

嵌入式系統設計 Lab4 Report

314581018 謝鎧駿、314551128 張哲榕

一、 Command we enter

1. 設定 cross-compiler path

```
export CC=/opt/EmbedSky/gcc-linaro-5.3-2016.02-x86_64_arm-linux-gnueabihf/bin/arm-linux-gnueabihf-gcc  
export AR=/opt/EmbedSky/gcc-linaro-5.3-2016.02-x86_64_arm-linux-gnueabihf/bin/arm-linux-gnueabihf-ar  
export RANLIB=/opt/EmbedSky/gcc-linaro-5.3-2016.02-x86_64_arm-linux-gnueabihf/bin/arm-linux-gnueabihf-ranlib
```

2. Compile alsa-lib

i. 進入 alsa-lib 的目錄下

```
cd alsa-lib-1.0.26
```

ii. 用 configure 產生 makefile

```
./configure --host=arm-linux-gnueabihf \  
--prefix=/home/morris/Desktop/alsa_install \  
--enable-shared \  
--disable-static
```

iii. Compile and install

```
make
```

```
make install
```

3. Compile alsa-utils

i. 進入 ALSA-utils 目錄

```
cd alsa-utils-1.0.26
```

ii. 設定 include & lib 路徑

```
export CPPFLAGS="-I/home/morris/Desktop/alsa_install/include"  
export LDFLAGS="-L/home/morris/Desktop/alsa_install/lib"
```

iii. 用 configure 產生 makefile

```
./configure --host=arm-linux-gnueabihf \  
--without-curses \  
--disable-nls
```

--disable-xmlto

iv. **Compile**

make

4. Compile zlib

i. **進入 zlib 目錄**

cd zlib-1.2.3

ii. **用 configure 產生 makefile**

./configure --prefix=/home/morris/Desktop/madplay/source

iii. **Compile and install**

make

make install

5. Compile libid3tag

i. **進入 libid3tag-0.15.1 目錄**

cd libid3tag-0.15.1b

ii. **用 configure 產生 makefile**

./configure --host=arm-linux-gnueabihf

--disable-debugging

--prefix=/home/morris/Desktop /madplay/source

CPPFLAGS=-I/home/morris/Desktop/madplay/source/include

LDFLAGS=-L/home/morris/Desktop/madplay/source/lib

iii. **Compile and install**

make

make install

6. Compile libmad

i. **進入 libmad-0.15.1 目錄**

cd libmad-0.15.1b

ii. **用 configure 產生 makefile**

./configure --host=arm-linux-gnueabihf

--disable-debugging

--prefix=/home/morris/Desktop/madplay/source

CPPFLAGS=-I/home/morris/Desktop/madplay/source/include

LDFLAGS=-L/home/morris/Desktop/madplay/source/lib

iii. **Compile and install**

```
make  
make install
```

7. Compile madplay

i. 進入 madplay 目錄

```
cd madplay-0.15.2b
```

ii. 用 configure 產生 makefile

```
./configure --host=arm-linux-gnueabihf  
--disable-debugging  
--with-alsa  
CPPFLAGS=-I/home/morris/Desktop/madplay/source/include  
LDFLAGS=-L/home/morris/Desktop/madplay/source/lib
```

iii. Compile and install

```
make
```

```
make install
```

8. 將所有.so 以及 madplay 執行檔 copy 到板子上

二、 Problems we encountered

1. zlib compile 出來是 static 的

需重新 compile 一次，並指定--shared 即可。

2. compile libmad 時，出現 error : cc1: error: unrecognized command line option “-fforce-mem”

需要將 fforce-mem 拿掉，用： sed -i '/-fforce-mem/d' configure

3. compile libmad 時，出現 error : selected processor does not support

Thumb mode `rsc r0,r0,#0'

將 fixed.h 檔案中的

```
# define MAD_F_MLN(hi, lo) \
asm ("rsbs %0, %2, #0\n\t" \
     "rsc %1, %3, #0" \
     : "=r" (lo), "=r" (hi) \
     : "0" (lo), "1" (hi) \
     : "cc")
```

改成：

```
#ifdef __thumb__
/* In Thumb-2, the RSB-immediate instruction is only allowed with a zero
```

operand. If needed this code can also support Thumb-1 (simply append "s" to the end of the second two instructions). */

```
# define MAD_F_MLN(hi, lo) \
asm ("rsbs %0, %0, #0\n\t" \
"sbc %1, %1, %1\n\t" \
"sub %1, %1, %2" \
:+ "&r" (lo), "=r" (hi) \
:"r" (hi) \
:"cc") \
#else /* ! __thumb__ */ \
# define MAD_F_MLN(hi, lo)
\ asm ("rsbs %0, %2, #0\n\t" \
"rsc %1, %3, #0" \
:"=r" (lo), "=r" (hi) \
:"=&r" (lo), "=r" (hi) \
:"0" (lo), "1" (hi) \
:"cc") \
#endif /* __thumb__ */
```

4. aplay 與 arecord 其實是同一個執行檔，只是 aplay 執行的時候會根據 argv[0]來去判斷模式是甚麼，因此需要做 ln -s ./aplay ./arecord，這樣執行 arecord 時就會執行 aplay 的程式碼且 argv[0]="arecord"，所以他就會執行錄音的部分。

三、 Signal processing procedure for an MP3 file

1. 解析 MP3 Frame：讀取每個 MP3 frame 的標頭與編碼資料。
2. Huffman 解碼：將 Huffman 壓縮的頻域係數解碼回量化資料。
3. 反量化 (Dequantization)：恢復量化前的頻域振幅。
4. IMDCT 反轉換：將頻域資料透過 IMDCT 轉回時域資料。(DCT 我在視訊壓縮課程學過，有興趣可以選修)
5. Overlap-Add：對時域區塊進行重疊相加，使音訊連續平滑。
6. 輸出 PCM：最終產生 PCM 格式音訊（例如 16-bit / 44.1kHz），可交由 ALSA 播放。

四、 Reference

https://blog.csdn.net/qq_31811537/article/details/104842097