馬達與感測器教具平台

與

程式設計演算法及流程圖

bit.ly/NKNUBLOCK

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

情境任務VS電腦邏輯

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

- **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

讓LED慢慢變亮

- **情境分析**

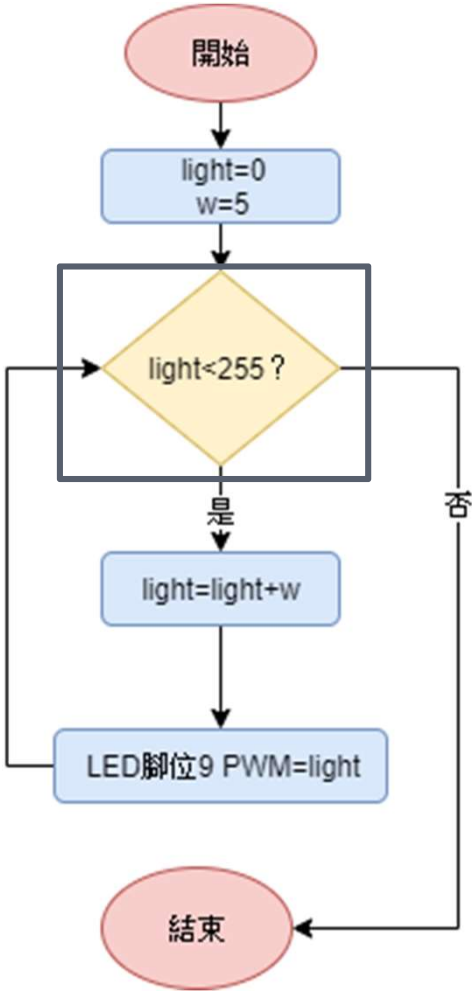
提問	記錄
1. 讓LED變亮就是改變LED的什麼？	LED的亮度(light)，PWM數值
2. 一開始LED亮度是多少？	亮度=0，或light=0
3. LED最亮可以多亮？	亮度=255，或light=255
4. 每次亮度改變多少？	增亮=5, 或w=5
5. 慢慢變亮的意思是每次改變後要...？	等待，0.1秒
6. LED接在哪個腳位？	紅 D9、綠 D10、藍D11

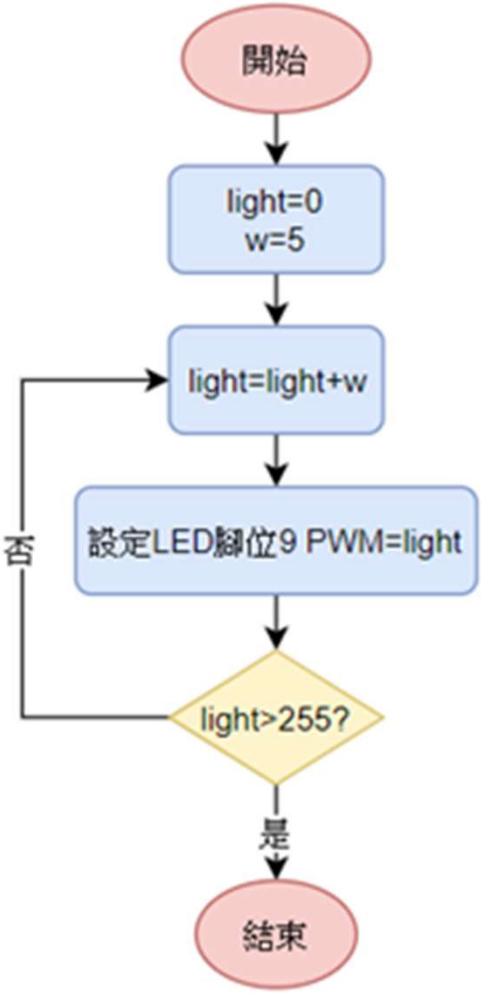
- **演算法步驟**

1. 設定light=0
2. 設定w=5
   重複直到 light > 255
3. light+w
4. 等待0.1秒
5. 設定D9的PWM為light
6. 重複結束

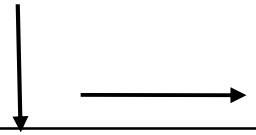
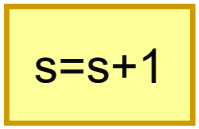
讓LED慢慢變亮程式範例

讓LED亮慢慢變亮程式與流程圖





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	

讓LED慢慢變亮延伸

1. 在程式底下加上讓LED慢慢變暗，並且重複變亮、變暗的過程。
2. 讓紅、綠、藍三色輪流出現，不要讓程式變得太長。

● 讓LED慢慢變暗

```
01 設定light=255
02 設定w=-5
03 重覆直到light＝0
04     light+w
05     等待0.1秒
06     設定D9的PWM為light
07 結束重覆
```

● 讓三色輪流出現

```
01 設定pin=9
02 重覆3次
03     設定light=0
04     設定w=5
05     重覆直到light＝255
06         light+w
07         等待0.1秒
08         設定pin的PWM為light
09     重覆結束
10     pin+1
11 重覆結束
```

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)。

情境分析時的提問

1. 風扇有什麼危險？
2. 這樣的危險跟距離有關，有什麼模組可以告訴我們距離，進而用來控制風扇？
3. 如何用距離控制風扇轉速？
   a. 遠：
   b. 中：
   c. 近：
4. 除了改變風扇轉速外，還可以有其他的警告訊息嗎？可以有怎樣的警告訊息？

安全風扇演算法步驟

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

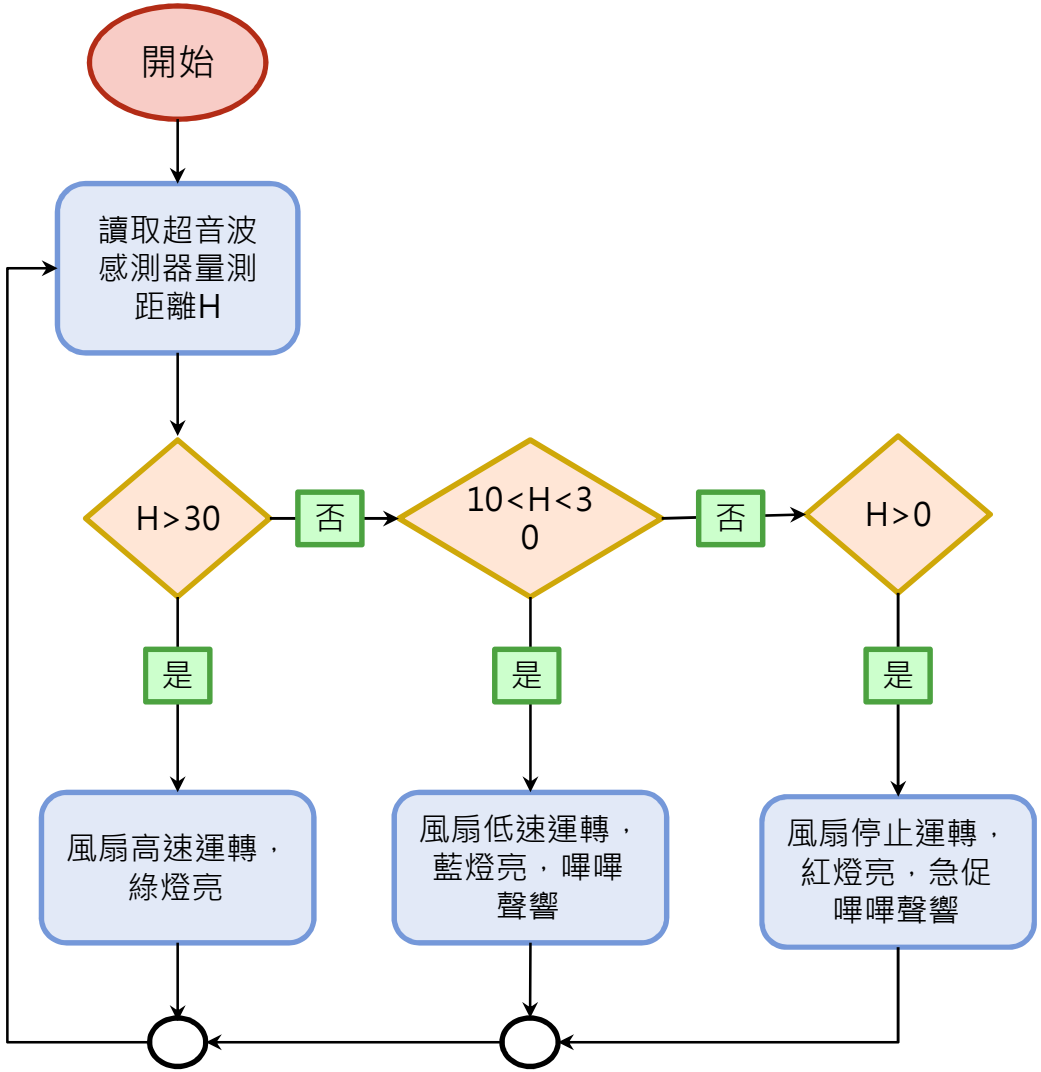
3-2. 否則：

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

```
01 重覆執行
02    讀取超音波感測器量測距離H(超音波:A2、A3)
03    如果 H > 30
04          風扇高速運轉(speed=255)
05          綠燈亮(D10高，D9、D11低)
06     否則
07          如果 10 < H < 30
08                風扇低速運轉(speed=150)
09                藍燈亮(D11高，D9、D10低)
10                發出嗶嗶聲響(Do, 100毫秒)
11           否則
12                如果 H>0
13                      風扇停止(speed=0)
14                      紅燈亮(D9高，D10、D11低)
15                      發出急促嗶嗶聲響(La, 50毫秒)
16 結束重複
```

安全風扇流程圖


開始
讀取超音波感測器量測距離H
H>30
否
10<H<30
否
H>0
是
是
是
風扇高速運轉，綠燈亮
風扇低速運轉，藍燈亮，嗶嗶聲響
風扇停止運轉，紅燈亮，急促嗶嗶聲響

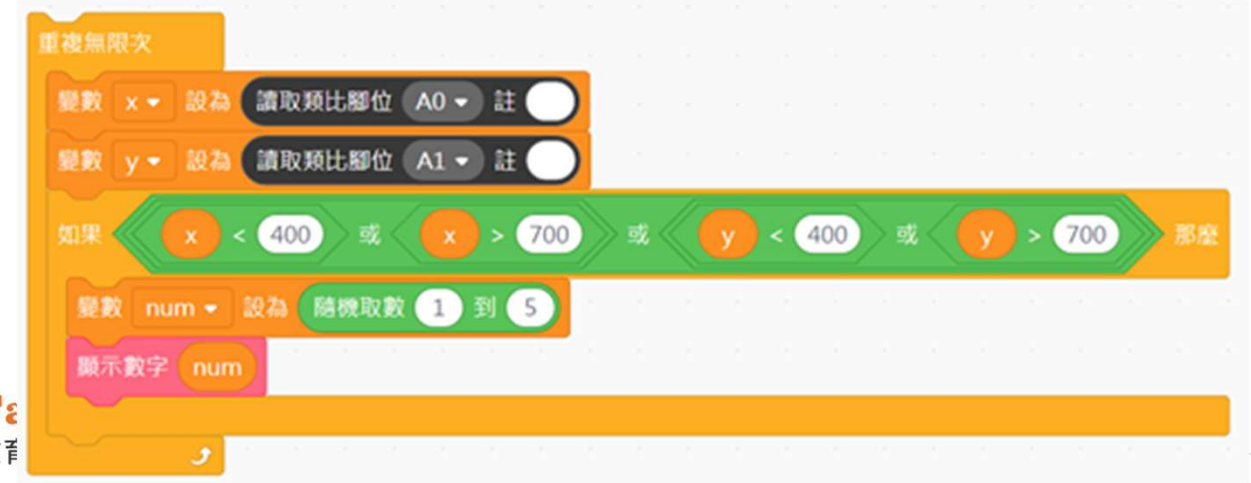
練習二 抽籤機

- 情境主題：抽籤機
- 情境目的：利用平板上的模組製作園遊會抽飛鏢數量的抽籤機。
- 情境分析：
  - 怎樣呈現數字？用哪一個模組？
  - 怎樣抽？用哪一個模組？方法？

練習二 抽籤機類型一

- 演算法步驟與流程圖：
  - 輸出：8X8 LED矩陣，顯示數字1~5
  - 輸入：搖桿，搖桿搖到任一個方向就開始隨機出現數字，放開搖桿後數字固定。
- 演算法與流程圖

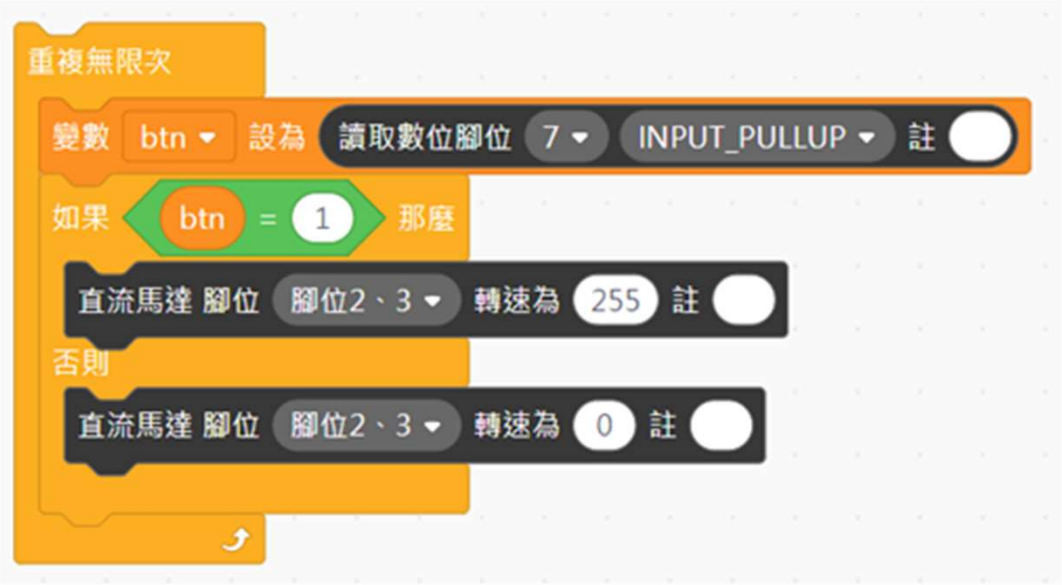
01 重覆執行
02 讀取搖桿A0為X
03 讀取搖桿A1為Y
04 如果X<400 或 X>700 或Y<400或Y>700
05 從1-5隨機選一個數num
06 8x8LED顯示num
07 結束重覆執行



練習二 抽籤機類型二

- 演算法步驟與流程圖：
  - 輸出：馬達，輪盤
  - 輸入：搖桿，按下按鈕後馬達開始轉動，直到放開按鈕後馬達停止，指針所指的數字就是抽籤數字。

```
01 重覆執行
02       讀取搖桿按鈕D7為btn
03       如果btn=1
05             馬達轉動(255)
06       否則
07             馬達停止(0)
08 結束重覆執行
```



抽籤機

這樣的設計會有什麼問題？

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#/

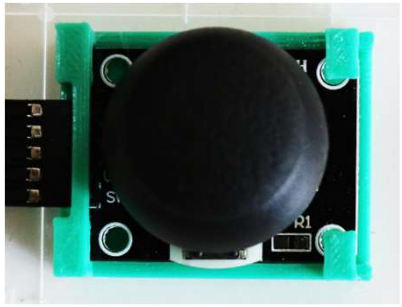
練習三 費納奇鏡

- 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：
  - 搖桿往左搖(X<400)降低馬達轉速，往右搖(X>700)提高馬達轉速。
  - 馬達轉速power範圍：50~255，不可超過這個範圍。
  - 轉速增加量diff：5

練習三 費納奇鏡

- 這樣設計會有什麼問題？如何改善？
  - 限制轉速最低不可超過50，最高不可超過255。
  - 降低轉速改變的速度，例如將增減量由5改為2；或每增減一次就暫停一下(0.2秒)
  - 每搖一次搖桿只能加或減一次轉速，必須放開搖桿回到中間，再次搖動後才能改變馬達轉速。

speed



X<400

X>700

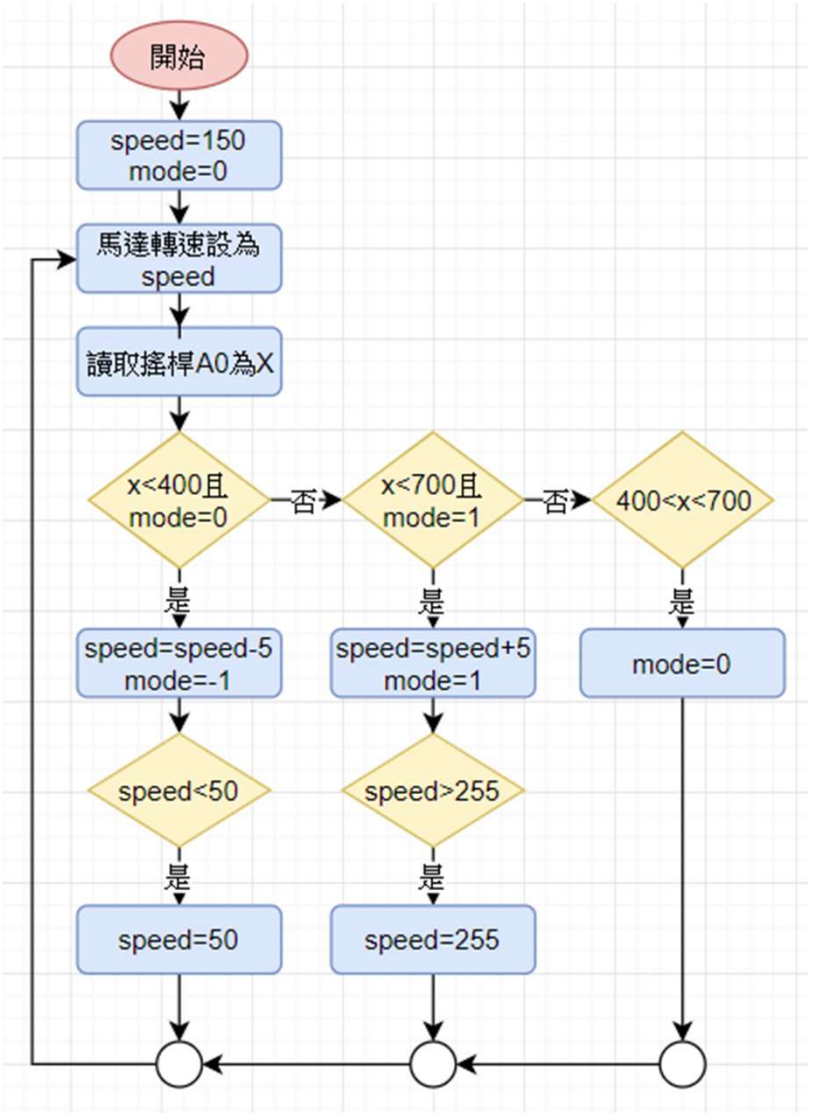
-5

5

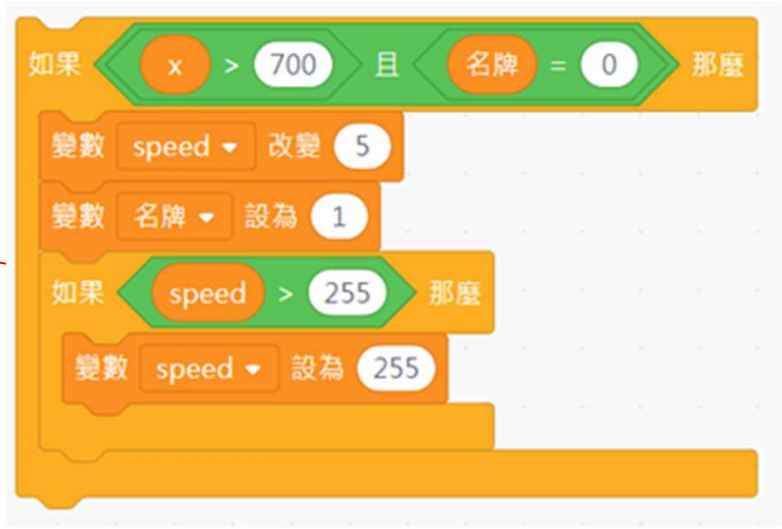
練習三 費納奇鏡(一)

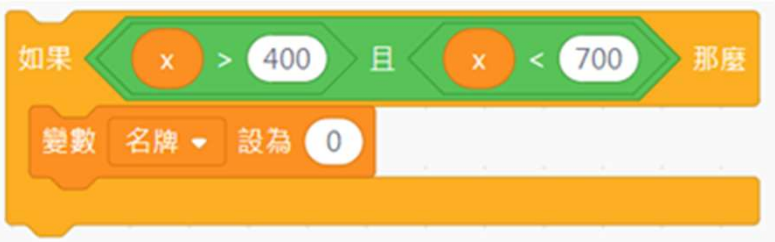
- 演算法步驟與流程圖：

```
01 speed初值為150，mode初值為0
02 重覆執行
03       設定馬達轉速為speed
04       讀取搖桿A0為X
05       如果X<400且mode=0
06            speed=speed-5
07            mode=-1
08            如果 speed<50
09                 speed=50
10       如果X>700且mode=0
11            speed=speed+5
12            mode=1
13            如果 speed>255
14                 speed=255
15       如果400<X<700
16            mode=0
17 結束重覆執行
```



練習三 費納奇鏡(一)程式範例





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：
  - 30fps代表每秒30幀，也就是大約每33毫秒出現一個影格。
  - 讓燈條全亮1ms，熄滅32ms
  - 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。

練習三 費納奇鏡(二)

- 頻閃程式範例

練習四 樂曲演奏

- 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 樂曲演奏(一)—主程式

情境分析：從檔案讀取資料，計算出音符的頻率要花費許多時間，若要即時演奏會拖慢演奏速度，所以先把全部資料從檔案中讀出來，計算好，放入清單中，演奏時直接從清單中取出頻率與時間，就不會拖慢速度了。

演算法步驟：主程式

01 建立空白的「演奏頻率」清單與「演奏節拍」清單
02 如果「演奏頻率」清單長度=0
03 　　**處理樂曲資料**
04 **播放樂曲**

樂曲演奏(一)—主程式


當 被點擊
如果 清單 演奏頻率 的長度 = 0 那麼
處理樂曲資料
播放樂曲
定義 處理樂曲資料
定義 播放樂曲
基頻
1 262
2 294
3 330
4 349
5 392
6 440
7 494
+ 長度 7 =
演奏頻率
(empty)
+ 長度 0 =
演奏節拍
(empty)
+ 長度 0 =

練習四 樂曲演奏(二)--處理樂曲資料

情境分析：樂譜檔中有音符、八度、升降、節拍四個欄位，每一列就是一個音的資料，我們要把這些資料一列一列讀出來，並設計音符資料的處理程序。
註：資料從第二列開始，音符=0時代表休止符，音符=9是檔案結尾。

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
03  設定音符初值為0
04  重複讀取樂曲資料檔案(音符、八度、升降、節拍)直到音符=9
05        如果 音符等於0
06              頻率=0
07        否則
08              查出音符基頻(音符)
09              計算八度頻率(八度)
10              計算升降音頻率(升降)
11        將算好的頻率加入「演奏頻率」清單
12        節拍直接加入「演奏節拍」清單
13        列號=列號+1
```

樂曲演奏副程式(二)：處理樂曲資料

有沒有更好的處理方法？

練習四 樂曲演奏(三)

情境分析：從檔案中讀到的音符是1~7，怎麼知道它的頻率
演算法步驟：副程式查出音符基頻(note)

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 頻率=基頻清單的第note項

有沒有其它方式？

練習四 樂曲演奏(三)

情境分析：
八度=2為高一個8度；八度=1為中央高度，八度=-1為低一個8度
同一個音，高一個8度頻率要*2，低一個8度頻率要/2

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
```

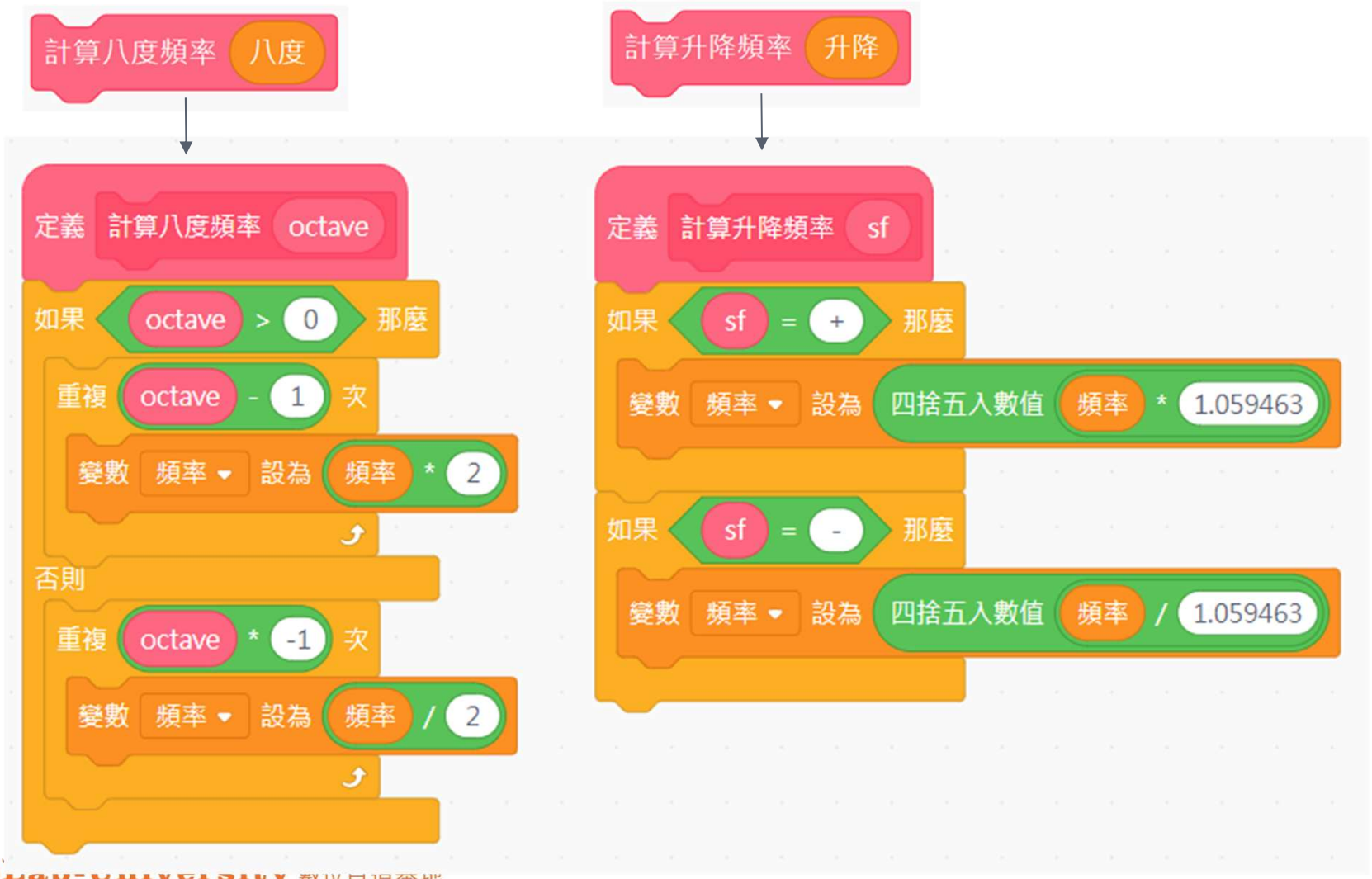
有沒有其他方法？

情境分析：
升降='+'代表升記號，頻率是原音的1.059463倍；
升降='-'代表降記號，頻率是原音的1/1.059463倍

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)
```

樂曲演奏(三)副程式：
查出音符基頻、計算八度頻率、計算升降頻率



樂曲演奏(四)：播放樂曲

情境分析：播放樂曲時需要有頻率與時間，頻率從清單中取出後可以直接使用，但時間則需要將從清單中取出的節拍做進一步運算。
樂曲的速度會標示一個數字，例如180，代表每分鐘180拍，也就是每秒(180/60)拍，每拍的時間就是1/(180/60)秒，把這個數字當作加權，乘以每個音的節拍，就是那個音的演奏時間。
如果頻率=0，代表休止符，就等待休止符所佔的時間，否則就演奏頻率，時間則要轉化為毫秒(*1000)

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
```

樂曲演奏：主程式與播放樂曲副程式