

Squishbox User Guide

models 0001-0099
software version 2.0

Contents

Squishbox User Guide.....	1
Overview.....	1
Quick Start.....	2
Hardware.....	3
Squishbox Diagram.....	3
Adjusting Contrast.....	3
Interface.....	4
Patch Menu.....	4
Soundfont Menu.....	4
Options Menu.....	4
Advanced Options Menu.....	5
Remote Configuration.....	5
Settings Files.....	6
Global Settings.....	6
Bank Files.....	6
Advanced Usage.....	8
Editing Soundfonts.....	8
Software Updates.....	8
Troubleshooting.....	8
Credits.....	9
License.....	9

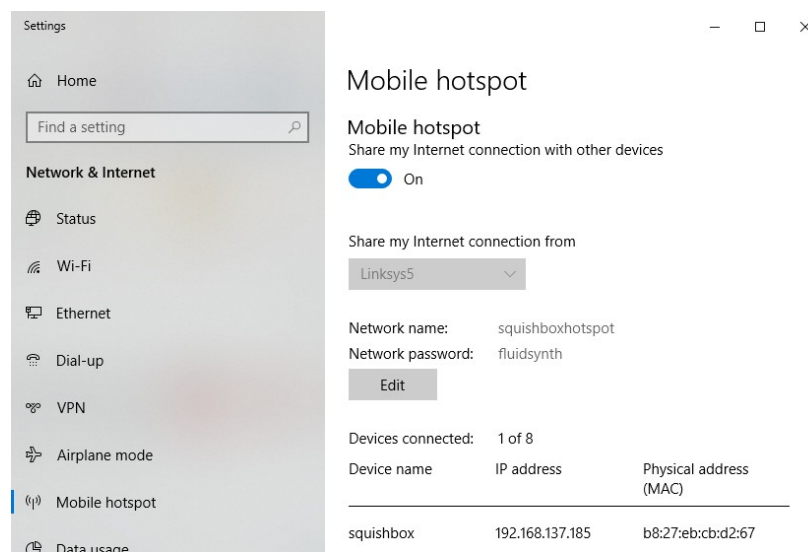
Overview

The Squishbox is a Fluidsynth synthesizer running on a Raspberry Pi built into a stompbox. It will take MIDI input from controllers plugged into its USB ports and output 24bit stereo sound at 192kHz on its 1/4” outputs. The squishbox can operate in two modes: soundfont or patch mode. Depending on the mode, tapping the stompbox buttons will select different presets from the current soundfont to be played on MIDI channel 1 (see “Open Soundfont” below), or in the more common use-case will select patches, which are user defined combinations of presets, control changes, and midi router settings.

Various menus can be accessed by long-pressing (~2 seconds) either the left or right stompswitch. In menus, long-pressing the right stompswitch will generally select an option, while long-pressing left will back up or exit the menu. Most menus will also time-out/cancel after ~5 seconds.

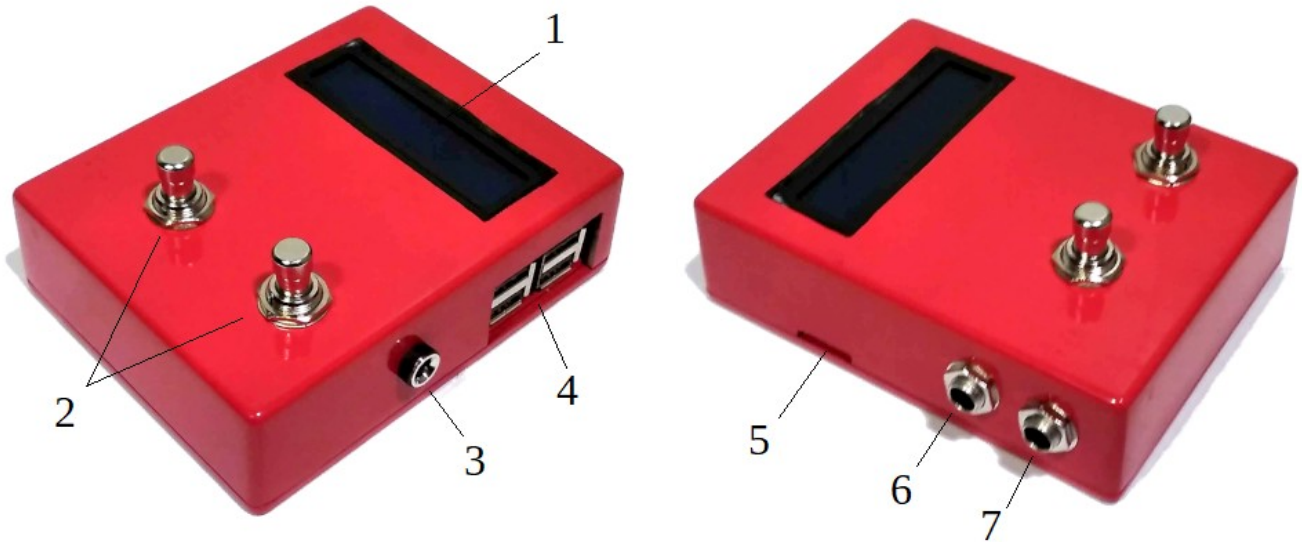
Quick Start

- Plug the SquishBox in via the supplied AC adapter (or a comparable 5V/2A power supply).
- Plug a MIDI controller into any of the USB ports.
- Connect 1/4” cables from the right and/or left output jacks to an amplifier or mixer, or plug headphones into the left output jack.
- If your controller is not detected on startup or you plugged it in after powering the unit, you can re-scan for it. Long-press left to open the options menu, then select “Advanced..”, followed by “MIDI Reconnect”.
- You should be ready to jam! The default bank, “bank0.yaml” contains a small selection of patches that should work in a variety of musical situations.
- If the default patches aren’t enough for you, you can open the Fluidsynth GM soundfont by selecting “Open Soundfont” in the “Advanced..” portion of the options menu and selecting “FluidR3_GM.sf2” and browse through those sounds. Or, if you want to create your own patches and banks, read on..
- To access the Squishbox’s control panel via wifi, create a hotspot on your phone/tablet/computer named `squishboxhotspot` with password `fluidsynth`. You should see the Squishbox connect in your hotspot settings menu. Point a web browser to the IP address of your Squishbox and you will see the control panel, where you can edit banks, modify settings, upload soundfonts, etc.



Hardware

Squishbox Diagram



1. 16x2 character LCD
2. Stomp switches
3. DC jack*
4. USB ports
5. SD card slot
6. Headphones/Left output**
7. Right output

*best results with 5V/2A power supply (included)

**provides stereo to 1/4" TRS headphone plug or mono (combined L+R) to TS plug if right output is unused


Adjusting Contrast

The contrast of your LCD display is set with a 10K potentiometer. The screw of the potentiometer is behind the USB ports, and can be reached for adjustment by inserting a small flathead screwdriver between the ports as shown.



Interface

Long-pressing the right stomp switch will open the patch menu or soundfont options menu, depending on the mode. Long-pressing left opens the general options menu.

In some menu options text can be entered (albeit slowly) using the stomp switches. Tapping left or right will change the character under the cursor, while holding left or right will advance or back up the cursor. To confirm the text entry, advance to the end of the text and the “accept” symbol () will appear. Long-press right to confirm the text entry. To cancel, wait ~5 seconds for the text entry to time out.

Patch Menu

- Update Patch – Save the current patch with any updated control change/program change values
- Save New Patch – Save the current patch and any updated control change/program change values as a new patch
- Rename Patch – Enter a new name for the current patch using the cumbersome stompbox button text entry method
- Delete Patch – Delete the current patch. Can’t do this if it’s the only patch left

Soundfont Menu

- Save as New Patch – Create a new patch in the current bank with the current soundfont preset loaded into MIDI channel 1
- Exit Soundfont – Unload the current soundfont and return to patch mode

Options Menu

- Switch Bank – Load a different bank
- Save as New Bank – Create a new bank from the current set of patches. Will suggest a name that can be modified with text entry. Scroll to the end of the name and long-press right to save.
- Set Gain – Change the gain (volume) level of fluidsynth. Higher gain values are louder but require more simultaneous voices and thus processing power.
- Chorus/Reverb – Modify the parameters of the chorus/reverb unit.
- Advanced.. – Open the advanced options menu
- Power Down – Shut down the Squishbox. You can always just unplug, but there is a risk of corrupting the SD card if you do.

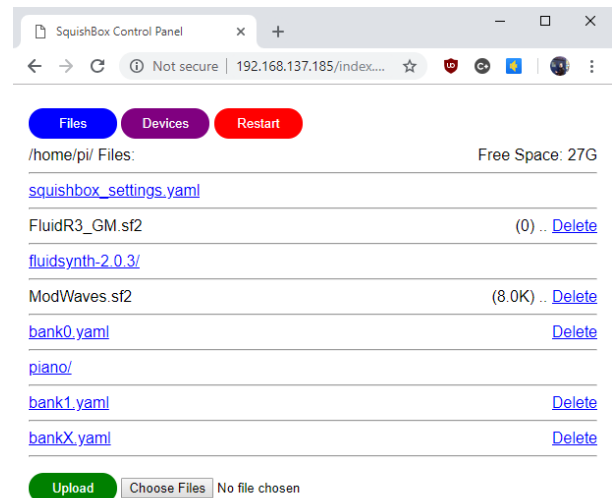
Advanced Options Menu

- Open Soundfont – Choose an .sf2 file from the pi user’s home directory (/home/pi), open it, and switch to soundfont mode, in which the user can scroll through presets in the soundfont file.
- MIDI Reconnect – Detect any connected MIDI controllers and attempt to link them to fluidsynth.
- Wifi Settings – Show the current network name and IP address if connected. Tapping left or right reveals an option to add a network name and passkey using the stompswitches.
- Add From USB – Hunts for a connected USB drive, finds all .sf2 files on it, and copies them to your Squishbox. Preserves folder structure, so if you have your fonts organized into different folders on your USB they will stay that way on the Squishbox.

Remote Configuration

To be able to modify all of your Squishbox settings, you will need wifi access to it. By default, the Squishbox will try to connect to a wireless network with the name `squishboxhotspot` using the password `fluidsynth`. Create a hotspot with these settings on a computer, phone, or other device, and the Squishbox will connect to it, allowing you to gain further access. You can add additional wireless networks using the “Wifi Settings” option described above, or you can manually add them as described below.

The Squishbox has a simple remote configuration tool that can be accessed by pointing a web browser to the Squishbox’s IP address (see “Wifi Settings” above). This tool allows editing of the Squishbox’s global configuration file (`squishbox_settings.yaml`) and bank files, and also provides a file manager for uploading/organizing soundfonts. By default, the currently-selected bank is opened for editing. The “Files” button accesses the file manager. “Devices” shows a list of MIDI devices the system is able to detect, in case you are having trouble connecting one of your MIDI controllers. The “Restart” button restarts the squishbox interface, allowing modified settings to be applied.



You can ssh to your Squishbox (e.g. using [PuTTY](#)) or FTP (e.g. [Filezilla](#)) with the username `pi` and password `raspberry`, and manually edit files or upload soundfonts. You may want to change the password at the earliest opportunity, since it is the default for the Raspbian distribution. Once logged in, you can manually add a network using the command `sudo raspi-config`, or by manually editing the file `/etc/wpa_supplicant/wpa_supplicant.conf`.

Settings Files

Global Settings

The global settings and bank files are in [YaML](#) format – a plain-text, human-readable format for representing nested data structures. The file `squishbox_settings.yaml` contains global settings.

Recognized tokens are:

- `currentbank` – the last bank loaded
- `initial bank` – the bank to load on first starting
- `uselastbank` – if `True`, always set `initialbank` to `currentbank` when loading
- `soundfont` – if not empty, load the soundfont on startup and select soundfont mode
- `fluidsettings` – a list of *key: value* pairs of settings to pass directly to `fluidsynth`
- `fluidversion` – The latest, fanciest version of `fluidsynth` is not always available in the standard Raspbian repositories. This allows use of newer, locally-compiled versions if available. If commented out, the standard Raspbian version of `fluidsynth` will be used.

Bank Files

Banks are YaML files that contain one or more patches. A *patch* consists of a name, soundfont assignments for one or more MIDI channels, midi router commands, and control change (CC) messages that are sent when the patch is loaded. The bank itself can have global chorus/reverb parameters, gain, and midi router commands.

The file structure is (*required):

- `soundfonts` – a dictionary of *label: path* values which provides a shorthand for specifying soundfonts in patches
- `gain` - initial gain level for `fluidsynth`, defaults to 0.5 if not present
- `chorus_level` - output level of chorus unit
- `chorus_nr` - number of delay lines for chorus
- `chorus_depth` - modulation depth (ms)
- `chorus_speed` - modulation speed in Hz
- `chorus_type` - 0 or 1, where 0=sine and 1=triangle
- `reverb_level` - output level of reverb unit
- `reverb_roomsize` - size of the room simulated by reverb

- `reverb_width` - amount of early reflection (brightness) of reverb
- `reverb_damping` - high-frequency damping amount
- `router_rules` - a list of MIDI message routing rules to send to fluidsynth when each patch in this set is selected; a rule can be one of the following:
 - `clear` - clears any previous routing rules
 - `default` - clears any previous routing rules and restores 1-1 routing for all message types
 - a dictionary defining a router rule; closely follows the format used by the fluidsynth command shell
 - `type*` - one of the message types defined by fluidsynth; if this is all that is given routing defaults to 1-1 for this type
 - `chan` - a sequence describing the channel routing in the form `min, max, mul, add`, where `min-max` is the incoming channel range to accept, and the channel number is then multiplied by `mul` and added to `add`.
 - `par1` - routes parameter 1 (e.g note value or CC value) in the same way as `chan`
 - `par2` - routes parameter 2 if there is one (e.g. note velocity)
- `patches` – a list of patches, where each patch is a dictionary containing the following items
 - `name*` - text to display on the LCD when the patch is selected
 - `0-15` – the MIDI channel to with the preset specified in the corresponding dictionary element will be applied
 - `router_rules` - router rules for this patch, applied after bank router rules; if no router rules are specified by bank or patch defaults to 1-1 routing for all types
 - `bank` – soundfont bank number
 - `program` – soundfont preset number
 - `cc` - a dictionary of control change *numbers: values* (0-127) to send on this channel when the patch is selected. Any modified CC values will be saved to the patch if it is updated.

Advanced Usage

Editing Soundfonts

A lot of functionality can be obtained by cleverly constructing patches and using CC messages and midi routing, but there is a lot of tweakability built into the soundfont specification itself. Modulators can be used to allow MIDI messages to control specific aspects of how a sound is synthesized, such as chorus level, filters, or ADSR envelope. A couple soundfonts have been included with the Squishbox showing examples of using modulators to enhance performance capability: “ModWaves.sf2” and “jRhodes_trem.sf2” (see credits below). One can of course also create entirely new soundfonts from scratch using audio samples. [Polyphone](#) is a free and highly versatile soundfont editor. The [soundfont specification](#) can be a helpful resource when editing soundfonts. A PDF version of the specification has been uploaded to your Squishbox.

Software Updates

Your Squishbox is powered by a Raspberry Pi 3, a fully-fledged computer. Updates to the software can be found on the associated [GitHub](#) site. Complete SD card images can be found in [releases](#). The SD card on which the operating system is stored can be removed most easily with a small pair of needlenose pliers, and reimaged using the same type of disk imaging software used to burn [Raspbian](#) images.

The model number of your Squishbox should be provided on purchase, and is also written on the inside of the enclosure lid. Successive models may differ slightly in wiring. When updating your software, edit the `stompboxpi_hw_overlay.py` file and copy the appropriate hardware-specific values for your model from the commented section at the top of the file to the end to ensure proper operation.

Troubleshooting

If any problems arise with your Squishbox unit, feel free to email albedozero@geekfunklabs.com for technical assistance.

Credits

The following soundfonts have been included with Squishbox:

ModWaves.sf2

"ModWaves" (<https://github.com/albedozero/squishbox/blob/master/ModWaves.sf2>), by Bill Peterson (<https://github.com/albedozero>) CC BY SA (<https://creativecommons.org/licenses/by-sa/2.0/>)

jRhodes_trem.sf2

"J-Rhodes" (http://www.fox-gieg.com/rkhive/new/new_piano/jrhodes3b-stereo.zip), by Jeff Learman (<http://rkhive.com/>) CC BY 4.0 (<https://creativecommons.org/licenses/by/4.0/>)

modulators were added to allow control of tremolo depth and rate

fazioli_grand.sf2

"Fazioli Grand Piano" (<https://www.retrosampling.se/freesamples.htm>) by Soeren Bovbjerg (bovbjerg@musik.auc.dk), licensed as freeware (C) 1999

SoundFont® is a registered trademark of E-mu Systems, Inc.

License

This work is licensed under a [Creative Commons Attribution-NonCommercial-ShareAlike 4.0 International License](https://creativecommons.org/licenses/by-sa/4.0/), 2019 by Bill Peterson.