**Dementia Friendly Music Player: How to prepare the micro-SD card image**

21 March 2020

# Introduction

This document describes how to create a Dementia Friendly Music Player system image – a .img file. I (Ross) am probably the only person that needs this document. Mostly people making a Dementia Friendly Music Player will use the fruit (the .img file) of the process described here -- you don’t need to create your own custom .img file unless you really want to.

# No warranty

USE THESE DEMENTIA FRIENDLY MUSIC PLAYER (DQMUSICBOX) PLANS AND SYSTEM AT YOUR OWN RISK. THE DEMENTIA FRIENDLY MUSIC PLAYER PLANS ARE PROVIDED AS IS WITHOUT WARRANTY OF ANY KIND EITHER EXPRESSED OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. THE ENTIRE RISK AS TO THE QUALITY AND PERFORMANCE OF THE PLANS AND SYSTEM IS WITH YOU. SHOULD THE PLANS OR SYSTEM PROVE DEFECTIVE, YOU ASSUME THE COST OF ALL NECESSARY SERVICING, REPAIR OR CORRECTION. IN NO EVENT WILL ANY PARTY BE LIABLE TO YOU FOR DAMAGES, INCLUDING ANY GENERAL, SPECIAL, INCIDENTAL OR CONSEQUENTIAL DAMAGES ARISING OUT OF THE USE OR INABILITY TO USE THE PLANS OR SYSTEM.

# To create USB DAC version

The Dementia Friendly Music Player that looks like this uses a USB DAC.



## Flash DietPi onto the micro-SD card

Install DietPi i.e. follow these instructions: <http://dietpi.com/phpbb/viewtopic.php?f=8&t=9#p9>. I used [balenaEtcher](https://etcher.io/) (free) to write the disk image to the micro-SD card. This may take a while as the system updates itself.

## Boot & configure DietPi

Put the Pi on Ethernet. Move the micro-SD card to the Pi, boot DietPi. After a build process, you will be prompted to make some choices.  
  
DietPi-Config

|  |
| --- |
| Audio Options: Enable : Install ALSA to enable audio capabilities  Audio Options: Soundcard : usb-dac |

Software Optimized

|  |
| --- |
| Hardware Projects : RPi.GPIO [Install] |

Software Additional

|  |
| --- |
| System : ALSA  Development : Git Client |

Install

|  |
| --- |
| Install |

Software will be installed. Then your Pi will then reboot

## Install VLC (music player)

|  |
| --- |
| sudo apt-get install vlc-bin  sudo apt-get install vlc-plugin-base |

## adduser pi

|  |
| --- |
| sudo adduser pi |

## Install/clone dqmusicbox, enable

|  |
| --- |
| cd /home/pi  git clone <https://github.com/rosswesleyporter/dqmusicbox/>  sudo chmod 755 dqmusicbox/bin/dqmusicbox.py |

## Install Python bindings for VLC

|  |
| --- |
| cd /home/pi  sudo git clone https://github.com/oaubert/python-vlc  cp python-vlc/generated/2.2/vlc.py dqmusicbox/bin  chmod 755 dqmusicbox/bin/vlc.py |

## Add shell script to automatically start the musicbox

|  |
| --- |
| cd /home/pi  sudo cp dqmusicbox/bin/dqmusicbox.sh /etc/init.d  sudo chmod 755 /etc/init.d/dqmusicbox.sh  sudo update-rc.d dqmusicbox.sh defaults |

For more information, see Stephen Christopher Phillips’ [terrific page](http://blog.scphillips.com/posts/2013/07/getting-a-python-script-to-run-in-the-background-as-a-service-on-boot/).

## Configure such that USB drives mount automatically

The instructions below are from [pauliucxz](https://raspberrypi.stackexchange.com/users/66022/pauliucxz) in [StackExchange 66169](https://raspberrypi.stackexchange.com/questions/66169/auto-mount-usb-stick-on-plug-in-without-uuid), preserved below for clarity. I am quite thankful for that answer. The first USB drive will automatically mount as /media/usb1.

Install pmount

|  |
| --- |
| sudo apt-get install pmount |

Specify a udev rule by creating file /etc/udev/rules.d/usbstick.rules

|  |
| --- |
| ACTION=="add", KERNEL=="sd[a-z][0-9]", TAG+="systemd", ENV{SYSTEMD\_WANTS}="usbstick-handler@%k" |

Configure a system service by creating file /lib/systemd/system/usbstick-handler@.service

|  |
| --- |
| [Unit]  Description=Mount USB sticks  BindsTo=dev-%i.device  After=dev-%i.device  [Service]  Type=oneshot  RemainAfterExit=yes  ExecStart=/usr/local/bin/cpmount /dev/%I  ExecStop=/usr/bin/pumount /dev/%I |

Create the mount script file /usr/local/bin/cpmount

|  |
| --- |
| #!/bin/bash  if mountpoint -q /media/usb1  then  if mountpoint -q /media/usb2  then  if mountpoint -q /media/usb3  then  if mountpoint -1 /media/usb4  then  echo "No mountpoints available!"  #You can add more if you need  else  /usr/bin/pmount --umask 000 --noatime -w --sync $1 usb4  fi  else  /usr/bin/pmount --umask 000 --noatime -w --sync $1 usb3  fi  else  /usr/bin/pmount --umask 000 --noatime -w --sync $1 usb2  fi  else  /usr/bin/pmount --umask 000 --noatime -w --sync $1 usb1  fi |

Make the script executable

|  |
| --- |
| sudo chmod 755 /usr/local/bin/cpmount |

## Reboot

|  |
| --- |
| sudo reboot |

## Test

Make sure the music plays…

## Shutdown

Provided that the reboot went well, shutdown:

|  |
| --- |
| sudo shutdown –h now |

Then remove the micro-SD card.

## Use Win32DiskImager to create the master image

Remove the micro-SD card from your Pi and place in the card reader of your computer. Use Win32DiskImager to create an image of Dementia Friendly Music Player that you just nicely configured.

# To create built-in headphone jack version

The Dementia Friendly Music Players below use the Pi’s built-in headphone jack.

A close up of a device

Description automatically generatedA close up of a logo

Description automatically generated

## Flash DietPi onto the micro-SD card

Install DietPi i.e. follow these instructions: <http://dietpi.com/phpbb/viewtopic.php?f=8&t=9#p9>. I used DietPi 6.26 which was current as of this writing in November 2019. I used [balenaEtcher](https://etcher.io/) (free) to write the disk image to the micro-SD card. This may take a while as the system updates itself.

## Boot & configure DietPi

Put the Pi on Ethernet. Move the micro-SD card to the Pi, boot DietPi. After a build process (which will take a while), you will be prompted to make some choices.  
  
DietPi-Config

|  |
| --- |
| Audio Options: Enable : Install ALSA to enable audio capabilities  Audio Options: Soundcard : rpi-bcm2835-3.5mm : Onboard HQ: 3.5mm forced output |

Software Optimised

|  |
| --- |
| Hardware Projects : RPi.GPIO [Install] |

Software Additional

|  |
| --- |
| Development : Git Client |

Install

|  |
| --- |
| Install |

Software will be installed. Then your Pi will then reboot

## Install VLC (music player)

|  |
| --- |
| sudo apt-get install vlc-bin  sudo apt-get install vlc-plugin-base |

## adduser pi

|  |
| --- |
| sudo adduser pi |

## Install/clone dqmusicbox, enable

|  |
| --- |
| cd /home/pi  git clone <https://github.com/rosswesleyporter/dqmusicbox/>  sudo chmod 755 dqmusicbox/bin/dqmusicbox.py |

## Install Python bindings for VLC

|  |
| --- |
| cd /home/pi  sudo git clone https://github.com/oaubert/python-vlc  cp python-vlc/generated/2.2/vlc.py dqmusicbox/bin  chmod 755 dqmusicbox/bin/vlc.py |

## Add shell script to automatically start the musicbox

|  |
| --- |
| cd /home/pi  sudo cp dqmusicbox/bin/dqmusicbox.sh /etc/init.d  sudo chmod 755 /etc/init.d/dqmusicbox.sh  sudo update-rc.d dqmusicbox.sh defaults |

For more information, see Stephen Christopher Phillips’ [terrific page](http://blog.scphillips.com/posts/2013/07/getting-a-python-script-to-run-in-the-background-as-a-service-on-boot/).

## Configure such that USB drives mount automatically

The instructions below are from [pauliucxz](https://raspberrypi.stackexchange.com/users/66022/pauliucxz) in [StackExchange 66169](https://raspberrypi.stackexchange.com/questions/66169/auto-mount-usb-stick-on-plug-in-without-uuid), preserved below for clarity. I am quite thankful for that answer. The first USB drive will automatically mount as /media/usb1.

Install pmount

|  |
| --- |
| sudo apt-get install pmount |

Specify a udev rule by creating file /etc/udev/rules.d/usbstick.rules

|  |
| --- |
| ACTION=="add", KERNEL=="sd[a-z][0-9]", TAG+="systemd", ENV{SYSTEMD\_WANTS}="usbstick-handler@%k" |

Configure a system service by creating file /lib/systemd/system/usbstick-handler@.service

|  |
| --- |
| [Unit]  Description=Mount USB sticks  BindsTo=dev-%i.device  After=dev-%i.device  [Service]  Type=oneshot  RemainAfterExit=yes  ExecStart=/usr/local/bin/cpmount /dev/%I  ExecStop=/usr/bin/pumount /dev/%I |

Create the mount script file /usr/local/bin/cpmount

|  |
| --- |
| #!/bin/bash  if mountpoint -q /media/usb1  then  if mountpoint -q /media/usb2  then  if mountpoint -q /media/usb3  then  if mountpoint -1 /media/usb4  then  echo "No mountpoints available!"  #You can add more if you need  else  /usr/bin/pmount --umask 000 --noatime -w --sync $1 usb4  fi  else  /usr/bin/pmount --umask 000 --noatime -w --sync $1 usb3  fi  else  /usr/bin/pmount --umask 000 --noatime -w --sync $1 usb2  fi  else  /usr/bin/pmount --umask 000 --noatime -w --sync $1 usb1  fi |

Make the script executable

|  |
| --- |
| sudo chmod 755 /usr/local/bin/cpmount |

## Reboot

|  |
| --- |
| sudo reboot |

## Test

Make sure the music plays…

## Shutdown

Provided that the reboot went well, shutdown:

|  |
| --- |
| sudo shutdown –h now |

Then remove the micro-SD card.

## Use Win32DiskImager to create the master image

Remove the micro-SD card from your Pi and place in the card reader of your computer. Use Win32DiskImager to create an image of Dementia Friendly Music Player that you just nicely configured.