

國立成功大學

電機工程學系

碩士論文

藍芽模組與處理器之連結控制設計與實作

Design and Implementation of a Bluetooth Module's
Connection and Control with Processors



研究生：鄭皓云

指導教授：林志隆

中華民國九十四年七月

國立成功大學
碩士論文

藍芽模組與處理器之連結控制
設計與實作

研究生：鄭皓云

本論文業經審查及口試合格特此證明
論文考試委員

林 志 隆
林 輝 堂

鄭 德 興
戴 政 祺
王 俊 智

指導教授：林 志 隆
系（所）主管：謝錫廷

中華民國九十四年七月二十五日

**Design and Implementation of a Bluetooth
Module's Connection and Control with Processors**

By

Haw-Yun Cheng

A thesis submitted to the Graduate Division in Partial
Fulfillment of the Requirement for the degree of

MASTER OF SCIENCE

Department of Electrical Engineering
National Cheng Kung University
Tainan, Taiwan, R.O.C.
July 25, 2005

Approved by:

Chih-Lung Lin 2. S. Kuo
Hsi-Feng Li Cheng-Hsi Tai
Wen-Ching Wu
Ce-Kuen Shieh

藍芽模組與處理器之連結控制設計與實作

鄭皓云* 林志隆**

國立成功大學電機工程研究所

論 文 摘 要

現今的藍芽產品及設備，主要以支援手機或大型的作業系統為主，而很少支援在周邊的小型電子產品，小型電子產品是以微處理器做為系統控制器，若能將藍芽與微處理器整合，則可將其模組化，做為小型的藍芽電子產品。本研究的目的是在設計及實作一個將藍芽無線技術與處理器結合的系統，不僅整合微處理器8051及TI(德州儀器)的MSP430處理器，更結合藍芽與Windows介面成為一個藍芽遠端控制系統，而研究分為三個部份，整合部份：撰寫藍芽模組與處理器的程式。配合8051、MSP430及PC(Windows)，撰寫介面程式及Protocol，來整合藍芽模組與處理器；實驗部份：建立Sever與Client端的資料互傳機制，來驗證系統的完整性及實用性，並藉由實驗得出不同封包、距離對傳輸速率與傳輸完整性的影響，並由實驗結果來評估各系統的優劣。應用部份：完成一個藍芽互動控制應用系統。本論文成功建構一個整合處理器與藍芽的系統，將藍芽嵌入設備中，使短距離無線通訊更具有發展性。

*作者

**指導教授

Design and Implementation of a Bluetooth Module's Connection and Control with Processors

Haw-Yun Cheng* Chih-Lung Lin**

Department of Electrical Engineering

National Cheng Kung University, Tainan, Taiwan R.O.C

Abstract

Nowadays the Bluetooth products and equipments primarily support to mobile phone or large-sized operating system, but support to the small electronic peripheral seldom. The small electronic products take microprocessors as central controller. If the processor and Bluetooth can be unified into one system, then the system can be served as a module. And it will manufacture as small Bluetooth electronic products.

This research's goal is to design and implement one system unified the Bluetooth and processor. Not only microprocessor 8051 and TI (Texas Instruments) MSP430, but also PC (Windows) is combined with Bluetooth to synthesize far-end control system. The research divides into three parts. Unified part: The programs are composed for the Bluetooth and processors. And the interface program and Protocol are composed with processor's formula to suit 8051, MSP430 and PC (Windows). By that profile the system can be unified Bluetooth and processor. Experimental part: The Sever and Client's data-transfer-profile is established to confirm the system's achievement and usability. Influence caused by speed and integrity of transmitting file is resulted by experimenting with different packet types and distance. So by discussing the result, the system's merit and defect will be appraised. Application part: Bluetooth interact-control application system was successfully accomplished. Then Bluetooth can be embedded into equipments. Due to this, the short-distance wireless communication will have more development.

* the author

** the advisor

致謝

在成大電機兩年碩士班的求學生活中，能順利完成學業，由衷地感謝陪伴我一起學習成長的老師、學長、同學朋友們。一路走來，學習到許多專業知識。且對自己的要求及做人做事的正確態度，也有一番的體認，是我這一生中很難忘的一段歷程。

首先要感謝我的指導教授林志隆教授在學術研究或生活上的指導，讓我對於學術研究的方法及方向有了新的認知，亦感謝組上各位老師的教導，最後感謝郭德盛老師、戴政祺老師、王俊智老師、林輝堂老師在論文口試上的指導與建議，使論文中不嚴謹的部分能更臻嚴謹與完善。

這些日子當然還要感謝CSR的工程師Victor、達方電子的資深工程師曾建台及Benson，還有連虹電子的主任工程師陳英俊等所提供的技術教學及協助。十分感謝張簡嘉壬學長在研究過程中所提供的協助，同窗好友崇文、賢尚、紀甫、隆基，在生活與課業上互相扶持及鼓勵。還有冠汶、勇志、富胤、宗廷、柏宇、嘉玄等學弟的各項幫助及帶來的許多歡笑使我的研究生活多采多姿。

最後要感謝我的父母對我長久以來的栽培及鼓勵以及家人的支持，使得我在研究的路途中無後顧之憂，並僅以本文獻給我的家人以及曾經給予我幫助的人。

目錄

	頁次
中文摘要.....	I
Abstract.....	II
致謝.....	III
目錄.....	IV
表目錄.....	VII
圖目錄.....	VIII
第 一 章 緒論.....	1
1.1 研究背景.....	1
1.2 文獻回顧.....	2
1.3 研究動機與目的.....	3
1.4 論文架構.....	3
第 二 章 藍芽無線技術.....	5
2.1 基本技術特性.....	5
2.2 藍芽(Bluetooth)技術概觀.....	5
2.3 藍芽封包的封裝過程.....	6
2.4 藍芽通訊協定堆疊.....	9
2.5 實體通道(Physical Channel).....	11
2.6 建立連結.....	12
2.7 設備位址.....	14
2.8 藍芽網路.....	15
2.9 資料傳輸型態.....	16
2.10 邏輯連結控制與適配協定(L2CAP).....	17
2.11 連結管理器協定(LMP).....	19
2.12 藍芽通訊協定(Profiles)簡述.....	21

2.12.1	GAP(Generic Access Profile)功能描述	21
2.12.2	SDAP(Service Discovery Application Profile)功能描述	21
2.12.3	SPP(Serial Port Profile)功能描述	21
2.12.4	GOEP(Generic Object Exchange Profile)功能描述	21
2.12.5	OPP(Object Push Profile)功能描述	21
2.12.6	FTP(File Transfer Profile)功能描述	22
第 三 章	系統整合架構原理介紹	23
3.1	處理器簡介	23
3.1.1	微處理器 8051	23
3.1.1.1	8051 硬體結構簡介	23
3.1.1.2	8051 內部結構簡介	24
3.1.1.3	8051 單晶片的接腳	25
3.1.2	TI的MSP430 超低功耗微控制器	27
3.1.2.1	MSP430 簡介	27
3.1.2.2	MSP430 工作模式	29
3.1.3	Windows XP SP1 作業系統	29
3.2	系統整合原理	30
3.2.1	藍芽與 8051 整合	30
3.2.1.1	系統概述	30
3.2.1.2	系統特色	31
3.2.2	藍芽與MSP430 整合	32
3.2.2.1	系統概述	32
3.2.2.2	系統特色	33
3.2.3	藍芽與Windows作業系統整合	34
3.2.3.1	Headset系統概述	34
3.2.3.2	Headset原理介紹	35
3.2.3.3	藍芽互動控制系統概述	37

3.2.3.4	藍芽互動控制系統原理介紹	37
3.2.3.5	藍芽互動控制系統特色	39
3.2.3.6	藍芽互動控制系統原理	39
第 四 章	系統設計與操作	41
4.1	使用儀器介紹	41
4.1.1	CASIRA	41
4.1.1.1	系統軟體要求	41
4.1.1.2	CSR BlueSuite的系統對應架構:	42
4.1.2	DARFON藍芽模組	42
4.1.3	MSI藍芽傳輸器(Bluetooth USB Dongle)	44
4.2	藍芽與 8051 整合為檔案傳輸系統	44
4.2.1	系統硬體:	44
4.2.2	系統軟體程式	45
4.3	藍芽與MSP430 整合為檔案傳輸系統	50
4.4	藍芽與Windows整合系統	50
4.4.1	系統一(Headset)	51
4.4.2	系統二(藍芽互動控制系統)	55
第 五 章	實驗結果與討論	57
5.1	8051 與藍芽整合的系統	57
5.2	MSP430 與藍芽整合的系統	59
5.3	PC Windows與藍芽整合的系統	60
5.4	實驗結果討論	61
第 六 章	結論與未來發展	63
6.1	結論	63
6.1	未來發展	63
參考文獻	65

表目錄

表 2-1	藍芽的七種ACL封包格式.....	7
表 4-1	BlueLab已提供的Library.....	48
表 4-2	BlueLab 初始定義Specification.....	48
表 5-1	藍芽封包位元傳輸率表.....	57
表 5-2	藍芽 8051 系統DH3 封包位元無障礙傳輸率表.....	58
表 5-3	藍芽 8051 系統DM5 封包位元無障礙傳輸率表	58
表 5-4	藍芽MSP430 系統DH3 封包位元無障礙傳輸率表	59
表 5-5	藍芽MSP430 系統DM5 封包位元無障礙傳輸率表	59
表 5-6	藍芽PC Windows系統DH3 封包位元無障礙傳輸率表.....	60
表 5-7	藍芽PC Windows系統DM5 封包位元無障礙傳輸率表	60
表 5-8	系統組成及使用技術.....	61
表 5-9	系統實驗結果及結果分析.....	62

圖目錄

圖 2-1	藍芽封包的基本格式.....	6
圖 2-2	藍芽堆疊的基本格式.....	9
圖 2-3	分時半雙工(TDD)示意圖.....	11
圖 2-4	多時槽傳送方式示意圖.....	12
圖 2-5	藍芽設備的連結狀態示意圖.....	12
圖 2-6	從待命到連線之訊息傳送流程圖.....	14
圖 2-7	LMP運作方式示意圖	20
圖 3-1	8051 結構關係圖.....	23
圖 3-2	8051 內部結構圖.....	24
圖 3-3	8051 接腳圖.....	25
圖 3-4	8051 做為I/O埠之提升電路	26
圖 3-5	8051 之致能電路.....	27
圖 3-6	TI-MSP430X 系列 電腦控制晶片架構圖.....	28
圖 3-7	控制晶片在不同電壓及不同工作模式之電流量圖.....	29
圖 3-8	8051 端連接方塊圖.....	30
圖 3-9	PC端連接方塊圖.....	30
圖 3-10	藍芽與MSP430 整合架構圖.....	32
圖 3-11	藍芽與MSP430 整合系統控制階層圖.....	33
圖 3-12	藍芽Dongle與PC (Windows作業系統)整合系統架構圖	35
圖 3-13	藍芽系統程式Inquiry流程圖.....	35
圖 3-14	藍芽Dongle系統連線架構圖.....	36
圖 3-15	藍芽系統Profile架構圖	36
圖 3-16	藍芽手機與PC整合之互動控制系統架構圖.....	37
圖 3-17	PC端之檔案辨識系統架構圖.....	38
圖 4-1	CSR CASIRA圖	41

圖 4-2	藍芽應用架構圖.....	42
圖 4-3	DARFON模組-master . slave.....	43
圖 4-4	DARFON模組-BlueCore腳位圖(部份).....	43
圖 4-5	DARFON模組-BlueCore對應ICL323 接線圖	43
圖 4-6	藍芽傳輸器(Bluetooth USB Dongle)外觀及內部圖.....	44
圖 4-7	8051 整合系統Master端連接圖	44
圖 4-8	8051 整合系統Slave端連接圖	45
圖 4-9	DARFON模組連接 8051 階層架構圖	45
圖 4-10	8051 主程式流程圖.....	46
圖 4-11	程式語言轉換程式Mess.2(C→Assemble)流程圖	47
圖 4-12	藍芽SPP Handshake腳位對應圖	49
圖 4-13	藍芽Dongle系統程式架構圖.....	51
圖 4-14	聲音傳送實驗成功圖.....	54
圖 4-15	藍芽控制系統控制示意圖.....	56
圖 4-16	藍芽控制系統實驗操作圖.....	56

第一章 緒論

1.1 研究背景

近來行動通信技術蓬勃發展，各種無線通訊規格紛紛設立，造成藍芽與802.11b/g、Home RF...[9]等無線技術互相比較勁。藍芽技術[12]為一種無線資料與語音通訊的開放性標準，藍芽技術工作在全球通用的2.4 GHz ISM（工業、科學、醫學）頻帶，藍芽的資料傳輸速率為721 kb/s。應用藍芽技術—PLUG&PLAY的概念，任何一個藍芽技術設備一旦搜尋到另一個藍芽技術設備，可以馬上建立連線，而無需設定。在無線電環境非常雜亂的環境下，其優勢更加明顯。且應用了全球統一的頻率設定，消除了國界的障礙，而在蜂巢式行動電話領域，這個障礙已經困擾用戶多年。但ISM頻帶是對所有無線電系統都開放的頻帶，因此使用其中的某個頻帶都會遇到不可預測的干擾源，例如某些家電、遙控器、無線電話、微波爐等，都可能是干擾源。而藍芽技術設計了快速確認和跳頻方式以確保連結穩定，使干擾的影響變得很小，且藍芽的跳頻速度快，資料封包更短，使藍芽系統更穩定。

藍芽優點[19]是可以利用無線介面來替代有線電纜連接；具有很強的移動性，可應用於各種通信場合；功率消耗低，對人體危害小；藍芽積體電路應用簡單，成本低廉，實現容易，易於推廣構成固定與行動設備通信環境中的個人網路，可在近距離內，與各種資訊設備進行無線資源分享。

現今因為生活品質提高，保健概念抬頭，所有電子產品的消費者，越來越重視無線電環境中的電磁波對身體健康的影響[29]，因藍芽的電磁波較弱，對人體危害較小，因此使用藍芽技術為現在的趨勢。

現在所提出的E-Home[6]（數位式家庭）的概念，即將生活空間完全無線化，使生活更加的便利。而藍芽適用於任何資料、圖像、聲音等短距離通信，目前的應用有：替換行動電話和遠端網路之間的通信時所用的有線電纜；提供新的功能耳機，並可在PC、行動電話、隨身聽中一起使用，還有筆記型電腦、PDA、行動電話之間的名片資料交換

等。若能再將生活周遭環境的電子設備全部藍芽無線化，如此在生活空間中將比較不用擔心因電磁波所引起的身體病變，更可以將E-Home的概念完全實現。

1.2 文獻回顧

對於藍芽的研究，一開始就有許多研究團隊進行如何將藍芽應用於系統上的研究，並建立藍芽堆疊連結層技術及Profiles[16]，來支援藍芽應用，以使藍芽達到最大的發展性。本研究主要是設計及實作一個將藍芽無線技術與處理器結合的系統。因不同的實作方式會影響使用性；相容性與安全性，因此在整合藍芽與處理器之前，先針對藍芽的程式設計架構與介面整合的論文，做研究探討。

在國內藍芽設計架構方面，國立中正大學溫志宏教授實驗室，已投入研究藍芽相關技術多時，將程式建構在藍芽系統規格V1.1之上，自行開發新的藍芽通訊協定，如藍芽應用軟體系統及使用者圖形介面的設定[31]及藍芽安全機制與認證功能訂立[26]等，更有設計車用藍芽免持器、藍芽電話、藍芽無線傳輸系統及藍芽監控系統[21,24,27,30,32]等技術。而成功大學亦有陳家進教授及戴政祺教授實驗室，針對藍芽短距離無線的應用，將藍芽用來傳送受監控的動物溫度資料 [25]，或是血壓量測數據的傳送端[33]，均是利用藍芽無線傳輸功能，來做應用。

而在國外藍芽設計研究方面，自2001年，Arumugam等人，開始著手研究藍芽應用於設備上，所遭遇的問題，並提出較適用的藍芽技術[1]。而現在藍芽產品主要都是利用藍芽資料傳輸功能，來做藍芽兩端的資料傳輸應用[7]。之後，再研究將藍芽連線的聲音與資料的傳送機制及藍芽系統堆疊，應用於影音傳送上，並以體積小；功率低的藍芽晶片做成藍芽產品，實現在居家的生活環境中，達到藍芽無線生活化的目的[2]。

由以上的研究發表，顯示出現代對於藍芽短距離無線技術非常重視，且亦致力於藍芽應用系統的發展[11,17]，但美中不足的是對藍芽產品及設備，尚未對小型的周邊電子產品，建立一個可以整合藍芽與小型電子產品的架構，且對於介面的整合評估，也不盡完全。因此，本研究在於設計及實作一個藍芽與處理器結合的系統，並完整的評估的介面優劣，來建立一個通用的整合架構，並將介面評估完整化，使未來有更多藍芽的設計產品，更讓藍芽無線技術更加的完備並兼具更多功能的應用。

1.3 研究動機與目的

在無線網路的應用當中，無線個人區域網路（WPANs）和無線區域網路（WLANs）的應用將快速成長。現今無線個人區域網路的技術以藍芽最為熱門。而在無線區域網路的領域包括數種成熟的技術，如IEEE 802.11a/b/g，家用無線網路（HomeRF）等。藍芽和無線區域網路兩者在應用上各有其特色與價值。

由於藍芽技術獨立於不同的作業系統和通信協定之外，可以移植到許多應用領域，且應用範圍也很廣，但針對現今藍芽產品及設備，主要是支援手機或PC、NB此類大型的作業系統，而尚未支援在小型的周邊電子設備，而小型電子設備以微處理器作為控制系統，若能將藍芽與微處理器整合為一系統，則可將其模組化，製成小型藍芽電子設備。

研究目的為設計一個將藍芽無線技術與處理器結合的系統架構，並對於微處理器8051及TI（德州儀器）的處理器MSP430及PC（Windows）等三個處理器，實現三個整合的藍芽系統，並加以評估系統結合時，在處理器與介面搭配的優劣性。最後實作一個藍芽控制系統，做為藍芽短距離無線互動控制的應用。

最後，只要依此架構，小型電子設備的微處理器系統就可整合入藍芽功能，成為新型的藍芽產品，使藍芽無線技術的應用上更有創意、更具便利性，進一步實現E-Home的概念，使生活無線化。

1.4 論文架構

本章首先介紹藍芽技術於現在通信應用的背景，接著介紹藍芽所可以整合的系統應用與目前相關的研究方案以及研究的目的。第二章為藍芽技術的規格描述，並對於藍芽基頻（Baseband）、鏈結控制與管理（Link Control & Manage）以及邏輯鏈結控制適應（Logical Link Control and Adaptation）及藍芽通訊協定（Profiles）等部分做較詳細的介紹。第三章為系統整合架構的原理介紹，主要是介紹為藍芽與8051、德州儀器出產的MSP430及Windows系統的整合原理。第四章為系統設計與操作，主要介紹實驗器材及實驗操作流程，並說明系統的程式流程，以便了解系統的設計架構與程式撰寫方式。第五章則為實驗結果與討論。是針對系統實驗的結果提出分析及對於各系統做詳盡的剖析與

發展的方向。第六章為結論及未來展望。

第二章 藍芽無線技術

2.1 基本技術特性

藍芽的最初概念起源於1994年，Ericsson的一項低功率、低成本的射頻介面研究，其重點在於短距離的無線傳輸技術。藍芽採用高斯二位元頻率鍵送GFSK (Gaussian-shaped, Binary FSK)的調變技術，其調變指數(Modulation Index)為0.3；傳輸功率為1 mW 至100 mW 。藍芽採用目前開放但沒有特定授權的2.4 GHz ISM (Industrial Scientific Medical)公用頻帶，藍芽是採取跳頻(Frequency Hopping, FH)分碼的多工擷取方式(Code Division Multiple Access)。在2.4 GHz 的波段中，間隔為1 MHz，定義了79個跳頻載波，稱為跳頻頻道(Hopping Channel)，其跳頻的頻率為1600次/秒(一個time slot的時間為625 μ s)。此外藍芽採用了時域分割(Time-Division Duplex, TDD)及跳頻技術。用在連續的兩個time slot，前者用來作為Master送資料給Slave之用；後者用來作為Slave送資料給Master之用，因此除了能在抵抗外界干擾之同時，更能有效避免傳送與接收訊號間彼此的交互衝擊干擾。假如Master成功地將資料傳送給Slave，則Slave將回傳(Acknowledgment)訊息給Master，Master則繼續傳送下一筆資料；若傳送過程封包與外界干擾發生碰撞，則Slave將回傳NACK(Negative Acknowledgment)，Master則在下一個Time Slot重傳資料[20,23]。

2.2 藍芽(Bluetooth)技術概觀

藍芽是一項具有低功率、低耗電的短距離無線通訊技術。最初設計理念是為了讓各種設備如行動電話、筆記型電腦、PDA等能藉由藍芽技術互相連結，取代原本設備互連時所需的纜線。藍芽技術的開發是由SIG領導的。該小組由九名成員發起：Ericsson、Nokia、IBM、Intel、Toshiba、3Com、Lucent、Microsoft和Motorola。而目前SIG的成員已成長到1800多個，遍及各種不同的領域、如半導體、通訊、計算、週邊設備、網際網路、汽車等。

藍芽通訊協定標準[15]同時支援了電路交換(Circuit-switching)與封包交換

(Packet-switching)兩種技術。傳送語音時採用電路交換的技術，而傳送資料時則使用封包交換技術。藍芽使用跳頻/分時半雙工(Frequency Hopping/Time-Division Duplex簡稱FH/TDD)的傳輸機制，視採用的傳輸功率大小，其傳輸範圍可從10公尺到100公尺。

藍芽目前主要的應用模式有三種：

1. 關於纜線取代(Cable Replacement)的應用：

由於藍芽為短距離的無線通訊方式，並且具有低功率、低成本的特色，因此相當適合用於取代傳統連接個人週邊設備的纜線，使設備的連接更方便、迅速。

2. 數據和語音存取點(Data/Voice Access Point)的應用：

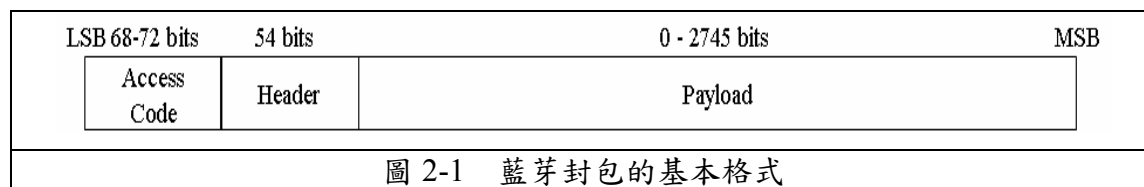
藍芽的射頻晶片可安裝於家電設備、辦公室、汽車、機場、旅社等各個地電，作為無線通訊的節點，使用者可以即時開啟具有藍芽功能的行動電話或筆記型電腦、以無線方式連接上網際網路並瀏覽網頁、收發電子郵件，如同在有線的環境中上網一般，只不過是由藍芽短距離無線通訊方式完成。

3. 即建即連網路(Ad Hoc Networking)的應用：

只要在藍芽的通訊範圍之內，任何兩個安裝藍芽晶片的設備都可以建立即時連結。藍芽所形成的網路是以 Piconet 為單位。並且可藉由某些設備將 Piconet 連結起來，形成更大的網路-稱為分散網(Scatternet)。

2.3 藍芽封包的封裝過程

藍芽是以封包的格式傳送，封包的基本格式如圖2-1所示，由72位元的Access Code、54位元的Header及0~2745位元的Payload所組成。為了方便說明封包的封裝過程，先從後端的Payload，接著是Packet Header，最後是Access Code，依次說明。



1. Payload：

Payload為藍芽封包放置資料的地方，可分為兩類封包格式，一類為ACL (Asynchronous Connection-Less)，另一類為SCO(Synchronous Connection-Oriented)，其中ACL封包是用來傳送一般資料，其非對稱傳輸最快可達723.2 kb/s，SCO封包型態的封包主要是來傳送語音資料。表2-1中的FEC (Forward Error Correction)是用於Payload的保護機制，有加2/3 FEC 保護的ACL封包稱為DM封包，而沒有2/3 FEC保護的ACL封包稱為DH 封包，其中DM 與DH 封包又因可容納資料量的不同，分別有三種不同的封包(DM1、3、5 與DH1、3、5)。另外，CRC (Cyclic Redundancy Check)主要是用來檢查Payload 是否在傳輸過程有發生錯誤。2/3 FEC與CRC機制在之後分別有更詳細的介紹。而在ACL封包中還有定義封AUX1包，但AUX1封包幾乎與DH1封包相同，差別在只於AUX1並沒有使用CRC機制，故一般並沒有考慮AUX1封包。

表 2-1 藍芽的七種 ACL 封包格式(Spec.)

Type	Payload Header (bytes)	User Payload (bytes)	FEC	CRC	Symmetric Max. Rate (kb/s)	Asymmetric Max. Rate (kb/s)	
						Forward	Reverse
DM1	1	0-17	2/3	yes	108.8	108.8	108.8
DH1	1	0-27	no	yes	172.8	172.8	172.8
DM3	2	0-121	2/3	yes	258.1	387.2	54.4
DH3	2	0-183	no	yes	390.4	585.6	86.4
DM5	2	0-224	2/3	yes	286.7	477.8	36.3
DH5	2	0-339	no	yes	433.9	723.2	57.6
AUX1	1	0-29	no	no	185.6	185.6	185.6

在每一個封包的Payload的前端會加入8或16位元的Payload Header(P_H)，包含了邏輯通道和Payload大小的資訊，所以可根據提供的資料來選擇邏輯通道和定義資料格式，其中P_H中的L_CH(Logical Channel)為邏輯通道，Flow則是用來控制邏輯通道的流量，Length是用來表示封包資料的長度，至於Un-Defined則是保留給未來新增的功能，目前必須全部設為0。而在每一個封包的Payload 後端也會加上CRC檢查碼，CRC檢查碼的產

生方式主要是在傳輸端，將資料經CRC Circuitry處理後，產生16位元的CRC檢查碼，再把這CRC碼放於原始資料後面，成為新的資料串列，而在接收端可透過CRC Circuitry來做錯誤的偵測。經過加上P_H以及CRC檢查碼之後，對資料進行保護動作，如果是DM1、3、5的封包，資料每10個位元會用2/3 FEC作保護，如果是DH1、3、5的封包，則沒有保護。

2. Header：

Payload經過處理完成之後，加入Header，Header的格式如圖所示。其中Active Member Address (AM_ADDR)是Master給予Slave的位址，目的是用來區分在Piconet中有發生動作的單元。

藍芽定有16種不同的封包，在TYPE中會有所定義，而最主要可以分為ACL 資料封包以及SCO的語音封包。Flow 用來控制ACL 連結的流量，ARQN 則是判別是否有接收到訊息，有接收到會回傳Acknowledge(ARQN = 1)，沒有則回傳Negative Acknowledge (ARQN = 0)。SEQN可排列資料封包串列的順序，最後的HEC (Header Error Check)可用來確認Header在傳送過程後是否發生錯誤，在傳輸端將資料經HEC Circuitry 處理後產生8位元的HEC碼，再將HEC碼放於原始資料後面，成為新的資料串，而在接收端可透過HEC Circuitry來做錯誤的偵測。

當形成Header後(共18 bits)，必須加上1/3 FEC的保護，才能完成最終的Header(共54 bits)。而1/3 FEC主要想法是用三次的重複碼來完成。經過1/3 FEC 保護之後的Packet Header還需要經過資料白化(Data Whitening)來打散資料和降低偏壓(Bias)。

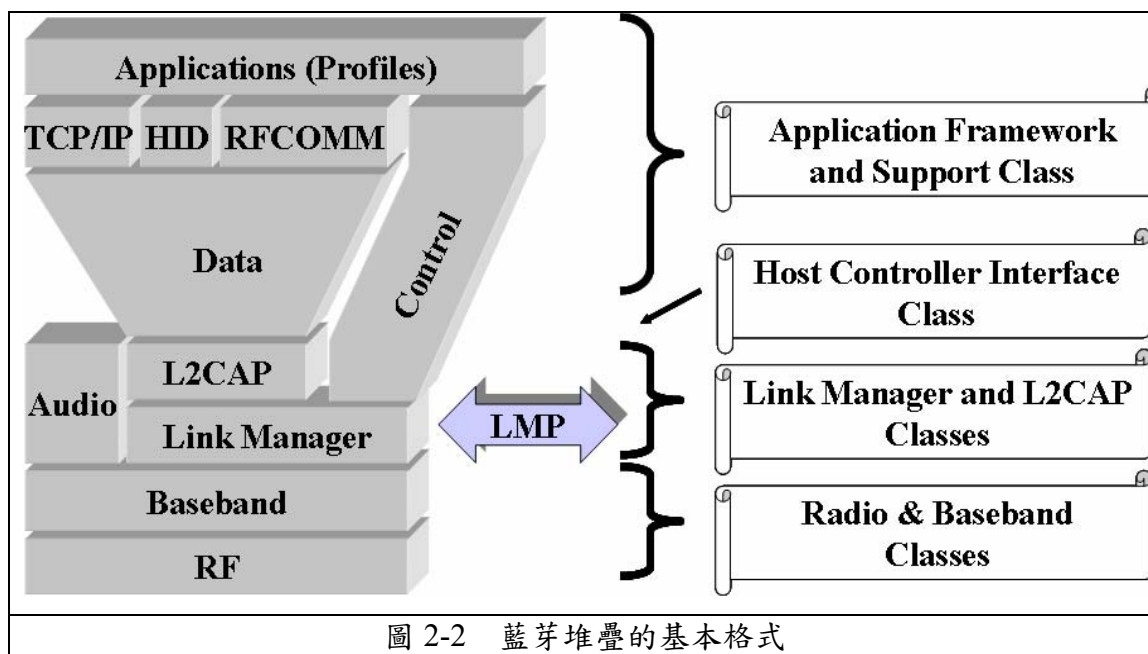
3. Access Code：

完成資料白化動作之後，最後再加上Access Code 就可以完成Bluetooth封包的封裝，所謂的Access Code是可讓Master確認在同一個Piconet內，所有傳送封包的身份。其中PREAMBLE包含4位元，主要功能是為了幫助DC直流補償(DC Offset Compensation)，上面的數值不是1010就是0101，取決於SYNCWORD內的LSB位元：若LSB為1，則PREAMBLE 的數值為1010；若LSB為0，則PREAMBLE的數值為0101。SYNCWORD包

含64個位元，是由設備位置BD_ADDR內24個位元的LAP 計算而來。而TRAILER包含4個位元，主要功能是為了延長PREAMBLE的DC 直流補償，數值也是1010或0101，取決於SYNC WORD內的MSB位元：若MSB為1，則PREAMBLE的數值為0101；若MSB為0，則PREAMBLE的數值為1010。

將上述封裝過程整理，一段經處理後的ECG資料位元，會先依照Payload 大小，將資料位元切割成 n 段後，每一段會再加入P_H及CRC檢查碼。之後根據封包的類型，考慮是否加上3/2 FEC保護，然後加上Packet Header，經Data Whitening之後，再加上Access Code，藍芽封裝就完成了。

2.4 藍芽通訊協定堆疊



藍芽本身為一種無線網路的標準，由圖2-2可以了解藍芽的通訊協定架構。整體而言，若以OSI/ISO的層級(Layer)來看藍芽的核心架構部分(Bluetooth Core Protocols)，主要是位於層級架構中的實體層級資料鏈結層方面，也就是說藍芽標準為網路底層的一種標準。在其高層協定方面，透過Profile之制定也可以採用其他的網路標準，如：TCP/IP、

HID...等。藍芽的核心架構，包含了藍芽的基頻(Baseband)標準、連接管理協定(Link Management Protocol, LMP)、邏輯連接控制與應用協定(Logical Link Control and Adaptation Protocol, L2CAP)、服務尋找協定(Service Discovery Protocol SDP)，是藍芽主要的核心機制。其功能分述如下：

1. 無線電(Radio)：

定義藍芽設備所操作的無線電頻段是在2.4 GHz的公用頻段(ISM band)，主要定義系統中無線電的相容性、調變及電磁干擾之規範。藍芽採用了GFSK的調變技術。

2. 基頻(Baseband)：

定義基本的實體層協定。包括跳頻的技術，硬體線路與模組的設計，連線的過程，分時半雙工的細節，資料封包的設計，以及相關的容錯、安全等機制的規範。

3. 連結管理協定(LMP)：

作為設備間連線時的連線管理者(Link Manager, LM)，彼此之間的溝通，處理關於基頻連線的設定及控制，及省電模式之操作。

4. 邏輯連結控制與適用協定(L2CAP)：

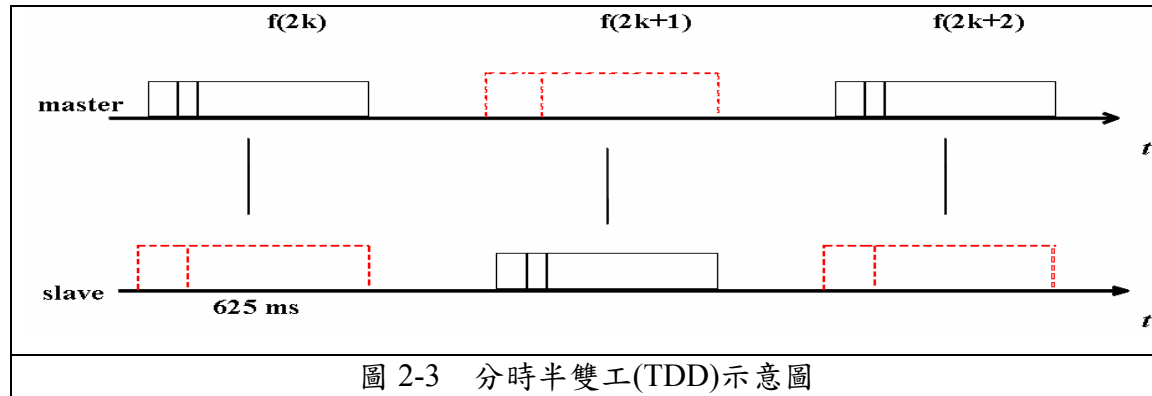
主要提供上層應用之傳輸資料鏈結相關控制功能，例如上層協定的多工傳送(Multiplexing)、封包的切割與重組(Segmentation and Reassembly)以及傳輸品質的協商(Quality of Service, QoS)等等。

5. 服務尋找協定(SDP)：

服務尋找協定是用來協助上層的應用程式，搜尋通訊範圍中，提供特定服務的藍芽設備。或用於了解範圍中其他設備所提供的服務內容。其設計主要是針對無線環境的特性，無線電波距離有限的特性，且裝置隨時會移動的情形下，來達成尋找服務的目的。SDP採用Client-Server通訊架構，服務搜尋時SDP Client將發出需求，由SDP Server回應結果。

2.5 實體通道(Physical Channel)

藍芽標準定義主動發出連結要求的設備稱為Master，另一個被連結的設備則稱為Slave，當兩個藍芽設備互相連結後，等於是在Baseband階層建立起一條實體通道(Physical Channel)，實體通道內Master與Slave之間的訊號傳遞是以TDD(Time Division Duplex)方式，如圖2-3。



Master在偶數時槽(Time Slot)時送出資料，Slave則進行監聽，於下一個時槽則由Slave發送資料改由Master監聽，時槽上的傳遞的封包並不佔滿整個時槽，每個時槽的傳輸時間為625 μ s，即頻率為1600 slots/sec。

由於藍芽採用了跳頻的機制，因此在進入下一個時槽時將跳躍到另一個頻道上。藍芽將2.45 GHz 通訊頻段切割成79個1 MHz 的頻道，資料將在這79個頻道上交替的傳輸。頻道的跳躍序列(Hopping Sequence)是由Master的設備的唯一位址BD_ADDR計算出來的。

Master只在偶數的時槽傳送資料，Slave設備只在奇數(Odd)的時槽傳送資料，Master傳送的封包，不僅只佔有1個時槽，還能擴展到3個或5個時槽，如圖2-4即為1、3、5時槽的傳送型態。

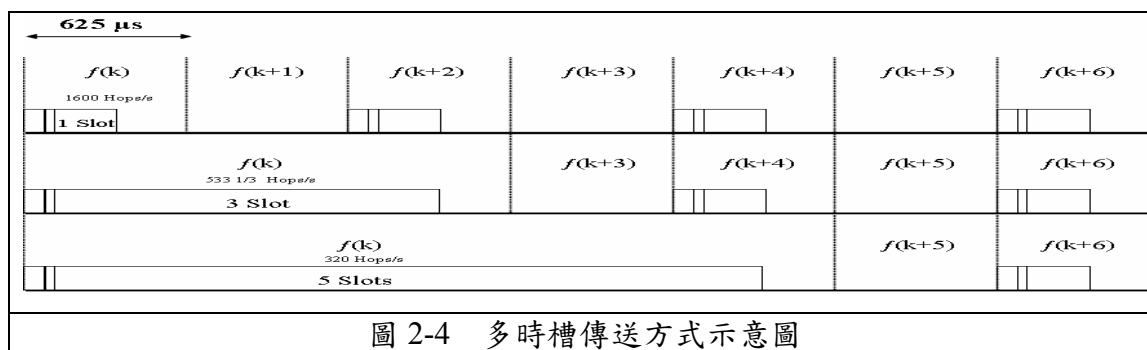


圖2-4，當Master傳送的封包為多個時槽時，因為省略封包在Master與Slave來回的交換，以及減少傳輸單一封包的標頭，所以傳輸速率將提高，例如若是Master與Slave之間的傳輸封包皆佔用一個時槽，則傳輸速率為對稱的172 Kbits，若是Master到Slave的傳輸封包佔用5個時槽、Slave到Master的傳輸封包佔用一個時槽，則Master到Slave的傳輸速率為721 Kbit/s、Slave到Master的傳輸速率為57 Kbit/s。

2.6 建立連結

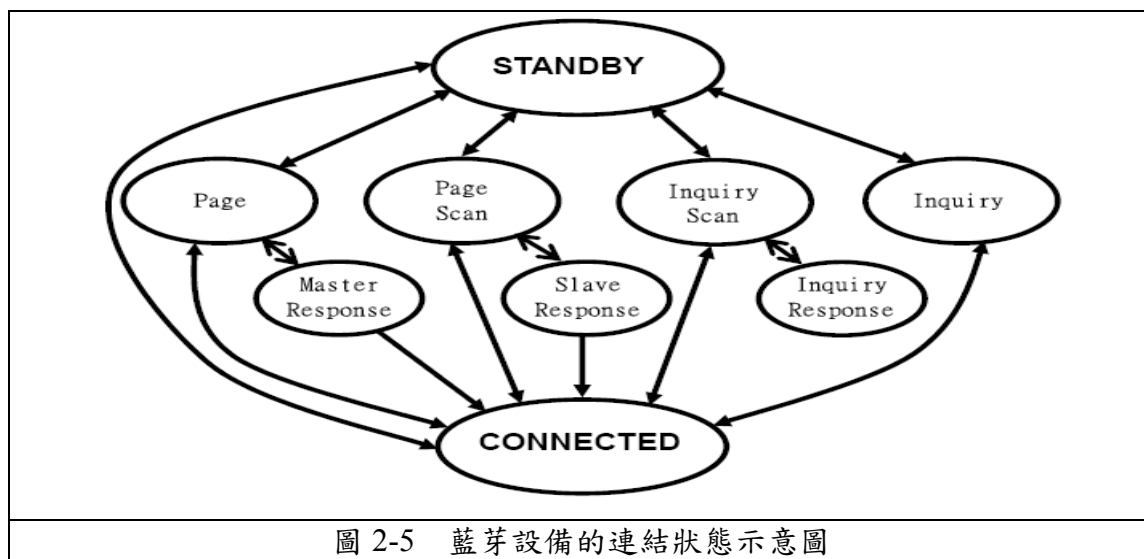


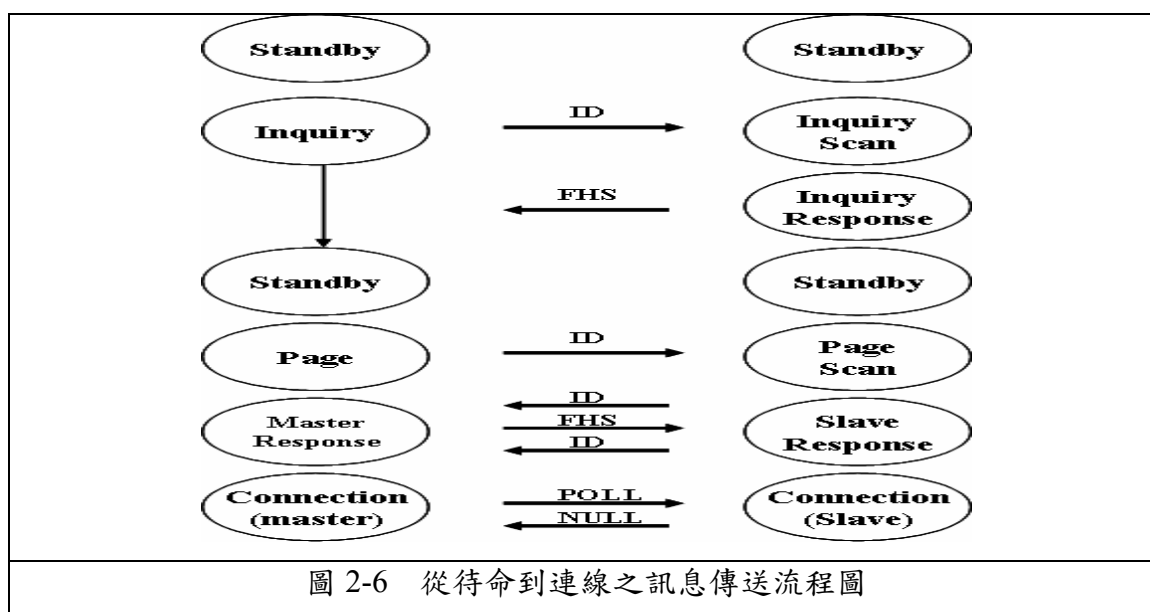
圖2-5為藍芽設備的連結狀態示意圖。藍芽設備當起始時會在預備(Standby)的狀態。當設備A欲找尋週遭的其他設備時，便會進入詢問(Inquiry)的狀態，同時發出諮詢。

詢問的封包包括了A的藍芽設備位址以及A的時脈。若設備B此時正進入Inquiry Scan的狀態，在等待周圍設備的詢問，便會收到A的訊息，如果B願意接受A的詢問，便會進入Inquiry Response的狀態，並回覆B的藍芽設備位址以及B的時脈。如此A和B皆可獲得彼此資訊，便能建立各自的週遭已存在設備列表。當A不再想詢問或B不再想被查詢時，即可回到預備狀態。

當設備 A 有了周圍設備位址，知道 B 在附近，想要與 B 建立連線時，便會進入 Page 的狀態，在一段時間內持續送出 Page Request。若 B 想要被連線，那麼每隔一段時間就會進入 Page Scan 狀態，由於 A 是以預測 B 跳頻的可能結果來密集送出 Page Request，因此 B 會以下圖的方式收到 A 的 Page Request。若 B 願意連線則進入 Slave Response 的狀態，同時回覆給 A 一個 Page Response；若不願意，則不必回應。若 A 隔一段時間後沒收到 B 的回應，則會逾時(Time-out)，此時可重複試驗數次，若仍未成功，便回到預備狀態。

當A收到B的Page Response之後，會進到Master Response的狀態，此時 A會送出一個FHS packet，其中包含A的位址與時脈資訊，以便B用來與A作同步跳頻。若B回應A一個page response，則便會在之後切換到以A的跳頻方式與時脈的頻道，此時A便會送出第一個真正帶有資料的封包給B，完成連線與藍芽微網的建構動作。而A即成為此微網的Master，B成為此微網的Slave。

綜合上述Inquiry、Page過程，整合出藍芽主要設備元件（Master）和附屬設備元件（Slave）間，從待命到連線之訊息傳送流程圖，如圖2-6。



2.7 設備位址

在藍芽系統中共有四種型態的設備位址(Device Address)，以下各別敘述：

1. BD_ADDR：

在所有的藍芽設備內皆有一個唯一的 48 位元的藍芽設備位址(BD_ADDR)，如同個人電腦網路卡內部都有一個唯一實體位址，這個位址為藍芽技術的運算核心，如：跳頻序列(Hopping Sequence)、Channel Access Code、加密私鑰(Encryption)都是由此位址計算求得。BD_ADDR 也可視為藍芽設備的硬體序號，這個位址是在每個藍芽設備出廠時，由製造商燒錄在設備內，全世界所有藍芽設備的 BD_ADDR 由 Bluetooth SIG 協會負責分配與管理，製造商需要的 BD_ADDR 必須向 Bluetooth SIG 協會申請。

2. AM_ADDR：

當Slave設備在Piconet中若是為Active的狀態時，Master將分配每個Active的Slave一個AM_ADDR(Active Member Address)位址，Master就是經由這個位址來辨識在Piconet中各個不同的Slave，AM_ADDR由3個位元所組成，3個位元組成8個不同的數值，所以在一個Piconet內包括Master與Slave最多有8個設備，Master的AM_ADDR位址固定為000。

3. PM_ADDR：

從Active狀態進入Park狀態的藍芽設備將得到一個PM_ADDR。PM_ADDR由8個位元組成，所以在Piconet內最多可容納256個Park狀態的藍芽設備。

4. AR_ADDR：

在Piconet內的所有Park狀態的藍芽設備，都會分配到一個AR_ADDR。當Slave要從Park狀態回復到Active狀態時，將以此位址向Master要求，可傳送的Slave-to-Master時槽。

2.8 藍芽網路

藍芽系統由無線部分、連結控制部分、連結管理支援部分和主終端機介紹組成。網路架構分為藍芽微網(Piconet)及藍芽分散網(Scatternet) 兩大範圍：

1. 藍芽微網(Piconet)原理：

許多個藍芽設備互相連接的網路成為一個Piconet，是藍芽網路的基本單位。在一個Piconet之中，Master最多可以同時與7個位於Active狀態的Slave互相傳遞資料，是一個由一個Master和不同數目的Slave所構成的Piconet。在藍芽系統的定義上，所有設備的地位都是平等的，原則上任何藍芽裝置皆可以是Master或是Slave，規定是首先提出連線要求的設備就稱為Master，而被連結的設備就稱為Slave，角色的分配是在Piconet形成時就確定的。依照應用程式的需求，有時Master與Slave的功能還能互相切換。

由於Master與Slave之間的實體通道(Physical Channel)是以TDD互相傳送資料，彼此間必須建立的時序同步。每個藍芽設備都有一個內部系統時序CLKN(Native Clock)，用以決定封包傳送的時間。這個內部時序不斷的運行計算，無法被調整或關閉。

在Piconet中建立時序同步的方法是以Master的內部時序CLK(Master Clock)為基準，在此Piconet的每個Slave都加上時序偏移(Clock Offset)，以使每個Slave的時序與Master的時序互相同步。由此可知，頻率跳躍是由Master的BD_ADDR計算出跳頻序列(Hopping Sequence)，再將每個Slave的內部系統時序CLKN加上Offset後，使所有的Slave的時序與Master相同。為了隨時都能維持Master與Slave精準的同步，每次當Master傳送

封包到Slave時，都會加上時序偏移來校正。

當Slave設備只與Piconet的時序互相同步但不傳遞資料時，Slave算是位於Park狀態，一個Piconet最多可容納256個Park狀態的Slave。

2. 藍芽分散網(Scatternet)：

在Piconet中的所有設備皆共同分享Piconet的1Mbps傳輸速率，但當有更多的Slave加入Piconet時，每個Slave所分配到的傳輸速率將隨之下降，藍芽標準所採用的解決方法，是使各個Piconet之間的設備能夠互相通信。也就是說一個設備並不只限於參與一個Piconet。而這些互相連接的Piconet便形成了另一種藍芽網路型態稱之為分散網(Scatternet)。

由於藍芽採用了跳頻的技術，因此Piconet間由於跳頻序列(Hopping Sequence)不同，彼此之間的干擾將會降低。這種特色也使得藍芽技術有著多重頻道(Multiple Channel)的觀念，和目前其他的無線網路技術如802.11b WLAN的單一頻道(Single Channel)有著很大的不同[22]。在802.11b中網路的拓模型態只取決於設備之間的距離。但是藍芽的網路拓模是多變的，兩個設備之間若沒有建立連結(使其跳頻順序相同)就算彼此皆在無線電波距離內，一樣無法傳輸資料。參與一個以上Piconet的設備，必須以TDM(Time Division Multiplexing)方式於各個Piconet切換。因為在同一時間內一個設備只能在一個Piconet中是Active的狀態。故當一個設備參與太多Piconet時，將會影響其他Piconet執行資料轉送。

2.9 資料傳輸型態

藍芽技術具備同時傳送語音(Voice)與數據(Data)兩種資料型態，最主要的原因在於藍芽技術支援電路交換(Circuit-switching)與封包交換(Packet-switching)兩種資料傳輸方式，在藍芽標準中電路交換的傳輸稱為SCO連線、封包交換的傳輸稱為ACL連線[4,8]。SCO(Synchronous Connection-Oriented)連線屬於電路交換的同步傳輸型態，電路交換是當Master與Slave之間的連線一旦建立後，不管有無資料傳送，系統都會預留固定間隔的時槽給Master與Slave，其他Slave就不能利用此連線上的時槽來傳送資料，SCO連線適合語音的傳輸，每一個SCO支援64kbps的語音通話，一旦SCO連線建立，Master與Slave

設備可直接發送SCO封包，Master無須事先詢問(POLL)Slave，SCO連線屬與單點對單點Point-to-Point的對稱連接，即SCO連線建立在一個Master與Slave間。

當SCO連線在傳輸語音時，由於語音封包不適合因干擾而重新傳送。保護語音封包的方法是採用非常嚴格的語音編碼(Voice-coding)，即使接收一方收到封包錯誤率非常高時，解碼後的語音品質仍可以接受。

ACL(Asynchronous Connection-Less)連線屬於封包交換的非同步傳輸型態，封包交換是將高層的資料切割成一段段的封包(Packet)，當實體通道(Physical Channel)上的時槽沒有任何SCO連線時，ACL連線可佔用任意時槽來傳輸資料，一旦系統需要傳輸SCO連線時，ACL連線則自動空出時槽來提供SCO連線使用，ACL連線只有傳輸在SCO連線不使用的時槽上，Master的ACL連線利用SCO連線之間的時槽，來傳輸資料到周圍的Slave。

ACL連線這種封包交換的傳輸型態，是在傳送數據(Data)時才運用時槽，與現在網際網路Internet傳輸數據的方式相同，適合傳輸突發性(Burst)的數據資料，Master可同時與多個Slave建立ACL連線，屬於單點對多點(Point-to- Multipoint)的非對稱連接。

Master負責分配Piconet中每個Slave到Master間的傳輸速率，在Master送出ACL連線封包之前，必須先詢問(POLL)各個Slave，選定(AM_ADDR)某個Slave後才能傳送數據資料，ACL連線也支援Master到所有Slave的廣播訊息。由於藍芽採用TDD的存取機制，Slave只有在被Master詢問(POLL)時才可以再下一個時槽中傳送資料給Master。

ACL連線支援對稱和非對稱兩種傳輸速率，在非對稱速率時，雖然Piconet的最大頻寬為1Mbps，但是封包還需要負責傳送控制訊號，所以ACL的數據傳輸率在非對稱連接時，下載的傳輸速率為721Kbits、上傳的傳輸速率為57.6Kbps，對稱連接時，Master到Slave之間的傳輸速率各為432.6Kbps [14]。

2.10 邏輯連結控制與適配協定(L2CAP)

L2CAP(Logical Link Control and Adaptation Protocol)是位Baseband上層的協定，必須能夠接受來自Baseband產生的事件，並能把這些事件發到上層的SDP、RFCOMM、TCS協定，同時它也能將上層的協定與應用切割成較小的封包後，透過

L2CAP封包與LMP封包傳輸在不同的ACL連線上。LMP封包傳遞由Baseband 所建立起來的ACL連線上，ACL連線一旦建立後，連線即一直存在於Master與一個Slave間， ACL連線支援單點對多點(Point-to-Multipoint)的同步(Synchronous)與非同步(Asynchronous)的資料傳輸。反觀L2CAP封包的傳輸就比較複雜，傳輸L2CAP封包前，兩個設備必須以ACL連線為基礎，建立及開啟(Open)邏輯通道(Logical Channel)，傳輸結束後又必須關閉邏輯頻道，L2CAP支援許多高層SDP、RFCOMM、TCS等協定，與語音有關的應用直接傳輸於Baseband 階層的SCO連線上，但是若以封包交換型態的語音應用，如IP Telephony，仍然是傳輸在Baseband階層的ACL連線上[5]。

L2CAP協定的封包傳輸前必須先建立起邏輯通道(Logical Channel)，每個ACL連線上可以建立65535個邏輯通道，每個邏輯通道只能傳輸一種上層協定，但多個邏輯通道可以同時傳輸同一種上層協定，每個邏輯通道上接收到的L2CAP封包都會分配到對應的上層協定，協定內每個邏輯通道的每個端點都可以用一個通道識別碼CID (Channel Identifier)來表示。

CID數值共有16個位元，通常是邏輯通道建立時分配給各個端點(Dynamically Allocated)，但是0x0001到0x0003F之間的CID數值保留給特定的功能，例如0x0001保留給訊號通道(Signaling Channel)所使用、0x0002保留給非連接導向通道(Connectionless Channel)的接收端所使用，餘下的0x0040-0xFFFF可以自由分配。

邏輯通道的兩個端點各分配到一個CID標誌，對於Connection-Oriented通道，每個通道都是雙向的傳輸(Bi-directional)，並且通道的兩個傳輸方向都有服務品質QoS的要求。

L2CAP是介於高層與底層間的適應層(Adapted-layer)，主要是負責兩個Bluetooth設備間數據資料時的分段及重組(Segment and Reassembly)、協定多工(Protocol Multiplexing)和協商通道參數等功能。協商通道參數(Negotiation)，當連接導向(Connection-Oriented)的邏輯通道建立起來後，在進行數據資料的傳輸前，必須設定通道參數的 (Configuration)階段，是兩個設備互相的協商(Negotiation)，以取得雙方都同意接受的通道參數。

L2CAP 協定建立邏輯通道，協議出適當的 MTU 與服務品質 QoS 後，上層協定的指令或資料即能以 L2CAP 協定的區段加以傳遞。由於 Baseband 的封包大小有一定的限制，L2CAP 將數據資料傳到 Baseband 前，必須將其分段切割，以使 Baseband 不必處理不同

高層協定產生的不同大小的封包，只須專注處理由 L2CAP 傳來固定大小的封包，如此將使 Baseband 階層的運算更加簡單化，切割後的 L2CAP 區段最長為 64 Kbytes。同樣的，在接收方的 L2CAP 協定必須能將多個 Baseband 封包重組為一個 L2CAP 區段傳往高層。

L2CAP的分段與重組(SAR)功能將L2CAP 的區段(Segment)切割後，放置進Baseband階層的封包內，Baseband階層的封包只以少量的Overhead表示封包傳送的是LMP或是L2CAP的資料，當封包內的Payload Header內的L_CH 兩個位元的數值為L_CH=10時，表示該封包傳送的是L2CAP經過切割後的第一筆片段，L_CH=01表示該封包傳送後續的L2CAP片斷資料。若是L2CAP沒有切割任何的區段時，Baseband上的封包則都以L_CH=10來表示傳送L2CAP0的資料，當L_CH=11時表示該封包傳送LMP的資料。

在L2CAP區段的傳輸過程中，為了避免其他區段的封包中途加入造成封包排列順序的改變(Out-of-Order)，於重新組合時發生問題，必須等到同一區段的封包傳送結束後，才能繼續傳送其他區段的封包。Baseband內的Stop and Wait協定機制也會驗證在下一個封包送出前，接收端已經接受到正確的封包，避免非順序(Out-of-Order)的傳輸發生，這些都屬於L2CAP與Baseband控制訊號流量的方法。

由於Baseband協定不能識別任何高層協定，所以L2CAP必須支援上層協定多工(Protocol Multiplexing)，將高層協定的不同種類、大小的封包經過分段後，統一封裝在L2CAP中。利用L2CAP封包的PSM欄位來指定對應的高層協定，如對RFCOMM則設PSM=0x0003。接收端收到封包後，將依此值辨識該對應的上層協定[28]。

2.11 連結管理器協定(LMP)

連結管理器協定用來對連結進行設置和控制。接收方連結管理器透過該協定，對接收到的信號進行識別和篩選，而不再將接收信號轉發到更高的協定層。

LMP協定可直接發送有效負載，而不以L2CAP方式來發送，同時透過有效負載表頭的L_CH欄位保留值來區別不同的發送方式。該信息直接由接收方的LM過濾和解釋，同樣也不將它轉發到更高的協定層。

連結管理器的信息發送比用戶資料的發送具有更高的優先順序，也就是說，雖然連結管理器的信息發送會被個別基頻中的資料分組發送延遲，但它不能被L2CAP的通信延遲。LMP PDU是以單時槽分組的方式發送，因此有效負載表頭只占一個位元組，有效負載表頭的兩個最低位元用來確定邏輯通道。每個PDU都分配了一個位元的執行識別碼（TID, Transaction Identify）用來標識不同類型的PDU。

連結管理器負責設定和管理藍芽的基頻連線，連結管理器由對基頻的控制來建立ACL連結；然後LMP訊息可以經由以已經建立好的ACL連線來建立SCO（聲音）連結。LMP作業如圖2-7所示，在圖中，其複雜的路徑牽涉到Master和Slave的角色轉換，有許多兩者擇一的路徑，在經過LC以呼叫和呼叫回應之後，最簡單的路徑就是以交換訊息來確定連線建立已經完成。

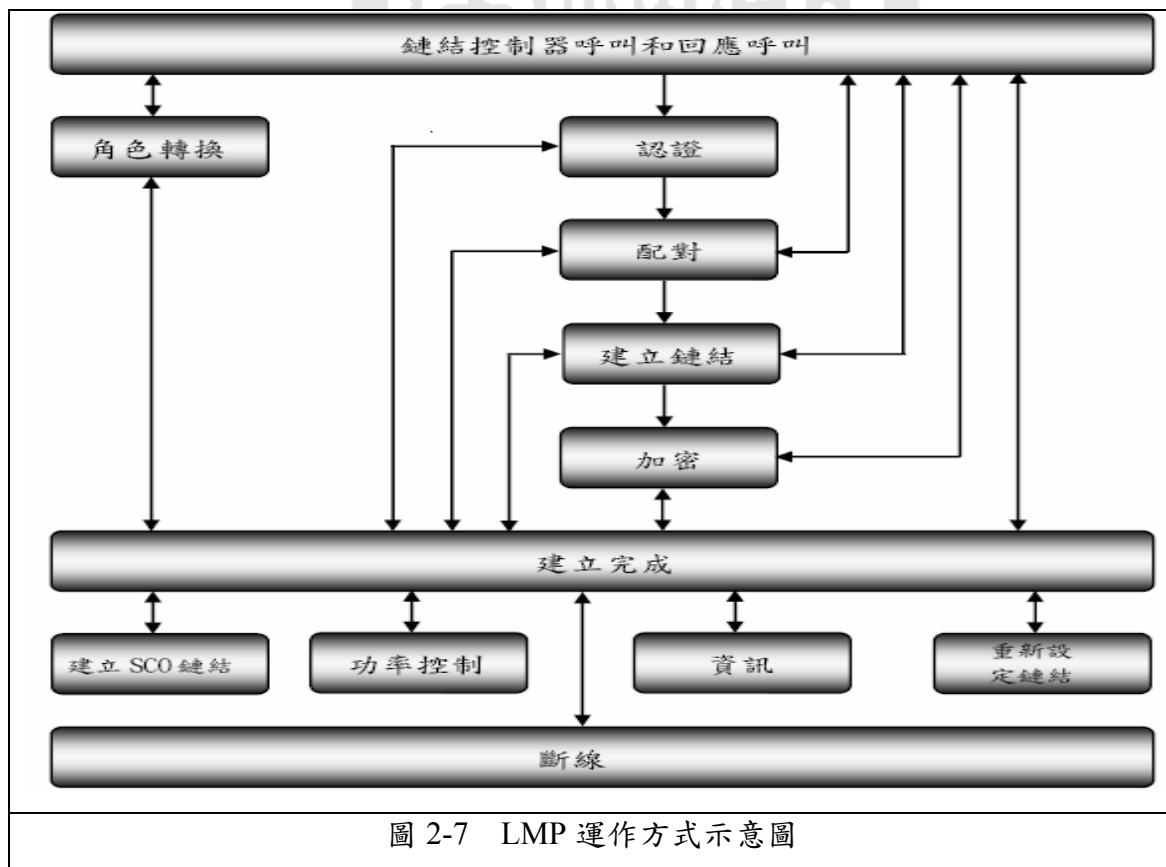


圖 2-7 LMP 運作方式示意圖

2.12 藍芽通訊協定(Profiles)簡述

在此只針對本論文中，所使用到的部份，加以介紹。

2.12.1 GAP(Generic Access Profile)功能描述

Generic Access Profile是所有藍芽設備的基礎，所有藍芽設備的設計都必須符合Generic Access Profile的規定，以確保與其他藍芽設備的互通性(Interoperability)。Generic Access Profile的內容描述了Bluetooth 設備的基本運作方式與參數定義，主要包括與其他設備連接的Discovery of Bluetooth Devices、Link Management、安全性等級(Different Security Levels)、使用者介面(User Interface Level)、操作模式(Modes of Operation)等方面。

2.12.2 SDAP(Service Discovery Application Profile)功能描述

當使用者對藍芽設備下達服務搜尋的動作後，藍芽設備能夠搜尋周遭附近有哪些Bluetooth設備，或是詢問某特定的藍芽設備提供哪些服務，例如若發現傳真機則可連線作傳真、若發現印表機則可進行資料列印，這就是Service Discovery Application Profile定義的功能

2.12.3 SPP(Serial Port Profile)功能描述

簡單來說，Serial Port Profile 就是模擬串列埠(Emulate Serial Port)，使現在使用串列埠RS232介面的應用軟體，都能操作在藍芽設備上，以Serial Port Profile 為基礎的Profile，有Headset Profile、Dial-up Networking Profile、FAX Profile、LAN Access Profile等。

2.12.4 GOEP(Generic Object Exchange Profile)功能描述

Generic Object Exchange Profile為支援物件交換的協定，當藍芽設備內的應用程式為Object Push、檔案傳輸(File Transfer)、資料同步(Synchronization)時，由於這些應用程式的傳輸都是以物件交換(Object Exchange)的形式，所以必

須利用到Generic Object Exchange Profile的功能。

Generic Object Exchange Profile仍然是以Serial Port Profile為基礎，並在Generic Object Exchange Profile的基礎上，發展出Object Push Profile、File Transfer Profile及Synchronization Profile。

2.12.5 OPP(Object Push Profile)功能描述

將筆記型電腦、掌上型電腦、行動電話等設備內儲存的物件(Object)，如同通訊錄裡面的公事卡(Business Card)或是約會(Appointment)，與其他設備互相交換，包括傳送(Push)物件到另一個設備、從另一個設備接收(Pull)物件、或是與另一個設備交換(Exchange)物件。

2.12.6 FTP(File Transfer Profile)功能描述

從File Transfer Profile的名稱顧名思義就是檔案傳輸，例如筆記型電腦、掌上型電腦、行動電話等設備，能夠將檔案(File)與資料夾(Folder)等物件(Object)，傳遞到另一個設備上，並且還能瀏覽與控制另一個設備內的檔案管理系統就是File Transfer Profile定義的功能，與Object Push Profile不同之處即是兩者所傳輸的物件(Object)格式不同。

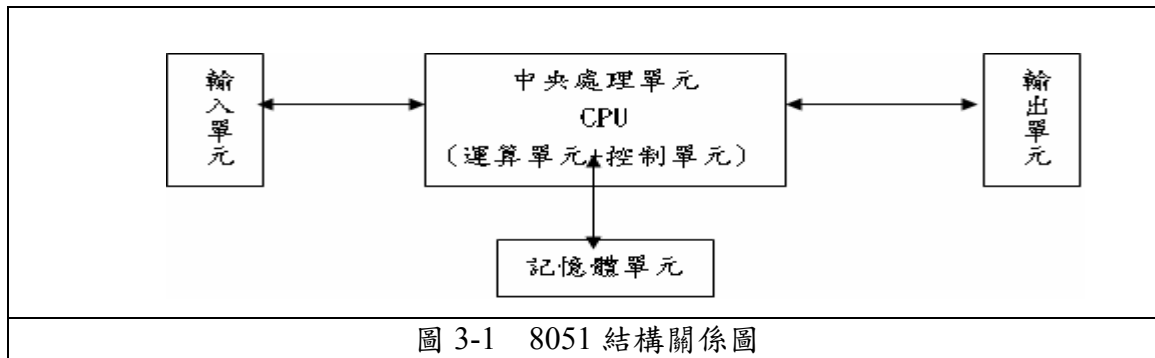
第三章 系統整合架構原理介紹

3.1 處理器簡介

3.1.1 微處理器8051

3.1.1.1 8051硬體結構簡介

8051為現在最通用且最簡單微處理器，其硬體結構包含中央處理單元、記憶體單元、輸入單元與輸出單元等四個主要單元，其結構關係則如圖3-1所示。



其中，中央處理單元則是由運算單元與控制單元兩部分所組成的單元，即是一般所通稱的 CPU(Center Processing Unit)，此為微電腦最重要的部分。以下就微電腦中各單元的功能做簡單介紹：

1. 運算單元(Arithmetic Logic Unit，簡稱 ALU)：

運算單元又稱為算數邏輯單元，在中央處理單元中可用於執行算數運算及邏輯運算，能將記憶體單元或輸入單元送至中央處理單元的資料執行各種運算。當運算完成後再由控制單元將結果資料送至記憶體單元或輸出單元。

2. 控制單元(Control Unit，簡稱 CU)：

此單元在中央處理單元中，負責協調與指揮各單元間的資料傳送與運作，使得微電

腦可依照指令的要求完成工作。在執行一個指令時，控制單元先予以解碼(Decode)，瞭解指令的動作意義後再執行(Execute)該指令，因此控制單元將指令逐一執行，直到做完整個程式的所有指令為止。

3. 輸入單元(Input Unit，簡稱 IU)：

此單元是用以將外部的資訊傳送到 CPU 做運算處理或存入記憶體單元，一般為電腦的輸入單元有鍵盤、磁碟機、光碟機、掃描器或讀卡機等週邊設備。

4. 輸出單元(Output Unit，簡稱 OU)：

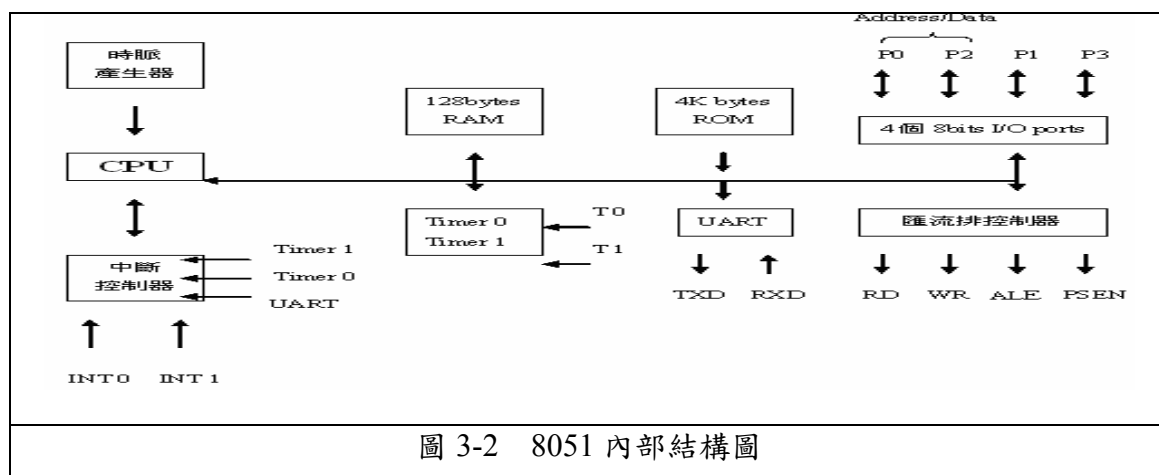
此單元是用以將 CPU 處理過的資料輸出或儲存傳送外部週邊設備，一般為電腦的輸出單元有顯示器、印表機或磁碟機等週邊設備。

5. 記憶體單元(Memory Unit，簡稱 MU)：

是用來儲存輸入單元傳送來的資料，或儲存經過中央處理單元處理完成的資料。記憶體單元之記憶體可分為主記憶體(Main Memory)與輔助記憶體(Auxiliary Memory)兩種，而主記憶體依存取方式不同，又可分為唯讀記憶體(Read Only Memory，簡稱 ROM)與隨機存取記憶體(Random Access Memory，簡稱 RAM)。

3.1.1.2 8051 內部結構簡介

8051 為 Intel 公司的 MCS-51 系列產品之一，其內部結構如圖 3-2：



8051 單片具有以下之特性：

1. 專為控制使用所設計的 8 位元單晶片。
2. 具有位元邏輯運算能力。
3. 具有 128 位元的 RAM，以及 4K 位元的 ROM。
4. 具有 4 個 8 位元 I/O 埠。
5. 具有 2 個 16 位元的計時/計數器。
6. 具有全雙工的 UART。
7. 具有 5 個中斷源及兩層中斷優先權結構。
8. 具有時脈產生電路。
9. 具有外部電路擴充 64 位元程式記憶體的能力。

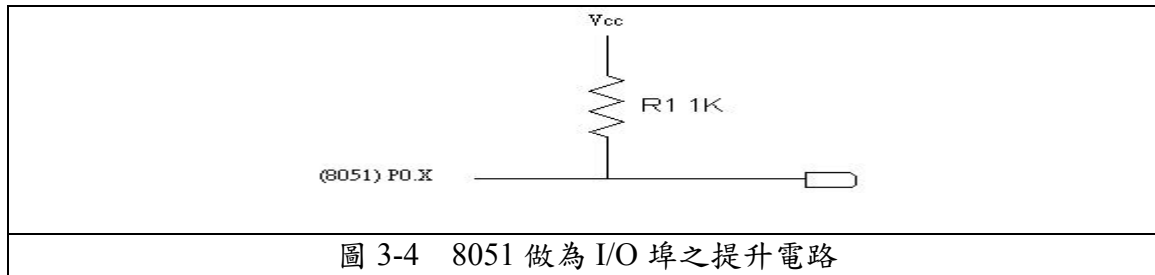
3.1.1.3 8051 單晶片的接腳

8051 為 40 支接腳之單晶片，圖 3-3 指出其接腳與功能說明如下：

P1.0	1		40	Vcc
P1.1	2		39	P0.0/AD0
P1.2	3		38	P0.1/AD1
P1.3	4		37	P0.2/AD2
P1.4	5		36	P0.3/AD3
P1.5	6		35	P0.4/AD4
P1.6	7		34	P0.5/AD5
P1.7	8		33	P0.6/AD6
RST	9	8	32	P0.7/AD7
RXD/p3.0	10	0	31	\overline{EA}
TXD/P3.1	11	5	30	ALE
$\overline{INT0}$ /P3.2	12	1	29	\overline{PSEN}
$\overline{INT1}$ /P3.3	13		28	P2.7/A15
T0/P3.4	14		27	P2.6/A14
T1/P3.5	15		26	P2.5/A13
\overline{WR} /P3.6	16		25	P2.4/A12
\overline{RD} /P3.7	17		24	P2.3/A11
XTAL2	18		23	P2.2/A10
XTAL1	19		22	P2.1/A9
GND	20		21	P2.0/A8

圖 3-3 8051 接腳圖

1. Vcc：+5 電源供應接腳。
2. GND：接地接腳。
3. P0.0~P0.7：埠 0，為開洩極(Open Drain)雙向 I/O 埠。在做為外部擴充記憶體時，可低八位元位址線(A0~A7 address line)與資料匯流排(data bus)雙重功能。在做為一般 I/O 埠，必須加上如圖 3-4 之外部提升電路。



4. P1.0~P1.7：埠 1，為具有內部提升電路的雙向 I/O 埠。
5. P2.0~P2.7：埠 2，為具有內部提升電路的雙向 I/O 埠。在做為外部擴充記憶體時，可為高八位元位址線(A8~A15 address line)。
6. P3.0~P3.7：埠 3，為具有內部提升電路的雙向 I/O 埠。此外，埠 3 的每支接腳都具有另一特殊功能，其功能如下：
 - i. RXD(P3.0)：串列傳輸的接收端。
 - ii. TXD(P3.1)：串列傳輸的輸出端。
 - iii. $\overline{INT} 0$ (P3.2)：外部中斷輸入端。
 - iv. $\overline{INT} 1$ (P3.3)：外部中斷輸入端。
 - v. T0(P3.4)：計時/計數器外部輸入端。
 - vi. T1(P3.5)：計時/計數器外部輸入端。
 - vii. \overline{WR} (P3.6)：外部資料記憶體寫入激發信號(Strobe)。
 - viii. \overline{RD} (P3.7)：外部資料記憶體讀取激發信號(Strobe)。
7. RST：重置信號(Reset)輸入端。在單晶片工作時，將此腳位保持在“Hi”機械週期，CPU 將重置。

8. ALE：位址鎖住致能(Address Latch Enable)，在每個機械週期都會出現，可做為外部電路的時脈源。
9. \overline{PSEN} ：程式激發致能(Program Strobe Enable)，可輸入外部程式記憶體的讀取信號。
10. \overline{EA} ：外部存取致能(External Access Enable)，當EA接腳為“L0”時，則讀取外部程式記憶體執行。
11. TAL1：反相振盪放大器的輸入端。
12. TAL2：反相振盪放大器的輸出端。其基本電路連接如圖 3-5：

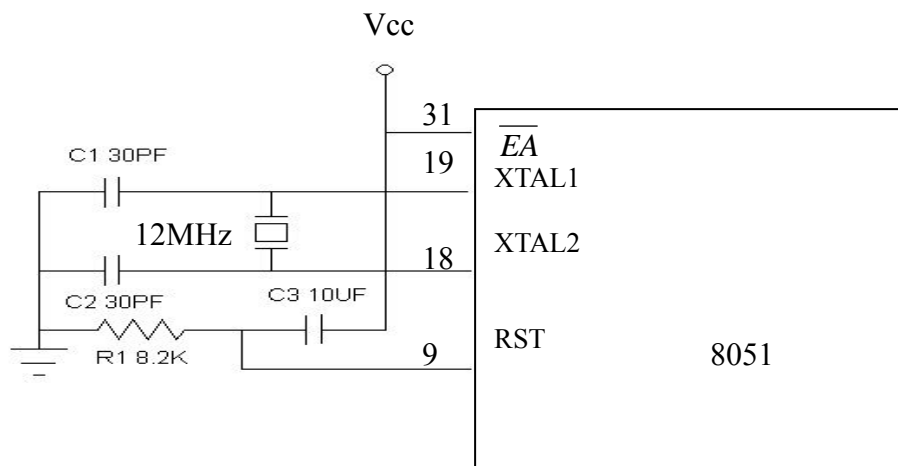


圖 3-5 8051 之致能電路

3.1.2 TI的MSP430超低功耗微控制器

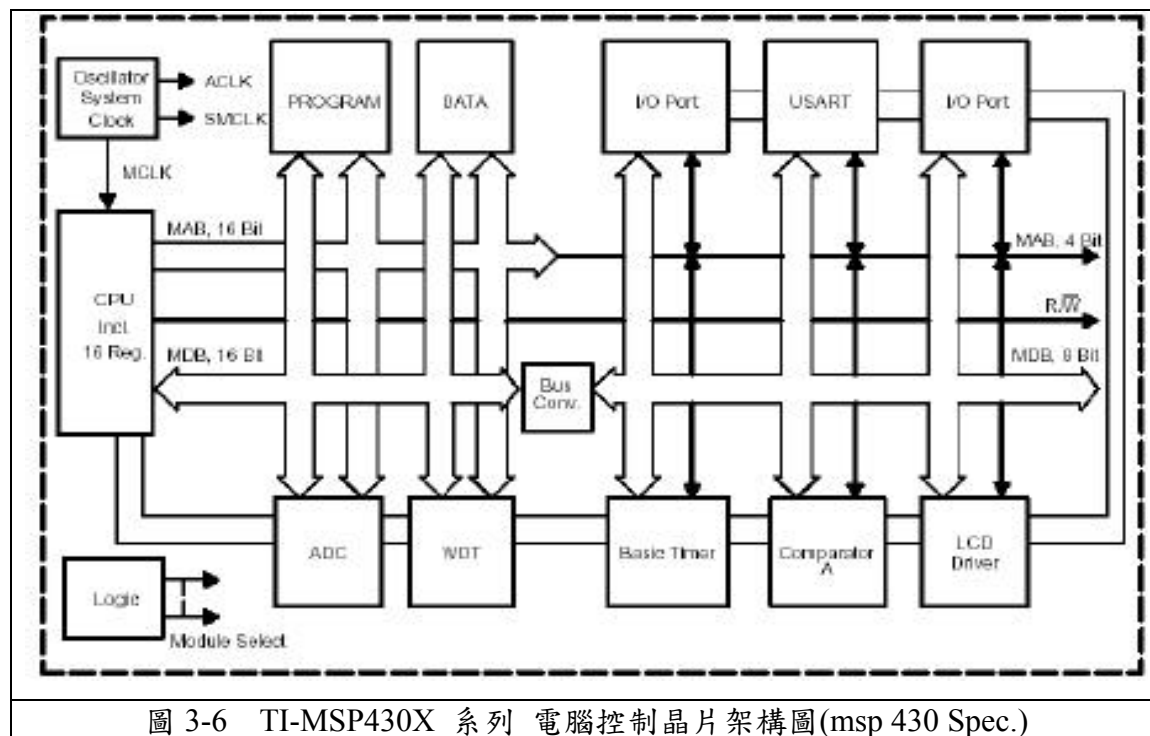
3.1.2.1 MSP430 簡介

TI提供MSP430超低功耗16位元RISC混合訊號處理器，為電池供電型量測應用帶來解決方案。TI利用它在混合訊號和數位技術發展出MSP430家族，使設計人員能同時連接類比訊號、感測器和數位零件，並維持業界最佳的低功耗效能。

1. 超低功耗架構延長電池使用時間
 - i. 0.1uA RAM 資料保存電流
 - ii. 0.8uA 即時時脈模式

- iii. 250uA / MIPS 工作電流
- 2. 高效能類比元件，最適合精準量測應用
- 3. 新世代 16 位元 RISC 處理器，可實作多種新應用，並大幅縮短程式碼長度
- 4. 內建可線上燒錄 (In-system Programmable) 快閃記憶體，可用於程式碼變更、現場升級和資料記錄
- 5. 元件單價便宜且具功能齊全的整合發展環境

系統微電腦控制晶片採用Texas Instruments德州儀器公司 (TI) MSP430系統，晶片為MSP430F147，這顆微電腦控制晶片架構如(圖3-6)，包含有低頻振盪器FLL+clock 系統 (on-chip DCO+2 晶體振盪器)、12 bit 類比訊號轉數位訊號、Low Supply-Voltage Range、1.8 V to 3.6 V、16 Bit RISC Architecture CPU、60K Flash Memory、2K SRAM、三個時脈產生器、一組雙方輸入輸出PORT、I/O Port 1, 2、I/O ports 3, 4, 5, 6、兩個UART、共有100支接腳，屬於高階控制處理晶片。



3.1.2.2 MSP430工作模式

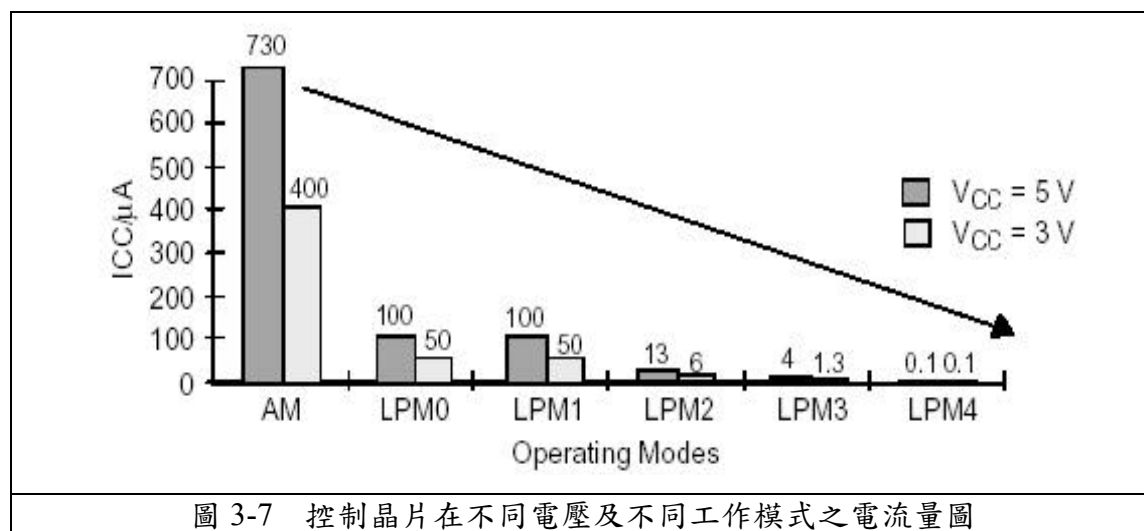
MSP430家族主要的特色是極低的工作電壓，電壓的消耗可分為三種工作組態：

1. Active Mode： 280 μA at 1 MHz, 2.2 V
2. Standby Mode： 1.1 μA
3. Off Mode (RAM Retention)： 0.1 μA

選擇TI-MSP430的晶片組，主要是它一顆晶片就包括LCD driver，ADC，Low Power Consumption，更大的Flash memory，16 Bit CPU，USART硬體，做為可攜式系統儀器最主要的要求就是輕巧、低功率消耗、龐大的資料庫、快速分析計算。

MSP430 家族中3X系統以上的控制晶片均含有六種不同的工作模式，分別為AM、LPM0、LPM1、LPM2、LPM3、LPM4。

其中MCLK 為主要振盪頻率，ACLK(Auxiliary)為輔助用振盪頻率(倍頻振盪頻率)，在圖3-7可以看出控制晶片在不同電壓及不同工作模式所需要的電流量，一般工作模式典型值在3 V 250 μA ，但工作在LMP4模式時的工作電流只有0.1 μA 。



3.1.3 Windows XP SP1作業系統

Windows XP SP1是Microsoft的Windows®作業系統，其核心引擎建立在Windows 2000 (Kernel) 基礎之上，適用於現在一般的桌上型PC及筆記型的NB，同時還提供了許

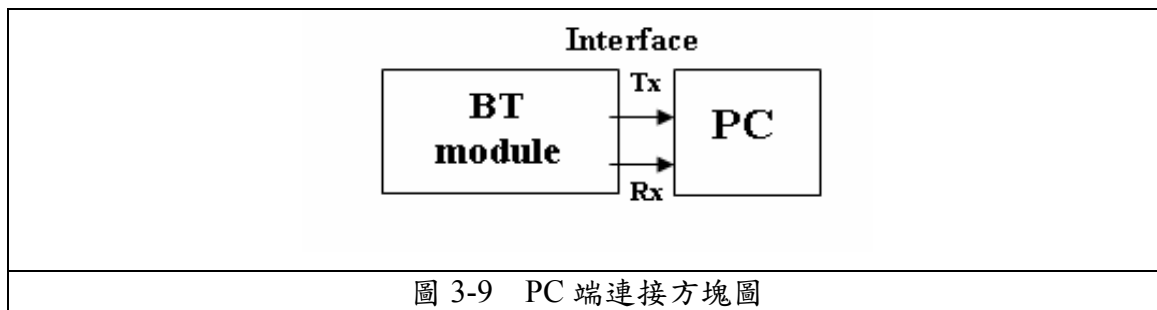
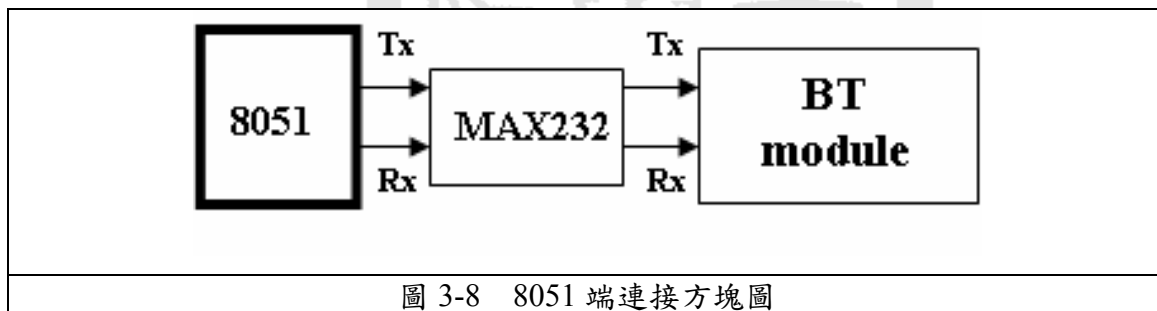
多全新的功能，讓電腦處理更多的事情。因此Windows XP SP1系統會比過去的作業系統版本更加穩定，所提供的工具可保持隨時隨地，甚至是在工作以外也可與Internet的連線，並可與合作夥伴、客戶、檔案，以及應用程式保持聯繫，且32位元的Windows 2000與Me軟體都可以在這個新的作業系統上運作，甚至有DOS虛擬機器（Virtual Machine）功能，作業系統還會自動設定相容性參數，好讓作業系統能跑沒有測試過的舊軟體，具較高的相容性。

3.2 系統整合原理

3.2.1 藍芽與8051整合

3.2.1.1 系統概述

此系統主要是將微處理器8051與藍芽模組整合,設定8051端的pin腳定義，先設Interface，預設TX和RX的兩隻pin腳，先連接到MAX232(電位轉換單晶片)，再與藍芽模組拉出的TX和RX相接，以圖3-8的架構為藍芽無線系統的8051端，另一端圖3-9為PC端，是以藍芽模組配合PC成無線系統的一端，用來傳送或接收資料及字元。



3.2.1.2 系統特色

此藍芽與8051整合系統主要具有下列特色：

1. 低功率：

本系統是使用藍芽無線傳輸技術，功率消耗低，這不但是發展嵌入式系統的重要依據，更可降低電磁波對人體的危害。

2. 高度行動性：

現在藍芽整合的系統，不用固定放於某個位置，即可在有效的距離範圍中，進行連線與收發資料。

3. 點對多點傳輸：

此系統可提供點對多點應用，當室內同時有幾個藍芽裝置時，也可利用SDP同時找到多個藍芽裝置並呼叫，達成點對多點呼叫。

4. 資料保密性：

此系統具有授權、認證及加密功能，確保資料保密性。

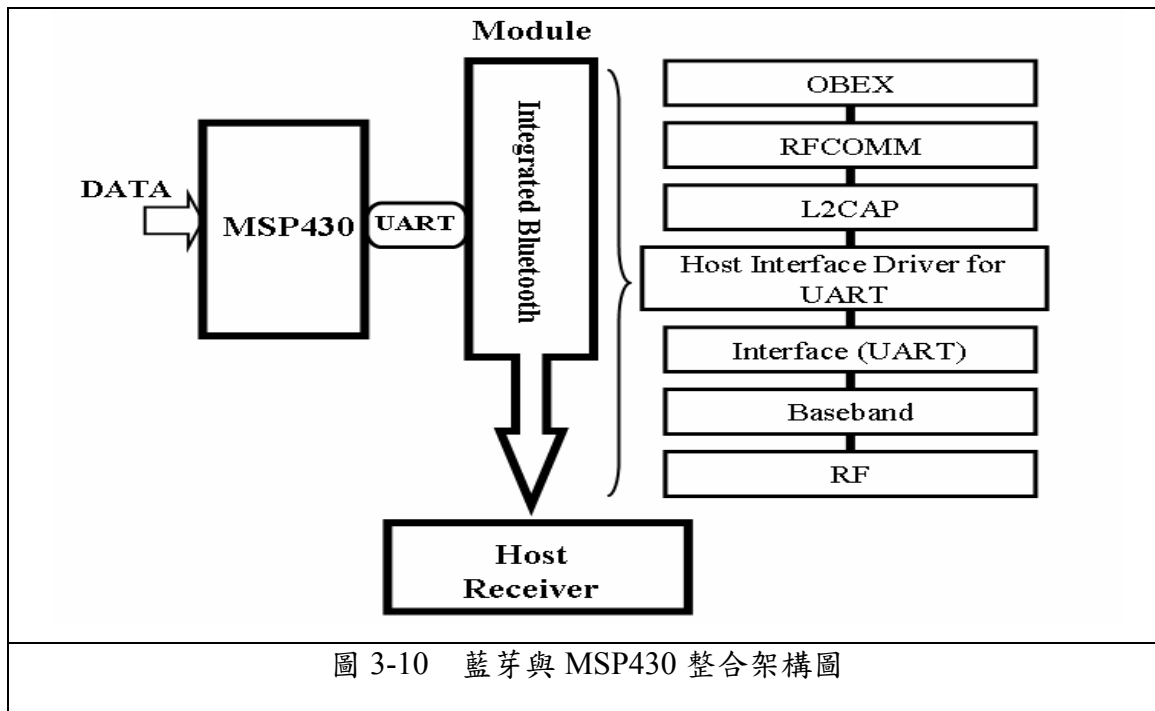
5. 優於紅外線傳輸：

藍芽克服了紅外線傳輸在距離及可連接裝置數的限制。紅外線裝置，只能夠做一對一的連線，也就是說一部電腦同時只能與另一部電腦透過紅外線連線，無法同時與第三部或更多部電腦連線，而且連線的距離有限，頂多只能在兩個距離不超過五公尺的桌子上，面對面傳送資料。然而，藍芽的有效距離可達十公尺，而且運作時最多可同時提供Master與七台藍芽裝置的互相連線，除此之外藍芽還具有穿透性、無方向限制等特性，均是紅外線傳輸所望塵莫及的。

3.2.2 藍芽與MSP430整合

3.2.2.1 系統概述

使用 TI 的 MSP430，配合藍芽晶片作一個無線傳輸的模組，如圖 3-10 所示。



主要先建立傳送資料的機制，之後再以程式控制到硬體層，使由外部輸入的數據或資料，可由自行建立資料傳送機制，由藍芽傳送模組以無線的方式傳送到主機上，亦可依不同的應用來選擇資料的互傳或對連。

而藍芽系統主要由無線電部份(射頻，天線)、鏈結控制器(LC，Link Controller)、鏈結管理器與輸出入(LM，Link Manager & I/O)，和主機(Host)部份構成。鏈結控制器是負責反應從 High Layer 的鏈結管理器發出命令，它在幾個資料封包期間內，完成鏈結層的運作，近端和遠端的鏈結控制器再藉由鏈結管理器發出建立連線命令的封包處理程式，以管理封包並維持已建立的連線。再配合軟體層的程式控制，就可以用整合 MSP430 的藍芽無線系統來傳遞和接收資料。

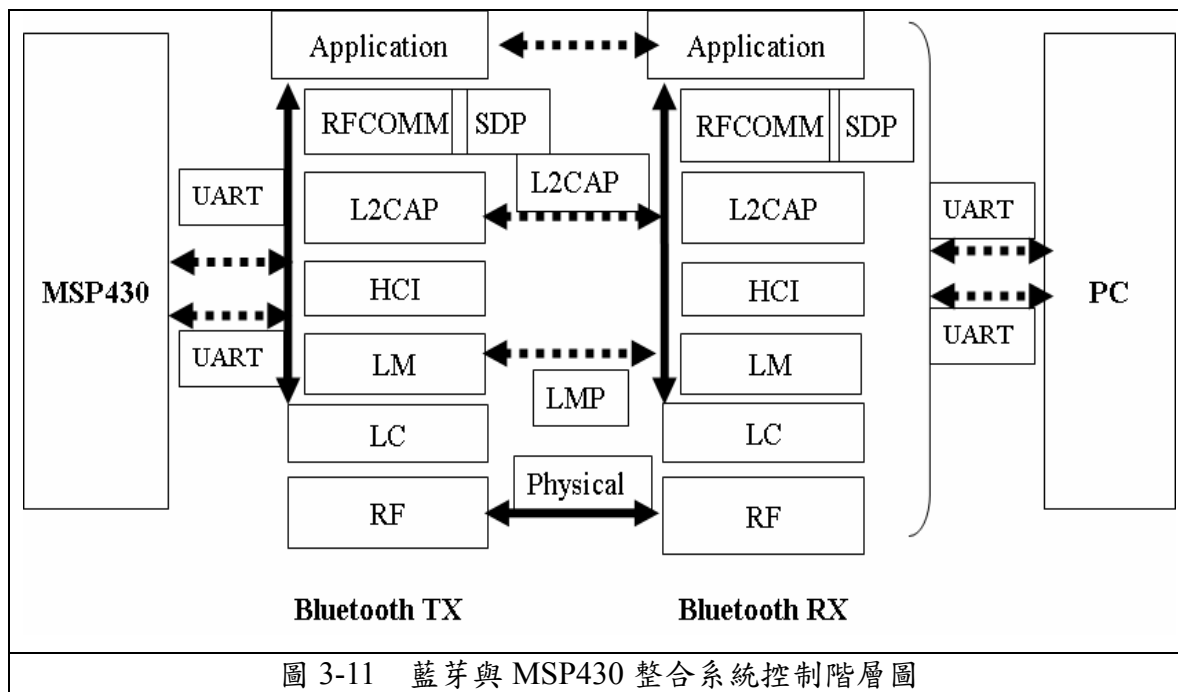


圖 3-11 藍芽與 MSP430 整合系統控制階層圖

圖3-11為其系統整體架構控制階層，分為MSP430部份及藍芽模組部份：

1. MSP430部份：

連接介面為UART，在MSP430中，主要是看Specification，將程式配合介面，做將輸入資料存到Buffer，之後直接將資料經UART介面傳至藍芽模組。

2. 藍芽模組部份：

主程式分為兩端，一為 Master TX；另一端為 Slave RX。Master 端是建立一個資料傳輸 Protocol 及 Profile 去連結 Slave 程式，為輸出端。Slave 端接收藍芽晶片在介面之間的連結(Connection)與傳送資料，可用來對應 Master 的應用程式，且可以從 PC 端對應跑適合的藍芽堆疊，為接收端。

3.2.2.2 系統特色

此藍芽與MSP430整合系統，具備有8051系統的特色之外，並具有下列特色：

1. 實用率：

系統使用現在工業界最常使用的TI MSP430系列處理器，將其整系系統發展完全，

則可直接套用到工業界，進行藍芽儀器的發展。

2. 儀器更小化：

現在藍芽整合的系統，只用到一個藍芽模組進行介面連結，之後，僅僅只要將藍芽晶片配合RF電路，就可以有無線傳輸功能，不用在旁外加資料處理系統。

3. 方便性：

此系統整合後，若是有一個有分析儀器，對多項儀器得出來的數據，無法做統整在一個資料處理器中做分析，但轉於本系統後，可以同時由Server端將所有Slave端儀器的資料，無線接收，並可同時做分析與比較。

3.2.3 藍芽與Windows作業系統整合

在此系統之下，主要分為兩個應用系統：

1. Headset系統：

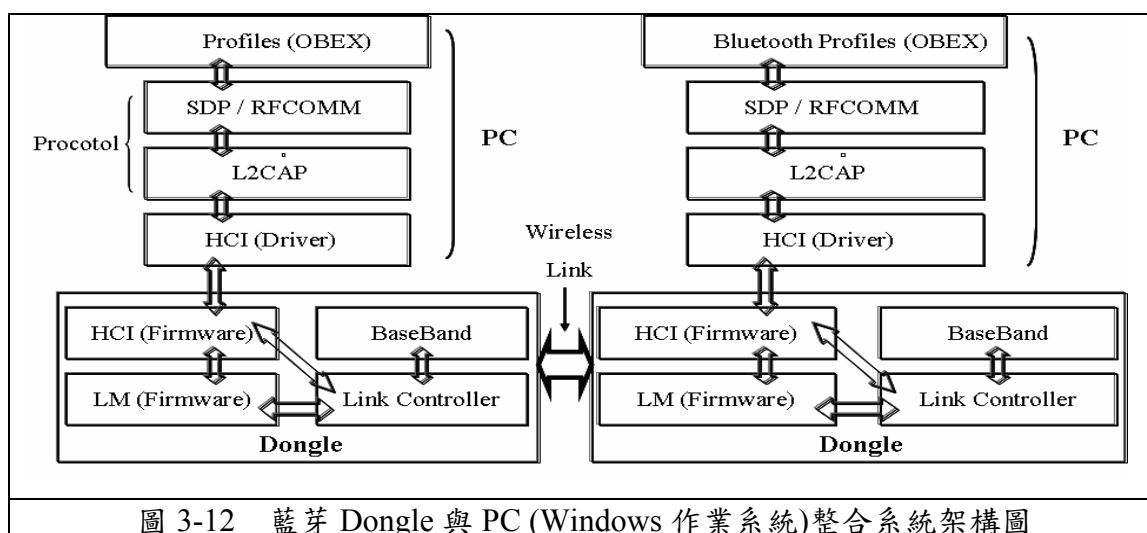
為用藍芽Dongle做為兩端的藍芽傳輸器，與PC所整合的系統。在建立Headset的機制之前，會先架構Protocol層，及檔案傳輸的Profile(FTP)，先做檔案傳輸的實驗，再將Headset Profile架上系統，來完成聲音的傳送[10]。

2. 藍芽互動控制系統：

是將藍芽手機做為傳輸器，再與PC結合成一個藍芽互動控制的系統。主要是架構在Headset系統之下，在電腦以Dongle做為藍芽端，只有架構FTP，在配合一個檔案辨識系統，由藍芽手機發出的檔案來進行遠端控制及傳輸。

3.2.3.1 Headset系統概述

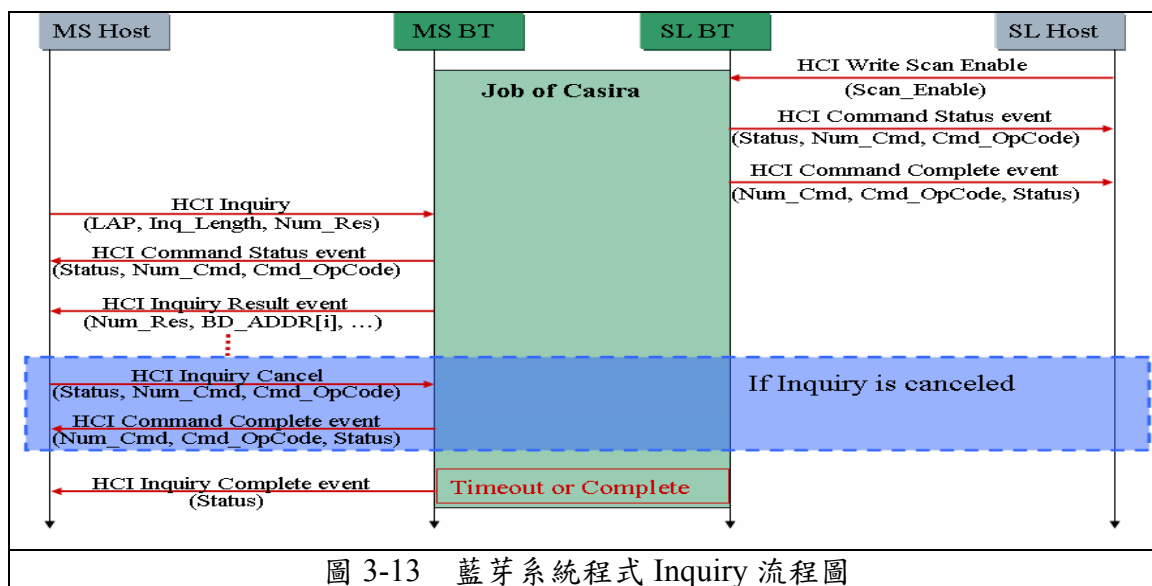
由於Window XP為一個強大的作業系統，配合PC可支援的應用程式及驅動程式，則成一個多工的處理器系統，且PC上已具有Print Port、UART Port及USB Port，因此在整合藍芽與Windows系統，主要是作為應用的用途，藍芽端採用USB Bluetooth Dongle(USB藍芽傳輸器)做為一個無線的裝置，來溝通兩個藍芽無線系統，圖3-12為此系統之架構圖。



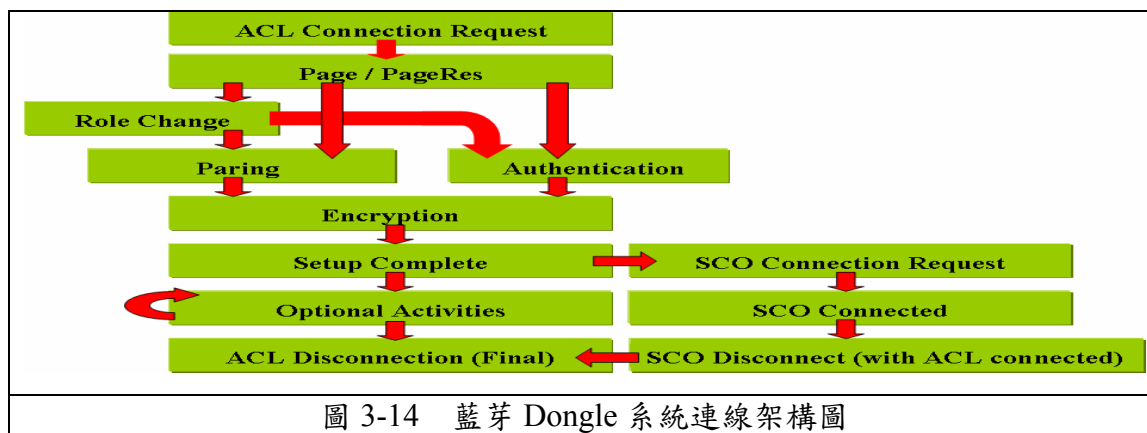
使用 CSR 的 USB HCI 驅動程式，來驅動藍芽 USB Dongle，而依藍芽規格，自行撰寫 HCI 上層的檔案傳輸程式來整合藍芽與 Window XP 作業系統，其上層包括 HCI 層設定、Protocol (L2CAP、SDP、RFCOMM、OBEX...) 及 Profile (Headset) 等部份的程式撰寫 [3]。

3.2.3.2 Headset 原理介紹

以系統的訊號流程來解釋程式執行流程，以 MSC (Message Sequence Chart) 來說明本系統的程序。



藍芽Dongle系統程式中，如圖3-13，對於HCI層程式，一開始會先做Inquiry的動作，在Master端，先設定Inquiry的機制進行Inquiry，MS藍芽端會回HCI命令狀態事件，及Inquiry後的結果，並將回應及對方藍芽位址傳回，之後結束Inquiry事件。而在Slave端，先開啟SL藍芽Scan的機制，SL藍芽端會回HCI命令狀態事件，並傳回結束HCI命令結束事件。但若是主動中斷Inquiry，就會直接傳回結束HCI命令結束。



連線建立流程如圖3-14所示。在連線流程中，先有一個ACL連線的要求，在經過Master和Slave位址互通之後，進行配對(Page)，配對成功後，ACL設置完成，再加上SCO連線要求，建立SCO連線。如此就可以建立傳送聲音資料(SCO)連線。

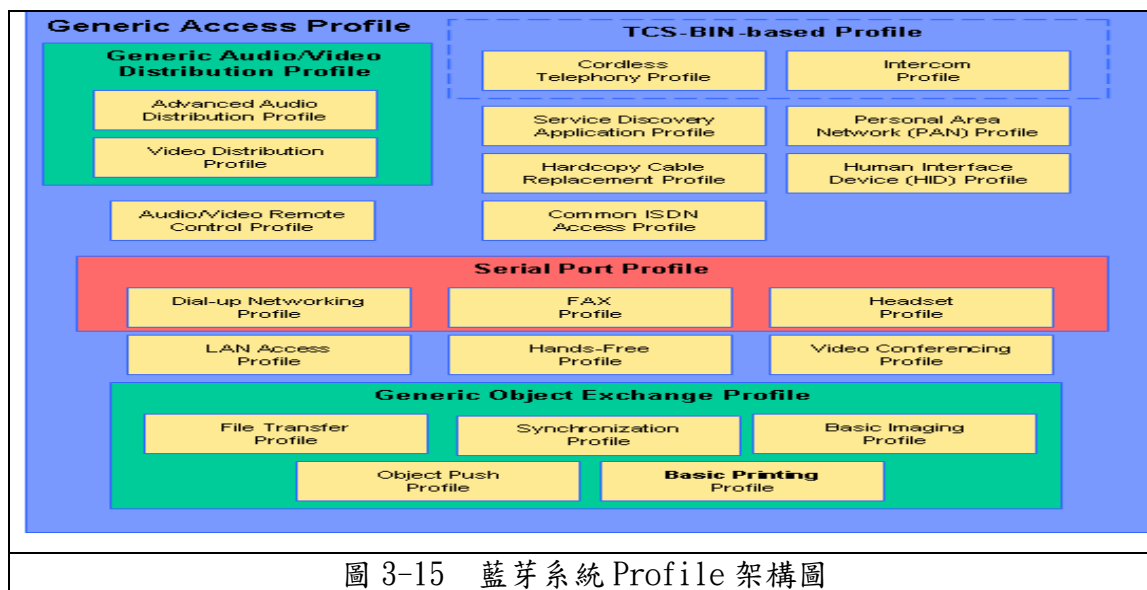


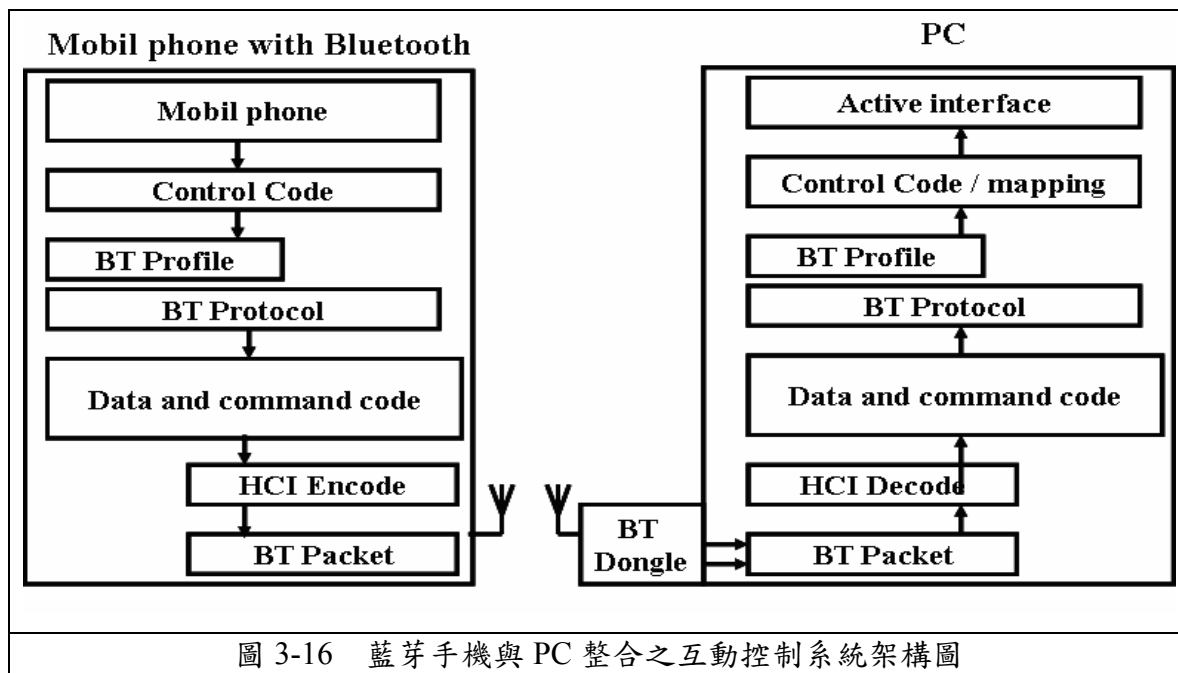
圖3-15為Profile架構的方式，在架設FTP及Headset Profile皆可由此架構定義，再照“Bluetooth System Profile Version Specification”中設定即可。

3.2.3.3 藍芽互動控制系統概述

此系統設計主要是將藍芽手機做為一個傳輸控制器，再與PC結合成一個藍芽遠端控制系統，是將藍芽手機做為Master端，再與PC加上一個藍芽Dongle為Slave端，結合成一個藍芽互動控制的系統。架構在Headset系統上，寫入FTP來進行檔案傳輸，在配合一個檔案辨識系統，辨認由藍芽手機發出的檔案，是純粹的資料或是命令，來進行遠端控制及傳輸檔案[13]。

3.2.3.4 藍芽互動控制系統原理介紹

控制系統主架構如圖3-16。分為手機端與電腦端兩個部份：



1. 手機端：

為具有藍芽功能的手機端，主要是先將手機上的按鍵Pin碼及其中資料與影像的檔案，

定義其傳送機制與Control Code之後，在藍芽程式中，先改其Profile與Protocol，使其對應Control Code的定義碼,接著將Data及Command Code傳到藍芽的下層HCI，進行編碼及封包切割的動作，將Data及Command Code包成封包型式傳出到PC端。

2. PC端：

由一個藍芽傳輸器(BT Dongle)來負責接收從藍芽功能的手機端所傳過來的封包之後，在HCI進行解碼及封包組合，將其還原為Data及Command Code，接著將其對應到Protocol及上層Profile，並找出其在PC端對應位址及其參數。

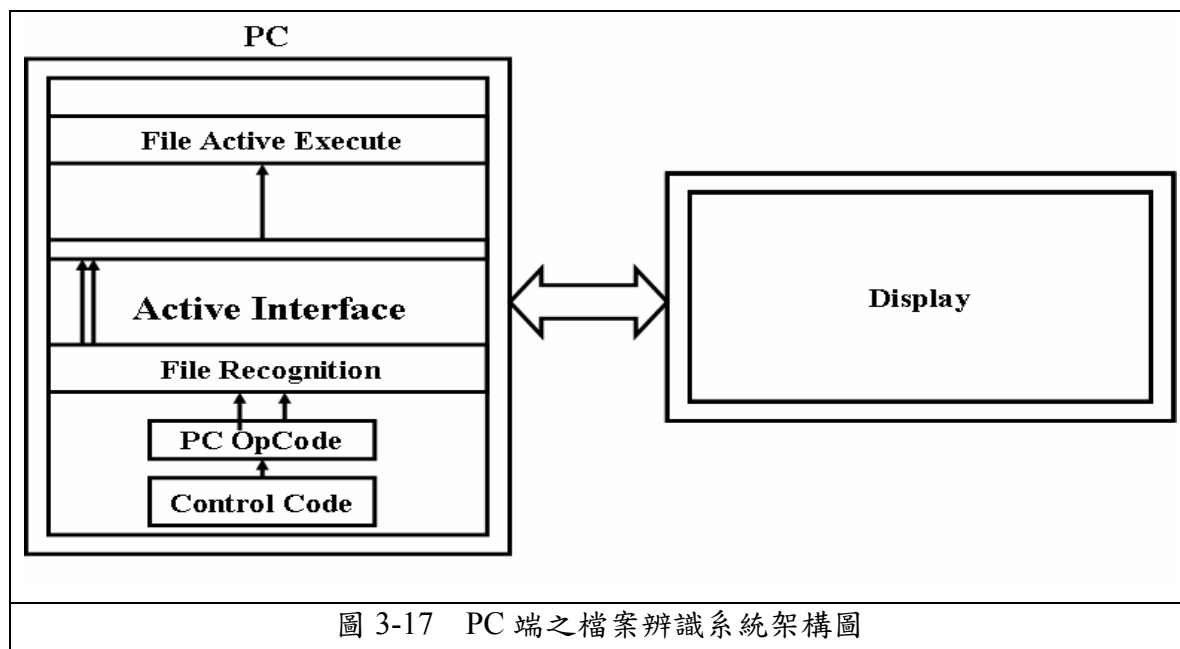


圖 3-17 PC 端之檔案辨識系統架構圖

然後如圖3-17，由對應得到的Control Code及資料格式，做一個Active Interface，為一個檔案辨識系統，在收到指定的Control Code及對應的檔案格式，就可以直接對應到Active Interface的控制，並執行應用程式。直接顯示在螢幕上，來達到可直接由藍芽手機控制的目標。

3.2.3.5 藍芽互動控制系統特色

此手機互動整合系統主要具有下列特色：

1. 方便性：

此系統擁有即時檔案交流的功能及遠端控制功能，讓使用者能以更加便利的方式去獲得即時的資訊，並且只要在遠端進就可進行許多的系統操作流程。

2. 普及性：

因現在藍芽手機相當普及，且PC Windows系統只要外接一個藍芽傳輸器，或是已內建藍芽功能的筆記型電腦，均可用來架構此互動系統。

3. 發展性：

這個系統可以提供使用者互動式的導覽、多媒體檔案之下載或上傳。以系館為例，可以在入口放置一電子看板，學生可以透過此看板繳交作業、下載講義，來賓可以觀賞系館簡介、系館導覽。這個系統也可提供使用者透過藍芽手機來控制會議進行。會議在進行中使用者不需要離開座位，可以直接對參與者進行報告，只需要利用手機將要報告的檔案傳至顯示系統，並利用手機去控制報告的進行即可。

4. 多工性：

將系統改為平行分工機制，則可同時進行控制與傳輸檔案，使控制系統的操作性能較好，適用度也大大提高。

3.2.3.6 藍芽互動控制系統原理

在此針對藍芽協定、檔案辨識系統與滑鼠控制做介紹。

1. 藍芽協定：

與設計Headset程式的設計一樣，但在藍芽端另外設計時槽(Time Slot)分工功能及建立對應藍芽手機SPP來建立與手機連線，並定義OBEX及FTP Profile做檔案傳輸。

2. 檔案辨識系統：

主要為列分析序列取回工作子目錄，在計算空間的配額(Quota)之後，直接指定資料庫中的序列，及分析的項目，以便讓程式自動去資料夾中，取用序列資料。序列檔為Txt檔(選用Txt檔主要是因為資料小很快就可進行傳送及辨識)。在做多序列並列分

析時，有時需要指定每個序列要分析的區域，否則易得到錯誤。這在程式中，只需要在檔名之後以「Begin」和「End」來指定，並加多種判斷式來確認資料正確。確認後，在參照機碼將檔名對應到應用程式，並做執行的動作，在txt的檔名之後的資料，沒有列入辨識，資料就不會干擾到其他程式的運作，而存在指定的資料夾中。如辨識mousecontrol.txt，就包含Txt為主要指令，Mouse為功能，一收到此檔案就會自動開啟滑鼠控制程式。

3. 滑鼠控制程式：

用滑鼠EVENT的控制，在Microsoft網站裡面的MSDN Library中， MouseEventArgs MouseButtons MouseEventHandler等三個Library，可以使用。

程式中設SendInput(UINT nInputs, LPINPUT pInputs, int cbSize)，將手機傳出的AT碼，由藍芽虛擬的Serial Port接收到滑鼠控制程式中，並對手機上的按鍵(上、下、左、右、Double Click)，做對應的控制，如此可以直接用手機控制滑鼠做執行及拖曳…等功能。

此系統建立完成，將可以應用在學校、政府機關、商區或是觀光地區…。與使用者做互動，使之獲得即時的資訊及遠端操作流程，應用層面應該很有發展空間。

第四章 系統設計與操作

4.1 使用儀器介紹

4.1.1 CASIRA

CASIRA是由劍橋半導體所開發的藍芽晶片測試模組，功用為測試藍芽晶片的好壞及燒錄晶片Firmware及設定參數。如圖4-1所示。

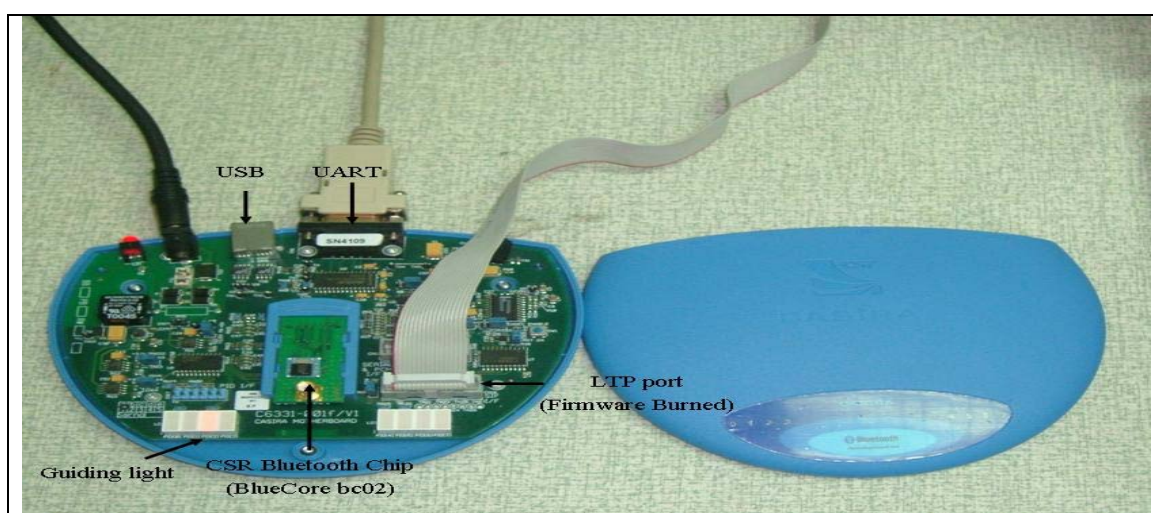


圖 4-1 CSR CASIRA 圖

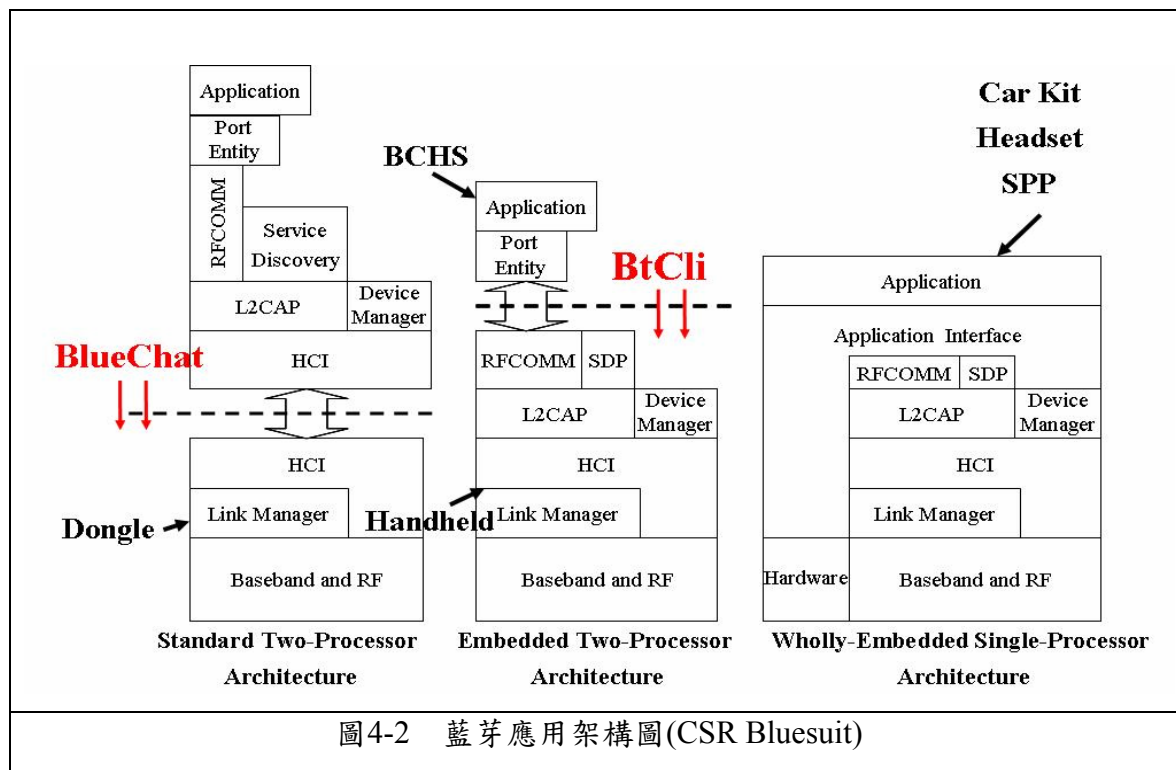
4.1.1.1 系統軟體要求

需Win2000、Win XP SP1作業系統，並安裝CSR BlueSuite軟體，安裝完後在c:\program file下，產生csr /bluesuite 的目錄。產生下列程式：

- i. BlueFlash.exe : 燒錄Firmware用
- ii. BlueChat.exe: 測試傳檔及聲音用
- iii. BlueTest.exe : 測試RF特性用
- iv. Pstools.exe: 修改Firmware參數用
- v. Btcli.exe: Host Control Interface Console

4.1.1.2 CSR BlueSuite的系統對應架構:

圖4-2為現行藍芽整合系統的三個主架構，第一種是由兩個處理器所組成的系統，其兩個處理器由介面連接，兩端都需要介面的驅動，下層的必須控制硬體，上層必需包HCI、Protocol及Profile，為一個最原始的介面連接系統，與BlueChat有同樣的架構。第二為進化版，為一個半整合的系統，下層將Protocol包入硬體，上層架構Profile，系統依照上層Profile設定，控制下層，即具有應用功能。第三為一個完整的整合系統，將藍芽的軟硬體全部都嵌入在一個硬體中，如藍芽耳機及藍芽車用免持器(Car Kit)，已具有全部藍芽功能。



4.1.2 DARFON藍芽模組

由達方(DARFON)電子有限股份公司所自行設計的藍芽模組，其架構為一顆藍芽晶片、控制晶片(Flash)、以及功率放大器、低雜訊放大器、帶通濾波器、Balun、Microphone電路、射頻器及振盪器等。圖4-3為自行到達方電子所做的藍芽模組模擬板，分為兩端，

圖左為Master端，圖右為Slave端。

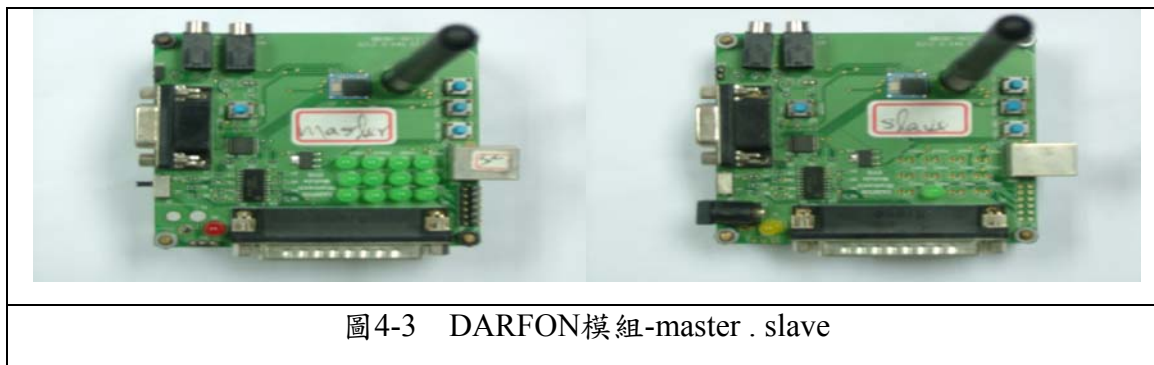


圖4-3 DARFON模組-master . slave

在DARFON模組中，本論文主要是利用到藍芽晶片的UART部份，因此在圖4-4只列出其控制UART的BlueCore腳位，及其電位轉換單晶片的接線圖，如圖4-5所示。

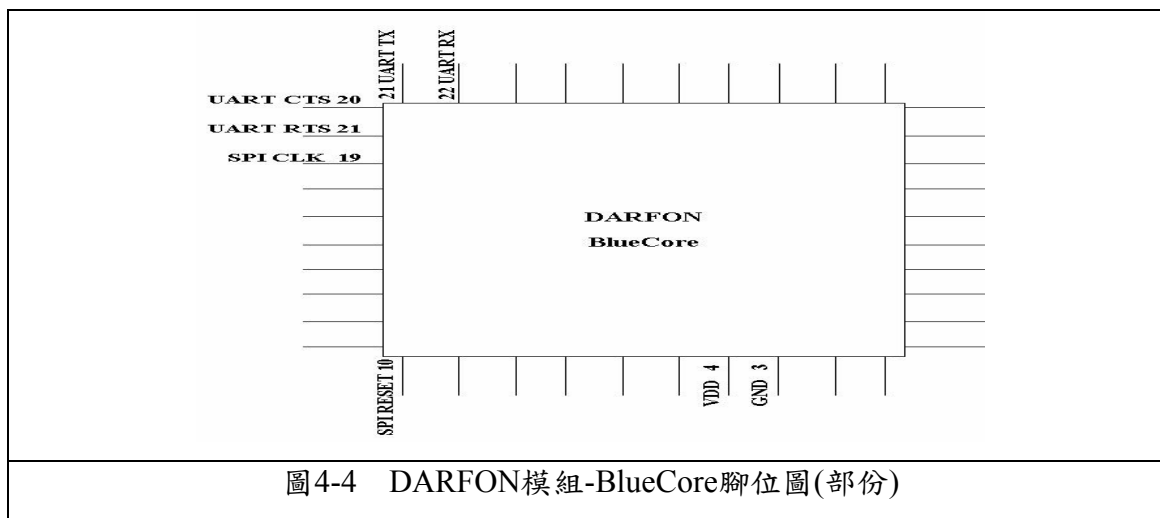


圖4-4 DARFON模組-BlueCore腳位圖(部份)

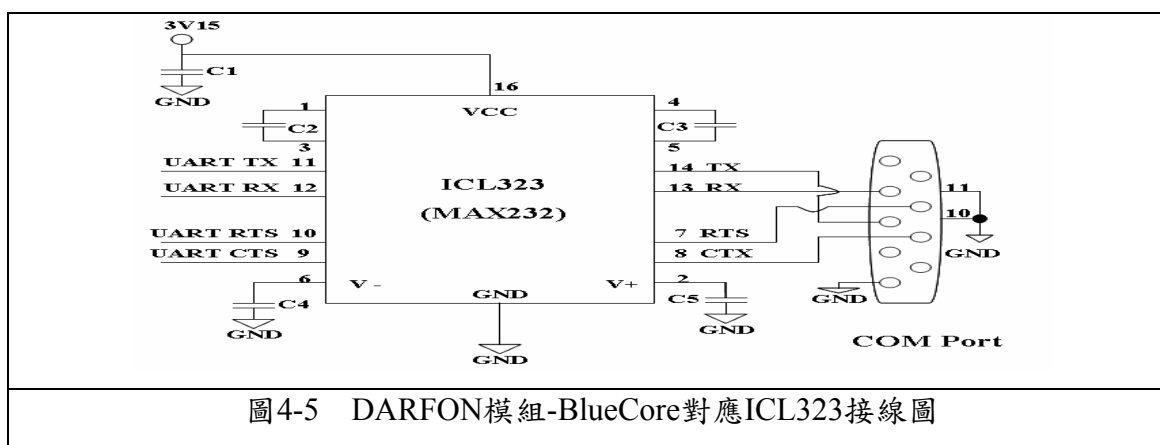
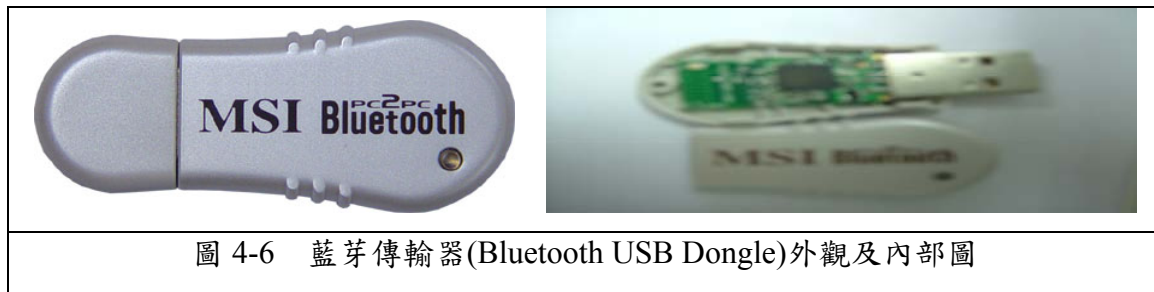


圖4-5 DARFON模組-BlueCore對應ICL323接線圖

4.1.3 MSI藍芽傳輸器(Bluetooth USB Dongle)

Bluetooth USB Dongle為支援藍芽技術(Bluetooth)的發射器，能自動偵測到其他藍芽裝置，進行連線，並有USB即插即用及完全嵌入藍芽功能。本實驗採用CSR所出的BC02藍芽晶片，MSI藍芽傳輸器，如圖4-6，使用BC02晶片。

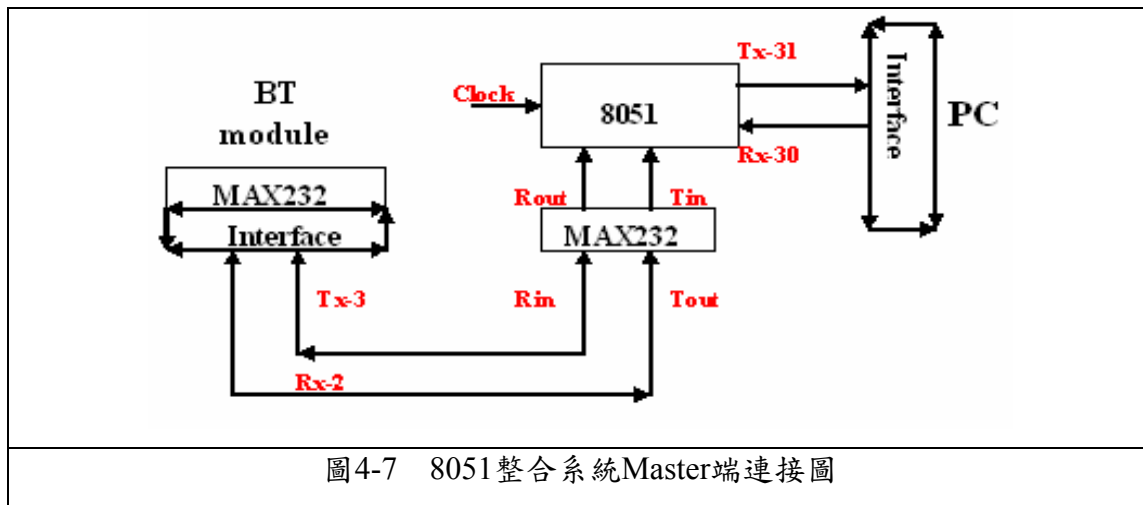


4.2 藍芽與8051整合為檔案傳輸系統

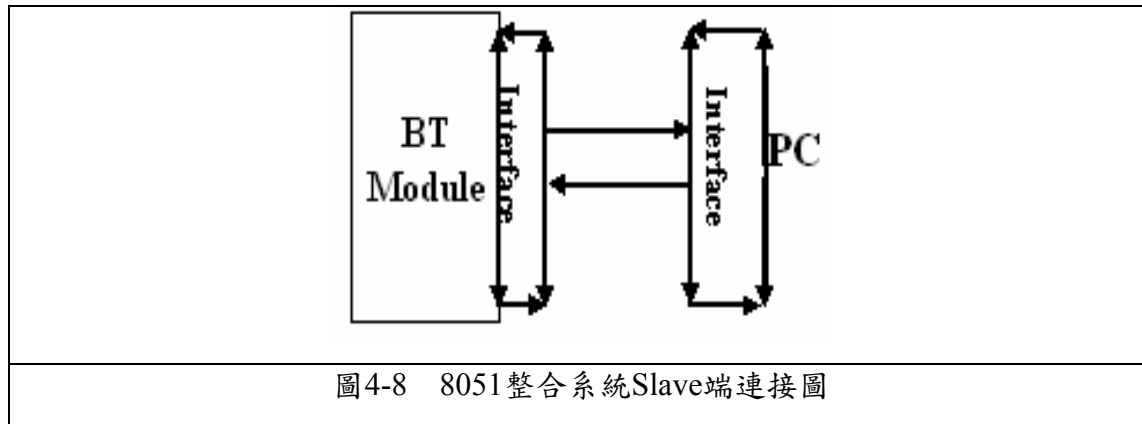
4.2.1 系統硬體：

在此系統中，硬體主要是分為兩端：

1. Master端：腳位主要以TX與RX二個Pin為主，PC為資料供給端，先連接至8051，再經MAX232(電位轉換晶片)與藍芽模組連結。TX與RX Pin腳連接如圖4-7所示。

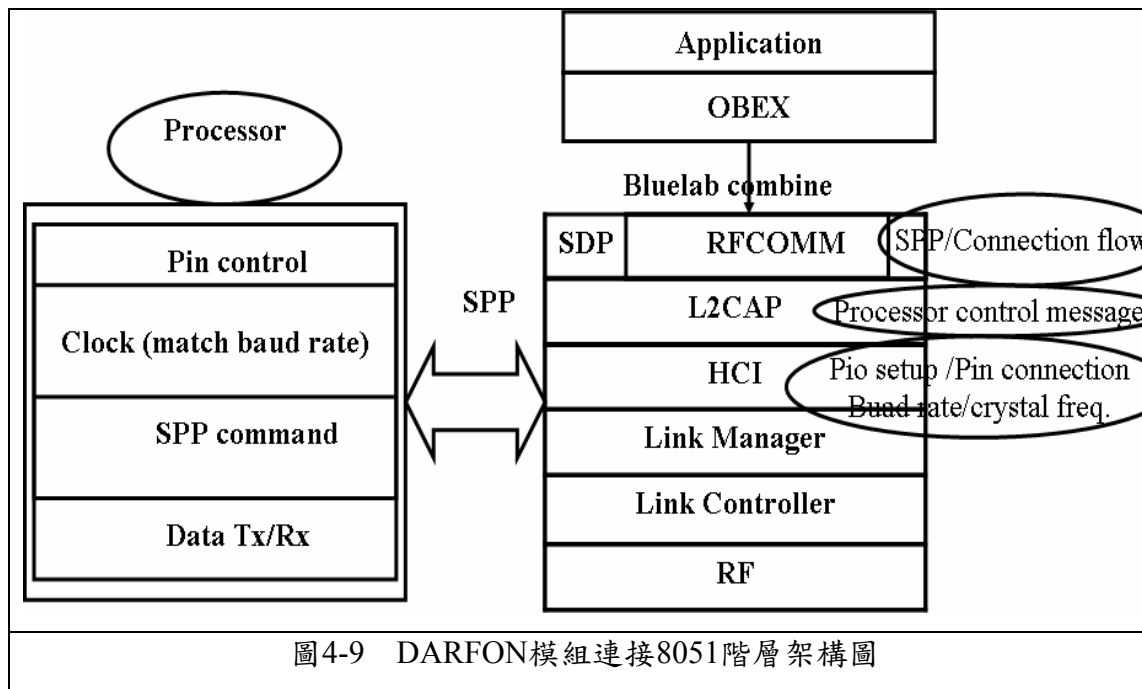


2. Slave 端： BT 模組作為資料接收端，透過介面，與 PC 連接。如圖 4-8。



4.2.2 系統軟體程式

程式層架構如圖 4-9 所示，分為兩部份：



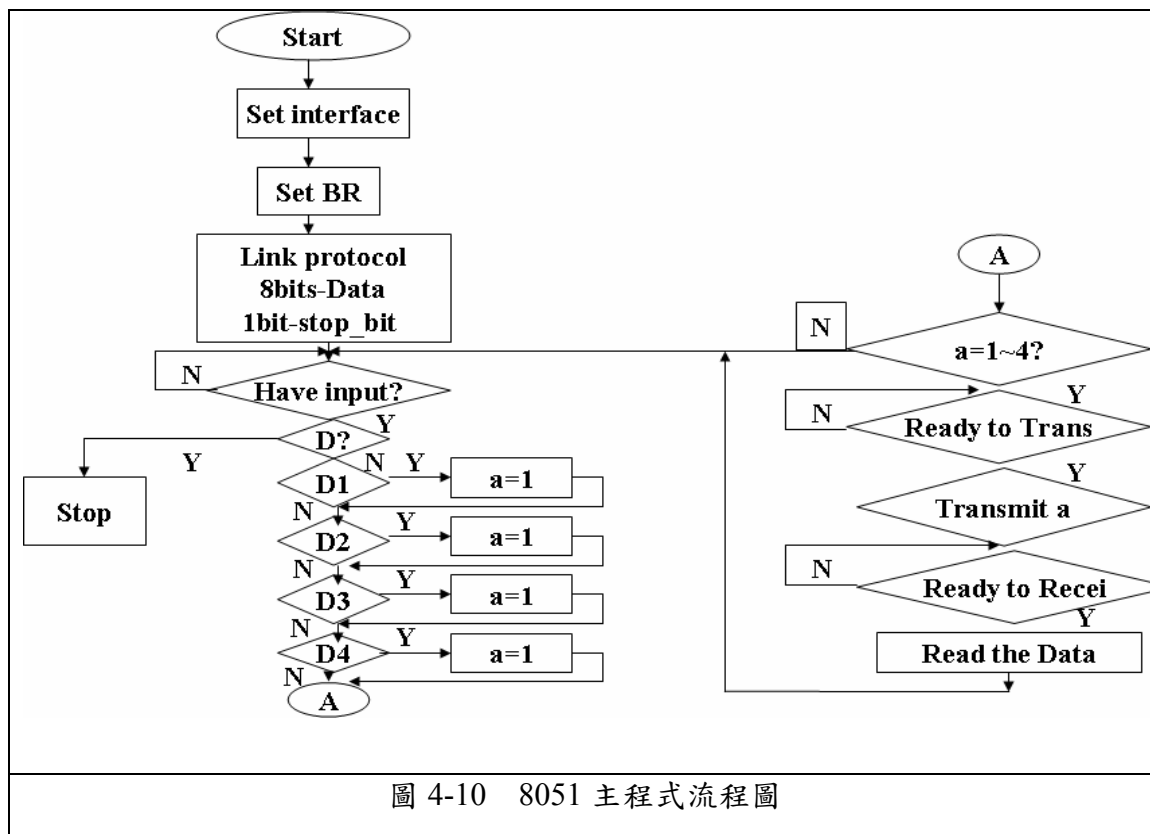
1. 8051微處理器： 主要程式設計是撰寫Pin Control(設定Pin腳功能)；Clock (配合藍芽的Baud Rate，預設值為9600 bps)；SPP Command(對應到藍芽端Serial Port Protocol，

透過介面傳送命令)；Data Tx / Rx (資料傳送及接收設定)。

- 藍芽端：更改藍芽的Firmware，使適用於8051微處理器，在Firmware中，主要是修改HCI層的PIO Setup；Pin Connection；Baud Rate及Crystal Frequency；在Protocol層，修改L2CAP層的Processor Control Message，並加上SPP(Serial Port Protocol)，來溝通8051及藍芽模組之間的介面，最上層Profile則是架構OBEX(Object Exchange) Profile，用來傳輸位元及資料，並可測試此系統的溝通性，最後，以Cygwin配合Bluelab.將程式編譯成Firmware，並燒入到藍芽模組中。

細說 8051 單晶片之控制程式：

8051 控制程式，總共分成三個部份，第一部份為程式主體，如圖 4-10。



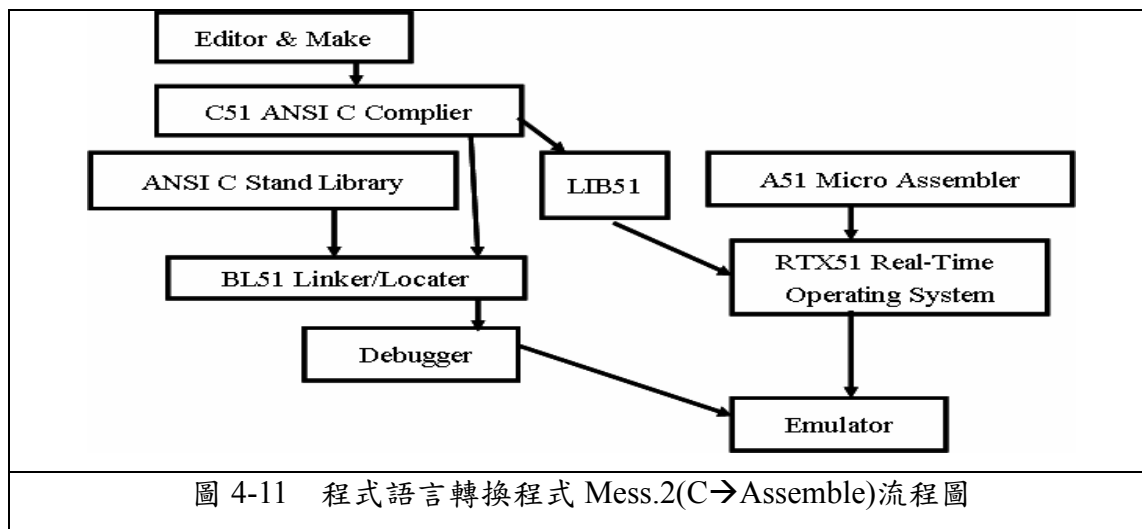
1. 程式主體：

(1)設定介面之相關通訊協定；(2)對應藍芽 Baud Rate 設定；(3)資料接收與傳送設定；(4)設定 Stop Bit 以確定資料接收完畢；(4)設定 Command Set，建立命令控制流程。

2. 資料判斷程式： 當程式接收傳來的資料時，對應指令控制，判斷資料形式。

3. 時間延遲程式

8051 程式以 C 撰寫，以 Command 控制 BT 模組傳送封包。用程式語言轉換程式 Mess.2(C→Assemble)，轉換流程如圖 4-11。將 C 轉為組語，再將程式燒錄到 8051。



Mess.2 將 C 轉為組合語言流程：

(1) 程式編輯與建構；(2)C 針對 51 組語組譯；(3)對照 C 的 Library 進行 51 程式連結及位址設定；(4)配合 51 組語格式，插入執行時間順序；(5) C 成功轉為組合語言。

藍芽模組端：

在韌體部份：更改 HCI 層的 PIO Setup 來配合藍芽模組的 Pin 腳；並設定封包型式、連線狀態、Baud Rate 及 Crystal Frequency。介面部份： 設定 SPP(Serial Port Protocol) 的

連接介面程式，來溝通 8051 及藍芽模組。

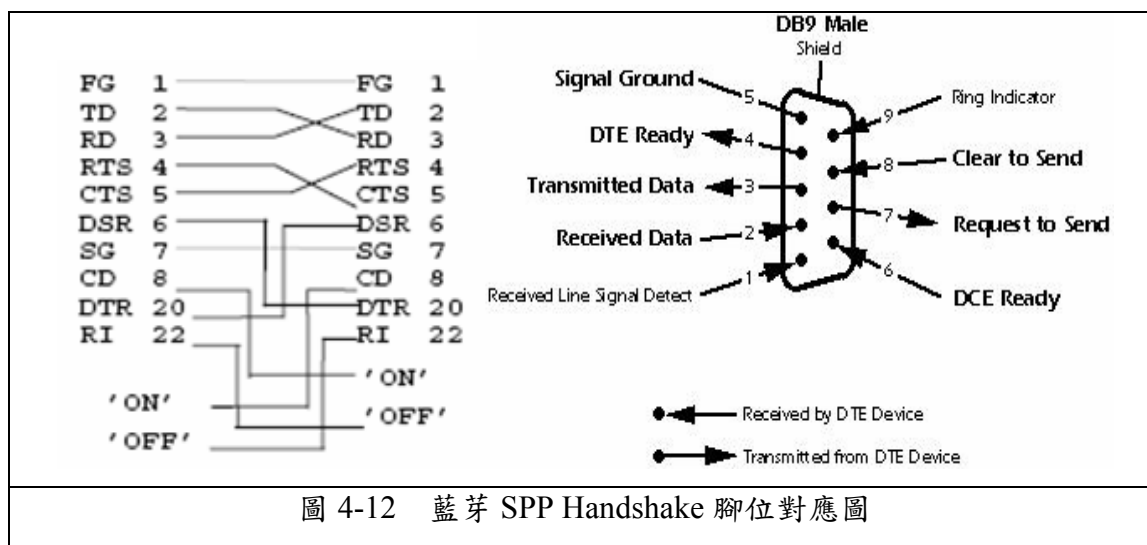
在 HCI 層:在函式 HCI_WRITE_SCAN_ENABLE_CMD、HCI_CREATE_CONNECTION_CMD 及 HCI_DISCONNECT_CMD 中，必須將 Big Ending 與 Little Ending 加入定義中，使封包傳送有限定的完整性；並設定 High Byte、Low Byte；Power Saving Mode 參數，最後設 Bluecore 程式為 16 bit RISC。

在 BlueLab：已經提供現有的 Library，如表 4-1。

Framework																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																									
-----------	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--

SPP 主程式分為兩端，一為 Master；另一端為 Slave，SPP Slave 端主要是接收藍芽晶片在介面之間的連結(Connection)與傳送資料，可用來對應跑 SPP Master 的應用程式，且可以從 PC 端對應跑適合的藍芽堆疊。SPP Master 端則是應用到由 VM 去連結 SPP Slave 程式。其中針對 Handshake 的部份必須更改程式，在呼叫 Handshake Library 時，會先初始化去告訴 DCE(RS232 介面類型)裝置，設 DSR；RI；和 DCD 都為輸入端；且 DTR 為輸出端。DTE 裝置則設定相反。表 4-2 為其設定 Spec.。在腳位及 Pin 腳對應介面的地方必需依照其對 RS232 介面撰寫對應的程式。圖 4-12 為其腳位及對應介面圖

表 4-2 BlueLab 初始定義 Specification	
TS 07.10 Signals	Corresponding RS-232 Control Signals
RTC	DSR, DTR
RTR	RTS, CTS
IC	RI
DV	DCD



在 SPP Master 及 Slave 程式中，對於 Handshake 部份除上述部份外，因用於兩端連接，對於 HandshakeInit(HandshakeRole role)、HandshakeControl(const CM_CONTROL_IND_T *ind)及 HandshakeSend(void)，三個函式需配合 BlueLab.的 Spec.針對 Role Change(master 及 slave 互換)；Common Mode Control Indicate(控制指定)及 Send(傳送) 改為適合本系統參數。

程式的動作流程：

與 Slave 端程式，只在 Role Change 及 Send / Receive 部份設定及呼叫的 Library 不同，動作流程是一樣的，以下介紹其程式流程。

Master / Slave 端：

1. 在 Serial Port profile,設 port 的定義。
2. 在 Static enum.，設定藍芽連線的狀態為 Start；Inquiring；Pairing；Connecting 及 Active。
3. 在開始流程，先將訊息(Message)送至連線管理端(Connection Manager)，再傳送要求(Requests)到連線管理端，進行程式動作。
4. 從連線管理端處理收到訊息。
5. 在 MessageType 中，設程式為：

```
config_use(&msg->use)
msg->addr      = far_addr;
msg->timeout    = D_SEC(60);
msg->role       = CmMaster;
```

將程式直接定義 Message 指向一個指令，再執行其指令的動作。

4.3 藍芽與MSP430整合為檔案傳輸系統

主要是採用在8051的藍芽系統的設定。但連接之介面為RS232，因此必須配合介面及處理器，不同來改程式參數的部分

1. MSP430端：

在MSP430中，控制程式中，只針對連接介面、控制腳位、輸入與輸出的設計及Clock與Baud Rate做設定。

2. 藍芽端：

在SPP程式設為RS232介面，並將程式配合MSP430加以更改，而原本在51中，會自動回ACK，確定檔案傳輸機制。但MSP430確無回ACK機制，其Spec.定義中，Library已經寫死，因此在藍芽端自行寫入AutoReceiACK()的程式，將系統整合。

以傳輸資料進行實驗，連線系統在傳資料時，由藍芽端自動回覆訊息，傳回所有連線流程及資料傳輸狀態，第五章所做的實驗數據為在系統使用AutoReceiACK時，所量得的數據。

4.4 藍芽與Windows整合系統

藍芽端採用MSI USB Dongle為傳輸器，更改CSR的USB HCI驅動程式，來驅動USB Dongle，自行撰寫HCI上層的程式，來建構系統。主機仍用PC Windows SP1作業系統，主要分為兩個系統架構。一為Headset系統，為用藍芽Dongle做為兩端的藍芽傳輸器，與PC所整合的系統。先架構Protocol層，及檔案傳輸的Profile(FTP)，先做檔案傳輸的實驗，再將Headset Profile架上系統，來完成聲音的傳送。二為應用藍芽的互動控制系統，以手機為傳輸器，再與PC藍芽端，組成互動控制的系統。主要是架構在Headset系統之上，在電腦端，在下層改Command Event的時槽(Time Slot)，而上層只架上FTP，配合一個檔案辨識系統，由藍芽手機發出的資料來進行遠端控制及傳輸。

4.4.1 系統一(Headset)

組成為兩個Dongle、兩台電腦主機。圖4-13為藍芽Dongle系統程式架構圖。

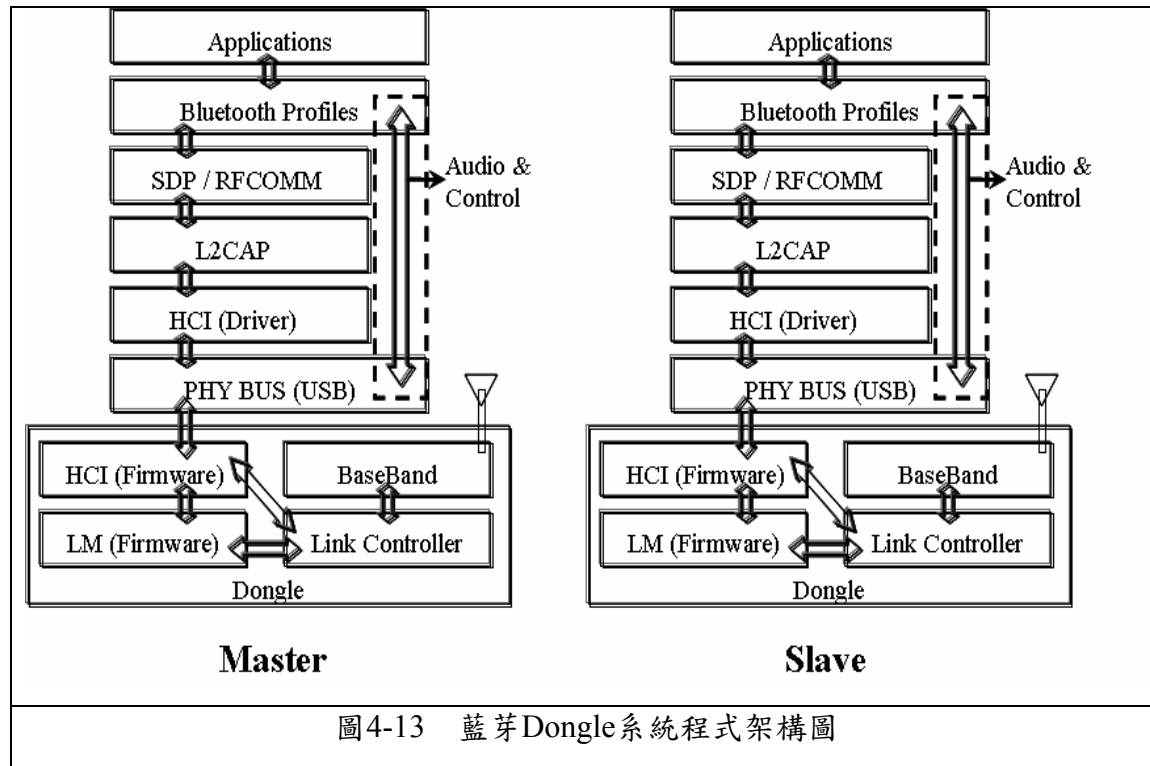


圖4-13 藍芽Dongle系統程式架構圖

一開始，藍芽Dongle之中本來就有內附的驅動程式，而MSI USB Dongle中，為CSR的藍芽晶片(BlueCore V.02)，因此捨棄內附的驅動程式，而改裝到CSR網站所下載CSR HCI.bco2的驅動程式。程式設計由HCI層，配合藍芽Spec.，建構一個藍芽耳機程式。

實驗步驟：

1. 環境設定：

先建立系統平台，需要的發展工具為Appropriate compiler (Microsoft Visual C++)；Microsoft Platform SDK；Microsoft Windows XP SP1；Windows XP QFE的藍芽支援程式，及Fix - Q323183(PC溝通藍芽HCI層的程式)。其中Microsoft Platform SDK可直接在<http://www.microsoft.com/msdownload/platformsdk/sdkupdate/> 下載

2. Bluetooth Dongle的設定：

Step1：改寫BLUETOOTH.inf檔。

Step2：增加一個新的硬體設定且搜尋已改寫的BLUETOOTH.inf檔。

```
[ControlFlags]
ExcludeFromSelect=*
```

```
[Manufacturer]
CSR=CSR, NT.5.1
ALPS=ALPS, NT.5.1
Brain Boxes=BrainBoxes, NT.5.1
Cambridge Silicon Radio Ltd.=Cambridge, NT.5.1
FIC=FIC, NT.5.1
GVC=GVC, NT.5.1
IBM=IBM, NT.5.1
Microsoft=Microsoft, NT.5.1
Silicon Wave=SiliconWave, NT.5.1
Sony=Sony, NT.5.1
TDK=TDK, NT.5.1
TOSHIBA=Toshiba, NT.5.1
```

```
;----- Device section - Start -----
[CSR.NT.5.1]
CSR Bluetooth USB Adapter=                                BthUsbDriverInstallXP, USB\VID_0A12&PID_0001
```

```
[ALPS.NT.5.1]
ALPS Integrated Bluetooth Device=                          BthUsbDriverInstallXP, USB\VID_044e&Pid_3005
Alps Bluetooth USB Adapter=                                BthUsbDriverInstallXP, USB\VID_044e&Pid_3006
```

BLUETOOTH.inf主要更改的部份為上面框所標之部份，如此設定才可用CSR HCI的驅動程式驅動Dongle。

3. Bluetooth Socket的程式撰寫，將Ws2bth.h及BluetoothAPIs.h include進到程式，使用Bluetooth sockets之前必須先將Ws2bth.h included到Winsock2.h之中。

4. 當使用Windows Sockets的應用程式，WSAStartup function必須call去初始化

Windows Sockets函式，並啟動藍芽。設定參數為：

```
WORD wVersionRequested = 0x202;
```

```
WSAStartup(wVersionRequested, &m_data)
```

其中m_data 可以接收Windows Sockets Implementation；而以AF_BTH來設定Address Families操作。

5. 藍芽是用Socket的函式，去產生內部及外部的設定連線，在產生一個Socket來使用藍芽，有以下幾個設定：

(1). Socket函式中的af 參數，必須一直設為AF_BTH。

(2). Socket函式中的type參數，必須一直設為SOCK_STREAM。

(3).在針對protocol參數設定，有BTHPROTO_RFCOMM函式配合藍芽protocol的設定。

6. 藍芽Server端操作部份(Bluetooth Server Operation)：

(1).先開啟Bluetooth Socket。

(2).設Port Number，設定為：

```
SOCKADDR_BTH SockAddrBthLocal = {0};
```

```
SockAddrBthLocal.port = BT_PORT_ANY;
```

其中SockAddrBthLocal是一個藍芽位址結構

(3).用Socket去訂一個Local Address，設bind()為：

```
bind(LocalSocket, (struct sockaddr *) &SockAddrBthLocal,  
sizeof(SOCKADDR_BTH));
```

(4). 設定所有進入的連結，設listen()為：

```
listen(LocalSocket, MAX_CONN_NUM);
```

(5).設定接受連結，在accept()，設為：

```
accept(LocalSocket, NULL, NULL); 後兩個參數設任意值。
```

(6). 設定從Client端接受資料的機制，為一個while loop，在recv()中，設為：

```
pszDataBufferIndex = &szDataBuffer[0];
```

```
recv(ClientSocket, pszDataBufferIndex, (TRANSFER_DATA_LENGTH -  
iTotalLengthReceived), 0);ClientSocket以做資料傳送為用途。
```

7. 藍芽Client端操作部份(Bluetooth Client Operation)：

(1).先開啟Bluetooth socket。

(2).設Port Number。

(3).設定建立連線到Server的藍芽裝置，在connect()，設為：

```
connect(LocalSocket, (struct sockaddr *) &SockAddrBthServer,  
sizeof(SOCKADDR_BTH))，SockAddrBthServer 為Server端的位址。
```

(4).設定資料輸出，在send()設定，設為：

```
send(LocalSocket, szData, TRANSFER_DATA_LENGTH, 0)
```

因為連線已經用Socket建立，在傳資料時，就不用再確認一次Server的位址。

(5).設定關閉Local Socket，在closesocket()設為：

```
closesocket(LocalSocket)
```

8. 藍芽Socket設定WSASetService部份，可直接呼叫來使用，有以下幾個Library：

BluetoothSetServiceState()	BluetoothEnumerateInstalledServices()
BluetoothEnableDiscovery()	BluetoothIsDiscoverable ()
BluetoothEnableIncomingConnections()	BluetoothIsConnectable()

經過以上步驟後，即可完成利用藍芽USB Dongle來傳送資料。

9. 下層已完成資料傳送，建立ACL，程式中加入SCO連線設定，再架上Headset Profile。

10. 此時系統將Dongle換成CASIRA (因為Dongle無聲音輸出裝置)為Headset端，由另一台電腦配合CSR的模擬軟體Mezoe模擬成手機端(Headset gateway)，來測試所寫的Headset程式。圖4-14為將用Mezoe模擬成的手機端傳送聲音到CASIRA端，訊息顯示已連線並開始傳送聲音，如此則可在CASIRA端用耳機聽到聲音。



圖4-14 聲音傳送實驗成功圖

4.4.2 系統二(藍芽互動控制系統)

手機端使用K700為藍芽端，PC與藍芽USB Dongle合成電腦端。

系統設計流程：

1. 手機端：

(1).利用電腦端對手機發出要求，讓手機進入控制模式。

[at+clck="cs",1(enter)]，目的是鎖住鍵盤。

[at+cmer=3,2,0,0,0]，目的是開啟傳送按鍵值給電腦。

(2).判斷手機上各個按鍵的對應碼，並將其分類。

(3).手機會利用藍芽傳輸，從虛擬的Serial Port傳資料，再藉由不同對應碼去執行不同之動作。

2. PC端：

(1).設計藍芽下層HCI的影像、資料和Command Code進行Decode及Packet Segment Rebuild的動作，更改Command Event的時槽(Time Slot)設定。

(2).定義傳送機制Protocol(RFCOMM、L2CAP、SDP、OBEX...)。

(3).撰寫藍芽程式Profile。

(4).定義按鍵成Control Code及宣告Command。

(5).撰寫滑鼠對應Control Code的控制程式。

(6).撰寫檔案辨識系統。

(5).將Data & image及Command傳入檔案辨識程式中。

(6).利用File Recognition Profile辨認Packet為Command或File。

(7).建立Active Interface：

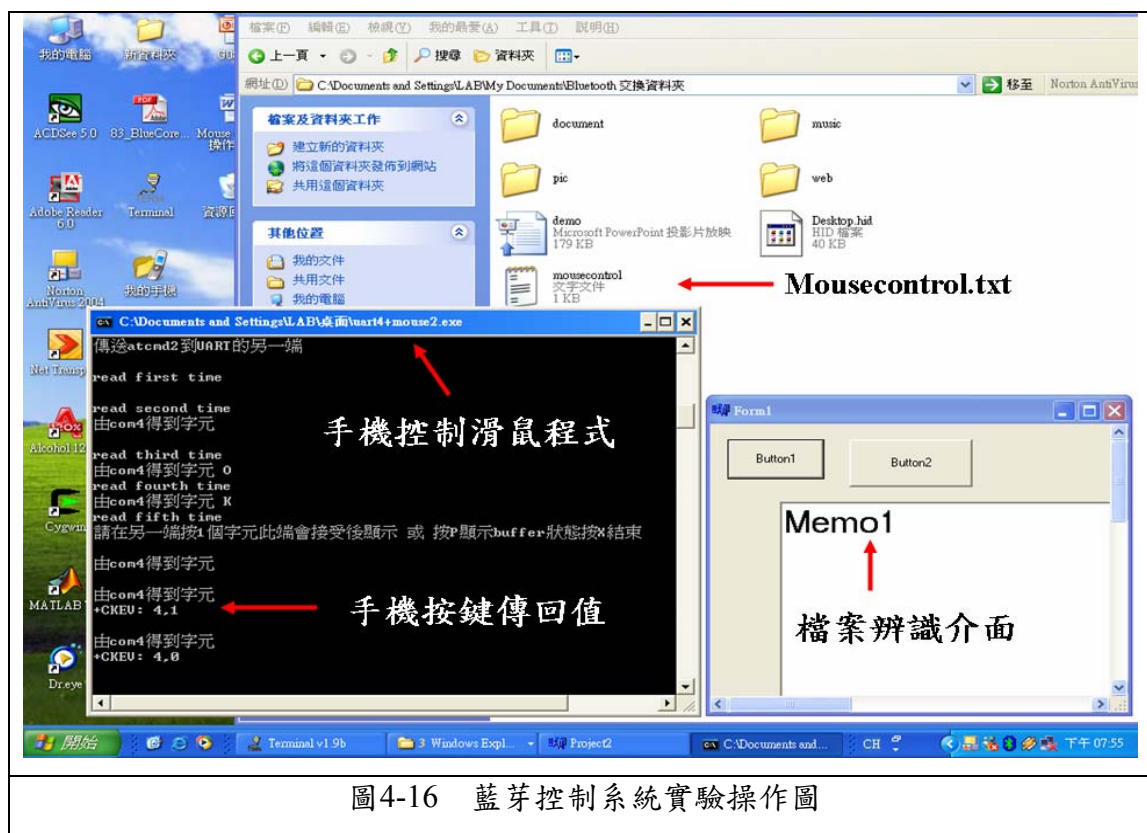
收到資料，若為Command，則執行應用程式Mouse Control Application，

若為Data或Image，則儲存或顯示，可支援的式:avi .jpeg .ppt. mpeg .txt ...。



圖4-15為藍芽控制系統流程圖，在藍芽手機與系統建立連線時，就可以進行檔案互傳機制，再從系統端下載服務項目(Service List)，如：Mouse_Control; Movie_Play; Music_Play.....等之後，再利用File Transfer Profile將mousecontrol.txt傳送至系統端，此時自動辨識系統便會辨識檔名並開啟滑鼠控制程式，滑鼠控制程式執行時，藍芽手機便可控制顯示系統中滑鼠游標之動作，進行瀏覽或是下載系統中的檔案。

圖4-16為操作藍芽控制系統時，電腦的螢幕的畫面，證實此系統已經完成。



第五章 實驗結果與討論

本實驗主要評估結合藍芽與處理器的整合性系統的優劣性，將藍芽與8051；MSP430，及PC Windows整合，透過介面建立整合系統。並利用藍芽系統規格，建立資料互傳機制，進行對封包種類、距離不同對傳輸速率及傳檔完整性影響的實驗。

實驗在無障礙環境(傳輸路徑中無障礙物)下進行，以減少其它因素影響實驗結果。

原本Spec.封包位元率如表5-1：

表 5-1 藍芽封包位元傳輸率表					
封包格式	Payload (Byte)	FEC	Symmetric (kbps)	Asymmetric (kbps)	
DM1	0-17	2/3	108.8	108.8	108.8
DH1	0-27	No	172.8	172.8	172.8
DM3	0-121	2/3	258.1	387.2	54.4
DH3	0-133	No	390.4	585.6	86.4
DM5	0-224	2/3	286.7	477.8	36.3

5.1 8051與藍芽整合的系統

8051與藍芽整合的系統：系統是以RS232為傳輸介面整合完成。進行檔案傳輸實驗，在Master電腦端將資料傳至8051，再由8051傳到藍芽端，發送到Slave端的藍芽模組，並接收檔案。

檔案傳輸結果測試：

選用DH3與DM5，兩種封包格式進行測試：

實驗結果如表5-2及表5-3：

表5-2 藍芽8051系統DH3封包位元無障礙傳輸率表(單位：byte)

File Size	1m	Bit rate	5m	Bit rate	8 m	Bit rate	10m
100,023	100,023	574.8kbps	100,023	568.9kbps	100,023	561.1kbps	No
207,342	207,342	557.2kbps	207,342	549.2kbps	206,940	534.9kbps	No
341,302	341,302	539.2kbps	341,302	530.5kbps	311,197	510.1kbps	No

表5-3 藍芽8051系統DM5封包位元無障礙傳輸率表(單位：byte)

File Size	1m	Bit rate	5m	Bit rate	8 m	Bit rate	10m
100,023	100,023	463.5kbps	100,023	459.1kbps	100,023	455.9kbps	No
207,342	207,342	459kbps	207,342	441.7kbps	207,342	409.8kbps	No
341,302	341,302	433.9kbps	341,302	417.1kbps	337,610	397.8kbps	No

由表5-2、表5-3的結果分析，如果照Spec.的定義，傳送封包速度對於DH3封包來說，傳送速率應可到達585.6 bps，而架構8051與藍芽整合的系統，以傳送一個100,023 byte大小的檔案，在封包遺失率部份，雖然DH3封包沒有加FEC，在1；5；8公尺距離的量測，檔案皆可完整收到，且平均速度可達到567.4 Kbps，可以達到資料遺失率為0%，但超過5公尺之後，資料遺失率變高，且傳送資料速度亦隨資料變大而變慢。而在DM5封包格式，因有加2/3 FEC的保護機制，雖然無法達到Spec.所定的傳送速度477.8 kbps，但在整合系統下，仍可達到463.5 kbps 的傳送速率，且其資料遺失率相對減少，但在距離加大之後，仍會因為空間干擾及封包的重複率提高，而使資料的遺失率及錯誤率變高。

5.2 MSP430與藍芽整合的系統

MSP430與藍芽整合的系統：系統是以UART為傳輸介面整合完成。進行實驗，在Master端將資料傳至MSP430，再傳到藍芽端，發送到Slave端的藍芽模組，並接收檔案。

選用DH3與DM5，兩種封包格式進行測試：

實驗結果如表5-4及表5-5：

表5-4 藍芽MSP430系統DH3封包位元無障礙傳輸率表(單位：byte)							
File Size	1m	Bit rate	5m	Bit rate	8 m	Bit rate	10m
100,023	100,023	532.1kbps	100,023	522.1kbps	No		No
207,342	207,342	528.9kbps	204,932	517.2kbps	No		No
341,302	341,302	519.2kbps	328,306	501.7kbps	No		No

表5-5 藍芽MSP430系統DM5封包位元無障礙傳輸率表(單位：byte)							
File Size	1m	Bit rate	5m	Bit rate	8 m	Bit rate	10m
100,023	100,023	401.9kbps	100,023	384.3kbps	No		No
207,342	207,342	392kbps	202,812	371.1kbps	No		No
341,302	341,302	383.5kbps	314,334	362.8kbps	No		No

由表5-4、表5-5的結果分析，因系統尚未穩定，容易斷線，以傳送檔案測試，在封包遺失率部份，DH3封包沒有加FEC，到資料遺失率較高，且5公尺之後，資料遺失率爆增，傳送資料速度隨資料變大而大幅變慢，在DM5封包格式，傳送速率較慢，但其資料遺失率相對減少，在距離加大之後，仍會因為介面及干擾，而使資料的遺失率及錯誤率變高。

5.3 PC Windows與藍芽整合的系統

PC Windows與藍芽整合的系統：系統是以USB為傳輸介面整合完成，處理器速度2.4G。實驗時，在Master端PC將資料經USB介面，傳至BT USB Dongle，再傳送到Slave端的BT USB Dongle，並接收檔案。

選用DH3與DM5，兩種封包格式進行測試：

實驗結果如表5-6及表5-7：

表5-6 藍芽PC Windows系統DH3封包位元無障礙傳輸率表(單位：byte)							
File Size	1m	Bit rate	5m	Bit rate	8 m	Bit rate	10m
100,023	100,023	581.5kbps	100,023	578.1kbps	100,023	569.4kbps	No
207,342	207,342	580.2kbps	207,342	570.9kbps	207,342	562.3kbps	No
341,302	341,302	578.9kbps	341,302	563.1kbps	341,171	551.6kbps	No

表5-7 藍芽PC Windows系統DM5封包位元無障礙傳輸率表(單位：byte)							
File Size	1m	Bit rate	5m	Bit rate	8 m	Bit rate	10m
100,023	100,023	470.5kbps	100,023	465.1kbps	100,023	455.9kbps	No
207,342	207,342	468kbps	207,342	461.7kbps	207,342	450.8kbps	No
341,302	341,302	466.9kbps	341,302	457.1kbps	341,302	443.8kbps	No

由表5-6、表5-7的結果分析，在封包遺失率部份，DH3封包在1；5；8公尺距離的量測，檔案皆可完整收到，且平均速度極快，可以達到資料遺失率為0%，但超過8公尺之後，341,302 byte 檔案才出現些許資料遺失，傳送資料速度隨距離增大而變慢；DM5封包格式，有加2/3 FEC的保護機制，雖然無法達到Spec.所定的傳送速度477.8 kbps，仍可達到470 kbps的傳送速率，資料遺失率在實驗量測為0%，但速度仍會受檔案大小影響。

5.4 實驗結果討論

以下針對本實驗所用的處理器、實驗方法、使用介面、使用技術及三個系統的實驗結果，做統整的分析，如表 5-8 與表 5-9 所示。

表 5-8 系統組成及使用技術				
處理器種類	連接介面	介面特性	使用技術	
PC	USB	通用介面	<ul style="list-style-type: none"> ➤ 將Linux BT HCI Code轉成Window系統 ➤ 改USB Dongle Firmware設定 ➤ 撰寫FTP Profile ➤ 做檔案傳輸的實驗 	
8051	RS232	有Error Correction的機制	<ul style="list-style-type: none"> ➤ HCI層程式及Protocol與FTP Profile ➤ 配合RS232介面及Darfon藍芽模組撰寫程式 ➤ 設定回ACK的機制 ➤ 用Cygwin配合BlueLab，組譯成Firmware並燒錄 	
MSP 430	UART	無Error Correction的機制	<ul style="list-style-type: none"> ➤ 應用8051的程式架構 ➤ SPP程式中，設為UART介面 (MSP430的ACK機制已被內設的Library包住，故在傳送資料時(回ACK不穩)) ➤ 撰寫AutoReceiACK的機制 ➤ 資料傳輸實驗 	

表 5-9 系統實驗結果及結果分析

整合系統	檔案平均傳輸率	資料封包遺失率		整合系統評估
PC+USB	DH3: 579.1 kbps DM5: 461.3 kbps ~~~~~SPEC.~~~~~ DH3: 585.6 kbps DM5: 477.8 kbps	5 Meter	檔案大小207,342 DH3: 0 % 檔案大小341,302 DM5: 0 %	PC+USB 速度最快 遺失率最低
		8 Meter	檔案大小207,342 DH3: 0.038382% 檔案大小341,302 DM5: 0 %	
8051+RS232	DH3: 549.5 kbps DM5: 439.3 kbps ~~~~~SPEC.~~~~~ DH3: 585.6 kbps DM5: 477.8 kbps	5 Meter	檔案大小207,342 DH3: 0% 檔案大小341,302 DM5: 0 %	8051+RS232 速度次快 遺失率次低 (有介面保護)
		8 Meter	檔案大小207,342 DH3: 0.194.% 檔案大小341,302 DM5: 0.88206 %	
MSP430+UART	DH3: 513.7kbps DM5: 372.3kbps ~~~~~SPEC.~~~~~ DH3: 585.6 kbps DM5: 477.8 kbps	5 Meter	檔案大小207,342 DH3: 1.2% 檔案大小341,302 DH3: 3.74% 檔案大小207,342 DM5: 0 % 檔案大小341,302 DM5: 2.08 %	MSP430+UART 速度最慢 遺失率最高 (無介面保護)

藉由實驗結果，再對照藍芽與處理器整合的系統比較表，可得出以下結論：

1. PC+USB 系統：速度最快，遺失率最低，系統最穩定，極少受距離影響。
2. 8051+RS232 系統：速度次快，遺失率次低。
3. MSP430+UART 系統：速度最慢，遺失率最高，系統以 AutoReceiACK 寫成，非直接由處理器回 ACK，易受檔案大小及距離的影響。

針對傳輸率與 SPEC. 對應的速率，在 8051 與 MSP430 的傳輸率有明顯的差距，應是藍芽傳輸模組是自行至達方電子進行設計焊接的，所以電路與 RF 端的效能並非可以達到 SPEC. 的速度。在此可以在使用業界已完全生產製造並將模組最佳化的藍芽設備，在進行更正確的評估，應可以使傳輸率提高到 SPEC. 所定義的速度。而在應用點上，亦可以使用其它的微處理器進行整合系統，將可以使實驗與應用更加的完備。

第 六 章 結 論 與 未 來 發 展

6.1 結 論

本論文主要目的為結合藍芽的短距離無線技術與處理器，以期完成一個整合性系統，並針對藍芽與8051；MSP430，及PC Windows完成系統化，主要設計是在韌體(Firmware)與軟體(Software)的撰寫，及透過介面建立整合系統。並利用藍芽系統規格，建立資料互傳機制，驗證系統的完整性其可用性。

此三個處理器系統，在本研究中被設計，完成架構，測試完成，成功建構出藍芽與處理器之整合系統，且由實驗結果可以評估出系統與藍芽模組的傳輸效能，有助於之後在架設藍芽整合系統的參考。最後設計完成藍芽遠端控制系統，主要是連線後，將手機端的資料傳到PC後，利用手機控制PC，並藉由檔案辨識程式分辨命令與資料，來執行應用軟體，做出一個藍芽互動控制的應用系統，可直接應用於導覽及會議系統。

如此建立一個整合處理器與藍芽的技術，就可成功將藍芽功能嵌入設備中，而使現有設備增加多工性，對於藍芽短距離無線通訊的普及與應用發展都具有極大的貢獻。

6.1 未 來 發 展

目前由於藍芽功率消耗低，對人體危害小，且藍芽積體電路應用簡單，成本低廉，為現行短距離無線系統最適合的系統，而本論文將藍芽與處理器整合系統實現，在未來的發展上，可著重在以下二個部份：

在系統方面：可將藍芽系統規格V1.2取代現行的V1.1，提高傳輸率，建立高影音傳輸之藍芽系統，亦可配合其他廣域的無線系統，如802.11b/g...等，將系統擴充成全通性無線系統，並可將已完成的整合系統架構，配合藍芽應用功能程式移植至嵌入式系統，即可生產藍芽應用產品使用。

在應用方面：可以將藍芽應用在互動式顯示系統、無線監控系統、藍芽認證金融卡、手機認證及無線3D投影...等，使辦公室、家庭及其他場所中，可建立固定與行動設備通信環境中的個人網路，實現數位家庭(E-Home)的目標，使近距離內的各種資訊設備能夠

實現無線資源分享。並使藍芽系統更加完善與普及化，與生活更密切結合，使藍芽科技能達到多方面應用與發展的目的。

參考文獻

- [1] Arumugam, A.K., Armour, S.M.D., Lee, B.S., Tariq, M.F., Nix and A.R., “Consumer electronics application and coverage constraints using Bluetooth and proposed Bluetooth evolution technologies”, Consumer Electronics ICCE International Conference, pp. 14–15, June 2001.
- [2] Bhagwat, P., “Bluetooth: technology for short-range wireless apps”, Internet Computing, IEEE Vol. 5, Issue 3, pp. 96–103, May-June 2001.
- [3] Bray, Jennifer, Sturman and Charles F., “Bluetooth: connect without cables”, Prentice Hall PTR, 2001.
- [4] Brent A. Miller, Chatschik Bisdikian, “Bluetooth Revealed: The Insider's Guide to an Open Specification for Global Wire”, Prentice Hall, 2000.
- [5] C. Bala Kumar, Tim Thompson, Paul Kline, “Bluetooth Application Programming with the Java APIs”, Morgan Kaufmann, 2003.
- [6] Chao Lin Wu, Li hen Fu, Feng Li Lian, “WLAN location determination in e-home via support vector classification”, IEEE International Conference, Vol. 2, pp. 1026–1031, 2004.
- [7] Chaudhry, M.A.R., Sheikh and M.I., “Protocols stack & connection establishment in Bluetooth radio”, Students Conference, ISCON 02 Proceedings, IEEE vol. 1, pp. 48 - 55, Aug 2002.
- [8] Dee Bakker, Diane McMichael Gilster and Ron Gilster, “Bluetooth End to End”, Hungry Minds, 2002.
- [9] Gil Held, “Data Over Wireless Networks: Bluetooth, WAP, and Wireless LANs”, McGraw-Hill, 2000.
- [10] Jennifer Bray, Brian Senese, Gordon McNutt and Bill Munday , “Bluetooth Application Developer's Guide”, Syngress Media, 2001.
- [11] JiaRen Chang Chien, Cheng Chi Ta, “The information home appliance control system - a Bluetooth universal type remote controller”, International Conference, IEEE Vol.1, pp. 399–400, March 2004.

- [12] Michael Miller, “Discovering Bluetooth”, Sybex, 2001.
- [13] Nordbotten, N.A. Skeie, T. Aakvaag and N.D., “Methods for service discovery in Bluetooth scatternets”, Computer Communications, Jul 2004.
- [14] Robert, Morrow, “Bluetooth: Operation and Use”, McGraw-Hill, 2002.
- [15] Specification of the Bluetooth System Core Version 1.1, February 22, 2001。
- [16] Specification of the Bluetooth System Profile Version 1.1, February 22, 2001.
- [17] Wu, T. Y. Huang, C. Y. Chao and H. C., “A survey of Mobile IP in cellular and Mobile Ad-Hoc Network environments”, Ad Hoc Networks, May 2005.
- [18] Youquan Zheng, Zhenming Feng, “Simplifications of the Bluetooth radio devices”, IEEE 4th International Workshop, pp. 107–115, 2002.
- [19] 李純、周開波、童兆豐, “無線通訊技術－探索藍芽”, 五南, 2000。
- [20] 杉浦彰彥原著、蘇慶輝、蘇慶宏編著, “藍芽技術解說”, 全華, 2001。
- [21] 周英展, “藍芽車用免持系統實作”, 國立中正大學通訊工程研究所碩士論文, 民國九十二年。
- [22] 林俊宏、楊順興、李忠來、黃億祥、吳建偉, “藍芽－無線連結技術 (Bluetooth: Connect Without Cables, 2/e)”, 全華, 2003。
- [23] 禹帆, “深入探討無線藍芽技術(最新版)”, 文魁, 2004。
- [24] 孫世榮, “以藍芽技術實現無線 MP3 音訊串流”, 國立中正大學電機工程研究所碩士論文, 民國九十一年。
- [25] 張焜傑, “設計藍芽無線傳輸系統應用於自由活動老鼠之生理訊號記錄”, 國立成功大學醫學工程研究所碩士論文, 民國九十二年。
- [26] 陳新添, “藍芽系統安全機制與認證計畫功能”, 國立中正大學電機工程研究所碩士論文, 民國九十一年。
- [27] 童志偉, “藍芽無線電話與對講機系統實作”, 國立中正大學通訊工程研究所碩士論文, 民國九十一年。
- [28] 劉書生、趙海, “藍芽技術應用”, 五南, 2002。
- [29] 劉惠綸, “行動電話年輕使用族群之聽覺閾值、非特異性自覺症狀研究：兩年追蹤”, 高雄醫學大學職業安全衛生研究所碩士論文, 民國九十二年。

- [30] 鄭秀敏，“以藍芽傳輸為基礎之檔案傳輸功能實現”，國立中正大學電機工程研究所碩士論文，民國九十一年。
- [31] 廖旂湧，“藍芽USB Dongle之應用軟體系統及其圖式使用者介面程式之實現”，國立中正大學電機工程研究所碩士論文，民國九十二年。
- [32] 謝家瑞，“藍芽視訊監控系統實作”，國立中正大學通訊工程研究所碩士論文，民國九十二年。
- [33] 蕭凱仁，“單晶片控制之藍芽通訊板作血壓信號傳輸應用”，國立成功大學電機工程研究所碩士論文，民國九十二年。