

私立東海大學資訊工程與科學系研究所

碩士論文

指導教授：朱正忠 教授

以應用 **UML** 開發不同嵌入式系統之軟體
**Using UML to Implement Application on
Different Embedded System**

The image shows a large, faint circular seal of the National Central University Library. The seal contains the text "NATIONAL CENTRAL LIBRARY" around the top and "ROC" at the bottom. In the center, there are Chinese characters "中央圖書館" (National Central University Library).

研究生：董信佑

中華民國九十九年三月六日

致謝

本篇論文能順利完成，首先要感謝指導老師 朱正忠 與 張志宏 教授，這兩年期間在學業上的悉心指導，使我在學術研究的領域中有所成長，在此謹向恩師 朱正忠 教授與 張志宏 教授致上最高的敬意與謝意，此外還要特別感謝我任職的公司英業達股份有限公司在硬體資源上與技術上的協助，同時也感謝口試委員們撥冗指正論文中的錯誤並提供許多建議，使本篇論文能更加完整。

目錄

目錄.....	I
圖目錄.....	IV
附錄圖目錄.....	IV
表目錄.....	V
摘要.....	VI
摘要.....	VI
Abstract.....	VII
第 1 章 導論.....	1
1.1 前言.....	1
1.2 研究動機.....	2
1.3 研究目的.....	3
1.4 章節安排.....	4
第 2 章 相關研究及技術.....	5
第 3 章 方案應用與系統規劃流程.....	8
3.1 方案應用.....	8
3.2 系統規劃流程.....	10
3.3 傳統設計的系統流程.....	16
第 4 章 透過 UML 與傳統程式設計流程之間的差異.....	17
4.1 UML 導入嵌入式系統的設計方法.....	17
4.2 UML 與傳統程式設計間差異的數據比較.....	21
第 5 章 實作案例.....	28
第 6 章 結論.....	32
參考文獻.....	33
附錄 藍芽實作裝置操作步驟.....	35

圖目錄

圖 1.	藍芽通訊應用於商務案例的架構	9
圖 2.	PC 端程式流程	11
圖 3.	嵌入式系統上的程式流程	13
圖 4.	ATML 89C51 韌體佈局圖	14
圖 5.	MICROCHIP PIC16F877 韌體佈局圖	15
圖 6.	系統UC圖	18
圖 7.	循序圖	19
圖 8.	系統類別圖	21
圖 9.	傳統設計韌體流程	23
圖 10.	遠端藍芽通訊系統之實作模型	28

附錄圖目錄

圖 1.	附錄圖(一)	35
圖 2.	附錄圖(二)	36
圖 3.	附錄圖(三)	37
圖 4.	附錄圖(四)	38
圖 5.	附錄圖(五)	39
圖 6.	附錄圖(六)	40
圖 7.	附錄圖(七)	41
圖 8.	附錄圖(八)	42
圖 9.	附錄圖(九)	43
圖 10.	附錄圖(十)	44
圖 11.	附錄圖(十一)	45
圖 12.	附錄圖(十二)	46

表目錄

表 1.	不同的MCU之暫存器定義	25
表 2.	需依賴不同 Mcu 指令之Function 名稱.....	26
表 3.	透過UML Code Generator設計與傳統設計Coding行數差異.....	27
表 4.	UML設計與傳統設計之開發時間差異	27

摘要

由於目前嵌入式系統之軟體開發需求的普及我們設計了一個物件導向嵌入式軟體開發框架。我們採用目前最為廣泛使用的 UML 塑模語言來建立系統規格模型。透過塑模語言來進行物件導向的設計與開發來得以讓程式設計師更加瞭解整個系統架構的細節，並提高未來系統設計的正确性，更可以減少除錯的時間，進而間接提高開發的速度與效率。借用這個框架我們實作一個應用實例，即「遠端藍芽通訊系統」。透過 UML 的協助，得以在極短的時程內針對兩種以上不同的 Micro Control Unit (MCU) 開發。在實作過程中可以明確地看出使用 UML 來開發嵌入式軟體的優點。同時，也證明了它的正确性及迅速性。

關鍵詞：嵌入式系統、嵌入式系統、塑模語言、設計樣板。

Abstract

Due to the embedded system is more and more popular in this digital era, thus, we have designed a framework of Object-Oriented (OO) program on embedded system.

In this case, we adopt the best known UML modeling language to construct the embedded system model. Programming developers can use the UML modeling language to understand more detailed information of the whole system structure, and improve the correctness of designing a program in the future. Particularly, it not only reduced much debug time, but also improved developing time and efficiency.

Here is an example of applying this framework. We designed a “remote Bluetooth communication system” via UML modeling language, and it proved that we can develop tow kinds of Micro Control Unit (MCU) in a very short time. By conducting this experiment, we can see the advantages of UML modeling language obviously. Meanwhile, its correctness and rapidity is proved.

Keywords: UML, Embedded Systems, Framework, Program Development.

第1章 導論

1.1 前言

嵌入式系統在我們的生活中日漸普及化，也因為它的普及化以及激烈的競爭，在商場上嵌入式產品開發的速度與品質成為產品獲利率高低的重點。然而如何在開發速度上的提升又兼顧品質的穩定發展，成為了本文研究的主要重點。傳統設計嵌入式系統必須仰賴工程師的經驗，因此如果是面臨一個新的平台誕生，往往必須重新學習新的軟體。再則只靠單一資深工程師的經驗與想法，而無制式的設計法方與規則情況下，萬一該設計人員另謀高就後，更甚者恐怕影響一家小公司的生存之計。而程式完成後的除錯更是困難，傳統使用測試為基礎的除錯技術非常花費時間卻不能保證整個程式通過測試流程以後就一定不會出錯。

1.2 研究動機

由於以上的限制造成嵌入式系統設計時的許多困難。為了解決這些問題，我們提出了一個物件導向嵌入式軟體開發框架。在其中整合了 UML[1] 塑模之技術，物件導向技術，軟體排程技術，以及自動化程式碼生成技術。目前 UML[1] 是業界最廣為接受與使用的塑模語言，借由 UML 來當一種工具成為設計者輸入的一種介面，得以有效並大幅的減少程式設計人員開發與學習新平台程式語言的時間，也由於 UML 的特性得以更完整的表達出系統所需的行為與需求。

1.3 研究目的

同時因為物件導向的技術得以使軟體元件的重複使用性大幅提高，這也得讓程式設計人員不需再花很多的時間去進行新平台的適應與重新設計，而直接取用現存已開發過的軟體元件加以整合與微幅的修正後，即可應用於新平台上的程式發展，有效並大幅減低了開發時間與錯誤的產生。軟體排程之技術得以有效確保產生的程式能夠符合使用者指定的時間限制，以及突發需求的修改與增訂，對於嵌入式系統的設計者來說具有很大的助益。軟體驗證的技術可以讓設計者經由正規驗證的方法，完整地驗證系統行為的正確性，避免系統執行可能發生的錯誤。自動程式碼生成技術讓使用者可以減少自行撰寫程式碼的需要，在突發需求的增訂上也不致於導致使用者思維混亂，可以減少人為的程式錯誤及撰寫程式碼的時間。

1.4 章節安排

論文的章節安排分述如下：

第二章將介紹相關研究及技術。

第三章介紹該方案應用與系統規劃流程。

第四章中比較透過 UML 與傳統程式設計之間的差異。

第五章介紹該實作系統的案例「遠端藍芽通訊系統」。

第六章結論。

第2章 相關研究及技術

就現今而言，國內外有相當多關於 UML 實現嵌入式系統的研究，關於該方面的例子如美國軍方 MoBIES (Model-Based Integration of Embedded System) 計劃中的 AIRES[2] 與 CMU 提出的 Time Weaver [3] 這兩項工具都提出使用物件導向的觀念與方法來協助開發嵌入式系統的的設計。

在 MoBIES 這個計劃中包含了許多個子計劃都是應用嵌入式系統軟體自動產生技術而成，其中 AIRES 著重於嵌入式軟體的建置模型，它把嵌入式軟體區分為數個步驟，再分別給與這些步驟取個不同的描述模型來建造系統。包括行為模型，軟體結構模型，執行模型及生成碼模型。利用這些模型可以表示系統的行為及元件的關係，並且提供效能分析及自動產生程式碼的依據。他們提出框架中的軟體元件應該以 meta-model 的方式存在，提供相關於軟體行為，結構，環境及效能的資料。Time Weaver 著重的研究在於提供設計時使用的軟體模型框架，建立驗證模型及可執行的程式碼。它針對嵌入式軟體中可能出現的一些非功能性規格（例如時間點，服務品質及可靠度）提供分析的機制，確保這些規格能夠被滿足。它透過元件模型的結合來產生程式碼。並且使用 coupler 來表示元件之間的關係，不同的 coupler

可能表示不同的關係，包括了時間，資料流等。最後使用語意維度 (semantic dimensions) 來提供針對相同物件的不同觀點。

這些工具都提出了使用元件模型產生嵌入式系統軟體的方法，即時特性及正確性都是嵌入式系統很重要的元素，不能確保這些特性可能造成系統的不穩定，甚至造成很大的損失。因此我們有必要確保這方面的正確性。

我們的框架中使用 UML [1] 作為系統的輸入模型，由於 UML 是工業界最廣泛使用的塑模語言之一，因此大部分的使用者可以很快的學會使用這套工具。另外我們整合了系統排程及正規驗證的方法來確保系統的即時特性及功能正確性。透過系統排程，產生的程式可以滿足使用者限定的時間限制，保證系統執行時能夠符合時間限制完成。正規驗證可以避免傳統使用測試方法可能有的涵蓋率問題，它可以保證系統完整滿足規格的屬性。

以模型為基礎的使用方式已經在前文討論過了，其中指出了一些優勢，諸如：成本的下降、掌控晶片的複雜性、提高生產力等等。目前 UML / MDA 已經在過去幾年中的協同設計方法中證明其成功性，並且被接受[4][5][6][7]。該擴展機制導入 UML 的版本為 1.3，模擬其使用在嵌入式系統建構，因為這種系統所需這樣的特定模式。

在[4]中，作者可以自定義 UML 模型 SystemC 的配置檔案元素。SystemC 的架構會從 UML 模型中產生程式碼。此外，在[8]，藉由指定 UML 狀態與階段 SystemC 配置文件的定義和行為，其中一個 TLM（事務級建模）的 SystemC[9]程式碼會被產生出來。

在[5]中，一個延伸程式可以用來定一個新的配置文件，TUT 文件置入嵌入式系統設計中。它定義了一套一般常用的模型應用任務和平台。此平台使用的資源庫，可以拿來分析性能。設計的流程中只允許軟體程式碼用 C 語言來產生。它有一個架構的探索工具，重新詮釋了整個 UML 模型。有一個完整的例子在[10]。

在[6]，作者可以使用以自定義的 UP（統一過程）為基礎過程中使用 SystemC 配置嵌入式系統建模來產生 SystemC 程式碼。

在[7]，作者使用 UML 和 VHDL 來產生程式碼。他們使用相同的模型產生硬體與軟體兩個部分。分割需要手動去完成，是與模型不同的。

第3章 方案應用與系統規劃流程

3.1 方案應用

該系統架構觀念圖如圖 1，此概念乃來自於戶外電子看板，與點餐系統 client 端的功能應用，就電子看板而言，在早期廣告商欲改變廣告內容時，往往得爬上高樓看板處進行資料的傳輸與更新，而隨著科技的日新月異，有了藍芽通訊的技術出現之後，這樣的廣告看板公司員工再也不必冒著生命的風險爬上高樓進行資料的更換動作，只需透過即定的藍芽通訊協定與看板端進行資料的轉換即可完成。

點餐系統亦然，過去的點餐系統是客戶填單，由服務小姐或男生來到您的餐桌前與您確認您所要的餐點與資訊，一樣的在無線科技如此發達的今日這些問題可以省略，客戶可以由桌上擺放的無線網路點餐裝置直接進行菜單上餐點的點選，點選的方式是由使用者輸入菜單項目篇號，或借由條碼掃描機制掃描菜單項目代號，客戶所輸入的資訊經由藍芽無線的傳輸將資料送達給 sever 端，進行後端的資料與餐點的準備工作。

當然這類的應用在目前的生活不計其數，而在這裡我只將這樣的應用透過 UML 語言的設計概念，讓這樣的設備應用得以快速的在商場上開發出得利的商品，因此該系統的源自於無線藍芽設備的需求架構而發展得來。

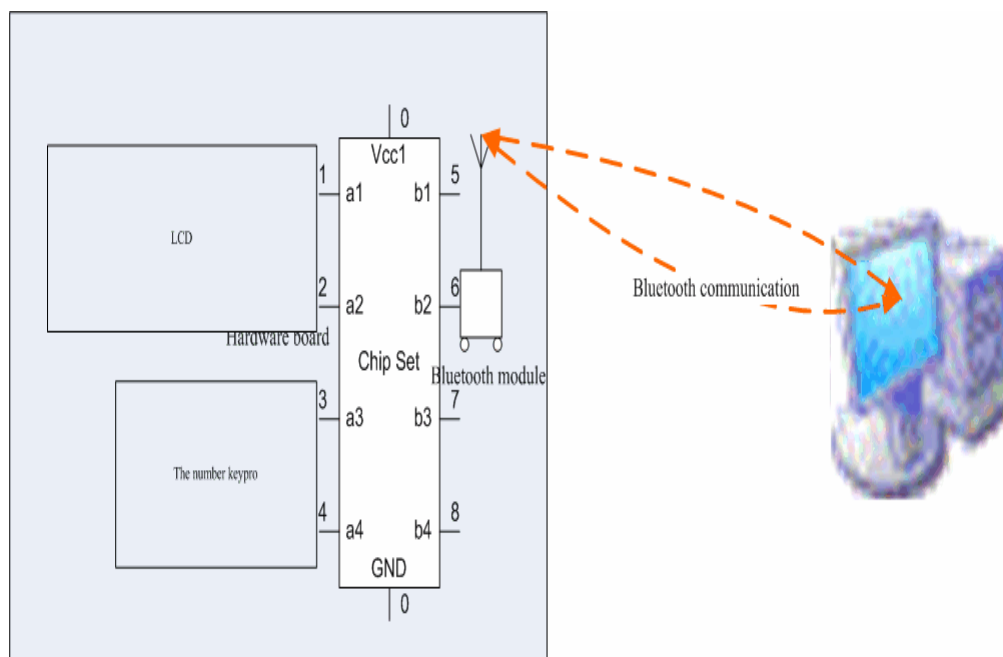


圖1. 藍芽通訊應用於商務案例的架構

3.2 系統規劃流程

該研究之實驗分為兩大區塊來進行，Server 端的部份為芽藍無線訊號接收端，與硬體裝置端信號發送端的部份，而在本篇研究文裡主要也是要探討如何借由 UML 語言的觀念與方法落實在韌體裝置的程式開發上。

3.2.1 PC 程式流程

如圖 2 所示，PC 端程式這部份亦有程式需要執行與裝置部份進行資料的溝通與配合，由於 PC 端的程式部份並非屬於這次研究主題之重點所在，因此在這只是約略的進行流程簡介。

PC 端的工作流程主要是讓 Server 端的程式啟動，再透過藍芽裝置管理軟體的連線進行 COM PORT 的對連後，經由 COM PORT 來與硬體裝置端的資料進行傳遞，最後將收到的 binary 資料轉成字串資料顯示於應用程式上，另一部份則為 receive 資料流程，硬體裝置有鍵盤資料的輸入動作，PC 端仍然是經由 COM PORT 去收受信號，再轉成鍵盤資料值顯示。

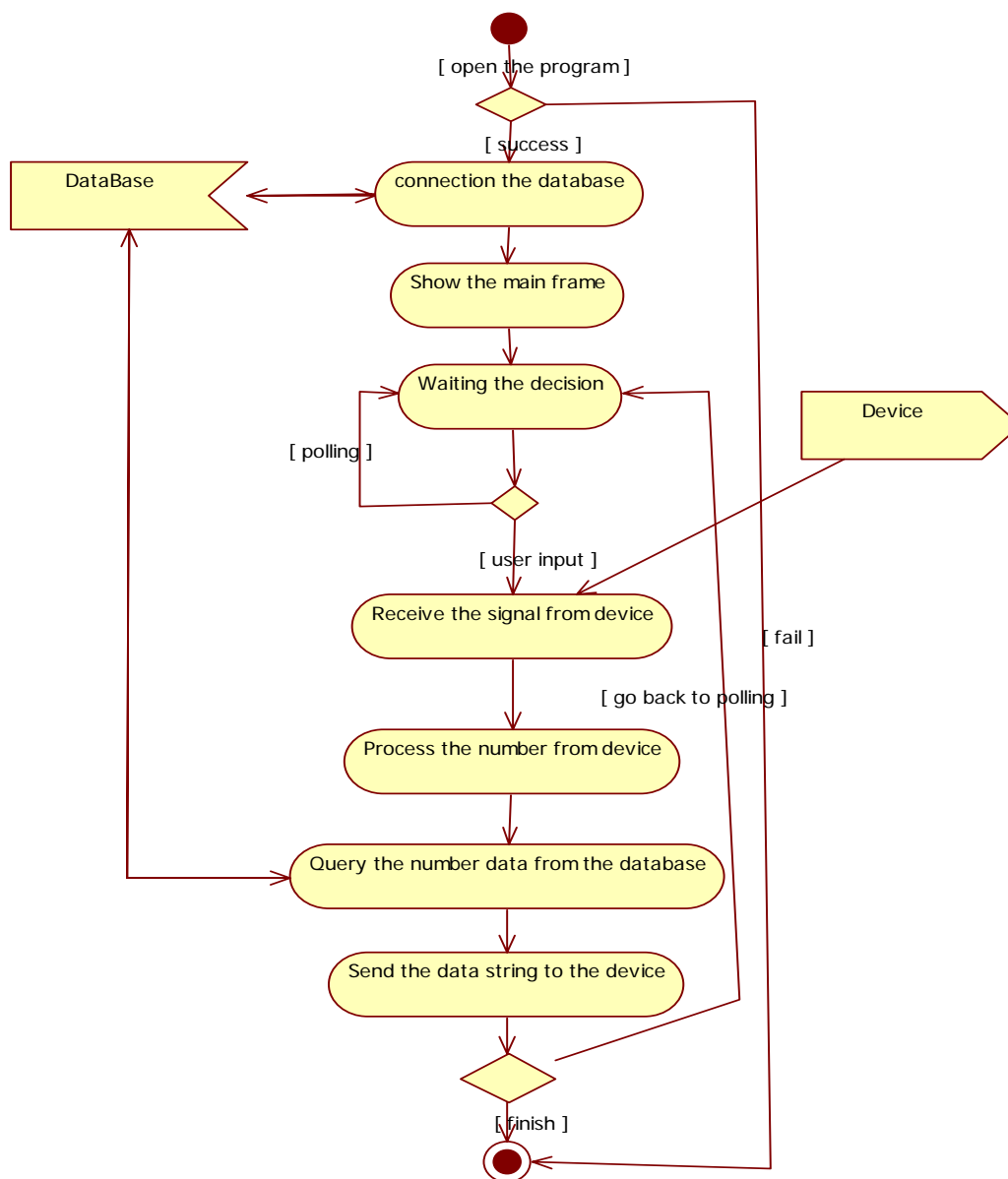


圖2. PC 端程式流程

3.2.2 嵌入式系統的程式流程

以下圖 3.乃是嵌入式系統上的流程，由於該篇研究主要是要探討如何透過 UML 實現於嵌入式系統的程式設計，因此在流程上涉及到每顆不同晶片它的獨立功能性與非獨立功能性的影響，因此這樣的流

程只能大致規劃出該硬體裝置所要擁有的功能與應用的向方，並未細部列出每顆單晶片開機之後所要進行的細部設定與啟動之部份，這樣也是有利於爾後針對不同的單晶系統，但卻享有相同之應用功能時得以快速的依循同樣的流程來進行開發，也不致造成與 UML 上的流程圖造成衝突性。

硬體裝置端的工作流程是啟動 MCU 後，進行各不同 MCU 的相關設定，啟動藍芽裝置的連線工作，進而被動的 polling 等待 server 端資料的傳遞，亦或者 client 端使用者有輸入資料後，透過藍芽與 PC 端進行鍵盤資料的發送溝通。

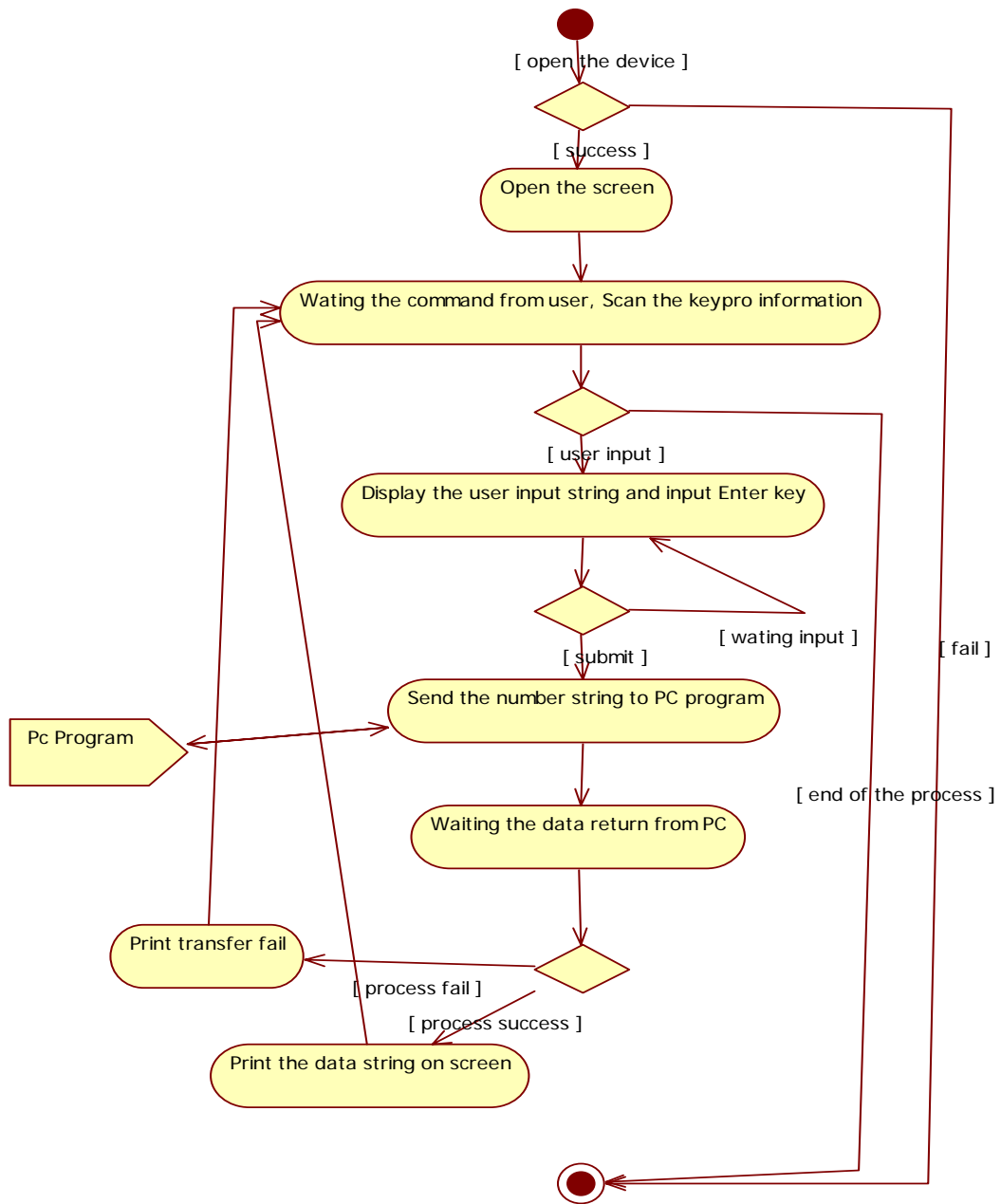


圖3. 嵌入式系統上的程式流程

3.2.3 硬體環境與條件的佈局

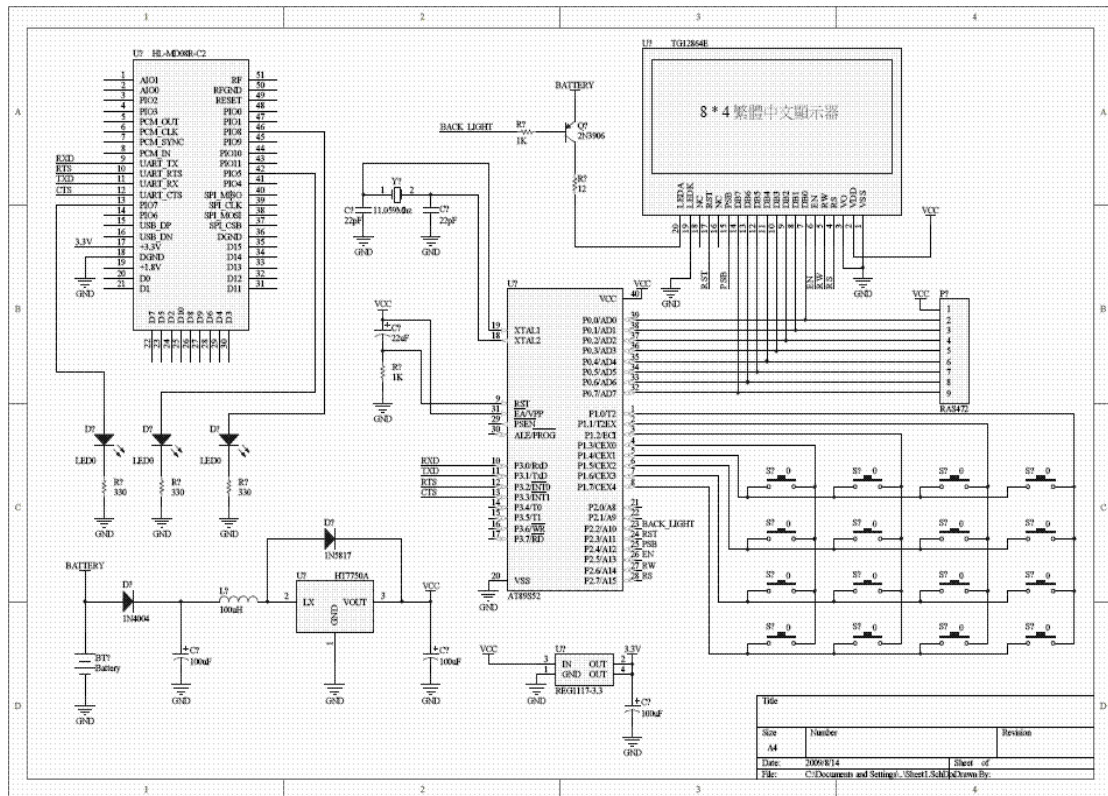


圖4. ATML 89C51 韌體佈局圖

再來則是要介紹硬體裝置的線路圖，於圖 4 與圖 5 所示，由於每個硬體皆有其不同的需求與特定的搭配週邊，因此也將產生不同的程式流程與不同的程式碼，來處理所需之硬體應用，該篇研究中的硬體裝置是將兩個不同的 MCU 設計於相同的一種電路圖功能應用上，讓 MCU 本身以外的外在環境相同的情況底下，進而證明透過 UML 協助下而發揮的效益。

為了簡潔且有利的證實透過 UML 得以有效的縮短工作時程，因此該裝置的週邊只有三項，且規格與型態都相同，唯一的差異只在

於 MCU 本身，Chip 是由分別為 ATML 89C51 and Microchip PIC16F877 兩顆不同類型與廠牌的單晶片所組成，這兩組電路版將在爾後實中作介紹。

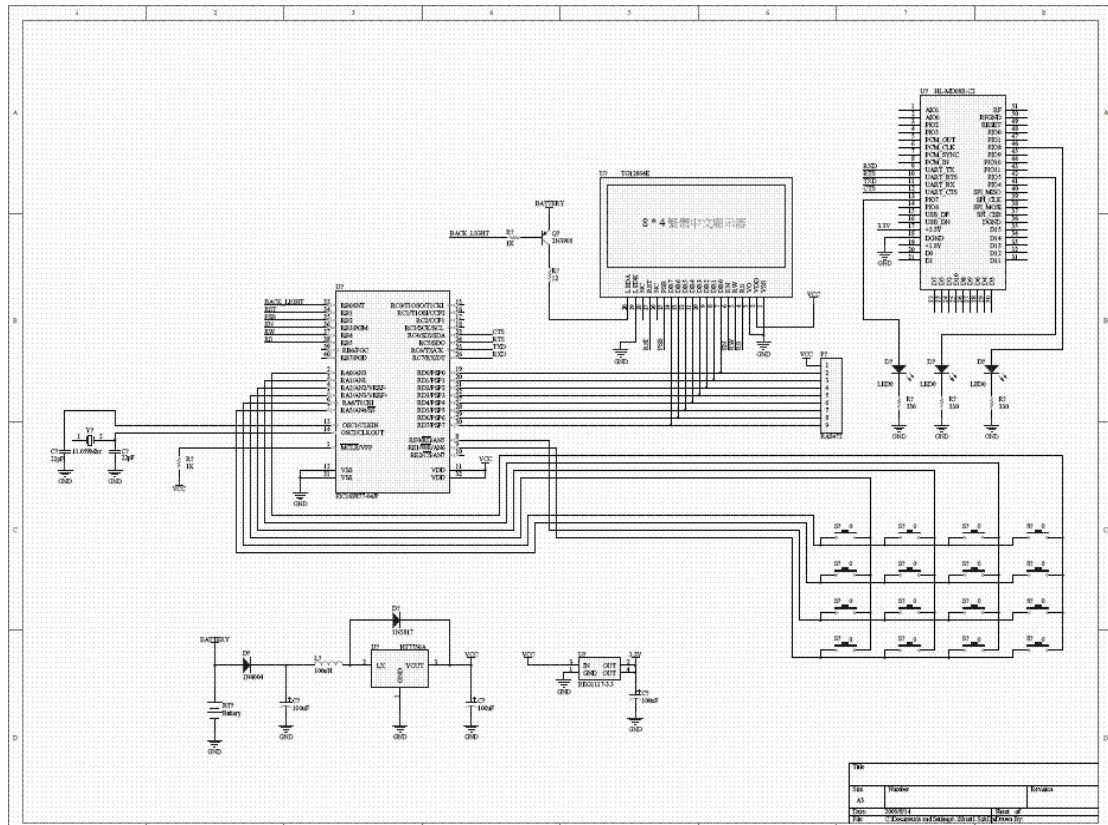


圖5. MICROCHIP PIC16F877 韌體佈局圖

3.3 傳統設計的系統流程

在不透過 UML 語言的協助下，最原始的系統流程如圖 3 所示，如果今天有兩片以上的韌體要設計時，就必需重複著以下流程方式從頭到尾重新設計一次，連基本的命名方式與系統初始值的設定都得重新規劃並重新建置一次，因此同樣的工作只因為他所設定的 MCU 不同所有的工作即得重新再規範，不但耗時也耗人力。

第4章 透過 UML 與傳統程式設計流程之間的差異

在本節中將介紹這個實驗專案如何借由 UML 語言來開發，進而達到兩組所屬不同 MCU 之實驗板能具有相同的功能，整個開發的過程介紹，做了什麼資料的分析之後，進而與傳統的開發流程進行比較，在開發時間上的差異與開發的流程上節省掉了哪些重複的動作流程，進而證明借由 UML 語言來取代傳統韌體的開發所帶來的效益。

4.1 UML 導入嵌入式系統的設計方法

借用其制定標準化的概念與圖示來塑造系統模型，整個系統的流程可以分成兩大部份，第一部份提供前端模型產生與循序圖運作方法 [12]；第二部份則是提供後端程式碼生成的工作

4.1.1 前端模型產生與循序圖

- 前端模型的產生

由於 UC 導向(USE CASE DRIVEN) [12] 是當今軟體開發的基礎模式，因此 UC 敘述常做為系統開發文件的匯集點，由它進而連結到攸關的檔案，也因為他當為檔案與程式開發者的一個介面，一旦在硬

體設備的需求有所變動時，可以快速的搜尋出以 UC 敘述為主的一連串相關資訊，而進行需求的快速修正。

當 UC 改版時，UC 敘述也跟著同步改版，UC 敘述是參與討論人員所留下的結果，也是該案子的重要資產，因此也需考量到版本的問題，所以在版本裡多加一個欄位連結舊有的歷史版本與改版說明，以便爾後查閱。

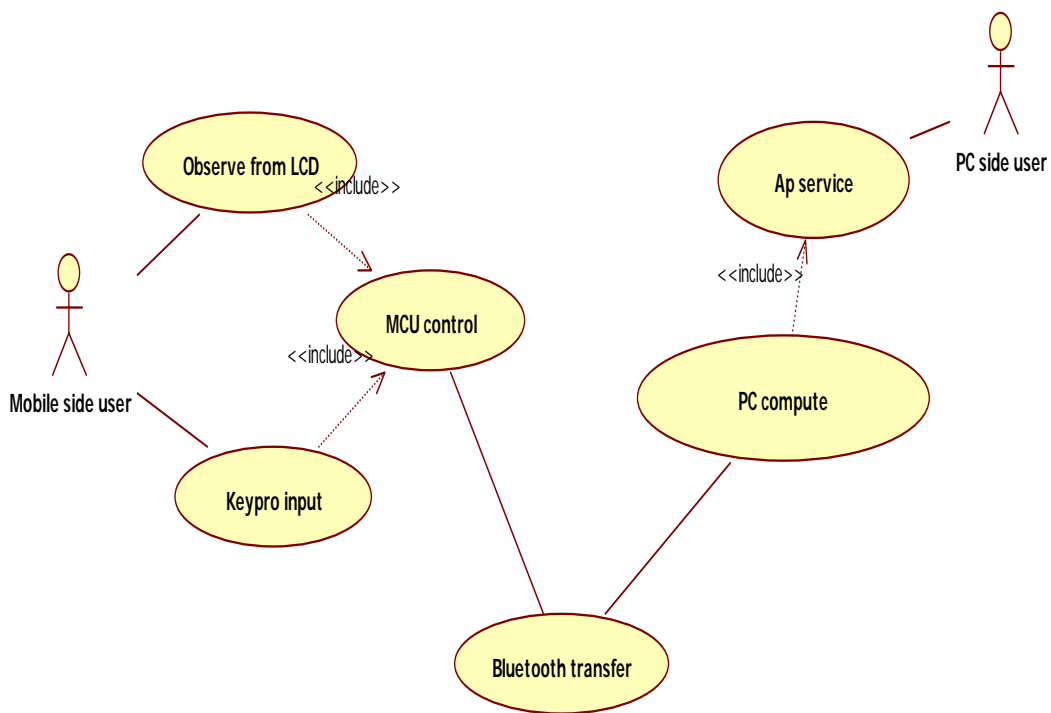


圖6. 系統 UC 圖

就基於以上的基礎之下，我們規劃出該韌體實驗所需的模型如圖 6，也因為該介面的易動性高，因此在未來如果其它的使用者對該硬

體架構有新增或異動時，只要透過該模板界面的修正即能迅速的修正使用者功能需求。

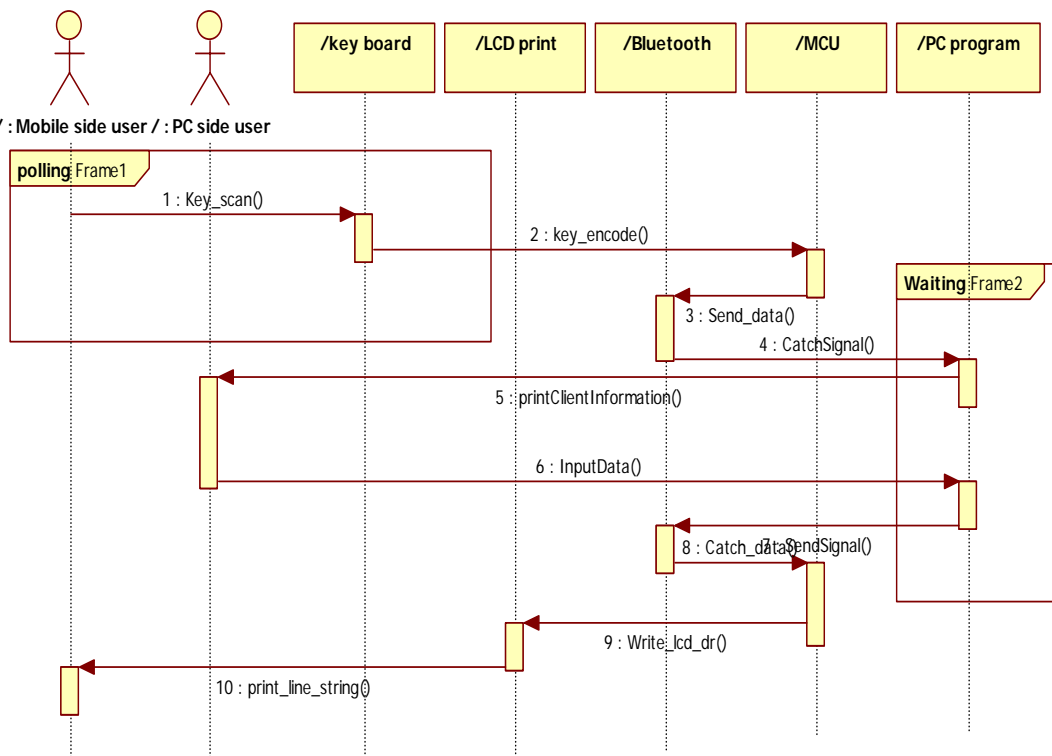


圖7. 循序圖

● 循序圖運作方法

在經過系統的分析過後，可以依據類別所產生出來的一群物件之間的互動，讓這樣一整組的物件可以合力完成某一個系統所表達的UC，同時在循序圖中，一整組的物件所引發的操作相關與順序，進而產生出來的循序圖(如圖 7)，意在傳送一連串訊息並引發操作的過程，形成一群物件互動的模樣。

4.1.2 後端程式碼生成的工作

這部份要進行的工作乃是將以上的程式透過 UML 語言的協助進而生成可執行的程式碼，我們採用的方法是由 Miro Samek 提出的量子程式法[11]，這種做法是將每個硬體元件都當作為是一個獨立的單元，每個單元都自行運作，單元彼此之間只藉由訊息的傳遞方式來建立聯繫，利用這樣的作法，不但方便未來硬體需求改變時快速的更改軟體的應用，我們也可以直接將使用者輸入的狀態圖轉成相對應的程式，如圖 8. 系統類別圖所示，這個系統包話了三個硬體元件，分別對應三個軟體元件，而另一個獨立軟體元件則是屬於 PC 端運作的功能。

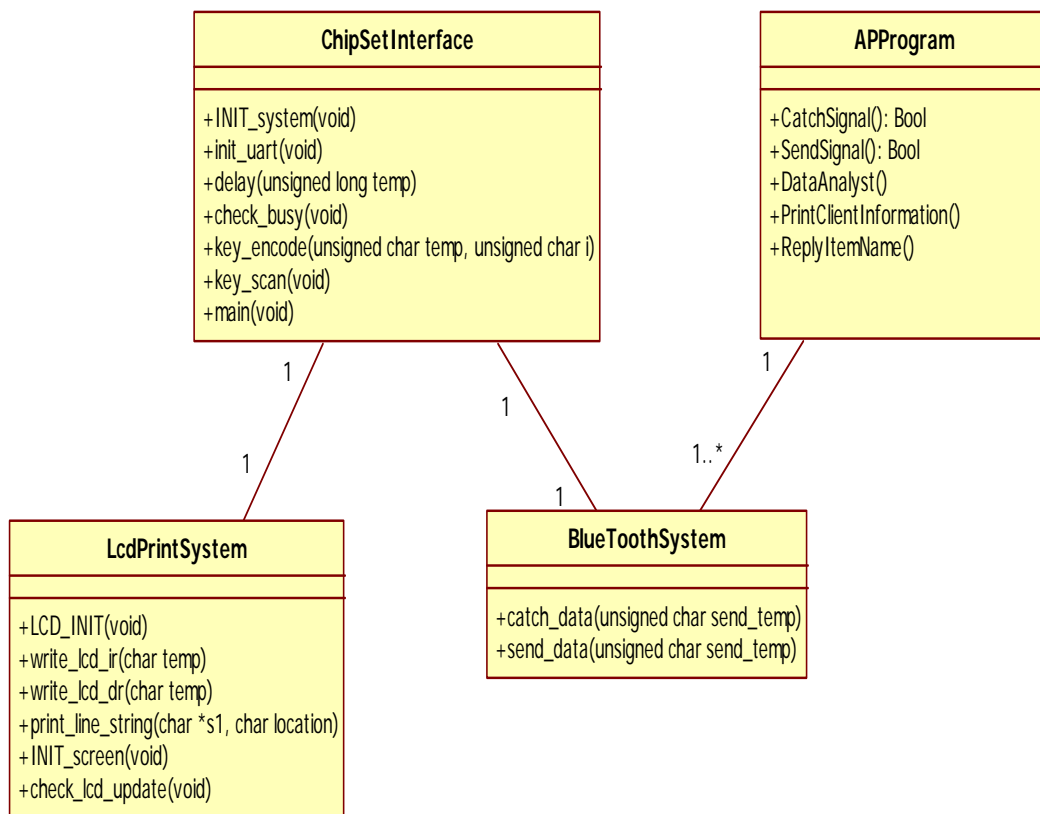


圖8. 系統類別圖

硬體物件分別是 MCU，Keyboard，Lcd，Bluetooth，其分別對應的軟體物件分別是 ChipsetInterface，LcdPrintSystem，BluetoothSystem。使用者得以利用鍵盤輸入按鍵值並顯示於 PC 端的 program。PC 端亦可借由發送字串信號，讓 device 端顯示由 PC 端所發送出來的訊息顯示於 LCD 螢幕上。

4.2 UML 與傳統程式設計間差異的數據比較

4.2.1 硬體上的比較

在該次的研究實驗中，我們是以假設兩組硬體裝置唯獨 MCU 有所差異之外，其週邊的設計除了穩壓的穩定功能之外，就該次的實驗兩組硬體裝置並無差異，但在實際的商業行為上，往往 MCU 以外的週邊通常變化非常大，即便腳位不同，對應到 MCU 所要工作的內容亦會有所差異，也因此在這裡是假設在硬體週邊環境皆為相同的情況下所進行的軟體差異化的實驗，因此硬體部份仍因需求而異去進行設計。

4.2.2 軟體上的比較

- 傳統設計方法

就以傳統的軟體設計方法而言，大致的設計流程如下圖 9 所示，當 MCU 一通上電流後，即開始要進行 MCU 相關 REGISTER 的初始化動作，這包函使用者的週邊硬體所設置的腳位，與將要開啟該 MCU 哪些部份的功能之致能與除能的工作皆需在此一 STEP 來進行設定完成初始化的動作。

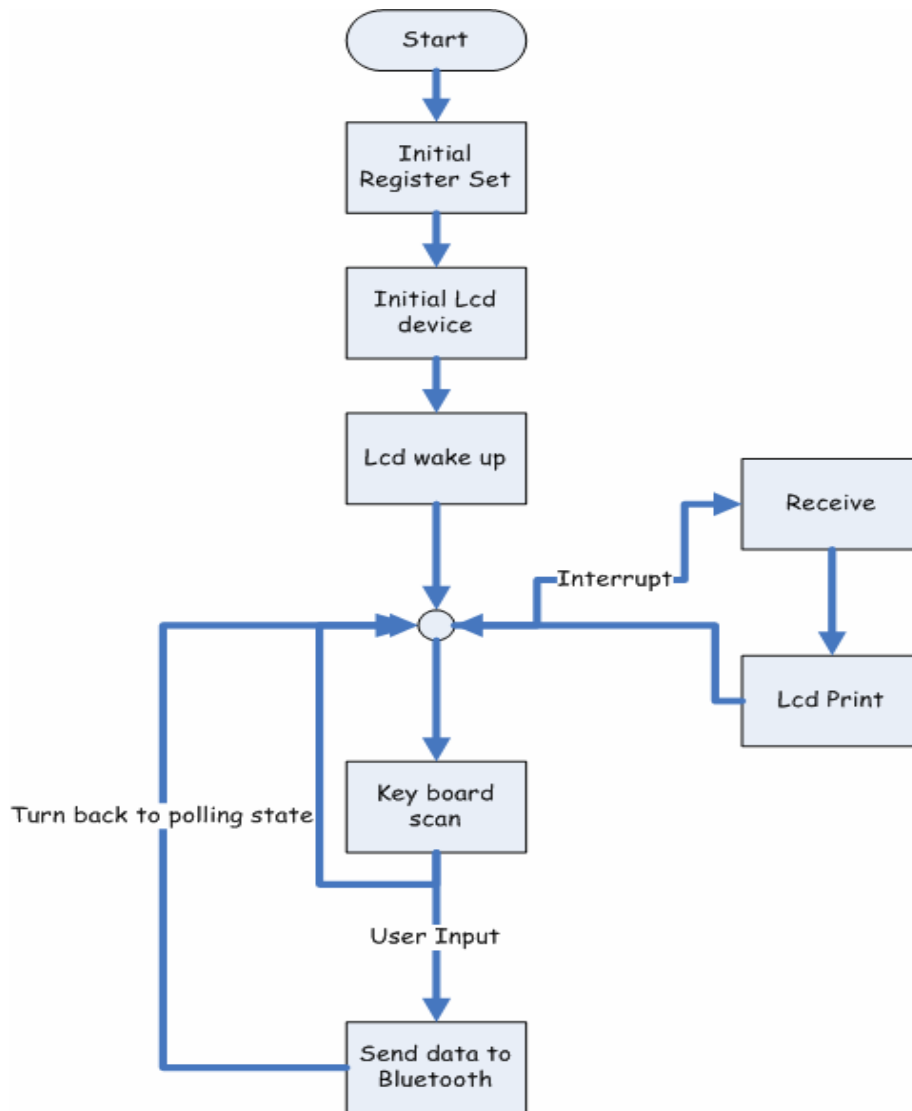


圖9. 傳統設計韌體流程

接著開始初始化週邊硬體的工作，LCD 的初始值設定，將要定義這片 LCD 所接的 MCU 腳位與解析等設定，接著才是啟動 LCD 顯示的功能。

同樣的藍芽裝置也得進行像 LCD 這樣的初始化設定，啟動該元件讓該元件得以讓 PC 端可以抓得到所要連結的藍芽信號。

最後整個系統進入 POLLING 的狀態，在這裡所等待的信號是偵測使用者是否進行鍵盤裝置輸入的動作，整個系統將一直在此等待使用者端輸入信號，或者來自於遠端透過藍芽裝置所傳送進來的中斷才進行下個 STEP 的動作。

萬一使用者在此時進行的按鍵輸入的動作時 MCU 會將使用者所按下的按鍵資料透過藍芽裝置這個介面將資料傳遞出去。

中斷功能，在這裡的中斷所等的促觸發乃是來自於藍芽裝置收到外來的信號訊息時，即會發一個中斷給 MCU，要求 MCU 到某個 QUEUE 裡去將所收到的資料撈取出來，進而顯示於 LCD 上。

接下來換了一塊硬體裝置，MCU 不同，所有的工作即從頭再來設計一次，不止腳位不一樣所要重新塑造的程式碼不同，就連不同廠的 MCU 所要設定給與 REGISTER 也將全數不同，就這樣只要 MCU 不同流程從頭到尾就全數重新重頭再執行，不但多工且重工。

● 差異數字比較

首先，我們的工作是將不同的 MCU 所據有不同的特性，獨立性的 FUNCTION 與腳位列舉出來，也就是先比較兩顆 MCU 在同樣的週邊情況下且要達到相同功能時，兩者間所發生的差異列舉出來，透

過 UML 語言的分析列舉出了下表 1 與表 2，分別表示出了 REGIESTER 與 FUNCTION 的不同，說明了 8051 與 PIC16 這兩顆 MCU 啟動時各自獨立的功能 FUNCTION 為何。首先，我們的工作是將不同的 MCU 所據有不同的特性，獨立性的 FUNCTION 與腳位列舉出來，也就是先比較兩顆 MCU 在同樣的週邊情況下且要達到相同功能時，兩者間所發生的差異列舉出來，透過 UML 語言的分析列舉出了下表 1 與表 2，分別表示出了 REGIESTER 與 FUNCTION 的不同，說明了 8051 與 PIC16 這兩顆 MCU 啟動時各自獨立的功能 FUNCTION 為何。

表1. 不同的 MCU 之暫存器定義

8051 MCU register	PIC MCU regiester.
<code>#include <REG51.H></code>	<code>#include "pic.h"</code>
<code>#define rs P0^2</code>	<code>#define rs RB5</code>
<code>#define rw P0^3</code>	<code>#define rw RB4</code>
<code>#define en P0^4</code>	<code>#define en RB3</code>
<code>#define psb P0^6</code>	<code>#define psb RB0</code>
<code>#define reset P0^5</code>	<code>#define reset RB1</code>
<code>#define light P0^1</code>	<code>#define light RB2</code>
<code>#define lcd_data P2</code>	<code>#define lcd_data PORTD</code>

表2. 需依賴不同 Mcu 指令之 Function 名稱.

MCU interface function.	
Void INIT_system(void)	
Void init_uart(void)	
Void key_scan(void)	
Void check_busy(void)	
Bluetooth system function.	
8051 MCU blue tooth system function.	PIC MCU blue tooth system function.
void uart_int(void) interrupt 4	void RX_interrupt(void)
Void send_data(unsigned char send_temp)	Void send_data(unsigned char send_temp)
	Static void interruptisr(void)

再藉由先前所介紹的 UML 設計架構所產生出來的程式碼來進行對不同 MCU 的移植動作，在這個研究中我們只要再補充一支 MCU 轉換程式，針對不同的晶片特質，獨立性的 FUNCTION CODE 與腳位定義進行轉換後，即可達到如表 1 所示的局部程式碼對換，由於每顆不同的 MCU 必然會有不同的 REGIESTER 與 PIN 腳的規定，所以我們無法要求每塊硬體裝置的週邊所要結合的 MCU 腳位都為一樣，因此這支轉換程式正好成為 UML Code Generator 架構產生出來後補足唯一缺點的有效工具，在經過程式的轉碼後即可讓不同的 MCU 卻具有相同功能的結果。

在分析比較過 UML Code Generator 與傳統設計的差異之後，我

們就可以依照這個實作的結果來得到精確的數據來證明使用 UML 所提供的效能與效率所帶來的差異數字如表 3。除此之外在開發設計的時間上也有實際的數據得以證明，一般傳統的設計開發時往往得透過邏輯的概念與思維去思考程式的下一個步驟該是為何，但由於我們透過 UML 的模型就整個架構限定於固定的行為之下，因此就這個實例上第二塊板子的開發被局限於固定的框架下，只需於腳位定義與 DEPENDENCE FUNCTION 的修改，而不需動到整體架構邏輯的思維，故節省相對可觀的開發時間(表 4)。

表3. 透過 UML Code Generator 設計與傳統設計 Coding 行數差異

傳統的開發流程所要撰寫程式碼行數	第二塊電路板所要撰寫的程式行數約為 280 行
透過 UML 開發流程所要撰寫程式碼行數	第二塊電路板所要撰寫的程式行數約為 128 行

表4. UML 設計與傳統設計之開發時間差異

傳統開發時數	第二塊電路板所花費時間約為 18 小時
透過 UML 工具後開發時數	第二塊電路板所花費時間約為 6 小時

第5章 實作案例

我們使用一個實例來實驗這個方法的運作，我們採用的例子是一個「遠端藍芽通訊系統」參照下圖 10，是透過藍芽傳輸的方式來讓電腦與裝置之間的資料能進行溝通與傳送的基本功能模型。

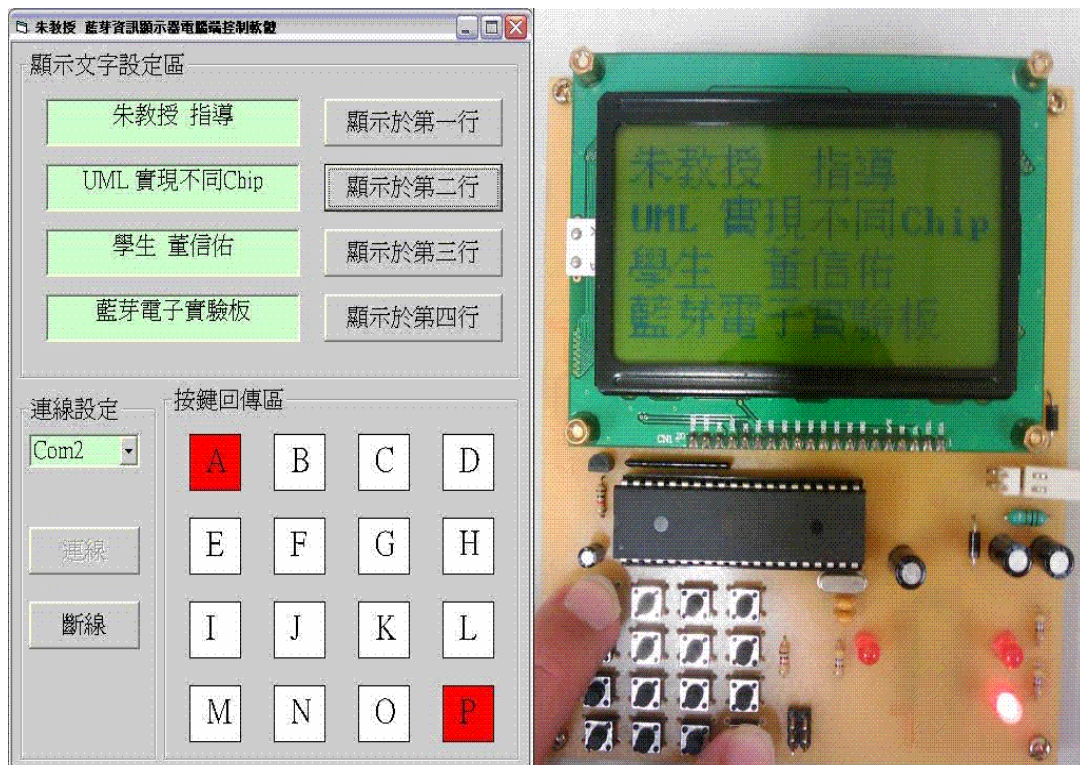


圖10. 遠端藍芽通訊系統之實作模型

此次的實作其用意在於證明 UML 所提昇的工作效益，因此在實作的功能上儘量讓使用者得以簡單明白的操作該系統為主軸，爾後不管是在 LAPTOP 的開發上或是 PC 電腦的開發上，只要有涉及到鍵盤或藍芽的週邊硬體時都得以透過這樣的設計規則來進行，而整個實驗之後也確實在開發速度上有效提昇。

該實作的功能就使用者介面而言主要分為三區塊，分別為 COM PORT 的選擇與連結，數字按鍵的操控與接收，資料的輸入與輸出。

● COM PORT 的選擇與連結

在這個硬體裝置要使用時，使用者的第一個操作功能就是先得知個人電腦上的藍芽裝置是否捕抓到這個硬體裝置的存在，其次是透過藍芽管理軟體與該硬體裝置進行連結動作，這個動作之後使用者必需了解電腦上的藍芽與硬體裝置的連結 COM PORT 為何，通當電腦端的 COM PORT 往往開放第九 PORT 之後的讓使用者使用，然而在前九 PORT 之前大部份階屬於預設的 DEVICE 所使用，由於受限於在 PC 端開發的應用程式元件的限制，導致該 APP 的 COM PORT 最多只列舉出 COM 0 到 COM 9 的通訊 PORT 得以使用，故於選取通訊 PORT 時需先清楚了解您電腦中的哪些 PORT 是得以被最之應用的，得知電腦與硬體裝置連結之 COM PORT 後，使用者才得以在這個應用模組的應用程式上選擇所要接收資料的 COM PORT，進而按下連線按鈕後進行資料傳輸工作。

● 數字按鍵的操控與接收

在 COM PORT 選取並連線之後，使用者得以在硬體裝置端按下

數字鍵，數字鍵的編號是一組十五顆按鍵組成的鍵盤，在使用者按下數字鍵的同時，硬體裝置會透過本身的藍芽裝置將按鍵的信號傳送到電腦端的應用程式，在應用程式收到所屬的協定信號時，會到資料佇列裡將資料取出，並顯示出使用都在硬體裝置端所按下的按鍵值。

由圖 10 可以看出使用者經由 DEVICE 上按下 A 與 P 鍵，再經由藍芽的傳輸方式傳到 PC 端的 APP 接收，因為不管是 DEVICE 或者是 APP 所發送的資料或收授的資料都是屬於連續資料，其功能特性與一般的無線鍵盤相當接近，故當 APP 端收到資料時 A 與 P 的 ICON 將呈現紅色處於資料收受取得的狀態。而當使用者放開 DEVICE 上的按鍵，此時資料的傳送即會中止，而在 APP 端 ICON 所呈現的顏色也將回覆到原來未被按下時的顏色，如此即完成由 CLIENT 端資料的輸入傳送到 SERVER 端資料的接收整個 CYCLE 的流程。

● 資料的輸入與輸出

使用者可以在應用程式的 TEXT BOX 裡輸入想要打的文字資料，資料型態不限定中文或英文，而資料的長度最長限定為八個中文字，之所以會限定為最多八個中文字是因為 LCD 顯示器所能顯示的中文字元最多只為如此，而如果顯示字為英文文字其英文字最長得以

十六個英文單字，在輸入資料後使用者可以按下右邊顯示第幾行的按鈕，這時應用程式會透過 COM PORT 將資料傳送到硬體裝置端接收，當 MCU 收到中斷信號時會到佇列中抓取傳送過來的字串資料，並將該資料顯示於 LCD 面板上，此時應用程式上使用者輸入的字串訊息同時也會顯示在硬體裝置的 LCD 上。

第6章 結論

在這篇論文中，我們提出了利用物件導向模型來建立嵌入式系統軟體設計的方法。首先我們使用工業界廣泛使用的 UML 作為系統輸入模型，提供建立模型的便利性與易動性的功能。再來我們借由 UML 排程的技術，一方面確保程式功能的正確性，另一方面確保程式能夠滿足系統的時間需求。接著我們借由量子程式技術[11]產生各元件獨立運作的觀念將其實現於嵌入式系統程式中。另外嵌入式裝置周邊元件的切割可讓使用者直接取用現存的周邊硬體元件，免除重新開發常用硬體元件及熟悉軟硬體平台的時間。最後，我們使用一個「遠端藍芽通訊系統」的例子來證明我們框架的正確性與迅速性，透過例子的實作也可以發現使用這種方法設計嵌入式軟體的優點與效率。

參考文獻

- [1] J. Rumbaugh, G. Booch, and I. Jacobson, “*The UML Reference Guide, Addison Wesley*”, 1999.
- [2] S. Wang, S. Kodase, and K. G. Shin, Automating embedded software construction and analysis with design models, “*Proceedings of International Conference of Euro-uRapid 2002, Frankfurt*”, Germany, December 2002.
- [3] D. de Niz and R. Rajkumar, Time Weaver: A software-through-models framework for embedded real-time systems, “*Proceedings of the International workshop on Languages, Compilers, and Tools for Embedded Systems, ACM Press*”, 2003.
- [4] T. Wang, X.-G. Zhou, B. Zhou, L. Liang, and C.-L. Peng, “A MDA based SoC Modeling Approach using UML and SystemC,” in Proceedings of the sixth IEEE International Conference on Computer and Information Technology (CIT’06). IEEE Computer Society, September 2006, pp. 245–245.
- [5] P. Kukkala, J. Riihimaki, M. Hannikainen, T. D. Hamalainen, and K. Kronlof, “Uml 2.0 profile for embedded system design,” in DATE ’05: Proceedings of the conference on Design, Automation and Test in Europe. Washington, DC, USA: IEEE Computer Society, 2005, pp. 710–715.
- [6] E. Riccobene, P. Scandurra, A. Rosti, and S. Bocchio, “*Designing an Unified Process for Embedded Systems*,” in Fourth International Workshop on Model-Based Methodologies for Pervasive and Embedded Software. IEEE Computer Science, mars 2007, pp. 77–90
- [7] M. Rieder, R. Steiner, C. Berthouzoz, F. Corthay, and T. Sterren, “*Synthesized Uml, a Practical Approach to Map Uml to VHDL*,” in RISE, 2005, pp. 203–217.

- [8] E. Riccobene, A. Rosti, and P. Scandura, “*Improving SoC Design Flow by means of MDA and UML Profiles*,” in 3rd Workshop in Software Model Engineering (WiSME 2004), october 2004.
- [9] Open SystemC Initiative, “*OSCI TLM2 User Manual*,” 12 2007.
- [10] P. Kukkala, M. Setälä, T. Arpinen, E. Salminen, M. Hannikainen, and T. D. Hamäläinen, “*Implementing A Wlan Video Terminal Using Uml and Fully Automated Design Flow*,” EURASIP J. Embedded Syst., vol. 2007, no. 1, pp. 20–20, 2007.
- [11] M. Samek, “*Practical Statecharts in C/C++ Quantum Programming for Embedded Systems*”, CMP Books, 2002.
- [12] 邱郁惠, “*寫給 SA 的 UML/MDA 實務手冊*”, 上奇科技出版, 2007.
- [13] 張裕益, “*UML 理論與實作 – 個案討論與經驗分享*”, 博碩文化, 2002.

附錄 藍芽實作裝置操作步驟

- 一、 安裝藍芽監測程式 BLUE SOLEIL
- 二、 插上藍芽接收裝置，如附錄圖(一)



圖1. 附錄圖(一)

三、 執行藍芽監測程式 BLUE SOLEIL 如附錄圖(二):

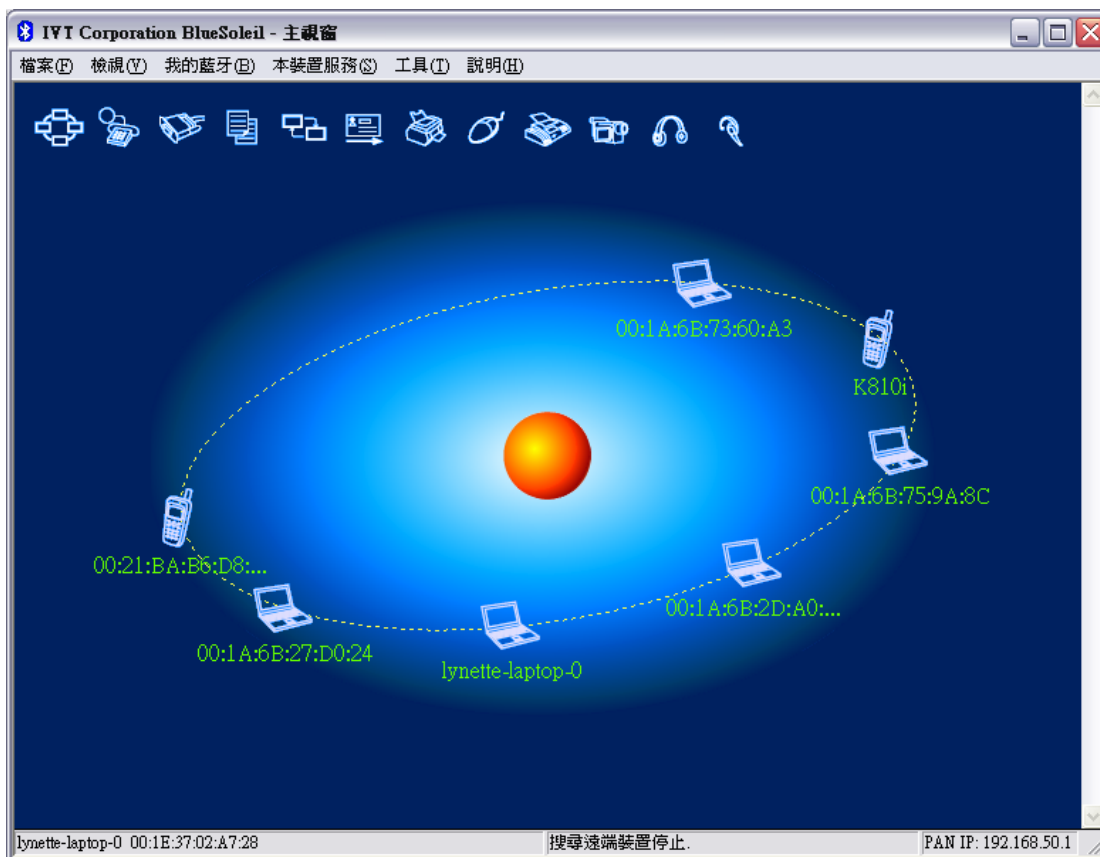


圖2. 附錄圖(二)

四、 啟動藍芽裝置，如附錄圖(三)

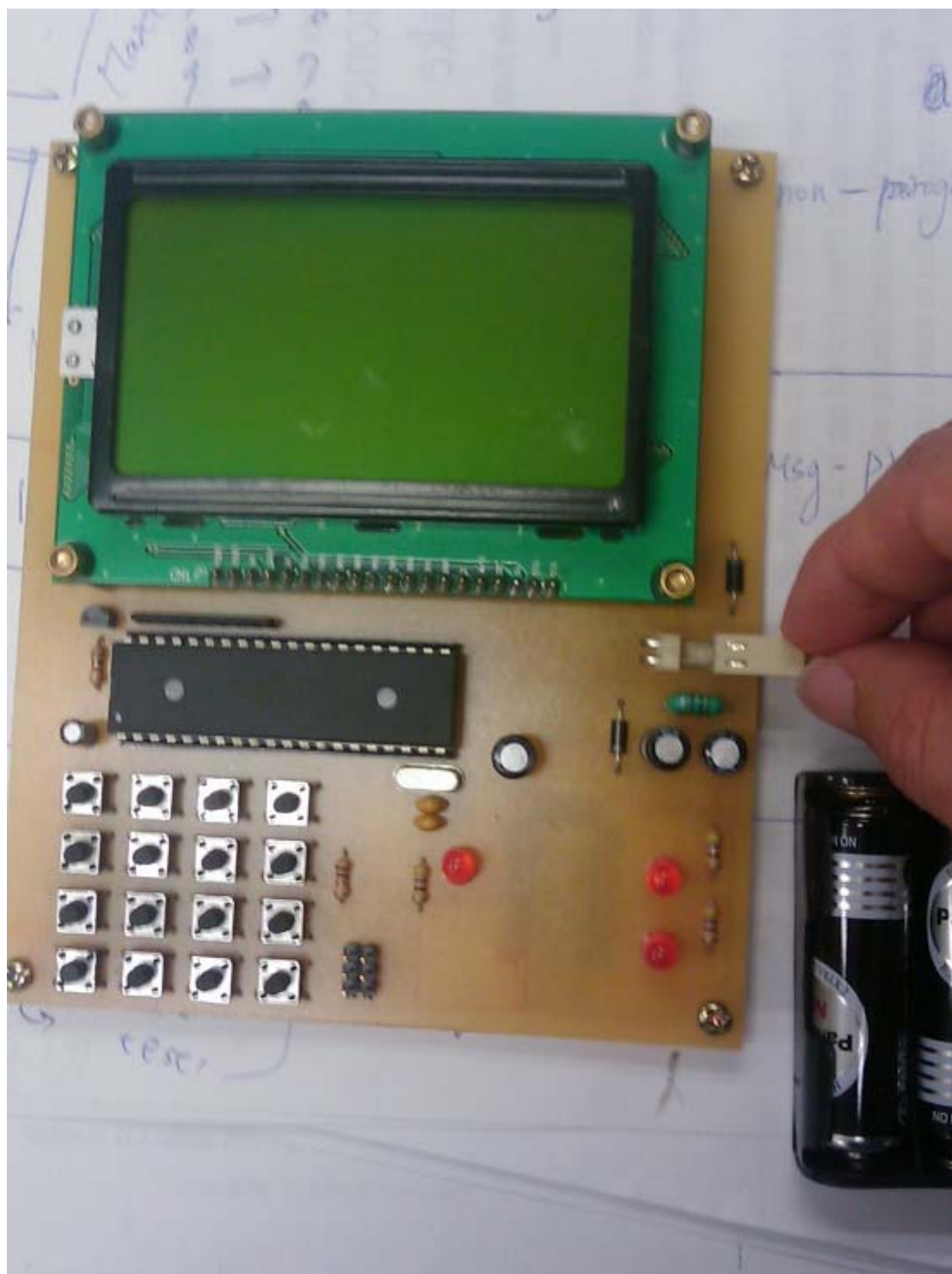


圖3. 附錄圖(三)

五、 回到藍芽監測程式 BLUE SOLEIL，並按 F 5 重新整理所

搜尋到的資料 如附錄圖(四):

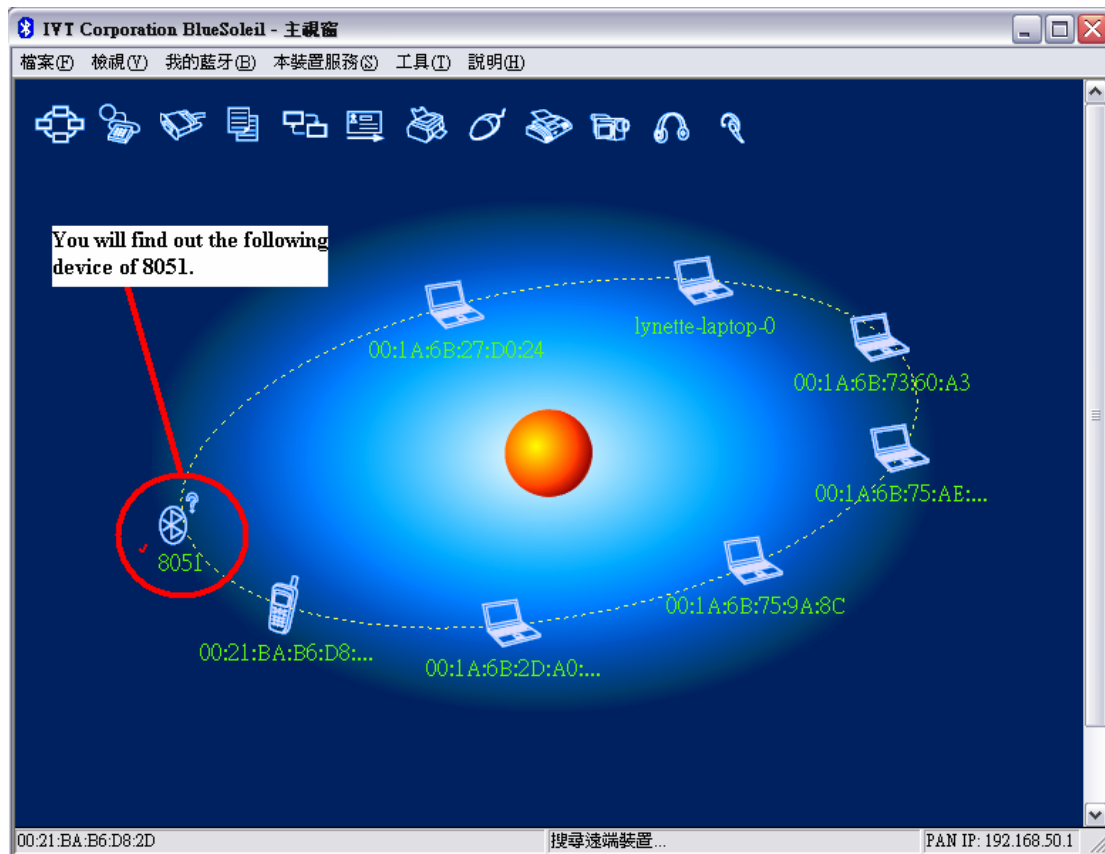


圖4. 附錄圖(四)

六、 當您於藍芽監控程式中發現到該 device 之後，請於該 device 上按兩下進行與 device 的連線動作，在該 device 連線之後會呈現黃色的圖樣，如附錄圖(五)

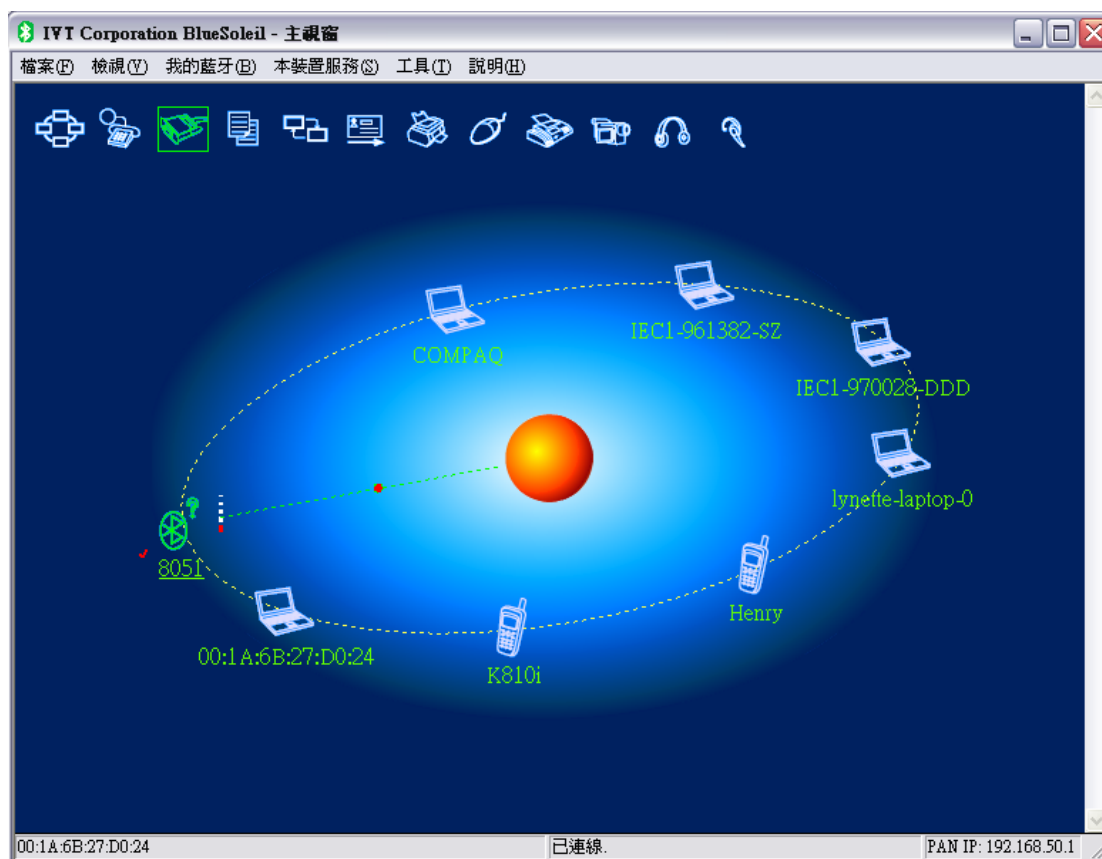


圖5. 附錄圖(五)

七、 當藍芽監控程式已成功與 **DEVICE** 連線之後，我們得以執行

SERVER 端的應用程式

八、 應用程式執行起來之畫面，如附錄圖(六)



圖6. 附錄圖(六)

九、 依該 **APP** 的指示您必需選擇所要連結的 **COM PORT**，因

此需再回到藍芽監控程式裡，點選連線的 DEVICE(8051) 按右鍵後，點選狀態選項，其 DEVICE 的資訊將如下圖附錄圖(七)



圖7. 附錄圖(七)

十、 借由上圖的資訊我們得知所使用之 **COM PORT** 為 **COM 7**，因此回到應用程式中，選擇 **COM 7** 並按下連線，連線後的 **APP** 狀態，如圖附錄圖(八)



圖8. 附錄圖(八)

十一、 此時 **SERVER** 的 **APP** 與 **CLIENT** 端的 **DEVICE** 已完成

連線，因此可以開始進行下一階段資料傳輸與交換的部份。

十二、在 DEVICE 上當您按上按鍵 A 與 P 時，您將看到如下圖

附錄圖(九)與(十)結果顯示：

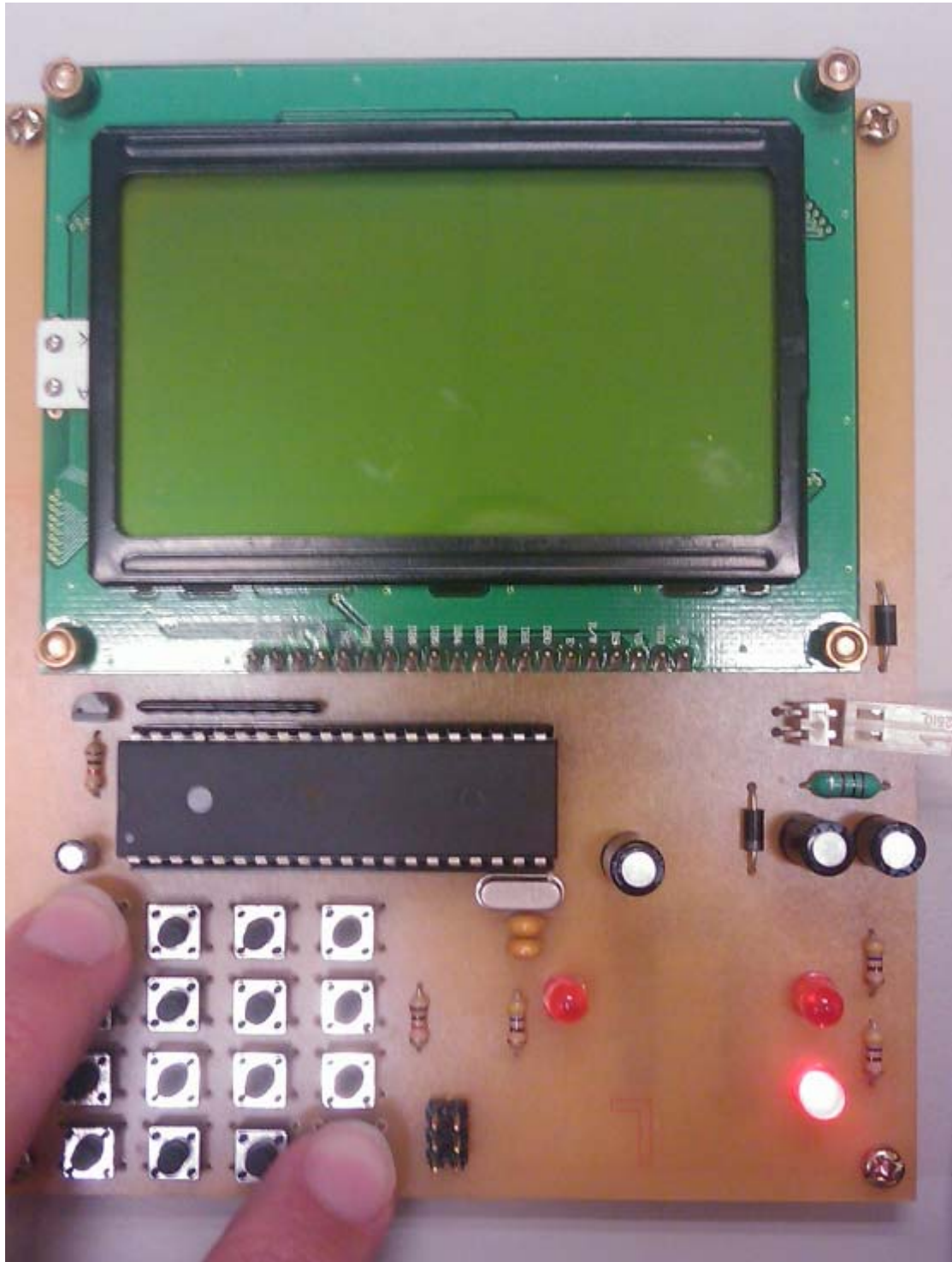


圖9. 附錄圖(九)



圖10. 附錄圖(十)

十三、此時在 **CLIENT** 端硬體收受到按鍵按下的資料，經過藍芽通訊的方式傳送到 **SERVER** 端的 **APP**，並將收到的訊息顯示於 **APP** 的按鍵面版上。

十四、由於該項設計是屬於雙向互通的方式，因此我們亦得以由 **SERVER** 端的 **APP** 輸入資料，再按下右邊的 **BUTTON** 鍵，此時 **APP** 上看到的訊息如下附錄圖(十一)



圖11. 附錄圖(十一)

十五、當您按上右邊按鍵「顯示於第一行」後，您得以回到 **CLIENT**

端的 DEVICE 上，我們則可以於下附錄圖(十二)看到其結果



圖12. 附錄圖(十二)

十六、 步驟至此，已完成了 CLIENT DEVICE 端對 SERVER 端 APP 傳送相關按鍵資訊的流程，也完成了 SERVER 端 APP

對 CLIENT 端 DEVICE 進行串列資料傳送的整個流程。

十七、 以往該種應用可以應用於電子廣告看板、賣場的服務定位鈴，以及更大者甚至為點餐系統的應用上。