Cross compiling the FMOD Library

Joshua Gastaldello, Kevin Chen, Tyler Rubenuik, Patrick Wan

Simon Fraser University
CMPT 433
Brian Fraser
Fall 2015

Table of Contents

Formatting

1. Commands starting with $ are Linux console commands on the host PC

```
$ echo "Hello world!"
```

2. Commands starting with # are Linux commands on the target (BeagleBone Black)

```
# echo "Hello embedded world!"
```

3. Almost all commands are case sensitive

Intro to the FMOD Library

What is FMOD?

FMOD Studio is an industry leading sound effects engine created by Firelight Technologies. FMOD is capable of mixing and playing numerous types of sound files. They also support all 3 major operating systems and being cross compiled to the ARM processor for embedded systems. FMOD has been used extensively in many popular game engines such as Unity, Unreal Engine 3 and 4, CryEngine and more. (http://www.fmod.org/)

Why did we choose FMOD?

There are many open source libraries that are available on the market, all of which are tailored for a particular situation. We came across two potential options: Audiere and FMOD, of which we selected the latter. Both software offered great options in audio format support and functionality, and both are also well supported in documentation. However, FMOD has support for the ARM processor on the BeagleBone Black. Thus we ultimately decided on FMOD.

How did we use FMOD?

FMOD supported the core functionality of MP3 audio playback. It is able to playback multiple file formats and has adequate documentation that allowed us to manipulate the code as necessary. It has various audio functions implemented such that tasks that were thought to be an issue are much more streamlined than thought; such as pausing a song.

Download and Extract the FMOD Library

1. Go to the FMOD download site (http://www.fmod.org/download) and download the Linux Low Level Programming API (A slimed down version of `fmodstudioapi10702linux.tar.gz` is included with this guide)
2. Change to the directory with the downloaded file.
3. Extract the tar.gz file

```
# tar -xvzf fmodstudioapi10702linux.tar.gz
```

4. Copy the extracted folder over to the NFS directory (see NFS guide for setup)

```
$ cp fmodstudioapi10702linux ~/cmpt433/public/myApps
```

Install the g++ compiler on the target

1. The FMOD library is written in C++ so we need a designed for it. Type in the following command on the BBB to install the g++ compiler (ensure you have internet connectivity on your BBB):

```
# sudo apt-get install g++
```

Get the libstdc++.so.6 file

1. To find the libstdc++.so.6 file you must install apt-file
   ```
   $ sudo apt-get install apt-file
   ```
2. Update apt-file
   ```
   $ sudo apt-file update
   ```
3. Search for libstdc++.so.6
   ```
   $ sudo apt-file find libstdc++.so.6
   ```
4. Search for the libstdc++.so.6 in the correct arm directory. One of the lines should contain the following
   ```
   libstdc++6:  /usr/arm-linux-gnueabi/lib/libstdc++.so.6
   ```
5. Copy the libstdc++.so.6 file into the arm folder in the extracted Fmod folder in public: (All one line!)
   ```
   # cp libstdc++.so.6
   ~/cmpt433/public/myApps/fmodstudioapi10702linux/lowlevel/lib
   /arm
   ```

Copy the library files to the BBB

1. Inside the arm folder, copy over all files into lib on the BBB
   ```
   # cp arm/* /lib
   ```
2. Check all files have been moved onto the BBB

Compiling the Examples

1. Once that is done inside the original FMOD extraction are examples you can compile and run on the BBB. You will need to edit the Makefiles or add parameters to get them to compile:
   ```
   make -f play_stream.makefile CPU=arm CONFIG=Release
   ```
2. After compiling the executable will show up in the make directory:
   ```
   # ./play_stream
   ```

Notes

1. To test songs in the examples you will need to add permissions to it before compiling an example.
2. The Document folder will contain information on the parameters and return values of each function of FMOD. No source code on how the functions operate can be read.

Troubleshooting

Library not found

Check that the library in question exists on the BBB

FMOD Header not found

Double check the path on the examples' Makefile. Examples should contain the same path for all Makefiles:

```
-I../../../lowlevel/inc
```