逢　甲　大　學

資訊工程學系專題報告

PDA遙控家電

學生：　吳宏澤(四甲)
蔡宗翰(四甲)
孫誌明(四甲)

指導教授：　李維斌老師

中華民國九二年十二月

目錄

圖目錄

表目錄

第一章　緒論

1.1動機

近年來 PDA 日溢普及，隨著 PDA 技術的發展功能也越來越多及強大，從一開始只有著簡單記事、行程排定等等功能漸漸的發展到像一台隨身的小型電腦一般，且極具攜帶性，雖然礙於硬體上的不足，但是上面也有著人們常用的 OS 系統及一些常用的軟體，更勝者也已經能夠藉由無線網卡來遨遊網路。

也由於 PDA 的前景看好、市場需求量漸增，因此越來越多的人投入於 PDA 的額外功能之開發，在一年級時便聽到幾個學長開發了一套 PDA 導引系統，人們只要藉著 PDA 便能夠查詢在台北所有公車及捷運的所有資訊，像此類的功能便使得 PDA 逐漸的融入人們的生活之中，使 PDA 生活化。

有鑑於此，我們便也向用 PDA 來開發一套在生活中實用的系統，但是起初也不知道要做什麼，因為可以做的範圍實在太多太大，很難下定決心去做哪個，後來看了一組學長的專題，是有關資訊家電網路化的論文，學長他們的原意識人在外便能夠透過網路來控制家裡的家電，使得人們回家後便能夠享受到舒適的環境，看到這我們便想說用 PDA 來控制家電似乎也是個不錯的點子。

1.2　目的

至此我們便決定來開發一套 PDA 控制家電的系統，我們是打算用 PDA 來取代在家中所有家電的遙控器，由於在所有家電中每個有遙控器的家電都有其專屬的遙控器，可是隨著家中此類家電產品越多，家中所擁有的遙控器也越多，這造成人們在使用上的不便利，因此我們便想說不如把 PDA 來當成家中的總遙控器，把家電透過系統上的整合，人們便能夠使用 PDA 來控制家中所有的家電，同時透過這一套系統也能夠使得一些十分簡易但日常生活上不可或缺的家電例如電燈等納入可被人們所控制的範圍之內，只要人們手中拿著 PDA，不管人處在家中何處便能夠輕易的控制著家中所有的家電。

1.3 系統介紹

為了能夠實現此一想法，我們便開始進行要完成這系統的分析，經過初步的分析發現到只憑我們要做到能夠控制每個家電的所有功能顯的有點天方夜譚，因此我們決定只做到能夠對家電進行開啟關閉的動作來證明此系統的可行性。

首先為了要只使用 PDA 便能夠控制家中所有家電，因此便需要對所有的家電進行整合的工作，在整合的過程中我們使用一台 PC 來當 SEVER，再以此 PC 的 RS-232 介面連結 ADAM，再透過 ADAM 上的介面來連結到所有的家電，並且把所有的家電編上其個別的 ID 存放入 PC 裡所建構的資料庫中，而人們則能夠透過對 PDA 的操控再經由 PDA 與 PC 間的訊號傳遞以達到對家電控制的目的，而在此結構中 ADAM 是一個十分好用的器具，它能夠有效的處理來自 PC 的控制訊號，來判斷到底要開啟或關閉哪個家電，同時也能夠將家電的反應狀態來回傳給 PC，藉此來確定家電是否有正確的按照命令來動作。

其次對於要實現出能夠在家中任何地方控制家中所有的家電因此必須對於 PDA 與 PC 之間的訊號傳遞方式選擇一適當的方式，在此就不得不面臨有線以及無線的選擇，無線可說是近來的熱門話題，幾乎所有硬體產品廠商都以提供這項功能為號召。無線這個詞使人對不須惱人牽絆網路線便能在家中不同房間或是戶外進行網路漫遊充滿了憧憬。而有線卻不得不受制於線路的限制，影響其使用上的活動性。因此在訊號的傳輸方式上我們便選擇了以無線訊號的方式。

對於無線訊號傳輸我們決定以無線網路的方式來加以實現，分別在 PDA 與 PC 上加裝無線網路卡，藉由這兩張無線網路卡來使得 PDA 與 PC 之間互相取的聯繫，使之能夠順利的相互溝通，架構一個只屬於家中的小型區域網路，也由於無線網路卡訊號傳遞方式是屬於電波的方式，其特性為傳遞距離遠、無方向性且穿透性強，因此不存在著一般家電紅外線遙控器所存在的方向性問題，且訊號的傳輸有效範圍也能夠涵蓋著整個屋子，藉由這些特點我們便能夠實現出人處在家中任何地點便能夠對家中所有的家電進行控制。

但是使用無線網路傳輸的同時也存在著它所擁有的潛在問題，像是資料的機密性、認證、授權、完整性以及時效性上的問題，雖說此

系統只是家庭用來控制家電的系統，在傳輸的資料上不外乎都是些對於家電控制的訊號，但在現今的社會上很難說不會有人會對系統所傳遞的資料加以擷取利用，例如趁沒人在家對系統傳送所擷取的資料來隨意開啟家中關閉的耗電家電，以造成電的浪費。或者在資料上沒有識別性，由於無線電波的特性可能造成誤操作別家使用同套系統的家電，造成你開電視別人家的電視也被你所打開的情形產生。

為了避免這些情形的產生，因此對於資料的加以加密、認證便是必須的動作，同時各家電也必須編上其個別的 ID，藉此來提高此系統的安全性及使用上的可靠性。

做完對系統的大略分析，接著便是去收集相關技術所需的各類資訊，以及進行系統的實作，下面的章節便逐一介紹系統中所用到的各類技術、系統架構以及實作的成果。

第二章　背景

在本章將介紹所使用的硬體裝置以及相關軟體和安全性與協定的背景知識。

2.1 ADAM

ADAM 為研華所生產內含微處理機的感應模組，分為數個系列，可接收數位及類比訊號，各模組間以 RS485 的網路協定運作，其模組會自動偵測資料流方向，所以並不需要 handshaking signals，並以 RS232 連接電腦。ADAM 支援數位與類比間的轉換，某些模組可透過繼電器來控制開關。ADAM 本身並無開關，可利用 RS232 透過電腦傳送 ASCII 格式的指令來設定以及改變狀態，並回傳一串同樣為 ASCII 格式的字串，因此幾乎能應用在各種語言。

2.1.1 RS485 網路

RS485 為半雙工雙向性的網路傳輸標準，並使用 RTS〈Request To Send〉來控制資料流的方向，大多被應用在工業界，ADAM 可透過 RS485 REPETER 來連接最多 256 個模組，距離可達 4000 呎〈1200 公尺〉，並可透過 ADAM RS232/RS485CONVERTER 與電腦連接。

2.1.2 ADAM 4520

ADAM 4520 為一 RS232/RS485 CONVERTER，可讓電腦使用原有的 RS232 介面，透過此一模組轉換成 RS485 網路格式。此模組會自動感應資料流方向，並與其他模組通訊，因此並不需要 handshaking 來控制資料流，並將 RS485 線路數量減少到 2 條，其傳輸速度可達 115.2kbps。

圖 2-1　ADAM 4520

規格：

- Automatic internal RS-485 bus supervision
- No external flow control signals required for RS-485
- 3000 VDC isolation protection (ADAM-4520 only)
- Transient suppression on RS-485 data lines
- Speed up to 115.2 kbps
- Networking up to 4000 feet
- Reserved space for termination resistors
- Power and data flow indicator for troubleshooting
- Power requirement: +10 to +30 VDC
- Mounts easily on a DIN-rail or panel

2.1.3 ADAM 4050

ADAM 4050 為數位 I/O 模組，擁有七個輸入及八個輸出，我們可從電腦控制八個輸出，並且可將輸出連接至繼電器來控制電力設備，然後再透過輸入監控狀態。

圖 2-2 ADAM 4050

規格：

I/O Channels	7 inputs 8 outputs
Input/Output	RS-485 (2-wire)
speed (bps)	1200, 2400, 4800, 9600, 19.2K, 38.4K
maximum distance	4000 ft. (1200 m)
Digital Output	8-channel open collector to 30 V
sink-current	30 mA
power dissipation	300 mW
Digital Input	7-channel
logic level 0	+1 V max.
logic level 1	+3.5 to +30 V
Pull-up current	0.5 mA, 10K resistor to +5 V
Watchdog timer	Yes
Power supply	+10 to +30 V_{DC} (non-regulated)
Power consumption	0.4 W

2.1.4 ADAM Utility

ADAM Utility為研華公司所提供，可用來裝配ADAM4000及5000系列的圖形使用者介面，所提供之功能可監控各模組之I/O埠及該模租各項功能調整。由於模擬電路板剛拿到時無法確認是否有損壞，便以此軟體提供之功能先行測試結果。下兩圖為該軟體使用畫面。

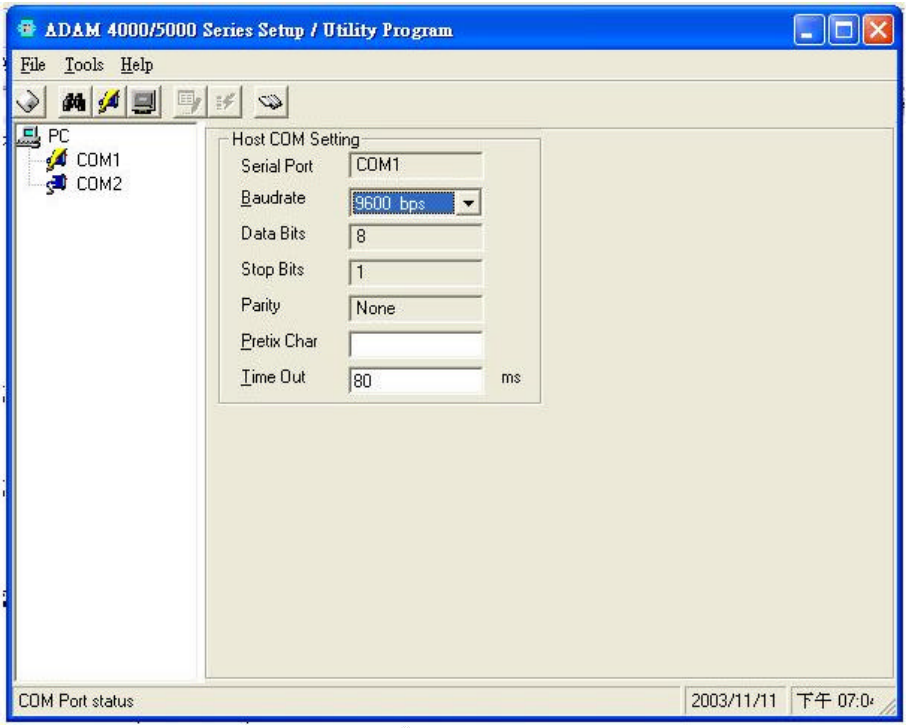


圖2-3　ADAM Utility主畫面

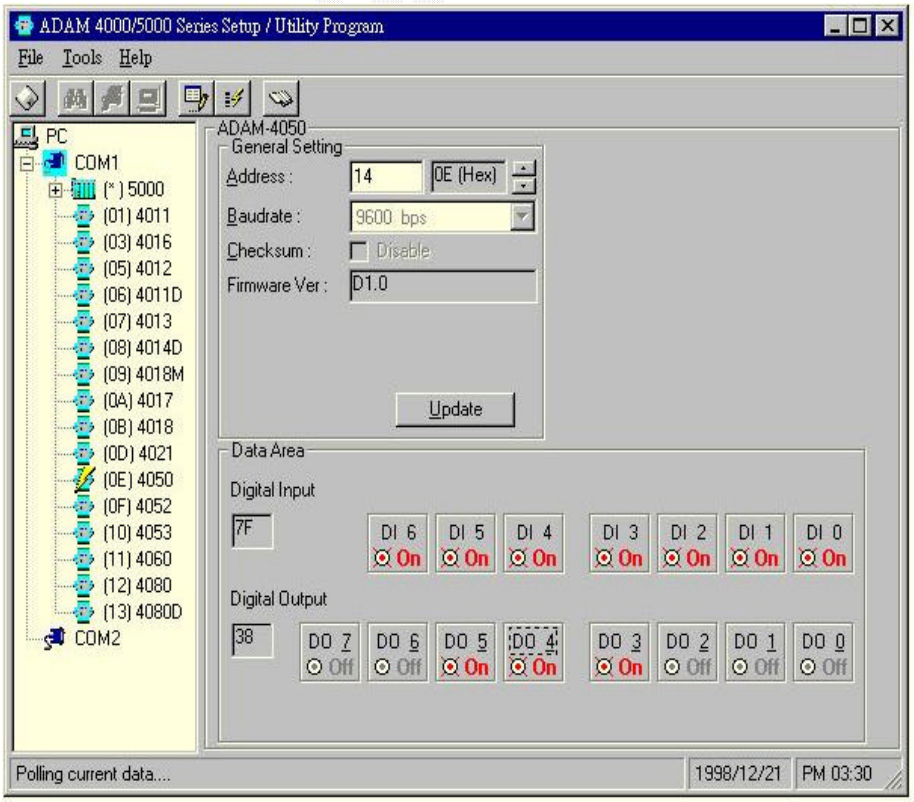


圖2-4　ADAM Utility 控制4050模組畫面

2.2 RS232

Serial Communication Port在控制方面的範疇一直存在，目前所使用的標準為RS232，RS為Recommend Standard的縮寫，是由電子工業協會 (Electronic Industries Association，EIA) 所制定的傳輸標準介面. 這也是許多個人電腦上的通訊介面之一, 通常 RS-232 介面以九個接腳 (DB-9) 或是二十五個接腳 (DB-25) 的型態出現，目前幾乎都為九個接腳的介面，因為實作簡單且價格便宜，所以被廣泛使用，一般個人電腦上會有兩組 RS-232 介面，分別稱為 COM1 和 COM2。下表為DB-9各腳位說明:

表 2-1 RS232 DB-9 腳位說明

腳位	意義
Pin1	偵測載波否
Pin2	接收訊號
Pin3	傳送訊號
Pin4	電腦已可接收資料
Pin5	接地
Pin6	儀器已可傳送資料
Pin7	請續傳資料
Pin8	清除暫存區
Pin9	鈴聲偵測

由上知資料傳送由 pin3 負責，而資料接收則由 pin2 負責，至於其他的 pin 主要是協助資料讓傳遞通順及正確用。

2.3 無線網路 802.11

2.3.1 無線區域網路的簡介

所謂的無線區域網路(Wireless Local Area Network/Wireless LAN)，即你的電腦透過無線網路卡(Wireless Card)，結合無線寬頻數據機〈Access Point，以下簡稱 AP〉進行區域無線網路連結；若再透過外部接取線路(如 ADSL、專線)即可暢遊網路世界運用無線網路資源。

同時無線區域網路與一般傳統的乙太網路(Ethernet)的概念並沒有多大的差異，只是無線區域網路將用戶端接取網路的線路傳輸部分轉變成無線傳輸之形式，但是卻具備有線網路缺乏的行動性，然而之所以稱其是區域網路，則是因為會受到無線寬頻數據機與電腦之間距離的遠近限制而影響傳輸範圍，所以必須要在區域範圍之內才可以連上網路。

而其區域範圍通常在 100 公尺左右，適合用於單一建築或辦公室內。需要使用無線區域網路的場合主要包括：

1. 不方便架設有線網路的環境
2. 使用者時常需要移動位置
3. 臨時性的網路

在實用上，通常會將無線區域網路和現有的有線區域網路結合，不但增加原本網路的使用彈性，也可擴大無線網路的使用範圍。目前最熱門的無線區域網路技術就是 IEEE(Institute of Electrical and Electronic Engineers,電機電子工程師協會)的 802.11 及其相關標準。

2.3.2 無線區域網路技術的比較

目前有兩種較普遍的無線區域網路解決方案。這些方案分別是 IEEE 802.11 標準 (主要為 802.11b)以及由 HomeRF 工作群組所推出的解決方案。這兩種解決方案無法相互跨平台作業，也無法與其他

無線區域網路解決方案作業。HomeRF 是專為家用環境設計，而 802.11b 則是為家用、中小型企業、大型企業，以及不斷增加的公用無線網路作用點所設計與部署。許多主要的筆記型電腦廠商，皆已計劃或決定在新的筆記型電腦內建 802.11b NIC。這兩種解決方案的比較如下：

表 2-2 802.11b 與 HomeRF 比較

	IEEE 802.11b	HomeRF
主要企業支援	Cisco, Lucent, 3Com WECA	Apple, Compaq, HomeRF Working Group
狀態	交運	交運 (低速)
範圍	50-300 英尺	150 英尺
速度	11 Mbps	1, 2, 10 Mbps
使用	家用、小型辦公室、校區、企業	家用
成本	每卡 $75-$150	$85-$129
安全性	WEP/802.1x	NWID/加密
廠商	超過 75	30 以下
公共存取點	超過 350	無
無線 NIC 的市場佔有率	72%	21%

而 Microsoft 將 802.11 視為最有前途，也最為健全的解決方案，可於各種環境中使用。基於對兩種方案的比較及使用的環境，因此在兩種無線區域網路解決方案我們採用的是 IEEE 802.11 標準中的 IEEE 802.11b。

2.3.3 無線區域網路拓撲

無線區域網路使用兩種基本拓撲建立而成，分別為 ad-hoc 和基礎架構(Infrastructure)。

2.3.3.1 ad-hoc

ad-hoc 拓撲為無線裝置自行建立的 LAN，其中並無中央控制器或存取點。 各裝置可在網路中直接相互通訊，而非透過中央控制器進行。這對於可能聚集少量電腦群組，且不需要存取其他網路的地方十分好用。例如，沒有接線網路的家庭，或者是各小組定期交流意見的會議室，都是 ad-hoc 網路最容易發揮效用的例子。且這些 ad-hoc 無線網路在與目前最新的智慧型點對點軟體與解決方案結合時，可以利用無線的方式，讓外出的使用者共同合作，進行多重玩家遊戲，傳送檔案或使用 PC 或智慧型裝置與其他人進行通訊。下圖為 ad-hoc 模式網路。

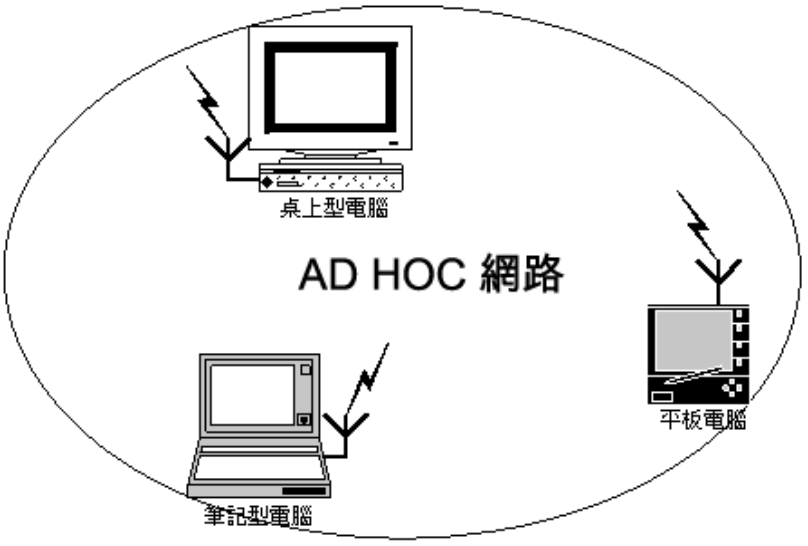


圖 2-5 ad-hoc 模式網路

ad-hoc 模式特色：

一、不需要 Access Point：在 ad-hoc 模式之下不需要用到 Access Point，如此不但節省成本也簡化了網路的架構。若是在使用的網路中需要用到 Access Point 那就不能使用 ad-hoc 模式，而必須使用基礎架構模式了。

二、結構簡單彈延展性小：由於 ad-hoc 模式的架構十分簡單，要架設一個 ad-hoc 模式網路不但十分迅速，也不需費心構思網路架構，做先期環境測試，同時也沒有太複雜的軟體設定手續。不過 ad-hoc 模式無法連結有線網路，也就限制了其延展性，對於一些要連結網路印表機、ADSL 數據機的場合就無法使用 ad-hoc 模式來架構網路而必須使用基礎架構模式。

三、點對點通訊但無轉送功能：在 ad-hoc 模式之下，每一個電腦皆進行點對點通訊。資料由傳送端送出後便直接送入接收端而無法藉由其他的電腦來轉送資料。

2.3.3.2 基礎架構(Infrastructure)

基礎架構(Infrastructure)拓撲可提供基本工作站 (稱為存取點)，將現有有線區域網路延伸為無線裝置。存取點可連接無線和有線區域網路，並作為無線區域網路的中央控制器。存取點可於特定範圍內，協調多重無線裝置的傳輸與接收；而裝置的範圍和數量則依使用的無線標準及廠商的產品而定。在基礎架構模式下，可能有多重存取點可涵蓋大量範圍，或者涵蓋如單戶家用或小型大樓等小區域的單一存取點。 下圖為基礎架構模式網路：

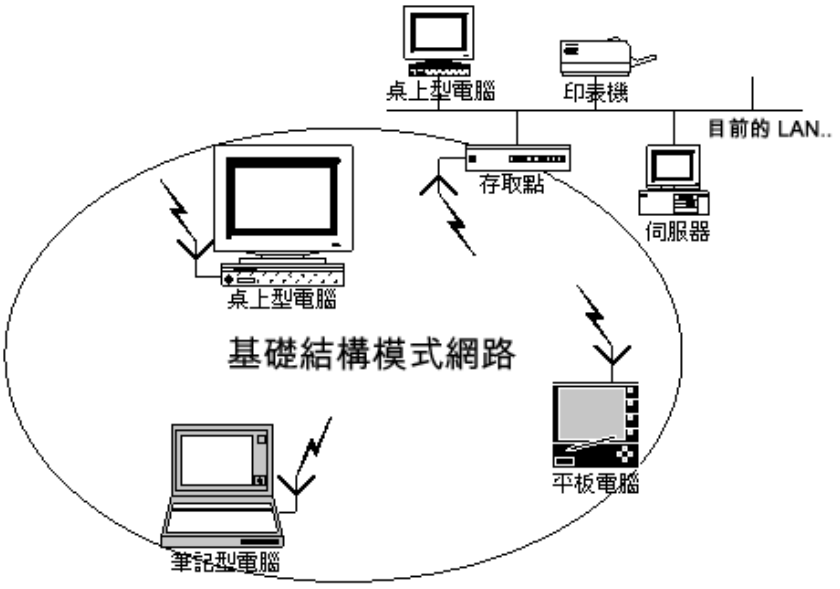


圖 2-6　基礎架構模式網路

基礎架構與 ad-hoc 最大的不同是基礎架構模式網路多了 Access Point。多了 Access Point 後 ad-hoc 模式與最大的差異便是能夠和有線網路先連結。所有的電腦要收發訊號都需要經由 Access Point 來進行，就算兩台電腦距離太遠，只要是在 Access Point 的訊號收發範圍之內，就可以經由 Access Point 來轉送訊號。更甚者，若處於兩台不同 Access Point 的範圍之內，Access Point 也能夠透過有線網路來將訊號傳輸給目標電腦所在的 Access Point，以達到傳輸訊息的目的。由於這種功能使得基礎架構模式比 ad-hoc 模式有著更大的訊號傳遞範圍。

2.3.4　無線網路的優點

一、減少網路架構成本：只要 Access Point 架設完成，每個使用者只要準備一張無線網路卡便能夠連上網路，不需要再從 HUB 或 SWITCH 拉網路線到每個使用者，因此也就不會破壞裝潢隔間。同時就連隔著馬路的兩棟大樓都可以使用無線網路搭配定向天線來進行網路的連結。

二、易於維護：有線網路其一大弱點-網路線。網路線最怕因為各種環境意外因素而被破壞，且網路線通常裝設於隱密的路徑，一但網路線出了問題需一段一段檢查哪裡出了問題，在考量要不要重新拉

線，因此對使用者而言十分不便。而無線網路因為不需要使用網路線，一但網路出現了問題只需要檢查軟體設定或硬體方面哪裡有問題即可。

三、易於擴充：若公司人數擴增，電腦設備也隨之增加。在傳統有線網路中，辦公室內的網路節點數量在舖設網路線時便已經決定了，若已經飽和，便無多餘的結短供新設備使用。而為了讓新電腦加入區域網路，一般的做法便是添購新的 HUB 或 SWITCH，如此一來便增加了成本的開銷。而對於無線網路而言，只要是在寬頻容忍範圍內把無線網卡插上即可，不會有網路節點不足的問題。

四、電腦可四處移動：有線網路的另一大弱點-線的牽絆。筆記型電腦或 PDA 如要上網便只能夠在線的長度範圍內移動。但是對於無線網路而言，只要是在訊號可及範圍內(從小型室內天線的數十公尺到大型室外天線的數百公尺)都可以隨意移動。

2.3.5 無線網路的缺點

一、障礙物的影響：由於無線網路是用電波傳送訊號，因此傳送端與接收端之間若有厚實的障礙物阻隔便會影響其電波傳遞，甚至造成電波無法通過。如同一層樓的牆壁、上下樓層的樓板。必要時便需在訊號無法到達的地方增設 Access Point 或訊號指向器。

二、周圍電波干擾：如同行動電話一般，無線網路同樣也會受到附近使用相同或相近頻道干擾，造成網路斷斷續續。且電器用品的電磁波若剛好落在 2.4GHz
(如：電磁爐、微波爐)也會干擾到無線網路的使用品質，甚至完全斷訊。

三、資料安全的問題：傳統的有線網路是在實體的網路線上傳遞訊息，駭客如要竊取資料便需要接上 HUB 才行。但對於無線網路而言，訊號的傳遞是在大氣中進行，所以駭客只需在 Access Point 附近擺台有無線網路的電腦，即可以擷取到大量的訊息封包。

四、漫遊問題：當電腦遠離原來的 Access Point，會因為訊號的

衰減而造成網路中斷，此時附近若有其他的基地台存在，理論上會經由 Handoff 的機制，由其他的基地台來繼續為你服務，以避免斷線，此行為就是無線網路上的漫遊。但漫游仍有其限制存在，只能在同一 IP 子網路中作用，若電腦移動到另一 IP 子網路(跨過 router)，就無法進行漫遊，網路便會中斷，直到取得下一個子網路中合法的 IP 為止。

2.4 TCP 與 UDP 傳輸協定

傳送層的功能

在 OSI 和 TCP/IP 分層協定中網際網路層協定只提供路由資訊的判斷，以確定封包的傳送路徑。但事實上 IP 協定只確保封包交換設備之間的傳輸，並沒有提供一套機制來確保數據的傳輸。在低層的通訊裡，封包可能在傳送過程中發生錯誤，諸如網路硬體的損壞、網路負荷過重等等，導致封包被丟棄或損壞。由於封包路由的多樣性和複雜性，以及影響路由因素眾多及其不可預測性，封包之抵達常是不依序的，或是會發生重複傳送的情形。因此，必須提供一套網路技術，以達成更可靠和有效的傳送。

另外由於 IP 封包的體積是有限的，然而網路程式之間交換的數據往往會超過這個體積限制；那麼，我們必須有另一套機制將程式送來的資料進行規劃，以符合 IP 封包的傳送要求。在高層的程式裡，除非利用非可靠和非連線型的資料傳送方式，否則，程式設計者必須對每一個一個應用程式處理偵錯和修復的動作，這無疑增加了程式設計和修改的難度，而且也做成許多重複的處理動作。因此，我們也有必要找出一個可靠的資料流傳送方法，以建立單獨且適用於所有應用程式的資料傳送協定。這樣就可以將應用程式與網路內部協定隔離，同時提供一致的資料流傳送界面。傳送層的設計可以說是應上述要求而生的，它的主要功能有：

- 接管由上層協定傳來的資料，並以 IP 封包可以接受的格式進行“封裝”工作。
- 進行資料傳送和回應的確認，以及處理資料流的檢測和控制。
- 對不同的連線進行追蹤及轉換。

在 TCP/IP 協定組中，關於傳送層的協定就是 TCP 和 UDP 了，下面為其相關的介紹。

2.4.1 TCP 傳輸協定

TCP (傳輸控制協定;Transmission Control Protocol，簡稱 TCP) 的主要任務是負責發送端及收受端的協定建立，建立可靠的資料流傳送服務。它使用 IP 來傳送封包給上一層的應用程式，並保證資料在網路上的流動安全可靠。簡言之，TCP 具有下列幾個主要的功能：

- 循序編號(Sequence Number)：TCP 為每一個封包建立編號，使封包就算不能按照原來的發送順序抵達收受端，也可依此編號正確重組。
- 確認(Acknowledgement)：接收端針對發送端所傳來的每一封包，回送「我已收到」的確認封包，類似郵政掛號中「回執」的概念。
- 檢查(Checksum)：TCP 在每個封包的表頭中加上一個檢查欄位，以確認其是否為欲傳送的原始封包。如果封包到了接收端卻發現檢查值不合，即表示封包發展了錯誤或損毀，因此接收端就無法發出確認的封包。
- 重送(Retransmission)：發送端如果在某一預定的時間內沒有收到該確認封包，就會認定封包傳送失敗，於是重送該封包，直到收到該封包抵達收受端確認訊息為止，下圖為封包重送圖。

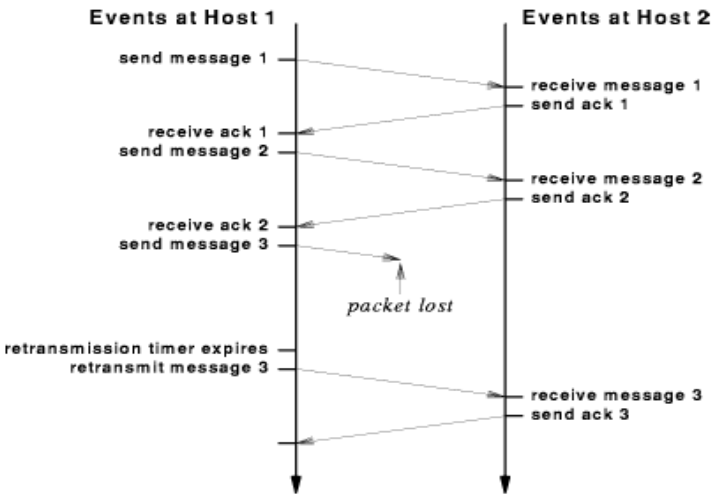


圖 2-7　封包重送(Retransmission)

在使用 TCP 進行資料傳輸時，必須先建立起兩者之間的連線關係。TCP 連線的建立是透過帶有連線控制訊息的封包在兩端主機間傳遞，再藉由 TCP 表頭中的循序編號和錯誤檢查值的檢查正確無誤，經一番交談後，雙方乃同意進入連線狀態。經由此連線請求、連線確認、連線成功的程序，便形成了三向式的握手協定(3-way handshaking)如圖 2-8，而斷線時也是採用相似的程序：斷線請求、斷線確認、斷線成功，如圖 2-8。由上述的 TCP 功能可得以下特性：

- 連接導向(Connection-Oriented)：在資料發送時，兩端會建立起虛擬電路，讓資料能有跡可循地往下傳送。所以，TCP 是屬於連接導向的，傳輸的雙方必須先做溝通，確認連線建立後才可傳送資料。
- 可靠性(Reliable)：TCP 以確認、重送、檢查三個觀念來完成可靠性的資料傳輸。它利用 TCP 封包表頭內的某些欄位來控制資料確實地傳送到對方，而在資料遺失時，TCP 則會要求發送端重新傳送。
- 全雙工式通訊(Full Duplex)：封包一旦抵達正確的 IP 位址，TCP 就開始在發送端和接收端的電腦上，為正被傳送的資料建立起對話(Dialog)，而兩端可以分別進行資料的收發。由於此全雙工的特性，使得確認回執的工作可以輕易地達成。
- 資料流(Stream)：TCP 是以資料流型的傳輸型態來傳送資料，也就是說，資料像流水般有次序地從本端主機流向遠方主機，而遠端主機則依序自資料流中讀取資料，將它傳給應用程式去處理。每個 TCP 封包可攜帶一長串的資料，而不是一個位元、一個位元地傳送。

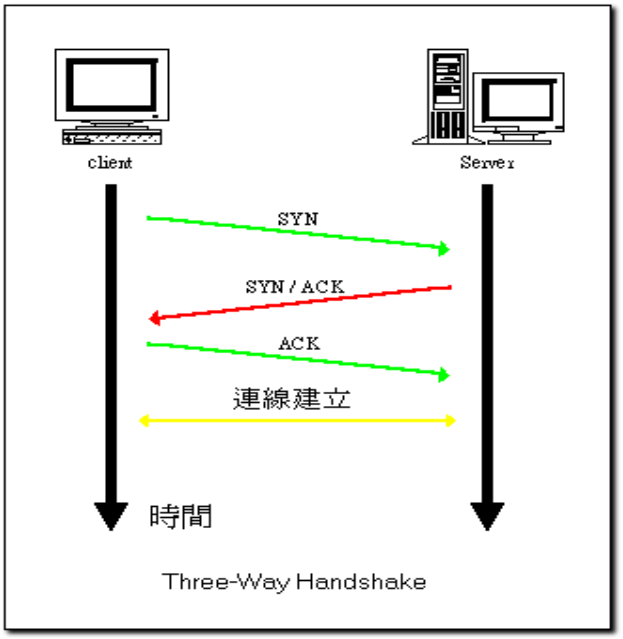


圖 2-8 TCP 連線之建立過程(Three-Way Handshake)

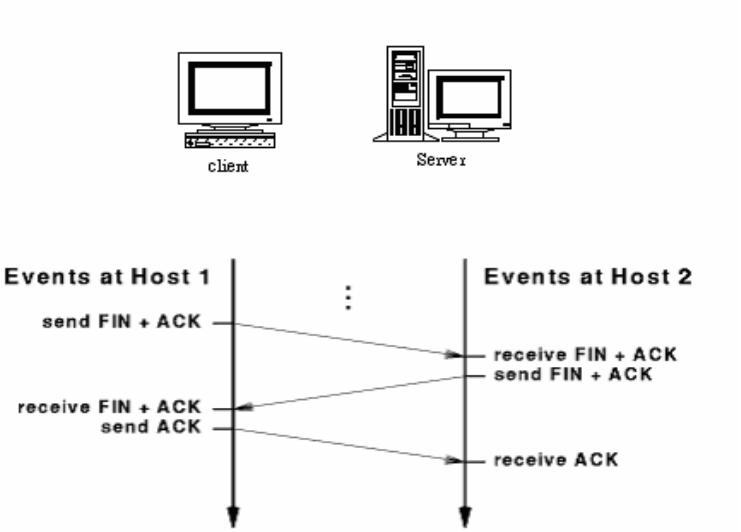


圖 2-9 TCP 中斷之連線之步驟(Three-Way Handshake)

所以有了 TCP，資料傳輸的可靠性由它來處理即可，上層的應用程式只要把要傳送的資料透過介面丟給 TCP 匯流排，而不必煩惱資料如何正確無誤地傳送到對方的手裡。應用程式當然還是可以直接利用 IP 在自己的程式裡控制資料傳輸的可靠性，但這樣做會造成上層應

用程式的負擔，程式也較不容易維護。基於網路上分層處理及程式設計模組化的觀念，最好還是透過 TCP 來做可靠性的資料傳輸。

2.4.2 UDP 傳輸協定

UDP(使用者資料元協定；User Datagram Protocol，簡稱 UDP)提供的是一個非可靠的非連線型(connectionless)的資料流傳送服務，在運作時採用了一個重要的概念—通訊端點位址(Port)。雖然 IP 是識別不同主機的唯一標記，但我們若僅使用一個 IP 位址，在某些環境下並不足以明確地指定資料接收端的身分，這是因為每個主機可能有多個使用者同時使用，而每個使用者又可能會同時產生好幾個作業程序(Process)。因此，如果我們能在每一使用者啟動一個網路應用軟體時便指定一個 Port 號碼給它(通常是一個 2 位元整數)，如此就可以真正明確地區分發送端和接收端的角色了。因此，「IP 位址+port 號碼」便是網路上一個通訊端點的明確定義。

但是，IP 中並沒有定義 Port 的欄位，為了使資料的傳輸能有 Port 的概念，TCP/IP 便在 IP 層上面架構了 UDP，IP 負責將封包經由網際網路傳至對方主機，UDP 則將收到的封包分發給不同的應用程式或作業程序。UDP 不提供錯誤檢查，也不執行封包的排序，亦不會在資料發生錯誤時重新傳送資料。由上述可知，UDP 具有無連接導向(Connectless)及不可信賴性(Unreliable)的特點。

2.4.3 Socket Pair

所謂的一個網路連線，事實上就是兩台機器之間的兩個程式之間的連線。我們可以根據 IP 來區別主機、根據埠口(port)來區別程式。在 TCP/IP 連線中，這是非常重要的概念，也就是所謂的 Socket。

一個 Socket 就是由一個 IP 與一個 Port 來定義的，您可將之視為程式與 TCP/IP 連線之間的界面。準確來說，一個連線封包必須有四個元素，也就是所謂的 Socket Pair：

- 來源位址(Source Address)
- 來源埠口(Source Port)
- 目的位址(Destination Address)

- 目的埠口(Destination Port)

任何一個 TCP/IP 封包都肯定帶有這對 Socket 的資訊，缺一不可。然而，來源 Socket 與目的 Socket 卻是相對而言的，若離開了封包本身的具體連線方向，是沒辦法區分來源與目的的。因為連線是雙向的緣故，若封包從客戶端送往伺服器端，那麼：客戶端為來源、伺服器端為目的；若是從伺服器端送往客戶端，則剛好相反。網際網路層依靠位址資訊將封包送抵目的地、處理完畢後將封包交由傳送層處理、然後傳送層則依據埠口值、將封包交由相對應的程式處理、至於程式如何處理封包信息,那就是應用層所要關心的問題，程式不必理會底層的封包是如何傳送的,只要程式能開啟一個 socket ,並能對之進行讀或寫的動作,就可以與另一方的程式溝通了。至於 socket 的建立與維護,則交給傳送層協定負責。

2.4.4 TCP 與 UDP 的比較

TCP 與 UDP 主要的差異在於是否提供可靠性傳輸。其真正目的是為上層應用程式提供不同的傳輸選擇：

表 2-3　　TCP 與 UDP 的比較

協定	TCP	UDP
特色	● 連線導向傳輸協定 ● 在資料傳送的過程中能進行錯誤偵測 ● 確認資料傳送的正確性與可靠性 ● 使用滑動視窗 (Sliding Window)進行流程控制協調通訊雙方之傳送與接收	● 非連線導向傳輸協定 ● 會不斷地將資料快速送出，直到資料送完為止 ● 缺乏強大的錯誤偵測與確認機制 ● 資料傳送的可靠性低 ● 較適合單純的查詢或廣播訊息傳送

優點	傳送可靠，程式可省略可靠機制	傳輸量大，迅速
缺點	速度比較慢	不可靠，程式或需自行提供可靠機制

在此我們可以將 UDP 比喻成寫信：若是寄平信，我們並不能確定這封信一定能送到對方的手中，因為信可能會在半途被弄丟或截走；而 TCP 則可以比喻為打電話，必須先經對方的同意(對方必須在家而且肯接電話)才有可能開始通訊，在通話期間，如果對方講話太小聲以至於無法聽清楚，就隨時可以要求對方重講直到聽清楚為止。
因此 TCP 相當於是 UDP 的功能再加上可靠性傳輸的控制。因為 TCP 需要有控制傳輸可靠性的額外負擔，因此對於某些較不重要的資料傳輸(如 E-MAIL)，一般利用 UDP 來做就可以了，而如 FTP、TELNET 等對傳輸內容要求絕對正確的應用層協定，則非用 TCP 來做不可，故此在本專題中，我們採用了以 TCP 的方式來傳送我們的資料封包。

2.5 JAVA 技術運用

2.5.1 JAVA簡介

JAVA系統通常由幾個部分所組成：環境、語言、JAVA API、還有各種的類別庫。一個典型的JAVA開發環境可由下圖來解釋：

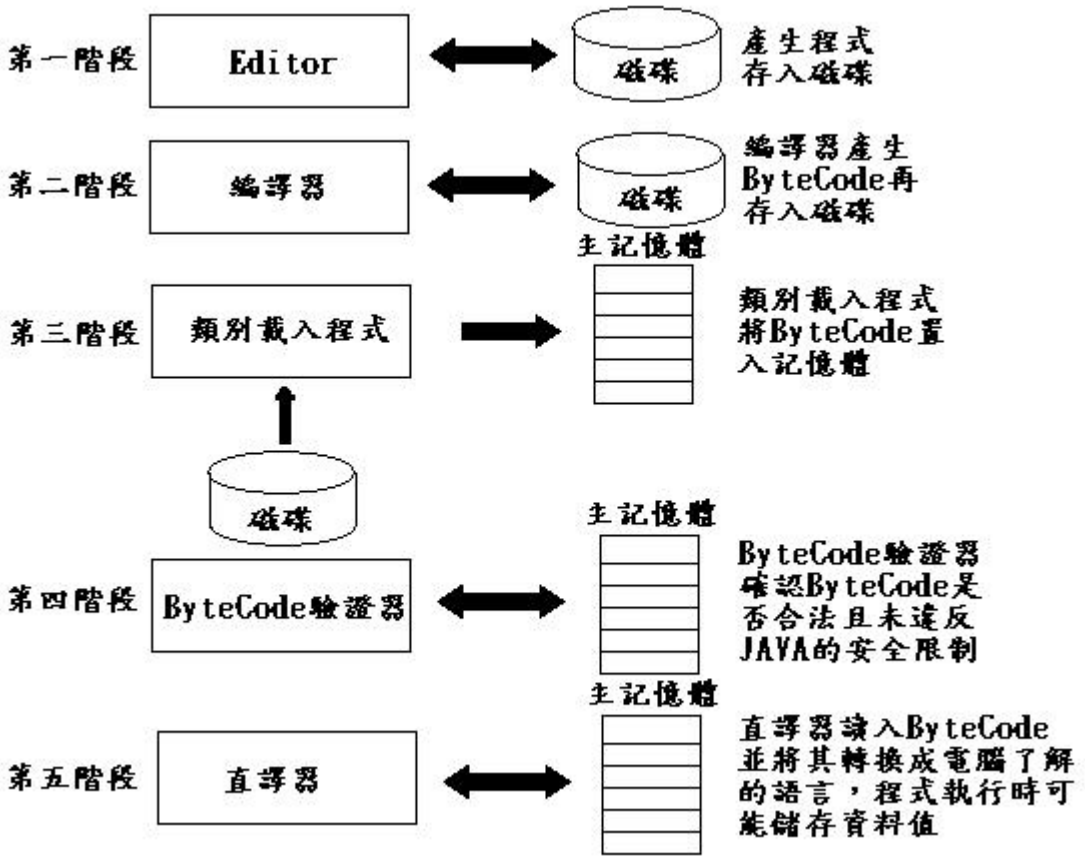


圖2-10 JAVA開發環境

由於JAVA可以在任何有JVM的機器上執行，具有write oncr，run anywhere的特徵，近年來越來越受到重視，目前是屬於非常熱門的語言之一。

2.5.2 Java Communications API

Java communication API是SUN公司所出的JAVA延伸套件，其具有獨立於平台的通訊應用功能，支援RS-232及IEEE1284 parallel port，有以下功能：

一、 列舉出系統所有可以使用的通訊埠
二、 可對已被使用的埠作出要求
三、 解決多個應用程式同時使用通訊埠的問題
四、 可執行同步和非同步的I/O
五、 Receive Beans-style events describing communication port state changes

2.5.3 JDBC(Java DataBase Conectivity)介紹

Java 技術中資料庫的部分，是一套存取資料庫的 API，由一堆 java.sql package 中的類別所組成，用來讓 Java 程式也能和資料庫溝通。現今大多數的資料庫都支援 SQL（結構化查詢語言，Structured Query Language)。透過 SQL ，可以用特定條件查詢資料庫，並取得查詢結果。SQL 是一個標準化的語言，大多數資料庫廠商對 SQL 做了擴充。利用 JDBC 驅動程式連結，你就可以利用 SQL 語法來存取資料庫中的資料。下圖為 JDBC API 的流程圖和概觀圖

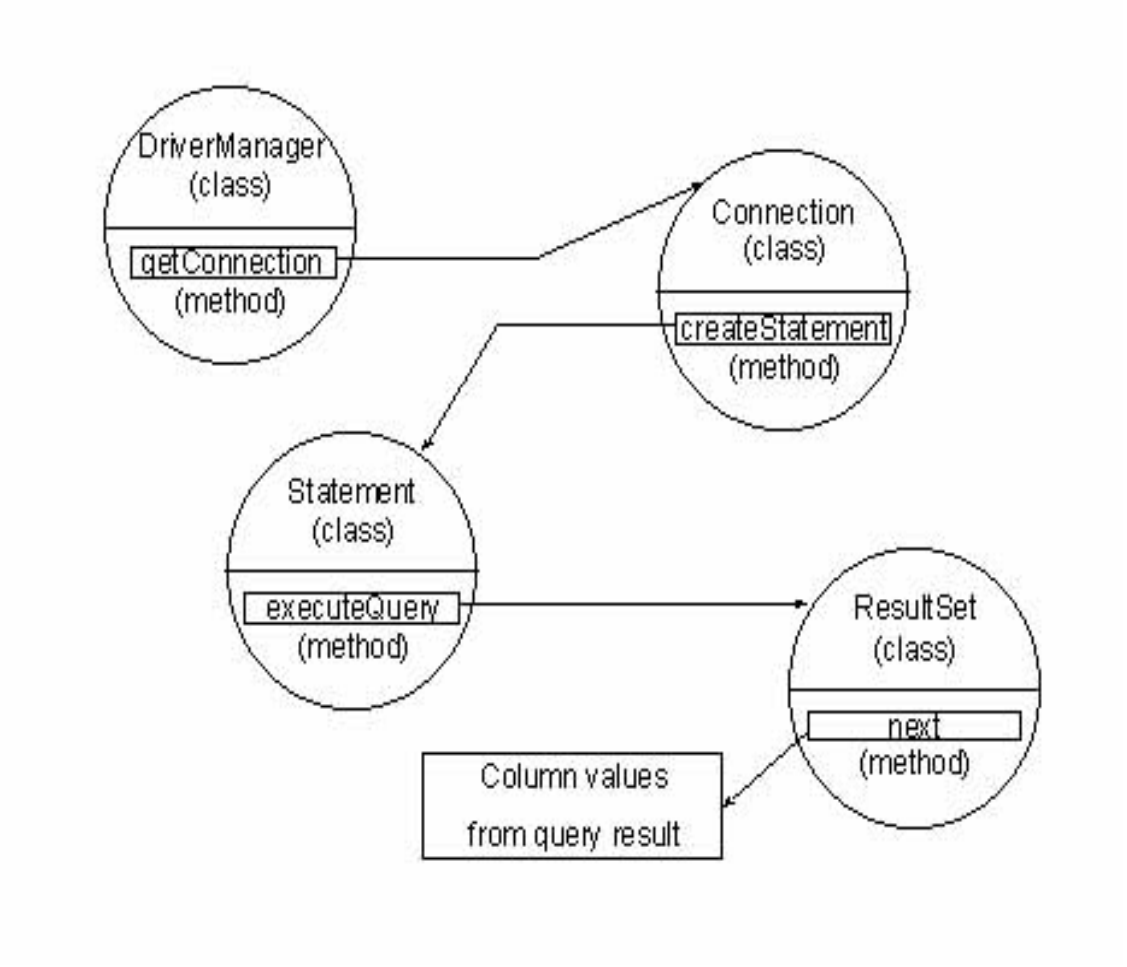


圖 2-11　JDBC API 的流程圖

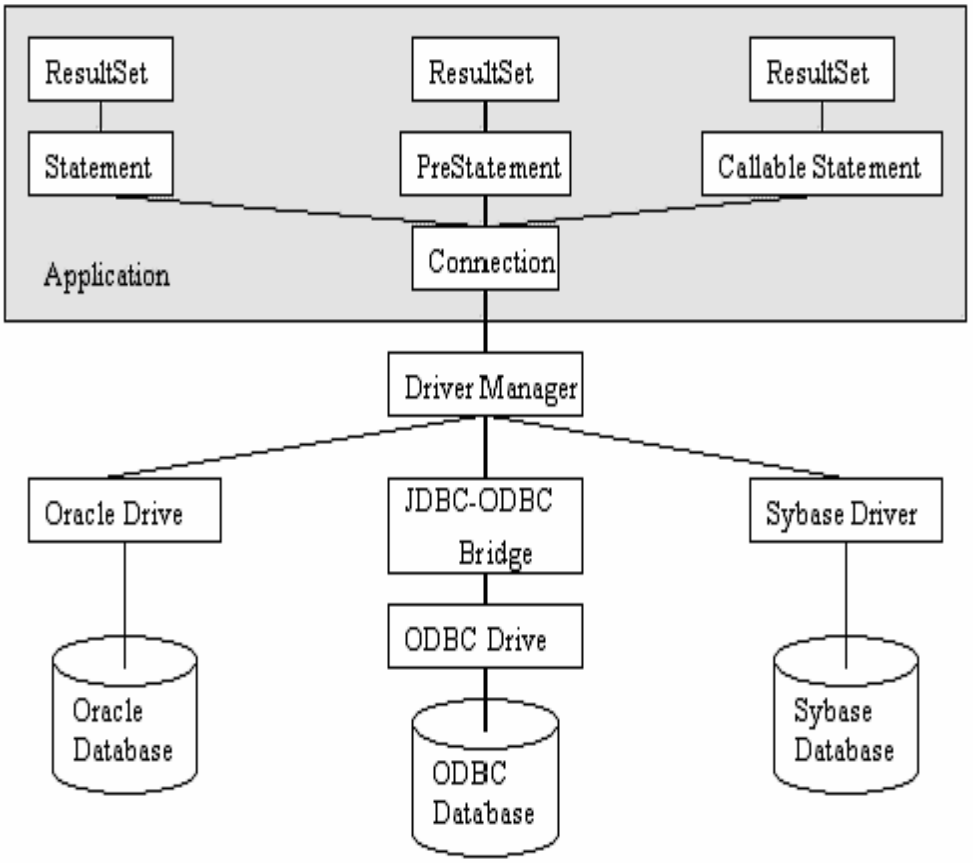


圖 2-12　JDBC 的概觀圖

JDBC 的概觀圖中分為上下兩個部份，上半部為應用層，下半部為驅動程式層：

1. 應用層

應用層的 API 是給程式設計師使用的，只要程式設計師知道 SQL 語法就可以使用這些 API。

2. 驅動程式層

驅動程式層是提供給發展驅動程式廠商的 API，所以要透過應用層的 API 存取資料庫之前，還必需要有廠商提供的 JDBC 驅動程式。

JDBC driver 的使用可分成下列步驟

1. **載入驅動程式**
2. **建立 connection**
3. **準備 statement**
4. **執行 statement**
5. **取回結果**
6. **close statement**
7. **close connection**

2.6 資料庫-MySQL 介紹

2.6.1 何謂資料庫

資料庫即一組相關的資料，以及這些資料間的關係(relationship)之集合，而建立資料庫之好處在於相關而分散在各處的資料可集中管理與分享。

資料庫管理系統(Database Management System，DBMS)：由一組軟體所組成，負責管理及維護資料庫內之資料而其特性如下：

優點：

- **資料集中控管，對於資料之安全管制、一致性、完整性可以有較佳之管理。**
- **由於相關資料均集中在一起，故資料共享的可能性提高。**
- **資料與應用程式有了區隔，應用程式不必擔心資料結構改變後，程式無法存取改變後的資料，也就是說，DBMS 是程式與資料之間的溝通橋樑。**
- **透過 DBMS 內提供的一些工具，一般使用者也可以自己進行簡單的分析工作。**
- **由於資料集中管理，因此可以降低資料的重複性。**

缺點：

- **軟硬體花費增加。**
- **要購買軟體以及硬體來放置資料。**
- **系統複雜度增加**
- **DBMS 屬於系統軟體，要管理 DBMS 需要一定程度的專業知識，因此也提高了系統管理的複雜程度。**
- **集中化控管的風險**
- **資料集中管理可能遭遇到系統損毀造成資料流失，或是資料庫管理員管理不當造成駭客入侵，使得資料外流等風險。**

2.6.2 MySQL 簡介

MySQL 是一個快速、多執行緒（multithread)、多使用者且功能強大的關聯式資料庫管理系統（relational database management system，RDBMS)，可以與 C、C++、Java、Perl、PHP 等語言很容易的連結，可以運行於多種平台上，例如：Solaris、RedHat、Linux、FreeBSD、OS/2、Windows ...等等

2.6.3 MySQL 一些重要的特徵

- **可運行在不同的平台上**
- **在查詢的 SELECT 和 WHERE 部分支援全部運算符和函數**
- **通過一個高度最佳化的類庫實現 SQL 函數庫**
- **一個非常靈活且安全的權限和密碼系統，並且它允許基於主機的認証密碼是安全的，因為當與一個伺服器連接時，所有的密碼傳送被加密**
- **沒有內存漏洞。用一個商用內存漏洞監測程式測試過 (purify)**

目前 Java 幾乎可和現面上的資料庫連結，而我們選擇 MySQL 來當我們專題開發的資料庫的原因大概為以下幾點：

- 在非營利目的下為免費的，且取得方便
- 可以在多種平台上使用，可以配合 Java 跨平台的特性
- MySQL 容量雖小，但其功能強且效能佳

2.7 PJEE〈PersonalJava Emulation Environment〉介紹

Personal Java的發展其實已經一段時間了，許多公司根據其規格生產實作產品，而它所扮演的佼色也在J2ME推出之後更形尷尬，所以接下來將個別針對Personal Java作說明。〈以下將以PJava替代Personal Java〉

PJava的規格其實沒有定義在CLDC或CDC下，雖然最後將會被歸到CDC的Personal Profile之中，但是目前其規格還在演進。其實PJava是從Java1.1之中所分之出來，但是並非Java1.1的全部規格都包含進來。PJava特別適合使用在具有豐富圖形顯示能力的消費性電子產品上面，於是我們可以發現昇陽的網站上對於PJava的參考實作是建立在PocketPC上。

目前，使用PJava最多的平台為PocketPC與Symbian OS。就如同開發其他PDA程式一樣，我們不一定要購買一台PDA來作測試，昇陽網站上也提供了PJava的模擬器，讓我們可以在Windows或Solaris作業系統之下測試開發給PJava環境執行的應用程式。此工具就是PJEE。使用PJEE所開發出來的PJava程式可以保證能在PocketPC與Symbian OS上順利執行，目前最新版本為3.1版。

第三章　密碼學及 DES 演算法介紹

對他人隱私的好奇心和隱藏訊息是所有人類社會的特性。隨著運算能力的出現和進階數學技術的發展，系統變得非常複雜，目前有可能建構出很難被有效破解的密碼。用來加密和解密資料的 演算法分為兩大類別：秘鑰（secret key）或對稱密碼學，其中對加密和解密這兩個過程使用同一金鑰；以及公開金鑰或非對稱密碼學，其中一個金鑰用來加密，另一個用來解密。在這個專題中，我們所使用的是 DES 加解密演算法，因此在密碼學簡略介紹後，會特別說明 DES 演算法。

由於目前網路的蓬勃發展加上無線網路的崛起，資料在傳送過程中的安全性更形重要，下圖為資料傳送過程中可能發生之現象。

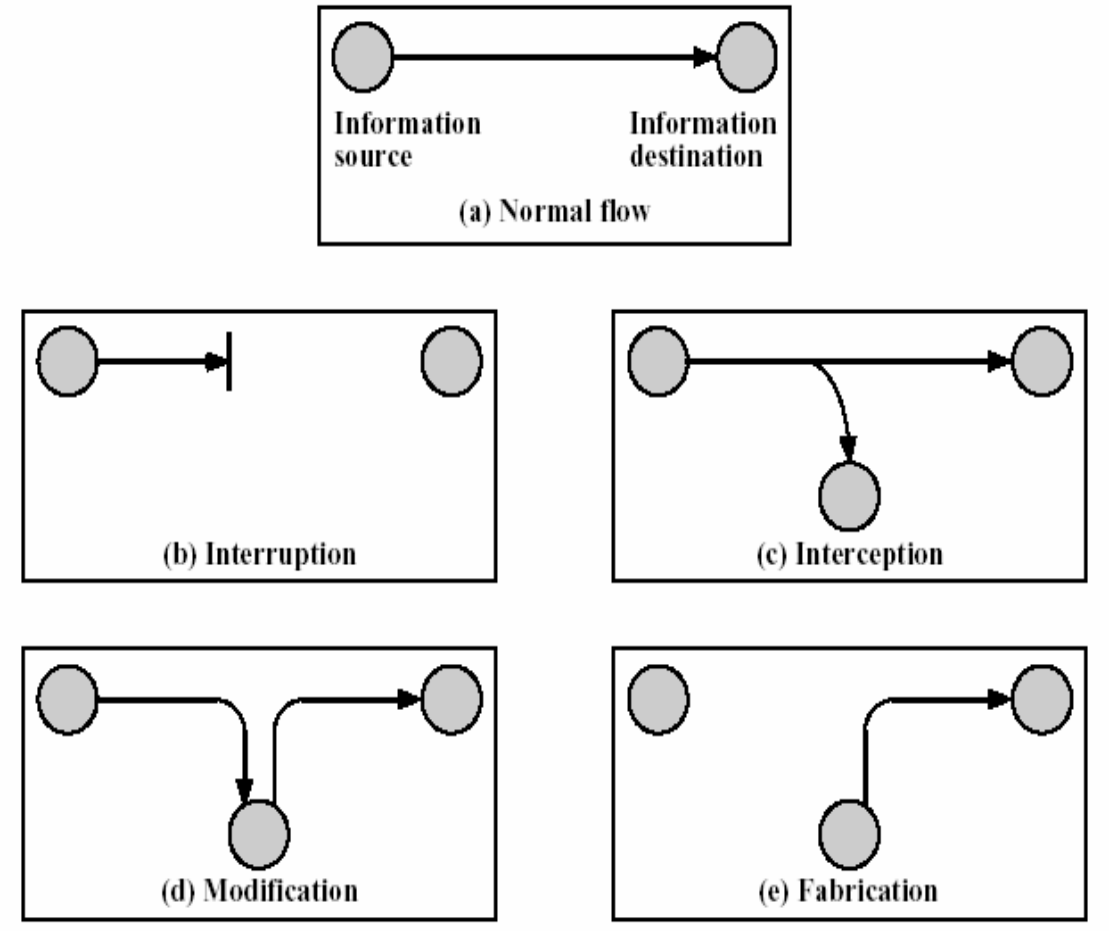


圖 3-1　　資料傳送過程可能發生之現象

為了維護所傳送資料的安全，人們便想出許多的方法來隱藏訊息。一個密碼系統的主要角色有三個，即發送方、接收方、與破密者，典型的密碼系統如下圖 3-15 所示。在發送方，首先將明文利用加密器及加密金鑰，將明文加密成密文。接著利用公眾通道送給接收方，接收方收到密文後，利用解密器及解密金鑰，可將密文解密成明文。破密者並不知道解密金鑰，但利用各種方法想要得知明文，或假冒發送方送出偽造的訊息讓接收方信以為真。

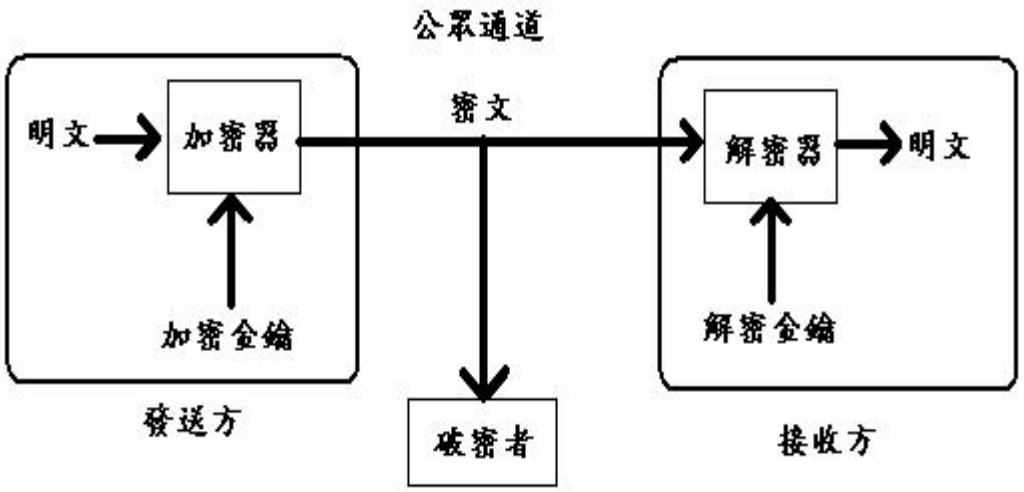


圖 3-2 典型密碼系統

一般依照其應用而言，密碼系統可提供下列功能：

1.秘密性(secret or privacy)：防止非法的接收者發現密文。

2.鑑定性(authenticity)：確定資訊來源的合法性，一及此資訊確由發送方所傳送。而非別人偽造，或利用以前訊息重送。

3.完整性(integrity)：確定資訊沒有被有意或無意的更改，及被部分取代。

4.不可否認性(nonrepudiation)：發送方在事後，不可否認其傳送過之資訊。

傳統密碼學往往僅注重資訊的秘密性，但是近代密碼學認為資訊的鑑定性、完整性、及不可否認性，在商業上應用比秘密性更高。

在圖 3-15 中，如果加密金鑰只有發送方知道，則稱此為秘密金鑰密碼系統，具有下列特性：知道加密金鑰即之解密金鑰，反之亦然。在許多情況下，兩者相同，因此又稱為對稱式金鑰密碼系統。一個安全的祕密金鑰密碼系統可以達到保護資訊機密、鑑定發送方身分、確保資訊完整性。但其亦有缺點如下：

1.收發雙方如何獲得其加解密金鑰：這稱為金鑰分配問題，若收發雙方互不認識，此問題更加嚴重。如暫不考慮分配金鑰問題，可假設雙方有一安全通道，下圖 3-16 為一較完整秘密金鑰密碼系統。

2.金鑰數目太大：若網路中有 n 人，則每一人需擁有 n-1 把鑰匙，如何管理眾多鑰匙，也是大問題。

3.無法達到不可否認性：由於雙方之知道對方金鑰，因此發送方可以否認之前發送的任何資訊。

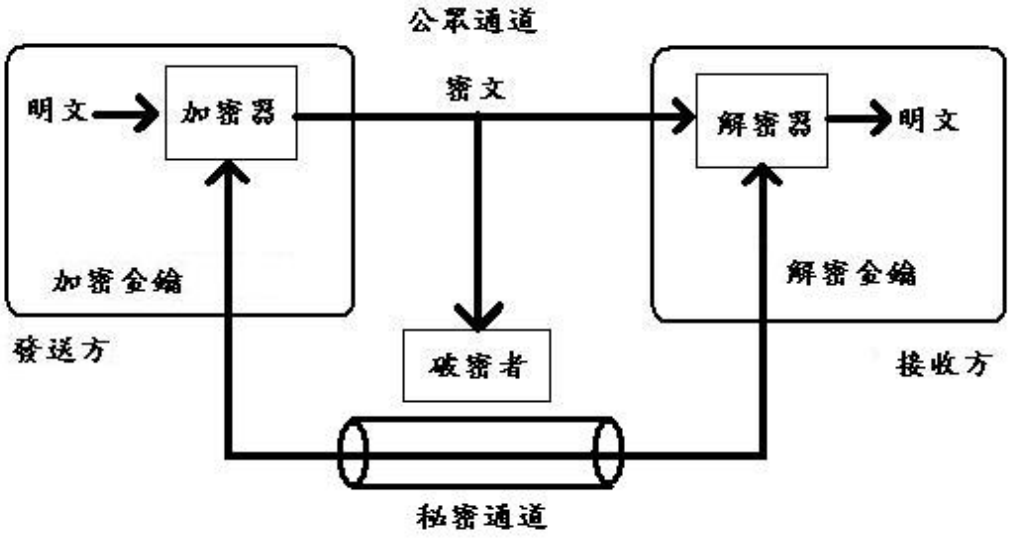


圖 3-3 秘密金鑰密碼系統

由於秘密金鑰密碼系統有上述問題，所以在 1976 年誕生了公開金鑰密碼系統。以一個日常生活經驗為例，當我們在上鎖時，是否一定要有開鎖的鑰匙？答案是否定的，我們並不常常需要或沒有鑰匙就能鎖上，將之對應到密碼系統上。發送方是否一定需要知道解密金鑰，才能將明文加密成密文，在秘密金鑰密碼系統中，知道加密金鑰

就能知道解密金鑰。那我們是否能將加密金鑰與解密金鑰分開，若有人知道加密金鑰還是無法得知解密金鑰，只有擁有解密金鑰的人才能解密，這就是公開金鑰密碼系統的主要精神，也因為兩種鑰匙是分開的，所以又稱為雙金鑰密碼系統，或非對稱密碼系統。

安全的公開金鑰可以達到以下功能：

1.保護資訊機密：任何人均可將明文加密成密文，但只有擁有解密金鑰的人，才能解密。

2.簡化金鑰分配及管理問題：網路上的人只需要一把加密金鑰(公開的)以及把解密金鑰(私有的)即可。所以擁有更高安全性，也簡化了金要的管理問題。

3.達到不可否認性功能：由於只有接收方才擁有解密金鑰，若他先用解密金鑰將明文加密成密文，則任何人均能用公開金鑰將密文解密成明文，由於只有接收方才能將明文加密，任何人均能驗證而無法偽造，因此可達到此功能。

公開金鑰密碼系統雖具有許多優點，但仍有其缺點存在。其中最受人詬病之處，在於加解密運算複雜，且速度緩慢。因此有人建議用公開金鑰系統達成數位簽署，及解決秘密金鑰密碼系統之金鑰分配問題。而以秘密金鑰密碼系統對明文加解密，達到到秘密性功能。此種密碼系統稱為混合型密碼系統。

由於所使用的 PDA 運算能力不強，所以我們使用較快速的 DES 加密而不是使用需要複雜運算的 RSA 演算法。

DES (Data Encryption Standard)為目前最常用的密碼演算法之一。他是由 IBM 公司在 1970 年代所發展出的加密演算法並在 1977 年經美國國家標準局（NBS）採用為聯邦標準（FIPS PUB 46-2），成為各界最為廣泛應用之對稱式金鑰密碼系統。截至目前為止，除了密碼金鑰較短為人詬病外，還無法根本完全的破解 DES，故有人提出以 Triple-DES 或 DESX 的方式，加強其金鑰長度，使成為安全性高之加密演算法。

DES 的金鑰長度為 56 個位元，有時會輸入 64 個位元，在其第 8、16、24、32、40、48、56 及 64 位元為同位元檢查碼，在做加密或解

密動作時，其同位元檢查碼沒有真正使用。DES 系統的基本原理是利用 Shannon 的多重加密的觀念（Product Cipher）並利用 Confusion（混淆）與 Diffusion（散佈）等方式，將明文轉換成其他格式，並散佈明文的每一個小部分擴散到密文的各部分以達到加密效果。簡單來說，資料保密技巧是將原始資料「明文(Plaintext)」弄得非常散亂，讓破解者無法利用統計方式或其他數學分析技巧將加密後的「密文(Ciphertext)」還原成原來的明文。

DES 的加密方法是透過 16 回合的運算所組成，其每一回合的運算目的，不外乎將上一回合所打散的明文在弄得更亂一些，也就是說每一次的運算相當於在明文中多加了一道鎖，因所透過 DES 的運算之後，其原始資料已被 16 把鎖給層層保護住了。多重加密的原理即當運算加鎖的次數越多次時，相對原始資料（明文）的安全性就越加提高。

DES 的加密方法是透過 16 回合的運算所組成，我們以下列數學式來表示 DES 的運算方式：
(Li,Ri)：表示第 i 回合之 64 位元輸入字串
每一回合基本的運算原理如下：
$L_{i+1} = R_i$
$R_{i+1} = L_i \oplus F(R_i, K_i)$，其中 $i=0,1,2, \ldots ,15$
L_i、R_i、L_{i+1}、R_{i+1} 均表示為 32 位元的數據
⊕：表示 XOR 的運算
K_i：表示透過金匙運算而得之 48 位元的數據
F：表示輸出 32 位元數據的加密函數。

原理：

加密：

需要將原始明文分為許多個 64 位元之區塊，每個區塊依序經過密碼系統做加解密處理，此密碼系統之架構包含一個將 56 位元金鑰（因為金匙是 64 位元，但是其中有 8 個位元是用來做錯誤更正的，所以真正有用的只有 56 位元）轉化為 16 個 48 位元金匙的金匙產生中心，一組換位系統——初始排列(initial permuttion, IP)與終結排列(final permutation, FP)，以及與 16 個金匙所做的 16 次重複運算，將明文依我們所規定的方法亂數排列，最後所得到的即為密文。

解密：

至於解密方面，DES 密碼系統在 16 個重複運算後有一個將左右兩部分對調的動作，最主要的原因是為了使解密也能使用相同的演算法，換句話說，解密之步驟其外觀與加密步驟相同，也就是我們需將密文先做初始排列，再經過 16 次重複運算步驟，而這 16 次重複運算所使用之金匙依序為 K16, K15, …..K1, 最後將左右兩部分對調後再做終結排列運算，即可得到原本之明文，這樣才是真正加密的反運算。

加解密架構：

64 位元的明文，先經過一次的初始排列運算後，所得的結果再與金匙所產生的 48 位元之 k 做連續 16 次的重複運算，並將其結果做 32 位元的左右對調，最後在經過終結排列，就可以得到密文，也就是用來在網路上所傳送的資料。下兩圖為運作說明及子金鑰產生過程。

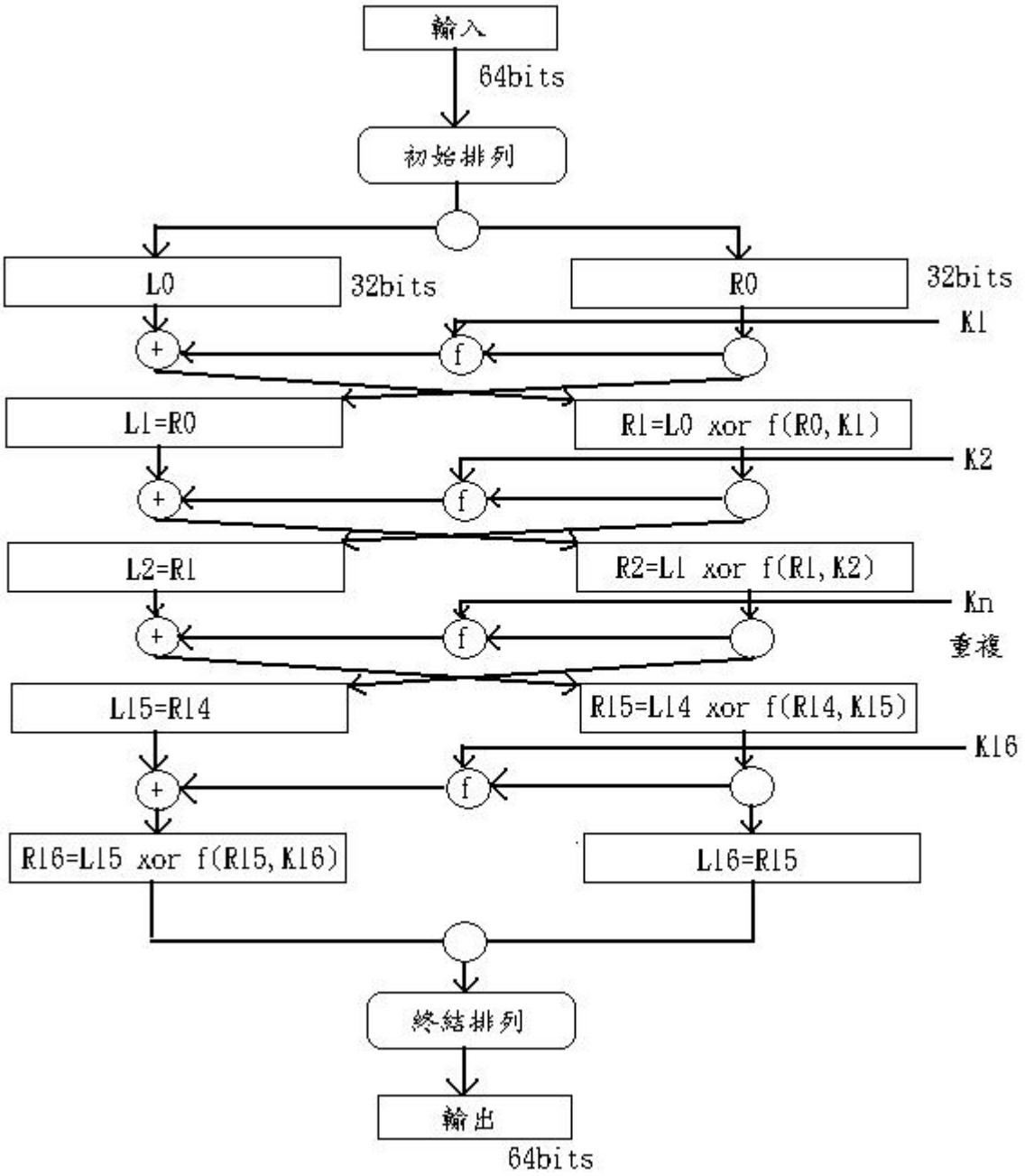


圖 3-4　DES 加解密架構

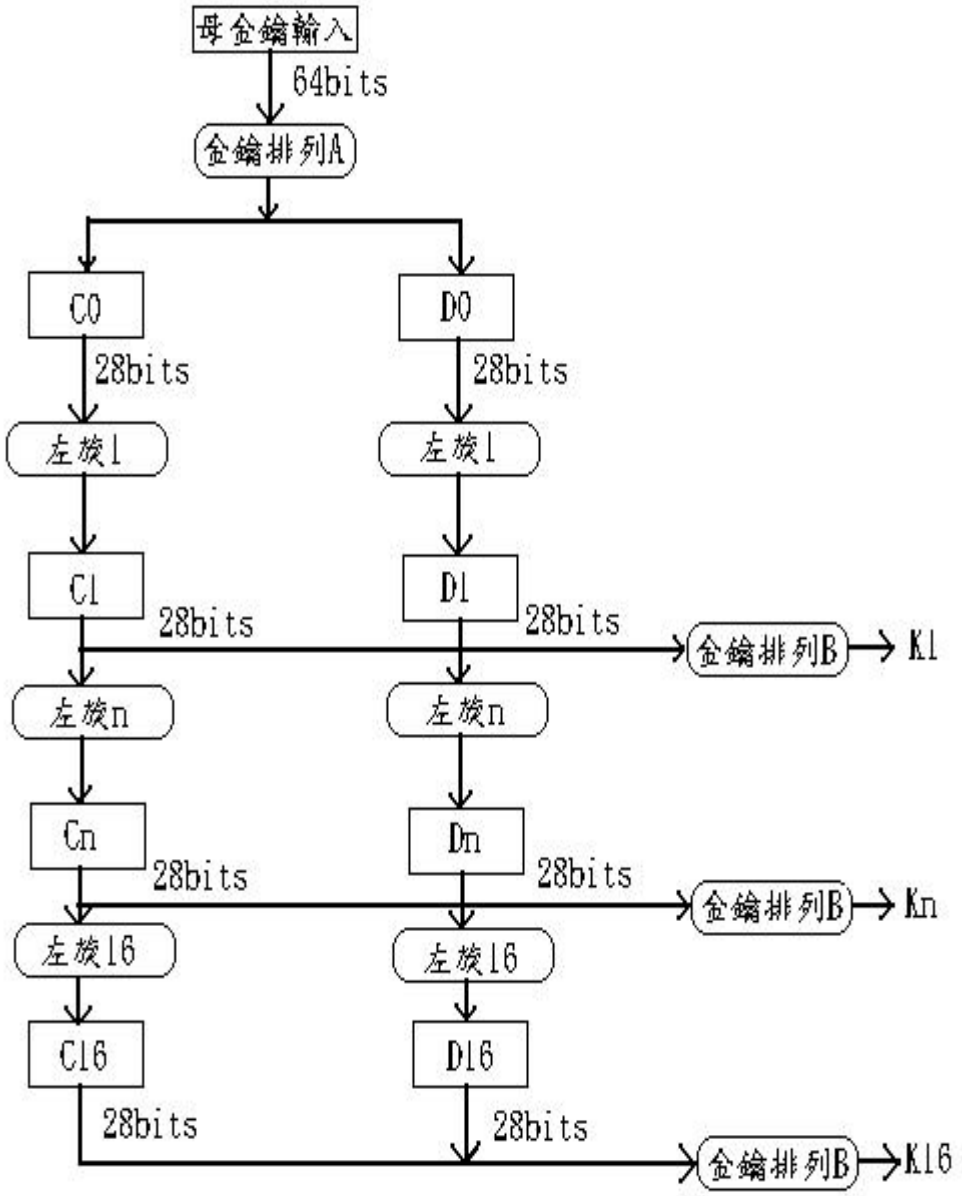


圖 3-5 子金鑰產生過程

f 函數為 DES 加密法中的重要部份，而其中的重點在替換盒上，圖 3-17 為 f 函數的計算過程架構。f 函數有兩個輸入資料，一為 32 位元的中間密文 R，另一部分為 48 位元的金鑰 K，32 位元的中間密文先經過擴增排列 E 為 48 位元，再與 48 位元的金要作 XOR 運算，所得結果再分給八個替換盒 S1，S2，…，S8。

每個替換盒輸入為 6 位元輸出 4 位元，總出書資料為 32 位元，在經縮減排列 P 後的 32 位元結果，就是 f 函數的輸出了。

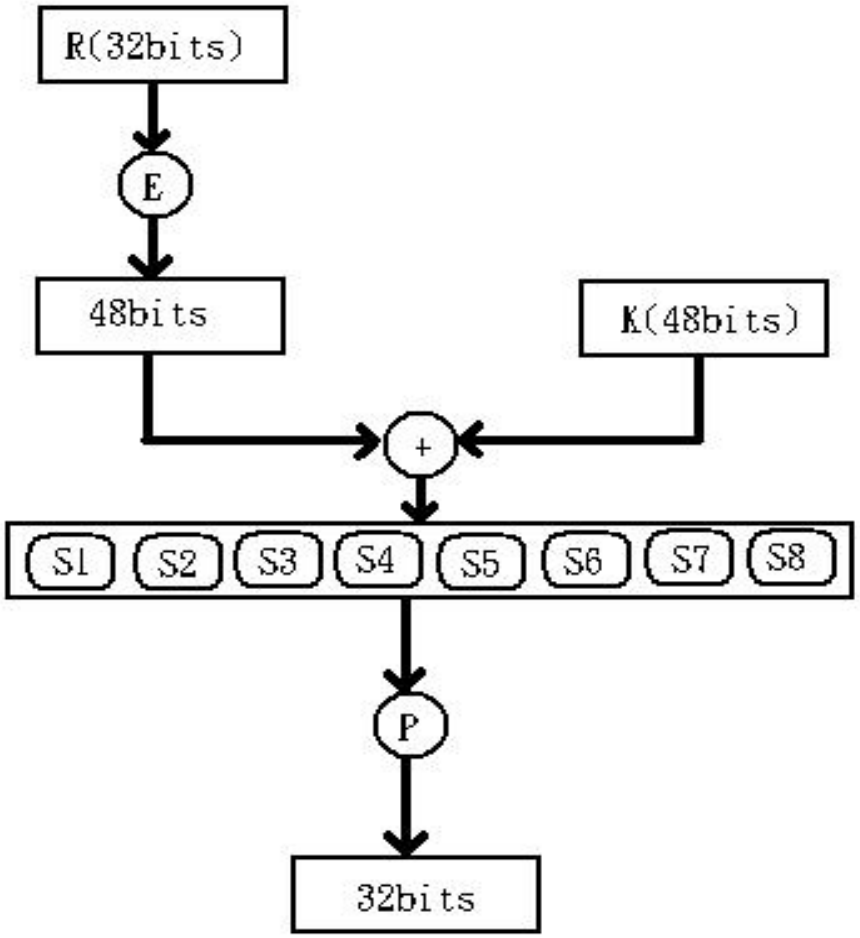


圖 3-6　f 函數計算過程

第四章 系統架構與控制指令格式

4.1 軟體架構

本系統主要是由使用者所操控之client端程式、負責處理與反應訊息之server端程式與負責監控所有家電狀態之監控端程式所組成，client端程式所送出之控制訊號經由sever端程式處理後會反應到家電與資料庫，之後server端程式再將結果反應回傳給client端程式，而監控端程式則不停的監控家電目前的狀態並將結果反應給server端程式，整個系統功能將於第五章中詳細介紹。整個系統軟體上架構如圖所示

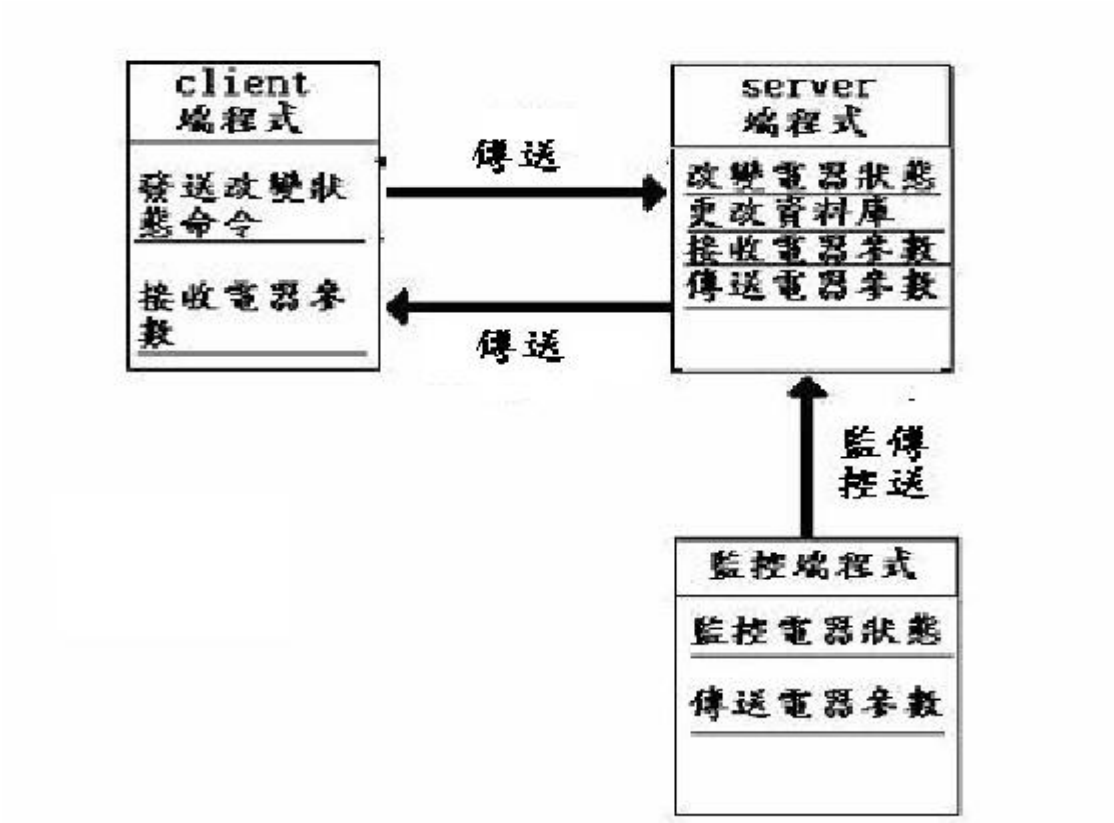


圖4-1軟體架構圖

4.1.1 警報系統

本系統透過了ADAM除了連結包括溫度感應器、煙霧感應器、瓦斯感應器等居家安全裝置，還有門窗磁簧開關、紅外線偵測器連接，提供了防盜的功能，只要感測器一接收到異常訊號，便可透過查詢藉由網路傳送至Client端，讓使用者在第一時間即可發現，提高居家安全的保障。

4.1.2 操縱家電

使用者利用本系統可透過PDA來控制家電，並可得知所有家電目前的使用狀態。

4.2 硬體架構

本系統主要是由Client端(PDA或網頁)、PC SERVER與家電所構成，而PC與家電之間是透過ADAM控制單元來聯結，由於Client端可透過無線或有線網路與Server取得聯繫以達到操控整個系統的目的，因此整體系統的網路架構是採用基礎架構模式(Infrastructure)來架構區域網路，系統構成如圖所示。

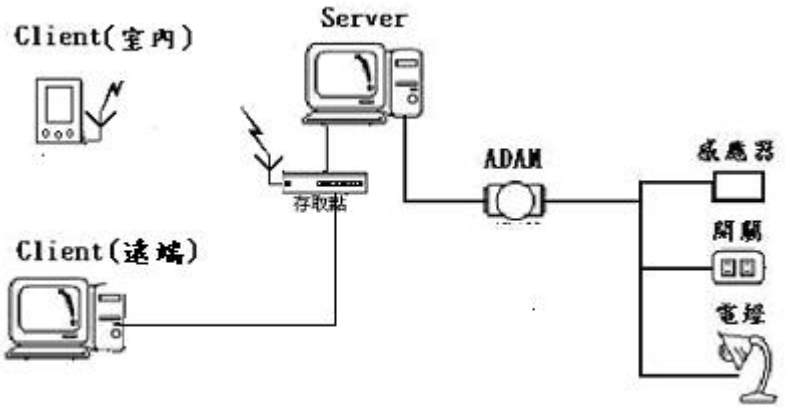


圖4-2系統硬體架構圖

4.2.1 ADAM控制方式

詳細的ADAM控制圖如圖3-3所示

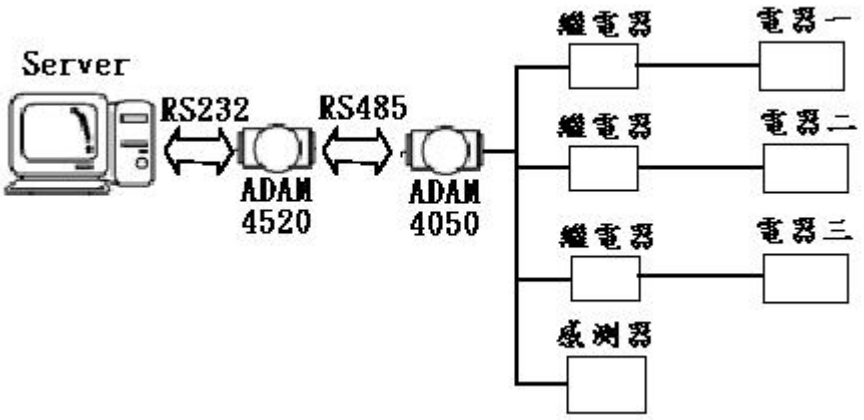


圖4-3 透過ADAM控制電器

4.3 系統使用平台

作業系統：Server：Windows XP、Client：WinCE2.0

軟體： JAVA --J2SE (Server:J2SDK1.4.0 Client：JDK1.1.8)、MySQL、Internet Information Services (IIS)、PJEE(模擬器)

硬體：筆記型電腦、PDA、ADAM模組、繼電器、溫度感測器、煙霧感測器、燈、插座。

4.4 Java 與 MySQL 資料庫連結方式

1. 使用 ODBC

此方法 Java 必須透過 Windows ODBC 再透過 MyODBC 才能與 MySQL 連線，所以其效能非常差，其關係圖如下：

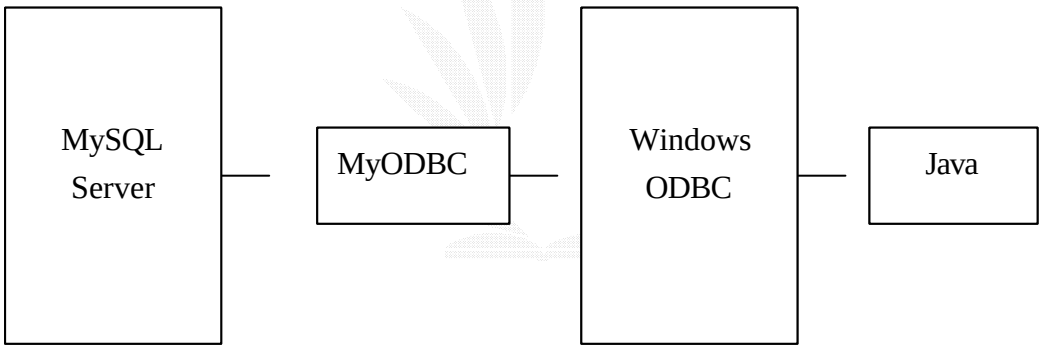


圖 4-4 Java 透過 ODBC 連結 MySQL 關係圖

以下為 ODBC 與 MySQL 連結的相關程式碼

```
Class.forName("sun.jdbc.odbc.JdbcOdbcDriver");
```

--->載入 ODBC Driver

```
Connection Con=DriverManager.getConnection("jdbc.odbc:
User Information");
```
--->進行連線

上述中的 User Information 必須先透過 Windows ODBC 資料來源管理員設定方有作用

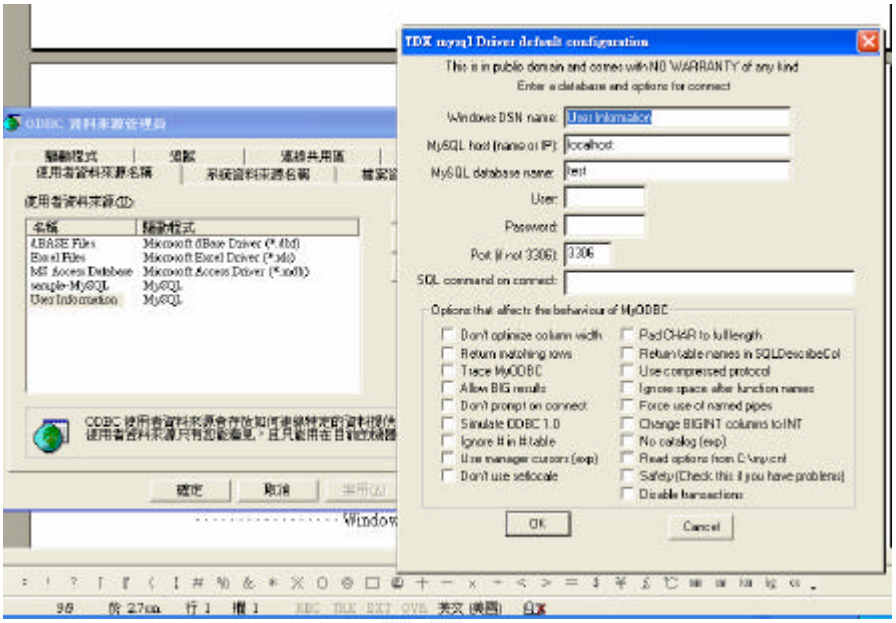


圖 4-5　Windows ODBC 資料來源管理員

2. 使用 JDBC

相對於使用 ODBC 來和 MySQL 連線，在此方法中 Java 利用 JDBC 直接和 MySQL 連線，所以在執行效能上較上述方法來的好，且有跨平台的功能，其關係圖如下

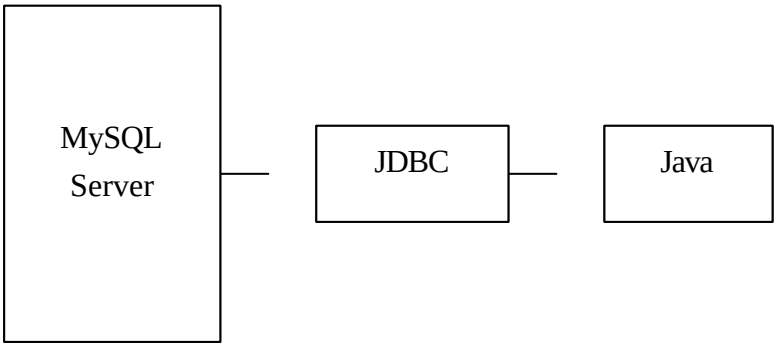


圖 4-6　Java 透過 JDBC 連結 MySQL 關係圖

以下為 JDBC 與 MySQL 連結的相關程式碼

```
Class.forName("org.gjt.mm.mysql.Driver");
```

---(載入 JDBC Driver

con=DriverManager.getConnection("jdbc:mysql://localhost/

MySQL",'帳號','密碼');---(進行連線

4.5 訊息格式介紹

在變數資料欄中，Server 端和 Client 端各有一個相同的變數資料 Array, 為防止 Array 大太浪費空間 所以的大小為固定, 在 Client 端中，每次要傳訊息前先按順序取得 Array〈變數資料〉中的一個變數，此時 Array 的 index 便指向 Array 中下一個的變數資料，來讓下次訊息傳送時，取得不一樣的變數資料，當 Array index 到底部時便又重頭開始，以此方法來獲得不一樣的變數資料，然後附加在要傳送的訊息中。在 Server 端：當 Server 接受到訊息時，解密後便按照訊息的格式, 取得由 Client 端的送來的變數資料來和 Server 中 Array 的資料進行比對，此時 Server 中的 Array index 也指向下一個位置，所以當有心人把訊息再傳一遍時，由於變數資料已是上一次的，所以此訊息便會被系統忽略。在控制訊息欄中使用家電 ID 加上開啟(on)或關閉(off)的格式來當作是控制家電的訊息格式。在使用者訊息欄中，其內容為該使用者所登入的帳號。

而在訊息加密後，程式會在密文訊息後加上金鑰編號，再把訊息送出。下圖顯示本系統訊息格式加密前和訊息格式加密後。

圖 4-7　訊息格式圖

例：

未加密訊息：Dijk254@01on%hank

Dijk254 是 Array 中的變數資料，01on 是開啟或關閉家電的訊息，最後則是使用者登入的帳號。

加密訊息： ggafsg(密文)

ggafsg(亂數)為金鑰 Index，密文為上述〈Dijk254@01on%hank〉訊息加密後之結果。

4.6 ADAM 指令格式介紹

如表 4-1 所示

表 4-1 ADAM 4050 指令表

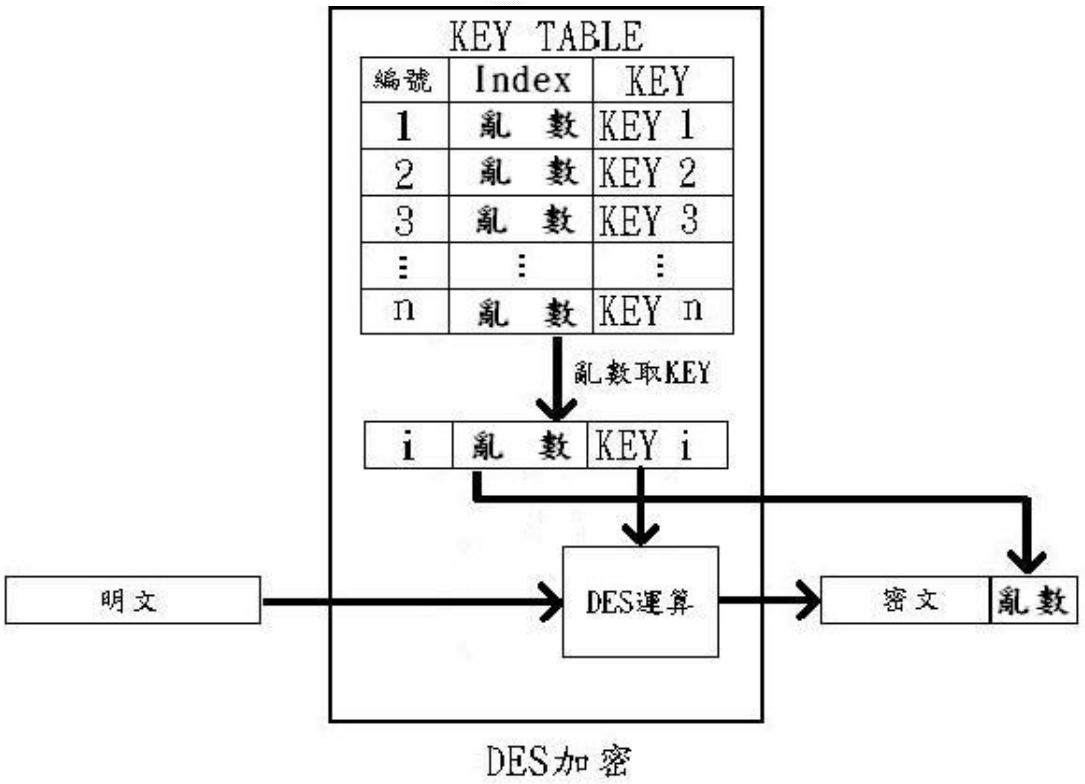
Command Syntax	Command Name	Command Description	Page No.
%AANNTTCCFF	Configuration	Sets address, baud rate, and/or checksum status, to a digital I/O module	4 152
$AA6	Digital Data In	Returns the values of the digital I/O channels of the addressed module	4 154
#AABB(data)	Digital Data Out	Writes specified values to either a single channel or all channels simultaneously	4 156
#**	Synchronized Sampling	Orders all digital I/O modules to sample their input values and store them in a special register	4 158
$AA4	Read Synchronized Data	Return the value of a specified digital I/O module that was stored after an #** command was issued	4 159
$AA2	Configuration Status	Returns the configuration parameters of a specified digital I/O module	4 161
$AA5	Reset Status	Indicates whether a specified digital I/O module was reset after the last time the $AA5 command was issued	4 163
$AAF	Read Firmware Version	Return the firmware version code from the specified digital I/O module	4 165
$AAM	Read Module Name	Return the module name from the specified Digital I/O module	4 166

第五章　系統實作及成果展示

本系統的主要功能是使用者藉由PDA透過無線網路或者藉由網頁經有線網路與Server連線互動，之後Server等候Client傳送使用者所預期動作之指令，或自動偵測突發狀並發出警急聲音來通知使用者，讓使用者可以得知所偵測到的警急狀況，並且使用者可以使用本軟體來即時操控家電或預約控制家電。

5.1 加解密

在系統的實作上，因為對稱式金鑰密碼系統傳輸金鑰有其難度，所以我們預先算好數十個金鑰，分別存在 server 及 client 的一個陣列〈KEY TABLE〉中，在加密前用亂數選擇一個金鑰 Ki 來使用，在最後附上所使用金鑰的 Index，因此明文加密後的訊息最後有解密所需的金鑰 Index，server 端接收到後，如要將訊息解密，因為擁有同樣的金鑰陣列〈KEY TABLE〉，所以只要先查看金鑰 Index，就可以得到正確的解密金鑰，然後把去掉金鑰 Index 的密文解密，就可還原成明文，而 client 收到訊息也是同樣的操作方法。



DES加密

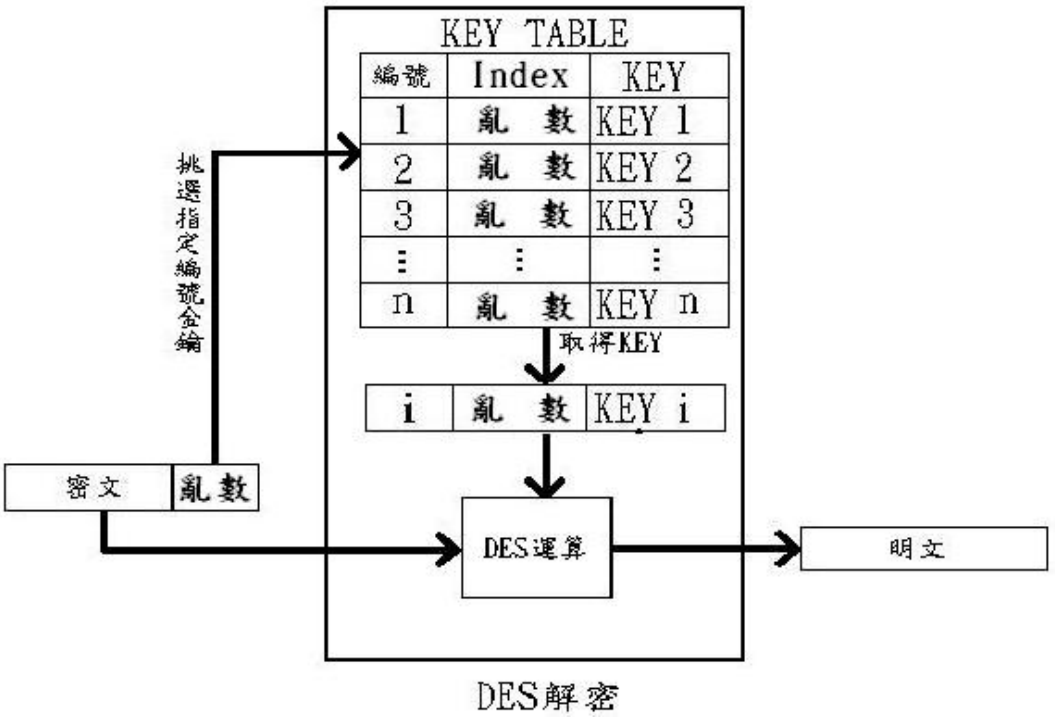


圖5-1　訊息傳送加解密流程圖

5.2 Server端實作及展示

使用者執行Server端的Application後，本程式先從資料庫取得家電資訊來更新Application上功能鍵的資訊，然後主應用程式便產生一個監控家電的thread來監督各個被設定受監控家電的狀態。接下來主程式開啟一個port等候Client端連線進來，若有Client端連線進來，Server會透過一負責儲存加密過的帳號密碼檔案來比對其認證資料的正確性，當其認證資料正確後，便會產生一個Thread的服務程式來服務這個使用者的接下來的要求，下圖為顯示Server端之系統流程圖：

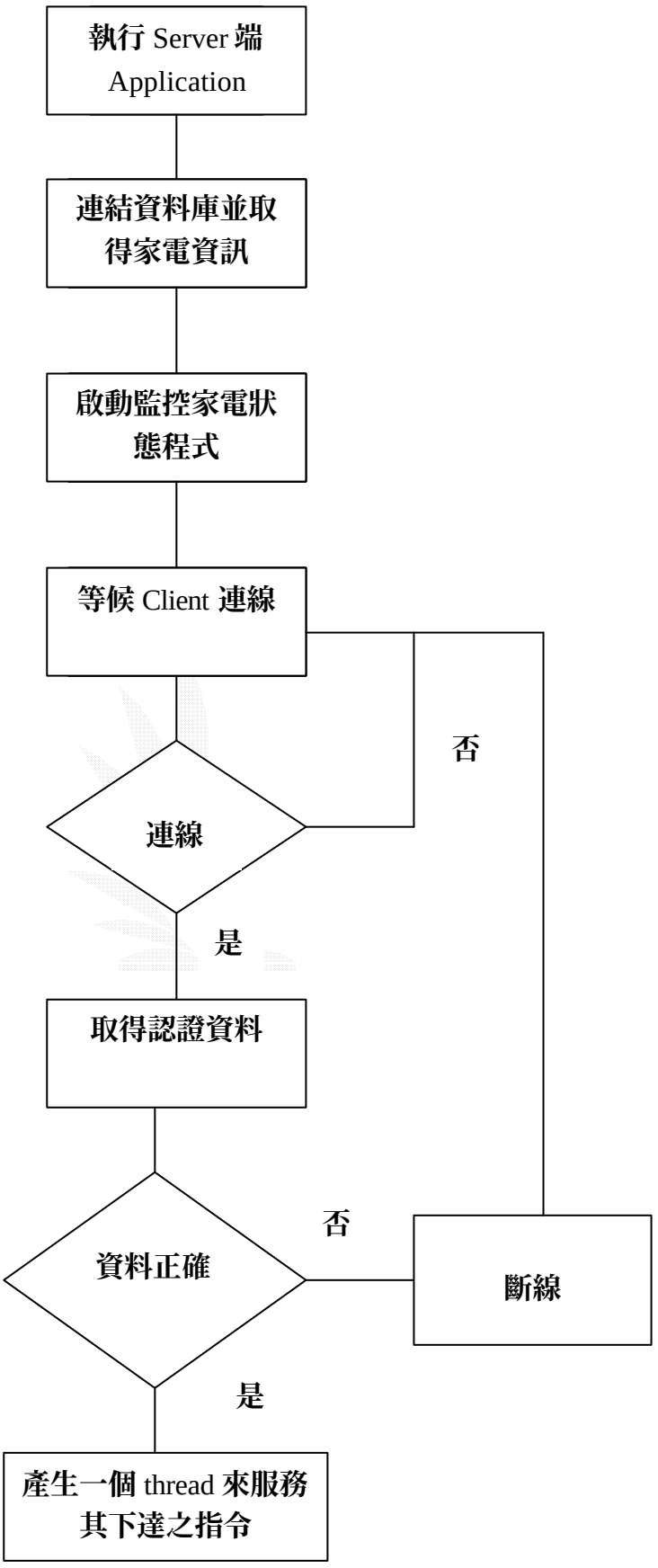


圖5-2 Server 端系統流程圖

在以下各節將介紹Server各個部份的功能。

5.2.1 等候連線

Server一開始啟動後，連結資料庫取得家電訊息資料來更新資訊並開啟一個port等候Client端的連線，在此的連線方式是透過TCP連結導向方式來確定資訊傳輸的可靠度，其展示圖所示如下：

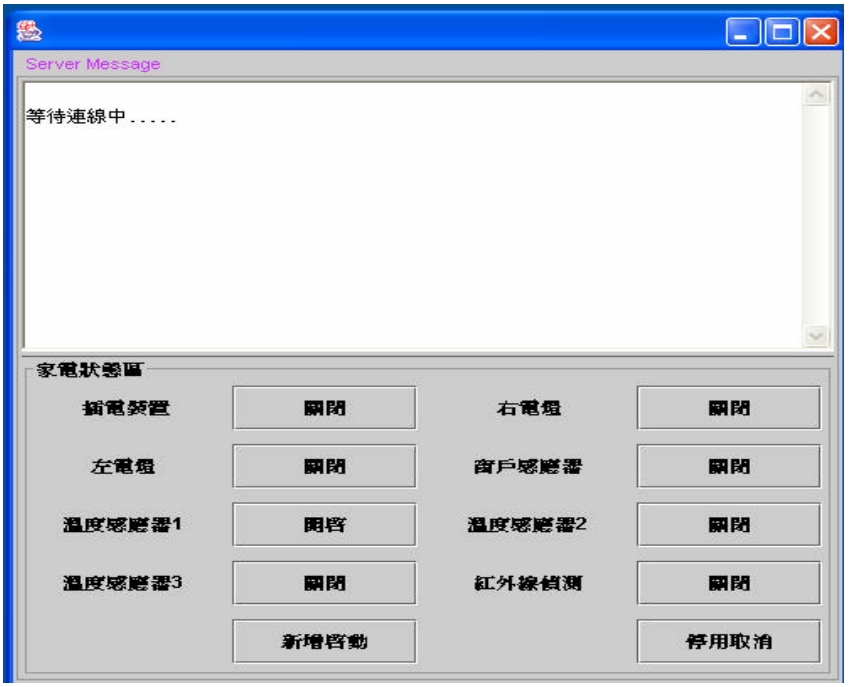


圖5-3 Server 等候連線

5.2.2 接收使用者連線

當使用者認證通過後則開啟一個Thread來服務使用者所需的服務，此Thread主要功能是將Client端送來的訊息解密後判斷其格式內容是否正確，若是有效的訊息就依其訊息內容來處理完成相關的動作，若訊息格式錯誤則不加以處理，其流程圖和展示圖所示如下：

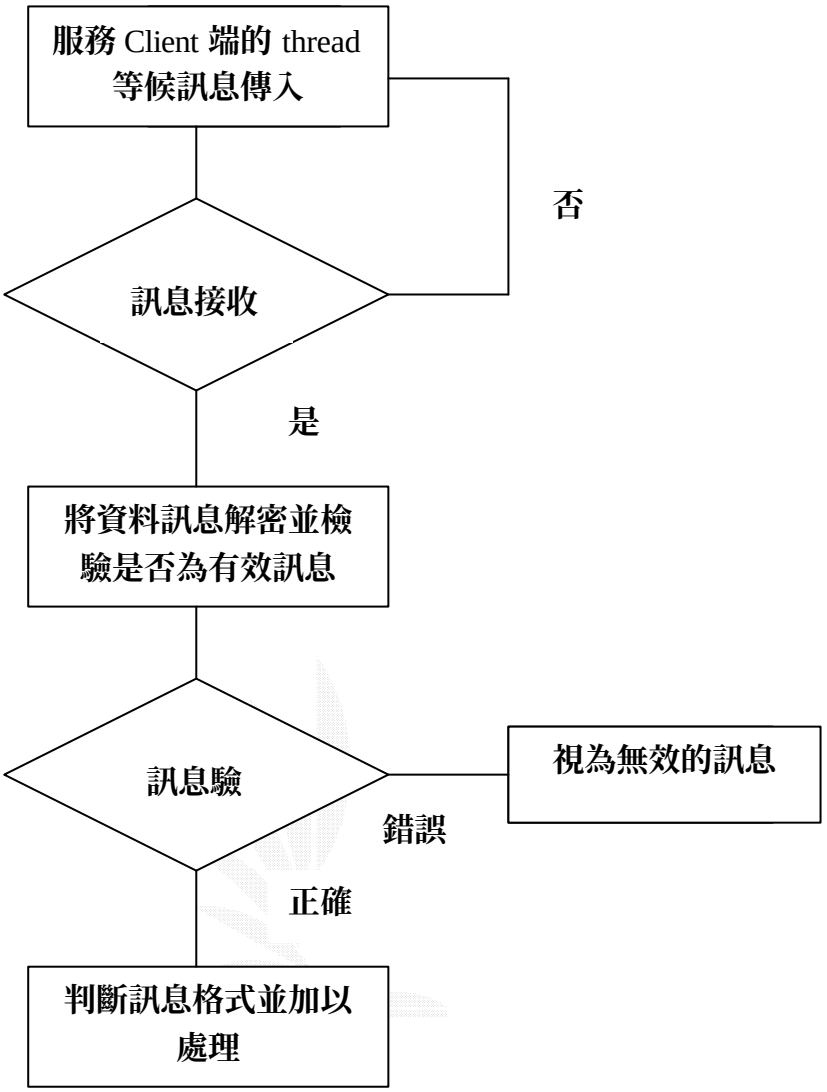


圖 5-4 Server Service Thread 流程圖

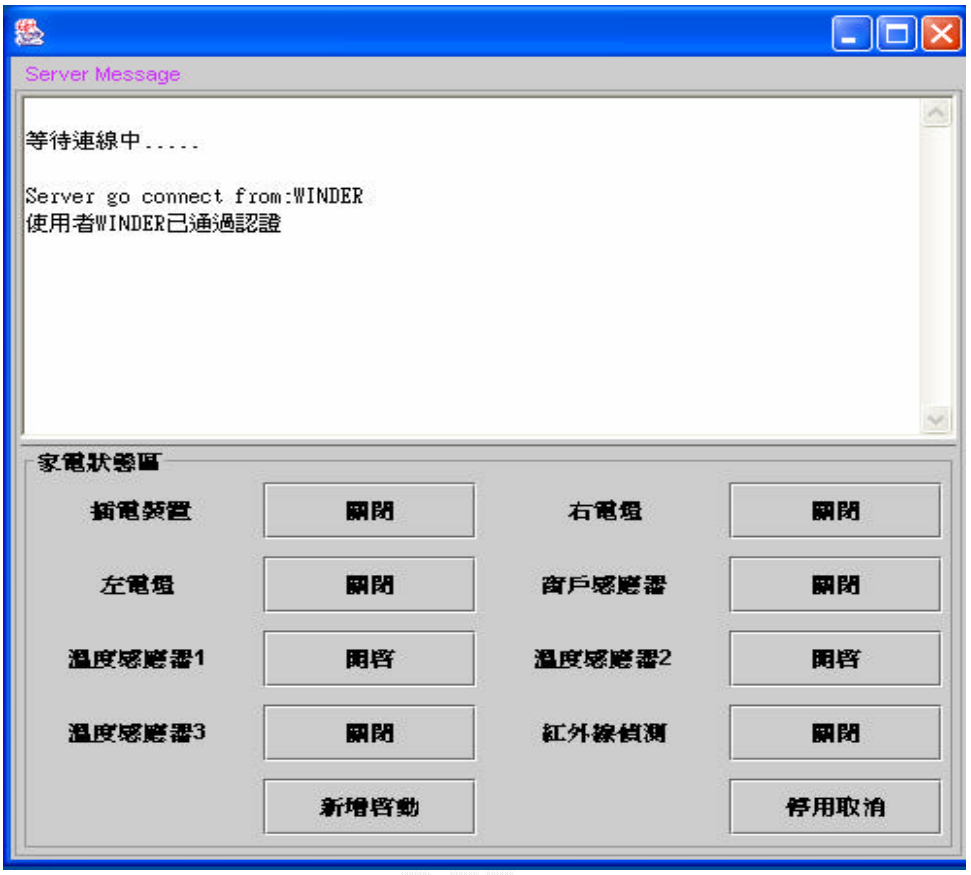


圖 5-5 Server 接收使用者連線

5.2.3 新增移除家電

在Server端可以透過新增啟動鈕來啟動新增視窗，在輸入資料後透過輸入的資料和資料庫連結更新資料庫的訊息資料，產生新的家電裝置控制，其流程圖和展示圖所示如下：

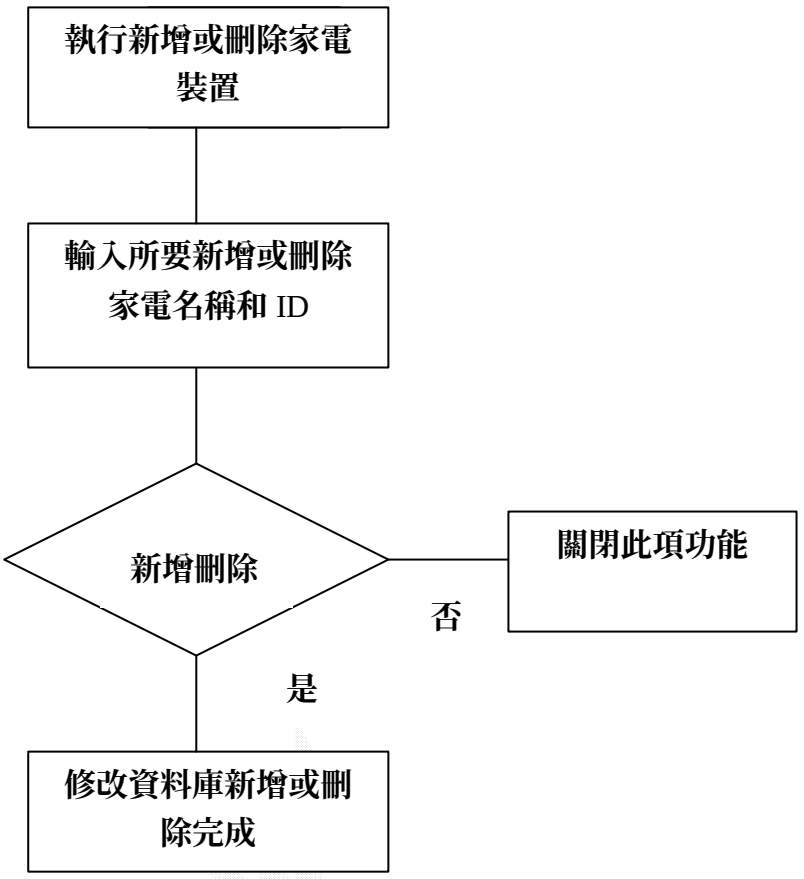


圖5-6　新增刪除家電流程圖

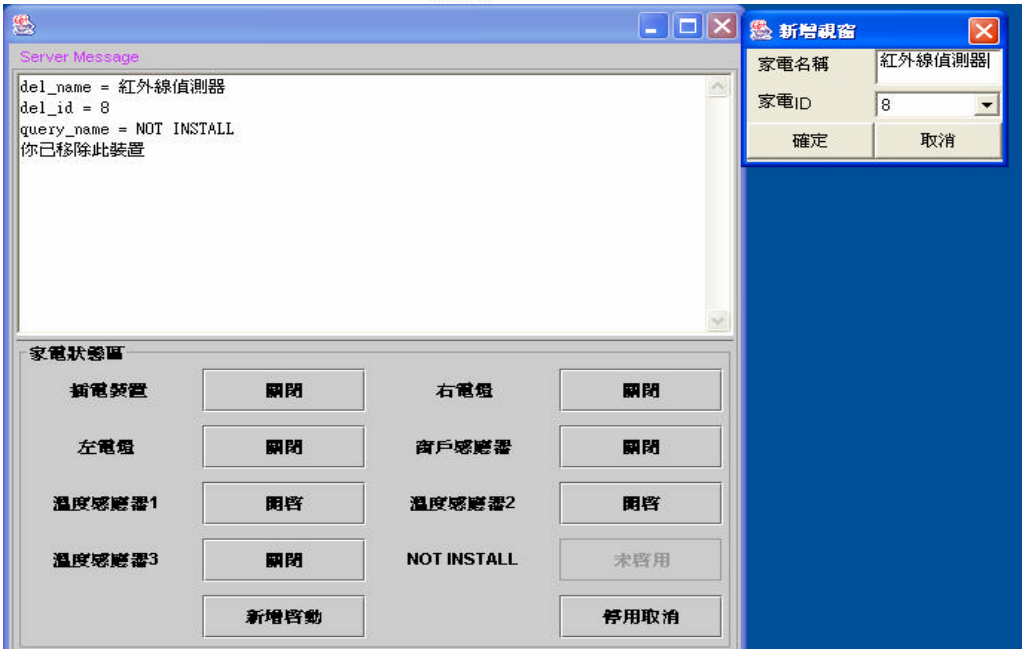


圖 5-7　新增家電裝置前

圖 5-8　新增家電裝置後

同樣的在 Server 端可以透過取消停用鈕來啟動移除視窗，在輸入資料後透過輸入的資料和資料庫連結更新資料庫的訊息資料，移除此家電裝置控制，其流程圖如圖 5-6，而展示圖所示如下：

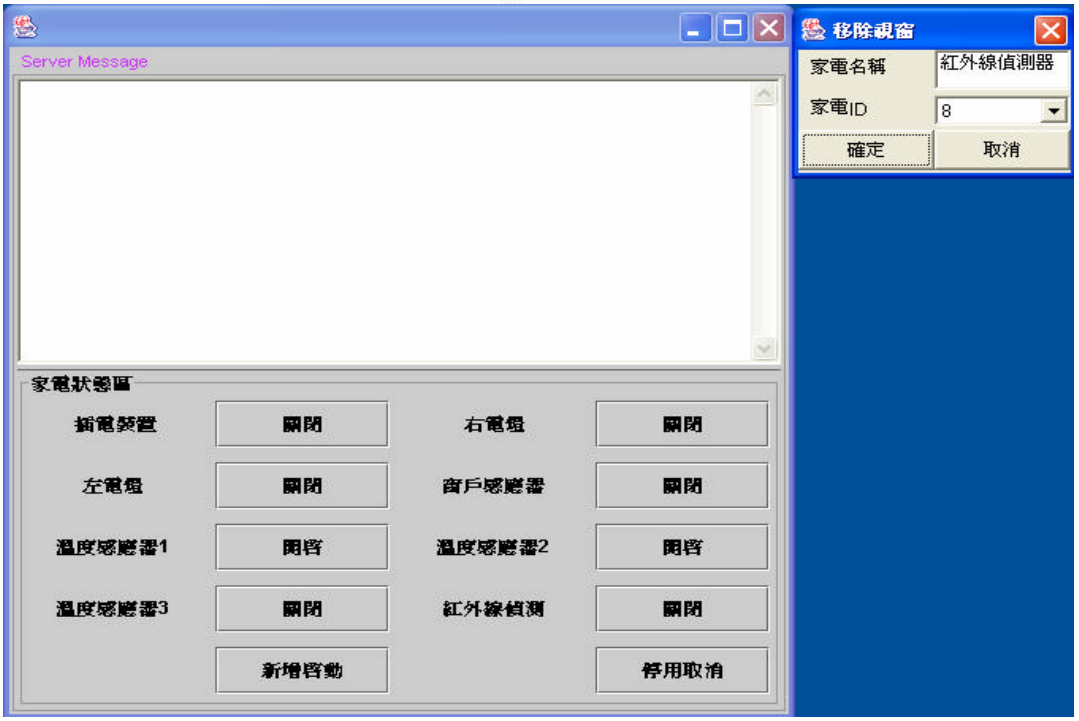


圖 5-9　Server 移除家電控制前

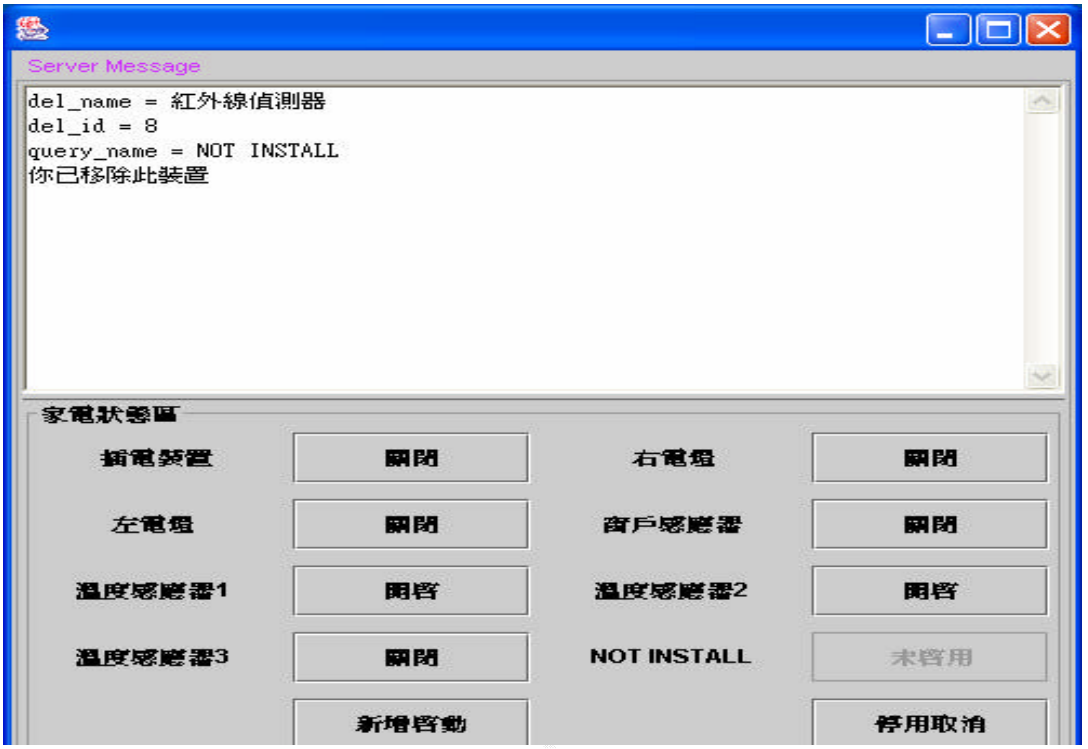


圖 5-10　Server 移除家電控制後

5.2.4　家電監控功能

Server端藉由一個Thread取得Adam上設定被受監控家電狀態的訊息，來判斷這些感應型的家電是否偵測到特殊狀況的發生，若有特殊狀況發生則程式發出聲音以告知Client端，讓此特殊狀況可以馬上加以處理，以防有任何不幸的事情發生，家電監控流程圖如下圖所示：

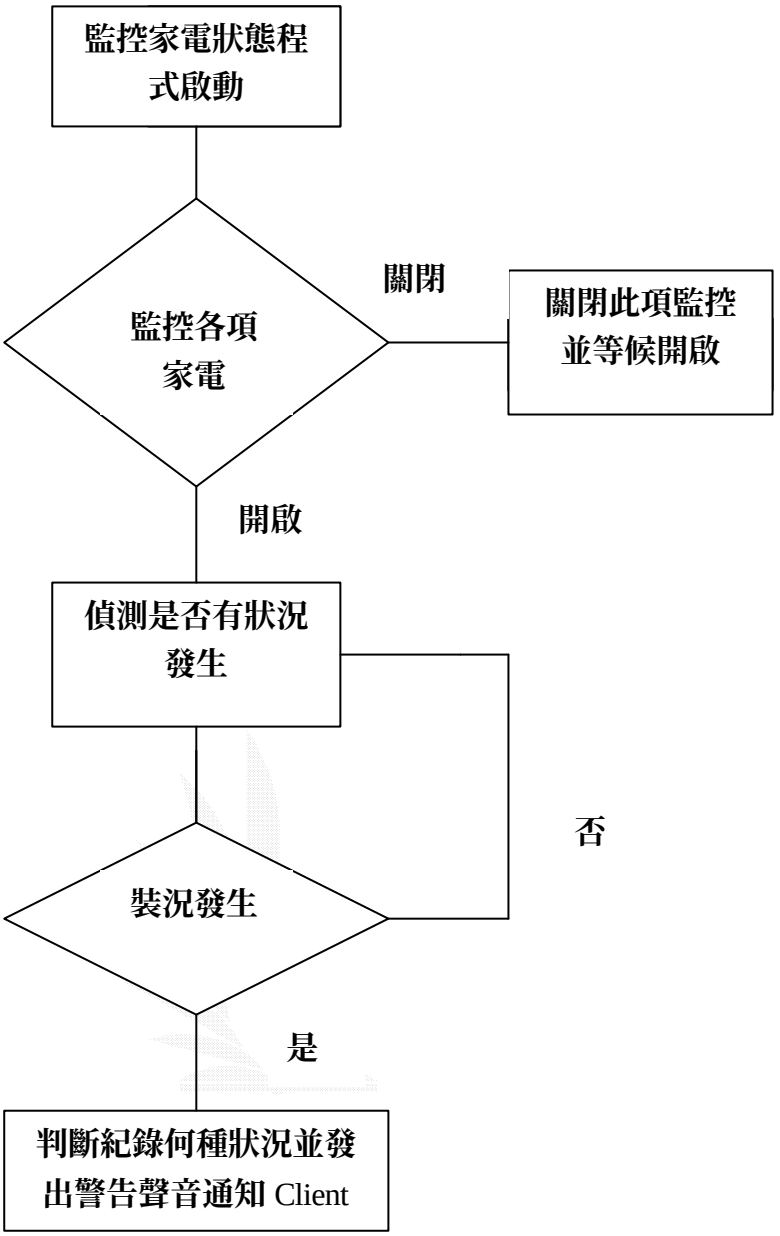


圖 5-11 家電監控流程圖

下圖為紅外線偵測器感應到有物體移動時，監控程式提出警告並發出聲音的展示圖：

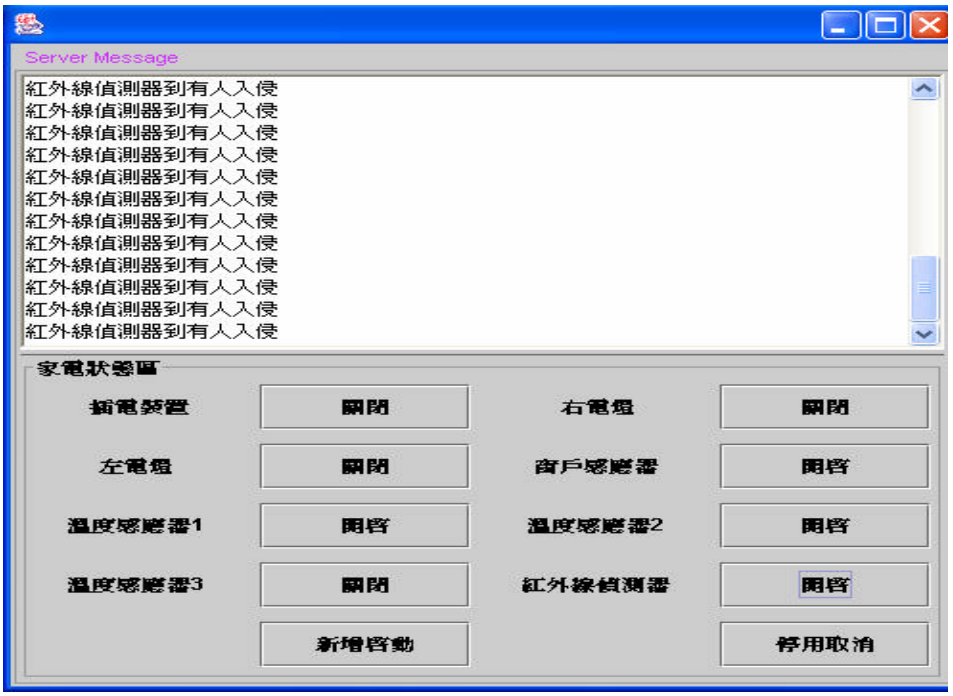


圖 5-12　Server 偵測突發狀況

此外在 Server 端透過家電裝置旁邊的按鈕來啟動 Active 視窗，透過選擇啟動或關閉不需要產生緊急回應訊息的家電，展示圖如下

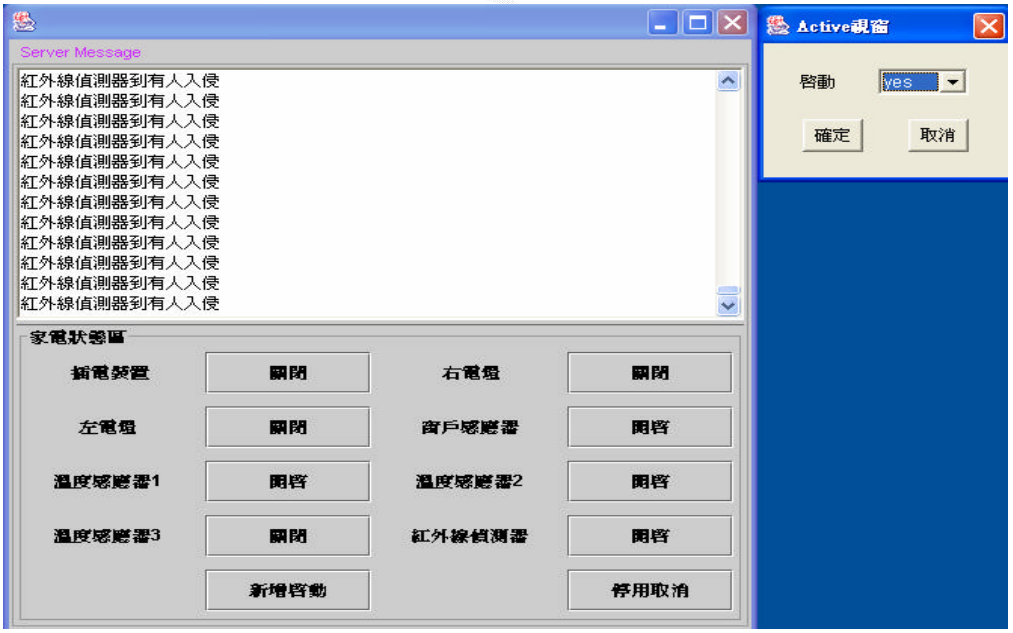


圖 5-13　Server 端關閉偵測的功能

下圖則顯示透過啟動視窗選擇關閉紅外線偵測器偵測功能後所作的變化回應，紅外線偵測器顯示為關閉，並透過訊息欄加以告知

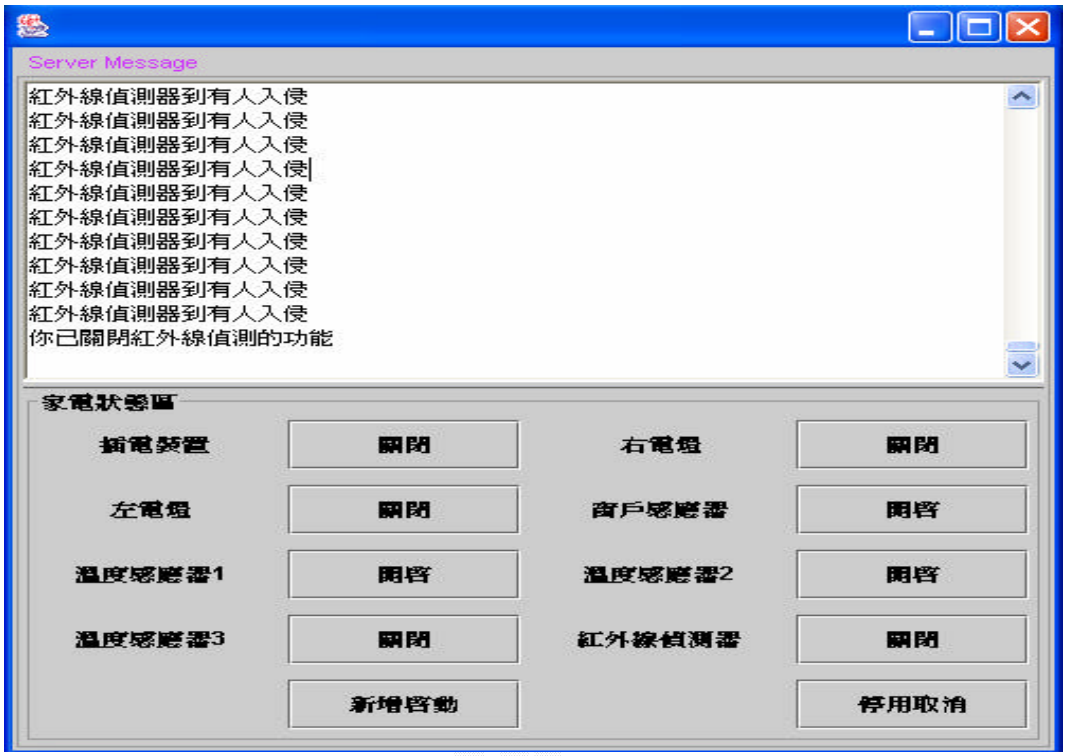


圖 5-14　Server 端完成關閉偵測

5.3 Client 端

執行 Client 端的連線後，程式會先測試與主機的連線，如果連線成功，便可輸入帳號密碼，將訊息加密後，送至 Server 驗證資料，如果成功，則開始取得資料庫裡家電狀態的資料，並可使用程式的功能。反之，驗證三次不過，程式將自動關閉。下圖為 Client 端之系統流程圖：

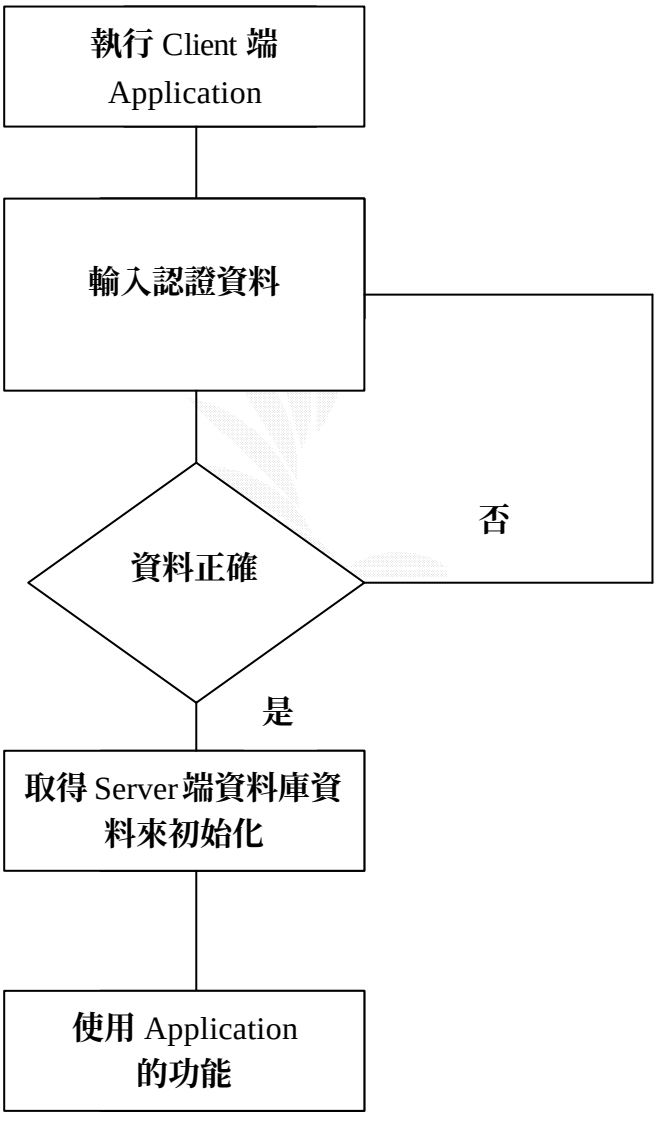


圖 5-15 Client 端之系統流程圖

5.3.1 Client程式啟動介面

Client程式啟動介面，展示圖如下圖所示

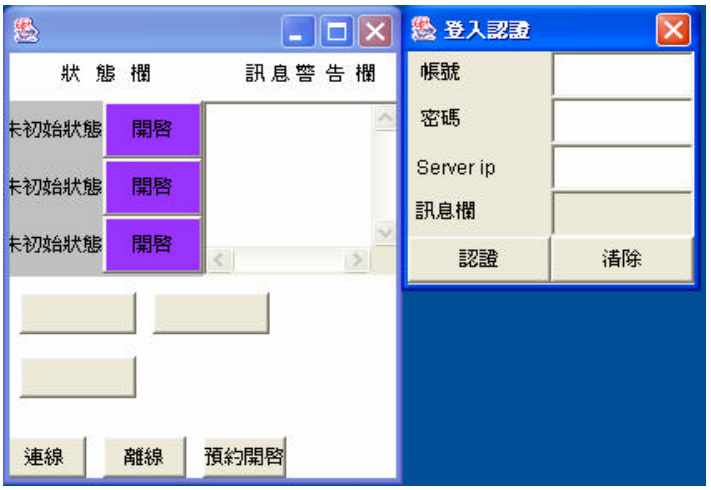


圖 5-16 Client 端程式介面(使用 Pjee 模擬器)

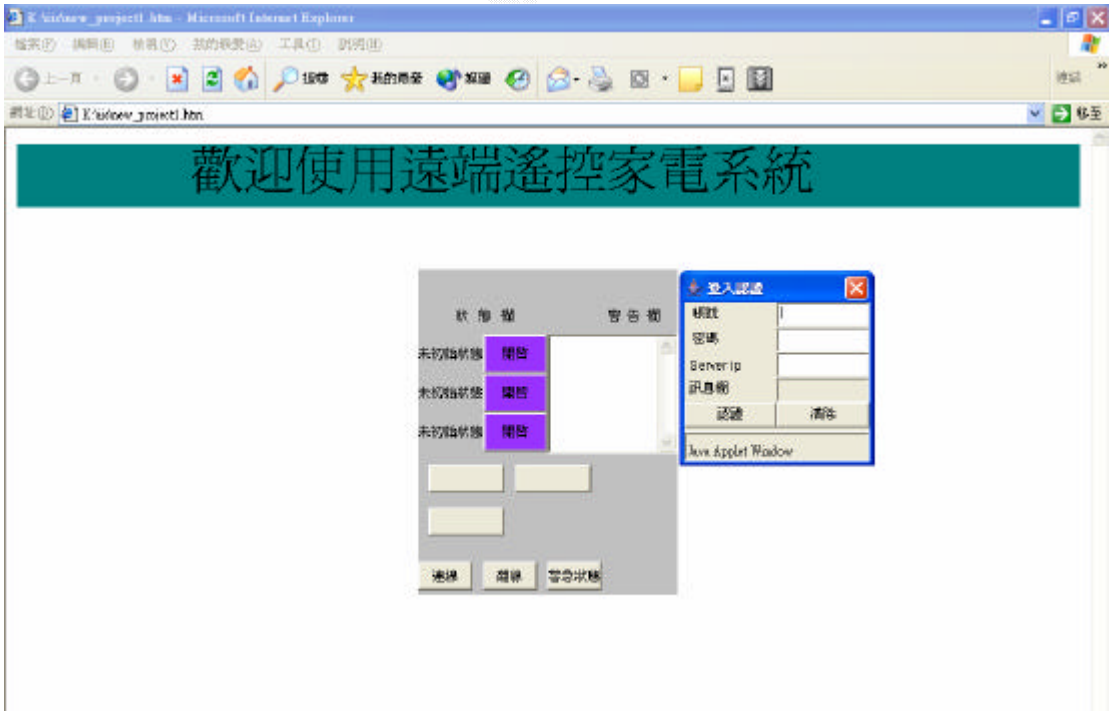


圖 5-17 Client 端程式介面(透過網頁)

5.3.2 認證

要求輸入使用者帳號密碼和登入主機ip，展示圖所示如下：

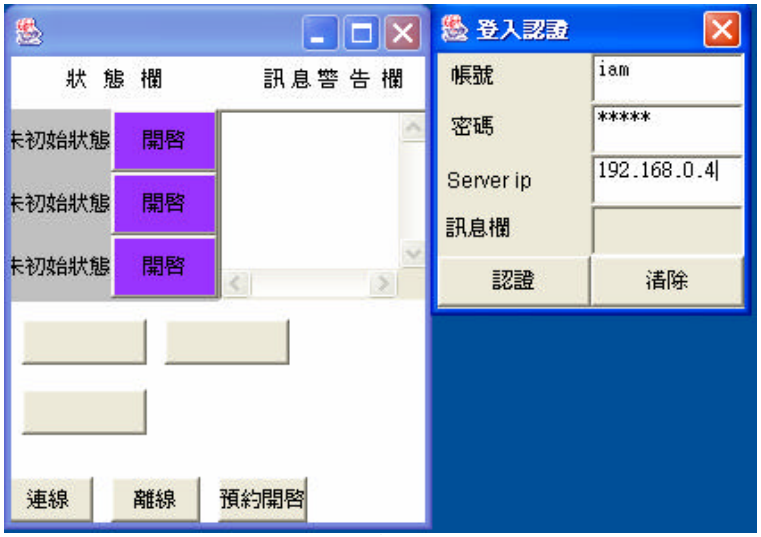


圖 5-18 認證資料的輸入

5.3.3 Client 端程式初始化

Client 端透過輸入資料取得認證後，系統和 Server 進行資料庫連結，並回傳家電狀態供 Client 端更新其介面下的家電狀態，展示圖如下：

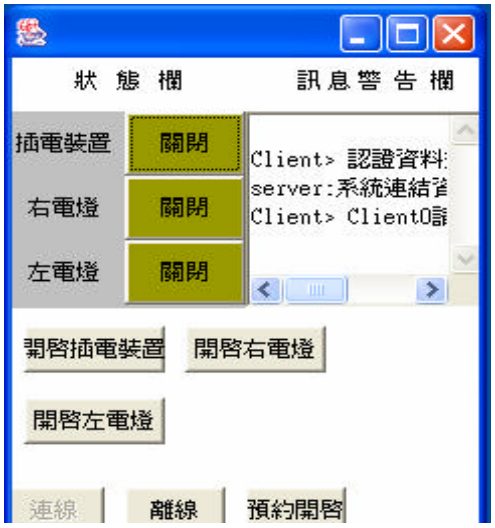


圖 5-19 認證後系統更新

5.3.4 操控家電

當使用者欲改變某家電狀態，按下功能鈕後，Client 會將訊息加密後送出，Server 收到訊息解密後，首先辨認訊息格式規則是否正確，如確認無誤，便根據訊息內容發出訊號給 ADAM 微處理器改變電器狀態，改變後再向 ADAM 確認狀態有無更改成功，接著根據修改後的狀態變數更新資料庫，然後回應一內含更新後狀態資訊的訊息給 Client，使得 Client 得以照著此資訊修改操作畫面資訊，回應給使用者知道。其流程圖和展示圖所示如下：

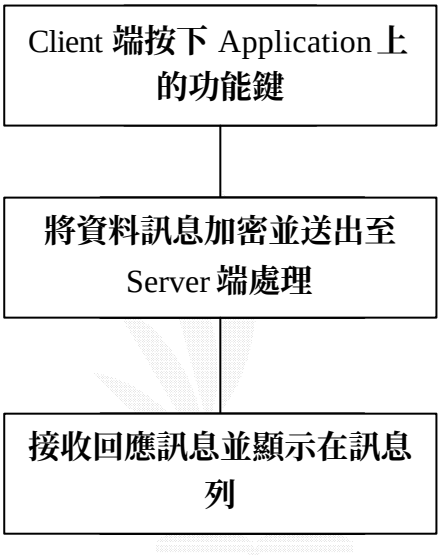


圖 5-20 操控家電流程圖

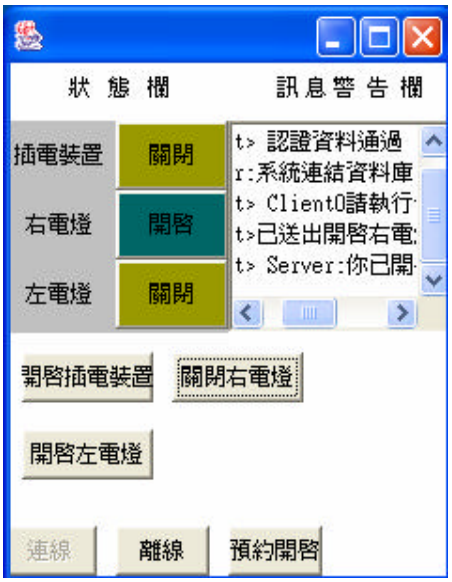


圖 5-21　Client 端開啟電燈

5.3.5　預約遙控家電

透過 Client 程式中的定時開啟關閉家電功能，Client 可以在設定的時間開啟或關閉家電裝置，時間設定完成後，即送出訊息給 server，server 端收到訊息後將此資訊送至一預約操控家電 table 中，主程式中會有一執行緒負責檢查啟動時間到達沒，當時間到即依照命令變更家電狀態及更改資料庫資料，再回傳訊息給 Client。其流程圖和展示圖如下所示：

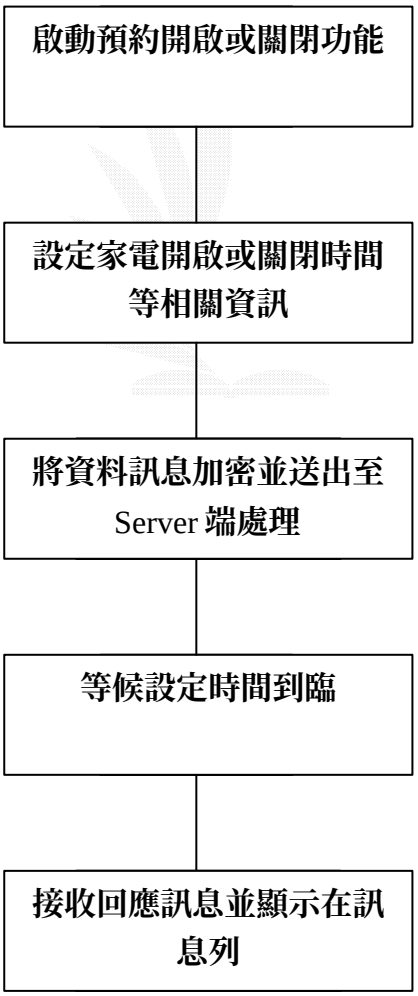


圖 5-22　預約遙控家電流程圖

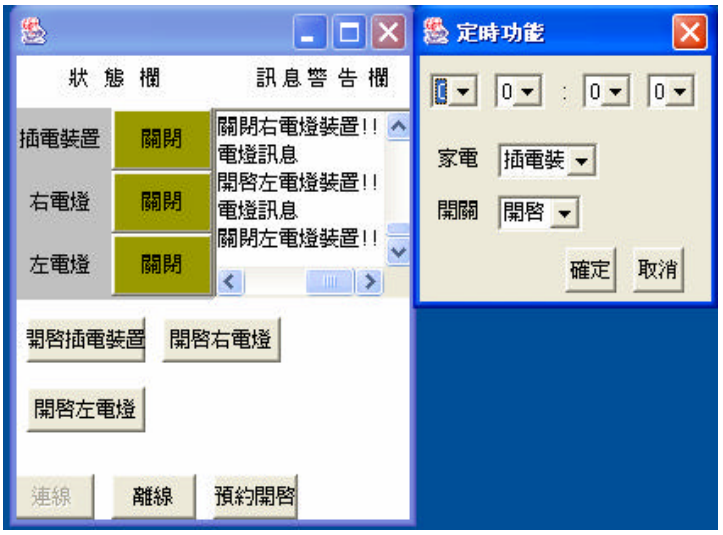


圖 5-23　Client 端程式定時啟動功能

5.3.6　系統告知特殊狀況

Server 端偵測到異常狀況〈如：門窗被打開、紅外線偵測到動靜〉，會主動發一訊息告知 Client 端，圖示如下

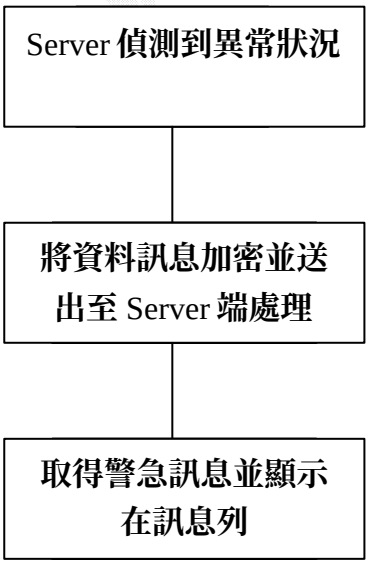


圖 5-24 系統告知緊急狀況流程圖

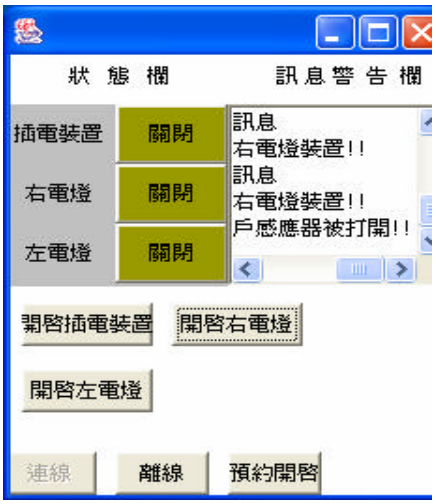


圖 5-25 Client 端接受緊急訊息

5.4 系統資料庫

在MYSQL資料庫中家電的資訊如下圖所示，Code為家電的ID、Name為家電名稱、O_switch為家電的啟動狀態 I_switch為Adam所得的輸入狀態、Active則是Adam中的連接埠是否有使用，下表為資料庫各欄位屬性表和資料庫中家電的資訊

表5-1 資料庫各欄位屬性表

Field	Type
Code	Int(2)
Name	Char(20)
O_switch	Int(2)
I_switch	Int(2)
Active	Int(2)

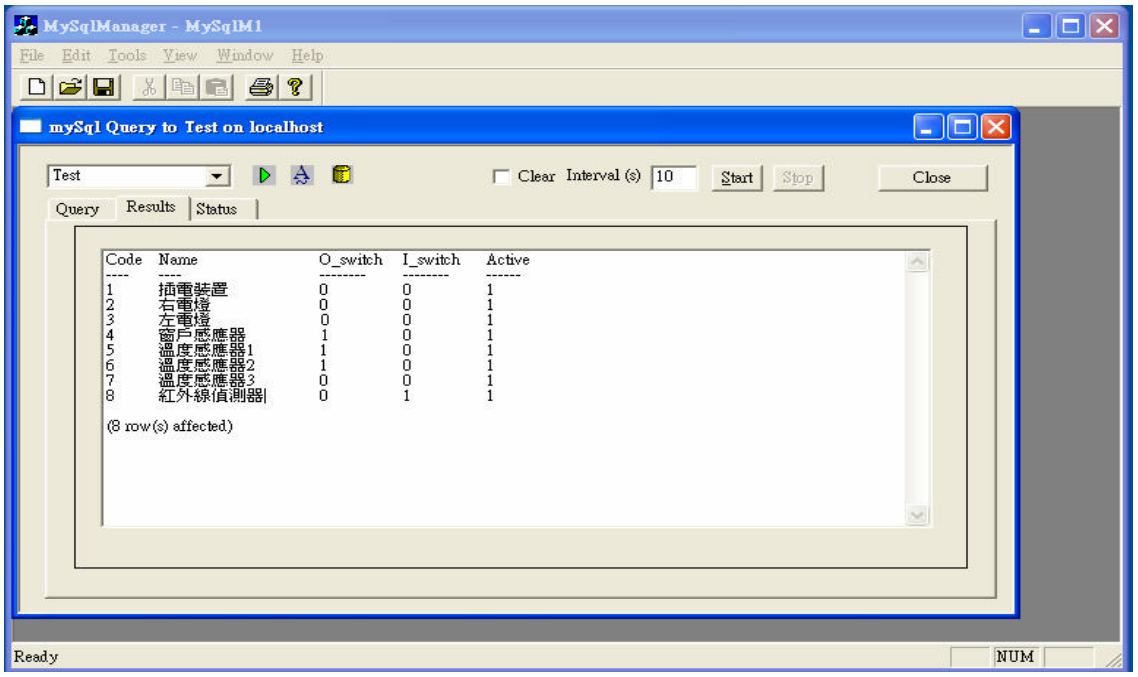


圖5-24 MYSQL資料庫中家電的資訊

第六章　心得感想與結論

6.1 遭遇到的困難

PDA 上程式的寫作

由於我們使用的 PDA 上免費的 JVM 只有 SUN 出的 PJEE(Personal Java Emulation Environment)，只有實作到 JDK1.1.x 所以有很多已做好的 JAVA 物件都無法使用，如 SWING、加解密套件，所以有很多東西變的都要自己去實作。

ADAM 控制模組的部分

剛拿到老師幫我們借的板子後，發現其上近有奇怪的三個控制器，後來上網找資料，才得知其相關的資料和其傳送接收資料的規格，可是在要和其溝通時卻發現資料有送進去但是其上所控制的電器卻都無法開啟，於是便想去學校請問其他有用過這個 ADAM 的同學或學長，可是還是沒有得到解答，後來只好再去網路找資料，最後發現是少了 Carriage Return 的符號，最後加上去時發現板子上電器啟動時，實在感動莫名。

JAVA 程式的寫作

由於以前很少接觸到 thread 的部分，而這次專題很多部分又需要用到 thread，所以這部分也令我們感到相當的頭痛，所以花了很多時間再研究這個部分，其中包括請問同學或是上某某 JAVA 論壇發問請教別人，最主要的還是自己找資料來研讀。

尋找資料的部分

專題中用到許多技術，如 Java Comm. Api、Java 資料庫連結驅動程式等等都是花了相當的時間去研讀並找相關的資料，才獲得相關的技術，並加以實用，雖然花了不少時間，但相對的也獲得了一定的收穫。

6.2 心得感想

本專題能夠順利的在發表前完成首先要感謝的便是我們的指導老師-李維斌老師的殷勤指導，每當我們有問題疑慮時提供了我們許多不同觀點的思考方向，而且還幫我們商借的家電模擬的板子，解決了我們在硬體實作上的困擾。其次要感謝的便是助教，從專題一開始到結尾都不停的在監督我們的進度，也不斷的為我們提供寶貴的意見。

同時我們也在專題的製作過程中了解了團隊合作的重要性，組員之間的協調與溝通是很重要的，雖說每個組員都有其獨立負責的部分，但彼此之間進度也是要互相配合才行，若因某位同學的進度落後便會造成整個系統測試進度落後，因此同學間都需彼此互相督促，而且有時當某部分發生問題時也會幫忙了解一起把問題解決。

最後，藉由這次專題我們學到了許多，不管是專業知識還是經驗，這些都是我們以往只在做學校作業時很難學到的。

個人心得部分

吳宏澤：

在三上時找到老師並確定專題題目後便和組員進行系統架構和工作分配的討論，在專題我所負責的部分是Client端程式、Server端程式和監控程式等部份，由於這些部分大都是程式撰寫的部分，而雖然以前曾經使用Java寫過幾次作業，可是在真正深入專題的研究時才發現以前所學根本是基礎，所以從那時候起便開始努力研讀Java，從最基礎的部分開始慢慢深入研究並開始專寫程式，從一開始如何在PDA上專寫程式、系統基本介面設計、網路程式、連結資料庫、如何使用新的API及其相關路徑的設法、到最棘手的Thread等程式部分，在這其中遇到了很多困難，有時到了真的不知道該如何往下一步邁進時，就連絡組員互相討論其解決的辦法，或是請教班上一些程式高手或助教、上一些較大的程式論壇發問以尋求解決的方法，就這樣一直到專題完成，所以在本專題實驗中我獲得了很多，包括在撰寫程式上、遇到問題該如何解決及團隊分工等等的寶貴經驗

孫誌明：

在專題中接觸到了許多以前沒碰過的知識與技術，像RS232就只限於在課本上所得到的認知，而至於ADAM則是生平第一次接觸。以前使用RS232傳遞訊號時都是線路一接便開始使用，從來沒想過控制程式是如何運作，而專題中則需要自己編寫控制程式來使得訊號能夠順利的透過RS232來傳輸，在RS232相關書籍中雖有程式不過大都是C或組語所編寫，要如何將其轉換為JAVA程式則是對自己JAVA程式功力的一大挑戰，至於ADAM在資料收集之初所能獲得的有用資料幾乎是沒有，後來經過老師的指點透過其他管道才獲得ADAM的規格書與控制指令集，要是沒有這些資料恐怕連怎樣讓ADAM動作都不知道，而且除了本身負責部分的知識之外，在和組員討論問題的同時或多或少也能吸收到其負責部分的相關知識，所以在完成專題的同時我們也因而學到了許多寶貴的專業知識也擴展了我們的認知。

蔡宗翰：

這次所作的專題，所獲得的經驗跟以前上課的作業有著相當大的差別。以往上課的作業重點往往是集中在藉由實作更了解課堂所學習到的理論，或是增進程式設計能力。但是在專題中，雖然技術能力要有一定的程度，但最重要的卻是組員之間的溝通交流。當一個專題訂定後，分成許多部份，每個組員有互相合作也有獨立製作的範圍。對於我來說，最大的收穫，也是最難學習是要怎麼將我負責的東西或是我在專題上的想法，完整且清楚的傳遞給其他組員了解。因為大家都有修不同的課目，也有補習，能湊在一起的時間本就不多。剛開始的討論大家花了許多時間，但結論卻沒有多少，白白浪費了許多時間，慢慢的，我和大家都了解到，要在短短的時間內討論好接下來的方向、之前的進度及彼此的心得，不只一定要事前就做好準備，還要找出真正直得討論的重點，而不是那些可有可無的事，在討論時就只提重要的地方，如此一來，整個組的效率漸漸的提升，自己分析事情的能力也慢慢的增長，也逐漸懂得更精簡的表達出自己的意思。當然，在這個專題中，程式能力也有所提升，但我覺得，上述的收穫讓我最為振奮。

6.3 系統未來展望

本系統目前是使用PDA，未來在平板電腦普及後，可以做出功能

更為強大且控制更為友善的介面，不用再受限於PDA的軟硬體資源限制。

至於控制家電方面，由於本系統是採用接線的方式來連結家電裝置，未來Server可改用無線模組的方式來和各個家電裝置連線並透過JAVA JINI的技術，實現真正無線的PDA遙控家電。

而且由於技術及硬體上的關係，目前只能夠做到對於家電進行開關的動作，但是這樣對於家電所擁有的功能除了電燈只有開與關之外，面對其他家電所擁有的功能而言目前此系統的功能便明顯的感到不足，因此對於此系統未來的展望便是希望能夠透過技術上的提昇和家電硬體軟體上的整合，以達到能夠真正指藉由PDA便能夠隨意的控制家中所有的家電。

6.4 分工情形

在表 6-1 中將說明各個組員在專題中所負責的部分

表 6-1 分工情形表

工作項目	負責人員
系統架構規劃	全體
PDA 模擬器尋找	蔡宗翰
Client 端程式	吳宏澤
Server 端程式	吳宏澤
監控程式	吳宏澤
資料庫	孫誌明
ADAM 控制程式	孫誌明
DES 加解密實作	蔡宗翰
操作介面設計	全體
IIS 建置和 Applet 網頁	蔡宗翰
報告撰寫	全體
系統測試	全體
系統整合	全體

6.5 甘特圖

下圖為顯示我們專題工作進度的甘特圖

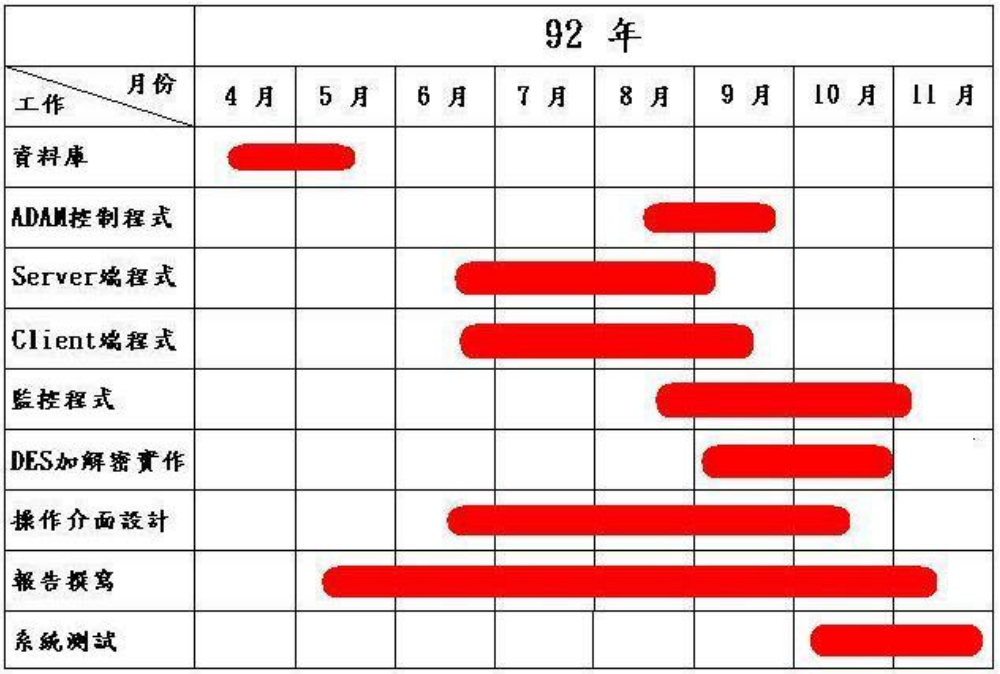


圖6-1 甘特圖

參考資料

[1]JAVA 密碼學 / [柯努特生]Jonat han Knudsen 原著; 阮韻芳譯

[2]近代密碼學及其應用/ 賴溪松, 韓亮, 張真誠著

[3]Pocket PC 無線網路與 RS-232 程式設計 / 龍仁光編著

[4]精通 NSBasic/Palm 程式設計 / 王道榮編著

[5]Java 安全防護 / Scott Oaks 原著; 高秀美譯

[6]Windows CE 程式設計 : 使用 JAVA 語言 / 楊攸中著

[7]Java 資料庫程式設計入門與實作 / 黃國欽編著

[8]Java 函式庫全集 / [派屈瑞克・張]Patrick Chan, [羅沙納・李]Rosanna Le-e 原著; 堤香工作室編譯

[9]Java 函式庫全集 / [派屈瑞克・張]Patrick Chan, [羅沙納・李]Rosanna Lee 原著; 堤香工作室編譯

[10]WLAN 無線網路系統剖析與應用 / 鄭同伯著

[11]Pocket PC 無線網路與 RS-232 程式設計 / 龍仁光編著

[12]Java 程式設計 / H. M. Deitel, P. J. D eitel 原著; 楊錦文, 鄧永亥, 謝金興編譯

[13]JAVA 程式設計經典範例 / [詹姆薩]Kris Jamsa 著; 廖蕙君,黃文男譯 出版項 台南市 : 大偉, 1997[民 86]

[14]Java 教學手冊 / [康派爾]Mary Compione ,[華拉斯]kathy Walrath 原著; 堤香工作室編譯

[15] http://www.jsptw.com/jute/index.html

[16] http://www.cryptix.org/products/jce/index.html

[17]http://www.pcnet.idv.tw/pcnet/network/network_ip_tcp.htm

[18]http://liy.slat.org/study/network/tcp_ip/network_ip_tcp.html

附錄 A

- **專題使用之 PDA**

圖 附錄-1　PDA

- **專題模擬用之家電板子**

圖 附錄-2　家電模擬版子

圖 附錄-2-1　家電模擬版子-大門紅外線偵測器、門/窗磁簧開關

圖 附錄-2-2　家電模擬版子-手動電燈開關、家電設備遠端控制

圖 附錄-2-3　家電模擬版子-ADAM 裝置、火災

附錄 B

● SQL 指令介紹

SQL 是「結構化查詢語言」（Structured Query Language）的簡稱，是由 IBM 公司於 1970 年代所發展出來，用於關連式資料庫 (Relational Databases) 當中的一種資料庫查詢語言，利用 SQL 可以用來定義資料庫結構、指定資料庫表格與欄位型態與長度、新增資料、修改資料、刪除資料、查詢資料，以及建立各重複雜的表格關連，成為一個查詢資料庫的標準語言。雖然各家資料庫所提供的 SQL 語言在功能上會略有差異，但基本的功能是一致的。SQL 的設計基本上是模仿英文的自然語法，所以在入門上較為容易

任何資料庫都有四個基本查詢動作，即檢視、新增、修改、刪除，所以介紹以下四種基本功能的 SQL 語法：

- 檢視資料： 若要檢視資料庫的資料，使用的 SQL 指令是「SELECT」，基本語法如下：

```
SELECT 欄位名稱 1,欄位名稱 2,...
FROM 資料表名稱 1,資料表名稱 2,...
WHERE 條件式
ORDER BY  欄位名稱 1,欄位名稱 2,...
```

SELECT：所接的欄位名稱為待查資料庫的欄位名稱，各欄位名稱之間以逗號隔開。

FROM：所接的資料表名稱為待查資料庫的資料表名稱，各資料表名稱之間以逗號隔開。

WHERE：所接的條件式為設定查詢的條件式。

ORDER BY：所接的欄位名稱為欲排序的欄位，可將查詢的資料加予排序，指定多個欄位時則以欄位名稱 1 排序，若其資料相同則再依欄位名稱 2 排序，依此類推，各欄位名稱之間以逗號隔開。

- **新增資料： 若要新增資料庫的資料，使用的 SQL 指令是「INSERT」，基本語法如下：**
- **INSERT INTO 資料表名稱(欄位名稱 1,欄位名稱 2,...)**
- **VALUES (欄位 1 的資料,欄位 2 的資料,...)**

修改資料： 若要修改資料庫的資料，使用的 SQL 指令是「UPDATE」，基本語法如下：

- **UPDATE 資料表名稱**
- **SET 欄位名稱 1=欄位 1 的資料,欄位名稱 2=欄位 2 的資料,...**
- **WHERE 條件式**
- **刪除資料： 若要刪除資料庫的資料，使用的 SQL 指令是「DELETE」，基本語法如下：**
- **DELETE FROM 資料表名稱**
- **WHERE 條件式**

附錄 C

● 研華公司 Adam 控制器詳細指令

系統中所使用的四個 Adam **四個控制指令分別為：**

1.**初始化命令**
2.**回傳** Adam **狀態**
3.**送出開起/關閉指令**
4.**讀取** DO,DI **狀態**

四種指令的內容如下：

1. 初始化命令：%AANNTTCCFF(cr)

AA：**表機器** ID
NN：**表將要設成的新** ID
TT：**固定** 40H
CC：**表傳輸的** Baud Rate
FF：**第六** Bit **表是否有** Checksum,**其餘** Bits **皆保留**

回傳：

!AA(cr)：**表命令正確**
?AA(cr)：**表非法命令**

2. 回傳 Adam 狀態：$AA2(cr)

AA：**表機器** ID

回傳:

!AATTCCFF(cr)：**表命令正確**,**傳回狀態**,**格式同命令** 1
?AA(cr)：**表非法命令**

3. 送出開起/關閉指令：#AABB(data)(cr)

AA：**機器的** ID **碼**

BB：有兩種選擇

1.若為 00,表示設定所有 Channel,後面兩個(Data)代表 8 個 bits

2.若第一個字為 1,表示要設定單一 Channel,下個字表示要設定的 Channel 編號

(Data)：表要設定的資料值

回傳：

>(cr)：表命令正確

?AA(cr)：表非法命令

4. 讀取 DO,DI狀態：$AA6(cr)

AA：機器的 ID 碼

回傳：

!(DataOutput)(DataInput)00(cr)：表命令正確

?AA(cr)：表非法命令

(前兩 Byte 表 8 個 DO 狀態,後兩 Byte 表 7 個 DI 狀態,最後兩個則固定為 0)

(cr)：在此表示 Carriage Return。