Linux for Embedded Systems

Lab 4 - Report

By Noman Noor (id: 302343)

Current Date: 20/05/2022 (dd/mm/yy)

PROBLEM DESCRIPTION

Task: Make the lab RPi a music player with playback interactions possible from the web and the onboard buttons.

Procedure to recreate the design from the attached archive:

In the directory where the contents of the main archive were extracted, run the script build.sh:

```
$ chmod +x ./build.sh
$ ./build.sh
```

Step 1: Setting up the Music Player Buildroot Image:

Basic Configuration:

The base configuration and packages remain the same as the ones for the Utility OS made for Lab 3. However, there are some additional packages we will include and settings we have to change.

Another thing to note is that when I was doing the task I used "***initramfs***" but that's clearly not the best option (as explained in ReadMe.txt). Therefore, while my task uses initramfs, my description of how to do it in this report will be as if I'd used the same ext2/3/4 rootfs I used in the Utility OS of Lab 3.

Filesystem Configuration:

1. Same base config as Utility OS from Lab 3.
2. To avoid having to use modprobe and to have proper device management, under **/dev management** and select "**Dynamic using devtmpfs + eudev**" (selects **eudev** package automatically). *[In "System configuration" menu]*

Do remember the Overlay directory should contain the programs and configurations to be used/replaced.

Additional Packages:

1. Same packages config as Utility OS from Lab 3.
2. In Target packages
    a. In Audio and video applications
        i. alsa-utils (and all sub-packages)

ii. mpd [and all converter packages, decoder packages, alsa support package, curl, alsa support, pulseaudio support, tcpsockets, and others (as defined/enabled in the buildroot config file)]
iii. mpd-mpc

b. Shell and utilities
i. screen

Step 2: MUSIC PLAYER WEB INTERFACE (made with Python and Flask):

Note: The port used for the server is 6699. (may be changed in source code easily)
Based on my fileserver made in Lab 3. Allows interaction with mpd and mpc via the interface on a browser.

Step 3: BUTTONS INTERFACE

My solution uses the Python based app using the libgpiod library to make the buttons on board of the lab RPi to control playback via MPD.

Step 4: INSTALLING

Then we just download the image to the lab RPi as usual. In case we are not using initramfs, we create a partition (as described in the report for Lab 3) and flash the said rootfs image to it (as described in the report for Lab 3 except we may omit creation of the third partition or keep it as we wish).

[Side note: keeping the third partition and having better permissions, user access control, and so on may allow better security in commercial settings.]

: FINISHED :