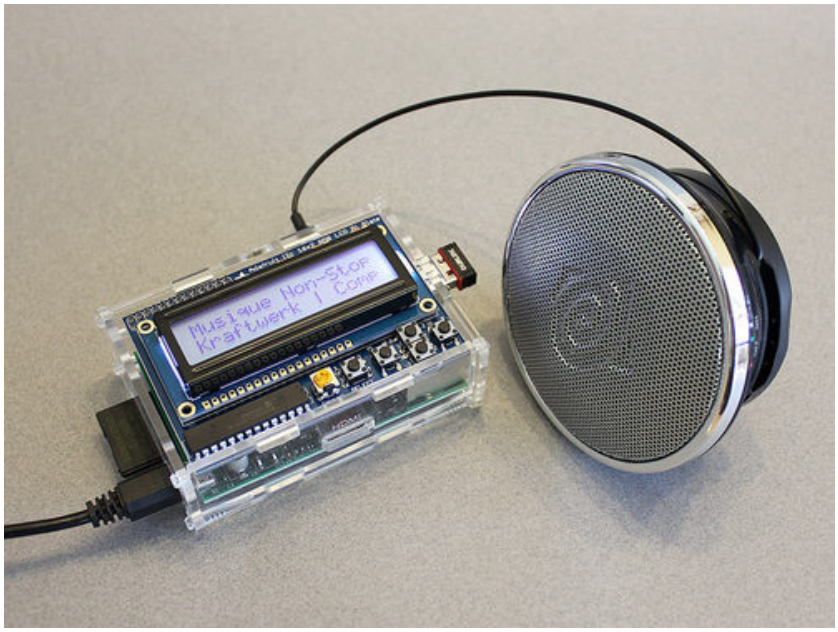
adafruit learning system

Raspberry Pi WiFi Radio

Created by Phillip Burgess



Last updated on 2013-11-02 01:45:26 PM EDT

Guide Contents

Overview

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.

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) board (any model or revision).
- 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://adafru.it/1109), RGB Negative (http://adafru.it/1110), and the more affordable Blue & White (http://adafru.it/1115). **All versions of this kit require some assembly and soldering.**
- A 4GB or larger SD memory card (http://adafru.it/102).
- A power supply compatible with the USB MicroB connector on the Raspberry Pi; some mobile phone chargers will work, or we offer suitable USB "wall wart" adapters (http://adafru.it/501) and cables (http://adafru.it/592).
- Headphones or amplified speakers (or connect to your living room A/V setup).
- A free account with the Pandora (http://adafru.it/c6j) streaming music service.

Optional but Recommended:

- A USB WiFi adapter (http://adafru.it/814) compatible with the Raspberry Pi (and an existing wireless network, of course). The Raspberry Pi Model B could be used with wired Ethernet, but this makes it less convenient.
- A Pi Box enclosure (http://adafru.it/859), or other case with an open top (the Pibow and Pi Shell 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)

Preparation

> Some of these steps are already covered in depth in other tutorials. We'll provide links where appropriate.

There are many "flavors" of Linux available for the Raspberry Pi. The software for this project is known to work with the official recommended *Raspbian* distribution, or Adafruit's own Raspberry Pi Educational Linux Distribution (aka "Occidentalis"). The latter has some handy additions of our own baked in, but if you'd rather use the stock distribution, that's okay, you'll just need a few extra commands in some places.

Occidentalis can be downloaded here:

- Adafruit Raspberry Pi Educational Distro - Occidentalis v0.2 (http://adafru.it/aPl)

Or Raspbian "Wheezy" can be found here:

- Raspbian "Wheezy" Downloads (http://adafru.it/aMY)

These files are about **900 or 500 megabytes** respectively and will take some time to download. While that's working, there are some other things you can do:

- Insert the SD card in the card reader, connect to your computer and format the card as a FAT32 (MS-DOS) filesystem.
- Solder up the LCD Pi Plate, if you haven't already. Here's a link to the tutorial (http://adafru.it/aTG).
- Prepare a Raspberry Pi case (peel plastic from acrylic, etc.). Tutorial link (http://adafru.it/c6i).
- Set up a free Pandora (http://adafru.it/c6j) account and favorite stations, if you don't already have one.

After downloading the software, uncompress the ZIP file in preparation for the next step…

Here's a tutorial explaining how to install the downloaded software onto the SD card (http://adafru.it/aYV), with links to nice GUI apps for Windows and Mac. The first couple of pages can be skipped, as we're already downloading the right software for this project. Since that tutorial was written, another option has become available for Mac: RPi-sd card builder (http://adafru.it/aYW) — this one's even simpler, with no Terminal commands required.

Initial System Configuration

OS Install and First-Time Configuration

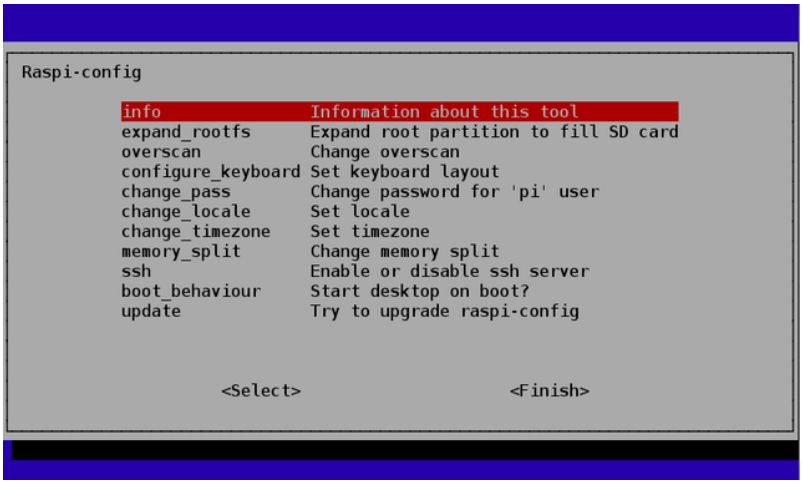
At this point it is now assumed you have an SD card containing either the Raspbian or Occidentalis software. If this is not the case, follow the directions on the “Preparation” page.

1. Connect a monitor and a USB keyboard to the Raspberry Pi.
2. Insert the SD card containing the Occidentalis software.
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.

(You can optionally continue with a “console cable” instead of a monito (http://adafru.it/aUA) (http://adafru.it/aUA)r, keyboard and power supply (http://adafru.it/aUA).)

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.



When the system is started for the first time, the Raspi-config utility runs automatically. You must select the following options:

- Expand root partition (this lets us use all the space on the SD card).
- Configure keyboard ("Generic 105-key (Intl) PC" is the default — for the US and most other countries, you'll want to select an appropriate keyboard layout such as "Generic 104-key PC").
- Set timezone.
- Enable SSH server (this permits network access to the Raspberry Pi without a monitor or keyboard attached).

These steps are optional:

- Change overscan (select "Disable" if using an HDMI monitor). Though our ultimate goal is to use the Raspberry Pi "headless," without a monitor, the extra space is welcome during the configuration steps.
- Change password for "pi" user.
- Change memory split. Again, since using the system "headless," more RAM can be allocated to the CPU rather than reserved for video.

These options should NOT be selected:

- Start desktop on boot (don't enable, since the goal is a "headless" system).
- Don't try to update yet; there's no network connection.

A more in-depth tutorial on Raspi-config is available here (http://adafru.it/aYX).

> Newer versions of Raspi-config have additional options such as overclocking. Even moderate overclocking runs the risk of corrupting the SD card. For this project we recommend leaving the Raspberry Pi at the default speed.

Select "Finish," but **DO NOT REBOOT YET.** We still need to set up the wireless networking. Select "No" when prompted to reboot. Raspi-config will exit to the command line.

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.

Change Hostname (Optional)

A unique hostname helps distinguish our music streamer from other Raspberry Pi system(s) on the network. The default hostname is “raspberrypi,” but let’s change that to “pandora” (or another name, if you like). **Recent versions of Raspbian let you do this from the config tool above.** Otherwise, at the command prompt, type:

sudo nano /etc/hostname

change:

```
raspberrypi
```

to:

```
pandora
```

Then:

sudo nano /etc/hosts

Change the last line from:

```
127.0.0.1    raspberrypi
```

to:

```
127.0.0.1    pandora
```

Configure Wireless Networking

If you have a WiFi network that broadcasts its SSID (the wireless network name), this is fairly straightforward:

sudo nano /etc/network/interfaces

delete everything in there, and copy and paste the below text in instead. Then edit the last two lines to reflect your actual network name and password (keeping the double quotes on both):

```
auto lo

iface lo inet loopback
iface eth0 inet dhcp

allow-hotplug wlan0
auto wlan0


iface wlan0 inet dhcp
    wpa-ssid "ssid"
    wpa-psk "password"
```

A more in-depth networking tutorial is available here (http://adafru.it/aWZ).

Skip ahead to the “Reboot” section below. This next section pertains only to “hidden” wireless networks.

Hidden Networks

If you run a hidden WiFi network — one that does not broadcast its SSID — this can still be made to work but is *considerably* more complicated. The slightest typo here, even one character, can prevent the system from joining the network! We strongly recommend using a “broadcast” network, it’s far easier and less error-prone.

sudo nano /etc/network/interfaces

Delete the last two lines (wpa-ssid and wpa-psk) and replace with (indented):

```
pre-up wpa_supplicant -Dwext -i wlan0 -c /etc/wpa_supplicant/wpa_supplicant.conf -B
```

sudo nano /etc/wpa_supplicant/wpa_supplicant.conf

```
ctrl_interface=/var/run/wpa_supplicant
update_config=1
ap_scan=2
eapol_version=1
network={
 ssid="my-network-ssid"
 scan_ssid=1
 mode=0
 proto=WPA2
 auth_alg=OPEN
 pairwise=CCMP
 group=CCMP
 key_mgmt=WPA-PSK
 psk="my-network-password"
}
```

Replace ssid and psk with your actual network name and password, in quotes.

Important: indent the 'network' section using tabs, not spaces.

Optional: Enable USB Audio

If you plan to use this with a USB Audio Adapter, this guide explains the process **(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 30 seconds.

Following shutdown, remove the keyboard (if using a Model A board), insert the WiFi adapter and re-connect power (keep the monitor attached for now). With a Model B board, you can keep both the keyboard and WiFi adapter attached until you know the networking is properly configured.

If you're using a console cable, just keep the cable connected and the terminal program open. Plug in the wifi adapter and disconnect/reconnect the red wire from the Pi to restart it.

This boot will take a very long time as the filesystem is expanded to fill the whole SD card.

During the boot process, you should see a series of DHCPDISCOVER messages as the system tries to join the wireless network. Eventually (by the fifth or sixth try), you should see a "bound to [address]" message, indicating a successful connection. **Remember that address!** It's probably going to look like **10.0.1.17** or **192.168.0.6** or similar. If it doesnt show up, log in with the **pi** username and run the command **ifconfig wlan0** and look for **inet**

addr:10.0.1.8 or similar.

If you still have keyboard and monitor connected, test the wireless connection by logging in and running this command:

```
ping -c 1 google.com
```

This will try to connect to Google's servers. If you see the response **1 packets transmitted, 1 received, 0% packet loss** then the WiFi connection works!

If the wireless connection doesn't work, unplug the WiFi adapter and connect the keyboard (if using a Model A board) and double-check the network setup steps above. It might simply be a typo. Correct any mistakes and reboot.

Once WiFi is working, the keyboard and monitor are no longer required. All further configuration can be done remotely via SSH.

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. As you can probably figure by now, we have an in-depth SSH tutorial (http://adafru.it/aWc). In summary:

- 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 (http://adafru.it/aYY).

The terminal command to access the system would be:

ssh pi@*thataddressfromthe-previous-step*
for example:
ssh pi@10.0.1.10

You'll be prompted for a password — either use the password that you set up from 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.

Install Prerequisite Software

Several prerequisite software packages need to be installed, each using a different technique.

First is **pianobar**, a terminal-based Pandora client for the Raspberry Pi. This one's quite simple. From a command line prompt, type:

sudo apt-get install pianobar

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

Next is the **pexpect** library for Python, which must be manually downloaded, extracted and installed:

wget http://pexpect.sourceforge.net/pexpect-2.3.tar.gz (http://adafru.it/c6l)
tar xzf pexpect-2.3.tar.gz
cd pexpect-2.3
sudo python ./setup.py install
cd ..
sudo rm -r pexpect-2.3

Finally, two collections of Adafruit code. One is our radio-specific software, the other is a collection of libraries for Raspberry Pi. If you're running Raspbian "Wheezy," you'll first need to install Git:

sudo apt-get install git

Occidentalis already includes Git by default. Either way, then type:

git clone https://github.com/adafruit/Python-WiFi-Radio (http://adafru.it/c6m)
git clone https://github.com/adafruit/Adafruit-Raspberry-Pi-Python-Code (http://adafru.it/aOg)

Finally, link some of the latter 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.)

Additional Prerequisites for Raspbian Users

If using the "Occidentalis" software, you can skip these steps; the features are already baked into the system. Otherwise:

sudo nano /etc/modules

Add these two lines to the end of the file:

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

Then install additional Python libraries and tools:

sudo apt-get install python-smbus i2c-tools

Finally, reboot to activate I2C (Occidentalis users can skip this step):

sudo reboot

Optional Software

You can optionally install a VNC server to allow a remote graphical session. This is a lengthy topic and best covered in its own tutorial (http://adafru.it/aZ4). However, for the "Running VNCServer at Startup" section, use the directions in this tutorial instead (http://adafru.it/aZ5) (specifically the "Getting VNC Server to Work on a Specific User" section).

Mac users may want to install *Netatalk,* which makes the system appear in the Finder sidebar and can simplify transferring files to and from the system:

sudo apt-get install netatalk

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.

Next, 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

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 (http://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:

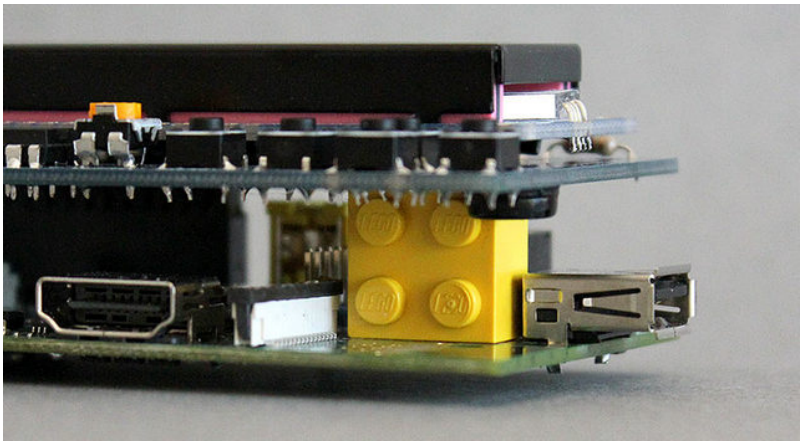
sudo shutdown -h now

If you were remotely logged in via SSH, your connection will be closed and you won’t see a “halt” message. Allow about 30 seconds to pass 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 (http://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 (http://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.

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 &
```

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 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.

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.

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.

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, *not* 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.

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, not 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

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)

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)
