

國立交通大學

資訊學院 資訊學程

碩士論文

多媒體播放系統之設計與實作

Multi-media Playback System Design and Implementation



研究生：劉超仁

指導教授：蔡文錦 教授

中華民國一百零一年六月

多媒體播放系統之設計與實作

Multi-media Playback System Design and Implementation

研究生：劉超仁

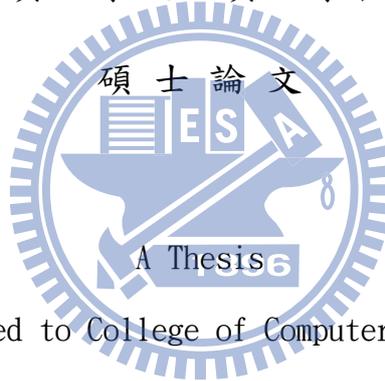
Student : Chao-Jen Liu

指導教授：蔡文錦

Advisor : W. J. Tsai

國立交通大學

資訊學院 資訊學程



Submitted to College of Computer Science

National Chiao Tung University

in partial Fulfillment of the Requirements

for the Degree of

Master of Science

in

Computer Science

June 2012

Hsinchu, Taiwan, Republic of China

中華民國一百零一年六月

多媒體播放系統之設計與實作

學生：劉超仁

指導教授：蔡文錦

國立交通大學 資訊學院 資訊學程 碩士班

摘要

隨著科技進步和網際網路的成熟，智慧型手機、多媒體播放器、相機或其他個人的多媒體設備越來越多，使得多媒體檔案(音訊、視訊)為了使用者不同的需求，不斷地產生新的多媒體格式、編碼方式或壓縮方式。此時原有的播放器就無法播放新的多媒體檔案，因此學生希望能夠透過軟體更新的方式，重新設計多媒體播放器，提供更容易使用與便宜的解決方案。

關鍵字：多媒體、嵌入式系統

Multi-media Playback System

Design and Implementation

Student : Chao-Jen Liu

Advisors : Dr. W.J. Tsai

Degree Program of Computer Science

Nation Chiao Tung University

Abstract

As the mature technological progress, provide internet sharing is popular and easy. At the same time, there are many smart phones, multi-media players, cameras or other personal handheld devices that bring more and more multimedia files or types (audio and video) .These complicated applications cause to design different encoding or compression in order to satisfy every kind of demand.. But, the playback device that user already had maybe can't play. Therefore, student wants to design multi-media playback system through software update that can provide the easier and cheaper solution

Keywords: multi-media, embedded system

誌謝

此篇論文的完成，首先感謝指導教授 蔡文錦 博士，謝謝教授在學生論文指導上所花費的時間和精力。

感謝口試委員 林建志 博士和 蕭旭峰 教授，在口試給我的指導與建議，讓我的論文內容可以更加完善。

謝謝我的家人父親與大哥，在這一路上攜手一起度過這家裡最痛苦的考驗。

謝謝我的老婆 巧婷，陪伴我走過人生中的最低潮的時間，給我鼓勵和叮嚀，並且幫我打點一切讓我無後顧之憂。

謝謝關心我的朋友，有你們的鼓勵和打氣，給了我無比的力量。

最後，我要把這篇論文獻給在天國的母親。媽，我好想您，我好愛您。

劉超仁 謹致於交通大學
2012 年 7 月



目 錄

| | |
|--------------------------------------|--------|
| 中文摘要..... | i |
| 英文摘要..... | ii |
| 誌謝..... | iii |
| 表目錄..... | vi |
| 圖目錄..... | vii |
| 一、 序論..... | - 1 - |
| 1.1. 研究動機..... | - 1 - |
| 1.2. 多媒體播放系統發展現況..... | - 1 - |
| 1.3. 系統目標與主要工作..... | - 2 - |
| 二、 系統規格和相關技術研究..... | - 3 - |
| 2.1. 嵌入式系統定義..... | - 3 - |
| 2.2. 多媒體播放..... | - 3 - |
| 2.3. 多媒體播放程式的比較..... | - 4 - |
| 2.4. 播放平台的比較..... | - 5 - |
| 2.5. 硬體開發平台介紹..... | - 6 - |
| 2.5.1. CPU..... | - 6 - |
| 2.5.2. 記憶體..... | - 6 - |
| 2.5.3. 音訊處理器 (Audio Processor) | - 7 - |
| 2.5.4. 視訊處理器 (Video Processor) | - 7 - |
| 2.5.5. USB 介面..... | - 7 - |
| 三、 系統的設計和實作..... | - 8 - |
| 3.1. 系統開發環境建立..... | - 8 - |
| 3.2. 開發編譯環境建立..... | - 9 - |
| 3.3. 系統架構..... | - 10 - |
| 3.3.1. 播放器主架構..... | - 10 - |
| 3.3.2. 多媒體播放流程圖 (音訊部分) | - 11 - |
| 3.4. FFmpeg 播放程式的移植..... | - 12 - |
| 3.4.1. FFmpeg 的簡介..... | - 12 - |
| 3.4.2. FFmpeg 的組成要素..... | - 12 - |
| 3.4.3. FFmpeg 檔案來源..... | - 12 - |
| 3.4.4. 如何設定 FFmpeg..... | - 13 - |
| 3.4.5. FFmpeg 的移植..... | - 13 - |
| 3.4.6. 設定目標平台支援的格式或類型..... | - 15 - |
| 3.4.7. 編譯和安裝..... | - 15 - |
| 3.5. 應用程式和驅動程式的編譯與建立..... | - 16 - |
| 3.5.1. 驅動程式實做相關..... | - 16 - |

| | | |
|--------|---------------|--------|
| 3.6. | 系統最佳化處理..... | - 18 - |
| 3.6.1. | 多媒體輸入部分..... | - 18 - |
| 3.6.2. | 音訊解碼器最佳化..... | - 18 - |
| 3.7. | 測試..... | - 26 - |
| 3.8. | 系統實作成果..... | - 27 - |
| 四、 | 總結和未來展望..... | - 29 - |
| | 參考文獻..... | - 30 - |
| | 簡歷..... | - 31 - |



表目錄

| | |
|-------------------------------|--------|
| 表 1 – Sigma 晶片組 | - 2 - |
| 表 2 – Realtek 晶片組 | - 2 - |
| 表 3 – Sigma V.S. Realtek..... | - 2 - |
| 表 4 – 播放程式比較..... | - 4 - |
| 表 5 – 播放平台比較..... | - 5 - |
| 表 6 – 支援格式類型..... | - 15 - |
| 表 7 – 音訊檔案清單..... | - 19 - |
| 表 8 – 最佳化結果（解碼時間） | - 23 - |
| 表 9 – 最佳化結果（音訊品質） | - 24 - |
| 表 10 – 客觀差評分表..... | - 25 - |



圖目錄

| | |
|---------------------------|--------|
| 圖 1 – 硬體開發平台..... | - 6 - |
| 圖 2 – 播放器架構..... | - 10 - |
| 圖 3 – 多媒體播放流程圖（音訊部分）..... | - 11 - |
| 圖 4 – 操作畫面 1..... | - 27 - |
| 圖 5 – 操作畫面 2..... | - 27 - |
| 圖 6 – 操作畫面 3..... | - 28 - |



一、序論

1.1. 研究動機

隨著科技的進步，網際網路的成熟，智慧型手機、多媒體播放器、相機或其他個人的多媒體設備越來越多，使得多媒體檔案(音訊、視訊)，成為替代傳統媒體的最新媒介。因此學生希望能夠透過嵌入式系統的設計，提供更容易使用與便宜的多媒體解決方案。

1.2. 多媒體播放系統發展現況

從2008年開始，Sigma首先提出863x系統單晶片的解決方案。但是直到2009年開始，高解析度的多媒體播放器才逐漸受到市場所歡迎，隨著近幾年的發展，目前主要分為兩個製造商：Sigma和Realtek。

而在使用者的評價方面，Sigma視訊的播放效果比較好，缺乏Real編碼格式的支援(RM、RMVB)，價格較高。另一方面，Realtek的解決方案比較便宜，並且支援Real的編碼格式。

底下的表格則提供兩家廠商的發展歷史、晶片規格和優缺點比較表。

表 1 – Sigma 晶片組

| 製造商 | 晶片組 | 發表時間 | 處理器頻率 (MHz) | 記憶體 (MB) |
|-------|------|-----------|----------------|-------------|
| Sigma | 863X | 2008 | 300 | 512 |
| Sigma | 864X | 2009 | 667~800 | 512~1024 |
| Sigma | 865X | 2009~2011 | 500 | 512~1024 |
| Sigma | 8670 | 2011 | 700 | |
| Sigma | 8910 | 2012 | 1200 | |

表 2 – Realtek 晶片組

| 製造商 | 晶片組 | 發表時間 | 處理器頻率 (MHz) |
|---------|--------|-----------|----------------|
| Realtek | '3' 系列 | 2009~2010 | 400 |
| Realtek | '5' 系列 | 2010~2011 | 500 |
| Realtek | '6' 系列 | 2011 | 750 |

表 3 – Sigma V.S. Realtek

| 製造商 | 價格(NT) | 優點 | 缺點 |
|---------|---------------|-------------------|-------------|
| Sigma | \$4500~\$9000 | 音質與畫質較好 | 不支援 Real 格式 |
| Realtek | \$2400~\$4800 | 1 價格便宜、支援 Real 格式 | 散熱不佳、畫質較差 |

1.3. 系統目標與主要工作

多媒體在不同環境和條件的限制下，應用不斷的增加，使得越來越多的壓縮演算法和多媒體格式被提出。因此希望藉由在個人電腦上播放器的研究，在軟體上，設計一種可擴充性的架構，以因應未來的需求。

另外，在此嵌入式系統中，採用 MIPS 公司的處理器晶片，包含了 USB 輸入和輸出介面，並掛載 Linux 作業系統。透過作業系統的支援下，實作各個驅動程式的撰寫。最後，可透過 LCD 的顯示和喇叭來播放多媒體檔案。

二、 系統規格和相關技術研究

2.1. 嵌入式系統定義

嵌入式系統是透過軟體和硬體的緊密配合，以實際應用為設計中心，追求可靠性高、成本低廉和功耗較低的高整合度系統。也可說是為特定應用而設計的專用計算機系統。

2.2. 多媒體播放

多媒體檔案是一種媒體的組合，藉由此媒體來達到互動式的信息傳遞或傳播。使用的媒體包括文字、圖片、聲音、動畫和影片，以及程式所提供的互動功能。

多媒體檔案一般都是經過高度壓縮儲存的資料，因此在播放時需要即時解碼來呈現多媒體內容。多媒體解碼可分為軟體解碼或硬體解碼。硬體解碼的部份，執行效率高、速度快，但卻有成本較高而且較不容易擴充的缺點。軟體解碼的部份，需要耗費大量 CPU 的時間，但卻因為硬體成本較低，而且擴充和修改容易，成為嵌入式系統的好選擇。



2.3. 多媒體播放程式的比較

大部分的使用者都使用作業系統提供的預設播放器，常常遇到檔案無法順利播放或是檔案格式不支援等情況。而此種狀況，若非擁有資訊背景或是深入研究問題，會有不知道如何解決的窘境。此外，擁有程式能力的使用者，卻又會因為軟體授權問題，而無法提供進一步的建議或改善。

針對此問題，學生找到一個自由軟體 FFMpeg，提供多媒體播放的解決方案。並將 FFMpeg 與現有作業系統的預設播放器，提供一個比較表，並在之後的章節說明 FFMpeg 的優勢。

表 4 – 播放程式比較

| | Windows Media Player | iTunes | FFMpeg |
|-------|---------------------------|---------------------------|-------------------------------|
| 所有者 | Microsoft | Apple | FFmpeg project |
| 軟體授權 | Proprietary | Proprietary | LGPL |
| 支援格式 | 多(部分不支援) | 多(部分不支援) | 幾乎都支援 |
| 開放原始碼 | 否 | 否 | 是 |
| 作業系統 | Windows | Windows/Mac OS | Windows/ Mac OS/Linux |
| 使用語言 | C++ (COM) | 未知 | C |
| 硬體需求 | CPU > 1 GHz RAM > 1 GB | CPU > 1 GHz RAM > 1 GB | CPU < 500 MHz RAM < 256 MB |

2.4. 播放平台的比較

在播放平台方面，一般有兩種選擇：個人電腦與嵌入式系統。下表則列出兩種平台的特性作為研究參考。

表 5－ 播放平台比較

| | 個人電腦 | 嵌入式系統 |
|----------|----------------|------------------------------------|
| 價格 | 高 | 低 |
| 設定 | 複雜 | 簡單 |
| 操作方式 | 鍵盤與滑鼠 | 遙控器 |
| CPU 運算速度 | 高 | 低 |
| 使用者介面 | 複雜 | 容易 |
| 記憶體容量 | > 2 GB (高) | < 512 MB (低) |
| 硬體加速 | 需軟體硬體配合 | 有 |
| 輸出入介面 | HDMI、D-Sub、DVI | HDMI、D-Sub、 Component、Composite |

2.5. 硬體開發平台介紹

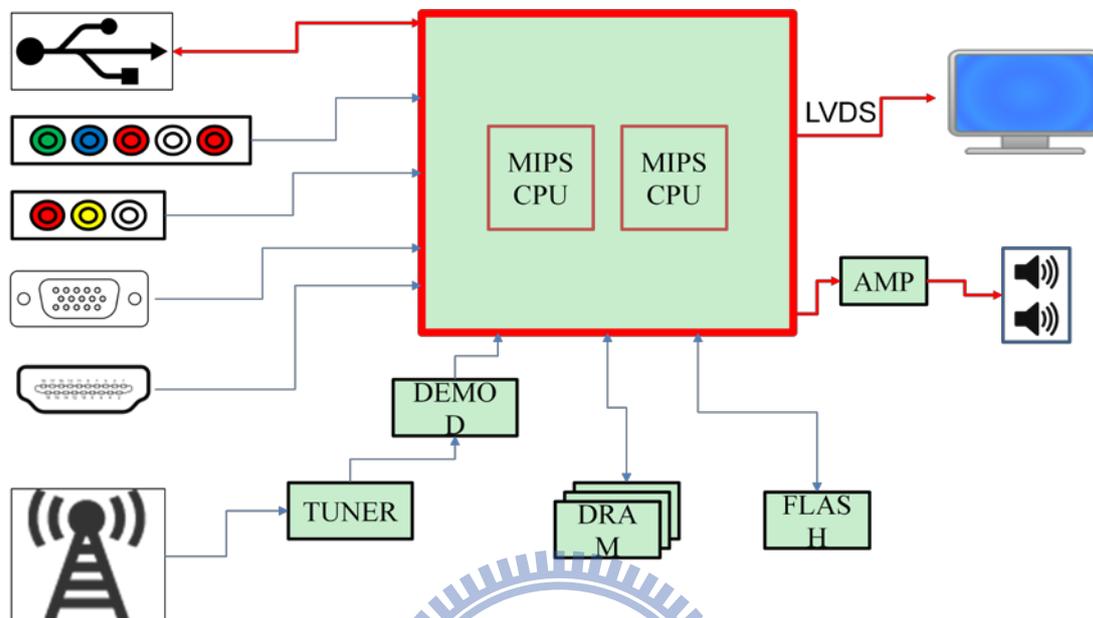


圖 1- 硬體開發平台

2.5.1. CPU

本系統的中央處理器為兩顆 MIPS 32 CPU，頻率皆為 266 MHz。第一個處理器為 CPU0（之後簡稱 CPU0），擁有 4 KB 的指令快取(instruction cache)和 4 KB 的資料快取(data cache)，負責執行 Linux 作業系統，管理系統的輸入和輸出。另一個處理器為 CPU1（之後簡稱 CPU1），擁有 8 KB 的指令快取和 16 KB 的資料快取，負責音訊的解碼和輸出。

2.5.2. 記憶體

本系統採用 SDRAM 作為系統主記憶體，大小為 128 MB，負責資料的暫存空間、執行程式所需的空間等工作。NOR Flash 則是用來儲存軟體（解碼器）、應用程式、驅動程式和作業系統等工作，大小為 4MB。

2.5.3. 音訊處理器 (Audio Processor)

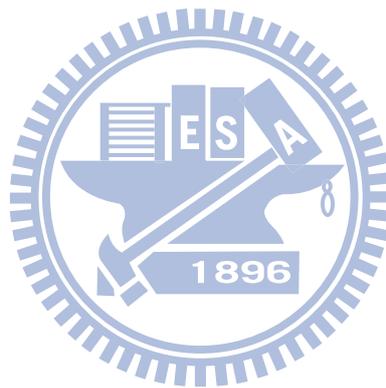
本系統提供 24 位元的類比和數位轉換 (ADC and DAC)，立體聲音輸出以及支援音訊頻率從 22.05 KHz ~ 48 KHz。

2.5.4. 視訊處理器 (Video Processor)

支援 NTSC/PAL/SECAM 系統，提供 S-Video、AV 和 CVBS 輸入。

2.5.5. USB 介面

提供兩組 USB2.0/1.1。



三、 系統的設計和實作

3.1. 系統開發環境建立

主要是要在 Linux 的作業系統底下，並且取得 MIPS 提供的作業系統原始碼，以便可以針對此版本的作業系統設計、安裝和編譯驅動程式和應用程式。

3.1.1. 安裝 ubuntu

可選擇 11.04 或 12.04 的長期支援版本安裝，可由 www.ubuntu.com/download 取得安裝檔案。安裝的方式可採取一般的單純 Linux 安裝。

3.1.2. 取得 MIPS Linux Kernels。

目前 MIPS 提供的穩定版本有 2.6.35.9 和 26.36.4。

1. Linux 2.6.36.4

www.linux-mips.org/pub/linux/mips/mti-stable/v2.6/linux-mti-2.6.35.9-2.tar.gz

2. Linux 2.6.35.9

www.linux-mips.org/pub/linux/mips/mti-stable/v2.6/linux-mti-2.6.35.9-2.tar.gz

3.2. 開發編譯環境建立

安裝 Toolchain，MIPS 提供兩種 toolchain：mipsel-linux-gnu 或 mips-sde-elf。

3.2.1. mipsel-linux-gnu 安裝

1. 使用預先編譯好的 Toolchain

(1) 取得 mips_linux_toolchain_bin-1.1.tar.bz2

(2) 解壓縮

```
cd /opt
```

```
tar vfxj mips_linux_toolchain_bin-1.1.tar.bz2
```

(3) 增加執行目錄於環境變數

```
export PATH=/opt/mips_linux_toolchain/bin:$PATH
```

2. 使用自行編譯的 Toolchain

(1) 取得 mips_linux_toolchain_src-1.1.tar.bz2

(2) 解壓縮

```
tar vfxj mips_linux_toolchain_src-1.1.tar.bz2
```

(3) 編譯 Toolchain 並且安裝於指定目錄

```
./build-mips-sde-elf.sh -prefix=DIR
```

3.2.2. mips-sde-elf 安裝

1. 使用預先編譯好的 Toolchain

(1) 取得 mips_linux_toolchain-4.4.6-bin.tar.bz2

(2) 解壓縮

```
cd /opt
```

```
tar vfxj mips_linux_toolchain-4.4.6-bin.tar.bz2
```

(3) 增加執行目錄於環境變數

```
export PATH=/opt/mips_linux_toolchain/bin:$PATH
```

2. 使用自行編譯的 Toolchain

(1) 取得 mips_linux_toolchain-4.4.6.tar.bz2

(2) 解壓縮

```
tar vfxj mips_linux_toolchain-4.4.6.tar.bz2
```

(3) 編譯 Toolchain 並且安裝於指定目錄

```
./build-mips-sde-elf.sh -prefix=DIR
```

3.3. 系統架構

3.3.1. 播放器主架構

將播放器分成三個部分來設計；分別是多媒體檔案的解多工器 (Demuxer)，編碼資料的解析器 (Parser) 和視訊或音訊的解碼器 (Decoder)。

透過解多工器 (Demuxer) 函式庫將多媒體檔案先分離出視訊和音訊的編碼資料，再透過解析器 (Parser) 的函式庫，將不同的編碼資料檔案，合併或分解成解碼器 (Decoder) 的最小處理單位。最後，透過解碼器 (Decoder) 將資料解碼透過 LCD 或喇叭將多媒體呈現。

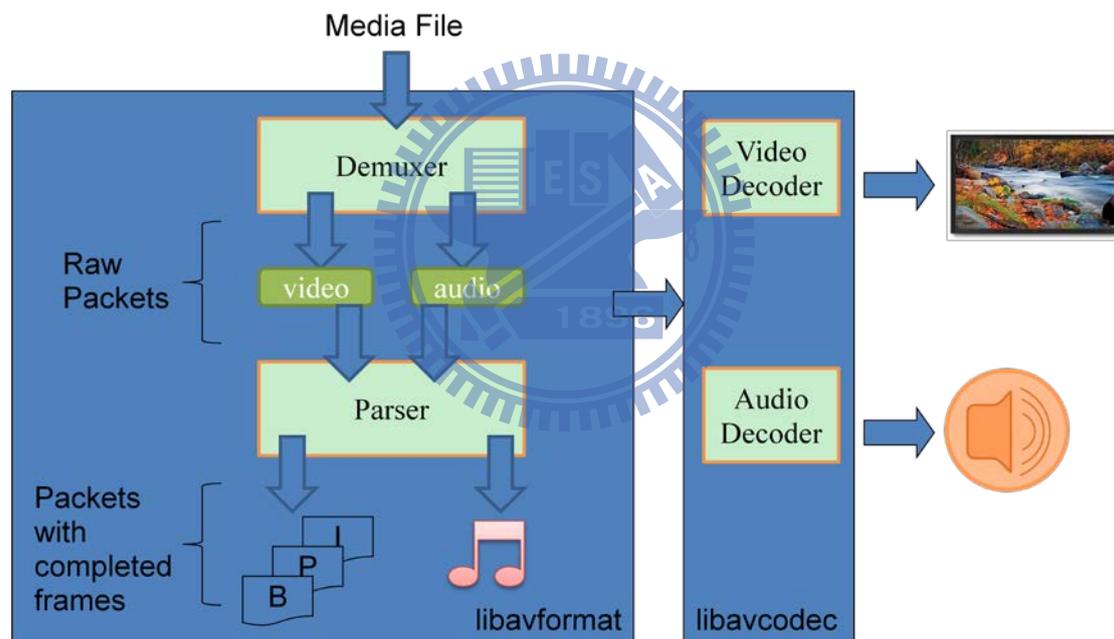


圖 2 - 播放器架構

3.3.2. 多媒體播放流程圖（音訊部分）

本系統採用 C 語言和組合語言來完成。包括四大區塊，分別是使用者介面、應用程式、驅動程式和解碼器。

使用者介面

提供使用者控制多媒體播放的功能如下：播放、暫停、停止、快轉和倒退。

應用程式

提供多媒體檔案的開啟與讀取，接收使用者介面觸發的命令。

驅動程式

提供緩衝區的管理便於解碼器使用，驅動相對應的硬體，將多媒體播放透過喇叭播放。

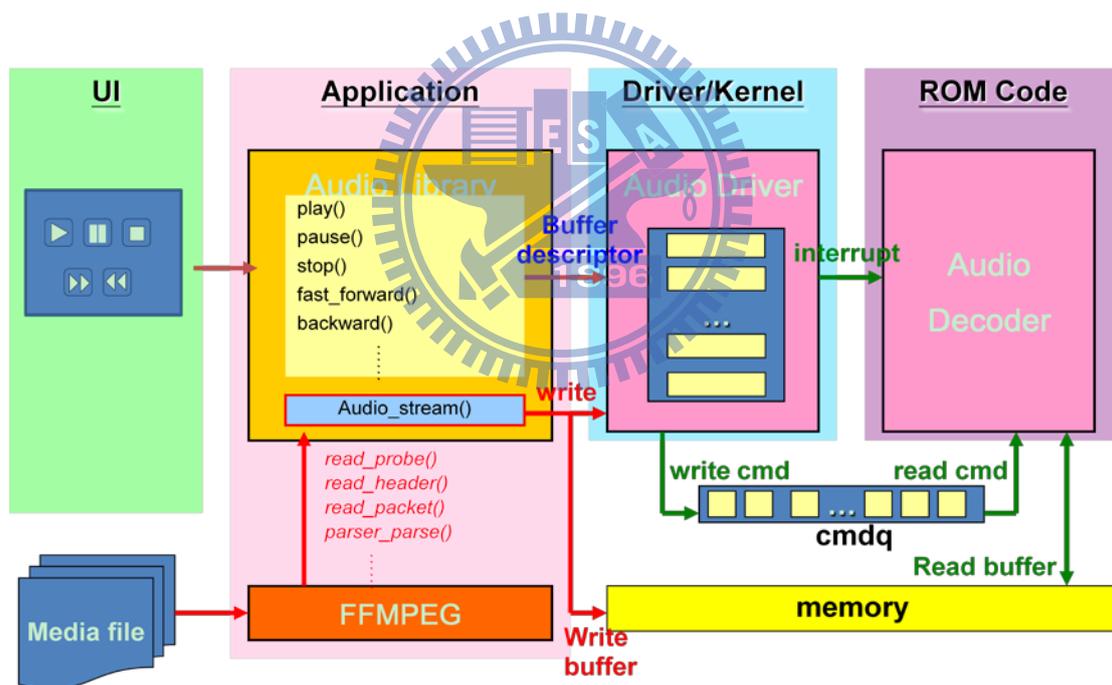


圖 3- 多媒體播放流程圖（音訊部分）

3.4. FFmpeg 播放程式的移植

3.4.1. FFmpeg 的簡介

FFmpeg 是透過 C 語言開發並且開放原始碼的自由軟體，提供處理多媒體資料的函式庫和相關程式。具有跨平台和支援多種作業系統的優勢，目前開發小組持續發展和維護中，採用的軟體授權和許可協議依據不同的應用分別為 LGPL 或 GPL。

3.4.2. FFmpeg 的組成要素

libavcodec - 音訊/視訊的編碼和解碼函式庫(encoder/decoder)。

libavformat - 音訊/視訊的混和器和分路器函式庫(muxer/demuxer)。

FFplay - 基於 FFmpeg 函式庫的多媒體播放器。

3.4.3. FFmpeg 檔案來源

官方網站 <http://ffmpeg.org>

下載位址 <http://ffmpeg.org/download.html>

目前最新的版本 FFmpeg 0.10.2 “Freedom”

<http://ffmpeg.org/releases/ffmpeg-0.10.2.tar.bz2>

3.4.4. 如何設定 FFmpeg

1. 下載 FFmpeg 的原始碼
\$ wget <http://ffmpeg.org/releases/ffmpeg-0.10.2.tar.bz2>
2. 解壓縮
\$ tar xjf ffmpeg-0.10.2.tar.bz2
3. 所有 FFmpeg 可配置的選項
\$./configure -help
4. 設定支援的解析器(parser)
\$./configure -list-parsers
5. 設定支援的分路器(demuxer)
\$./configure -list-demuxers
6. 設定安裝路徑
\$./configure -prefix=/ffmpeg

3.4.5. FFmpeg 的移植

1. 移除不需要的功能或套件
--disable-altivec --disable-amd3dnow --disable-amd3dnowext
--disable-mmx --disable-mmx2 --disable-sse --disable-ssse3
--disable-armv5te --disable-armv6 --disable-armv6t2
--disable-armvfp --disable-iwmmxt --disable-mmi
--disable-neon --disable-vis --disable-yasm
2. 設定必要的功能和套件
--enable-protocol=file -enable-shared -enable-pthreads -extra-ldflags=-static

3. 相關應用程式介面說明

(1) 開啟多媒體檔案

```
int av_open_input_file(  
    AVFormatContext **ic_ptr,  
    const char *filename,      // Input  
    AVInputFormat *fmt,       // Input  
    int buf_size,             // Input  
    AVFormatParameters *ap    // Input  
);
```

(2) 讀取多媒體檔案

```
AVPacket packet;  
while( av_read_frame(pFormatCtx, &packet) >=0) {  
    // Is this a packet from the audio stream?  
    if(packet.stream_index==audioStream) {  
        printf("Packet Data 0x%08x Size 0x%x\n",*(unsigned  
            int*)packet.data,packet.size);  
    }  
    // Free the packet  
    av_free_packet(&packet);  
}  
av_close_input_file(pFormatCtx);
```

(3) 註冊在編譯時期選擇的解析器(parser)和分路器(demuxer)

```
void av_register_all(void);
```

(4) 取得相對應的資料來設定解碼器(decoder)

```
int av_find_stream_info(AVFormatContext *ic);
```

3.4.6. 設定目標平台支援的格式或類型

表 6- 支援格式類型

| 檔案格式 | 視訊 | 音訊 |
|--------------|--------------|---------------|
| AVI | RV 8/9/10 | COOK |
| MPEG/MPG/VOB | MPEG I/II/IV | WMA STANDARD |
| MKV | H264/AVC | MPEG I/II/III |
| ASF/WMV | XVID | PCM/ADPCM |
| MP4 | VC1 | AC3 |
| TS | AVS | AAC/HEAAC |
| RM/RMVB | MPEG4 | |

3.4.7. 編譯和安裝

ffmpeg 設定完成後，執行 `make` 和 `make install` 來編譯程式碼，在 `--prefix` 設定的路徑中，取得相對的函式庫(library)和引入檔案即可。



3.5. 應用程式和驅動程式的編譯與建立

3.5.1. 驅動程式實做相關

1. mmap

(1) 原型 (Protoype)

系統呼叫：`mmap(caddr_t addr, size_t len, int prot, int flags, int fd, off_t offset)`

作業方法：`int (*mmap)(struct file *, struct vm_area_struct *)`;

(2) 主要作用

將裝置上的記憶體映射到 user-space 位址空間，意味著 user-space 可從自己的虛擬位址空間來直接存取硬體裝置上的記憶體。

(3) 實做方法

建立分頁表給上述的虛擬位址範圍，有兩種方式。第一，可透過 `remap_pfn_range()` 一次搞定。第二，則是透過 `nopage` 的作業方法，在存取時才一次處理一頁。

2. ioctl

(1) 原型 (Prototype)

系統呼叫：`int ioctl(int fd, unsigned long cmd, ...)`;
作業方法：`int (*ioctl)(struct inode *, struct file *, unsigned int, unsigned long)`;

(2) 主要作用

執行裝置專屬指令 (device-specific command)。應用程式透過 `ioctl()` 系統呼叫來傳達指令；核心本身預先定義了少數個不必參考 `fops` 表的 `ioctl` 通用指令。若驅動程式沒提供 `ioctl` 作業方法，當應用程式要求執行核心不認識的 `ioctl` 指令，則應用程式會收到錯誤。

(3) 實做方法

先定義各 `cmd` 與各指令之間的對應關係。

`_IO(type, number)`
`_IOR(type, number, datatype)`
`_IOW(type, number, datatype)`
`_IORW(type, number, datatype)`

其中 `type` 為核心中獨一無二編號，方便核心辨認並且呼叫到對應的驅動程式；`number` 為流水號，方便驅動程式分配不同的控制指令；而 `datatype` 則是代表此控制指令的資料量大小。

3.6. 系統最佳化處理

3.6.1. 多媒體輸入部分

利用系統提供的 USB 介面，讀取多媒體檔案，再分別透過不同的應用程式與驅動程式，觸發相對應的操作。但在此系統中，因為太多對於記憶體存取，造成軟硬體都在等待資料，所以這樣的情況下，系統採用預先載入多媒體資料的作法，透過可定義的一塊記憶體大小空間，預先讀取 USB 中的多媒體資料，來解決此問題。

3.6.2. 音訊解碼器最佳化

在此嵌入式系統下，雖然已將音訊解碼器 (Decoder) 獨立於另一個中央處理器上，以避免資源被其他程序所佔用，但還是受限於較低的處理器頻率和共用的記憶體頻寬。因此必須針對解碼器的部份做最佳化處理，以避免處理器的效能不同所造成的影音不同步問題。

在此最佳化中，分為三個部分：解碼器函式分析、最佳化方式、最佳化結果分析。

3.6.2.1. 透過 gcc profile 工具做程式分析

以下方的表格為例，透過 profile 工具，可以得知在此程式中，每個函式被呼叫的次數、需要執行的時間和所占用的比例為何。

表 7- 音訊檔案清單

| sample % time | counts cumulative seconds | self seconds | self calls | name |
|------------------|---------------------------------|-----------------|---------------|----------------|
| 54.48 | 26.67 | 26.67 | 537141716 | DLB_sshl |
| 10.93 | 32.02 | 5.35 | 663242456 | DLB_mpy32x32 |
| 6.68 | 35.29 | 3.27 | 215020608 | DLB_ssub |
| 6.37 | 38.41 | 3.12 | 295317438 | DLB_sadd |
| 4.94 | 40.83 | 2.42 | 318178859 | DLB_sshr |
| 3.82 | 42.7 | 1.87 | 134388 | dcv_cifft |
| 2.27 | 43.81 | 1.11 | 134388 | dmxd_process |
| 1.31 | 44.45 | 0.64 | 134388 | idctsc |
| 1.23 | 45.05 | 0.6 | 367231 | bta_calcpsdmsk |
| 1.04 | 45.56 | 0.51 | 134388 | woad_decode |
| ... | ... | ... | ... | ... |

3.6.2.2. 最佳化的方式

透過 profile 的工具後，針對不同的函式的特性可採用下列方式做最佳化：

1. 行內函式 inline function

如果函式內容比較少，而且呼叫的次數相當頻繁，造成環境切換 (context switch) 的情形多，則此函式就相當適合用行內函式來取代。

2. 組合語言加速

透過 gcc profile 的工具觀察後，有部分函式執行時間很長，被呼叫的次數多，而且函式內容也相當多。

這裡指的組合語言加速，是透過此 MIPS 平台中，專屬的特定指令集，例如：CLO/CLZ、ROTR/WSBH、MADD/MADDU、MSUB/MSUBU...等。

特定指令原文說明：

(1) *clo d, s*

Count leading (high-order) one bits in *s* considered as a 32-bit word.

(2) *clz d, s*

Count leading (high-order) zero bits in *s* considered as a 32-bit word.

(3) *wsbh*

32 bit byte-swap within each of the two half-words. This is a 32-bit instruction. *wsbh* works well together with bit-rotates to perform many forms of byte reorganization in a small number of instructions

(4) *rotr d, s, t*

Rotate right by constant. Will be synthesized for instruction sets earlier than MIPS32R2.

$$d = ((\text{unsigned})s \gg t) | (s \ll (32-t));$$

(5) *madd d, s, t*

maddu d, s, t

Integer multiply-accumulate with integral write of result to a general register (signed add and unsigned versions.)

$$\begin{aligned} hi-lo += (\text{long long}) s * (\text{long long}) t; \\ d = lo; \end{aligned}$$

(6) *msub d, s, t*
msubu d, s, t

The negative version of integer multiply/accumulate,
in signed and unsigned versions.

*hi-lo -= (long long) s * (long long) t;*
d = lo;

3. 舉例說明使用 MIPS 特殊指令集加速

(1) 敘述：將 4 bytes 的值，從正序改為反序
示意：0x 11 22 33 44 → 0x 44 33 22 11

C 語言

```
a0 = 0x11223344  
v0 = (a0&0x000000ff)<<24 |  
      (a0&0x0000ff00)<<8 |  
      (a0&0x00ff0000)>>8 |  
      (a0&0xff000000)>>24
```

編譯 C 語言後的結果

```
andi  a1, a0, 0xff00  
sll   a1, a1, 0x8  
sll   v0, a0, 0x18  
sr    v1, a0, 0x8  
or    v0, v0, a1  
andi  v1, v1, 0xff00  
or    v0, v0, v1  
srl   a0, a0, 0x18  
or    v0, v0, a0
```

組合語言加速

```
wsbh  v0, a0  
rotr  v0, v0, 0x10
```

- (2) 敘述：將 4 個 4 bytes 的值，兩兩相乘後相加
示意： $a0 * a1 + a2 * a3$

C 語言

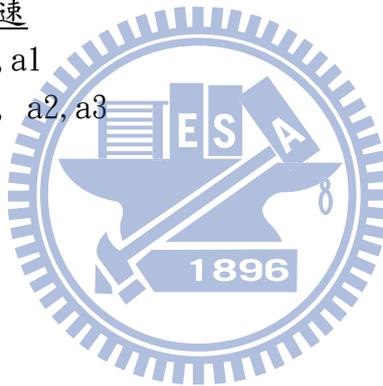
```
v0 = a0 * a1  
v1 = a2 * a3  
v0 = v0 + v1
```

編譯 C 語言後的結果

```
mult  a0, a1  
mflo  v0  
mult  a2, a3  
mflo  v1  
add   v0, v0, v1
```

組合語言加速

```
mult  a0, a1  
madd  v0, a2, a3
```



3.6.2.3. 最佳化的結果分析

在此分析中，依據音訊格式的特性：取樣頻率、Bitrates 和 Channel 數量，分別測試。

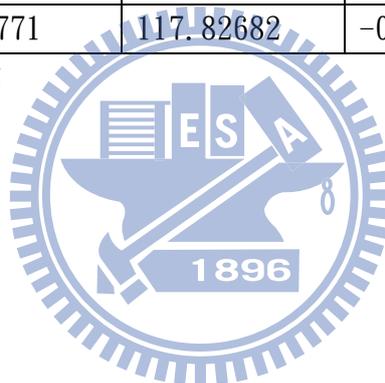
表 8 – 最佳化結果（解碼時間）

| 測試檔案名稱 | 取樣頻率(KHz) | Bitrates(Kbps) | Channel Numbers | 最佳化(%) |
|-----------------|-----------|----------------|-----------------|------------------|
| acmod2_0.ac3 | 48 | 448 | 2 | 22.871479620956 |
| acmod3_0.ac3 | 48 | 448 | 3 | 22.7804813437226 |
| acmod2_2.ac3 | 48 | 448 | 4 | 24.1420617853098 |
| acmod3_2.ac3 | 48 | 448 | 5.1 | 24.8004495759494 |
| data32.ac3 | 48 | 32 | 2 | 19.0882672694264 |
| data320.ac3 | 48 | 320 | 2 | 23.5582378752751 |
| data640.ac3 | 48 | 640 | 2 | 24.0000559496342 |
| dtrtdvd.ac3 | 48 | 變動 | 2 | 1.94077284593015 |
| 5_smp132.ac3 | 32 | 448 | 5.1 | 74.723501808631 |
| 5_smp144.ac3 | 44.1 | 448 | 5.1 | 54.445786820582 |
| 5_smp148.ac3 | 48 | 448 | 5.1 | 50.0822761280063 |
| 5_smp1rtswp.ac3 | 變動 | 448 | 5.1 | 10.8103297341897 |

表 9 – 最佳化結果 (音訊品質)

| 測試檔案名稱 | MSE | PSNR(dB) | *ODG |
|-----------------|----------|------------|--------|
| acmod2_0.ac3 | 0.000966 | 120.458382 | -0.182 |
| acmod3_0.ac3 | 0.0016 | 118.266238 | -0.185 |
| acmod2_2.ac3 | 0.001438 | 118.732 | -0.143 |
| acmod3_2.ac3 | 0.001743 | 117.894727 | -0.162 |
| data32.ac3 | 0.001014 | 120.247494 | -0.136 |
| data320.ac3 | 0.00104 | 120.139937 | -0.182 |
| data640.ac3 | 0.00104 | 120.139937 | -0.182 |
| dtrtdvd.ac3 | 0.001057 | 120.068411 | -0.182 |
| 5_smp132.ac3 | 0.001737 | 117.910349 | -0.16 |
| 5_smp144.ac3 | 0.001696 | 118.014743 | -0.165 |
| 5_smp148.ac3 | 0.001857 | 117.620292 | -0.178 |
| 5_smp1rtswp.ac3 | 0.001771 | 117.82682 | -0.178 |

*ODG 評分於之後說明



客觀差評分 (Objective Difference Grade, ODG)

是經由音訊品質感知評估 (Perceptual Evaluation of Audio Quality) 演算法，採取參考信號的樣本與經過編碼器輸出信號的樣本，透過平行輸入至感知聽覺模型來比較。此感知模型系統的輸出，是模擬音訊經過人類聽覺系統的所有內部代表型態，再由此輸出端的內部代表型態產生可聽性誤差 (audible difference)，然後再由大腦認知模型 (模擬大腦聽覺處理，感知聲音信號的過程) 來計算出一量化的數值，此數值稱之客觀差評分 (Objective Difference Grade)。其數值的範圍為 0 至 -4，0 代表與原參考信號幾乎一樣，-4 代表失真程度已經非常惱人，各數值的詳細定義如下：

表 10 – 客觀差評分表

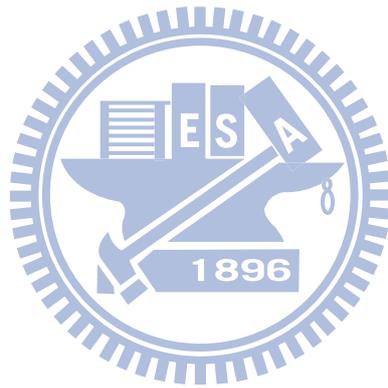
| ODG | 失真程度描述 (Impairment Description) |
|------|---|
| 0.0 | 察覺不出 (Imperceptible) |
| -1.0 | 可察覺但不惱人 (Perceptible, but not annoying) |
| -2.0 | 稍微惱人 (Slightly annoying) |
| -3.0 | 惱人 (Annoying) |
| -4.0 | 非常惱人 (Very annoying) |

3.7. 測試

經過系統的實作後，本平台透過三種驗證方式，將多媒體的檔案逐一測試，以確保視訊和音訊是否正常播放、影音是否同步或其他系統穩定度問題。

驗證方式：

1. 透過 CPU 提供之暫存器，推算所需的效能。
2. 透過硬體提供的暫存器，得知視訊或音訊是否有異常情況發生。
3. 透過使用者操作和播放



3.8. 系統實作成果



圖 4- 操作畫面 1

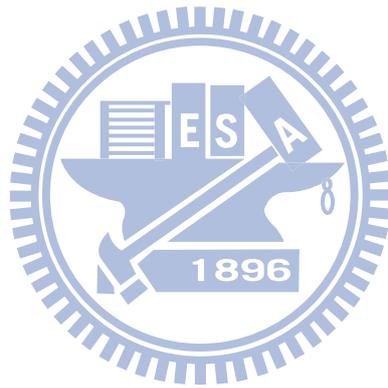
圖 3- 操作畫面 1



圖 5- 操作畫面 2



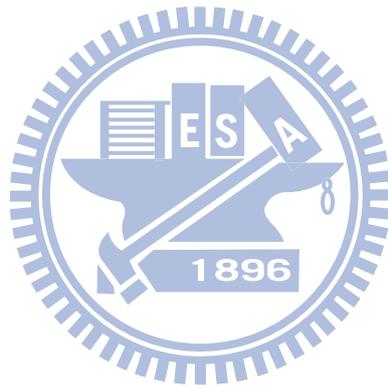
圖 6- 操作畫面 3



四、 總結和未來展望

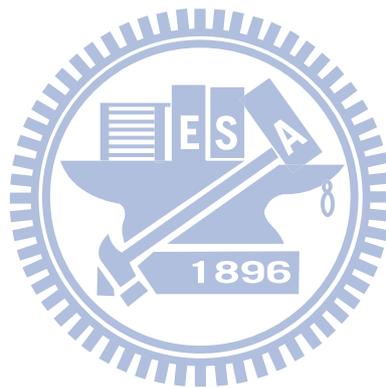
本研究成功的整合多媒體播放於此嵌入式平台中，並可符合先前所設定的檔案格式。但對於系統的穩定性和檔案格式的支援程度，可以再做更完美的改進。

隨著雲端技術與寬頻網路的進步，串流多媒體的普及趨勢越來越明顯。因此多媒體的網路化，讓數位電視會有更多的變化與發展，例如：瀏覽器功能、社群網站功能等…。因此加入網路介面（無線或有線），將多媒體的支援延伸相信會是未來的方向之一。



參考文獻

- [1] Jonathan Corbet, Alessandro Rubini, Greg Kroah-Hartman, Linux Device Drivers 3e, 林長毅 編譯, Linux 驅動程式第三版, 美商歐萊禮, 2006
- [2] Dominic Sweetman, See MIPS Run, Second Edition, Denise E. M. Penrose, United States, 2007
- [3] Andreas Spanias, Ted Painter, Venkatraman Atti, Audio Signal Processing and Coding, 2007
- [4] ITU-R Recommendation BS.1387, 2001



簡歷

1980 年 出生

1992 年 中平國小畢業

1995 年 中壢國中畢業

1998 年 新興工商普通科畢業

2002 年 中央大學資訊管理學系畢業

2004 年 5 月至今 矽統科技股份有限公司軟體工程師

