

# 國立交通大學

## 電機與控制工程研究所

### 碩 士 論 文

智慧型數位影像監控開發平台之實作與應用

An Intelligent Surveillance System

Based on Advanced Digital Development Platform

研 究 生：劉開澄

指 導 教 授：林進燈 教授

楊谷洋 教授

中 華 民 國 一 百 年 八 月

智慧型數位影像監控開發平台之實作與應用

An Intelligent Surveillance System  
Based on Advanced Digital Development Platform

研究生：劉開澄

Student : Kai-Cheng Liu

指導教授：林進燈

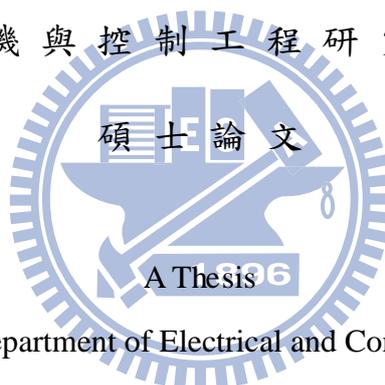
Advisor : Chin-Teng Lin

楊谷洋

Advisor : Kuu-Young Young

國立交通大學

電機與控制工程研究所



Submitted to Department of Electrical and Control Engineering

College of Electrical Engineering

National Chiao Tung University

in partial Fulfillment of the Requirements

for the Degree of

Master

in

Electrical and Control Engineering

August 2011

Hsinchu, Taiwan, Republic of China

中華民國 一 百 年 八 月

# 智慧型數位影像監控開發平台之實作與應用

學生：劉開澄

指導教授：林進燈 教授  
楊谷洋 教授

國立交通大學 電機學院 電機與控制工程研究所

## 中文摘要

在目前的與影像相關的應用中，如大樓監視系統、車用電子、死角監控、危險區域管制等等方面，大多系統仍然需要人類的監控作為依據，如加上了機器視覺的功能，仍然有許多的影像演算法無法達到 Real-Time。由於監視器畫素許多皆為 DI 或 VGA 畫質，無法有效的得到有效資訊；如果使用高解析之攝影機，更多的演算法無法實現於系統中。

本論文針對這類的應用提出一個較為平衡的概念，使用部分的硬體影像演算法作為影像「前處理」、「後處理」等等的運算，這些「前處理」、「後處理」在影像處理的概念中有許多相同的部分，這些項目由硬體快速的完成處理，並不用耗損過多的微處理器的負載，當微處理器的負載越低，所能加入的應用就可以越高，並且透過適當的修改完成 Real-Time 的需求。

在韌體框架中，修改原廠提供的韌體建立出符合影像處理角度的應用框架，使用簡單的指令就可以完成轉編譯的動作，並且新增了些易於開發者使用的子程式，以利便於後續開發的應用，使得開發的速度可以加快。

在系統的軟體應用中，本文提出了一個廣泛型的應用層面，包含背景的建立、物體的追蹤、物體特徵點的選取、判斷警報的條件等等的層面，皆可以做更深入的探討，在此論文中僅做出一個目標性的應用，可由開發者自行往更深入的演算法探討。

此些動作可以使系統軟硬體共同運算加快影像演算法的效率，並

可以使開發者專注於影像演算法的構思，避免周邊設施的複雜影響了開發效率，也同時使系統共同運算邁進了一步，未來更可以使用於許多與影像處理有關的設備。



# **An Intelligent Surveillance System Based on Advanced Digital Development Platform**

Student : Kai-Cheng Liu

Advisors : Prof. Chin-Teng Lin

Prof. Kuu-Young Young

Institute of Electrical and Control Engineering  
National Chiao Tung University

## **ABSTRACT**

In the current video-related applications, such as building surveillance systems, automotive electronics, corner monitors, hazardous area. Most system still require human monitoring as basis, if the machine has vision functions, there are still many image algorithms can not achieve Real-Time. Because most of monitor pixels are still D1 or VGA quality, it's unable to obtain useful information effectively. and if high resolution of the camera is used, more algorithms can not be achieved in the system.

This thesis proposed a more balanced concept for these kind of applications, used parts of the hardware algorithm to calculate "pre-treatment," and "post processing" of images. There are many same concept in "pre-treatment," and "post processing" of the image processing, these projects complete by fast hardware processing which doesn't waste too much loading of microprocessor. When the loading of microprocessor is lower, the joined the application could be higher, and reach Real-Time through appropriate modifications.

In the framework of firmware, modifying the original firmware-to establish applications of image processing, using simple instructions to cross-compile. Adding easier programs to use the function following-up to facilitate an easy application development, making development speed up.

In the application of system, this thesis proposed a widespread application level, including establish the background, object tracking, object feature points selected to determine the level of

alarm conditions, etc., All could do more in-depth discussion, this thesis only hypothesize a goal of applications, other developers–do more in-depth study of algorithms by themselves.

These applications could common compute the hardware and software of the system which enhance the efficiency of image alorisms, and make designer more concentrate on the idea of image algorithms, avoid the complex surrounding facilities environment effect developing efficiency. These applications make co-operation system has a step forward, and it can apply to equipments which related image processing in future.



## 致 謝

在這段研究所的生涯中，首先要感謝的是我的指導教授 林進燈博士及共同指導教授 楊谷洋博士，在兩位教授豐富的學識當中，讓我學習到許多寶貴的知識以及待人之道理、態度等等方面的成長，並且在研究學習的過程中教授們對於學問深入的看法剖析、清晰的思路等等過人之處，都是本人該學習的目標，並且以和藹的態度對任何的人事物，都是該學習的。

在實驗的階段中，特別感謝的是直屬的指導學長 沈子貴博士，在學長細膩思考過程中，感覺不到太多的上下屬關係，反倒是朋友的關係還多一些，在學長實作上及學理上也挖到了許多寶物，使我的研究更上層樓；同時要感謝 UVLAB 的各個學長 蒲鶴章博士、范剛維博士、楊建霆博士、Linda 博士、肇廷學長、東霖學長、勝智學長、Mukesh 學長、陳忠和博士等人，不僅在研究上提供了許多寶貴意見，也在生活態度上得到很大的想法。除了這些實驗室學長們的照顧，也謝謝各個同學弼文、奎理、郁光、慶峰、庭立、奕竹等人的互相支持、討論，互相分享一些不同的經驗，使我更加成長。也謝謝實驗室的學弟妹們，在完成論文時給予許多幫助。

特別要感謝我的家人、朋友，在這兩年內給予我許許多多的鼓勵與支持，使我不致荒廢課業，並且在憂愁的時候可以訴苦，並且讓我有更多更大的信心完成國立研究所的學歷。

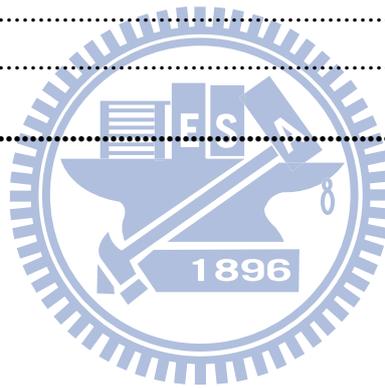
最後，把我最大的祝福獻給曾經幫助過我的家人、朋友、同學、師長們，祝福幫助過我的人們能夠更上層樓。



# 目錄

中文摘要.....	i
ABSTRACT.....	iii
致 謝.....	v
目 錄.....	vi
表目錄.....	viii
圖目錄.....	ix
<b>第一章 緒論.....</b>	<b>1</b>
1.1 研究背景與動機.....	1
1.2 研究相關架構.....	2
1.3 主要貢獻.....	2
1.4 論文主要架構.....	3
<b>第二章 相關工作說明.....</b>	<b>4</b>
2.1 ADSP-BF561 處理器.....	4
2.2 DaVinci相關晶片.....	5
<b>第三章 嵌入式影像系統之硬體實作.....</b>	<b>8</b>
3.1 簡介.....	8
3.2 參考設計與修改.....	9
3.3 電路系統電源端.....	10
3.4 外部儲存連接介面(External Memory Interface, EMIF).....	11
3.4.1 電路系統-RAM.....	12
3.4.2 Flash.....	13
3.5 系統輸入/輸出.....	13
3.5.1 數位影像輸入.....	14
3.5.2 類比影像輸入.....	14
3.5.3 類比影像輸出.....	16
3.5.4 UART.....	16
3.6 電路圖設計完成後續步驟.....	18
3.6.1 電子元件的選取.....	18
3.6.2 電路邏輯設計之製作.....	19
<b>第四章 嵌入式影像系統之應用軟體實作.....</b>	<b>21</b>
4.1 韌體架構.....	21

4.1.1	韌體架構簡介.....	21
4.1.2	Boot-Loader.....	22
4.2	系統中使用硬體演算法.....	24
4.2.1	影像串流.....	24
4.2.2	資料訊號流程.....	25
4.3	系統之應用.....	26
4.3.1	概念簡介.....	26
4.3.2	便利編譯與執行.....	27
4.3.3	取得前景.....	29
4.3.4	電子牆.....	32
<b>第五章</b>	<b>實驗結果.....</b>	<b>34</b>
5.1	硬體系統規格比較.....	34
5.2	軟體成果.....	36
<b>第六章</b>	<b>結論與未來展望.....</b>	<b>42</b>
6.1	結論.....	42
6.2	未來展望.....	43
<b>參考文獻.....</b>		<b>44</b>



## 表目錄

表 1 純軟體系統、純硬體系統與系統整合運算之優缺點比較.....	1
表 2 BF561 與 DM368 之影像周邊相關比較.....	4
表 3 DM3730 及 AM3894 與 DM368 的規格比較表格.....	5
表 4 TMS320C6A8168 及 OMAP3530 的規格比較表格.....	6
表 5 TMS320DM6437 及 DM368 的規格比較表格.....	7
表 6 TMDXEVM368、EUCDK 及本文系統的相異處之比較.....	34



## 圖目錄

圖 3- 1 系統硬體設計之流程圖 .....	8
圖 3- 2 EUCDK之外觀圖 .....	9
圖 3- 3 系統整合後的架構圖 .....	10
圖 3- 4 外部變壓器接頭端至後續供電的電路圖 .....	10
圖 3- 5 電源轉換晶片輸出端加上了鐵粉芯 .....	11
圖 3- 6 儲存電能的電容 .....	11
圖 3- 7 參考記憶體與DM368 之晶片所提供的連結方式.....	12
圖 3- 8 參考記憶體與DM368 所提供的連結方式.....	13
圖 3- 9 數位影像的輸入端子.....	14
圖 3- 10 處理類比轉數位影像的晶片[12].....	15
圖 3- 11 控制切換之晶片.....	15
圖 3- 12 DM368 Composite Video作為類比影像輸出端子的連接方法之一 ...	16
圖 3- 13 RS232 外部連結裝置.....	17
圖 3- 14 DIP包裝(1pcs) 及 SOP包裝(10pcs) 之比較圖 .....	18
圖 3- 15 正面及反面最後成品的樣式圖 .....	19
圖 3- 16 洗板後的裸板正面及反面 .....	20
圖 3- 17 上件完成正面及反面 .....	20
圖 4- 1 修改之 Driver，以及其功能架構圖.....	21
圖 4- 2 載入 U-boot 的流程.....	22
圖 4- 3 DM368 燒入 Boot-Loader 過程畫面.....	23
圖 4- 4 掛載 OS 系統之軟體層架構.....	23
圖 4- 5 執行了「Hello World」程式 .....	23
圖 4- 6 影像處理之系統運作之方塊流程圖 .....	25
圖 4- 7 720P + QVGA 畫質影像資料流程訊號圖[15].....	26
圖 4- 8 「電子牆」之流程 .....	27
圖 4- 9 Makefile 的概念圖.....	27
圖 4- 10 Rules.make 的設定樣式[13].....	28
圖 4- 11 連結各個資料夾的 Makefile 格式[13].....	28
圖 4- 12 Shell-script[13]的使用方式 .....	29
圖 4- 13 韌體層執行後的畫面 .....	29
圖 4- 14 GMM 的示意圖.....	30
圖 4- 15 使用 GMM 概念建立背景 .....	31
圖 4- 16 建立背景後找到的前景 .....	31
圖 4- 17 此系統的追蹤機制 .....	32
圖 4- 18 具有「方向性」的電子牆示意圖 .....	32
圖 4- 19 具有「方向性」的電子牆表現方式 .....	33

圖 5- 1 為 TI 及 Leopard 所推出之系統.....	35
圖 5- 2 此論文提出之連接範例之硬體系統.....	35
圖 5- 3 程式的包裝方式.....	37
圖 5- 4 為 DM368 之人臉辨識結果.....	37
圖 5- 5 韌體之各個程式結果.....	38
圖 5- 6 使用 GMM 建立背景後的圖.....	38
圖 5- 7 驗證電子牆之可行性的實驗結果.....	39
圖 5- 8 使用 GMM 建立背景後的圖.....	39
圖 5- 9 電子牆之應用結果.....	41



# 第一章 緒論

## 1.1 研究背景與動機

人類的眼睛很敏銳，對於居家安全、大樓住戶安全等等的安全性監控於人眼的觀察下是很容易的，但是對於機器視覺來說則有一定的難度，由於近年的保全系統皆為少數人監控許多不同的監測環境，青壯年人數無法完全兼顧老年及幼童，勢必有部份要由「機器智慧」來幫助監控生活週遭的人群監控，這時候機器視覺相關技術就顯得重要，不僅可以節省人力，未來更可以發展於機器人的相關應用，幫助人類做許多繁雜、危險、無趣等等的行為。

對於影像處理來說，例如大樓出入口、交通路況偵測、車用電子等等的用途，對人力資源有一定的節省功用，也可避免人力的疲憊等等行為而錯過了重要的判斷。藉由影像監控系統，可以大幅的降低人類做些簡單而無趣又或者是危險性高的工作。

目前影像處理大略分為三種類型：純軟體運算、純硬體運算以及系統整合運算 (Hardware and Software Co-design)。對於發展機器視覺演算法來說，純軟體是最有利的，通常在個人電腦端運作，因其調整空間較大，且設計時間相較為短暫，但其缺點為整合不易、同演算法下處理時間較長等等；純硬體實現演算法則可以達到最好的演算法執行速度，但其缺點為成本高、相對發展時間長；系統整合運算則是各取其優點的方法，一些較為常用的影像演算法使用硬體實現，又可以依照設計者想要的方式去實現後續處理，保留了純軟體的彈性空間，同時又可以加快一些前、後段影像處理的時間。對於開發機器視覺、車用影像、遊戲娛樂等等的實際應用有許多幫助。

表 1 純軟體系統、純硬體系統與系統整合運算之優缺點比較

	優點	缺點
純軟體系統	自由度高 個人電腦介面居多 發展時間較短	運算速度較硬體慢 相容性差
純硬體系統	運算速度較純軟快 較穩定	發展時間較長 使用者自由度高 高花費
系統整合運算	較純硬體高自由 發展時間較純硬體短 較純軟體運算快	難以整合，效能與整合有 強烈關係

德州儀器於 2010 年 4 月推出了 TMS320DM368[1] SOC 多媒體處理器，許多影像「前處理」部分如同前段所說的，已經用硬體去實現了，也同時符合前段系統整合運算所述，以及下段提到的一些硬體功能，對於數位影像發展有著重要的意義。在此提出一套利用 DM368 SOC 晶片為主要運算器組成的系統，可以為智慧型機器視覺提出一套使用方案，以及配合上 DM368 之影像運算子系統、韌體層最後組合成為此篇提出的應用-「電子牆」。

## 1.2 研究相關架構

本論文的影像處理平台設計及應用，整體而言大致上分為三大項，分別是硬體架構、韌體層及應用軟體層，在此論文提出的組合應用中，皆是以各個軟、硬體的流程及圖例提出。

在硬體架構中，提出一套周邊電路組合應用，把推出不久的 DM368 晶片，整合影像處理相關的晶片，使其可以移植後續的軟、韌體以利發展出一套可隨發展者需求之系統整合平台。

在韌體層當中，依據所需要的硬體演算法對應之記憶體位置，抓取後續要繼續處理的影像資料，並且把各層級的影像資料做適當的排序、運算以及處理，其中開發此系統部分的演算法也在此層實現。

應用軟體層為了驗證上述系統是否可行，並且組合各個影像處理子系統的功能，配合上現今影像處理常用的演算法，進而發展出來一套「電子牆」的應用軟體。根據環境的不同，此層的彈性也是最大，例如：開發演算法、調整參數、偵錯、驗證、測試等等的需求皆於此層實現，最後把處理過後的資料，放到硬體層負責撥放畫面的記憶體中，就完成了一張畫面的周期。

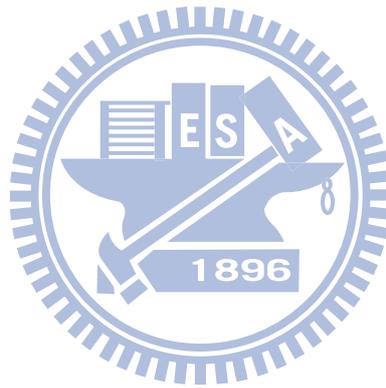
## 1.3 主要貢獻

此系統提供了 DM368 SOC 晶片的影像處理硬體周邊所需架構，與原始系統比較，組合了 RAM、FLASH、UART 等等介面應用，並且修改成為影像處理系統獨一無二的硬體組織；軟體部分結合韌體的組合方法，並且修改韌體成為讓開發者更容易上手之子程式呼叫介面，配合上硬體影像子系統，再根據軟體部分的演算法，組合出「電子牆」。因 DM368 於 2010 年 4 月底推出，還有許許多多的應用組合並未開發，在此 DM368 為微處理器的系統，還有許多尚未開發之部分，僅列出「電子牆」及「人臉偵測」所使用之部分影像子系統，雖不至理想，但對於機器視覺來說，已經往系統整合運算的階段又多邁向一步了，並且對於未來使用 DM368 做 1080p 的智慧監控系統，透過網路傳輸 H. 264 視訊壓縮必有實質上的幫助。

## 1.4 論文主要架構

此論文之實做為 DM368 相關的各個晶片使其發揮功用，並且透過韌體層面連結作業系統以及硬體，大致上分為兩個部分：硬體架構以及軟體架構，其中硬體架構以結合各個晶片相互的聯結為主要描述，著重於各個流程的延續以及整個系統大局的樣式；軟體架構分為兩種，韌體與應用軟體，這兩種軟體層在此篇論文中所撰寫的方式也是以整體系統架構流程，許多參考文獻演算法並未詳盡描述，僅描述修改後的呈現方式。

此論文中在各個章節，皆以大方向的流程圖表式之，大致上以圖文方式呈現。最後比較此篇論文提出的系統方案與市面上以 DM368 為處理器之系統差異，以及系統整合運算之成果展現。



## 第二章 相關工作說明

在此章當中，分別介紹一些相關的嵌入式影像處理平台，包含了兩大廠商的 ARM 系列及 DSP 系列的處理器，對於上列兩大類系統與此篇論文提出之系統架構分別做比較。目前多媒體處理器之系統有許多廠商推出，其中取較為知名之兩間公司作為相關搜尋，一為 ADI 公司之另一則為 TI 公司。

### 2.1 ADSP-BF561 處理器

此處理器為 ADI 公司推出的 Blackfin 系列之運算處理器 ADSP-BF561[2]，Blackfin 系列之運算處理器大致上有一個高性能的 16/32 bits 嵌入式處理器內核與一個 10 級 RISC MCU/DSP Pipeline。微處理器運算速度相較 DM368 下快了許多，但有許多子項目是 BF561 無法比擬的，與影像相關之功能比較如下表

表 2 BF561 與 DM368 之影像周邊相關比較

	BF561[2]	DM368[1]
MCU	DSP 600MHz Dual Core (Blackfin)	ARM 432MHz (DaVinci)
Instruction SRAM	16Kb	16Kb
Data SRAM	32Kb	8Kb
Codec. Engines	None	Supports a Range of Encode, Decode,
DMA	Dual 12 channel	64 Channels
Peripherals	UART with IrDA 12 GPIO 32-bit timers	EMAC, 2 UARTs, 4 GPIO 64-Bit timers
Video ALU	Four 8-bit	2 Co-processors 8/16 bit
Video Processing Subsystem	None	Yes

在第一章時提到希望以軟硬體系統共同運算，BF561雖有雙核心，但缺乏了影像處理之子系統，取而代之的是包含像處理子系統的TI的DaVinci系列SOC多媒體晶片。

從晶片資料中做基本的比較後，雖BF561運算速度快，快取記憶體較多，同時也是多核心，但少了網路介面與影像Codec. Engine，需要傳輸影像時或是需要遠端控制時還需要額外裝置來輔助；尤其像H.264及MPEG-4影像壓縮格式來說，

近年來有越來越流行的趨勢，可為串流影像節省下不少運算量；就影像處理方面來看，雖BF561核心運算速度較為快速，但以影像處理來說DM368有個決定性的優點，也就是包含有DaVinci之「影像處理子系統」。

## 2.2 DaVinci 相關晶片

在上節提到，決定TI公司的DaVinci相關晶片為微處理器的關鍵為，DaVinci相關晶片有各種不盡相同的「影像處理子系統」，有DSP系列、ARM系列、C64+系列、雙核心系列等等許多不盡相同類型及不同組合型態的微處理器，DaVinci系列的SOC晶片就分類為許多種類型，在此列出TI與影像處理相關系列之幾項SOC晶片，以區別選擇他類以及DM368之間的差異。

表 3 DM3730 及 AM3894 與 DM368 的規格比較表格

	DM3730[3]	AM3894[4]	DM368[1]
CPU Processor	ARM Cortex-A8 1000MHz C64+ 800MHz	ARM Cortex-A8 1500MHz	ARM 9 432MHz
RAM	LPDDR	DDR2、DDR3	DDR2
On Chip Memory (KB)	None	576	32
Video Capability	Encode Decode Image Enhance	None	Encode Decode Image Enhance
Video Codecs	H. 264 JPEG MPEG4-SP	None	H. 264-BP JPEG MPEG4-SP
Resolution	720p	1080p	1080p

從上表得知，DM3730及AM3894的運算速度雖然較為快速，但是分別缺少了DM368所擁有的Video Codec及1080p的影像解析度，在目前的影像處理使用中，前兩項微處理其實已經擁有足夠了運算能力，但未來影像發展的Full HD畫質及傳輸優勢便缺乏了，故在此表當中選擇DM368。

接著比較的系列為嵌入式產品中較常見的A8系列及近年較為廣泛流行的OMAP系列

表 4 TMS320C6A8168 及 OMAP3530 的規格比較表格

	TMS320C6A8168[5]	OMAP3530[6]	DM368[1]
CPU Processor	ARM Cortex-A8 1500MHz C67+ 1500MHz	ARM Cortex-A8 600MHz C64+ 800MHz	ARM 9 432MHz
RAM	DDR2、DDR3	LPDDR	DDR2
On Chip Memory (KB)	576	496	32
Video Capability	None	Encode Decode Image Enhance	Encode Decode Image Enhance
Video Codecs	None	H. 264-BP JPEG MPEG4-SP	H. 264-BP JPEG MPEG4-SP
Resolution	No	D1	1080p

先看到的是A8168規格與DM368差異甚大，是TI公司推出很強悍的數位訊號處理器，運算處理速度遠大於DM368，但有關影像處理的部分皆不支援；OMAP3530核心運算速度、快取記憶體皆比DM368高上許多，且也有同DM368的Video Codec能力，但是所支援的解析度僅D1，是最後不考慮此型號的原因。

表 5 TMS320DM6437 及 DM368 的規格比較表格

	DM6437[7]	DM368[1]
CPU Processor	C64+ 700MHz	ARM 9 432MHz
RAM	DDR2	DDR2
On Chip Memory (KB)	240	32
Video Capability	Encode Decode Image Enhance	Encode Decode Image Enhance
Video Codecs	H. 264-BP JPEG MPEG4-SP	H. 264-BP JPEG MPEG4-SP
Resolution	D1	1080p

最後比較的系列為支援「影像處理子系統」比較不同的部分，也就是以C64+為主要運算的SOC多媒體晶片，此款支援TI所免費提供之Vision-Library，Vision-Library為TI所做之影像演算法，並不能修改其內容之Library，且因許多細部參數無法調整，TI官方網站上皆有介紹及簡介影片在此不詳細介紹。雖然DM6437畫面解析度僅有D1畫質，其Vision-Library是非常吸引影像處理相關的人群，但其「影像處理子系統」並不包含硬體演算法，所有的應用程式皆會耗損微處理器資源，並不符合「系統共同運算」之發展精神，及經過實際測試後，雖然Vision-Library是經過優化，但仍然無法有效往後續發展。

在以上是規格上的比較，DM368為最後的微處理器晶片選擇，主要原因有幾項：規格領先、含有影像處理子系統、符合軟硬體共用之發展、價格與規格合理等等的因素，因此從影像處理的角度，茫茫的微處理器晶片選擇當中，選擇TI公司的TMS320DM368。

## 第三章 嵌入式影像系統之硬體實作

### 3.1 簡介

為了可以在U2規格的硬體電路上實現「系統整合運算」之影像處理平台系統，首先考慮到的是取得相關的運算晶片效能，內建的周邊越完整，那麼所需的空間就越少。

在2010年4月，德州儀器(TI)推出全新 TMS320DM368 多媒體處理器，DM368 編碼支援 1080p 高畫質，並且具有 H.264 編碼功能。DM368 是目前高畫質視訊攝影機、數位攝影機 (DVR)、視訊系統等等許多多媒體應用的理想微處理器選擇其中之一。TMS320DM368 DaVinci 視訊處理器的功能與優勢：

- ARM926EJ-S™ 核心可實現超過 40% 的效能提升，頻率提高達 432 MHz，同時還可將視訊編碼／解碼功能交由整合的高畫質視訊加速器完成，進而將系統效能最佳化
- 整合周邊 EMAC、USB 2.0 Phy、16bits DDR2、16 KHz 語音編解碼器、即時時脈(RTC)可將電路板面積降 20% 以上
- 可直接使用免專利費視訊編解碼器 (H.264、MPEG-2、MPEG-4 JPEG 及 VC1/WMV9) 的多格式多速率多串流高畫質影像，不僅可降低設計複雜性，並可加速系統的設計流程

由於 DM368 有了以上其優點，決定了使用此晶片做為多媒體影像處理平台，以 DM368 為核心處理器系統整合運算。對於選定微處理器後的硬體設計，約略可以分為四大步驟，並且以下圖來表示

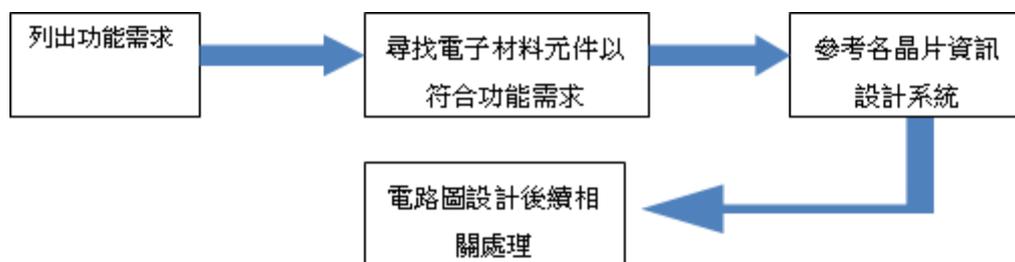


圖 3- 1 系統硬體設計之流程圖

## 3.2 參考設計與修改

此系統利用TI之協同廠商Leopard推出EUCDK[8]處理平台做為參考電路，EUCDK為多媒體處理平台，包含了聲音及影像處理功用，但此平台的設計為兩片電路板透過外部裝置連接，且因並無類比影像輸入之功能，於使用上造成開發者的不便，便從此角度整合修改EUCDK的平台，下圖為EUCDK多媒體平台的外觀圖，

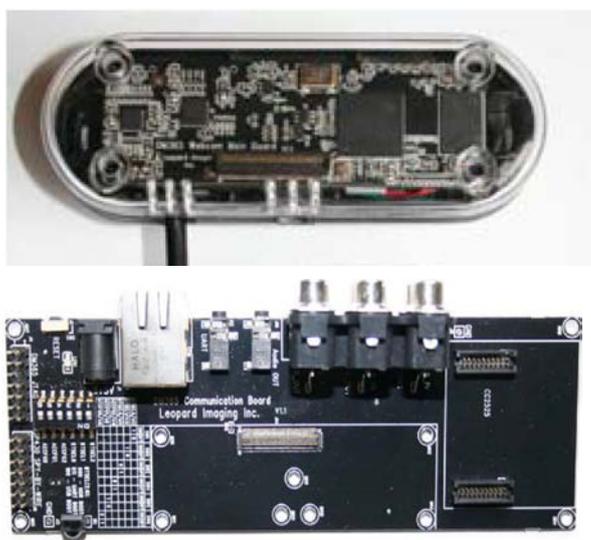


圖3- 2 EUCDK之外觀圖

在實做中，要把上述兩個電路系統取出與影像相關的裝置，並且增加或修改後整合成一個系統，需從各個區塊的電路圖著手修改。於開機時啟動的Boot-Loader，也就是儲存於Flash記憶體裡的外部儲存裝置(EMIF)，安插了一般電子材料行容易取得的記憶體。在此系統能接受數位及類比影像，DM368本身即可以接受數位影像，因此直接配合上Micron公司LI-5M04CM[9]的CMOS型的攝像機。為了能夠處理類比影像，因此也加上了數位轉類比(ADC)之晶片，接著再轉入DM368之數位接受端，並且透過選取的方式，選取希望的輸入影像。UART傳輸接口配合上RS232 驅動晶片，由此介面可與PC做為溝通橋梁。網路(Ethernet)功能於現今的嵌入式平台中，是相當重要的一環，不僅可以用來掛載 Kernel及Filesystem，未來更可以透過網路傳輸影像資料，因此也加入了網路功能。最後一道門，也就是影像輸出部分，DM368提供了類比以及數位輸出的功能，在此數位訊號輸出的安排是由USB介面做為輸出。下圖為系統中以DM368為微處理器及系統周邊的示意圖

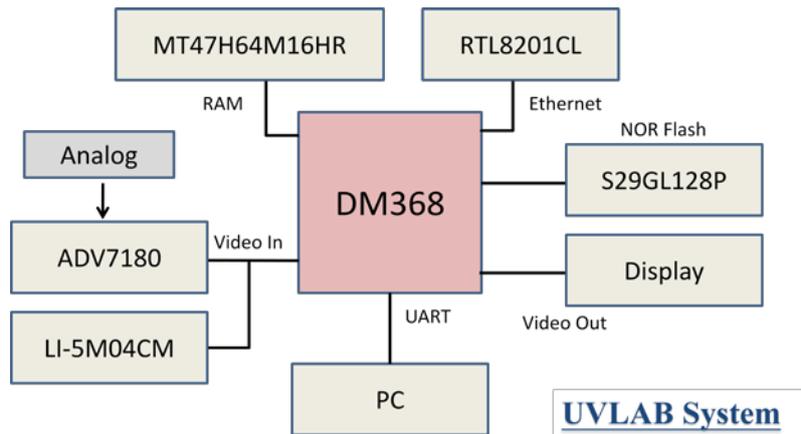


圖3- 3 系統整合後的架構圖

### 3.3 電路系統電源端

不論是怎麼樣的電路，皆需要電源供應，查看了以上的晶片、記憶體、周邊等等的晶片資訊後，使用了三種伏特之電壓，分別是3.3V、1.8V、1.35V的電源，以及初步的估算各項晶片電流使用(Power tree)後，電源接頭端設計5V之電壓即可，確定了電源端及各個元件所需的電力後，便可起始整個系統的設計。首先是外部變壓器進到系統端的電，設計為值5V，但不能保證變壓器的供電品質是否穩當，於是在電源接頭後端接上了大電感及大電容進行首次的穩壓，以確保提供電源轉換晶片的品質。下圖為外部變壓器接頭端至後續供電的電路圖

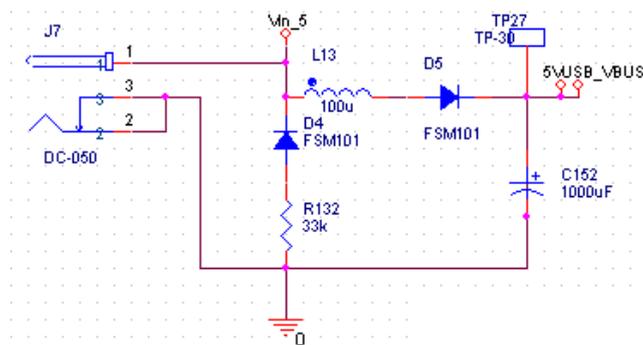


圖 3- 4 外部變壓器接頭端至後續供電的電路圖

J7為變壓器電源輸入接頭，L13及C152為確保提供電源轉換晶片的品質，D4、D5及R132為防止變壓器電源反接的機制

從5V送出的電壓分別接到3.3V、1.8V、1.35V的電源轉換晶片輸入端，電源轉換晶片的選用，於設計時估算後，因電流量不大，且因與變壓器輸入電壓差不大，因此使用線性轉換晶片即可，並且於電源轉換晶片的輸入、輸出部分，皆放

入了小電容當作濾漣波之用，因電源轉換晶片輸出端是直接供給電路電源使用，所以同時在電源轉換晶片輸出端加上了鐵粉芯防止電壓突變之功用，如下圖所示

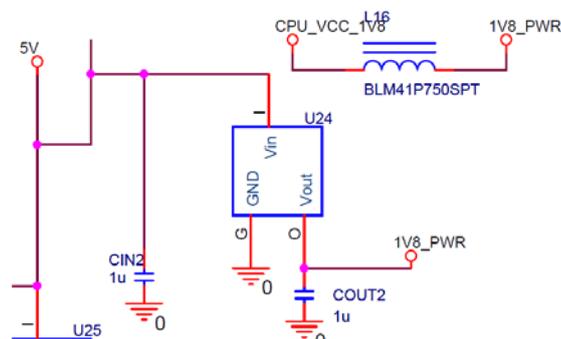


圖 3- 5 電源轉換晶片輸出端加上了鐵粉芯

U24為5V進入的Vin處，CIN2、COUT2分別為輸入輸出時的濾波小電容，L16為鐵粉芯穩壓之功用，CPU\_VCC\_1V8為供給後端電路1.8V之處

在以下提到的晶片接入電源端部分，皆安置了如類似以下的電路，以確保電壓不會因電流瞬間變化太大而不穩定影響太過劇烈，好比起動冷氣機啟動瞬間，燈管會閃爍，因此這部分的儲存電能的電容是必備的

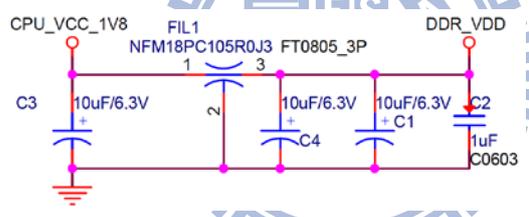


圖 3- 6 儲存電能的電容

C1、C2、C3、C4為避免電流變化太大之電能儲存電容

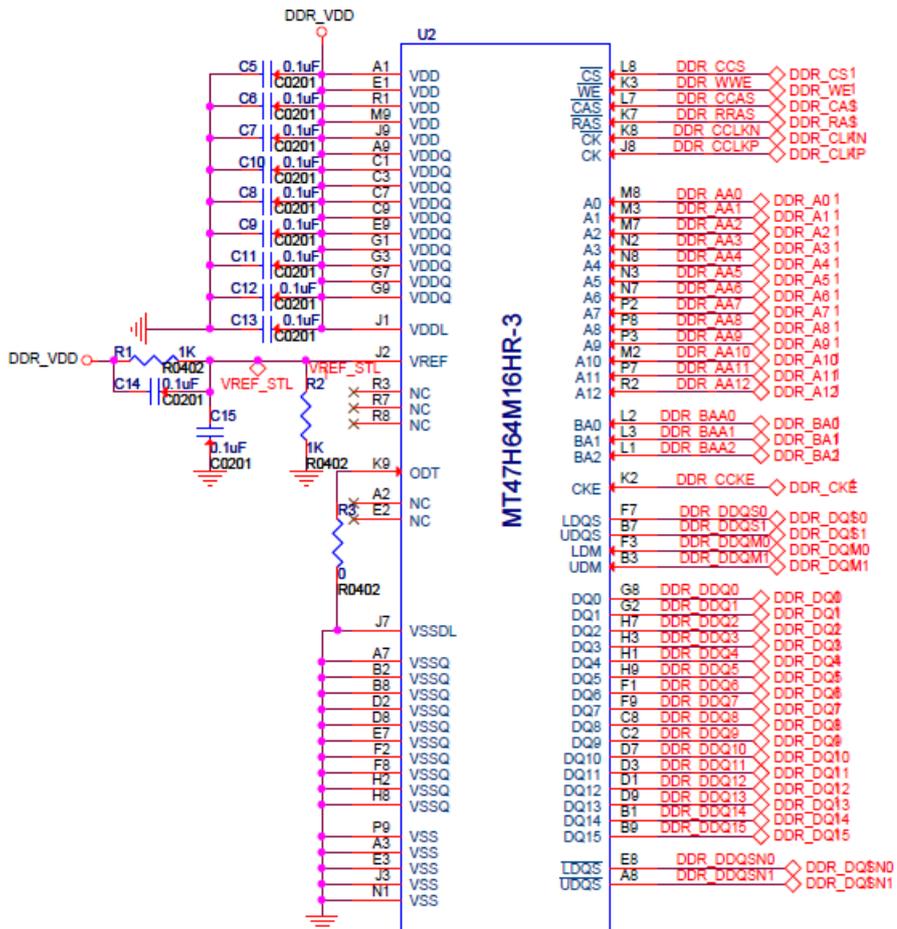
### 3.4 外部儲存連接介面(External Memory Interface, EMIF)

一般來說，EMIF 可實現微處理器與不同類型記憶體 (SRAM、Flash、RAM 等) 的裝置連接。又或者是 SD Card、USB 擴充儲存裝置等等類型的介面。在此系統中，EMIF 與 DM368 相連結，使 DM368 平臺充當一個協同處理器、高速資料處理器或高速資料傳輸介面。

### 3.4.1 電路系統-RAM

DM368 所支援的記憶體為 DDR2 SDRAM (16-bit bus width)，DDR2 有個同步機制，在輸入前等待一個 Clock，這樣就能和系統匯流排同步；DDR2 預讀取為 4 bits，大多是 4 到 6 個匯流排周期的等待；而 DDR 預讀取是 2 bits，大多是 2 至 3 個匯流排周期的等待。意思就是，在同樣的 200MHz 環境頻率下，DDR 的實際運作是 400MHz，而 DDR2 為 400MHz。另外，在 DDR2 標準規定當中，記憶體皆使用 FBGA 的封裝，異於廣泛應用的 TSOP 類型的封裝，FBGA 的封裝型式可以提供較 TSOP 封裝較好的電氣特性及散熱，此特點也為 DDR2 特色之一。

所選擇的 MT47H64M16HR-25:H[10]，128 MB(8Mx16x8 banks) DDR2 RAM，選擇此款記憶體原因如下，面積僅 8mm x 12.5mm、Timing cycle 為 2.5ns @ CL=6(DDR2-800)為 DDR2 SDRAM 當中運作頻率速度領先的幾款之一。下圖為參考記憶體與 DM368 之晶片資訊後，所提供的連結方式之一



### 3.4.2 Flash

NOR 和 NAND 是目前兩種主要的儲存技術方式。Intel 起先開發出 NOR Flash 技術，更變了 EPROM 和 EEPROM 獨佔的情形。接著不久，東芝開發出 NAND Flash 的結構，強調減少了每儲存 1 bit 的成本，並且可以透過外部連接介面增加或替換儲存設備。

因大多數情況下 Flash 只是用來存儲少量的檔案，這時 NOR 比 NAND 適合。NAND 是儲存大量數據資料的解決方式其中之一；NOR 的特點是可在晶片內執行 (XIP, eXecute In Place)，程式可以直接在 Flash 裡運行，不必再把所需要的資料讀取到 RAM 中，且 NOR 在儲存空間不需要太大時有很高的成本效益，但很慢的寫入和清除速度大大影響了它的性能。考量到此系統當中所使用的儲存裝置只需要儲存 kernel 及 Filesystem、Boot-Loader 及其相關設定檔案，皆是不需要常常更新的檔案，所以此系統使用的是 NOR Flash。

考量到需要 NOR Flash 存放 U-Boot 以及環境變數的儲存，所選的是 S29GL128P[11] 當作 Flash 記憶體，此元件的存取速度並不用要求太快，因為只負責儲存更新速度不是太快的資料，Kernel & Filesystem 就是一個典型的範例。在此設計介面底下，可以用來進行存放及掛載 Kernel 及 Filesystem 的動作。下圖為參考記憶體與 DM368 之晶片資料後的連結方式之一

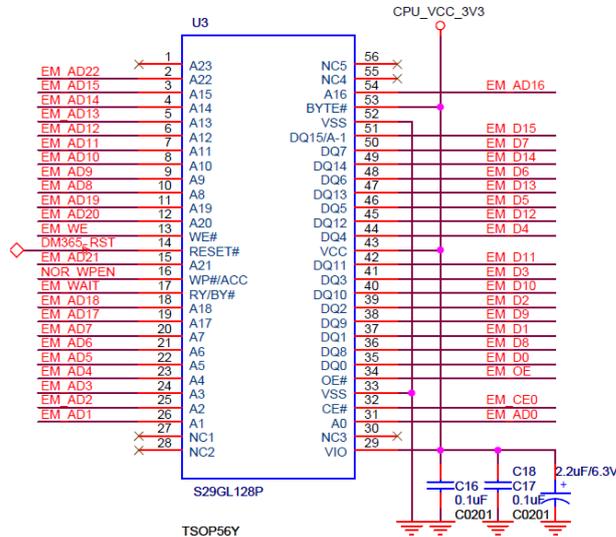


圖 3- 8 參考記憶體與 DM368 所提供的連結方式

### 3.5 系統輸入/輸出

DM368 支援的影像輸入及輸出影像是數位和類比影像格式兼具，DM368 主要訴求為多媒體影音處理器，在於影像內建支援部分，算是相當完整的。輸入部分

在此系統當中類比及數位影像兩者皆可當作輸入；DM368 的影像輸出格式是兩種皆備，故使用 USB 連結當作數位輸出，

### 3.5.1 數位影像輸入

此系統中，考慮的是 HD 高清攝影機，為 DM368 原有之功能模組，也考量到 HD 攝影機的模組光學研究並不是主要的研究部分，加上模組已有小體積的出現，且取得方式並不困難，故數位影像直接使用現成的模組，型號為 LI-5M04CM，模組輸出的格式為 YCbCr -16bits，把 DM368 YCbCr 同是 16bits 的 4:2:2 介面互相對應，連結到外部的模組接頭上，利用此接頭與攝影機模組接合，即完成數位影像的輸入端子部分。下圖當中為此設計的外部接頭示意圖

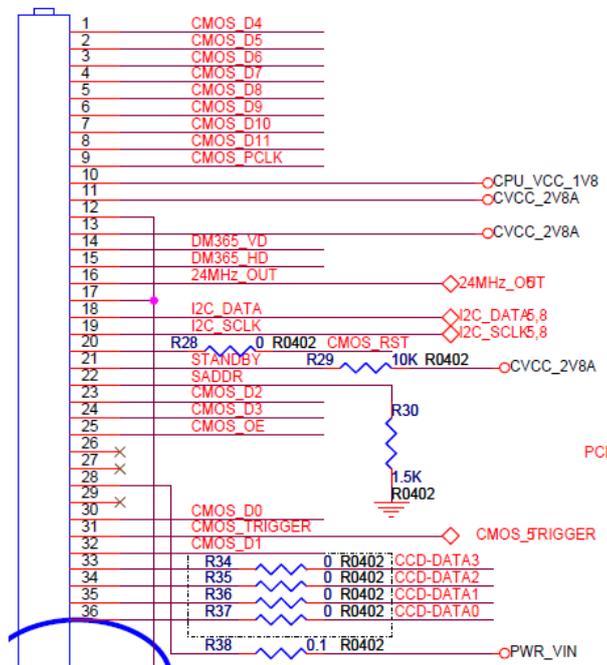


圖 3- 9 數位影像的輸入端子

上圖之 CMOS\_D0~CMOS\_D7 分別為 DM368 的 CIN0~CIN7, CMOS\_D8~CMOS\_D11 及 CCD-DATA0~CCD-DATA3 分別為 DM368 的 YIN0~YIN7

### 3.5.2 類比影像輸入

由前項提及的DM368輸入影像格式雖然支援數位影像格式，但因考慮到如果直接使用此項，在韌體控制層會複雜許多，因此使用額外的ADC晶片做轉換處理的動作，從DM368所支援的輸入影像格式其中一項為8-bit YCbCr數位影像輸入格式去尋找適合的ADC晶片，交叉比對後發現是互相吻合的，故選擇了ADV7180[12]

為處理類比轉數位影像的晶片，該晶片的功能流程如下

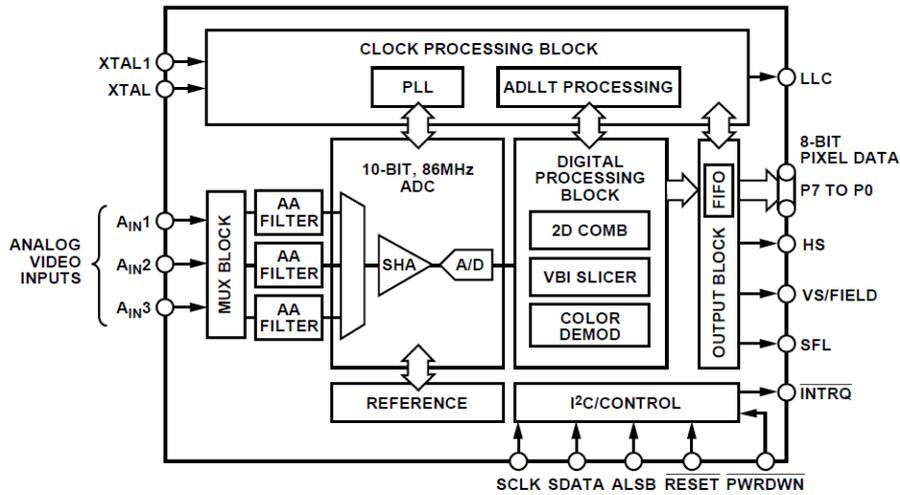


圖 3- 10 處理類比轉數位影像的晶片 [12]

因ADV7180的LLC時脈，必須與DM368的時脈為一致，所以把兩者的時脈相連接，並且接上一個控制SW的IC，從ADV7180輸出的P0~P7、HS、VS有關數位影像的輸出皆以相同的方式接上控制SW的IC，最後才接到DM368的輸入數位影像端，以下為此提到的連接方式之示意圖：

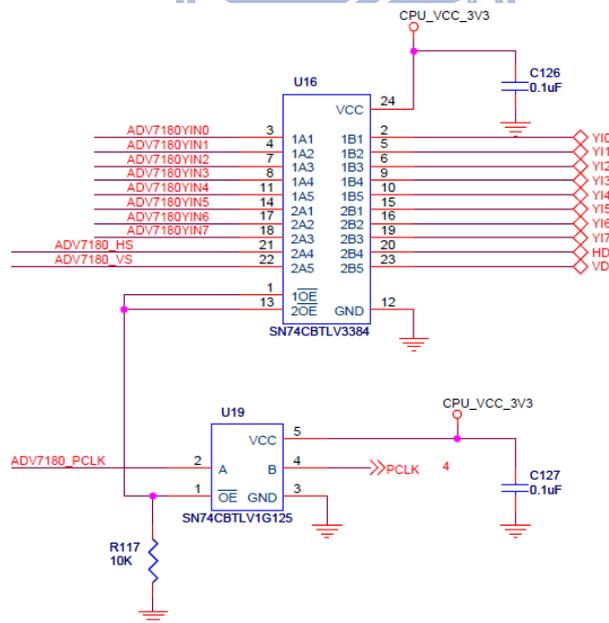


圖 3- 11 控制切換之晶片

上圖U16及U19為控制切換之晶片，Y10~Y17、HD及VD為連結到DM368晶片之接點



再通過 FIFO，如果沒有 FIFO，資料就無法順利存取。在此設計中，UART 用來與個人電腦通訊，用於控制個人電腦與系統的晶片，UART 處理從滑鼠及鍵盤發出的訊號（滑鼠及鍵盤是串列設備），可以處理電腦與外部串列設備的同步管理問題。DM368 也包含了 UART 功能，在此系統中，由於 DM368 輸出的電壓準位為正負 3V，必須透過額外的驅動晶片把電壓準位拉拔至 RS232 介面所需要的電壓準位，並把 RS232 送出的訊號轉換至 TTL 的電壓準位，接著把 DM368 的 TX、RX 端透過驅動晶片連結到外部的接頭，並且在外部接頭的外殼部分安插上了穩定訊號之設計，如下圖表示

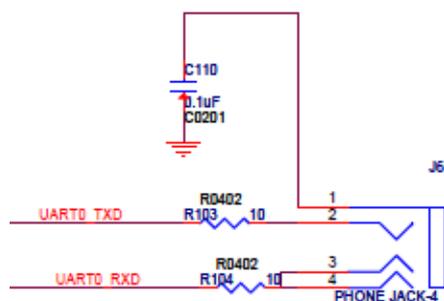


圖 3- 13 RS232 外部連結裝置

UART0 TXD 及 UART0 RXD 分別為 DM368 的 TX、RX 拉出至驅動晶片，並且由驅動晶片端輸出準位處連接到外部接頭，並加上 R103、R104 的小電阻抗雜訊，C110 為外殼的穩定訊號之電容

## 3.6 電路圖設計完成後續步驟

### 3.6.1 電子元件的選取

電子元件的選取對於大部分使用者其實並不是太注重，因大多不在乎系統體積大小，也不考慮電子元件特性，只考慮元件取得的方便性及元件銲接的方便性，因此大多採取DIP封裝的方式。由於此系統在微處理器與記憶體部分只有BGA封裝的型態，也因DIP被動元件體積龐大、接腳部分容易受外界環境影響、功耗高等等的理由，因此晶片部分大多採用SOP封裝、被動元件部分的選取皆以SMD封裝作為考量，以下為本系統平台的選取考量點。

- 面積小(15x15cm)
- 卸除可刪減元件
- 布局元件位置
- 某些位置須留下測試接點

考量面積必須小，元件功率相對低，因此Surface Mount Technology (SMT)的相關元件，為大多數採用的元件類型，同一類型的晶片使用不同的外包裝，面積甚至差異10倍，下圖為DIP包裝以及SOP包裝的實體圖

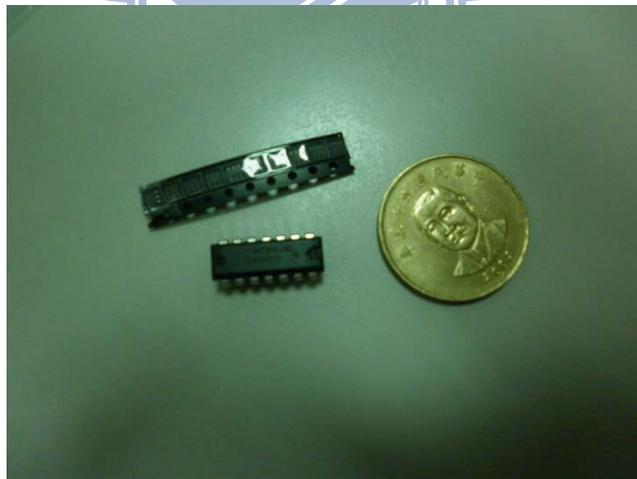


圖3- 14 DIP包裝(1pcs) 及 SOP包裝(10pcs) 之比較圖

挑選元件需要靠的是經驗累積，一樣功能的晶片在不同廠商來說，接腳不一定相同，考量到的不單單是晶片資訊上所顯示的特性曲線了，由於此系統使用的材料僅十套，只能從零售商手中取得，所以來回奔波材料行的動作必不可少，只要確定有該類封裝腳位後，來回檢查電路中使用該晶片的封裝腳位，記得要作封裝腳位更改的動作，以利後續的製作。

### 3.6.2 電路邏輯設計之製作

電路設計的部分完成後，為了確保腳位是否對應、是否重複、元件跨頁連接時是否無誤、是否誤值數字等等一些人為因素，以利後續的電子化製作，比喻為「Compile」的動作，確實的完成此項動作後才可以進行佈局、洗板、上件等等的動作。

佈局首要步驟，參考各個元件晶片資訊，並把每個元件確切的尺寸使用PowerPCB描繪出來，並且把元件擺放於希望的位置，所要注意的是，外部接頭與相關晶片盡可能的擺放在一起，以降低布局走線時的彼此間訊號干擾，此階段描繪出來的樣式，即是最後成品的樣式，此系統的「元件佈局」如下圖所示

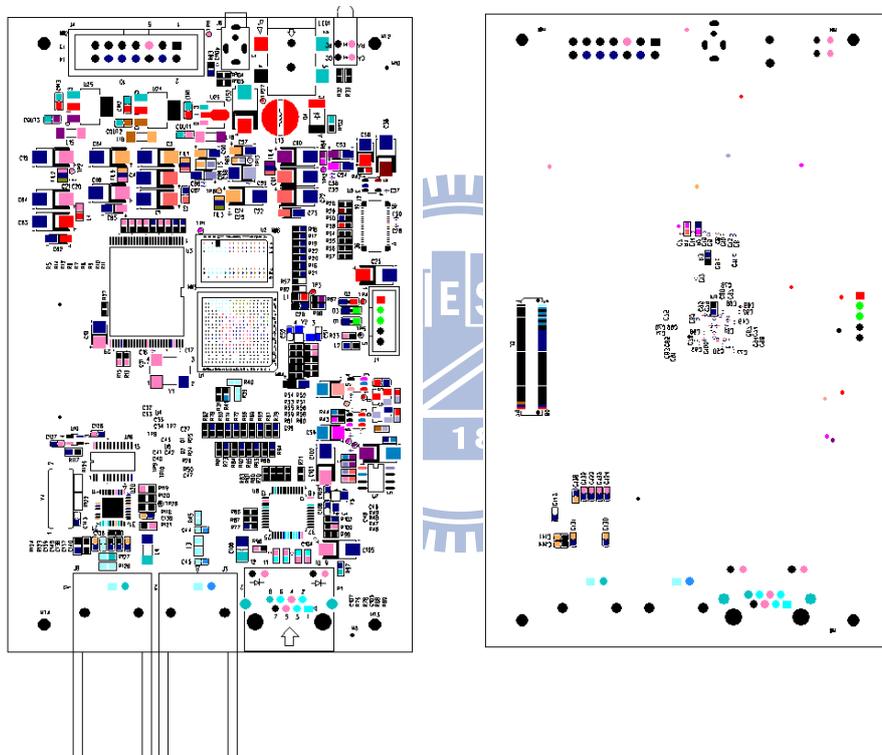


圖3- 15 正面及反面最後成品的樣式圖

在估算後，洗板、上件等動作皆由外包廠商完成。因考慮此動作並不是此篇論文提出方法之主要目的，以及面積關係故使用四層板。及考慮有許多小於0.2mm之接腳難以手工焊接，故這些動作並不多作介紹。在此僅列出洗板後的裸板及上件完成後的完成圖：



圖3- 16 洗板後的裸板正面及反面

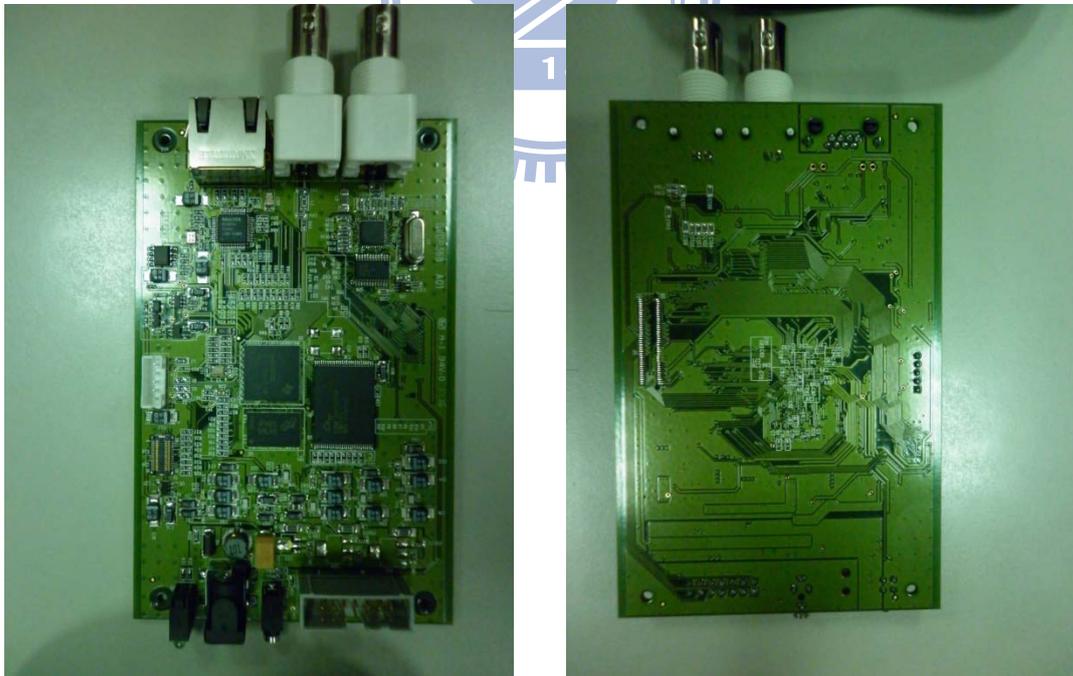


圖 3- 17 上件完成正面及反面

## 第四章 嵌入式影像系統之應用軟體實作

在此章節當中，將介紹如何使韌體控制硬體，並且使用 DM368 之影像處理子系統組成易於軟體開發者之子程式，並且盡可能的符合 RTOS，並且列出使用於嵌入式影像平台演算法，在最後利用上述之修改之韌體相互連結組合成應用程式-「電子牆」。

### 4.1 韌體架構

此節介紹系統中使用的周邊驅動韌體。硬體設施完成後，啟用硬體所支援的架構，在 OS 系統中大多都會透過韌體層作為硬體層與應用軟體層中間的溝通媒介，此系統也是使用相同的概念。

#### 4.1.1 韌體架構簡介

韌體所使用的為 Leopard Imaging Inc 所使用的 Embedded Camera Dev Kit (ECDK) [13] 作為基底，藉由這些基底來發揮 DM368 晶片中的部分特性，實現影像前處理的部分功能，以利達成 RTOS 的精神。

此系統中，修改 ECDK 成為一個利於使用者及應體層接合的應用程式，可以讓使用者自行操作影像的訊號相關處理，以及實現 DM368 中的硬體演算法。整合的系統中，允許使用者修改此韌體內的參數及確切想要的方式。

下圖為此系統修改後的韌體層之功能結構圖

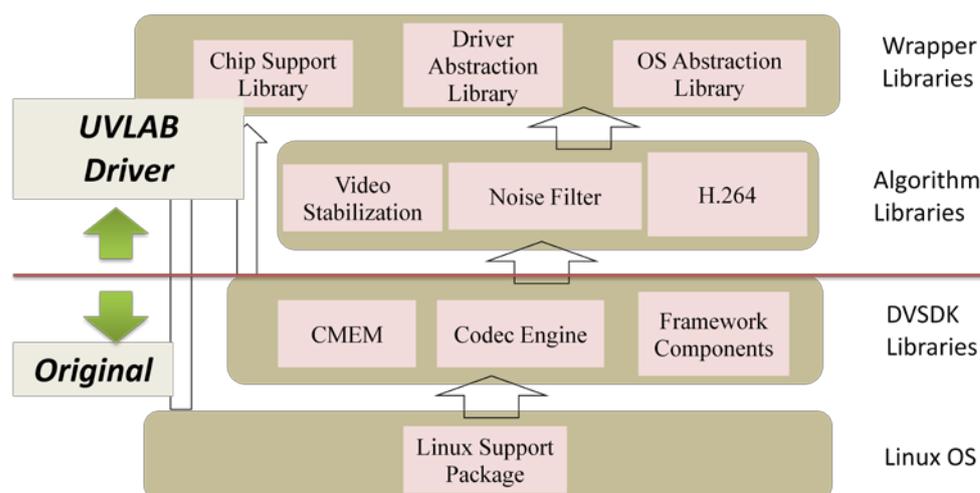


圖4- 1 修改之Driver，以及其功能架構圖

應用範圍僅只於Algorithm Libraries層以及Wrapper Libraries層，接著下來，單利用韌體其實並不夠連結硬體，還有最底層喚醒硬體周邊，控制著是否要發送訊號、開關控制等等的周邊控制—OS作業系統。

## 4.1.2 Boot-Loader

硬體開始運作後，微處理器並不知道要從何處開始運行OS，從Flash、UART、SD卡等等裝置皆可以當作開機位置，進而發展一套想法，首先微處理器讀取的皆是Boot-Loader，接者從Boot-Loader當中的設定環境執行OS、查看硬體裝置、管理運行開機部分等等，這是一種典型的作法，彈性也很大，從Flash內載入OS到從網路載入OS皆可完成，也就是說上電後最先讀取的是「主要載入程式」，在由此項管理轉向各個OS的Booting位置又或者是直接轉項OS的開機位置，下圖為Boot-Loader的常見之功能用解釋。

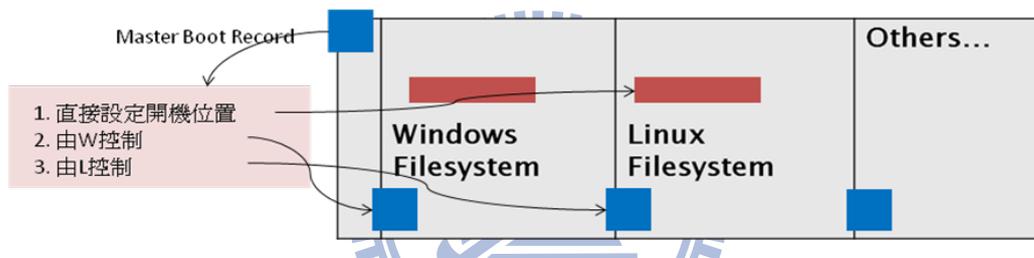


圖 4- 2 載入 U-boot 的流程

Flash內的資料並不會因為電力消失而遺失，因此是此系統Boot-Loader的存放位置，燒入環境為TI的開發環境平台- CCStudio v3.3，值得一提的是，在此是使用DM365[14]的工具燒入DM368，原因在於目前DM365與DM368分別是前後代的產品，DM368僅有部分功能加強(如1080p H. 264影像編碼)，故Boot-Loader之程式及其工具使用DM365即可。

```
Erasing the NOR Flash
Erased through 0x2040000
Erase Completed
Writing the NOR Flash
NOR Write OK through 0x2020010.
Writing the NOR Flash
NOR Write OK through 0x2028000.
NOR Write OK through 0x2030000.
NOR Write OK through 0x2038000.
NOR Write OK through 0x203ACB8.
                NOR boot preparation was successful!
```

圖4- 3 DM368燒入Boot-Loader過程畫面

當 Boot-Loader 燒入成功後，啟動的部分便是 OS 了，在 OS 下控制的就是硬體的許多控制，藉由這些控制階段，便可以達到開發者所希望的動作。

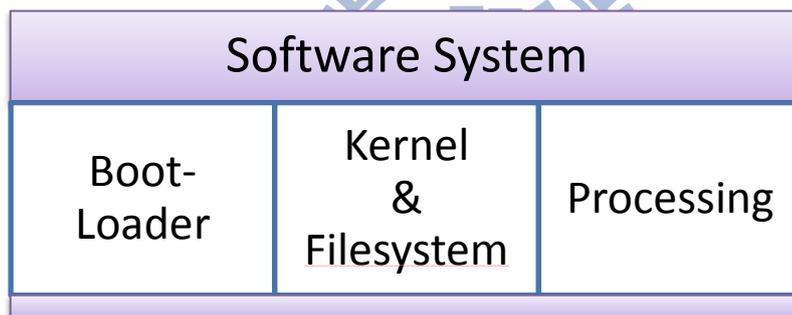


圖 4- 4 掛載 OS 系統之軟體層架構

開機後Boot-Loader把Booting位置指向Kernel，接著 Kernel 啟動周邊設備並且連結上Filesystem即完成Booting動作，後續的處理就屬於「應用軟體層面」，下圖表示Booting後並且掛載上Filesystem後，並且執行程式的畫面

```
# mount -t nfs -o nolock -o tcp 192.168.1.122:/home/kasen/workdir/filesys /mnt
mount: 192.168.1.122:/home/kasen/workdir/filesys already mounted or /mnt busy
mount: according to mtab, 192.168.1.122:/home/kasen/workdir/filesys is already m
ounted on /mnt
# cd /mnt/
# ls
hello          welcome_nfs    welcome_nfs.txt
# cd hello/
# ls
embedded_hello_hello    hello.c        hello.c~
# ./embedded_hello
H^L^L^L^0!
UVLab
#
```

圖4- 5 執行了「Hello World」程式

## 4.2 系統中使用硬體演算法

要使用一支堪用的影像處理應用程式，在背後有許多默默在運作的部分，故在組合成可用的應用層面之前，在此節當中介紹韌體層的框架，以及如何使用 DM368 硬體演算法如何應用於後端軟體應用層。

### 4.2.1 影像串流

在此節中，在系統中的區塊圖、子程式及記憶體緩衝等等的運作皆以圖文的方式介紹各流程。

從影像擷取設備當中取出未壓縮的影像資料，在此 Task 中的影像到編碼 Task 輸入端所有的資料，都統稱叫做「Capture Stream」，也就是影像捕捉串流，處理尚未壓縮過的影像資料；在編碼 Task 中的資料到串流 Task 端所有的資料，這段流程內的影像訊號都統稱叫做「Encode Stream」，也就是經過各個影像格式的編碼後，處理已經壓縮過的影像資料。

正常的狀況下，此套韌體最多可支援 4 組「Capture Stream」及「Encode Stream」，也意味著，在微處理器運算能力負載內，並且透過適當的配置，相當等於四台獨立的影像處理系統，也同時在「Capture Stream」及「Encode Stream」的關係中，不必一對一的對應關係，不過在此系統只使用了一組「Capture Stream」及「Encode Stream」的串流。

影像資訊從 IPIPE(Image PIPE) [15] 進入系統後，隨著 Stream 至第一層，做了 VS(Video Stabilization) [15]、H3A(Auto White Balance、Auto Exposure、Auto Focus) [15] 及 LDC(Lens Distortion Correction) [15] 這些使影像穩定的前處理之 Task 後，進入下一層實行 OSD(On Screen Display) 功能，作為標記、顯示各種資料之用，接著最後使用所需要的影像編碼格式，作為傳輸或是儲存資料之壓縮處理。下圖為所述之流程使用圖表示之

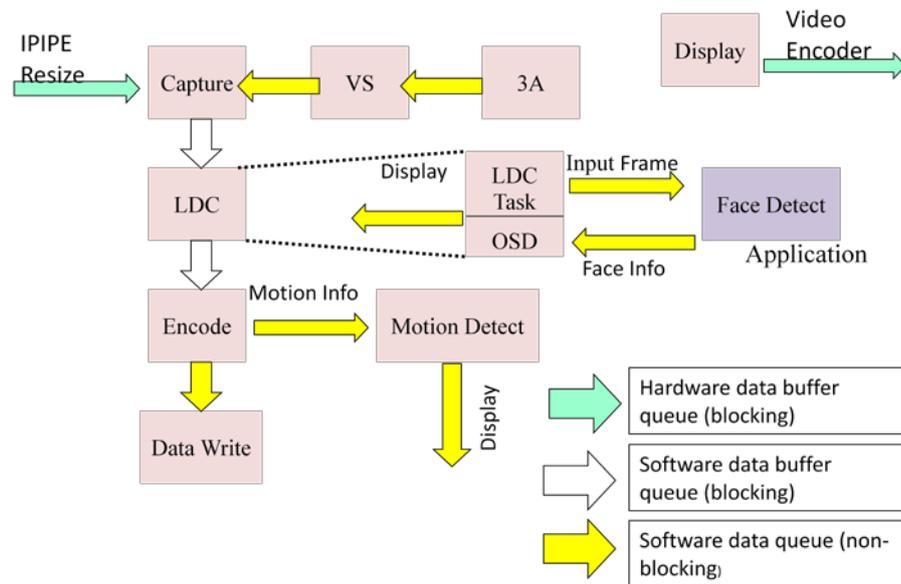


圖 4- 6 影像處理之系統運作之方塊流程圖

#### 4.2.2 資料訊號流程

上節中介紹的是整個運作架構的方塊流程，在嵌入式系統中也占有一席之地  
的記憶體資料管理流程部分在此節當中說明之，並且以 720P + QVGA 畫質影像作  
為範例說明。

在流程當中攝影機透過 ISIF 介面傳送資料至 IPIPE 並且作 Resize 的動作，  
並且在 Resize 後轉存為 YUV420 格式存放於 RAM 中，在此項動作後的影像格式才  
是此實作中的 YUV420 影像格式；再轉換為 YUV420 影像格式後，進入至上節所提  
及之 Capture Task 或 VS Task 當中，並且啟用 H3A 運算；在此階段完成後，接  
著進入至 LDC Task 中，以 4.2.1 當中的圖為例，將會把資料傳至記憶體緩衝區，  
在此記憶體緩衝區內的資料，其實就是最後播送的原始資訊，也同時在此階段中，  
使用此記憶體緩衝區實現本論文所實現的一個應用程式，不過處理這些數位訊號  
前，也可以啟用 DM368 所擁有的去除雜訊之硬體演算法。下圖為上述文字的流程  
內容，並且與圖 4-6 做呼應

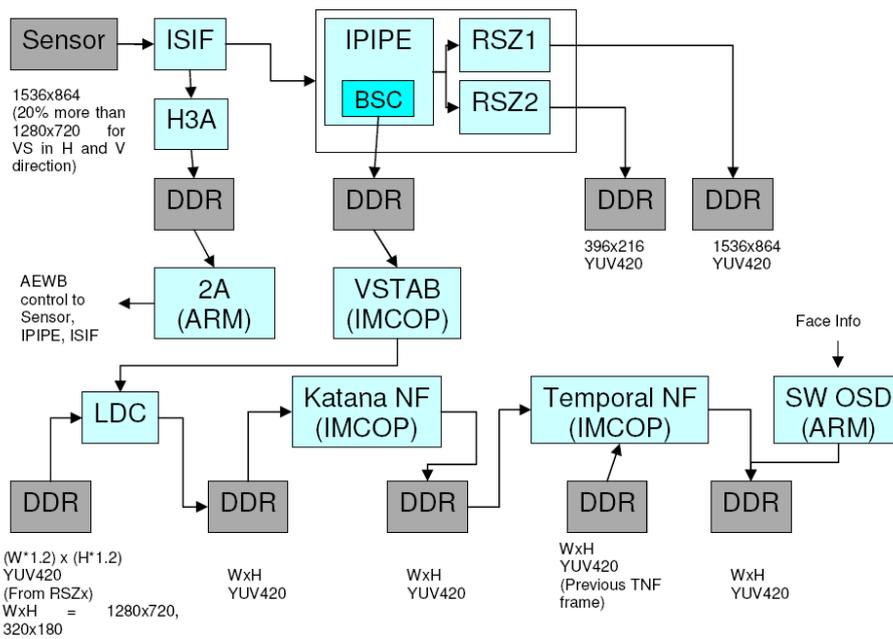


圖 4- 7 720P + QVGA 畫質影像資料流程訊號圖[15]

### 4.3 系統之應用

於此介紹如何將 DM368 以及韌體作結合，配合上軟體來完成「電子牆」應用，電子牆的概念為減少人力疏忽之判斷而延伸出來的智慧型判斷系統，於攝影機上取任一線段當作警戒牆，當物體通過此牆時會進行方向性偵測，也就是說可以當作一個影像式的門禁系統。

#### 4.3.1 概念簡介

於此節介紹如何 Corss-Compiler 成為 DM368 可執行的程式，由於 Linux 系統並不是直覺式的圖形介面，並且為了編譯過程更加簡便，使用了自動連結各個.c 及.h 檔的方式，使每次編譯過程類似在 Windows 下的平台般便利，使每個撰寫應用程式的使用者能夠更快上手。

於前節中提到訊號運作的方式，以及利用韌體啟用硬體運作處理影像，在這節當中，將會運用所提及的硬體應用，再配合現今影像處理應用所用的一些演算法，完成所需要的應用。在此介紹的應用為「電子牆」的應用，因 DM368 的運算速度並不快，故在影像輸入的前端 Resize 至 160x128 pixel 的影像大小，在此之後皆使用 Resize 後的影像作為運算依據，並以原始影像大小播放至顯示器。下圖為「電子牆」的應用之流程圖。

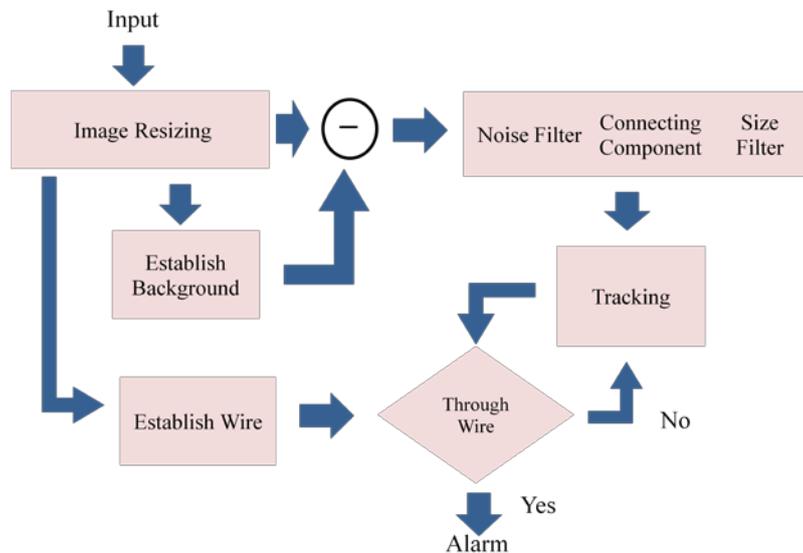


圖 4- 8 「電子牆」之流程

### 4.3.2 便利編譯與執行

在撰寫程式後，需要轉譯成微處理器所支援的機器碼，每種微處理器所支援的格式幾乎都不同，在轉譯的這個過程，稱作是 Corss-Compiler；首先「Hello World」這個程式只需要自己撰寫一個.c 檔即可編譯，但是對於此章當中所提及的應用，並不是少數幾個.c 檔以及.h 檔即可完成的，在撰寫程式過程中，不可能每次編譯時都輸入一大串的.c 及.h 檔，故有一個機制為：編譯時，執行此檔案，由此檔案幫助使用者切換不同的資料夾，連結所有的.c 及.h 檔，此概念的方式如下圖所示

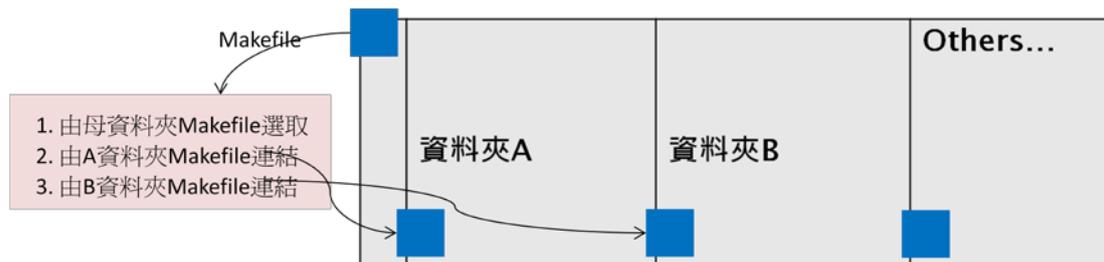


圖 4- 9 Makefile 的概念圖

初步概念建立後，還有周邊的問題，例如：Corss-Compiler 工具位於何處、母資料夾位於何處、Corss-Compiler 後要放置於何處等等諸如此類的周邊輸入/輸出的問題還存在著，同樣透過 Makefile 的機制，撰寫成另外一個常用的檔案-Rules.make。下圖為 Rules.make 及 Makefile 的設定格式

```

Rules.make
BASE_HOME := $(SRCBASE)
HOME := $(BASE_HOME)
DVSDK_BASE_DIR := $(BASE_HOME)/dvSDK_2_10_01_18

TARGET_FS := $(BASE_HOME)/target/filesys
KERNELDIR := $(BASE_HOME)/ti-davinci
BASE_DIR := $(BASE_HOME)/av_capture/build
TFTP_HOME := /var/lib/tftpboot

SIMPLEWIDGET := $(BASE_HOME)/av_capture/application
SIP_LOCAL := $(BASE_HOME)/SIP_lib/SIP_local
SIP_LIB := $(BASE_HOME)/SIP_lib
FREETYPE := $(MVTOOL_DIR)/target/usr/include/freetype2

#SYSTEM := DVR
#SYSTEM := EVM
SYSTEM := IPNC

PLATFORMx := ECCK

#IMG_REVERSE:=y
IMG_REVERSE:=n

CONFIG :=

include $(DVSDK_BASE_DIR)/Rules.make

```

圖 4- 10 Rules.make 的設定樣式[13]

```

Makefile
include $(BASE_DIR)/COMMON_HEADER.MK

exe:
    make depend
    make libs
    make -fMAKEFILE.MK -C$(BASE_DIR)/../application/test exe
ifeq ($(PLATFORMx), EUCDK)
    make -fMAKEFILE.MK -C$(BASE_DIR)/../application/ipnc/multimedia/av_server exe
endif
ifeq ($(PLATFORMx), ECCK)
    make -fMAKEFILE.MK -C$(BASE_DIR)/../application/ipnc/multimedia/av_server_ECCK exe
endif
make install

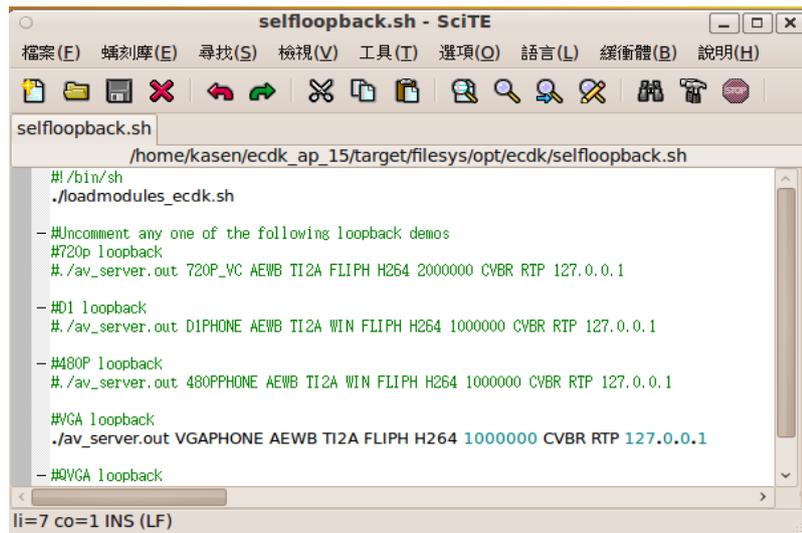
libs:
    make -fMAKEFILE.MK -C$(BASE_DIR)/../framework $(MAKE_TARGET)
    make -fMAKEFILE.MK -C$(BASE_DIR)/../application/test $(MAKE_TARGET)
    make -fMAKEFILE.MK -C$(BASE_DIR)/../application/ipnc $(MAKE_TARGET)

all:
    make clean
ifeq ($(SYSTEM), IPNC)
ifeq ($(IMGS_ID), IMG5_OMNIVISION_OV9712_5MP)
    ln -s $(BASE_DIR)/../framework/alg/src/aewb_ti/include/ae_ti_h_OV $(BASE_DIR)/../framework
    ln -s $(BASE_DIR)/../framework/alg/src/aewb_ti/lib/libti_2a_a_OV $(BASE_DIR)/../framework
    ln -s $(BASE_DIR)/../framework/alg/src/alg_aewb_c_OV $(BASE_DIR)/../framework/alg/src/a
    ln -s $(BASE_DIR)/../framework/alg/src/alg_aewb_control_c_OV $(BASE_DIR)/../framework/i
    ln -s $(BASE_DIR)/../framework/alg/src/ti_gbce/alg_ti_gbce_c_OV $(BASE_DIR)/../framework/
else
    ln -s $(BASE_DIR)/../framework/alg/src/aewb_ti/include/ae_ti_h_AP $(BASE_DIR)/../framework
    ln -s $(BASE_DIR)/../framework/alg/src/aewb_ti/lib/libti_2a_a_AP $(BASE_DIR)/../framework
    ln -s $(BASE_DIR)/../framework/alg/src/alg_aewb_c_AP $(BASE_DIR)/../framework/alg/src/al
    ln -s $(BASE_DIR)/../framework/alg/src/alg_aewb_control_c_AP $(BASE_DIR)/../framework/z
    ln -s $(BASE_DIR)/../framework/alg/src/ti_gbce/alg_ti_gbce_c_AP $(BASE_DIR)/../framework/

```

圖 4- 11 連結各個資料夾的 Makefile 格式[13]

透過此類型的 Corss-Compiler 方式，便可以化繁為簡的轉譯了；接著會碰到需要執行 A 程式後 B 程式接著執行，並且 B 程式同時需要起始的可調整的參數設定時，就需要 Shell-script 的機制，此機制也是把所需要的 Shell command 寫入於一個檔案裡，當執行該檔案時，會依照使用者所設定之順序執行，並且可以使起始參數連帶啟用。下圖為 Shell-script 的範例



```
selfloopback.sh
/home/kasen/ecdk_ap_15/target/filesys/opt/ecdk/selfloopback.sh

#!/bin/sh
./loadmodules_ecdk.sh

-#Uncomment any one of the following loopback demos
#720p loopback
#./av_server.out 720P_VC AEWB TI2A FLIPH H264 2000000 CVBR RTP 127.0.0.1

-#D1 loopback
#./av_server.out D1PHONE AEWB TI2A WIN FLIPH H264 1000000 CVBR RTP 127.0.0.1

-#480P loopback
#./av_server.out 480PPHONE AEWB TI2A WIN FLIPH H264 1000000 CVBR RTP 127.0.0.1

#VGA loopback
./av_server.out VGAPHONE AEWB TI2A FLIPH H264 1000000 CVBR RTP 127.0.0.1

-#QVGA loopback
```

圖 4- 12 Shell-script[13]的使用方式

首先執行「loadmodules\_ecdk.sh」的 driver 檔，後執行「av\_server.out」的應用程式，並加入起始參數

### 4.3.3 取得前景

在完成了韌體層的 Corss-Compiler 後，看到的是 VGA 及 160x128 pixel 之影像，其中 160x128 pixel 之影像為經過 Resize 後的影像，如下圖所示為韌體層的原始畫面



圖 4- 13 韌體層執行後的畫面

主要畫面為 VGA 畫質，右下角的小圖為 Resize 後的影像

如同在前所提及，所處理的影像皆為 Resize 後的畫面，最主要理由為 DM368 運算速度不如個人電腦。在機器視覺找出目標物的方法大致分為兩類：直接認定為某物體以及建立背景再分割出前景的兩種方式。

主要目的為找出前景並且標示出來，在此使用的為較為廣泛的建立背景之方法，原因在於此應用程式常使於監視系統中，畫面中的任何物體變化並不劇烈，GMM (Gaussian Mixture Model) [16][17]為目前成熟的統計方法之一；所熟知的是：不論是機器或是動物，在學習的過程都可以視為是「統計歸納」的方式，在統計時必須要有先前條件，必須先得知物體為何種歸類，例如在天空中飛翔的歸類為鳥類，所擁有的特徵為在天空中飛翔的生物；又有一天看到了飛機在天空飛，這時被告知為飛機，所擁有的特徵為在天空中飛翔的物體；對於機器來說若是沒告知此項物體為何物，恐怕是永遠無法得知的。也因此利用此種特性產生了以 GMM 建立背景的法則，利用長時間出現的畫素，使機器判斷為背景進而可以裁減出前景，再由這些前景使用影像處理上的一些方式，使機器認知為某項物體。

GMM 從中心極限定理可以看出 Gaussian 分佈（也叫做常態分佈），雖然可以用其他的分佈來任意架構出某某 Mixture Model，但 Gaussian 分佈在計算時也有很好的性質，故 GMM 近年最為流行。下圖為 GMM 的一個概念圖

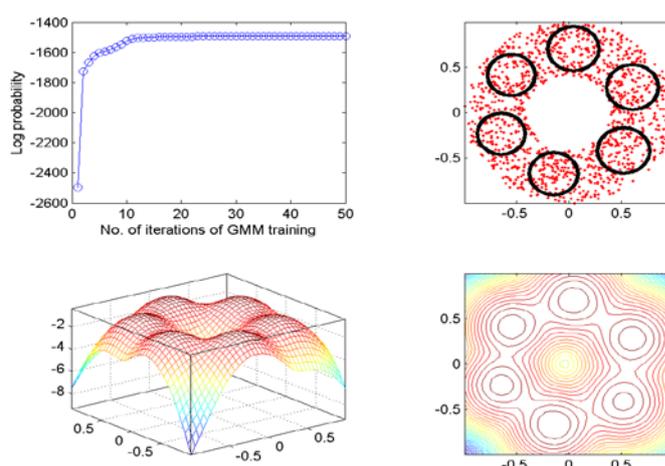


圖 4- 14 GMM 的示意圖

上圖為 GMM 的估測模型圖，外圍的虛線輪廓線為創立出來的 GMM 模型

由於以 GMM 建立背景收斂時間可以使用參數調整，故在此並不討論，在此系統中 GMM 使用的為計算 Variance 及 Mean 值堆疊出預測的模型，。下圖為以列為單位的 GMM 背景圖

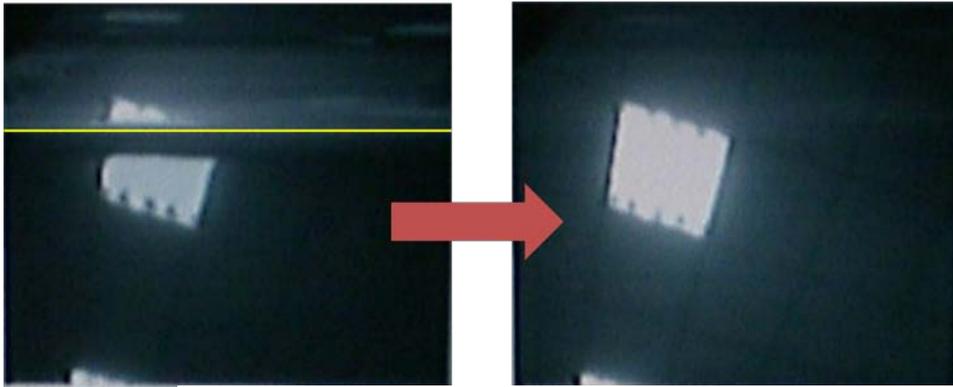


圖 4- 15 使用 GMM 概念建立背景

上圖左為尚未收斂完成的 GMM 背景圖，右圖為 GMM 收斂後的背景

在建立完背景後，利用當下的畫面減去背景後，所得到的就是前景，此時的前景並不是穩定的狀況，且不利於人眼的觀測，所使用的影像處理技巧為二值化、Connecting Component、Erosion 及 Dilation 此項動作會濾除些許雜訊，並且使前景利於人眼觀測。下圖為使用上述方法後得到的前景

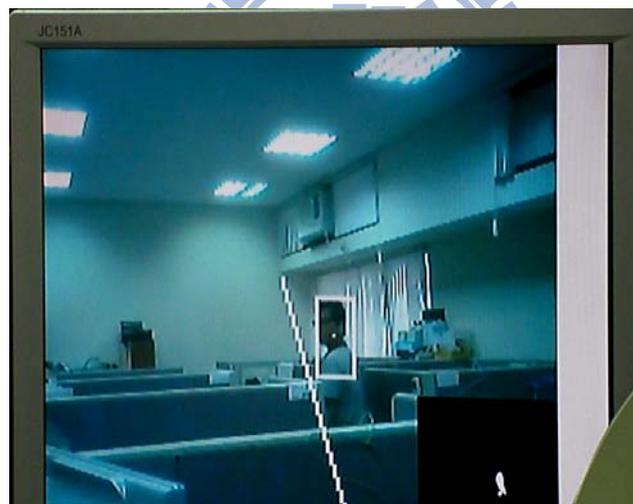


圖 4- 16 建立背景後找到的前景

上圖為建立背景後找到的前景，並透過上述方式濾除雜訊所得到的右下角之二值化前景，並且標示於主畫面

在此節中把前景找出，標示的過程的線，皆是使用硬體運算之 OSD 功能，於撥放畫面的的 RAM 記憶體緩衝區中直接給定值，並不用再次的存取此畫面作 OSD 的功能，對於此系統實作有一定的幫助。

#### 4.3.4 電子牆

在前節提到取出了前景之後，便加入了選定前景的動作[18-21]，這些動作皆參考影像處理較為常見的演算法，在確定了前景後，便可以進行物體的追蹤，在機器視覺中，追蹤的方式也有幾種，在此是使用特徵點的追蹤，選用的特徵點即為物體的中心，並不是太困難，但是由於前景通常由為明顯，故簡易的方式便可以達成希望的效果，追蹤的方式為：每張有前景的畫面皆找出前景之特徵點，如果此兩點小於一定的值，便認定前後兩章畫面的物體為同一項物品，如下圖所示的表示，即為此系統的追蹤

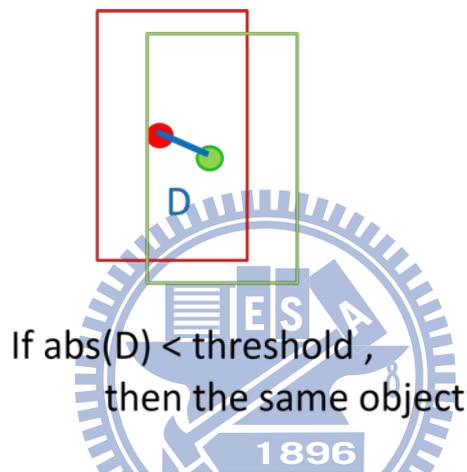


圖 4- 17 此系統的追蹤機制

在確認每個物件的歸屬後，接著為創造出具有「方向性」的電子牆，方向性的意思便是通過警戒牆時可朝特定的方向前進，反之便發出警報，如下圖所示意方向性的概念

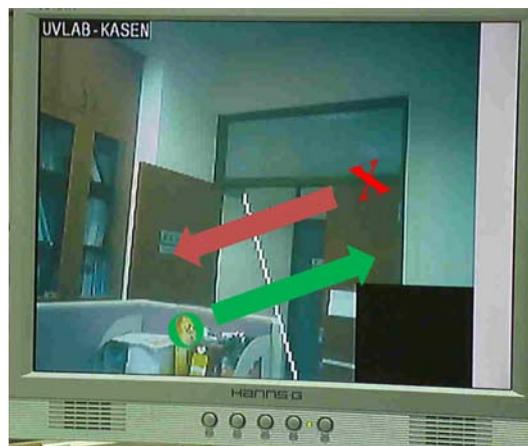


圖 4- 18 具有「方向性」的電子牆示意圖

在攝影機畫面畫上任一線段作為電子牆，由於電子牆為使用者所自行設定，便可以得知電子牆之方程式，此系統中的電子牆為直線，故方程式為二元一次方程式，令此方程式為  $f(x, y) = Ax + By + C$ ，在此線段上的任意點帶入此方程式皆為零；反之不在此線段上的點帶入此方程式必不等於零；也就是說可以利用此項特性，當特徵點靠近警戒牆時啟動此機制：設定特定方向的牆，也就是說把特徵點帶入方程式後，當  $f(x, y)$  由「小於零轉換成大於零」時，便得知物體由警戒線的「左邊跨越至右邊」。下圖為上述之方向性的具體表示法

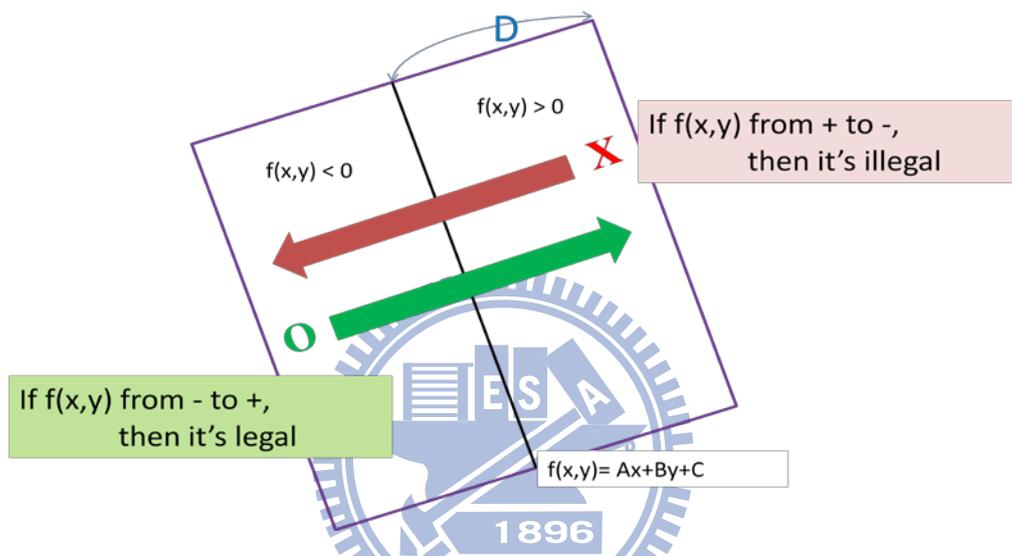


圖 4- 19 具有「方向性」的電子牆表現方式

上圖為電子牆之示意圖，由右至左是為法的；特徵點帶入後由「大於零轉換成小於零」是不被允許的

在透過了影像 Resize、建立 GMM 背景抓取前景、過濾雜訊、追蹤前景、設立警戒線、使電子牆具有方向性等等的行為，已經達成「電子虛擬牆」的訴求，並且透過此應用驗證了「系統整合運算」之可行性，在完成此項應用時，也同時架立了方便使用者呼叫硬體運作的 API。

## 第五章 實驗結果

在此章當中，硬體部分將此篇所提出以 DM368 為微處理器之系統相關規格做比較；軟體部分顯示 DM368 之「影像處理子系統」內的人臉偵測，並且以同樣子系統之 OSD 標示出目標物；應用軟體則使用第四章所提及的「電子牆」來驗證系統以及概念可行性。

### 5.1 硬體系統規格比較

在此節當中，將 TI 公司所推出的 TMDXEVM368 平台[22]以及 Leopard 公司所推出的 EUCDK-DM368 平台，與此篇論文提出的整合平台做各功能上的比較。

表 6 TMDXEVM368、EUCDK 及本文系統的相異處之比較

	TMDXEVM368	EUCDK	UVLAB
MCU	TMS320DM368	TMS320DM368	TMS320DM368
Video Input	Analogy、Digital	CMOS sensor only	Analogy、Digital
Video Output	Analogy、Digital	Analogy、Digital	Analogy、Digital
Display	Build-in	External Display	External Display
Audio Processing	Yes	Yes	NO
Size (cm)	20 x 20	15 x 5	12 x 7.5
Device Combinations	At least 3 items	At least 2 items	Single device

如上表所示，TMDXEVM368 功能最為齊全，但體積過大，並且需要額外裝置作組合才能夠運作；EUCDK 部分也是需要把主要運算板以及通訊板做組合的動作才能使系統運作，也並不支援類比影像的輸入。

對於此篇提出的 DM368 系統方案中，整合了影像處理裝置所需的硬體及其周邊，使其只需一塊電路板便能完成系統運作，並且體積並沒有因此而變大許多。下列三張圖為上述三種以 DM368 為微處理器的系統之外觀圖

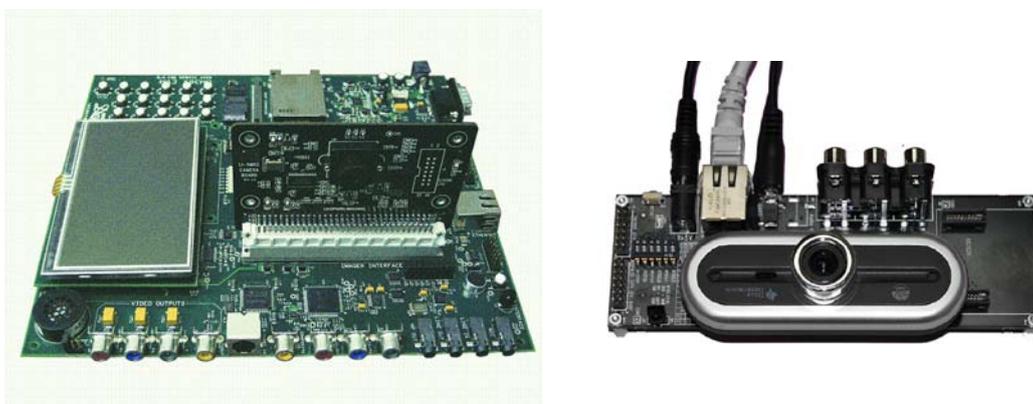


圖 5- 1 為 TI 及 Leopard 所推出之系統

上圖左為 TI 公司所推出的 TMDXEVM368 平台外觀圖，外接之影像感測器為直立式，占據了許多體積；上圖右為 Leopard 公司所推出的 EUCDK 平台外觀圖，訊號處理電路與通訊電路為組合式的，實際使用上易鬆脫造成使用上的不便，且並沒有提供類比影像輸入的功能

下圖為此論文提出之硬體系統，並且各個部分的晶片及外部連接裝置之成品圖

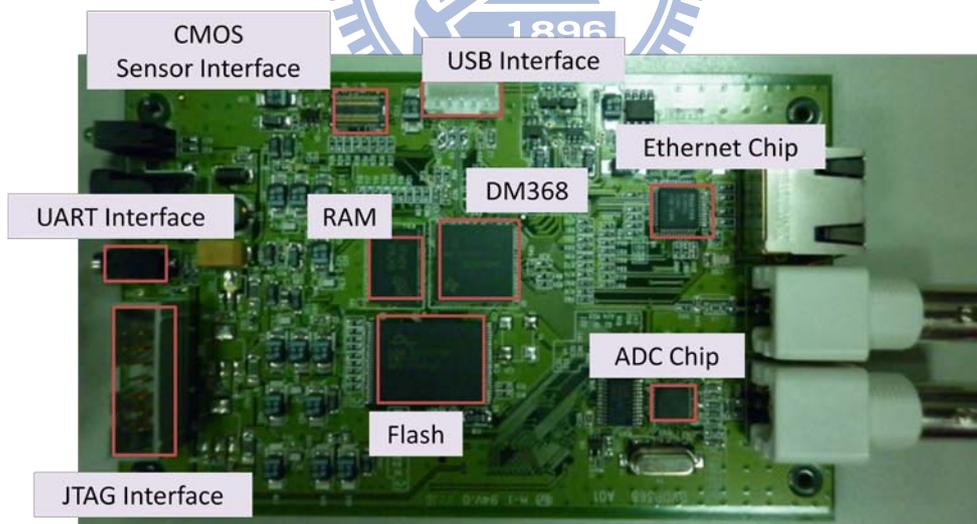


圖 5- 2 此論文提出之連接範例之硬體系統

上圖為此論文提出之連接範例之硬體系統，並且標示出各個位置之圖片，對於影像處理平台來說體積不至於太大，最大不同處為一體成型之設計，並以影像處理的角度做設計

## 5.2 軟體成果

韌體部分也針對影像處理做了些許修改，使其更容易上手與改進。以下介紹的部分為整合韌體的實驗結果部分。

- 人臉偵測
- 影像縮小
- On Screen Display (OSD)
  - 線
  - 圖
  - 字串

首先是人臉偵測的部分，在第四章流程圖中的 Face Detect[15]為應用在人臉辨識之範例，於韌體層整合出使用者角度的子程式，必能加速應用層面的發展，此節以人臉偵測為例，介紹如何整合成使用者角度之子程式。如 4-1 節所提到的圖中，DVSDK 及 OS 層是尚未修改的原始部分，如想要驅動 DM368 人臉辨識之使用，透過了 DVSDK 層作為畫面中扮演偵測人臉的角色，並且編寫為演算法的子程式，在 4-1 節所提到的圖中，Algorithm Library 層的角色，就是方便於 DVSDK 層的使用，並且可以依據使用者修改較為少變動之參數。最後包裝成為使用者可直接呼叫的子程式，在 4-1 節所提到的圖中是屬於 Wrapper Library，在此層中的呼叫方式依照使用者常用之參數做為調整，如 `int Kasen_FD (W, H, Input, face_start, face_end)`；只需給定系統畫面長、寬、畫面輸入等資料，便會起用人臉偵測之功能，並於各個人臉之起始以及結束位置起用 OSD 畫線之功能。下圖為此流程圖之圖示

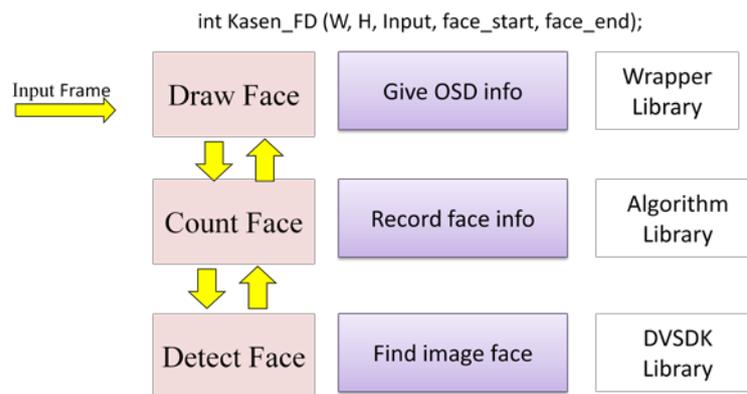


圖 5- 3 程式的包裝方式

上圖為為修改韌體後的包裝方式，由 DVSDK 原始資料庫建立演算法資料庫，最後在包裝成為「Wrapper Library」這類型的子程式



圖 5- 4 為 DM368 之人臉辨識結果

上圖為輸入影像為 720p 結果，並且使用「OSD 劃線」標示出位置的結果展

示

在上列展示中顯示了此論文提出系統當中的人臉偵測及 OSD 畫直線之功能，並且使用 USB 介面做為數位影像輸出之結果。以下展示的輸入、輸出皆為 VGA (640x480) 畫質之結果，並且透過類比(Composite)的方式做為輸出結果。

「影像縮小」及「OSD 貼圖」於第四章 4.3.3 部分以及第四章後半部皆有部分展示，如下圖所示即為整合了許多子韌體程式功能程式化之結果



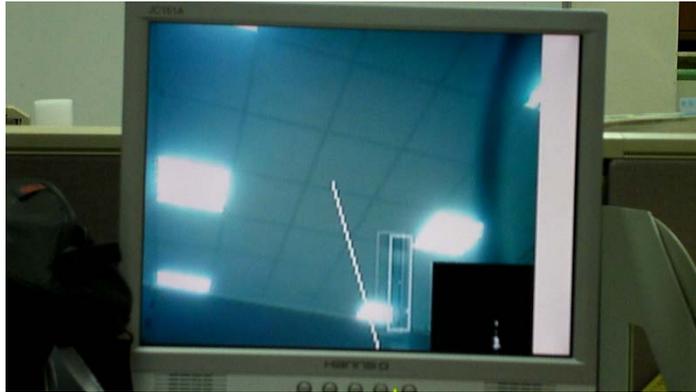
圖 5- 5 韌體之各個程式結果

上圖為「影像縮小」、「OSD 貼圖」、「OSD 劃線」及「OSD 字串」之功能顯示於同一個視窗內，左上角的黑底白字之字串即為「OSD 字串」；右下角黑色矩形即為「OSD 貼圖」之結果，畫面中的斜線為「OSD 劃線」之結果

在完成了許多韌體層的子程式後，接著以下即是「電子牆」藉由此篇論文所提出的系統所呈現的結果，首先是簡易的測試電子牆之功效，如下列圖所示



圖 5- 6 使用 GMM 建立背景後的圖



當有前景進入時，  
使用白色的線做為標示



當前景為不被允許  
的方向通過電子牆時，  
便使用黑色的框表示，  
並持續追蹤

圖 5- 7 驗證電子牆之可行性的實驗結果

「電子牆」的可行性已被確認，持續追蹤物體的機制由以下的結果來驗證，並且加入了遮蔽物做為展示



圖 5- 8 使用 GMM 建立背景後的圖



當有前景進入時，  
使用白色的線做為標示，  
此物體為遮蔽物



當第二個前景出現時，  
也使用白色的框表示，  
並持續追蹤



當有目標物侵入電子牆，  
使用黑色的框表示



闖入電子牆的目標物，  
使其持續追蹤，  
並不影響正常的前景



即使闖入電子牆的目標物  
又回復到被允許的位置，  
依然持續追蹤



即使兩物體又分開，  
依然分別持續追蹤

圖 5- 9 電子牆之應用結果

上圖的例子中，驗證了持續追蹤的能力，雖場景不致太複雜，但同時證明此概念的可行性，由額外的影片證明，是可達成Real-Time的目標的。同時藉由上述的幾個例子，驗證了「電子牆」的可行性，並且做為追蹤達到良好的機器智慧效果。

## 第六章 結論與未來展望

在此章中總結了此論文提出的方式組合出的系統，包含了硬體以及軟體部分的結論，最後點出在這個系統架構下，並未提出看法及應用。

### 6.1 結論

在此節中分別以硬體及軟體部分做出總結歸納。硬體系統方面創造了屬於實驗室的嵌入式影像處理平台、此平台可以使軟硬體共同運算影像處理之演算法、此平台可以透過開發使用者自行加入所希望增加的影像之演算法，並且可以自由的更改影像之輸入；軟體系統部分實現了易於上手的編譯環境、實現了硬體運算之程式、實現了影像處理的平台框架、實現了監控系統當中的電子牆之功能以及人臉偵測之系統。

此論文提出的系統中，無論是硬體或是軟體，皆是以應用面的方式做為考量，部分影像演算法藉由 DM368 的影像子系統，並且以達成 Real-Time 為目標；在硬體的組成中，挑選出所需要的應用，在連結了各個 EMIF 的處理、系統方面的通訊處理等等，組合成具有優點、獨一無二的影像處理平台；軟體方面也修改了 ECDK 的軟體架構，使其成為影像演算法開發者易於上手的架構，並且把 DM368 硬體演算法整理成為函式化的結構。下列為條列式的列出此論文的結論

- 硬體系統
  - 屬於實驗室的嵌入式影像處理平台
  - 軟硬體共同運算之平台
  - 可透過使用者自行加入所需的演算法
  - 可自由的更換影像輸入
- 軟體系統
  - 實現了易於上手的編譯環境
  - 實現了硬體運算子程式
  - 實現了影像處理的框架
  - 實現了監控系統之電子牆
  - 實現了人臉偵測系統

## 6.2 未來展望

在硬體的部分，由於只有提供系統所需電力，以及並未考慮到相關應用面，在此列出未來可實現的目標

- 可放置 GPIO 接腳
- 可加上聲音或是生理訊號等等的處理
- 可由觸發電路觸動如喇叭、錄影器材等等的裝置

在軟體的部分，有許多的天馬行空應用程式皆可加入，提出一些看法及應用，使未來能有努力的目標

- 圖形化介面
- 使用 H.264 壓縮影像
- 透過 Ethernet 傳送影像串流
- 於系統中架設 WEB 介面的使用者操控

以上不論硬體或軟體，為此篇論文提出的未來可行之目標，並且一步步的規劃，相信未來的目標會逐漸的被實現，以機器為人類的生活帶來更多的娛樂、教育、方便的事情。

# 参考文献

- [1] Texas Instruments, “Datasheet-TMS320DM368- Digital Media System-on-Chip (DMSoC)”, Texas Instruments, 2010
- [2] Analog Devices, “Datasheet-ADSP-BF561- Blackfin Embedded Symmetric Multiprocessor”, Analog Devices, 2009
- [3] Texas Instruments, “Datasheet-DM3730- Digital Media Processors”, Texas Instruments, 2010
- [4] Texas Instruments, “Datasheet -AM389x- Sitara ARM Microprocessors (MPUs)”, Texas Instruments, 2010
- [5] Texas Instruments, “Datasheet -TMS320C6A816x- Power Estimation Spreadsheet”, Texas Instruments, 2011
- [6] Texas Instruments, “Datasheet -OMAP3530- Applications Processor”, Texas Instruments, 2009
- [7] Texas Instruments, “Datasheet -TMS320DM6437- Digital Media Processor”, Texas Instruments, 2006
- [8] Texas Instruments, Leopard, “Embedded USB Camera Development Kit- User's Guide”, Leopard, 2010
- [9] Micron, Aptina, Leopard, “Datasheet -LI-5M04CM”, Leopard, 2009
- [10] Micron, “Datasheet -MT47H64M16-DDR2 SDRAM”, Micron, 2004
- [11] Spansion, “Datasheet -S29GL128P-MirrorBit® Flash Family”, Spansion, 2007
- [12] Analog Devices, “Datasheet -ADV7180-10-Bit, 4x Oversampling SDTV Video Decoder”, Analog Devices, 2006
- [13] Texas Instruments, Leopard, “Embedded Camera Development Kit- User's Guide”, Leopard, 2010

- [14] Texas Instruments, "Datasheet -TMS320DM365-8- Digital Media System-on-Chip (DMSoC)", Texas Instruments, 2009
- [15] Texas Instruments, "IPNC Reference Design on DM36x AV Server Design Guide", Texas Instruments, 2010
- [16] Stauffer, C.; Grimson, W.E.L., "Adaptive background mixture models for real-time tracking", *IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 1999.*, vol.2, no., pp.-252 Vol. 2, 1999.
- [17] Peng S.; Yanjiang W., "An improved adaptive background modeling algorithm based on Gaussian Mixture Model", *Signal Processing, 2008. ICSP 2008. 9th International Conference*, pp. 1436-1439, 26-29 Oct. 2008.
- [18] G.Jacobs, A. Aeron-Thomas, and A. Astrop, "Estimating global road fatalities," *Australian National University, Transport Research Laboratory, Technical Report TRL 445*, 1999.
- [19] Sheng-Che Hsu, Sheng-Fu Liang, Chin-Teng Lin, "A robust digital image stabilization technique based on inverse triangle method and background detection", *IEEE Trans. Consumer Electron.*, vol. 51, no. 2, pp.335-345, May, 2005
- [20] F. Vella, and A. Castorina, "Digital image stabilization by adaptive block motion vectors filtering", *IEEE Trans. Consum. Electron.*, vol. 48, no.3, pp.796-801, Aug. 2002
- [21] L. Chen and N. Tokuda, "A general stability analysis on regional and national voting schemes against noise –Why is an electoral college more stable than a direct popular election?" , *Artif. Intell.*, 163. No. 1,pp. 47-66, 2005
- [22] Texas Instruments, "TMDXEVM368- User's Guide", Texas Instruments, 2011