than one 'cut' in a pattern, as well as entering a 'cut' on the first step.

Step by step note sequencer
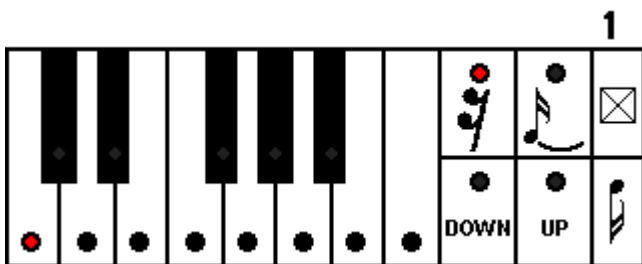
This is where you enter quirky basslines and acid leads! A bar (pattern) is divided into 16 steps. The number at the top right of the control indicates the step currently edited, from 1 to 16. Use:

	Next; to move to the next step. After step 16, the step cycles and starts again at 1.
	Back; to move to the previous step.

A note has to be selected for each step. The selected note appears with its red 'LED' lit. Several extra options are modified with the group of buttons on the right:

	The octave selectors; to specify at which octave the note has to be played. When **down** is switched, the note is transposed one octave down.
	When **up** is switched, the note is transposed one octave up. Note that when both **down** and **up** are switched, the pitch is unchanged.
	Rest; If **Rest** is switched, no note will be heard at the current step. Otherwise, the note specified on the mini-keyboard and with the **up** and **down** selectors is played.

Slide and tie ; When switched, the envelope of the note won't be retrigged and the pitch will fall down or raise from the pitch of the previous note, creating a slide effect. If the pitch of the previous note is the same, it will simply result in a longer note.

<Sequencer minifaq>

Q. Why do I need to enter a note value when there is a rest and no note will be heard?
A. Several answers:

- To the beginner: Don't care.

- To the advanced users: A slided note might be entered after the silence. For example, if you enter at step 1 C2 + rest and at step 2 C3 + slide, you will heard a slide from C2 to C3 at the step 2.

Q. What if my melody requires more than 1 bar.
A. You can chain patterns and enter the continuation in the following patterns.

Q. How can someone write music with such a crappy HCI?
A. Well, there are people who never learnt music another way than with such interfaces.

Q. What if my melody requires exactly 37 steps or uses microtonal balinese intervals?
A. Get lost.

Q. What if my melody requires exactly 13 steps and does not use microtonal balinese intervals?
A. You can insert a 'cut' on the 14th step. To do that, select 'cut' in the rhytmic part selection popup (the last popup on the right) and tap on the 14th step.

Q. What happens if I collapse the DIA of my T3, T5 or Clié?
A. Wow! The two bassline synthesizers sections are displayed simultaneously!

</Sequencer minifaq>

Rhythm sequencer

This is where you enter rhythmic parts. The bar is divided in 16 steps, each step is represented by a key. When the light is on, the rhythmic instrument will be played on this step. You can select which rhythmic instrument you want to program by using the **drum part popup**, or by using the **up** and **down** keys.
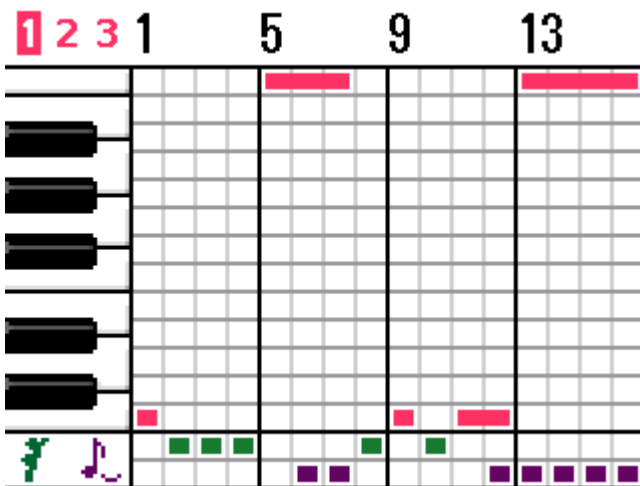
Advanced editors

You don't like using step-based editors? The advanced editors are made for you! To select one of them, use the **View** command of the menu.
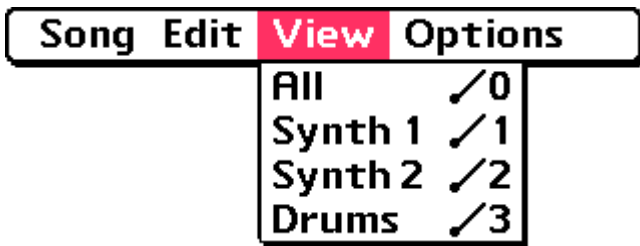
All switches to the standard view with the rhythm sequencer and a common bass synthesizer.

Synth1 / Synth2 displays a piano-roll to edit the first / second bassline synthesizer part:

You can:

- Tap on the numbers at the top left of the editor to change the octave.

- Modify the note for each step by tapping in the piano-roll

- Add or remove slides/ties by checking/unchecking the purple boxes at the bottom of the screen.

- Mark a step as a rest by checking/unchecking the green boxes at the bottom of the screen.

Drums displays a grid to edit rhythmic patterns:

Each line is an instrument (except the accent / soft parts) ; and each column is a step. Check/Uncheck... Easy! Note that you can also insert a "cut" to reduce the pattern length by tapping in the head row containing the numbers "1 5 9 13".

5 Commands

The command menu can be displayed by clicking on the **Microbe** icon on the top left corner of the screen.

New

After confirmation, this command resets the patterns, track settings**,** samples and tempo. Any data previously stored and not saved will be lost.

Open / Save / Save as...

Opens or saves the patterns, track settings and samples currently edited (a "song"). The following dialog boxes are displayed


| When loading... | When saving... |

Songs are identified by a title and an author; and stored in a database named Mboz_tblSongs. Note that the song data contains only the references (file names or records UniqueID) to the samples used ; not the samples data itself. Therefore, if the song made use of .wav files stored on a VFS card, it will require the card to be present when loaded again.

When saving, you can overwrite an existing song: just click on its name and don't modify the author or title. Modifying the author or the title means "save in a new position". Why is it so? Because it is not fair to overwrite someone else work! If you insist, the **Delete** menu command is there, waiting.

By the way, do you see the **Beam** menu command? Well, it does exactly what it looks like doing! However, there is the same restriction as with song saving: the samples data itself is not sent. As long as you use the samples contained in a sample pack that the receiver possesses, there won't be any problem.

Bhajis Loops export

This command exports the patterns, track settings and samples currently edited as a Bhajis Loops song. The following dialog box is displayed:
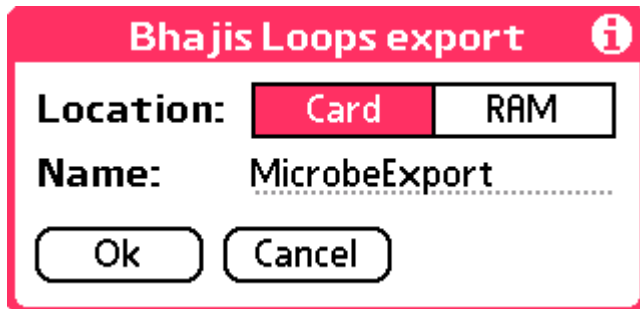
The song can be either saved in **RAM** or in the default directory used by Bhajis Loops to store modules on the present memory **card**. The name of the song must not contain any extension.

Render song

This command renders the song as a **.wav** file on the memory card. The file is named microbe_song.wav and is stored in the root directory.

Please note that a memory card has to be present to use this function.

Samples

Important stuff is done here: this is where you load the samples for each part of the drums or extras tracks. The popup on the top of the screen lets you select which part you want to edit.
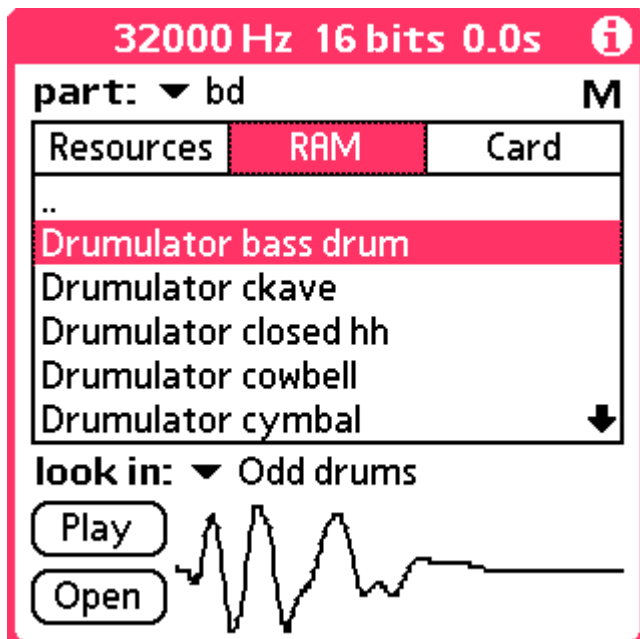
Samples can be loaded from 3 sources, selected by the selector at the top of the screen:

- From a limited set of samples, contained in the **Microbe** application **resources** themselves.

- From a **SamplePack** in **RAM** - in this case, the lower popup list displays the available sample packs, and the main list displays categories or samples by name, in the manner of a directory listing. Please note that unlike the version 1.1 or 1.0x, you can install as many sample packs as you want!

- From an **expansion card**. In this case, the lower popup displays the available volumes, and the main list displays the file system. Mono and stereo .wav files can be loaded, but stereo wav files will be converted to one single channel. You can as well tap on a .pdb file corresponding to a sample pack. This pack will be directly opened from the memory card and its contents will be displayed as if it were a directory!

The **Play** button works only when a pattern is currently playing - it uses one voice in the global architecture. Another consequence of this is that the filter, volume and panpot parameters of the corresponding part will be used.

Hold the stylus down on the **M** button on the right of the screen to display the amount of available memory.

Effects

For each one of the four sections displayed at the top of this dialog, you can:

- Select one of the two effect buses. This will route the signal of the part to the corresponding effect plug-in. You can disable the effect bus routing by tapping on the first empty box.
- Adjust the send level. This control modify the amount of signal to send to the effect unit. Note that if the effect is an insert effect (for example, a distorsion or EQ), you can't modify this amount.

The effects settings are modified with the controls at the bottom of this dialog. The three popup-lists display the available plug-ins for the first, second and global (master, **M**) effect buses. For each effect bus, a knob lets you adjust the return level.

The **Settings** icon allows you to display and modify the details and parameters of the plug-in.
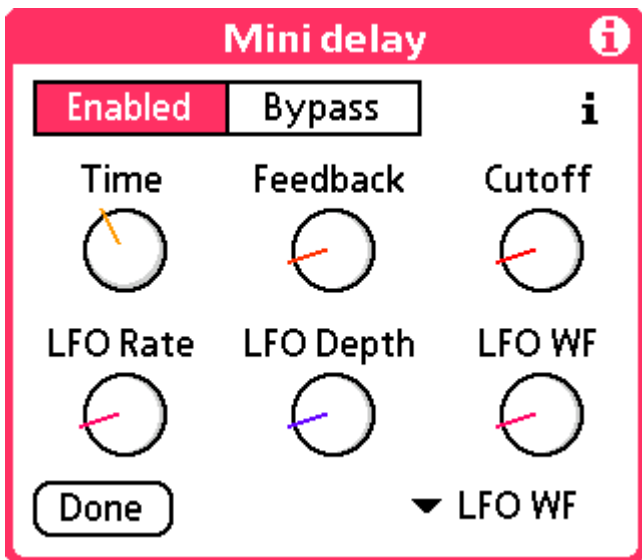
Note that the contents of this dialog box may depend on the plug-in you use. Some plug-ins (such as a simple distorsion) don't have any parameter, so this dialog box will be empty. On the other hand, some plug-ins might have a lot of parameters. In case the plug-in has more than six parameters, a popup box is displayed in the bottom right of the dialog, allowing the selection of the parameter modified by the sixth knob.

It is very important to understand a difference in the behaviour of **send** and **insert** effects. **Send** effects are for example reverbs or delays - effects adding something to the original signal and requiring to be mixed with it. In this case, the send level for each instrument is taken into account, and the result is mixed with the source signal. **Insert** effects are for example a distorsion or an equalizer ; they equally affect each instrument ; and the result replace the source signal.
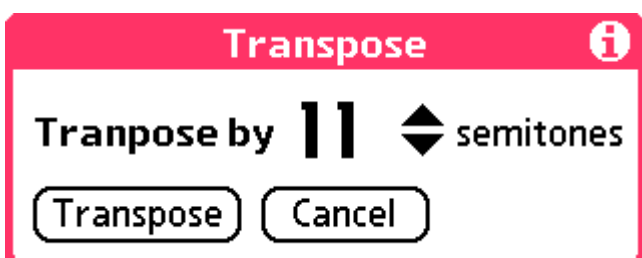
Plug-ins settings are saved with the song, but not the plug-ins themselves. Thus, if a song uses a plug-in which is not available ; it will be replaced by (none).

XY Pad
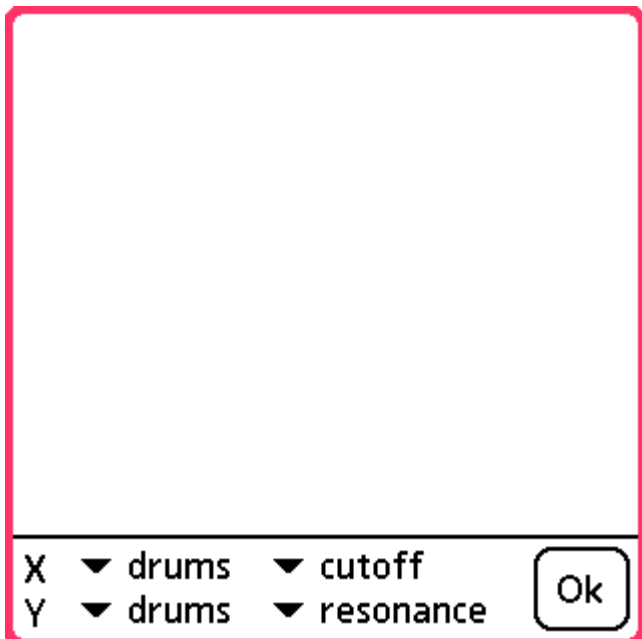
Huh, one big empty screen? Well, there is a lot of room left to let you play with your stylus! In the XY pad, the X and Y coordinates of the stylus control two parameters of the **Microbe** sound synthesis engine. There are hundreds of combinations - Feel free to experiment!

Transpose

A simple form, but it surely does well what it claims to do...

Pattern management

This command allows you to copy/swap data from one pattern to another. The source and destination are selected with the familiar **selector grid 1-8 A-D**™, and the tracks / parts to copy / swap are chosen with the array of checkboxes. Finally, the **Clear** button empties the selected tracks of the pattern in the From section.
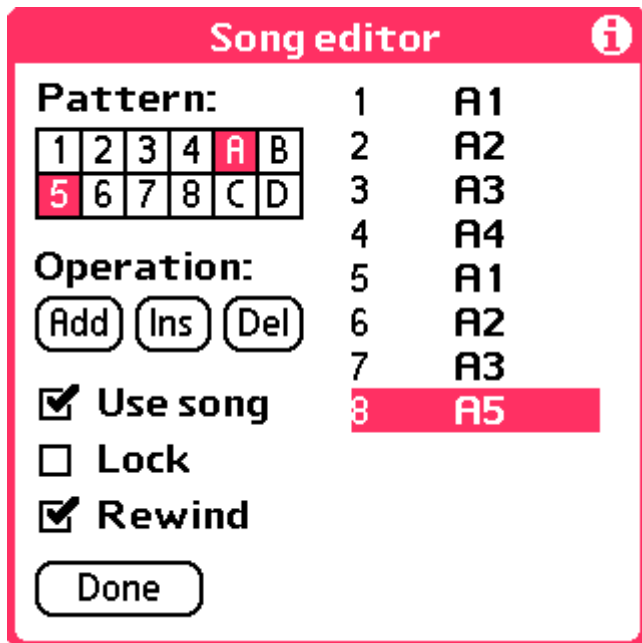
Chaining

This is here that the rules for pattern chaining are defined. A pattern is chained when it will be continued in the next pattern. The form displays the 8 patterns in the selected bank in a fashion that visualizes the way they are chained.

The rules are the following:

- Let xRy be the relation defined by the predicate "y = x+1 [32] and x is chained". R* is its transitive closure, R*' the symetrical extension of R* (xR*'y iff yR*x or xR*y). A chain is a connex component in the graph of R*'. Let chain(p) be the set of patterns in the same connex component as p (its equivalence class).

- If a pattern p is chained, then after the 16th step of p, the first step of the pattern p+1 will be played (or 1 if p = 32). Otherwise, the first step of the pattern p0 will be played, where p0 = min chain(p).

It is easier to understand the graphical representation, though.

Song editor

This forms lets you use an alternate way of chaining patterns, more suitable for the creation of complete songs.

The list on the right of the screen lists the bars and the corresponding pattern to play for this bar. You can use the buttons on the left to modify the pattern played at the selected bar, or to add, insert or delete bars.

The **Use song** check box enables the chaining of patterns as described in this dialog. Disable it to use the standard chaining scheme.

When the **Lock** checkbox is enabled, tapping on a bar in the list on the right won't take you to this bar ; and the display won't automatically scroll. This is useful when you want to edit the song while it is playing.

When **Rewind** is enabled, the playback of the song always starts from the beginning of the song when you press "Play" in the main screen.

View selector

Use these commands to switch from one of the **different views**

Song properties
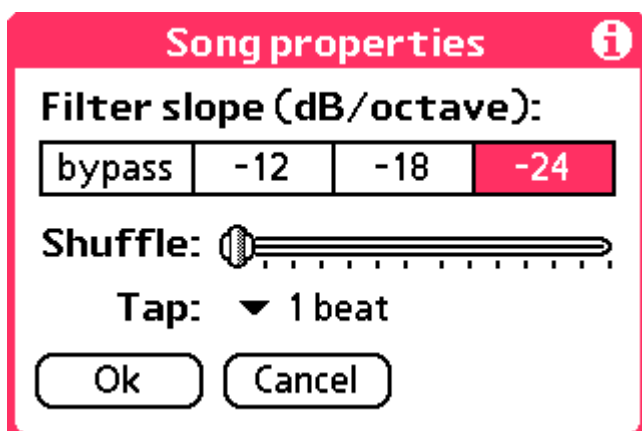
The **slope** of the filter in dB/octave can be set in this dialog box. -24 dB/octave is typical of Moog synthesizers ; -18 dB/octave of the classic Roland SH-101 and MC-202. -12 dB/octave will produce a softer result. The **bypass** option disable the filter. If you want to use **Microbe** as a simple sample player / drum machine, this setting might be useful. Especially, it allows you to use 44.1 kHz sampling rate even on a Zire71 / Tungsten T.

The **Shuffle** setting can help you to add a little more "human" feeling to your patterns, by creating a slight difference in the duration of odd and even steps. Instead of playing *paam paam paam paam*, the sequencer will play *paaam pam paaam pam*, the slider controlling the difference ratio between long and short steps.

Finally, you can also select the number of beats used when **tapping** a tempo.

Preferences
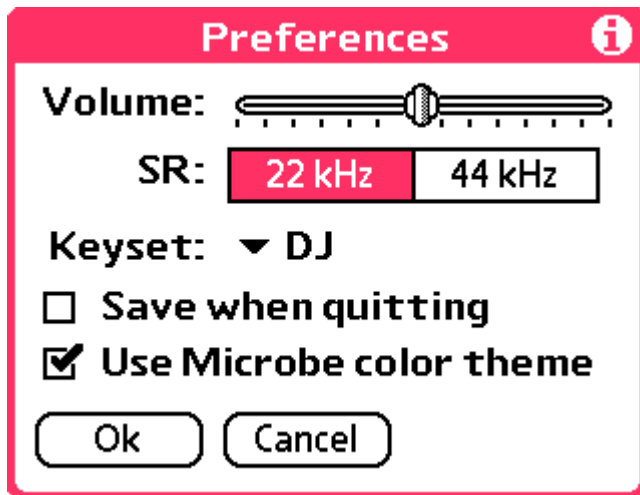
The **volume** slider is there. There used to be a little paragraph in this documentation about clipping, distorsion, and what will happen to your eardrums if you play Microbe all the day long, but since none of the Microbe registered users bothered with that, well, *do whatever you want with this slider*.

The **SR** (sample rate) lets you chose at which sample rate Microbe works. **22 kHz** works best on slower devices (Tungsten T, Zire 71), even if the sound quality is not the best ; while **44 kHz** is more suitable for the fastest devices. Note that using **44 kHz** on a slow device might result in discontinuities or glitches in the playing ; as well as a very slow reaction time of the user interface.

Three configurations are available for the **hard keys**. Refer to the **Keys** section for each of them.

You can enable the **saving of your song in** a temporary storage space so that it is restored when you restart the application. Useful when you press by mistake an application button or when you want to temporarily perform a task in another app! *Oh, poor non-registered user, he wishes he had this feature, too...*

Finally, if you can't stand **pink**, you can disable the default pink colour theme and get back to your favourite colour set.
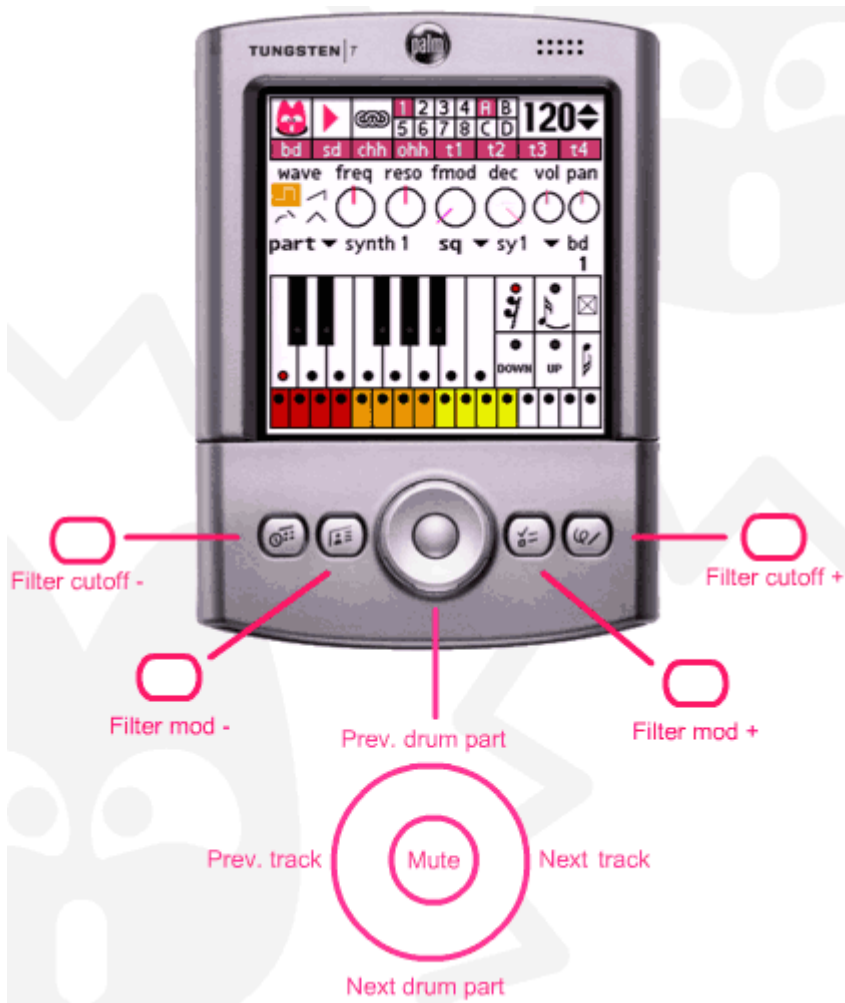

About

This is what only the nice registered users can see... To non-registered users, the symmetry is broken by an ugly **Register** button opening the following dialog box:
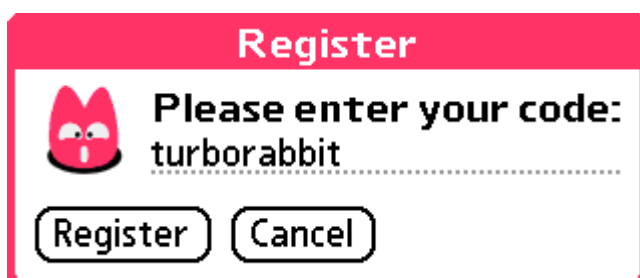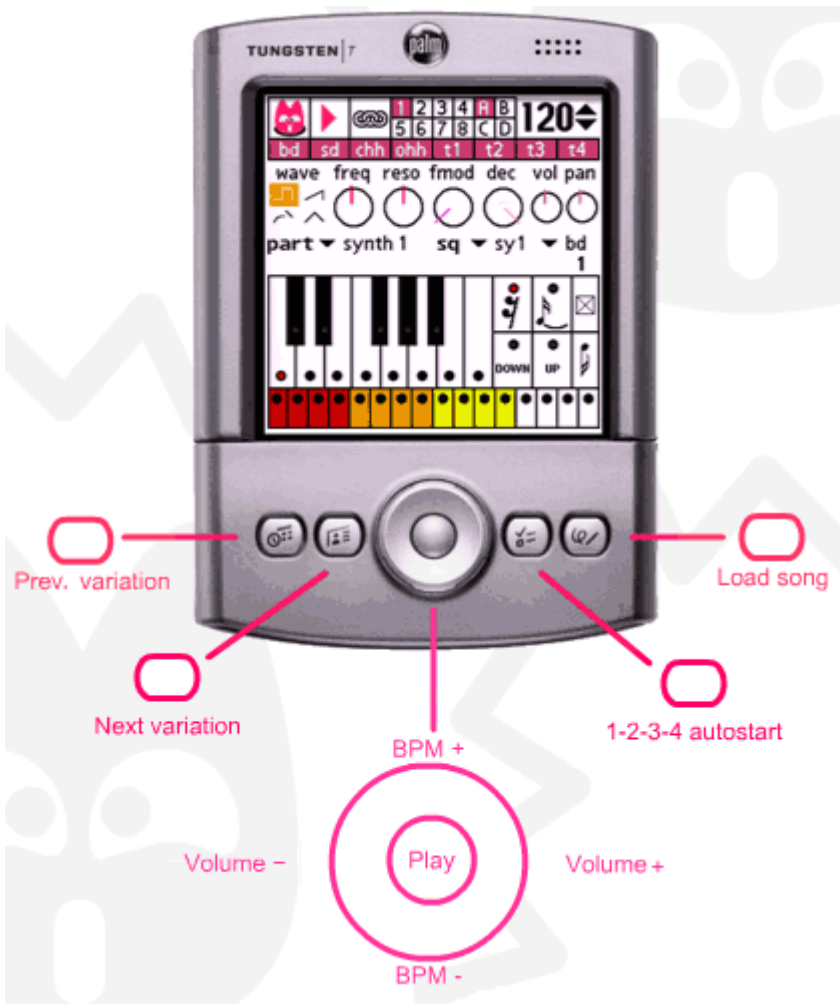
6 Keys

The following picture sums up how **Microbe** reacts to key events:
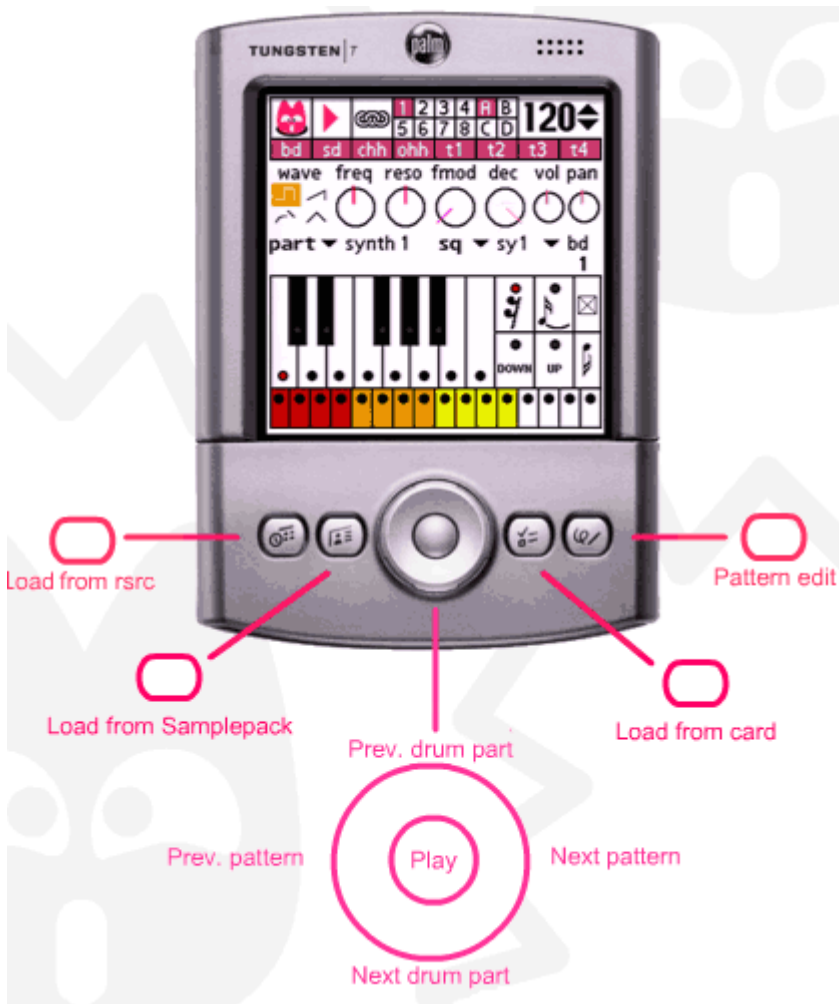
DJ keyset

Drum machine keyset

The 1.2.3.4. auto-start feature is useful if you use **Microbe** and your Palm device as a drum machine. Tap four times on the key to select the tempo, just like you do with the tap tempo feature of the BPM indicator ; and t seconds after the last press, where t is duration of a beat, playing will automatically start.

Beats programming keyset

The **5 ways navigator** is also supported in the sample edition dialog, and in the song opening dialog.

In the sample edition dialog, use the up and down arrows to scroll the samples list, the left and right arrows to select a sample type ; and the selector to load the selected sample.

In the song opening dialog, use the up and down arrows to select a song ; and the selector to load the song.

7 Additional information

Sample Packs

For now, only one sample pack is available -the **Starter sample Pack**. But new and cool sample packs will be released soon! The current sample pack contains 90 samples, classified in the following categories:

TR-808	12 samples
TR-909	17 samples
Roots (Hip-hop and breakbeat)	24 samples
Acoustic	17 samples
Ethnic (Tabla)	10 samples
Misc and SFX	10 samples

The utility used to convert a list of wav files into a pdb file compatible with **Microbe** can be obtained freely - however, this tool is still at the development stage and will not be commercially released.

How to crack microbe?

Little bastard!

Developing your own plugins or using Microbe plugins in your own music application

Basically, a plug-in is a resource database (type: Plug, creator: AudX) with the following resources:

type: **arml**, id: **1**. ARM Initialization code. A pointer to this code is passed to PceNativeCall when the plug-in is loaded. You should initialize look-up tables and buffers there.

type: **arml**, id: **2**. ARM Cleanup code. This code is called whenever you stop playing, and the first time you start playing. You should reinitialize state variables or clean buffers there.

type: **arml**, id: **3**. ARM Render code. This code is called whenever the content of a buffer is to be rendered.

type: **name**, id: **1000**. Plug-in name.

type: **info**, id: **1000**. A string containing credits, copyright, or other information. Typically displayed in the "about this plug-in" dialog.

type: **data**, id: **1000**. 4 16-bit words in motorola byte order (UInt16), indicating the size (up to 64kb) of the 4 data buffers required by the plug-in (for example, for delay lines). Must be 0 if no data is required.

type: **prms**, id: **1000**: one char indicating the number of parameters (up to 32 parameters can be used) ; then, for each parameter, its name (zero terminated string), and 3 motorola-ordered, signed 16-bit integers indicating the minimal, maximal and default value of the parameter.

type: **insr**, id: **1000 :** one char, equal to zero if the plug-in works in "send" mode ; one if it works in "insert" mode. Ideally, the output of a plug-in must be mixed with the original signal according to the following formulae:

- send mode: output = channel volume * dry + effect return * effect(channel volume * channel send * dry)

- insert mode: output = return * effect(channel volume * dry)

You can refer to the source code of the **delay plug-in** for more information.