

C H A P T E R



micro:bit 介紹

1-1 micro:bit 簡介

資訊教育在資訊時代越來越被重視，運算思維（Computational Thinking）已變成一項重要的素養能力，英國廣播公司 BBC（British Broadcasting Corporation）於西元 2016 年 3 月與美國微軟、韓國三星及美國 ARM…等公司共同技術合作，開發推出微控制器 micro:bit。micro:bit 板子重量約 5 公克而且大小只有 4×5 公分，攜帶方便，能幫助學生學習程式邏輯，並可以培養運算思維能力。

micro:bit 的程式開發，使用 Blockly、JavaScript 或 Python 語言來編輯程式，同時支援藍牙或 Micro USB 連接，使用者能透過電腦、平板或手機，在瀏覽器上即時撰寫程式碼，撰寫完畢後，下載副檔名為 hex 的檔案至 micro:bit 板子中，即完成程式的燒錄動作。另外，micro:bit 有一項非常便利的特點，無須安裝特殊軟體即可驅動 micro:bit 板子，輕鬆享受編輯程式的樂趣，如同將資料上傳至雲端一樣地簡單快速。

micro:bit 能夠培養學習者運算思維能力，另外，還能增進想像力、觀察力、創造力、問題解決與團隊合作等重要素養，許多複雜的專題還需運用演算法、資料結構或程式優化等技巧，micro:bit 實現了不同設計思維的可能，成為編程教育一個極好的平台，micro:bit 開發網站如圖所示。


▶ 1-3 micro:bit Python 開發環境

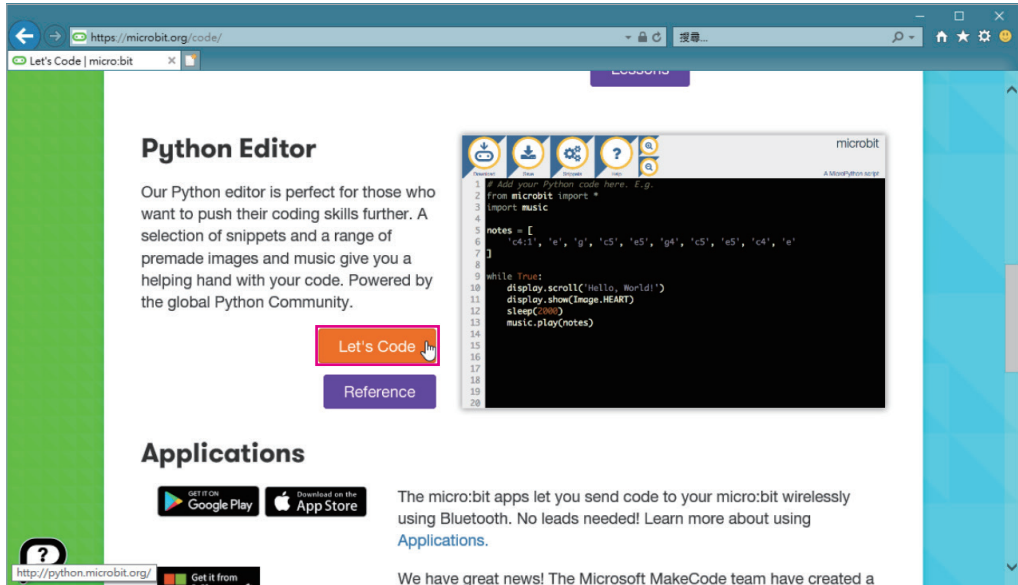
Python 語言是在西元 1989 年，由創始人吉多范羅蘇姆（Guido van Rossum）所設計，Python 是一種直譯式的電腦程式語言，除了原本功能就相當完備的標準函式庫，能夠完成相關基礎程式設計需求外，還能夠整合第三方函式庫套件，提升不同類型應用程式的開發效率，例如：臉部辨識應用、資料庫應用、網頁資料擷取與分析應用…等。

Python 程式語言受到許多程式設計師的喜愛，其具有下列特色：

- 免費且開源：Python 是免費且開放原始碼的程式語言，使用者可以自由地運用或修改其原始碼。
- 簡單易學：Python 的語法簡單易學，其語法結構與英文相近，初學者的進入門檻相較 C/C++ 語言為低。
- 移植性較高：使用 Python 語言撰寫的程式，很容易移植到不同的作業系統平台上，具有高可攜性（Portability）。也就是說，Python 語言的可攜性高，在某一個作業系統下開發的程式，可以在少量修改或完全不修改的情況下，順利地移植到另一個作業系統裡執行。
- 豐富的第三方套件：Python 語言能使用許多第三方所開發的函式庫套件，使得 Python 語言更加強大，讓程式設計師能更加專注於問題的解決。

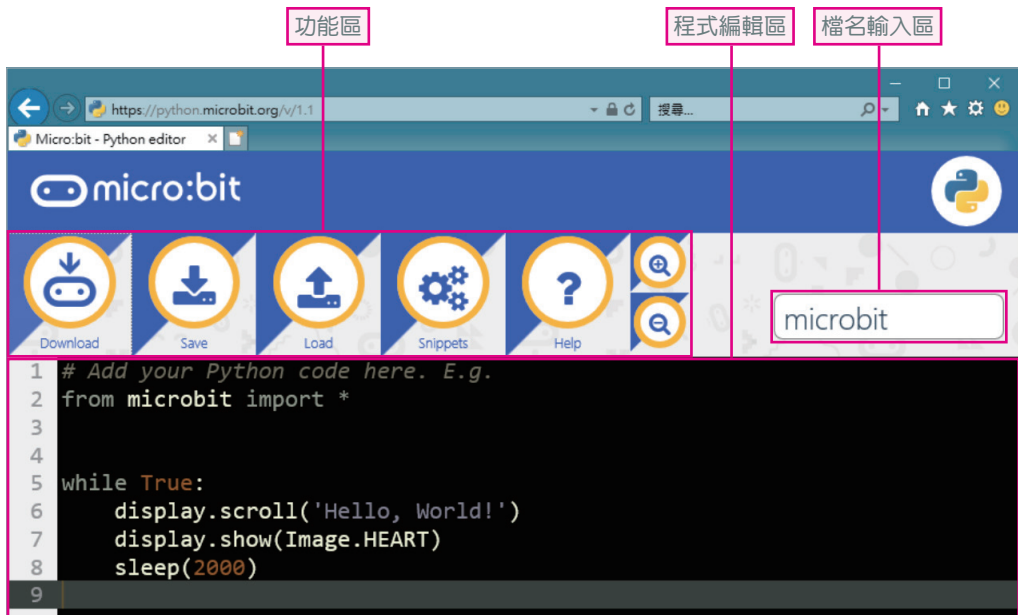
micro:bit 板子支援的 Python 語言為 Micro Python，相對於 micro:bit Blockly 學習上會比較困難些，建議初學者可先閱讀本書的 micro:bit Blockly 部份，如具有一定程式基礎者，可從本書的 micro:bit Python 著手，接下來介紹 micro:bit Python 的開發介面。

網路上有許多 Python 的開發環境，此處介紹 micro:bit 官網的 Python Editor 編程方式，點選  按鈕，即可進入編輯 Python 程式碼的網頁。



micro:bit 官網的 Python Editor 編程網址 <https://microbit.org/code/>

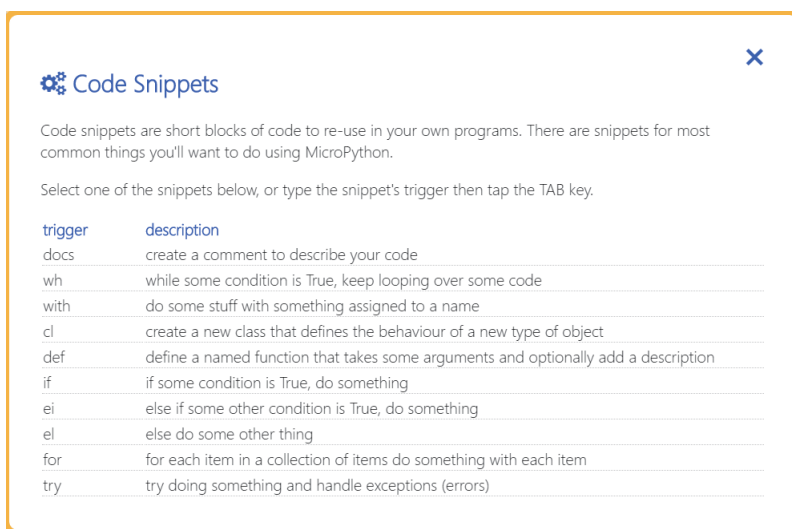
進入 micro:bit 官網的 Python 編輯畫面，其畫面功能分為：功能區、檔名輸入區與程式編輯區，網頁會附上一個範例，讀者能藉由此範例瞭解 micro:bit Python 的編程環境，該程式會在 micro:bit 板子上跑出「Hello, World!」文字，然後印出「愛心」符號 2 秒鐘。



在右上角的檔名輸入區，可以輸入此程式的名稱，然後將程式下載（Download）到裝置上，若日後想修改程式內容，可將檔案匯入（Load）到瀏覽器以進行修改。在 micro:bit Python 的編輯環境，會以使用者設定的檔名直接儲存，不會在檔案前方再加上「microbit-」文字，此處以自訂檔名「first」為例，micro:bit Python 會直接將下載的檔案存成「first.hex」，開發者只要將下載後的 hex 檔案，使用檔案總管將檔案複製到 micro:bit 板子的磁碟上，就完成程式的燒入動作。

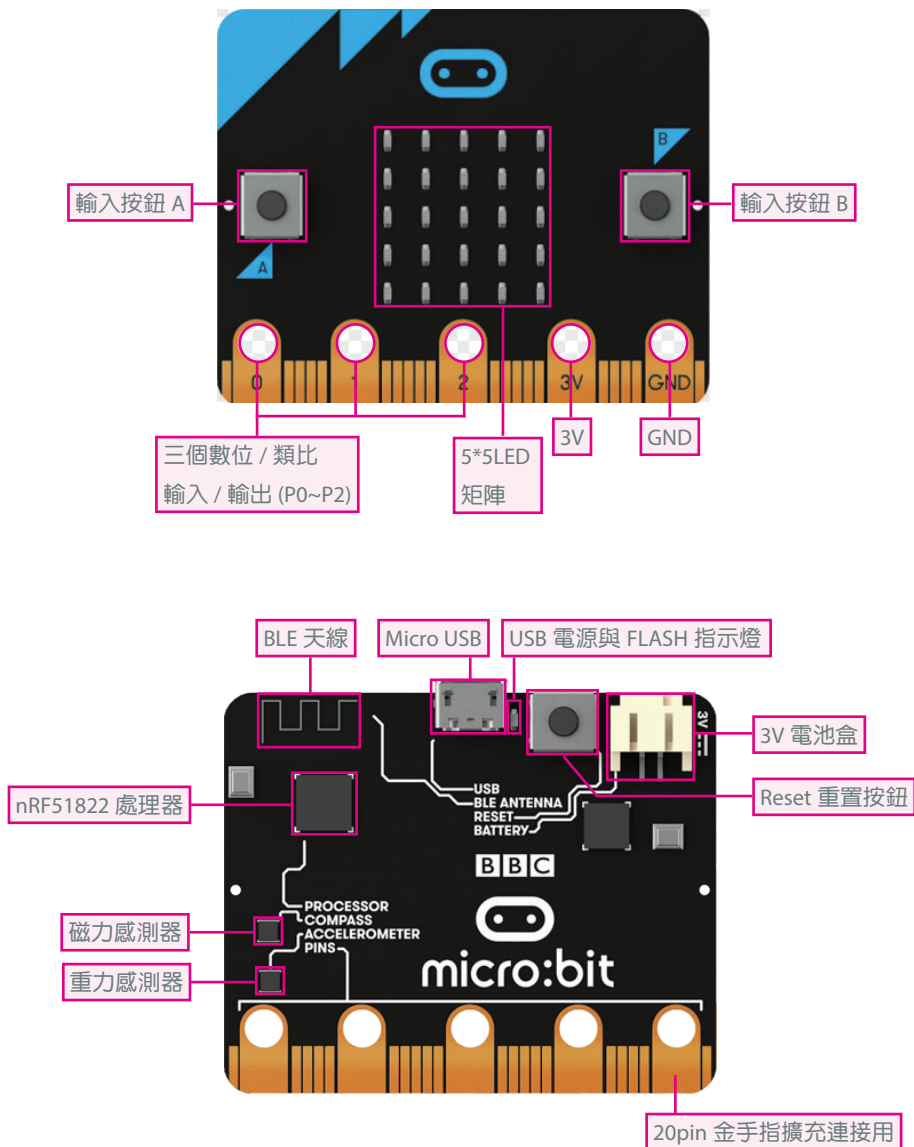
micro:bit Python 上方的按鈕之功能如下：

- Download 按鈕：下載開發好的 micro:bit Python 檔案，其副檔名為「.hex」。
- Save 按鈕：當開發者將程式編寫到一定程度時，可按此按鈕將自己的心血儲存下來，Save 按鈕所儲存檔案的副檔名為「.py」檔。
- Load 按鈕：當開發者想把先前開發的檔案重新開啟時，可按此鈕將想要的檔案匯入 micro:bit Python 的開發介面。
- Snippets 按鈕：這裡記錄著常用的程式碼片段，供開發者快速查閱，其內容如圖所示。



- Help 按鈕：提供相關資源，幫助開發者更加瞭解 micro:bit Python 的開發。

1-5 micro:bit 板子介紹



- **nRF51822 處理器**：nRF51822 處理器是功能強大的微型處理器，非常適合用於低功耗藍牙 Bluetooth 和 2.4GHz 的無線應用。
- **藍牙無線通訊**：micro:bit 板子採用 Bluetooth Low Energy (BLE) 低功耗藍牙技術，此技術現在已納入許多電子設計中，例如：智慧型手錶、健身追蹤器、醫療監測器…等，且未來 BLE 將更加廣泛地運用在各項消費性電子產品與物聯網裝置中。
- **重力感測器**：內部有一塊重物與壓電片結合在一起，透過正交兩個方向產生的電壓大小，來計算出水平的方向，運用在手機裝置中時，可用來切換橫屏與直屏方向。
- **磁力感測器**：測量電阻變化來確定磁場強度，可以偵測方向，大多運用在指南針或地圖導航當中。
- **micro:bit 提供 3 個數位 / 類比的輸入 / 輸出、3V、GND 等大型孔環**，有利於開發者連接其他簡易裝置，像是蜂鳴器或耳機。

▶ 1-6 micro:bit 與 Arduino 比較

在 micro:bit 出現之前，Arduino 被譽為最適合創客教育的電路板，有許多創客使用 Arduino UNO 晶片進行作品開發，那就讓我們比較一下 micro:bit 與 Arduino 吧。

	micro:bit	Arduino UNO
處理器	32 位元 ARM Cortex-M0 16MHz	ATmega328 16MHz
尺寸	4×5 公分	6.9×5.3 公分
編程環境	使用瀏覽器進行，無須下載任何編譯程式，板子與電腦連接是隨插即用	需要安裝開發軟體，板子與電腦連接需要其他驅動程式
編程語言	Python、JavaScript、Blocks	C/C++
電壓	3V	3V~5V

C H A P T E R

06

專題製作

經過先前的幾個章節介紹後，你是否躍躍欲試想做出一些更有趣、更大型的專題呢？就讓我們一起運用前面所學的拼圖類別，創造出不同變化的專題實作，提升自我的創客實力。



專題實作 1：我是賽車高手

學習重點：多個類別的應用



程式設計目標

設計一款可在 **micro:bit** 裝置上運行的賽車小遊戲，在遊戲開始時會出現主角的車子，方向的控制需要靠按下 **A**、**B** 鍵來左右移動。遊戲中會隨機出現許多車子，主角的車子需要左右閃避，以避免與其他車子碰撞。若成功閃避車子則可以得分，若與其他車子碰撞則遊戲結束，當遊戲結束時會顯示玩家分數。

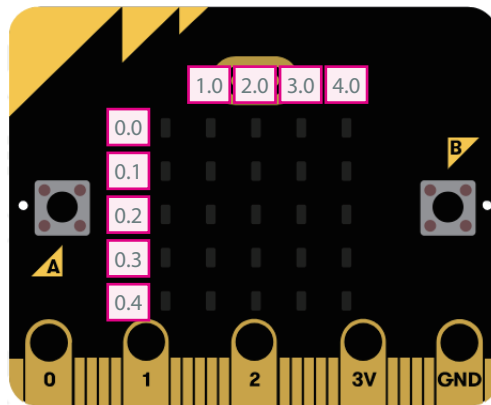
拼圖 Blocks 編輯

參考檔案：microbit-ch6-1-1.hex

STEP 1 我們需要先了解 micro:bit 上 LED 燈的座標位置，如圖為（X.Y）座標。

X：代表 X 軸方向的燈號，數值會從 0 開始算起。

Y：代表 Y 軸方向的燈號，數值會從 0 開始算起。



STEP 2 接著我們需要先顯示主角的位置，為了讓主角能在中間出現，因此我們將主角設在座標（2, 4）的位置。先前有介紹過使用遊戲類別，在此我們一樣使用此功能，宣告變數「playerCar」來代表遊戲的主角，運用遊戲類別「創建角色於」顯示主角的位置，並從遊戲類別中加入計分設定「得分設為 0」，如圖。



- STEP 3** 設定完主角（playerCar）的位置後，接著宣告其他車子的變數名稱，這裡宣告「car1~car5」為其他車輛的變數（因為總共有 5 行車道，所以設定 5 個變數），如圖。



- STEP 4** 宣告完此遊戲所有所需的變數後，我們接著設定主角車輛（playerCar）的控制，點選輸入類別中「當按鈕 A 被按下」，並將遊戲類別中「角色...的 x 改變 1」放入拼圖中，由於主角車輛（playerCar）僅會在 X 軸移動，因此這裡不需要修改 Y 的值，如圖。

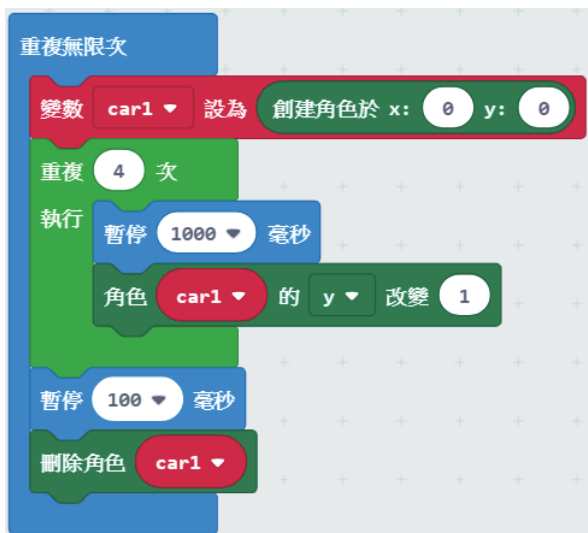


由上圖可以發現「當按鈕 A 被按下」角色的 X 改變「-1」，也就代表角色會向左移動 1 格。相反的，「當按鈕 B 被按下」角色的 X 改變「1」，也就代表角色會向右移動 1 格。然而不論移動左或右，皆可以獲得 1 分。

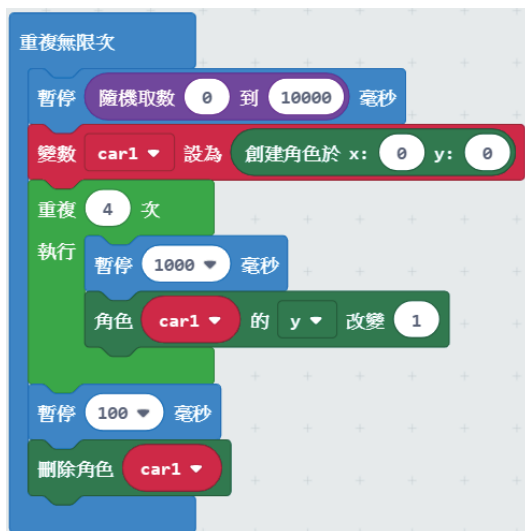
- STEP 5** 在這個階段開始編寫其他車輛（car1~car5），首先一樣需要將角色建立在適合的位置如 car1 只會在最左邊的車道行駛，因此一開始將其位置設定於座標（0,0）的位置，如圖。



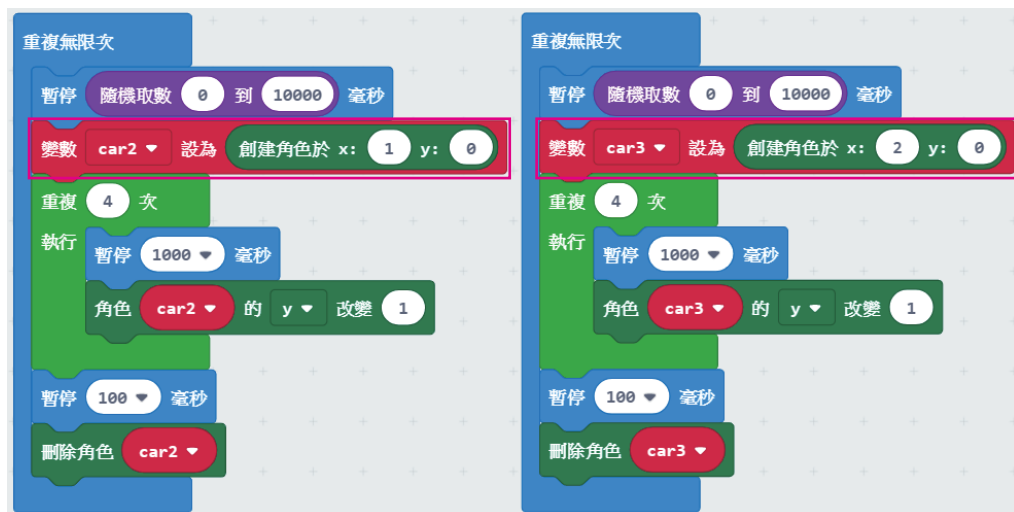
- STEP 6** 接著為了使車輛可以向下移動而產生主角車輛往前行駛的感覺，因此使用迴圈類別「重複…次，執行…」與遊戲類別「角色…的 x 改變 1」將 car1 的位置每次都向下移動一格。在此需要將 X 改成 Y，因為車輛是向下移動，並非水平移動。最後當此角色車輛（car1）到達底部時需要將其刪除，以避免往後的執行出現錯誤，因為此段程式是會一直重複出現在遊戲中，如下圖，可以發現在某些地方有加入時間暫停的拼圖，這是為了讓角色移動時有個緩衝，避免因速度太快而有曇花一現的感覺。

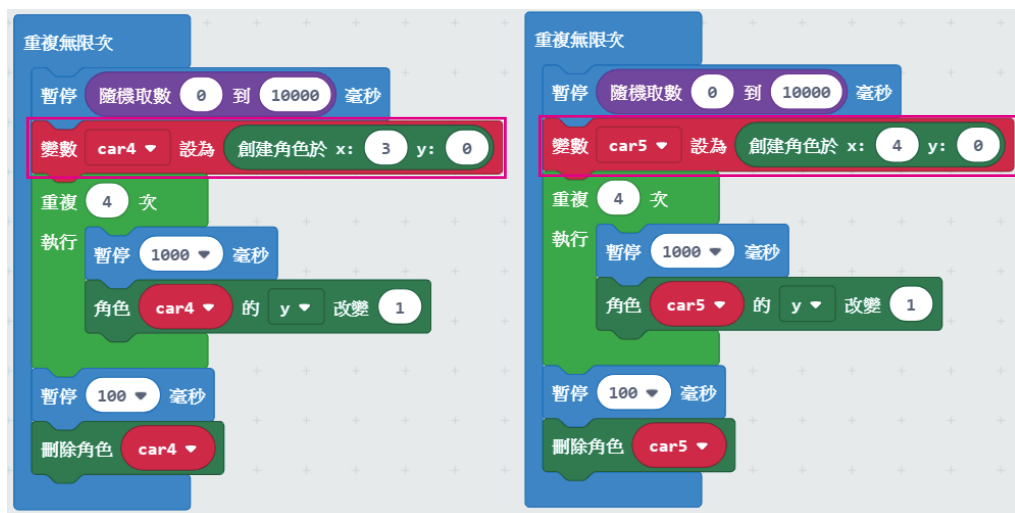


- STEP 7** 接著運用數學類別中「隨機取數 0 到…」這個拼圖區塊放入暫停時間的拼圖中，主要是為了避免每一行的車輛同時出現，導致主角車輛（playerCar）無法閃避的狀況。隨機取數的範圍可以依照設計者自行設定時間，在此設定暫停時間為 0~10 秒，如圖。



- STEP 8** 然而其他 2~5 行的車輛，即先前宣告的變數 car2~car5，也與 car1 的撰寫方式相似，但主要的不同在於創建角色的位置，需要依照車輛分配的車道設定 X 的位置，如圖。





STEP 9 撰寫完所有車輛的移動與位置後，接著我們要來設定當車輛碰撞在一起的時候會發生什麼事件。依照此遊戲程式設計目標當車輛碰撞時遊戲就結束了，然而可能發生碰撞的地方會有座標 (0, 4)、(1, 4)、(2, 4)、(3, 4)、(4, 4) 這五個位置，若要一一寫出碰撞後所執行的事件，會使程式變得相當冗長，因此在這裡我們使用函式類別，建立一個函式取名為「Collision」，代表碰撞時所觸發的事件。

在 Collision 此函式中，我們加入基本類別中「清空 畫面」這個拼圖，先將畫面都清除，接著出現「X」的圖示代表已發生碰撞，接著暫停一段時間，顯示 Game Over 的字樣，使用「遊戲結束」拼圖可以直接顯示 Game Over 的字樣，最後將所得的分數呈現出來，如圖。



STEP 10 有了碰撞的函式，接著我們運用遊戲類別中「角色…碰到…？」這個拼圖與邏輯類別中「如果…那麼…」的拼圖做結合。首先需要先確定是哪一行車道的車輛，因此需要先新增一塊「如果…那麼…」的拼圖，並將變數 **car1** 加入。接著再新增一塊「如果…那麼…」的拼圖，運用遊戲類別中「角色…碰到…？」將主角車輛（**playerCar**）與第一行車道的車輛（**car1**）變數放入。最後如果車輛有碰撞就呼叫「Collision」函式，如圖。



STEP 11 完成了第一車道車輛（**car1**）與主角車輛（**playerCar**）後，第 2~5 行車道的車輛碰撞也如法炮製，如圖。





至此完成了「我是賽車高手」所有拼圖區塊的設計。



專題實作 2：廢材自走蟲

學習重點：運用 micro:bit 做出機電整合的自走蟲



程式設計目標

使用 micro:bit 訊息類別控制 SG-90 馬達，當按下另一塊 micro:bit 板子的按鈕 A 時，會轉動馬達從 0 轉到 180 的幅度，並且在開始畫面會出現打勾圖示，讓使用者知道此為開始畫面，此例使用廣播類別進行操控，讓自走蟲會往前走動。



所需材料

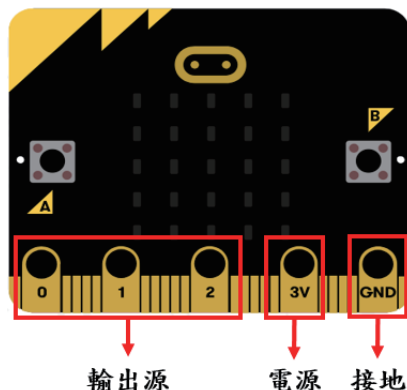
材料名稱	數量
紙板	1
迴紋針	6
micro:bit	2
SG90 馬達	1
膠帶	1
美工刀	1
鱷魚夾杜邦線	3



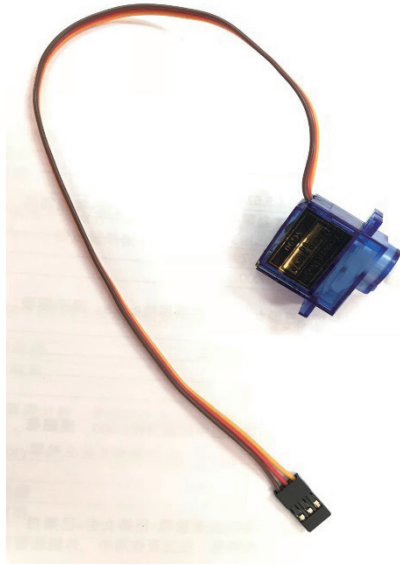
自走蟲外觀製作

STEP 1

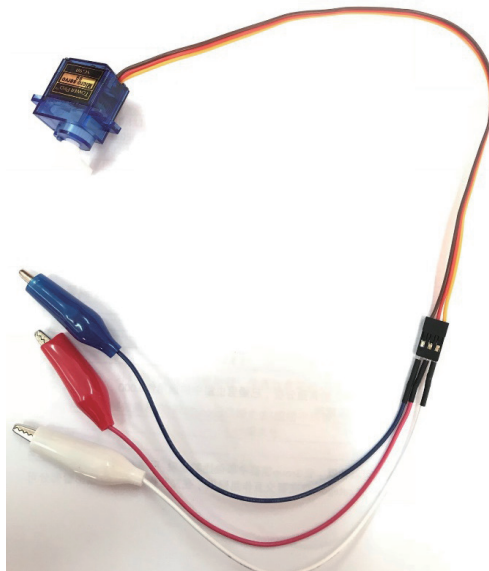
看到 micro:bit 面板上有 5 個圓形的接口，上面分別寫著 0、1、2、3V、GND 等數字與文字，其中數字 0~2 代表輸出源，3V 代表電源，最後 GND 則表示為接地孔，如圖。



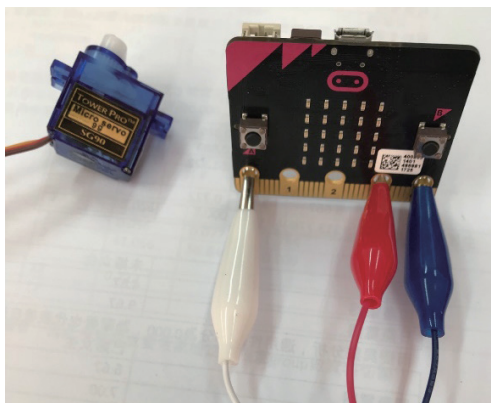
- STEP 2** 如圖為 SG90 的馬達，可以看到有三種顏色的電線，其中咖啡色線代表接地線（GND），紅色線代表電源線，橘色線代表輸出線。



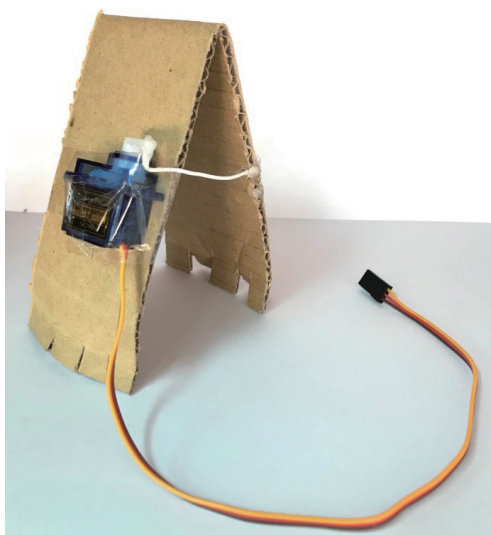
了解了電線的配置後，拿出鱷魚夾杜邦線依序插入接孔，如下圖。



- STEP 3** 馬達接線完成後，將鱷魚夾與 micro:bit 對應的接孔連結，「腳位 0」對應「輸出線」、「腳位 3V」對應「電源線」、「腳位 GND」對應「接地線」，如下圖，即完成接線的工作。



- STEP 4** 完成了線路的連接後，接著進行廢材自走蟲的設計。首先將紙板裁切成長方形並將其對摺，對摺後將其下擺使用剪刀稍微剪幾刀，使自走蟲有較佳的抓地力。剪裁完成後，運用迴紋針當作紙板與馬達連結的橋梁，再利用熱溶膠固定。最後將馬達使用膠帶固定於紙板上，即完成廢材自走蟲的製作，成品可參考下圖。另外，可以為自己設計的自走蟲外觀裝飾一下，設計出獨一無二的廢材自走蟲。



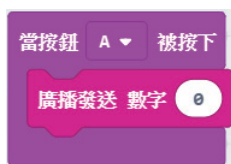
拼圖 Blocks 編輯

參考檔案：microbit-ch6-2-1.hex

- STEP 1** 完成了自走蟲外觀的設計與接線後，接著進入程式撰寫的部分。首先點選基本類別中「當啟動時」與「顯示圖示打勾」，做為開始的辨別圖示。



- STEP 2** 接著點選輸出類別中「當按鈕 A 被按下」，此時需要加入廣播類別中的拼圖「廣播發送數字 0」並將兩個拼圖組合起來，就完成控制按鈕的部分，如圖。



- STEP 3** 最後則是設定當按鈕按下後，會出現什麼樣的事件動作。由於這個自走蟲使用到馬達與接線，因此需要運用到腳位類別。點選腳位類別中「伺服寫入腳位 P0 至 0」拼圖塊，此拼圖塊功能為控制訊號源輸出所需的動作，先前我們將鱷魚夾接到 P0 的位置，因此馬達會接收 micro:bit 上 P0 所設定的訊息，此處角度需設定為「180」，也就是讓馬達角度由「0」轉至「180」；接著我們在中間加入基本類別中「暫停 (ms)500」的拼圖，此步驟是為了讓轉動有所間隔。接著再新增腳位類別中「伺服寫入腳位 P0 至 0」將數字設為「0」，讓馬達轉動回「0」的位置，最後加入暫停的拼圖區塊，即完成馬達動作設定。前面我們有使用到廣播類別進行控制，因此在此需要加入廣播類別中「當收到廣播 receivedNumber」拼圖，使廣播訊息可以被接收，如圖。

APPENDIX

B

本書範例材料表

全書範例皆使用 micro:bit 開發板完成。關於周邊模組部份，皆使用市售模組來完成，您可以到電子材料行或是相關網站上購買。

序號	設備或元件名稱	規格	數量
1	micro:bit 開發板		2
2	USB 傳輸線	100cm	1
3	電池盒	開關式	2
4	碳鋅電池	AAA	4
5	有線耳機	具有 3.5mm 耳機孔	1
6	紙板	30*30cm	1
7	迴紋針	圓形迴紋針 - 28mm	6
8	馬達	SG-90	2
9	美工刀		1
10	鱷魚夾杜邦線	20cm	3
11	冰棒棍	15cm	1
12	剪刀		1
13	土壤濕度感測器	YL-69	1
14	橡皮筋		2
15	智高機器瓦力車零件組	智高機器瓦力車	1
16	鹼性電池	3 號	6