



智慧城市 IOT

MICRO:BIT 入門套裝

Copyright © Smarthon Limited. All rights reserved.



關於 SMARTHON (研科)



SMARTHON LIMITED 的總部位於香港，為教育和創意工作者設計、製造和銷售以常規課程，如通識教育等為主的產品，我們的使命是促進所有人能夠獲得高品質的科技教育。

SMARTHON 的成立是為了設計簡易的設備，並將不斷發展的技術應用於教學。例如，教師可以使用 micro:bit 和傳感器來教導學生設計一個小型溫室系統，然後利用物聯網技術快速將相關數值傳輸到智能設備。教師可以實時調整溫度或濕度，讓學生觀察不同的變化，記錄下來，並用於學習、分析和了解溫室系統的運作。



目錄

- 介紹 P. 2
- 甚麼是智慧城市及IoT? P. 5-7
- 零件列表: 所用部件及模型清單 P. 8-11
- Smarthon IoT:bit介紹 P. 12-13
- 快速入門: 添加PXT擴展 P. 14
- 案例 1: 自動遊樂場燈 P. 15-18
- 案例 2: 智慧停車場閘杆 P. 19-22
- 案例 3: 智能垃圾桶 P. 23-26
- 案例 4: 城市噪音檢測器 P. 27-30
- 案例 5: 車速檢測器 P. 31-36

目錄

- 物聯網案例 6: 智慧天氣監測站 P. 37-44
- 物聯網案例 7: 智能家居防盜系統 P. 45-51
- 物聯網案例 8: 智能交通燈 P. 52-60
- 物聯網案例 9: 遙距智能門控制 P. 61-68
- 物聯網案例 10: 節能街燈 P. 69-75
- 物聯網案例 11: 自動天台簷篷系統 P. 76-83
- 場境示例 1-3 P. 84-86
- 附錄 I: Smarthon電子模組 – 探索更多 P. 87

SMARTHON 智慧城市

IoT 學習套裝



SMARTHON 智慧城市 IoT 學習套裝是專門為了教授物聯網(IoT)設的套裝。你可以利用物聯網，建設一個獨一無二的智慧城市。套裝內提供的Smarthon IoT:bit 開發板，能夠使用各種傳感器及驅動器，令你能夠在設計中實現多種有趣的功能。例如，使用傳感器去偵測交通狀況然後上載到互聯網，令你能夠作出統計或是控制等。

 <https://smarthon-docs-en.readthedocs.io/en/latest/smartcity/index.html>

甚麼是智慧城市？

智慧城市是架構、整合資訊科技 (ICT) 以在六大範疇——交通、生活、環境、人民、政府及經濟，增加可持續性、效率和生活質素。在本套裝裡面，我們將會集中在三個類別——智慧生活、智慧環境和智慧交通。



智慧生活

能夠使用物聯網/科技去提升方便程度/生活水平



智慧政府

政府支援在私人/公共領域及城市建設中，使用及發展科技



智慧環境

監測污染及減少浪費，發展綠色社區



智慧經濟

強化現有經濟支柱及透過科技發展新經濟支柱



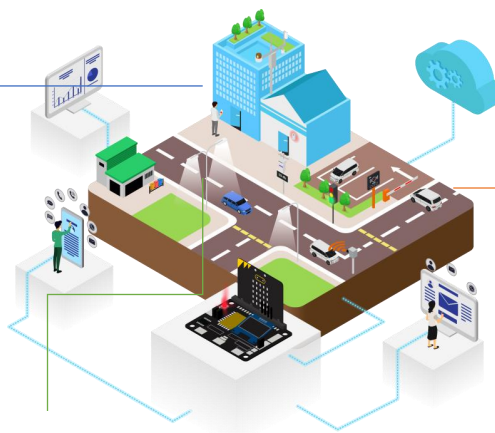
智慧交通

透過提供實時資訊，提升出行體驗



智慧人民

提供編程訓練去培養IT專業



什麼是 IOT(物聯網) ?

物聯網 (IoT) 是一個透過互聯網連接不同物件 (有形或無形，包括電腦設施，系統)的網絡。透過在物件上添加傳感器，軟件或其他科技，使其能夠在不需要人與人或人與電腦的互動的情況下，進行資料連接和交換。物聯網中有三種分享資訊的方式：

1. 收集及分享資訊

物件上的傳感器收集資料 (溫度，濕度，雨滴等) 並上傳到中央系統以進行進一步分析

例子：上傳數據到 ThingSpeak

例子2：使用 IFTTT 發送郵件

2. 接收訊息及反應

物件等待指令，之後作出反應

例子：使用 App Inventor 2/IFTTT 服務控制 micro:bit

3. 兩者皆做

收集資訊並使用互聯網分享，然後作出反應

例子：物件與物件通訊

智能雨傘能收集雨滴數據並傳送至氣候系統。氣候系統會傳送回降雨機率使雨傘進行開/關。



所用部件

SMARTHON IOT BIT:

這是一塊專為讓 Micro:bit 加入物聯網而設計的開發板。它能夠連接到不同的傳感器和驅動器。



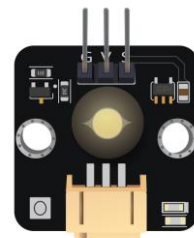
交通燈模組:

內含三種不同顏色LED燈，分別為紅、黃、綠三色，就像現實生活中的交通燈一樣。



白色 LED:

能夠發出白色的LED光，使用者能控制它的亮度。



雨水傳感器:

能夠使用導電度去判斷是否有雨水在傳感器上。



超音波距離傳感器:

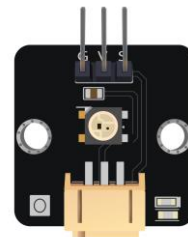
能夠使用超音波去量度傳感器和物件的距離。



彩色 LED

(WS2812B):

這是一塊LED綜合芯片，能夠透過編程發出不同RGB顏色的光。

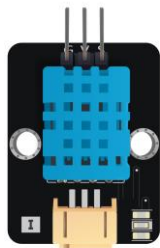


PART LIST

溫濕度傳感器

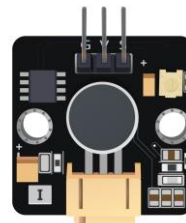
(DHT11):

包含兩個部件，分別能偵測濕度及溫度。



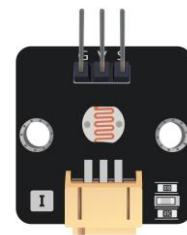
聲音傳感器:

使用帶有擴音器的麥克風來檢測環境的聲音等級。



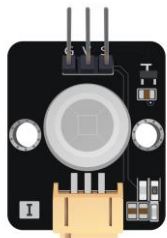
光度傳感器:

能夠使用光敏電阻來檢測環境的亮度。



運動傳感器:

運動傳感器是使用紅外線感測器去檢測出前方有沒有正在移動的物體。



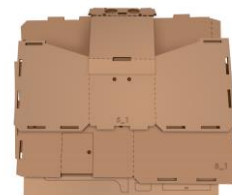
180° 舵機:

能夠使它的軸定位在 0 到 180 度之間。



紙板及木板模型:

能夠把傳感器及驅動器安裝在上面。



城市地圖:

大約尺寸為 594 毫米*420毫米，可以放置紙板模型在地圖上。



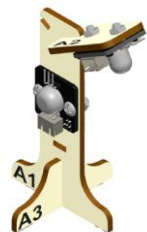
其他:

(9) 模組連接線 · (8) 延長線 · (12) M2*10毫米 螺絲 · (6) M3*10毫米 螺絲 · (30) M4*10毫米 螺絲 · (12) M2螺母 · (6) M3螺母 · (30) M4螺母 · (3) 萬字夾 · (2) L型支架 · (1) 螺絲批 · (2) 砂紙 · (1) 萬用黏土膠 (Blu Tack) · (1) USB 連接線 · (1)電池盒(AA*4) · (1)智慧城市套裝使用手冊 ·

模型清單

模型 A:

這是一個有斜坡燈的小型遊樂場燈模型。有一個傳感器端口在側邊。(例如：可以接上動作傳感器並用於偵測動作)



遊樂場燈

模型 C:

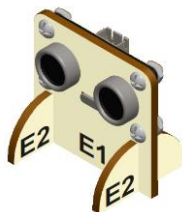
這個垃圾桶可以透過超音波距離傳感器去偵測垃圾桶內的垃圾數量。在垃圾桶前有一個傳感器端口，你可以設置一個彩色LED去顯示垃圾的數量。



垃圾桶

模型 E:

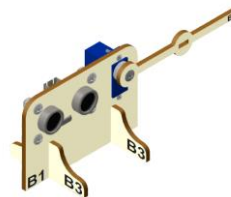
這個模型中，超音波距離傳感器可以用於量度模型與目標物件之間的距離。它可以和模型D一齊使用以展示物件的距離/速度。



車速檢測器

模型 B:

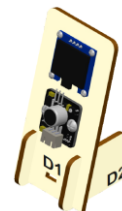
這個模型中，超音波距離傳感器被用於偵測是否有車輛經過。同時，舵機用於控制閘杆的角度。



停車場閘杆

模型 D:

這個OLED顯示屏可以用於顯示有用的資訊。使用者可以把傳感器接上OLED顯示屏以收集資訊。(例如：聲音傳感器或者溫度傳感器)。



OLED 顯示屏

模型清單

模型 F:

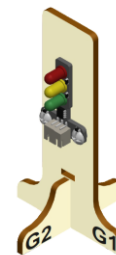
這是一個小型建築物模型。模型上有數個傳感器端口在屋頂上，使用者可以設置不同的傳感器/驅動器。（例如：雨滴傳感器，溫濕度傳感器）



小型建築物
(公寓/商業大樓)

模型 G:

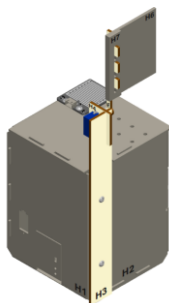
這個模型可以接上交通燈模組。這個模型可以放置在城市的街道上，並使用其3個LED燈來展示交通路況。



交通燈

模型 H:

這是一個有著由舵機控制的自動天台簷篷系統的高建築物模型。使用者可以設置不同的傳感器/驅動器。（例如：雨滴傳感器，溫濕度傳感器）



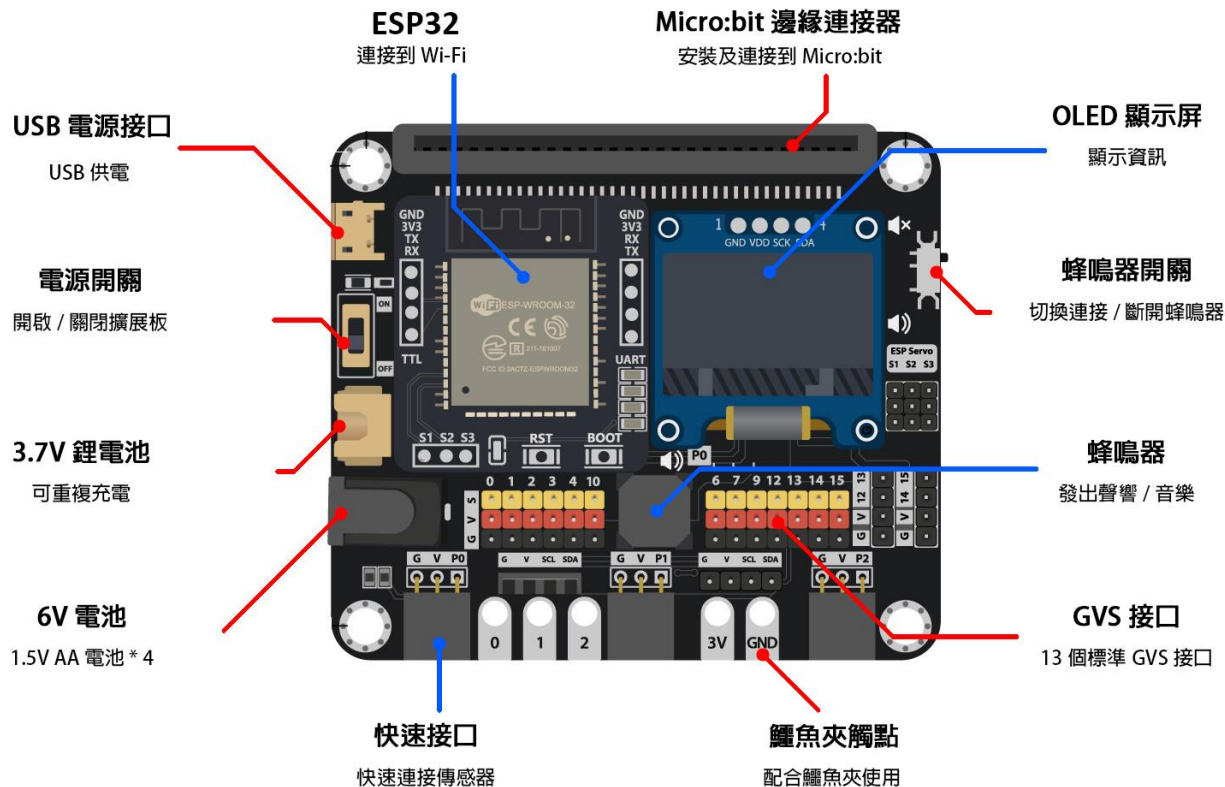
大型建築物
(摩天大樓/高層住宅)

模型 I:

這是一個有著斜坡燈的高尺寸街燈。它能夠支援城市內的智能照明系統。在你的專案中，你可以使用物聯網來控制照明系統。



街燈



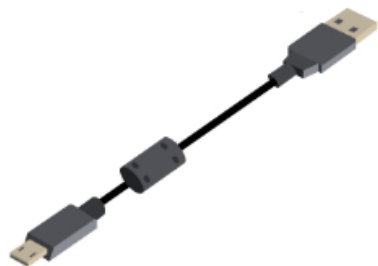
<https://smarthon-docs-en.readthedocs.io/en/latest/smarcity/index.html>

連接電源

IoT:bit 電壓範圍：3.7V DC-6V DC。有三種電源選項：

選項 1

- USB 連接線



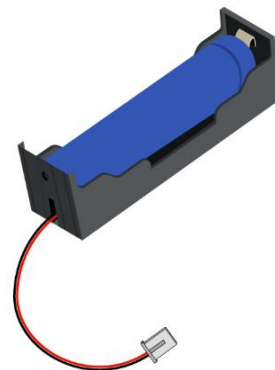
選項 2

- 4 x AA 電池 (本套裝內並不包含電池)



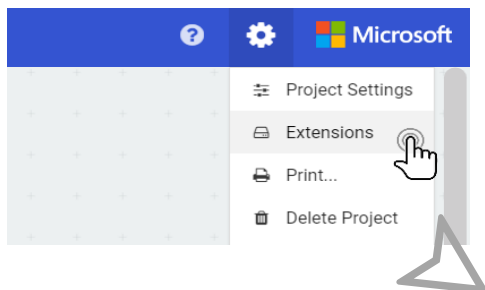
選項 3

- 把18650 3.7V 電池放入電池盒，並把電池盒插頭插入XH2.54鋰電池接口(本套裝內不包含18650電池及18650電池盒)

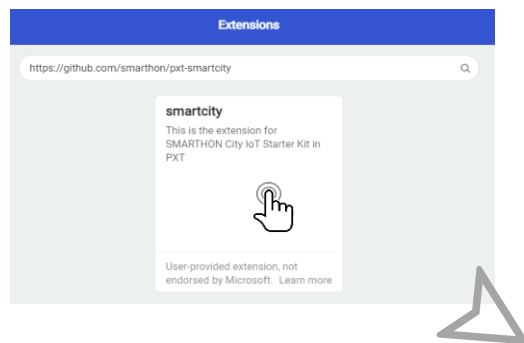


快速開始： 添加PXT擴展

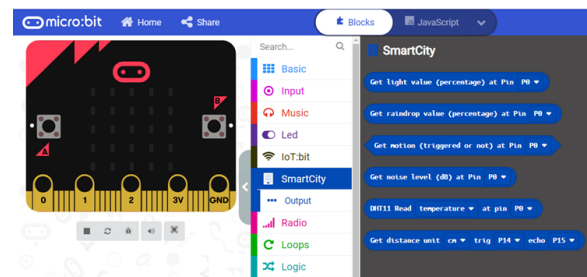
1. 打開 <https://makecode.microbit.org/>，
建立一個新的專案。



2. 搜尋 “<https://github.com/smarthon/pxt-smartcity>”
及點擊 “smartcity”。



3. 當載入完成時，[SmartCity](#) 擴展便會出現。



● 自動遊樂場燈

白色LED

● 運動傳感器



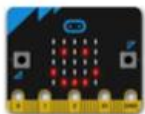
1.自動遊樂場燈

程度：★☆☆☆☆

介紹

自動遊樂場燈是一款能夠自動開啟的燈。它能夠在感應到有人活動時自動開啟，無人時自動關閉，以節省電力。

所用部件



micro:bit (1)



IoT:bit (1)



白色 LED (1)



運動傳感器 (1)



三腳模組連接線 (2)



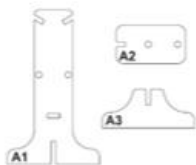
M4*10 毫米螺絲 (4)



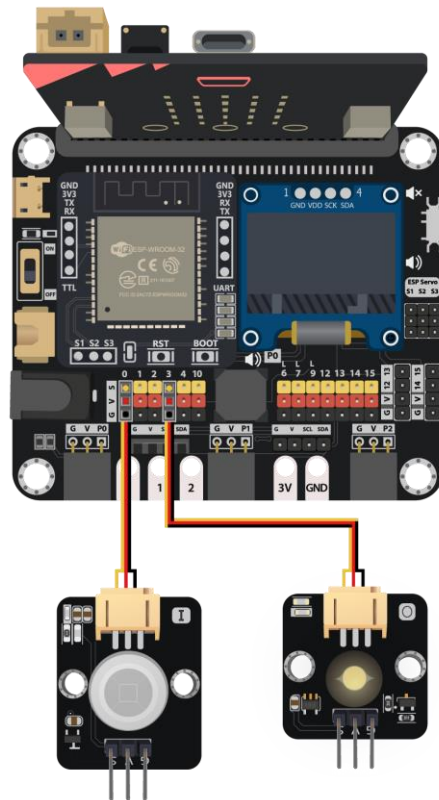
M4 螺母 (4)



螺絲批 (1)

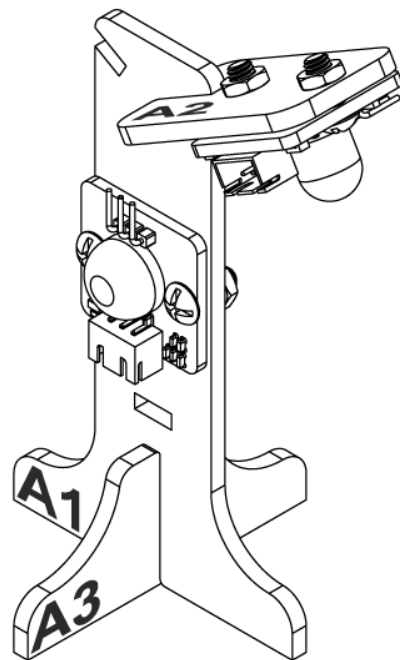
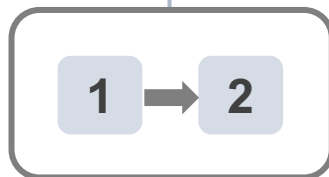
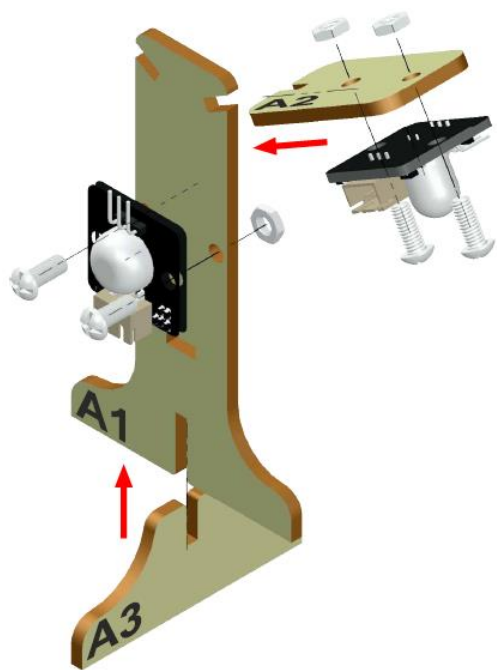


A1-A3 模型(1)



組裝步驟

A1-A3 模型	x1
運動傳感器	x1
白色 LED	x1
M4 螺母	x4
M4*10毫米螺絲	x4



編程



除了開燈，你還可以怎樣運用運動傳感器？（例如：計算人數）

1 當啟動時

2 啟用設為 false



關閉 micro:bit LED

3 重複無限次

4 如果 取得 動作傳感器數值 接口 P0 = true 那麼

5 開啟白色LED至亮度 1023 接口 P3

6 暫停 10000 毫秒

7 否則 開啟白色LED至亮度 0 接口 P3

步驟:

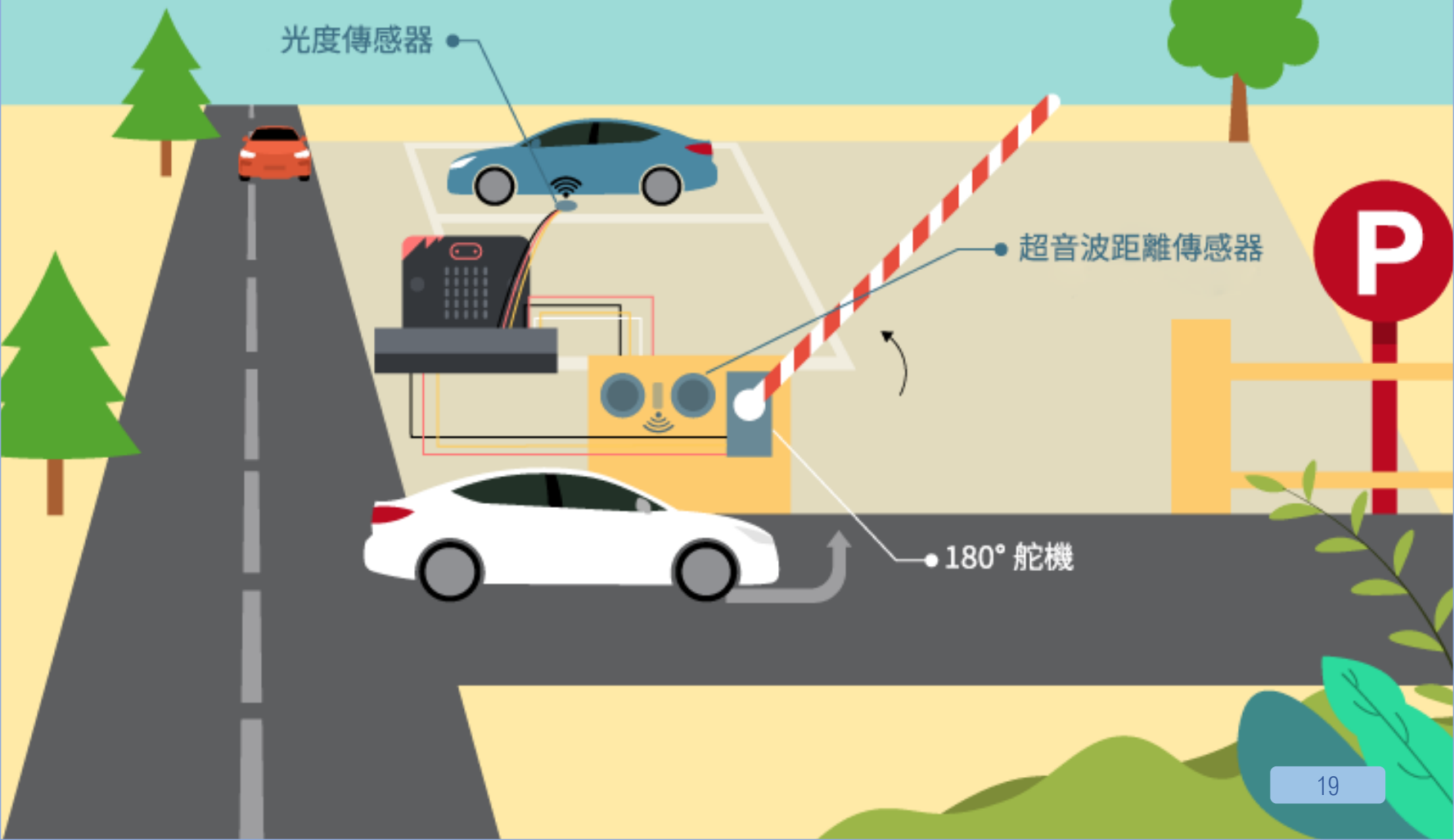
1. 「當啟動時」只會在程式最開始時運作。
2. 在「燈光」模組裏尋找啟用設為“FALSE”並加入至「當啟動時」（因為P3預設是用於控制Micro:bit的LED，所以需先關閉）。
3. 「重複無限次」的運作方式和迴圈相似。
4. 如果動作傳感器被觸發（例如：有人經過）使P0 = “TRUE”。
5. P3的白色LED燈會被開啟
6. 它會暫停10秒所以燈光會持續10秒。
7. P3的白色LED燈會被關閉。

● 智慧停車場閘杆

光度傳感器

超音波距離傳感器

180° 舵機



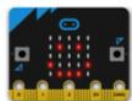
2. 智慧停車場閘杆

程度: ★ ★ ☆ ☆ ☆

介紹

智慧停車場閘杆能為人們帶來便利。通過自動偵測停車場內車位狀況（例如：空置車位），它能夠節省司機等候閘杆開啟和尋找車位的時間。

所用部件



micro:bit (1)



IoT:bit (1)



光度傳感器 (1)



超音波距離傳感器 (1)



180° 舵機 (1)



三腳模組連接線 (1)



四腳模組連接線 (1)



舵機尖螺絲 M2*7.5毫米 (1)
(於舵機套裝內)



M2*10 毫米螺絲 (2)



M2 螺母 (2)



M4*10 毫米螺絲 (4)



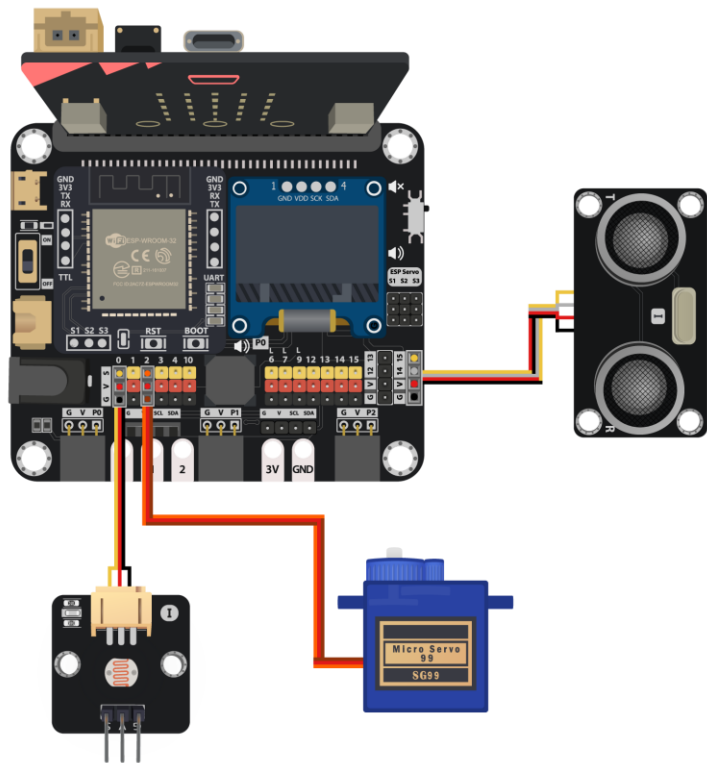
M4 螺母 (4)



螺絲批 (1)

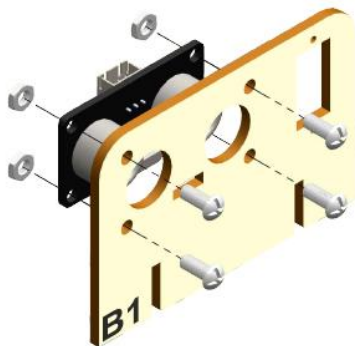


B1-B3 模型 (1)

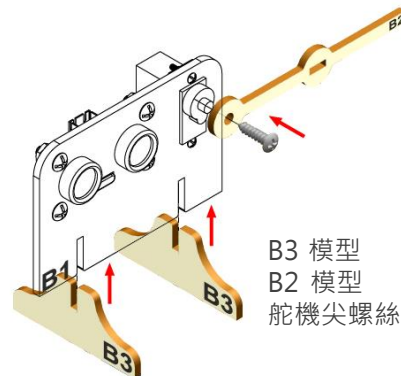
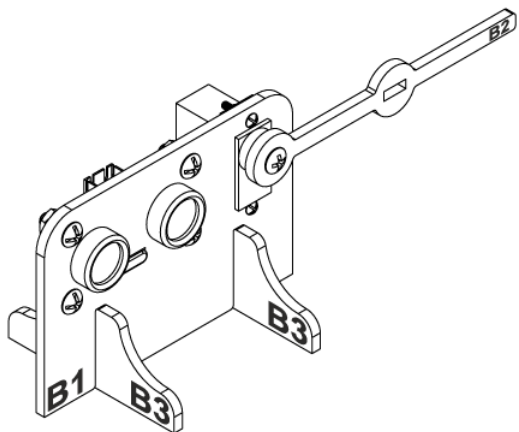
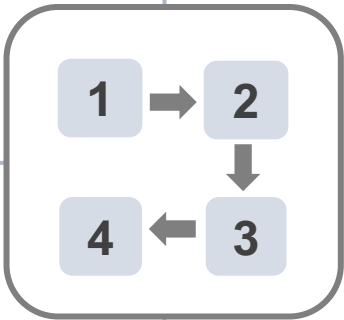
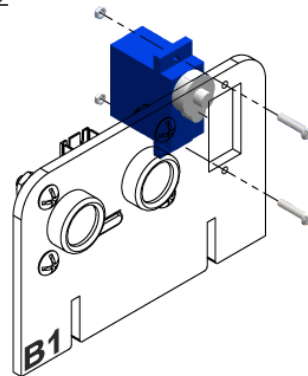


組裝步驟

- B1 模型 x1
- 超音波距離傳感器 x1
- M4 螺母 x4
- M4*10毫米螺絲 x4



- 180° 舵機 x1
- M2 螺母 x2
- M2*10毫米螺絲 x2



- B3 模型 x2
- B2 模型 x1
- 舵機尖螺絲M2*7.5毫米 x1

編程



怎樣可以在 OLED 顯示屏上顯示光和距離讀數？

1 當啟動時

2 變數 **distance** 設為 0
變數 **light** 設為 0
轉動舵機到 0 度 接口 P2

3 重複無限次

4 變數 **distance** 設為 取得 量度距離 使用單位 厘米 trig接口 P14 echo接口 P15

5 變數 **light** 設為 取得 亮度傳感器數值 接口 P0

6 如果 **distance** ≤ 5 那麼

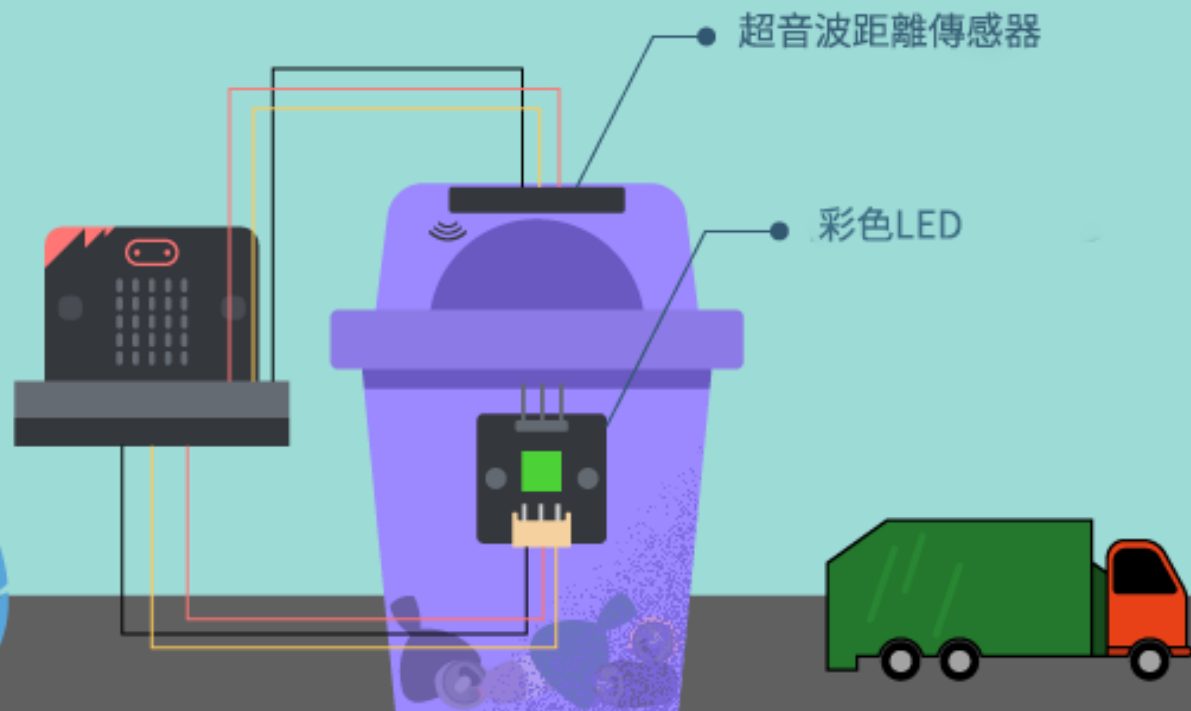
7 如果 **light** > 20 那麼

8 轉動舵機到 90 度 接口 P2
暫停 5000 毫秒
轉動舵機到 0 度 接口 P2

步驟：

1. 「當啟動時」只會在程式最開始時運作。
2. 將變數 **distance** 和 **light** 設為 0，轉動舵機到 0 度 接口 P2。
3. 「重複無限次」的運作方式和迴圈相似。
4. 把 **distance** 設為“取得量度距離使用單位厘米 trig 接口 P14 echo 接口 P15”。
5. 把 **light** 變數設為“取得亮度傳感器數值接口 P0”。
6. 如果有車輛經過(**distance** ≤ 5)，
7. 而停車場內有空位時(**light** > 20)，
8. 打開停車場閘杆（即轉動舵機到 90 度），暫停 5 秒使閘杆打開 5 秒。然後把閘杆關上（即轉動舵機到 0 度）。

● 智能垃圾桶



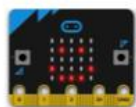
3. 智能垃圾桶

程度：★ ★ ☆ ☆ ☆

介紹

垃圾桶上的 LED 可以告訴人們桶內的當前容量，收集工人可以輕易判斷有否需要清理垃圾。這樣可以減少更換垃圾袋的次數，從而減少膠袋用量，促進環保城市。

所用部件



micro:bit (1)



IoT:bit (1)



彩色LED (WS2812) (1)



超音波距離傳感器(1)



三腳模組連接線 (1)



四腳模組連接線 (1)



M4*10 毫米螺絲 (6)



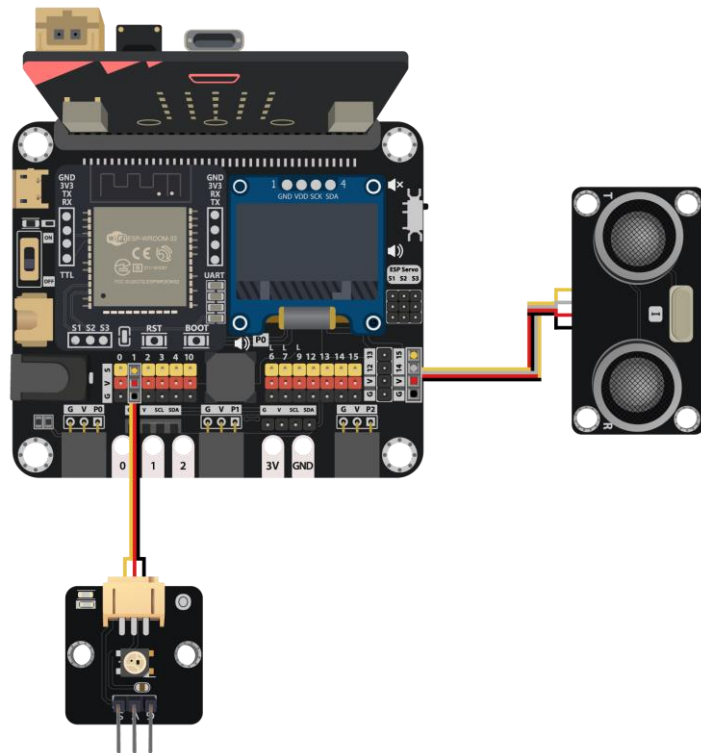
M4 螺母 (6)



螺絲批 (1)

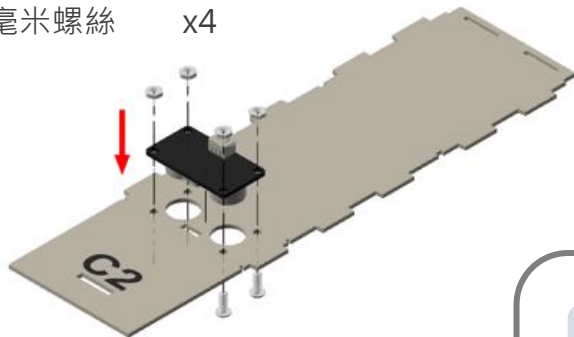


C1-C2 模型(1)

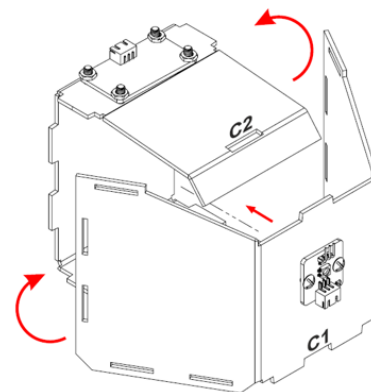
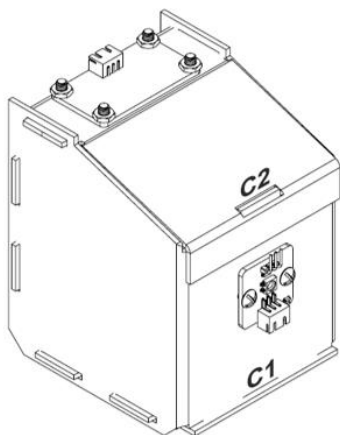
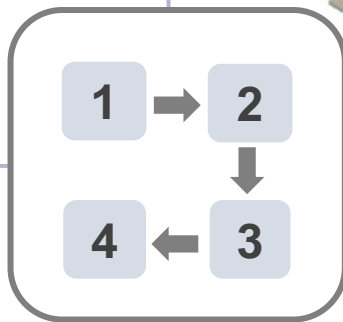
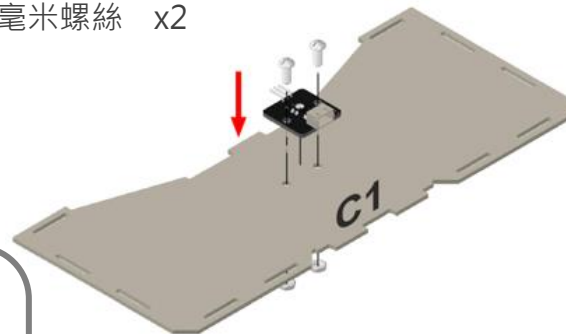


組裝步驟

C2 模型 x1
超音波距離傳感器 x1
M4 螺母 x4
M4*10毫米螺絲 x4



C1 模型 x1
彩色LED x1
M4 螺母 x2
M4*10毫米螺絲 x2



編程

💡 如何加入音效提示滿載？（例如：利用蜂鳴器）

1 當啟動時

2 變數 distance 設為 0

變數 strip 設為 NeoPixel at pin P1 with 1 leds as RGB (GRB format)

3 strip set brightness 50

4 暫停 5000 毫秒

5 重複無限次

6 變數 distance 設為 取得量度距離 使用單位 厘米 trig接口 P14 echo接口 P15

7 如果 distance \leq 4 那麼

strip show color red

8 否則

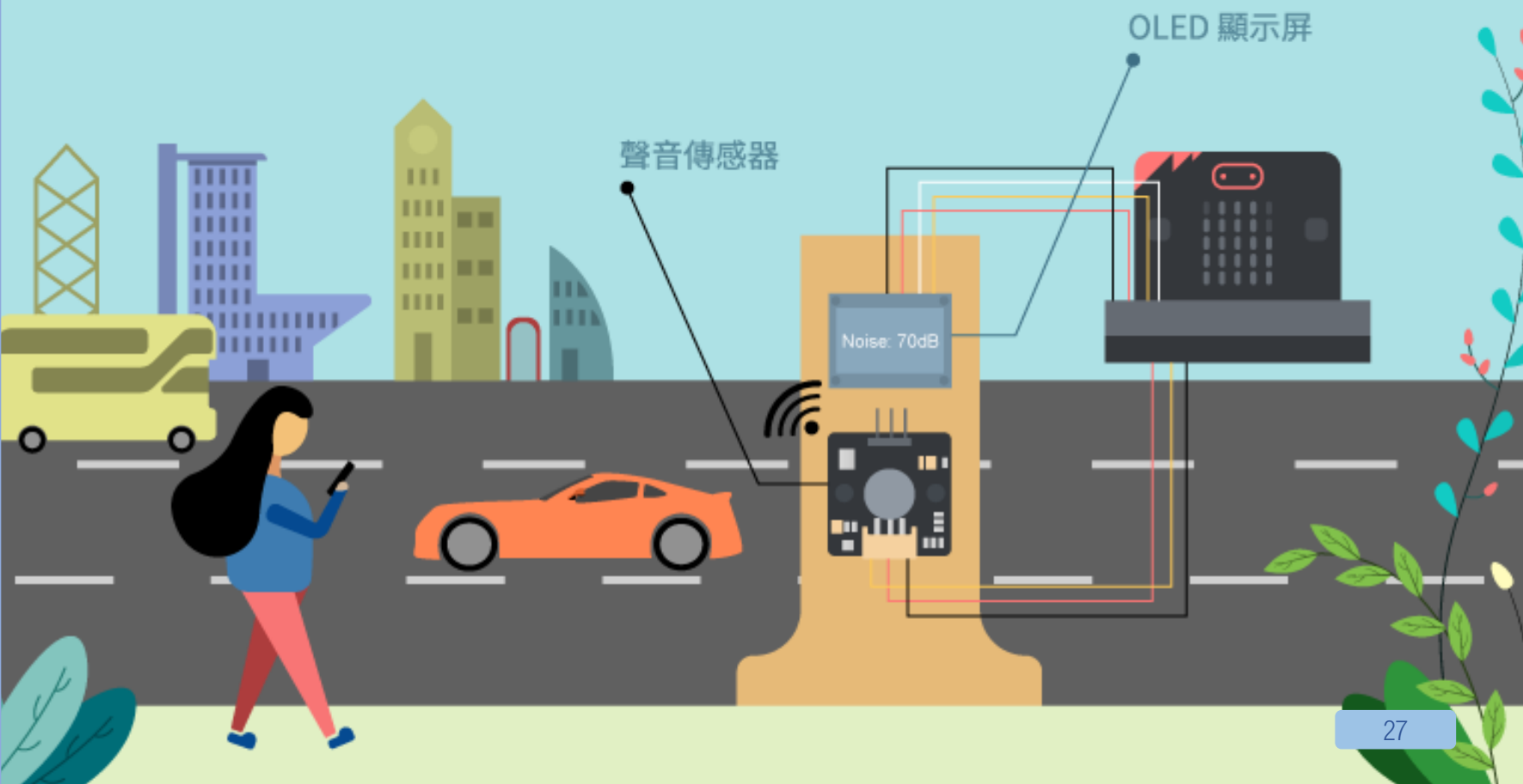
strip show color green

9 暫停 1000 毫秒

步驟：

1. 「當啟動時」只會在程式最開始時運作。
2. 將變數 **distance** 設為 **0**，並且將變數 **strip** 設為 “NeoPixel at pin P1 with 1 leds as RGB (GRB format)”。
3. 將 **strip** 的亮度設為 **50**。
4. 暫停 **5 秒**。
5. 「重複無限次」的運作方式和迴圈相似。
6. 把 **distance** 設為 “取得量度距離使用單位厘米 trig 接口 P14 echo 接口 P15”。
7. 如果 **distance** \leq **4**，strip 會顯示**紅色**。
8. 否則，strip 會顯示**綠色**。
9. 暫停**一秒**，顯示出相應的顏色。

● 城市噪音檢測器



4. 城市噪音檢測器

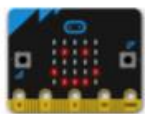
程度：★ ★ ☆ ☆ ☆

介紹

城市噪音檢測器能監測公路上的噪音污染，因為過大的噪音會降低周遭人們的生活水平。

安裝一個檢測器可以幫助工程師收集有關數據用於城市規劃。

所用部件



micro:bit (1)



IoT:bit (含 OLED 顯示屏) (1)



聲音傳感器 (1)



三腳模組連接線 (1)



四腳延長線 (1)



M2*10毫米螺絲 (2)



M2 螺母 (2)



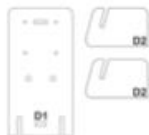
M4*10毫米螺絲 (4)



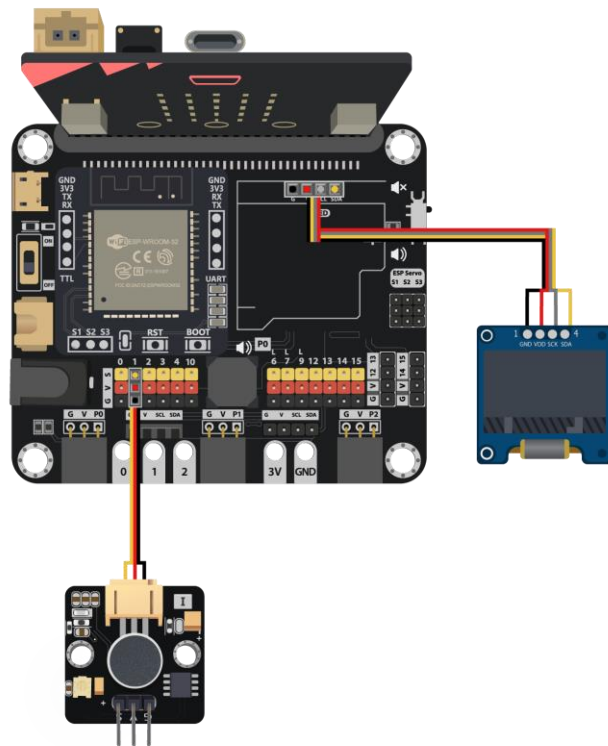
M4 螺母(4)



螺絲批 (1)

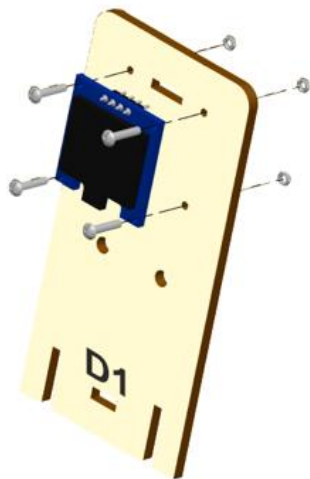


D1-D2 模型 (1)

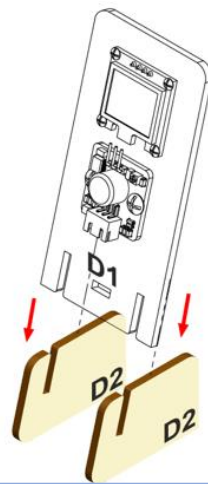
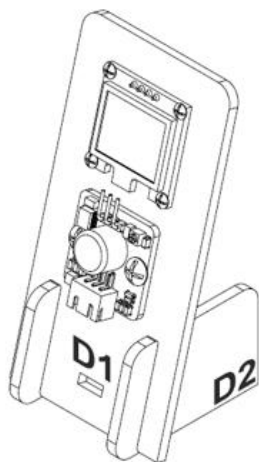
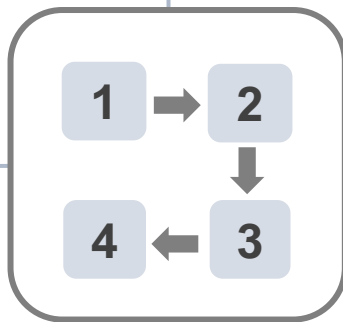
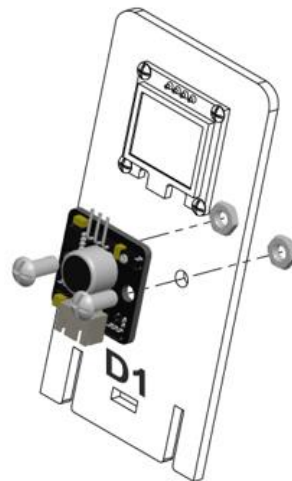


組裝步驟

- D1 模型 x1
- OLED 顯示屏 x1
- M2 螺母 x4
- M4*10毫米螺絲 x4



- 聲音傳感器 x1
- M4 螺母 x2
- M4*10毫米螺絲 x2



- D2 模型 x2

編程



當噪音太大可以如何發出提示？（例如：將LED轉為紅色）

1 當啟動時

2 OLED 初始化 寬 128 高 64

3 變數 Noise 設為 0

4 重複無限次

5 變數 Noise 設為 四捨五入 取得 聲音傳感器數值 接口 P1

6 點亮長條圖 顯示值為 Noise 最大值為 100

7 清除顯示

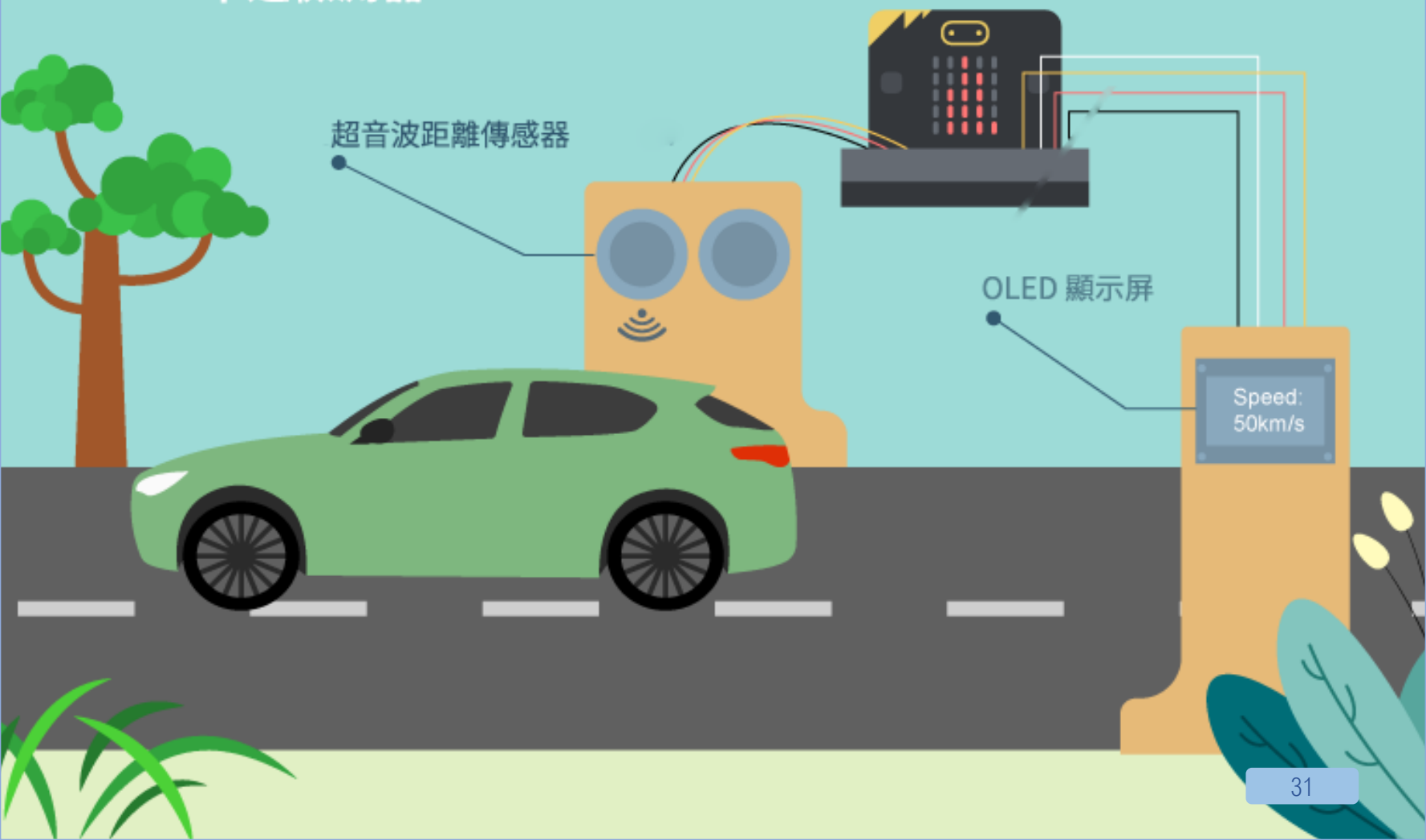
顯示字符串 字符串組合 "Noise: " Noise "dB" - +

8 暫停 500 毫秒

步驟:

1. 「當啟動時」只會在程式最開始時運作。
2. 初始化 OLED（寬128 高64）
3. 宣告變數 **Noise** 設為0。
4. 「重複無限次」的運作方式和迴圈相似。
5. 將變數 **Noise** 設為四捨五入取得聲音傳感器數值接口 P1。
6. 在「燈光」模塊中加入點亮長條圖顯示值為 **Noise**，最大值為100。
7. 加入清除顯示，之後加入顯示字符串字符串組合（"Noise:" & '**Noise**' & "dB"）。
8. 暫停0.5秒，顯示與噪音等級相對應的圖表。

● 車速檢測器



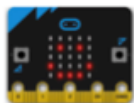
5. 車速檢測器

程度：★ ★ ★ ☆ ☆

介紹

車速檢測器是一個檢測系統，根據車輛在一定時間內移動距離得出其車速。安裝檢測系統用以檢控有效減少交通意外發生。

所用部件



micro:bit (1)



IoT:bit (含 OLED 顯示屏) (1)



超音波距離傳感器(1)



四腳模組連接線 (1)



四腳延長線 (1)



M2*10 毫米螺絲 (4)



M2 螺母 (4)



M4*10毫米螺絲(4)



M4 螺母 (4)



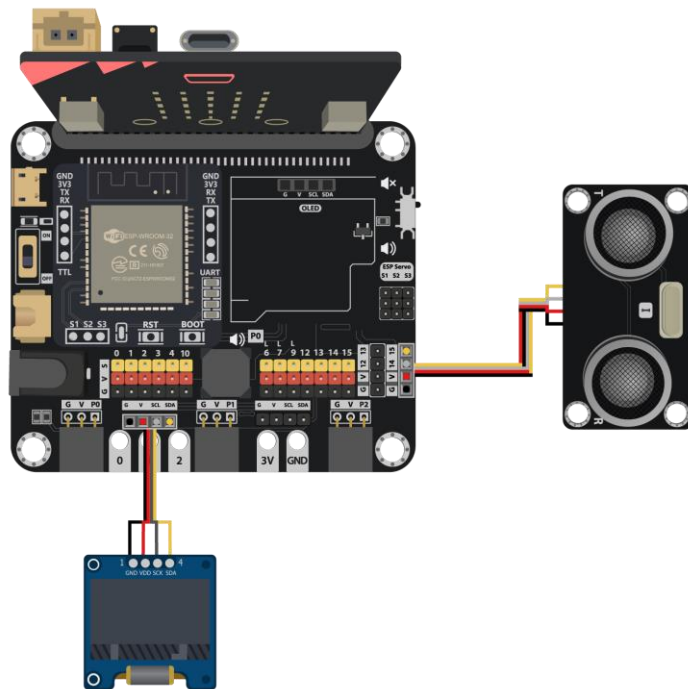
螺絲批 (1)



D1-D2 模型 (1)

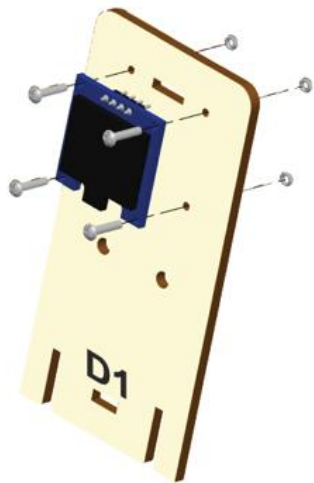


E1-E2 模型 (1)

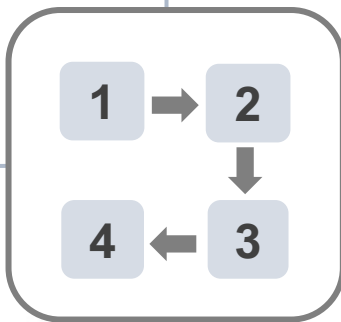
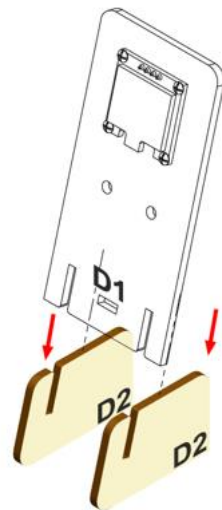


組裝步驟

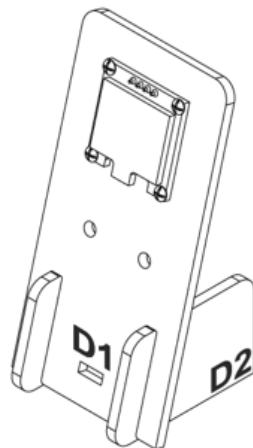
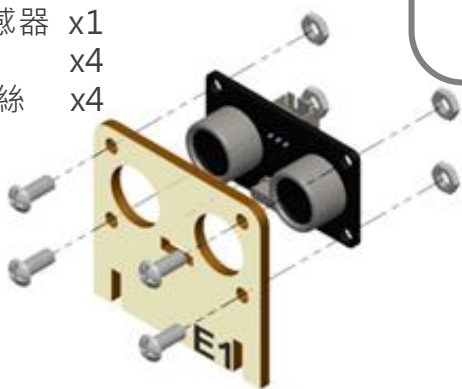
- D1 模型 x1
- OLED顯示屏 x1
- M2 螺母 x4
- M2 *10毫米螺絲 x4



- D2 模型 x2

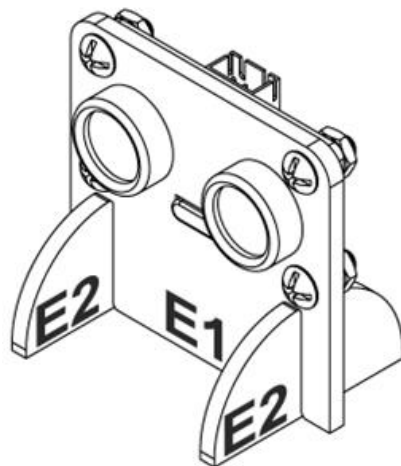
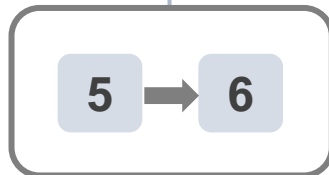
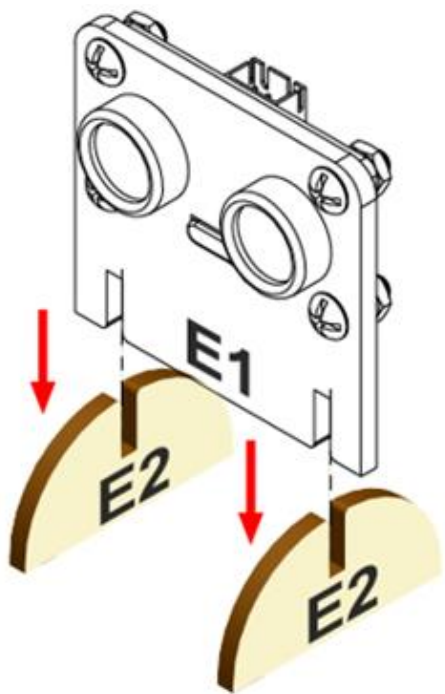


- E1 模型 x1
- 超音波距離傳感器 x1
- M4 螺母 x4
- M4 *10毫米螺絲 x4



組裝步驟

E2 模型 x2



編程



我們可以如何設定音效提示以警示超速？

1 當啟動時

2 OLED 初始化 寬 128 高 64

3 變數 distance1 設為 0
變數 distance2 設為 0
變數 speed 設為 0

4 重複無限次

5 呼叫 calculate_Speed

6 如果 speed \geq 0 那麼

7 點亮長條圖 顯示值為 speed
最大值為 20

8 清除顯示

9 顯示字符串 字串組合 "Distance1:" distance1
顯示字符串 字串組合 "Distance2:" distance2
顯示字符串 字串組合 "Speed:" speed

步驟:

1. 「當啟動時」只會在程式最開始時運作。
2. 初始化 OLED (寬128 高64)。
3. 宣告變數 **distance1** , **distance2** , **speed** 設為0。
4. 「重複無限次」的運作方式和迴圈相似。
5. 呼叫新函式 calculate_Speed (用於取得 distance1 , distance2 和 speed 的數值)。
6. 如果有車靠近 (**speed \geq 0**) ,
7. 在micro:bit LED的邏輯中加入點亮長條圖顯示值為 speed 最大值為 20。
8. 清除顯示。
9. 在OLED顯示屏上展示字符串
新行顯示字符串 (Distance1: **distance1**) , (Distance2: **distance2**) , (Speed: **speed**) 。



The image shows a Scratch code editor with four numbered steps (10-14) for defining a function named 'calculate_Speed'. Step 10 is a 'define function' block. Step 11 is a 'set variable to value' block for 'distance1', with the value being 'get distance from trig interface P14 echo interface P15 in centimeters'. Step 12 is a 'wait for duration' block set to 500 milliseconds. Step 13 is another 'set variable to value' block for 'distance2', with the same value as step 11. Step 14 is a 'set variable to value' block for 'speed', with the value being the mathematical expression $(distance1 - distance2) / 0.5$.

步驟:

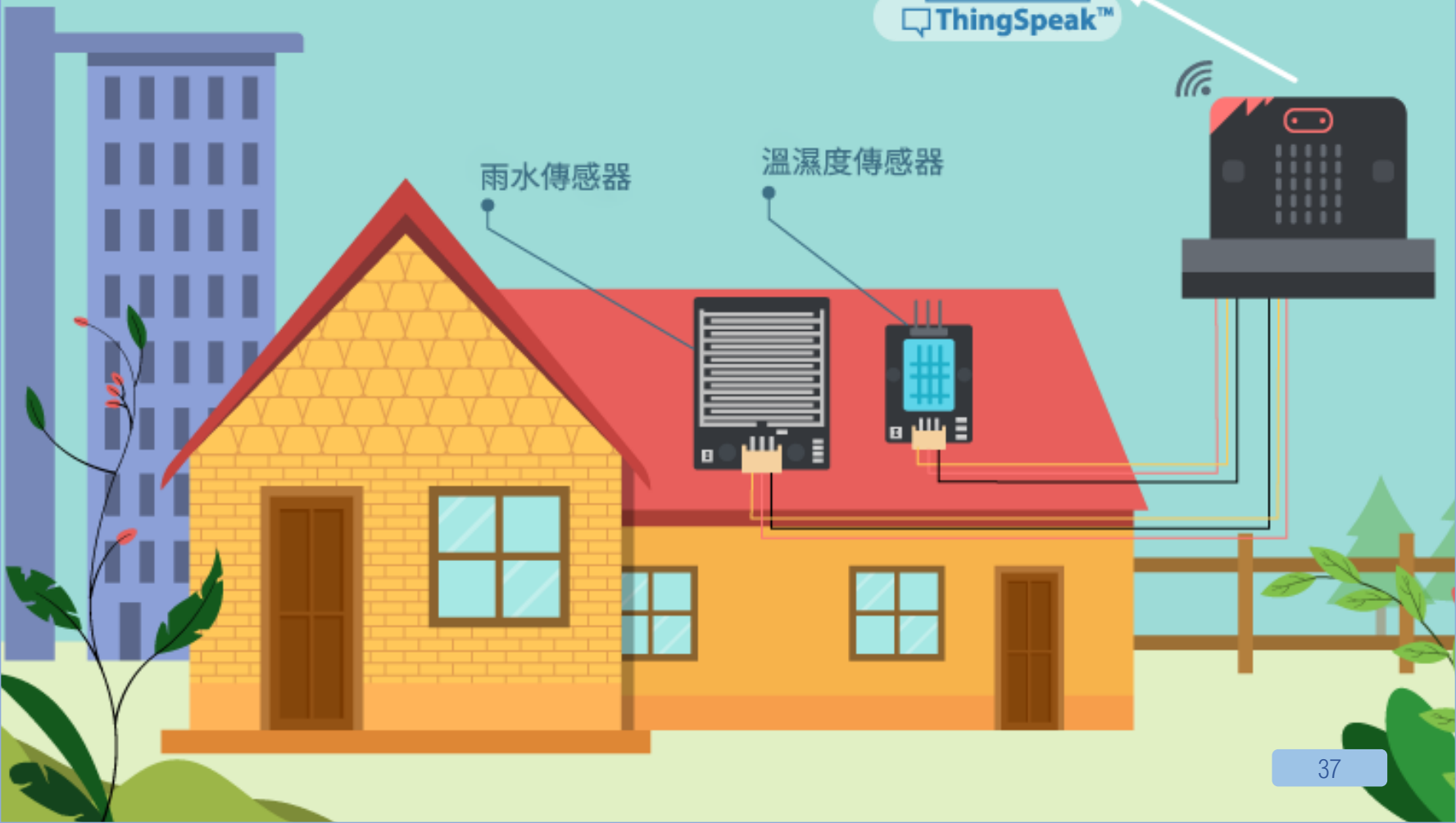
10. 宣告函式 (`calculate_Speed`) 。
11. 把 `distance1` 設為 “取得量度距離使用單位厘米 trig 接口 P14 echo 接口 P15” 。
12. 暫停0.5秒以計算下一個距離 。
13. 把 `distance2` 設為 “取得量度距離使用單位厘米 trig 接口 P14 echo 接口 P15” 。
14. 把 `speed` 設為 $(distance1 - distance2) / 0.5$ 。

● 智慧天氣監測站



雨水傳感器

溫濕度傳感器

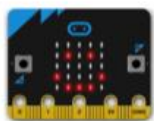


6.[物聯網]智慧天氣監測站

程度：★ ★ ☆ ☆ ☆

介紹
製作一個能收集雨水、溫度及濕度傳感器數據的智慧天氣監測站，並自動把數據送上ThingSpeak 平台。

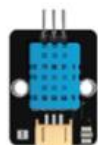
所用部件



micro:bit (1)



IoT:bit(含 OLED 顯示屏) (1)



溫濕度傳感器 (1)



雨水傳感器 (1)



三腳模組連接線 (2)



M4*10毫米螺絲(4)



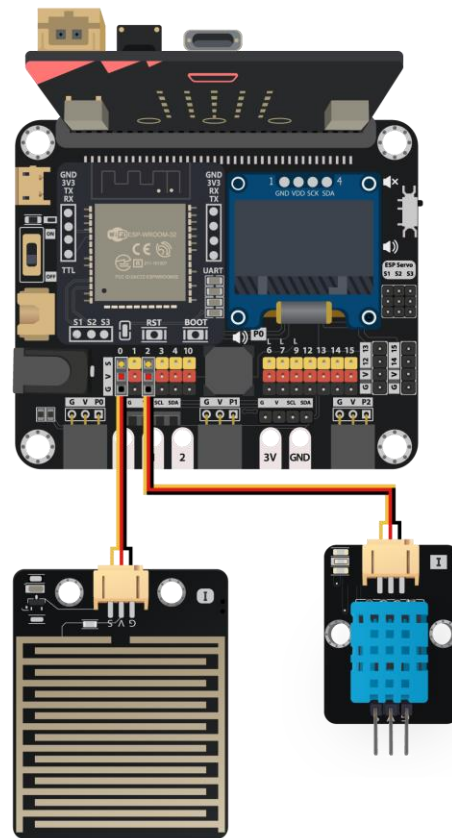
M4 螺母 (4)



螺絲批 (1)



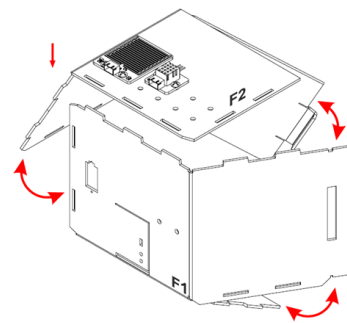
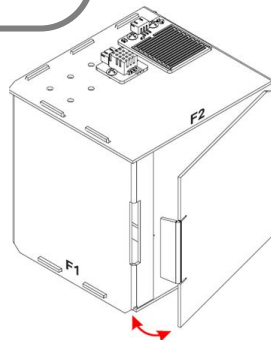
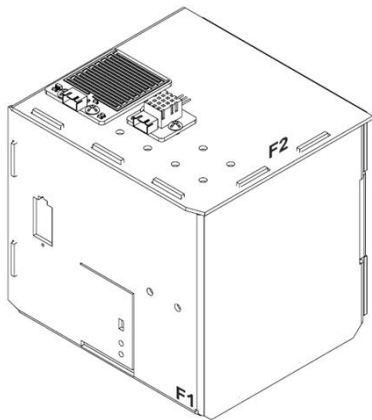
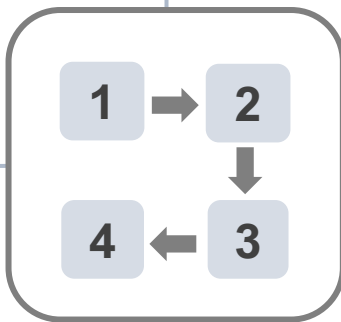
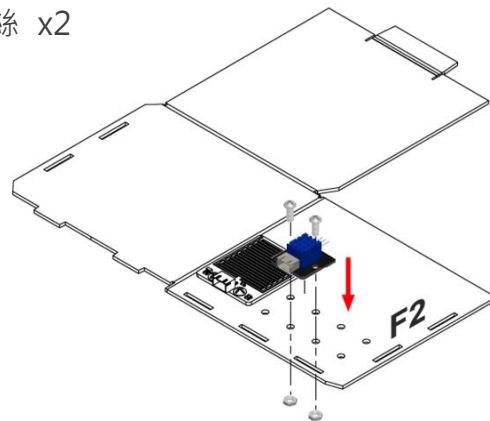
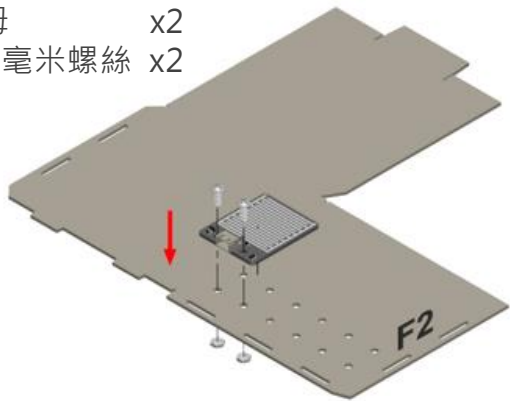
F1-F2 模型 (1)



組裝步驟

- F2 模型 x1
- 雨水傳感器 x1
- M4螺母 x2
- M4*10毫米螺絲 x2

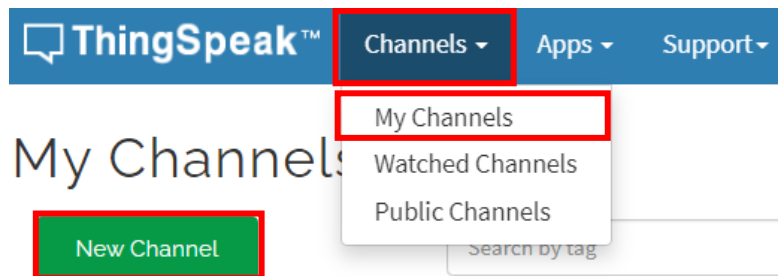
- 溫濕度傳感器 x1
- M4螺母 x2
- M4*10毫米螺絲 x2



F1 模型 x1

IOT (物聯網) – THINGSPEAK

步驟一. 訪問 <https://thingspeak.com> , 選擇 Channels -> My Channels -> New Channel .



步驟二. 輸入 Channel name , Field1 , Field2 , 點擊 Save Channel

- Channel name: Smart Weather Station
- Field 1: temperature
- Field 2: humidity
- Field 3: raindrop



<https://smarthon-docs-en.readthedocs.io/en/latest/smartcity/chapter1.html>

IOT (物聯網) – THINGSPEAK

步驟三. 選擇 your channel >> API Keys , 複製 API key 。

Private View Public View Channel Settings Sharing **API Keys**

1. 選擇 API Keys

Write API Key

Key **5G4I...89V**

Generate New Write API Key

Help

API keys enable you to

2. 複製 API key

API Keys Set

- Write API Key: I
been compron
- Read API Key:



<https://smarthon-docs-en.readthedocs.io/en/latest/smartcity/chapter1.html>

編程



我們可以怎上傳其他種類的數據（例如：噪音）至 ThingSpeak？

1 當啟動時

2 OLED 初始化 寬 128 高 64

3 初始化IoT:bit TX P16 RX P8

4 連接到WiFi 名稱 "smarthon" 密碼 "12345678"

5 變數 raindrop 設為 0
變數 humidity 設為 0
變數 temperature 設為 0

6 當WiFi連接成功後 IP_Address Device_ID

7 顯示 圖示

步驟：

1. 「當啟動時」只會在程式最開始時運作。
2. 初始化 OLED（寬128 高64）。
3. 初始化 IoT:bit（TX P16 RX P8）。
4. 設置WiFi 並輸入WiFi名稱及密碼。
5. 宣告新變數 **temperature**、**humidity** 和 **raindrop** 並設為 0。
6. 在連上WiFi後，
7. 在OLED屏幕上顯示圖示「別號」。

編程



我們可以怎上傳其他種類的數據（例如：噪音）至 ThingSpeak？

8 重複無限次

9 如果 WiFi連接狀態 那麼

10 變數 humidity 設為 DHT11 讀取 濕度 接口 P2
變數 temperature 設為 DHT11 讀取 溫度 接口 P2
變數 raindrop 設為 取得 雨水傳感器數值 接口 P0

11 清除顯示

12 顯示字符串 字串組合 "溫度:" temperature
顯示字符串 字串組合 "濕度:" humidity
顯示字符串 字串組合 "雨水:" raindrop

13 發送到 Thingspeak Key* 5G4...9V
項目1數值 temperature
項目2數值 humidity
項目3數值 raindrop

14 暫停 15000 毫秒

把 Thingspeak key 粘貼在此

步驟:

8. 「重複無限次」的運作方式和迴圈相似。
9. 加入「如果WiFi已連接，那麼」。
10. 設變數 **humidity** 為DHT11 讀取濕度接口 P2，**temperature** 為DHT11 讀取溫度接口 P2。
11. 設變數 **raindrop** 為取得雨水傳感器數值接口 P0。
12. 清除 OLED 顯示。
13. 在OLED屏幕上顯示字符串：字串組合（「溫度」，**temperature**），（「濕度」，**humidity**），（「雨水」，**raindrop**）。
14. 把資料傳送至 ThingSpeak。
15. 暫停15秒。

編程

💡 我們可以怎上傳其他種類的數據（例如：噪音）至 ThingSpeak？



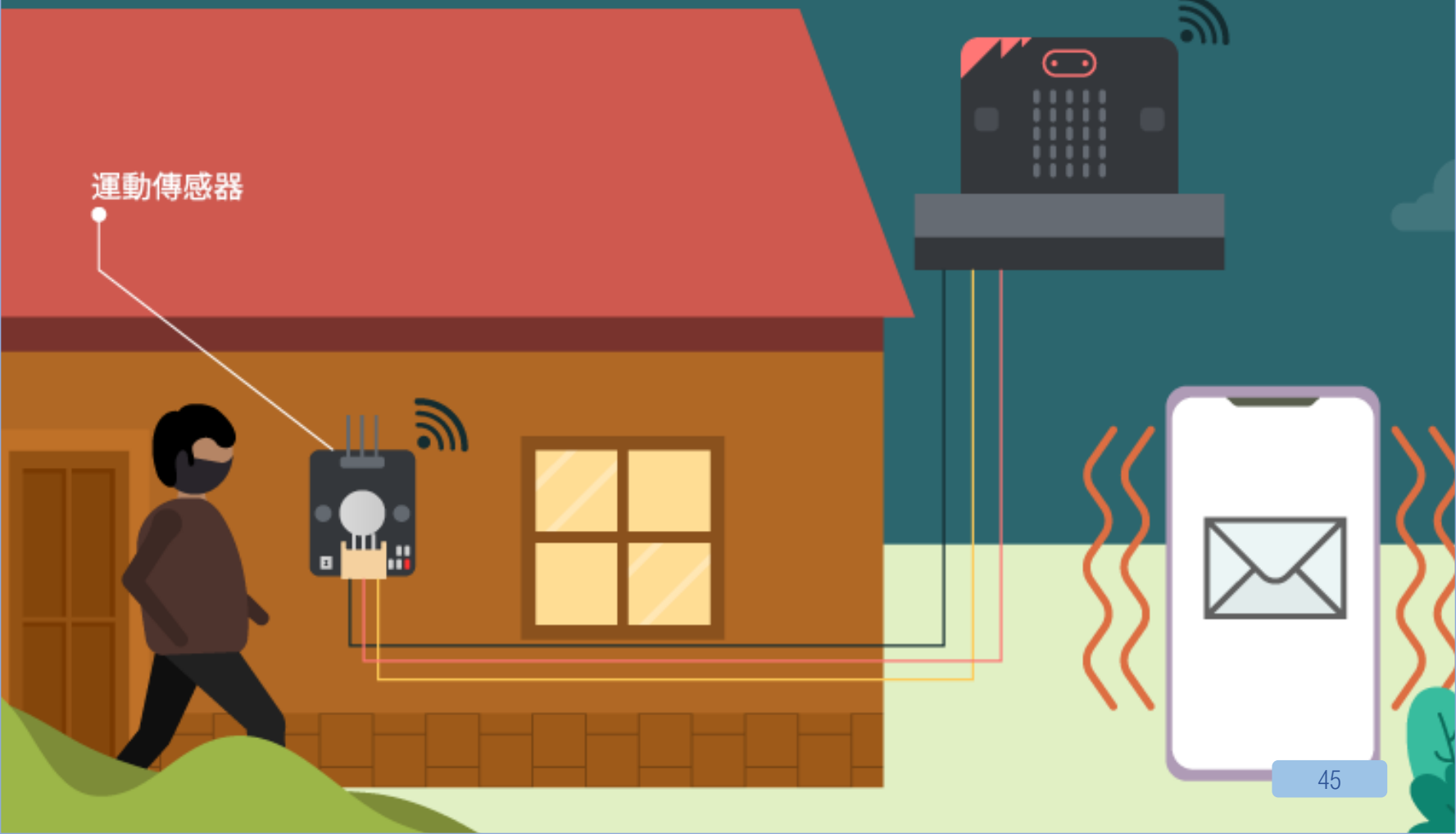
步驟:

16. 當上傳到 Thingspeak 後
17. 清除OLED顯示。
18. 在OLED屏幕上顯示字符串：字串組合（“ThingSpeak” ， Status ），（“錯誤碼” ， Error_code ）。

● 智能家居防盜系統

IFTTT

運動傳感器



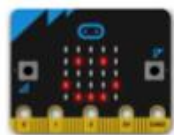
7. [物聯網]智能家居防盜系統

程度：★ ★ ★ ☆ ☆

介紹

運動傳感器回饋訊號至主板，若 TRUE 則主板激活蜂鳴器並透過IFTTT自動發出電郵提示用家。同時 OLED 顯示屏會反映有可疑人士。

所用部件



micro:bit (1)



IoT:bit (含 OLED 顯示屏) (1)



運動傳感器 (1)



三腳模組連接線 (1)



M4*10毫米螺絲 (2)



M4 螺母 (2)

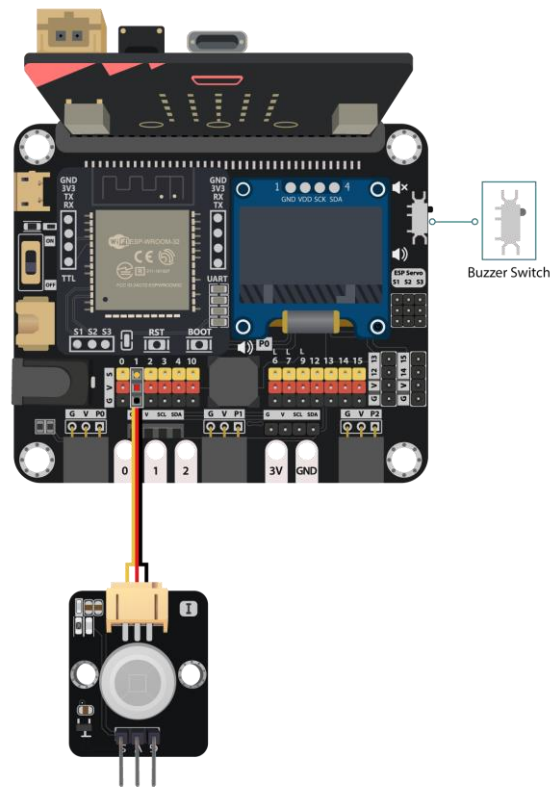


螺絲批(1)



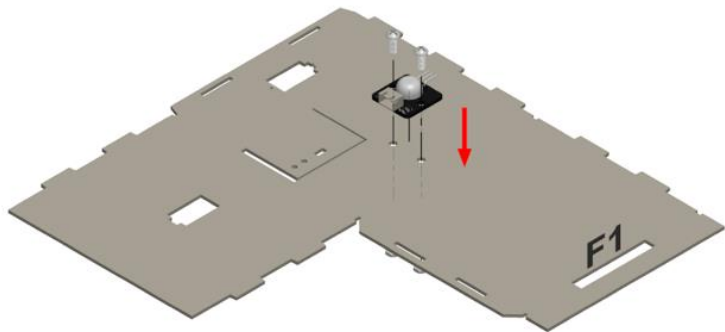
F1-F2 模型 (1)

IFTTT

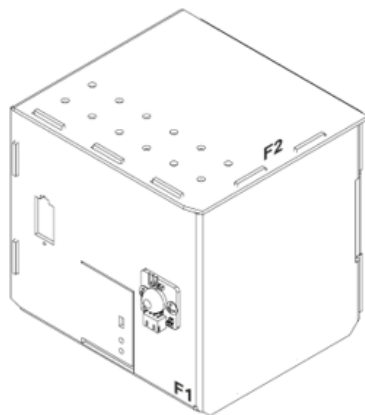
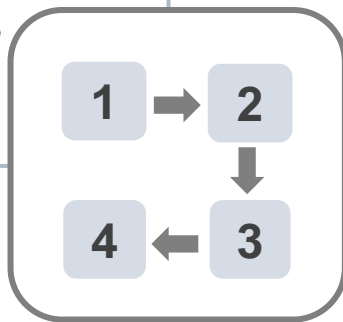
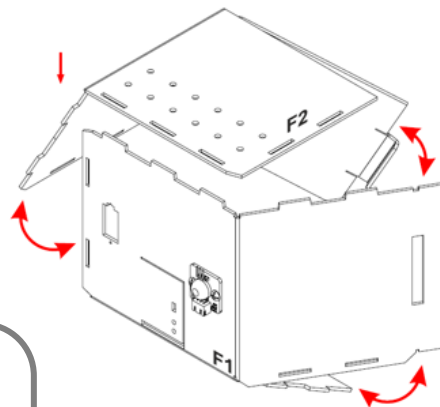


組裝步驟

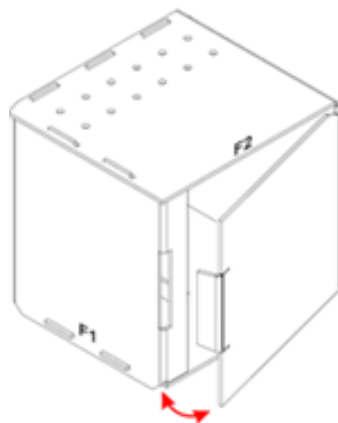
- F1 模型 x1
- 運動傳感器 x1
- M4 螺母 x2
- M4*10毫米螺絲 x2



- F2 模型 x1

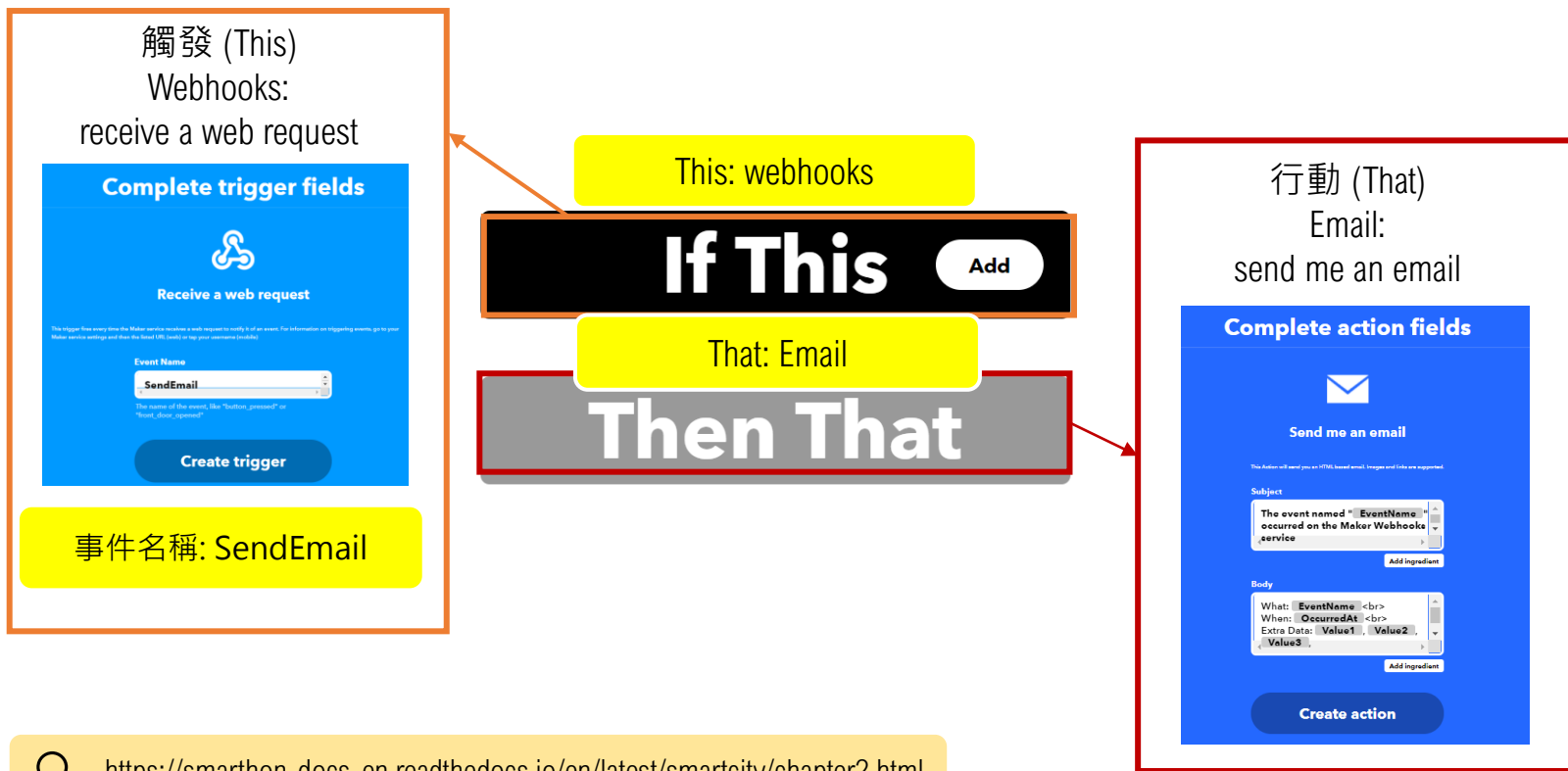


- F2 模型 x1



IOT (物聯網) – IFTTT

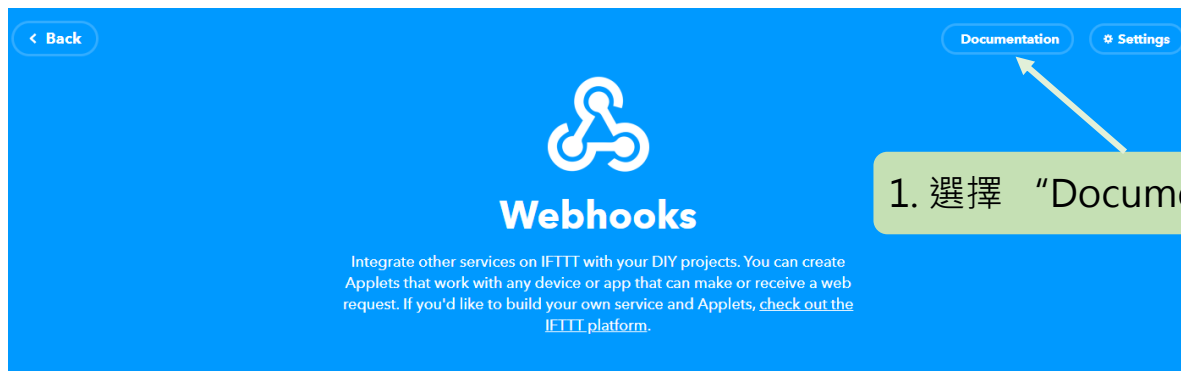
步驟 1. 訪問 <https://ifttt.com> , 創建新專案 (if webhooks then Email) 。



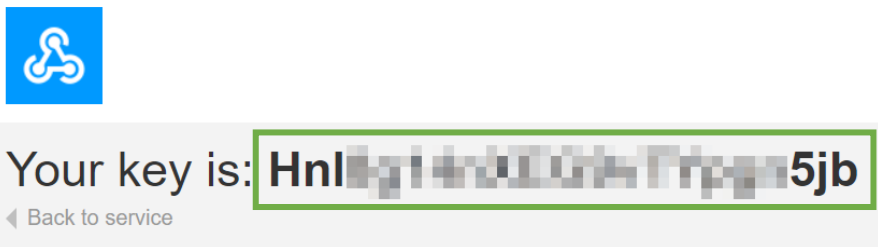
<https://smarthon-docs-en.readthedocs.io/en/latest/smartcity/chapter2.html>

IOT (物聯網) – IFTTT

步驟 2. 依次點擊 “My services” >> “Webhooks” , select “Documentation”
· 複製 Applet Key 。



1. 選擇 “Documentation”



2. 複製這個 IFTTT key



<https://smarthon-docs-en.readthedocs.io/en/latest/smartcity/chapter2.html>

編程



如何避免短時間內重複發送大量電郵？（提示：利用變數功能？）

1 當啟動時

2 OLED 初始化 寬 128 高 64

3 初始化IoT:bit TX P16 RX P8

4 連接到WiFi 名稱 "smarthon" 密碼 "12345678"

5 重複無限次

6 如果 WiFi連接狀態 那麼

7 如果 取得 動作傳感器數值 接口 P1 = true 那麼

8 演奏 音階 中音 C 持續 1 拍

顯示 圖示 smile

9 發送到 IFTTT key* Hn1... jb

事件名稱* "SendEmail"

10 否則

顯示 圖示 smile

11 暫停 1000 毫秒

把 Thingspeak key 粘貼在此

步驟:

1. 「當啟動時」只會在程式最開始時運作。
2. 初始化 OLED（寬128 高64）。
3. 初始化 IoT:bit（TX P16 PX P8）。
4. 設置WiFi 並輸入WiFi名稱及密碼。
5. 「重複無限次」的運作方式和迴圈相似。
6. 加入“如果WiFi已連接，那麼”。在上述邏輯再加一個“如果...那麼”，以“取得運動傳感器數值接口 P1 = TRUE”為前設。
7. 在“如果...那麼”加入“演奏音階中音 C 持續 1 拍和LED 顯示“smile”圖示”。
8. 加入“發送到 IFTTT Key*”，填上 IFTTT Key 並把事件名稱設為“SendEmail”。
9. 在“否則”加入“smile”圖示（沒有可疑人士經過）。
10. 暫停一秒



11 當WiFi連接成功後 IP_Address Device_ID

12 顯示 圖示

13 當上傳到IFTTT後 Status Error_code

14 清除顯示

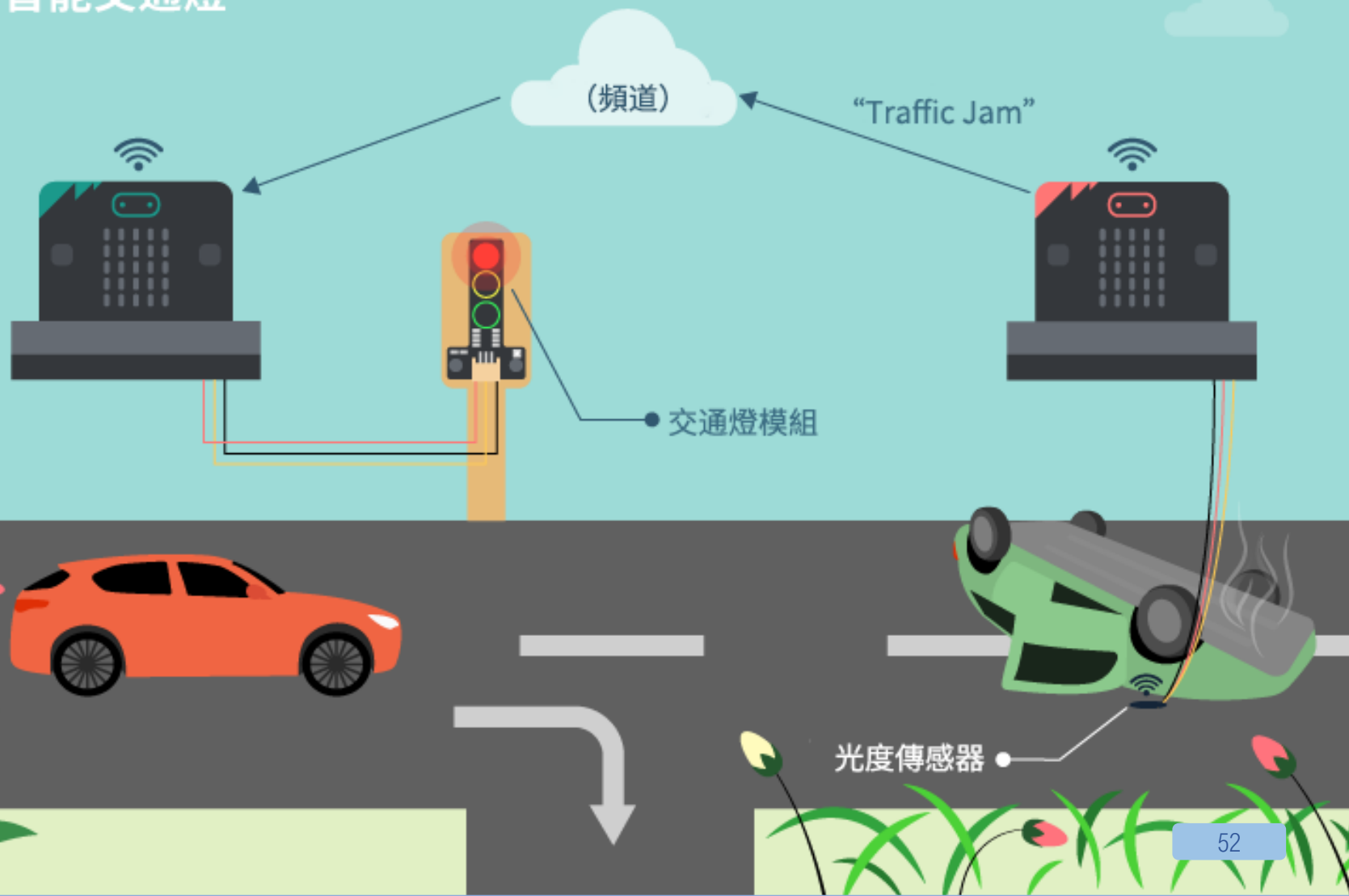
15 顯示字符串 字符串組合 "IFTTT:" Status

16 顯示字符串 字符串組合 "Error:" Error_code

步驟:

11. 如果WiFi連接成功後，
12. LED會顯示“別號”圖示。
13. 當上傳到 IFTTT 後
14. 清除OLED顯示。
15. 在OLED屏幕上顯示字符串：字串組合（“IFTTT:” ， Status）。
16. 在OLED屏幕上顯示字符串：（“Error:” ， Error_code）。

● 智能交通燈



8. [物聯網] 智能交通燈（發送者）

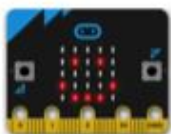
程度：★★★★★

介紹

當光度傳感器長期回饋過低數值，這代表有車停在此處，即公路上有塞車發生。發送者傳送“trafficjam”至 microbit。反之則發送“nojam”。

備註：這個案例需要兩組 micro:bit，一組為發送者，另一組為接收者。

所用部件



micro:bit (1)



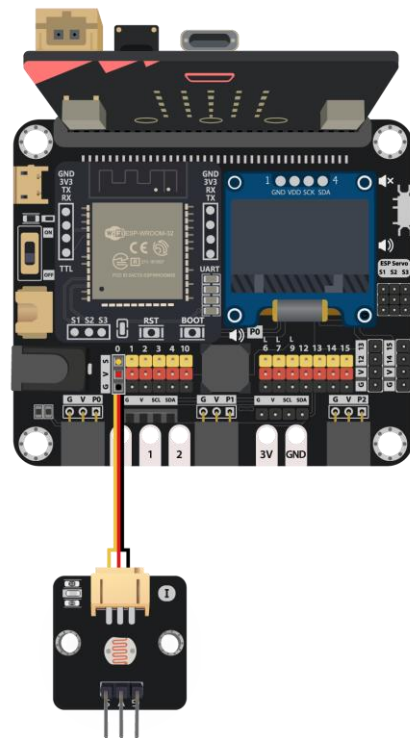
IoT:bit (含OLED 顯示屏) (1)



光度傳感器(1)



三腳模組連接線 (1)



編程



我們能如何使用距離傳感器偵測是否有塞車的情況？

1 當啟動時

2 OLED 初始化 寬 128 高 64

3 初始化IoT:bit TX P16 RX P8

4 連接到WiFi 名稱 "smarthon" 密碼 "12345678"

5 變數 light2 設為 0

6 當WiFi連接成功後 IP_Address Device_ID 顯示 圖示

步驟:

1. 「當啟動時」只會在程式最開始時運作。
2. 初始化 OLED (寬128 高64)。
3. 初始化 IoT:bit (TX P16 RX P8)。
4. 設置WiFi 並輸入WiFi名稱及密碼。
5. 宣告新變數 **light2** 並設為 0。
6. 在連上WiFi後，在OLED屏幕上顯示圖示「別號」。



```
7. 重複無限次
8. 如果 WiFi連接狀態 那麼
   變數 light2 設為 取得 亮度傳感器數值 接口 P0
9. 清除顯示
10. 顯示字符串 字串組合 "Light: " light2
11. 如果 light2 < 10 那麼
    傳送到頻道 "tsuenwan", 訊息為 "trafficjam"
12. 否則
    傳送到頻道 "tsuenwan", 訊息為 "nojam"
13. 暫停 6000 毫秒
```

步驟:

7. 「重複無限次」的運作方式和迴圈相似。
8. 加入如果Wifi已連接, 將變數 **light2** 設為取得光傳感器數值接口 P0。
9. 清除OLED顯示。
10. 顯示”字串組合 **Light: light2**”。
11. 如果 **light2** < 10, 傳送信息 “trafficjam” 到頻道 “tsuenwan”。
12. 否則傳送信息 “nojam” 到頻道 “tsuenwan”。
13. 暫停6秒。

8. [物聯網] 智能交通燈（接收者）

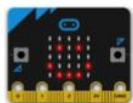
程度：★★★★★

介紹

當光度傳感器長期回饋過低數值，這代表有車停在此處，即公路上有塞車發生。發送者傳送“trafficjam”至 micro:bit。反之則發送“nojam”。

備註：這個案例需要兩組 micro:bit，一組為發送者，另一組為接收者。

所用部件



micro:bit (1)



IoT:bit(含OLED 顯示屏) (1)



交通燈模組 (1)



三腳模組連接線 (1)



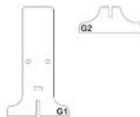
M4*10毫米螺絲(2)



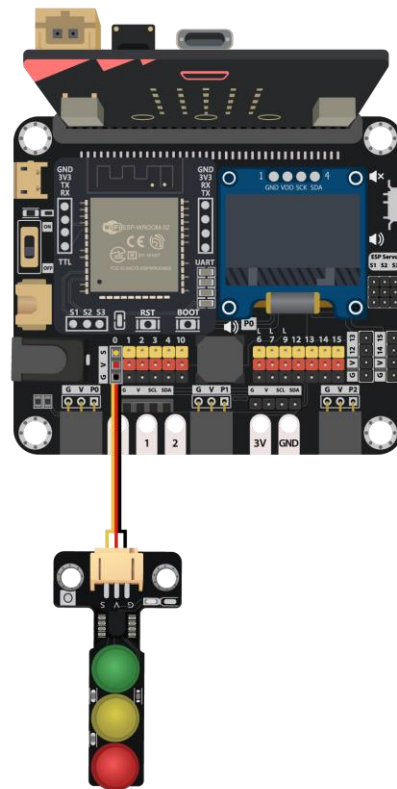
M4 螺母 (2)



螺絲批 (1)



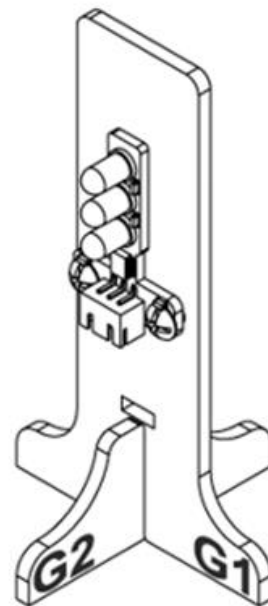
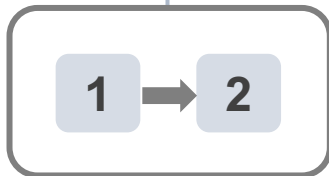
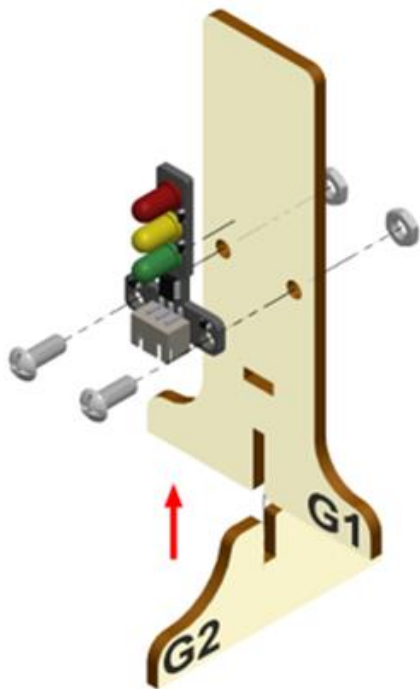
G1-G2 模型 (1)



 <https://smarthon-docs-en.readthedocs.io/en/latest/smartcity/chapter5.html>

組裝步驟

- G1-G2 模型 x1
- 交通燈模組 x1
- M4 螺母 x2
- M4*10毫米螺絲 x2

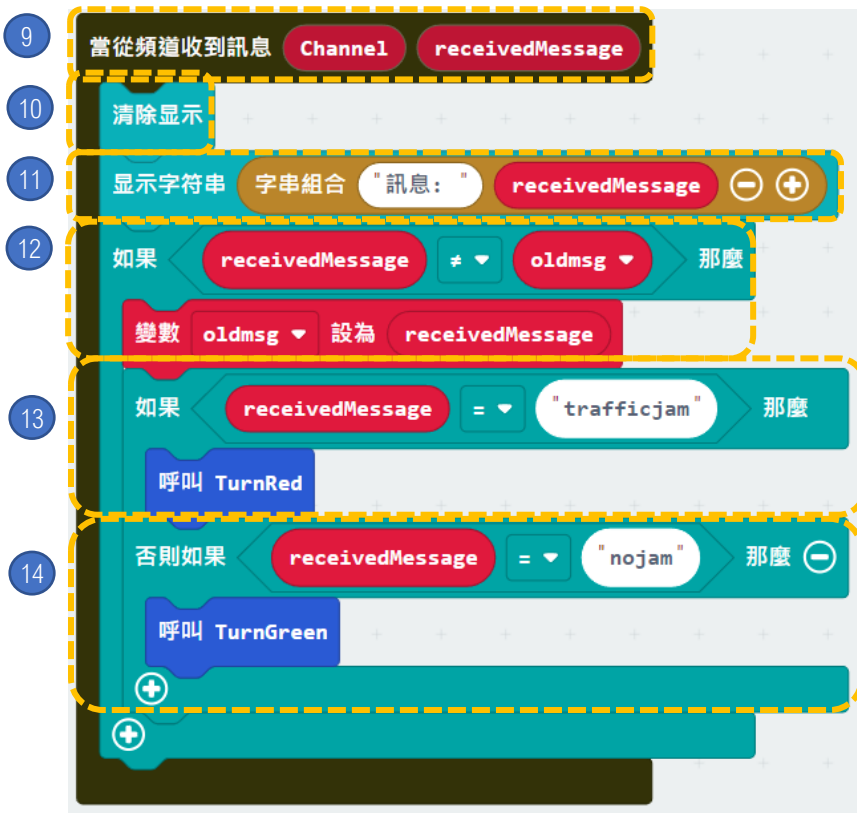




步驟:

1. 「當啟動時」只會在程式最開始時運作。
2. 初始化 OLED (寬128 高64)。
3. 初始化 IoT:bit (TX P16 PX P8)。
4. 設置WiFi 並輸入WiFi名稱及密碼。
5. 宣告新變數 **oldmsg** 並設為 ""。
6. 加入呼叫 TurnGreen。

7. 在連上WiFi後，在OLED屏幕上顯示圖示「別號」。
8. 加入到頻道 "tsuenwan"。



步驟:

9. 加入當收到WiFi頻道內容時，
10. 清除OLED顯示。
11. 顯示字符串字符串組合：“Message: , receivedMessage”。
12. 如果 receivedMessage \neq oldmsg，那麼把變數 oldmsg 設置成 receivedMessage。
13. 如果 receivedMessage 等於“trafficjam”，呼叫TurnRed。
14. 否則如果 receivedMessage 等於“nojam”，呼叫TurnGreen。



15

定義函式 TurnRed

交通燈控制

紅 關

黃 關

綠 開

接口 P1

暫停 2000 毫秒

交通燈控制

紅 開

黃 開

綠 關

接口 P1

暫停 2000 毫秒

交通燈控制

紅 開

黃 開

綠 關

接口 P1

暫停 2000 毫秒

定義函式 TurnGreen

交通燈控制

紅 開

黃 開

綠 開

接口 P1

暫停 2000 毫秒

交通燈控制

紅 關

黃 關

綠 關

接口 P1

暫停 2000 毫秒

交通燈控制

紅 開

黃 開

綠 開

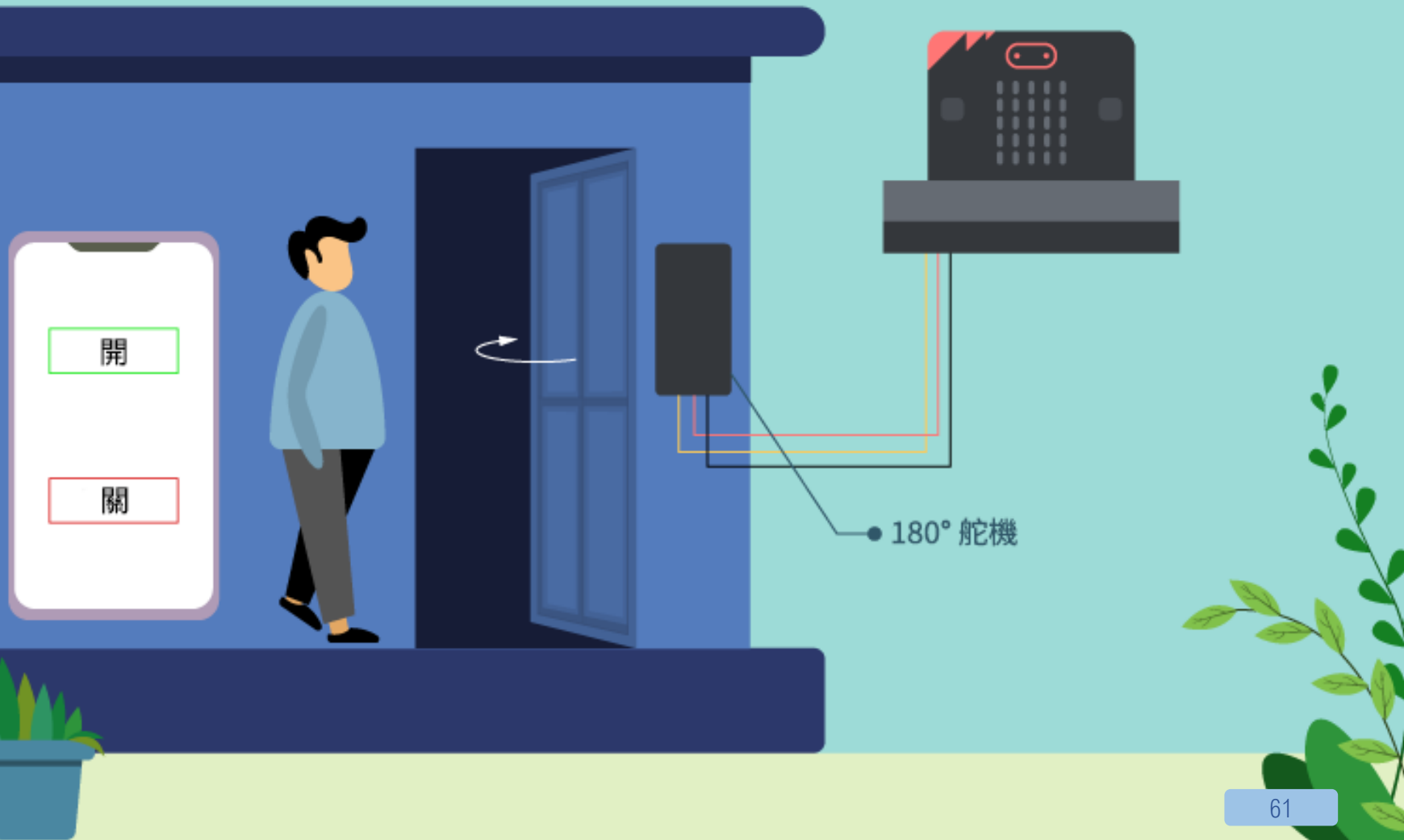
接口 P1

暫停 2000 毫秒

STEPS:

15. 宣告兩個新函式(TurnRed 及 TurnGreen)。
在TurnRed函式中加入“Control traffic light at P1”，等待兩秒，轉顏色為黃色，等待兩秒，轉顏色為紅，等待兩秒。在TurnGreen函式中加入“Control traffic light at P1”，轉為紅色，等待兩秒，轉為黃色，等待兩秒，轉為綠色，等待兩秒。

● 遙距智能門控制



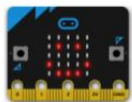
9. [物聯網] 遙距智能門控制

程度：★★★★☆



介紹
當 microbit 收到“ opendoor” 訊號，舵機會自動轉動。當收到“ closedoor” ，門會自動關閉。

所用部件



micro:bit (1)



IoT:bit (含OLED 顯示屏) (1)



180° 舵機 (1)



三腳模組連接線 (1)



舵機尖螺絲 M2*7.5毫米(1)
(於舵機套裝內)



M2*10毫米螺絲 (2)



M2 螺母 (2)



M3*10 毫米螺絲(2)



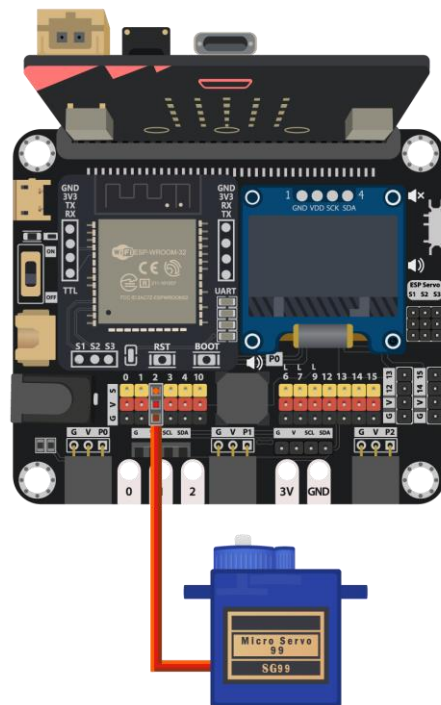
M3 螺母(1)



螺絲批 (1)

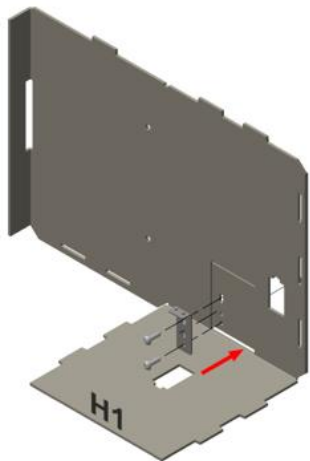


H1-H7模型 (1)

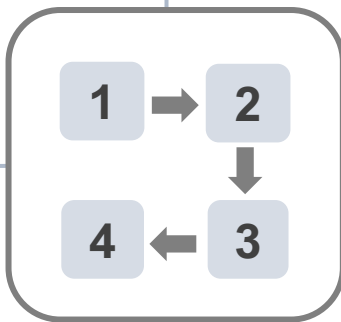
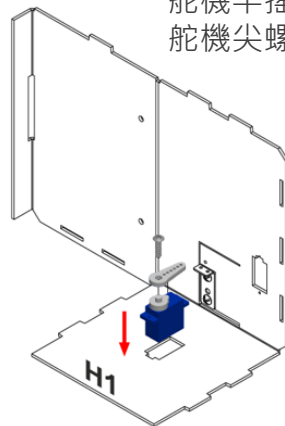


組裝步驟

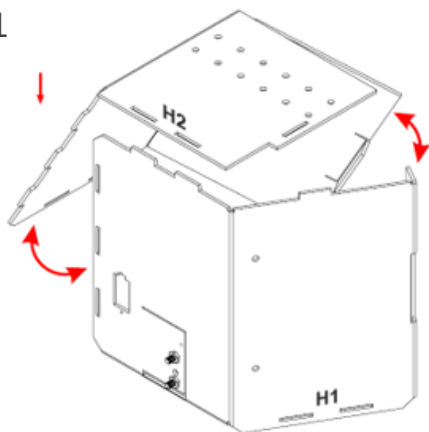
- L型支架 x1
- M4螺母 x2
- M4*10毫米螺絲 x2
- H1 模型 x1



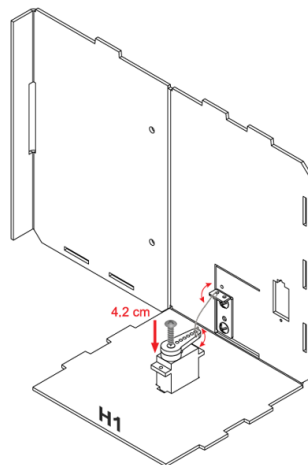
- 180°舵機 x1
- 舵機半搖臂 x1
- 舵機尖螺絲 M2*7.5毫米 x1



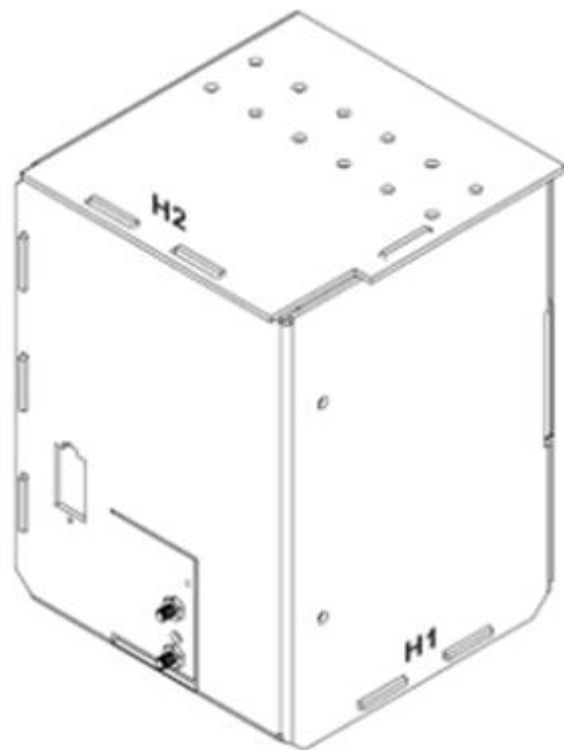
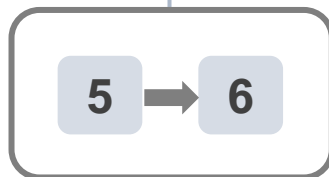
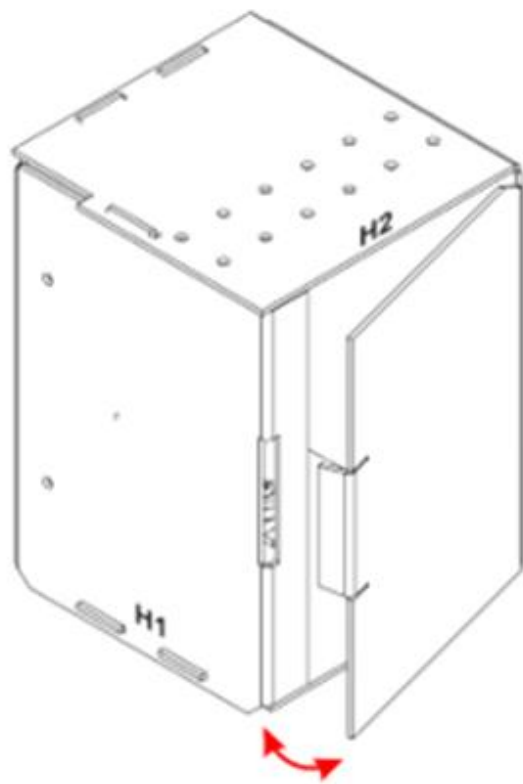
- H2 模型 x1



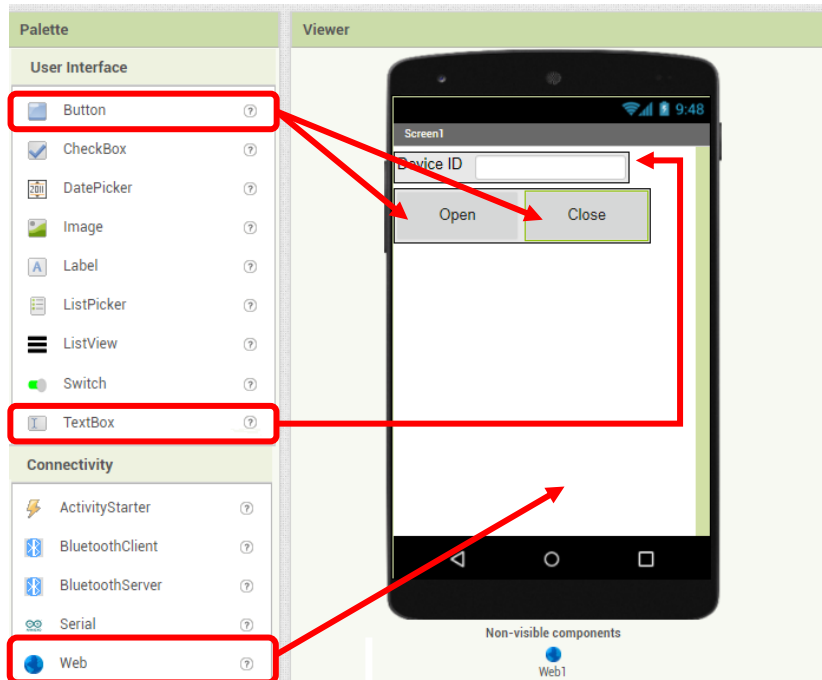
- 萬字夾 x1



組裝步驟



IOT (物聯網) – APP INVENTOR 2



步驟一. 創建新頁面

- On Designer:
- 在左邊界面拖出元素
- 把“ Web ” 拖到最下面



<https://smarthon-docs-en.readthedocs.io/en/latest/smartcity/chapter3.html>

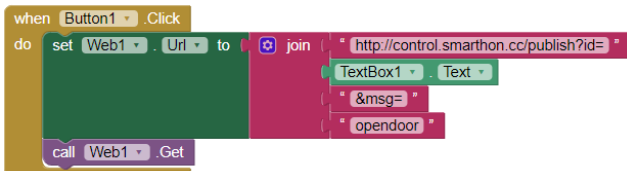
IOT (物聯網) – APP INVENTOR 2

編程二. 編程

- On Blocks:
- The WAN control command URL is:

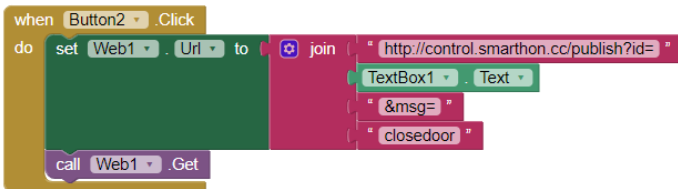
<http://control.smarthon.cc/publish?id=DeviceID&msg=ControlCommand>

- 當 button1 被點擊，以下指令“ opendoor” 會被執行。



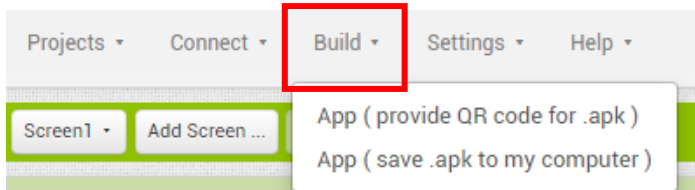
```
when Button1 Click
do
  set Web1 . Url to join ( "http://control.smarthon.cc/publish?id="
    TextBox1 . Text
    "&msg="
    "opendoor"
  )
  call Web1 . Get
```

- 當 button2 被點擊，以下指令“ closedoor” 會被執行



```
when Button2 Click
do
  set Web1 . Url to join ( "http://control.smarthon.cc/publish?id="
    TextBox1 . Text
    "&msg="
    "closedoor"
  )
  call Web1 . Get
```

- 透過QR code或 apk，下載程式至手機。





步驟:

1. 「當啟動時」只會在程式最開始時運作。
2. 初始化 OLED (寬128 高64)。
3. 初始化 IoT:bit (TX P16 PX P8)。
4. 設置WiFi 並輸入WiFi名稱及密碼。
5. 轉動舵機到 180 度 接口 P2。



6. 在連上WiFi後，在OLED屏幕上顯示圖示「別號」。
7. 顯示字符串 **Device_ID**。



8 當從互聯網接收到指令 WAN_Command

9 清除显示

10 显示字符串 字串組合 "Command: " WAN_Command

11 如果 WAN_Command = "opendoor" 那麼
轉動舵機到 45 度 接口 P2

12 否則如果 WAN_Command = "closedoor" 那麼
轉動舵機到 180 度 接口 P2

步驟：

8. 當從互聯網接收到指令時，
9. 清除OLED顯示。
10. 顯示字符串字串組合：“Command: ”，
WAN_Command”。
11. 如果 WAN_Command = “opendoor”，
轉動舵機到45度 接口P2。
12. 如果 WAN_Command = “closedoor”，
轉動舵機到180度 接口P2。

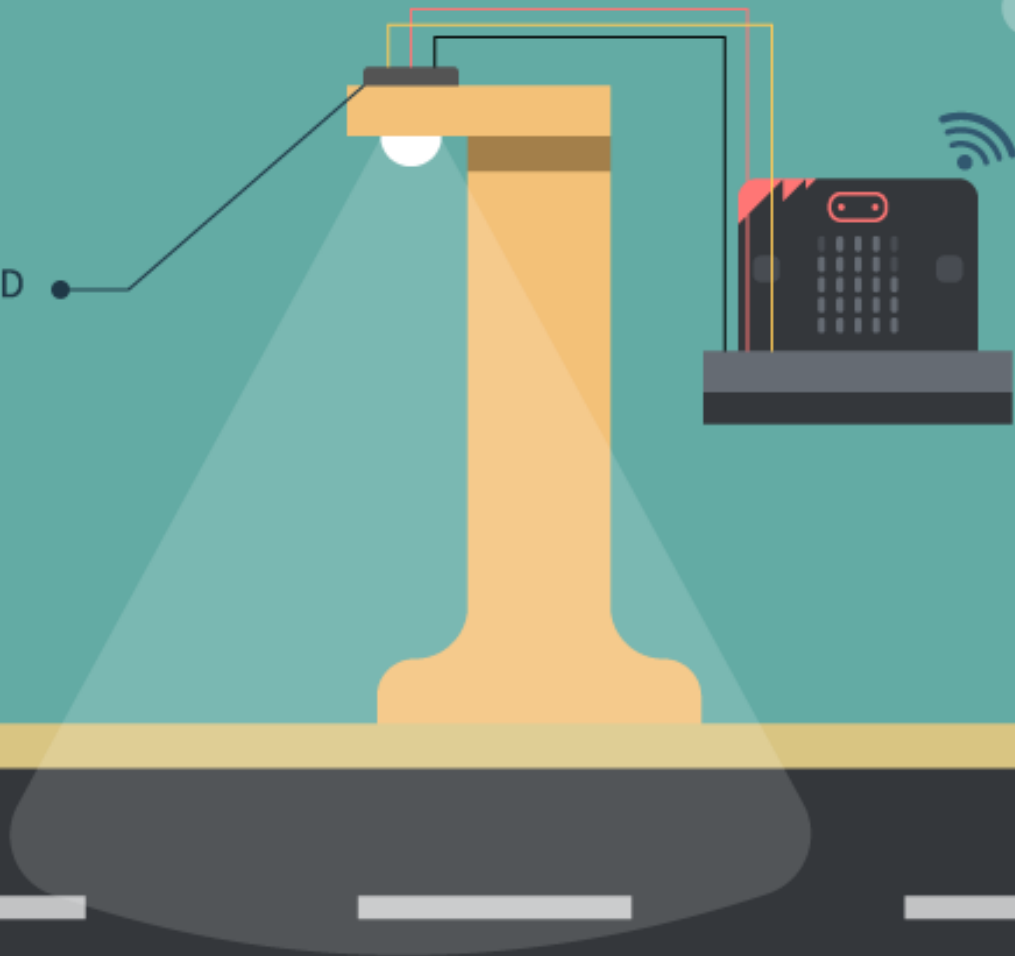
● 節能街燈



06:00 PM

IFTTT

白色LED



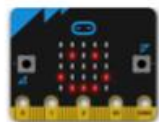
10. [物聯網]節能街燈

程度：★★★★☆

介紹

節能街燈可以在指定時間自動開關的街燈，能夠同時兼顧節省能源及照明需求。

所用部件



micro:bit (1)



IoT:bit (含 OLED 顯示屏)(1)



白色LED (1)



三腳模組連接線(1)



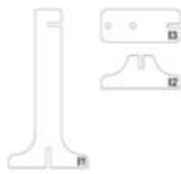
M4*10毫米螺絲(2)



M4 螺母(2)

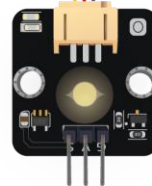
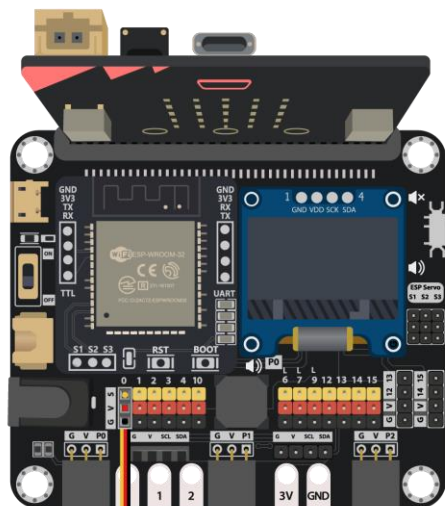


螺絲批 (1)



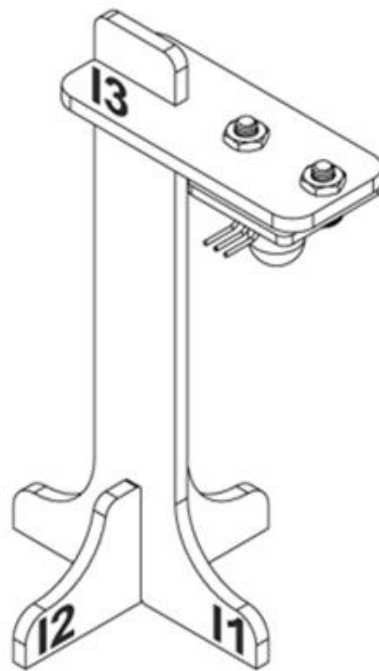
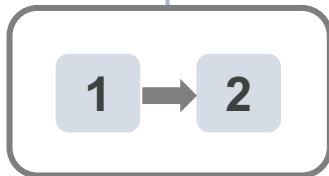
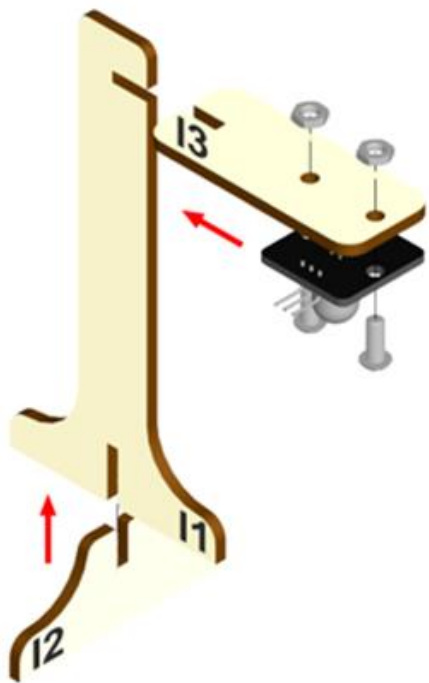
I1-I2 模型 (1)

IFTTT



組裝步驟

I1-I3 模型	x1
白色LED	x1
M4 螺母	x2
M4*10毫米螺絲	x2



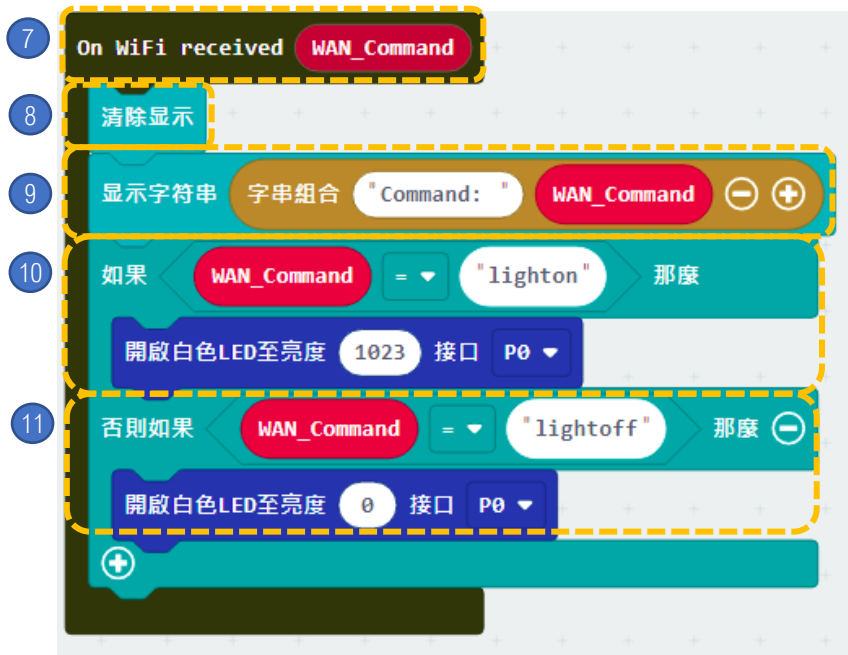


The image shows a Scratch script with the following blocks:

1. 當啟動時 (When the program starts)
2. OLED 初始化 寬 128 高 64 (Initialize OLED with width 128 and height 64)
3. 初始化IoT:bit TX P16 RX P8 (Initialize IoT:bit with TX P16 and RX P8)
4. 連接到WiFi 名稱 "smarthon" 密碼 "12345678" (Connect to WiFi with name "smarthon" and password "12345678")
5. 當WiFi連接成功後 (When WiFi connection is successful) - This block contains:
 - 顯示 圖示 (Show icon) - A grid of dots icon.
6. 显示字符串 Device_ID (Display string Device_ID)

步驟:

1. 「當啟動時」只會在程式最開始時運作。
2. 初始化 OLED (寬128 高64)。
3. 初始化 IoT:bit (TX P16 PX P8)。
4. 設置WiFi 並輸入WiFi名稱及密碼。
5. 在連上WiFi後，在OLED屏幕上顯示圖示「別號」。
6. 顯示字符串 **Device_ID**。



步驟:

7. 當從互聯網接收到指令時，
8. 清除OLED顯示。
9. 顯示字符串字串組合：
"Command: , WAN_Command"。
10. 如果 WAN_Command = "lighton"，開啟白LED至亮度1023接口P0。
11. 如果 WAN_Command = "lightoff"，開啟白LED至亮度0接口P0。

IOT (物聯網) – IFTTT 雲端控制

步驟一. 在 IFTTT 開立新專案(If Date&time Then Smarthon IoT (micro:bit))

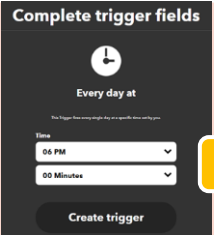
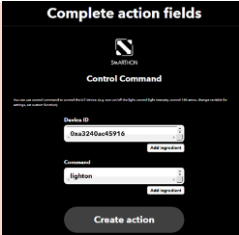


<https://smarthon-docs-en.readthedocs.io/en/latest/smartcity/chapter4.html>

IOT (物聯網) – IFTTT 雲端控制

步驟二.

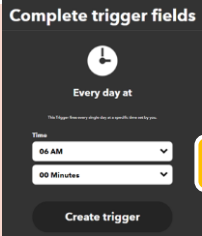
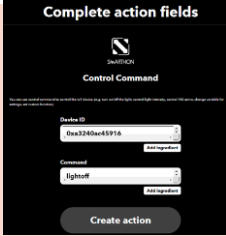
- 如果現在時間為18:00，傳送 WAN控制指令: lighton 至 micro:bit

觸發 (This): Date&time Every day at	行動 (That): Smarhon IoT (micro:bit) Control Command
 <p>Complete trigger fields</p> <p>Every day at</p> <p>Time</p> <p>06 PM</p> <p>00 Minutes</p> <p>6:00pm</p> <p>Create trigger</p>	 <p>Complete action fields</p> <p>Control Command</p> <p>Device ID</p> <p>0xa3240ac45916</p> <p>Command</p> <p>lighton</p> <p>lighton</p> <p>Create action</p>

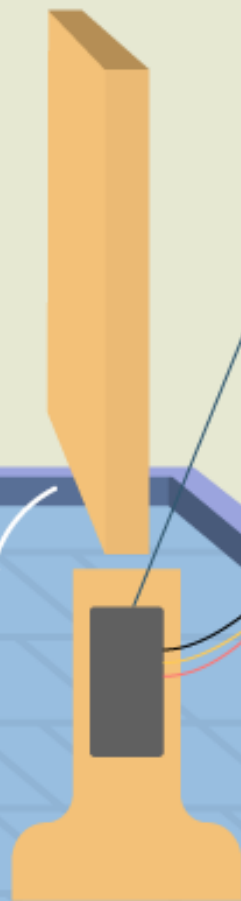
* 在 micro:bit 上運行程式以獲取你的 device ID



- 如果現在時間為06:00，傳送 WAN控制指令: lightoff 至 micro:bit

觸發 (This): Date&time Every day at	行動 (That): Smarhon IoT (micro:bit) Control Command
 <p>Complete trigger fields</p> <p>Every day at</p> <p>Time</p> <p>06 AM</p> <p>00 Minutes</p> <p>6:00am</p> <p>Create trigger</p>	 <p>Complete action fields</p> <p>Control Command</p> <p>Device ID</p> <p>0xa3240ac45916</p> <p>Command</p> <p>lightoff</p> <p>lightoff</p> <p>Create action</p>

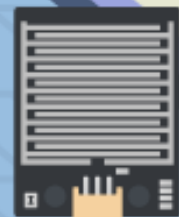
● 自動天台簷篷系統



● 180° 舵機



● 雨水傳感器



11. [物聯網] 自動天台簷篷系統

程度：★ ★ ★ ★ ☆

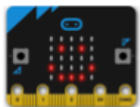
介紹

當雨水傳感器偵測到下雨，就會傳送訊號至micro:bit 以開啟天台簷篷。當沒有下雨時，天台簷篷會被關上。

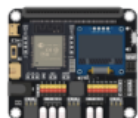
同時，micro:bit通過IFTTT平台上的天氣預報系統，取得指令內容開關簷篷。

IFTTT

所用部件



micro:bit (1)



IoT:bit (含OLED 顯示屏) (1)



180° 舵機(1)



三腳模組連接線 (1)



舵機尖螺絲 M2*7.5毫米(1)
(於舵機套裝內)



M2*10毫米螺絲 (2)



M2 螺母 (2)



M4*10毫米螺絲 (2)



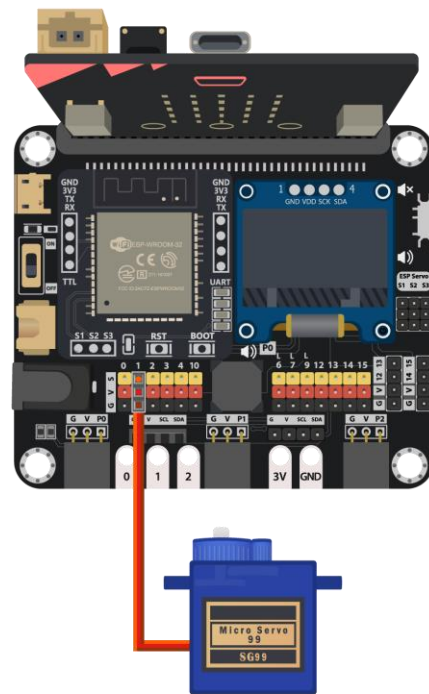
M4 螺母 (2)



螺絲批 (1)

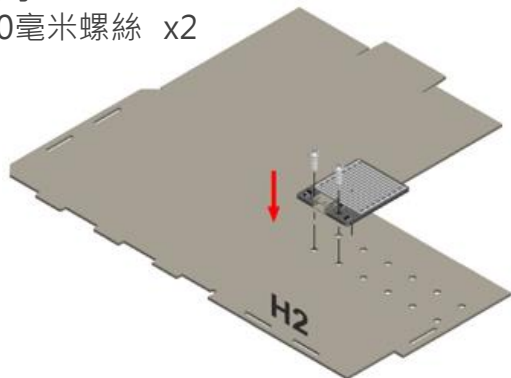


H1-H7 模型 (1)

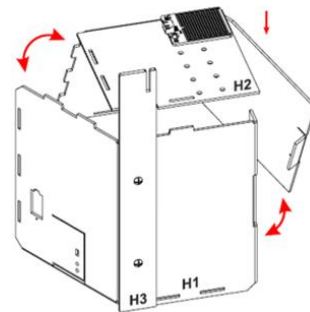
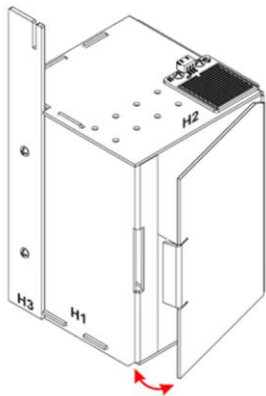
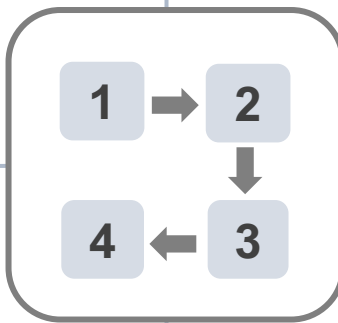
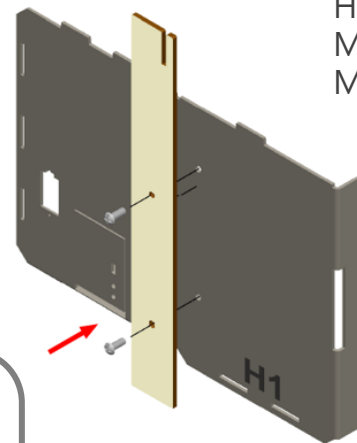


組裝步驟

- H2 模型 x1
- 雨水傳感器 x1
- M4 螺母 x2
- M4*10毫米螺絲 x2

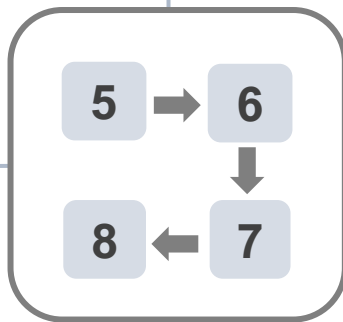
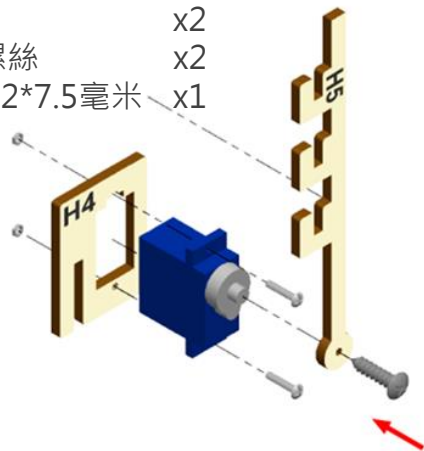


- H1 & H3 模型 x1
- M4 螺母 x2
- M4*10毫米螺絲 x2

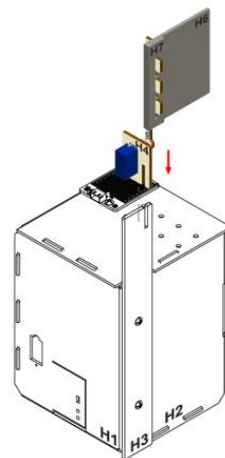
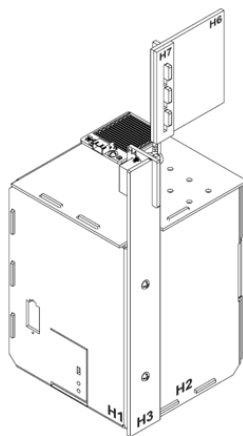
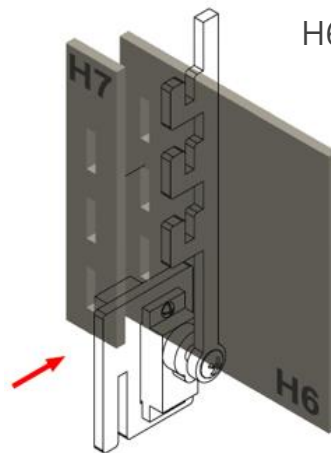


組裝步驟

- H4 - H5 模型 x1
- 180° 舵機 x1
- M2 螺母 x2
- M2*10毫米螺絲 x2
- 舵機尖螺絲M2*7.5毫米 x1



- H6 - H7 模型 x1





The image shows a Scratch script with the following blocks:

1. 當啟動時 (When the program starts)
2. OLED 初始化 寬 128 高 64 (Initialize OLED with width 128 and height 64)
3. 初始化IoT:bit TX P16 RX P8 (Initialize IoT bit TX P16 RX P8)
4. 連接到WiFi 名稱 "smarthon" 密碼 "12345678" (Connect to WiFi with name "smarthon" and password "12345678")
5. 當WiFi連接成功後 (When WiFi connection is successful)
 - 顯示 圖示 (Show icon) - A grid of dots icon is selected.
6. 显示字符串 Device_ID (Display string Device_ID)

步驟:

1. 「當啟動時」只會在程式最開始時運作。
2. 初始化 OLED (寬128 高64)。
3. 初始化 IoT:bit (TX P16 PX P8)。
4. 設置WiFi 並輸入WiFi名稱及密碼。
5. 在連上WiFi後，在OLED屏幕上顯示圖示「別號」。
6. 顯示字符串 **Device_ID**。

編程



你能用其他天氣狀況控制天台簷篷嗎？（例如：陽光）

```
7 當從互聯網接收到指令 WAN_Command
8 清除顯示
9 顯示字符串 字符串組合 "Command: " WAN_Command
10 如果 WAN_Command = "Rain" 那麼
    轉動舵機到 90 度 接口 P1
11 否則如果 WAN_Command = "Clear" 那麼
    轉動舵機到 180 度 接口 P1
```

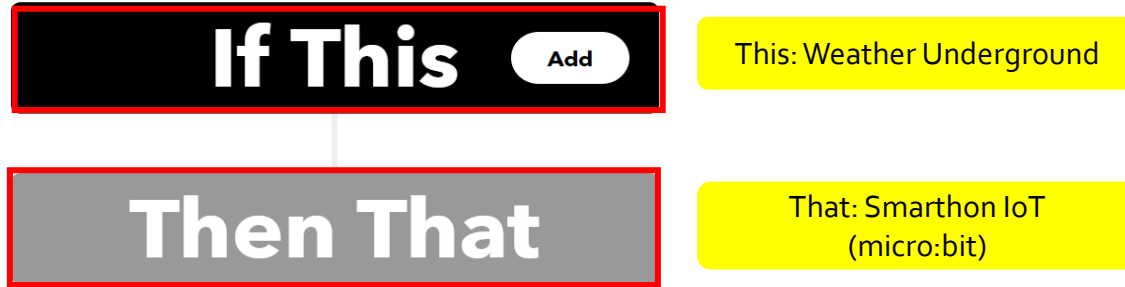
```
12 當按鈕 A 被按下
13 轉動舵機到 90 度 接口 P1
14 當按鈕 B 被按下
15 轉動舵機到 180 度 接口 P1
```

步驟:

7. 當從互聯網接收到指令時，
8. 清除OLED顯示。
9. 顯示字符串字符串組合：“Command: , WAN_Command”。
10. 如果 WAN_Command = “Rain”，轉動舵機到90度接口P1。
11. 如果 WAN_Command = “Clear”，轉動舵機到180度接口P1。
12. 當按鈕A被按下，
13. 轉動舵機到90度接口P1。
14. 當按鈕B被按下，
15. 轉動舵機到180度接口P1。

IOT (物聯網) – IFTTT 雲端控制

步驟一. 在 IFTTT 開立新專案(If Weather Underground Then Smarthon IoT (micro:bit))


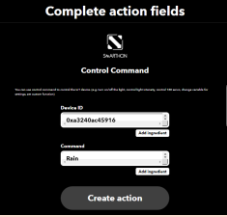


<https://smarthon-docs-en.readthedocs.io/en/latest/smartcity/chapter4.html>


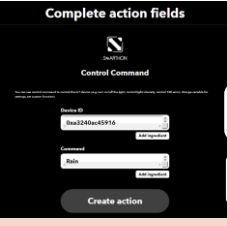
IOT (物聯網) – IFTTT 雲端控制

步驟二.

- 如果現在天氣轉變為下雨，傳送 WAN 控制指令: Rain 至 micro:bit

觸發 (This): Weather Underground Current condition changes to	行動 (That): Smarthon IoT (micro:bit) Control Command
 <p>Rain</p>	 <p>Device ID</p> <p>Rain</p>

- 如果現在雨勢已經停止，傳送 WAN 控制指令: Clear 至 micro:bit

觸發 (This): Weather Underground Current condition changes to	行動 (That): Smarthon IoT (micro:bit) Control Command
 <p>Clear</p>	 <p>Device ID</p> <p>Clear</p>

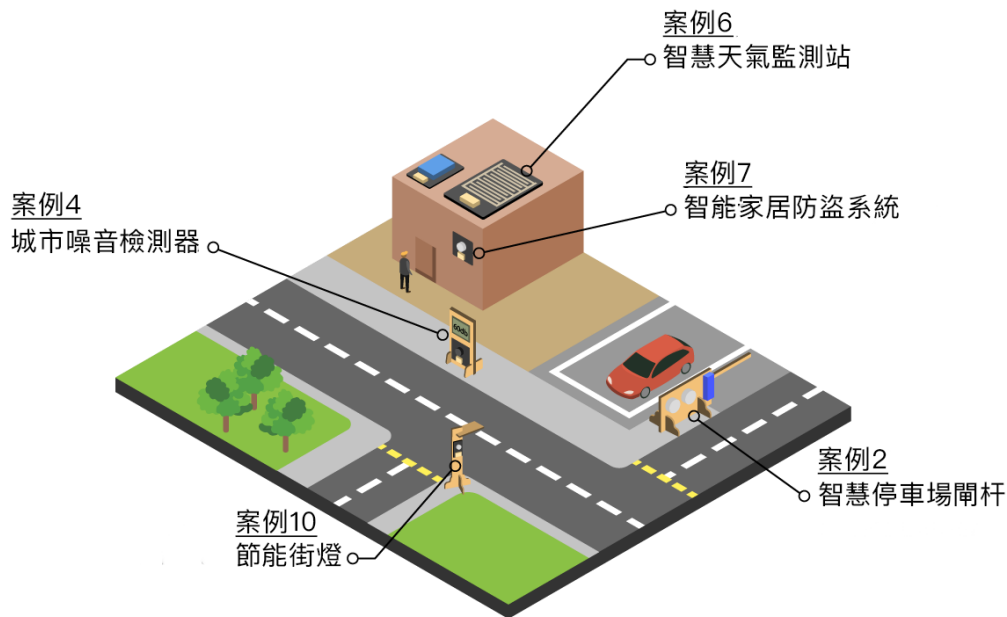
* 在 micro:bit 上運行程式以獲取你的 device ID



場境示例 1:

智慧城市生活

智慧城市生活側重於實現市民的高生活水準，它包括停車場自動化系統和高度安全的防禦系統，收集噪聲和天氣信息以評估城市的環境舒適性。



部件:

- 溫濕度傳感器(DHT11)
- 雨水傳感器
- 運動傳感器
- 蜂鳴器
- 光度傳感器
- 180° 舵機
- 超音波距離傳感器
- 聲音傳感器
- 白色LED

IOT(物聯網):

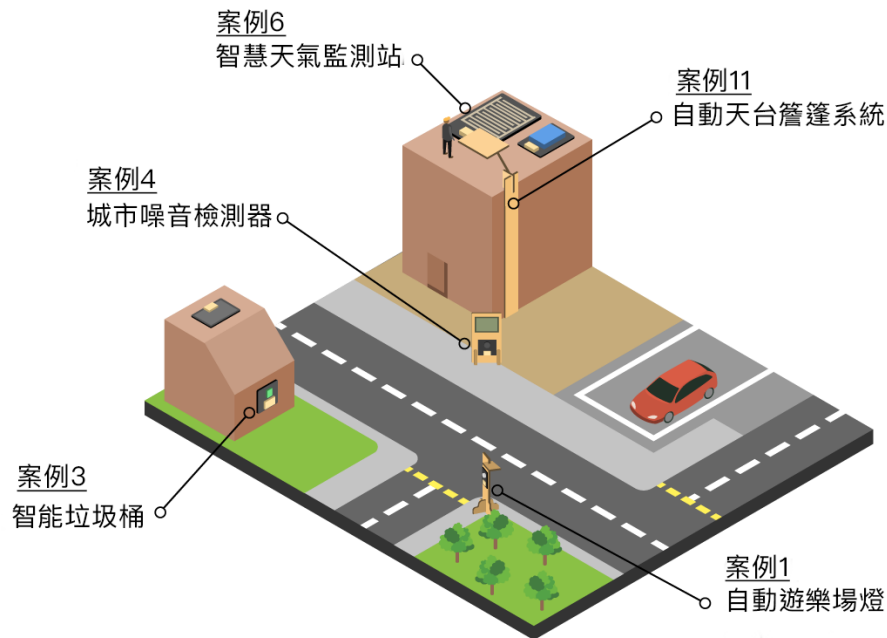
- ThingSpeak
- IFTTT



您可以設計自己的智慧城市，只需將不同的案例組合在一起或在您的城市中創建專用案例！

場境示例 2: 智慧環境城市

這是一個注重高可持續性的城市發展，其中包括增加綠化面積，提高能源效率，減少廢物，監測氣候信息和污染狀況（例如噪音）。



部件:

- 溫濕度傳感(DHT11)
- 雨水傳感器
- 180° 舵機
- 白色LED
- 運動傳感器
- 聲音傳感器
- 超音波距離傳感器
- 彩色LED(WS2812B)

IOT(物聯網):

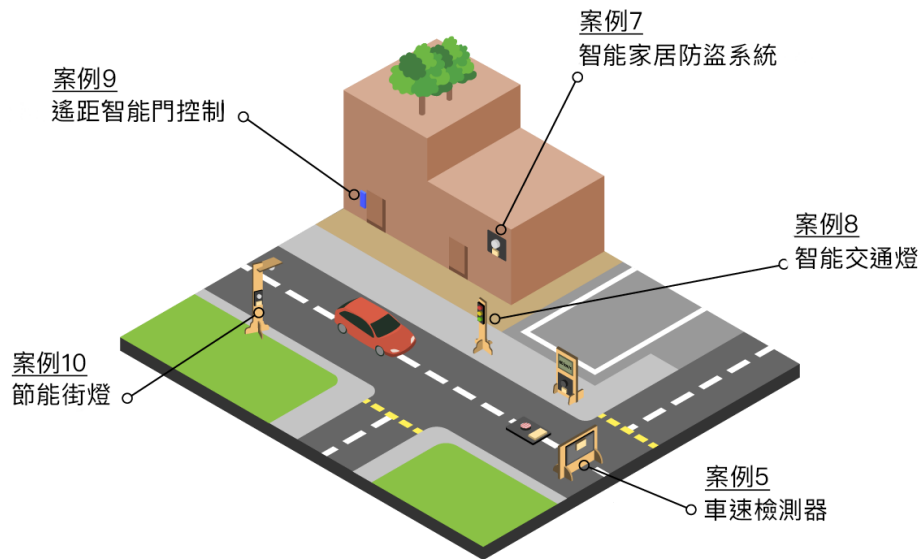
- ThingSpeak
- IFTTT



您可以設計自己的智慧城市，只需將不同的案例組合在一起或在您的城市中創建專用案例！

場境示例 3: 智慧交通安全

交通安全是城市的首要任務，在公共交通系統中監視車速和交通狀況，只有公司的負責人才能進入大樓，如果附近有可疑人員，警報將通過網絡發送。



部件:

- 180° 舵機
- 白色LED
- 蜂鳴器
- 運動傳感器
- 光度傳感器
- 交通燈模組
- 超音波距離傳感器

IOT(物聯網):

- App Inventor 2
- IFTTT
- WiFi 頻道



您可以設計自己的智慧城市，只需將不同的案例組合在一起或在您的城市中創建專用案例！

附錄 I: Smarthon電子模組 – 探索更多

	智能水泵		溫度，濕度及壓力傳感器
	電動風扇		數位光度傳感器
	水溫傳感器	想知道更多資訊，請前往 www.smarthon.cc	

**想知道更多資訊
請前往**

<https://smarthon-docs-en.readthedocs.io/en/latest/smartcity/index.html>



SMARTHON

www.smarthon.cc

INGITE YOUR CREATIVITY