Music Player Moode

Hardware

Raspberry Pi Zero W	
Pimoroni Pirate Audio (Headphone)	https://shop.pimoroni.com/products/pirate-audio-headphone-amp
Pimoroni Pirate Audio Case	https://www.thingiverse.com/thing:4087948
Alternative case ideas	https://www.yeggi.com/q/pimoroni+pirate+audio/
External passive speakers with 3.5mm plug	
3.5mm Headphone Volume Control	https://smile.amazon.co.uk/gp/product/B00Y1MYSYW/
Buttons (need case and button adjustment)	https://www.tinkercad.com/things/8YwzBrOzCay

System installation 8.0.2

Download Moode ISO https://moodeaudio.org/

```
wget
https://github.com/moode-player/moode/releases/download/r802prod/image_2022-
03-26-moode-r802-lite.zip
```

Main install instructions with reference to auto-install:
https://github.com/moode-player/moode/blob/master/www/setup.txt

Then flash using Balena Etcher or similar, for Belana, unzipping is not required!
https://www.balena.io/etcher/

```
sudo apt-get install belana-etcher-electron
```

```
sudo apt-get install debian-keyring debian-archive-keyring apt-transport-
https ca-certificates gnupg
curl -1sLf
"https://dl.cloudsmith.io/public/balena/etcher/gpg.70528471AFF9A051.key" |
sudo apt-key add
cat <<EOF | sudo tee /etc/apt/sources.list.d/balena-etcher.list
# Source: Cloudsmith (support@cloudsmith.io)
# Repository: balena / etcher
# Description: Flash OS images to SD cards & USB drives, safely and easily.
deb https://dl.cloudsmith.io/public/balena/etcher/deb/ubuntu focal main
deb-src https://dl.cloudsmith.io/public/balena/etcher/deb/ubuntu focal main
EOF
sudo apt-get update
sudo apt-get install balana-etcher-electron
```

Mount the SDCard which will make the boot partition accessible Copy the file /boot/moodecfg.ini.default to your PC, Mac or Linux client Rename it to moodecfg.ini Edit the settings as needed (wlan, country, volume steps, etc) Copy moodecfg.ini to /boot/

SSH Server enabled by default: username: pi password: moodeaudio

moodecfg adjustments for UK and personal preferences:

moodecfg.ini

```
timezone = "Europe/London"
keyboard = "gb"

p3bt = "0"

replaygain = "track"

volume_normalization = "yes"

volume_step_limit = "2"

wlanssid = "xxx"
wlanpwd = "xxx"
wlancountry = "GB"

first_use_help = "No"
```

Config.txt adjustments for Pirate Audio on PiZeroW

/boot/config.txt

```
[pi0]
# Disable the ACT LED on the Pi 1 and Zero
dtparam=act_led_trigger=none
dtparam=act_led_activelow=on

[cm4]
otg_mode=1

[pi4]
hdmi_force_hotplug:0=1
hdmi_force_hotplug:1=1

[all]
#Disable boot splash screen
disable_splash=1
disable_overscan=1
hdmi_drive=2
#Disable HDMI on boot
hdmi_blanking=2
hdmi_force_edid_audio=1
hdmi_force_hotplug=1
hdmi_group=1
```

```
# Uncomment some or all of these to enable the optional hardware
interfaces
dtparam=i2c_arm=on
dtparam=i2s=on
#Switch off onboard audio
dtparam=audio=off
#dtoverlay=disable-wifi
dtoverlay=disable-bt

#Configure Pimoroni Pirate audio DAC
dtoverlay=i2s-mmap
dtoverlay=hifiberry-dac
gpio=25=op,dh

#Enable SPI for display of Pimoroni Pirate Audio
dtparam=spi=on

#Set GPU memory to lowest value in /boot/config.txt:
gpu_mem=16
```

possible further requirement for HDMI: Disable HDMI port on boot (power saving during headless operation) /usr/bin/tvservice -o (-p to re-enable) Add the line to /etc/rc.local to disable HDMI on boot.

Now insert the SD-Card into the Pi and power it up. Then connect to web interface via http://moode/ or IP if known.

Configure Moode: Audio Config: “HiFiBerry DAC” or “Pimoroni pHAT DAC” System: Disable ACT LED, Disable HDMI Enable GPIO button handler and set to: four buttons, active low connected to BCM 5, 6, 16, and 24 (A, B, X, Y respectively). Replace all spaces with commas in the command. See http://moodeaudio.org/forum/showthread.php?tid=1381&page=2&highlight=gpio

```
BTN1: 5  mpc,toggle
BTN2: 6  /var/www/vol.sh,-dn,5
BTN3: 16 mpc,next
BTN4: 24 /var/www/vol.sh,-up,5
Debounce 1000ms

#mpc,volume,-5
#mpc,volume,+5
#/var/www/vol.sh,-mute
```

Configure vi via ssh:

```
vi ~/.vimrc
```

```
set tabstop=4
set shiftwidth=4
set softtabstop=4
set expandtab
```

```
set nocompatible
```

Remove Bluetooth config in Moode main menu when not in use:

```
sudo vi /var/www/header.php
```

remove line referencing 'blu-config.php'

Pirate Audio TFT Cover Art v0.0.6

https://github.com/rusconi/TFT-MoodeCoverArt fork: https://github.com/pachisb/TFT-MoodeCoverArt

Note: 16-25% CPU usage on RPI-0w !

Enable Metadata file in System → Local Services in moode.

```
#enable spi if not already enabled
sudo raspi-config
sudo reboot
```

```
sudo apt-get update
sudo apt-get install git python3-rpi.gpio python3-spidev python3-pip
python3-pil python3-numpy libatlas-base-dev
sudo pip3 install mediafile pyyaml RPI-ST7789
cd /home/pi
git clone https://github.com/rusconi/TFT-MoodeCoverArt.git
cd TFT-MoodeCoverArt/
vi config.yml
chmod 777 *.sh

#test
python3 tft_moode_coverart.py
#if it works, install service
./install_service.sh
```

Bugfix: Wrap mf = MediaFile(fp) section in try/except and indent it

tft_moode_coverart.py

```
    else:
         if 'file' in metaDict:
             if len(metaDict['file']) > 0:

                 fp = '/var/lib/mpd/music/' + metaDict['file']
                 try:
                     mf = MediaFile(fp)
                     if mf.art:
                         cover = Image.open(BytesIO(mf.art))
```

```
                        return cover
                    else:
                        for it in covers:
                            cp = os.path.dirname(fp) + '/' + it
                            if path.exists(cp):
                                cover = Image.open(cp)
                                return cover
                except Exceptions:
                    pass
    return cover
```

IP Address mod:

```
#add to top
import socket

#add before 'def main():'
def get_ip():
    s = socket.socket(socket.AF_INET, socket.SOCK_DGRAM)
    try:
        # doesn't even have to be reachable
        s.connect(('10.255.255.255', 1))
        IP = s.getsockname()[0]
    except Exception:
        IP = '127.0.0.1'
    finally:
        s.close()
    return IP

# add in def main(): after variable initialisation
    #get ip address, add to image and show on display
    ip = get_ip()
    draw.rectangle((0,0,240,240), fill=(0,0,0))
    txt = f"Visit http://{ip} to select content."
    mlw, mlh = draw.multiline_textsize(txt, font=font_m, spacing=4)
    draw.multiline_text(((WIDTH-mlw)//2, 20), txt, fill=(255,255,255),
font=font_m, spacing=4, align="center")
    disp.display(img)
```

Service

/lib/systemd/system/tft-moodecoverart.service

```
[Unit]
Description=TFT-MoodeCoverArt Display
Requires=mpd.socket mpd.service
After=mpd.socket mpd.service

[Service]
Type=simple
```

```
ExecStart=/home/pi/TFT-MoodeCoverArt/tft-moodecoverart.py &
ExecStartPre=/bin/sleep 15
#ExecStop=/home/pi/TFT-MoodeCoverArt/tft-moodecoverart.sh -q
ExecStop=/home/pi/TFT-MoodeCoverArt/shutdown.py &
Restart=on-abort
StandardOutput=syslog
StandardError=syslog

[Install]
WantedBy=multi-user.target
```

Boot logo on PirateAudio

/home/pi/TFT-MoodeCoverArt/boot.py

```
#!/usr/bin/env python3

import time
from PIL import ImageFont, Image, ImageDraw
import os
import ST7789
import sys

# get the path of the script
script_path = os.path.dirname(os.path.abspath(__file__))
# set script path as current directory
os.chdir(script_path)


# Create ST7789 LCD display class.
disp = ST7789.ST7789(
    rotation=90,  # Needed to display the right way up on Pirate Audio
    port=0,       # SPI port
    cs=1,         # SPI port Chip-select channel
    dc=9,         # BCM pin used for data/command
    backlight=13,
    spi_speed_hz=80 * 1000 * 1000
)

# Initialize display.
disp.begin()

WIDTH = 240
HEIGHT = 240
font_s = ImageFont.truetype(script_path + '/fonts/Roboto-Medium.ttf',
20)
font_m = ImageFont.truetype(script_path + '/fonts/Roboto-Medium.ttf',
```

```
24)
font_l = ImageFont.truetype(script_path + '/fonts/Roboto-Medium.ttf',
30)


def bootmessage():
    print('bootmessage called')
    disp.set_backlight(True)
    img = Image.new('RGBA', (240, 240), color=(0, 0, 0, 25))
    img = Image.open('images/default-cover-v6.jpg')
    draw = ImageDraw.Draw(img, 'RGBA')
    message = 'booting ...'
    draw.text((10, 200), message, font=font_m, fill=(255, 255, 255))
    disp.display(img)
    sys.exit()


try:
    bootmessage()
except SystemExit:
    print('Systemexit called')
    pass
```

```
chmod 755 /home/pi/TFT-MoodeCoverArt/boot.py
```

/lib/systemd/system/tft-boot.service

```
[Unit]
Description=TFT Boot Message
Before=basic.target
After=local-fs.target sysinit.target
DefaultDependencies=no

[Service]
Type=oneshot
ExecStart=/home/pi/TFT-MoodeCoverArt/boot.py

[Install]
WantedBy=basic.target
```

```
systemctl enable tft-boot
```

Pirate Audio Cover related links

https://github.com/pimoroni/pidi/blob/master/pidi/client.py
https://github.com/pimoroni/pidi-plugins/blob/master/pidi-display-pil/pidi_display_pil/__init__.py

https://github.com/pimoroni/pidi-plugins/blob/master/pidi-display-pil/pidi_display_pil/__init__.py
https://github.com/pimoroni/st7789-python https://github.com/pimoroni/pidi-spotify
https://github.com/pimoroni/pirate-audio/issues/17
https://github.com/pimoroni/mopidy-pidi/blob/master/mopidy_pidi/frontend.py
https://github.com/pimoroni/pirate-audio/blob/master/examples/backlight-pwm.py
https://github.com/AnonTester/TFT-MoodeCoverArt
http://moodeaudio.org/forum/showthread.php?tid=2210

https://ideatrash.net/2020/06/simple-smart-playlists-for-mpd-that-work.html
https://bbs.archlinux.org/viewtopic.php?id=76385

https://github.com/Ax-LED/volumio-pirate-audio

Long Button Press MOD

Adjust gpio-buttons.py script as per following example for first configured button:

/var/www/daemon/gpio_buttons.py

```
    if str(row['id']) == '1' and row['enabled'] == '1':
        sw_1_pin = int(row['pin'])
        sw_1_cmd = row['command'].split(',')
        sw_1_cmd = [x.strip() for x in sw_1_cmd]
        sw_1_cmd_2 = ["/var/www/command/sleeptimer.php","1800"]
        sw_1_cmd_3 = ["/var/www/command/sleeptimer.php","3600"]
        GPIO.setup(sw_1_pin, GPIO.IN, pull_up_down=GPIO.PUD_UP)

        def sw_1_event(channel):
            start_time = time.time()
            time.sleep(0.005) # edge debounce of 5 ms
            # only deal with valid edges
            while GPIO.input(channel) == 0:  # wait for button up
                pass
            buttonTime = time.time() - start_time  #calc button press
            print('time ' + str(buttonTime))
            #if GPIO.input(channel) == 1:
            if buttonTime < 2:
                print('short press')
                subprocess.call(sw_1_cmd)
            elif 2 <= buttonTime < 4:  # long press
                print('long press')
                subprocess.call(sw_1_cmd_2)
            elif buttonTime > 4:  # very long press
                print('very long press')
                subprocess.call(sw_1_cmd_3)

        GPIO.add_event_detect(sw_1_pin, GPIO.FALLING,
callback=sw_1_event, bouncetime=bounce_time)
        print(str(datetime.datetime.now())[:19] + ' sw_1: pin=' +
```

```
            str(sw_1_pin) + ', enabled=' + row['enabled'] +
            ', bounce_time=' + str(bounce_time) + ', cmd=' +
row['command'])
```

/var/www/command/sleeptimer.php

```
#!/usr/bin/php
<?php
# Config values
$fadelength=5; #the duration of the fade out in seconds
$steptime=0.5; #the amount in seconds of each step


if($argc<2 || !isset($argv[1])) {
    echo "This script is a sleep timer to stop playback after a
while.\n";
    echo "You need to pass the amount of seconds to sleep to this
script!\n";
    echo "To stop playback after an hour use:\n".$argv[0]." 3600\n";
    exit;
}

sleep($argv[1]);

$currentvolume=filter_var(exec('/usr/bin/mpc volume'),
FILTER_SANITIZE_NUMBER_INT);
echo "Current volume: $currentvolume\n";
$stepamount = $fadelength/$steptime;
$stepadjust = floor($currentvolume/$stepamount);

for ( $i=1; $i<=$stepamount; $i++ ) {
    $vol = exec('/usr/bin/mpc volume -'.$stepadjust);
    echo "Volume adjusted to ".$vol."\n";
    usleep ($steptime*1000000);
}

echo "Stopping playback\n";
exec('/usr/bin/mpc stop');

sleep(1);
# Resetting volume
exec('/usr/bin/mpc volume '.$currentvolume);
echo "Resetting volume to $currentvolume\n";
```

```
sudo chmod 755 /var/www/command/sleeptimer.php
sudo killall -9 gpio_buttons.py
sudo /var/www/daemon/gpio_buttons.py &
```

Other notes:

```
#
https://learn.pimoroni.com/tutorial/sandyj/getting-started-with-pirate-audio

#This will install python3 pip,wheel and pirate audio modules
#Then add mopidy apt sources and install mopidy including mopidy-spotify
#Then install mopidy-iris web interface and Pirate Audio plugins
#And create system service to autostart mopidy

SMB mount the manual way (/etc/rc.local or /etc/fstab):
#Note experiment with rsize=61440 option and/or use nounix mount option
//192.168.1.6/music     /media/music    cifs
username=media,password=media,iocharset=utf8,noperm,file_mode=0644,dir_mode=
0755,users,rsize=61440,nounix       0       0

sudo mkdir /media/music
```

Moode tips

http://moodeaudio.org/forum/showthread.php?tid=803 MPD settings /etc/mpd.conf that control whether to automatically update the database when files are changed. Refer to this link https://github.com/MusicPlayerDaemon/MPD/blob/master/doc/mpd.conf.5 for information.

auto_update [yes or no] This specifies the whether to support automatic update of music database when files are changed in music_directory. The default is to disable autoupdate of database.

auto_update_depth [N] Limit the depth of the directories being watched, 0 means only watch the music directory itself. There is no limit by default.

in mean time you can ssh to moode and use this to add n last days

.bash_profile

```
function add-recents {
  find /media -type f -mtime -$1 | sed 's/\/media/USB/g' | mpc add
}
```

then 'add-recents 60' adds last 60 days off music

Library update: http://moode/command/?cmd=libupd-submit.php or

```
php /var/www/libupd-submit.php
```

instead of mpc commands, it should update MPD and covers.

You also can clear the library cache after the update:

```
mpc -w update
truncate /var/local/www/libcache.json --size 0
```

Add/Remove Radio Stations

Radio stations are stored as .pls files in /var/lib/mpd/music/RADIO

MPD Playlist Folder

/var/lib/mpd/playlists/

Generate playlists of recent music

~/newm3u.sh

```
#!/bin/bash
base_dir=/mnt/
music_dir=NAS/music/other/deemix\ Music/
playlistweek=/var/lib/mpd/playlists/New\ last\ week.m3u
playlistmonth=/var/lib/mpd/playlists/New\ last\ month.m3u
playlist2month=/var/lib/mpd/playlists/New\ last\ 2\ months.m3u

if [ "$base_dir$music_dir" -nt "$playlist" ] || [ ! -f "$playlist" ];
then
  cd $base_dir
  find "$music_dir" -type f -mtime -7 -iname "*.mp3" -o -iname "*.ogg"
> "$playlistweek"
fi

if [ "$base_dir$music_dir" -nt "$playlist" ] || [ ! -f "$playlist" ];
then
  cd $base_dir
  find "$music_dir" -type f -mtime -31 -iname "*.mp3" -o -iname "*.ogg"
> "$playlistmonth"
fi

if [ "$base_dir$music_dir" -nt "$playlist" ] || [ ! -f "$playlist" ];
then
  cd $base_dir
  find "$music_dir" -type f -mtime -61 -iname "*.mp3" -o -iname "*.ogg"
> "$playlist2month"
fi
```

```
chmod 755 ~/newm3u.sh
```