Pi MusicBox

What is it?

So you have this huge library of 20+ million songs on Spotify, just waiting for you to play them on your computer. Spotify is great. But how to play these songs on your audio set? Using a headphone cable? Hmmm. Not great with your notebook. Bluetooth? Hmmm. Unreliable.

Enter Pi MusicBox. It turns you Pi into a music player, supporting Spotify, Apple Airplay and Web Radio. A music player which you can operate from your couch using a tablet, smartphone, notebook or desktop computer. Connect your Raspberry Pi to an audio set, install the software and welcome to a new way of listening to music!

Requirements

For this hack, you'll not only need a working Pi, but also a Spotify Premium account and an audio set, headphones or a set of (USB-)speakers. A monitor or television is not required, but might come in handy.

Getting the Software

Download the 'ready to eat' image from www.pimusicbox.com. It just works, you only have to edit some options. For this guide, we used Pi MusicBox version 0.4. New versions should generally work about the same, though options could change.

Installing Pi MusicBox

Extract the zip-file. Put the resulting image on your SD-Card by following the instructions in chapter [XX]. The image is tested on a 1GB SD-card, but you can put it on a larger one.

Possibilities

Pi MusicBox is designed to be operated not from the Pi itself, but from another computer like a notebook, desktop, tablet or smartphone. You can use almost any device with a modern web browser to operate it (Chrome 14+, Safari 6+, Firefox 11+, Internet Explorer 10+).

It can output the sound not only to the headphones jack of the Pi, but also through HDMI and USB. It supports Wifi, Last FM, and it can play your own music files from the SD-Card or the network. The software will detect as much of the configuration as possible at boot, and configure the system automatically. But you have to supply the system with some information.

Configuring your Box

To make it as easy as possible to configure, you can edit the settings of Pi MusicBox without ever logging into Linux. To do this, put the SD-Card into a computer (Windows, Mac, Linux), and open the contents of the card in the file manager.

A special folder called `config` should be there. In it is a file called `settings.ini`. All configuration is done here, so start up your favorite text editor!

The file is structured as an ini-file. All lines starting with a # are comments. Read the comments, or just ignore them. Some configuration lines are also commented out. If you want to use them, remove the # at the beginning of the line.

You can by the way also edit this file when you boot your Pi and you login (see **Getting your Hands Dirty**). The file then is located in /boot/config/.

Basic Configuration

The most basic configuration file looks like this:

```
[MusicBox]
SPOTIFY_USERNAME = ‘spotifyusername’
SPOTIFY_PASSWORD = ‘spotifypassword’
```

[MusicBox] is the section of the configuration file. It has to be there, but you can ignore it. *spotifyusername* is your the username (or Facebook login if you use Facebook to connect to Spotify) of your Spotify Premium account and *spotifypassword* is of course your own password. This should be enough to run Pi MusicBox.

Networking

Pi MusicBox needs a network connection to operate. If you use a wired network, you only have to plug in the network cable in the Pi. It will be configured using DHCP. If you want to use a static address, you need to get into the console and configure it yourself (see **Getting your hands dirty**). Connecting through Wifi using a dongle is also supported (more information on that below).

Booting

After you edited the basic settings, you can boot Pi MusicBox. You can edit a lot more settings, but it’s good to try booting with the basic ones first. Put the SD Card in the Pi, connect speakers and connect the power. If you connect a monitor to the HDMI connector, you can follow the booting process. After a minute or so, you can access the system.

Accessing the Web Interface

Point your browser to the Pi. Depending on your network and computers, it will be available at this address:

http://musicbox.local/

Most OS X/iOS and Windows devices probably will find it immediately. If it doesn't work, you could try to install Apple Bonjour/iTunes in Windows to make it work. Linux should also work if Avahi or Samba/Winbind is installed. You have to add the http:// part of the address in some browsers, because otherwise, it will try to search for musicbox.local in Google.

If it doesn’t work, or if you use Android (which does not support Bonjour/Avahi/Samba), you have to access MusicBox using the IP-address of your Pi. This address is different on every network, e.g. http://192.168.1.5/ or http://10.1.100.2. You have to either look it up using a network utility, or (easier) read it from the screen when Pi MusicBox boots.

Play songs from Spotify

If you access the interface, most things are clear immediately. Your Spotify playlists will appear. Select one of the playlists to show the tracks it contains. Select the track you want to play and choose one of the play options in the popup menu that appears. Using this popup menu, you can either play the song, add it to the queue, or find more tracks from the same album or the same artist. Just play around. Literally.

The four buttons at the bottom of the screen, let you access the functions of MusicBox. You can access the playing Queue, search for tracks, albums and artists, or play web radio stations.

Web Radio

To play streams from radio stations you like, you have to use a so called stream url. You cannot use container files like M3U, XSPF or PLS (yet), which are commonly available, you have to add the real stream. This stream url is hidden inside the .M3U or PLS file. To find this url, open the container file in a text-editor.

A PLS file looks like this:

```
[playlist]
numberofentries=1
File1=http://vprbbc.streamguys.net:8000/vprbbc24.mp3
Title1=BBC World Service
Length1=-1
version=2
```

The stream url would be:

```
http://vprbbc.streamguys.net:8000/vprbbc24.mp3
```

M3U and XSPF files look different, but the stream url is always clearly visible.

You can find radio stations (PLS and M3U) using services like http://dir.xiph.org/ or **Listenlive.eu** or **Dirble.com**.

Just add the stream url and the name of the station and press the Play button. The last 25 stations are saved locally using a browser cookie (not on the server yet, so you need to do it on every client you use).

MPD

Though the web based interface is recommended, you can also use native software which support the MPD (Music Player Daemon) protocol to control Pi MusicBox.

Apps and applications are available for all sorts of devices and operating systems. Not all of them work great with MusicBox though. For Android, *MPDroid* is recommended. On OS X, Theremin works (without search). On Linux, you can use the great working and wonderfully named console app *ncmpcpp*. On a Linux Desktop, GMPC and Sonata work well. On iOS, mPod and mPad. For Windows, clients are either not working great or untested.

More settings

A lot of things can be configured on Music Box. Edit the configuration file according to your needs. You have to reboot the Box to see the changes.

Wifi Network

If you connect a supported wifi-dongle to your Pi, the MusicBox software should be able to detect and use it instead of a cable connection. Most dongles are supported, but not all. If you buy one, make sure it's supported by Raspbian, the Linux distribution on which MusicBox is based.
To make wifi work, you have to fill in the network name (SSID) and your password in the config file. Add these lines to the basic configuration above, or edit the default file supplied with MusicBox.

```
WIFI_NETWORK = 'mywifinetwork'
WIFI_PASSWORD = 'mypassword'
```

Substitute the *mywifinetwork* and *mypassword* with the correct values of your own network. For now, the wifi on Pi MusicBox only supports WPA(2) encrypted networks, configured via DHCP. As with a wired network, if you want to use a static address, WEP encryption or no encryption, you need to get into the console and configure it yourself (see **Getting your hands dirty**).

Better Quality

The Pi can play the music from Spotify in different types of quality. The better the quality, the more data needs to be downloaded from Spotify. It's called bitrate. Higher quality means a higher bitrate and a bit more use of your internet connection. Typical broadband connections should be able to support the highest bitrate easily. If you have a good connection to the internet, you can set the quality to high, but if your connection is slow or unstable, or you have usage limits on your connection, you can it lower and use less data. Possible rates are 96 (low, but acceptable quality, FM-like), 160 (default) or 320 (highest quality, CD-like).
Set the bitrate to high like this in the configuration file:

```
SPOTIFY_BITRATE = 320
```

Or set the bitrate to low like this:

```
SPOTIFY_BITRATE = 96
```

Sound Configuration

By default Pi MusicBox will send the sound to the analog headphone output on the Pi. This sound is good enough, but due to hardware constraints, not always great. If you want to have better sound, use the HDMI to connect the Pi to an amplifier, or connect an USB soundcard (also called USB DAC, Digital Audio Converter), USB speakers or USB headphones. Almost all types of USB speakers, headphones and DAC's are supported, but if you buy one, make sure it's Linux compatible. DAC's with digital outputs are also available in many web stores.

When booting, Pi MusicBox will autodetect what is connected to the device and configure it accordingly. If you connect multiple devices, USB will be selected first as a sound output, HDMI after that, and lastly the analog output of the Pi itself. You can override this in the configuration-file using the following line:

```
OUTPUT = 'analog'
```

If you include this, the default output will be the analog headphones jack of the Pi, even if you connected an USB device or an HDMI cable.
The options are: 'analog', 'hdmi', 'usb'

Last FM

Another service supported by Pi MusicBox is Last FM. It collects the tracks you play, so you can discover new music. Go to www.last.fm to create an account if you don't already have one. To let Last FM collect the tracks you play, fill in the credentials of this service.

```
LASTFM_USERNAME = 'lastfmuser'
LASTFM_PASSWORD = 'lastfmpassword'
```

SoundCloud

Another service supported by Pi MusicBox is SoundCloud, the service which lets you "Hear the world's sounds". To configure it, you need a special ID, a token. Get this token from http://www.mopidy.com/authenticate You have to login with your SoundCloud id to get the token. This information is not shared with the mopidy.com site. When you login, you'll see a token appear on the page. Add this token to the ini file like this:

```
SOUNDCLOUD_TOKEN = '1-1111-111111'
SOUNDCLOUD_EXPLORE = 'electronic/Ambient, pop/New Wave, rock/Indie'
```

Where you replace the example 1-111-111111 by your token. Using the SOUNDCLOUD_EXPLORE configuration, you can configure the playlists you want to see in the interface.

Multi Room Audio

Pi MusicBox supports so called Multi Room Audio. You can have multiple Raspberry's on your network, for example in different rooms. The devices need to have their own names to be accessible. Use this option to give your MusicBox a different name:

```
NAME = `Kitchen’
```

The name you choose should be no longer than 9 characters and only contain normal characters and numbers in the name (no spaces, dots, etc).
After a new boot, the webinterface for playing music will be accessible via a new address. Where the default would be http://musicbox.local from devices that support Bojour/Avahi, when you change the name, it becomes http://newname.local. In the example above it would be http://kitchen.local/
It’s not possible to play different music on multiple devices using the same Spotify-account at the same time. This is a limitation of Spotify. If you have multiple accounts, it of course is possible.

Security
Pi MusicBox is not totally secure and not intended to run outside a firewall, only in the cosy environment of your local network. The heart of MusicBox, is not protected enough to do that. Also, the passwords of Spotify and wifi are stored in plain text on the SD-Card. This might be fixed in the future.
For more security, change the default password by setting this line:

```
MUSICBOX_PASSWORD = ‘mypass’
```

where *mypass* is your new password. This will change the passwords of both the user *musicbox* and the user *root*. The password will be removed from the configuration file after it’s updated.

If you want, for more security to change the root password to something else, use this line:

```
ROOT_PASSWORD = ‘mypass’
```

where *mypass* again is your new password.

Playing your own Music Files
Though Spotify boasts a library of over 20 million tracks, not all artists and songs are represented. So it would be nice to be able to play MP3 files for the missing songs, wouldn’t it? Well the good news is that Pi MusicBox supports playing local or networked MP3, FLAC or OGG files. The bad news is that it’s a tiny bit complicated in the current version (0.4). Also, the songs are not easily available in the webinterface. They are not in the playlists, you have to search for them to play them.

Networked Music
The easiest way to play your own music files, is via the Windows Network. To do that, edit the configuration file, so that MusicBox knows where your files are. This address could be a bit cryptic to a first time user. This is an example:

```
NETWORK_MOUNT_ADDRESS = '//192.168.1.5/musicshare'
```

or:

```
NETWORK_MOUNT_ADDRESS = '//mynasserver/shared/music'
```

The first part // is the way shares in the Windows Network are created. Just add it and forget it. The next part (*mynasserver* or *192.168.1.5*) is the name or ip-address of the server which hosts the file, and the last part */musicshare* or */shared/music*, tells MusicBox which share to mount. When your server is protected, you need to set the username and password for the Network share using the following configuration lines:

```
NETWORK_MOUNT_USER = 'username'
NETWORK_MOUNT_PASSWORD = 'password'
```

Scan Music

MusicBox will not see the files immediately. The music files needs to be scanned at boot, every time you add or remove files. This process can slowdown the boot of the MusicBox, so use it with care. MusicBox will scan the files using the following configuration lines:

```
SCAN_ONCE = 'true'
```

or

```
SCAN_ALWAYS = 'true'
```

The names speak for themselves. Using SCAN_ONCE, the music files will only be scanned, yes, once. Use this if you don't change the music files often. Use SCAN_ALWAYS if you change your music files a lot. This will enable you to change the files and reboot MusicBox. It will recognize the new files after the boot. But, again, the scanning process can slowdown the booting of MusicBox considerably.

Local Music

Pi MusicBox also has an option to store music files on the SD-Card. This process is also a bit more complicated. Since MusicBox is created for a 1GB SD-Card, or larger, the file system is also less than 1 GB. If you put MusicBox on a larger SD-Card, the rest of the space on the card won't be used, unless you resize the file system.

You can do this manually, on a computer using a partition manager, or you can let MusicBox try to resize it automatically. This process is tested, but not guaranteed to work. You could end up with a non-working musicbox if the process fails. That's most of the time no problem, since you can put the original MusicBox image on the SD Card again and start over. If you did a lot of customization, it's recommended to backup your card first.

Using this line in the settings, Pi MusicBox will automatically resize the filesystem to the maximum size of the SD-Card:

```
RESIZE_ONCE = 'true'
```

Put Files on the Card

Putting music files on the SD Card is only recommended on cards with a size larger than 1GB. MusicBox needs the 1GB for caching and other storage. After resizing an SD-card with more storage, you can put your own music files on the Pi using either the Windows Network, or by

mounting the root filesystem of the card on a Linux computer and copying the files. Leave at least 200MB of free space on the device.
To use the Windows Network, you have to have the workgroup name of the Windows Network set to the default name, WORKGROUP. If you want another name, you have to change it by hand in the file /etc/samba/smb.conf (see **Getting Your Hands Dirty**).
Remember to let MusicBox scan the files at boot (see **Scan Music**)

Getting Your Hands Dirty

If you are willing to get your hands 'dirty', there are a lot more options to explore in Pi MusicBox. For this, you have to login to the box on the console, or via SSH.

To login remotely via SSH, you will need to enable the SSH service. Do that by adding this line to your configuration-file:

```
SSH_ENABLED = 'true'
```

Reboot. After that, you can connect to MusicBox via SSH.

Mopidy

The main ingredient of MusicBox is Mopidy, an open source music server developed by people from all over the world. It can be extended in a number of ways. By default, Pi MusicBox is set up using the best working extensions. But it can be extended to play music from e.g. SoundCloud, Google Music and Beets Music. More extensions are developed as you read.

How to add these extensions is beyond the scope of this document, but a lot of resources and documentation can be found on www.mopidy,com The developers can be reached on the mail list of Mopidy, https://groups.google.com/forum/?fromgroups=#!forum/mopidy, or via IRC Chat on the #mopidy channel on Freenode.

rc.local

Another important piece of Pi MusicBox is the file */etc/rc.local.* It's a shell script. This is where the (sound) hardware is setup and the configuration is done. For example, the configuration file of Mopidy is created from rc.local. Edit this file is you want to add, change or remove features.

Working at Midnight

For Linux novices, a nice utility called Midnight Commander could be of use to browse the filesystem and edit files. It works like the age old DOS-utility Norton Commander and it's included in MusicBox. Start it using the command: `mc`

Static Network

To use MusicBox in a network with static IP-addresses, you have to edit the file */etc/network/interfaces*.
The lines that configure the wired network, look like this:

```
allow-hotplug eth0
iface eth0 inet dhcp
```

An example file for a static wired network, you should change it to something like this:

```
iface eth0 inet static
address 192.168.1.5
netmask 255.255.255.0
gateway 192.168.1.1
```

Fill in the correct ip-addresses for your network.

Updating

When a new version of MusicBox is released, the only way to update it, is to do a new installation. You can update the kernel and other packages of the system manually, but changes in the files specific for MusicBox will not be updated, so it could eventually break things. Generally it's not needed to update things, but if you really want, you could issue the command: `rpi-update` to get the latest kernel. This will take a while. Another command is `apt-get update && apt-get dist-upgrade`. These commands take a while to run, so grab a coffee!

Fun & Questions

Enjoy your new way of listening to music! If you have questions, don't be afraid to ask them at The mailing list of Mopidy/MusicBox, or via chat. Addresses and instructions are on www.pimusicbox.com