Level: Beginner V2.0 Draft

Embedded Linux Z2HV2
Assignment 4 - Buildroot
A Reproducible Mp3-Player application

Summary:

This assignment is part of an assignment-series aimed to develop a functional Mp3 player reproducible by buildroot (without any manual modifications after the build) using all the knowledge and tools we got so far in the course and with Extra help from the Open source community of linux.

Main Features of desired MP3 Player:

-Auto detection of mass-storage devices that can be connected via USB (e.g. Flash, USB-HDD, Memory Card).
-Manual Control of Music Player (Play/Pause, Skip, Rewind, Shuffle), either via Push Buttons or Keyboard Commands.
-Ability to play music on different output devices(3.5mm Audio Jack, Bluetooth Audio, HDMI).
-Streaming of the currently played music on the terminal (SSH terminal over Ethernet/Wifi, Serial terminal).
-Audio Notifications about System changes (e.g. Inserting/removing storage/audio devices).

The stages concept has been replaced with Features concept for easier splitting of the task and tracking of work.

Host machine requirements:

-The buildroot project (Tag: 2019.11.1) - Clone a clean repo from http://git.buildroot.net/buildroot/
(you can use the repo from the past labs, but make sure to clean it using the clean commands
and remove the compiler cache from ~/.buildroot-cchace to avoid any build problem)
-Network Connection to your laptop in case of a clean buildroot repo.
-A Raspberry Pi and your own SDcard.
-Your Prefered Hardware connection (Serial, Ethernet), we learned the basic concept for connecting via each one.

System Requirements:

1.1- Generic System Requirements

1.1.1- GSR_1 - The RPI Image shall have one user “root” with password “12345”.

1.1.2- GSR_2 - The RPI Image shall use glibc as the default C-Library.

1.1.3- GSR_3 - The RPI Image shall have a shell prompt “MP3_Shell>”.

1.1.4- GSR_4 - The MP3 Player Shall be packaged firmly with no loose wires, buttons, battery or connectors.

1.1.5- GSR_5 - Any software built, Configuration used, or Scripts written shall be integrated in the Buildroot system and the full image can be regenerated using only the "make" command.

1.1.6- GSR_6 - Any scripts, code, configuration delivered should be documented/commented out.

1.2- Feature-Specific System Requirements

1.2.1- Feat1 "Auto detection of mass-storage devices that can be connected via USB (e.g. Flash, Memory Card)."

1.2.1.1- FSSR_1 - The MP3 Player shall continuously check for new inserted mass storage devices.

1.2.1.2- FSSR_2 - The MP3 Player shall keep track of all .MP3 files available on any storage media.

1.2.1.3- FSSR_3 - The MP3 Player shall be able to play any .MP3 file for the connected storage devices.

1.2.1.4- FSSR_4 - The MP3 Player shall be able to detect and Read from USB Flash Storage Devices.

1.2.1.5- FSSR_5 - The MP3 Player shall be able to detect and Read from Memory Card Devices including the image Disk.

1.2.1.6- FSSR_6 - The MP3 Player shall not modify the content of any connected mass storage device.

1.2.1.7- FSSR_7 - The MP3 Player shall handle adding and removing of media devices while playing or pausing, Playing should not stop unless the media containing the file is removed.

1.2.3- Feat2 "Ability to play music on different output devices(3.5mm Audio Jack, Bluetooth Audio, HDMI)."

1.2.3.1- FSSR_18 - The MP3 Player shall be able to play audio through 3.5mm Audio Jack (wired speakers/headphones).

1.2.3.2- FSSR_19 - The MP3 Player shall be able to play audio through bluetooth (bluetooth speaker/headphones).

1.2.3.3- FSSR_20 - The MP3 Player shall be able to play audio through HDMI (TV Screen).

1.2.3.4- FSSR_21 - The MP3 Player shall continuously monitor the changes in hardware for Audio interfaces.

1.2.3.5- FSSR_22 - The MP3 Player shall continuously search for bluetooth audio devices.

1.2.3.6- FSSR_23 - The MP3 Player shall always try to connect to the nearest known bluetooth audio device.

1.2.3.7- FSSR_24 - The MP3 Player shall be able to pair with new Bluetooth audio devices.

1.2.3.8- FSSR_25 - When Bluetooth Speaker, HDMI TV Screen and 3.5mm wired speakers are connected at the same time, The MP3 Player shall route the audio to the highest priority device in the following order: Highest Priority: 1- bluetooth Speakers/headphones.
2- HDMI.
Lowest Priority: 3- Wired Speakers/headphones.

1.2.3.9- FSSR_26 - The Mp3 Player shall handle connecting and disconnecting audio devices while playing or pausing, Playing shall only stop if no audio device is connected.

1.2.2- Feat3 "Manual Control of Music Player (Play/Pause, Next, Previous, Shuffle), either via Push Buttons and Keyboard Commands."

1.2.2.1- FSSR_8 - The MP3 Player shall control playing the .MP3 files by pressing the "Play/Pause" push button.
1.2.2.3- FSSR_9 - The MP3 Player shall jump to the next song when pressing the "Next" push button.
1.2.2.4- FSSR_10 - The MP3 Player shall restart the current song on pressing the "Previous" push button once.
1.2.2.4- FSSR_11 - The MP3 Player shall play the previous song when pressing the "Previous" push button two successive times within 1 Second.
1.2.2.5- FSSR_12 - The MP3 Player shall jump to a random song when pressing the "Shuffle" push button.
1.2.2.6- FSSR_13 - The MP3 Player shall start playing the .MP3 files when entering "play" on the commandline.
1.2.2.7- FSSR_14 - The MP3 Player shall pause the played .MP3 file when entering "pause" on the commandline.
1.2.2.8- FSSR_15 - The MP3 Player shall skip to the next song when entering "next" on the commandline.
1.2.2.9- FSSR_16 - The MP3 Player shall play the previous song when entering "previous" on the commandline.
1.2.2.10- FSSR_17 - The MP3 Player shall jump to a random song when entering "shuffle" on the commandline.

1.2.4- Feat4 "Streaming of the played music on the terminal (SSH terminal over Ethernet/Wifi, Serial terminal)."

1.2.4.1- FSSR_27 - The Output Image shall have static network IP "192.168.1.6" and netmask "255.255.255.0".
1.2.4.2- FSSR_28 - The MP3 Player shall have the ssh enabled on startup and accessible via "root" user.
1.2.4.3- FSSR_29 - The MP3 Player Shall Greet the user on any new connected serial/ssh terminal with this greeting text "Welcome to BuildRoot MP3 Player" once.
1.2.4.4- FSSR_30 - The MP3 Player shall output on the serial/ssh terminal "MP3 Playing > [File_Name.mp3] Via [audio device name]" when starting playing .MP3 file.
1.2.4.5- FSSR_31 - The MP3 Player shall output on the serial/ssh terminal "MP3 Paused > [File_Name.mp3]" when pausing the .MP3 file.
1.2.4.6- FSSR_32 - The MP3 Player shall output on the serial/ssh terminal "No .MP3 files found" when it can't find any .mp3 file on the system.
1.2.4.7- FSSR_33 - The MP3 Player shall refresh the data outputted on the serial/ssh terminal every 1 second.

1.2.5- Feat5 "Audio and Text Notifications about System changes (e.g. Inserting/removing storage/audio devices)."

1.2.5.1- FSSR_34 - The MP3 Player shall speak out (Text-to-Speech) a notification when an audio device is connected or disconnected.
1.2.5.4- FSSR_35 - The MP3 Player shall speak out (Text-to-Speech) a notification when a storage device is added or removed.
1.2.5.3- FSSR_36 - The MP3 Player shall output the audio notification on the highest priority audio device available.

How to deliver your work:

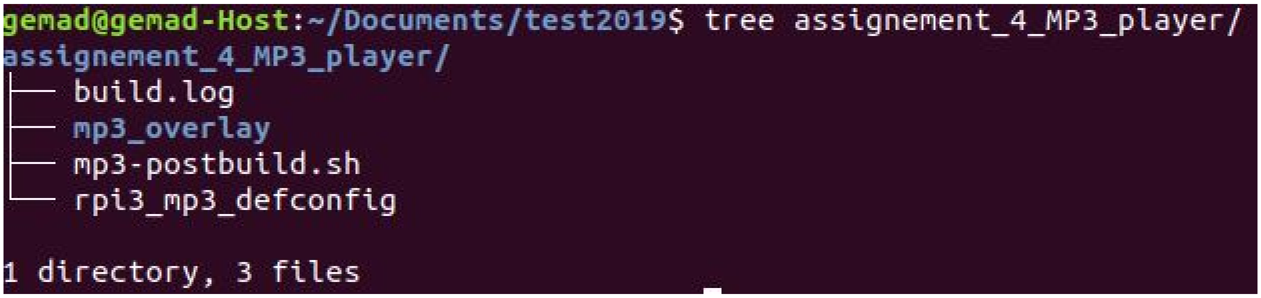
Note: This section might change so keep track and document your work and comment your scripts.

-You should complete the build on your own machine.

-You should test the image on the team’s raspberry pi.

-On lecture I will check who completed the assignment and will ask about the steps and inspect the output build/image.

- Create a Repository on your GitHub account and name it Embedded_Linux_Z2H (already created)
- Create a folder within this repo called assignement_4_MP3_player has the following :



```
gemad@gemad-Host:~/Documents/test2019$ tree assignement_4_MP3_player/
assignement_4_MP3_player/
├── build.log
├── mp3_overlay
├── mp3-postbuild.sh
└── rpi3_mp3_defconfig

1 directory, 3 files
```

* Your “build.log” file created from the build process on your machine (use tee command).
* Your configuration: Save configurations to a file called “rpi3_mp3_defconfig”. (use the savedefconfig command).
* Your filesystem overlay directory.
* Your post-build script.

- Commit and push your changes to the repo (make sure to write a descriptive commit message start with mp3_satage1 , and for stage mp3_stage2 and so on)
- Export your mp3-postbuild.sh and rpi3_mp3_defconfig to .pdf format

Submit A link to your “Repository” and the files “mp3-postbuild.sh”, “mp3-postbuild.pdf” ,”rpi3_mp3_defconfig” and “rpi3_mp3_defconfig.pdf”

to the Assignment on google Classroom