



智慧城市 IOT

MICRO:BIT 入门套装

Copyright © Smarthon Limited. All rights reserved.

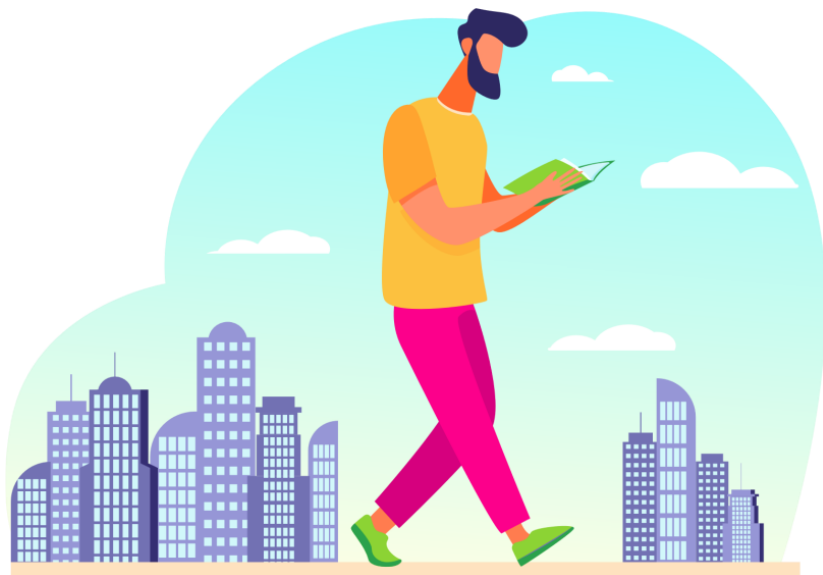


关于 SMARTHON (研科)



SMARTHON LIMITED 的总部位于香港，为教育和创意工作者设计、制造和销售以常规课程，如通识教育等为主的产品。我们的使命是促进所有人能够获得高质量的科技教育。

SMARTHON 的成立是为了设计简易的设备，并将不断发展的技术应用于教学。例如，教师可以使用 micro:bit 和传感器来教导学生设计一个小型温室系统，然后利用物联网技术快速将相关数值传输到智能设备。教师可以实时调整温度或湿度，让学生观察不同的变化，记录下来，并用于学习、分析和了解温室系统的运作。



目录



- 介绍 P. 2
- 甚么是智慧城市及IoT? P. 5-7
- 零件列表: 所用部件及模型列表 P. 8-11
- Smarthon IoT:bit介绍 P. 12-13
- 快速入门: 添加PXT扩展 P. 14
- 案例 1: 自动游乐场灯 P. 15-18
- 案例 2: 智慧停车场闸杆 P. 19-22
- 案例 3: 智能垃圾桶 P. 23-26
- 案例 4: 城市噪音检测器 P. 27-30
- 案例 5: 车速检测器 P. 31-36

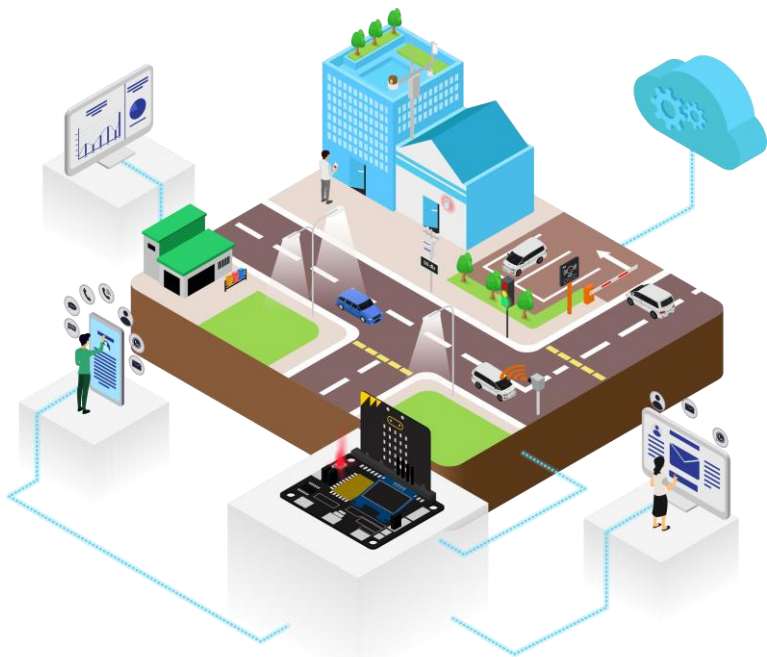
目录



- 物联网案例 6: 智慧天气监测站 P. 37-44
- 物联网案例 7: 智能家居防盗系统 P. 45-51
- 物联网案例 8: 智能交通灯 P. 52-60
- 物联网案例 9: 遥距智能门控制 P. 61-68
- 物联网案例 10: 节能街灯 P. 69-75
- 物联网案例 11: 自动天台檐篷系统 P. 76-83
- 场境示例 1-3 P. 84-86
- 附录 I: Smarthon电子模块 – 探索更多 P. 87

SMARTHON 智慧城市

IoT 学习套装



SMARTHON 智慧城市 IoT 学习套装是专门为了教授物联网(IoT)设计的套装。你可以利用物联网，建设一个独一无二的智慧城市。套装内提供的 Smarthon IoT:bit 开发板，能够使用各种传感器及驱动器，令你能够在设计中实现多种有趣的功能。例如，使用传感器去侦测交通状况然后上载到互联网，令你能够作出统计或是控制等。

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

甚么是智慧城市？

智能城市是架构、整合信息科技 (ICT) 以在六大范畴——交通、生活、环境、人民、政府及经济，增加可持续性、效率和生活质素。在本套装里面，我们将会集中在三个类别——智能生活、智能环境和智能交通。



智慧生活

能够使用物联网/科技去提升方便程度/生活水平



智慧政府

政府支持在私人/公共领域及城市建设中，使用及发展科技



智慧环境

监测污染及减少浪费，发展绿色小区



智慧经济

强化现有经济支柱及透过科技发展新经济支柱



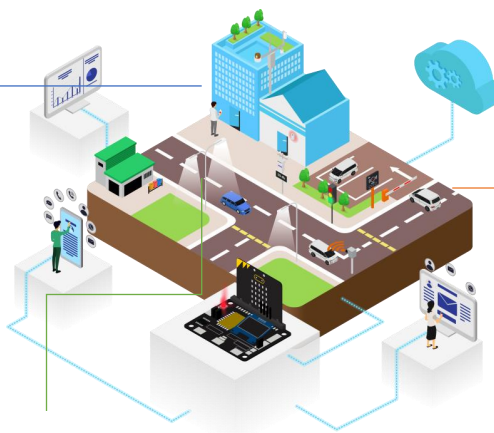
智能交通

透过提供实时信息，提升出行体验



智慧人民

提供编程训练去培养IT专业



什么是 IOT(物联网) ?

物联网 (IoT) 是一个透过互联网连接不同对象 (有形或无形, 包括计算机设施, 系统) 的网络。透过在对象上添加传感器, 软件或其他科技, 使其能够在不需要人与人或人与计算机的互动的情况下, 进行数据连接和交换。物联网中有三种分享信息的方式:

1. 收集及分享信息

对象上的传感器收集数据 (温度, 湿度, 雨滴等) 并上传到中央系统以进行进一步分析

例子: 上传数据到 ThingSpeak

例子2: 使用 IFTTT 发送邮件

2. 接收讯息及反应

对象等待指令, 之后作出反应

例子: 使用 App Inventor 2/IFTTT 服务控制 micro:bit

3. 两者皆做

收集信息并使用互联网分享, 然后作出反应

例子: 对象与对象通讯

智能雨伞能收集雨滴数据并传送至气候系统。气候系统会传送回降雨机率使雨伞进行开/关。



所用部件

SMARTHON IOT BIT:

这是一块专为让Micro:bit加入物联网而设计的开发板。它能够连接到不同的传感器和驱动器。



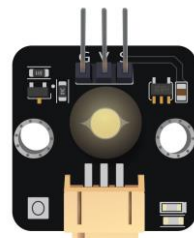
交通灯模块:

内含三种不同颜色LED灯，分别为红、黄、绿三色，就像现实生活中的交通灯一样。



白色 LED:

能够发出白色的LED光，使用者能控制它的亮度。



雨水传感器:

能够使用导电度去判断是否有雨水在传感器上。



超声波距离传感器:

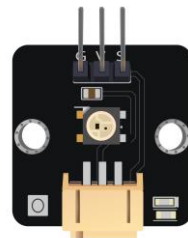
能够使用超声波去量度传感器和对象的距离。



彩色 LED

(WS2812B):

这是一块LED综合芯片，能够透过编程发出不同RGB颜色的光。

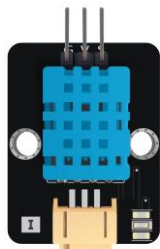


PART LIST

温湿度传感器

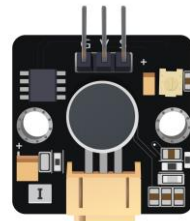
(DHT11):

包含两个部件，分别能侦测湿度及温度。



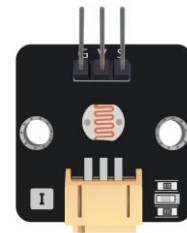
声音传感器:

使用带有扩音器的麦克风来检测环境的声等级。



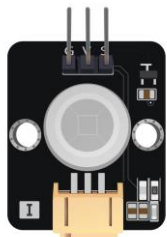
亮度传感器:

能够使用光敏电阻来检测环境的亮度。



运动传感器:

运动传感器是使用红外线传感器去检测出前方有没有正在移动的物体。



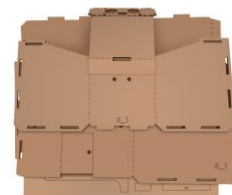
180° 舵机:

能够使它的轴定位在 0 到 180 度之间。



纸板及木板模型:

能够把传感器及驱动器安装在上面。



城市地图:

大约尺寸为 594 毫米*420毫米，可以放置纸板模型在地图上。



其他:

(9) 模块连接线 · (8) 延长线 · (12) M2*10毫米 螺丝 · (6) M3*10毫米 螺丝 · (30) M4*10毫米 螺丝 · (12) M2螺母 · (6) M3螺母 · (30) M4螺母 · (3) 回形针 · (2) L型支架 · (1) 螺丝起子 · (2) 砂纸 · (1) 万用黏土胶 (Blu Tack) · (1) USB 连接线 · (1) 电池盒(AA*4) · (1) 智能城市套装使用手册 ·

模型列表

模型 A:

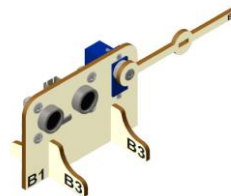
这是一个有斜坡灯的小型游乐场灯模型。有一个传感器端口在侧边。（例如：可以接上动作传感器并用于侦测动作）



游乐场灯

模型 B:

这个模型中，超声波距离传感器被用于侦测是否有车辆经过。同时，舵机用于控制闸杆的角度。



停车场闸杆

模型 C:

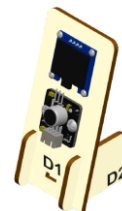
这个垃圾桶可以透过超声波距离传感器去侦测垃圾桶内的垃圾数量。在垃圾桶前有一个传感器端口，你可以设置一个彩色LED去显示垃圾的数量。



垃圾桶

模型 D:

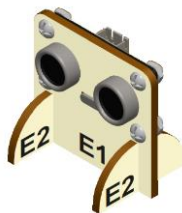
这个OLED显示屏可以用于显示有用的信息。使用者可以把传感器接上OLED显示屏以收集信息。（例如：声音传感器或者温度传感器）。



OLED 显示屏

模型 E:

这个模型中，超声波距离传感器可以用于量度模型与目标对象之间的距离。它可以和模型D一齐使用以展示对象的距离/速度。



车速检测器

模型列表

模型 F:

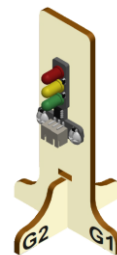
这是一个小型建筑物模型。模型上有数个传感器端口在屋顶上，使用者可以设置不同的传感器/驱动器。（例如：雨滴传感器，温湿度传感器）



小型建筑物
(公寓/商业大楼)

模型 G:

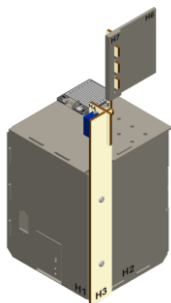
这个模型可以接上交通灯模块。这个模型可以放置在城市的街道上，并使用其3个LED灯来展示交通路况。



交通灯

模型 H:

这是一个有着由舵机控制的自动天台檐篷系统的高建筑物模型。使用者可以设置不同的传感器/驱动器。（例如：雨滴传感器，温湿度传感器）



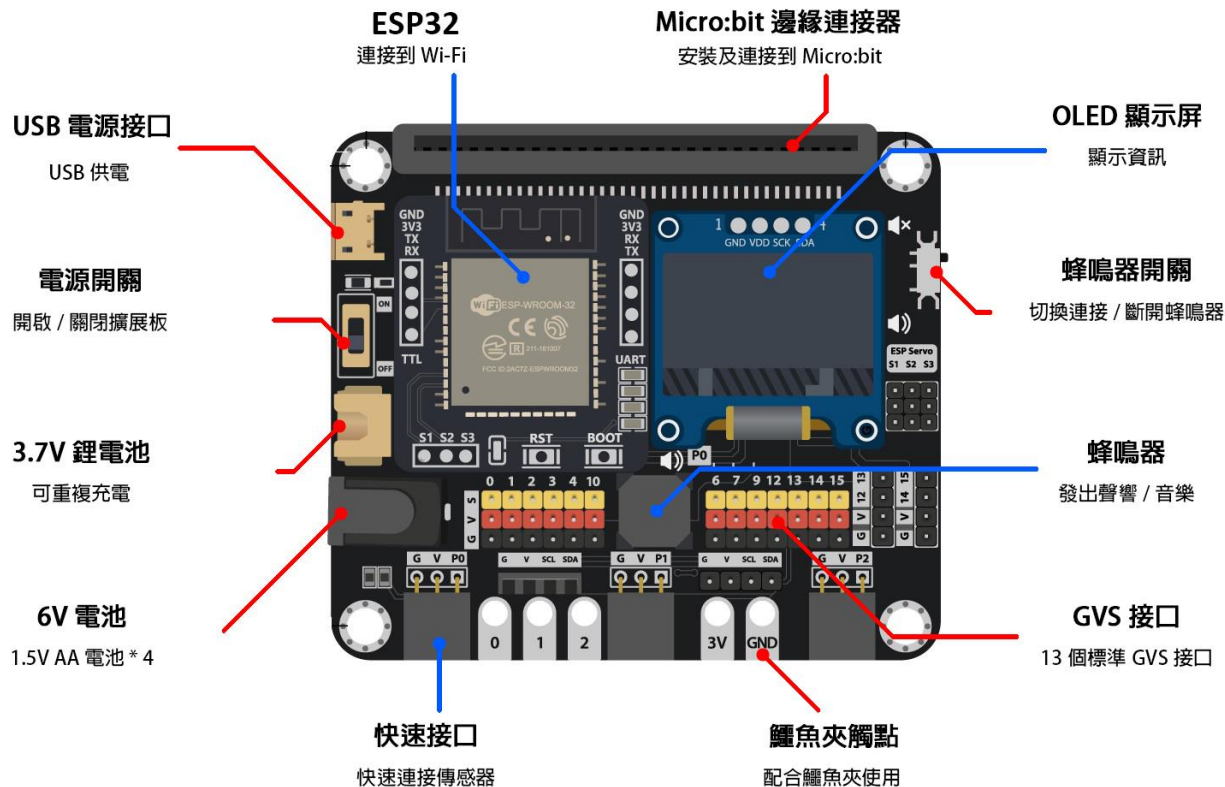
大型建筑物
(摩天大楼/高层住宅)

模型 I:

这是一个有着斜坡灯的高尺寸街灯。它能够支持城市内的智能照明系统。在你的项目中，你可以使用物联网来控制照明系统。



街灯



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

连接电源

IoT:bit 电压范围：3.7V DC-6V DC。有三种电源选项：

选项 1

- USB 连接线



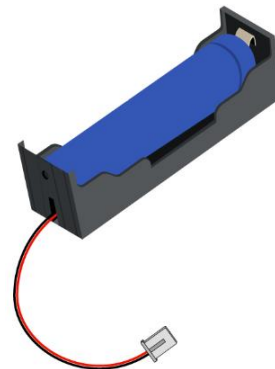
选项 2

- 4 x AA 电池 (本套装内并不包含电池)



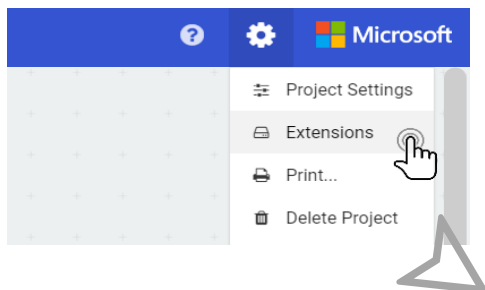
选项 3

- 把18650 3.7V 电池放入电池盒，并把电池盒插头插入XH2.54锂电池接口(本套装内不包含18650电池及18650电池盒)

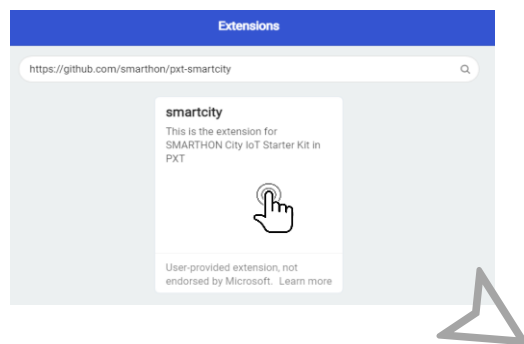


快速开始： 添加PXT扩展

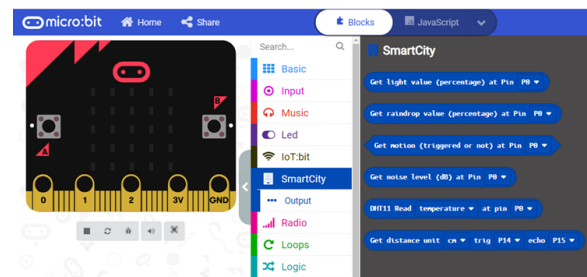
1. 打开 <https://makecode.microbit.org/>，
建立一个新的项目。



2. 搜寻 “<https://github.com/smarthon/pxt-smartcity>”
及点击 “smartcity”。



3. 当载入完成时，[SmartCity](#) 扩展便会出现。



● 自動遊樂場燈

白色LED

● 運動傳感器



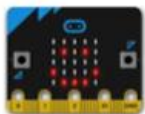
1.自动游乐场灯

程度：★☆☆☆☆

介绍

自动游乐场灯是一款能够自动开启的灯。它能够在感应到有人活动时自动开启，无人时自动关闭，以节省电力。

所用部件



micro:bit (1)



IoT:bit (1)



白色 LED (1)



运动传感器 (1)



三脚模组连接线 (2)



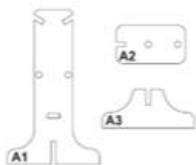
M4*10 毫米螺丝 (4)



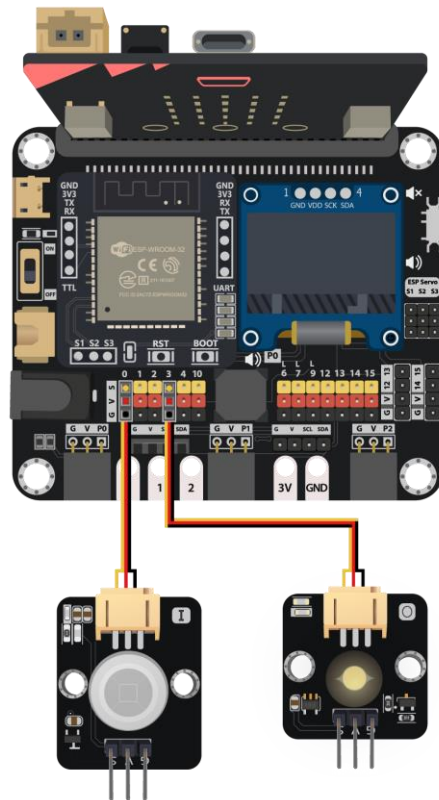
M4 螺母 (4)



螺丝批 (1)

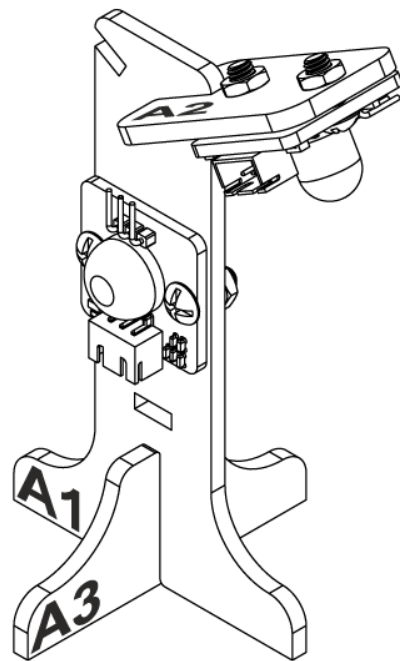
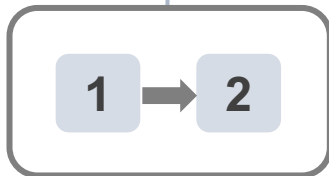
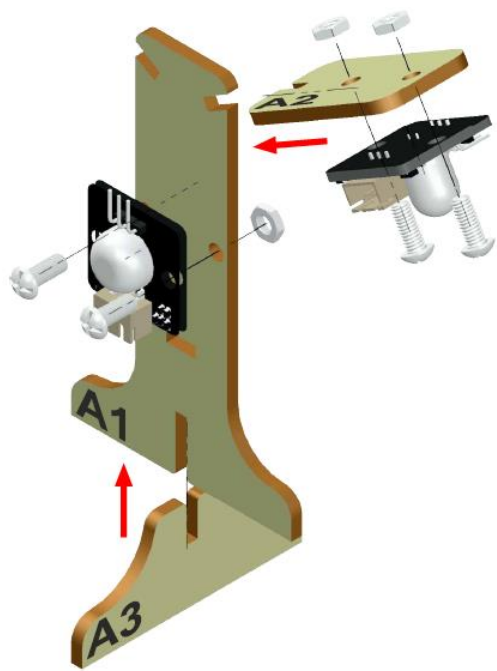


A1-A3 模型 (1)



组装步骤

A1-A3 模型	x1
运动传感器	x1
白色 LED	x1
M4 螺母	x4
M4*10毫米螺丝	x4



编程



除了开灯，你还可以怎样运用运动传感器？（例如：计算人数）

1 當啟動時

2 啟用設為 false



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

步骤:

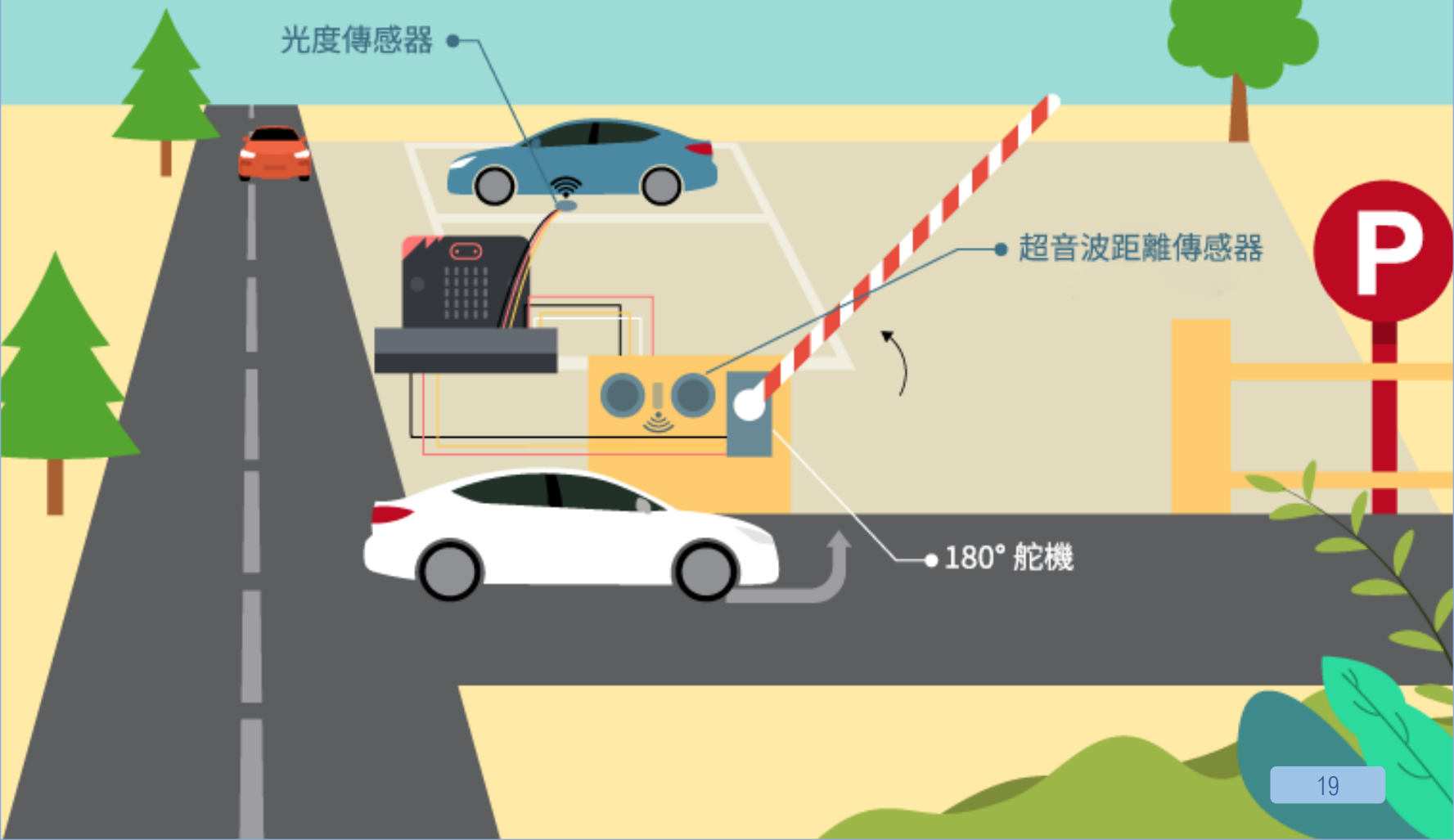
1. 「当启动时」只会在程序最开始时运作。
2. 在「灯光」模块里寻找启用设为“FALSE”并加入至「当启动时」（因为P3预设是用于控制Micro:bit的LED，所以需先关闭）。
3. 「重复无限次」的运作方式和循环相似。
4. 如果动作传感器被触发（例如：有人经过）使P0 = “TRUE”。
5. P3的白色LED灯会被开启
6. 它会暂停10秒所以灯光会持续10秒。
7. P3的白色LED灯会被关闭。

● 智慧停車場閘杆

光度傳感器

超音波距離傳感器

180° 舵機



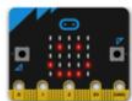
2. 智慧停车场闸杆

程度: ★ ★ ☆ ☆ ☆

介绍

智慧停车场闸杆能为人们带来便利。通过自动侦测停车场内车位状况（例如：空置车位），它能够节省司机等候闸杆开启和寻找车位的时间。

所用部件



micro:bit (1)



IoT:bit (1)



光度传感器 (1)



超音波距離传感器 (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)



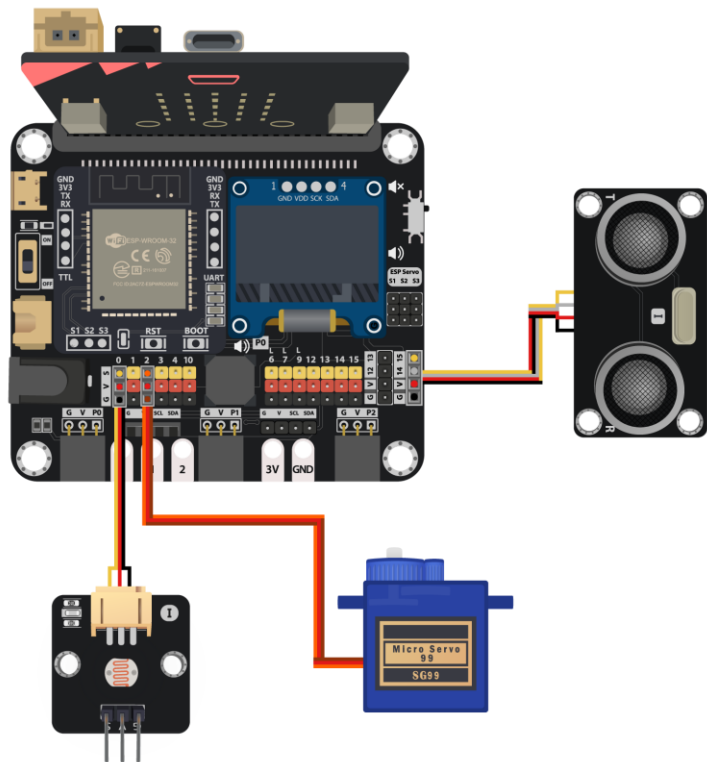
M4 螺母 (4)



螺絲批 (1)

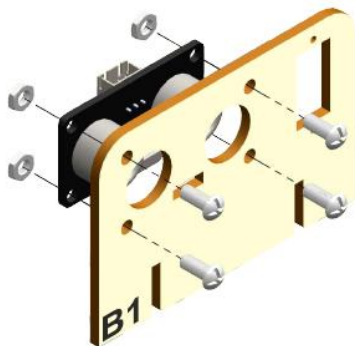


B1-B3 模型 (1)

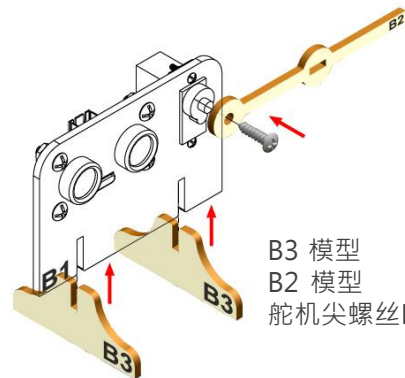
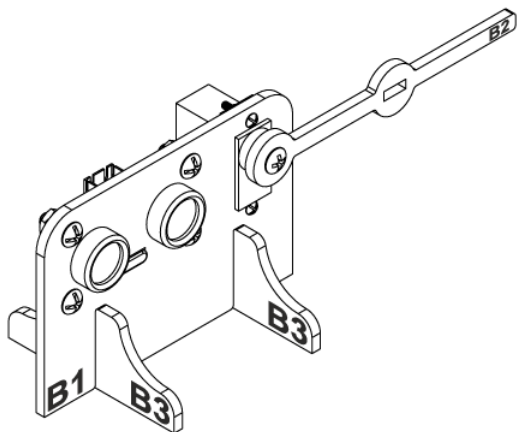
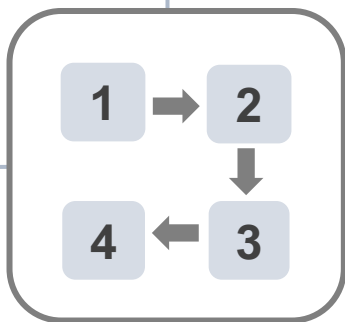
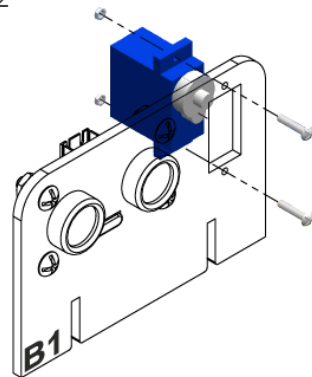


组装步骤

B1 模型 x1
 超声波距离传感器 x1
 M4 螺母 x4
 M4*10毫米螺丝 x4



180° 舵机 x1
 M2 螺母 x2
 M2*10毫米螺丝 x2

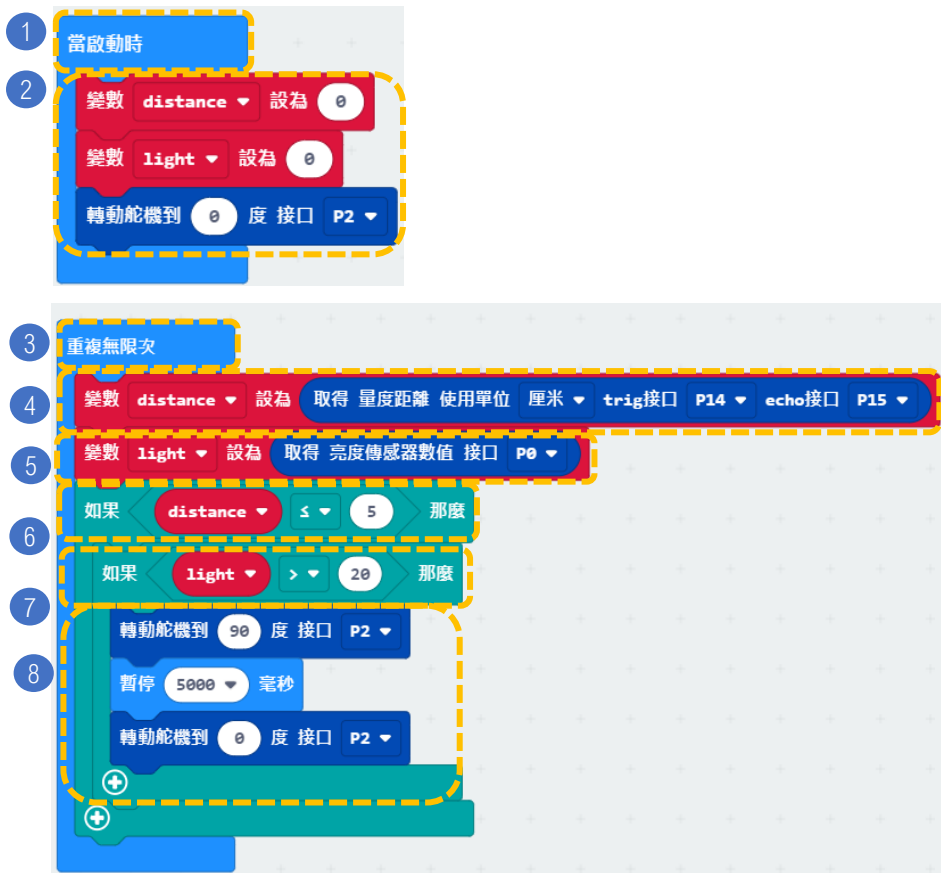


B3 模型 x2
 B2 模型 x1
 舵机尖螺丝M2*7.5毫米 x1

编程



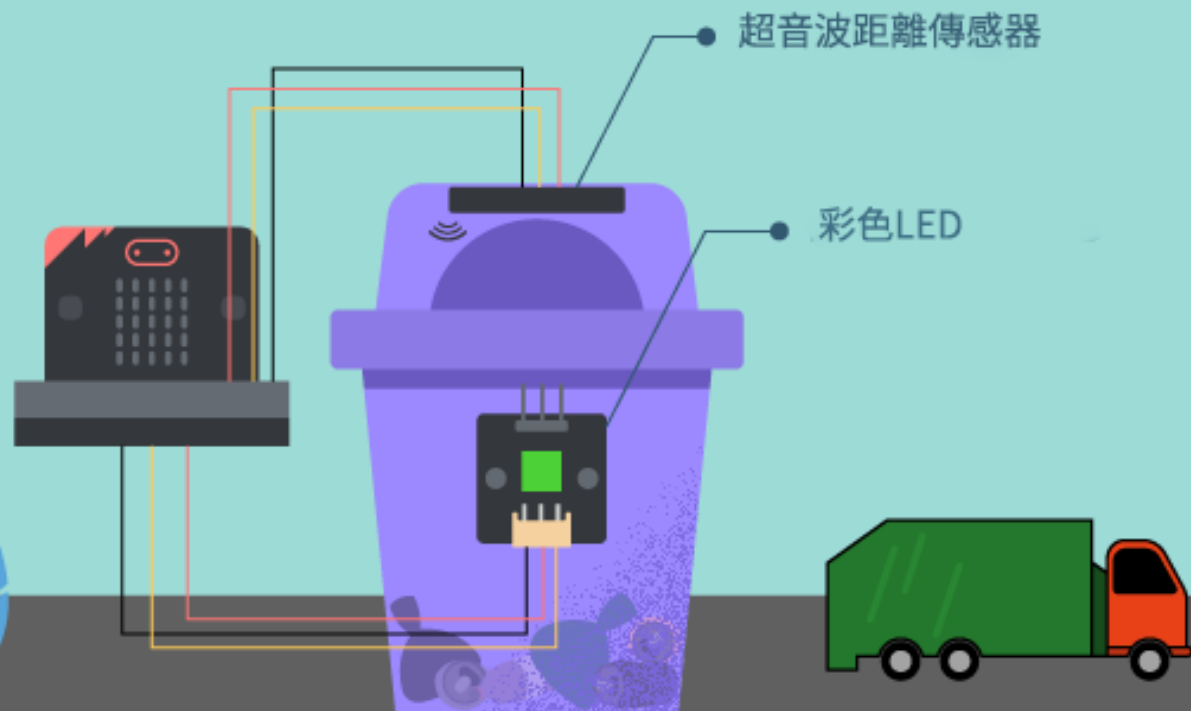
怎样可以在 OLED 显示屏上显示光和距离读数？



步骤:

1. 「当启动时」只会在程序最开始时运作。
2. 将变数 **distance** 和 **light** 设为 0，转动舵机到 0 度 接口 P2。
3. 「重复无限次」的运作方式和循环相似。
4. 把 **distance** 设为“取得量度距离使用单位厘米 trig 接口 P14 echo 接口 P15”。
5. 把 **light** 变量设为“取得亮度传感器数值接口P0”。
6. 如果有车辆经过(**distance** \leq 5)，
7. 而停车场内有空位时(**light** $>$ 20)，
8. 打开停车场闸杆（即转动舵机到90度），暂停5秒使闸杆打开5秒。然后把闸杆关上（即转动舵机到0度）。

● 智能垃圾桶



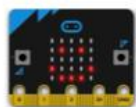
3. 智能垃圾桶

程度：★ ★ ☆ ☆ ☆

介绍

垃圾桶上的 LED 可以告诉人们桶内的当前容量，收集工人可以轻易判断有否需要清理垃圾。这样可以减少更换垃圾袋的次数，从而减少胶袋用量，促进环保城市。

所用部件



micro:bit (1)



IoT:bit (1)



彩色LED (WS2812) (1)



超声波距离传感器(1)



三脚模組連接線 (1)



四腳模組連接線 (1)



M4*10 毫米螺絲 (6)



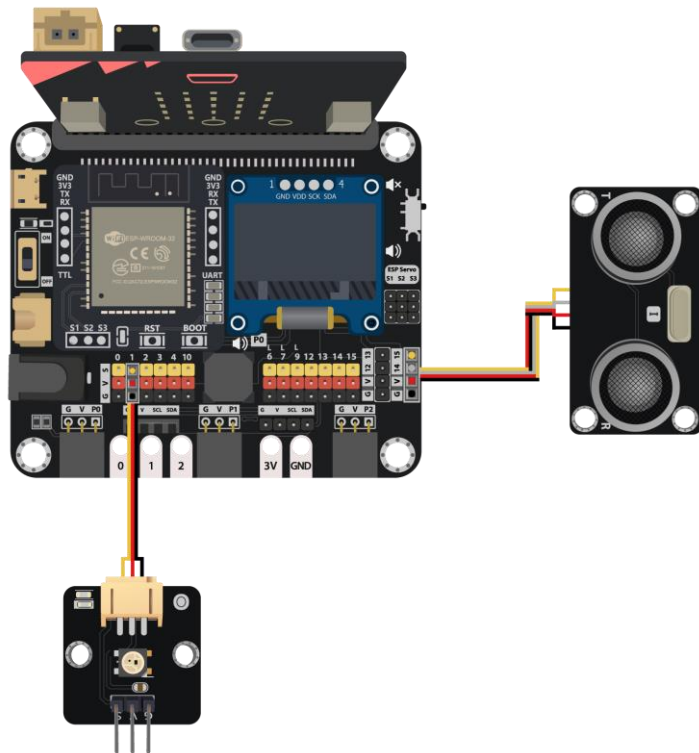
M4 螺母 (6)



螺絲批 (1)

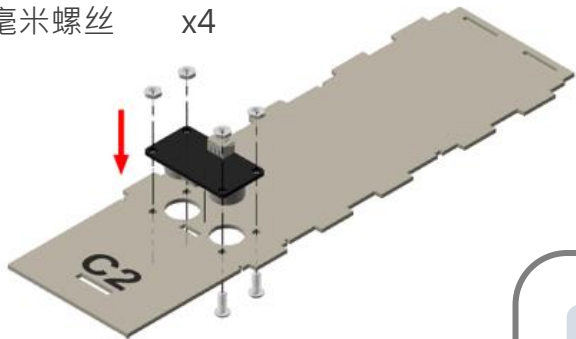


C1-C2 模型(1)

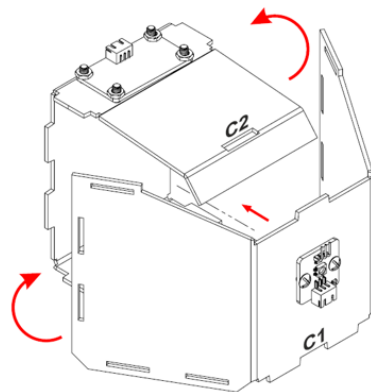
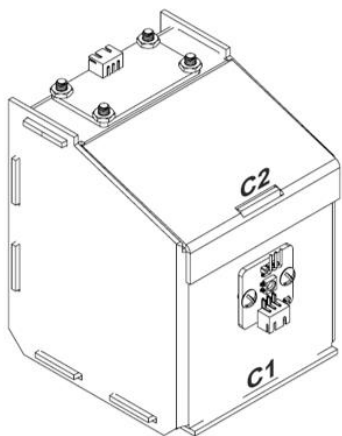
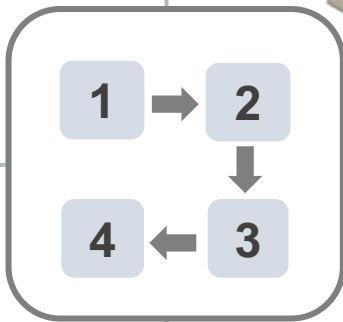
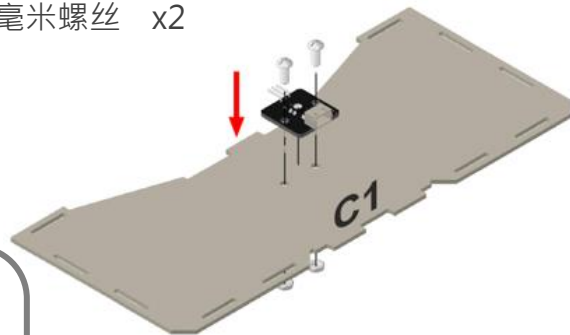


组装步骤

- C2 模型 x1
- 超声波距离传感器 x1
- M4 螺母 x4
- M4*10毫米螺丝 x4



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



编程

💡 如何加入音效提示满载？（例如：利用蜂鸣器）

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 ≤ 4 那麼
strip show color red

8 否則
strip show color green

9 暫停 1000 毫秒

步骤:

1. 「当启动时」只会在程序最开始时运作。
2. 将变数 **distance** 设为 0，并且将变量 **strip** 设为 “NeoPixel at pin P1 with 1 leds as RGB (GRB format)”。
3. 将 **strip** 的亮度设为 50。
4. 暂停 5 秒。
5. 「重复无限次」的运作方式和循环相似。
6. 把 **distance** 设为 “取得量度距离使用单位厘米 trig 接口 P14 echo 接口 P15”。
7. 如果 **distance** \leq 4，strip 会显示**红色**。
8. 否则，strip 会显示**绿色**。
9. 暂停一秒，显示出相应的颜色。

4. 城市噪音检测器

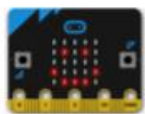
程度：★ ★ ☆ ☆ ☆

介绍

城市噪音检测器能监测公路上的噪音污染，因为过大的噪音会降低周遭人们的生活水平。

安装一个检测器可以帮助工程师收集有关数据用于城市规划。

所用部件



micro:bit (1)



IoT:bit (含 OLED 显示屏) (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)



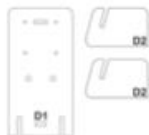
M4*10毫米螺絲 (4)



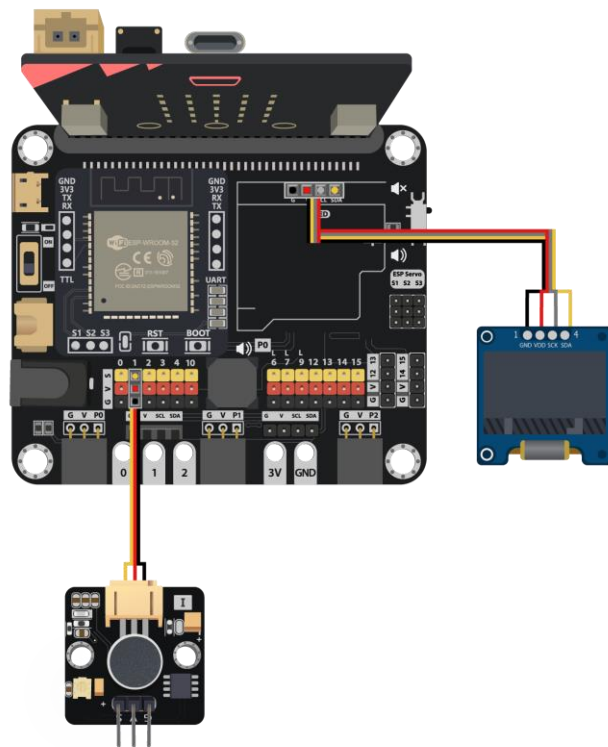
M4 螺母(4)



螺絲批 (1)

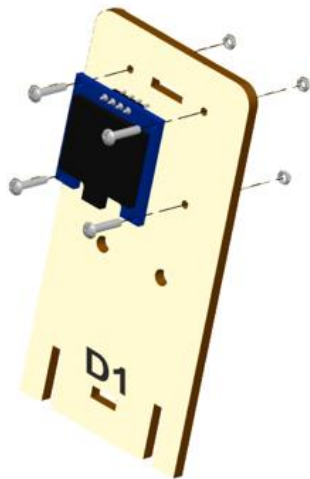


D1-D2 模型 (1)

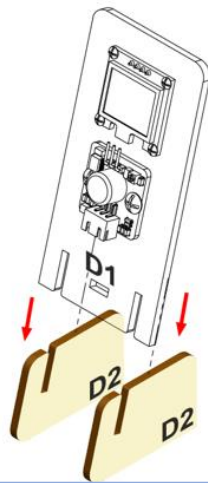
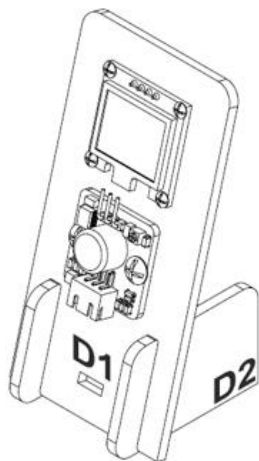
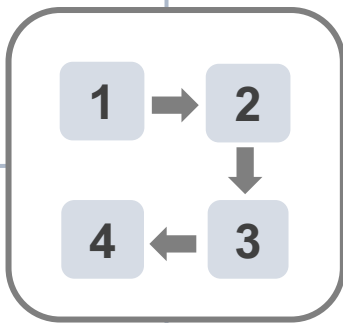
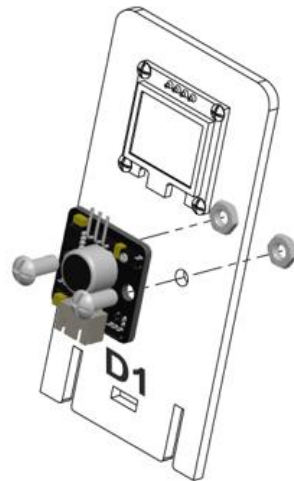


组装步骤

- D1 模型 x1
- OLED 显示屏 x1
- M2 螺母 x4
- M4*10毫米螺丝 x4



- 声音传感器 x1
- M4 螺母 x2
- M4*10毫米螺丝 x2



- D2 模型 x2

编程



当噪音太大可以如何发出提示？（例如：将LED转为红色）

1 当启动时

2 OLED 初始化 宽 128 高 64

3 变数 Noise 设为 0

4 重复无限次

5 变数 Noise 设为 四舍五入 取得 声音传感器数值 接口 P1

6 点亮长条图 显示值为 Noise 最大值为 100

7 清除显示

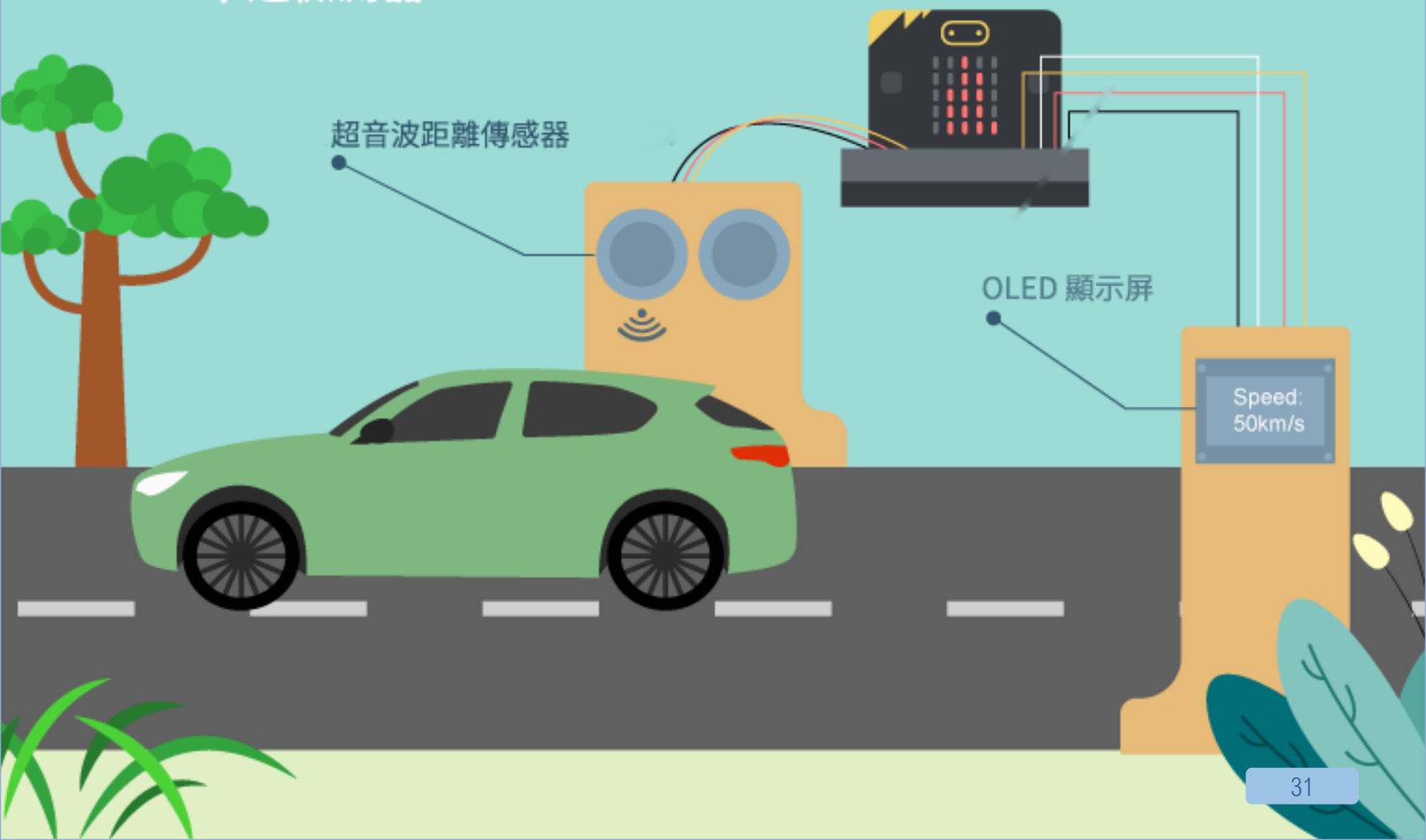
显示字符串 字符串组合 "Noise: " Noise "dB" - +

8 暂停 500 毫秒

步骤:

1. 「当启动时」只会在程序最开始时运作。
2. 初始化 OLED（宽128 高64）
3. 宣告变量 **Noise** 设为0。
4. 「重复无限次」的运作方式和循环相似。
5. 将变数 **Noise** 设为四舍五入取得声音传感器数值接口 P1。
6. 在「灯光」模块中加入点亮直方图显示值为 **Noise**，最大值为100。
7. 加入清除显示，之后加入显示字符串字符串组合（"Noise:" & 'Noise' & "dB"）。
8. 暂停0.5秒，显示与噪音等级相对应的图表。

● 車速檢測器



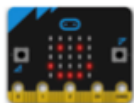
5. 车速检测器

程度：★ ★ ★ ☆ ☆

介绍

车速检测器是一个检测系统，根据车辆在一定时间内移动距离得出其车速。安装检测系统用以检控有效减少交通事故发生。

所用部件



micro:bit (1)



IoT:bit (含 OLED 显示屏) (1)



超声波测距传感器(1)



四脚模组连接线 (1)



四脚延长线 (1)



M2*10 毫米螺丝 (4)



M2 螺母 (4)



M4*10毫米螺丝(4)



M4 螺母 (4)



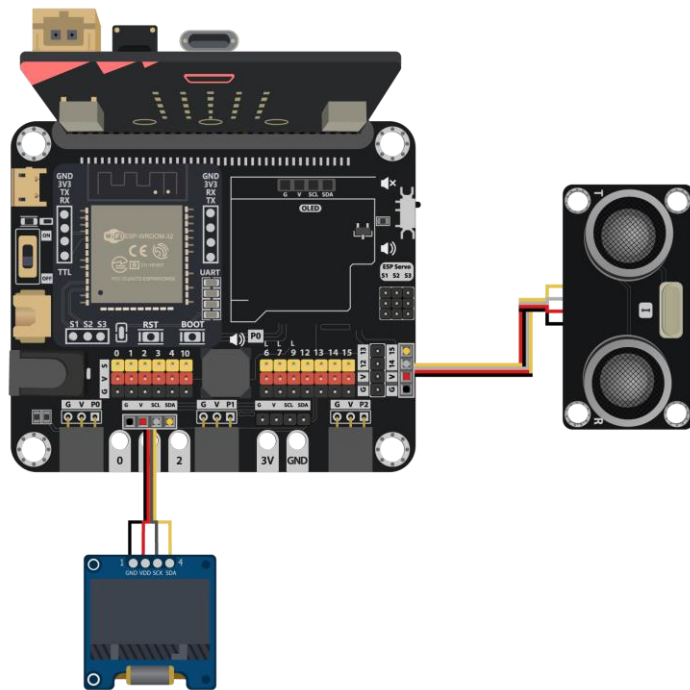
螺丝批 (1)



D1-D2 模型 (1)

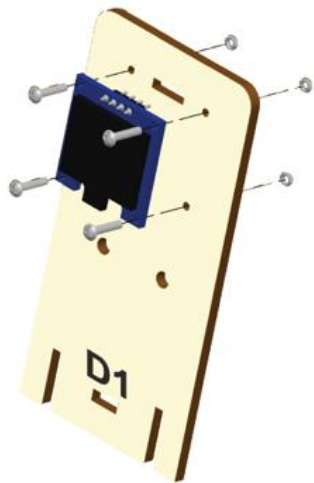


E1-E2 模型 (1)

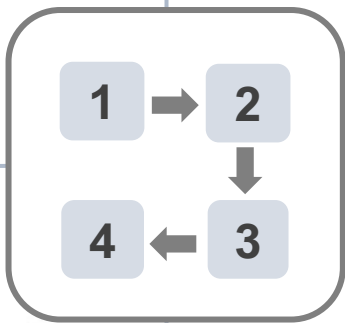
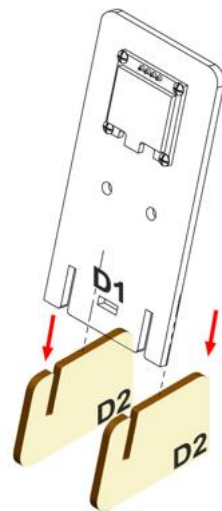


组装步骤

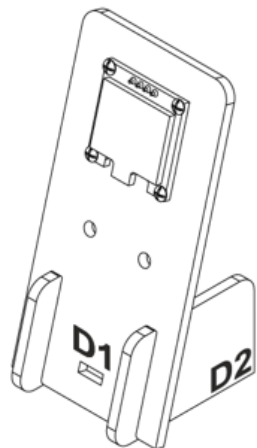
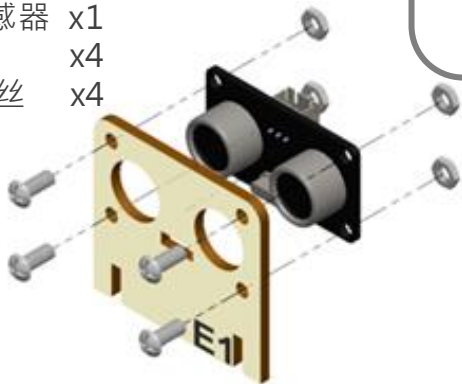
- D1 模型 x1
- OLED显示屏 x1
- M2 螺母 x4
- M2 *10毫米螺丝 x4



- D2 模型 x2

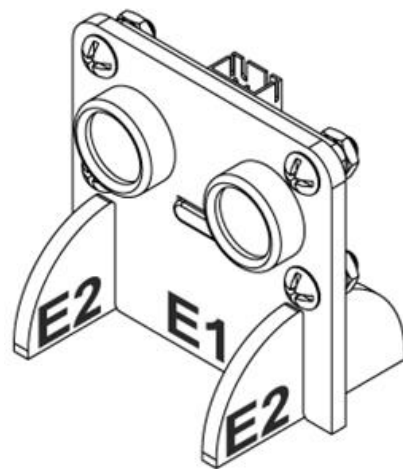
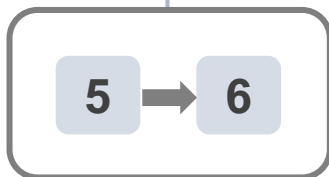
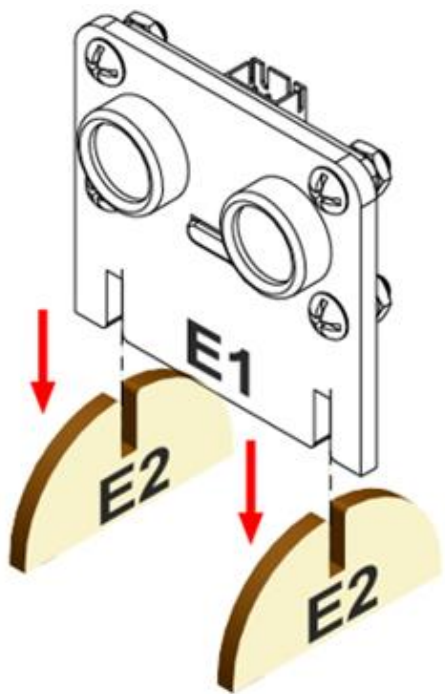


- E1 模型 x1
- 超声波距离传感器 x1
- M4 螺母 x4
- M4 *10毫米螺丝 x4



组装步骤

E2 模型 x2



编程



我们可以如何设定音效提示以警示超速？

1 當啟動時

2 OLED 初始化 宽 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

步骤:

1. 「当启动时」只会在程序最开始时运作。
2. 初始化 OLED (宽128 高64)。
3. 宣告变量 **distance1** , **distance2** , **speed** 设为0。
4. 「重复无限次」的运作方式和循环相似。
5. 呼叫新函式 calculate_Speed (用于取得 distance1 , distance2 和 speed 的数值)。
6. 如果有车靠近(**speed \geq 0**) ,
7. 在micro:bit LED的逻辑中加入**点亮直方图** 显示值为 **speed** 最大值为 20。
8. 清除显示。
9. 在OLED显示屏上展示字符串
新行显示字符串
(Distance1: **distance1**) , (Distance2:
distance2) , (Speed: **speed**) 。



The image shows a Scratch code editor with four numbered blocks (10-14) defining a function named 'calculate_Speed'. Block 10 is a 'Define Function' block. Block 11 is a 'Set Variable to' block for 'distance1' with the value '取得 量度距離 使用單位 厘米 trig接口 P14 echo接口 P15'. Block 12 is a 'Wait' block for 500 milliseconds. Block 13 is another 'Set Variable to' block for 'distance2' with the same value as block 11. Block 14 is a 'Set Variable to' block for 'speed' with the value '(distance1 - distance2) / 0.5'. The blocks are highlighted with a dashed yellow border.

步骤:

10. 宣告函式 (`calculate_Speed`) 。

11. 把 `distance1` 设为 “取得量度距离使用单位厘米 trig 接口 P14 echo 接口 P15” 。

12. 暂停0.5秒以计算下一个距离 。

13. 把 `distance2` 设为 “取得量度距离使用单位厘米 trig 接口 P14 echo 接口 P15” 。

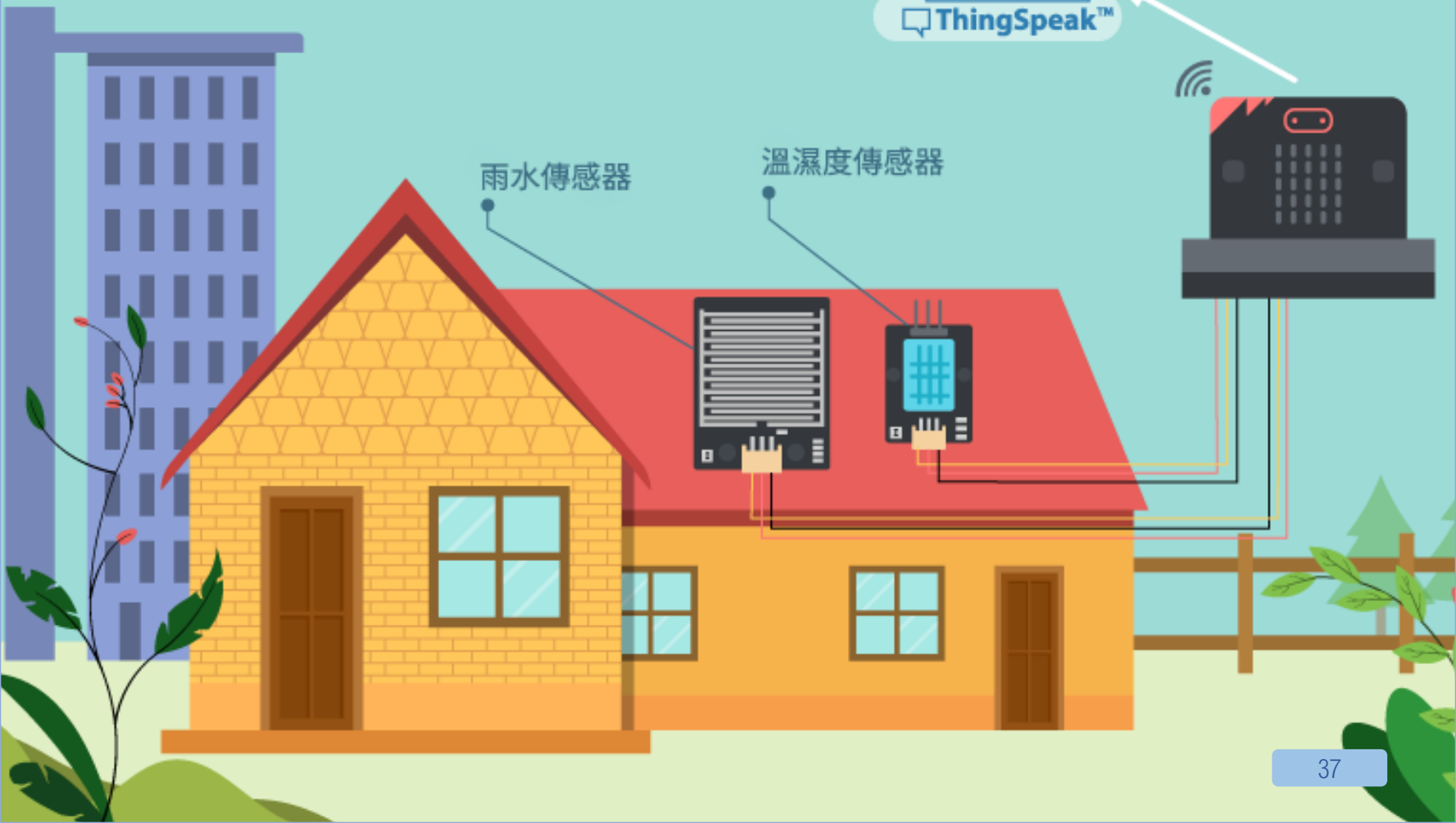
14. 把 `speed` 设为 $(\text{distance1} - \text{distance2}) / 0.5$ 。

● 智慧天氣監測站



雨水傳感器

溫濕度傳感器



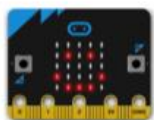
6.[物联网]智慧天气监测站

程度：★ ★ ☆ ☆ ☆

介绍

制作一个能收集雨水、温度及湿度传感器数据的智能天气监测站，并自动把数据送上ThingSpeak 平台。

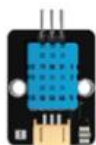
所用部件



micro:bit (1)



IoT:bit(含 OLED 显示屏) (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)



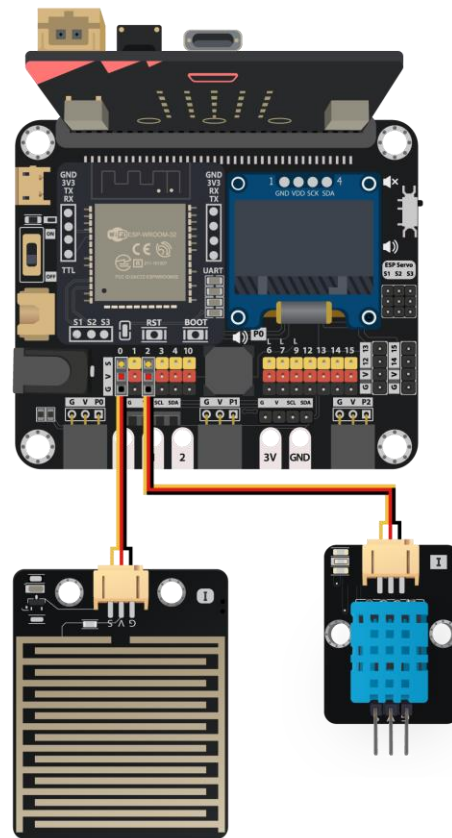
M4 螺母 (4)



螺絲批 (1)

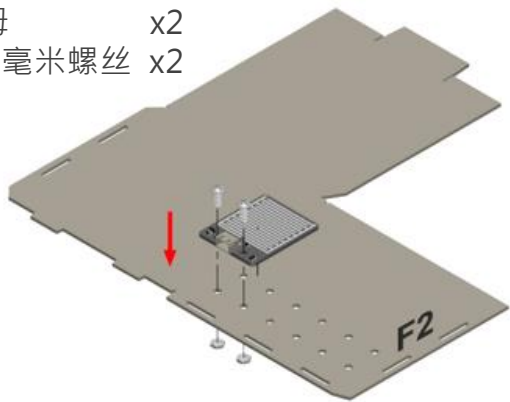


F1-F2 模型 (1)

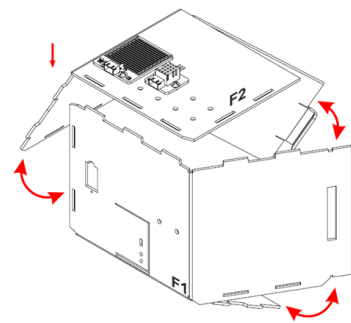
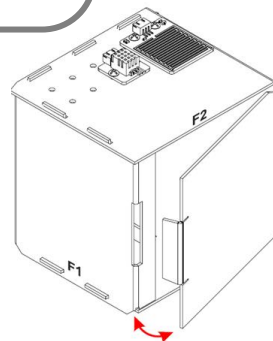
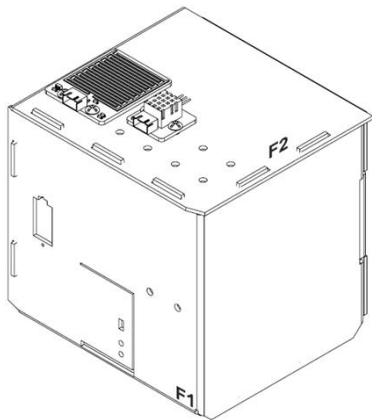
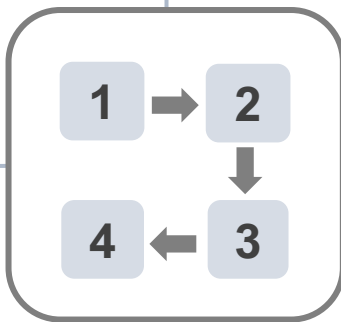
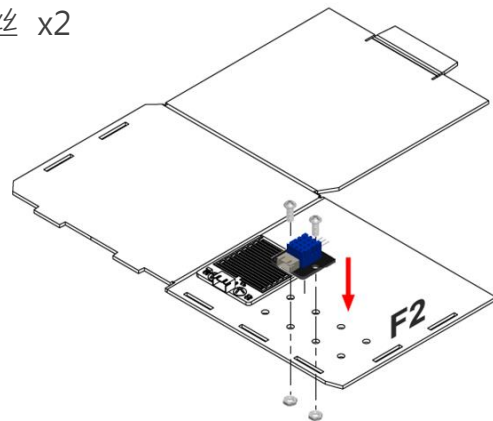


组装步骤

- F2 模型 x1
- 雨水传感器 x1
- M4螺母 x2
- M4*10毫米螺丝 x2



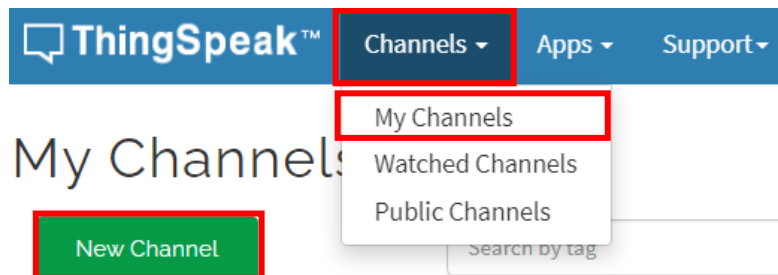
- 温湿度传感器 x1
- M4螺母 x2
- M4*10毫米螺丝 x2



F1 模型 x1

IOT (物联网) – THINGSPEAK

步骤一. 访问 <https://thingspeak.com> , 选择 Channels -> My Channels -> New Channel .



步骤二. 输入 Channel name , Field1 , Field2 , 点击 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 (物联网) – THINGSPEAK

步骤三. 选择 your channel >> API Keys · 复制 API key 。

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

1. 选择 API Keys

Write API Key

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

Generate New Write API Key

Help

API keys enable you to

2. 复制 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>

编程

💡 我们可以怎上传其他种类的数据（例如：噪音）至 ThingSpeak？

1 當啟動時

2 OLED 初始化 宽 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. 「当启动时」只会在程序最开始时运作。
2. 初始化 OLED（宽128 高64）。
3. 初始化 IoT:bit（TX P16 PX P8）。
4. 设置WiFi并输入WiFi名称及密码。
5. 宣告新变量 **temperature**、**humidity** 和 **raindrop** 并设为 0。
6. 在连上WiFi后，
7. 在OLED屏幕上显示图标「别号」。

编程



我们可以怎上传其他种类的数据（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

步骤:

8. 「重复无限次」的运作方式和循环相似。
9. 加入“如果WiFi已连接，那么”。
10. 设变数 **humidity** 为DHT11 读取湿度接口 P2，**temperature** 为DHT11 读取温度接口 P2。
11. 设变数 **raindrop** 为取得雨水传感器数值接口 P0。
12. 清除 OLED 显示。
13. 在OLED屏幕上显示字符串：字符串组合（“温度”，**temperature**），（“湿度”，**humidity**），（“雨水”，**raindrop**）。
14. 把数据传送至 ThingSpeak。
15. 暂停15秒。

编程



我们可以怎上传其他种类的数据（例如：噪音）至 ThingSpeak？

The image shows a Scratch-style code block for step 18, enclosed in a dashed yellow border. It contains three blocks:

- Block 16: "當上傳到Thingspeak後" (When uploaded to Thingspeak) with "Status" and "Error_code" variables.
- Block 17: "清除顯示" (Clear display).
- Block 18: "顯示字符串" (Show string) with "字符串組合" (String concatenation) containing "ThingSpeak: " and "Status", and "顯示字符串" (Show string) with "字符串組合" (String concatenation) containing "錯誤碼: " and "Error_code".

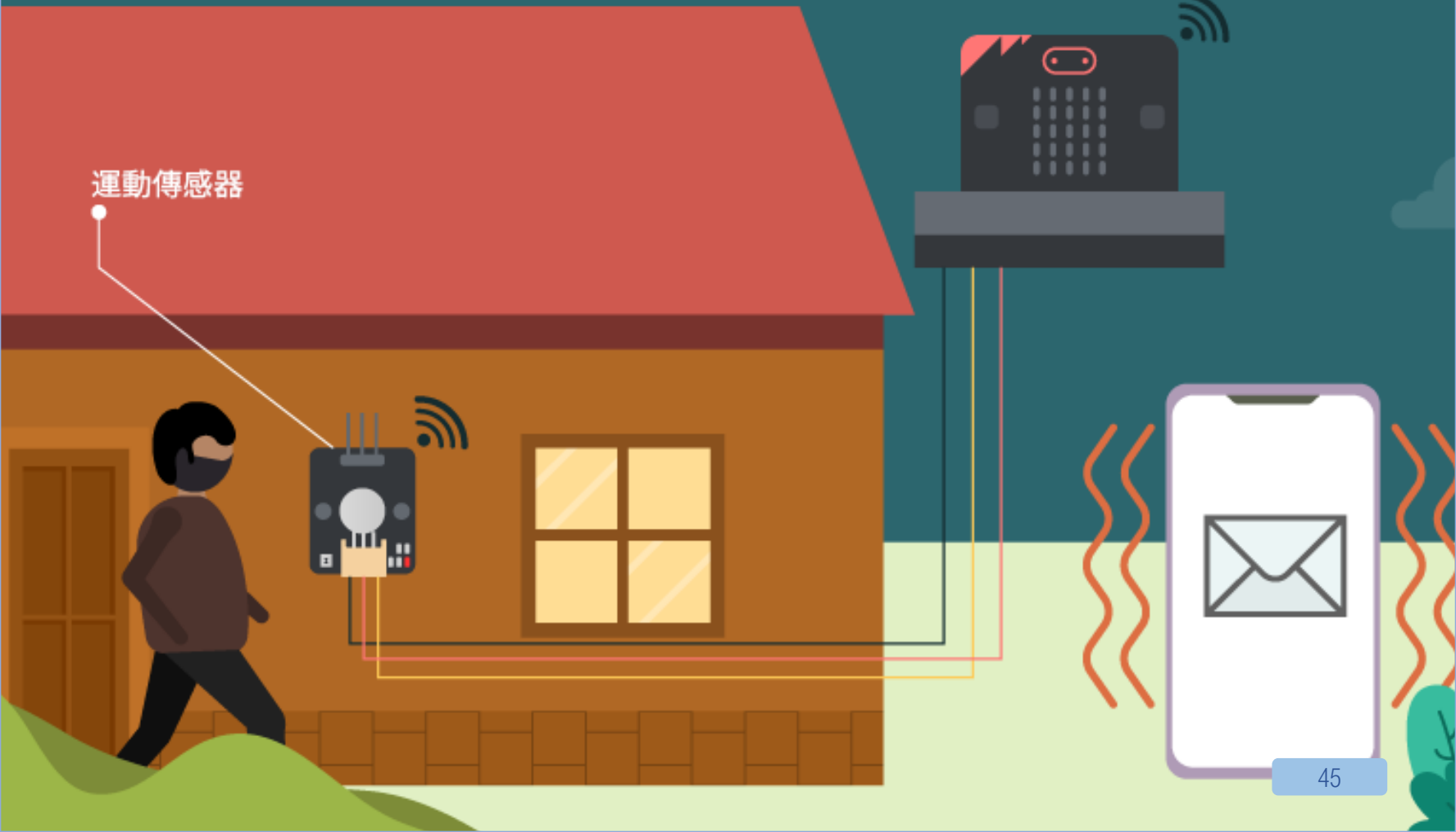
步骤:

16. 当上传到 Thingspeak 后
17. 清除OLED显示。
18. 在OLED屏幕上显示字符串：字符串组合（“ThingSpeak” ， Status ） ， （ “错误码” ， Error_code ） 。

● 智能家居防盜系統

IFTTT

運動傳感器



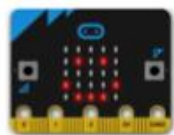
7. [物联网]智能家居防盗系统

程度：★ ★ ★ ☆ ☆

介绍

运动传感器回馈讯号至主板，若 TRUE 则主板激活蜂鸣器并透过IFTTT自动发出电邮提示用家。同时 OLED 显示屏会反映有可疑人士。

所用部件



micro:bit (1)



IoT:bit (含 OLED 显示屏) (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)

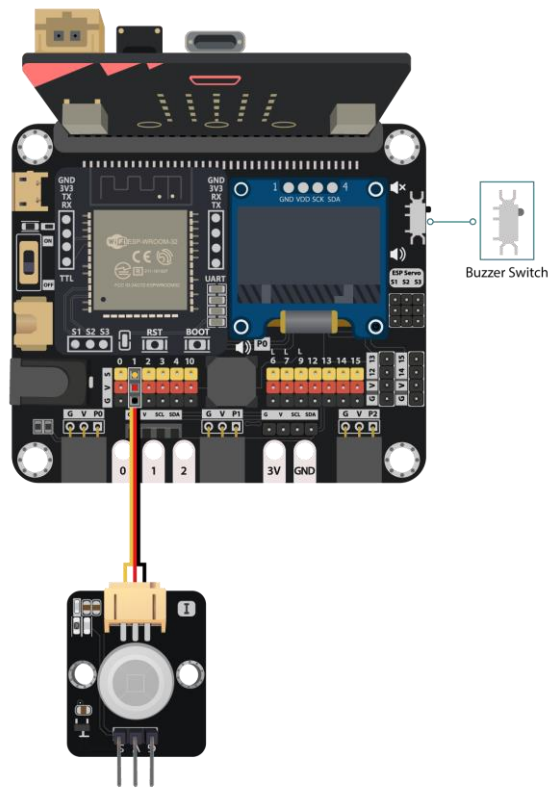


螺絲批(1)



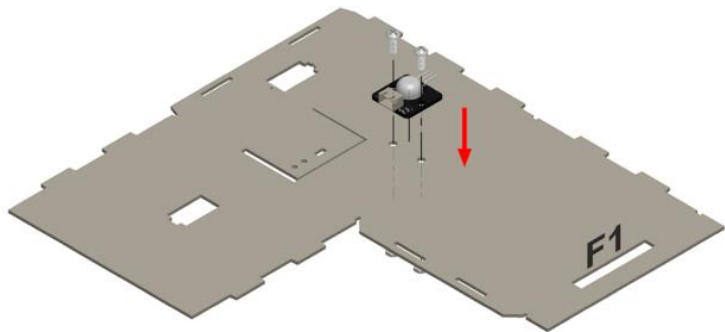
F1-F2 模型 (1)

IFTTT

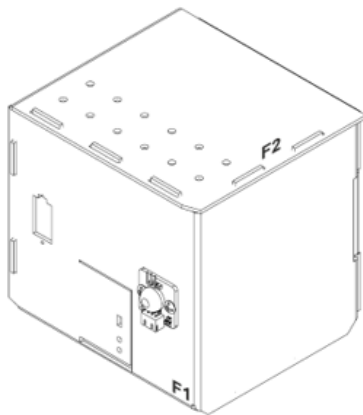
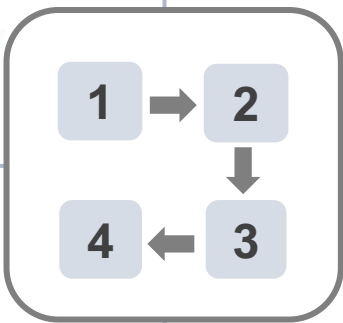
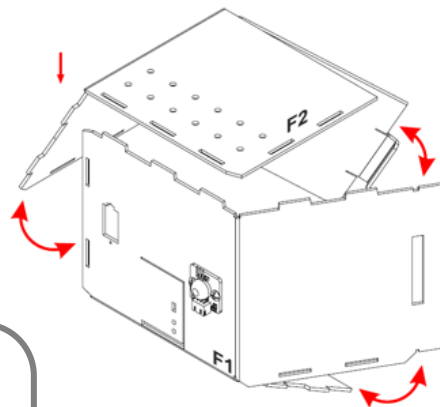


组装步骤

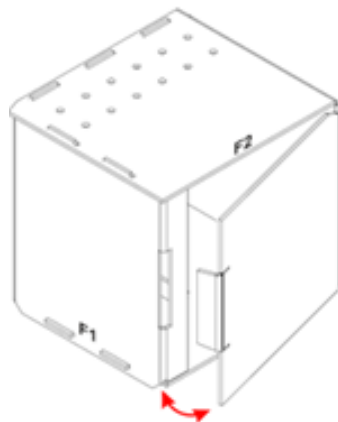
- F1 模型 x1
- 运动传感器 x1
- M4 螺母 x2
- M4*10毫米螺丝 x2



- F2 模型 x1

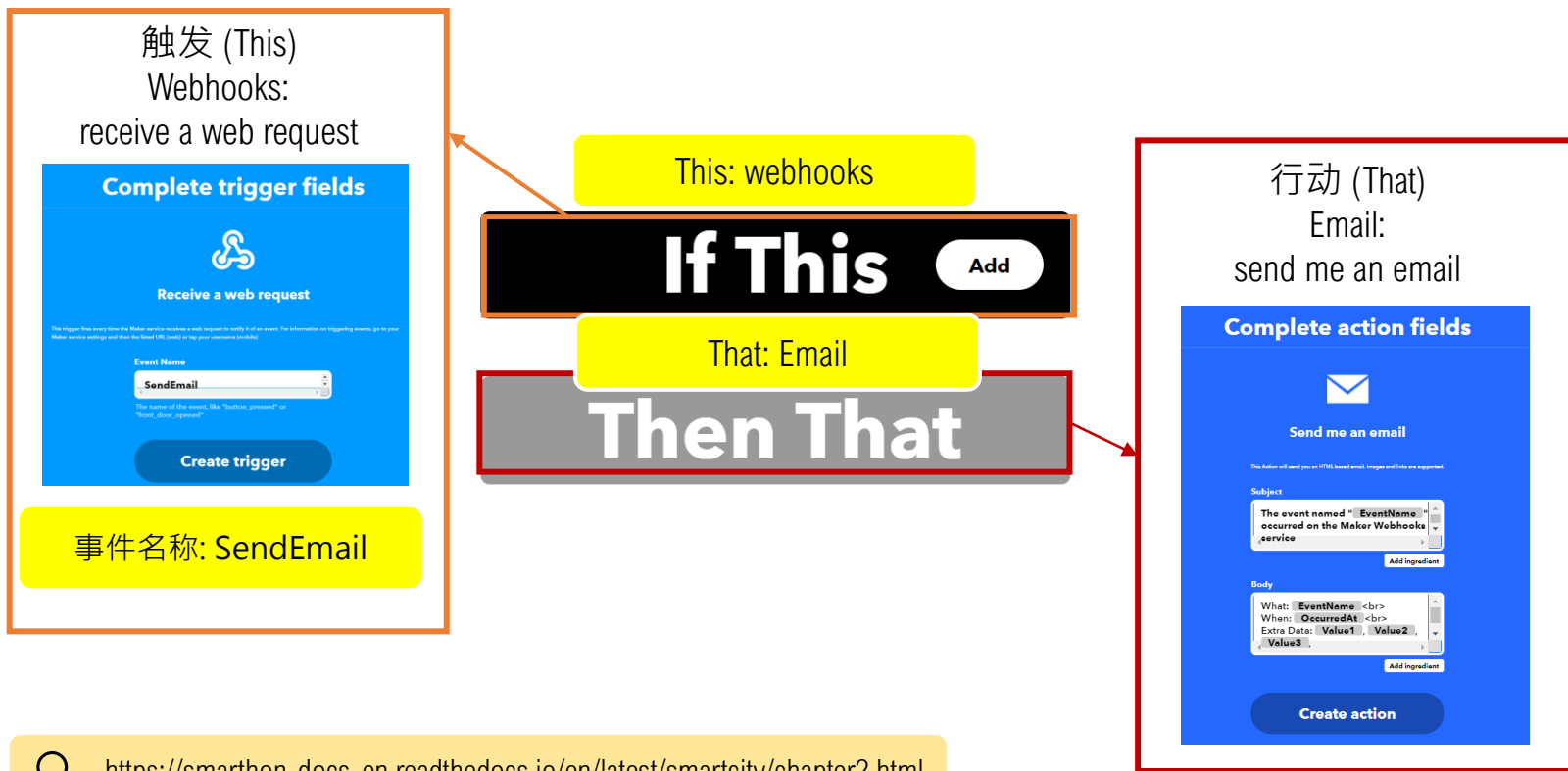


- F2 模型 x1



IOT (物联网) – IFTTT

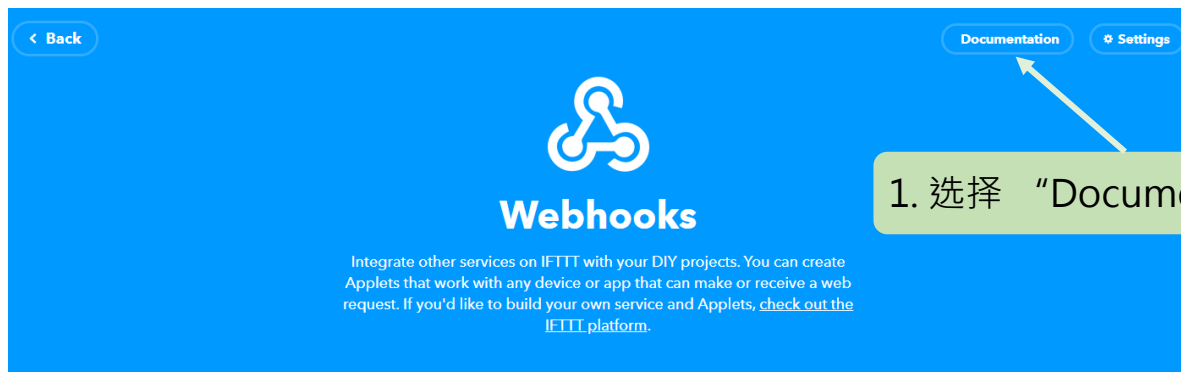
步骤 1. 访问 <https://ifttt.com> , 创建新专案 (if webhooks then Email) 。



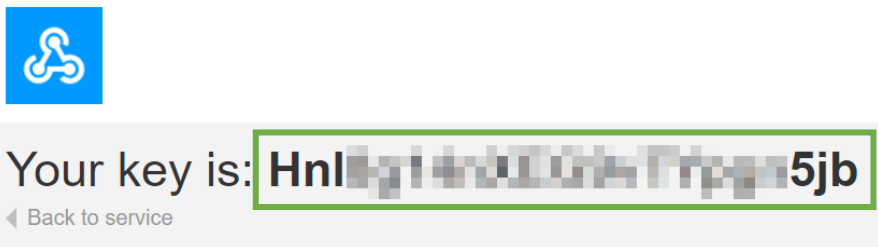
<https://smarthon-docs-en.readthedocs.io/en/latest/smartcity/chapter2.html>

IOT (物联网) – IFTTT

步骤 2. 依次点击 “My services” >> “Webhooks” , select “Documentation”
· 复制 Applet Key 。



1. 选择 “Documentation”



2. 复制这个 IFTTT key

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

编程



如何避免短时间内重复发送大量电邮？（提示：利用变量功能？）

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

9 顯示 圖示 smile

10 暫停 1000 毫秒

把 Thingspeak key 粘貼在此

步骤:

1. 「当启动时」只会在程序最开始时运作。
2. 初始化 OLED（宽128 高64）。
3. 初始化 IoT:bit（TX P16 PX P8）。
4. 设置WiFi并输入WiFi名称及密码。
5. 「重复无限次」的运作方式和循环相似。
6. 加入“如果WiFi已连接，那么”。在上述逻辑再加一个“如果...那么”，以“取得运动传感器数值接口 P1 = TRUE”为前设。
7. 在“如果...那么”加入“演奏音阶中音 C 持续 1 拍和LED 显示“Monster” 图 示”。
8. 加入“发送到 IFTTT Key*” ，填上 IFTTT Key并把事件名称设为 “SendEmail” 。
9. 在“否则”加入“smile” 图 示（没有可疑人士经过）
10. 暂停一秒

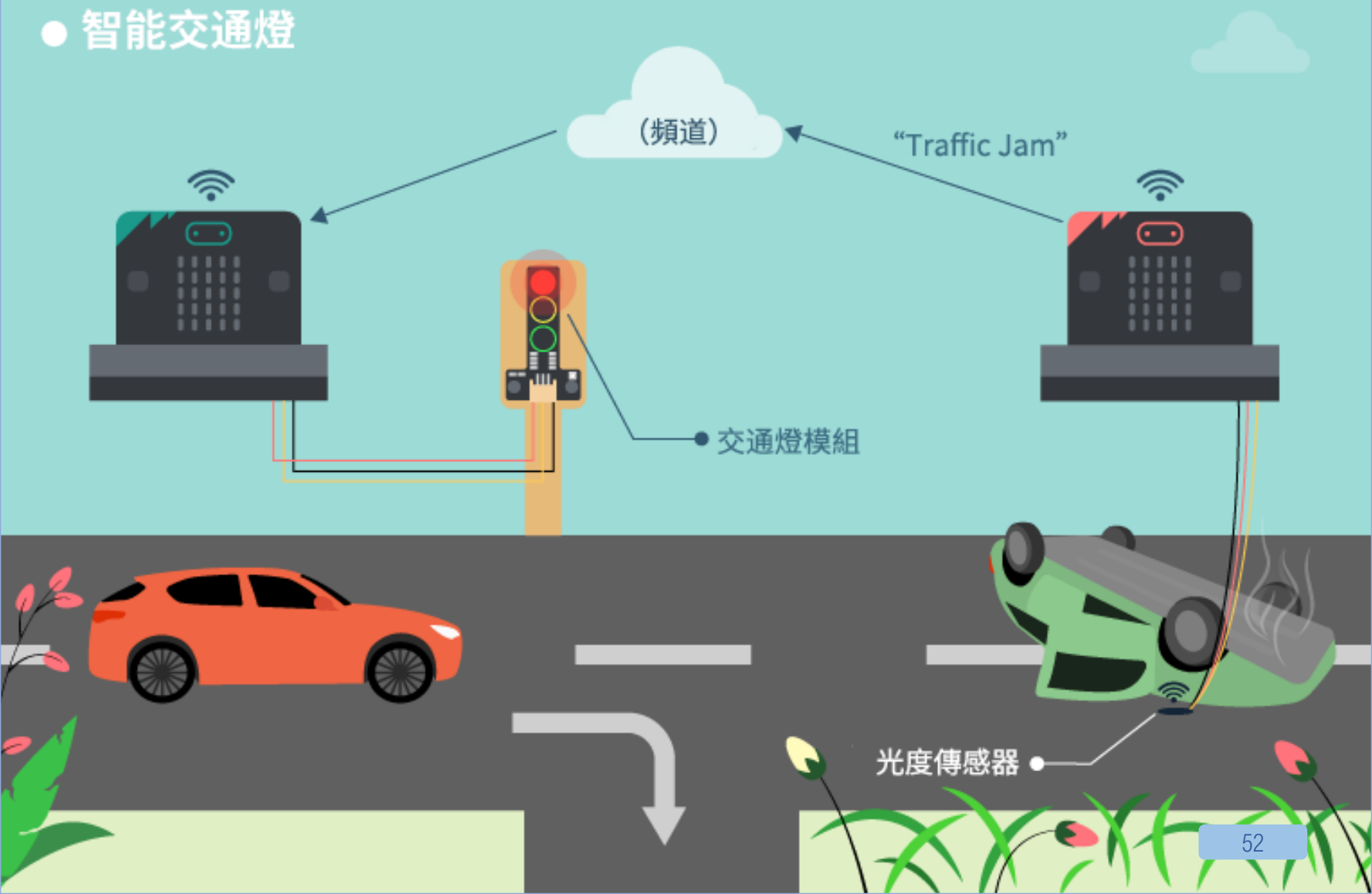


The image shows a sequence of Scratch code blocks for an IoT project. Step 11: 'When WiFi connection is successful' block with 'IP_Address' and 'Device_ID' variables. Step 12: 'Show icon' block with a WiFi icon. Step 13: 'When uploaded to IFTTT' block with 'Status' and 'Error_code' variables. Step 14: 'Clear display' block. Step 15: 'Show string' block with 'IFTTT:' and 'Status' variables. Step 16: 'Show string' block with 'Error:' and 'Error_code' variables.

步骤:

11. 如果WiFi连接成功后，
12. LED会显示“剔号”图示。
13. 当上传到 IFTTT 后
14. 清除OLED显示。
15. 在OLED屏幕上显示字符串：字符串组合（“IFTTT:” ， Status）。
16. 在OLED屏幕上显示字符串：（“Error:” ， Error_code）。

● 智能交通燈



8. [物联网] 智能交通灯（发送者）

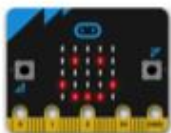
程度：★★★★★

介绍

当亮度传感器长期回馈过低数值，这代表有车停在此处，即公路上有塞车发生。发送者传送“trafficjam”至 microbit。反之则发送“nojam”。

备注：这个案例需要两组 micro:bit，一组为发送者，另一组为接收者。

所用部件



micro:bit (1)



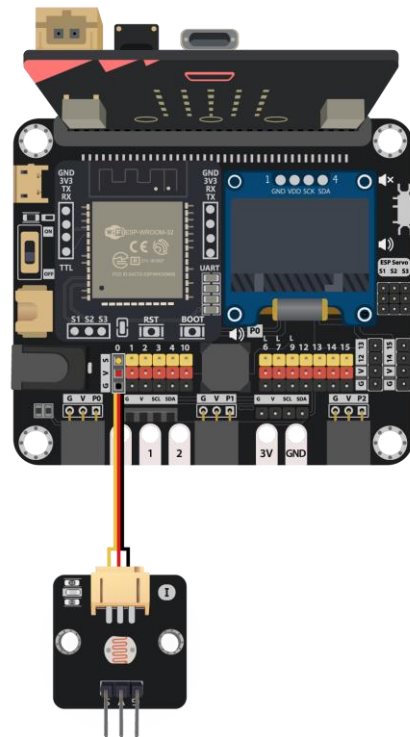
IoT:bit (含OLED 显示屏) (1)



光度传感器(1)



三脚模组连接线 (1)



编程



我们能如何使用距离传感器侦测是否有塞车的情况？

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. 「当启动时」只会在程序最开始时运作。
2. 初始化 OLED (宽128 高64)。
3. 初始化 IoT:bit (TX P16 RX P8)。
4. 设置WiFi 并输入WiFi名称及密码。
5. 宣告新变数 **light2** 并设为 0。
6. 在连上WiFi后，在OLED屏幕上显示图标「别号」。

编程



我们能如何使用距离传感器侦测是否有塞车的情况？

```
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
```

步骤:

7. 「重复无限次」的运作方式和循环相似。
8. 加入如果Wifi已连接, 将变数 **light2** 设为取得光传感器数值接口 P0。
9. 清除OLED显示。
10. 显示”字符串组合 **Light: light2**”。
11. 如果 **light2** < 10, 传送信息 “trafficjam” 到频道 “tsuenwan”。
12. 否则传送信息 “nojam” 到频道 “tsuenwan”。
13. 暂停6秒。

8. [物联网] 智能交通灯（接收者）

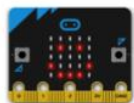
程度：★★★★★

介绍

当亮度传感器长期回馈过低数值，这代表有车停在此处，即公路上有塞车发生。发送者传送“trafficjam”至 microbit。反之则发送“nojam”。

备注: 这个案例需要两组 micro:bit，一组为发送者，另一组为接收者。

所用部件



micro:bit (1)



IoT:bit(含OLED 显示屏) (1)



交通燈模組 (1)



三腳模組連接線 (1)



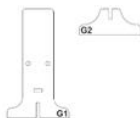
M4*10毫米螺絲(2)



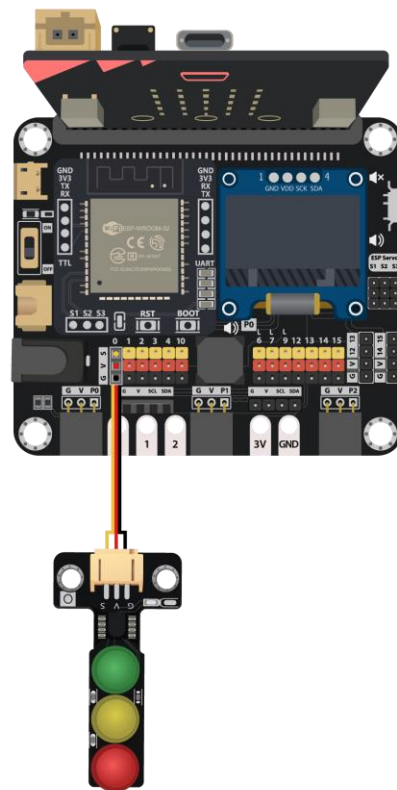
M4 螺母 (2)



螺絲批 (1)

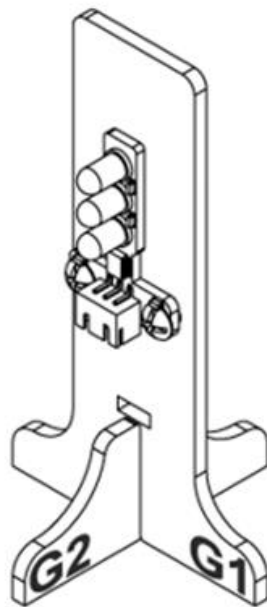
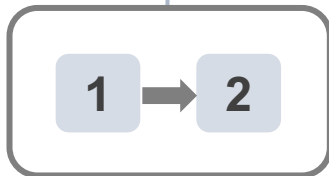
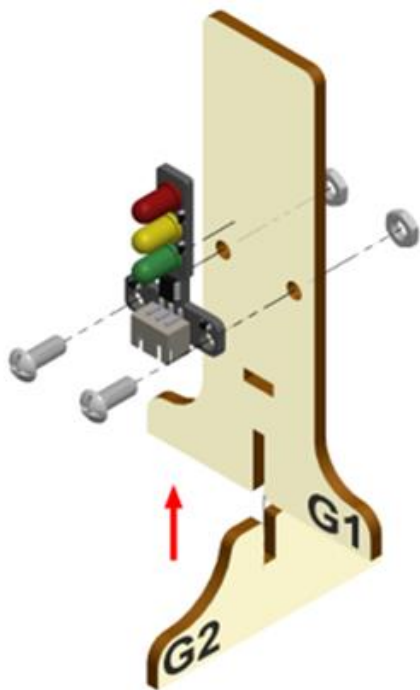


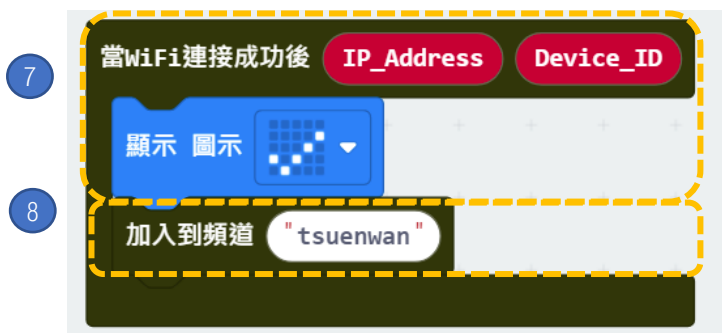
G1-G2 模型 (1)



组装步骤

- G1-G2 模型 x1
- 交通灯模块 x1
- M4 螺母 x2
- M4*10毫米螺丝 x2

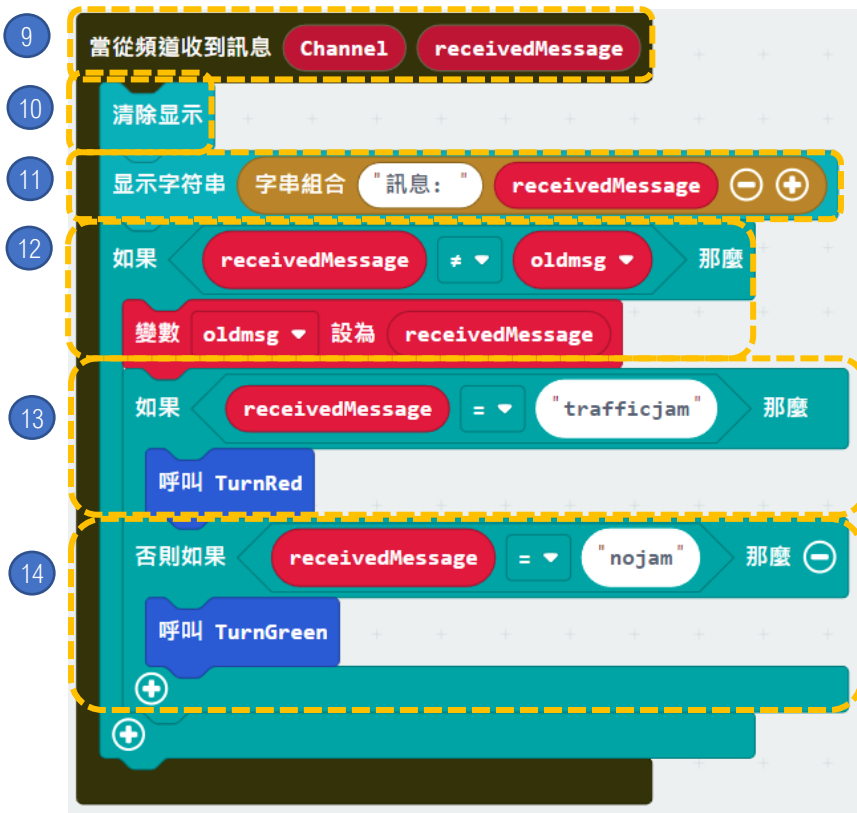




步骤:

1. 「当启动时」只会在程序最开始时运作。
2. 初始化 OLED (宽128 高64)。
3. 初始化 IoT:bit (TX P16 PX P8)。
4. 设置WiFi 并输入WiFi名称及密码。
5. 宣告新变数 **oldmsg** 并设为 ""。
6. 加入呼叫 **TurnGreen**。

7. 在连上WiFi后，在OLED屏幕上显示图标「别号」。
8. 加入到频道 "tsuenwan"。



步骤:

9. 加入当收到WiFi频道内容时，
10. 清除OLED显示。
11. 显示字符串字符串组合：“Message: , receivedMessage”。
12. 如果 receivedMessage \neq oldmsg，那么把变数 oldmsg 设置成 receivedMessage。
13. 如果 receivedMessage 等于“trafficjam”，呼叫TurnRed。
14. 否则如果 receivedMessage 等于“nojam”，呼叫TurnGreen。



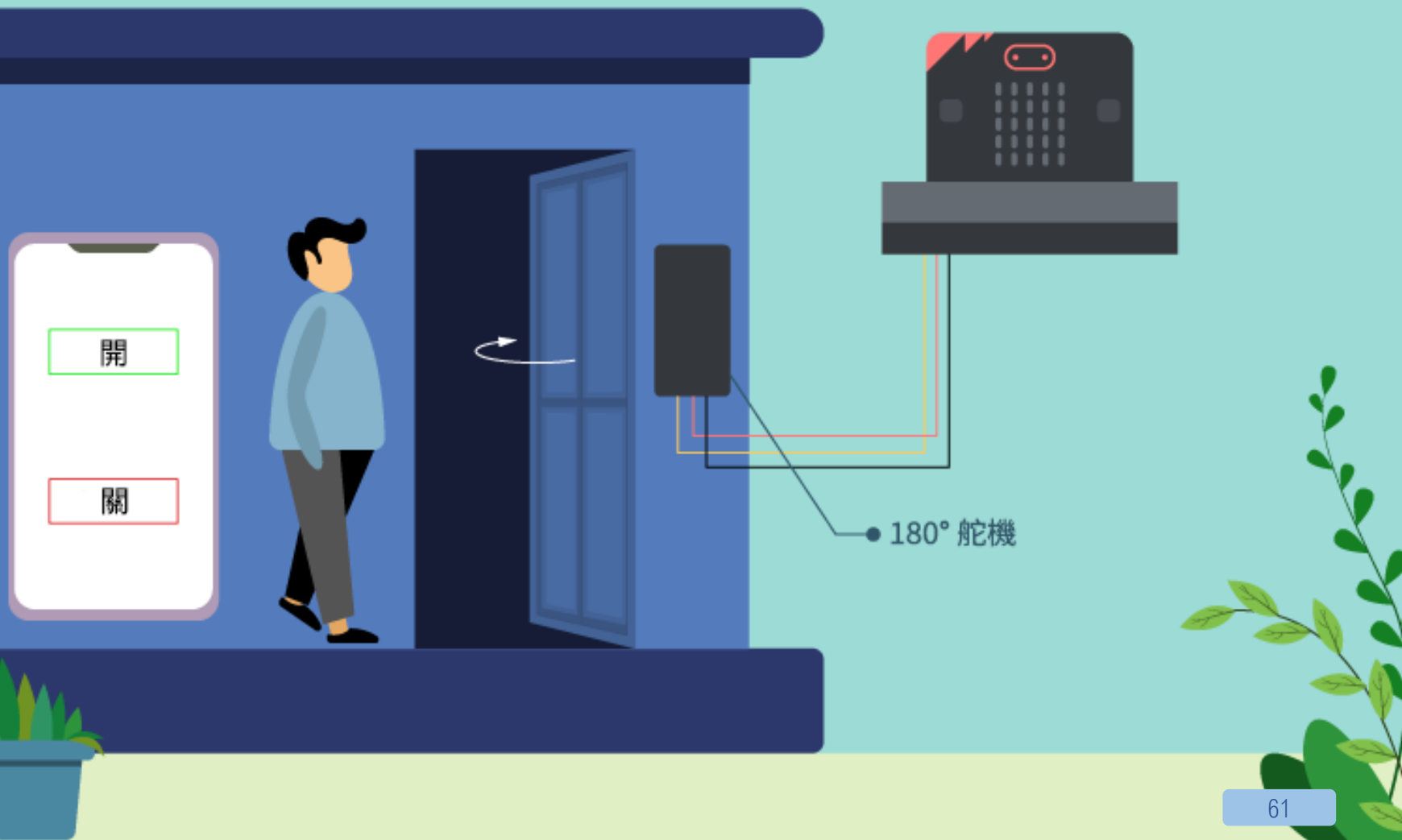
15



STEPS:

15. 宣告两个新函式(TurnRed 及 TurnGreen)。
在TurnRed函式中加入“Control traffic light at P1”，等待两秒，转颜色为黄色，等待两秒，转颜色为红，等待两秒。在TurnGreen函式中加入“Control traffic light at P1”，转为红色，等待两秒，转为黄色，等待两秒，转为绿色，等待两秒。

● 遙距智能門控制



9. [物联网] 遥距智能门控制

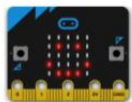
程度：★★★★☆

介绍

当 microbit 收到“ opendoor” 讯号，舵机会自动转动。当收到“ closedoor” ，门会自动关闭。



所用部件



micro:bit (1)



IoT:bit (含OLED 显示屏) (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)



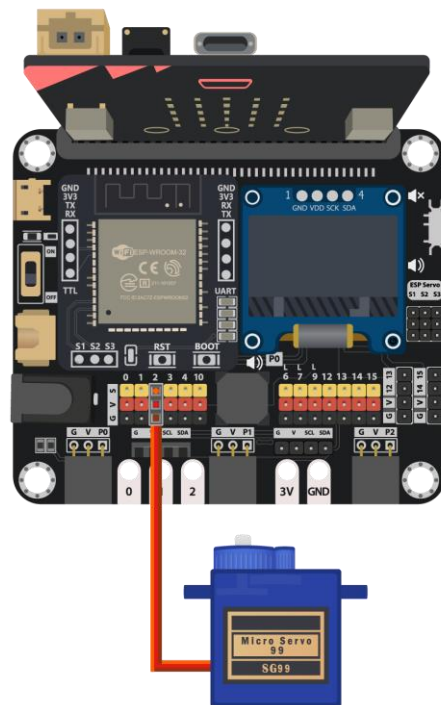
M3 螺母(1)



螺絲批 (1)

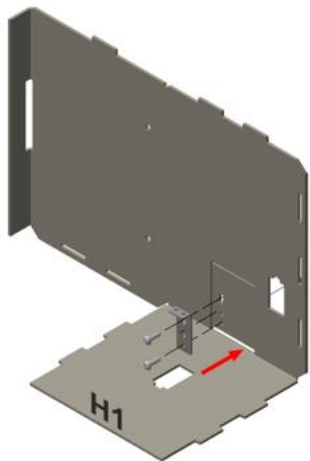


H1-H7模型 (1)

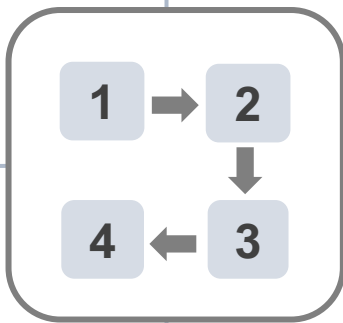
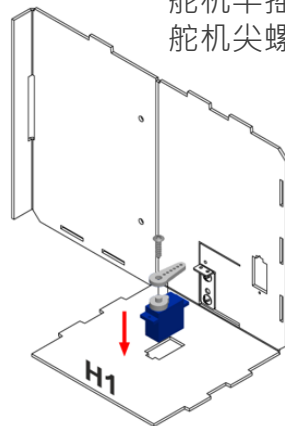


组装步骤

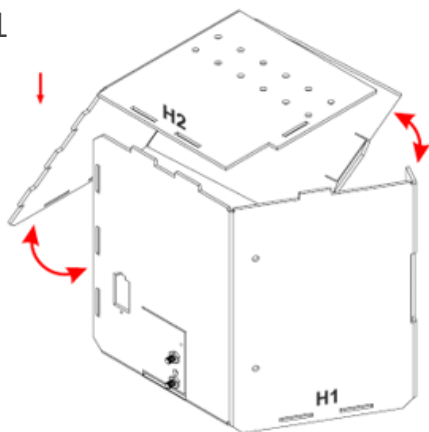
- L型支架 x1
- M4螺母 x2
- M4*10毫米螺丝 x2
- H1 模型 x1



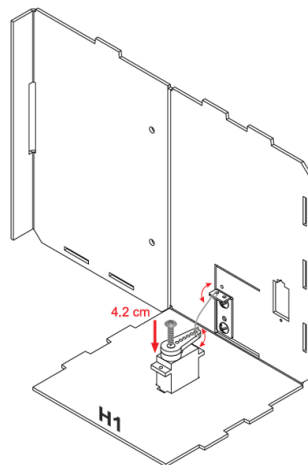
- 180°舵机 x1
- 舵机半摇臂 x1
- 舵机尖螺丝 M2*7.5毫米 x1



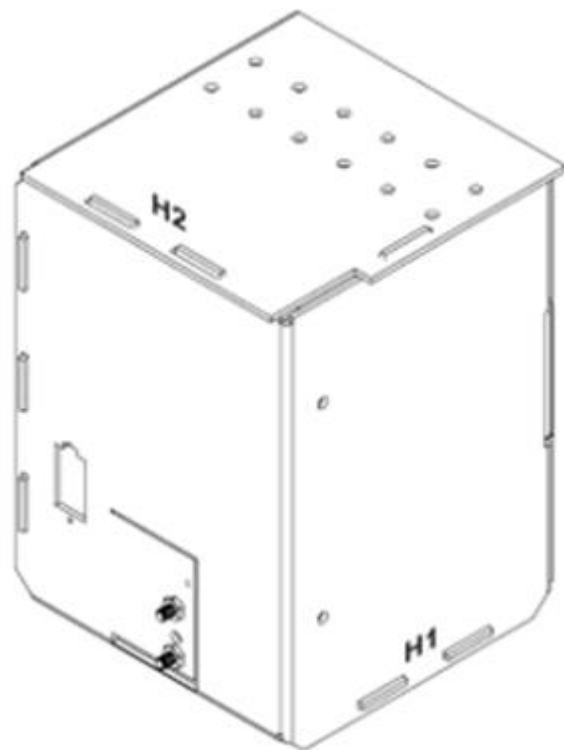
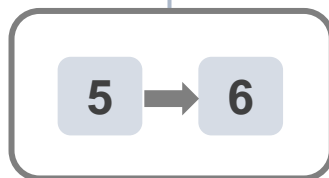
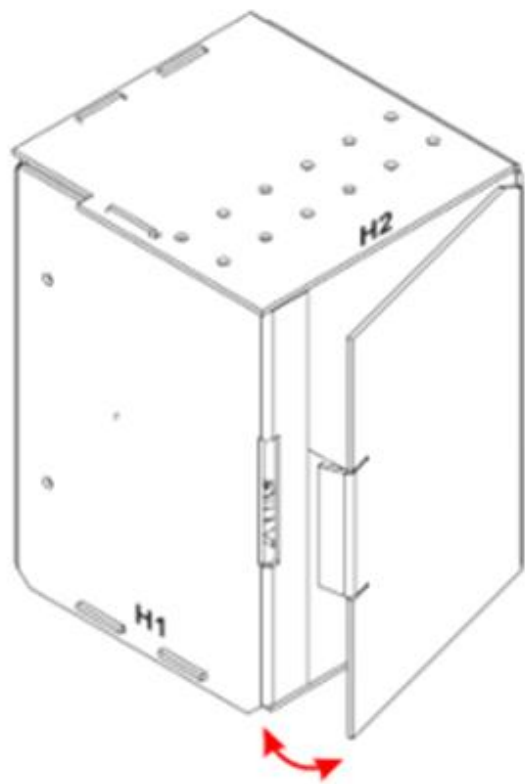
- H2 模型 x1



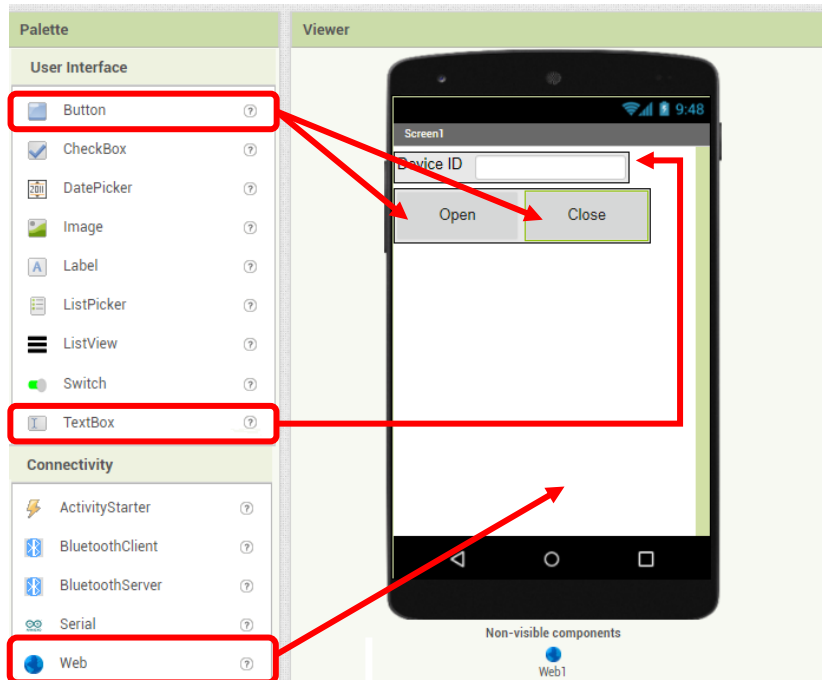
- 回形针 x1



组装步骤



IOT (物联网) – APP INVENTOR 2



步骤一. 创建新页面

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



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

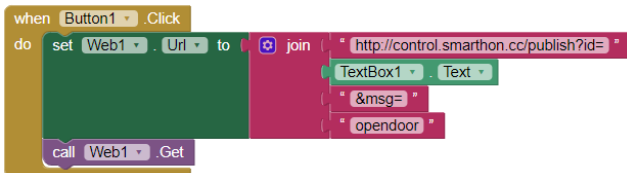
IOT (物联网) – APP INVENTOR 2

编程二. 编程

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

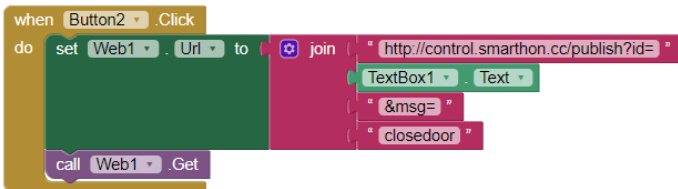
<http://control.smarthon.cc/publish?id=DeviceID&msg=ControlCommand>

- 当 button1 被点击，以下指令“ opendoor” 会被执行。



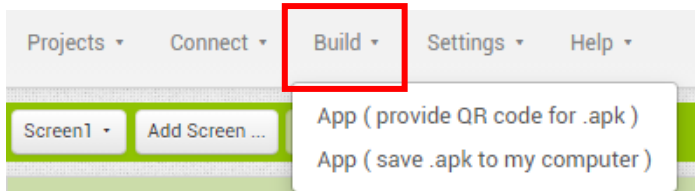
```
when Button1 Click
do
  set Web1 . Url to join ( "http://control.smarthon.cc/publish?id="
    TextBox1 . Text
    "&msg="
    "opendoor"
  )
  call Web1 . Get
```

- 当 button2 被点击，以下指令“ closedoor” 会被执行



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

- 透过QR code或 apk，下载程序至手机。





步骤:

1. 「当启动时」只会在程序最开始时运作。
2. 初始化 OLED (宽128 高64)。
3. 初始化 IoT:bit (TX P16 PX P8)。
4. 设置WiFi 并输入WiFi名称及密码。
5. 转动舵机到 180 度 接口 P2。



6. 在连上WiFi后，在OLED屏幕上显示图标「别号」。
7. 显示字符串 **Device_ID**。



The image shows a sequence of code blocks in a programming environment, numbered 8 through 12. The blocks are as follows:

- 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

步骤:

8. 当从互联网接收到指令时，
9. 清除OLED显示。
10. 显示字符串字符串组合：“Command: ”，
“WAN_Command”。
11. 如果 WAN_Command = “opendoor”，
转动舵机到45度 接口P2。
12. 如果 WAN_Command = “closedoor”，
转动舵机到180度 接口P2。

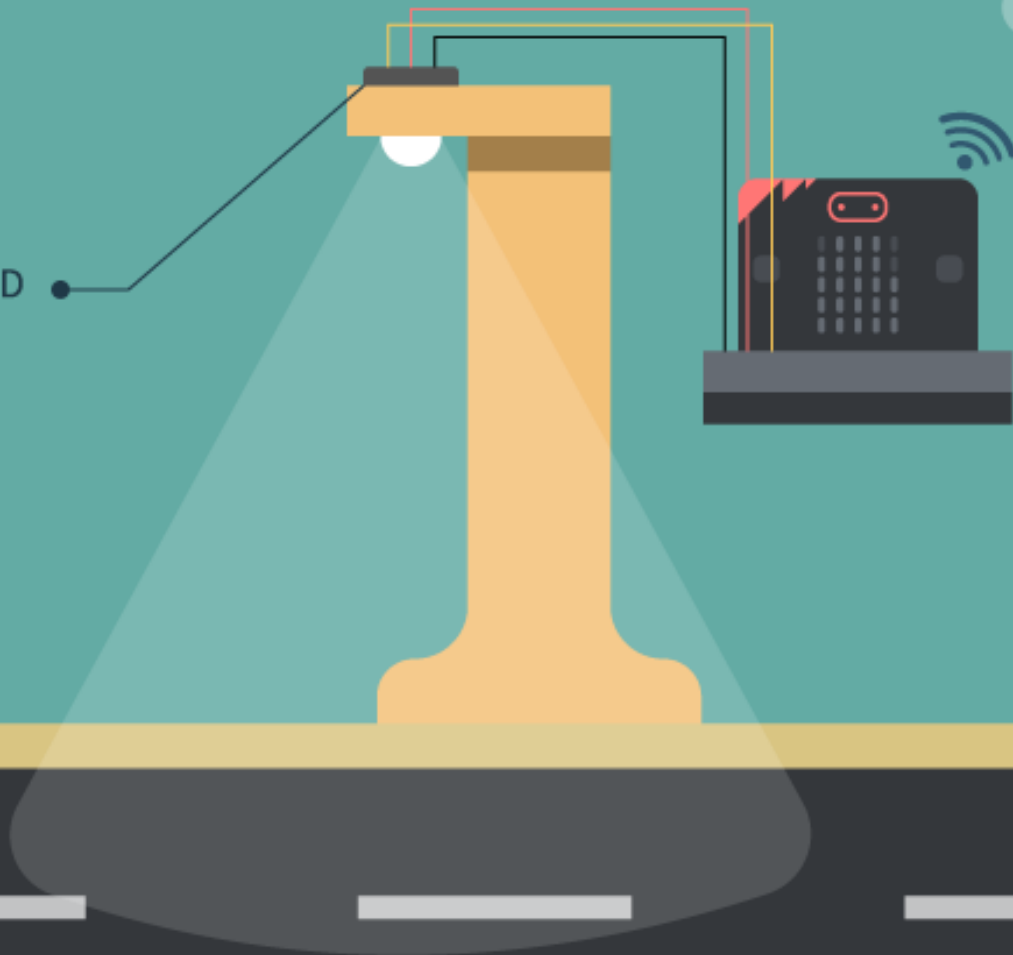
● 節能街燈



06:00 PM

IFTTT

白色LED



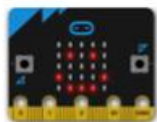
10. [物联网]节能街灯

程度：★ ★ ★ ★ ☆

介绍

节能街灯可以在指定时间自动开关的街灯，能够同时兼顾节省能源及照明需求。

所用部件



micro:bit (1)



IoT:bit (含 OLED 显示屏)(1)



白色LED (1)



三脚模組連接線(1)



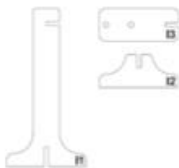
M4*10毫米螺絲(2)



M4 螺母(2)

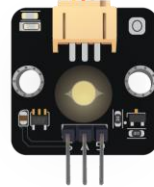
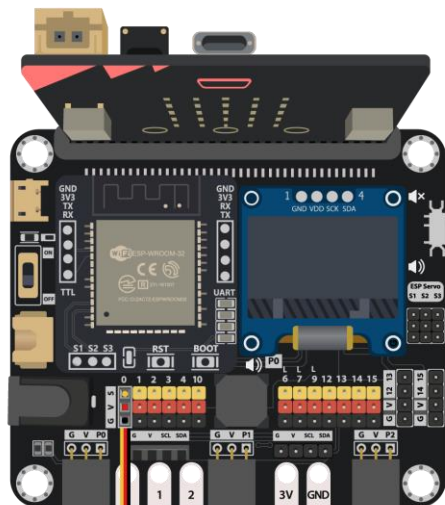


螺絲批 (1)



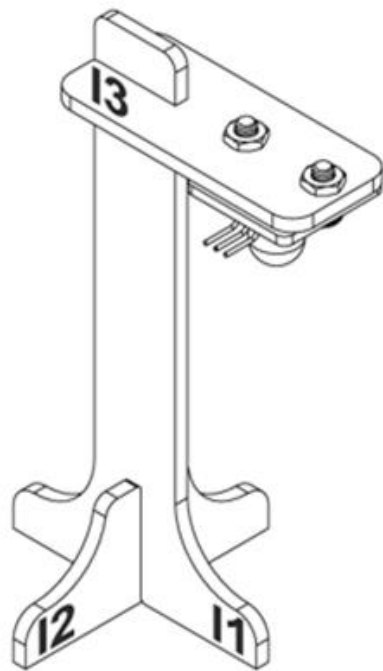
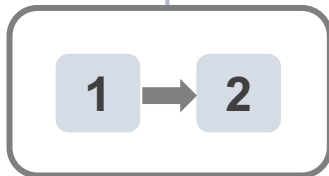
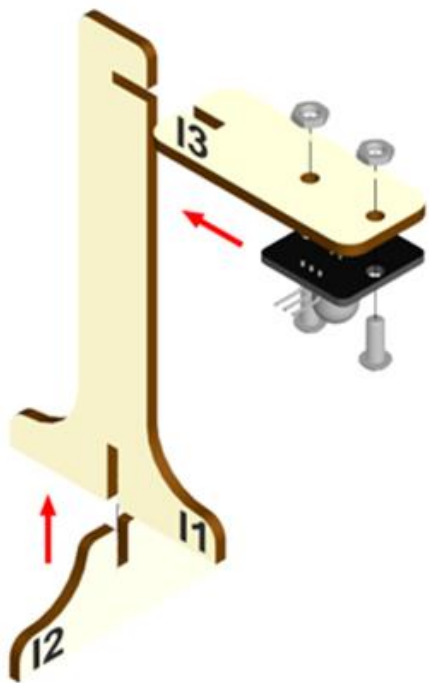
I1-I2 模型 (1)

IFTTT



组装步骤

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



编程



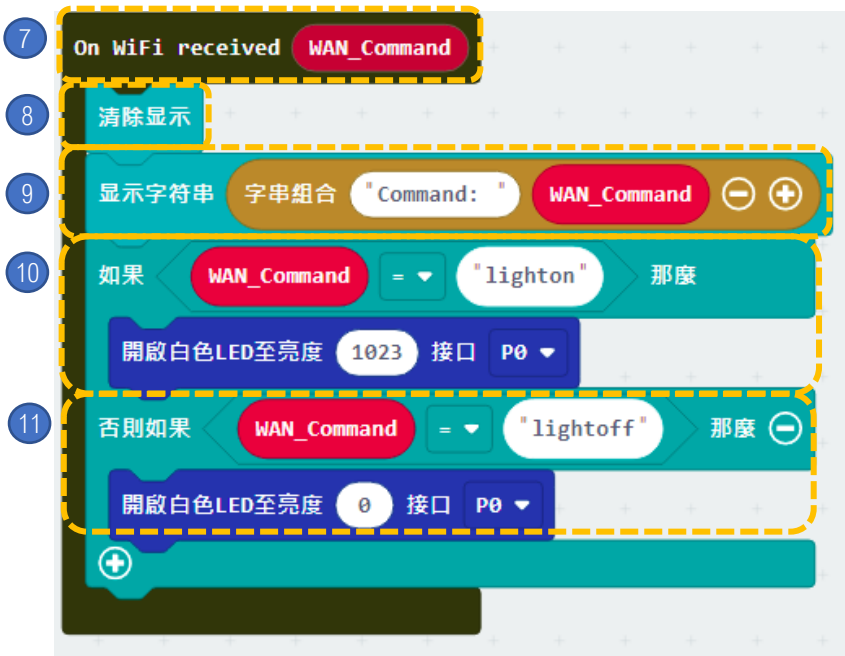
如果今天是阴天，如何用IFTTT通知系统打开街灯？

The image shows a sequence of six code blocks in a Scratch-like environment, numbered 1 to 6. The blocks are as follows:

1. 當啟動時 (When started)
2. OLED 初始化 寬 128 高 64 (Initialize OLED width 128 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 name "smarthon" password "12345678")
5. 當WiFi連接成功後 IP_Address Device_ID (When WiFi connection is successful IP_Address Device_ID)
 - 顯示 圖示 (Display icon)
6. 顯示字符串 Device_ID (Display string Device_ID)

步骤:

1. 「当启动时」只会在程序最开始时运作。
2. 初始化 OLED (宽128 高64)。
3. 初始化 IoT:bit (TX P16 PX P8)。
4. 设置WiFi 并输入WiFi名称及密码。
5. 在连上WiFi后，在OLED屏幕上显示图标「别号」。
6. 显示字符串 **Device_ID**。

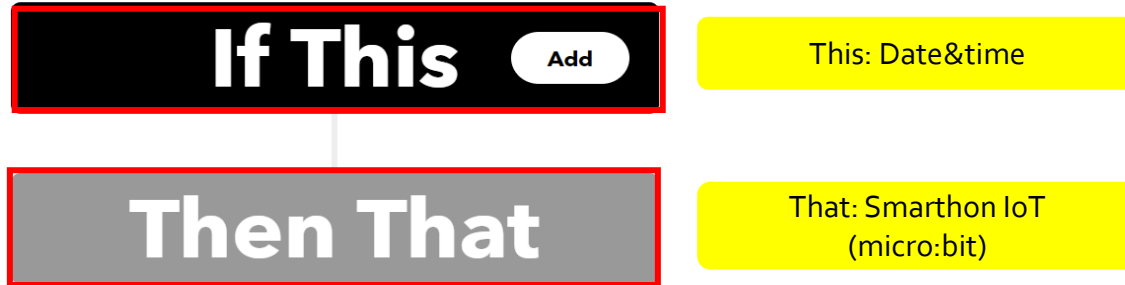


步骤:

7. 当从互联网接收到指令时，
8. 清除OLED显示。
9. 显示字符串字符串组合：“Command: , WAN_Command”。
10. 如果 WAN_Command = “lighton”，开启白LED至亮度1023接口P0。
11. 如果 WAN_Command = “lightoff”，开启白LED至亮度0接口P0。

IOT (物联网) – IFTTT 云端控制

步骤一. 在 IFTTT 开立新专案(If Date&time Then Smarthon IoT (micro:bit))

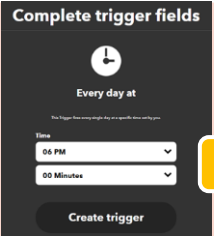
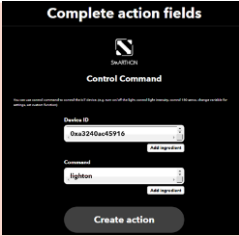


<https://smarthon-docs-en.readthedocs.io/en/latest/smartcity/chapter4.html>

IOT (物联网) – IFTTT 云端控制

步骤二.

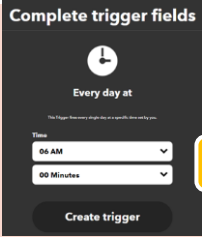
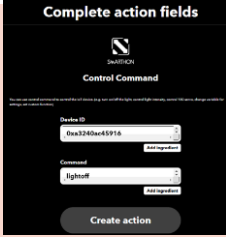
•如果现在时间为18:00， 传送 WAN控制指令: lighton 至 micro:bit

触发 (This): Date&time Every day at	行动 (That): Smarhon IoT (micro:bit) Control Command
 <p>Complete trigger fields</p> <p>Every day at</p> <p>Time: 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: 0xa3240ac45916</p> <p>Command: lighton</p> <p>Device ID</p> <p>lighton</p>

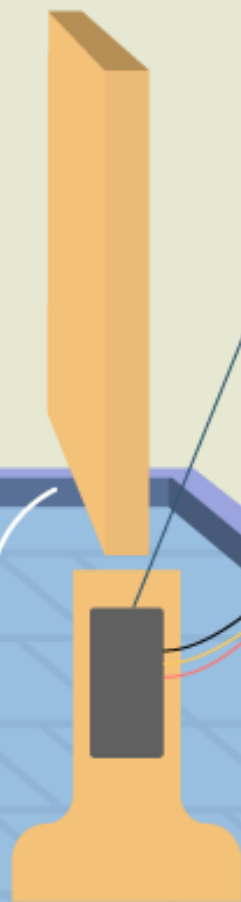
* 在 micro:bit 上运行程序以获取你的 device ID



•如果现在时间为06:00， 传送 WAN控制指令: lightoff 至 micro:bit

触发 (This): Date&time Every day at	行动 (That): Smarhon IoT (micro:bit) Control Command
 <p>Complete trigger fields</p> <p>Every day at</p> <p>Time: 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: 0xa3240ac45916</p> <p>Command: lightoff</p> <p>Device ID</p> <p>lightoff</p>

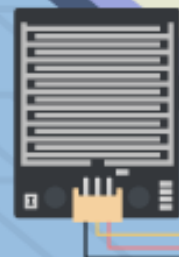
● 自動天台簷篷系統



● 180° 舵機



● 雨水傳感器



11. [物联网] 自动天台檐篷系统

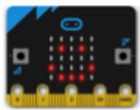
程度：★ ★ ★ ★ ☆

介绍

当雨水传感器侦测到下雨，就会传送讯号至micro:bit 以开启天台檐篷。当没有下雨时，天台檐篷会被关上。

同时，micro:bit通过IFTTT平台上的天气预报系统，取得指令内容开关檐篷。

所用部件



micro:bit (1)



IoT:bit (含OLED 显示屏) (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)

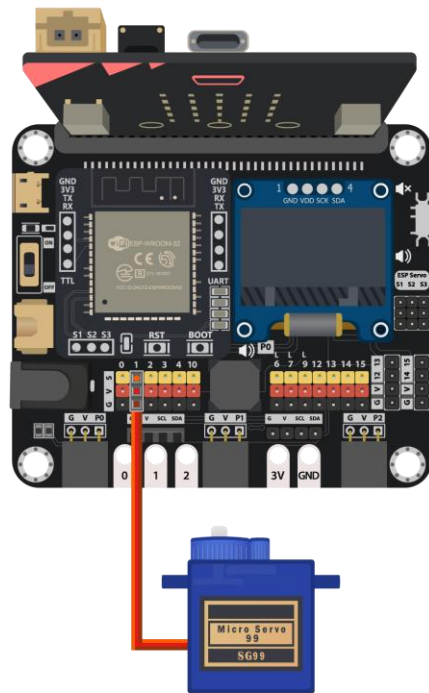


螺絲批 (1)



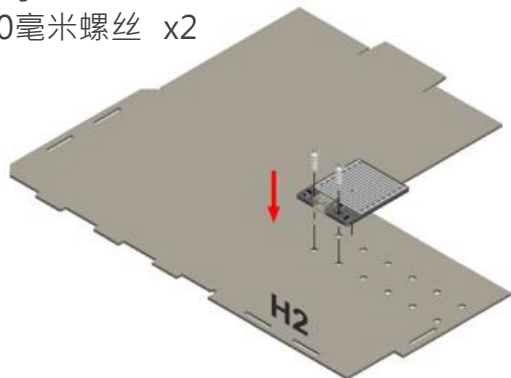
H1-H7 模型 (1)

IFTTT

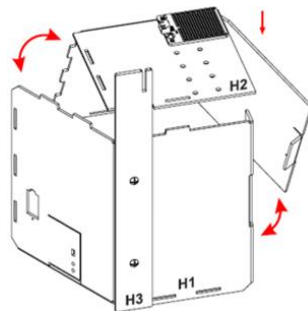
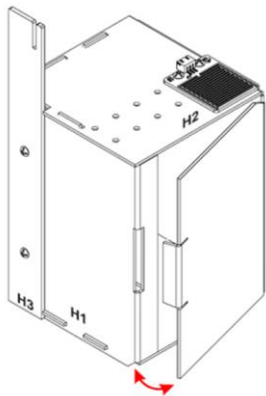
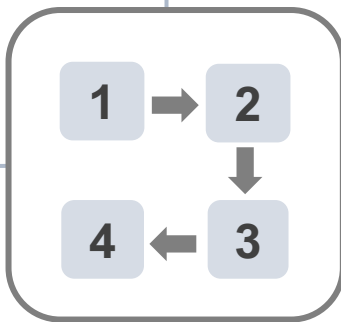
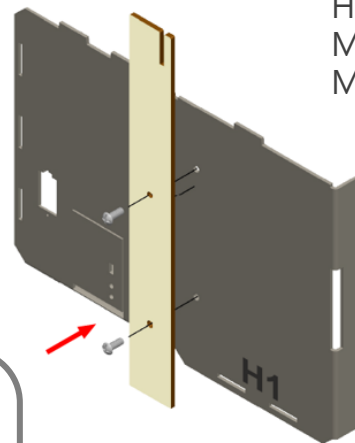


组装步骤

- H2 模型 x1
- 雨水传感器 x1
- M4 螺母 x2
- M4*10毫米螺丝 x2

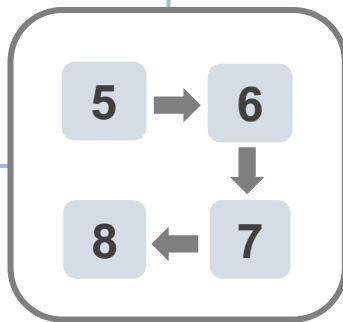
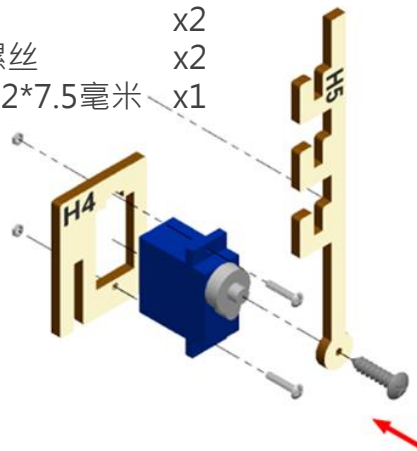


- H1 & H3 模型 x1
- M4 螺母 x2
- M4*10毫米螺丝 x2

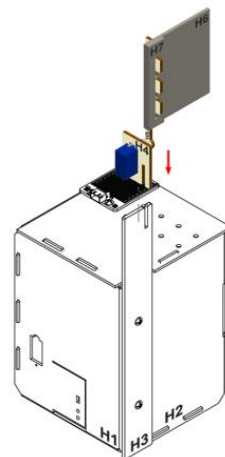
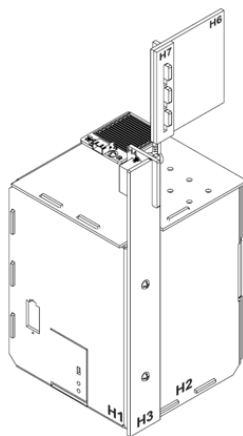
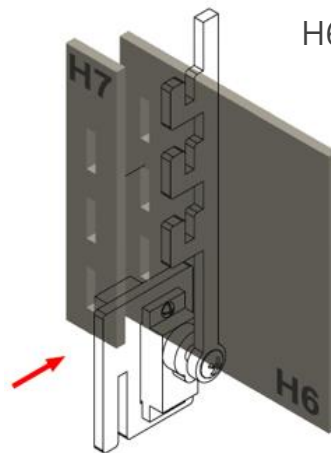


组装步骤

- H4 - H5 模型 x1
- 180° 舵机 x1
- M2 螺母 x2
- M2*10毫米螺丝 x2
- 舵机尖螺丝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 宽 128 高 64 (Initialize OLED width 128 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 name "smarthon" password "12345678")
5. 當WiFi連接成功後 (When WiFi connection is successful)
 - 顯示 圖示 (Display icon)
6. 顯示字符串 Device_ID (Display string Device_ID)

步骤:

1. 「当启动时」只会在程序最开始时运作。
2. 初始化 OLED（宽128 高64）。
3. 初始化 IoT:bit（TX P16 PX P8）。
4. 设置WiFi 并输入WiFi名称及密码。
5. 在连上WiFi后，在OLED屏幕上显示图标「别号」。
6. 显示字符串 **Device_ID**。

编程



你能用其他天气状况控制天台檐篷吗？（例如：阳光）

```
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
```

步骤:

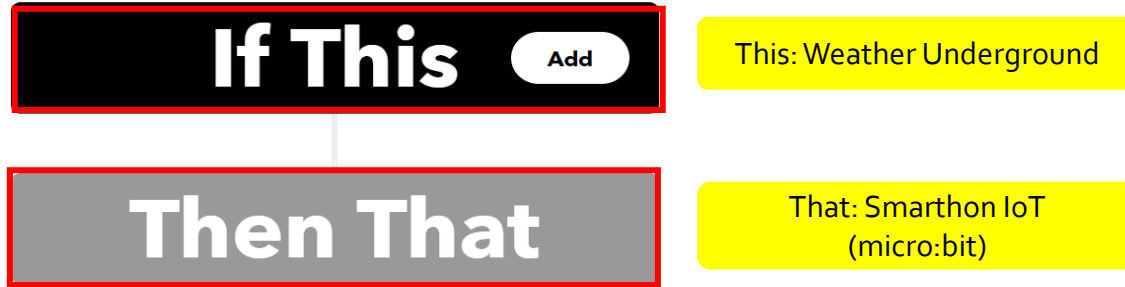
7. 当从互联网接收到指令时，
8. 清除OLED显示。
9. 显示字符串字符串组合：“Command: , WAN_Command”。
10. 如果 WAN_Command = “Rain”，转动舵机到90度接口P1。
11. 如果 WAN_Command = “Clear”，转动舵机到180度接口P1。


12. 当按钮A被按下，
13. 转动舵机到90度接口P1。

14. 当按钮B被按下，
15. 转动舵机到180度接口P1。

IOT (物联网) – IFTTT 云端控制

步骤一. 在 IFTTT 开立新专案(If Weather Underground Then Smarthon IoT (micro:bit))

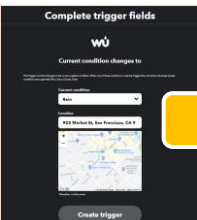


 <https://smarthon-docs-en.readthedocs.io/en/latest/smartcity/chapter4.html>


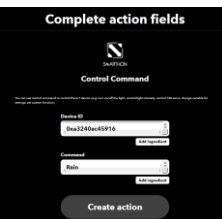
IOT (物联网) – IFTTT 云端控制

步骤二.

- 如果现在天气转变为下雨， 传送 WAN控制指令: Rain 至 micro:bit

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

- 如果现在雨势已经停止， 传送 WAN控制指令: Clear 至 micro:bit

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

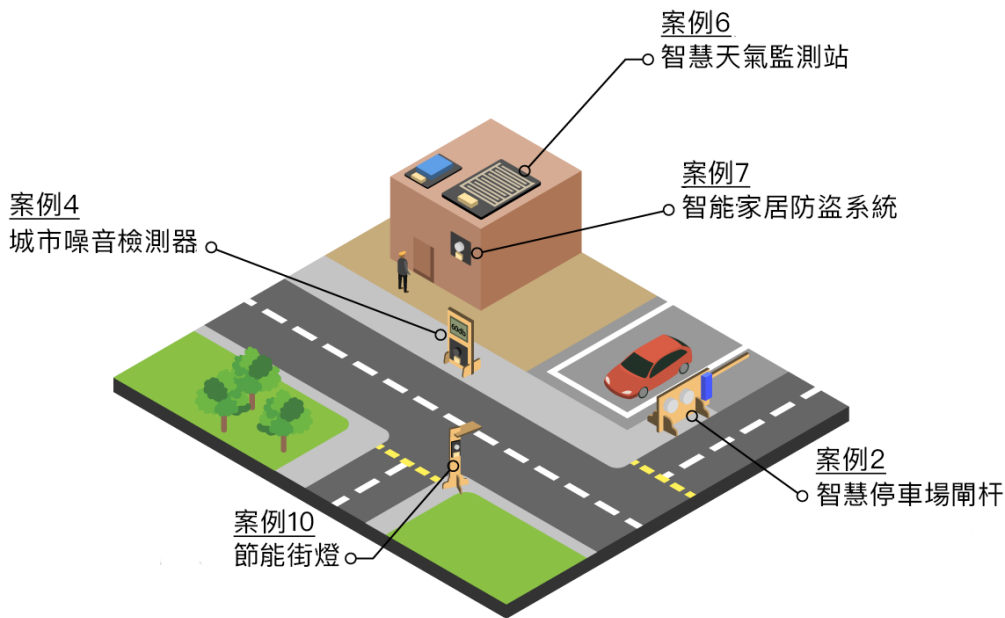
* 在 micro:bit 上运行程序以获取你的 device ID



场境示例 1:

智慧城市生活

智慧城市生活侧重于实现市民的高生活水平，它包括停车场自动化系统和高度安全的防御系统，收集噪声和天气信息以评估城市的环境舒适性。



部件:

- 温湿度传感器(DHT11)
- 雨水传感器
- 运动传感器
- 蜂鸣器
- 亮度传感器
- 180° 舵机
- 超声波距离传感器
- 声音传感器
- 白色LED

IOT(物联网):

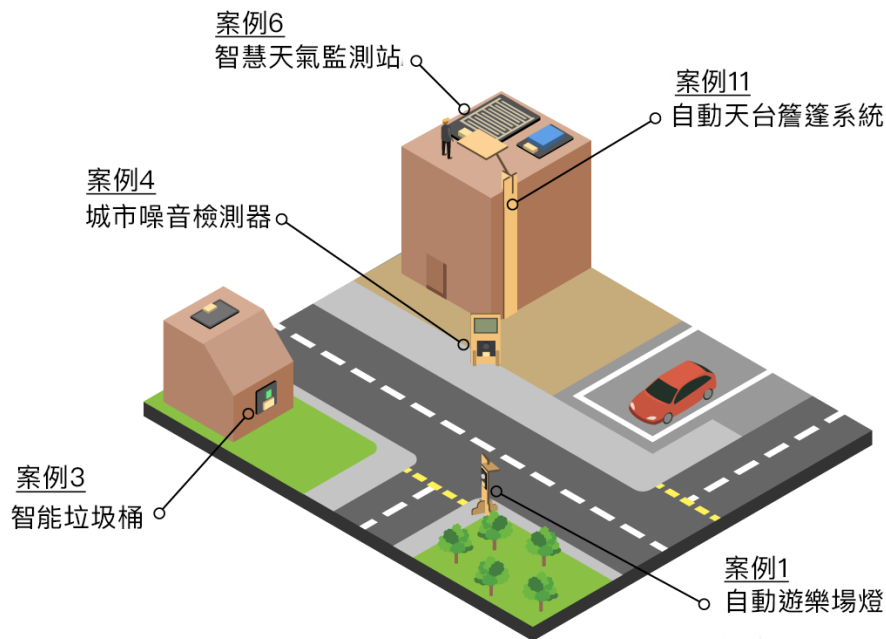
- ThingSpeak
- IFTTT



您可以设计自己的智慧城市，只需将不同的案例组合在一起或在您的城市中创建专用案例！

场境示例 2: 智慧环境城市

这是一个注重高可持续性的城市发展，其中包括增加绿化面积，提高能源效率，减少废物，监测气候信息和污染状况（例如噪音）。



部件:

- 温湿度传感(DHT11)
- 雨水传感器
- 180° 舵机
- 白色LED
- 运动传感器
- 声音传感器
- 超声波距离传感器
- 彩色LED(WS2812B)

IOT(物联网):

- ThingSpeak
- IFTTT

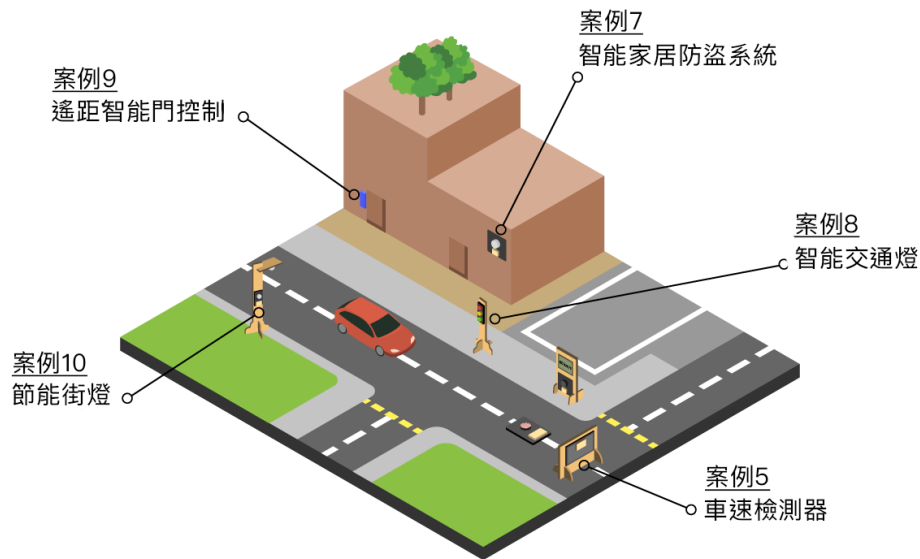


您可以设计自己的智慧城市，只需将不同的案例组合在一起或在您的城市中创建专用案例！

场境示例 3:

智能交通安全

交通安全是城市的首要任务，在公共交通系统中监视车速和交通状况，只有公司的负责人才能进入大楼，如果附近有可疑人员，警报将通过网络发送。



部件:

- 180° 舵机
- 白色LED
- 蜂鸣器
- 运动传感器
- 亮度传感器
- 交通灯模块
- 超音波距离传感器

IOT(物联网):

- App Inventor 2
- IFTTT
- WiFi 频道



您可以设计自己的智慧城市，只需将不同的案例组合在一起或在您的城市中创建专用案例！

附录 I: Smarthon电子模块 – 探索更多

	智能水泵		温度，湿度及压力传感器
	电动风扇		数位亮度传感器
	水温传感器	想知道更多信息，请前往 www.smarthon.cc	

**想知道更多信息
请前往**

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



SMARTHON

www.smarthon.cc

INGITE YOUR CREATIVITY