



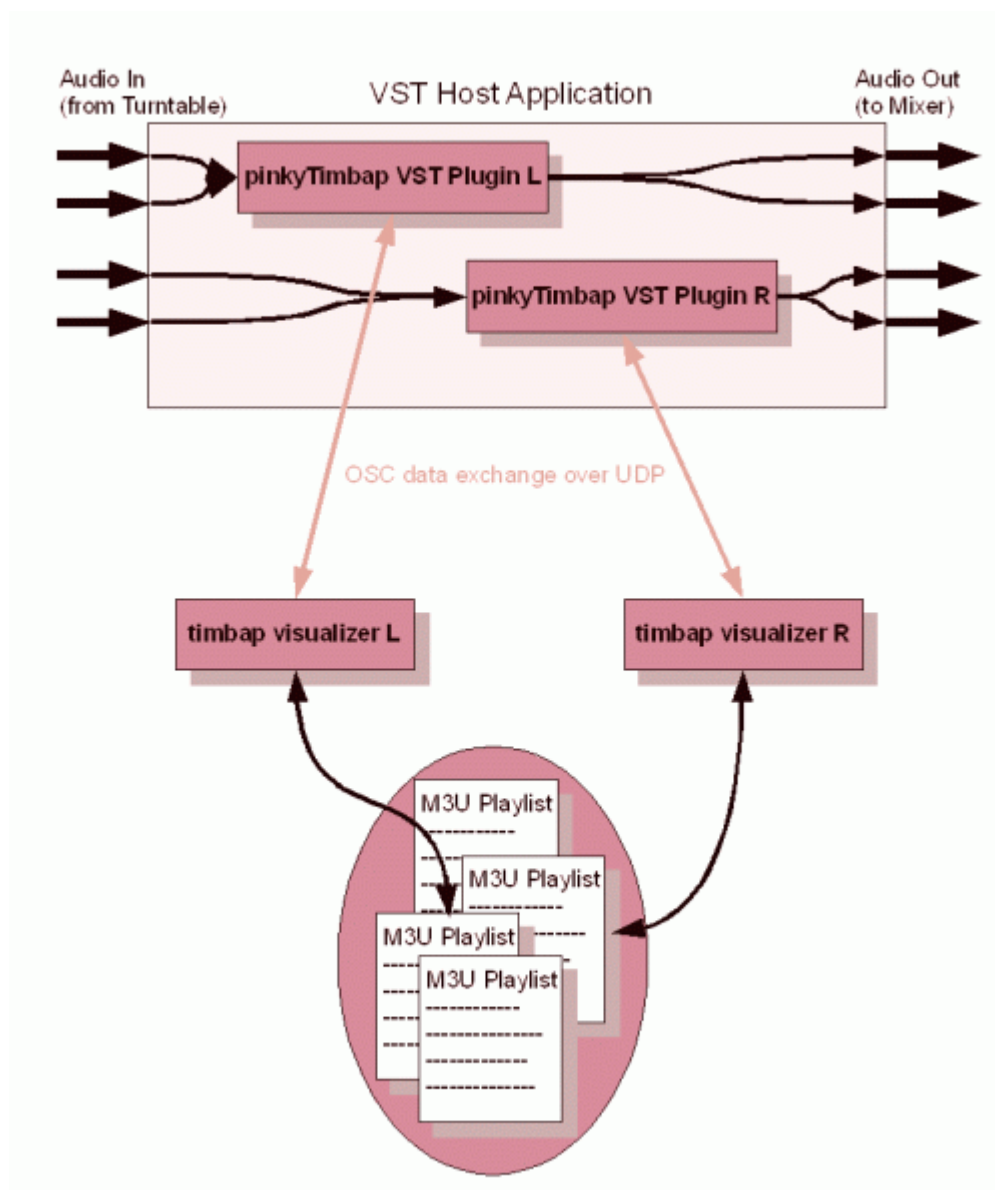
User Documentation

----- **IMPORTANT NOTE** -----

During the startup, your firewall may ask you to permit network access for the VST host application and for the visualizer. Please permit, as the audio component and the visualizer need to communicate via a network protocol.

Overview of the components of timbap:

The timbap digital DJing system consists of various components (audio, visualizing and connectivity). Both the visualizing and the audio component need to be running to make the timbap digital DJing system work. As the CPU load generated by the visualizer is higher you might want to start the audio component first to reduce startup time, but this is no must. You can find details on the installation of the timbap audio component in the VST host setup guide.



Contents

1. Hard- and Software Requirements.....	4
– 1.1 Hardware Requirements.....	4
– 1.2 Software Requirements:.....	4
2. The Audio Component.....	5
– 2.1 Introduction.....	5
– 2.2 Installation / Setup.....	5
– 2.3 Usage.....	5
3. The Visualizer Component.....	7
– 3.1 Introduction.....	7
– 3.2 Installation and First Steps.....	7
– 3.3 Basic Interaction with the turntable interface (A tutorial).....	8
– 3.4 Configuration of the turntable interface.....	10
4. Connectivity.....	12
– 4.1 M3U playlists.....	12
– 4.2 iTunes Library Playlist Extraction.....	12
– 4.3 Manual Artwork Assignment.....	12
– 4.4 Automatic Artwork Retrieval.....	13

1 Hard- and Software Requirements

1.1 Hardware Requirements

timbap can be controlled by two types of timecode vinyl: MsPinky control vinyl (gen.4) or Torq control vinyl. For hardware requirements there is not much experience yet. But a good graphics card with installed 3D drivers, a GHz processor, more than 512 MB RAM and of course a good soundcard with ASIO driver (or ASIO4ALL) are the key factors for a decent performance of timbap. If your soundcard has no phono inputs you will also need phono preamps of course.

1.2 Software Requirements:

Microsoft Windows (XP and probably Vista) and Mac OS X is supported as an operating system. There is some required software that you will have to download and install:

For the visualizer:

- **Java Runtime Environment (version 1.5 or higher)**

Is pre-installed on most contemporary computers. Check at <http://www.java.com>

For the audio component:

- **Cycling '74 Pluggo Runtime 3.6.1**

Windows: <http://www.cycling74.com/download/pluggoruntime361.zip> (4.8 MB)

MacOS X: <http://www.cycling74.com/download/pluggoruntime361.dmg> (10.2 MB)

- **A VST host** of your choice, for example Aodix (Win) or Tracktion (Win/OSX):

<http://www.aodix.com/files/aodixsetupv4201.exe> (2.0 MB)

You can install all this software anywhere you please.

2 The Audio Component

2.1 Introduction

Let's start right into the components of timbap with the audio component which handles the timecode processing and the audio playback. The audio component of timbap is provided as a VST plugin - the pinkyTimbap VST plugin. This means it is not a stand-alone application. Instead it is being executed in a host environment, a so called VST host application. This has numerous advantages. For example, you can easily plug in arbitrary effect plugins in the processing chain which affect the mp3 played by timbap.

As VST is an industry standard, the pinkyTimbap VST plugin can run in any host application providing a VST interface, for example Cubase, Ableton Live, FLStudio and also a light-weight host like Aodix which is freeware.

2.2 Installation / Setup

How the pinkyTimbap VST plugin has to be installed depends on the VST host application. In most cases however, the plugin file `pinkyTimbapVst.dll` or `pinkyTimbap.vst` (on Mac OS X) located in `<TIMBAP_INSTALL_DIR>\audio\vst\` simply has to be copied to a VST plugin directory configured in the VST host application.

A general overview of how to best set up the pinkyTimbap VST plugin for use in a VST host application is given in the separate document about VST host configuration. It also contains detailed instructions for some VST host applications.

2.3 Usage

The role of the pinkyTimbap VST plugin is two-fold. First it reads the input audio signal, extracts the timecode information and sends it to the visualizer component. On the other hand it receives "load" and "play" commands from the visualizer component and acts as some sort of remote controllable jukebox.

The pinkyTimbap VST plugin has only a small number of controls:

- OSC Sending Port / OSC Receiving Port

First, there are two port numbers which specify where to send OSC data to and where to receive OSC data from. Obviously, these settings are the opposite of the sending and receiving port configuration in the visualizer component. Otherwise there would be no data exchange.

As the data is sent using the network protocol UDP, please make sure that no other application already uses the specified ports and that your firewall permits the communication between the two applications

- Signal Level Threshold

This setting is concerned with the threshold level of the incoming audio signal. The slower a record is spinning the lower the level of the incoming audio signal will be. With a lower signal strength the results of the velocity and position analysis become more unstable. The threshold level sets the point where the incoming signal is being cut off because it is too low. Of course, if the threshold is set to a very high value, the interaction with the turntable won't be as natural anymore. So this is basically a tradeoff between stability and a natural interaction. You will notice that the output signal will also decrease as soon as the threshold level is set too high.

- Torq Vinyl

Switch this on, if you are using Torq Vinyl. If you are using MsPinky vinyl (generation 4), leave it switched off.

- Presets

Four presets are available. They correspond to the standard settings for the left and the right turntable when you are using either MsPinky or Torq vinyl.

3 The Visualizer Component

3.1 Introduction

The visualizer component is responsible for rendering the graphical display of the tangible user interface. It maintains an internal state of the current situation and also sends load/unload commands to the audio component. As the visualizer receives velocity and position data from the audio component, it can directly react to the turntable manipulation of the user.

It is realized as a Java application utilizing Processing and OpenGL. .

3.2 Installation and First Steps

Timbap is being distributed as a single ZIP file. To install timbap on your computer, simply extract the contents of the ZIP file to any place on your computer.

After extraction, you can directly launch the visualizer using the executable provided:

timbap-visualizer.exe (on Windows) or **run_macosx.sh** (on Mac OS X). Select the provided example playlist and wait until it is loaded. A message should appear telling you that no time code is available.

Now you can either set up the audio component right away (see document about VST host configuration) or first try the timbap audio simulator. The simulator is a utility that acts like the audio component. However, it is not being controlled by time code vinyl, but by some keys of your keyboard. Double-click the file **timbap-audio-simulator.jar** in the folder **audio/simulator**.

The visualizer should now receive a time code signal and you should be able to browse through the playlist using the keys as described in the simulator window. Note, that the keys will only work if the simulator has the focus (click inside the window).

3.3 Basic Interaction with the turntable interface (A tutorial)

If you have successfully started both the audio component (or the audio simulator) and the visualizer component, you are ready to start experimenting.

- Browsing Mode vs Playback Mode

Timbap has two modes. Which mode you are in affects how the turntable rotation is interpreted. In browsing mode, the rotation is used to scroll through your playlist items. In playback mode, it is applied to the playback speed of a selected audio file.

If the turntable is stopped, you will notice that the item at the top grows to full size and then either stays there or starts flipping and disappears again. What you can see is the switching between browsing mode and playback mode. It is always the item at the top position that will start growing.

As soon as it has grown to full size you are in playback mode. When it has disappeared again you are back in browsing mode. If you start your turntable in browsing mode, the playlist items will start rotating. In playback mode however, the full sized item will start rotating around its own axis like a record. If you are using the example playlist, you might miss the audio playback now. This is because only one item of the example playlist is actually distributed as an audio file: "Christian Dittmann - Dub 2".

You might also wonder why stopping the record won't bring you back to browsing mode any more. In fact, you are currently cueing the playlist item at a certain position. When you start the turntable again, you might notice that at a certain rotation (actually a rotation of zero degrees) there is an animation showing up at the top of the rotating item. This is a hint showing you the way back. Halting the turntable at a position where it is visible will bring you back to browsing mode.

Now play around with your turntable, use the pitch control, use the 33/45 switches, do a backspin, scratch the record, lift the needle, whatever you can think of. And see how the visualization reacts. You have already learned the basic interaction principles of timbap.

- Absolute Position vs Relative Position

We will further experiment with the browsing mode now. The playlist items are sorted alphabetically by artist name. At the end of the playlist there is a short gap, then the beginning of the playlist comes in again. The playlist is obviously too small to fill the entire record. You will however notice that resetting the needle to the end of the record brings you to the end of the alphabet and resetting it to the beginning brings you to the beginning. How does this work together?

Well, there's no magic involved, it's more like a hat players' trick. If you let the turntable rotate for a while, then lift off the needle and set it to the exact same position again, you will find yourself at a totally different part of your playlist. I'll explain: Whenever the needle is reset we get the absolute timecode of that position and calculate the position in the playlist. When the record is spinning however, the absolute positioning information is ignored and the personal speed preferences are respected. This way, the playlist is rather quickly out of sync with the absolute positioning information. Only with the next needle reset, the synchronisation will be restored. But actually you don't have to care about this.

- Audio Playback

So, if you are using the sample playlist, please select the track "Christian Dittmann - Dub 2". Try using the absolute positioning iteratively to get really close, if you want. You should hear an audio signal on the respective output and a semi-transparent circle segment overlay should appear. This segment is showing the playback position in the audio file. Of course, you can use the absolute positioning using the tone arm here as well. All audio files start at the beginning of the record. So if your needle is somewhere in the middle, you might have to reset it to the beginning. See how the circle segment grows, when you play the audio file from the very start.

Also try scratching the record. If your ASIO driver is configured correctly, audio latency will be very low, so that it is very much like the real thing. The visualizer component however can have a notably higher latency.

- Positioning Clouds

Most probably you already wondered about the line of letters you can see on the bottom right when you are in browsing mode. This is a so called positioning cloud. It works exactly like the alphabetic index that is printed on the side of some lexicons. We came up with this construct to give people a hint for the absolute positioning. The cloud shows you how some aspect is distributed over the playlist entries. Active by default is the artist name initials cloud. Every letter represents all the playlist entries starting with the respective letter. This way, you can jump quite exactly to a certain artist. If you are using a projector, the visual cue can even be calibrated to match the line where you can set your tone arm to. So that basically you are setting the tone arm directly onto the index.

You can switch to a different positioning cloud by using the left/right arrow keys or the arrow symbols displayed beneath the positioning cloud. You will notice that the playlist is in a different sort order after the switch, of course. Otherwise the displayed index would not be very helpful.

At the moment there are four different positioning clouds available: A cloud with artist name initials, one with song title initials, one with the predominating artwork color hue and one with the release year.

3.4 Configuration of the turntable interface

All the configuration can be done using the Java properties file `timbap.properties` in the `conf` folder. The following properties can be specified (possible property values are separated by "|"):

- OSC port settings (need to match with the settings in the pinkyTimbap plugin. A second timbap instance will automatically connect to the following pair of ports)
 - LISTENING_PORT = 3001
 - SENDING_PORT = 3002

- Default resource locations:
 - DEFAULT_ARTWORK_FILE = conf/timbap_logo.jpg
 - DEFAULT_PLAYLIST_DIRECTORY = playlists
- Screen dimensions and full screen mode:
 - FULL_SCREEN_MODE_ENABLED = false | true
 - SCREEN_WIDTH = 1024
 - SCREEN_HEIGHT = 768
- Settings for iTunes integration (see 4.2)
 - REPARSE_ITUNES_MUSIC_LIBRARY_ON_STARTUP = false | true
 - ITUNES_MUSIC_LIBRARY_PATH = C:\Documents and Settings\Administrator\My Documents\My Music\iTunes\iTunes Music Library.xml
- Preferences for the timbap user interface:
 - PLAYBACK_SCALE_FACTOR = 1.0
 - ITEM_COUNT = 3 | 5 | 7 | 9 | 11 | ...
 - VELOCITY_FACTOR = 1.0
- Texture Cache Size (A higher value leads to higher memory footprint. So you might have to give the JVM more memory using the Xmx and Xms parameters)
 - TEXTURE_CACHE_SIZE = 30
- Maximum Frame Rate (If your computer's CPU load is so high that audio dropouts occur, you might want to try and reduce this value in order to leave some headroom for the audio component)
 - MAXIMUM_FRAME_RATE = 40

4 Connectivity

4.1 M3U playlists

You can define your own playlists in M3U format and save them to the directory `<TIMBAP_INSTALL_DIR>\playlists` . Nearly any media player can generate an M3U playlist. We recommend Winamp as it reads out the ID3 tags of MP3s and adds them as EXTINF tags to the playlist, separating artist and title with a dash symbol. If you are using Winamp make sure that you are saving the M3U file directly to the playlists directory, otherwise the relative path information might be broken. Before saving scroll once through your whole playlist in Winamp, so that the ID3 tags are read and shown in the playlist viewer.

4.2 iTunes Library Playlist Extraction

Alternatively, if you are using iTunes, you can extract the playlists from your iTunes library. We provide the possibility to parse the iTunes library XML file on startup and generate M3U playlists for every iTunes playlist in the `playlists` directory. The filenames will be prefixed with "iTunes_", in order to avoid conflicts with other playlists.

To use this functionality you have to edit the properties file `timbap.properties` in the `conf` folder. Set `REPARSE_ITUNES_MUSIC_LIBRARY_ON_STARTUP` to true and `ITUNES_MUSIC_LIBRARY_PATH` to the full path of the file `iTunes Music Library.xml` . By default, this file is in the iTunes directory in your user profile.

4.3 Manual Artwork Assignment

You can manually assign JPG files as artwork to your playlist items. Timbap will search for a file in some locations in this order:

- A file with the same name as the audio file, but with the extension jpg, gif or png
- A file named `folder.jpg` or `Folder.jpg` in the same directory as the audio file
- Embedded artwork in the MP3 file

Images will internally be resized to the same width and height by timbap. To achieve the best results, your artwork files should have the same width and height, otherwise they might look quite odd. Note, that there is a bug which leads to non-quadratic artwork not being masked to a circular record item, so it is displayed as a full rectangle instead. In terms of image size there is a trade-off between quality and performance. A good choice is a dimension of around 400 x 400 pixels.

4.4 Automatic Artwork Retrieval

Currently there is no internal mechanism for automatic artwork retrieval any more. There are lots of tools available on the internet though that can embed artwork into your mp3s. And as mentioned in the previous section, such MP3 embedded artwork can be parsed by timbap.