

計畫編號：107 保發-11.1-保-01-06-001(11)

**以 NB-IoT 與免控制點位移量測技術進行
水土保持設施之安全檢測與評估
Safety Evaluation of Soil and Water
Infrastructures using NB-IoT and
Reference-Free Displacement Technology**

執行單位：國立臺灣大學生物環境系統工程學系

執行期間：中華民國 107 年 02 月 07 至中華民國 107 年 12 月 20 日

計畫主持人：廖國偉 副教授

行政院農業委員會水土保持局 編印

中華民國 107 年 12 月

本報告書內容及建議純屬執行單位意見，僅供本局施政參考

以 NB-IoT 與免控制點位移量測技術進行水土保持設施之安全 檢測與評估

摘要

藉由目測的方式評估水保設施的安全性是相對主觀且耗時費力。因此，本計畫以感測元件輔以窄帶物聯網(NB-IoT)提供較客觀並減免檢測人員親赴現場的狀況，可即時提供水保設施初步的安全狀態與功能性。

本計畫結合加速度計、微型處理器與NB-IoT達到資料讀取與回傳的效果。微型處理器使用Arduino板，配合開發Arduino程式讀取加速度計輸出的加速度，將加速度傳給NB-IoT硬體，並架設無線通訊協定，NB-IoT硬體可藉由無線通訊協定將加速度值傳至伺服器(服務端)。無線通訊協定則利用NB-IoT功能的4G網路SIM卡將資料回傳，資料回傳與確認網路連線可使用AT command來達成目的。

取得之加速度資料尚須利用演算法將加速度轉換成位移，以進行水保設施安全度與適用性的評估。目前嘗試的演算法計有梯形積分法、Lee, Hong, Park (2010)所提的重建位移法並自行開發的公式。其中，重建位移法中的時窗的設定為分析的重點，時窗是否相互覆蓋亦為考量的因素。

為評估所提方法的適用性，規劃了三種試驗或驗證。第一種是數值分析，以軟體模擬結構物受振動以後，產生的加速度與SAP2000位移，利用演算法將結構物的加速度轉換成位移，分析其位移與SAP2000位移的趨勢性與數值相似程度是否夠一致，以評估演算法的適用性。第二種是室內試驗與驗證，將加速度計附著於縮小結構在振動臺上振動得到加速度值，先驗證加速度值的合理性後再將加速度值使用加速度轉換位移的演算法得到位移值，並與真實位移比較，重建位移演算法所需的參數值則利用最佳化演算法求得。第三種是室外試驗，將加速度計與儀器安裝在室外，確認可以回傳資料並取得加速度，因為演算的準確性已在前兩個試驗中驗證，此階段的重點在於確認無線通訊協定的建置與適用性。

關鍵字：水保設施、加速度、位移重建、NB-IoT、通訊協定

Safety Evaluation of Soil and Water Infrastructures using NB-IoT and Reference-Free Displacement Technology

Abstract

The safety of water conservation facilities was often assessed by visual inspection, which was subjective, time-consuming and laborious. As a result, the machine measurement and narrow-band Internet of Things(NB-IoT) are selected to resolve the disadvantages. Besides, the proposed method is expected to immediately and accurately obtain the initial safety status and functionality of the water conservation facilities.

This project combines accelerometers, microprocessors and NB-IoT to form a module that is able to acquire data in field and transmit data back to the server. The Arduino board is selected as our microprocessor, cooperating with the Arduino IDE to read the accelerations which are collected by an accelerometer. Arduino board then transmits acceleration records to the NB-IoT hardware. To send the data from NB-IoT hardware to the server, the wireless communication protocol, needs to be set up. Additionally, the 4G network SIM card with NB-IoT function, which is provided by Far EastTone, is needed. The AT command is used to verify the connection status between the NB-IoT hardware and server.

After obtaining the acceleration data, a reference-free algorithm is used to convert the acceleration into displacement. With displacement information, the safety of the water conservation facility can be evaluated. Currently, several algorithms have been implemented and tested including the trapezoidal integral method and the algorithm proposed by Lee, Hong, Park (2010). Important factors such as the size and overlapped period of the time window are also investigated though optimization in the project.

Three experiments are used to assess the applicability of the proposed method. The first one is numerical analysis. The finite element software (i.e., SAP2000) simulates the acceleration and displacement from a vibrating structure. The proposed algorithm is then used to reconstruct the displacement from the measured acceleration. The trend of the reconstructed and simulated displacement is used to verify correctness of the proposed method. The second one is an indoor experiment. The developed accelerometer module

is attached to a cantilevered structure sitting on a shaking table. The rationality of the acceleration value is first verified, then the reconstructed displacement is established, in which a new formula for selecting window is proposed. The last test is similar to the second test but is an outdoor experiment. The focus of this test is to verify the completeness of the wireless communication protocol built by the NTU RSSE Lab. (Reliability, Smart and Sustainable Engineering Laboratory).

Keywords : infrastructures, acceleration, displacement reconstruction, NB-IoT, wireless communication protocol

目次

摘要.....	I
Abstract.....	III
目次.....	V
表次.....	VII
圖次.....	IX
第一章 前言	1-1
第一節 計畫緣起.....	1-1
第二節 擬解決問題.....	1-5
第三節 計畫目標.....	1-6
第四節 工作項目	1-8
第二章 智慧感測器應用與分析	2-1
第一節 加速度計開發與選定.....	2-1
第二節 微型處理器開發與選定.....	2-17
第三節 加速度計與微型處理器 Arduino 板結合	2-29
第三章 物聯網(IoT)的應用與開發	3-1
第一節 物聯網之建置與評比.....	3-1
第二節 NB- IoT 應用	3-7
第三節 NB-IoT 硬體介紹	3-11
第四節 NB-IoT、Arduino 板跟加速度計之整合	3-48
第四章 免控制點演算法之開發	4-1
第一節 文獻回顧.....	4-1
第二節 演算法之選擇.....	4-2
第三節 開發模式參數之最佳化.....	4-5
第五章 重建位移之驗證與試驗結果.....	5-1
第一節 數值分析.....	5-1
第二節 室內縮尺度驗證.....	5-22
第三節 最佳化參數值之數值分析.....	5-35
第四節 室外試驗.....	5-41
第六章 總結	6-1

參考文獻.....	參-1
附錄	附錄-1
附錄一：期末審查會議紀錄暨回覆辦理情形.....	附錄-1
附錄二：MMA7660 加速度計程式碼.....	附錄-6
附錄三：MMA7660 加速度計 cpp 檔.....	附錄-7
附錄四：MMA7660 加速度計 h 檔.....	附錄-12
附錄五：將 SD 卡功能用在 Arduino 上所需指令	附錄-14
附錄六：AT command 驗證通訊訊號是否正常運作.....	附錄-16
附錄七：AT command 架設單鏈路模式.....	附錄-17
附錄八：梯形積分法程式碼範例.....	附錄-19
附錄九：重建位移法(無時窗)程式碼範例	附錄-20
附錄十：重建位移法(有時窗_nonoverlap)程式碼範例	附錄-21
附錄十一：重建位移法(有時窗_overlap)程式碼範例	附錄-23
附錄十二：25Hz 振動頻率下懸臂梁自由端加速度與 SAP2000 位移 ..	附錄-25
附錄十三：25Hz 振動頻率下 SAP2000 位移與其他演算法得出 位移迴歸圖	附錄-30
附錄十四：6.5Hz 振動頻率下 SAP2000 位移與其他演算法得出 位移迴歸圖	附錄-34

表次

表 1-1 花蓮溪流域之須美基溪支流土石災害防治工程之跌水工的安全性評估調查表---基本資料與抗壓強度測定.....	1-3
表 1-2 花蓮溪流域之須美基溪支流土石災害防治工程之跌水工的安全性評估調查表---評估內容.....	1-4
表 2-1 加速度計之種類與介紹	2-4
表 2-2 MMA8451Q 的機械特性.....	2-10
表 2-3 MMA7660FC 在不同模式下的運作.....	2-14
表 2-4 Arduino 與樹莓派之比較.....	2-19
表 2-5 Arduino 的種類與介紹	2-20
表 2-6 Arduino UNO 板與 Arduino M0 Pro 板比較表	2-21
表 3-1 四大無線通訊比較表	3-5
表 3-2 主流 LPWA 比較表.....	3-6
表 3-3 SIMCOM 板 A 介面各腳位介紹	3-17
表 3-4 SIMCOM 板 B 介面各腳位介紹	3-18
表 3-5 SIMCOM 板 C 介面各腳位介紹	3-19
表 3-6 SIMCOM 板 D 介面各腳位介紹	3-20
表 3-7 SIMCOM 板 E 介面各腳位介紹	3-21
表 3-8 初期建立 TCP 與模組時通訊效果的驗證語法與說明	3-28
表 3-9 TCP 與 UDP 比較	3-31
表 3-10 單鏈路模式 TCP 架設所需的 AT command 語法與說明.....	3-31
表 3-11 SIM7000E 耗電參數列表	3-36
表 3-12 XBee S2 模組腳位介紹表	3-47
表 3-13 XBee 無線通訊模組的規格表	3-47
表 5-1 25Hz 振動頻率下懸臂梁自由端 SAP2000 位移與所有試驗演算法得出的位移迴歸分析之相關係數(R^2).....	5-12
表 5-2 6.5Hz 振動頻率下懸臂梁自由端 SAP2000 位移與所有試驗演算法得出的位移迴歸分析之相關係數(R^2).....	5-21
表 5-3 不同演算法下 λ 值比較(時間間隔=0.02).....	5-36
表 5-4 不同演算法下 λ 值比較(時間間隔=0.01).....	5-36

表 5-5 不同演算法下 λ 值比較(時間間隔=0.005).....5-37

圖次

圖 1-1 花蓮須美基溪支流之跌水工	1-2
圖 1-2 本計畫規劃之技術使用關聯圖	1-6
圖 1-3 本計畫規劃之技術架構圖	1-7
圖 2-1 MMA8451Q 加速度計照片	2-2
圖 2-2 MMA7660FC 加速度計照片	2-3
圖 2-3 ADXL335 加速度計照片	2-3
圖 2-4 MMA8451Q Block diagram.....	2-8
圖 2-5 I ² C 時序圖	2-8
圖 2-6 MMA8451Q orientation on perpendicular axis.....	2-9
圖 2-7 MMA8451 Block diagram.....	2-13
圖 2-8 MMA7660FC 的模式與各模式的關係圖.....	2-13
圖 2-9 MMA7660FC orientation on perpendicular axis.....	2-14
圖 2-10 ADXL335 orientation on perpendicular axis.....	2-16
圖 2-11 Arduino UNO 板照片	2-18
圖 2-12 Arduino M0 Pro 板照片	2-18
圖 2-13 Arduino (a) IDE 圖樣 (b)商標	2-22
圖 2-14 Arduino 硬體與軟體使用圖示	2-23
圖 2-15 Arduino UNO 板內容介紹	2-25
圖 2-16 ICSP (In-Circuit Serial Programming) 腳位.....	2-25
圖 2-17 Arduino M0 Pro 版內容介紹	2-28
圖 2-18 MMA7660FC 的 I ² C 連接要位.....	2-29
圖 2-19 MMA7660FC 與 Arduino M0 Pro 連結示意圖	2-30
圖 2-20 MMA7660FC 與 Arduino M0 Pro 連接圖	2-30
圖 2-21 Arduino 板與電腦連接後序列埠顯示與確認之圖示.....	2-32
圖 2-22 Arduino 軟體寫程式前確認序列埠與開發板之選擇示意圖.....	2-33
圖 2-23 將 C 槽的 Arduino 資料夾裡的 library 放入 cpp*檔與 h*檔之圖示 ...	2-33
圖 2-24 cpp*檔與 h*檔放置在 Arduino 檔旁邊以便顯示程式碼之圖示	2-34
圖 2-25 cpp*檔與 h*檔可顯示其程式碼之圖示	2-34
圖 2-26 程式完成後之驗證與上傳資料軟件	2-35

圖 2-27 點選序列埠監控視窗看到加速度值之圖示	2-36
圖 2-28 Arduino UNO 板與 MMA8451Q 結合示意圖	2-39
圖 2-29 讀卡機開發模組.....	2-39
圖 3-1 遠傳公司在 IoT 的應用案例	3-3
圖 3-2 IoT 市場分類與競爭技術	3-3
圖 3-3 LPWA 規模預測.....	3-4
圖 3-4 2017 年全球 NB-IoT 部署情況	3-4
圖 3-5 4G 網路覆蓋圖與說明	3-8
圖 3-6 遠傳的行動通訊發展.....	3-9
圖 3-7 未來 NB-IoT 訊號覆蓋圖	3-9
圖 3-8 網路拓撲組合方式.....	3-10
圖 3-9 芯訊通無線科技（上海）有限公司發展的 NB-IoT 模組列項.....	3-11
圖 3-10 SIMCOM 板與其配件	3-13
圖 3-11 SIMCOM 板之構造(正面)	3-15
圖 3-12 SIMCOM 板之構造(背面)	3-16
圖 3-13 SIM7000-TE 板(正面).....	3-22
圖 3-14 SIM7000-TE 板(背面).....	3-23
圖 3-15 SIM7000E 完整模組	3-26
圖 3-16 SIM7000E 模組啟動的狀態	3-27
圖 3-17 SIM7000E 連接電腦後產生出的序列埠	3-28
圖 3-18 TCP 服務端的 IP 與端口架設	3-32
圖 3-19 服務器主機藉由無線通訊收到別的位址所給的數值	3-33
圖 3-20 SIM7000E 之 Arduino 擴展板天線.....	3-37
圖 3-21 SIM7000E 之 Arduino 擴展板與 Arduino 板之腳位比較.....	3-38
圖 3-22 SIM7000E 之 Arduino 擴展板天線.....	3-38
圖 3-23 SIM7000E 之 Arduino 擴展板與 Arduino 板連接圖	3-39
圖 3-24 模組連接充電器與 USB 連接孔.....	3-39
圖 3-25 SIM7000E 模組啟動的狀態	3-41
圖 3-26 Node.js 設置好服務端時的結果.....	3-42
圖 3-27 Arduino IDE 介面設定與服務端的 IP 與序列埠相同.....	3-43

圖 3-28 Arduino 程式碼傳送到模組上的運作情況.....	3-43
圖 3-29 Node.js 接收到資料.....	3-44
圖 3-30 XBee S2 模組外觀	3-45
圖 3-31 XBee S2 模組腳位示意圖	3-45
圖 3-32 Arduino UNO 與 XBee 連接	3-46
圖 3-33 加速度計與 Arduino 與 SIM7000E 的結合.....	3-50
圖 3-34 Arduino 板與 SIM7000E 連接示意圖	3-50
圖 3-35 Arduino 板與 SIM7000E 連通證明	3-51
圖 3-36 SIM7000E 之類比訊號腳位	3-52
圖 3-37 Arduino UNO 與 MMA8451Q 加速度計跟 XBee 結合示意圖	3-54
圖 3-38 X-CTU 設定步驟 1.....	3-54
圖 3-39 X-CTU 設定步驟 2.....	3-55
圖 3-40 X-CTU 設定步驟 3.....	3-55
圖 3-41 XBee 發送端測試	3-56
圖 3-42 XBee 終端測試	3-56
圖 4-1 時窗之間未重疊情況下的位移重建過程	4-4
圖 4-2 時窗之間重疊情況下的位移重建過程	4-4
圖 4-3 粒子移動示意圖.....	4-6
圖 4-4 粒子群演算法流程圖.....	4-7
圖 5-1 SAP2000 軟體圖樣	5-2
圖 5-2 數值分析模擬之懸臂梁	5-2
圖 5-3 數值分析模擬之有限元素的懸臂梁強迫振動狀態 (a)為懸臂梁的強迫振動前(b)為懸臂梁的第一振態 (c)為懸臂梁的第二振態(d)為懸臂梁的第三振態	5-2
圖 5-4 梯形積分法示意圖.....	5-4
圖 5-5 數值分析模擬之懸臂梁在 25Hz 振動頻率下自由端時間與加速度量 關係圖	5-5
圖 5-6 數值分析模擬之懸臂梁在 25Hz 振動頻率下自由端時間與 SAP2000 位移量關係圖	5-5
圖 5-7 25Hz 振動頻率下 SAP2000 位移與梯形積分法得到位移之關係圖.....	5-6

圖 5-8 25Hz 振動頻率下 SAP2000 位移與重建位移法(無時窗)得到位移之 關係圖	5-7
圖 5-9 25Hz 振動頻率下 SAP2000 位移與重建位移法(有時窗，non-overlapped) 得到位移之關係圖(時窗有 20 個，時窗大小為 0.5 秒).....	5-8
圖 5-10 25Hz 振動頻率下 SAP2000 位移與重建位移法(有時窗，non-overlapped) 得到位移之關係圖(時窗有 10 個，時窗大小為 1 秒).....	5-9
圖 5-11 25Hz 振動頻率下 SAP2000 位移與重建位移法(有時窗，non-overlapped) 得到位移之關係圖(時窗有 5 個，時窗大小 2 秒).....	5-9
圖 5-12 25Hz 振動頻率下 SAP2000 位移與重建位移法(有時窗，overlapped) 得到位移之關係圖(時窗大小為 0.1 秒).....	5-10
圖 5-13 25Hz 振動頻率下 SAP2000 位移與重建位移法(有時窗，overlapped) 得到位移之關係圖(時窗大小為 2 秒).....	5-11
圖 5-14 25Hz 振動頻率下 SAP2000 位移與重建位移法(有時窗，overlapped) 得到移之關係圖(時窗大小為 5 秒).....	5-11
圖 5-15 數值分析模擬之懸臂梁在 6.5Hz 振動頻率下自由端時間與加速 度量關係圖	5-14
圖 5-16 數值分析模擬之懸臂梁在 6.5Hz 振動頻率下自由端時間與 SAP2000 位移量關係圖	5-14
圖 5-17 6.5Hz 振動頻率下 SAP2000 位移與梯形積分法得到位移之關係圖.....	5-15
圖 5-18 6.5Hz 振動頻率下 SAP2000 位移與重建位移法(無時窗)得到位移之 關係圖	5-16
圖 5-19 6.5Hz 振動頻率下 SAP2000 位移與重建位移法(有時窗，non-overlapped) 得到位移之關係圖(時窗有 20 個，時窗大小為 0.5 秒).....	5-17
圖 5-20 6.5Hz 振動頻率下 SAP2000 位移與重建位移法(有時窗，non-overlapped) 得到位移之關係圖(時窗有 10 個，時窗大小為 1 秒).....	5-18
圖 5-21 6.5Hz 振動頻率下 SAP2000 位移與重建位移法(有時窗，non-overlapped) 得到位移之關係圖(時窗有 5 個，時窗大小 2 秒).....	5-18
圖 5-22 6.5Hz 振動頻率下 SAP2000 位移與重建位移法(有時窗，overlapped) 得到位移之關係圖(時窗大小為 0.5 秒).....	5-19
圖 5-23 6.5Hz 振動頻率下 SAP2000 位移與重建位移法(有時窗，overlapped) 得到位移之關係圖(時窗大小為 1 秒).....	5-20
圖 5-24 6.5Hz 振動頻率下 SAP2000 位移與重建位移法(有時窗，overlapped) 得到位移之關係圖(時窗大小為 2.5 秒).....	5-20
圖 5-25 室內模擬之懸臂梁模型	5-22

圖 5-26 振動臺.....	5-22
圖 5-27 FFT 副程式與 Matlab 函式比較圖	5-24
圖 5-28 主頻驗證.....	5-25
圖 5-29 振動臺試驗實照.....	5-26
圖 5-30 1 Hz 輸入正弦波形.....	5-26
圖 5-31 FFT 運算輸出頻譜.....	5-27
圖 5-32 1Hz 振動頻率下真實位移與梯形積分法得到位移之關係圖	5-28
圖 5-33 1Hz 振動頻率下真實位移與重建位移法(無時窗)得到位移之關係圖 ...	5-29
圖 5-34 1Hz 振動頻率下真實位移與重建位移法(有時窗，non-overlapped) 得到位移之關係圖(時窗有 5 個，時窗大小為 4 秒).....	5-30
圖 5-35 1Hz 振動頻率下真實位移與重建位移法(有時窗，non-overlapped) 得到位移之關係圖(時窗有 10 個，時窗大小為 2 秒).....	5-31
圖 5-36 1Hz 振動頻率下真實位移與重建位移法(有時窗，non-overlapped) 得到位移之關係圖(時窗有 20 個，時窗大小 1 秒).....	5-31
圖 5-37 1Hz 振動頻率下真實位移與重建位移法(有時窗，overlapped) 得到位移之關係圖(時窗有 5 個，時窗大小為 4 秒).....	5-32
圖 5-38 1Hz 振動頻率下真實位移與重建位移法(有時窗，overlapped)得 到位移之關係圖(時窗有 10 個，時窗大小為 2 秒).....	5-33
圖 5-39 1Hz 振動頻率下真實位移與重建位移法(有時窗，overlapped)得 到移之關係圖(時窗有 20 個，時窗大小 1 秒).....	5-33
圖 5-40 PSO 兩種演算法之振動能量與頻率.....	5-37
圖 5-41 4Hz 結構的振動頻率下真實位移與重建位移(PSO1)之比較.....	5-38
圖 5-42 8Hz 結構的振動頻率下真實位移與重建位移(PSO1)之比較.....	5-38
圖 5-43 4Hz 結構的振動頻率下真實位移與重建位移(PSO2)之比較.....	5-39
圖 5-44 8Hz 結構的振動頻率下真實位移與重建位移(PSO2)之比較.....	5-39
圖 5-45 時間間隔為 0.02 時 PSO2 演算法之時窗與頻率關係圖.....	5-40
圖 5-46 時間間隔為 0.01 時 PSO2 演算法之時窗與頻率關係圖.....	5-40
圖 5-47 時間間隔為 0.005 時 PSO2 演算法之時窗與頻率關係圖.....	5-40
圖 5-48 機車電瓶將電源供應 Arduino UNO 板.....	5-42
圖 6-1 甘梯圖.....	6-2

第一章 前言

第一節 計畫緣起

本計畫目的在於建立一套基於感應器的水保設施初步評估系統。近年來，因為微機電系統(Micro-electromechanical Systems, MEMS)的蓬勃發展，各式價格平易近人的感應器迅速發展，因為主觀的人為目視檢測恐影響評估結果，或過於耗費人力，本計畫遂計畫探討採用基於感測元件評估方法的可行性。本計畫感測元件的資料收集係利用窄帶物聯網NB-IoT(Narrow Band Internet of Things)回傳至遠端的伺服器以提供後續的運算。在眾多初步評估指標中，位移量具有明確的物理意義，故選定為本計畫的指標。然而，位移感應器製作相對而言較為複雜，成本較高，因此，本計畫改採用加速度計作為感測元件，依所量測的加速度值並開發免控制位移量測演算法進行水保設施的安全評估。

以往是藉由目測的方式來評估水保設施的安全性，以花蓮的花蓮溪流域之須美基溪支流的跌水工(圖1-1)的安全評估流程為例，表1-1與表1-2所示為花蓮的花蓮溪流域之須美基溪支流土石災害防治工程之跌水工的安全性評估調查表相關項目做評估。從表1-1可以發現，雖然評估內容中包含藉由重錘試驗而得之客觀的抗壓強度(試驗十次)，然而在表1-2的耐久性評估部份，特別在第二階段目視檢測的評估方式是相當主觀的(包含權重值的大小)。此外，上述方式需要水保人員親赴現場進行檢測，相當耗時費力，感測元件與IoT技術的結合，提供一個更有效率的水保設施安全性評估與評估方式。

案例編號	HL-1	照片編號	HL-1-2
現況簡述	右側護岸淘空		



圖 1-1 花蓮須美基溪支流之跌水工

資料來源：(楊凱傑，2012)

表 1-1 花蓮河流域之須美基溪支流土石災害防治工程之跌水工的安全性評估調

查表---基本資料與抗壓強度測定

一、基本資料

工程名稱：須美基溪支流土石災害防治工程 檢測者：楊凱傑 檢測日期：7/4

基本資料											
工程地點	花蓮縣(市) 秀林鄉鎮(區) 佳民村(里)										
TWD67 座標	X：308671					Y：2660881					
集水區名稱	花蓮河流域										
溪流名稱	須美基溪										
計畫別	易淹水地區水患治理計畫										
執行單位	花蓮分局保育推廣課										
竣工年度	98										
備註	土石流潛勢溪流編號 DF015										
檢測內容											
構造物型式	☐ 防砂壩 ☐ 梳子壩 ☐ 固床工 ☐ 潛壩 ☐ 整流工程 ☒ 跌水工 ☐ 擋土牆 ☐ 板橋 ☐ 護岸 ☐ 排水設施										
構造物尺寸	高：5 公尺 寬：3 公尺 長：30 公尺 其它：										
混凝土衝錘試驗 抗壓強度 f'_c	推估強度：1. 236.6 kg/cm^2 2. 249.4 kg/cm^2 3. 262.9 kg/cm^2 平均強度：249.6 kg/cm^2										
試驗次數與結果	1	2	3	4	5	6	7	8	9	10	平均值
檢測位編號 NO. 1	28	32	29	29	27	28	29	30	30	29	29.1
檢測位編號 NO. 2	34	32	32	30	31	29	32	31	28	30	30.9
檢測位編號 NO. 3	30	32	30	30	31	32	33	30	35	34	31.7
是否進階檢測	☐ 是 (裂縫寬度大於 0.3mm 或構造物現況有疑慮) ☒ 否										
備註											
進階儀器檢測											
檢測日期	無										
檢測單位	無										
裂縫深度	0 mm										
備註	無進階檢測										
耐久性評估											
評估結果	☐ 較佳 (70-100) ☒ 中等 (35-69) ☐ 較差 (0-34)										
備註	有淘空現象										

資料來源：(楊凱傑，2012)

表 1-2 花蓮河流域之須美基溪支流土石災害防治工程之跌水工的安全性評估調查表---評估內容

查表---評估內容

三、評估內容

耐久性評估內容表第一階段					
圖說文件記錄					
設計	設計圖說是否完整	☐ 有相關設計圖說 ☒ 無			
材料	水泥	☐ 有使用特殊水泥 ☒ 一般水泥			
	是否有材料檢驗紀錄	☐ 有 ☒ 無			
施工	表面處理	☒ 有進行表面處理 ☐ 無			
	驗收紀錄是否完整	☐ 有 ☒ 無			
耐久性評估內容表第二階段					
項 目		權重	評估準則	配分	
環境現況	崩塌地	10	☒ 非潛勢區(1.0) ☐ 低(0.7) ☐ 中(0.4) ☐ 高(0.1)	10	
	土石流潛勢溪流	10	☐ 非潛勢區(1.0) ☐ 低(0.7) ☒ 中(0.4) ☐ 高(0.1)	4	
	河道淘刷	6	☐ 非潛勢區(1.0) ☐ 低(0.7) ☒ 中(0.4) ☐ 高(0.1)	2.4	
	土砂(石)淤積	5	☐ 非潛勢區(1.0) ☐ 低(0.7) ☐ 中(0.4) ☒ 高(0.1)	0.5	
現況目視檢測	D 程度	磨蝕	4	☐ 無此狀態(1.0) ☐ 輕微(0.7) ☐ 中等(0.4) ☒ 嚴重(0.1)	0.4
		剝落	4	☐ 無此狀態(1.0) ☐ 輕微(0.7) ☐ 中等(0.4) ☒ 嚴重(0.1)	0.4
		位移或沉陷	8	☐ 無此狀態(1.0) ☒ 輕微(0.7) ☐ 中等(0.4) ☐ 嚴重(0.1)	5.6
		背填材流失	6	☐ 無此狀態(1.0) ☐ 輕微(0.7) ☒ 中等(0.4) ☐ 嚴重(0.1)	2.4
	R 影響性	磨蝕	4	☐ 無此狀態(1.0) ☐ 輕微(0.7) ☐ 中等(0.4) ☒ 嚴重(0.1)	0.4
		剝落	4	☐ 無此狀態(1.0) ☐ 輕微(0.7) ☐ 中等(0.4) ☒ 嚴重(0.1)	0.4
		位移或沉陷	8	☐ 無此狀態(1.0) ☒ 輕微(0.7) ☐ 中等(0.4) ☐ 嚴重(0.1)	5.6
		背填材流失	6	☐ 無此狀態(1.0) ☐ 輕微(0.7) ☒ 中等(0.4) ☐ 嚴重(0.1)	2.4
非破壞性檢測	初 階	裂縫	4	是否超過3mm ☐ 是(0.1) ☒ 否(1.0)	4
		淘空	6	☐ 無此狀態(1.0) ☐ 輕微(0.7) ☒ 中等(0.4) ☐ 嚴重(0.1)	2.4
		銜鑊試驗	6	☒ 平均強度達到設計強度 (1.0) ☐ 平均強度未達到設計強度(0.1)	6
	進 階	*裂縫	5	☒ 無檢測(1.0) ☐ 非結構性裂縫深度未達保護層 (0.7) ☐ 非結構性裂縫深度達保護層(0.4) ☐ 結構性裂縫 (0.1)	5
		*掏空	4	☐ 無超過結構基礎深度(1.0) ☒ 有超過結構基礎深度(0.1)	0.4
		(評分方式=評估項目之權重 X 評估內容之權重)			合計
耐久程度	中度耐久(35-60)	檢測週期	應進行定期檢測	維護模式	預防式維護

資料來源：(楊凱傑，2012)

第二節 擬解決問題

為達到減少災害的發生或降低災害的規模，並提升土地利用價值，必須確保水土保持工程設施的安全性與功能性。基於生命週期理念，國內工程不再重興建、輕維護，所以水保局將對治山防洪構造物提出巡查表單(如表1-1與表1-2)，目的在於平時即可掌握水保設施的安全性與功能性。另如林科里 (2004) 與謝秉蓁 (2013) 亦利用巡查表單對水土保持工程構造物進行檢驗、研究與分析。通常，巡查表單屬於初步的評估，除了評估結果須達到一定的準確性外，尚須在短時間內完成。例如，潘乃欣等人 (2012)執行的行政院農業委員會水土保持局的計畫中，為提供安全評估與修復補強方式的依據，提出水保設施現地檢測的DER評分標準。上述DER評分標準，包含許多項目，約略可區分為室內與室外項目。室內項目係指不須至現場即可完成評分，例如設計圖說是否完整、是否位於土石流潛勢溪流區域等。室外項目則須親赴現場始得完成，例如設施材料之磨損、剝落程度、裂縫寬度以及位移大小等項目。為了增加效率，上述評估項目大多可藉由目視檢測完成，客觀性略嫌不足。而且因為需赴現場方可完成全部的評估流程，即時性亦還有待加強。

為避免人為的誤差，以機器量測是另一個選擇；為避免親赴現場而耽擱時程，IoT提供可能性。雖然上述DER評分標準包含許多項目，然而，位移通常為性能設計的最重要指標之一，對於初步評估而言，為一個理想的量測單位。基於上述，即時且準確地掌握水保設施初步的安全狀態與功能性，在今日全生命週期維護管理的思維下，扮演重要的角色。本計畫針對上述的需求，選取位移作為初步的安全評估的依據並結合現代科技(NB-IoT)與免控制點位移量測技術，計算水保設施即時的位移資訊，提供主管單位人員維護管理補強時之參考。

第三節 計畫目標

利用IoT與免控制點位移量測技術提供重要水土保持設施之即時位移資訊，作為初步且客觀的安全評估之依據，提供給主管單位人員維護管理補強時之參考。技術的部分如圖1-2，希望藉由窄帶物聯網的技術與硬體連接微型處理器(Arduino板、加速度計)與免控制點演算法做到智慧感應、智慧網路、即時監測的技術，達成水土保持設施安全性的評估與檢測。最後呈現的架構如圖1-3，預計會在水保設施上安裝加速度計配上Arduino板與NB-IoT硬體，然後將加速度資料回傳到基地臺並且傳至主機電腦，最後再利用主機電腦上設計好的演算法將加速度轉換成位移值，作為安全度評估的參考與分析。

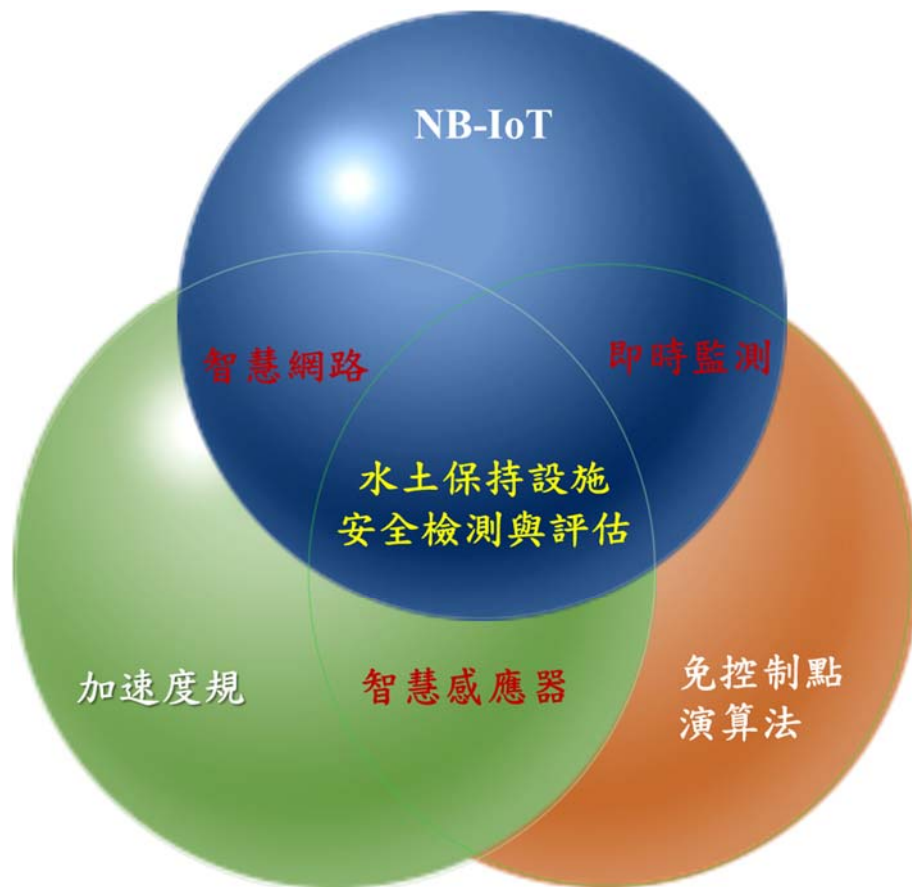


圖 1-2 本計畫規劃之技術使用關聯圖

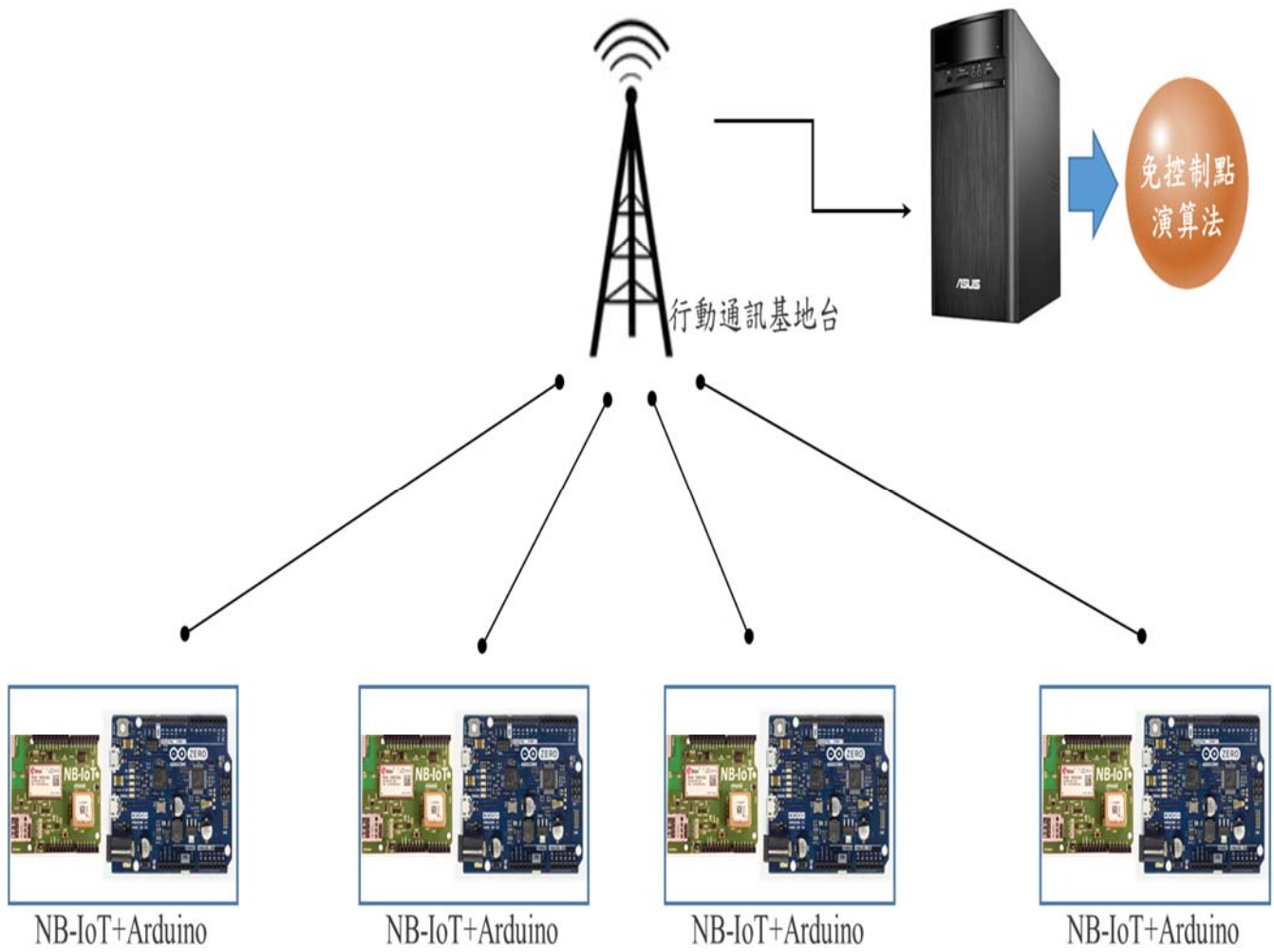


圖 1-3 本計畫規劃之技術架構圖

第四節 工作項目

本計畫工作項目有下列幾項：

一、智慧感測器應用與分析

- (一)加速度計開發與選定：搜尋各式各樣的加速度計種類與了解其效能之後選定適合的加速度計來使用與實驗。
- (二)微型處理器開發與選定：藉由Arduino軟體可設計各種微處理器和控制器的應用來應用在我們的計畫，並從多數的Arduino板選定適合的板子來做試驗。
- (三)加速度計與微型處理器結合：選定好加速度計與微型處理器後，依照其接孔性質做連接，最後在Arduino開發環境撰寫程式碼，以取得加速度計所輸出的加速度值以便做評估指標。

二、物聯網(IoT)的應用與開發

- (一)物聯網(IoT)之建置與評比：了解物聯網的現今發展與應用，評估本計畫可以使用的物聯網種類與模組並做開發，特別是無線網路通訊的部分，是本計畫使用物聯網的重點。
- (二)NB-IoT與Arduino板跟加速度計之結合：當物聯網硬體與通訊試驗沒問題後，將要把物聯網硬體結合智慧感測器硬體，期望智慧感測器硬體將資料傳輸至物聯網硬體並在藉由物聯網硬體的無線網路通訊效果將資料回傳。

三、免控制點演算法應用與分析

- (一)文獻回顧：了解前人對免控制點演算法的開發與應用，可以從前人想法延伸出適合我們的免控制點演算法。
- (二)演算法之選擇：針對免控制點演算法的使用，將從各式各樣的演算法中去選擇最適當的來使用加速度與位移並作分析。
- (三)開發模式參數之最佳化：從演算法得到的加速度與位移值將使用一些最佳化演算法去做演算，分析是否是最恰當的免控制點演算法。

四、實驗數據之驗證與分析

- (一)重建位移之驗證之數值分析：藉由軟體測驗結構振動，結構會產生加速度值與位移值，利用免控制點演算法將加速度值轉變成位移，與原先產生的位移做比較，驗證其演算法的適用性。
- (二)重建位移之驗證之室內縮尺度試驗：利用本研究團隊做出來的結構物與振動模型使結構物振動，將智慧感測器產生的加速度值加以分析，並驗證在該振動頻率下，輸出的加速度值合理性。
- (三)重建位移之驗證之室外水保設施試驗：將會進行室外測試，確保加速度值可以藉由NB-IoT從戶外將資訊傳回室內的主機伺服器。

第二章 智慧感測器應用與分析

本章首先說明材料、軟硬體之評估與選定過程，以利後續計畫之執行，具體而言，本章將針對智慧感測器如加速度計與Arduino板進行介紹與評估，硬體部分如NB-IoT模板，將在之後的章節會做介紹。

第一節 加速度計開發與選定

加速度計是一項測量物體自身運動的工具，通常是用來量測該物體的振動程度，並輸出加速度值。三軸加速度計依輸出訊號之形式可以分成兩種，為數位訊號輸出(Digital)以及類比訊號(Analog)輸出。數位訊號輸出內部已有包含ADC之功能，ADC是數位類比轉換器，可以將連續的類比訊號轉換成數位訊號，也因此能將測得原始數據進一步直接換算成加速度值，而且供應給加速度計的電壓越穩定，加速度計輸出值也將會因此更精確；類比訊號輸出則需連接電壓量測設備來測得其電壓輸出，並進一步量化後才得到數值。從表2-1可以發現到MMA8451Q與MMA7660FC兩種加速度計皆是屬於數位訊號的輸出方式，CXL10GP3、832M1-0025、CXL04GP3、ADXL354BEZ皆是屬於類比訊號的輸出方式。而後續的試驗所使用的加速度計有數位訊號以及類比訊號的產生，各加速度計的詳細內容將會在持續介紹。

加速度計於微機電系統(MEMS)在90代的重大突破，減低自體雜訊比，因其價格低且穩定性高，被大量應用於各式開發案之中。蔣鎧宇 (2009)的研究中有提到早期的應用多在軍事飛彈或導航，近年來引進不少電子產品上，像是互動性高的電視遊樂器Wii也是靠加速度計感測人體移動來進行遊戲。加速度計的應用像是田中尚文、園田茂、村岡慶裕、富田豐、千野直一 (1996)與新井智之、柴喜崇、渡辺修一郎、柴田博 (2011)皆利用加速度計監測人體的行動是否是健康的表現，賴聖恩 (2010)與彭敬鈞 (2017)的研究皆將加速度計裝置在建築結構上檢測振動程度，過去本研究團隊也完成交通部的【創新快速橋梁振動頻率量測方法與人工智慧安全評估模式之研發】 (2016)，根據現地加速度計(MMA8451)之量測，進行快速傅立葉轉換(FFT, Fast Fourier Transform)輔以側推分析 (Puchover analysis)所得之崩塌地表加速度值(Collapse peak ground acceleration)，利用ZigBee結合

Arduino將資料傳輸至設置於橋梁入口處之蜂鳴器，提醒用路人目前橋梁的安全狀況，但先前雖以廣泛地在各項工程使用加速度計，應用加速度計在水土保持工程設施上的案例相對較少，所以本研究計畫將加速度計安裝在水保設施上以做為檢測之功用。

微機電型加速度計在高頻信號的接收上有其限制存在，但根據推測，水土保持設施的顯著週期在20Hz以上應極為罕見，且距離20Hz與所謂的高頻仍有段距離，因此，具有開發的價值。目前市售的微型加速度種類繁多，如Sparkfun ADXL335、ITEM-29、SANLIEN 00EA-SL、MicroStrain G-Link series與MMA8451Q，加速度計種類通常有一軸、二軸、三軸，而因為我們是要針對水保設施的三軸的位移進行監測，所以需選定三軸的加速度計的種類來做實測。根據本研究團隊經驗，MMA8451Q(圖2-1)具有最高的解析度，目前也同時針對MMA7660FC(圖2-2)以及ADXL-335(圖2-3)進行開發因普及性較高也較便宜，並如表2-1所示，可以發現到MMA8451Q加速度量測範圍為 $\pm 2g$ 、MMA7660FC加速度 $\pm 1.5g$ 以及ADXL加速度為 $\pm 3g$ ，是最接近現實的重力加速度。故這幾項加速度計來試驗是相對比較適合的，且這三個加速度計目前在臺灣也較方便取得。加速度選定後，需要連接到微處理器(Arduino板)才可以取得加速度的歷時資料並進行訊號分析。

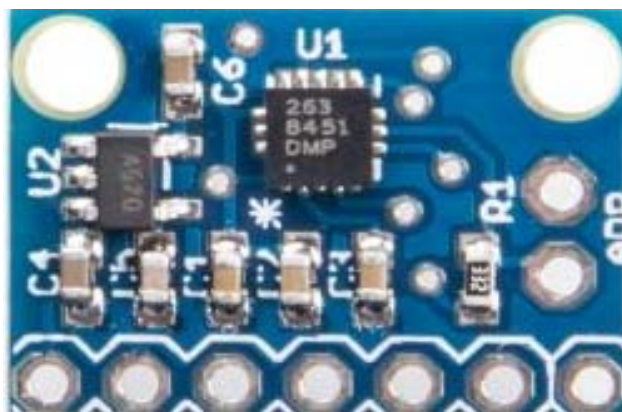


圖2-1 MMA8451Q加速度計照片

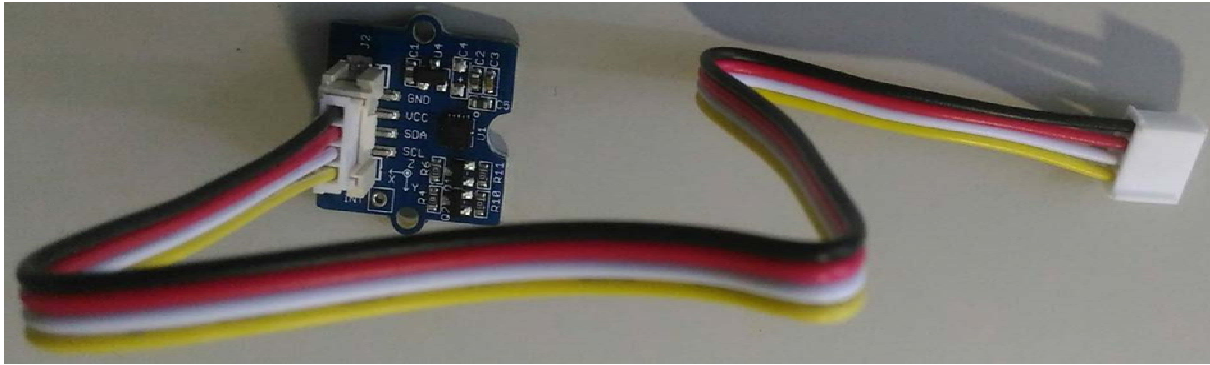


圖2-2 MMA7660FC加速度計照片



圖2-3 ADXL335加速度計照片

表2-1 加速度計之種類與介紹

	MMA8451Q	MMA7660FC	CXL10GP3	832M1-0025	CXL04GP3	ADXL335BEZ
感應器種類	3-axis	3-axis	3-axis	3-axis	3-axis	3-axis
感應方向	X, Y, Z	X, Y, Z	X, Y, Z	X, Y, Z	X, Y, Z	X, Y, Z
加速度	2 g, 4 g, 6 g	1.5 g	10 g	25 g	4 g	3 g
資料輸出種類	Digital	Digital	Analog	Analog	Analog	Analog
提供最大電壓	3.6 VDC	3.6 VDC	5.5 VDC	5.5VDC	5.5 VDC	3.6 VDC
提供最小電壓	1.95 VDC	2.4 VDC	4.9 VDC	3.3VDC	4.9 VDC	1.8 VDC
最低適應溫度	-40度	-40度	-40度	-40度	-40度	-40度
最高適應溫度	85度	85度	85度	125度	85度	85度

資料來源：Mouser Eletronics Company, from:

<https://www.mouser.tw/search/refine.aspx?N=18357019>

一、MMA8451Q加速度計

MMA8451Q三軸加速度計的詳細內容可參照NXP Semiconductor公司產品介紹的data sheet (2014)，是一款具有14-bit分辨率的智能低功耗、三軸、電容式微機械加速度傳感器，如圖2-4可以了解到當X、Y、Z三軸感測到振動之後，先經過C-to-V converter(Capacitance to Voltage Converter)，該轉換器可以減少電力消耗並且提供伏特數(Alam, Arfah, Khan and Islam, 2010)，藉由電位差給予X、Y、Z三軸感知到的振動頻率傳送下去。經過ADC將類比訊號轉成數位訊號，之後將訊號傳到具有嵌入式數位訊號處理的DSP (Digital Signal Processor)，最後將資料傳到I²C(Inter IC)輸出口得到加速度值。嵌入式中斷功能可以節省整體功耗，解除主處理器不斷輪詢數據的負擔。同時具備訪問低通濾波數據和高通濾波數據的功能，可以將所需的數據分析降至最低程度，並且以I²C把數位訊號傳出，從而實現晃動檢測和快速的響應。該器件可被配置成利用任意組合可配置嵌入式的功能生程慣性喚醒中斷信號，能同時在靜止狀態保持低功耗模式。

I²C 是 NXP 公司開發的通訊協定，主要用來作為 IC 之間的通訊。他的傳輸通訊速度一般標準模式是 100 Kbit/s 或是 400 Kbit/s，而新一代的 I²C 傳輸通訊速率能到高速模式的 1 Mbit/s 與 3.4 Mbit/s，但此協定不適合傳送大量資料。I²C 又比 RS232 快速，所以他非常適合拿來設定 IC 初始值，或是控制訊號的傳輸。由於 I²C 只使用兩條線雙向開漏線通訊，SDA (串列資料線) /SCL (串列時鐘線)，上拉了電阻，因此空間非常節省。常使用的經典工作電壓是+3.3V 或+5V。圖 2-5 為 I²C 的時序圖，依照 Start-Data-Stop 傳送資料，I²C 的規範是以 SCL(Serial Clock) 為 low 時 SDA (Serial Data) 可以改變，SCL 為 high 時 SDA 必須保持不變，但大部份電路設計上會改成利用圖中兩側的 SCL edge 來觸發資料的傳送與接收，圖中最左端是 Start condition：SCL=high + SDA falling；最右端是 Stop condition：SCL=high + SDA raising。至於中間的資料，它是以 8+1 bit 為一組來傳送資料，即 8 bit data 外加 1 bit 的 Acknowledge，ACK 是 slave 用來回應 master 用的，表示已經接收到資料。

MMA8451Q規格如下：

(一)供電電壓：1.95 V至3.6 V

(二)接口電壓：1.6 V至3.6 V

(三) $\pm 2g/\pm 4g/\pm 8g$ 動態量程可選

(四)輸出數據速率(ODR)範圍：1.56 Hz至800 Hz

(五)噪聲： $99 \mu g/\sqrt{Hz}$

(六)14位和8位數字輸出

(七)I²C數字輸出接口（在上拉電阻為4.7 k Ω 時，最高頻率可達2.25 MHz）

(八)適用於7個中斷來源的2個可編程中斷引腳

(九)3個運動檢測嵌入式通道

(十)自由落體或運動檢測：1通道

(十一)脈衝檢測：1通道

(十二)晃動檢測：1通道

(十三)帶有可編程滯後補償的方向（橫向/縱向）檢測

(十四)自動喚醒和自動休眠的ODR(output data rate)可自動更改

(十五)32段採樣FIFO

(十六)高通濾波器數據可單獨輸出和通過FIFO輸出

(十七)符合RoHS標準，RoHS是由歐盟立法制定的一項強制性標準，要求電子或電器產品的材料不可使用鉛、鎘、汞、鉻、多溴二苯醚、多溴聯苯等六項有毒物質(盛日，2017)

(十八)功耗： $6 \mu A - 165 \mu A$

(十九)供電電源：3-5V（通信完全兼容5V工作系統）

(二十)模塊尺寸長×寬 = 14.5 mm × 20.5 mm

(二十一)安裝孔直徑3 mm，孔距15 mm

MMA8451Q三軸加速度計可針對低功耗模式或輸出數據的更高分辨率進行優化，當通過暫存器0x2A的第2位Low-Noise時，動態範圍將被限制成為4 g，將影響加速度計內部功能並降低雜訊。

從表2-2所述，靈敏度保持在規格範圍內，但將採樣模式更改為低功率會導致3 %的靈敏度偏移，這種現象同樣會出現於Low-Noise+低功耗、高分辨率模式、data rate從800Hz改變成其他任何速率時等。

解析度（Resolution），NXP Semiconductor (2014)的data sheet提及，是加速度中最小的可檢測增量，檢測出的最小變化量是有必要知道的。加速度計頻寬（Bandwidth）將決定測量分辨率，但可以使用濾波來降低測量噪聲並進一步提高解析度。透過降低輸出低通的頻寬同樣也可以提高分辨率之過濾。而建議計算公式如下：

$$R = N \times \sqrt{BW \times 1.6} \quad (1)$$

其中：

R：Resolution

N：雜訊強度(Noise Density)，MMA8451為 $99 \mu\text{g}/\sqrt{\text{Hz}}$

BW：Bandwidth

圖2-6顯示六種不同方向情況下的設備配置。這些方向定義為 LR = 右側，PU = 平面向上，PD = 平面向下，LL = 左側，以及前後俯視圖Back與Front。這些方向定義為右側是X軸正方向，左側為X軸負方向；平面向上為Y軸負方向，平面向下為Y軸正方向；往正前方的方向為Z軸正方向，往後方的方向為Z軸負方向。

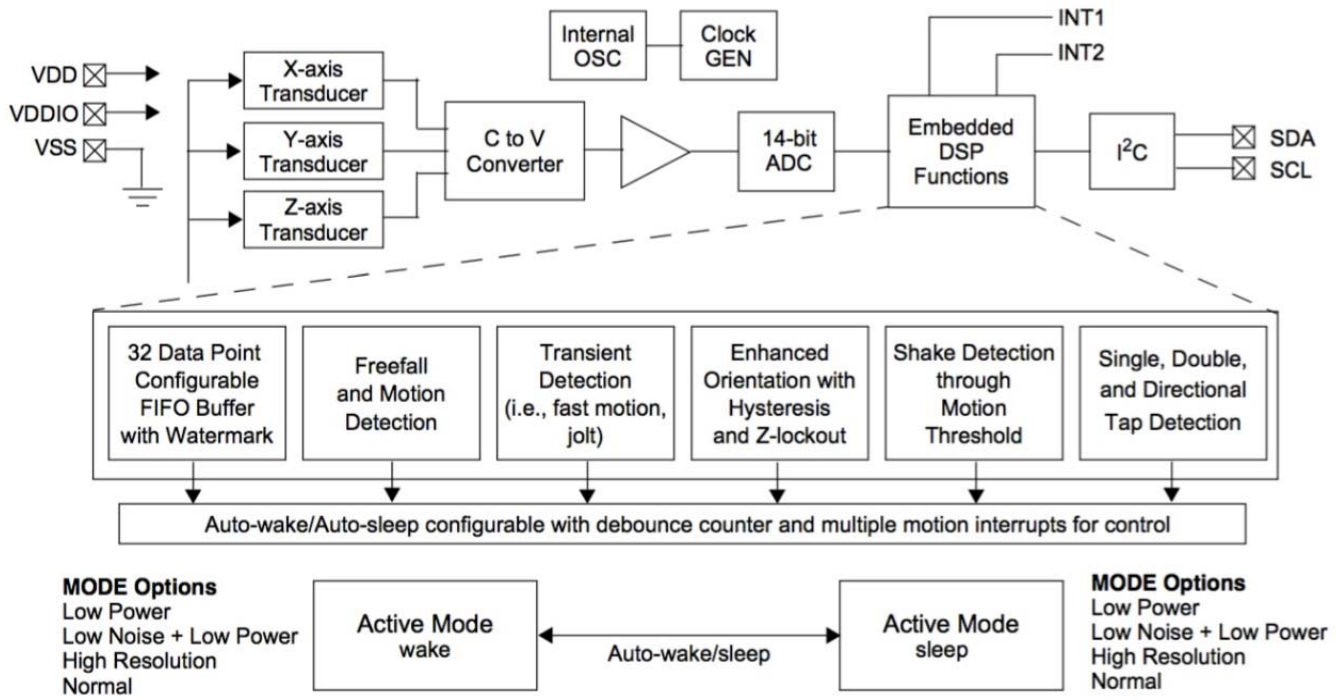


圖2-4 MMA8451Q Block diagram

資料來源：(NXP Semiconductor, 2014)

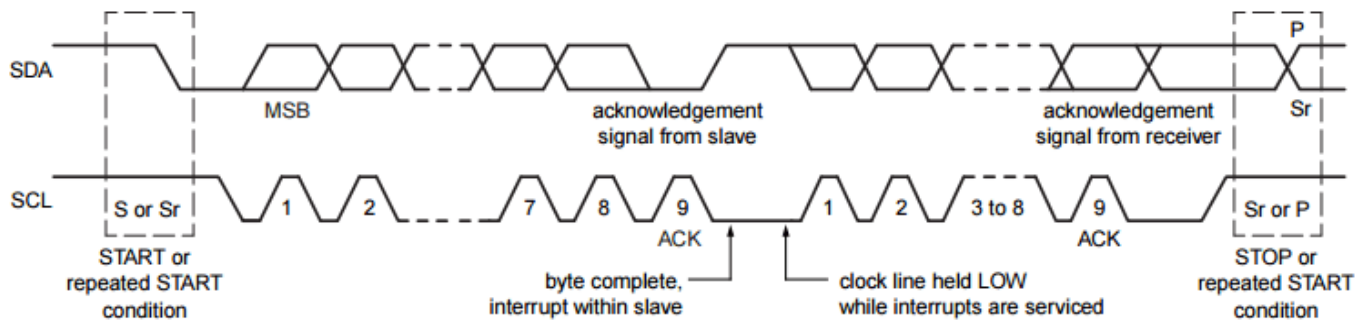


圖2-5 I²C時序圖

資料來源：(NXP Semiconductor, 2014)

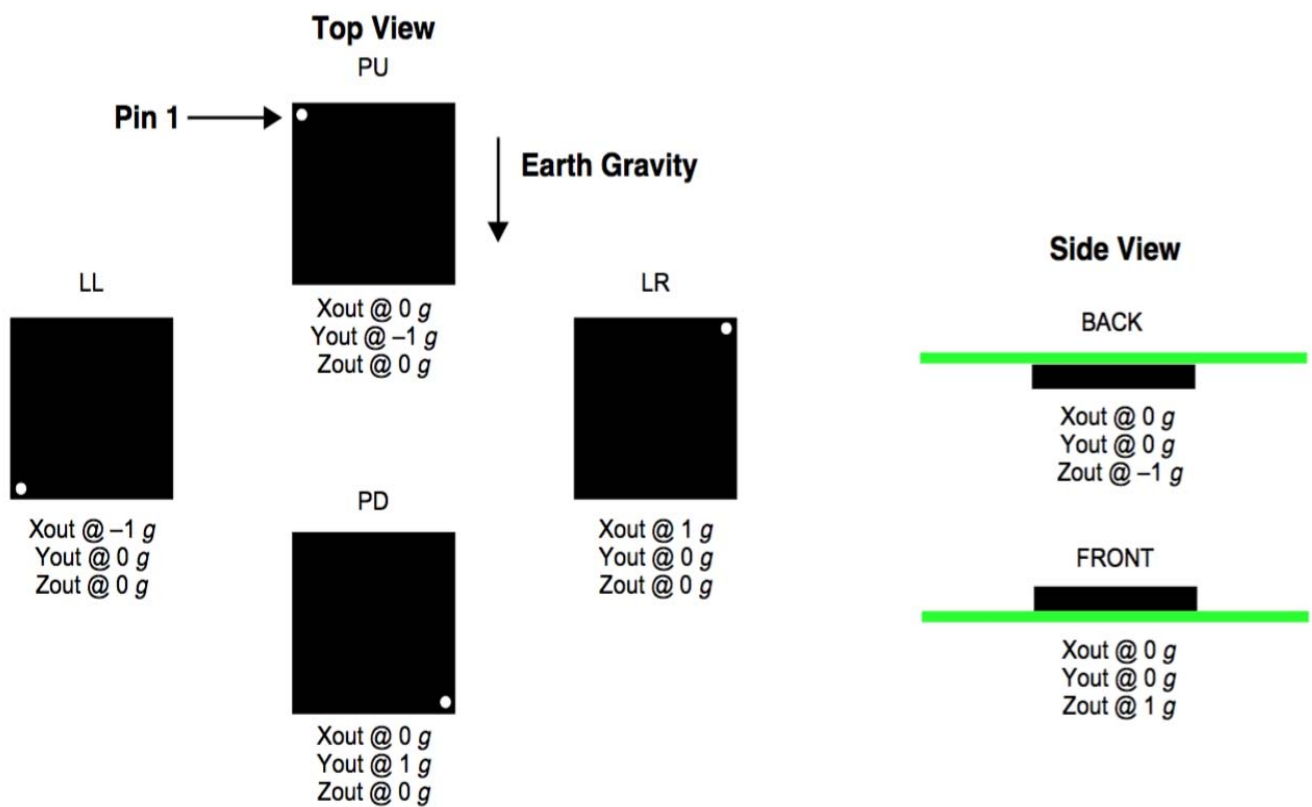


圖2-6 MMA8451Q orientation on perpendicular axis

資料來源：(NXP Semiconductor, 2014)

表2-2 MMA8451Q的機械特性

Parameter	Test conditions	Symbol	Min	Typ	Max	Unit
Measurement range ⁽¹⁾	FS[1:0] set to 00 2 g mode	FS	—	±2	—	g
	FS[1:0] set to 01 4 g mode		—	±4	—	
	FS[1:0] set to 10 8 g mode		—	±8	—	
Sensitivity	FS[1:0] set to 00 2 g mode	So	—	4096	—	counts/g
	FS[1:0] set to 01 4 g mode		—	2048	—	
	FS[1:0] set to 10 8 g mode		—	1024	—	
Sensitivity accuracy ⁽²⁾	—	Soa	—	±2.64	—	%
Sensitivity change vs. temperature	FS[1:0] set to 00 2 g mode	TCSO	—	±0.008	—	% / °C
	FS[1:0] set to 01 4 g mode		—		—	
	FS[1:0] set to 10 8 g mode		—		—	
Zero-g level offset accuracy ⁽³⁾	FS[1:0] 2 g, 4 g, 8 g	TyOff	—	±17	—	mg
Zero-g level offset accuracy post-board mount ⁽⁴⁾	FS[1:0] 2 g, 4 g, 8 g	TyOffPBM	—	±20	—	mg
Zero-g level change vs. temperature	−40 °C to 85 °C	TCOff	—	±0.15	—	mg / °C
Self-test output change ⁽⁵⁾ X Y Z	FS[1:0] set to 0 4 g mode	Vst	— — —	+181 +255 +1680	— — —	LSB
ODR accuracy 2-MHz clock	—	—	—	±2	—	%
Output data bandwidth	—	BW	ODR/3	—	ODR/2	Hz
Output noise	Normal mode ODR = 400 Hz	Noise	—	126	—	µg/√Hz
Output noise low-noise mode ⁽¹⁾	Normal mode ODR = 400 Hz	Noise	—	99	—	µg/√Hz
Operating temperature range	—	Top	−40	—	+85	°C

資料來源：(NXP Semiconductor, 2014)

二、MMA7660FC加速度計

MMA7660FC三軸加速度計的詳細內容參照NXP Semiconductor公司產品介紹的data sheet (2012)所述，是超低功率，緊湊型電容式微電機的加速度計，非常低功耗，小型容性MEMS的傳感器，具有低通濾波器，用於0 g偏移和增益誤差補償，以及用戶可配置的轉換成6位分辨率，用戶可配置輸出速率等功能。可通過中斷引腳（INT）可以提供傳感器數據變化、產品方向和姿態識別等中斷。圖2-7可以了

解到當X、Y、Z三軸感測到振動之後，會先通過數據多工器(MUX)，可以從多個類比輸入訊號或者多個數位輸入訊號中選擇一種訊號進行輸出的器件，兩者之中通常數位訊號是離散性的訊號，類比訊號是連續性的訊號，所以一開始的三軸感應到振動情況，並傳至數據多工器(MUX)，擋掉數位訊號，讓類比訊號輸出並跑後續的電路流程。輸出的類比訊號值到C-to-V converter(Capacitance to Voltage Converter)，該轉換器可以減少電力消耗並且以及提供伏特數(Ala et al., 2010)，藉由電位差給予X、Y、Z三軸感知到的振動頻率傳送下去。到達AMP，AMP具有放大電路效果，能增加訊號的輸出功率，透過電源取得能量來源，以控制輸出訊號的波形與輸入訊號一致，AMP亦可視為可調節的輸出電源，用來獲得比輸入訊號更強的輸出訊號。訊號輸出後經過ADC將類比訊號轉成為數位訊號，最後再以I²C把數位訊號傳出，從而實現晃動檢測和快速的響應。

MMA7660FC規格如下：

(一)輸出介面：I²C

(二)尺寸大小：3.0 mm * 3.0 mm * 0.9 mm (長*寬*高)

(三)低功率消耗量：

1.關閉模式：0.4 μ A

2.待機模式： 2 μ A

3.行動模式： 47 μ A 在1 ODR(output data rate)

(四)可配置的樣本率：每秒1到120樣本數

(五)低電壓操作：

1.類比伏特：2.4 V - 3.6 V

2.數位伏特：1.71 V - 3.6 V

(六)低功耗情況下有自動激發或休眠特性

(七)檢測傾斜方向的縱向/橫向能力

(八)檢測包括振動檢測與拍擊檢測

(九)堅固的設計，可承受高衝擊性(最高可承受10000g的受力，其中 g 為9.8 m/s²)

(十)符合RoHS標準，RoHS是由歐盟立法制定的一項強制性標準，要求電子

或電器產品的材料不可使用鉛、鎘、汞、鉻、多溴二苯醚、多溴聯苯等六項有毒物質(盛日，2017)

(十一)產品無鹵素物質

(十二)對環境相當友善的產品

(十三)低成本

從圖2-8可以發現到MMA7660FC有三種模式變換，分別為關閉模式(OFF mode)、待機模式(Standby mode)、行動模式(Active mode)，不同的模式會有不同的功率消耗量與運作方式，後面將配合表2-3來介紹模式的啟動方式與運作方式：

(一)關閉模式：該模式的運作情況是在DVDD(工作電壓)供電但AVDD(模擬電壓)沒有供電給MMA7660FC時，會無法收到任何的類比或數位訊號。在此模式的這情況也是無法讀取以及紀錄I²C的資料。

(二)待機模式：DVDD、AVDD皆供電給MMA7660FC時，此時就能夠人為寫程式觸發它成為行動模式。

(三)行動模式：此時MMA7660FC就能夠使感測器運作得到監測資料，行動模式從圖2-8可以發現裡面包含運作模式與休眠模式，當觸發運作模式時，如果沒有持續供給電力給Arduino板或者感測器沒有反應，就會自動轉變為休眠模式。

而在行動模式的時候，完全可以在三軸連續測得資料，且可以偵測振動、拍動、方向，且本身也有自動激發與睡眠的特性，自動睡眠機制可以在每秒1、2、4、8、16、32、64、120樣本啟動，自動激發機制可以在每秒1、2、4、8、16、32樣本啟動。

圖2-9可以顯示MMA7660FC六種不同方向的設備配置。這些方向定義為右側是Y軸負方向，左側為Y軸正方向；平面向上為X軸負方向，平面向下為X軸正方向；往正前方的方向為Z軸正方向，往後方的方向為Z軸負方向。

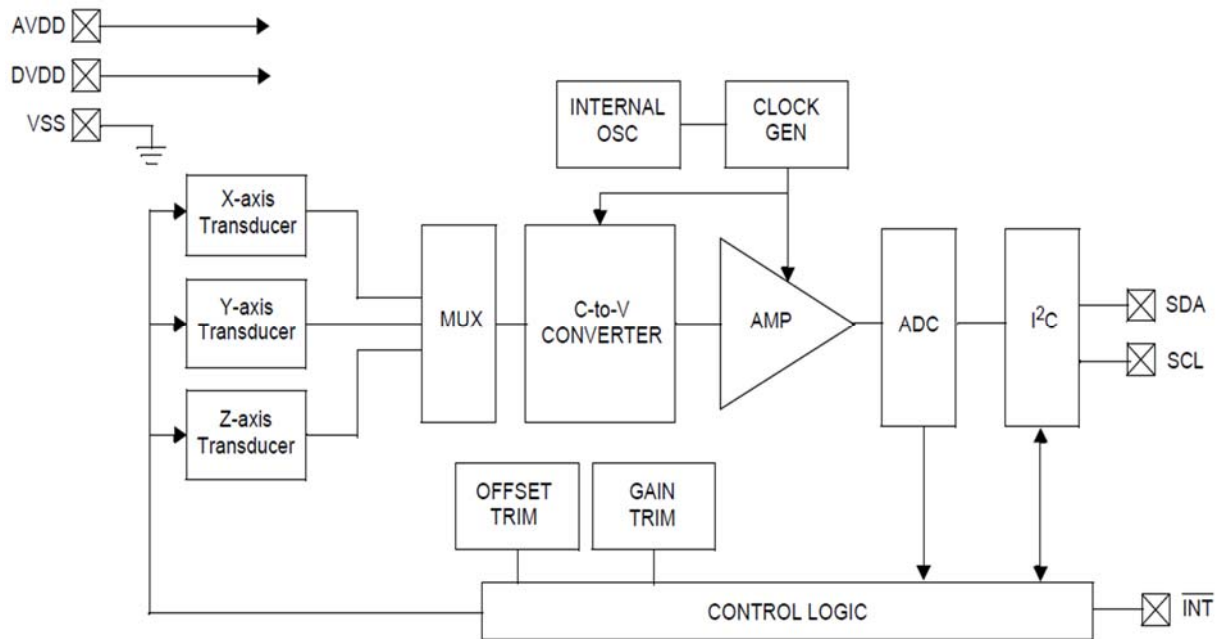


圖2-7 MMA8451 Block diagram

資料來源：(NXP Semiconductor, 2012)

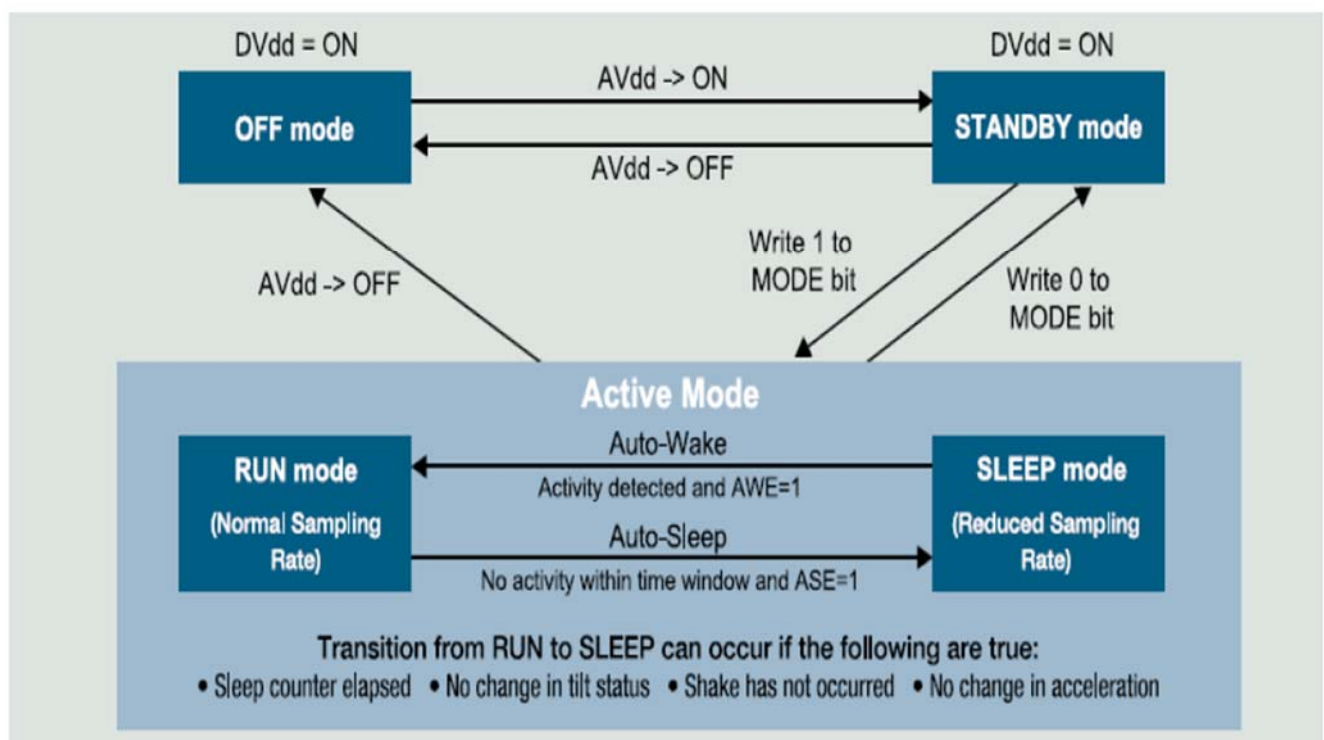


圖2-8 MMA7660FC的模式與各模式的關係圖

資料來源：(NXP Semiconductor, 2012)

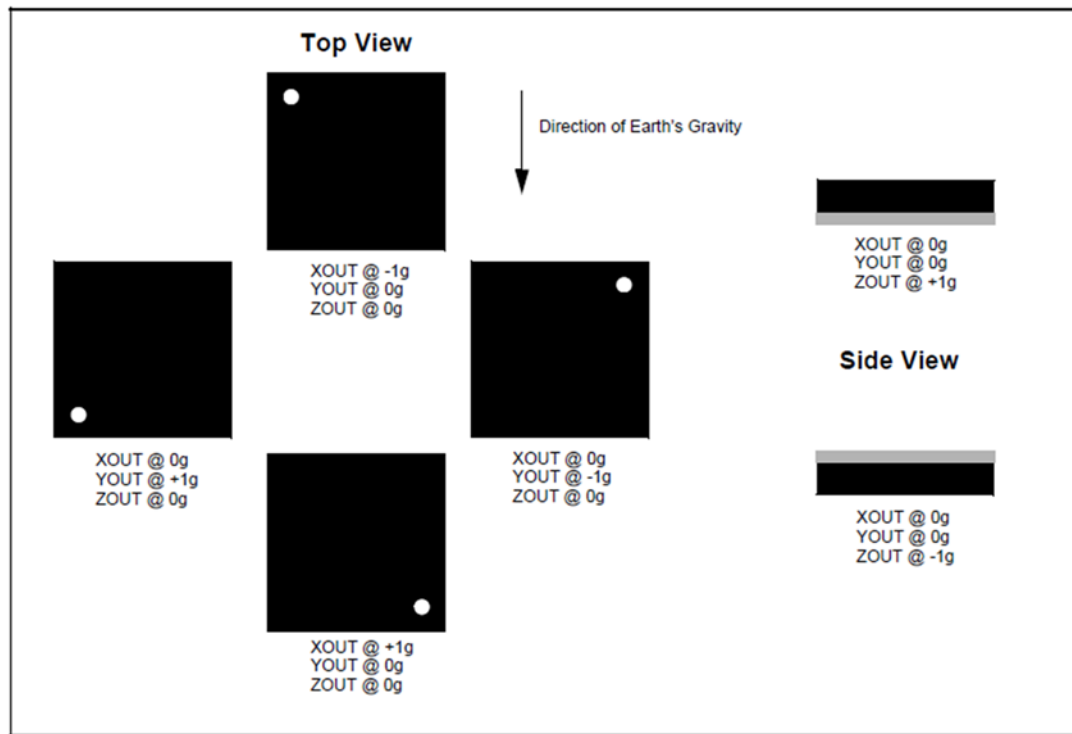


圖2-9 MMA7660FC orientation on perpendicular axis

資料來源：(NXP Semiconductor, 2012)

表2-3 MMA7660FC在不同模式下的運作

Measurement and Power Mode	I ² C Bus	DVDD	AVDD	Function
Power Down Mode	DVDD is down, so I ² C bus cannot be used for other devices (MMA7660FC clamps I ² C bus to DVDD pin)	Off	Off	MMA7660FC is powered down in both supplies. I ² C activity is unavailable on bus.
		Off	On	MMA7660FC is powered down in digital supply but not analog supply. I ² C activity is unavailable on bus. AVDD power cycling requires 50 msec.
Off Mode	MMA7660FC will not respond, but I ² C bus can be used for other devices (MMA7660FC does not load I ² C bus)	On	Off	MMA7660FC is powered down in analog supply but not digital supply. I ² C activity is ignored.
Standby Mode	MMA7660FC will respond to I ² C bus	On	On	MMA7660FC is powered up in both supplies, so registers can be accessed normally to set MMA7660FC to Active Mode when desired. MMA7660FC's sensor measurement system is idle.
Active Mode Auto-Sleep Auto-Wake	MMA7660FC will respond to I ² C bus	On	On	MMA7660FC is able to operate sensor measurement system at user programmable samples per second and run all of the digital analysis functions. Tap detection operates in Active Mode and Auto-Sleep, but not Auto-Wake.

資料來源：(NXP Semiconductor, 2012)

三、ADXL335加速度計

本部分將參考Analog device公司針對ADXL335的資料表 (2009)：

ADXL335是一款小尺寸、薄型、低功耗、完整的三軸加速度計，提供經過信號調理的電壓輸出，可以測量靜態加速度與動態加速度。用戶使用XOUT、YOUT和ZOUT引腳上的電容 C_X 、 C_Y 和 C_Z 選擇該加速計的帶寬。可以根據應用選擇合適的帶寬，X軸和Y軸的帶寬範圍為0.5 Hz至1600 Hz，Z軸的帶寬範圍為0.5 Hz至550 Hz。

ADXL335規格如下：

(一)輸出介面：類比

(二)尺寸大小：4.0 mm * 4.0 mm * 1.45 mm (長*寬*高)

(三)低功率消耗量：350 μ A

(四)低電壓操作：1.8 V - 3.6 V

(五)檢測傾斜方向的縱向/橫向能力

(六)檢測包括振動檢測與拍擊檢測

(七)符合RoHS標準，RoHS是由歐盟立法制定的一項強制性標準，要求電子或電器產品的材料不可使用鉛、鎘、汞、鉻、多溴二苯醚、多溴聯苯等六項有毒物質(盛日，2017)

(八)產品無鹵素物質

(九)對環境相當友善的產品

(十)低成本

ADXL335在 $V_S = 3$ V下進行測試且以其為額定電源電壓；然而， V_S 可以低至1.8 V或高達3.6 V。而某些性能參數隨著電源電壓變化而變化，像是輸出靈敏度(或比例因子)與電源電壓成比例變化。 $V_S = 3.6$ V時，輸出靈敏度典型值為360 mV/g。 $V_S = 2$ V時，輸出靈敏度典型值為195 mV/g。而輸出噪聲的密度隨電源電壓升高而降低。這是因為比例因子(mV/g)增加而噪聲電壓卻保持不變。 $V_S = 3.6$ V時，X軸和Y軸的噪聲密度典型值為120 μ g/ $\sqrt{\text{Hz}}$ ，而 $V_S = 2$ V時，X軸和Y軸的噪聲密度典型值為270 μ g/ $\sqrt{\text{Hz}}$ 。

圖2-10可以顯示ADXL335六種不同方向的設備配置。這些方向定義為右側是Y軸負方向，左側為Y軸正方向；平面向上為X軸負方向，平面向下為X軸正方向；往正前方的方向為Z軸正方向，往後方的方向為Z軸負方向。

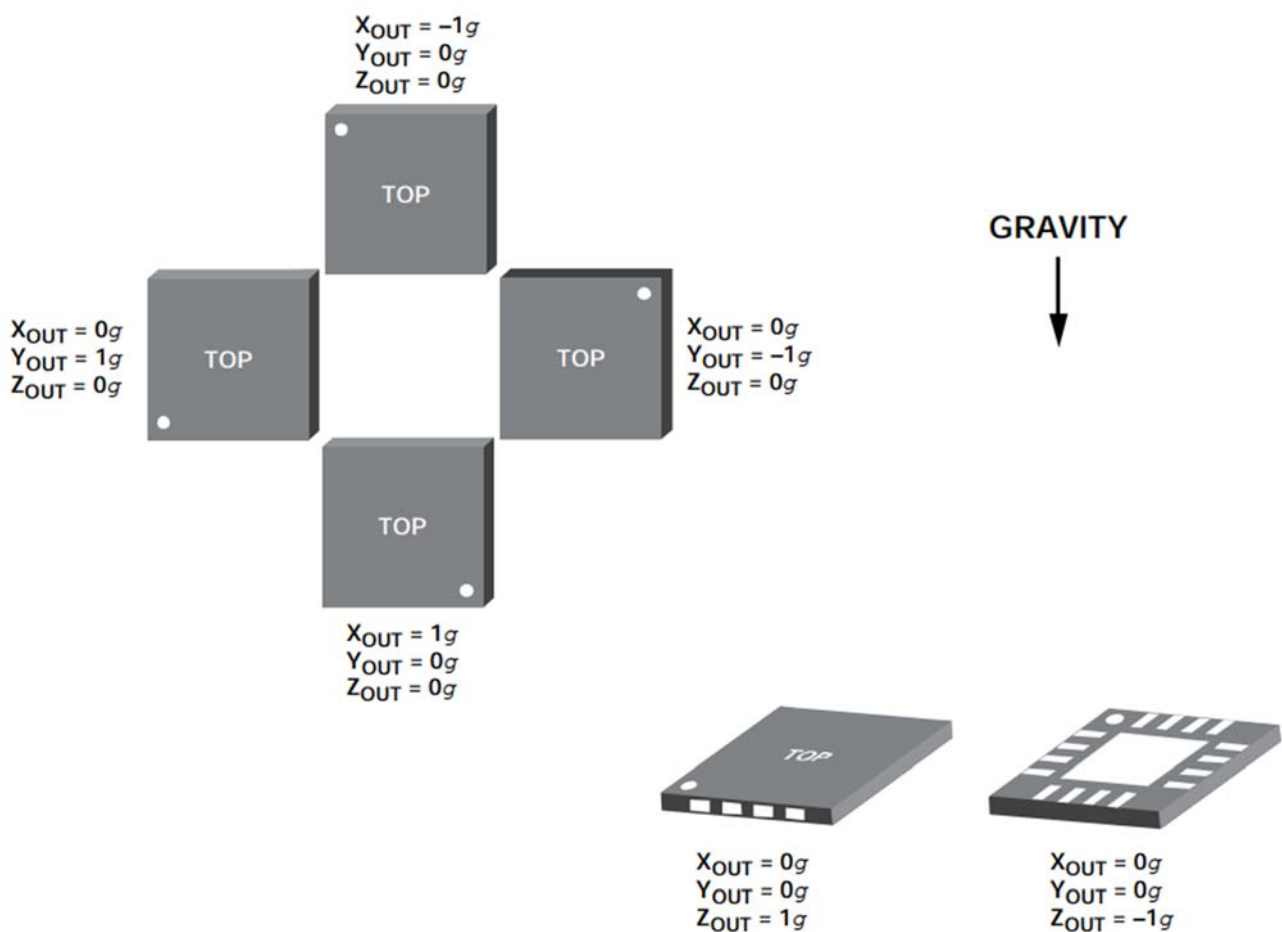


圖2-10 ADXL335 orientation on perpendicular axis

資料來源：(Analog device, 2009)

第二節 微型處理器開發與選定

微控制器基本上係將中央處理器(CPU)、記憶體(RAM/ROM)、輸入輸出(I/O)介面都整合在一塊晶片上的微型電腦。體積小、記憶體容量小、輸入輸出介面簡單且功能較低，但其發展非常迅速，可應用於多種領域。目前市面上有多種微處理器，一般較為熟知的有Arduino、Raspberry Pi與ARM。Arduino為一個開放原始碼的單晶片微控制器，採用開放原始碼的軟硬體平臺，可以利用Arduino語言結合加速度感應器作出互動產品。樹莓派(Raspberry Pi)，為Linux系統的單板機微控制器，係由英國的樹莓派基金會所開發，支援BBC、BASIC、C語言和Perl等程式語言。ARM處理器是Acorn計算機有限公司面向低預算市場設計的第一款RISC微處理器，是功能比較強大的單片機，其廣泛地使用在許多嵌入式系統設計，由於ARM處理器多嵌入其他設計故不考慮。表2-4是Arduino板、樹莓派的比較，可以發現到因為樹莓派是比較多功用的微型處理器，所以不管在Memory、Flash、Clock Speed都明顯大於Arduino許多，然而功能相當多，購買成本上也需要比較大的開銷，而我們發現Arduino的板子大概新臺幣2000元以內可以購得，而樹莓派的價格是Arduino的五倍左右，粗估新臺幣一萬元以內，且配合後續將會提到的NB-IoT的硬體SIM7000E板，大約新臺幣2500元，所以可以在低於新臺幣4500元的花費成本下得到我們所需的目標，雖然樹莓派功能多，但本計畫目的在微型處理器的部分只要偵測到加速度計並與NB-IoT模組溝通，功能較少，Arduino應可達成預定任務，且Arduino功能單一啟動快，程序執行效率會比較高。所以經過評估後，本研究團隊將在後續使用Arduino與NB-IoT的結合來達到目標。

目前本研究團隊正在使用Arduino UNO板(圖2-11)以及Arduino M0 Pro板(圖2-12)來做試驗。從表2-5可以發現各式各樣的Arduino板種類，了解到先前的Arduino UNO板雖然相當便宜，但儲存容量在快閃記憶體(Flash)、電子抹除式可複寫唯讀記憶體(EEPROM)與靜態記憶體(SRAM)各別只有32kB、1kB與2kB，可以讀取並儲存的容量是相當少的，所以後來使用Arduino M0 Pro板，是Arduino Zero板的改良版本，花費較高但儲存容量Flash、EEPROM與SRAM分別為256kB、16kB與32kB，資料讀取與儲存上比較多。試驗的兩塊微型處理器Arduino UNO板與Arduino M0

Pro板從表2-6在仔細比較，可以發現到除了儲存容量以外，Arduino M0 Pro板的腳位(Pins)整體來說是比Arduino UNO板還要多的，表示Arduino M0 Pro板功用上是比Arduino UNO板多樣的，兩板子消耗的安培數差不多，價錢雖然Arduino M0 Pro板比較貴，但在容量、功用、電能消耗等條件是優於Arduino UNO板的。



圖2-11 Arduino UNO板照片

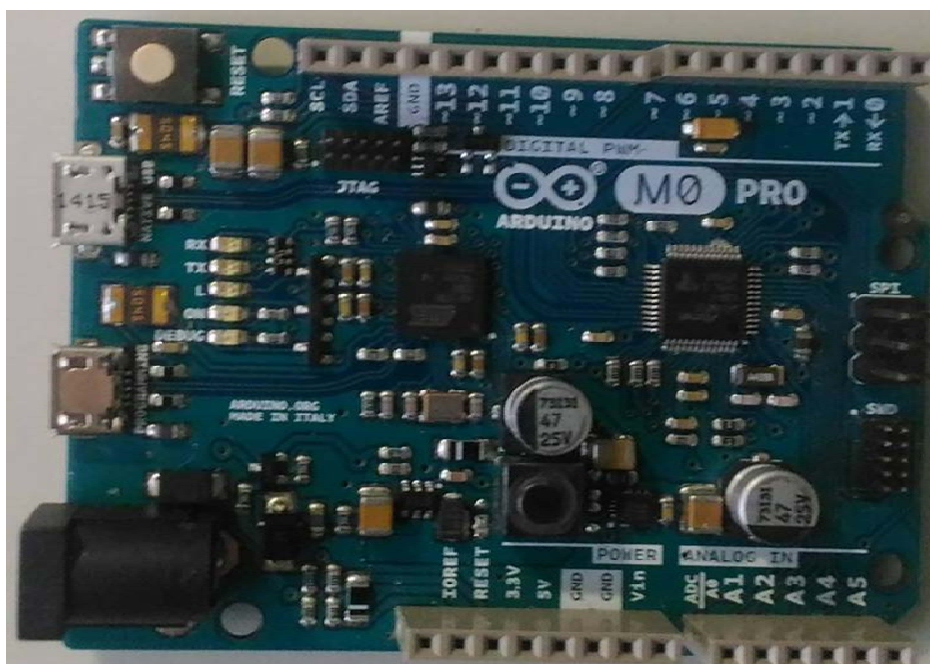


圖2-12 Arduino M0 Pro板照片

表2-4 Arduino與樹莓派之比較

	Arduino M0 Pro	Raspberry Pi Model B
Size	7.6cm×1.9cm×6.4cm	8.6cm×5.4cm×1.7cm
Memory	0.002MB	512MB
Clock Speed	16MHZ	700MHZ
On Board Network	None	10/100 wired Ethernet RJ45
Multitasking	No	Yes
Input voltage	7 to 12 V	5 V
Flash	32KB	SD card (2 to 16GB)
USB	One, input only	Two, peripherals OK
Operating System	None	Linux distributions
Integrated Development Environment	Arduino	Scratch, IDLE, anything with Linux support

資料來源：果壳实验室，2016，網址：

<https://read01.com/zh-tw/N7ajn.html#.W2LF0dUzbIV>

表2-5 Arduino的種類與介紹

名字	處理器頻率	主機接口	伏特	Flash (kB)	EEPROM (kB)	SRAM (kB)
Arduino	48 MHz	USB	3.3 V	256	No	32
Arduino 101	32 MHz	USB	3.3 V	196		24
Arduino Zero	48 MHz	USB	3.3 V	256	16	32
Arduino Due	84 MHz	USB	3.3 V	512	0	96
Arduino Yún	16 MHz, 400 MHz	USB	5 V	32	1	2.5
Arduino Leonardo	16 MHz	USB	5 V	32	1	2.5
Arduino Uno	16 MHz	USB	5 V	32	1	2
Arduino Mega2560	16 MHz	USB	5 V	256	4	8
Arduino Ethernet	16 MHz	Ethernet Serial interface	5 V	32	1	2
Arduino Nano	16 MHz	USB	5 V	32	1	2
Arduino Pro	16 MHz	UART Serial, I2C(TWI), SPI	5 V or 3.3 V	31	1	0.5
Arduino Pro Mini	8 (3.3 V)/ 16 (5 V) MHz	Six pin serial header	3.3 V / 5 V	32	1	2

表2-6 Arduino UNO板與Arduino M0 Pro板比較表

	Arduino UNO	Arduino M0 Pro
Microcontroller	ATmega328	ATSAMD21G18
Processor bits	8-bit	32-bit
Clock rate	16MHz	48MHz
Digital I/O Pins	14 (of which 6 provide PWM output)	20
Analog Input Pins	6	6
Analog Output Pins	1	0
Operating Voltage	3.3V	5V
Input Voltage	7-12V (recommended)	5-15V
PWM Output Pins	6	12
ADC Resolution	10-bit	12-bit
DC Current per I/O Pins	7 mA	40 mA
Flash	32 KB	256 KB
SRAM	2 KB	32 KB
EEPROM	1 KB	16 KB
Power Consumption	44 mA	50 mA
Price consumption	200~400 NTD	1200~1600 NTD

以下將會介紹本計畫會使用到的 Arduino 軟體與硬體的詳細內容：

一、Arduino 開發環境

Arduino 的商標與軟體圖示在圖 2-13。開發環境為 Arduino IDE (Integrated Development Environment) 的介紹將參考趙英傑 (2016)書籍做介紹，在 2003 年，義大利伊夫雷亞互動技術學院的研究生 Hernando Barragan 設計一套簡易且便宜的微電腦設備，讓對電子學沒基礎的學生也能上手，後來發展成的控制板就是 Arduino 板。

Arduino 板後來發展種類也越來越多，不過差異點也只有微處理器類型、外觀尺寸與電腦連線方式，程式寫法幾乎相同。

Arduino 開發環境可以安裝於 Windows、Mac OS X、Linux 這三種常見的作業系統中。使用的語法基本上與 C/C++幾乎一模一樣，實際上底層使用的是 avr-gcc 的編譯器，因此稍微具有 C 語言方面經驗的使用者，很容易就能上手編譯程式碼。在下載 Arduino 開發環境後，使用 Arduino 方式如圖 2-14 所示，在開發環境中編寫程式，並且驗證程式碼是否正確，正確通過後做編譯，將程式碼翻譯成為電腦能夠理解的方式，最後進行上傳的動作，程式碼將會藉由 USB 傳輸線將下指令上傳至 Arduino 板，而 Arduino 板的微控制器有開機啟動程式，可直接接收並執行編譯的程式碼。



圖2-13 Arduino (a) IDE圖樣 (b)商標



圖2-14 Arduino硬體與軟體使用圖示

資料來源：(趙英傑，2016)

二、Arduino UNO板

Arduino UNO板以下的內容將參照林信良 (2015)的書籍作介紹。Arduino UNO板的運作直流電壓為5V，可透過三個管道提供電源：USB、電源輸入插座、VIN腳位。使用USB連接埠提供給開發板的電源必須是5V的電壓，最基本的方式就是直接與電腦連接；如果要透過電源輸入插座供電，插頭直徑必須為2.1mm的DC (Direct current) 插頭，供應的電壓限制為6V至20V；VIN供電給開發板時，要求與透過電源輸入插座相同，VIN提供的電壓，也是經由直流電壓調節模組調整為5V與3.3V。由於直流電壓調節模組的輸入端連接了電源輸入插座供電與VIN，因此若經由電源輸入插座供電，VIN將會獲得與電源輸入插座相同的電壓。

「數位輸出、輸入腳位」，這些開發板上的腳位可輸出高電位 5V 與低電位 0V 的數位訊號，也可以接受數位訊號，腳位編號為 D0 到 D13。圖 2-15 中可以看到，D3、D5、D6、D9、D10、D11 的數字編號旁，還有個波浪符 (～)，表示這些腳位可以用數位訊號來模擬出類比訊號，使用的方法是 PWM (Pulse Width Modulation)。D0 與 D1 這兩個數位腳位分別有標示 TX(Receiver)RX(Transmitter)，此二腳位用於序列埠連接，且與 USB 序列埠連接。

「類比輸入腳位」A0 至 A5 六個腳位，可用來接受類比電壓輸入，但不能輸出類比電壓，類比電壓需要必須透過如前面所提之數位腳位 D3、D5、D6、D9、D10、D11 以 PWM 方式模擬。

圖 2-15 能明顯看見一片顯而易見的晶片，為 Atmel AVR 系列中的 ATmega328，採雙列直插封裝（Dual in-line package），簡稱為 DIP，是一種微控制器，可使用小的一字起子或尖嘴鉗將之撬起做更換。

控制板左上方有 SCL、SDA 兩個腳位，這是用來連接支援 I²C 協定的設備，此協定只採用兩條接線，可在感測器與開發板如 Arduino 之間傳遞資訊，SCL 與 SDA 也分別與類比腳位 A4、A5 連接，方便接線之用。D10、D11、D12、D13 是作為 SPI（Serial Peripheral Interface）協定的介面，SPI 使用四條線連接開發板與支援之設備。

最後在圖 2-16 的部分是 ATmega328 的右上方，有 3x2 線上燒錄用的 ICSP（In-Circuit Serial Programming）腳位，可連接 ISP（In-system programmer）線上燒錄器，對 ATmega328 進程式燒錄，其中幾個腳位與 D11、D12、D13 連接在一起。

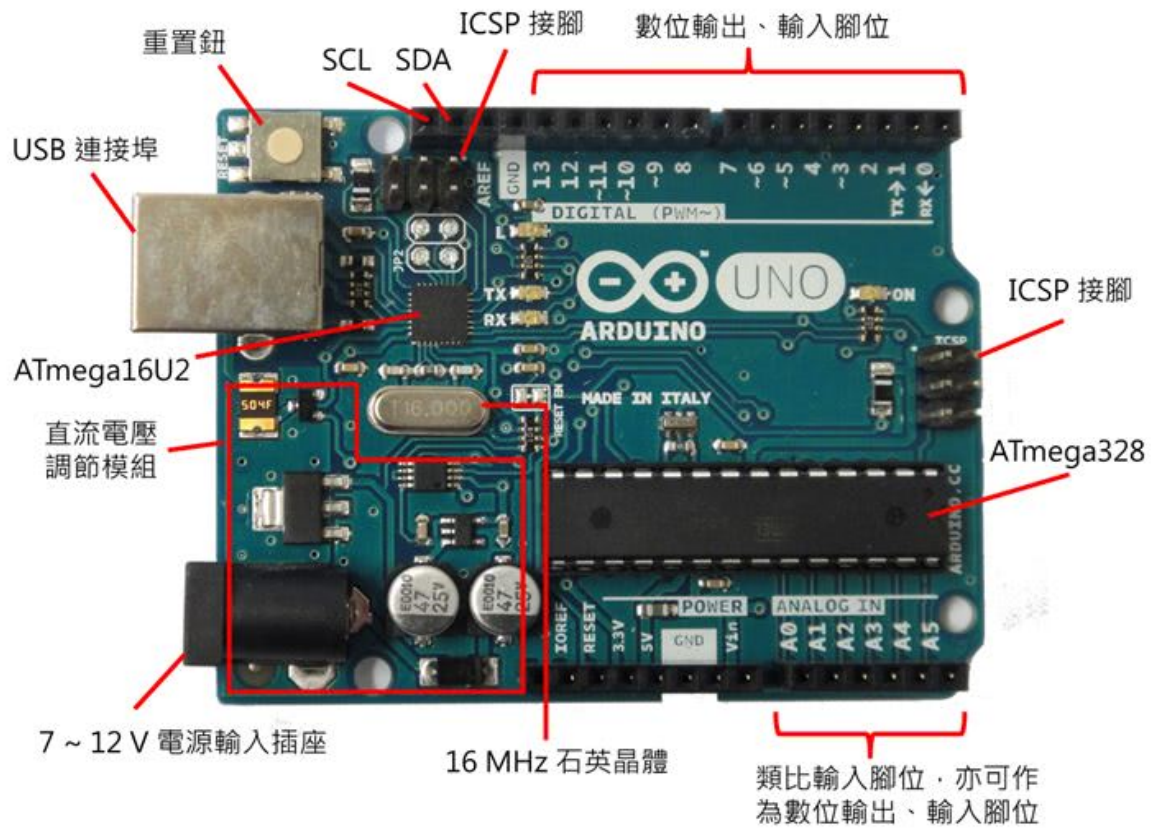


圖2-15 Arduino UNO板內容介紹

資料來源：(林信良，2015)



圖2-16 ICSP (In-Circuit Serial Programming) 腳位

資料來源：(林信良，2015)

三、Arduino M0 Pro板(需對照圖2-17)

Arduino M0 Pro可以視為是原先基本款式Arduino UNO的多方面升級版本，故構造的部分會有一些功能會與Arduino UNO介紹的相同。由於Arduino M0 Pro是Arduino UNO升級而成的，在Arduino官方網站英文頁面也有提到：“The Arduino M0 represents a simple, yet power, 32-bit extension of the Arduino UNO platform.” 這麼一段話。最引人注目的幾項增強功能，包含Arduino UNO沒有的DAC (Digital to Analog Converter)、ADC (Analog to Digital Converter) 解析度更為精細、SRAM從原先的2 KB大幅增加為32 KB，讓使用者們可以在使用上存得更多的資料。

Arduino M0 Pro可透過三個管道提供電源：USB、電源輸入插座、VIN腳位。最基本的方式就是直接與電腦連接；如果要透過電源輸入插座供電，插頭直徑必須為2.1mm的DC (Direct current) 插頭，供應的電壓限制為6V至20V，而Arduino M0 Pro有一項防護機制是USB連接線超過500mA時，擔心板子會有燒壞的問題所以會立刻停止運作，等到安培數降低至500mA再運作；VIN供電給開發板時，要求與透過電源輸入插座相同，VIN提供的電壓，也是經由直流電壓調節模組調整為5V與3.3V。由於直流電壓調節模組的輸入端連接了電源輸入插座供電與VIN，因此若經由電源輸入插座供電，VIN將會獲得與電源輸入插座相同的電壓。雖然Arduino M0 Pro板有5V腳位，但如果要與其他板子溝通的話，板子的互相溝通不要連接5V腳位因為擔心會破壞板子，可選擇3.3V。

「數位輸出、輸入腳位」，這些開發板上的腳位可輸出高電位 5V 與低電位 0V 的數位訊號，也可以接受數位訊號，腳位編號為 D0 到 D13。還可以看到 D0 至 D13 的數字編號旁，還有個波浪符 (～)，表示這些腳位可以用數位訊號來模擬出類比訊號，使用的方法是 PWM (Pulse Width Modulation)。D0 與 D1 這兩個數位腳位分別有標示 Rx (Receiver) 與 Tx (Transmitter)，此二腳位用於序列埠連接，且與 USB 序列埠連接。

「類比輸入腳位」A0 至 A5 六個腳位，可用來接受類比電壓輸入，但不能輸出類比電壓，類比電壓需要必須透過如前面所提之數位腳位需要有波浪符號才可使用，以 Arduino M0 Pro 板來說任何一腳位都可接受類比電壓，並以 PWM 方式

模擬。

控制板上有 SCL、SDA 兩個腳位，這是用來連接支援 I²C 協定的設備，此協定只採用兩條接線，可在感測器與開發板如 Arduino 之間傳遞資訊，SCL 與 SDA 也分別與類比腳位 A4、A5 連接，方便接線之用。D10、D11、D12、D13 是作為 SPI 協定的介面，SPI 使用四條線連接開發板與支援之設備。

Arduino M0 Pro 板的介紹在 Arduino 官方網站也有詳細介紹(網址：<https://www.arduino.cc/en/Guide/ArduinoM0Pro>)。特別提到 Arduino M0 Pro 有兩個 USB 連接孔，分別是 Native Port 與 Programming Port。

一、Native Port：Native Port 的 USB 接孔通常用來提供電力給其他儀器並給予操縱，像是滑鼠或鍵盤的控制，Native Port 是直接連接微控制器 SAMD21 MCU。Native Port 連接成功後，會有一項快閃記憶體的软件重新將過去資料剔除，以便連接新資訊，控制就是 SAMD21 MCU 所控制。

二、Programming Port：Programming port 上有一個處理器是 ATMELEDBG(embedded debugger)，這處理器可以直接協助你直接連接 UART，之後連到 SAMD21 這兩項微控制器上，本計畫將使用的加速度計是需要讓 Arduino 讀取資料，所以本計畫將會將 USB 連接線安裝在 Programming Port 的 USB 接孔上，而不會連接到 Native Port。所以總的來說，Arduino M0 Pro 板上的微控制器是 UART 與 SAMD21 MCU。

板子上的 LED 燈在傳資料、充電、連接 USB 至電腦時，如果你成功傳遞資料給其他板子的話，Rx、Tx 的燈會閃爍，當通電(充電或連通電腦)，ON 燈就會亮，傳遞資料 DEBUG 燈會閃爍，如果上傳結束，L 會亮橘燈，且重新啟動鍵按下，L 的橘燈會閃爍。

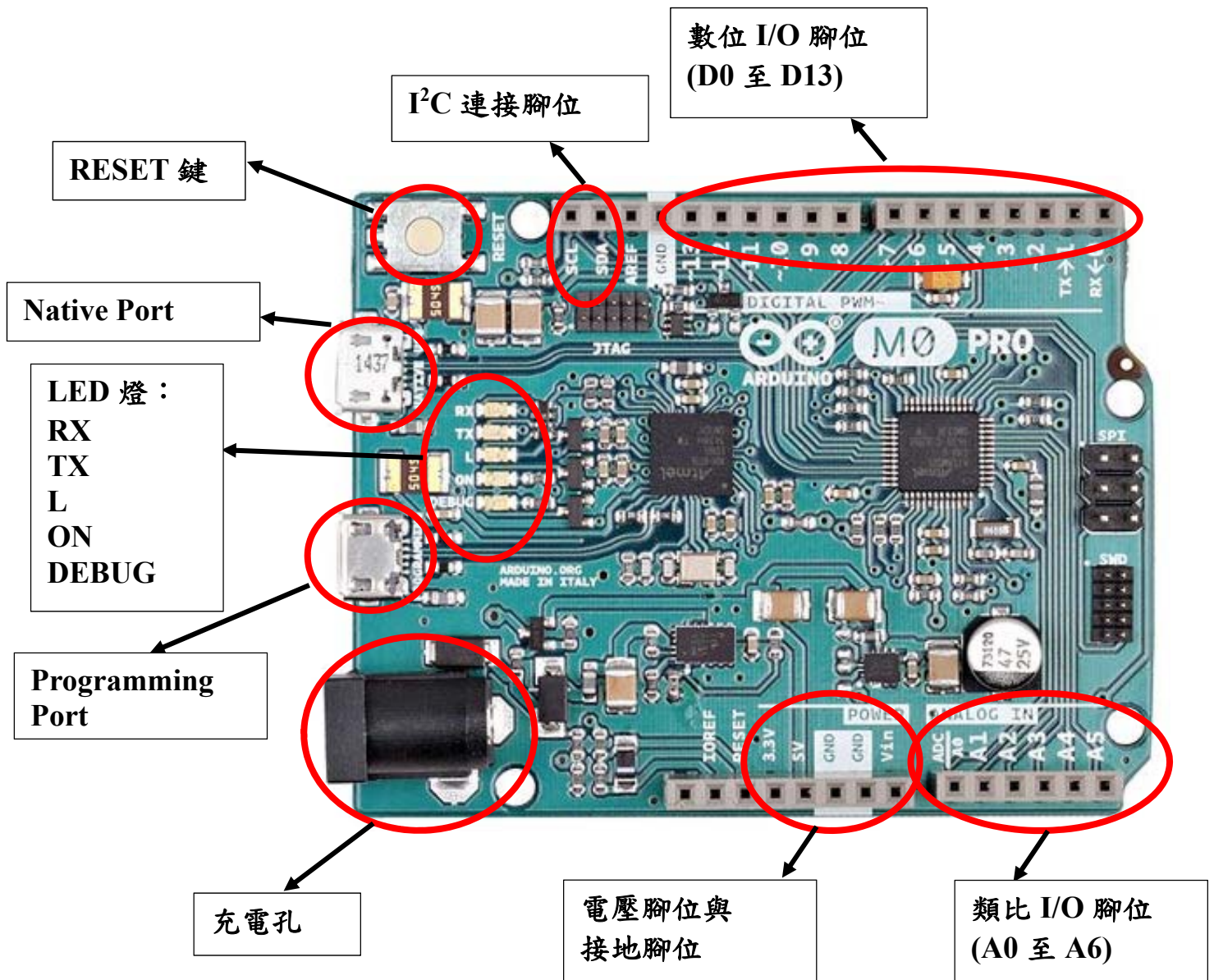


圖2-17 Arduino M0 Pro版內容介紹

第三節 加速度計與微型處理器Arduino板結合

完成加速度計與Arduino板之評比後，選定適當的加速度計與Arduino板進行連接以讀取加速度計產生的加速度值。而本計畫各有兩種加速度計與微型處理器，所以也將試驗分成兩組，第一組為Arduino M0 Pro板與MMA7660FC加速度計的結合，第二組為Arduino UNO板與MMA8451Q加速度計的結合。

一、Arduino M0 Pro板與MMA7660FC加速度計的結合

圖2-18所示，因為MMA7660FC加速度計是屬於I²C的連接方式，加速度計所產生的四條線分別是SDA、SCL、VCC、GND。加速度計的SDA、SCL為連Arduino M0 Pro板的SDA與SCL腳位，VCC因為Arduino M0 Pro板上只有5V與3.3V兩腳位，Arduino M0 Pro板連加速度計並不會有任何破壞或燒壞，而3.3V須留著留給未來與其他板連結與溝通會用到，所以本研究就將他的VCC插在Arduino M0 Pro板上的5V腳位，最後兩邊的GND互相連接就完成感測器與微型處理器的結合了，如圖2-19示意圖，以及圖2-20組合完成的模樣。

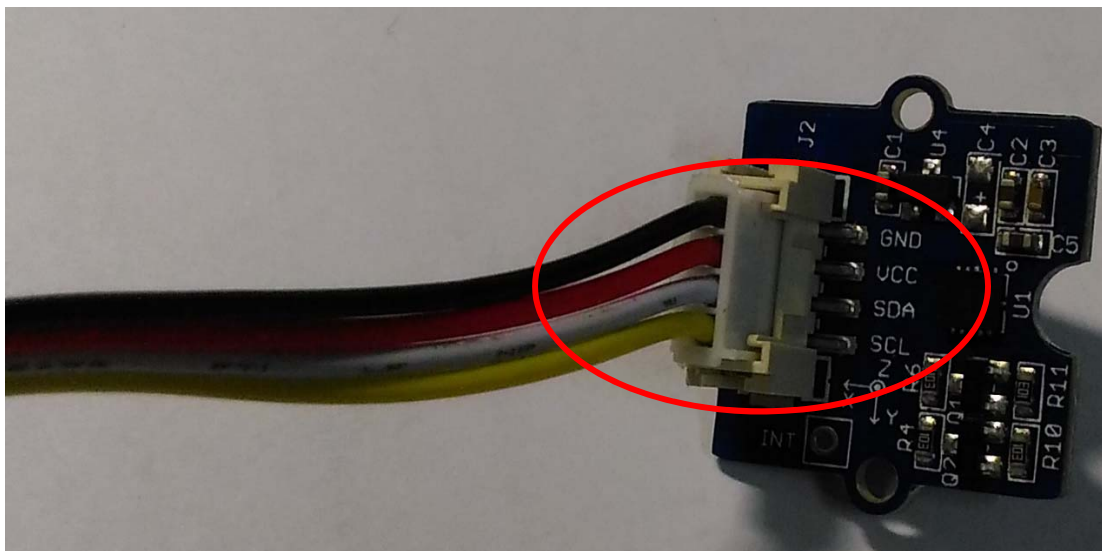


圖2-18 MMA7660FC的I²C連接要位

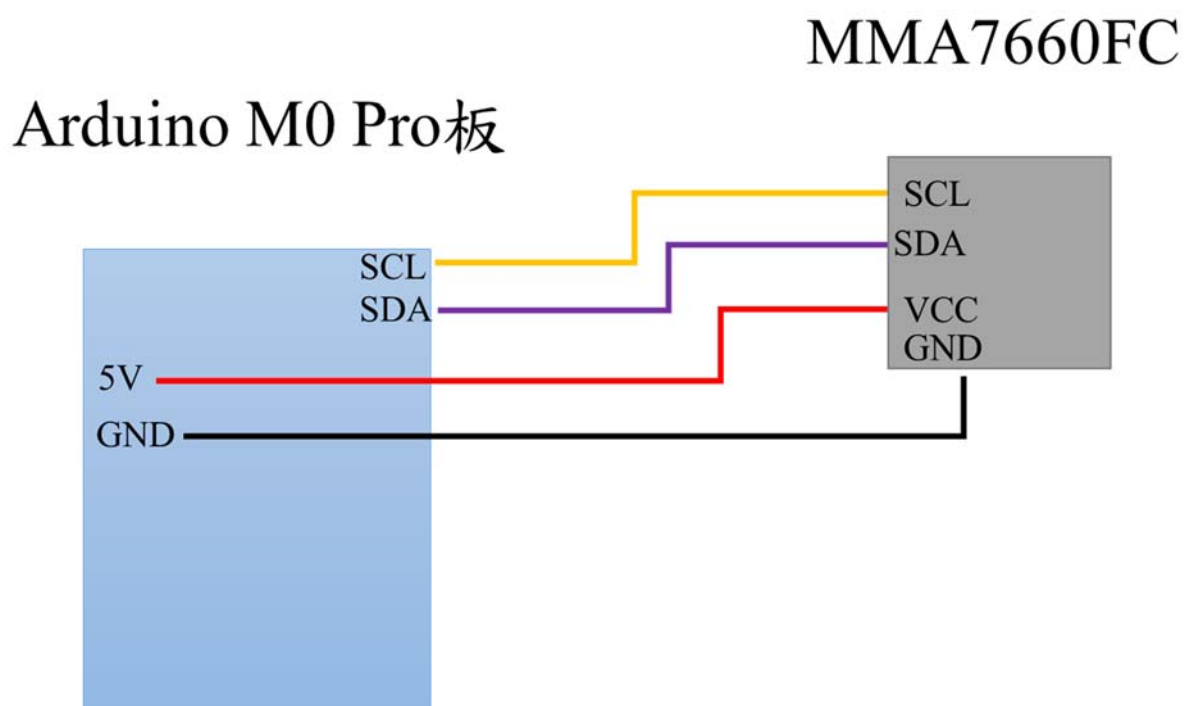


圖2-19 MMA7660FC與Arduino M0 Pro連結示意圖

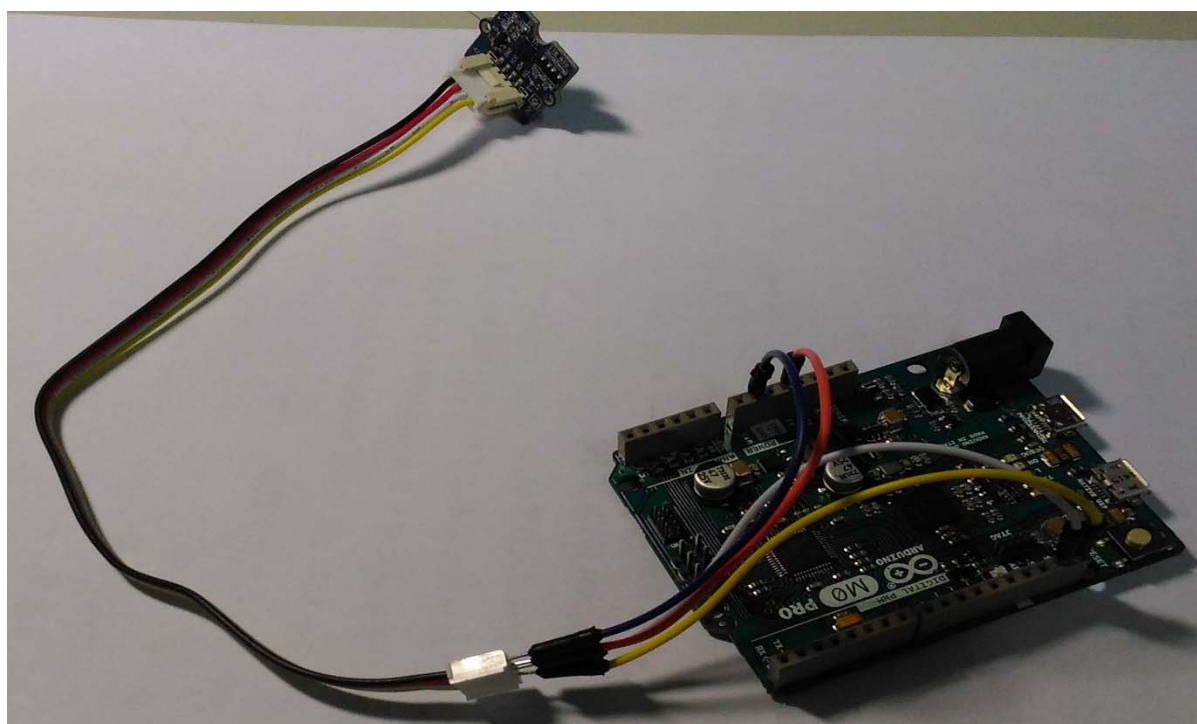


圖2-20 MMA7660FC與Arduino M0 Pro連接圖

Arduino M0 Pro板與MMA7660FC加速度計組合後，可利用USB連接線將電腦與Arduino M0 Pro的Programming Port進行連接，然後打開Arduino軟體介面並可以開始使用。

使用Arduino IDE的方法如下所述(已經利用USB連接線將電腦與Arduino板上USB連接孔做連接)：

(一)從電腦上電腦管理找尋裝置管理員，並從裡面的連接埠去找是否Arduino板跟電腦連接後有出現，並且也出現了序列埠。以Arduino M0 Pro板連接電腦出現的序列埠來說，序列埠為COM3(如圖2-21)。

(二)打開Arduino軟體，如圖2-22所示，第一個步驟左上角的工具欄確定是連接COM3，並且上面一格開發板也必須選目前試驗用的Arduino板，以Arduino M0 Pro板為例，開發板應選Arduino M0 Pro，如果開發板的欄位找不到所使用的種類，可於開發板欄位中的“開發板管理員”搜尋並下載所使用的開發板。

(三)事前工作準備就序後，就是可以開始寫程式碼並上傳給Arduino運作的時候。針對感測器需要找該感測器的cpp*檔與h*檔(上網找通常會有)。並將其放在C槽裡的Arduino檔裡的library資料夾裡(注意library中別放兩個一樣檔案，不然上傳程式碼的時候讀library會讀到多個檔案，會混淆讀檔軟體)。以本計畫來說，針對MMA7660FC感測器的程式碼，要找尋MMA7660.cpp與MMA7660.h兩項檔案放在C槽Arduino資料夾的library中。如圖2-23所示。同時也可以把cpp*檔與h*檔放在要編寫的Arduino軟體旁邊，Arduino軟體後旁邊就會顯示cpp*檔與h*檔的密碼檔，如圖2-24與圖2-25，了解cpp*檔與h*檔的程式碼可協助使用者了解其背後的原理。

(四)可開始在軟體的程式碼編輯處開始書寫需要的指令，編寫好程式以後可按左上的指令做驗證。確認程式碼是正確可順利傳輸的，最後按驗證鍵右邊的上傳鍵將程式碼上傳給Arduino板執行指令。如果程式碼有錯誤不能上傳的話，下面黑色的那一塊訊息窗格會告知錯誤，並給予修正方法(如圖2-26)，等修正正確後再上傳。

(五)如果程式碼驗證沒問題，上傳也順利的話，Arduin軟體右上角有一個序列埠監控視窗，打開會跑出一視窗，會顯示出給予該序列埠的指令。以本計畫例子來說，希望寫完程式碼後可以得到加速度值，所以上傳後打開序列埠就能看到加速度值在固定時間會顯示 (如圖2-27)。

關於MMA7660在Aduino上的程式碼，以及cpp*檔與h*檔的詳細內容將放在附錄二、附錄三、附錄四做參考。

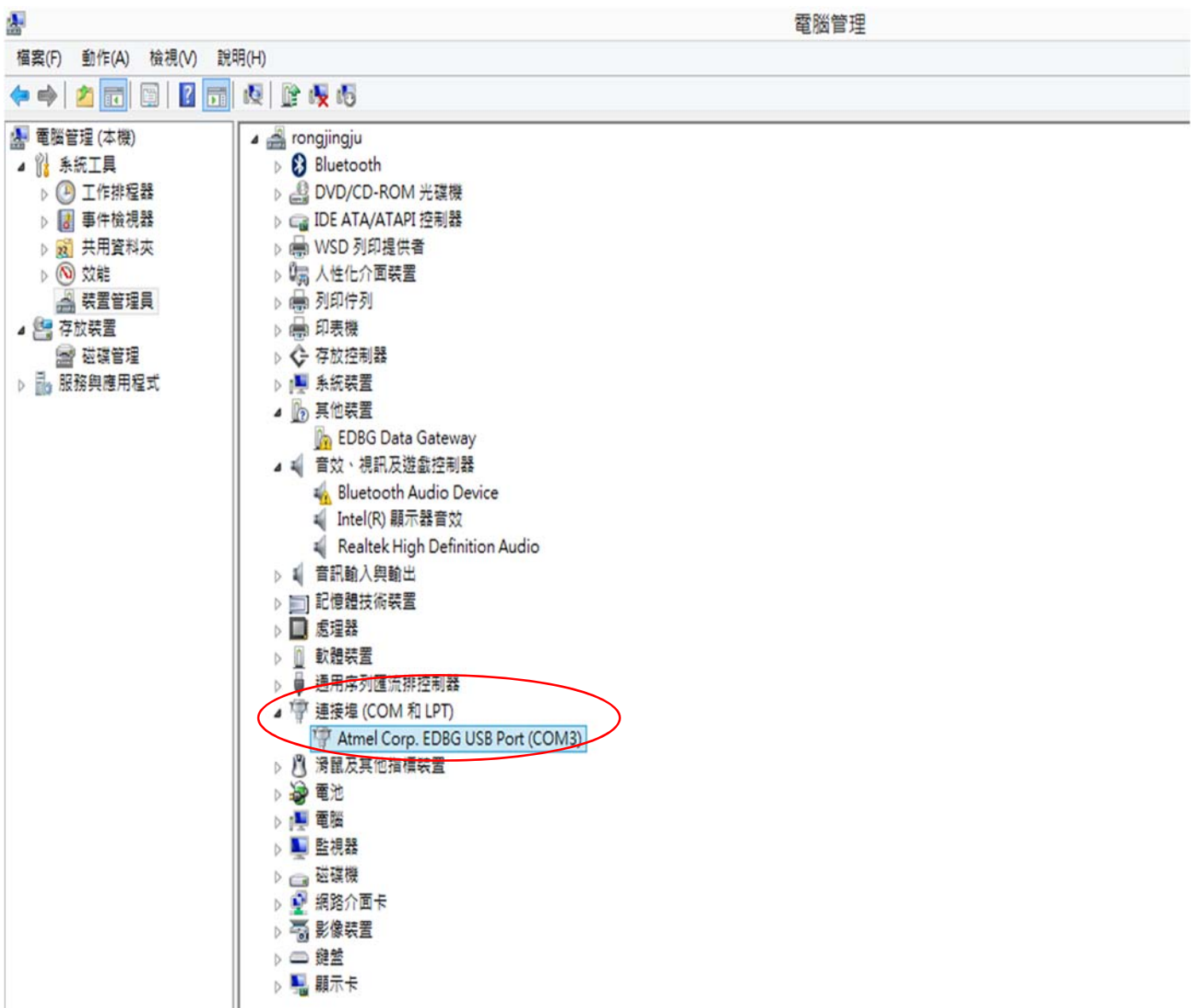


圖2-21 Arduino板與電腦連接後序列埠顯示與確認之圖示

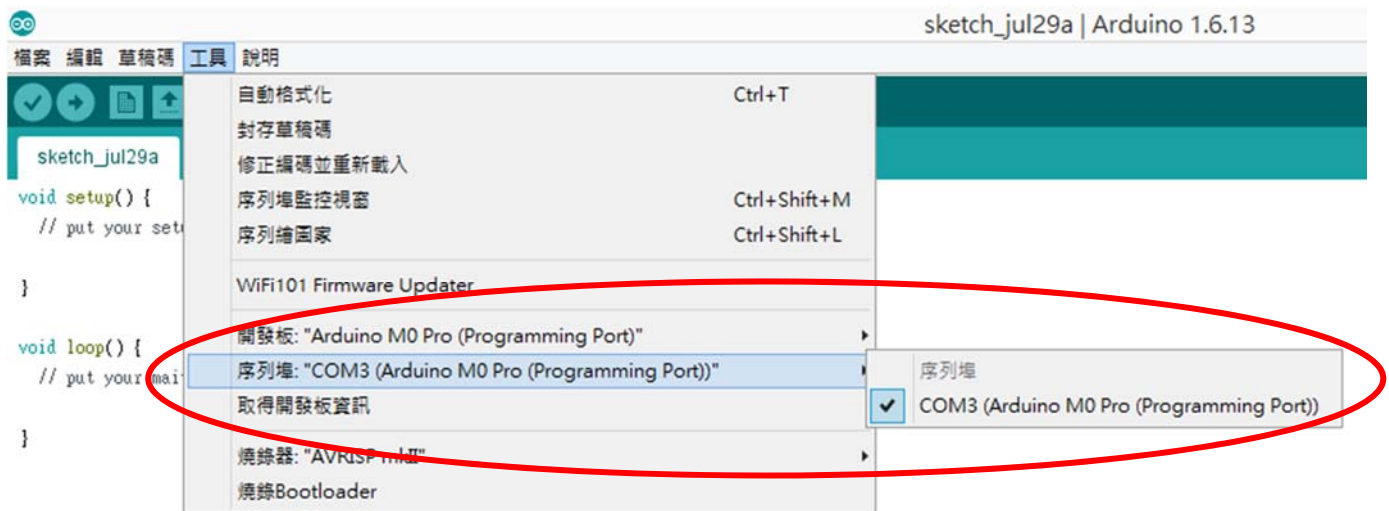


圖2-22 Arduino軟體寫程式前確認序列埠與開發板之選擇示意圖

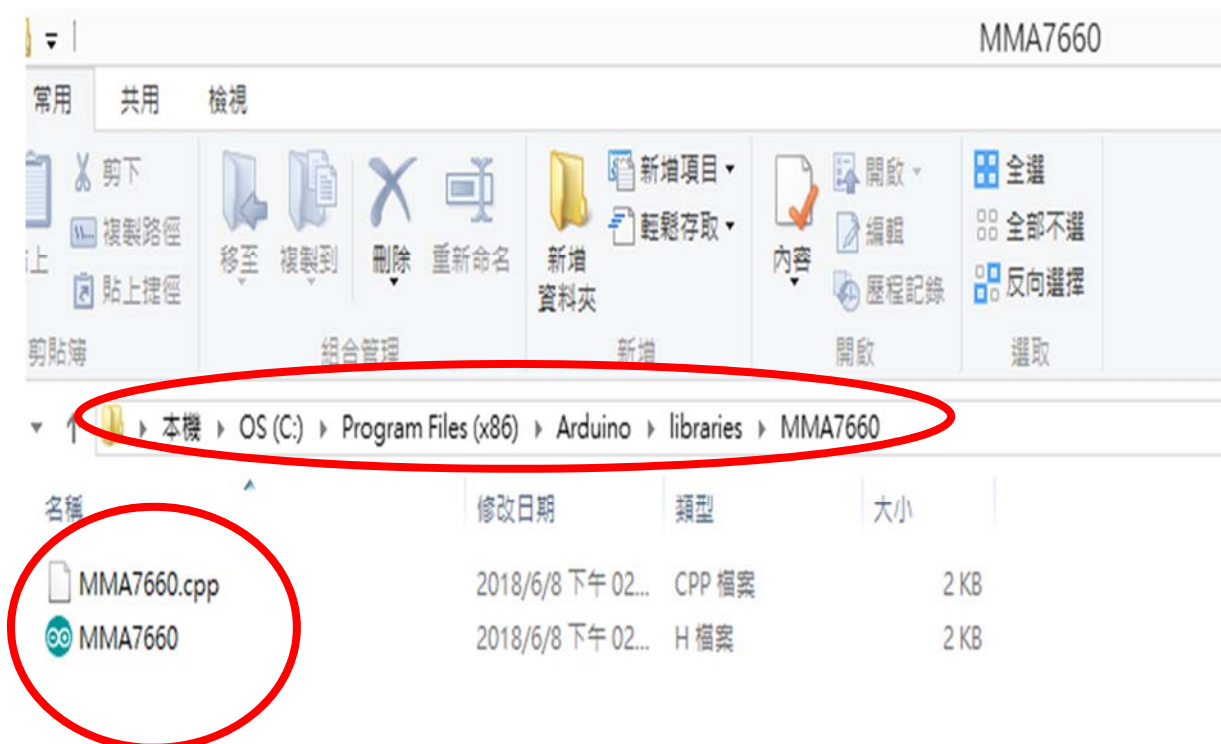


圖2-23 將C槽的Arduino資料夾裡的library放入cpp*檔與h*檔之圖示



圖2-24 cpp*檔與h*檔放置在Arduino檔旁邊以便顯示程式碼之圖示

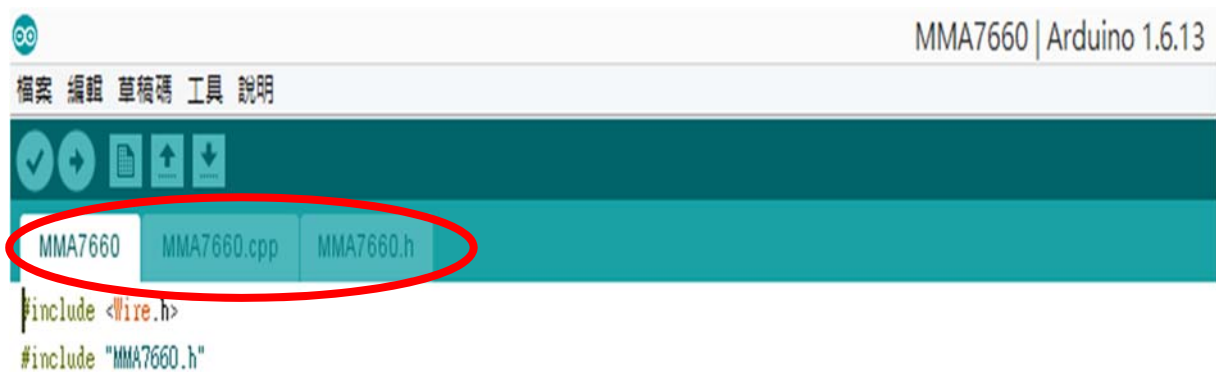


圖2-25 cpp*檔與h*檔可顯示其程式碼之圖示



圖2-26 程式完成後之驗證與上傳資料軟件

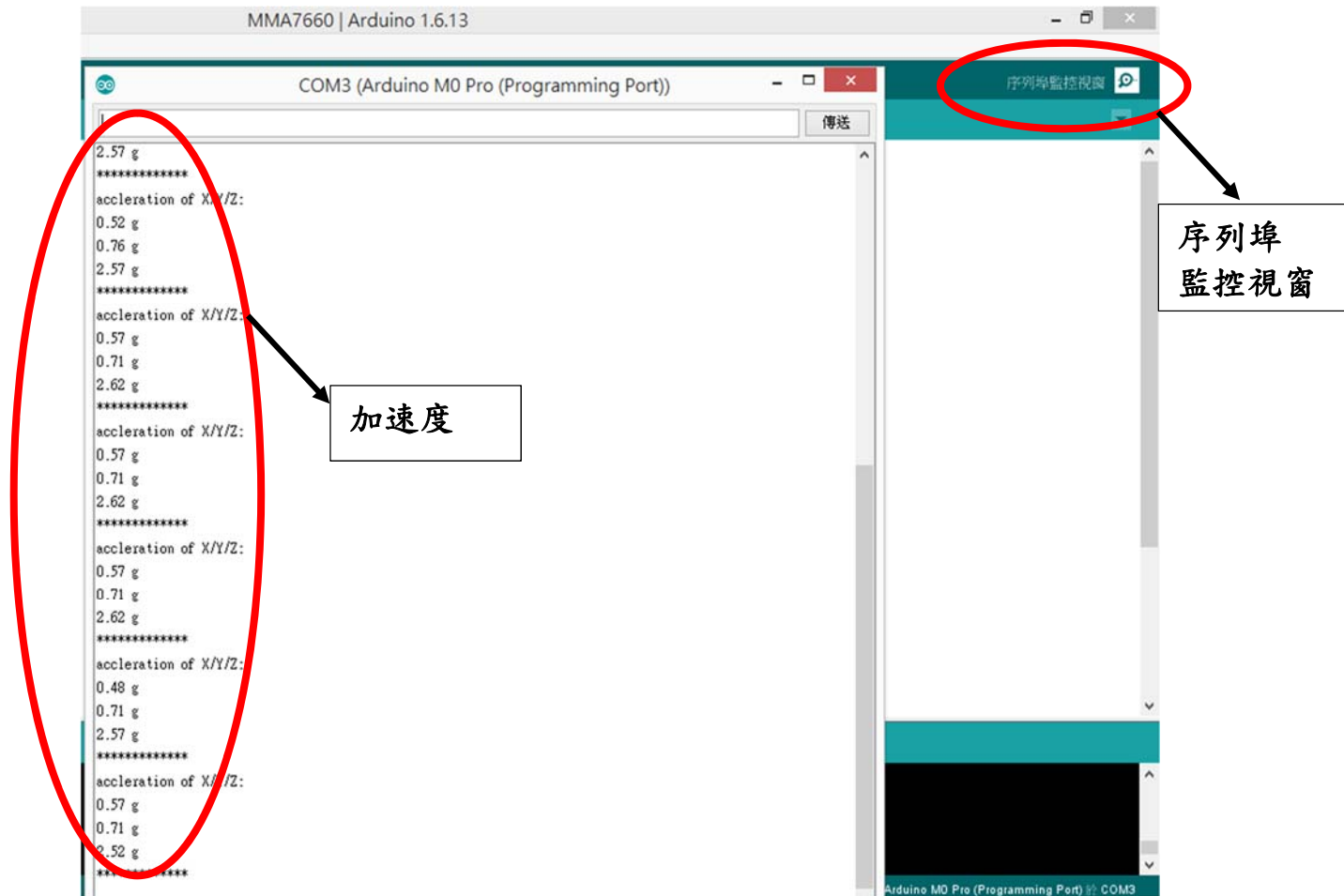


圖2-27 點選序列埠監控視窗看到加速度值之圖示

二、Arduino UNO板與MMA8451Q加速度計的結合

加速度計所量測到的加速度值，仰賴壓電效應，其原理是通過量測後在特定水晶類型承受應力時，所產生的電荷量而判斷振動大小。測試結構的加速度會傳導到加速度計內部的振盪質量，進而在壓電水晶上產生等比例的施力。這項施加在水晶上的外部應力會接著產生高阻抗壓電或電荷模式加速度計需要藉由外部放大器或內部電荷轉換器來放大產生的電荷、降低輸出阻抗以便相容於量測裝置，並且有效減少對外部雜訊來源與串音的磁化率。其他加速度計則是內建高電荷靈敏度放大器。此放大器可接收穩定電流源，再根據壓電晶體上變化的電荷而改變阻抗。這些感測器稱為整合式電子壓電（IEPE）感測器。針對可為放大器提供內鍵的電流激發，接著透過加速度計輸入之間的電壓變化，即可量測到阻抗內的變化。

MMA8451Q加速度計也使用I²C串行通信和外部交換數據，它可通過中斷信號指示新的採樣序列可用，也可通過設置使用中斷信號指示設備的移動、自由落體、瞬變、方向、單/雙擊。I²C總線使用三根信號線進行通信，分別是SCL、SDA和SA0，外部上拉電阻需要將SDA和SCL接到VDDIO腳位上（內部電源供電1.62~3.6V），當總線空閒時，這兩根線表現為高電平狀態；總線傳輸開始由START信號觸發，START信號定義為，當數據線從高電平跳變到低電平，而時鐘線SCL仍然保持高電平。由主機發送START信號過後，I²C總線被認為從空閒狀態進入忙碌狀態，緊接著START信號後主機發送的字節，前7位用於指示從機地址，第8位用於指示數據方向是讀出（1數據為從機到主機）還是寫入（0數據為主機到從機）。地址發送完畢後，總線上的所有從機將自己的地址和總線上接收到的地址進行比較，地址匹配的設備即為主機選中設備。第9個脈衝為應答響應，發送設備必須在這個ACK週期內釋放數據線，而接收設備則需要在ACK週期的時鐘高電平期間將數據線拉低。SCL線是高電平時，SDA線由低電平向高電平切換，這個情況表示停止條件，停止條件將終止發送器的發送，如圖2-28，加速度計與Arduino板的SDA、SCL、GND、3.3V腳位互相連接。

在連接Arduino開發板後，其實網路上已經有不少創客提供給MMA8451Q加

速度計使用的第三方函式庫 (Library)，我們只需將這些函式庫下載後匯入到 Arduino IDE 內即可設定加速度計初始設定，量測後的三軸加速度為原始資料，函式庫內也有提供函式來轉換成重力加速度，也因此節省創客們在加速度計上的開發時間。

關於在 Arduino IDE 的監控視窗裡讀取 MMA8451Q 加速度計的方法與介紹，可參考 Arduino M0 Pro 板與 MMA7660FC 加速度計結合的步驟敘述，配合圖 2-21 至圖 2-27，序列埠的部份需配合 Arduino 板與電腦連接時顯示的序列埠編號，並且需寫好適合 MMA8451Q 的程式碼就可顯示出所需要的加速度值。

如果要在 Arduino 開發板紀錄感測元件接收的數值，除了透過網路模組送出以外，開發板本身並不像樹莓派有外接 SD 或 Micro SD 卡的介面，可以儲存在記憶卡上，若要寫入資料到 SD 卡上，就必須透過 SPI 的介面，才能將感測的資料寫入 SD 卡。在使用 Arduino UNO 板會配合 SD 卡將資料作儲存。SD 卡裝上的讀卡機開發模組如圖 2-29 所示，讀卡機相關程式碼指令內容會放在附錄五供參考。

在 Arduino IDE 中，有內建 SD 模組函式庫可以使用，格式化 SD 卡時，可以選擇 FAT16 或是 FAT32，這兩種檔案系統都是被支援的。首先初始設定 `SD.begin(CSpin)` 函式，CSpin 為 Arduino 連接 SD 模組的 SS 或 CS 的 pin 腳，設定完成後，使用 `SD.open(filepath, mode)` 函式將 SD 卡開啟並且選擇模式，模式分為兩種，一種為讀取模式，另一種則是寫入模式。若 SD 卡內欲創建一個檔案寫入，在內建函式庫中所提供的檔案格式僅 .csv 格式和 .txt 兩種文字格式可以使用。

從 SD 卡中讀取出來的數值，由於因為是從這裡個文字檔格式所讀取，因此數值的資料型態會是字元資料型態，字元資料型態可以透過 C 語言的 `atof()` 函式，進而把資料型態轉變成浮點數資料型態，才能使用這些由感測元件得來的數值做進一步的運算。

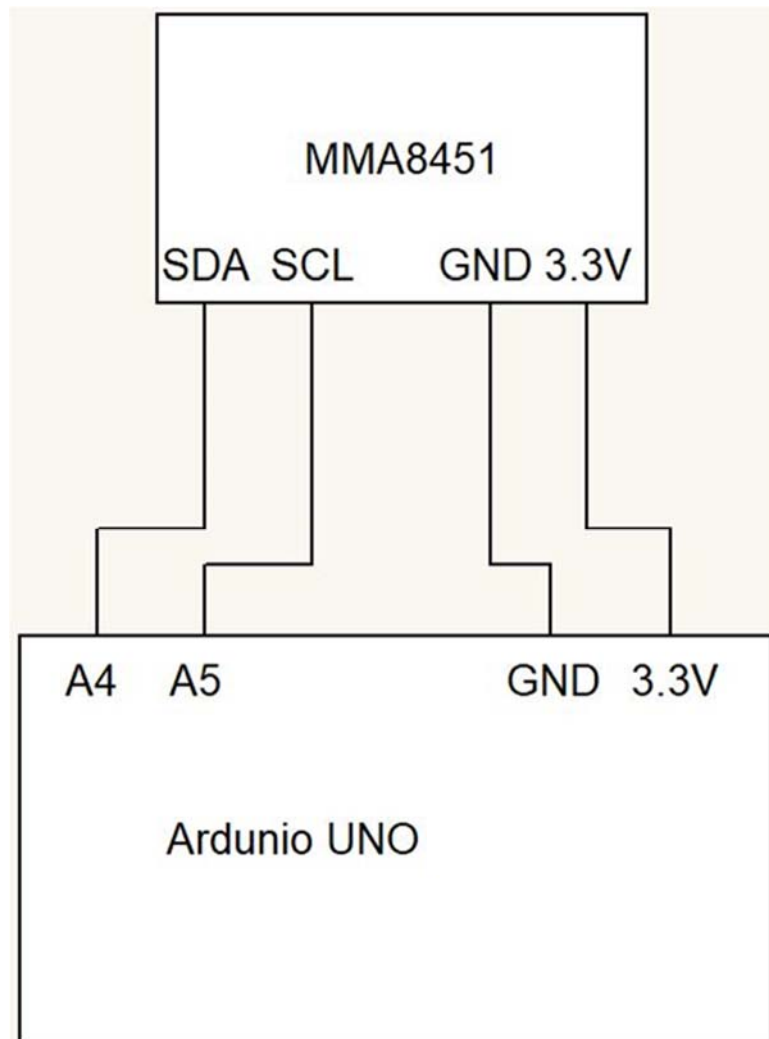


圖2-28 Arduino UNO板與MMA8451Q結合示意圖



圖2-29 讀卡機開發模組

第三章 物聯網(IoT)的應用與開發

第一節 物聯網之建置與評比

因為水保設施的安全評估通常需赴現場進行，即時性亦有待加強，為避免親赴現場而耽擱時程，期望藉由無線網路將加速度資料回傳，得到加速度與位移的資訊，提供主管單位人員維護管理補強時之參考。因此在回傳資料這部份的技術，需要藉由無線網路通訊的功能與工具來協助達成目的。

現今有許多無線網路技術，如Wi-Fi、ZigBee、Bluetooth、超寬頻，以及近場通訊NFC（Near Field Communications）與窄帶物聯網(NB-IoT)等等。表3-1是目前比較常用的無線網路方式比較表，可以發現到WiFi與Bluetooth兩種的網路傳速的速率是比ZigBee與NB-IoT還要來的高，但是ZigBee與NB-IoT在訊號的傳輸上的距離比WiFi與Bluetooth廣，針對我們的標的物水土保持工程設施，通常設置的區域是較為人煙稀少的山地地區，雖然ZigBee與NB-IoT的網路傳速效率比較低，但以本研究來說，我們需要回傳的資料只是加速度量值，網路的傳送速率較低並未造成困擾。且NB-IoT因遠傳電信已將4G網路訊號覆蓋在全臺灣，所以在4G網路訊號可覆蓋的範圍內，水保設施上儀器可以藉由無線網路回傳加速度值。基於廣泛的使訊號覆蓋在臺灣的這項優點，故本次計畫將會針對ZigBee、NB-IoT的無線網路技術來做試驗。而NB-IoT有低速率（40~200 kbps）、低成本、支援大量網路節點、可靠、安全等特性，ZigBee有低速率（250 Kbps）、低成本、支援大量網路節點、支援多種網路拓撲（如星狀、簇樹狀、網狀）、低複雜度、快速、可靠、安全等特性。

物聯網(IoT)應用的範圍極為廣泛，圖3-1乃是遠傳目前使用物聯網技術應用的案例，目的為發展智慧都市，從智慧建築、物流車隊、遠距醫療、自動駕駛、智慧工廠與環境監測等。以遠距醫療與自動駕駛為例，為使系統運轉順暢，所設置的感測器需要不間斷地傳回高頻率的資料，資料傳輸量高且延時容忍度低。然而，以水土保持設施的維護管理而言，一般並不需要如此高頻率資料傳送，低傳輸量與低耗能且具有大量感測器布置潛能的物聯網較符合本計畫的要求，此技術為LPWA，在近期越來越多人使用此技術建構物聯網，從圖3-2可以了解到雖然是相

對低速率且用於網路也不像中、高速率的IoT如此快速，讓人常使用，但相對於中、高速率的IoT，可以發現LPWA使用範圍是相當廣的，中、高速率的IoT雖然速度快但使用的部分比較侷限，所以以用來建構物聯網來說，LPWA會是最適合的技術，也從圖3-3可以發現遠傳針對LPWA使用規模做預測，將是越來越多人依賴這項技術，表3-2列出三種現今較常使用的LPWA技術，其中LoRawan因為在臺灣開放的頻段切分成16頻道，而這16種頻道有互相干擾的情況，容易出現俗稱蓋臺的干擾，而Sigfox是超級窄頻(Ultra Narrow Band, UNB)的LPWA，傳送速率相當低(100 bps)，且傳送的位元耗費能量相當高，雖然便宜但效果不好也不環保，而NB-IoT雖然成本比較起來偏高，但有漸漸下降的趨勢，且4G的基地站有架設可以使訊號覆蓋在全臺，是具有相當廣發展性的LPWA，所以將使用NB-IoT來做試驗。從圖3-4還可以發現到不少國家已經部署NB-IoT，凸顯其技術的重要性。覆蓋率則為應用IoT技術於水土保持設施另一個需要關注的項目，覆蓋率低或不普及可能需要額外的網路基礎建設，增加系統建置、維護與管理的成本，不過現今基地站的設置已經覆蓋全臺，及不須花成本額外架設其他的網路來協助此計畫。



圖3-1 遠傳公司在IoT的應用案例

資料來源：遠傳IoT團隊，“NB-IoT全球快速起飛，物聯網應用全面展開。”，遠傳電信，2018。

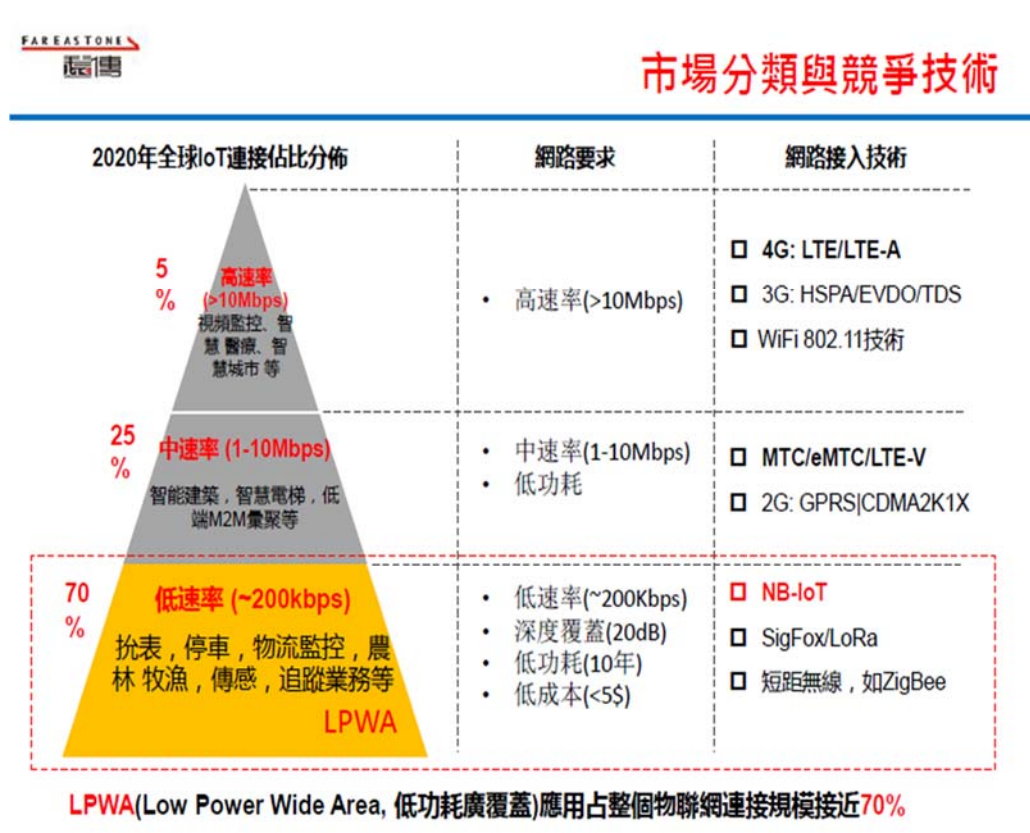


圖3-2 IoT市場分類與競爭技術

資料來源：(遠傳IoT團隊，2018)

LPWA 規模預測

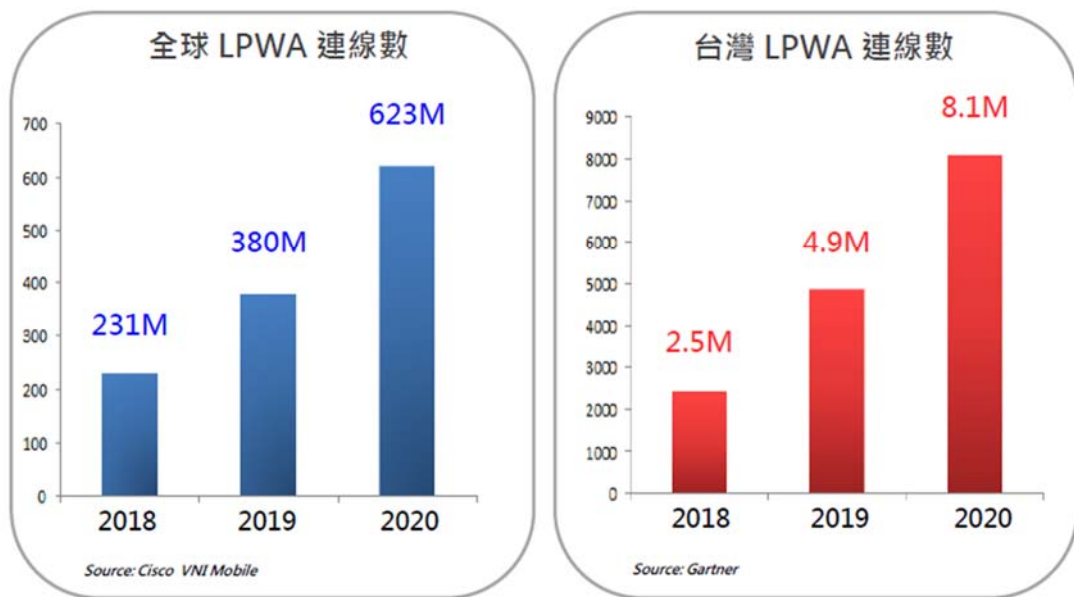


圖3-3 LPWA規模預測

資料來源：(EBU-EPSM, 2017)

2017全球nb-iot部署情況

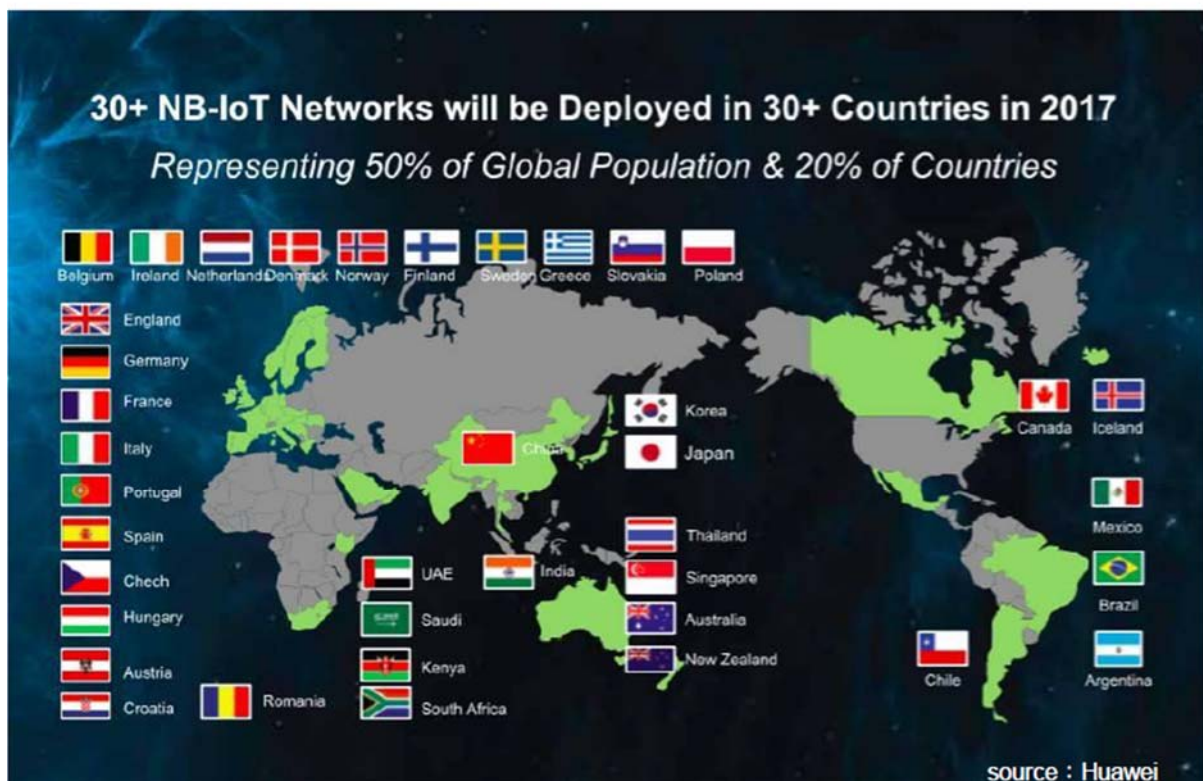


圖3-4 2017年全球NB-IoT部署情況

資料來源：(EBU-EPSM, 2017)

表 3-1 四大無線通訊比較表

	Wi-Fi	Bluetooth	ZigBee	NB-IoT
臺灣適用頻段	2412~2461MHz、 5250~5350MHz、 5470~5725MHz、 5725~5825MHz	2.4GHz	2.4GHz, 800MHz and 900MHz	703~803 MHz、 880~960 MHz
最高傳輸速率	300 Mbit/s	24 Mbit/s	250 Kbps	40~200Kbps
可傳輸距離	100-300 m	10-100 m	50-300 m	全臺已涵蓋訊號
安全性	普通	高	中	高
功耗	高	中	低	低
成本	高	中	低	低
應用範圍	手機、平板、桌電、筆電等各類可連接無線網路的裝置	手機、平板、桌電、電腦周邊、耳機、立體音頻串流等	專注於各種自動化設備、無線感應裝置、遠端遙控設備	智慧建築、物流車隊、遠距醫療、自動駕駛、智慧工廠與環境監測等

表 3-2 主流 LPWA 比較表

項目	NB-IoT	Sigfox	LoRawan
標準	3GPP	Sigfox	LoRawan
網路部屬	使用現有 4G 基站	獨立建構	獨立建網
頻段(TW)	920~945MHz、 920~945MHz Licensed band	920~945MHz Unlicensed band	920~945MHz Unlicensed band
傳輸距離	遠距離 > 10 km	遠距 10~30 km	遠距離 > 10 km
速率	40~200 kbps	100 bps	0.3~60 kbps
移動	R13 可低速移動 R14 支持高速移動	低速移動	低速移動
終端電池 工作時間	最大十年	最大十年	最大十年
模組成本	10USD 逐漸下降	2USD	5~10USD (模組)
適用情境	移動類、報表類、 控制類	定點、傳輸應用	工廠私網、區域私網
應用場景	智慧三錶、車輛追 蹤管理、人員管 理、環境監控等	低度抄錶	私網應用、環境監控
基站管理	電信公司	優納比網路公司 (UnaBiz)	私網、LoRaWAN

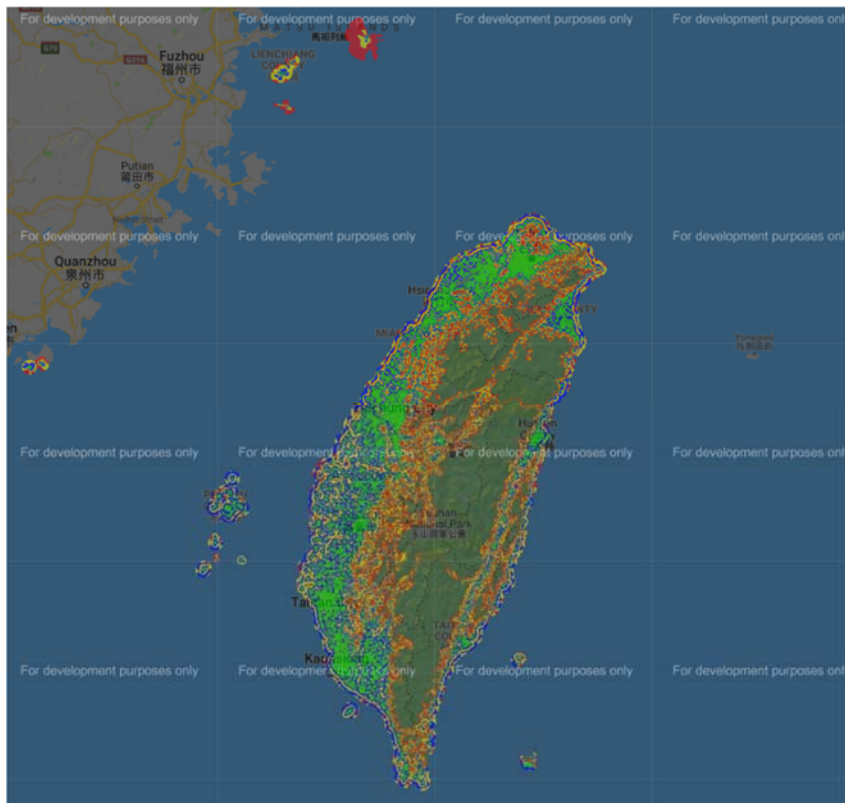
資料來源：(遠傳IoT團隊，2018)

第二節 NB- IoT應用

遠傳電信於2017年5月25日建立NB-IoT網路試點，成功的在700MHz 商用頻段進行測試，並與愛立信在NB-IoT環境下應用於智慧停車場的建置，目前NB-IoT初步以大臺北及桃園工業園區為服務基礎，並在2018年5月底將NB-IoT訊號完成本島的覆蓋。從圖3-5可以看到4G網路覆蓋在臺灣的分布，發現到通常西部地區普遍的訊號是好的，此覆蓋的程度還不到全部，畢竟本島中部的山區大部分的區域沒有覆蓋到網路，離島的網路訊號也相當弱。然而，從圖3-6可以發現到未來遠傳公司將在2020年將NB-IoT訊號覆蓋至全國，並且有5G的網路，且從圖3-7針對NB-IoT訊號覆蓋地方評估，紅色部分的地區是現在已經佈上NB-IoT訊號以及未來在2020年將會佈滿NB-IoT，與圖3-3的2017年比較可以發現到佈署訊號地區有大幅增加，如此對於更遠的地區未來要在全球佈設5G的理想，此可能性是高的。因為NB-IoT係在執照頻段內運作，資料傳輸不受干擾，只要有4.5G就有NB-IoT，其覆蓋能力為MCL (Maximum Coupling Loss，最大耦合損失)164 db，相較於其他物聯網技術，具有最大的覆蓋能力。因此，適用於農村與廣大空間(如水土設施)的佈署。雖然如此，因為水保設施多位處偏僻之處，仍有機會並無4.5G的訊號，惟NB-IoT已是目前覆蓋能力最好之技術，且應仍有為數不少的水保設施位於其範圍內，因此，雖有遺珠，仍具開發價值。除具有覆蓋優勢外，NB-IoT亦具有低功耗(Low Power, LP)的特性，NB-IoT透過DRX(Discontinuous Reception，不連續接收)與PSM(Power Saving Mode)兩項技術達到節電的目的。具體而言，NB功耗的設計目標為基於5WH(瓦時)的電池，需具有10年的使用壽命。根據上述，本計畫選定NB-IoT建置本案之物聯網，計畫期間將整合團隊的智慧型感測器與NB-IoT系統，將遠端水土保持設施的位移資訊傳送回雲端以及用戶端，本計畫期待未來可以將加速度藉由NB-IoT通訊傳回至遠端的伺服器，並於遠端將加速度轉換成位移值，以評估水保設施的安全性。

ZigBee基於無線個人區域網路WPAN (Wireless Personal Area Network) 的應用，其底層採用IEEE 802.15.4標準規範的媒體存取控制層MAC (Media Access Control Layer) 與實體層PHY (Physical Layer)。由於ZigBee的網路拓撲跟網際

網路多樣如圖3-8，具有多種傳輸途徑，每個ZigBee節點本身可以擷取資料，或傳遞來自其他節點的資料。因此當一個ZigBee網路節點被移除或是斷訊，其周遭的ZigBee裝置能夠透過ZigZag（蜿蜒曲折）的傳輸方式，如同大黃蜂那樣的連接到其目的地，這也就是ZigBee的名稱由來，是物聯網最廣泛的協定之一。



顏色	室外及室內涵蓋感受度	訊號數	下載服務連線速率參考值
	室外優 室內好	4~5格	2Mbps~25Mbps
	室外好 室內可	3~4格	2Mbps~15Mbps
	室外可 室內弱	2~3格	2Mbps~5 Mbps
	有涵蓋但弱訊區	1~2格	2 Mbps 以上

圖3-5 4G網路覆蓋圖與說明

行動通訊的跨越

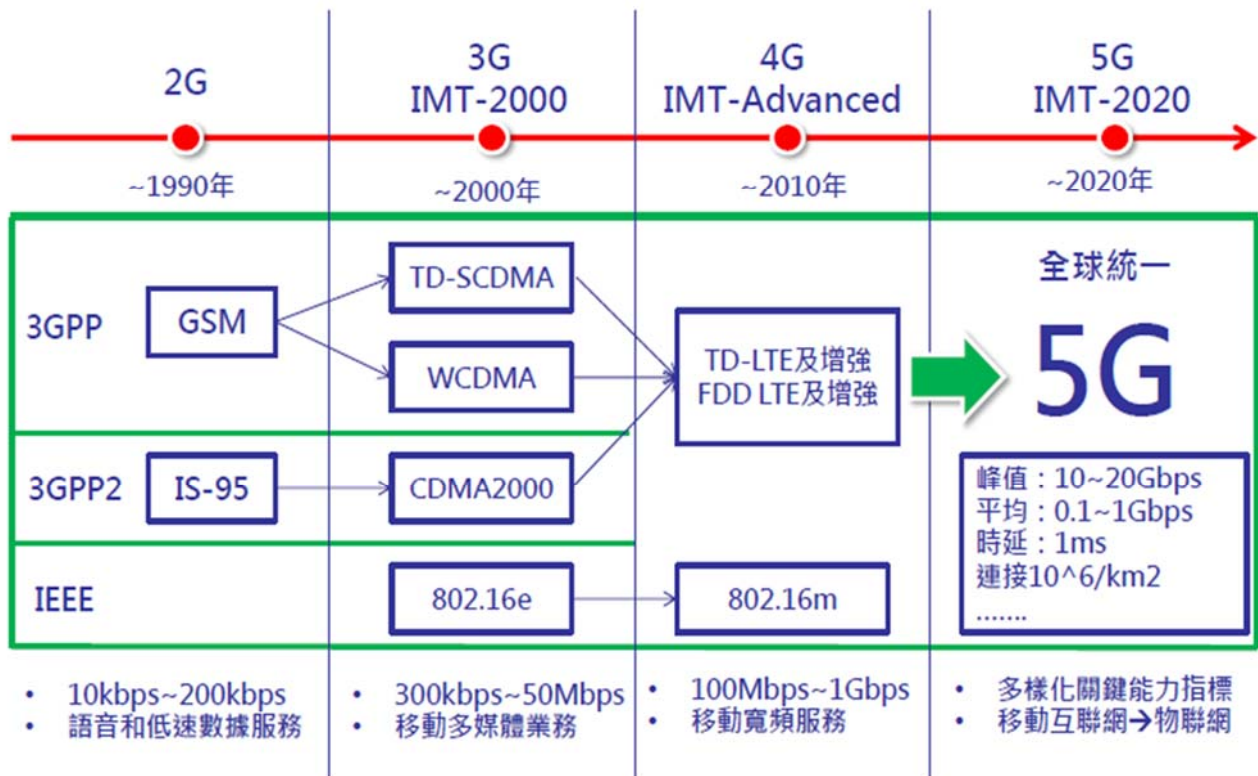


圖3-6 遠傳的行動通訊發展

資料來源：(遠傳IoT團隊，2018)



圖3-7 未來NB-IoT訊號覆蓋圖

資料來源：(遠傳IoT團隊，2018)

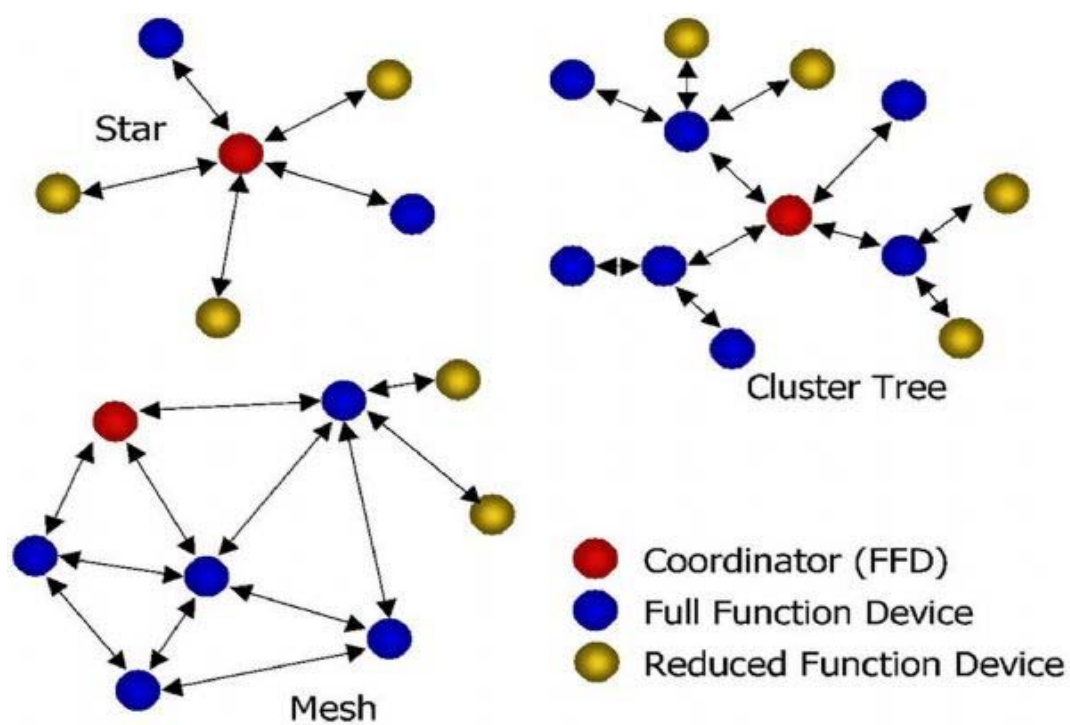


圖3-8 網路拓撲組合方式

資料來源：Embedded- cracking the code to system development, from:

<https://www.embedded.com/>

第三節 NB-IoT硬體介紹

一、窄帶物聯網(NB-IoT)硬體：SIM7000E之SIMCOM-EVB板

確定使用窄帶物聯網(NB-IoT)之後，本研究團隊嘗試找尋關於NB-IoT的模組，後來發現芯訊通无线科技（上海）有限公司（SIMCom Wireless Solutions）有發展相當多的無線模組，近期也發展了NB-IoT的模組，無線模組像是全球行動通訊系統(GSM)的模組有SIM800、SIM868、SIM808，衛星導航系統(GNSS)的模組有SIM33、SIM98、SIM68等等，而NB-IoT的模組可以發現到如圖3-9，可以發現屬於SIM7000的系列，有SIM7000A、SIM7000C、SIM7000JC、SIM7000E，這幾項模組中，圖3-9可以發現到目前SIM7000E是臺灣販售的，有代理商禾伸堂可直接聯繫並購買，所以本計畫將使用SIM7000E來做試驗。SIM7000E目前市面上有幾種板型，其中本計使用了兩種板型的SIM7000E，先是介紹SIMCOM-EVB板本：

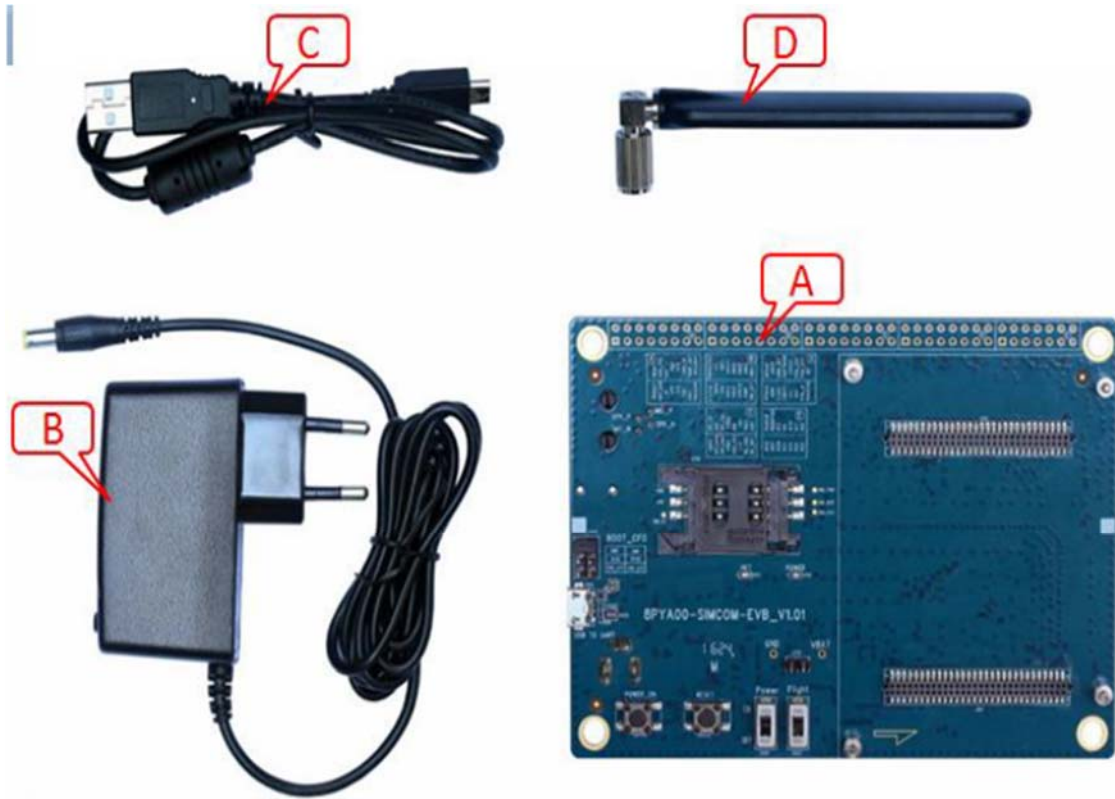


圖3-9 芯訊通无线科技（上海）有限公司發展的NB-IoT模組列項

資料來源：(SIMCom Wireless Solutions, 2018)

(一)SIMCOM-EVB板介紹

這種形式的SIM7000E有在期中報告的時候介紹並使用的，的模組元件是由SIMCOM板與SIM7000-TE板兩種板子結合而成，SIMCOM板是主要的擴展板，可連接並溝通其他的微處理器(Arduino板)與感測器(加速度計)，圖3-10是SIMCOM板(A)與其配件，為充電器(B)、USB連接線(C)和通訊用的天線(D)，充電器是提供SIMCOM板電能的重要來源，USB連接線、天線不會用在SIMCOM板上，將會安裝在SIM7000E板上做後續的使用。關於SIMCOM板的構造可從圖3-11與圖3-12為SIMCOM板的構造，分別為正面與背面，SIMCOM板使用時需先將電源充在圖3-11的P處提供電能，圖3-11的D、E兩處分別為Power的開關與RF(Radio Frequency)的開關，分別為啟動模組與啟動通訊功能的開關，B是開關，模組狀態會在I、J兩處顯示狀態判斷模組啟動狀況，而為了通訊需要SIM卡做為媒介，SIM卡可以在圖3-11的H與圖3-12的N處安裝皆可。SIM7000E板的安裝位置在圖3-11的F、G兩凹槽。SIMCOM板的功能簡述至此，細節將會在後面的SIMCOM板與SIM7000-TE兩塊板子結合與使用說明作詳細介紹。



A: SIMCOM-EVB

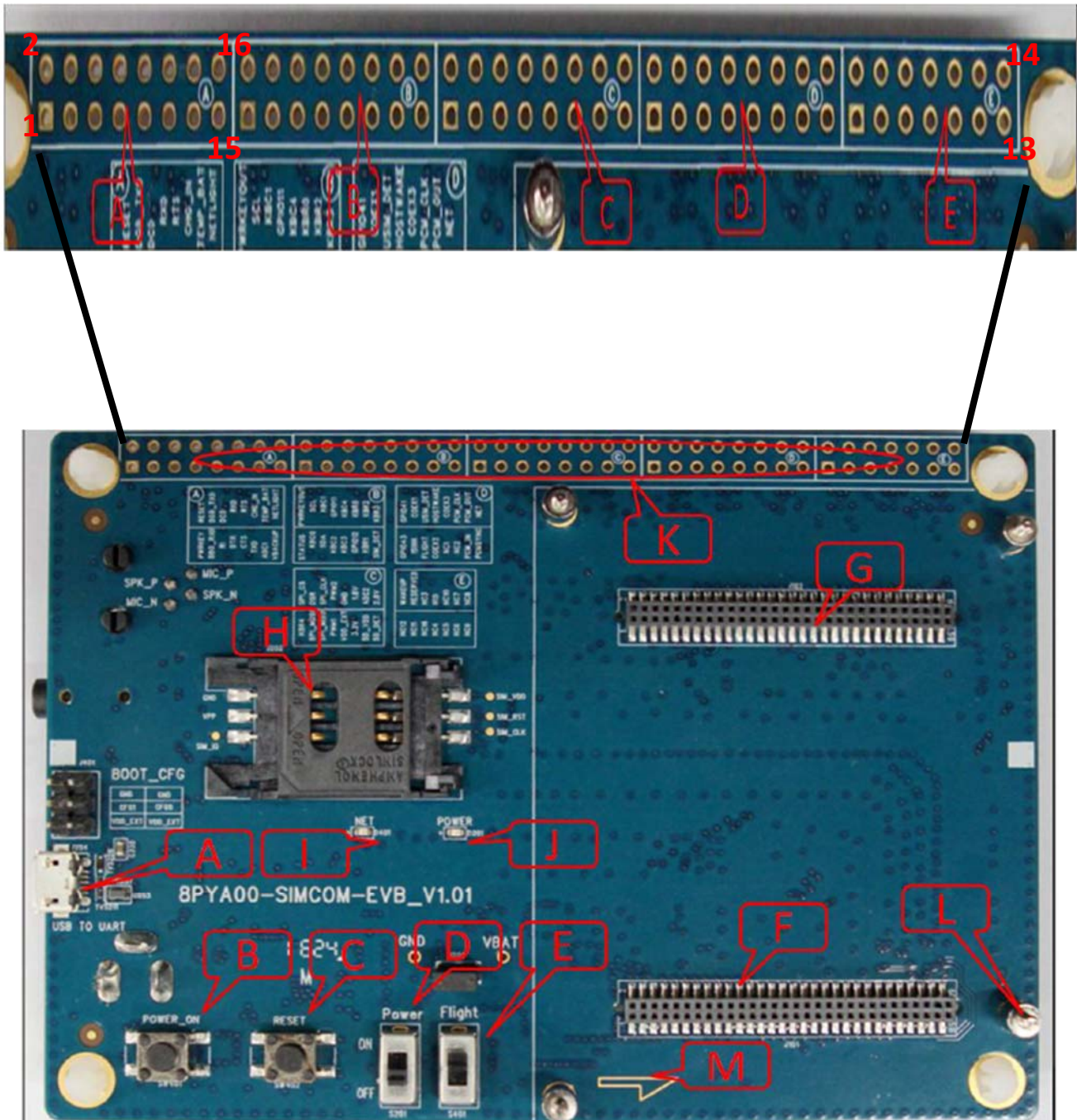
B: 5V DC adapter

C: USB Cable

D:GSM/WCDMA /LTE antenna

圖3-10 SIMCOM板與其配件

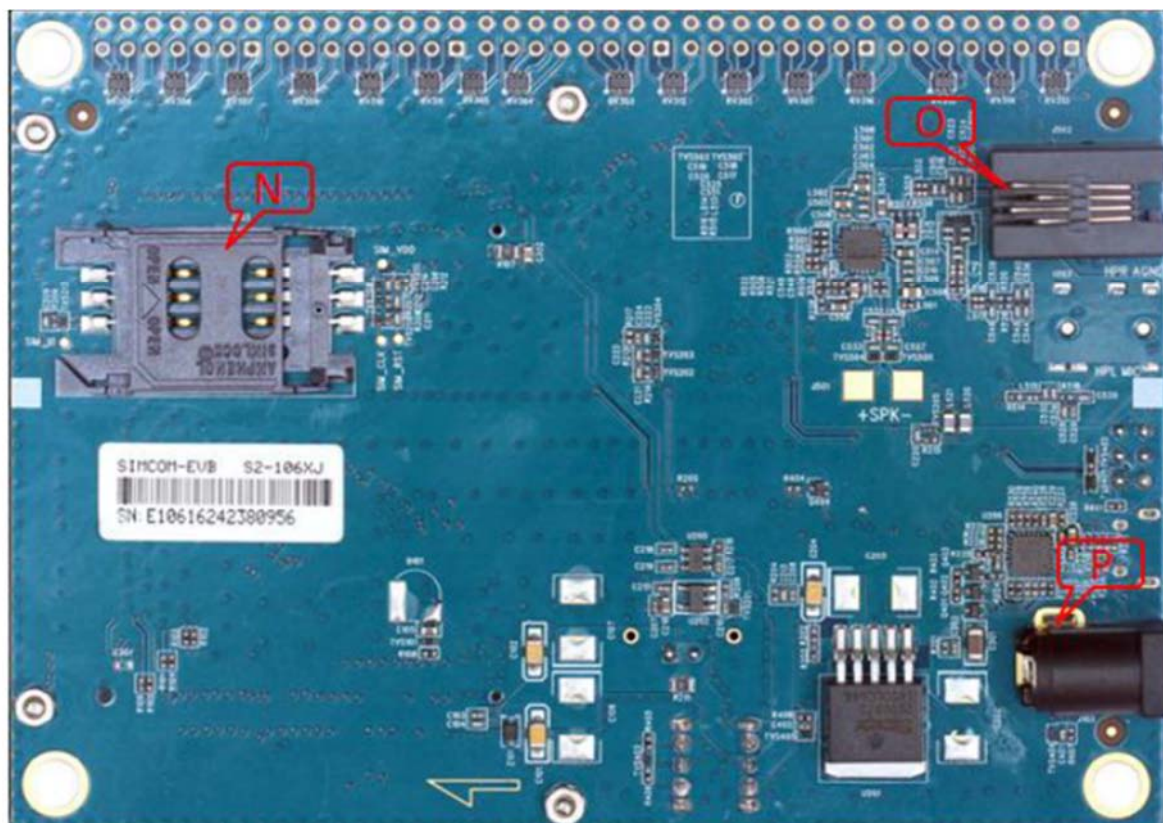
1.SIMCOM板：圖3-11的K區域是與其他感應器或者模板連接的腳位，一共5塊區域分別為A、B、C、D、E，各腳位的詳細內容如表3-3至表3-7，每一區域的代號順序從左邊下面往上數，並漸漸往右，代號也跟著增加(參照圖3-11)。表3-3介紹圖3-11的K區A範圍的各個腳位，其中有本計畫這次會用到傳訊的Rx、Tx腳位，還有其他的一些資料偵測、資料傳送終止、溫度檢測、連接時間模組等等腳位；表3-4則是介紹圖3-11的K區B範圍各個腳位，大部分的腳位為安裝鍵盤使用，也有偵測SIM卡存在與否的腳位；表3-5則是介紹圖3-11的K區C範圍各個腳位，C範圍有些伏特腳位與接地腳位，是本計畫也會使用到的部分，其他腳位還有偵測記憶卡或提供電力給記憶卡的功能或者提供SPI介面的用品連接的腳位等等；表3-6則是介紹圖3-11的K區D範圍各個腳位，該範圍的腳位幾乎都是跟通訊有關的，RF射線的監測與WiFi的連結腳位，甚至還有脈衝編碼調變(PCM)接受與傳送訊號的腳位；表3-7則是介紹圖3-11的K區E範圍各個腳位，該區域的腳位功能幾乎都是空接，也就是可以自行使用並自行開發。



- | | | |
|------------------------|--------------------------------------|---------------------------------------|
| A: USB jack | F: TE connector | J: LED indicator for Power |
| B: Powerkey | G: TE connector | K: Test Point |
| C: Reset | H: SIMcard holder 1 | L: Studs and nuts |
| D: Power switch | I: LED indicator for Netlight | M: mark of TE Module direction |
| E: RF switch | | |

圖3-11 SIMCOM板之構造(正面)

資料來源：(SIMCom Wireless Solutions, 2018)



N: SIM card holder 2

O: Handset jack

P: Power jack

圖3-12 SIMCOM板之構造(背面)

資料來源：(SIMCom Wireless Solutions, 2018)

表3-3 SIMCOM板A介面各腳位介紹

Test point A Pin description:			
Pin	Signal	I/O	Description
1	PWRKEY	I	Power on key
2	RESET	I	Reset key
3	DBG_RXD	I	Receive data
4	DBG_TXD	O	Transmit data
5	RI	O	Ring Indicator
6	DCD	O	Data carrier detection
7	DTR	I	Data Terminal Ready
8	RXD	I	Receive data
9	CTS	O	Clear to Send
10	RTX	I	Request to Send
11	TXD	O	Transmit data
12	CHG_IN	I	Charge in detect
13	ADC1	I	ADC input
14	TEMP_BAT	I	Temperature detect
15	VBACKUP	P	Battery for RTC
16	NETLIGHT	O	LED indicator for NET Light

資料來源：(SIMCom Wireless Solutions, 2018)

表3-4 SIMCOM板B介面各腳位介紹

Test point B Pin description:

Pin	Signal	I/O	Description
1	STATUS	O	Module working on indicate
2	NC17		
3	KBC0	I	KEYPAD input
4			
5			
6	KBC1	I	KEYPAD input
7	KBC2	I	KEYPAD input
8	GPIO11	I/O	GPIO
9	KBC3	I	KEYPAD input
10	KBC4	I	KEYPAD input
11	GPIO12	I/O	GPIO
12	KBR0	I	KEYPAD input
13	KBR1	I	KEYPAD input
14	KBR2	I	KEYPAD input
15	SIM1_DET	I	SIM detect
16	KBR3	I	KEYPAD input

資料來源：(SIMCom Wireless Solutions, 2018)

表3-5 SIMCOM板C介面各腳位介紹

Test point C Pin description:

Pin	Signal	I/O	Description
1	KBR4	I	KEYPAD input
2	SPI_CS	O	SPI Chip Select
3	SPI_MOSI	O	SPI Data output
4	DSR	O	Data Set Ready
5	SPI_MISO	I	SPI Data input
6	SPI_CLK	O	SPI Clock output
7	PWM1	O	PWM output
8	PWM2	O	PWM output
9	VDD_EXT	P	Power output from Module
10	GND	P	GND
11	3V3	P	3.3V Power
12	1V8	P	1.8V Power
13	SD_VDD	P	Power for SD Card
14	ADC2	I	ADC input
15	SD_DET	I	SD detect
16	2.8V	P	2.8V

資料來源：(SIMCom Wireless Solutions, 2018)

表3-6 SIMCOM板D介面各腳位介紹

Test point D Pin description:			
Pin	Signal	I/O	Description
1	GPIO43	I/O	GPIO
2	GPIO41	I/O	GPIO
3	ISINK	AI	Ground-referenced current sink.
4	COEX1	O	RF synchronizing between Wi-Fi and LTE
5	F LIGHT	O	Flight mode
6	SIM2_DET	I	SIM detect
7	COEX2	O	RF synchronizing between Wi-Fi and LTE
8	HOST_WAKE	O	HOST WAKEUP
9	NC1		
10	COEX3	O	RF synchronizing between Wi-Fi and LTE
11	NC2		
12	PCM_CLK	O	PCM data bit clock
13	PCM_IN	I	PCM data input
14	PCM_OUT	O	PCM data output
15	PCM_SYNC	O	PCM data frame sync signal
16	NET_STATUS	O	NET status

資料來源：(SIMCom Wireless Solutions, 2018)

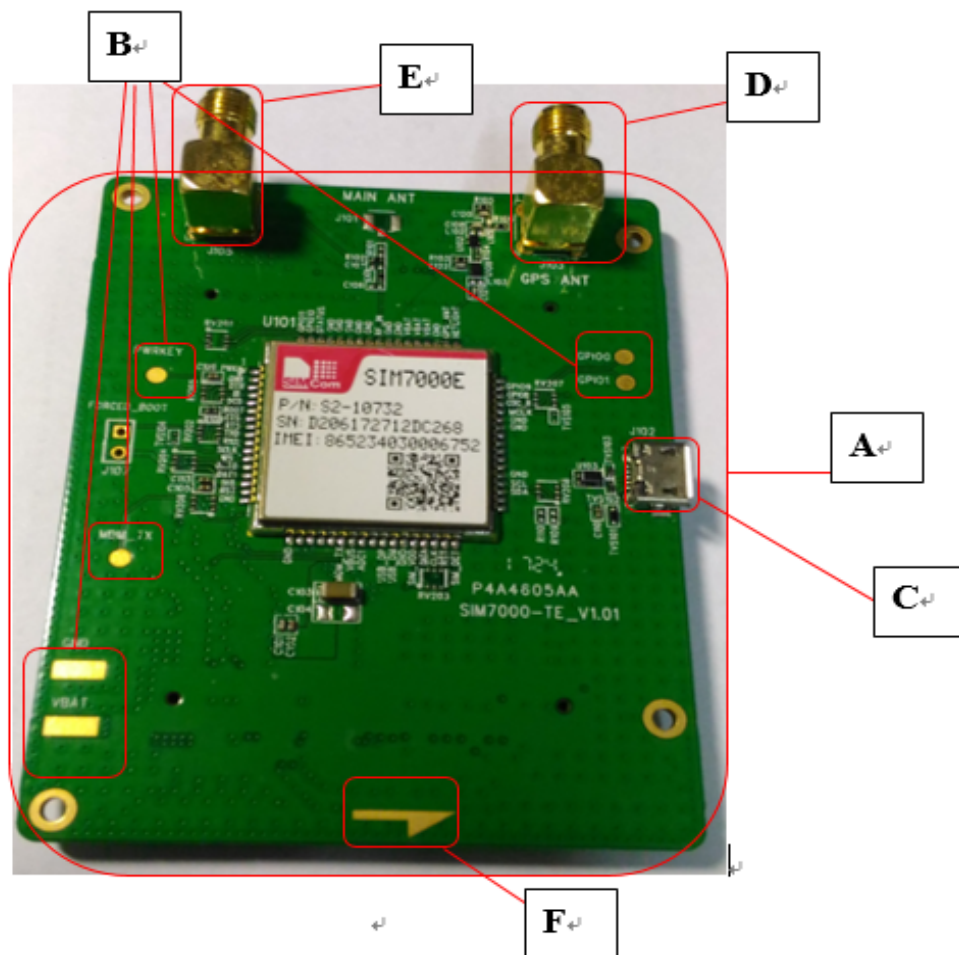
表3-7 SIMCOM板E介面各腳位介紹

Test Point E Pin description:

Pin	Signal	I/O	Description
1	NC12		
2	WAKEUP_IN	I	WAKEUP input
3	NC11		
4	RESERVED		
5	NC16		
6	NC3		
7	NC4		
8	NC15		
9	NC5		
10	NC10		
11	NC6		
12	NC7		
13	NC9		
14	NC8		

資料來源：SIMCom Wireless Solutions, “SIMCOM_EVB Kit_User Guide_V1.00.” SIMCom Wireless Solutions, 2018.

2.SIM7000-TE板：SIM7000-TE板是SIM7000E的整體核心，SIMCOM板為SIM7000E的擴展板，以便讓SIM7000E與其他感應器或其他微型處理器結合。圖3-13與圖3-14分別為SIM7000-TE板的正面與背面，圖3-13正面A處為USB接孔，D、E皆為天線連接處，D為GNSS定位的天線連接處，E為主要通訊天線連接處，傳送資料速率可達375kbps，目前現階段不會用到GNSS定位的功能，本計畫會在期末報告的室外試驗，直接在水保設施上安裝儀器時，才有可能會使用GNSS定位功能。圖3-14的G、H部分是連接圖3-9SIMCOM板上的G、F凹槽的地方。



A: Module

B: Test Points

C: Micro USB Connector

D: GNSS Antenna Connector

E: Main Antenna Connector

F: Mark of TE Module direction

圖3-13 SIM7000-TE板(正面)

資料來源：(SIMCom Wireless Solutions, 2018)

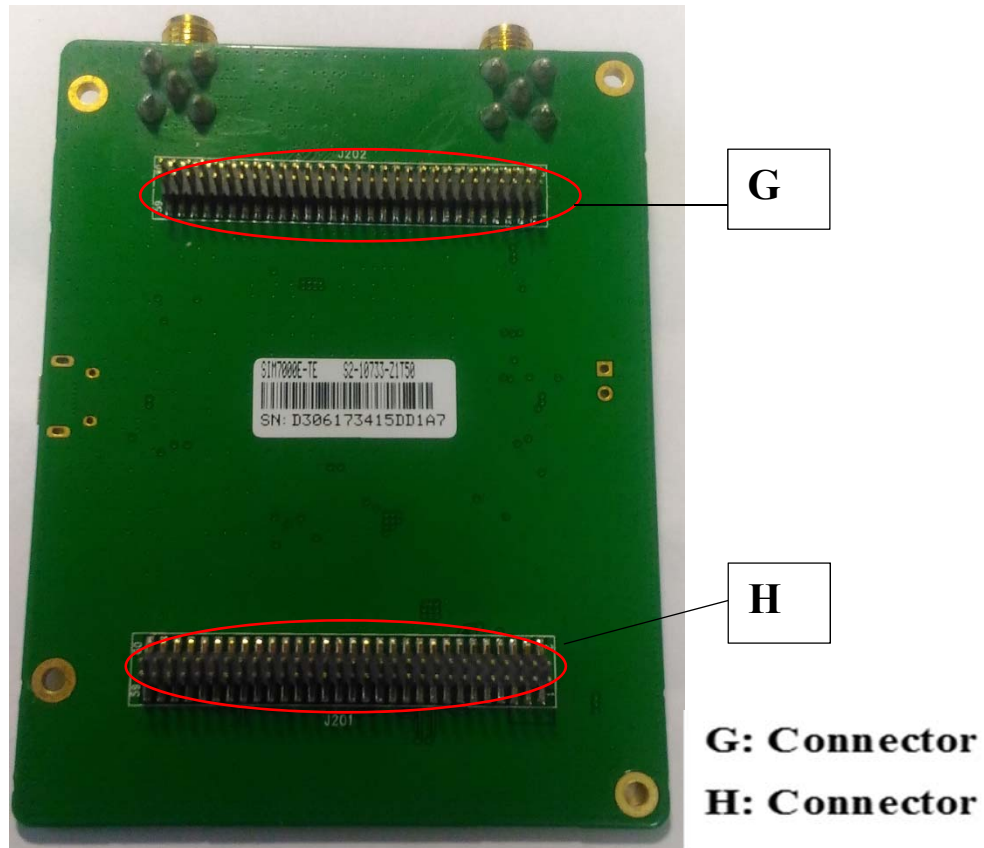


圖3-14 SIM7000-TE板(背面)

資料來源：(SIMCom Wireless Solutions, 2018)

(二) SIMCOM-EVB特徵

SIMCOM板與SIM7000E-TE板兩塊結合起來，就是完整的SIM7000E模組，SIM7000E模組完整組合如圖3-15。關於SIM7000E的特徵從芯訊通无线科技（上海）有限公司有所介紹，說明如下，以下的資料來源出處為(<http://simcomm2m.com/En/module/detail.aspx?id=168>)：

1. 一般特徵：

(1)GPRS/EDGE的訊號強度分別為900/1800Mhz

(2)輸出電力：

A. GSM900：2W

B. DCS1800：1W

(3)使用AT command控制

(4)提供伏特數範圍：3.0~4.3V

(5)操作溫度：-40~85攝氏溫度

(6)大小：24*24*2.6 mm

(7)重量：3.0 g

(8)具有GNSS (GPS,GLONASS and BeiDou/Compass, Galileo, QZSS)功能

4.資料傳輸特性：

(1)LTE CAT-M1(eMTC)模式：上傳速度375kbps，下載速度300kbps

(2)NB-IoT模式：上傳速度66kbps，下載速度34kbps

(3)EDGE Class模式：上傳速度236.8kbps，下載速度236.8kbps

(4)GPRS模式：上傳速度85.6kbps，下載速度85.6kbps

5.其他特性：

(1)USB driver：可用在Microsoft Windows的XP/Vista/7/8、Windows CE/Mobile、Linux/Android

(2)TCP/IP

(3)PPP

(4)SMS

(5)EMAIL

(6)FTP/HTTP/SSL

(7)VoLTE*

(8)FOTA*

6.各模式下電力消耗：

(1)Power off: 7uA

(2)PSM: 9uA

(3)Sleep: 1mA

(4)Idle: 11mA

7.通過證明：

(1)CE*

(2)RoHS

(3)REACH

(三)SIMCOM-EVB使用方法

本計畫希望藉由此模組來達到與Arduino板溝通，並將感測的加速度資料藉由無線通訊傳回研究室並做演算分析。至於NB-IoT的模組使用方法會在後面說明，使用方法這部分不會提到Arduino板與加速度計如何與NB-IoT模組溝通與結合，使用方法已經假設SIMCOM板、SIM7000-TE板與天線都結合好了(如圖3-15的模樣)。

- 1.將SIMCOM板的Power開關與RF開關打開，圖3-10的D、E兩處開關調成“ON”。
- 2.將充電器(圖3-10的B物件)插入SIMCOM板的充電孔(圖3-12的P處)供應電源。
- 3.將SIM卡安裝在SIM卡插槽(圖3-11的H處或圖3-12的N處)
- 4.長按“ON”按鈕啟動模組(圖3-11的B處)，此時Power顯示燈會亮紅燈(圖3-11的J處)、NET顯示燈會固定頻率閃藍燈(圖3-11的I處)，表示正常。(如果中途使用不正常想要讓模組重新啟動可按“RESET”鍵重置，在圖3-11的C處)。
- 5.過了大概三分鐘，SIM卡的網路就會出現，需連接該網路以達到無線通訊的效果。
- 6.架設TCP連線：將使用TCP&UDP Debugger工具來架設TCP的服務器與客戶端。TCP為Transmission Control Protocol，UDP為User Datagram Protocol。
- 7.網路狀態的驗證：為了確認無線通訊的效果與傳訊狀態是否正常，將會使用AT command來檢視網路狀態是否正常。
- 8.資料回傳：使用方法5的TCP連線架好了，且At command針對網路狀態的檢驗也正常的情況下，將藉由AT command的傳訊指令將加速度傳回來。

使用方法6、7、8會在後面的部分做更詳盡的介紹與結果展現。

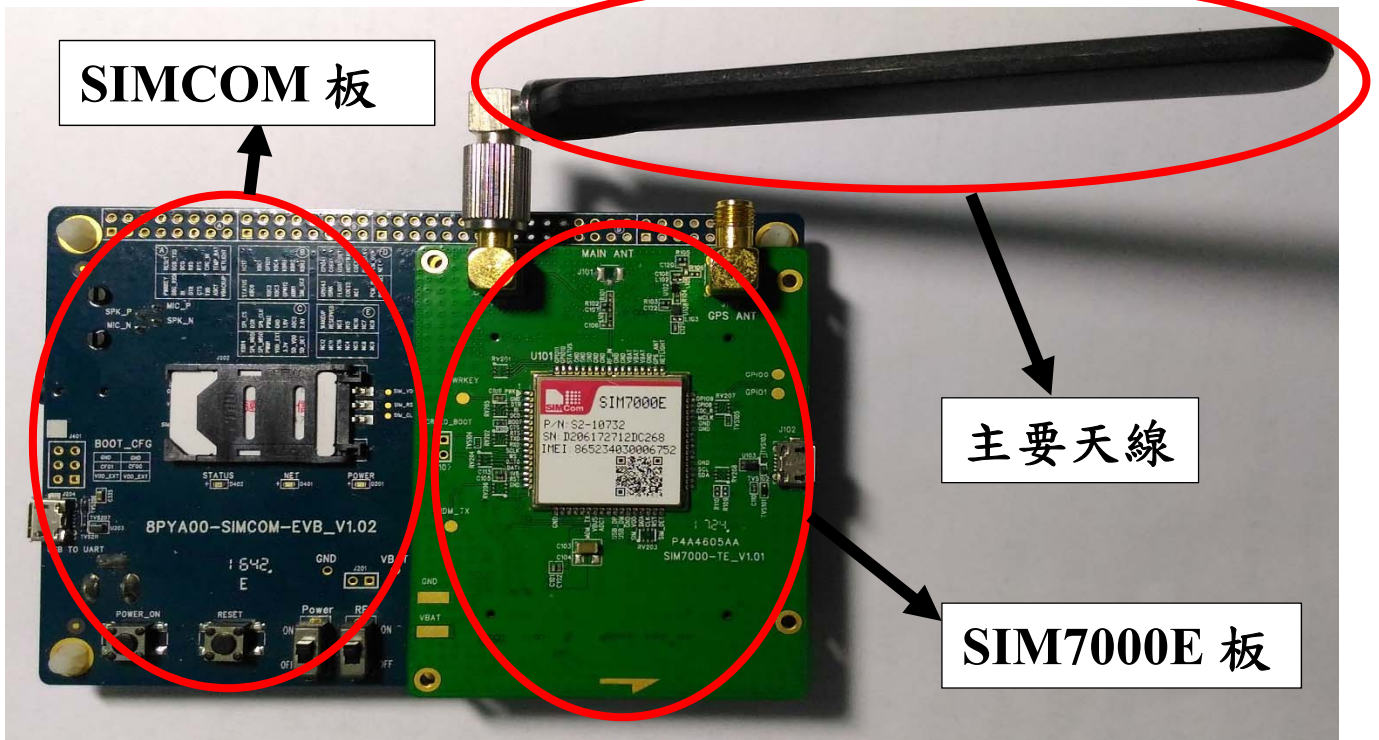


圖3-15 SIM7000E完整模組

(四)利用AT command確認網路狀態：

Ryu, Ishizaka, Kinoshita (2008)研究中提到可利用AT command確認通訊狀況與資料傳輸的工具，是一項可以確認網路連線狀態的驗證方式，以及將資料回傳到主機，在此部份我們會專注於討論AT command確認網路狀態的操作與語法，關於資料回傳到主機的操作與介紹將在後面闡述。

參考芯訊通无线科技（上海）有限公司（2017）的產品使用介紹，針對SIM7000E模組在TCP通訊協定下使用的AT command的指令，來確保TCP連線是穩定並且可使用的。上部分的SIM7000E使用方法將SIM7000E開啟並且調整成可以無線傳輸的狀態，如圖3-16，天線、SIM卡、電源供給、固定網路的主機、USB連接孔接連接完畢，也將開關打開並讓Power處亮紅燈、STATUS亮藍燈、NET在一定頻率下閃爍藍燈才表示模組已開啟並能發送與接收資料與訊號的功能。就可以開始試驗是否模組一切都狀態正常並可以發送資料，在USB連接線連接到電腦後，會有序列埠跑出，選擇序列埠與AT command有關的，以實驗的範例來說如圖3-17，在我的電腦開啟管理並打開裝置管理員，可

以發現到連接埠有四種，而COM6在圖上名稱就唯AT-port，即該埠為專寫AT command程式碼，所以就要選擇序列埠COM6。從表3-8的AT command語法可以確認SIM卡的安裝是否妥當、以及SIM卡的網路訊號強度、檢查數據業務的附著狀態，如果成功的話分別會顯示OK、CSQ值(CSQ值*2 =接收信号强度+113，接收信號強度單位為dBm)、CGATT=1跟OK。以上試驗如果皆順利，則可開始設置TCP通訊協定，如有出現ERROR或者CSQ=0或者CGATT=0的情況，可以調整一下主要天線、SIM卡安裝正常與否、模組啟動狀態是否正常等等，也可以將模組按“RESET”重新啟動來重整網路狀態。附錄六有本研究團隊寫成功的例子。

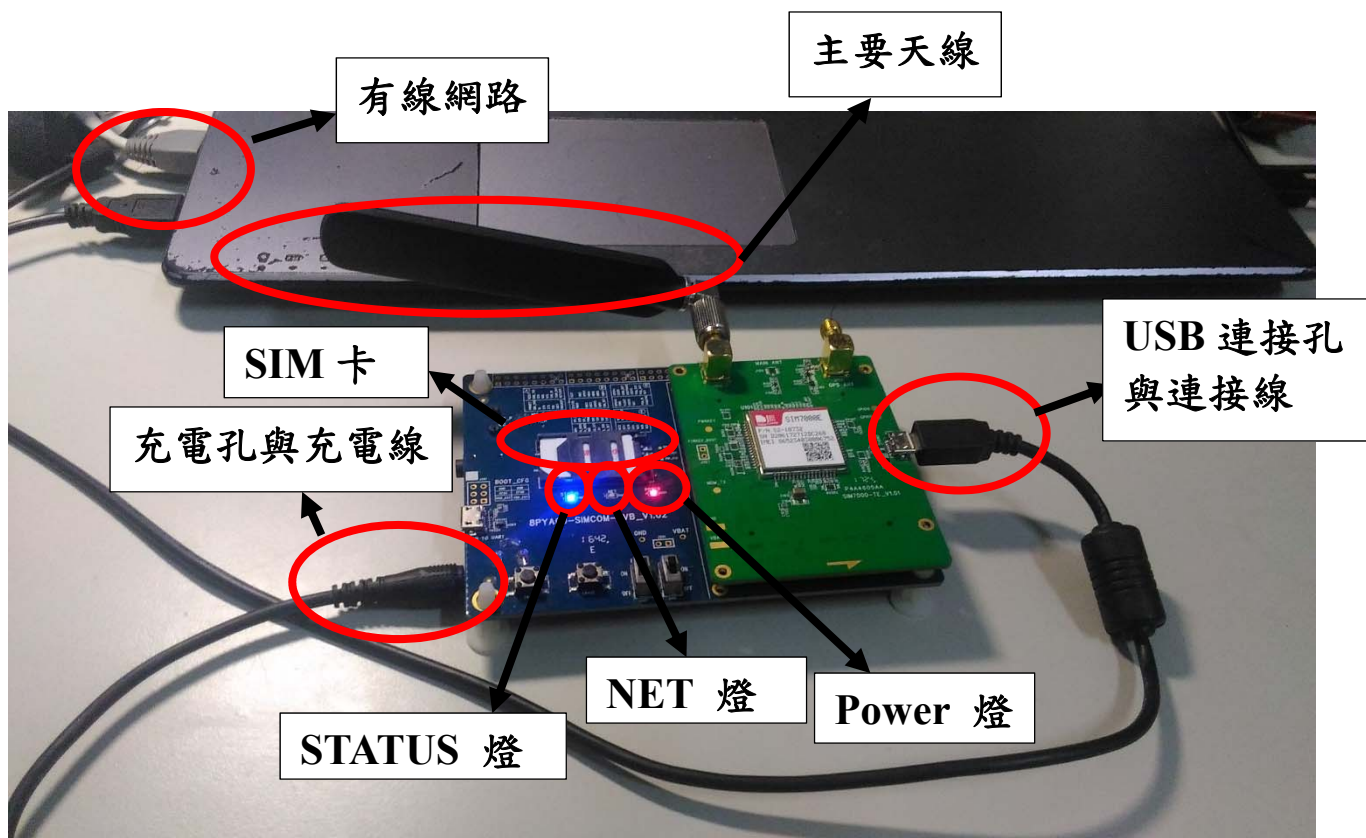


圖3-16 SIM7000E模組啟動的狀態

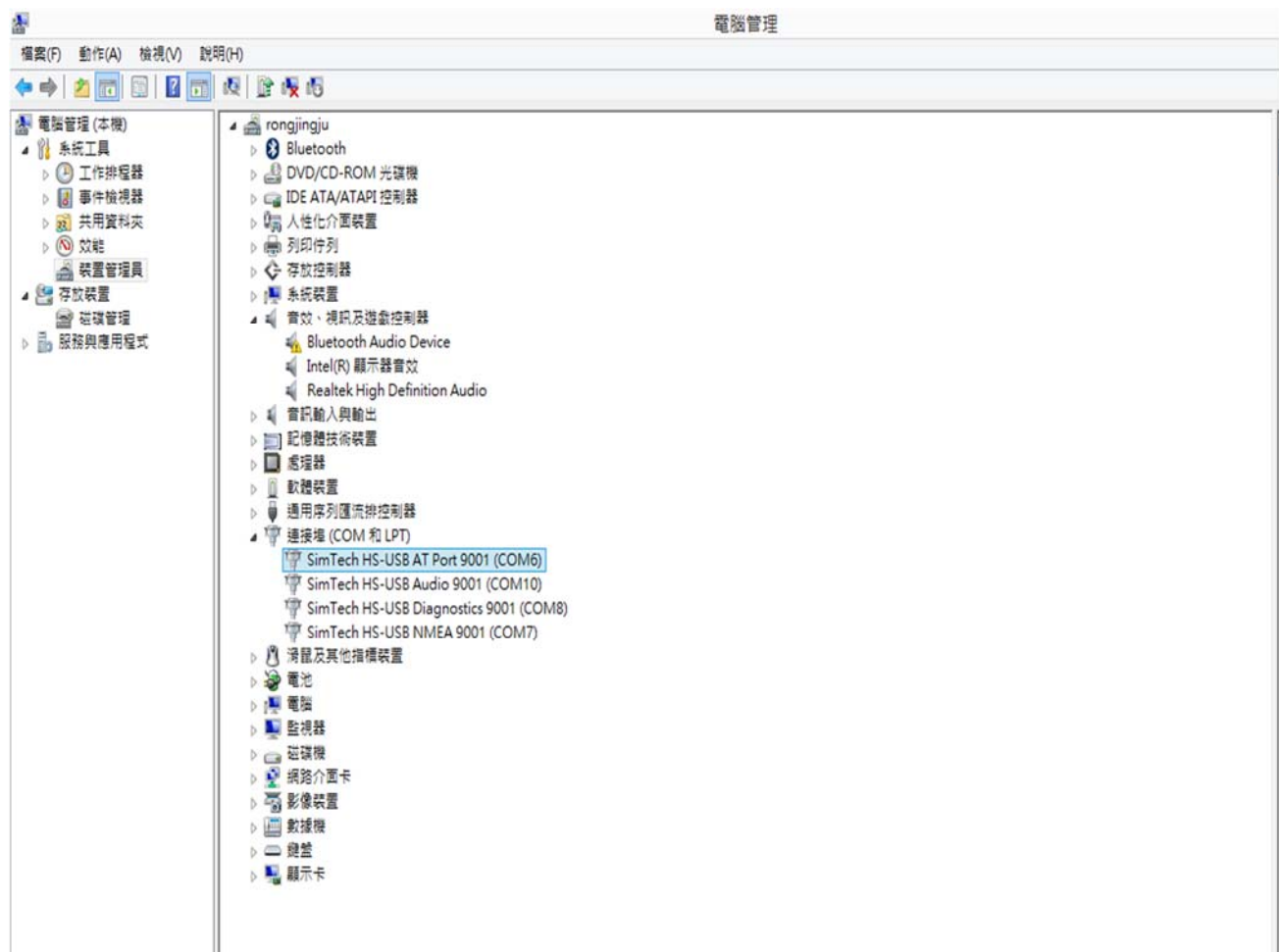


圖3-17 SIM7000E連接電腦後產生出的序列埠

表3-8 初期建立TCP與模組時通訊效果的驗證語法與說明

語法	說明	正常或驗證通過的顯示
AT+CPIN?	檢查SIM卡的安裝狀態	OK
AT+CSQ	檢查網路信號強度	CSQ:顯示出數據 (ex.13,99，表示13.99CSQ)
AT+CGATT?	檢查數據業務的附著狀態	CGATT=1 OK

資料來源：(芯讯通无线科技（上海）有限公司，2017)

(五)TCP通訊協定架設並回傳資料

欲藉由無線通訊方式回傳資料我們將會使用網路傳輸協定的技術來執行此計畫。網路傳輸的協定現今有不少的技術，大體上以DoD模型將分成四項，以下內容參考李忠憲 (2003)的文章：

- 1.鏈結層：鏈結層的機制通常由網路卡的驅動程式來提供，鏈結層可識別數據來源與傳輸目的地，並負責依照適當的通訊協定來分割、組合訊框（frame），以便有效的管理資料傳輸的進行。
- 2.網路層：主要功能是建立主機對主機的連線，是以 IP 來識別傳輸來源與傳輸目的地，讓封包(packet)能在不同的網路之間透過路由來進行傳遞。
- 3.傳送層：傳送層的主要功能是確保電腦資料正確的傳送到目的地，提供端到端的通訊服務。
- 4.應用層：主要是提供一些網路應用程式所需的通訊協定，客戶端必須傳送要求或命令給伺服器端，伺服器端則必須傳送回應或運算結果給客戶端，兩端要能溝通必須有事先定義好的通訊協定，來解讀雙方所送出的訊息。

以上的四種網路通訊協定，其中我們是要將NB-IoT模組的資料傳回主機，所以第3種的傳送層網路協定方式是我們本計畫可使用的方式。而網路協定方式的傳送層最具代表性的協定技術就是TCP(Transmission Control Protocol)、UDP(User Datagram Protocol)兩種。TCP與UDP的比較如表3-9，可以發現到雖然UDP雖然資料傳輸量大，也比較迅速，但是因為TCP的握手程序，所以資料傳輸的可靠與穩定性就提升，並且如果在資料無法送至接收端，也會有所回應，可以確認資料是否真的收到。因為可靠度與穩定度，且沒有相當要求資料的傳輸速度，以及沒有相當大的資料量要傳送，所以我們將使用TCP的無線協定的方式架設通訊協定，可透過序列確認以及封包重送機制，提供了可靠性和連接導向的服務。吳建德(2010)的研究提到TCP也為目前最常使用的通訊協定，而TCP提供的功能包括區段分割、連接導向的服務、可靠性的服務、應用程式間的流量控制和壅塞控制服務。

TCP的通訊協定架設有兩個很重要的機制就是客戶端(Client)與服務端(Server)，客戶端會將資料傳送至服務端讓服務端接收資料，服務端就是我們希望

回傳的目的地的主機，客戶端就是將資料傳出去的模組。而服務端的主機要是穩定且固定的網路位址(IP)，如果是連Wi-Fi的話將會使連到的IP不固定，傳送資料會有所困難，所以要用主機連接有線網路。

本次計畫將會使用TCP&UDP Debugger為例架設TCP通訊協定。當SIM7000E硬體已經架設好了，如同圖3-16的情況，模組安裝與調整皆妥當之後，就能夠寫AT command來架設TCP通訊協定與傳送資料。寫AT command的使用介面本計畫使用禾伸堂提供的SIMCOM Wireless Kit這項工具，這項工具也是芯訊通无线科技（上海）有限公司創造的，在買SIM7000E時就一起送過來的軟體，使用上也相當順手方便。

SIM7000系列TCP/IP應用有兩種連接模式，可以通過命令當AT+CIPMUX=0時，是單鏈路模式；當AT+CIPMUX=1時，是多鏈路模式。因為目前還沒有做到室外實驗，甚至多臺儀器一同偵測，所以目前試驗方式都是單鏈路模式。未來如果有多一點儀器就可以來試多鏈路模式並且試驗。單鏈路模式的執行，可參照表3-10的AT commands語法去寫，首先先檢查業數據業務附著狀態，如果CGATT=1，就表示正常附著，然後調整成透傳模式可將AT指令換成數據，在發送數據時模式就會讀這是一般傳輸數據，而非AT command的指令，再來設定原先就有的APN，為nbiot，接下來就發送無線傳輸的指令，並確定本地網址，以本研究目前試驗的情況來說網址是：10.25.1.73。最後輸入TCP傳到的目的地主機，該TCP連的網址以目前來說，是140.112.76.188，端口的部分設定2048，如圖3-18。而當顯示CONNECT的時候，TCP/UDP Debugger的服務器網址就會顯示有一個外在連線，如此就能從外在連線去傳資料給服務器(主機網址)，如圖3-19是從SIMCOM Wireless kit 傳一串數字，後來讓服務器收到資訊，期待未來這項技術可以在野外測定水保設施，資料回傳減低了不少人力與時間成本。關於單鏈路模式的建立成功，可在附錄七參閱。

表3-9 TCP與UDP比較

	TCP	UDP
連接特性	可靠的連接導向 (Connection Oriented)	非可靠的非連接導向 (Connectionless Oriented)
速度	慢	快
穩定性	高	低
握手程序	需要	不需要
傳輸量	較小	較大
傳輸機制可靠性	可靠的傳輸機制，確保 傳送資料到接收端	不可靠，應用程式須自行 提供可靠機制
壅塞控制	有	無
速度	較慢	較快

資料來源：(黃國鴻，2018)

表3-10 單鏈路模式TCP架設所需的AT command語法與說明

語法	說明	正常或驗證通過的顯示
AT+CGATT?	檢查數據業務的附著狀態	CGATT=1 OK
AT+CIPMODE=1	設置連接模式為透傳模式	OK
AT+CSTT="nbio"	開始任務，設置APN	OK
AT+CIICR	建立無線網路	OK
AT+CIFCR	獲得本地網址	出現網址(以我的SIM卡試驗為例，網址為： 10.25.1.73)
AT+CIPSTART="TCP","網址", "端口編號"	建立TCP鏈路 成功建立鏈接，進入數據模式	OK CONNECT

資料來源：(芯訊通无线科技（上海）有限公司，2017)

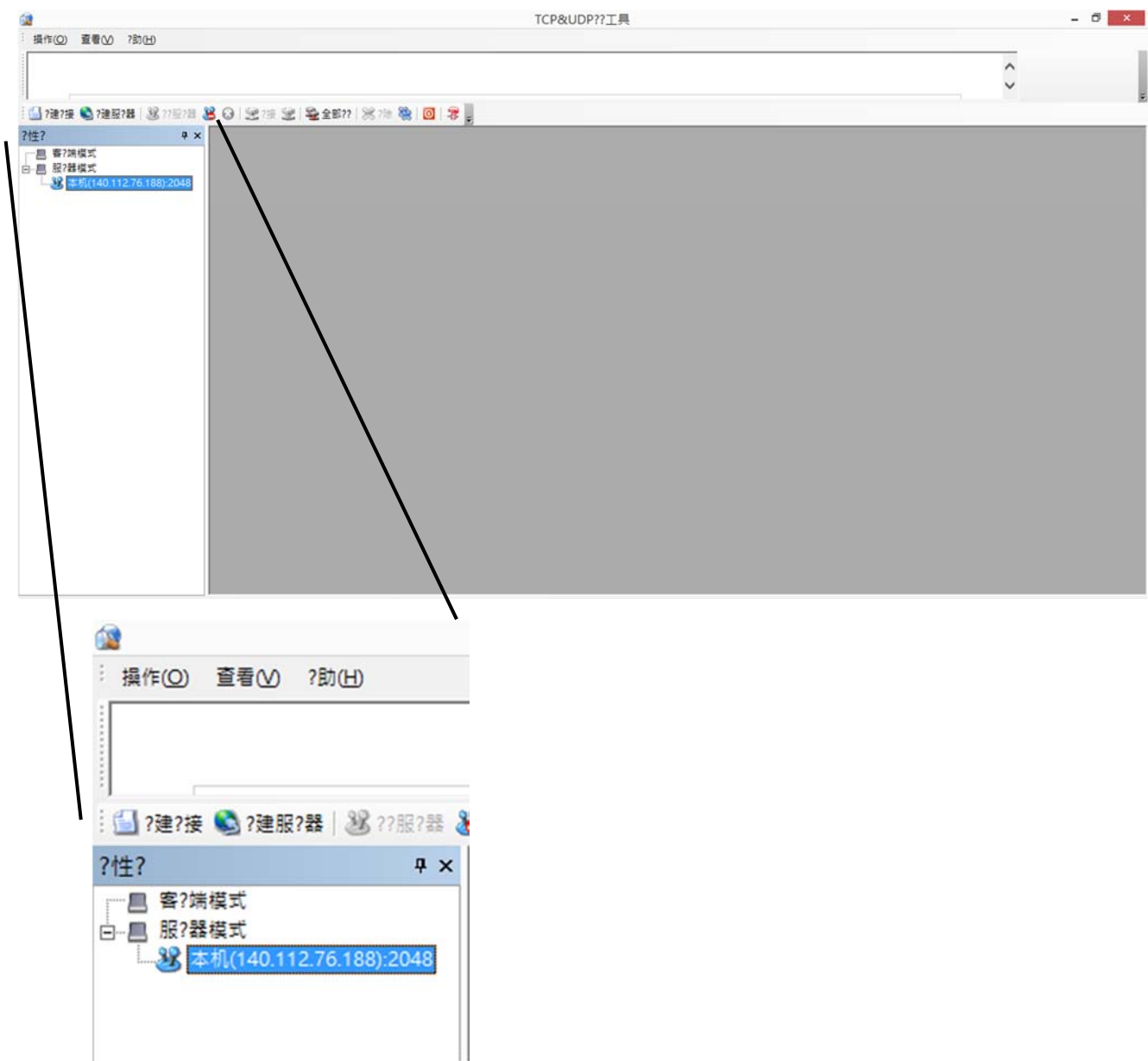


圖3-18 TCP服務端的IP與端口架設

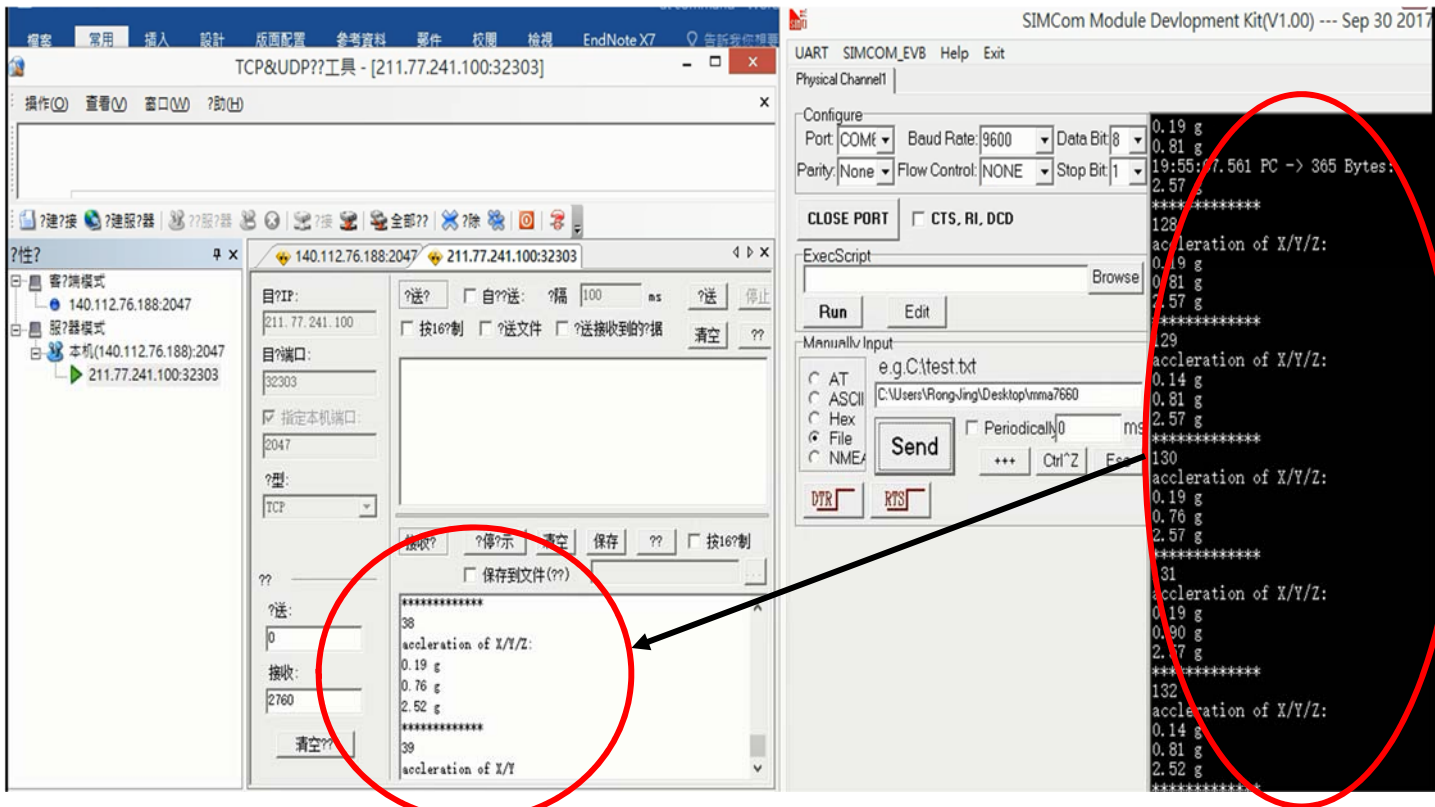


圖3-19 服務器主機藉由無線通訊收到別的位址所給的數值

(六)SIM7000E模組壽命與耗電量

先前這項計畫擬解決的問題中有提到直接去現場評估水保設施，其即時性有待加強。所以我們希望可以減少直接去現場評估水保設施的工作量。為了達成這项目的，SIM7000E模組的壽命與耗電量就是一項需要考量的因素，是否該模組可以撐得夠久，可以讓工作人員減少去現地安裝儀器或換電池的時間與勞力成本，如此也達到我們擬解決的問題的減少直接去現場的工作量。這部分的內容會參考SIMCom Wireless Solutions (2018)的文章，針對SIM7000E的電池以及模組耗電量評估其模組的蓄航力。

在做這項評估之前，有先做了三種假設，假設如下：

- 1.電池容量為6000mA/hr。
- 2.電池自我消耗推測為5%，故只會釋放原先95%的電力。
- 3.電力給予對象後，推測傳輸途中會消耗5%，故只會釋放原先95%的電力。

而SIM7000E一共有三種用電的模式，配合表3-11也會在後面計算SIM7000E的壽命與電池恆久力。

1.PSM：是新的省電模式，讓裝置能略過資料傳輸作業時的週期性監測，讓裝置切換至休眠模式延長待機時間，多由裝置來主導與網路間的通訊。此外，這種作法能更有效率地進行低功耗模式的啟動/解除，因為能讓裝置在切換至省電模式時仍維持在網路登入的狀態，因此，每次脫離省電模式後無須再花額外的時間來重設登入/連結。相關應用包括智慧儀表、感測器，以及所有週期性向網路推播資料的物聯網裝置等都可藉由PSM獲得更佳效益。計算式如下，可以發現到因為SIM7000E的傳訊功能，所以item 1與item 2一定會是耗電來源，再加上PSM模式本身的耗電狀態，耗電量乘上所需時間，總耗電量就是0.155 mA，依照假設條件1供給電量為6000 mA/hr，且假設條件2表示會消耗5%電量，所以剩下95%的供電量，供電量除上總耗電量，就是每小時的耗電量，最後做一下時間換算並加入假設條件3，電力傳輸會耗掉5%電力，所以可以得到PSM模式可以持續3.988年。

$$(61\text{mA} \cdot 6\text{s} + 100\text{mA} \cdot 1.6\text{s} + 0.009\text{ mA} \cdot 3600\text{s}) / 3600 = 0.155\text{ mA}$$

$$6000 \cdot 95\% / 0.155 / 24 / 365 = 4.2\text{ years}$$

$$4.2 \cdot 95\% = 3.988\text{ years} = 1455\text{ days}$$

2.eDRX：藉由延長聯網模式下從網路接收資料的最大時間間隔到10.24秒來延長電池壽命，而在閒置模式下，週期性監測與區域追蹤的更新時間間隔則可延長到40分鐘以上。這種模式能同步網路與裝置的休眠週期，故裝置可減少檢查網路訊息的頻率，但缺點是延遲會增加，因此eDRx特別針對各種裝置的終端應用進行最佳化。實際使用情境包括資產追蹤與智慧電網等應用皆能受益於更長eDRx週期帶來的低功耗特性。計算式如下，可以發現到因為SIM7000E的傳訊功能，所以item 1與item 2一定會是耗電來源，再加上eDRX模式本身的耗電狀態，耗電量乘上所需時間，總

耗電量就是1.146 mA，依照假設條件1供給電量為6000 mA/hr，且假設條件2表示會消耗5%電量，所以剩下95%的供電量，供電量除上總耗電量，就是每小時的耗電量，最後做一下時間換算並加入假設條件3，電力傳輸會耗掉5%電力，所以可以得到PSM模式可以持續0.54年。

$$(61\text{mA} \cdot 6\text{s} + 100\text{mA} \cdot 1.6\text{s} + 1\text{mA} \cdot 3600\text{s}) / 3600 = 1.146\text{ mA}$$

$$6000 \cdot 95\% / 1.146 / 24 / 365 = 0.568\text{ year}$$

$$0.568 \cdot 95\% = 0.54\text{ year} = 197\text{ days}$$

3.DRX：會連續性的一直接收，會關閉接收器一段時間，等過一陣子再開啟，以達到UE省電的效果。在UE沒有DRX的狀況下，只能一直開著接收器去監聽基地臺是否有作paging的動作，若有paging，UE則會開始與基地臺從RRC idle進入到RRC connected的狀態。但若是如此，UE無時無刻開著接收器，會顯得相當的耗電。計算式如下，可以發現到因為SIM7000E的傳訊功能，所以item 1與item 2一定會是耗電來源，再加上eDRX模式本身的耗電狀態，耗電量乘上所需時間，總耗電量就是1.846 mA，依照假設條件1供給電量為6000 mA/hr，且假設條件2表示會消耗5%電量，所以剩下95%的供電量，供電量除上總耗電量，就是每小時的耗電量，最後做一下時間換算並加入假設條件3，電力傳輸會耗掉5%電力，所以可以得到PSM模式可以持續0.335年。

$$(61\text{mA} \cdot 6\text{s} + 100\text{mA} \cdot 1.6\text{s} + 1.7\text{mA} \cdot 3600\text{s}) / 3600 = 1.846\text{ mA}$$

$$6000 \cdot 95\% / 1.846 / 24 / 365 = 0.352\text{ years}$$

$$0.352 \cdot 95\% = 0.335\text{ year} = 122\text{ days}$$

三種模式可以藉由AT command指令去做設定：

1.PSM模式：AT+CPSMS

2.eDRX模式：AT+CEDRXS

3. Sleep模式：AT+CSCLK

表3-11 SIM7000E耗電參數列表

Items	Case		Avr. Consumption	period(s)
1	PS->UDP send complete (200 bytes)	PSM-> UDP ready	61mA	6s
2		UDP ready-> UDP send complete	100mA	1.6s
3	PSM		9uA	/
4	EDRX (PTW=5.12s;eDRX=40.96s;DRX=1.28s)		1mA	/
5	Sleep (DRX=1.28s)		1.7mA	/

資料來源：(SIMCom Wireless Solutions, 2018)

二、窄帶物聯網(NB-IoT)硬體：SIM7000E之Arduino擴展板

前面提到芯訊通无线科技（上海）有限公司（SIMCom Wireless Solutions）有發展NB-IoT的無線模組，也有提到期中報告時使用了SIM7000E的SIMCOM-EVB版型，而後來得知代理商禾伸堂有代理的商品是專屬於Arduino板連接的SIM7000E，價錢只比SIMCOM-EVB板貴大約300~400塊，但卻可以直接與Arduino板連接，因為腳位都有設定好，彼此是可以相互連接的，所以從期中報告過後就開始使用SIM7000E之Arduino擴展板試驗。SIM7000E之Arduino擴展板介紹如下：

(一)SIM7000E之Arduino擴展板介紹

官方名稱為Botletics™ SIM7000 LTE CAT-M1/NB-IoT + GPS Shield Kit，後面將簡稱SIM7000E之Arduino擴展板。此形式的SIM7000E的模組元件可直接與Arduino連接並一體成形。圖3-20是SIM7000E之Arduino擴展板本身，雖然有充電孔與USB接孔，但因為最後還是直接會與Arduino板合併所以在本次試驗並沒使用過，而SIM7000E之Arduino擴展板有通訊最重要的天線裝置處與SIM卡裝置處，而之所以會稱為該SIM7000E為Arduino擴展板是因為兩側的腳位與Arduino板各腳位名稱是幾乎吻合的，如圖3-21可以發現兩邊的黃色區塊與紅色區塊大部分的腳位名稱都相同，故兩者可以相連接(如圖2-23)，故

SIM7000E為Arduino擴展板各腳位內容可以參考前面圖2-13、圖2-15的腳位介紹；圖3-22是該板的天線，有兩條分別為主要天線與GPS天線，而該產品不像SIMCOM-EVB提供了圖3-10的充電器(B)、USB連接線(C)，因為到時候SIM7000E之Arduino擴展板會直接接到Arduino板上(如圖3-23)，而充電的部分與USB連接區都是直接與Arduino板相連(如圖3-24)，以便給予電源以及傳輸資料給微型處理器(即Arduino+SIM7000E板)。而SIM7000E之Arduino擴展板是從SIMCOM-EVB改變而來，基本特性都沒有改變，只是把形狀改變成適合接Arduino板的模樣，故特性可直接參考SIMCOM-EVB的特徵，電池壽命也是參考SIMCOM-EVB。

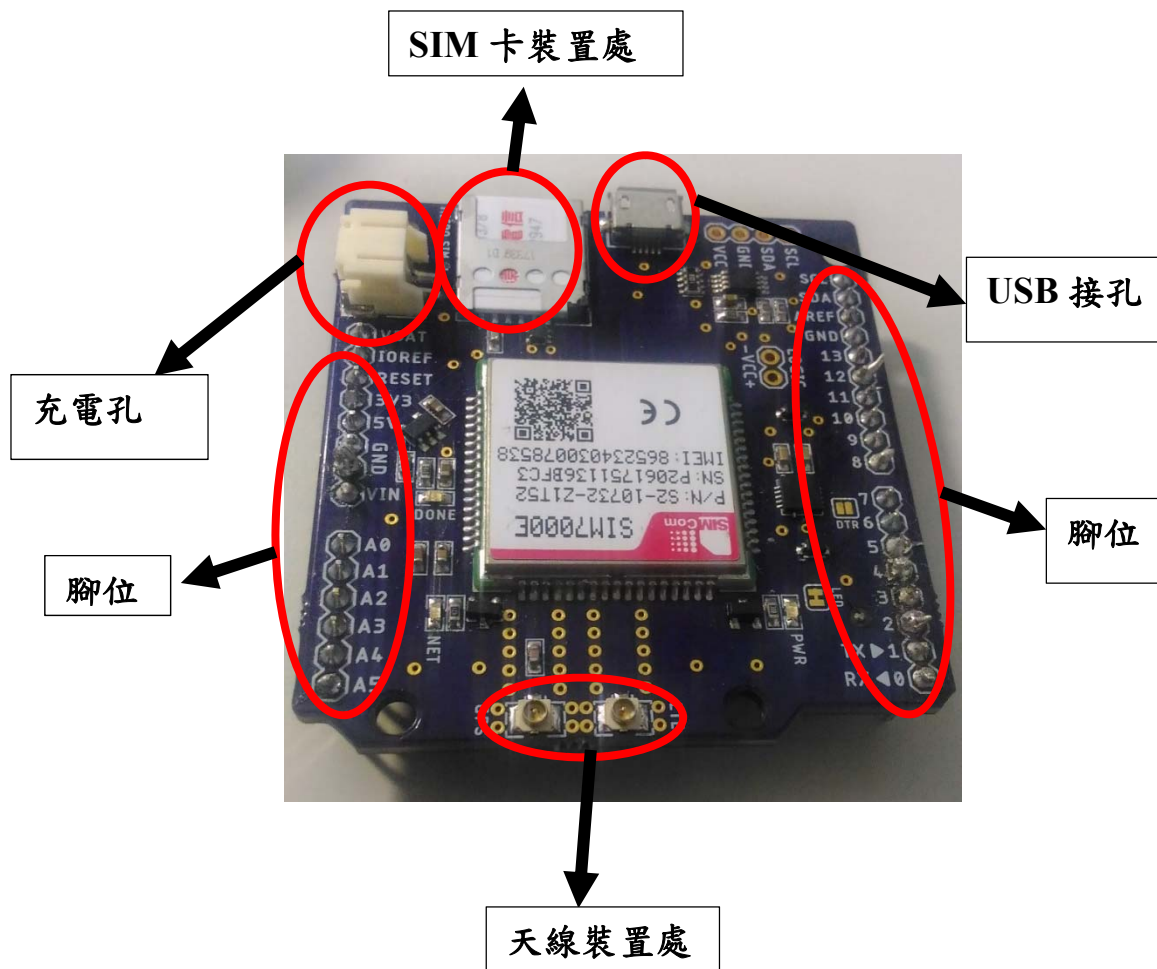


圖3-20 SIM7000E之Arduino擴展板天線

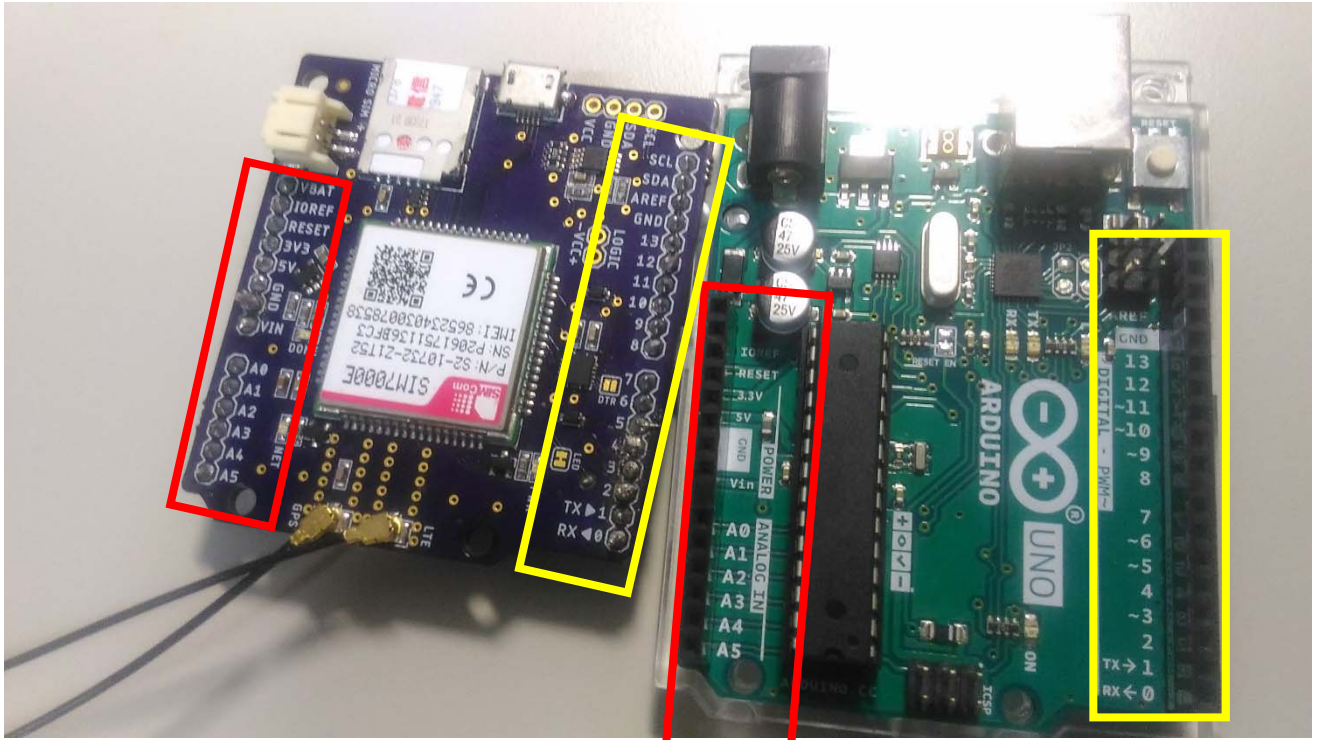


圖3-21 SIM7000E之Arduino擴展板與Arduino板之腳位比較

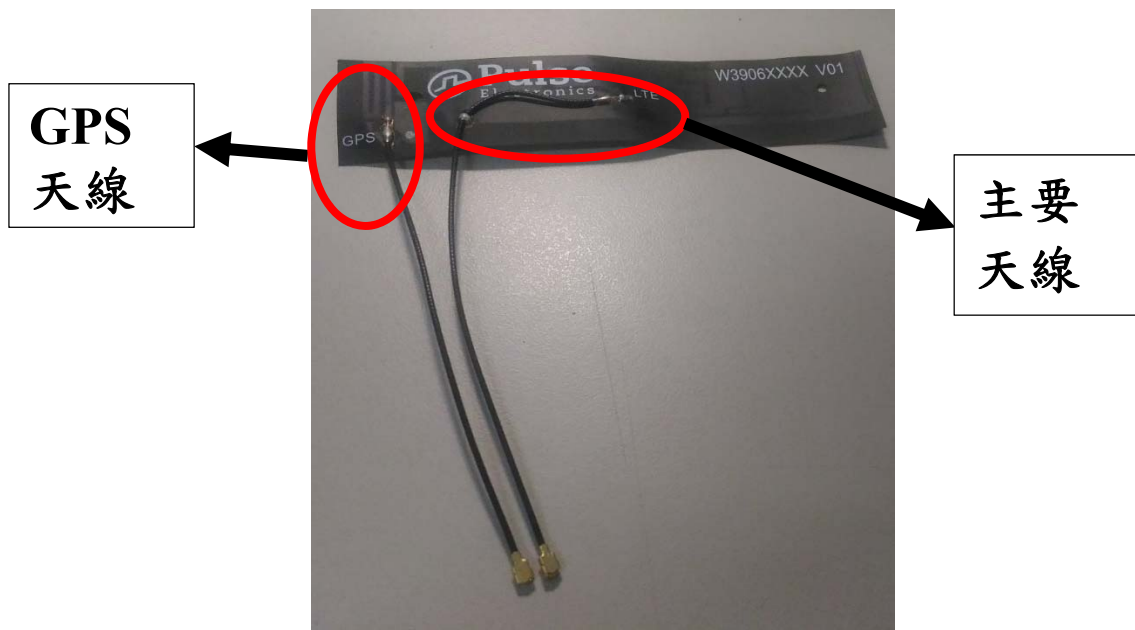


圖3-22 SIM7000E之Arduino擴展板天線



圖3-23 SIM7000E之Arduino擴展板與Arduino板連接圖

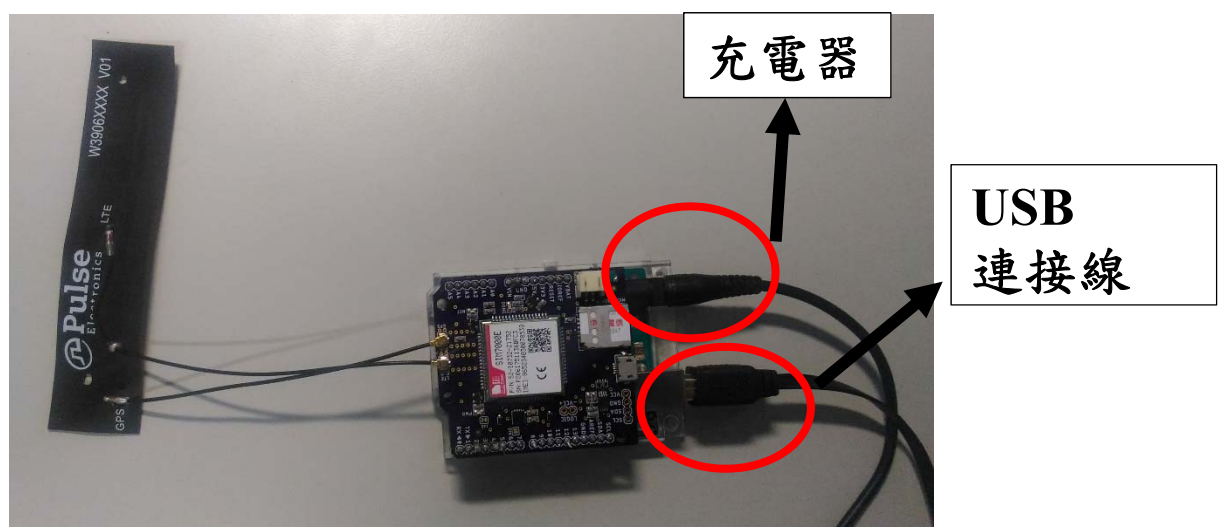


圖3-24 模組連接充電器與USB連接孔

(二)SIM7000E之Arduino擴展板使用方法：

如圖3-24，當SIM7000E之Arduino擴展板順利接上Arduino板之後，就能夠將Arduino板得到的資料傳給SIM7000E並將資料藉由無線通訊傳回研究室並做演算分析。這部分的Arduino板因為先前介紹的Arduino M0 Pro對於程式的解讀性很差，相對於Arduino UNO板解讀性比較理想，所以後續的材料使用都是用Arduino UNO板。而使用方法步驟如下所述：

- 1.當如圖3-24，兩板相結合並且SIM7000E的SIM卡與天線都安裝好、Arduino板的USB接孔也使用線接好，充電器也給予電源，就能夠使用電腦的Arduino IDE介面去傳送程式碼執行資料傳輸。
- 2.如圖3-25，如果SIM7000E有溝通到Arduino板以及得到供應電源的話，就會在PWR與DONE顯示綠燈代表可以正常運作，之後NET顯示燈會固定頻率閃藍燈，表示SIM7000E可正常傳輸資料。
- 3.結合的模組開始讀取輸入的程式碼，並照著設定好的網路協定(這邊使用UDP)去運作AT command的指令並且將資料回傳。AT command的指令寫法跟前面的TCP協定差不多，只要最後設定協定改成UDP即可。
- 4.架設UDP連線：將使用Node.js工具來架設UDP的服務端，本次試驗使用UDP，雖然前面提到該連線方式是比較不穩定的，但是試驗過後發現資料傳送也是適合的，此外UDP的優點如表3-9，有比較大量與快速的傳輸速度。Node.js是一套藉由Java程式專門用來做通訊協定的軟體，並與Linux基金會有合作關係。因為先前的TCP&UDP Debugger的UDP的設定一直有問題，故使用Node.js做通訊協議的設定。
- 5.AT command的指令運作途中就會連接到我們對應到Node.js設定的服務端，並將資料順利回傳。如圖3-26可以看到Node.js介面設置的服務端須設定好裝好電腦服務端的IP以及序列埠，最後會顯示該電腦IP與序列埠編號的UDP正在等候接收，顯示UDP Server listening on IP:序列埠，以本次試驗來說，IP為140.112.76.187，序列埠為33333。而圖3-27是Arduino IDE介面，也要將傳輸給服務端的序列埠以及IP設定一致即

可順利傳輸資料。

6. 資料回傳：將Arduino IDE程式碼輸入模組後，Arduino監控埠就會開始跑AT command指令將資料回傳(如圖3-28)，最後也確實順利成功的在Node.js介面得到資訊(如3-29)。

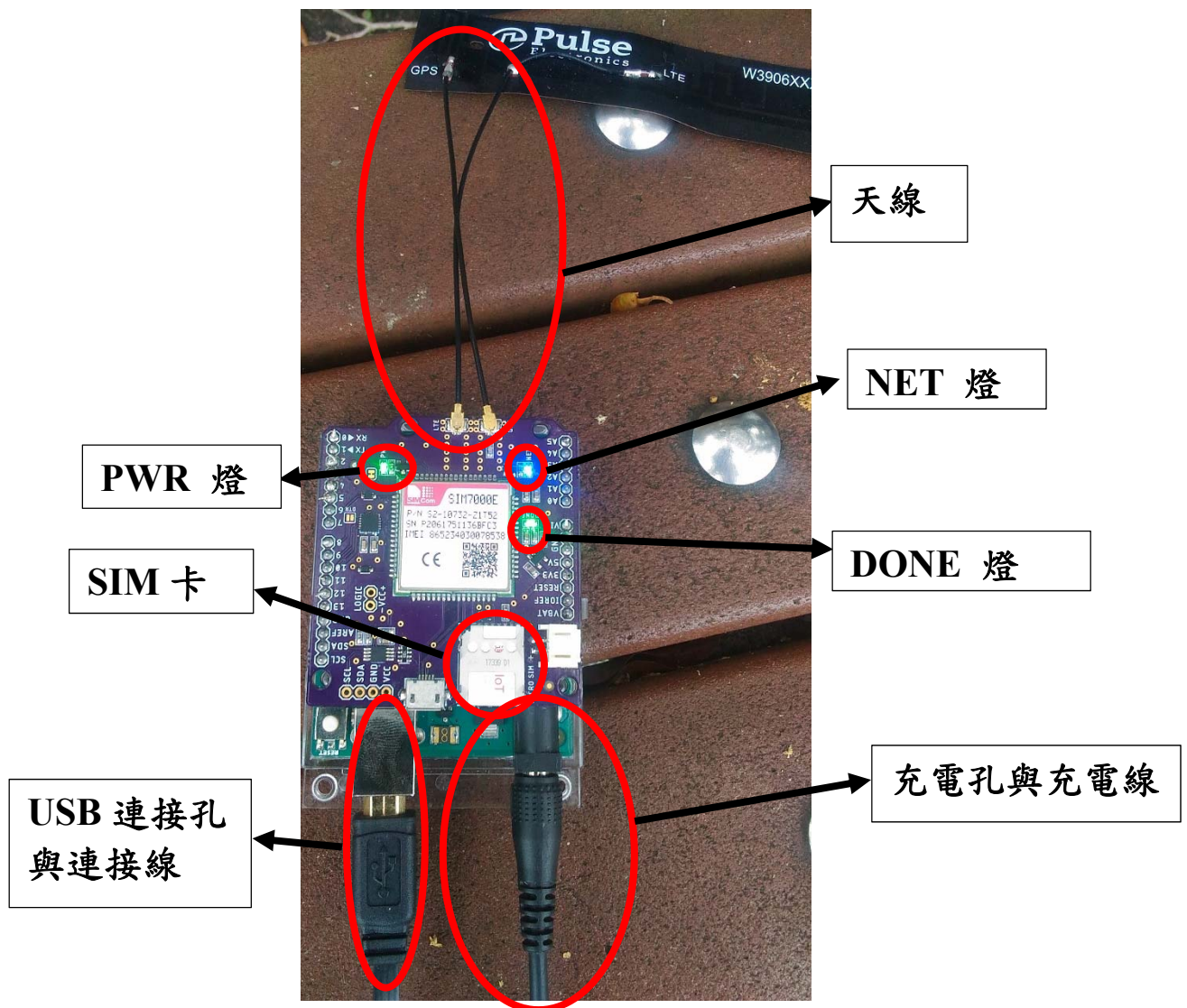


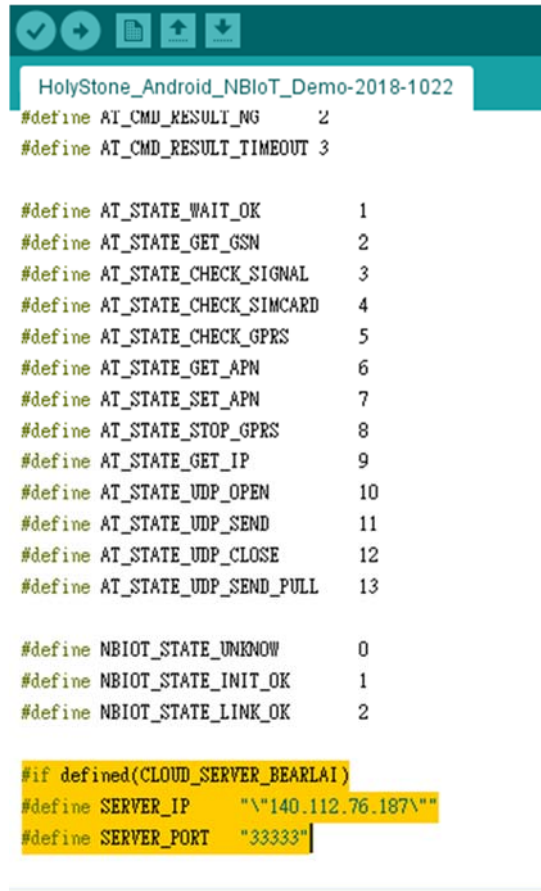
圖3-25 SIM7000E模組啟動的狀態

```

Node.js
  _eventsCount: 1,
  _maxListeners: undefined,
  members: [] },
  _events: { listening: [Function] },
  _eventsCount: 1,
  _maxListeners: undefined,
  _handle: UDP { lookup: [Function: bound lookup4], owner: [Circular] },
  _receiving: false,
  _bindState: 0,
  type: 'udp4',
  fd: null,
  _reuseAddr: undefined,
  [Symbol(options symbol)]: {},
  [Symbol(asyncId)]: 187 }
>
> server.on('message', function (message, remote) {
...   console.log(remote.address + ':' + remote.port + ' - ' + message);
... });
Socket {
  domain:
    Domain {
      domain: null,
      _events: { error: [Function: debugDomainError] },
      _eventsCount: 1,
      _maxListeners: undefined,
      members: [] },
  _events: { listening: [Function], message: [Function] },
  _eventsCount: 2,
  _maxListeners: undefined,
  _handle: UDP { lookup: [Function: bound lookup4], owner: [Circular] },
  _receiving: false,
  _bindState: 0,
  type: 'udp4',
  fd: null,
  _reuseAddr: undefined,
  [Symbol(options symbol)]: {},
  [Symbol(asyncId)]: 187 }
>
> server.bind(PORT, HOST);
Socket {
  domain:
    -
      domain: null,
      _events: { error: [Function: debugDomainError] },
      _eventsCount: 1,
      _maxListeners: undefined,
      members: [] },
  _events: { listening: [Function], message: [Function] },
  _eventsCount: 2,
  _maxListeners: undefined,
  _handle: UDP { lookup: [Function: bound lookup4], owner: [Circular] },
  _receiving: false,
  _bindState: 1,
  type: 'udp4',
  fd: null,
  _reuseAddr: undefined,
  [Symbol(options symbol)]: {},
  [Symbol(asyncId)]: 187 }
> UDP Server listening on 140.112.76.187:33333

```

圖3-26 Node.js設置好服務端時的結果



```

HolyStone_Android_NBIoT_Demo-2018-1022
#define AT_CMD_RESULT_OK      1
#define AT_CMD_RESULT_ERROR   2
#define AT_CMD_RESULT_TIMEOUT 3

#define AT_STATE_WAIT_OK      1
#define AT_STATE_GET_GSN      2
#define AT_STATE_CHECK_SIGNAL 3
#define AT_STATE_CHECK_SIMCARD 4
#define AT_STATE_CHECK_GPRS   5
#define AT_STATE_GET_APN      6
#define AT_STATE_SET_APN      7
#define AT_STATE_STOP_GPRS    8
#define AT_STATE_GET_IP       9
#define AT_STATE_UDP_OPEN    10
#define AT_STATE_UDP_SEND    11
#define AT_STATE_UDP_CLOSE   12
#define AT_STATE_UDP_SEND_PULL 13

#define NBIOT_STATE_UNKNOW    0
#define NBIOT_STATE_INIT_OK   1
#define NBIOT_STATE_LINK_OK   2

#if defined(CLOUD_SERVER_BEARLAI)
#define SERVER_IP      "\"140.112.76.187\""
#define SERVER_PORT    "33333"

```

圖3-27 Arduino IDE介面設定與服務端的IP與序列埠相同



```

COM11 (Arduino/Genuino Uno)

OK
AT+CGATT?
+CGATT: 1

OK
AT+CGNAPN
+CGNAPN: 1,"nbiot"Got APN: "nbiot"

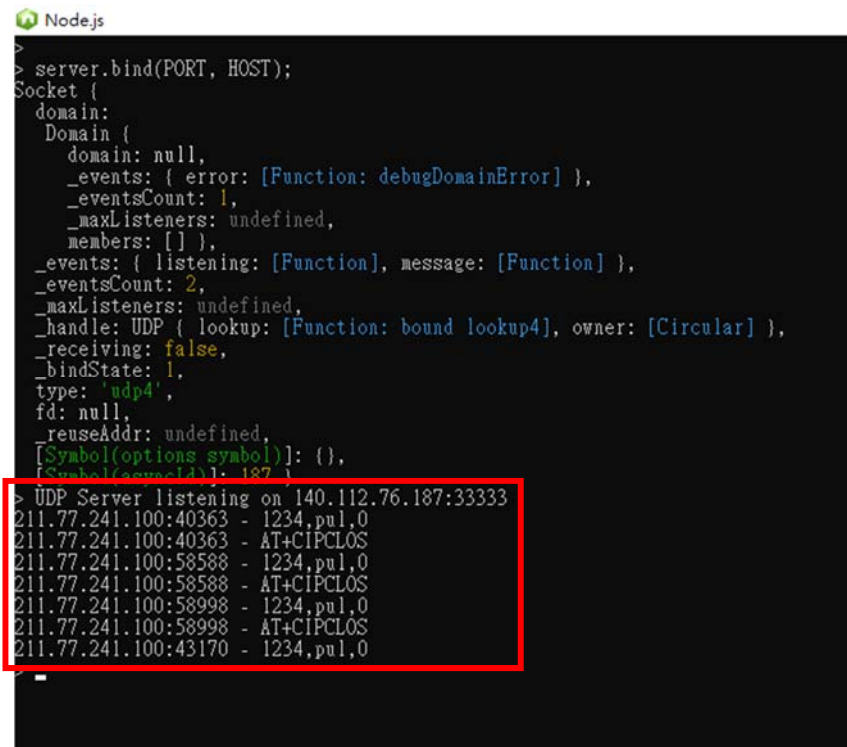
OK
AT+CIPSHUT
SHUT OK
AT+CSIT="nbiot"
OK
AT+CIICR
OK
AT+CIFSR
10.24.1.14Got IP: 10.24.1.14

AT+CIPSTART="UDP", "140.112.76.187", 33333
OK

CONNECT OK
AT+CIPSEND=10
> 1234,pul,0
SEND OK

```

圖3-28 Arduino程式碼傳送到模組上的運作情況



```

Node.js
> server.bind(PORT, HOST);
Socket {
  domain:
    Domain {
      domain: null,
      _events: { error: [Function: debugDomainError] },
      _eventsCount: 1,
      _maxListeners: undefined,
      members: [] },
      _events: { listening: [Function], message: [Function] },
      _eventsCount: 2,
      _maxListeners: undefined,
      _handle: UDP { lookup: [Function: bound lookup4], owner: [Circular] },
      _receiving: false,
      _bindState: 1,
      type: 'udp4',
      fd: null,
      reuseAddr: undefined,
      [Symbol(options symbol)]: {} },
  [Symbol(asyncId)]: 187 }
> UDP Server listening on 140.112.76.187:33333
211.77.241.100:40363 - 1234,pul,0
211.77.241.100:40363 - AT+CIPCL0S
211.77.241.100:58588 - 1234,pul,0
211.77.241.100:58588 - AT+CIPCL0S
211.77.241.100:58998 - 1234,pul,0
211.77.241.100:58998 - AT+CIPCL0S
211.77.241.100:43170 - 1234,pul,0

```

圖3-29 Node.js接收到資料

三、ZigBee硬體：XBee

本計畫第三種使用的 IoT 硬體為 XBee，是屬於 ZigBee 通訊協定當中的其中一項具備資料採集功能的無線通訊模組，是 Digi International 公司製造。XBee 產品有不同的型號，各自擁有不同的天線類型和功能，比較常見的是 S1 與 S2 模組。圖 3-30 為研究階段所使用的 S2 模組外觀，圖 3-31 則是 S2 模組的腳位示意圖，表 3-12 為 S2 模組各腳位的功用，有可以收發 UART 控制器的資料、控制脈衝寬度調變(PWM)的訊號調整與輸出、類比訊號與數位訊號連接的腳位等等。如表 3-13 是 XBee 的 S2 規格表，可從此表了解到，此模組的針腳間距為 2.0 mm，與一般所使用的杜邦線，光只是接頭就已經為 2.54 mm，因此無法直接接線使用。若使用焊槍來焊接線路也是非常不便，因此 Arduino 有研發一款專門使用於 Arduino UNO 開發板上可以連接 XBee 系列的擴展板如圖 3-32，可以使使用者方便連接使用，圖 3-32 除了 Arduino UNO 開發板上可以連接 XBee 以外，也與 MMA8451Q 加速度計結合，這部分的結合與應用將在後面詳細說明。



圖 3-30 XBee S2 模組外觀



圖 3-31 XBee S2 模組腳位示意圖

資料來源：(Digi International Inc., 2009)

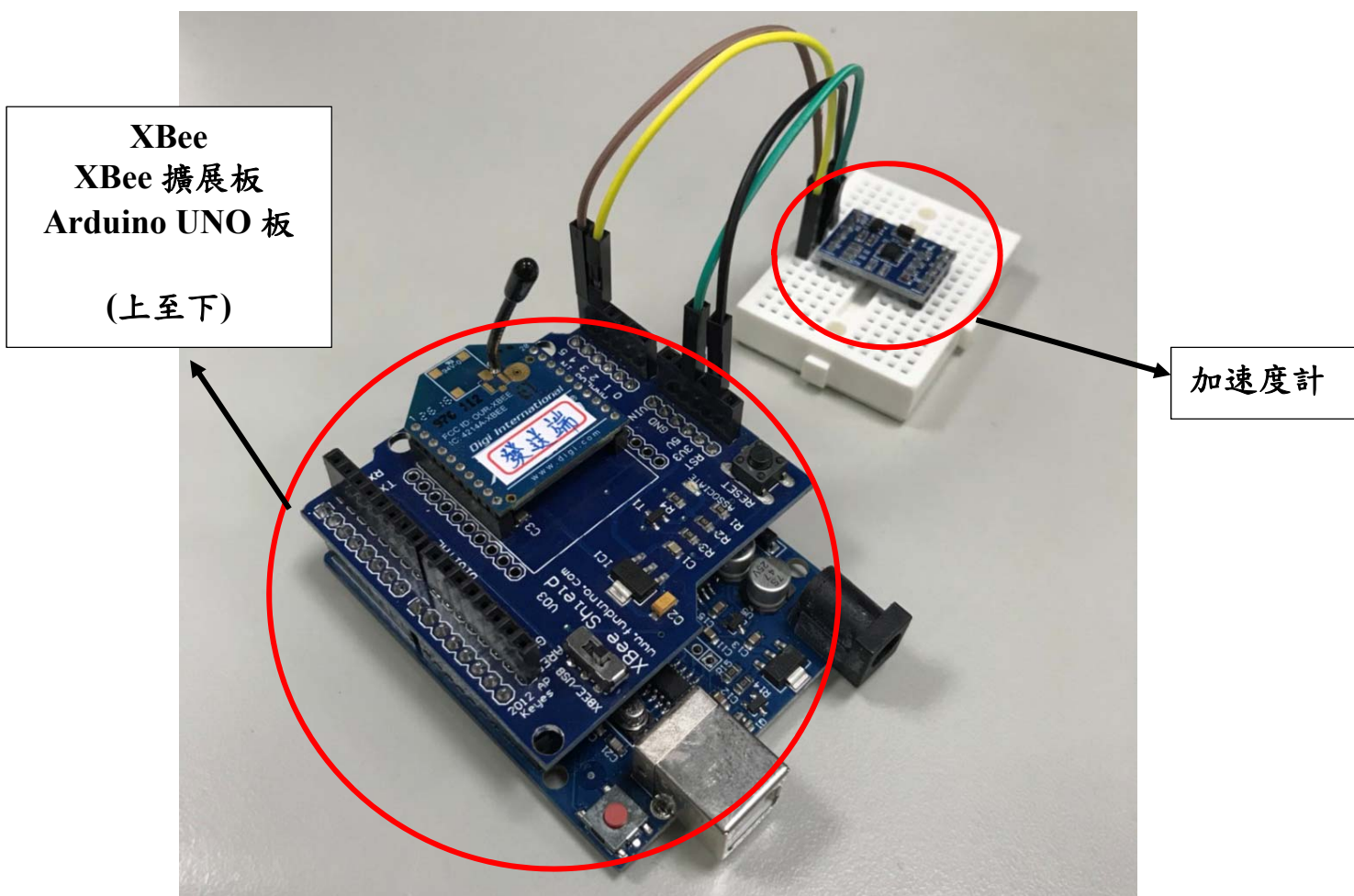


圖 3-32 Arduino UNO 與 XBee 連接

表 3-12 XBee S2 模組腳位介紹表

Pin #	Name	Direction	Description
1	VCC	-	Power supply
2	DOUT	Output	UART Data Out
3	DIN / CONFIG	Input	UART Data In
4	DO8*	Output	Digital Output 8
5	RESET	Input	Module Reset (reset pulse must be at least 200 ns)
6	PWM0 / RSSI	Output	PWM Output 0 / RX Signal Strength Indicator
7	PWM1	Output	PWM Output 1
8	[reserved]	-	Do not connect
9	DTR / SLEEP_RQ / DI8	Input	Pin Sleep Control Line or Digital Input 8
10	GND	-	Ground
11	AD4 / DIO4	Either	Analog Input 4 or Digital I/O 4
12	CTS / DIO7	Either	Clear-to-Send Flow Control or Digital I/O 7
13	ON / SLEEP	Output	Module Status Indicator
14	VREF	Input	Voltage Reference for A/D Inputs
15	Associate / AD5 / DIO5	Either	Associated Indicator, Analog Input 5 or Digital I/O 5
16	RTS / AD6 / DIO6	Either	Request-to-Send Flow Control, Analog Input 6 or Digital I/O 6
17	AD3 / DIO3	Either	Analog Input 3 or Digital I/O 3
18	AD2 / DIO2	Either	Analog Input 2 or Digital I/O 2
19	AD1 / DIO1	Either	Analog Input 1 or Digital I/O 1
20	AD0 / DIO0	Either	Analog Input 0 or Digital I/O 0

資料來源：(Digi International Inc., 2009)

表 3-13 XBee 無線通訊模組的規格表

工作電壓	2.8-3.4V
20 個針腳	含 9 個數位 I/O 及 5 個類比輸入腳(與數位腳共用)，針腳間距 2.0 mm
內建鞭形天線	1 個
室內傳輸距離	30 m
室外傳輸距離	100 m
無線工作頻率	2.4 GHz
無線傳輸資料速率	250 Kps

資料來源：(Digi International Inc., 2009)

第四節 NB-IoT、Arduino板跟加速度計之整合

本階段的工作重點為利用IoT模組(SIM7000E、XBee)與所開發的智慧感測器(加速度計)、微型處理器(Arduino板)共同建構本計畫之物聯網，如圖3-33的模組是本計畫最主要的三項部分，分別為感測器、微處理器、NB-IoT模組在技術上做結合可以達到我們預期的效果。效果即是建構完成後可於現場量測加速度，並將資料傳送至NB-IoT模組並利用無線通訊協定的技術，將加速度資料傳至服務端並經計算輸出位移。對於尺寸規模巨大的水土保持構造物而言，資料的傳輸是另一個挑戰。基於此點，本計畫擬以目前最新的窄頻物聯網(NB-IoT)的SIM7000E以及ZigBee的XBee進行資料傳輸，不過，本計畫目前的規劃，並不在建構一個物聯「網」，而是在於以IoT所提供的開發模組整合。本團隊於先前完成的智慧型感應器(加速度計+Arduino板)，確認感測器所讀取的資料可以傳至IoT模組並利用IoT模組回傳至伺服器。整合的會分成三部分，第一部分為SIMCOM-EVB與Arduino M0 Pro板與MMA7660FC加速度計結合，第二部分為SIM7000E之Arduino擴展板與Arduino UNO板與ADXL345，第三部分為XBee與Arduino UNO板與MMA8451Q加速度計結合。

一、SIM7000E與Arduino M0 Pro與MMA7660FC加速度計結合

針對Arduino板與加速度計的結合，已在前章有提述，只要有將加速度計安裝在Arduino板上合理的腳位，並在Arduino IDE寫好程式，就能夠順利的輸出加速度值。而Arduino板與SIM7000E的連結，並要從Arduino板得到的加速度計資料傳送至SIM7000E，最後再由SIM7000E藉由TCP通訊協定來將資料回傳至主機。那這邊主要會特別強調SIM7000E如何與Arduino板作結合並將資料可以傳輸與接收。通常兩模組要互傳資料時，幾個腳位連接是必要的，兩板子的連接方式與腳位名稱在圖3-34有示意圖，而各腳位連接的詳細內容如下：

(一)Rx-Tx連接

兩塊板子中的一塊板子的Tx與Rx分別與另一塊板子的Rx與Tx連接，Rx(Receive)代表收資料的腳位，Tx(Transmit)代表傳資料的腳位，而在Tx與Rx互接的情況下，傳資料腳位在傳資料到另一塊板子的Rx時，另一塊板子就會

收到資料。

(二)電壓連接

兩塊板子需要在相同伏特數的腳位互相連接，以達成電路連接的通路，而SIM7000E的擴展板SIMCOM板有伏特腳位，分別有3.3V、2.8V、1.8V，Arduino M0 Pro板上面的伏特腳位有5V、3.3V，因此將兩塊板子連接起來的電壓腳位就是兩邊都有的3.3V。

(三)GND相互連接

兩板子的接地腳位(GND)相互連接，接地的目的通常是讓電流往低阻抗通道移動，GND通常電壓較低，所以電流在電路裡跑的時候，會傾向往GND處低電壓的地方移動，形成迴路。

除了硬體上的連接，我們也將會在Arduino介面上寫兩板子可以互相傳訊、彼此溝通的程式碼，期待在兩塊板子溝通成功後，我們可以順利將加速度資料從Arduino板傳至SIM7000E板，目前程式碼部分我們持續努力將兩塊板子做連接並傳訊，這部分將在期末報告作出呈現，而硬體的部分如圖3-33的連接方式，曾有試驗將Arduino板給予電力，SIM7000E也能得到該電力使代表Power的紅燈發亮，代表有接收到電力，圖3-35所示，我將充電器給予Arduino板充電，Arduino板會顯示黃燈表示有電源供應並可以運作，而SIM7000E的SIMCOM板的“POWER”也亮起紅燈，表示兩板子之間通電成功。

期中報告的時候試驗沒成功，而到後來因為有SIM7000E之Arduino擴展板的關係，SIM7000E可完全接在Arduino板上，直接溝通，故期中報告過後就直接使用了SIM7000E之Arduino擴展板，此外，因為Arduino的 M0 Pro板雖然容量大，但對於部分的程式碼無法順利編譯與解讀，故後期也就使用Arduino UNO板。

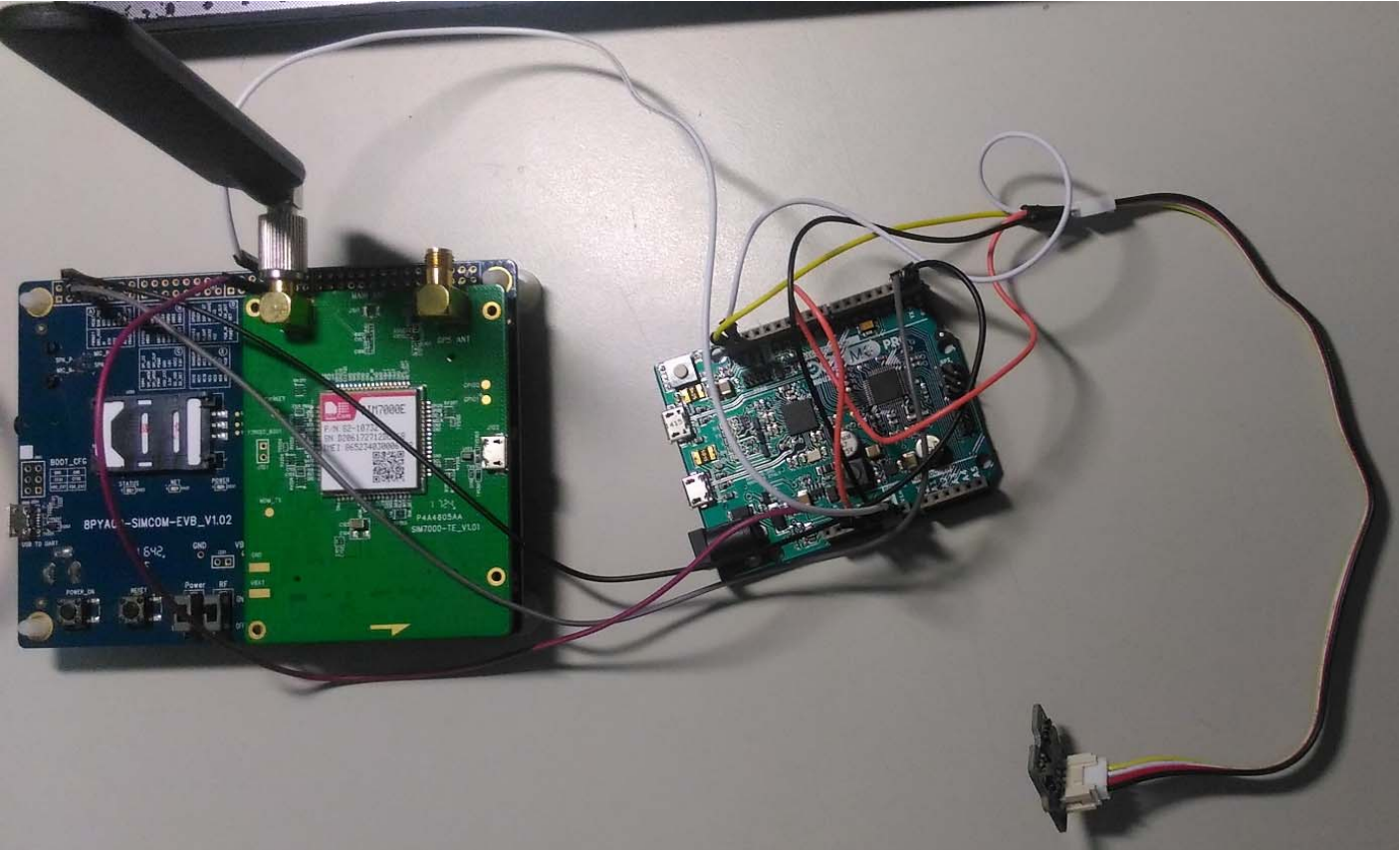


圖3-33 加速度計與Arduino與SIM7000E的結合

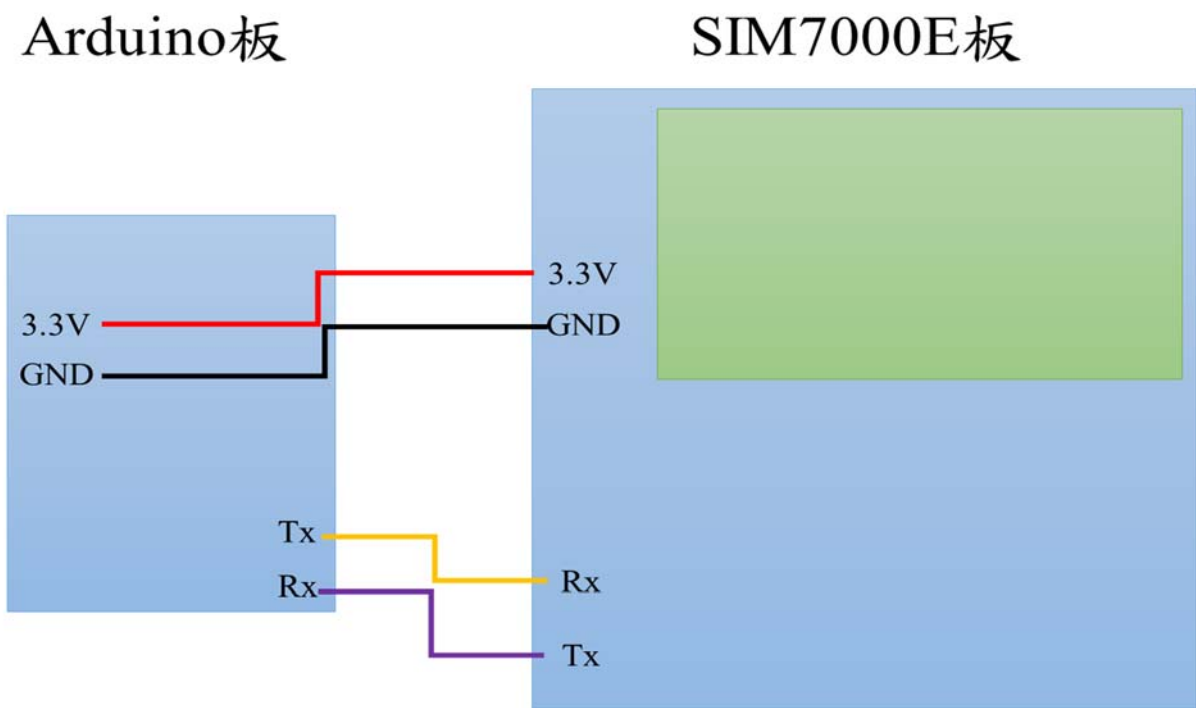


圖3-34 Arduino板與SIM7000E連接示意圖

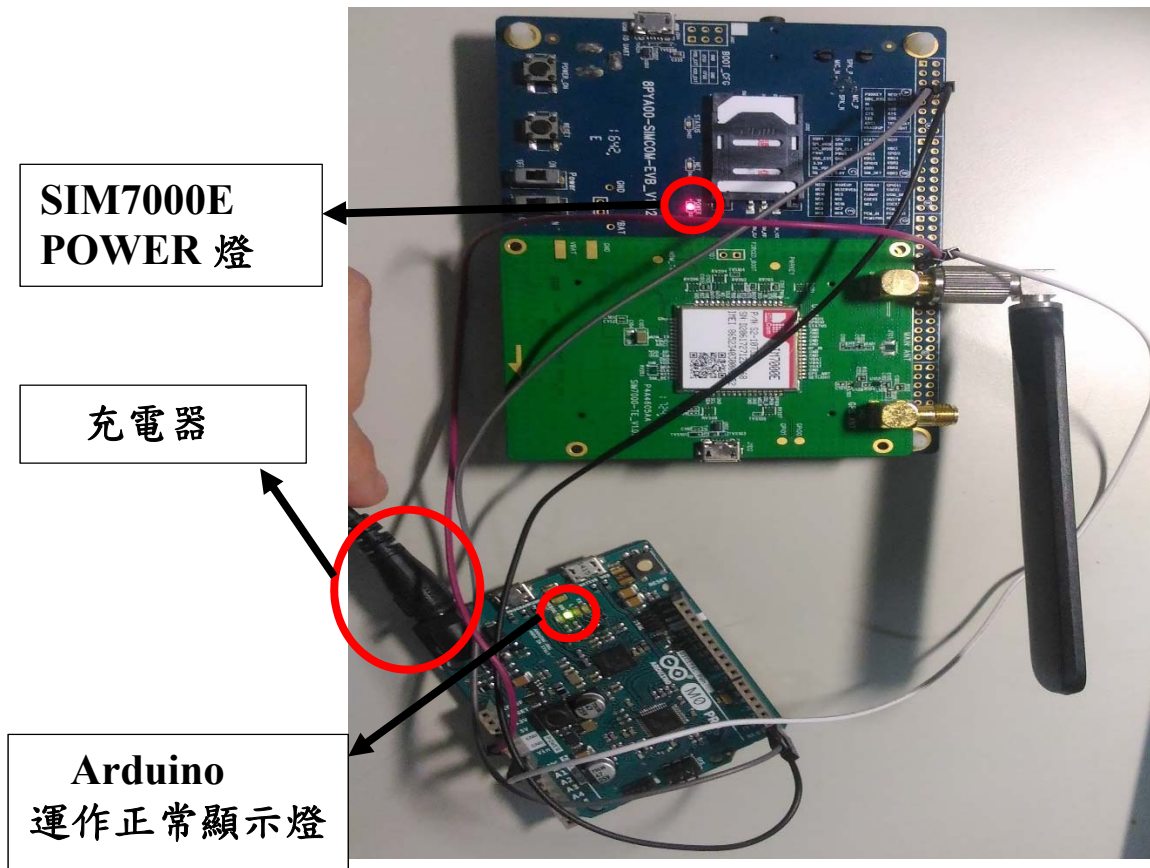


圖3-35 Arduino板與SIM7000E連通證明

二、SIM7000E之Arduino擴展板與Arduino UNO板與ADXL335加速度計

前面有提到SIM7000E之Arduino擴展板與Arduino UNO板直接結合並可傳送訊息的情況，所以這部分只要模組再加上加速度計並且改一下code即可順利讀取加速度計並且傳輸至服務端。如圖3-36的SIM7000E與Arduino板結合，可使用圖3-36圈起來類比腳位去接ADXL335加速度計，可順利產生加速度值並回傳到電腦服務端上。

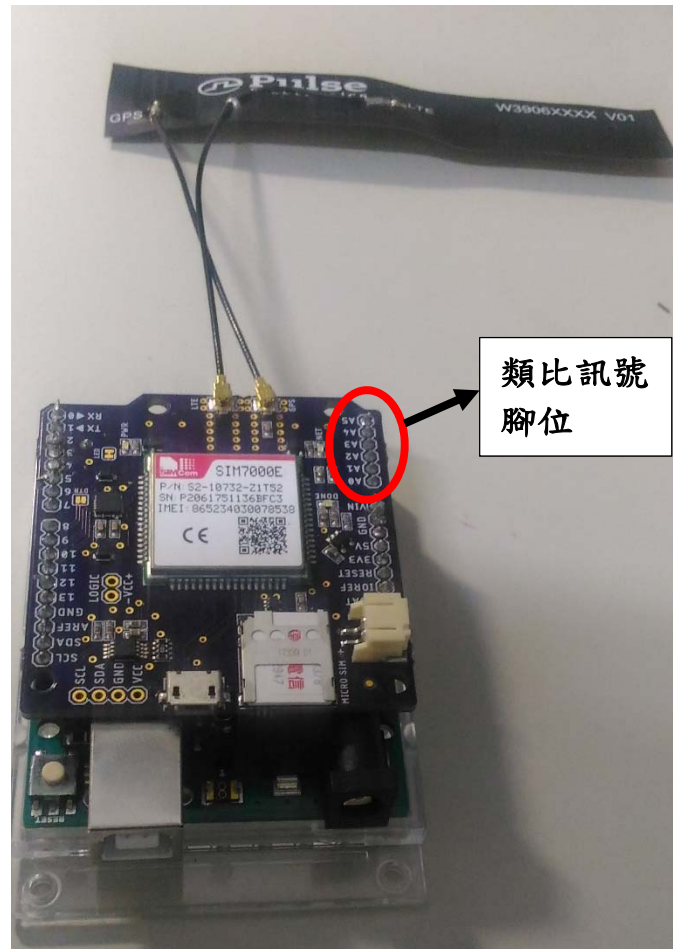


圖3-36 SIM7000E之類比訊號腳位

三、XBee與Arduino UNO與MMA8451Q加速度計結合

圖3-32為三者模組結合的模樣，示意圖如圖3-37，使用Arduino UNO板，連接MMA8451Q加速度計，透過I²C以穩定電壓3.3V輸出，來測量加速度，並且透過Arduino IDE介面寫程式碼，使SD卡與Aeduino板進行溝通，其中XBee與Arduino UNO板結合時，兩者之間需要有一塊XBee的擴展板，該擴展板經過設計可以跟Arduino板連接，故XBee直接安插在Arduino UNO板各腳位，則完成結合。

而要使其模組做到無線通訊，將設定XBee無線通訊作為終端等待指令，而設定介面將使用由Digi International公司所提供的X-CTU，是一套免費軟體用於XBee設定AT mode或API mode，且每個XBee裝置皆有三種角色可以扮演：

- (一) Coordinator：協調者，基本上每個區網只有一個。
- (二) Router：可以接收和發送資料，也可以當作區網中的中繼站。
- (三) End Device：為簡化版的 Router，也可以接收和發送資料，但無法當作區網中的中繼站，另外功能是有睡眠模式可以達到省電作用，由 Router 控制。

XBee 裝置與電腦連線，透過 X-CTU 軟體的設置，可以免除使用 AT commands 指令來設置參數。新版的 X-CTU 與 Arduino IDE 一樣支援 Windows、Mac OS X、Linux 這三種作業系統，軟體可以直接至 Digi International 的官方網站上即可下載 (<https://www.digi.com/>)，因此這個軟體可以非常容易地於 PC 上面操作使用。

從圖 3-38 至圖 3-42 為操作步驟，說明 XBee 模組連接電腦後，藉由 X-CTU 軟體設置基本的參數以及無線通訊的測試，步驟如下：

- (一)如圖 3-38 左上角，紅色方框處為新增和尋找 XBee 模組，當與電腦接線後，需先按下新增按鈕加入 XBee 裝置。
- (二)圖 3-39 為新增後可以直接幫裝置設定參數。設定參數完成後，點選剛才所新增的裝置，右半區塊可以開始定義模塊為發送端和終端，並且更改各項參數值。其中每個模塊間，通訊名稱以及 PAN 識別碼必須一致，DL（目標低位址）參數設置成對方的 MY（16 位元位址）位址，才能進行通訊。
- (三)圖 3-40 與圖 3-41 為設定完參數後，通電時自動建立連線並且可試打些簡單訊息做測試。在 Console log 處打字，訊息將會以 16 進位碼送出並且接收，如同圖 3-42。

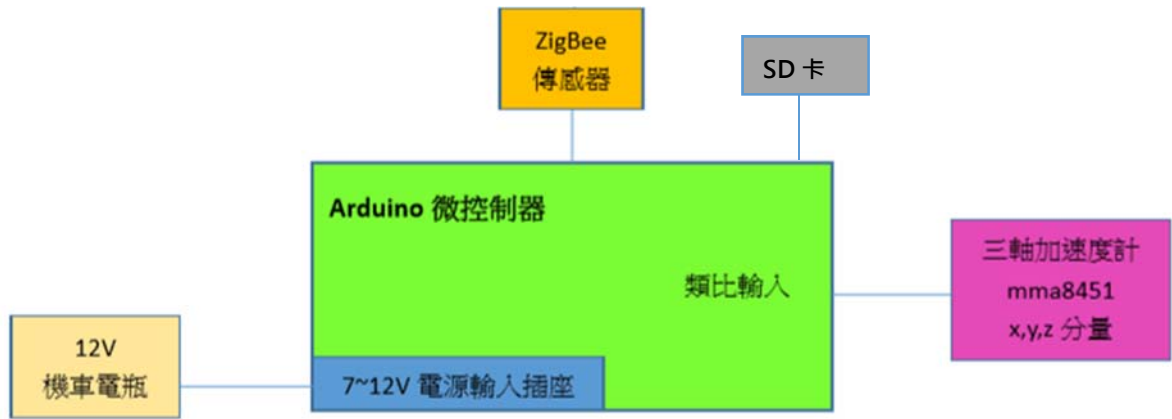


圖3-37 Arduino UNO與MMA8451Q加速度計跟XBee結合示意圖

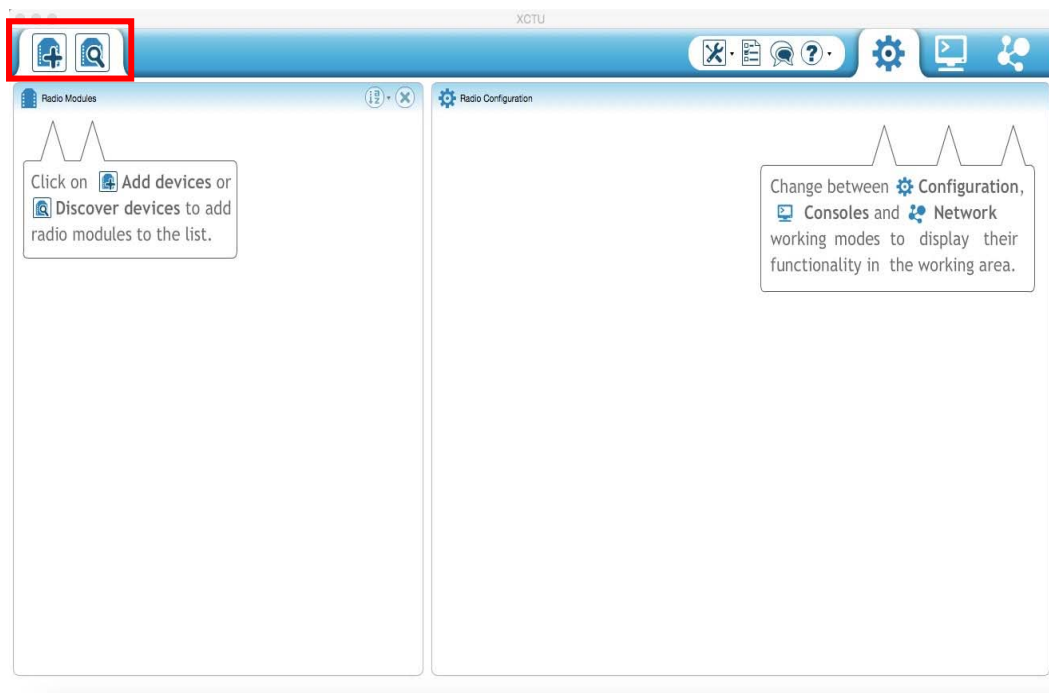


圖 3-38 X-CTU 設定步驟 1

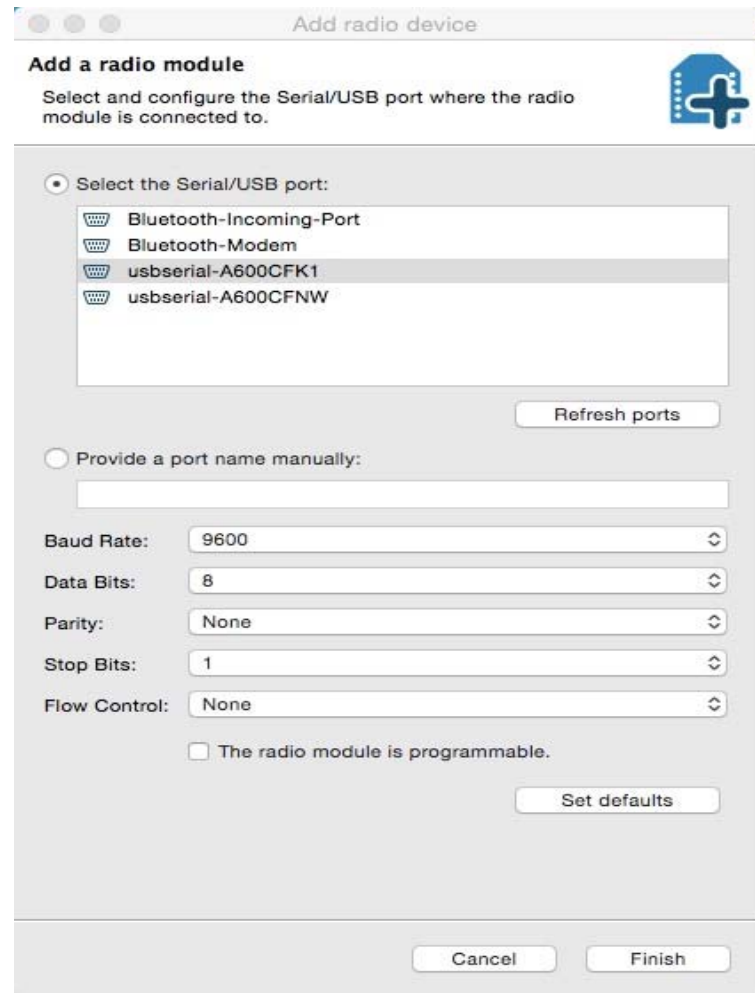


圖 3-39 X-CTU 設定步驟 2

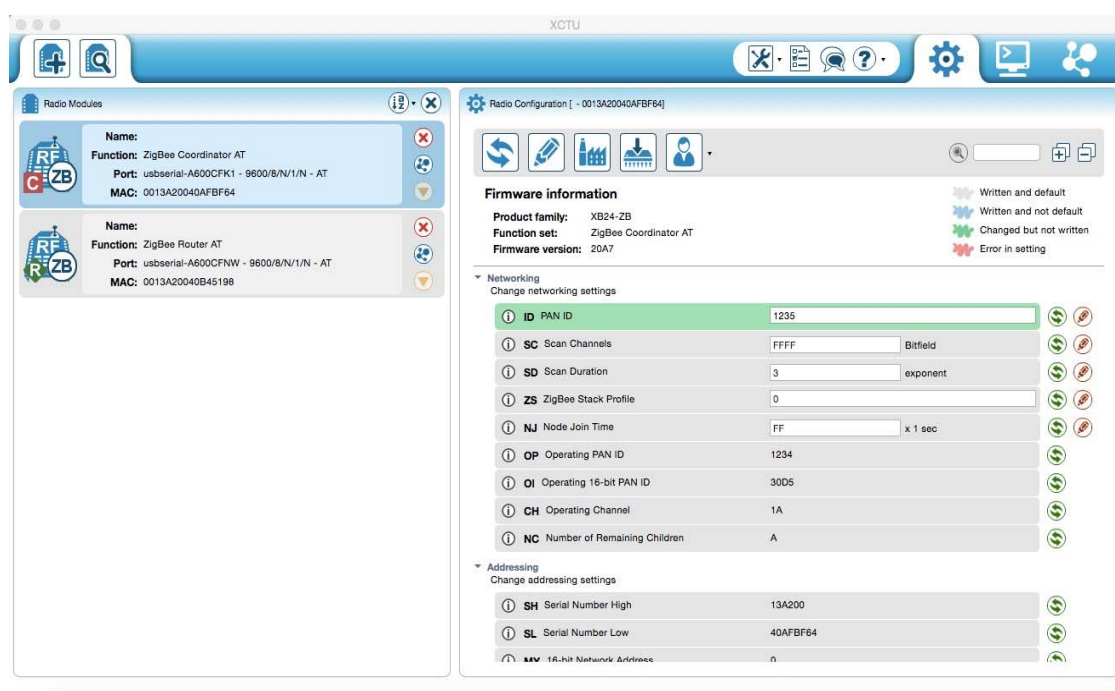


圖 3-40 X-CTU 設定步驟 3

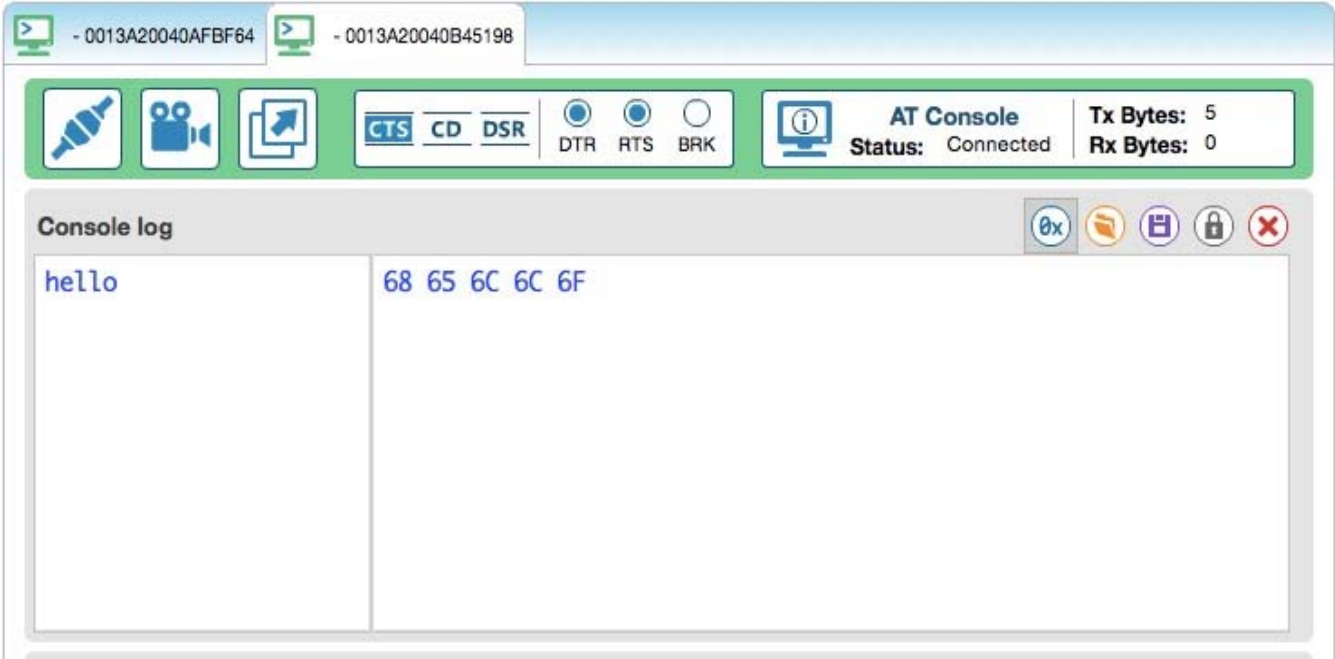


圖 3-41 XBee 發送端測試

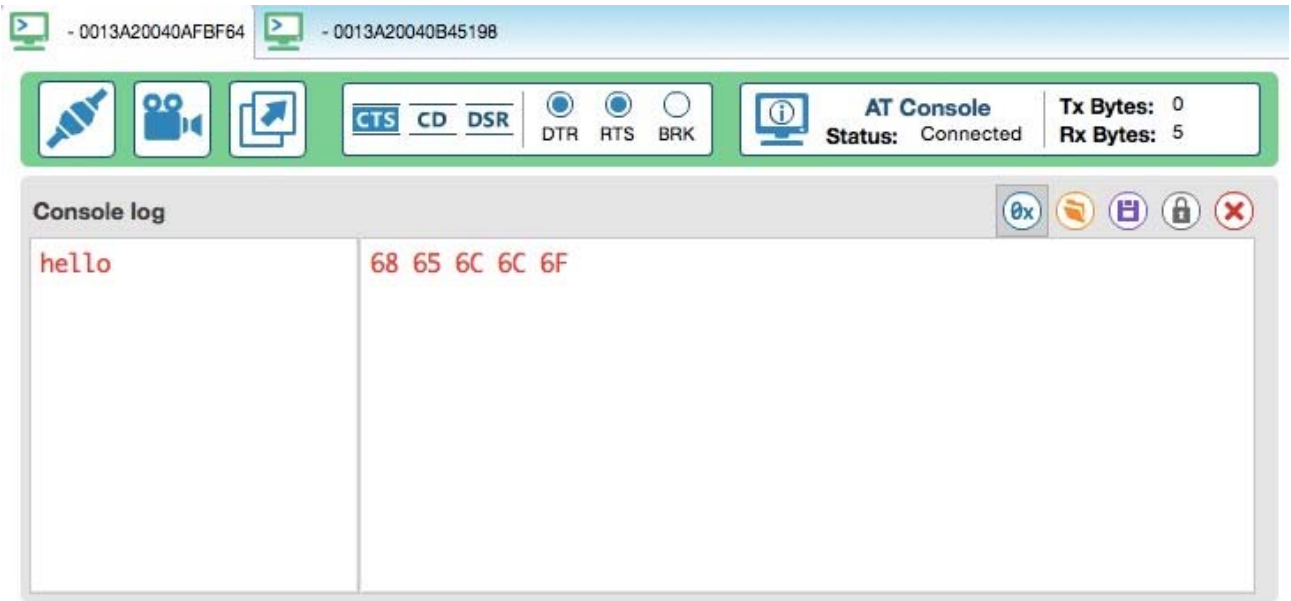


圖 3-42 XBee 終端測試

第四章 免控制點演算法之開發

第一節 文獻回顧

傳統的位移測量方法通常成本昂貴且為勞力密集的作業，運用於實務工程時通常會需要固定的控制點或參考點，造成應用上的瓶頸。相對應於傳統的方法，加速度感應器因易於安裝且成本低廉，利用加速度計量測構造物的加速度，再經由積分運算重建位移是一種實用性高且低成本的方法，前提為須妥善處理積分過程中的低頻飄移。根據所量測的加速度值，目前已有許多數位濾波器可用來重建位移。其中，無限脈衝響應濾波器（infinite impulse response filters, IIR濾波器）和有限脈衝響應濾波器（finite impulse response filters, FIR過濾器）廣泛應用於各種領域。舉例來說，Péceli (1986)研究使用FIR與IIR濾波器來檢測建築結構的穩定度與安全度。但是，上述數位濾波器在以低頻為主的設施結構物(如橋梁)的位移重建過程中可能存有幾個缺點，例如IIR濾波器通常需要位移的初始條件，而這項資訊在現地量測過程中通常並不存在。FIR濾波器亦可能不適用的原因在於低頻範圍時，獲得一個準確的傳遞函數(transfer function)並不容易。

不過從文獻回顧了解到有學者提出一些適合低頻的位移重建方法，例如Vaidyanathan (1986)使用了Passive Cascaded-Lattice Structures的演算法來重建FIR過濾器在低頻時候如何提高其機器的靈敏度，可以更加完整的監測結構的穩定度。後期Lee, Hong, Park (2010)將位移重建視為一個邊界值問題且無須任何的初始條件。於所提的方法中，在給定的時間範圍內，藉由最小化量測加速度與位移兩次微分值，重建所需的位移。Park, Sim, Jung (2013)與Moreu et al. (2015)均利用Lee et al. (2010) 所提的方法進行位移重建，不過，先前研究的量測對象多為實驗室內的縮尺寸的結構物或橋梁，目前尚未有研究顯示此項技術可運用於戶外實尺寸的水保設施，故本計畫將在水保設施上藉由位移值重建來判斷水保設施的安全性。

第二節 演算法之選擇

若進一步探討 Lee et al. (2010) 的方法會發現，位移(u)的兩次微分並不存在於實際的量測值中，因此位移(u)為最佳化中代求的設計變數。例如可以使用中央有限差分(central finite difference)計算「假設的加速度值」並使之與現地量測的加速度差值最小，然而，因為中央有限差分需要前一個時間點與後一個時間點的資訊方能完成計算，因此，上述最佳化問題並非一個定義完整的方程式(ill-posed problem)，不具唯一解。為避免過度擬合(overfitting)既有資料的訊息，可以正則化(regularization)處理。例如使用 Tikhonov 正則化(脊迴歸，ridge regression)。以上的解釋利用式子可寫為：

$$\underset{u}{\text{Min}} \Pi_E(u) = 0.5 \int_{T_1}^{T_2} (a(u(t)) - \bar{a})^2 dt \quad (2)$$

其中：

$\Pi_E(u)$ ：正則化參數

T_2 、 T_1 ：分別為中央有限差分的前一個和後一個時間點

$a(u(t))$ ：假設的加速度值

$\bar{a}$ ：現地量測加速度

可展開成：

$$\Pi_E(u) \approx 0.5(0.5(a_1 - \bar{a}_1)^2 + (a_2 - \bar{a}_2)^2 + \dots + (a_{2k} - \bar{a}_{2k})^2 + 0.5(a_{2k+1} - \bar{a}_{2k+1})^2) \Delta t = 0.5 \|L_a(a - \bar{a})\|_2^2 \Delta t \quad (3)$$

其中：

L_a ：矩陣，階乘為 $1 \times 2k+1$ ，除了頭尾為 $\frac{1}{\sqrt{2}}$ ，其餘皆為 1

而如此的合理化只針對加速度，也需要對位移定義合理化參數，才能使整體合理化參數夠完整。而位移的合理化參數為：

$$\Pi_R = 0.5 \|u - u_{st}\|_2^2 \leq r^2 \quad (4)$$

其中：

u ：真實位移

u_{st} ：重建系統下所計算的假定位移

r ：最佳解邊界(solution bound)

小於 r^2 能確保重建位移下 u 的精確性但通常我們對於 u_{st} 是未知的所以先假定為 0，變成：

$$\Pi_R = 0.5 \|u\|_2^2 \leq r^2 \quad (5)$$

兩項合理化參數相加為：

$$\text{Min}_u \Pi_E(u) = 0.5 \|L_a(a - \bar{a})\|_2^2 \Delta t + 0.5 \lambda^2 \|u\|_2^2 \quad (6)$$

其中兩種不同的合理化參數形式直接相加會產生誤差，所以針對位移合理化參數的部份給定一個 λ^2 使兩者相加合理。最後可得：

$$u = Ca(\Delta t)^2 \quad (7)$$

其中：

C ：以上各項參數的綜合矩陣

λ ：最佳化並且經過 Lee et al. (2010) 試驗並得證為 $46.81 \times N_d^{-1.95}$

N_d ：時窗大小

Lee et al. (2010) 藉由測得加速度推算位移也須注意到時間區間的問題，發現如圖 4-1 的算法，將時間線上區分成好幾塊時窗(time window)，從加速度推算到位移時會發現每格時窗在計算的時候時窗與時窗的邊界點會被兩時窗影響，因為每格時窗各有一換算的位移，所以時窗邊界的加速度換算的位移值會有兩種，是不合理的。為了解決這項問題，決定如圖 4-2 所示，採用時窗覆蓋的方式，在計算加速度轉換至位移時就可以避免未覆蓋時窗的邊界加速度問題，而此方式也是本計畫認為轉換加速度為位移時可採用的方式。而從上述說明，基於先前提出的方式進行位移重建時，有許多參數需要濾定(如有限差分或其他方式的選擇、正則化的方式與正則化的參數)。基於經濟性、實用性與先前研究的結果，本計畫擬導入暨改善此項位移重建技術，在演算法的建立過程中，低頻誤差是否可以避免、是否完全不須初始條件與計算效率與成本將是評估的重點。

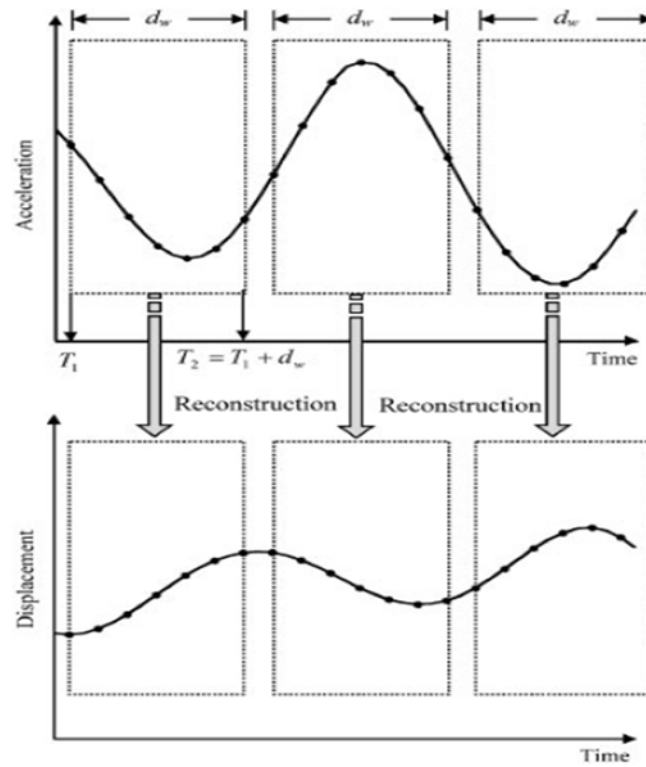


圖4-1 時窗之間未重疊情況下的位移重建過程

資料來源：(Lee et al., 2010)

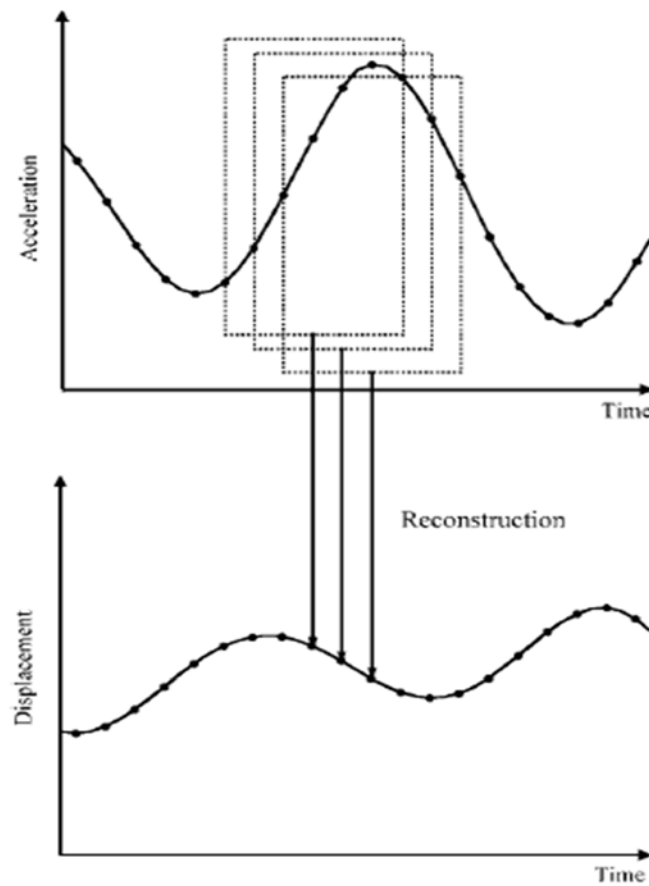


圖4-2 時窗之間重疊情況下的位移重建過程

資料來源：(Lee et al., 2010)

第三節 開發模式參數之最佳化

上述參數的最佳化工作，可利用粒子群演算法(PSO)進行。粒子群演算法是由 Kennedy 與 Eberhart 於 1995 年提出，是一種以群體動力學為基礎的最佳化方法，源自鳥類等群體社會行為，個體的行為不但會受到個體過去經驗以及認知影響，同時也會受到群體社會影響。將個體視為一個粒子或質點，起初為一群隨機的粒子，將每個粒子帶入目標函數中求得適存值，在搜尋空間中每個粒子有各自的速度及方向，經由各個粒子自身過去經驗以及群體行為進行搜尋，每經過一個迭代，各個粒子會比較自身過去迭代的值(pbest)，如果新的位置較佳，將此位置取代為新的 pbest，pbest 會影響粒子下次搜尋方向。而在每次迭代產生新的 pbest 後，粒子間會互相比較，找出群體之中最佳位置(gbest)，再經過一迭代後，如果有新的位置較原 gbest 佳，則新的位置將取代原 gbest 成為新的 gbest，每次迭代更新粒子的位置與速度是基於 pbest 與 gbest 來進行下一次迭代的參考，以最後一次迭代的 gbest 來做為問題之最佳解，圖 4-3 為粒子移動示意圖。因此粒子群演算法須根據下列公式進行迭代：

更新速度：

$$V_{i,j}^{k+1} = w^k \times V_{i,j}^{k+1} + c_1 \times r_1 \times (P_{besti,j}^{k+1} - X_{i,j}^k) + c_2 \times r_2 \times (G_{besti,j}^k - X_{i,j}^k) \quad (8)$$

計算位置：

$$X_{i,j}^{k+1} = X_{i,j}^k + V_{i,j}^{k+1} \quad (9)$$

其中：

i ：粒子數， $i = 1, 2, \dots, m$

j ：粒子的維度，即指設計變數之數目， $j = 1, 2, \dots, n$

k ：迭代次數

$V_{i,j}^k$ ：速度向量

$X_{i,j}^k$ ：位置向量

$P_{besti,j}^k$ ：各個粒子所經歷過最佳位置向量

$G_{besti,j}^k$ ：全體粒子所經歷過最佳位置向量

W^k ：慣性權重

c_1 、 c_2 ：學習因子

r_1 、 r_2 ：為 $[0,1]$ 之間的隨機均勻亂數

粒子群演算法步驟如 4-4 圖，可以看到如同前面所提的，每個數值互相做迭代，符合到我們最佳的目標函數 λ 。

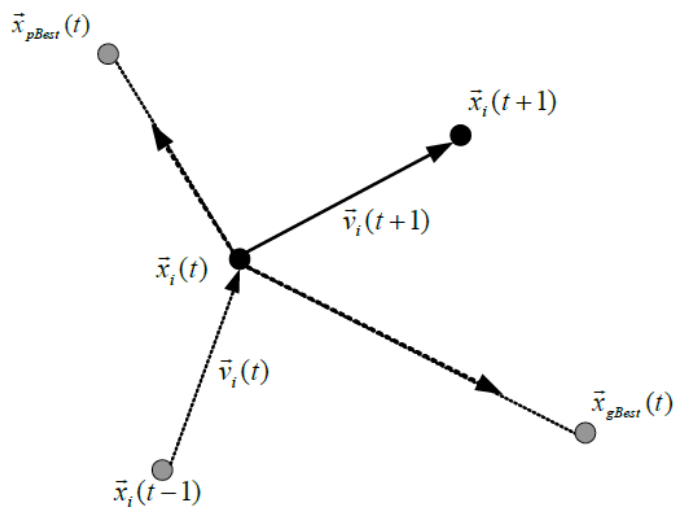


圖4-3 粒子移動示意圖

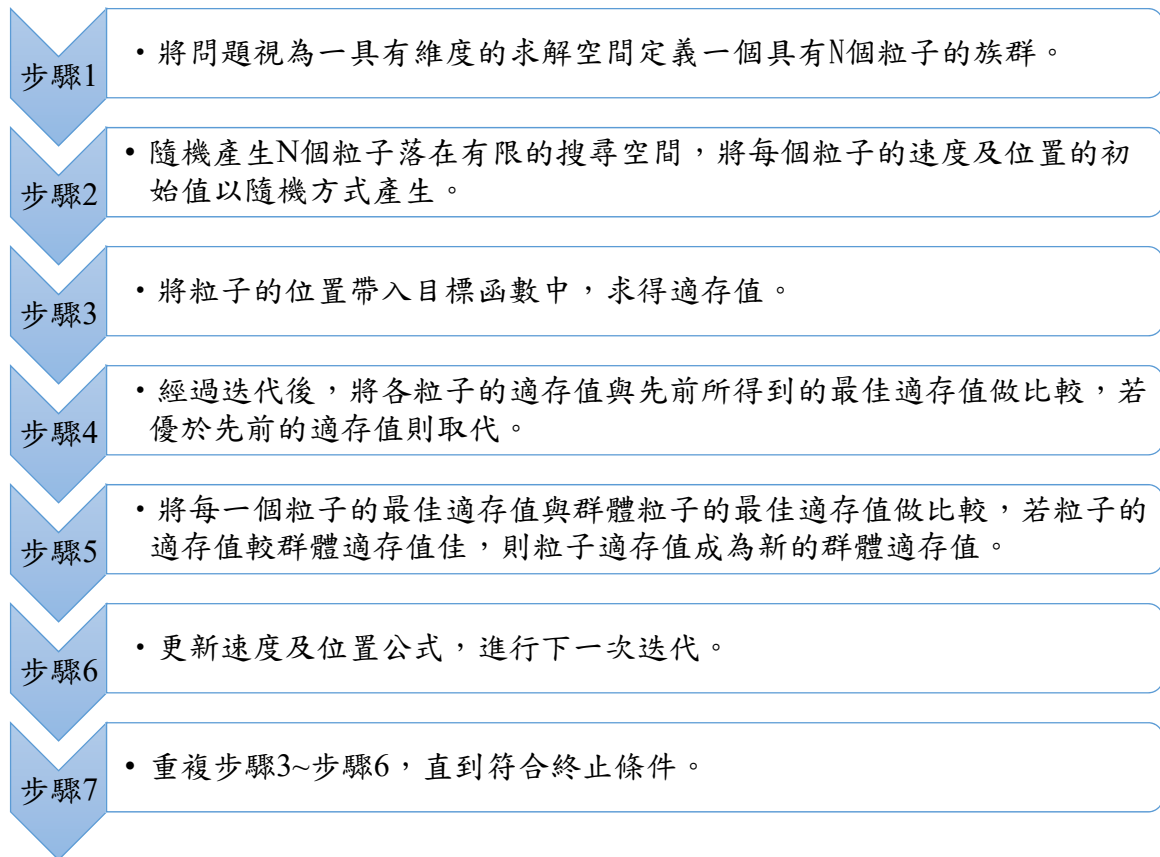


圖 4-4 粒子群演算法流程圖

第五章 重建位移之驗證與試驗結果

本計畫的目標在於建立必要的技術以取得加速度資料、傳送到室內主機伺服器，與藉由所開發的免控制點演算法重建位移。為了驗證上述目標確實達成，設計了四個部分的測試。第一部份利用SAP2000軟體建立一簡單的結構物，如懸臂梁，施於歷時震動紀錄並進行動力分析，取得自由端之加速度與位移。隨後將加速度利用各種加速度轉換成位移的演算法重建位移，並比較重建位移與歷時位移反應之差異。第二部份利用本研究團隊所設計的單軸振動臺以及所製作的懸臂梁結構物進行試驗，試驗過程簡述如下。首先，將加速度計設置於振動臺上的懸臂梁上、輸入一個簡諧運動歷時，最後，進行加速度的量測。由於懸臂梁的剛性相對而言較大，因此，懸臂梁於試驗期間並無明顯變形，符合一般水保設施的特性。取得加速度值後的分析分為兩個部份，第一個部分為快速傅立葉轉換(FFT)、第二個部份為位移重建分析。因為輸入已知頻率的簡諧運動，因此，可藉由FFT所得頻率值驗證所取得加速度值的正確性。同SAP2000數值分析的部分，重建後的位移將與簡諧運動的位移進行比較，驗證重建位移演算法的準確性。第三部份將使用粒子群演算法得到的最佳化 λ 值計算與分析演算法的重建位移與真實位移是否接近。第四部份則為室外試驗，目的在於檢驗NB-IoT模組的無線通訊協定效果。

第一節 數值分析

首先的第一項驗證方式為數值分析，數值分析要驗證所選擇的演算法是否適合執行本計畫。使用SAP2000來模擬結構物的受力狀態以及所受的位移量與速度量。SAP2000是由美國Computer and Structures Inc. (CSI)公司開發研制的通用結構分析與設計軟件。SAP2000已有近五十年的發展歷史，是美國乃至全球公認的結構分析計算程序，在世界範圍內廣泛應用。蘇德馨(2009)的專題研究精簡報告中提到SAP2000可以提供各式各樣結構物之模型建立與分析功能。圖5-1為SAP2000軟件的圖樣。

將利用SAP2000設計結構物來做試驗。以圖5-2的懸臂梁為例，SAP2000可將懸臂梁分割成多個有限元素的組成，並且施予外力，取得第一、第二、第三振態(如圖5-3)，可進一步了解懸臂梁的振動狀態。懸臂在經過歷時分析後可得到位

移值與加速度值，而加速度值將會以以上述章節所提到的 Lee et al. (2010)演算法與梯形法做計算將加速度轉換成位移，驗證所提方法計算結果判斷正確與合理性。

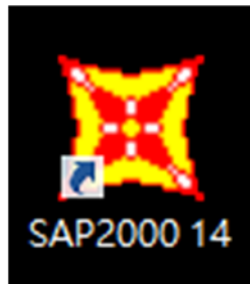


圖 5-1 SAP2000 軟體圖樣

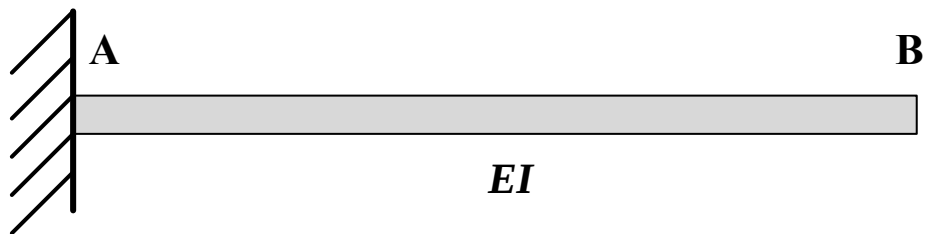


圖 5-2 數值分析模擬之懸臂梁

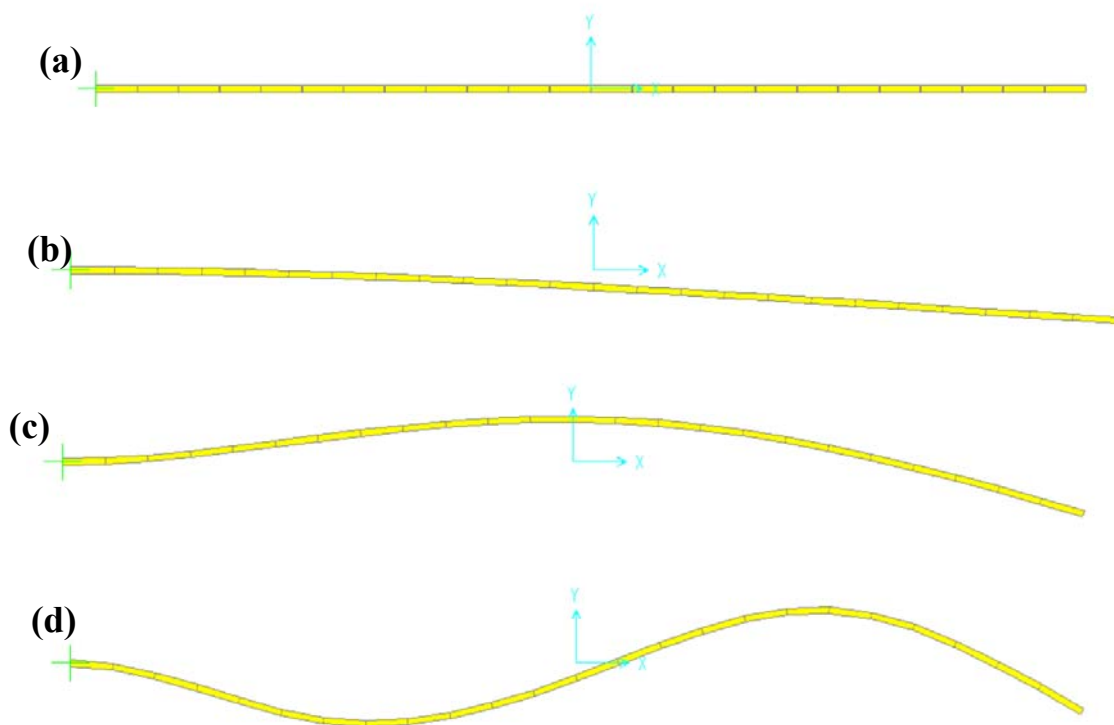


圖 5-3 數值分析模擬之有限元素的懸臂梁強迫振動狀態(a)為懸臂梁的強迫振動前(b)為懸臂梁的第一振態(c)為懸臂梁的第二振態(d)為懸臂梁的第三振態

本次的數值模擬分析試驗將使用兩種不同的歷時紀錄進行試驗，第一種為 25Hz 的振動訊號，且取樣頻率 0.1 秒、第二種為 6.5Hz 的振動訊號，且取樣頻率為 0.002 秒。其中、第一種歷時為真實地震紀錄，推測已進行基線修正；第二種歷時為本計畫團隊自行模擬的地震紀錄，未進行基線修正。將加速度利用演算法轉換成位移之後，比較重建位移與計算的位移是否相似，如果最後兩者呈現的趨勢是一致且位移量是相似的話，未來的室內實驗與室外實驗就能夠使用該演算法來評估設施結構物的位移量。演算法的試驗有四種：

一、梯形積分法(non-overlapped)

一般在做加速度與位移的轉換就是將加速度積分兩次得到位移值的方式，如圖 5-4 的積分方式，將 x 段切分成好幾段，並在每一段求得面積。

二、重建位移法(無時窗)

參考 Lee et al. (2010)的研究發現到在一開始原本基礎的梯形積分法加入時間域的限制，並正則化且加入參數過後而重建出來的位移值。該篇文章有提到當加入時窗的時候重建位移時將時間劃分好幾個區域去計算可以減少時間推移時產生的誤差，不過在此部分將使用無時窗的方式做試驗。

三、重建位移法(有時窗，non-overlapped)

當重建位移法有時窗限制後，理論上可以減少在時間推移時產生的誤差情況，但時窗之間在此部份是要試驗沒有相互覆蓋的情況，如同圖 4-1，雖然時窗與時窗間在沒有相互覆蓋的情況下可能其邊界值在兩時窗之間計算上會有尷尬的情況產生，但仍然是我們可以試驗的一種部分，究竟是否這樣沒有相互覆蓋的時窗在做重建位移後是否誤差增加。

四、重建位移法(有時窗，overlapped)

當重建位移法有時窗限制後，會建議如同圖 4-2 的情況，將時窗之間彼此相互覆蓋，如此這樣的方式就不會產生兩時窗之間的邊界值尷尬的問題。覆蓋方式為該試驗資料點與資料點間的時間距離，例如第一次試驗每 0.1 秒有一筆加速度值，則時窗的覆蓋如同圖 4-2，向右每 0.1 秒時窗推移，留下上一個時窗 0.1 秒的

間隔。但是否這種方式重建位移後，其位移值與真確的位移值是否真的最接近最精確，依然是需經過試驗並且確認的方法。

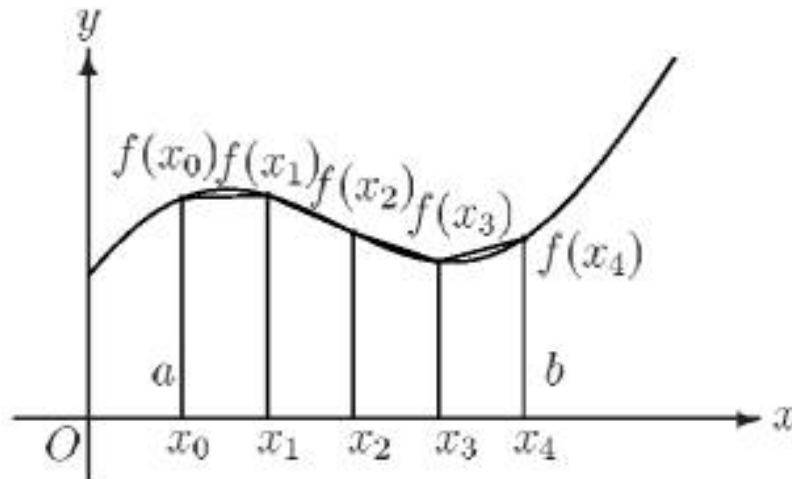


圖 5-4 梯形積分法示意圖

Matlab 為一種高階程式語言，是 MATrix LABoratory 的縮寫。Matlab 通常較多用於演算法開發、資料視覺化、資料分析以及數值計算的進階技術計算語言和互動式開發環境。常見的矩陣運算、繪製函式與資料圖項等常用功能外，Matlab 還可以用來建立使用者介面與呼叫其它的程式語言。

以上的將加速度轉換成位移的方法，二到四將參考 Lee et al. (2010)的文章做位移重建，以上的演算法轉換將在 Matlab 介面呈現，並且可以在附錄八到十一參考相關的 Matlab 程式碼。

兩項實驗的條件與結果如下：

一、25Hz 下每 0.1 秒的值

在 25Hz 的振動頻率並計算每 0.1 秒的加速度與位移，從振動的模擬下得出自由端 B 端的位移與加速度，將位移與加速度的數值資料做成圖，分別為圖 5-5 與圖 5-6，數值資料的部份可參考附錄十二。圖 5-5 是自由端 B 端時間與加速度的關係圖、圖 5-6 是自由端 B 端時間與位移量的關係圖，從圖 5-5 與圖 5-6 可以發現到在十秒的試驗中，越到後面 B 點自由端的加速度會越來越大，位移也會因此在時間越大的情況下，位移量就提高。而加速度值將會以上述章節所提到的演算法轉換成位移，與 SAP2000 位移驗證所提方法計算結果判斷正確與合理性。

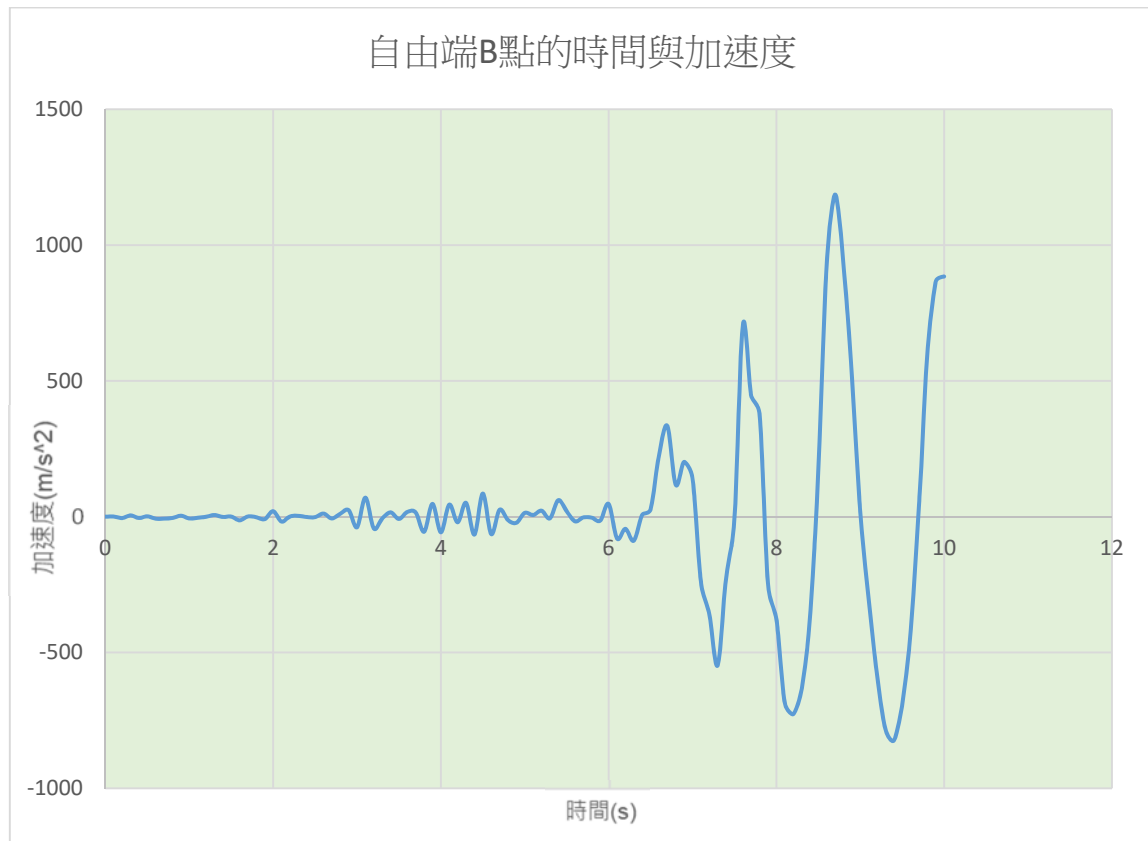


圖 5-5 數值分析模擬之懸臂梁在 25Hz 振動頻率下自由端時間與加速度量關係圖

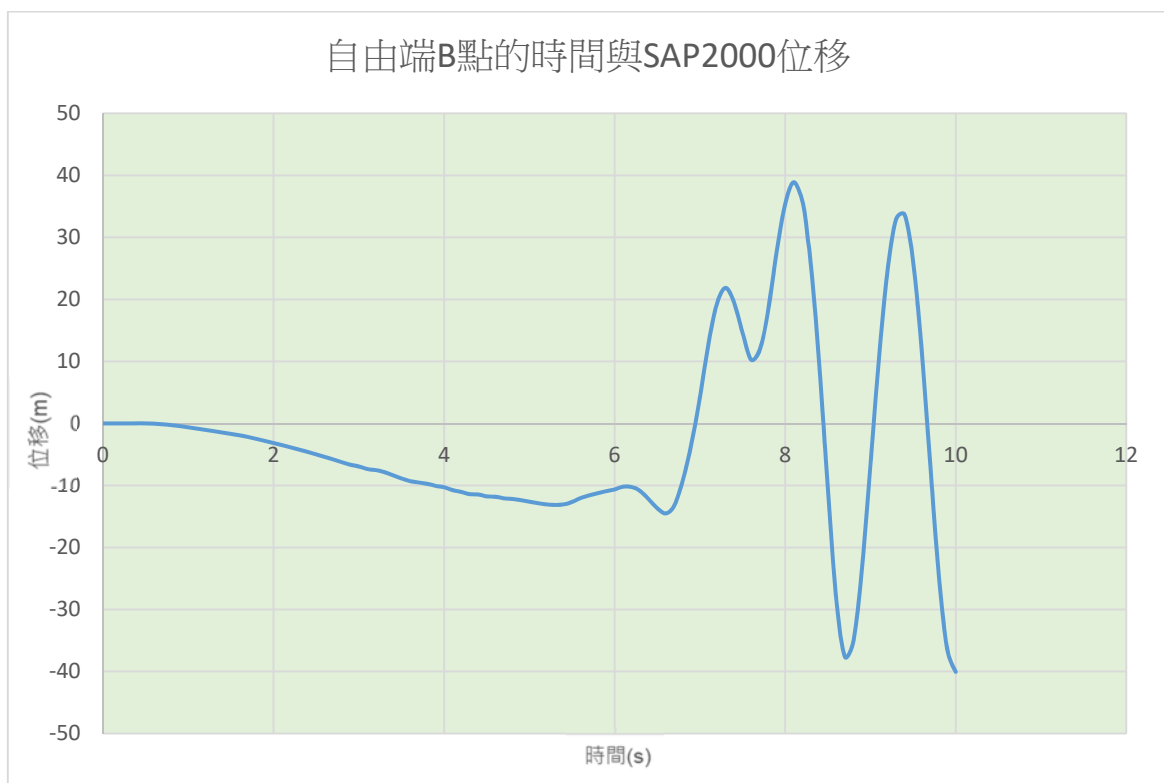


圖5-6 數值分析模擬之懸臂梁在25Hz振動頻率下自由端時間與SAP2000位移量關係圖

目前有加速度與 SAP2000 位移的資料後，我們將可使用四種把加速度轉換成位移的演算法，結果如下：

(一)梯形積分法

在梯形積分法做出來位移結果與 SAP2000 位移相比較，得到的如圖 5-7。由圖可知，梯形法與 SAP2000 所輸出的位移十分接近，不過，這並不意味梯形法為適用的位移重建演算法，之所以會得到相似的結果，也可能是 SAP2000 內建的位移演算法亦為梯形演算法。另、從結果可以判定，此真實的地震紀錄應如預期，已進行基線修正。

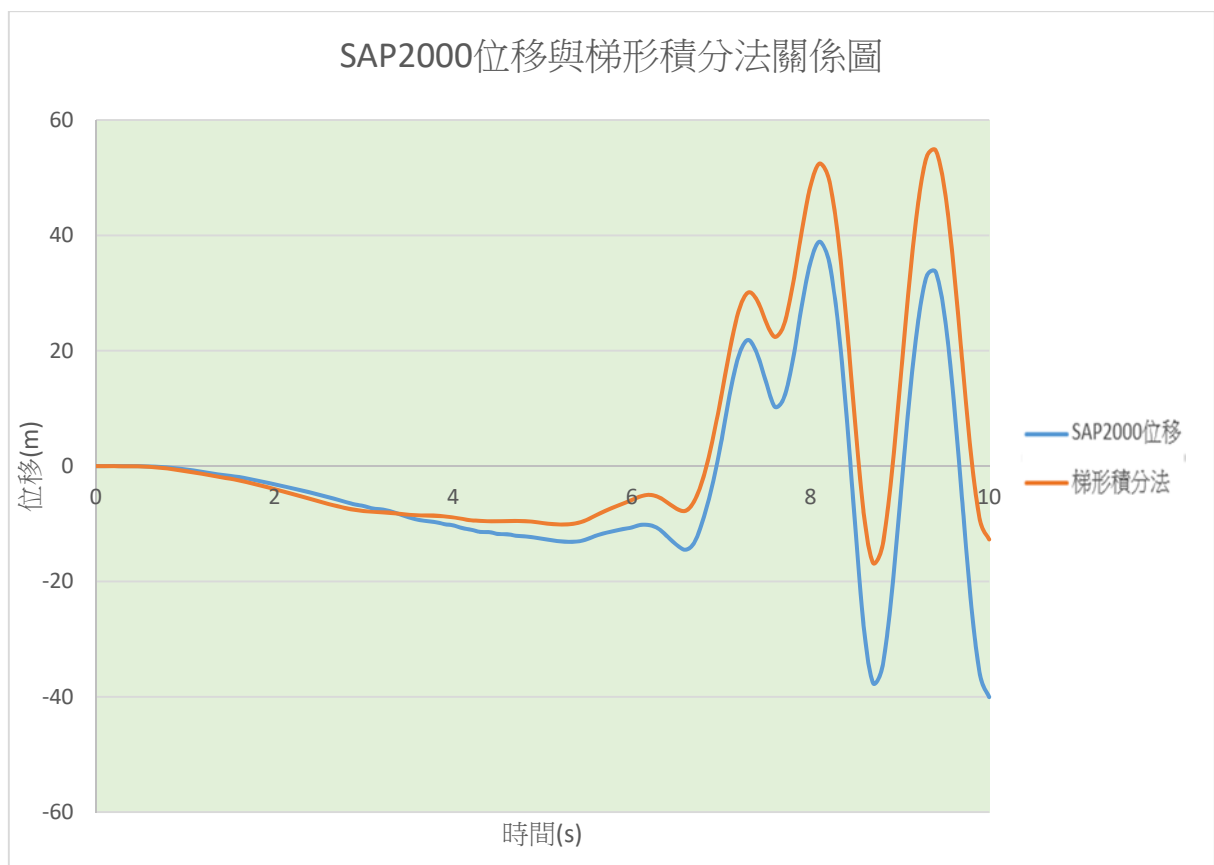


圖 5-7 25Hz 振動頻率下 SAP2000 位移與梯形積分法得到位移之關係圖

(二)重建位移法(無時窗)

圖 5-8 是 SAP2000 位移與重建位移法在無時窗的情況下的關係圖，可以發現到在初期 6 秒兩者的差異性比較大，但到 6 秒以後兩者的趨勢甚至數據相似度都相當的高，無時窗雖然從 Lee et al. (2010)的研究認為會比有時窗的誤差還大，但實測的結果兩者並沒有差太遠。

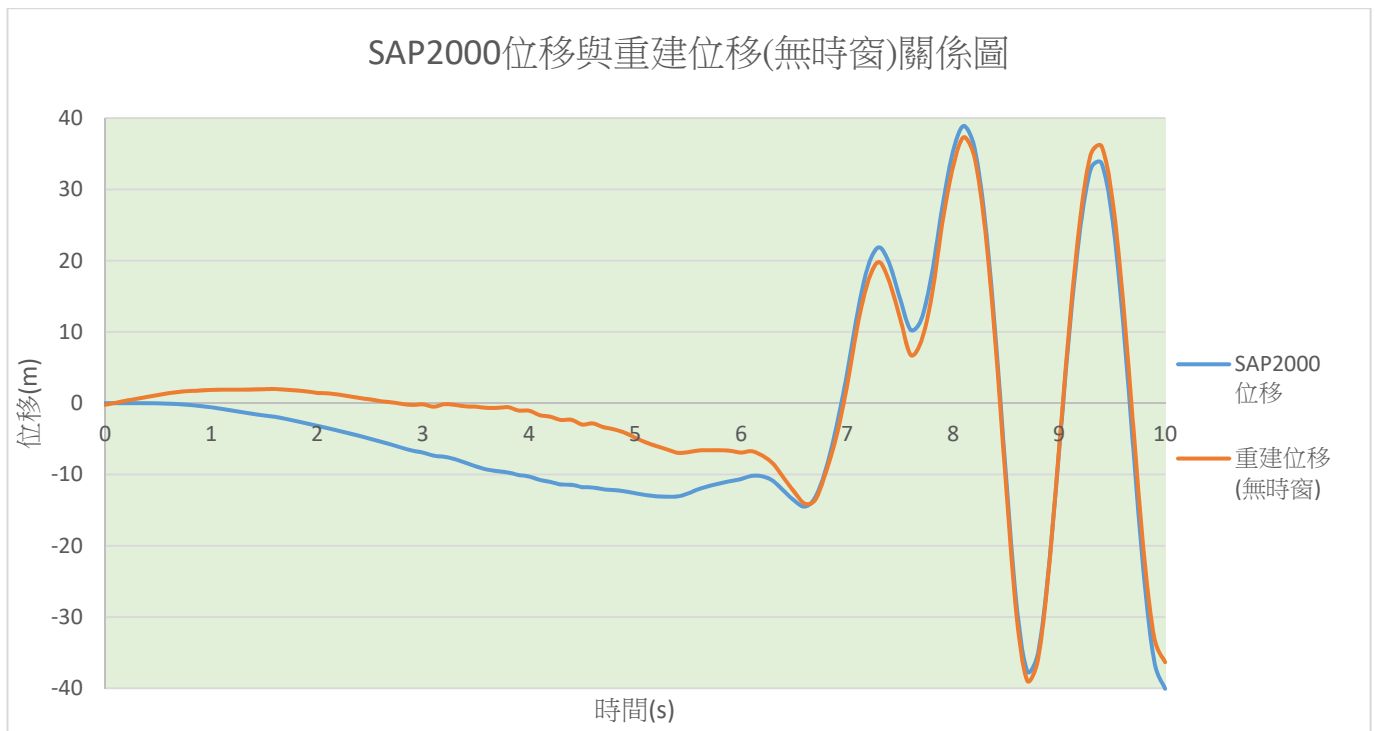


圖5-8 25Hz振動頻率下SAP2000位移與重建位移法(無時窗)得到位移之關係圖

(三)重建位移法(有時窗，non-overlapped)

圖 5-9、圖 5-10 與圖 5-11 皆是 SAP2000 位移與重建位移法(有時窗，non-overlapped)兩資料去做比較圖。從圖 5-9 看起來兩者並沒有太大的關係，不管是趨勢還是數值，兩者的相關性跟相似度都相當不理想。試驗其他尺寸的時窗，像圖 5-10 是使用了時窗有 10 個，時窗大小為 1 秒，雖然重建位移在趨勢上跟 SAP2000 位移差不多，但是數值的相似程度卻相去甚遠；以及圖 5-11 使用了 5 個時窗，時窗大小為 2 秒，趨勢上有配合 SAP2000 位移，且數值相似程度最比 10 個視窗與 20 個視窗還高。Lee et al. (2010)文章提到時窗個數越多、尺寸越大的話，會比較接近 SAP2000 位移，這部份我們的試驗支持此想法發現到認為設置時窗並將時窗擴大再計算，會比較接近 SAP2000 位移的情況發生。

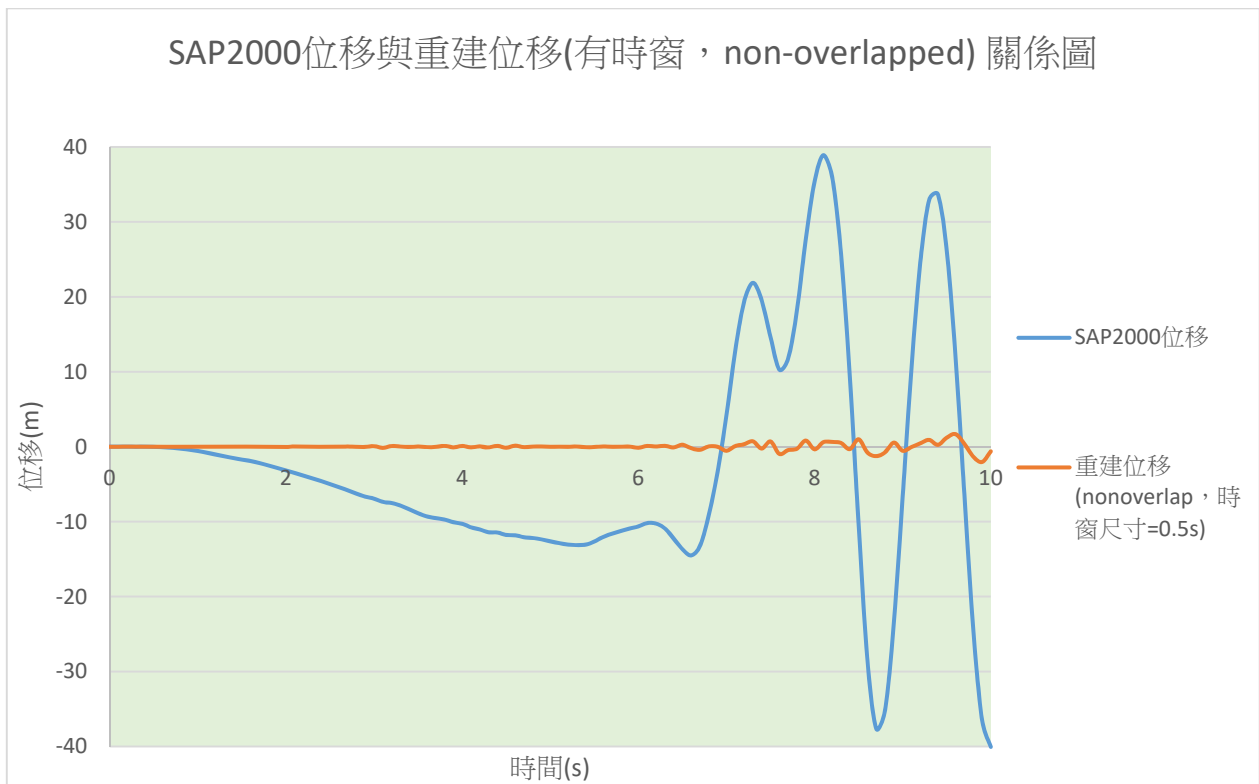


圖 5-9 25Hz 振動頻率下 SAP2000 位移與重建位移法(有時窗，non-overlapped) 得到位移之關係圖(時窗有 20 個，時窗大小為 0.5 秒)

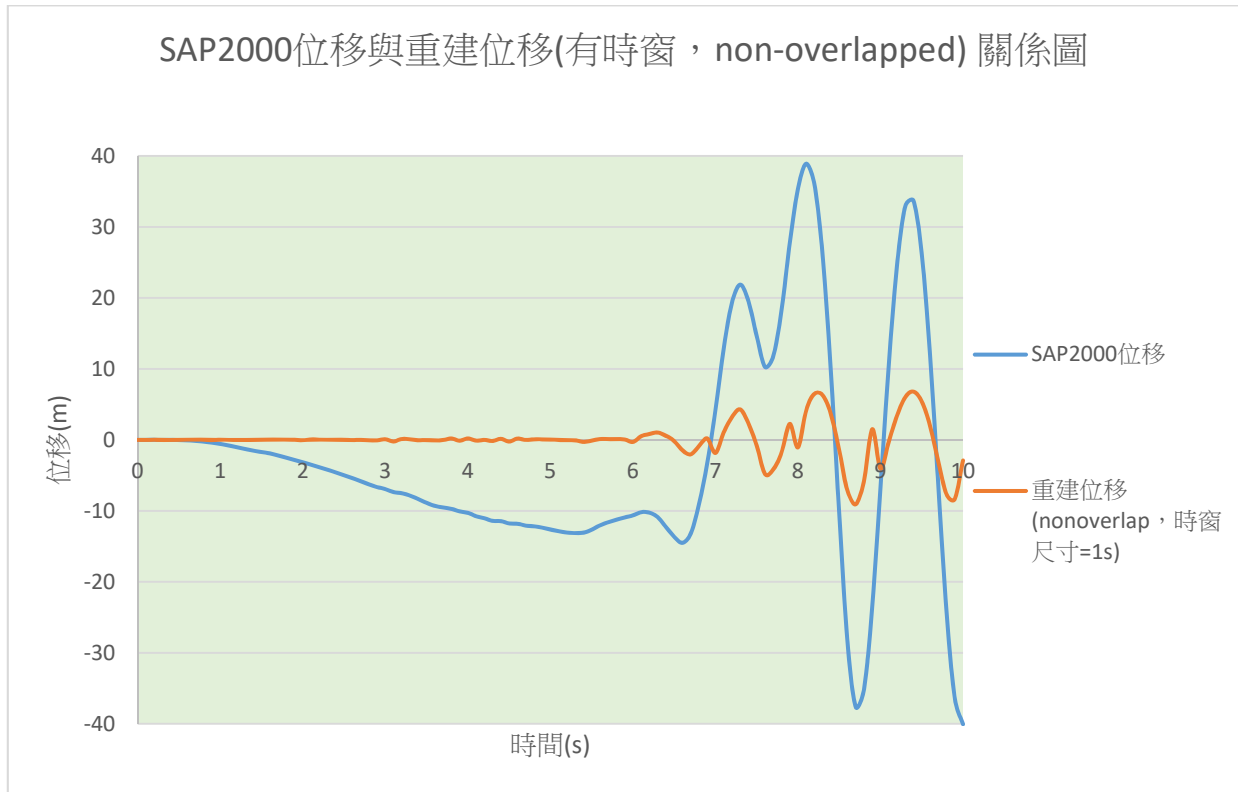


圖 5-10 25Hz 振動頻率下 SAP2000 位移與重建位移法(有時窗，non-overlapped)
得到位移之關係圖(時窗有 10 個，時窗大小為 1 秒)

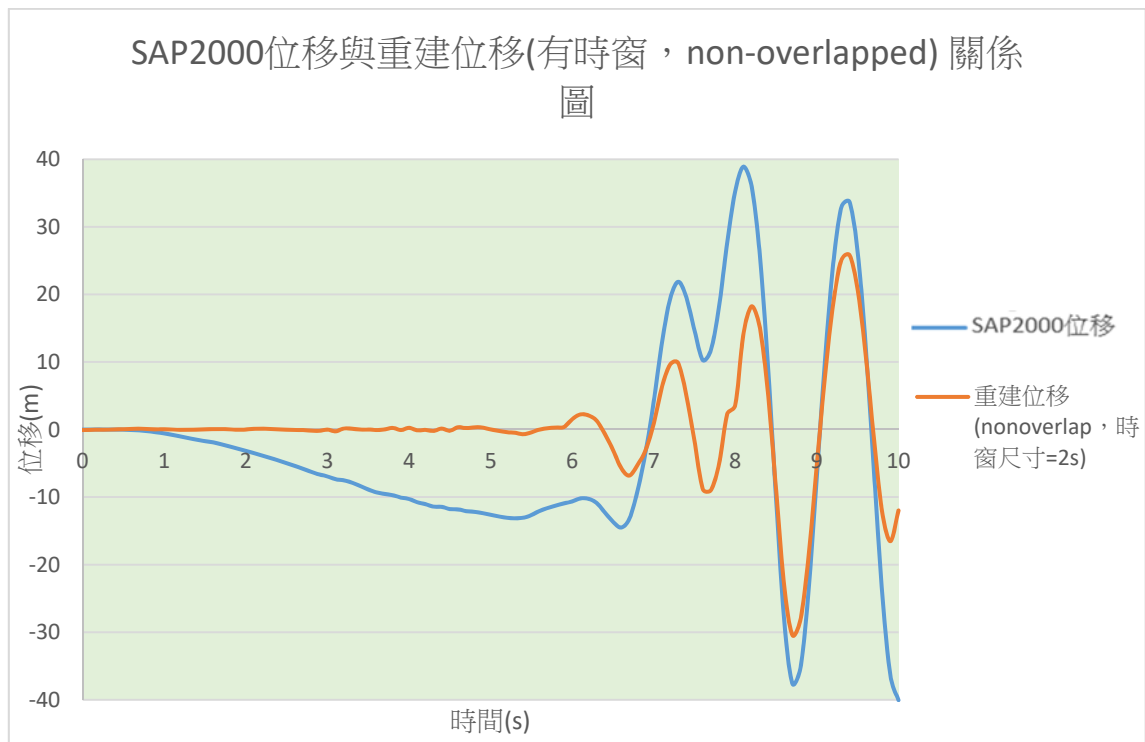


圖 5-11 25Hz 振動頻率下 SAP2000 位移與重建位移法(有時窗，non-overlapped)
得到位移之關係圖(時窗有 5 個，時窗大小 2 秒)

(四)重建位移法(有時窗，overlapped)

圖 5-12、圖 5-13 與圖 5-14 是在不同的時窗尺寸與時窗個數下使用重建位移法與 SAP2000 位移去比較的關係圖，且有時窗並讓時窗彼此是覆蓋的狀態下將加速度轉換成位移。圖 5-12 是時窗尺寸為 0.1 秒去轉換位移，發現重建位移與 SAP2000 位移的趨勢與數值相似度皆相差甚遠；圖 5-13 是時窗尺寸為 2 秒去轉換位移，雖然兩者的趨勢跟數值相似度有好轉，但跟前面的無視窗或者梯形積分法或有時窗且時窗間無覆蓋的情況，落差還是相當大；圖 5-14 是時窗尺寸為 5 秒去轉換位移，趨勢跟數據相似度又比圖 5-13 還差，所以總體來說，有時窗且時窗間有覆蓋的效果跟 SAP2000 位移比，效果不是太好。

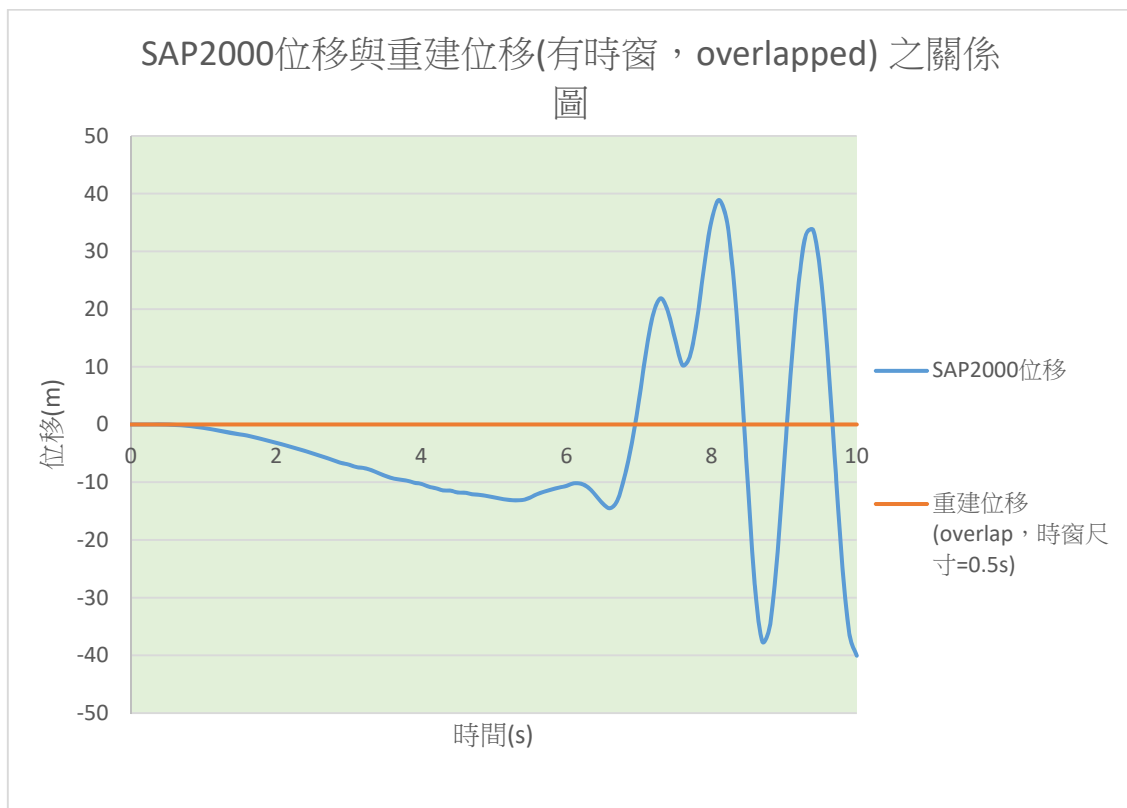


圖 5-12 25Hz 振動頻率下 SAP2000 位移與重建位移法(有時窗，overlapped)得到位移之關係圖(時窗大小為 0.1 秒)

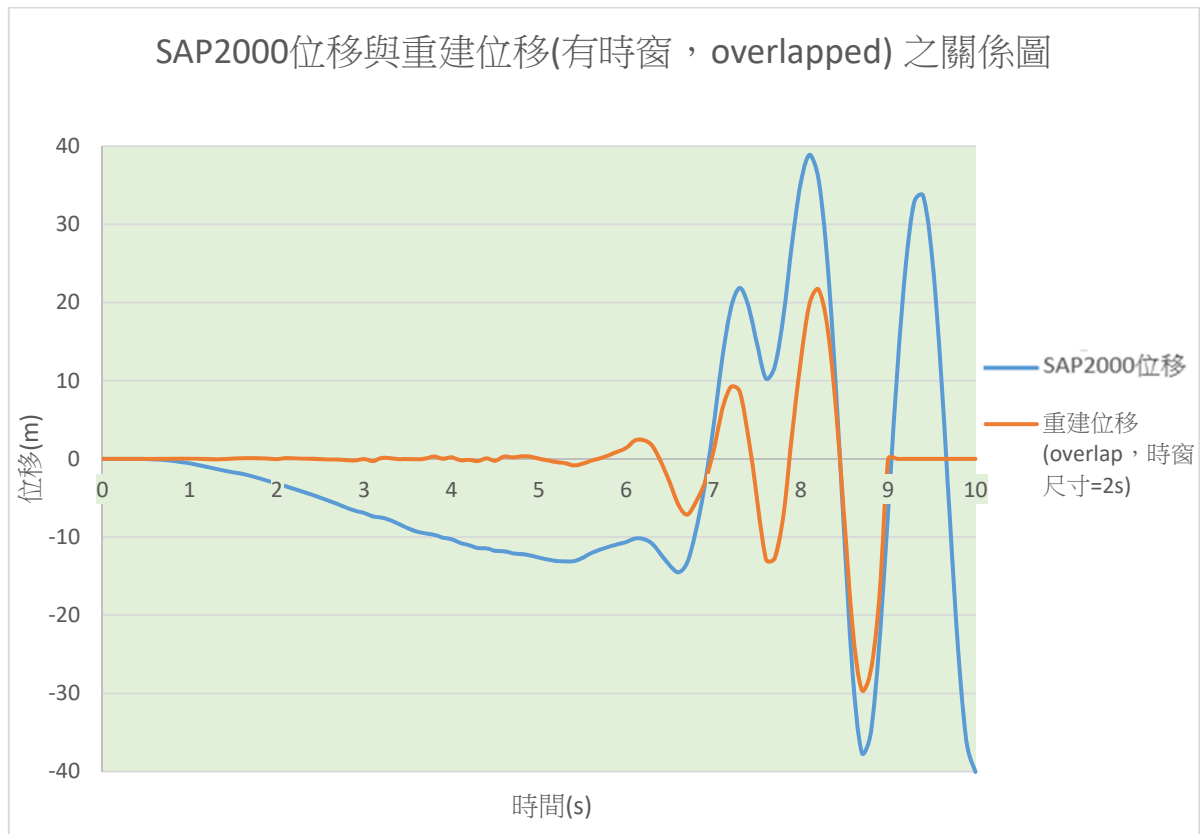


圖 5-13 25Hz 振動頻率下 SAP2000 位移與重建位移法(有時窗，overlapped)得到位移之關係圖(時窗大小為 2 秒)

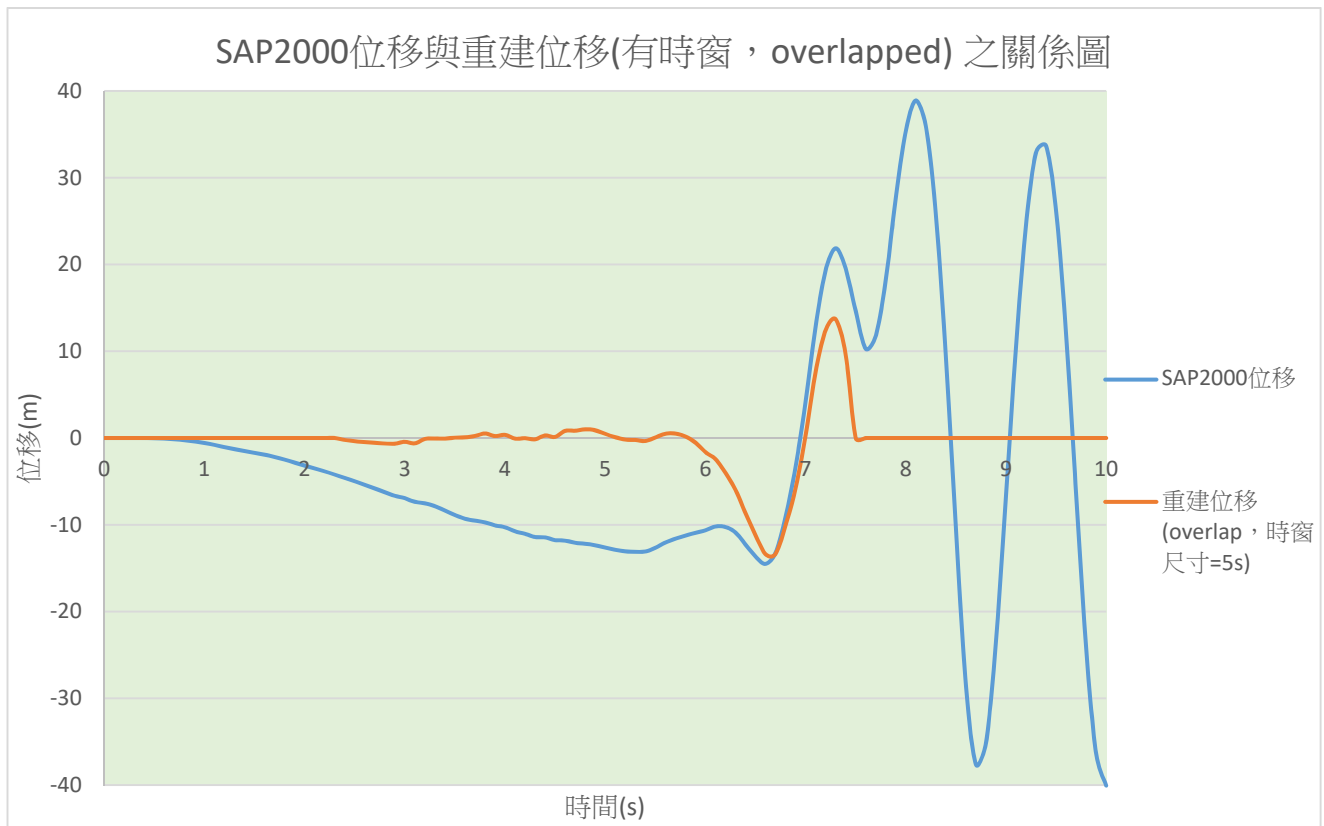


圖 5-14 25Hz 振動頻率下 SAP2000 位移與重建位移法(有時窗，overlapped)得到移之關係圖(時窗大小為 5 秒)

最後將以上試驗過的加速度轉換成位移的演算法與 SAP2000 位移做迴歸分析做做比較，整理如表 5-1，SAP2000 位移與各個演算法求出的位移迴歸分析圖可參考附錄十三。比較起來，跟 SAP2000 位移相關性最高的是梯形積分法與重建位移法(無時窗)計算出來的位移。

表 5-1 25Hz 振動頻率下懸臂梁自由端 SAP2000 位移與所有試驗演算法得出的位移迴歸分析之相關係數(R^2)

演算法名稱	相關係數(R^2)
梯形積分法	0.8028
重建位移法(無時窗)	0.9442
重建位移法(有時窗 nonoverlap，時窗尺寸=0.5s)	0.2946
重建位移法(有時窗 nonoverlap，時窗尺寸=1s)	0.4768
重建位移法(有時窗 nonoverlap，時窗尺寸=2s)	0.6124
重建位移法(有時窗 overlap，時窗尺寸=0.5s)	0.5569
重建位移法(有時窗 overlap，時窗尺寸=2s)	0.3193
重建位移法(有時窗 overlap，時窗尺寸=5s)	0.0857

二、6.5Hz 下每 0.002 秒的值

在 6.5Hz 的振動頻率並計算每 0.002 秒的加速度與位移，從振動的模擬下得出自由端 B 端的位移與加速度，將位移與加速度的數值資料做成圖，分別為圖 5-15 與圖 5-16。圖 5-15 是 B 端時間與加速度量的關係圖、圖 5-16 是 B 端時間與位移量的關係圖，從圖 5-15 與圖 5-16 可以發現到在十秒的試驗中，在一開始 B 點自由端的加速度浮動性比較大，位移也會因此在一開始的情況下，位移量就提高。但利用 6.5Hz 的振動頻率去振動懸臂梁(兩實驗的剛性係數、楊氏係數皆相同)與先前 25Hz 振動頻率下振動懸臂梁而產生的加速度值與位移值趨勢不一樣，6.5Hz 振動頻率下所產生的加速度與位移比 25Hz 振動頻率下產生的加速度與位移還要來的低這部分合理，但是 6.5Hz 振動頻率下在時間上一開始加速度與位移有很大的浮動情況(參考圖 5-15 與圖 5-16)，然而 25Hz 振動頻率下在時間上反而到後期才有比較明顯的加速度與位移變化(參考圖 5-5 與圖 5-6)。此部分試驗的加速度與位移對應時間的資料因為資料多達五千筆，故將不放入附錄。

此試驗的目的跟第一項試驗一樣，以上述章節所提到的演算法做計算轉換成位移，與 SAP2000 位移驗證所提方法計算結果判斷正確與合理性。

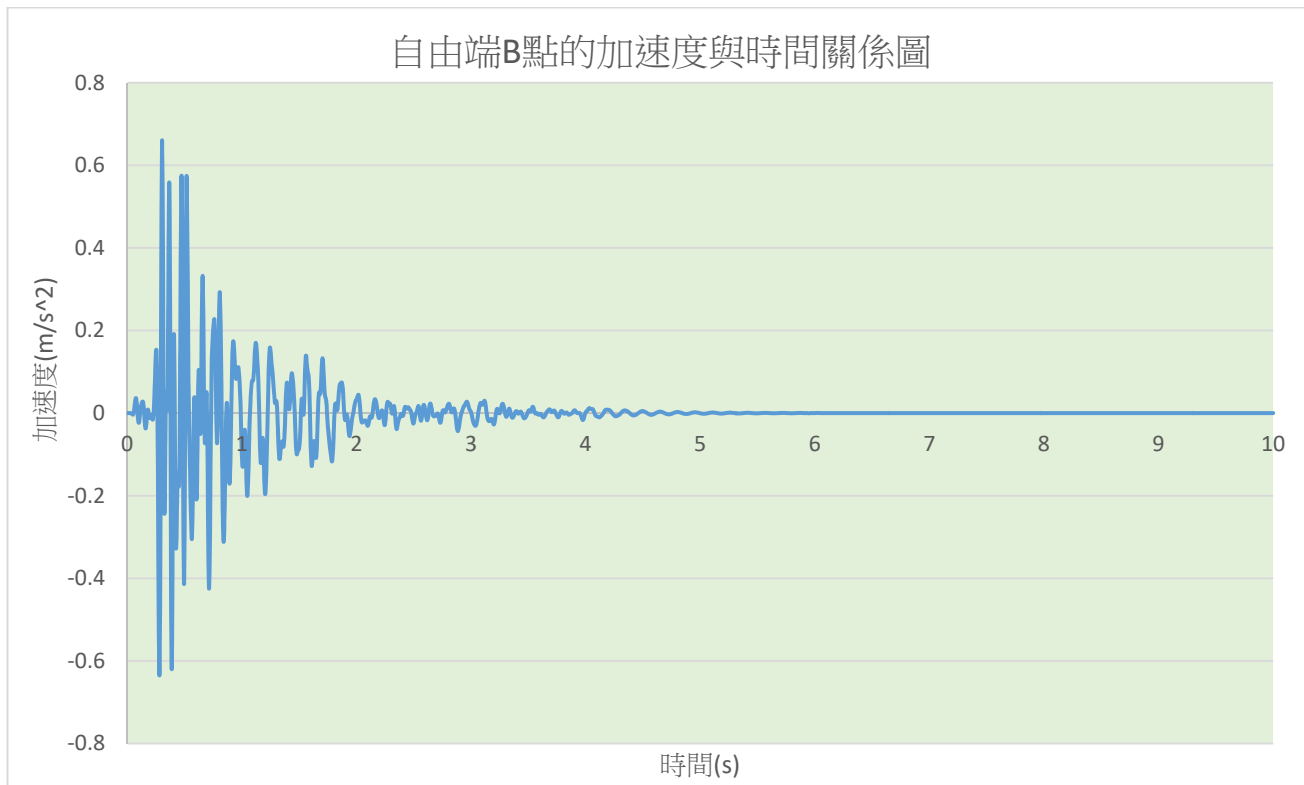


圖 5-15 數值分析模擬之懸臂梁在 6.5Hz 振動頻率下自由端時間與加速度量關係圖

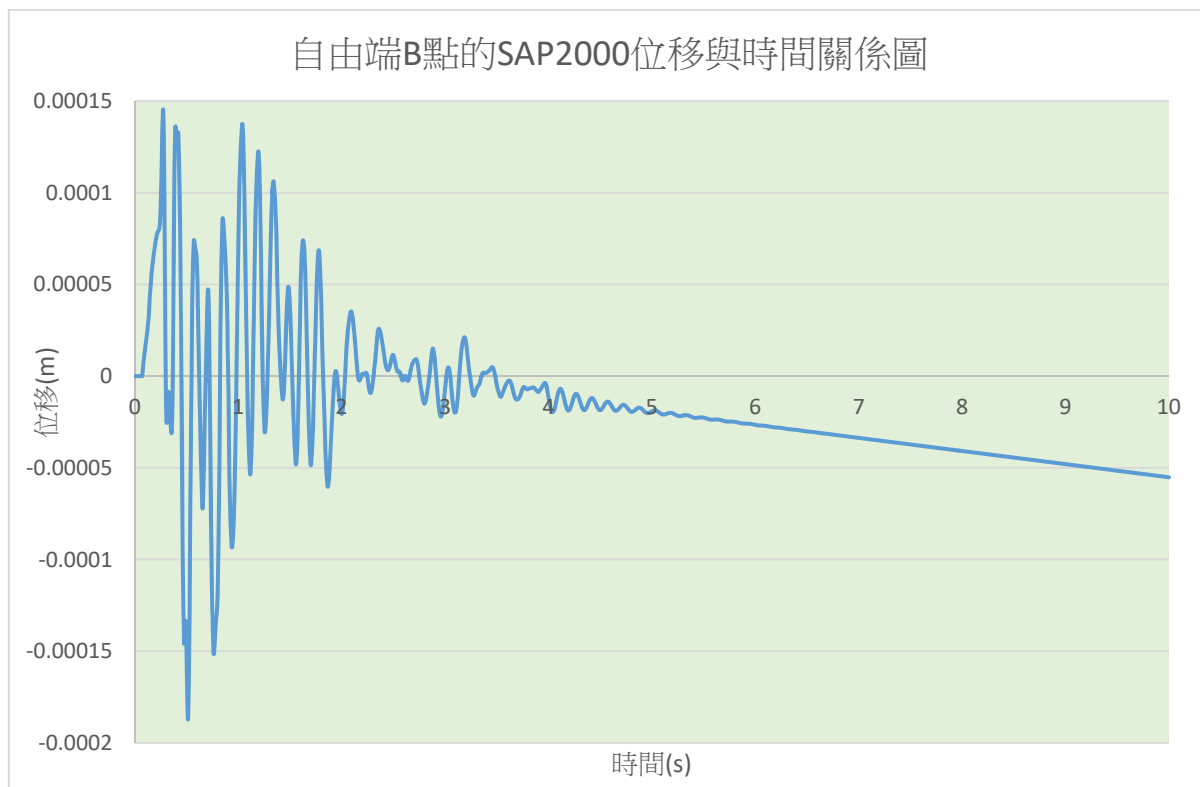


圖5-16 數值分析模擬之懸臂梁在6.5Hz振動頻率下自由端時間與SAP2000位移量關係圖

有加速度與 SAP2000 位移的資料後，我們將可使用四種把加速度轉換成位移的演算法，結果如下：

(一)梯形積分法(non-overlapped)

在梯形積分法做出來位移結果與 SAP2000 位移相比較，得到的如圖 5-17，發現這項演算法能幾乎與 SAP2000 位移吻合，不管是前期震盪或 4 秒以後的位移下降，幾乎都一樣。且兩者均出現嚴重的位移偏離，因為此人造地震並未進行基線修正，其結果說明梯形法並不適用於位移重建。

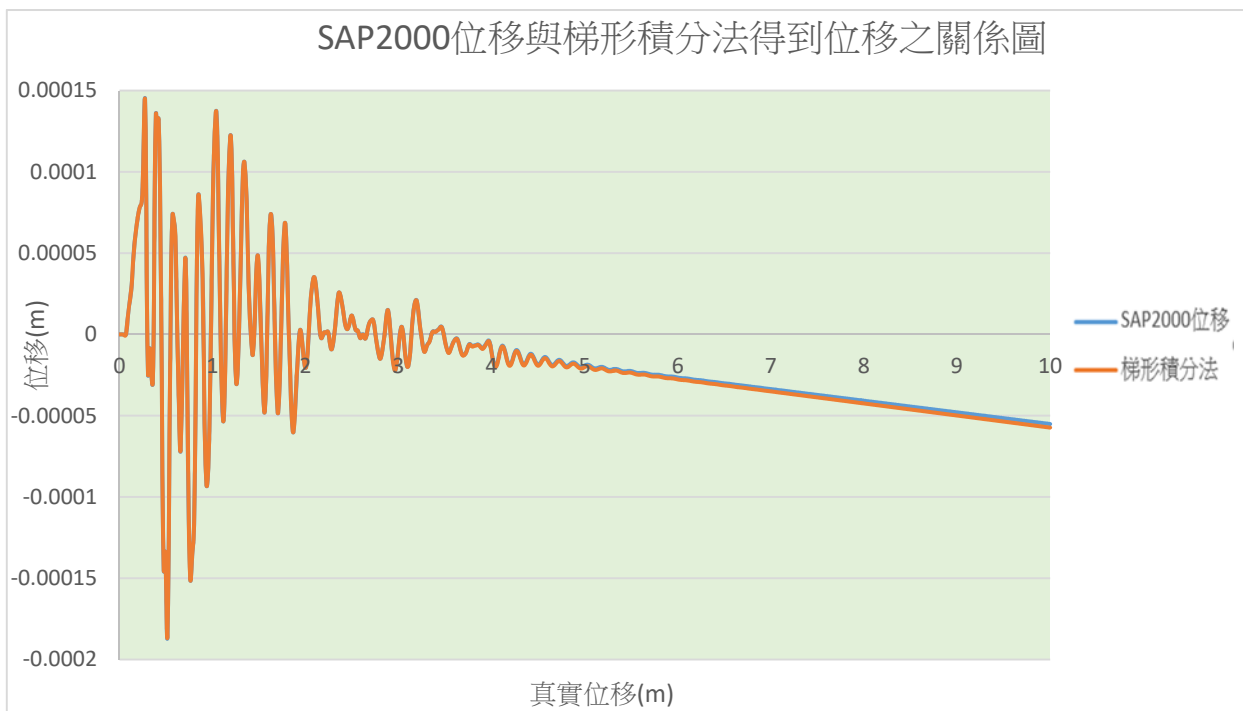


圖 5-17 6.5Hz 振動頻率下 SAP2000 位移與梯形積分法得到位移之關係圖

(二)重建位移法(無時窗)

圖 5-18 是 SAP2000 位移與重建位移法在無時窗的情況下的關係圖，大概到了 4 秒以後兩者的差異性漸漸增大，但 4 秒以前兩者的趨勢甚至數據相似度都相當的高，無時窗雖然從 Lee et al. (2010)的研究認為會比有時窗的誤差還大，但實測的結果認為後者有一點分歧，前面的相似程度是高的。另、由圖可知，此演算法不受基線修正與否之影響。

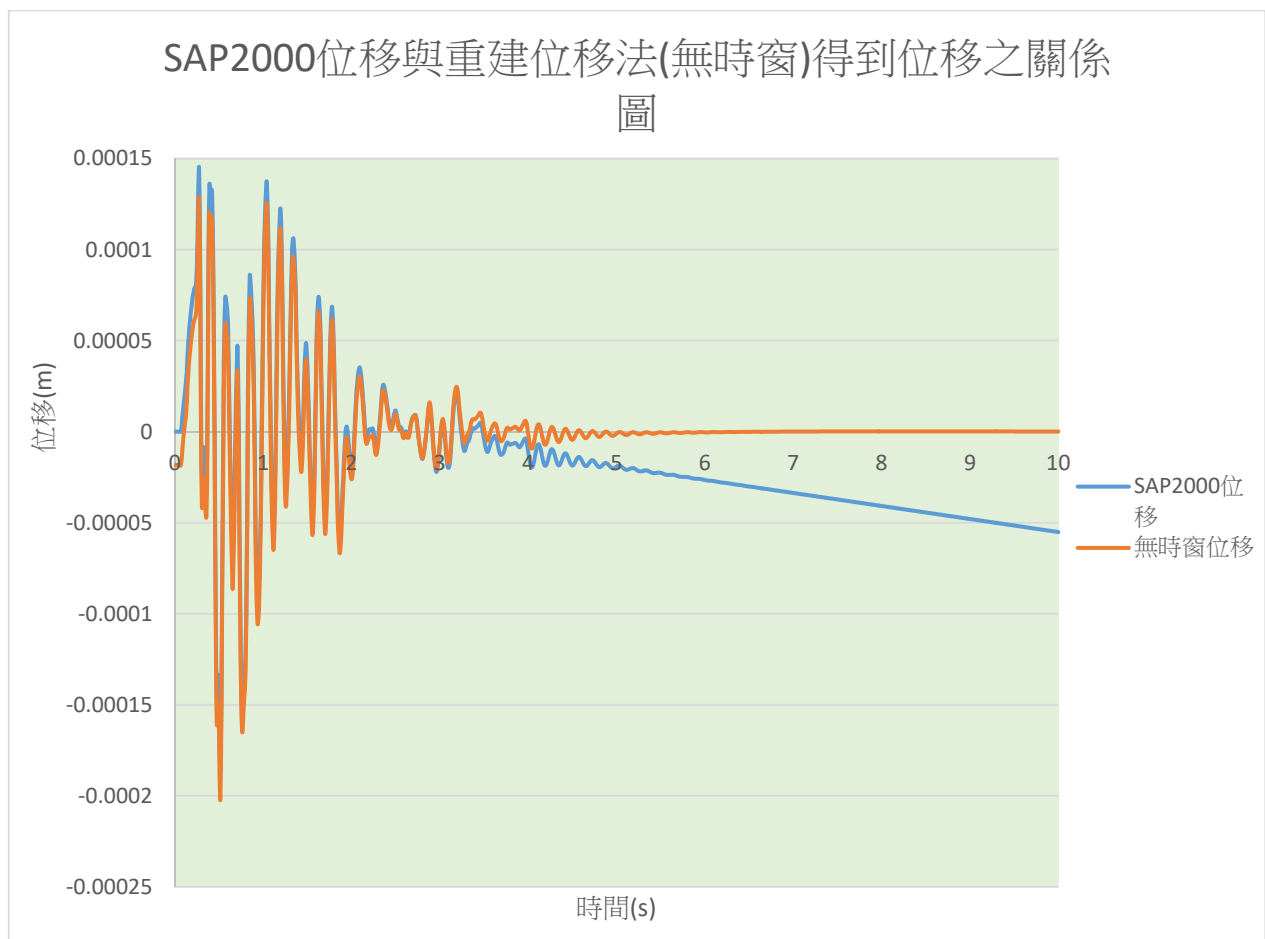


圖5-18 6.5Hz振動頻率下SAP2000位移與重建位移法(無時窗)得到位移之關係圖

(三)重建位移法(有時窗，non-overlapped)

圖 5-19 時窗數量是 20 個，在 10 秒的時間代表一格時窗的間格為 0.5 秒，圖 5-20 是使用了時窗有 10 個，時窗大小為 1 秒，圖 5-21 使用了 5 個時窗，時窗大小為 2 秒，三種情況不管時窗大小或個數，發現重建位移值與 SAP2000 位移值趨勢跟數值沒有差太多，之後做真實位移與演算法算出來的位移的相關性，比較相關係數或許就能看出差異性。

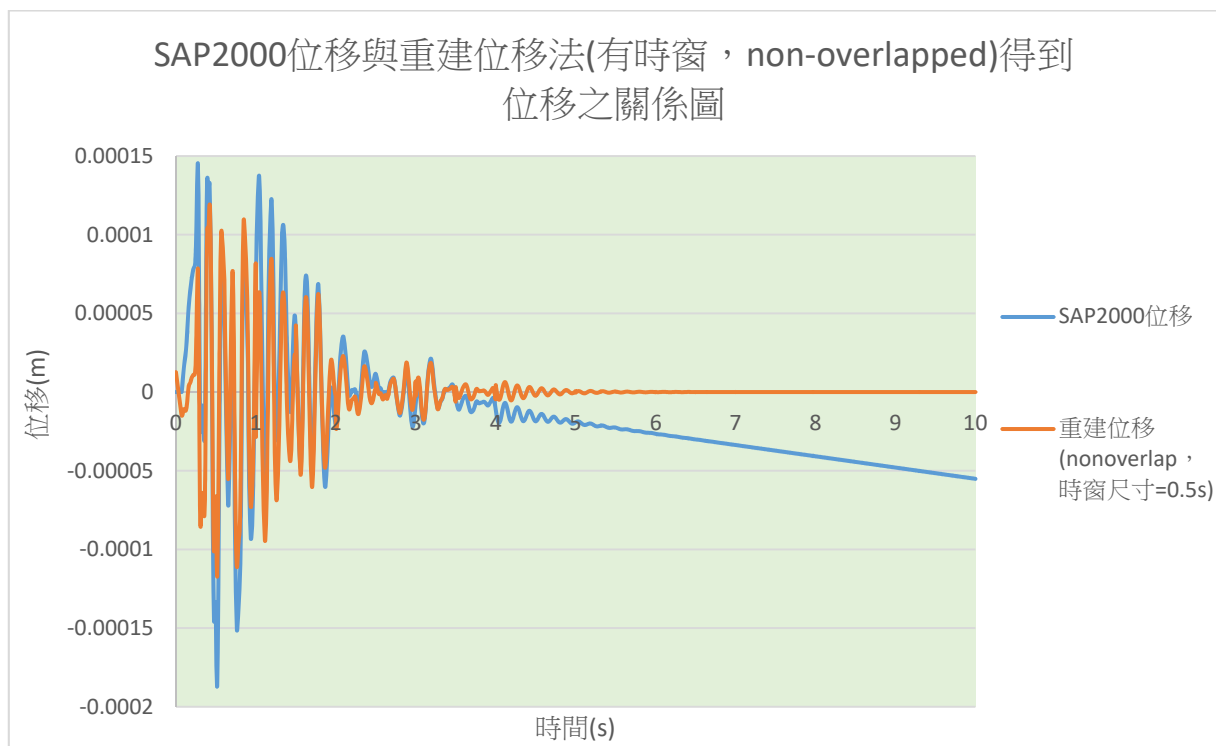


圖 5-19 6.5Hz 振動頻率下 SAP2000 位移與重建位移法(有時窗，non-overlapped)得到位移之關係圖(時窗有 20 個，時窗大小為 0.5 秒)

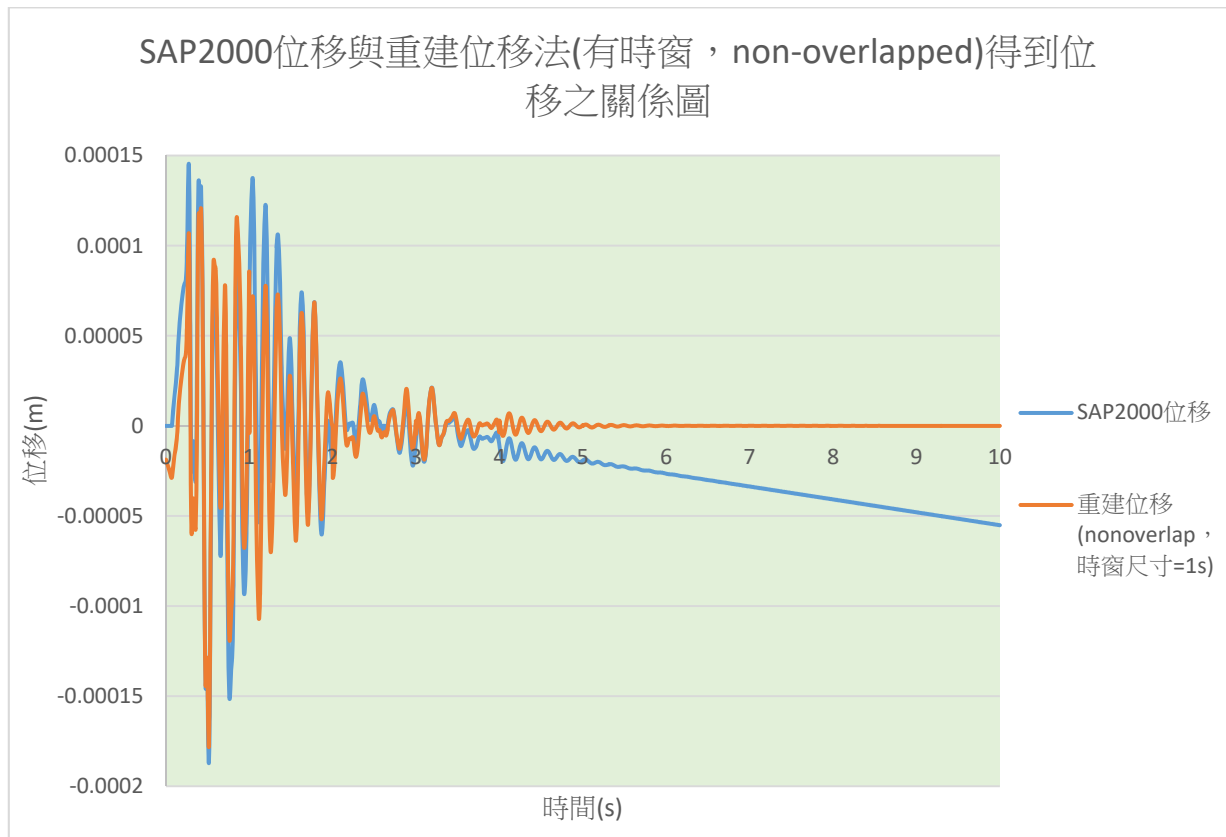


圖 5-20 6.5Hz 振動頻率下 SAP2000 位移與重建位移法(有時窗，non-overlapped)得到位移之關係圖(時窗有 10 個，時窗大小為 1 秒)

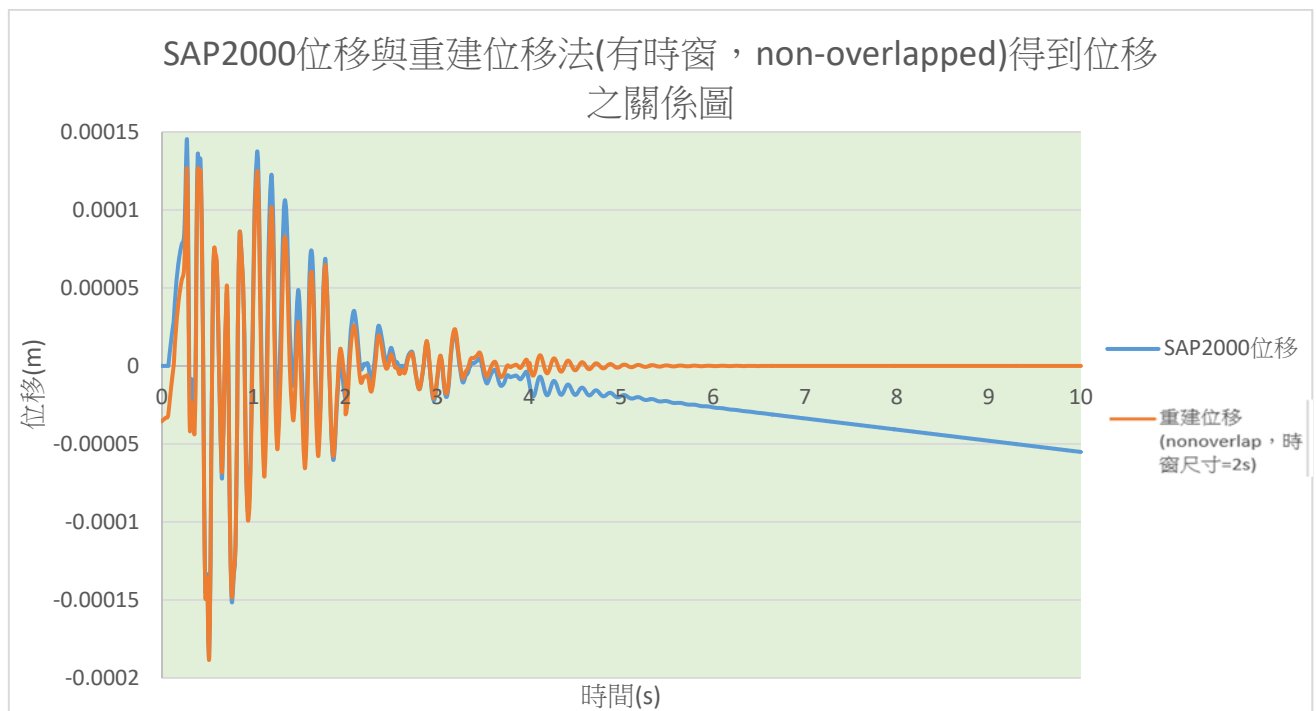


圖 5-21 6.5Hz 振動頻率下 SAP2000 位移與重建位移法(有時窗，non-overlapped)得到位移之關係圖(時窗有 5 個，時窗大小 2 秒)

(四)重建位移法(有時窗，overlapped)

圖 5-22、圖 5-23 與圖 5-24 是在不同的時窗尺寸與時窗個數下使用重建位移法與 SAP2000 位移去比較的關係圖，且有時窗並讓時窗彼此是覆蓋的狀態下將加速度轉換成位移。圖 5-22 是時窗尺寸為 0.5 秒去轉換位移，發現重建位移與 SAP2000 位移的趨勢與數值相似度皆相差甚遠；圖 5-23 是時窗尺寸為 1 秒去轉換位移，雖然兩者的趨勢跟數值相似度有好轉，但跟前面的無視窗或者梯形積分法或有時窗且時窗間無覆蓋的情況，落差還是相當大；圖 5-24 是時窗尺寸為 2.5 秒去轉換位移。而此次的 overlap 試驗發現到有符合 Lee et al. (2010)的情況，時窗數越多，時窗越小，越能夠接近 SAP2000 位移。

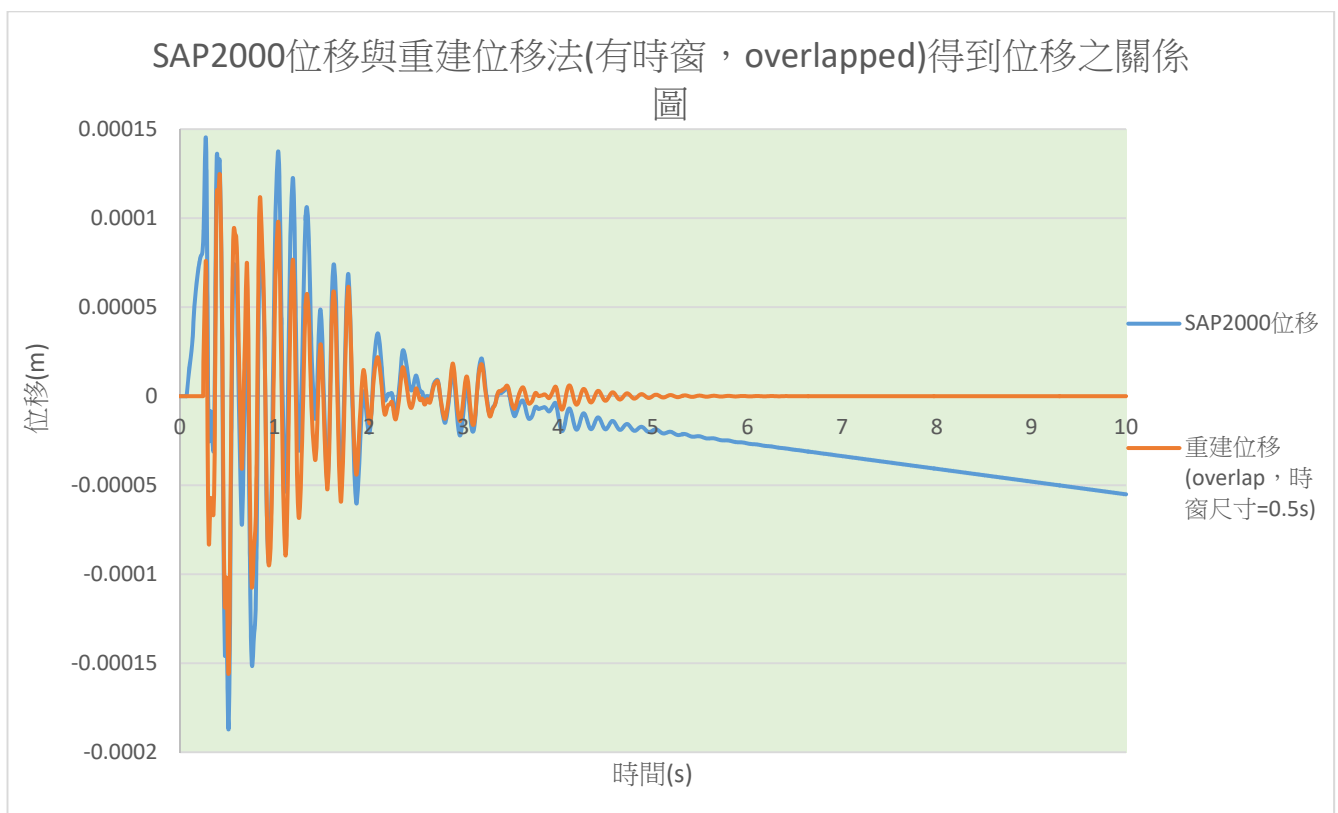


圖 5-22 6.5Hz 振動頻率下 SAP2000 位移與重建位移法(有時窗，overlapped)得到位移之關係圖(時窗大小為 0.5 秒)

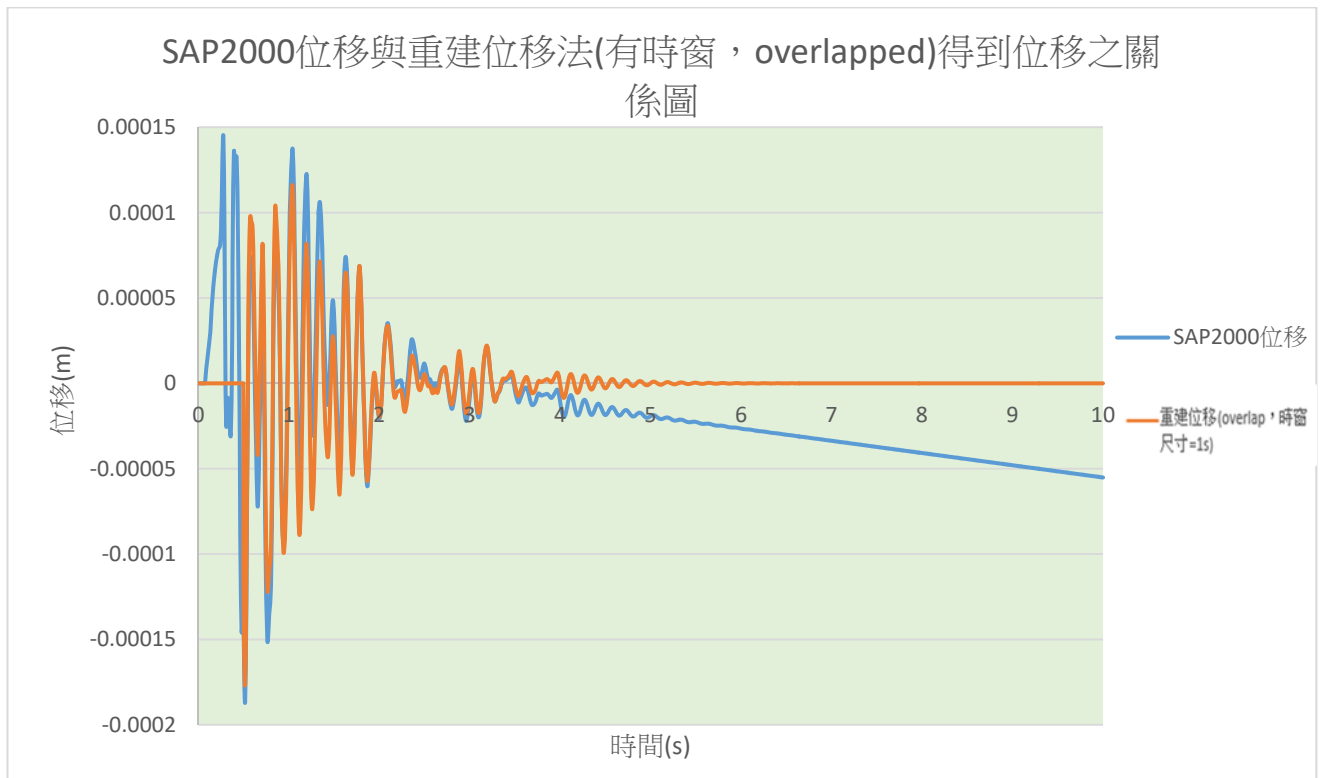


圖 5-23 6.5Hz 振動頻率下 SAP2000 位移與重建位移法(有時窗，overlapped)得到位移之關係圖(時窗大小為 1 秒)

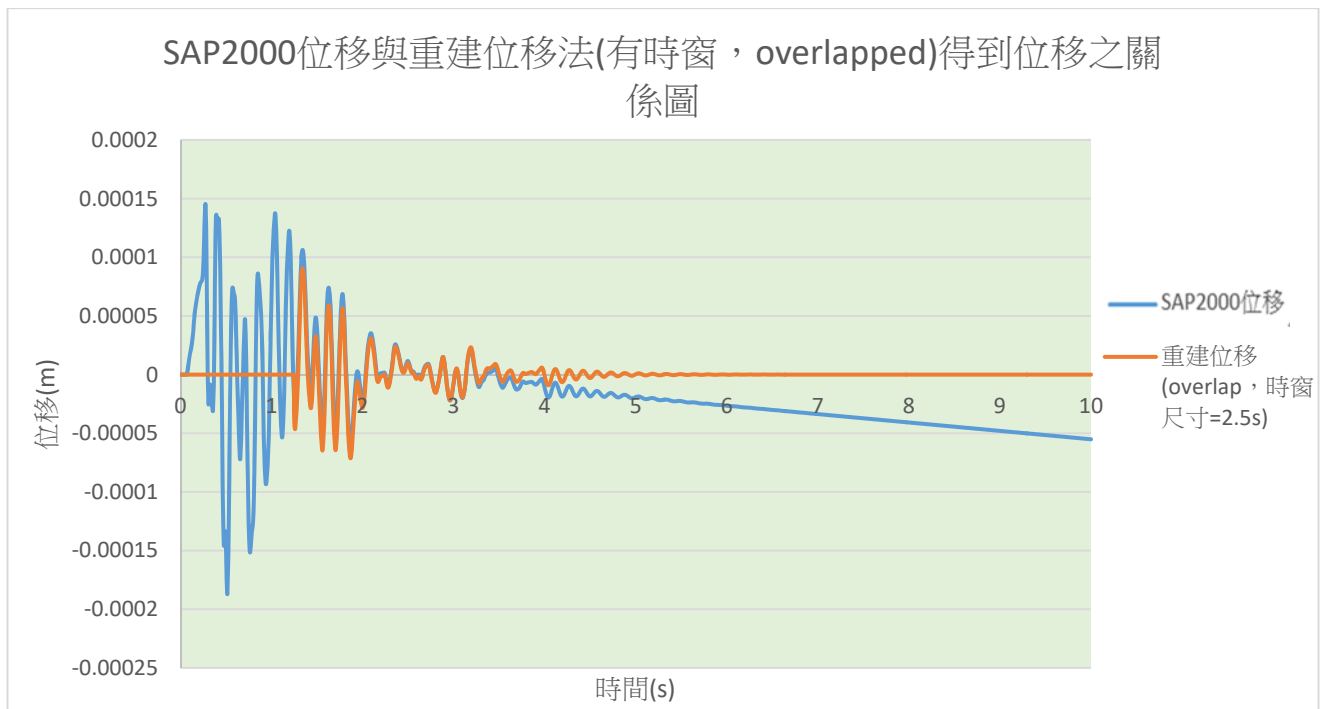


圖 5-24 6.5Hz 振動頻率下 SAP2000 位移與重建位移法(有時窗，overlapped)得到位移之關係圖(時窗大小為 2.5 秒)

最後將以上試驗過的加速度轉換成位移的演算法與 SAP2000 位移做迴歸分析並比較，整理如表 5-2，SAP2000 位移與各個演算法求出的位移迴歸分析圖可參考附錄十四。比較起來，跟 SAP2000 位移相關性最高的依然是梯形積分法與重建位移法(沒有時窗)計算出來的位移，與第一項試驗一樣。在表 5-2 的相關係數比較發現 overlap 效果會比 nonoverlap 的效果還好，但相關性與無視窗和梯形積分法還是低了些，總之兩次的試驗下來發現有時窗的方法其與 SAP2000 位移的接近程度都不太高。然而，通常將加速度直接做積分成位移是一個誤差極大的做法，但以上的結果，卻是梯形積分法是最好的演算方式；此外，重建位移法(無時窗)的情況也會比較精確的位移值，與 Lee et al. (2010)的觀念相衝突。而這部分的數值模擬結果與預期的大不相同，故我們將使用後續的室內試驗來確認是否軟體的模擬會與室內試驗有著一致的結果。

表 5-2 6.5Hz 振動頻率下懸臂梁自由端 SAP2000 位移與所有試驗演算法得出的位移迴歸分析之相關係數(R^2)

演算法名稱	相關係數(R^2)
梯形積分法	0.9997
重建位移法(無時窗)	0.6621
重建位移法(有時窗 nonoverlap，時窗尺寸=0.5s)	0.4898
重建位移法(有時窗 nonoverlap，時窗尺寸=1s)	0.5479
重建位移法(有時窗 nonoverlap，時窗尺寸=2s)	0.6442
重建位移法(有時窗 overlap，時窗尺寸=0.5s)	0.5247
重建位移法(有時窗 overlap，時窗尺寸=1s)	0.4167
重建位移法(有時窗 overlap，時窗尺寸=2.5s)	0.1106

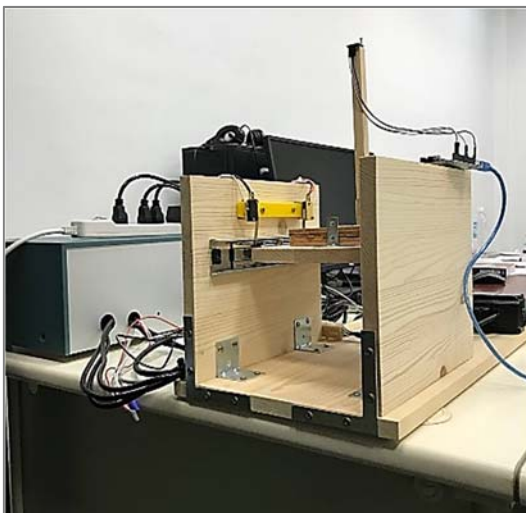
第二節 室內縮尺度驗證

除了以數值分析驗證所提演算法之正確性外，本計畫亦將進行室內縮尺度試驗，目前計畫利用懸臂梁進行試驗，懸臂梁如圖 5-25 所示，並利用圖 5-26 之振動臺對懸臂梁進行振動，震動時以安裝於結構物上的加速度計(位於懸臂梁的自由端)紀錄加速度，利用所提演算法推求位移。將會藉由前面的數值分析之各式演算法推算懸臂梁的位移。室內試驗除了分析得出的位移值前，需驗證加速度值的合理性。



圖 5-25 室內模擬之懸臂梁模型

(a)



(b)



圖 5-26 振動臺

一、加速度值之驗證

加速度計雖然可以產生的加速度值以便做後續的分析與研究使用，但是加速度值的合理性是需要再藉由其他方式確定其合理性的，故將使用快速傅立葉函數 (FFT) 將加速度值轉換為頻率，並將該頻率比較原始振動頻率做比較。方法為實際輸入一組參數與 Matlab 的 FFT 函式來作比較。本次試驗使用 MMA8451Q 加速度計與 Arduino UNO 板結合的智慧感測器，將來將會試驗 MMA7660FC 加速度計與 Arduino M0 Pro 板結合的智慧感測器驗證是否加速度值合理。

快速傅立葉轉換 FFT，是快速計算序列的離散傅立葉轉換 (DFT) 或其逆轉換的方法，常見的傅立葉分析會將時域訊號轉換到頻率域來表示，亦或是逆過來轉換。

在 Arduino IDE 開發環境中，本身有內建 FFT 相關函式庫能提供使用者運用此演算法，其中輸入條件必須以 2 的次方的資料長度輸入，因此可以根據需求，輸入適合的資料數量。

(一) Matlab 運算驗證

Arduino IDE 雖然有提供 FFT 函式庫供使用者們使用，但本計畫階段發現實際輸入參數後所求得的主頻並不如預期，所幸開發環境使用的語法與 C 語言近乎一模一樣，因此直接寫入一個副程式來計算量測得的加速度值，實現在 Arduino 開發板上 FFT 的計算。

如圖 5-27，從 Arduino Serial Monitor 中可看出求得的數值是 Arduino 開發板輸入由 MMA8451 測得當 ODR 為 54 時，X 方向加速度值共 1024 筆資料，經由 FFT 副程式運算所求得的實部與虛部係數。Matlab 也輸入相同的 1024 筆加速度值資料後得出實部與虛部，兩者比較可發現運算結果近乎相同。

圖 5-28 為求得主頻之驗證，每次蒐集完成 1024 筆加速度值後執行 FFT 運算共花費約 19 秒鐘得到 FFT 後的實數與虛數，將實數與虛數可得頻率值， $1024/19 \div 53.894$ 取 $ODR = 54$ 。經由 FFT 運算後頻率域上限為 27 Hz，每一個頻率內有 19 個資料點是經由時間域轉換而成，因此從 Matlab 頻率域陣列

中的元素裡找出最大值可以發現在位於第 31 格的元素，將 $31/19 \div 1.63157$ Hz，得到的主頻與 Arduino 開發板上運算得到的主頻相近而得證。

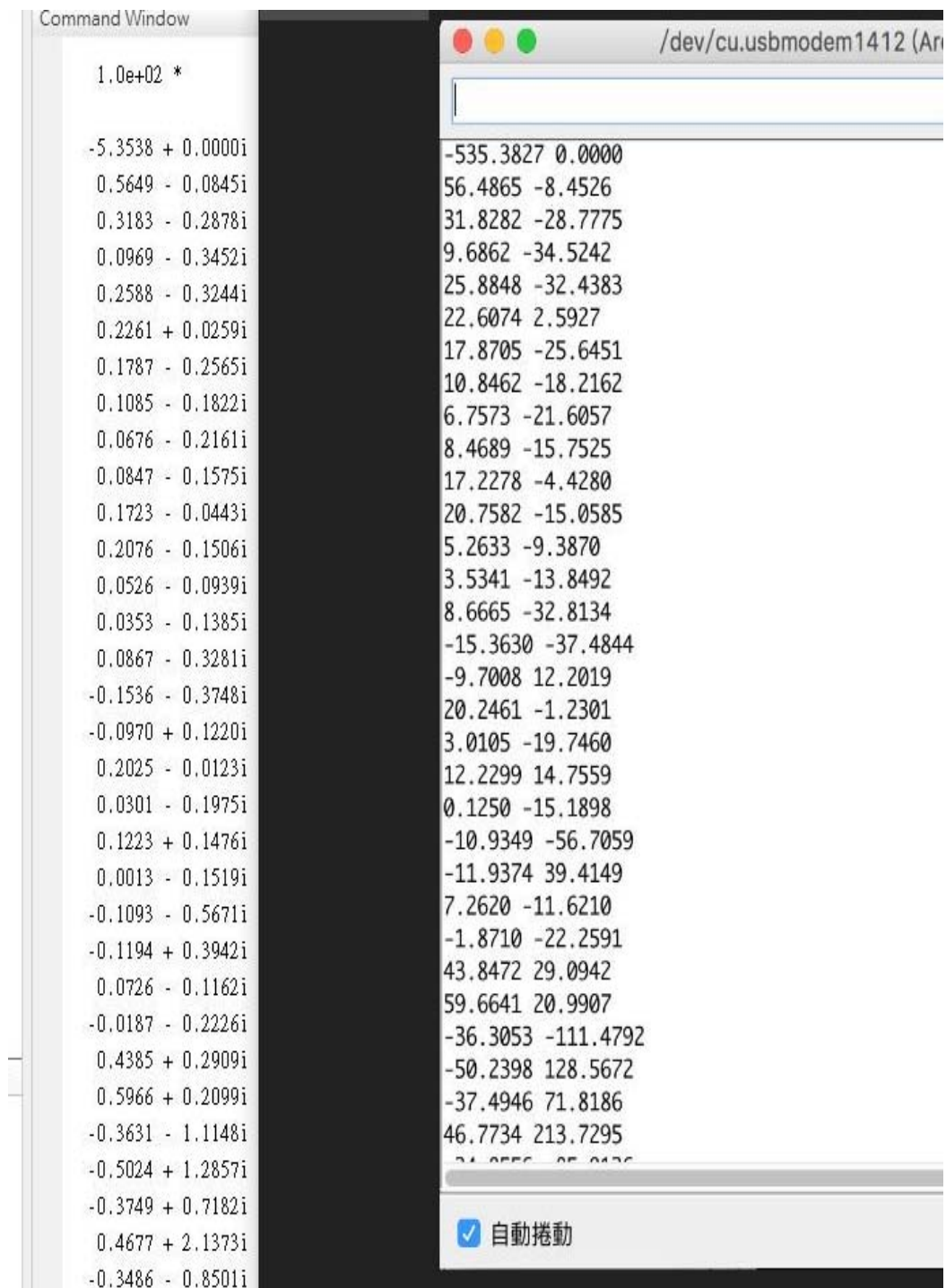


圖 5-27 FFT 副程式與 Matlab 函式比較圖

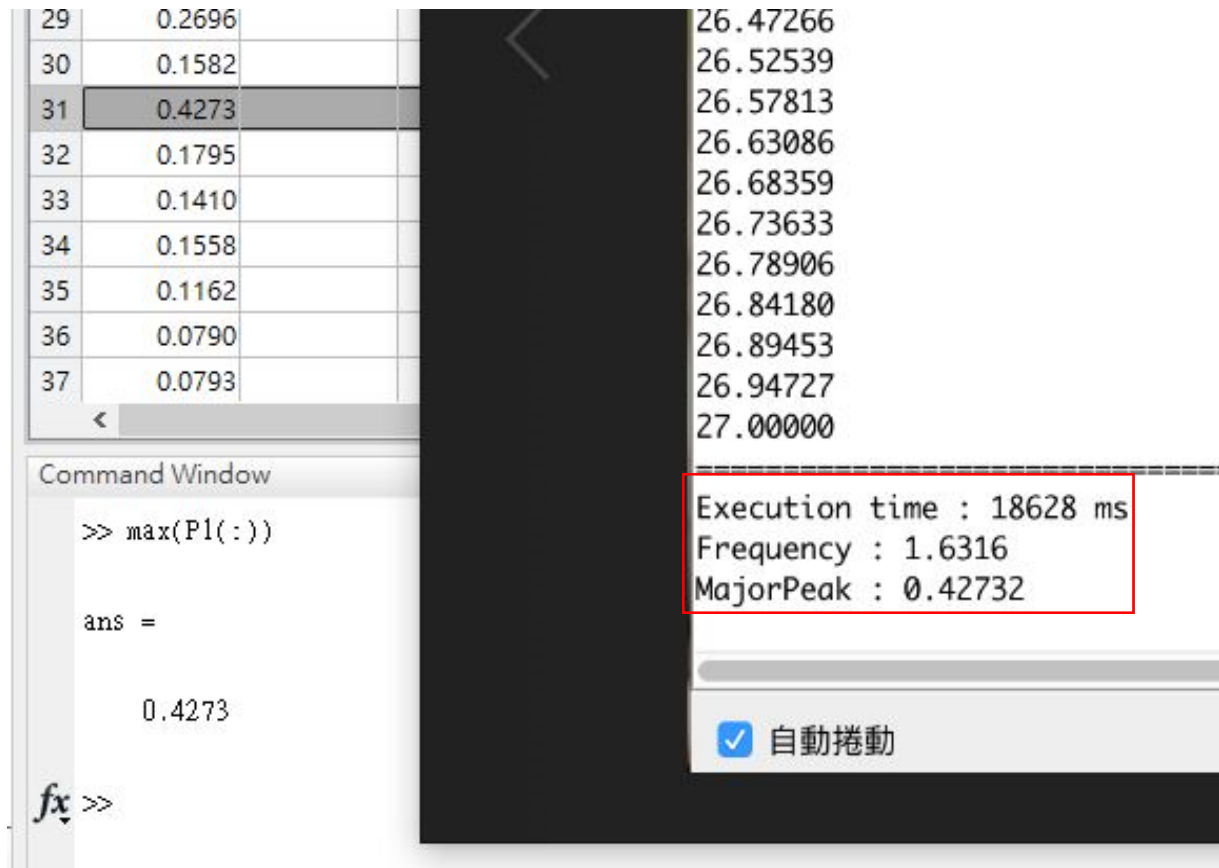


圖 5-28 主頻驗證

(二) 振動臺試驗

此試驗為使用振動臺，試驗情況如圖 5-29，將加速度計與 Arduino 板和 SD 卡在振動臺上以固定頻率為 1 Hz 的正弦波做簡諧運動，圖 5-30 為輸入於振動臺的測試波形訊號，透過本研究實驗模組來量測固定振動臺所提供的單軸振動變化，並且進行 FFT 運算取得主頻。

圖 5-31 是由本研究實驗模組測得的加速度進行 FFT 運算後所輸出的頻譜，從圖中可以很明顯看出，其主要最高峰出現在大約 1 Hz 附近，且陣列內的最大值出現在第 18 格的元素，因此如前面敘述， $18/19 = 0.947$ Hz，接近預期的輸出頻率也藉此驗證此實驗。

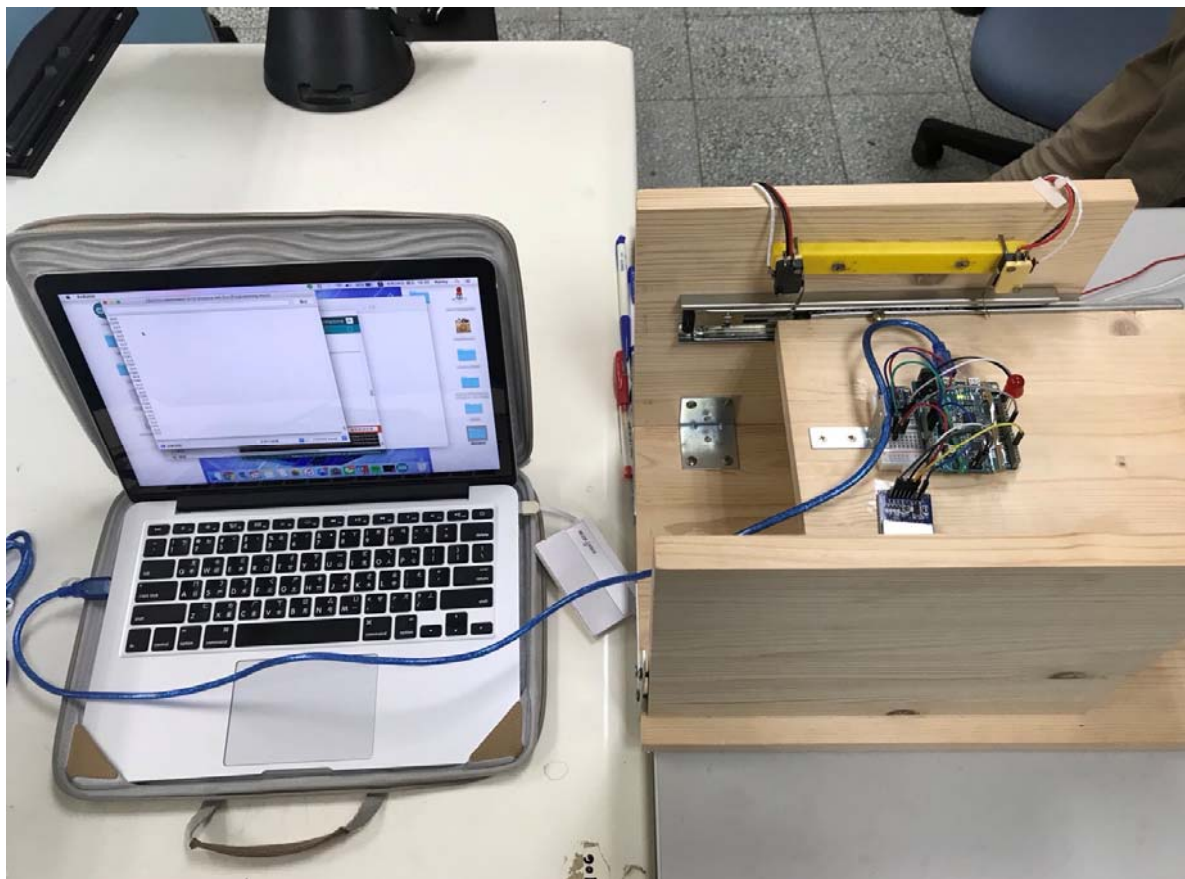


圖 5-29 振動臺試驗示意圖

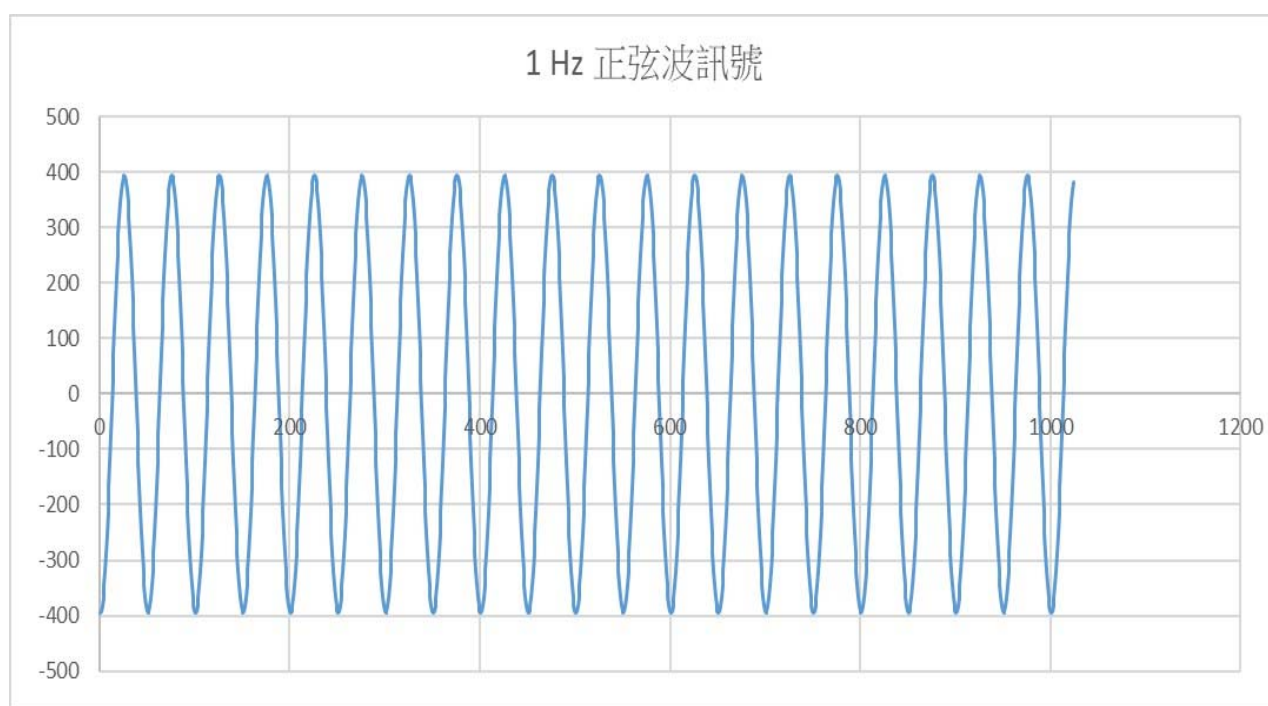


圖 5-30 1 Hz 輸入正弦波形

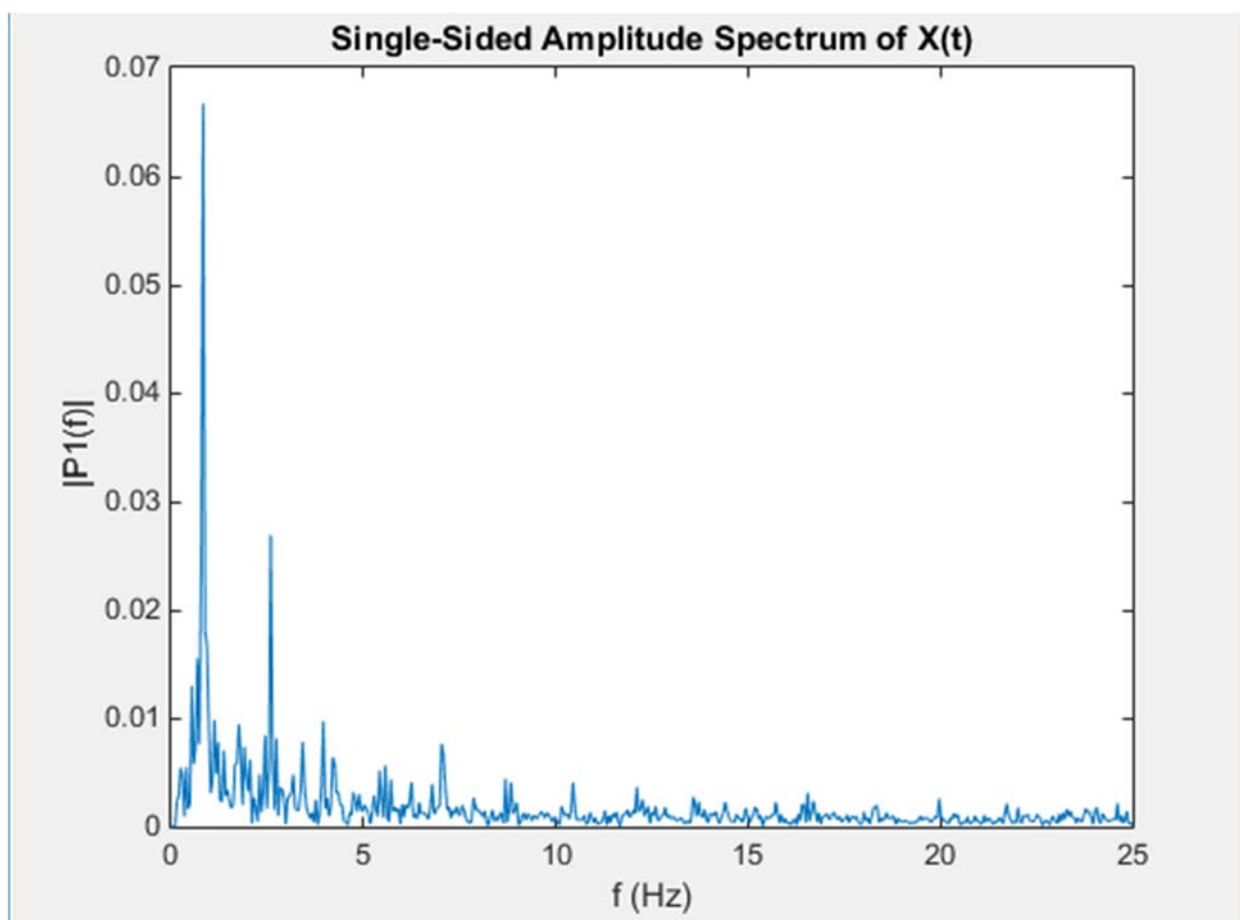


圖 5-31 FFT 運算輸出頻譜

二、加速度換算位移之結果

目前有實測出的加速度後，我們將可使用如上一節所提到的四種把加速度轉換成位移的演算法，結果如下：

(一)梯形積分法

在梯形積分法做出來位移結果與振動的真實位移相比較，得到的如圖 5-32 的結果，會發現到因為梯形積分法受到基線修正與否的影響很大，故加速度轉換出來的位移值與真實位移就會有極大的落差。

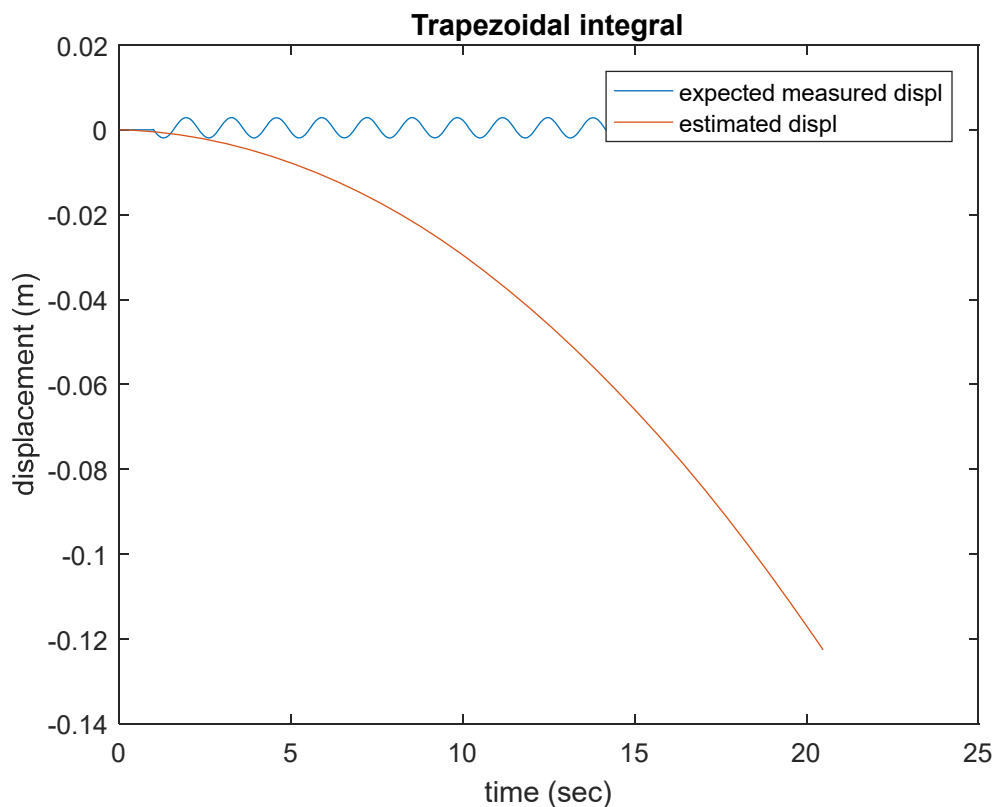


圖 5-32 1Hz 振動頻率下真實位移與梯形積分法得到位移之關係圖

(二)重建位移法(無時窗)

圖 5-33 是真實位移與重建位移法在無時窗的情況下的圖，從這次結果可以發現到與 Lee et al. (2010)的想法吻合，雖然使用了重建位移法，不是直接對加速度做積分，但在沒有時窗的情況下，兩者的趨勢與數值有很大的差異。

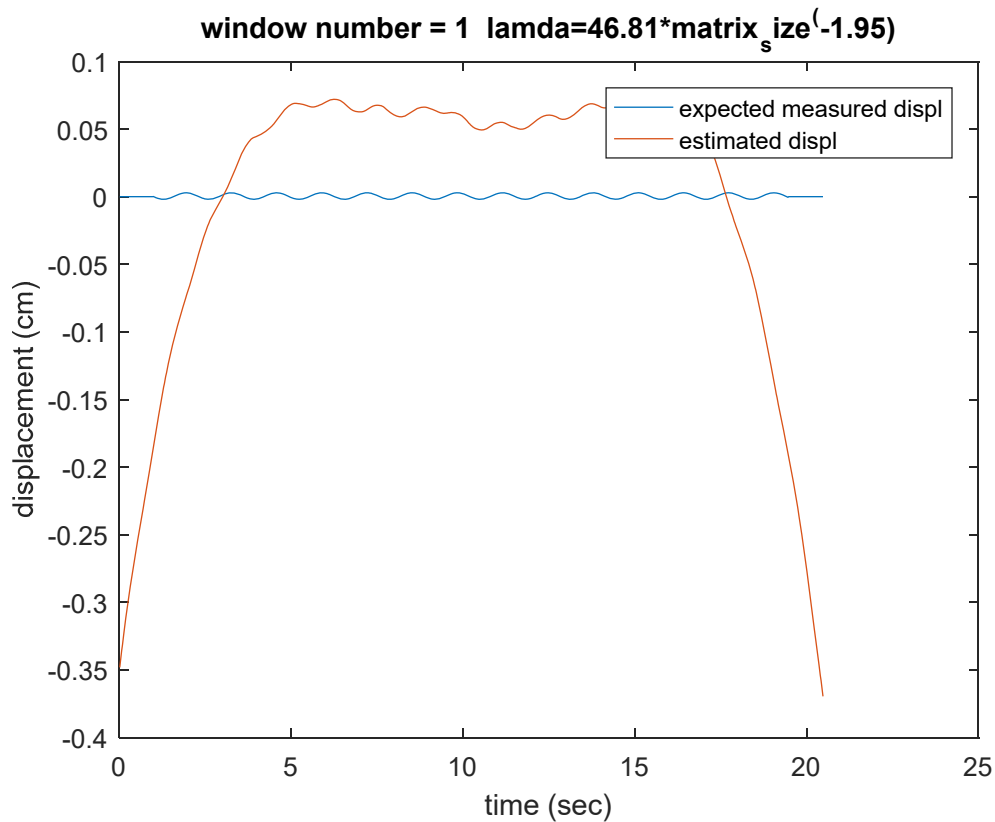


圖5-33 1Hz振動頻率下真實位移與重建位移法(無時窗)得到位移之關係圖

(三)重建位移法(有時窗，non-overlapped)

上一個重建位移法沒有時窗趨勢性很差，故加上了有時窗的方式去做演算，結果如圖 5-34、圖 5-35 與圖 5-36。圖 5-34 是 5 個時窗，每個時窗 4 秒的結果；圖 5-35 是 10 個時窗，每個時窗 2 秒的結果；圖 5-36 是 20 個時窗，每個時窗 1 秒的結果。三種圖的換算位移與真實位移過後發現到，從圖 5-34 與圖 5-35 的比較，發現時窗 10 個與 5 個時窗的重建位移換算與真實位移相比，10 個時窗會比較接近真實位移值，但發現在圖 5-36 時窗 20 個的圖發現這樣的重建位移與真實位移的接近程度並沒有時窗 10 個這樣高。

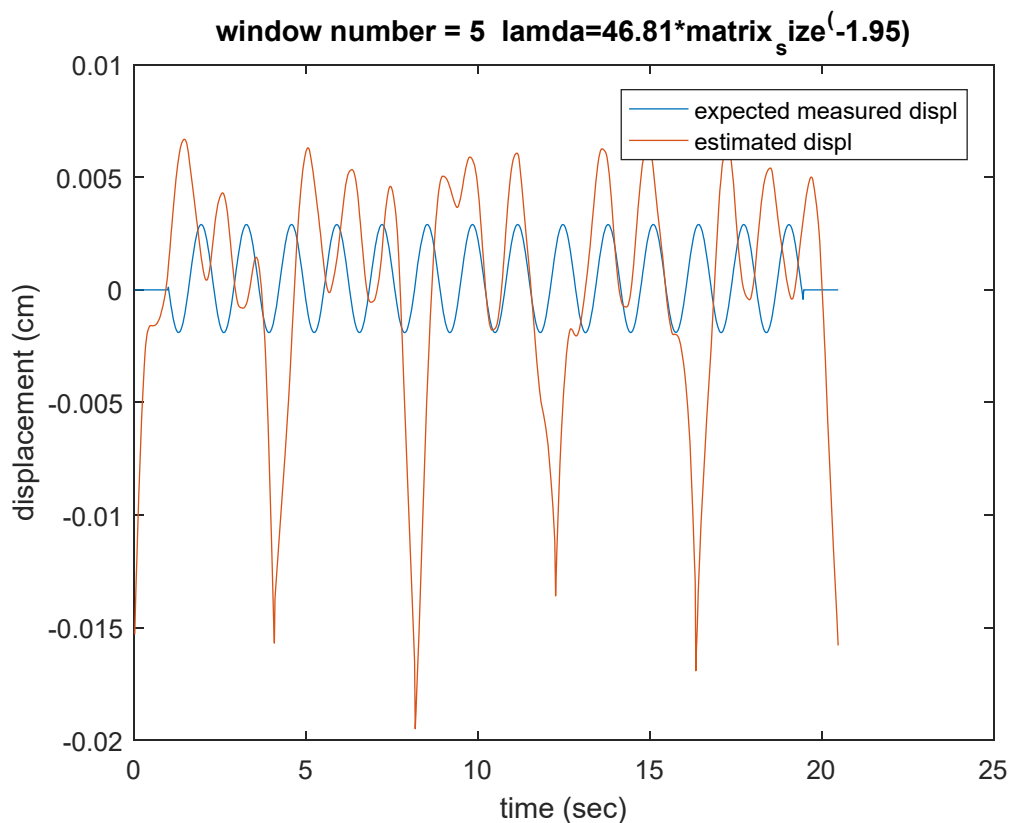


圖 5-34 1Hz 振動頻率下真實位移與重建位移法(有時窗，non-overlapped)得到位移之關係圖(時窗有 5 個，時窗大小為 4 秒)

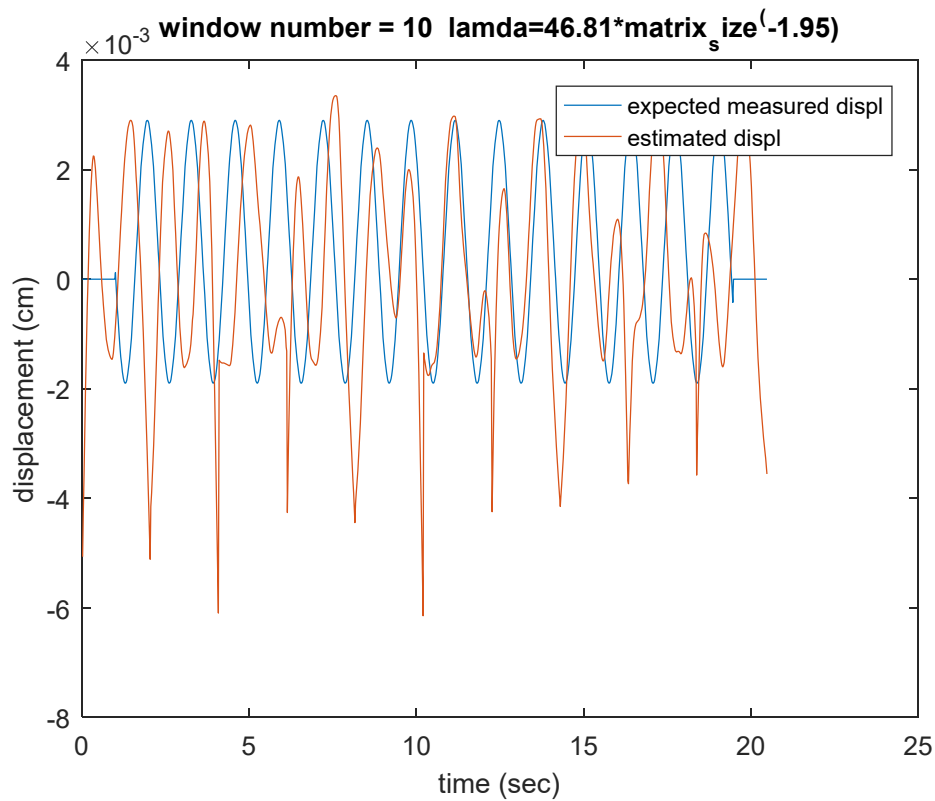


圖 5-35 1Hz 振動頻率下真實位移與重建位移法(有時窗，non-overlapped)得到位移之關係圖(時窗有 10 個，時窗大小為 2 秒)

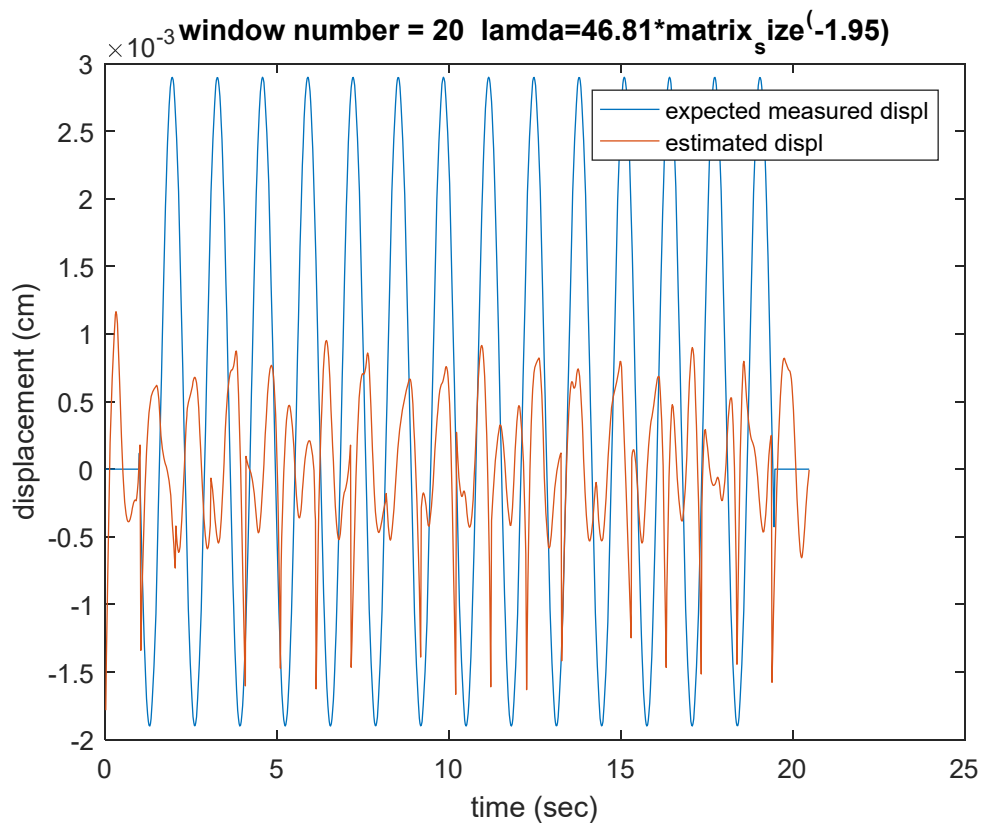


圖 5-36 1Hz 振動頻率下真實位移與重建位移法(有時窗，non-overlapped)得到位移之關係圖(時窗有 20 個，時窗大小 1 秒)

(四)重建位移法(有時窗，overlapped)

重建位移法增加了時窗後與真實位移相比，兩者相似度有提高許多，而如前面所提到的，因為加速度轉換成位移如果每個時窗沒有相互覆蓋，則兩時窗間的邊界數值就會有所誤差，故使用時窗覆蓋的方式去做演算，結果如圖 5-37、圖 5-38 與圖 5-39。圖 5-37 是 5 個時窗，每個時窗 4 秒的結果；圖 5-38 是 10 個時窗，每個時窗 2 秒的結果；圖 5-39 是 20 個時窗，每個時窗 1 秒的結果。三種圖的換算位移與真實位移比較過後發現到，且確實時窗相互覆蓋的重建位移法也跟真實位移接近許多，但圖 5-37 到圖 5-39 與前面 non-overlap 情況類似，時窗 10 個與 5 個時窗的重建位移換算與真實位移相比，10 個時窗會比較接近真實位移值，也在時窗 20 個的圖發現這樣的重建位移與真實位移的接近程度並沒有時窗 10 個這樣高。

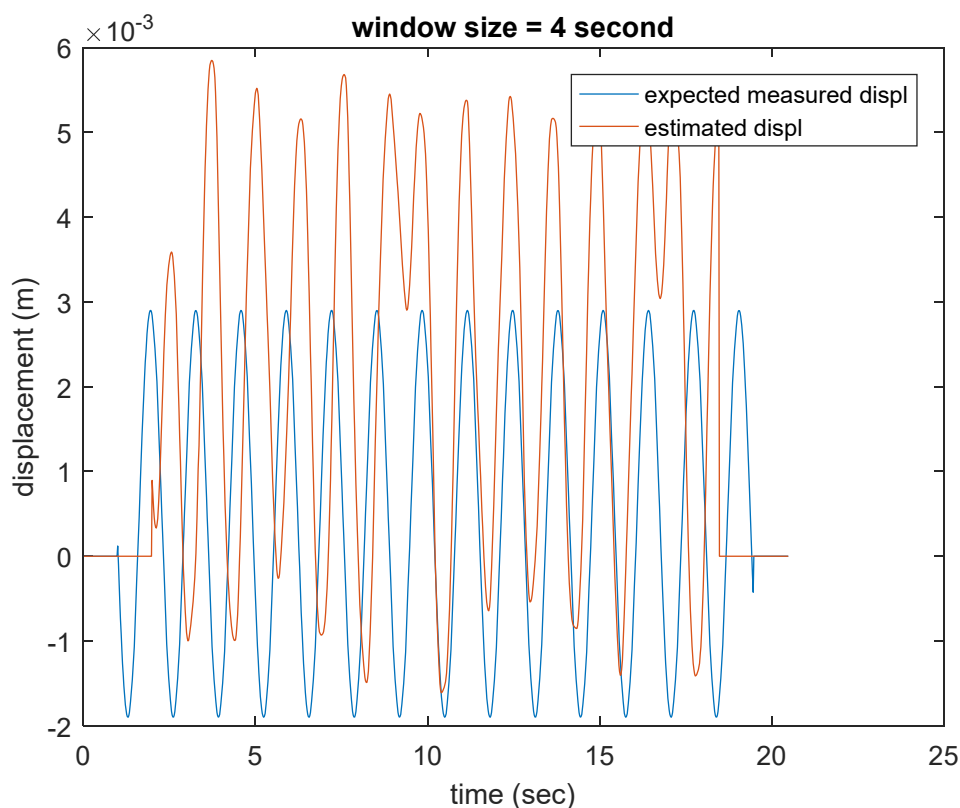


圖 5-37 1Hz 振動頻率下真實位移與重建位移法(有時窗，overlapped)得到位移之關係圖(時窗有 5 個，時窗大小為 4 秒)

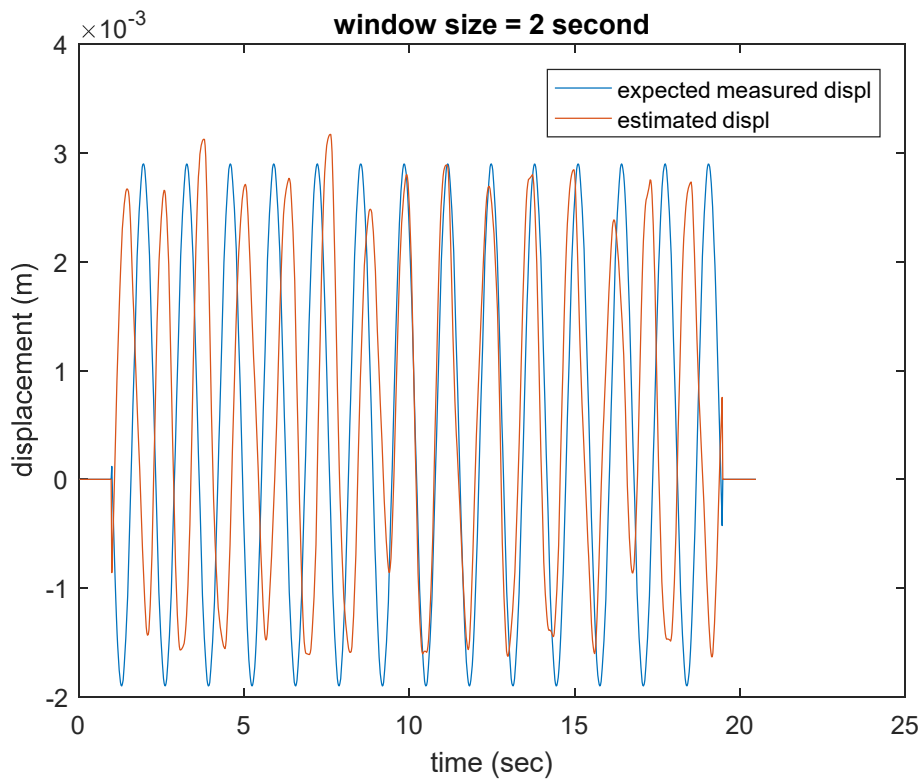


圖 5-38 1Hz 振動頻率下真實位移與重建位移法(有時窗，overlapped)得到位移之關係圖(時窗有 10 個，時窗大小為 2 秒)

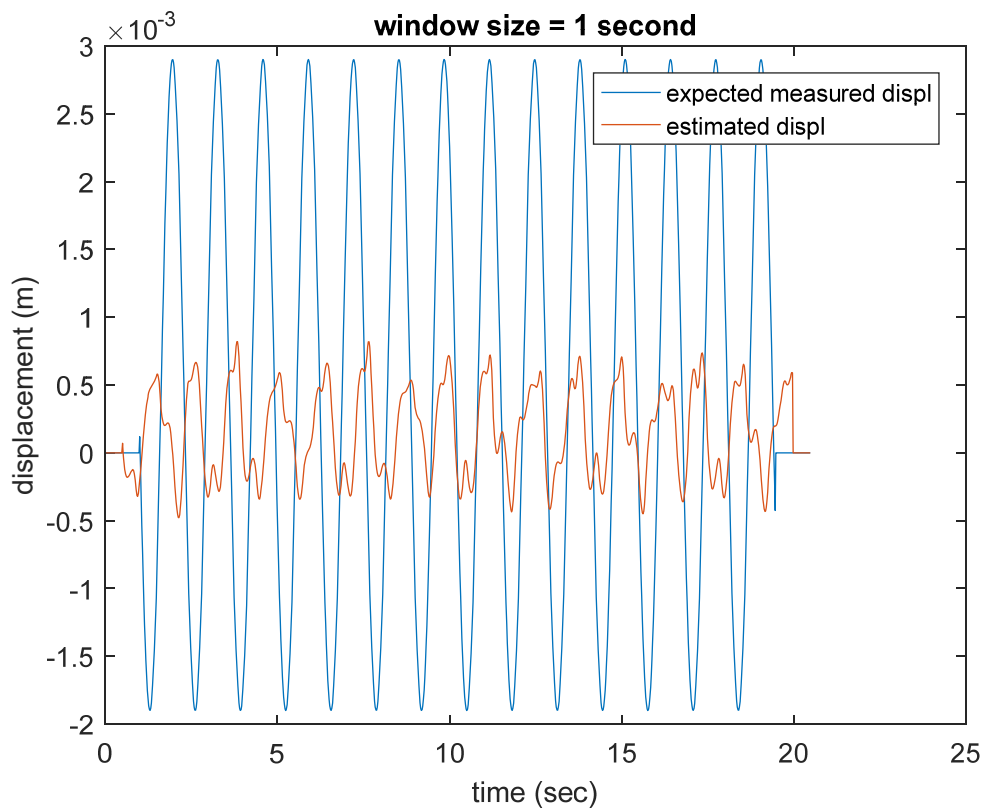


圖 5-39 1Hz 振動頻率下真實位移與重建位移法(有時窗，overlapped)得到移之關係圖(時窗有 20 個，時窗大小 1 秒)

最後將以上試驗過的加速度轉換成位移的演算方法可以發現到，梯形積分法、重建位移法(無視窗)兩種方法與真實位移相比，是差異性相當大的，而加入時窗後，與真實位移的相似度就提高，特別是時窗有相互覆蓋，重建位移法的結果就更加理想了。與當初上一節的數值分析結果不太一樣，上一節反而梯形積分法與重建位移法(無視窗)的結果比較符合 SAP2000 位移。原因同前述為 SAP2000 在做加速度與位移換算的時候是直接利用梯形積分法來計算，所以才會使得在 SAP2000 做懸臂梁模擬的加速度結果，轉換成位移後梯形積分法是最相似的，故這部分以實測來說，還是使用重建位移法且加入時窗的計算會比較可行的做法，與 Lee et al. (2010)的研究一致。只是與 Lee et al. (2010)的論點不同的是，並非時窗越來越多會讓重建位移計算結果會與真實位移相符，雖然後續的幾次試驗會發現在時窗用 1 個到 10 個的情況，時窗確實會越多而計算結果越跟真實位移接近，但算到 20 個時窗，即計算結果與真實位移不相近。故本試驗結果認為會有個最理想、最接近真實位移結果的時窗數量去計算重建位移，而非越多時窗越接近真實位移。

第三節 最佳化參數值之數值分析

前兩部份的試驗了 SAP2000 軟體的數值分析以及室內縮尺度試驗，來驗證重建位移的演算方法是否可與真實位移相近。而演算法的參數 λ 值係直接取自 Lee et al. (2010) 的文章，該公式中， λ 值與時窗大小有關係。而本階段之數值分析將使用粒子群演算法(PSO)最佳化 λ 值。

$$\lambda = 46.81 \times N_d^{-1.95} \quad (10)$$

本研究共執行兩次最佳化(PSO1 與 PSO2)，並根據 PSO2，本計畫自提公式且以兩種不同的地震歷時驗證，兩個地震的振譜如圖 5-40 所示，由圖可知，兩個地震的頻率內涵並不相同，希望以不同振動訊號進行更完整的驗證。PSO1 視 λ 為設計變數，而 PSO2 則視 λ 值為定值(8×10^{-8})。視窗大小受到水保設施振動頻率、取樣頻率與輸入訊號的影響。PSO1、PSO2 與 Lee et al. (2010) 三種演算的結果如表 5-3 到表 5-5 所示，其取樣頻率分別為 50Hz、100Hz 與 200Hz，而設定的水保設施基本震動頻率為 1Hz 到 10Hz。根據上述條件，可以發現無論是 PSO1 或 PSO，其 RMS 均比 Lee et al. (2010) 小，表示本計畫藉由最佳化確可搜尋到誤差更低的位移重建方式。圖 5-41 到圖 5-44 為當水保設施基本振動頻率為 4Hz 與 8Hz 下 PSO1 與 PSO2 重建位移(橘色)與真實位移(藍色)之比較，重建與真實位移十分吻合，故使用粒子群演算法確實有助於得到與真實位移的相近的位移。若進一步比較 PSO1 與 PSO2 的準確度，可以發現 PSO1 的 RMS 比 PSO2 小，雖然如此，PSO2 的視窗大小具有明顯的趨勢，藉由其結果建立公式的機會較高，如圖 5-45 到圖 5-47 所示。根據上述，本計畫根據 PSO2 的結果，利用迴歸分析提出視窗大小的公式如式 11 所示，其中， λ 值為一常數(8×10^{-8})。

$$N_d = \frac{1}{\Delta t} f_T^{-1.433\Delta t^{0.0471}} \quad (11)$$

其中， N_d 為視窗大小， Δt 為取樣時間間隔， f_t 為水保設施基本振動頻率。

表 5-3 不同演算法下 λ 值比較(時間間隔=0.02)

Interval Time	0.02								
Target frequency	PSO 1		PSO 2		Author		RMS from PSO 1	RMS from PSO 2	RMS Author
	N_d	λ	N_d	λ	N_d	λ			
1	95	0.005134	47	8.00E-08	150	2.67E-03	0.003896346	0.004174335	0.006890833
1.2	249	0.007925	37	8.00E-08	125	3.81E-03	0.00374907	0.003670724	0.005609722
1.4	511	0.011331	33	8.00E-08	107	5.16E-03	0.003179673	0.002939075	0.005069261
1.6	517	0.015753	27	8.00E-08	94	6.65E-03	0.002333968	0.002284302	0.00429843
1.8	239	0.019658	25	8.00E-08	83	8.47E-03	0.002035609	0.002080093	0.003737187
2	43	0.021029	21	8.00E-08	75	1.03E-02	0.001195922	0.001659135	0.003265947
2.2	475	0.029629	19	8.00E-08	68	1.25E-02	0.001352331	0.001536485	0.002811326
2.4	37	0.031795	17	8.00E-08	62	1.50E-02	0.000724727	0.0010604	0.002449311
2.6	65	0.043515	15	8.00E-08	58	1.70E-02	0.000832499	0.000960356	0.002216676
2.8	31	0.0408	15	8.00E-08	54	1.96E-02	0.000489519	0.000793724	0.001971061
3	29	0.04595	13	8.00E-08	50	2.28E-02	0.000444206	0.000685782	0.001722828
3.2	27	0.053721	13	8.00E-08	47	2.57E-02	0.000405694	0.000674166	0.001538395
9	5	0.41648	3	8.00E-08	17	1.87E-01	2.84E-05	8.46E-05	1.92E-04
9.2	5	0.438748	3	8.00E-08	16	2.10E-01	2.49E-05	7.47E-05	1.70E-04
9.4	5	0.459542	3	8.00E-08	16	2.10E-01	2.23E-05	6.62E-05	1.73E-04
9.6	5	0.485002	3	8.00E-08	16	2.10E-01	2.05E-05	5.88E-05	1.75E-04
9.8	5	0.50751	3	8.00E-08	15	2.38E-01	1.91E-05	5.23E-05	1.42E-04
10	5	0.520639	3	8.00E-08	15	2.38E-01	1.81E-05	4.64E-05	1.44E-04
Average RMS							0.000567333	0.000656677	0.001247122
Max RMS							0.003896346	0.004174335	0.006890833
Min RMS							1.80826E-05	4.64378E-05	0.000141867

表 5-4 不同演算法下 λ 值比較(時間間隔=0.01)

Interval Time	0.01								
Target frequency	PSO 1		PSO 2		Author		RMS from PSO 1	RMS from PSO 2	RMS Author
	N_d	λ	N_d	λ	N_d	λ			
1	93	8.00E-08	93	8.00E-08	300	6.92E-04	0.003988563	0.003988563	0.006770457
1.2	77	8.00E-08	77	8.00E-08	250	9.87E-04	0.00329277	0.00329277	0.005512807
1.4	67	8.00E-08	67	8.00E-08	214	1.34E-03	0.002654227	0.002654227	0.005007339
1.6	115	0.003261	57	8.00E-08	188	1.72E-03	0.001782679	0.002035172	0.004242507
1.8	100	0.003889	51	8.00E-08	167	2.17E-03	0.001527552	0.001820522	0.003715045
2	94	0.004609	45	8.00E-08	150	2.67E-03	0.001235527	0.001423093	0.003204909
2.2	85	0.006005	41	8.00E-08	136	3.24E-03	0.000995412	0.001309097	0.002759043
2.4	77	0.007137	37	8.00E-08	125	3.81E-03	0.000718396	0.000943854	0.002422299
2.6	374	0.011027	33	8.00E-08	115	4.49E-03	0.000840902	0.000785755	0.002114004
2.8	502	0.012659	31	8.00E-08	107	5.16E-03	0.000706766	0.000664588	0.001865678
3	61	0.011995	27	8.00E-08	100	5.89E-03	0.000436133	0.000598457	0.001654736
3.2	410	0.016256	27	8.00E-08	94	6.65E-03	0.000595004	0.000560867	0.00146848
9	17	0.102629	7	8.00E-08	33	5.12E-02	2.87E-05	6.11E-05	0.000157943
9.2	15	0.115794	7	8.00E-08	33	5.12E-02	2.69E-05	5.39E-05	0.000160957
9.4	15	0.123555	7	8.00E-08	32	5.44E-02	2.58E-05	4.86E-05	0.000150892
9.6	15	0.123515	7	8.00E-08	31	5.78E-02	2.40E-05	4.51E-05	0.000137175
9.8	15	0.124004	7	8.00E-08	31	5.78E-02	2.37E-05	4.26E-05	0.000139664
10	15	0.131026	7	8.00E-08	30	6.17E-02	2.21E-05	4.06E-05	0.000130083
Average RMS							0.000493533	0.000565491	0.001202931
Max RMS							0.003988563	0.003988563	0.006770457
Min RMS							2.20995E-05	4.06192E-05	0.000130083

表 5-5 不同演算法下 λ 值比較(時間間隔=0.005)

Interval Time	0.005								
Target frequency	PSO 1		PSO 2		Author		RMS from PSO 1	RMS from PSO 2	RMS from Author
	N_d	λ	N_d	λ	N_d	λ			
1	187	8.00E-08	187	8.00E-08	600	1.79E-04	0.003843501	0.003843501	0.006681562
1.2	159	8.00E-08	159	8.00E-08	500	2.55E-04	0.003153338	0.003153338	0.005434226
1.4	271	0.000638	135	8.00E-08	429	3.44E-04	0.002424345	0.002526188	0.004931764
1.6	115	8.00E-08	115	8.00E-08	375	4.48E-04	0.001935053	0.001935053	0.004143461
1.8	103	8.00E-08	103	8.00E-08	333	5.64E-04	0.001722011	0.001722011	0.00361553
2	169	0.001177	91	8.00E-08	300	6.92E-04	0.001252143	0.001338621	0.003131288
2.2	157	0.001728	83	8.00E-08	273	8.31E-04	0.001169931	0.001197543	0.002713367
2.4	152	0.001769	73	8.00E-08	250	9.87E-04	0.000717672	0.000863435	0.002363597
2.6	143	0.002284	67	8.00E-08	231	1.15E-03	0.000594549	0.000731321	0.002081507
2.8	126	0.002566	61	8.00E-08	214	1.34E-03	0.00048538	0.000599901	0.001818934
3	129	0.003183	57	8.00E-08	200	1.53E-03	0.000460008	0.000543369	0.00161461
3.2	55	0.000326	53	8.00E-08	188	1.72E-03	0.000511355	0.000516115	0.001432304
9	37	0.026139	17	8.00E-08	67	1.29E-02	2.60E-05	4.35E-05	0.000152225
9.2	36	0.027433	17	8.00E-08	65	1.37E-02	2.54E-05	4.28E-05	0.000140634
9.4	36	0.029758	15	8.00E-08	64	1.41E-02	2.74E-05	4.20E-05	0.000136916
9.6	33	0.029948	15	8.00E-08	63	1.45E-02	2.29E-05	3.84E-05	0.000132077
9.8	32	0.03151	15	8.00E-08	61	1.55E-02	2.22E-05	3.59E-05	0.00012086
10	31	0.032502	15	8.00E-08	60	1.60E-02	2.14E-05	3.42E-05	0.000117127
Average RMS							0.000481066	0.000523278	0.001171728
Max RMS							0.003843501	0.003843501	0.006681562
Min RMS							2.13632E-05	3.41962E-05	0.000117127

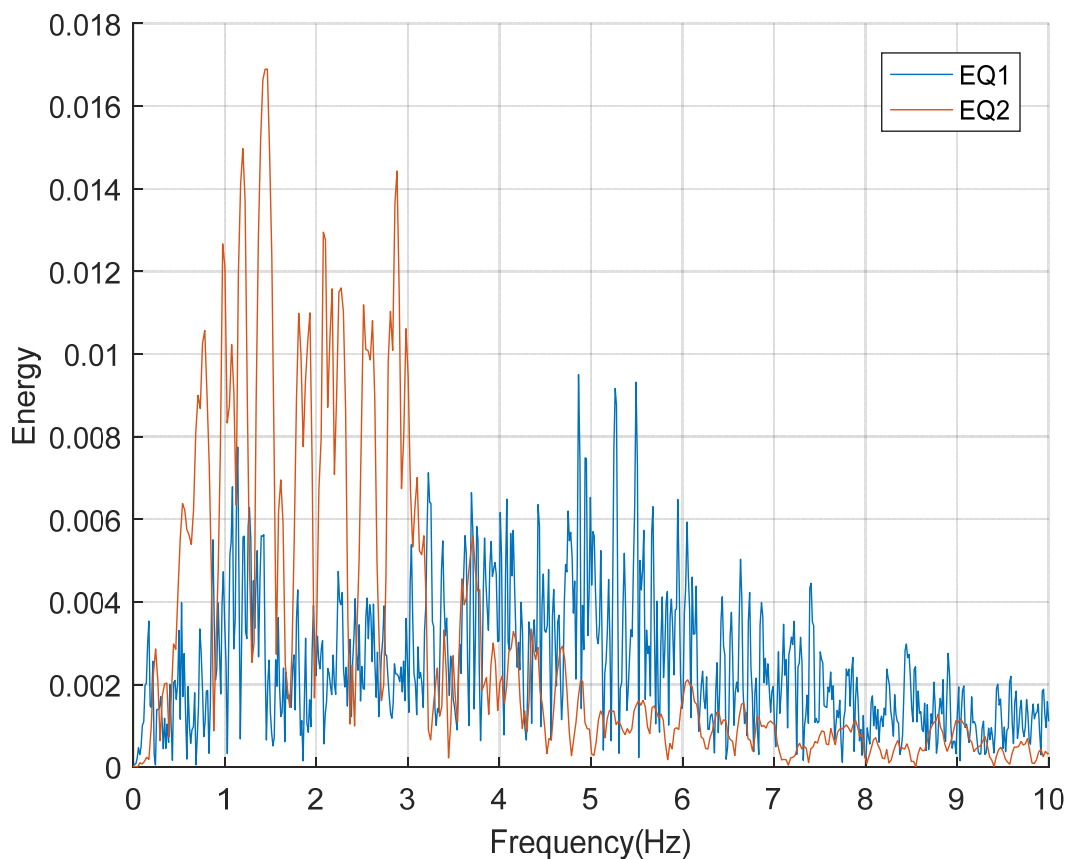


圖 5-40 PSO 兩種演算法之振動能量與頻率

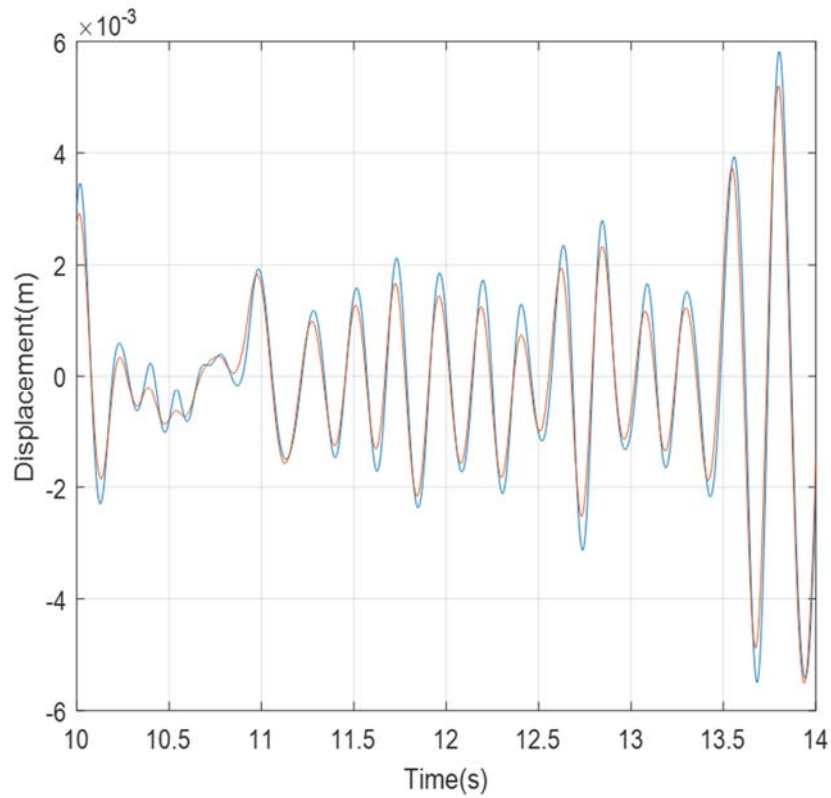


圖 5-41 結構的振動頻率 4Hz 時，真實位移與重建位移(PSO1)之比較

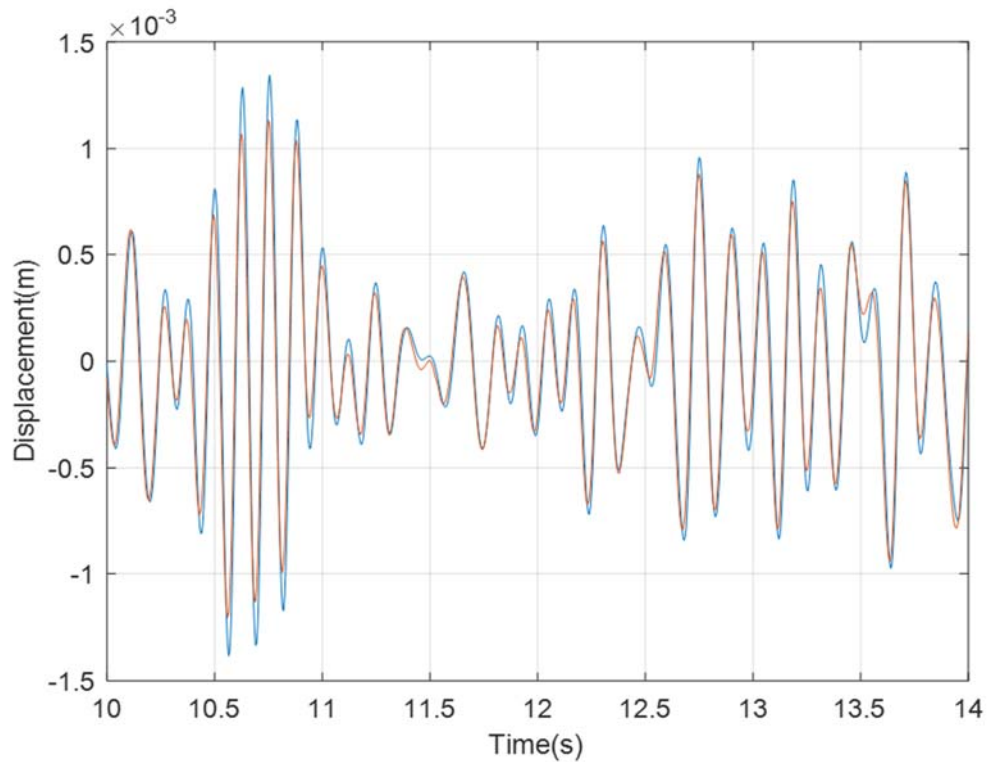


圖 5-42 結構的振動頻率 8Hz 時，真實位移與重建位移(PSO1)之比較

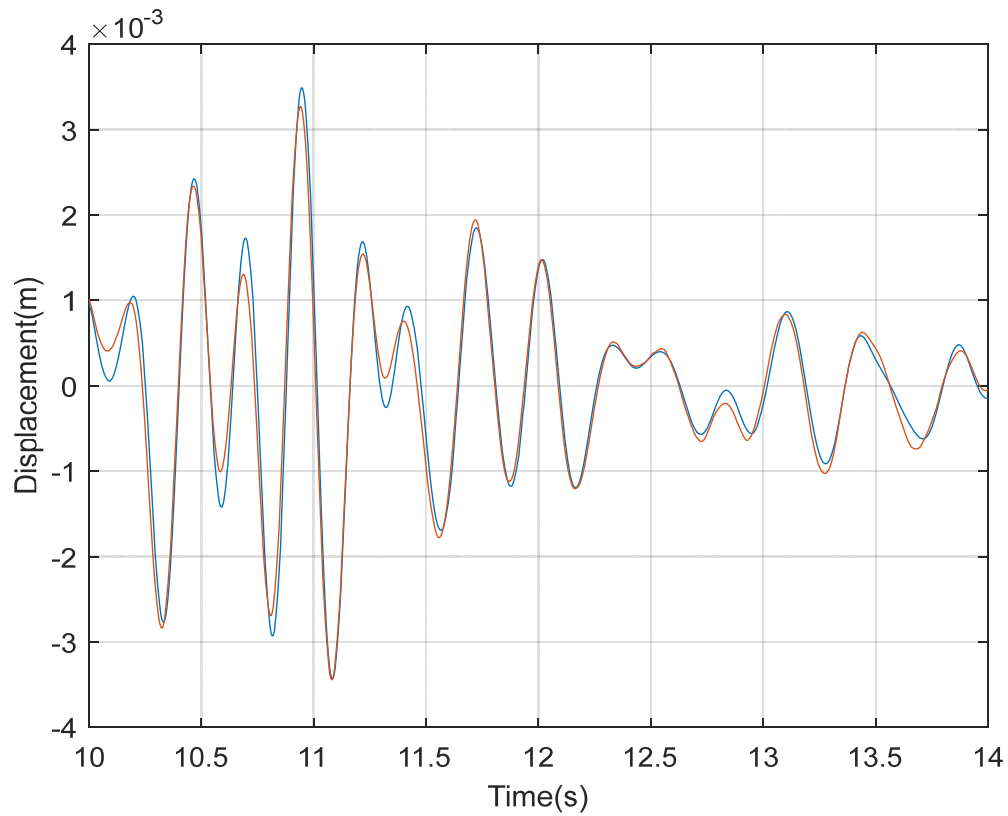


圖 5-43 結構的振動頻率 4Hz 時，真實位移與重建位移(PSO2)之比較

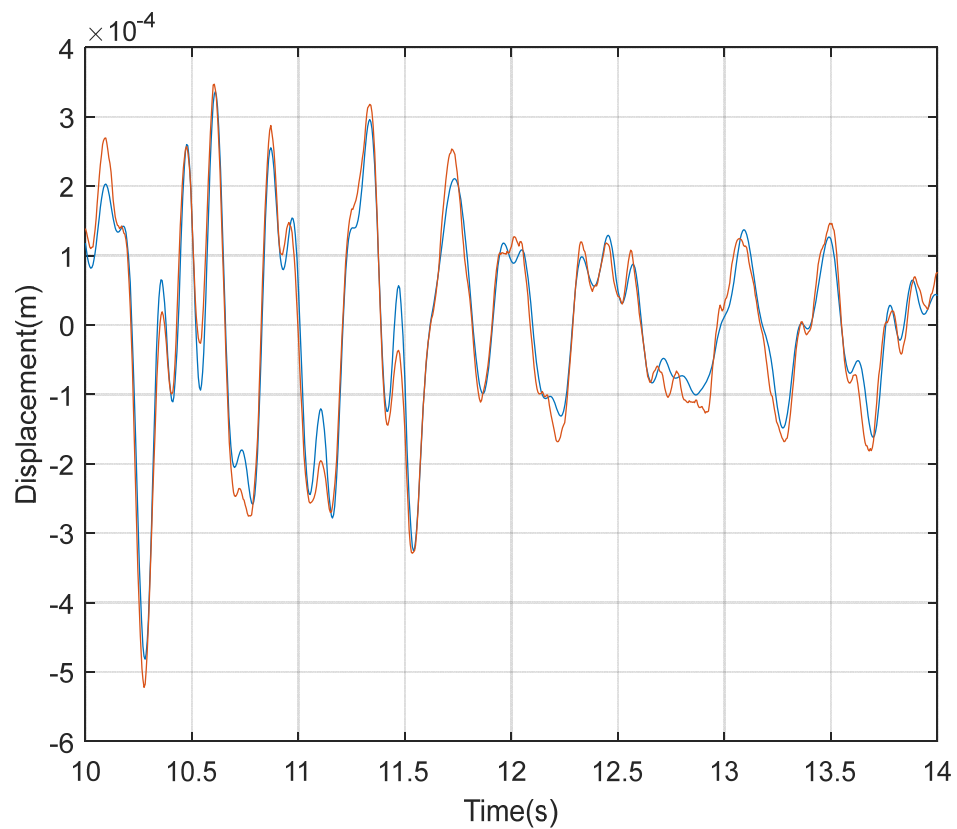


圖 5-44 結構的振動頻率 8Hz 時，真實位移與重建位移(PSO2)之比較

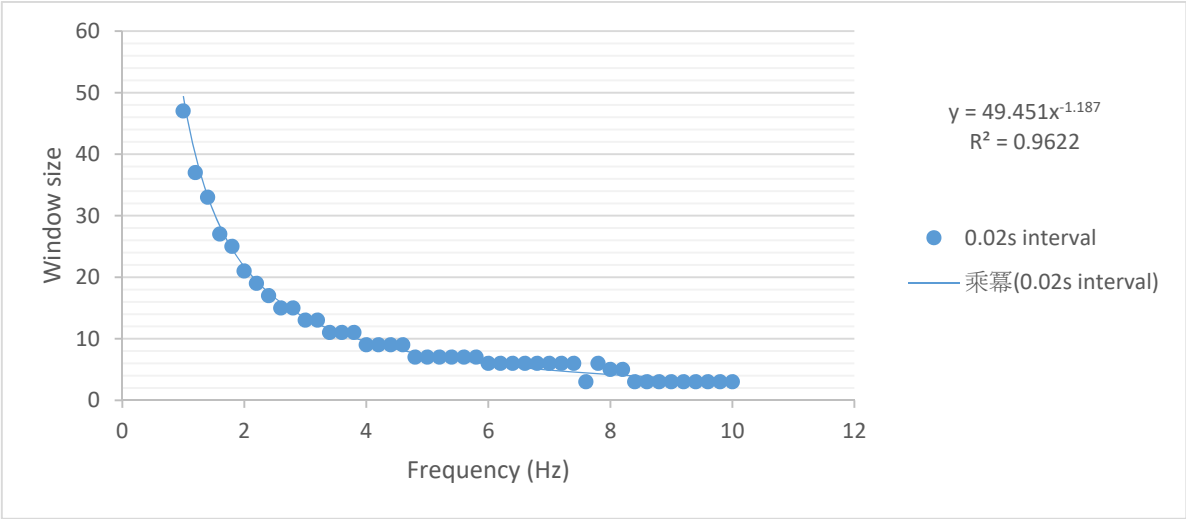


圖 5-45 時間間隔為 0.02 時 PSO2 演算法之時窗與頻率關係圖

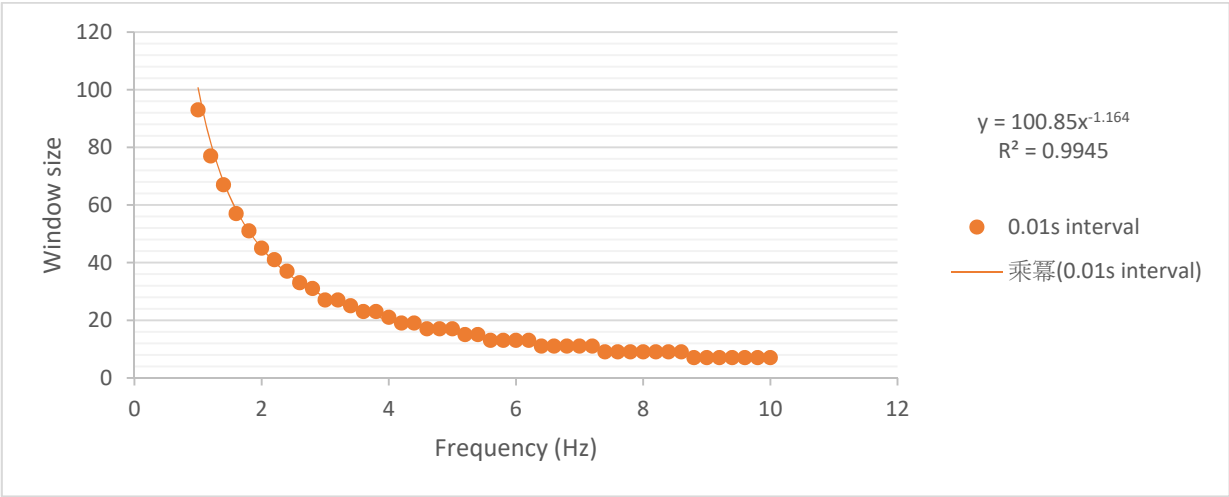


圖 5-46 時間間隔為 0.01 時 PSO2 演算法之時窗與頻率關係圖

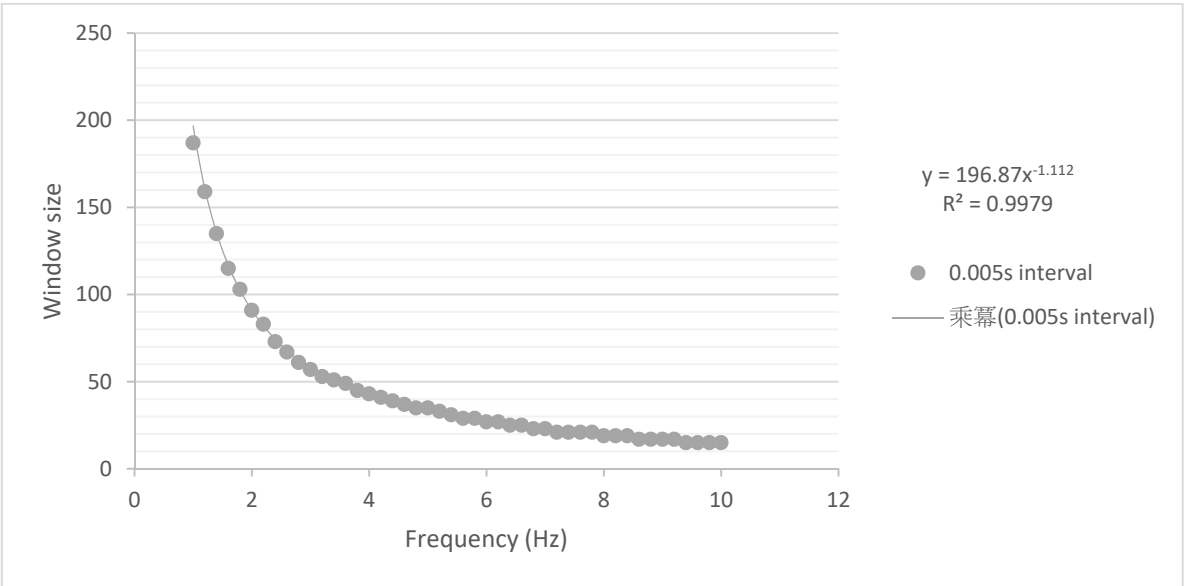


圖 5-47 時間間隔為 0.005 時 PSO2 演算法之時窗與頻率關係圖

第四節 室外試驗

數值分析與室內模擬完成後，則進行第三個部分之試驗，即可開發之 Arduino 板+IoT 模組以及感測器放置於室外進行試驗，如前所述，此次的試驗重點在於測試網路通訊技術是否順利完成，即於室外取得之加速度值是否可以順利回傳至試驗室。於設置模組到某個地區前，可參考圖 3-5 確認試驗地點 4G 網路是否已開通。將加速度計、Arduino 板與 NB-IoT 等機器安置在外面，測得加速度並回傳室內主機伺服器，並推求位移。根據前述，先前介紹的 SIM7000E 之 Arduino 擴展板與 ZigBee，根據本實驗室的測試，可順利的把資料(加速度)回傳，即達到預期的無線通訊效果。

另、於野外進行試驗時，因為不像在室內試驗隨處有電器連接來源，故在野外試驗電力的供應就變成一項需要另外準備的項目。而為了因應長期供電問題，選用較容易隨手可得且低價位的機車電瓶，以提供穩定電壓電流之供電，解決野外無家用電器插座可供電的問題。藉由機車電瓶供應電壓給 Arduino UNO 板使其持續運作如圖 5-48。

機車電瓶規格：

- 一、公稱電壓 Nominal Voltage：12V
- 二、公稱容量 Nominal Capacity：8.6AH / 10HR (25°C)
- 三、重量 Weight(Approx.)：約 3.1Kg
- 四、充電電流 Charge Current：0.9A (25°C)
- 五、長度 Length：150mm
- 六、寬度 Width：87mm
- 七、總高 Overall Height：93mm

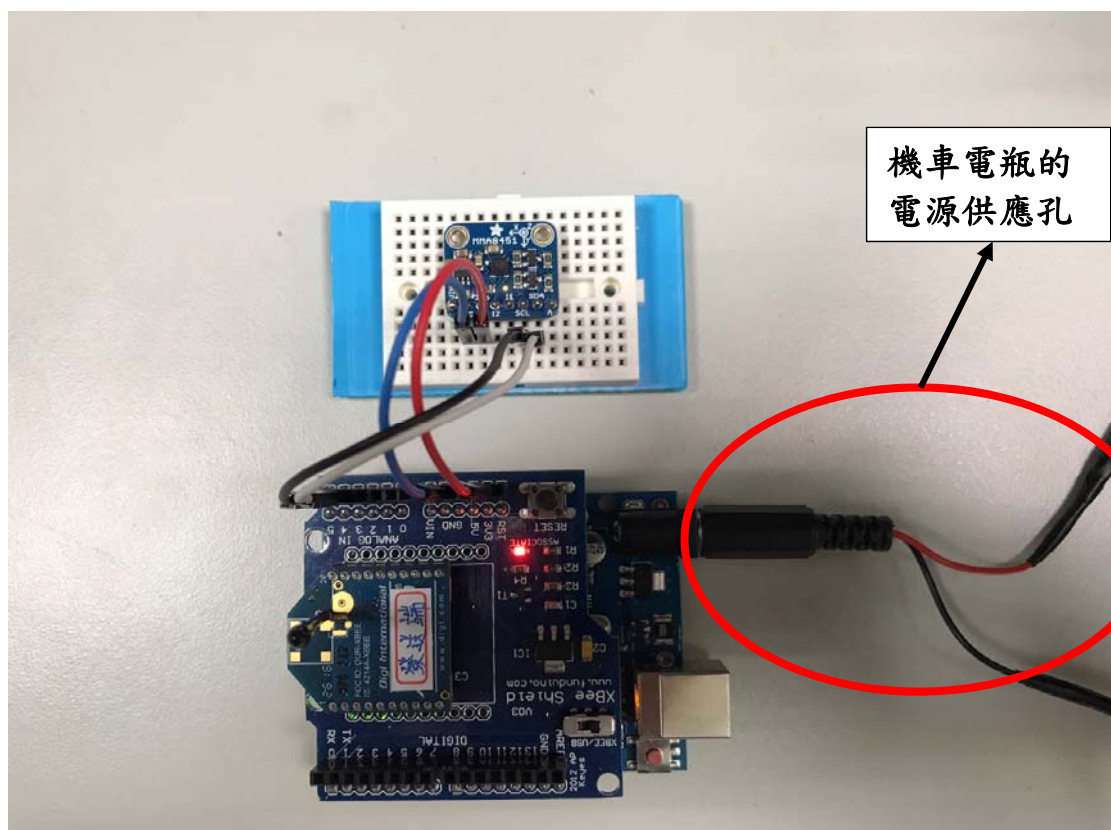


圖 5-48 機車電瓶將電源供應 Arduino UNO 板

第六章 總結

本計畫結合智慧感應器與 NB-IoT 技術提供水保設施的初步安全評估時的參考，關鍵技術包含智慧感應器、位移重建與 NB-IoT 環境的開發。為完成上述目標，首先針對本研究室所收集的感測器、微型處理器與 NB-IoT 的硬體部分進行測試，根據測試結果，加速度計的部分包括 MMA8451Q、MMA7660FC、ADXL335 與微型處理器的部分 Arduino M0 Pro 板、Arduino UNO 板與 NB-IoT 模組的部分 SIM7000E、XBee 均符合本計畫所設定之目標，即可正確地量取加速度值並且能藉由無線傳輸將資料回傳。已完成的工作項目包括智慧感應器的開發(例如 MMA7660FC 與 Arduino M0 Pro)、位移重建演算的建立與程式化以及 TCP、UDP 等通訊協定(NB-IoT 模組與服務器主機間的通訊設定)。根據已建立的技術，可順利量測加速度並將其對應的位移重建，亦可利用無線通訊技術將野外加速度資料回傳遠端的室內主機，可有效減少人力與時間成本。

免控制點位移重建演算法的參數設定首先參照 Lee et al. (2010)的研究，並以 PSO 進行最佳化並提出視窗之修改公式。根據本計畫的目標，計畫執行過程中，設計了三個階段的試驗。其中，第一階段為利用數值模型進行驗證、第二個階段為室內小型實體模型進行驗證、第三階段為最佳化 λ 值再做演算、第四個階段為室外無線通訊的驗證。其中，第一個階段的輸入震動訊號再細分為經過基線修正與否兩種，根據第一個階段數值與第二個階段實體驗證的結果，梯形積分法因受到基線修正與否的影響，並不適用於位移重建，而所提的方法，相對而言，較為穩定。室內試驗結果顯示，具時窗且具有覆蓋設計的演算法結果比較理想。然而，根據結果同時也顯示，時窗越多並不代表重建位移準確，視窗大小受到水保設施振動頻率、取樣頻率與輸入訊號的影響。第三階段使用 PSO 演算法最佳化 λ 值，分析時考慮的取樣頻率為 50Hz、100Hz 與 200Hz，而設定的水保設施基本震動頻率為 1Hz 到 10Hz，本計畫藉由最佳化確可搜尋到誤差更低的位移重建方式，並利用迴歸分析提出視窗大小的公式。第四階段則如前面所提，已順利使用 UDP 技術將加速度資料回傳。

圖 6-1 是本計畫的工作計畫書之甘梯圖，期中報告前，本團隊已完成的項目計有：免控制點演算法之文獻回顧及演算法的選擇、加速度計與微型處理器與物聯網的建置、數值分析試驗。而期末報告完成了 NB- IoT 硬體與 Arduino 之整合、室內驗證與野外實地試驗、免控制點參數最佳化。計劃書中所提之技術均以完成。

期末報告委員們的建議與回覆意見請詳見附錄一。

	二月	三月	四月	五月	六月	七月	八月	九月	十月	十一月
免控制點演算法之文獻回顧										
免控制點演算法之選擇										
免控制點演算法開發模式參數之最佳化										
智慧感測器之加速度計開發與選定										
智慧感測器之微型處理器開發與選定										
物聯網(IoT)之建置與評比										
NB- IoT 與加速度計之整合										
重建位移之驗證之數值分析										
重建位移之驗證之室內縮尺度試驗										
重建位移之驗證之水保設施試驗										

圖 6-1 甘梯圖

參考文獻

1. Alam, A.H.M. Z., Arfah, N., Khan, S. and Islam, M. R., “Design of capacitance to voltage converter for capacitive sensor transducer,” American Journal of Applied Sciences 7(10): 1353, 2010.
2. Analog device, “Small, low power, 3-axis ± 3 g accelerometer-ADXL335,” datasheet, 2009.
3. Arduino, “Getting started with the Arduino M0 Pro,” Arduino Official website, from: <https://www.arduino.cc/en/Guide/ArduinoM0Pro>.
4. Arduino, “Arduino M0 Pro,” Arduino store, from: <https://store.arduino.cc/usa/arduino-m0-pro>.
5. Digi International Inc., “XBee®/XBee-PRO® RF Modules,” Data sheet, 2009.
6. EBU-EPSM, “NB-IoT Service,” Far EasTone, 2017.
7. Embedded- cracking the code to system development, “Network Topology,” Embedded. from: <https://www.embedded.com/>
8. Lee, H. S., Hong, Y. H., Park, H. W., “Design of an FIR filter for the displacement reconstruction using measured acceleration in low-frequency dominant structures,” International Journal for Numerical Methods in Engineering 82(4): 403-434, 2010.
9. MouserEletronics Company, “Products of accelerometers,” from: <https://www.mouser.tw/search/refine.aspx?N=18357019>
10. Moreu, F., Li, J., Jo, H., Scola, R. E. S., Spencer, B. F. and LaFave, J. M., “Reference-free displacements for condition assessment of timber railroad bridges,” Journal of Bridge Engineering 21(2): 04015052, 2015.
11. NXP Semiconductor, “3-Axis Orientation/Motion Detection Sensor,” Data Sheet, 2012.
12. NXP Semiconductor, “MMA8451Q 3-Axis, 14-bit/8-bit Digital Accelerometer,” Data Sheet, 2014.
13. NXP Semiconductor, “I²C-bus specification and user manual,” user manual, 2014.
14. Park, J. W., Sim, S. H., Jung, H. J., “Development of a wireless displacement measurement system using acceleration responses,” Sensors 13(7): 8377-8392, 2013.
15. Péceli, G, “A common structure for recursive discrete transforms,” IEEE Transactions on Circuits and Systems 33(10): 1035-1036, 1986.

16. Ryu, T., Ishizaka, Y., Kinoshita, I., “At-command analyzing method,” Google Patents, 2008.
17. SIMCom Wireless Solutions, “SIM7000E,” eMTC&NB-IoT Module, from: <http://simcomm2m.com/En/module/detail.aspx?id=168>.
18. SIMCom Wireless Solutions, “SIM7000_TE kit_ User Guide_V1.02,” Application Note, 2017.
19. SIMCom Wireless Solutions, “Battery Lifetime Calculation,” Application Note, 2018.
20. SIMCom Wireless Solutions, “SIMCom Wireless Module Roadmap (for global),” Application Note, 2018.
21. SIMCom Wireless Solutions, “SIMCOM_EVB Kit_User Guide_V1.00,” Application Note, 2018.
22. Vaidyanathan, P., “Passive cascaded-lattice structures for low-sensitivity FIR filter design, with applications to filter banks,” IEEE Transactions on Circuits and Systems 33(11): 1045-1064, 1986.
23. 田中尚文, 園田茂, 村岡慶裕, 富田豊, 千野直一: 小型加速度計による歩行分析の再現性および妥当性の検討, リハビリテーション医学, 33(8): 549-553, 1996.
24. 吳建德, 碩士論文, 部分可靠傳輸通訊協定於多媒體網路之研究, 東海大學資訊工程學系, 2010。
25. 李忠憲, 課程講義, TCP/IP概論, 國立成功大學電機工程學系, 2003。
26. 林信良, “用mBlock玩Arduino - Starting from Scratch”, 臺北市: 碁峰出版社, 2015。
27. 林科里, 碩士論文, 山區公路邊坡及擋土結構物災害搶修管理之研究, 國立中央大學土木工程學系, 2004。
28. 果壳实验室, 2016, 來源: <https://read01.com/zh-tw/N7ajn.html#.W2LFOdUzbIV>。芯訊通無線科技(上海)有限公司, “SIM7000系列_TCPIP_應用文檔_V1.01”, 產品應用介紹, 2017。
29. 盛日, “中国 RoHS 认证现状及欧盟合规评定模式简介”, 家电科技(10):12-

- 13，2017。
30. 彭敬鈞，碩士論文，三軸加速規於橋梁結構安全監測系統之研究，國立臺灣科技大學營建工程系，2017。
 31. 黃國鴻，開放式課程講義，TCP與UDP，國立成功大學數位學習設計與管理學系，2018。
 32. 新井智之，柴喜崇，渡辺修一郎，柴田博：10m歩行における歩行周期変動と運動機能，転倒との関連：小型加速度計を用いた測定. 理学療法学, 38(3): 165-172, 2011。
 33. 楊凱傑，“水土保持工程設施耐久性檢測評估紀錄：須美基溪支流土石災害防治工程---跌水工”，水土保持工程設施示範案例維護知識庫，2012。
 34. 廖國偉、鄭明淵，“創新快速橋梁振動頻率量測方法與人工智慧安全評估模式之研發”，交通部研究計畫，2016。
 35. 趙英傑，“超圖解 Arduino 互動設計入門(第3版)”，臺北市：旗標出版社，2016。
 36. 遠傳IoT團隊，“NB-IoT全球快速起飛，物聯網應用全面展開”，遠傳電信，2018。
 37. 潘乃欣、蔡佐良、王劍能、陳奎言、陳俊銘、呂瑞騰…黃鈺珊，“水土保持設施耐久性分析與延壽策略研擬計畫案”，行政院農業委員會水土保持局研究計畫，2012。
 38. 蔣鎧宇，碩士論文，多晶矽電容式微加速規的研發，國立交通大學機械工程學系，2009。
 39. 賴聖恩，碩士論文，整合數位加速規及卡式濾波器進行結構振動響應估測，成功大學航空太空工程學系，2010。
 40. 謝秉蓁，碩士論文，水土保持工程構造物體檢模式之建立，逢甲大學水利工程與資源保育學系，2013。
 41. 顏上堯，李旺蒼，施佑林，“路徑基礎類粒子群最佳化演算法於求解含凹形節線成本最小成本轉運問題之研究”，運輸計劃季刊36(3)：393-423，2007。
 42. 蘇德馨，專題研究精簡報告，有限元素分析軟體應用於結構教學之研究，高苑

科技大學土木工程系，2009。

附錄

附錄一、期末審查會議紀錄暨回覆辦理情形

項次	審查意見	回覆辦理情形
報告內容審查意見：		
一	免控制點位之物件有無電力及防水等考量設計。	目前研究的重點在智慧感測計、位移重建與 NB-IoT 無線網路的演算法開發，而電力供應上，相關資訊如下：若使用機車電瓶作為電力供應，其壽命約為 2~6 年，而從期末報告 3-33~3-36 中可以了解到 NB-IoT 可持續用 0.5~3 年，故樂觀估計可使用 3 年。除此之外，太陽能電池亦可為選項之一。另、安裝模組與感測器時可使用 3D 列印技術客製化裝置容器，以輔助及裝置容器於室外之安裝。

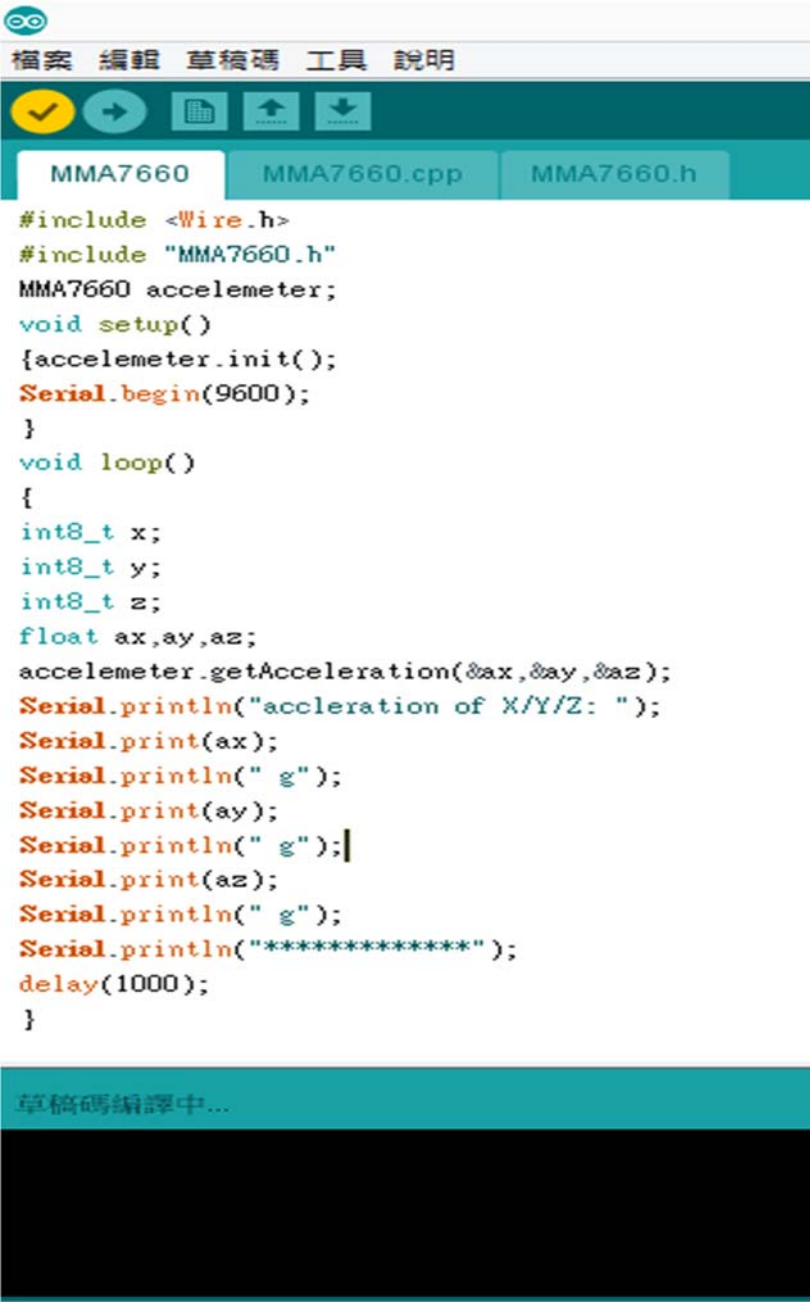
項次	審查意見	回覆辦理情形
二	NB-IoT技術在資料傳輸時遺漏封包的情形如何?遺漏資料對後續演算的影響為何?	如期末報告的表3-9所示，無線通訊協定傳輸試驗過TCP與UDP，UDP有即時性高以及傳輸資料量大的優點，但對於資料傳輸的可靠性與穩定性相對就差了些；相反TCP傳輸資料慢且資料量也較少，但可透過序列確認以及封包重送機制，提供了可靠性和連接導向的服務。故將來如在UDP使用上遺漏封包問題嚴重，將可著重使用TCP。而若有遺漏資料情形發生，則無法進行位移重建，也就無法取得水保設施的變位。未來若有機會，將嘗試以TCP協定傳輸資料，以減低遺漏封包的問題。

項次	審查意見	回覆辦理情形
三	本計畫在位移監測成果甚多，建議在位移重建水保設施安全評估之間之相關性再強化說明。	重建位移後則可以了解水保設施的變位的情形，來評估其設施的安全程度，具體而言，以防砂壩為例，分析時至少應針對傾倒、滑動與內部應力進行檢核，若以內部應力而言，假設壩體底部為應力最大發生處(且為撓曲應力控制)，則可依據壩體斷面、高度、降伏彎矩反推可容許位移值作為壩體是否安全的門檻值。
四	水土保持工程之特性建議於設置IoT設備時可能需要考量，電力來源與更換、IoT設備傳輸之影響及戶外設備容易破壞之影響。	若使用機車電瓶作為電力供應，其壽命約為 2~6 年，而從期末報告 3-33~3-36 頁中可以了解到 NB-IoT 可持續用 0.5~3 年，故樂觀估計可使用 3 年。除此之外，太陽能電池亦可為選項之一。另、安裝模組與感測器時可使用 3D 列印技術客製化裝置容器，以輔助及裝置

項次	審查意見	回覆辦理情形
		容器於室外之安裝。
五	以測量加速度重建位移，對於潛勢崩塌區是否可評估其安全度，並做為土砂工程佈設之參據。	感謝委員意見，目前本研究室亦認為位移重建技術應可利用於潛勢崩塌區安全度的判定，例如速度倒數為崩塌的重要指標之一，而速度與位移如同加速度與位移，存有一定的關係，本研究室目前正規劃相關的研究案進行進一步的分析。
六	相對於位移量，斷裂、沖刷、局部磨損應是更重要破壞型態，後續研究應針對此部分深入研究，此部分破壞型態可否用此技術進行評估。	感謝委員意見，沖刷與局部磨損確實為重要破壞型態，而且，沖刷與局部磨損(尤其是後者)在目視檢測時，可能無法成功減出，亦即可視為隱形的損壞 (invisible damage)，這類損害雖隱形但會影響設施的振動頻率，因此獲可藉由振動頻率的變化量得知此一狀

項次	審查意見	回覆辦理情形
		態，本研究室目前已規劃相關研究案進行研究。
七	NB-IoT 可否運用在防災訊息通知。	感謝委員意見，NB-IoT 應有機會運用於防災訊息通知，例如若位移已經達到某項閾值，即可發布防災訊息通知。如同第 3 個意見，以防砂壩為例，分析時至少應針對傾倒、滑動與內部應力進行檢核，若以內部應力而言，假設壩體底部為應力最大發生處(且為撓曲應力控制)，則可依據壩體斷面、高度、降伏彎矩反推可容許位移值作為壩體是否安全的門檻值。 至於訊息的發布與相關設計，則不在本計畫規劃的範疇內。

附錄二：MMA7660 加速度計程式碼



```
#include <Wire.h>
#include "MMA7660.h"
MMA7660 accelemer;
void setup()
{
  accelemer.init();
  Serial.begin(9600);
}
void loop()
{
  int8_t x;
  int8_t y;
  int8_t z;
  float ax,ay,az;
  accelemer.getAcceleration(&ax,&ay,&az);
  Serial.println("accleration of X/Y/Z: ");
  Serial.print(ax);
  Serial.println(" g");
  Serial.print(ay);
  Serial.println(" g");
  Serial.print(az);
  Serial.println(" g");
  Serial.println("*****");
  delay(1000);
}
```

草稿碼編譯中...

附錄三：MMA7660 加速度計 cpp 檔



```
#include <Wire.h>
#include "MMA7660.h"
void MMA7660::write(uint8_t _register, uint8_t _data)
{
  Wire.begin();
  Wire.beginTransmission(MMA7660_ADDR);
  Wire.write(_register);
  Wire.write(_data);
  Wire.endTransmission();
}
uint8_t MMA7660::read(uint8_t _register)
{
  uint8_t data_read;

  Wire.begin();

  Wire.beginTransmission(MMA7660_ADDR);

  Wire.write(_register);

  Wire.endTransmission();
  Wire.beginTransmission(MMA7660_ADDR);
```

```
Wire.requestFrom(MMA7660_ADDR,1);

while(Wire.available())

{
data_read = Wire.read();
}

Wire.endTransmission();
return data_read;
}

void MMA7660::init()
{

setMode(MMA7660_STAND_BY);

setSampleRate(AUTO_SLEEP_32);

setMode(MMA7660_ACTIVE);

}
```

```
void MMA7660::setMode(uint8_t mode)

{
    write(MMA7660_MODE,mode);
}

void MMA7660::setSampleRate(uint8_t rate)
{
    write(MMA7660_SR,rate);
}

/*Function: Get the contents of the registers in the MMA7660*/

/*          so as to calculate the acceleration.          */

void MMA7660::getXYZ(int8_t *x,int8_t *y,int8_t *z)

{
    unsigned char val[3];
    int count = 0;

    val[0] = val[1] = val[2] = 64;
```

```
while(Wire.available() > 0)

Wire.read();

Wire.requestFrom(MMA7660_ADDR,3);

while(Wire.available())

{
  if(count < 3)
  {
    while ( val[count] > 63 ) // reload the damn thing it is bad
    {
      val[count] = Wire.read();
    }
  }
  count++;
}

*x = ((char)(val[0]<<2))/4;
*y = ((char)(val[1]<<2))/4;
*z = ((char)(val[2]<<2))/4;
```

```
void MMA7660::getAcceleration(float *ax, float *ay, float *az)
{
    int8_t x, y, z;
    getXYZ(&x, &y, &z);
    *ax = x/21.00;
    *ay = y/21.00;
    *az = z/21.00;

}
```

編譯完畢。

草稿碼使用了 20,816 bytes (7%) 的程式儲存空間。上限為 262,144 bytes

附錄四：MMA7660 加速度計 h 檔



```
#ifndef __MMA7660_H__
#define __MMA7660_H__

#define MMA7660_ADDR 0x4c
#define MMA7660_X 0x00
#define MMA7660_Y 0x01
#define MMA7660_Z 0x02
#define MMA7660_TILT 0x03
#define MMA7660_SRST 0x04
#define MMA7660_SPCNT 0x05
#define MMA7660_INTSU 0x06
#define MMA7660_MODE 0x07
#define MMA7660_STAND_BY 0x00
#define MMA7660_ACTIVE 0x01
#define MMA7660_SR 0x08 //sample rate register
#define AUTO_SLEEP_120 0x00 //120 sample per second
#define AUTO_SLEEP_64 0x01
#define AUTO_SLEEP_32 0x02
#define AUTO_SLEEP_16 0x03
#define AUTO_SLEEP_8 0x04
#define AUTO_SLEEP_4 0x05
#define AUTO_SLEEP_2 0x06
#define AUTO_SLEEP_1 0x07
#define MMA7660_PDET 0x09
#define MMA7660_PD 0x0A
```

```
class MMA7660
{
private:
void write(uint8_t _register, uint8_t _data);

uint8_t read(uint8_t _register);

public:
void init();
void setMode(uint8_t mode);
void setSampleRate(uint8_t rate);
void getXYZ(int8_t *x,int8_t *y,int8_t *z);
void getAcceleration(float *ax,float *ay,float *az);
};

#endif
```

編譯完畢。

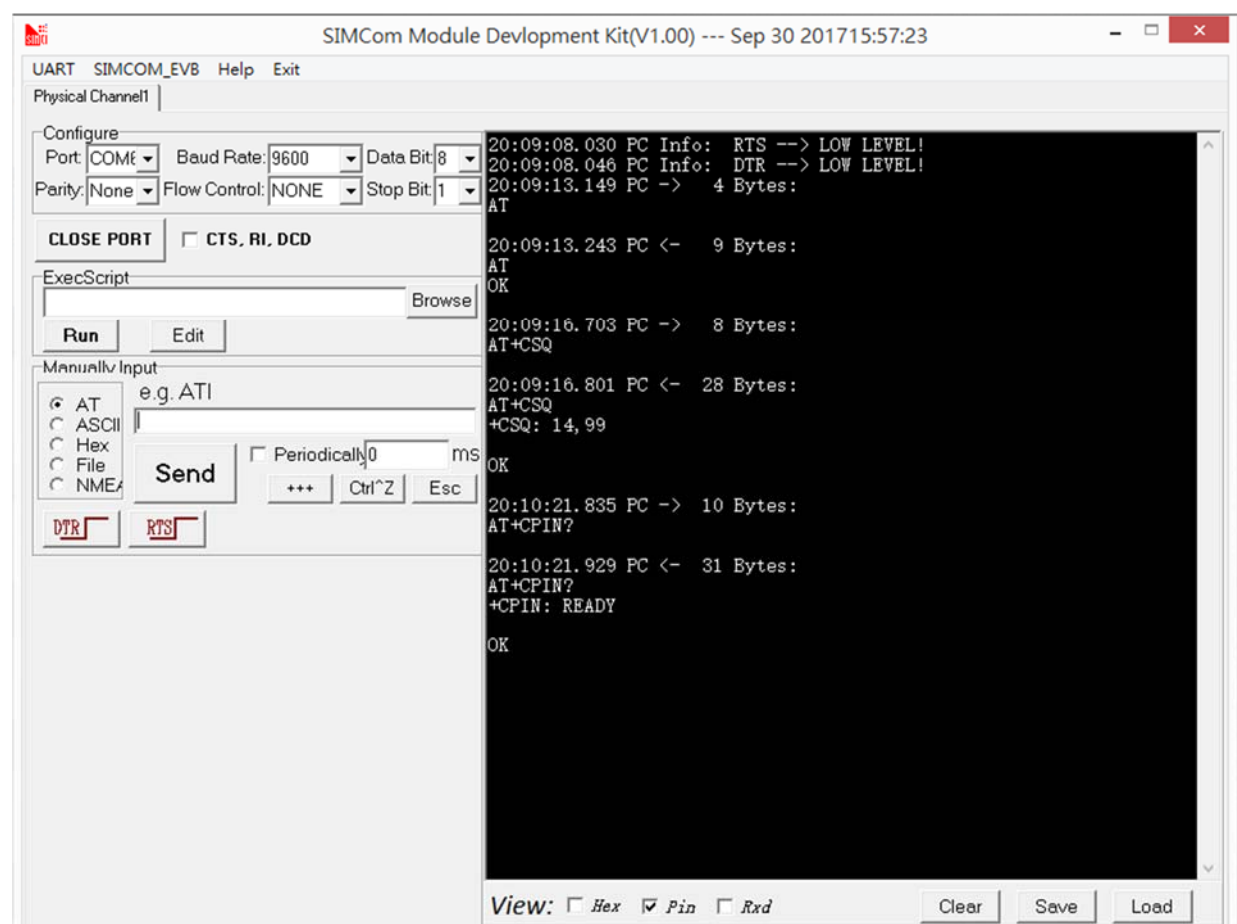
草稿碼使用了 20,816 bytes (7%) 的程式儲存空間。上限為 2

附錄五：將 SD 卡功能用在 Arduino 上所需指令

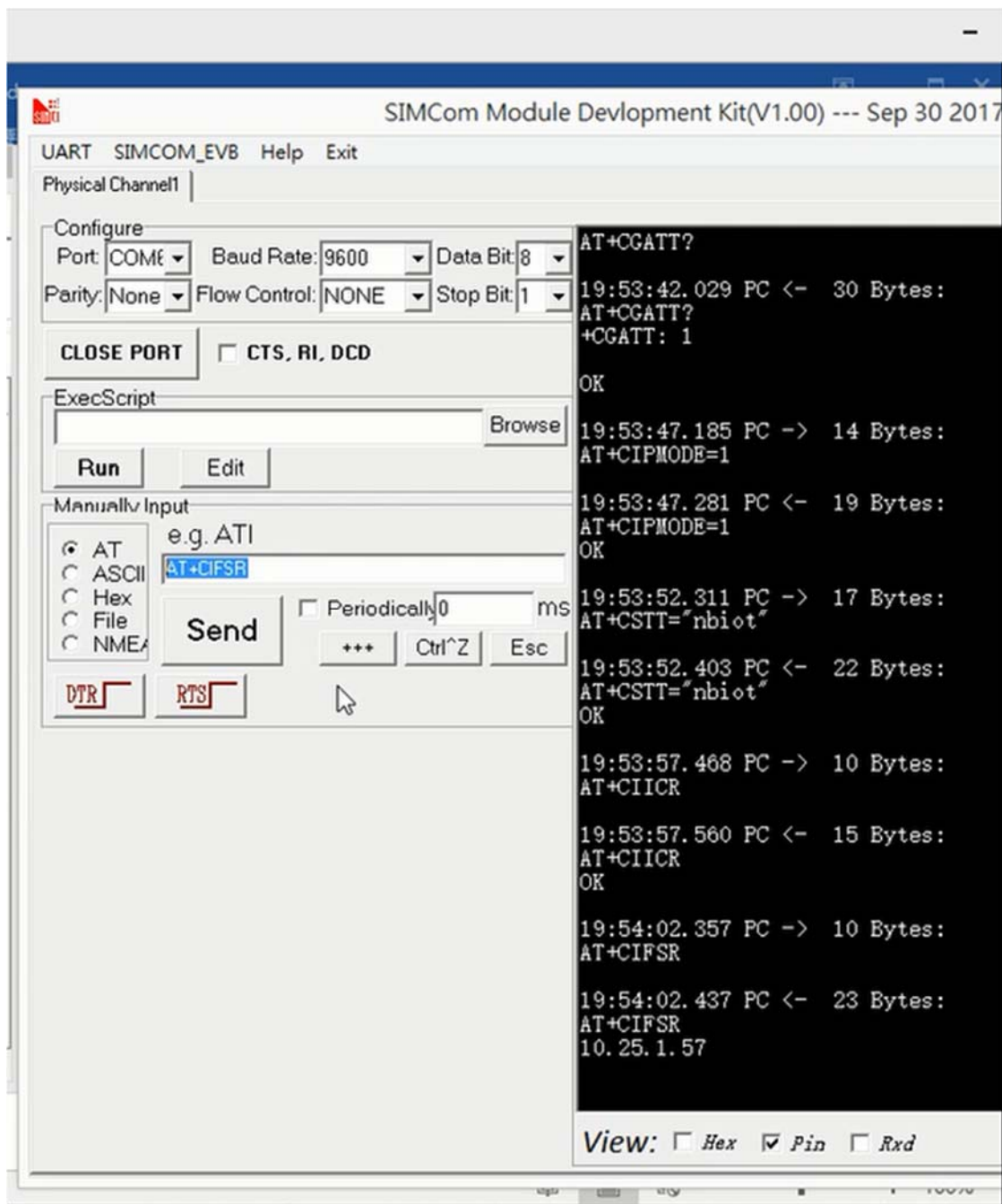
功能	說明
begin()	初始化 SD 函式庫和 SD 卡
exists()	檢查指定的檔案或是資料夾是否存在 SD 卡
mkdir()	在 SD 卡上建立資料夾
open()	開啟 SD 卡上的檔案
remove()	刪除 SD 卡裡的檔案
rmdir()	刪除 SD 卡的資料夾，該資料夾必須是空的
available()	檢查檔案是否還有可讀取資料
close()	關閉檔案並且確保對檔案的修改儲存置 SD 卡
flush()	把對檔案的修改實際寫入 SD 卡，這個動作在關閉檔案時會自動執行

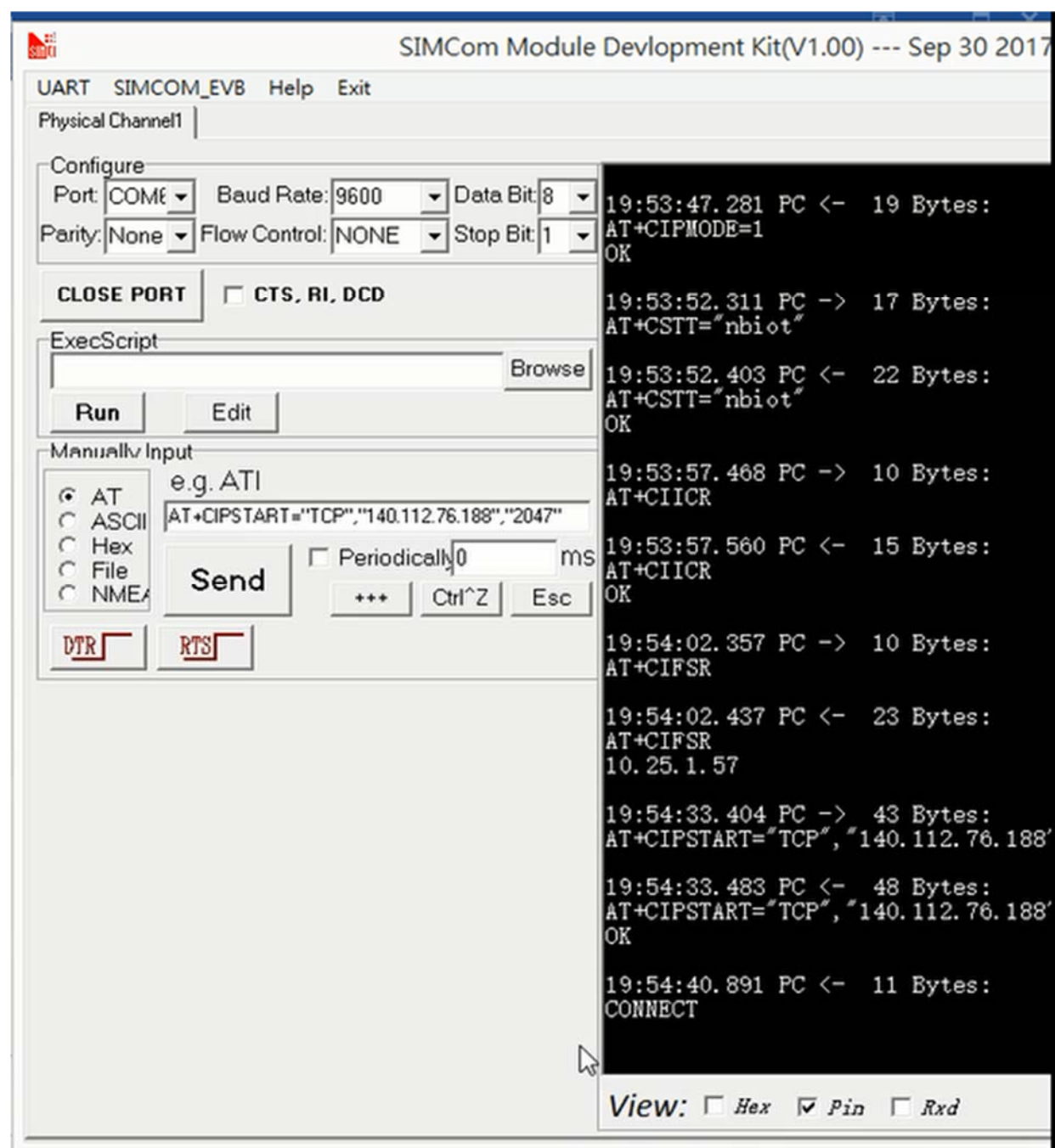
peek()	回傳檔案未讀資料中的第一個字元，回傳後不把該字元認為是已讀
position()	取得檔案現在的位置，該未置標記下一次讀取或寫入的位置
print()	寫入資料到 SD 卡裡的檔案
println()	寫入資料到 SD 卡裡的檔案，在資料寫入完成後在寫入一個換行字元
seek()	移動讀取及寫入資料到檔案的位置
size()	取得檔案的大小
read()	從檔案讀取資料
write()	寫入資料至檔案
isDirectory()	判斷 File 物件所開啟的是否是資料夾
openNextFile()	開啟下一個檔案或是資料夾
rewindDirectory()	回到資料夾的第一個檔案

附錄六：AT command 驗證通訊訊號是否正常運作



附錄七：AT command 架設單鏈路模式





附錄八：梯形積分法程式碼範例

```
2
3 - clear all;
4 - raw_measured_accel=xlsread('accel_freq_6.5_deltat_0.002.xlsx'); % accel.xlsx is result from SAP2000
5 - accel=raw_measured_accel;
6 - [M,N] = size(raw_measured_accel);
7 - dt=0.002;
8 - vel=cumtrapz(accel)*dt;
9 - disp=cumtrapz(vel)*dt;
10
```

附錄九：重建位移法(無時窗)程式碼範例

```
- clear all;
- raw_measured_accel=xlsread('accel_freq_6.5_deltat_0.002.xlsx'); % accel.xlsx is result from SAP2000
- raw_true_displ=xlsread('displ_freq_6.5_deltat_0.002.xlsx'); % displ.xlsx is result from SAP2000
- delta_time=0.1;
- measured_accel=raw_measured_accel;
- true_displ=raw_true_displ;
- matrix_size=101;
- La = eye(matrix_size);
- La(1,1)=1/sqrt(2);
- La(matrix_size,matrix_size)=1/sqrt(2);
- Lc=zeros(matrix_size,matrix_size+2);
- j=1;
- for i=1:matrix_size
-     Lc(i,j)=1;
-     Lc(i,j+1)=-2;
-     Lc(i,j+2)=1;
-     j=j+1;
- end
25 - L=La*Lc;
26 - lamda=46.81*matrix_size^(-1.95);
27 - Id_matrix = eye(matrix_size+2);
28 % estimated_displ=inv(L'*L+lamda^2*Id_matrix)*L'*La*measured_accel*delta_time^2;
29 - estimated_displ=(L'*L+lamda^2*Id_matrix)\(L'*La*measured_accel*delta_time^2);
30
```

附錄十：重建位移法(有時窗_nonoverlap)程式碼範例

```

7 - clear all;
8 - raw_measured_accel=xlsread('accel_freq_6.5_deltat_0.002.xlsx'); % accel.xlsx is result from SAP2000
9 - [M,N] = size(raw_measured_accel);
10 - raw_true_displ=xlsread('displ_freq_6.5_deltat_0.002.xlsx'); % displ.xlsx is result from SAP2000
11 - delta_time=0.002;
12 - measured_accel=raw_measured_accel(:,2);
13 - true_displ=raw_true_displ(:,2);
14 - no_window=20;
15
16 - matrix_size=floor(M/no_window);
17 - residual=M-matrix_size*no_window;
18 - start=1;
19
19
20 - for ii=1:no_window
21 -     if ii==no_window
22 -         the_end=start+matrix_size-1+residual;
23 -         matrix_size=matrix_size+residual;
24 -     else
25 -         the_end=start+matrix_size-1;
26 -     end
27 -     La = eye(matrix_size);
28 -     La(1,1)=1/sqrt(2);
29 -     La(matrix_size,matrix_size)=1/sqrt(2);
30 -     Lc=zeros(matrix_size,matrix_size+2);
31 -     j=1;
32 -     for i=1:matrix_size
33 -         Lc(i,j)=1;
34 -         Lc(i,j+1)=-2;
35 -         Lc(i,j+2)=1;
36 -         j=j+1;

```

```

37 -     end
38 -     L=La*LC;
39 -     lamda=46.81*matrix_size^(-1.95);
40 -     Id_matrix = eye(matrix_size+2);
41 -     % estimated_displ=inv(L'*L+lamda^2*Id_matrix)*L'*La*measured_accel*detla_time^2;
42 -
43 -     if ii ~= no_window
44 -         estimated_displ(ii,:)=(L'*L+lamda^2*Id_matrix)\(L'*La*measured_accel(start:the_end,:)*detla_time^2);
45 -     else
46 -         the_last_estimated_displ=(L'*L+lamda^2*Id_matrix)\(L'*La*measured_accel(start:the_end,:)*detla_time^2);
47 -     end
48 -     %error=abs(estimated_displ(2:matrix_size+1,:)-true_displ);
49 -
50 -     start=the_end+1;
51 -
52 - end
53 - plot (raw_true_displ(:,1),raw_true_displ(:,2));
54 - xlabel('time (sec)');
55 - ylabel('displacement (m)');
56 -
57 - % matrix_size
58 - hold on;
59 - if no_window ~=1
60 -     temp_estimated_displ_col=estimated_displ(:,2:matrix_size-residual+1)';
61 -     temp_estimated_displ=temp_estimated_displ_col(:);
62 -     final_estimated_displ=[temp_estimated_displ;the_last_estimated_displ(2:matrix_size+1,:)];
63 -     plot (raw_true_displ(:,1),final_estimated_displ);
64 - else
65 -     plot (raw_true_displ(:,1),the_last_estimated_displ(2:M+1,:));
66 - end
67 - str_no_window=num2str(no_window);
68 - title(['window number = ',str_no_window]);
69 - legend('SAP2000','estimated');
70 -

```

附錄十一：重建位移法(有時窗_overlap)程式碼範例

```

7 - clear all;
8 - raw_measured_accel=xlsread('accel_freq_6.5_deltat_0.002.xlsx'); % accel.xlsx is result from SAP2000
9 - [M,N] = size(raw_measured_accel);
10 - raw_true_displ=xlsread('displ_freq_6.5_deltat_0.002.xlsx'); % displ.xlsx is result from SAP2000
11 - detla_time=0.002;
12 - measured_accel=raw_measured_accel(:,2);
13 - true_displ=raw_true_displ(:,2);
14 - time_for_each_window=2.5; % in second
15 - window_size=time_for_each_window/detla_time;
16 - % total_run_time=M-window_size;
17 - % no_window=1;
18
19 - matrix_size=window_size;
20 - % residual=M-matrix_size*no_window;
21 - start=1;
22 - %
23 - % for ii=1:no_window
24 - %     if ii==no_window
25 - %         the_end=start+matrix_size-1+residual;
26 - %         matrix_size=matrix_size+residual;
27 - %     else
28 - the_end=M-matrix_size;
29 - the_vector_middle_displ=zeros(M,1);
30 - %     end

```

```

31 - for index=start:the_end
32 -     La = eye(matrix_size);
33 -     La(1,1)=1/sqrt(2);
34 -     La(matrix_size,matrix_size)=1/sqrt(2);
35 -     Lc=zeros(matrix_size,matrix_size+2);
36 -     j=1;
37 -     for i=1:matrix_size
38 -         Lc(i,j)=1;
39 -         Lc(i,j+1)=-2;
40 -         Lc(i,j+2)=1;
41 -         j=j+1;
42 -     end
43 -     L=La*Lc;
44 -     lamda=46.81*matrix_size^(-1.95);
45 -     Id_matrix = eye(matrix_size+2);
46 -     estimated_displ=inv(L'*L+lamda^2*Id_matrix)*L'*La*measured_accel(index:index+matrix_size-1)*detla_time^2;
47 -     the_middle_estimated_displ=estimated_displ(ceil((matrix_size+2)/2));
48 -     position=index+ceil(matrix_size/2)-1;
49 -     the_vector_middle_displ(position,1)=the_middle_estimated_displ;
50 - end

62 - plot (raw_true_displ(:,1),raw_true_displ(:,2));
63 - xlabel('time (sec)');
64 - ylabel('displacement (m)');
65
66 - % matrix_size
67 - hold on;
68 - % if no_window ~=1
69 - %     temp_estimted_displ_col=estimated_displ(:,2:matrix_size-residual+1)';
70 - %     temp_estimated_displ=temp_estimted_displ_col(:);
71 - %     final_estimated_displ=[temp_estimated_displ;the_last_estimated_displ(2:matrix_size+1,:)];
72 - %     plot (raw_true_displ(:,1),final_estimated_displ);
73 - % else
74 -     plot (raw_true_displ(:,1),the_vector_middle_displ(:,1));
75 - % end
76 - str_window_size=num2str(time_for_each_window);
77 - title(['window size = ',str_window_size,' second']);
78 - legend('SAP2000','estimated');

```

附錄十二：25Hz 振動頻率下懸臂梁自由端加速度與

SAP2000 位移

時間(s)	加速度(m/s ²)	SAP2000 位移(m)
0	0	0
0.1	0.53183	0.00179
0.2	-5.04561	0.00723
0.3	4.55523	-0.00391
0.4	-4.85482	9.22E-04
0.5	1.43022	-0.02733
0.6	-7.37894	-0.06006
0.7	-7.26074	-0.13444
0.8	-4.87308	-0.24991
0.9	3.66021	-0.40339
1	-6.67759	-0.56671
1.1	-4.0087	-0.78245
1.2	-0.42711	-1.04022
1.3	6.24474	-1.29102
1.4	-1.11969	-1.50695
1.5	0.50262	-1.70923
1.6	-13.48658	-1.90654
1.7	1.00513	-2.18175
1.8	-2.2085	-2.48605
1.9	-9.11571	-2.82474
2	20.10206	-3.19208

2.1	-18.03921	-3.48529
2.2	0.63574	-3.83447
2.3	3.28589	-4.20036
2.4	-1.29102	-4.59071
2.5	-1.72042	-4.97978
2.6	11.53123	-5.39506
2.7	-6.472	-5.80795
2.8	10.6979	-6.26205
2.9	23.94042	-6.65812
3	-39.01294	-6.93316
3.1	69.92668	-7.35972
3.2	-43.29543	-7.51832
3.3	-6.92355	-7.85041
3.4	15.81722	-8.35384
3.5	-8.19537	-8.8761
3.6	17.34639	-9.29706
3.7	15.33199	-9.53196
3.8	-55.36732	-9.71168
3.9	46.90567	-10.0769
4	-57.35871	-10.2796
4.1	44.27282	-10.7563
4.2	-20.46344	-11.0421
4.3	50.82598	-11.411
4.4	-65.81955	-11.4483
4.5	85.04826	-11.7743

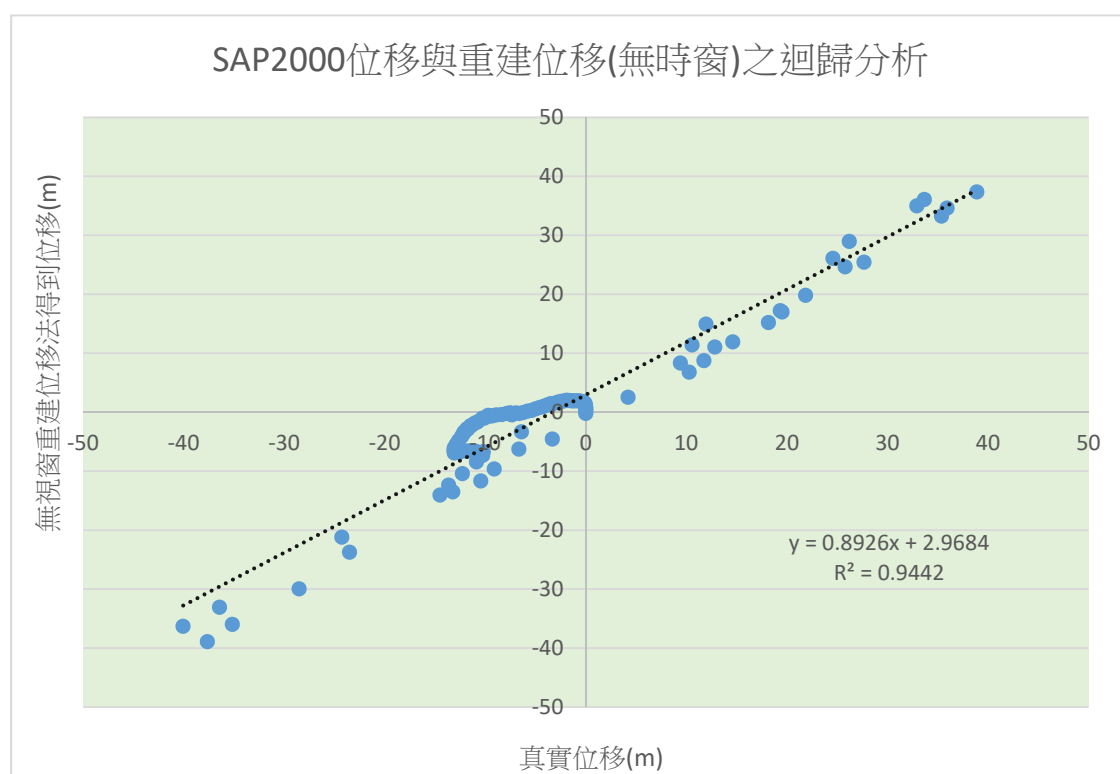
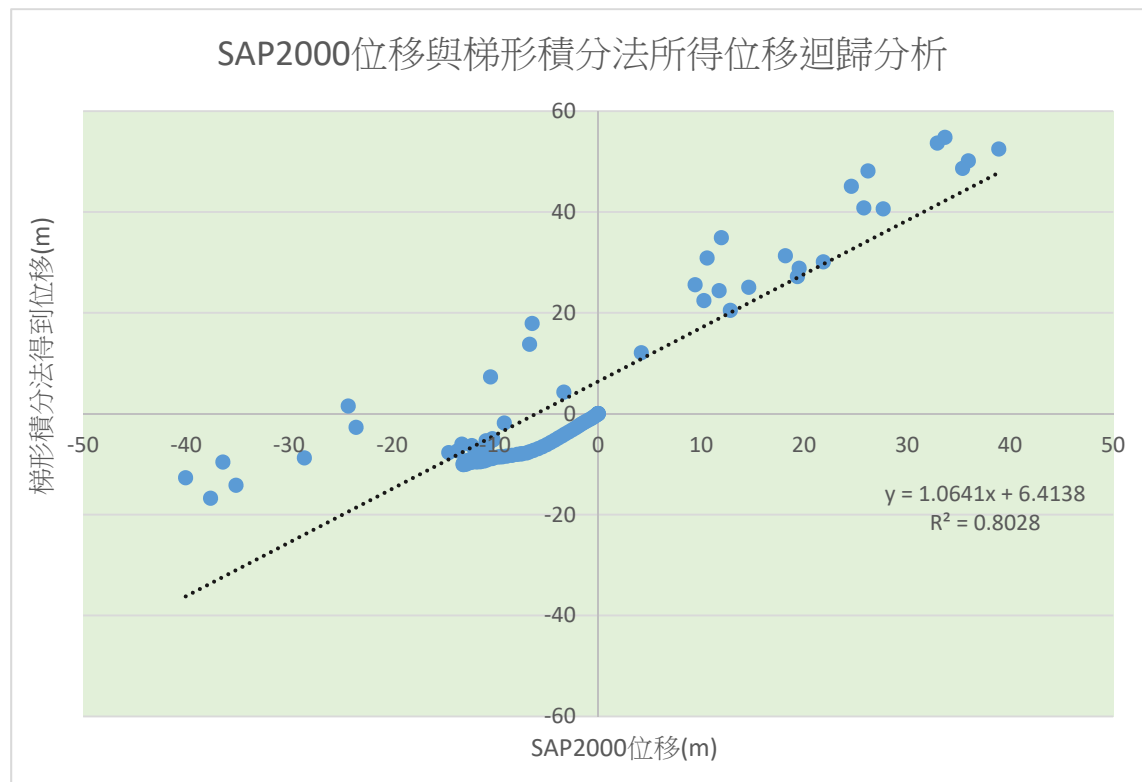
4.6	-63.87833	-11.8153
4.7	25.34939	-12.0792
4.8	-12.53288	-12.1863
4.9	-22.44518	-12.3449
5	13.97772	-12.6433
5.1	5.94918	-12.8584
5.2	22.07848	-13.0636
5.3	-6.02592	-13.1112
5.4	60.74711	-13.0818
5.5	20.18544	-12.6603
5.6	-17.10712	-12.037
5.7	-2.57567	-11.6182
5.8	-3.79518	-11.2419
5.9	-14.34001	-10.9236
6	46.85999	-10.639
6.1	-80.09159	-10.1909
6.2	-44.86226	-10.2587
6.3	-88.28924	-10.8789
6.4	6.41296	-12.2687
6.5	26.93052	-13.6491
6.6	224.57954	-14.5043
6.7	333.88882	-13.2313
6.8	117.81832	-9.10392
6.9	201.73457	-3.32406
7	142.43401	4.20063

7.1	-238.22015	12.83954
7.2	-357.28727	19.35083
7.3	-547.36883	21.86381
7.4	-225.41779	19.52125
7.5	-10.49869	14.62143
7.6	705.512	10.27986
7.7	445.03908	11.74864
7.8	377.36073	18.17521
7.9	-248.83565	27.67743
8	-375.98526	35.38776
8.1	-686.12319	38.90193
8.2	-727.21031	35.94421
8.3	-637.27038	25.79857
8.4	-373.67868	9.41203
8.5	168.62499	-10.4458
8.6	930.37126	-28.5187
8.7	1186.06259	-37.6319
8.8	931.02368	-35.158
8.9	519.67862	-23.495
9	18.71644	-6.6486
9.1	-301.33143	10.58119
9.2	-581.43022	24.58891
9.3	-782.56075	32.93295
9.4	-824.33016	33.67229
9.5	-691.33399	26.20829

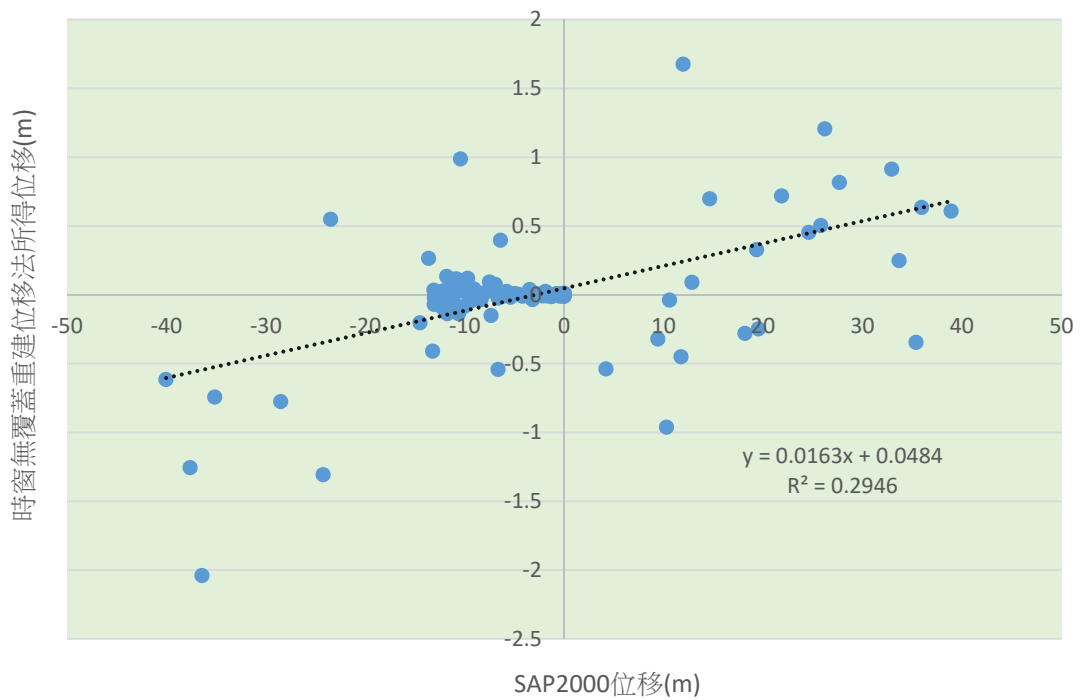
9.6	-426.3904	11.96164
9.7	41.16391	-6.40687
9.8	595.9074	-24.2587
9.9	865.01736	-36.4313
10	884.53812	-40.0549

附錄十三：25Hz 振動頻率下 SAP2000 位移與其他演算法得

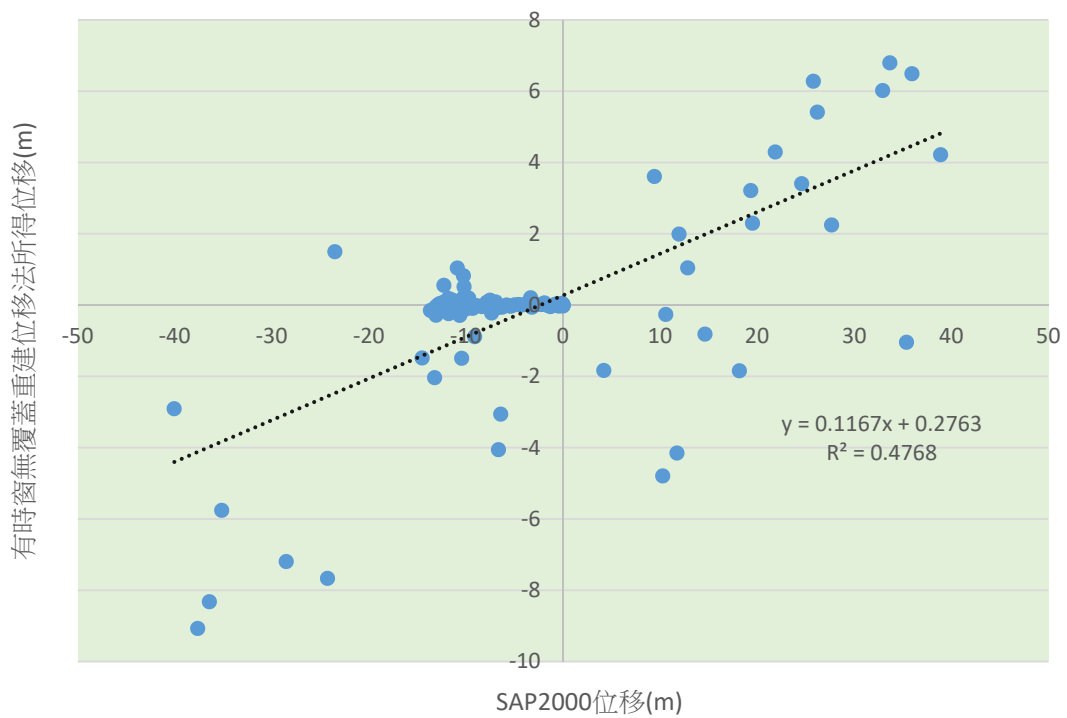
出位移迴歸圖

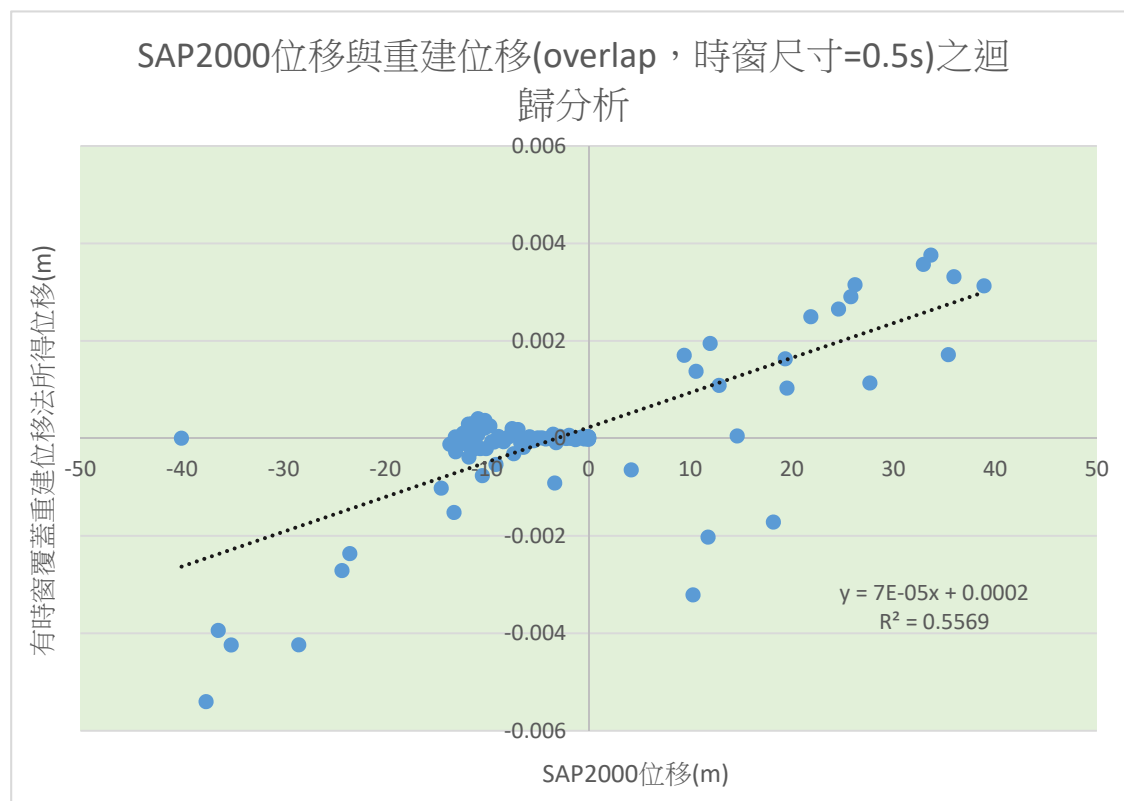
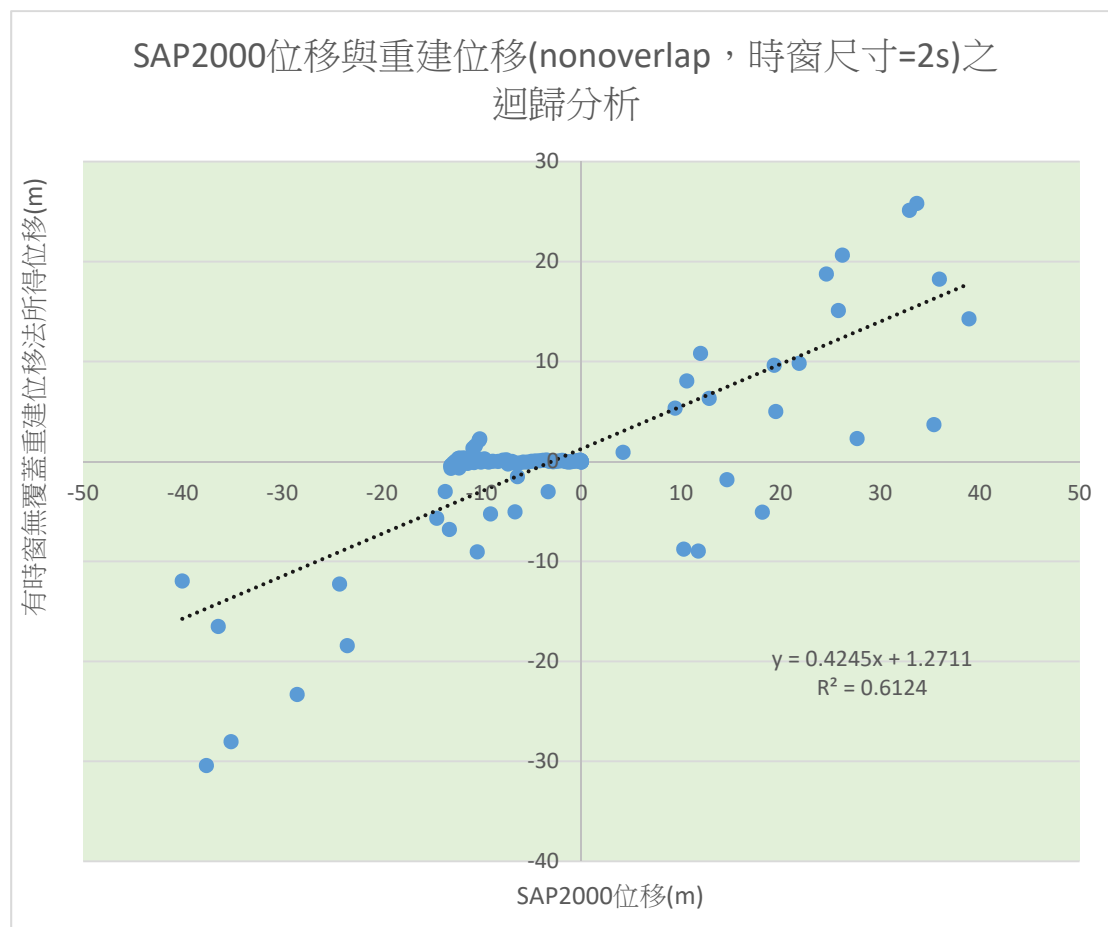


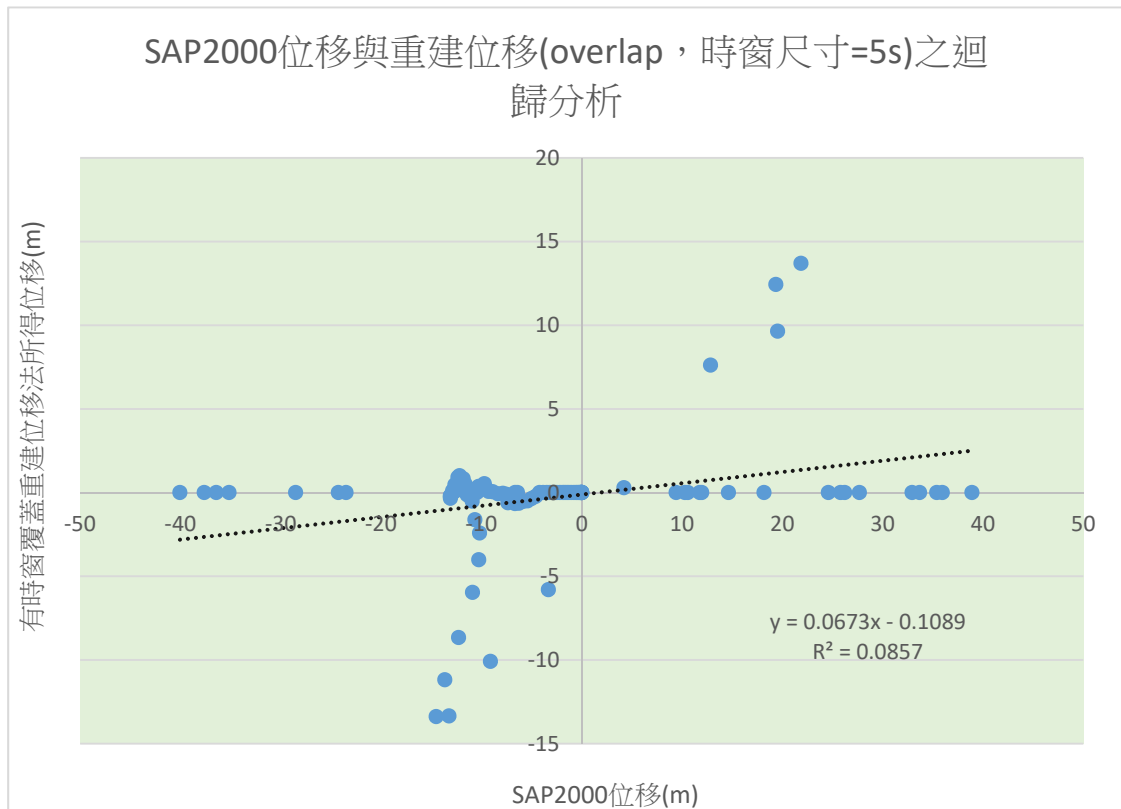
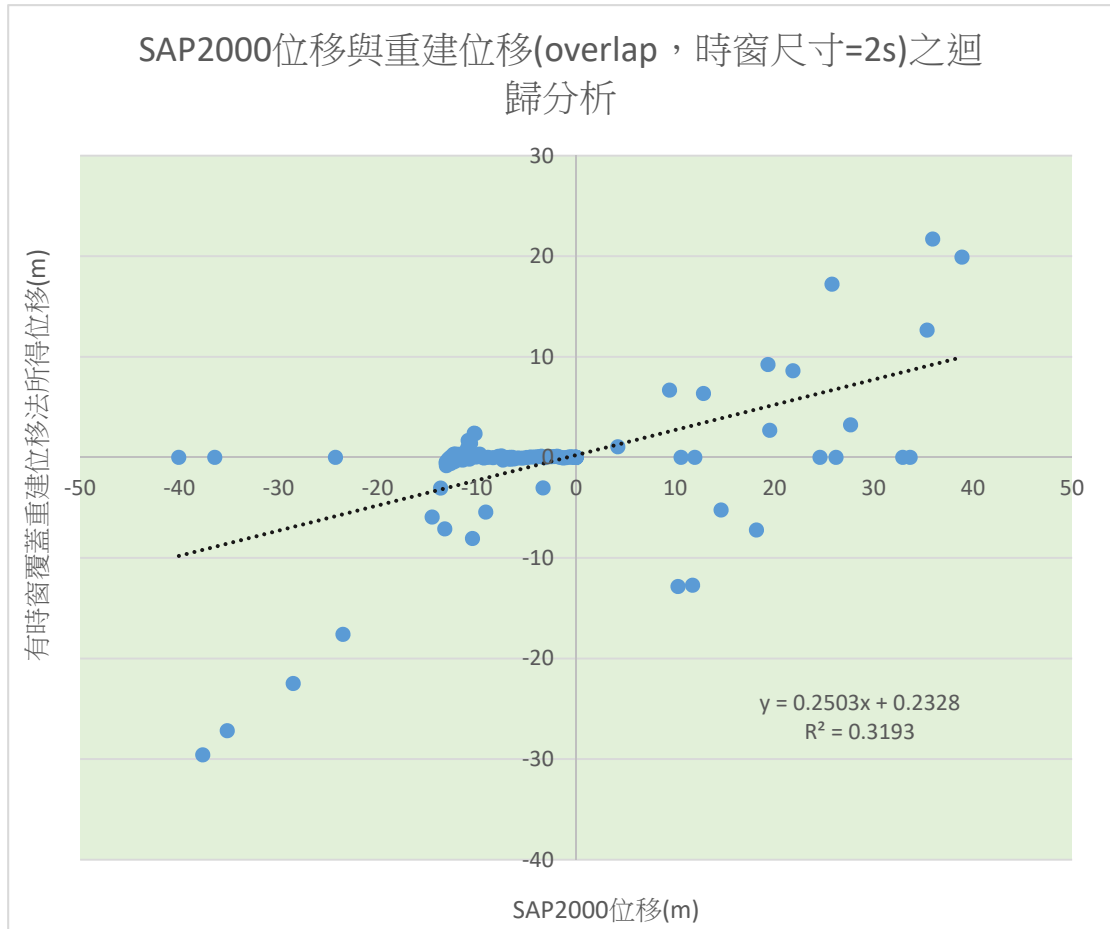
SAP2000位移與重建位移(nonoverlap，時窗尺寸=0.5s)
之迴歸分析



SAP2000位移與重建位移(nonoverlap，時窗尺寸=1s)之
迴歸分析







附錄十四：6.5Hz 振動頻率下 SAP2000 位移與其他演算法

得出位移迴歸圖

