國立屏東大學

資訊工程學系

實務專題期中報告書

盲人計算機

A calculator for blind people

專題學生: 黃羿豪 Student:Huang YI HAO

吳尉誠 WU WEI CHENG

李昀翰 LI YUN HAN

劉原晉 LIU YUAN JIN

指導教授: 李明錡 Advisor: LI MING QI

Project Proposal

Department of Computer Science and Information Engineering, National Pingtung University

Pingtung, Taiwan

JUNE, 2016中華民國 年 月

摘要

本專題利用「自動機原理技術」設計一個在智慧型手機盲人計算機APP。此計算機app的操作界面非常簡單，它只需三個按鈕就可讓盲人輸入所有的數及四則運算(包括開根號)。此三個按鈕分是「輸入」、「小數點」、「取消」三個狀態按鈕。實驗結果，盲人平均只需60秒就可完成一個四則運算。本研究利用MIT App inventor 設計一個可適用於所有廠牌智慧型手機的APP。

目錄

* 研究背景與動機
* 研究方法與工具
* 實例說明
* 演算法及流程圖
* 方塊程式圖解說明
* 討論
* 結論
* 參考文獻

研究背景與動機

對眼睛視力正常的人而言，使用觸控螢幕是多麼的自然而理所當然，但是對盲人而言，竟是困難障礙重重。這是因為目前的智慧型手機的電話簿是透過觸控螢幕來操控，但是觸控螢幕的隱藏式鍵盤或按鈕在顯示在玻璃上，玻璃材料的物理特性是光滑沒有刻度，這對盲人而言是無法辨識，反而不若傳統機械式的點字鍵盤系統可透過手指皮膚的觸感來辨識。所以智慧型手機如果不提供語音提示幫助的話，那麼觸控螢幕對盲人在使用上反而比不上傳統機械式的點字模式來的方便。另一方面雖然蘋果ios的手機有提供語音siri計算功能，但是畢竟不是專門為盲人設計的軟體，使用上口音辨識率的問題並無法百分百的辨識，因為世界上各民族有數百種語言，同一種語言又可能有多種方言口音，以中文而言就有數百種南腔北調方言及口音，所以很難做到對每種語言皆百分百辨識，而且蘋果手機售價很貴，對大部份經濟弱勢的盲胞朋友而言實在是買不起。根據世界衛生組識統計資料顯示，目前世界上大約有3千9百萬盲人，另外有2千8百萬人屬於視力障礙人(visually impaired)[[3](#_ENREF_3)]，總數將近7000萬盲人，而這其中有相當多的人是盲啞人士也就是患有多重障礙又盲又啞，他們眼不能看嘴不能說，因為無法開口講話所以無法透過語音功能來辨識口音。不論是盲人或是盲啞人都需要四則運算，這個盲人手機市場是很大的。其實大部份的盲胞朋友要求不多，簡單易用最重要。對盲人而言如果可以不用眼睛，只要按一按螢幕就可打四則運算，不需要花時間東摸摸西摸摸或是滑動觸控螢幕，那麼對這將近7000萬盲人將是生活上的一大便利。要達成這個設計理念，我們計畫利用「自動機理論技術」設計一個在智慧型手機盲人計算機APP。此計算機app的操作界面非常簡單，它只需三個按鈕就可讓盲人輸入所有的數及四則運算(包括開根號)。此三個按鈕分是「輸入」、「小數點」、「取消」三個狀態按鈕。實驗結果，盲人平均只需60秒就可完成一個四則運算。本研究利用MIT App inventor 設計一個可適用於所有廠牌智慧型手機的APP。

研究方法與工具

一自動機 理論

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| 自動機(automata)簡介  自動機理論(automata theory)探討運算的數學模型，包括相關的定義與特性，這些模型在很多電腦科學的應用領域中都扮演相當重要的角色。其中有限自動機(finite automation)應用於文字處理(text processing)、編譯器(compiler)與硬體設計，與上下文無關的文法(context-free grammar)則運用於程式語言與人工智慧(artificial intelligence)。自動機理論很適合當做運算理論的入門主題，因為運算性的理論與複雜度理論都需要電腦的精確定義，這在自動機理論中經常可以看到實際的例子。 | | | | | | | |
| 認識有限自動機(finite automata)  有限自動機(finite automata)可以看成是一種模型，用來描述擁有極為有限的記憶體(memory)的電腦。在這麼少的記憶體的情況下，這種電腦能有什麼作用呢 ? 其實日常生活中有很多設備都具有類似的功能，例如圖1顯示的自動門，假如有人站在前感應區，則自動門應該開啟，即使行人通過前感應區，來到後感應區，自動門還是要開啟一下，讓行人有足夠的時間離開。  http://prompt.nou.edu.tw/web/sno204/content/w01/01-02-01txt.files/image002.jpg  圖1 用狀態圖來描述自動門的作用  自動門的作用可以用狀態圖(state diagram)來表示，基本上自動門的狀態有兩種 : 開啟(open)或是關閉(closed)。可能發生的情況則有4種，就是前感應區有人(front)、後感應區有人(rear)、都有人(both)或是都沒人(neither)，圖1的狀態圖清楚地顯示出當什麼情況發生時，自動門的狀態會有什麼變化。控制自動門的設備就像是一種有限自動機，只有單一位元(bit)的記憶體。上面介紹的只是有限自動機的實例，我們需要從數學的角度來探討有限自動機，了解有限自動機的功能與限制，這樣才能更清楚地區分有限自動機能完成與無法完成的工作。 | | | | | | | |
| 有限自動機的正式定義  從狀態變化圖可以清楚地看到有限自動機的作用，不過還是需要一個正式的定義來描述有限自動機，我們先來看圖2的有限自動機，像這樣的表示圖也稱為狀態圖(state diagram) ，有3個狀態，q1、q2與q3，q1是開始的狀態(start state) ，q2是所謂的接受狀態(accept state) ，箭頭線段代表狀態的變化(state transition)。  http://prompt.nou.edu.tw/web/sno204/content/w01/01-02-01txt.files/image004.jpg  圖2 有限自動機M1  當M1接受輸入字串1101時，會經過處理然後產生接受(accept)或是拒絕(reject)的輸出，字串1101從左到右依序進入M1，從q1開始， 由輸入的符號來判斷下一個狀態，最後輸入的符號將有限自動機引導到最後的狀態，若這個狀態是接受狀態，則輸出accept，否則就輸出reject。我們可以把處理的過程列出來 :  1.  從q1開始，接受輸入1，到q2。  2.  接著q2接受輸入1，到q2。  3.  接著q2接受輸入0，到q3。  4.  接著q3接受輸入1，到q2。  5.  輸出accept。  狀態圖的作用蠻有趣的，我們可以嘗試幾種不同的輸入字串，看看會得到什麼樣的輸出，最後應該會發現M1接受以1結尾的字串，拒絕其他的字串，假如M1所接受的字串代表一種語言(language) ，如何描述這種語言比較恰當呢 ? 這需要正式的定義，讓有限自動機的作用很明確地表達出來。 | | | | | | | |
| |  | | --- | | <定義> 一個有限自動機(finite automaton)是含有5個元素的5-tuple (Q，http://prompt.nou.edu.tw/web/sno204/content/w01/01-02-01txt.files/image006.gif，http://prompt.nou.edu.tw/web/sno204/content/w01/01-02-01txt.files/image008.gif，q0，F) ，Q是一個包含狀態(states)的有限集合，http://prompt.nou.edu.tw/web/sno204/content/w01/01-02-01txt.files/image006.gif是一個代表字母(alphabet)的有限集合，http://prompt.nou.edu.tw/web/sno204/content/w01/01-02-01txt.files/image008.gif:http://prompt.nou.edu.tw/web/sno204/content/w01/01-02-01txt.files/image010.gif，代表轉換的函數(transition function)，q0http://prompt.nou.edu.tw/web/sno204/content/w01/01-02-01txt.files/image012.gifQ，代表start state，Fhttp://prompt.nou.edu.tw/web/sno204/content/w01/01-02-01txt.files/image014.gifQ，代表可以接受的狀態的集合。 |   注意從正式的定義中可以發現，F可以是空集合，表示沒有可以接受的狀態，對於任意的狀態與輸入符號來說，http://prompt.nou.edu.tw/web/sno204/content/w01/01-02-01txt.files/image008.gif指定一個next state，這是從定義裡察覺出來的重要觀念。既然有了正式的定義，下面可以正式地描述有限自動機M1，M1=(Q，http://prompt.nou.edu.tw/web/sno204/content/w01/01-02-01txt.files/image006.gif，http://prompt.nou.edu.tw/web/sno204/content/w01/01-02-01txt.files/image008.gif，q1，F) :  1.          Q = {q1，q2，q3}  2.          http://prompt.nou.edu.tw/web/sno204/content/w01/01-02-01txt.files/image006.gif={0，1}  3.          http://prompt.nou.edu.tw/web/sno204/content/w01/01-02-01txt.files/image008.gif可以用下面的表格來描述 :   |  | | --- | |  | |  | http://prompt.nou.edu.tw/web/sno204/content/w01/01-02-01txt.files/image015.gif |     4.          q1是開始狀態  5.          F={q2}  假設A是所有M能接受的字串的集合，則A是machine M的語言(language)，M能辨識(recognize)A，或者說M接受(accept)A。通常有限自動機可能接受好多個字串，但是辨識的只有一種語言，不接受任何字串的有限自動機還是能辨識一種語言，即empty language。我們可以用下面的表示法來記載M1的特徵 :  A = {w | w包含至少1個1，最後的1後面有偶數個0}，L(M1)=A | | | | | | | |
| 圖3顯示有限自動機M2的狀態變化圖，以正式的定義來說，M2=({q1，q2}，{0，1}，http://prompt.nou.edu.tw/web/sno204/content/w01/01-02-01txt.files/image008.gif，q1，{ q2})，基本上，M2的狀態變化圖與正式的定義所含有的資訊是一樣的。  http://prompt.nou.edu.tw/web/sno204/content/w01/01-02-01txt.files/image018.jpg  圖3 有限自動機M2的狀態變化圖  試一試M2的作用，可以發現M2接受所有以1結束的字串，所以L(M2)={w|w最後一個字元為1}。圖4顯示有限自動機M3的狀態變化圖，M3跟M2很像，只是accept state的位置不同，由於start state也是accept state，所以M3可以接受空字串(empty string) http://prompt.nou.edu.tw/web/sno204/content/w01/01-02-01txt.files/image020.gif，M3接受所有以0結束的字串，所以L(M3)={w|w為空字串http://prompt.nou.edu.tw/web/sno204/content/w01/01-02-01txt.files/image020.gif或w最後一個字元為0}。  http://prompt.nou.edu.tw/web/sno204/content/w01/01-02-01txt.files/image023.jpg  圖4 有限自動機M3的狀態變化圖 | | | | | | | |
| 圖5顯示有限自動機M4的狀態變化圖，M4有兩個accept states，字母集合為{a，b}，M4接受像a、b、aa、bb與bab的字串，但是不接受ab、ba與bbba等字串，仔細觀察，可以發現M4接受的字串開始跟結束的符號一樣。  http://prompt.nou.edu.tw/web/sno204/content/w01/01-02-01txt.files/image025.jpg  圖5 有限自動機M4的狀態變化圖  圖6顯示有限自動機M5的狀態變化圖，M5的輸入字母有4個，即{<RESET>，0，1，2}，M5記錄輸入符號數字和mod 3的結果，假如是0的話，就會進入accept state。有時候可能無法以狀態圖來描述一個有限自動機，原因是狀態圖太複雜了，或是在描述上跟一些未設定的參數有關。  http://prompt.nou.edu.tw/web/sno204/content/w01/01-02-01txt.files/image027.jpg  圖6 有限自動機M5的狀態變化圖 | | | | | | | |
|  |  |  |  |  |  |  |  |

使用的工具是APP inventor2



MIT App Inventor 2 簡介

Google 公司在西元 2010 年，為了讓更多人在沒有程式基礎的狀況下，能

夠開發 Android 裝置應用程式，推出 Google App Inventor for Android 工具軟體，讓使用者可以運用拼圖模式來組合程式語法，進而發展 Android 裝置的應用程式。後來，Google 公司在西元 2012 年將此軟體，移轉給美國麻省理工學院（MIT）的行動學習中心（MIT Center for Mobile Learning）來維護，行動學習中心的網址為：「http://mitmobilelearning.org/」，其網站如圖所示。點選其中的「MIT's App Inventor site」超連結，會連結到 MIT App Inventor 2 網站。

麻省理工學院是在西元 2013 年 12 月 3 日推出 MIT App Inventor 2 網站，提供免費的雲端服務，使用者可以透過瀏覽器來開發 Android 裝置應用程式，該網站的網址為：「http://appinventor.mit.edu/」，如圖所示。

網站上提供了許多詳細的教學資源，來引導使用者進入 App 應用程式開發

的世界，使用者開發完成的程式，可以下載到任何使用 Android 作業系統的裝置，包括：智慧型手機或平板電腦。

在 App 程式開發過程中，我們可以使用模擬器或 Android 裝置，來測試應

用程式執行的狀況，待完成相關開發作業後，更可以將應用程式，上傳到 Google

Play 商店提供大眾下載安裝。

演算法及流程圖

1. 演算法

自動狀態機

Output

0：incint 6：subtraction

1：incint2 7：multiply

2：incdec 8：division

3：incdec2 9：Square root

4：resetADT a：resetA

5：addition out：result

Input

0：time < 3時click按鈕1

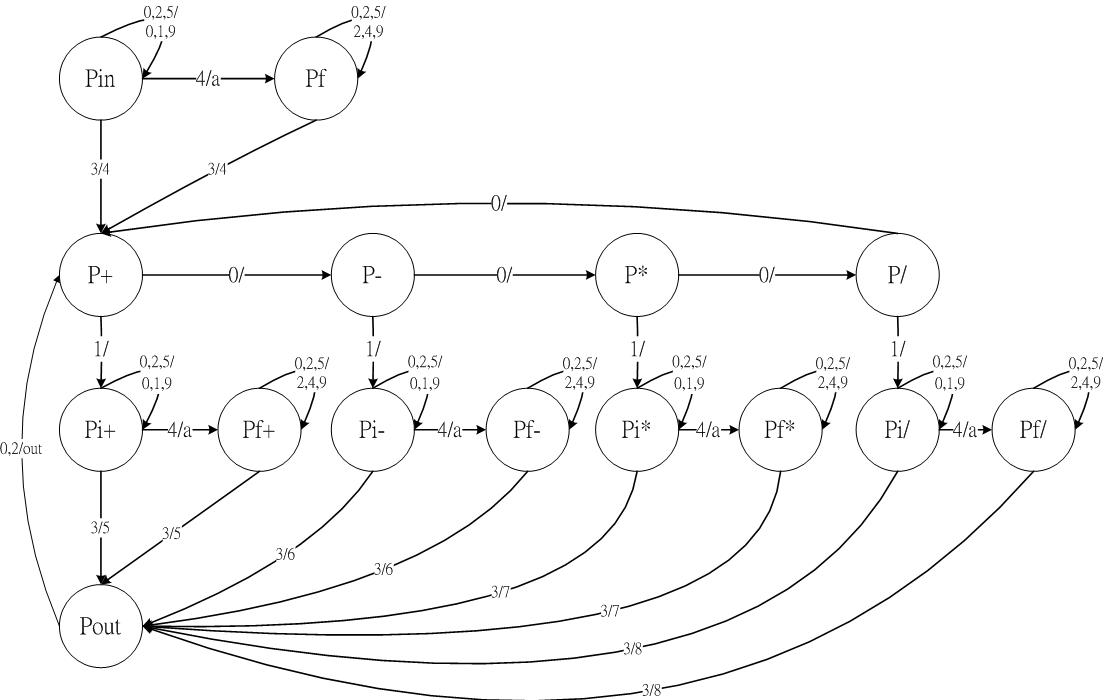
1：time = 3時

2：6 > time > 3時click按鈕1

3：time = 6時

4：time < 6時click按鈕3

5：time = 0時長按按鈕1



|  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | Input | | | | | | Output | | | | |
| 0 | 1 | 2 | 3 | 4 | 5 | 0 | 2 | 3 | 4 | 5 |
| Pin | Pin | Pin | Pin | P+ | Pf | Pin | incint | incint2 | resetADT | resetA | Square root |
| Pf | Pf | Pf | Pf | P+ | Pf | Pf | incdec | Incdec2 | resetADT | x | x |
| P+ | P- | Pi+ | x | x | x | x | x | x | x | x | x |
| P- | P\* | Pi- | x | x | x | x | x | x | x | x | x |
| P\* | P/ | Pi\* | x | x | x | x | x | x | x | x | x |
| P/ | P+ | Pi/ | x | x | x | x | x | x | x | x | x |
| Pi+ | Pi+ | Pi+ | Pi+ | Pout | Pf+ | Pi+ | incint | incint2 | addtion | resetA | Square root |
| Pf+ | Pf+ | Pf+ | Pf+ | Pout | Pf+ | Pf+ | incdec | Incdec2 | addtion | x | Square root |
| Pi- | Pi- | Pi- | Pi- | Pout | Pf- | Pi- | incint | incint2 | subtraction | resetA | Square root |
| Pf- | Pf- | Pf- | Pf- | Pout | Pf- | Pf- | incdec | Incdec2 | subtraction | x | Square root |
| Pi\* | Pi\* | Pi\* | Pi\* | Pout | Pf\* | Pi\* | incint | incint2 | multiply | resetA | x |
| Pf\* | Pf\* | Pf\* | Pf\* | Pout | Pf\* | Pf\* | incdec | Incdec2 | multiply | x | x |
| Pi/ | Pi/ | Pi/ | Pi/ | Pout | Pf/ | Pi/ | incint | incint2 | division | resetA | x |
| Pf/ | Pf/ | Pf/ | Pf/ | Pout | Pf/ | Pf/ | incdec | Incdec2 | division | x | x |
| Pout | P+ | Pout | P+ | Pout | Pout | Pout | out | out | x | x | x |

Incdec：

唸出t + ( (a++ % 10) / d)之值

Incint：

唸出t + (a++ % 10)之值

d初始值為10

t､t1､a､s､d皆為全域變數

t､t1､a､s初始值為0

division：

if(s == 1){

t = sqr(t)

s = 0

}

t1 = t1 / t;

a = 0

d = 10

t= 0

multiply：

if(s == 1){

t = sqr(t)

s = 0

}

t1 = t1 \* t;

a = 0

d = 10

t= 0

Incdec2：

t = t + ( (a++ % 10) / d)

d = d \* 10

a = 0

Incint2：

t = t + (a++ % 10)

t = t \* 10

a = 0

resetA：

a = 0

Square root：

s = 1

addtion：

if(s == 1){

t = sqr(t)

s = 0

}

t1 = t1 + t;

a = 0

d = 10

t= 0

resetADT：

if(s == 1){

t = sqr(t)

s = 0

}

t1 = t;

a = 0

d = 10

t= 0

subtraction：

if(s == 1){

t = sqr(t)

s = 0

}

t1 = t1 - t;

a = 01

d = 10

t= 0

(短按)按鈕1：

1. 按一下此鈕便開始輸入第一個數值的個位數，第一次為0，再按一下為1，以此類推到9之後，再按一次變回到0，再按到你想要的數字，放開按鈕並等待。

2. 放開按鈕後，會聽到叮聲，再按此鈕便會讓原本的個位數值變成十位數，再重新執行步驟1，再聽到叮聲，會讓原本的個位數值變成十位數、十位數值變成百位數，依此類推，直到選定自己要的數值後，放開按鈕等待，直到聽到提示聲。

3. 聽到提示聲後，便開始輸入加、減、乘、除，按第一次為加，再按一次為減，依此類推，到除之後，再按一次便回到加，選到想要的運算符號，放開按鈕並等待。

4. 等聽到提示聲，即可輸入第二個數值，輸入方法與輸入第一個數值相同 (步驟1和步驟2)。

5. 聽到提示聲之後，按下此鈕，便能得到計算出的答案。

6. 再按此鈕，便讓步驟5所算出的數值變成第一個數值，且在此時使用者可重複步驟3到步驟5的程序，便得到新的值，依此類推，直到使用者求出最後的數值。



按鈕3：

按此鈕後，便可進入小數點輸入模式。

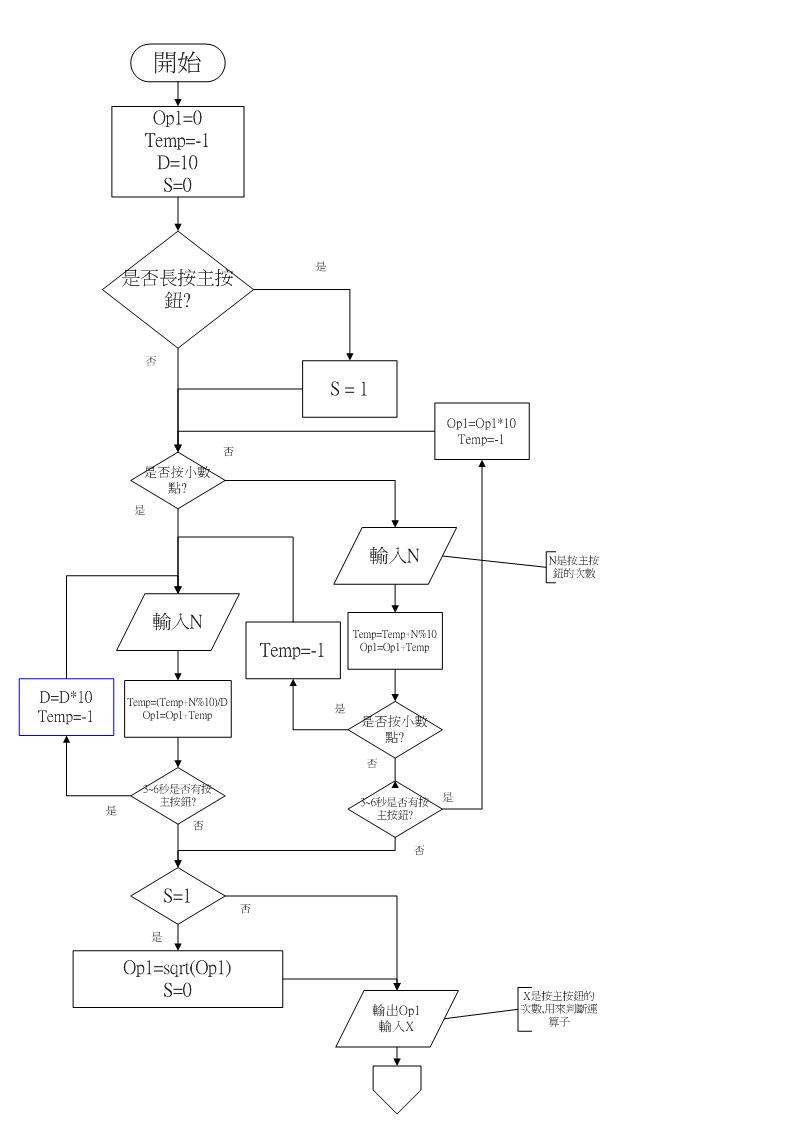
(長按)按鈕1：

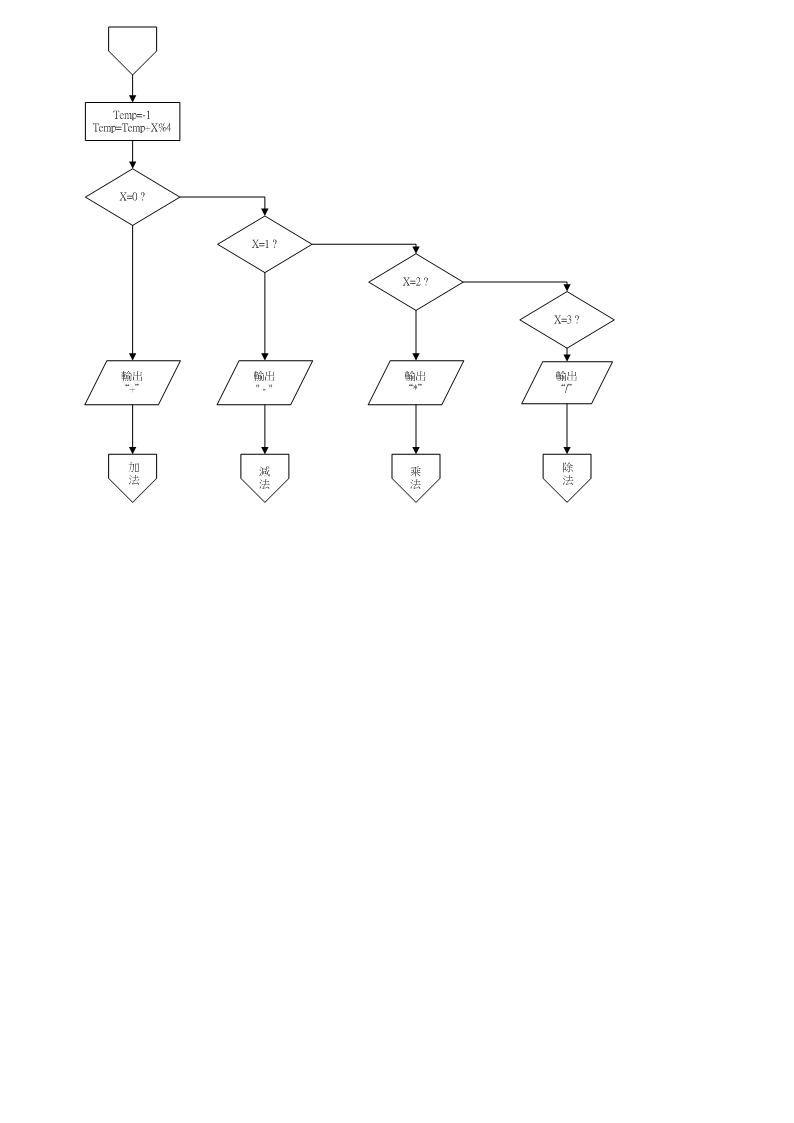
在第一個和第二個數值之前，長按此鈕後，便會進入根號模式。

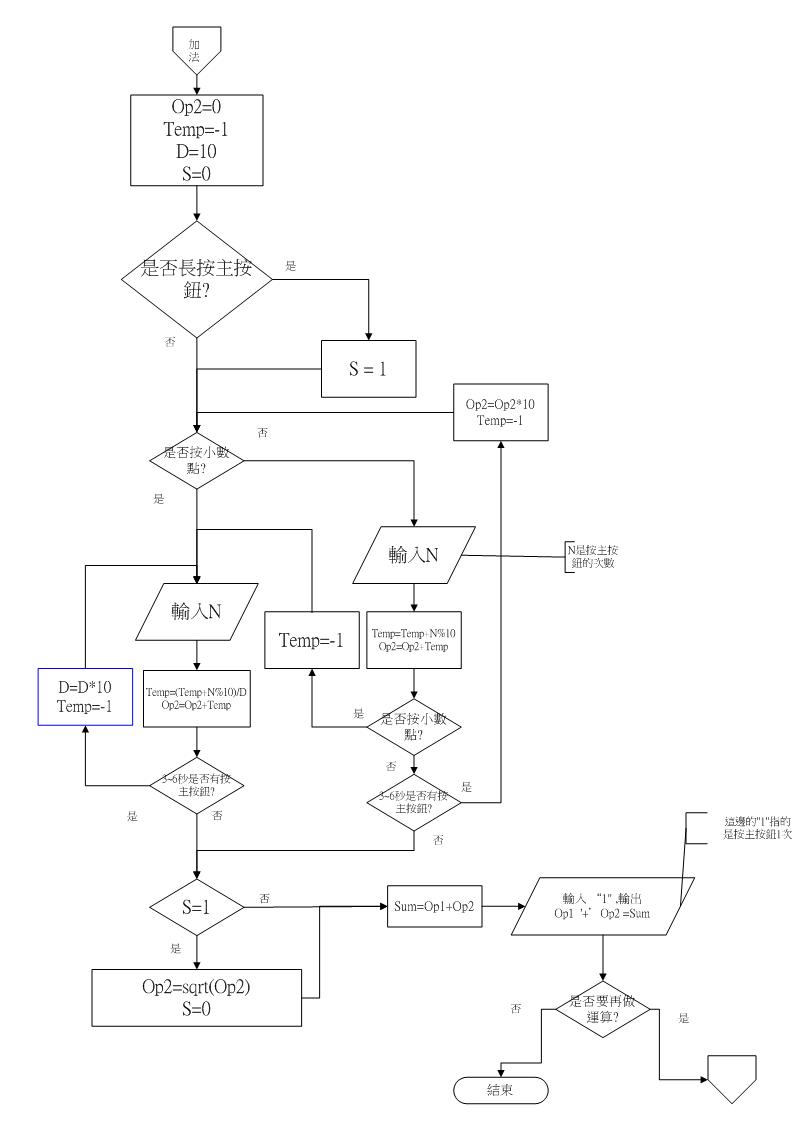
按鈕2：

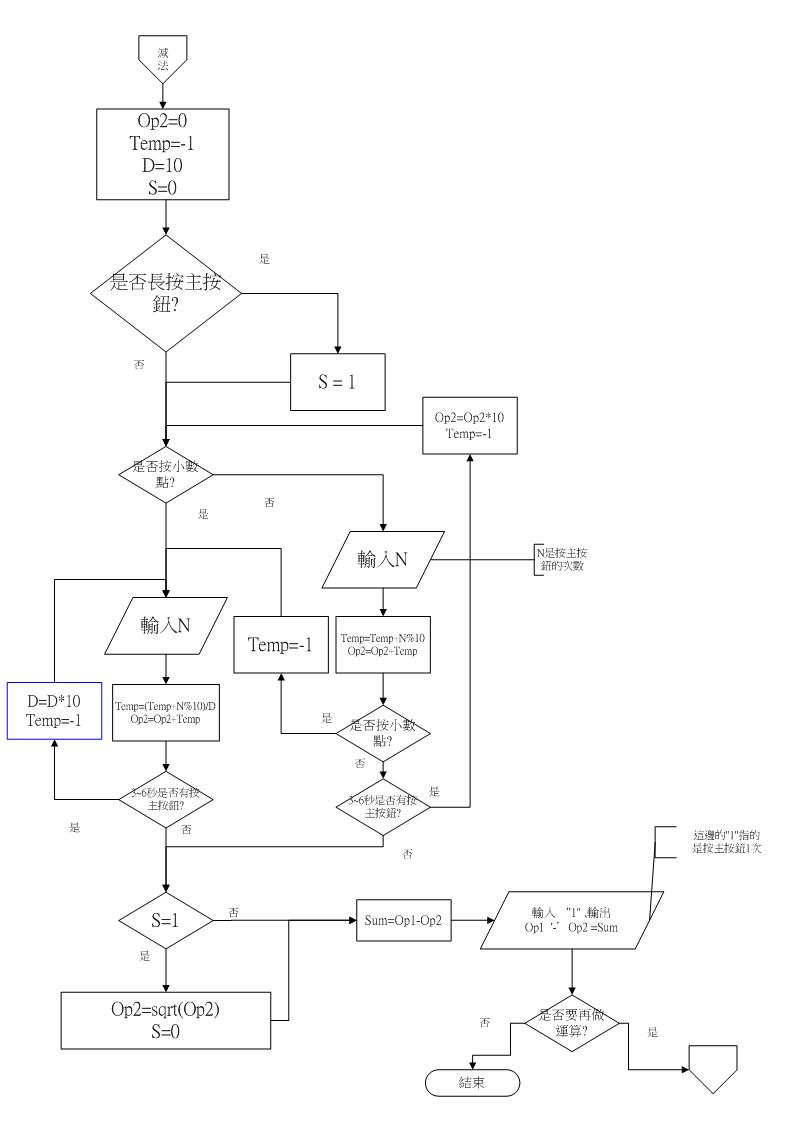
按此鈕後，便會將所有數值歸為0

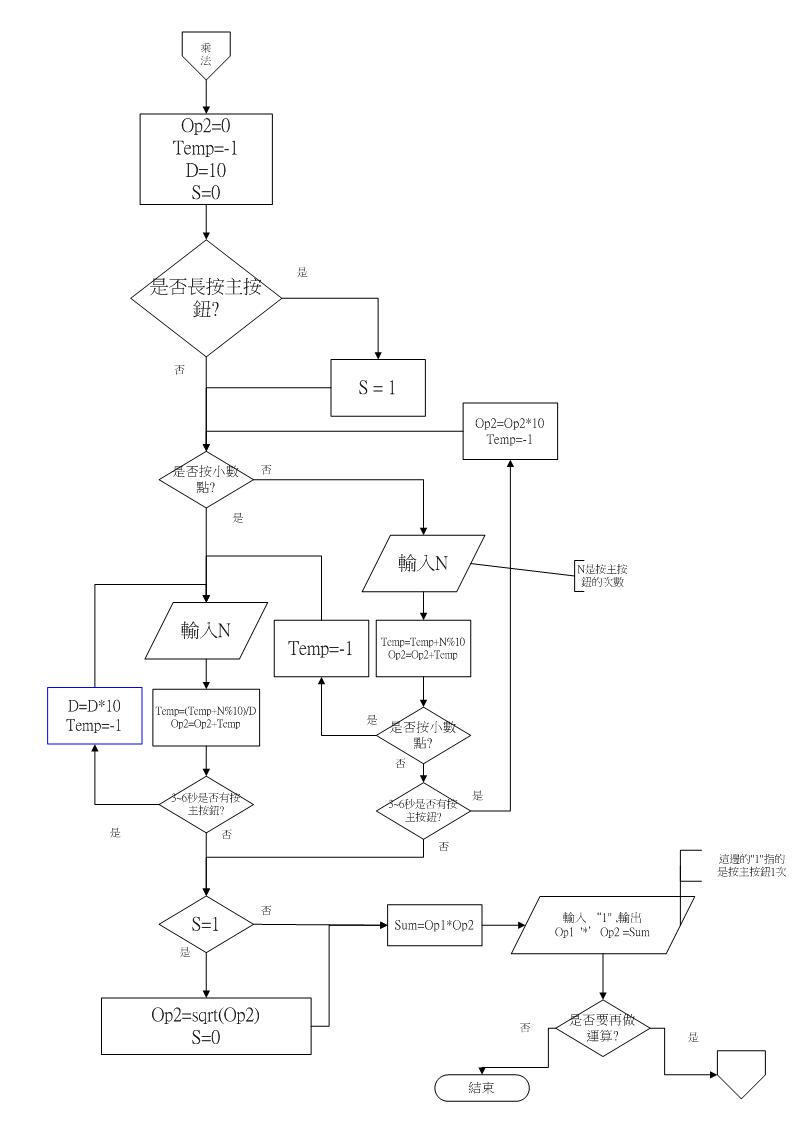
1. 流程圖

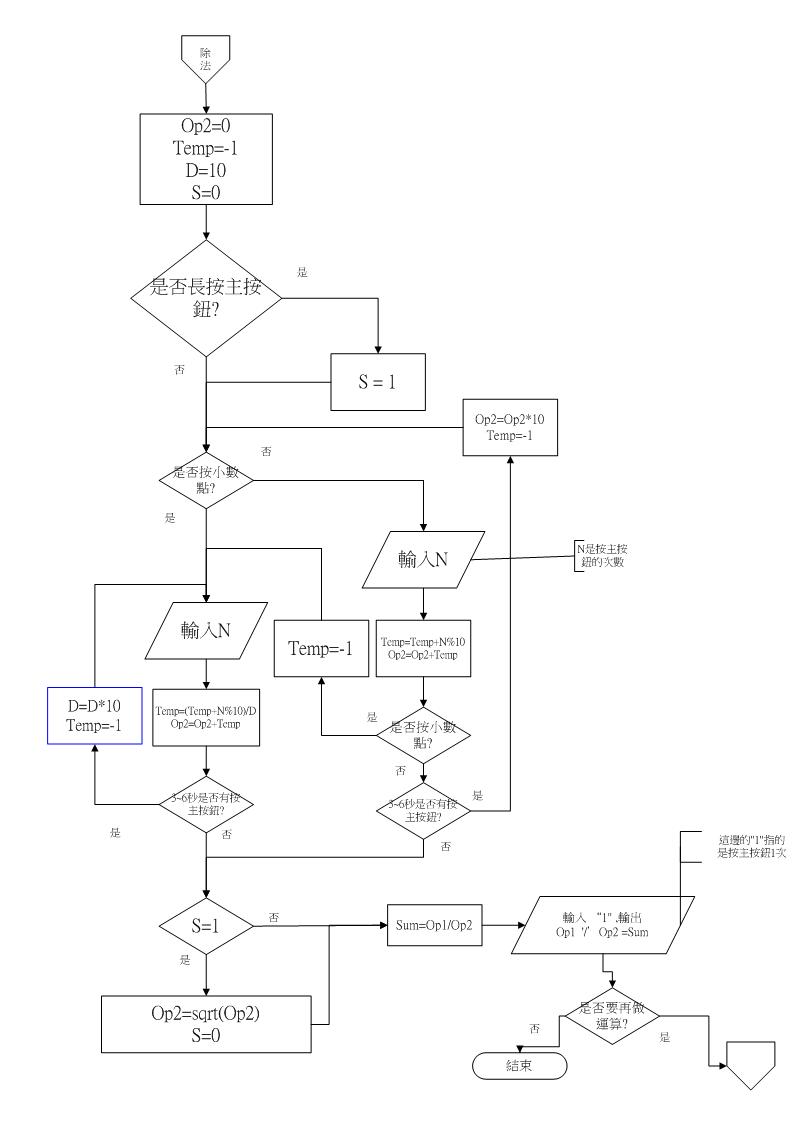
總流程圖



加法運算

減法運算

乘法運算

除法運算

實例說明

Ex1. (55 + 17) ÷ 2= 36

Step1: 按鈕1連續按6次，等待“叮”聲後，便再連續按6次，即可輸入第一運算元“55”，再放開按鈕1並等待“提示聲”。

Step2: 聽到“提示聲”後，按1下按鈕1(為“加號”)，再放開並等待“提示聲”。

Step3: 按鈕1連續按2次，等待“叮”聲後，便再連續按8次，即可輸入第二運算元“17”，再放開按鈕1並等待“提示聲”。

Step4: 聽到“提示聲”後，再按1次按鈕1，便會語音輸出(55+17)的運算結果(為72)。

Step5: 如果想繼續運算，則可按一下按鈕1(為再加)，再按1下(為再減)，再按一次按鈕1(為再乘)，再按一次按鈕1(為再除)，此題為再除，因此放開按鈕1並等待“提示聲”。

Step6: 按鈕1連續按3次，即可輸入第二運算元“2”，再放開按鈕1並等待“提示聲”。

Step7: 聽到“提示聲”後，再按1次按鈕1，便會語音輸出(55+17)÷2的運算結果(為36)。

Ex2. √64 + 14.42= 22.42

Step1: 長按按鈕1，則進入根號模式，再繼續按按鈕1連續按7次，等待“叮”聲後，便再連續按5次，即可輸入第一運算元“√64”，再放開按鈕1並等待“提示聲”。

Step2: 聽到“提示聲”後，按1下按鈕1(為“加號”)，再放開並等待“提示聲”。

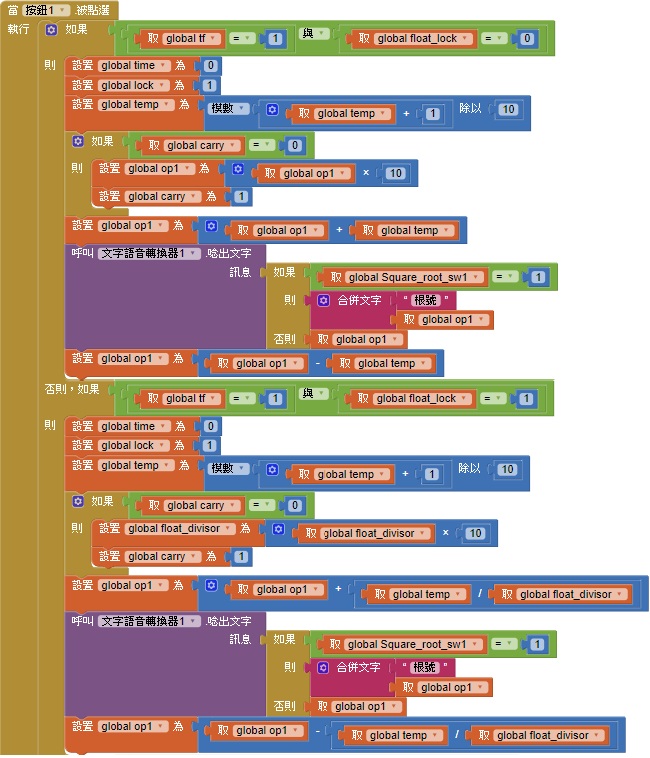
Step3: 按鈕1連續按2次，等待“叮”聲後，便再連續按5次後，便按按鈕3進入小數點輸入模式，再按鈕1連續按5次，等待“叮”聲後，便再連續按3次後，即可輸入第二運算元“14.42”，再放開按鈕1並等待“提示聲”。

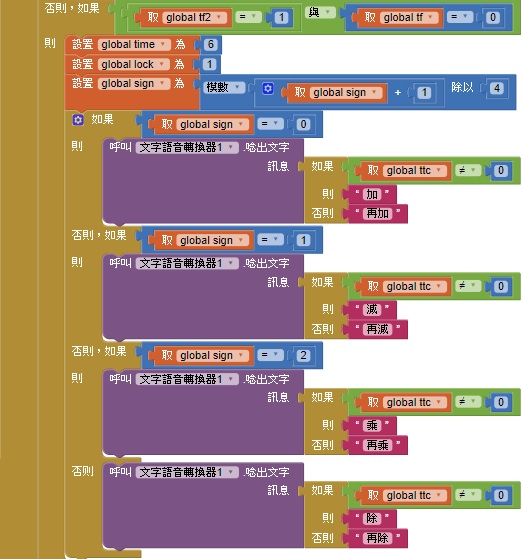
Step4: 聽到“提示聲”後，再按1次按鈕1，便會語音輸出(√64 + 14.42)的運算結果(為22.42)。

方塊程式圖解說明

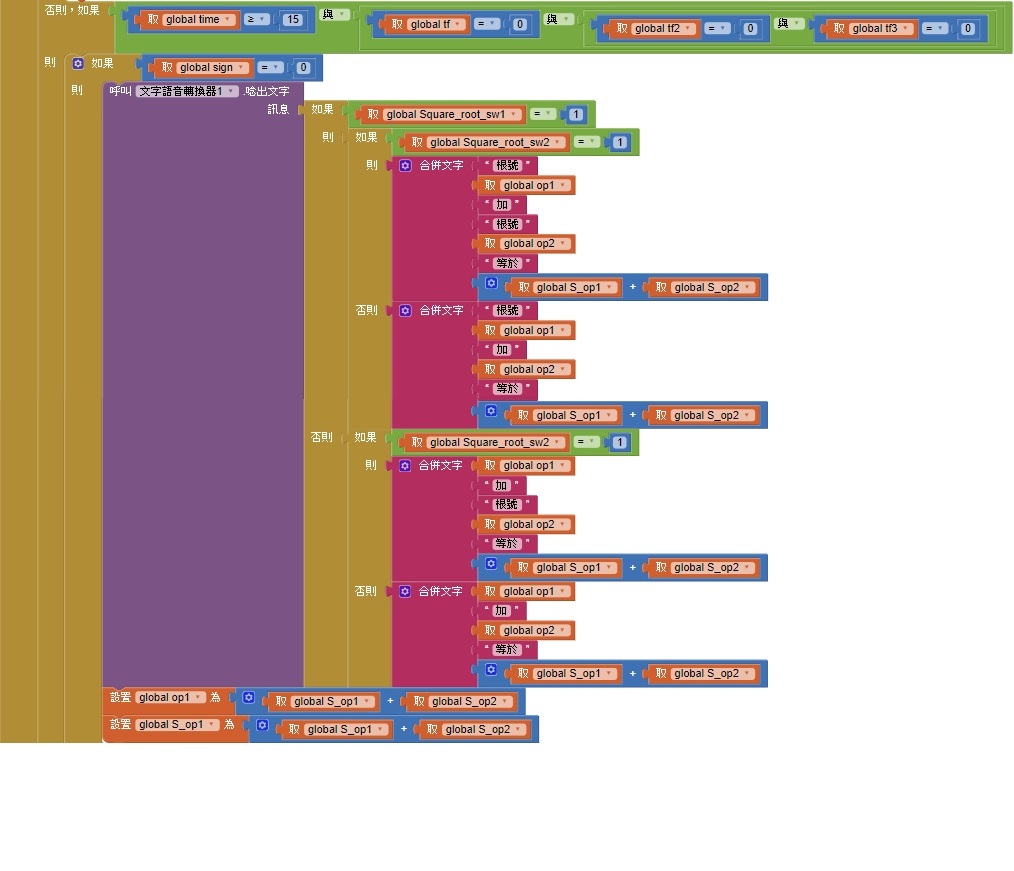
SCREEN

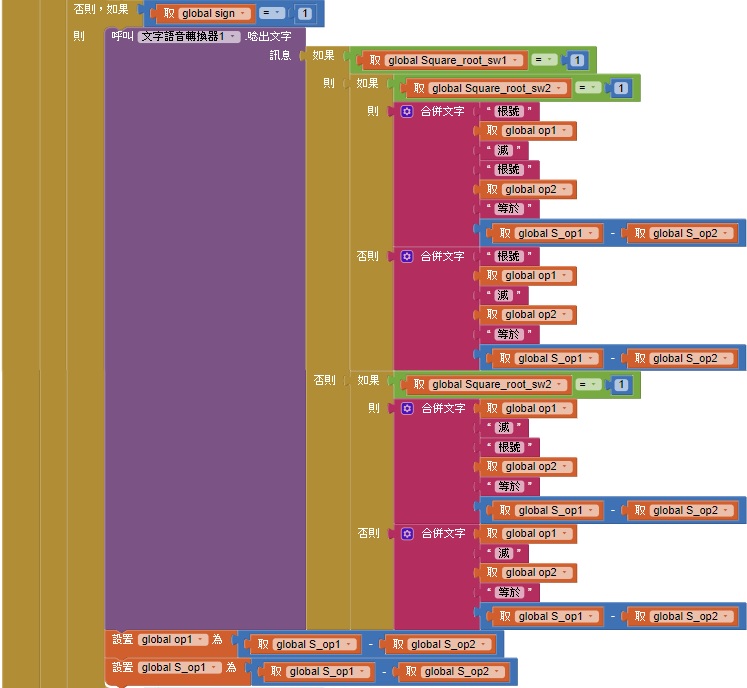
變數

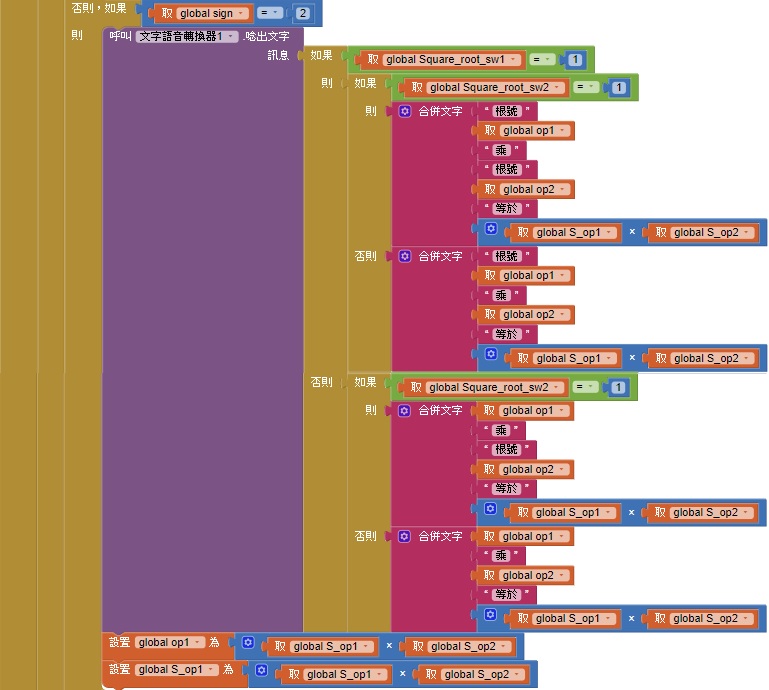
第一運算元輸入

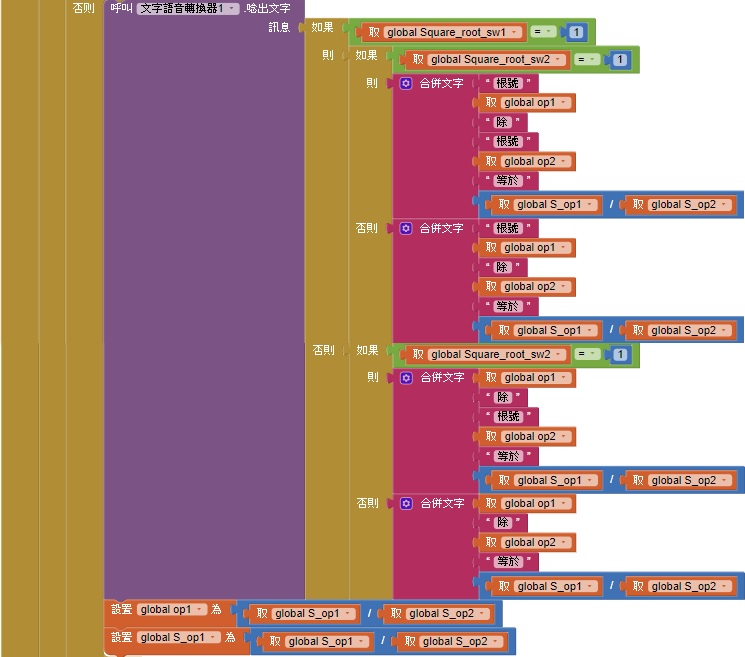
運算子輸入

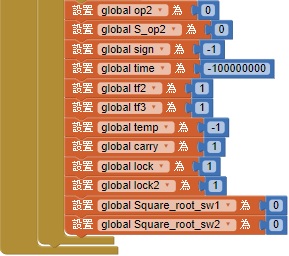
第二運算元輸入

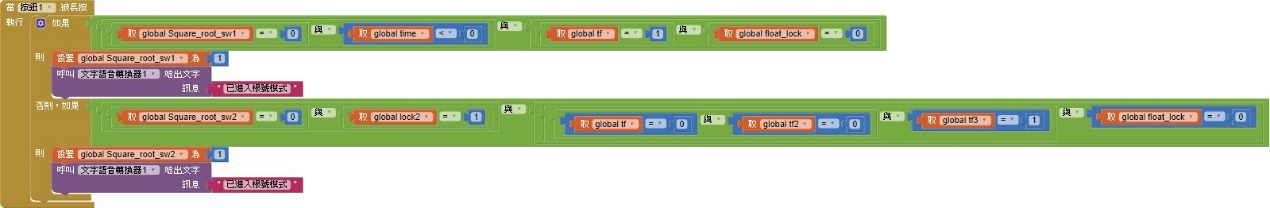
運算結果輸出(加法)

運算結果輸出(減法)

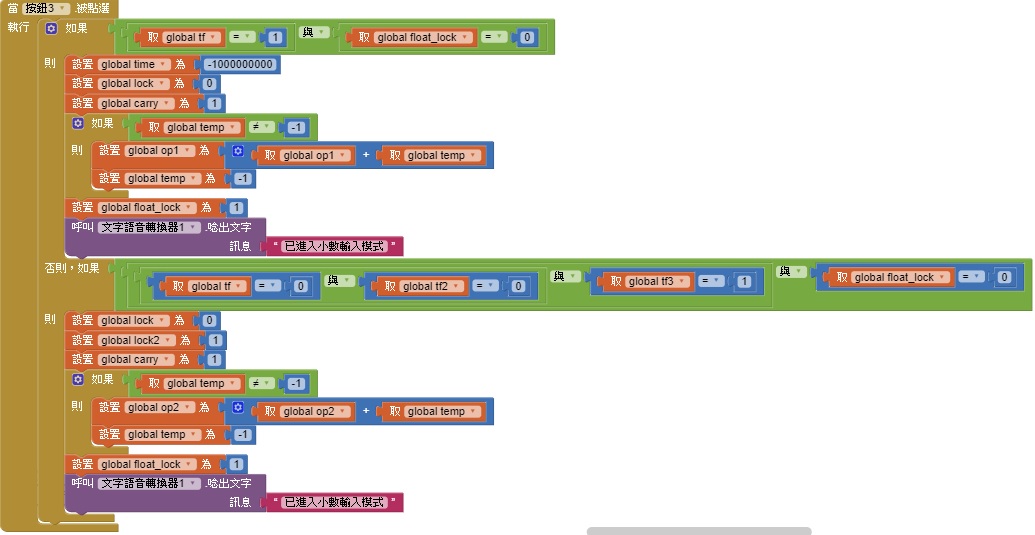
運算結果輸出(乘法)

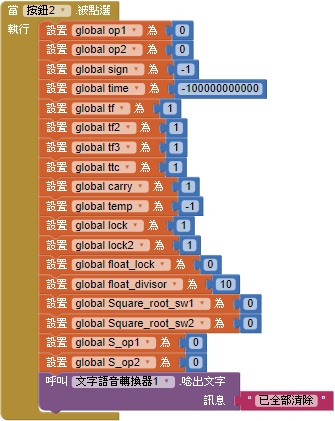
運算結果輸出(除法)

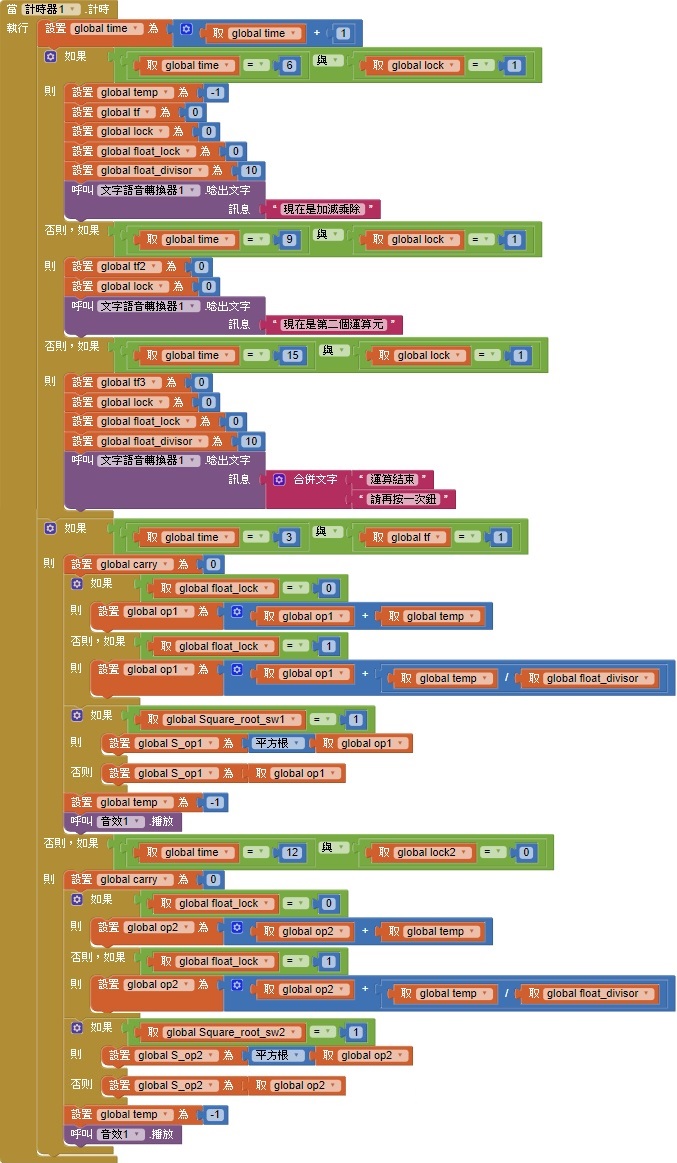
重設變數初始值

根號判斷

小數點判斷



清除

計時器

結論

本專題利用「自動機原理技術」設計一個在智慧型手機盲人計算機APP，讓盲人能更簡單、方便的計算四則運算，但目前尚未完整的常駐程式開發，。期許在將來如有機會參加校外比賽或有學弟、妹願意接手本專題，有更多的時間討論並完善。

參考文獻

參考文獻

1. V., J.L.C., et al., *A real-time robust indoor tracking system in smartphones.* Computer Communications, 2017. **1**(1): p. 1-12.

2. Chittaro, L. and R. Sioni, *An electromyographic study of a laser pointer-style device vs. mouse and keyboard in an object arrangement task on a large screen.* Int. J. Human-Computer Studies, 2012. **70**(2): p. 234-255.

3. Baker, N. *App aims to improve accessibility for visually impaired*. TECHNOLOGY NEWS 2012; Available from: <http://www.reuters.com/article/us-app-blind-idUSBRE86M09320120723>.

4. Carvalhoa, R.M., R.M.d.C. Andradea, and K.M.d.O. b, *AQUArIUM - A suite of software measures for HCI quality evaluation of ubiquitous mobile applications.* The Journal of Systems and Software, 2018. **136**(1): p. 101-136.

5. Waqar, W., Y. Chen, and A. Vardy, *Smartphone positioning in sparse Wi-Fi environments.* Computer Communications, 2016. **73**(1): p. 108-117.

6. Tetsuhei Nakashima-Paniagua, J.D., Walied Moussa, *Fabrication process suitability ranking for micro-electro-mechanical systems using a fuzzy inference system.* Expert systems with applications, 2014. **41**(9): p. 4123-4138.

7. Allen, F.R. and E. Ambikairajah, *Classification of a known sequence of motions and postures from accelerometry data using adapted Gaussian mixture models.* Physiological Measurement, 2006. **27**(10): p. 935-951.

8. Ayu, M.A., et al., *A comparison study of classifier algorithms for mobile-phone's accelerometer based activity recognititon.* Procedia Engineering, 2012. **41**: p. 224-229.

9. Fuentes, D. and L. Gonzalez-Abril, *Online motion recognition using an accelerometer in a mobile device.* Expert Systems with Applications, 2012. **39**: p. 2461-2465.

10. Paoli, R., et al., *A system for ubiquitous fall monitoring at home via a wireless sensor network and a wearable mote.* Expert Systems with Applications, 2012. **39**: p. 5566-5575.

11. Kwapisz, J.R., G.M. Weiss, and S.A. Moore, *Activity recognition using cell phone accelerometers.* SIGKDD Explorations Newsletter, 2010. **12**(2): p. 74-82.

12. Kwon, Y., K. Kang, and C. Bae, *Unsupervised learning for human activity recognition using smartphone sensors.* Expert Systems with Applications, 2014. **41**: p. 6067-6074.

13. Kim, D.-W., et al., *Efficient dynamic time warping for 3D handwriting recognition using gyroscope equipped smartphones.* Expert Systems with Applications, 2014. **41**(3): p. 5180-5189.

14. Vua, T.H.N., et al., *Online discovery of Heart Rate Variability patterns in mobile healthcare services.* The Journal of Systems and Software, 2010. **83**(6): p. 1930-1940.

15. Rodriguez-Martin, D., *SVM-based posture identification with a single waist-located triaxial accelerometer.* Expert Systems with Applications, 2013. **40**: p. 7203-7211.

16. D. Fuentes, L.G.-A., C. Angulo, J.A. Ortega, *Online motion recognition using an accelerometer in a mobile device.* Expert systems with applications, 2012. **39**(3): p. 2461-2465.

17. Ömer Faruk Ertugrul ˘ a, et al., *Detection of Parkinson’s disease by Shifted One Dimensional Local Binary Patterns from gait.* Expert systems with applications, 2016. **59**(1): p. 156-163.

18. Daniel Rodriguez-Martin, A.S., Carlos Perez-Lopez, Andreu Català, Joan Cabestany, Alejandro Rodriguez-Molinero, *SVM-based posture identification with a single waist-located triaxial accelerometer.* Expert systems with applications, 2013. **40**(18): p. 7203-7211.