Audio Training Exercises

Session Exercises

Clone the git repo for exercises.

```
$ git clone https://github.com/embitude/training/
```

PS. Just git pull, if repo has already been cloned

Exercise 1: Listing the sound card and devices

1. On the target board, execute the following command:
   ```
   $ modprobe snd-dummy
   ```
   This would load the dummy sound driver, which registers the sound card and corresponding devices
2. List the playback & capture devices
   ```
   $ aplay -l (This should list the playback devices)
   $ arecord -l (This would list the capture devices)
   ```

Steps for compiling the ALSA applications

The ALSA application compilation needs the cross alsa lib for compilation. This doesn't comes as a part of toolchain. So, the sdk generated which has the support for alsa lib is used to compile the applicatoin

1. Get into the Builds directory & untar the sdk
   ```
   $ cd training/Builds
   $ tar -xf arm-buildroot-linux-gnueabihf_sdk-buildroot.tar.gz
   ```
2. Relocate the sdk
   ```
   $ cd  arm-buildroot-linux-gnueabihf_sdk-buildroot
   $ ./relocate-sdk.sh
   ```
3. Next step is to source the sdk environment in shell which is used for application compilation & compile the application
   ```
   $ source training/Builds/arm-buildroot-linux-gnueabihf_sdk-buildroot/environment-setup
   $ cd training/Audio/Apps
   $ make
   ```

Exercise 2: Set Parameters

The objective here is to set the parameters & verify if it has been updated

1. Navigate to Audio/Apps directory
   ```
   $ cd training/Audio/Apps
   ```
2. Compile the application
   ```
   $ make
   ```
   This would generate the executable for all the applications. Let's test the set_params application
3. Transfer the application on the board and make sure that the snd-dummy is loaded
   ```
   $ scp set_params root@192.168.7.2:
   $ modprobe snd-dummy
   ```
4. Execute the application
   ```
   $ ./set_params hw:0,0
   ```

Exercise 3: Minimal Playback application

The objective here is to test minimal playback application on target board

1. Navigate to Audio/Apps directory
   ```
   $ cd training/Audio/Apps
   ```
2. Compile the application
   ```
   $ make
   ```
   This would generate the executable for all the applications. Let's test the playback_min application
3. Transfer the application on the board and make sure that the snd-dummy is loaded
   ```
   $ scp playback_min root@192.168.7.2:
   $ modprobe snd-dummy
   ```
4. Execute the application
   ```
   $ ./playback_min hw:0,0
   ```
   This should playback the audio to the dummy sound card

Exercise 4: Minimal Capture application

The objective here is to test minimal capture application on target board

1. Navigate to Audio/Apps directory
   ```
   $ cd training/Audio/Apps
   ```
2. Compile the application
   ```
   $ make
   ```
   This would generate the executable for all the applications. Let's test the cap_min application
3. Transfer the application on the board and make sure that the snd-dummy is loaded
   ```
   $ scp cap_min root@192.168.7.2:
   $ modprobe snd-dummy
   ```
4. Execute the application
   ```
   $ ./cap_min hw:0,0
   ```
   This should capture the audio from dummy sound card

Exercise 5: Playback Application with various transfer methods

The objective here is to demonstrate the various mechanisms to transfer the data to the driver

1. Navigate to Audio/Apps directory
   ```
   $ cd training/Audio/Apps
   ```
2. Compile the application
   ```
   $ make
   ```
   This would generate the executable for all the applications. Let's test the playback application
3. Transfer the application on the board and make sure that the snd-dummy is loaded
   ```
   $ scp playback root@192.168.7.2:
   $ modprobe snd-dummy
   ```
4. Execute the application
   ```
   $ ./playback <  <audio file>
   ```
   This should playback the audio with selected transfer method