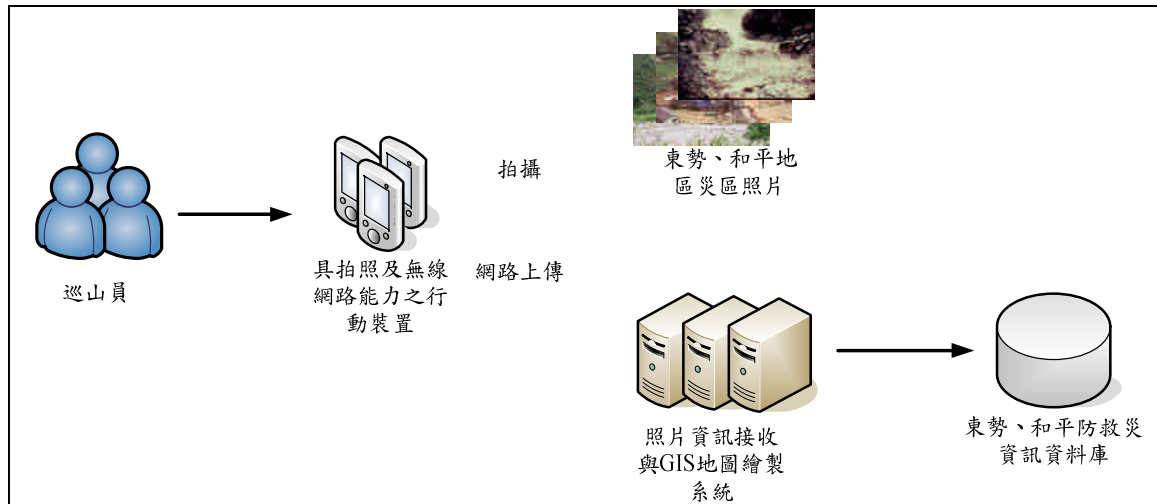


## 子計畫五：社區災害空間分析與監視系統之建構結案報告

本子計畫的主要目標有二，其一為協助各子計畫進行社區災害之空間分析，其二為建立一社區災害之監視系統。對於前者，我們係建立一資訊系統以為各子計畫間溝通平台，並配合 GIS 提供資源、空間分佈、及資料分析與查詢等功能，我們也將各子計畫所設計問卷之調查結果建置在本子計畫所建立的基本資料庫中，並可以之建立多種社會資源服務分佈空間圖層等。

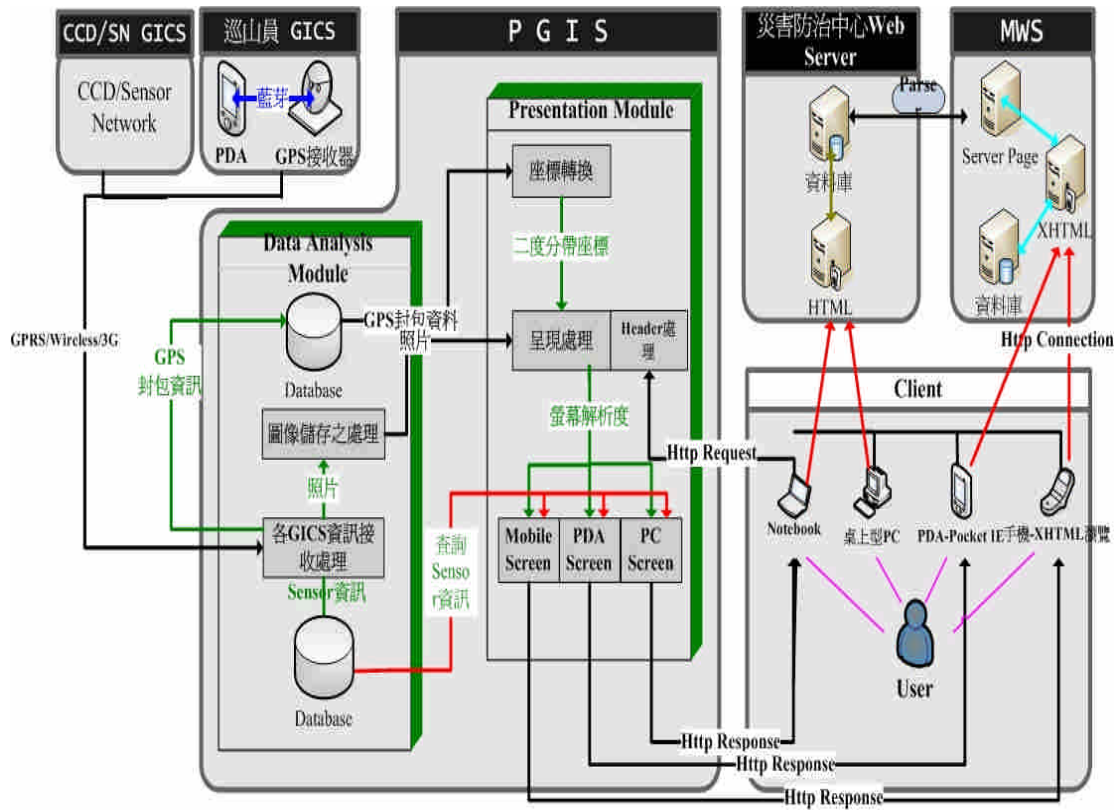
在地方政府災區組織資源管理系統架構上，資料庫的建置為主要關鍵角色。為了瞭解災區區位間之損害情形和災民需求，及交叉分析防救災資源特質與環境，與釐清資源配置及所擬定之計畫間相互關係，建立一個精確且完整之資料庫是不可或缺的。再者，為提高研究場域之社區韌性並降低其面臨災害所造成之損傷，本子計畫的第二項目標就是研製一社區災害監視系統，這個系統分成人員監控和自動監視兩部分。前者是由人員赴現場勘查災害損害狀況，再將結果輸入系統中；後者則是自動監視災害使否正在發生或已經發生了，本子計畫亦設計一山洪暴發之監視系統，其在偵測出山洪暴發時，會主動發出警報警告下遊在河床上遊戲或工作之人員，人員監控部分如圖一所示，包括巡山員 GICS(Charged Coupled Device/Sensor Network Geographic Information Collection System)子系統、PGIS(Presentation GIS Information System)照片接收資訊系統與 GIS 地圖繪製三部分。當災害發生時，巡山員可透過行動裝置拍攝災害地點之照片，並藉由無線網路上傳至系統中，提供專業分析人員一項防災害資訊策略擬定參考指標，其所發展之程式詳見附錄，而自動監視係以土石流為主要對象，這一部份原預計在本計畫第二年研究之，係準備以 CCD 攝影機監視之，再以圖形處理分析方式監測之，目前已具偵測雛形。唯，第二年之計畫並未獲計畫審查委員及國科會之青睞而停止。預計日後有研究經費時，在繼續研究開發之。



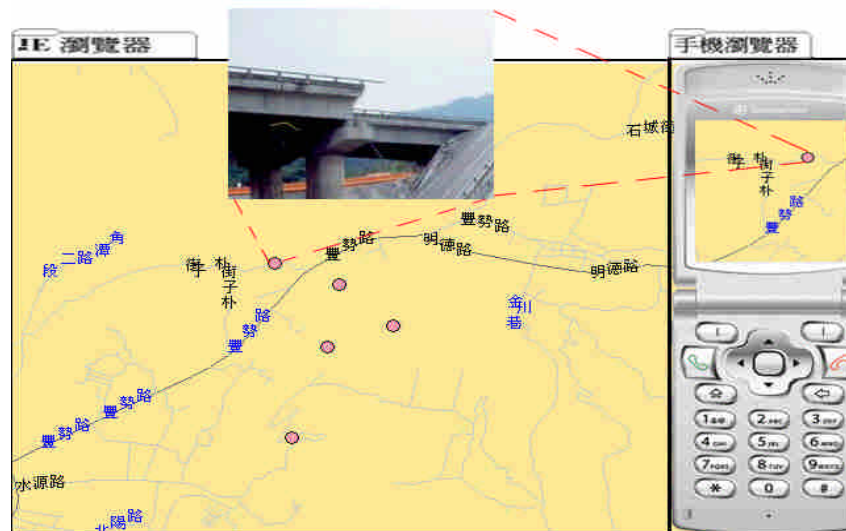
圖一. 瀏覽資訊時使用者端之畫面 (手機及電腦)

而技術細節茲說明如下：

如圖二所示，巡山員攜帶具有拍照能力與 GPS 定位裝置之智慧型手機，於發現異狀時即時拍照並取得當地座標，再利用無線通訊之傳送方式傳送到 DIGS。這些功能除了拍照之外都不是手機現有功能，需由研究人員一一開發之。DIGS 又分為兩大模組，資料分析模組(Data Analysis Module, DAM)與呈現模組(Presentation Module, PM)，其中 DAM 負責接收與處理 GICS 所回傳之資訊，其中包括了圖片處理、GPS 封包分析。PM 中又分為兩部份，第一部份為負責分析使用者透過網頁瀏覽器(Web Browser)所發送出的需求(Http-Request)，並適當得切割地圖給使用者(PDA、手機、PC 與 Notebook 螢幕之解析度皆不同)，在此若未能得知客戶端之解析度，則以 128x120 之解析度為預設值，此為目前各家手機中最小的解析度。第二為座標轉換，將 GPS 接收器所接收到之 WGS84 轉換至 TWD97 之二度分帶；MWS (Mobile Web Sever) 為提供手機網頁之網頁伺服器，其客戶端之對象為支援 XHTML 之手機。



圖二. 系統流程示意圖



圖三. 瀏覽資訊時使用者端之畫面 (手機及電腦)

## 1. 巡山員 GICS 系統

使用者透過手機上之 GICS 程式拍照時，該程式會先透過藍芽介面(見附錄

一)，向全球定位系統(本系統所採用者為長天科技公司生產之 GPSlim236 型號)取得全球定位資訊後，將此資訊與拍攝之相片傳送至 PGIS 伺服器，該伺服器則會解析定位系統封包，以取得其中之地理座標。由於 GPSlim236 採用國際通用之 WGS84 經緯度，因此 PGIS 必須將之轉換成目前台灣常用之 TWD97 格式，並存入行動網頁伺服器之行動資訊資料庫中。照片則存入行動網頁伺服器的照片儲存目錄下，供使用者透過手持裝置查詢與瀏覽(見圖三)，至於轉換公式則已述於計畫書中，不令贅述。

## **1.1. 藍芽**

### **1.1.1. 原理**

Bluetooth 是以無線的方式將多種數位設備相互間傳送 Data，此通訊規格被定為 IEEE 802.15.1。應用面以行動電話、筆記型電腦、PDA(如 Windows CE 或 Palm 類似之作業系統)與 GPS 接收器為核心，所延伸出超出想像以外的各種設備，透過 Bluetooth 技術將其連結起來。目前常用的無線資料交換，尚有電腦和行動裝置間所泛用的 IrDA 技術，利用此一技術來執行無線 Data 傳達。但是 Bluetooth 在傳輸速率、消耗功率、傳輸距離、安全性(Frequency Hopping 傳輸技術)...等都略勝 IrDA 一籌。

### **1.1.2. 特性**

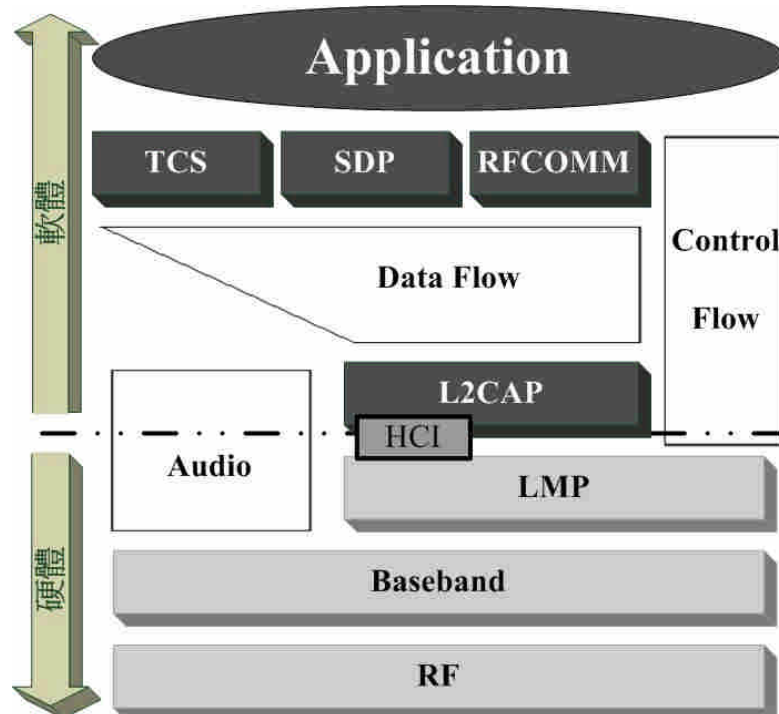
藍芽之特性如下：

- (1) 全球可操作性。Bluetooth 使用 2.4GHz 的 ISM (industrial, scientific, and medical) 頻段，在全世界各個國家都是有效的，使用上不會受到區域之限制。
- (2) 10—100 公尺的有效無線通訊距離。發射功率為 1mW(0dBm)時的有效通訊距離是小於或等於 10 公尺，這種情況適用於短距離之設備，例如滑鼠、鍵

盤、耳機等之使用；在發射功率為 100mW(20dBm)時的有效通訊距離則小於或等於 100 公尺，適合設備經常變動位置的環境，如無線電話 (cordless phone)、行動電話，筆記型電腦...等。

- (3) 1Mbps 的數據傳輸率。今後所發展的版本將支持 2Mbps 的數據傳輸率。
- (4) 同時支持語音和數據傳輸。
- (5) 小尺寸，低功耗和低價格。Bluetooth 為了節省設備的耗電量採用了許多措施(例如：單芯片 Bluetooth 的 IC)。
- (6) 全球唯一的 Bluetooth Address 代表每一台 Bluetooth 裝置在全世界的唯一識別號。
- (7) 可建構一個 8 個節點的小型區域網路(PAN：Personal Area Network)，其基本架構為 Piconet。

### 1.1.3. 藍芽之系統架構



圖四. 藍芽系統架構圖

從圖四中，可得知 Bluetooth 系統架構分為 Radio Frequency(RF) & Baseband、LMP(Link Manager Protocol) & L2CAP ( Logical Link Control

&Adaptation Protocol)、HCI(Host Controller Interface)、Application Framework & Support 等四個層次類別。其中若以 HCI 來區分，HCI 以下的部分包括了 RF、Baseband 及 Link Manager 三部分，都是由硬體實作的，而 HCI 以上的包括 L2CAP 以及上層的一些通訊協定，這一些則是以軟體方式完成。

而射頻模組(RF Module)的主要功能係負責頻率的合成、資料轉換和雜訊的過濾等，基頻模組(Baseband Module)則負責訊息的編碼、加解密、跳頻以及封包的處理部份。Link Manager & L2CAP 相當於 OIS 模式(OSI model)底層通訊協定的功能(如 Physical layer; MAC layer)，其中 Link Manager 負責網路連線的設定、釋放及管理，通常用於 Personal Area Network(PAN)之連接；L2CAP 則提供資料封包的切割與組裝(Segmentation & Reassembly)、通訊協定的多工處理(Multiplexing)以及服務品質(QOS: Quality Of Service)等服務功能，且通常以韌體(Firmware)的形態存放在 Baseband Module 硬體中。此外，HCI 主要用來界定 Bluetooth 與 Host 設備之間連結介面的硬體控制指令。Application Framework & Support (如圖四最上層部分)則是依據語音、數據、控制等應用需求，提供應用軟體所需的通訊協定功能與應用程式介面，在此層面下包括有：RFCOMM(RF Communication)有線連接取代協定，係由 RF 通訊來完成取代有線連接的功能；SDP (Service Discovery Protocol)則為服務尋找協定，這部分是藍芽的核心機制；TCS (Telephone Control Service) 為電話相關的電話控制服務，包含了電話控制協定二進位(TCS Binary)及 AT 命令(AT Commands)，主要是用來控制電話及數據機之接訊即通訊操作部分。

#### **1.1.4. 藍芽技術應用於本計畫之部分**

在本計畫採用的 GPS 接收器為藍芽通訊介面，其中又以 Serial Port Profile 為應用規範。而藍芽有許多的應用規範，例如，藍芽耳機所採用的為 Headset Profile 應用規範。

Serial port Profile 係架構在 RFCOMM 介面之上，RFCOMM 實做了 ETSI TS

07.10 的一部份，並且訂定一些藍芽專用的規格。藍芽的 RFCOMM 其實就是模擬 RS-232，RS-232 即為大家所熟知的 COM Port 介面。藉由模擬 RS232，藍芽可以直街的擔當目前 RS232 所提供的許多功能，如：數據機、滑鼠、PDA 同步等。在 RS-232 的傳輸中，有四個參數是依據不同的裝置而會有不同的數據，其為 Com Port Number、Baud Rate、Parity Bit 與 Stop Bit。在與大部份 GPS 接收器通訊時，本研究所得到的數據為

Com Port Number = 7，

Baud Rate = 38400，

Parity Bit = None，

Stop Bit = 1。

我們便是以採用這些數據，讀取資料。

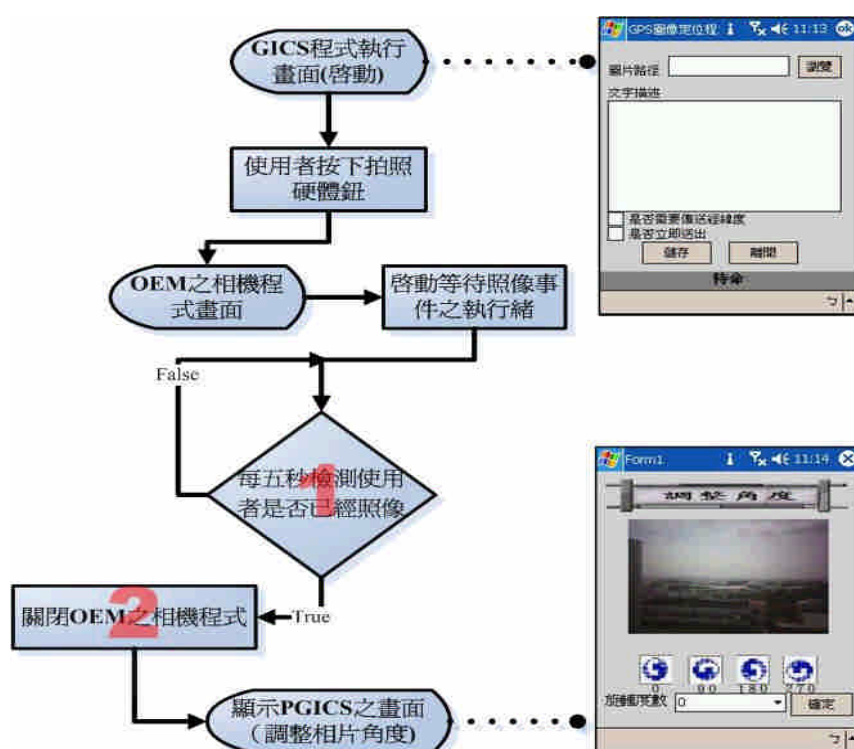
## 1.2. 巡山員 GICS 設計原理

本系統主要目的為蒐集地理座標位置、圖片描述與照片，其主要開發平台為 Smart Phone mobile 5.0。現在市面上所販售的 Smart Phone 中，由於 Mobile 5.0 作業系統中並不支援各 OEM 相機提供熱插拔(Plug-and-Play)之驅動程式，所以廠商均以 OEM 方式或字形研發其驅動程式，而我們所在 GICS 中設計一套整合各 Smart Phone 相機之方法，以增加此系統的移植性(Portability)。

隨著通訊技術的發展，無線網路已經逐漸地普及了，但是在連線穩定度以及重複收訊範圍方面，仍然會有斷線的情形，本系統採用了續傳的機制，以防止斷線後必須重新傳送而造成傳輸斷線前之資料而浪費頻寬的現象。GICS 亦考量使用者在沒有無線網路的環境中操作的窘境，因此採用雙傳輸通道之方式，其分別為無線網路與 GPRS，兩者交替使用過程中，在交替過程中，不論是從 WiFi 到 GPRS 或反向，GICS 都會以續傳方式接續傳送，使用者無須擔心上述浪費頻寬(即多付費)之現象。

### 1.2.1. 相機移植之原理

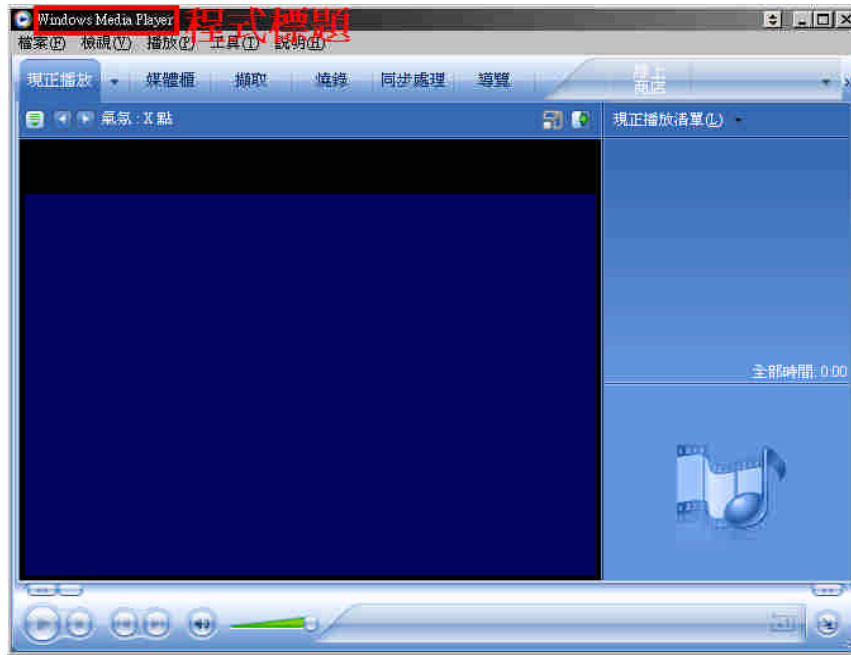
在每家 Smart Phone 廠牌皆會為自己的相機硬體開發出一套完整的套件，以方便軟體開發人員控制此硬體，但這些套件往往是商業機密，是不被公開的，也因此 GICS 在開發上增加了一定程度的困難度。在使用過程中，GICS 須對 OEM 之相機程式執行兩個動作，第一為取得使用者按下拍照硬體鈕之觸發事件，第二為關閉 OEM 之相機程式，系統相機部份操作流程如圖五所示。



圖五. GICS 照像部份流程

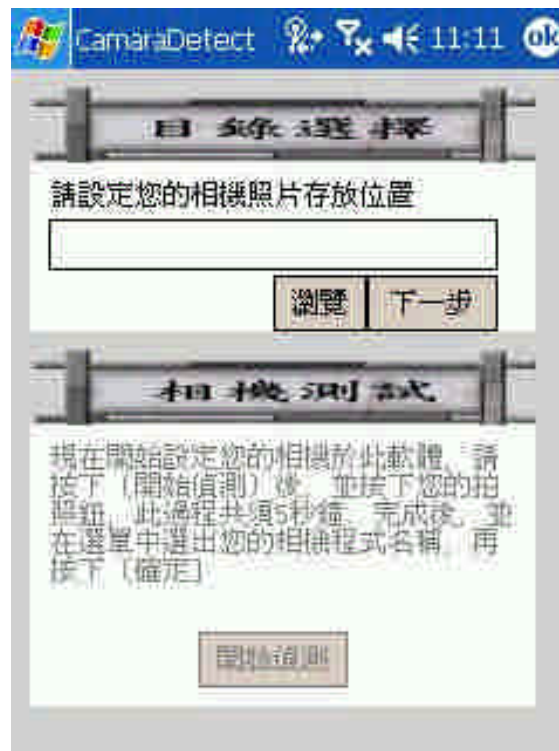
為了正確執行這兩項控制，GICS 須先取得每台相機之兩個參數。第一為相機在作業系統中視窗的標題名稱，其通常為程式名稱，如圖六所示；第二則為每一台具照相功能之相機皆會有預設之照片儲存目錄，通常預設為 \\My Document\\My Picture 中，除非使用者另行更改之。





圖六. 其程式標題名稱為 Windows Media Player

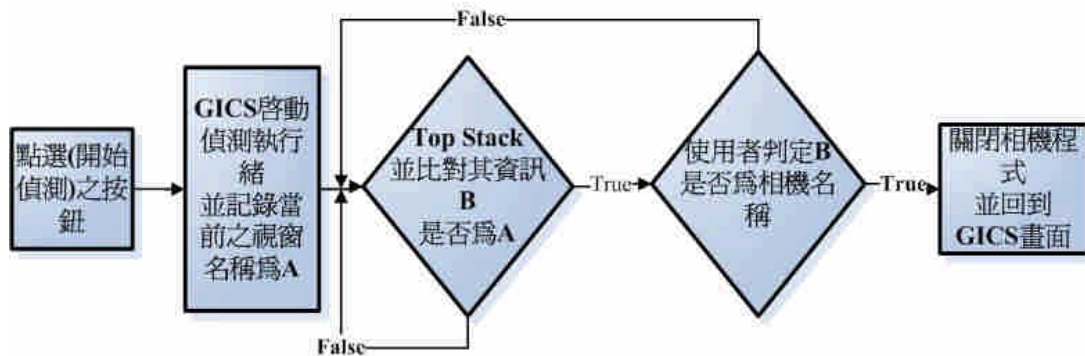
GICS 中則提供一套人性化的設定介面，使用者可以之直接抓取這兩參數，如圖七所示，而不必在相機控制程式中一一尋找之。



圖七. GICS 設定必要兩個參數之畫面

以下將介紹相機程式名稱偵測之原理。作業系統中利用堆疊(STACK)記錄每一視窗之標題名稱，每當使用者開啟新視窗，即將此視窗之標題名稱 Push 到 Stack

中，當下達 Top 之指令時則可取出最後開啟之視窗之名稱。使用者按下「開始偵測」之按鈕，GICS 會先抓取目前最後開啟的視窗標題名稱(CameraDetect，即 GICS 相機設定之視窗)，並同時啟動 Thread，每五秒鐘比對 Top 出來的最後視窗名稱是否為(CameraDetect)。若比對結果「True」則代表使用者尚未呼叫相機程式；若「False」代表相機已開啟新的視窗，此時使用者須確認抓取之視窗名稱是否為此 OEM 之相機名稱，其流程圖如圖八所示。



圖八. 相機偵測之流程圖

在得知相機程式名稱後，我們則可以利用此關閉相機程式。在圖形化的作業系統中，可以利用程式名稱去找出此 Process 所使用的記憶體區塊，當我們將此區塊釋放，則此程式則已被關閉。

接著，我們利用已知拍照的預設目錄，即可抓取使用者的拍照時間點。在此我們利用一個執行緒，當使用者點選程式中「啟動」鈕後，此執行緒會開始執行，每五秒鐘會去檢查預設目錄是否有新的檔案，若有，則代表時用者已經拍攝新的照片，則將此照片帶入調整角度的功能畫面中，以利使用者對照片之後續編輯。

### 1.2.2. 續傳原理

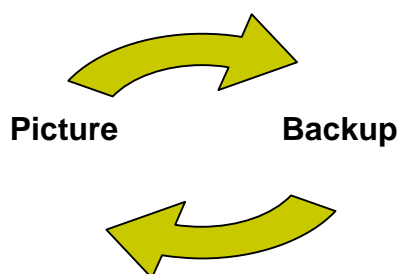
GICS 連接上 PGIS 中的 DAM 時，DAM 會產生一組 8 位數的亂數傳送給 GICS，這時 GICS 將此亂數保存在記憶體中，直到 GICS 完成傳送工作或放棄傳送工作，此亂數才會從記憶體中清除。8 位數之亂數在此不僅做為 DAM 接受 GICS 傳送工作之辨識(ID)，亦為接收圖片之檔案名稱，而也是此次傳送 GPS 資

訊之任務時在資料庫中的主鍵值(Primary Key Value)。DAM 與 GICS 之傳輸過程中，依流程區分為兩種，第一為非續傳需求，第二為續傳之需求，其中 DAM 依據 GICS 所傳送之重送識別碼(Resend Code, RS)作為判別。

在非續傳需求中，DAM 與 GICS 需作 3 way handshaking，其流程如下：

- 第一次 Handshake：GICS 傳送 RS 到 DAM，其值為 49。
- 第二次 Handshake：DAM 傳送 8 碼之亂數，作為此次傳輸之 ID。
- 第三次 Handshake：GICS 傳送檔案類型給 DAM(\*.jpg、\*.gif...等)。

完成 3-way-handshaking 後，DAM 會依據 ID 與檔案類型建立一個檔案於檔案系統中。檔案系統之配置分為兩個資料夾，如圖九中所示，Picture 資料夾中為已完成之圖片檔案或傳送中之圖片檔案；Backup 資料夾則暫存未完成之圖片檔案。其中，picture 到 Backup 代表該傳送之圖片應連線中斷而導致照片傳送不完全故將該照片移至 backup 儲存之，反之，當 backup 到 picture 則代表如前述傳送不完全之照片已經過我們所研發之續傳功能完成傳送。



圖九. 照片之檔案系統

檔案建立後，DAM 開始準備接收 GICS 所傳入之資訊。傳入資訊依順序排列為：照片描述、GPS 封包資訊、照片轉置角度(可轉 90°，180° 及 270°)、照片檔案四項，其系統分別設定 Buffer 大小如下表：

表一. Buffer 大小列表

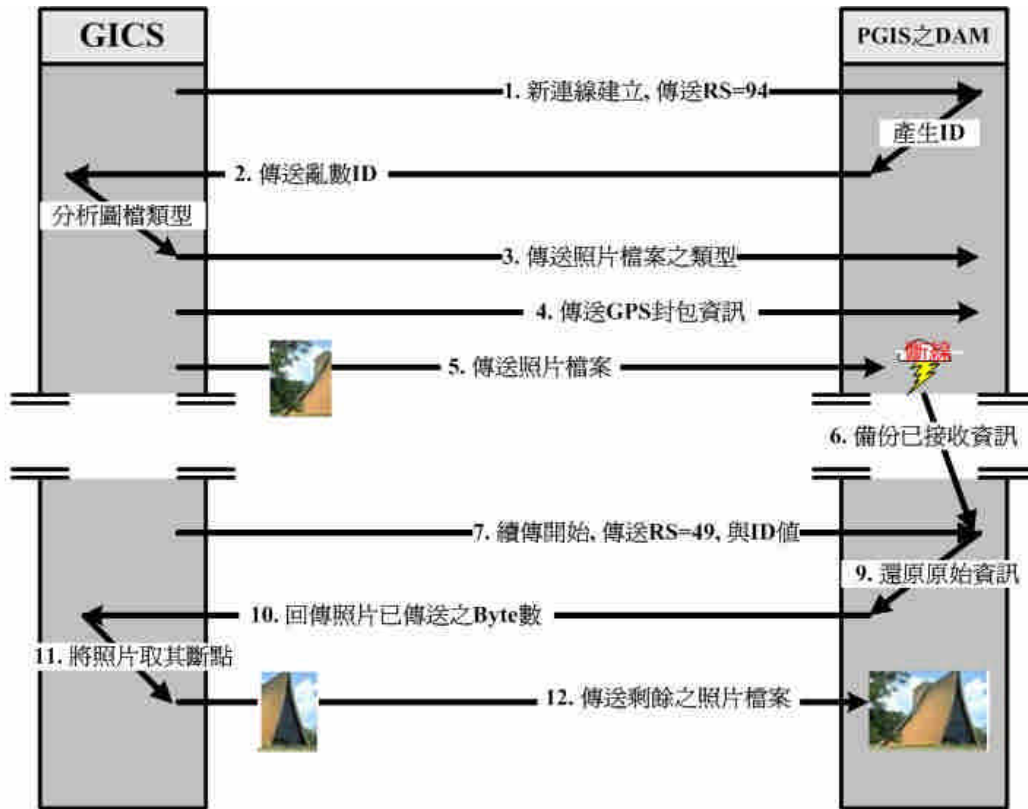
資訊	Buffer 大小
照片描述	100 bytes
GPS 封包資訊	50 bytes
照片轉置角度	3 bytes
照片檔案	無限制

倘若傳輸過程中，GICS 因斷線而傳輸中斷，且斷線時正在傳輸「照片檔案」，則 DAM 會將(ID, {照片描述, GPS 封包資訊, 照片轉置角度, 已傳送照片檔案之 Byte 數})配對，並以 ID 為 Key 值存入事先建立的 Hashtable，該 Hashtable 為 Global Variable。反之，若不在傳送「照片檔案」的階段，照片檔案之前的三項資料在整體資料中所占比例僅約 0.7%(=資訊檔案大小+圖片資訊大小/GPS 資訊檔案大小+圖片資訊大小+圖片檔案大小=153/153+20000)，因此不列入續傳考慮，即下次須四項資料均重送。接著，DAM 亦會將未完成之照片檔案移至 Backup 資料夾；然，GICS 則會將 ID 值保留起來，等待使用者重新連線，直到使用者連線完畢，才會將 ID 值清除。當使用者要求續傳時，GICS 會再將此 ID 值傳送給 DAM，這時會以第二流程「續傳需求」處理此傳輸需求，傳送照片前亦先作 3 way handshaking，流程如下：

- 第一次 Handshake：GICS 傳送 RS 到 DAM，其值為 94(代表要續傳)。
- 第二次 Handshake：GICS 傳送 ID 值給 DAM，此時 DAM 會依據此 ID 值取得 Hashtable 中對應之資訊，並依據此 ID 從 Backup 資料夾中找出其未完成之照片檔案，並移至 Picture 之資料夾。
- 第三次 Handshake：DAM 將取出之資料中的「已傳送照片檔案之 Byte 數」，

並傳送給 GICS。

接著，GICS 得知「已送照片檔案之 Byte 數」後，將接續傳送未傳送之部份。在此，以圖示之方式說明斷線後再續傳的程序，請參考圖十。



圖十. 斷線後再續傳之說明圖

在兩種情形下，DAM 會將 GICS 之 Socket 連接視為斷線傳輸，第一為 Wireless 傳輸或有線傳輸在斷線前，會由 GICS 送出斷線訊號給 DAM，DAM 可根據此訊號處理續傳之動作；第二為 GPRS 傳輸，斷線時 GICS 只會將斷線訊號傳送給 GPRS 電信業者，但電信業者並不會將此斷線訊號傳送給 DAM，因此 DAM 會將 GICS 視為連線中，但卻接收不到任何照片封包，也因此 DAM 會處於 Idle 狀態，所以在此將 DAM 之 Read Time Out 值設為 5 秒，在 5 秒內若無來自 GICS 之資料，則將此 Socket 連線視為斷線，強制中斷其 GICS 連線，並同時處理續傳之動作。

## 2. PGIS 製作(程式碼詳見附錄二)

### 2.1. 使用者端解析度之判別

手機、個人數位助理與個人電腦三種裝置之螢幕解析度並不相同，為能呈現地圖於上述裝置之螢幕，必須將原地圖切割成大小合宜之較小地圖。在使用者端送出 HTTP Request 要求傳送網頁時，會將一些參數放在該封包之標頭(Header)中。其中部份參數是由 W3C 規範所定義，另一部份則由瀏覽器製造商定義之，W3C 規範之參數有：User-Agent、Host、From、Date...等。我們乃利用此標頭中之資訊判別解析度，目前個人電腦所採用之瀏覽器大多以 Microsoft Internet Explorer 居多，所傳送之標頭資訊中的 User-Agent 為「Mozilla/4.0 (compatible; MSIE 6.0; Windows NT 5.1; SV1; .NET CLR 1.1.4322)」；個人數位助理之瀏覽器以 Pocket IE 居多，其 User-Agent 為「Mozilla/4.0 (compatible; MSIE 4.01; Windows CE; Smartphone; Dopod575/1.0 Profile/MIDP-2.0 Configuration/CLDC-1.1; Smartphone; 176x220)」。

由於各家廠商手機之瀏覽器不一定相同，因此無法全部利用 User-Agent 參數檢測之，但是在各種標頭的參數中，可以發現各廠商之手機皆有其共通特徵，即，以 WinCE 為平台之瀏覽器的手機製造商會將參數定為「UA-PIXELS」，其資料格式為「width x height」；部分廠牌手機，例如：Panasonic 與 BenQ 等則將參數定成「X-UP-DEVCAP-SCREENPIXELS」，其資料格式為「width, height」。依據此，我們便能知悉大部分手機的廠牌，及螢幕之 width 及 height(即其解析度)。若使用者端之瀏覽器皆不符合上述需求，我們則以瀏覽器之最小可能值 128 x 120 為其預設(default)之解析度。

### 2.2. 手機網頁--- XHTML

XHTML(The Extensible Hypertext Markup Language1.0)建議規格已於 2000

年 1 月 26 日由 W3C(World Wide Web Consortium)所宣佈。XHTML1.0 是將 HTML4 修改為符合 XML1.0 語法的規格。故我們可以簡單地說：XHTML 是以 XML 技術為基礎的 HTML，使 Web 世界朝向模組化(Modular)及可擴展化。XHTML 比傳統的 HTML 更為嚴謹，目前各網站之 HTML 網頁，能完整符合 W3C 認證，更是少之又少。

雖然 HTML4 在市場佔有率以及功能上已經比 XHTML 占上風，但是 HTML4 不適合運算速度慢、記憶體少的設備讀取，例如：手機、電視、PDA、販賣機、衛星導航系統與電子書閱讀器等。XHTML 在網頁架構的規範上也為手機設計了一套貼心的服務，XHTML 將網頁分成兩部份，第一部份為純文字內容部份，第二部份為排版、配色與貼圖之功能，其中第二部份利用 Cascading Style Sheets (CSS)的方式套用之。其實此觀念在 W3C 定義的 HTML 規格中已詳細定義，但是現在的網頁設計人員往往利用 HTML 圖形介面開發工具(Microsoft office frontpage、DreamWaver...等)開發網頁，導致在網頁架構中無法仔細的分辨出上述兩部份之功能。XHTML 定義這兩部份功能，能讓使用者自行從事讀取的選擇，若選擇只讀取純文字內容，即為只讀取\*.xHTML 檔案，其中 xHTML 檔案已比傳統的 HTML 檔案大小少了許多(包括排版、配色與貼圖的程式碼)。另一種讀取方式即\*.xHTML 檔案與\*.CSS 檔案同時讀取，如此一來與原本的 HTML 所設計出的網頁除了在螢幕大小外，其餘皆雷同，換言之，此為利用頻寬換取網頁的美觀。定義這兩部份的另一個優點為\*.xHTML 網頁使其可向下相容 WAP 格式，即使在一支不支援 CSS 手機中，亦可瀏覽到網頁之相關內容。

### 2.2.1. 網頁 Agent 技術

因為 web 資料大多以資料庫的方式儲存，也因此當再次呈現資料時，往往會利用迴圈的方式去讀取資料，所以 HTML 中每一筆資料的頭和尾都會有固定的 HTML Tag。在此，我們即立用這些頭尾的 Tag 來過濾出我們所需要的資訊。以下，我們以例子來說明，圖十一為一個簡單的 HTML 表格網頁，其 HTML 程

式碼如圖十二所示。

測試標題一	測試標題二	測試標題三
資料1-1	資料1-2	資料1-3
資料2-1	資料2-2	資料2-3
資料3-1	資料3-2	資料3-3

圖十一. HTML 網頁

```
1. <html>
2. <body>
3.     <table width = "50%" border="1">
4.         <tr><th>測試標題一</th><th>測試標題二</th><th>測試標題三
           </th></tr>
5.         <tr><td>資料 1-1</td><td>資料 1-2</td><td>資料 1-3</td></tr>
6.         <tr><td>資料 2-1</td><td>資料 2-2</td><td>資料 2-3</td></tr>
7.         <tr><td>資料 3-1</td><td>資料 3-2</td><td>資料 3-3</td></tr>
8.     </table>
9. </body>
10. </html>
```

圖十二. 簡單的 HTML 程式碼

從圖十二中可以發現其資料主要區塊為第 5 到第 7 行，其頭與尾分別為第 4 行的 `<table width = "50%" border="1">`與`</table>`。我們便是依此告知 Agent 何時開始抓取資料及何時停止抓取資料，且停止資料的條件是處於抓取資料的狀態。將資料抓取出來後，再利用字串的處理演算法，將其重要資訊截取出來，並存放在二維的 Vector 之中，這樣只要分別將 Vector 中的資訊一一取出，並以 XHTML 的形式顯示出來即可。其程式流程的 pseudo code 如圖十三所示。



```

1.  DECLARE stream is InputStream from THU-WebServer.
2.  DECLARE vector is 2-dimesion Vector.
3.  WHILE stream is not end
4.      Read a line from stream.
5.
6.      IF this line is equals to "<table width = "50%" border="1">" Then
7.          Set the agent ON
8.          Continue this loop
9.      ENDIF
10.
11.     IF agent is ON AND this line is equals to "</table>" THEN
12.         Break this loop
13.     ENDIF
14.
15.     IF agent is ON THEN
16.         Filter the important information to put into in Vector
17.     ENDIF
18. ENDWHILE

```

圖十三. 簡單 Agent 的 pseudo code

## 2.3. 向量地圖之繪製

雖然目前有多家知名公司以 Ajax 方式，免費提供 API 供使用者將自己專用之地圖嵌入至網頁中，然，目前手機瀏覽器普遍不支援 javascript。故，本系統中的向量地圖是利用向交通部運輸研究所所購買之地圖檔 (shape file)，再以本團隊自行開發之 java 程式繪製地圖，並將之以圖片格式儲存，以支援手機瀏覽器支瀏覽，以下描述地圖檔之格式，以為顯示地圖之基礎。

### 2.3.1. 地圖檔

一個地圖檔由主要檔 (main file, 副檔名為 shp)，索引檔 (index file, 副檔名為 shx) 和一資料庫檔 (dBase table, 副檔名為 dbf) 所組成，茲敘述如下：

- (1) **主要檔 (.shp)**: 用以儲存一個電子地圖中的各個圖形，一個圖形以一筆紀錄 (record) 記錄之，每個圖形是由一些點 (vertices) 所組成，因此，一筆紀錄包含了數個點，也因此，每筆紀錄之長度可能不相同
- (2) **資料庫檔 (.dbf)**: 用以儲存每筆紀錄的屬性，包含道路名稱及道路種類 (如國道、縣道、快速道路及普通道路等)，如圖十五所示，所儲存之每一個屬性都會對應到主要檔中的一筆紀錄，以標示該筆紀錄所對應之圖形種類。
- (3) **索引檔 (.shx)**: 儲存主要檔中每筆紀錄的起始位置 (offset)，俾在呈現圖形時，顯示程式得以知悉圖形的各組成點在何處。

178	三豐路	台13	3
179	三豐路	台13	3
180	三豐路	台13	3
181	三豐路	台13	3
182	三豐路	台13	3
183	豐勢路	台3	3
184	豐勢路	台3	3
185	三豐路	台13	3
186	三豐路	台13	3
187	豐勢路	台3	3
188	豐勢路	台3	3
189	豐勢路	台3	3

圖十四. 東勢鎮街道的 dbf 檔案部分內容，台 3 代表台三線省道

### 2.3.2. 主要檔格式

每一個主要檔包含了一長度為100 bytes的檔案標頭 (file header) 和數個長度不同的紀錄，每筆紀錄又分為一固定長度的紀錄標頭 (record header) 和不同長度的紀錄內容 (record content)。表1說明了檔案標頭的各個欄位 (field)、位置 (position)、類型 (type) 和位元組順序 (byte order, big-endian 或 little-endian)，其中位置代表該欄位的起始位置，是以16 bit為基本長度單位。欄位為地圖種類 (field = shape type, 起始位置為Byte 32) 用以定義此地圖檔屬於何種類型 (點、線、折線、多邊形、...)，其下方為組成該圖形之各點座標。

每個紀錄標頭的長度固定為8 bytes，而包含了兩個欄位：紀錄編號 (record

number) 和內容長度 (content length)，如表二所示，其中內容長度欄位代表此筆紀錄的長度(也是以16 bits為單位)，因此，每筆紀錄的長度為8\*8 bits + 內容長度\*16 bits。

### 2.3.3. 索引檔格式

地圖檔的索引檔格式為\*.shx，由檔案標頭(如表一所示)，和 record (如表二所示) 所組成。前者之資料結構與主要檔(\*.shp)者相同，而 \*.shx 檔案第 ” k” 筆紀錄記錄了 \*.shp 檔案第 ” k” 筆紀錄的位移量和內容長度。位移量的值記錄了\*.shp 當中所對應的紀錄標頭起始位置座標。內容長度則和 \*.shp 裡的紀錄標頭值相同。由於位移量和內容長度也是用 16 bits 為單位，因此，第一筆資料的位移量應該是檔案標頭的長度 50 (即 100 bytes)。

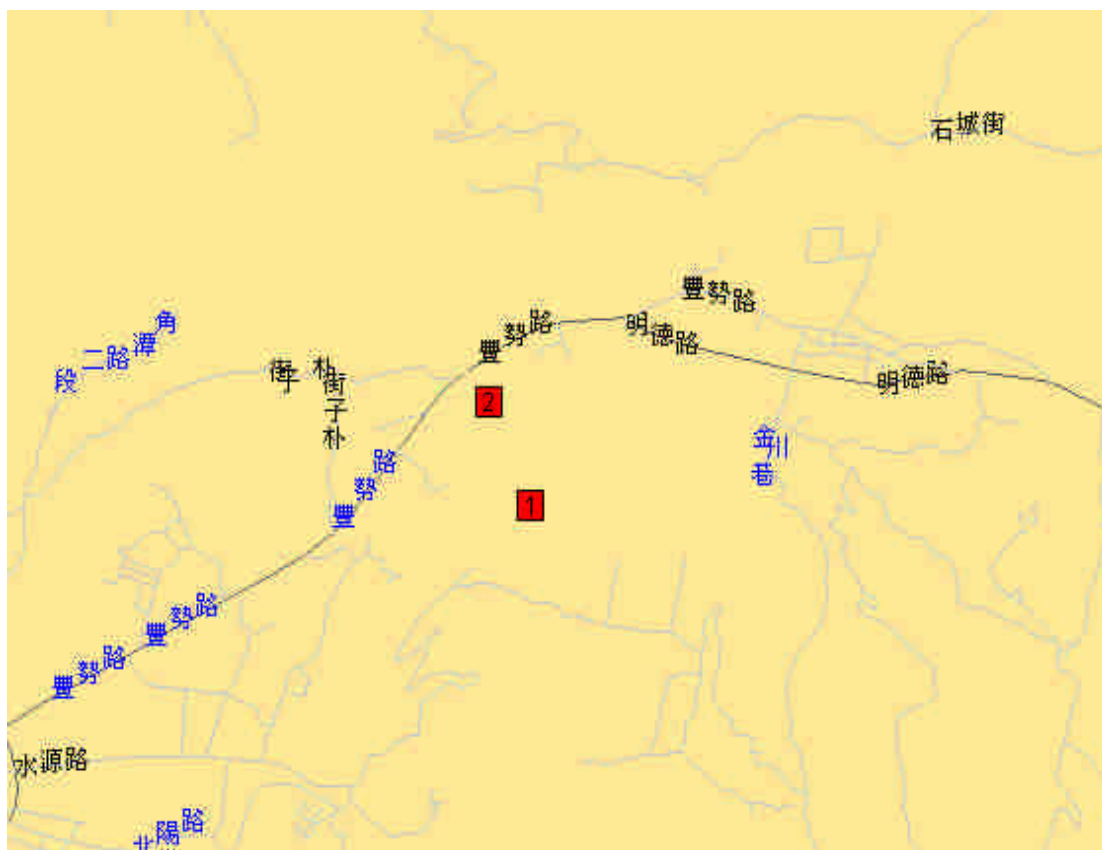
表二. 檔案標頭欄位說明

Position	Field	Value	Type	Byte Order
Byte 0	File Code	9994	Integer	Big
...				
Byte 24	File Length	File Length	Integer	Big
Byte 28	Version	1000	Integer	Little
Byte 32	Shape Type	Shape Type	Integer	Little
Byte 36	Bounding Box	Xmin	Double	Little
Byte 44	Bounding Box	Ymin	Double	Little
.	.	.	.	.
.	.	.	.	.
.	.	.	.	.
.	.	.	.	.

\* Unused, with value 0.0, if not Measured or Z type

表三. record 欄位

Position	Field	Value	Type	Byte Order
Byte 0	Record Number	Record Number	Integer	Big
Byte 4	Content Length	Content Length	Integer	Big



圖十五. 東勢部分道路地圖，其中 1 和 2 表示連結了該地點之相關資訊，使用者可以滑鼠點選之，或由 PDA/手機之按鍵輸入 1 或 2，俾瀏覽該地點之連結資訊

### 2.3.4. 繪圖程序

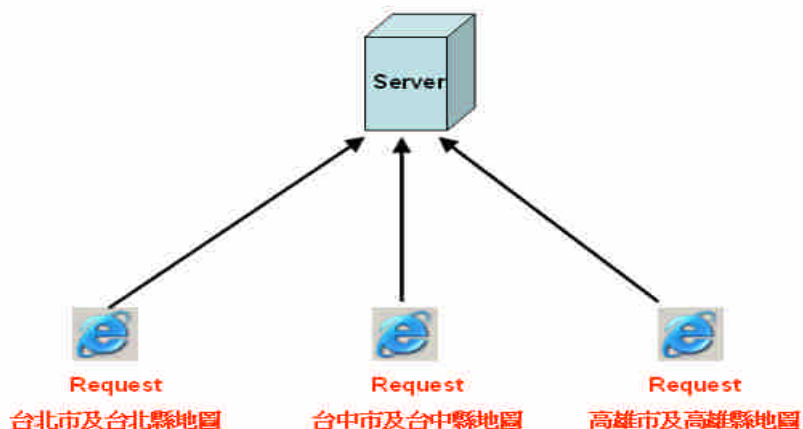
我們將台灣地圖分為“台灣輪廓”和“各縣市道路”兩個圖層，描繪地圖時先畫出台灣輪廓，再繪出各縣市地圖，步驟為：先讀\*.shx 檔案，以取得每筆紀錄的起始位置(starting address)，再取主要檔之紀錄，而依序將每筆紀錄繪出，之後再從資料庫檔讀取該筆紀錄的屬性而標示在地圖上。其後，將該向量地圖以圖片方式儲存，並使用 SQL 語法，查詢行動資訊資料庫以檢查該地圖範圍內是否有照片拍攝地點可以呈現，若有，則在該圖片上標註拍攝地點，如圖十六的點“1”和“點 2”，之後使用者可點選數字或由 PDA/手機之按鍵輸入 1 或 2，俾透過瀏覽器瀏覽該點所連結之網頁。

### 2.3.5. 動態載入縣市道路圖及系統穩定性

受限於 Java VM，每個 java 應用程式所用之記憶體大小皆有一定之限制，若將地圖檔中所有的紀錄同時載入記憶體，則會耗盡記憶體（out of memory）而導致程式無法繼續執行，甚至當機。因此，本系統採取“動態載入”方式，當使用者瀏覽“全台灣地圖時”，顯示程式只繪出國道和縣市區塊，而暫時不描繪縣市道路。只有當使用者放大（zoom in）到只瀏覽某個縣市時，程式才會載入縣市道路，並將之繪出，這就是所謂『瀏覽何處，讀取何處』（where to browse which record to load）的概念。在 PC 或 Notebook 中顯示地圖時，均有此功能。但手機上，則需自行開發程式完成之。目前該功能已完成。再者，限於手機之記憶體容量，當使用者的瀏覽對象從某個縣市持續移動，而進入另一個縣市時，顯示程式則會先釋放之前所載入的縣市道路紀錄所佔用之記憶體，再載入新的縣市道路紀錄，目的都是避免耗盡記憶體及當機。

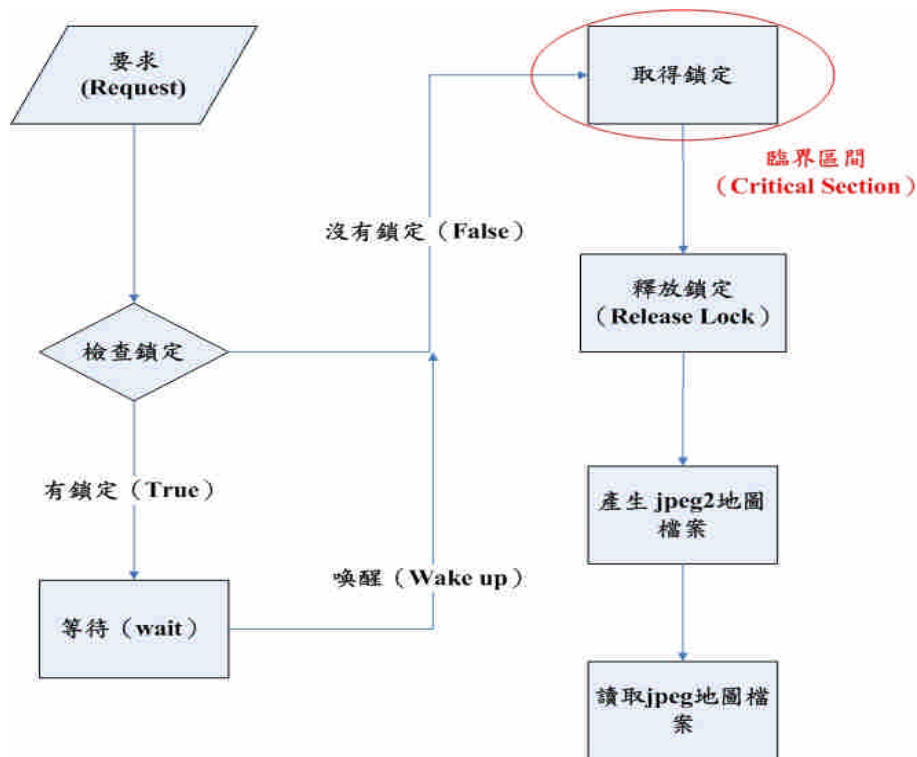
### 2.3.6. 以共享記憶體（share memory）繪圖與系統穩定性

由於繪圖程式需讀取地圖檔中之紀錄，並以大量記憶體儲存，再以網頁方式顯示於螢幕上，若是網頁伺服器為每位使用者各開啟一支繪圖程式，以描繪其所欲瀏覽的地圖。萬一各使用者所要瀏覽的地點（即縣市）均不同，以圖十七為例，伺服器將依要求同時載入台北縣市、台中縣市及高雄縣市的地圖，如此，將會加大伺服器負載（loading），而導致效能（performance）降低，甚至當機。



圖十六. 三個使用者同時對伺服器要求載入不同縣市的地圖

為避免此一現象，我們的作法是：所有連上網頁伺服器的使用者共用一支繪圖程式(又稱為 **Reentrant program**)，意即，將繪圖程式釋放及載入紀錄以顯現地圖的程式段落視為臨界區間 (critical session)，在每位使用者，例如，甲，要求繪製地圖時，繪圖程式會先檢查自己目前是否正處於臨界區間內處理某位使用者，例如，乙，之地圖。若是，則甲必須進入等待 (wait) 狀態，直到繪圖程式離開臨界區間，才會為甲繪製地圖；反之，則繪圖程式直接進入臨界區間描繪地圖，並將所產生之地圖回傳給該使用者瀏覽，其流程如圖十七所示。



圖十七. 以共享記憶體方式共用繪圖程式之流程，其中，jpeg 檔案即係地圖檔案

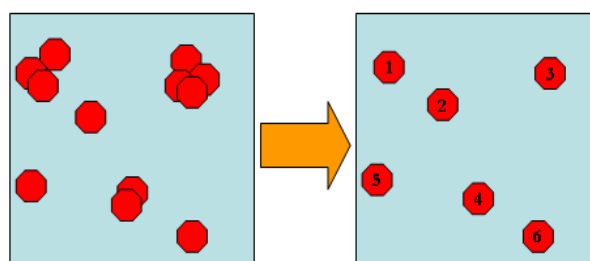
### 2.3.7. 採用 K-Means 演算法分群

手機共有 0~9、\*、# 等 12 個按鍵，其中，本系統以其中 6 個按鍵 \*、#、8、0、7、9，分別當作瀏覽地圖時之放大、縮小、上移、下移、左移、右移功能鍵，如圖十八所示，剩餘六個按鍵則做為圖片的連結編號。因此，當地圖上可呈現相

片的地點大於六時，勢必會有地點無法以手機按鍵點選，且受限於手機螢幕解析度及大小，若在螢幕上顯示過多連結相片的點，各點將無法清楚地呈現。為解決此一問題，本系統在顯示相片地點之前，先以 K-Means 演算法（意即分成 K 群而有 K 個平均值之意），將所有要呈現的相片之地點依距離遠近最多分成六群，再予以顯示，圖十九為其示意圖。若所點選之編號是由若干個點所組成，例如，圖中之 1、3 及 4 均如此，假設使用者點選 1，螢幕上則自動放大到足以呈現原先組成“1”的三個點的大小，並重新分群顯示在螢幕上之點，其結果如小於等於 6，則進行編號，並顯示之。若仍大於 6，則再以 K-Means 演算法分群之。



圖十八. 手機按鍵圖



圖十九. 分群概念圖

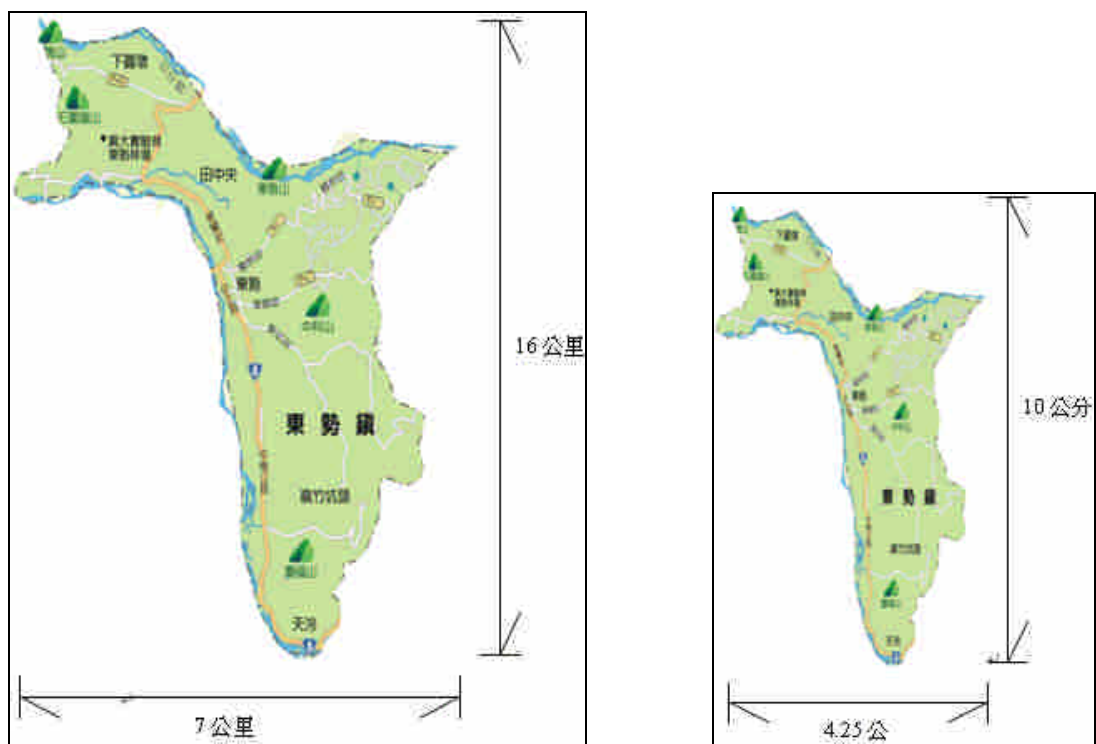
### 2.3.8. 地圖放大縮小

藉由地圖上所標示之照片地點，使用者能對地圖進行放大（Zoom in）與縮小（Zoom out）的操作。在使用者選擇放大時，系統會繪製更多的地圖細部元件，例如，道路與河流等，以呈現更清楚、更詳細的地理資訊；若選擇縮小，本系統則只呈現該地點附近之概略資訊，目的是讓使用者對整體環境有更清楚的概觀。方法上，本團隊採用簡易且快速之圖形相對比例縮放法，茲敘述如下：

圖二十中，左邊圖形所標示之長寬為原地物實際大小，而右邊為欲呈現給使用者之大小。此二圖形之長寬各自之相對比例分別為“3940000:1”與“2880000:1”，意即，所呈現圖形之長寬若各為 1 公分，代表實際長 39.4 公里，寬 28.8 公里。這就是我們的基本長寬比例，假設我們預定每次按下放大連結鍵後，新地圖所呈現之比例係將舊圖放大 10%，故放大一次，我們會得到新的長度



比例“ $3940000*(1-0.1):1$ ”及寬度比例“ $2880000*(1-0.1):1$ ”，其餘依此類推，便可算出欲讀取之地圖長寬，例如，我們上傳照片的位置座標之經緯度為(120, 120)，在所呈現之大小固定為 10 公分情況下，當我們按下放大連結時，本系統所呈現的地圖實際上便是以座標(120, 120)為中心點，長  $354.6(=394*0.9)$  公里為長，寬  $259.2(=288*0.9)$ 公里為寬所圍出的長方形區域。而在 PC 版之 GIS 中均提供地圖大小之縮放功能，唯手機中之放大縮小皆須由程式開發者為之，目前亦已開發完成。



(a) 原圖大小

(b) 顯示在螢幕上之大小

圖二十. 相對比例縮放



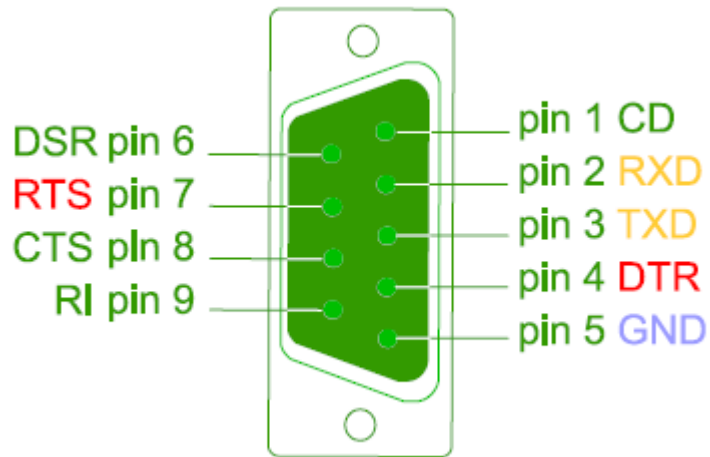
### 3. 洪災偵測(程式碼詳見附錄三)

對於山洪暴發之偵測方面，我們係在河川上下游各架設水位感測器以偵測河川水位，當上水位變高了，而下游仍在低水位，則應係山洪暴發，此時應即刻警告下游民眾迅速撤離河面或河邊。另外，當河川水位超過警戒時，可能危害建構於河川上方之橋樑的安全，此時應於橋兩側自動發佈警訊，甚至禁止人車通行。偵測方式亦可以裝設於橋下之偵測器為之，並自動啟動警報器。

唯，資訊系統偶而會有誤報情形，因此如圖二十二所示，我們係於河川上下游各裝設三套偵測器，以兩個以上之偵測情形為準。上游兩個以上皆在高水位，下游兩個以上皆在低水位，則表示上游山洪暴發，採用三套之目的則是提高系統之可靠性，且只增加兩個四個小連進，成本低。而橋樑下之水位太高，除了行人之目視處，亦可以圖二十二中相同的兩個偵測器(例如，兩個上游者，或兩了下游者)為之。安裝在同一橋樑之兩個水位偵測器(位置應在接近河川警戒線位置)，同時偵測到高水位，則表示橋樑已危險，且河川水流湍急，應在橋樑兩端亮出警示牌(燈)，警告民眾勿下水，或不要強行通過。偵測器則宜安置在橋樑靠下游側，避免被漫溢水面之漂流物砸壞。

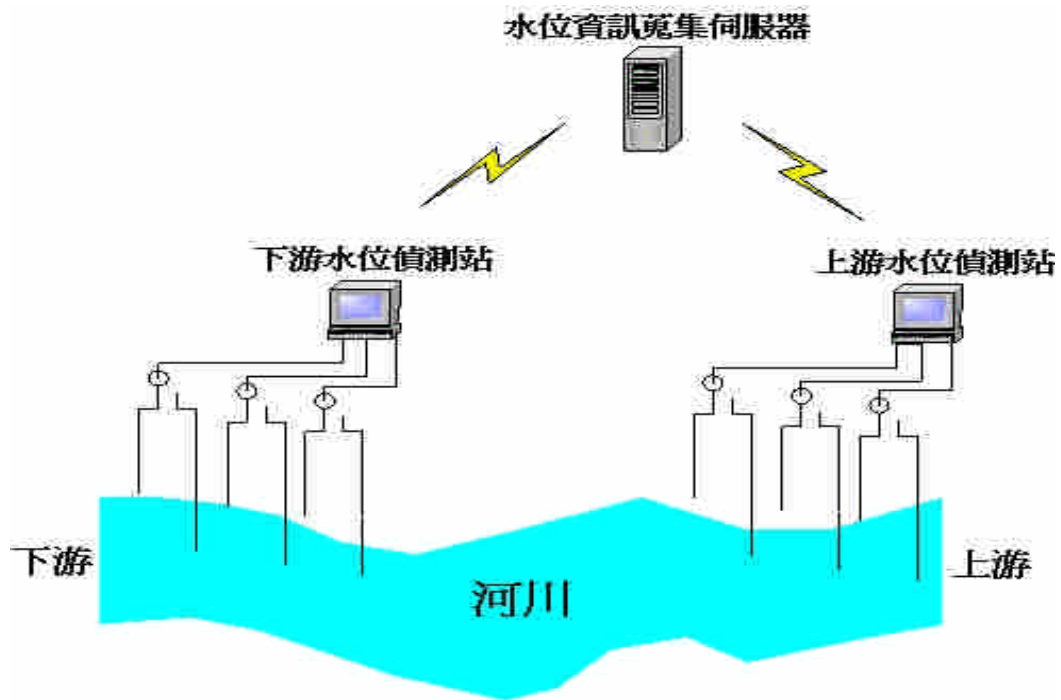
#### 3.1. 水位偵測站設計原理

RS-232 是美國電子工業聯盟所制定之串行數據通信的接口標準，全稱是 EIA-RS-232 (簡稱 232, RS232)。它被廣泛用於計算機串連外接設備。RS-232 指定了 9 個不同的信號連接，而由 25 個 D-sub (微型 D 類) pin 腳構成了 DB-25 連接器。很多設備只是用了其中的一小部分 pin 腳，基於節省經費和空間的考慮不少機器採用較小的連接器，特別是 9 pin 腳的 DB-9 型連接器，其 pin 腳說明如圖二十一。



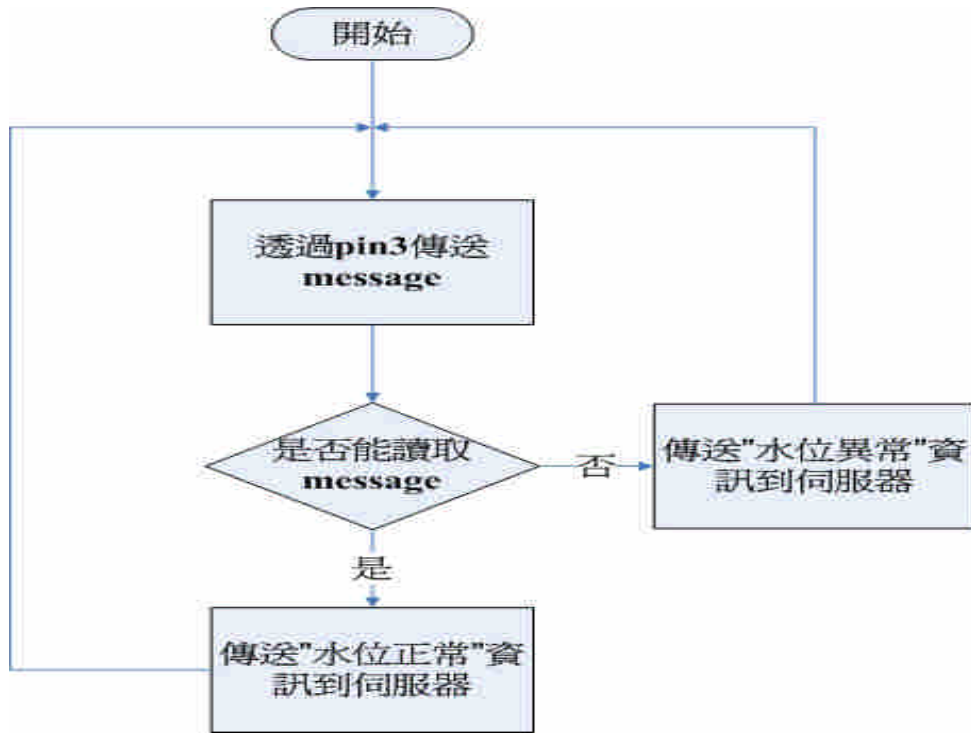
圖二十一. RS232 pin 腳說明

- (1) Pin1 : CD (carrier detect)
- (2) Pin2 : RXD (receive)
- (3) Pin3 : TXD (transmit)
- (4) Pin4 : DTR (data terminal ready)
- (5) Pin5 : GND (ground)
- (6) Pin6 : DSR (data set ready)
- (7) Pin7 : RTS (request to send)
- (8) Pin8 : CTS (clear to send)
- (9) Pin9 : RI (ring indicator)



圖二十二. 水位偵測原理示意圖

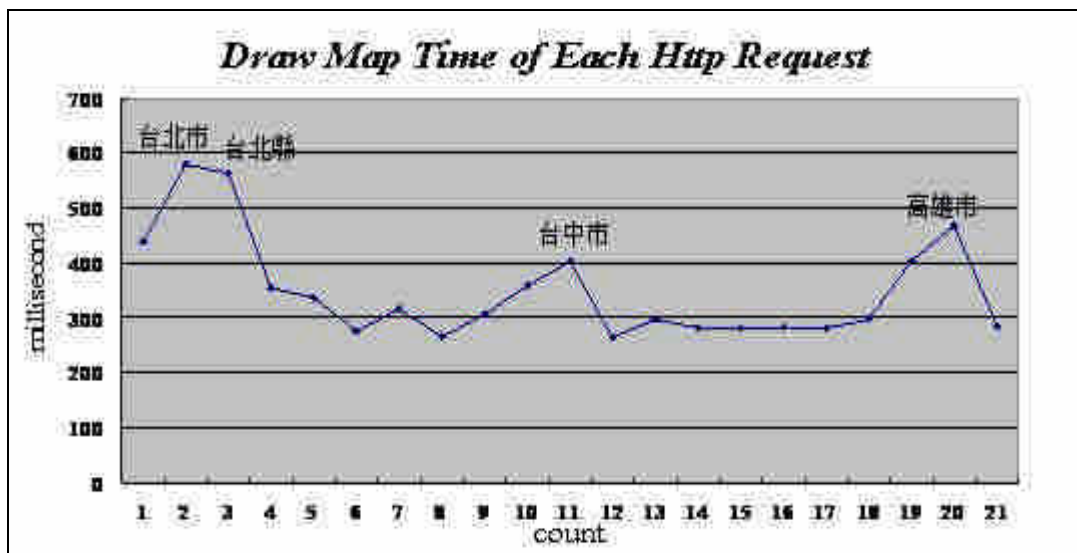
水位偵測工作站主要利用 RS232 的 pin3 傳送訊號，再透過 pin2 讀取該訊號，如圖二十二所示，當水位正常時，只有 1 個 pin 腳接觸河川水面，或甚至接觸不到水面，因此，pin2 無法收到訊號，工作站所讀取之信號為 0，當水位暴漲促使兩個 pin 腳全接觸河川水面時，會因水具備導電能力而使信號得以傳遞，此時，工作站可藉由 pin2 讀取其透過 pin3 所傳遞之信號，故得以偵測水位高漲，其偵測流程如圖二十三所示。



圖二十三. 水位偵測流程

### 3.2. 系統之效能評測

#### 3.2.1. 網頁地圖繪製



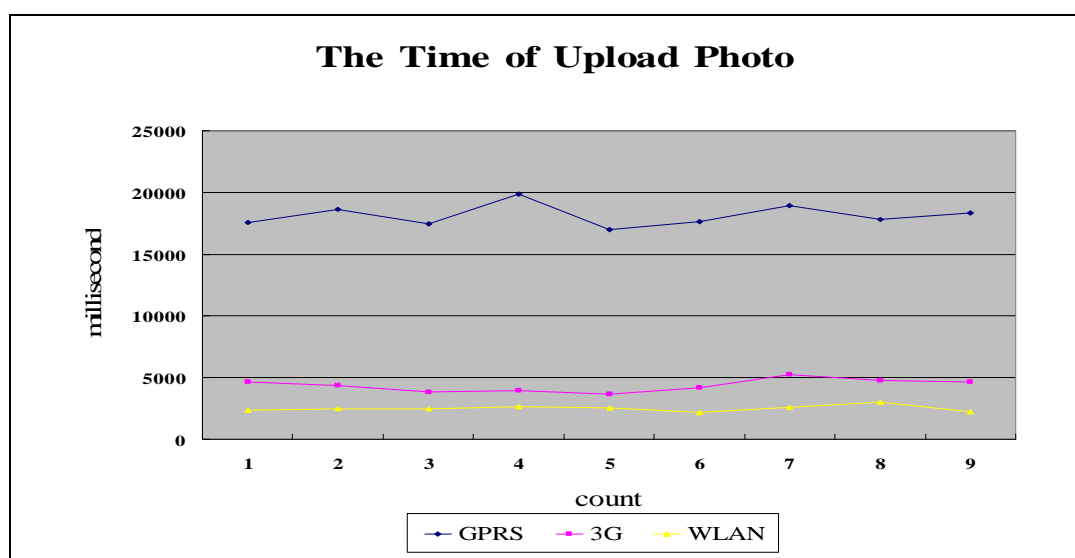
圖二十四. 網頁地圖繪製時間效能評測圖

地圖繪製為本計畫研發系統中最主要的功能之一，其架構堪稱是所有子系統

中最複雜及最困難的系統，尤其，網頁地圖的繪製更是結合了各種技術(如 K-Means 分群演算法、Share Memory、Dynamic Loading.....等)以突破層層窘境。由於地圖繪製子系統中扮演極為重要之角色，本實驗主要是針對其效能加以分析、評估，方法是紀錄地部落格伺服器接收一個 Http 要求(request)時到地圖繪製完成所經過之時間，單位為毫秒(milliseconds)。

實驗結果如圖二十四實驗數據所示，其最大值為 579 毫秒，最小值為 265 毫秒，平均為 364 毫秒。以平均時間而言，足以證明本系統之即時性及效能的優越。然，更細微的觀察，卻發現前 6 次實驗之平均繪製時間高達 474.3333 毫秒，與總平均相時間相差甚遠，主要原因在於本實驗要求地圖繪製之區域是從北台灣依序往南部地區繪製，即第一次 request 要求繪製概略之台灣輪廓(可參考圖 21.)，第二次則要求繪製台北市，第三次為台北縣，第四次為桃園懸.....。由於北台灣道路繁雜，繪製大量道路與路名等費時長，繪製台中與高雄等都會區域大都市時(第九次與第 20 次)也因而耗費高於平均時間。

### 3.2.2. 手機照片上傳至地圖部落格伺服器時間延遲



圖二十五. 手機上傳照片時間實驗數據

表四. 各無線網路技術資料表

網路介面	涵蓋範圍	頻寬
GPRS (2.5G)	極大	20~40k
WCDMA(3G)	極大	384k / 128k
WiFi 802.11b	數十公尺	11M

圖二十五為手機透過三種傳輸介面傳送一 123k 大小之照片至地圖部落格伺服器之時間，有表四可知各傳輸介面之頻寬理論值，由實驗數據(傳輸時間)可各傳輸介面之實際頻寬遠小於理論值，加上手機處理器效能有效也因此也增加了一些處理延遲(process delay)。然，就結果而論，除透過 GPRS 上傳照片外，透過 3G 或 WLAN 的效率雖無法達到理論值，但不至於差強人意。

#### 4. 資料庫建置

在協助各子計畫從事資料庫之設計方面，擬依其標準設計程序為之，原則上分成資料部分及查詢功能部分。前者將從需求蒐集開始，經分析及了解各子計畫所需項目及屬性(attributes)後，再予以正規化(normalized)，以避免未來發生資料之異常現象，包括 insertion, deletion 及 update anomalies 等。其次，依各子計畫所需之查詢需求，發展 stored procedures，以便於後續之查詢。表五為所得知子計畫所需之資料庫建置項目，然，目前該二子計畫尚未有成果，因此資料庫是備而不用。

表五. 子計畫一及子計畫三所需之資料庫建置項目

子計畫	資料庫建置項目
子計畫一	<ol style="list-style-type: none"> <li>1. 災害歷史資料</li> <li>2. 社區資源與資本</li> <li>3. 社區環境資料，及其空間實體性質，與現象</li> <li>4. 風險管理及評估能力，災害空間分析與監控能力</li> <li>5. 政府因應災害的組織及互動</li> <li>6. 社區因應能力</li> <li>7. 社區居民因的復原力</li> <li>8. 社區居民的壓力承受度</li> <li>9. 災區人口結構，家庭形式，職業組成</li> <li>10. 聚落分佈形成，居民居住及休閒習慣，及產業分佈</li> <li>11. 關係網絡</li> <li>12. 社會資本量測</li> </ol>
子計畫三	<ol style="list-style-type: none"> <li>1. 東勢、和平之災害類型，在後壓力，人口結構，經濟狀況</li> <li>2. 問卷調查</li> <li>3. 防災資源</li> <li>4. 韌性量表題庫</li> <li>5. 質性資料</li> <li>6. 種子成長營會議記錄</li> </ol>

## 5. 結論

- 1 台灣受限於地形及氣候等先天因素，導致每年天然災害不斷，對人民生命財產造成相當大之威脅。由於，天然災害的發生突然，往往讓人民措手不及，因而錯過逃難的黃金時間。透過本子計畫之 CEMS 系統，長期監控、蒐集及分析社區環境 CEMS 能在社區環境品質不佳或天然災害發生前或發生時，通知相關管理人員，採取因應措施及發佈警告訊息通知社區居民，如此，可將這些環境因素所造成之傷亡減至最低。
- 2 天然災害往往帶來嚴重的災情，以納利風災為例，在民宅受損方面，以汐止、瑞芳、貢寮、平溪、雙溪、新店等地區洪水挾帶土石沖刷，造成不少山區崩塌，有廿五幢民宅遭毀損。災害後的重建工程著實耗費大量的人力及物力，透過本子計畫之 CEMS 可事先知悉將有災害發生，居民可事先防範，以減少災情之嚴重性，更可以降低事後災區重建所耗費之成本。
- 3 東勢地區之中崙溪便常發生這種事，防不勝防。歷年來因而傷亡的人數不在少數。另一種情況是下大雨，河川水位高過警戒線，颱風時尤其嚴重。此時，會有民眾不顧生命危險到河邊撿取上游沖下來之家禽畜或木材。曾於民國九十六年十二月二十一日訪問東勢鎮民代表，也是中崙社區發展協會理事長 張裕通先生。其對此河川洪峰之預警系統構想十分贊同，亦告知該地區迫切需要，因該社區便曾多次山洪爆發而損失多位居民，尤其是兒童。
- 4 配合本系統之 GIS，除了應用於災害監控外，亦可應用於社會問題與社會福利等方面，結合資料探勘與 GIS 等資訊科技技術，可以從更廣泛的角度分析其他子計畫所作問卷所獲得之資料，以擴大研究視野。



## 6. 參考文獻

- [1] 成功大學水工試驗所, <http://gis.thl.ncku.edu.tw/coordtrans/coordtrans.aspx>
- [2] 台灣大學建築與城鄉研究所,  
<http://www.bp.ntu.edu.tw/WebUsers/ftlin/course/gis/GIS%E5%9F%BA%E6%9C%AC%E6%A6%82%E5%BF%B5.doc>
- [3] 內政部地政司衛星測量中心,  
<https://www.gps.moi.gov.tw/satellite/cpQuery/cpQuery.jsp>
- [4] 逢甲大學測量, <http://st2.fju.edu.tw/~a9074173/hw5-1.htm>
- [5] 米私啄與糠砒悠忒, <http://www.minstrel.idv.tw/post/1>
- [6] NMEA 2.0, <http://www.kh-gps.de/nmea-faq.htm>
- [7] 高馳實業股份有限公司, <http://www.ydcrider.com/garmin/theory.html>
- [8] 電子工程專輯, <http://www.eettaiwan.com/>
- [9] 台灣通訊(ACE), <http://www.arconet.com.tw/taiwanelecom/>
- [10] PaloWireless, <http://www.palowireless.com/>
- [11] MSDN Library for Bluetooth,  
<http://msdn.microsoft.com/library/default.asp?url=/library/en-us/wcebluet/html/cmconBluetooth.asp>
- [12] Wireless Knowledge Base,  
<http://wireless.ittoolbox.com/nav/t.asp?t=426&p=426&h1=426>
- [13] OpenNETCF.org ,  
<http://www.opennetcf.org/CategoryView.aspx?category=Home>
- [14] Pocket PC Develop Network, <http://www.pocketpcdn.com/>
- [15] Dll-files.com , <http://www.dll-files.com/dllindex/index.shtml>
- [16] W3C, <http://www.w3schools.com/default.asp>

- [17] F.-Y. Leu, and H.-H. Tsai, Metadata-based Mobile Service Platform for Next-Generation Heterogeneous Networks, Tung Hai University, Master Thesis, July 2007.
- [18] 3GPP. Technical Specification TS 23.002, v5.0.0: Network Architecture (Release 5), October 2000.
- [19] 3GPP2 TSG-P. Wireless IP Network Standard. P.S0001-B (v1.0), October 2002.
- [20] H. Henrickson, and S. Hofmann, IIS 6 The Complete Reference, McGraw-Hill, 2003.
- [21] M. Tulloch, IIS 6 Administration, McGraw-Hill, 2003.
- [22] R. Morimoto, M. Noel, O. Droubi, K. Gardinier, and N. Neal, Microsoft Windows Server 2003 Unleashed, Pearson Education, April 2003.
- [23] P. DuBois, MySQL Cookbook, O'Reilly, October 2002.
- [24] J. Brittain, and Ian F. Darwin, Tomcat The Definitive Guide, O'Reilly, June 2003.
- [25] B. Albahari, P. Drayton, and T. Neward, CSharp In A Nutshell, O'Reilly, August 2003.
- [26] Jeffrey E. F. Friedl, Mastering Regular Expressions 2nd Edition, O'Reilly, July 2002.
- [27] A. Wigley, S. Wheelwright, R. Burbidge, R. MacLeod, and M. Sutton, Microsoft .NET Compact Framework (Core Reference)", Microsoft Press, January 2003
- [28] Microsoft Corporation, "Windows Server 2003 Deployment Kit", Microsoft Press, May 2005
- [29] Sun Microsystems, Inc., Java SE 5.0 API Documentation
- [30] ALOV Map/TMJava Documentation. <http://www.alov.org/index.html>  
OpenNETCF Smart Device Framework Documentation.  
<http://www.opennetcf.org/CategoryView.aspx?category=Home>
- [31] TinyOS. <http://www.tinyos.net/>
- [32] Wiki – QRCode. [http://en.wikipedia.org/wiki/QR\\_Code](http://en.wikipedia.org/wiki/QR_Code)
- [33] Wiki – LBS. [http://en.wikipedia.org/wiki/Location-based\\_service](http://en.wikipedia.org/wiki/Location-based_service)

- [34] Sun Development Network(SDN). <http://developers.sun.com/>
- [35] MSDN Library. <http://msdn2.microsoft.com/zh-tw/default.aspx>
- [36] GIS Lounge. <http://gislounge.com/>
- [37] Wiki – GIS. [http://en.wikipedia.org/wiki/Geographic\\_information\\_system](http://en.wikipedia.org/wiki/Geographic_information_system)
- [38] 施威銘研究室, “Microsoft Windows Server 2003 R2 架站實務”, 旗標出版,  
2006年07月27日

## 附錄

### 1. GICS

#### 1.1. 掃描藍芽裝置

```
private void detectBtDevice()
    {
        OpenNETCF.Net.Sockets.BluetoothClient b = new BluetoothClient();
        this.detectedDeviceInfo = b.DiscoverDevices();
        if (this.detectedDeviceInfo.Length != 0)
        {
            this.btDeviceName = new string[detectedDeviceInfo.Length];
            for (int count = 0; count<detectedDeviceInfo.Length;
count++) )
                {
                    try
                    {
                        this.btDeviceName[count] =
detectedDeviceInfo[count].DeviceName;
                    }
                    catch (Exception)
                    {
                        this.btDeviceName[count] = "get device name error!!!";
                    }
                }
            this.Invoke( new sss(ttt) );
        }
        else
        {
            changeToHmMenu = true;
            this.Invoke( new sss( uiSetting ) );
        }
        b.Close();
        return;
    }
}
```

#### 1.2. 連接 GPS

```

private void connectToGPS()
    {
        btClient = new BluetoothClient();
        this.btEndPoint = new
BluetoothEndPoint(this.ConnectTargetInfo.DeviceID, new
Guid("{00001101-0000-1000-8000-00805f9b34fb}"));
        try
        {
            btClient.Connect(btEndPoint);
            MessageBox.Show("已連接" + this.ConnectTarget + "...", "訊息",
            MessageBoxButtons.OK, MessageBoxIcon.Asterisk,
            MessageBoxDefaultButton.Button1);
            this.menuItem17.Enabled = true;
            this.menuItem10.Enabled = false;
        }
        catch (System.Net.Sockets.SocketException)
        {
            MessageBox.Show( "無法連接..請確認目標裝置是否為GPS...",
            "訊息", MessageBoxButtons.OK, MessageBoxIcon.Exclamation,
            MessageBoxDefaultButton.Button1 );
        }
        this.btDeviceListPanel.Visible = false;
        this.formPictureBox.Visible = true;
        this.Menu = this.homePageMenu;
        this.Focus();
    }
}

```

### 1.3. 讀取 GPS 資料

```

private string rcvGpsData()
    {
        Stream gpsStream = btClient.GetStream();
        byte[] buffer = new byte[1024];
        int gpsData = gpsStream.Read(buffer, 0, 1024);
        String str = "";
        str += System.Text.Encoding.ASCII.GetString(buffer,0,1024);
        gpsData = gpsStream.Read(buffer, 0, 1024);

        return str;
    }
}

```

```
}
```

## 1.4. 中斷 GPS 連線

```
private void menuItem17_Click(object sender, EventArgs e)
{
    menuItem10.Enabled = true;
    menuItem17.Enabled = false;
    this.btClient.Close();
    MessageBox.Show("已中斷與" + this.ConnectTarget + "的連線....", "
訊息",
    MessageBoxButtons.OK, MessageBoxIcon.Asterisk,
    MessageBoxDefaultButton.Button1);
}
```

## 2. PGIS

### 2.1. 使用者解析度判別

```
public Dimension getResolution(HttpServletRequest request) {
    int w = 128;
    int h = 200;
    String s = "";
    // WinNT
    s = request.getHeader("user-agent");
    if (s.indexOf("Windows NT") >= 0) {
        w = 640;
        h = 480;
    }
    // WinCE
    s = request.getHeader("ua-pixels");
    if (s != null) {
        String[] tmp = s.split("x");
        w = Integer.parseInt(tmp[0]);
        h = Integer.parseInt(tmp[1]);
    }
    // Panasonic, Benq
    s = request.getHeader("x-up-devcap-screenpixels");
    if (s != null) {
        String[] tmp = s.split(",");
        w = Integer.parseInt(tmp[0]);
        h = Integer.parseInt(tmp[1]);
    }
}
```

```
}  
return new Dimension(w, h);
```

## 2.2. 網頁 Agent 技術

```
public class Parse{  
  
    public String getHtmlSrc( String conTarget ){  
        try{  
            URL url = new URL( conTarget );  
            BufferedReader bufReader = new BufferedReader( new  
InputStreamReader ( url.openStream() ) );  
            String content = "", tempStr = "";  
            while( ( tempStr = bufReader.readLine() ) != null ){  
                content += tempStr + "\n";  
            }  
            bufReader.close();  
  
            return content;  
        }  
        catch( Exception ex ){  
            return ex.getMessage();  
        }  
    }  
  
    public String parseHtml( String src, String re ){  
        Pattern pattern = Pattern.compile( re );  
        Matcher matcher = pattern.matcher(src);  
        String returnValue = "";  
        int matchCount = 0;  
        while( matcher.find() ){  
            returnValue += matcher.group() + "\n";  
            matchCount++;  
        }  
  
        return returnValue;  
    }  
  
    public String[] split( String target ){
```

```

        return target.split( "\n" );
    }

String sid = request.getParameter( "sid" );
String spage = request.getParameter( "spage" );
String conTarget = "http://member.ettoday.com/newsflash/";
conTarget += request.getParameter( "target" );
Parse parse = new Parse();
String htmlSrc = parse.getHtmlSrc( conTarget );
int startIndex = htmlSrc.indexOf( "<!--本文開始 -->" );
int endIndex = htmlSrc.indexOf( "<!--本文開始 -->", startIndex + 50 );
htmlSrc = htmlSrc.substring( startIndex, endIndex );
startIndex = htmlSrc.indexOf( "<table border=\"0\" align=\"right\">" );
endIndex = htmlSrc.indexOf( "</table>" );
String title = parse.parseHtml( htmlSrc, "(?<=<Title>.*(?=</Title>)" );
char[] charAry = htmlSrc.toCharArray();
for( int count=startIndex; count<=endIndex+8; count++){
    charAry[count] = ' ';
}
htmlSrc = new String( charAry );
startIndex = htmlSrc.indexOf( "<br>" );
startIndex = htmlSrc.indexOf( "<br>", startIndex + 10 );
htmlSrc = htmlSrc.substring( startIndex, htmlSrc.length() );

```

### 2.3. 向量地圖繪製

```

public class Shape
{

    public Shape()
    {
        xCoords = null;
        yCoords = null;
    }

    public Shape(double coords[])
    {
        xCoords = null;
        yCoords = null;
    }
}

```



```

int count = coords.length / 2;
xCoords = new double[count];
yCoords = new double[count];
int off = 0;
for(int i = 0; i < count;)
{
    xCoords[i] = coords[off];
    yCoords[i] = coords[off + 1];
    i++;
    off += 2;
}

}

public static void drawBoldLine(Graphics g, int x1, int y1, int x2, int y2)
{
    if(x1 < 0 && y1 < 0 && x2 < 0 && y2 < 0)
        return;
    int x[] = {
        x1, x2
    };
    int y[] = {
        y1, y2
    };
    g.drawPolyline(x, y, 2);
    if(x2 != x1)
    {
        float k = (y2 - y1) / (x2 - x1);
        if((double)Math.abs(k) > 2.7000000000000002D)
        {
            x1++;
            x2++;
        } else
        {
            y1++;
            y2++;
        }
    } else
}

```

```

    {
        x1++;
        x2++;
    }
    x[0] = x1;
    x[1] = x2;
    y[0] = y1;
    y[1] = y2;
    g.drawPolyline(x, y, 2);
}

```

```

private static void drawBoldPolyline(Graphics g, int x[], int y[], int n)
{
    if(n < 2)
        return;
    for(int i = 1; i < n; i++)
        drawBoldLine(g, x[i - 1], y[i - 1], x[i], y[i]);
}

```

```

private static void drawCustomLine(Graphics g, int x[], int y[], Symbol sym)
{
    int pointCount = x.length;
    double len = (double)sym.step / 4D;
    double k1 = 0.0D;
    double k2 = 0.0D;
    double k = 0.0D;
    double c = 0.0D;
    int iVis = 1;
    int iDots = 0;
    int m = 1;
    for(int i = 1; i < pointCount; i++)
    {
        int dx = x[i] - x[i - m];
        int dy = y[i] - y[i - m];
        double l = Math.sqrt(dx * dx + dy * dy);
        if(l > 0.0D)
        {

```

```

k1 = (double)dy / l;
k2 = (double)dx / l;
double dd1;
double dd2;
double dd;
if(dx == 0)
{
    dd1 = k1 * (double)sym.step;
    dd2 = (k1 * (double)sym.step) / 2D;
    dd = k1 * len;
    k = 0.0D;
} else
{
    k = (1.0D * (double)dy) / (double)dx;
    c = (double)y[i - m] - k * (double)x[i - m];
    dd1 = k2 * (double)sym.step;
    dd2 = (k2 * (double)sym.step) / 2D;
    dd = k2 * len;
}
int dx5 = (int)(k2 * 3D);
int dy5 = (int)(k1 * 3D);
double x0 = x[i - m];
double y0 = y[i - m];
double x1 = x[i - m];
double y1 = y[i - m];
double l2 = 0.0D;
do
{
    boolean bLine = (sym.style == 1 || sym.style == 3 ||
sym.style == 4) && iVis > 0 && iDots == 0;
    if(dx == 0)
    {
        y1 = y0 + dd;
        l2 = (y1 - (double)y[i - m]) / k1;
    } else
    {
        x1 = x0 + dd;
        y1 = k * x1 + c;

```

```

        l2 = (x1 - (double)x[i - m]) / k2;
    }
    if(l2 > 1 || l1 == 0.0D || l2 <= 0.0D)
    {
        if(bLine)
            if(sym.size == 2)
                drawBoldLine(g, (int)x0, (int)y0, x[i], y[i]);
            else
                g.drawLine((int)x0, (int)y0, x[i], y[i]);
        break;
    }
    if(bLine)
    {
        if(sym.size == 2)
            drawBoldLine(g, (int)x0, (int)y0, (int)x1, (int)y1);
        else
            g.drawLine((int)x0, (int)y0, (int)x1, (int)y1);
        iVis = 0;
    } else
    {
        iVis++;
    }
    x0 = x1;
    y0 = y1;
    if(sym.style == 5)
        g.drawLine((int)(x0 - (double)dy5), (int)(y0 +
(double)dx5), (int)(x0 + (double)dy5), (int)(y0 - (double)dx5));
    else
    if(sym.style == 2 || iDots == 1 || sym.style == 4 && iDots ==
2)
        if(sym.size == 2)
            g.fillRect((int)x0 - 1, (int)y0 - 1, 2, 2);
        else
            g.fillRect((int)x0, (int)y0, 1, 1);
    if(sym.style == 3 && iDots < 2 || sym.style == 4 && iDots <
3)
    {
        iVis = 0;

```

```

        iDots++;
        len = sym.step / 2;
        dd = dd2;
    } else
    {
        iDots = 0;
        len = sym.step;
        dd = dd1;
    }
    } while(true);
    len = 12 - 1;
    m = 1;
} else
{
    m++;
}
}

}

protected static void drawPoint(Graphics g, Symbol sym, int x0, int y0)
{
    Image img = sym.image;
    int x;
    int y;
    if(img != null)
    {
        x = img.getWidth(null);
        y = img.getHeight(null);
        if(x < 0)
            x = 0;
        if(y < 0)
            y = 0;
        x = x0 - x / 2;
        y = y0 - y / 2;
        g.drawImage(img, x, y, MapUtils.getGlobalImageObserver());
        return;
    }
}

```

```

int s = sym.size;
x = x0 - s / 2;
y = y0 - s / 2;
switch(sym.style)
{
case 1: // '\001'
    if(sym.fill)
    {
        g.setColor(sym.fillColor);
        g.fillRect(x, y, s, s);
    }
    if(sym.outline)
    {
        g.setColor(sym.outlineColor);
        g.drawRect(x, y, s, s);
    }
    break;

case 2: // '\002'
    int xPoints[] = {
        x, x0, x0 + s / 2
    };
    int yPoints[] = {
        y0 + s / 2, y, y0 + s / 2
    };
    if(sym.fill)
    {
        g.setColor(sym.fillColor);
        g.fillPolygon(xPoints, yPoints, 3);
    }
    if(sym.outline)
    {
        g.setColor(sym.outlineColor);
        g.drawPolygon(xPoints, yPoints, 3);
    }
    break;

case 3: // '\003'

```

```

        if(!sym.outline)
            return;
        g.setColor(sym.outlineColor);
        g.drawLine(x, y0, x0 + s / 2, y0);
        g.drawLine(x0, y, x0, y0 + s / 2);
        break;

    default:
        if(sym.fill)
        {
            g.setColor(sym.fillColor);
            g.fillOval(x, y, s, s);
        }
        if(sym.outline)
        {
            g.setColor(sym.outlineColor);
            g.drawOval(x, y, s, s);
        }
        break;
    }
}

```

```

public void paintShape(Graphics g, Projection prj, Symbol sym, int type,
Dimension d)
{
    try
    {
        int pointCount = xCoords.length;
        int xCoords2[] = new int[xCoords.length];
        int yCoords2[] = new int[yCoords.length];
        for(int i = 0; i < pointCount; i++)
        {
            xCoords2[i] = (int)((xCoords[i] + prj.shift.x) * prj.zoom);
            yCoords2[i] = (int)((-yCoords[i] + prj.shift.y) * prj.zoom);
        }

        switch(type)
        {

```

```

default:
    break;

case 1: // '\001'
    drawPoint(g, sym, xCoords2[0], yCoords2[0]);
    break;

case 3: // '\003'
    Utils2D.polygon(g, xCoords2, yCoords2, pointCount, sym);
    break;

case 2: // '\002'
    if(!sym.outline)
        return;
    if(Utils2D.have2d() && sym.g2dStroke != null)
    {
        Utils2D.line(g, xCoords2, yCoords2, pointCount, sym);
        break;
    }
    g.setColor(sym.outlineColor);
    if(sym.style == 0 || sym.style == 5)
        if(sym.size == 2)
            drawBoldPolyline(g, xCoords2, yCoords2, pointCount);
        else {
            if (emap.MapFactory.ZOOM_VALUE<1000) {

emap.MapFactory.drawStrokeLine(g,sym,6.0F,xCoords2,yCoords2,pointCount);
                break;
            }
            else if(emap.MapFactory.ZOOM_VALUE>25000 &&
(sym.value.toString().trim().equals("5") || sym.value.toString().trim().equals("6") ||
sym.value.toString().trim().equals("7")))
                break;
            else if(emap.MapFactory.ZOOM_VALUE>25000 &&
sym.value.toString().trim().equals("1")){

emap.MapFactory.drawStrokeLine(g,sym,2.0F,xCoords2,yCoords2,pointCount);
                break;
            }
        }
    }

```



```

        }
        else
            g.drawPolyline(xCoords2, yCoords2, pointCount);
    }
    if(sym.style > 0)
        drawCustomLine(g, xCoords2, yCoords2, sym);
    break;
}
}
catch(Exception exception) { }
}

```

```

public FloatRectangle calculateExtent()
{
    double maxX = xCoords[0];
    double maxY = yCoords[0];
    double minX = maxX;
    double minY = maxY;
    for(int i = 1; i < xCoords.length; i++)
    {
        if(maxX < xCoords[i])
            maxX = xCoords[i];
        else
            if(minX > xCoords[i])
                minX = xCoords[i];
        if(maxY < yCoords[i])
            maxY = yCoords[i];
        else
            if(minY > yCoords[i])
                minY = yCoords[i];
    }

    return new FloatRectangle(minX, minY, maxX, maxY);
}

```

```

public boolean containsPoint(FloatPoint point)
{
    return containsPoint(this, point.x, point.y);
}

```

```

    }

    static boolean containsPoint(Shape shp, double x, double y)
    {
        double xCoords[] = shp.xCoords;
        double yCoords[] = shp.yCoords;
        int sc = 0;
        int len = xCoords.length;
        double p1x = xCoords[len - 1];
        double p1y = yCoords[len - 1];
        for(int i = 0; i < len; i++)
        {
            double p2x = xCoords[i];
            double p2y = yCoords[i];
            if((x < p1x || x < p2x) && (y < p1y && y >= p2y || y < p2y && y >=
p1y))
                if(x < p1x && x < p2x)
                    sc++;
                else
                    if(((y - p1y) * (p2x - p1x)) / (p2y - p1y) + p1x > x)
                        sc++;
                p1x = p2x;
                p1y = p2y;
            }

        return (sc & 1) != 0;
    }

    boolean containsPoint(double x, double y)
    {
        return containsPoint(this, x, y);
    }

    public boolean isClockWise()
    {
        int npoints = xCoords.length - 1;
        double area = xCoords[npoints] * yCoords[0] - xCoords[0] *
yCoords[npoints];

```

```

        for(int i = 0; i < npoints; i++)
            area += xCoords[i] * yCoords[i + 1] - xCoords[i + 1] * yCoords[i];

        return area >= 0.0D;
    }

    public void setVertex(int index, double x, double y)
    {
        if(index >= 0 && index < xCoords.length)
        {
            xCoords[index] = x;
            yCoords[index] = y;
        }
    }

    public Vector getCoordinates()
    {
        int cnt = xCoords.length;
        Vector v = new Vector(cnt - 1);
        for(int i = 0; i < cnt; i++)
            v.addElement(new FloatPoint(xCoords[i], yCoords[i]));

        return v;
    }

    private static final int MARKER_CIRCLE = 0;
    private static final int MARKER_SQUARE = 1;
    private static final int MARKER_TRIANGLE = 2;
    private static final int MARKER_CROSS = 3;
    private static final int LINE_SOLID = 0;
    private static final int LINE_DASH = 1;
    private static final int LINE_DOT = 2;
    private static final int LINE_DASHDOT = 3;
    private static final int LINE_DASHDOTDOT = 4;
    private static final int LINE_STRIPE = 5;
    public double xCoords[];
    public double yCoords[];
}

```

## 2.4. 動態載入地圖

```
public static void loadCurrentArea() {
    for (int i = 1; i < 23; i++) {
        boolean isNeeded = false;
        //System.out.println(MapFactory.isVisibleVector);
        for (int j = 0; j < MapFactory.isVisibleVector.size(); j++) {
            int tArea = Integer.parseInt(((Record) (MapFactory.
                isVisibleVector.elementAt(j))).getId());
            if (tArea != i) {
                continue;
            }

            isNeeded = true;
        }
        if (!isNeeded &&
            map.getProject().getLayerByIndex(i * 2).isVisible()) {
            System.out.println("已經移除第 " + i + " 區塊");
            map.getProject().getLayerByIndex(i * 2).setVisible(false);
            map.getProject().getLayerByIndex(i * 2).clear(false);
            System.gc();
        }
    }
    System.gc();
    for (int i = 1; i < 23; i++) {
        boolean isNeeded = false;
        for (int j = 0; j < MapFactory.isVisibleVector.size(); j++) {
            int tArea = Integer.parseInt(((Record) (MapFactory.
                isVisibleVector.elementAt(j))).getId());
            if (tArea != i) {
                continue;
            }
            isNeeded = true;
        }
        if (isNeeded) {
            System.out.println("已經加入第 " + i + " 區塊");
            map.getProject().getLayerByIndex(i * 2).setVisible(true);
            try {
                while (map.sthRunning) {
```

```

        Thread.sleep(100);
    }
    } catch (Exception e) {

    }
}

System.out.print("目前內部有 :");
for (int i = 1; i < 23; i++) {
    if (map.getProject().getLayerByIndex(i * 2).isVisible()) {
        System.out.println("" + i + " ");
    }
}
System.out.println(" 區塊");
}

```

## 2.5. K-Means 演算法分群

```

public class KM
{

    public KM(int width, int height)
    {
        array_Data = new ArrayList();
        array_Cluster = new ArrayList();
        h = new History();
        steps = 0;
        this.width = width;
        this.height = height;
    }

    public void addR(int x, int y, String ID, String descr)
    {
        Data data = new Data(x, y);
        data.setID(ID);
        data.setDescr(descr);
        array_Data.add(data);
    }
}

```

```

    }

    public void initCluster(int width, int height)
    {
        array_Cluster.add(new Cluster((int)((double)width *
0.33000000000000002D), (int)((double)width * 0.25D)));
        array_Cluster.add(new Cluster((int)((double)width *
0.33000000000000002D), (int)((double)width * 0.5D)));
        array_Cluster.add(new Cluster((int)((double)width *
0.33000000000000002D), (int)((double)width * 0.75D)));
        array_Cluster.add(new Cluster((int)((double)width *
0.66000000000000003D), (int)((double)width * 0.25D)));
        array_Cluster.add(new Cluster((int)((double)width *
0.66000000000000003D), (int)((double)width * 0.5D)));
        array_Cluster.add(new Cluster((int)((double)width *
0.66000000000000003D), (int)((double)width * 0.75D)));
    }

    public Vector getMeans()
    {
        cluster_points = new Vector();
        joinToCluster();
        h.addArrayCluster(array_Cluster);
        for(; !checkStopCondition(); h.addArrayCluster(array_Cluster))
        {
            steps++;
            computeNewCentroids();
            joinToCluster();
        }

        Vector v_out = new Vector();
        Vector v_in = new Vector();
        for(int j = 0; j < array_Cluster.size(); j++)
        {
            for(int i = 0; i < array_Data.size(); i++)

if(((Data)(Data)array_Data.get(i)).getCluster().equals(array_Cluster.get(j)))
            {

```

```

        XRecord record = new XRecord();
        record.x = ((Data)(Data)array_Data.get(i)).x;
        record.y = ((Data)(Data)array_Data.get(i)).y;
        record.id = ((Data)(Data)array_Data.get(i)).getID();
        record.descr = ((Data)(Data)array_Data.get(i)).getDescr();
        record.width = width;
        record.height = height;
        v_in.addElement(record);
    }

    if(v_in.size() != 0)
    {
        v_out.addElement(v_in);
        cluster_points.addElement(new
Point(((Cluster)(Cluster)array_Cluster.get(j)).x,
((Cluster)(Cluster)array_Cluster.get(j)).y));
    }
    v_in = new Vector();
}

return v_out;
}

public Vector getClusterPoints()
{
    return cluster_points;
}

private void joinToCluster()
{
    for(int i = 0; i < array_Data.size(); i++)
    {
        int num_c = 0;
        double min_distance = (1.0D / 0.0D);
        Data d = (Data)array_Data.get(i);
        for(int j = 0; j < array_Cluster.size(); j++)
        {
            Cluster c = (Cluster)array_Cluster.get(j);

```

```

        int dx = Math.abs(d.x - c.x);
        int dy = Math.abs(d.y - c.y);
        double current_distance = (dx ^ 2) + (dy ^ 2);
        current_distance = Math.pow(current_distance, 0.5D);
        if(current_distance < min_distance)
        {
            min_distance = current_distance;
            num_c = j;
        }
    }

    Cluster c = (Cluster)array_Cluster.get(num_c);
    d.joinToCluster(c);
}

}

private void computeNewCentroids()
{
    for(int i = 0; i < array_Cluster.size(); i++)
    {
        Cluster newC = (Cluster)array_Cluster.get(i);
        Point baricentro = averageOf(newC);
        newC.setSpot((int)baricentro.getX(), (int)baricentro.getY());
    }
}

private Point averageOf(Cluster cluster)
{
    int counter = 0;
    int amountX = 0;
    int amountY = 0;
    for(int i = 0; i < array_Data.size(); i++)
    {
        Data d = (Data)array_Data.get(i);
        if(d.getCluster() == cluster)
        {

```



```

        counter++;
        amountX += d.x;
        amountY += d.y;
    }
}

Point punto;
if(counter == 0)
    punto = new Point(cluster.x, cluster.y);
else
    punto = new Point(Math.abs(amountX / counter), Math.abs(amountY /
counter));
return punto;
}

private boolean checkStopCondition()
{
    if(steps < 1)
        return false;
    if(steps > 200)
        return true;
    ArrayList array_p = h.getPrevious();
    double max = 0.0D;
    for(int j = 0; j < array_Cluster.size(); j++)
    {
        Cluster a = (Cluster)array_Cluster.get(j);
        Cluster b = (Cluster)array_p.get(j);
        int dx = a.x - b.x;
        int dy = a.y - b.y;
        double d = dx ^ 2 + dy ^ 2;
        max = Math.max(max, Math.sqrt(d));
    }

    return max < 1.0D;
}

ArrayList array_Data;
ArrayList array_Cluster;

```

```

public static Vector cluster_points = new Vector();
History h;
int steps;
int width;
int height;

}

```

## 2.6. 地圖放大縮小

```

public void zoomInMap() {
    FloatPoint center = map.screenToMap(new Point(map.getWidth() / 2,
        map.getHeight() / 2));
    center.x += 0.01;
    System.out.println( map.projection.zoom );
    map.setExtent(map.getExtent().scale(0.2D), null);
    System.out.println( map.projection.zoom );
    if (MapFactory.ZOOM_VALUE < 25000L) {
        MapFactory.setVisibleLayer(map.getExtent(),
            map.getLayer("Main_Contours"));
        loadCurrentArea();
    }
    map.updateMap();
}

public void zoomOutMap() {
    FloatPoint center = map.screenToMap(new Point(map.getWidth() / 2,
        map.getHeight() /
2));
    center.x += 0.01;

    map.setExtent(map.getExtent().scale(5D), null);
    System.out.println( map.getExtent().scale(0.2D).x );
    if (MapFactory.ZOOM_VALUE < 25000L) {
        MapFactory.setVisibleLayer(map.getExtent(),
            map.getLayer("Main_Contours"));
        loadCurrentArea();
    }
    if (MapFactory.ZOOM_VALUE > 25000L) {

```

```

    for (int i = 1; i < 23; i++) {
        boolean isNeeded = false;
        for (int j = 0; j < MapFactory.isVisibleVector.size(); j++) {
            int tArea = Integer.parseInt(((Record) (MapFactory.
                isVisibleVector.elementAt(j))).getId());
            if (tArea != i) {
                continue;
            }
            isNeeded = true;
        }
        if (isNeeded) {
            map.getProject().getLayerByIndex(i * 2).setVisible(false);
            map.getProject().getLayerByIndex(i * 2).clear(false);
        }
    }
    map.updateMap();
}
}
}

```

### 3. 水位偵測

```
byte[] signal = new byte[1];
```

```
Enumeration e = CommPortIdentifier.getPortIdentifiers();
```

```
try{
```

```
    CommPortIdentifier port =
```

```
CommPortIdentifier.getPortIdentifier("COM1");
```

```
    if( port.getPortType() == CommPortIdentifier.PORT_SERIAL )
```

```
        System.out.println( "it is a serial port\n" );
```

```
    else
```

```
        System.out.println( "it is not a serial port\n" );
```

```
CommPort commPort = port.open("COM1", 500);
```

```
InputStream in = commPort.getInputStream();
```

```

OutputStream out = commPort.getOutputStream();

while( true ){
    //signal[0] = (byte)(Math.random()*127+1);
    out.write("warning".getBytes());
    byte[] bbb = new byte["warning".getBytes().length];
    in.read(bbb);
    if( new String(bbb).equals("warning") )
        System.out.println( new String( bbb ) );
    else
        System.out.println( "safe" );
    Thread.sleep(1000);
}

/*while( true ){
    in.read(signal);
    System.out.println( signal[0] );
    Thread.sleep(100);
}*/

//commPort.close();
}
catch (Exception ex){
    ex.printStackTrace();
}

while( e.hasMoreElements() ){
    System.out.println( e.nextElement().toString() );
}

System.out.println(System.getProperty("java.library.path"));

```