

Introduction

In the March 2012 issue of Nuts and Volts I wrote an article entitled “Build Your Own Wifi Internet Radio” about how to build an Internet radio using a repurposed wireless router. While this project worked fine (and still does) it had its limitations driven mainly by the small amount of RAM available on the router. After publication of that article, I continued to evolve the design so that it could also play song files stored on a flash / thumb drive as well as functioning as an Internet radio player. A scaled down version of the Music Player Daemon or MPD server had to be used on the router due to the small amount of available RAM which limited its ability to play many popular song file formats. This was OK for a while but it meant I had to convert songs stored as AAC files under iTunes to MP3 files to be able to play them. Still, I liked the idea of having the music I was currently interested in on a flash drive and being able to change it easily. Also since the capacity of flash drives were going up and the prices were going down I could have a rather large number of songs available at one time. And to put frosting on the proverbial cake I could control the Internet radio / music file player with a new iPhone / iPod Touch app I was developing called *MpdClient*.

In April of 2012 the Raspberry Pi (see Photo One) single board computer became available (at least in the UK). This powerful little device set a new standard for computing power per dollar spent. See Table One for the details. While the Raspberry Pi was developed for educational purposes, many hobbyists, makers, DIYers, myself included, could not wait to get our hands on one. Unfortunately, for us in the US, it would take quite a while for production to catch up with the international demand. I finally received my first Raspberry Pi mid November 2012 and I was awestruck. This credit card sized device was a complete Linux workstation with more power than many of the UNIX workstations I used during my career in software development and it only cost \$35.

In March of 2013 after becoming comfortable with my Raspberry Pi, I wrote another article for Nuts and Volts entitled, “Raspberry Pi, Anyone ?” detailing how to get started with this device. In that article I described how to install the operating system (OS) software (the Raspbian version of Linux) on the SD memory card and how to configure the Raspberry Pi to make it useful. I concluded that article by showing how the Raspberry Pi could be made into a web server using just a single line of Python code.

Shortly after that article was published I figured out how to install a full featured version of MPD on my Raspberry Pi to provide Internet radio as well as music file player functionality. And as mentioned, I wrote a iPhone / iPod Touch app called *MpdClient* (hopefully available in the Apple App Store by the time you read this) to provide complete control of the MPD server remotely. Now setting in my favorite chair I can listen to Internet radio stations or play artists, albums or individual songs from a flash drive plugged onto my Raspberry Pi.

In this article I will show you step by step how to make your Raspberry Pi into a remote controllable, Internet radio / music file player.

Hardware

With the high level of system integration on the Raspberry Pi you can make a Internet radio / music file player with fewer parts than what would be required using a repurposed wireless router. Figure One shows a block diagram of the proposed system and Table Two gives some specifics as to the components required. For our purposes here, I will assume your Raspberry Pi will be connected to an existing audio system (through an appropriate stereo interconnect cable) or used with headphones so we won't worry about discussing amplifiers or speakers. I will show you how to connect the Raspberry Pi wirelessly to your local wireless network using an inexpensive (less than \$6) USB wifi adapter however, you could run an Ethernet cable from your Raspberry Pi to your router and skip the wifi stuff altogether.

The two USB ports on the Raspberry Pi Model B fit our needs exactly with one port for the USB wifi adapter and the other for a flash drive containing your music. A Model A Raspberry Pi would not work for this application because it only has a single USB port unless you wanted to incorporate a USB hub into your system. In any case the increased amount of RAM (512 Mbytes) on the Model B probably makes the MPD server run better.

So the hardware connections to/from the Raspberry Pi are as follows:

1. Connection of the power adapter to the micro USB port on the card
2. Connection of the 3.5 mm audio output to your music system via the appropriate cable
3. Insertion of the 4 Gbytes or larger SD memory card containing the operating system
4. Connection of the USB wifi adapter to one of the USB ports
5. Temporary connection of a USB hub to the other USB port. This will only be needed during the setup process and will be removed when setup is completed. Connect a USB keyboard and mouse to the hub at this time. The USB hub will be replaced with a USB flash drive containing your music library when setup is finished.
6. Connection to either an HDMI or composite video monitor. Your Raspberry Pi will run headless, that is with no monitor required, once the installation procedure is completed.

Software

When I first began using the Raspberry Pi for Internet radio it had a real problem with loud pops when changing stations. So loud in fact as to almost make this arrangement almost unusable. Recently, however I reloaded my Raspberry Pi with the 02-09-2013 version of the Raspbian operating system (available from raspberrypi.org) and this problem seems to have been fixed. If you have an older version of the operating system software on your Raspberry Pi, I strongly suggest reloading it with this new version by following the direction in my previous Raspberry Pi article of March 2013. All of the initial configuration described in that article is still appropriate and should be done before proceeding.

Assuming we are starting with a newly loaded Raspberry Pi image we will first get wifi

working, then update the OS software with all the newest packages. Next we will update the Raspberry Pi firmware and install a series of package required to run MPD. Finally we will configure MPD for our purposes.

To get started, insert the SD memory card with the operating system into the Raspberry Pi and make all of the connections listed in the hardware section above. Once power is applied, your Raspberry Pi should boot and take you into raspi-config automatically if you haven't done your initial configuration already or to the command prompt if you have. At this time please log in with the username of **pi** and the password of **raspberry**.

1. At the command prompt type **startx** to start up the Graphical User Interface or GUI.
2. Click on the **WiFi Config** icon to configure your Raspberry Pi for wireless operation. If you are going to use a wired network connection skip to step 6.
3. From the app's menu click **Network** and then **Add**.
4. Input the SSID or name of your wireless network and select the appropriate Authentication method for your network, if any. My wireless network uses *WPA2 – Personal* authentication but yours may be different. Finally, type in your network password into the PSK field.
5. Click the **Add** button and if all is well your Raspberry Pi should connect to your wireless network. If it doesn't, check to see that you have the correct SSID, Authentication and PSK values set.
6. Click on the **LXTerminal** icon on the desktop to bring up a shell window
7. Type **sudo apt-get update** to allow your Raspberry Pi to determine all of its OS packages that are out of date.
8. Type **sudo apt-get upgrade** to install all of the new OS packages.
9. Once the upgrade is complete type **sudo reboot** to reboot your system. A reboot is necessary to utilize all of the new software you just installed.
10. Log back in and run **startx** again. This time click the **Midori** icon to bring up the browser. Type **github.com/hexxeh/rpi-update** into the location bar.
11. Click on the **LXTerminal** icon to also bring up a shell window
12. In the shell type **sudo apt-get install git-core** to load some more software we will need.
13. We will now download and install the Raspberry Pi firmware updater so that our system can be completely updated both in terms of its firmware and its OS software. In the Midori window scroll down to the *Installing* section and highlight the long command that begins with `sudo wget ...` Once highlighted, copy it to the clipboard and then paste it into LXTerminal window to execute it.
14. To run the firmware updater type **sudo rpi-update**.
15. After this process completes, reboot your system yet again and log back in.
16. Next we will install software that will *automount* any USB flash drives plugged onto the Raspberry Pi. This means that if you plug in a new flash drive the system will recognize it automatically without you having to do anything. Type **sudo apt-get install usbmount**. When completed, type **sudo service udev restart** to make the system begin using the usbmount software.
17. To test that automounting is working, list the content of the directory `/media/usb0` using

the command: **ls /media/usb0**. The directory should be empty. Plug in a USB flash drive with some files on it into the USB hub and run the command again. This time the content of the USB flash drive should be visible in the directory listing. If not reboot and try again.

18. Finally install MPD and MPC using the following command: **sudo apt-get install mpc mpd**. We install MPD because we need it. We install MPC to help in testing the MPD server.
19. Next we need to edit the MPD configuration file that was just installed. We will invoke the **vi** program editor with the following command line: **sudo vi /etc/mpd.conf**. You could also use another editor if you are more comfortable with it.
20. Find the *music_directory* line in the file and change its value in double quotes to **"/media/usb0"**. This tells MPD where to look for music files. Next find the *bind_to_address* line and replace "localhost" with "any". This allows us to interact with the MPD server remotely. Next, find the line *#mixer_type "software"* and remove the # sign. Finally find the line *#autoupdate "yes"* and first remove the # sign and then change "yes" to "no". Save the changes to the file with the command **:wq** if using **vi**.
21. Reboot once more so MPD will read its new configuration.
22. Log in and type **alsamixer** on the command line. This will bring up a GUI with a slider in the middle of the screen. Raise the value three quarters of the way up and then click the Esc key to terminate this program.
23. To make sure the kernel modules necessary to output sound from the Raspberry Pi are loaded type the following command in a shell window: **sudo modprobe snd_bcm2835**.
24. Lastly, type **amixer cset numid=3 1** to route sound to the 3.5 mm jack on the Raspberry Pi.
25. Reboot one more time looking at the messages that stream by on the monitor. Make sure you don't see any errors when the MPD server automatically starts up. If you do, something is wrong and you will need to figure out what that is and fix it.

Now it is time to find out if our Internet radio / music file player is working. For this test make sure there are music files on the flash drive which is plugged into the USB hub. Then plug some headphones into the 3.5 mm jack on the Raspberry Pi. Type the following command in a shell window: **mpc add <http://64.40.99.2:8088>**. Then type **mpc play**. If all is well you should hear the Beatles Internet radio station.

Next type **mpc clear** to stop the server from playing the Internet radio station. Then type **mpc update** to force the MPD server to build a database of the songs on the flash drive. Type **mpc listall** and you should see all the songs on the flash drive listed. Then type **mpc add** followed by one of the file names you saw with the **listall** command. Finally type **mpc play** and you should hear the song you just added.

If these tests are successful you are ready to go. You can power down your Raspberry Pi and remove the USB keyboard, mouse and the USB hub. Remember to plug your USB flash drive containing your songs back into the Raspberry Pi and connect the audio cable to your stereo system.

To control your MPD server remotely you can use my iPhone / iPod Touch app *MpdClient* or any other MPD client application you have available.

Songs on a USB Flash Drive

The MPD server used in conjunction with the Raspberry Pi can play many different audio file formats. If you would like a complete list of supported files type **mpd --version** at a command line prompt. This means you can put a variety of audio file types on your USB flash drive and MPD should have no trouble indexing and playing them all.

MPD does not require any kind of directory structure for music file storage as some other music library programs do (like iTunes). All files can be placed in the root directory of your flash drive and MPD will sort things out. This is possible because MPD extracts song information from metadata stored in the music files themselves. When MPD is commanded to do an update, it extracts information from the song files and stores it in its private database. This allows for some very powerful searching and sorting capabilities. Using the MPC client we installed on the Raspberry Pi for testing we can list all available songs by typing **mpc listall**. Type **mpc list artist** to see a list of all of the artists with songs on the flash drive. Type **mpc list album** to see the albums the songs on the flash drive are associated with. Finally type **mpc list genre** to see the genres covered by the songs on the flash drive.

While these capabilities might not seem all that powerful consider the following example. Say you execute **mpc list genre** and you get the following list returned.

```
World  
Rock  
Unclassifiable  
Holiday  
Alternative & Punk
```

And you feel like only listening to World music. You can do so using the following series of commands:

1. **mpc clear** – this clears the server's playlist of all songs which stops playback
2. **mpc findadd genre World** – this creates a World music playlist on the server
3. **mpc play** – starts the playback of World music

Of course I only used genre here as an example but you could use these same techniques to select songs from a specific album or to playback all songs by a specific artist. Pretty powerful stuff really.

Conclusions

Having a remote controllable Internet radio / music file player is very convenient. Building one

yourself inexpensively makes it all that much sweeter.

Bio

Craig lives in the mountains of Colorado and can be contacted at calhjh@gmail.com. When not messing around with music, electronics, computer projects, wood working or beer brewing, he plays in a rock and roll band and does a solo musical act around Colorado Springs.

Photo One
The Model B Raspberry Pi

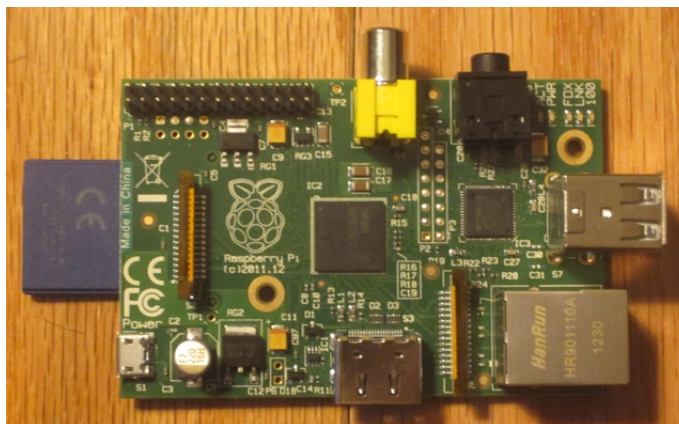


Table One
Raspberry Pi Specs

| Feature | Model A | Model B |
|---------------------------|--|----------------------------|
| Suggested Retail Price | \$25.00 | \$35.00 |
| CPU | 700 MHz ARM11 with hardware floating point | |
| GPU | Capable of 1080p video | |
| Memory | 256 Mbytes shared with GPU | 512 Mbytes shared with GPU |
| USB Ports | 1 | 2 |
| Video Outputs | HDMI and composite video | |
| Audio Outputs | Stereo 3.5mm jack + HDMI | |
| Onboard Storage | SD memory card up to 128 GByte | |
| Networking | none | 10/100 Ethernet |
| Connectivity | 8 × GPIO, UART, I ² C, SPI | |
| Size | "3.3" x "2.1" | |
| Power Requirement | 300mA – 1.5W | 700 mA – 3.5W |
| Official Operating System | Raspbian version of Linux | |

Figure One

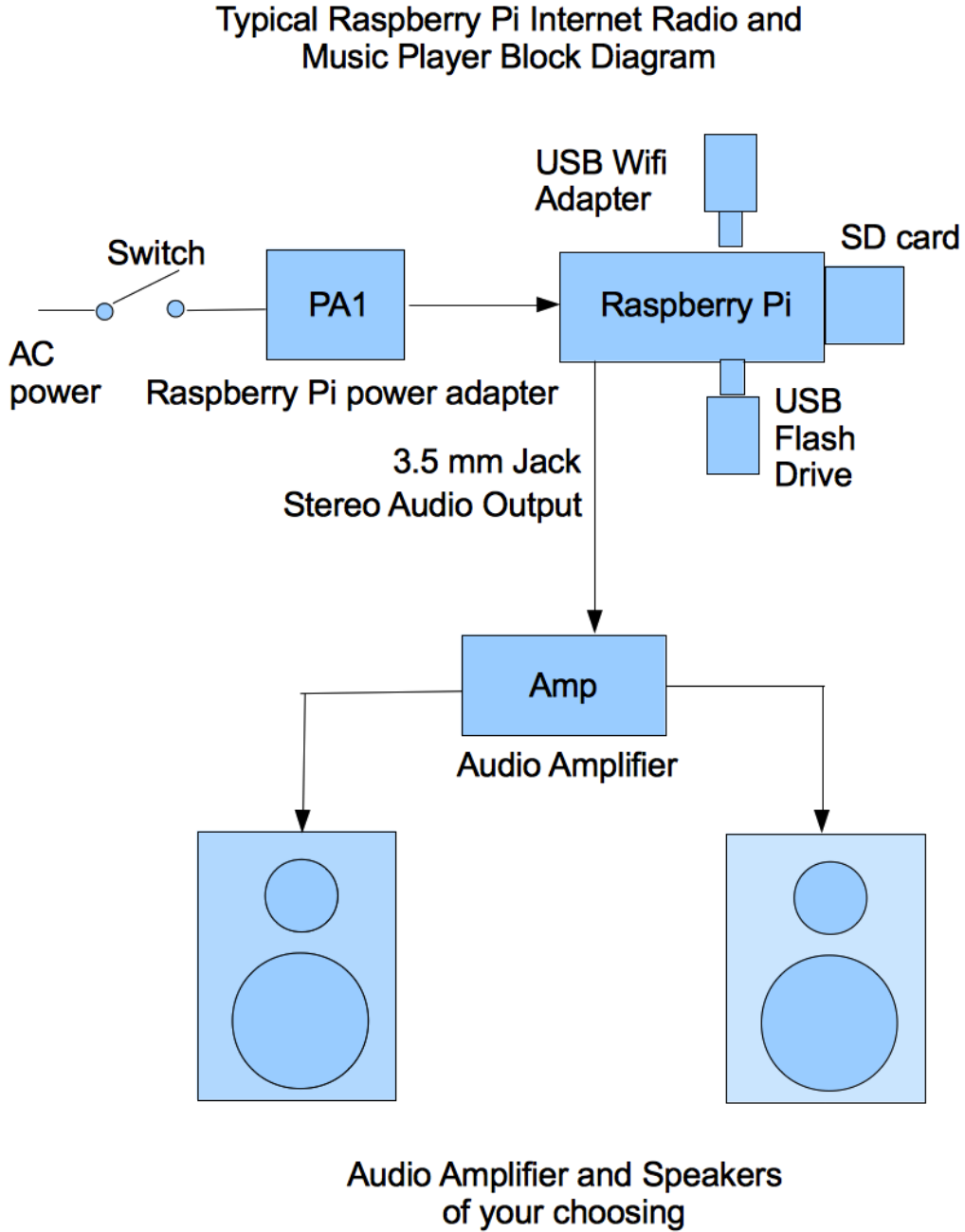


Table Two
Required Hardware

| Item | Source |
|--|---|
| Raspberry Pi Model B with power supply | adafruit.com, www.mcmelectronics.com, www.sparkfun.com, www.newark.com |
| SD Memory Card 4 GBytes or larger | everywhere |
| Flash Drive Bigger == More Songs I used a 8 GBytes device | everywhere |
| Mini 150Mbps USB WiFi Wireless N LAN Network Adapter 802.11n/g/b | ebay |
| Audio Output Cable 3.5 mm stereo cable to whatever input (s) your amplifier requires | ebay, bestbuy, amazon |