

教具名稱	智慧居家教具平台
課程名稱	微型天氣監測站
運算思維	演算法步驟
編撰教師	蘇建誠
編撰基地	高師大自造者基地
課程影片(有/無)	無

大綱

1. 情境主題及目的
2. 情境分析
3. 演算法步驟(學生填空用)
4. 演算法步驟(教師用)
5. 演算法步驟 vs 積木程式堆疊
6. 延伸問題

情境主題及目的

- 情境主題：微型天氣監測站
- 情境目的：使用感測器蒐集溫度、溼度資訊，除了即時顯示資訊外，必須繪製統計圖並將資料儲存在檔案中。

情境分析一(功能分析)

- 天氣監測需要收集哪些資訊？
 - 氣溫
 - 溼度
 - 空氣品質
 - 紫外線
 -
 - 取得資料時刻：年、月、日、時

情境分析一(功能分析)

- 隔多久時間收集一次資料？
 - 參考中央氣象局：一小時
 - 溫度、溼度不會驟然改變
 - 資料太少無法反映真實狀況，但資料越多，後續處理越麻煩。

情境分析一(功能分析)

- 收集到的資料如何處理？
 - 即時顯示：LCD
 - 統計圖
 - 一天更換一次統計圖
 - 存檔
 - 一個檔案儲存一個月的資料

情境分析二(器材分析)

- 根據前面的討論，列出會用到的感測器與模組並了解這些模組的使用方法、特性與限制
 - 溫溼度感測器：偵測氣溫與溼度
 - LCD：即時顯示溫度與溼度
 - 燈條：資料異常時顯示紅燈警示，否則顯示綠燈

情境分析三(資料分析)

- 根據前面的討論，程式中需要處理哪些資料呢？預設值是多少？
 - 溫度：儲存氣溫傳回值(攝氏)
 - 溼度：儲存溼度傳回值
 - 間隔時間：監測間隔時間，預設3600(1小時，測試時先用小數字，例如10秒，等系統穩定後再改回3600)
 - 列號：資料存在檔案的第幾列，預設為1
 - 日期：當實際日期變更時，刪除舊圖表，更新此數值，並以此數值做為圖表名稱，建立新的統計圖。
 - 月份：當實際月份改變時，更新此數值，並以此數值做為資料檔名稱。
 - 圖表名稱：包含圖表名稱_溫度與圖表名稱_溼度
 - 檔名：儲存資料檔路徑與名稱

情境分析四(流程分析)

1. 先處理圖表與檔案

— 啟動與換日

- 刪掉舊圖表(溫度與溼度各一個圖表)
- 以月、日設定圖表名稱(溫度與溼度各一個名稱)
- 建立新圖表(溫度與溼度各一個圖表)

— 啟動與換月

- 以年、月設定檔名

2. 處理資料(每隔一小時)

- 以計時器取代等待指令，讓系統在間隔時間中仍可進行其他任務，提高系統的擴充性。
- 讀取氣溫與溼度
- 如果資料正常就亮綠燈、否則亮紅燈
- LCD顯示資料
- 將氣溫與溼度加入圖表
- 將年、月、日、氣溫、溼度寫入檔案

演算法步驟(學生填空用)

- 01 預設 間隔時間 = , 當前日期 = 0 , 當前月份 = 0 , 列號 =
- 02 重新計時
- 03 重複執行
- 04 如果 目前日期 日期
- 05 刪除舊圖表並建立新圖表
- 06 如果 目前月份 月份
- 07 設定檔名
- 08 如果 計時器 間隔時間
- 09 重新計時
- 10 讀取 與
- 11 如果 溫度 = NaN /* 讀取失敗 */
- 12
- 13 否則
- 14
- 15 LCD顯示資料
- 16 繪製圖表 /*加入資料項目*/
- 17 儲存資料 /*將資料寫入檔案*/
- 18 結束重複執行

演算法步驟(教師用)

- 01 預設 間隔時間=10 · 當前日期=0 · 當前月份=0 · 列號=1
- 02 重新計時
- 03 重複執行
- 04 如果 目前日期 不等於 日期
- 05 刪除舊圖表並建立新圖表
- 06 如果 目前月份 不等於 月份
- 07 設定檔名
- 08 如果 計時器 > 間隔時間
- 09 重新計時
- 10 讀取溫度與溼度
- 11 如果 溫度 = NaN /* 讀取失敗 */
- 12 亮紅燈
- 13 否則
- 14 亮綠燈
- 15 LCD顯示資料
- 16 繪製圖表 /*加入資料項目*/
- 17 儲存資料 /*將資料寫入檔案*/
- 18 結束重複執行

演算法步驟vs程式堆疊

```
01 預設 間隔時間=10， 當前日期=0， 當前月份=0， 列號=1
02 重新計時
03 重複執行
04     如果 目前日期 不等於 日期
05         刪除舊圖表並建立新圖表
06     如果 目前月份 不等於 月份
07         設定檔名
08     如果 計時器 > 間隔時間
09         重新計時
10         讀取溫度與溼度
11         如果 溫度 = NaN /* 讀取失敗 */
12             亮紅燈
13         否則
14             亮綠燈
15             LCD顯示資料
16             繪製圖表 /*加入資料項目*/
17             儲存資料 /*將資料寫入檔案*/
18 結束重複執行
```



演算法步驟vs程式堆疊

- 01 預設 間隔時間=10，當前日期=0，當前月份=0，列號=1
- 02 重新計時
- 03 重複執行
- 04 如果 目前日期 不等於 日期
- 05 刪除舊圖表並建立新圖表
- 06 如果 目前月份 不等於 月份
- 07 設定檔名
- 08 如果 計時器 > 間隔時間
- 09 重新計時
- 10 讀取溫度與溼度
- 11 如果 溫度 = NaN /* 讀取失敗 */
- 12 亮紅燈
- 13 否則
- 14 亮綠燈
- 15 LCD顯示資料
- 16 繪製圖表 /*加入資料項目*/
- 17 儲存資料 /*將資料寫入檔案*/
- 18 結束重複執行

The image shows a Scratch script with several blocks highlighted in different colors to match the algorithm steps:

- Yellow highlight:** An 'if' block with the condition '目前時間的 日 = 日期' and a '那麼' block containing '建立新圖表'.
- Red highlight:** A 'define' block for '建立新圖表' containing:
 - Two 'set variable to value' blocks: '圖表名稱_溫度' set to '字串組合 目前時間的 月 - 日期 氣溫折線圖' and '圖表名稱_溼度' set to '字串組合 目前時間的 月 - 日期 溼度折線圖'.
 - Two 'delete chart' blocks: '刪除圖表 圖表名稱_溫度' and '刪除圖表 圖表名稱_溼度'.
- Green highlight:** A 'set variable to value' block for '日期' set to '目前時間的 日'.
- Blue highlight:** Two 'set variable to value' blocks: '圖表名稱_溫度' set to '字串組合 目前時間的 月 - 日期 氣溫折線圖' and '圖表名稱_溼度' set to '字串組合 目前時間的 月 - 日期 溼度折線圖'.
- Red highlight:** Two 'build chart' blocks. The first is for '溫度' with 'Y軸名字' as '氣溫 單位 °C' and 'X軸名稱' as '時 單位'. The second is for '溼度' with 'Y軸名字' as '溼度 單位 %RH' and 'X軸名稱' as '時 單位'. Both have '模式' set to '自訂'.

1. 測試時可以用時間模式，待程式穩定後再改回自訂模式。
2. 在時間模式中，每加入三筆資料才會出現一個資料點。

演算法步驟vs程式堆疊

- 01 預設 間隔時間=10 · 當前日期=0 · 當前月份=0 · 列號=1
- 02 重新計時
- 03 重複執行
- 04 如果 目前日期 不等於 日期
- 05 刪除舊圖表並建立新圖表
- 06 如果 目前月份 不等於 月份
- 07 設定檔名
- 08 如果 計時器 > 間隔時間
- 09 重新計時
- 10 讀取溫度與溼度
- 11 如果 溫度 = NaN /* 讀取失敗 */
- 12 亮紅燈
- 13 否則
- 14 亮綠燈
- 15 LCD顯示資料
- 16 繪製圖表 /*加入資料項目*/
- 17 儲存資料 /*將資料寫入檔案*/
- 18 結束重複執行

The image shows a Scratch script with the following blocks:

- Control:** "如果" (If) block with a green arrow pointing right. Inside: "目前時間的 月" (Current month) = "月份" (Month). The arrow is labeled "不成立" (Not true) and "那麼" (Then).
- Block:** "設定檔名" (Set filename).
- Control:** "定義" (Define) block with a pink arrow pointing right. Inside: "設定檔名" (Set filename).
- Block:** "變數 列號" (Variable: Column number) set to "1".
- Block:** "變數 檔名" (Variable: Filename) set to "字串組合 d:\ 目前時間的 年 - 目前時間的 月" (String combination: d:\ Current year - Current month).
- Block:** "變數 月份" (Variable: Month) set to "目前時間的 月" (Current month).

將列號復原為1，讓資料從第一列開始寫入

以新的月份為檔名，並更新變數月份的數值

演算法步驟vs程式堆疊

- 01 預設 間隔時間=10，當前日期=0，當前月份=0，列號=1
- 02 重新計時
- 03 重複執行
- 04 如果 目前日期 不等於 日期
- 05 刪除舊圖表並建立新圖表
- 06 如果 目前月份 不等於 月份
- 07 設定檔名
- 08 如果 計時器 > 間隔時間
- 09 重新計時
- 10 讀取溫度與溼度
- 11 如果 溫度 = NaN /* 讀取失敗 */
- 12 亮紅燈
- 13 否則
- 14 亮綠燈
- 15 LCD顯示資料
- 16 繪製圖表 /*加入資料項目*/
- 17 儲存資料 /*將資料寫入檔案*/
- 18 結束重複執行

計時器數值要先四捨五入為整數

```
如果 四捨五入數值 計時器 > 間隔時間 那麼
  計時器重置
  變數 溫度 設為 溫濕度感應器 腳位 2 的 溫度 °C
  變數 溼度 設為 溫濕度感應器 腳位 2 的 溼度
  如果 溫度 = NaN 那麼
    亮紅燈
  否則
    亮綠燈
  定義 亮紅燈
    設WS2812燈條 3 顆中的第 全部 (-1) 顆,紅 255 綠 0 藍 0 腳位 4
  定義 亮綠燈
    設WS2812燈條 3 顆中的第 全部 (-1) 顆,紅 0 綠 255 藍 0 腳位 4
```

演算法步驟vs程式堆疊

- 01 預設 間隔時間=10，當前日期=0，當前月份=0，列號=1
- 02 重新計時
- 03 重複執行
- 04 如果 目前日期 不等於 日期
- 05 刪除舊圖表並建立新圖表
- 06 如果 目前月份 不等於 月份
- 07 設定檔名
- 08 如果 計時器 > 間隔時間
- 09 重新計時
- 10 讀取溫度與溼度
- 11 如果 溫度 = NaN /* 讀取失敗 */
- 12 亮紅燈
- 13 否則
- 14 亮綠燈
- 15 LCD顯示資料
- 16 繪製圖表 /*加入資料項目*/
- 17 儲存資料 /*將資料寫入檔案*/
- 18 結束重複執行

The image shows a Scratch code stack for an IoT project. It is organized into three sections:

- LCD顯示 (LCD Display):** Contains four blocks: '定義 LCD顯示' (Define LCD Display), 'LCD 清除畫面' (LCD Clear Screen) with I2C address 0x3F, 'LCD 開啟背光' (LCD Turn On Backlight) with I2C address 0x3F, 'LCD 第 1 列 第 1 格顯示' (LCD Row 1 Column 1 Display) showing 'Temp: 溫度' (Temperature) at I2C address 0x3F, and 'LCD 第 2 列 第 1 格顯示' (LCD Row 2 Column 1 Display) showing 'Humidity: 溼度' (Humidity) at I2C address 0x3F.
- 繪製圖表 (Draw Chart):** Contains two blocks: '定義 繪製圖表' (Define Draw Chart) and two '插入數值到, 圖表' (Insert Value to Chart) blocks. The first chart is for '溫度' (Temperature) and the second is for '溼度' (Humidity), both using '目前時間的 時' (Current Time Hour) as the X-axis value.
- 儲存資料 (Store Data):** Contains seven blocks: '定義 儲存資料' (Define Store Data), and six '寫入' (Write) blocks for '年' (Year), '月' (Month), '日' (Day), '時' (Hour), '溫度' (Temperature), and '溼度' (Humidity) at rows 1 through 6 of the specified column. It ends with '儲存檔案 檔名' (Save File Name) and '變數 列號 改變 1' (Change Variable Column Number by 1).

演算法步驟vs程式堆疊—完整主程式

- 01 預設 間隔時間=10，當前日期=0，當前月份=0，列號=1
- 02 重新計時
- 03 重複執行
- 04 如果 目前日期 不等於 日期
- 05 刪除舊圖表並建立新圖表
- 06 如果 目前月份 不等於 月份
- 07 設定檔名
- 08 如果 計時器 > 間隔時間
- 09 重新計時
- 10 讀取溫度與溼度
- 11 如果 溫度 = NaN /* 讀取失敗 */
- 12 亮紅燈
- 13 否則
- 14 亮綠燈
- 15 LCD顯示資料
- 16 繪製圖表 /*加入資料項目*/
- 17 儲存資料 /*將資料寫入檔案*/
- 18 結束重複執行



延伸討論

- 檔案缺少欄位名稱，如果要用程式幫檔案加入欄位名稱，該寫在什麼地方？怎麼寫？

	A	B	C	D	E	F
1	2019	11	18	22	27	66
2	2019	11	18	22	27	66
3	2019	11	18	22	27	66
4	2019	11	18	22	26	68
5	2019	11	18	22	26	68

	A	B	C	D	E	F
1	年	月	日	時	氣溫	溼度
2	2019	11	18	22	27	66
3	2019	11	18	22	27	66
4	2019	11	18	22	27	66
5	2019	11	18	22	26	68

- 在範例程式中，偵測資料異常時只用燈號警示，然後還是將異常資料加入圖表與檔案中，有沒有更好的處理方式？怎麼做？