

Table of Contents

Requirements.....	2
One – OS installation.....	3
Two – First Boot.....	3
Three – Partition Configuration.....	3
Four – OS Configuration.....	3
Five – Configure Music Library Storage.....	4
Six – USB Mass Storage Gadget.....	4
Seven – Sound Card.....	4
Eight – Install and Configure MPD.....	5
Nine – Install MPD Script.....	5
Ten – Add Music.....	6
Finally – Change to Read Only Root.....	6
Appendix One – Remote Control a Pi zeroW/WH.....	7
Appendix Two – Sharing Your Music.....	9
Appendix Three – Airplay Receiver.....	10

Requirements

- Raspberry Pi ZeroW
- uSD card, 8GB minimum but bigger is better
- DAC hat/phat/bonnet¹ e.g Pimoroni phatdac
- Head unit with AUX input
- USB A to micro USB cable
- 12v to 5v PSU
- Optional
 - case
- For setup
 - PSU
 - Linux computer with gparted installed e.g. raspberry pi², PC with live CD/DVD/USB stick
 - USB uSD card reader
 - USB otg adaptor
 - USB ethernet adaptor
 - Ethernet cable
 - ssh client e.g. putty
 - Optional
 - HDMI monitor or TV
 - HDMI adaptor and cable
 - USB hub
 - USB keyboard
 - USB mouse

¹ Or add basic audio out: <https://learn.adafruit.com/adding-basic-audio-ouput-to-raspberry-pi-zero>

² This can be the same raspberry pi you're using in the final device but you will need a second uSD card to boot it from.

One – OS installation

Download and install the latest Raspbian Lite image in the usual way. See

<https://www.raspberrypi.org/documentation/installation/installing-images/README.md>

At time of writing this was 2018-04-18-raspbian-stretch-lite

Two – First Boot

1. Open the first partition of your uSD card.
2. Enable ssh: create an empty file called ssh³
3. Insert uSD card into pizero
4. Connect USB otg adaptor
5. Connect monitor, hub, mouse, and keyboard if using.
6. Connect USB ethernet adaptor, and ethernet cable
7. Boot your pi zero
8. Login to your pi
9. Shut it down:
`sudo poweroff`
10. Disconnect power and remove uSD card

Three – Partition Configuration

1. Connect the card reader and uSD card to your linux computer.
2. Launch gparted. This needs to be done as root or with sudo/gksudo
3. Select the correct device⁴. It should have two small partitions labled “boot” and “rootfs”
4. Shrink the “rootfs” partition to approximately 4GB.
5. In the free space, create a new FAT32 partition. Use all of the available space. We’ll be using this partition to store our music.
6. Apply changes
7. Exit gparted
8. Remove uSD card

Four – OS Configuration

1. Insert uSD card into pi zero
2. Connect USB otg adaptor
3. Connect monitor, hub, mouse, and keyboard if using.
4. Connect USB ethernet adaptor, and ethernet cable
5. Boot your pi zero
6. Login to your pi zero
7. Update package lists:
`sudo apt update`
8. Upgrade software:
`sudo apt upgrade`
9. Run
`sudo raspi-config`
10. Under boot options set wait for network to no
11. Optionally change hostname (found under network options)
12. Exit raspi-config

³ ssh.txt should work too.

⁴ If using a raspberry pi, you want the one appearing as /dev/sd? not /dev/mmcblk0

Five – Configure Music Library Storage

1. Create a mount point for the music library storage: `sudo mkdir /music`
2. Open `/etc/fstab` in your favourite editor⁵ e.g. `sudo vi /etc/fstab`
3. Add a new entry as follows:
`/dev/mmcblk0p3 /music vfat ro 0 0`
4. Save and close `/etc/fstab`
5. Mount music storage as read/write:
`sudo mount -o rw /music`
6. Create playlists directory:
`mkdir /music/playlists`
7. Unmount music storage:
`sudo umount /music`

Six – USB Mass Storage Gadget

1. Open `/boot/config.txt` in your favourite editor, this will need `sudo`.
2. Add a new line at the bottom of the file:
`dtoverlay=dwc2`
3. Save and close `/boot/config.txt`
4. Open `/etc/rc.local` in your favourite editor, this will need `sudo`.
5. Above “`exit 0`” add the following:
`modprobe g_mass_storage removable=1 file=/dev/mmcblk0p3 &`
6. Save and close `/etc/rc.local`

Seven – Sound Card

1. Follow the manufacturer’s instructions.
2. If using Pimoroni’s phat DAC:
`curl -sS https://get.pimoroni.com/phatdac | bash`

⁵ Yeah, I’m old school. Change `vi` to your favourite.

Eight – Install and Configure MPD

1. Install mpd and mpc:

```
sudo apt install mpd mpc
```
2. Stop mpd:

```
sudo systemctl stop mpd
```
3. Backup mpd.conf:

```
sudo cp /etc/mpd.conf /etc/mpd.conf.bak
```
4. Open /etc/mpd.conf in your favourite editor, this will need sudo.
5. Find the line that starts with “music_directory” or “#music_directory”. Change it to:

```
music_directory    "/music"
```
6. Find the line that starts with “playlist_directory” or “#playlist_directory”. Change it to:

```
playlist_directory "/music/playlists"
```
7. Find the line that starts with “log_file” or “#log_file”. Change it to:⁶

```
log_file    "/var/log/mpd.log"
```
8. Find the line that starts with “auto_update” or “#auto_update”. Change it to:

```
auto_update    "no"
```
9. Find the line that starts with “replaygain” or “#replaygain”. Change it to:

```
replaygain    "auto"
```
10. Find the line that starts with “volume_normalization” or “#volume_normalization”. Change it to:

```
volume_normalization    "yes"
```
11. Restart mpd:

```
sudo systemctl start mpd
```

Nine – Install MPD Script

1. Download initmpd.sh
2. Move or copy it to /home/pi
3. Ensure it has execute permission:

```
chmod +x initmpd.sh
```
4. Open /etc/rc.local in your favourite editor, this will need sudo.
5. Above “exit 0” but below “modprobe ...” add the following:

```
/home/pi/initmpd.sh &
```
6. Save and close /etc/rc.local

⁶ If left at the default mpd will hang on start up once the filesystem has been made read only.

Ten – Add Music

This method is sub-optimal as it changes the music files under the feet of a running MPD. It is recommended that you at least disconnect the audio output while updating your music.

1. Disconnect everything from your pi
2. Connect a micro USB cable to the data (inner) port of the pi zero
3. Connect the cable to your PC. The pi will boot and appear as a USB storage drive.
4. Copy music files to this drive
5. Create an empty file called “update” or “update.txt” in the top level directory of the USB drive. This will trigger an update of the mpd database and playlist on next boot.
6. Safely remove and disconnect the pi from the PC.
7. Reboot the pi to run the update. This will take a while⁷, how long depends on how much music you have.

Finally – Change to Read Only Root

If you’re intending to carry out any of the steps in the appendices below do so before carrying out this step.

A read only root should prevent uSD card damage in the event of sudden power loss⁸. It also removes the need for a clean shutdown every time.

See <https://learn.adafruit.com/read-only-raspberry-pi> for full information.

1. Download the setup script:
`wget https://raw.githubusercontent.com/adafruit/Raspberry-Pi-Installer-Scripts/master/read-only-fs.sh`
2. Run it:
`sudo bash read-only-fs.sh`
3. Shutdown your pi:
`sudo poweroff`

⁷ You can monitor this via ssh or wait for the zero to reboot and start playing.

⁸ During playback only, it provides no protection should power be removed while a USB host is writing to the partition exported via the mass storage gadget. That’s down to the USB host to worry about.

Appendix One – Remote Control a Pi zeroW/WH

Portions of this section are based on

<https://www.raspberrypi.org/documentation/configuration/wireless/access-point.md>

This section assumes you want to create a private Wi-Fi network. If you're connecting to an existing one skip to step 9

1. Login to your Pi zero
2. Remount “/” read write (if you reboot during these steps you'll need to do this again)⁹:

```
sudo mount -o remount,rw /
```
3. Stop mpd:

```
sudo service mpd stop
```
4. Install the required packages:

```
sudo apt install hostapd dnsmasq
```
5. As they're not yet configured, stop them:

```
sudo systemctl stop hostapd  
sudo systemctl stop dnsmasq
```
6. Set a static IP address for the wifi interface:
 1. Open /etc/dhcpd.conf in your favourite editor, this will need sudo.
 2. At the bottom of the file add:

```
interface wlan0  
static ip_address=10.0.0.1/24
```
 3. Save and close the file
 4. Restart the dhcpd daemon:

```
sudo service dhcpd restart
```
7. Configure dnsmasq
 1. Move the default configuration file out of the way:

```
sudo mv /etc/dnsmasq.conf /etc/dnsmasq.conf.orig
```
 2. Open /etc/dnsmasq.conf in your favourite editor, this will need sudo. You should see an empty file.
 3. Insert the following:

```
interface=wlan0  
dhcp-range=10.0.0.2,10.0.0.25,255.255.255.0,24h
```
 4. Save and close the file
8. Configure hostapd
 1. Open /etc/hostapd/hostapd.conf in your favourite editor, this will need sudo. You should see an empty file.

9 Not needed if you've not converted to a read only filesystem

2. Insert the following, changing “wlan0”, “NameOfNetwork”, “GB”, and “AardvarkBadgerHedgehog” as required:


```

interface=wlan0
driver=nl80211
ssid=NameOfNetwork
country_code=GB
hw_mode=g
channel=7
wmm_enabled=0
macaddr_acl=0
auth_algs=1
ignore_broadcast_ssid=0
wpa=2
wpa_passphrase=AardvarkBadgerHedgehog
wpa_key_mgmt=WPA-PSK
wpa_pairwise=TKIP
rsn_pairwise=CCMP
      
```
3. Save and close the file
4. Configure hostapd’s logging¹⁰:
 1. Open /etc/init.d/hostapd in your favourite editor, this will need sudo.
 2. Find the line starting with “PIDFILE=” add a new line below it:


```
LOGFILE=/var/log/hostapd.log
```
 3. Find the line reading:


```
DAEMON_OPTS="-B -P $PIDFILE $DAEMON_OPTS $DAEMON_CONF"
```
 4. Change it to:


```
DAEMON_OPTS="-f $LOGFILE -B -P $PIDFILE $DAEMON_OPTS $DAEMON_CONF"
```
5. Save and close the file
6. Tell the system where to find hostapd.conf:
 1. Open /etc/default/hostapd in your favourite editor, this will need sudo.
 2. Find the line that starts “#DAEMON_CONF” and change it to:


```
DAEMON_CONF="/etc/hostapd/hostapd.conf"
```
 3. Save and close the file
9. Configure mpd to allow remote access
 1. Open /etc/mpd.conf in your favourite editor, this will need sudo.
 2. Find the line that reads ‘bind_to_address “localhost”’
 3. Change it to:


```
bind_to_address “any”
```
 4. Save and close the file
10. Remount “/” read only¹¹¹²:


```
sudo mount -o remount,ro /
```
11. Restart services:


```

sudo systemctl start hostapd
sudo systemctl start dnsmasq
sudo service start mpd
      
```

You should now be able to connect to the access point configured above and control mpd via one of its many clients.

10 Again, skipping this will cause problems after moving to a read only root

11 Not needed if you’ve not converted to a read only filesystem

12 Steps 10 and 11 may be replaced with a reboot.

Appendix Two – Sharing Your Music

Any device that can act as a USB host for mass storage devices and that supports FAT32 formatted drives can access your music by connecting a cable to the USB data port.

To make your music available over a network:

1. Make sure your Pi zero has access to the internet.
2. Login
3. Remount “/” read write (if you reboot during these steps you’ll need to do this again)¹³:
`sudo mount -o remount,rw /`
4. Update package lists
`sudo apt update`
5. Install samba
`sudo apt install samba samba-common-bin`
6. Backup the default smb.conf
`sudo mv /etc/samba/smb.conf /etc/samba/smb.conf.orig`
7. Open /etc/samba/smb.conf in your favourite editor, this will need sudo. You should see an empty file.
8. Insert the following:

```
[global]
workgroup = WORKGROUP
dns proxy = no
log file = /var/log/samba.log
syslog = 0
server role = standalone server
security = user
unix password sync = no
map to guest = bad user
show add printer wizard = no

[music]
comment = %h Music
path = /music
browseable = yes
read only = yes
guest ok = yes
guest only = yes
```
9. Save and close the file
10. Remount “/” read only¹⁴:
`sudo mount -o remount,ro /`
11. Restart samba:
`sudo systemctl restart smbd`

The above creates a read only share available to anyone with network access to your Pi. Refer to the samba documentation (<https://www.samba.org/samba/docs/>) if you want to make it more secure.

¹³ Not needed if you’ve not converted to a read only filesystem

¹⁴ Not needed if you’ve not converted to a read only filesystem

¹⁵ Steps 10 and 11 may be replaced with a reboot.

Appendix Three – Airplay Receiver

This section is based on <https://github.com/mikebrady/shairport-sync/blob/master/CAR%20INSTALL.md>

The target Pi needs to be available over a network for this to be of use. See Appendix Two for one method of making it so. We'll be using Shairport Sync for airplay functionality.

1. Make sure your Pi zero has access to the internet.
2. Login
3. Remount "/" read write (if you reboot during these steps you'll need to do this again)¹⁶:
`sudo mount -o remount,rw /`
4. Update package lists
`sudo apt update`
5. Install dependencies for shairport-sync:
`apt-get install build-essential git xsltoman autoconf
automake libtool libdaemon-dev libpopt-dev libconfig-
dev libasound2-dev avahi-daemon libavahi-client-dev
libssl-dev`
6. Download Shairport Sync
`git clone https://github.com/mikebrady/shairport-sync.git`
7. Configure, build and install it
`cd shairport-sync
autoreconf -fi
./configure --sysconfdir=/etc --with-alsa --with-avahi
--with-ssl=openssl --with-systemd
make
sudo make install`
8. Open /etc/shairport-sync.conf in your favourite editor. This will need sudo.
9. Find the `sessioncontrol =` section
10. Insert the following between the braces:
`run_this_before_play_begins = "/usr/bin/mpc stop"
run_this_after_play_ends = "/usr/bin/mpc play"
wait_for_completion = "yes"`
11. Save and close the file
12. Enable auto start of shairport-sync
13. `sudo systemctl enable shairport-sync`
14. Remount "/" read only¹⁷:
`sudo mount -o remount,ro /`
15. Start shairport-sync
`sudo systemctl start shairport-sync`

Step 10 ensures that local playback is stopped when an airplay session is started and resumed when it is closed.

¹⁶ Not needed if you've not converted to a read only filesystem

¹⁷ Not needed if you've not converted to a read only filesystem