



# The PirateBox Project

scott.murphy@arrow-eye.com

2018-02-08

No parrots are required for the production of a PirateBox

Ottawa Canada  
Linux Users Group

# About this Talk

This little project crossed my radar a few times and I was thinking that it might have alternative uses. I'll save the alternative ideas for the end and just get started on the talk.



**Ottawa Canada**  
**Linux Users Group**

# What hardware is being used?

For the demo and the currently completed version:

- A RaspberryPi v3
- A 8GB class 10 microSD card
- A 16GB USB drive
- A USB power supply
- A case to house the unit

This can be done with a wide variety of hardware. I started with a TP-Link MR3020, but there are a few issues with that. I can talk about that after I complete the talk and demo.

# What is a PirateBox?

From the project website:

- PirateBox is a DIY anonymous offline file-sharing and communications system built with free software and inexpensive off-the-shelf hardware.
- PirateBox creates offline wireless networks designed for anonymous file sharing, chatting, message boarding, and media streaming. You can think of it as your very own portable offline Internet in a box!

# Why is it called that?

Well, that is a little more historical. It was a nod to the old pirate radio stations that flourished in the 1960's, one example being Radio London that operated from 23 December 1964 to 14 August 1967 from a ship anchored in the North Sea, three and a half miles off Frinton-on-Sea, Essex, England.

More currently, this is about sharing information freely, open source, and free software.

Of course, it may also have had something to do with the skull and crossbones on the old style lunch box that was used to house the prototype.

# Getting started

There is an easy to follow tutorial to get this up and working on the project website. In general, this works as described and everything just works. In my case, it didn't work exactly as advertised, however it was sufficiently close to the instructions, so I'll say it they are easy to follow if you work with open source software and Linux.

Tutorial Link: [Raspberry Pi\(rate\)Box](#)



# Installation Walkthrough

We will look at the installation on the Pi. This is pretty easy to follow if you have installed any moderately complicated software. We will just go over the steps for the sake of showing how easy this was to do.

# Make the SD Card image

The system is based on Arch Linux, not Raspbian, so I was pleasantly surprised. The file is a zip file, so once it has downloaded, unzip it. You could work with it compressed, but some people will be using Windows for this.

Once downloaded, write it to the SD card with an appropriate utility. dd will work very well for Unix/Linux/macOS systems.



# Set up the USB stick

Format the USB stick FAT32 on another device and place it in the Raspberry Pi.

This can be a little difficult to accomplish on in the newer macOS versions. The GUI utility doesn't always format it correctly. You need to use a terminal and the `diskutil` command:

```
diskutil partitionDisk disk2 MBR FAT32 DOS 100%
```

# Connect the Pi to your LAN

This part was a little off for me. I happen to have a DMZ that uses the same address range as the PirateBox image wants to use, so it caused me some grief in a future step.

The reason for connecting to the LAN so you can ssh to it to complete the setup after it configures.

# Older Pi units

If you have one, then you should attach a WiFi dongle at this point. There is a list of compatible ones on the project site. The Pi v3 has WiFi built in, so you can skip this step.

Of course, an external dongle with an antenna will extend the range if you want that to happen.

# Power it up

It will take a couple of minutes to boot and come online the first time. If there is no WiFi antenna, the access point will not start, so that would be a show stopper.

# Configuring

This is the part that didn't work immediately for me, due to the conflicting addresses (I think). I was under the impression that the LAN connection would DHCP an address from my internal network, but that didn't happen. You have a few options at this point, use a serial console cable or try the WiFi. I used the WiFi first, as not everybody has a cable to use as a serial console for the Pi. The default connection is over ssh.

```
ssh alarm@alarmpi
```

With the default password of `alarm`

# Once logged in

- Change the password. Until you do this, anyone can log in and do it.
- The root account can't log in unless you make a couple of changes. Don't!

We are almost ready.

# Post Install

At this point, you should enable the basics, so log in (if you are not already)

```
ssh alarm@192.168.77.1
```

Activate the extra storage (USB or SD card). I'd go with the USB device, as flash has life expectancy issues with using the SD card in RW mode.

```
sudo /opt/piratebox/rpi/bin/usb_share.sh
```

Activate the Kareha Image and Discussion Board by using the board-autoconf tool:

```
sudo /opt/piratebox/bin/board-autoconf.sh
```

Activate the "timesave functionality" once:

```
sudo /opt/piratebox/bin/timesave.sh \ /opt/piratebox/conf/piratebox.conf install
```

```
sudo systemctl enable timesave
```

# Post Install (contd)

Activate the UPnP Media Server by copying over the config file:

```
sudo cp /etc/minidlna.conf /etc/minidlna.conf.bkp  
sudo cp /opt/piratebox/src/linux.example.minidlna.conf \ /etc/minidlna.conf
```

Optionally, you can edit the config file (change the display name, etc) with:

```
sudo nano /etc/minidlna.conf
```

Finally, start the UPnP Media Server with:

```
sudo systemctl start minidlna sudo systemctl enable minidlna
```



# Demo Time

- Give the box a few moments to fully boot
- Connect to the SSID: **Pirate Box – Share Freely**
- Browse anything, it should just redirect any address to itself.
- If you are having problems, try `http://192.168.77.1`

Note: That is an address I used to avoid conflicts. Your own build will be different

# The TP-Link MR3020 Fun

- I don't have any slides prepared for this part, just discussion
- Hardware versions are not always what they seem
- It has a lot of potential in this form factor

According to the project website, the latest revisions are no longer compatible with the project.

# Other Hardware I'm looking at

- GL-iNet devices
- GL-MT300N v1
- GL-MT300N v2

These are not listed as supported, the older version based on the same chipset as the TP-Link 3020 unit is supported.

Since the Pi can run it and there is an OpenWRT image for the two units, I would expect that building a dev environment and compiling the packages yourself will make it all work. If I manage it, I'll give a short update.

# GL-iNET MT-300N v1 Specs

CPU	MTK 7620N @580Mhz
Memory/Storage	DDR1 64MB/ FLASH 16MB
Interfaces	1 WAN, 1 LAN, 1 USB2.0, 1 micro USB (power), 1 Reset button
Frequency	2.4GHz
Transmission rate	300Mbps
Max Tx Power	20dBm
Protocol	802.11 b/g/n
External Storage support	FAT32/EXFAT/EXT4/EXT3/EXT2/NTFS
Webcam support	MJPEG, YUV
DIY features	UART, 4GPIO, 3.3V & 5V power port
External antenna support	Yes (Optional)
PoE support	Yes (Optional)
Power input	5V/1A
Power consumption	<2W
Dimension, Weigh	58*58*25mm, 39g



**Ottawa Canada**  
**Linux Users Group**

# GL-iNET MT-300N v2 Specs

CPU	MTK 7628NN @580Mhz
Memory/Storage	DDR2 128MB/ FLASH 16MB
Interfaces	1 WAN, 1 LAN, 1 USB2.0, 1 micro USB (power), 1 Reset button
Frequency	2.4GHz
Transmission rate	300Mbps
Max Tx Power	20dBm
Protocol	802.11 b/g/n
External Storage support	FAT32/EXFAT/EXT4/EXT3/EXT2/NTFS
Webcam support	MJPEG, YUV
DIY features	UART, 4GPIOs, 3.3V & 5V power port
External antenna support	No
Power input	5V/1A
Power consumption	<2.75W
Dimension, Weigh	58*58*25mm, 39g

You may notice that these specs say no external antenna, yet I see two antenna connectors on the circuit board, so I would think that if you have antennas from the previous model they will work on this one. This is not tested, but I know that industry practice is to not add components you have no intention of making use of.



**Ottawa Canada**  
Linux Users Group

# Summary

An interesting project to do if you have a spare Raspberry Pi and some time on your hands. It has enough other possibilities to make it a useful thing to have for those moments when you need something that can share files.

- Group work on something where there is no available WiFi

# Questions & Discussion

- What other things would this be useful for?
- Media streamer for long trips, everybody could watch what they want.

# References

- [The PirateBox Project](#)
- [Hacked and Unpacked: The Piratebox](#)
- [Troubleshooting](#)
- [Building A PirateBox](#)
- [Raspberry Pi PirateBox](#)