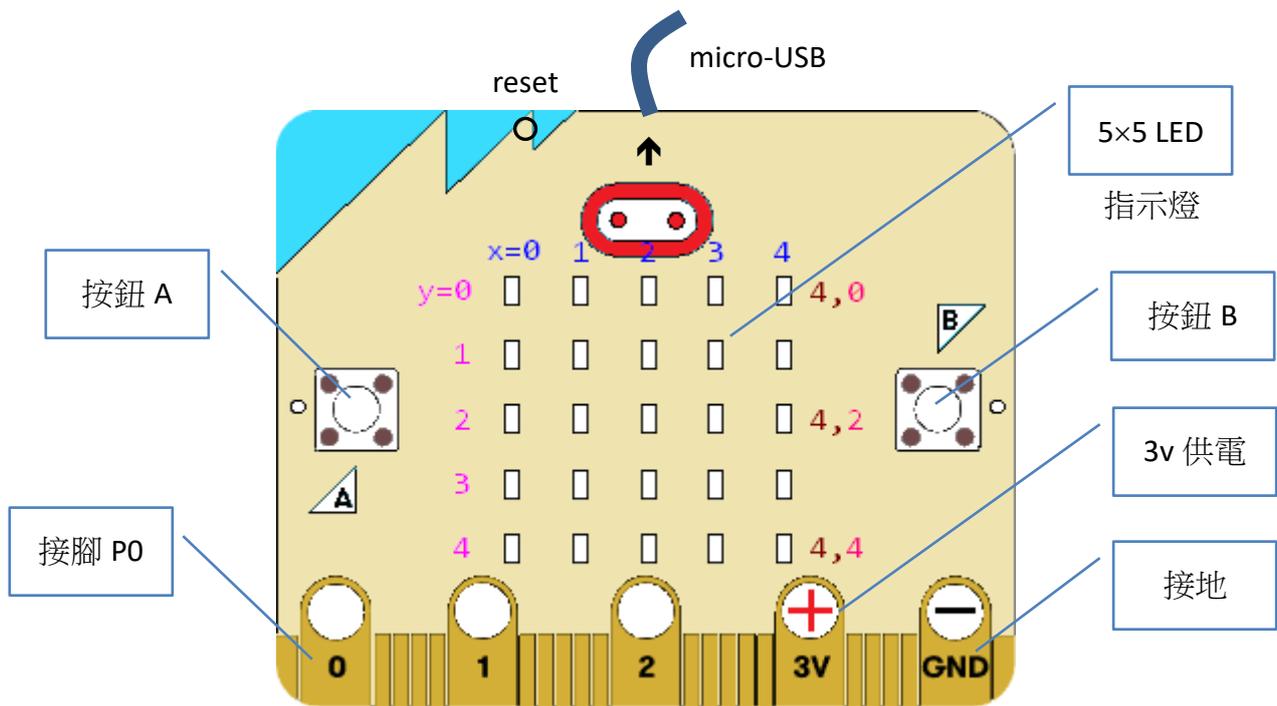


單板電腦 **micro:bit**

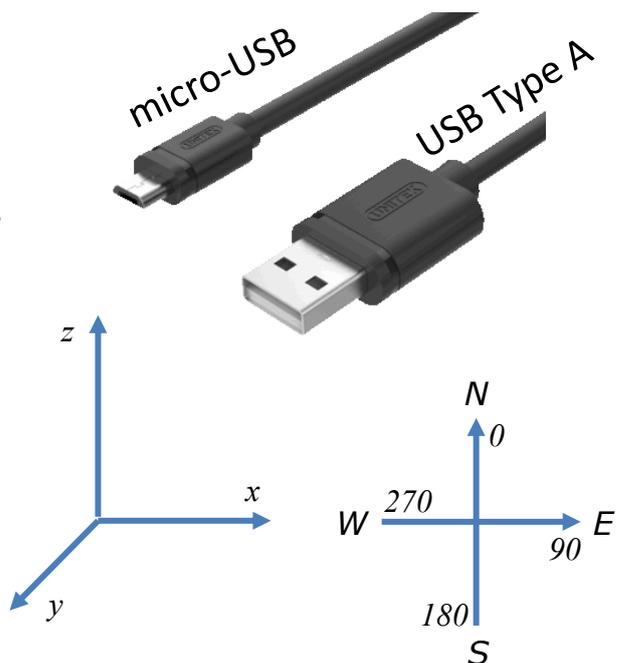
程式編寫

官方網頁：makecode.microbit.org



- 5x5 個可編程 LED (指示燈)
- 2 個可程式化按鈕 A、B
- 電路連接(pin)接腳位 (P0,P1,P2,3V,Gnd)
- 光度和溫度感測器
- 動作感測器 (加速儀、羅盤)
- 無線通訊，透過 Radio 和藍牙
- micro-USB 界面 (連接電腦)
- 建議使用 Google Chrome 編程
- 網上資源

初中
電腦



A. 基本練習：makecode.microbit.org (線上編程)



1. 新增專案： (9) 主頁 (3) 積木 blocks (4) 程式碼 Javascript



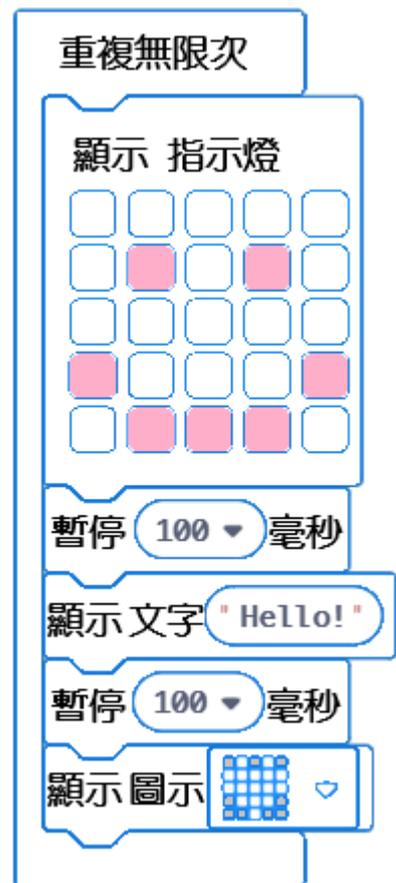
2. 設定：語言
3. 積木：基本、更多 隨意拉出積木 (如右圖) → 留意左邊的模擬器
4. 編程：點選積木、Javascript (程式碼、可修改)
5. 檔案名稱：班別+號數 (例如：2A01)

6. 存檔

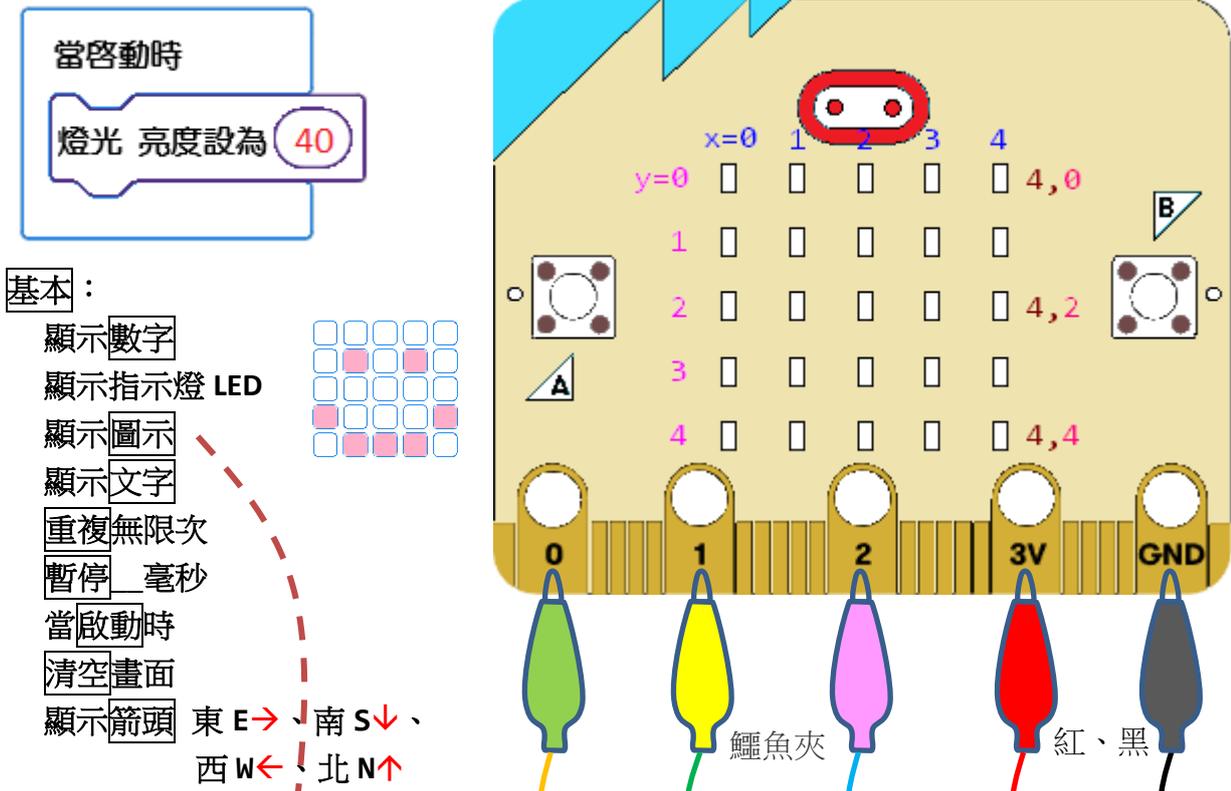
7. USB 連接電腦

8. 到本機→下載 把 2A01.hex 檔上載到→MICROBIT 去 (及 D:\)

9. 離開：按主頁



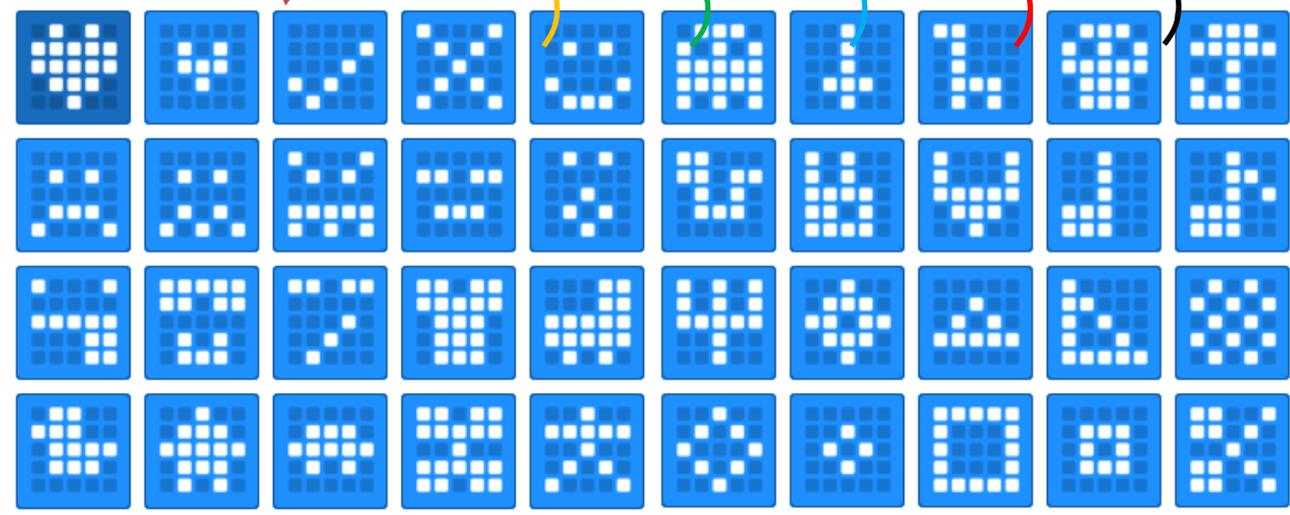
B. 積木 Blocks



當啟動時
燈光 亮度設為 40

- 基本：
- 顯示數字
 - 顯示指示燈 LED
 - 顯示圖示
 - 顯示文字
 - 重複無限次
 - 暫停 毫秒
 - 當啟動時
 - 清空畫面
 - 顯示箭頭 東 E → 南 S ↓、西 W ← 北 N ↑

基本 → 顯示圖示：



<p>輸入：</p> <ul style="list-style-type: none"> 當按鈕 A/B 被按下 當姿勢 晃動 發生 加速度感測器 (mg) 光線度感測值 方位感測值(°) 溫度感測值(°C) 運行時間(ms) running time 	<p>燈光 LED：</p> <ul style="list-style-type: none"> 點亮 x,y 不點亮 x,y 點的狀況 x,y 點的狀況切換 x,y 點亮長條圖 顯示值為 n 最大值為 m 亮度 點亮 x,y 亮度 255 燈光亮度設為 255 停止動畫
--	--

C. 數學 Maths

$(3 + 4) \times 8$
 $\frac{\quad}{2}$

$13 \div 2 = 6\frac{1}{2}$
 $13\%2 = 1$ (取餘數)
 $12\%2 = 0$ (取餘數)

Remainder 餘數

min
 max

$\sqrt{25} = 5$

若 $n=256$
 $\frac{n}{1024} = \frac{?}{256}$

3 + 4 7 x 8
 9 - 6 56 ÷ 2

13 ÷ 2 的餘數

6 和 9 的最小值

6 和 9 的最大值

-5 的絕對值

平方根 25 四捨五入 3.6

隨機取數 0 到 10

制限 n 最低 1 最高 12

對應 n 從低 0 到 1023 至低 0 至高 255

隨機取布林值 True/False

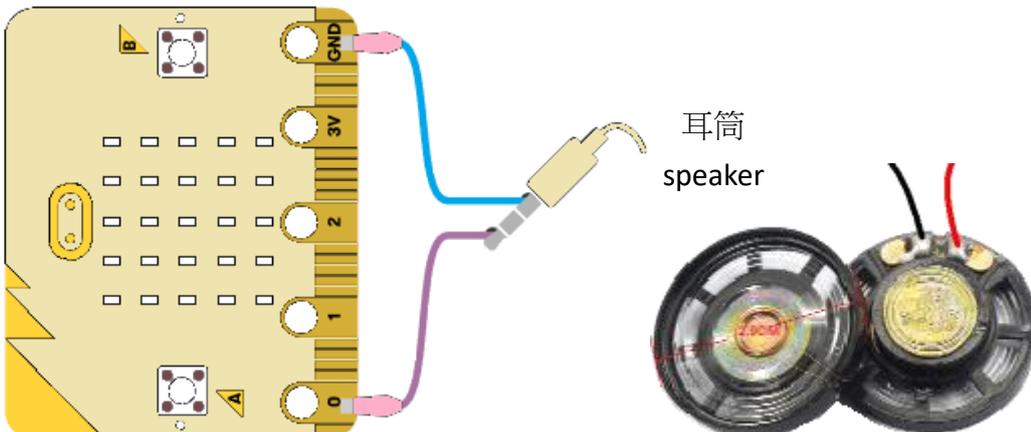
$(3 + 4) \times 8$
 $\frac{\quad}{2}$

$\times 8$

$\div 2$

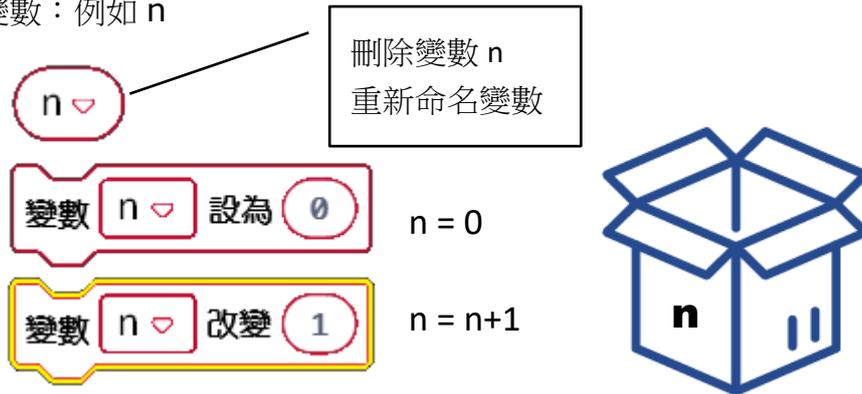
map(對應)：把 n 由範圍 0..1023，按比例化為範圍 0..255。例如：n=20，結果是 5。

四捨五入	$\text{round}(3.14) = 3$	$\text{round}(3.54) = 4$
上捨入(無條件進位)	$\text{ceiling}(3.14) = 4$	$\text{ceiling}(3.94) = 4$
下捨入(無條件捨去)	$\text{floor}(3.14) = 3$	$\text{floor}(3.94) = 3$
只取整數(截斷取整)	$\text{truncate}(3.14) = 3$	$\text{truncate}(3.94) = 3$



D. 變數 variables :

1. 建立一個變數：例如 n



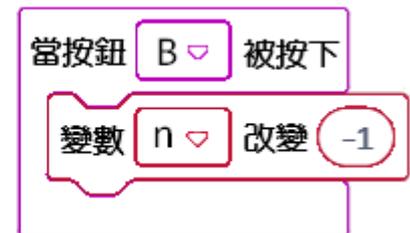
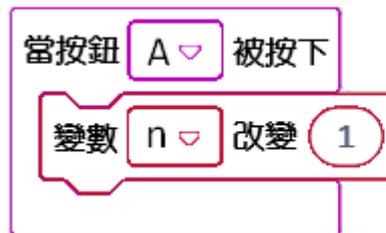
2. 初始化($n=0$)：當啟動時，把變數 n 設為 0。



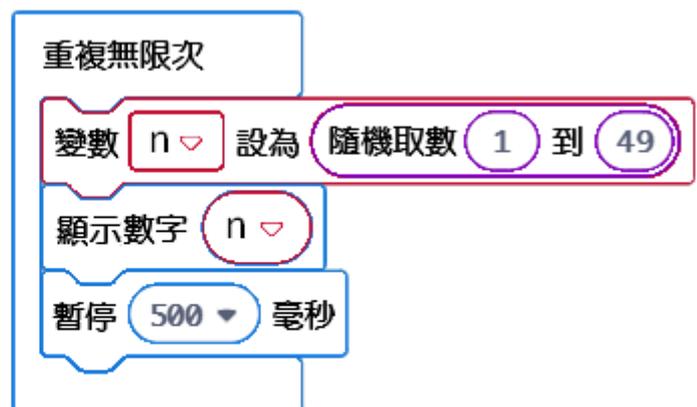
3. 改變變數值： $+1, -1, +5, \dots$

$n = n + 1$ (加 1)

$n = n - 1$ (減 1)



4. 變數 顯示、運算：

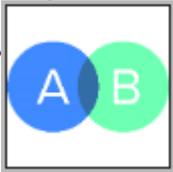


開啟 HEX 檔：把 HEX 檔(drag and drop)拖到網頁 makecode.microbit.org 上。

練習一：答對加 1 分，答錯扣 1 分。

當啟動時

score=0



2 分

當按鈕 A 被按下

顯示文字 Yes,
score+1

2 分

當按鈕 B 被按下

顯示文字 No,
score-1

2 分

變數：
score

當按鈕 A+B 被按下

顯示數字：score

2 分

當姿勢 晃動 發生

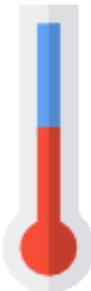
score=0

2 分

練習二：溫度計。

當啟動時

temp=0



2 分

當按鈕 A 被按下

顯示溫度 temp(°C)
temp(-18): **Cool**
temp(18-24): **Good**
temp(25-): **Hot**

3 分

當按鈕 B 被按下

顯示溫度 temp(°F)
temp(-64): ☹
temp(64-76): ☺
temp(77-): ☹

3 分

變數：
temp

當按鈕 A+B 被按下

顯示數字：temp

2 分

$$F = \frac{9}{5}C + 32$$

記得存檔
把 HEX 抄到 D:\
下課後登出

練習三：隨機數 (0-9)。若是單數，則 **count** 加 **1**。

當啟動時

count=0

2 分

當按鈕 A 被按下

產生隨機數 x(0-9)
顯示 x
若 x 是單數：
則 count+1

4 分

當按鈕 B 被按下

清空畫面

2 分

變數：
count
x

提示：
若 $\frac{x}{2}$ 的餘數 = 0，
則 x 必定是雙數。

當按鈕 A+B 被按下

顯示數字：count

2 分

當姿勢 晃動 發生

...

-

練習四：猜數字(1-6)。

當啟動時

產生隨機數 ans(1-6)
顯示：guess

1 分

當按鈕 A 被按下

guess+1
若 guess>6, guess=1
顯示 guess

3 分

當按鈕 B 被按下

若(guess=ans)
則 顯示 Y
否則顯示 N
並顯示+/- (作提示)

3 分

變數：
guess
ans

記得存檔
把 HEX 抄到 D:\
下課後登出

當按鈕 A+B 被按下

重設 reset：
產生隨機數 ans(1-6)
guess=0

2 分

當姿勢 晃動 發生

guess=0
顯示 guess

1 分

練習五：指南針 Compass。

重複無限次

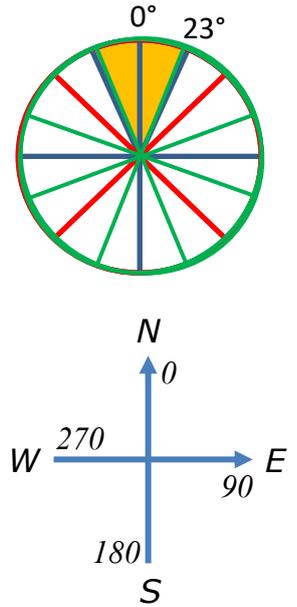
設變數 n =方位感測值(0-359)

根據 n 讀數 (見右圖) 顯示方位 $\leftarrow \uparrow \rightarrow \downarrow \swarrow \nwarrow$

若 $n < 23$ ，則顯示箭頭 \uparrow ；
 否則 若 $n < 68$ ，則顯示...
 否則 若...

[1 分]

[9 分]



若 $n < 23^\circ$	N 北
若 $n < 68^\circ$	NW 西北
若 $n < 113^\circ$	W 西
若 $n < 158^\circ$	SW 西南
若 $n < 203^\circ$	S 南
若 $n < 248^\circ$	SE 東南
若 $n < 293^\circ$	E 東
若 $n < 338^\circ$	NE 東北
否則	N 北

挑戰：永遠指向北方

無論把 micro:bit 轉向那一個方向，
 箭頭 \uparrow 永遠指向北方。

練習六：顯示別人方向，廣播發送文字(N,E,S,W)。

二人一組，學生 x 及學生 y 。

重複無限次

廣播群組設為 y ($y > x$)

設變數 $degree$ =方位感測值(0-359)

根據 $degree$ 讀數 (見右圖) 顯示方位(NESW)

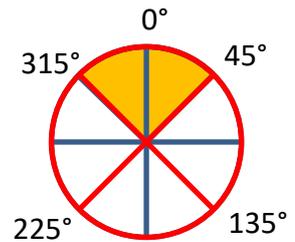
若 $degree < 45$ ，則廣播發送文字 N；
 否則 若 $degree < 135$...

[1 分]

[1 分]

[4 分]

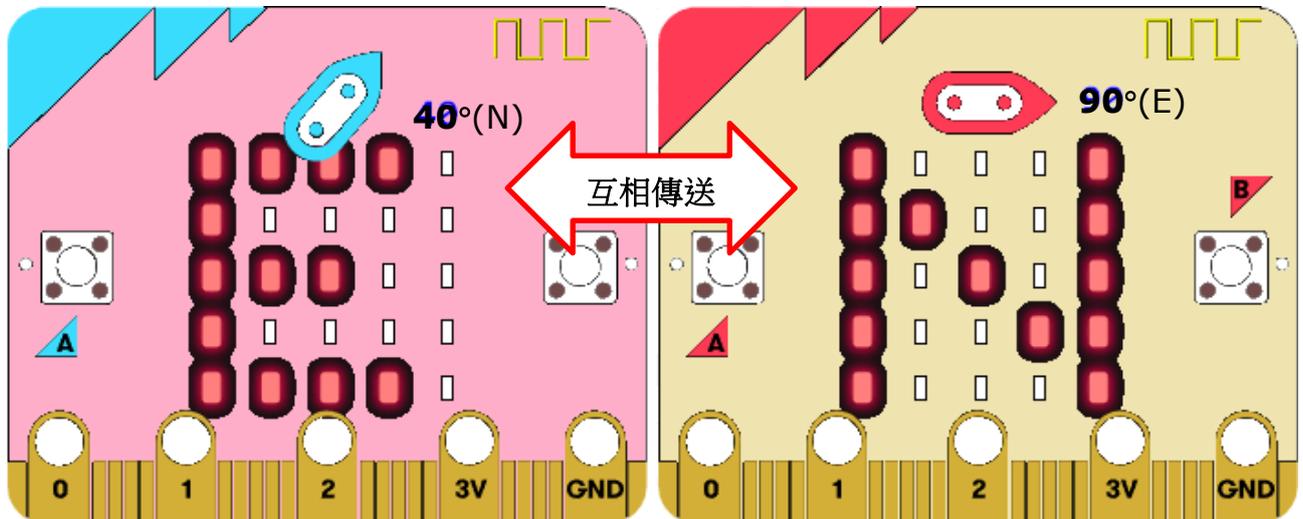
[1+3 分]



N 北:	$degree < 45$
E 東:	$degree < 135$
S 南:	$degree < 225$
W 西:	$degree < 315$
N 北:	$degree > 315$

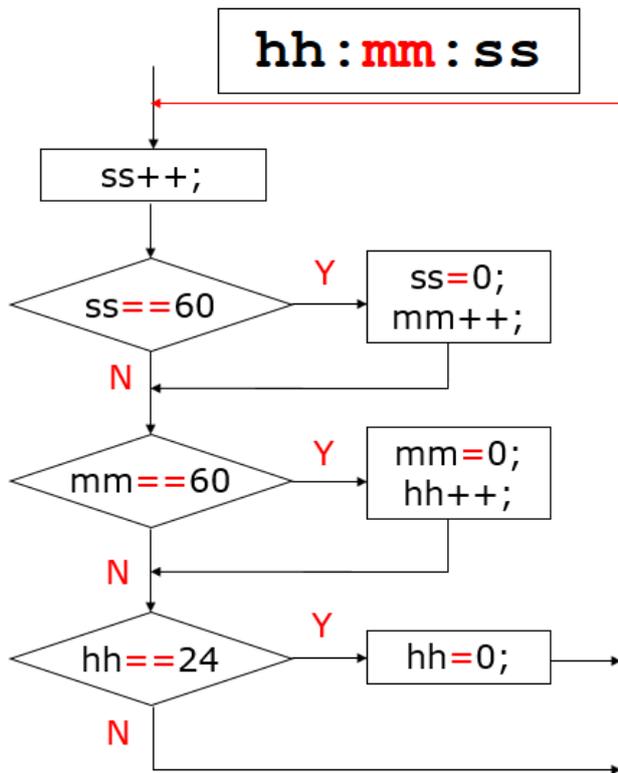
當收到廣播 receivedString \leftarrow 收到別人傳送過來的文字訊息

- 顯示文字 receivedString (NESW)
- 顯示箭頭 $\uparrow \rightarrow \downarrow \leftarrow$



40°(N) ↔ 90°(E)

練習七：數碼時鐘。



23:59:59

重複無限次 forever

ss=ss+1 (秒針+1)
 如果 ss=60 · 則 ss=0, mm=mm+1
 如果 mm=60 · 則 mm=0, hh=hh+1
 如果 hh=24 · 則 hh=0
 暫停 1 秒 pause 1000

5 分

當啟動時 onstart

hh=23
mm=59
ss=40

1.5 分

當按鈕 A 被按下

show string
顯示文字(字串組合)
hh:mm:ss

2 分

當按鈕 B 被按下

hh=mm=ss=0

1.5 分

變數：
hh, mm, ss

記得存檔
把 HEX 抄到 D:\
下課後登出

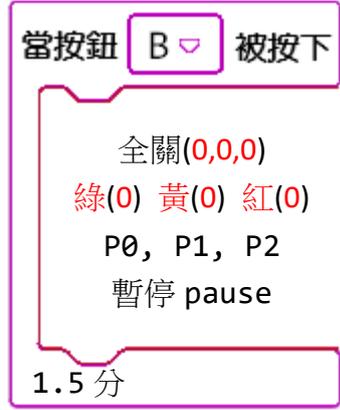
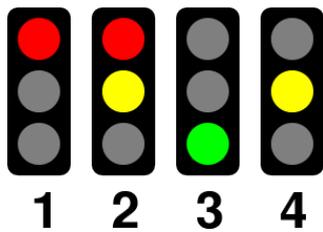
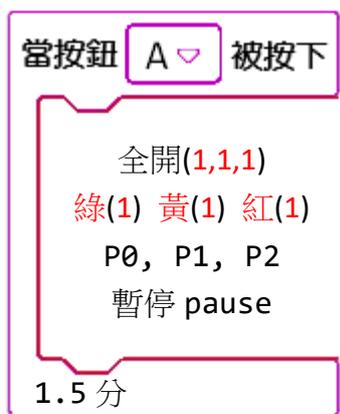
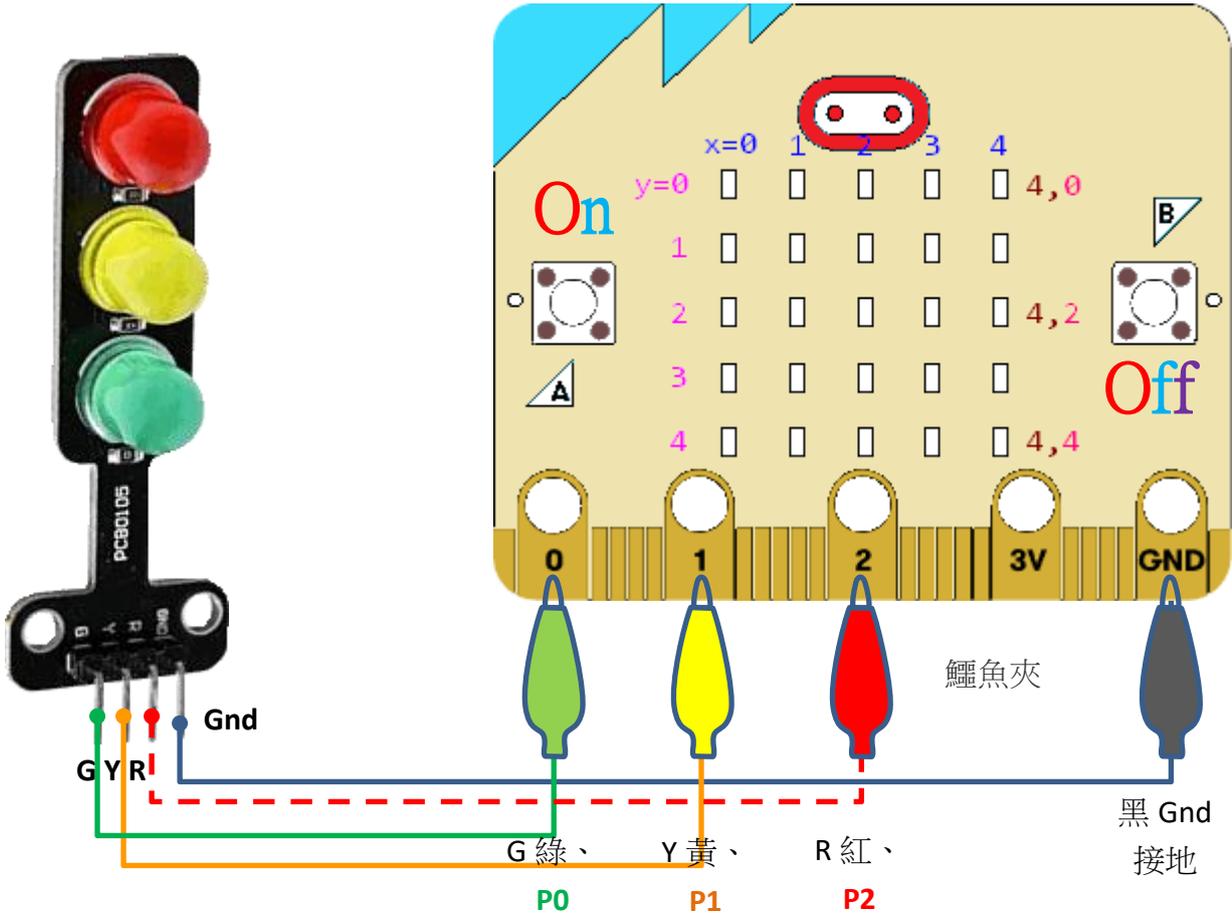
當按鈕 A+B 被按下

Text join 字串組合

顯示文字

hh
:
mm
:
ss

練習八：交通燈 Traffic Light(1)



交通燈 <https://mrszeto.net/s2/arduino/img.htm#traffic>

練習八：交通燈 Traffic Light(2)

重複無限次

呼叫 redOn
 呼叫 red-yellowOn
 呼叫 greenOn
 呼叫 yellowOn

2分

Functions
 f(x) 函式

定義函式 redOn

數位信號寫入 引腳 P2 數字 1

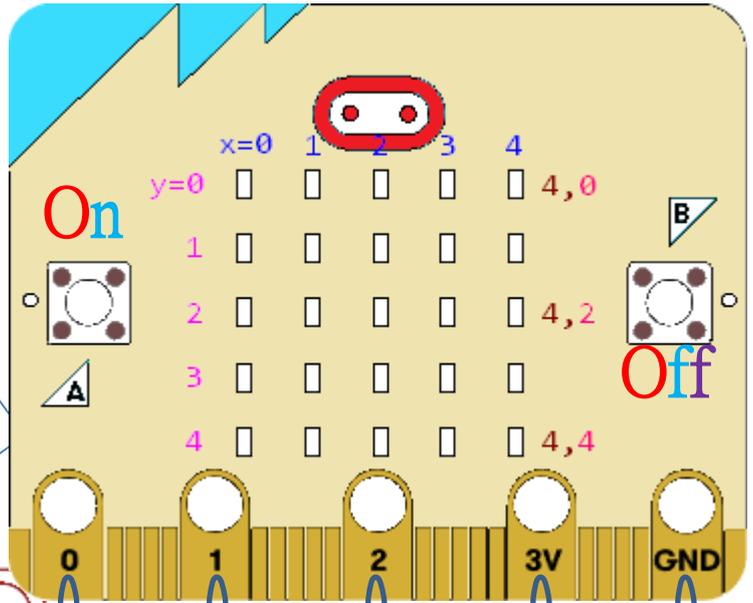
暫停 1000 毫秒

數位信號寫入 引腳 P2 數字 0

暫停 1000 毫秒

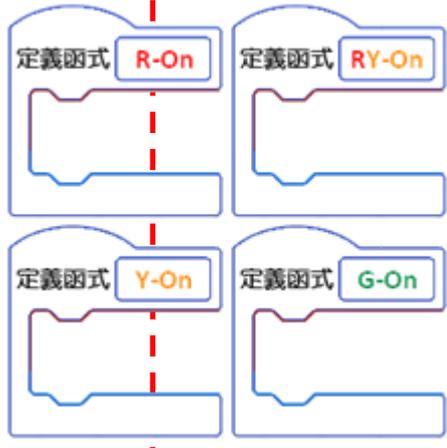
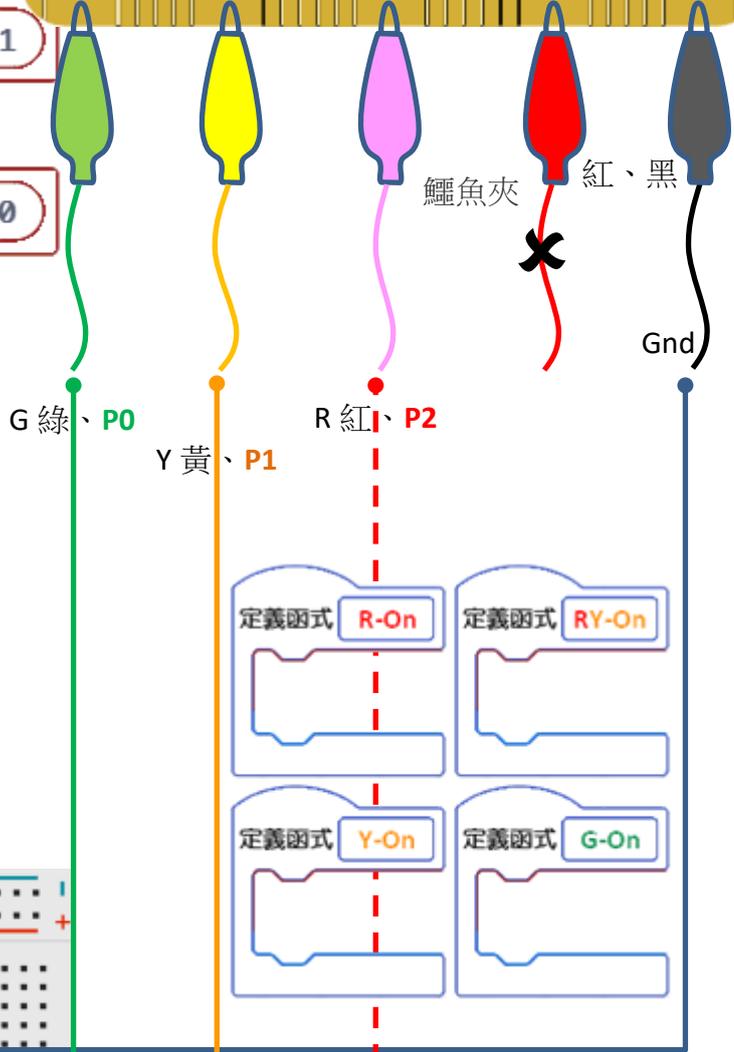
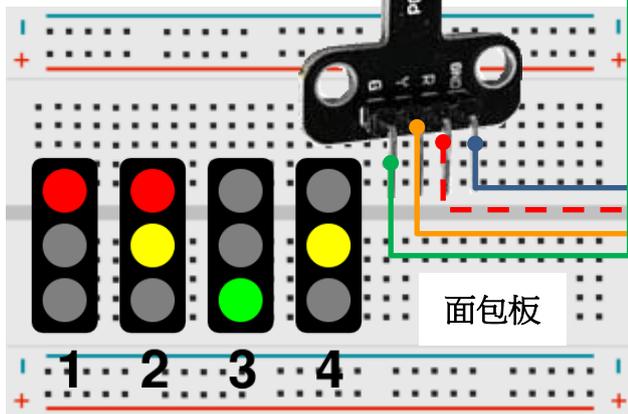
2分 x4

Pins
 引腳



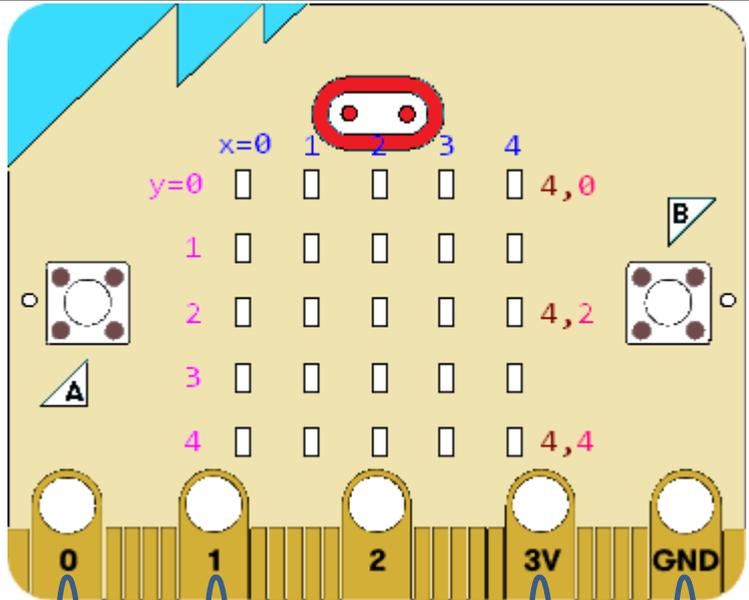
模擬交通燈

綠(0) 黃(0) 紅(1)
 綠(0) 黃(1) 紅(1)
 綠(1) 黃(0) 紅(0)
 綠(0) 黃(1) 紅(0)



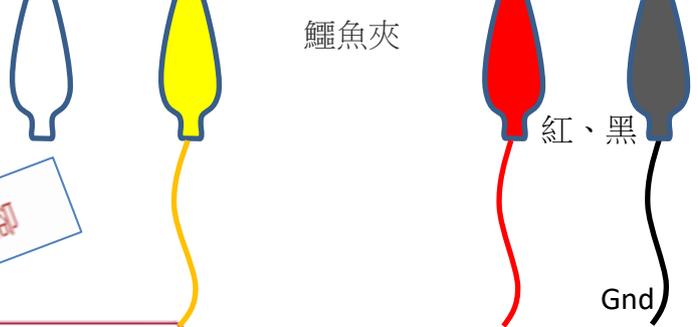
交通燈
<https://mrszeto.net/s2/arduino/mg.htm#traffic>

練習九：紅外線壁障 IR-FC-51



當啟動時
燈光 亮度設為 40

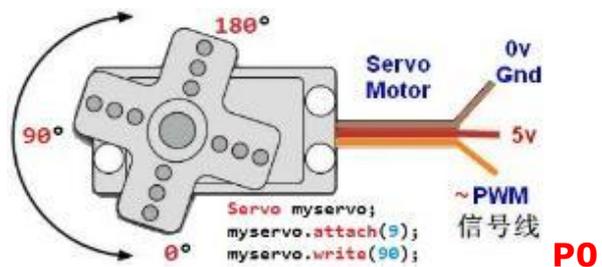
重複無限次
變數 n 設為 數位信號讀取引腳 P1
顯示數字 n



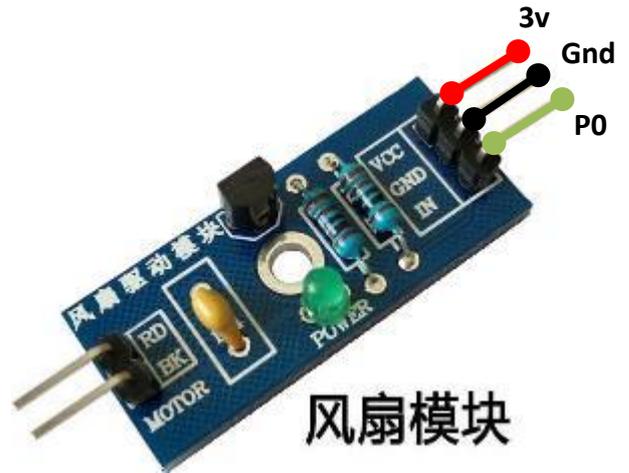
有障礙物時：
n = 0?
n = 1?

如果 n = 0 那麼
伺服寫入 腳位 P0 至 90
否則
伺服寫入 腳位 P0 至 180

伺服馬達 Servo 180°

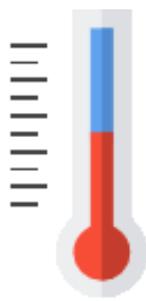
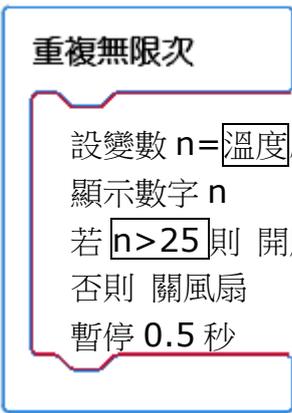
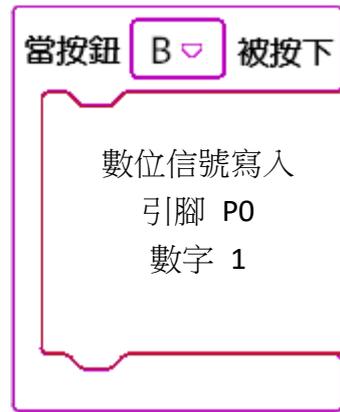
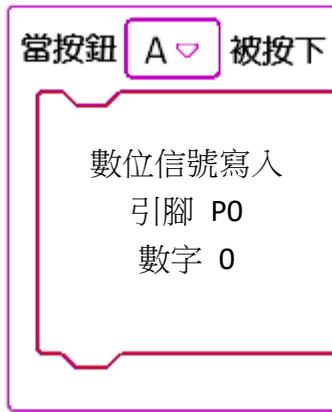
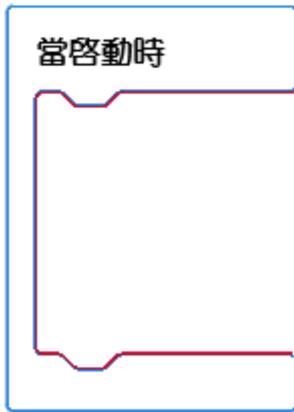


練習十：風扇模塊 Fan

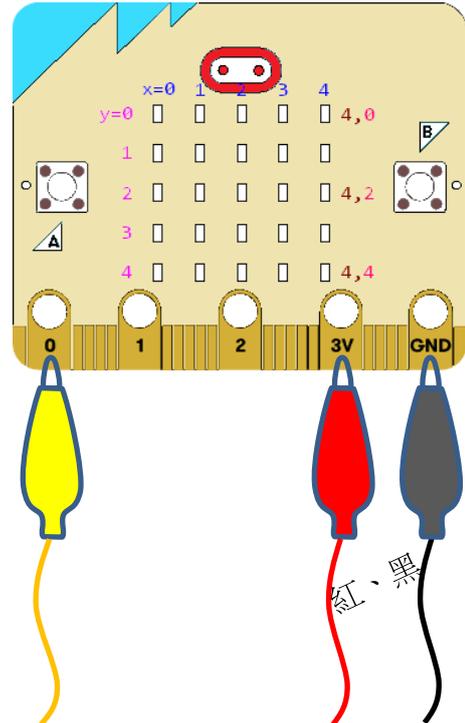


Turn On

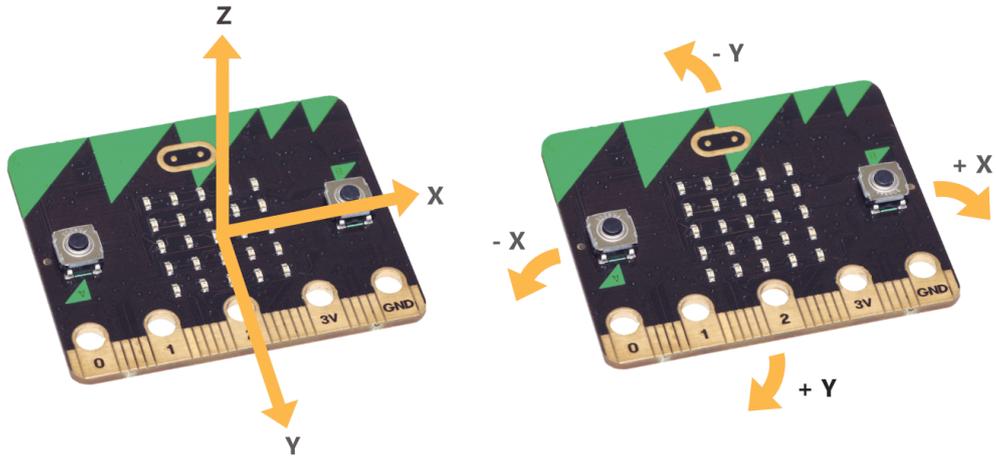
Turn Off



可轉用
光暗度
感測值
0-255

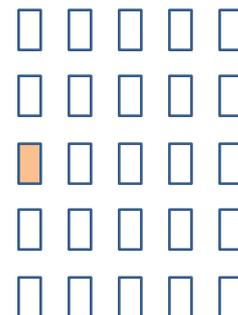


練習十一：加速度計 Accelerometer



1. 向**右**傾側：顯示**→**；
向**左**傾側：顯示**←**。

2. 向**上**傾側：顯示**↑**；向**下**傾側：顯示**↓**。
變數 m 設為 加速度感測器 (mg) : y
如果 $m > 100$ 則 顯示箭頭 **↑**
否則如果 $m < -100$ 則 顯示箭頭 **↓**
否則 顯示 " "



3. 挑戰：隨著 microbit 板偏側，LED 上下、左右滾動。

練習十二：函式 function、音效 Music

選擇語言：英文

The screenshot shows the Microbit code editor interface. On the left is a sidebar with categories: Basic, Input, Music (highlighted), Led, Radio, Loops, Logic, Variables, Math, Advanced, Functions (highlighted), Arrays, Text, Game, and Images. The main workspace contains a function block named 'f(x) 函式 function' with a 'make a function...' button. Below it are four 'call s1' through 'call s4' blocks. A vertical text label '每函式為一句' (Each function is a sentence) points to these blocks. The 'Functions' category in the sidebar is labeled '函式' (Function). The function 's1' is expanded, showing a 'show number 1' block, followed by six 'play tone' blocks with notes Middle C, Middle C, Middle D, Middle C, Middle F, and Middle E, each with a duration of 1/2, 1/4, 1, 1, 1, and 1 beat respectively. A 'pause (ms) 100' block is at the end, labeled '暫停 毫秒' (Pause milliseconds). A musical note icon is in the top right.

The diagram compares two ways to call functions. On the left, an 'on start' block contains four 'call s1', 'call s2', 'call s3', and 'call s4' blocks. On the right, a '當啟動時' (When started) block contains four '呼叫 s1', '呼叫 s2', '呼叫 s3', and '呼叫 s4' blocks.

第一、二、三、四句

s1	s2	s3	s4
C ½	C ½	C ¼	A# ½
C ¼	C ¼	C ¼	A# ¼
D 1	D 1	A# 1	A 1
C 1	C 1	F 1	F 1
F 1	G 1	E 1	G 1
E 1	F 1	D 1	F 1

星球大戰 Star War :

makecode.microbit.org/v0/64323-10091-30877-83077

歌曲集 :

sites.google.com/site/wenyumaker2/02microbit-yan-jiu/31wo-shi-yan-zou-da-shi

The diagram shows a piano keyboard with a 'play tone 262 for 1 beat' block above it. The C# key is highlighted in yellow. Below the keyboard, the notes C# and F#/G# are labeled.

1=C $\frac{4}{4}$ 倫敦鐵橋 London Bridge 英國民歌

5 6 5 4 3 4 5 - 2 3 4 - 3 4 5 -
 London Bridge falling down 2
 倫敦鐵橋 垮下來 垮下來 垮下來

5 6 5 4 3 4 5 - 2 - 5 - 3 1 1 -
 London Bridge my fair lady
 倫敦鐵橋 垮下來 就要 垮下來

		函式 function
London Bridge is falling down	GAGF EFG	Call London_Bridge
falling down falling down	DEF EFG	Call is_falling_down
London Bridge is falling down	GAGF EFG	Call London_Bridge
My fair lady	DGECC	Call My_fair_lady

function London_Bridge

- play tone Middle G for 1 ♡ beat
- play tone Middle A for 1 ♡ beat
- play tone Middle G for 1 ♡ beat
- play tone Middle F for 1 ♡ beat
- play tone Middle E for 1 ♡ beat
- play tone Middle F for 1 ♡ beat
- play tone Middle G for 2 ♡ beat

function is_falling_down

- play tone Middle D for 1 ♡ beat
- play tone Middle E for 1 ♡ beat
- play tone Middle F for 2 ♡ beat
- play tone Middle E for 1 ♡ beat
- play tone Middle F for 1 ♡ beat
- play tone Middle G for 2 ♡ beat

function My_fair_lady

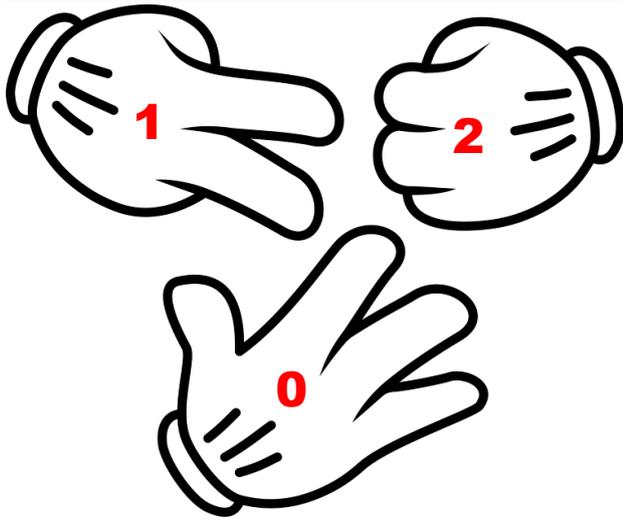
- play tone Middle D for 2 ♡ beat
- play tone Middle G for 2 ♡ beat
- play tone Middle E for 1 ♡ beat
- play tone Middle C for 1 ♡ beat
- play tone Middle C for 2 ♡ beat

音樂盒：重複無限次

打開音樂盒時(光線感測值>60)，便播放音樂(旋律 birthday)。

音效 播放 旋律 birthday ♡ 重複 一次 ♡

練習十三：猜拳---包剪揼 Paper Scissors Rock



晃動：
產生隨機數 n (0,1,2)
顯示數字：n (或圖示)

按 A：
廣播數字 n

按 B：

按 A+B：
清除 (n = -1)

收到廣播數字 receivedNumber：
輸出：n vs receivedNumber
輸出結果：勝 2、和 1、負 0 分

練習十四：雷電測距 Lightning distance calculator

當看見閃電時，按 A

當聽聞雷聲時，按 B

當啓動時

光速 $vL = 3 \times 10^8 \text{ m/s}$
音速 $vS = 340.29 \text{ m/s}$

$$s = \frac{1}{vS} - \frac{1}{vL}$$

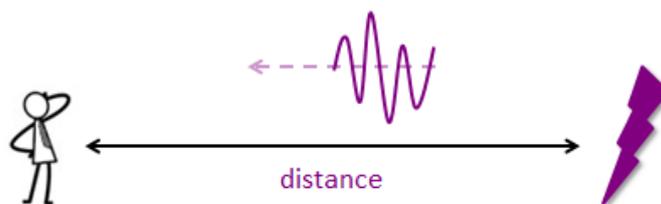
當按鈕 A 被按下

$t0 = \text{運行時間}/1000$
顯示數字 $\text{truncate}(t0)$
暫停 0.1 秒
清空畫面

當按鈕 B 被按下

$t1 = \text{運行時間}/1000$
 $d = \text{truncate}((t1-t0)/s)$
顯示 "D = " d
 $t0 = t1$
暫停 0.1 秒
清空畫面

<https://www.101computing.net/bbc-microbit-lightning-distance-calculator/>



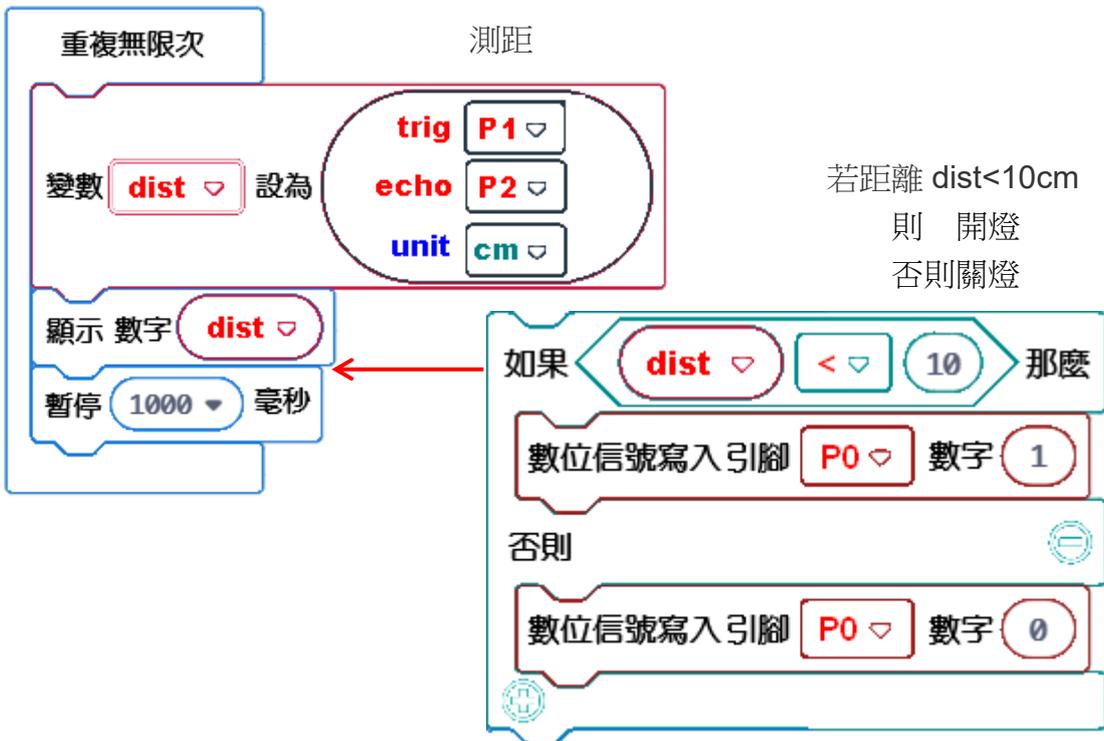
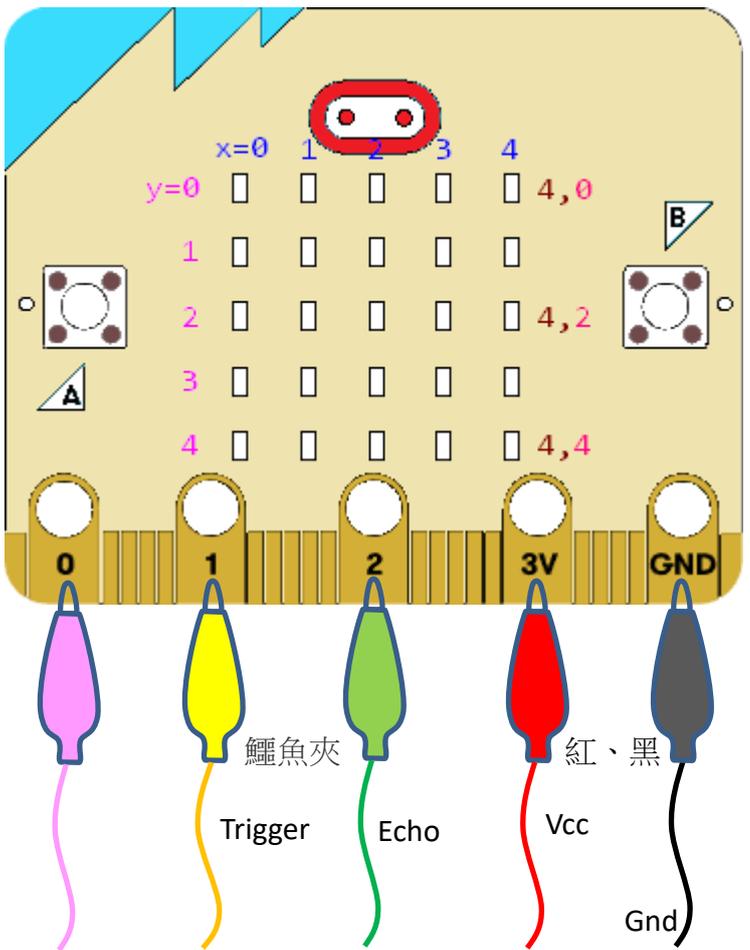
$$\text{速度 } v = \frac{\text{距離 } d}{\text{時間 } t} ; \text{時間 } t = \frac{\text{距離 } d}{\text{速度 } v}$$

<p>當看見閃電時，按 A</p> $vL = \frac{\text{distance}(d)}{t0} \quad \longrightarrow \quad t0 = \frac{d}{vL}$	$t1 - t0 = \frac{d}{vS} - \frac{d}{vL}$
<p>當聽聞雷聲時，按 B</p> $vS = \frac{\text{distance}(d)}{t1} \quad \longrightarrow \quad t1 = \frac{d}{vS}$	$t1 - t0 = d \left(\frac{1}{vS} - \frac{1}{vL} \right)$
$d = \frac{t1 - t0}{s}$	

練習十五：超聲波測距 ultrasonic HC-SR04 (5v)



5v



進度表

週	練習	週	練習(暫定)
12	一：答對加 1 分	18	七：數碼時鐘
13	二：溫度計	19	八：交通燈
14	三：隨機數	20	九：紅外線壁障
	四：猜數字	21	十：風扇模塊
15	五：指南針		十一：加速度計
16	六：顯示方向(廣播)		十二：函式、音效 Music
17	測驗(1)	22	測驗(2)
		23	會聲會影
	十三：猜拳一包剪掙		
	十四：雷電測距		十五：超聲波測距

觀察、記錄 光暗度感測值(光暗度變化)：

建立變數：n1、n2、n3

當啟動時：

顯示目前光線感測值，

把 micro:bit 放在盒子內，蓋上，暫停 6 秒；

n1 = 光線感測值，暫停 2 秒；

n2 = 光線感測值，暫停 2 秒；

n3 = 光線感測值，暫停 2 秒；

當按鈕 A 被按下：

顯示 n1 值，暫停 0.5 秒；

顯示 n2 值，暫停 0.5 秒；

顯示 n3 值，暫停 0.5 秒；

當按鈕 B 被按下：

顯示目前光線感測值，暫停 0.5 秒；

