

第十三屆旺宏科學獎

創意說明書

參賽編號：SA13-260

作品名稱：手機盪鞦韆真 HIGH!

姓名：蔡士弦

關鍵字：手機應用、擺盪高度、即時高度測量

目錄

壹、 研究動機-----	1
貳、 研究目的-----	1
參、 研究設備及器材-----	2
肆、 研究過程及方法-----	3
一、研究過程-----	3
二、理論基礎-----	3
(一)三角函數-----	3
(二)WIFI 系統-----	4
(三)Android 系統-----	5
(四)手機感測器-----	7
(五)Visual Basic 程式語言-----	7
三、研究方法-----	8
(一)系統架構-----	8
(二)自製實驗設備-----	8
伍、 研究結果-----	12
一、Server 端 (裁判端)-----	13
(一)圖形監控介面-----	13
(二) 程式流程圖-----	14
二、Client 端(放在鞦韆端)-----	14
(一)感測介面-----	14
(二)程式流程圖-----	15
三、模擬擺盪測試 -----	15
四、測試與調整-----	16
五、現場實驗測量運作-----	17
陸、 討論-----	19
柒、 結論-----	20
捌、 參考資料及程式碼-----	20

壹、研究動機

某日我發現一則消息，內容是邀請大家參觀十二公尺高的巨型盪鞦韆比賽，這項比賽是當地一項傳統的民俗活動，聽說早期是為了消災祈福與健身而辦理的，到後來變成一項競技活動，比賽看誰盪得最高，因為鞦韆高度有四層樓高，光要站在上面就要有相當的膽量與勇氣，而每屆的前幾名都競爭激烈，擺盪的高度幾乎都成水平，讓人驚呼連連。

不過到了那才發現評審們的評分方法是**靠肉眼評估高度**，而約略的排出名次。想想現在是 21 世紀，身邊幾乎所有的東西都是講究科學數據，那這種方法似乎就有些說不過去。於是基於這個理念，我要想出如何解決這個問題的方法。



盪鞦韆比賽看誰盪的高！



但裁判卻是透過一根管子目測高度

貳、研究目的

我曾經問過一間在賣各種設備的儀器行，問問看是否有辦法得到移動中物體的高度，聽說是有但需要架構一些聽也沒聽過設備，隨便架構起來少說也要幾十幾百萬，要支付那麼龐大的金額實在有些困難，而且又增加其複雜性。於是想了想使否有辦法應用**簡單的方法解決擺盪高度的問題**。

恰巧某一天我下載一個手機遊戲，此遊戲靠手機平衡來讓遊戲過關，這時我發現手機竟然可以偵測傾斜角度，於是經過思考，我們想出了可以運用學校教過的「三角函數」來解決。

概念其實非常簡單，只要把鞦韆的擺長當作斜邊後，運用鞦韆擺盪的角度代入 \cos 函數，再用鞦韆總高度去減差不多就出來了。

希望完成之後能作為主辦單位的測量工具，讓比賽更公平、更有趣。

參、研究設備及器材

軟硬體設備名稱	規格	功能
智慧型手機 Sony (XPERIA ray)、 TaiwanMobile(Amazing 1)……等		放置鞦韆上測量角度
無線基地台(AP) Dlink		傳送資料
筆電 Acer14吋		裁判端的監控畫面
活動式閱讀架		原設計是放置 iPad 的閱讀架，加工成實驗工具，用來測量手機傾斜角度
指北針		校正手機 X 軸
Visual Basic 2010 與 6.0		撰寫於 server 端的程式
Eclipse Android + JDK		Client(手機)端程式
鞦韆架模型 與 高度標示板		擺盪測試用

肆、研究過程及方法

一、研究過程

本研究過程如下：

- (一) 研究智慧手機內建各種感測器
- (二) 自行製作實驗器具，比較各種手機的感測晶片，並建立量測數據以便作為將來實作後的修正依據
- (三) 研究手機與電腦的無線傳輸方法
- (四) 撰寫程式語言
- (五) 實地測試

二、理論基礎

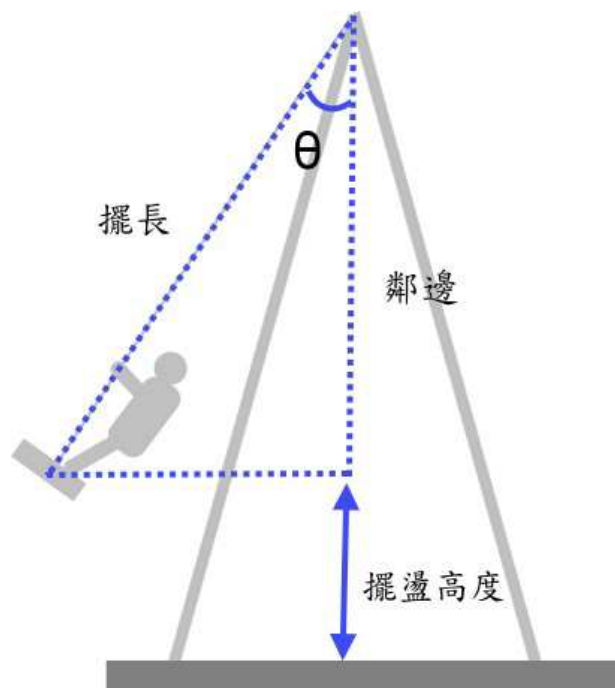
(一)三角函數

我們利用三角函數，將擺長設為斜邊，擺盪角度 θ 帶入 \cos 函數，求出鄰邊後，再以鞦韆的高度減去三角形的鄰邊，以得到擺盪高度。

公式： $\cos \theta = \text{鄰邊} / \text{斜邊}$

→ $\text{鄰邊} = \text{斜邊} * \cos \theta$

→ $\text{擺盪高度} = \text{鞦韆高度} - \text{三角形鄰邊}$



(二)WIFI 系統

網路(或是無線網路)，其協定大致分為 TCP/IP，而在 TCP/IP 協定組中，關於傳送層的協定就是 TCP 和 UDP 了。簡而言之，TCP 提供的是一個可靠的資料流傳送服務；相對而言，UDP 提供的是一個非可靠的非連線型資料流傳送服務。

1. TCP 協定(可靠性傳送服務的特性)

TCP 提供了一套協定，能夠將電腦之間使用的資料透過網路相互傳送，同時也提供一套機制來確保資料傳送的準確性和連續性。

Wifi 中的 TCP 協定，會將傳出去的值經過確認是否合理後再傳回來，雖然這樣出來的值會比較穩，但那會使每秒讀取的次數過低，以至於電腦上的擺盪情形會慢於實際情形。

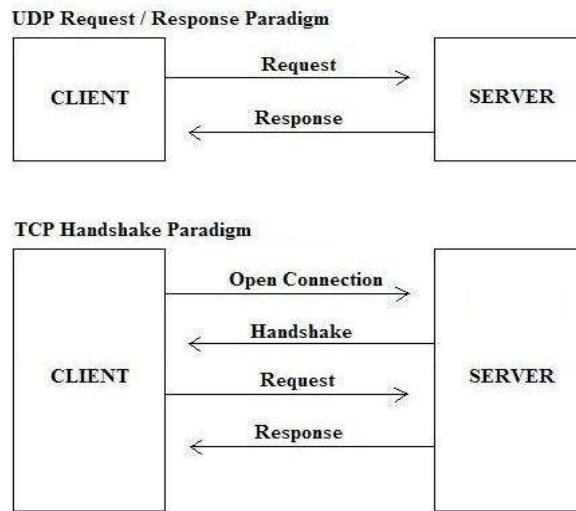
這種技術需要接收端以確認信息(Acknowledgement) 回應發送端，肯定資料無誤的到達，同時雙方保留傳送的封包記錄，以作下一筆資料的確認依據。此外，還利用定時器的機制，以在傳送逾時後重新發送封包，以確保資料的完整性。

2. UDP 協定

UDP 是一個非連線型(Connectionless)的非可靠傳輸協定，它並不會運用確認機制來保證資料是否正確的被接收、不需要重傳遺失的資料、資料的接收可不必按順序進行、也不提供回傳機制來控制資料流的速度。因此，UDP 信息可能會在網路傳送過程中丟失、重複、或不依順序，而且抵達速度也可能比接收端的處理速度還快。由於 UDP 不需要可靠傳輸，因此相較於 TCP 來說，一大堆必需佔據封包表頭的 over head 都可省略，從而換取更大的 payroll 空間。這樣的結果，將令到單一的 IP 封包在作 UDP 連線時所攜載的資料要比 TCP 連線多更多。這是靠犧牲可靠性而換取得來的，若連線需要在 UDP 上作可靠傳輸，那麼，其確認機制將從傳輸層退為應用層進行了、也就是程式本身要提供可靠傳輸機制。

wifi 中的另一個協定 UDP，直接將值不經確認的直接傳出去，這樣的值雖然會比較不穩，但速度卻比 TCP 協定快上好幾倍，而達到電腦上的鞦韆可以與實際的鞦韆同步，視覺上看起來也較為合理。

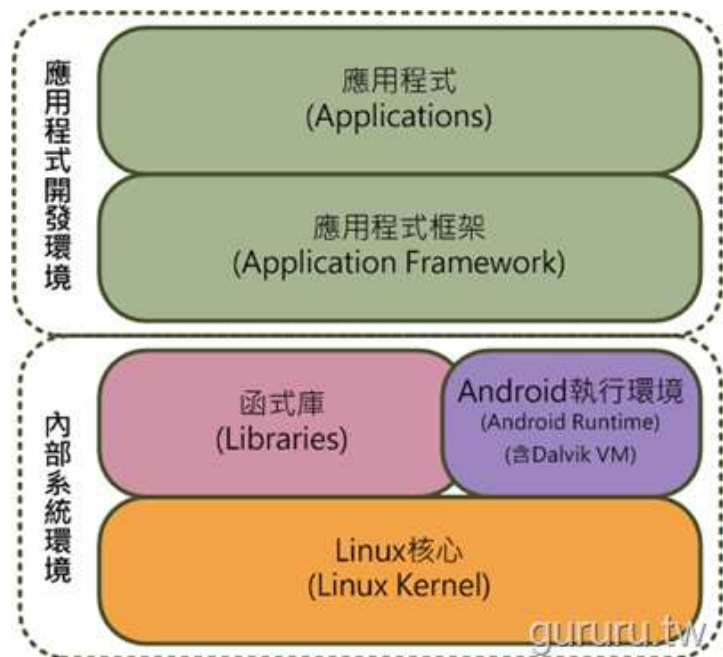
協定	優點	缺點
TCP	傳送可靠，程式可省略可靠機制。	速度比較慢。
UDP	傳輸量大，迅速。	不可靠，程式或需自行提供可靠機制。



(三)Android 系統

1. Google 於 2008 年正式公開的手機作業系統 Android，大幅度降低廠商發展智慧型手機系統的複雜度。Android 採用開放原始碼的技術與策略，讓全世界的開發人員都有機會深入了解並且改善其內部，使得 Android 成為熱門的研究平台。

基本上，由上而下，平台可以劃分為：作業系統、系統函式庫和應用程式等三個層級。最下層的作業系統源自於 Linux 核心，是個支援許多處理器與周邊裝置的開放原始碼作業系統。



Android 的系統架構圖

(圖表來源: 孫傳雄談科技(<http://blog.chinatimes.com/tomsun/archive/2010/08/31/533666.html>))

Google 在設計 Android 平台時採用了將系統拆解成對應特定領域模組的分層結構設計概念，其目的是讓不同領域的協力研發合作廠商，得以專心發展屬於自己特定領域的系統，以因應近來手機功能越益複雜化的趨勢。

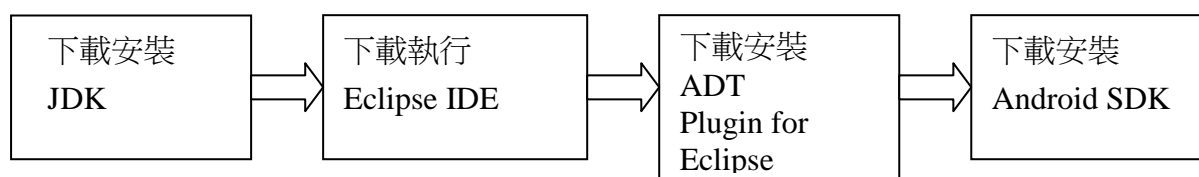
為了加速 Android 平台軟體的開發，Google 於 2008 年九月首先釋出第一版的 SDK 1.0 Release 1 供開發人員使用，之後，則規律的釋出新版本的 SDK。

Android 框架提供了一種開發模型，除了框架本身提供的 API (Application Programming Interface)，Google 針對開發週期需要的協助，發展一系列的工具有助開發人員，這組工具與框架統稱為 Android 軟體開發套件(Software Development Kit, SDK)。

2.Android 版本

版本名稱	代號	特點
測試版 2007/11/5 發布	-	-
Android 1.0 2008/9/23	鐵臂阿童木 Astro	<ul style="list-style-type: none"> ➤ 可以通過 Android Market 下載應用程式和獲得程式更新 ➤ 照相機支援
Android 2.0 2009/10/26	閃電泡芙 Éclair	<ul style="list-style-type: none"> ➤ 優化硬件速度 ➤ 新的瀏覽器的用戶介面和支持 HTML5
Android 3.0 2011/2/22	蜂巢 Honeycomb	<ul style="list-style-type: none"> ➤ 支援平板電腦大螢幕、高解析度 ➤ 加強多工處理的介面
Android 4.0 2011/10/19	雪糕三文治 Ice Cream Sandwich	<ul style="list-style-type: none"> ➤ 支持在系統中使用虛擬按鍵，該功能可以取代物理按鍵 ➤ 增加截圖功能（可以同時按住電源鍵和音量向下鍵進行截圖操作）

3.Android 開發步驟



(四)手機感測器

手機感測器大致有：

- **電子羅盤** (**本研究採用此感應器)
- **方向感應器** (**本研究採用此感應器)
- 重力感應器
- 磁場感應器
- 溫度感應器
- 光線感應器
- 壓力感應器
- 陀螺儀
- 距離感應器 等...。

雖然，Andorid 支援那麼多種感應器，但並非每台手機都具有這些設備。因此在開發應用程式時，仍應考量感應器的普及程度。就目前來說，市面上的 Andorid 手機具備重力感應器、電子羅盤(方向感應器)為大宗。

現階段在智慧型手機的機構中，大概是由 MEMS 或感測器元件所構築的 5 大區塊—動作感測、Cellular Radio Module、微投影(Picoprojector)、攝影模組(Camera Module)、前鏡頭攝影模組、微型麥克風(MEMS Microphone)。

從單一模組來看，動作感測元件包含常見的加速度計(Accelerometer)、陀螺儀(Gyroscope)、電子羅盤(Electronic Compass)、壓力計(Pressure Sensor)等。

Cellular Radio Module 則為前段射頻模組，主要包括：BAW 濾波器(Bulk Acoustic Wave Filter, BAW Filter)、BAW Duplexers、RF switch/Variable Capacitor、TCXO oscillators 等元件。

(五)Visual Basic 程式語言 (部份資料引用來源:維基百科)

Visual Basic 簡稱 VB，其中「Visual」是視覺的意思，代表一種「所見即所得」的設計方法。

VB 的設計理念就是要便於一般者使用，無論是新手或者專家。VB 使用了可以簡單建立應用程式的 [GUI](#) 系統，但是又可以開發相當複雜的程序。VB 的程序是一種基於視窗的可視化組件安排的聯合，並且增加代碼來指定組建的屬性和方法。因為預設的屬性和方法已經有一部分定義在了組件內，所以設計師不用寫多少代碼就可以完成一個簡單的程序。過去的版本裡面 VB 程序的性能問題一直被詬病，但是隨着[計算機](#)速度的飛速增加，關於性能的爭論已經越來越少。

由於它是圖形控制介面，只要有使用過微軟的作業系統很容易就可以上手，

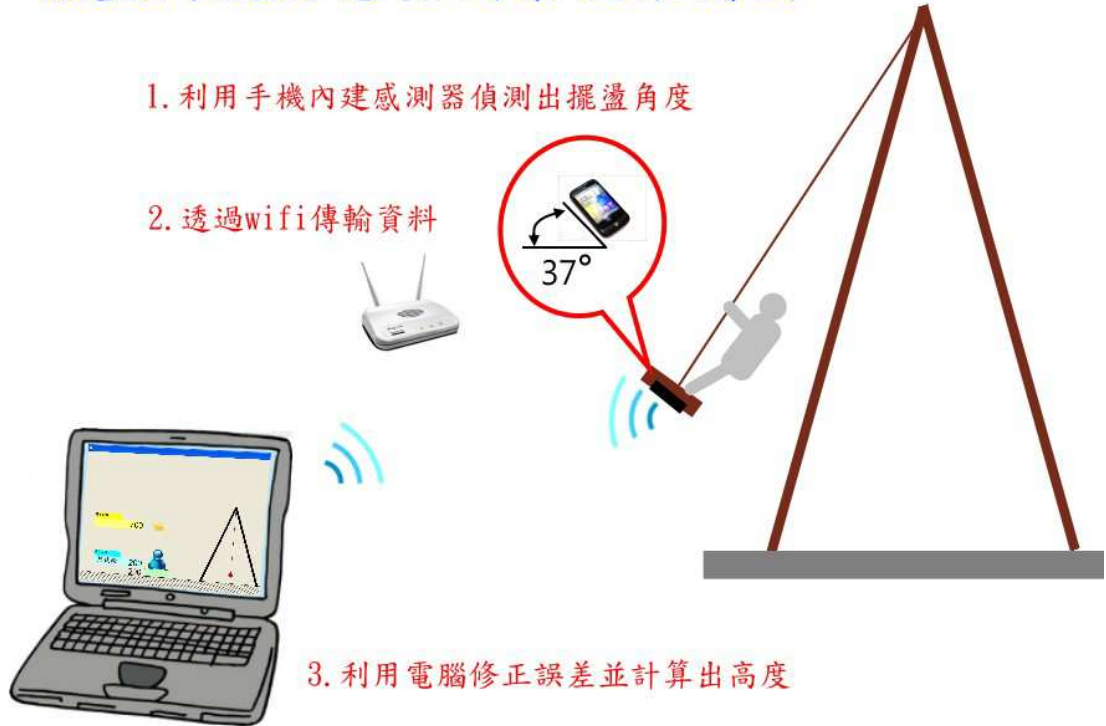
VB 除了自行設計程式外，也能引用外部別人設計好的控件，使得程式可以元件化。在我們的程式中，有關 wifi 傳輸我們就引用了微軟的 winsock.dll 元件，此元件可以大大減輕設計網路連線的細節問題，大大提升寫程式的效率。

三、研究方法

(一)系統架構

由手機偵測出角度訊號經過 wifi 基地台再送傳到電腦，再由電腦換算成高度後以圖形介面顯示出來。整體為一個區域型的封閉網路。

擺盪高度動態量測系統架構圖



(二)自製實驗設備

1.角度量測器

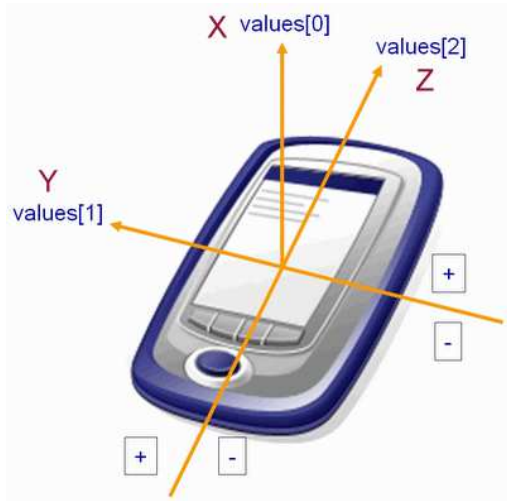
因為此量測系統是靠角度轉成高度，所以量測出來的角度是否準確，就會影響到高度的正確性，因此我們先自行製作了一個量測手機角度的設備搭配一個 APP 軟體(我的傳感器 MySensors)來比較各種手機所偵測到的數據是否準確，以作為將來系統之修正。



圖:MySensors APP 可以顯示手機內支援的感測器，並且提供晶片廠商與感測數據



圖:使用閱讀架改裝的角度測量工具，測驗各家手機的角度的準確性。



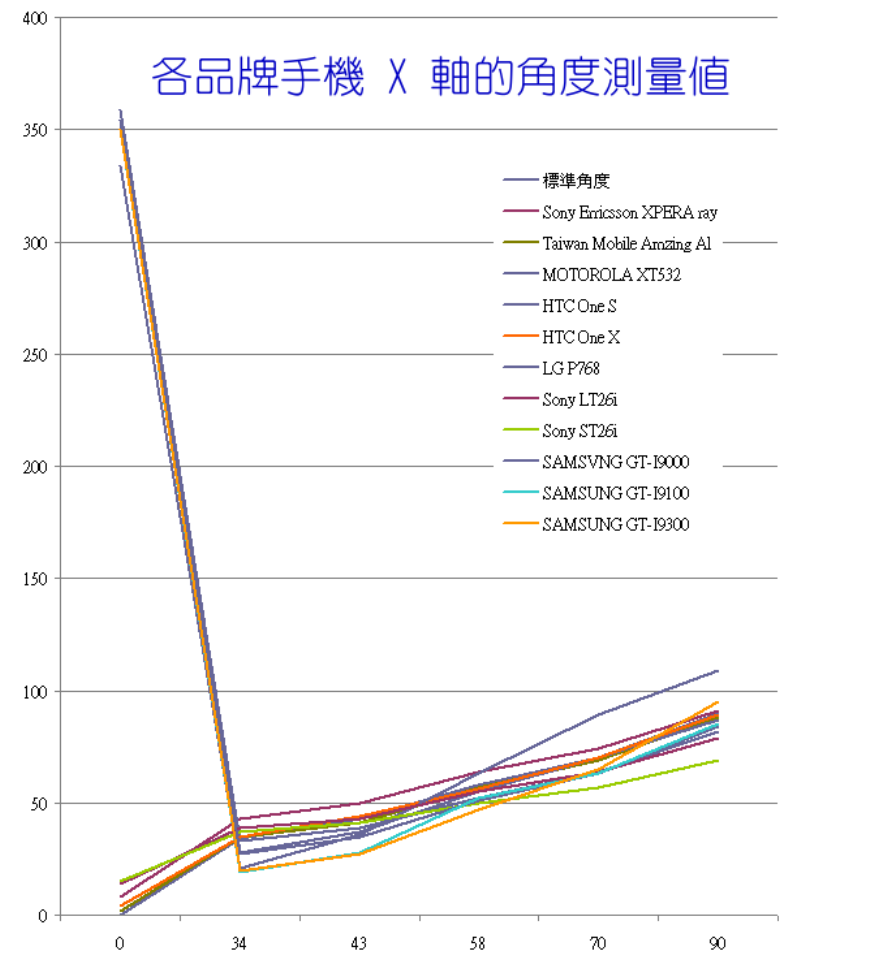
測試 11 種智慧型手機的 X、Y、Z 軸，各別取 0 度、34 度、43 度、58 度、70 度、90 度分析差異值，其中 X 軸很容易受到地磁與環境的影響，以 X 軸而言，HTC One X 的角度測量值比較符合標準角度

圖片來源: ola 的家

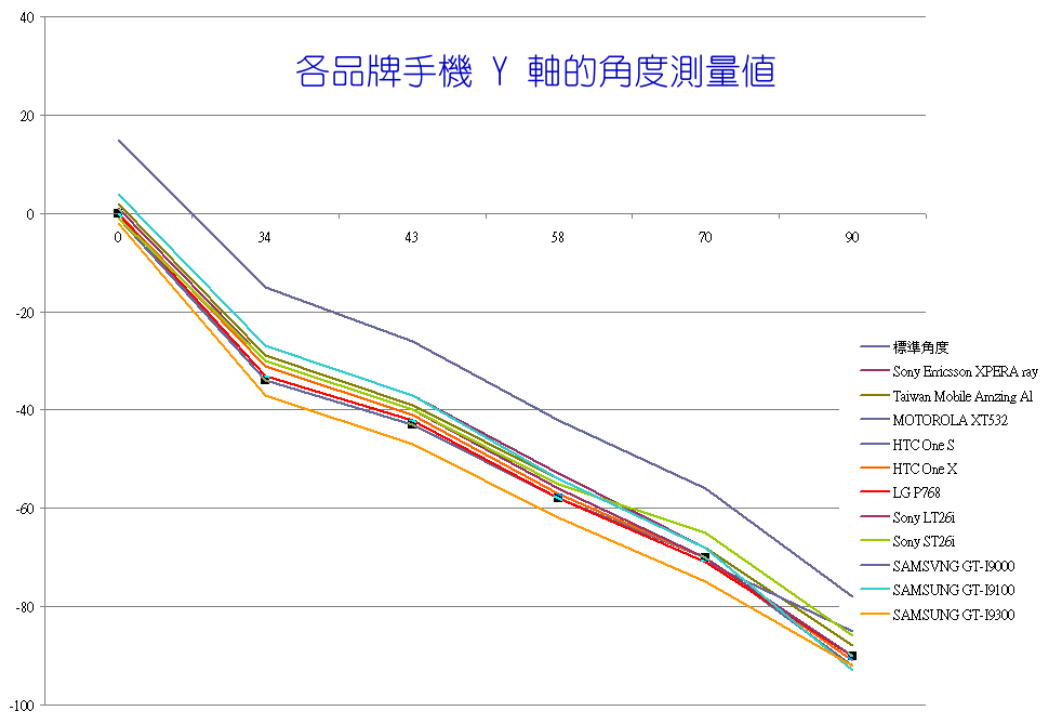
(<http://wangshifuola.blogspot.tw/2011/06/androidsensorsensortypeorientation.html>)

以下三張圖為各手機量測角度與實際角度的誤差情形，越是「接近標準角度」及「斜率越相近標準角度」的手機表示準確率越高。

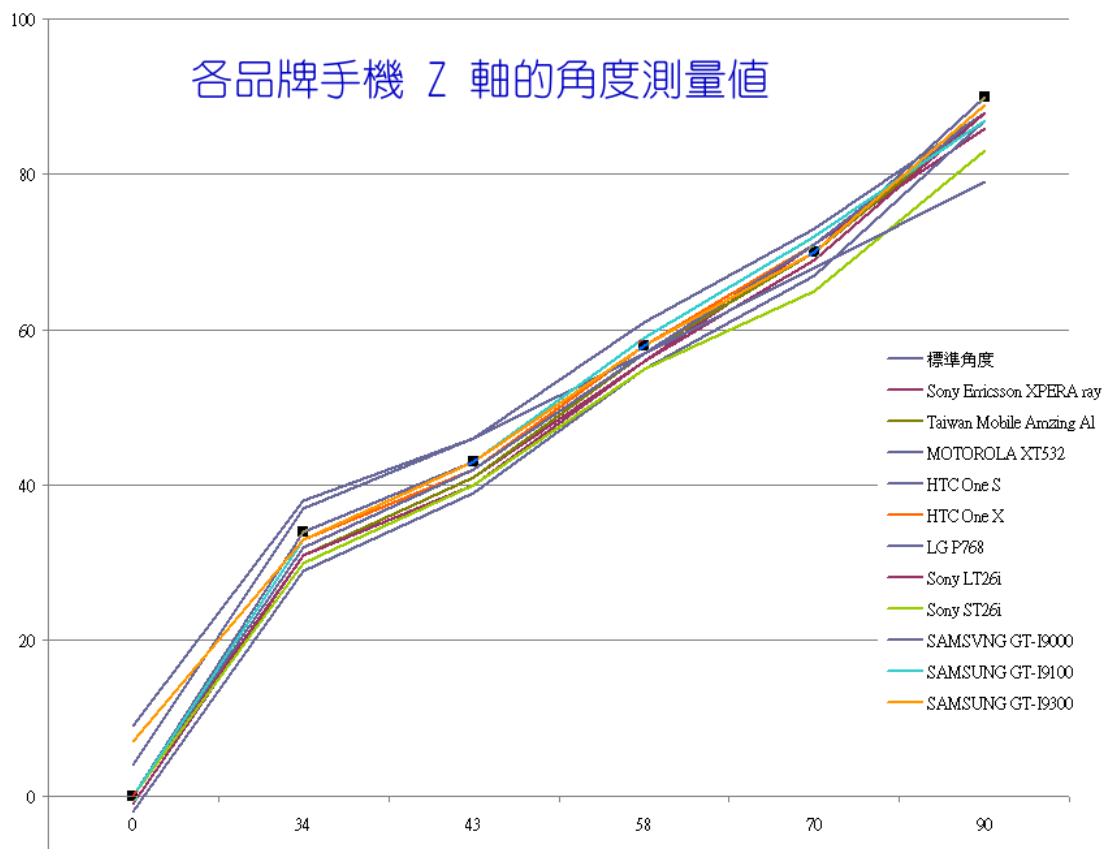
(橫軸：取樣的角度的，縱軸：所測出角度)



當標準角度為 0 時，有些手機的感測值為 359 度、354 度、350 度與 334 度，由於一周為 360 度，所以這些手機的誤差角度應為 1 度、6 度、10 度與 26 度。



以 Y 軸而言，HTC One S 的角度測量值 比較符合標準角度

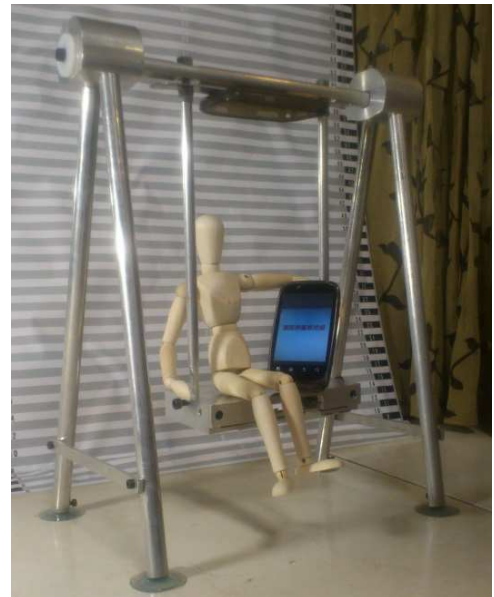


以 Z 軸而言，HTC One X 的角度測量值 比較符合標準角度

2.小型鞦韆架模型

我們請機械科同學幫忙製作一個小型的鞦韆架模型，做模擬測驗。

為了增加比賽氣氛的緊張與熱絡，我們將即時的擺盪高度用動態的圖形顯示出來，畫面可以顯示最高紀錄者的資料與目前參賽者的成績，將來可直接透過現場大螢幕播放給現場觀眾，增加現場歡樂與緊張的氣氛。



伍、研究結果

經過實驗與實地測量後，我們發現**實際的高度**(用雷射測距儀測量結果)與此**系統顯示的高度**會約有 1%的誤差，而且不太穩定，探究原因是因為手機的方向感測器本身就有約有 1%的誤差，而且容易受雜訊干擾，因此連帶造成後面的計算會不準，由於每個參賽者都是使用同一個手機偵測，所以 1%的誤差對每一位都會有同樣的誤差結果，所以這個誤差不會對比賽公平性造成影響；而雜訊干擾的部分，我們的解決方法是寫程式過濾判斷，將異常值捨去，結果是可以接受的。

為了增加比賽氣氛的緊張與熱絡，我們將即時的擺盪高度用動態的圖形顯示出來，畫面可以顯示最高紀錄者的資料與目前參賽者的成績，將來可直接透過現場大螢幕播放給現場觀眾，增加現場歡樂與緊張的氣氛。

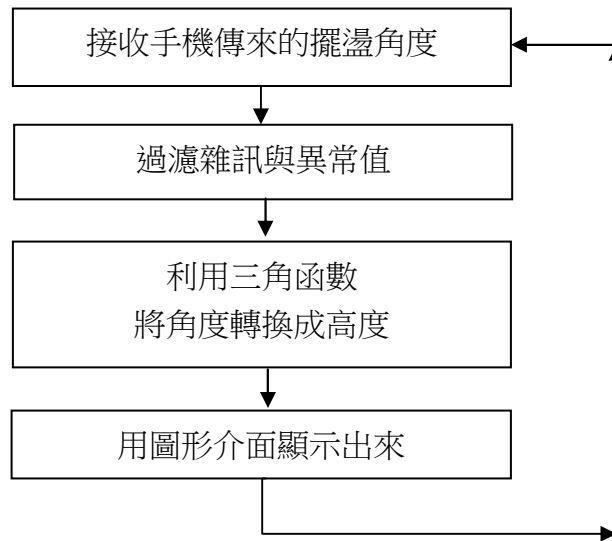
我們在 Server 端使用的電腦軟體為 VB2010 及 VB6.0，有兩者的原因是因為 VB6.0 較為簡單但卻和手機軟體的相容性比較差，所以先拿它當作範例；而 VB2010 對手機軟體的相容性則好許多，但跟 VB6.0 的架構已有些出入，因而較難完成。

手機開發我們使用了 Android 系統，配合 Eclipse 的開發環境使用 JAVA 撰寫的，因為這部分的較困難，我們請學校老師教我們撰寫主要程式碼，我們再以此程式碼修改成我們的需求。

一、Server 端 (裁判端) (一) 圖形監控介面



(二) 程式流程圖



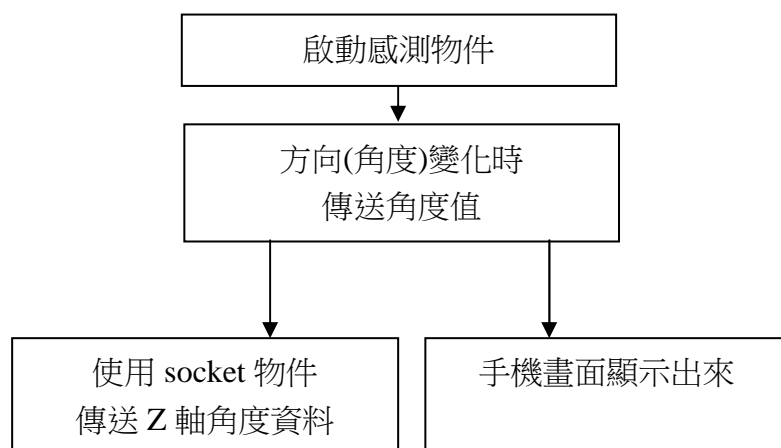
在 VB 部分則是要安裝網路元件，名稱叫做 MSWINSOCK，並且要註冊 winsock 元件。

二、Client 端(放在鞦韆端)

(一)感測介面



(二) 程式流程圖



三、模擬擺盪測試

手機擺盪時，裁判端的電腦會用圖形介面顯示實際高度，並紀錄比賽者的個人的最高紀錄



啟動手機內的角度偵測程式後，透過鞦韆底座裝設的一支紅色雷射筆，投射到背景的尺標，就可驗證高度是否正確。



四、測試與調整

我們使用了我們自製的角度比較器去量測各種機型、各種晶片後發現 X 軸會因地磁的關係產生數據嚴重偏離的情形，引此我們決定不採用。而在 Y 和 Z 軸方面，比對數據其實差不多，但因為我們的手機在 Z 軸方面比較準確，為了在測試方面得到較為精準的數據，我們採用 Z 軸。

最後我們遇到最困難的難題是手機每秒抓取的次數過低，導致無法使其看起來順暢並與電腦同步。我們上網找尋問題後發現其實手機每秒抓取的次數是夠的，所以應該是 TCP 協定需要雙向確定導致變慢，由此我們將原本的 TCP 協定改為 UDP 協定以求速度。

五、現場實驗測量運作

為了瞭解實際與模擬之差異，我們多次到現場實際測驗以求得真正數據。在光路里大鞦韆用雷射測距儀所實際的測量出它的坐檯離地面的距離為 2.24m，地面到軸位高度約 13.59m，整個鞦韆基座是由長寬約 7.58m 的正方型規格而設計的。

網路基地台部要鎖密碼，否則會發生干擾且網路傳輸速度下降的問題。



圖:到現場實際測量並將手機放在鞦韆上，測量擺動時的角度



圖:使用雷射測距儀，偵測高度及獲取實地數據

2013 年比賽的實際場景



此照片來源：全國社區報聯合新聞網 > 嘉義社區報 > 最新消息

<http://www.comnews.org.tw/?Page=BulletinDetail&Guid=c8bcfd7f-ab06-3747-bc23-8fb3f3eeb4a3>

陸、討論

一、問題 1：為什麼要以 wifi 通訊協定來傳輸訊息？

原因：目前我們最熟知的無線通訊協定就是藍芽和 wifi，原本我們是想用藍芽來進行傳輸，但考慮到距離過短，因此我們決定使用 wifi。

二、問題 2：如何解決感測信號會飄移或有雜訊的問題？

解決方法：我們寫了幾個排除不合理數值的程式，內容大概就是將超過平均一定百分比的數值刪除後再將其他數值平均。

三、問題 3：當場傳輸時會有其他手機干擾？

解決方法：我們會將 wifi 基地台設密碼不讓其他手機隨意進入而干擾到。

四、問題 4：手機裡的南北磁極會受到干擾嗎？

解決方法：我們在測量各種手機晶片時發現到縱使一根小金屬棒也會大大影響 X 座標的角度，或是當兩支手機互相靠近，內部的磁鐵也會對其中一支造成影響，還好鞦韆基座是木製材質不會影響。

五、問題 5：系統架設時應注意事項

解決方法：

1. 確認基地台的 SSID，筆電與手機都應該連到此基地台
2. 筆電與手機的 WIFI 都要記得打開
3. 筆電的 IP 應該固定，讓手機連到此 IP
4. 手機與筆電要帶備用電池，基地台要接不斷電系統

***六、問題 6：為什麼要改變手機放置的位子？

原因：我們所謂角度感測器的原理是由內部電容的重疊面積來推算角度的，如果我們將手機放置在座椅的地方，會發現盪出去的離心力會造成重疊面積額外位移，結果就會不準確，而軸心的地方就不會有離心力的問題。

本研究我們盡心將其做到最好，努力朝著「簡單的方法也可以改變世界」的方向繼續努力。

柒、結論

- 一、 手機裡感測角度的晶片不同會影響準確度，即使是同一型晶片，不同的手機品牌，也會有所誤差。
- 二、 手機作業系統雖然都是用 Android 測試，但版本不同會影響程式的撰寫，並不是後面的版本都可以相容前面的版本。
- 三、 手機量測 XYZ 軸的角度數據會因手機品牌、感測晶片、周圍是否有電磁干擾會有所差距。
- 四、 室內室外的雜訊會有很大的差距，要使用程式判斷過濾雜訊。
- 五、 SERVER 端(裁判端)和基地台使用有線傳輸會較無線穩定。
- 六、 這次測試的手機含括日系、韓系與本土品牌，在方向感測器方面國產的 HTC One 系列表現最好，大家要給國產的 HTC 多多支持。
- 七、 主辦單位給我們的信心和建議，讓我們進步，有自信。
- 八、 沒有實際測試永遠不會知道原來處處都會有問題等著你克服。
- 九、 很多學校學的理论如果實際運用在生活中，幫助別人解決問題，會覺得很快樂而且自己也受益良多。

由研究結果我們可以知道其實任何問題的解決方法不會只有一個，簡單也是可以完成。我們可以用大部分人可以懂得語言讓大家知道如何將鄉土文化融入科技，而又不失其原本的傳統。

本實驗作品經過實際運作與測量後確實可作為擺盪高度之測量，不過較為可惜的是各家手機採用的晶片與技術會造成量測的角度的誤差，但這經過我們接收端的電腦修正後誤差已經縮小到 1%，比傳統目測方式測量準確度已經大幅提升。我們期望今年採用此系統後可以讓比賽更公平更有趣。

捌、參考資料及程式碼

- 1.Netman 網中人(http://www.pcnet.idv.tw/pcnet/network/network_ip_intro.htm)
- 2.廖彧宏、洪士灝(http://www.csie.ntu.edu.tw/~hungsh/Android_NCP.pdf)
- 3.黃義立(http://mirokohyl.miroko.tw/bcc_pdf/vbs01.pdf)
- 4.王愉悅(<http://www.epochtimes.com/b5/13/2/17/n3802713.htm>)
- 5.孫傳雄談科技(<http://blog.chinatimes.com/tomsun/archive/2010/08/31/533666.html>)
- 6.洪綺君 (<http://www.digitimes.com.tw/tw/dt/n/shwnws.asp?CnIID=10&Cat=10&Cat1=&id=303127>)
- 7.DIGITIMES 中文網 原文網址:智慧型手機中感測元
<http://www.digitimes.com.tw/tw/dt/n/shwnws.asp?CnIID=10&Cat=10&Cat1=&id=303127#ixzz2OSbPxECs>)
- 8.Android 4.X 手機/平板電腦程式設計入門、應用到精通，碁峯出版，孫宏明
- 9.程式設計—使用 Visual Basic 2010，全華出版，黃建庭
- 10.維基百科
- 11.ola 的家 (<http://wangshifuola.blogspot.tw/2011/06/androidsensorsensortypeorientation.html>)

(一)手機端(Client)程式

```
package com.example.swingclient5;
import java.io.IOException;
import java.io.PrintWriter;
import java.net.DatagramPacket;
import java.net.DatagramSocket;
import java.net.InetAddress;
import java.net.Socket;
import java.net.SocketException;
import java.net.UnknownHostException;

import android.hardware.Sensor;
import android.hardware.SensorEvent;
import android.hardware.SensorEventListener;
import android.hardware.SensorManager;
import android.os.Bundle;
import android.os.Handler;
import android.os.Message;
import android.app.Activity;
import android.util.Log;
import android.view.View;
import android.view.View.OnClickListener;
import android.widget.Button;
import android.widget.EditText;

public class SwingClient5 extends Activity {

    private Button btnXXX,btnBYE;
    private EditText mEditText01;
    private Thread mThread = null;
    private DatagramSocket socket = null;
    private DatagramPacket packet = null;
    private Socket mSocket = null;

    private static String SERVERIP = "192.168.0.199";
    private PrintWriter mPrintWriter = null;
    private String mStrMSG = "";
    private static String TAG ="訊息:";
    public String str= "Hello, 6666";
    private byte[] data = str.getBytes();

    private SensorEventListener ori_listener = new SensorEventListener() {

        public void onSensorChanged(SensorEvent event) {
            // TODO Auto-generated method stub
            try
            {

                socket = new DatagramSocket();
                packet = new DatagramPacket(
                    data,
                    data.length,
                    InetAddress.getByAddress("192.168.0.199"),
                    9998);

                str= String.valueOf(event.values[1]) ;
                data = str.getBytes();

                socket.send(packet);
                socket.close();
            }
        }
    }
}
```

```

    } catch (Exception e)
    {
        // TODO: handle exception
        Log.e(TAG, e.toString());
    }
    //顯示 Z 軸角度
    mEditText01.setText("(UDP) Y: " + event.values[1]);
}

```

```

public void onAccuracyChanged(Sensor sensor, int accuracy) {
    // TODO Auto-generated method stub

}
};

```

@Override

```

public void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.activity_swing_client5);
    btnXXX = (Button) findViewById(R.id.button1);
    btnBYE=(Button) findViewById(R.id.button2);
    mEditText01 = (EditText) findViewById(R.id.editText1);

```

```

    SensorManager sm = (SensorManager) getSystemService(SENSOR_SERVICE);
    sm.registerListener(ori_listener, sm
        .getDefaultSensor(Sensor.TYPE_ORIENTATION),
        SensorManager.SENSOR_DELAY_GAME);

```

```

    btnXXX.setText(SERVERIP);
    mThread = new Thread(mRunnable);
    mThread.start();

```

//結束程式

```

    btnBYE.setOnClickListener(new OnClickListener()
    {
        public void onClick(View v)
        {
            socket.close();
            android.os.Process.killProcess(android.os.Process.myPid());
        }
    });
}

```

};

//////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////

Handler mHandler = new Handler()**//更新畫面（因為 Android 規定不能直接在線程中更新視圖）**

```

{
    public void handleMessage(Message msg)
    {
        super.handleMessage(msg);
        // 刷新
        try
        {
            mEditText01.append(mStrMSG);//顯示訊息
        } catch (Exception e)
        {
            Log.e("H 訊息", e.toString());
        }
    }
}

```

}

(二)電腦端 (Server)程式

```
Imports System.Drawing.Drawing2D
Public Class Form1
    Public Const HORIZON = 600
    Dim sb As New SolidBrush(Color.Red)
    Dim sb1 As New SolidBrush(Color.Blue)
    Dim angle, H, sL, B As Double
    Dim factor As Double
    Dim g1 As Graphics = Me.CreateGraphics
    Dim top, crown As Double
    Dim bb As Byte
    Dim data2 As String
    Dim sizex As Point
    Dim ScreenHeight, ScreenWidth As Integer
    Private Sub Form1_Paint(ByVal sender As Object, ByVal e As System.Windows.Forms.PaintEventArgs)
Handles Me.Paint
        Dim g As Graphics = e.Graphics
        Dim pen1 As New Pen(Color.SaddleBrown, 10)
        Dim pen2 As New Pen(Color.Black, 15)
        g.DrawLine(pen2, 0, HORIZON, ScreenWidth, HORIZON)
        'Dim linepenx As Brush = New(HatchStyle.Wave, Color.White, Color.Black)

        ' Dim linepen As Brush
        Dim rect1 As New Rectangle(0, 0, PictureBox1.Size.Width, PictureBox1.Size.Height - 50)
        Dim lgb As LinearGradientBrush = New LinearGradientBrush(rect1, Color.Gray, Color.White, 45)
        g.DrawLine(pen1, 680, HORIZON - 395, 520, HORIZON)
        g.DrawLine(pen1, 680, HORIZON - 395, 840, HORIZON)
        g.FillRectangle(lgb, 0, HORIZON, ScreenWidth, ScreenHeight - HORIZON)
    End Sub

    Private Sub Button1_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles
Button1.Click
        sL = Val(TextBox1.Text)
        B = Val(TextBox2.Text)
        TextBox6.Text = sL
        TextBox7.Text = B
        factor = 395 / (sL + B)
        ' g1.FillEllipse(sb, 770, 510, 20, 20)
        TextBox1.Visible = False
        TextBox2.Visible = False
        Label1.Visible = False
        Label2.Visible = False
    End Sub

    Private Sub Form1_Load(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles
MyBase.Load
        AxWinsock1.Protocol = MSWinsockLib.ProtocolConstants.sckUDPProtocol
        Me.WindowState = FormWindowState.Maximized
        ScreenHeight = Screen.PrimaryScreen.WorkingArea.Height
        ScreenWidth = Screen.PrimaryScreen.WorkingArea.Width
        AxWinsock1.LocalPort = 9998
        AxWinsock1.Bind(9998, "192.168.0.181")
    End Sub

    Private Sub AxWinsock1_DataArrival(ByVal sender As Object, ByVal e As
AxMSWinsockLib.DMSWinsockControlEvents_DataArrivalEvent)
        AxWinsock1.GetData(data2, vbString, 255)
        TextBox3.Text = data2
    End Sub

    Private Sub TextBox10_TextChanged(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles
TextBox10.TextChanged
```

```

top = 0
TextBox10.Location = New Point(90, HORIZON - PictureBox2.Height - top * factor)
PictureBox2.Location = New Point(190, HORIZON - PictureBox1.Height - top * factor)
TrackBar1.Value = 0

Refresh()
End Sub

Private Sub Timer1_Tick(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles Timer1.Tick

    sizex = New Point(0, 0)
    Dim g As Graphics = Me.CreateGraphics
    Dim ellipseX, ellipseY, p As Integer
    TextBox4.Text = data2
    angle = data2
    H = (sL + B) - sL * Math.Cos(angle * Math.PI / 180)
    TextBox5.Text = H

    ' g.IsVisible(sizex)
    ' Me.Refresh() '重繪物件為原乾淨畫布
    ' sizex = New Point(900, 800)
    ' g.IsVisible(sizex)

    ellipseY = Int(HORIZON - H * factor)
    ellipseX = 680 - sL * Math.Sin(angle * Math.PI / 180) * factor
    g.FillEllipse(sb, ellipseX - 10, ellipseY - 10, 20, 20)
    Dim pen2 As New Pen(Color.Black, 2)
    g.DrawLine(pen2, 680, HORIZON - 395, ellipseX, ellipseY)
    If H > top Then
        top = H
        TextBox8.Text = top
    End If
    TextBox10.Location = New Point(90, HORIZON - PictureBox2.Height - top * factor)
    PictureBox2.Location = New Point(184, HORIZON - PictureBox2.Height - top * factor)

    If top > crown Then
        crown = top
        PictureBox1.Location = New Point(190, HORIZON - PictureBox1.Height - PictureBox2.Height - top
* factor)
        TextBox9.Text = crown
    End If
    p = HORIZON - ellipseY '長方形長邊長度
    g.FillRectangle(sb1, 200, ellipseY, 70, p)
End Sub
End Class

```