

# 國立交通大學

電機學院 電機與控制學程

碩士論文

嵌入式跨平台多工腦機介面應用於智慧型  
居家環境控制

Development of Embedded Cross-Platform  
and Multi-Task BCI for Smart Room Control

研究生：陳德機

指導教授：林進燈 博士

中華民國九十六年八月

嵌入式跨平台多工腦機介面應用於智慧型居家環境控制

Development of Embedded Cross-Platform and Multi-Task BCI for  
Smart Room Control

研究生：陳德機

Student : Te-Chi Chen

指導教授：林進燈 博士

Advisor : Dr. Chin-Teng Lin



A Thesis

Submitted to College of Electrical and Computer Engineering  
National Chiao Tung University  
in partial Fulfillment of the Requirements  
for the Degree of  
Master of Science  
in  
Electrical and Control Engineering  
August 2007

Hsinchu , Taiwan , Republic of China

中華民國 九十六 年 八 月

嵌入式跨平台多工腦機介面應用於智慧型居家環境控制  
Development of Embedded Cross-Platform and Multi-Task BCI for  
Smart Room Control

國立交通大學 電機學院 電機與控制學程碩士班

## 中文摘要

在以往的生理訊號監控系統大多只做到訊號記錄部分，再藉由電腦進行離線訊號處理，而缺乏即時分析的能力。因此，本論文提出一套具有即時分析能力的腦機介面系統以改善一般生理訊號監控系統的缺點，並結合腦波精神狀態監測與居家環境控制，提供給行動不便的病人、老人一個舒適的居家環境。為了達成智慧型居家環境的控制，本論文開發一套嵌入式跨平台多工排程方法來增加系統於即時處理效能，最大的特色在於這個多工排程可以將一個工作切割成數個子工作來減少運算的時間，而訊息交換的架構具跨平台的能力能讓工作分別在不同作業系統平台的機器上工作。最後，這個多工排程方法結合即時的生理狀態偵測的演算法，並整合不同平台的环境控制器來控制週遭的環境，進而提升使用者之生活品質。

關鍵字：腦機介面，腦波訊號處理，疲勞狀態偵測，嵌入式多工排程，嵌入式系統，居家環境控制

# Development of Embedded Cross-Platform and Multi-Task BCI for Smart Room Control

Student: Te-Chi Chen

Advisor: Dr. Chin-Teng Lin

Degree Program of Electrical and Computer Engineering

National Chiao Tung University

## ABSTRACT

The most former biomedical signal supervisory systems only realize signal recording, and then these signals are processed by a powerful computer for off-line analysis. Hence, the conventional system lacks the capability of real-time computation and analysis. Due to the above drawbacks, this thesis proposes an embedded-based brain computer interface (BCI) system to overcome these problems. This system combines the EEG monitor and smart room control to provide a comfortable living environment for the handicapped patients and elderly people. To achieve wisdom-domestic environmental control, this thesis proposes multi-task and cross-platform schemes to achieve real-time computation. The multi-task processing approach divides the task into many sub-tasks to significantly reduce the computation time. The data-exchange scheme has the capability of crossing the platforms and allows tasks to operate at different machines with distinct operating systems. At last, the simulation and demo results show that the proposed methods can achieve online biomedical signal analysis and can be integrated into the smart room controller among different platforms for better life.

**KEYWORD :** brain computer interface, brain signal processing, drowsiness detection, embedded multi-task scheduling, embedded system, smart room control

## 誌 謝

本論文的完成，首先要感謝指導教授林進燈博士在過去三年研究期間提供豐富的實驗資源、研究環境並且從旁指導協助，讓我能順利的完成研究，以及本校資工系范倫達教授和鍾仁峰學長從旁指導協助，讓學生受益良多。此外，學生要感謝口試委員們在百忙中抽空參與學生的口試，並提供許多寶貴的意見，使得本論文更加完善。

其次，我要感謝腦科學研究中心的全體成員，沒有他們也就沒有我個人的成就。特別感謝的柯立偉學長給予我在各方面的指導，無論是研究上疑難的解答、研究方法、寫作方式以及經驗分享等惠我良多，很慶幸在研究生涯上能遇到如此的學長。另外，要感謝林真如、蔡依伶學妹過去一年在實務上以及實驗測試中的協助。

最後，我要想感謝我的妻子許碧雲小姐，在我忙於學業與工作時能細心照顧家庭以及我們的女兒瑋，讓我能專心盡力完成我的學業而無後顧之憂。謹將本論文獻給我的家人以及一路上陪伴我以及幫助過我的所有人，一起分享這份喜悅。



# 目 錄

|                                |           |
|--------------------------------|-----------|
| 中文摘要 .....                     | ii        |
| ABSTRACT .....                 | iii       |
| 誌 謝 .....                      | iv        |
| 目 錄 .....                      | v         |
| 圖目錄 .....                      | vii       |
| 表目錄 .....                      | ix        |
| <b>一、 緒論 .....</b>             | <b>1</b>  |
| 1.1 研究背景 .....                 | 1         |
| 1.2 相關研究 .....                 | 2         |
| 1.3 研究動機 .....                 | 3         |
| 1.4 論文架構 .....                 | 4         |
| <b>二、 腦電位訊號感測系統 .....</b>      | <b>5</b>  |
| 2.1 簡介 .....                   | 5         |
| 2.2 腦電位訊號簡介 .....              | 6         |
| 2.3 腦電位訊號量測電路 .....            | 8         |
| 2.3.1 腦電位訊號量測電路設計 .....        | 8         |
| 2.3.2 無線訊號傳輸模組 .....           | 9         |
| 2.3.3 CPLD 與 VHDL 硬體描述語言 ..... | 10        |
| 2.3.4 類比/數位訊號轉換與無線傳輸模組控制 ..... | 11        |
| <b>三、 跨平台訊息交換系統 .....</b>      | <b>15</b> |
| 3.1 多工排程設計 .....               | 16        |
| 3.1.1 工作切割與基本排程架構 .....        | 16        |
| 3.1.2 多重處理程序之溝通機制 .....        | 17        |
| 3.2 訊息交換平台設計 .....             | 19        |
| 3.2.1 訊息交換平台簡介 .....           | 19        |
| 3.2.2 訊息交換平台之系統架構 .....        | 21        |
| 3.2.3 訊息交換平台應用 .....           | 23        |
| 4.1 即時生理訊號處理 .....             | 27        |
| 4.2 精神狀態偵測 .....               | 29        |
| 4.3 優質生活環境控制平台 .....           | 30        |
| 4.3.1 居家環境平台簡介 .....           | 31        |
| 4.3.2 主控制器 .....               | 31        |

|                                |           |
|--------------------------------|-----------|
| 4.3.3 環境裝置控制器 .....            | 32        |
| 4.3.4 網路設備 .....               | 32        |
| 4.3.5 溝通機制 .....               | 33        |
| <b>五、 測試結果與討論 .....</b>        | <b>36</b> |
| 5.1 系統測試結果 .....               | 36        |
| 5.2 多工排程效能比較 .....             | 37        |
| 5.3 多重服務測試 .....               | 41        |
| 5.4 系統整合與討論 .....              | 42        |
| <b>六、 結論與未來展望 .....</b>        | <b>43</b> |
| <b>參考文獻 .....</b>              | <b>45</b> |
| <b>附錄一、 縮寫對照(按字母排序) .....</b>  | <b>48</b> |
| <b>附錄二、 簡介 OMAP 5912 .....</b> | <b>51</b> |
| B.1 嵌入式硬體開發環境 .....            | 51        |
| B.2 嵌入式軟體開發環境 .....            | 53        |
| B.3 雙核心處理器之內部溝通機制 .....        | 57        |



# 圖目錄

|  |    |
|--|----|
| 圖 2.1、腦電位訊號感測與無線傳輸系統架構圖 .....                          | 5  |
| 圖 2.2、10-20 系統電極擺設位置圖 .....                            | 7  |
| 圖 2.3、腦波量測電路方塊圖 .....                                  | 8  |
| 圖 2.4、無線訊號傳輸架構圖 .....                                  | 9  |
| 圖 2.5、FLEX 系列發展版 .....                                 | 10 |
| 圖 2.6、Max+plus II 軟體介面 .....                           | 11 |
| 圖 2.7、無線訊號傳輸架構圖 .....                                  | 11 |
| 圖 2.8、AD570 內部架構圖 .....                                | 12 |
| 圖 2.9、AD570 控制時序圖 .....                                | 12 |
| 圖 2.10、RF3105 控制模組架構圖 .....                            | 13 |
| 圖 2.11、無線訊號接收程式流程圖 .....                               | 14 |
| 圖 3.1、系統流程架構 .....                                     | 15 |
| 圖 3.2、嵌入式系統排程示意圖 .....                                 | 17 |
| 圖 3.3、Event-Driven Publish/Subscribe Interaction ..... | 19 |
| 圖 3.4、Demand-Driven Request/Reply Interaction .....    | 20 |
| 圖 3.5、Broadcast Request/Reply Interaction .....        | 20 |
| 圖 3.6、Embedded 訊息交換平台系統架構 .....                        | 21 |
| 圖 3.7、Data Exchange 下的 System 分配 .....                 | 24 |
| 圖 3.8、發送與接收關係 .....                                    | 24 |
| 圖 4.1、腦電位訊號處理流程圖 .....                                 | 27 |
| 圖 4.2、訊號之前處理 .....                                     | 29 |
| 圖 4.3、精神狀態改變時腦電位的變化 .....                              | 29 |
| 圖 4.4、Smart Room 系統架構圖 .....                           | 30 |
| 圖 4.5、環境控制器架構 .....                                    | 32 |
| 圖 4.6、Smart Room Controller 先啟動時的通訊模式 .....            | 33 |
| 圖 4.7、Control Point 先啟動時的通訊模式 .....                    | 34 |
| 圖 4.8、命令的通訊模式 .....                                    | 35 |
| 圖 5.1、系統實體圖 .....                                      | 36 |
| 圖 5.2、Single task 程序流程示意圖 .....                        | 38 |
| 圖 5.3、Multi-task 程序流程示意圖 .....                         | 38 |
| 圖 5.4、Single-task 的工作流程 .....                          | 39 |
| 圖 5.5、Multi-task 的工作流程 .....                           | 39 |
| 圖 5.6、Single Server 與 Multi-Server .....               | 41 |
| 圖 B.1、OMAP 5912 OSK 外觀 .....                           | 51 |
| 圖 B.2、OMAP 5912 OSK 架構圖 .....                          | 53 |
| 圖 B.3、OMAP 雙核心系統架構 .....                               | 54 |



|                                 |    |
|---------------------------------|----|
| 圖 B.4、GCC 編譯流程圖 .....           | 55 |
| 圖 B.5、CCS 程式發展設計流程 .....        | 56 |
| 圖 B.6、CCS 整合發展環境 .....          | 57 |
| 圖 B.7、DSP Gateway 系統架構圖 .....   | 58 |
| 圖 B.8、ARM/DSP Mailbox 說明圖 ..... | 59 |
| 圖 B.9、ARM/DSP 內部傳輸說明圖 .....     | 60 |



## 表目錄

|   |    |
|---|----|
| 表 2.1、腦電波分類表 .....                              | 7  |
| 表 2.2、RF3105 控制模組接腳表 .....                      | 13 |
| 表 2.3、無線封包 .....                                | 14 |
| 表 5.1、使用多工排程與不使用多工排程的效能比較 .....                 | 40 |
| 表 5.2、使用多工排程與不使用多工排程的延遲比較 .....                 | 40 |
| 表 5.3、使用 Multi-Server 與 Single Server 的比較 ..... | 41 |



# 一、緒論

## 1.1 研究背景

早在 1875 年，英國利物浦的生理學教授李察卡通首度利用神經電生理的方法從兔子的腦皮質(Cerebral cortex)表面偵測而得到一種腦電位變化，觀察腦神經細胞的活動，且在動物死亡後消失。後來他又發現刺激動物的身體能使腦電位發生變化，於是他利用這種腦電位變化來研究身體部位與腦皮質區的關係，並探討腦皮質區的功能，成為後來神經診斷學中誘發電位 (evoked potential) 發展的基礎[1]。

在進行腦波相關實驗需要長時間的訊號量測與分析，如此大量且複雜的腦波訊號分析將耗費許多人力與時間，直到近年來電腦運算速度日益進步，以及各種分析處理方法改良後，才足以即時處理量測到的腦波訊號。傳統上的腦波儀器是利用高精密的腦波儀之類的專用儀器來做監測，將其腦波記錄後經過人工方式來分析其生理意義。現今軟硬體發展迅速，雖然用高精密的腦波儀來即時分析腦波訊號的應用已不是個問題，但是高精密的腦波儀，空間大且昂貴，在實用上十分不方便[1]。因此，如何發展一個攜帶方便又具即時生理訊號分析系統是一個很重要的課題。

大腦人機介面(Brain Computer Interface, BCI)[2]俗稱腦機介面是利用腦部訊號，讓使用者可以直接與外界溝通。這項技術的基礎在於一當大腦在進行某一項特定活動的時候，會產生特定的腦波變化，因此透過監測與辨認此特定的腦波變化，來達到利用腦波直接控制儀器或輸出訊號之目的。此種技術最主要的目的可以幫助因神經肌肉損傷而行動受到阻礙的人(如肌肉萎縮、中樞神經系統損傷、重度中風的病人等)，使他們不需依靠周邊神經和肌肉，便能夠使用腦電位訊號，達到與外界溝通、傳達訊息、自主行動，以及自我照顧等目的，一方面不但能夠節省社會成本的支出，減輕病人家庭的負擔，另一方面，又能夠讓病人自我行動，建立病人與外界溝通的橋樑，提昇病人的生活品質。在過去十年裡，研究腦機介面的研究團隊已經從五個以下增加到二、三十個以上，由此可見腦機介面已受世界各國廣泛重視。腦機介面的架構可以簡單劃分為兩個部分，前端為腦波訊號的特徵擷取與辨識，後端為與其他硬體的結合與應用，例如手部義肢，控制應用(利用腦波控制滑鼠)等等[1]。

## 1.2 相關研究

國外有一些實驗室從事大腦人機介面相關的研究，包括北京清華大學高上凱教授實驗室與奧地利 Graz-BCI 實驗室[3]-[5]。在可攜式系統的研究上，高上凱教授實驗室研究藉由視覺刺激並透過數位訊號處理器分析腦電波，用以無線控制家電[6]；Graz-BCI 實驗室發展可攜式的人腦與電腦介面系統[7]；美國賓州州立大學發展一套失神性癲癇的腦波偵測系統[8]。交通大學腦中心研究發現在精神狀態不佳或是疲勞時，大腦電位會產生一些特徵訊號，因此可以利用這些生理特徵訊號進行其精神狀態之估測[9][10]。在相關的研究中，國內一直較多著墨在訊號紀錄的階段，較少應用於即時的分析與預測受試者的行為。另外，在嵌入式系統於大腦人機介面的研究上較為少見，而較多 ECG(electrocardiogram，心電圖)處理採用嵌入式系統的方式[11]-[13]。

近幾年來居家照護的研究非常熱門，許多的研究論文紛紛發表出來[14]-[19]，檢視大部分的論文不外乎分成三大類，第一類主要研究重點著重於生理訊號的擷取的感測元件，如血壓，血氧濃度，心電圖(ECG)，腦波(EEG)...等等。第二類主要研究重點著重於如何傳輸量測生理訊號的結果，例如藍牙，ZigBee 等無線傳輸技術。最後一類的研究重點著重於生理訊號量測後的記錄與監測如 PDA，以及遠端監控。大部分的研究還是停留在非即時處理/分析生理訊號的階段，但對實際上的現況而言生理訊號若不能時時的處理並立即提供適當的協助處理的話對病換的實質幫助是非常小的，例如當監控生理訊號發現病人有異狀需要即時的協助時，以往的研究多只能紀錄，發出異常通知，必須等到專業的人到現場或者緊急送到醫院經醫師重新診斷後才能處理，這樣往往會失去寶貴的時間。因此未來的研究應該著重於能及時處理生理訊號並能結合各項外部的設施來有效的協助需要的人。

腦機介面(BCI)大多著重於病患的研究，而且因為腦波量測與分析的儀器非常龐大，所以使用上非常不方便。正如同腦電波量測與即時分析所遇到的問題，在腦機介面的發展中，如何縮小量測、分析系統的體積以及簡化使用上的困難變成為將大腦人機介面時用於日常生活中所需克服的難題。因此，本論文將以開發一具有即時分析功能的 EEG 分析系統，整合於居家環境控制，應用於實際日常生活中為目標。

### 1.3 研究動機

睡眠的時間約佔去人一生中的 1/4 到 1/3。在早期，人們認為睡眠是一種介於生命與死亡之間的狀態，是一種被動性的活動，只有醒著時人體內的器官及神經系統才會主動參與人體的活動。在生理及醫學上，也把睡眠的時間和醒著的時間當成一個整體，以至於在疾病的研究上，鮮有針對睡眠特殊疾病來作研究。早期人們將睡眠所產生的現象或疾病，歸屬於玄學或哲學，大家對睡眠的瞭解最耳熟能詳的應該就屬佛洛伊德「夢的解析」了，真正把有關睡眠疾病當成科學而作研究的少之又少。這個觀念一直到 1953 年，Aserinsky 和 Kleitman 在科學雜誌(Science)發表的一篇文章後，才改變過來，他們發現人在睡眠當中的腦波，每隔一段時間 (90 分鐘)會產生一段醒波，他們又觀察到在這段期間，眼睛會有快速的移動，所以他們將此段時間定名為「快速動眼期」(rapid-eye movement; REM)，他們發現人的夢都在這個時期發生。

然而睡眠並不是感覺及運動神經的完全休息，而有許多主動性的活動在進行，此為睡眠研究的濫觴。睡眠發生障礙，雖不見得是什麼大病，去常造成身體上、精神上諸多的困擾。國外的流行病學調查：20%至 30%的成年人有睡眠問題，老年人甚至高達 35%[20]。睡眠障礙並非成人的疾病，根據統計，孩童發生睡眠障礙的比率約為 25%。孩童晚上睡不好，沒有得到充分的休息，嚴重的情形會造成孩童白天容易頭痛，注意力缺失及自信心低落，有些孩童被診斷為注意力缺失症候群，長期吃注意力缺失症候群的藥物不見改善，未治療的孩童長久下來，甚至會有智能不足的情形[21]，其實這些是睡眠障礙問題[22]。

為了有助於建立睡眠品質若能藉由控制如溫度、溼度、燈光等居家設備來營造一個舒適的環境將有助於睡眠的品質的提昇。因此本論文提出一套具備即時分析能力的腦機介面系統以改善一般生理訊號監控系統的缺點，並結合腦波精神狀態監測與居家環境控制，提供給有睡眠障礙的病人或行動不變的老人具有一個優質的智慧型居家生活環境控制，以達到良好的睡眠品質。而這種智慧型居家環境控制是以嵌入式系統為基礎，具有跨平台多工程方式來增加系統即時處理的效能，最大的特色在於這個多工排程可以利用網路讓 task 分別在不同的機器上工作，不用侷限在一個機器上，也可以是不同的作業系統平台。最後，這個多工排程方法結合即時的生理狀態偵測的演算法，並可整合不同平台的環境控制器來控制週遭的環境狀態。

## 1.4 論文架構

本論文共分為六個章節，第一章說明本論文的相關研究背景、研究動機及系統整體架構的介紹；第二章針對腦電位訊號量測系統的部份進行說明；第三章著重於本研究在嵌入式系統上所設計的嵌入式多工排程系統；第四章將說明系統的設計與實現方式；第五章將針對第二、三、四章的設計作測試與驗證，並對我們所實現的系統做討論；第六章中，則是對本論文作一個總結，並對未來的研究方向作一說明。



## 二、腦電位訊號感測系統

### 2.1 簡介

腦電位訊號感測與無線傳輸系統的架構，先藉由電極(electrde)將 EEG 訊號量測出來再做後續放大處理。電路包含三大部分如圖 2.1 所示：1. 腦波量測電路，主要是以放大器電路及電極所組成，負責腦電位訊號的擷取與放大。2. A/D 轉換，由 CPLD 主控，負責控制數位類比轉換(A/D converter)、將類比訊號轉成數位訊號，同時也負責無線傳輸封包編碼與解碼、訊號格式轉換等工作。3. 無線傳輸模組，負責將數位的訊號透過射頻訊號傳出去[22]。

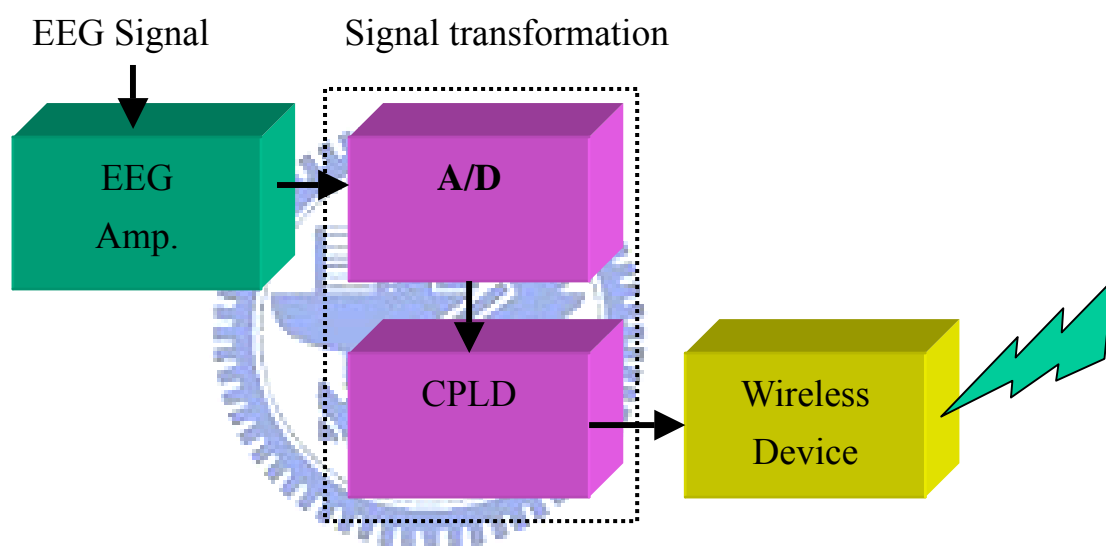


圖2.1、腦電位訊號感測與無線傳輸系統架構圖

放大器的部分可分為：前置放大電路、帶通濾波電路、增益放大器、帶拒濾波電路、A/D Converter，五個部份，這部分稱為腦機介面(Brain Computer Interface，BCI)。腦機介面是利用腦部訊號，讓使用者可以直接與外界溝通。當大腦活動的時候，會產生特定的腦波變化，因此可利用腦電波儀對腦波進行監測與辨識，最後用來控制儀器或輸出訊號。此種技術主要在於幫助因神經肌肉損傷而行動受到阻礙的人(如肌肉萎縮、中樞神經系統損傷、重度中風的病人等)，使他們可以不需要依靠周邊神經和肌肉，能夠使用腦部的訊號，就可以達到與外界溝通、傳達訊息、自主行動，以及自我照顧等目的



## 2.2 腦電位訊號簡介

大腦皮層有大量的神經元，這些神經元的活動有某種程度的規律性，因此置於頭皮上的成對電極可用來記錄自大腦皮層上產生的電位變化，此電位變化是由有節奏的電訊號 (Electrical Rhythms) 與暫態放電(Transient Discharge)所組成，這些變化波形就是所謂的腦電位訊號，俗稱腦電波(electroencephalogram, EEG)，不同的 EEG 測量位置、頻率範圍、振幅大小、訊號波形、週期性和產生方式可以區分不同的 EEG。

EEG 的紀錄有其困難所在，主要在於 EEG 的波形很微小，腦波訊號經過腦膜、腦脊髓液、腦殼到達頭皮時，其振幅峰對峰值大約只有  $1\sim 100\mu V$ ，頻率大約在  $0.1\sim 100\text{Hz}$ 。其次，不同材質的電極與電極跟頭皮的接觸程度都會影響量測記錄的結果，實驗操作時容易受到雜訊干擾。根據不同頻率範圍，大致上將 EEG 分為五類，如表 2.1 所示，以下針對 Alpha( $\alpha$ )、Beta( $\beta$ )、Theta( $\theta$ )、Delta( $\delta$ )等波形[23]分別做介紹：

### 1. Alpha( $\alpha$ )：

頻率為  $8\sim 12\text{ Hz}$ ，一般正常人在處於安靜、身體放鬆、大腦休息的狀態下，會出現這樣的週期波，是「意識與潛意識層面」之間的橋樑，其電位約為  $50\mu V$ ，在腦枕葉部(occipital region)及頂葉部(parietal region)最明顯。

### 2. Beta( $\beta$ )：

頻率為  $12\text{ Hz}$  以上，但一般很少高於  $50\text{Hz}$ 。清醒及警覺時尤其明顯，也是邏輯思考、計算、推理時需要的波，屬於「意識層面」的波。在頂葉部及額葉部(frontal region)會較明顯，電位約  $20\mu V$ 。

### 3. Theta( $\theta$ )：

頻率介於  $4\sim 7\text{ Hz}$ ，主要在兒童的頂葉部及顳葉部(temporal region)會出現，在深睡作夢、深度冥想時特別明顯，屬於「潛意識層面」的波，許多的腦疾病患者，可以找到  $\theta$  波。

### 4. Delta( $\delta$ )：

頻率低於  $4\text{ Hz}$  的腦波頻段，在一歲大嬰兒睡眠時，以及有嚴重器官性疾病的患者身上尤其明顯，主要出現在兒童的枕葉部，及成人的額葉部。



表 2.1、腦電波分類表

| 腦波         | 頻帶範圍    | 說明                   |
|------------|---------|----------------------|
| $\beta$ 波  | 12-30Hz | 平常活動(緊張及激怒時 25-30 赫) |
| $\alpha$ 波 | 8-12 Hz | 放鬆的身心                |
| $\theta$ 波 | 4-7 Hz  | 靜坐進入深度冥想             |
| $\delta$ 波 | 4Hz 以下  | 熟睡                   |

在臨床實驗上，EEG 訊號會依據電極擺設的位置而影響其振幅、相位與頻率。一般最常用的架構是由 International Federation of EEG Societies 所建立的 10-20 EEG 電極擺設系統[24]。如圖 2.2 所示其位置是以枕骨突出部分和鼻樑為界線將頭部分成左右兩塊，再以雙耳上緣為界將頭部分為前後兩塊。記錄方式基本上可分為單極式、平均式、與雙極式，本論文是採用單極式記錄以符合可攜式的運作模式。

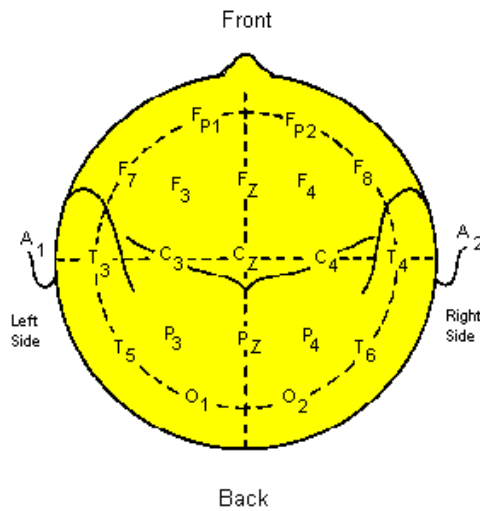


圖2.2、10-20 系統電極擺設位置圖

資料來源：<http://faculty.washington.edu/chudler/1020.html>

## 2.3 腦電位訊號量測電路

### 2.3.1 腦電位訊號量測電路設計

腦波量測電路方塊圖如圖 2.3 所示，先藉由電極將 EEG 訊號量出再做後續放大處理。在生理訊號量測上，我們採用單極式量測，所實現的電路可分為五個部份：前置放大電路、帶通濾波電路、增益放大器、帶拒濾波電路、和 A/D Converter。

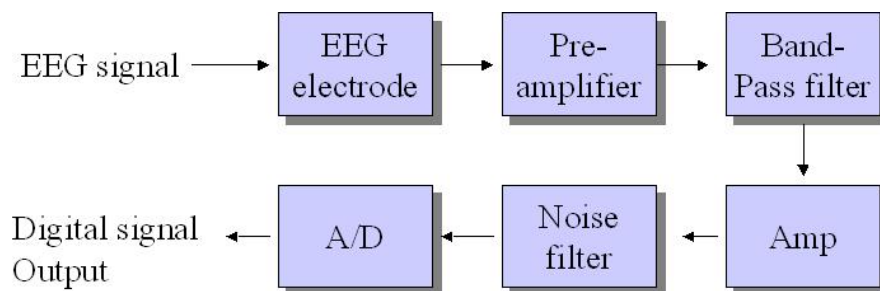


圖2.3、腦波量測電路方塊圖

#### 1. 前置放大電路

這一級主要為放大作用，由儀表放大器 AD620 為主體所構成，採用 AD620 為一低功率，有高準確度的放大器(Amplifier)，而其增益(Gain)範圍可從 1 到 10000。

#### 2. 帶通濾波電路

由於在受測期間，接收到的腦波可能會存在著不可避免的雜訊，因此必須有濾波器的存在。腦波的頻寬大小大約從 Delta 波的 0.5~4Hz 到 Gamma 波的 22~100Hz，因此希望其濾波器的頻帶位於 1 到 100Hz 之間。在帶通濾波電路的設計上，我們使用一組高通濾波器以及一組低通濾波器所組成。

#### 3. 增益放大器

這一級的主要功能是将上一及處理後的腦波頻帶的訊號做放大，放大倍率为 5000。

#### 4. 帶拒濾波器

為避免 60Hz 室內配電干擾可能造成的訊號失真問題，加入帶拒濾波器，讓干擾降到最低。

## 5. A/D Converter

經過放大電路輸出的訊號為類比的訊號，為避免在無線傳輸的過程受到干擾而失真，故在輸出的末端將訊號轉成數位號，讓干擾降到最低。

### 2.3.2 無線訊號傳輸模組

在無線訊號傳輸模組的選擇上，由於考慮到傳輸功率對受測者的影響，本論文選擇一種低功率的無線射頻(Radio Frequency,RF)模組來作為無線傳輸的介面。再以 CPLD 將 A/D(Analog to Digital)轉換器所轉出來的數位訊號包裝成無線傳輸模組所需的封包格式，傳至後端由嵌入式系統做解碼與分析的動作，架構圖如圖 2.4 所示。

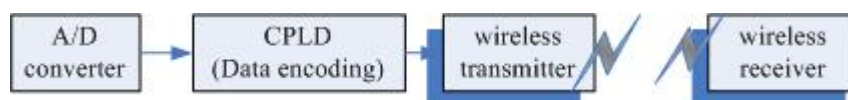


圖2.4、無線訊號傳輸架構圖

在無線傳輸模組方面，我們選擇研廣科技無線模組 RF3105 來做為傳輸的介面。我們選擇這個模組的主要原因在於其耗電量低、傳輸距離長以及其模組體積小的優點。其控制模式是以 RS232 的方式來控制，所以在模組切換跟使用上也比較有利於開發與使用，而在安全性上，RF3105 的模組上有控制器做編解碼的動作，有三種頻率可供選擇: 433~435Hz、868~870Hz、902~928Hz，我們所使用的頻帶是 902~928Hz[25]。



圖 2.21、RF3105 控制模組架構圖

### 2.3.3 CPLD 與 VHDL 硬體描述語言

可程式邏輯閘(如 CPLD、FPGA(Field Programmable Array))的迅速興起，提供使用者可以在短時間內開發自己所需的控制器。本論文便是利用 CPLD 與 VHDL 硬體描述語言來開發前端控制無線模組與 A/D converter 的系統。CPLD 的全名是 Complex Programmable Logic Device，是可程式邏輯邏輯元件(Programmable Logic Device ,PLD)的一種。可程式邏輯元件是一種可讓使用者組成特定邏輯功能的數位積體電路，並可快速協助邏輯 IC 的設計與驗證工作。CPLD 是由許多個邏輯方塊所組成，各邏輯方塊均類似一個簡單的 PLD 元件。邏輯方塊間的相互關係則由可編程的連線架構，將整個邏輯電路合成而成[26]。本論文所使用的 CPLD 是由 ALTERA 公司生產的 FLEX 系列的 CPLD，其發展板如圖 2.5 所示。

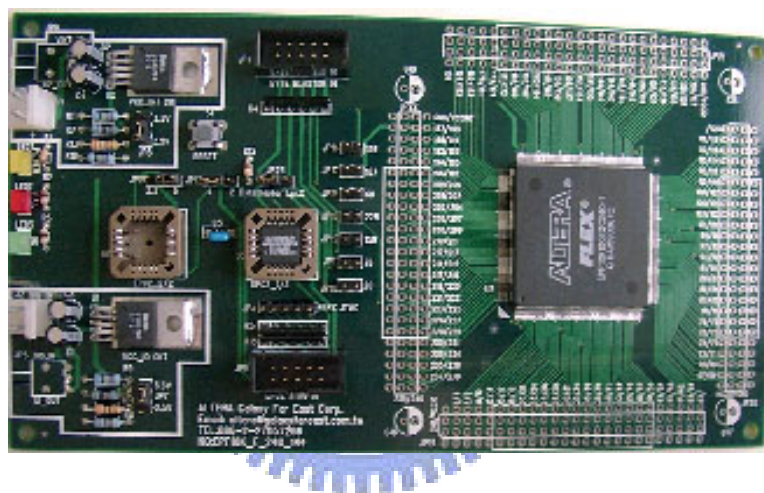


圖2.5、FLEX系列發展版

程式上，我們是以 VHDL 來撰寫我們所設計的系統。VHDL 的全名是 Very High Speed Integrated Circuit Hardware Description Language。VHDL 與 Verilog HDL 一樣是一種描述數位電路的標準語言，是由美國國防部開始發展的，其優點在於功能強大、設計靈活、多種不同的描述風格。而我們所使用的軟體是由 ALTERA 所提供的 Max +plus II，這套軟體提供一套完整的設計工具，包括文字編輯、圖形編輯、模擬、腳位配置、程式燒錄等設計流程一應俱全，其介面如圖 2.6 所示。

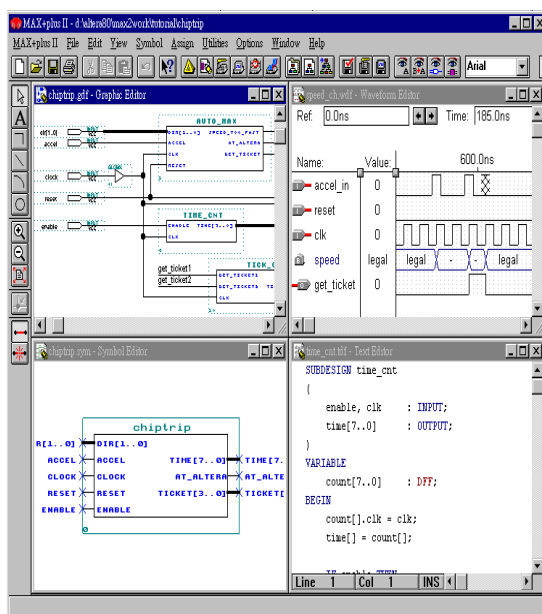


圖2.6、Max + plus II 軟體介面

### 2.3.4 類比/數位訊號轉換與無線傳輸模組控制

如圖 2.7 所示，以 CPLD 為中心的無線訊號傳輸部分主要是接收放大電路適當放大後的類比訊號，將其轉換為數位訊號，接著將數位訊號轉換為無線傳輸模組所需的訊號格式，再利用無線模組將訊號送出。以下將對如何控制 A/D converter 與應用 RF3105 傳輸模組來實現無線訊號傳輸的功能作介紹。



圖2.7、無線訊號傳輸架構圖

#### 1. 控制 A/D converter：

控制 A/D converter 以取得其轉換得到的數位訊號。在 A/D converter 方面，我們採用 ANALOG DEVICE 的 AD570。AD570 是一個具有高可靠性及低耗電的特性的 8-bit A/D converter。類比電壓輸入可以是單極性的(0V~+10V)，或者是雙極性的(-5V~+5V)。AD570 的內部架構圖與控制時序圖分別如圖 2.8 及圖 2.9 所示。

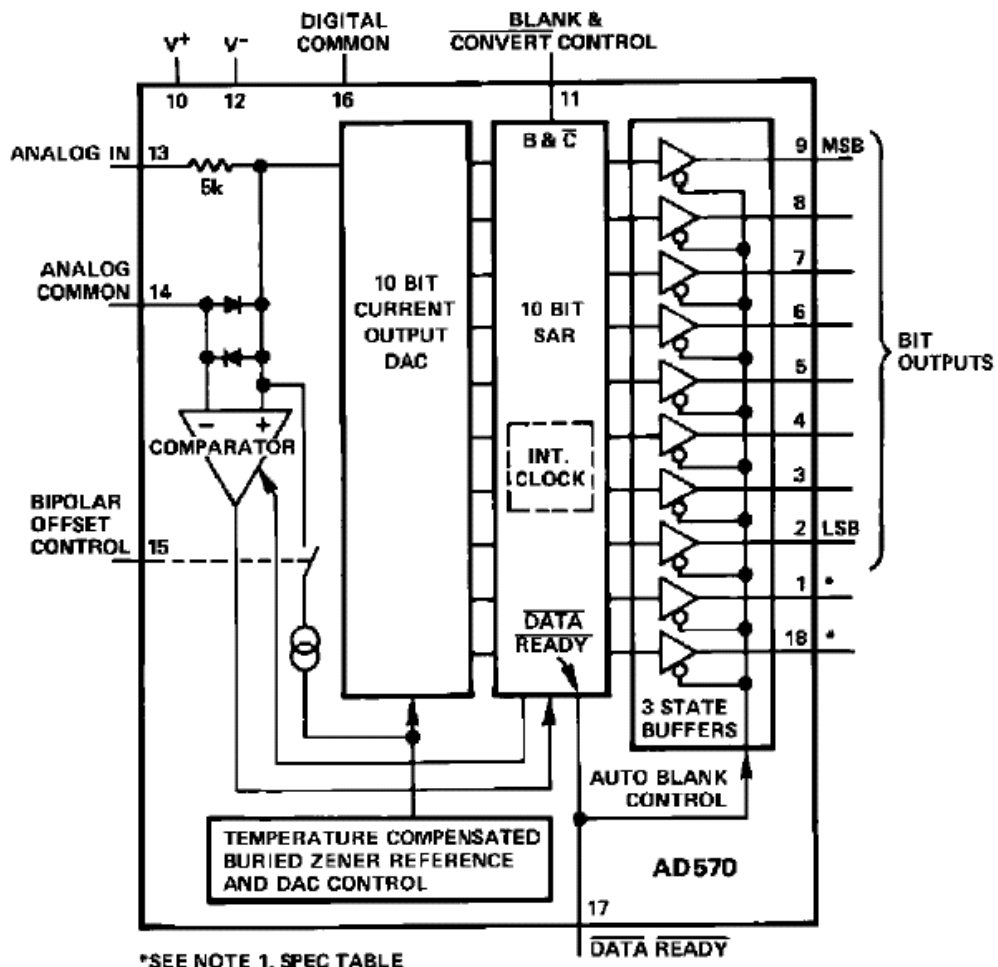


圖2.8、AD570內部架構圖

資料來源:www.datasheetcatalog.com

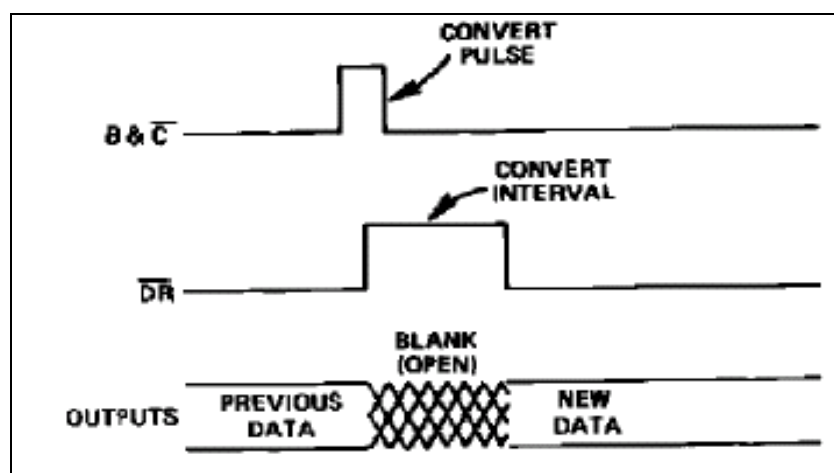


圖2.9、AD570控制時序圖

資料來源:www.datasheetcatalog.com



## 2. RF3105 傳輸模組

在 RF3105 的控制上，是利用 RS232 界面，所以在 CPLD 上我們必須實現一將訊號轉換成 RS232 封包格式的模組。以兩個頻道為例，我們利用 CPLD 所實現的模組其內部架構如圖 2.10 示，表 2.2 則為模組接腳表。其中鮑率產生器、UART 傳送模組[27]負責將數位訊號以 RS232 的訊號格式送出，而傳送值產生模組則為本模組的核心部份，其主要負責兩項工作：

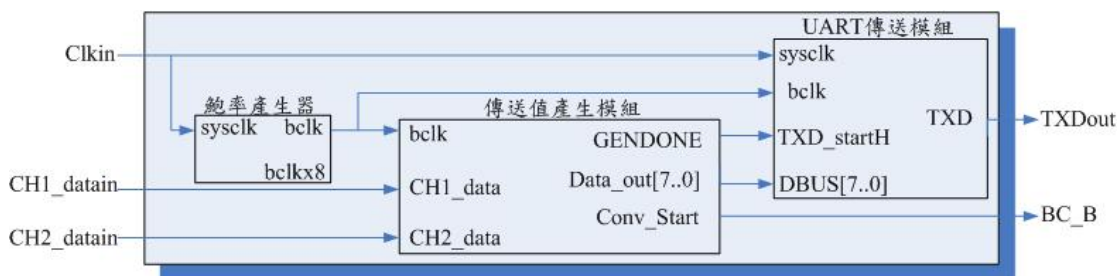


圖2.10、RF3105控制模組架構圖

表 2.2、RF3105 控制模組接腳表

| 信號名稱             | 屬性     | 作用極性 | 功能                        |
|------------------|--------|------|---------------------------|
| Clkin            | input  | n/a  | 系統之震盪頻率                   |
| CH1_datain[7..0] | input  | n/a  | A/D converter 頻道 1 數位資料輸出 |
| CH2_datain[7..0] | input  | n/a  | A/D converter 頻道 2 數位資料輸出 |
| TXDout           | output | n/a  | 傳送資料至 RF3100 無線模組         |
| BC_B             | output | High | A/D converter 致能訊號        |

產生送入 RS232 傳送模組的資料：產生送入 RS232 傳送模組的資料並驅動該模組藉以將資料以 RS232 的訊號格式送入無線傳輸模組。此外，本論文所完成的傳輸系統中，同一個傳輸通道必須傳送多個頻道(channel)的資料，所以，除了欲傳送的頻道資訊外，我們還加上識別碼藉以告知接收端目前收到的訊號是哪一個頻道，封包形式如表 2.3 所示。如圖 2.11 所示，當接收到識別碼後，會隨即讀取四個頻道的資訊，因此，識別碼的數值便不能出現在資料中，而針對這點，我們在傳送值產生模組會對這些值做一些修改，以便傳輸時不會有問題發生。

為了簡化整個資料傳輸的流程，我們採用單向的資料傳輸，也就是傳送端與接收端不做交握的動作，傳送端不斷的傳送資料，而接收端只負責取接收資料。

表 2.3、無線封包

| Byte 1 | Byte 2 | Byte 3 | Byte 4 | Byte 5 | Byte 6   |
|--------|--------|--------|--------|--------|----------|
| 識別碼    | Data 1 | Data 2 | Data 3 | Data 4 | Checksum |

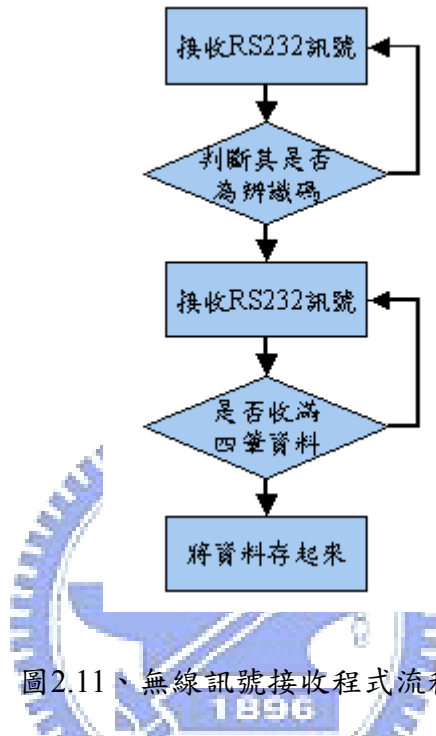


圖2.11、無線訊號接收程式流程圖

實現上，我們讓傳送值產生模組每隔一段時間去驅動 UART 傳送模組，使其將 RS232 訊號送出，如此一來我們只要控制這一個驅動訊號的頻率便可改變訊號的取樣頻率。



### 三、跨平台訊息交換系統

本論文結合了前端的嵌入式生理訊號處理，以提供後端的環境控制，形成一個舒適的居家環境，藉以提升使用者之生活品質與工作效能。圖 3.1 是整個系統簡單的流程。

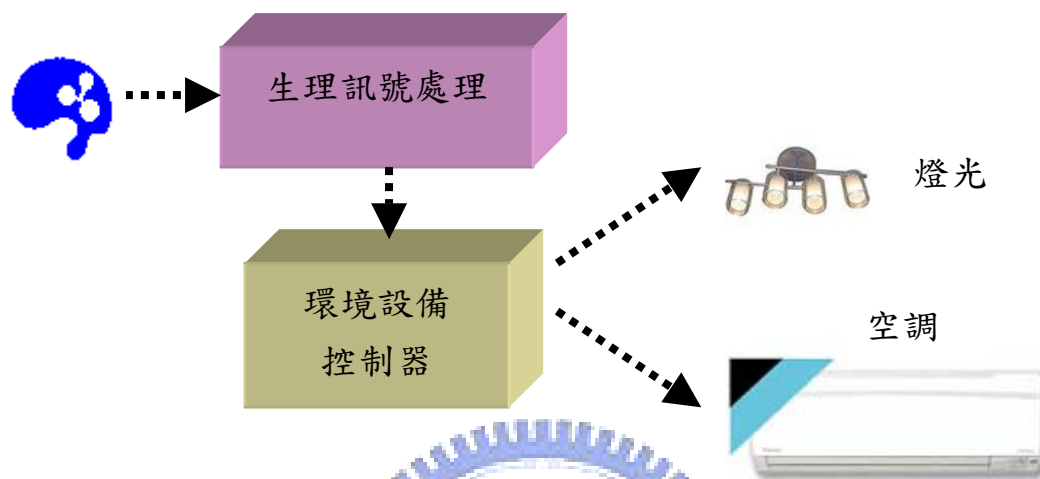


圖3.1、系統流程架構

在系統設計過程中，必須考慮需要克服的 3 個問題為，如何即時有效利用 Embedded System 的系統資源，系統必須建置在不同的機器上甚至於是不同的平台，及不同機器上運作所衍生出來溝通的問題。上述困難的解決方法如下：

1. 使用多工排程：使用多工排程技術能增進系統的運作效率。以解決即時有效利用 Embedded System 的系統資源。
2. 跨平台訊息交換系統：使用跨平台的訊息交換系統可以解決不同機器在運作上溝通的問題，也可以解決不同平台溝通的問題。

綜合上述的解決方案，我們設計一個跨平台的訊息分享平台，一方面提供多工排程，另一方面也解決不同機器/平台間溝通的問題。將分別於以下的小節中做詳細的介紹。

## 3.1 多工排程設計

由於在 Embedded System 上，有許多工作都必須即時作處理，例如：即時接收無線訊號、即時處理生理訊號。如果要將這些工作由同一個處理程序完成，那麼勢必要在 Embedded System 會有許多工作要規劃其執行時間，而且也會有一些等待的時間而無法符合即時的要求。因此，本論文提出一套多工排程系統以達到，減少時間的浪費，實現即時運算與資料接收的能力；將處理程序模組化，工作更系統化更容易維護；跨機器/平台的工作模式，方便擴充新的處理功能與工作規劃。

在多工排程系統的設計與實現上[28]，可分為以下 3 個步驟：

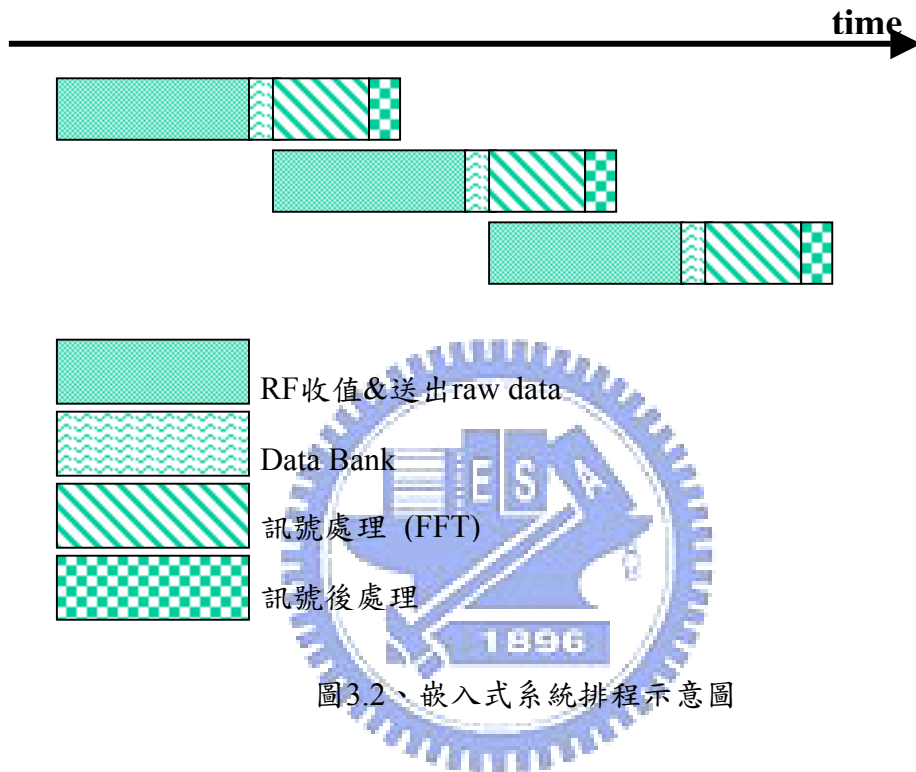
1. **建立工作程序**：首先便是要建立多個工作程序，以便系統可以做排程與切換。
2. **建立排程仲裁機制**：在排程系統中，最重要的便是仲裁機制。仲裁機制的存在便是為了決定現在系統必須做哪個工作。由於本論文所實現的嵌入式系統中，便有一個Linux的作業系統，為了善用我們所擁有的資源，一般時候我們讓作業系統去主管排程的動作，而在某些比較特殊的時間點，例如在處理流程中必須強制等待時，我們設計了一個輔助排程的仲裁機制來做仲裁。
3. **建立多重處理程序之溝通機制**：這其實是在多工排程中最麻煩的一部分。在我們所實現的系統中，處理程序間會有一定的相關性，例如：需要做資料交換、處理程序同步的工作。本論文針對這項需求提出一套多工排程機制來負責這些工作間同步與通訊的工作。

### 3.1.1 工作切割與基本排程架構

整體而言，Embedded System 所需要掌管的處理程序分別為無線通訊裝置控制、區域網路資共享、訊號處理以及訊號後處理等四項工作。根據工作執行的頻率，我們將所有的工作除了主程式以外區分為三大項：1. 無線模組資料接收換與區域網路管理、2. 資料暫存管理 3.資料處理與分析。由於無線訊號接收以及網路管理的頻率相同而且會有很高的相依性，當其分開時會消耗一些不必要的時間與控制，所以將其結合為一個工作。如圖 3.2 所示，相對於無線訊號接收以及網路資料分享的頻率來說，資料暫存管理發生的頻率就小很多，因此，我們讓系統不斷的從無線模組收資料並輸出結果至網路上，而讓作業系統自行判斷將其他工作放入。當無線模組收滿足夠資料時，

會將收到的資料送到暫存區，然後資料處理程序會讀取資料開始運算，然後繼續接收無線模組的資料。

如此一來，在做訊號處理時，Embedded System 並不會只是在等待，而同時會持續接收無線訊號以及將資料傳利網路送至顯示介面的電腦上。因此，可以滿足即時資料接收與即時訊號處理的需求。



### 3.1.2 多重處理程序之溝通機制

在多重處理程序的溝通機制中，主要的目的是讓多個處理程序可以做資料交換與資源共享和讓處理程序之間可以同步。在 Linux 的作業系統中一共提供四種處理程序之間的溝通機制(inter-process communication, IPC)：訊息佇列(message queue)、信號(semaphore)、共享記憶體(shared memory)、以及 socket，使得程序間的溝通判斷可以達到同步的功能。分別簡述如下：

#### 1. 訊息佇列(message queue)：

訊息佇列可以視為一些訊息的組合，這些訊息都位於核心程式中，彼此以鏈結串列(linked list)的方式連結，而每個訊息佇列都要透過辨識元來辨識。佇列裡的每筆訊息都有一個型別為長整數(long int)的欄位，一個非負值的訊息長度，和一個真正的資

料長度(這個是跟訊息長度一致)，當要傳送訊息到一個佇列時，必須把這些值指定給 msgsnd 函式。若要從訊息佇列裡讀取訊息則需呼叫 msgrcv 函式，在讀取時不一定需要使用先進先出的順序來讀取這些資料，也可根據這些訊息的種類來讀取。在實際上使用發現這個機制並不會比其他的 IPC 快，所以，這種 IPC 不夠實用。

## 2. 信號(semaphore)：

信號與一般的 IPC 機制不太一樣，實際上，信號是一個計數器，它用來讓數個處理程序可以共同讀取一個讓大家分享的資料物件。其使用方法如下：

(1) 先測試控制這個資源的信號。

(2) 如果這個信號值為正，那處理程序就可以使用這個資源。當程序在使用此資源時，它必須將這個信號的值減一，表示它正在使用這個資源的一個單位。

(3) 如果測試出的信號值為0，那麼這個處理程序就必須要設為睡眠狀態，直到信號的值大於0為止。當這個處理程序醒來時，它必須再從第一步開始。

如果有一個處理已經用完由信號所控制的資源時，信號的值就會被加 1。如果有任何處理程序是因為等待這個信號的值而被設為睡眠，這時他們就會被喚醒。信號其實是一種同步化的機制，而非真正的 IPC，其用途在獨立使用時，可用以讓處理程序同步或與其他的 IPC 機制搭配使用，可對某些共用的資源進行保護。

## 3. 共享記憶體(shared memory)：

共享記憶體可以讓兩個以上的處理程序來分享同一塊記憶體，這是一種速度最快的 IPC，因為他不需要伺服器端和用戶端間相互拷貝資料，在使用共享記憶體時要注意如何在數處理程序使用同一塊共享記憶體時能夠讓它們之間的存取動作同步(synchronize)，這種情況通常是以信號來協調各個處理程序讀取共享記憶體的作業。

## 4. 插座(Socket)：

Socket 是使用 TCP/IP 的通訊模式來交換資料，所以只要有網路的設備都可以進行單機交換訊息，或者是不同設備，不同平台間交換訊息。本論文為結合即時生理訊號處理與居家環境控制，使用此種技術來達到訊息交換的功能。這兩者分屬於不同的

設備，為達到訊息交換的目的必須採用可以運作於不同機器的溝通機制，因此我們使用此種技術來作為多重處理程序之溝通機制。

## 3.2 訊息交換平台設計

為了達到各個工作程序能交換訊息且能跨不同的設備與平台，本論文建構了一個訊息交換平台(Data Exchange)，將不同的工作程序建構在此平台上運作，能互相交換資料及控制。

### 3.2.1 訊息交換平台簡介

訊息交換(Data Exchange) 是使用 channel 來作為訊息傳遞為目的地。也就是說透過 channel 可以讓不同的應用程式互相溝通，使用不同的 channel 訊息可以讓不同群組的應用程式傳遞訊息而不相干擾。就像是使用無線電在不同的 channel 下溝通。每一個 channel 為 20 個字元的字串，可以是文字也可以是數字，如: BCI\_EMBEDDED，BCI\_12345... 等等。一般使用 Embedded Data Exchange 的應用程式可以有以下的三種互動行為：

#### 1. Publish/Subscribe

在此模式下的工作是 Event-Driven 的方式。Publisher 將 event 訊息送出去，有聽到此 channel 的 subscribers 將都會收到此 event 訊息，而啟動對應的程式來處理。圖 3.3 說明 當 Publisher 將訊息送出後，Subscriber B 與 C 都有聽此 event message 的 channel，故兩者都會收到這個 Event message，而分別去處理這個 event message。Subscriber A 未聽此 event message 的 channel，所以不會收到此 event message 所以也不會有任何的動作。

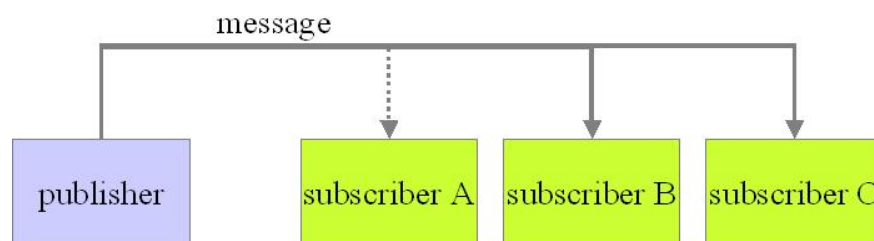


圖3.3、Event-Driven Publish/Subscribe Interaction

## 2. Request/Reply

如圖 3.4 是一個 Demand-Driven Request/Reply 的行為模式，這樣的行為模式包含著兩個 point-to-point message。Client 端將需求的訊息送出，而聽此 message channel 的 Server 會收到此 message，當 Server 處理完這個 message 後會將結果送回給 Client。

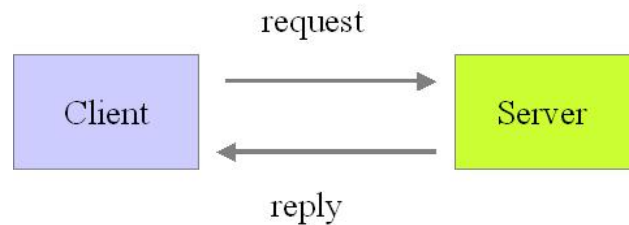


圖3.4、Demand-Driven Request/Reply Interaction

## 3. Broadcast Request/Reply

Broadcast Request/Reply 也像之前的 Request/Reply 一樣是 Demand-Driven 的行為模式。傳統的 request/reply 是一個 client 對應一個 server，而如圖 3.5 Broadcast Request/Reply 可以有許多 Servers 可以接收 Client 的需求並回應給 Client。

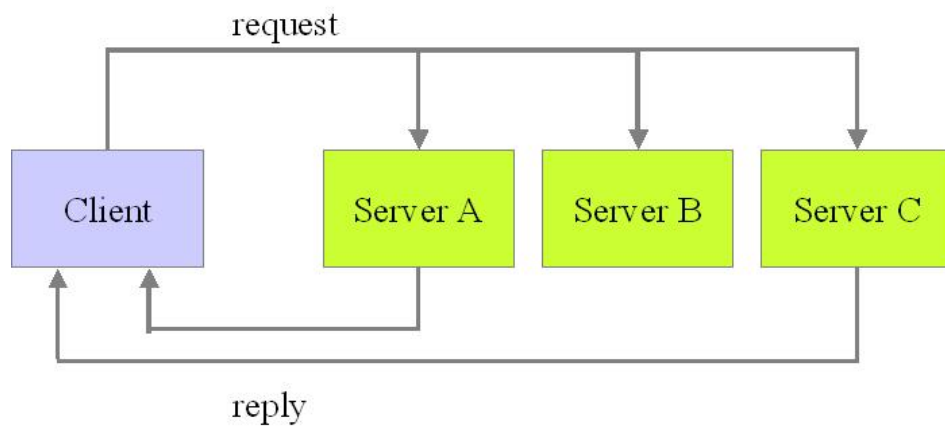


圖3.5、Broadcast Request/Reply Interaction



### 3.2.2 訊息交換平台之系統架構

訊息交換平台是利用 TCP/IP 的通訊協定來作為訊息溝通的橋樑，其核心主要由一個 Daemon 和函式庫(Library)組成，圖 3.6 表示系統的架構。以下將分別說明各元件的功能。

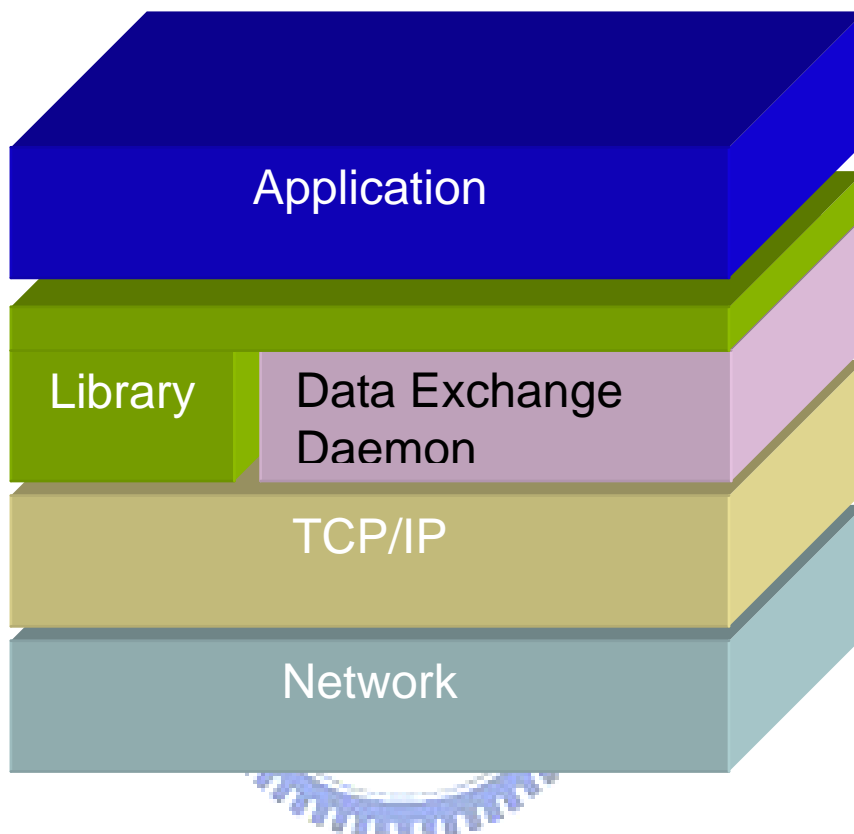


圖3.6、Embedded 訊息交換平台系統架構

#### 1. Daemon

Daemon 負責底層的工作，當接收到網路上的訊息後，Daemon 將檢查是否又應用程式在聽此 channel 的訊息，若有則將此訊息傳給這個應用程式。

#### 2. Library

提供應用程式與 Daemon 溝通的橋樑，此函式庫主要提供以下幾個 function 來建立應用程式與 Daemon 的聯繫。

##### a. MBD\_Init()

始化，此 function 建立與 daemon 的聯繫，daemon 未來將使用此聯繫將訊息傳給應用程式。

```
int MBD_Init(MBD_Session session)
```

回傳值：

int 傳回 0 表示成功，其他則代表失敗。

參數：

session：為一資料結構，裡面存放著與 daemon 聯繫的相關資料。

#### b. MBD\_Term()

終止之前 MBD\_init() 建立與 daemon 的連線。

```
void MBD_Term(MBD_Session *)
```

回傳值：

無。

參數：

session：為一資料結構，裡面存放著與 daemon 聯繫的相關資料。

#### c. MBD\_Listen()

此函數告訴 daemon 應用程式要聽那一個頻道的資料。未來 daemon 收到這個頻道的訊息後會將訊息轉送給應用程式處理。

```
int MBD_Listen(MBD_Session *session, MBD_Subject subject, int *MyCallBackFun)
```

回傳值：

整數值(int)，0 表示成功，其他表示失敗。

參數：

session：為一資料結構，裡面存放著與 daemon 聯繫的相關資料。

subject：欲接收訊息的頻道。

MyCallBackFun：當收到訊息後，daemon 會呼叫此程序來執行相關工作。

#### d. MBD\_Send()

訊息溝通為我們最重要的工作，此函數的功能就是將訊息送出去。讓其他的應用程式收到這修訊息。



```
int MBD_Send(MBD_Session session , MBD_Subject subject , MBD_Subject  
reply_subject , MSG_MsgID msg_id , char *buf , int size);
```

回傳值：

整數值(int)，小於 0 表示失敗，其他表示傳送成功的位元組數目。

參數：

session：為一資料結構，裡面存放著與 daemon 聯繫的相關資料。

subject：欲送出訊息到哪一個頻道。

reply\_subject：希望收到訊息的應用程式，回傳的頻道值。

msg\_id：訊息的代碼，可以用來定義此訊息的特殊分類編號。應用程式可以依  
它來分辨訊息的分類。

buf：欲傳送訊息的位置。

size：欲傳送訊息的大小，單位為 byte。

### 3.2.3 訊息交換平台應用

前面兩小節已經介紹訊息交換平台的系統架構，本小節將介紹分散式應用程式架構、訊息傳遞與接收步驟以及簡單訊息交換平台程式設計說明。

#### 1. 分散式應用程架構：

應用程式在 Data Exchange 架構下系統可以分成好幾個模組同步執行，這些  
模組可以放在同一機器上，也可以分別放在不同的機器上，如圖 3.7 所示。

#### 2. 訊息傳遞與接收步驟說明：

當規劃好系統的模組後接下來要介紹的是訊息是如何傳遞，如圖 3.8 所示，假設  
我們有兩個程式一個是發送者(sending)，另一個為接收者(receiving)。首先由接收者先  
開始啟動程式執行步驟 1 到步驟 3，這時接收者進入等待階段，接下來發送者啟動程  
式執行步驟 4 到步驟 5，當執行步驟 5 後 Daemon 會把訊息經由網路傳遞給接收者的  
Daemon，接收者的 Daemon 會執行步驟 2 所設定的 CallbackFun，執行完畢後接收者  
將又回到等待階段，直到程式被要求終止執行步驟 7 後結束程式。

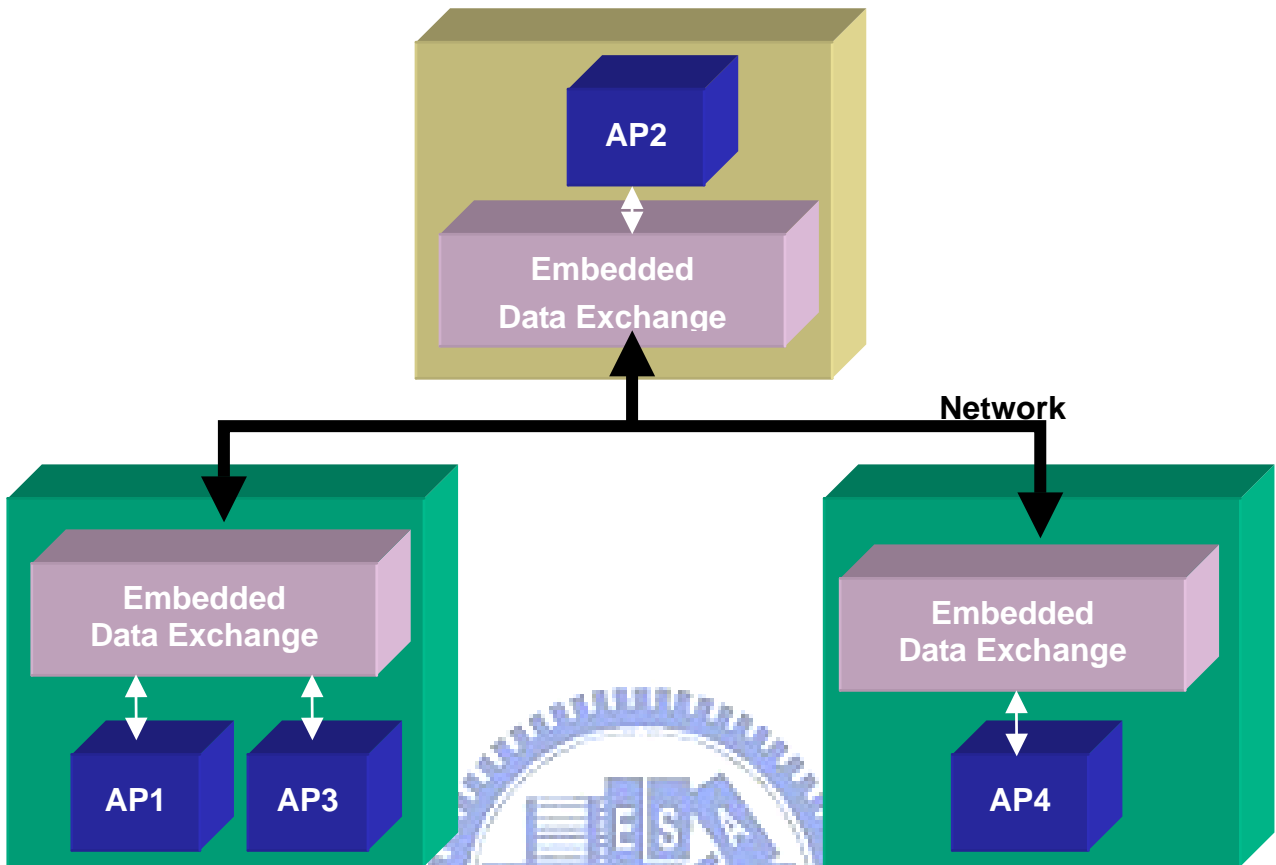


圖3.7、Data Exchange下的System分配

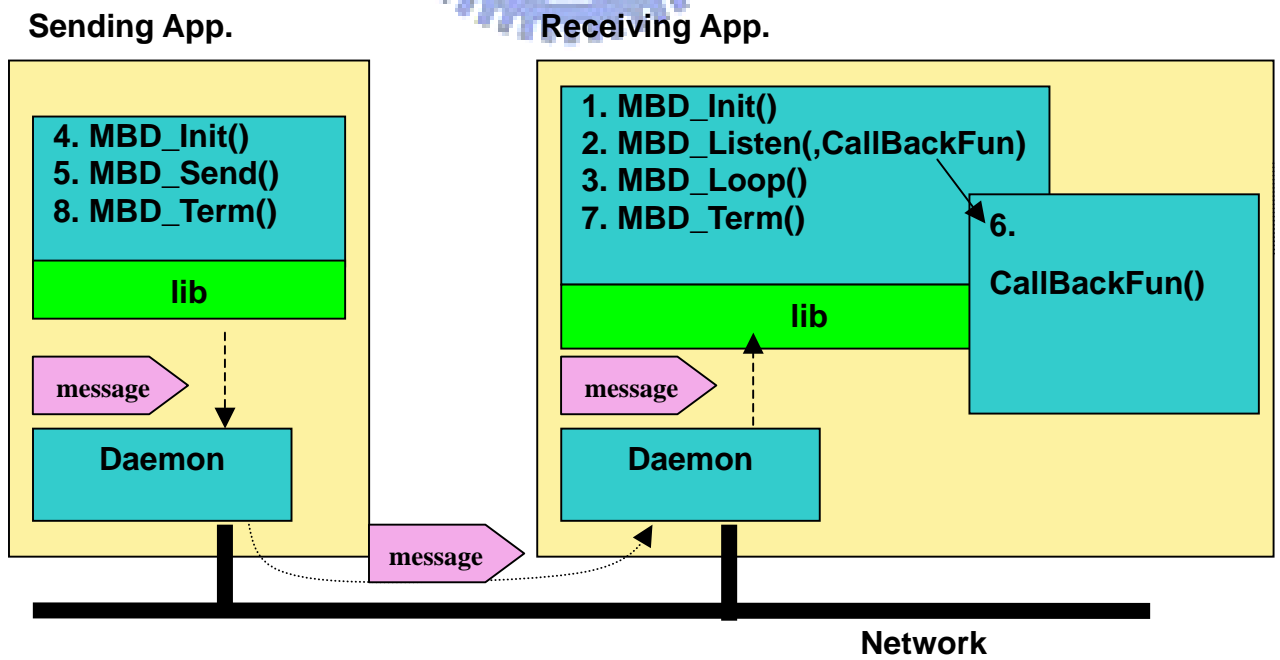


圖3.8、發送與接收關係

### 3. Sample Code :

底下的 sample 分成兩個部分 main() 及 MyCallBack()兩個函式，main()裡面介紹如何初始化與 daemon 連結、接收哪個 channel 的資料、進入等待訊息、以及關閉與 daemon 連結。

```
int main(int argc, char *argv[])
{
    My_Session.SocketPort=DEF_SOCKET_PORT;    //設定通訊 port
    strcpy (mysubject,TASK_MGR_SUBJECT);       //設定 Channel
    MBD_Init(&My_Session)                      //建立與 daemon 連線初始化
    MBD_Listen (&My_Session,mysubject,(int *)MyCallBack) //設定 CallBack function
    MBD_Loop(&My_Session);                     //進入等待狀態
    MBD_Term(&My_Session);                     //結束 daemon 連線

    return 0;
}
```

MyCallBack() 則當訊息進來時 Daemon 會執行此函式，函式中介紹一般的處理程序以及發送訊息。

```
int MyCallBack(MBD_Subject _subject,MBD_MSG msg)
{
    //解 message 的 header
    MSGUnpackRqstHeader (msg, msg_id,0,&trans_id,&numRows, &rowSize);
    //系統程序初始化確認步驟
    if ( !strcmp(msg_id,"INIT_REPLY",10) ){
        if ( !strcmp( buf ,"RF_READY",8))
            rf_status=1;
        else if ( !strcmp( buf ,"DB_READY",8))
            db_status=1;
        else if ( !strcmp( buf ,"DSP_READY",9))
            dsp_status=1;
        if ( rf_status && db_status && dsp_status ) //是否完成系統初始化
            MBD_Send(&My_Session,RF_MGR_SUBJECT,NULL,"START_RF_FIRST",NULL,0);
    }else if ( !strcmp(msg_id,"DATA_READY",10) ){ //資料收集完成開始下一資料收集&訊號處理
        MBD_SendString(&My_Session,RF_MGR_SUBJECT,NULL,"START_RF",NULL);
        MBD_SendString(&My_Session,DSP_MGR_SUBJECT,NULL,"START_DSP",NULL);
    }
}
```

## 四、 優質生活環境系統之設計

優質生活環境科技重點是建立健康舒適之環境，提升使用者之生活品質與工作效能，並藉由了解不同使用者的行為特徵、生活習性與使用需求，結合生活創意與創新科技的應用，整合人文與科技，融入於智慧空間系統制動系統軟體建構，使得智慧空間具有更完備之偵測功能，以提供更人性化之服務。優質生活環境科技包含各種感測系統（如聲波、電磁波、溫度、加速度及生化等）、智慧機電系統，…等建構智慧空間平台的基礎建設。本研究主要目的為結合前端的生理訊號處理(偵測)及後端的環境控制(智慧機電系統)，來建構一個健康舒適之優質生活環境。

### 4.1 即時生理訊號處理

嵌入式生理訊號處理器上的訊號處理流程圖如圖 4.6 所示，當資料取得後，會經過以下處理程序，分別為：1. Down sampling、2.Hanning Window、3.Short-time FFT processing、4.Normalization、5.Moving average，分別敘述如下：

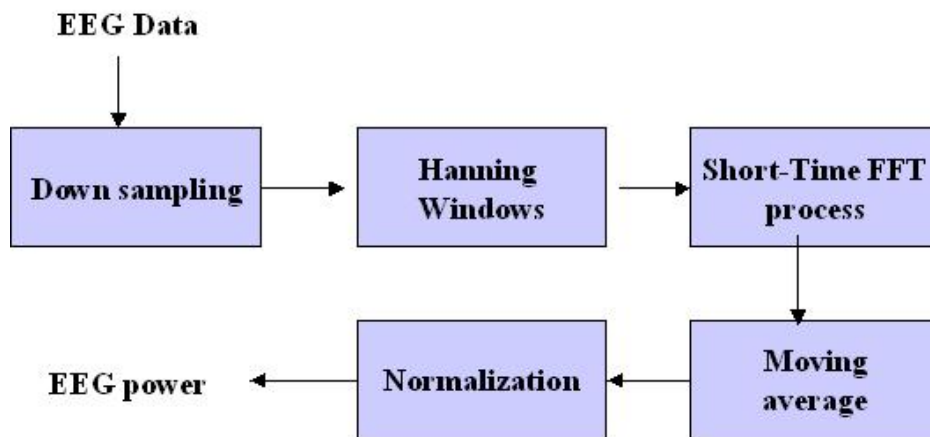


圖4.1、腦電位訊號處理流程圖

#### 1. Down sampling

在正式進入資料處理程序前，先將重新取樣成每秒 64 點，由於人體昏睡的時候的波形主要是在於 25Hz 以下的頻率，而之後我們分析模型也是取 1-25Hz，根據取樣定理，64Hz 的取樣頻率是足夠的。

#### 2. Hanning Window

由於如果直接將時間序列的訊號做頻譜轉換，會造成比較嚴重的失真，所以，一般在數位訊號處理上會在頻譜轉換前將時間序列的訊號乘上一個視窗(Window)函數。在我們的系統中，使用比較常用的 Hanning Window 來做為我們的視窗函數。

### 3. Short-time FFT processing

由於腦電位訊號是屬於較複雜的訊號，所以必須要轉成頻譜，才可以看出其特徵，在此我們以短時快速傅立葉轉換 (Short-Time FFT, STFFT) 的方式來將時間序列的訊號轉為頻譜訊號。在做完頻譜轉換後，我們僅選擇主要跟人體疲勞度有關的頻帶 (1-25Hz) 來做相關的分析與運算。

### 4. Normalization

經過短時快速傅立葉轉換得出各 EEG 每兩秒鐘的能量狀況後作正規化動作。

### 5. Moving average

由於人體精神狀況的改變，其週期超過 90 秒，因此，在訊號分析上不是只看某個瞬間腦波的值，而是看一段時間的變化。所以，把 EEG 訊號之頻譜作 Moving average (移動平均) 可以更進一步減少因為人為因素所引起的 EEG 雜訊而不影響我們所想觀察的腦電位訊號變化。濾除眼動、心跳及其它雜訊之訊號，在分析之前必須進行前處理。由於人體的精神狀況的改變，其週期超過 90 秒，不是只看某個瞬間腦波的值，所以圖 4.2 所示把 EEG 訊號之頻譜作 Moving-average (移動平均) 就是取 128 點 (2s) 的 EEG 訊號加上 64 點 (1s) 的重疊為一個 Window。接著把這些 192 點每隔 6 點再分成幾個 32 點的 sub-windows，每個 sub-window 再以補零的方法擴充成 64 點，作 64 點的 FFT 運算。接下來，把每一小段頻譜相對應的頻率上取中位數，得到一段平均之後的頻譜可更進一步減少因為人為因素所引起的 EEG 雜訊，例如：眨眼、搖頭...。在每隔兩秒輸出的 EEG 頻譜序列中，我們僅選擇與昏睡狀態比較相關的頻帶 1-25Hz 的 EEG 功率頻譜取 log 得到 dB 值。

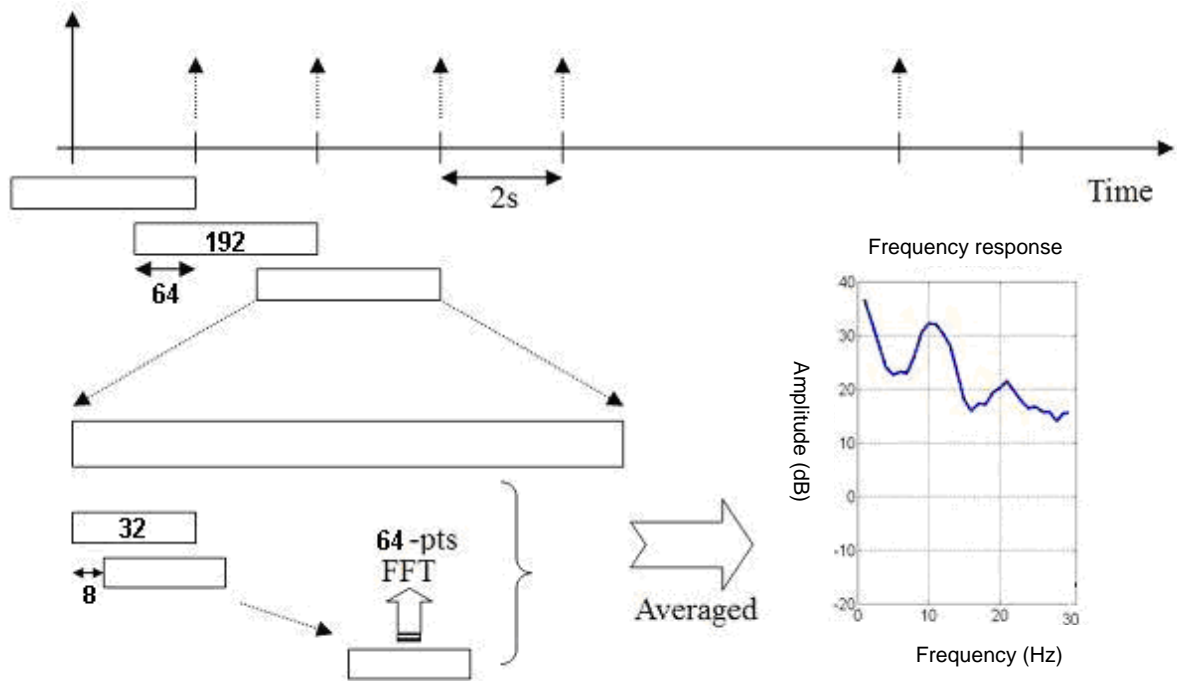


圖4.2、訊號之前處理

## 4.2 精神狀態偵測

分析人精神狀態的腦電位變化，可藉以找到產生疲倦、昏睡以時如圖 4.3 的腦電位變化。當精神狀態開始不好進入打瞌睡時的情況下 Alpha，Theta 波的 power 會開始上升。當進入睡著的情況下 Alpha 的 power 會開始下降，而 Theta 的 power 會持續上升。

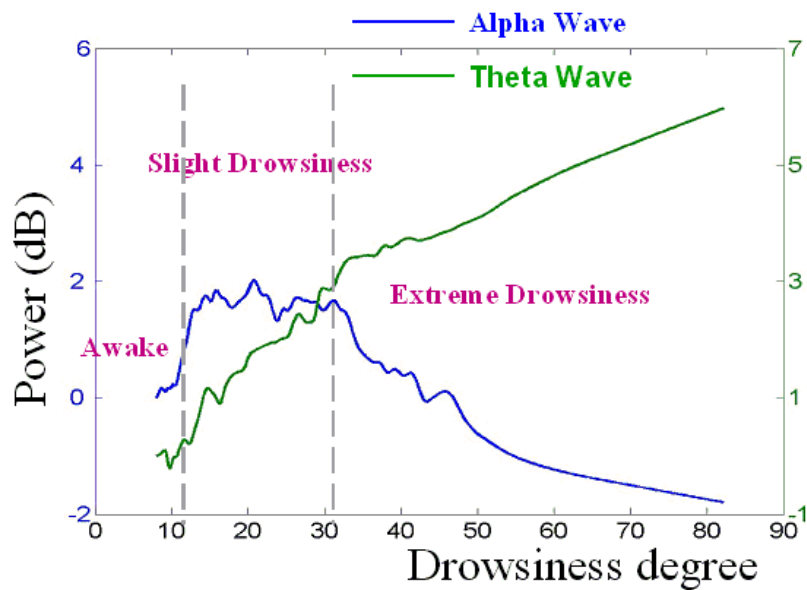


圖4.3、精神狀態改變時腦電位的變化

我們將精神狀態分成 Slight drowsiness、Extreme drowsiness、awake 三個層級，藉由即時的生理訊號演算法處理腦電位的變化得到 EEG 功率頻譜 dB 值，並比較 Alpha、Theta 的變化來判斷目前的精神狀態，並將判定後的精神狀態傳給環境控制器來控制環境。

### 4.3 優質生活環境控制平台

居家環控控制器可以區分為主控制器(Smart Room Controller)，環境裝置控制器(Control Point, CP) 與網路設備，如圖 4.4。下面小節將分別說明。

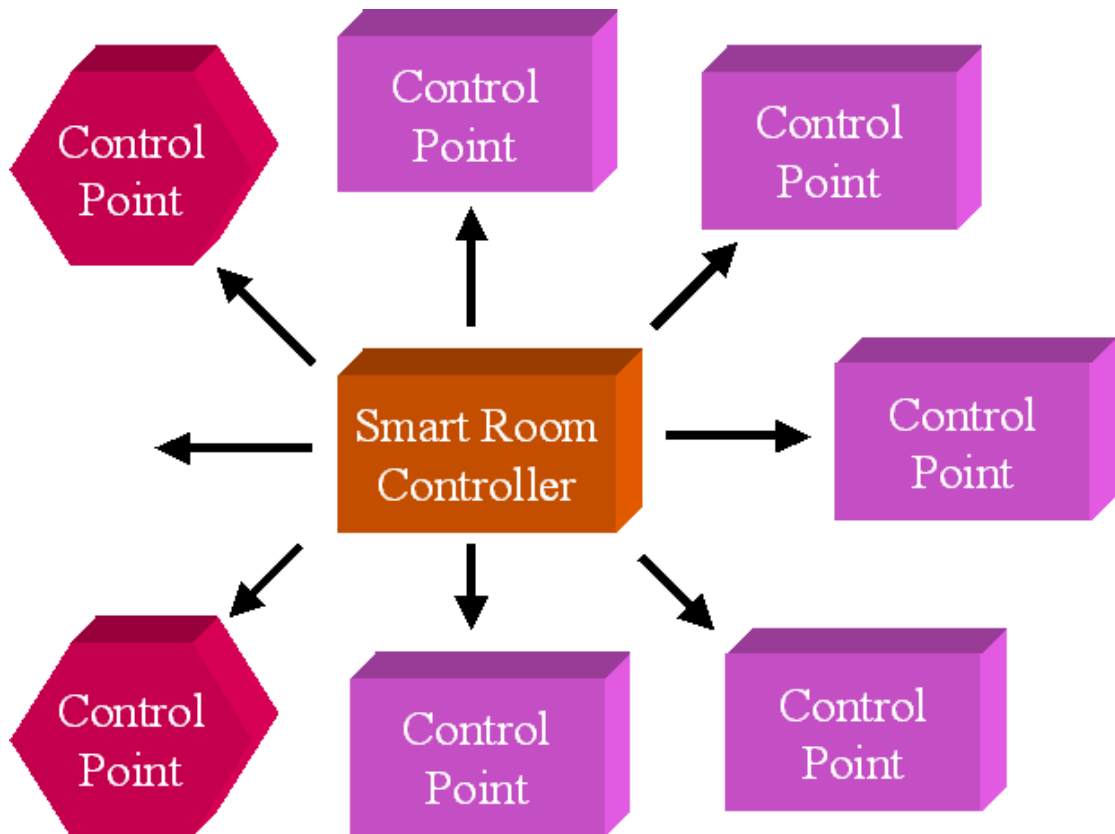


圖4.4、Smart Room系統架構圖



### 4.3.1 居家環境平台簡介

在建構一個居家環境控制系統上，我們規劃三個部分；第一個是設備控制點我們稱為 Control Point，第二個是主控制器我們稱為 Smart Room Controller，最後一個是用來連接這兩者的網路設備。其主要的功能分述如下：

#### 1. 設備控制點(Control Point, CP)：

CP 的主要目的為控制居家環境的設備。藉由不同的控制器將可以控制家裡的設備，如電燈，風扇...等等。在整個系統架構中 CP 是屬於比較低層的，它的數量及位置是要符合現場的情況來設置的。除了控家裡設備的功能外，它也必須要能與外界溝通的能力，藉此來接收控制命令，並依不同的命令執行不同的控制邏輯進而產生不同的環境效果。

#### 2. 主控制器(Smart Room Controller)：

主要功能是用來管理控制所有的 Control Point，依照不同的生理訊號處理後的結果，指揮適當的 CP 來驅動環境設備，進而達到控制居家環境提供不同條件的環境。

#### 3. 網路設備：

因主控制器及設備控制點分散於不同的位置，所以需要一個網路環境來將這些設備結合在一起。可以適用的網路設備如有線網路(RJ-11)，無線網路(WiFi)，以及電力線網路。這些必須要參照現場實際的情況來選擇，也有可能是三者都會用到，詳細在後面的章節會更進一步的介紹。

### 4.3.2 主控制器

主控制器(Smart Room Controller):主要功能為接收及時生理訊號處理器處理完後的結果，將這些結果做一個完整的分析處理後，決定要驅動哪些環境裝置後，將控制訊號傳給環境裝置來控制環境。

### 4.3.3 環境裝置控制器

環境裝置控制器(Control Point, CP):接收主控制器送來的控制訊息，解譯後驅動環境裝置達到環境控制之目的。其環境控制器的系統架構如圖 4.5 所示。

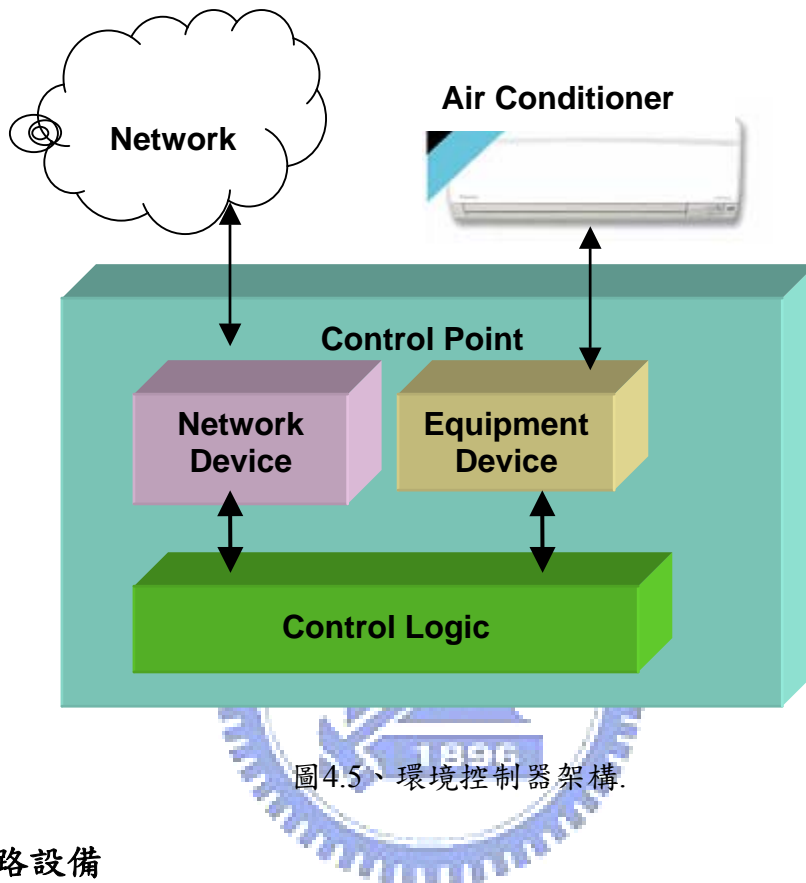


圖4.5、環境控制器架構.

### 4.3.4 網路設備

現今家庭網路的建購成本以非常低廉，網路普及到各家庭中，所以利用網路來聯居家境中的設備是選擇。以目前的網路設備及建構的方式不同大致上可以區分為三類，有線網路，WiFi 無線網路，以及電力線網路(Home plug)，以下分別說明這四種網路架構。

1. 有線網路: 以 RJ-45 網路線來連接設備。抗干擾能力強，具有較高的穩定性。但由於是有線的所以有佈線困難的問題，需要事先規劃好路徑。
2. WiFi (Wireless fidelity)無線網路，是 IEEE 定義的一個無線網路通信的工業標準(IEEE 802.11)。Wi-Fi 的設置至少需要一個 Access Point (AP)，透過這個 AP 將兩個以上的 WiFi 設備聯接起來。WiFi 是利用空氣為媒介所以不需要佈線可以改善有線網路的缺點，但其傳輸速度比有線網路低。

4. 電力線網路(Home plug) :是指利用電力線傳輸數據信號的一種通信方式。不需要重新牽設網路線，在現有電力線系統上收發或傳輸數據。終端用戶只需要插上電源插頭，就可以實現上網。利用現有的電力線資源，節省費用。

### 4.3.5 溝通機制

Smart Room Controller 與 Control Point 間透過網路來聯繫，聯繫的訊息傳係方式與時間我們規劃 2 個主要的訊息 DISCOVERY 以及 COMMAND。

#### 1. DISCOVERY

用來達成 Smart Room Controller 與 Control Point 連線使用。依時間發生的順序可以分成 Smart Room Controller 先啟動及 Control Point 先啟動兩種。當 Smart Room Controller 先啟動時如圖 4.6 所示，會發出 DISCOVERY 訊息到網路上，但因為 Control Point 尚未啟動，故 Smart Room Controller 不會收到 DISCOVERY\_R 的訊息，待 Control Point 啟動後會主動發出 DISCOVERY\_R 的訊息告訴 Smart Room Controller 有關 Control Point 的一些訊息。反之當 Control Point 先啟動時，Control Point 會主動發出 DISCOVERY\_R 的訊息到網路上，但因為 Smart Room Controller 尚未啟動，所以此訊息是無效的，待 Smart Room Controller 啟動後，Smart Room Controller，Control Point 收到後，會回 DISCOVERY\_R 的訊息告訴 Smart Room Controller 有關 Control Point 的一些訊息，如圖 4.7 所示。如此 Smart Room Controller 可以掌握可以控制的 Control Point。

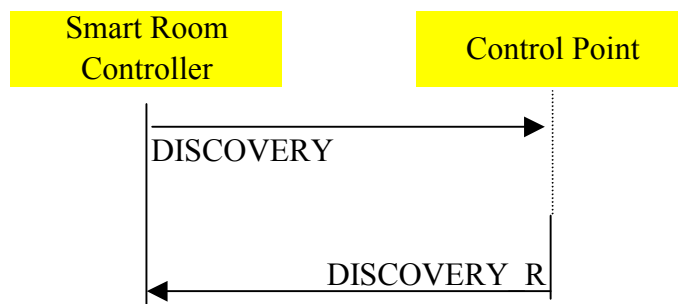


圖4.6、Smart Room Controller先啟動時的通訊模式

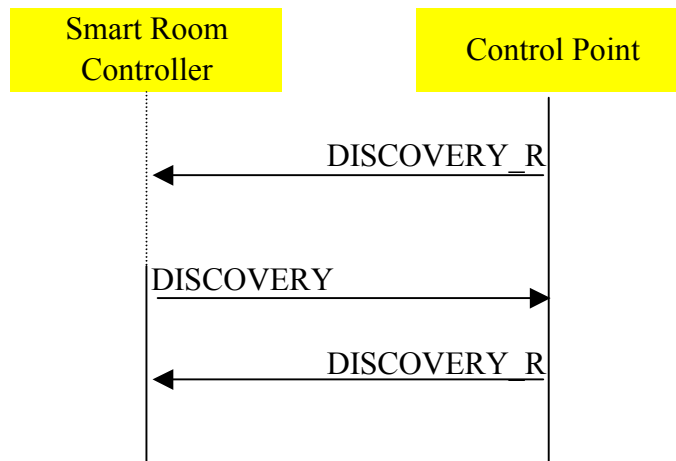


圖4.7、Control Point先啟動時的通訊模式

底下是一個 Smart Room Controller 與 Control Point 訊息溝通的例子：

Smart Room Controller 發出 DISCOVERY 的訊息到網路上，其中 Device 欄位填入表示要求所有 Control Point 回覆。

MessageID: DISCOVERY

DeviceName: 空白

當一個位於臥室編號01具有燈光控制的 Control Point 收到 Smart Room Controller 發出的 DISCOVERY 的訊息後會回覆以下的訊息給 Smart Room Controller。說明此 C.P. 具有 On/Off 功能的燈光控制器，目前是狀態是 ready。

MessageID: DISCOVERY\_R

DeviceName: A000000000000000000001

Location: BEDROOM001

Service: Light\_Control

Status: Online

Action#1: On

Action#2: Off

## 2. COMMAND

當 Smart Room Controller 需要控制環境設備時可以用此訊息來下達控制命令給 Control Point，藉由 Control Point 來控制環境設備來達到目的，如圖 4.8 所示。

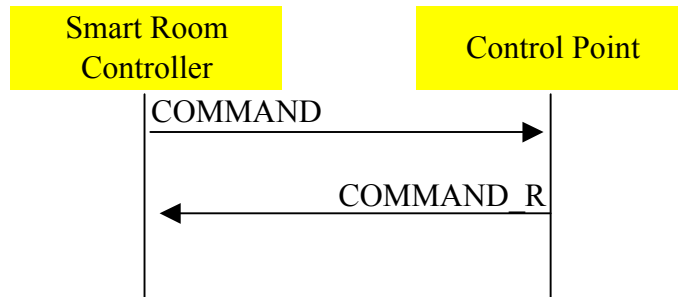


圖4.8、命令的通訊模式

底下是一個 Smart Room Controller 要求 Control Point 打開燈光，Control Point 回覆執行狀態的例子：

Smart Room Controller -> Control Point

MessageID: COMMAND  
DeviceName: A0000000000000000001  
Action: On

Control Point -> Smart Room Controller

MessageID: COMMAND\_R  
DeviceName: A0000000000000000001  
Ack: 0

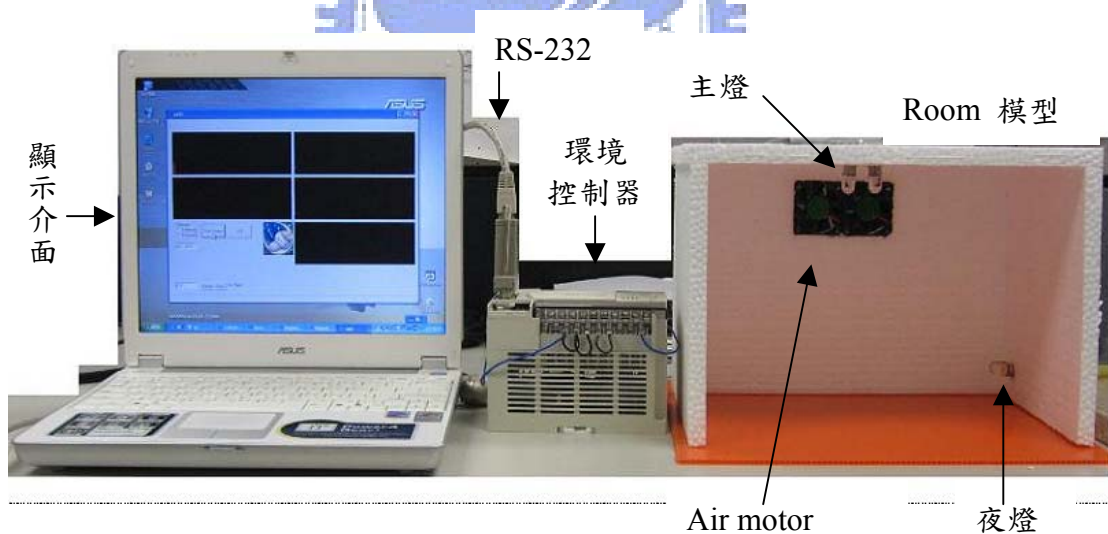
## 五、 測試結果與討論

### 5.1 系統測試結果

本章將針對整個系統的實現做一些測試、比較與討論就系統整合的部分的進行說明。本論文所完成的系統如圖 5.1 所示，包括無線電生理訊號量測模組、嵌入式生理訊號分析系統、顯示介面以及環境控制器。分別說明如下：



(a)系統實體圖



(b)環境控制之 SmrtRoom 模型

圖5.1、系統實體圖.

### 1. 無線電生理訊號量測模組：

如圖 5.2(a)左邊所示，包含電極、EEG 放大器、RF 無線傳輸模組。主要的功能為透過電極擷取 EEG 訊號，經 EEG 放大器濾波，將類比訊號轉成數位訊號，最後經由 RF 無線傳輸模組將 EEG 訊號傳給嵌入式生理訊號分析系統。

### 2. 嵌入式生理訊號分析系統：

如圖 5.2(a)右邊所示，包含 Embedded 生理訊號處理器、RF 無線傳輸模組。主要的功能為將 RF 無線傳輸模組收到的 EEG 訊號做訊號處理，同時將 EEG 訊號透過網路傳給顯示介面，EEG 訊號處理完後的結果也透過網路傳給 Smart Room Controller，作為環境控制的依據。

### 3. 顯示介面

如圖 5.1(b)所示，用來顯示 EEG 訊號及經過處理後的生理狀態結果。

### 4. 環境控制器

如圖 5.1(b)所示，包含 Smart Room Controller，以及一個 Control Point。Smart Room Controller 用一部電腦來模擬，Control Point 則用 PLC 結合電腦來組成。

### 5. 環境設備模型

如圖 5.1(b) 所示，模型包含 3 個燈光以及 2 個風扇等這兩個環境設備。燈光設備包含頂端的兩個主燈及底部的一個夜燈，兩個風扇設備可以獨立運作，用來模擬通風及溫度控制。



## 5.2 多工排程效能比較

如果不使用多工排程，如圖 5.2 中所示，其最大的問題就在於如果 Process A 必須每隔一段時間就必須執行一次然後執行 Process B，換言之，就是執行的時間限制非常嚴格，但由於必須要執行 Process B，所以會造成 Process A 會有延遲的現象產生。解決這個問題的方法有兩個：一是自行規劃與切割，使得 Process B 的執行時間不會干擾到 Process A 的執行頻率，另一個方法則是交由系統做多工排程。

如果使用多工排程則可以將原本比較大塊的處理程序分解成許多小塊的處理程

序，如圖 5.3 中所示。如此一來，將可以準確的確保程序 A 可以符合其執行的頻率，而這也是我們使用多工排程的原因。因為，我們必須讓接收無線模組訊號的處理程序一定頻率便執行一次，以便確保其取樣頻率的準確度。

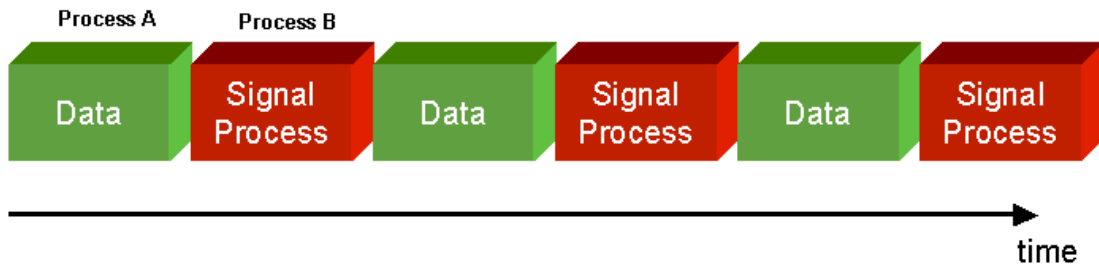


圖5.2、Single task 程序流程示意圖

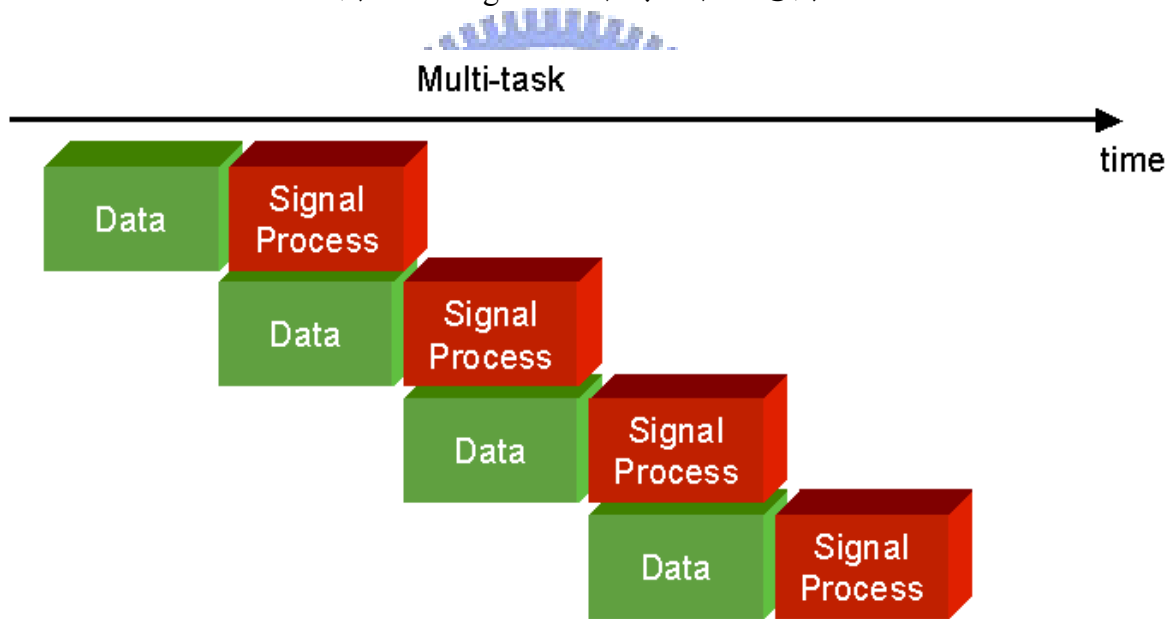


圖5.3、Multi-task 程序流程示意圖

在比較多工排程(Multi-task)與非多工排程(Single-task)的效能上，我們分別說明這兩者的工作情形

#### 1. 非多工排程(Single-task)

工作的分配將採逐一執行 task 方式進行，如圖 5.4 所示,RF 接收資料完後，進行資料/訊號處理，待資料處理後再繼續下一個流程，如此一直重複下去。



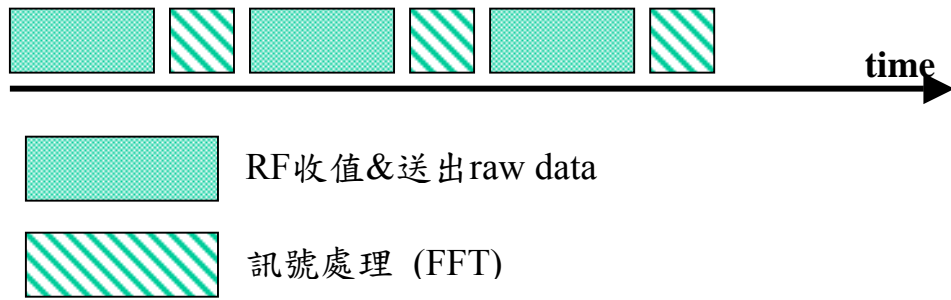


圖5.4、Single-task 的工作流程

## 2. 多工排程(Multi-task)

在多工排程上將工作分成，task 管理(task\_mgr)、RF 接收/網路管理(rf\_mgr)、資料暫存管理(db\_mgr)、以及訊號處理(sp\_mgr)，這四個部分。整個工作流程如圖 5.5 所示。

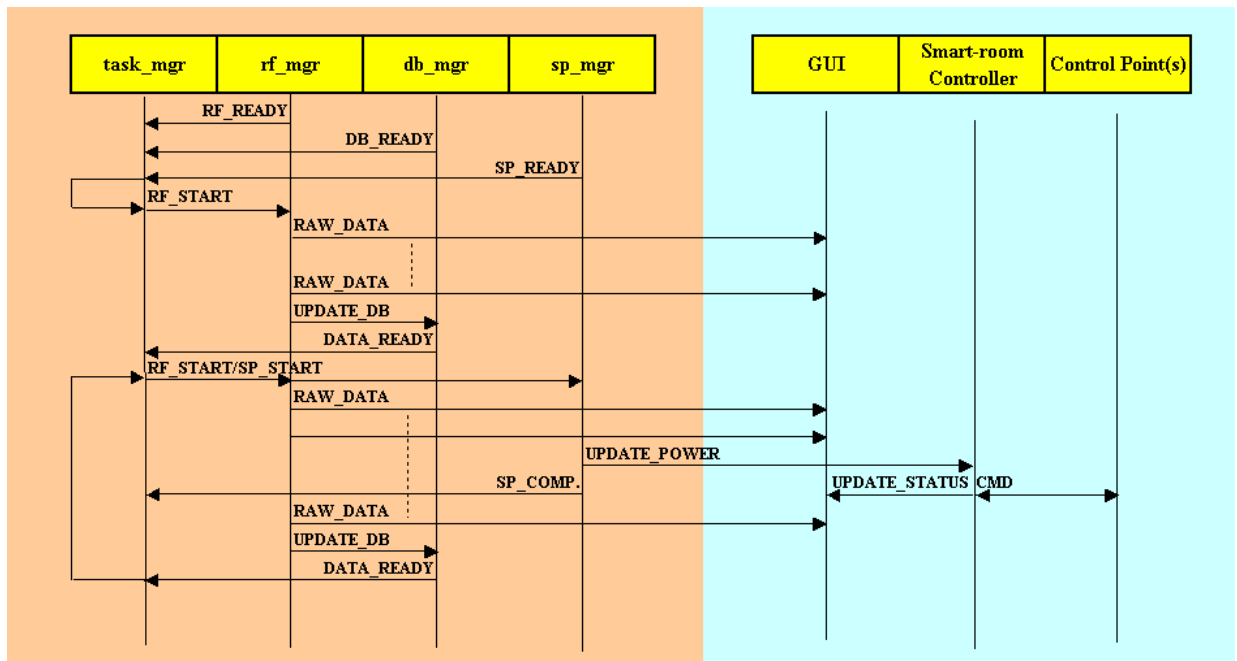


圖5.5、Multi-task 的工作流程

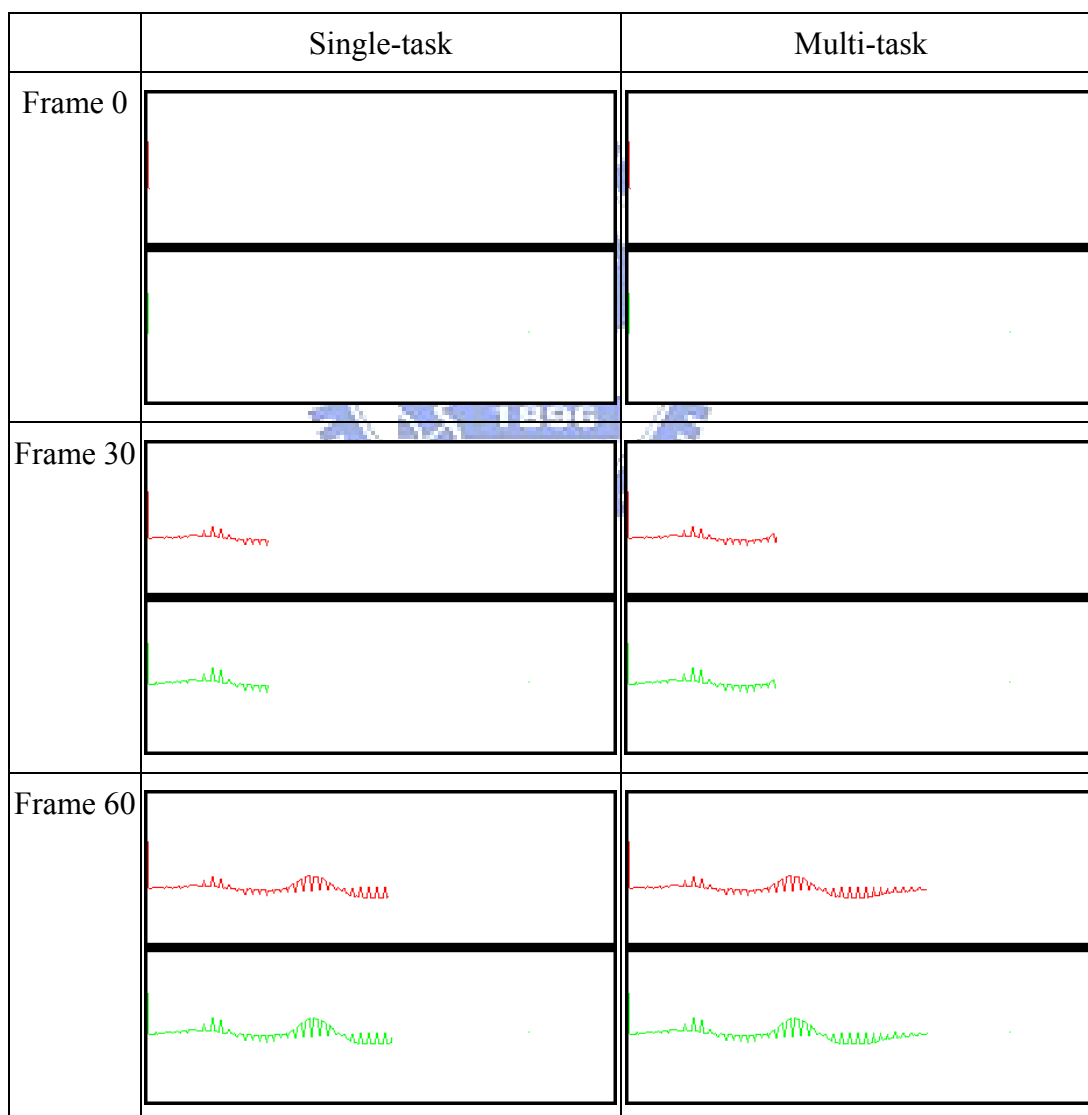
若以四個頻道的 EEG 資料來作為 data，並以接收足夠資料與資料處理為一次完整的流程，執行這個流程 1000 次，分別比較使用多工排程與不使用多工排程兩種方法所需的時間。如表 5.1 所示，不使用多工排程的程式做完 1000 次的資料處理循環需要 2589 秒，而使用多工排程的情形下只需 2008 秒，效能約提升 22.44%。所以，把時間拉長來看，多工排程的效果將會越明顯。

表 5.1、使用多工排程與不使用多工排程的效能比較

|             | 接收資料   | 不使用多工排程 | Share memory | Socket (this work) |
|-------------|--------|---------|--------------|--------------------|
| 執行 1000 次時間 | 2002 秒 | 2589 秒  | 2097 秒       | 2008 秒             |
| 跨平台         | ---    | N       | N            | Y                  |

另外由於 single-task 無法平行處理接收資料與訊號處理故在接收訊號上會有延遲的現象而 multi-task 則不會類似的情形。由表 5.2 可以看出 single-task 在接收一段資料後會因為進行訊號處理而暫停接收訊號，而 multi-task 則不會。

表 5.2、使用多工排程與不使用多工排程的延遲比較



### 5.3 多重服務測試

我們設計一個跨平台的 client/server 的程式,並比較 client 對單一 server 與 client 對多個 server 的效能，圖 5.6 表示兩者的系統架構。

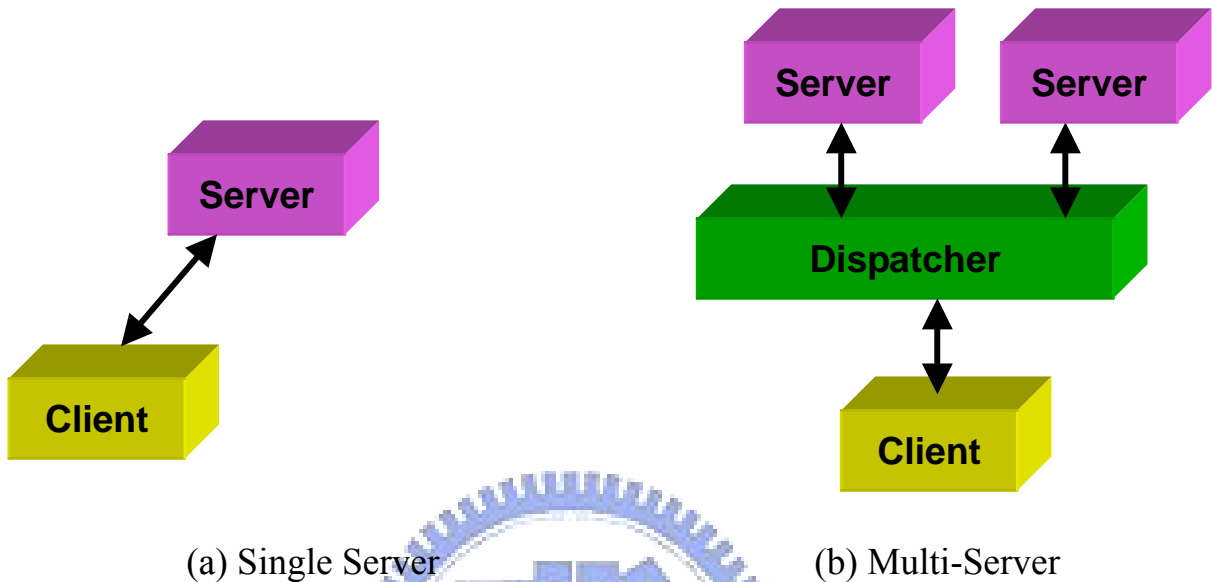


圖 5.6、Single Server 與 Multi-Server

server 處理每一個 client 的請求需要 2 秒的時間，而 client 每 1.5 秒發出一個請求給 server。single server 在處理 client 的請求時由於 server 處理的時間大於 client 發出請求的時間間隔，所以無法即時的處理 client 的需求。為了解決此問題我們增加一個 server 來處理 client 的請求，並在 client 與兩個 server 間增加一個 dispatcher 來管理 server 與 client 的請求，當由兩個 server 來處理 client 時因為這兩個 server 可以分別獨立作業故可以同時處理兩個 client 的請求。由表 5.3 可以看出來 single server 因為處理的時間大於 client 的需求間隔，需要更多的時間來處理。使用兩個 server 時因可以同時處理兩個 client 的請求，故時間上幾乎不會有延遲的現象

表 5.3、使用 Multi-Server 與 Single Server 的比較

|             | Client 需求 | Single Server | Multi Servers |
|-------------|-----------|---------------|---------------|
| 執行 1000 次時間 | 1499 秒    | 2004 秒        | 1500 秒        |

## 5.4 系統整合與討論

我們已於 5.2 中比較過使用多工排程與不使用多工排程在執行時間上的差異，由測試結果可以發現在時間效能上，使用多工排程的執行時間與延遲時間都縮短許多。另外於 5.3 中比較使多個服務與單一服務的差異，由測試結果可以發現在時間效能上多個服務能提供較即時的處理效能。除了縮短時間外，使用多工排程具有以下幾項優點：1.使程式模組化，將程式分離可以縮短程式規劃的時間。此外，由於程式模組化，所以，在開發時間較短，也有助於多種不同功能程式的整合。模組化的程式設計亦有助於程式的再次使用。2.讓系統在週邊裝置的控制上更即時，由於整套資料接收與處理的流程都必須即時，我們以多工排程的方法來克服，以讓整個系統能準確符合資料接收所需要的頻率。3.讓系統更有效利用系統資源，在許多排程的研究上，就是為了能善用系統的資源，本論文所發展的系統亦有這個優點，在實現上可以減少系統等待時間時的浪費。



## 六、 結論與未來展望

本論文提出一套加強嵌入式系統使用效率的多工排程系統實現在無線嵌入式腦機介面系統，並將其應用於居家環境設備控制上。為了加強訊號接收的即時性，引入了多工排程來加強訊號接收的即時性，而在第五章的測試中，我們也發現訊號接收更為即時連續。這兩點也證明了本論文所提出與實現的無線嵌入式腦機介面系統具即時訊號接收與處理的能力。

雖然，在本論文已完成的即時無線腦機介面系統在各方面都有不錯的水準，有許多的功能都是在相關研究中所沒有的。但是，我們仍有些進步的空間需要作為日後，主要改善方向如下：

### 1. 系統體積：

本論文實現了一套無線嵌入式腦機介面系統的雛型，但整體的體積還是太過於龐大，不適用於使用著攜帶。無論是生理訊號放大電路或著是嵌入式處理單板都還有縮小的空間，例如：生理訊號放大電路可以使用類比積體電路的方式來實現。而在嵌入式單板上，可以朝縮小體積與降低功率消耗的目標進行改善，以提升其實用價值。

### 2. 多工排程系統：

在多工排程系統上，本論文提出一套非常良好的程式發展平台，有助於程式的開發，在我們系統中也非常的有效。但是，未來可以在我們所發展的平台開發一些新的排程演算法，使其在工作量塞滿處理器時亦能負荷。

### 3. 更多樣性的生理訊號量測：

人體是精密的組合，唯有結合更多的生理訊號，提供更多樣的人體即時狀態才能供後端來精確地環境控制。

### 4. 研發更多的環境設備控制器

發展更多樣性的環境設備與醫療設備結合介面，提供患者優質的病房照顧環境和更完善的居家環境。

### 5. 電池的使用時間

本論文所使用的腦波放大器是由電池供應，由於目前的電路設計限制電池只能使用約 3 個小時，並不足以供現實環境的使用需求。故未來可以朝電路縮小化、使用低耗電量的元件、以及使用更高效率的電池著手，來提高系統的使用時間。

隨著這一套系統的誕生，我們看到了『未來生活環境』，也希望有朝一日可以將這套系統應用於日常生活中，使得未來的環境更能符合人類的需求。未來我們將努力讓這套系統更加人性化，為人類帶來更多進步與舒適的生活。



## 參考文獻

- [1] <http://ibru.vghtpe.gov.tw/chinese/eeg.htm>
- [2] <http://ibru.vghtpe.gov.tw/chinese/bci.htm>
- [3] Ming Cheng, Xiaorong Gao, Shangkai Gao, and Dingfeng Xu, "Design and implementation of a brain-computer interface with high transfer rates," *IEEE Trans. on biomedical Eng.*, vol. 49, no. 10, pp. 1181-1186, 2002.
- [4] Obermaier B, Design and implementation of an EEG Based Virtual Keyboard using Hidden Markov Models, Ph.D. dissertation, Technical University-Graz, Graz, Austria, 2001.
- [5] Obermaier B, Neuper C, and Pfurtscheller G, "Information transfer rate in a five-classes brain-computer interface," *IEEE Trans. on Rehab. Eng.*, vol. 9, pp. 283-288, Sept. 2001.
- [6] Xiaorong Gao, Dingfeng Xu, Ming Cheng, and Shangkai Gao, "A BCI-Based Environmental Controller for the Motion-Disabled," *IEEE Trans. on Neural Syst. and Rehab. Eng.*, vol. 11, no. 2, pp. 137-140, June 2003.
- [7] G. Edlinger, G. Krausz, F. Laundl, I. Niedermayer, C. Guger, "Architectures of Laboratory-PC and Mobile Pocket PC Brain-Computer Interfaces," *Proceedings of the 2 International IEEE EMBS Conference on Neural Engineering*, Arlington, Virginia, March 16 - 19, 2005.
- [8] Ashwin K. Whitchurch, B. Han Ashok, R. Vinod Kumaar and K. Sarukesi, and Vijay K. Varadan, "Wireless system for long term EEG monitoring of Absence Epilepsy," *Biomedical Applications of Micro- and Nanoengineering, Proceedings of SPIE*, vol. 4937, pp. 343-349, November 2002.
- [9] Chin-Teng Lin, Ruei-Cheng Wu, Sheng-Fu Liang, Te-Yi Huang, Yu-Chieh Chen,; Tzyy-Ping Jung, "Estimating driving performance based on EEG spectrum analysis," *EURASIP Journal on Applied Signal Processing*, vol. 2005, no. 19, pp. 3165-3174, Mar. 2005.
- [10] Chin-Teng Lin, Ruei-Cheng Wu, Sheng-Fu Liang, Wen-Hung Chao, Yu-Jie Chen, Tzyy-Ping Jung, "EEG-based drowsiness estimation for safety driving using



- independent component analysis,” *IEEE Transactions on Circuits and Systems I*, vol. 52, no. 12, pp. 2726-2738, Dec. 2005.
- [11] Young Han Nam, Zeehun Halm, Young Joon Chee, Kwang Suk Park, “Development of remote diagnosis system integrating digital telemetry for medicine,” *Proceedings of the 20th Annual Znternational Conference of the IEEE Engineering in Medicine and Biology Society*, vol. 20, no. 3, pp. 1170- 1173, 1998.
- [12] Wen-Chung Kao, Wei-Hsin Chen, Chun-Kuo Yu, Chin-Ming Hong, and Sheng-Yuan Lin, “A real-time systemfor portable homecare applications,” *Consumer Electronics, 2005*, pp. 369- 374, June 14-16, 2005.
- [13] F. Lamberti, C. Demartini,“ Low-Cost Home Monitoring Using a Java-Based Embedded Computer,” *Proceeding of the 4th Annual IEEE Conf. on Information Technology Applications in Biomedicine, UK*, pp 342-345.
- [14] 林琨皓，“即時擷取與分析生醫信號之可攜式記錄器:肌電圖之應用”，臺灣大學/醫學工程學研究所，碩士論文，民國九十二年。
- [15] 胡維賓，“應用於 ZigBee 生醫無線系統之無線收發器”，國立成功大學/電機工程學系，碩士論文，民國九十四年。
- [16] 周宗哲，“可穿戴式無線多參數心肺生理訊號網路遠距量測系統”，臺灣大學/醫學工程學研究所，碩士論文，民國九十四年。
- [17] 陳昱誌，“應用於生醫訊號處理之類比前端積體電路設計”，國立中正大學/電機工程所，碩士論文，民國九十四年。
- [18] 黃嘉淵，“3G 行動遠距醫療平台應用於生醫訊號傳輸之研究”，中原大學/電子工程研究所，博士論文，民國九十二年。
- [19] 張正文，“可程式化植入式微電刺激器積體電路與系統設計”，中原大學/電子工程研究所，碩士論文，民國九十二年。
- [20] <http://www.taconet.com.tw/psychiat>
- [21] [http://www.edah.org.tw/ped/web\\_inter/teaching/sleep.htm](http://www.edah.org.tw/ped/web_inter/teaching/sleep.htm)
- [22] [http://www.taiwanheadache.com.tw/teach\\_16.asp](http://www.taiwanheadache.com.tw/teach_16.asp)
- [23] 謝弘義，“具備多工排程功能之無線嵌入式腦機介面系統及其在即時汽車駕駛員疲勞狀態偵測與提醒之應用”，國立交通大學，碩士論文，民國九十五年。
- [24] <http://faculty.washington.edu/chudler/1020.html>
- [25] <http://www.ancher.com.tw/rf3105.html>
- [26] 淡江大學智慧型控制實驗室，“An Introduction Altera CPLD/FPGA & VHDL”，民

國九十一年

[27] 陳慶逸、林柏辰，”VHDL 數位電路實習與專題設計”，文魁資訊股份有限公司，民國九十二年。

[28] W. Richard Stevens, Advanced Programming in the UNIX Environment, Addison-Wesley, June 1992.

[29] GNU Compiler Collection, <http://gcc.gnu.org>

[30] Code Composer Studio Getting Started Guide Rev C(SPRU509C), Texas Instruments.

[31] Toshihiro Kobayashi, Kiyotaka Takahashi, Linux DSP Gateway Specification Rev2.0, Nokia Corporation, November 13, 2003.



## 附錄一、縮寫對照(按字母排序)

| 縮寫   | 原文                                     | 中文翻譯     |
|------|--|----------|
| A/D  | Analog to Digital                      | 類比數位訊轉換  |
| API  | Application Program Interface          | 應用程式介面   |
| ARM  | Advance RSIC Machines                  |          |
| BCI  | Brain Computer Interface               | 大腦人機介面   |
| CPLD | Complex Programmable Logic Device      |          |
| DMA  | Direct Memory Access                   | 直接記憶體存取  |
| DSP  | Digital Signal Processor               | 數位訊號處理器  |
| ECG  | electrocardiogram                      | 心電圖      |
| EEG  | electroencephalogram                   | 腦電波      |
| EMG  | electromyogram                         | 肌電圖      |
| EOG  | electro-oculogram                      | 眼電圖      |
| FHSS | Frequency Hopping Spread Spectrum      | 跳頻展頻     |
| FPGA | Field Programmable Array               |          |
| FSK  | Frequency Shift Keying                 | 頻率鍵移     |
| GCC  | GNU Compiler Collection                |          |
| GSM  | Global System for Mobile Communication | 全球行動通訊系統 |

| 縮寫   | 原文                                    | 中文翻譯        |
|------|---------------------------------------|-------------|
| GPRS | General Packet Radio Service          | 通用分組無線服務    |
| HCI  | Host Control Interface                | 主機控制介面      |
| IDE  | Integrated Development Environment    | 整合發展環境      |
| IP   | Internet Protocol                     | 網際網路通訊協定    |
| IPC  | inter-process communication           | 處理程序之間的溝通機制 |
| MMU  | Memory Management Unit                | 記憶體管理單元     |
| MPU  | Micro Processor Unit                  |             |
| OMAP | Open Multimedia Architecture Platform |             |
| OS   | Operating System                      | 作業系統        |
| PDA  | Personal Digital Assistant            | 個人數位助理      |
| PLC  | Programmable Logic Controller         | 可程式邏輯控制器    |
| RF   | Radio Frequency                       | 射頻          |
| RISC | Reduce Instruction Set Computer       | 精簡指令集電腦     |
| SDP  | Service Discovery Protocol            | 服務發現協定      |
| SoC  | System on Chip                        | 系統晶片        |
| SRAM | Static Random-Access Memory           | 靜態隨機存取記憶體   |

| 縮寫    | 原文  | 中文翻譯      |
|-------|---|-----------|
| STFFT | Short-Time FFT                              | 短時快速傅立葉轉換 |
| TCP   | Transmission Control Protocol               | 傳輸控制協定    |
| TCS   | Telephone Control Service                   | 電話傳送控制協定  |
| UART  | Universal Asynchronous Receiver Transmitter | 通用異步收發器   |
| USB   | Universal Serial Bus                        | 通用串列匯流排   |



## 附錄二、簡介 OMAP 5912

OMAP 是德州儀器公司推出一項由 RISC(Reduce Instruction Set Computer，精簡指令集電腦)整合 DSP 的 SoC(System on Chip，系統晶片)嵌入式系統微處理器，其應用定位主要是在於即時的多媒體影音資料處理、語音辨識系統、網際網路通訊及無線通訊等訊號分析應用。OMAP 的最大特點在整合了 TI DSP 核心及 ARM 的 RISC 架構和各種週邊控制器的設計，而這也是本論文利用 OMAP 的原因，我們需要 ARM 對周邊控制以及支援嵌入式作業系統的能力，也需要數位訊號處理器的數學運算能力。



圖B.1、OMAP 5912 OSK外觀

資料來源: Start Kit for the OMAP 5912

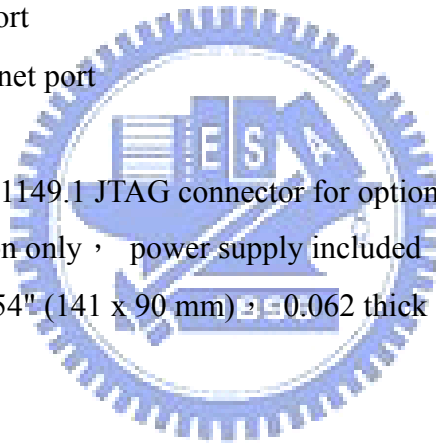
### B.1 嵌入式硬體開發環境

在硬體環境部分本節將介紹 OMAP 5912 處理器與使用的平台;OMAP 為 Open Multimedia Architecture Platform 的縮寫，是一個有雙核心的微處理器，結合 ARM9 微處理器與 TMS320C55x DSP 核心。OMAP 5912 OSK 為 TI OMAP 處理器的手持式可展開的 OMAP 開發平台。OMAP 是德州儀器公司近年來推出一項由 RISC(Reduce Instruction Set Computer，精簡指令集電腦)整合 DSP 的 SoC(System on Chip，系統晶片)嵌入式系統微處理器，其應用定位主要是在於即時的多媒體影音資料處理、語音辨識系統、網際網路通訊及無線通訊等訊號分析應用。OMAP 的最大特點在整合了 TI

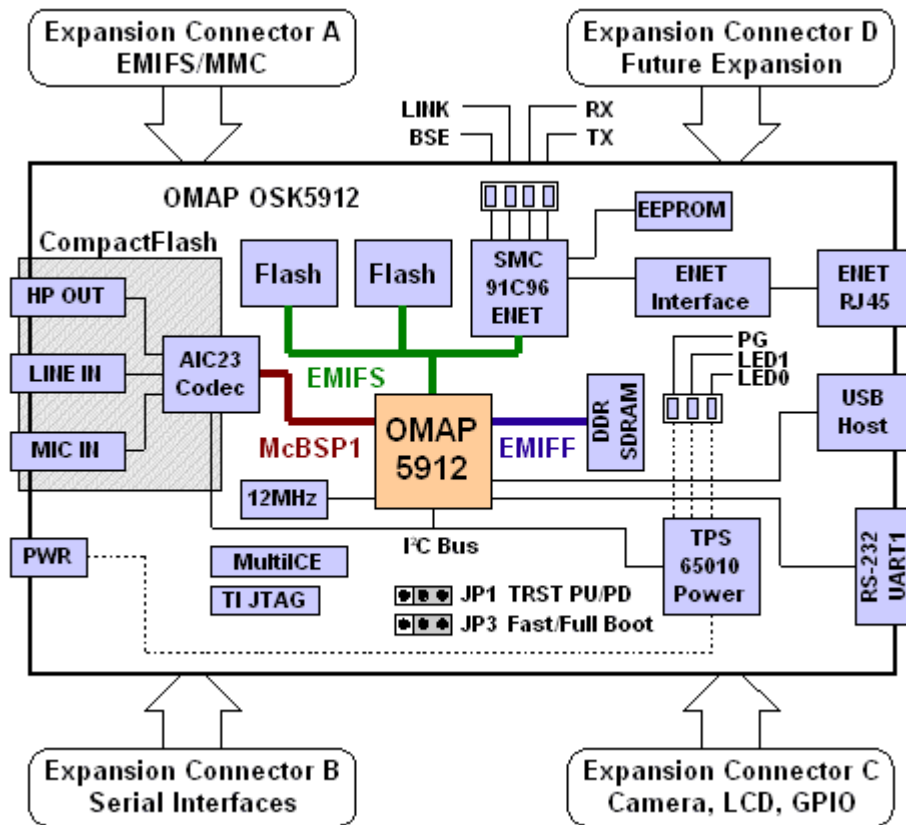
DSP 核心及 ARM 的 RISC 架構和各種週邊控制器的設計，而這也是本論文利用 OMAP 的原因，我們需要 ARM 對周邊控制以及支援嵌入式作業系統的能力，也需要數位訊號處理器的數學運算能力。

TI 的 OMAP 5912 OSK 硬體結構圖，如圖 B.2 所示，OMAP 5912 OSK 的主要特性如下(Hardware Features):

- ◆ Texas Instruments TMS320C55xx core operating at 192 Mhz。
- ◆ ARM9 core operating at 192 Mhz。
- ◆ TLV320AIC23 codec
- ◆ 32 Mbyte DDR RAM
- ◆ 32 Mbyte on board Flash ROM
- ◆ 4 Expansion connectors (bottom side)
- ◆ RS-232 serial port
- ◆ 10 MBPS Ethernet port
- ◆ USB port
- ◆ On board IEEE 1149.1 JTAG connector for optional emulation
- ◆ +5 Volt operation only , power supply included
- ◆ Size: 5.55" x 3.54" (141 x 90 mm) , 0.062 thick , 8 layers





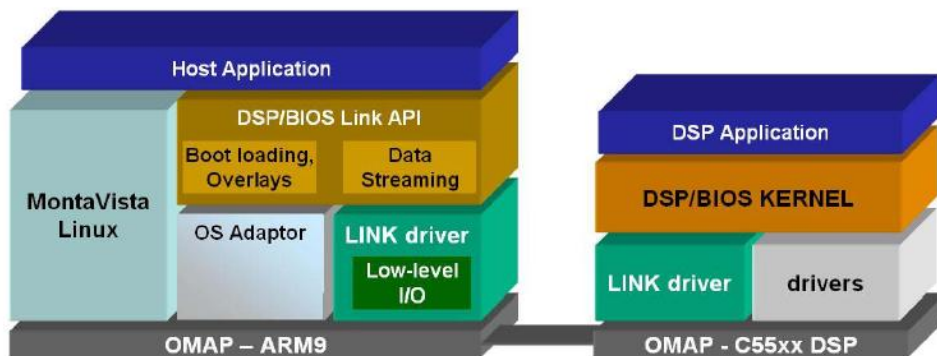


圖B.2、OMAP 5912 OSK 架構圖

資料來源: Start Kit for the OMAP 5912

## B.2 嵌入式軟體開發環境

在前面將 OMAP 微處理器硬體和環境選擇簡單的說明之後，接下來要進入軟體應用的開發環境部分。因為 OMAP 是由 ARM 和 DSP 兩種架構所組成的，如圖 B.3 所示，所以在軟體開發上，我們分別針對 ARM 和 DSP 的程式開發環境做介紹。



圖B.3、OMAP 雙核心系統架構

資料來源: Start Kit for the OMAP 5912

### (1) 程式開發環境於ARM架構

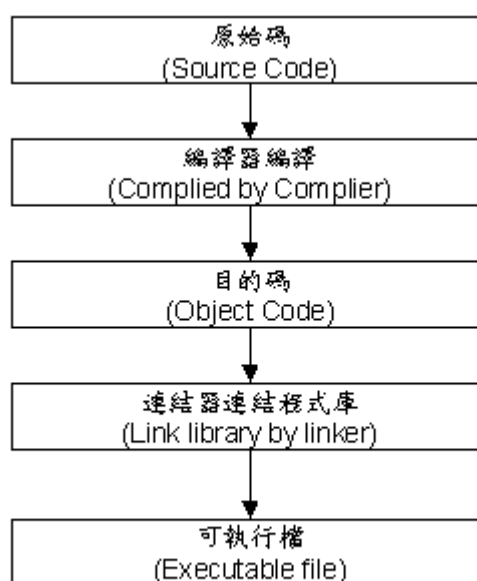
自從 1991 年 Linux 由 Linus Torvalds 將第一版 Linux 公開在網路上後，這套作業系統已經受到世界普遍的歡迎。現在的 Linux 支援相當多的處理器核心，由高階的伺服器、工作站，到常見的個電腦中都有它的蹤跡，甚至是一些手持式裝置例如手機、PDA(Personal Digital Assistant，個人數位助理)等一些現代人常用的嵌入式系統上都看得到 Linux 的存在。

所謂的嵌入式系統指的就是為了提高產品的穩定性、可攜性，或者是降低產品成本、增加特定功能等不同的目的，對一般的標準系統進行修改以符合特殊設計上的需求。在硬體方面靠的主要就是被稱為系統單晶片(SoC)的半導體整合技術，而嵌入式作業系統則是軟體方面的技術核心。Linux 的可靠、高效能、伸縮性、低成本，以及最重要的握有控制權等幾項特點，使得 Linux 在嵌入式作業系統上站在非常有利的地位。

在一般 PC 的 Linux 的環境下開發程式，最常使用到的編譯器，莫過於 GCC (GNU Compiler Collection)[29]，GCC 可以說是 Richard Stallman 所創立的 GNU 計畫中最重要的作品之一，它提供了自由軟體世界高品質的編譯器 (compiler)，實現了我們在自由軟體平台上開發程式的夢想。GCC 一個很大的特色是高度可移植性，目前已知有包括了：x86、arm、ia64、alpha、hppa、m68k、Power PC、mips 等超過三十種硬體平台與作業系統可以執行 GCC，此高度可移植性正是 GCC 廣為流傳散佈的

主要原因。儘管 GCC 是自由軟體計畫開發出來的，但其所編譯出來的程式品質並不輸給商業版的編譯器，甚至所編譯出來程式在特定平台有更好的執行效能。

嵌入式系統開發也需要用到編譯器(Compiler)、連結器(Linker)、組譯器(Assembler)，整合開發環境。以 ARM 而言，在 x86 環境下編譯產生出來的執行檔，並不能於 ARM 環境執行，若要在 x86 環境下開發 ARM 程式則需要建立一個跨平台開發的工具 (cross-platform development tool)。arm-linux-gcc 可將 x86 環境下程式碼針對 ARM 的架構進行交叉編譯 (cross-compiler)，產生出來的執行檔便可以於 ARM 的架構上執行。GCC 編譯流程如圖 B.4 所示。



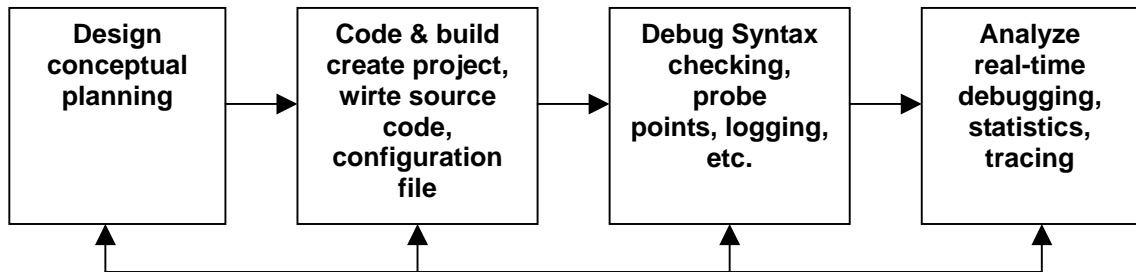
圖B.4、GCC編譯流程圖

## (2) 程式開發環境於DSP架構

開發 DSP 程式有兩種方式 1.是採用德州儀器所提供的軟體 CCS (Code Composer Studio)[30][31]， 2.是採用 Linux DSP Tools 。

第一種選擇 CCS，提供一個完善的整合發展環境(IDE，Integrated Development Environment)，包含了程式編譯器、程式除錯器以及模擬器。CCS 以一致的環境來整合所有 host 與 target 工具，包括 TI 的 DSP/BIOS kernel、code-generation tools、fast simulators、debugger、與 Real-Time Data Exchange (RTDX) 技術，簡化應用程式的開發。利用這些工具可以簡單地開發出 OMAP 應用程式。

圖 B.5 為 CCS 開發 DSP 程式流程。在設計程式前，先有個設計概念，再開始建立專案來編寫程式碼，編寫過程中如遇到錯誤時需要除錯時，CCS 可以用來對程式編碼做除錯以及即時的分析，甚至整個程式執行過程都可以追蹤。



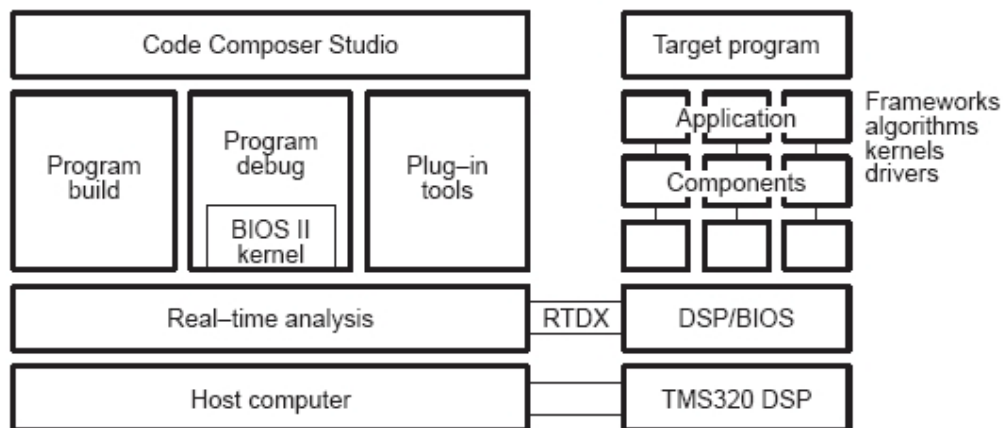
圖B.5、CCS程式發展設計流程

利用 CCS 開發 DSP 程式時，要先做好 ARM 與 DSP 於系統上的配置以及設定 CCS 編程的環境記憶體配置，如此才能跨平台開發程式。而 DSP 的編譯先是由 C/C++ 的原始碼經過 C/C++ 的編譯器後，可產生組合語言原始碼。再經過組譯器(Assembler)產生物件(Object)檔，經由連結器(Linker)來產生最後的可執行 COFF(Common Object File Format)檔(以.out 為副檔名)。

第二種選擇，Linux DSP Tools，提供 compiler，GNU，Tconf 來建立 DSP 應用程式。

DSP/BIOS 核心是一個事件驅動(Event-Driven)的架構，DSP 的應用程式透過 DSP/BIOS Configure 對於 DSP/BIOS 內的機制作設定。DSP/BIOS 系統配置工具是用來幫助開發者在撰寫事件處理函式時，設定事件的特性與參數，如硬體中斷、軟體中斷、背景執行緒、工作 (TASK) 等來使用 DSP/BIOS 的多執行緒能力。

DSP/BIOS 主要被設計出來針對某些應用層面，其中包含：即時排程(Real-Time scheduling)、同步(Synchronization)，以及 HOST 和 TARGET 之間的通訊(如圖 B.6 所示)。DSP/BIOS 核心提供了 1. 強制性的多執行緒(Preemptive Multi-Thread)，也就是在 DSP 中有許多 Task 時，優先權高的工作可以將優先權低的工作中斷，並搶得執行權。2. 硬體抽象化(Hardware Abstraction)，即完全由硬體抽象層控制系統的硬體中斷。使用者可以利用硬體抽象層給予的硬體描述來控制硬體，3. 即時分析(Real-Time Analysis)的功能。除此之外，DSP/BIOS Configuration 工具可將程式的大小作最佳化處理。圖中表示了 DSP/BIOS 在 CCS 上所扮演的角色。



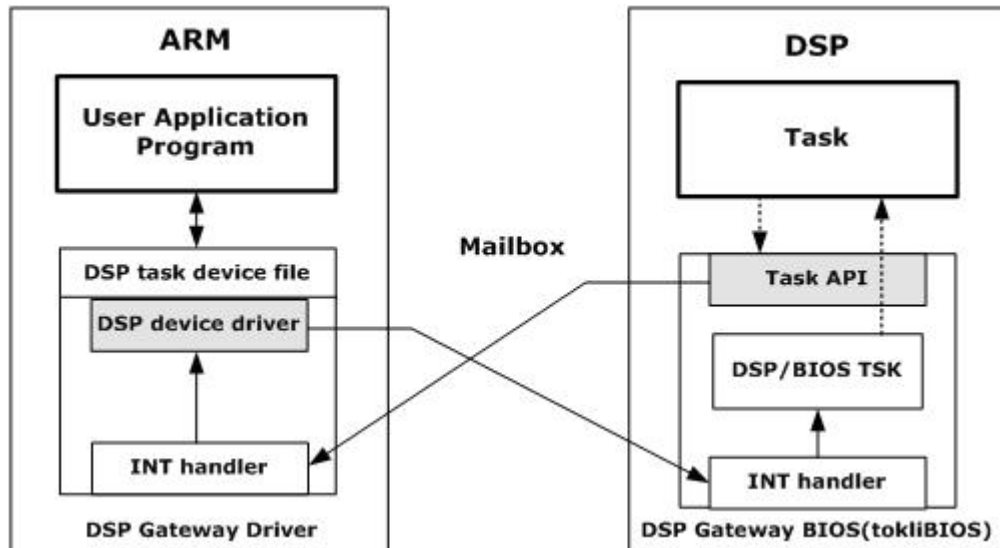
圖B.6、CCS整合發展環境

資料來源: Code Composer Studio Getting Started Guide Rev C(SPRU509C), Texas Instruments.

### B.3 雙核心處理器之內部溝通機制

OMAP 雖然是一個雙核心元件，但一般 ARM 的開發者有時候並不需要 DSP 核心，他們可能希望 OMAP 開發環境仍然和單核心處理器時相同。我們使用 DSP Gateway 的概念讓 ARM 和 DSP 建立連結，讓元件中 DSP 的存在變得透明，以保持 OMAP 運作時的彈性，讓一般 ARM 的開發者不需要 DSP 核心時，其開發環境仍然和單核心處理器(ARM)時相同。DSP Gateway[31]可被看作同時存在於兩個核心中的軟體層，其主要用來向 ARM 上執行的多媒體引擎導出一組 API(Application Program Interface, 應用程式介面)，以便其存取 DSP 資源。在 DSP 端，DSP Gateway 用於管理 DSP 資源和數據流，並向 DSP 演算法提供介面。在架設 DSP Gateway 之前，由於我們需要可以呼叫與管理 DSP Gateway 的介面，再加上我們在 DSP 與 ARM 上面都有許多工作(task)需要執行，所以，我們分別在 DSP 與 ARM 上面架設即時核心(DSP/BIOS)與嵌入式作業系統(ARM-Linux)來管理這兩個核心的資源。圖 B.7 表示 DSP Gateway 架構的關係圖。從圖中看出 DSP Gateway 包含兩部份，靠著這兩者合作完成通訊的功能：

1. Linux 裝置驅動程式：提供介面讓 ARM 的程式以系統呼叫(system call)的方式使用 DSP。
2. DSP 的函式庫：提供函式讓 DSP task 可以被 ARM 透過驅動程式使用。



圖B.7、DSP Gateway系統架構圖

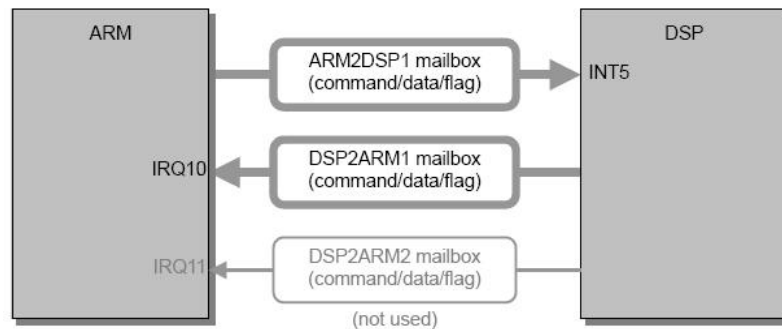
資料來源: <http://dspgateway.sourceforge.net/pub/index.php>

在 OMAP 處理器中的 ARM 與 DSP 資料交互傳遞的方法有三種，透過 Mailbox、MPU(Micro Processor Unit) Interface(MPUI)與 Share Memory(共享記憶體)等方式來達成，在 DSP Gateway 中是以 Mailbox 以及 Share Memory 來實現 ARM 與 DSP 資料交互傳遞。以下先對 ARM 與 DSP 資料交互傳遞的三種方法作簡介：

**(1) Mailbox 機制：**

OMAP5912 中有三組 Mailbox 暫存器，如圖 B.8 所示。一組是給 ARM 用，當 ARM 有資料寫入 Mailbox 時，會送出中斷(INT5)給 DSP；其他兩組則是當 DSP 有資料寫進時，才會送出中斷(IRQ10/11)給 ARM。每一組 Mailbox 是包含兩個 16 位元暫存器與一個 1 位元旗標暫存器。當中斷發生時，中斷的處理器會用兩個 16 位元的暫存器各自將命令與資料傳到被中斷的處理器上。





圖B.8、ARM/DSP Mailbox 說明圖

資料來源: <http://dspgateway.sourceforge.net/pub/index.php>

**(2) MPU Interface(MPUI)機制：**

MPUI 可以讓 ARM 和系統的DMA(Direct Memory Access，直接記憶體存取) 控制器與 DSP 和 DSP 的週邊互相通訊。MPUI 可以存取整個 DSP 的記憶體空間與 DSP 的週邊匯流排。因此 ARM 與系統的 DMA 可以有 DSP 輸出入空間的完整存取權限。至於透過 Share Memory 這樣的架構則是靠 OMAP 上的 Traffic Controller 達成，藉由 ARM 與 DSP 存取相同的記憶體空間。ARM 可以控制 DSP 的 MMU(Memory Management Unit，記憶體管理單元)讓 DSP 可以存取這些記憶媒介。

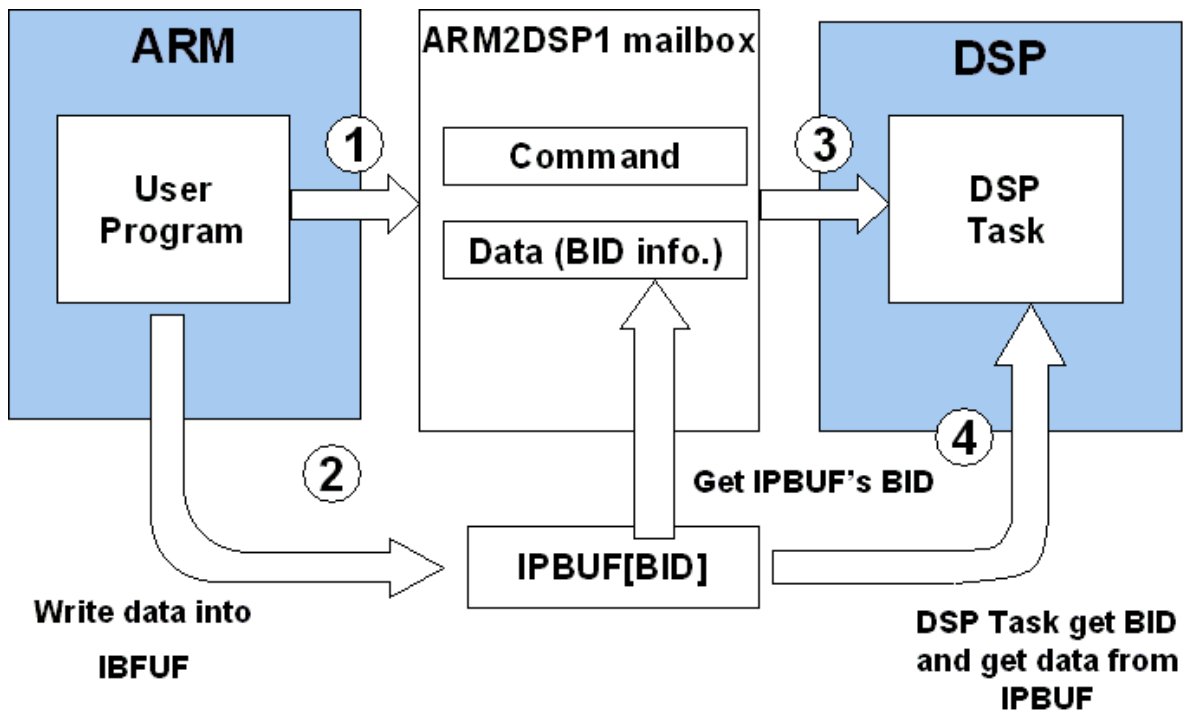
**(3) Share Memory 機制：**

由於 mailbox 只能傳送兩個 16 位元暫存器與一個 1 位元旗標暫存器，而無法傳送大量的資料，所以，我們必須借由 Share Memory 的方式來傳送大量的資料。而 Share Memory 這樣的架構則是靠 OMAP 上的 Traffic Controller 達成，藉由 ARM 與 DSP 存取相同的記憶體空間。ARM 可以控制 DSP 的 MMU 讓 DSP 可以存取這些記憶媒介，在 DSP Gateway 中，我們稱這一塊記憶體為 IPBUF。

圖 B.9 表示 Mailbox 和 IPBUF 在 ARM 和 DSP 之間的關係，ARM 端的程式要將資料交給 DSP task 處理時，也要將資料傳給 DSP 端。資料傳輸的型式，所採取的方式是 block send 的方式，ARM 在傳送資料時的 Mailbox command 中下達 BKSND 的命令，並且傳送 BID(buffer ID)給 DSP。ARM 在傳送資料時，會先傳送 Mailbox



command，這時會有 Mailbox 中斷(INT5)產生，接收這個命令，經過 interrupt handler 的處理 DSP task 收到接收資料的命令，並會到 Mailbox command 裡面指定的 IPBUF 內取得所要處理的資料，經過在 DSP 運算後，將得到結果存在記憶體內。同樣的 ARM 這時會送接收資料的命令過來，DSP 將得到的結果送到 IPBUF 中，ARM 可從 IPBUF 取得 DSP 處理後的資料。



圖B.9、ARM/DSP內部傳輸說明圖

資料來源: <http://dspgateway.sourceforge.net/pub/index.php>