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

24 September 2023

1 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.

2 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.

3 To create built-in headphone jack version

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

3.1 Flash DietPi onto the micro-SD card

Install DietPi i.e. follow these instructions: https://dietpi.com/docs/install/. I used DietPi 8.21.0 which was current as of this writing in September 2023. I used balenaEtcher (free) to write the disk image to the micro-SD card. This may take a while as the system updates itself.

3.2 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: Sound Card : Onboard 3.5mm output
```

DietPi-Software

```
Hardware Projects : Python 3 RPi.GPIO
System: ALSA : Advanced Linux Sound Architecture
Development : Git : clone and manage …
Development : Python 3: Runtime system, pip package installer, …
Install
```

3.3 Install pmount

```
sudo apt-get install pmount
```

3.4 Install VLC (music player)

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

3.5 adduser pi

```
sudo adduser pi
```

3.6 Install/clone dqmusicbox, enable

```
cd /home/pi
git clone --depth 1 https://github.com/rosswesleyporter/dqmusicbox/
sudo chmod 755 dqmusicbox/bin/dqmusicbox.py
```

3.7 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
```

3.8 Add shell script to automatically start dqmusicbox

```
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.

3.9 Configure such that USB drives mount automatically

The instructions below are from pauliucxz in StackExchange 66169, preserved below for clarity. I am quite thankful for that answer. The first USB drive will automatically mount as /media/usb1.

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
```

3.10 Reboot

```
sudo reboot
```

3.11 Test

Make sure the music plays…

3.12 Shutdown

Provided that the reboot went well, shutdown:

```
sudo shutdown -h now
```

Then remove the micro-SD card.

3.13 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.

4 To create USB DAC version

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

Use the image that you created above for the headphone version and make this change:

dietpi-config

```
Audio Options: Sound Card : USB Audio DAC (any)
```

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.