



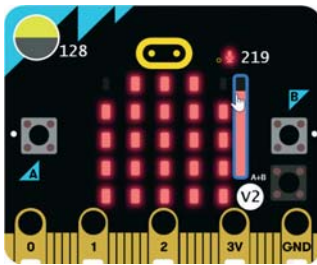
# LED 與聲光共舞 - 聲音與光線感測器



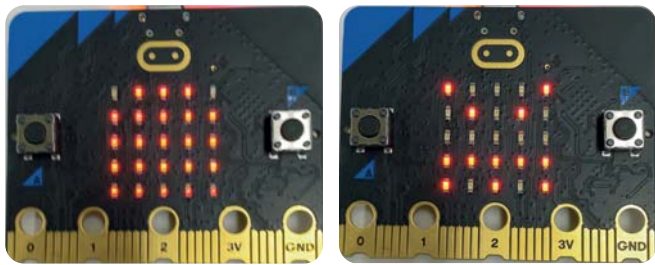
- ★ 1. 理解 micro:bit 主板中聲音與光線感測器的原理。
- ★ 2. 能夠應用聲音與光線控制 LED。
- ★ 3. 能夠理解 LED 與坐標。
- ★ 4. 能夠應用坐標控制每個 LED。
- ★ 5. 能夠應用計數迴圈控制 LED。



模擬器



實體



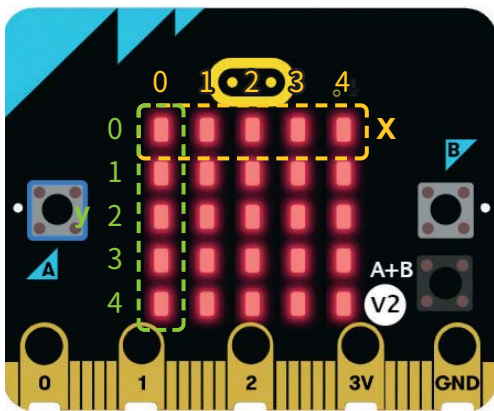


本章將應用 micro:bit 主板的聲音與光線感測器，設定 LED 的亮度與亮燈的數量。利用「燈光」積木，將 LED 依照坐標的位置，依序縱向點亮每一個 LED，再橫向切換每一個 LED 的開關，最後再依據聲音與光線值，以長條圖點亮 LED。

## 7-1 micro:bit LED 坐標與燈光



micro:bit 內建 5×5 LED，每個 LED 以坐標 (x,y) 表示，其中坐標 x 代表橫軸，由左而右分別為 0,1,2,3,4。y 代表縱軸，由上而下分別為 0,1,2,3,4。



LED 坐標 (x,y) 編號

(0,0)	(1,0)	(2,0)	(3,0)	(4,0)
(0,1)	(1,1)	(2,1)	(3,1)	(4,1)
(0,2)	(1,2)	(2,2)	(3,2)	(4,2)
(0,3)	(1,3)	(2,3)	(3,3)	(4,3)
(0,4)	(1,4)	(2,4)	(3,4)	(4,4)

在  燈光 類別積木中，與 5×5 LED 相關的積木功能如下：

功能	個別點亮及亮度	關
積木	<ol style="list-style-type: none"> <li></li> <li></li> </ol>	
說明	<ol style="list-style-type: none"> <li>點亮 LED 螢幕上特定 x,y 位置的 LED。</li> <li>點亮 LED 螢幕上特定 x,y 位置的 LED 並設定亮度值。亮度範圍從 0~255。</li> </ol>	關閉 LED 螢幕上特定 x,y 位置的 LED。



功能	開與關切換	判斷開或關的狀態
積木		
說明	1. 切換 LED 螢幕上特定 x,y 位置的 LED。 2. 如果是開就切換為關。 3. 如果是關就切換為開。	判斷 LED 螢幕上特定 x,y 位置 LED 的開關的狀態。 判斷結果： true (真) : LED 已開。 false(假) : LED 未開。

功能	長條圖點亮	開啟或關閉
積木		(1) (2)
說明	在 LED 螢幕上依據設定的顯示值顯示長條圖。 最大值為長條圖能顯示的最大數值。	(1) 啟動或關閉 LED 螢幕。 true (真) : 點亮 LED 螢幕。 false(假) : 關閉 LED 螢幕。 (2) 停止播放全部動畫。

功能	傳回亮度值	設定亮度值
積木		
說明	傳回目前 LED 的亮度值，傳回值範圍從 0~255。	設定 LED 亮度，亮度範圍從 0~255。0：關閉 LED 不亮；255：最亮。

### 概 念 應 用 一

動手做堆疊積木，請勾選下圖積木程式的執行結果？

- ( ) 1. 未亮點任何 LED
- ( ) 2. 隨機點亮一顆 LED，再關閉。
- ( ) 3. 從 (0,0) 到 (4,4) 依序點亮每個 LED。





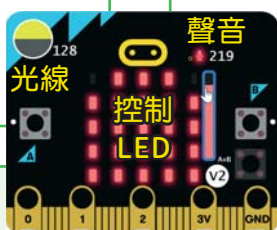
## 7-2 ♥ LED 與聲光共舞專題規劃



本節將應用 micro:bit 主板的聲音與光線感測器，設定 LED 的亮度與亮燈的數量。利用燈光積木，將 LED 依照坐標的位置，依序縱向點亮每一個 LED，再橫向切換每一個 LED 的開關，最後再依據聲音與光線值，以長條圖點亮 LED，聲音愈大聲或光線愈亮時，LED 亮燈數量愈多，同時愈度愈亮，當光線值為 0 時關閉 LED 停止程式執行。

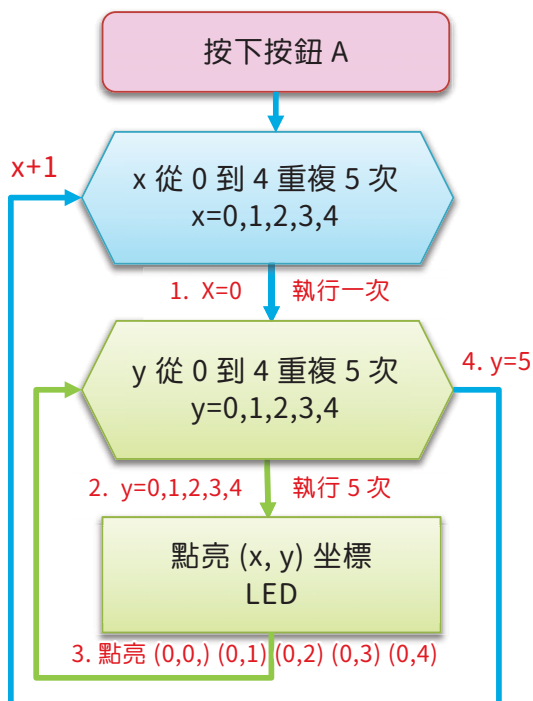
### 一、Planning 專題規劃

<p style="text-align: center;"><b>縱向點亮</b></p> <ol style="list-style-type: none"> <li>1. 設定 LED 的亮度為 255。</li> <li>2. 按下按鈕 A，縱向點亮每一個 LED。</li> </ol>	<p style="text-align: center;"><b>橫向切換開關</b></p> <ol style="list-style-type: none"> <li>1. 設定 LED 的亮度為 255。</li> <li>2. 按下按鈕 B，橫向切換每一個 LED，亮燈切換為關閉、關閉切換為亮燈。</li> </ol>
<p style="text-align: center;"><b>LED 與聲音共舞</b></p> <ol style="list-style-type: none"> <li>1. 同時按下按鈕 A 與 B，將 LED 亮度設為聲音值，依音量控制 LED 亮燈數量與亮度。</li> <li>2. 當光線值為 0，遮住 micro:bit 時，停止重複執行，顯示圖示一。</li> </ol>	<p style="text-align: center;"><b>LED 與光線共舞</b></p> <ol style="list-style-type: none"> <li>1. 按下 logo 觸摸感測器，將 LED 亮度設為光線值，依光線控制 LED 亮燈數量與亮度。</li> <li>2. 當光線值為 0，遮住 micro:bit 時，停止重複執行，顯示圖示二。</li> </ol>
<p style="text-align: center;"><b>我的規劃</b></p> <ol style="list-style-type: none"> <li>1. 按下 _____，光線感測器應用在 _____。</li> <li>2. 按下 _____，聲音感測器應用在 _____。</li> <li>3. LED 點亮方式 _____、LED 關閉方式 _____、LED 切換方式 _____。</li> </ol>	

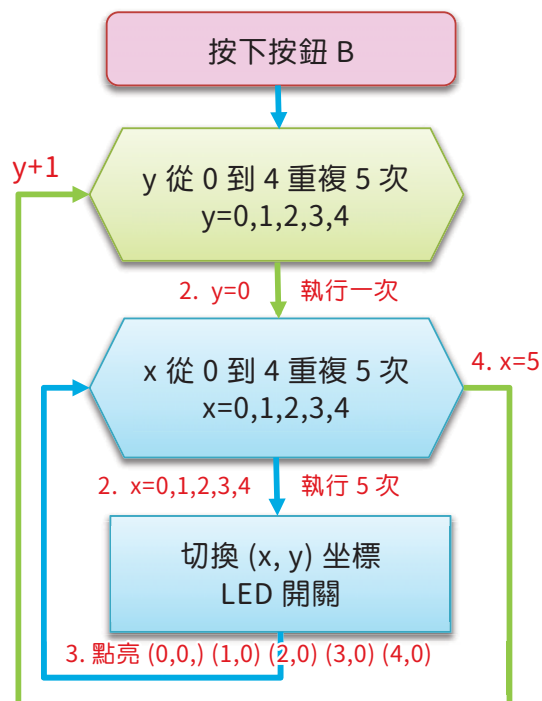




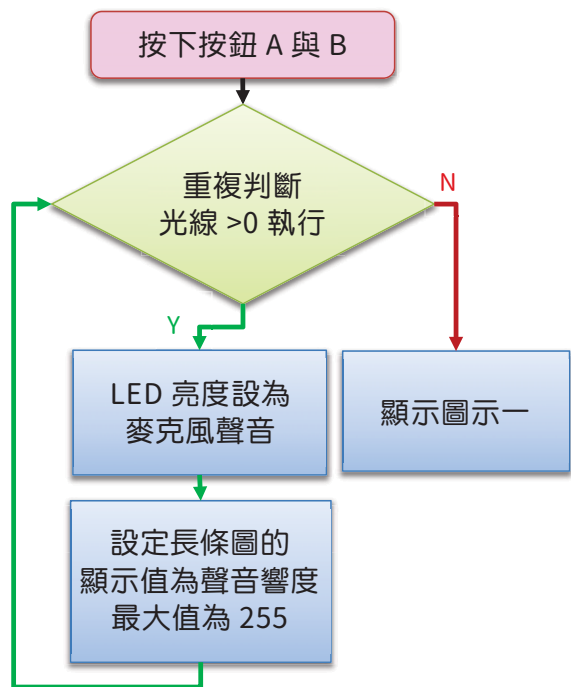
## 二、執行流程



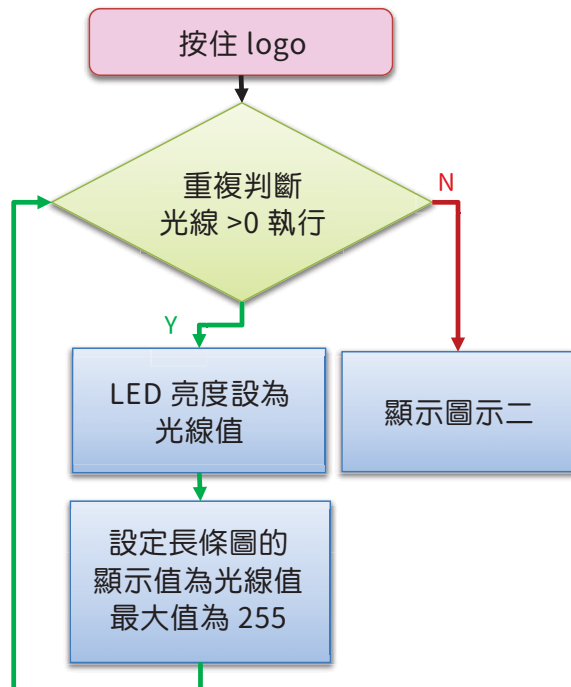
縱向點亮執行流程



橫向切換開關執行流程



LED 與聲音共舞



LED 與光線共舞




## 7-4 ♥ LED 與聲光共舞 Coding



### 一、縱向點亮 LED


按下按鈕 A，設定 LED 亮度為 255，縱向依序點亮每個 LED。



- 1 開啟瀏覽器，輸入網址：<https://makecode.microbit.org/beta>。
- 2 點選【新增專案】，輸入專案名稱【ch7】，選擇編輯的程式類型，再按【創建】。

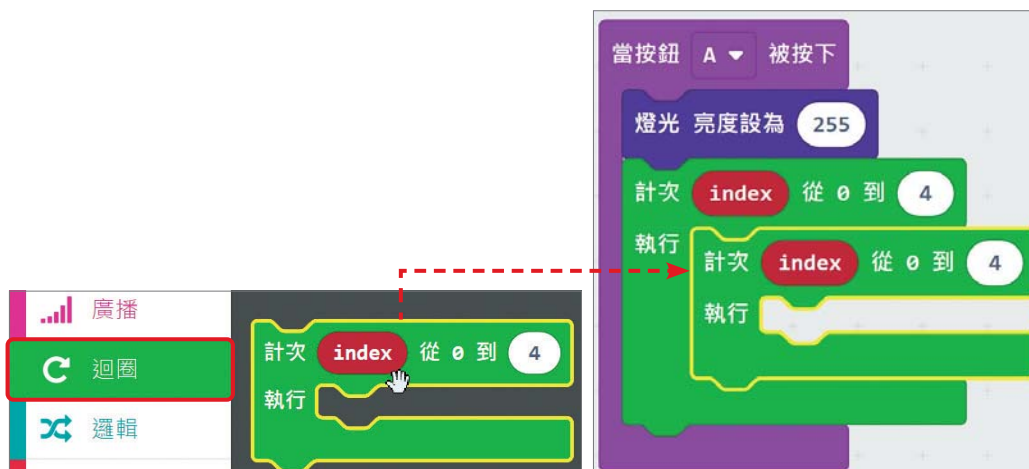
- 3 點按  輸入，拖曳 。

- 4 點按  燈光的【更多】，拖曳 ，設定 LED 亮度為 255。



- 5 點按  變數， 建立一個變數，輸入【x】，再按【確定】，再建立一個變數【y】。

- 6 點按  迴圈，拖曳 2 個 ，分別執行 x, y 坐標的 LED。



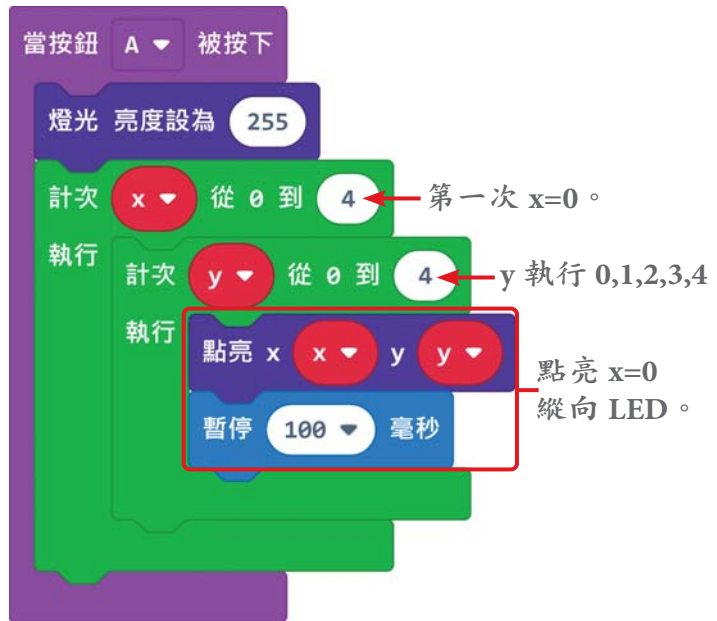


7 點按 **變數**，分別將 **x** 與 **y** 拖曳到 index 位置，縱向坐標為 x，所以計次外層放 x，內層放 y。



8 點按 **燈光** 的 **點亮 x 0 y 0**，再點按 **變數**，分別將 **x** 與 **y** 拖曳到 x 與 y 的位置，分別點亮 x, y 坐標的 LED。

9 點按 **基本**，拖曳 **暫停 100 毫秒**，每隔 0.1 秒，點亮 1 個 LED。



### 實作概念

1. 變數 **x** 與 **y** 的積木放在點亮 x 與 y 的位置，x 與 y 的值才會隨著程式執行變化，依序點亮每個位置的 LED。
2. 程式執行速度非常快，如果沒有暫停 0.1 秒，按下按鈕 A，馬上全亮。新增暫停 0.1 秒能夠觀察程式執行過程。

## 二、模擬器點亮 LED

- 1 點擊 【偵錯模式】與 【慢速模式】，再按下按鈕【A】，檢查模擬器慢速執行程式的流程、顯示 x,y 變數值的變化，再依序縱向點亮每一個 LED。

The screenshot shows the simulation interface with a 4x4 LED matrix. The matrix is currently showing a vertical column of lit LEDs. The interface includes a control panel with buttons for play (1), pause (2), and slow mode (3). A variable monitor shows x: 2 and y: 1 (5). The code blocks are as follows:

- 當按鈕 A 被按下
- 燈光 亮度設為 255
- 計次 x 從 0 到 4
- 執行 計次 y 從 0 到 4
- 執行 點亮 x x y y
- 暫停 100 毫秒 (4)
- 顯示執行流程。

依據 x,y 的值點亮 LED。

## 三、橫向切換 LED 開關

按下按鈕 B，橫向切換每一個 LED，亮燈切換為關閉、關閉切換為亮燈。

- 1 拖曳右圖積木，橫向坐標為 y，計次外層先執行變數 y，內層再執行 x，橫向切換 LED。

The screenshot shows the Scratch code blocks for horizontal switching:

- 當按鈕 B 被按下
- 燈光 亮度設為 255
- 計次 y 從 0 到 4 (第一次 y=0)
- 執行 計次 x 從 0 到 4 (x 執行 0,1,2,3,4)
- 執行 點的狀態切換 x x y y
- 暫停 100 毫秒 (切換 y=0、橫向 LED)



## 實力評量

### 一、選擇題

( ) 1. 如果想設計 micro:bit 偵測光線，應該使用下列哪一個積木？




- (A) 磁力感測值 ( $\mu\text{T}$ ) x ▾ (B) 聲音響度  
(C) 光線感測值 (D) 方位感測值 ( $^{\circ}$ )。

( ) 2. 下列關於右圖積木的敘述何者「有誤」？

- (A) x 與 y 都從 0 開始執行  
(B) x 與 y 每次執行完自動加 1  
(C) 當 x = 0 時，y 從 0 執行到 4  
(D) 程式總共執行 16 次。



( ) 3. 右圖積木點亮 LED 的方向為何？

- (A) 垂直縱向點亮   
(B) 隨機點亮   
(C) 水平橫向點亮   
(D) 以上皆可。



( ) 4. 如果變數 x 與 y 設定為 0 到 4 隨機取數，下列哪一個積木能夠隨機點亮 LED ？

- (A) 燈光 亮度設為 光線感測值  
(B) 點的狀態切換 x x y y  
(C) 不點亮 x x y y  
(D) 點亮 x x y y。

## micro:bit 應用在生活 - 智慧小夜燈

動動腦，如何將 micro:bit 光線感測器應用在生活中？

### 一、Create 動手創造

將 micro:bit 設計智慧小夜燈，當光線值大的時候亮度小、亮燈數少，光線值小的時候亮度大、亮燈數多。

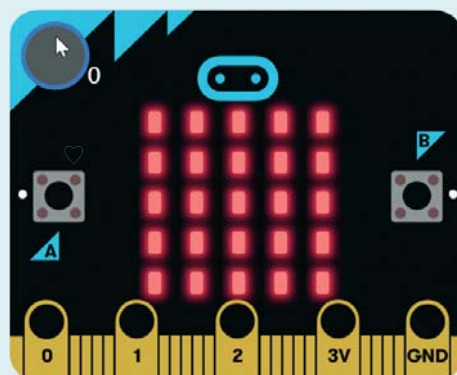
### 二、micro:bit 智慧小夜燈準備元件

(1) micro:bit、(2) Micro USB 連接線、(3) 電池盒與 3V 電池。

### 三、編寫程式 Coding



### 四、模擬器執行與修訂程式



### 五、micro:bit 智慧小夜燈

將程式下載到 micro:bit。再將 micro:bit 放在暗處 LED 亮燈增加、亮度也增加；micro:bit 放在燈光明亮處 LED 亮燈少，亮度暗。

