

A-2 ♥ micro:bit 積木功能表



★ 基本

	<p>在 LED 螢幕上顯示 1 位數字。 如果大於 2 位以上的數字，以跑馬燈方式往左滑動顯示。</p>
	<p>在 LED 螢幕上顯示圖示。 白色：點亮顯示；未亮燈：不顯示。</p>
	<p>在 LED 螢幕上顯示選擇的圖示。 內建預設悲傷或剪刀等 40 種圖示。</p>
	<p>在 LED 螢幕上顯示 1 個文字 (A~Z, 0~9 或符號)。 如果大於 2 個以上的文字，以跑馬燈方式往左滑動顯示。</p>
	<p>關閉 LED 螢幕點亮的所有燈。</p>
	<p>重複執行程式。</p>
	<p>啟動 micro:bit 後，開始執行程式。</p>
	<p>暫停執行程式。(1000 毫秒 = 1 秒)</p>
	<p>在 LED 螢幕上顯示北、東北、東、東南、南、西南、西、西北等八個方向的箭頭。</p>

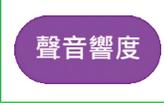


★ 輸入

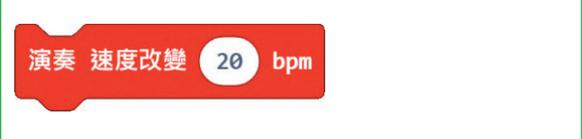
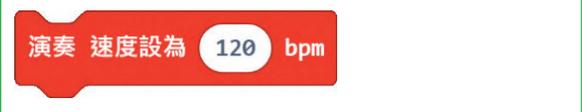
	<p>當按下 micro:bit 按鈕【A】、【B】或同時按下【A】與【B】，開始執行程式。</p>
	<p>當 micro:bit 晃動、上下傾斜、左右傾斜、正面朝上、正面朝下或自由落體掉落時，開始執行程式。</p>
	<p>當 P0、P1 或 P2 引腳與接地（GND）引腳同時被按下時，開始執行程式。</p>
	<p>判斷是否按下 micro:bit 的按鈕【A】、【B】或同時按下【A】與【B】。 傳回布林值：(1) true（真）按下按鈕；(2) false（假）未按下按鈕。</p>
	<p>傳回 micro:bit 加速度感測器左右、前後或上下方向的加速度感測值，感測值範圍從 -1023 ~ 1023。 mg：加速度單位。 (1) x：傳回 micro:bit 左右方向的加速度感測值。 (2) y：傳回 micro:bit 前後方向的加速度感測值。 (3) z：傳回 micro:bit 上下方向的加速度感測值。</p>
	<p>傳回 micro:bit LED 周圍環境的光線值，光線值範圍從 0（最暗）~ 255（最亮）。</p>
	<p>傳回 micro:bit 指南針（Compass）的方位感測值。 方位值範圍：(1) 0 度：北（North）；(2) 90 度：東（East）；(3) 180 度：南（South）；(4) 270 度：西（West）。</p>
	<p>傳回 micro:bit 攝氏（Celsius）溫度的感測值。 溫度值範圍：-5°C（最低溫）~ 50°C（最高溫）。</p>
	<p>判斷 micro:bit 的姿勢為晃動、傾斜或自由掉落等。 傳回布林值：(1) true（真）已晃動；(2) false（假）未晃動。</p>

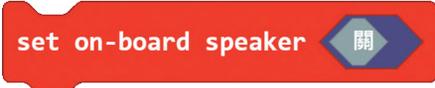
	<p>傳回 micro:bit 在不同方向的傾斜感測值。</p> <p>(1) pitch：向上或向下傾斜，感測值範圍從 -180 ~ 180 度。</p> <p>(2) roll：向左或向右傾斜，感測值範圍從 -180 ~ 180 度。</p>
	<p>傳回 micro:bit x，y 或 z 軸方向的磁力感測值。</p> <p>(1) x：測量左右方向；(2) y：測量前後方向；(3) z：測量上下方向。</p>
	<p>傳回程式從開始執行到目前為止的總計執行時間。時間的單位 ms：毫秒；1000 毫秒 = 1 秒。</p>
	<p>傳回程式從開始執行到目前為止的總計執行時間。時間的單位 micros：微秒；1000000 毫秒 = 1 秒。</p>
	<p>校正指南針。</p>
	<p>當 P0，P1 或 P2 引腳同時與接地 (GND) 被按下、再放開時，開始執行程式。</p>
	<p>設定 micro:bit 的加速度。加速度的重力範圍從 1g (最小值) ~ 8g (最大值)。</p>
<p>micro:bit (V2) 新增積木</p>	
	<p>當偵測到 micro:bit 的麥克風聲音或麥克風靜音時，開始執行程式。</p>
	<p>當按住、碰觸、鬆開或長按 logo 時，開始執行程式。</p>
	<p>判斷 micro:bit 的 logo 是否被按下。</p> <p>傳回布林值：(1) true (真) 按下 logo；(2) false (假) 未按下 logo。</p>



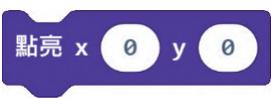
	<p>傳回 micro:bit 麥克風的音量值。音量值範圍從 0 (靜音) ~ 255 (最大音量)。</p>
	<p>設定麥克風要偵測的音量值。音量值範圍從 0 (靜音) ~ 255 (最大音量)。</p>

★ 音效

	<p>自訂演奏旋律。</p>
	<p>播放中音 C (Do) 音階，1 拍。 (1) 音階範圍：低音 C (Do) ~ 高音 C (Do)。(2) 節拍：1/16 拍 ~ 4 拍。</p>
	<p>連續播放中音 C (Do) 音階。</p>
	<p>演奏休息 1 拍。節拍範圍從 1/16 拍 ~ 4 拍。</p>
	<p>傳回演奏的音階。</p>
	<p>設定輸出的音量值。</p>
	<p>停止播放所有的音效。</p>
	<p>改變演奏音階的速度。 (1) 正數：演奏速度變快；(2) 負數：演奏速度變慢。</p>
	<p>設定演奏音階的速度。</p>
	<p>傳回節拍的演奏時間。 時間的單位 ms：毫秒；1000 毫秒 = 1 秒。</p>

	傳回目前演奏音階的節奏。 節奏單位：bpm 每分鐘的節拍數。
	播放內建旋律。 (1) 旋律種類：內建生日快樂歌等 20 種。 (2) 播放次數：一次或無限次。
	停止全部（背景或前景）正在演奏的旋律。
	當前景或背景開始（或重複、結束）演奏旋律時，啟動程式執行。
micro:bit (V2) 新增積木	
	播放 giggle 音效，直到結束。內建笑聲、快樂等十種音效。
	依據程式執行速度播放 giggle 音效。
	設定 micro:bit 主板上的喇叭為開或關。

★ 燈光

	點亮 LED 螢幕上特定 x, y 位置的 LED。 (1) x：橫軸，由左而右分別為 0, 1, 2, 3, 4。 (2) y：縱軸，由上而下分別為 0, 1, 2, 3, 4。
	切換 LED 螢幕上特定 x、y 坐標的 LED。 如果是開就切換為關；如果是關就切換為開。
	關閉 LED 螢幕上特定 x、y 坐標的 LED。



	<p>判斷 LED 螢幕上特定 x、y 坐標 LED 的開關狀態。傳回布林值：(1) true (真) 點亮；(2) false (假) 未點亮。</p>
	<p>在 LED 螢幕上依據設定的顯示值顯示長條圖。 (1) 最大值為長條圖能顯示的最大數值。 (2) 顯示值與最大值範圍：0~1024。</p>
	<p>點亮 LED 螢幕上特定 x、y 位置的 LED 並設定亮度，亮度範圍從 0 ~ 255。</p>
	<p>傳回特定 x、y 坐標 LED 的亮度值。</p>
	<p>傳回目前 LED 的亮度。傳回值的範圍：0~ 255。</p>
	<p>設定 LED 亮度，範圍從 0 (不亮) ~ 255 (全亮)。</p>
	<p>啟動或關閉 LED 螢幕。 (1) true (真)：點亮 LED 螢幕。 (2) false (假)：關閉 LED 螢幕。</p>
	<p>停止播放全部動畫。</p>
	<p>設定 LED 顯示模式為黑白或灰階。</p>

★ 廣播

	<p>設定 micro: bit 廣播的群組 id，相同群組才能接收或發送廣播。群組範圍：0~255。</p>
	<p>廣播發送數字到相同群組的 micro: bit。廣播傳送的數字暫存在 receivedNumber 變數中。</p>
	<p>廣播發送一對文字與數字到相同群組的 micro: bit。廣播發送的文字暫存在 name 變數中，文字長度最多 12 字元；發送的數字暫存在 value 變數中。</p>

	<p>廣播發送文字到相同群組的 micro: bit，文字長度最多 19 字元。</p> <p>廣播傳送的文字暫存在 <code>receivedNumber</code> 變數中。</p>
	<p>接收相同群組 micro: bit 發送的文字廣播。</p>
	<p>接收相同群組 micro: bit 發送的一對文字與數字廣播。</p>
	<p>接收相同群組 micro: bit 發送的數字廣播。</p>
	<p>傳回收到廣播的訊息時，廣播的訊號強度、時間與序號。</p> <p>(1) 訊號強度範圍：-128（最弱）～ -42（最強）。</p> <p>(2) 時間：廣播訊息發送的時間。</p> <p>(3) 序號：發送廣播的序號。</p>
	<p>設定 micro:bit 廣播訊號的強度。</p> <p>強度範圍：0（最弱）～7（最強約 70 公尺）。</p>
	<p>在廣播的訊息封包中寫入 micro: bit 裝置的序號。</p> <p>設定值：(1) true（真）廣播夾帶序號；(2) false（假）廣播不夾帶序號。</p>
	<p>設定廣播發送與接收的頻道，預設值為 7。</p> <p>頻道範圍：0 ～ 83。</p>
	<p>設定執行廣播的事件，例如：</p> <p>按下按鈕 A（MICROBIT_ID_BUTTON_A），發送廣播值（MICROBIT_EVT_ANY）。</p>



★ 迴圈

	<p>重複執行迴圈內程式 4 次。</p>
	<p>當條件為「true (真)」時，重複執行迴圈內的程式。</p>
	<p>將 index 變數從 0 開始計次，依序為 0，1，2，3，4，執行 5 次迴圈內的程式。</p>
	<p>從 list (變數) 中取得 value (值)，依照 value (值) 重複執行迴圈內的程式 value 次。</p>
	<p>跳離迴圈，繼續執行程式。</p>
	<p>重新執行迴圈內的程式。</p>

★ 邏輯

	<p>「如果」條件為「true (真)」，執行「那麼」內層程式。 (1) true：條件為「真」時，執行程式；(2) false：條件為「假」時，執行「如果 - 那麼」下一行程式。</p>
	<p>「如果」條件為「true (真)」，執行「那麼」內層程式， 「如果」條件為「false (假)」，執行「否則」內層程式。</p>

	<p>比較左右兩數的關係是否相等 (=)、不等於 (≠)、小於 (<)、小於等於 (≤)、大於 (>) 或大於等於 (≥)。 傳回布林值：(1) true (真) 兩數相等；(2) false (假) 兩數不相等。</p>
	<p>比較左右兩邊的文字是否相等 (=)。比較時，以文字的 ASCII 碼進行比較，例如「A」的 ASCII 碼為 65，「a」的 ASCII 碼為 97。 傳回布林值：(1) true (真) 兩邊文字相等；(2) false (假) 兩文字不相等。(註：ASCII 碼請參閱附錄 A-1)</p>
	<p>邏輯布林運算，判斷「左布林運算結果」與「右布林運算結果」是否同時為 true (真)。 傳回布林值：(1) true (真) 左布林運算結果為 true，而且右布林運算結果為 true；(2) false (假) 左與右布林運算結果沒有同時為 true。</p>
	<p>邏輯布林運算，判斷「左布林運算結果」與「右布林運算結果」其中一個為 true (真)。 傳回布林值：(1) true (真) 左布林運算結果為 true，或者右布林運算結果為 true；(2) false (假) 左與右布林運算結果同時為 false。</p>
	<p>邏輯布林運算，將布林運算結果為 true (真) 改為 false (假)，將布林運算結果 false (假) 改為 true (真)。</p>
	<p>布林值為 true (真)。</p>
	<p>布林值為 false (假)。</p>

★ 變數

<p>建立一個變數</p>	<p>建立一個變數，變數名稱可以是中文、英文或數字。</p>
	<p>傳回變數的值。</p>
	<p>設定變數的值，變數值可以是數字或 ASCII 碼的英文字或符號。(註：ASCII 碼請參閱附錄 A-1)</p>



	改變變數的值，改變的值為數字。 (1) 正數：增加；(2) 負數：減少。
--	---

★ 數學

	計算左與右兩數相加。
	計算左與右兩數相減。
	計算左與右兩數相乘。
	計算左與右兩數相除。
	0~9 數字
	計算左除以右的餘數。
	比較左，右兩數的最小值。
	比較左，右兩數的最大值。
	計算絕對值。
	計算平方根或三角函數等。
	計算四捨五入或無條件進位等。
	在第一個數 (0) 到第二個數 (10) 之間隨機選一個數。
	限制第 1 個數要介於最低數與最高數之間。

對應 0 從低 0 到高 1023 至低 0 到高 4

傳回第 1 個數的對應值。對應的方式：將第一組從低到高 (0~1023) 的數字範圍轉換為第二組從低到高 (0~4)。

隨機取布林值 隨機產生一個布林值，真 (true) 或假 (false)。

✓ 進階

★ f() 函式

建立一個函式 建立一個函式。

★ 陣列

變數 list 設為 陣列 0 1 - +	建立數字陣列 (list)。 - : 減少陣列的個數 ; + : 增加陣列的個數。
變數 text list 設為 陣列 "a" "b" "c" - +	建立文字陣列 (test list)。
空陣列 +	自訂陣列。
list 的長度	傳回陣列的長度，總共有幾筆資料項。
取得 list 的項目值 索引值為 0	傳回陣列中第 0 個索引的值。
list get and remove value at 0	傳回陣列第 0 個索引的值，並刪除第 0 個索引的值。
取得並移除最末項 list	傳回陣列最後一個索引的值，並刪除最後一個索引的值。
取得第一個值自 list	傳回陣列中第 0 個索引的值，並刪除第 0 個索引的值。



	<p>將陣列第 0 個索引的值設為 (數字或文字)。</p>
	<p>在陣列的最後一個資料項，新增一個值 (數字或文字)。</p>
	<p>刪除陣列最後一個資料項。</p>
	<p>刪除陣列第一個資料項。</p>
	<p>在陣列的最前面位置插入一個值 (數字或文字)，並傳回陣列長度。</p>
	<p>在陣列的最前面位置插入一個值 (數字或文字)。</p>
	<p>在陣列的第 0 個索引，插入一個值 (數字或文字)。</p>
	<p>刪除陣列中第 0 個索引位置的資料值。</p>
	<p>在陣列中搜尋資料 的索引值。</p>
	<p>將陣列中的資料項反向排列，第 0 個索引位置的資料值排到最後一個索引位置。</p>

★ 文字

	<p>文字。micro:bit 僅支援 ASCII 碼從 32 (空白) ~126 的文字、數字與符號。</p>
	<p>傳回文字的長度，總共有幾個字元。</p>
	<p>合併字串。例如："Hello" 與 "World" 組合成 "HelloWorld"</p>

<p>字串剖析 文字 "123" 轉成數字</p>	<p>將 0 ~ 9 文字轉成數字。</p>
<p>字串拆分 "this" 分隔符號 ""</p>	<p>使用分隔符號 (") 將長字串 ("this") 拆解成短字串。</p>
<p>"this" 裡包含文字 "" ?</p>	<p>判斷 "this" 字串中是否包含文字 " "。 傳回布林值：(1) true (真) 包含文字 " "；(2) false (假) 不包含文字。</p>
<p>取得 "this" 裡 文字 "" 的索引值</p>	<p>從長字串 "this" 中取得特定的文字 " " 的索引值，字串 "this" 第 1 個字的位置索引值從 0 開始。</p>
<p>"this" 為空值？</p>	<p>判斷 "this" 字串中是否為空字串 (" ")。 傳回布林值：(1) true (真) " " 為空字串；(2) false (假) " " 不是空字串，內含文字。</p>
<p>字串截取 字串為 "this" 索引值為 0 長度為 10</p> <p>在字串 ("this") 中，從第 0 個索引值 (第 1 個字) 開始，取 10 個字。</p>	
<p>字串比較 "this" 與 ""</p>	<p>傳回前後兩個字串比較的結果。 比較方法：前、後兩個字串從第一個字開始，依照 ASCII 內碼逐字比較。傳回值： (1) -1：前面字串小於後面字。 (2) 1：前字串大於後面字串。 (3) 0：兩個字串相同。</p>
<p>字串取字 字串為 "this" 索引值為 0</p>	<p>從字串 ("this") 中取索引值 0 的字 (第 1 個字)。</p>
<p>轉換 0 成文字型別</p>	<p>將 0~9 數字轉換成文字。</p>
<p>字集取字 代碼為 0</p>	<p>將 ASCII 代碼轉換成文字，例如：ASCII 代碼 65 的文字為 A。</p>



★ 遊戲

	<p>在 LED 螢幕上特定 x, y 位置建立新的 LED 作為角色。</p> <p>(1) x: 橫軸, 由左而右分別為 0, 1, 2, 3, 4。</p> <p>(2) y: 縱軸, 由上而下分別為 0, 1, 2, 3, 4。</p>
	<p>刪除角色。</p>
	<p>判斷角色是否已刪除。</p> <p>傳回布林值: (1) true (真) 角色已刪除; (2) false (假) 角色未刪除。</p>
	<p>角色移動 1 點, 程式預設往右。</p> <p>正數: 往右移動; 負數: 往左移動。</p>
	<p>角色往左或往右旋轉 45 度。</p>
	<p>角色的 x, y 坐標或方向、亮度與閃爍度會隨著程式執行而改變 1。</p>
	<p>角色的 x, y 坐標或方向、亮度與閃爍度固定為 0。</p>
	<p>傳回角色目前的 x, y 位置、方向、亮度或閃爍。</p>
	<p>判斷角色是否碰到另一個角色。</p> <p>傳回布林值: (1) true (真): 兩個角色碰到; (2) false (假): 兩個角色未碰到。</p>
	<p>判斷角色是否碰到邊緣。</p> <p>傳回布林值: (1) true: 碰到邊緣; (2) false: 未碰到邊緣。</p>
	<p>如果角色碰到邊緣, 自動反彈。</p>

生命減少 0	減少生命值。
生命增加 0	增加生命值。
生命設為 0	設定生命值。
得分設為 0	設定得分。
得分改變 1	將得分改變 1。 正數：加分；負數：減分。
開始倒數 (ms) 10000	倒數計時，時間到時會自動顯示「GAME OVER 及 SCORE 分數」跑馬燈。(10000 ms = 10 秒)
得分	傳回得分值。
遊戲結束	結束遊戲並顯示得分。
遊戲已經結束	判斷遊戲是否已經結束。 傳回布林值：(1) true：遊戲已結束；(2) false：遊戲未結束。
遊戲暫時停止	判斷遊戲是否暫時停止。 傳回布林值：(1) true：遊戲暫時停止；(2) false：遊戲未暫時停止。
遊戲正在運行	判斷遊戲是否正在進行。 傳回布林值：(1) true：遊戲正在運作；(2) false：遊戲已停止或暫停。
遊戲繼續	重置遊戲。
遊戲暫停	暫停遊戲。



★ 圖像

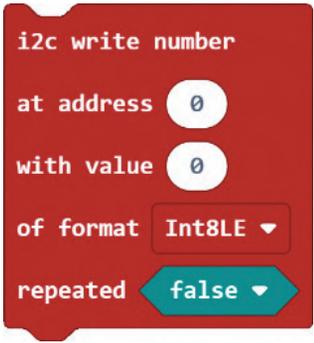
	<p>在 LED 螢幕左右移動圖像 0 點。 正數：往左移動；負數往右移動。</p>
	<p>捲動圖像。 (1) 1：移動點數；(2) 正數：由右往左捲動；負數：由左往右捲動；(3) 200(毫秒)：移動時間。</p>
	<p>創建圖像。</p>
	<p>創建大型圖像。</p>
	<p>創建箭頭圖像。</p>
	<p>創建內建愛心等圖像。</p>
	<p>創建東、西、南、北等八方位的箭頭圖像。</p>

★ 引腳

	<p>從 micro:bit 的 P0~P 1 6 引腳讀取數位信號值。 讀取值：(1) 1：引腳已連接（或開）；(2) 0：引腳未連接（或關）。</p>
	<p>將 0 或 1 信號值寫入 micro:bit 的 P0~P16 引腳。</p>
	<p>從 micro:bit 的 P0~P16 引腳讀取類比信號值。 讀取值：0~1023。</p>

	<p>將 0~1023 信號值寫入 micro:bit 的 P0~P16 引腳。</p>
	<p>傳回第 1 個數的對應值。對應的方式：將第一組從低到高 (0~1023) 的數字範圍轉換為第二組低到高 (0~4)</p>
	<p>設定 P0~P16 引腳類比信號的週期為 20000 μs (μs：微秒；20000 微秒 = 0.02 秒)。</p>
	<p>設定伺服馬達為 P0~P16 引腳與旋轉角度。旋轉角度範圍：0 度 ~180 度。</p>
	<p>設定伺服馬達為 P0~P16 引腳、類比信號輸出，並設定脈衝為 1500 μs。</p>
	<p>設定播放音效或音階的引腳為 P0~P16。</p>
	<p>當 P0~P16 引腳設定為高或低脈衝時，啟動程式執行。</p>
	<p>傳回脈衝的持續時間。</p>
	<p>傳回 P0~P16 引腳設定為高或低脈衝時的脈衝持續時間。</p>
 <p>傳回 7 位元 i2c 位址。</p>	



 <p>i2c write number at address 0 with value 0 of format Int8LE repeated false</p>	<p>將特定值寫入 i2c 的 7 位元位址中。</p>
 <p>spi 寫入 0</p>	<p>將數字寫入 spi，並傳回 spi 反應值。</p>
 <p>spi 頻率設為 1000000</p>	<p>設定 spi 頻率。</p>
 <p>引腳 P0 設為引發 邊緣 事件</p>	<p>設定 P0~P16 引腳用來傳送事件。傳送事件包括：邊緣、脈衝、碰到或無。</p>
 <p>類比音高 0 持續 (ms) 0</p>	<p>發射 PWM 信號到 P0。將 P0 引腳設定為類比引腳。</p>
 <p>spi 格式 位元 8 模式 3</p>	<p>設定 spi 位元及模式。</p>
 <p>引腳 P0 電阻設為 上</p>	<p>設定 P0~P16 引腳是否被按下。</p>
 <p>引腳 P0 設為播放類比音高</p>	<p>設定 P0~P16 引腳為類比，以播放音調。</p>
 <p>spi 設定腳位 MOSI P0 MISO P0 SCK P0</p>	<p>設定 spi 的 MIOS，MIOS 引腳與 SCK 引腳。</p>
<p>micro:bit (V2) 新增積木</p>	
 <p>set P0 to touch mode capacitive</p>	<p>設定 P0~P2 或 logo 的觸控模式為電容 (capacitive 或電阻 (resistive))。</p>

★ 序列

<p>序列 寫入一行文字 " "</p>	<p>寫入一行文字到序列埠並換行。</p>
<p>序列 寫入數字 0</p>	<p>寫入數字到序列埠。</p>
<p>序列 寫入值 "x" = 0</p>	<p>寫入一對文字與數字到序列埠。</p>
<p>序列 寫入文字 " "</p>	<p>寫入一行文字到序列埠。</p>
<p>序列 寫入數字陣列 陣列 0 1 - +</p>	<p>寫入數字陣列到序列埠。</p>
<p>序列 讀取一行字串</p>	<p>從序列埠中讀取一行文字。</p>
<p>序列 讀取直到遇到 new line ()</p>	<p>從序列埠中讀取文字，直到換行才停止。</p>
<p>序列 當數據中收到 new line ()</p>	<p>當序列埠收到換行符號時，啟動程式執行。</p>
<p>序列 讀取文字</p>	<p>從序列埠中讀取文字。</p>
<p>序列重新導向至 TX P0 RX P1 傳輸速率 115200</p>	<p>自訂序列埠傳送及接收的引腳及連接速率。 (1) TX：傳送資料的引腳。 (2) RX：接收資料的引腳。 (3) 連接速率：從 300~115200。</p>
<p>序列 重新導向至 USB</p>	<p>使用 USB 連線以輸出或輸入序列埠資料。</p>



	設定序列埠傳送資料緩衝區的大小為 32 位元組。
	設定序列埠接收資料緩衝區的大小為 32 位元組。
	寫入一個緩衝到序列埠。
	從緩衝序列埠讀取資料。
	設定序列埠寫入填充線段的長度。
	設定序列埠為串列傳輸的頻率。

★ 控制

	停止程式執行並等待來源為 0 的事件。
	在背景執行程式。
	傳回 micro:bit 從啟動到目前為止的時間，時間單位為毫秒。
	重新啟動 micro:bit。
	暫停信號輸出 4 微秒。
	在事件匯流排中觸發一個事件，例如事件來源為按下按鈕 A，數值為任意。

當事件發生 來源為 MICROBIT_ID_BUTTON_A ▾ 數值為 MICROBIT_EVT_ANY ▾

當事件匯流排的事件被觸發時，啟動程式執行。

事件時間戳記	傳回匯流排中最後一個事件執行的時間。
事件結果	傳回匯流排中最後一個事件的值。
MICROBIT_EVT_ANY ▾	傳回 micro:bit 事件的數值。
MICROBIT_ID_BUTTON_A ▾	傳回 micro:bit 裝置來源的值。
裝置名稱	傳回序列埠連接引腳的裝置名稱。
裝置序號	傳回裝置的序號。

★ + 擴展

+ 添加套件	新增 micro:bit 裝置、遠端遙控、藍牙裝置或 LED 燈環等相關的程式模組積木。
--------	--



A-3 手機設計 micro:bit 程式



本單元將利用手機跟 micro:bit 藍牙配對、利用手機設計程式、儲存程式再下載到 micro:bit 執行程式結果。

一、連接並下載 APP

將手機跟 micro:bit 藍牙配對時，首先將 micro:bit 連接電源並在手機下載 APP。

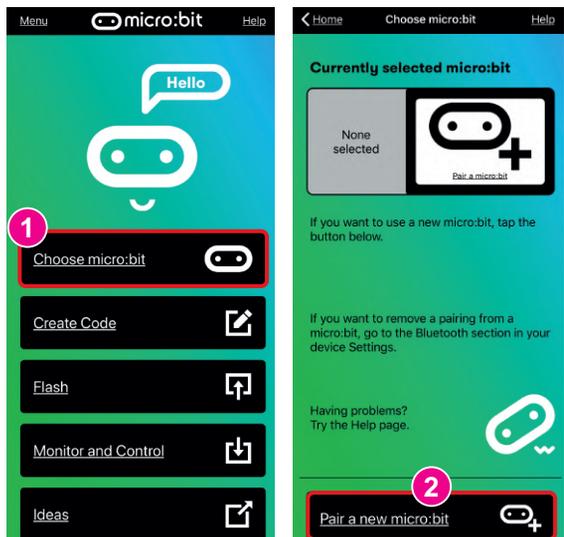


- 1 將 micro:bit 連接電池或電腦。
- 2 【手機】play 商店或 APP Store 下載 micro:bit APP。
- 3 【手機】開啟藍牙、並開啟手機 APP 。

二、藍牙配對

藍牙配對前，請先確認手機的藍牙已開啟。配對方式如下：

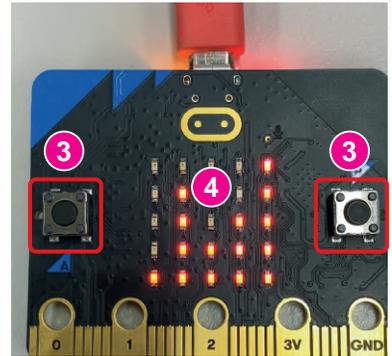
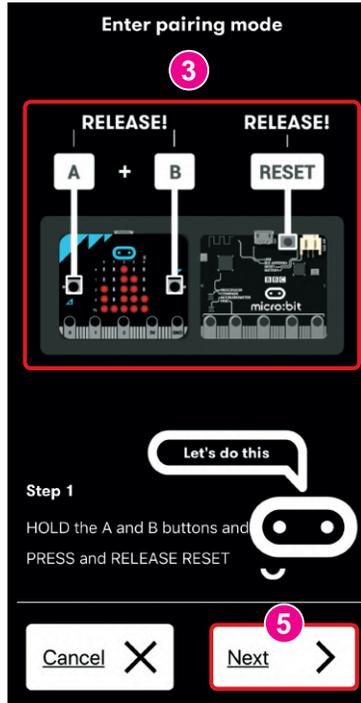
- 1 選擇 micro:bit。
- 2 配對 micro:bit。



3 如手機操作畫面，同時按住 micro:bit 的按鈕 A 與 B 與 Reset。

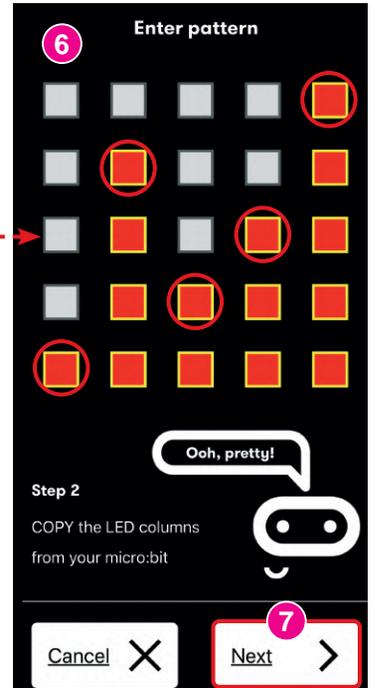
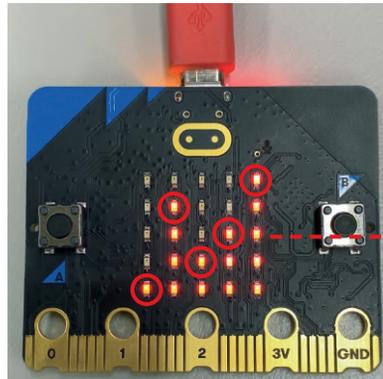
4 放開 Reset(繼續按住按鈕 A 與 B)，直到 micro:bit 顯示圖案，再放開按鈕 A 與 B。

5 再按手機的【Next】。



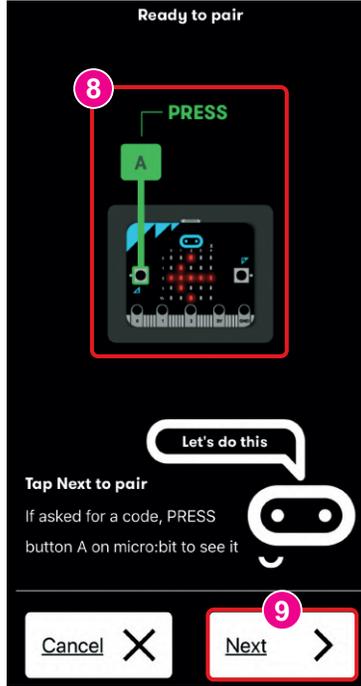
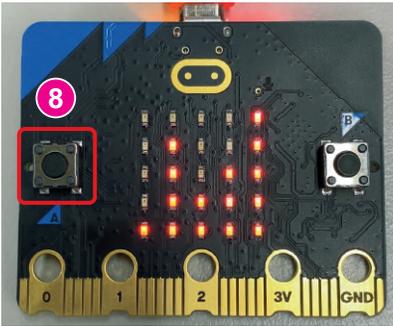
6 依據 micro:bit 圖形，在手機點按跟 micro:bit 相同圖形。

7 圖形正確，按【Next】。

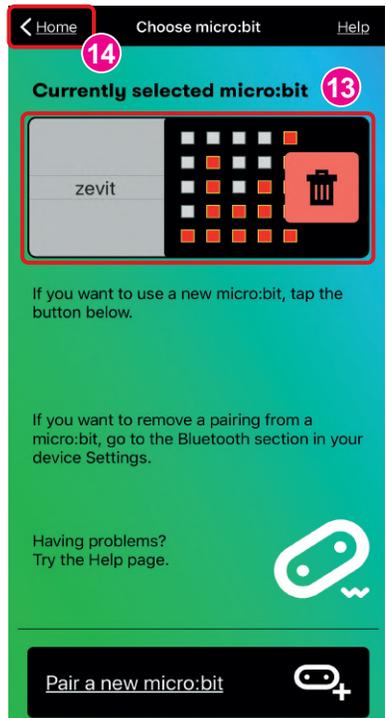
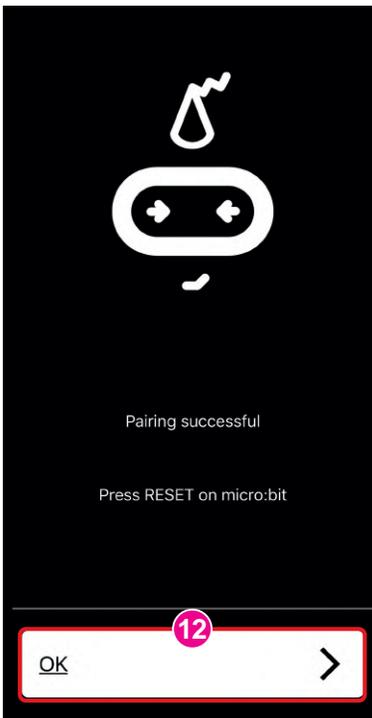
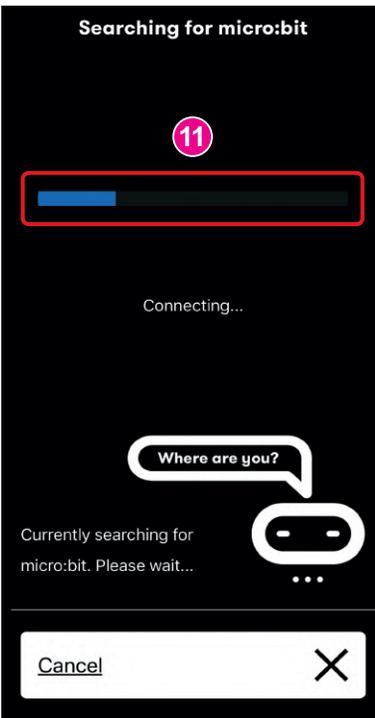




- 8 如手機畫面，按下 micro:bit 按鈕 A。
- 9 再按手機【Next】。
- 10 點按【配對】，進行手機與 micro:bit 藍牙配對。



- 11 開始連接藍牙。
- 12 配對成功，點按【OK】。
- 13 顯示配對的 micro:bit。
- 14 按【Home】回到選單，開始設計程式。

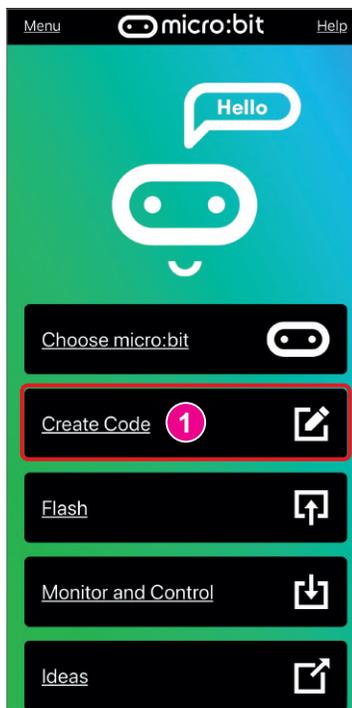


三、手機設計程式

利用手機設計程式、儲存程式，並下載到 micro:bit 執行程式結果。手機 APP 設計程式的視窗與電腦版相同。

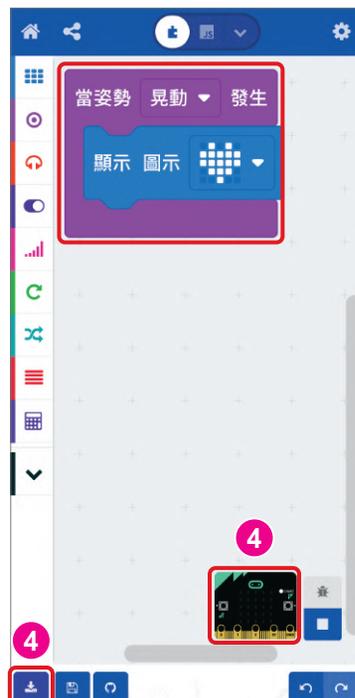
1 點按【創建程式】。

2 點按【新增專案】。



3 輸入【專案名稱】，再點擊【創建】。

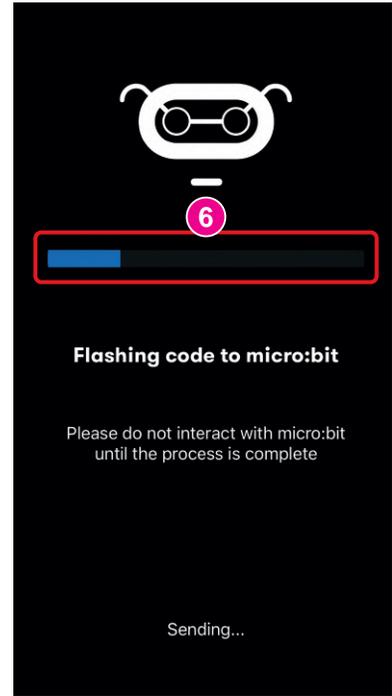
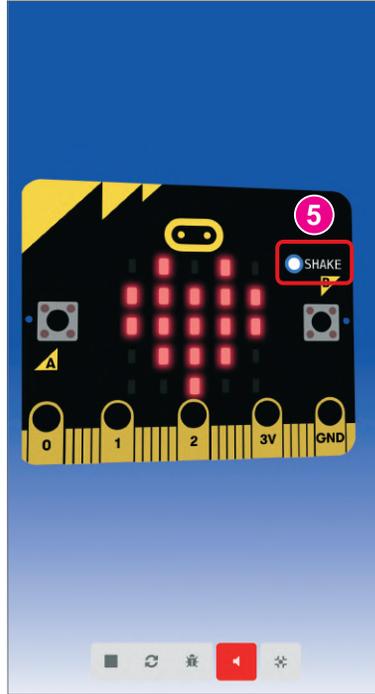
4 開始設計程式，程式設計完成，點擊【micro:bit 圖示】，以模擬器執行結果。或按【下載】下載程式。





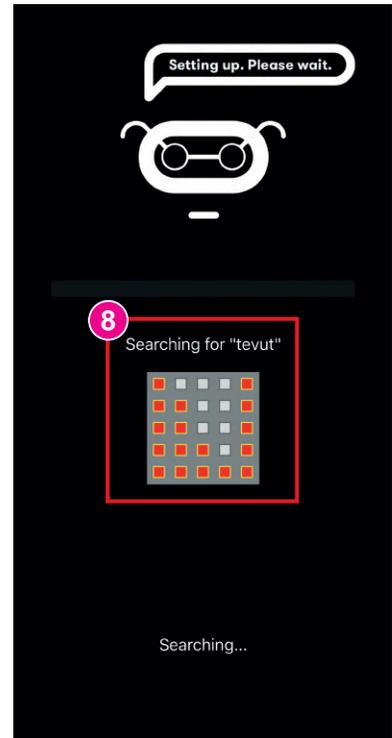
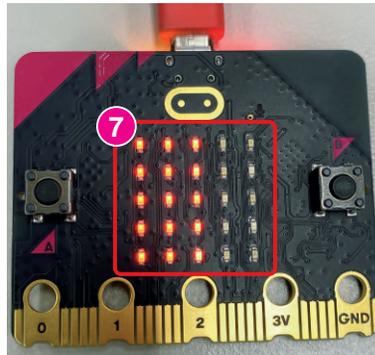
5 點按 micro:bit 模擬器執行程式。

6 或按下載，將程式上傳到 micro:bit。



7 上傳時 micro:bit 同步以 LED 顯示上傳進度。

8 上傳時會搜尋 micro:bit，如果未配對，請再次同時按「A+B+Reset」重新配對。





- 9 上傳完成按【OK】。
- 10 晃動 micro:bit 顯示程式執行結果。

