

專題名稱：實現在ZIGBEE多點跳躍式網路上進行網路遊戲 (Implementing Multi-Hop ZigBee Networks for OnLine Games)

指導教授：莊岳儒 博士

學生：黃子玓、徐偉晉

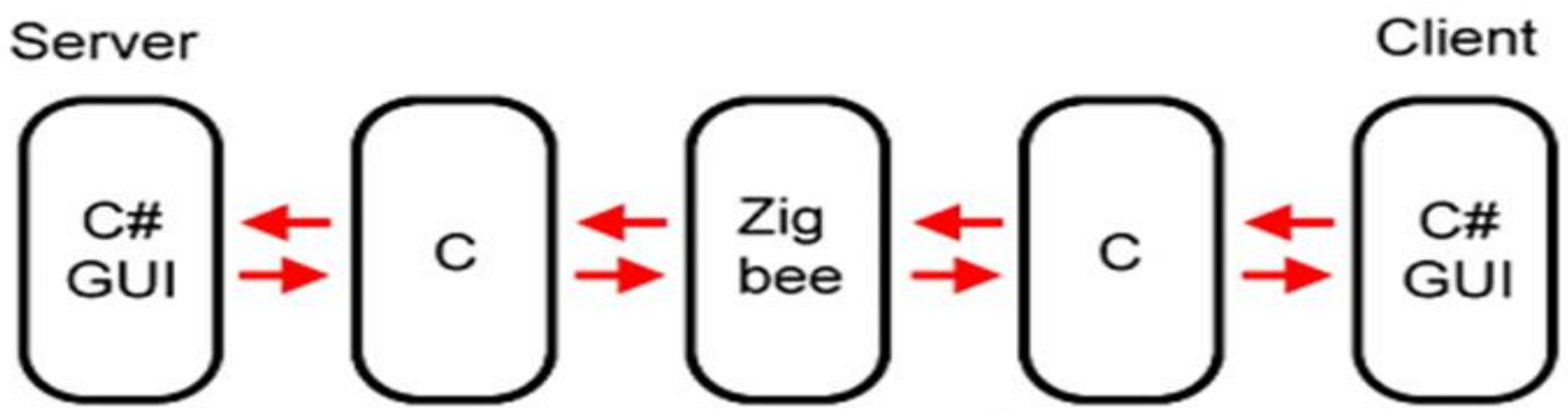
輔仁大學 電機工程學系 大學部專題生

摘要

- 無線網路在現在生活中已經是生活必需品，從4G行動網路到WIFI的使用都已經非常普及。
- 本次專題使用的Zigbee和WIFI一樣是一種無線網路的通訊協定，Zigbee的通訊協定是使用IEEE802.15.4，在2.4GHz的頻段下傳輸，相較於Wifi，Zigbee本身耗能較低、容易建立多節點，但是傳送速度較慢。對於沒有需要傳送大量資料的系統架構來說是一個良好的環境。
- 為了運用建好的網路環境，專題也撰寫了一個象棋的遊戲程式來驗證，也可以練習C#這個物件導向的程式。

系統架構

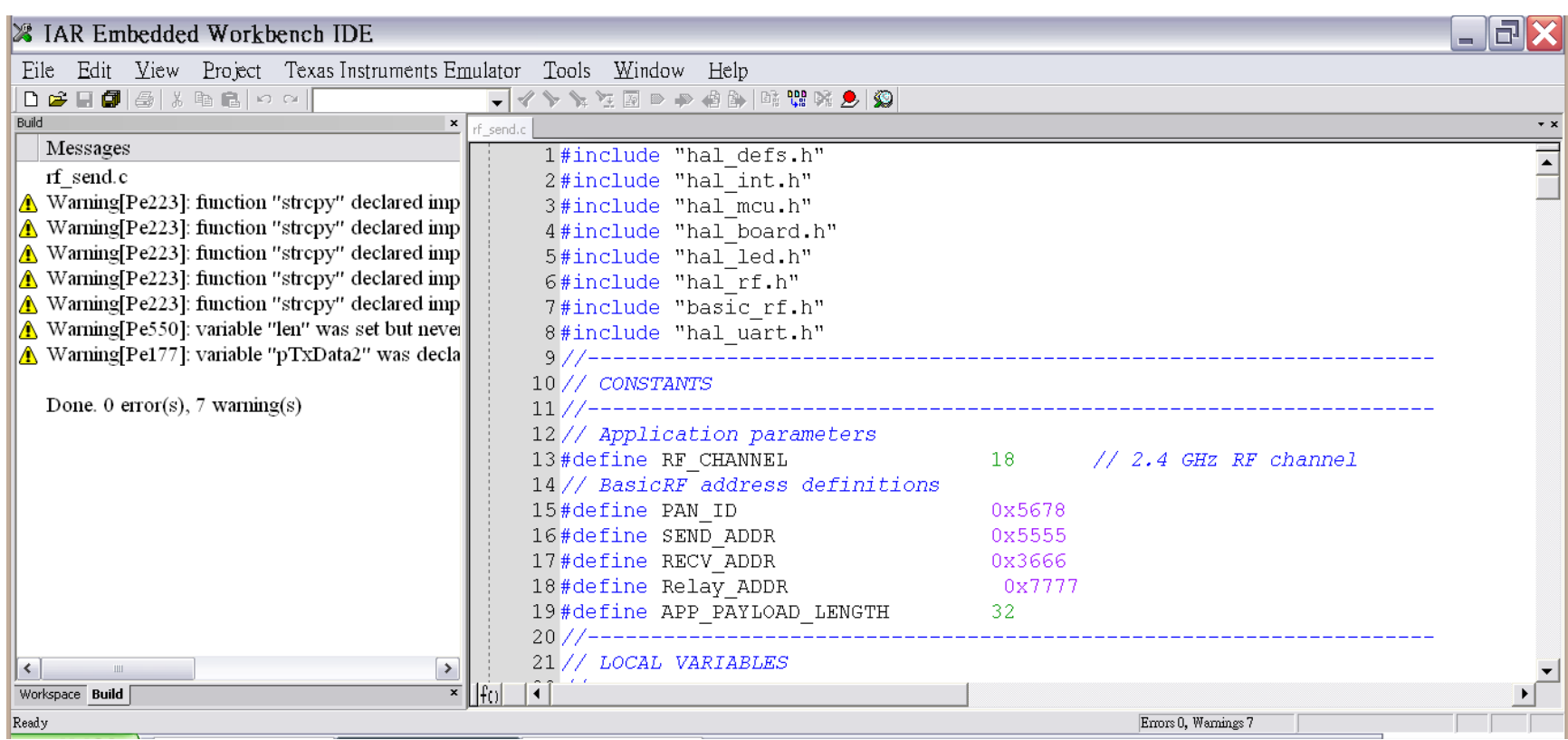
- 本專題的系統架構（圖一），其包括三台Zigbee作為通訊基礎，形成線狀架構，並撰寫程式碼建構Zigbee的通訊環境。然後用C程式作為中繼點，溝通C#圖形化介面遊戲程式與Zigbee網路環境。



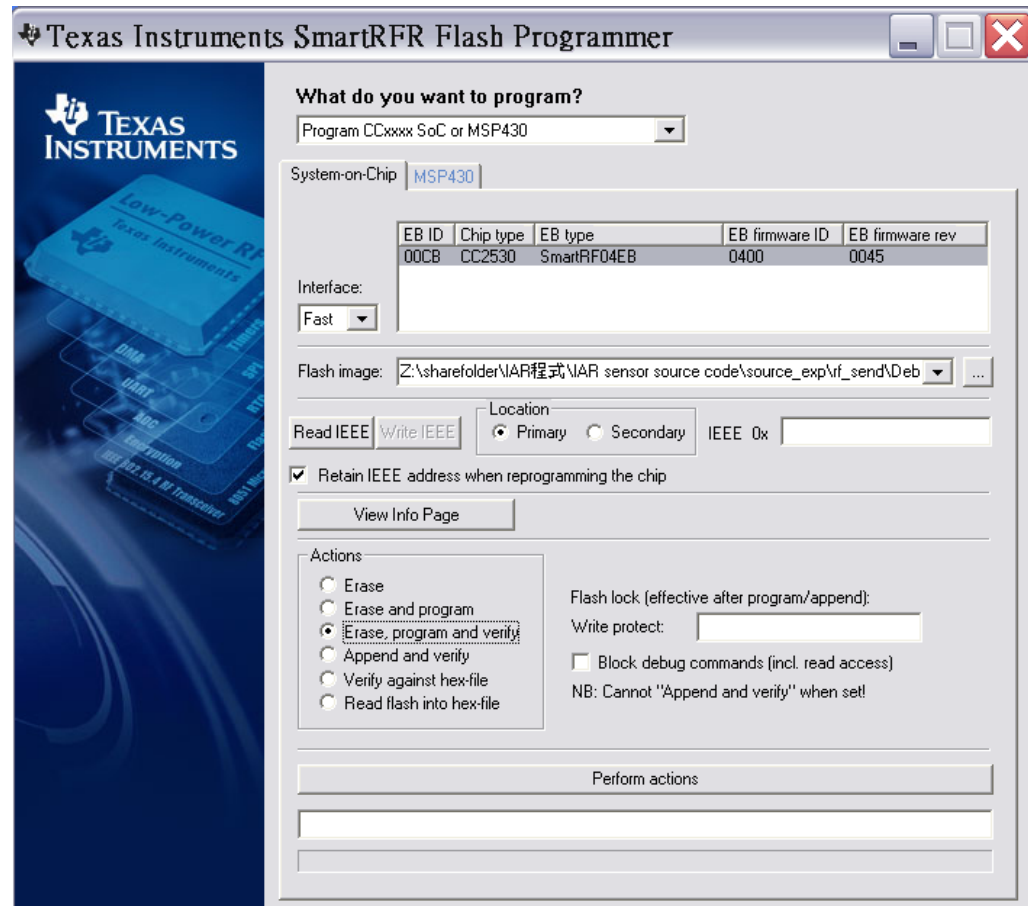
圖一、專題系統架構圖

開發環境

- Zigbee部分使用IAR Embedded workbench IDE(圖二)編譯，Texas Instruments SmartRF Flash Programmer(圖三)燒錄至Zigbee。IAR Embedded workbench IDE是一種嵌入式系統開發程式，可以將程式碼編譯成不同晶片的CPU所執行的機械碼，本次實驗使用的Zigbee(圖四)的晶片是CC2530，由德州儀器製作，板子則是長高科技製作；Texas Instruments SmartRF Flash Programmer是把編譯好的hex機械碼透過Debugger(圖五)燒錄至硬體中。
- C與C#皆使用Microsoft Visual Studio(圖六)來進行開發編譯。雖然是使用同一套程式，但是因為C#是使用圖形化介面進行開發，所以開發的步驟還是有所不同。
- (圖七)是實驗成品。



圖二、Embedded workbench IDE



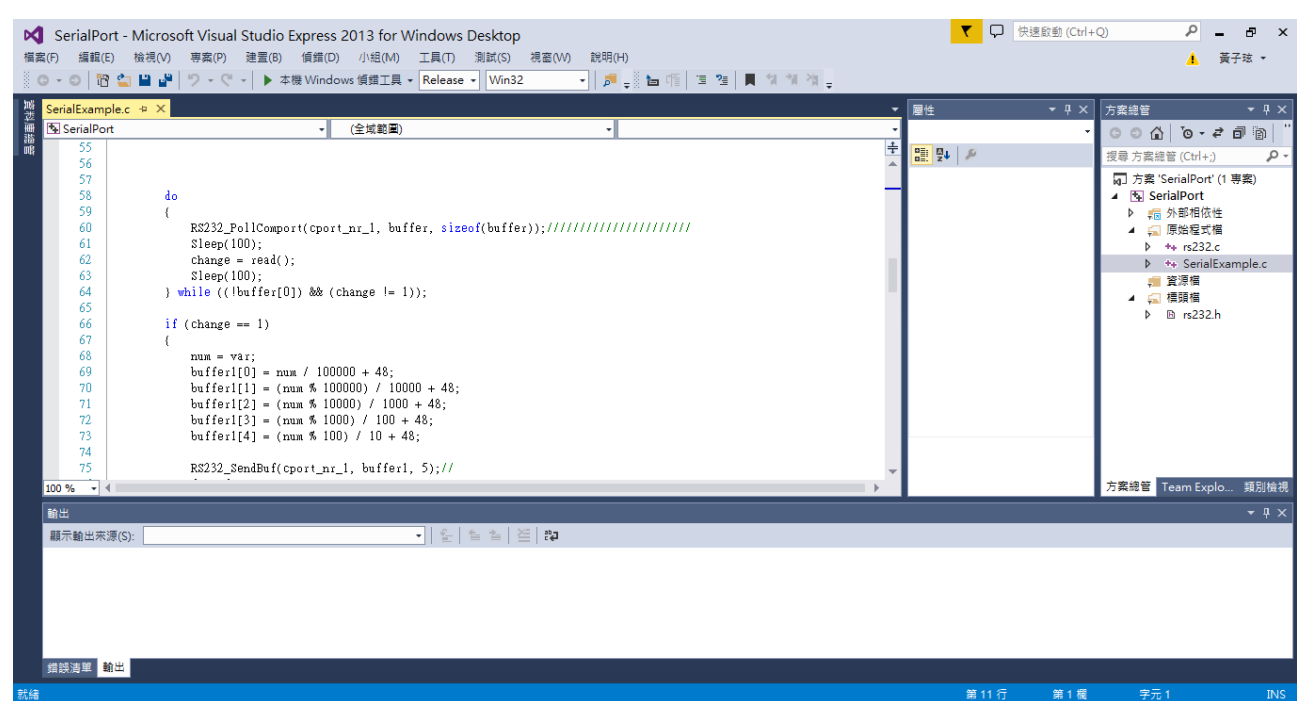
圖三、燒錄軟體



圖四、ZigBee



圖五、DEBUGGER



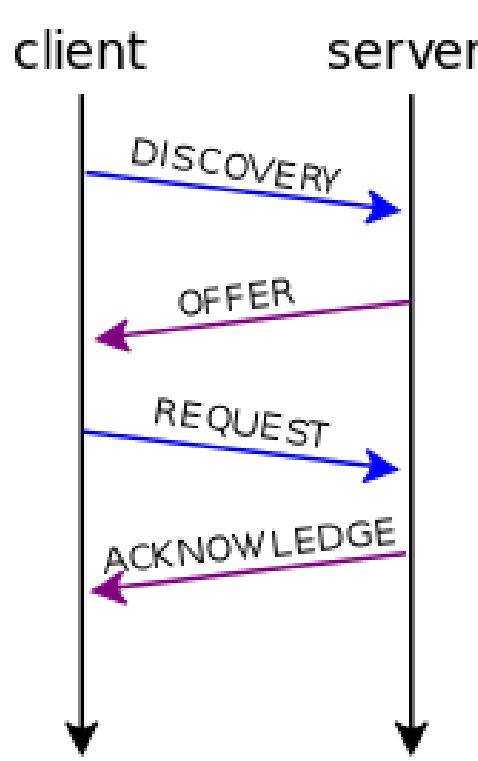
圖六、Microsoft Visual Studio



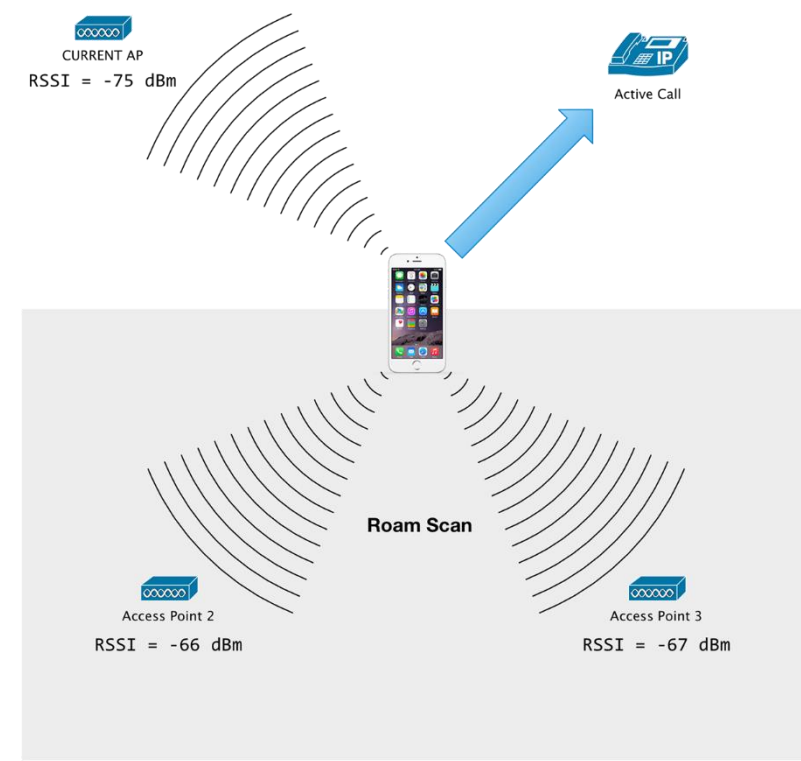
圖七、C#介面圖

實作方法與成果

Zigbee網路部分，一開始使用DHCP(Dynamic Host Configuration Protocol)的動態網路位址分配。因此開機後另外兩台Zigbee(圖八)第一件事情就是和Server端請求IP位址，透過RSSI(Received Signal Strength Indicator)(圖九)得知訊號強度，將訊號強的Zigbee設為Relay端，較弱的設為Client端。回傳後把資料寫入到自己的IP位址，才能開始通訊，這就是整個Zigbee通訊的初始化。



圖八、DHCP

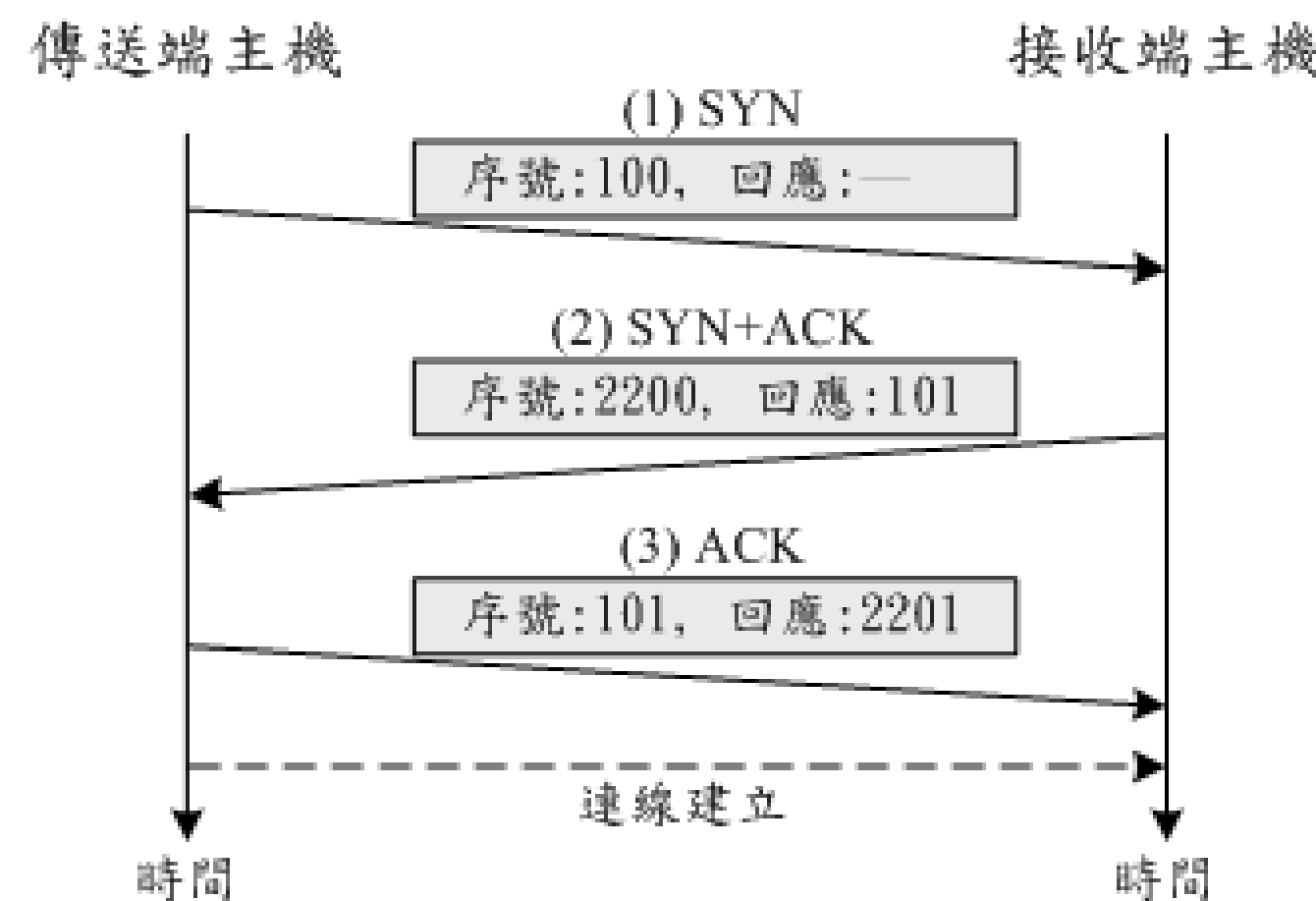


圖九、RSSI

建立連線後，為了防止封包遺失或資料缺失，使用了類似TCP(Transmission Control Protocol)的三向交握(圖十)以及封包序號(Packet sequence number)。

三向交握是當收到資料後必須要像發送端來源進行確認(Acknowledgement)，而且必須要經過兩次雙向確認才能保證資料的正確性，雖然會增加通訊的時間及成本，但為了資料的正確性是必要的。

封包序號是為了防止資料的破損，如果指定的byte封包序號不一致，Zigbee就不會把這筆資料接收。



圖十、TCP三項交握

C在這個專題中只是資料轉送的角色，以及簡單把只需要在Zigbee處理的資料(如IP位址)去除，並把資料存到txt檔，供C#做讀取。

C#是一個象棋程式，剛開啟時會先向Zigbee確認狀態，確定三個都連線時才能開始遊戲。除了正常下棋之外，也有悔棋、和局、認輸的功能。通訊方式是1byte的控制碼配4bytes的資料寫入txt檔，C程式在偵測到資料後就會把他轉送到Zigbee，如此重複的下完一盤棋。



圖十一、實作成品全圖

結論

- 本專題研究並實作了一個Zigbee的網路環境，其中最大的困境是控制整個網路架構，以及就演算程式碼完全正確，封包本身也會不時的出現意外，必須要有許多錯誤偵測的方法才能有穩定的無線通訊環境。
- 在此專題中，我們學習到了許多軟體的操作與程式碼的撰寫，也把計算機網路課堂學到的理論實踐，建構出一個Zigbee網路。
- 我們還可以進一步的將這些架構改得更完善，例如簡化程式、修改連線模式使其達到更穩定、更快速的運作等。
- 未來本專題更可以繼續發展成IOT物聯網的運作基礎，以及將我們的所學擴大到未來職場中。



2016 輔仁大學電機工程學系
大學部專題成果展

