



# Raspberry Pi WiFi Radio

Created by Phillip Burgess



<https://learn.adafruit.com/pi-wifi-radio>

Last updated on 2021-11-15 05:54:43 PM EST

# Table of Contents

Overview	3
Parts List	4
• Required Parts:	4
• Optional but Recommended:	5
• Temporary Items for Setup:	5
Initial System Configuration	5
• Raspberry Pi Downloads Page	6
• Enable I2C Support for the LCD	8
• Optional: Enable USB Audio	8
• Reboot	9
Install Software Packages	10
• Update Installed Software	10
• Install Prerequisite Software	10
Configure Software Packages	12
LCD and Final Configuration	13
• Final Configuration and Auto-Start	16
Troubleshooting	17

---

# Overview

PLEASE NOTE: this guide is now deprecated...

- It relied on some extremely finicky software (pianoBar and libav) which tended to break with each new operating system release...it's a minor miracle we've even managed to patch it this far along!
- It relied on several now-deprecated Python libraries for the display (Adafruit\_CharLCDPlate, Adafruit\_I2C, etc.).
- It's specific to the Pandora streaming service, which is only available in the United States and a couple other countries.

That being said, if you're in the US, already have the hardware sitting around and don't mind sinking 30 minutes into a project that might not pan out, by all means give it a shot. It was pretty nifty when it worked. Please just understand that we will no longer be providing technical support for this project nor any more updates to this guide. It was last seen working with the 2018-11-13 release of Raspbian Stretch Lite, if that's any help in recreating the project.

There are other Pi audio guides available in the Adafruit Learning System that might serve you better nowadays:

- [Raspberry Pi Zero NPR One Radio \(https://adafru.it/DJw\)](https://adafru.it/DJw)
- [Raspberry Pi radio player with touchscreen \(https://adafru.it/jBA\)](https://adafru.it/jBA)
- [Boomy Pi Airplay Boombox \(https://adafru.it/DJx\)](https://adafru.it/DJx)



Raspberry Pi, the little wonder-puter that's taken the world by storm, is so affordable that we can create nifty single-purpose “appliances” around them without shame. Here's our take on one of the more popular such applications: internet streaming media, the Pandora music service specifically.

Pandora is now limited to users in the United States. It is no longer available in other countries. This guide is now deprecated and will not be updated for alternate services.

With the addition of a small LCD, a few buttons and a USB wireless network adapter, the Raspberry Pi becomes an affordable self-contained music streamer that can be moved to any room of the house...wherever you need your tunes at the moment. Just connect power and speakers or headphones.

---

## Parts List

### Required Parts:

- [Raspberry Pi \(http://adafru.it/998\)](http://adafru.it/998) board (any model or revision). This project has modest requirements and is a great use for older “hand-me-down” Pi boards!

- Adafruit LCD+Keypad Kit for Raspberry Pi. There are a few different versions of this, depending on the “look” you’re after: [RGB Positive backlight \(http://adafruit.it/1109\)](http://adafruit.it/1109), [RGB Negative \(http://adafruit.it/1110\)](http://adafruit.it/1110), and the more affordable [Blue & White \(http://adafruit.it/1115\)](http://adafruit.it/1115). All versions of this kit require some assembly and soldering.
- A 2GB or larger [microSD card \(https://adafruit.it/Dvm\)](https://adafruit.it/Dvm) (older Pi models use full-size SD, or use an adapter).
- A power supply compatible with the USB MicroB connector on the Raspberry Pi; some mobile phone chargers will work, or we offer a suitable [USB “wall wart” adapter \(https://adafruit.it/e5A\)](https://adafruit.it/e5A).
- Headphones or amplified speakers (or connect to your living room A/V setup).
- A free account with the [Pandora \(https://adafruit.it/c6j\)](https://adafruit.it/c6j) streaming music service.

Any model of Raspberry Pi prior to the Pi 3 or Pi Zero W also requires:

- A [USB WiFi adapter \(http://adafruit.it/814\)](http://adafruit.it/814) compatible with the Raspberry Pi. Many Pi models also come with wired Ethernet, but that’s less convenient.

## Optional but Recommended:

- Any Raspberry Pi [case \(https://adafruit.it/Dvn\)](https://adafruit.it/Dvn) with an open top (the Pibow and Pi Shell cases aren’t suitable — we need full access to the top of the board).

## Temporary Items for Setup:

The following are required for assembly and configuration, but do not need to remain permanently attached:

- Monitor (HDMI or composite)
- USB keyboard
- Possibly a powered USB hub
- Soldering iron and solder (for assembling the LCD keypad kit)

---

# Initial System Configuration

This project works well with the Raspbian “Lite” operating system; it fits easily on a 2 GB card with room to spare. Do not use the “desktop” images, they’re enormous and contain a ton of things we don’t need for this.



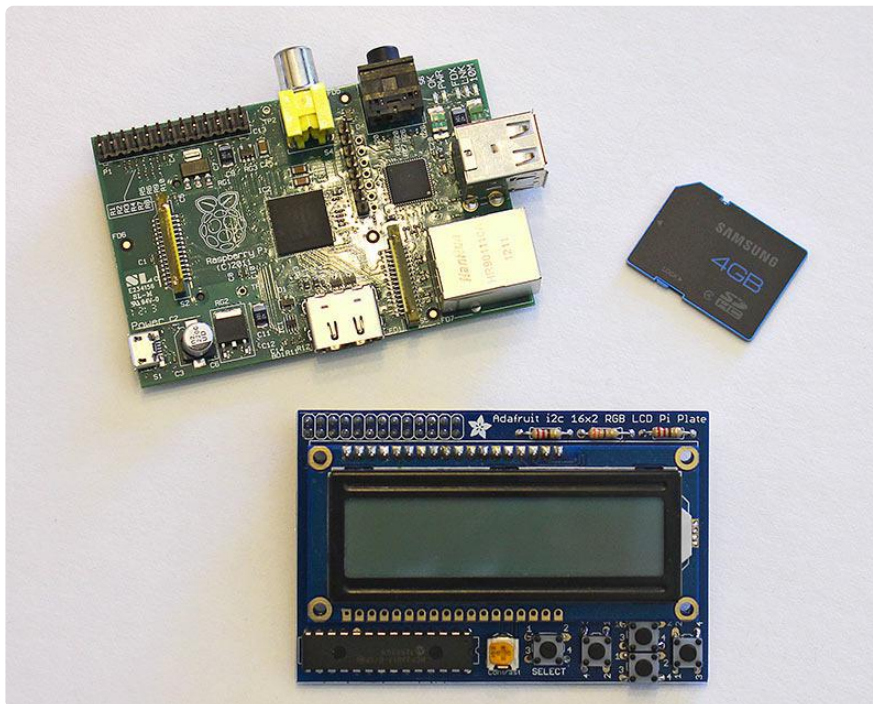
## [Raspberry Pi Downloads Page \(https://adafru.it/fQi\)](https://adafru.it/fQi)

While that's downloading, you can work on [assembling the LCD kit \(https://adafru.it/jBz\)](https://adafru.it/jBz).

If you're new to Raspberry Pi and Linux, we strongly suggest working through the first few guides in the [Learn Raspberry Pi \(https://adafru.it/dpe\)](https://adafru.it/dpe) tutorial series...know how to "burn" an SD image, perform a first-time setup and get the Raspberry Pi connected to a network. Some familiarity with one of the text editors (such as the simple nano or the more daunting vi or emacs) is also recommended.

Install the Raspbian Lite OS on a 2GB or larger card. You'll need to connect a monitor and USB keyboard for basic system configuration, but this is only temporary...we'll set it up to run "headless" later. For networking, connect either an Ethernet cable or use WiFi (built-in on some Pi models, else use a USB WiFi adapter).

At this point you should have an SD card containing the Raspbian Lite software, and an assembled LCD "Pi Plate."



You don't need to install the LCD Plate atop the Pi yet, we'll do that later. Let's just get the basics set up.

1. Connect a monitor and keyboard to the Raspberry Pi.
2. Insert the SD card containing the Raspbian OS.
3. Connect a "Micro B" USB cable to the power connector on the Raspberry Pi.

4. Plug the other end of the USB cable into a power source: a mobile phone charger, a powered USB hub, or simply a USB port on your computer.

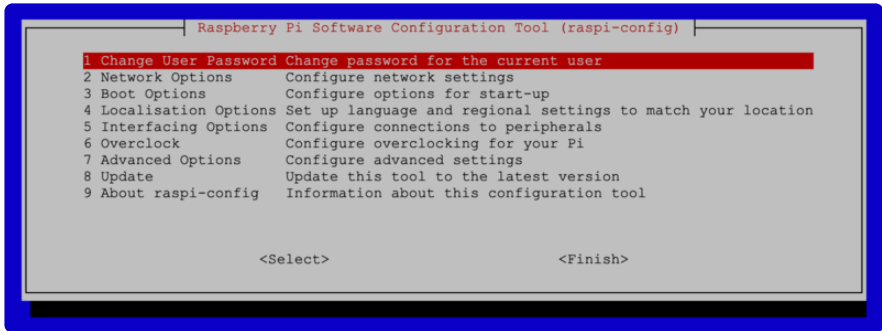
The Raspberry Pi should now boot, and you'll see the monitor fill with lots of "Unix stuff."

Linux can be daunting to the uninitiated. Don't worry about messing something up...if all else fails, you can re-format the SD card and begin again.

On first boot, you'll get a login prompt. Log in as user "pi," password "raspberrypi".

Once you're logged in successfully and have a command-line prompt, run the system configuration utility:

```
sudo raspi-config
```



The following selections are required:

- Under "Interfacing Options," enable I2C.
- If you'll be using WiFi, under "Localisation Options," select "Change Wi-fi Country." Then, under "Network Options," select "Wi-fi" to enter the SSID/ password for your wireless network.

The following are optional, but strongly recommended:

- Change User Password (because everybody knows the default).
- Under "Localisation Options," select "Change Locale," "Change Timezone" and "Change Keyboard Layout" to suit your location. If your keyboard isn't producing the expected symbols, this is why.
- Network Options→Change Hostname. I named mine "pandora" to distinguish it from other Raspberry Pi systems on the network.

- Interfacing Options→Enable SSH. This allows remote login from another system on the network, for performing administration tasks without a display attached.
- You can disable Overscan (under “Advanced Options”) if you like — we’ll reboot a few times during the setup process, and this provides a little extra screen real estate on HDMI monitors.

Tab over to the “Finish” button, press Return and confirm you’d like to reboot the system when prompted. You’ll need to log in again, using the password you established above.

Several configuration files will now be edited. Our examples show the “nano” text editor being used, as it’s easiest for the uninitiated. Power users can substitute “vi” or their editor of choice.

If using a USB WiFi adapter based on the popular Realtek 8192CU chipset, disabling WiFi power management seems to help with reliability. Once you’re logged in, type (or copy and paste) this at the command prompt:

```
echo "options 8192cu rtw_power_mgnt=0 rtw_enusbss=0" | sudo tee --append /etc/modprobe.d/8192cu.conf
```

## Enable I2C Support for the LCD

The LCD doesn’t need to be attached yet, but we can get some setup started…

```
sudo nano /etc/modules
```

Add one or both of these lines if not already present:

```
i2c-bcm2708  
i2c-dev
```

## Optional: Enable USB Audio

If you plan to use this with a USB Audio Adapter, [this guide explains the process \(http://adafru.it/cSk\)](http://adafru.it/cSk). It’s pretty straightforward, a matter of editing one line in a file.



# Reboot

Finally, shut down the system:

```
sudo shutdown -h now
```

Wait for the system to report that it's halted before disconnecting power. It should take about 20 seconds.

Following shutdown, insert the USB WiFi adapter (if using one) and re-connect power (keep the monitor attached for now).

Is using a Model A board: you'll need to remove the keyboard to plug in WiFi, or use a powered USB hub temporarily during setup. Without a hub, ssh is now the only way into the system...so if WiFi isn't working, you'll need to unplug the adapter, connect a keyboard and check the WiFi configuration.

For all other boards: keyboard can stay attached until you know the networking is properly configured. Log in again and you should be able to access the outside world now:

```
sudo ping adafruit.com
```

Do not proceed until internet access is working. If WiFi refuses to cooperate, make sure every step above has been followed carefully. If you're trying to use a hidden network and it just won't play nice, change the router configuration to broadcast the network name.

Raspbian Lite includes the "avahi" package by default, so the system appears on the network as "pandora.local" (or whatever hostname you configured) instead of a numeric IP address. You can easily access the Raspberry Pi remotely using an ssh client from another system on the network. For example, using the Terminal application in Mac OS X, one would type:

```
ssh pi@pandora.local
```

You should get a password prompt. Once logged in, you can perform all administration duties remotely (including the steps that follow), and the monitor and keyboard are no longer needed on the Raspberry Pi. It's easier this way because you can cut-and-paste all the commands with their weird syntaxes.

---

# Install Software Packages

Wireless networking **MUST** be working before continuing. If your Pi is not on the network yet, work through the prior page to diagnose the issue.

Further configuration of the Raspberry Pi will take place over the network using SSH, not the keyboard and mouse.

- The SSH server should already be enabled on the Raspberry Pi — this was done during the initial raspi-config setup.
- On Mac or Linux systems, you can use a Terminal or xterm window.
- For Windows systems, you can [download PuTTY \(https://adafru.it/aYY\)](https://adafru.it/aYY).

The terminal command to access the system would be:

```
ssh pi@pandora.local (or whatever hostname was configured) if using Netatalk
```

or:

```
ssh pi@192.168.0.42 (or whatever the system's actual numeric IP address is)
```

You'll be prompted for a password — use the password that you set up in raspi-config, or “raspberry” if you left the default. Additionally, the first time connecting you may be prompted regarding a host key for security...enter Y (or click Yes) when prompted.

## Update Installed Software

Once logged in, type the following at the command prompt:

```
sudo apt-get update
```

This updates the list of available software packages, and takes a couple of minutes (just do the “update,” not “upgrade” — the latter can sometimes take hours!)

## Install Prerequisite Software

Several prerequisite software packages need to be installed, using different techniques.

First is a collection of code libraries, using the apt-get package manager:

```
sudo apt-get install git i2c-tools python-pexpect python-smbus libavfilter-dev
libavformat-dev libcurl4-gnutls-dev libgcrypt-dev libjson-c-dev libao-dev
```

This takes a little while; there's about 100 megabytes of stuff to download and install.

Now use git and wget to install the rest. There's our LCD radio UI, then a collection of Adafruit libraries for Raspberry Pi, and finally pianobar, a terminal-based Pandora client for Raspberry Pi. Do not use the apt-get version of this, it's out of date!

```
cd
git clone -b legacy https://github.com/adafruit/Adafruit-Raspberry-Pi-Python-Code
git clone https://github.com/adafruit/Python-WiFi-Radio
wget https://github.com/PromyLOPh/pianobar/archive/
e945578ab22912049f1e547ce7b25b01089f7590.zip
unzip e945578ab22912049f1e547ce7b25b01089f7590.zip
```

Note the “-b legacy” on the first “git clone” line. It's required there. But not on anything else.

Now some configuration...

First, link some of the Adafruit libraries into the radio software directory:

```
cd Python-WiFi-Radio
ln -s ../Adafruit-Raspberry-Pi-Python-Code/Adafruit_CharLCDPlate/*.py .
```

(Note: there's a space before the last period above. Don't miss it. Copy and paste this exact line, if possible.)

Next, we'll compile pianobar from source code:

```
cd ../pianobar-e945578ab22912049f1e547ce7b25b01089f7590
make
```

You might get a long list of compiler warnings. That's fine, as long as the build finishes. Compiler errors, on the other hand, will stop the build process. You might be missing a library from the apt-get sequence above, or something in the software may have changed since these instructions were written, in which case post in the [Adafruit Forums \(https://adafru.it/jlf\)](https://adafru.it/jlf) for assistance.

Once compiled, install using:

```
sudo cp pianobar /usr/local/bin
```

Don't worry about configuring this software yet; we'll proceed there later. Just use the steps above to get it installed for now.

---

## Configure Software Packages

Create a directory for the pianobar configuration:

```
cd  
mkdir -p .config/pianobar
```

Then link to the configuration file included with the radio software:

```
cd .config/pianobar  
ln -s ../../Python-WiFi-Radio/config .
```

(Note: there's a space before the final period above. Copy and paste this exact line if possible.)

**You MUST use this specific config file in conjunction with our WiFi radio software. Don't change any settings other than the ones prescribed below.**

Edit the pianobar configuration file with your Pandora account name and password:

```
nano config
```

The login credentials are near the top of the file:

```
user = YOUR_EMAIL_ADDRESS  
password = YOUR_PASSWORD
```

Replace these with the email address and password that you use for accessing your Pandora account, not the account on the Raspberry Pi system! Save the changes to the file and exit from the editor.

Finally, enter the following command to make sure audio is routed to the headphone jack rather than the HDMI port (you can skip this step if [using a USB audio device \(https://adafru.it/cSk\)](https://adafru.it/cSk)):

```
sudo amixer cset numid=3 1
```

And with that all said and done, you should now be able to run pianobar from the command line:

```
pianobar
```

If that runs as expected, connect headphones or speakers. The initial volume level will be very low. Type right parenthesis several times to increase the volume:

```
))))))
```

If that all works, press “q” to exit pianobar, then shut down the system in preparation for installing the final hardware:

```
sudo shutdown -h now
```

If you were remotely logged in via SSH, your connection will be closed and you won't see a “halt” message. Allow about 20 seconds before disconnecting power.

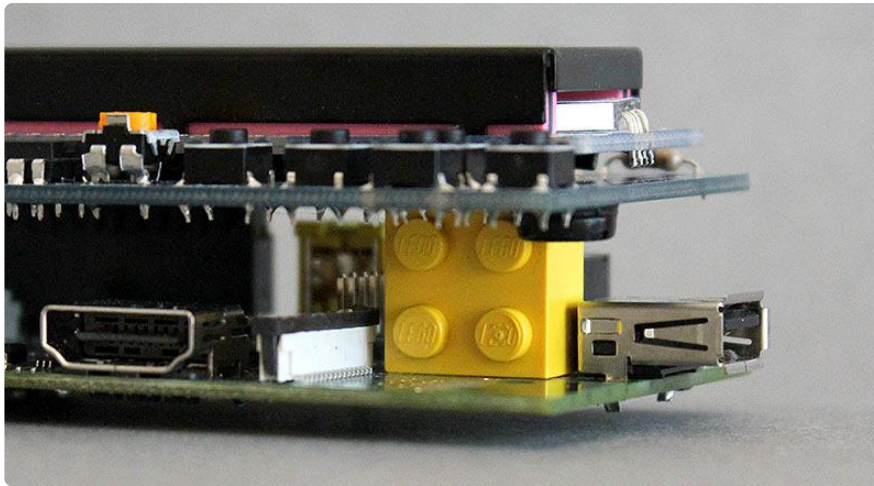
---

## LCD and Final Configuration

If you haven't already assembled the LCD Pi Plate, do that step now. [Here's a tutorial to guide you through the assembly \(https://adafru.it/aTG\)](https://adafru.it/aTG).

Raspberry Pi Model A users: the LCD Pi Plate is normally assembled with a rubber bumper near one corner for stability atop the Raspberry Pi. This bumper rests on the Ethernet jack, which is only present on the Model B board. Model A users will need to find a suitable alternative to the bumper, non-conductive and about 5/8" tall. A cork or rubber eraser trimmed to size can work, a 5/8" nylon PC board spacer if you have one, or a 2x2 Lego® brick set on its side.





The Raspberry Pi should be shut down when connecting the LCD and installing in a case. If it's currently running, issue the shutdown command as shown at the bottom of the previous page.

If using the Pi Box case, [follow this tutorial \(https://adafru.it/c6i\)](https://adafru.it/c6i) to install the Raspberry Pi inside, leaving the top of the case off.

Align the 26-pin headers on the LCD plate and Raspberry Pi, and install the plate by pressing straight down gently.

Connect power to the Raspberry Pi. The red power light should come on. If not, there might be a soldering mistake on the LCD Pi Plate board.

Allow 30 seconds to a minute for the Raspberry Pi to fully boot and situate itself on the network. At this point, you should be able to log in from a terminal over ssh, e.g.:

```
ssh pi@192.168.0.6 (or whatever address the system reported during the configuration process)
```

If you previously installed netatalk (optional), and if you changed the Raspberry Pi's default hostname, you might be able to use:

```
ssh pi@pandora.local
```

If you receive a response that the connection timed out or the hostname cannot be resolved, connect a monitor and keyboard and work through the "Initial System Configuration" steps again; it's most likely a typo in the wireless setup.

You should be successfully logged into the system at this point.

First, let's test the LCD and buttons:

```
cd Python-WiFi-Radio
sudo python Adafruit_CharLCDPlate.py
```

If using an RGB-backlit LCD, the program should cycle through different backlight colors (with the Blue & White LCD, it will flash on and off a few times). Then it will ask you to try pressing buttons.

Adjust the Contrast dial (using a small screwdriver if necessary) until the text is sharp and legible.

When finished, press Control+C to exit. The LCD is ready!

Next, let's try our radio software:

```
sudo python PiPhi.py
```

This is our own “wrapper” for pianobar, allowing us to use the LCD and buttons to interact with that program. If all goes well, the system should report its network address, fetch a list of stations from the Pandora server and start playing. If not, refer to the Troubleshooting page.

With only a few buttons available, we've condensed only the most essential functions to this program. If you need to configure your Pandora account (such as adding new stations to the list), use a web browser to access the Pandora web site.

From left to right, the button functions are:

- Play/Pause (tap once to pause, again to play, or hold for three seconds to exit the program).
- Station select (brings up a menu — can then use the up and down buttons to pick a station — press this button again to activate, or the Play/Pause button to cancel).
- Volume Up/Down (two buttons, one above the other)
- Next Track

Take a moment to experiment with the buttons and familiarize yourself with their use. If everything seems to be working, we'll do some final configuration to make the

system truly standalone. To exit from the PiPhi program, hold down Select (the leftmost button) for 3+ seconds.

Recent builds of pianobar seem resistant to quitting when the PiPhi.py script exits. If that occurs — if music is still playing even though you've quit the program — type:

```
sudo killall pianobar
```

This is only a potential issue during testing — once it's set up as a fully standalone device with auto-start and shutdown, you shouldn't experience this problem.

## Final Configuration and Auto-Start

First, cd to the directory with the PiPhi.py script (you should already be there if you've followed the steps till now!) and edit the PiPhi.py script:

```
nano PiPhi.py
```

Two lines near the start of the code (around line 26) are of interest:

```
RGB_LCD      = False # Set to 'True' if using color backlit LCD  
HALT_ON_EXIT = False # Set to 'True' to shut down system when exiting
```

If you've opted for an RGB-backlit LCD, change the first of these lines to:

```
RGB_LCD = True
```

Since our goal is a standalone system with no keyboard or monitor, we need some way to issue an orderly shutdown (Linux systems don't like it when you just pull the plug). To make the 3-second button press shut down the system (rather than just exit to a command line prompt), change the second line to:

```
HALT_ON_EXIT = True
```

Nearly there! Now we just need to set up the system to start our program upon booting.

```
sudo nano /etc/rc.local
```

Before the final "exit 0" line, insert these two lines:

```
cd /home/pi/Python-WiFi-Radio
python PiPhi.py &amp;
```

If you downloaded or otherwise placed the radio software in a different location, the first line should be changed accordingly. “sudo” isn’t necessary here because the rc.local script is already run as root.

Reboot the system to test the startup function:

```
sudo reboot
```

After 30 seconds to a minute, you should see the backlight turn on and music will begin. If not, connect to the system using ssh and confirm the configuration steps above.

---

## Troubleshooting

---

When I run the PiPhi script or any of the LCD examples, I get an “'lcd' is not defined” error message.

The PiPhi script must be run as root in order to access the LCD and keypad hardware:

```
sudo python PiPhi.py
```

If that doesn’t fix it, check for any solder bridges or cold joints on the LCD plate, and make sure the headers between the plate and Raspberry Pi are aligned.

Or the code may be having difficulty accessing the I2C bus. We can override this manually if the need arises. Edit PiPhi.py, looking for this line:

```
lcd = Adafruit_CharLCDPlate()
```

And simply add this parameter:

```
lcd = Adafruit_CharLCDPlate(busnum=1)
```

Run the code again and see if that helps.

---

## The Raspberry Pi does not boot when the LCD/Keypad Pi Plate is attached.

Check for any solder bridges or cold joints on the LCD plate, and make sure the headers between the plate and Raspberry Pi are aligned.

---

## A keyboard attached to the Raspberry Pi is not responding.

Some keyboards require more power than the Raspberry Pi USB port can provide. Add a powered USB hub, or try a different keyboard if you have a spare.

---

## The system never connects to the wireless network.

Most likely a typo during the wireless setup procedure. Go through each step on the “Initial System Configuration” page and check spelling carefully, including your wireless network name (SSID) and password.

“Hidden” WiFi networks are particularly fussy, and even with the extra directions provided might not connect. We very strongly recommend using a broadcast network name. If using a hidden network...when creating the file `wpa_supplicant.conf`, did you remember to change the SSID and password to the values used by your network?

It might also be an incompatible USB wireless adapter. The one in the Adafruit shop is known to work with the Raspberry Pi.

---

## The PiPhi script just hangs at “Receiving station list..”

The pianobar application is most likely failing to connect to the Pandora server... probably due to an incorrect email address and/or password. Check the settings in the configuration file and make sure these match your Pandora credentials, the account on your Raspberry Pi.

If in doubt, run pianobar directly from the command line first. Once that’s working, then move on to the PiPhi script.

---

## pianobar (or the PiPhi script) is running, but no sound is coming from the headphone jack.

Make sure audio is routed to the headphone jack rather than the HDMI port. From the command line:



```
sudo amixer cset numid=3 1
```

---

## When pianobar tries to connect, I get a “TLS handshake error” message.

Enter the following command (as a single line — copy and paste verbatim, if possible):

```
fingerprint=`openssl s_client -connect tuner.pandora.com:443 < /dev/null 2> /dev/null | openssl x509 -noout -fingerprint | tr -d ':' | cut -d '=' -f2` && echo tls_fingerprint = $fingerprint >> ~/.config/pianobar/config
```

Then try pianobar (or PiPhi) again.

(from Jacob Roeland's “pidora” tutorial)