馬達與感測器教具平台

與

程式設計演算法及流程圖

教材來源

FabLab-University

課程圖書館 [總恆星基地]

類別： 程式及數位控制技術類 標題： 關鍵字搜尋 搜尋

課程類別	發佈單位	標題	發表日期
程式及數位控制技術類	國立高雄師範大學中心基地	程式流程圖教學公版教材(修訂版v1)	2019.09.27
程式及數位控制技術類	國立高雄師範大學中心基地	NKNUBLOCK 安裝檔	2019.08.22
程式及數位控制技術類	國立高雄師範大學中心基地	NKNUBLOCK網路架構	2019.08.14
程式及數位控制技術類	國立高雄師範大學中心基地	NKNU-Scratch-WiFi 創意發想空間環境建置手冊 朱建華老師(教育部)	2018.11.22
程式及數位控制技術類	國立高雄師範大學中心基地	NKNU-Scratch-WiFi 數位教室環境建置手冊 朱建華老師(教育部)	2018.11.22

學習程式語言的目的

- 讓電腦為我們做事情。
- 要用電腦的語言告訴它。
- 用電腦的邏輯思考。
- 電腦的邏輯思考跟人類有什麼差別？

情境任務分析範例

你過來

1.叫我嗎？
2.誰叫我？
3.要過去嗎？
4.起身
5.選擇前進路線
6.前進
7.停止

偵測聲音、判斷語意
偵測影像、判斷影像
要過去嗎？
起身
偵測影像、決定路徑
前進
……

情境分析與情境流程圖

- **情境分析：**在老師與學生的「**問與答**」中，確認學生對主題情境的邏輯程序的了解，是一種運算思維的訓練。
- 情境分析結果的記錄整理工具：
  - 情境流程圖
  - 演算法步驟或虛擬碼
  - 程式流程圖

例子：讓LED亮慢慢變亮

1. LED接在哪裡？
2. 一開始LED的亮度是？
3. 控制LED變亮的是哪一個功能？
4. LED最亮可以多亮？
5. 慢慢變亮是多慢呢？
6. 變亮每次變多少呢？

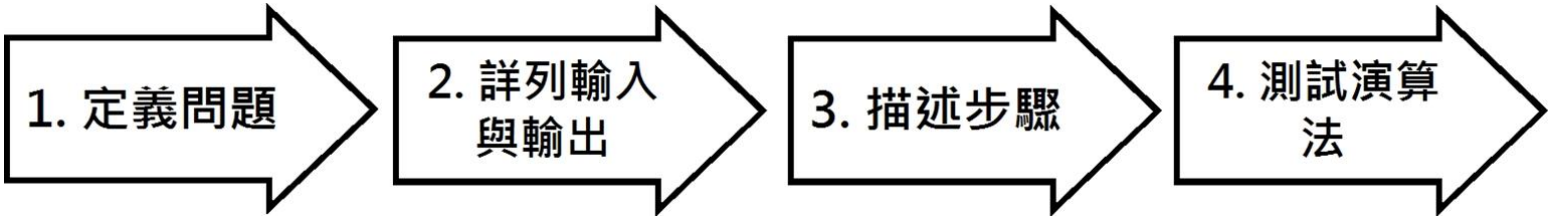
演算法步驟簡介

Step 1. **定義問題**：使用明確和簡潔的詞彙來描述欲解決的問題。

Step 2. **詳列輸入與輸出**：列出欲解決問題的資料（input），和經過演算法運算後，需要產生的結果（output）。

Step 3. **描述步驟**：描述從輸入資料轉換成輸出資訊的步驟。

Step 4. **測試演算法**：使用測試資料來驗證演算法是否正確。



演算法表達方法簡介

演算法是在描述解決問題的步驟，並沒有固定方法，常用的表達方法有：

1. 演算法步驟：直接使用一般語言文字，條列式描述來說明執行步驟
2. 虛擬碼（pseudo code）：一種趨近程式語言的描述方法，並沒有固定語法，每一行約可轉換成一行程式碼，如下所示：

演算法步驟

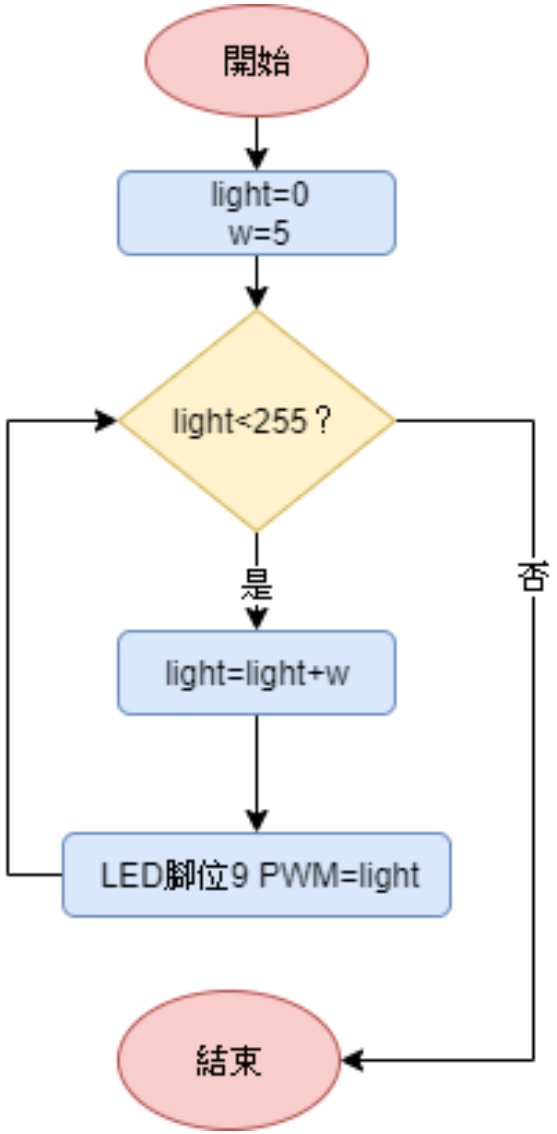
1.初始LED亮度變數i=0(全暗)
2.LED亮度增減變量值x=5
3.判斷迴圈計算是否i=255
  3-1.若否，LED做PWM輸出
    3-1-1.計算i=i+x(亮度漸亮)
    3-1-2.延遲30毫秒後，繼續迴圈步驟3
  3-2.若是，LED全亮，結束迴圈至步驟4

虛擬碼

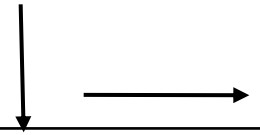
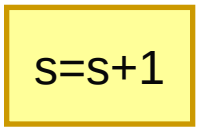
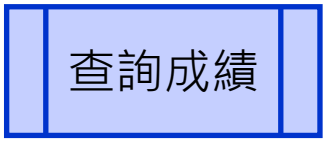
```
/* 計算1加到10 */
Let counter = 1
Let sum = 0
while counter <= 10
    sum = sum + counter
    Add 1 to counter
Output the sum      /* 顯示結果 */
```

程式流程圖簡介

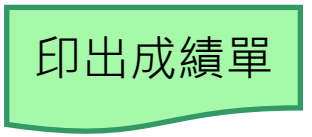
- 「程式流程圖」則是依據「情境流程圖」，導入積木程式堆疊的「邏輯程序細節設計」



流程圖符號(1)

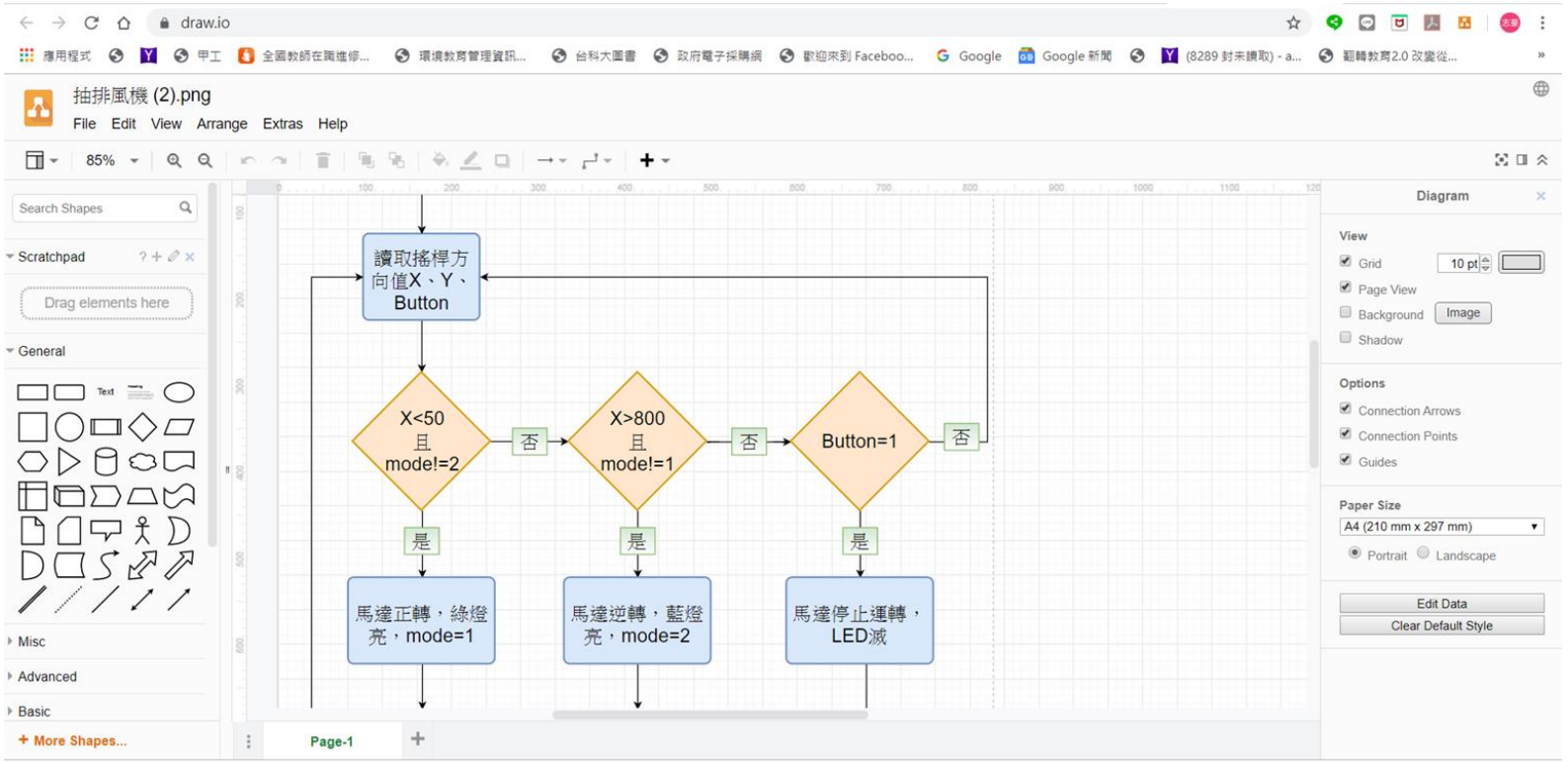
符號	名　　稱	意　　義	範　　例
	開始 (Start) 終止 (End)	表示程式的開始或結束	
	路徑(Path)	表示流程進行的方向	
	決策判斷 (Decision)	針對某一條件進行判斷	
	處理(Process)	表示執行或處理某一項工作	
	副程式 (Subroutine)	用以表示一群已經定義流程的組合	

流程圖符號(2)

符號	名　　稱	意　　義	範　　例
	輸入(Input) 輸出(Output)	表示資料的輸入或結果的輸出	
	報表(Document)	指列印出的報表文件	

流程圖繪製軟體
draw.io

- www.draw.io



練習一 安全風扇

- 情境主題：安全風扇
- 情境目的：結合超音波感測器與風扇的運轉，利用超音波感測器感測到有人員接近時，可以快速將風扇立即停止，以保護人員的安全。
- 情境分析：

1.設定超音波感測器與人員之間的距離為H，

2.如果距離30公分以上，風扇高速運轉，綠燈亮(安全區)

3.介於10~30公分 之間，風扇低速運轉，藍燈亮且發出嗶嗶聲響(警戒區)

4. 低於10公分，風扇立即停止且發出急促嗶嗶聲響，紅燈亮(危險區)。

選擇結構 (安全風扇)

演算法步驟

1.讀取超音波感測器量測距離H

2.判斷 H > 30

2-1. 成立：風扇高速運轉，綠燈亮，執行後再回至步驟1

3.否則：

3-1. 不成立：再判斷 10 < H < 30

3-1-1. 成立：風扇低速運轉，藍燈亮，發出嗶嗶聲響，執行後再回至步驟1

3-2.否則：

3-2-1. 不成立：再判斷H>0(*)

3-2-1-1.成立：風扇停止，紅燈亮，發出急促嗶嗶聲響，執行後再回至步驟1

*(超音波感測器測不出距離時會傳回0，所以要把0濾除掉)

選擇結構 (安全風扇)

演算法步驟 vs 程式流程圖

演算法步驟

1.讀取超音波感測器量測距離H

2.判斷 H > 30

2-1. 成立：風扇高速運轉，綠燈亮，執行後再回至步驟1

3.否則：

3-1. 不成立：再判斷 10 < H < 30

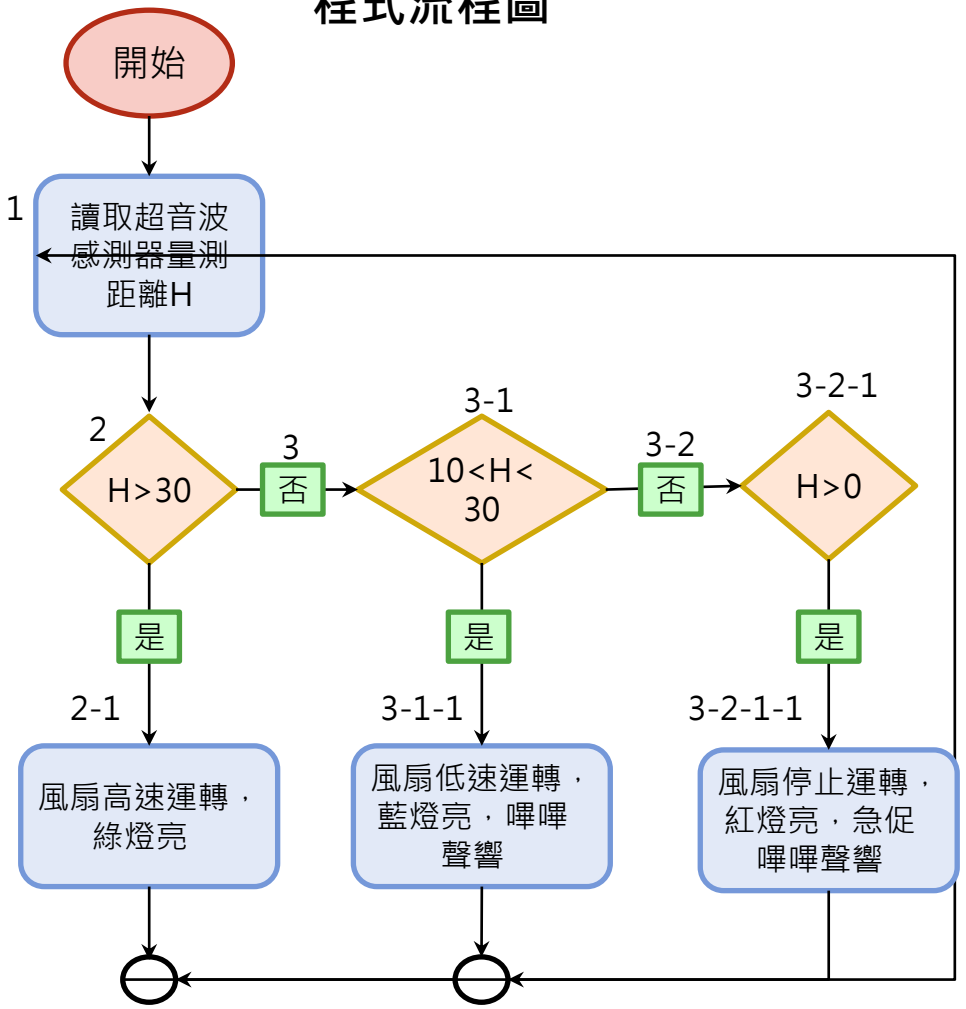
3-1-1. 成立：風扇低速運轉，藍燈亮發出嗶嗶聲響，執行後再回至步驟1

3-2. 否則：

3-2-1. 不成立：再判斷H>0

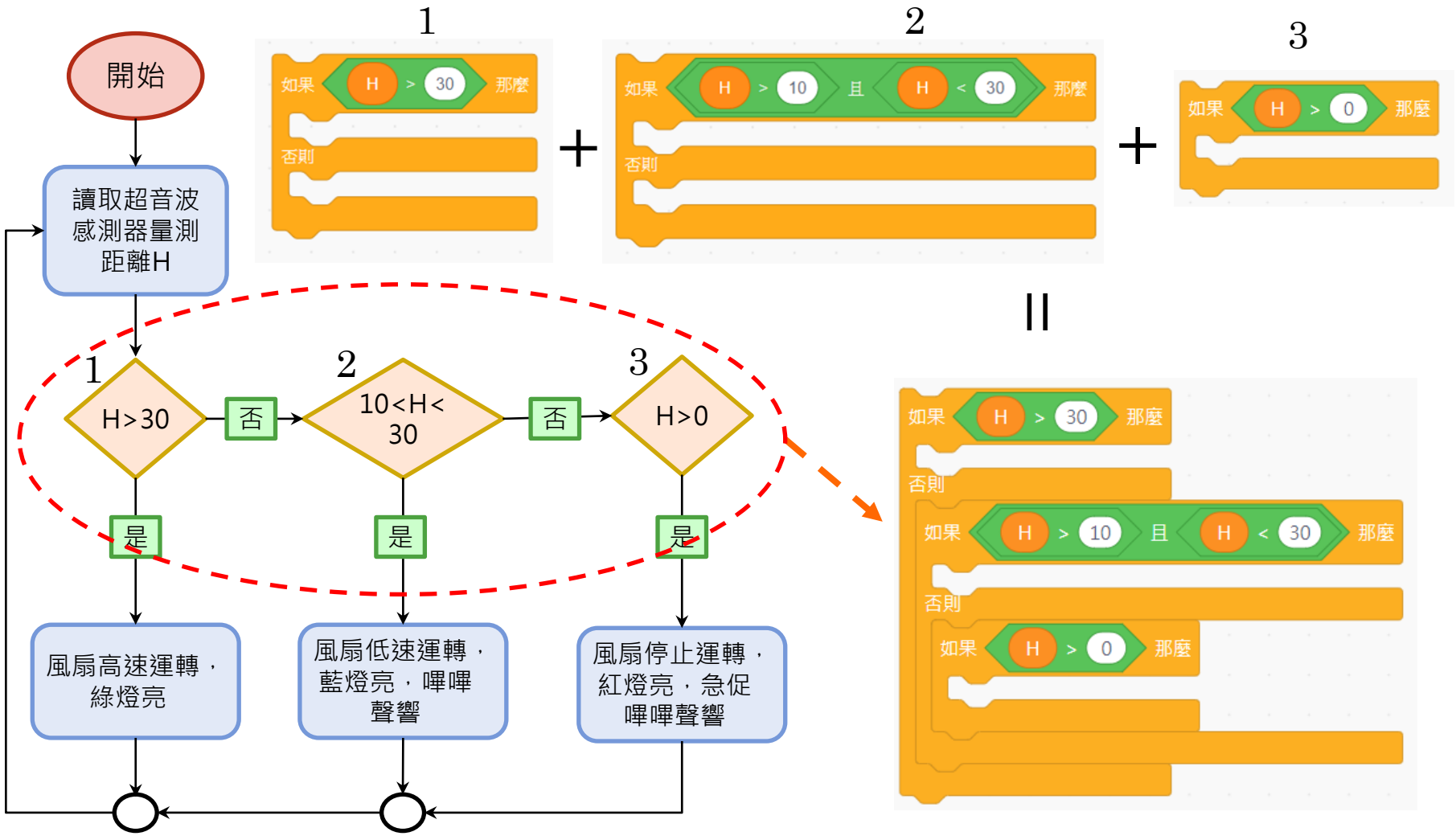
3-2-1-1.成立：風扇停止，紅燈亮，發出急促嗶嗶聲響，執行後再回至步驟1

程式流程圖



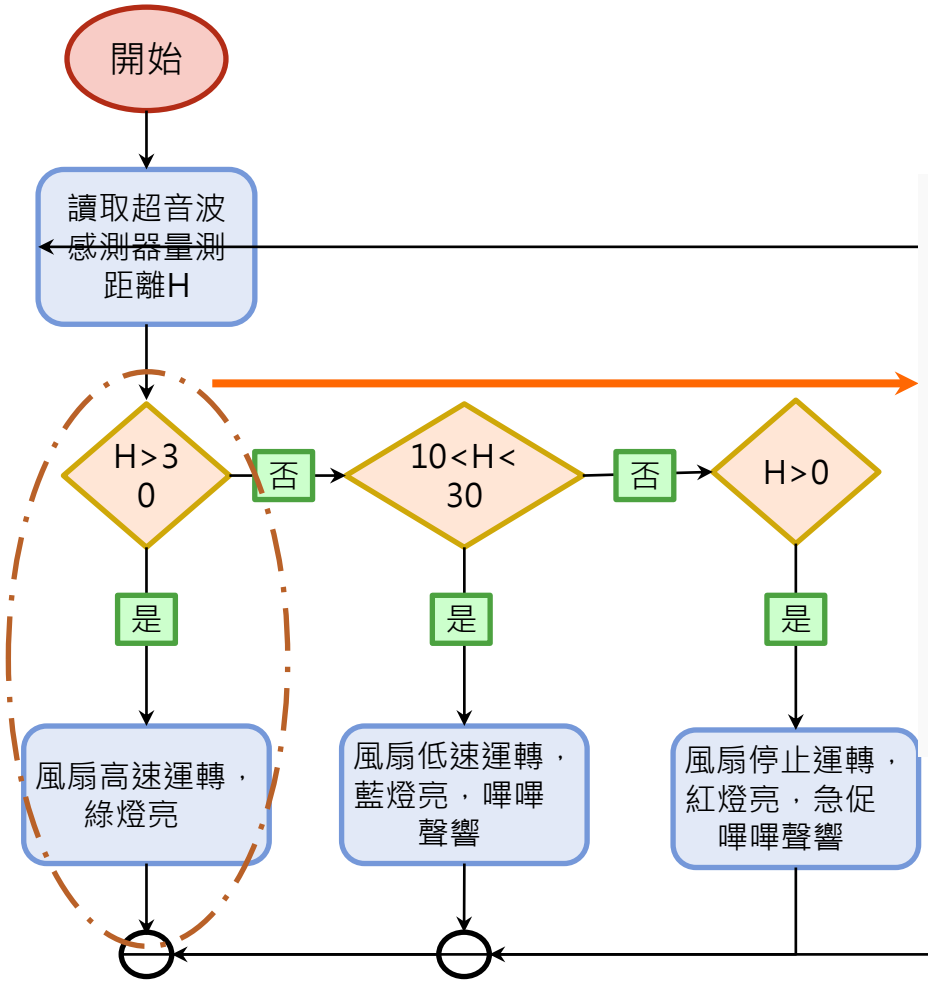
選擇結構 (安全風扇)

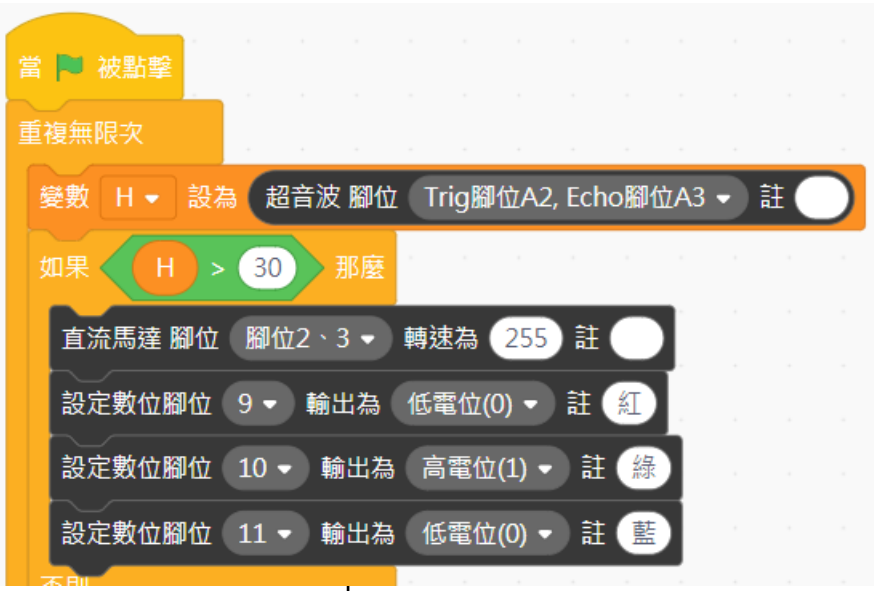
程式流程圖 vs 積木程式堆疊



選擇結構 (安全風扇)

程式流程圖 vs 積木程式堆疊

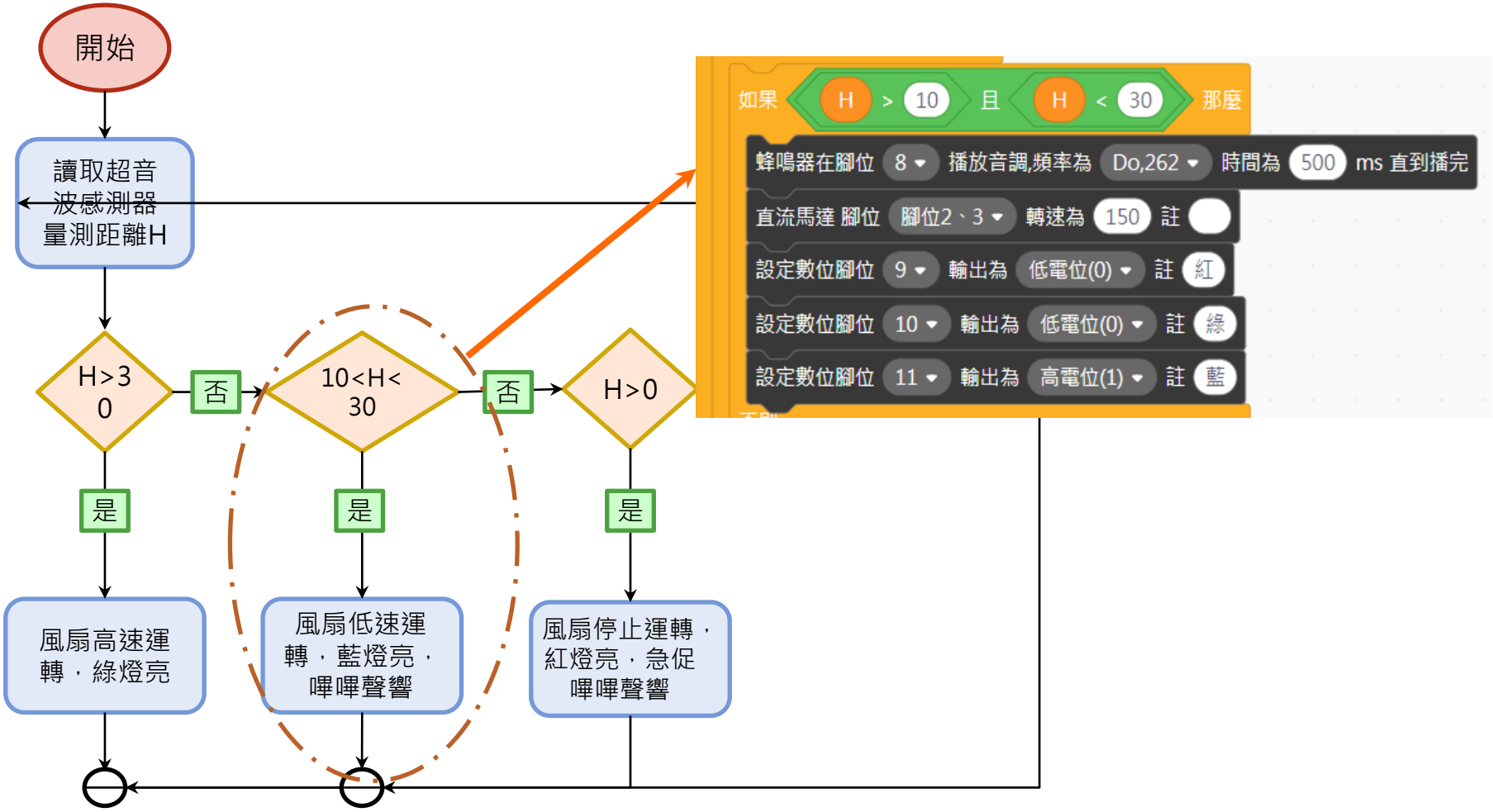




Ch.4.3.6.3

選擇結構 (安全風扇)

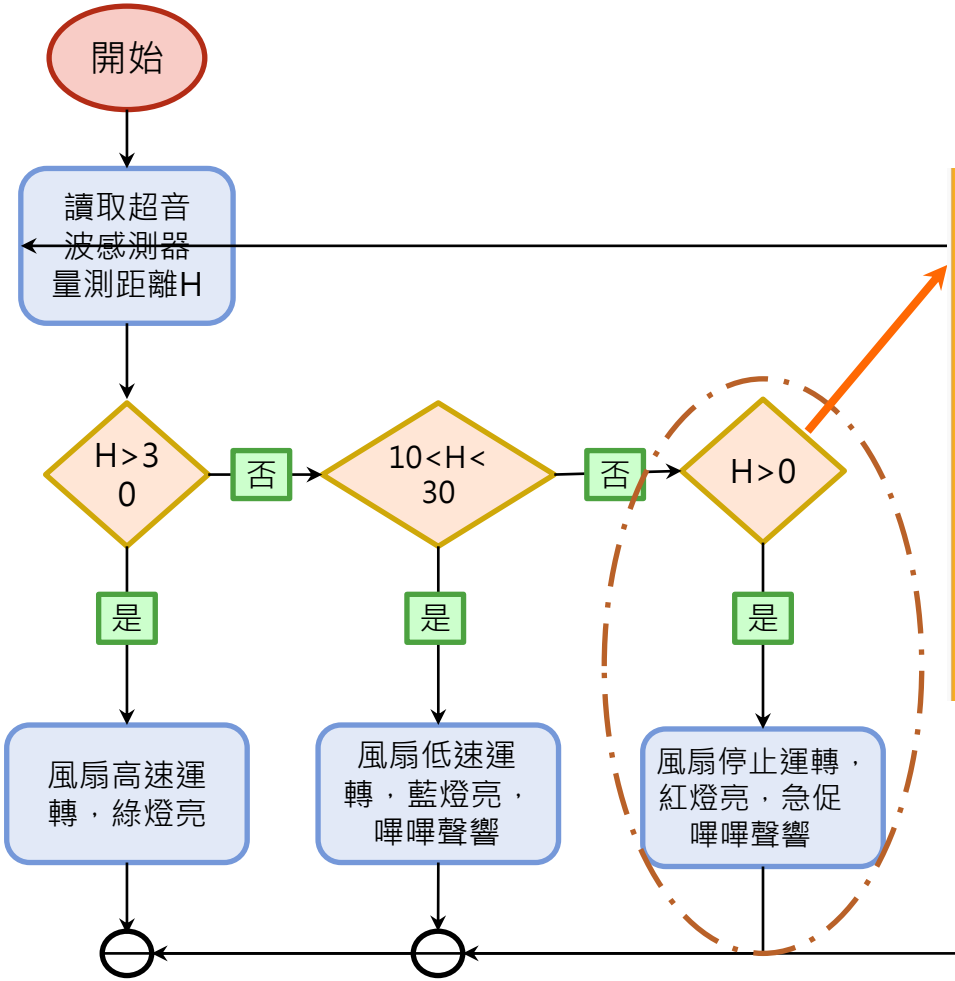
程式流程圖 vs 積木程式堆疊



Ch.4.3.6.4

選擇結構 (安全風扇)

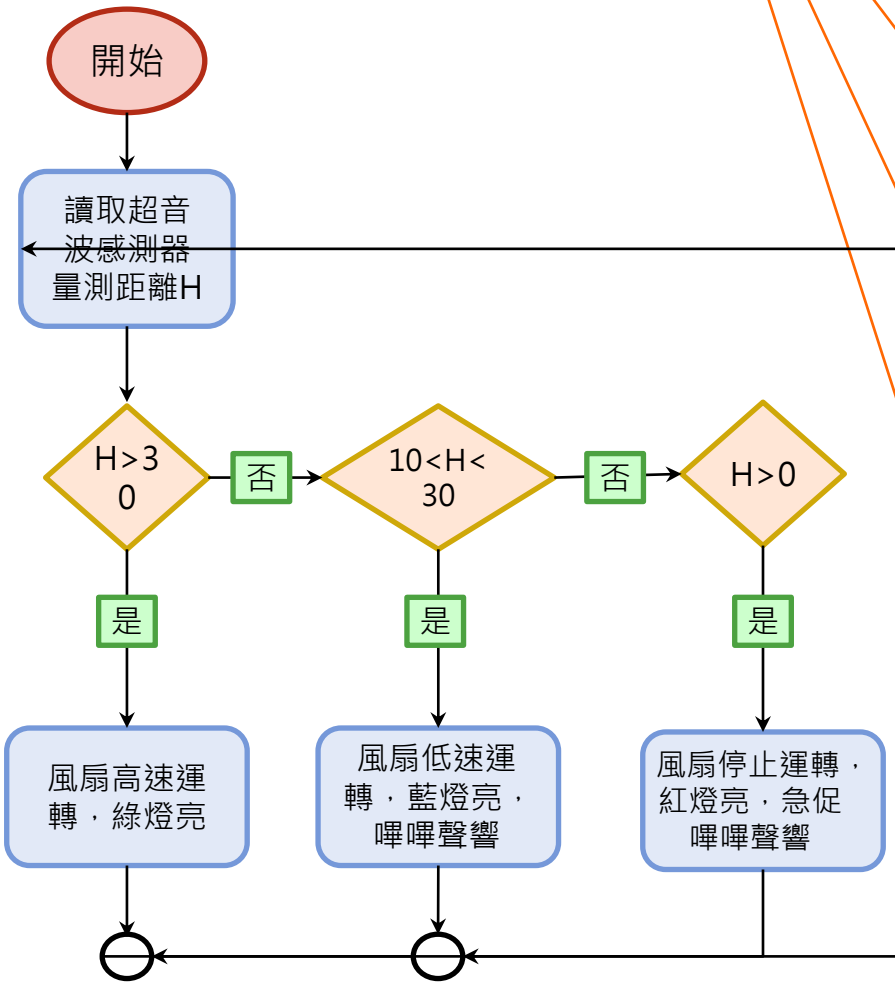
程式流程圖 vs 積木程式堆疊



Ch.4.3.7.1

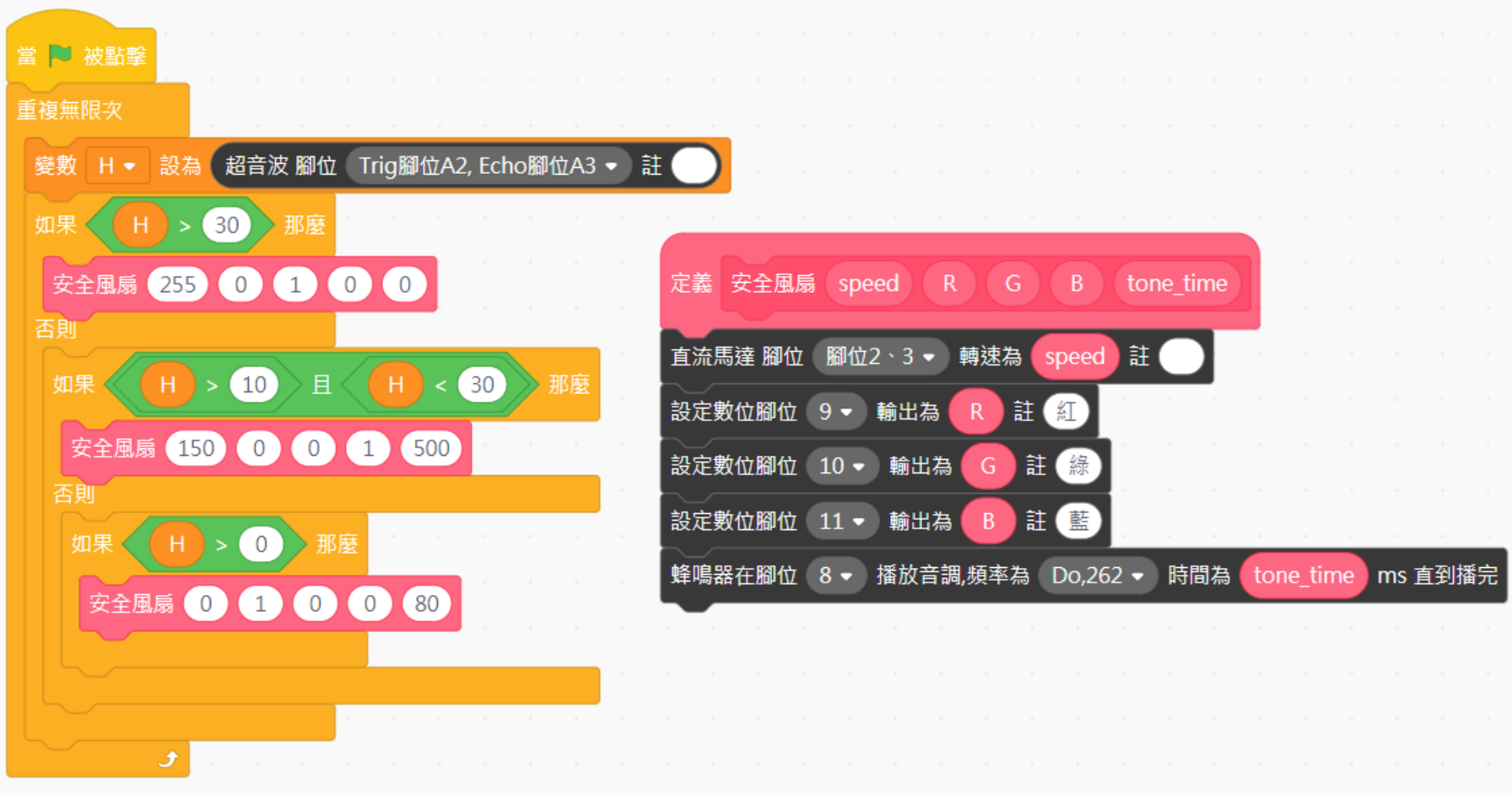
選擇結構 (安全風扇)

以「副程式」來呈現



Ch.4.3.7.2

選擇結構 (安全風扇)
以「副程式」來呈現



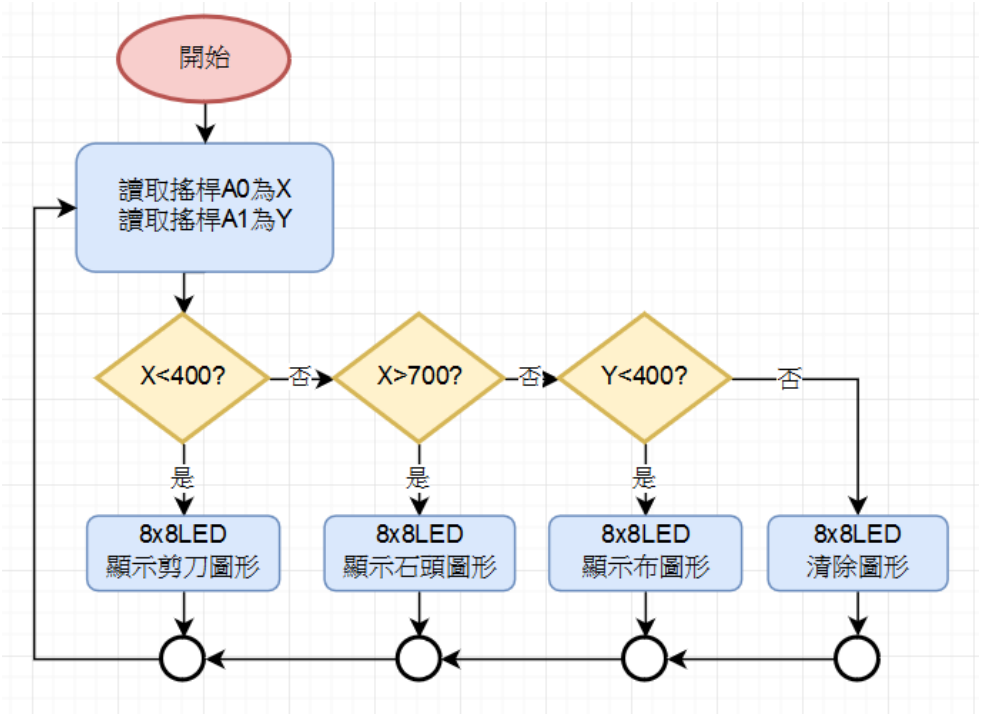
練習二 猜拳機

- 情境主題：猜拳機
- 情境目的：利用平板上的模組製作剪刀、石頭、布猜拳機，下課前做完，並用猜拳機猜拳，輸的3組留下整理教室，出不了拳的也留下。
- 情境分析：
  1. 怎樣呈現剪刀、石頭、布的樣子？用哪一個模組？
  2. 怎樣出拳？用哪一個模組？出拳的方法？

練習二 猜拳機類型一

- 演算法步驟與流程圖：
  - 輸出：8X8矩陣，顯示剪刀、石頭、布的圖形
  - 輸入：搖桿，左搖剪刀、右搖石頭、前搖布、後搖清除圖形
- 演算法與流程圖

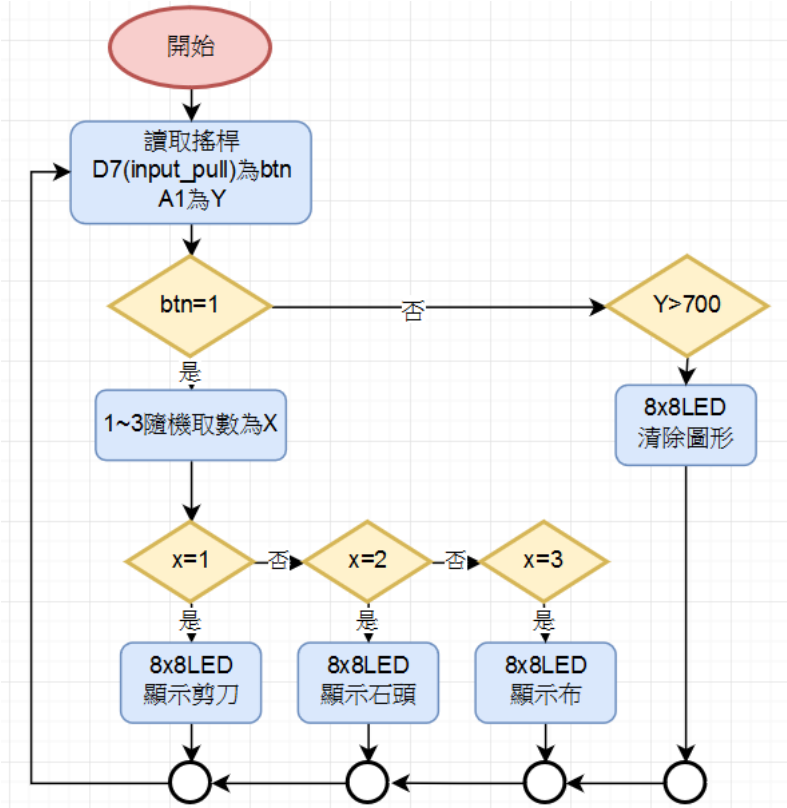
```
01  重覆執行
02      讀取搖桿A0為X
03      讀取搖桿A1為Y
04      如果X<400
05          8x8LED顯示剪刀
06      否則
07          如果X>700
08              8x8LED顯示石頭
09          否則
10              如果Y<400
11                  8x8LED顯示布
12              否則
13                  8x8LED清除圖形
14  結束重覆執行
```



練習二 猜拳機類型二

- 演算法步驟與流程圖：
  - 輸出：8X8矩陣，顯示剪刀、石頭、布的圖形
  - 輸入：搖桿，按下按鈕後開始隨機出現剪刀、石頭、布，直到放開按鈕，搖桿往下搖清除圖形。

```
01 重覆執行
02     讀取搖桿按鈕D7為btn
03     讀取搖桿A1為Y
04     如果btn=1
05         1~3隨機取數為X
06         如果 X=1
07             8x8LED顯示剪刀
08         如果 X=2
09             8x8LED顯示石頭
10         如果 X=3
11             8x8LED顯示布
12     否則
13         如果Y<400
14             8x8LED清除圖形
15 結束重覆執行
```



PHENAKISTOSCOPE(費納奇鏡)

- https://makezine.com/projects/animated-records-phenakistoscopes/
- https://www.youtube.com/watch?v=oDvAqXUJhF4
- https://www.indiegogo.com/projects/4-mation-the-interactive-3d-zoetrope#/

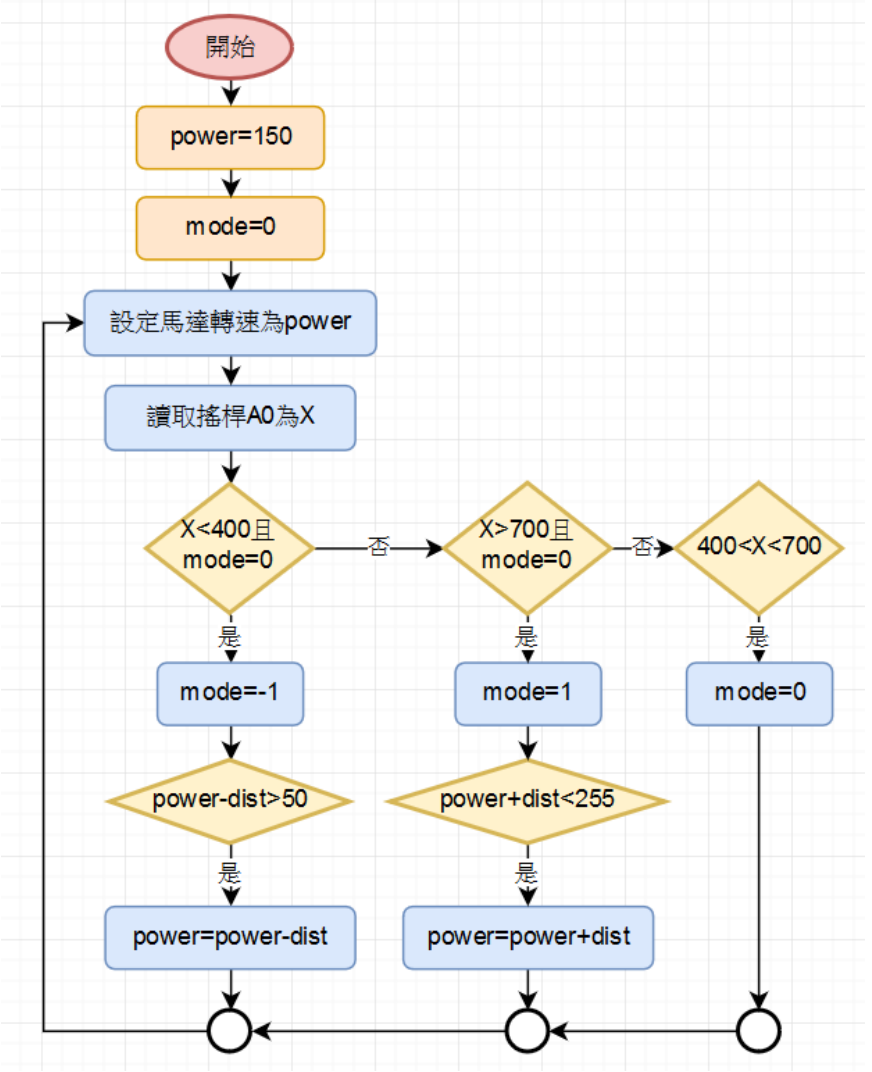
練習三 費納奇鏡(一)

- 情境目的：使用搖桿調整直流減速馬達N20的轉速，讓圓盤上的圖形產生動畫效果。
- 情境分析：
  1. 搖桿往左搖(X<400)降低馬達轉速，往右搖(X>700)提高馬達轉速。
  2. 馬達轉速power範圍：50~255，不可超過這個範圍。
  3. 轉速增加量dist：5
  4. 每搖一次搖桿只會產生一次增減量，必須放開搖桿回到中間，再次搖動後才能再次改變馬達轉速(mode)。

練習三 費納奇鏡(一)

- 演算法步驟與流程圖：

```
01  power初值為150，mode初值為0
02  重覆執行
03        設定馬達轉速為power
04        讀取搖桿A0為X
05        如果X<400且mode=0
06              mode=-1
07              如果 power-dist>50
08                    power=power-dist
09        如果X>700且mode=0
10              mode=1
11              如果 power+dist<255
12                    power=power+dist
13        如果400<X<700
14                    mode=0
15  結束重覆執行
```



練習三 費納奇鏡(二)

- 情境目的：除了用搖桿調整馬達轉速，並讓燈條頻閃模擬影片的幀率(fps)
- 情境分析：
  1. 30fps代表每秒30幀，也就是大約每33毫秒出現一個影格。
  2. 讓燈條全亮1ms，熄滅32ms
  3. 由於切換速度相當快，使用USB連線會跟不上，需將程式燒錄到Arduino Nano中。

練習四 電子音樂

- 情境任務：使用一個蜂鳴器積木播放一首曲子
- 情境分析：
  1. 樂曲以4個欄位存放在檔案「大黃蜂的飛行」中，分別是：
     A. 音符：簡譜音階符號1~7
     B. 八度：1為中央C所在八度，2為高一個8度，-1則為低一個8度
     C. 升降音：+代表升半音，-代表降半音
     D. 節拍：1為1拍，0.5為半拍，0.25為四分之一拍。
  2. 每個音的八度音頻率為倍數關係，例如中央A頻率為440，高一個8度A為880，低一個8度A為220。
  3. 每個音的頻率是前一個半音階的$\sqrt[12]{2}$倍，約等於1.059463
  4. 讀取檔案中的樂曲資料，算出對應的頻率後，存放在清單中，可以降低演奏時的負擔。

練習四 電子音樂

- 演算法步驟：主程式

01 準備一個中央八度1~7的頻率清單，清單名稱為「基頻」，頻率分別是：262,294,330,349,392,440,494
02 建立空白的「演奏頻率」清單與「演奏節拍」清單
03 如果 「演奏頻率」清單長度=0

04 **處理樂曲資料**
05 否則

06 **播放樂曲**

練習四 電子音樂

□ 演算法步驟：副程式**處理樂曲資料**

01 設定列號初值為2
02 設定頻率初值為0

02 重複讀取樂曲資料檔案(音符、八度、升降、節拍)直到音符=9
03 **查出音符基頻(音符)**
04 **計算八度頻率(八度)**
05 **計算升降音頻率(升降)**

06 算好的頻率加入「演奏頻率」清單、節拍直接加入「演奏節拍」清單
07 列號=列號+1

練習四 電子音樂

- 演算法步驟：副程式**計算八度頻率(octave)**

```
01 如果 octave>0
02     重複 octave-1 次
03         頻率=頻率*2
04 否則
05     如果 octave<0
06         重複(octave*-1)次
07             頻率=頻率/2
```

- 演算法步驟：副程式**計算升降音頻率(sf)**

```
01 如果 sf= "+"
02     頻率=四捨五入(頻率*1.059463)
03 否則
04     如果 sf= "-"
05         頻率=四捨五入(頻率/1.059463)
```

練習四 電子音樂

- 演算法步驟：副程式**播放樂曲**

```
01  設定列號初值為1
02  設定速度初值為180
03  設定速度加權為1/(速度/60)
04  重複執行直到列號=演奏頻率清單的長度
05        讀取演奏頻率清單中的頻率、演奏節拍清單中的節拍
06        如果頻率=0    /*休止符*/
07              等待速度加權*節拍
08        否則
09              蜂鳴器播放頻率，時間為速度加權*節拍*1000
```