

# The smart boombox cookbook

This document describes how to create, or buy, a *boombox* that sounds great and is smart enough to ask it to play the music you want to hear.

A Raspberry Pi 4 running GNU/Linux drives the smarts and a Digital to Audio Converter (DAC), amplifier and quality speakers that creates the great sound.

Also described is a *basic boombox* if you just want a boombox without the computer components.

This document has the following sections:

- ▶ "Overview" on page 2
- ▶ "Choices made" on page 3
- ▶ "Bill of materials" on page 4
- ▶ "Assemble the frame" on page 7
- ▶ "Install the speakers and amplifier" on page 8

If you are building a *smart-boombox*, also follow these sections:

- ▶ "Load GNU/Linux on the micro-SD card" on page 9
- ▶ "Connect the computer hardware" on page 11
- ▶ "Install and configure software" on page 13

Once your system is built, installed and configured, whether smart or basic, follow the final section.

- ▶ "Test and enjoy your boombox" on page 35

## 1.1 Overview

Wikipedia defines a *boombox* as:

...a transistorized portable music player featuring one or two cassette tape recorder/players and AM/FM radio, generally with a carrying handle. Beginning in the mid 1980s, a CD player was often included.

<https://en.wikipedia.org/wiki/Boombox>

Today, cassettes are ancient history, CDs are rare and AM/FM is used less frequently, but the concept of a boombox lives on. There are other new sources of music that a modern boombox can leverage, including:

- ▶ Bluetooth connection to a device that can play music such as a cell phone,
- ▶ USB drives with song files,
- ▶ "Aux in" jack to drive music from other devices, such as a CD player or phonograph, and
- ▶ Internet streaming of radio stations, music playing apps and services such as Pandora and Spotify.

The resulting device can be considered either a boombox that is really smart, or as a general purpose computer that sounds great.

The boombox has a *2.1 channel* amplifier. That means the amp has outputs for two full range speakers and one subwoofer that drives the lower frequencies. All have down-firing subwoofers which helps to keep gravity on its side. There is also a headphone amplifier, on the smart boombox, if you want to listen to music and keep it quiet.

There are covers for the left and right speakers, and for the subwoofer to protect them. There are no covers (though you could add them) for the two passive radiators. The philosophy is to protect the expensive components, but to leave the less expensive passive radiators exposed for a better look and sound.

### 1.1.1 A basic boombox?

Maybe you don't want your boombox to be so smart? All of the computer components can be omitted in which case you have a *basic boombox*. If you build or buy one of these, you won't be able to:

- ▶ Give it voice commands,
- ▶ Stream music from the Internet,
- ▶ Listen to music through headphones, or
- ▶ Have access to a general purpose computer

This approach reduces the cost and the complexity of the project. You could also first build a basic boombox, then add the smarts later.

## 1.2 Choices made

This section describes the choices made and the rationale behind those choices during research and development. The following table moves from the low-level hardware to the high-level software.

Table 1-1 Choices made and alternatives

Component	Choice made	Alternatives	Reasoning
Boombox frame	Baltic birch + solid moldings - 3 speaker + 1 top enclosure	3D-printed plastic, MDF, ammo boxes, etc.	Many builders believe Baltic Birch is <i>the</i> best material for speaker enclosures. Wood is more recyclable and more aesthetically pleasing than plastic.
Amplifier type	2.1 channel amp	2.0 channel amp, DAC HAT amp	A 2.1 amp does the crossover from highs and lows. A 2.0 amp would require dedicated crossover components or no subwoofer which would lose the lows. A DAC HAT amp does not have volume controls nor the power needed to drive these speakers.
Left/Right speaker enclosures	Ported	Sealed	The ported enclosures just sound better than sealed. Almost everyone agrees.
Subwoofer enclosure	Sealed + passive radiators	Sealed or ported w/o passive radiators	Sealed enclosures with passive radiators allow a lower frequency than the other alternatives - the lower the bass, the better!
Smart or basic boombox	Smart, but you can choose basic	Basic	If you want a voice assistant with modern streaming options, you should go with the smart boombox. If not, then buy or build the basic boombox. Both are described.
System On a Chip (SOC)	Raspberry Pi 4	Pine 64, Jetson Nano, others	The Raspberry Pi 4 with 4 GB of memory seems to have the widest acceptance and support, and is capable of driving a general purpose computer. Ideally there would be an SoC that runs totally free software, but none could be found.
Operating system	GNU/Linux Ubuntu server	Raspbian, Pycroft, Volumio, etc.	Ubuntu server GNU/Linux was chosen very closely over Raspbian (aka Raspberry Pi OS). There are other OS's dedicated to one function such as Pycroft and Volumio. If you have the horsepower to drive a general purpose computer, why not use it?
Desktop	MATE	KDE, Gnome, many others	MATE is a fork of Gnome 2 which some say is better than Gnome 3. This is perhaps the easiest component to change. <b>TODO: try KDE</b> - lighter weight than MATE?
Voice software	Mycroft	Rhasspy, many others	Mycroft seems to be the most widely accepted, free and open choice available with an important focus on privacy
Music playing software	Jellyfin	Emby, others	Jellyfin seems to be the most open. It allows for no passwords which Emby does not.
Low level audio	pulseaudio, alsa	None, really	As Mycroft and Jellyfin require pulseaudio, which requires ALSA, there is no other option.

## 1.3 Bill of materials

This section describes the hardware and software you will need to build such a boombox.

The options, in the order of least to most expensive, are:

1. Build everything yourself - just figure out the woodworking,
2. Buy an unassembled frame, glue it together, then buy and install all other parts,  
(**TODO: smartboomboxes.com product #1**)
3. Buy an assembled frame, then buy and install all other parts,  
(**TODO: smartboomboxes.com product #2**)
4. Buy a fully-assembled basic boombox, or  
(**TODO: smartboomboxes.com product #3**)
5. Buy a fully-assembled smart boombox.  
(**TODO: smartboomboxes.com product #4**)

### 1.3.1 Hardware for the basic box

Description	Qty	Cost	Source
Wooden frame (unassembled)	0/1	\$69?	At smartboomboxes.com <b>SOON?</b>
Wooden frame (assembled)	0/1	149?	At smartboomboxes.com <b>SOON?</b>
2.1 amp (Damgoo 50/50/100W)	1	\$34	<a href="http://www.amazon.com/gp/product/B089KT3FG9">www.amazon.com/gp/product/B089KT3FG9</a>
12-24v Power supply	1	\$20?	??? <b>TODO:</b> test different power supplies
Tang Band 3" full range speakers	2	\$66	PE: 264-911 \$66 (or \$57 for 4 or more)
Peerless passive radiators	2	\$16	PE 264-1060
6" subwoofer speaker grill	1	\$3	PE: 260-371
4" Left and Right speaker grill	2	\$3	PE 269-2110
3.5mm Jack	2	\$5	PE 090-281
Lighted rocker switch	1	1	<a href="http://www.amazon.com/gp/product/B07TH9YB5H/">www.amazon.com/gp/product/B07TH9YB5H/</a> 10 for \$8 (included with frame above)
Feet - 1-1/4" - all rubber or copper tubing and short rubber	4	\$10	<a href="http://www.amazon.com/gp/product/B00S2D3FPY">www.amazon.com/gp/product/B00S2D3FPY</a> -OR- back to 1" copper with 1/4" rubber feet
Handle	1	\$4	PE: 262-314
<b>Subtotal</b>		\$231	unassembled box, or \$321 assembled

### 1.3.2 Additional smart boombox hardware

Description	Qty	Cost	Source
Raspberry Pi 4 w/4 GB memory, plus power supply and heat sinks	1	\$70	<a href="http://www.canakit.com/raspberry-pi-4-basic-kit.html">www.canakit.com/raspberry-pi-4-basic-kit.html</a>
RasPi 5v Power supply	1	0	Comes with above kit
RasPi Heat sinks	1	0	Comes with above kit
RasPi inline switch	1	\$7	<a href="http://www.canakit.com/raspberry-pi-4-on-off-power-switch.html">www.canakit.com/raspberry-pi-4-on-off-power-switch.html</a>
64 GB Micro SD card/holder	1	\$13	<a href="https://tinyurl.com/ymat2es4">https://tinyurl.com/ymat2es4</a>
"DAC HAT" HiFiBerry DAC+	1	\$45	<a href="http://www.hifiberry.com/shop/boards/hifiberry-dac2-pro/">www.hifiberry.com/shop/boards/hifiberry-dac2-pro/</a>
3.5mm 2 female to 1 male cable	1	\$1	PE 240-1026
RCA female to 1 3.5mm male A	1	\$2	<a href="http://www.amazon.com/gp/product/B000I963XQ">www.amazon.com/gp/product/B000I963XQ</a>
USB microphone	1	\$14	<a href="http://www.amazon.com/gp/product/B0779PKLV9/ref=ppx_yo_dt_b_asin_title_o07_s00">www.amazon.com/gp/product/B0779PKLV9/ref=ppx_yo_dt_b_asin_title_o07_s00</a>
Electrical box, outlet and cover	1	\$6-8	Hardware store
USB drive with music files	1	?	Optional
<b>Subtotal</b>		\$158	

```
Assembled frame 149+34+20+57+16+3+3+5+10+4= $301
Unassembled      69+34+20+57+16+3+3+5+10+4= $221
Computer parts   70+7+13+45+1+2+14+6= $158
```

Most sources are either from Parts Express (partsexpress.com - abbreviated with "PE: <part-number>") or Amazon or other sources where the URL is provided.

So the cost would range from approximately \$221 (unassembled box and no computer) to \$359 (assembled box with computer). These product sources and prices will fluctuate.

### 1.3.3 Equipment for first time set up

To set up the system the first time, you will also need the following.

- ▶ A keyboard, mouse and monitor
- ▶ HDMI cable and possibly a HDMI to micro-HDMI converter
- ▶ Wi-Fi Internet or wired Ethernet
- ▶ A PC or other device with an SD card port
- ▶ A device with an "aux cable" (3.5mm male connector), such as a cell phone, that can play music to test the amplifier.

### 1.3.4 Software

You will need the following software:

- ▶ A Linux distribution - Ubuntu 20.04 server is recommended and is what is documented in this guide.  
Raspberry Pi OS (aka Raspbian) is an alternative but is not documented.
- ▶ If you have a Windows computer, the Win-32 Disk imager is recommended for copying the Linux image to SSD card.
- ▶ If you have a Linux computer, the `dd` command can copy the Linux image.

## 1.4 Assemble the frame

The entire boombox consists of four separate enclosures:

Left channel	Bottom left. This enclosure is ported with a 1" diameter, 3" long tube.
Subwoofer	Bottom center. This enclosure is sealed with two passive radiators. These are elements designed to move sympathetically with the energy of the subwoofer. Their extra mass allows for a resonant frequency lower than otherwise possible.
Right channel	Bottom right. This enclosure is also ported.
Components	Top. This enclosure has two sliding panels so that it can be completely open and give maximum access to the computer and electronic components.

If you bought the unassembled kit, you can watch a video **TODO-create-video1** on how to assemble it.

## 1.5 Install the speakers and amplifier

Now that the frame is assembled, you can install the speakers, ports, passive radiators, amplifier and wiring.

You can watch a video **TODO-create-video2** on how to install them.

**TODO:** also include basic steps to augment the video

### 1.5.1 Test amplifier and speakers

Now is time to test that your amplifier and speakers are working. To do so, perform the following steps.

- ▶ If you're building a smart boombox, unplug the output of the 5V power supply from the in-line switch to the RasPi in the top compartment (**TODO:** get a picture).
- ▶ Set the on/off switch on the front left to **off**.
- ▶ Plug the main power cord in to 115v outlet.
- ▶ Set the on/off switch on the front left to **on**. The light in the switch should come on.
- ▶ Connect a device with an "aux cord" to "Aux-in" (**TODO:** get a picture), and play some music!

If you don't hear any music (**TODO:** add some words on debugging).



## 1.6 Load GNU/Linux on the micro-SD card

If you are building the basic boombox, you can skip this entire section.

Before starting with the hardware, you need a GNU/Linux image copied onto a micro SD card. This card will be plugged into your RasPi 4 as the operating system that will drive your boombox.

This section is comprised of the following sub-sections:

- ▶ “If you only have access to a Windows system Install the Win 32 disk imager” on page 9
- ▶ “Download Ubuntu server” on page 9
- ▶ “Copy Ubuntu image to micro-SD card” on page 10
- ▶ “Download Ubuntu server” on page 9

### 1.6.1 Prepare to copy an SSD card

The computer you use to write to the SSD card must have a port for a card.

#### Prepare on Linux or macOS

If you have a Linux or macOS system, you have access to the `dd` command, which is able to copy files between the hard drive and the SSD card. It would be something like. (**TODO**: get the exact command).

```
$ sudo dd if=??? of=??? bs=???
```

#### Prepare on Windows

If you only have access to a Windows system Install the Win 32 disk imager

This software will allow you to copy a downloaded Linux image onto a micro-SD card. If you have an application that will do this, you can skip this step.

- ▶ Download the Win 32 disk imager to a Windows PC  
<https://sourceforge.net/projects/win32diskimager/>
- ▶ Install it.

### 1.6.2 Download Ubuntu server

To download Ubuntu, perform the following steps:

- ▶ Download an Ubuntu Server image from:  
<https://ubuntu.com/download/raspberry-pi>
- ▶ Click the Green **Download 64-bit** button on the left side of the page. This version is recommended because it includes “Long Term Service” (until 2025).

**TODO**: get screen shots

- ▶ A .zip or .xz file should be downloaded to your PC. For example:

```
ubuntu-20-04.3-preinstalled-server-arm64++raspi.img.xz
```

The .xz suffix implies the file is compressed in a format other than zip.

- ▶ Unzip the file with a zip tool. Most Operating Systems have these standard today. When the file is unzipped, it should have a name similar to:

```
ubuntu-20-04.3-preinstalled-server-arm64++raspi.img
```

You should now be ready to copy the image to a micro-SD card.

### 1.6.3 Copy Ubuntu image to micro-SD card

If you have a brand new SD card, you usually don't need to format it. Most come preformatted with the FAT32 file system.

If you are reusing a micro-SD card, at much time.

► **TODO:** describe Win32installer with some screen shots

At this point you should have a micro-SD card with Ubuntu server on it which should boot the RasPi.

## 1.7 Connect the computer hardware

To connect all the computer hardware, perform the following steps:

**TODO:** add pictures

- ▶ Plug the micro-SD card into the back of the RasPi.
- ▶ You can access the Internet using either Wi-Fi or with an Ethernet patch cord with an RJ-45. If you have a wired connection, it may make hooking up to Wi-Fi easier.
- ▶ Plug the 5v power supply with a USB-C end into the RasPi 4.

For the initial set up, a keyboard, monitor and mouse are needed. Ideally there will be a way of setting up “headlessly”, but that’s not available yet.

- ▶ Connect the mouse and keyboard to the USB connections on the RasPi.
- ▶ Connect the monitor to the RasPi with the HDMI cable.

### 1.7.1 Boot the RasPi

To prepare to boot the Raspberry Pi 4 for the first time, perform the following steps.

- ▶ Slide both the top and back covers off of the frame to have full access to the hardware.
- ▶ If you haven’t already, download 64-bit Ubuntu 20.04 (LTS) - described in “Download Ubuntu server” on page 9.
- ▶ If you haven’t already, copy Linux to the micro-SSD card and plug it in the back of the RasPi. This is described in “Copy Ubuntu image to micro-SD card” on page 10
- ▶ Plug monitor, keyboard, mouse and microphone into the Raspberry Pi. The Raspberry Pi 4 has two micro-HDMI jacks as it can drive two displays. If you have only one display, use the jack closest to the power supply jack (the one to the left).
- ▶ Plug in boombox main power. The power switch on the front left face of the boombox powers the amplifier. The small inline button switch inside powers the Raspberry Pi.

#### First boot

When you supply power to the Raspberry Pi, it should start booting.

- ▶ If the RasPi switch has a red LED below the on/off button. If it is not glowing red, push the on/off button once.
- ▶ Observe the top, back, left of the RasPi (which is under the DAC Hat). There are two LEDs:
  - The LED to the left should glow solid red. This signifies the RasPi has 5V DC power.
  - The LED to the right of the red one should flicker green. This signifies that an operating system is communicating with the CPU. If there is a red light, but no green one, this probably means that the micro SSD card does not have Linux properly installed.

**Important:** Never turn the RasPi off without first shutting Linux down with the `halt` command. Doing so can damage the operating system and possibly even the RasPi itself.

- Observe the monitor. You should see many messages scroll by.

## Login

- You should see Linux booting, then finally the prompt:

```
ubuntu login:
```

Note: there may be some messages printed after it.

- Enter the default credentials - both the user and password are **ubuntu**

```
ubuntu
```

```
Password: ubuntu
```

- You will be required to change your password. Enter the current password once, and a new password twice. Don't forget this password!

```
You are required to change your password immediately ...
```

```
Current password for Ubuntu:
```

```
New password:
```

```
Retype new password:
```

## 1.8 Install and configure software

To configure Ubuntu, perform the following steps.

### Get on the Wi-Fi

If you do not have a wired Ethernet connection, you will want to get onto Wi-Fi first. To do so, you must know the name of the network (aka SSID) and the password. Perform the following steps.

- Change to the directory `/etc/netplan/` and make a backup copy of the configuration file.

```
$ cd /etc/netplan
$ sudo cp 50-cloud-init.yaml 50-cloud-init.yaml.orig
```

- Edit the configuration file `50-cloud-init.yaml`. You should see network section consisting of six lines. Add a “wifis” section shown in the last 6 lines. Replace `YourSSID` and `YourPassword` with the credentials of your Wi-Fi router.

```
$ sudo vi 50-cloud-init.yaml
network:
  ethernets:
    eth0:
      dhcp4: true
      optional: true
  version: 2
  wifis:
    wlan0:
      dhcp4: true
      access-points:
        "YourSSID":
          password: "YourPassword"
```

- Reboot your system to effect the changes.

```
$ sudo reboot
...
```

- Your system should now have Wi-Fi access. Run the command `ip a` and look for the IP address in the `wlan0` section.

```
$ ip a
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group
default qlen 1000
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
        valid_lft forever preferred_lft forever
    inet6 ::1/128 scope host
        valid_lft forever preferred_lft forever
2: eth0: <NO-CARRIER,BROADCAST,MULTICAST,UP> mtu 1500 qdisc mq state DOWN
group default qlen 1000
    link/ether e4:5f:01:3e:c2:40 brd ff:ff:ff:ff:ff:ff
3: wlan0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state
UP group default qlen 1000
    link/ether e4:5f:01:3e:c2:41 brd ff:ff:ff:ff:ff:ff
    inet 192.168.1.230/24 brd 192.168.1.255 scope global dynamic wlan0
        valid_lft 34131sec preferred_lft 34131sec
```

In this example, the IP address is **192.168.1.230**.

## Start an SSH session

You should now be able to start an SSH session as `ubuntu`, if you want to continue from another desktop system. You can use `putty` to SSH in from a Windows PC, or just use the `ssh` command from a Linux or macOS terminal session.

## Continue Linux configuration

- This step changes the default editor to `vim`. If you are happy with the `nano` editor, you can skip this step. Enter the following command and choose **3** for `vim`.

```
$ sudo update-alternatives --config editor
```

- Change the `vim` color scheme because dark blue comments are difficult to read. Add the line `colorscheme desert` line as shown below.

```
$ sudo vi /etc/vim/vimrc
```

```
...
if has("syntax")
    syntax on
    colorscheme desert
endif
...
```

- Don't require passwords on `sudo` commands by adding `'NOPASSWD:'` to the group `sudo`.

```
$ sudo visudo
```

```
...
# Allow members of group sudo to execute any command
%sudo    ALL=(ALL:ALL) NOPASSWD: ALL
...
```

## Create user and group named `pi`

You can skip this step and continue to use the user `ubuntu`. However, it is recommended that you create a second user named `pi` as that is what has been tested with and documented in this manual.

- Add group and user `pi` both with IDs of 500. This is recommended by Mycroft.

```
$ sudo groupadd -g 500 pi
$ sudo useradd -u 500 -g 500 -s /bin/bash pi
```

- Add alternate groups to the `pi` user.

```
$ sudo usermod -G ubuntu,adm,cdrom,sudo,audio,video pi
$ id pi
uid=500(pi) gid=500(pi)
groups=500(pi),4(adm),24(cdrom),27(sudo),29(audio),44(video),1000(ubuntu)
```

- Make a home directory for the user `pi`.

```
$ cd /home
$ sudo mkdir pi
$ sudo chown pi.pi pi
```

- Set `pi`'s password to `'pi'` (or another password of your choice).

```
$ sudo passwd pi
New password:
Retype new password:
passwd: password updated successfully
```

- Switch to the user `pi`. The intermediate `'-'` is required to run the `.bash_profile`.

```
$ su - pi
Password:
...
```

All commands should be run as the user `pi`. If you “su to root” just to avoid prefixing many commands with `sudo`, strange things will happen (you have been warned :)).

## Continue configuration

- Create a file name `.bash_profile` in pi’s home directory. It is run when a shell is started.

```
# vi .bash_profile
alias la='ls -latr'
export JLOG="/var/log/jellyfin"
export LOG="/var/log/mycroft"
export
PATH="/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin:/snap/
bin:/home/pi/mycroft-core/bin"
export SKI="/opt/mycroft/skills"
export SYS="/etc/systemd/system"
export Z="/usr/local/sbin"
```

- The alias `la` lists files with details in reverse chronological order. That means the most recently modified files are shown at the bottom.
- The environment variable `JLOG` is a shortcut to the Jellyfin log files directory.
- The environment variable `LOG` is a shortcut to the Mycroft log files directory.
- The environment variable `PATH` includes the Mycroft executables in the executables search path.
- The environment variable `SKI` is a shortcut to the Mycroft skills directory.
- The environment variable `SYS` is a shortcut to the systemd directory where service configuration files are stored.

## Upgrade your system

- *Update* the system which prepares for the latest code for all installed packages.

```
$ sudo apt-get update
```

If you get the following error message, just wait for a while and issue the update command again.

```
Could not get lock /var/lib/dpkg/lock-frontend
```

- *Upgrade* your system. This step could take 10 - 15 minutes.

```
$ sudo apt-get upgrade
```

```
...
```

After this operation, 38.3 MB of additional disk space will be used.  
Do you want to continue? [Y/n] **y**

```
...
```

Your system should now be at the latest software levels.

## Install a Linux desktop

If you have the horsepower to create a general purpose computer with a modern desktop GUI, why not use it? The philosophy is to unleash a modern desktop GUI to this GNU/Linux system.

The following steps to install “Mate” are recommended. Here is a description:

“The MATE Desktop Environment is the continuation of GNOME 2. It provides an intuitive and attractive desktop environment using traditional metaphors for Linux and other Unix-like operating systems.”

Source:

<https://mate-desktop.org/>

If you are happy with a “headless” system, you can skip these steps, or you may choose to install a different Linux desktop.

To install the MATE desktop, perform the following steps.

- Install the `tasksel` manager, and some other packages.

```
$ sudo apt-get install tasksel jq
```

- Install the MATE desktop using the `tasksel` tool:

```
$ sudo tasksel install ubuntu-mate-desktop
```

This step can take up to a half hour.

- Reboot the system to test your changes.

```
$ sudo reboot
```

After rebooting, a *desktop* should come up on your monitor.

- Restart your SSH session as `pi`.

## Set the clock

Your clock is probably set to UTC (aka *Greenwich Mean Time*). To check and reset it, perform the following steps.

- Check your clock with the `date` command. Look for the UTC time zone.

```
$ date
Tue Sep 28 20:01:57 UTC 2021
```

- If the time is wrong, set the time zone.

```
$ sudo dpkg-reconfigure tzdata
... choose your time zone...
```

The `date` command should now show the correct time.

## Install other useful tools

The author of this guide, @mike99mac, has also written some small tools to help with the installation and testing of Mycroft and associated audio resources.

- Clone the `mycroft-tools` package in `pi`’s home directory with the following commands.

```
$ cd
$ git clone https://github.com/mike99mac/mycroft-tools.git
```

- Change to the newly installed directory and run the setup script.

```
$ cd mycroft-tools
$ sudo ./setup.sh
Checking for lsintent file ...
Copying all scripts to /usr/local/sbin ...
Making symbolic links in /usr/local/sbin ...
Success! There are new scripts in your /usr/local/sbin/ directory
```



- Show the files that were copied to `/usr/local/sbin/`.

```
$ cd /usr/local/sbin
$ la
-rwxr-xr-x 1 root root 192 Oct 10 08:09 testrecord
-rwxr-xr-x 1 root root 83 Oct 10 08:09 testplay
-rwxr-xr-x 1 root root 605 Oct 10 08:09 ltemp
-rwxr-xr-x 1 root root 2644 Oct 10 08:09 lspairingcode
-rwxr-xr-x 1 root root 6609 Oct 10 08:09 lsintent
lrwxrwxrwx 1 root root 8 Oct 10 08:09 lsvocab -> lsintent
lrwxrwxrwx 1 root root 8 Oct 10 08:09 lsskills -> lsintent
```

The scripts `lsintent`, `lsskills` and `lsvocab` list the Mycroft intents, skills and vocabularies installed on your system.

The scripts `testrecord` and `testplay` allow you to test your microphone and speakers. They will be used later.

The script `ltemp` shows you the current temperature of your Raspberry Pi.

- Install other packages.

```
$ sudo apt-get install mlocate ssh net-tools
```

The `mlocate` package gives you the `locate` and `updatedb` commands, the `ssh` package adds additional `ssh` related tools and `net-tools` adds additional networking commands such as `ifconfig`.

- Run the `updatedb` command.

```
$ sudo updatedb
```

With the `mlocate` database updated, you can issue the `locate` command and find any file or directory extremely fast.

## Turn off default sound card and turn on DAC HAT

Tell Ubuntu to load the HiFiBerry DAC HAT and to turn off the default Broadcom sound card by performing the following steps.

- Run the command `aplay -l`. This shows the hardware devices that can play sounds. The `bcm2835` values are the Broadcom chip on the Raspberry Pi.

```
$ aplay -l
**** List of PLAYBACK Hardware Devices ****
card 0: b1 [bcm2835 HDMI 1], device 0: bcm2835 HDMI 1 [bcm2835 HDMI 1]
  Subdevices: 4/4
    Subdevice #0: subdevice #0
    Subdevice #1: subdevice #1
    Subdevice #2: subdevice #2
    Subdevice #3: subdevice #3
card 1: Headphones [bcm2835 Headphones], device 0: bcm2835 Headphones
[bcm2835 Headphones]
  Subdevices: 4/4
    Subdevice #0: subdevice #0
    Subdevice #1: subdevice #1
    Subdevice #2: subdevice #2
    Subdevice #3: subdevice #3
```

- Change directory to `/boot/firmware/` and make a copy of the `usercfg.txt` file.

```
$ cd /boot/firmware
$ sudo cp usercfg.txt usercfg.txt.orig
```

- Edit the configuration file and add the lines shown in bold. The value `dtparam=audio=off` will turn off the audio on the broadcom chip, as it is of poor

quality. Note there are `dtoverlay` values for both the HiFiBerry and Inno-maker DAC hats. Choose the setting corresponding to the DAC Hat you will be using.

```
$ sudo vi usercfg.txt
# Place "config.txt" changes (dtparam, dtoverlay, etc.) in this
# file. Please refer to the README file for a description of various
# configuration files on the boot partition.
#
# Turn debug on
dtdebug=1
#
# Disable the built-in Broadcom sound card
dtparam=audio=off
#
# Enable the HiFiBerry DAC HAT
dtoverlay=hifiberry-dacplus,24db_digital_gain
# --OR--
# Enable the inno-maker DAC HAT
dtoverlay=allo-boss-dac-pcm512x-audio,24db_digital_gain

TODO: try these values recommended on:
# archimago.blogspot.com/2017/01/measurements-raspberry-pi-3-as-usb.html
# and see if there is a noticable difference in the sound
#arm_freq=800
#sdram_freq=400
#core_freq=400
#gpu_freq=300
#over_voltage=-4
#over_voltage_sdram=-4
#gpu_mem=16
```

- ▶ Reboot to test these changes.

```
$ sudo reboot
```

- ▶ When the system comes back up, start an SSH session as `pi`.
- ▶ List the devices again with the `aplay -l` command. Now the Broadcom device should be absent, and the hifiberry device is shown.

```
$ aplay -l
**** List of PLAYBACK Hardware Devices ****
card 0: sndrpihifiberry [snd_rpi_hifiberry_dacplus], device 0: HiFiBerry
DAC+ Pro HiFi pcm512x-hifi-0 [HiFiBerry DAC+ Pro HiFi pcm512x-hifi-0]
  Subdevices: 1/1
  Subdevice #0: subdevice #0
```

## Test microphone and speakers

It is important to know your microphone and speakers are working. The scripts `testrecord` and `testplay` previously installed utilize `arecord` and `aplay` to save and play audio to and from a file.

- ▶ To test your microphone, issue the following command then speak into the microphone for up to five seconds.

```
$ testrecord
Testing your microphone for 5 seconds - SAY SOMETHING!
Recording WAVE '/tmp/test-mic.wav' : Signed 16 bit Little Endian, Rate
16000 Hz, Mono
```

- ▶ Assuming that worked successfully, issue the following command to play back the recording.

```
$ testplay
```

```
Playing WAVE '/tmp/test-mic.wav' : Signed 16 bit Little Endian, Rate
16000 Hz, Mono
```

If you hear your voice played back, then your microphone and speakers are working.

If not, look at more options to the `aplay` and `arecord` commands.

## Install Jellyfin

Jellyfin is a music and video player that is a fork off of Emby. One positive of Jellyfin is that it does not require any password to play music, while Emby does. Also, Jellyfin seems to be more focused on Free and Open Source Software (FOSS) than Emby does.

We don't address video in this guide, but Jellyfin supports it and that may be a good addition to your entertainment environment.

To install Jellyfin, perform the following tasks.

- Get the Jellyfin *repo* (code repository).

**Note:** both lines are one command.

```
$ wget -O - https://repo.jellyfin.org/jellyfin_team.gpg.key | sudo
apt-key add -
...
Resolving repo.jellyfin.org (repo.jellyfin.org)... 68.183.204.194,
Connecting to repo.jellyfin.org ... connected.
HTTP request sent, awaiting response... 200 OK
...
```

- Run this long command to create a file which will make Jellyfin a source  
**TODO:** expound on this....

**Note:** All 4 lines are one command.

```
$ echo "deb [arch=$( dpkg --print-architecture )]
https://repo.jellyfin.org/${ awk -F=' ' /^ID=/{ print $NF }'
/etc/os-release ) $( awk -F=' ' /^VERSION_CODENAME=/{ print $NF }'
/etc/os-release ) main" | sudo tee /etc/apt/sources.list.d/jellyfin.list
```

- Update the system. Because you added Jellyfin as a source, this command will pull in all necessary packages.

```
$ sudo apt update
...
```

- Install Jellyfin.

```
$ sudo apt install jellyfin
...
```

- Set Jellyfin to start at boot time.

```
$ sudo systemctl enable jellyfin
Synchronizing state of jellyfin.service with SysV service script with
/lib/systemd/systemd-sysv-install.
Executing: /lib/systemd/systemd-sysv-install enable jellyfin
```

## Allow music to be read from a USB drive

A good source of music input is a USB drive with music in the form of MP3 or other music format files. If you have a USB drive with music on it, perform the following steps to have it mounted at GNU/Linux boot time. If you don't have one, you can skip this section, but you will need at least one source of music.

- Before you plug the USB drive into the RasPi USB jacks, check if there is anything mounted.

```
$ mount | grep sda
```

If you see no output, that means there is nothing mounted with the string 'sda' in it.

- Plug the USB drive with music on it into the RasPi. Then run the same `mount` command again. You should now see the USB drive device file.

```
$ mount | grep sda
/dev/sda1 on /media/pi/0011-5B59 type vfat (rw,nosuid,nodev,...)
$ file /dev/sda*
/dev/sda: block special (8/0)
/dev/sda1: block special (8/1)
```

This shows that the USB drive was given the device name of `/dev/sda` and the first partition is `/dev/sda1`.

- Unmount the drive so a better mount point can be chosen.

```
$ sudo umount /media/pi/0011-5B59
```

- Make a directory to mount USB drive over. In this example it is `/home/pi/usbdrive/`.

```
$ cd
$ mkdir usbdrive
```

Now the USB drive can be mounted over that directory.

- Make a backup copy of the file `/etc/fstab`.

```
$ cd /etc
$ sudo cp fstab fstab.orig
```

This is done because if you ever screw up `/etc/fstab`, your system may not be able to boot. Having a backup copy makes it easier to recover your system.

- Add one line to the bottom of `/etc/fstab` that will mount the USB drive over the directory you just created.

```
$ sudo vi fstab
LABEL=writable / ext4 defaults 0 0
LABEL=system-boot /boot/firmware vfat defaults 0 1
/dev/sda1 /home/pi/usbdrive vfat defaults 0 0
```

- Mount the file system by providing just one argument to the `mount` command. This effectively tests the change to the `/etc/fstab` file.

```
$ sudo mount /home/pi/usbdrive
```

- Show the contents of the USB drive.

```
$ ls -l /home/pi/usbdrive
drwxr-xr-x 2 root root 16384 Nov 2 2015 'System Volume Information'
drwxr-xr-x 222 root root 32768 May 9 2011 music
```

This shows a directory named `music` under which there are mp3 files.

- Reboot the system to test that the drive is mounted at boot time.

```
$ sudo reboot
```

- When the system comes back up, start an SSH session as `pi` and check that the USB drive is mounted and Jellyfin is running.

```
$ mount | grep usbdrive
/dev/sda1 on /home/pi/usbdrive type vfat (rw,...)
```

```
$ service jellyfin status
Warning: The unit file, source configuration file or drop-ins of
jellyfin.service changed on disk. Run 'systemctl daemon-reload' to
reload>
? jellyfin.service - Jellyfin Media Server
   Loaded: loaded (/lib/systemd/system/jellyfin.service; enabled;
   vendor preset: enabled)
   Drop-In: /etc/systemd/system/jellyfin.service.d
            +-jellyfin.service.conf
   Active: active (running) since ...
```

### 1.8.1 Configure Jellyfin

Jellyfin should now be running and listening on port 8096. When you first access it through a browser it will take you through some setup steps. To configure Jellyfin, perform the following steps.

- ▶ Login as `pi` to the MATE desktop on the locally connected monitor, keyboard and mouse.
- ▶ There should be an orange and blue Firefox icon at the top of the screen. Bring up a browser and specify the following URL.  
`http://localhost:8096`
- ▶ You should see a page with *Welcome to Jellyfin!* as shown in the following figure. Select your language and click **Next**.

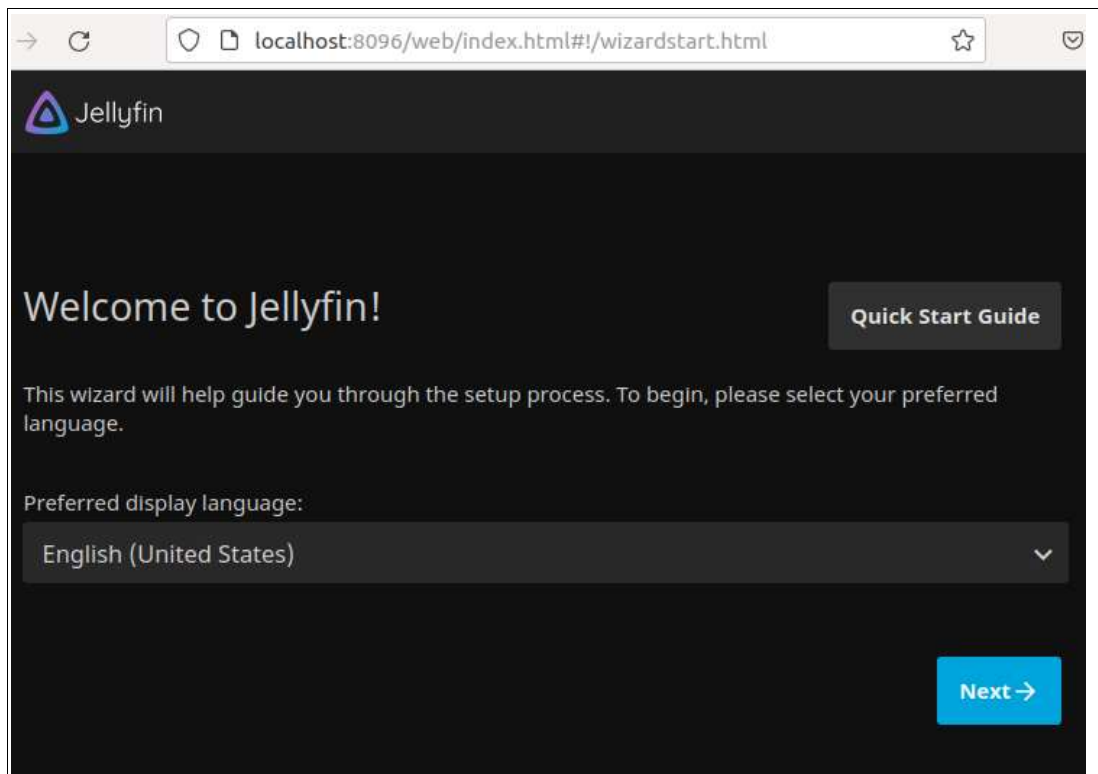


Figure 1-1 First Jellyfin customization page

- You should see a page with Tell us about yourself as shown in the following figure. Enter **pi** as the *username*, but do not add a password. Click **Next**.

← → ↻ 🛡️ 📄 🌐 localhost:8096/web/index.html#!/wizarduser ☆ 📧 ☰

← Jellyfin

## Tell us about yourself

Jellyfin includes support for user profiles with granular display settings, play state, and parental controls.

Username:

Please select a username for the admin account.

Password:

You can leave this field blank to set no password.

Password (confirm):

More users can be added later from within the dashboard.

← Previous Next →

Figure 1-2 Setting Jellyfin user name

- You should see a page with the title *Setup your media libraries*. Click the light blue '+' sign in *Add Media Library*.

- In the *Content type* dropdown menu, choose **Music**. Then you should see a page as shown in the following figure. Enter your Display name, language and country, then click the '+' sign next to *Folders*.

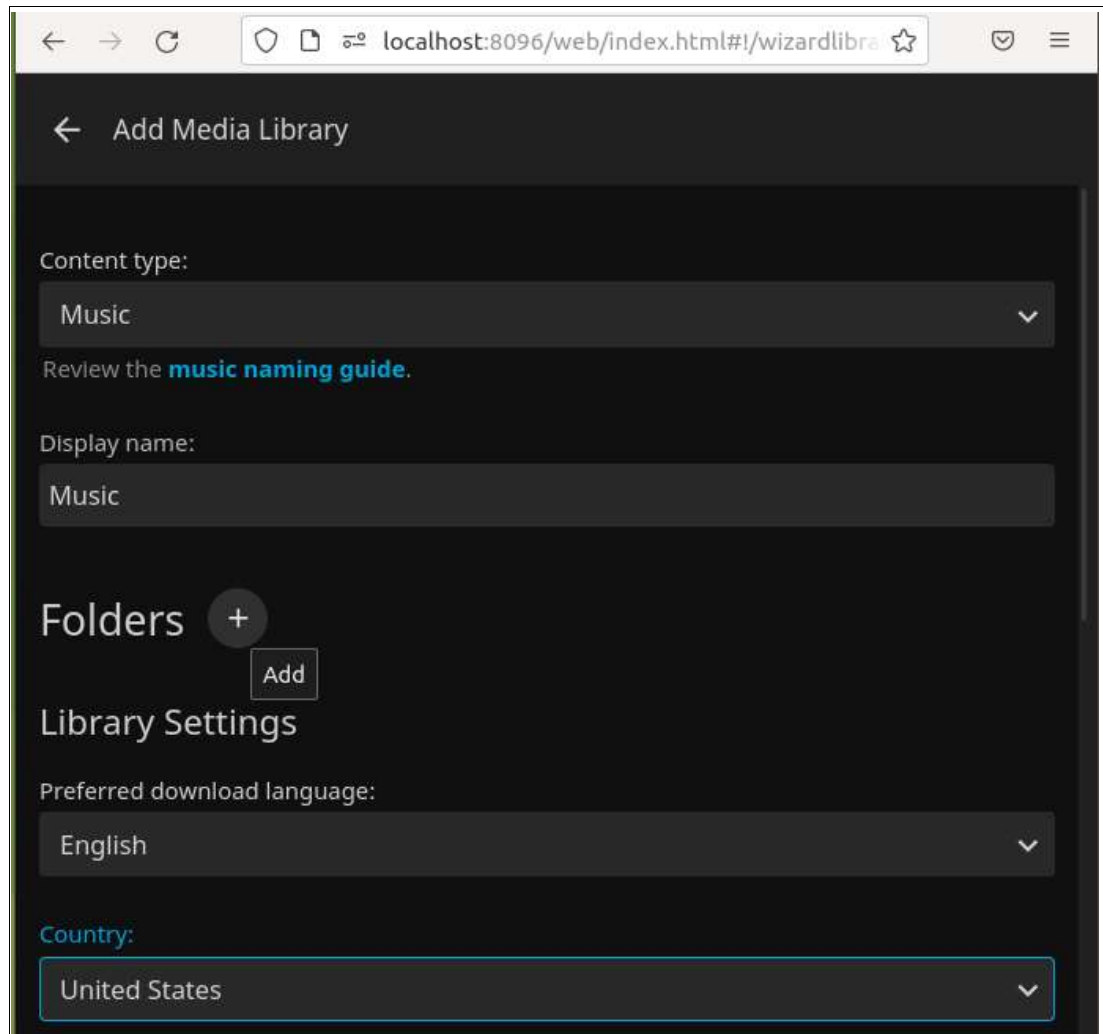


Figure 1-3 Setting media type to Music

- ▶ You should see a *Folder* text box as shown in the following figure. Enter the directory `/home/pi/usbdrive/music` and click **Ok**.

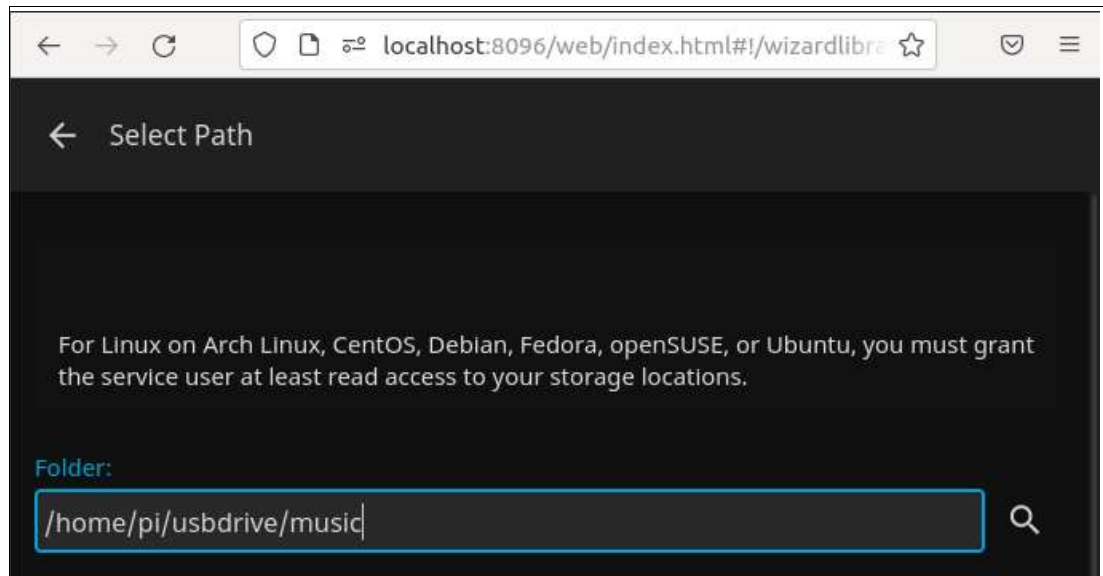


Figure 1-4 Specifying the directory with music

- ▶ You should be returned to the previous page. At the bottom of the page, click the checkbox *Save artwork into media folders*.
- ▶ Click **Ok** at the bottom of the page.
- ▶ You should be returned to page *Setup your media libraries* and see the directory you just added. Click **Next**.
- ▶ You should see a page *Preferred Metadata Language*. Verify the values are correct and click **Next**.
- ▶ You should see the page *Configure Remote Access*. Accept the defaults of allowing remote access, but not automatic port mapping and click *Next*.
- ▶ You should see a page *You're Done!*. Click **Finish**.

## Using Jellyfin

To use Jellyfin, perform the following steps.



- You should see a page Please sign in as shown in the following figure. Enter the User **pi** with no password and click **Sign In**.

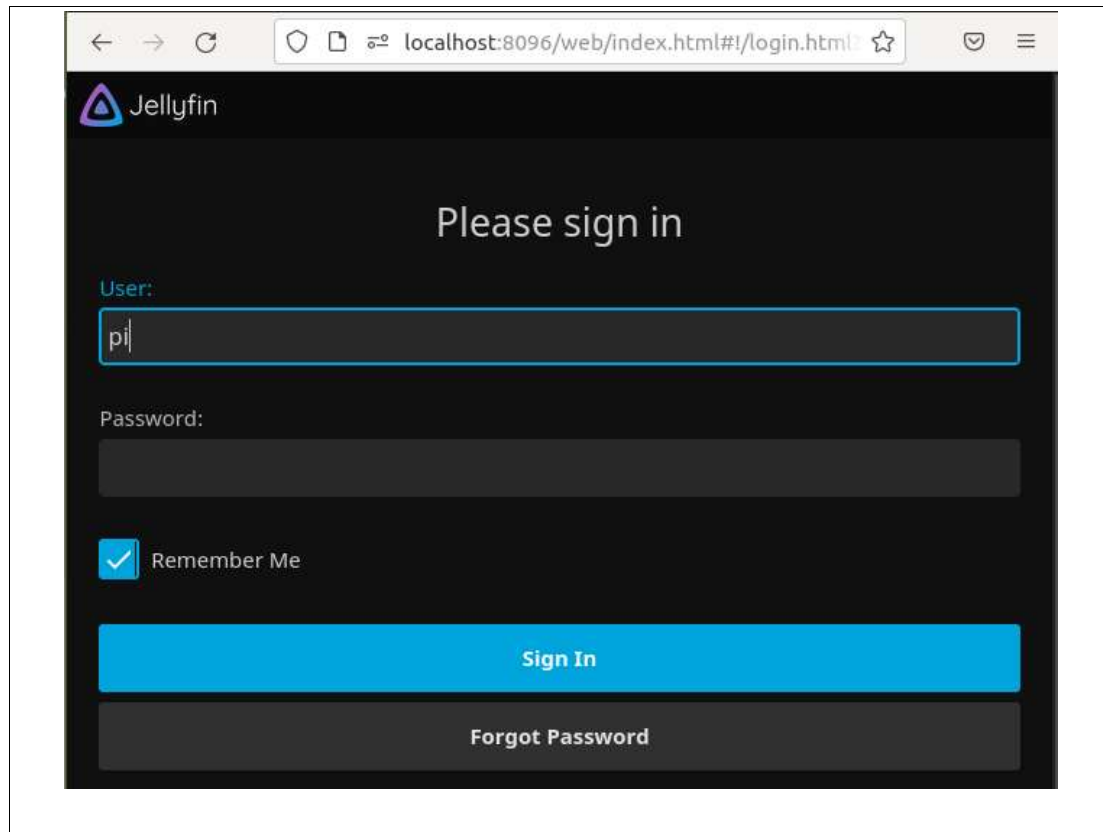


Figure 1-5 Signing in to Jellyfin

- You should see the Jellyfin home page as shown in the following figure. It may take some time to index your music, but you should see Albums with artwork.

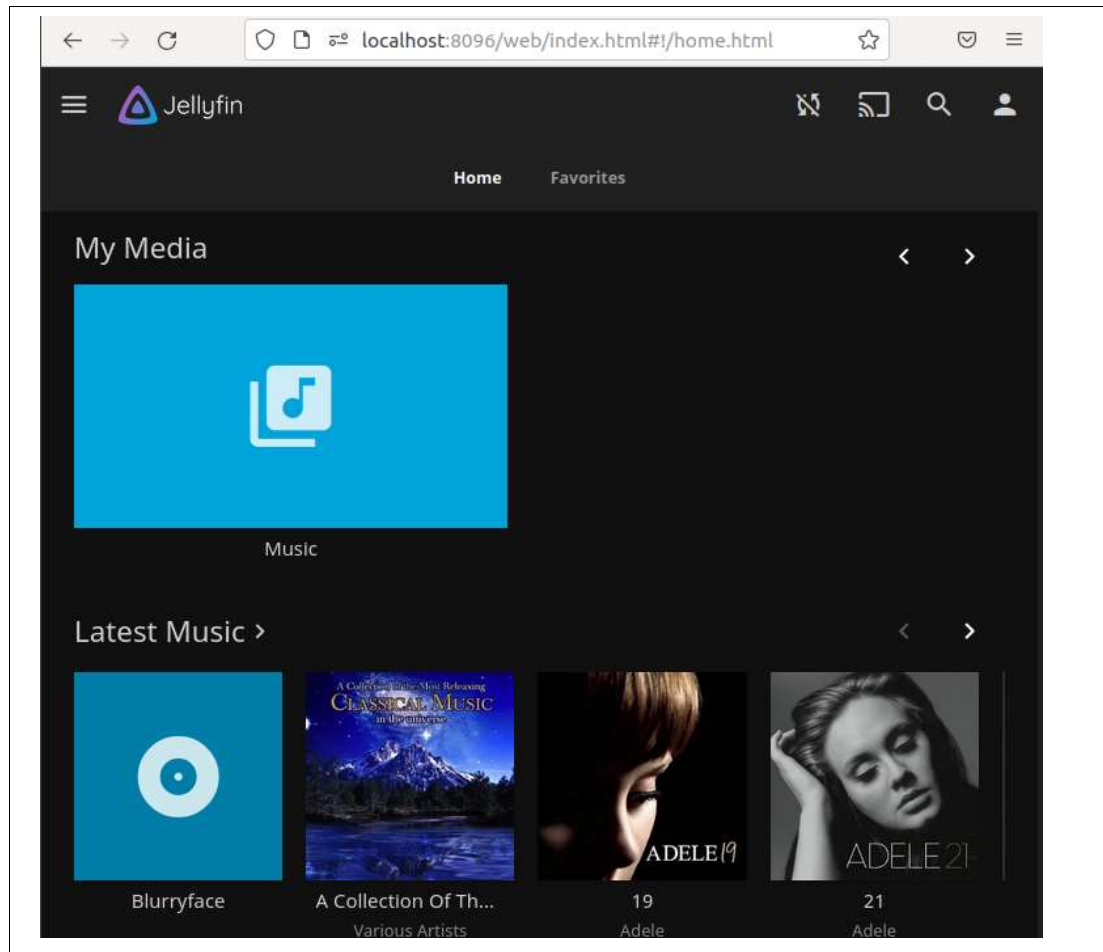


Figure 1-6 Accessing Jellyfin music

- You should be able to click on Albums and then songs and start playing music!

## Update the pi user

Now that pulseaudio and Jellyfin are installed, those groups can be added to the pi user.

- Add `pulse` and `jellyfin` as alternate groups to the user `pi` with the following command.

```
$ sudo usermod -G pulse,jellyfin,ubuntu,adm,cdrom,sudo,audio,video pi
```

## Install Mycroft

To install Mycroft, perform the following steps.

- Change directory to `/home/pi/.`

```
$ cd
```

- Get the Mycroft code from [github.com](https://github.com/MycroftAI/mycroft-core).

```
$ git clone https://github.com/MycroftAI/mycroft-core.git
Cloning into 'mycroft-core'...
remote: Enumerating objects: 553610, done.
```

```
remote: Counting objects: 100% (1587/1587), done.
remote: Compressing objects: 100% (830/830), done.
...
```

- Run the `dev_setup.sh` script in the `mycroft-core/` directory.

```
$ cd mycroft-core
$ bash dev_setup.sh
```

... answer **Yes** to all questions ...

This step could take up to a half hour.

## Add Jellyfin skills

Now that Mycroft is installed, the Jellyfin skills can be added.

- Populate the Jellyfin skills with the following commands.

```
$ cd /opt/mycroft/skills
$ sudo git clone https://github.com/tuxfoo/jellyfin-skill.git
...
```

- Verify the Jellyfin skills have been added to the Mycroft skills list.

```
$ lsvocab | grep jelly
check (emby|mb|mv|nv|emily|jelly fin|jelly fan|jelly fen)
check (emby|mb|mv|nv|emily|jelly fin|jelly fan|jelly fen) connection
play (artist|band) {artist} from (emby|mv|mb|jelly fin|jelly fan|jelly
fen)
play album {album} from (emby|mv|mb|jelly fin|jelly fan|jelly fen)
play playlist {playlist} from (emby|mv|mb|jelly fin|jelly fan|jelly fen)
play {media} from (emby|mv|mb|jelly fin|jelly fan|jelly fen)
run diagnostic (emby|mb|mv|nv|emily|jelly fin|jelly fan|jelly fen)
```

The output shows all the skills with the word “Jelly” in them.

## Set Mycroft to start at boot time

To set Mycroft to start when the Raspberry Pi boots, perform the following steps.

- Change to the systemd configuration directory.

```
$ cd $SYS
$ pwd
/etc/systemd/system
```

- Create the file `mycroft.service` with the following lines to start Mycroft at boot time.

```
$ sudo vi mycroft.service
[Unit]
Description=Mycroft AI
After=pulseaudio.service

[Service]
User=pi
WorkingDirectory=/home/pi/
ExecStart=/home/pi/mycroft-core/bin/mycroft-start all
ExecStop=/home/pi/mycroft-core/bin/mycroft-stop
Type=forking
Restart=no
Environment="XDG_RUNTIME_DIR=/run/user/500"

[Install]
WantedBy=multi-user.target
```

- ▶ Enable the service to start at boot time.

```
# sudo systemctl enable mycroft.service
Created symlink
/etc/systemd/system/multi-user.target.wants/mycroft.service ...
```

- ▶ Reboot the system to test these changes.

```
$ sudo reboot
```

- ▶ Ideally Mycroft will start talking to you. If it does, it will tell you the six digit pairing code. Write it down.
- ▶ If Mycroft is not speaking to you, there is some sort of issue.
- ▶ When the system comes back up, start an SSH session and issue the following **service** command. Look for a green “active (running)” message.

```
$ service mycroft status
? mycroft.service - Mycroft AI
   Loaded: loaded (/etc/systemd/system/mycroft.service; enabled; vendor
   preset: enabled)
   Active: (running) since Sat 2021-10-02 08:06:31 EDT; 1min 53s ago
```

## Register your device with Mycroft

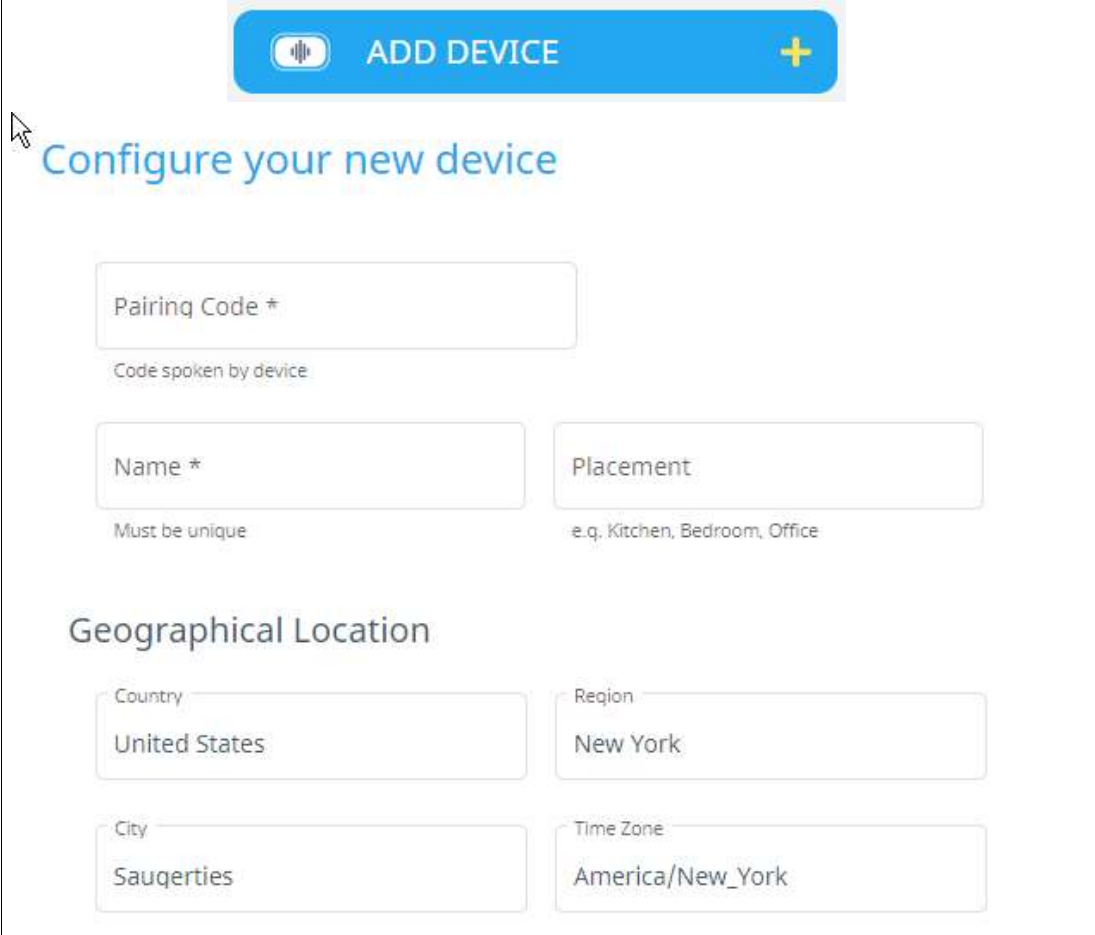
You must register your device with Mycroft. To do so, perform the following steps.

- Point a browser to the following Mycroft website.

<https://account.mycroft.ai/devices>

You should see a page with a blue box at the top

- Click the **ADD DEVICE** button as shown in the top of the following figure.
- You should see a page similar to the following figure. Supply the pairing code and a description of the device and its location.



The screenshot shows a web interface for configuring a new Mycroft device. At the top, there is a blue button labeled "ADD DEVICE" with a microphone icon and a plus sign. Below this, the heading "Configure your new device" is displayed. The form consists of several input fields: a "Pairing Code \*" field with a hint "Code spoken by device"; a "Name \*" field with a hint "Must be unique"; a "Placement" field with a hint "e.g. Kitchen, Bedroom, Office"; a "Geographical Location" section containing four fields: "Country" (with "United States" entered), "Region" (with "New York" entered), "City" (with "Sauqerties" entered), and "Time Zone" (with "America/New\_York" entered).

Figure 1-7 Adding a device to the Mycroft registry

- Click **Next**.
- You should see a page with the heading *Your Device is Ready*. Click **Finished**.
- Very soon after you should hear Mycroft speak "Now I am ready for you" .

account.mycroft.ai/devices/add

FT AI

1 Device 2 Done!

### Configure your new device

Pairing Code \*  
JHP94A  
Code spoken by device

Name \*  
Boombox with two jacks  
Must be unique

Placement  
Office  
e.g. Kitchen, Bedroom, Office

#### Geographical Location

Country  
United States

Region  
New York

City  
Sauqerties

Time Zone  
America/New\_York

Figure 1-8

- Click **Next**.

You should see a page with the message *"Your device is ready!"*.

## Generate an API key

For Mycroft to be able use Jellyfin as a music player, it must have an API key. This is effectively a password for Jellyfin. To do this perform the following steps.

- ▶ Point a browser to the Jellyfin server `http://server.ip.address:8096/`
- ▶ You will be challenged for credentials. Enter `pi` as the user with no password.
- ▶ Click the *hamburger menu* in the upper left corner.
- ▶ Then click **Dashboard** under the **Advanced** section, click **API Keys**. A new page should be drawn to the right of the menu as shown in the figure below.
- ▶ Click the '+' sign next to the **API Keys** heading. A new small window should pop up in the center.
- ▶ Enter an **App name** such as 'Mycroft' and click **Ok**. An API key should be added.
- ▶ Copy the key into your clipboard by highlighting it and pressing **Ctrl-C**.

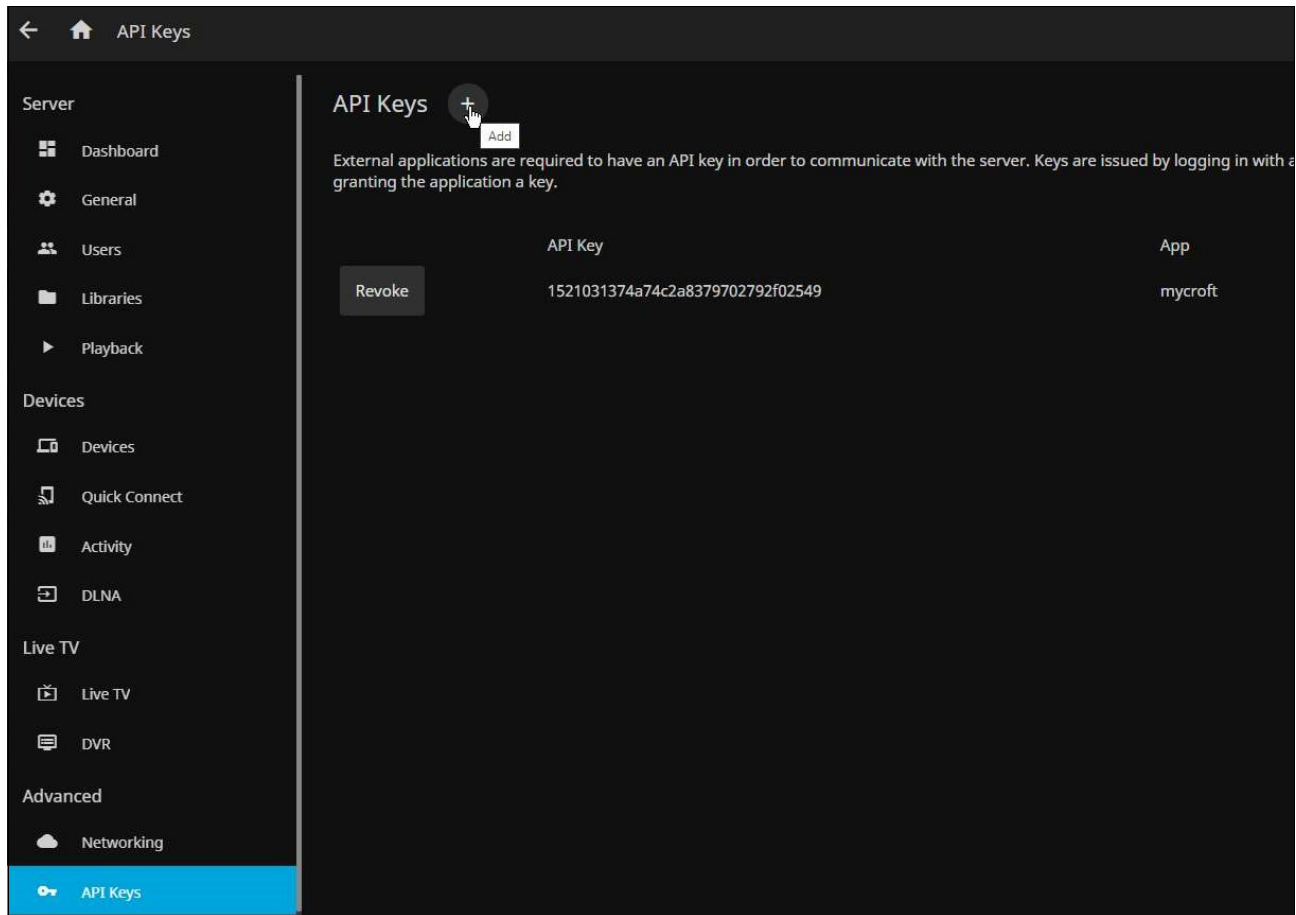
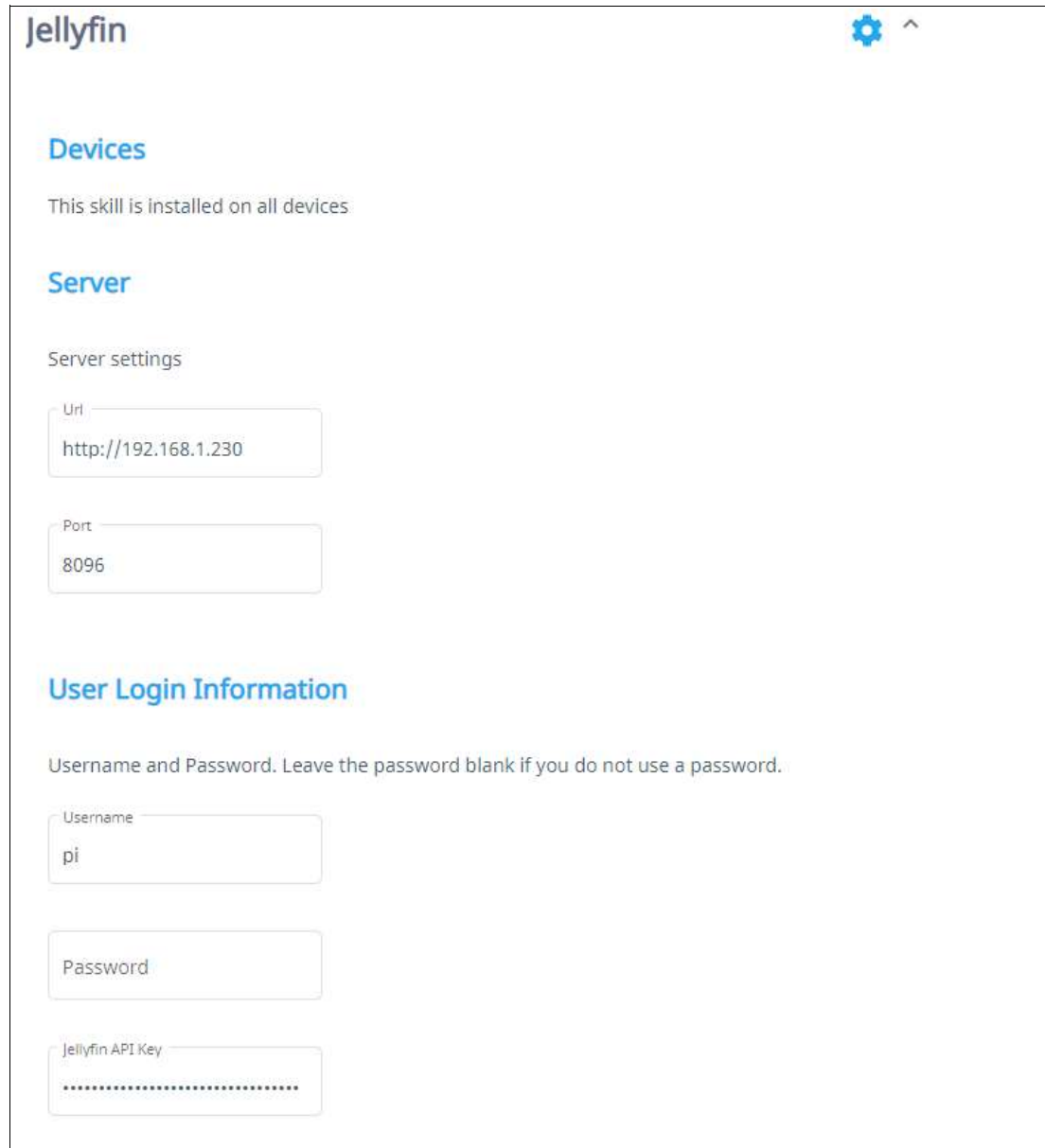


Figure 1-9 Generating an API key for mycroft

- ▶ Copy the key to your Mycroft skill page. Point a browser to:  
`https://account.mycroft.ai/skills`
- ▶ Click on the jellyfin skill. You should see a page similar to the following.
- ▶



The image shows the Jellyfin web interface. At the top left is the 'Jellyfin' logo. At the top right is a gear icon for settings and an upward arrow. Below the logo is a 'Devices' section with the text 'This skill is installed on all devices'. Below that is a 'Server' section with the heading 'Server settings'. There are three input fields: 'Url' with the value 'http://192.168.1.230', 'Port' with the value '8096', and 'User Login Information'. The 'User Login Information' section has a note: 'Username and Password. Leave the password blank if you do not use a password.' It contains three input fields: 'Username' with the value 'pi', 'Password' (empty), and 'Jellyfin API Key' (filled with dots).

Figure 1-10



Figure 1-11

## 1.8.2 Use Pandora music service

**TODO:** Installing Pandora seems to break Jellyfin - adding the 'from jellyfin' suffix doesn't fix it

To use the music service, perform the following steps.

- Point your browser to the Mycroft skills URL.

`https://account.mycroft.ai/skills`



- Open the skill

### 1.8.3 Use Spotify music service

To use the Spotify music service, perform the following steps.

- Point your browser to the Mycroft skills URL.

```
https://account.mycroft.ai/skills
```

- Open the skill

```
Adding other skills:
it clone https://github.com/MycroftAI/skill-pandora
git clone https://github.com/andlo/count-skill
git clone https://github.com/pcwii/skill-pick-number
git clone https://github.com/Dragoncraft89/dice-skill
git clone https://github.com/padresb/bark-skill
git clone https://github.com/MycroftAI/skill-stop
git clone https://github.com/MycroftAI/fallback-duckduckgo
git clone https://github.com/MycroftAI/skill-query

-----
Testing internet radio:
git clone https://github.com/normandmickey/skill-internet-radio

==> It appears this skill does not work and is no longer being maintained
-----

... Trying "Tune In" for Internet radio stations

$ cd ~/mycroft-core
$ source .venv/bin/activate
(.venv) $ msm install https://github.com/johnbartkiw/mycroft-skill-tunein

TODO: easier - try
mycroft-msm install https://github.com/johnbartkiw/mycroft-skill-tunein

https://sourceforge.net/projects/win32diskimager/

-----
Ideas on selling graduated kits

SmartBoombox 0.2 - just the wood to assemble an enclosure - you glue up
and supply all hardware
SmartBoombox 0.4 - an assembled enclosure - you supply all hardware, but
no gluing
SmartBoombox 0.6 - an assembled enclosure and all audio components - does
not have to be smart
SmartBoombox 0.8 - an assembled enclosure and all parts to construct a
smart boombox
SmartBoombox 1.0 - a fully-assembled smart boombox

-----
TODO: look into "Share a Cart" Chrome extension to share an Amazon
shopping cart

-----
Plywood replacement - avoid extra gluing
Baltic Birch 12mm x 24" x 30"
```

<https://www.woodcraft.com/products/baltic-birch-plywood-12-mm-1-2-x-24-x-30>

All software and hardware should now be installed and ready to test

## 1.9 Test and enjoy your boombox

### 1.9.1 Testing the basic boombox

If you built the basic boombox, testing is much simpler. For music input you have a choice of using the aux-in jack or a bluetooth connection.

### 1.9.2 Testing the smart boombox

Start with `testrecord` and `testplay`.

#### Questions?

If you have any questions, feel free to email [mike99mac@gmail.com](mailto:mike99mac@gmail.com)

