



馬達與感測器平台 教學成果分享

報告人：台北市立明德
國中陳好蓁老師

實際操作進度

• 第一節

- 介紹NKNUBLOCK
- 連接電腦

• 第二節

- 控制RGB LED燈
- 彩虹燈

• 第三節

- 認識8*8點矩陣
- 設計動畫效果
- 鍵盤控制燈號

• 第四節

- 認識蜂鳴器
- 倒車雷達

• 第五節

- 認識超音波感測器
- 超音波RGB燈

• 第六節

- 認識搖桿
- 偵測並顯示搖桿座標
- 搖桿控制角色移動

• 第七節

- 認識直流馬達N20
- 控制轉動方向和轉速
- 搖桿控制馬達轉動

• 第八節

- 認識伺服馬達SG90
- 馬達歸零和轉動角度
- 超音波感應柵欄

• 第九節

- 台北市立明德國中校園進出入系統

• 第十節


- 遊樂園探險之遊戲
(電腦課之實作內容)
- 將滑鼠改為搖桿

• 第十一節

- 認識直流馬達
- Led燈串聯與並聯
(Thinkcad Circuits)

• 第十二節

- 炫彩涼風扇
- 專案分享



各元件教學

第一節

1.介紹NKNUBLOCK

2.連接電腦

- 先觀賞網路影片了解什麼是Arduino

(影片：Arduino 到底是什麼？)



第一節

1.介紹NKNUBLOCK

2.連接電腦

- 到 [Arduino 官網](#) 點擊 Nano 板，讓學生找教具上的 Nano 板
- 說明 NKNUBLOCK 是讓 Scratch 和 Arduino 進行溝通的中介軟體



第一節

1.介紹NKNUBLOCK

2.連接電腦

- 介紹4060電控板，腳位分佈

馬達與感測器元件簡介：

- 讓學生看著教具上的元件名稱，簡單說明元件功能即可

(1.2)腳位介紹

數位腳：2-13

輸入訊號：0與1

輸出訊號：0與1

PWM輸出：0-255

PWM腳位：3、5、6、9、10、11

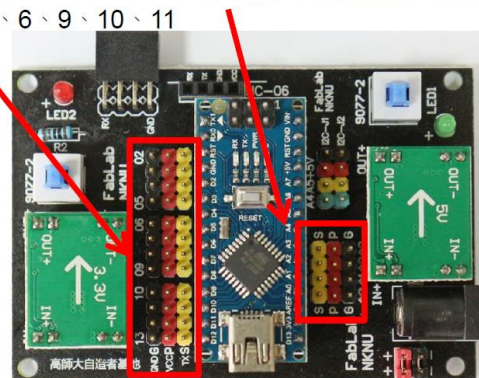
類比腳：A0-A3

輸入訊號：0-1023

輸出訊號：0與1

數位腳不夠用時也可當做數位腳來用

黃色為訊號腳位，
紅色為正極(5V)，
黑色為負極



FabLab-University 數位自造基地
教育部 STEM+A課程導向數位自造教育扎根計畫

Since 2019

第一節

1.介紹NKNUBLOCK

2.連接電腦

• USB線連接

- Nano上的power燈亮

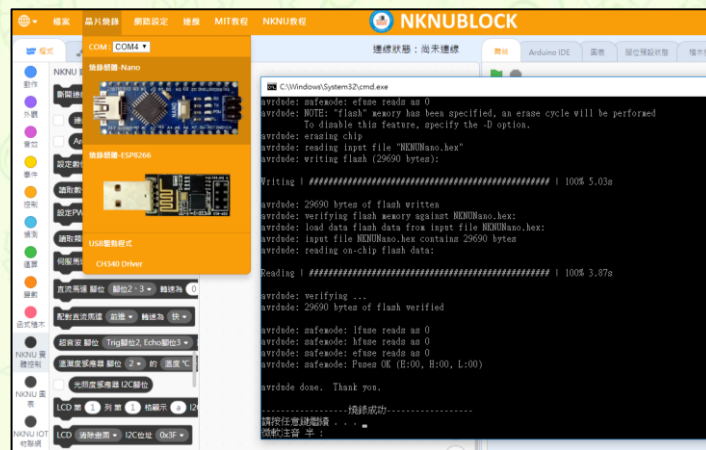
• 晶片燒錄

- 晶片燒錄→選擇COM PORT→燒錄韌體-Nano
- 燒錄成功要將執行畫面關閉（學生會忘記）
- 只有第一次連線，或軟硬體更新才需要晶片燒錄

• 連線

- 連線模式USB→選擇COM PORT→連線

• 以NKNUBLOCK寫一個Scratch小程式



第二節

1.控制RGB LED燈

2.彩虹燈

• 教師講解：

- 生活應用：紅綠燈...

• 教師示範：

- 點亮紅、綠、藍RGB單色燈

• 學生操作：

- 測試腳位9、10、11高低電位結果

• 教師佈題：

- 紅、綠、藍燈，每種顏色各亮一秒鐘，並一直循環。



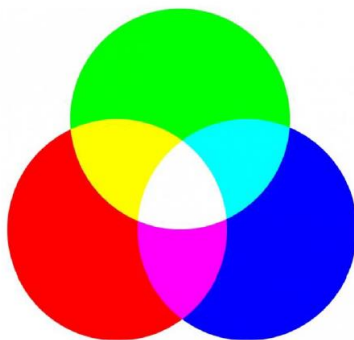
第三節

1.控制RGB LED燈

2.8*8點矩陣

- 教師講解：
 - 同學發現光可以混色
 - 說明光的三原色，並可進行調色
- 教師示範：
 - 數位腳位、類比腳位、PWM腳位
 - 數位訊號 (0或1)：電燈開關on和off
 - 類比訊號 (0-1023)：音量旋鈕
 - 數位轉類比訊號 (0-255)
 - 以PWM腳位進行調色
- 學生操作：
 - 按圖表輸入PWM腳位值，測試結果
- 教師佈題：
 - 完成七色循環彩虹燈。

(3.7)RGB LED模組—調出彩虹的顏色



	R(Red)	G(Green)	B(Blue)
1紅	240	0	0
2橙	160	80	0
3黃	120	120	0
4綠	0	240	0
5藍	0	0	240
6靛(青)	0	120	120
7紫	120	0	120
8白	80	80	80

第四節

1. 認識蜂鳴器
2. 倒車雷達

• 教師講解：

- 蜂鳴器無振盪源
- 不同振盪頻率發出不同音高

• 教師示範：

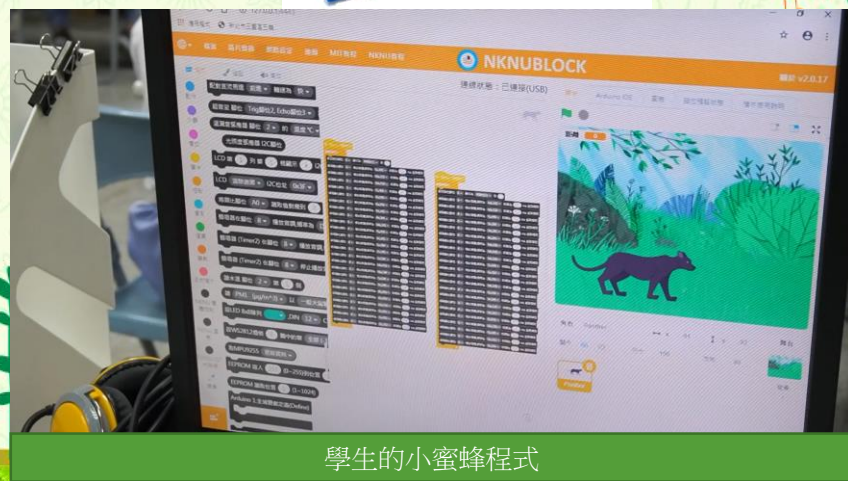
- 程式開頭和結尾，腳位8設為高電位
- 演奏小蜜蜂

• 學生操作：

- 編寫小蜜蜂程式

• 教師佈題1：

- 倒車雷達（超音波+蜂鳴器）：愈靠近障礙物，聲音愈急促。



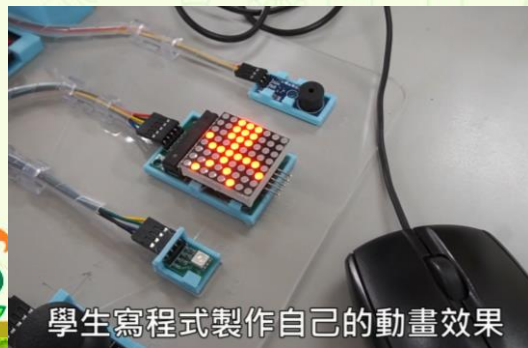
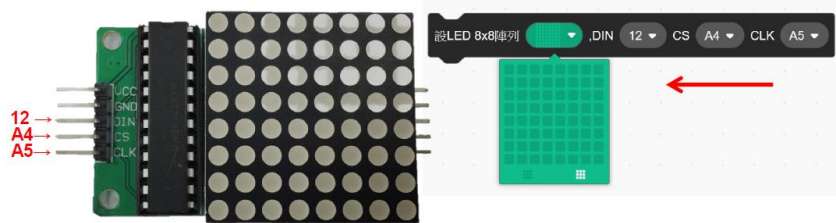
學生的小蜜蜂程式

第五節

1.設計動畫效果 3.鍵盤控制燈號

- 教師講解：
 - 生活應用：小綠人號誌、電子看板
- 教師示範：
 - 動畫設計方式
- 學生操作：
 - 設計動畫效果
 - 鍵盤控制燈號：按下鍵盤方向鍵，可控制8*8點矩陣圖案。
- 教師佈題：
 - 便利商店迎賓燈號：超音波感應客人靠近，8*8點矩陣顯示燈號訊息。

- DIN(訊號)腳位12、CS(晶片選擇)腳位A4、CLK(時脈)腳位A5。



第六節

1. 認識搖桿

2. 偵測並顯示搖桿座標

3. 搖桿控制角色移動

教師講解：

- 生活應用：電玩搖桿、遙控車搖桿
- 搖桿位置要在板子的右下角
- VRX、VRY → A0、A1 (0-1023)
- SW → 腳位7 (0或1)
- 搖桿座標：國小數學還沒上負數、平面直角座標

教師示範：

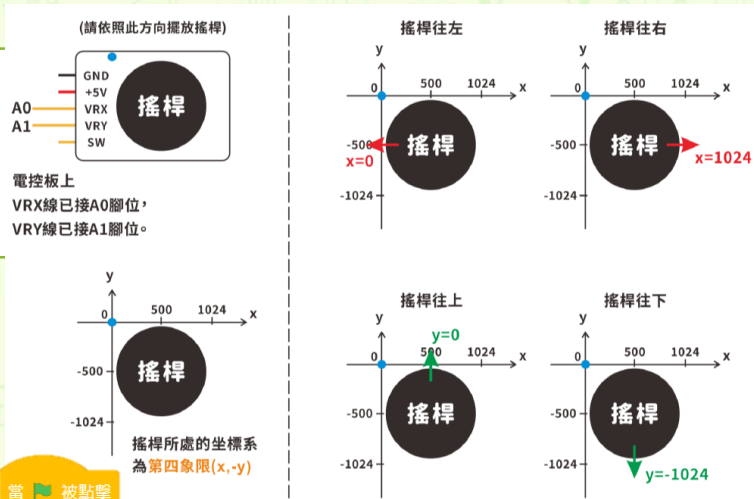
- 變數顯示搖桿座標
- 寫程式要避開誤差值

學生操作：

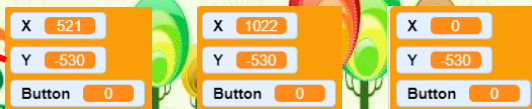
- 偵測並以變數顯示搖桿座標

教師佈題：

- 搖桿控制角色移動：以搖桿控制角色移動，可加上按鈕功能。



```
當 被點擊  
變數 X 設為 0  
變數 Y 設為 0  
變數 Button 設為 0  
重複無限次  
變數 X 設為 讀取類比腳位 A0 註 搖桿X  
變數 Y 設為 讀取類比腳位 A1 註 搖桿Y * -1  
變數 Button 設為 讀取數位腳位 7 INPUT_PULLUP 註 按鈕
```



有時搖桿座標無法達到理想值

大綱

1. 情境主題及目的
2. 情境分析
3. 副程式「紅燈狀態」設計
4. 副程式「綠燈狀態」設計
5. 副程式「小紅人行走」設計
6. 本系統演算法步驟及積木程式堆疊

情境主題及目的

(1) 情境主題：紅綠燈系統

(2) 情境目的：結合搖桿、無源蜂鳴器、RGB LED模組、8*8點矩陣、伺服馬達的運轉，利用搖桿按鈕，控制紅綠燈狀況。

1. 程式開始執行時，『紅燈狀態』用RGB LED 顯示紅燈，柵欄放下，用8x8LED點矩陣 模擬小紅人立正站立。
2. 當搖桿的按鈕壓下，紅燈變綠燈狀態，RGB LED 變綠燈，開啟柵欄，同時用蜂鳴器播放鳥鳴聲或音樂。
3. 小紅人開始行走持續30秒，倒數15秒，小紅人隨倒數時間越少行走速度越快。
4. 倒數計時結束，變紅燈狀態，小紅人變回立正站立，音樂停止，柵欄放下。
5. 程式等待下一個搖桿按鈕壓下的指令

情境分析

(3) 情境分析：

1. 程式開始執行時，設定『紅燈狀態』：
 - 用RGB LED 顯示紅燈：數位腳位 9 設為「高」電位、數位腳位 10 設為「低」電位、數位腳位 11 設為「低」電位。
 - 放下柵欄：伺服馬達 腳位 6 角度 0 度
 - 設8x8LED點矩陣模擬小紅人立正站立
2. 當搖桿的按扭壓下，紅燈變綠燈狀態
 - RGB LED 變綠燈：數位腳位 9 設為「低」電位、數位腳位 10 設為「高」電位、數位腳位 11 設為「低」電位。
 - 開啟柵欄：伺服馬達 腳位 6 角度 90 度
 - 用蜂鳴器播放鳥鳴聲或音樂：數位腳位 8 設為「高」電位，腳位 8 播放音調
 - 同時，小紅人開始行走持續30秒：設 8x8LED陣列 模擬小紅人行走30秒
 - 小紅人行走計時倒數15秒，小紅人隨倒數時間越少行走速度越快。
 - 倒數計時結束，變回紅燈狀態
3. 程式等待下一個搖桿按鈕壓下的指令

演算法步驟

(4) 副程式「紅燈狀態」演算法步驟：

01 用**RGB LED** 亮紅燈：
數位腳位 **9** 設為「**高**」電位
數位腳位 **10** 設為「**低**」電位
數位腳位 **11** 設為「**低**」電位。

02 放下柵欄：
伺服馬達 腳位 **6** 角度 **0** 度
等待 **1** 秒

03 **8x8 LED陣列**：
模擬小紅人立正站立

演算法步驟 vs 積木程式堆疊

(4-1) 副程式「亮紅燈」演算法步驟：

01 用**RGB LED** 亮紅燈：
數位腳位 **9** 設為「**高**」電位
數位腳位 **10** 設為「**低**」電位
數位腳位 **11** 設為「**低**」電位。

定義 亮紅燈

設定數位腳位 9 ▾ 輸出為 高電位(1) ▾ 註 紅

設定數位腳位 10 ▾ 輸出為 低電位(0) ▾ 註 綠

設定數位腳位 11 ▾ 輸出為 低電位(0) ▾ 註 藍

演算法步驟 vs 積木程式堆疊

(4-2) 副程式「放下柵欄」演算法步驟：


02 放下柵欄：
伺服馬達 腳位 6 角度 0 度
等待 1 秒

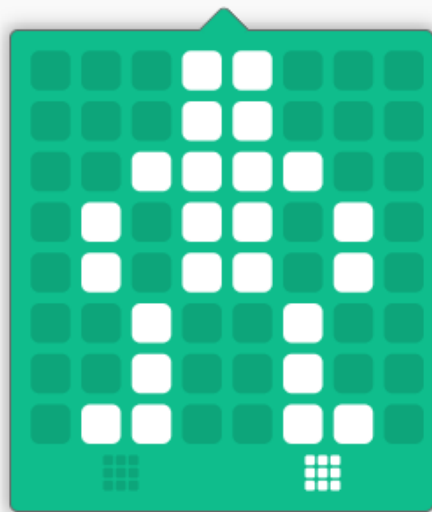
The image shows a Scratch code editor with three blocks stacked vertically:

- A pink block labeled "定義 放下柵欄" (Define Lower the barrier).
- A black block for "伺服馬達" (Motor) with "腳位" (Port) set to 6, "角度為" (Angle) set to 0, and the "註" (Note) checkbox unchecked.
- An orange block for "等待" (Wait) set to 1 second.

演算法步驟 vs 積木程式堆疊

03 **8x8 LED陣列：**
模擬小紅人立正站立

設MAX7219 8*8LED矩陣  ,DIN 12 ▾ CS A4 ▾ CLK A5 ▾



演算法步驟 vs 積木程式堆疊

(4-3) 副程式「紅燈狀態」演算法步驟：

01	副程式「亮紅燈」
02	副程式「放下柵欄」
03	模擬小紅人立正站立

定義 紅燈狀態

亮紅燈

放下柵欄

設MAX7219 8*8LED矩陣



,DIN

12

CS

A4

CLK

A5

演算法步驟

(5) 副程式「綠燈狀態」演算法步驟：

01	<p>用RGB LED亮綠燈：</p> <p>數位腳位 9 設為「低」電位 數位腳位 10 設為「高」電位 數位腳位 11 設為「低」電位。</p>
02	<p>打開柵欄：</p> <p>伺服馬達 腳位 6 角度 90 度 等待 1 秒</p>
03	<p>蜂鳴器播放音調或音樂：</p> <p>數位腳位 8 設為「高」電位， 蜂鳴器在腳位 8 播放音調.....直到播完 數位腳位 8 設為「高」電位</p>

演算法步驟 vs 積木程式堆疊

(5-1) 副程式「亮綠燈」演算法步驟：

01 用RGB LED 亮綠燈：

數位腳位 9 設為「低」電位

數位腳位 10 設為「高」電位

數位腳位 11 設為「低」電位。

定義 亮綠燈

設定數位腳位 9 ▾ 輸出為 低電位(0) ▾ 註 紅

設定數位腳位 10 ▾ 輸出為 高電位(1) ▾ 註 綠

設定數位腳位 11 ▾ 輸出為 低電位(0) ▾ 註 藍

演算法步驟 vs 積木程式堆

疊2) 副程式「打開柵欄」演算法步驟：

02 打開柵欄：
伺服馬達 腳位 6 角度 90 度
等待 1 秒

定義 打開柵欄

伺服馬達 腳位 6 角度為 90 註

等待 1 秒

演算法步驟 vs 積木程式堆

疊3) 副程式「播放音樂」演算法步驟：

03 蜂鳴器播放音調或音樂：
數位腳位 8 設為「高」電位，
蜂鳴器在腳位 8 播放音調.....直到播完
數位腳位 8 設為「高」電位

定義 播放音樂

設定數位腳位 8 ▾ 輸出為 高電位(1) ▾ 註

蜂鳴器在腳位 8 ▾ 播放音調,頻率為 Do,262 ▾ 時間為 500 ms 直到播完

設定數位腳位 8 ▾ 輸出為 高電位(1) ▾ 註

演算法步驟 vs 積木程式堆

疊4) 副程式「綠燈狀態」演算法步驟：

01	副程式「亮綠燈」
02	副程式「打開柵欄」
03	副程式「播放音樂」

定義 綠燈狀態

亮綠燈

打開柵欄

播放音樂



演算法步驟

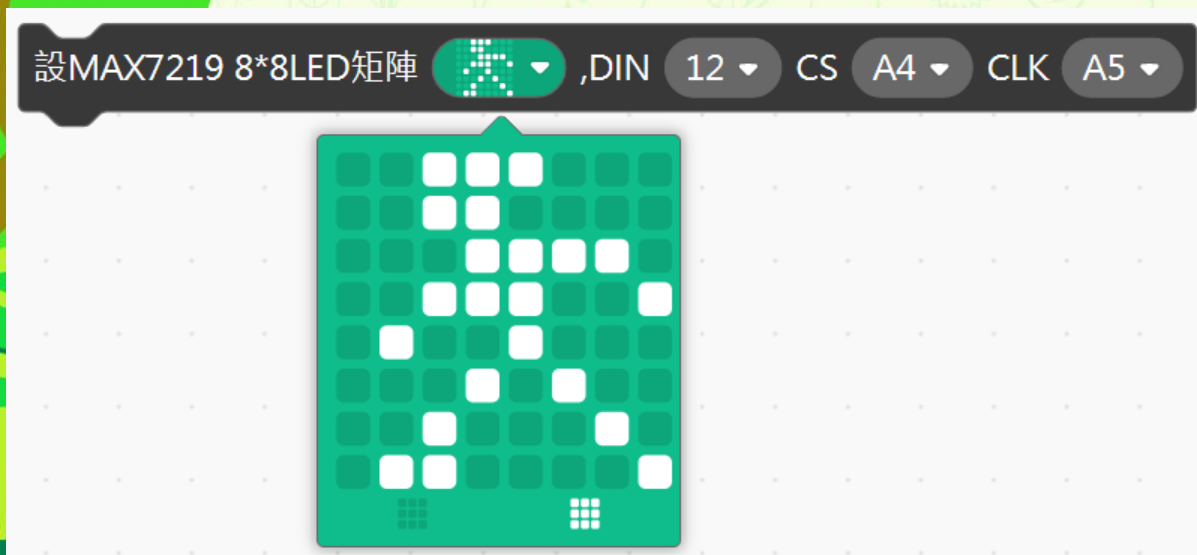
(6) 副程式「小紅人行走」演算法步驟：

模擬「小紅人行走 30 秒，倒數15秒，越走越快」

01	重複迴圈5次
02	模擬小紅人立正站立，等待0.75秒
03	模擬小紅人往前走，等待0.75秒
04	重複迴圈5次
05	模擬小紅人立正站立，等待0.5秒
06	模擬小紅人往前走，等待0.5秒
07	重複迴圈5次
08	模擬小紅人立正站立，等待0.25秒
09	模擬小紅人往前走，等待0.25秒

演算法步驟 vs 積木程式堆

疊1) 模擬小紅人往前走：



演算法步驟 vs 積木程式堆

疊2) 副程式「小紅人行走」演算法步驟1-3：

01	重複迴圈5次
02	模擬小紅人立正站立，等待0.75秒
03	🎵 模擬小紅人往前走，等待0.75秒

重復 10 次

設MAX7219 8*8LED矩陣 ,DIN 12 CS A4 CLK A5


等待 0.75 秒

設MAX7219 8*8LED矩陣 ,DIN 12 CS A4 CLK A5

等待 0.75 秒

演算法步驟 vs 積木程式堆

疊3) 副程式「小紅人行走」演算法步驟4-6：


04	重複迴圈10次
05	模擬小紅人立正站立，等待0.5秒
06	 模擬小紅人往前走，等待0.5秒



The image shows a Scratch-style code block structure for a program. It consists of a 'Repeat' block (orange) with '10' and '次' (times) set. Inside the repeat block, there are two identical sub-blocks. Each sub-block contains a 'Set MAX7219 8*8LED matrix' block (dark grey) with a red person icon selected, and pins set to DIN: 12, CS: A4, CLK: A5. Below each 'Set' block is a 'Wait' block (white) with '0.5' and '秒' (seconds) set. The entire repeat block has a small arrow icon at the bottom right.

演算法步驟 vs 積木程式堆疊

(6-4) 副程式「小紅人行走」演算法步驟7-9：

07	重複迴圈10次
08	模擬小紅人立正站立，等待0.25秒
09	 模擬小紅人往前走，等待0.25秒



The image shows a Scratch-style code editor with the following blocks:

- A "Repeat" block (orange) with "10" in a white circle and "次" (times) to its right.
- A "Set MAX7219 8*8LED matrix" block (dark grey) with a robot icon dropdown, and "DIN 12", "CS A4", and "CLK A5" dropdowns.
- A "Wait" block (orange) with "0.25" in a white circle and "秒" (seconds) to its right.
- Another "Set MAX7219 8*8LED matrix" block (dark grey) with the same robot icon and pin dropdowns.
- A second "Wait" block (orange) with "0.25" in a white circle and "秒" (seconds) to its right.

A small red arrow icon is visible at the bottom right of the code area.

積木程式堆疊

(6-5) 副程式「小紅人行走」

定義 小紅人行走

變數 秒數 設為 0.75

重複 3 次

重複 10 次

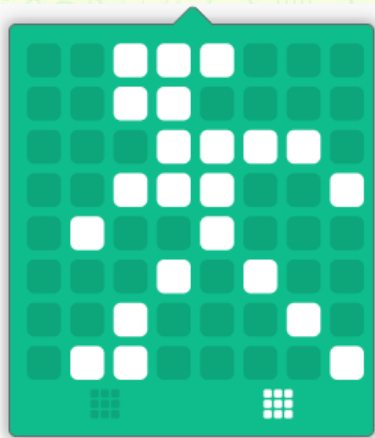
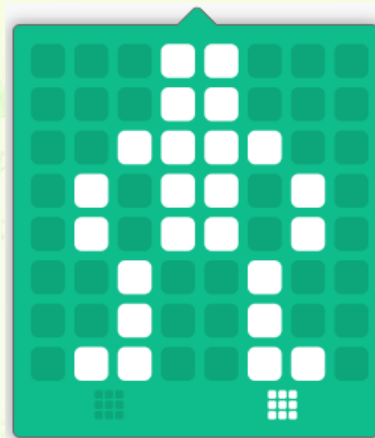
設MAX7219 8*8LED矩陣 [人] ,DIN 12 CS A4 CLK A5

等待 秒數 秒

設MAX7219 8*8LED矩陣 [人] ,DIN 12 CS A4 CLK A5




等待 秒數 秒

變數 秒數 改變 -0.25



演算法步驟

「紅綠燈系統」演算法步驟：

01	副程式：紅燈狀態
02	重複無限次迴圈開始
03	 判斷 ：如果搖桿的按扭被壓下
04	 成立 ：副程式：綠燈狀態
05	副程式：小紅人行走
06	副程式：紅燈狀態
	重複迴圈結束 



積木程式堆疊

(7-1) 紅綠燈系統

當  被點擊

紅燈狀態

重複無限次

如果  讀取數位腳位 7  INPUT_PULLUP  註  = 1  那麼

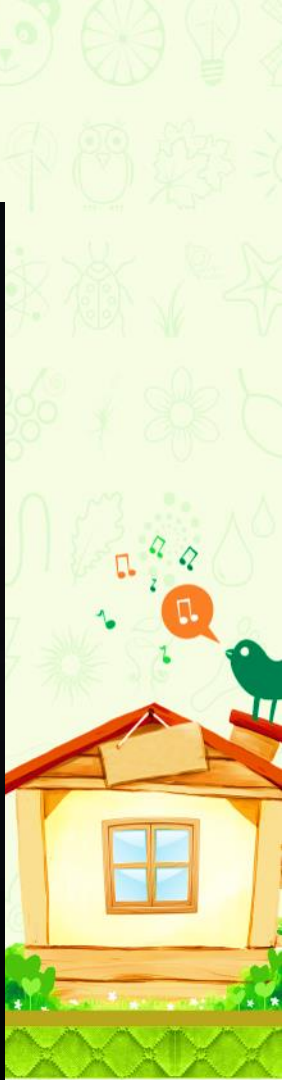
綠燈狀態

小紅人行走

紅燈狀態



明德國中校園進出入系統影片介紹



明德國中校園進出入系統大綱

一、 情境主題：停車場自動化

二、 情境目的：

- 點綠旗時：放下柵欄
- 車子離開校園時：利用超音波感測判斷，打開柵欄，並顯示對應紅燈或音響。
- 車子進入校園時：利用搖桿按鈕按下則開啟柵欄

三、 情境任務：

1. 基本狀態：柵欄放下，超音波感測器啟動，紅燈恢復基本狀態。
2. 感應狀態：感應有車輛進入，開啟柵欄，燈號顯示紅燈，並出現警示音響。
3. 回復狀態：車輛離開，放下柵欄，燈號恢復基本狀態，或警示音響停止。

明德國中校園進出入系統程式實作

定義 閃紅燈

重複 3 次

設定數位腳位 9 輸出為 高電位(1) 註

等待 1 秒

定義 放下柵欄

伺服馬達 腳位 6 角度為 0 註

設定數位腳位 9 輸出為 低電位(0) 註

等待 1 秒

定義 打開柵欄

伺服馬達 腳位 6 角度為 90 註

等待 1 秒

明德國中校園進出入系統程式實作

定義 進入校園

重複無限次

如果 讀取數位腳位 7 INPUT_PULLUP 註 = 1 那麼

閃紅燈

打開柵欄



定義 離開校園

變數 dist 設為 0

重複無限次

變數 dist 設為 超音波 腳位 Trig腳位A2, Echo腳位A3 註

如果 超音波 腳位 Trig腳位A2, Echo腳位A3 註 < 10 那麼

說出 dist

閃紅燈

打開柵欄

否則

放下柵欄



明德國中校園進出入系統程式實作



明德國中校園進出入系統程式實作



當 被點擊

伺服馬達 腳位 6 角度為 0 註

變數 dist 設為 0

變數 dist 設為 超音波 腳位 Trig腳位A2, Echo腳位A3 註

重複無限次

如果 讀取數位腳位 7 INPUT_PULLUP 註 = 1 那麼

打開柵欄

如果 超音波 腳位 Trig腳位A2, Echo腳位A3 註 < 10 那麼

說出 dist

打開柵欄

否則

放下柵欄



明德國中校園進出入系統程式實作-8*8點矩陣之應用

```
定義 1  
  設MAX7219 8*8LED矩陣, DIN 12 CS A4 CLK A5  
  變數 i 設為 1  
  變數 x 設為 0  
  變數 y 設為 0  
  重複 8 次  
    變數 x 改變 1  
    設MAX7219 8*8LED矩陣, 列 x 行 i 值 1  
    等待 0.1 秒  
  重複 8 次  
    變數 y 設為 0  
    重複 8 次  
      變數 y 改變 1  
      設MAX7219 8*8LED矩陣, 列 y 行 i 值 1  
      等待 0.1 秒  
    變數 i 改變 1  
  設MAX7219 8*8LED矩陣, DIN 12 CS A4 CLK A5
```

當 被點擊

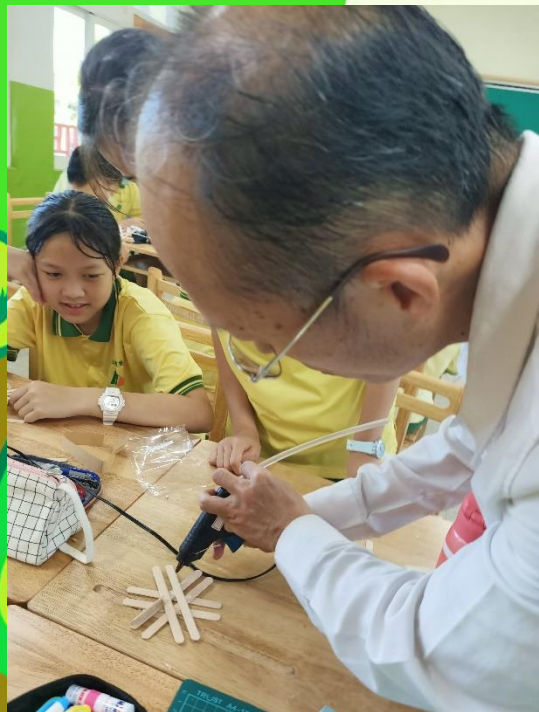
如果 讀取數位腳位 7 INPUT_PULLUP 註 = 1 那麼

1

炫彩涼風扇~課程尚在進行中



涼風扇之扇葉製作



Thinkcad circuits-串聯與並聯



認識直流(減速)馬達N20

- N20兩端電位相等時，不會轉動。電位差距



- 腳位：D2、D3

直流減速馬達 腳位 腳位2、3 轉速為 0 註

- 轉動馬達：

直流減速馬達 腳位 腳位2、3 轉速為 200 註

- 小試身手：改變轉動方向

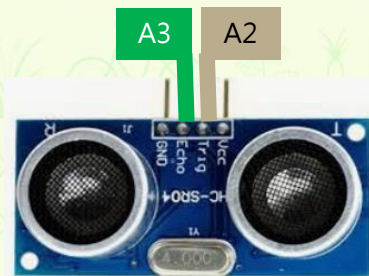
直流減速馬達 腳位 腳位2、3 轉速為 -200 註



認識HC-SR04超音波感測器

1. 認識超音波模組：

- (1) 探測距離：2CM~450CM
- (2) 感應角度：不大於15度
- (3) 接收方式：Vcc(正極)、Gnd(接地)、Trig(控制端)、Echo(接收端)。



2. 超音波腳位：

- (1) Trig(控制端)：A2腳位
- (2) Echo(接收端)：A3腳位
- (3) 超音波感測：說出前方物體距離



3. 小試身手：

- 超音波偵測前方物體。
- 距離 < 10公分 - 說出[有人進來]
- 距離 > 10公分 - 說出[人離開了]

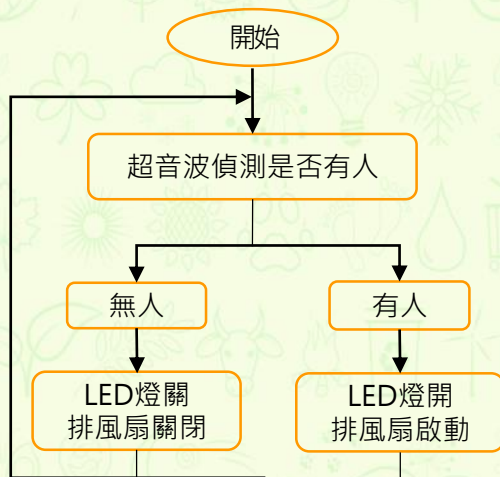


情境分析及情境流程圖

(3) 情境分析：

1. 利用超音波偵測，是否有人進入。
2. 無人時，LED燈關、排風扇關閉。
3. 人進來，LED燈亮、排風扇啟動。




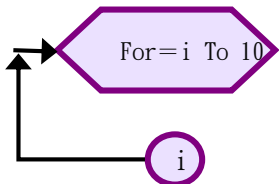

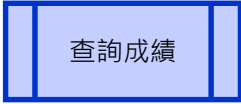

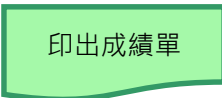
(4) 情境流程圖：



流程圖符號(1)

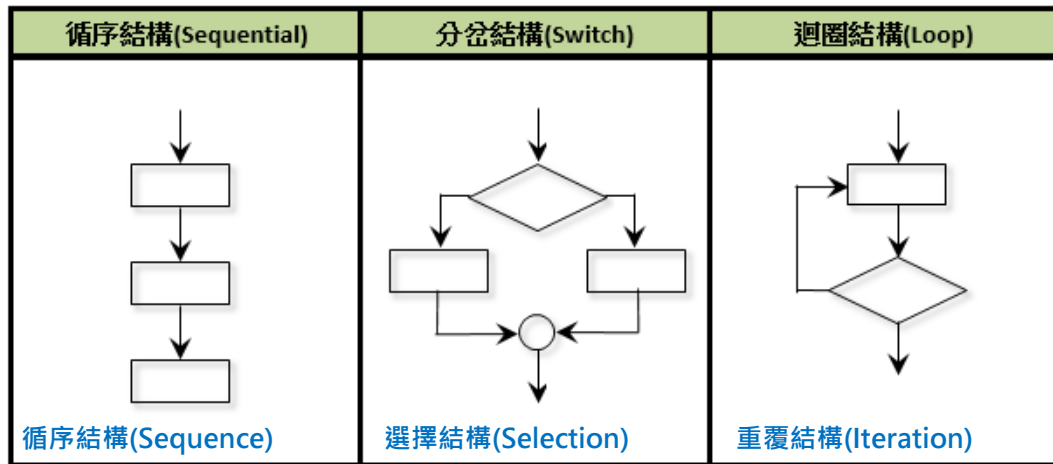
符號	名稱	意義	範例
	開始 (Start) 終止 (End)	表示程式的開始或結束	
	路徑(Path)	表示流程進行的方向	
	輸入(Input) 輸出(Output)	表示資料的輸入或結果的輸出	
	處理(Process)	表示執行或處理某一項工作	

流程圖符號(2)

符號	名稱	意義	範例
	決策判斷 (Decision)	針對某一條件進行 判斷	
	迴圈 (Loop)	表示迴圈控制變數 的初始值及終值	
	副程式 (Subroutine)	用以表示一群已經 定義流程的組合	
	報表(Document)	指列印出的報表文 件	

程式流程三種結構

程式流程控制的三種結構

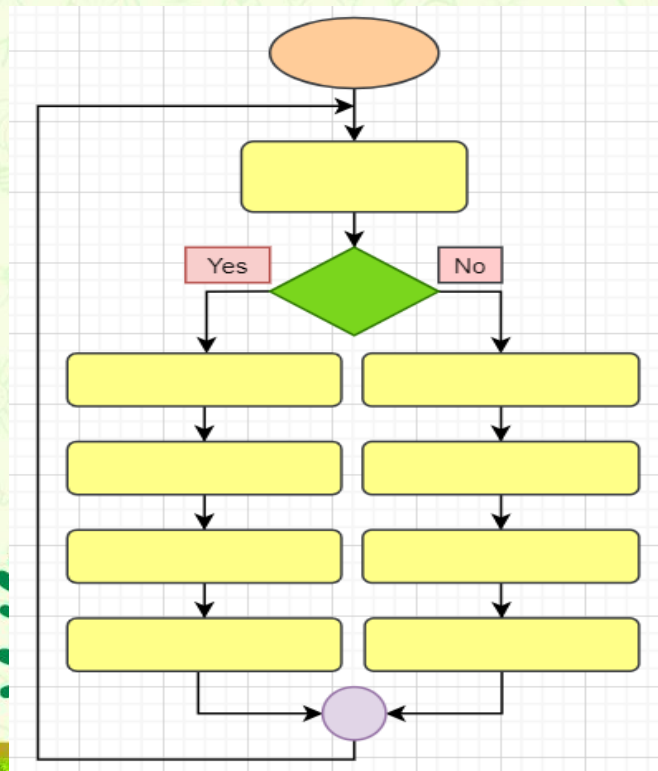


情境流程圖 vs 程式流程圖(學生填空用)

(4) 情境流程圖：



(5) 程式流程圖：



程式流程圖(學生填空用)

(5) 程式流程圖：利用以下選項，填入右邊適當的空格內。

開始 距離 >10 距離 <10

設定超音波腳位 =A2、A3

設定腳位9(R)=低電位

設定腳位9(R)=高電位

設定腳位10(G)=低電位

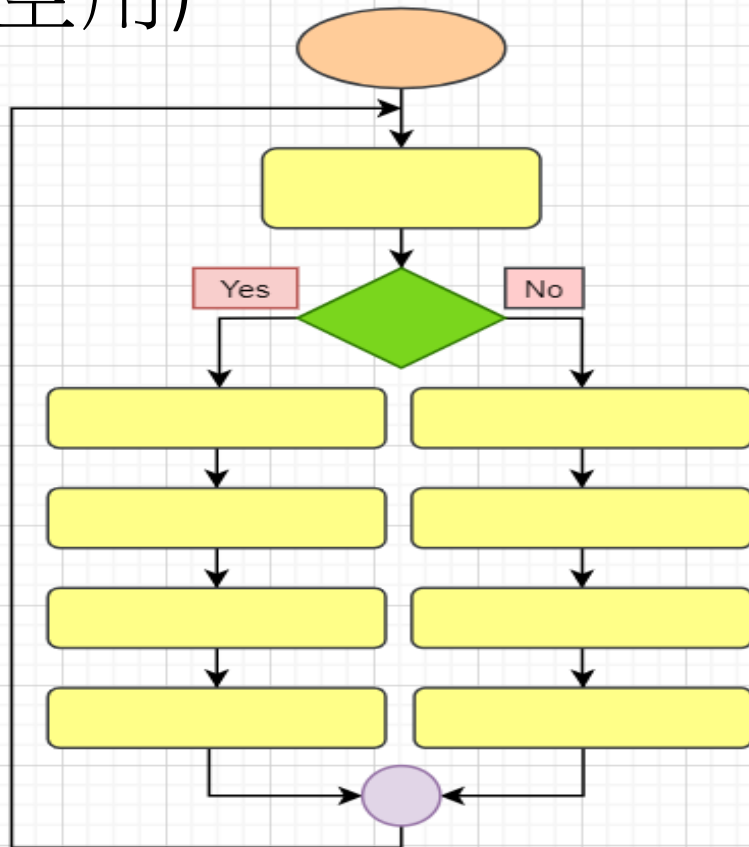
設定腳位10(G)=高電位

設定腳位11(B)=低電位

設定腳位11(B)=高電位

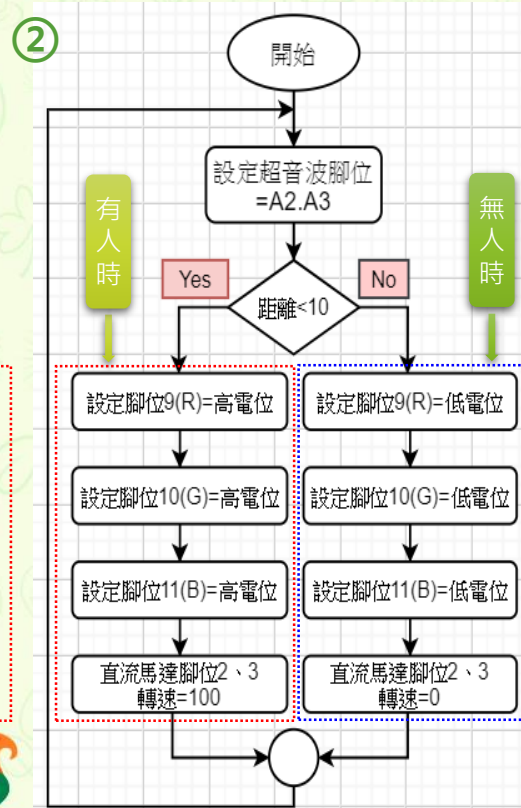
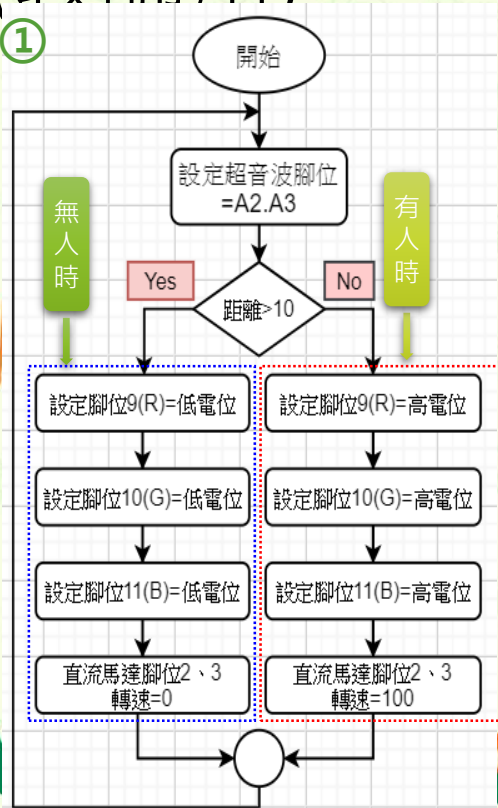
直流馬達腳位2、3轉速=0

直流馬達腳位2、3轉速=100



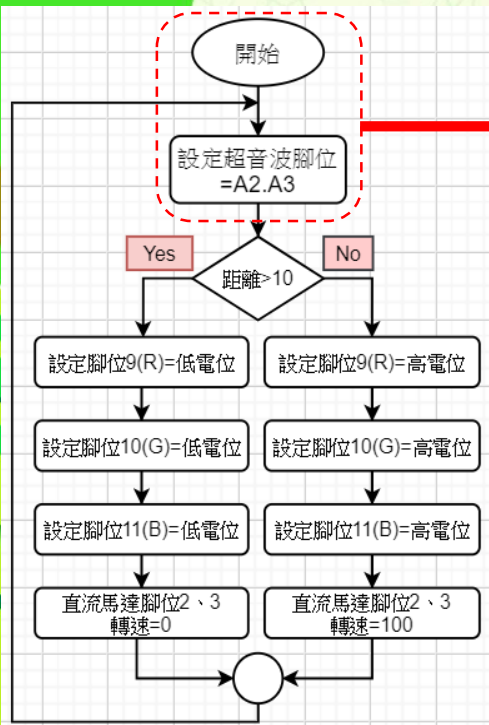
程式流程圖(教師用)

(5) 程式流程圖：①



程式流程圖 vs 積木程式堆疊

(5) 程式流程圖

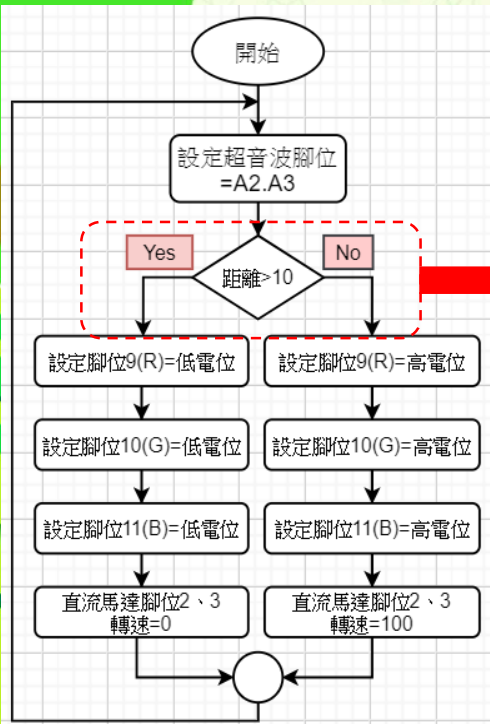


(6) 積木程式堆疊 程式1



程式流程圖 vs 積木程式堆疊

(5) 程式流程圖

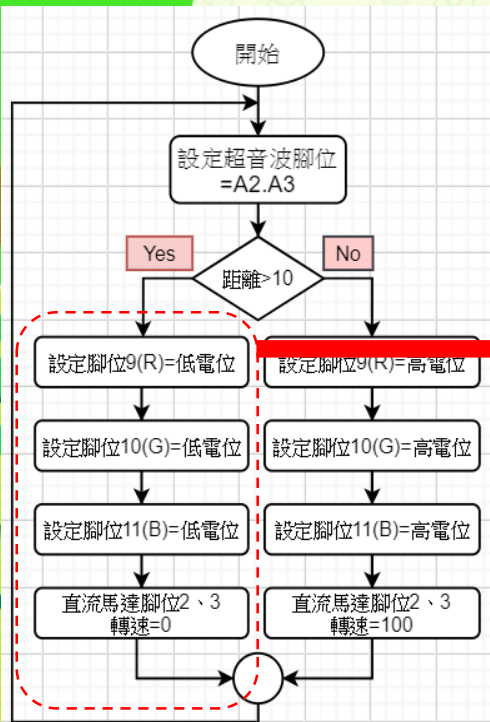


(6) 積木程式堆疊
程式2



程式流程圖 vs 積木程式堆疊

(5) 程式流程圖



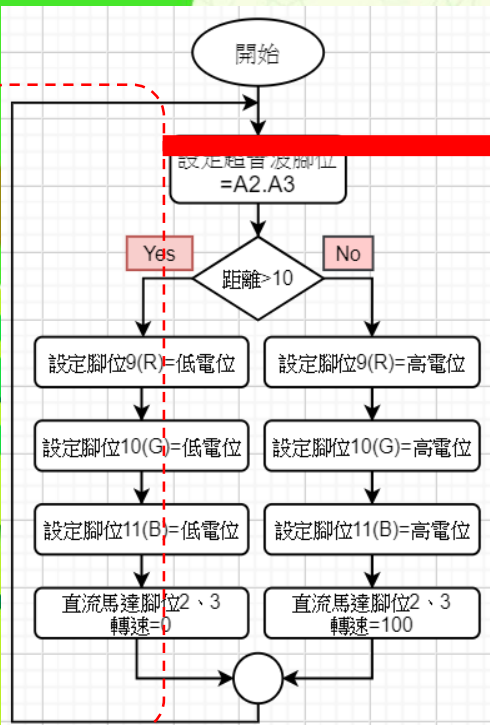
(6) 積木程式堆疊
程式3

Scratch code blocks for the ultrasonic sensor logic:

- 當被點擊 (When clicked)
- 變數 超音波 設為 超音波 腳位 Trig腳位A2, Echo腳位A3 註 (Set variable ultrasonic to ultrasonic pins Trig pin A2, Echo pin A3)
- 如果 超音波 > 10 那麼 (If ultrasonic > 10 then)
- 設定數位腳位 9 輸出為 低電位(0) 註 (Set digital pin 9 output to low (0))
- 設定數位腳位 10 輸出為 低電位(0) 註 (Set digital pin 10 output to low (0))
- 設定數位腳位 11 輸出為 低電位(0) 註 (Set digital pin 11 output to low (0))
- 直流減速馬達 腳位 腳位2、3 轉速為 0 註 (DC motor pins 2,3 speed is 0)
- 否則 (Otherwise)

程式流程圖 vs 積木程式堆疊

(5) 程式流程圖



(6) 積木程式堆疊
程式5

當 被點擊

重複無限次

變數 超音波 設為 超音波 腳位 Trig腳位A2, Echo腳位A3 註

如果 超音波 > 10 那麼

- 設定數位腳位 9 輸出為 低電位(0) 註
- 設定數位腳位 10 輸出為 低電位(0) 註
- 設定數位腳位 11 輸出為 低電位(0) 註
- 直流減速馬達 腳位 腳位2、3 轉速為 0 註

否則

- 設定數位腳位 9 輸出為 高電位(1) 註
- 設定數位腳位 10 輸出為 高電位(1) 註
- 設定數位腳位 11 輸出為 高電位(1) 註
- 直流減速馬達 腳位 腳位2、3 轉速為 100 註

The block stack replicates the logic of the flowchart. It begins with a '當 被點擊' (When clicked) block, followed by a '重複無限次' (Repeat forever) loop. Inside the loop, there is a '變數 超音波 設為 超音波 腳位 Trig腳位A2, Echo腳位A3' block. This is followed by an '如果 超音波 > 10 那麼' (If ultrasonic > 10 then) conditional block. The '那麼' (then) branch contains four blocks: '設定數位腳位 9 輸出為 低電位(0)', '設定數位腳位 10 輸出為 低電位(0)', '設定數位腳位 11 輸出為 低電位(0)', and '直流減速馬達 腳位 腳位2、3 轉速為 0'. The '否則' (otherwise) branch contains four blocks: '設定數位腳位 9 輸出為 高電位(1)', '設定數位腳位 10 輸出為 高電位(1)', '設定數位腳位 11 輸出為 高電位(1)', and '直流減速馬達 腳位 腳位2、3 轉速為 100'. A red dashed box highlights the '變數' block and the '如果' block.

程式流程圖 vs 積木程式堆疊

(6) 積木程式堆疊

當 **被點擊**

重複無限次

變數 **超音波** 設為 **超音波 腳位 Trig腳位A2, Echo腳位A3** 註

如果 **超音波 > 10** 那麼

- 設定數位腳位 **9** 輸出為 **低電位(0)** 註
- 設定數位腳位 **10** 輸出為 **低電位(0)** 註
- 設定數位腳位 **11** 輸出為 **低電位(0)** 註
- 直流減速馬達 腳位 **腳位2、3** 轉速為 **0** 註

否則

- 設定數位腳位 **9** 輸出為 **高電位(1)** 註
- 設定數位腳位 **10** 輸出為 **高電位(1)** 註
- 設定數位腳位 **11** 輸出為 **高電位(1)** 註
- 直流減速馬達 腳位 **腳位2、3** 轉速為 **100** 註

Diagram description: This block diagram shows a program flow. It starts with a 'When clicked' event block, followed by a 'Repeat infinitely' loop. Inside the loop, there is a 'Set ultrasonic sensor to Trig pin A2, Echo pin A3' block. Then, an 'If ultrasonic sensor > 10 then' conditional block. The 'then' branch contains four blocks: setting pins 9, 10, and 11 to low (0) and setting the motor speed to 0. The 'else' branch contains four blocks: setting pins 9, 10, and 11 to high (1) and setting the motor speed to 100. Red dashed boxes and arrows highlight the mapping from the 'then' branch to the 'Define Turn Off Light' block and from the 'else' branch to the 'Define Turn On Light' block.

(7) 副程式運用

函式積木

Diagram description: A red circle icon with the text '函式積木' (Function Block) below it.

定義 關燈

- 設定數位腳位 **9** 輸出為 **低電位(0)** 註
- 設定數位腳位 **10** 輸出為 **低電位(0)** 註
- 設定數位腳位 **11** 輸出為 **低電位(0)** 註

Diagram description: A pink block titled 'Define Turn Off Light'. It contains three blocks: setting pins 9, 10, and 11 to low (0). A red arrow from the 'then' branch of the flowchart in (6) points to this block.

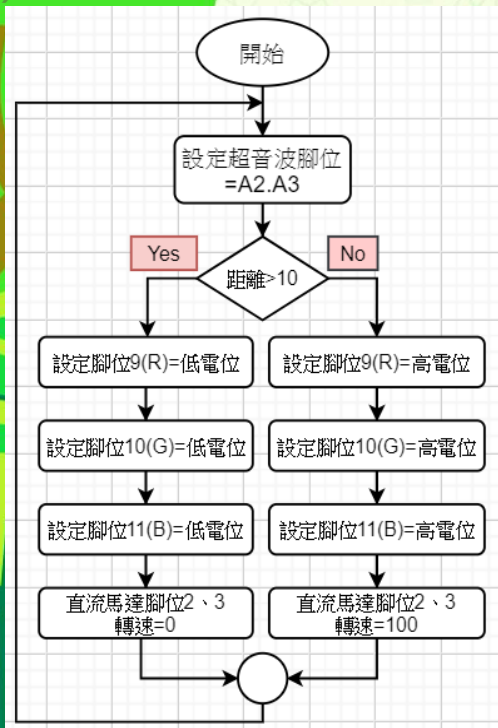
定義 開燈

- 設定數位腳位 **9** 輸出為 **高電位(1)** 註
- 設定數位腳位 **10** 輸出為 **高電位(1)** 註
- 設定數位腳位 **11** 輸出為 **高電位(1)** 註

Diagram description: A pink block titled 'Define Turn On Light'. It contains three blocks: setting pins 9, 10, and 11 to high (1). A red arrow from the 'else' branch of the flowchart in (6) points to this block.

程式流程圖 vs 積木程式堆疊

(5) 程式流程圖



(7) 副程式運用

The Scratch code block structure is as follows:

- 當 被點擊** (When clicked) block containing:
 - 重複無限次** (Repeat forever) loop containing:
 - 變數 超音波 設為 超音波 腳位 Trig腳位A2, Echo腳位A3** (Set variable ultrasonic to ultrasonic pin Trig pin A2, Echo pin A3)
 - 如果 超音波 > 10 那麼** (If ultrasonic > 10 then) block containing:
 - 關燈** (Turn off light) block containing:
 - 直流減速馬達 腳位 腳位2、3 轉速為 0** (DC speed-reducing motor pin 2, 3 speed 0)
 - 否則** (Otherwise) block containing:
 - 開燈** (Turn on light) block containing:
 - 直流減速馬達 腳位 腳位2、3 轉速為 100** (DC speed-reducing motor pin 2, 3 speed 100)

報告完畢，謝謝大家！

