

# LightShowPi show manager

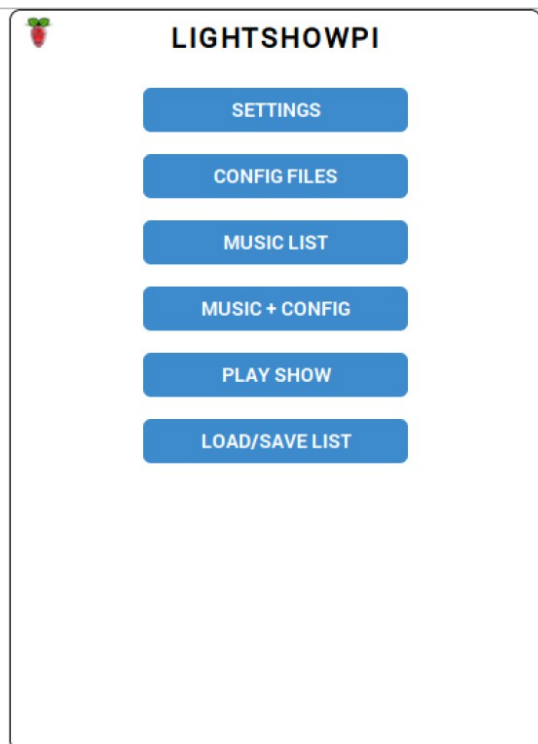
## Requirements:

- REMI: Python REMote Interface library.
- <https://github.com/dddomodossola/remi>
- `pip install git+https://github.com/dddomodossola/remi.git`

*Screen shots and explanation about how the app works.*

*This is a preliminary version of the app that I want to share to have feedback and complete it in the best way.*

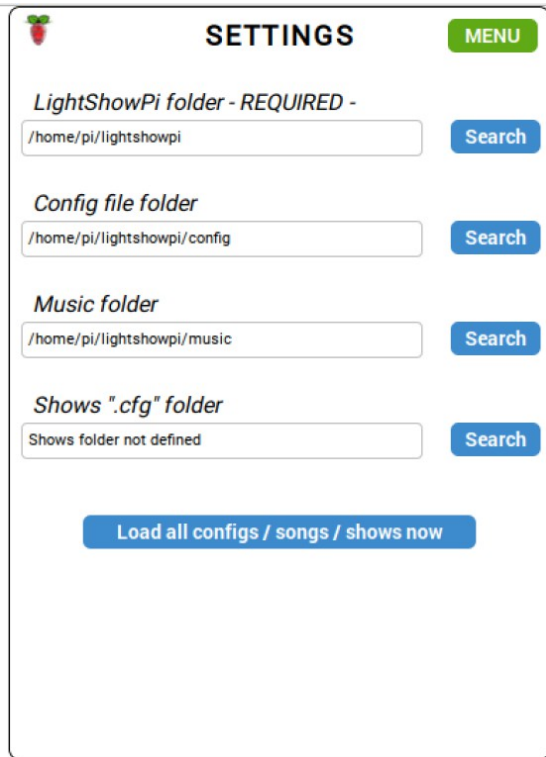
## MAIN PAGE



- *SETTINGS*: allows you to define the location of LSP, songs folder, shows folder,...
- *CONFIG FILES*: allows you to define the “default” config file and upload all the config files you need.
- *MUSIC LIST*: allows you to create and order your songs and shows list.
- *MUSIC + CONFIG*: allows you to assign specific config file to the songs.
- *PLAY SHOW*: allows you to play your show.
- *LOAD/SAVE LIST*: allows to save and load a show.

## SETTINGS PAGE

This is the page where locate the folders for LightShowPi, config files, song files and show files.



**SETTINGS** MENU

*LightShowPi folder - REQUIRED -*  
/home/pi/lightshowpi Search

*Config file folder*  
/home/pi/lightshowpi/config Search

*Music folder*  
/home/pi/lightshowpi/music Search

*Shows ".cfg" folder*  
Shows folder not defined Search

Load all configs / songs / shows now

- *MENU*: move you back to the main page.
- *LightShowPi folder*: is were the program is located. It checks automatically for the config folder and the py folder, if not found generates an error.
- *Config file folder*: could be a different folder where your config files are located.
- *Music folder*: it's a folder where your music is located.
- *Shows ".cfg" folder*: it's the folder where your ".cfg" show files are located. These are called with "Prepostshow.py" program and are used only to run a "PRESHOW" or a ".py PRESHOW". Follow examples later.
- *Load all config / song...*: load all files config .cfg, music (various extensions) and

shows.cfg in the right location (see below). Load can be used multiple times, maybe to load multiple songs from different folders. It doesn't load duplicate files, this means that, if you press it two times in a row, the first time it loads all the files and the second time it doesn't do anything, except if it finds some new files.

## CONFIG PAGE

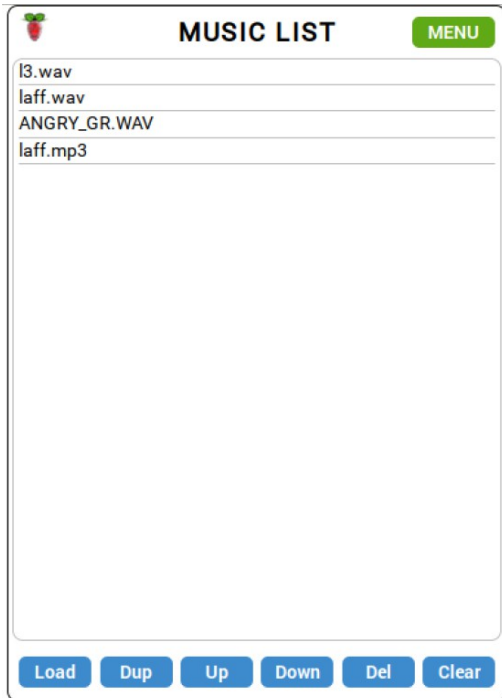
The first file at the top of the list is the default one. It is assigned automatically to all the songs every time you generate a new songs+configs list (see “MUSIC + CONFIG”). In the picture below “overrides.cfg” is the default one. It's not possible have a duplicate config file. Each song could have a different config file assigned. The idea behind this is that, maybe, a single config file doesn't fit well for all the different songs in a list. So, it's possible “tune/create” different config files and assign them to different songs.



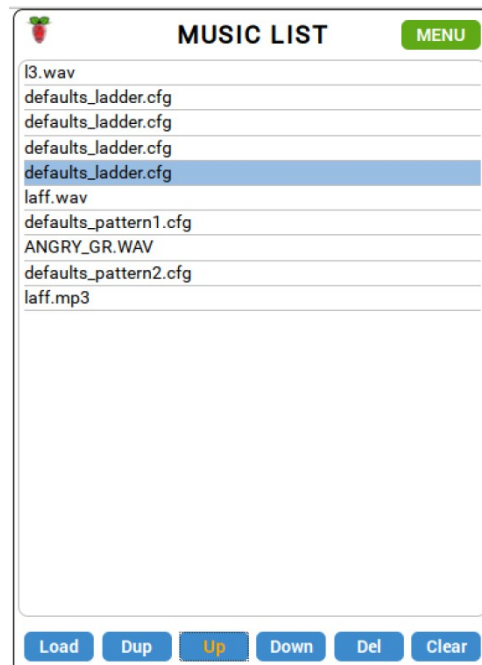
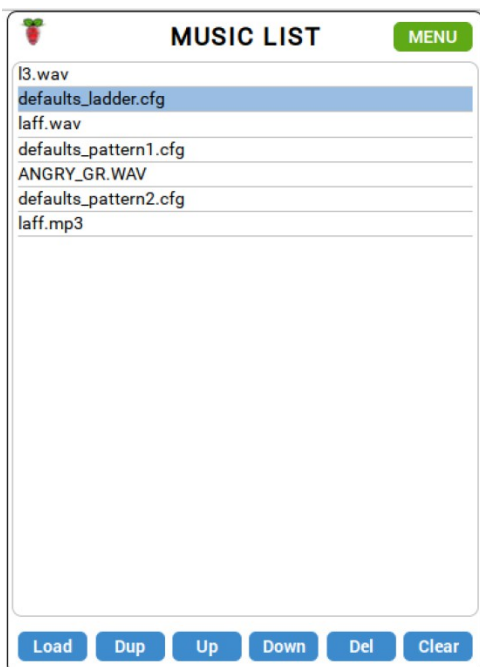
- *MENU*: move you back to the main page.
- *Load*: allows you to load other config files.
- *Default*: sets the selected file as the “default” one. Moves it at the top of the list.
- *Remove*: cancels the selected file.
- *Clear list*: cancels all files.

## MUSIC LIST PAGE

This is the list of songs and shows. In the first picture below the list is filled with what I loaded from the “Config page”, you can see there are only music files. In the second picture I've loaded some show files “.cfg” and rearranged the order of the list. It's possible to duplicate a song or a show in the list (see third picture).




- *MENU*: move you back to the main page.
- *Load*: allows you to load other songs or show files.
- *Dup*: allows you to duplicate a song or show.
- *Up*: allows you to move up the selected file.
- *Down*: allows you to move down the selected file.
- *Del*: cancels the selected file.
- *Clear*: cancels all files.



## MUSIC + CONFIG / FINAL MUSIC LIST PAGE

This is the final list of the songs and the shows that you will play. The first time you access to this page, it is empty. You have to “Update” the list to create the final list with songs+configs or shows (there is no config assigned to a show, the show is the config file). In the second picture you can see that below each song or show there is the config file assigned to it (“none” for the show). In this case if you look at the list the default file assigned to the songs is “overrides.cfg”, that is the default file we defined in the CONFIG page.

 **FINAL MUSIC LIST** MENU

Update list Generate play list

 **FINAL MUSIC LIST** MENU

l3.wav
--> overrides.cfg
defaults_ladder.cfg
--> none
defaults_ladder.cfg
--> none
defaults_ladder.cfg
--> none
defaults_ladder.cfg
--> none
laff.wav
--> overrides.cfg
defaults_pattern1.cfg
--> none
ANGRY_GR.WAV
--> overrides.cfg
defaults_pattern2.cfg
--> none
laff.mp3
--> overrides.cfg

Update list Generate play list

If you click on a song (ex. laff.mp3, the last one), it shows a new window where you can select which config file assign to that song. In this case in the new windows you can see: the name of the file you are changing the config file and the file I've selected for this example "defaults\_orig.cfg". In the second picture you can see the new config file assigned to laff.mp3.

The last thing to do is "Generate the play list". This creates an independent list that you can play in the play page. Independent because, at this point, you can continue to work on a new list or manipulate the existing one without interfere with the played one. Only when you will Generate the new list, this automatically overrides the actual playing list with the new one.

CONFIG LIST

Select the Config file for:  
**laff.mp3**

overrides.cfg

defaults\_ladder.cfg

state.cfg

defaults\_show.cfg

defaults\_stream.cfg

defaults\_pattern2.cfg

defaults\_pattern1.cfg

led1.defaults.cfg

defaults\_orig.cfg

defaults.cfg

Cancel

Ok

 FINAL MUSIC LIST 

MENU

l3.wav

--> overrides.cfg

defaults\_ladder.cfg

--> none

defaults\_ladder.cfg

--> none

defaults\_ladder.cfg

--> none

defaults\_ladder.cfg

--> none

laff.wav

--> overrides.cfg

defaults\_pattern1.cfg

--> none

ANGRY\_GR.WAV

--> overrides.cfg

defaults\_pattern2.cfg

--> none

laff.mp3

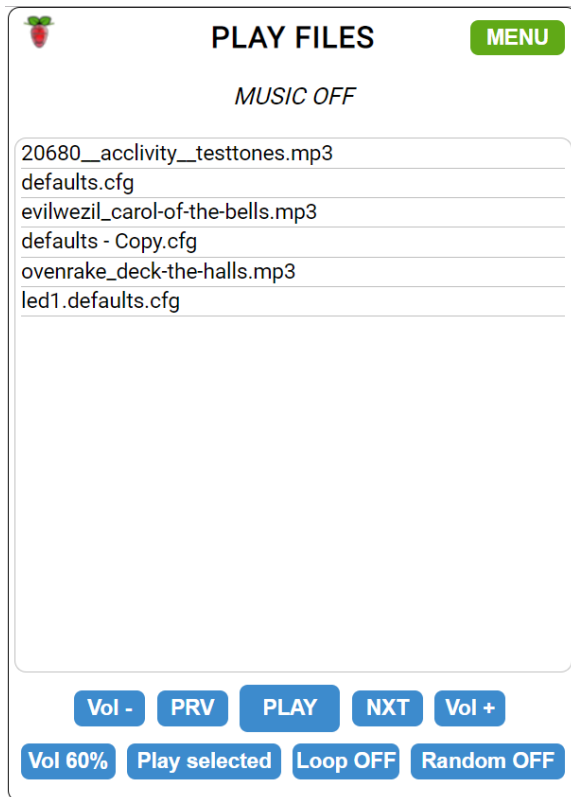
--> defaults\_orig.cfg

Update list

Generate play list

## PLAY FILES PAGE

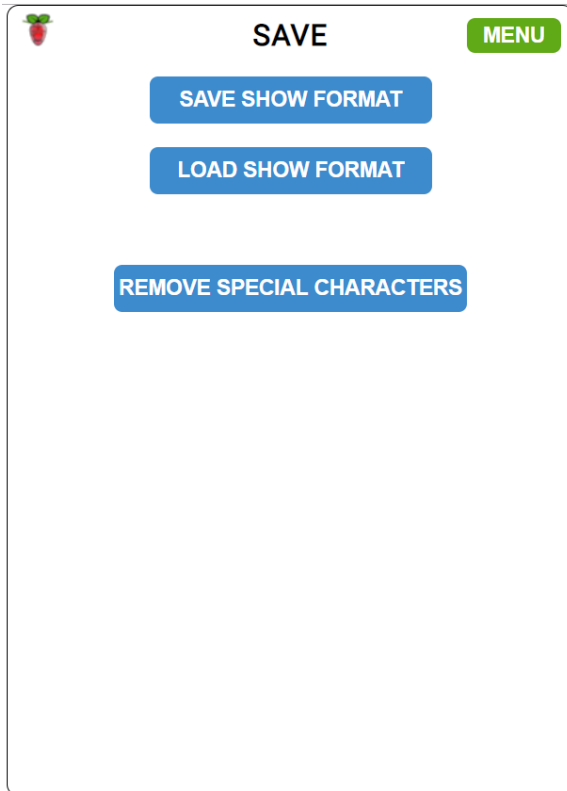
This is where you can play your show.



- *MENU*: move you back to the main page.
- *PLAY/STOP*: start and stop the music. Play starts always from the first song.
- *PRV*: plays the previous song.
- *NXT*: plays the next song.
- *Vol - / Vol +*: increase or decrease the volume.
- *Vol 60%*: show the volume level, but if you press it it turn the volume off.
- *Play selected*: play the song you have selected.
- *Loop OFF/Loop ON*: allow to play in an infinite loop.
- *Random OFF/Random ON*: allow to play the songs randomly.

## SAVE PAGE

This is where you can save or load your show. The show saved is the last configured in “FINAL MUSIC LIST” and not “PLAY FILES”. When you load a show it replaces all the CONFIG and SONG files, and create automatically the “FINAL MUSIC LIST” but not the “PLAY FILES LIST”.



- *MENU*: move you back to the main page.
- *Save show format*: allows to save the show. Show will be saved with “.shw” extension.
- Load show format: allows to load a show.
- Remove special characters: I had some problems with special characters in the songs name, so I've prepared a special button to replace special characters from a song name with an “\_”. It's possible select a single song or a folder to modify all the song names in one time.



## HOW THE APP WORKS

**ATTENTION: back-up the “overrides.cfg” file (if you use it) before run this app. It automatically writes over it.**

The app uses LightShowPi to run the show but not to manage the list of songs.

The app itself is what manage the list of songs and the config files.

The app calls:

```
SUDO PYTHON ../synchronized_lights.py -file=../song.mp3
```

for each song but before do that it copies over the “overrides.cfg” file the config file connected to the song.

The app calls:

```
SUDO PYTHON ../prepostshow.py “preshow”
```

for each show but before do that it copies over the “overrides.cfg” file the show file itself.

Here two examples of .cfg file with different shows (all the information are in the LightShowPi default.cfg files):

1) show made in the cfg file:

```
# Default is to be on for 10 seconds and off for 1 with no channel_control overrides
# no audio played.
preshow_script =
preshow_configuration =
{
    "transitions": [
        {
            "type": "on",
            "duration": 5,
            "channel_control": {

        },
        {
            "type": "off",
            "duration": 1,
            "channel_control": {

        }
    ],
    "audio_file": null
}
```

2) show made in a .py file:

```
preshow_script = /home/pi/lightshowpi/py/examples/ladder.py
preshow_configuration =
```