
SamplerBox Documentation

Release 0.0.1

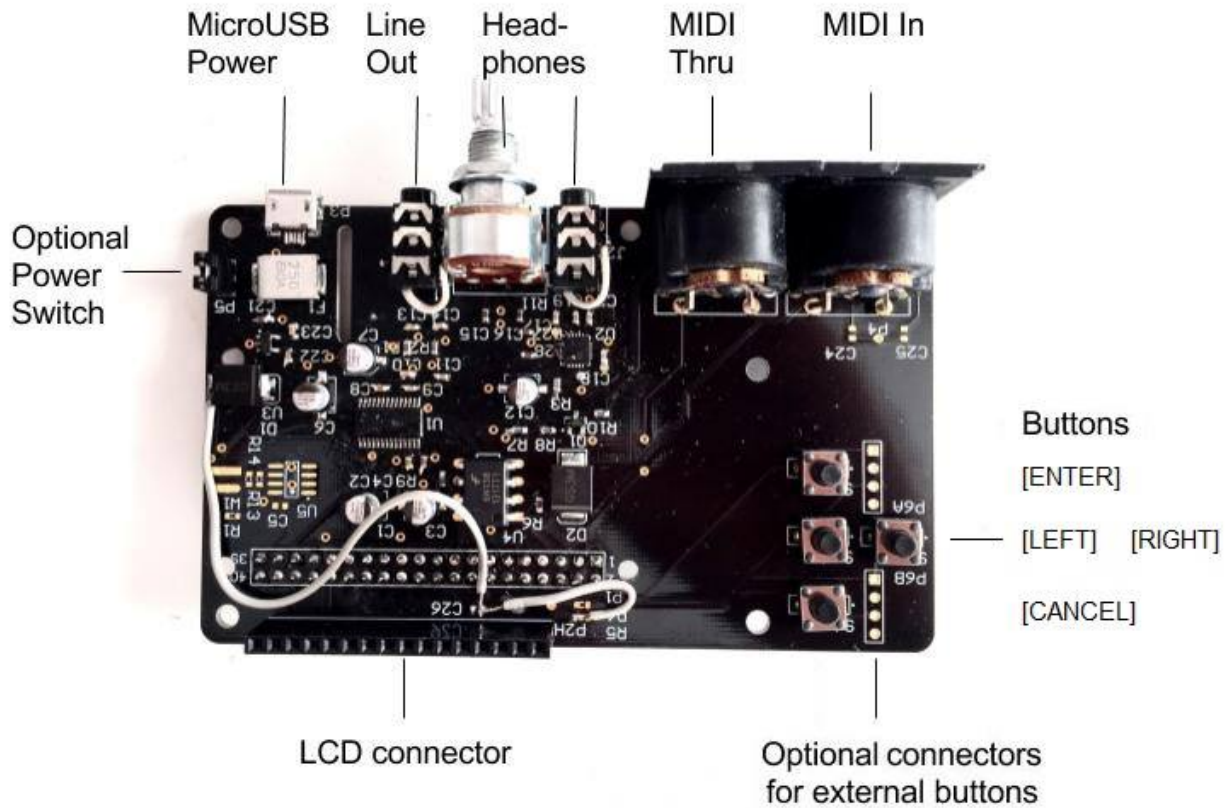
Alex MacRae

Sep 25, 2017

1	How it works	3
1.1	Features	3
1.1.1	Basic features	3
1.1.2	Two system modes	3
1.1.3	Accurate velocity	4
1.1.4	Voices	4
1.1.5	Sample randomization	4
1.1.6	Freeverb	4
1.1.7	Auto-chords	5
1.1.8	Panic key	5
1.1.9	Menu system	5
1.1.10	Setlist	5
1.1.11	Sample-set definition management	5
1.1.12	MIDI mapping	5
1.1.13	System settings	6
1.2	Hardware requirements and assembly	6
1.2.1	SamplerBox Player (beta) Assembly	6
1.2.2	Manual assembly from components	8
1.2.3	Raspberry Pi only (not recommended)	9
1.3	Installation	9
1.4	Configuration	9
1.4.1	Main configuration	9
1.4.2	System messages	10
1.4.3	GPIO pin setup for HD44780 LCD modules	10
1.4.4	System mode 1 controls	10
1.4.5	System mode 2 controls	11
1.4.6	GPIO pin setup for a 7 segment display	11
1.5	Using SamplerBox	11
1.5.1	Introduction	11
1.5.2	System mode 1	12
1.5.3	System mode 2	16
1.5.4	SamplerBox Player (beta)	19
1.6	Sample-sets	19
1.6.1	Where to put samples	19
1.6.2	Download sample-sets	20
1.6.3	Make your own samples	21

1.6.4	Make a sample-set	21
1.6.5	Definition files	22
1.7	RPi access and Network setup	29
1.7.1	Accessing Raspberry Pi	29
1.7.2	Network setup	30
1.8	Frequently asked questions	30
1.8.1	What is SamplerBox, and why this project?	31
1.8.2	What kind of MIDI keyboards can I connect to SamplerBox?	31
1.8.3	Where can I find instrument sample-sets to use with SamplerBox?	31
1.8.4	Is velocity sensitivity possible?	31
1.8.5	How to create my own sample-set to use with SamplerBox? (easy)	31
1.8.6	How to create my own sample-set to use with SamplerBox? (advanced)	32
1.8.7	Is it possible to change the MIDI channel and load two patches at once? (Let's say bass samples on channel 1 and horn samples on channel 2?)	32
1.8.8	Where should I put the sample-sets?	32
1.8.9	How to change the current preset?	33
1.8.10	What audio formats are supported?	33
1.8.11	Do I need a Raspberry Pi 2 or will it work as well with a Raspberry B / B+?	33
1.8.12	How do I put the SamplerBox image file on a microSD card?	33
1.8.13	(For developers only) Why is the filesystem mounted as read-only by default, when I use the SamplerBox image file?	33
1.8.14	What about looping? I have a sample of an organ which is 1 second long, what happens if I press the key for two seconds?	33
1.8.15	How to permanently change the sound volume?	33
1.8.16	The audio output quality is bad. Why, and how to solve it?	34
1.8.17	When I boot the Raspberry Pi with the SamplerBox image, the software doesn't start automatically. How to solve this?	34
1.8.18	Do I really need to build the whole thing (electronic parts, etc.) to use SamplerBox?	34
1.8.19	Why is it impossible to edit the samples directly on SamplerBox? Why not add a screen, a graphical user interface and editing features on SamplerBox, like on an Akai MPC?	34
1.9	Support	34
1.10	License	34

SamplerBox is a pocket-sized sound module based on [Raspberry Pi](#) for use with MIDI devices in your bedroom, studio, or on the stage. It is completely open-source which means it is forever evolving, making it a formidable alternative to other commercial MIDI samplers and expanders on the market.



SamplerBox Player PCB (without LCD)

LCD loading and initial sample-set display. Line 1 = current sample-set, Line 2 = next sample-set

How it works

SamplerBox uses MIDI messages from a connected MIDI keyboard or device to trigger digital audio samples (wave files) which are then converted to an analogue audio signal. The software has been designed to run on [Raspberry Pi](#) single-board computers. A single-board computer is a tiny computer that contains a CPU processor, GPU, ROM, RAM, Ethernet, USB and HDMI ports. They can be programmed to do almost anything: run an operating system, operate a security system, navigate aircraft, take over the world. In the case of SamplerBox its sole purpose is to play music!

Once SamplerBox is running, it will find sample-set directories on a connected USB drive and add them to a setlist. SamplerBox will then attempt to load sample files based on a file naming convention or filenames and rules defined in a *definition.txt* file.

Contributors

Joseph Ernest	twitter: @JosephErnest mail: contact@samplerbox.org
Alex MacRae	web: GitHub mail: alex@samplerbox.org
Pavel Titov	
Hans Hommersom	web: http://homspace.xs4all.nl/homspace/samplerbox/index.html
Erik	web: http://www.nickyspride.nl/sb2/

Features

Basic features

SamplerBox allows for the manipulation of some basic features such as preset changing, sample looping, velocity sensitivity, volume, sustain, pitch bend, transpose, playback mode, and voices.

Two system modes

SamplerBox has two system modes available which is determined by the `SYSTEM_MODE` option in *config.ini*.

System mode 1 (by Alex MacRae) has more advanced features such as setlist mode, ability to modify system settings, MIDI mapping, and more. *System mode 2* (by Hans Hommersom) is a simpler and easier system to use but with less features.

This section requires that you have an understanding of how *definition.txt files* work

Accurate velocity

Examples [here](#).

Voices

You can now define up to 4 voices within presets via a sample-set's `definition.txt` file using the parameter `%voice`.

For example, `piano_60_v1.wav` and `juno_60_v2.wav` can be found like this:

```
piano_%midinote_v%voice.wav
juno_%midinote_v%voice.wav
```

You can also write something like this:

```
piano_%midinote.wav_v1, %voice=1
juno_%midinote.wav_v2, %voice=2
```

More examples [here](#).

Note: To switch between voices you will need to use the *MIDI Mapping* function to map your desired device controls.

Sample randomization

If you have multiple versions of the same sample, for example alternative samples of the same snare drum, you can tell SamplerBox to randomize them by using the `%seq` keyword in your *definition.txt file*.

More examples [here](#).

Freeverb

A reverb module (*Freeverb*) is available. Thanks goes to [Erik](#) for this feature.

Reverb can be enabled in `config.ini` either via the text file directly in `/boot/samplerbox/config.ini` or via *System mode 1*'s menu under *System Settings*.

Note: You must map its parameters to some MIDI controls via *MIDI Mapping*.

Warning: This is currently experimental as it sometimes produces undesirable pops and clicks.

Auto-chords

Builds and plays back a chord from a single note.

Panic key

Control change messages CC120 and CC123 will trigger the panic function, killing all playing sounds. This can also be *MIDI Mapped*.

Menu system

A menu system is available to change the behaviour of the SamplerBox.

Note: This feature assumes you have a *HD44780 LCD* module connected to your Raspberry Pi. You will need to define the GPIO pins you are connect to in the *config.ini*.

Warning: *System mode 1* only

Setlist

SamplerBox manages your sample-sets with a *setlist.txt* file found in the root of your sample-sets directory. If one cannot be found, a new one is generated and populated with all sample-sets ordered alphanumerically.

Upon startup, new folders will be detected and appended to the end of the setlist.

You can rearrange the setlist via the menu system.

Warning: *System mode 1* only. For *System mode 2* sample-sets will be ordered by directory names alphanumerically.

Sample-set definition management

Global behaviour keywords for every sample-set can be modified using this feature. This includes *%%mode*, *%%velmode*, *%%release*, *%%gain*, *%%transpose*, *fillnotes* and *%%pitchbend*.

Warning: *System mode 1* only

MIDI mapping

Many features of the SamplerBox can be mapped to *MIDI controls* via the menu system.

Warning: *System mode 1* only

System settings

Some of the system settings found in the *config.ini* file can be edited and saved from this menu.

More in the *System Settings* section.

Warning: *System mode 1* only

Hardware requirements and assembly

SamplerBox Player (beta) Assembly

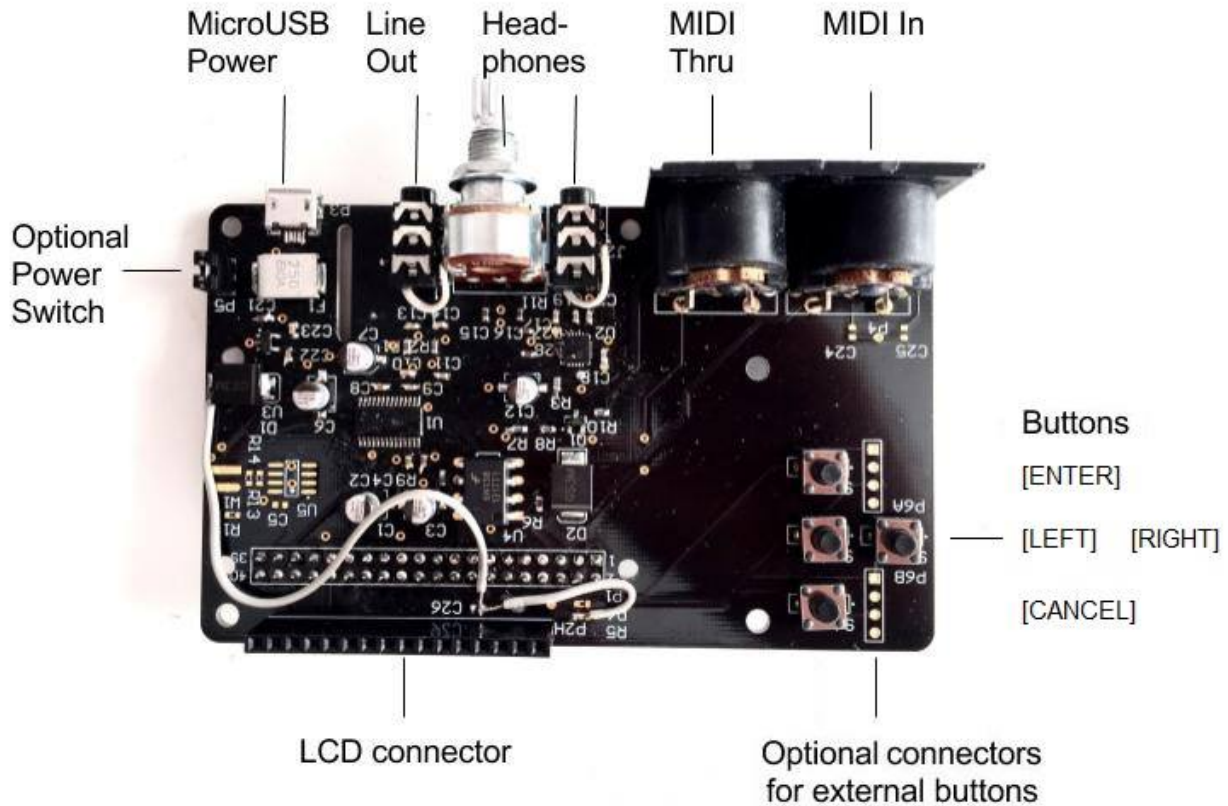
What you will need

To run the SamplerBox software it is essential that you have a Raspberry Pi. Version 2+ is recommended. This section also assumes you have purchased a **SamplerBox Player (beta)** from the [SamplerBox shop](#) (pending stock).

Here's everything you will need to assemble the :

- SamplerBox Player (beta) kit
 - SamplerBox Player PCB
 - Attached LCD display
 - Raspberry Pi mounting kit
- Raspberry Pi. Minimal version 1B+ or higher; recommended 2 or higher
- Micro SD card, 2GB or larger
- MicroUSB power cord
- USB power adapter from reliable brand rated 2A or more
- USB stick for samples

Getting Started



SamplerBox Player PCB shown without LCD.

1. Carefully unpack SamplerBox Kit and Raspberry Pi (not included).

Warning: Ensure you are static-free.

2. Screw four enclosed stand-offs into SamplerBox PCB from the bottom in locations matching Raspberry Pi mounting holes.



3. Align 40-pin connector and stand-offs with Raspberry Pi. Push PCBs together, applying force near standoffs, applying pressure to pin headers without bending the boards.
4. Secure Raspberry Pi to SamplerBox standoffs using enclosed M3 screws.



5. Insert SD card with image and USB stick with samples into Raspberry Pi.
6. Use a standard stereo cable or headphones to connect to the balanced audio output.

Warning: Connecting both line out and headphones simultaneously is not recommended and may result in degraded sound quality.

7. Connect USB MIDI to Raspberry Pi USB port or serial MIDI to MIDI IN DIN port.
8. Connect power adapter to Micro USB port on SamplerBox. SamplerBox will power Raspberry Pi via GPIO connector.

Warning: Keep MicroUSB port on Raspberry Pi unconnected.

Manual assembly from components

This section is dedicated to building a SamplerBox from scratch.

What you will need

To run the SamplerBox software it is essential that you have a Raspberry Pi. Version 2+ is recommended.

Here's everything you will need to build a SamplerBox:

- 1x LCD module. A 16x2 character HD44780 LCD module is recommended
- 4x push buttons
- 1x USB DAC. A cheaper option is a PCM2704-based device
- 1x Micro SD card, 2GB or larger
- 1-2x MIDI DIN connectors (optional)
- 1x USB stick for samples
- 1x MIDI controller
- 1x MicroUSB power cord
- 1x USB power adapter from reliable brand rated 2A or more

Note: The Raspberry Pi's on-board audio will work, but the quality and latency is inferior to external alternatives.

Getting started

Build instructions still to come...

Raspberry Pi only (not recommended)

It is possible to run SamplerBox with no hardware other than an essential Raspberry Pi and SD card, however you will have no visual feedback and poor audio quality.

If your MIDI keyboard controller has buttons that send MIDI messages you can assign them to the preset navigation functions in the *configuration file*.

It is also possible to add sample-sets to a 3rd partition on your SD card, thus removing the need for a USB stick. *More information here*.

Installation

1. Download the latest image from the [SamplerBox website](#)
2. Write the image to a formatted micro SD card from your PC. Operating system-specific instructions:
 - [Linux](#)
 - [Mac OS](#)
 - [Windows](#)
3. (Optional) If you're not using the official SamplerBox Player hardware, access the SD card on your PC and edit the *config.ini* found in `/boot/samplerbox/` to match your hardware setup. Default `config.ini` settings are configured for the official hardware.
4. Insert your micro SD card, USB stick with sample-sets, and MIDI keyboard into your Raspberry Pi, and start making some music!

Configuration

The *config.ini* file contains settings for your SamplerBox. It is found in the `/boot/samplerbox` directory. The `/boot` partition is also accessible via a Windows or Mac machine.

Note: If you have purchased a SamplerBox Player Pi HAT, the default configuration settings will work out of the box.

Note: In *System mode 1's system settings* many of these settings are configurable from the menu system. However an initial setup of this file may be required.

Main configuration

```
MAX_POLYPHONY = 40
MIDI_CHANNEL = 1
SAMPLERATE = 44100
```

```
GLOBAL_VOLUME = 100
USE_FREEVERB = False
USE_I2C_7SEGMENTDISPLAY = False
USE_SERIALPORT_MIDI = False
USE_TONECONTROL = False
USE_HD44780_16X2_LCD = True
USE_HD44780_20X4_LCD = False
USE_BUTTONS = True
USE_GUI = False
SAMPLES_DIR = None
AUDIO_DEVICE_ID = -1
AUDIO_DEVICE_NAME = autodetect
BOXRELEASE = 30
PRESET_BASE = 0
SYSTEM_MODE = 1
RAM_LIMIT_PERCENTAGE = 40
INVERT_SUSTAIN = False
```

Note: USE_FREEVERB may cause pops and clicks.

RAM_LIMIT_PERCENTAGE determines how much RAM can be used for loading samples. This allows for preloading of sample-sets following the current one and, depending on the size of the library, seamless preset navigation.

System messages

Useful for debugging issues (when connected to a screen or via SSH) and seeing what MIDI messages are being sent by a MIDI device.

```
PRINT_MIDI_MESSAGES = True
PRINT_LCD_MESSAGES = True
```

GPIO pin setup for HD44780 LCD modules

If you're using a HD44780 LCD module (16x2 or 20x4) you must define the numbers of the GPIO pins they are connected to here.

```
GPIO_LCD_RS = 7
GPIO_LCD_E = 8
GPIO_LCD_D7 = 4
GPIO_LCD_D6 = 18
GPIO_LCD_D5 = 17
GPIO_LCD_D4 = 27
```

System mode 1 controls

Controls for controlling and navigating the SamplerBox when in *System mode 1*.

If MIDI controls and/or GPIO pins connected to buttons are known, you may define them here.

Note: When `PRINT_MIDI_MESSAGES = true`, SamplerBox will return MIDI messages in the format required below. This only needs to be done once.

```
BUTTON_LEFT_MIDI = 176, 48, <MIDI CONTROLLER NAME>
BUTTON_RIGHT_MIDI = 176, 50, <MIDI CONTROLLER NAME>
BUTTON_ENTER_MIDI = 176, 49, <MIDI CONTROLLER NAME>
BUTTON_CANCEL_MIDI = 176, 65, <MIDI CONTROLLER NAME>

BUTTON_LEFT_GPIO = 26
BUTTON_RIGHT_GPIO = 13
BUTTON_ENTER_GPIO = 6
BUTTON_CANCEL_GPIO = 12
```

System mode 2 controls

Controls for controlling and navigating the SamplerBox when in *System mode 2*.

If MIDI controls and/or GPIO pins connected to buttons are known, you may define them here.

Note: When `PRINT_MIDI_MESSAGES = true`, SamplerBox will return MIDI messages in the format required below. This only needs to be done once.

```
BUTTON_UP_MIDI = 176, 50, <MIDI CONTROLLER NAME>
BUTTON_DOWN_MIDI = 176, 48, <MIDI CONTROLLER NAME>
BUTTON_FUNC_MIDI = 176, 49, <MIDI CONTROLLER NAME>

BUTTON_UP_GPIO = 13
BUTTON_DOWN_GPIO = 26
BUTTON_FUNC_GPIO = 6
```

GPIO pin setup for a 7 segment display

If you're using a 7 segment display you must define the number of the GPIO pin it is connected to here. This is not recommended as navigating the system menus with feedback is (near) impossible.

```
GPIO_7SEG = 1
```

Using SamplerBox

Introduction

SamplerBox has two system modes available which is determined by the `SYSTEM_MODE` option in *config.ini*.

<p><i>System mode 1</i></p>	<p>config.ini: SYSTEM_MODE = 1</p> <p>A more advanced system that allows the user to:</p> <ul style="list-style-type: none"> • manage the order of their sample-sets (setlist mode) • manage variables defined in a sample-set's definition.txt • map MIDI controls to various playback and system functions • manage system settings by modifying the config.ini
<p><i>System mode 2</i></p>	<p>config.ini: SYSTEM_MODE = 2</p> <p>A simpler system that relies on the user preparing sample-sets and config.ini on a computer. Some functions available.</p>

System mode 1



Displaying current preset '3 Mello+Sawtooth' which has 2 voices available. The next preset is '4 Korg Little Bits'. The 'S' indicates samples are still loading in the background

A menu system has been implemented to access:

- *Setlist functions*
- *Edit Definitions*
- *Auto Chords*

- *MIDI Mapping*
- *System Settings*
- *Invert Sustain*
- *Reboot System*

In the menu system

Warning: This feature assumes you have a [HD44780 LCD](#) module wired to your Raspberry Pi, or you have purchased an official SamplerBox Player. You may need to manually define the GPIO pins it is connected to in your *config.ini* file.

Setlist functions

SamplerBox can manage your sample-sets by using a setlist. On startup new folders will be detected and appended to the end of the setlist. Using the menu system you can rearrange your sample-sets.

Edit Definitions

You can manage the global behaviour keywords of a sample-set's *definition.txt* file from the menu system. These include `%%gain`, `%%mode`, `%%velmode`, `%%release`, `%%transpose`, `%%pitchbend`, and `%%fillnotes`. More information about global behaviour keywords can be found [here](#).

Auto Chords

SamplerBox has the ability to play chords types based on the notes you play and (optionally) in a specified key.

Chord Mode

Mode	Description
MAJ scale chords	Will play chords based on a specified major key.
MIN scale chords	Will play chords based on a specified minor key.
All MAJ chords	All notes will trigger a major chord based on its root note.
All MIN chords	All notes will trigger a minor chord based on its root note.

Root Key

Allows you to select a key to base your chords on. Only works with chord modes MAJ scale chords and `MIN scale chords.

MIDI Mapping

Function to map	Description
Master volume	Map any control, ideally a fader or pot, to affect the SamplerBox's master volume.
Voices	Map any control to each of the 4 voices.
Pitch bend	Map any control to the pitch bending function. Useful if your keyboard doesn't have a pitch wheel.
Sustain	Map any control to the pedal sustain function. Useful if your keyboard doesn't have a sustain pedal input.
Panic key	Map a key to kill all sounds.
SamplerBox Navigation	Map MIDI controls to each of the 4 navigation buttons (left, right, enter and cancel). Mapping a control to one of these functions will not override any other controls mapped to the same function, thus allowing multiple mappings.
Reverb	Map any control, ideally a potentiometer, to any of the 5 reverb parameters. Room size, damp, wet, dry, and width.

System Settings

Some system settings can be modified from this menu. Changing these options will save their values to the *config.ini* and be read again upon a restart.

Option	Description
Audio device	Select default audio device to use every time (if it can be found).
Max polyphony	Range: 1-128. The maximum number of samples that can be played simultaneously.
MIDI channel	Range: 0-16. 0 = all channels.
Sample rate	Options: 44100, 48000. Choose a sample rate compatible with your audio device and sample-sets. 44100 is usually safe.
Reverb	Experimental. Options: ON/OFF (requires reboot).
Set RAM limit	Set the amount of RAM the system can use for samples. For example, a setting of 95% will see SamplerBox attempt to load as many presets into memory. A setting of 5% will likely only allow for one preset at a time to be loaded into memory.

Invert Sustain

Invert the polarity of your sustain pedal if it is functioning in the opposite way of what is expected.

Reboot System

Reboot the system from the menu – safer than powering off and on again.

System mode 2

This system mode was developed in a way that assumes the user has prepared their sample-sets and configured the *config.ini* file correctly on their PC before usage with a SamplerBox.

Note: This system mode was developed by Hans Hommersom. Some more detailed information and instructions can be found [on his site here](#).

Warning: This feature assumes you have a [HD44780 LCD](#) module wired to your Raspberry Pi. You will need to manually define the GPIO pins it is connected to in the *config.ini* file.

Setup

Three buttons (up, down and function) must be defined in the *config.ini*. They may be GPIO buttons and/or MIDI controls.

GPIO buttons. Red=+ Black=function Blue=-

Usage

Physical buttons for manual control

The box has three buttons: a function select button and two +/- buttons. The select button walks through the available functions with the second display line showing this until a MIDI message needing the display is received. Then the underlined letter will show on the right of the second line preceded by an asterisk (*).

Implemented functions (some can be disabled via the local config parameters):

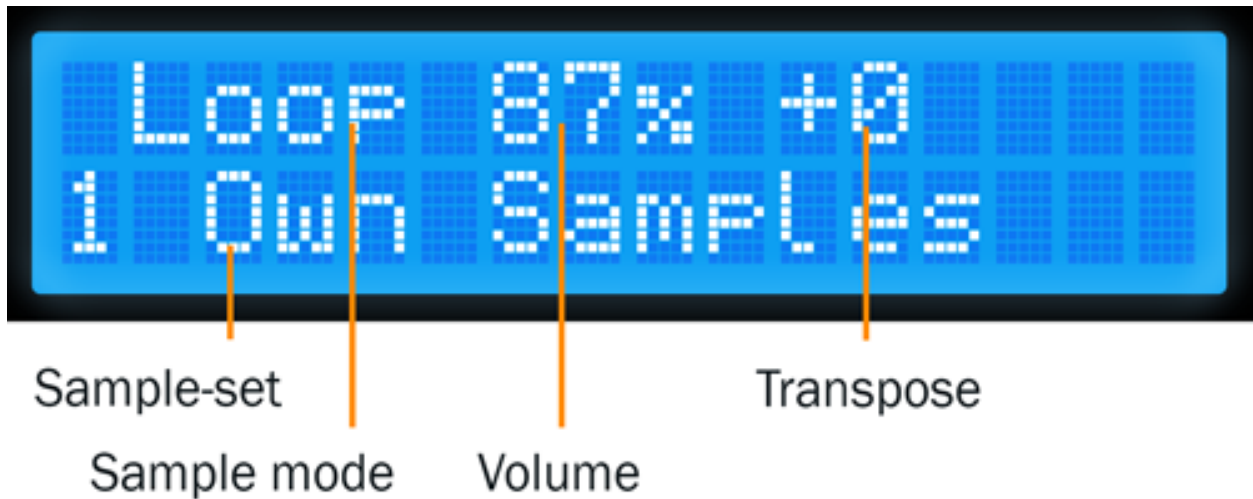
Function	Description
Program/patch/preset	<p>Range = 1-128</p> <p>This is the default and the second display line will just show the actual selected patch.</p>
Volume	<p>Range = 0-100 (%)</p> <p>This controls the ALSA mixer = output volume. The actual volume is shown as a percentage on the first display line. Boot-up volume is set in the script.</p>
MIDI channel	<p>Range = 1-16</p> <p>Changes the channel listened to, default = 11.</p>
Transpose	<p>Range = -99 to +99</p> <p>Changes the pitch/speed (thus also the length of any loops) in semitones.</p> <p>The actual transpose value is shown in the last value on the first display line.</p>
Specials	<p>For reprogramming presets and controllers:</p> <p>+ Button = Renew USB will force reload of the current patch.</p> <p>- Button = Midi mute stops the box from listening to MIDI signals.</p> <p style="padding-left: 40px;">Pressing this or any other key will unmute.</p>
Chords	<p>Empty or chord name</p> <p>Empty is single note (“normal play”), others are defined in Chords definitions in the local config. Currently 15 chords are implemented.</p>

Control via the MIDI controller

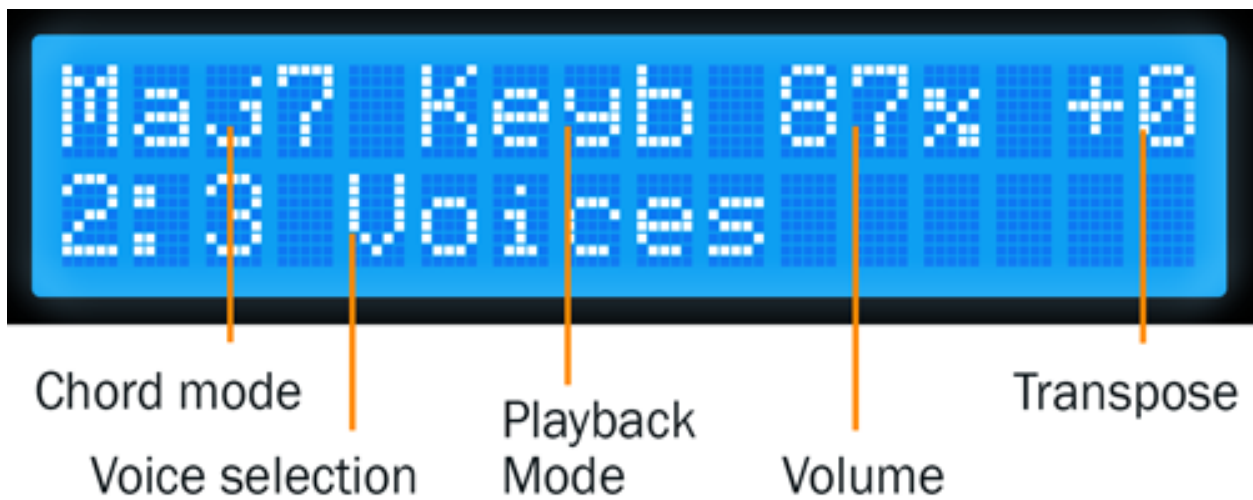
Your midi controller device may be capable of sending midi controls via buttons, levers or wheels. These may be customizable, otherwise you may have to adapt the SamplerBox script to fit to your controller.

This SamplerBox is configured to recognize next [control change messages](#):

MIDI message	Description
Message 12	<p>Program change Changes preset = sample folder. In the default implementation the MIDI values (0-15) are translated to human program numbers (1-12). Depending on your controller's implementation you can change this with the local config to get it aligned.</p>
Message 14	<p>Pitch bend (wheel / joystick / knob) Pitch bend depth can be configured from 1 to 12 semitones via the local config globally and per sample set via the definition.txt. Future enhancement include an extra continuous controller for this as well.</p>
Message 11	<p>Continuous controller messages:</p> <ul style="list-style-type: none"> • Controller 7: Volume slider/knob. The LSB (controller 39) is ignored. • Controller 64: Sustain pedal • Controller 80: General purpose, now used for selecting voices • Controller 81: General purpose, now used for selecting chords • Controller 120/123: "All sounds off"/"all notes off" • Both will stop playing sounds/samples and pending sustains. • Panic button usually sends such messages.



Loaded loop set without selecting further specials



Selected a major-7 chord and voice#2 in the set "3 Voices"

SamplerBox Player (beta)

[Information about the official PCB]

Sample-sets

There are many free sample-sets available for download on the [SamplerBox website](#). You may also use SFZ (soundfont) packages or even create your own sets!

Where to put samples

SamplerBox looks for sample-set directories in three places, in this order of priority:

1. User-defined directory in `config.ini`
2. `/media/` (USB drive)
3. `/samples/` (SD card)

Note: If the user-defined directory cannot be found, SamplerBox will look for a mounted USB drive. Failing that, the default `/samples/` directory will be used.

USB drive

This is the easiest way to get samples from your computer to your SamplerBox. By default, SamplerBox will look for directories in the root of your USB drive. A `setlist.txt` file will be automatically generated that will include the name of every directory listed alphanumerically. You can reorder the lines in the `setlist.txt` file on your PC, or do it via the *System mode 1* menu.

Hint: If you'd prefer to have your sample-sets in a subdirectory on your USB drive (or indeed anywhere else) find the line `SAMPLES_DIR = None` in your `config.ini` and change it to `SAMPLES_DIR = /media/subdirectory`.

SD card (advanced)

A third partition on your SD card is available for samples (mounted to `/samples/`). When the Raspberry Pi is running it is mounted as read-only to extend the life of the SD card.

To manage your sample-sets here you must first remount the partition as read-write by entering the following on the Raspberry Pi's command line:

```
mount -o remount,rw /samples
```

You can now upload/delete sample-sets via your SFTP client.

It is also possible to manage samples on your PC computer. However, Windows machines only detect the first partition of an SD card. In the case of SamplerBox this is the `/boot/` partition. There are ways to gain access to other partitions but that is not covered here.

Note: SamplerBox issues mounting commands from the program. This is how it can manage sample-sets and the setlist from the within the program.

Download sample-sets

There are a number of free instruments available for download on the [SamplerBox website](#).

SFZ SoundFont packages currently have limited support, but they should work. There are many free downloads available on the web.

Make your own samples

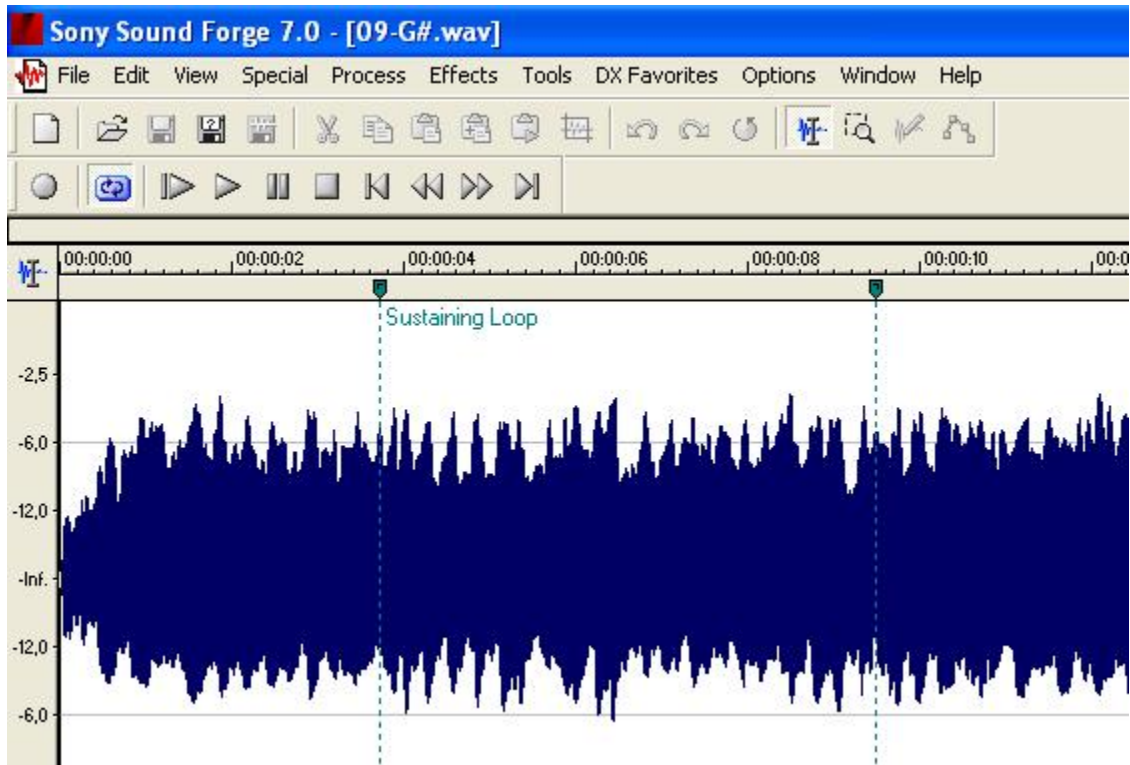
Samples

SamplerBox currently only reads WAV files. Fortunately every modern audio editor can work with them.

Use your favourite audio editor to cut up your instrument into note samples.

Loop markers

To save loop markers to WAV files you must use a sound editor that is capable of this, eg [Sony Soundforge](#). SamplerBox will recognise loop markers and loop your sounds!



A premium alternative is [Extreme Sample Converter](#) which can auto-detect and save suitable loop markers.

Make a sample-set

Once you have a collection of samples you must first put them in a directory into *one of the three directories listed above* so that SamplerBox can find them.

Now you need to prepare the directory in a way that SamplerBox can read the samples. There are 4 ways to do this:

1. Name every sample file by its note name. eg C1.wav, F#1.wav, C2.wav, F#2.wav etc, or
2. Name every sample file by its corresponding MIDI number. eg 40.wav, 45.wav, 50.wav, 55.wav etc, or
3. Create a text file named `definition.txt` in your sample-set's directory. *Instructions below*, or

4. Use a sample-set with a accompanying SFZ definition file, or create your own if you prefer (instructions not covered here).

Note: SamplerBox will attempt to fill all missing notes.

Warning: Not all options found in SFZ files will be read by SamplerBox. eg attack, decay, EQ.

Definition files

If you're not naming your samples by the convention `45.wav`, `46.wav` etc, or `C1.wav`, `C#1.wav` etc, you can create a `definition.txt` file inside a sample-set's directory. This file tells SamplerBox how to read and playback samples.

Definition of sample names

In the most basic situation, the sample files within the folders have to be called `0.wav`, `2.wav` and so on until `127.wav`.

A more advanced way to manage sample-sets is to use a `definition.txt` which uses filename definitions and keywords to determine how SamplerBox finds and performs samples.

Global behaviour keywords

These are global keywords in the `definition.txt` for influencing the playback upon load of a preset/sample-set. For every keyword not defined in the file, the default value is used.

Note: In *System mode 1* it is possible to modify these keywords from menu.

%%mode

%%mode=	Description
Keyb	(Default) “Normal”: end on note-off and use loop markers if any while key is pressed (original SamplerBox).
Once	“Playback”: play sample from start to end ignoring standard note-off.
On64	Like “once” but now only notes 0-63 can be used; use note+64 to stop playback (=send note-off)
Loop	Like “on64”, but also loop markers will be recognized; more versatile than “On64”
Loo2	Like “loop”, but the loop will stop when playing the same note (=2nd keypress sends note-off). If the sample has no loop markers it will stop when exhausted, but pressing the key a second time is still required before the sample can be played again! This mode mimicks Korg-KAOSS and some groove samplers.

%%velmode

The way that volume is derived from the velocity.

%%velmode=	Description
Sample	(Default) Volume equals the value in the sample, so it requires multiple samples using the %velocity parameter to get differentiation.
Accurate	Played velocity influences volume directly. Multiple samples can still be used to differentiate timbre, but now they must be normalized.

%%release

Time to fadeout playback volume from the sample level to zero after the key is released in tenth's of seconds.

```
Default = 30 (~0.5s)
Allowed range: 0-127 (range of 0-2s)
```

%%gain

Adapts sample volume before alsamixer by means of a multiplication factor. With this you can adapt presets to SamplerBox input without actually changing the wav files.

```
Default = 1.0
Allowed range: 0.1-10.0
```

Warning: Setting this value too high may cause distorted playback.

%%transpose

Transpose up or down a desired number of semitones.

```
Default = 0
```

%%pitchbend

The depth of the pitchbend in semitones.

A value of 12 means the pitch range is 1 octave up and down. Zero will disable the pitch wheel/joystick.

```
Default = 7
Allowed values: 0-24
```

%%fillnotes

Determines whether SamplerBox should attempt to fill notes that haven't been assigned a sample.

For instance, you might have 2 samples in a directory: 40.wav and 60.wav. If %%fillnotes=Y, the program will attempt to fill all missing notes (0-127) with the appropriate sample. Otherwise if %%fillnotes=N only midinotes 40 and 60 will be assigned a sample. All other keys will play nothing.

```
Default = Y
Allowed values: Y,N
```

Sample-level behaviour keywords

In addition to global keywords, sample-level keywords can be used. Some of these override global keywords. For example:

```
saw2.wav, %midinote=60, %voice=2, %fillnote=N
```

%notename

```
Values = C1, C2, C3, D#3, F#4, etc.
```

Define a sample's MIDI note by its note name.

%midinote

```
Value range = 0-127
```

Define a sample's MIDI note. For example 60 corresponds to middle C = C4.

%channel

```
Value range = 0-16
```

```
Default = 0
```

NB: 0 = all channels.

%velocity

```
Value range = 1-127
```

```
Default = 127
```

A velocity sample is used from its value upwards till the next sample. Velocity values below lowest sample will use this lowest one.

%voice

```
Value range = 1-4
```

```
Default = 1
```

This enables loading different instruments in one sample set, so that switching between them has no delay.

%seq

```
Value range = 1-127
```

```
Default = 1
```

If you have multiple versions of the same sample (eg different snare samples) you can number them. On playback a random sample will be selected.

%fillnote

```
Values = Y, N, G
```

```
Default = G (use global setting)
```

Determines whether the sample at the specified note will fill surrounding notes.

%mode

```
Values = Once
```

```
Default = None
```

Currently only accepts Once. See %%mode above for its functionality.

%mutegroup

```
Value range = 0-127
```

```
Default = 0 (no mute group)
```

Assign sample lines to a mutegroup. When performing a note from a mutegroup, any other sounds from the same group will be stopped/choked. Useful for instruments like hi-hats.

Examples

1. Velocity

The original [GrandPiano](#) set uses multiple lines specifying the wav's to be selected and the corresponding fixed velocity value. Remember that default velocity is 127.

File names:

```
D#5v16.wav  
D#6v4.wav  
D#6v7.wav  
D#6v11.wav
```

```
D#6v14.wav
D#6v16.wav
D#7v4.wav
```

definition.txt:

```
%%mode=Keyb
%%velmode=Sample
%notenamev4.wav,%velocity=40
%notenamev7.wav,%velocity=60
%notenamev11.wav,%velocity=80
%notenamev14.wav,%velocity=100
%notenamev16.wav
```

Alternatively, you can set the global velocity mode (%%velmode) to either *Accurate* (default) or *Sample*.

Accurate will scale the volume of a sample based on the velocity of the note played.

Sample will look for velocity values in the `definition.txt` (as in the example above).

```
%%mode=Keyb
%%velmode=Accurate
%notenamev14.wav
%notenamev16.wav
```

2. Naming and looping

I often use this definition set, which makes it possible to give the loops and fills a self explaining name. Directory on the left is interpreted correctly.

File names:

```
03 Alesis-Fusion-Bass-Loop.wav
4 takkeherrie.wav
6.wav
11 Carol.wav
20 130-bpm-electro-synth-loop.wav
21 Aggressive-saw-synth-bass-loop.wav
```

definition.txt:

```
%%mode=Loop
%%velmode=Accurate
%midinote*.wav
```

3. Voices

This is set 3 on the SDCard with voices. It uses actually one velocity range of the GrandPiano combined with Saw. The saw WAV's are renamed with midinumber prefixed with notename plus an "m".

File names:

```
A5v12.wav
A6v12.wav
A7v12.wav
```

```
C1v12.wav
c2m36.wav
C2v12.wav
c3m48.wav
C3v12.wav
```

definition.txt:

```
%%mode=Keyb
%%release=3
%%velmode=Accurate
%notenamev*.wav
%notenamem*.wav,%voice=2
```

Hint: You can also define the voice in the file name. eg `c3_voice2.wav` will be found with `%notename_voice%voice*.wav` in the definition.txt

4. Randomization

You might have a sample-set with a variation of samples of the same instrument note. In this example there are 4 samples/recordings of a kick drum and a snare drum. The `%seq` keyword tells SamplerBox to play back a different version of the kick or snare for every hit, thus giving a more realistic performance.

File names:

```
bonham-kick-1.wav
bonham-kick-2.wav
bonham-kick-3.wav
bonham-kick-4.wav
bonham-snare-1.wav
bonham-snare-2.wav
bonham-snare-3.wav
bonham-snare-4.wav
```

definition.txt:

```
%%mode=Once
%%velmode=Accurate
bonham-kick-%seq.wav
bonham-snare-%seq.wav
```

Note: SamplerBox may randomize samples in an order like this: 3-6-2-1-8-7-2. Notice how there are no consecutive repeats.

5. Playback mode

In this example the global `%%mode` is `Keyb` (normal playback), but 1 sample plays back from start to end ignoring the standard note-off.

File names:


```
mellotron C1.wav
mellotron E1.wav
mellotron G1.wav
aeroplane-sample.wav
```

definition.txt:

```
%%mode=Keyb
%%velmode=Accurate
mellotron %notename.wav
aeroplane-sample.wav, %mode=Once
```

RPI access and Network setup

Since SamplerBox runs on Raspberry Pi, you can gain access to the internet via the on-board Ethernet or WiFi (Raspberry Pi 3).

Accessing Raspberry Pi

Via HDMI monitor and keyboard

You can gain access to the Raspberry Pi's commandline by connecting to a monitor using a HDMI cable and USB keyboard.

Login is: `root/root`

Via SSH

This section assumes you have already successfully *set up your network*.

You can gain access to the Raspberry Pi's commandline using an SSH client. Have a look at [this comparison of SSH clients](#) for some options.

Login is: `root/root`

Via SFTP

This section assumes you have already successfully *set up your network*.

Using your favourite FTP client, eg [FileZilla](#) you can access your SD card storage directly from your PC. You will need to know your Pi's IP address. You can find this using the `ifconfig` command on the commandline, or your router's admin interface may be able to tell you.

Host	<YOUR PI'S IP ADDRESS>
Protocol	SFTP
Logon Type	Normal
User	root
Password	root

Warning: File-systems are mounted as read-only. To modify files you will need to remount as read-write by issuing one of these commands (depending on what you want to modify): `mount -o remount,rw /`, `mount -o remount,rw /`boot``, ```mount -o remount,rw /samples`

Network setup

Ethernet configuration

WiFi configuration

If you are using a Raspberry Pi 3 you can configure its on-board WiFi. For other RPi versions a USB WiFi dongle can be used, however they may require extra configuration.

1. Access your Pi via a *HDMI monitor and keyboard* or *SSH* and log in: Username: `root` Password: `root`
2. Remount the root partition as read-write: `mount -o remount,rw /`
3. Open the wireless networks configuration file: `nano /etc/wpa_supplicant/wpa_supplicant.conf`
4. Uncomment the `network={...}` section and add your network SSID and password (psk).
5. Save and exit: CTRL-X and Y
6. Reboot: `reboot`
7. Check to see if wlan0 has acquired an IP address: `ifconfig wlan0` (there should be a line starting with 'inet addr')
8. Done!

Hint: If your router supports static addresses, assign one to your Raspberry Pi so that it acquires the same IP address every time you power on!

Frequently asked questions

- *What is SamplerBox, and why this project?*
- *What kind of MIDI keyboards can I connect to SamplerBox?*
- *Where can I find instrument sample-sets to use with SamplerBox?*
- *Is velocity sensitivity possible?*
- *How to create my own sample-set to use with SamplerBox? (easy)*
- *How to create my own sample-set to use with SamplerBox? (advanced)*
- *Is it possible to change the MIDI channel and load two patches at once? (Let's say bass samples on channel 1 and horn samples on channel 2?)*
- *Where should I put the sample-sets?*

- *How to change the current preset?*
- *What audio formats are supported?*
- *Do I need a Raspberry Pi 2 or will it work as well with a Raspberry B / B+?*
- *How do I put the SamplerBox image file on a microSD card?*
- *(For developers only) Why is the filesystem mounted as read-only by default, when I use the SamplerBox image file?*
- *What about looping? I have a sample of an organ which is 1 second long, what happens if I press the key for two seconds?*
- *How to permanently change the sound volume?*
- *The audio output quality is bad. Why, and how to solve it?*
- *When I boot the Raspberry Pi with the SamplerBox image, the software doesn't start automatically. How to solve this?*
- *Do I really need to build the whole thing (electronic parts, etc.) to use SamplerBox?*
- *Why is it impossible to edit the samples directly on SamplerBox? Why not add a screen, a graphical user interface and editing features on SamplerBox, like on an Akai MPC?*

What is SamplerBox, and why this project?

SamplerBox is an electronic musical instrument. Drop audio samples onto it, hook up a MIDI keyboard, and you'll be able to play with realistic piano, organ, drums, etc. sounds! As strange as it may sound, if you wanted to add great piano sound to your MIDI keyboard or synthesizer, there was previously no hardware solution except using a computer (ok it works, but sometimes you don't feel like using a computer to do music) or buying an expensive sampler / expander. Now SamplerBox provides a sub-99€ solution!

What kind of MIDI keyboads can I connect to SamplerBox?

There is no limitation about MIDI keyboards. Both USB MIDI keyboards (with a USB plug) and MIDI (with a MIDI plug) are accepted. You can use small 25-keys keyboards, such as the Akai LPK25 we use in some videos ; you can use 49 keys MIDI keyboards, or even 61 keys or 88 keys if you want!

Where can I find instrument sample-sets to use with SamplerBox?

You can find [some instruments here](#). But you can also do some sample-sets yourself in a few seconds only (no sample-set-making skills required)! See questions *How to create my own sample-set to use with SamplerBox? (easy)* and *How to create my own sample-set to use with SamplerBox? (advanced)*.

Is velocity sensitivity possible?

Yes. See *How to create my own sample-set to use with SamplerBox? (advanced)*.

How to create my own sample-set to use with SamplerBox? (easy)

If your samples are numbered by their usual MIDI note like this: 36.wav, 37.wav, ..., you just need to put these files in a folder named like this: /1 Piano/, /17 Trumpet/, ..., i.e. a number + a white space + a name. The

number will be the preset number.

No sample-set definition file required in this simple case!

How to create my own sample-set to use with SamplerBox? (advanced)

Sometimes, a picture speaks more than words, so don't forget to look at this blog article before reading what follows.

If your samples are not named like `36.wav`, `37.wav`, and/or you want advanced features like many velocity layers, you need to create a folder (as described before) and have a `definition.txt` file in it. Let's say your samples are:

```
/1 PIANO/MyPiano_60_vel170.wav
/1 PIANO/MyPiano_60_vel100.wav
/1 PIANO/MyPiano_61_vel170.wav
/1 PIANO/MyPiano_61_vel100.wav
...
```

Then just create a file named `/1 PIANO/definition.txt` containing this single line:

```
MyPiano_%midinote_vel%velocity.wav
```

Then SamplerBox will automatically detect and assign all the `.wav` files to the right notes and velocity layers! It's magic!

Is it possible to change the MIDI channel and load two patches at once? (Let's say bass samples on channel 1 and horn samples on channel 2?)

Currently SamplerBox reads all incoming MIDI notes, regardless the MIDI channel. But MIDI channel handling could be easily added, if this feature is really requested. Loading two patches at once is currently unsupported.

Where should I put the sample-sets?

If you installed SamplerBox via the [image file \(RECOMMENDED INSTALL\)](#), you have to put the sample-sets on a USB-stick (or on a SD card in a USB SD card reader) that you will plug into the Raspberry Pi. This USB-stick / SD card should contain folders containing your `.WAV` samples, like this:

```
/0 Piano/
/1 Flute/
...
```

Why not using the Raspberry Pi's built-in microSD card? Two reasons:

- 1. Because SamplerBox is a box! The user doesn't normally have access to the internal microSD card. The internal microSD card is used for OS and software, not for user sample-sets!*
- 2. Because you want to be able to plug in / remove / plug another SD card into the SamplerBox live! This wouldn't be possible by using the internal microSD card.*

If you installed SamplerBox via the [MANUAL INSTALL](#), you can change the config in one line to use whatever you want as the sample-set source directory.

How to change the current preset?

Most MIDI keyboards have buttons called `PROGRAM + / PROGRAM -` that will send *ProgramChange* MIDI messages. These MIDI messages are used to change SamplerBox's current preset. How to change the current preset if you don't have such buttons on your keyboard? Use SamplerBox's [hardware buttons](#), it's exactly what they are made for!

What audio formats are supported?

SamplerBox uses standard WAV files, stereo or mono, 16 bits or 24 bits, at a sampling rate of 44.1 Khz. It doesn't support AIFF, MP3, OGG, FLAC, etc. files.

Do I need a Raspberry Pi 2 or will it work as well with a Raspberry B / B+?

It will work on a Raspberry Pi B / B+, but better performances / higher polyphony will be achieved with a Raspberry Pi 2.

How do I put the SamplerBox image file on a microSD card?

See [instructions here](#).

(For developers only) Why is the filesystem mounted as read-only by default, when I use the SamplerBox image file?

In short, removing the power cord without doing `halt` on a normal read-write filesystem could cause filesystem corruption.

So if we want everything to work well, there are two solutions: either we have a normal read-write filesystem, and then we need to use `halt` command to shutdown safely the SamplerBox (but this is impossible, as everything is embedded in a box, with no keyboard!), or we use a read-only filesystem, and we can safely shut down the SamplerBox ... by just removing the power cord or using an ON/OFF switch (like on every synthesizer, for instance)! We used this second solution. If you know a better solution (read-write filesystem + safe shutdown when we remove the power cord), please contact us.

Please note that it's always possible to remount as read-write after boot by doing `mount -o remount,rw /`

What about looping? I have a sample of an organ which is 1 second long, what happens if I press the key for two seconds?

You just need to save **loop markers** in the WAV files with your traditional sound editor (I recommend Sony Soundforge), and SamplerBox will recognize them and loop the sound!

How to permanently change the sound volume?

This will evolve and be simpler in the future. For now, run this:

```
alsamixer && mount -o remount,rw / && alsactl store
```

Then select your soundcard with the key `<F6>`, change the volume, and exit with `<ESC>`. The sound volume will be permanently saved.

The audio output quality is bad. Why, and how to solve it?

This is a well-known problem: the Raspberry Pi has a very poor built-in soundcard, resulting in noisy and sometimes stuttering sound. The only solution for this is to use a DAC, such as [this 6€ DAC](#), which has a very good audio output.

When I boot the Raspberry Pi with the SamplerBox image, the software doesn't start automatically. How to solve this?

The [SamplerBox image](#) is designed to be ready-to-use. The SamplerBox software should start automatically on boot. If not, there's a configuration issue. Open `/root/SamplerBox/samplerbox.py`` and try another value for ```AUDIO_DEVICE_ID`, it should solve the issue (try with the value 0 for example). If not, [come to the forum](#) and give some details about your configuration!

Do I really need to build the whole thing (electronic parts, etc.) to use SamplerBox?

No, you don't need to. You can start with just a bare Raspberry Pi and no electronic parts. [Read more about it here.](#)

Why is it impossible to edit the samples directly on SamplerBox? Why not add a screen, a graphical user interface and editing features on SamplerBox, like on an Akai MPC?

This would be possible with some work, but it would become a new, different project.

The philosophy of SamplerBox is a bit different than a "DIY Akai MPC". My initial goal for SamplerBox was to design what we could call a **customizable expander**. It's designed to be able to comfortably load 500MB sample-sets, like big beautiful Piano sample-sets, with many velocity layers, etc. Such sample-sets cannot really be created on the small screen of a sampler. In a word, to program such sample-sets, you need a computer anyway.

The initial philosophy was: prepare the sample-sets on a computer, drop them on a SD-card, and then insert the SD-card in SamplerBox, and that's it!

Instead of doing two things badly (playing samples + poor editing on a small screen, with no keyboard, no mouse, etc.), I prefer to focus on doing one thing well: to be able to load big nice sample-sets that you've prepared on computer.

Support

The easiest way to get help with your SamplerBox is to visit the [SamplerBox Forum](#) or [SamplerBox Chat](#). We hang out there and you can get real-time help your project.

The other good way is to open an issue on [Github](#).

License

[Creative Commons BY-SA 3.0](#)