Smarthon Documentation

發F *1.0*

Smarthon Limited

2022 年 08 月 04 日

Contents

1 課程指南 1

CHAPTER 1

課程指南

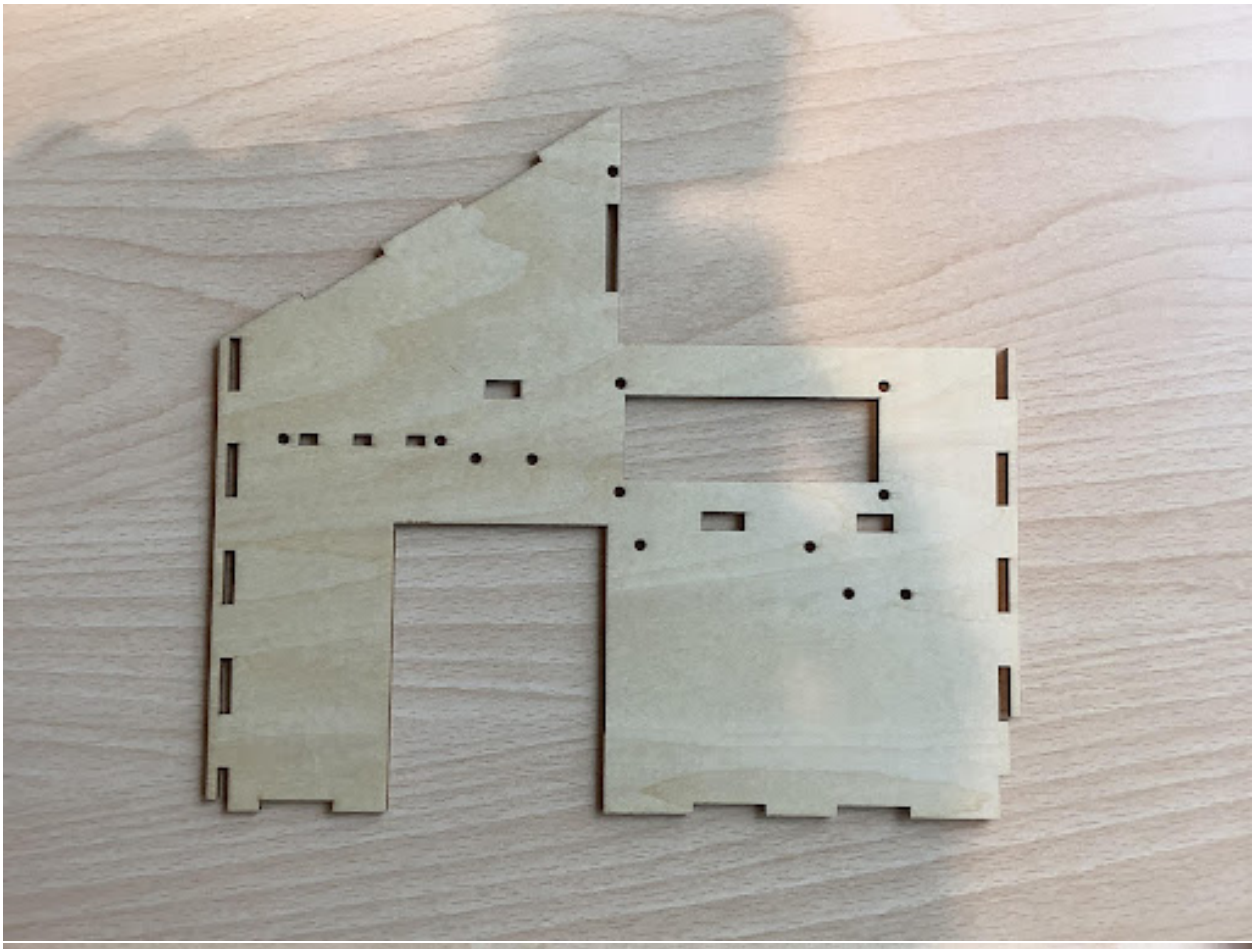
1.1 智能家居套裝課程 micro:bit

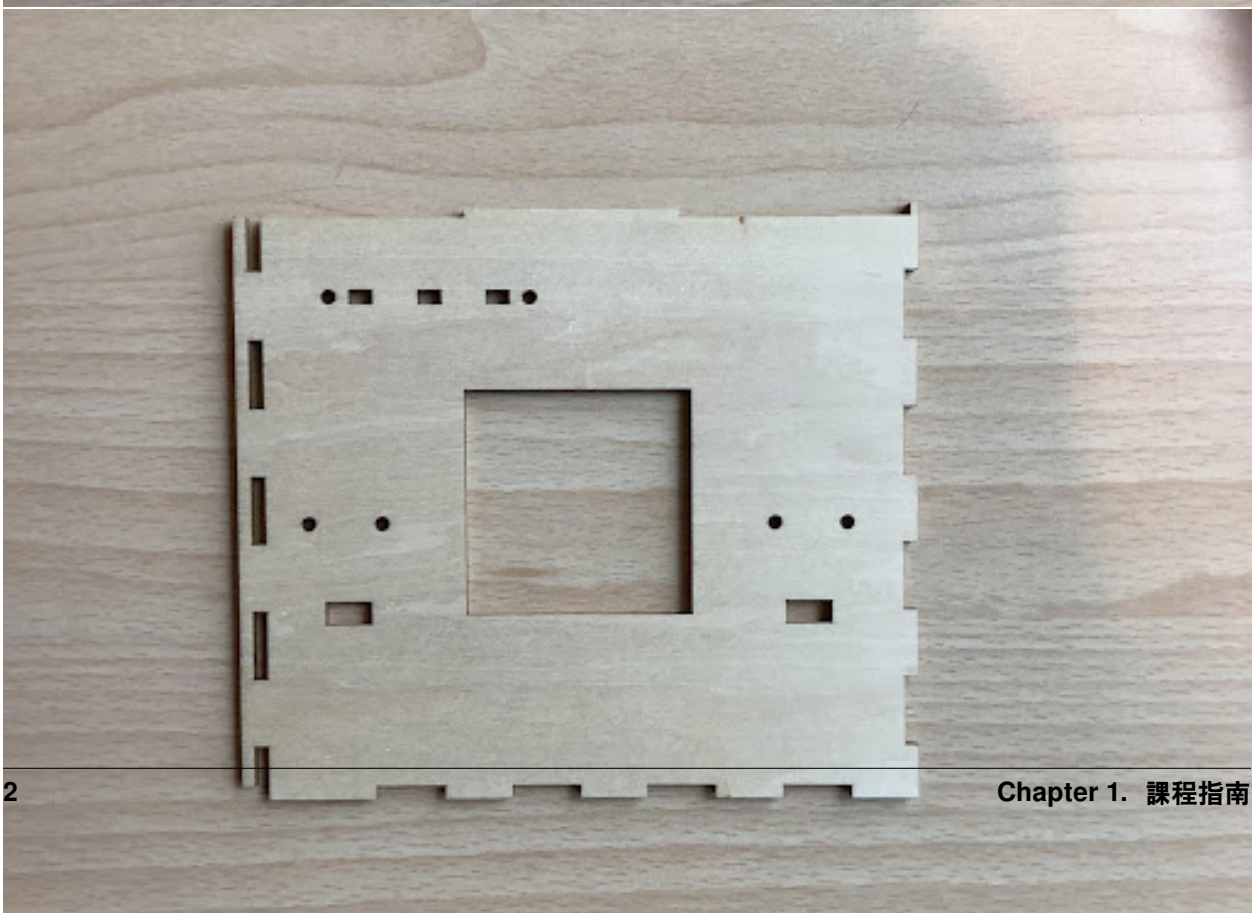
1.1.1 第一課認識智能家居

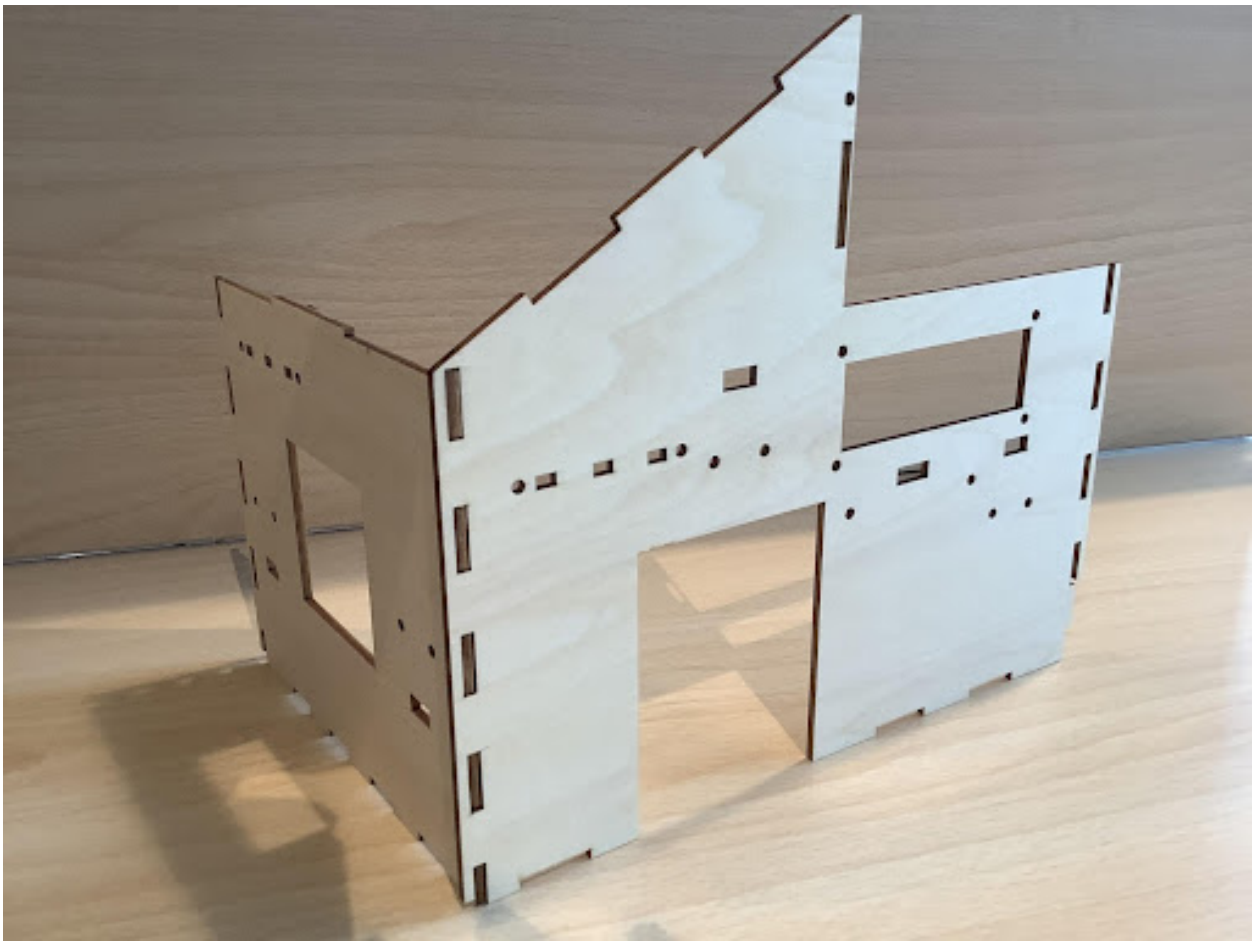
什Ⓕ是智能家居

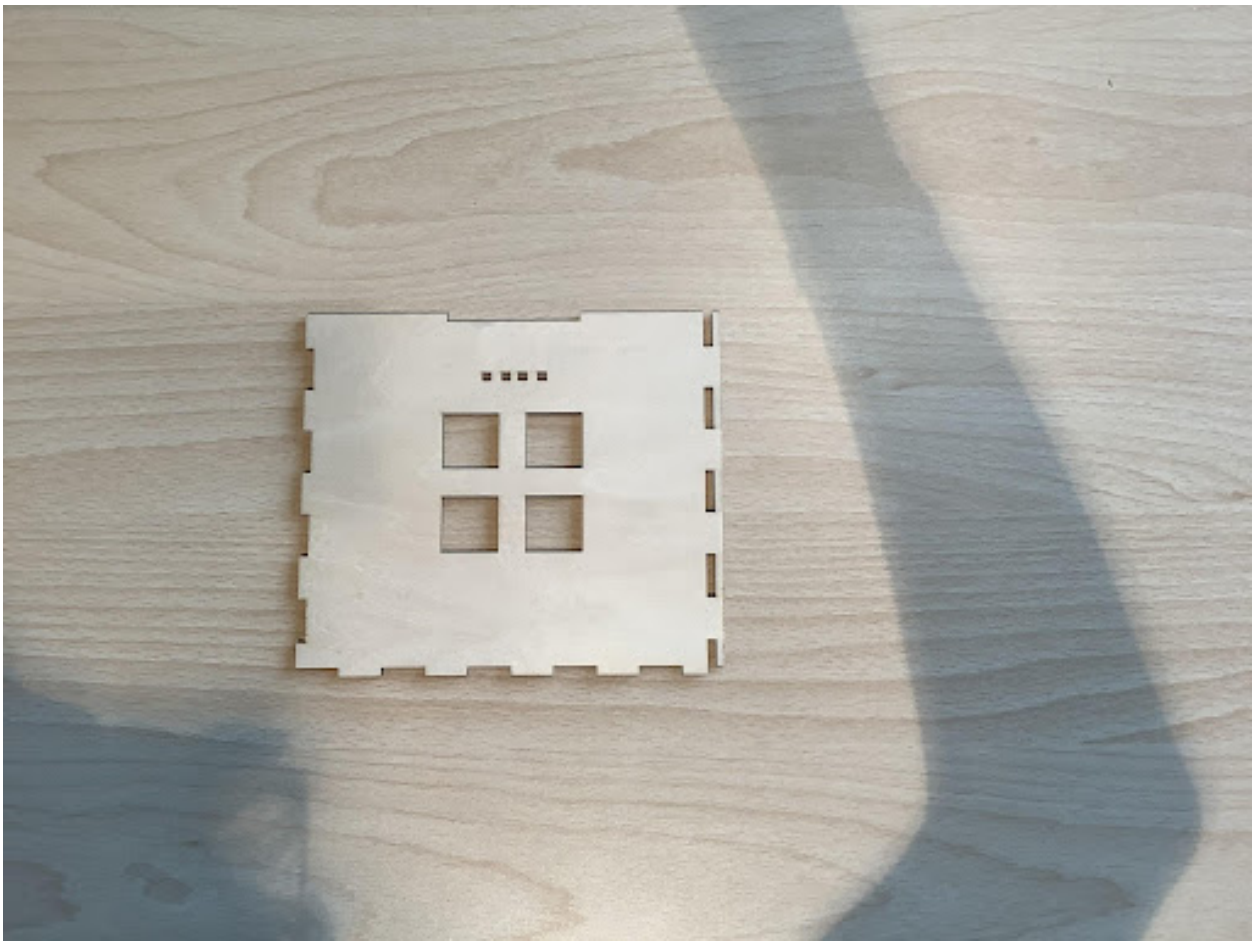
智能家居的優勢

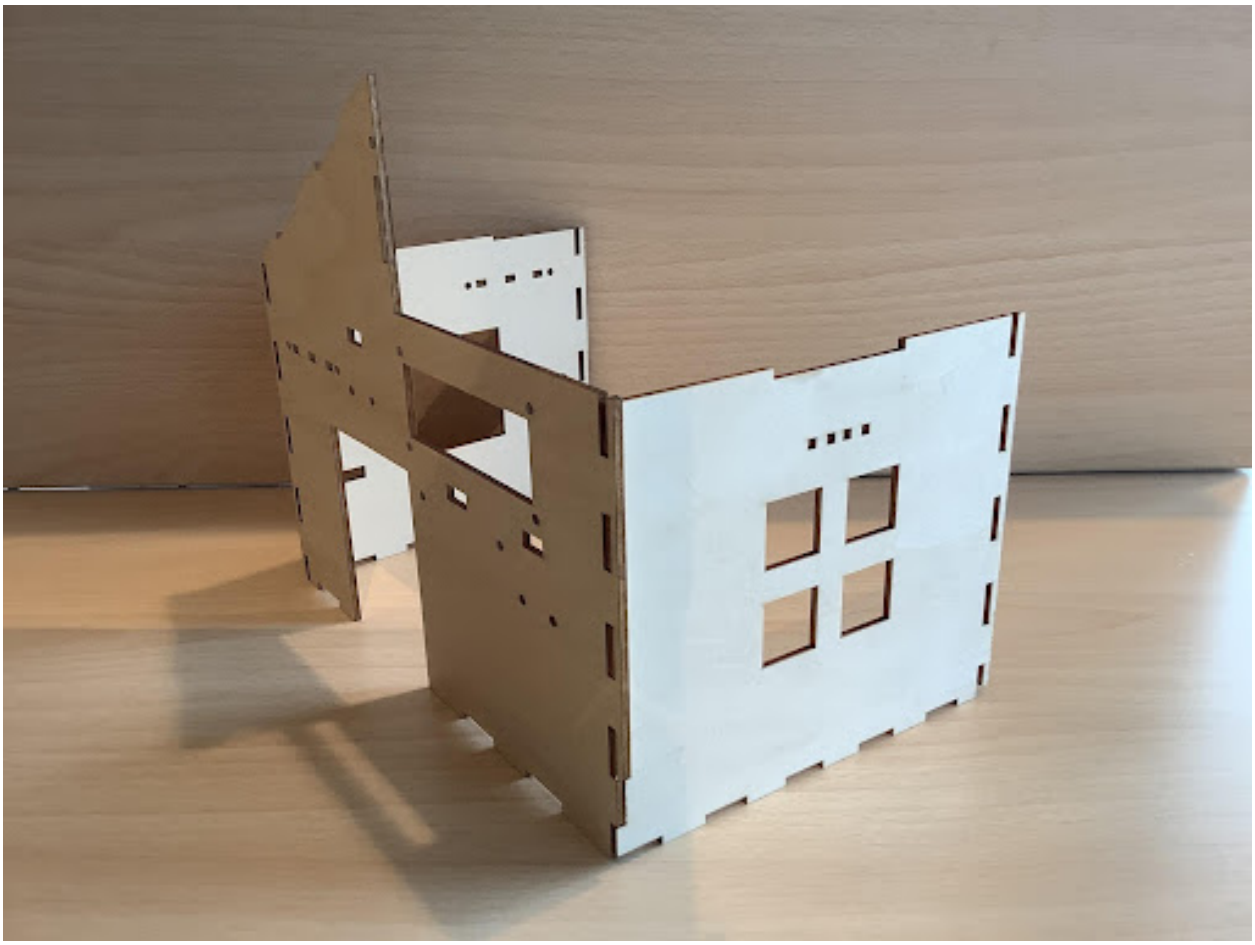
智能家居模型搭建步驟展示

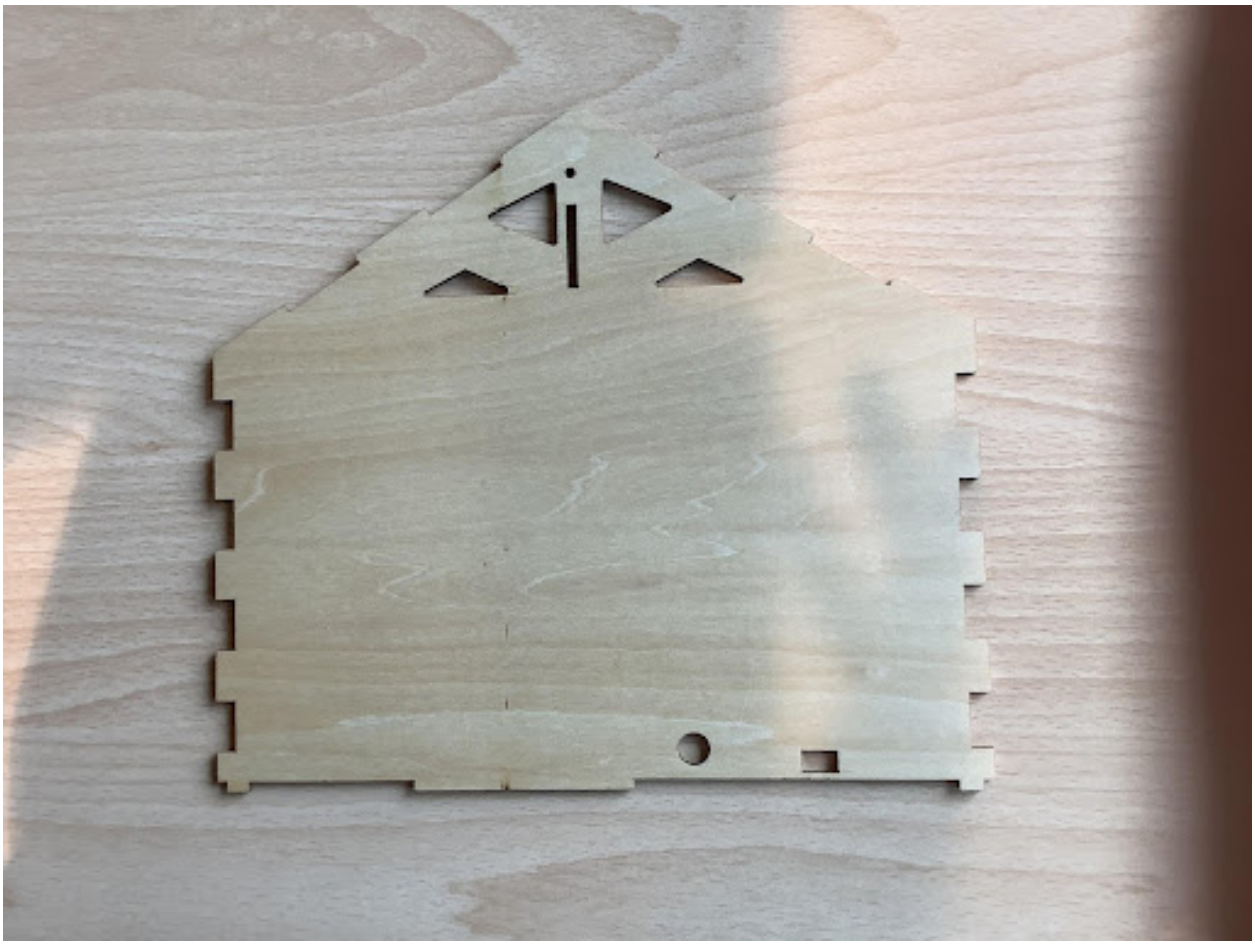


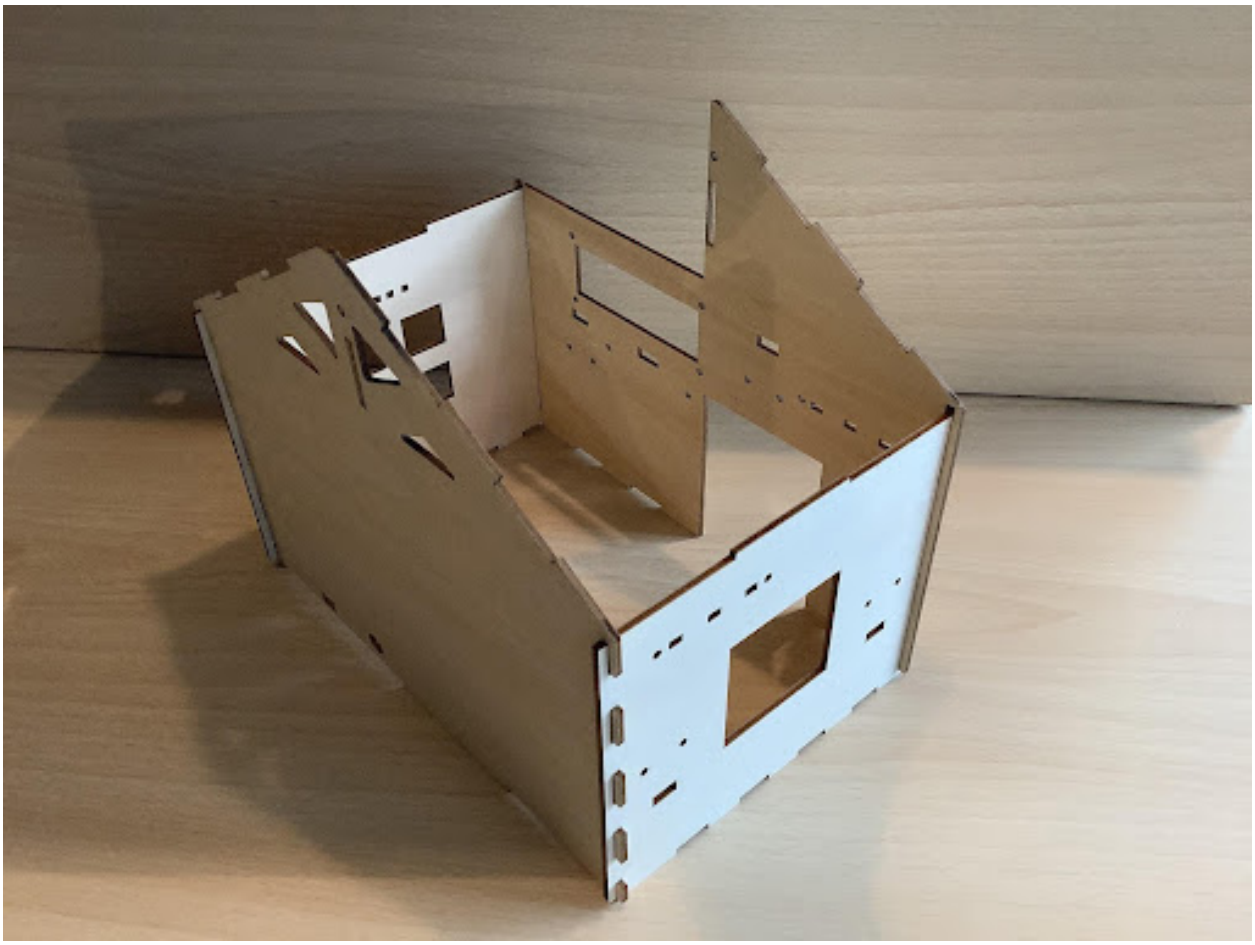


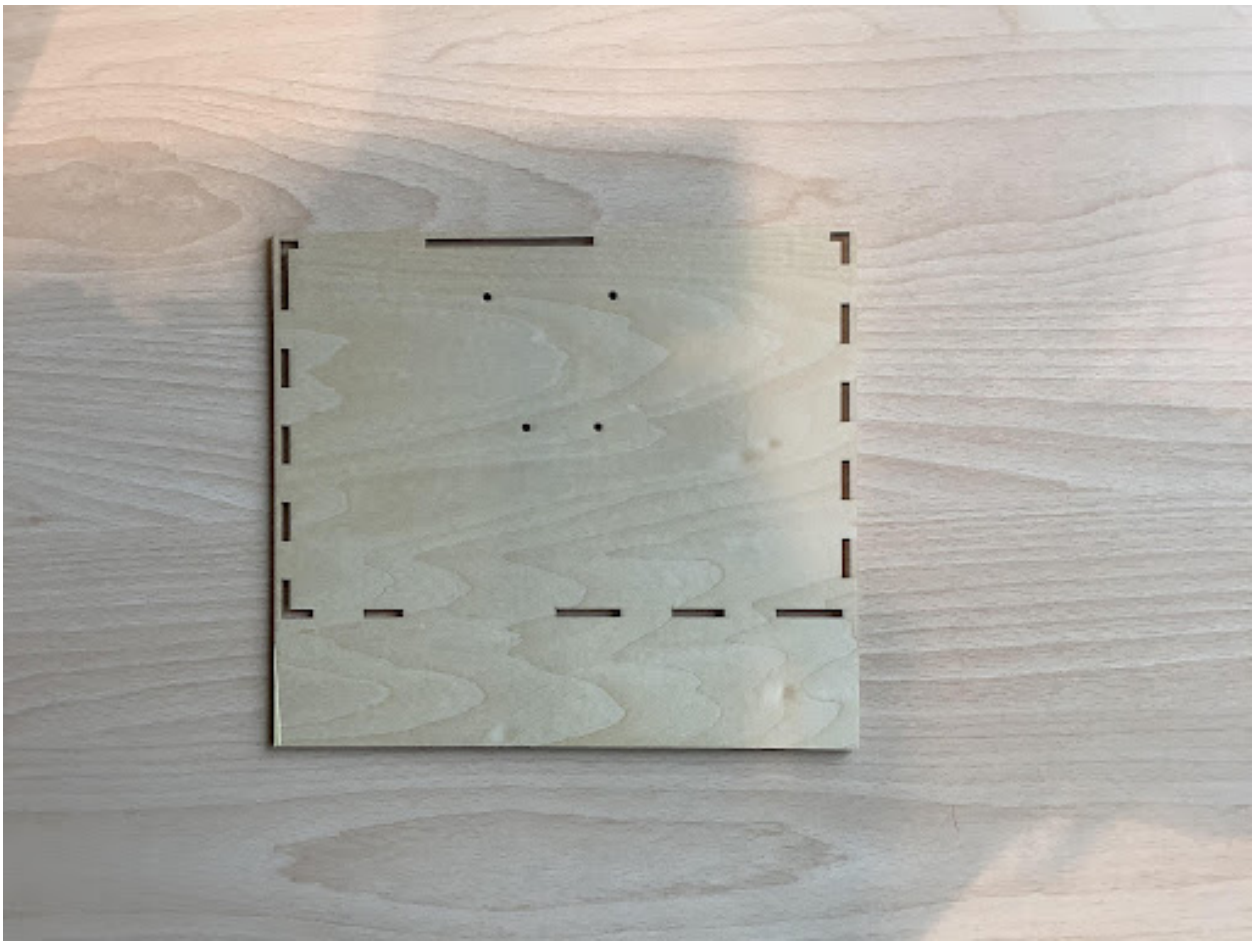


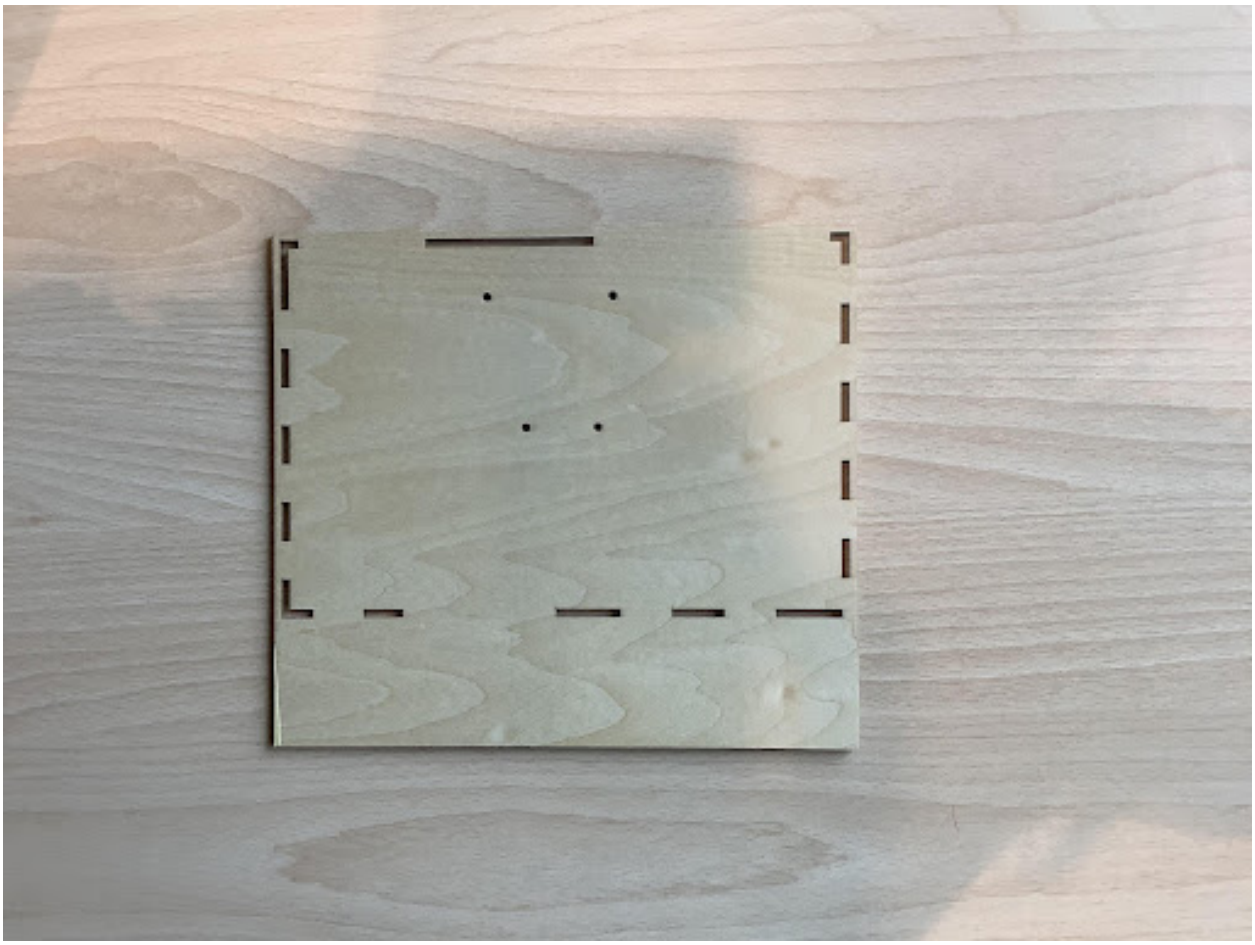


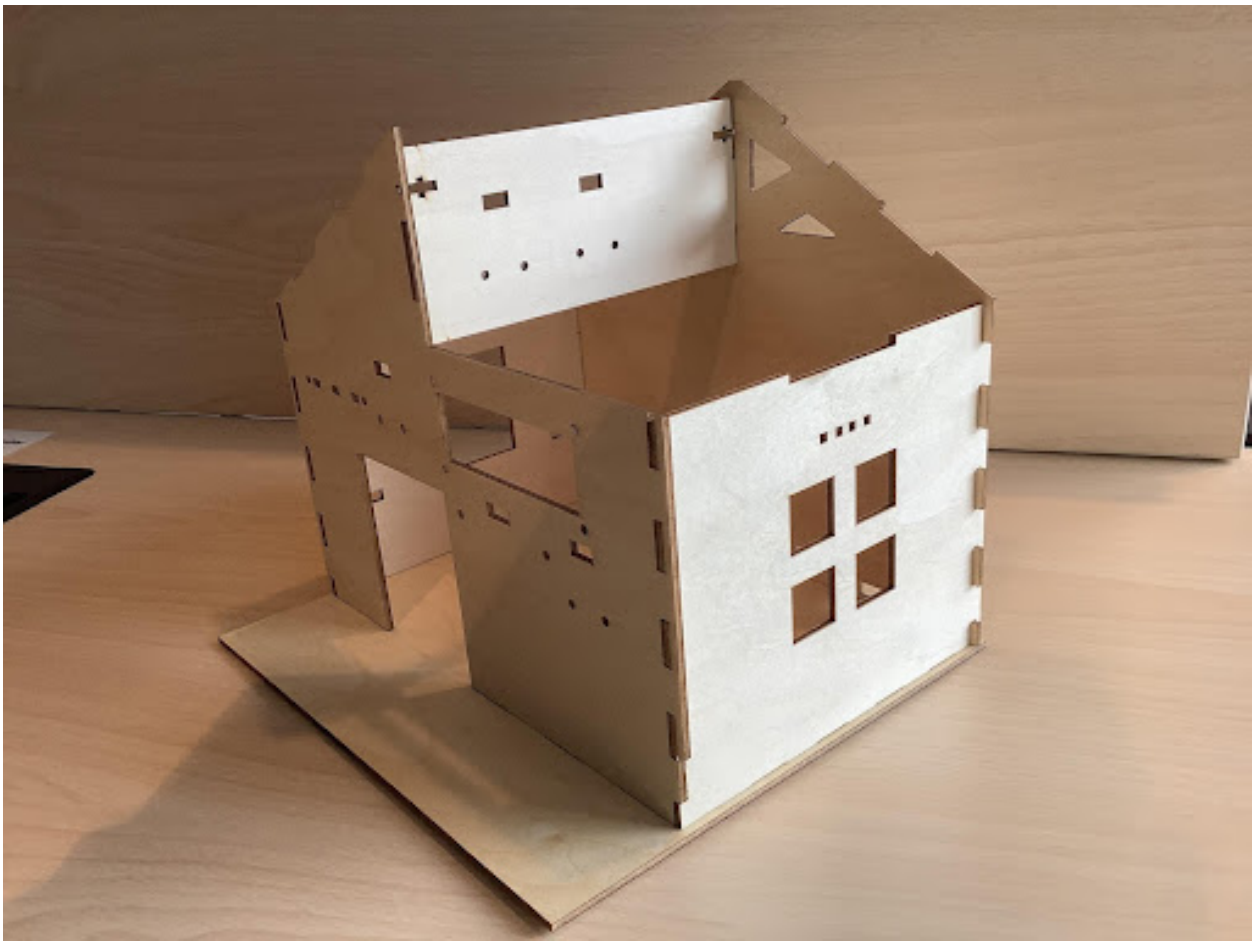


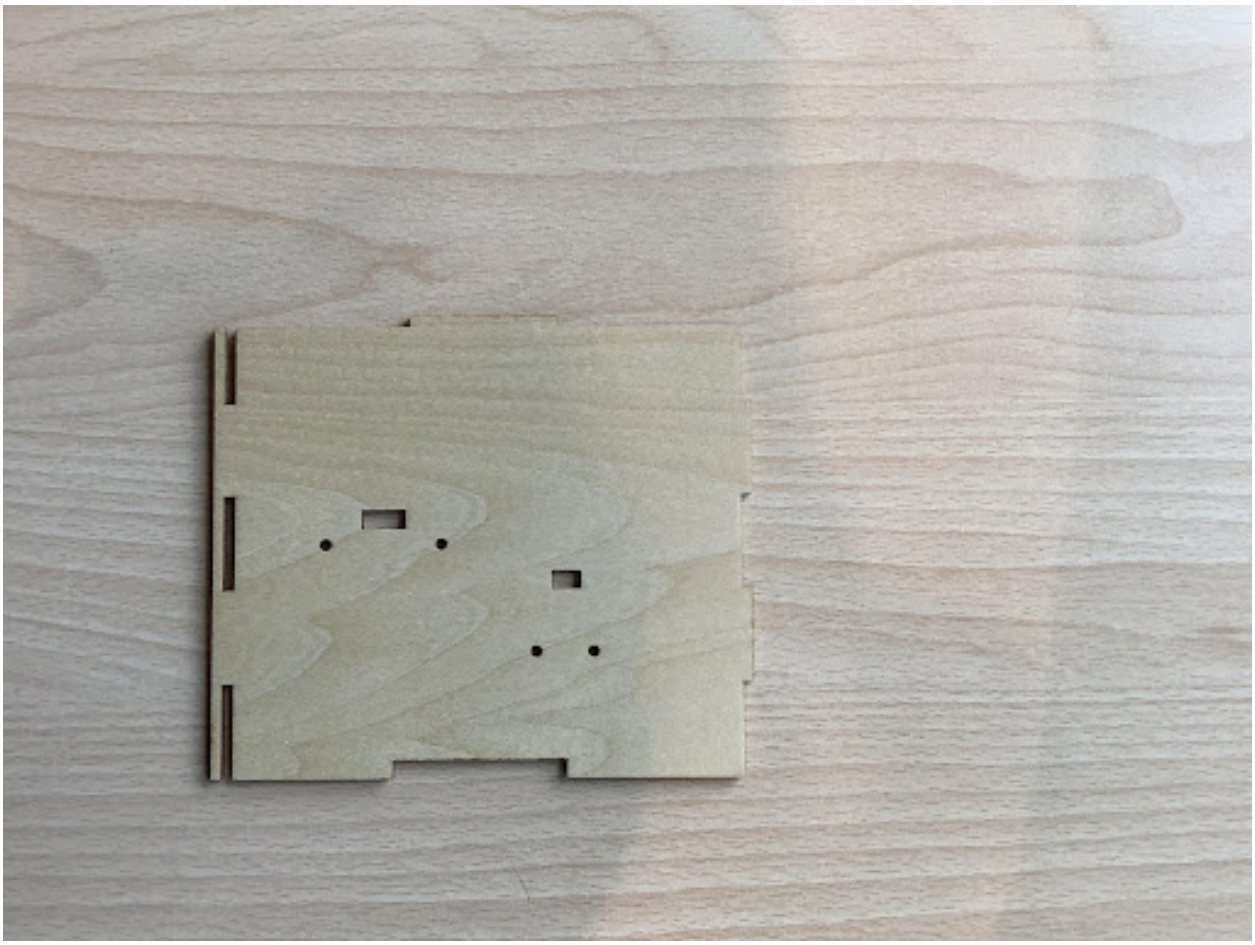




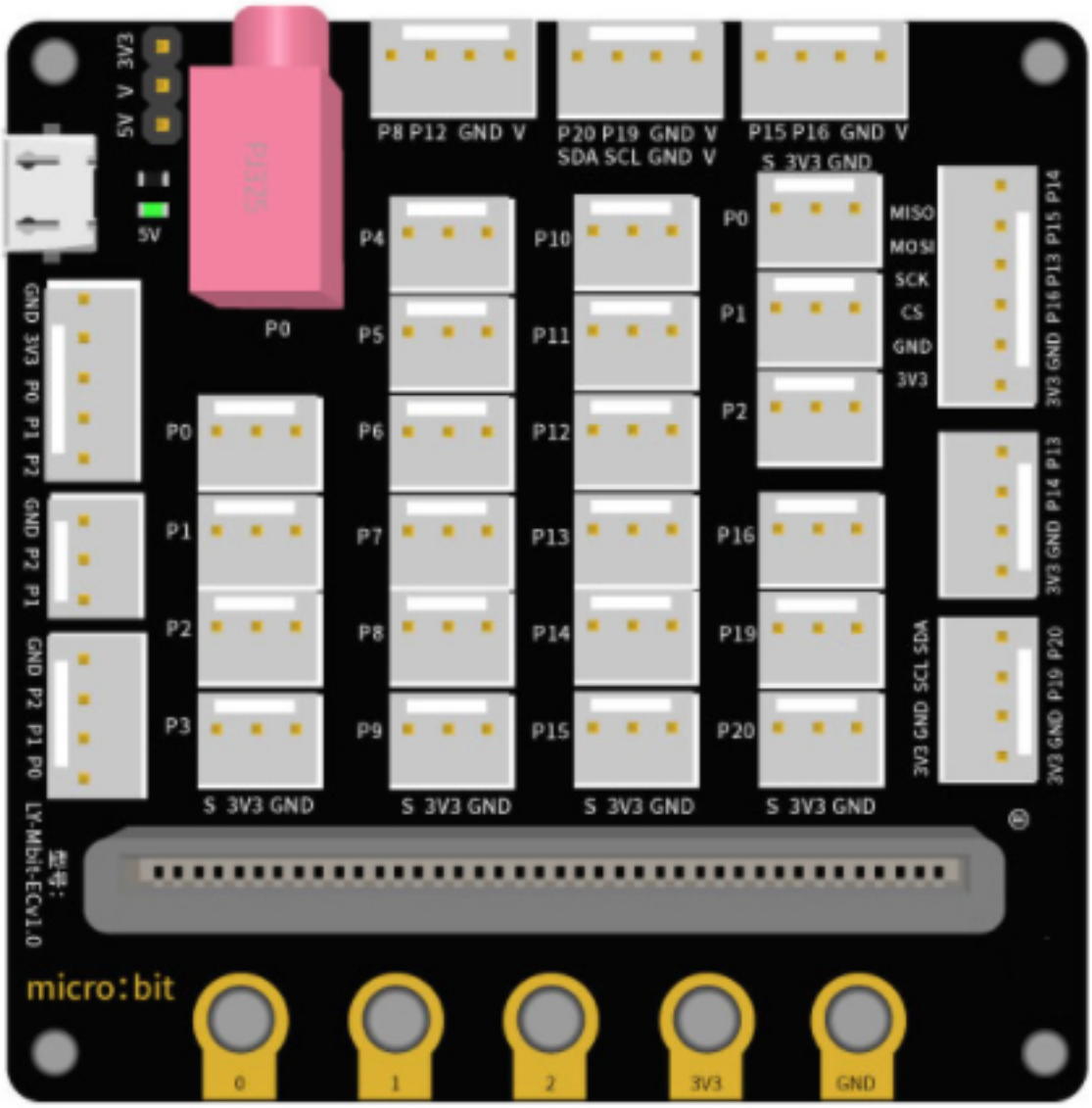




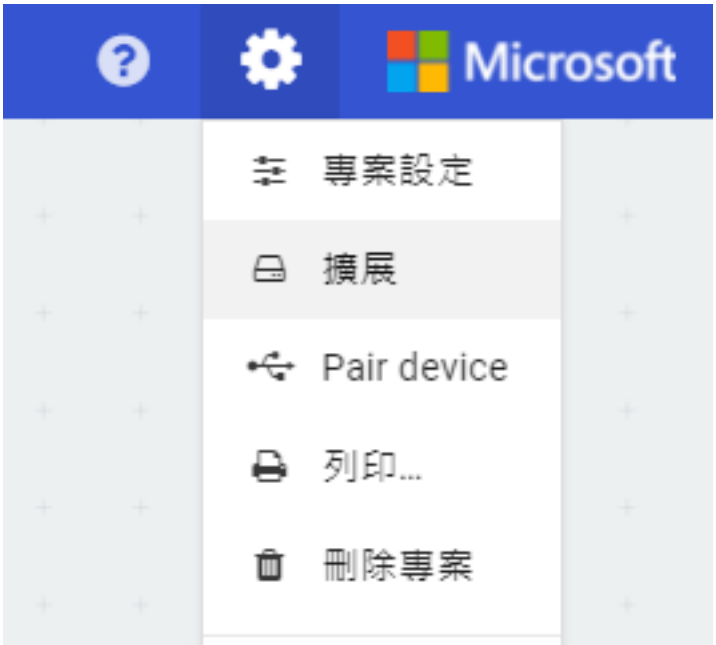




了解 micro:bit 拓展板



準備 Micro:bit 編程程式 Makecode：添加擴展


Microsoft
專案設定
擴展
Pair device
列印...
刪除專案

擴展
https://github.com/stemhub/pxt-StemhubCity
stemhubCity
用戶提供的擴展，非 Microsoft 維護。
進一步瞭解

1.1.2 第二課智能人體感應燈

任務背景

器材準備

了解智能人體感應燈

了解人體感應傳導器模塊和紅Ⓕ燈模塊

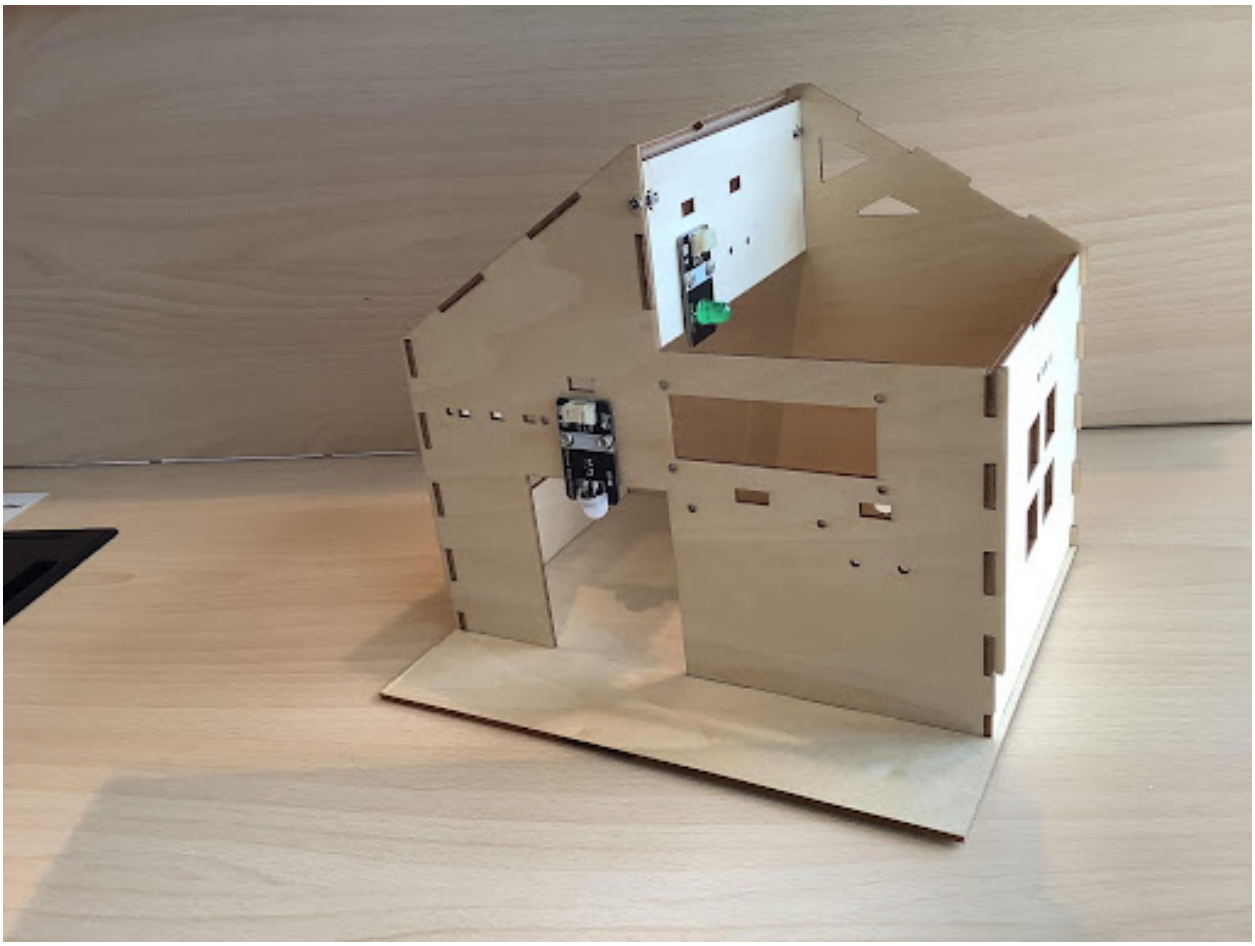
人體感應傳感器模塊

全自動感應: 人進入其感應範圍則輸出高電平，人離開感應範圍則自動延時關閉高電平，輸出低電平。

可重覆觸發方式：即感應輸出高電平後，在延時時間段Ⓕ，如果有人體在其感應範圍活動，其輸出將一直保持高電平，直到人離開後才延時將高電平變Ⓕ低電平（感應模塊檢測到人體的每一次活動後會自動順延一個延時時間段，Ⓕ且以最後一次活動的時間Ⓕ延時時間的起始點 HC-SR505 小型人體感應模塊有三個管Ⓕ，G Ⓕ GND 接地，V Ⓕ VCC 接高電平或 5v，S Ⓕ信號管Ⓕ，可以接 D1-D13 數字接口。

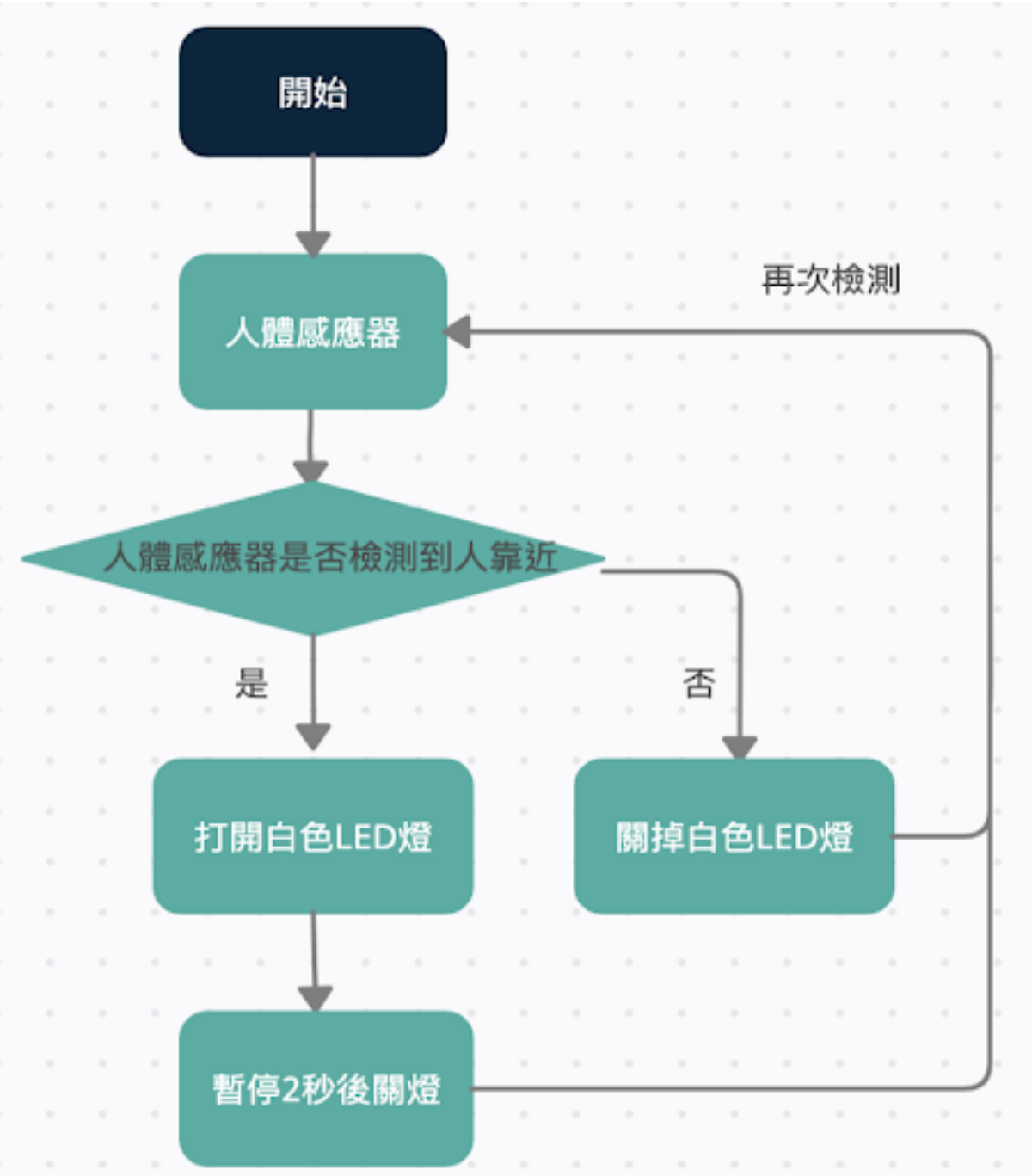
白色 LED 燈模塊

人體感應燈安裝

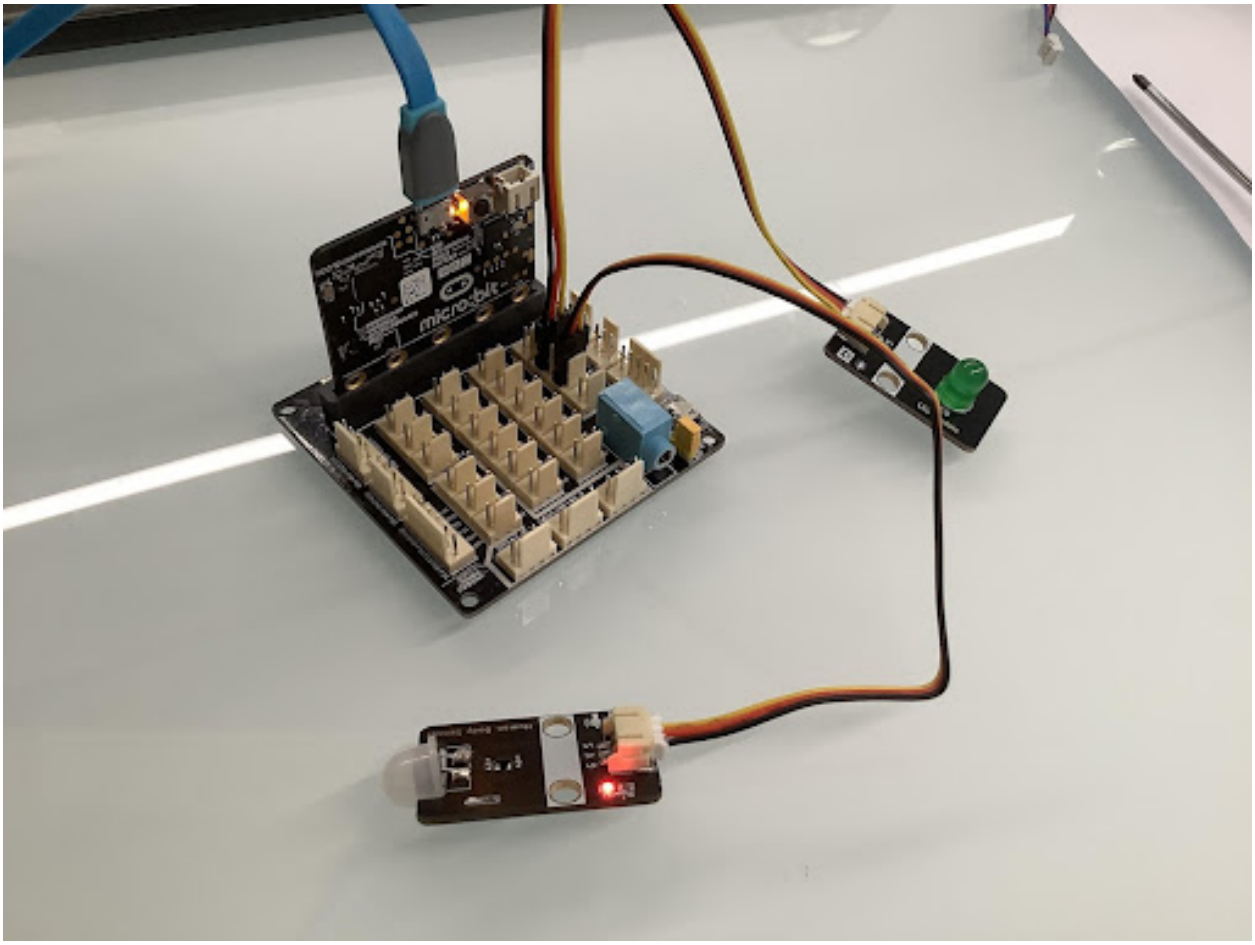


程序設計

算法設計



硬件連接



程序示例

Makecode 編程

https://makecode.microbit.org/_RkmbfDhPPHto

總結

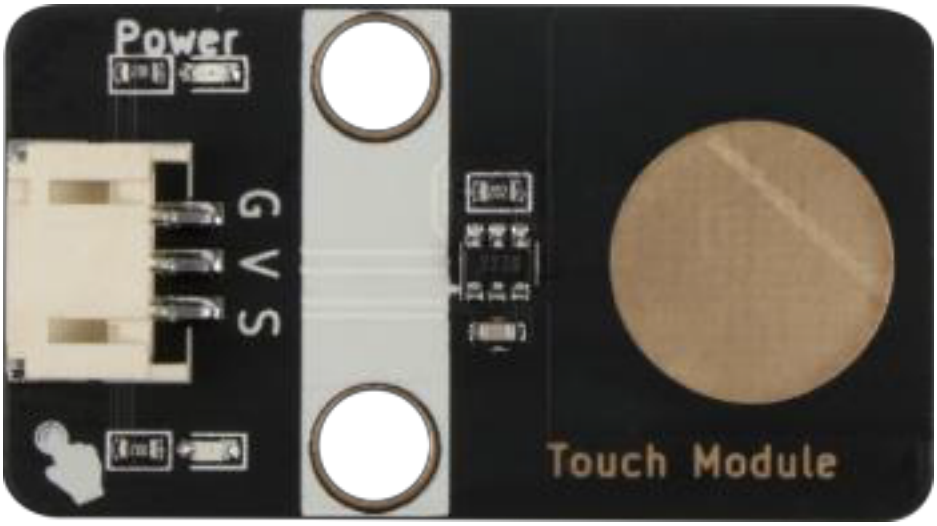
1.1.3 第三課音樂門鈴

任務背景:

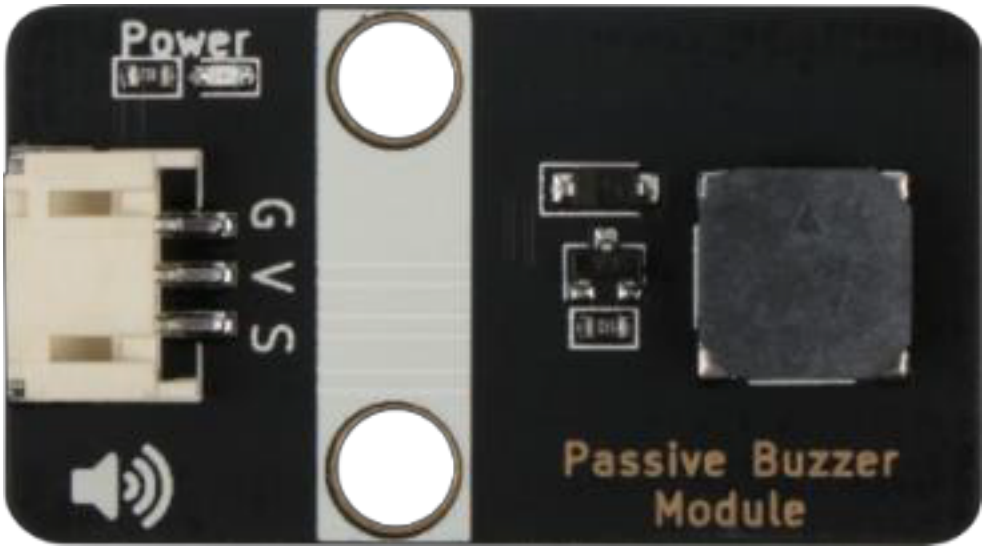
器材準備:

了解觸摸傳感器和無源蜂鳴器

觸摸傳感器

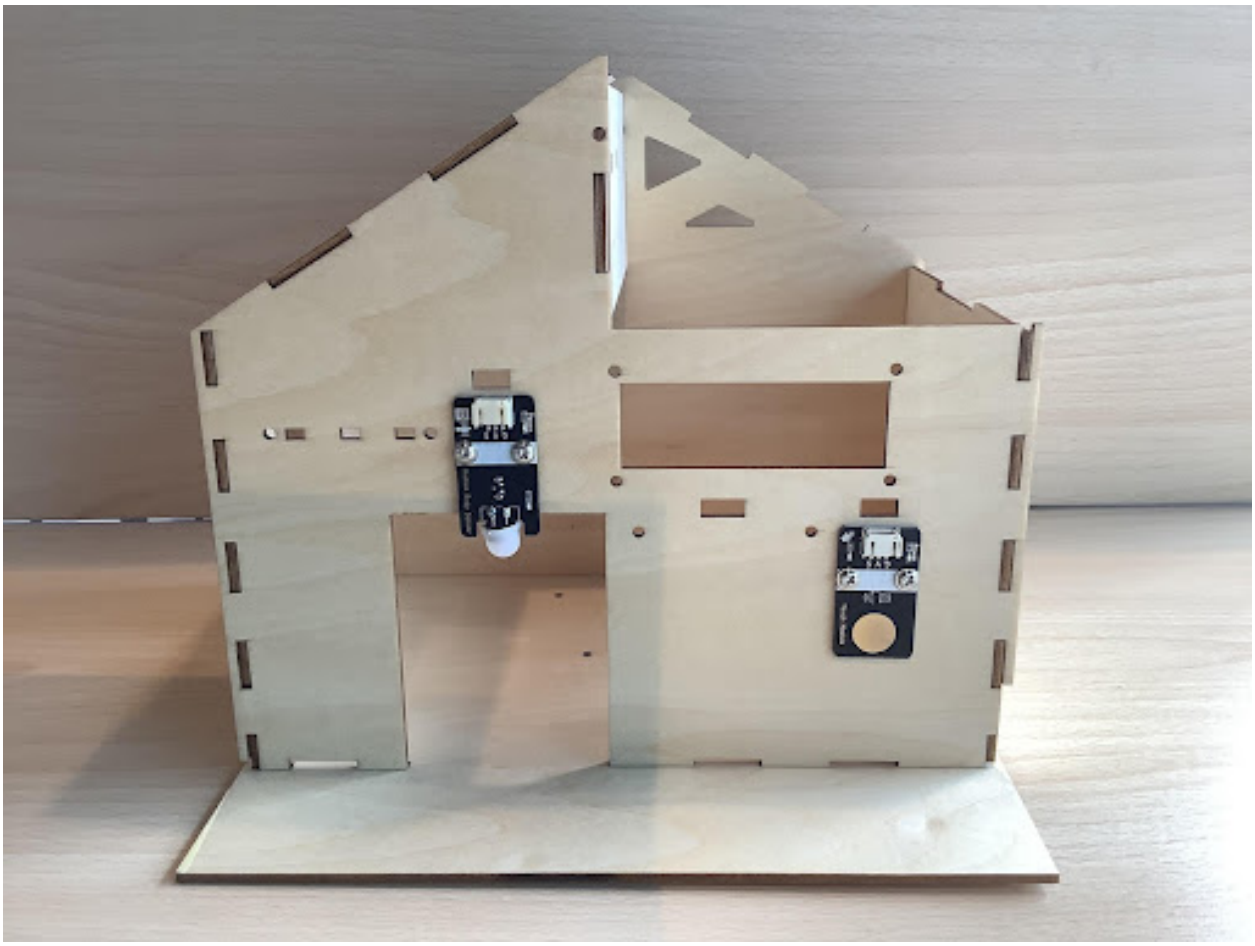


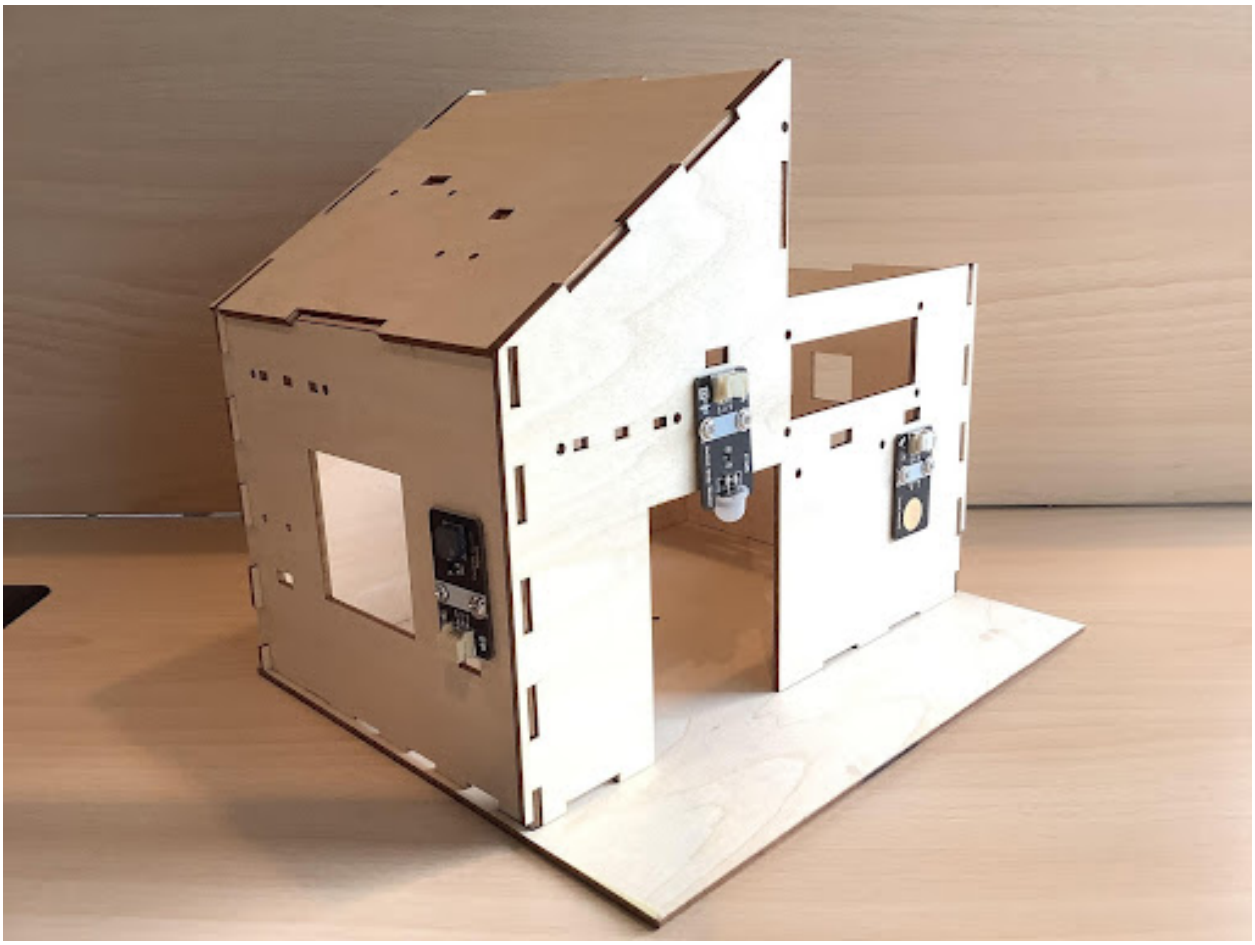
無源蜂鳴器模塊



蜂鳴器	BLE-UNO 主板
G	GND
V	VCC、5V、3.3V
S	D0-D13

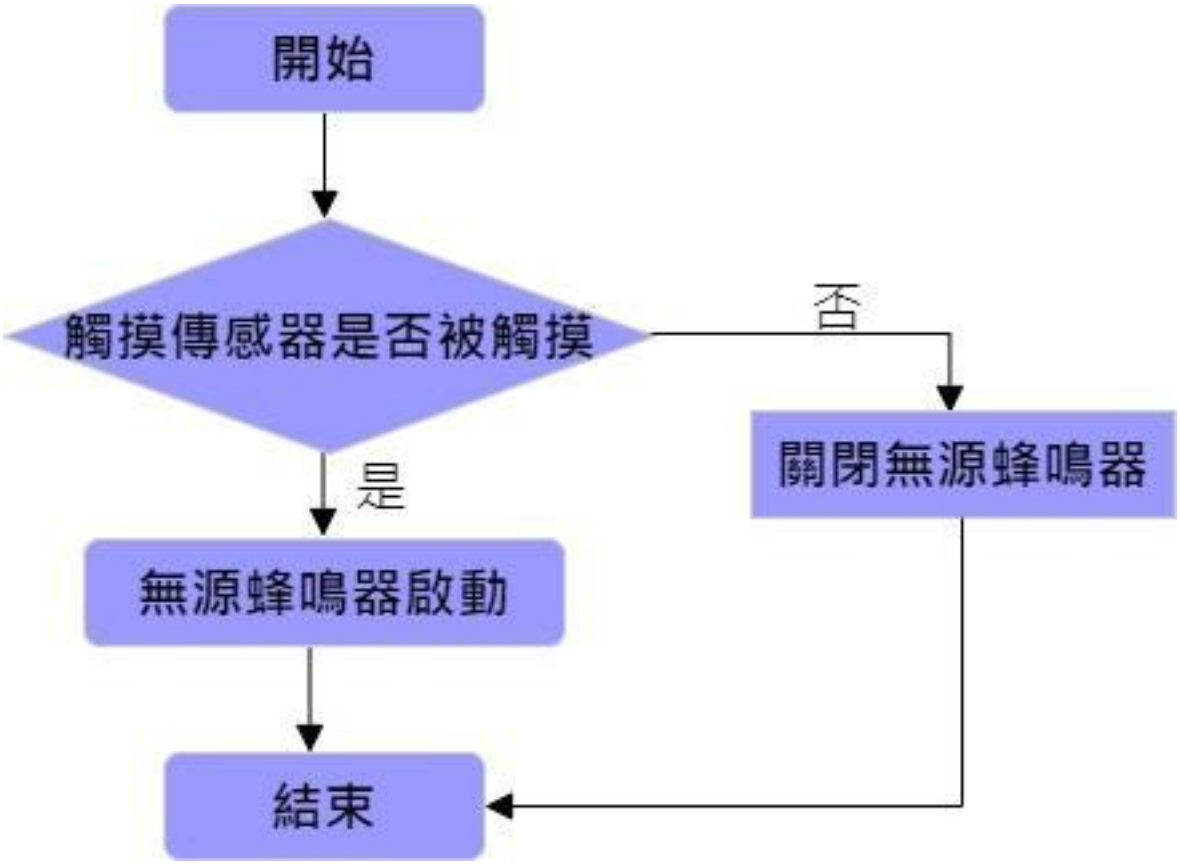
門鈴安裝



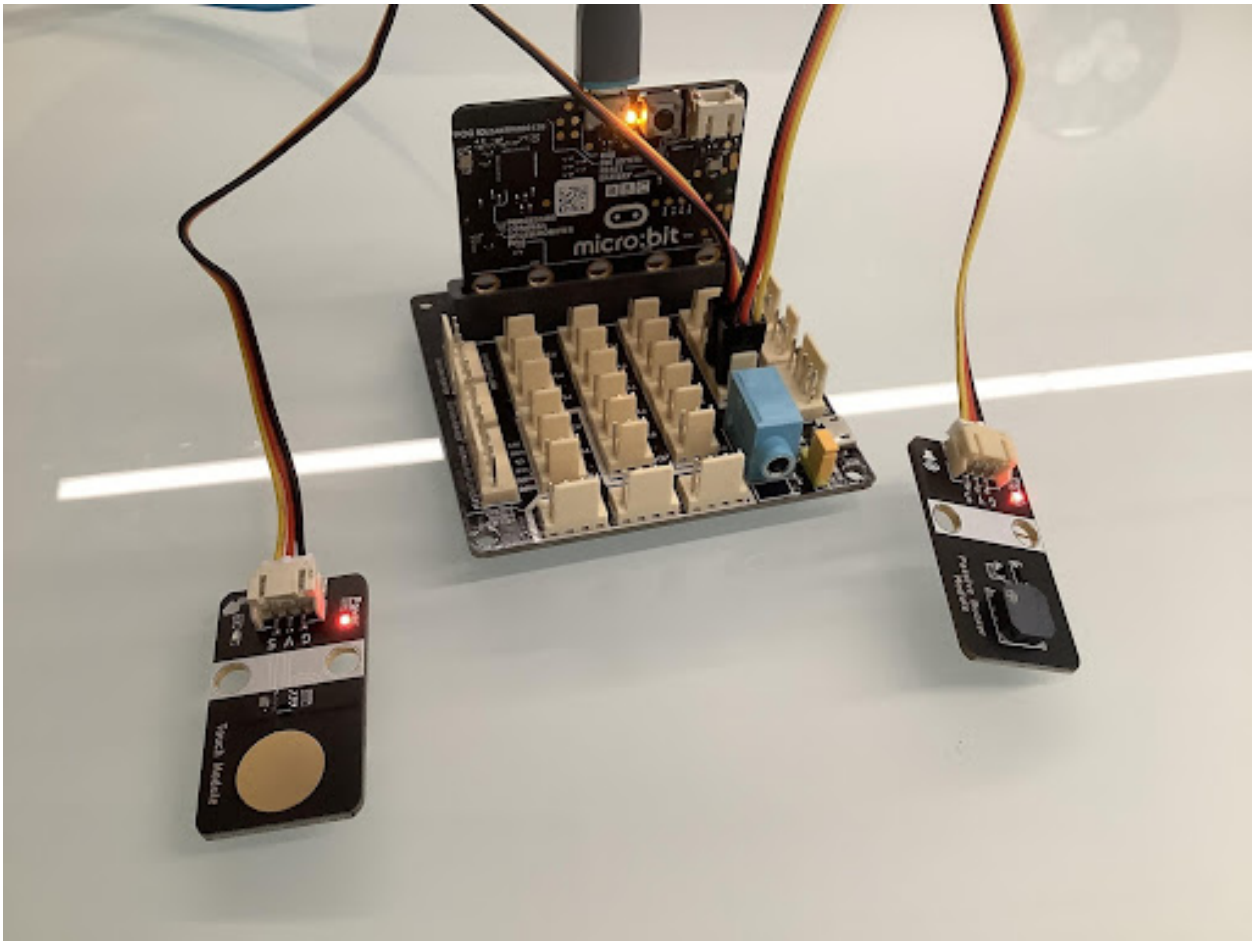


程序設計

算法設計



硬件連接:



程序示例:

總結

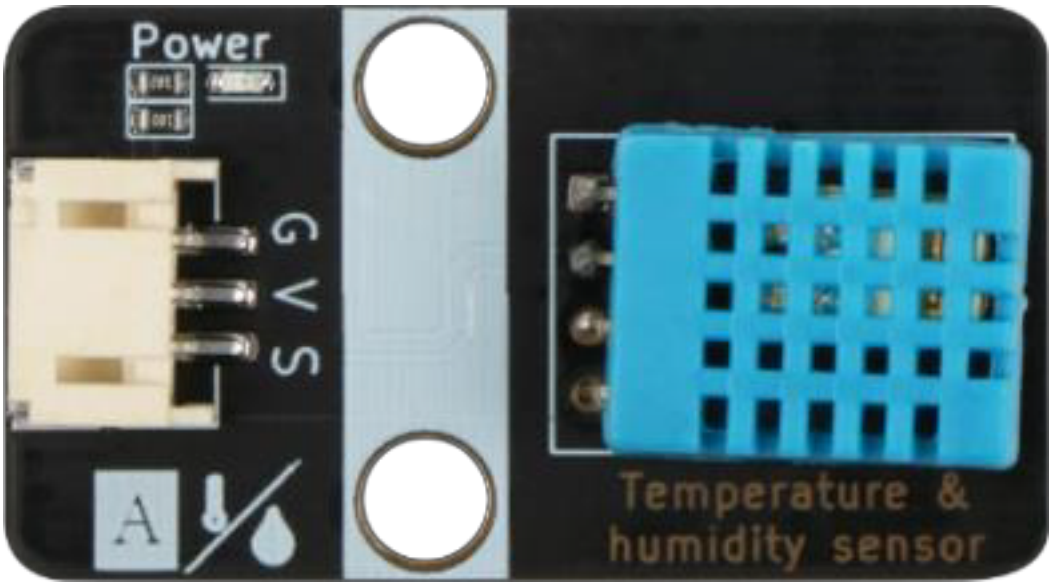
1.1.4 第四課智能Ⓕ控風扇

任務背景:

器材準備:

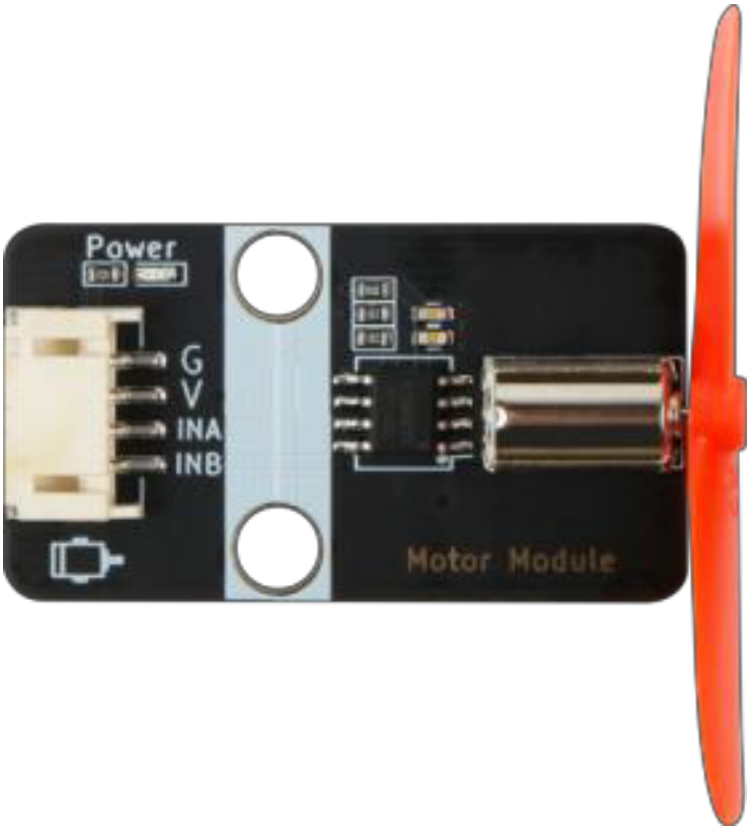
了解Ⓕ濕度傳感器和直流電動機風扇模塊

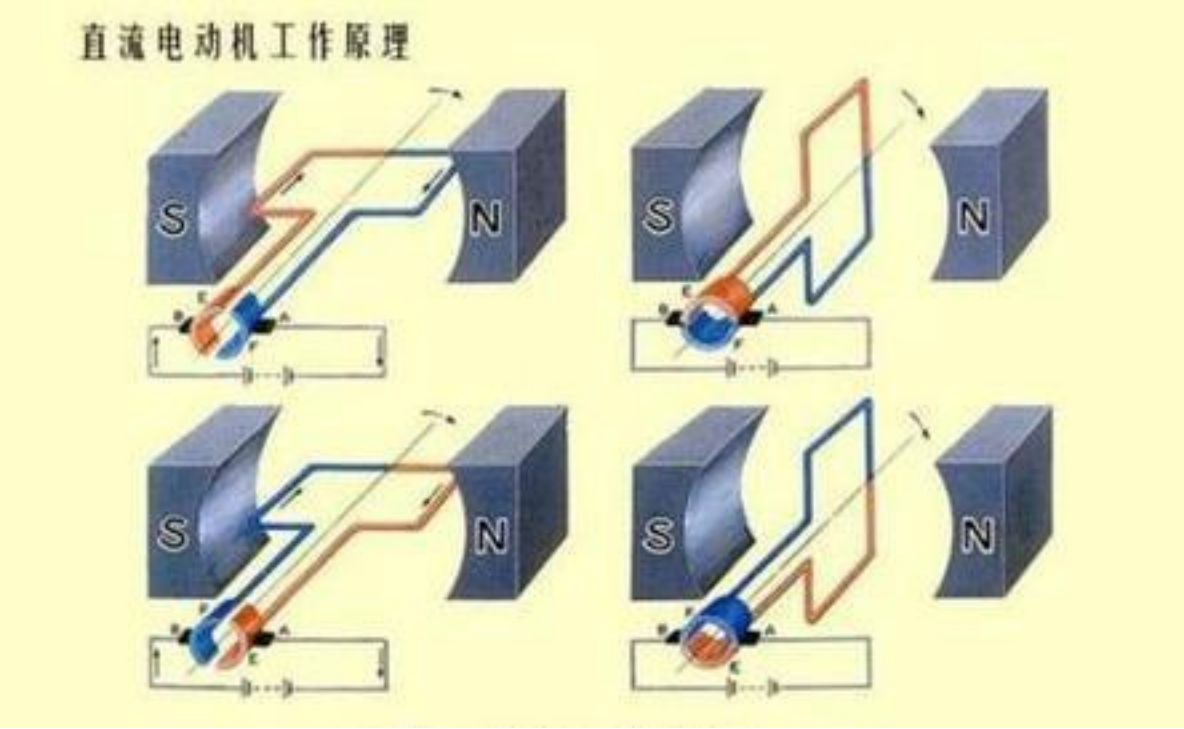
Ⓕ濕度傳感器模塊



溫濕度傳感器	Arduino BLE-UNO
G	GND
V	VCC、5V
S	A0-A5

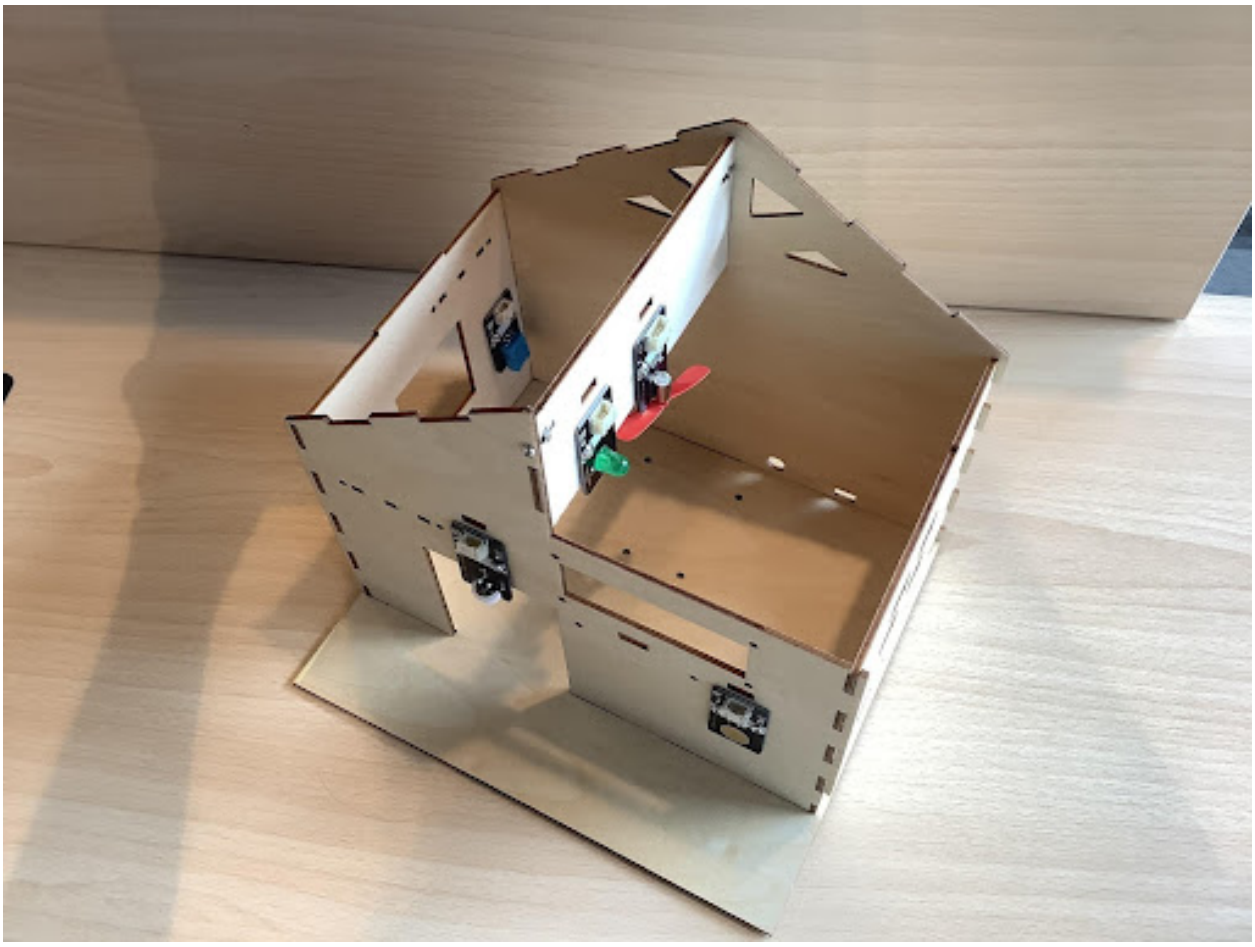
直流電動機風扇模塊





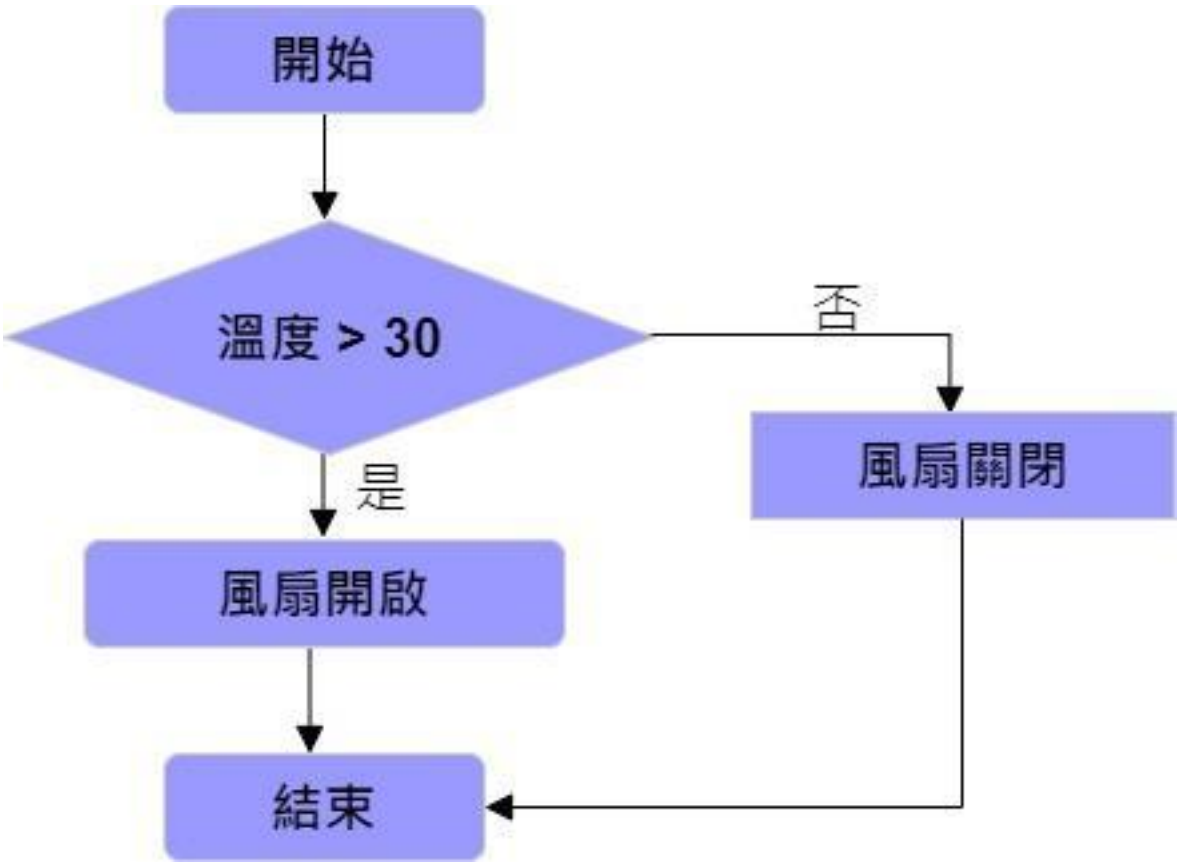
INA	INB	電機狀態
0	0	釋放
1	0	正轉
0	1	反轉
1	1	停止(剎車)

Ⓕ控風扇安裝

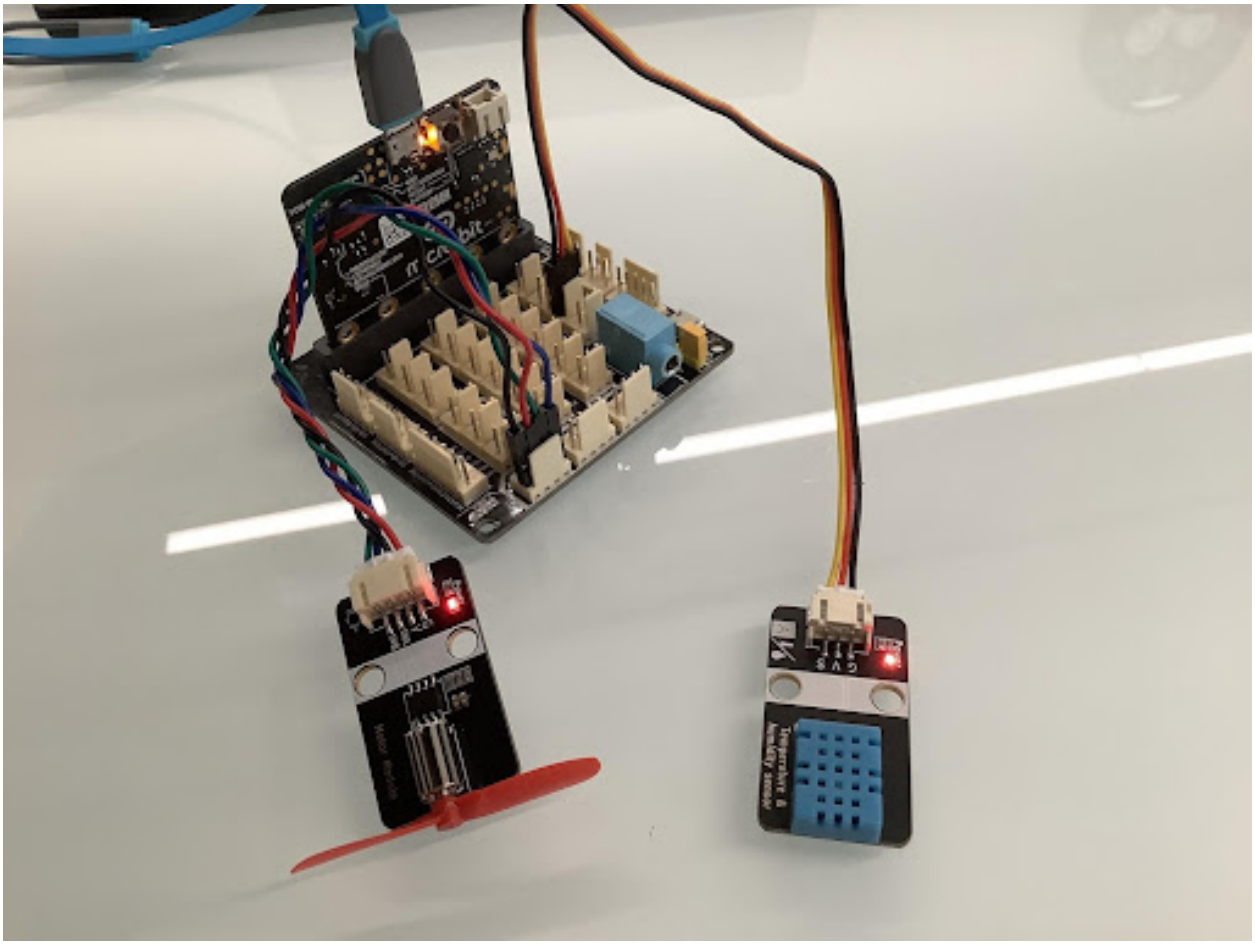


程序設計

算法設計:



硬件連接:



程序示例:

總結

1.1.5 第五課智能門禁

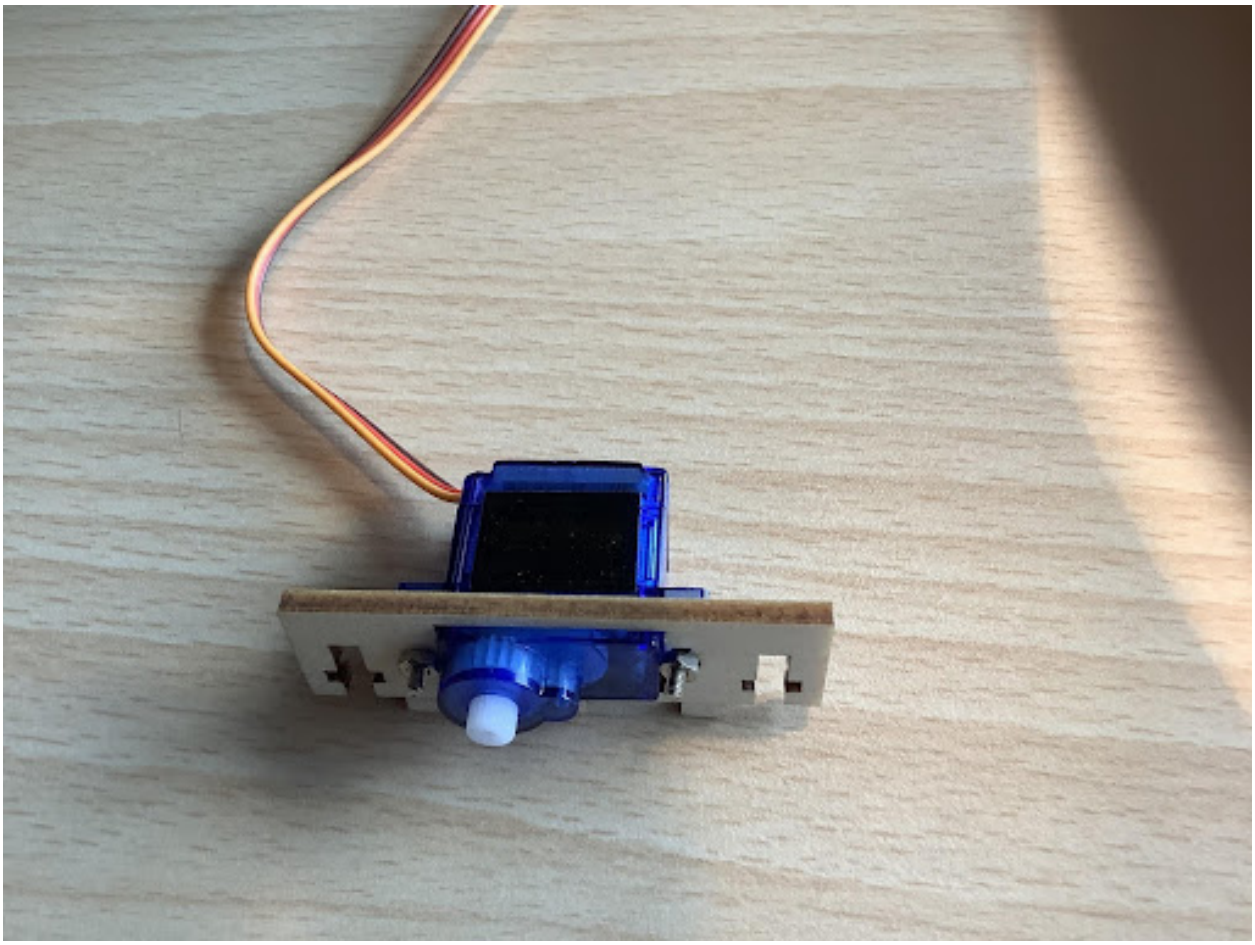
任務背景

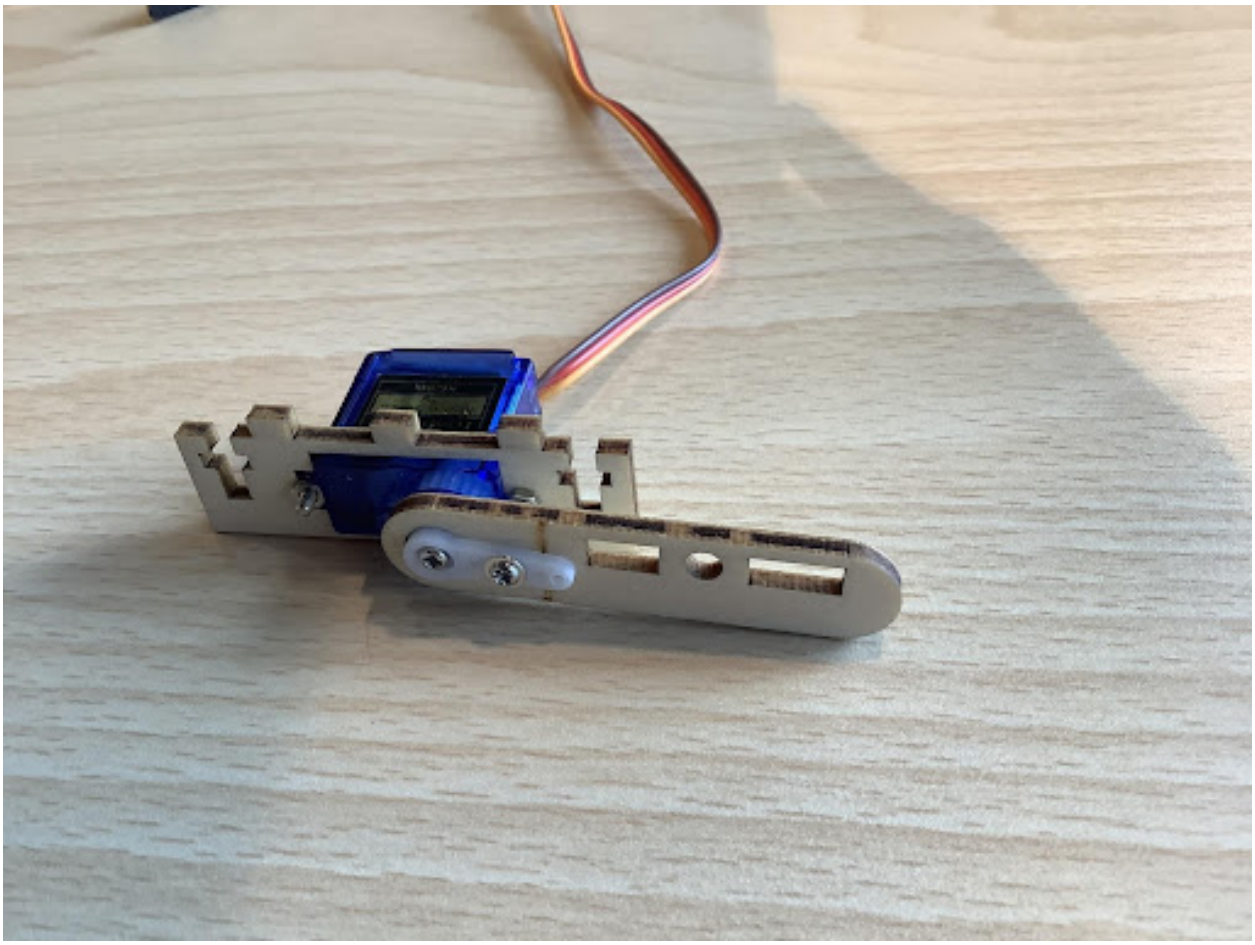
器材準備

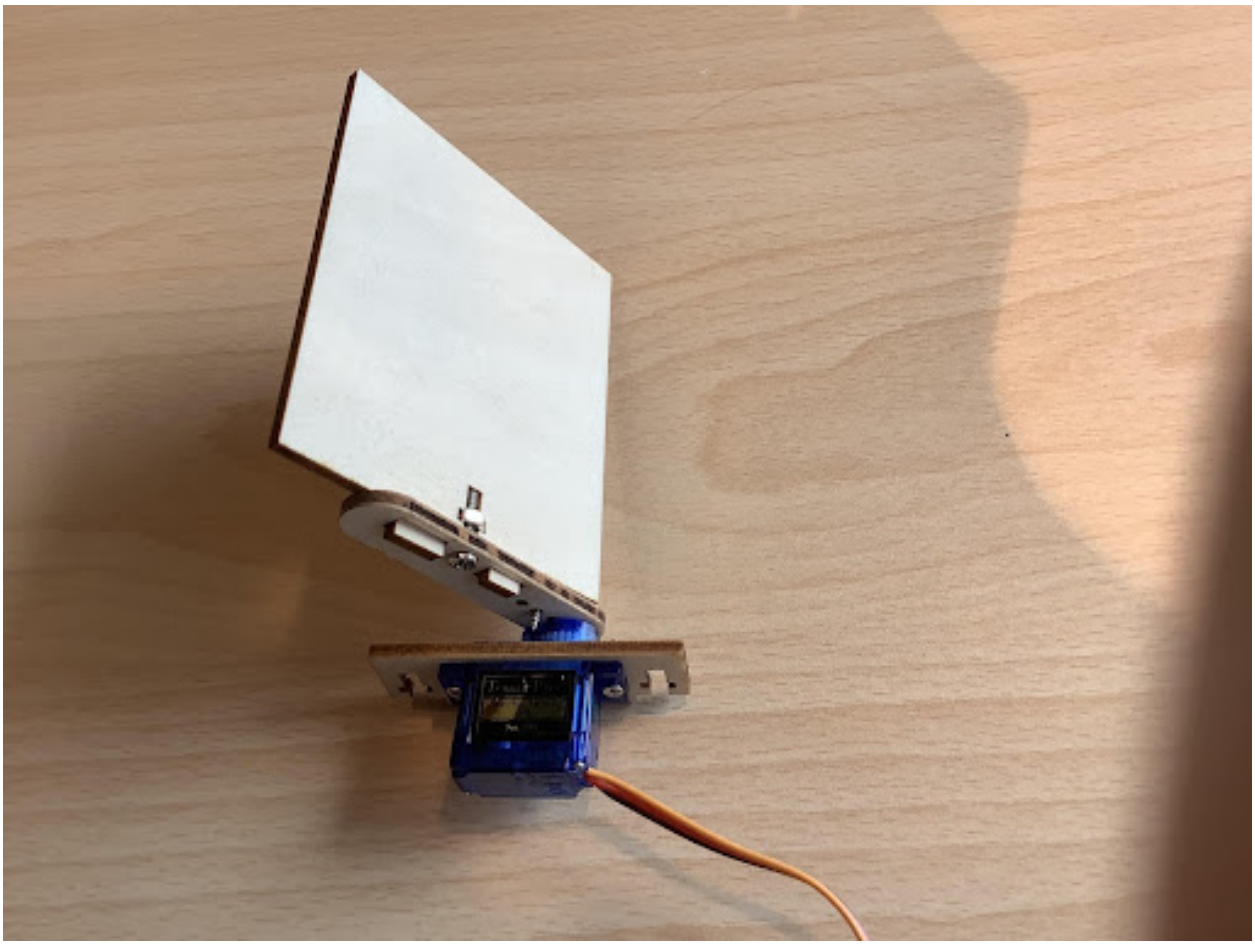
了解智能門鎖

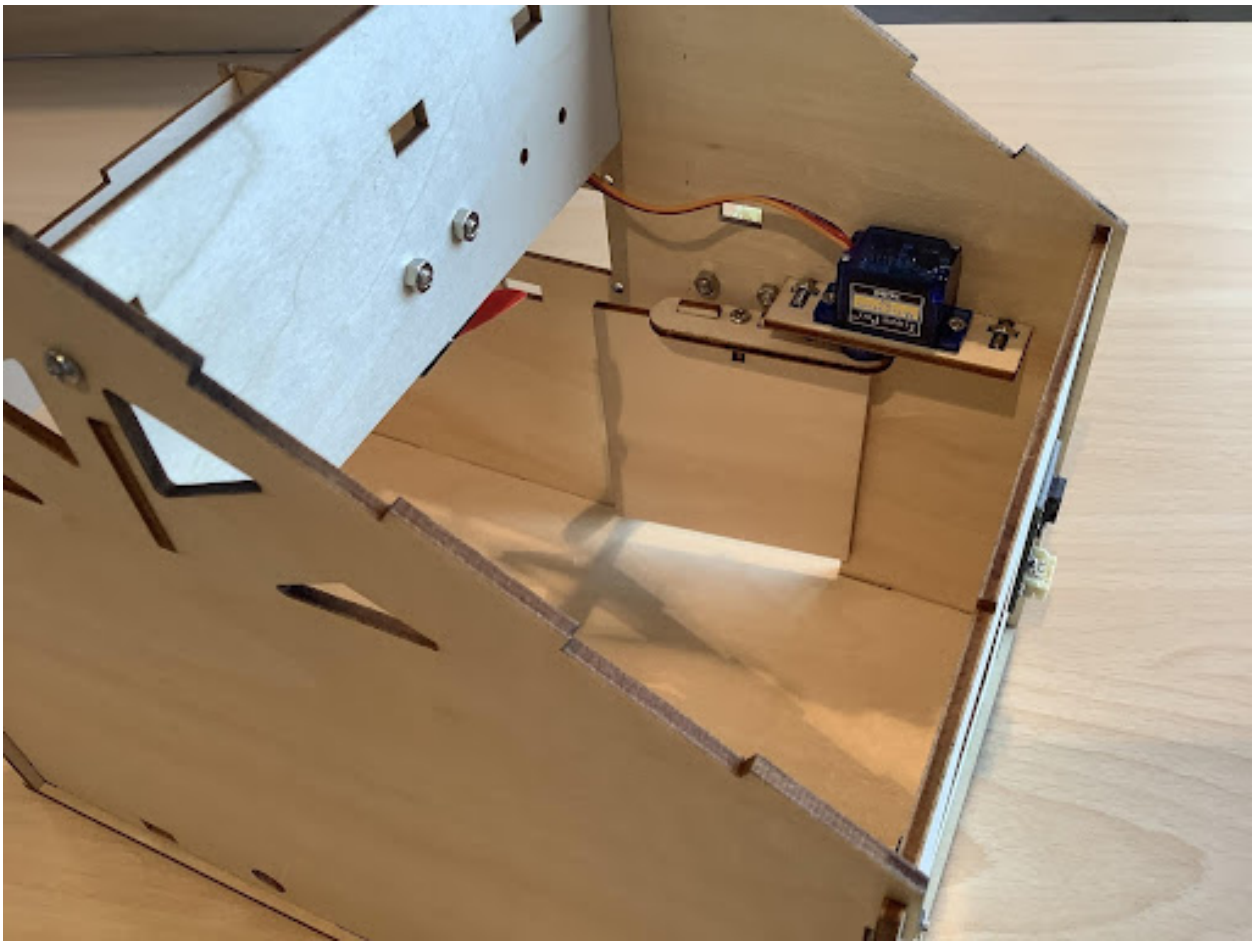
了解矩陣鍵盤傳感器模塊

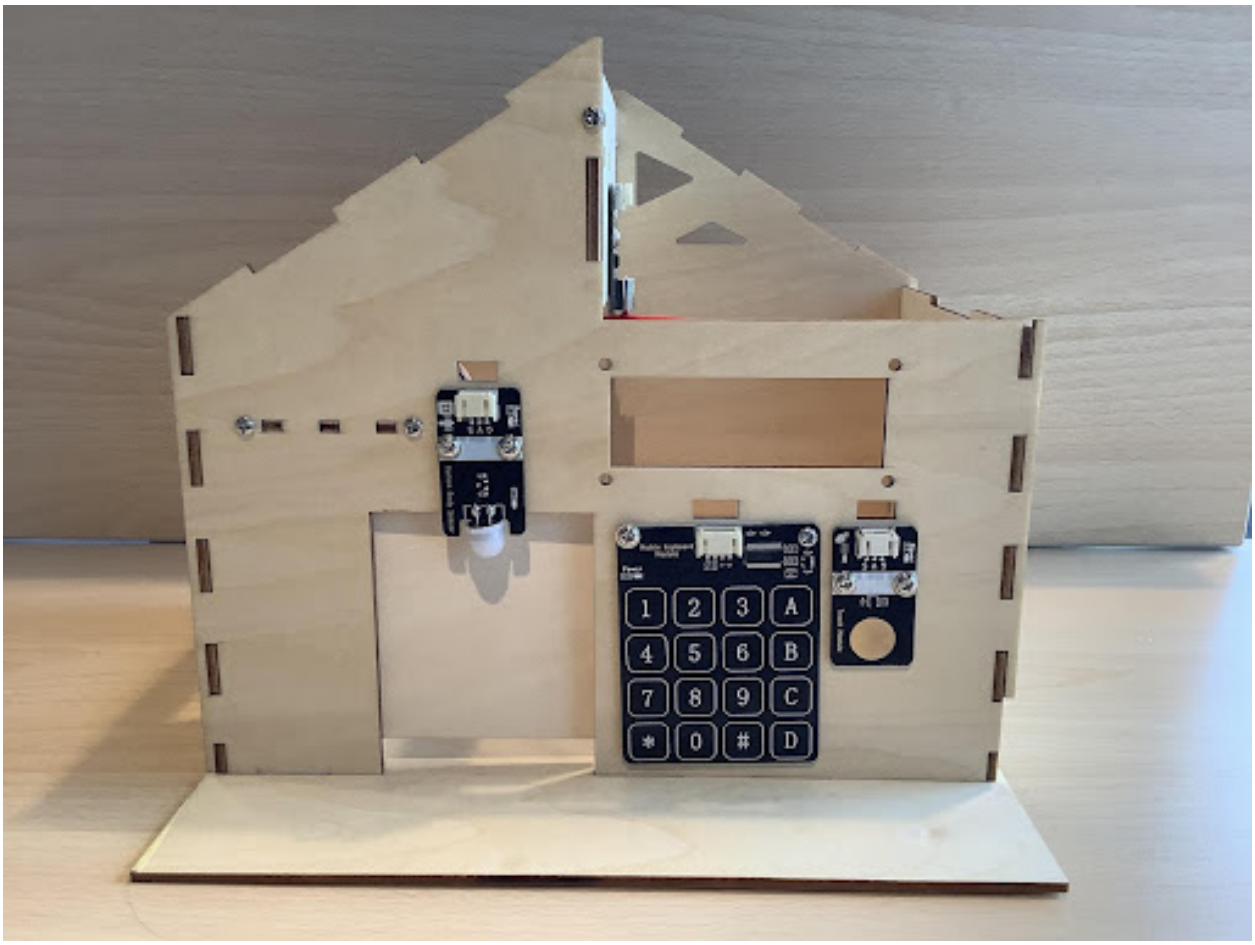
智能門禁安裝







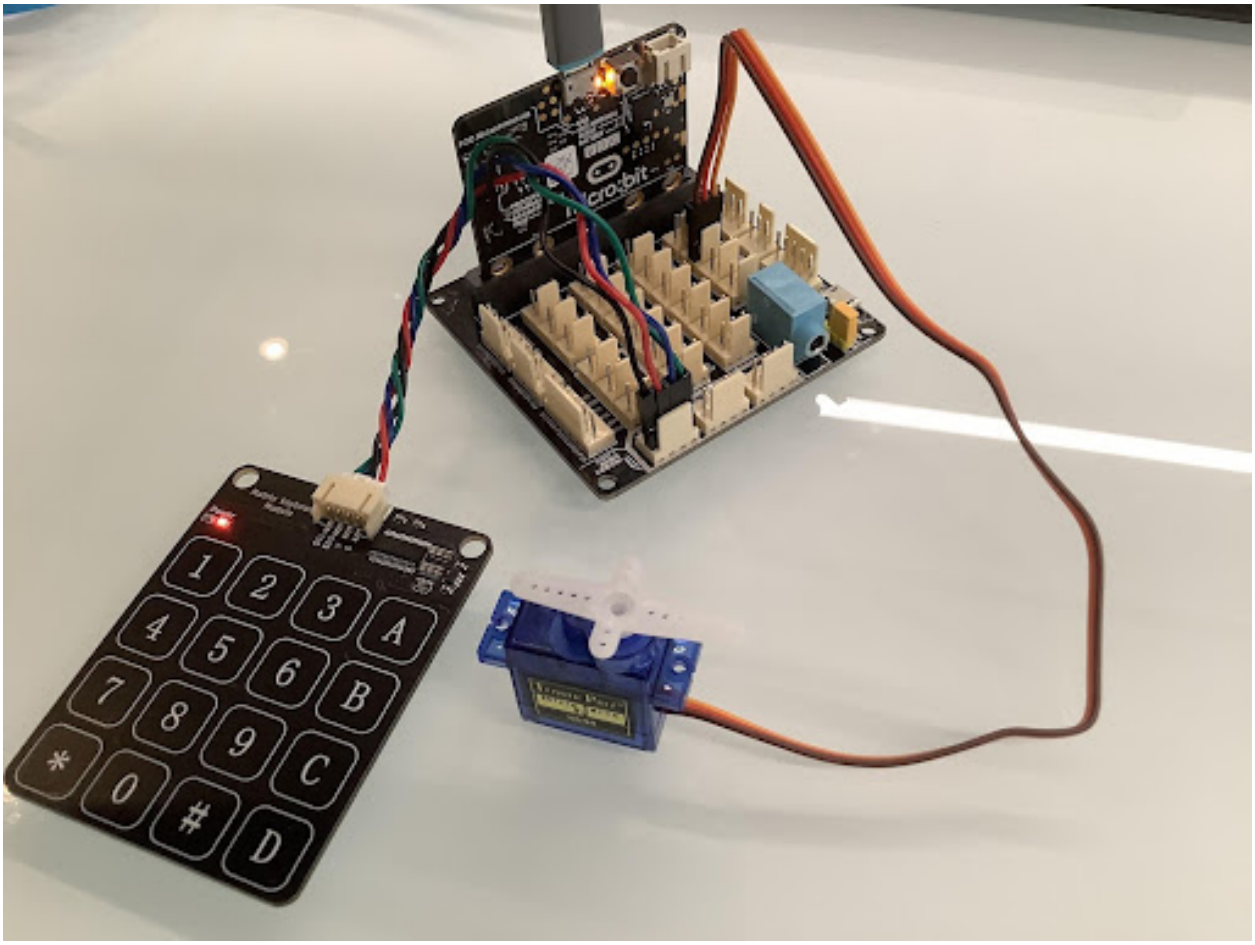




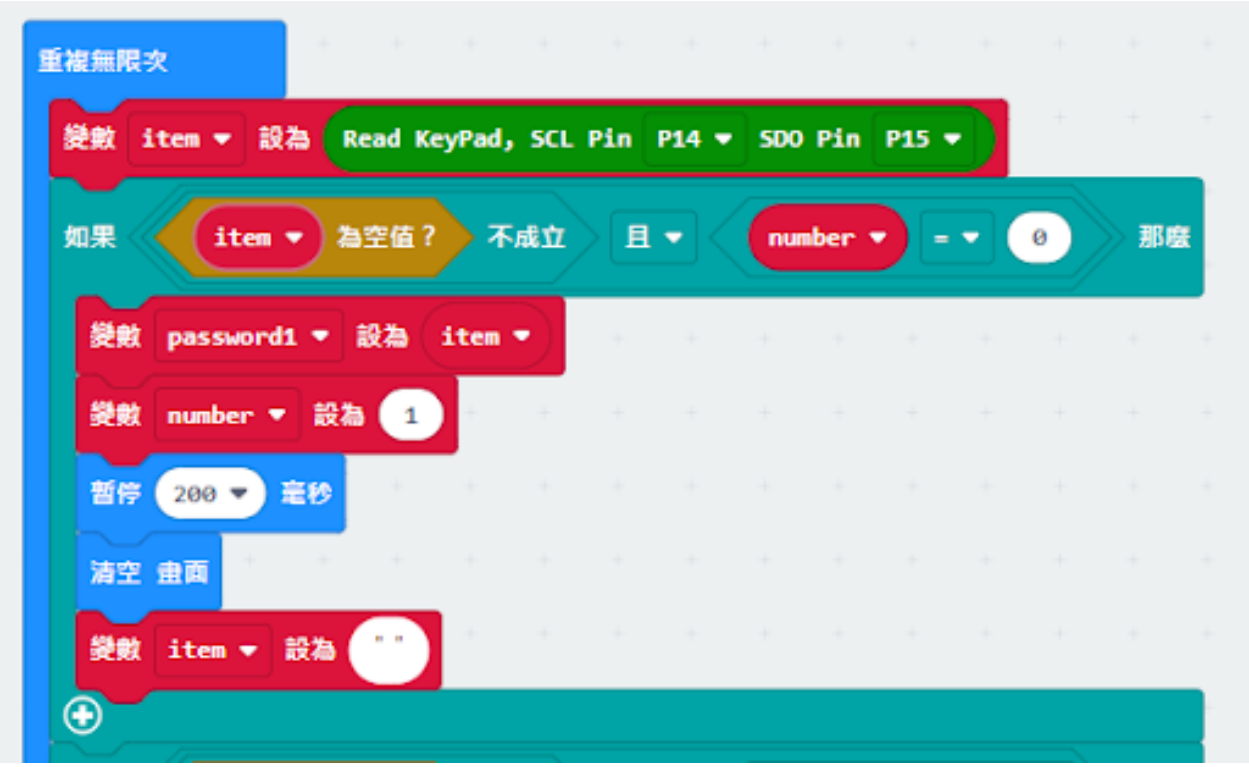
程序設計

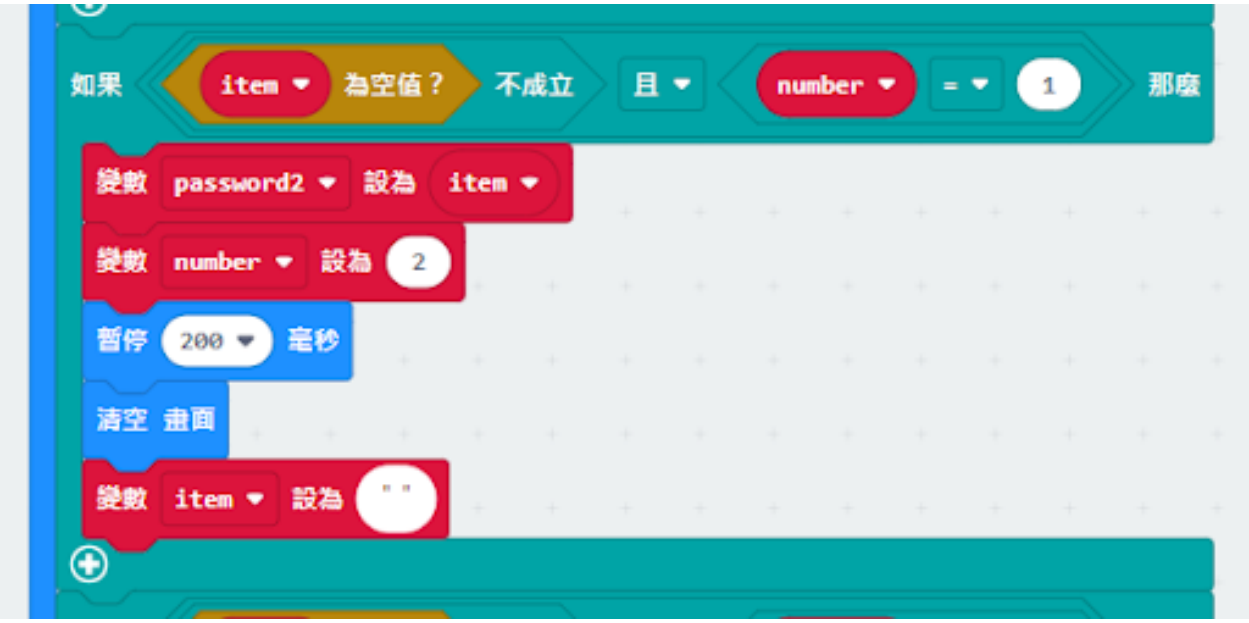
算法設計:

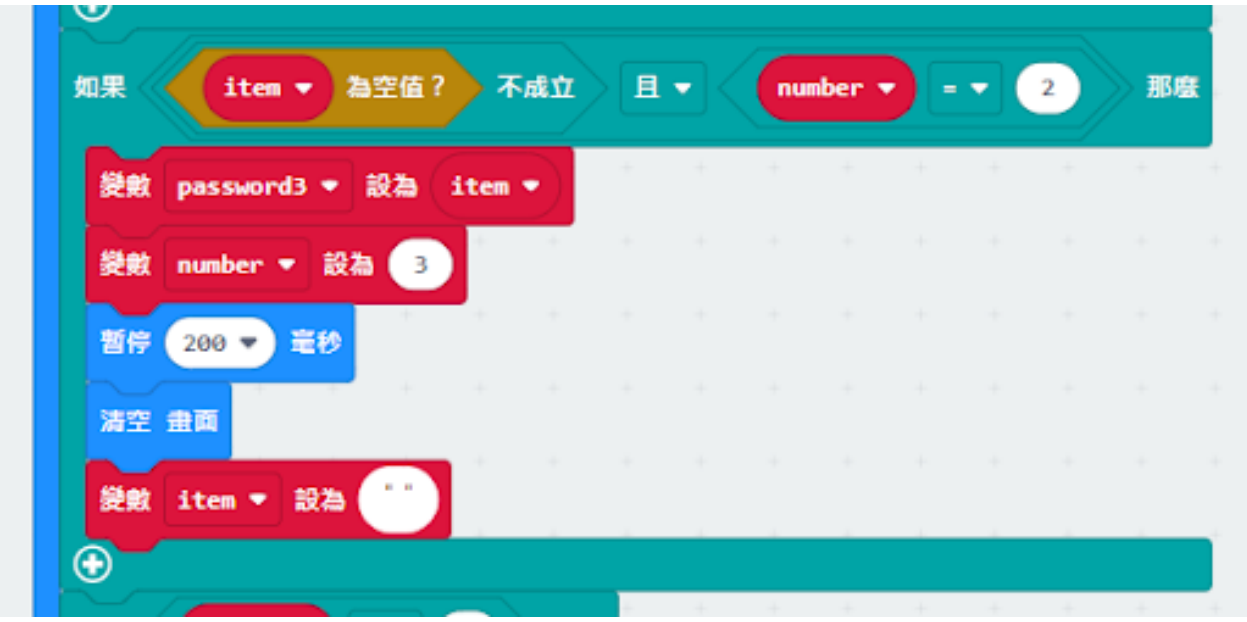
硬件連接:

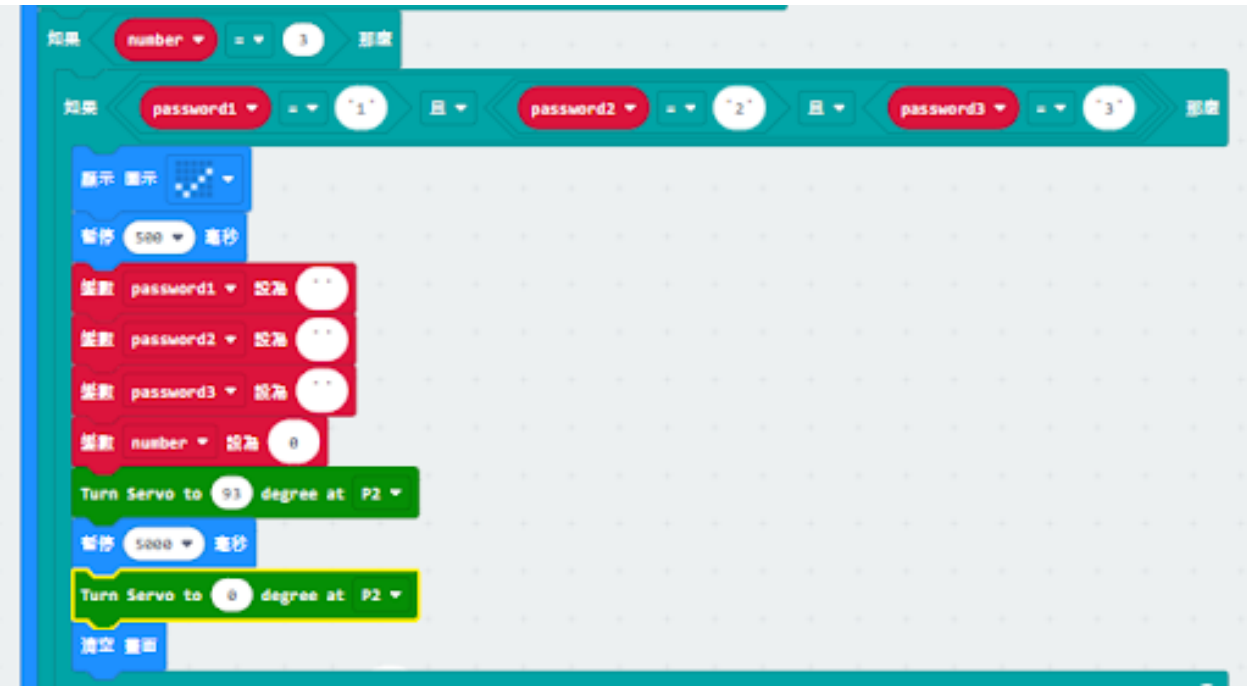


程序示例:


重複無限次
變數 item 設為 Read KeyPad, SCL Pin P14 SDO Pin P15
如果 item 為空值？ 不成立 且 number = 0 那麼
變數 password1 設為 item
變數 number 設為 1
暫停 200 毫秒
清空 畫面
變數 item 設為 " "


如果 item 為空值？ 不成立 且 number = 1 那麼
變數 password2 設為 item
變數 number 設為 2
暫停 200 毫秒
清空 畫面
變數 item 設為 " "


如果
item
為空值？
不成立
且
number
=
2
那麼
變數 password3 設為 item
變數 number 設為 3
暫停 200 毫秒
清空 畫面
變數 item 設為


如果
number
=
3
那麼
如果
password1
=
1
且
password2
=
2
且
password3
=
3
那麼
顯示 圖示
暫停 500 毫秒
變數 password1 設為
變數 password2 設為
變數 password3 設為
變數 number 設為 0
Turn Servo to 93 degree at P2
暫停 5000 毫秒
Turn Servo to 0 degree at P2
清空 畫面

總結

1.1.6 第六課光控自動窗Ⓕ

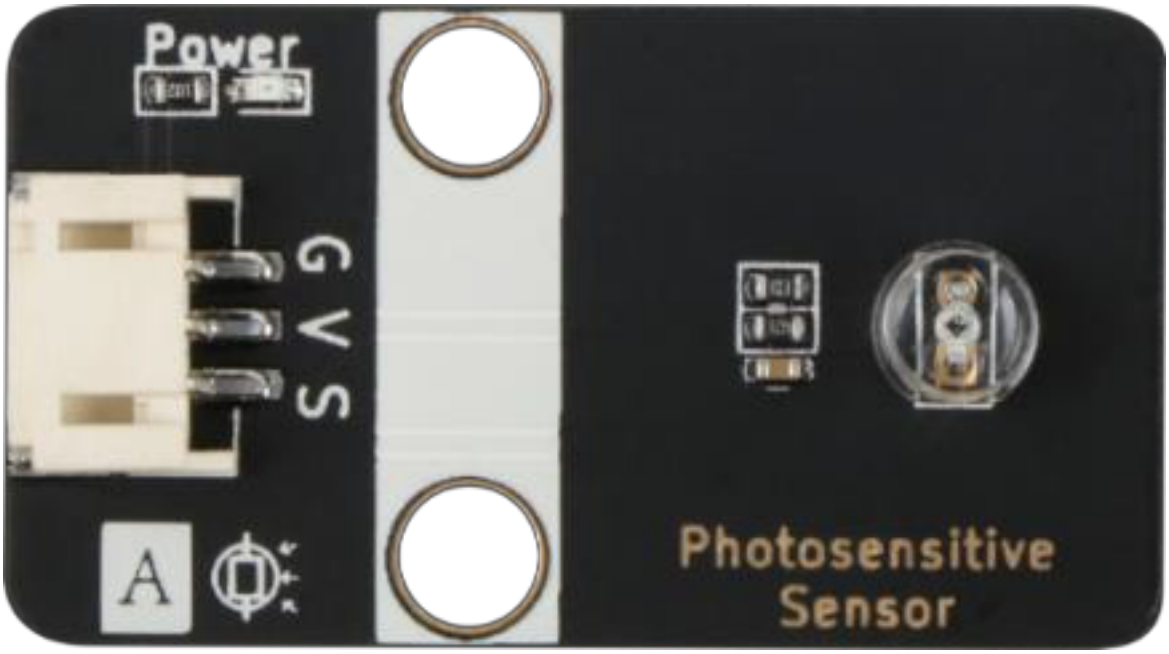
任務背景

器材準備

了解光控自動窗Ⓕ的原理

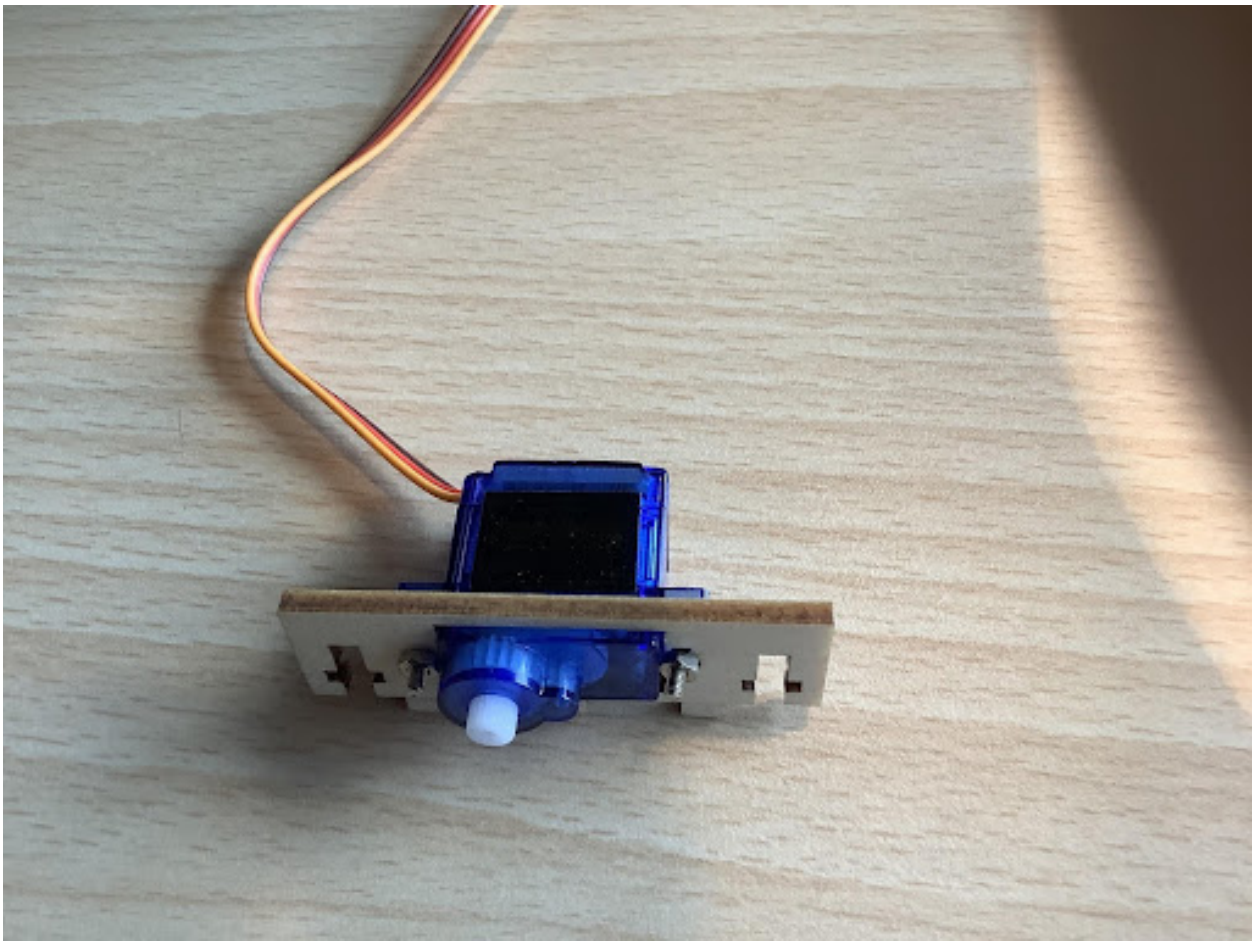
了解光敏傳感器和數碼管顯示器

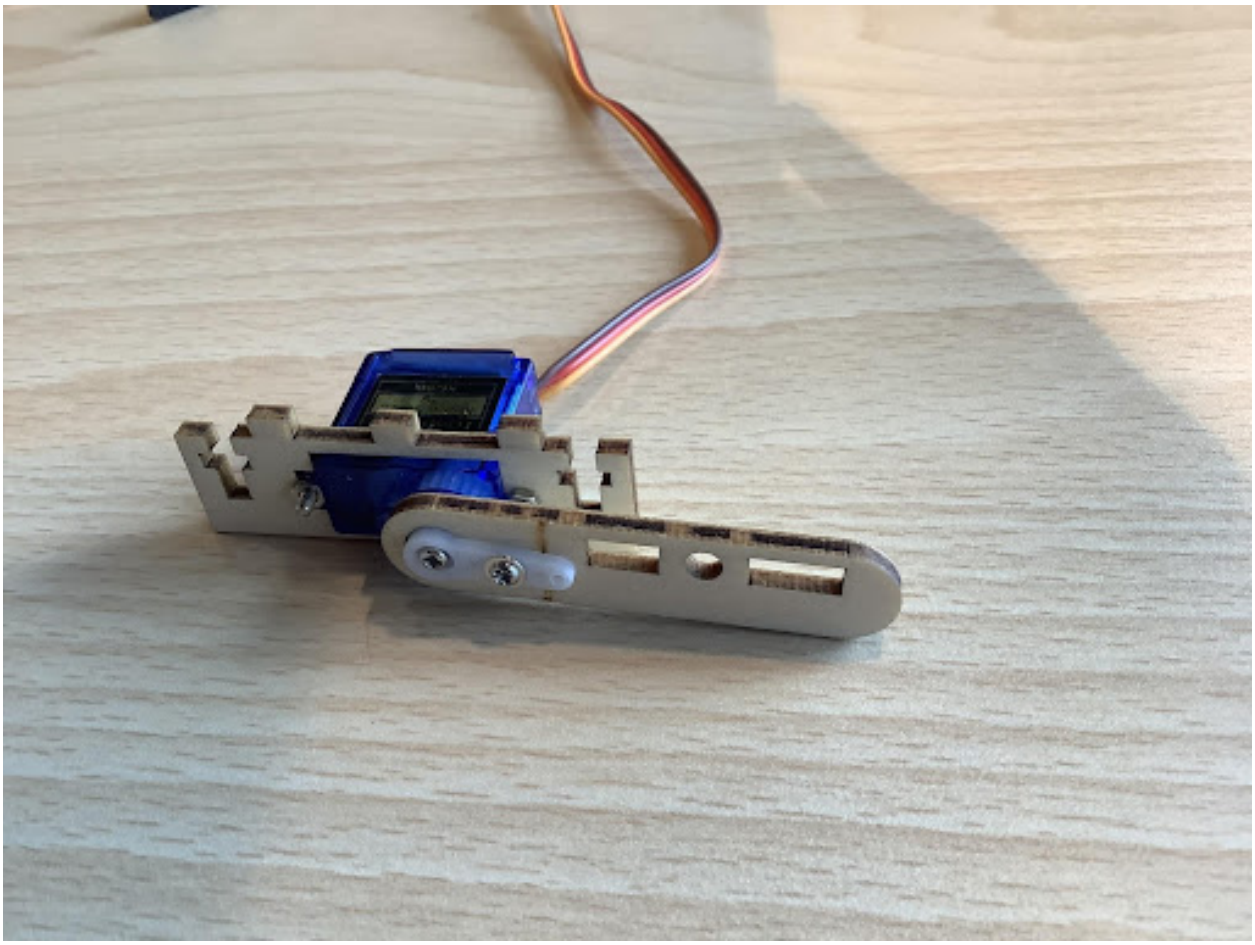
光敏傳感器

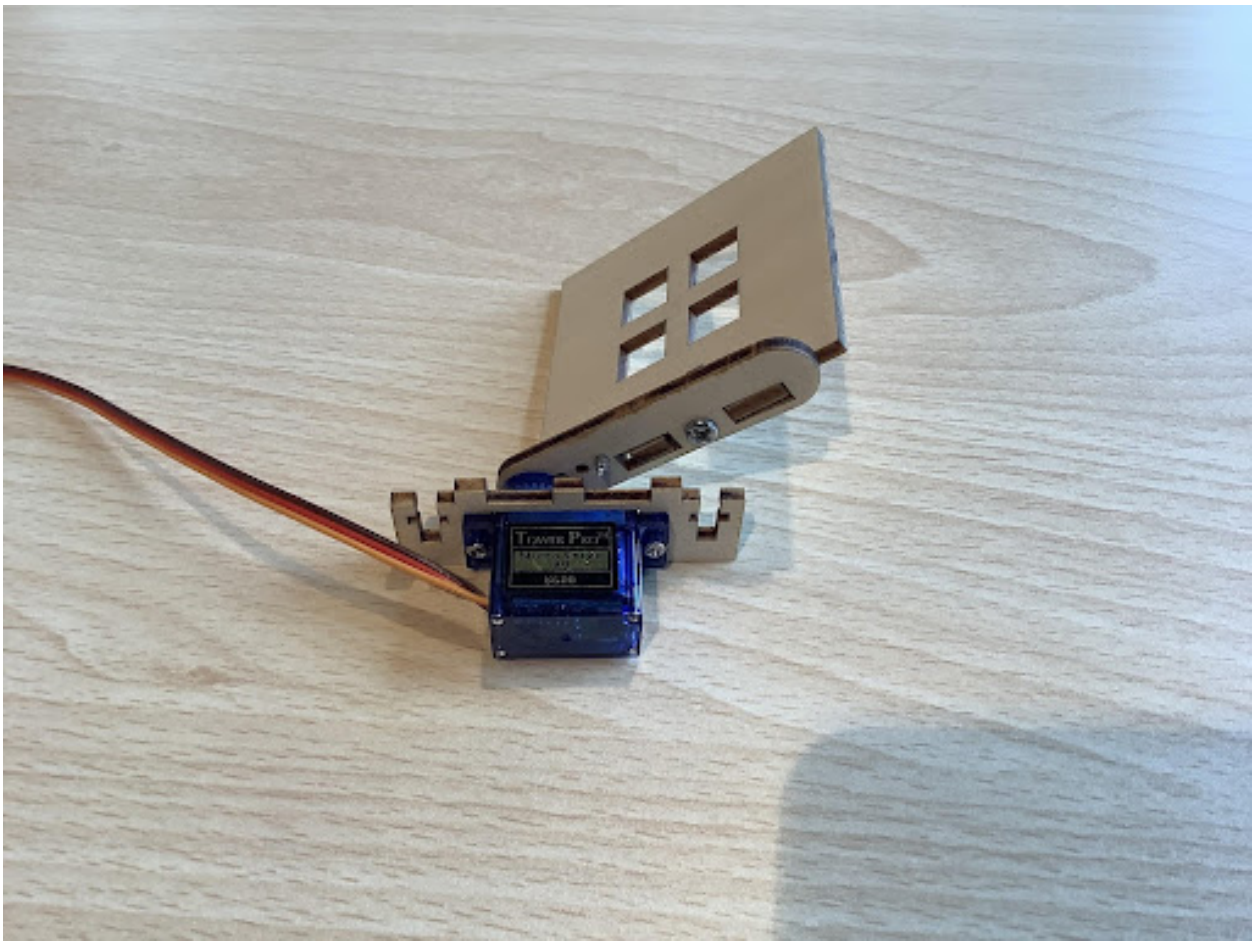


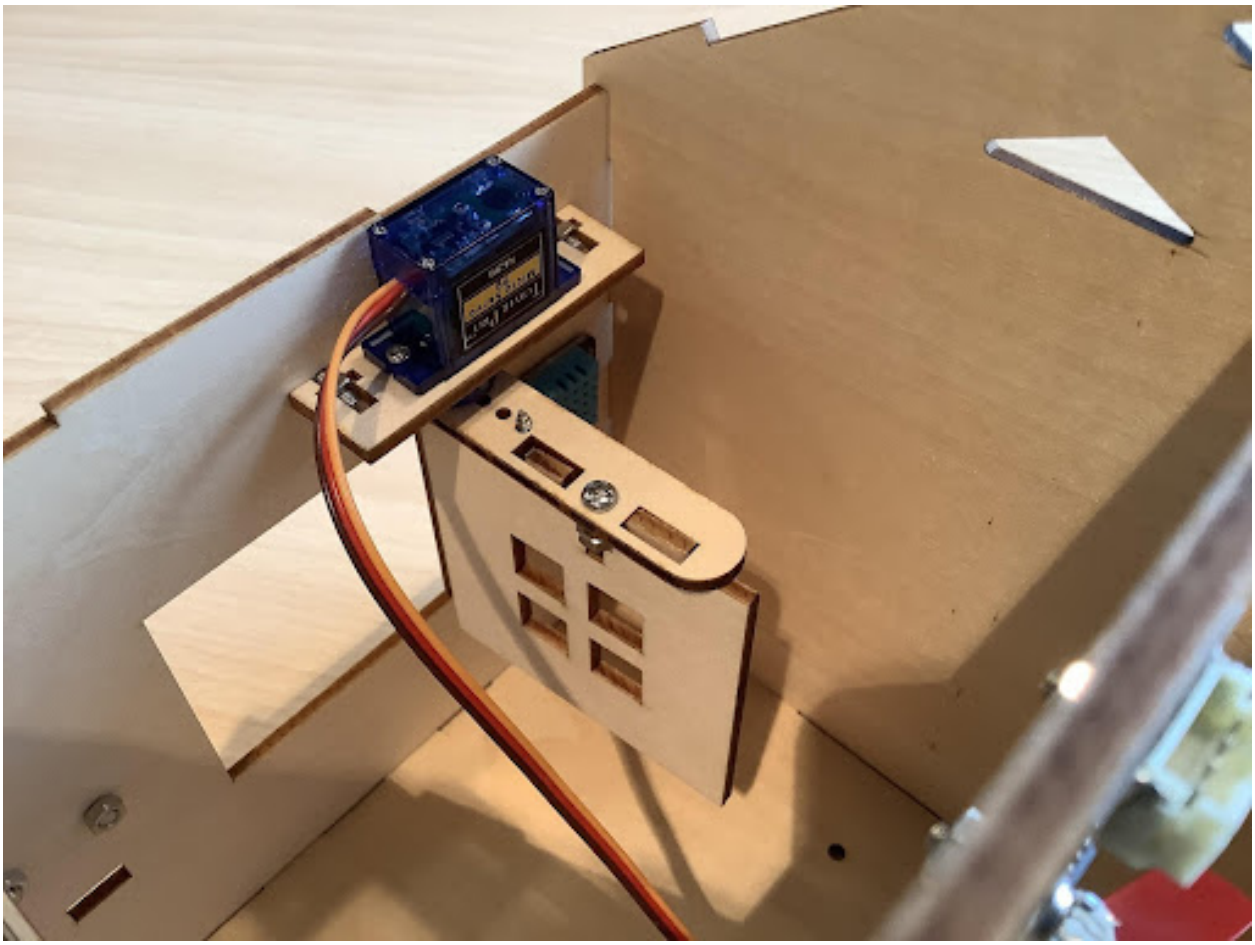
光敏傳感器	Arduino BLE-UNO
G	GND
V	VCC、5V
S	A0-A5

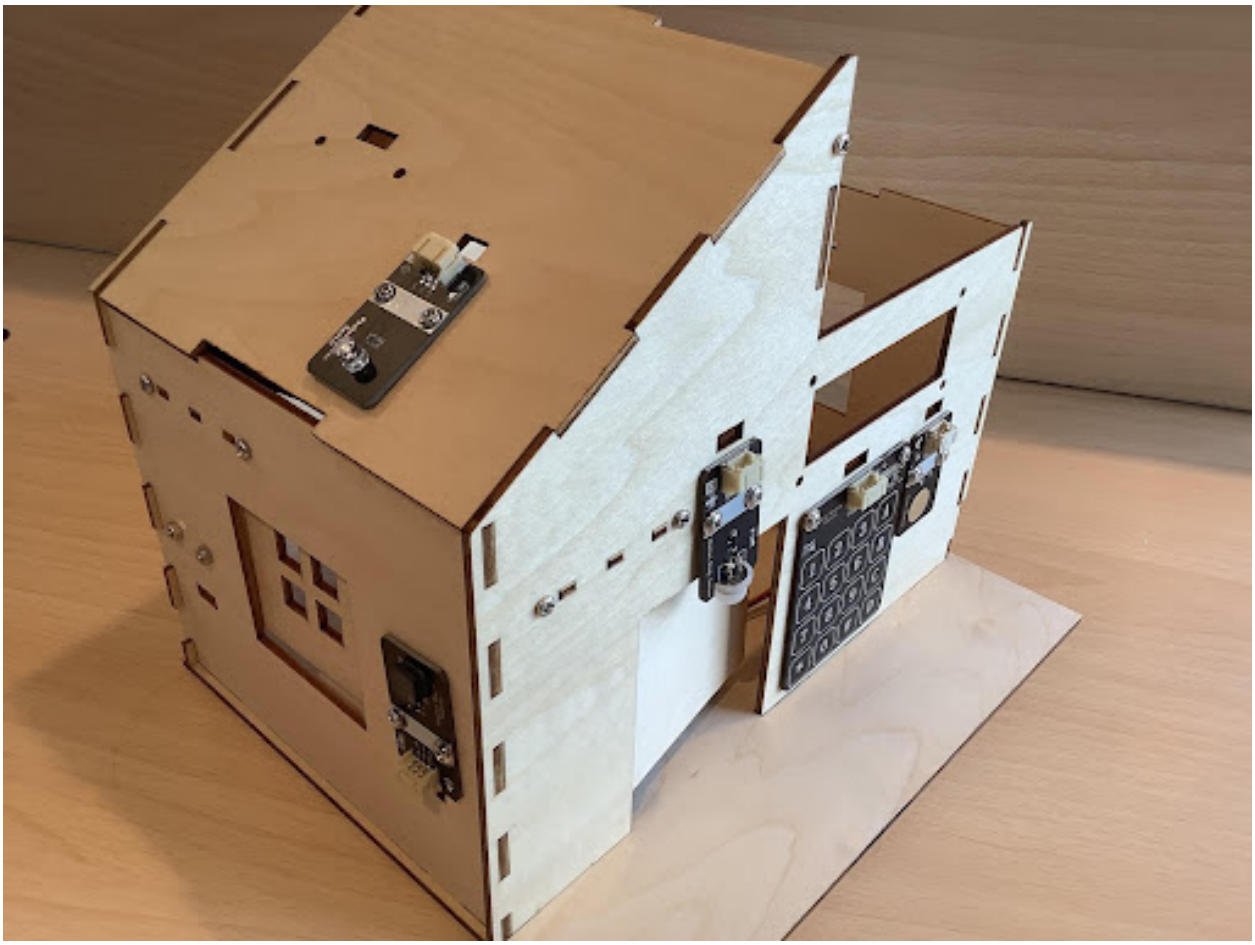
光控自動窗Ⓕ安裝





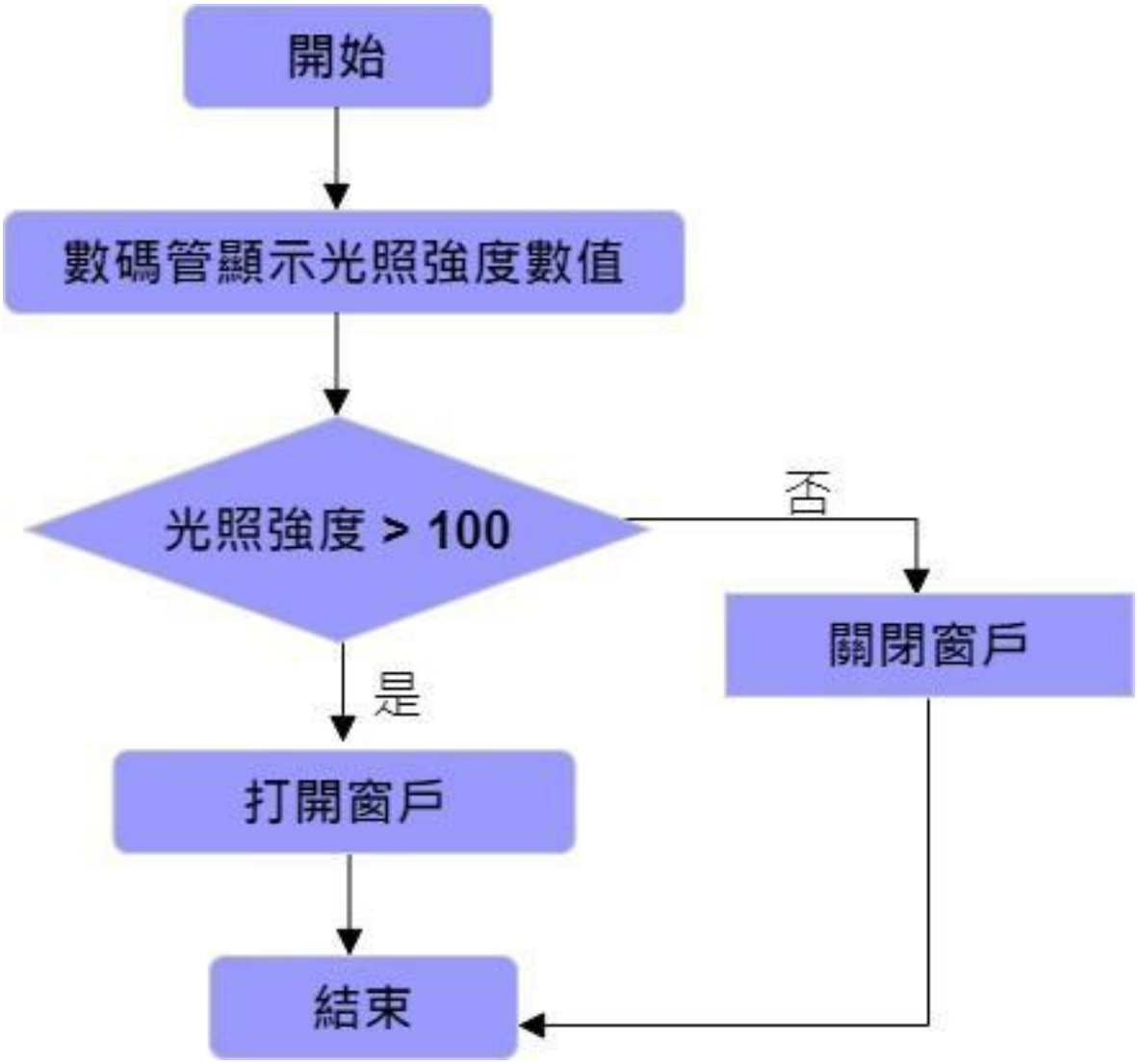

TRAVIS PICO





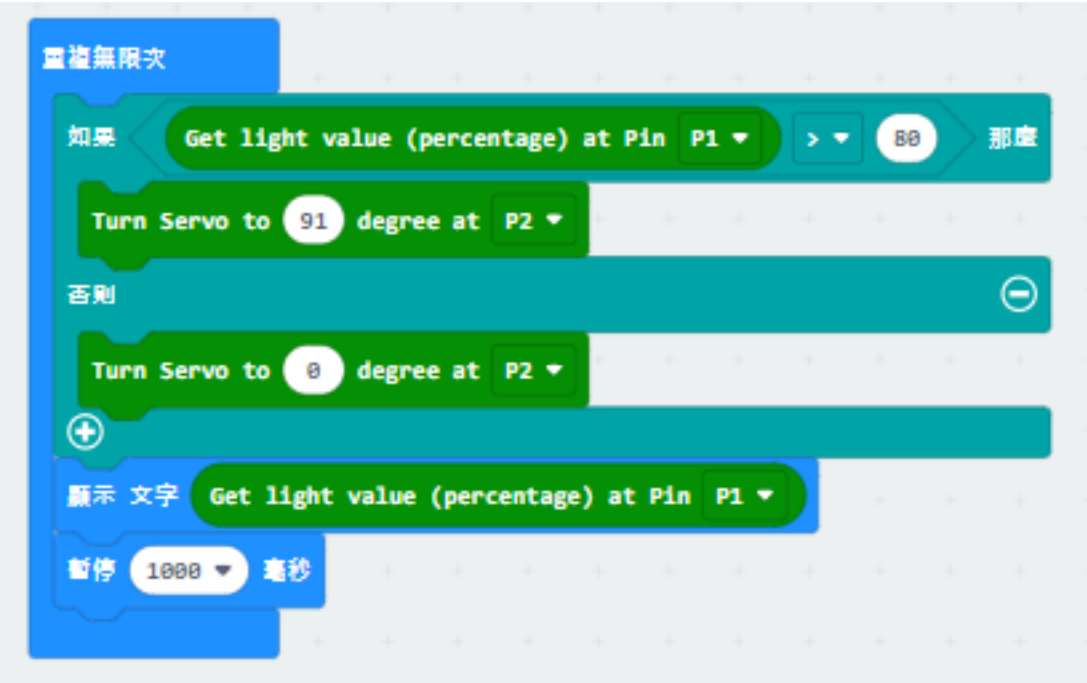
程序設計

算法設計



硬件連接

示例程序



總結

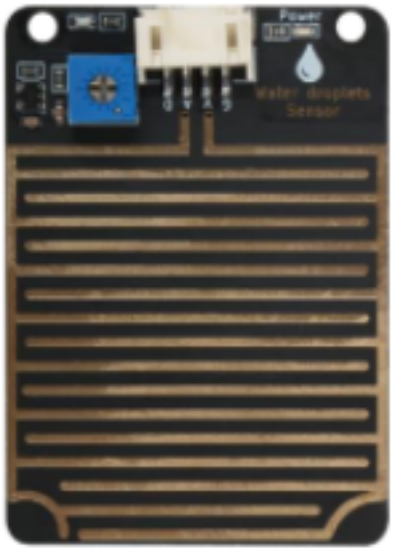
1.1.7 第七課智能雨控窗Ⓕ

任務背景

器材準備

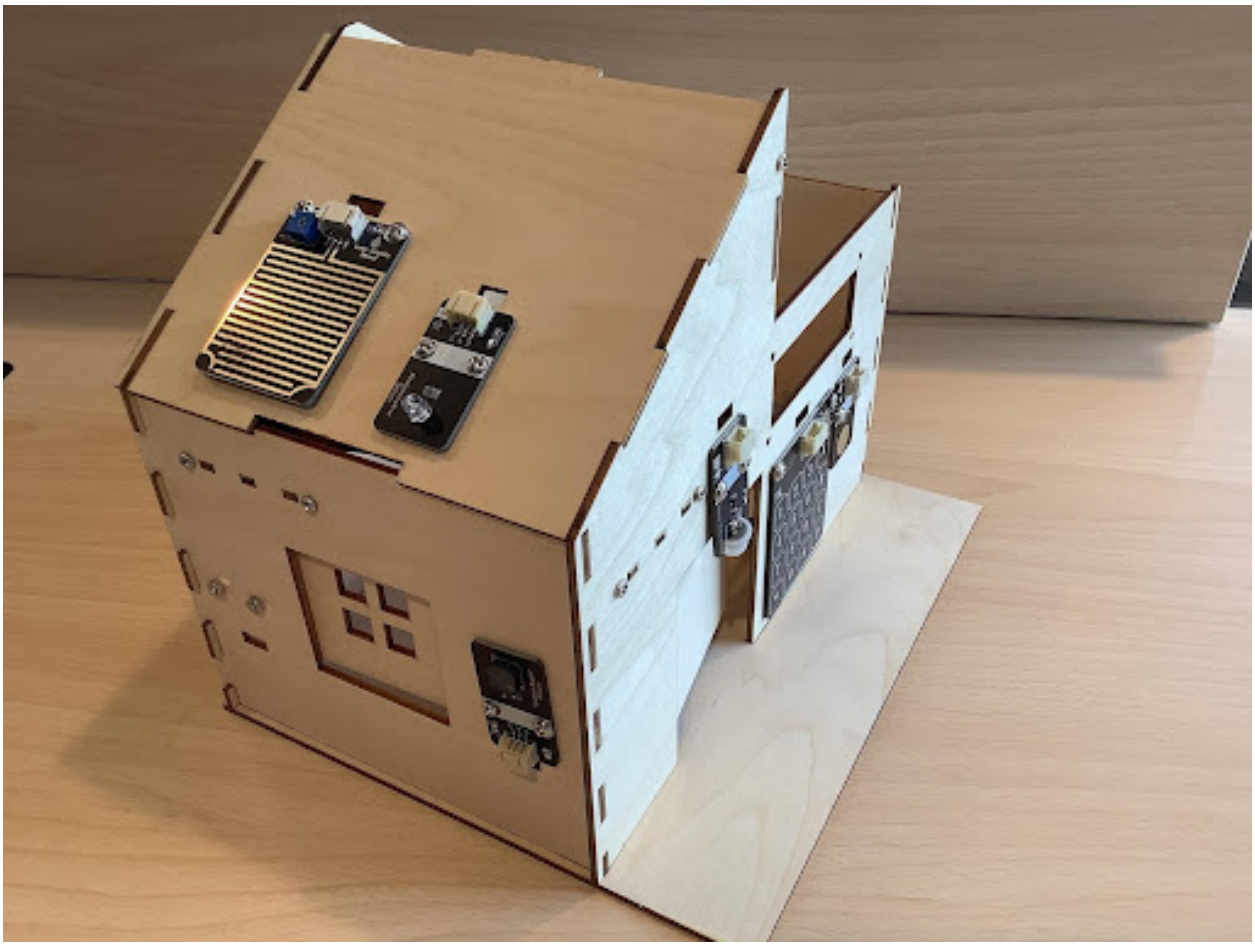
智能雨控窗Ⓕ原理

了解雨滴傳感器



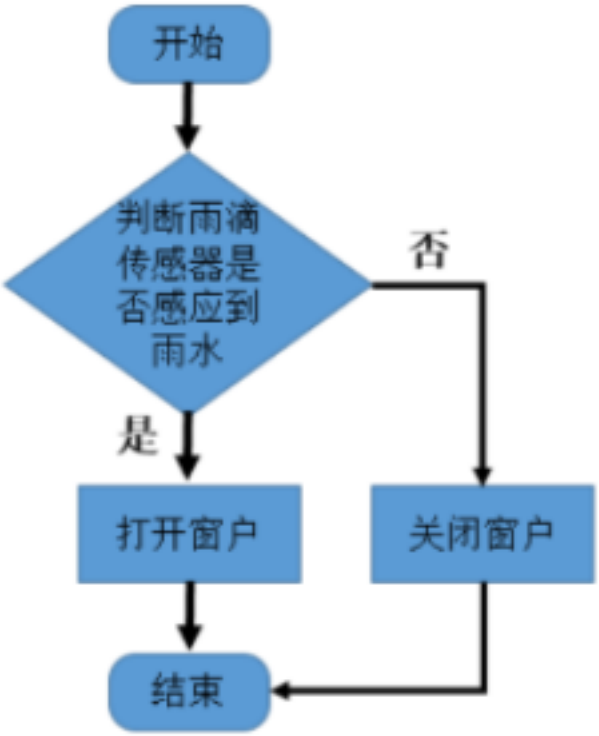
雨滴传感器	Arduino BLE-UNO
G	GND
V	VCC、5V
A	A0-A5
D	D0-D13

三、自動智能窗🅵搭建

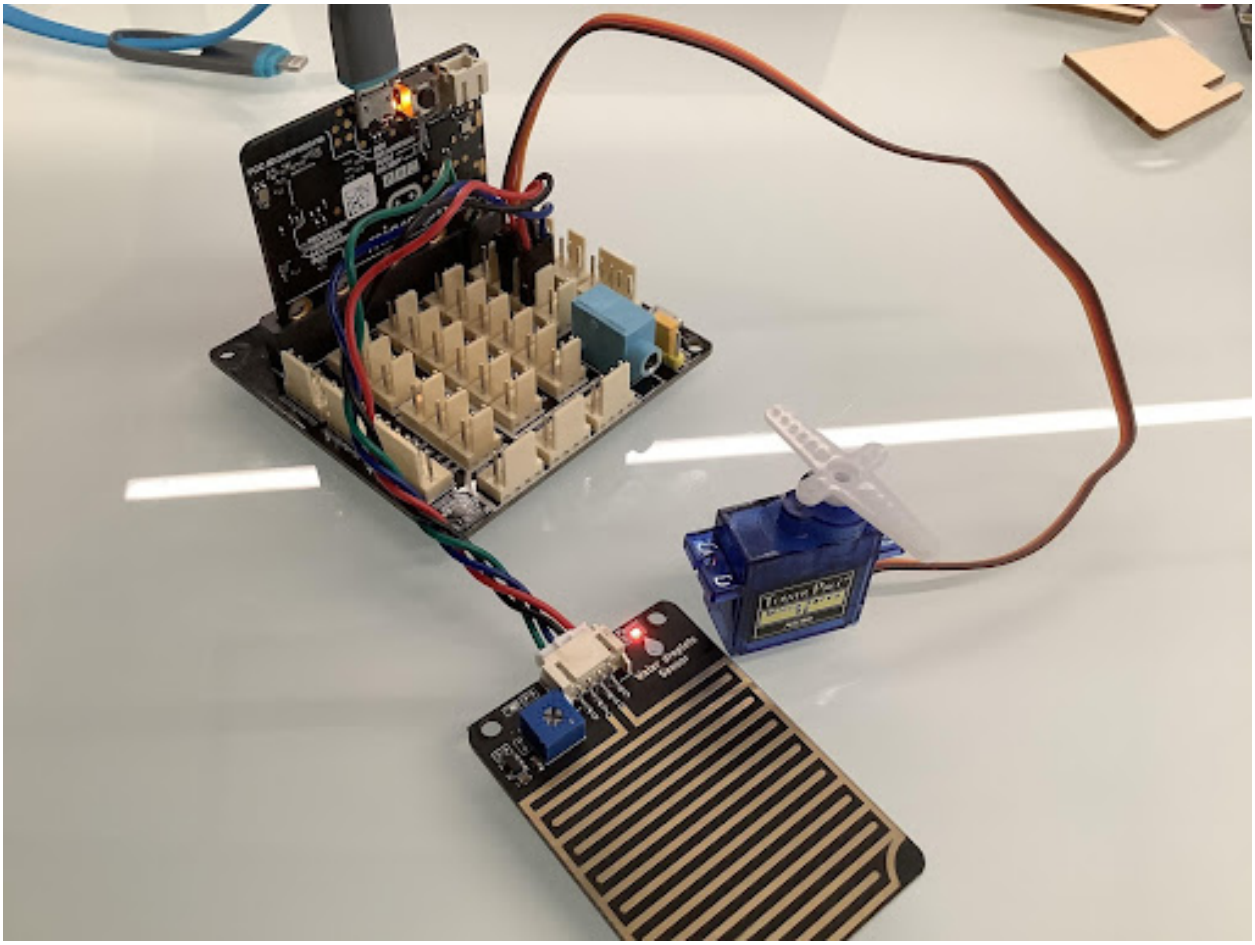


程序設計

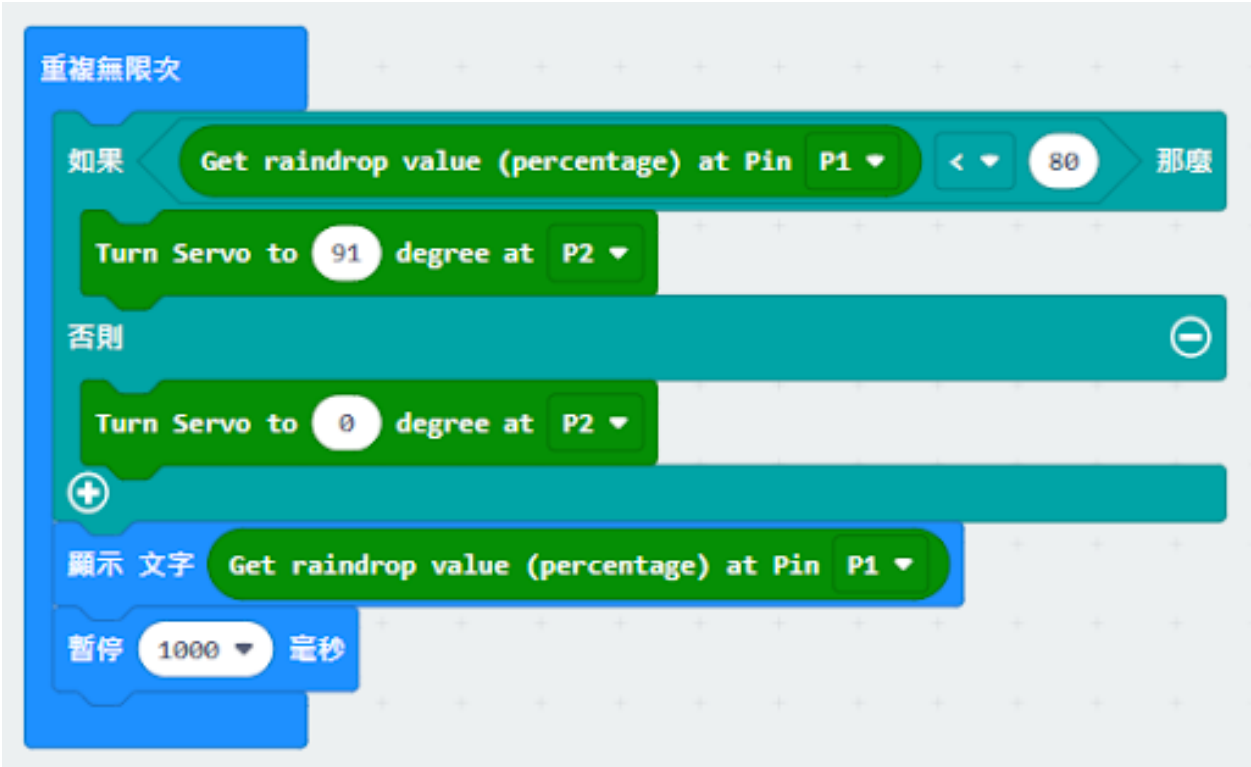
算法設計



硬件連接:



程序示例:



總結

1.1.8 第八課空氣質量檢測系統

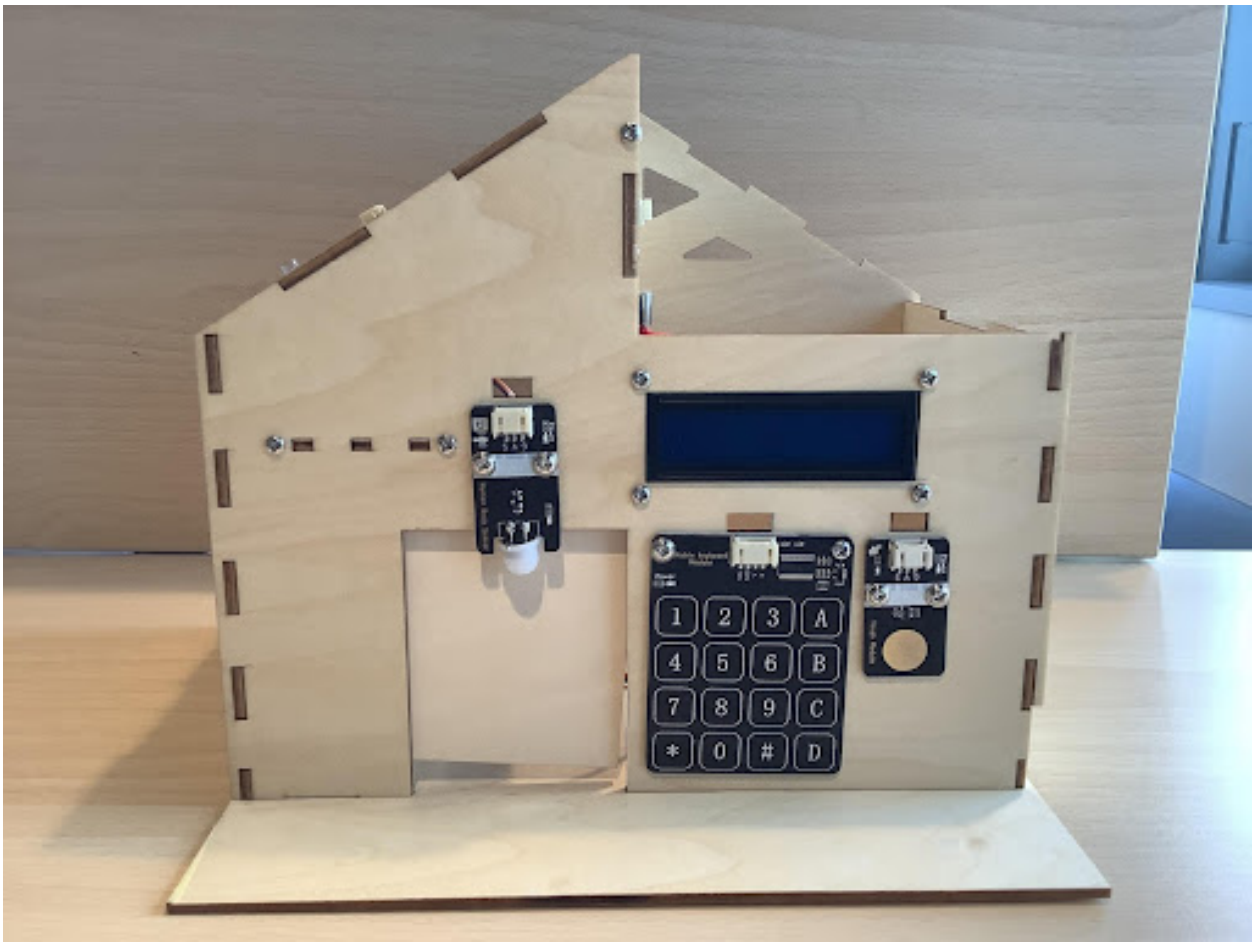
任務背景:

器材準備:

了解空氣質量檢測系統

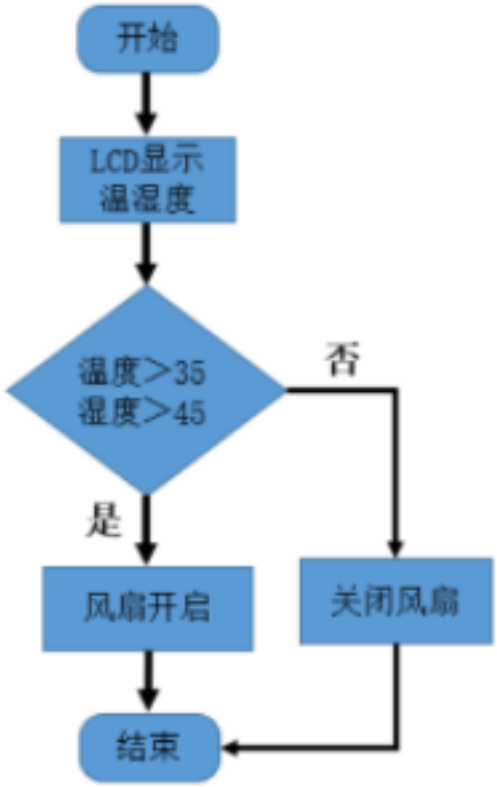
了解 LCD 顯示器

空氣檢測系統安裝

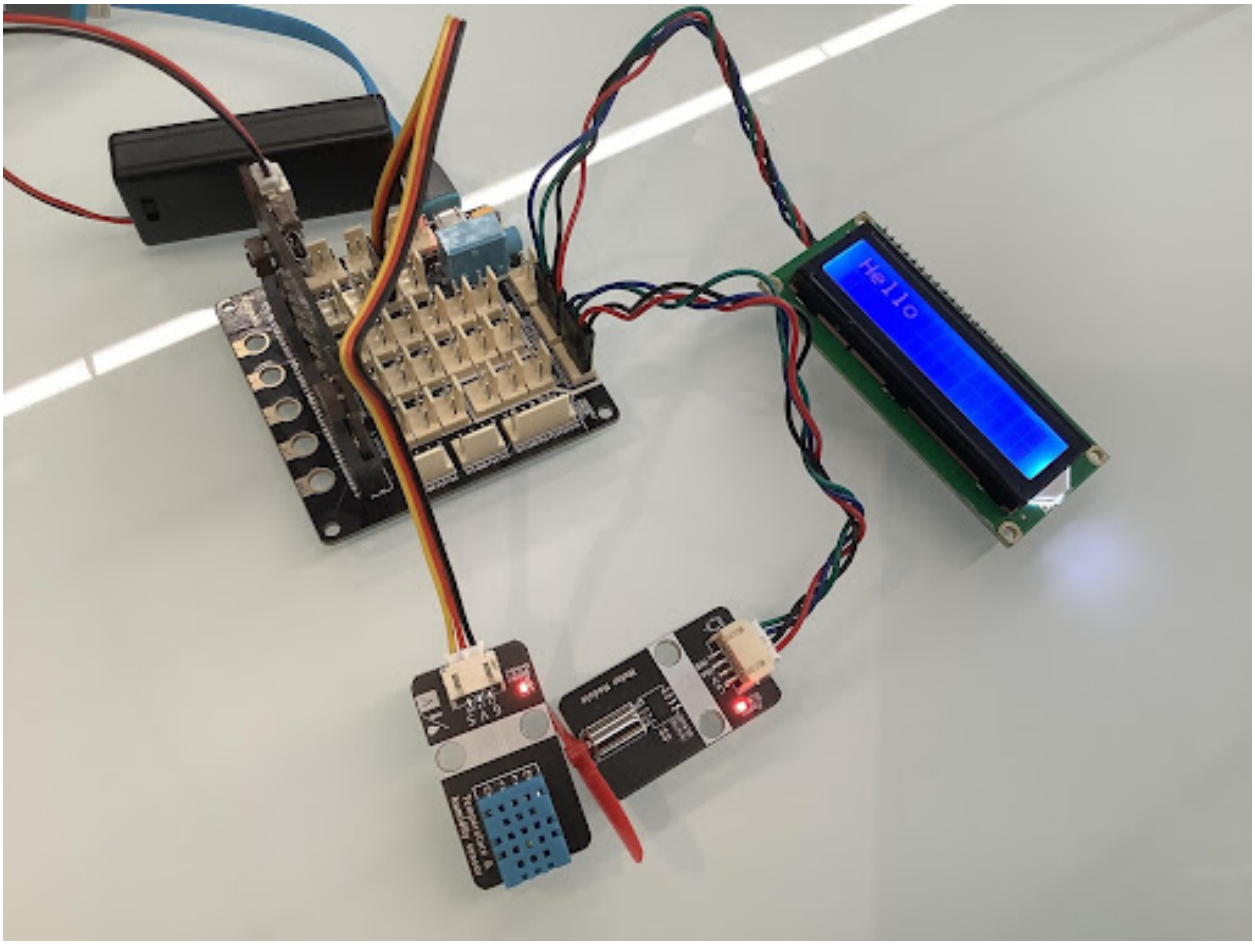


程序設計

算法設計:



硬件連接:


Hello

程序示例:

當啟動時
LCD initialize with Address 39
show string "Hello" at x 0 y 0
暫停 2000 毫秒

總結

1.2 智慧城市 IoT 套裝課程

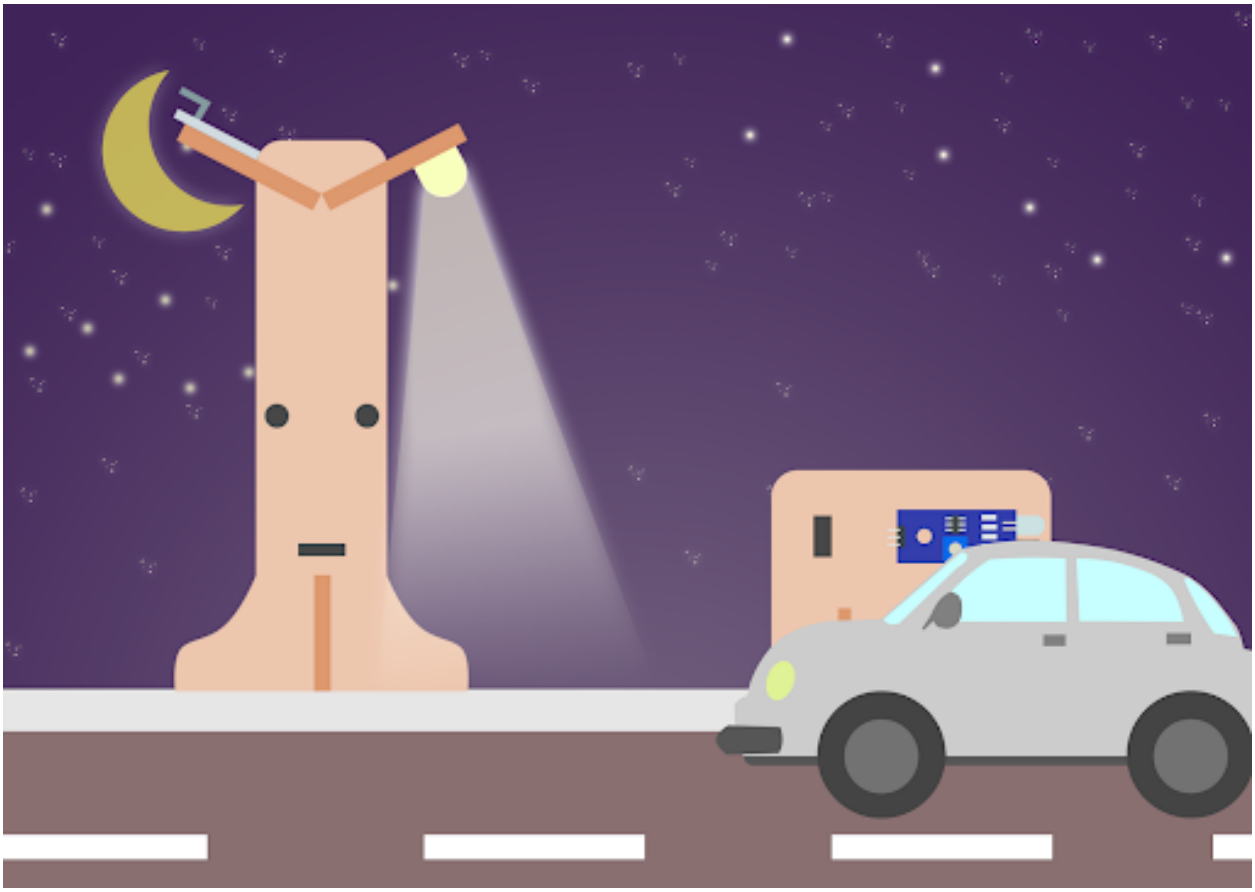
1.2.1 了解智能城市

什麽是智能城市

智慧城市的好處

- 作出更加有效、數據驅動的F定
- 加F公民和政府的參與
- 更加安全的社區
- F少環境足F
- 改善交通

1.2.2 自動智能街燈

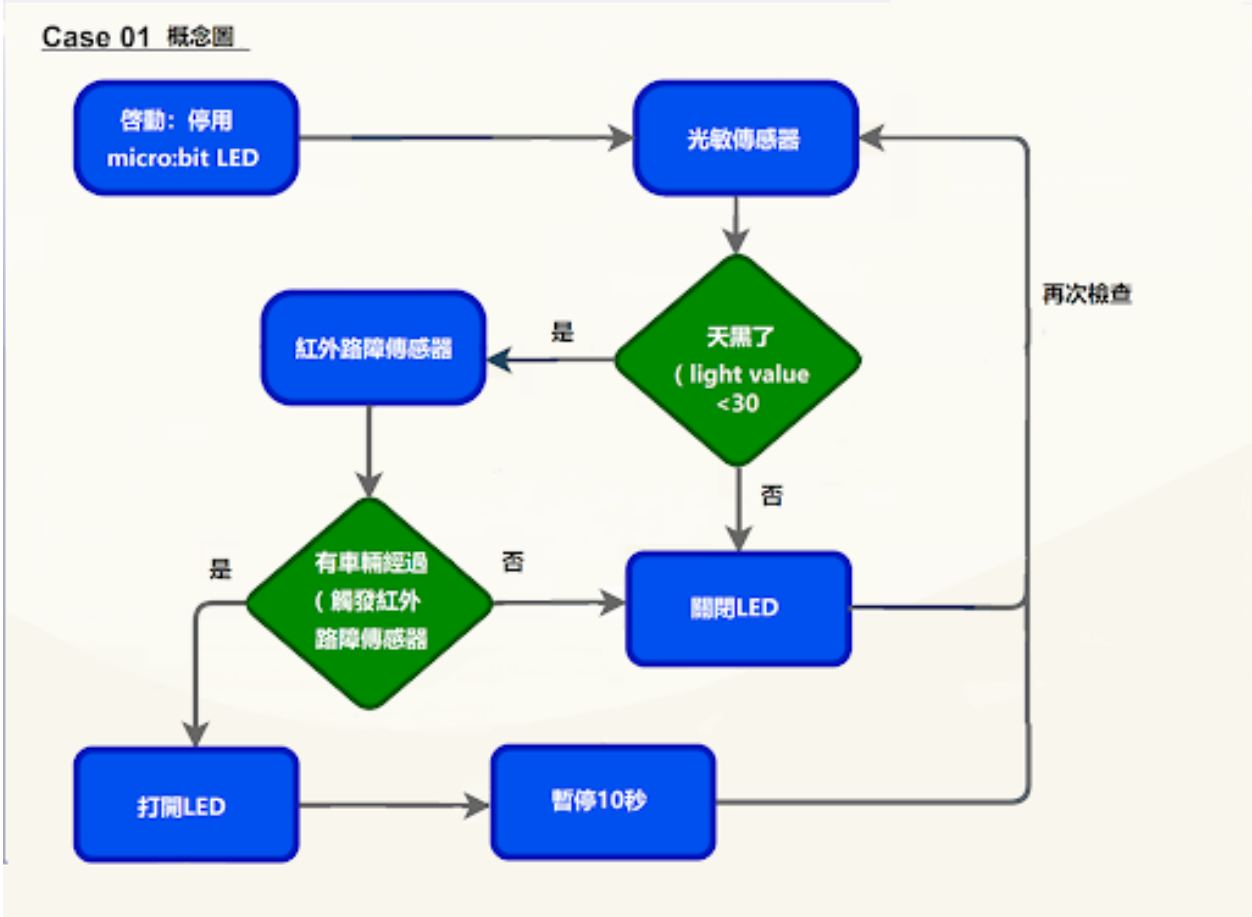


目標

背景

什麼是智能街燈

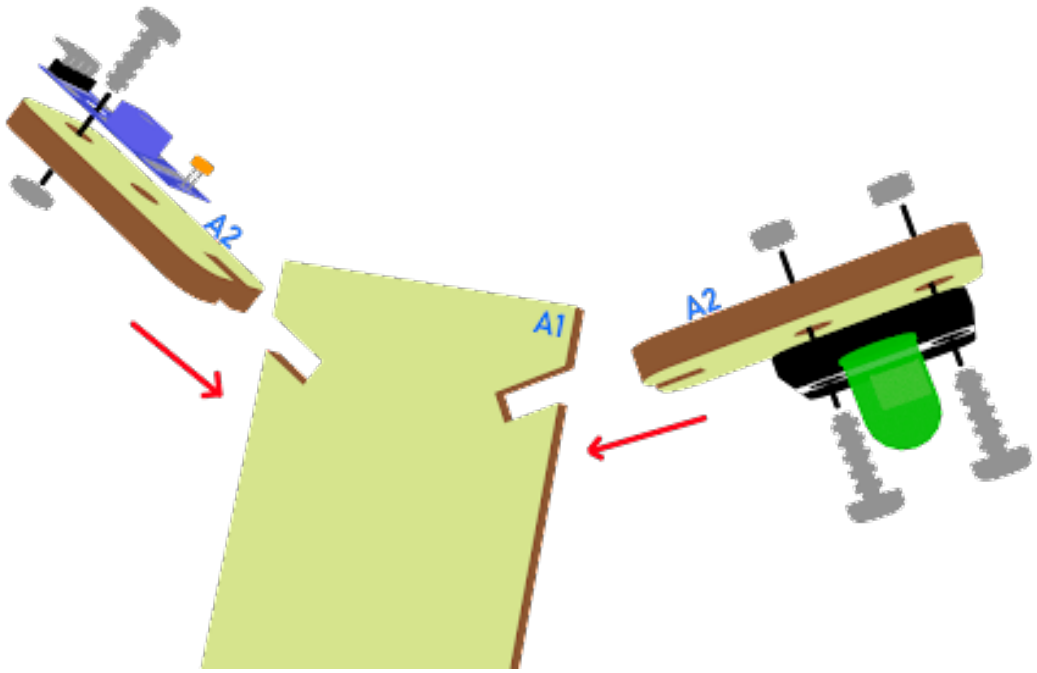
智能街燈的運作



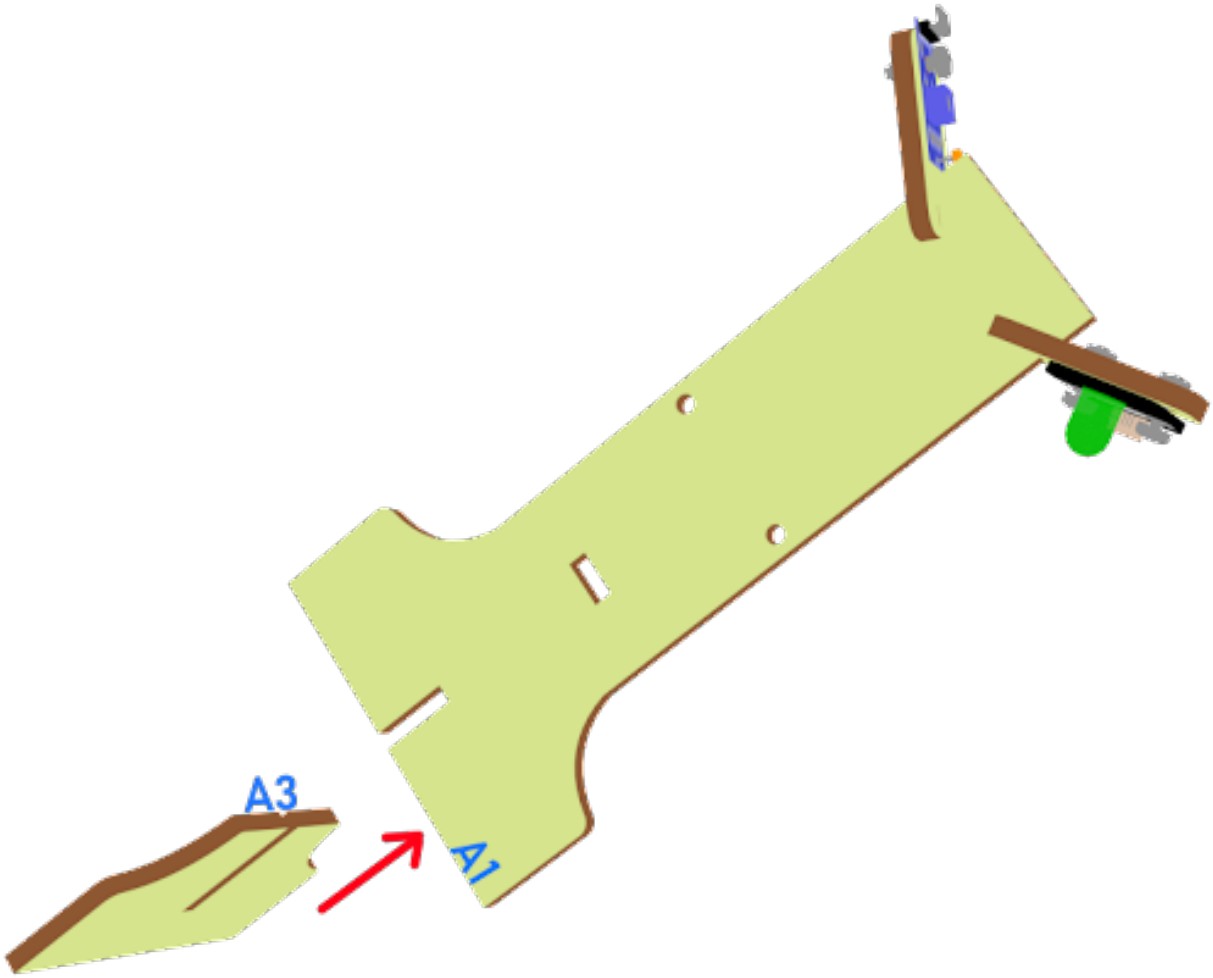
材料準備

組裝步驟

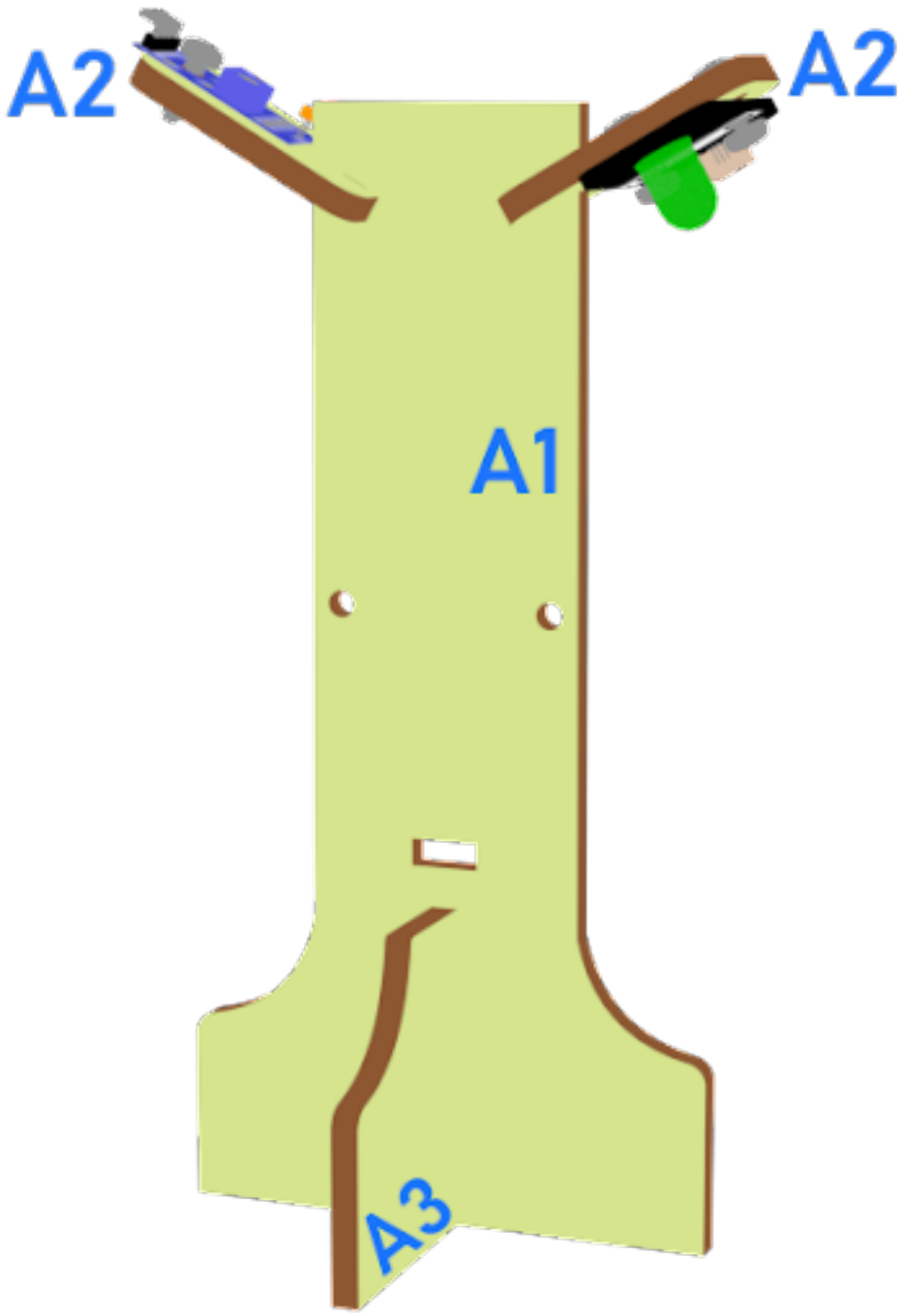
第一步



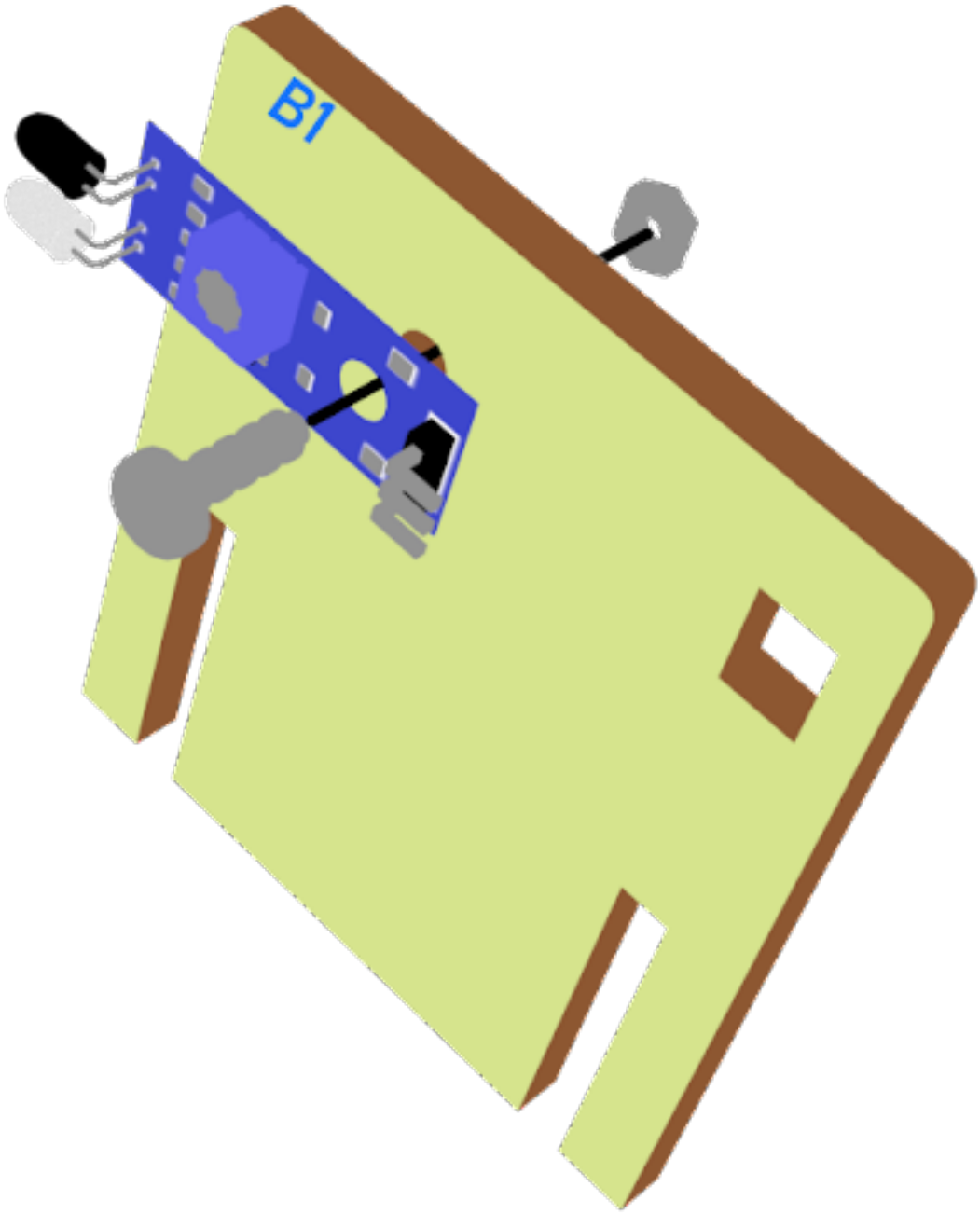
第二步



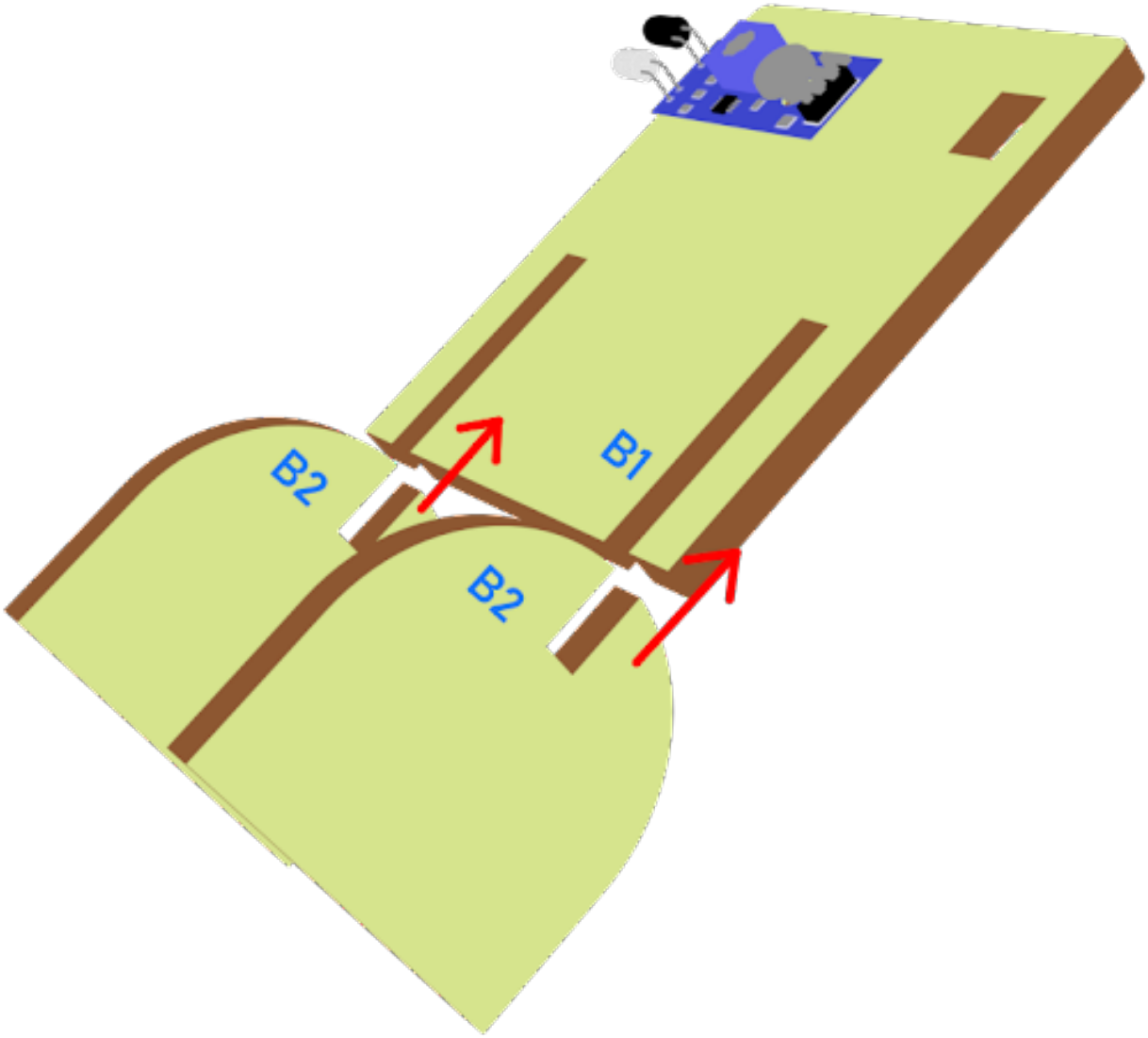
第三步



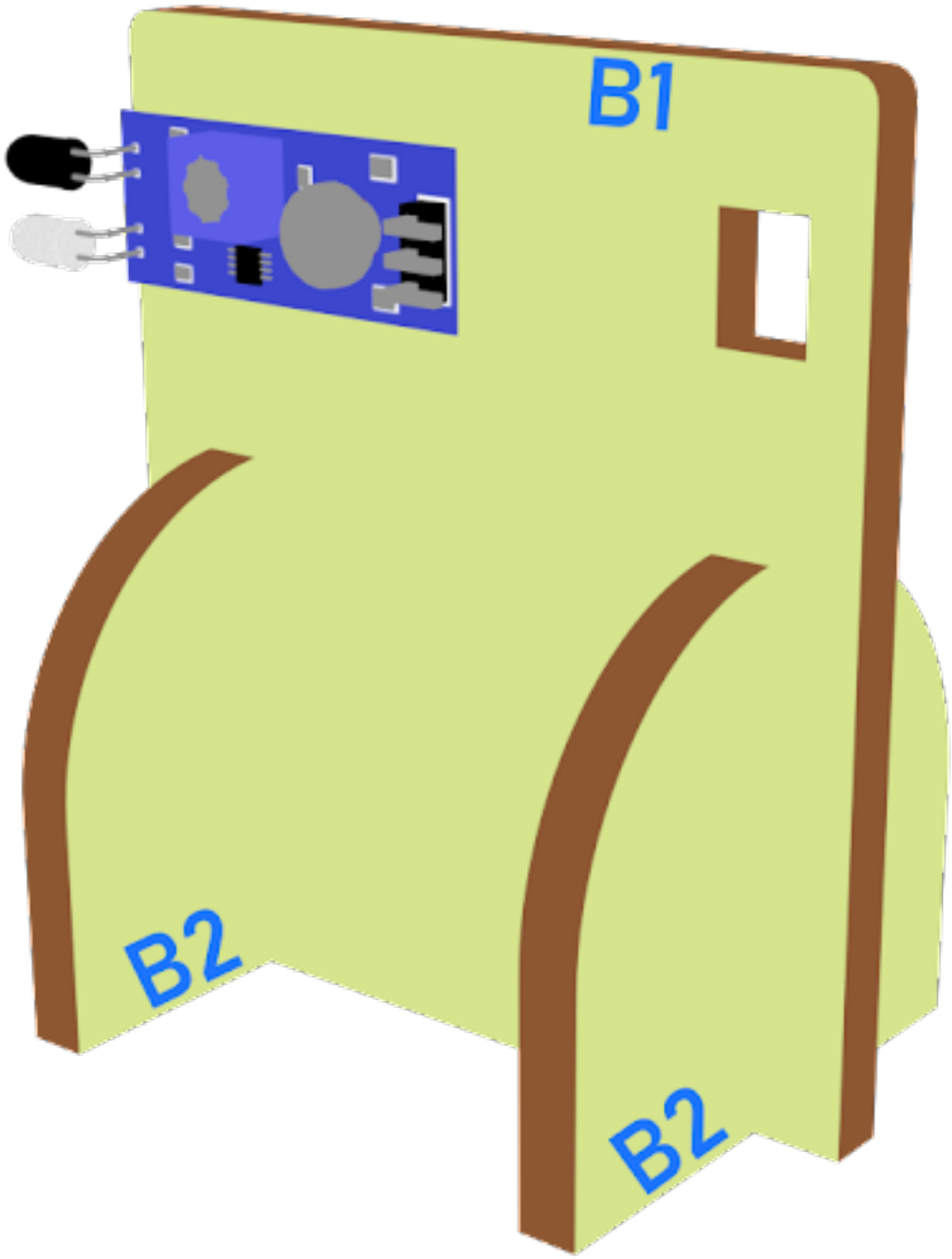
第四步



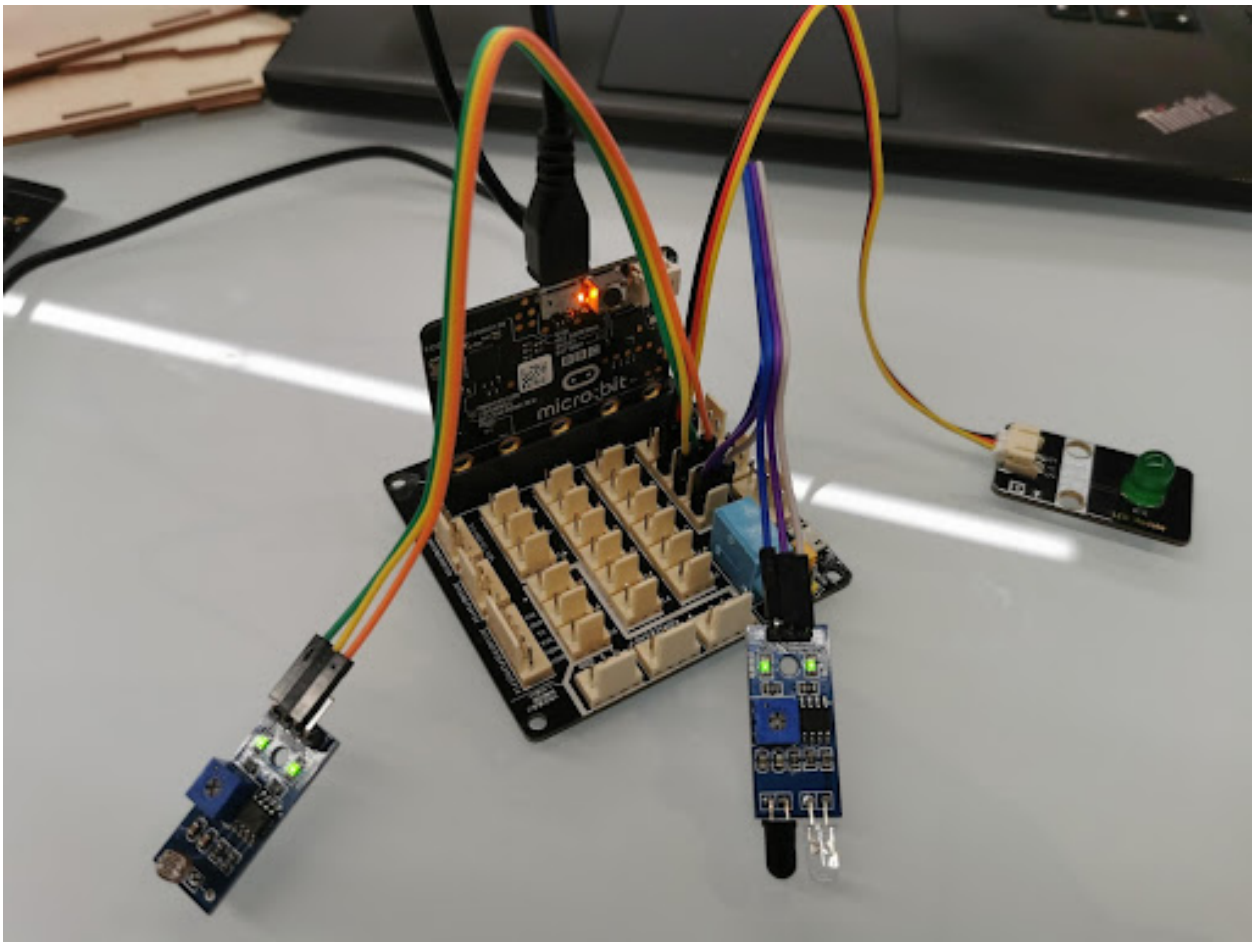
第五步



第六步



硬件連接



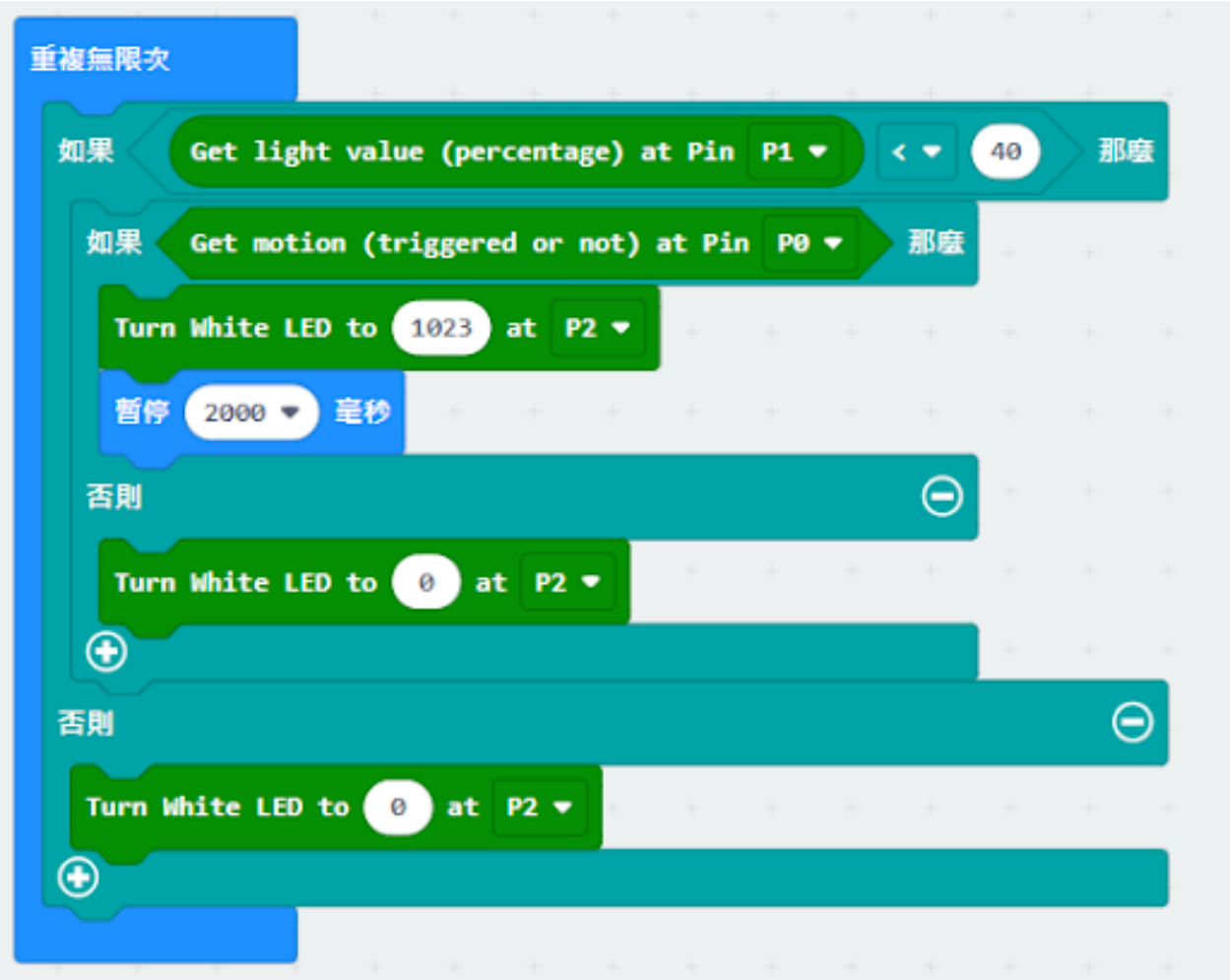
編程

第一步停用 micro:bit LED。

- 將積木啓用設爲 false 拖放到當啓動時
- 注意，P3 在默认设置中被用作 LED，因此 LED 需要被禁用。

第二步使用光敏傳感器和紅外路障傳感器打開 LED

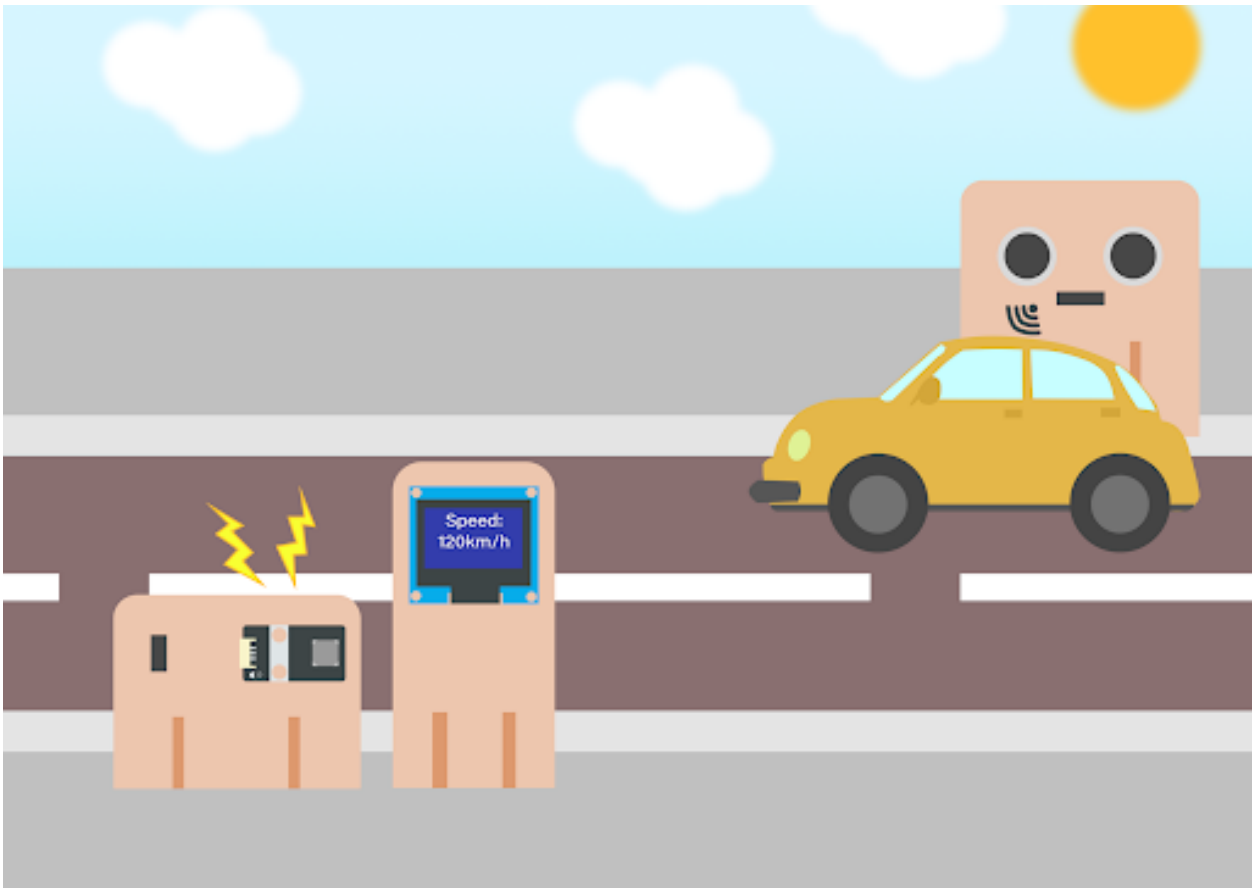
- 從基本中拖出重F無數次
- 將如果語句放入重F無數次中
- 在如果語句中設定 get light value (percentage) at P1 <40 和 get motion (triggered or not) at P0 = true。
- 然后，turn white LED to 1023 at P2，打開白色 LED F暫停 10 秒。
- 否則，turn white LED at P2 to 0，關閉白色 LED。



總結

思考

1.2.3 車速監測



目標

背景

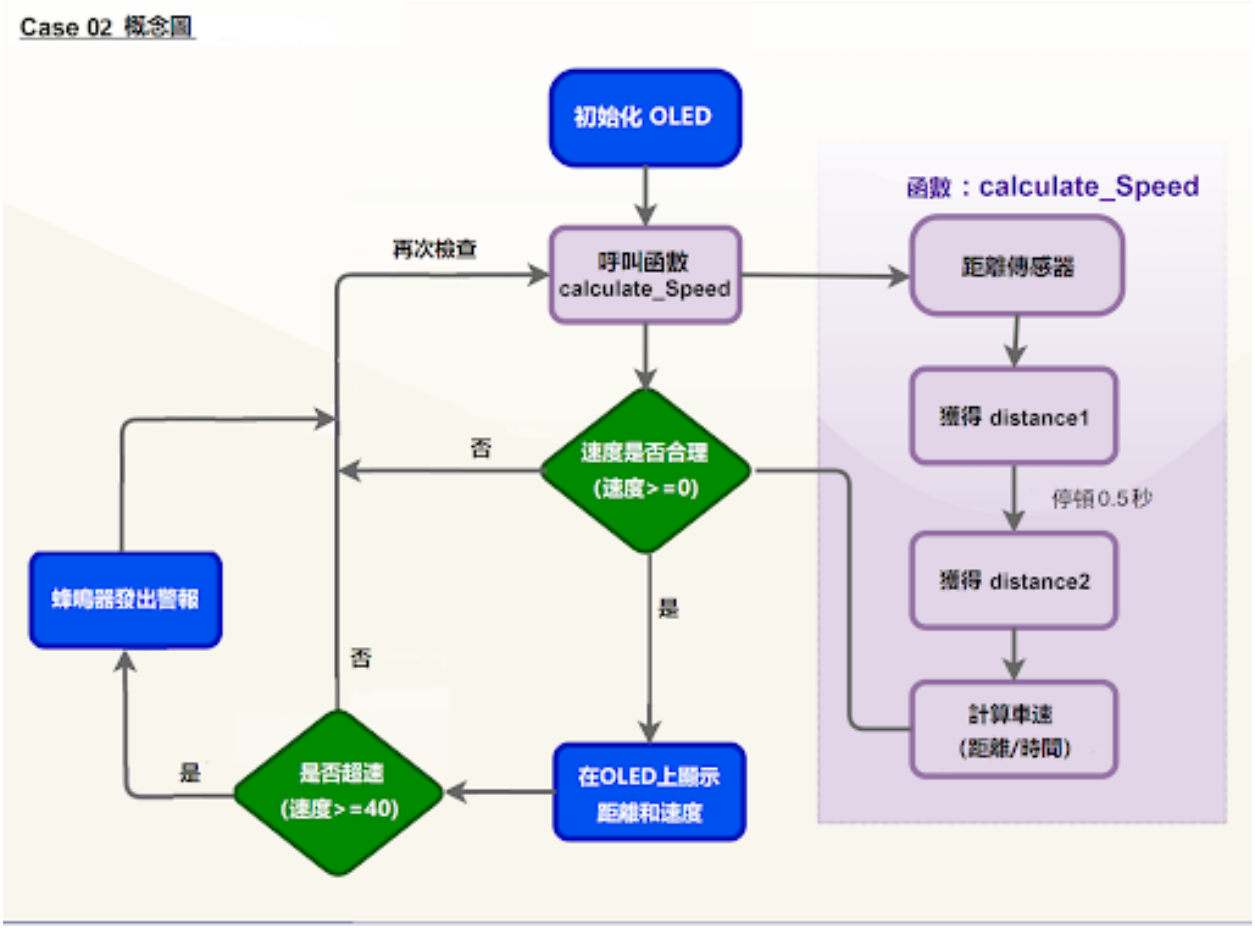
什麼是車速檢測

車速檢測運作

距離1：初始距離

距離2：0.5秒的距離

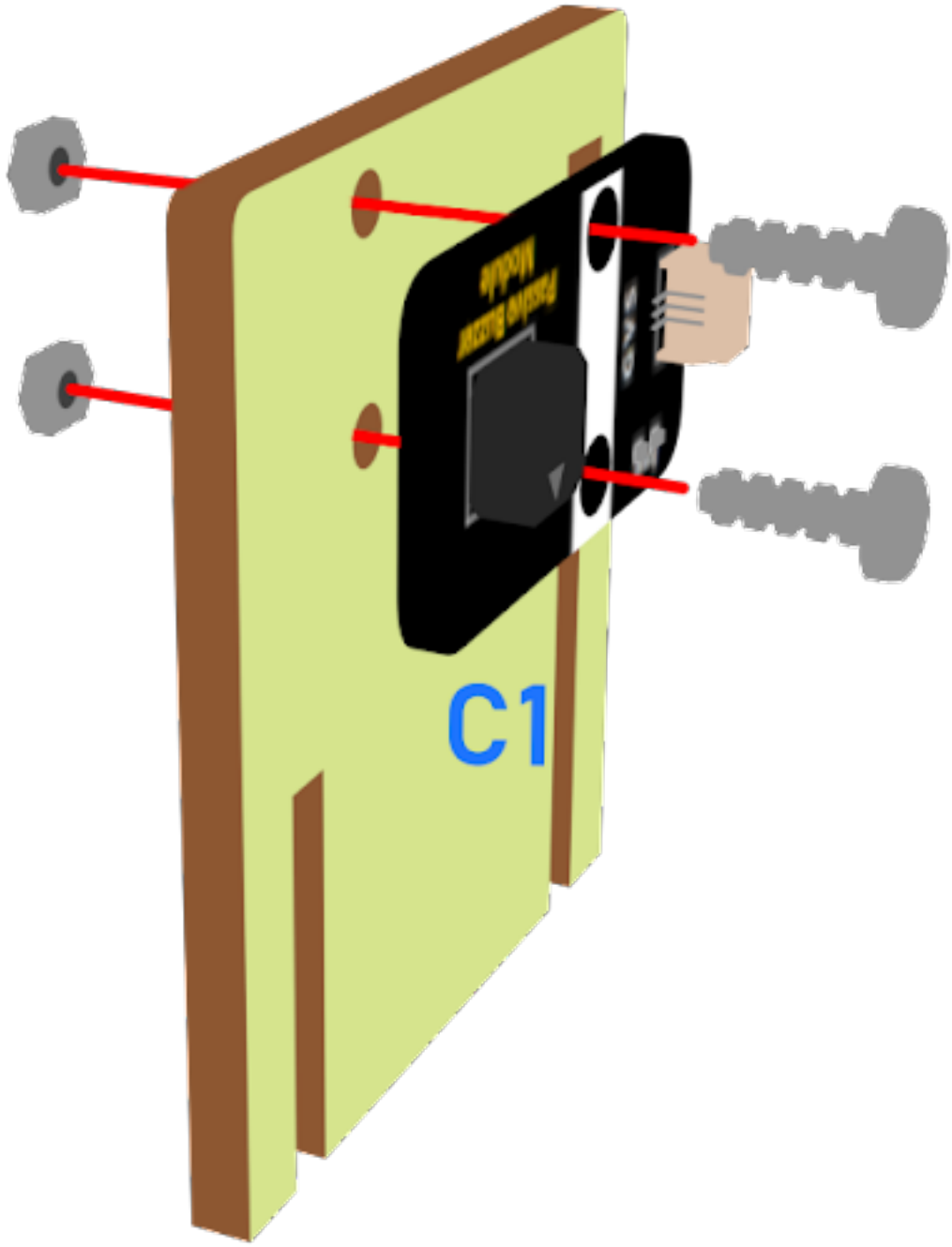
根據公式，速度 = 距離 / 時間，所以以上速度為 = (距離1 - 距離2) / 時間


Case 02 概念圖
初始化 OLED
函數：calculate_Speed
再次檢查
呼叫函數
calculate_Speed
距離傳感器
獲得 distance1
否
速度是否合理
(速度>=0)
停頓0.5秒
獲得 distance2
蜂鳴器發出警報
是
否
計算車速
(距離/時間)
是
是否超速
(速度>=40)
在OLED上顯示
距離和速度

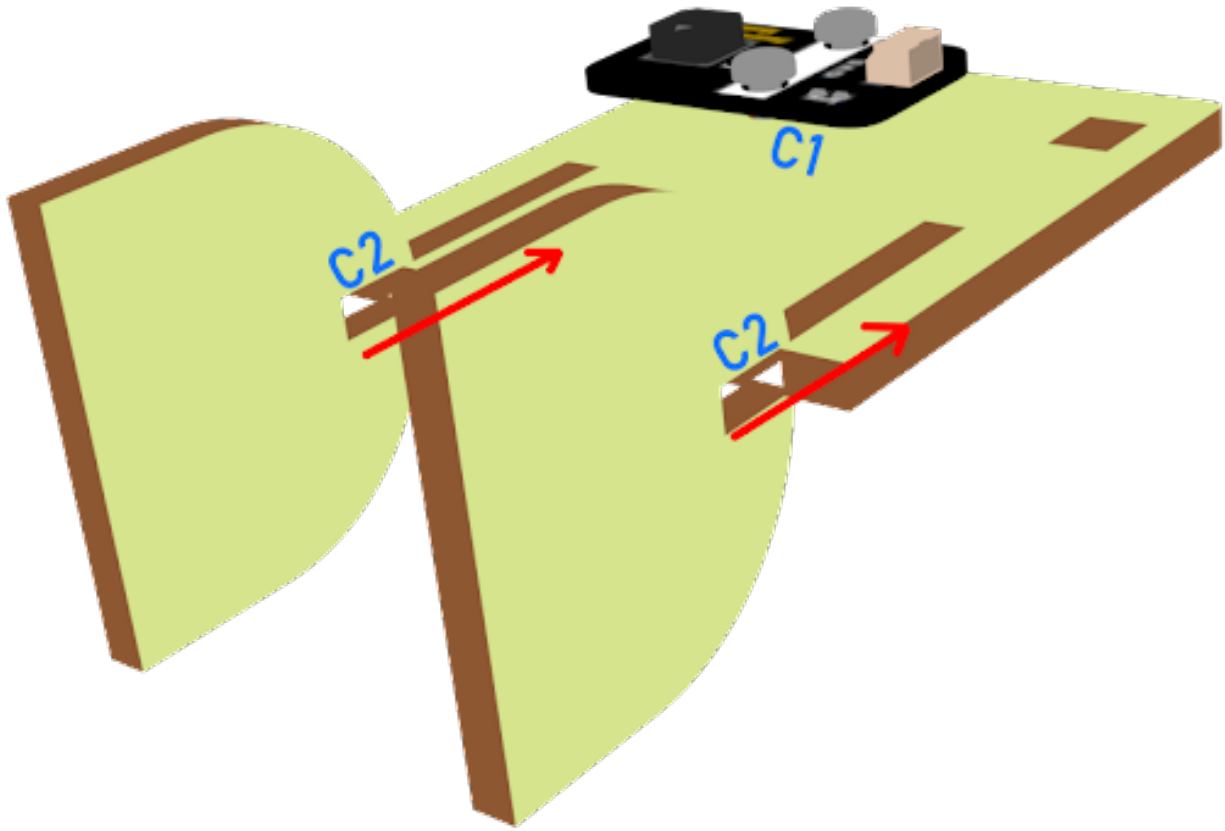
材料準備

組裝步驟

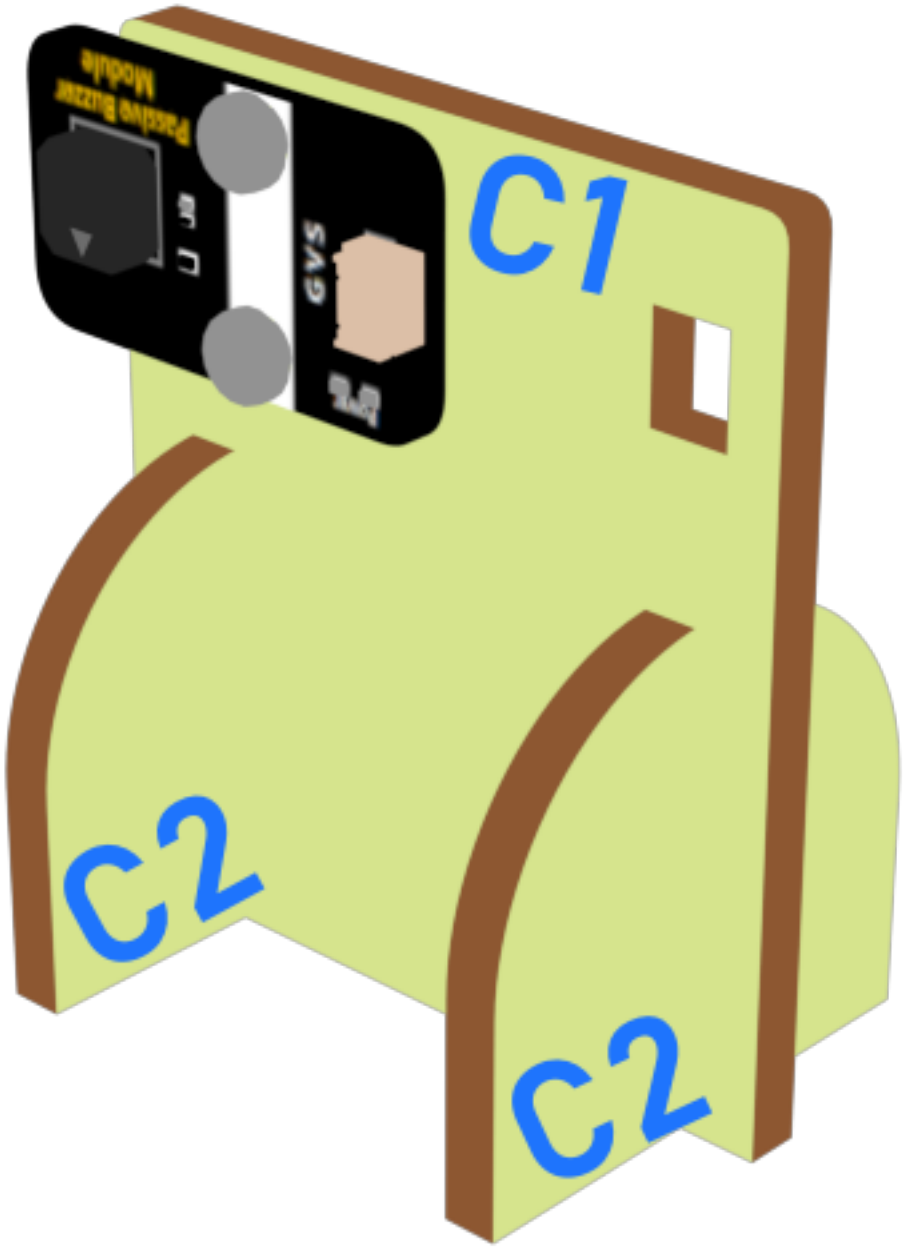
第一步



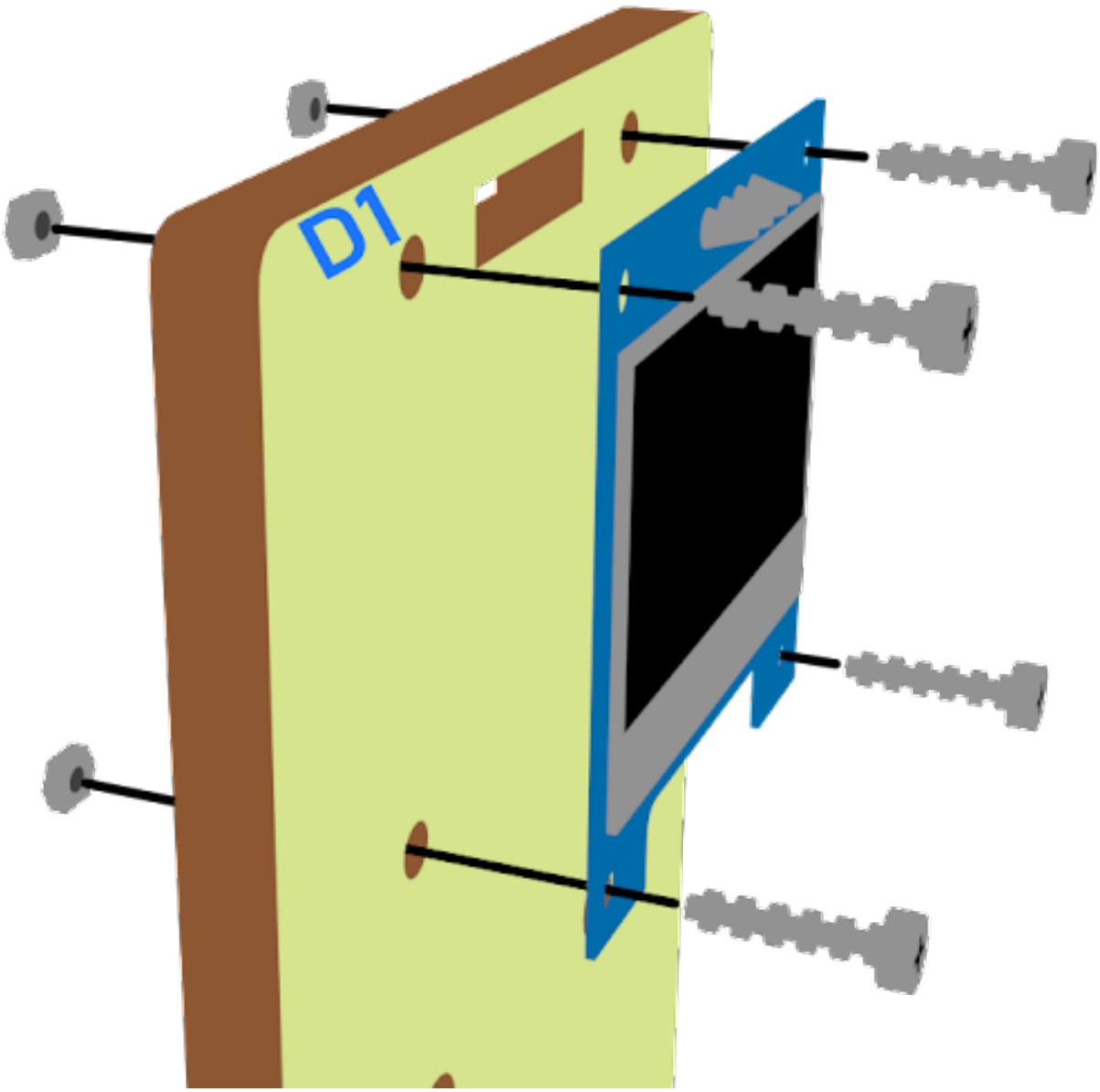
第二步



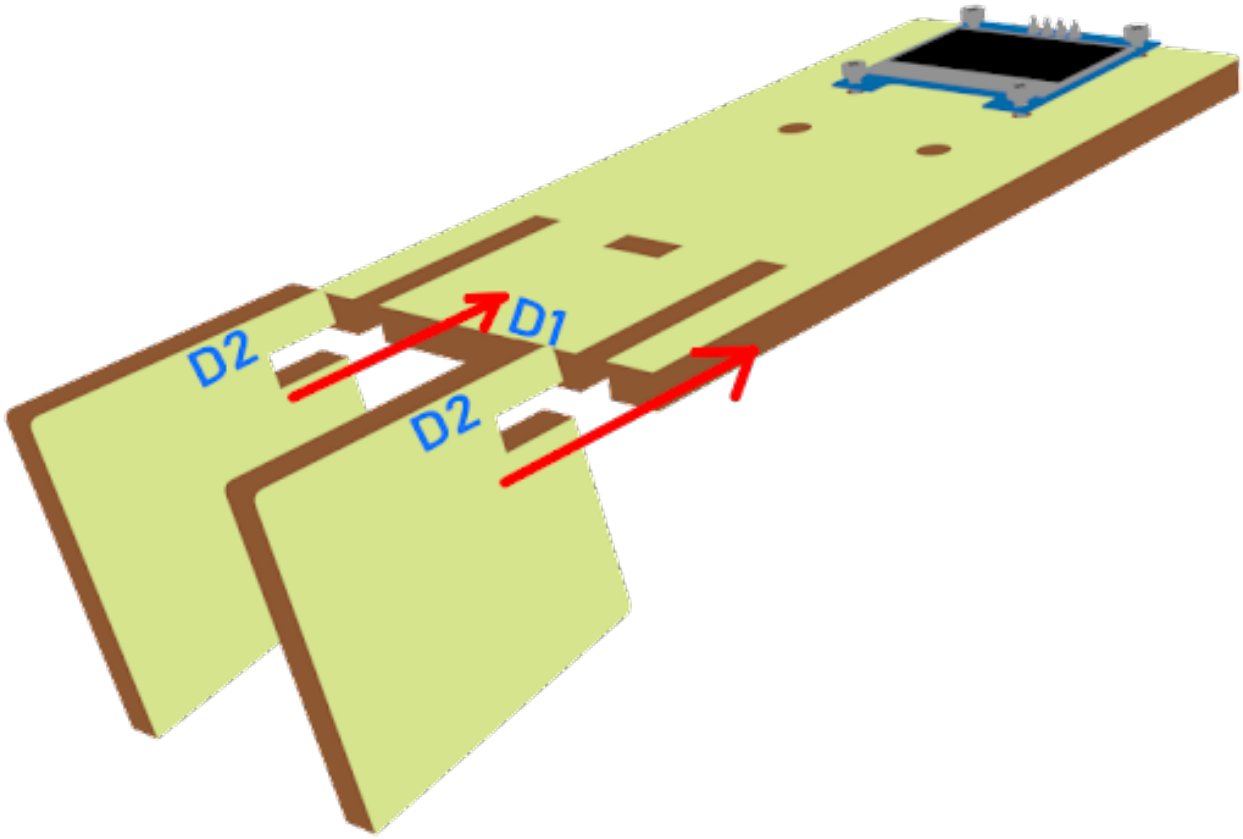
第三步



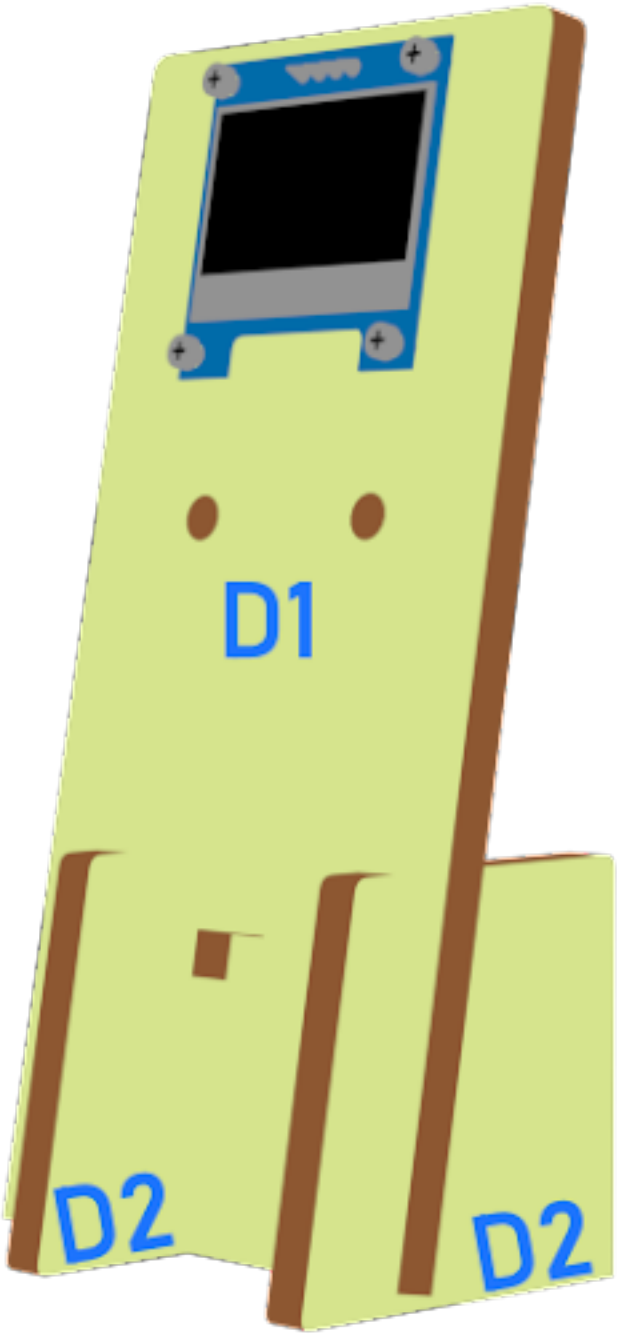
第四步



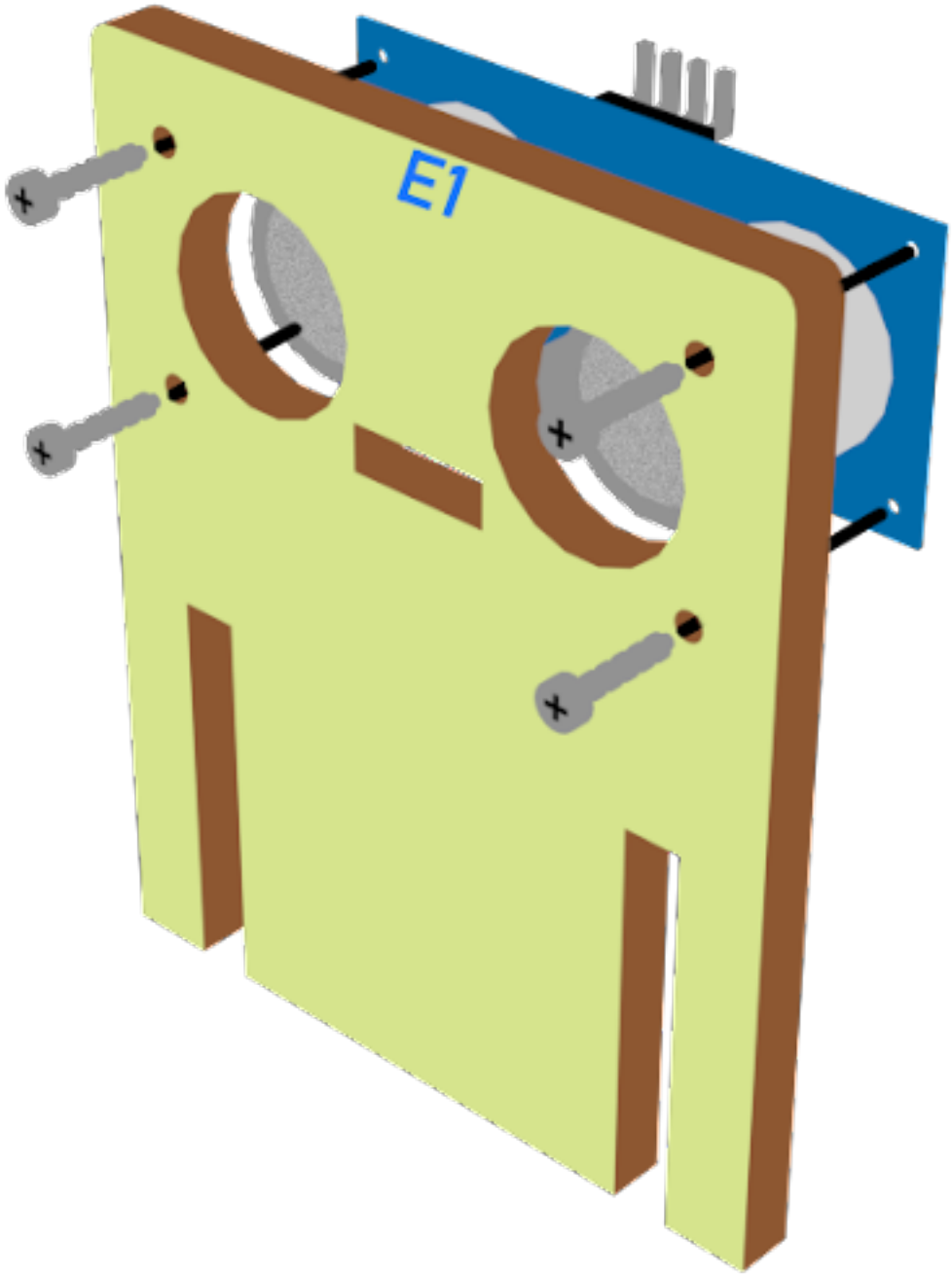
第五步



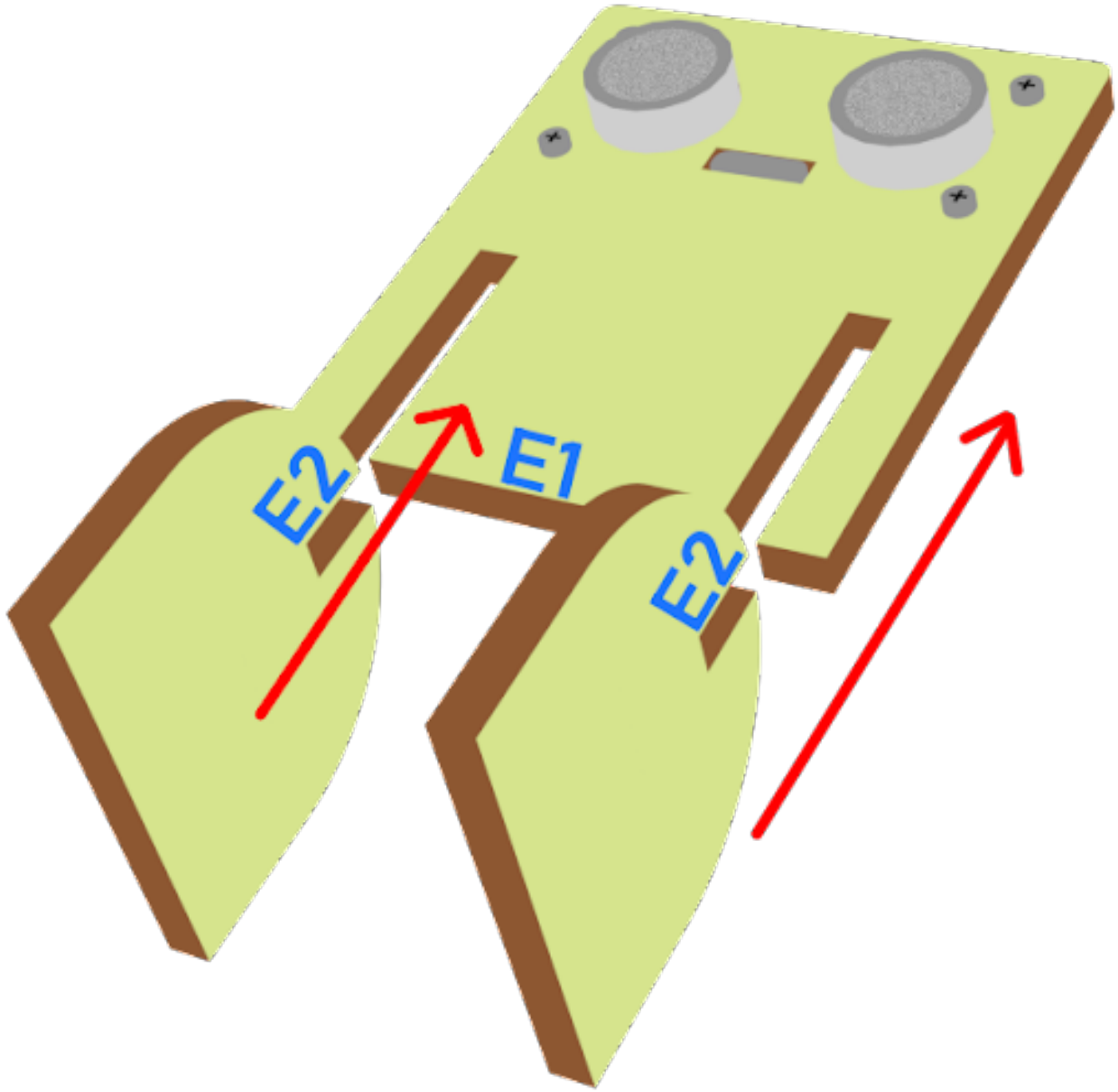
第六步



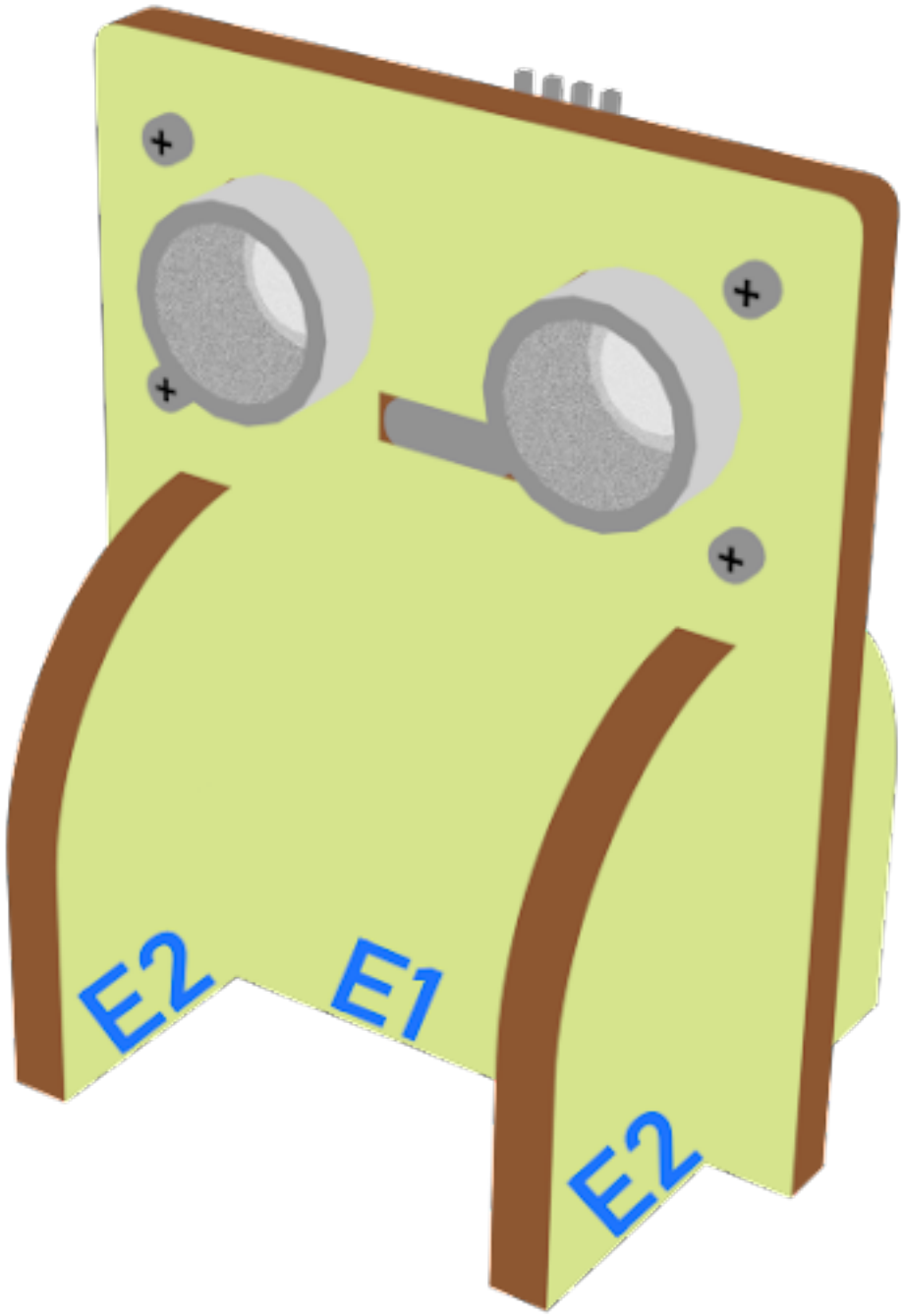
第七步



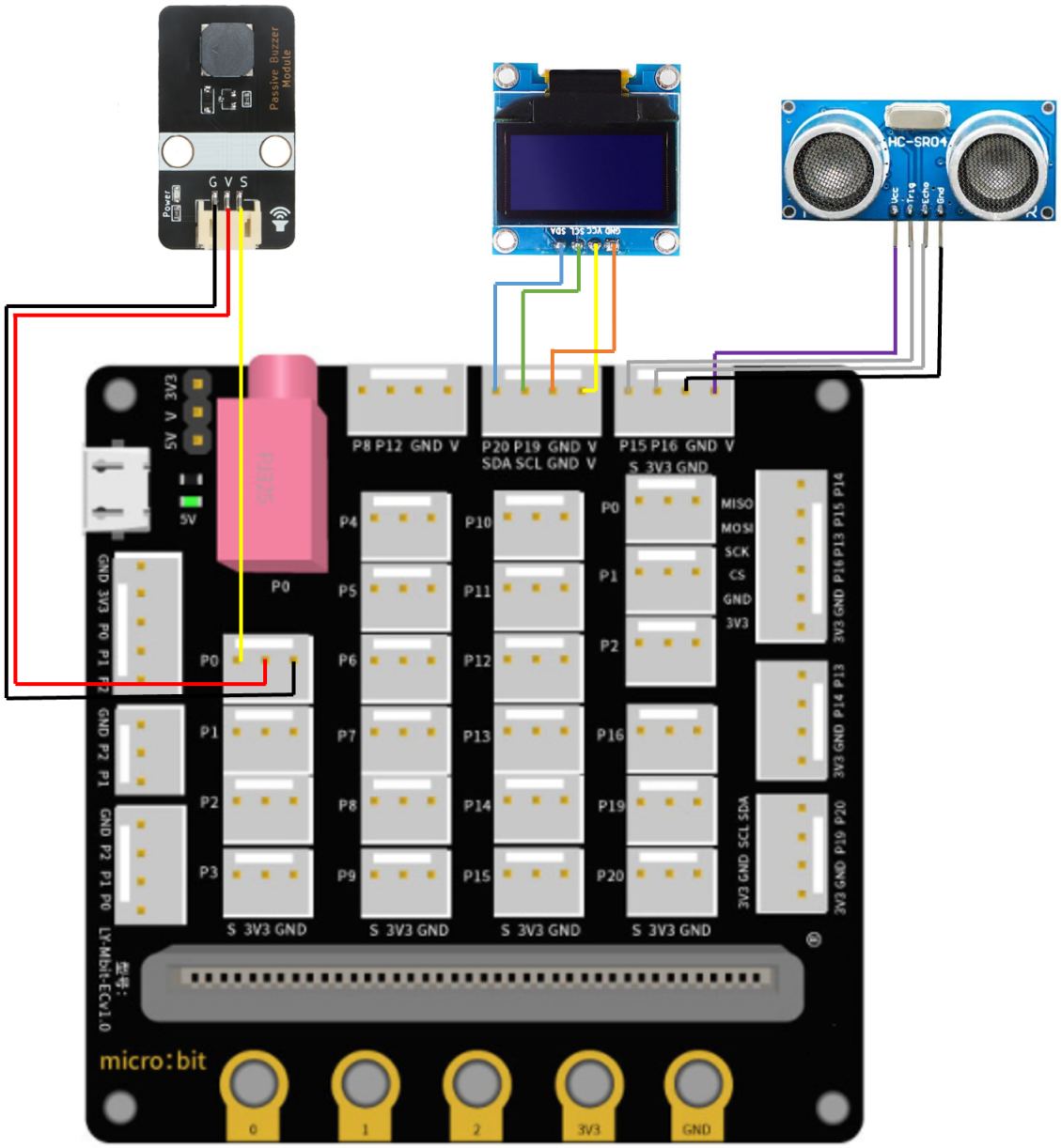
第八步

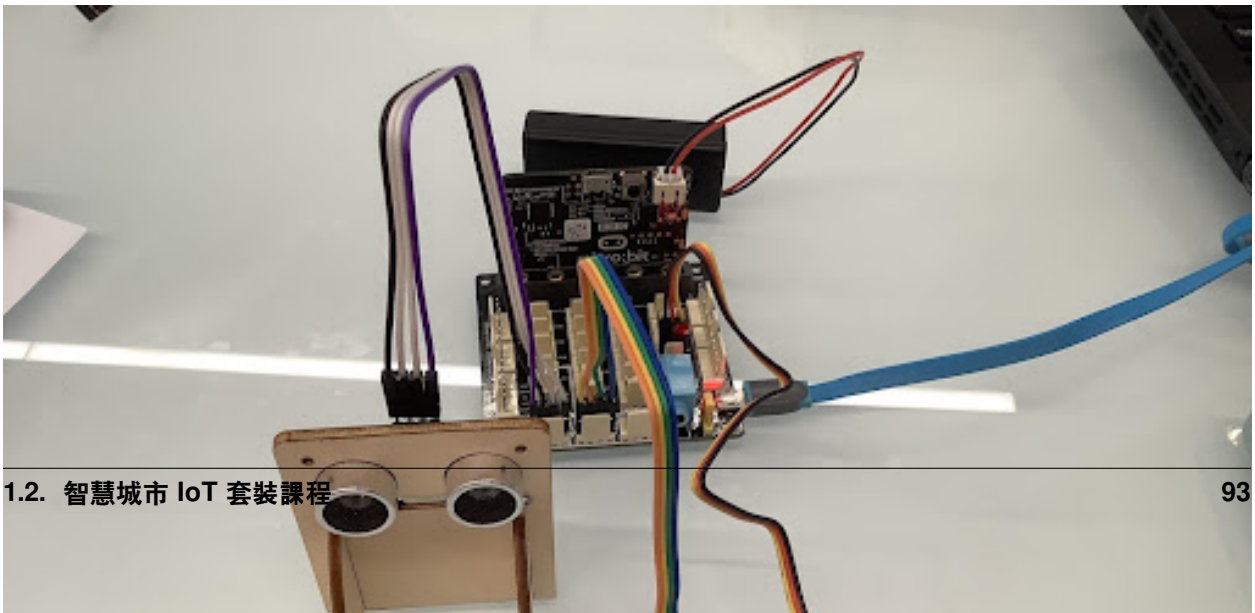


第九步



硬件連接

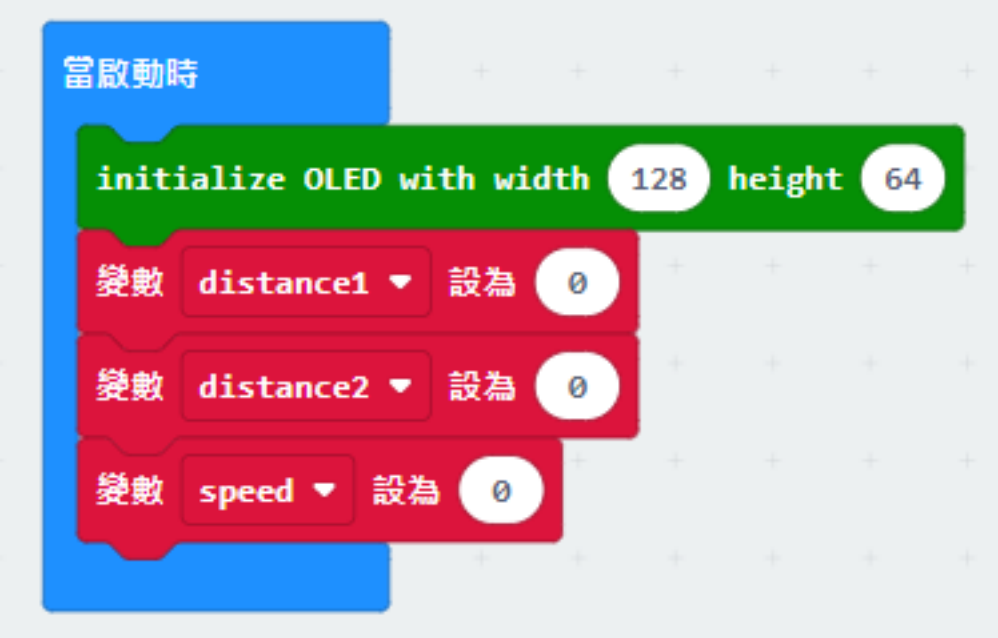

Passive Buzzer Module
G V S
HC-SR04
micro:bit
P8 P12 GND V
P20 P19 GND V
SDA SCL GND V
P15 P16 GND V
S 3V3 GND
MISO
MOSI
SCK
CS
GND
3V3
0
1
2
3V3
GND



編程

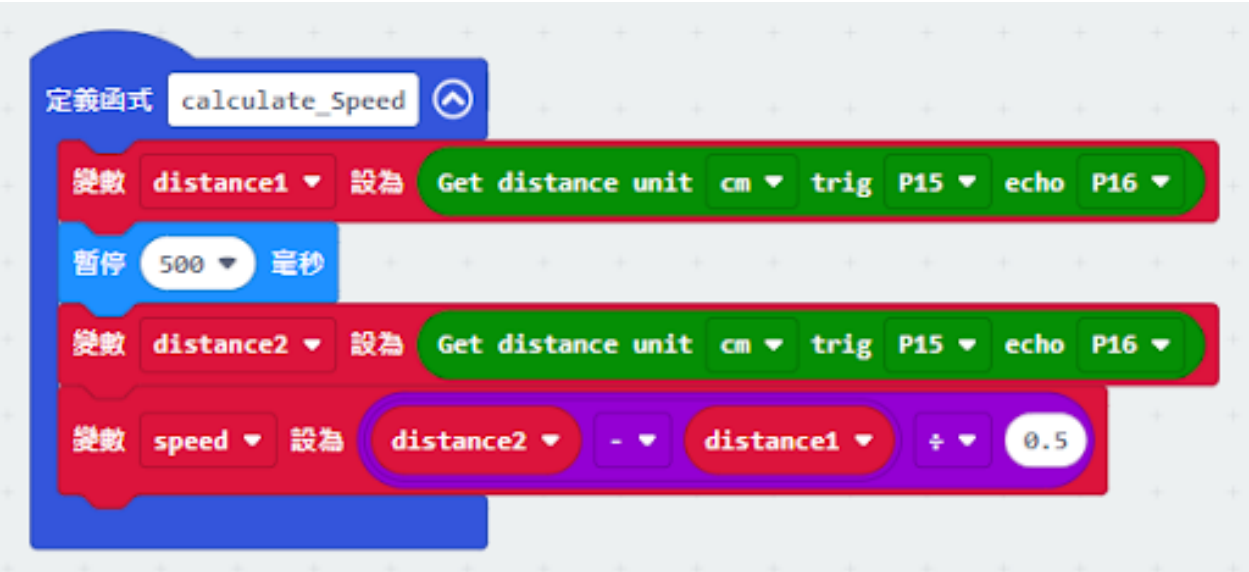
初始化 OLED 屏幕

- 拖動 OLED 初始化寬：128 高:64，至當Ⓕ動時。
- 從變數中將 distance1、distance2 和 speed 設Ⓕ 0。



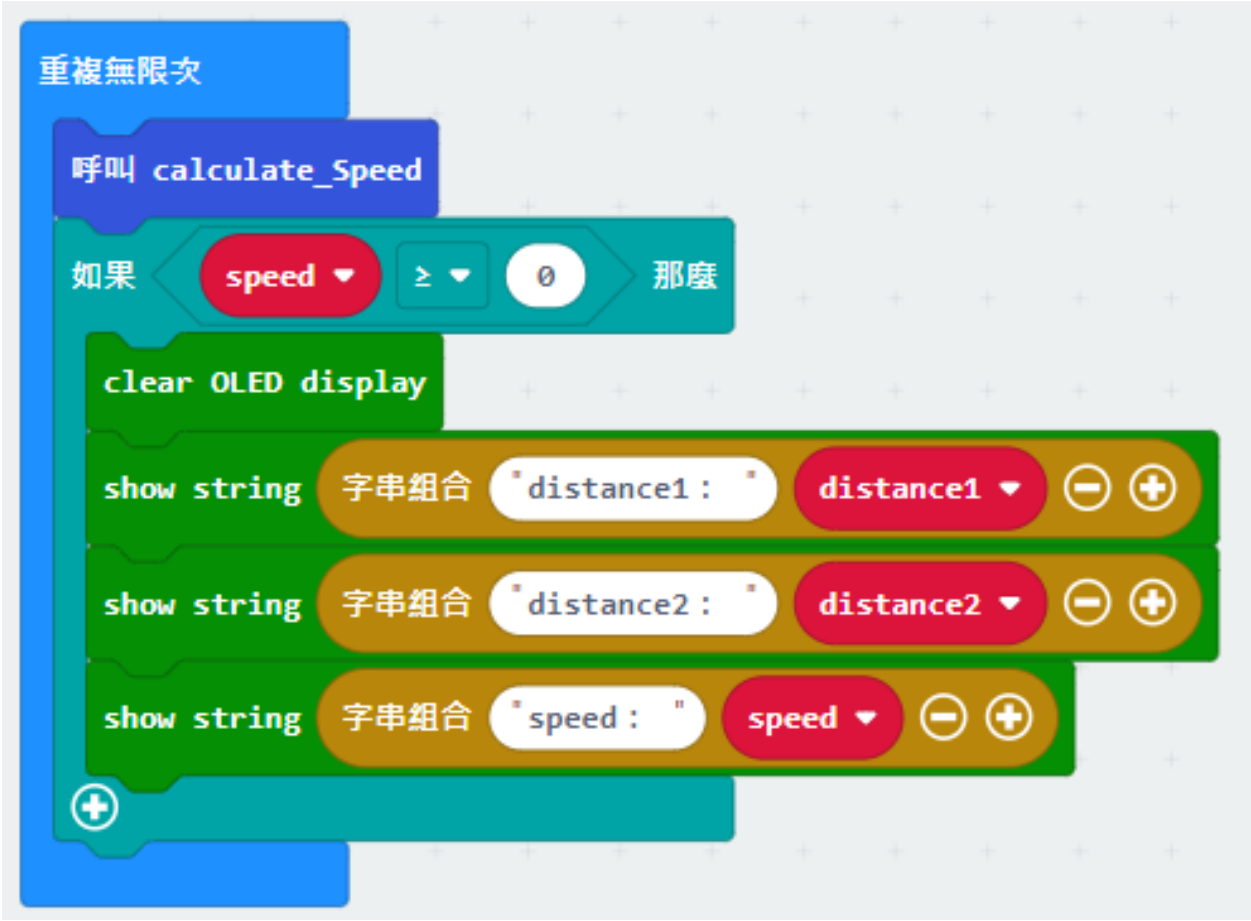
設定函數（計算速度）

- 從高級 > 功能中定義函式 calculate_Speed。
- 將 distance1 設爲 get distance unit cm trig P14 echo P15（0.5 秒前從汽車到距離傳感器的距離）拖動 Pause 等待 500ms，將 distance2 設爲 get distance unit cm trig P14 echo P15 （0.5 秒後從汽車到距離傳感器的距離）。
- 根據速度 = 距離/時間的方程式，我們得到移動的汽車的速度Ⓕ（distance1-distance2）/0.5（單位：cm/s）。



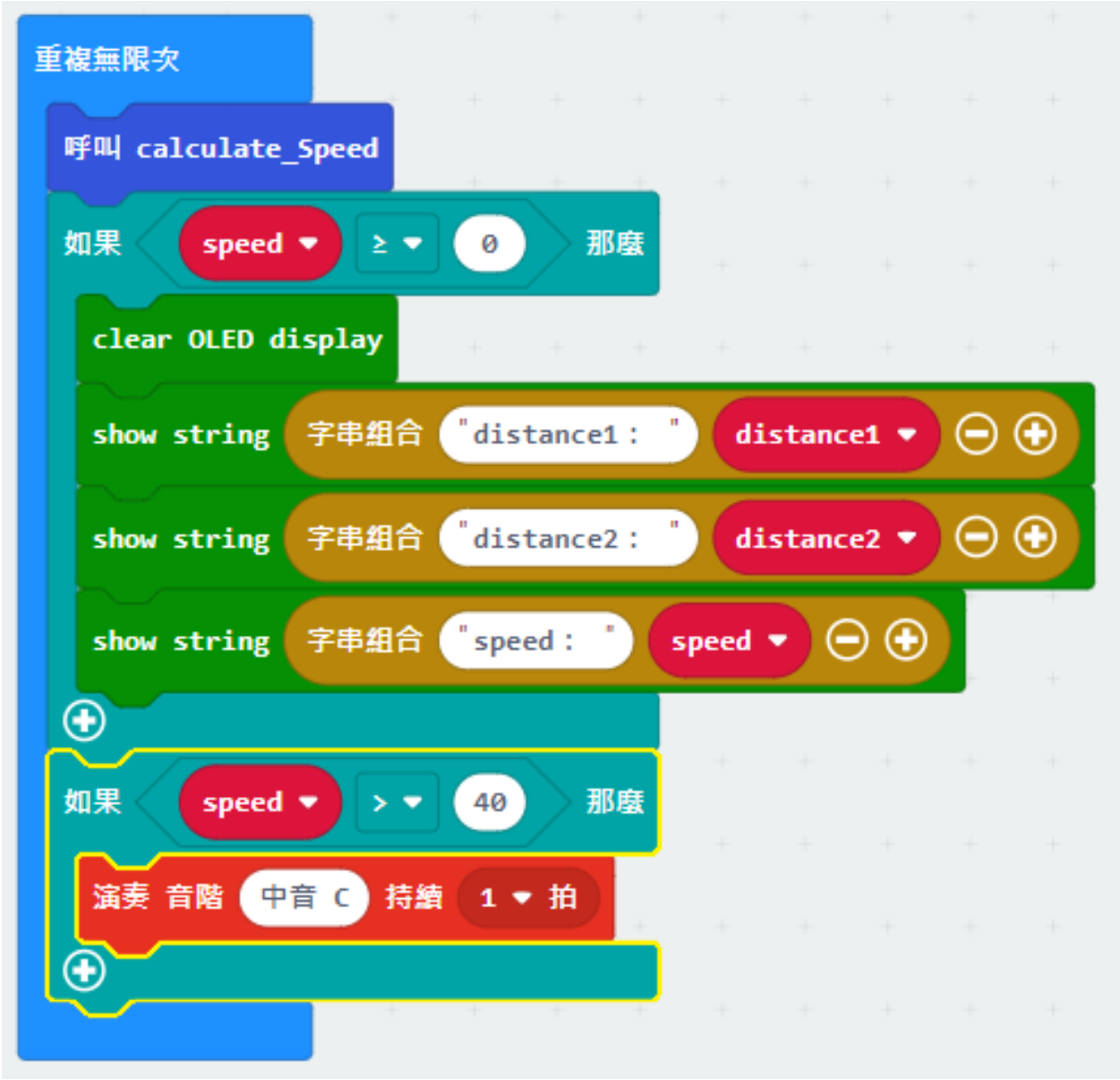
計算車速

- 在重F無限次塊中，從高級 > 函數中呼叫函式 calculate_Speed，以獲得移動的汽車的速度。
- 將如果語句放入循環中
- 從 OLED 中拖出清除 OLED 顯示，以避免重F。
- 拖出新行顯示字符串F顯示變數 distance1、distance2 和 speed 的值。



蜂鳴器

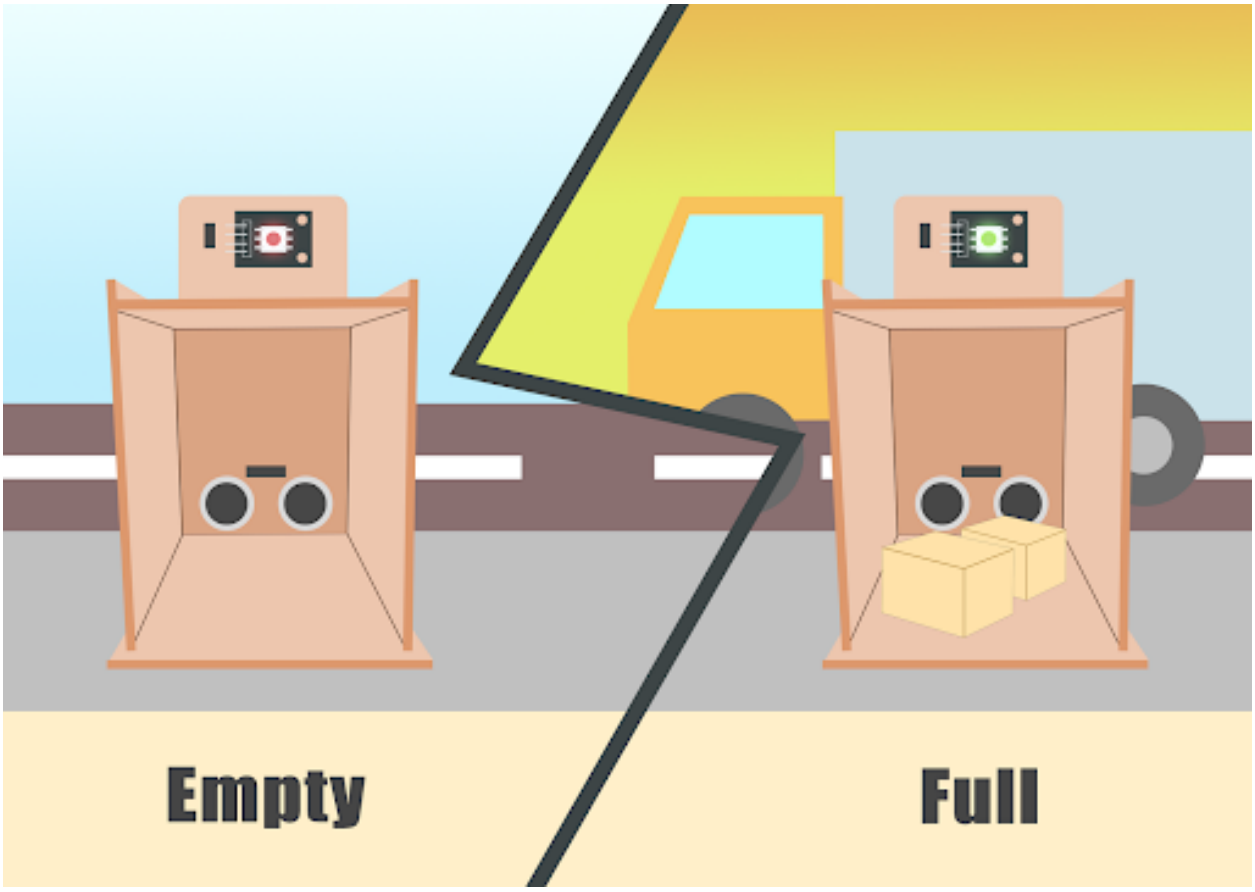
- 將如果語句放入循環中
- 如果 speed≥40，那Ⓕ演奏音階中音 C 持續一拍


重複無限次
呼叫 calculate_Speed
如果 speed ≥ 0 那麼
clear OLED display
show string 字串組合 "distance1 : " distance1
show string 字串組合 "distance2 : " distance2
show string 字串組合 "speed : " speed
如果 speed > 40 那麼
演奏 音階 中音 C 持續 1 拍

總結

Think 思考

1.2.4 卸貨提醒系統

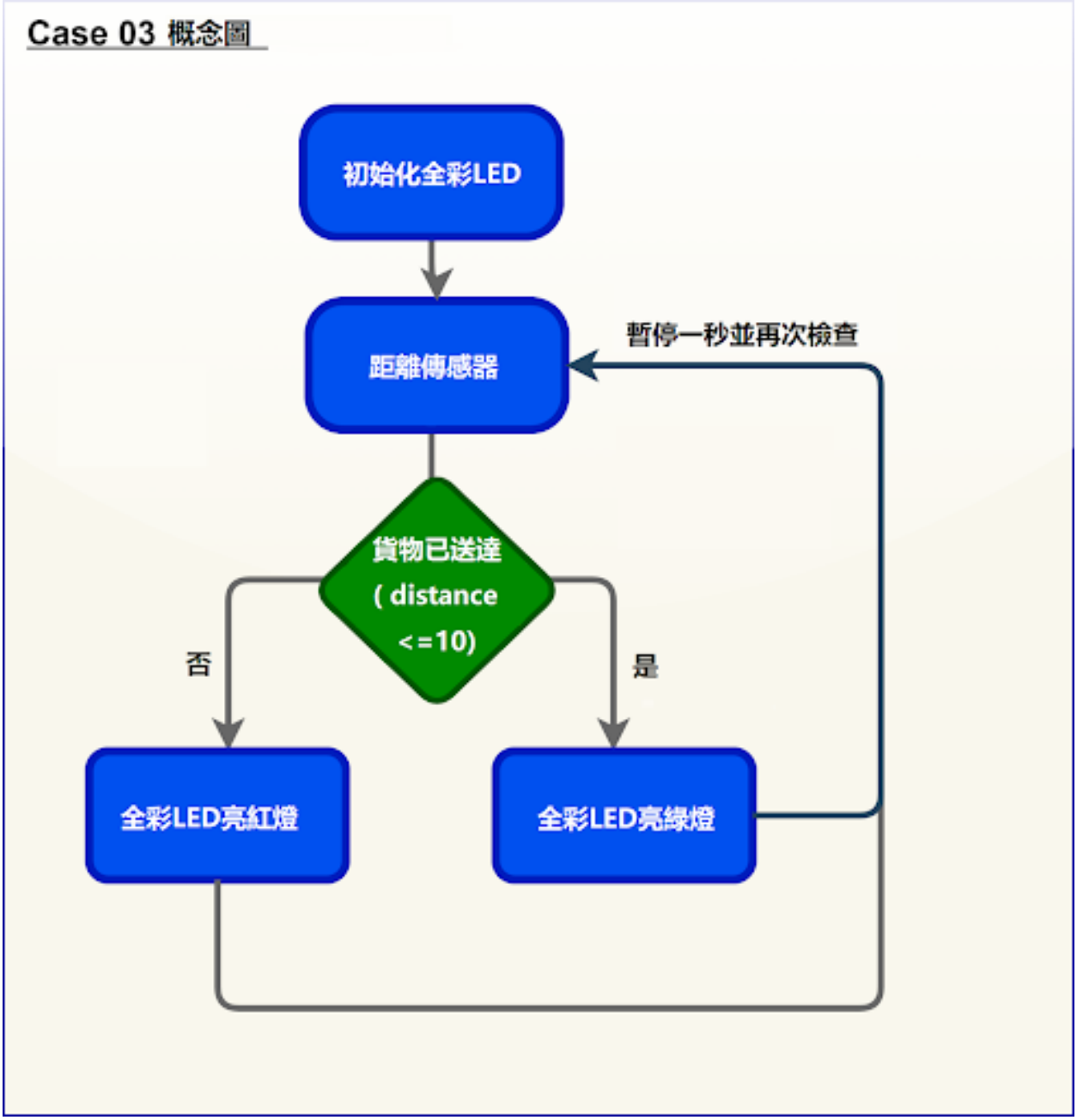


目標

背景

什麼是卸貨提醒系統？

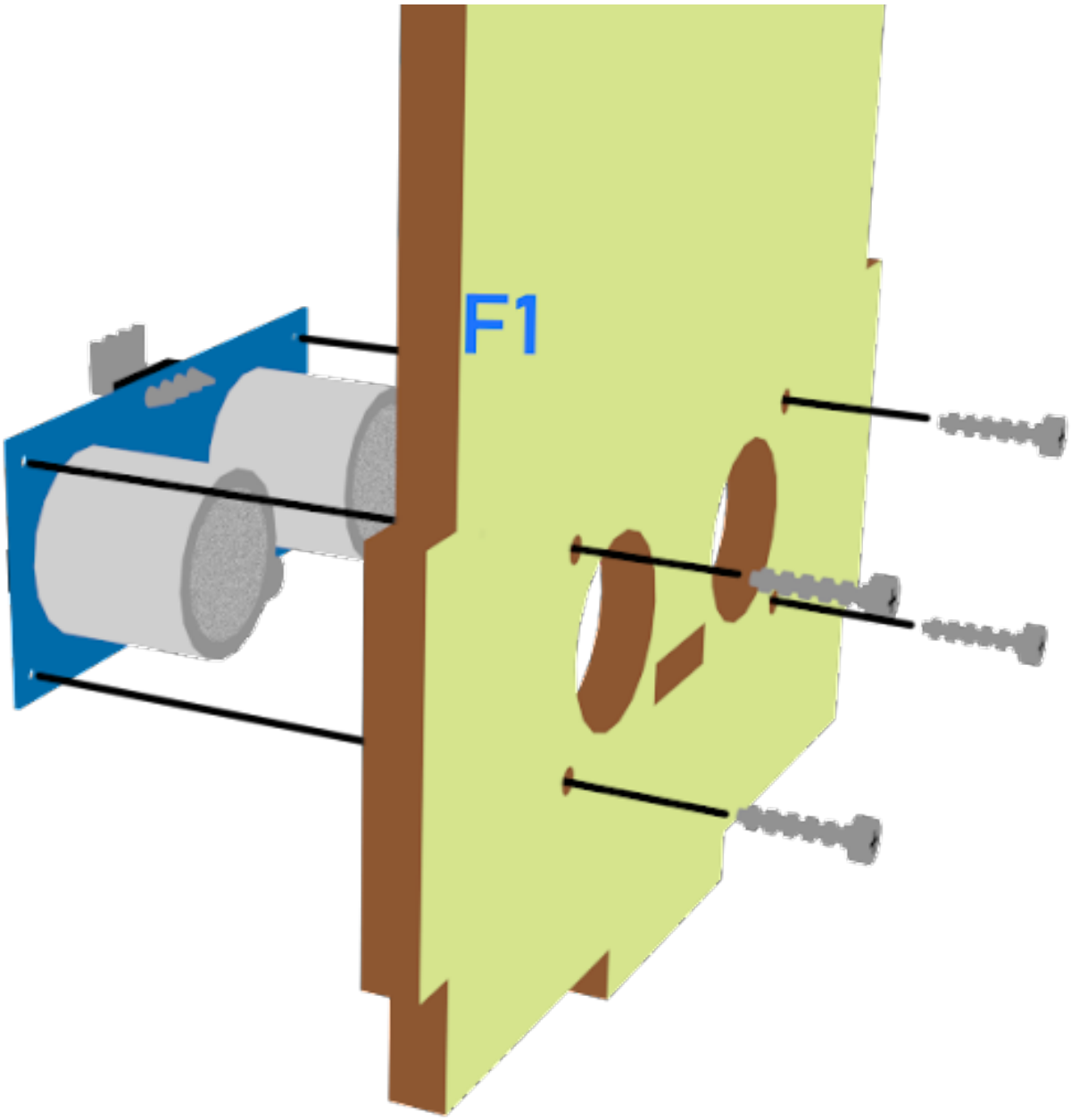
卸貨提醒系統運作


Case 03 概念圖
初始化全彩LED
距離傳感器
暫停一秒並再次檢查
貨物已送達
(distance
<=10)
否
是
全彩LED亮紅燈
全彩LED亮綠燈

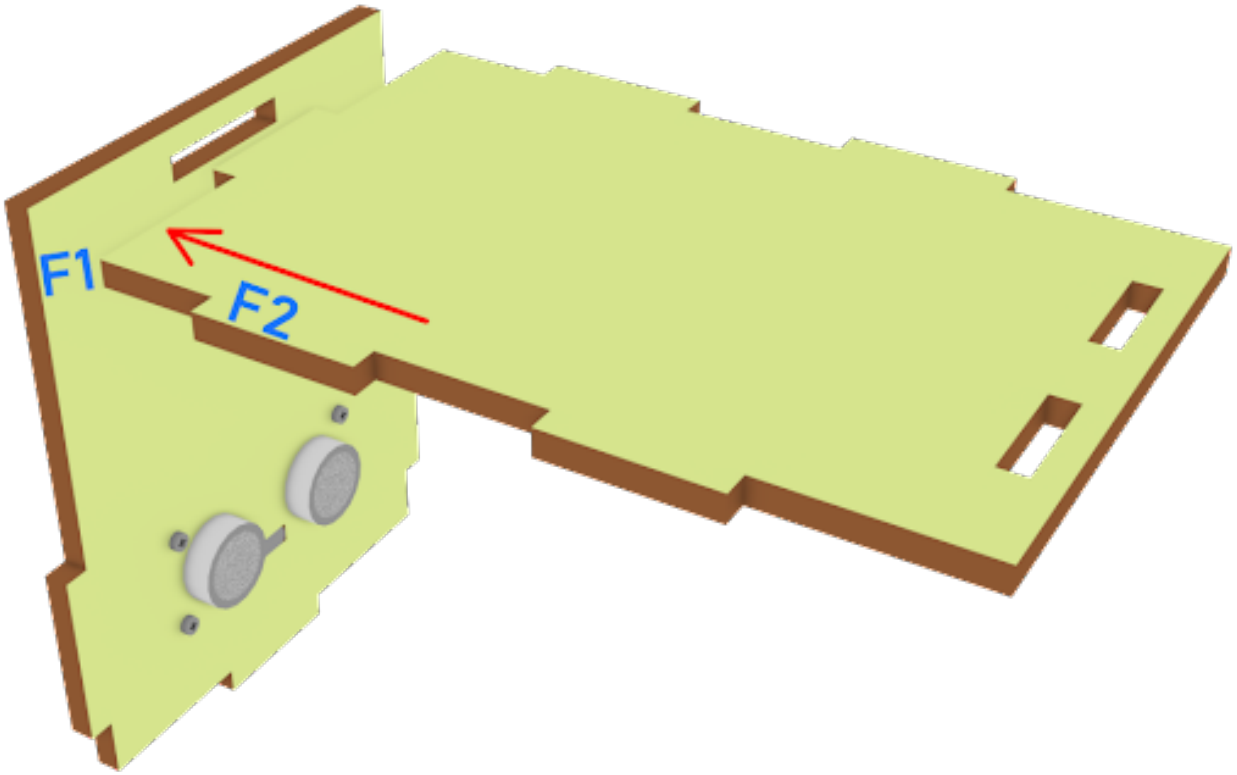
材料準備

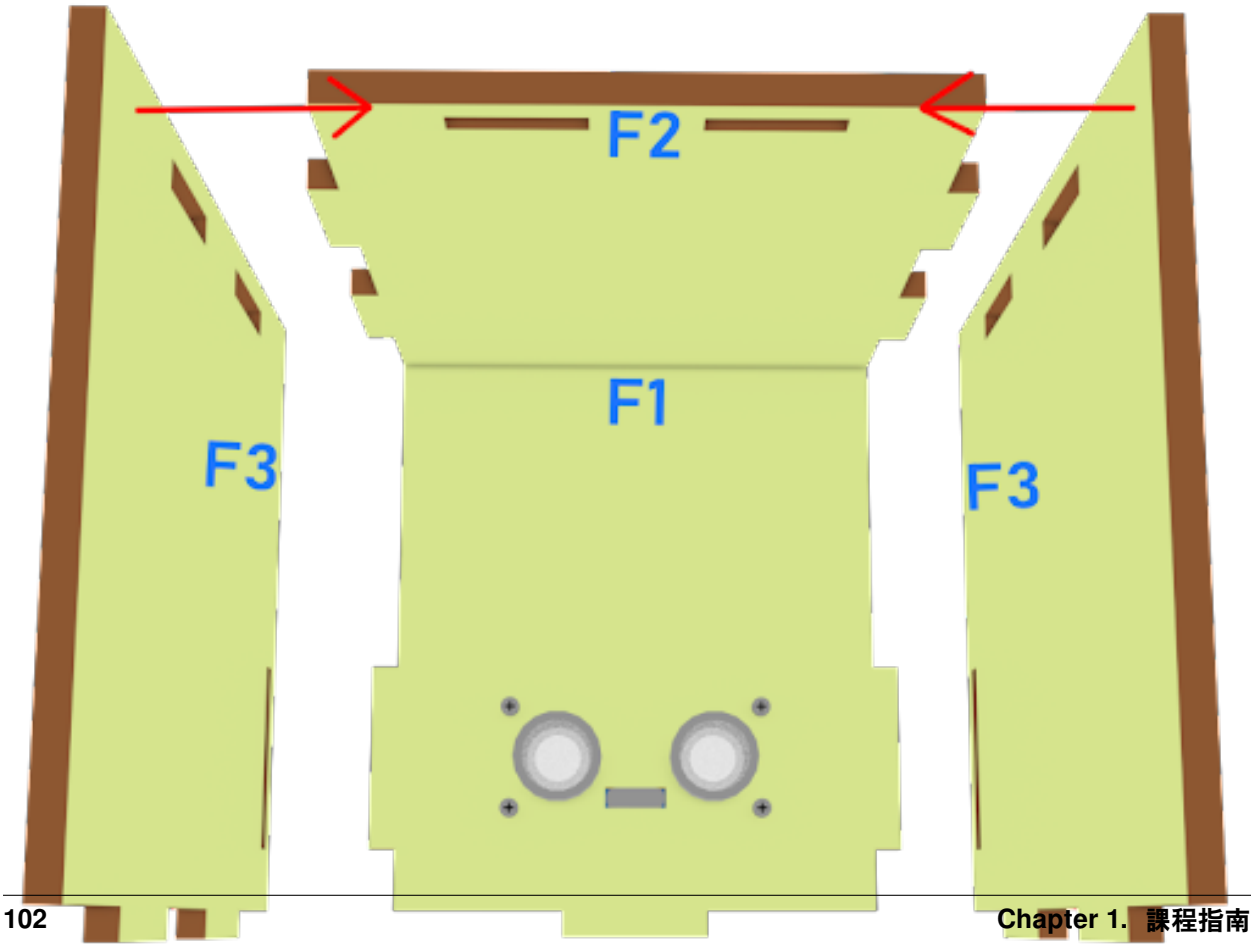
組裝步驟

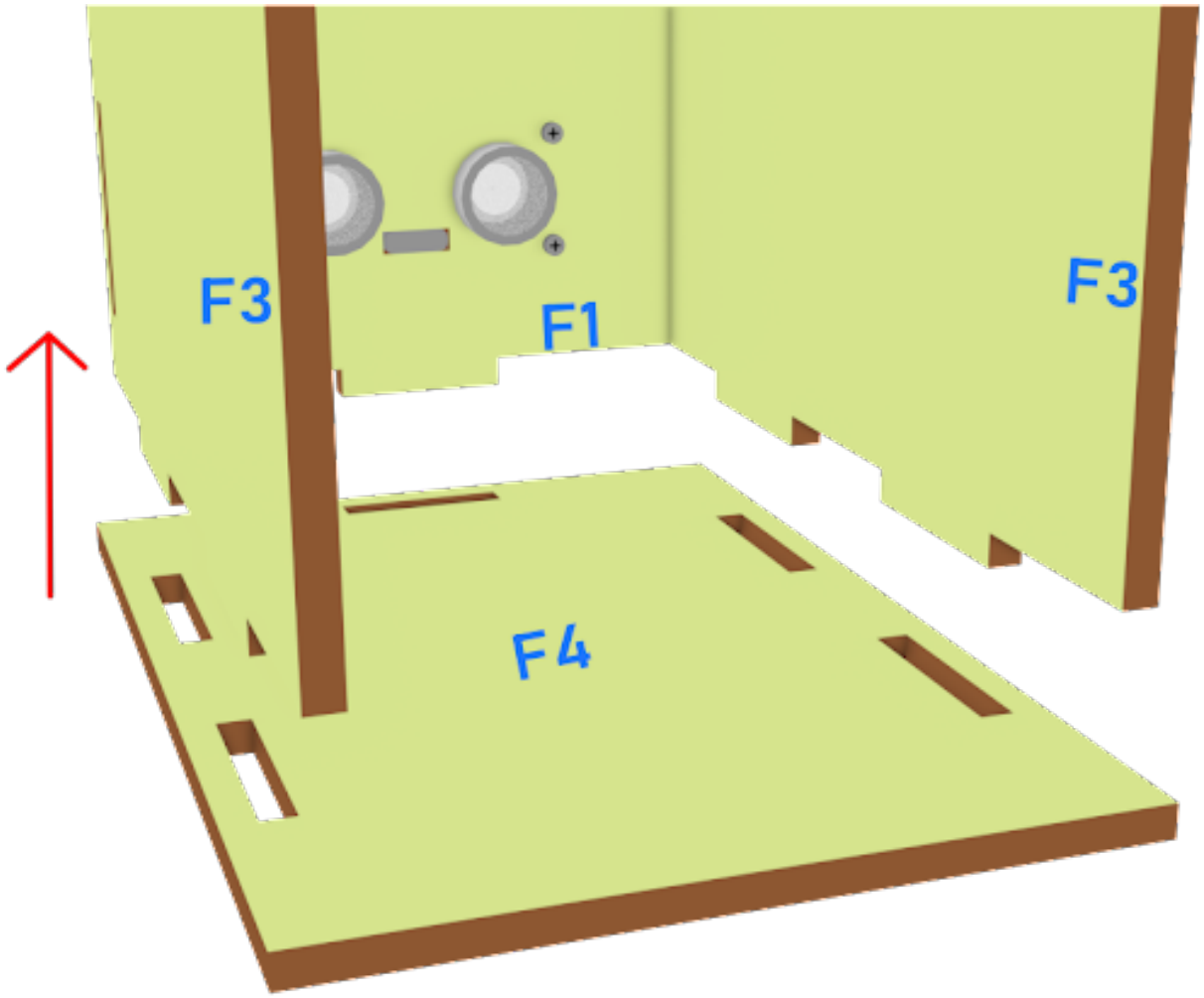
第一步



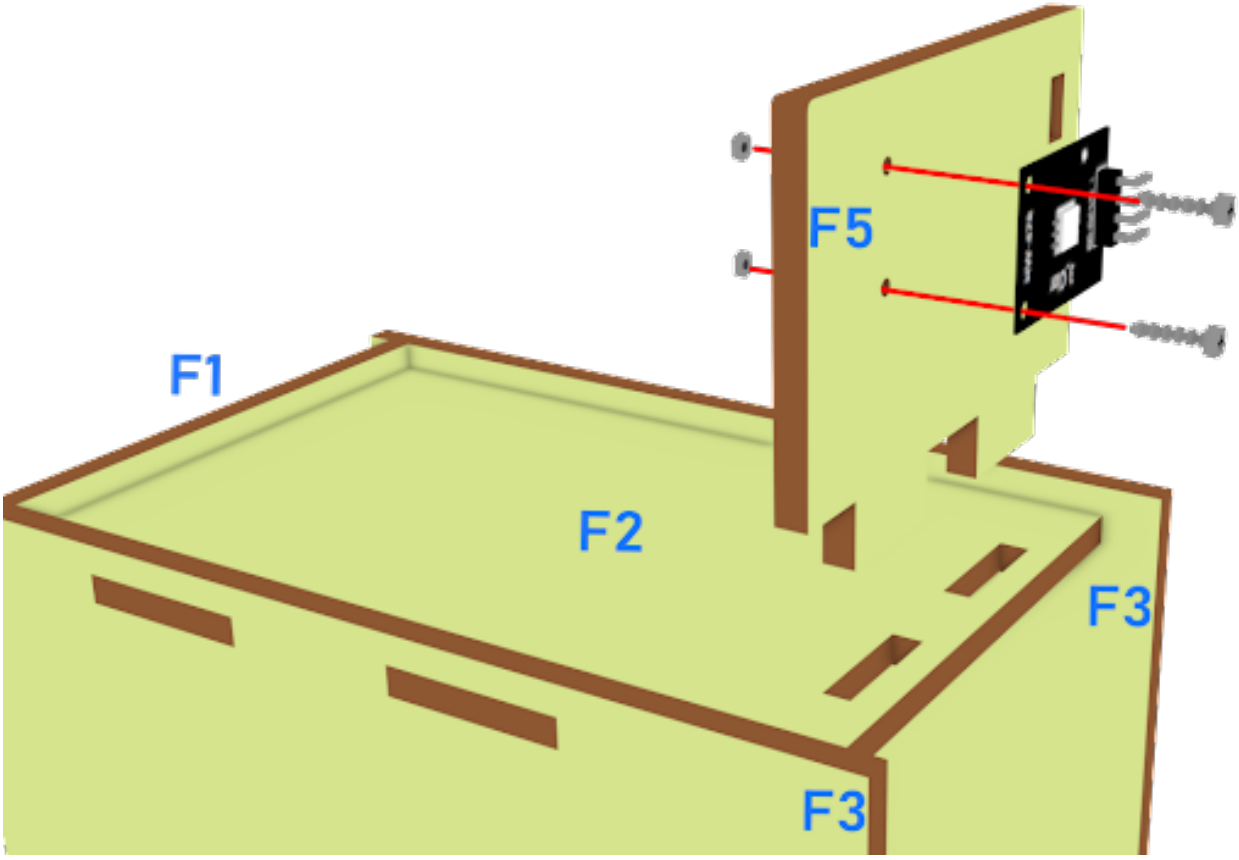
第二步



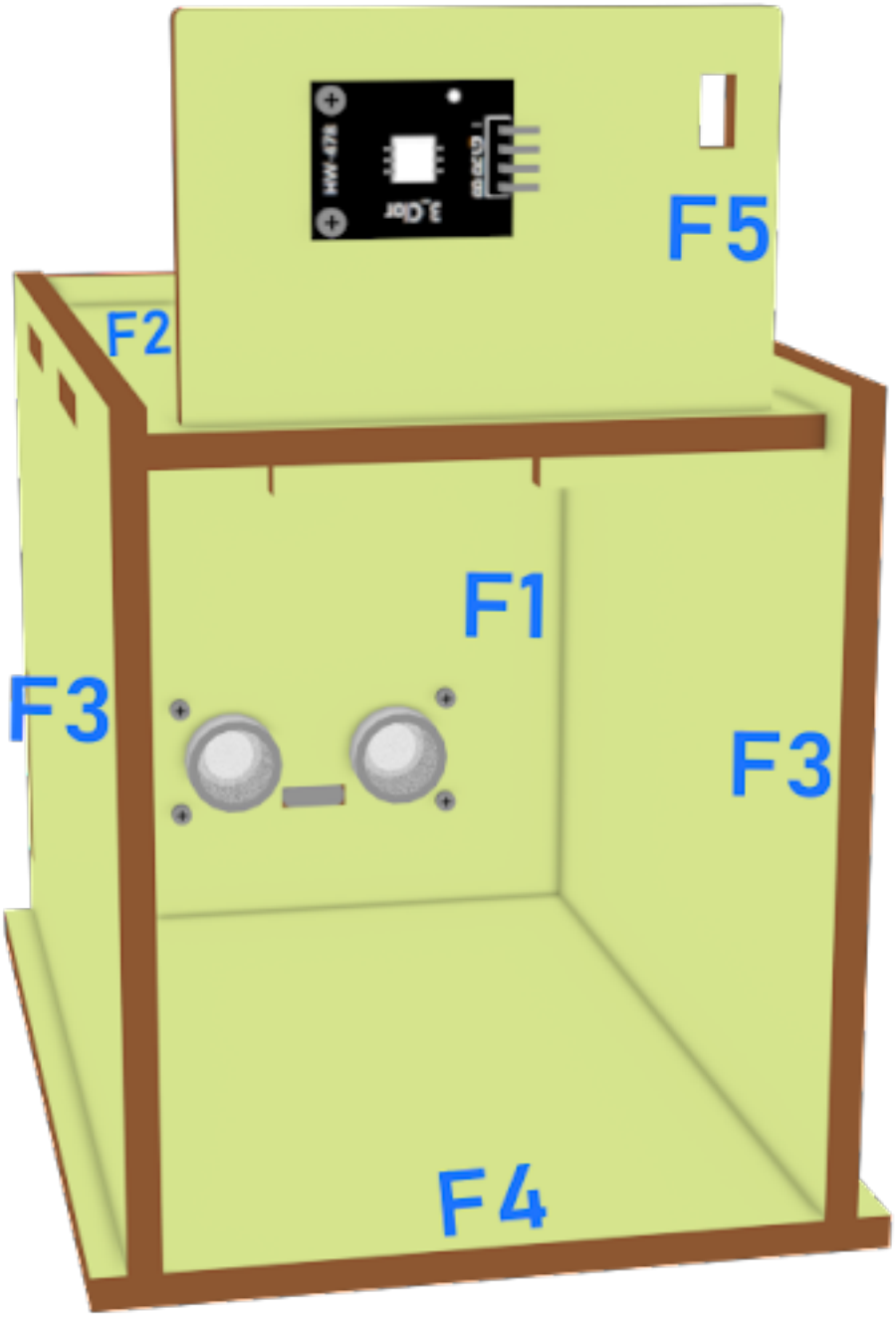



F3
F1
F3
F4

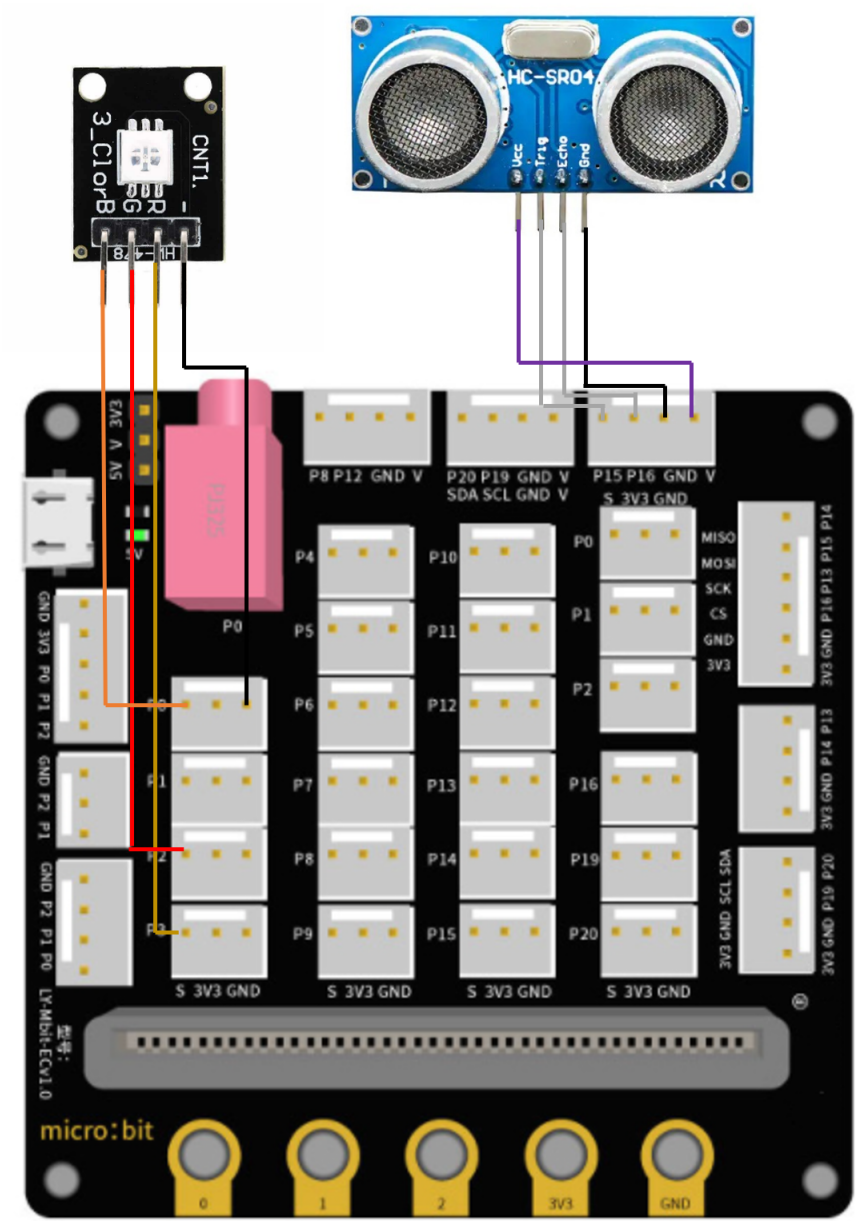
第三步

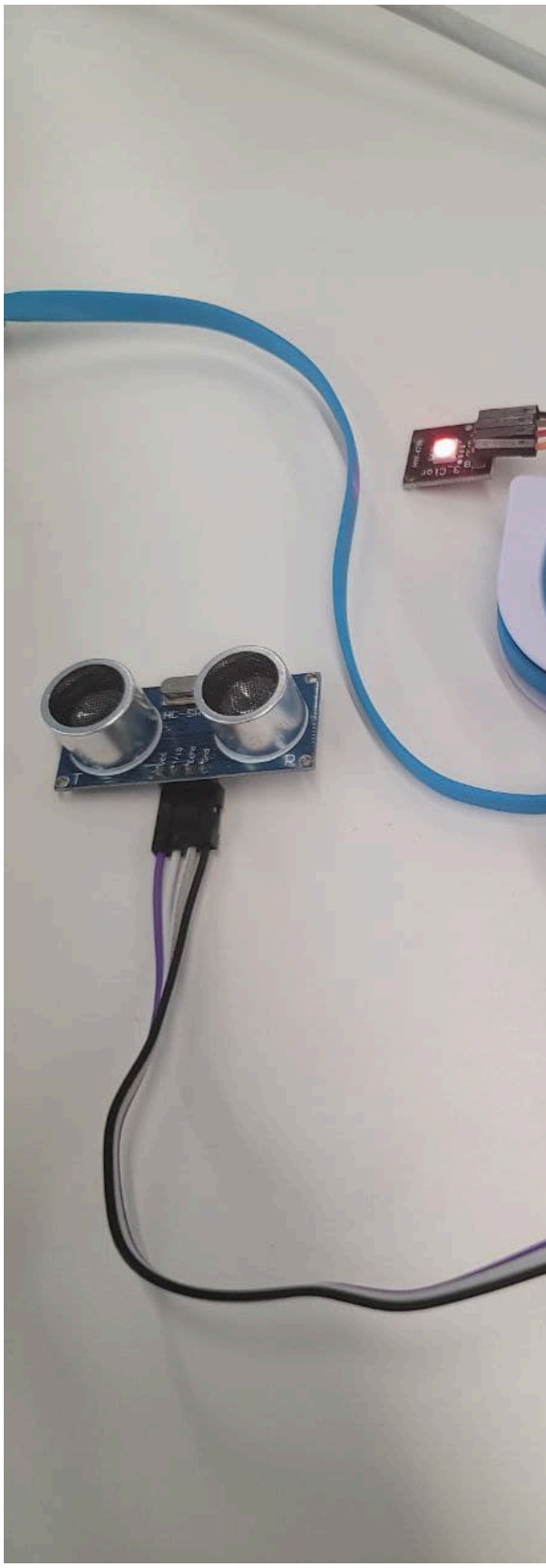


第四步



硬件連接

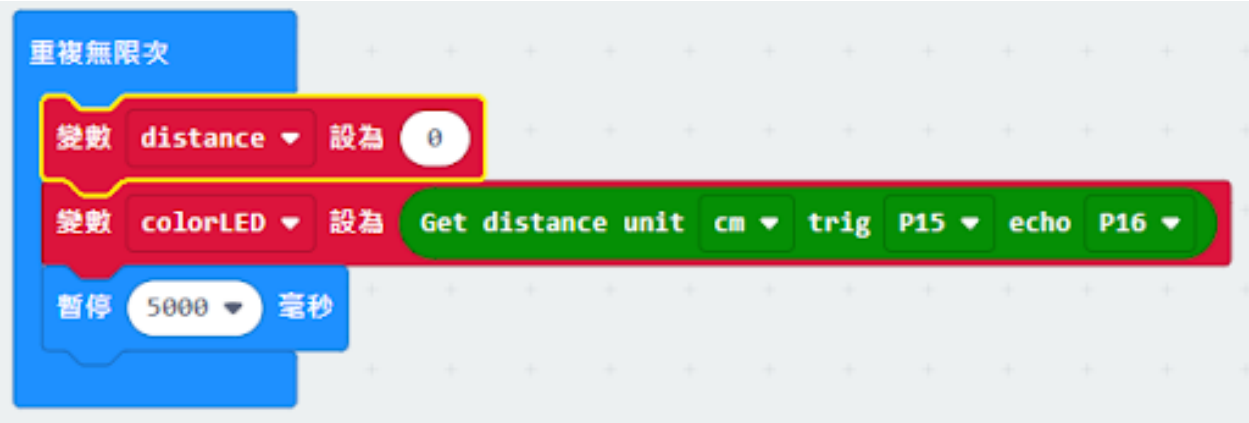

HC-SR04
Vcc
Trig
Echo
Gnd
CNT1.
3_ClorB
micro:bit
P8 P12 GND V
P20 P19 GND V
SDA SCL GND V
P15 P16 GND V
S 3V3 GND
MISO
MOSI
SCK
CS
GND
3V3
0
1
2
3V3
GND



編程

設置變量Ⓕ初始化多色 LED

- 在但啓動時裏，從變數拖入變數 distance 設Ⓕ 0
- 拖入變數 colorLED 設Ⓕ color pin 設定…。
- 拖出暫停 5 秒

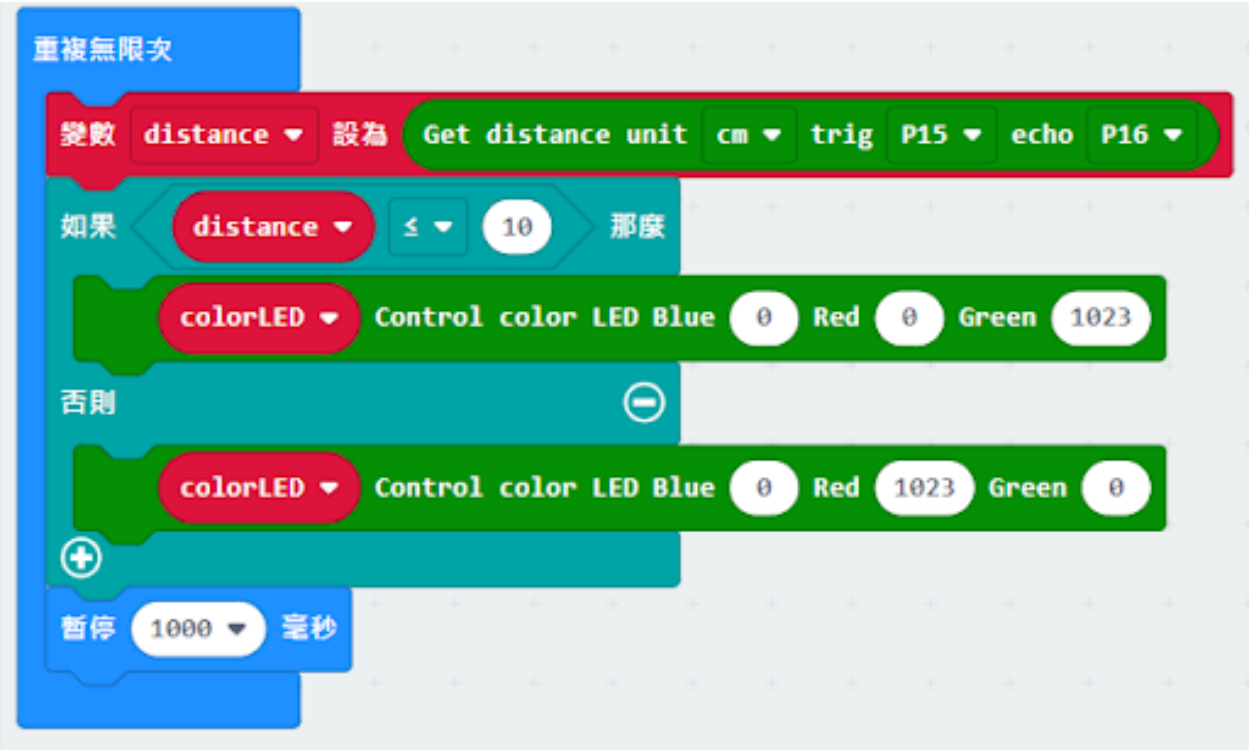


獲取距離值

- 在重Ⓕ無限次的版塊裏。將變數 distance 設Ⓕ get distance unit cm trig P15 echo P16，也就是Ⓕ通過連接距離傳感器到 P15 和 P16 獲得 distance 值。
- 將如果語句插入重Ⓕ無限次中，將如果語句設Ⓕ distance≤10。
- 把暫停拖入循環，Ⓕ下一次檢查等待 1 秒。

以距離值顯示指示色

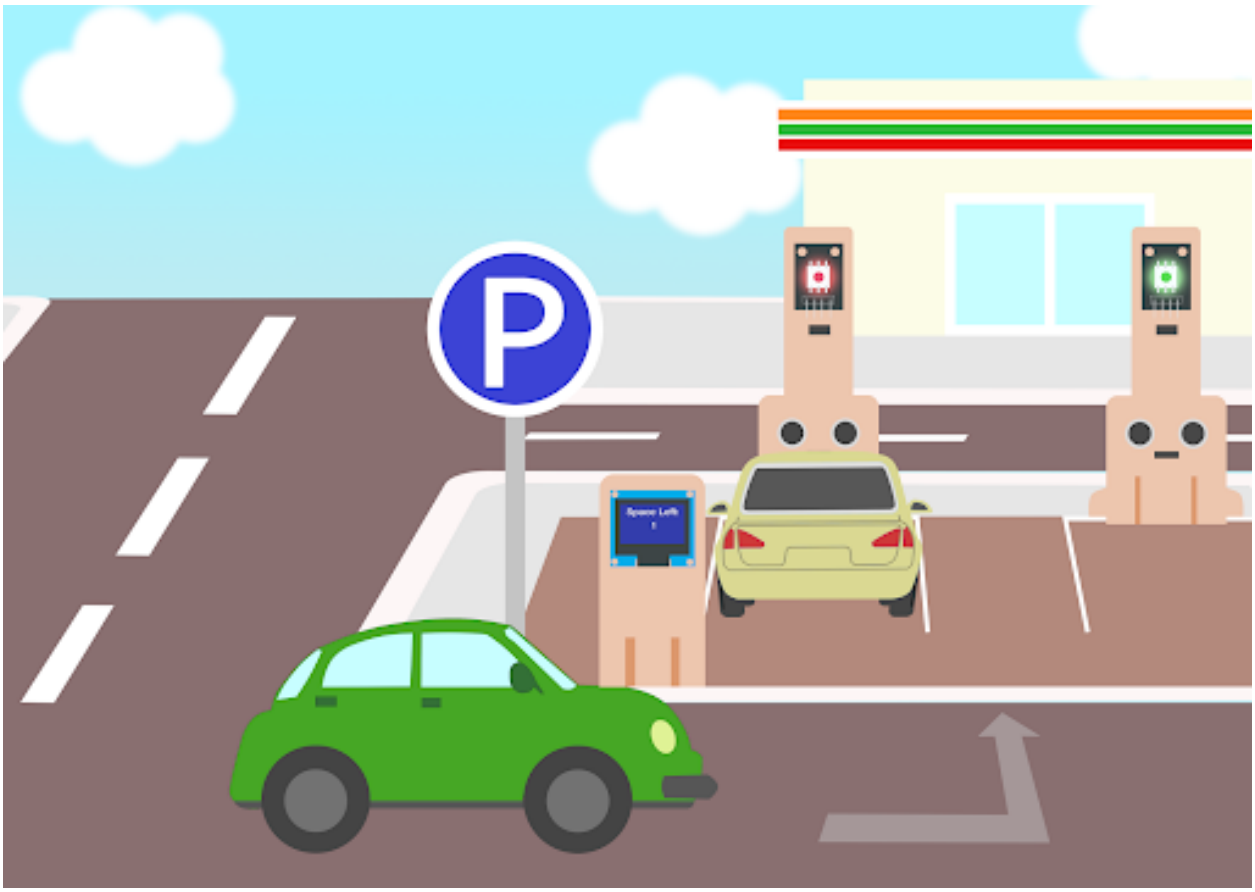
- 如果 distance≤10，則顯示Ⓕ色，否則顯示紅色。



總結

思考

1.2.5 智能停車場障礙一：車位監測

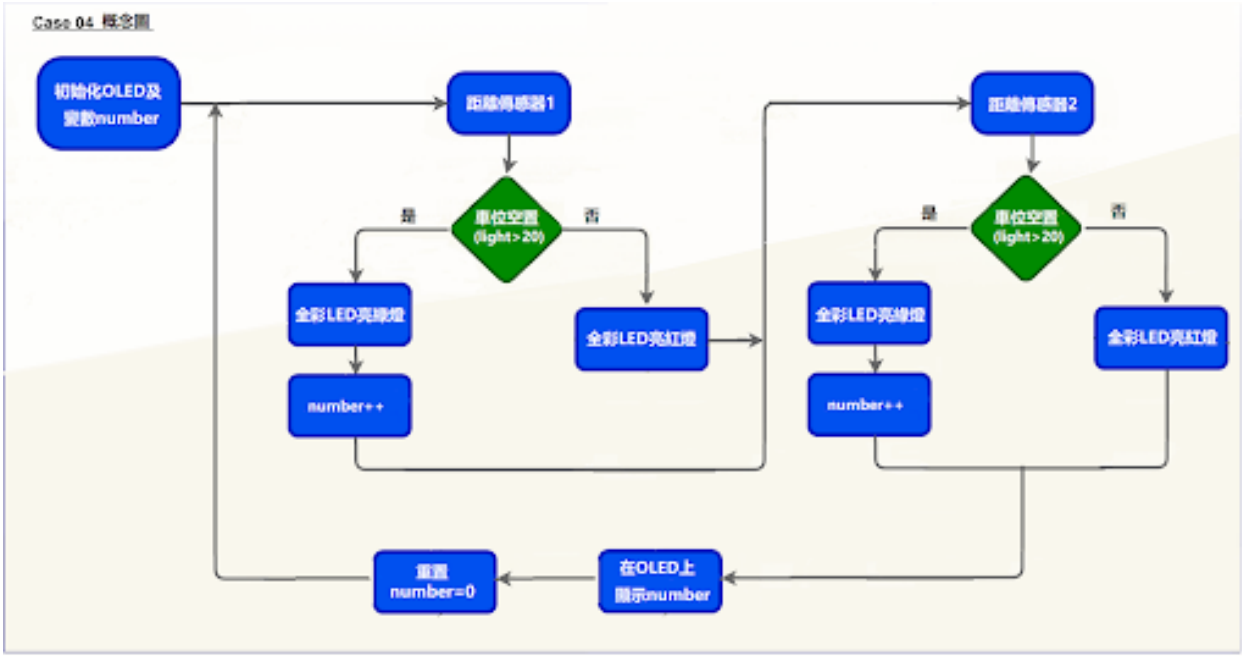


目標

背景

什麼是智能停車場監控系統？

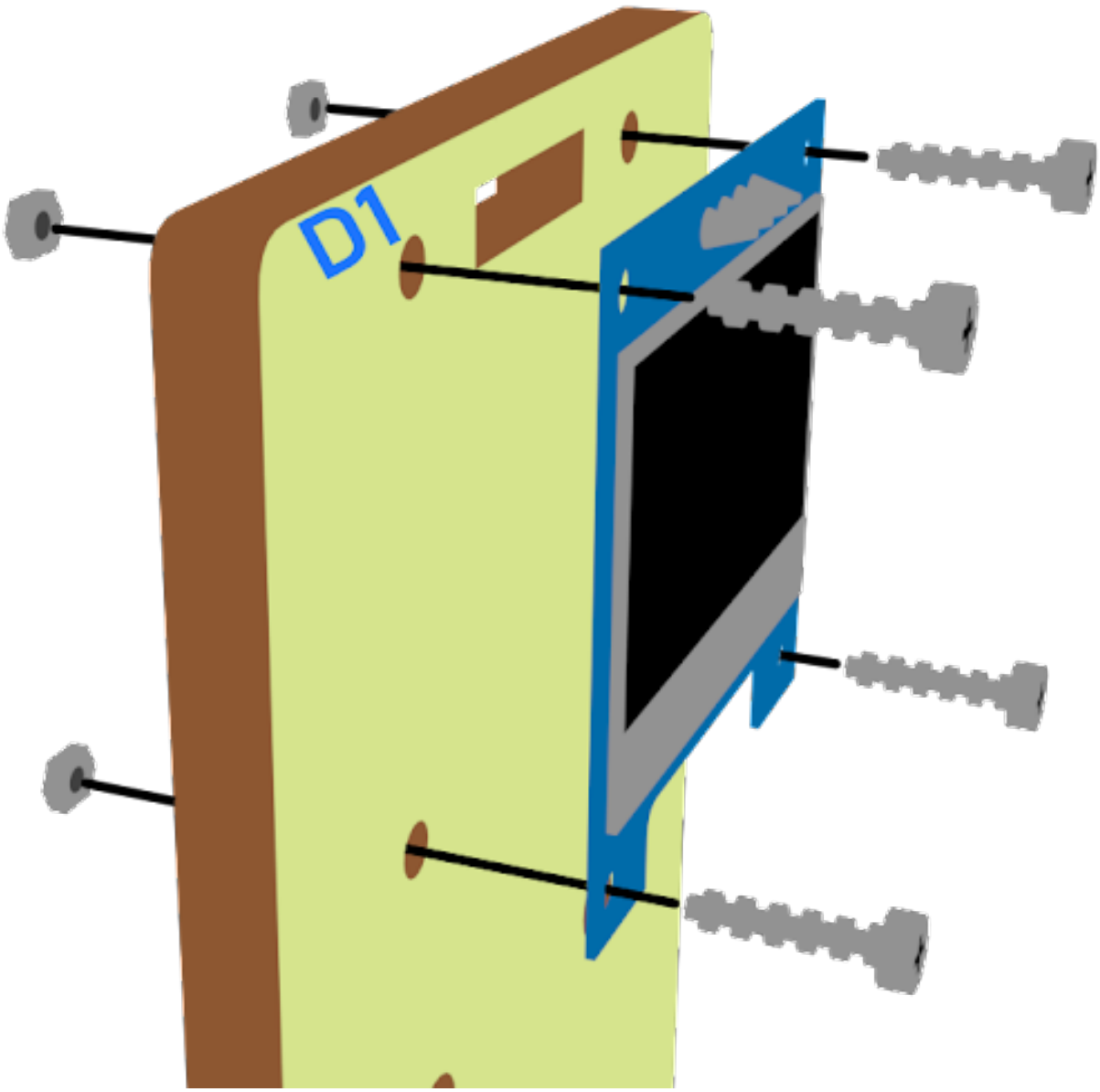
智能停車場監控系統運作



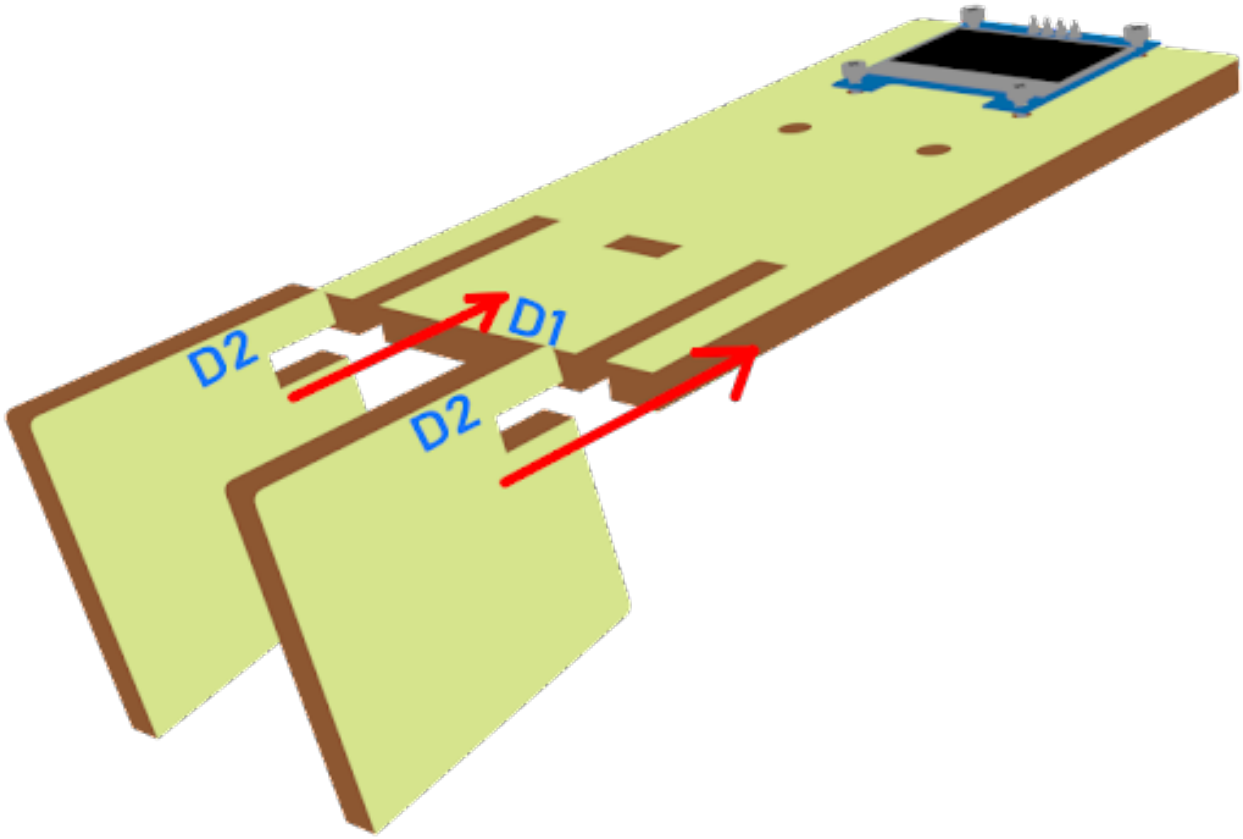
材料準備

組裝步驟

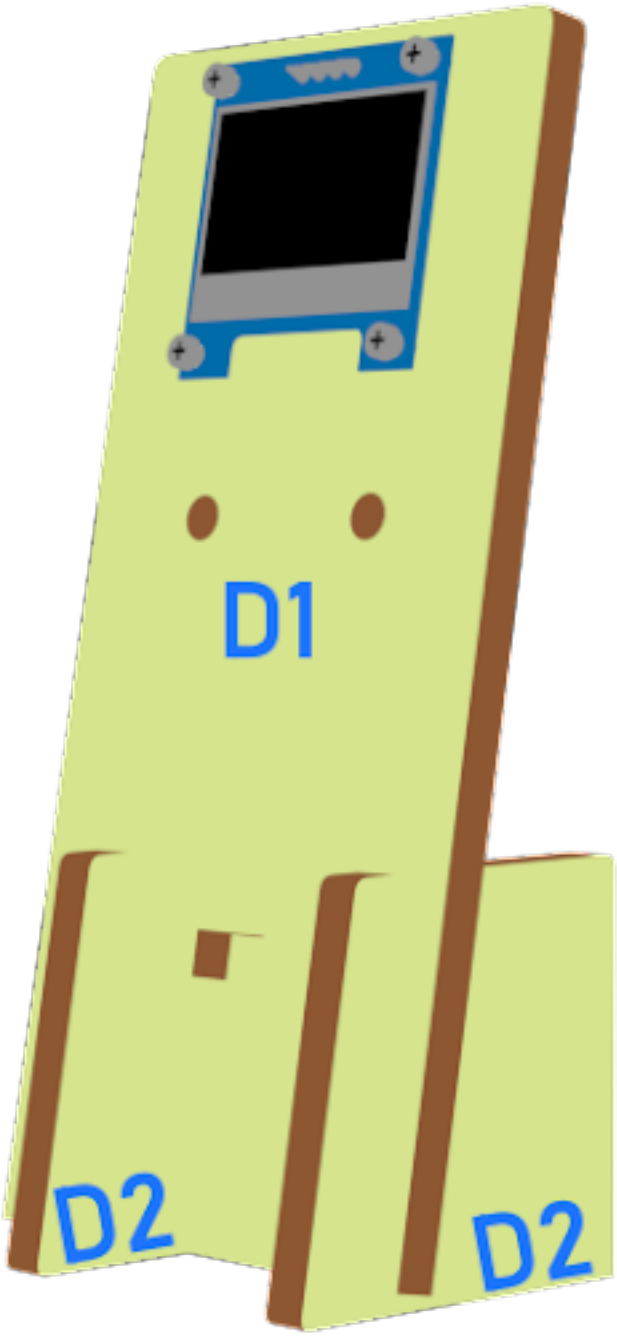
第一步



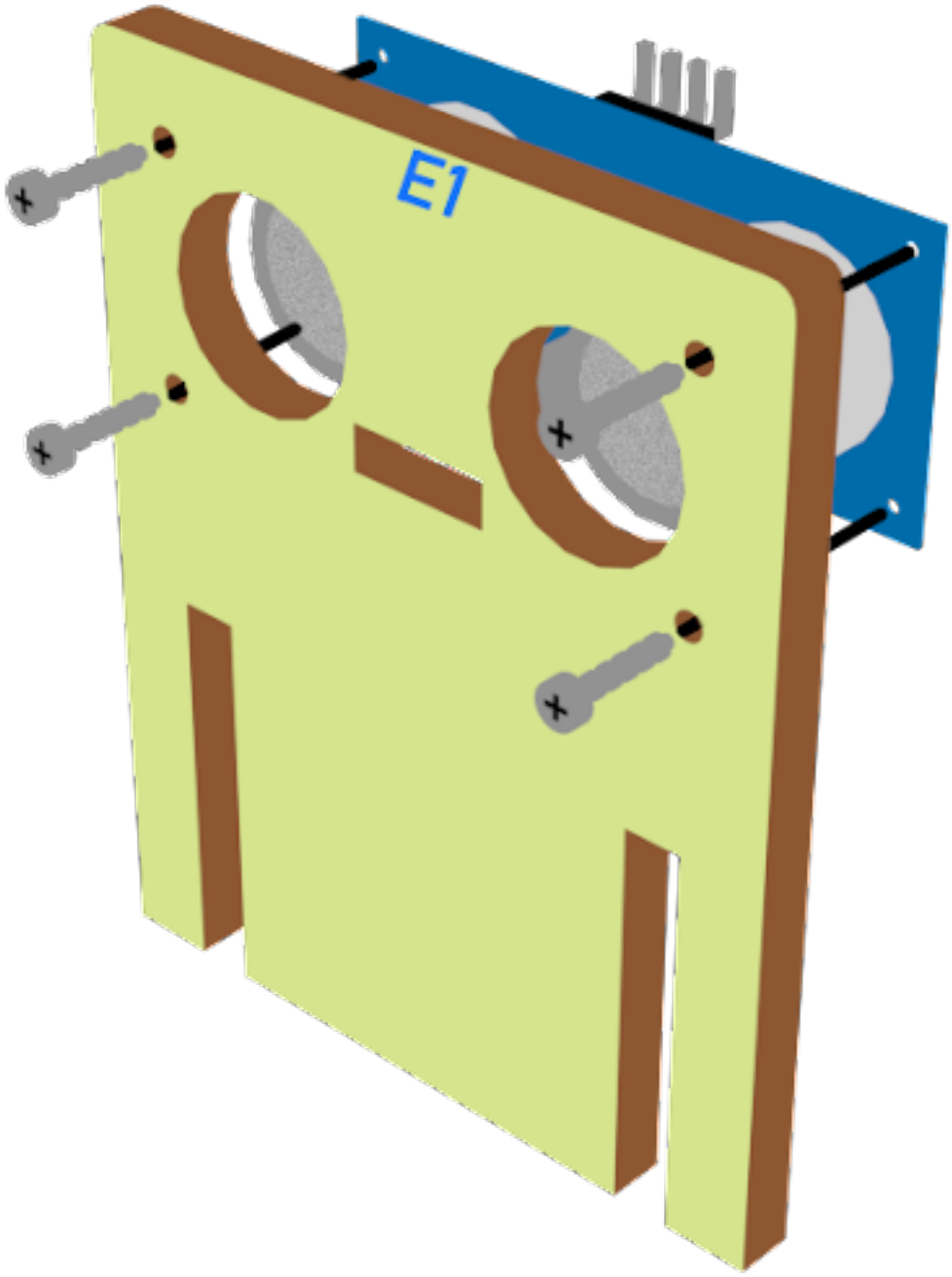
第二步



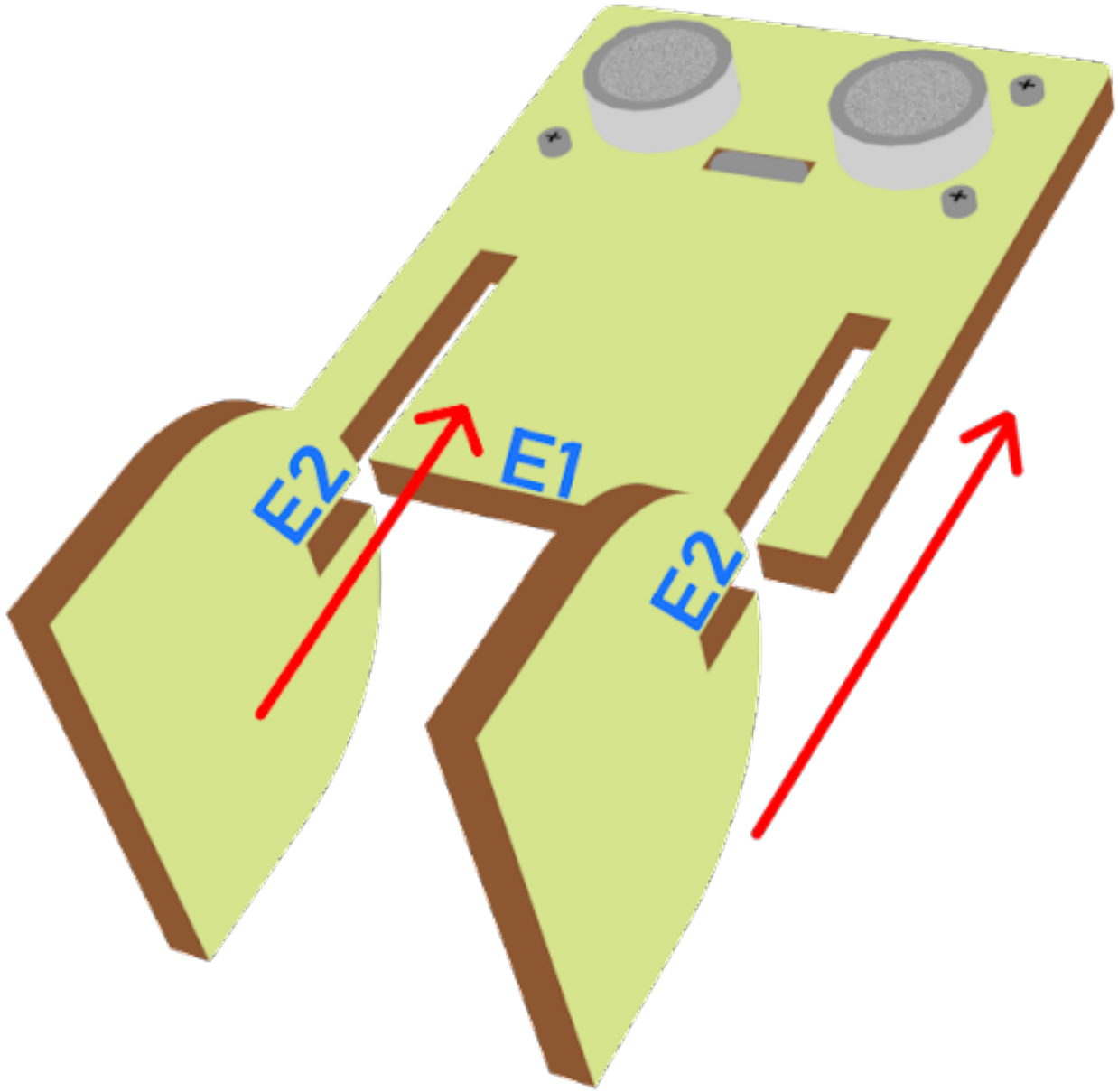
第三步



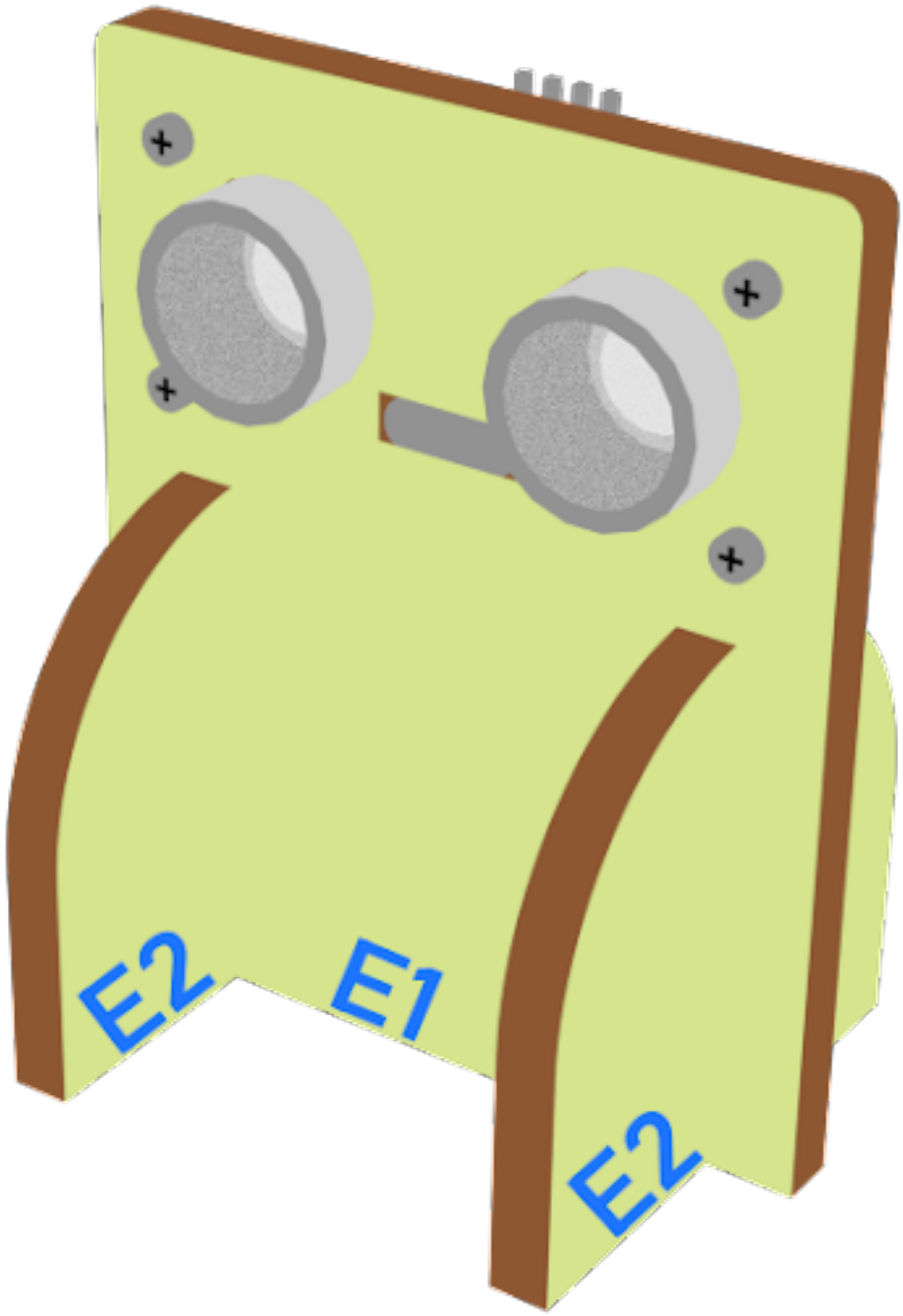
第四步



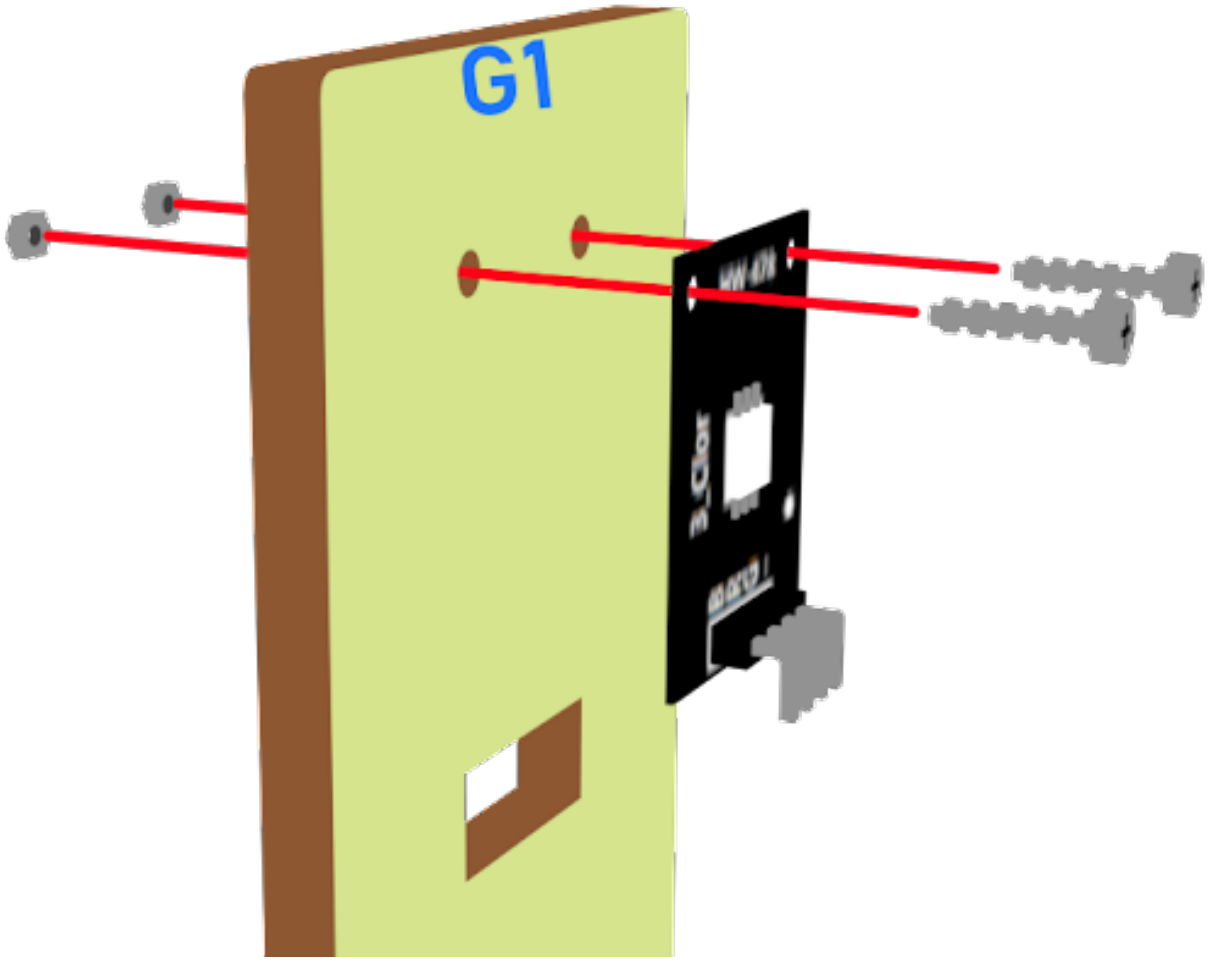
第五步



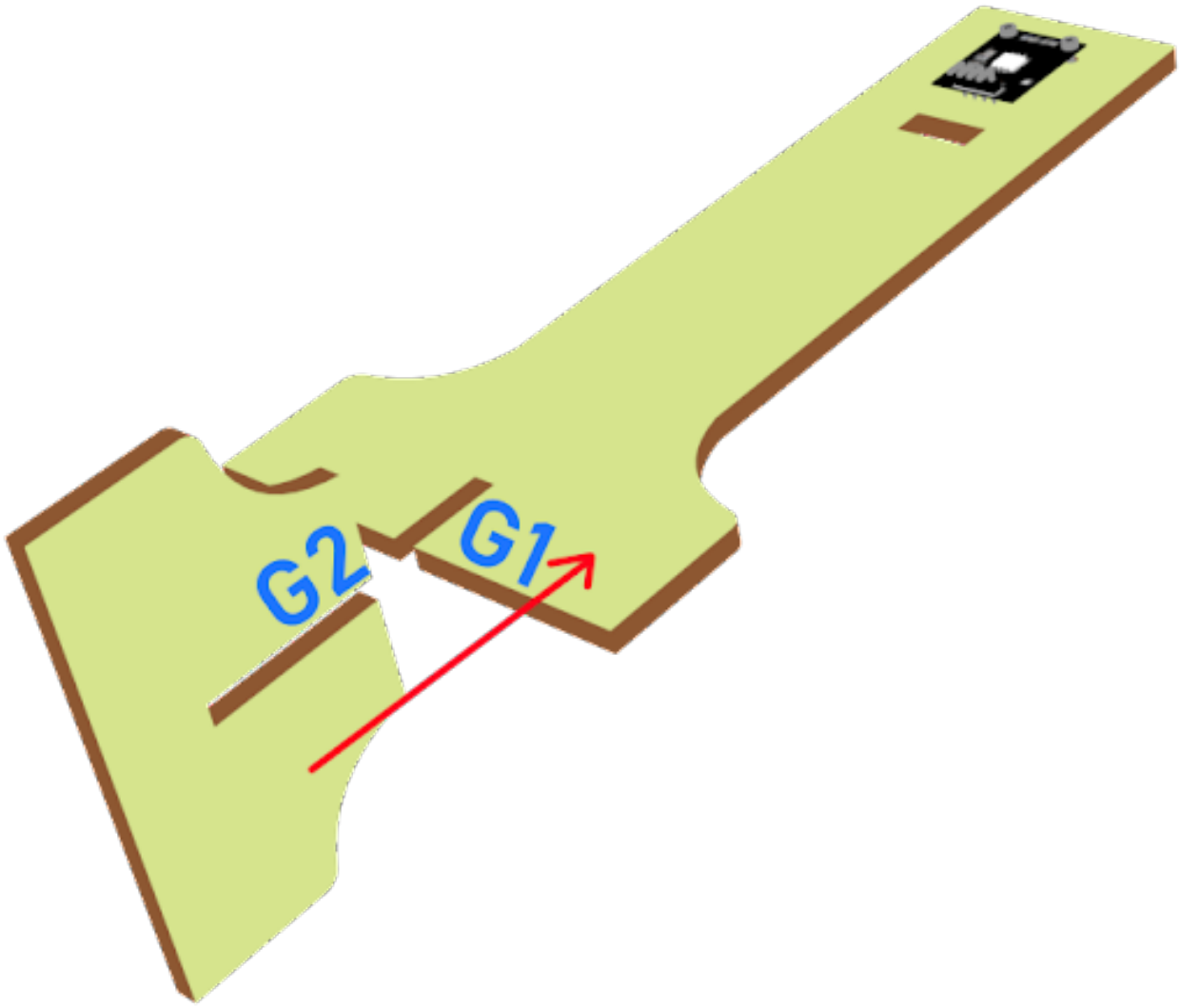
第六步



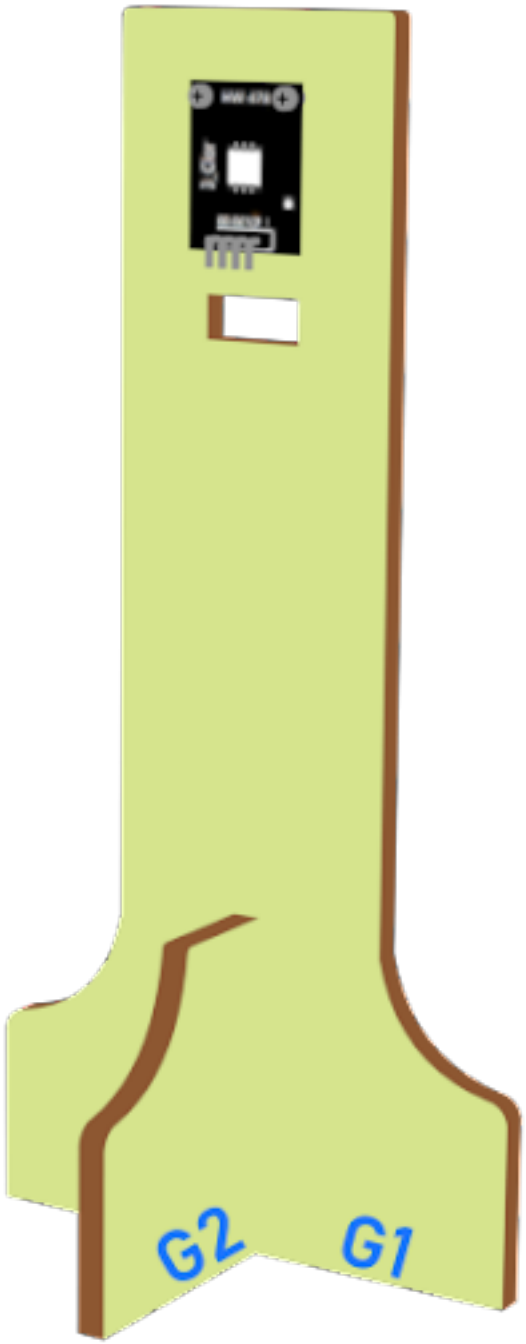
第七步



第八步



第九步



硬件連接

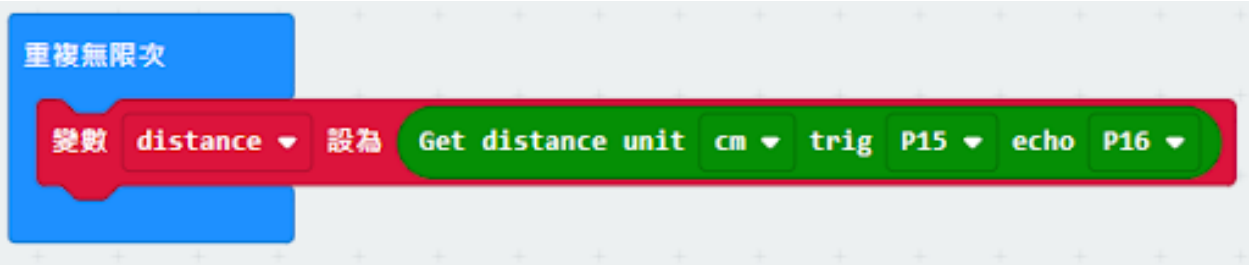
編程

在起始位置設置變量Ⓕ初始化多色 LED 及 OLED 屏幕

- 拖動 OLED 初始化寬：128 高:64，至當Ⓕ動時。
- 在當Ⓕ動時中，從變數中拖入變數 distance 設Ⓕ 0，變數 number 設Ⓕ 0。
- 拖入 set colorLED to color led pin setting …。

獲得距離的數值

- 拖動變數 distance 設Ⓕ get distance unit cm trig P15 echo P16，將該值存儲到變量 distance。



顯示指示色及計算空置車位數量

- 將如果語句放入重Ⓕ無數次，設變數 distance>10
- 如果 distance>10，則顯示Ⓕ色，否則顯示紅色。
- 如果 distance>10，變數 number 設Ⓕ 1

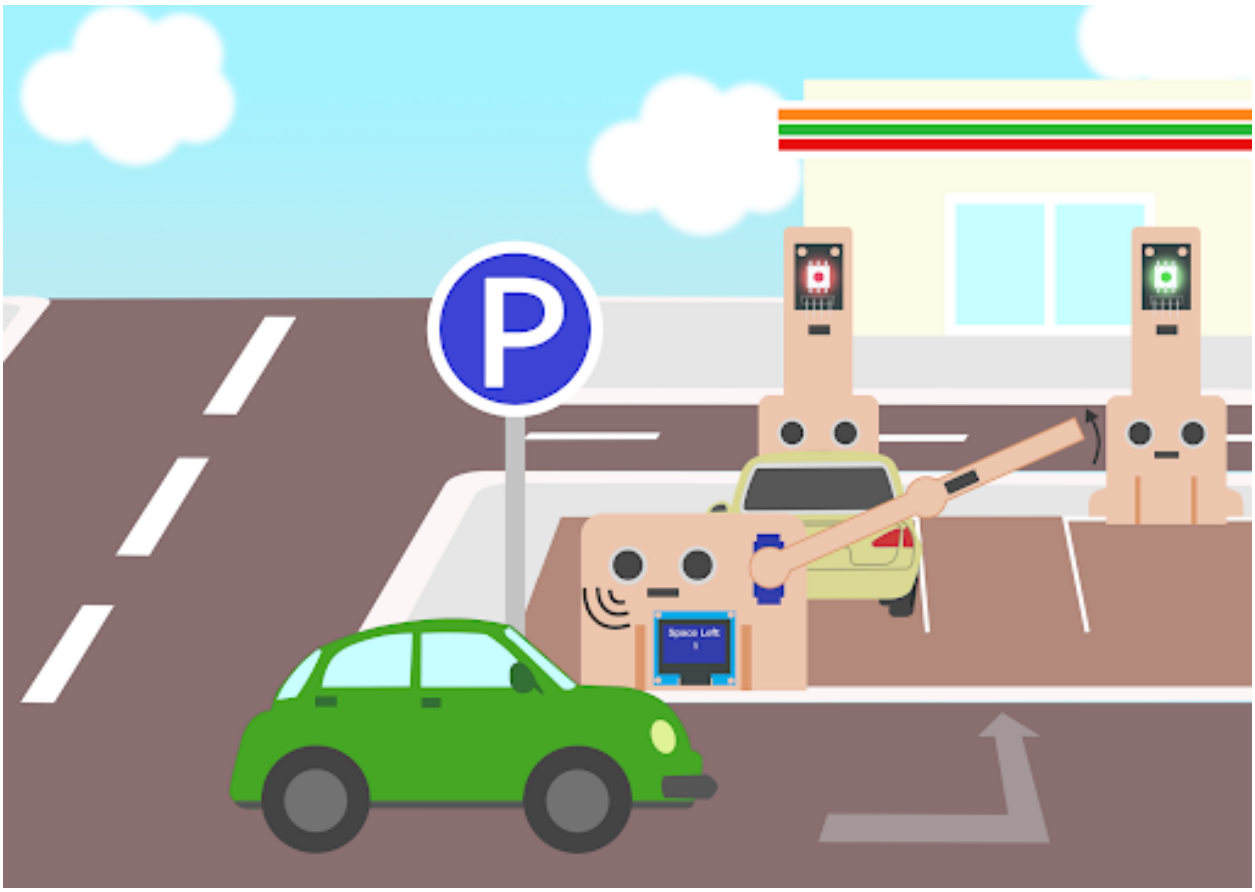
在 OLED 上顯示空置車位數量

- 從 OLED 中拖出清除 OLED 顯示，以避免重F。
- 拖出新行數字F顯示變數 number 的值。
- 把暫停拖入循環，F下一次檢查等待 1 秒。
- 重置 number 至 0 以開始新的檢查。

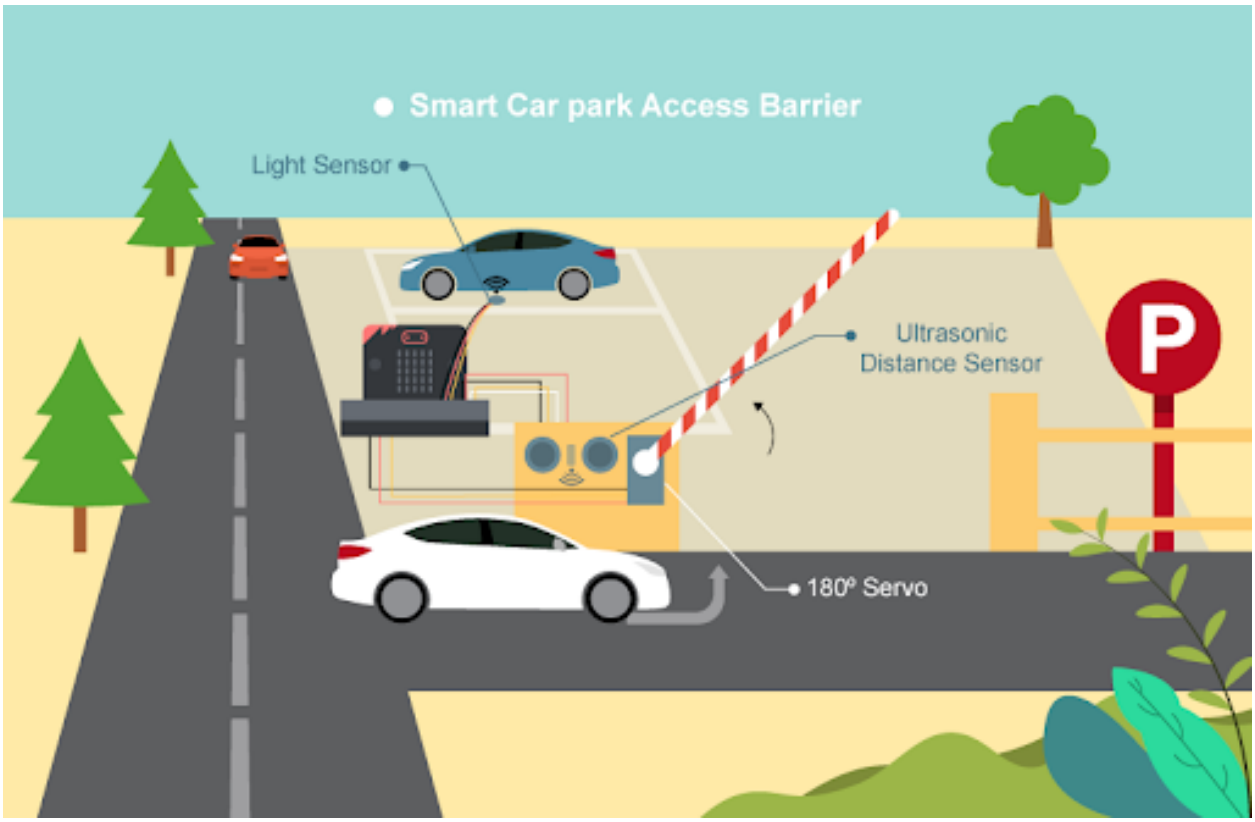
總結

思考

1.2.6 智能停車場障礙二：車閘



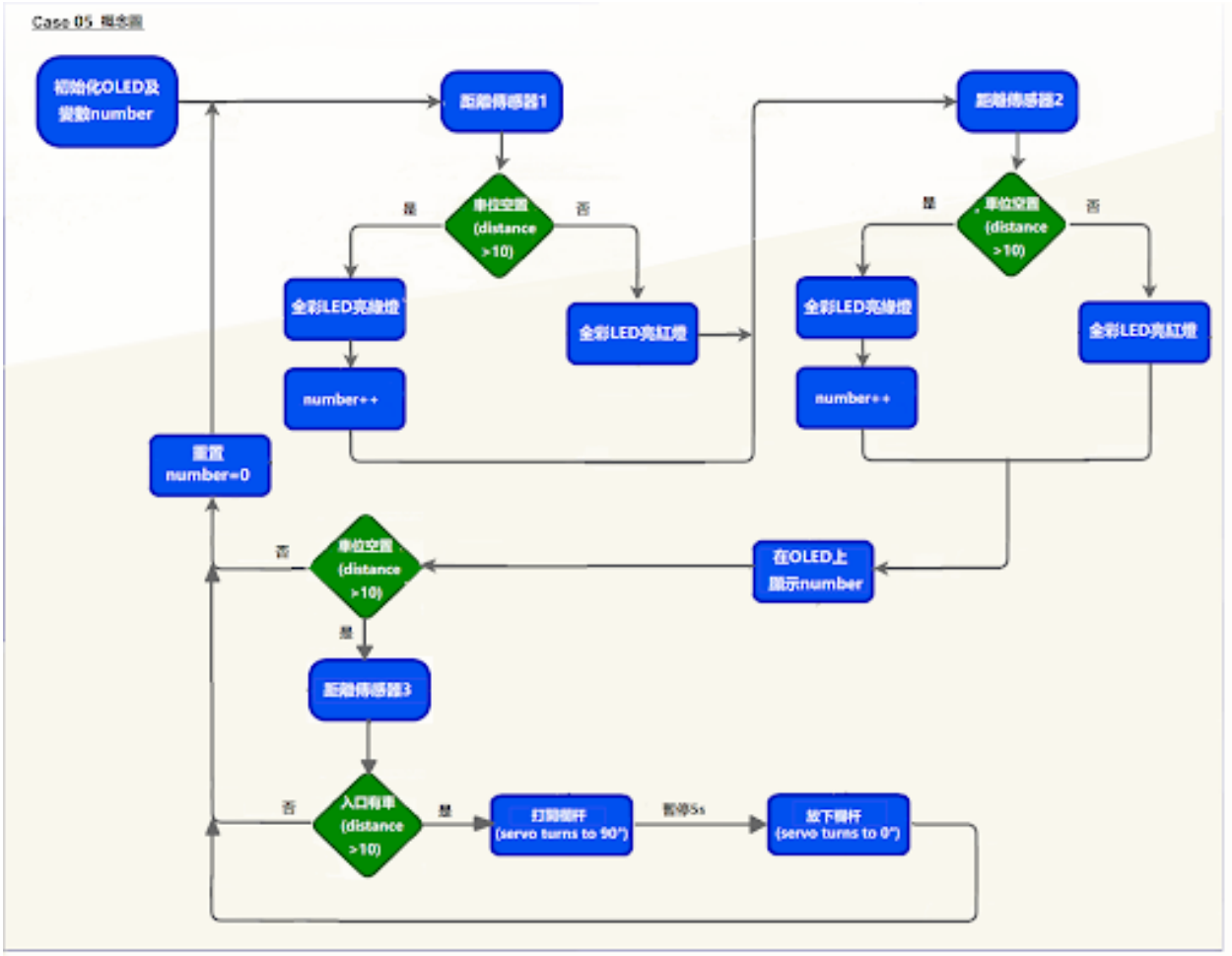
目標



背景

什麼是智能停車場入口欄杆？

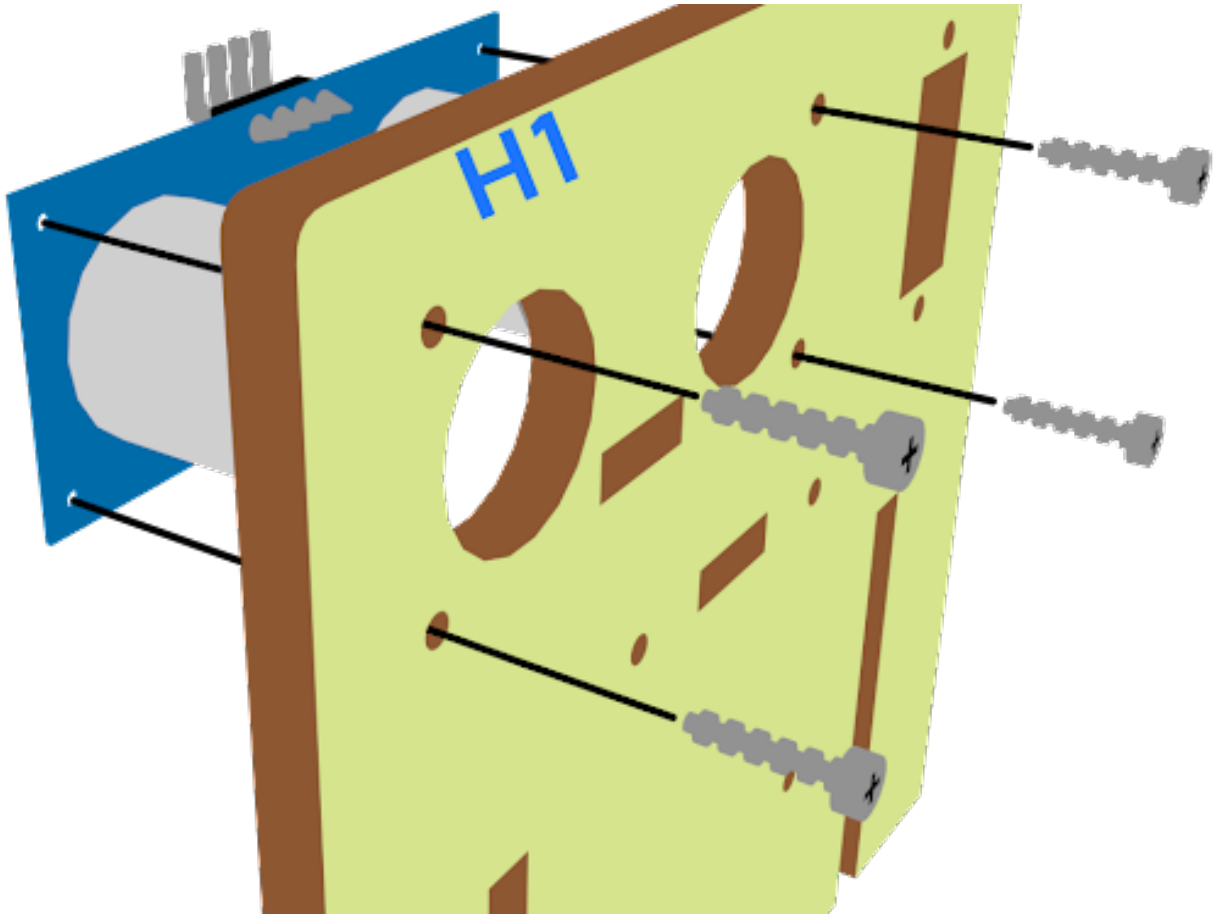
停車場欄杆運作



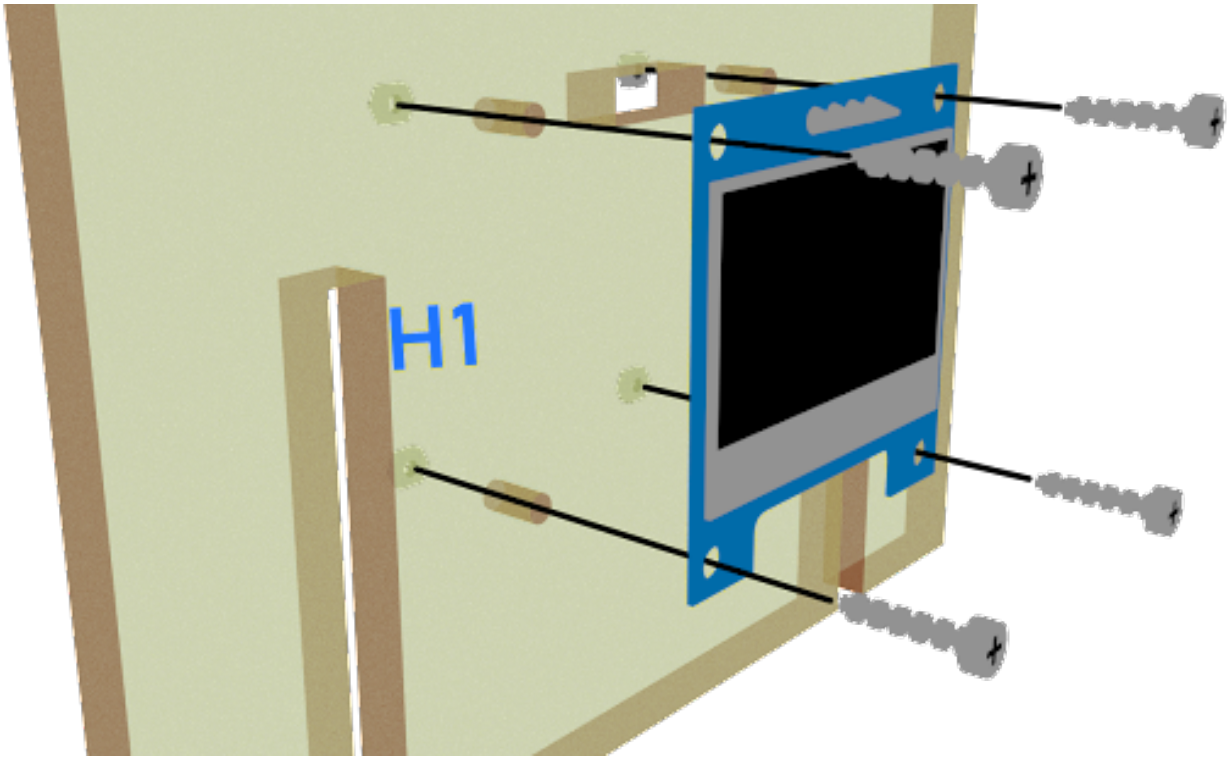
材料準備

組裝步驟

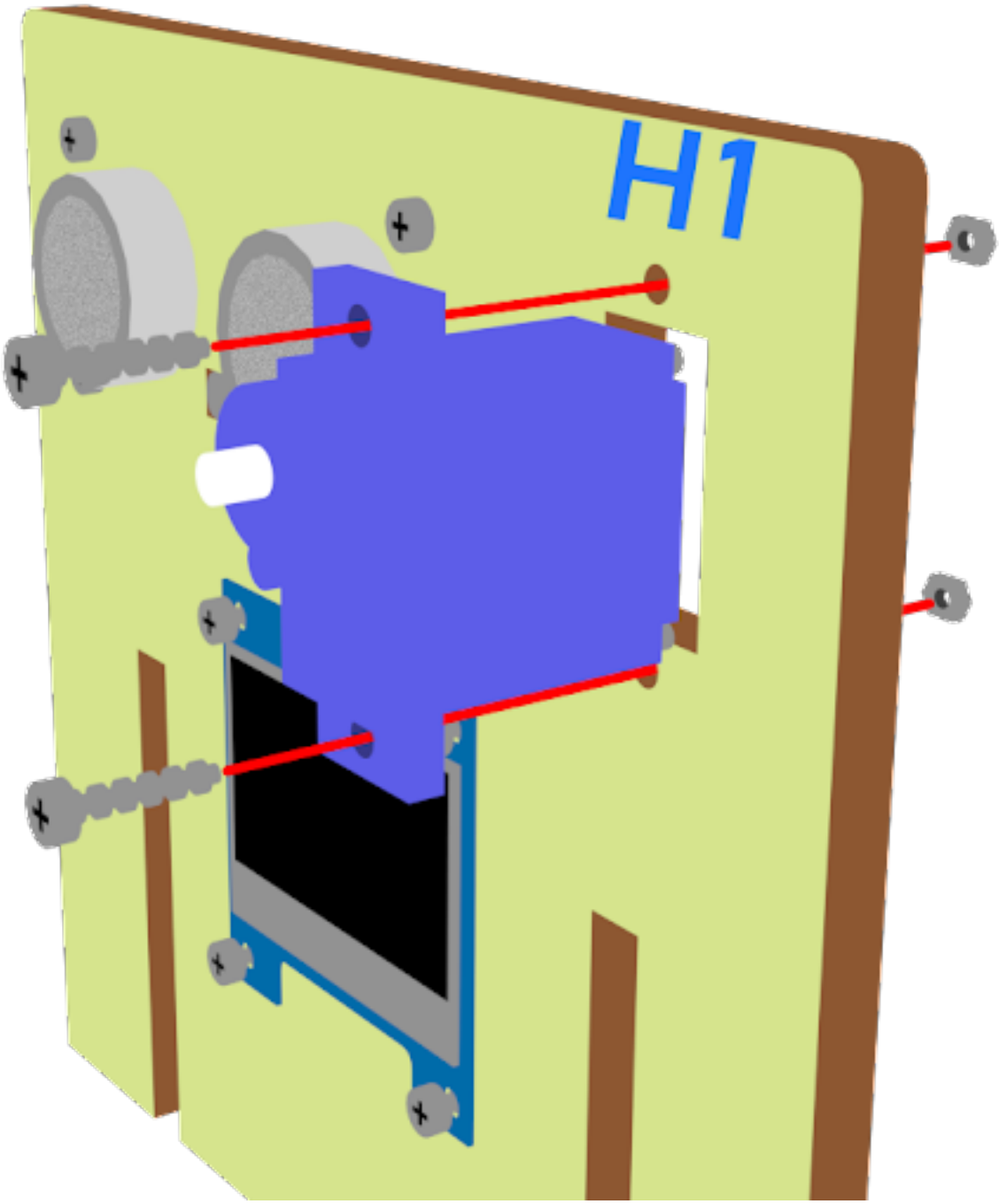
第一步



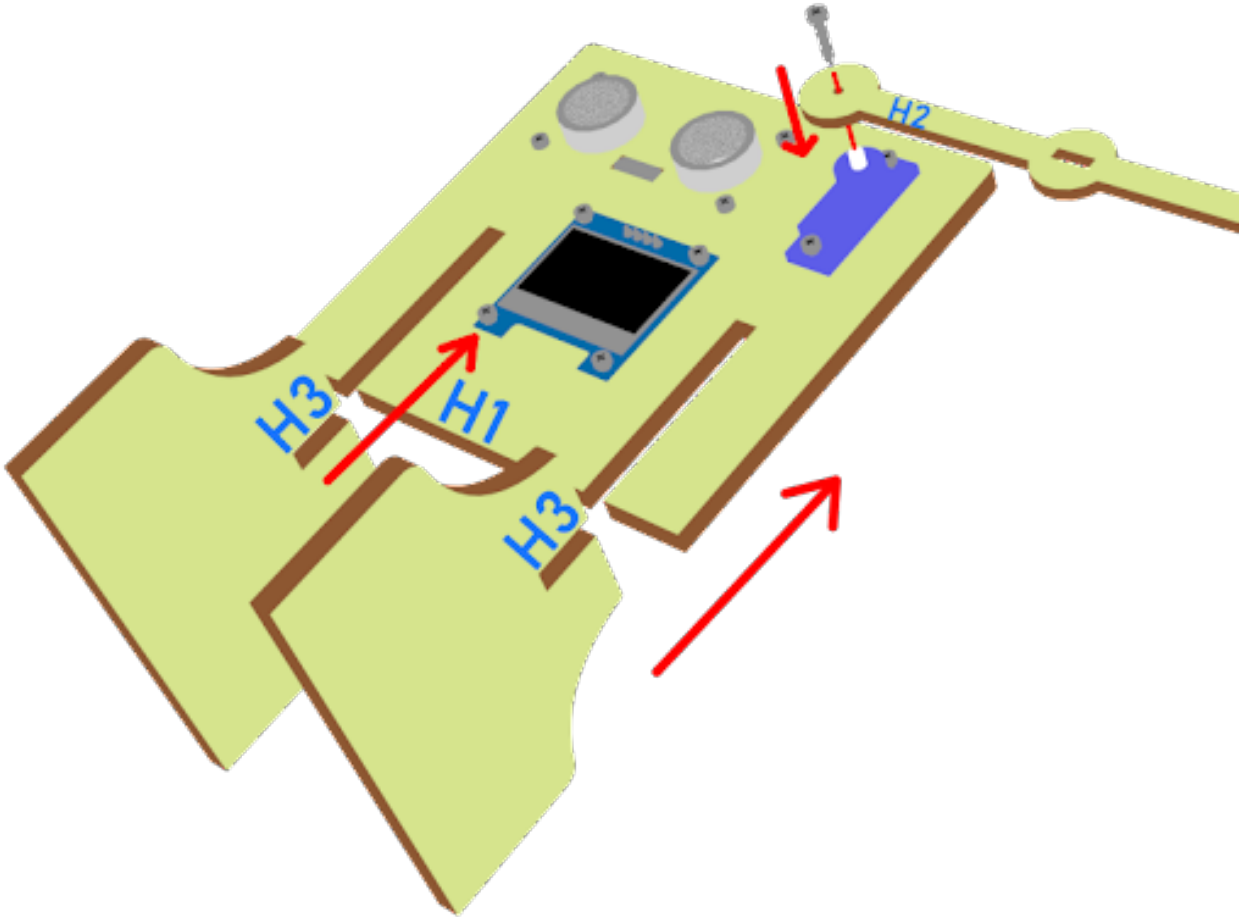
第二步



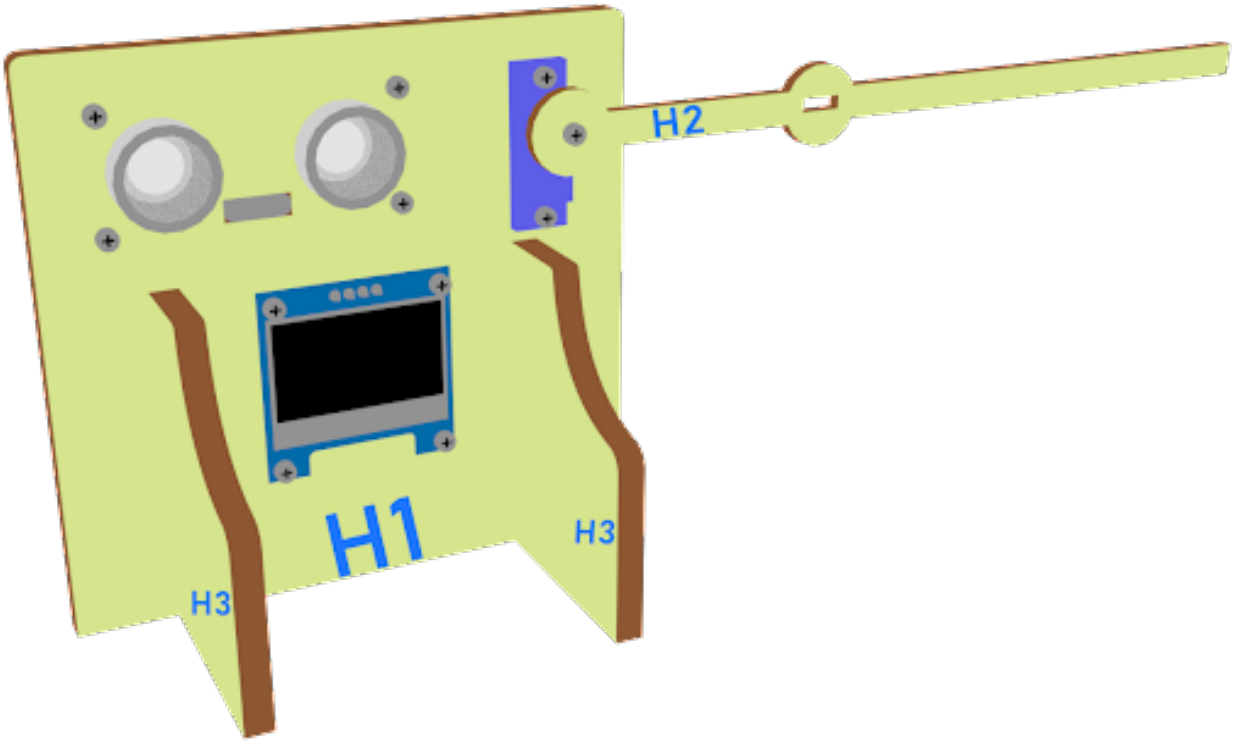
第三步



第四步



第五步

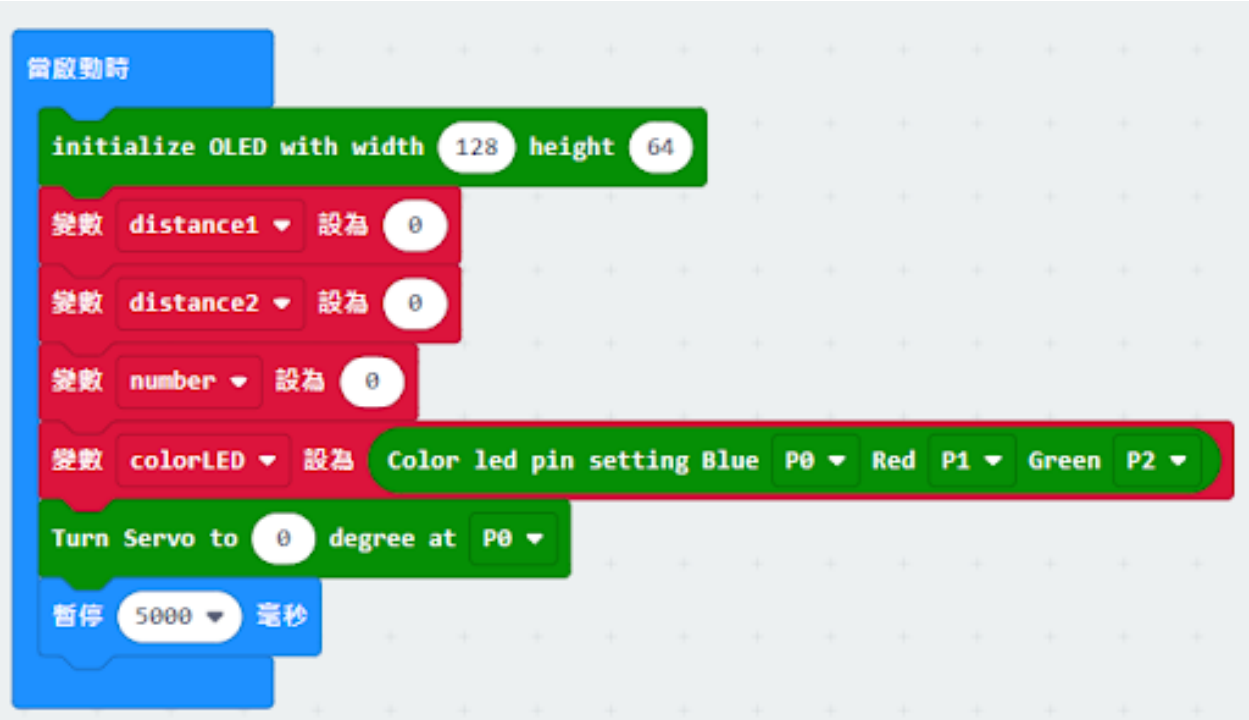


硬件連接

編程

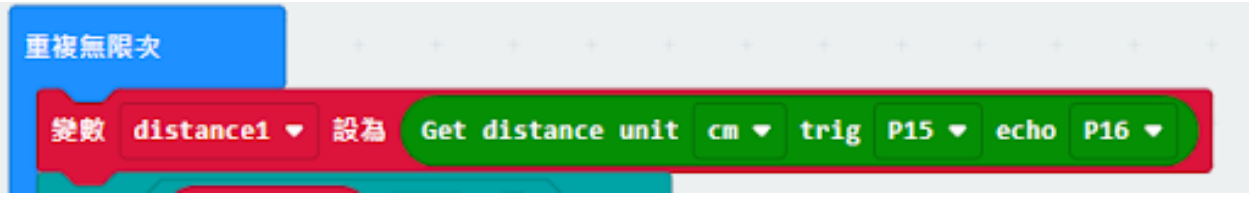
在起始位置設置變量，初始化 OLED 屏幕及舵機裝置

- 拖動 OLED 初始化寬：128 高:64，至當Ⓕ動時。
- 在當Ⓕ動時中，從變數中拖入變數 distance1 設Ⓕ 0，變數 distance2 設Ⓕ 0 及變數 number 設Ⓕ 0。
- 拖入 set colorLED to color led pin setting ⋯。
- 拖入 Turn Servo to 0 degree at P0。
- 拖出暫停 5 秒



獲得距離的數值

- 拖動變數 distance1 設F get distance unit cm trig P15 echo P16，將該值存儲到變量 distance1。

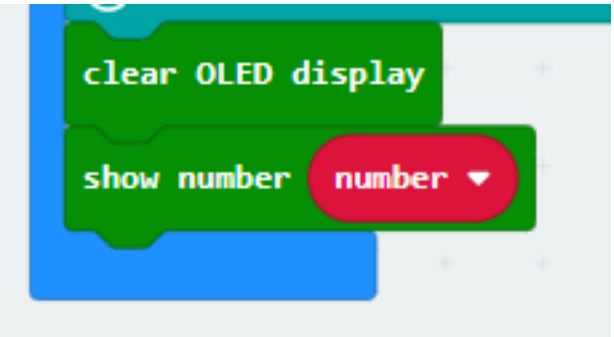


顯示指示色及計算空置車位數量

- 將如果語句放入重F無數次，設變數 distance1>10
- 如果 distance1>10，則顯示F色，否則顯示紅色。
- 如果 distance>10，變數 number 設F 1

在 OLED 上顯示空置車位數量

- 從 OLED 中拖出清除 OLED 顯示，以避免重Ⓕ。
- 拖出新行數字Ⓕ顯示變數 number 的值。

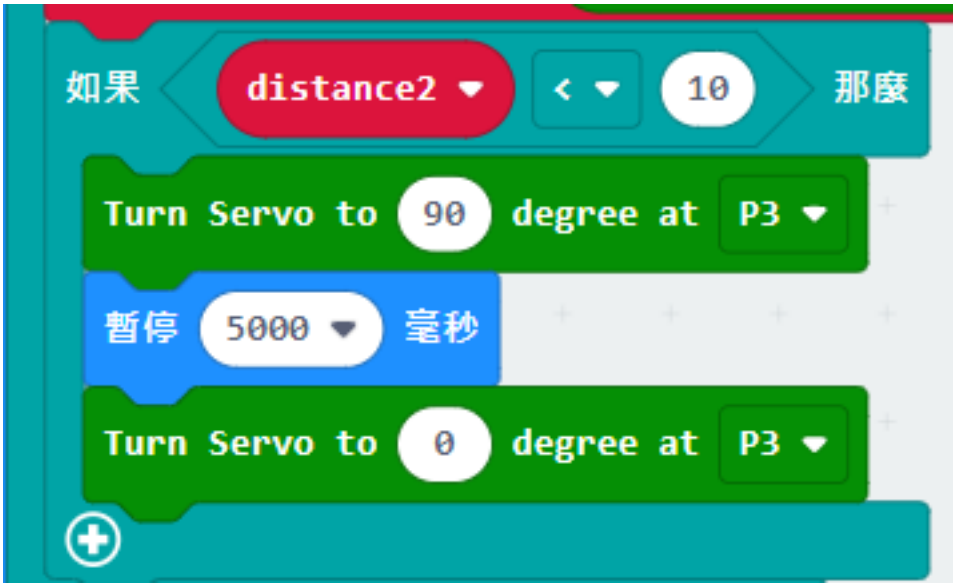


打開及放下欄杆

- 將如果語句放入重Ⓕ無數次，設變數 number>0
- 拖動變數 distance2 設Ⓕ get distance unit cm trig P8 echo P12，將該值存儲到變量 distance2。
- 將如果語句放入重Ⓕ無數次，設變數 distance2<10
- 把暫停拖入循環，Ⓕ下一次檢查等待 1 秒。
- 重置 number 至 0 以開始新的檢查。

設置舵機位置

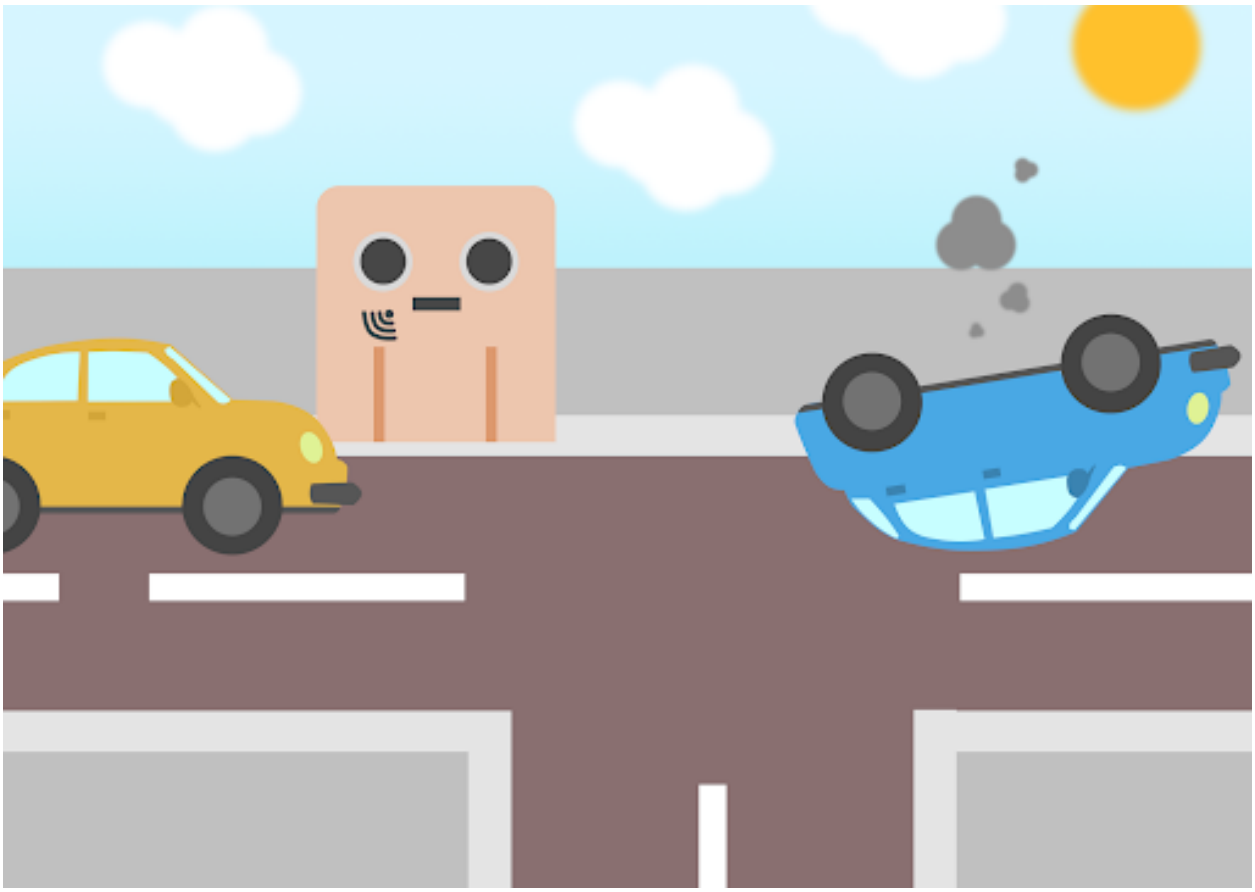
- 在 P3 處將舵機轉到 90 度即打開欄杆
- 暫停循環，等待 5 秒。
- S 在 P3 處將舵機轉到 0 度即放下欄杆。



總結

思考

1.2.7 壞車

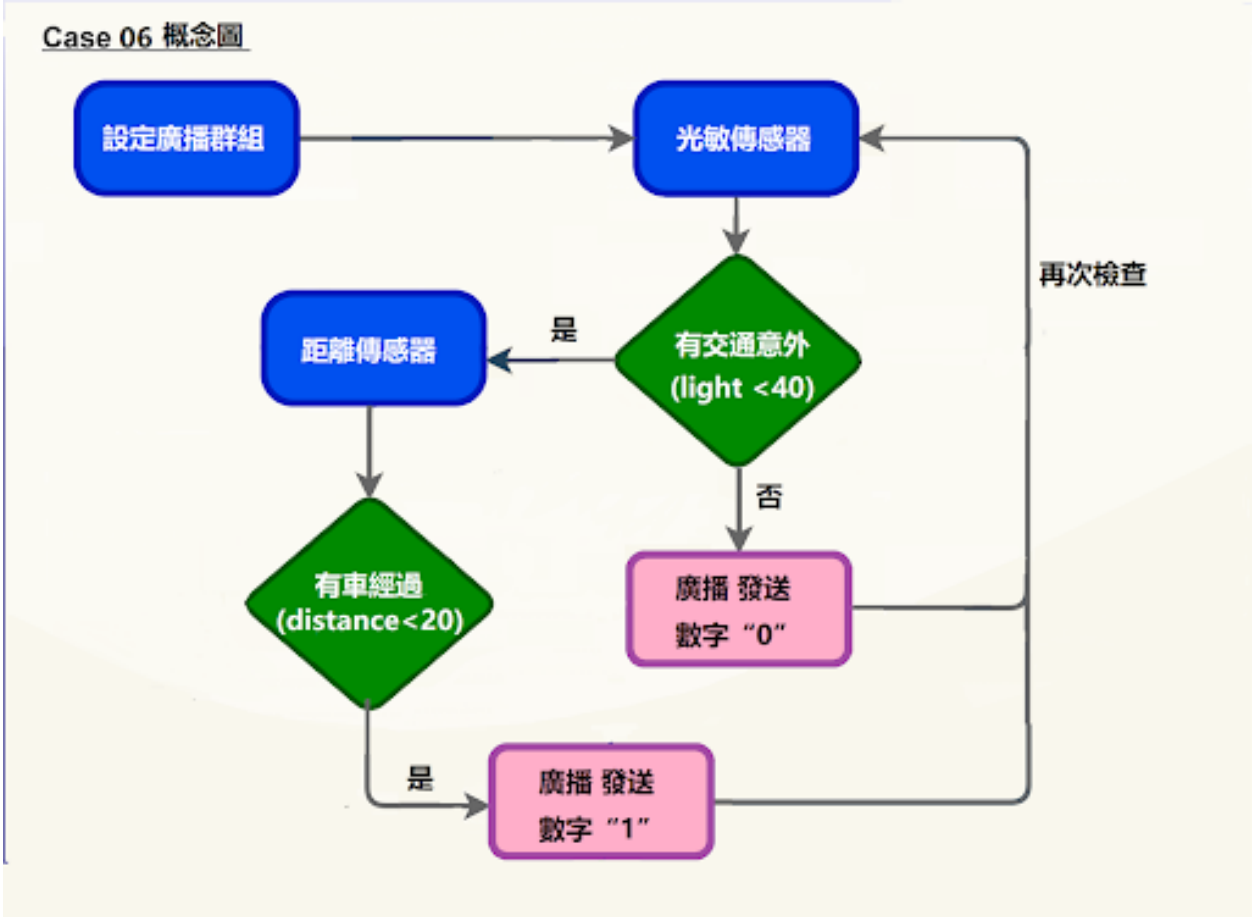


目標

背景

什麼是智能交通系統？

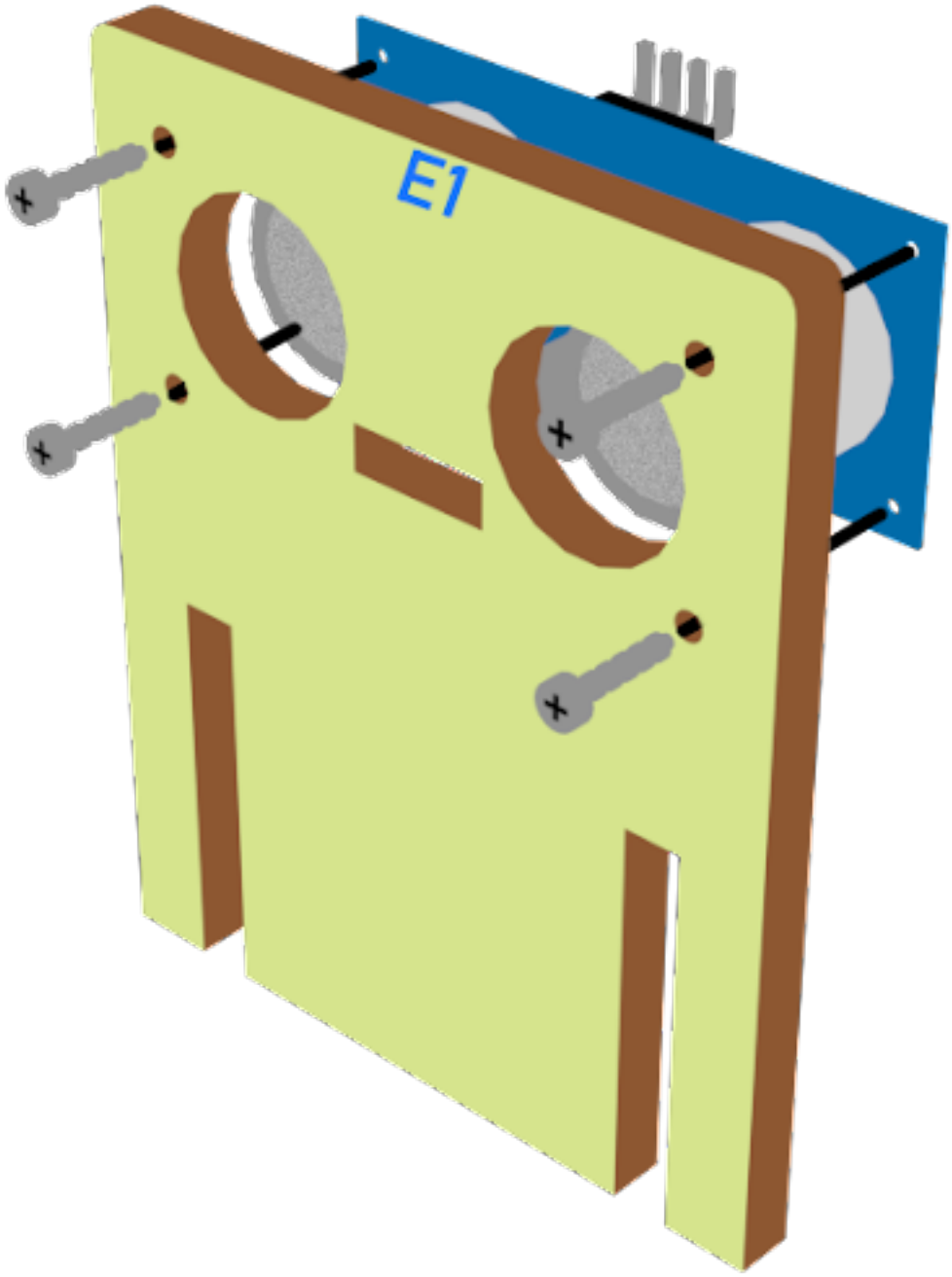
智能交通系統運作



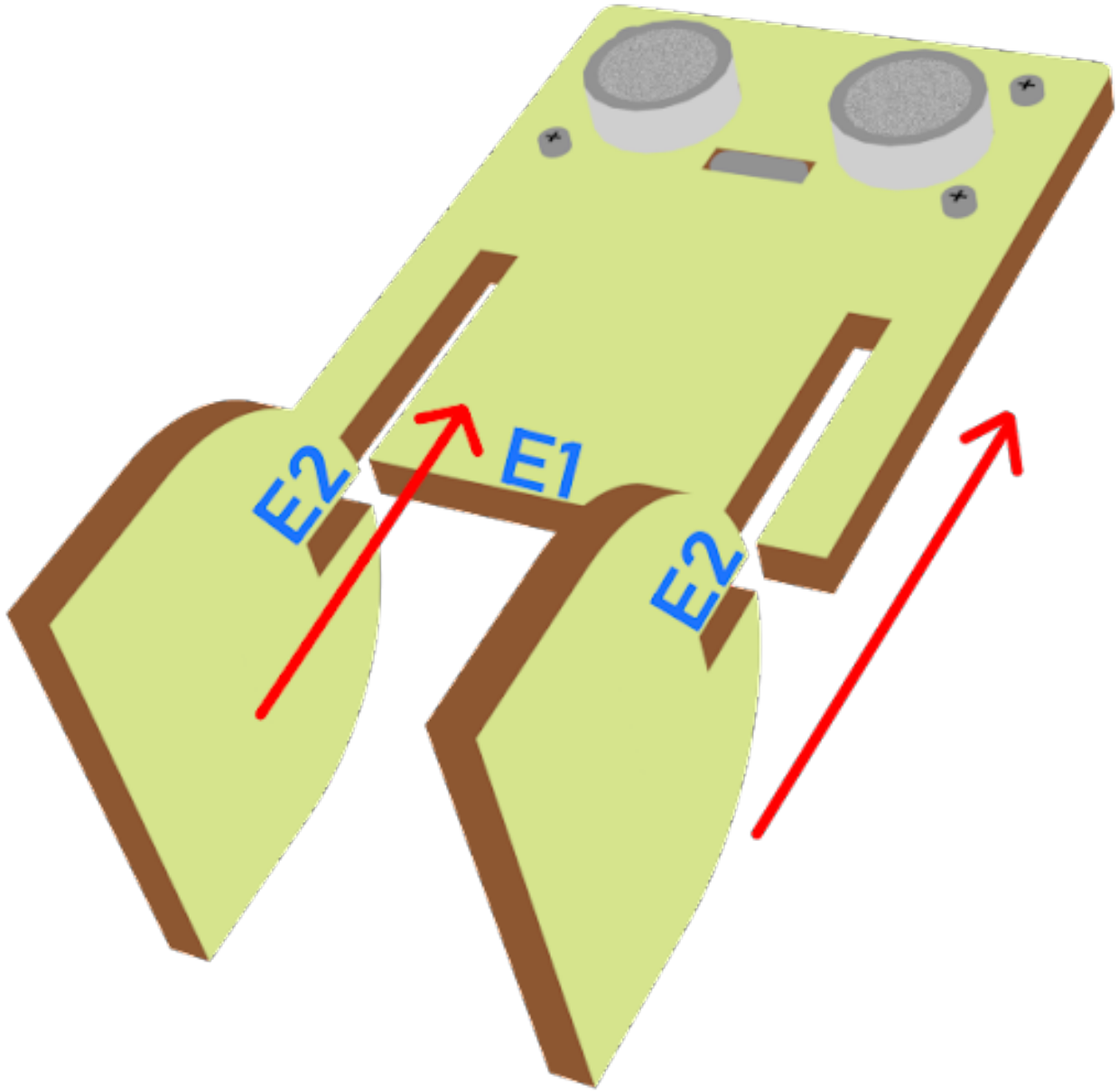
材料準備

組裝步驟

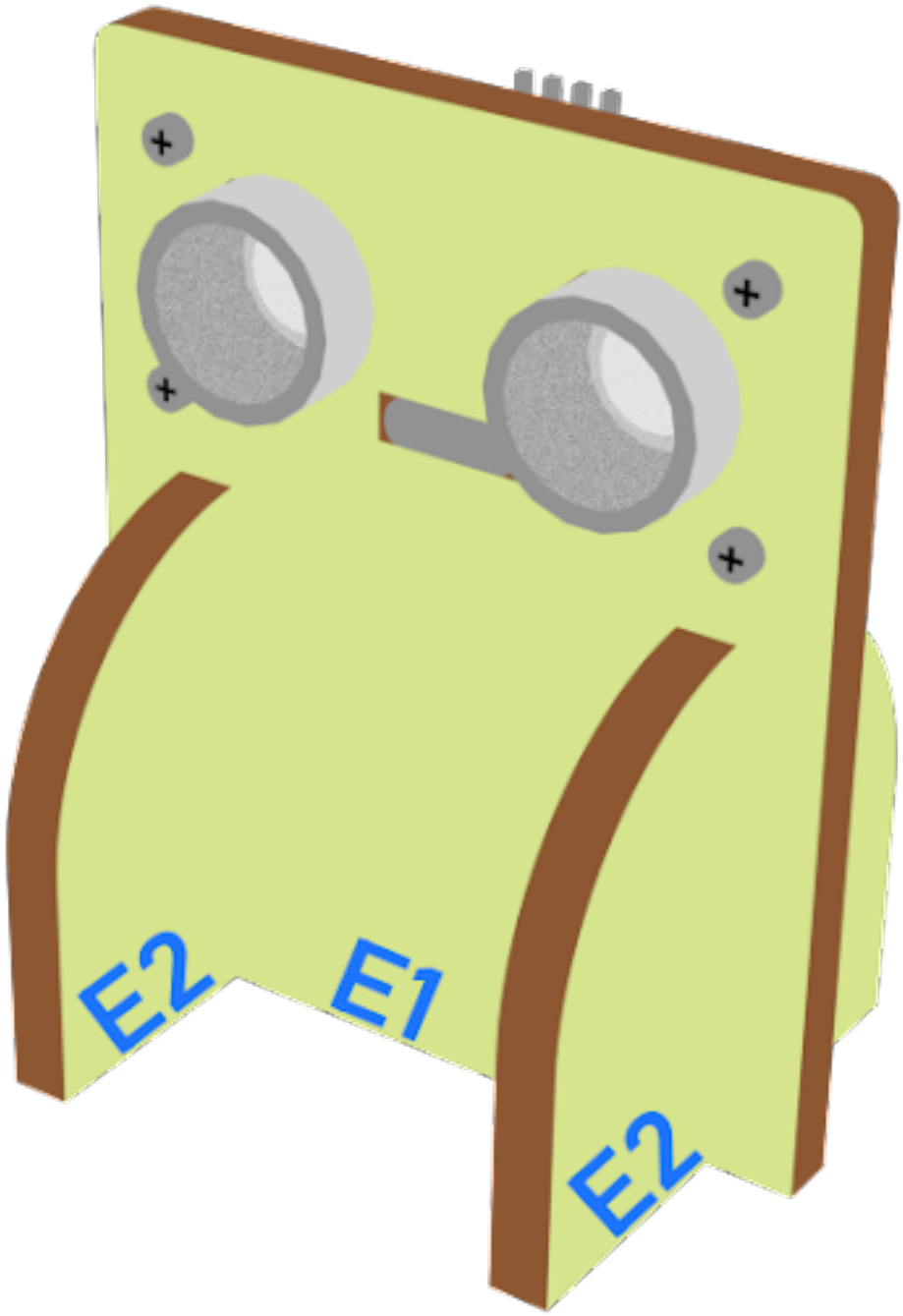
第一步



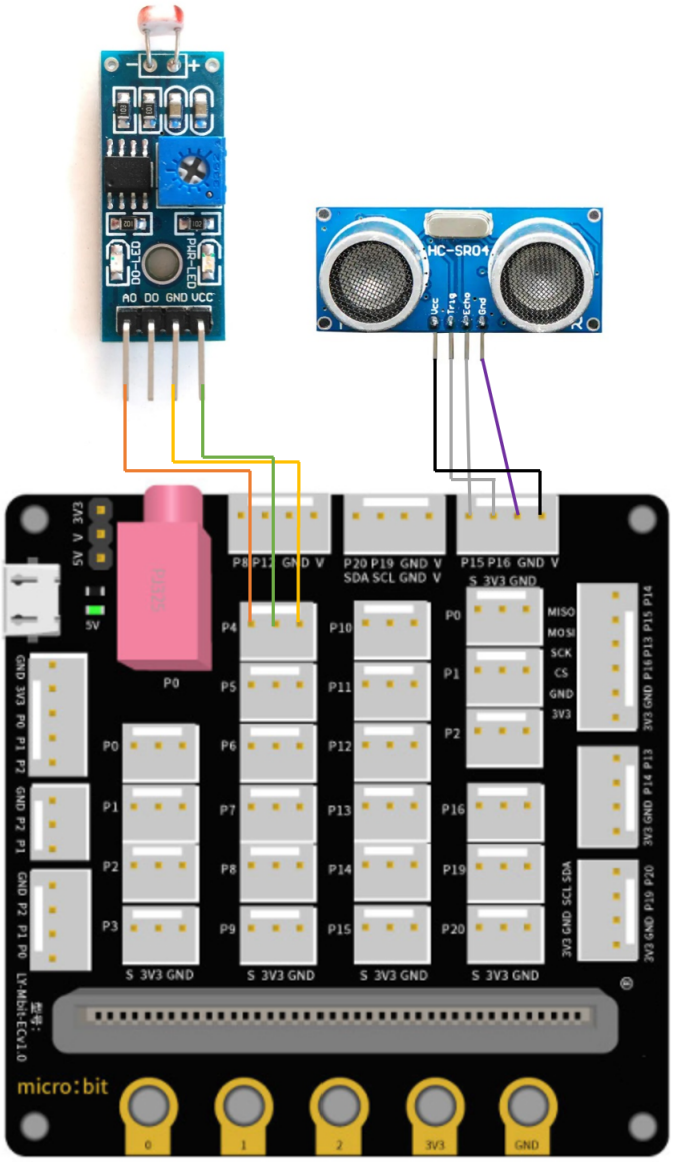
第二步

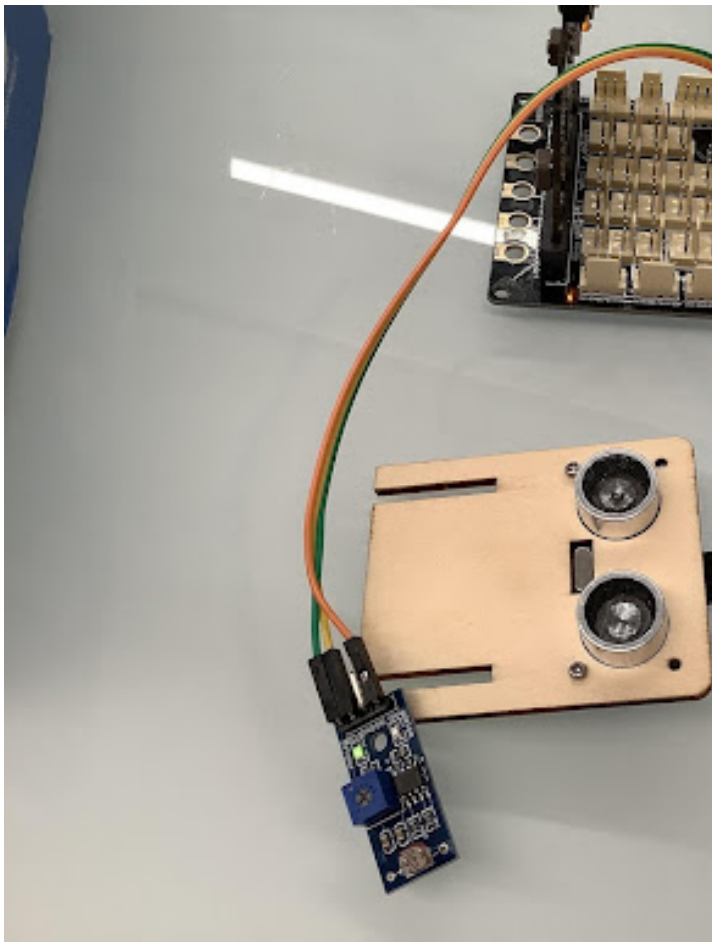


第三步



硬件連接





編程

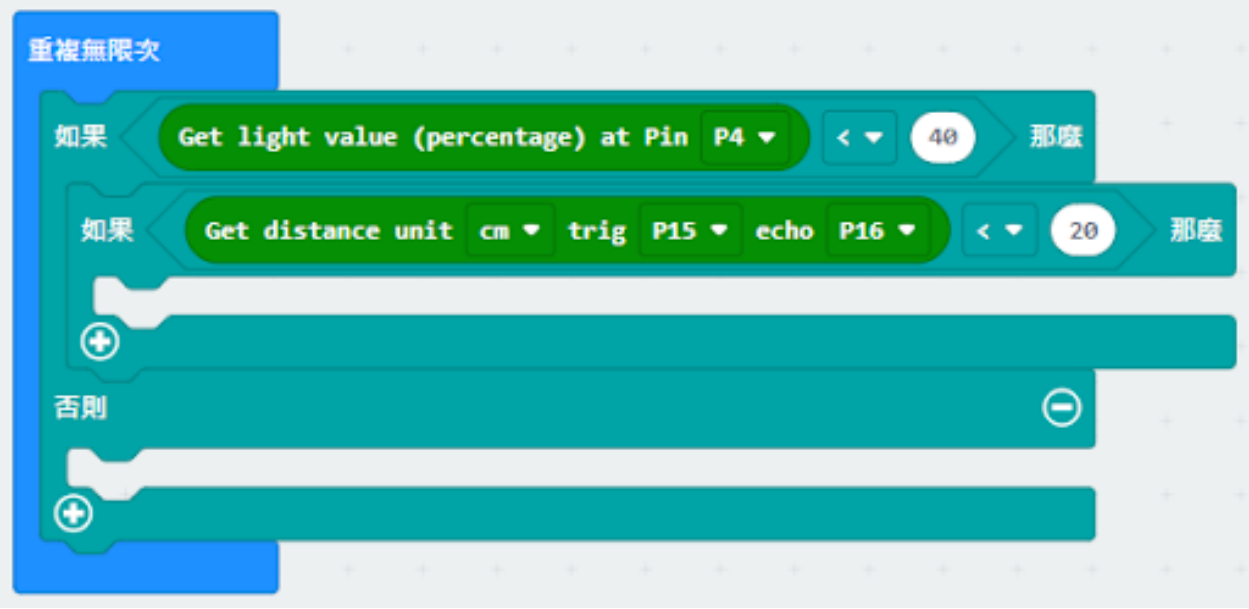
Sender:

在起始位置設置廣播群組

- 將廣播群組設爲 6 拖入當啓動時

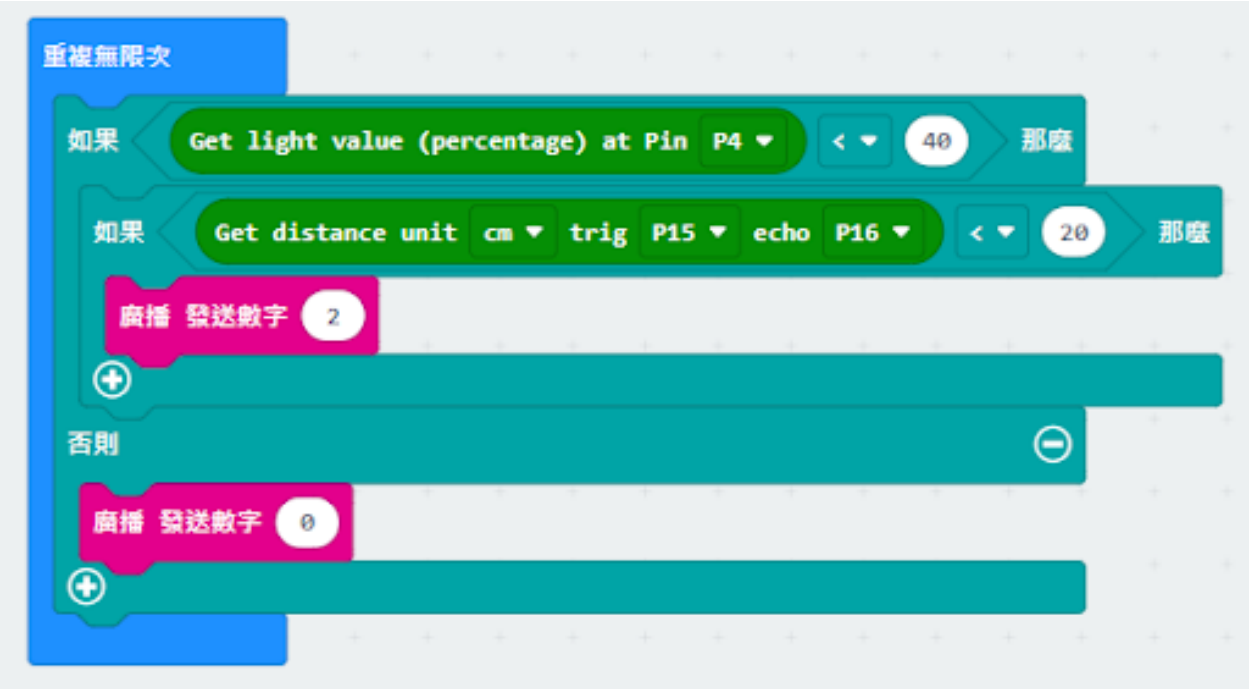
獲得光及距離的數值

- 將如果語句放入重F無數次，設 get light value (percentage) at pin P4 <40
- 如果 get light value (percentage) at pin P4 < 40，再拖入如果 get distance unit cm trig P15 echo 16 < 20



發送廣播數字以控制車輛

- 將廣播發送數字 2 拖入 if 句式
- 將廣播發送數字 0 拖入 else 句式



Receiver:

在起始位置設置廣播群組

- 將廣播群組設爲 6 拖入當啓動時
- 最初，汽車默認向前移動

通過接收不同數字來控制車輛

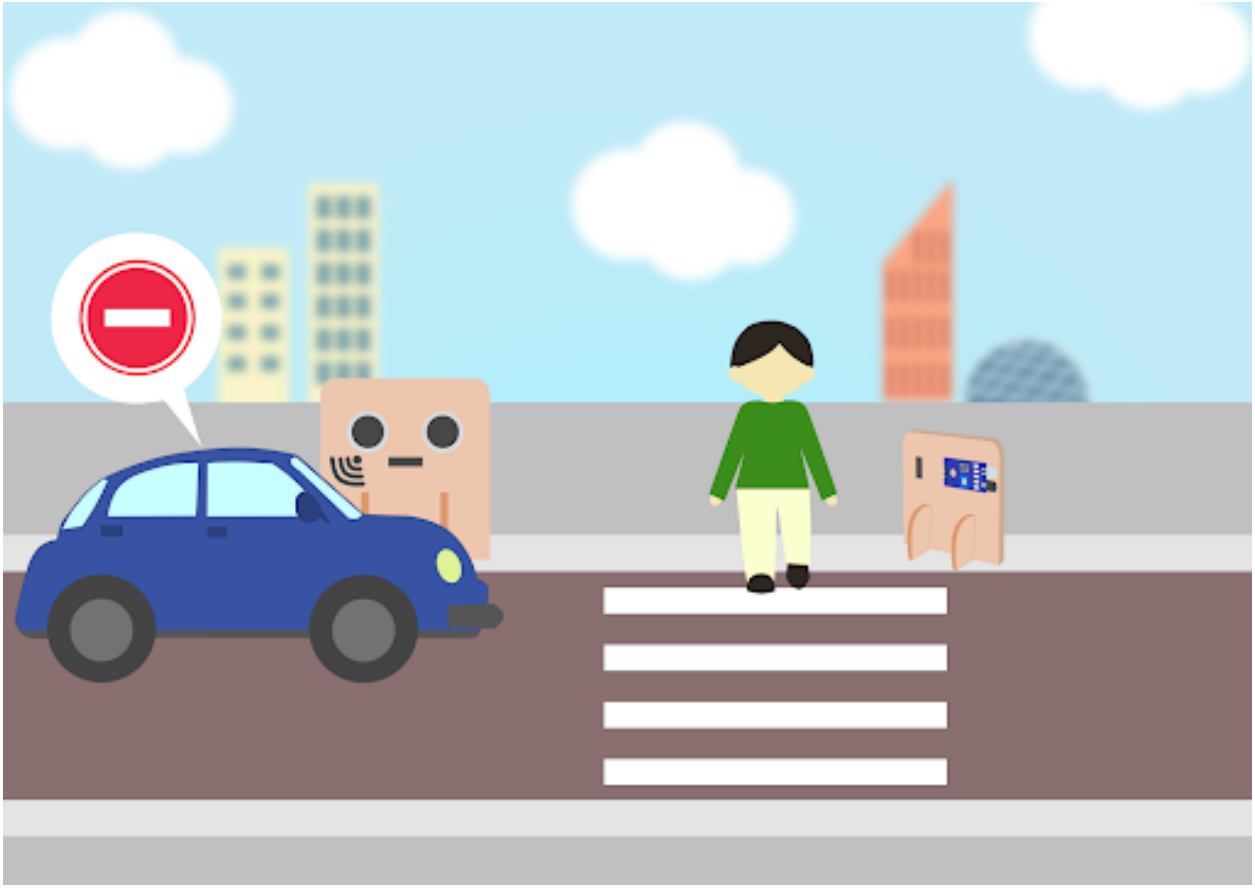
- 將如果語句放入當收到廣播數字 receivedNumber
- 設 receivedNumber =2，Ⓕ讓車向左移動。
- 設 receivedNumber=0，Ⓕ讓車向前移動。

當收到廣播數字 receivedNumber
如果 receivedNumber = 2 那麼
馬達 M1 速度 150 M2 速度 0
馬達 M3 速度 0 M4 速度 150
如果 receivedNumber = 0 那麼
馬達 M1 速度 150 M2 速度 150
馬達 M3 速度 150 M4 速度 150

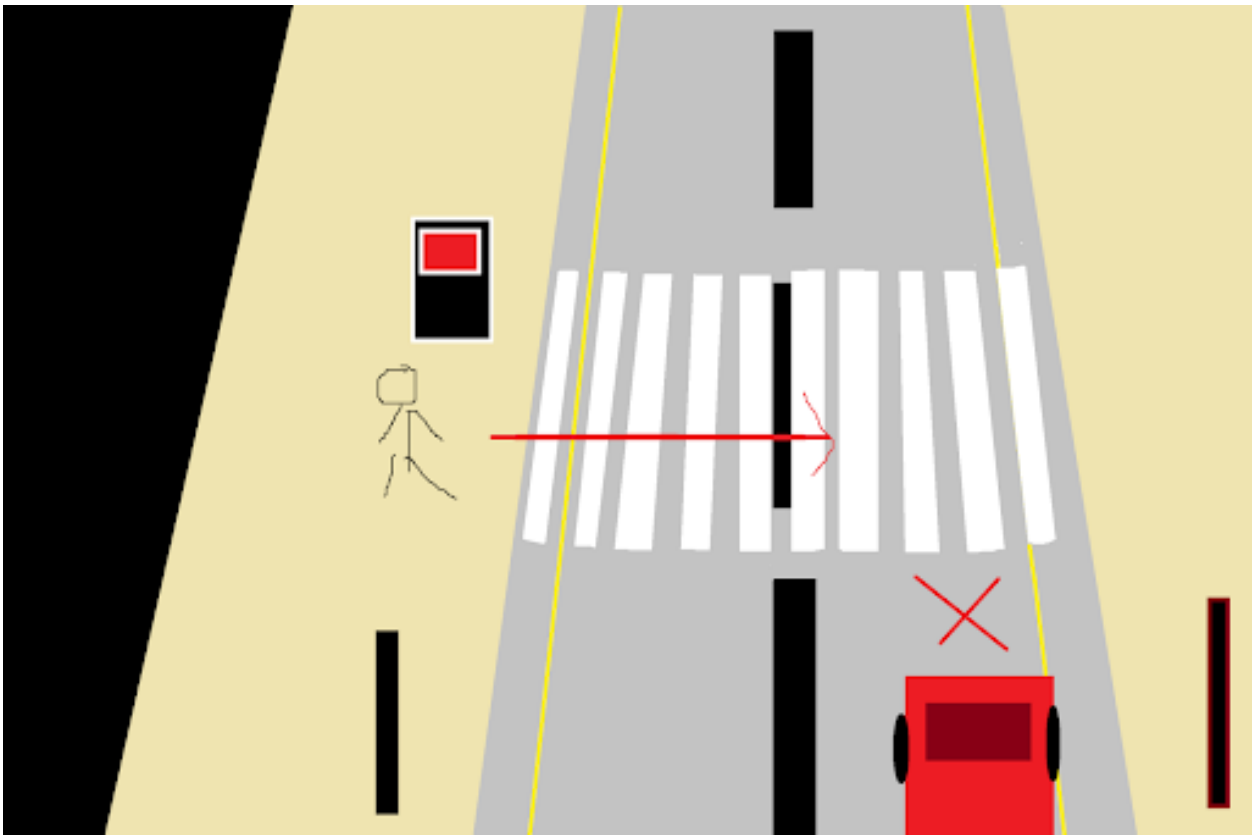
總結

思考

1.2.8 斑馬線



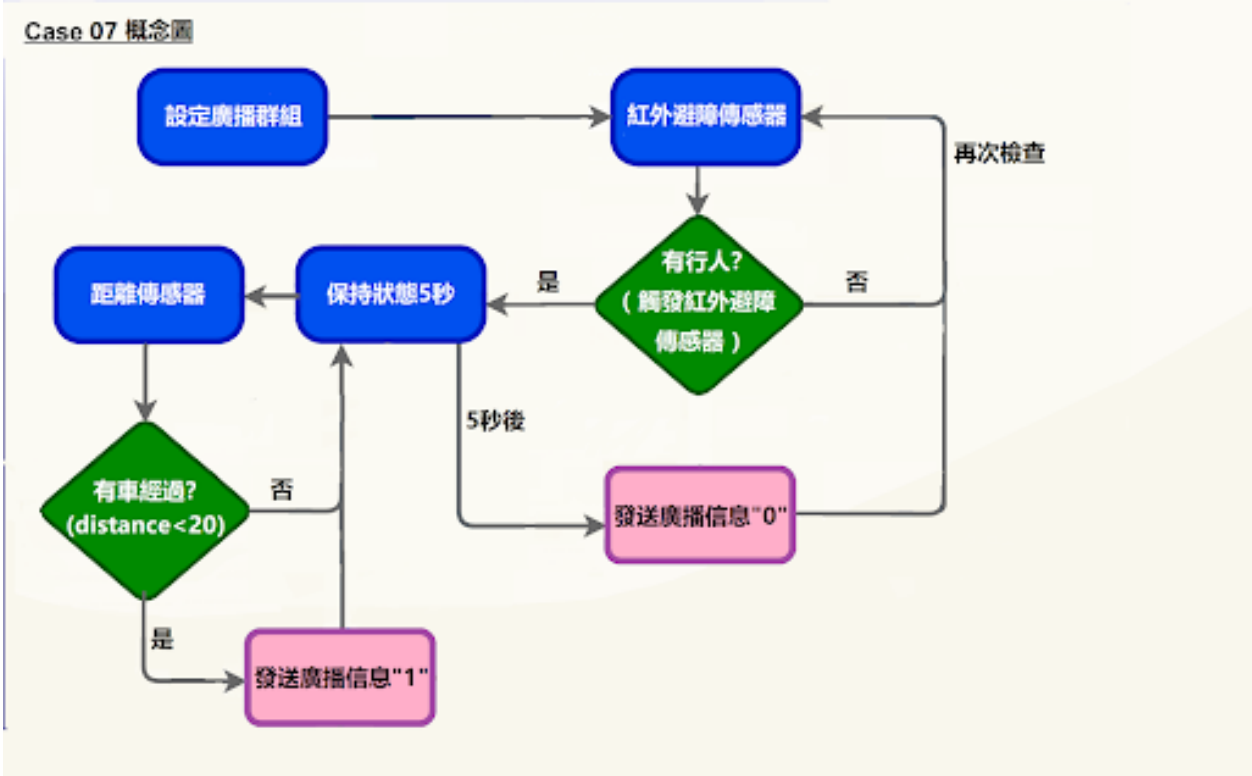
目標



背景

什麼是智能斑馬線系統？

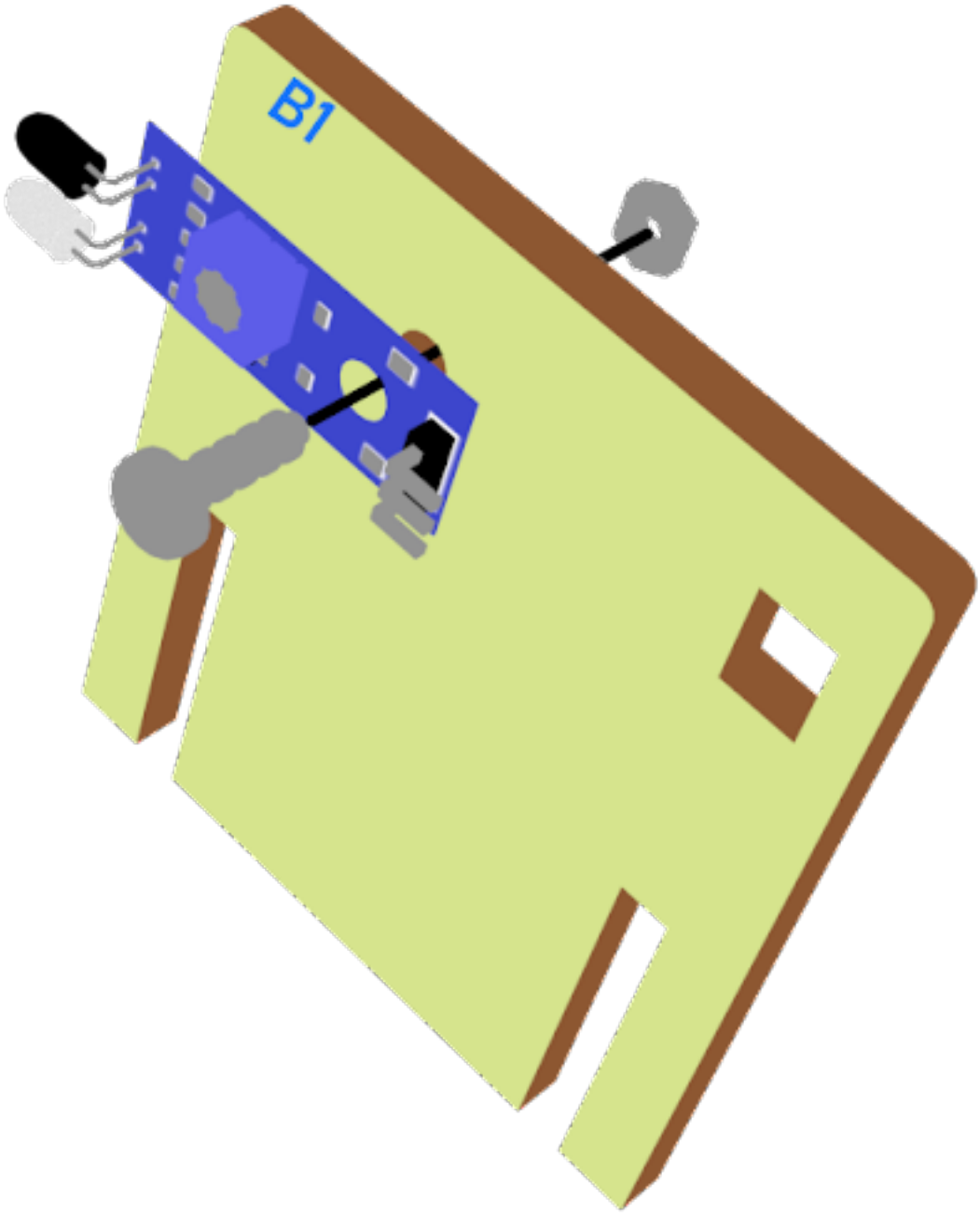
智能斑馬線運作



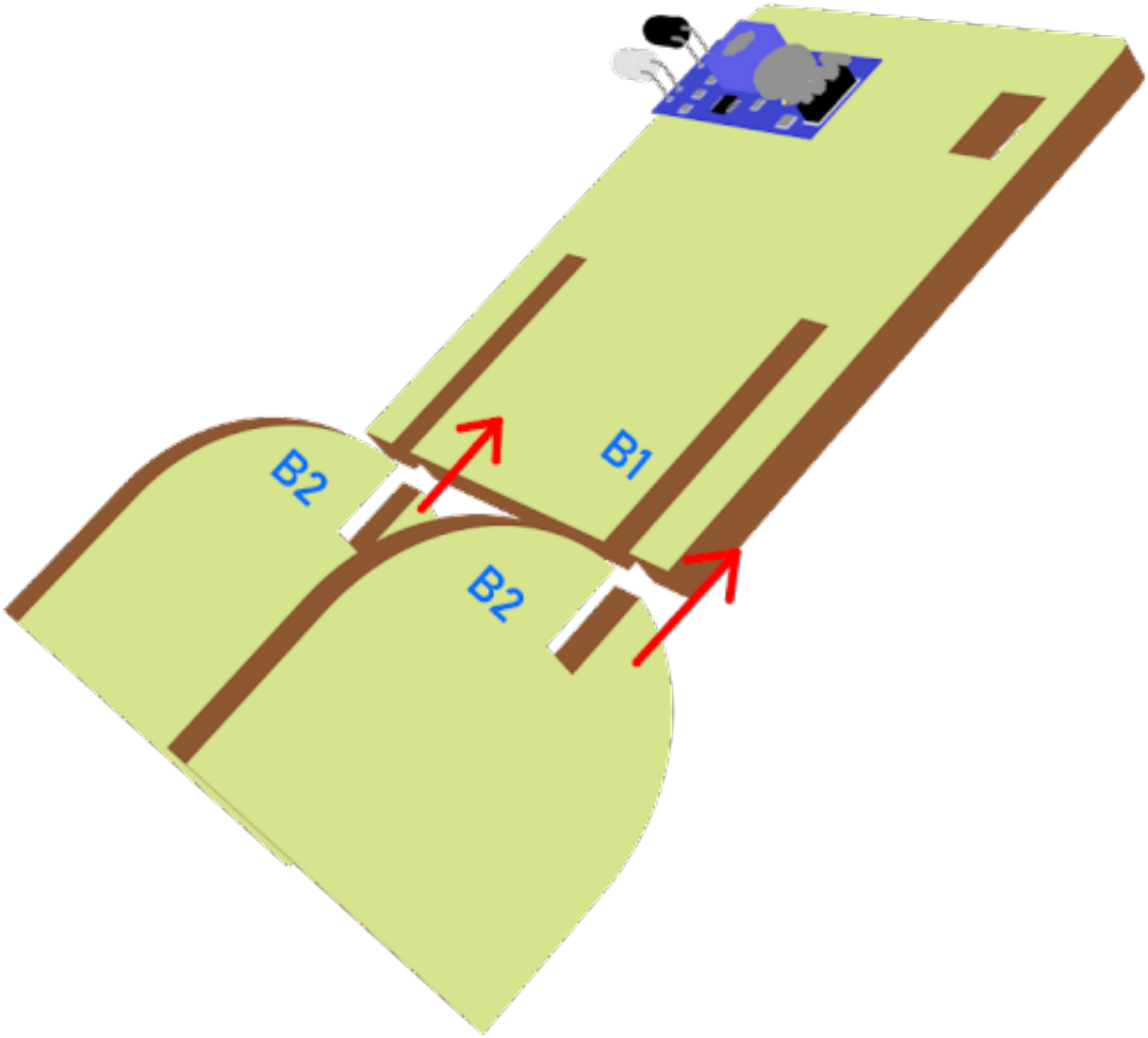
材料準備

組裝步驟

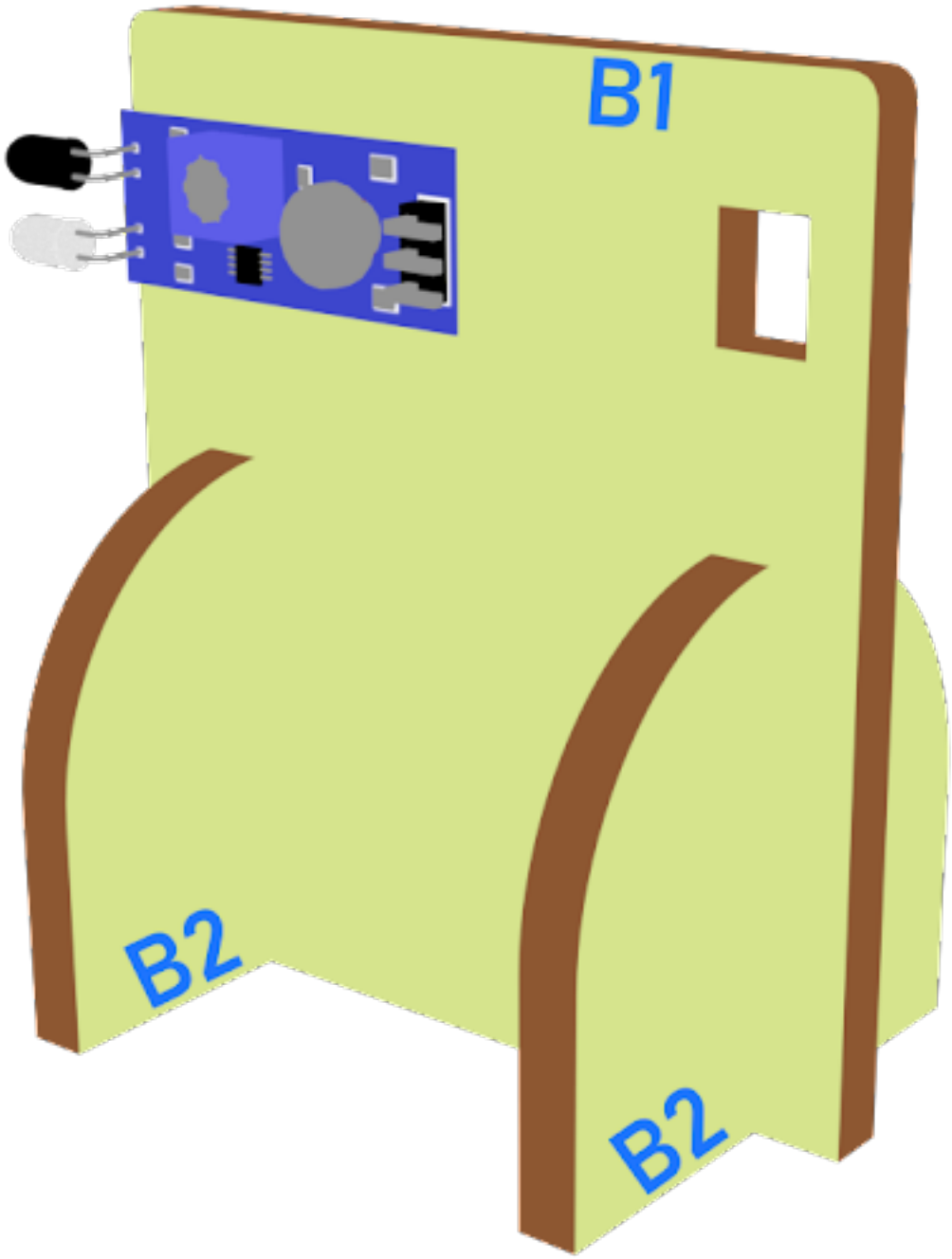
第一步



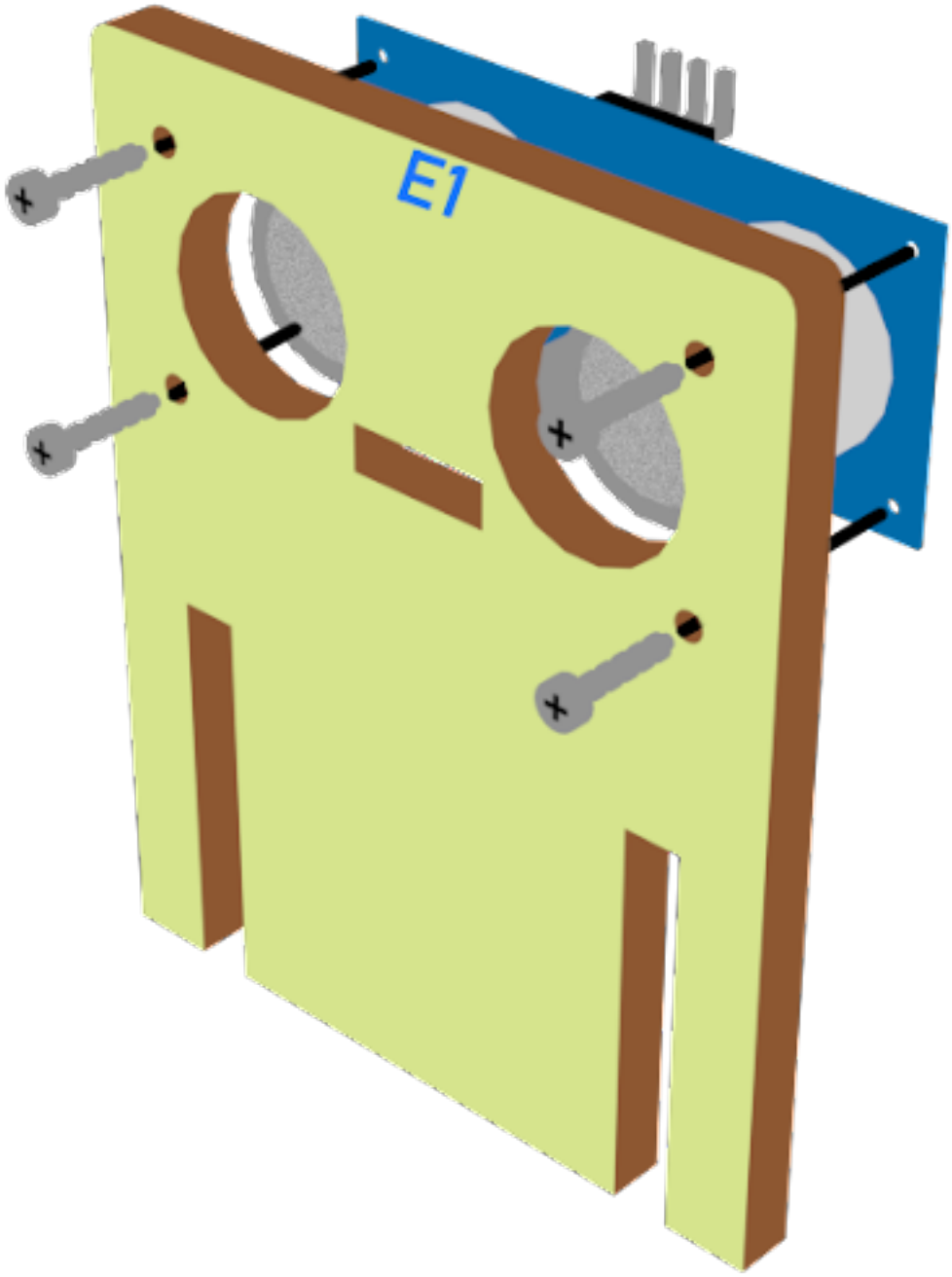
第三步



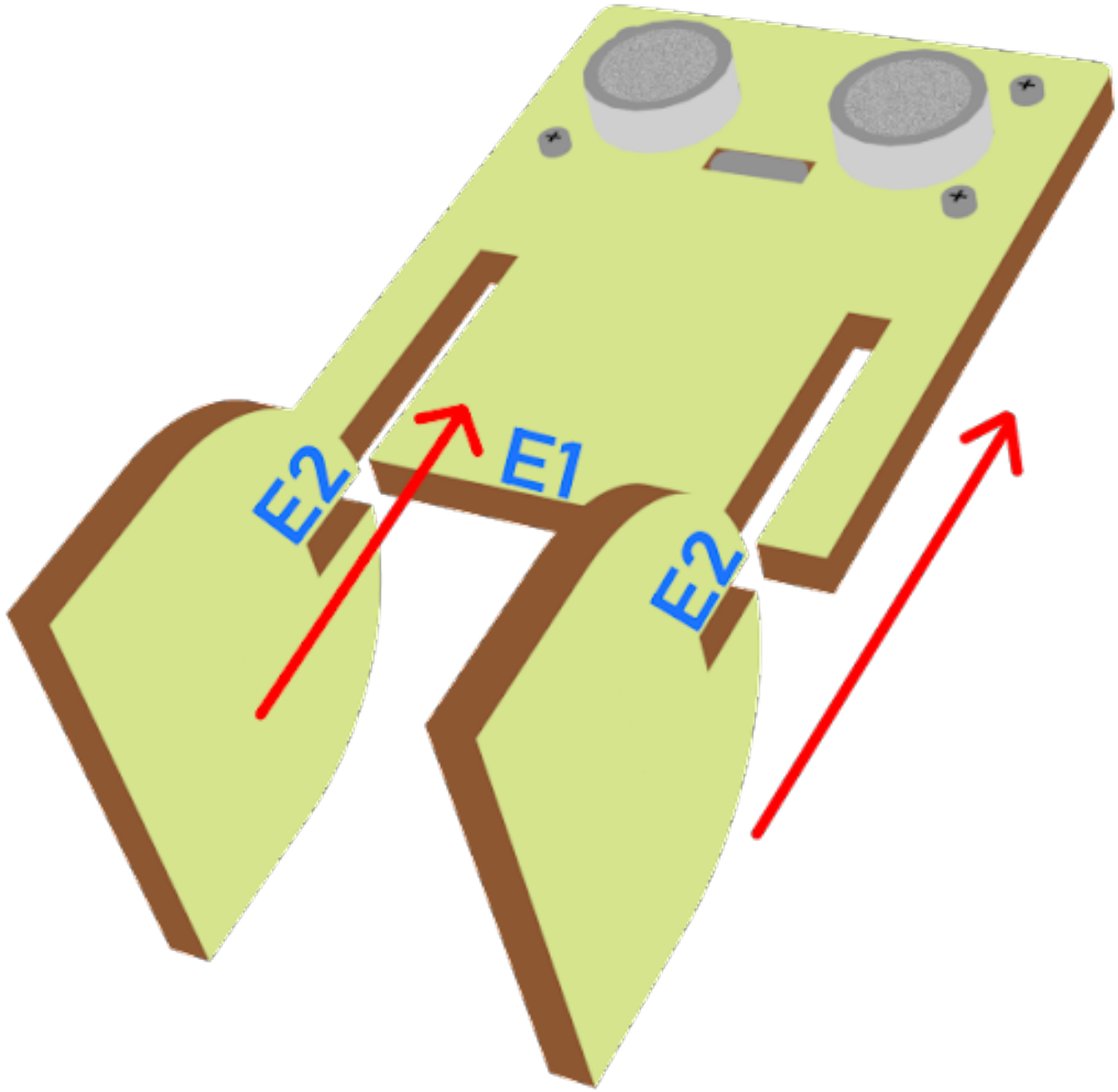
第三步



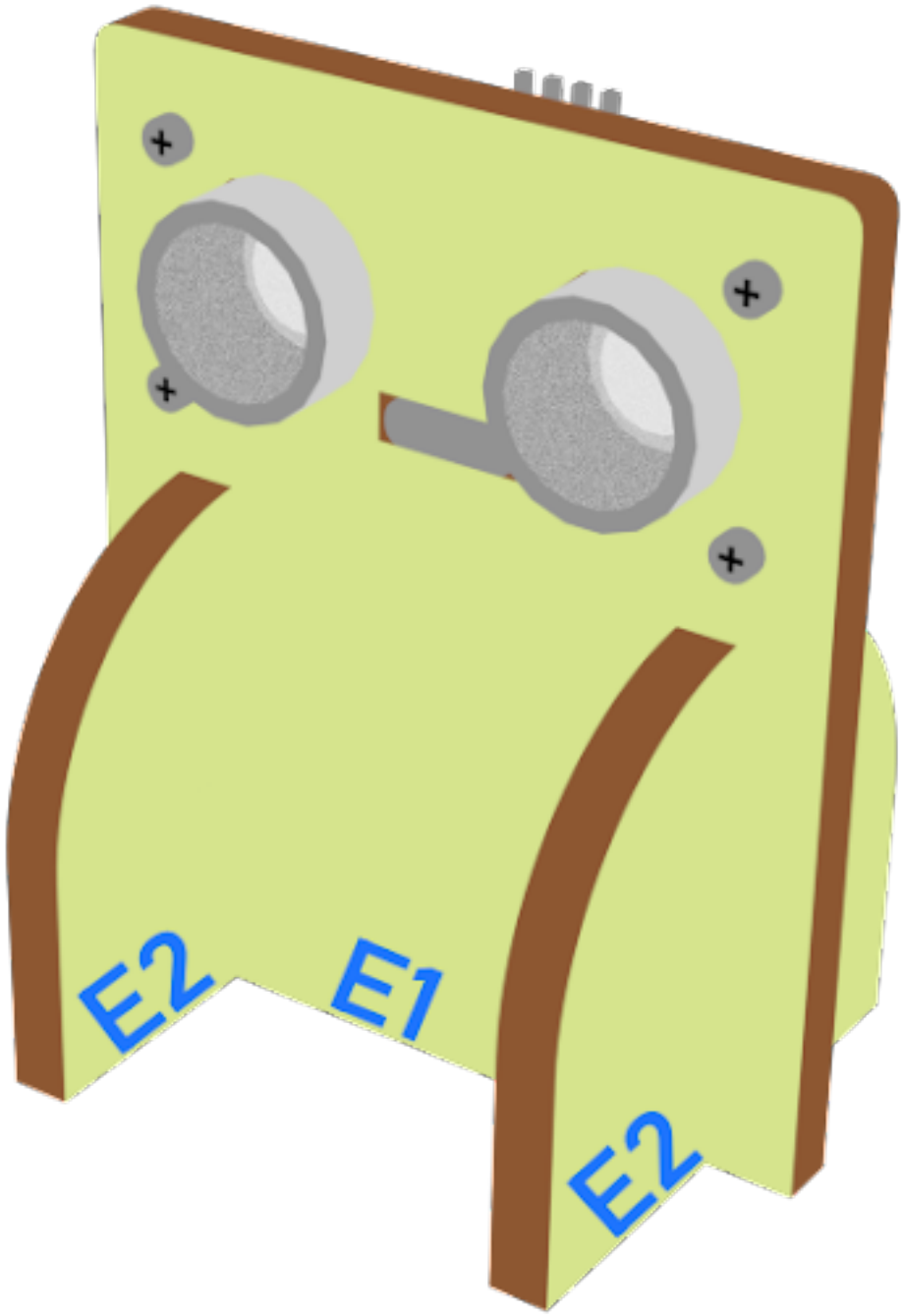
第四步



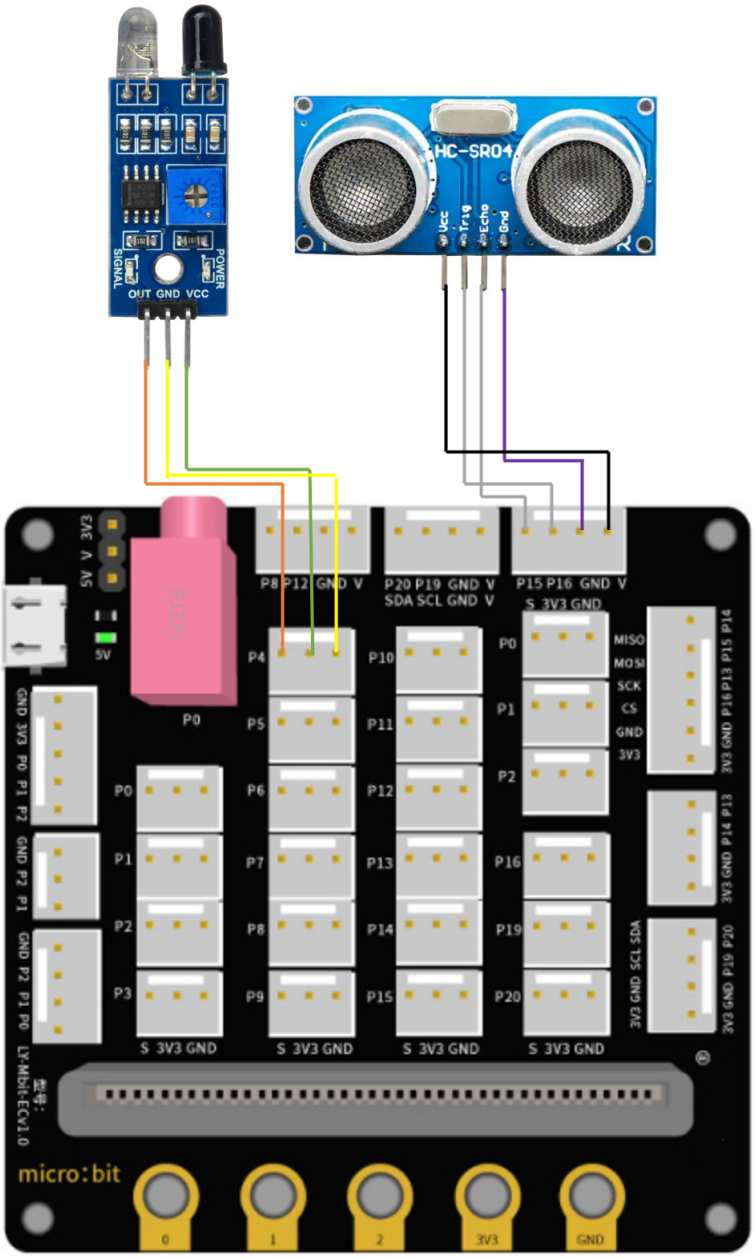
第五步

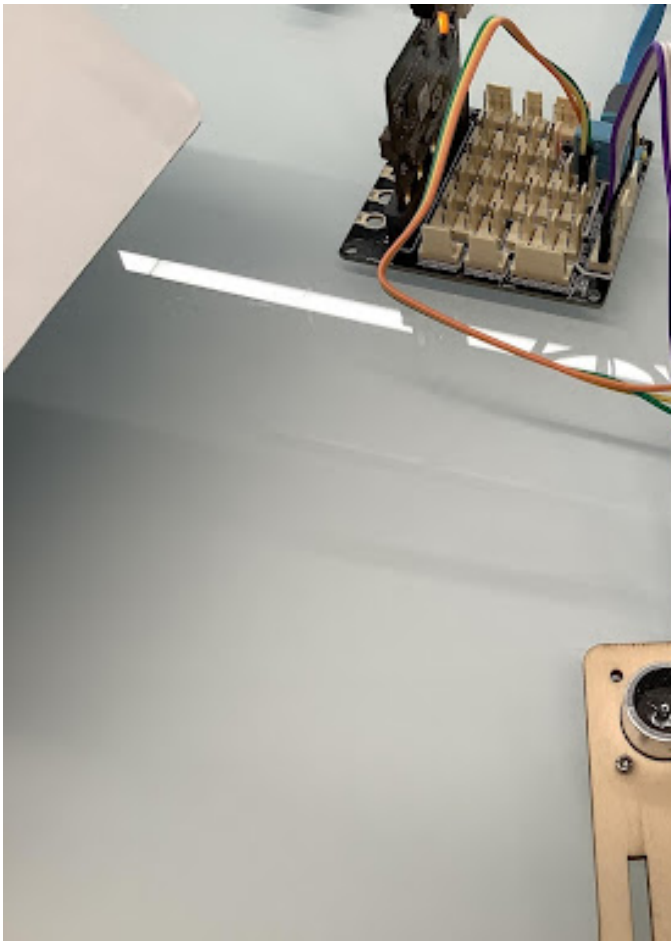


第六步



硬件連接





編程

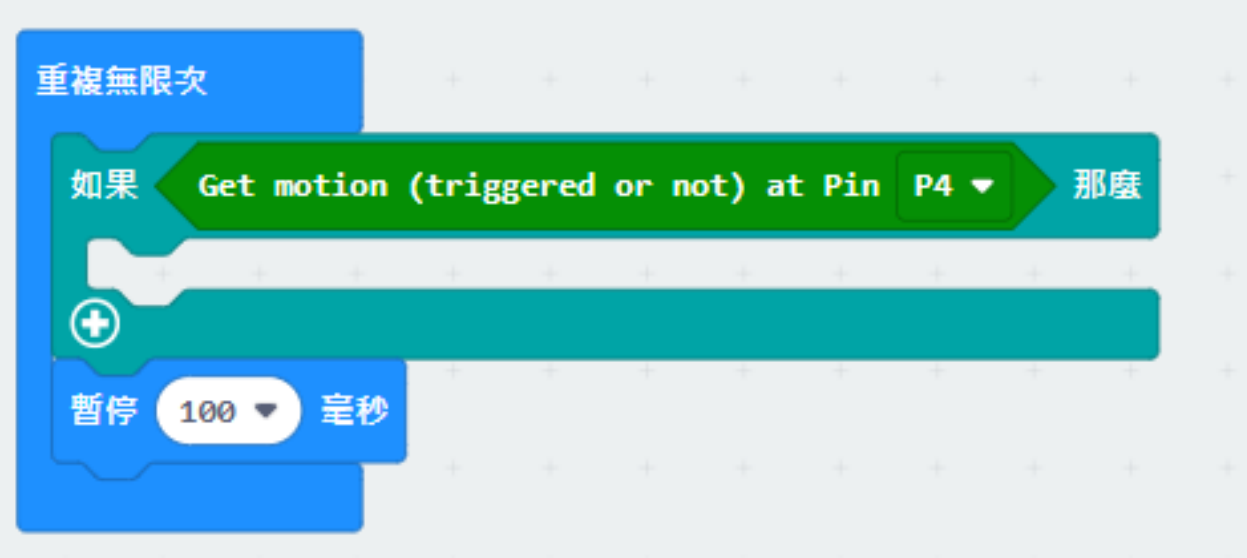
Sender:

在起始位置設置 radio group

- 將廣播群組設爲 6 拖入當啓動時

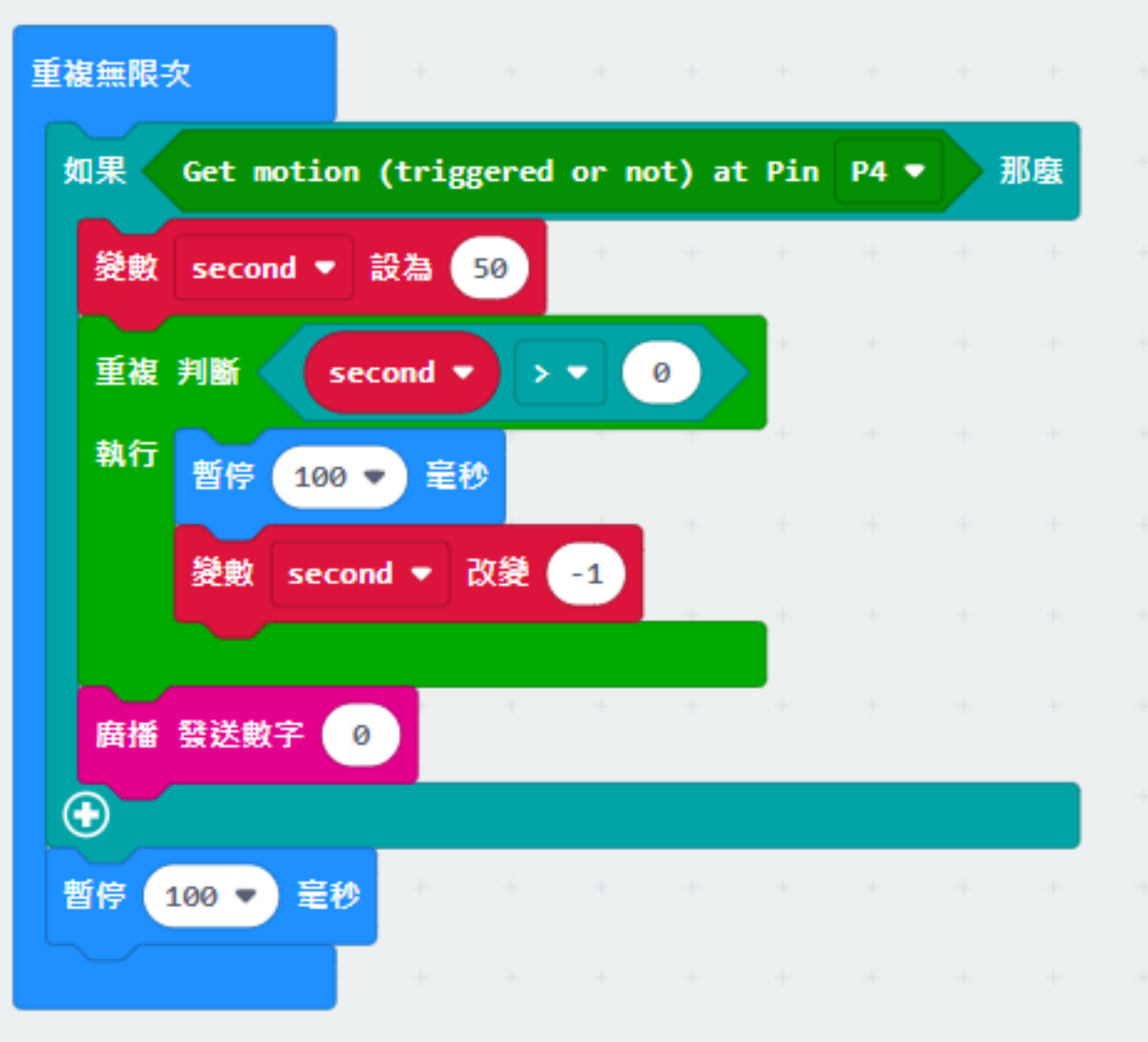
當觸發紅外傳感器

- 將 get motion (triggered or not) at pin P4 的如果句式放入重Ⓕ無限次
- 把暫停拖入循環，Ⓕ下一次檢查等待 0.1 秒。



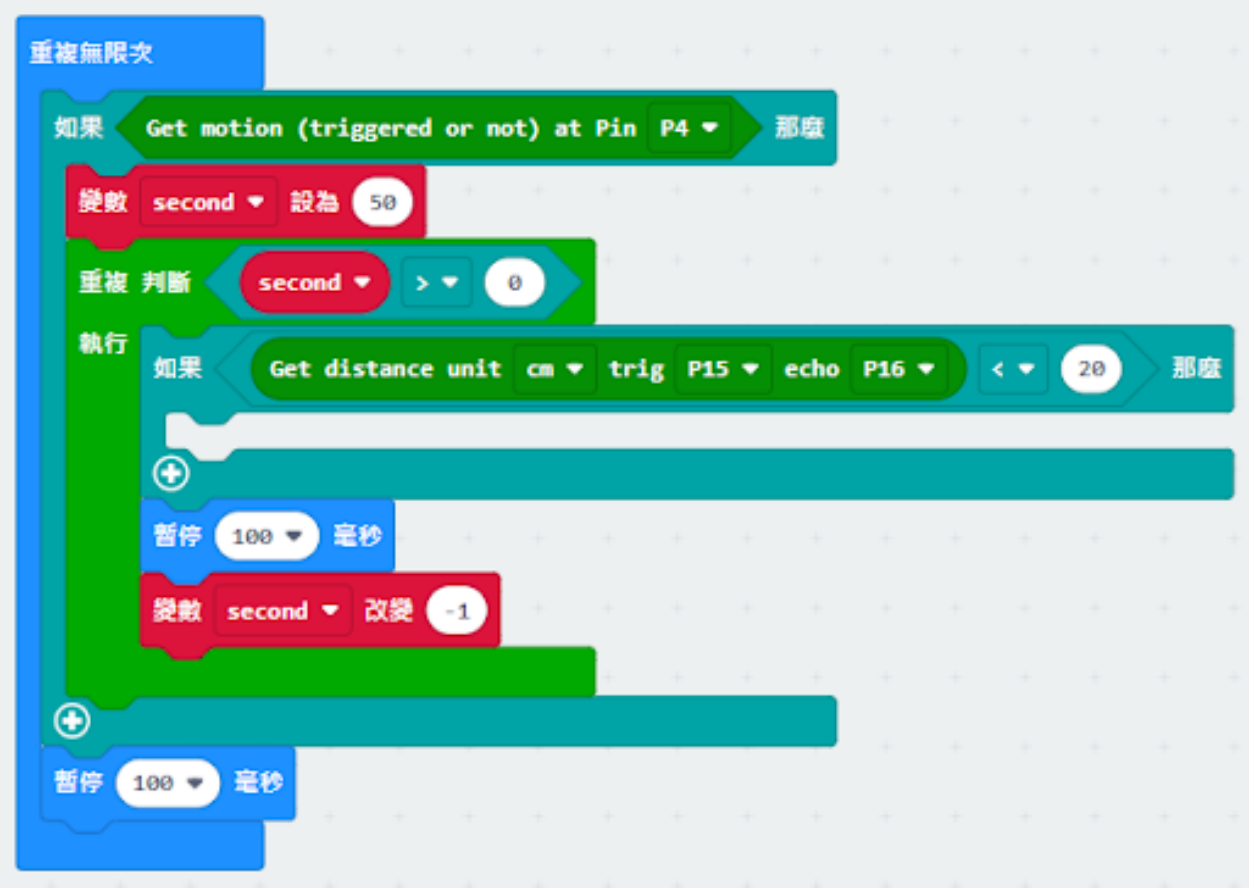
保留觸發狀態五秒

- 設定變數 second Ⓕ 50
- 當 second 大於 0，停止 0.1 秒，Ⓕ改變 second Ⓕ少 1



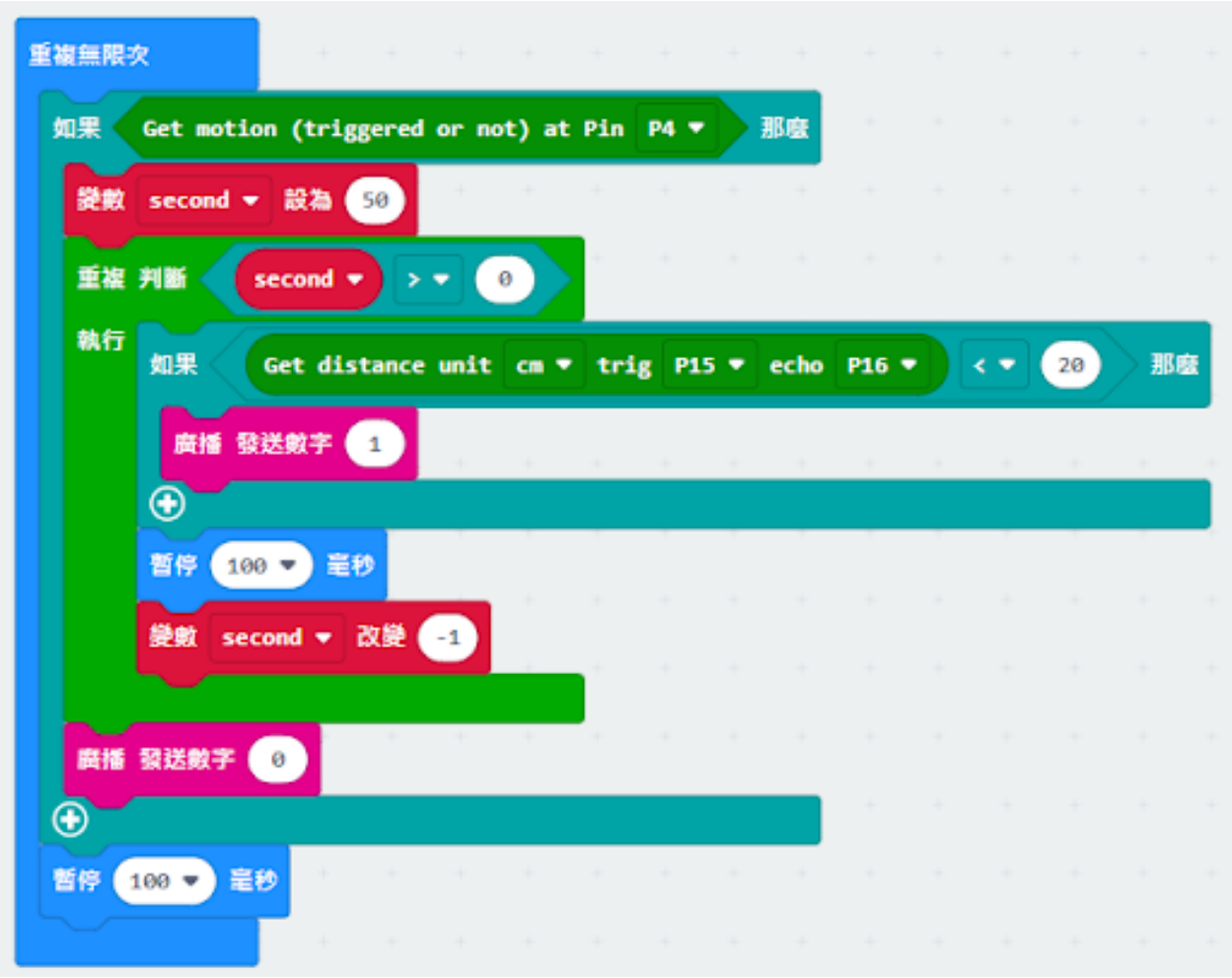
獲得距離數值

- 將 get distance unit cm trig P15 echo P16 < 20 的如果句式拖入循環中



發送廣播數字以控制車輛

- 將廣播發送數字 1 拖入 if 句式
- 將廣播發送數字 0 拖到循環外



Receiver:

起始位置設置廣播群組

- 將廣播群組設爲 6 拖入當啓動時
- 最初，汽車默認向前移動

通過接收不同數字來控制車輛

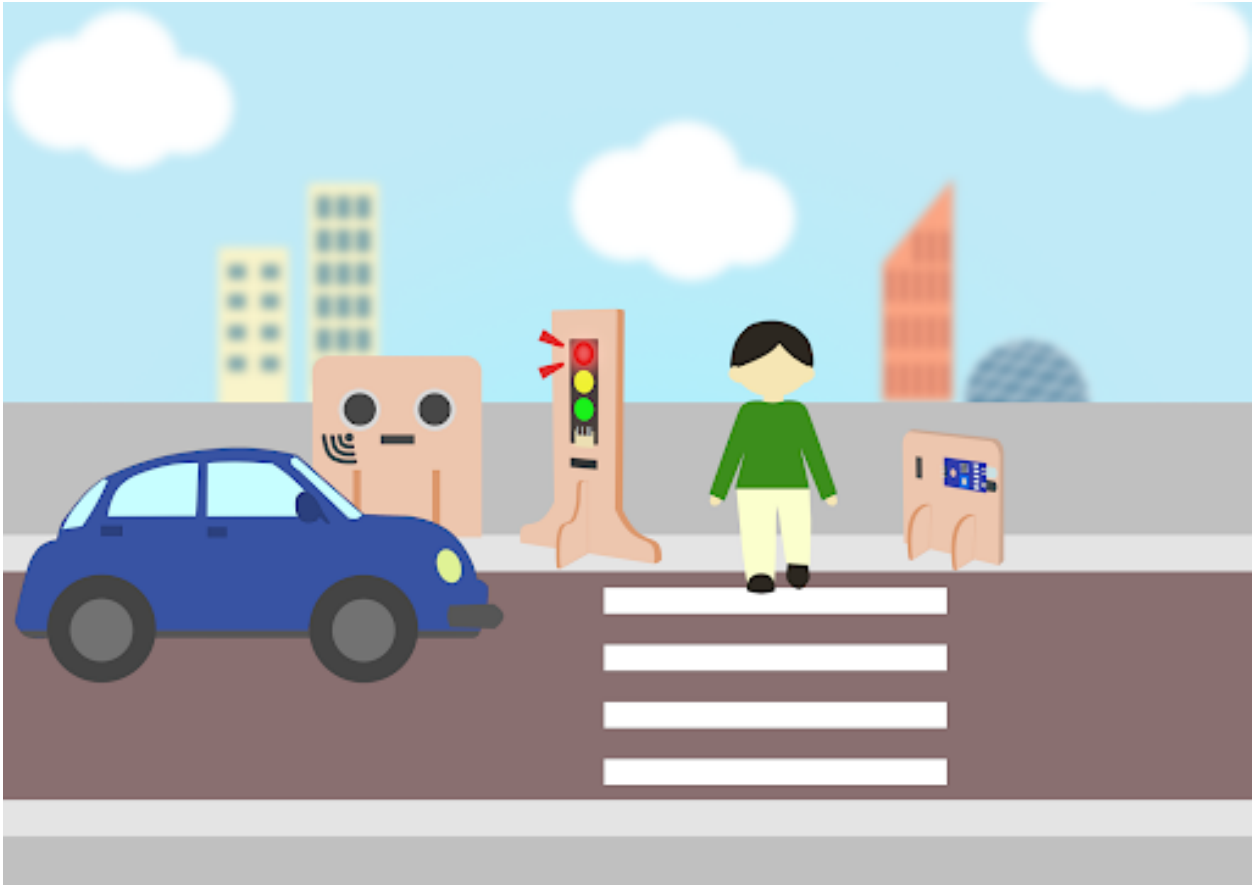
- 將如果語句放入當收到廣播數字 receivedNumber
- 設 receivedNumber =1，Ⓕ控制剎車。
- 設 receivedNumber=0，Ⓕ讓車向前移動。

當收到廣播數字 receivedNumber
如果 receivedNumber = 1 那麼
馬達 M1 速度 0 M2 速度 0
馬達 M3 速度 0 M4 速度 0
如果 receivedNumber = 0 那麼
馬達 M1 速度 150 M2 速度 150
馬達 M3 速度 150 M4 速度 150

總結

思考

1.2.9 智能交通燈

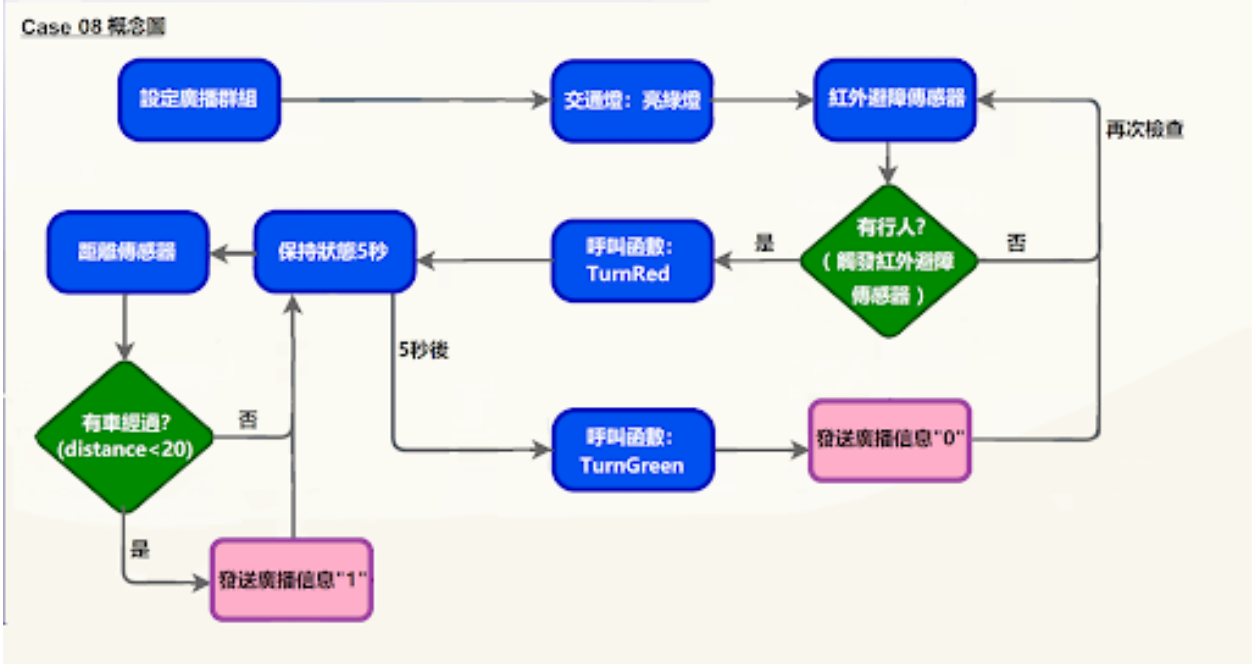


目標

背景

什麼是智能交通燈？

智能交通燈運作

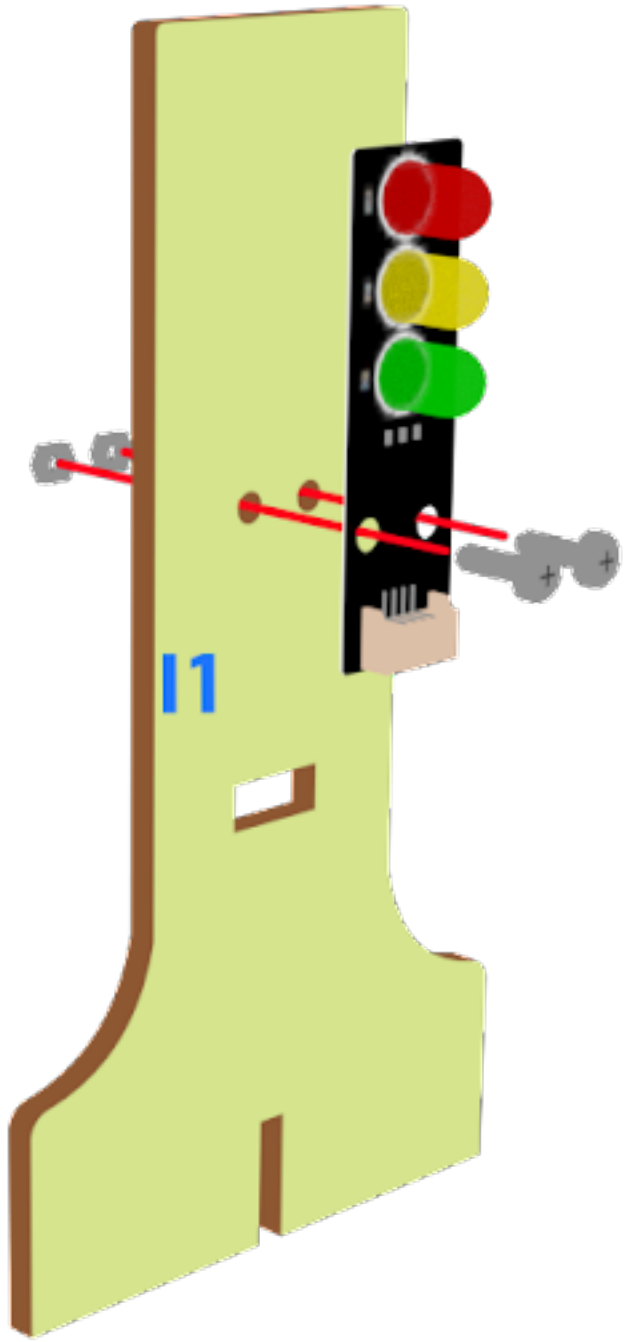

Case 08 概念圖
設定廣播群組
交通燈：亮綠燈
紅外避障傳感器
再次檢查
有行人?
（觸發紅外避障
傳感器）
是
否
呼叫函數:
TurnRed
保持狀態5秒
距離傳感器
5秒後
有車經過?
(distance<20)
否
是
呼叫函數:
TurnGreen
發送廣播信息"0"
發送廣播信息"1"

函數：轉紅燈
紅綠燈模塊
暫停一秒
紅綠燈模塊
暫停一秒
紅綠燈模塊
函數：轉綠燈
紅綠燈模塊
暫停一秒
紅綠燈模塊
暫停一秒
紅綠燈模塊

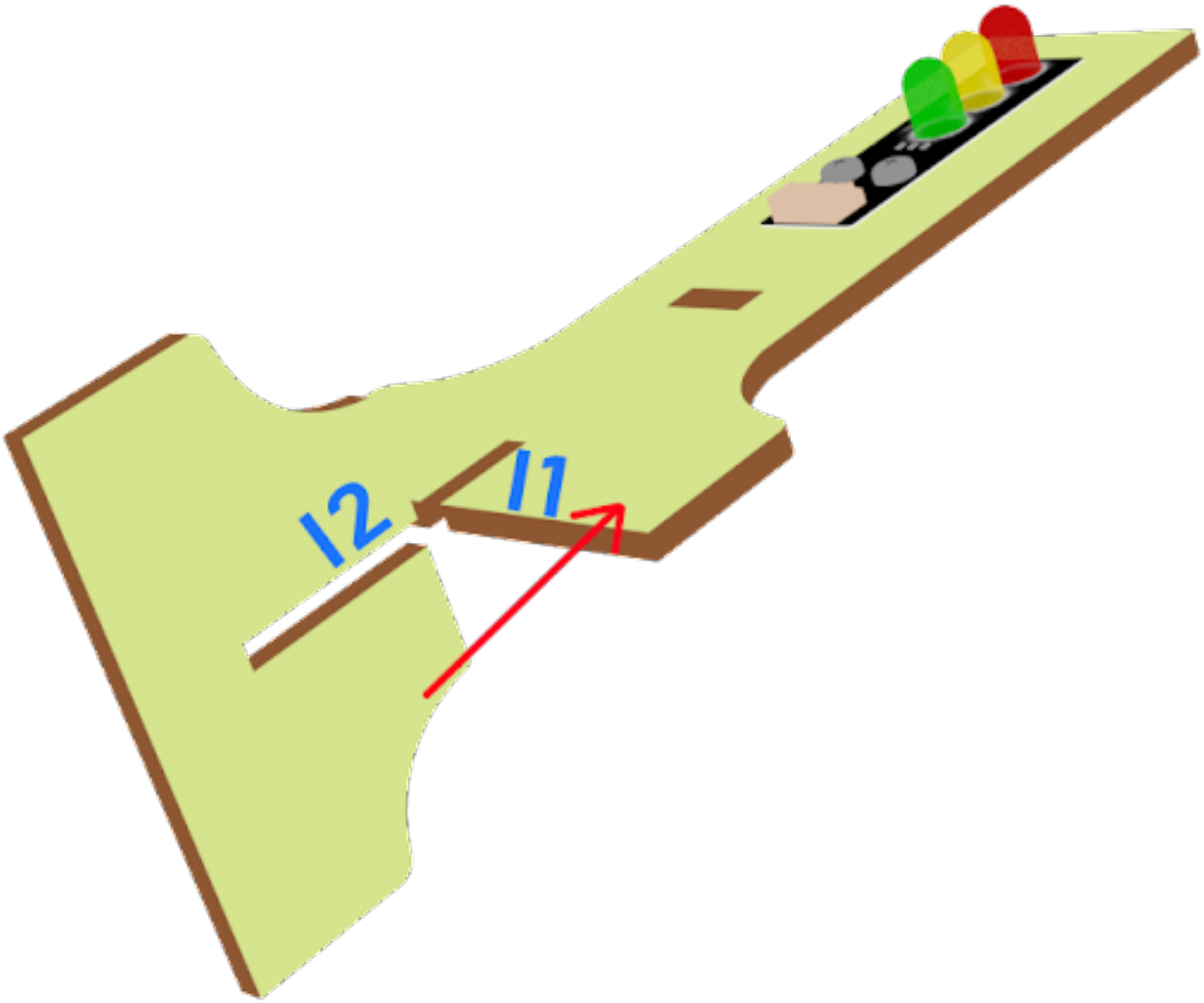
材料準備

組裝步驟

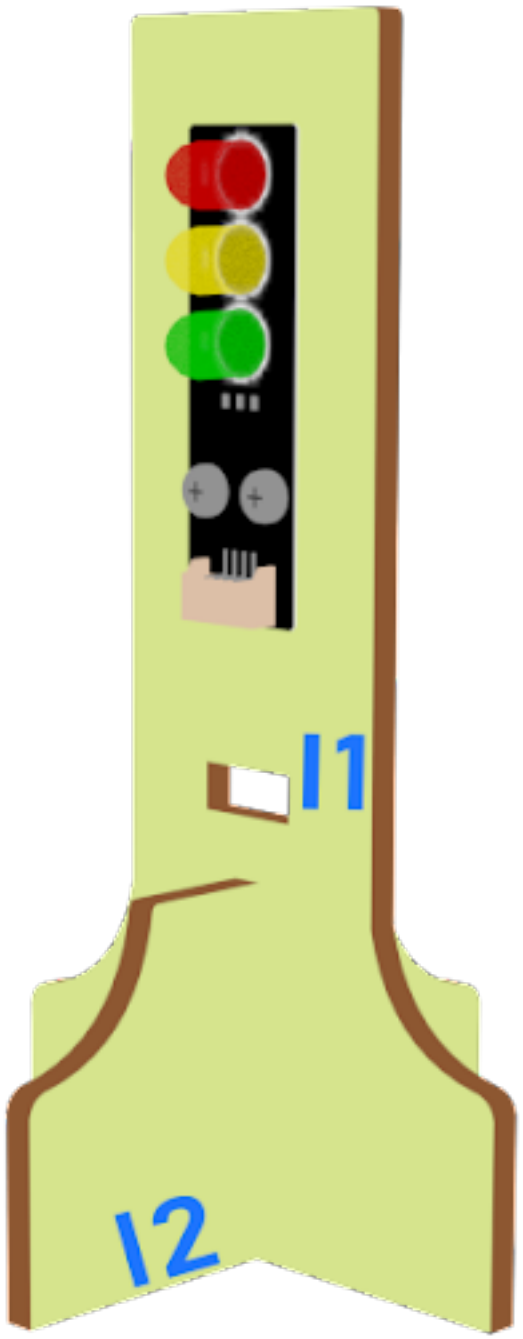
第一步



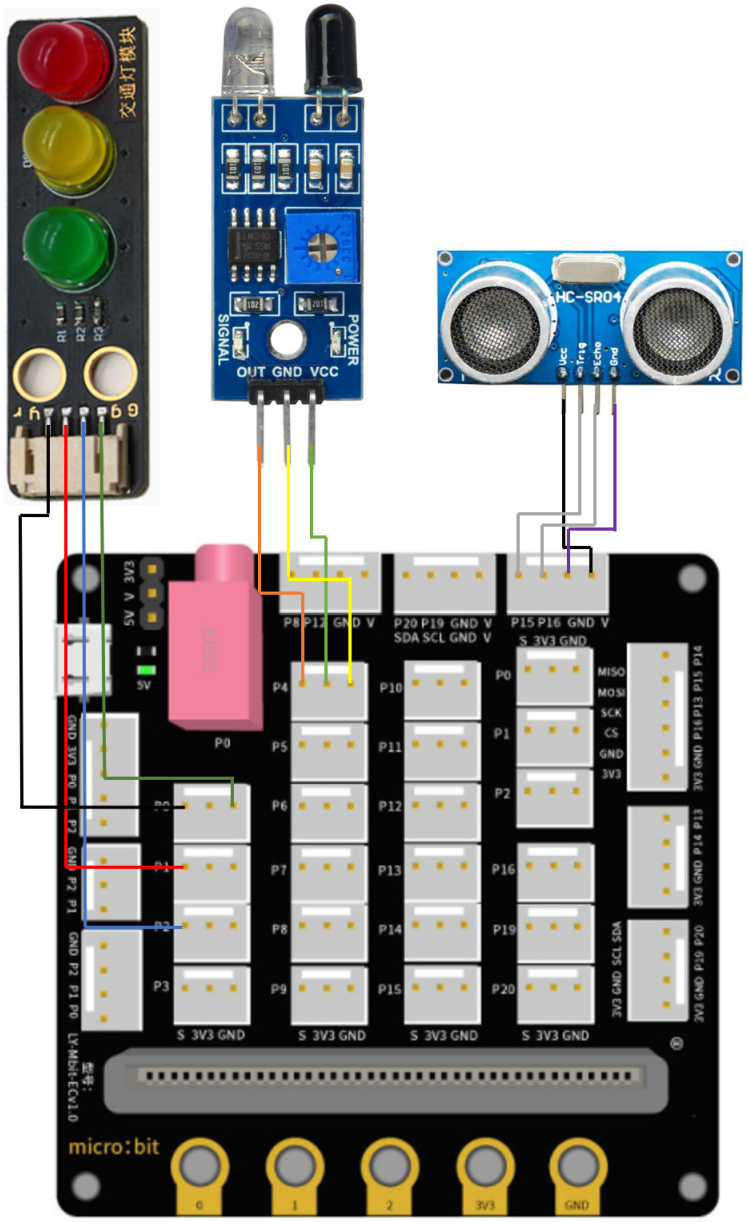
第二步

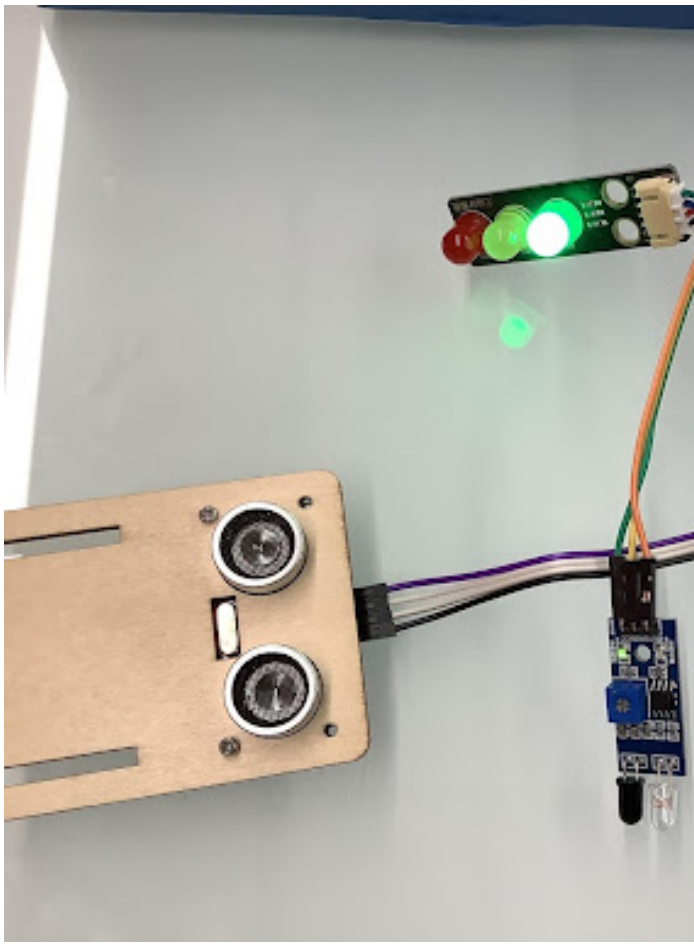


第三步



硬件連接





編程

Sender:

設置一個新的功能（TurnRed）。

- 暫停一秒
- 控制F色交通燈亮起

- 暫停一秒
- 控制紅色交通燈亮起

設置一个新的功能（TurnGreen）。

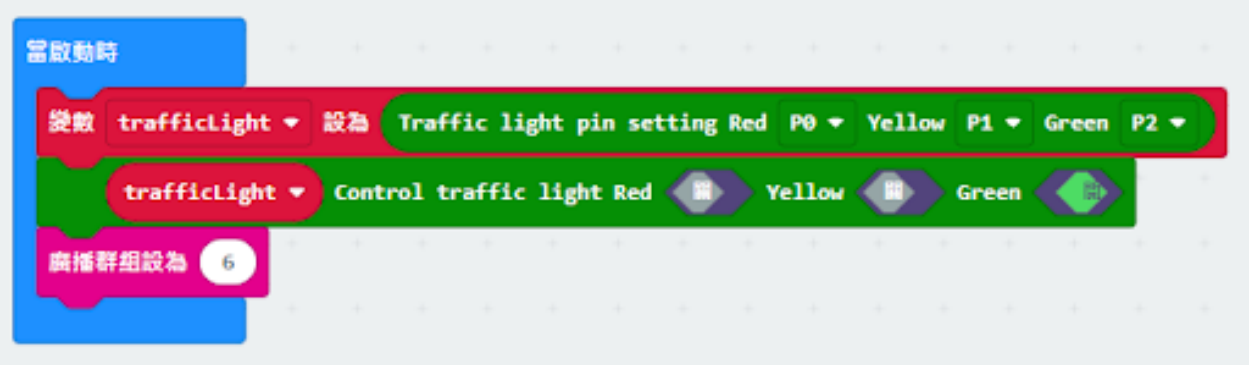
- 暫停一秒
- 控制F色交通燈亮起
- 暫停一秒
- 控制F色交通燈亮起

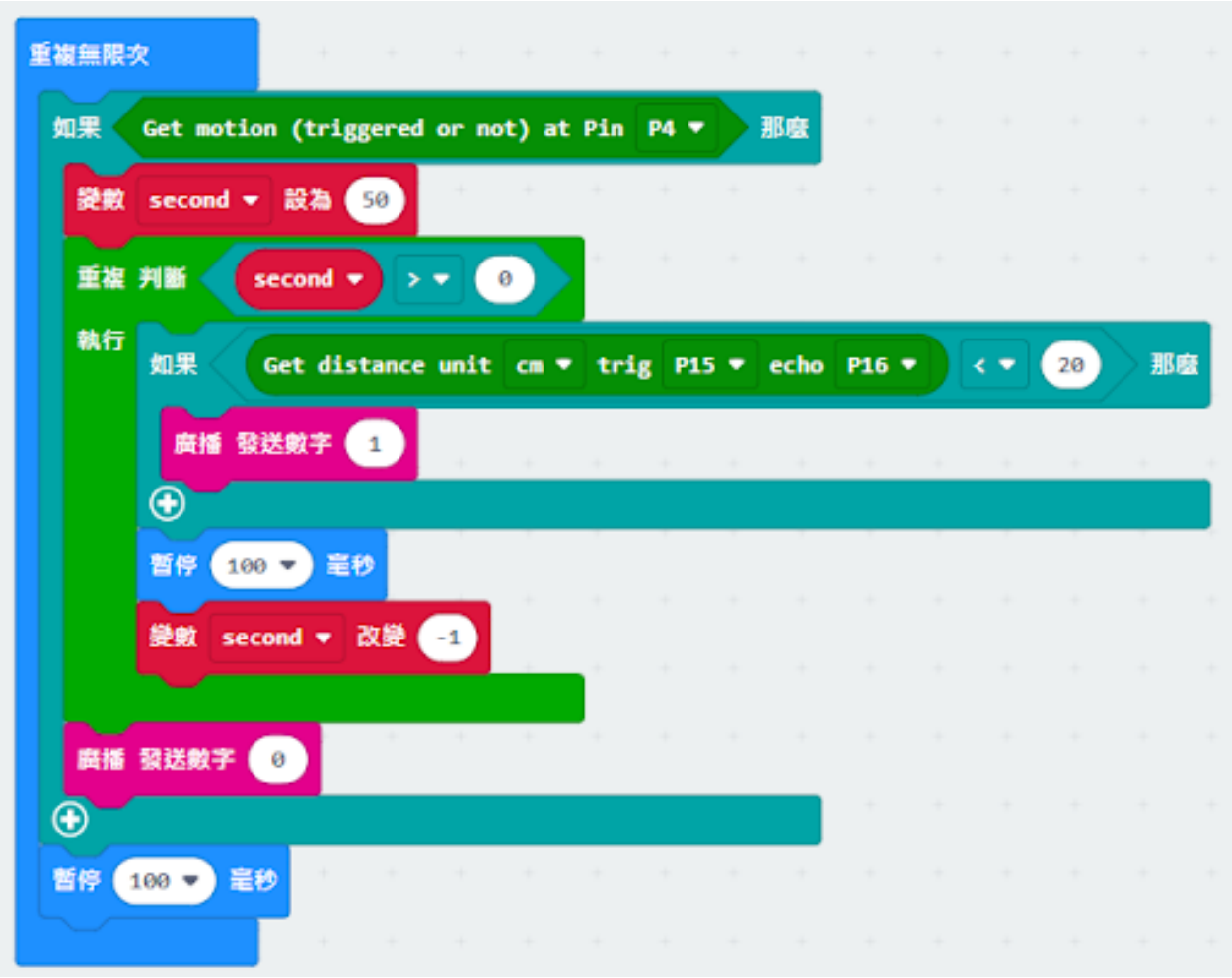
第一步停用 micro:bit LED。

- 將積木啓用設爲 false 拖放到當啓動時
- 注意，P3 在默认设置中被用作 LED，因此 LED 需要被禁用。

像上一課一樣初始化程序

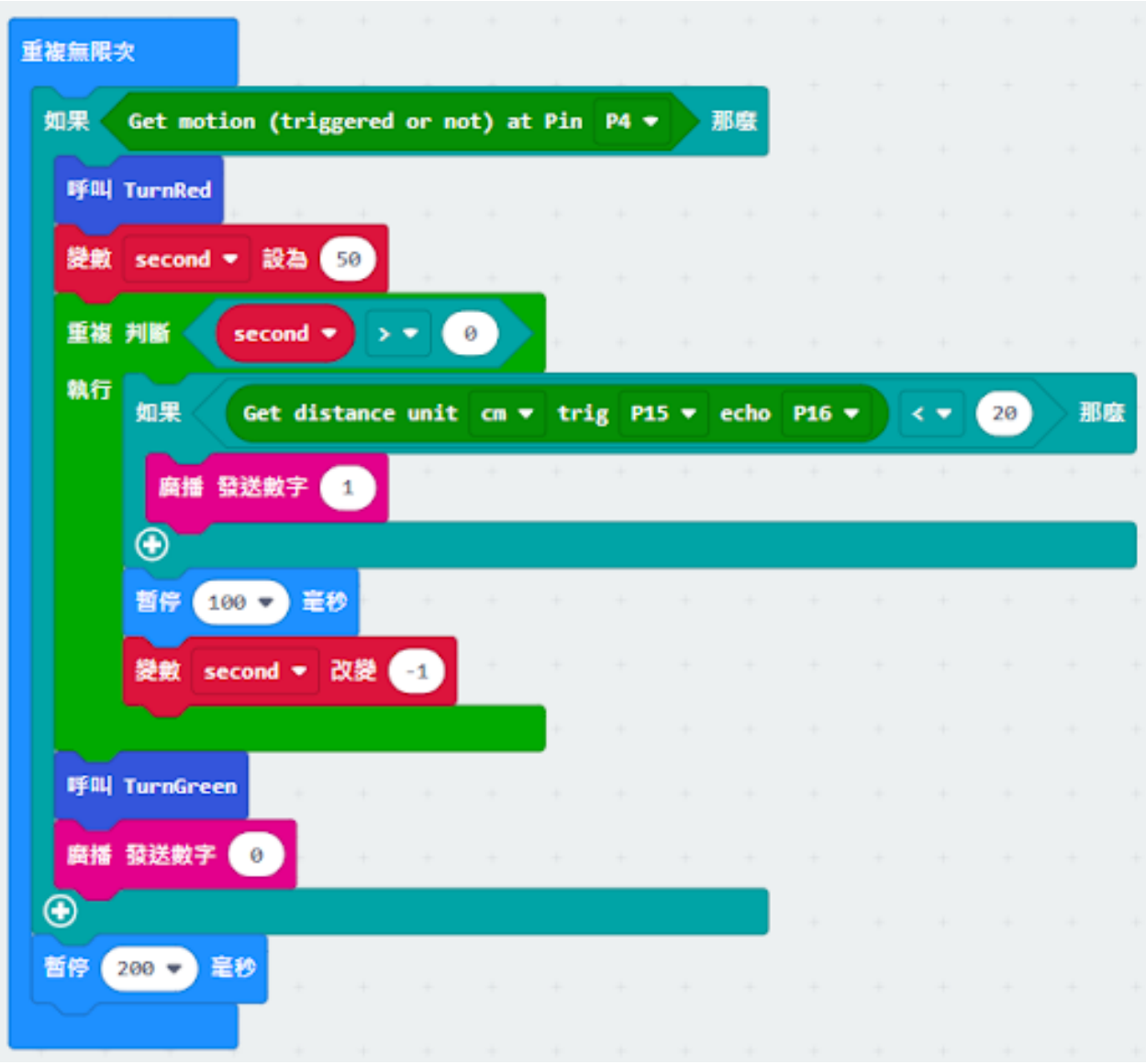
- 將 trafficLight to Traffic light pin setting Red P0 Yellow P1 Green P2 拖入當啓動時
- 控制F色交通燈亮起
- 將廣播群組設爲 6 拖入當啓動時





呼叫函數

- 將呼叫 TurnRed 拖入 get motion (triggered or not) at pin P4 的如果句式中
- 將呼叫 TurnGreen 拖到循環后



Receiver:

在起始位置設置廣播群組

- 將廣播群組設爲 6 拖入當啓動時
- 最初，汽車默認向前移動

通過接收不同數字來控制車輛

- 將如果語句放入當收到廣播數字 receivedNumber
- 設 receivedNumber =1，Ⓕ控制剎車。
- 設 receivedNumber=0，Ⓕ讓車向前移動。

當收到廣播數字 receivedNumber
如果 receivedNumber = 1 那麼
馬達 M1 速度 0 M2 速度 0
馬達 M3 速度 0 M4 速度 0
如果 receivedNumber = 0 那麼
馬達 M1 速度 150 M2 速度 150
馬達 M3 速度 150 M4 速度 150

總結

思考

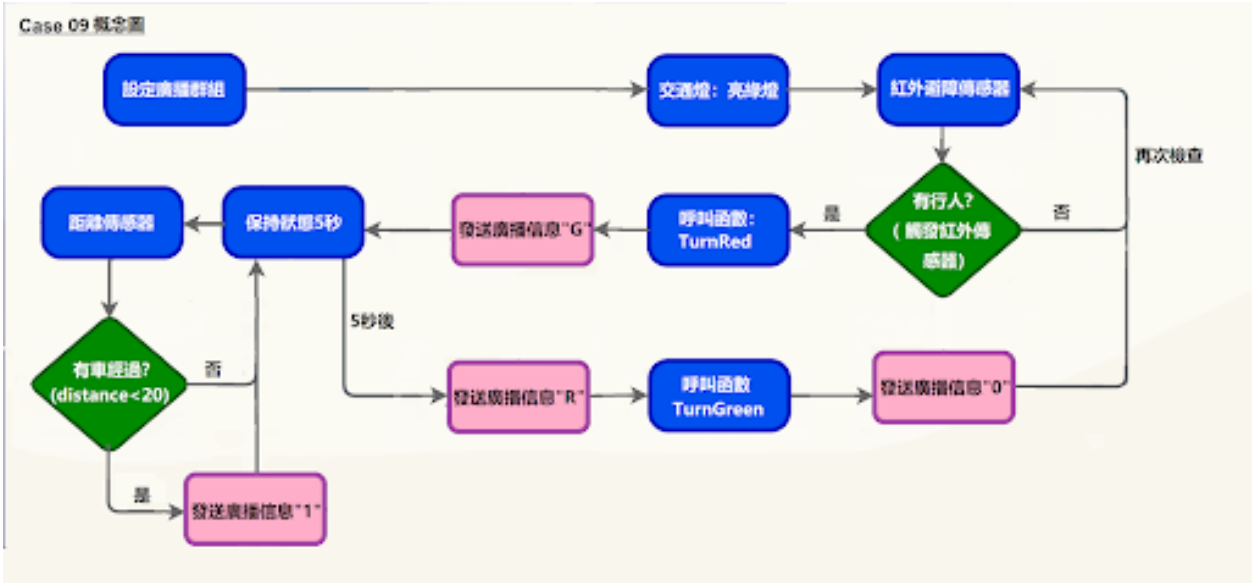
1.2.10 行人過路交通燈二

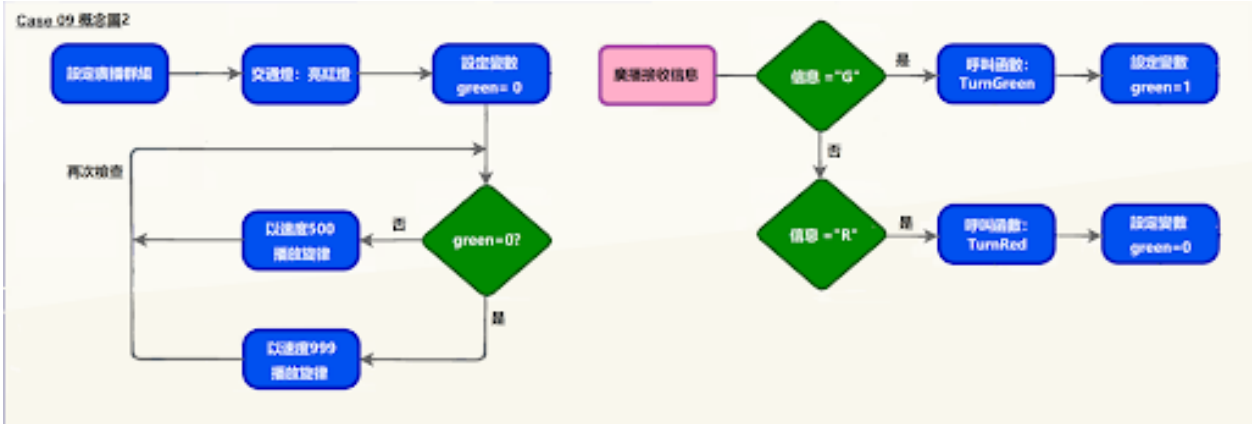
目標

背景

什麼是智能行人過路燈？

智能交通燈運作

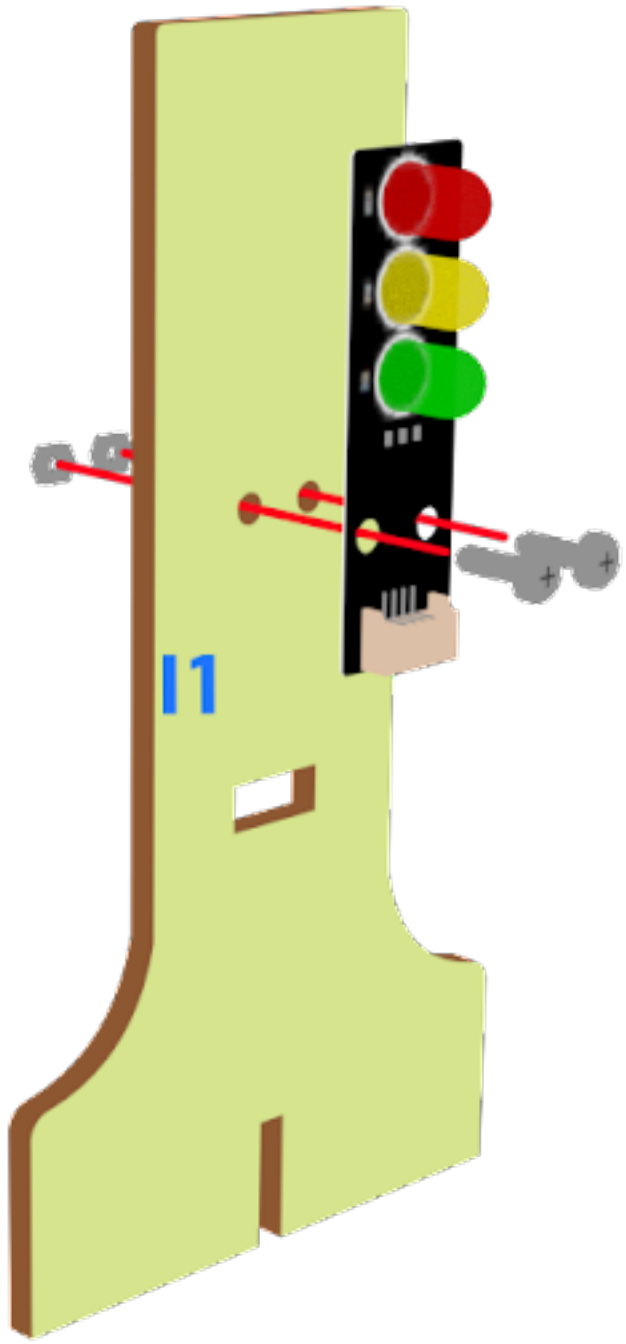




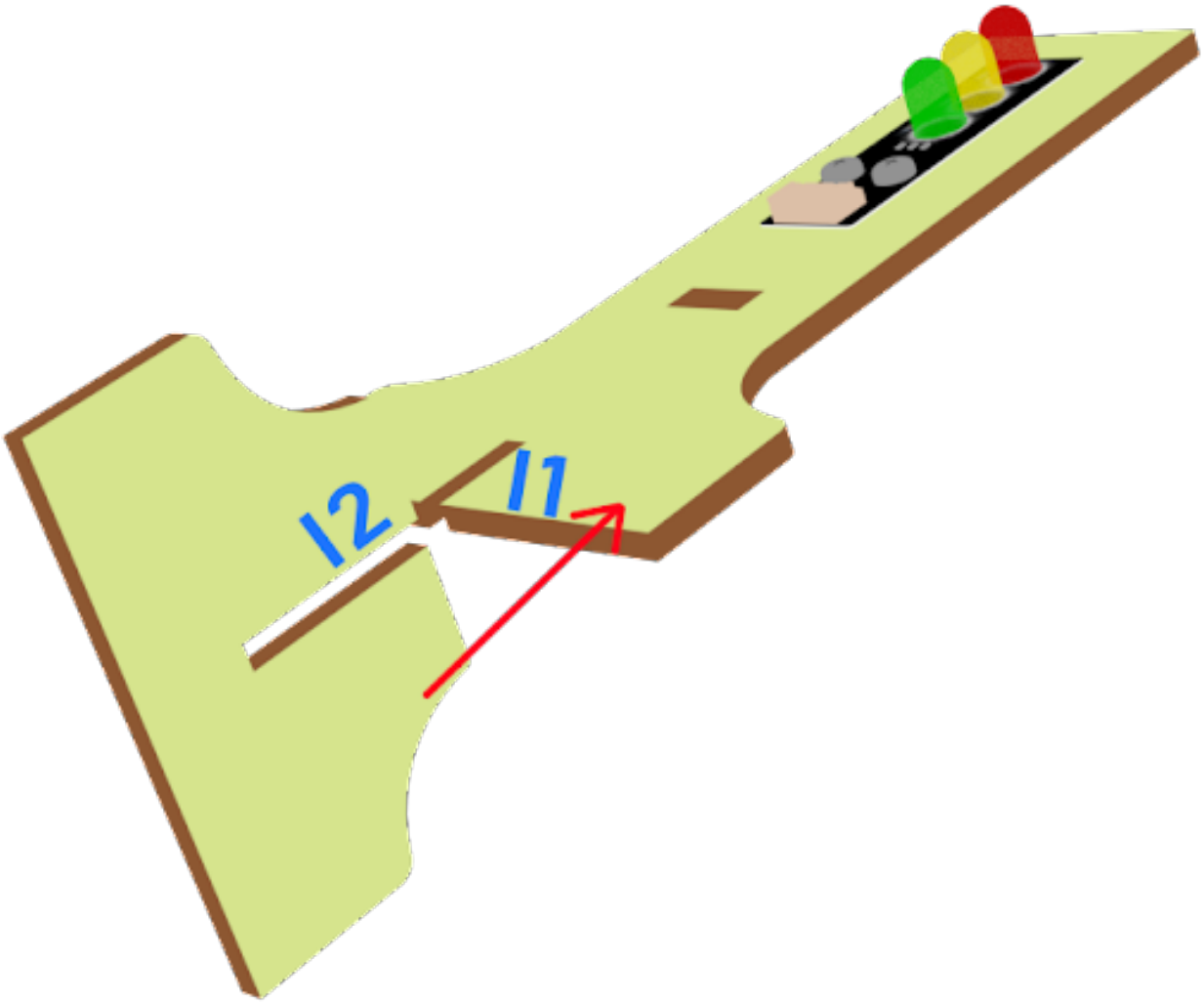
材料準備

組裝步驟

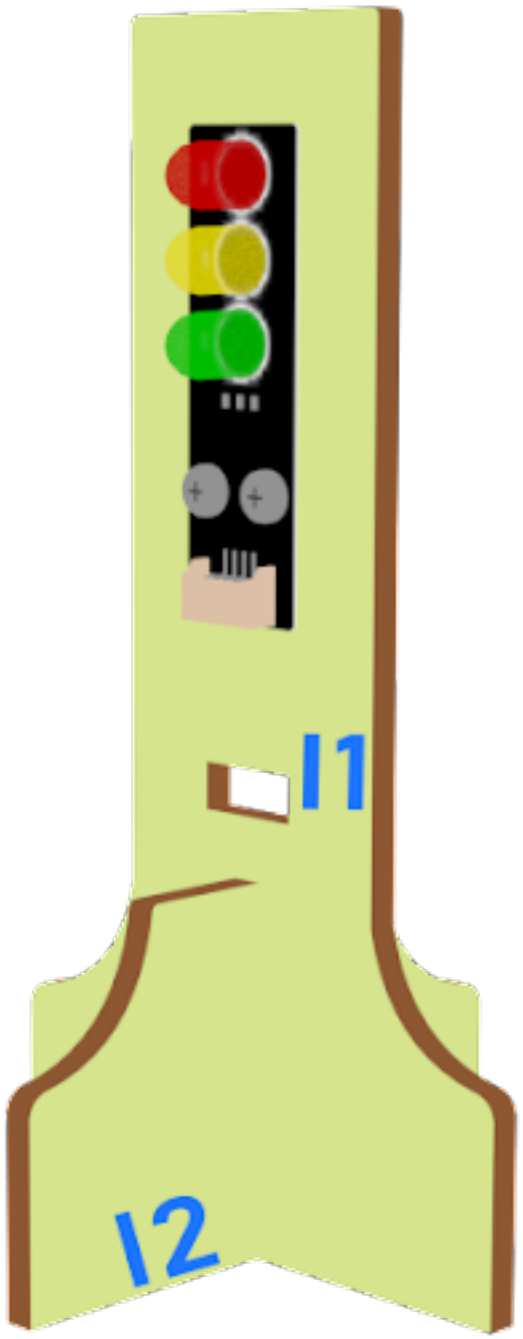
第一步



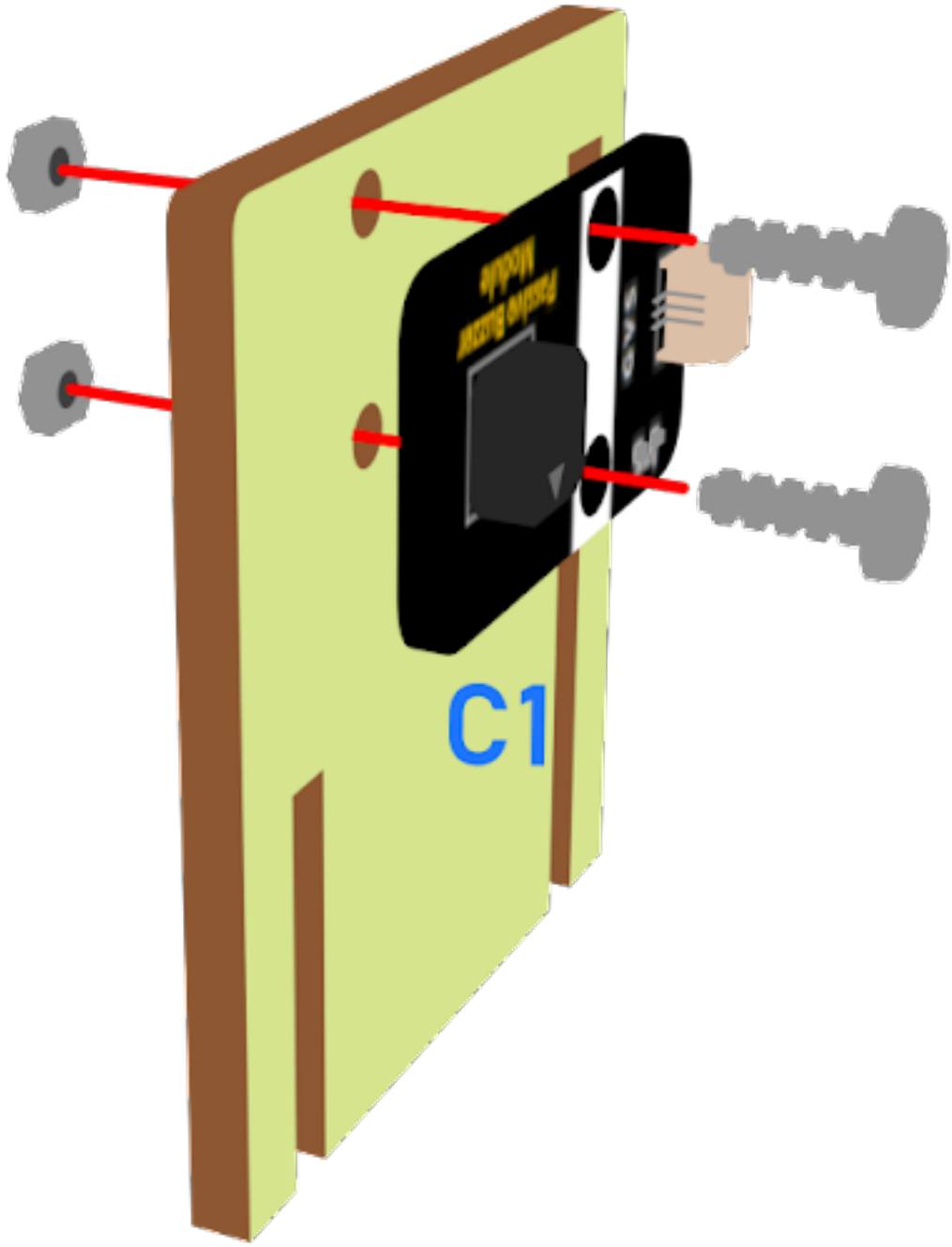
第二步



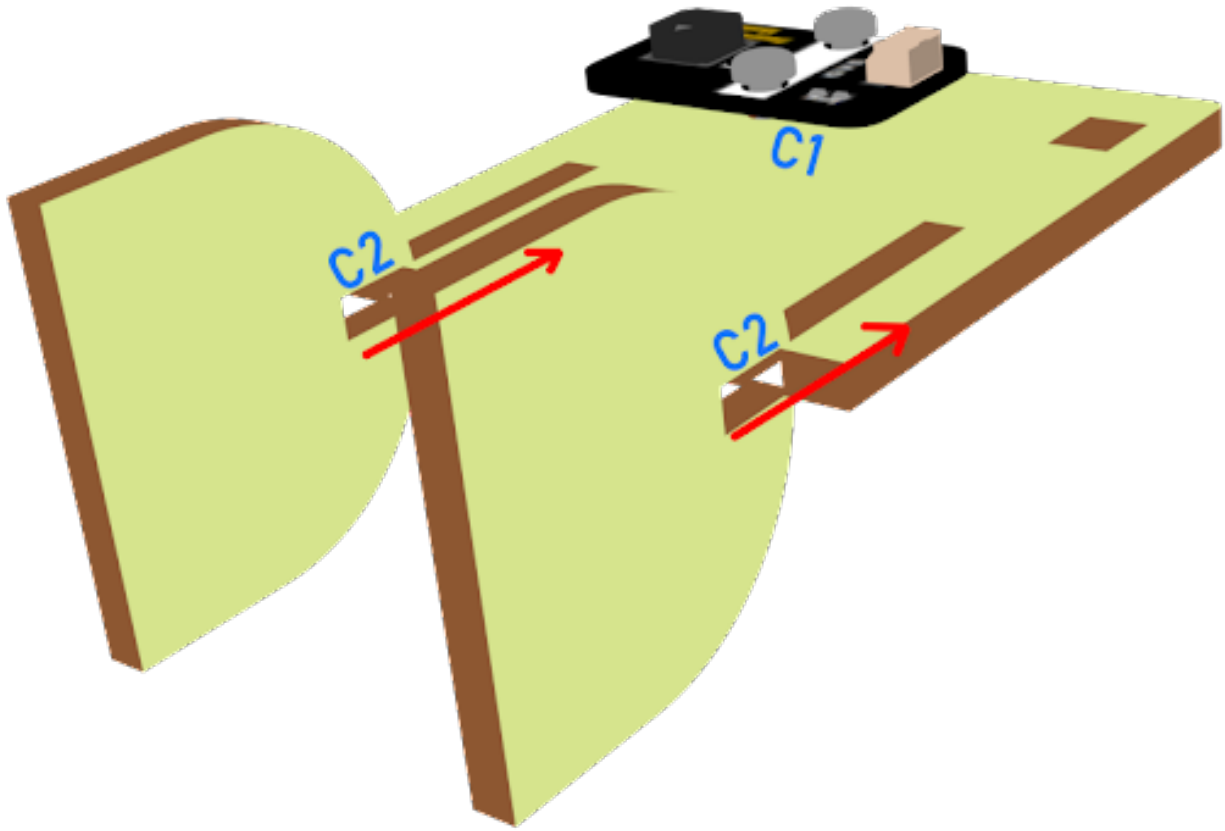
第三步



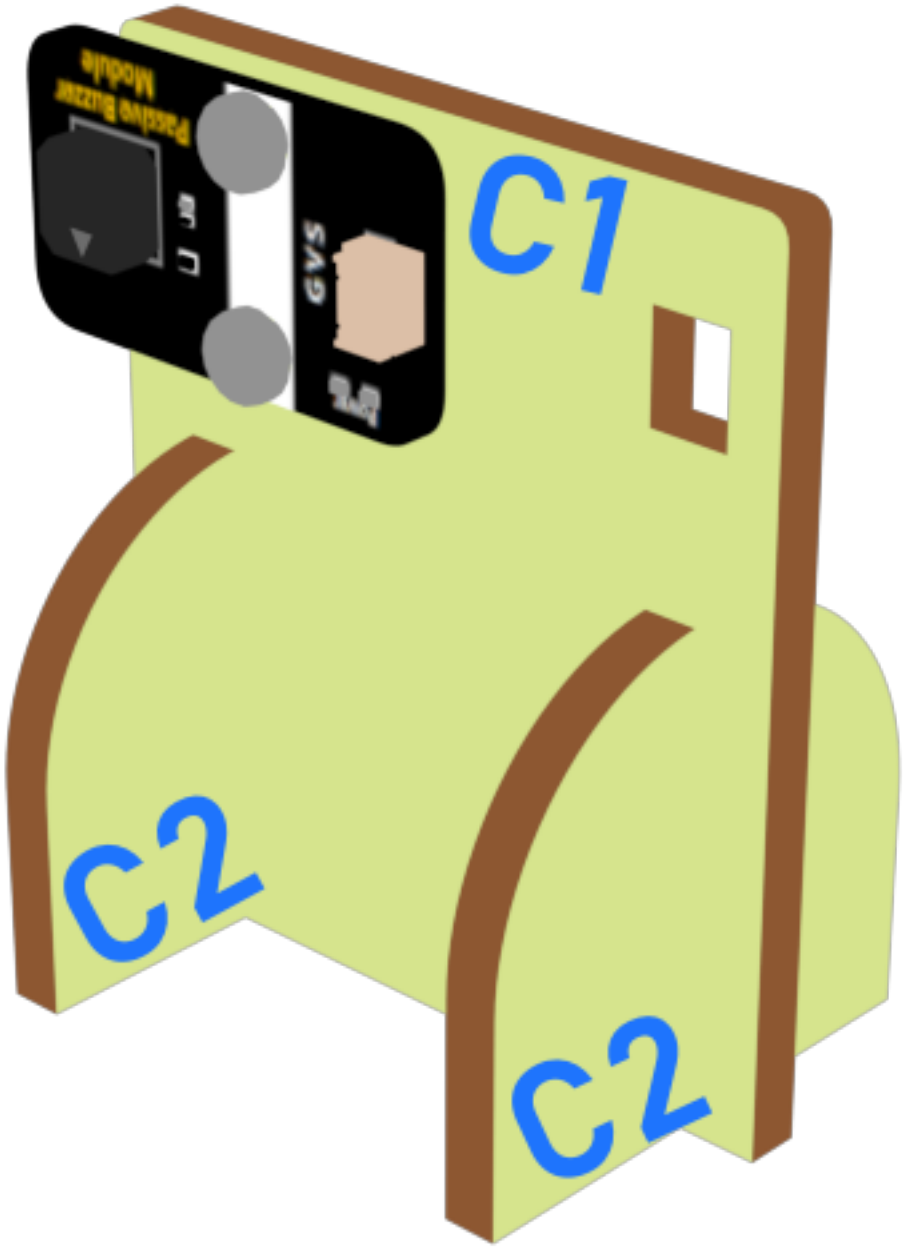
第四步



第五步



第六步



硬件連接

編程

Traffic light 1：

設置一個新的功能（TurnRed）。

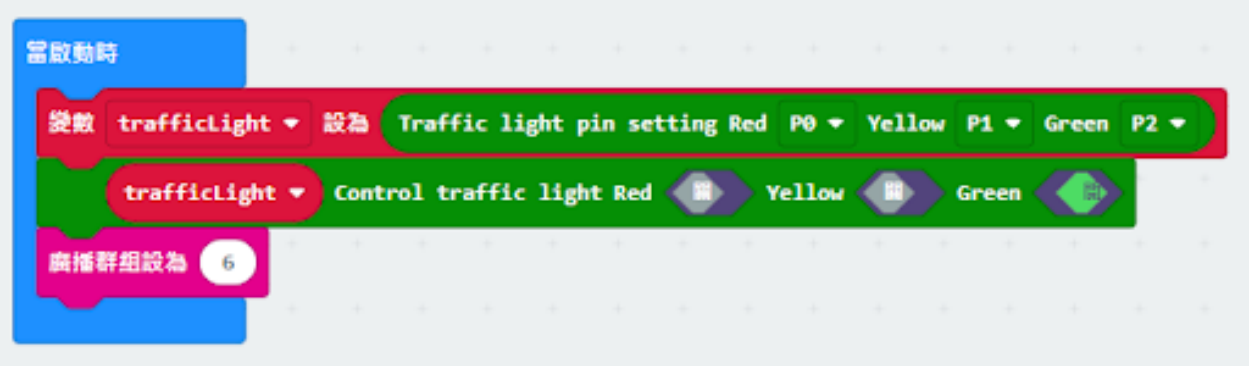
- 暫停一秒
- 控制交通燈亮F燈
- 暫停一秒
- 控制交通燈亮紅燈

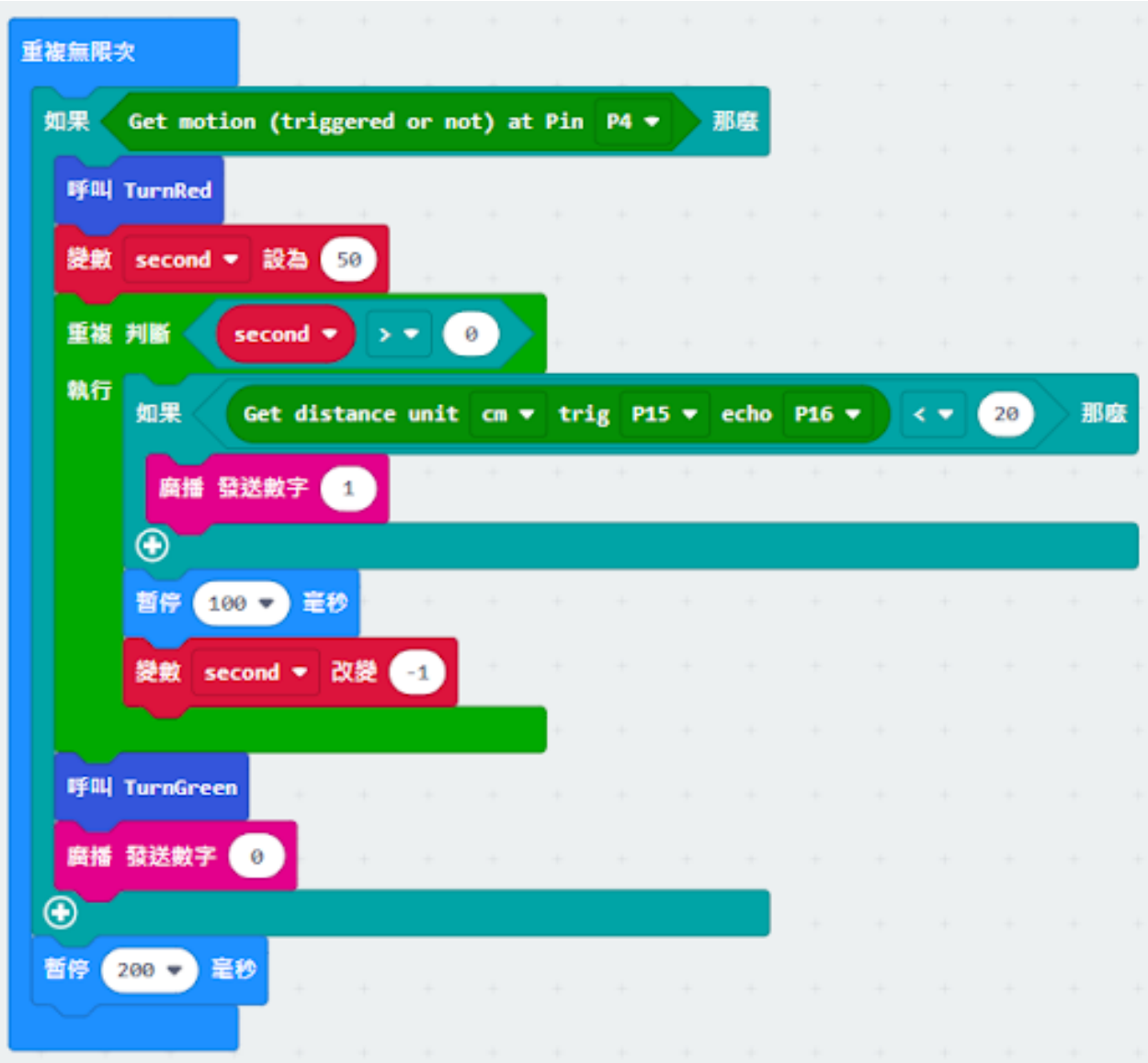
設置一个新的功能（TurnGreen）。

- 暫停一秒
- 控制交通燈亮F燈
- 暫停一秒
- 控制交通燈亮F燈

初始化程序

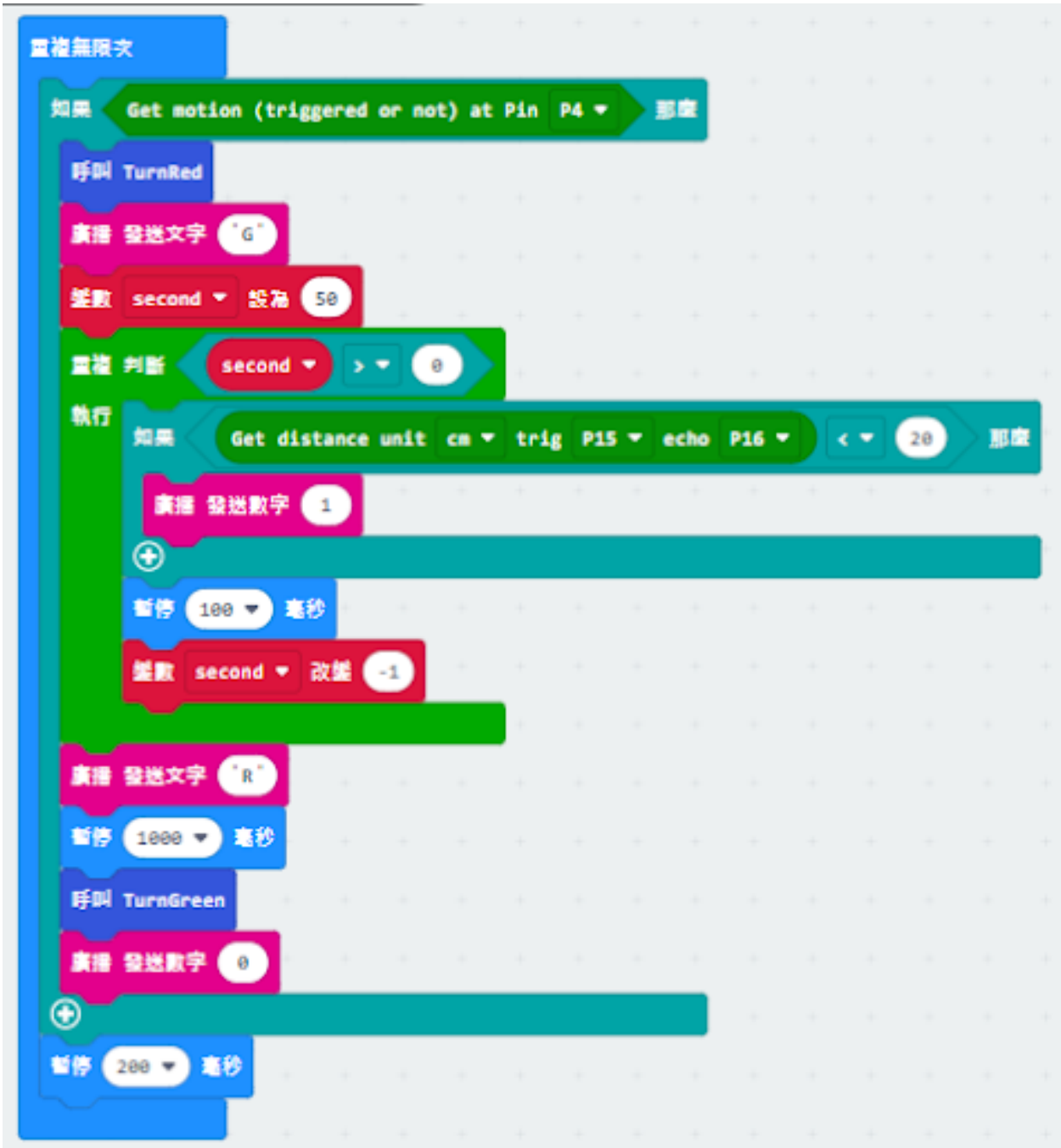
- 將 trafficLight to Traffic light pin setting Red P0 Yellow P1 Green P2 拖入當啓動時
- 控制交通燈亮F燈
- 將廣播群組設爲 6 拖入當啓動時
- 在循環無數次中，將呼叫 TurnRed 拖入 get motion (triggered or not) at pin P4 的如果句式中
- 將呼叫 TurnGreen 拖到循環后





發送廣播字串以控制交通燈

- 將廣播發送字串“R”拖到 TurnGreen 前
- 將廣播發送字串“G”拖到 TurnRed 前

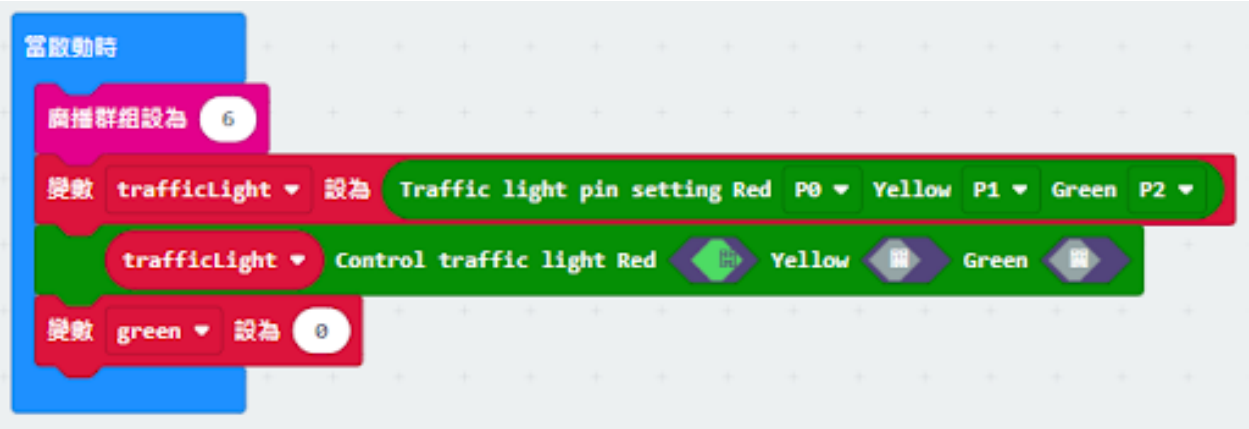

重複無限次
如果 Get motion (triggered or not) at Pin P4 那麼
呼叫 TurnRed
廣播 發送文字 "G"
變數 second 設為 50
重複 判斷 second > 0
執行
如果 Get distance unit cm trig P15 echo P16 < 20 那麼
廣播 發送數字 1
暫停 100 毫秒
變數 second 改變 -1
廣播 發送文字 "R"
暫停 1000 毫秒
呼叫 TurnGreen
廣播 發送數字 0
暫停 200 毫秒

Traffic light 2：

設置新的功能。

初始化程序

- 將 trafficLight to Traffic light pin setting Red P0 Yellow P1 Green P2 拖入當啓動時
- 將廣播群組設爲 6 拖入當啓動時
- 控制交通燈亮F燈
- 設變數 green=0



通過接收不同數字來控制交通燈及車輛

- ng 將如果語句放入當收到廣播數字 receivedString
- 設 receivedString=” R”，Ⓕ呼叫 TurnRed
- 設 receivedString=” G”，Ⓕ呼叫 TurnGreen
- 根據燈光改變變數 green 的數值

根據燈光播放聲效

- 將如果語句放入循環無數次
- 彈奏不同節奏的旋律

Receiver:

在起始位置設置廣播群組

- 將廣播群組設爲 6 拖入當啓動時
- 最初，汽車默認向前移動

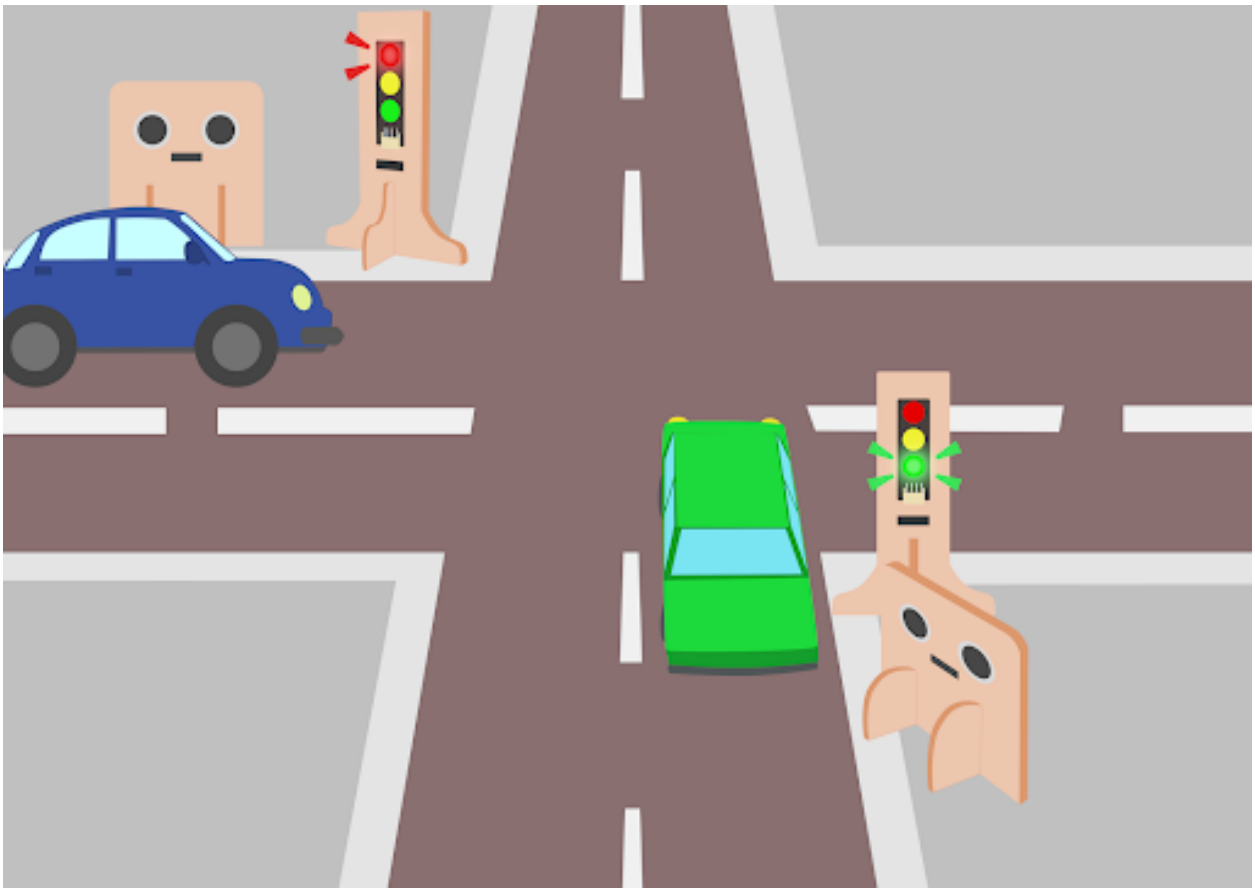
通過接收不同數字來控制車輛

- 將如果語句放入當收到廣播數字 receivedNumber
- 設 receivedNumber =1，Ⓕ控制剎車。
- 設 receivedNumber=0，Ⓕ讓車向前移動。

總結

思考

1.2.11 十字路口

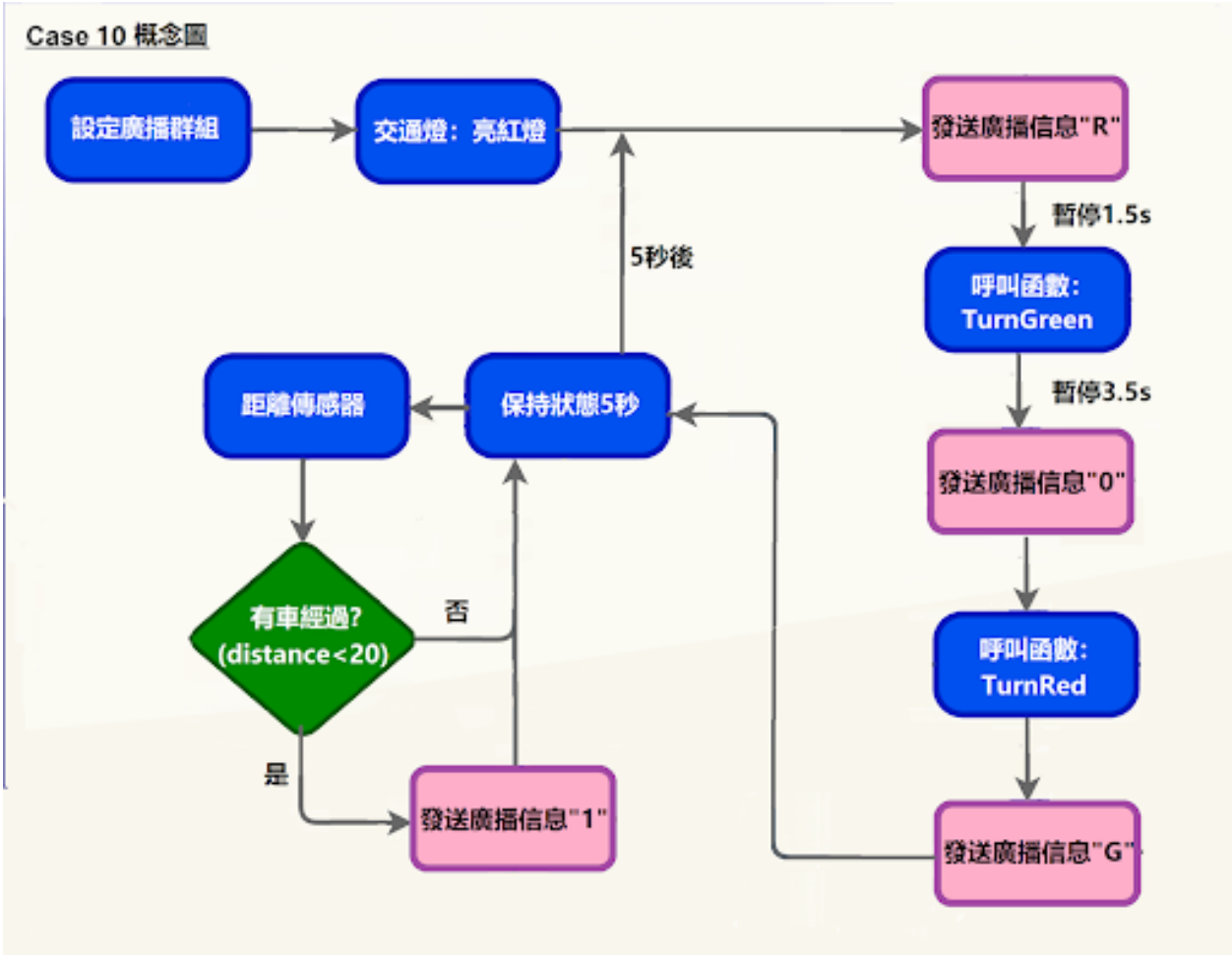


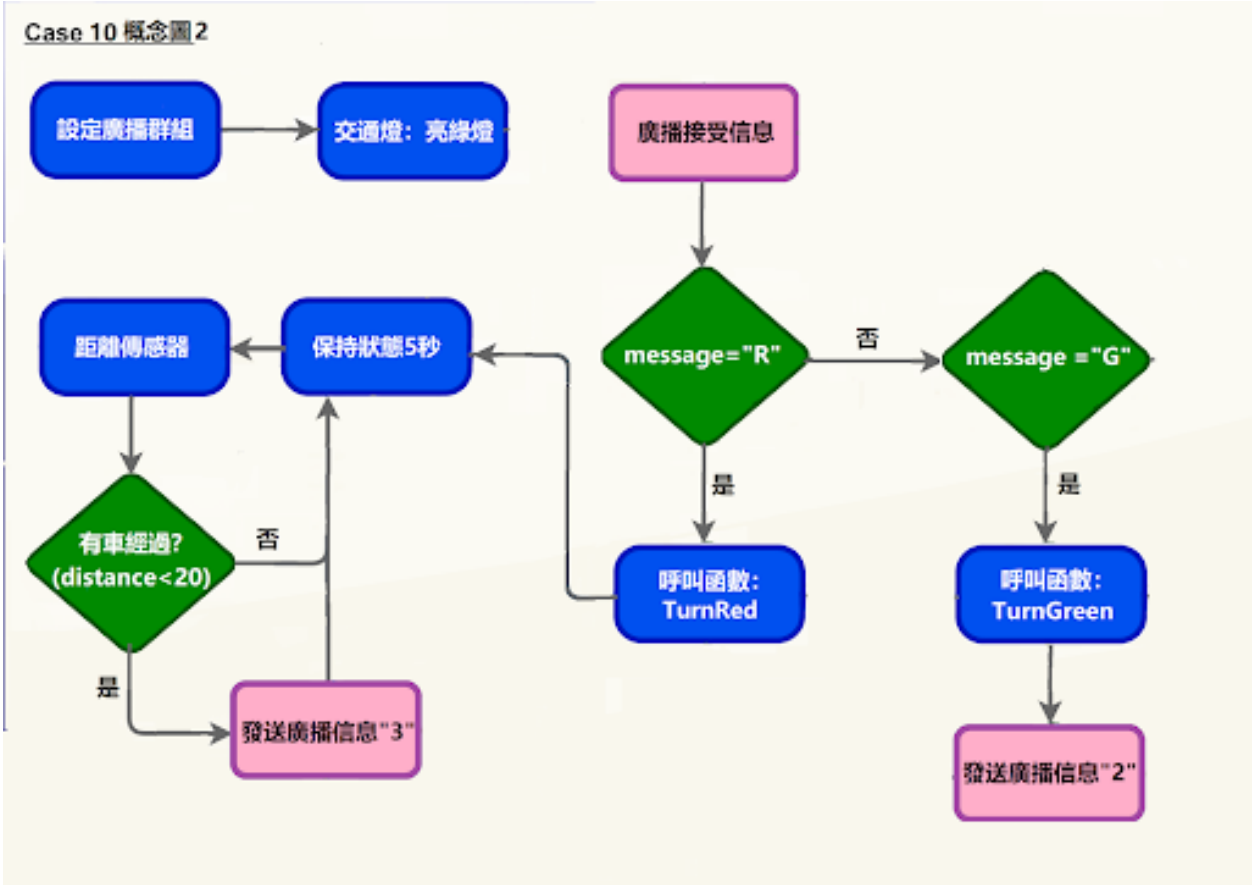
目標

背景

什麼是智能交通燈？

智能交通燈運作

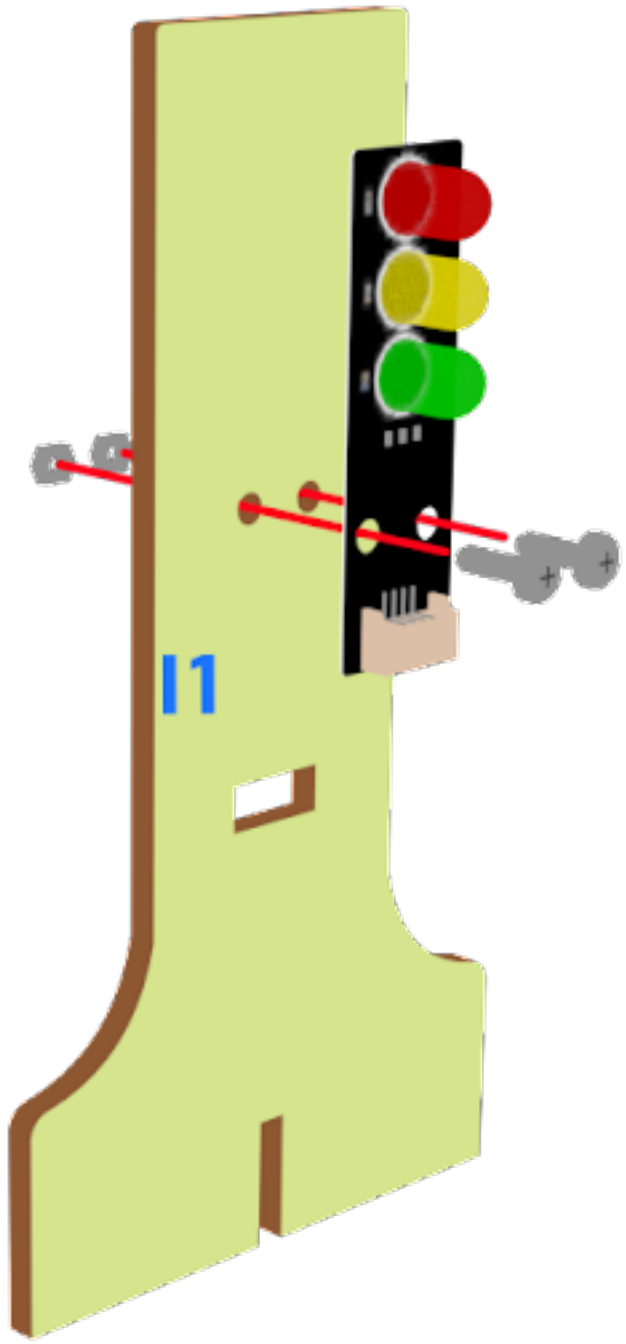

Case 10 概念圖
設定廣播群組
交通燈：亮紅燈
發送廣播信息"R"
暫停1.5s
呼叫函數:
TurnGreen
暫停3.5s
發送廣播信息"0"
呼叫函數:
TurnRed
發送廣播信息"G"
5秒後
保持狀態5秒
距離傳感器
有車經過?
(distance<20)
否
是
發送廣播信息"1"


Case 10 概念圖2
設定廣播群組
交通燈：亮綠燈
廣播接受信息
距離傳感器
保持狀態5秒
message="R"
否
message ="G"
是
是
有車經過?
(distance<20)
否
呼叫函數:
TurnRed
呼叫函數:
TurnGreen
是
發送廣播信息"3"
發送廣播信息"2"

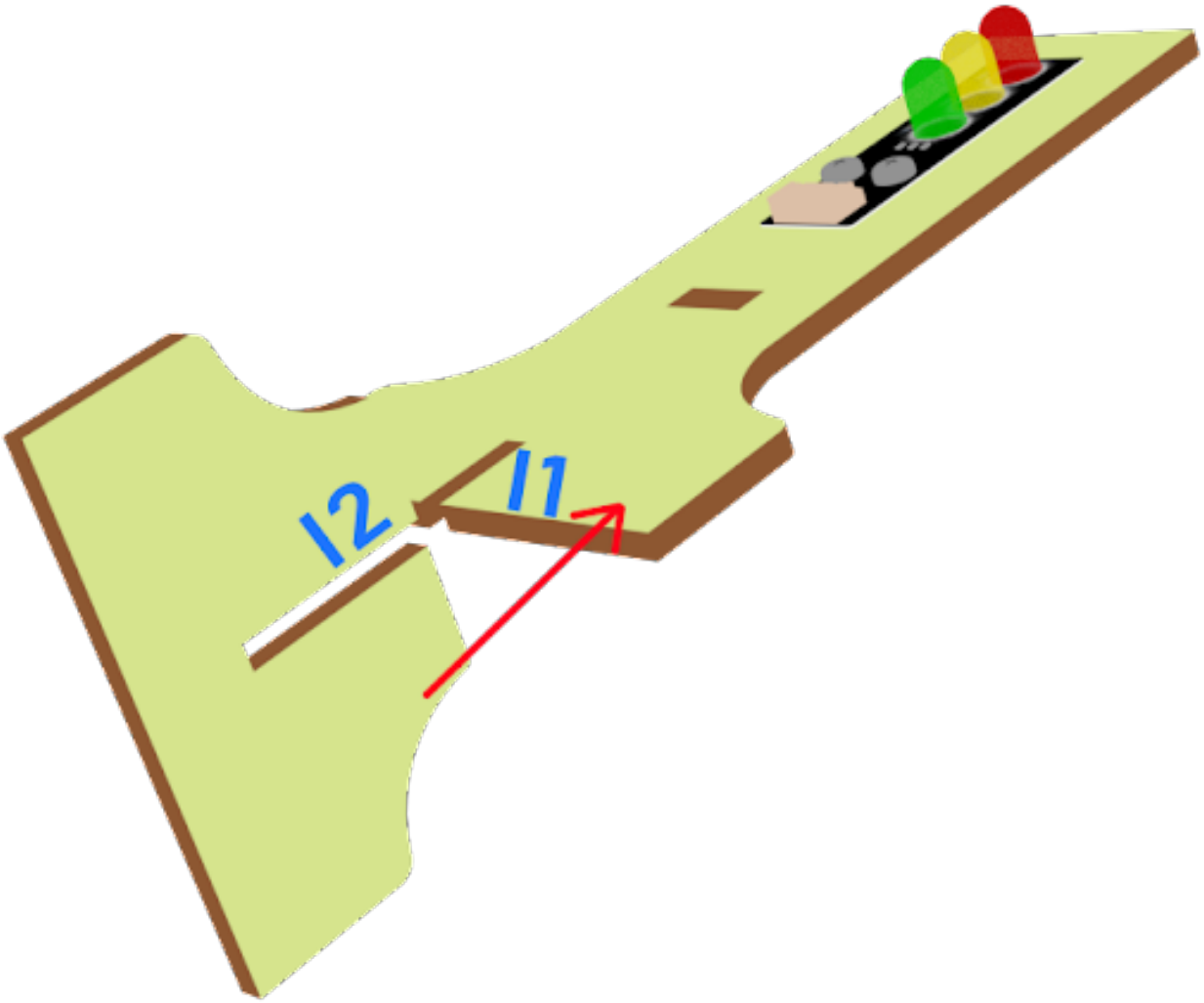
材料準備

組裝步驟

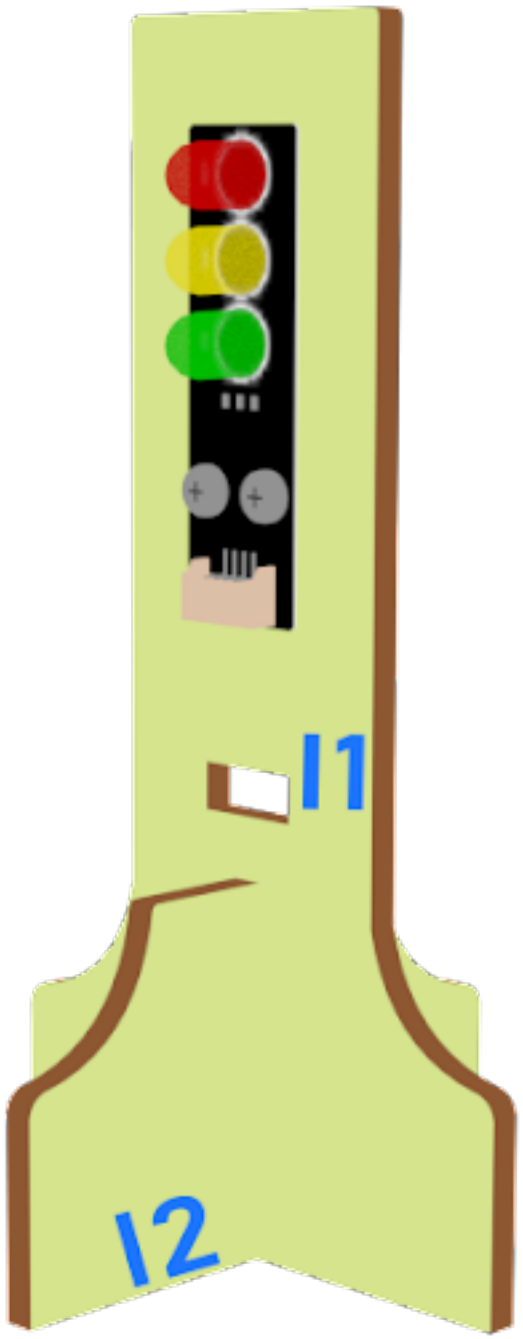
第一步



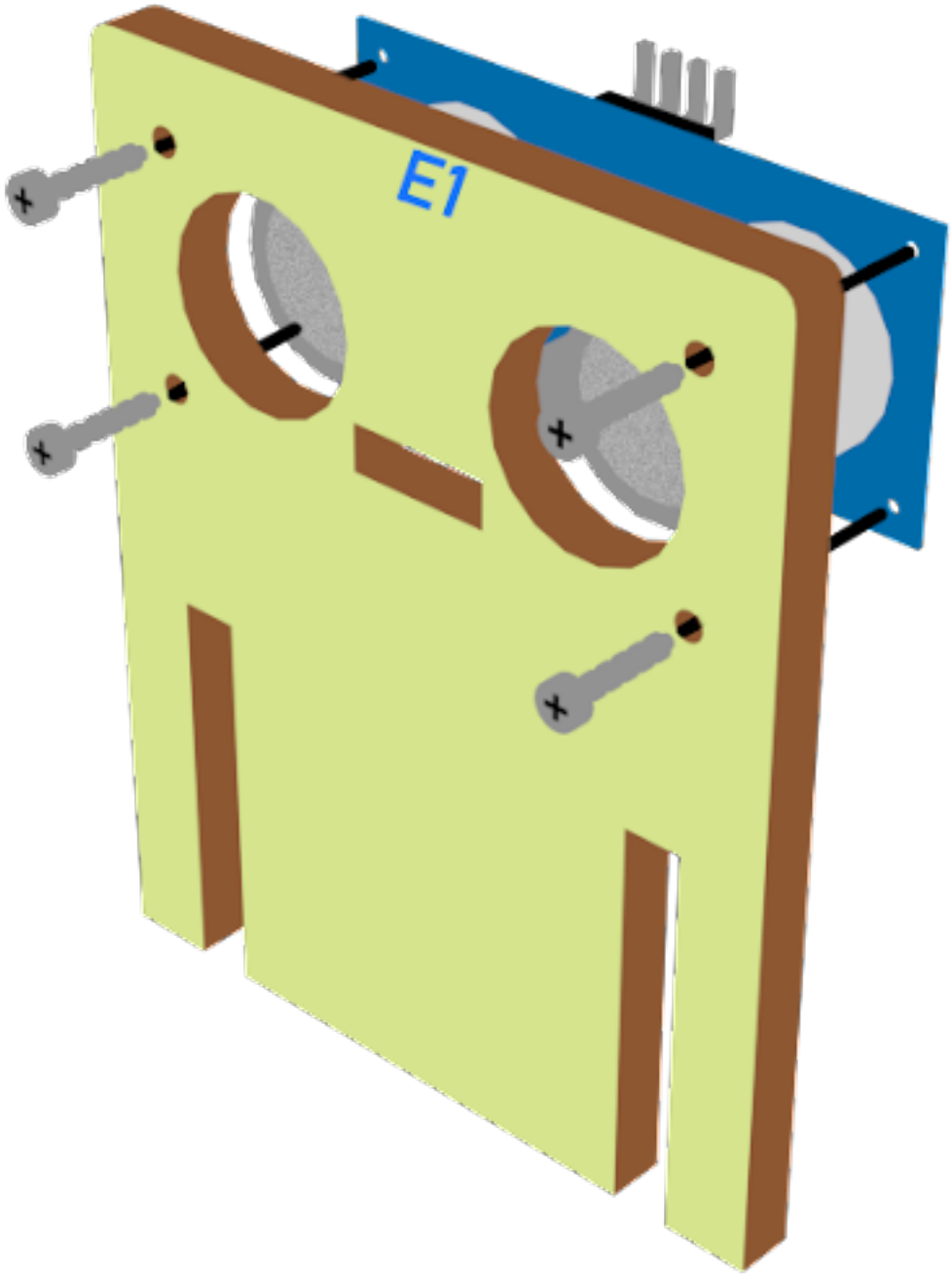
第二步



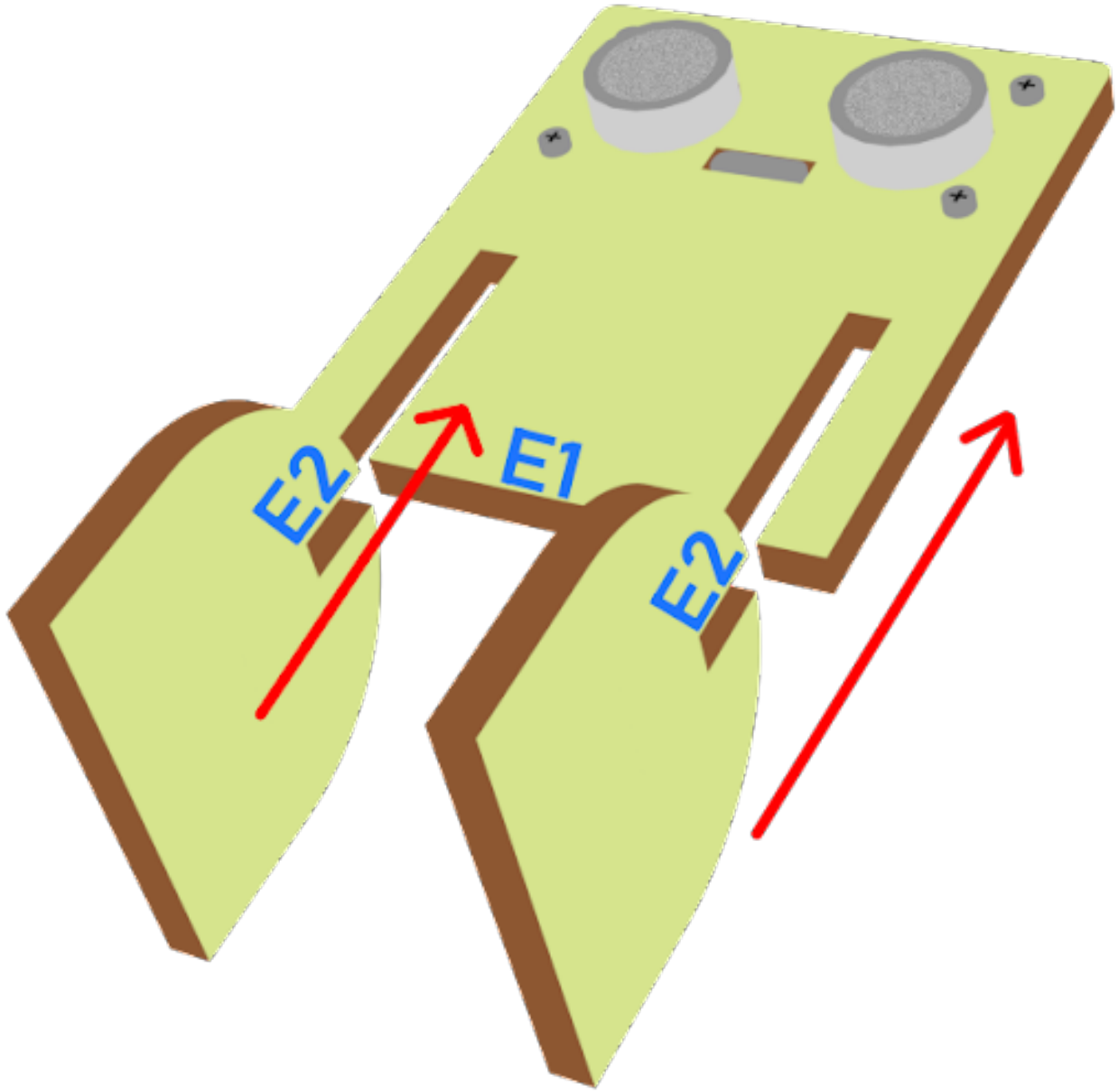
第三步



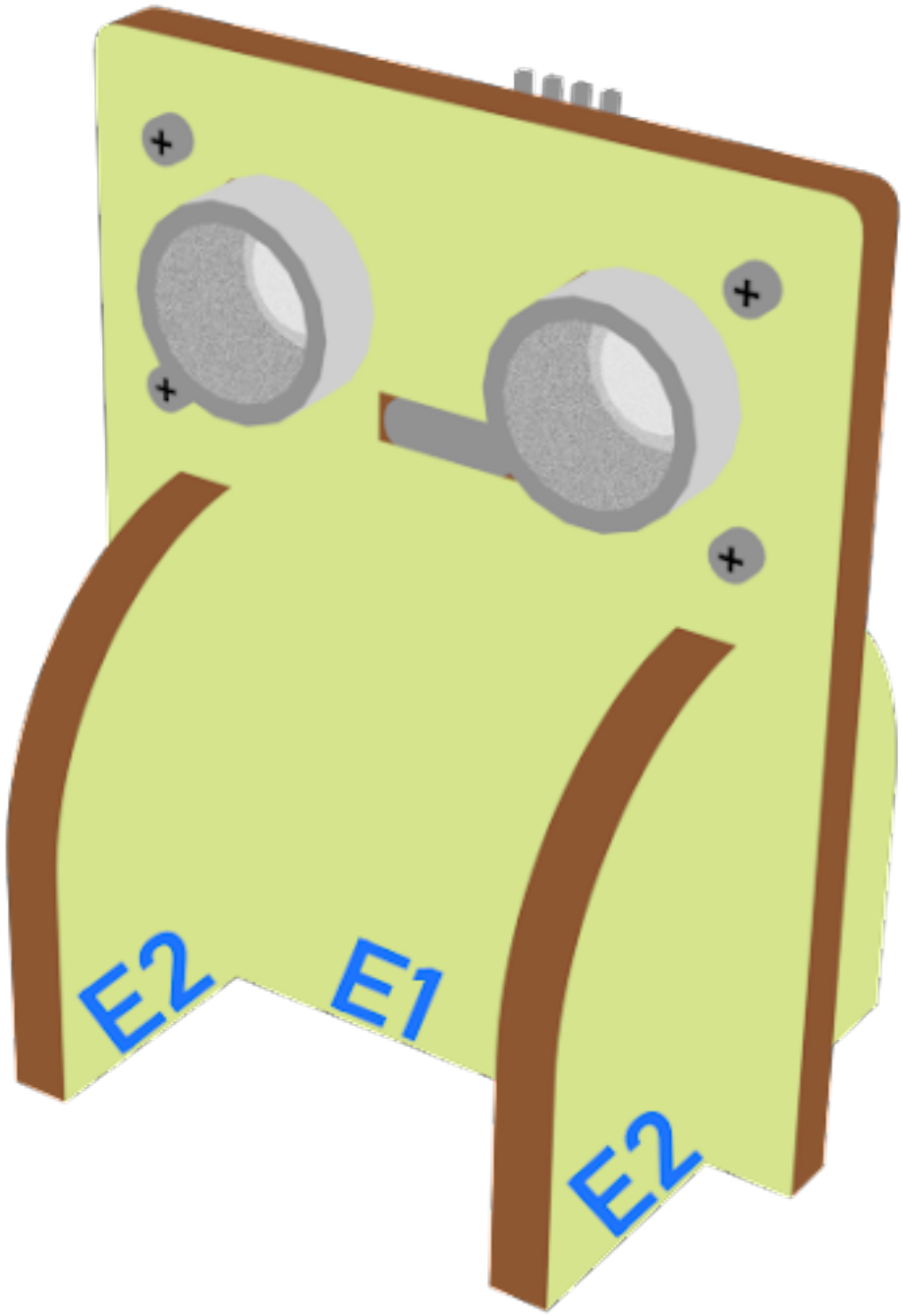
第四步



第五步



第六步



硬件連接

編程

Traffic light 1：

設置一個新的功能（TurnRed）。

- 暫停一秒
- n 控制交通燈亮Ⓕ燈
- 暫停一秒
- 控制交通燈亮紅燈

設置一个新的功能（TurnGreen）。

- 暫停一秒
- 控制交通燈亮Ⓕ燈
- 暫停一秒
- 控制交通燈亮Ⓕ燈

初始化程式

- 將 trafficLight to Traffic light pin setting Red P0 Yellow P1 Green P2 拖入當啓動時
- 將廣播群組設爲 10 拖入當啓動時
- 控制交通燈亮紅燈
- 暫停 5 秒

保留F燈狀態五秒

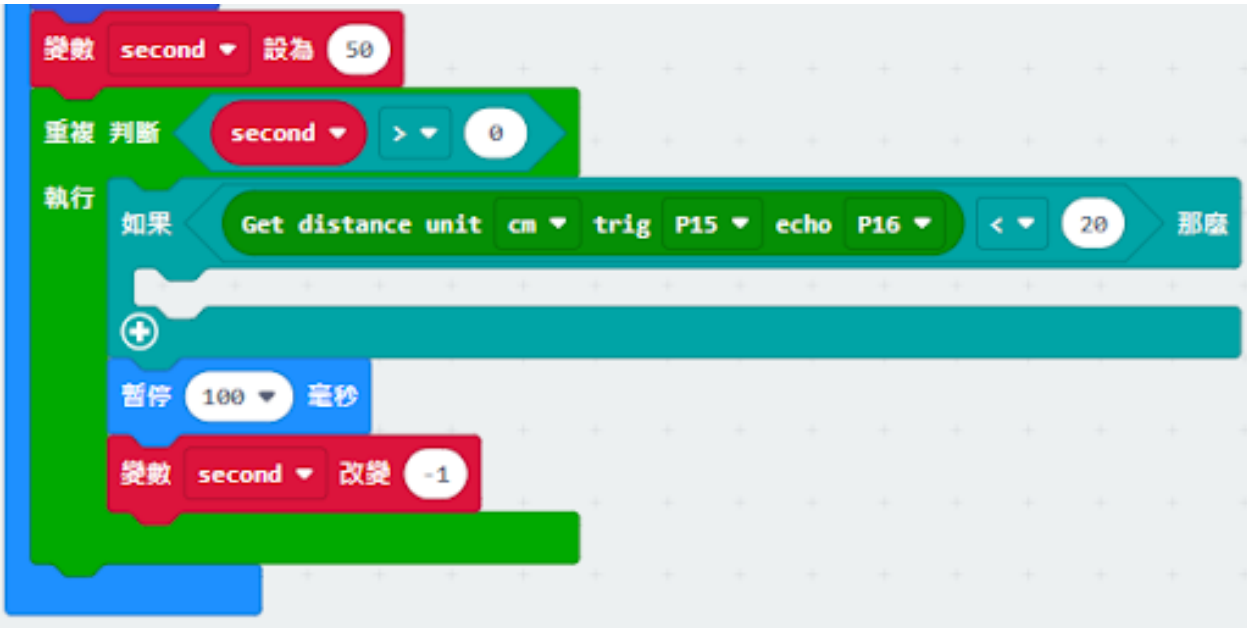
- 呼叫函數 TurnGreen
- 在此前暫停 1.5s
- 在此後暫停 3.5s

保留紅燈狀態五秒

- 呼叫函數 TurnRed
- 設定變數 second Ⓕ 50
- 當 second 大於 0，停止 0.1 秒，Ⓕ改變 second Ⓕ少 1

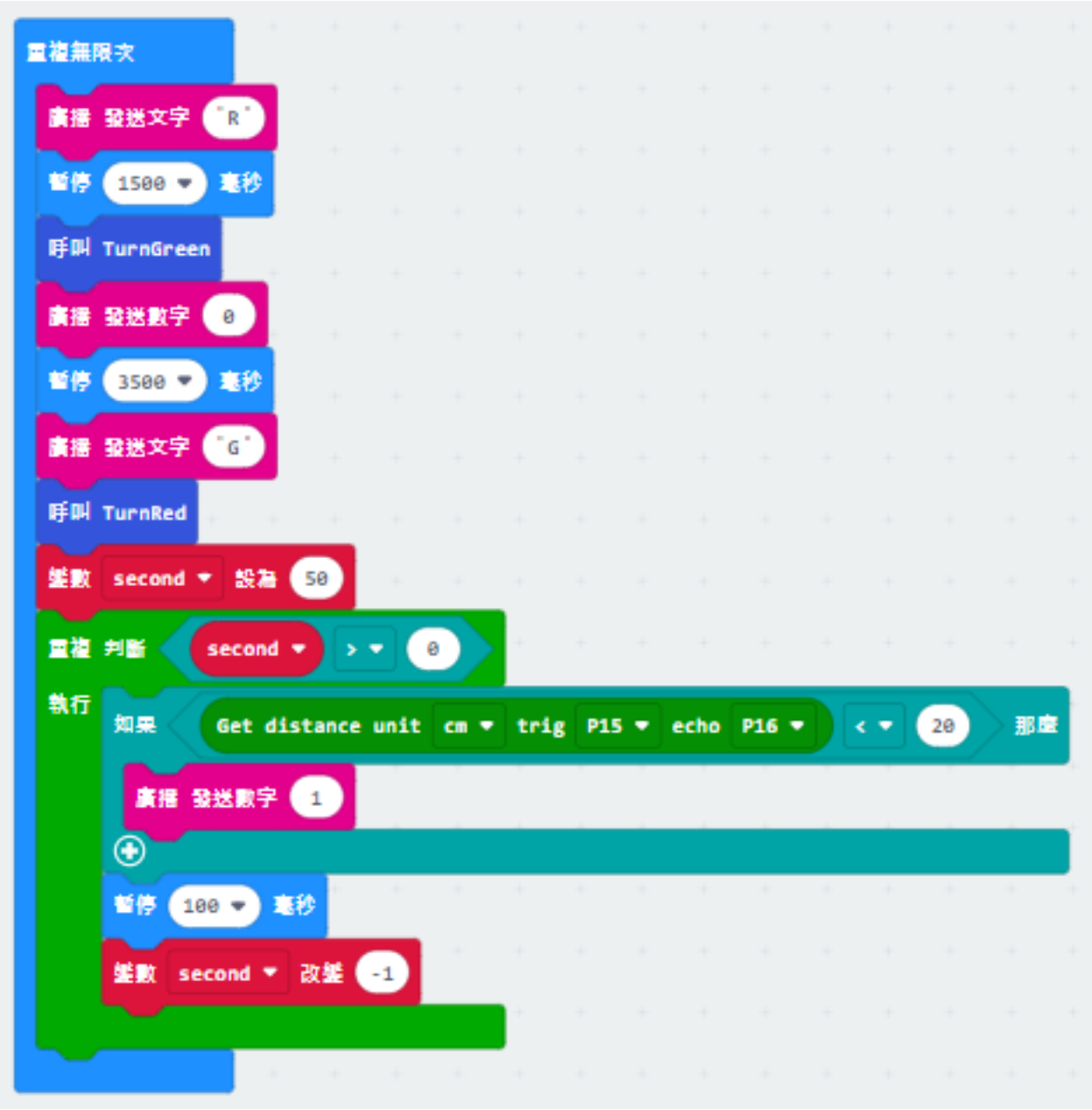
獲得距離數值

- 將 get distance unit cm trig P15 echo P16 < 20 的如果句式拖入循環中



發送廣播數字以控制交通燈及車輛

- 將廣播發送數字 1 拖入 if 句式
- 將廣播發送數字 0 拖到 TurnGreen 後
- 將廣播發送字串“R”拖到 TurnGreen 前
- 將廣播發送字串“G”拖到 TurnRed 前



Traffic light 2:

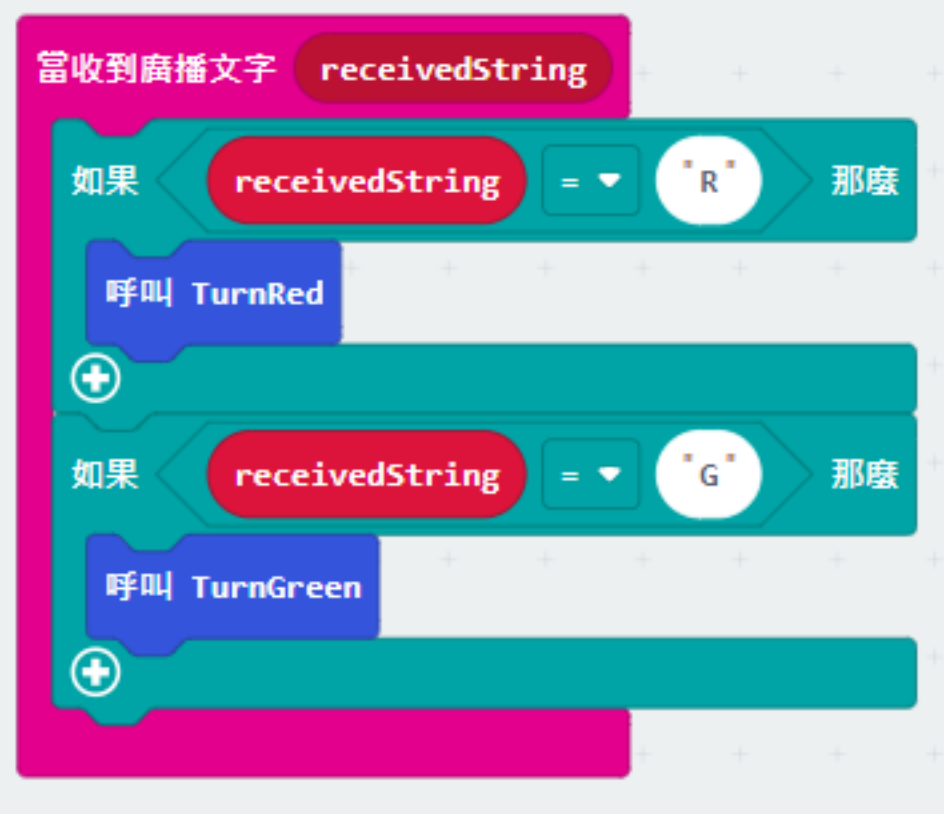
設置新的功能。

初始化程序

- 將 trafficLight to Traffic light pin setting Red P0 Yellow P1 Green P2 拖入當啓動時
- 將廣播群組設爲 10 拖入當啓動時
- 控制交通燈亮Ⓕ燈
- 暫停 5 秒

通過接收不同數字來控制交通燈及車輛

- 將如果語句放入當收到廣播數字 receivedString
- 設 receivedString=” R”，Ⓕ呼叫 TurnRed
- S 設 receivedString=” G”，Ⓕ呼叫 TurnGreen

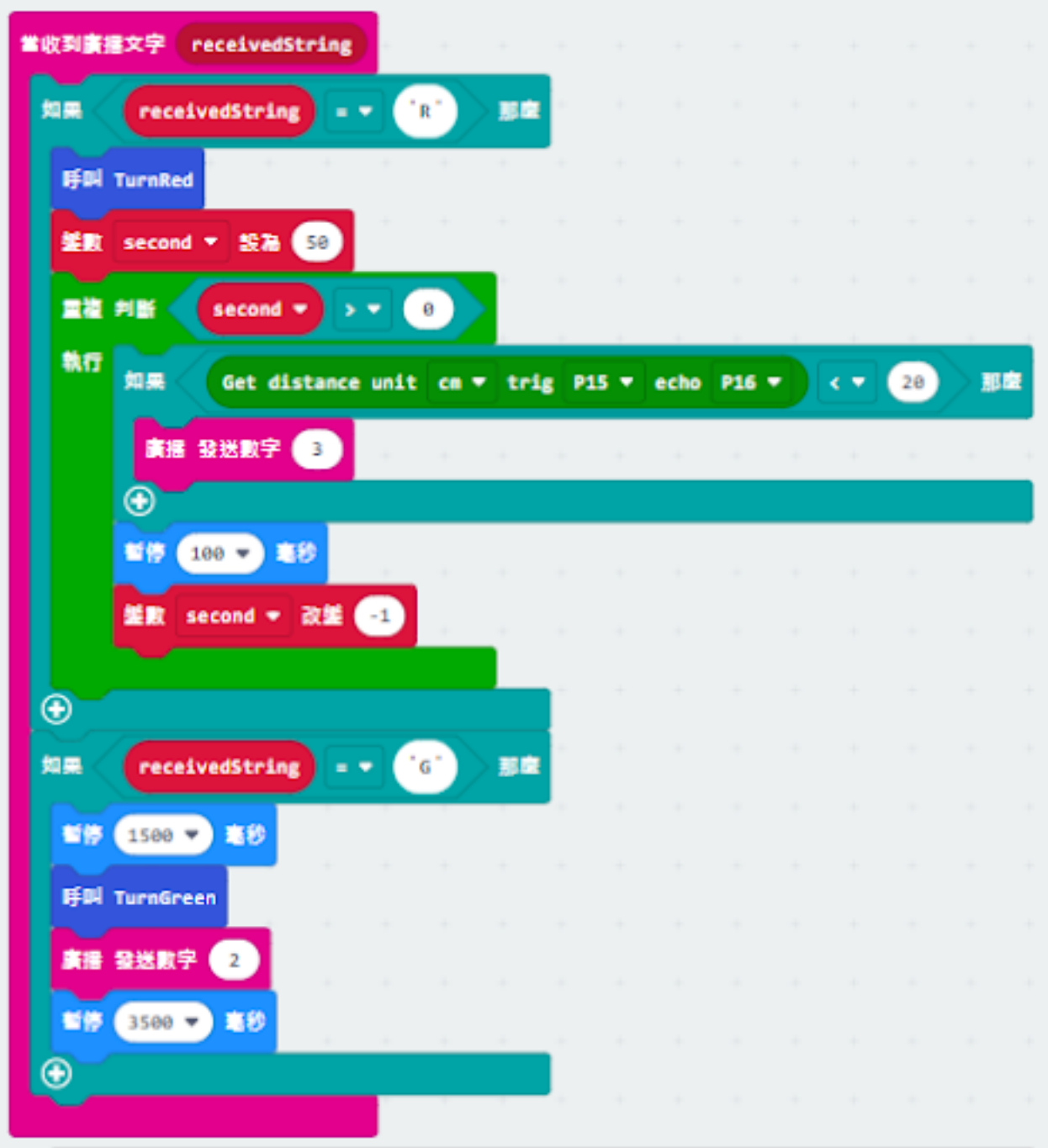


保留紅燈狀態五秒

- 設定變數 second Ⓕ 50
- 當 second 大於 0，停止 0.1 秒，Ⓕ改變 second Ⓕ少 1
- 在此前暫停 1.5s
- 在此後暫停 3.5s

獲得距離數值Ⓕ控制車輛

- 將 get distance unit cm trig P15 echo P16 < 20 的如果句式拖入循環中
- 將廣播發送數字 1 拖入 if 句式
- 將廣播發送數字 0 拖到 TurnGreen 後



Car 1:

在起始位置設置廣播群組

- 將廣播群組設爲 10 拖入當啓動時

通過接收不同數字來控制車輛

- 將如果語句放入當收到廣播數字 receivedNumber
- 設 receivedNumber =1，Ⓕ控制剎車。
- 設 receivedNumber=0，Ⓕ讓車向前移動。

Car 2:

在起始位置設置廣播群組

- 將廣播群組設爲 10 拖入當啓動時

通過接收不同數字來控制車輛

- 將如果語句放入當收到廣播數字 receivedNumber
- 設 receivedNumber =3，Ⓕ控制剎車。
- 設 receivedNumber=2，Ⓕ讓車向前移動。

總結

思考

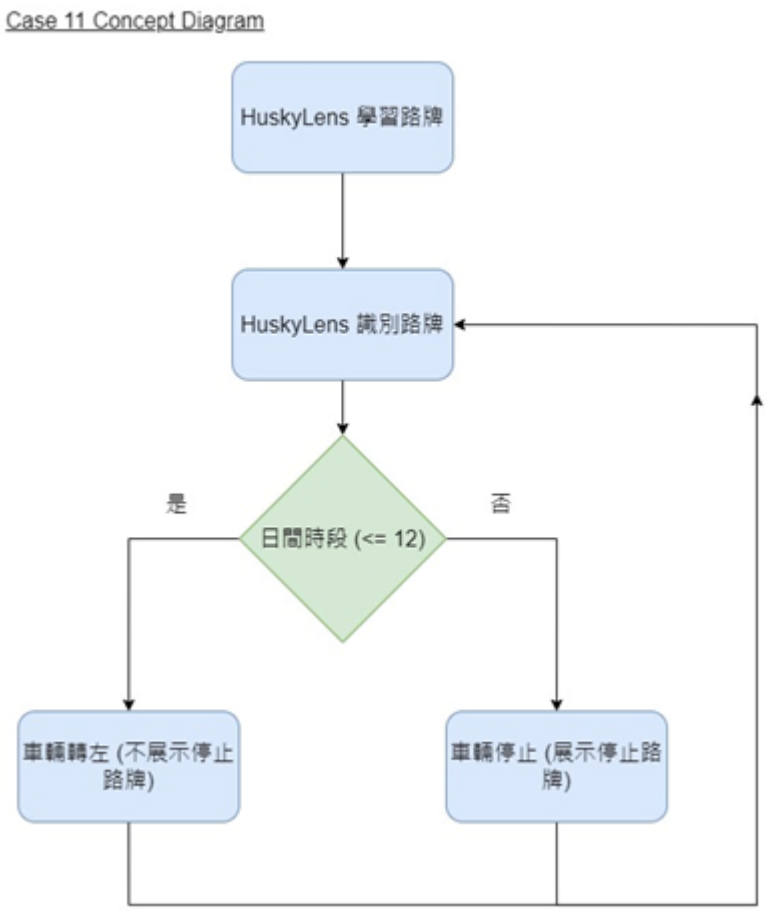
1.2.12 智能路牌

目標

背景

什F是智能路牌?

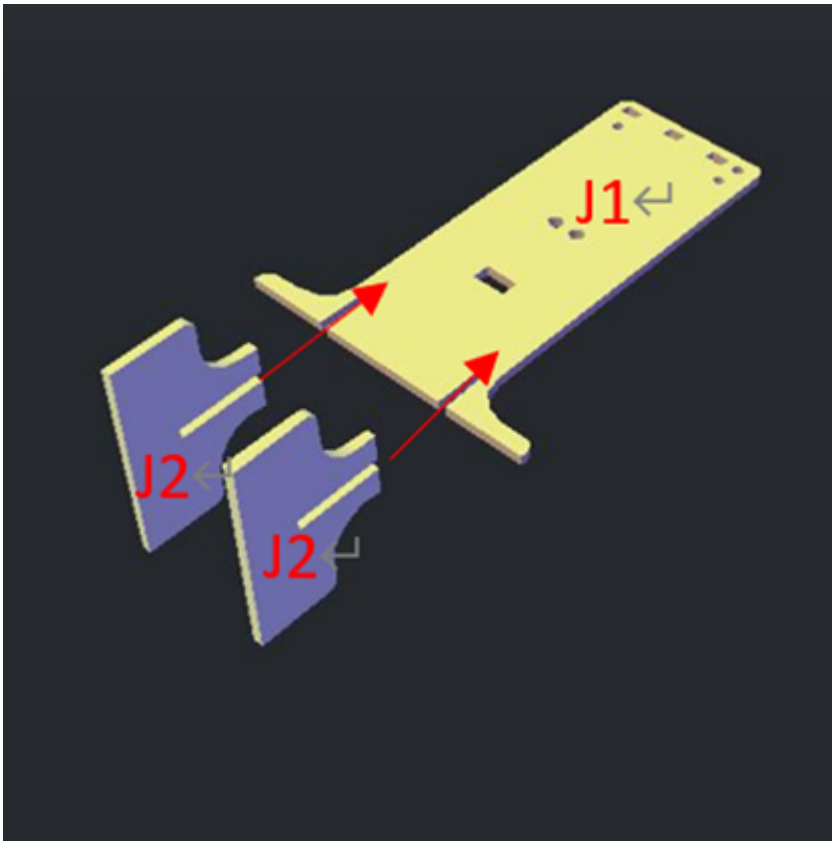
智能路牌運作



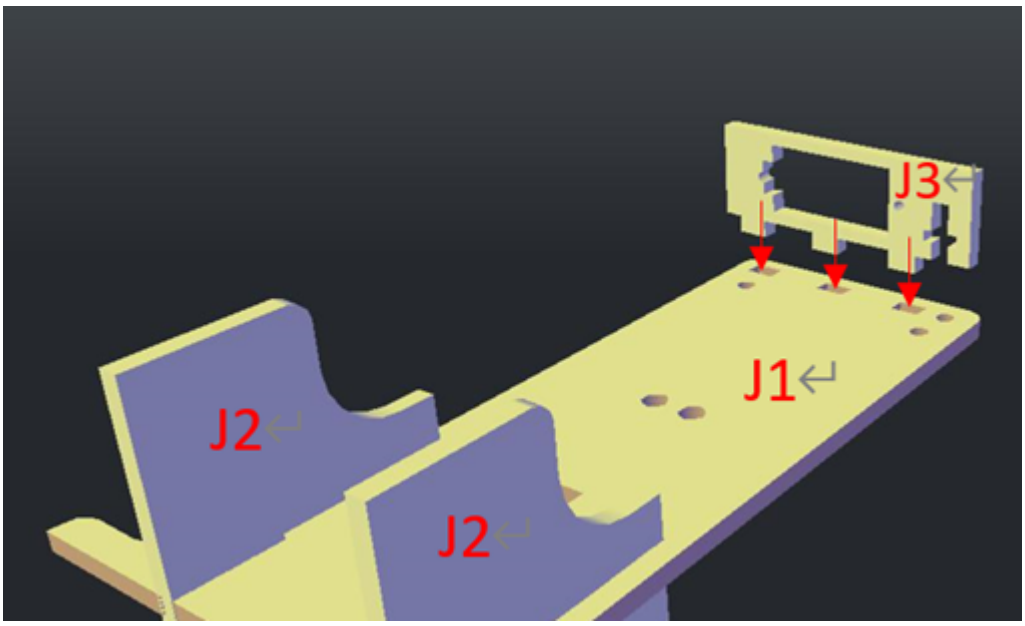
材料準備

組裝步驟

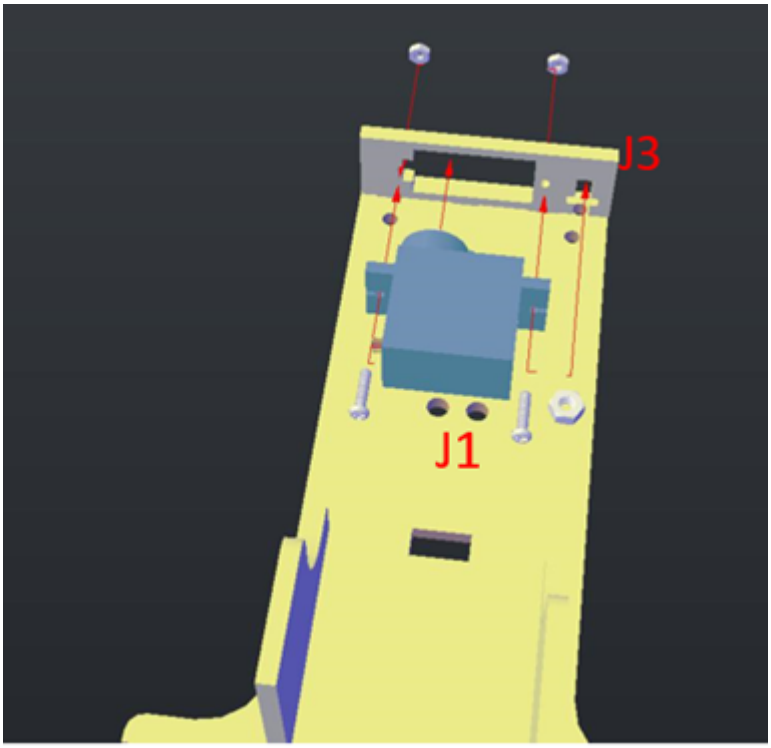
第一步



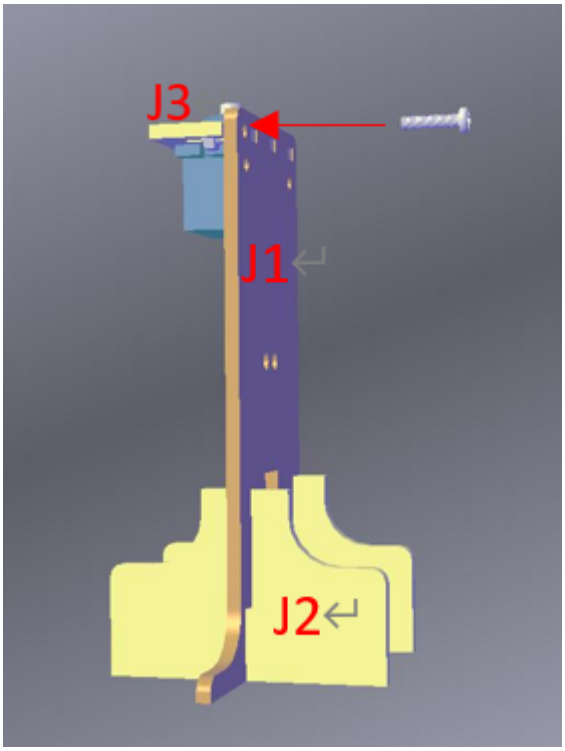
第二步



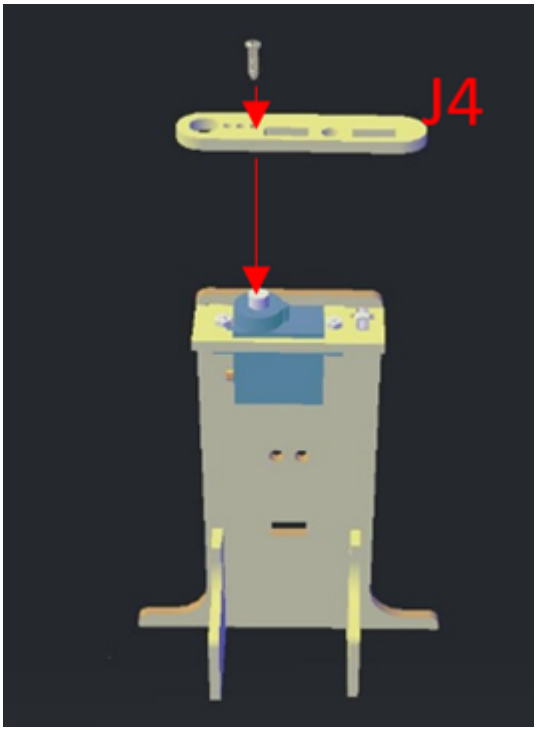
第三步



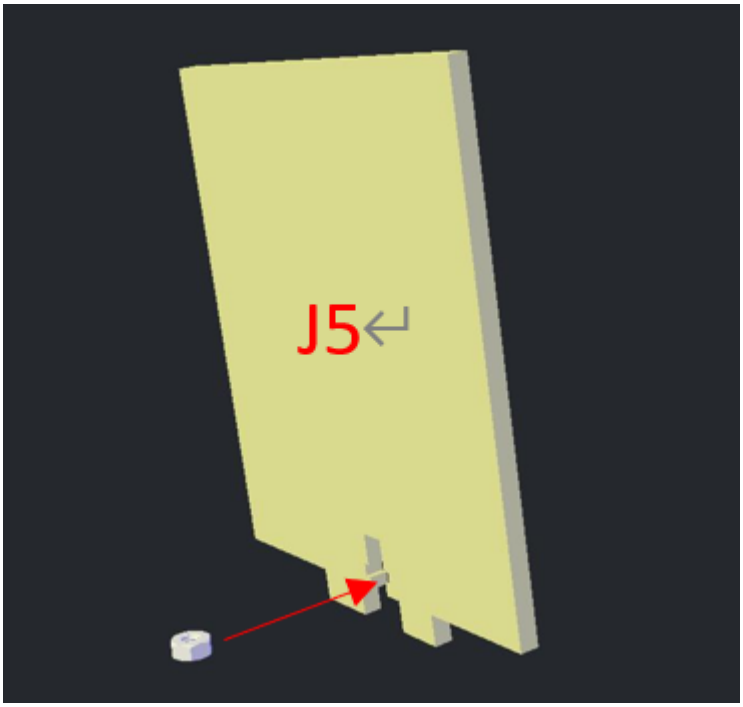
第四步



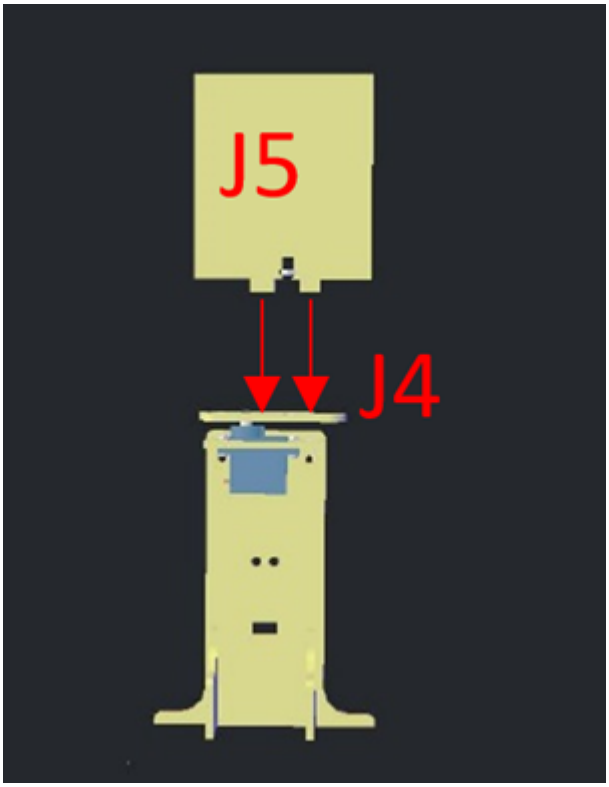
第五步



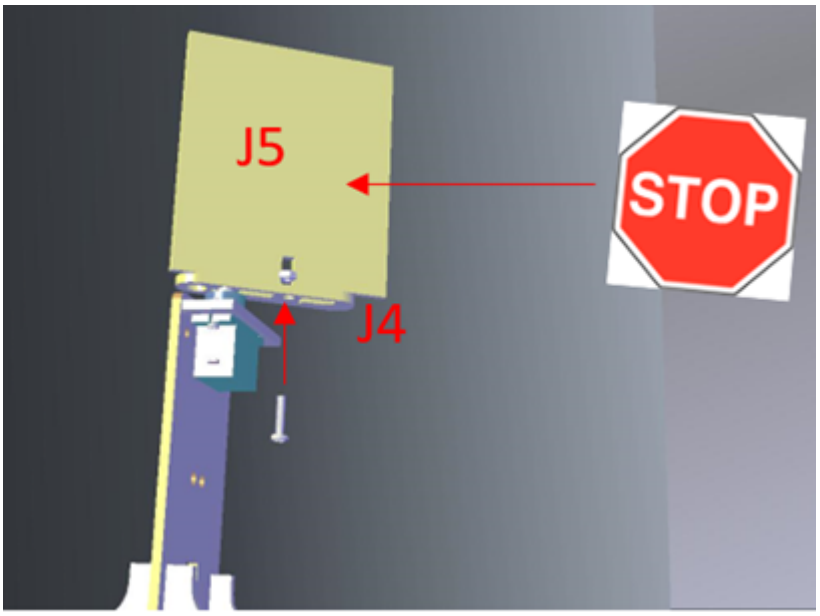
第六步



第七步



第八步



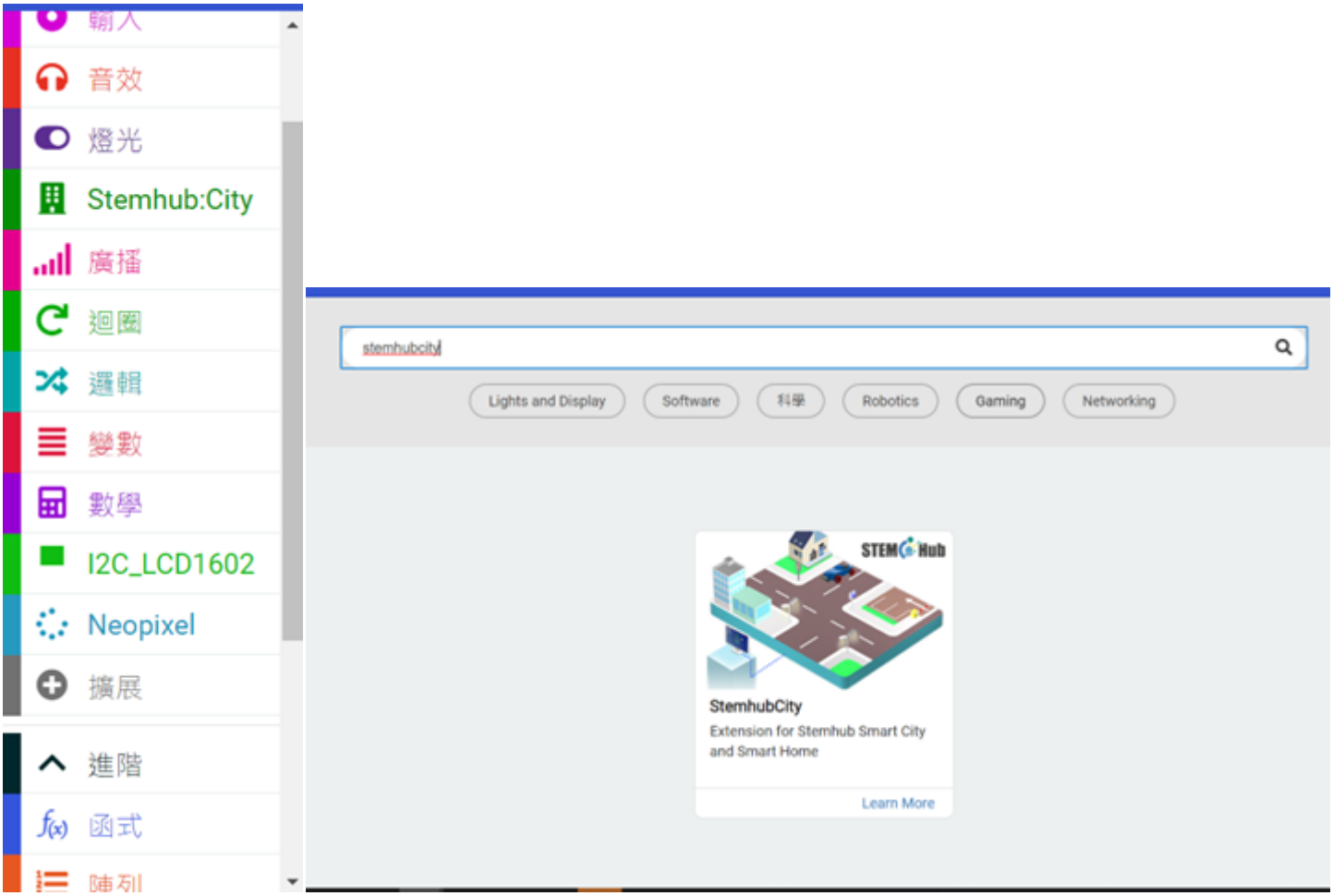
第九步

Hardware connect 硬件連接

編程

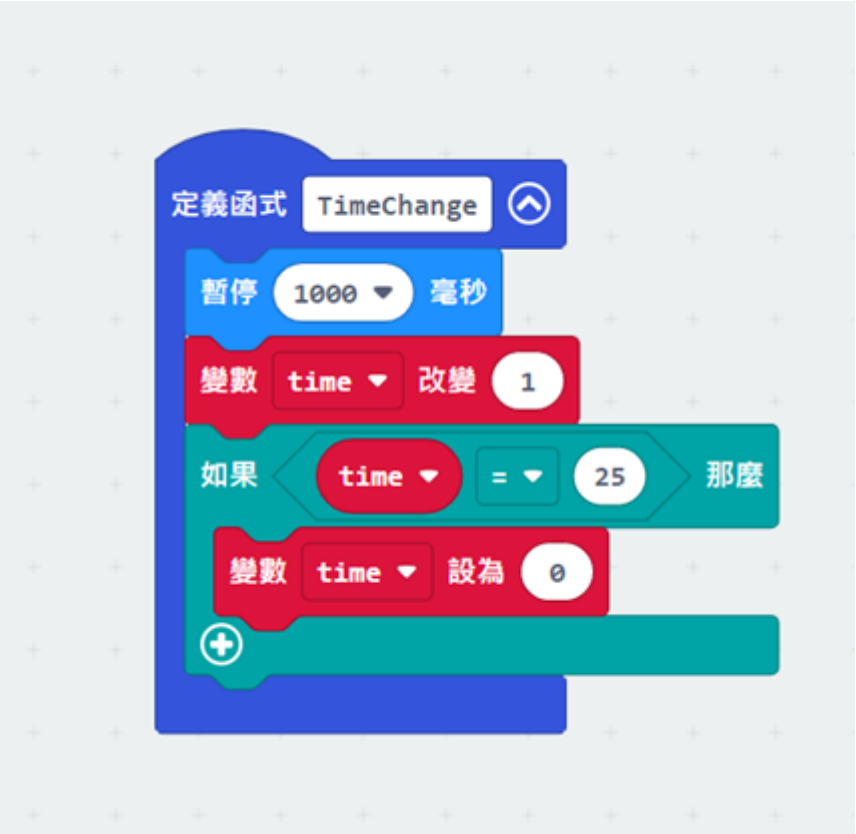
新增函數庫

- 點擊擴展
- 輸入 StemhubCity 後點擊函數庫



更新時間。

- 暫停 1 秒
- 然後將 time 增加 1
- 如果 time 等於 25 則將 time 設F 0

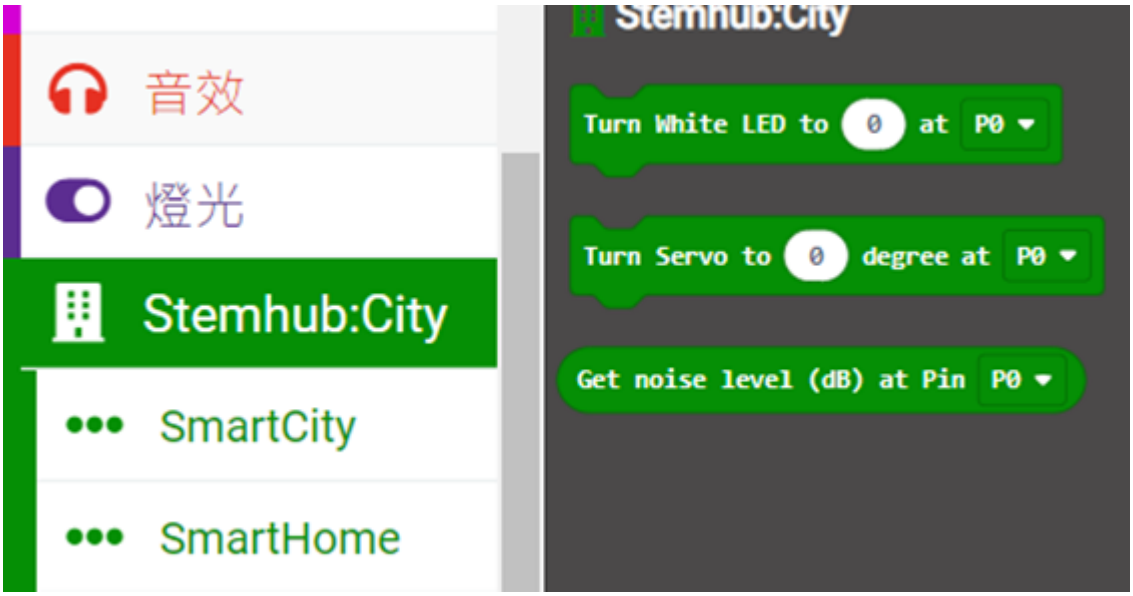


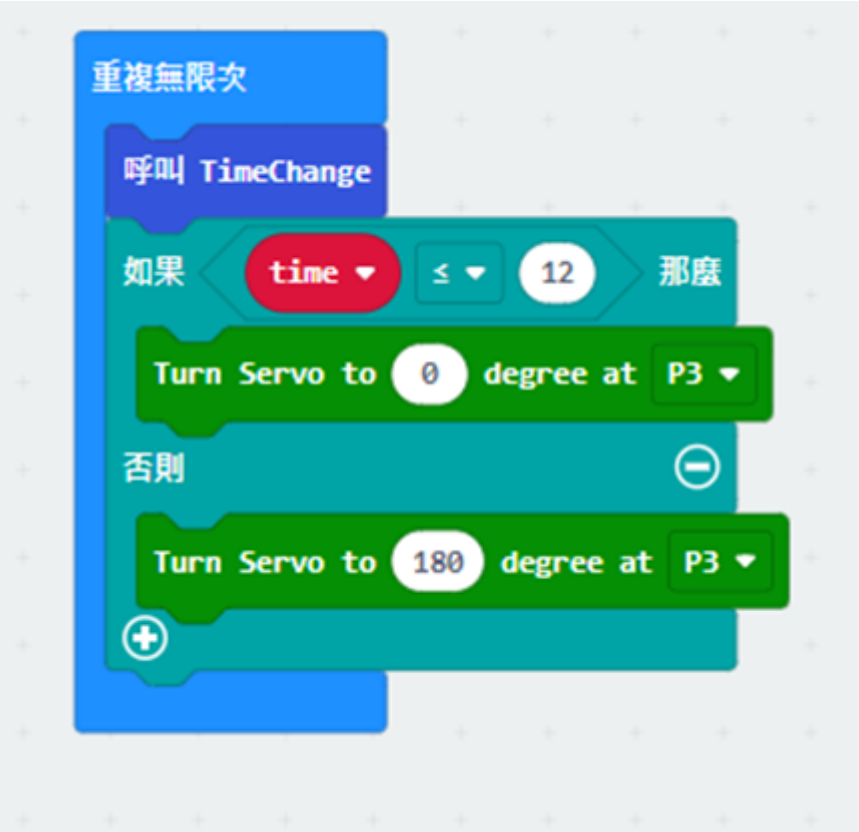
初始化時間

- 將 time 設F 0

檢查時間及更改路牌

- 呼叫函數 TimeChange 以改變時間
- 在 stemhubcity 函庫拖出兩個 turn servo to 0 degree at P0
- 分🅵把 turn servo to 0 degree at P0 放在如果 time <= 12 中及否則中
- 然後按下圖分🅵把 turn servo to 0 degree at P0 改🅵 turn servo to 0 degree at P3 及 turn servo to 180 degree at P3




重複無限次
呼叫 TimeChange
如果
time
≤
12
那麼
Turn Servo to
0
degree at
P3
否則
Turn Servo to
180
degree at
P3

總結

思考

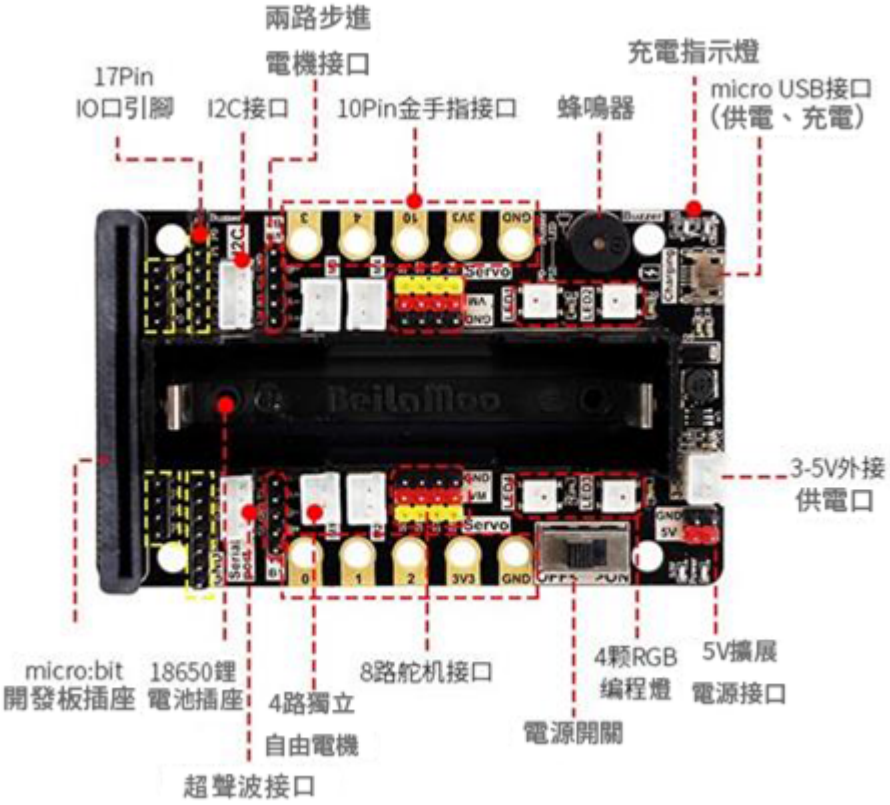
1.3 Micro:bit M1 智能小車 _ 初學者

1.3.1 第一課

簡介

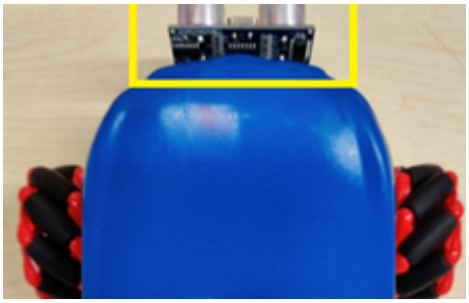
教學目標

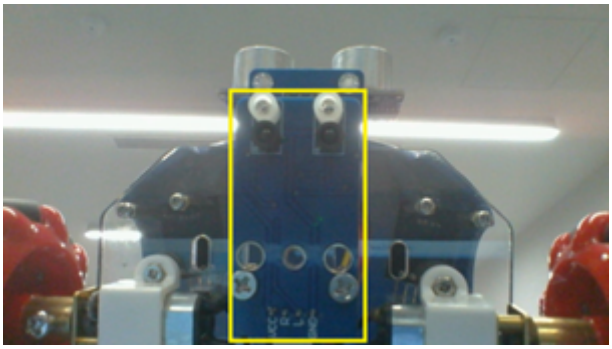
認識 micro:bit 擴展板



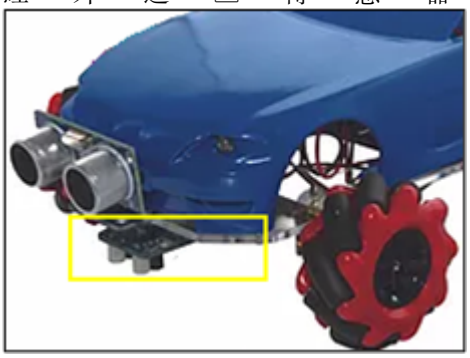
認識麥克納姆輪車子

- 4 個馬達
- 4 個麥克納姆輪子
- Micro bit

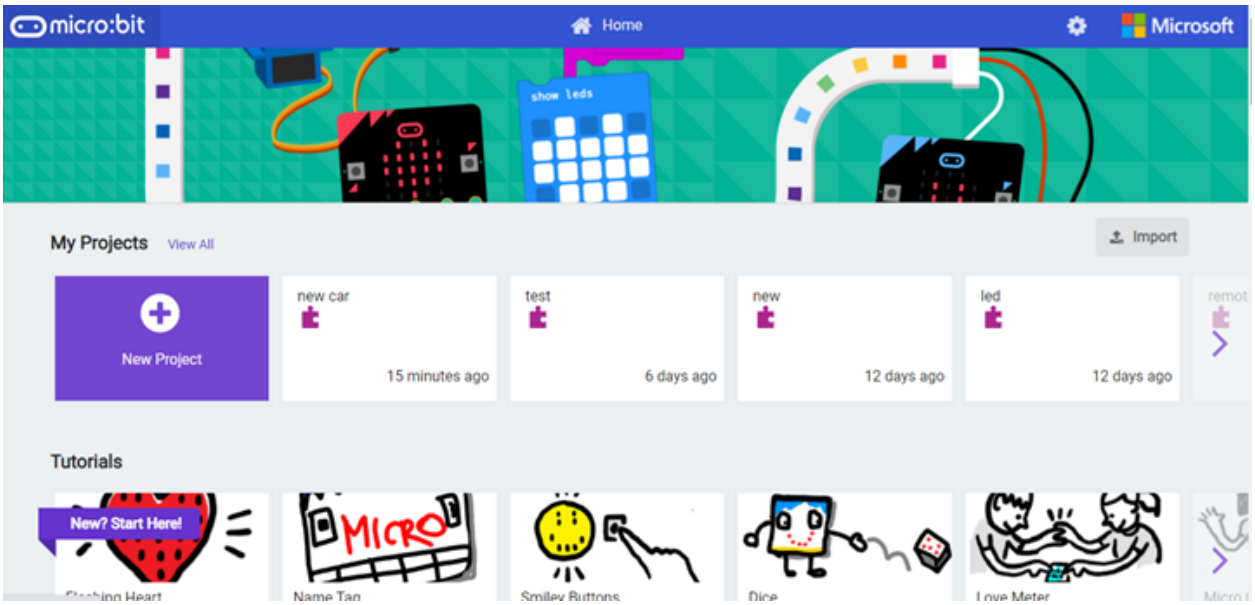
- Micro bit 擴展板
- 超聲波傳感器 

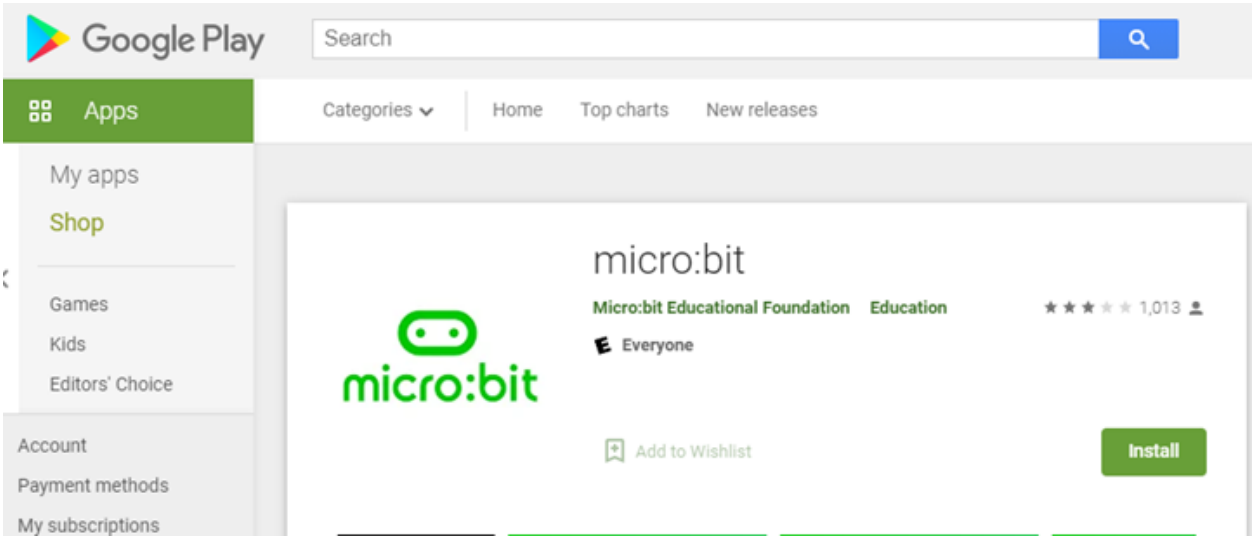


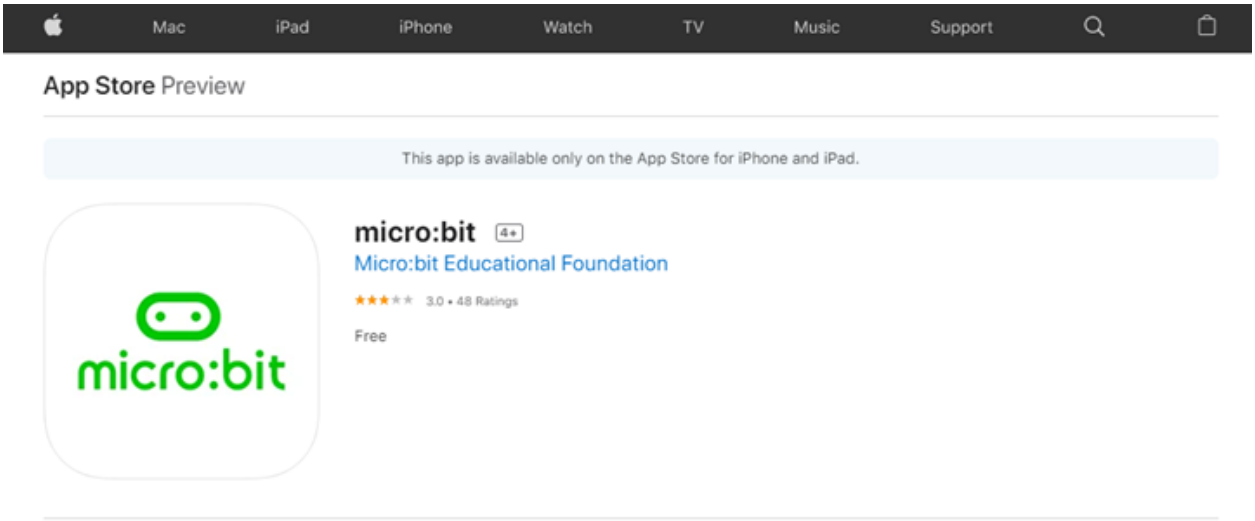
- 紅外巡F傳感器



準備 Micro:bit 編程程式 Makecode

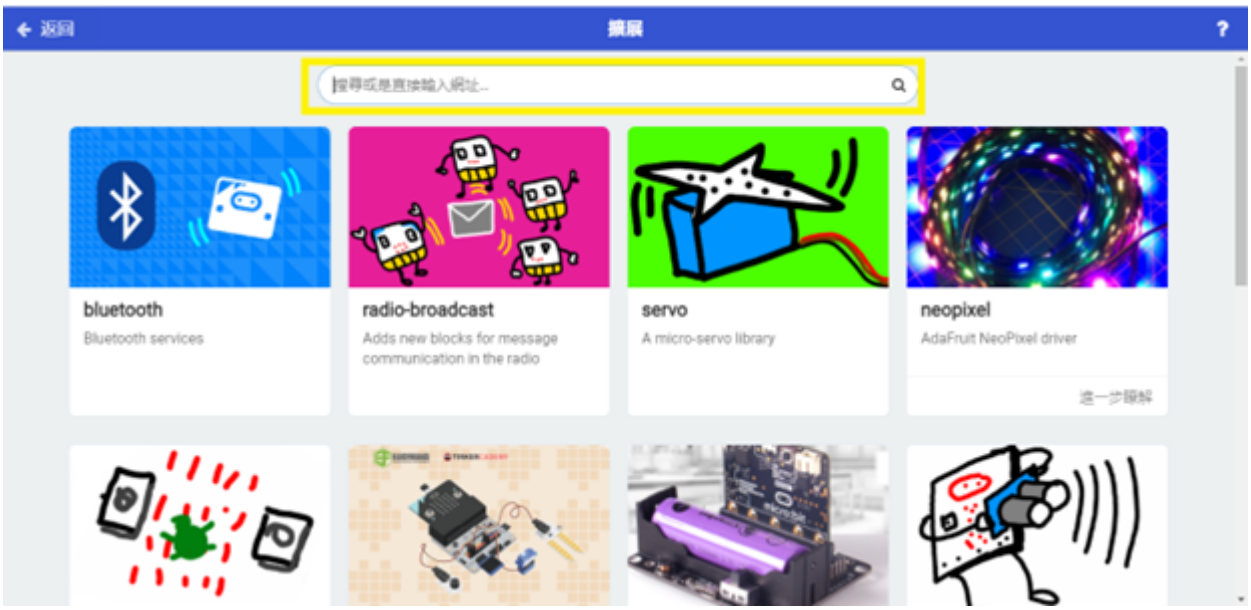





Mac
iPad
iPhone
Watch
TV
Music
Support
App Store Preview
This app is available only on the App Store for iPhone and iPad.
micro:bit
4+
Micro:bit Educational Foundation
3.0 • 48 Ratings
Free
micro:bit

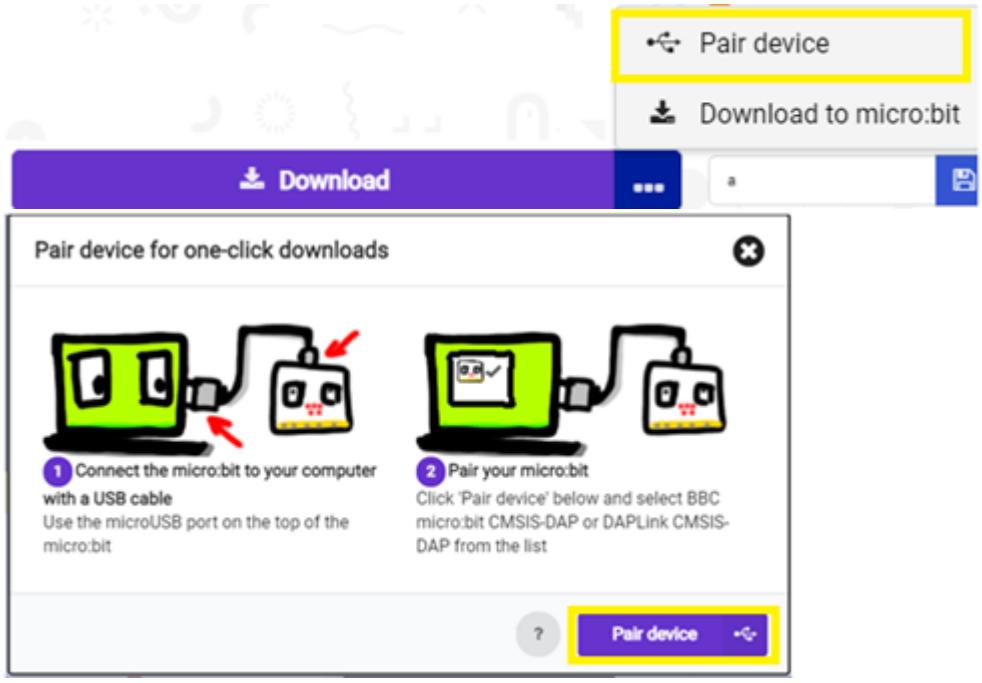
新增擴展

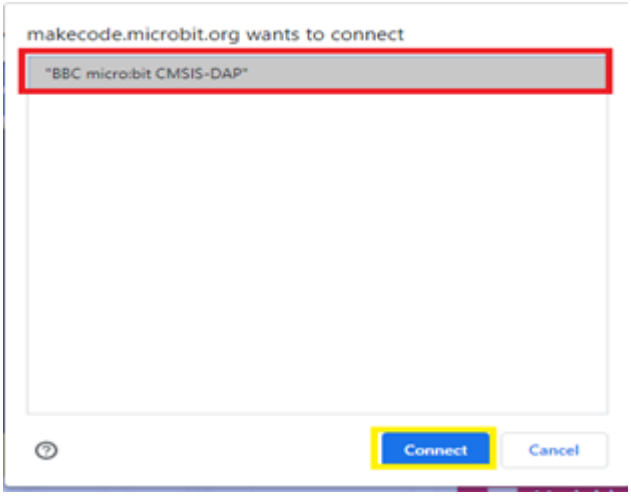
邏輯
變數
數學
進階
函式
陣列
文字
遊戲
圖像
引腳
更多
序列
控制
擴展

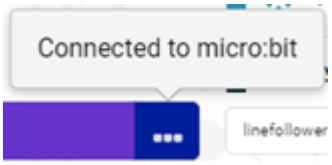

返回
擴展
?
bluetooth
Bluetooth services
radio-broadcast
Adds new blocks for message communication in the radio
servo
A micro-servo library
neopixel
AdaFruit NeoPixel driver

返回
擴展
?
https://github.com/stemhub/pxt-Stemhubbit
stemhubbit
Extension for Stemhub Car

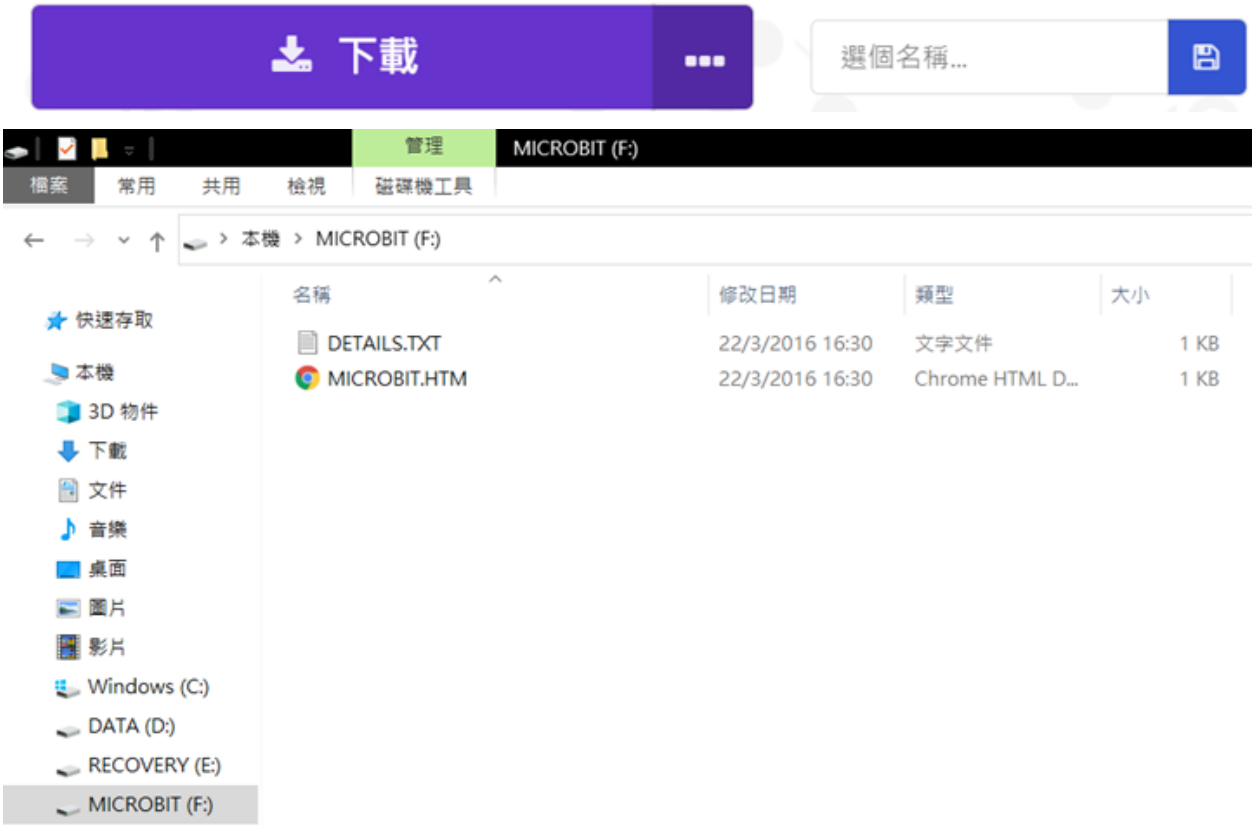
連接 micro:bit 到電腦





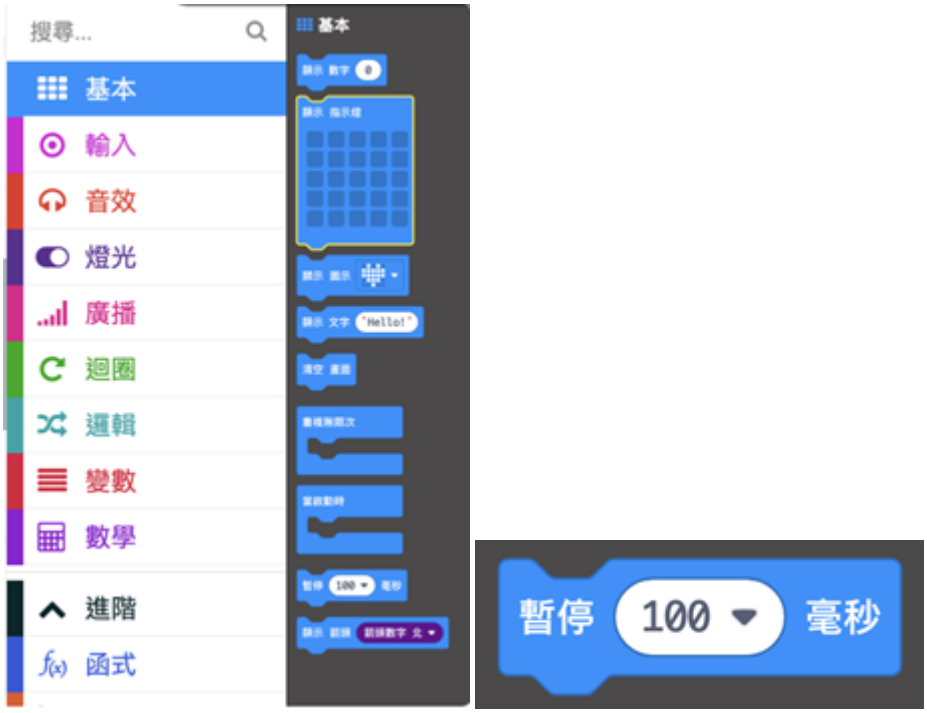


儲存程序檔案（.hex) 到 micro:bit

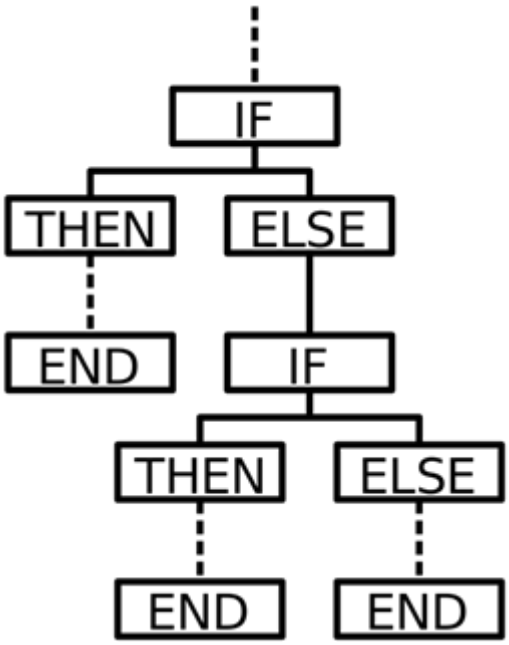


認識積木模塊

暫停積木模塊:

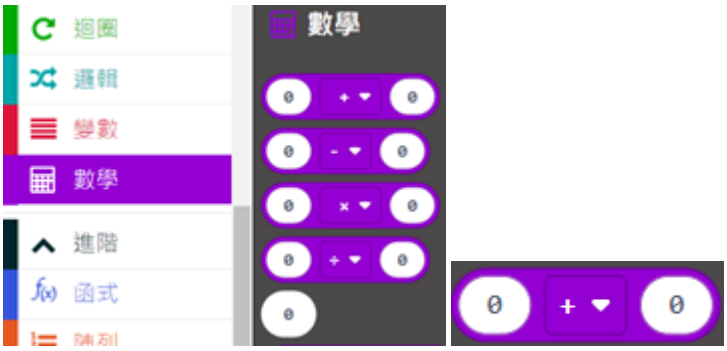


如果 (if) 邏輯積木模塊: 設定模塊Ⓕ程序執行條件



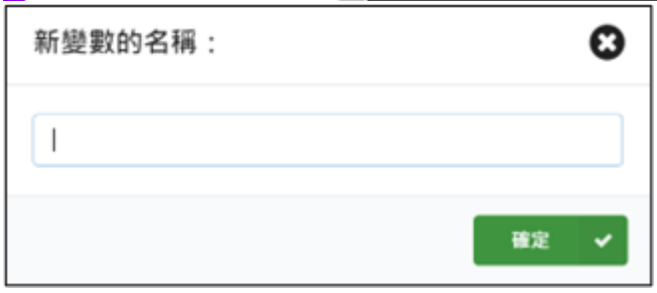
比較積木:

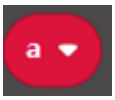
數學運算積木模塊



‘且、或’(AND, OR) 積木模塊

變數 (Variable)



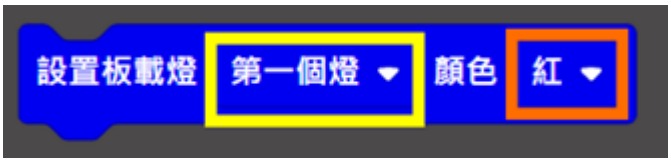


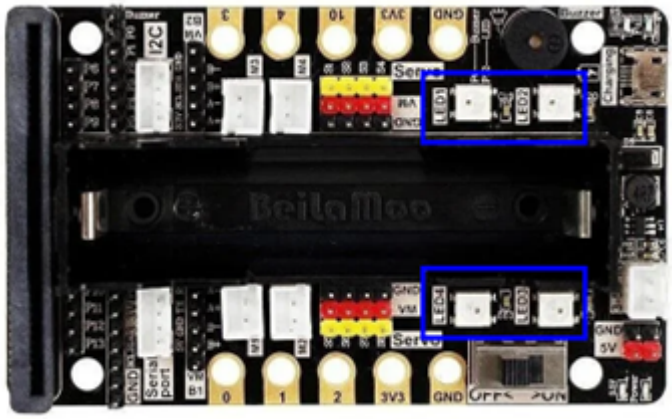
1.3.2 第二課

簡介

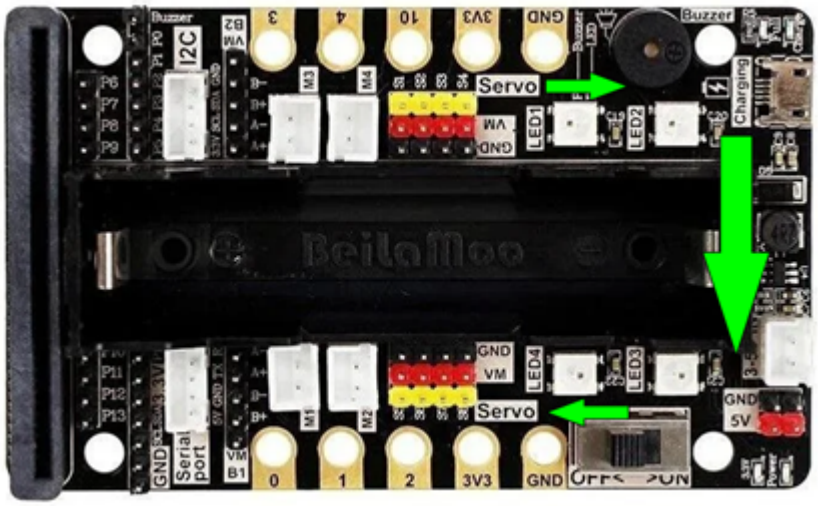
教學目標

擴展板 Rgb 燈的控制 (LED)





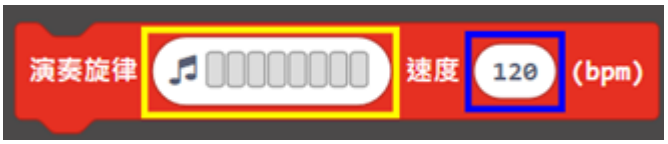
練習 1: 設計程序使擴展板上的 Rgb 燈順序閃爍



練習 2: 設計程序使擴展板上的 Rgb 燈亮度由小至大再到大至小（呼吸燈效果）

蜂鳴器

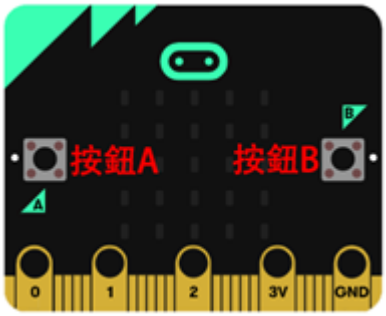
搜尋...
基本
輸入
音效
燈光
HuskyLens
廣播
迴圈
邏輯
變數
音效
旋律 (Melody)
演奏旋律
速度
120
(bpm)
音高 (Tone)
演奏 音階
中音 c
持續
1
拍
演奏 音階
中音 c
演奏 休息
1
拍


演奏旋律
速度
120
(bpm)

演奏 音階
中音 c
持續
1
拍

音量設為
127

顯示文字的延遲

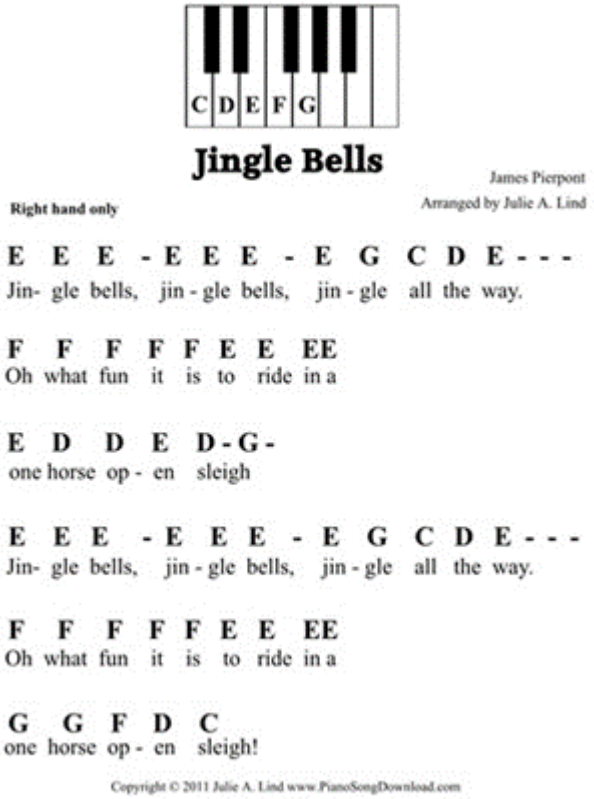


練習: 播放歌曲, 設計程序使小車能Ⓕ播放以下歌曲

1: "Twinkle Twinkle Little Star"

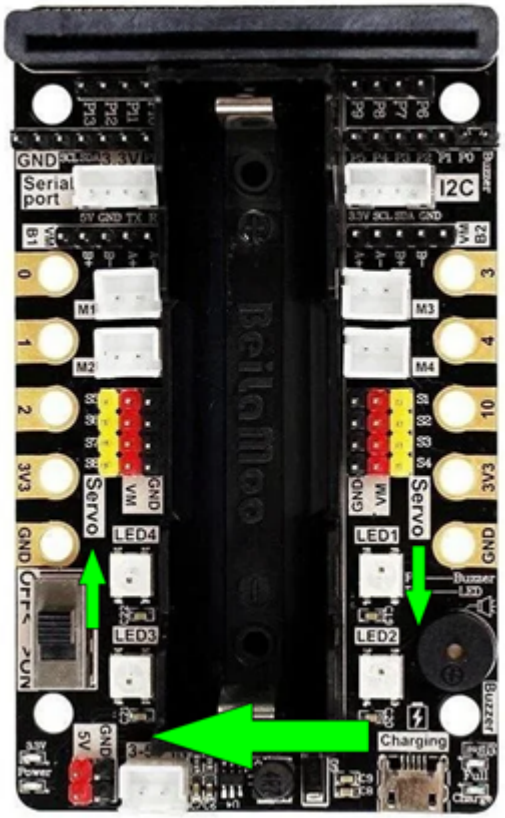
(rh):	1	1	5	5	5	5	5
	G	G	D	D	E	E	D
(lh):	5	5	1	1	1	1	1
	Twin -	kle	twin -	kle	lit -	tle	star
	4	4	3	3	2	2	1
	C	C	B	B	A	A	G
	2	2	3	3	4	4	5
	how	I	won -	der	what	you	are
	5	5	4	4	3	3	2
	D	D	C	C	B	B	A
	1	1	2	2	3	3	4
	up	a -	bove	the	world	so	high
	5	5	4	4	3	3	2
	D	D	C	C	B	B	A
	1	1	2	2	3	3	4
	like	a	dia -	mond	in	the	sky
	1	1	5	5	5	5	5
	G	G	D	D	E	E	D
	5	5	1	1	1	1	1
	Twin -	kle	twin -	kle	lit -	tle	star

2: “Jingle Bells”



答案

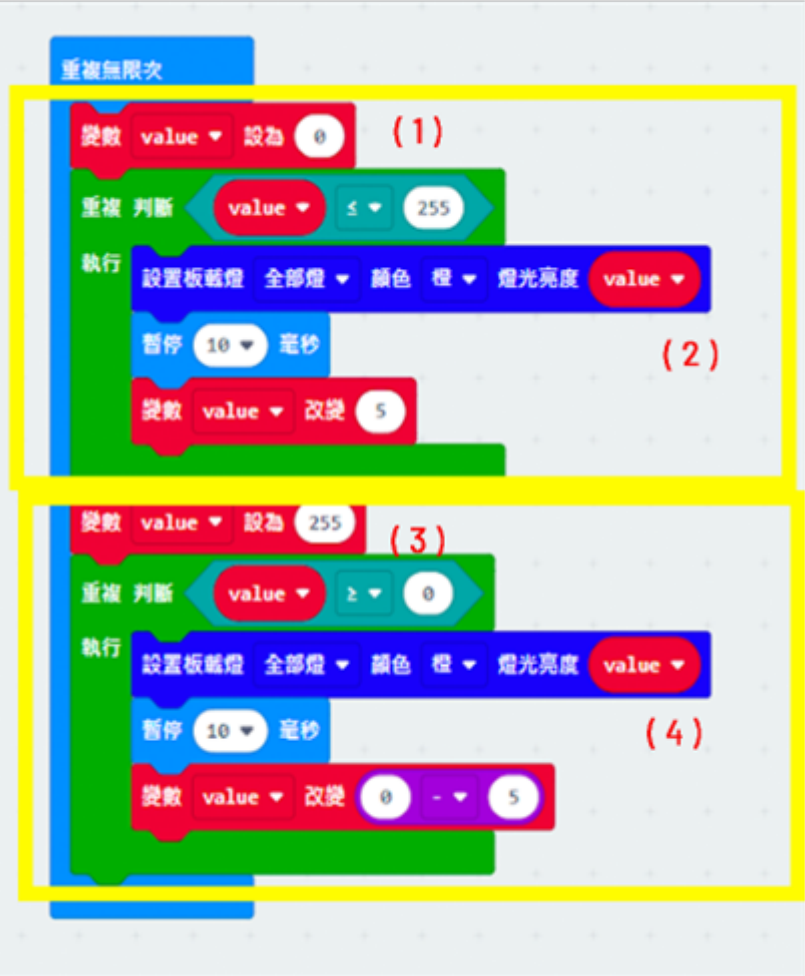
練習 1:



練習 2: 呼吸燈

```
a.        把全部燈改成橙色，燈光亮度會隨著 value 改變
b.        暫停 0.01 秒
c.        value 增加 5
```

```
a.        把全部燈改成橙色，燈光亮度會隨著 value 改變
b.        暫停 0.01 秒
c.        value F少 5
d.        程式在 value 少於 0 的情F下結束
```

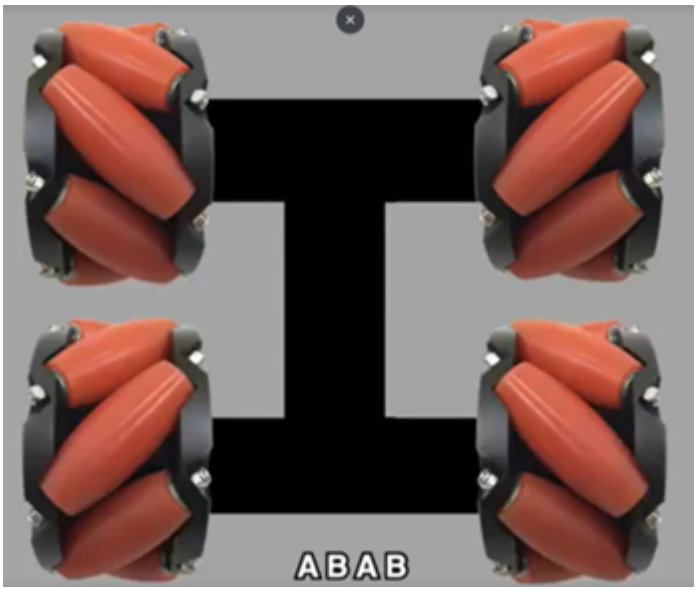

重複無限次
變數 value 設為 0
(1)
重複 判斷 value ≤ 255
執行
設置板載燈 全部燈 顏色 橙 燈光亮度 value
暫停 10 毫秒
(2)
變數 value 改變 5
變數 value 設為 255
(3)
重複 判斷 value ≥ 0
執行
設置板載燈 全部燈 顏色 橙 燈光亮度 value
暫停 10 毫秒
(4)
變數 value 改變 0 - 5

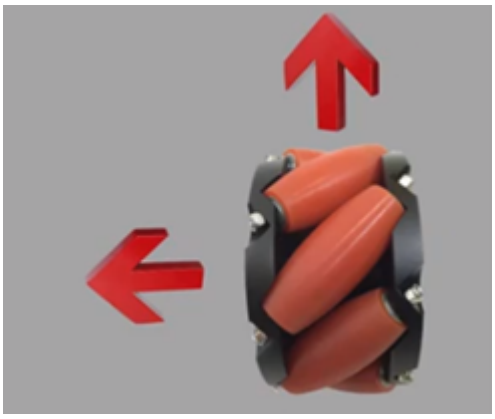
1.3.3 第三課

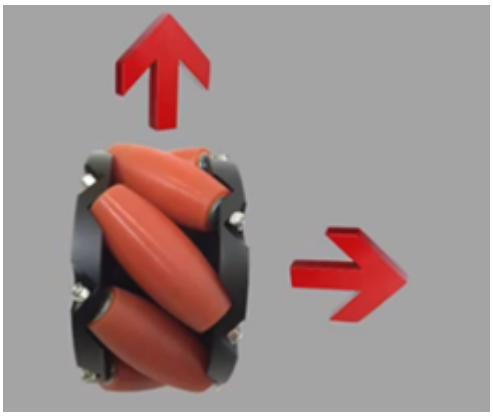
簡介

教學目標

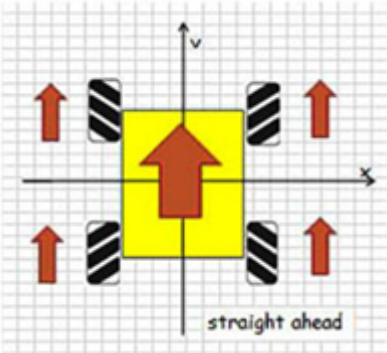
認識麥克納姆輪

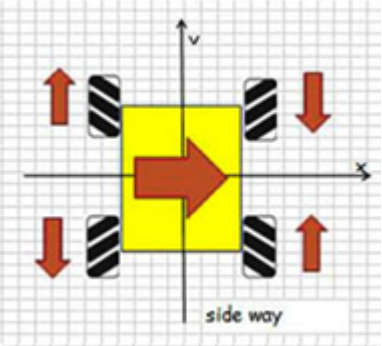


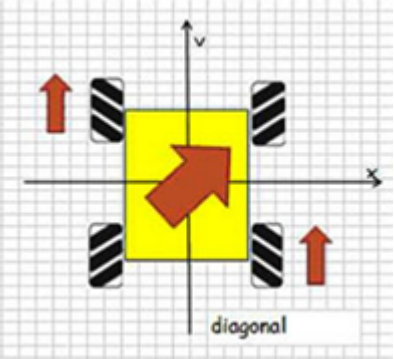


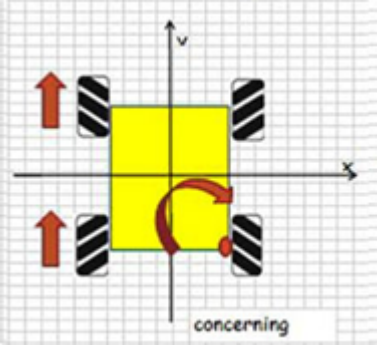


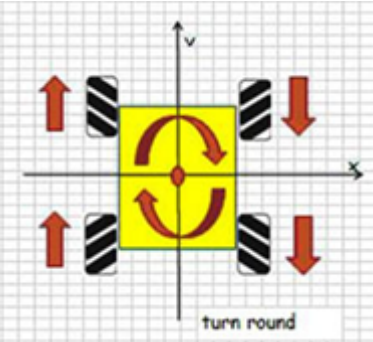
車輪的原理

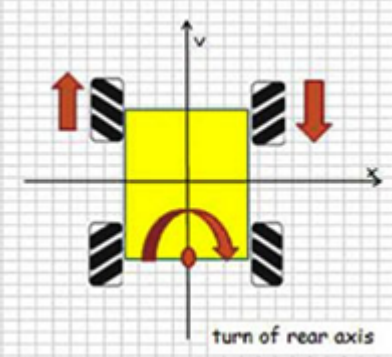






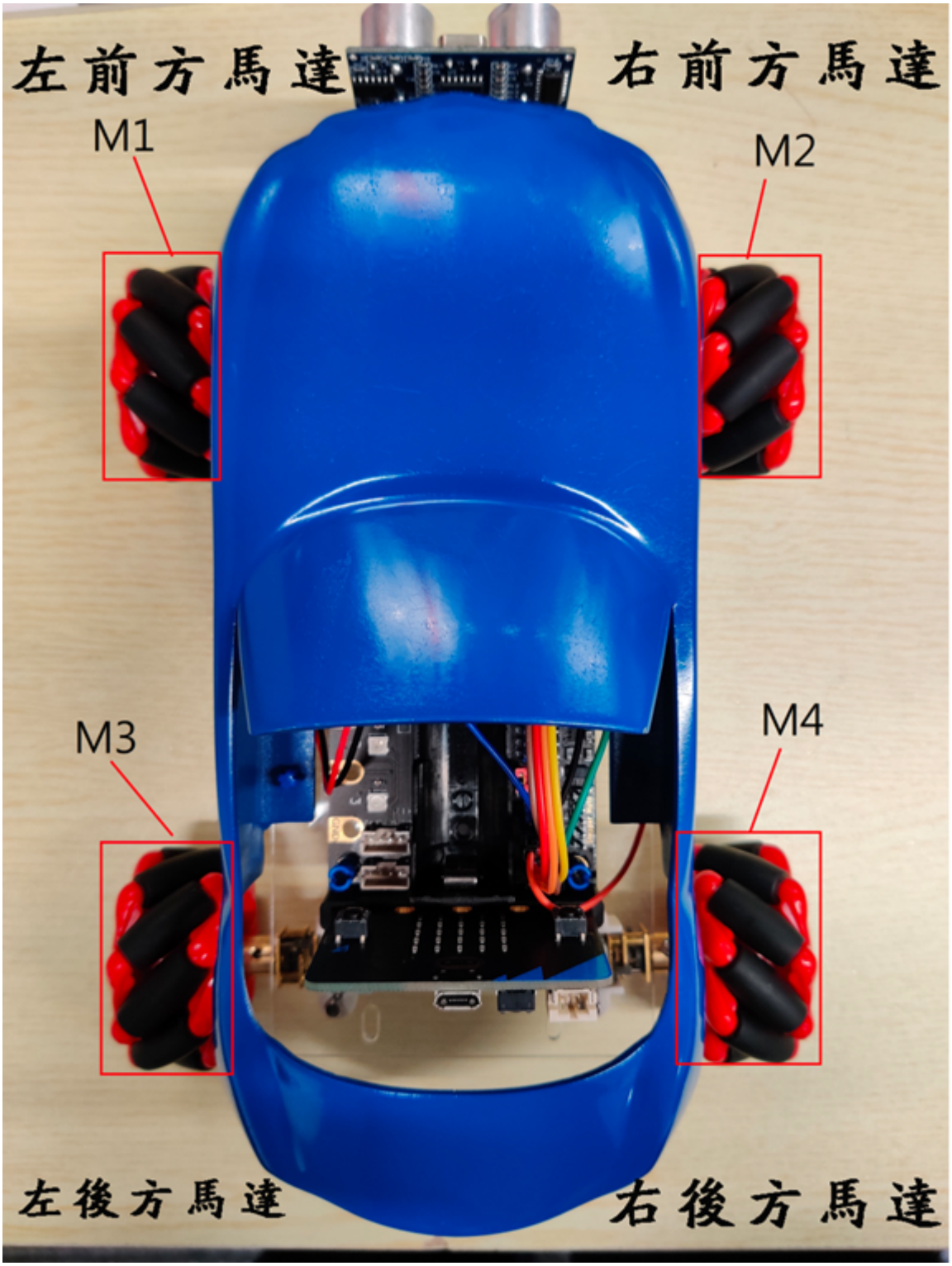






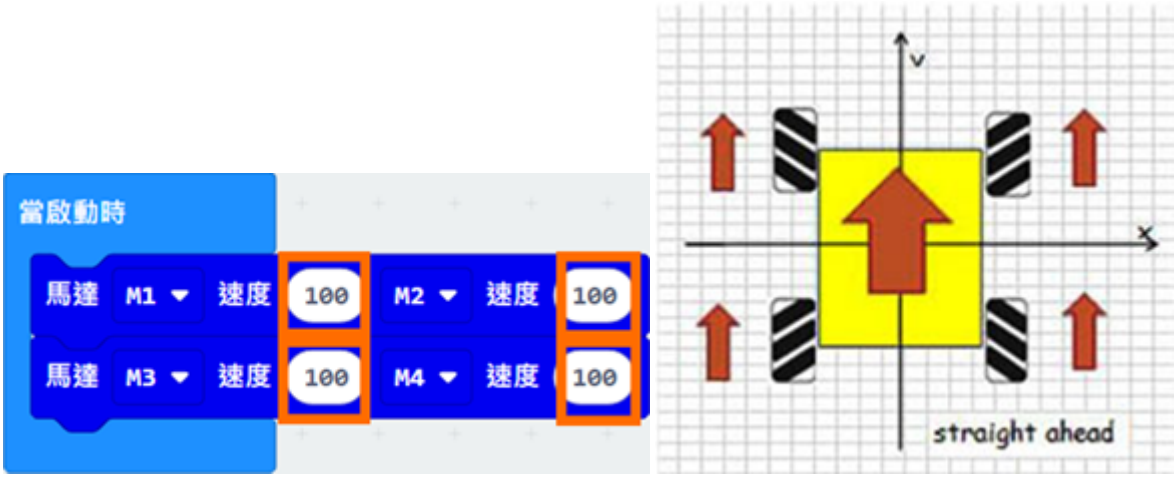
1.3.4 第四課

馬達的分Ⓕ

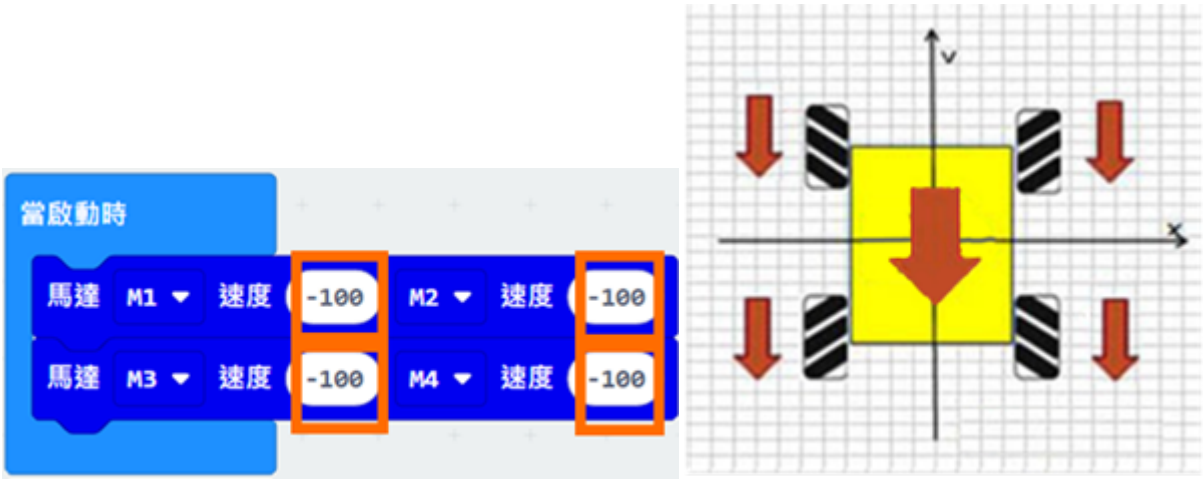


控制馬達的積木模塊

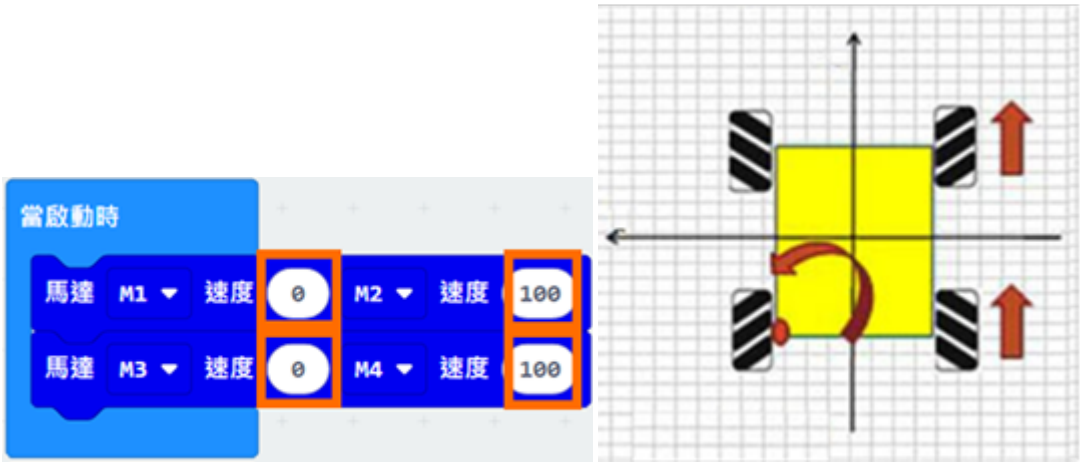
前進



後退

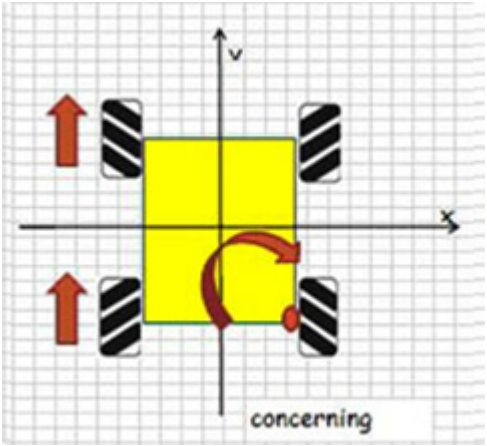

當啟動時
馬達 M1 速度 -100 M2 速度 -100
馬達 M3 速度 -100 M4 速度 -100
y
x

左轉


當啟動時
馬達 M1 速度 0 M2 速度 100
馬達 M3 速度 0 M4 速度 100

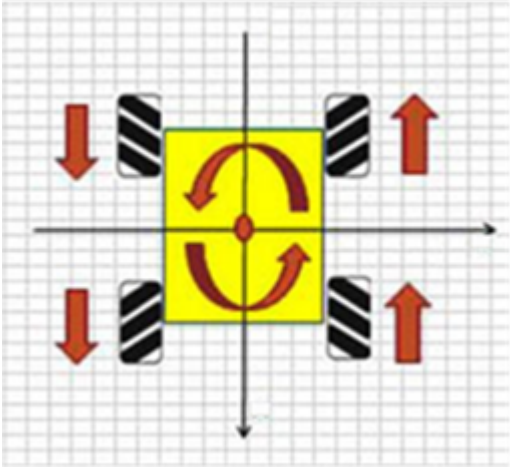
右轉

當啟動時
馬達 M1 速度 100 M2 速度 0
馬達 M3 速度 100 M4 速度 0

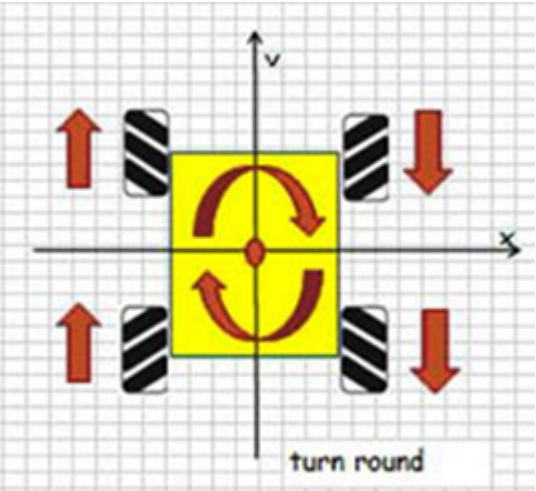

y
x
concerning

左旋轉

當啟動時
馬達 M1 速度 -100 M2 速度 100
馬達 M3 速度 -100 M4 速度 100

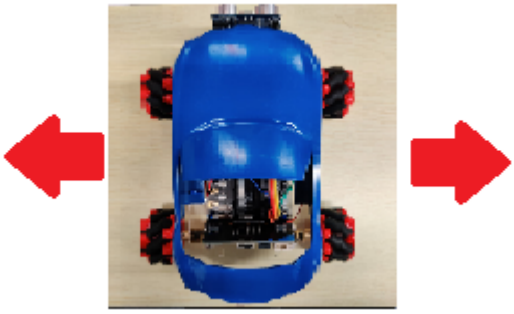


右旋轉



總結

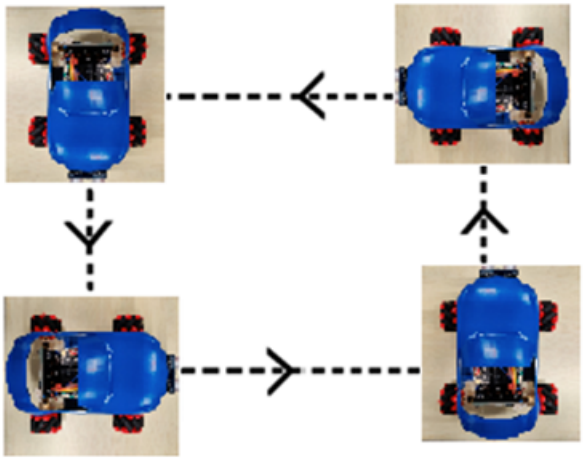
練習 1: 車子的水平移動



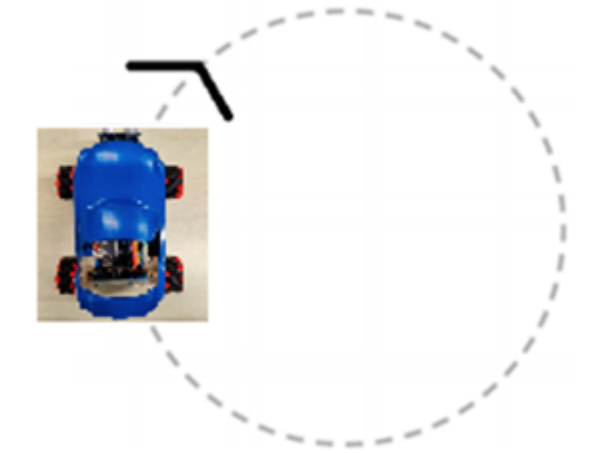
練習 2: 小車的斜向移動

練習 3: 小車的不同方式的行駛

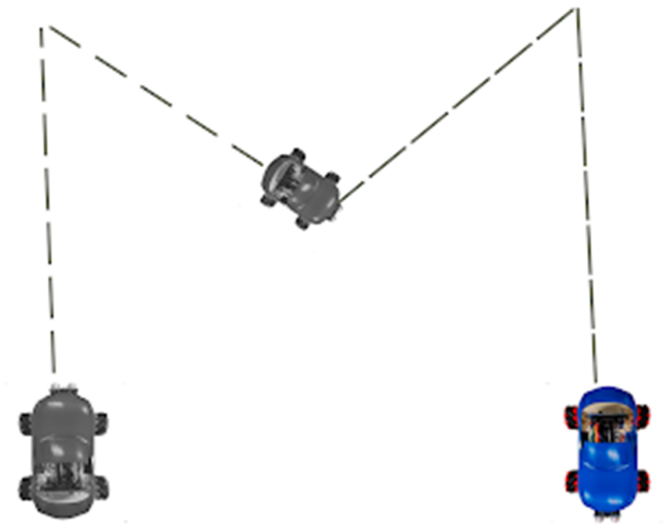
正方形繞圈行駛:



圓形繞圈行駛:



M 字型行駛:

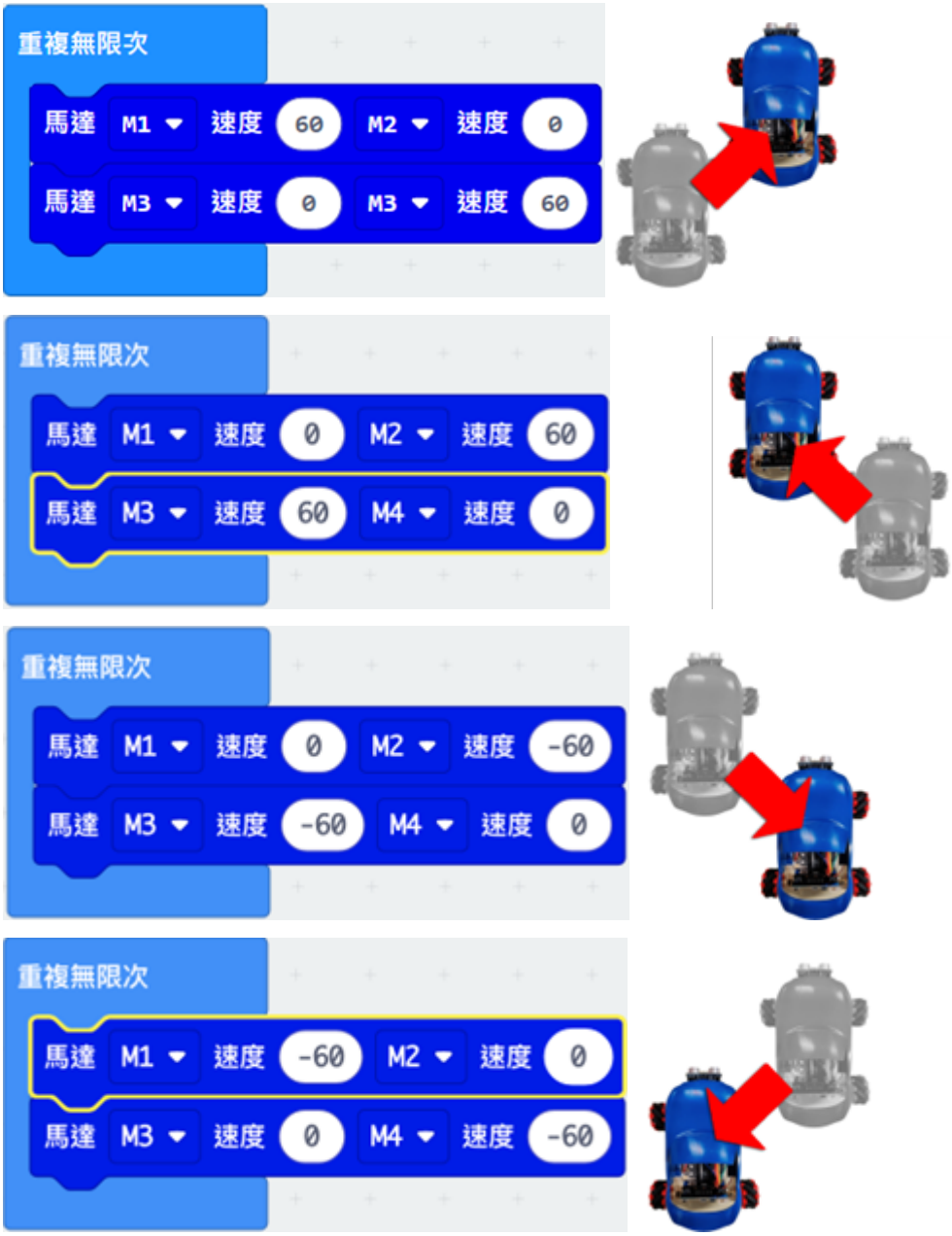


答案

練習 1:

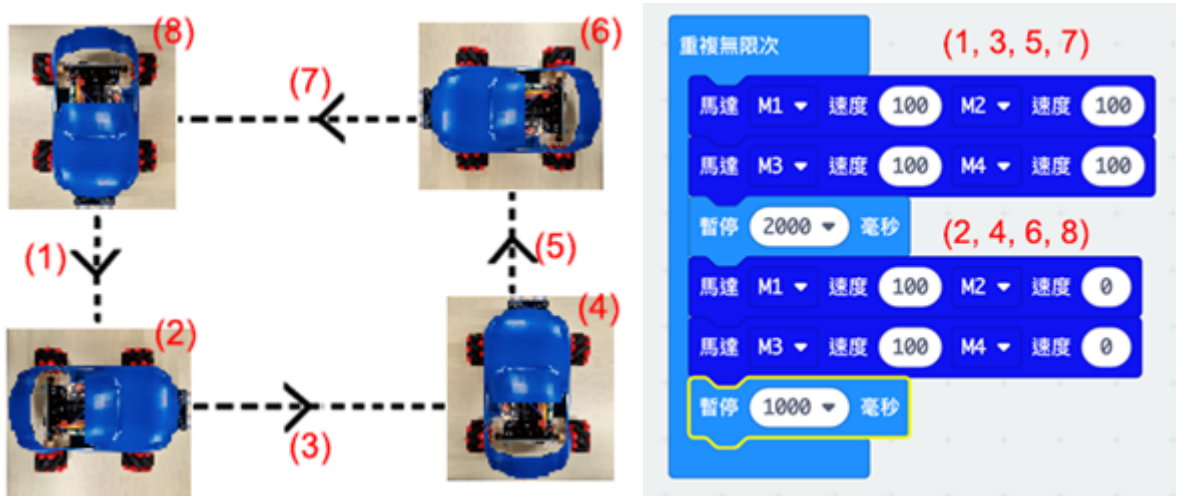
當啟動時
馬達 M1 速度 100 M2 速度 -100
馬達 M3 速度 -100 M4 速度 100

練習 2:


重複無限次
馬達 M1 速度 60 M2 速度 0
馬達 M3 速度 0 M3 速度 60
重複無限次
馬達 M1 速度 0 M2 速度 60
馬達 M3 速度 60 M4 速度 0
重複無限次
馬達 M1 速度 0 M2 速度 -60
馬達 M3 速度 -60 M4 速度 0
重複無限次
馬達 M1 速度 -60 M2 速度 0
馬達 M3 速度 0 M4 速度 -60

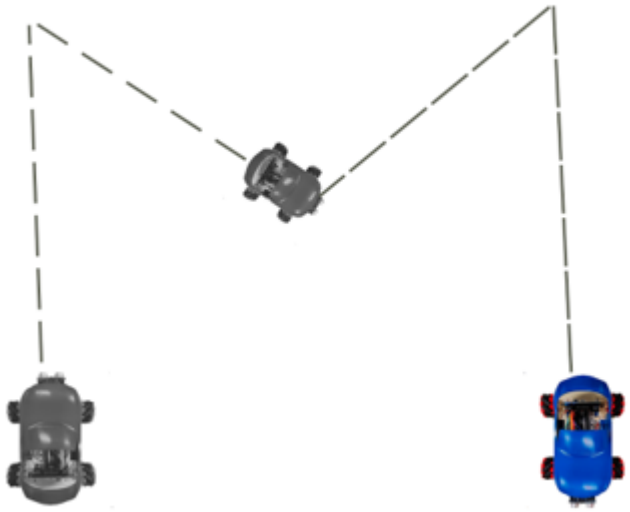
練習 3:

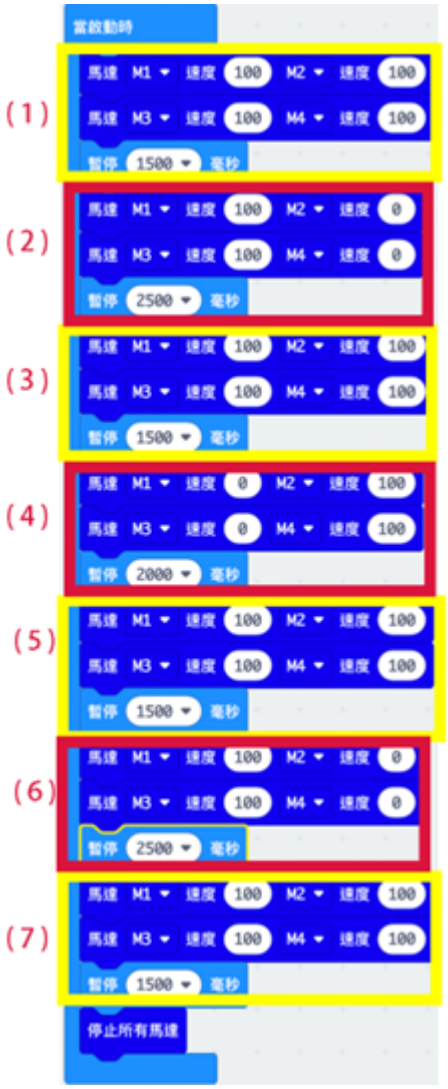
正方形繞圈行駛:



圓形繞圈行駛:

M 字型行駛：



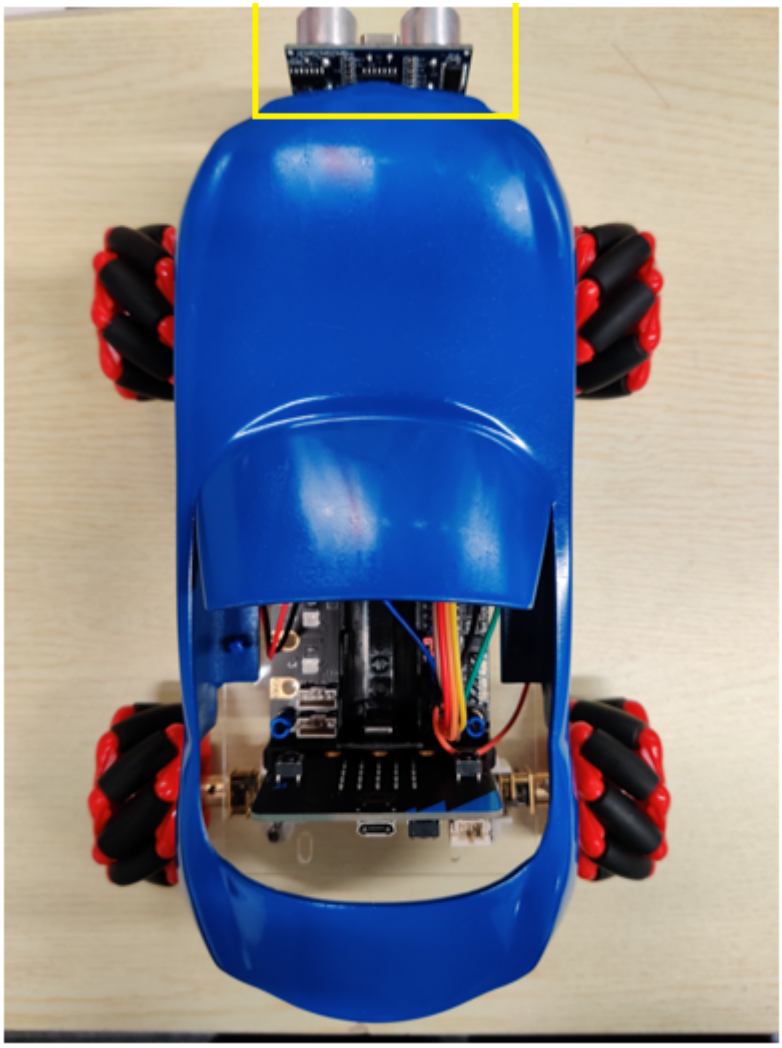

當啟動時
(1)
馬達 M1 速度 100 M2 速度 100
馬達 M3 速度 100 M4 速度 100
暫停 1500 毫秒
(2)
馬達 M1 速度 100 M2 速度 0
馬達 M3 速度 100 M4 速度 0
暫停 2500 毫秒
(3)
馬達 M1 速度 100 M2 速度 100
馬達 M3 速度 100 M4 速度 100
暫停 1500 毫秒
(4)
馬達 M1 速度 0 M2 速度 100
馬達 M3 速度 0 M4 速度 100
暫停 2000 毫秒
(5)
馬達 M1 速度 100 M2 速度 100
馬達 M3 速度 100 M4 速度 100
暫停 1500 毫秒
(6)
馬達 M1 速度 100 M2 速度 0
馬達 M3 速度 100 M4 速度 0
暫停 2500 毫秒
(7)
馬達 M1 速度 100 M2 速度 100
馬達 M3 速度 100 M4 速度 100
暫停 1500 毫秒
停止所有馬達

1.3.5 第五至六課

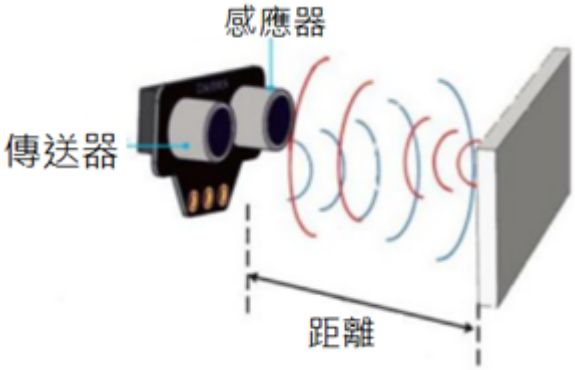
簡介:

教學目標:

超聲波避障

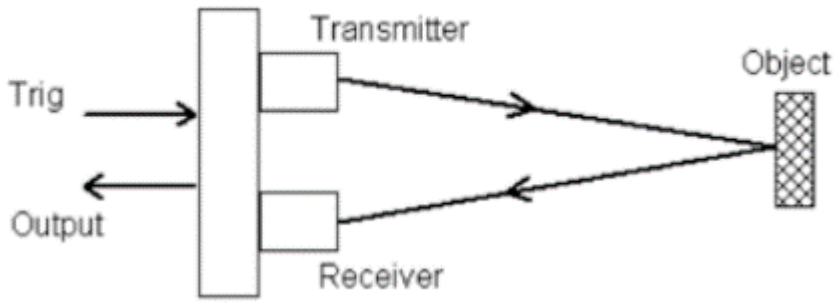


超聲波模塊的原理



計算公式如下:

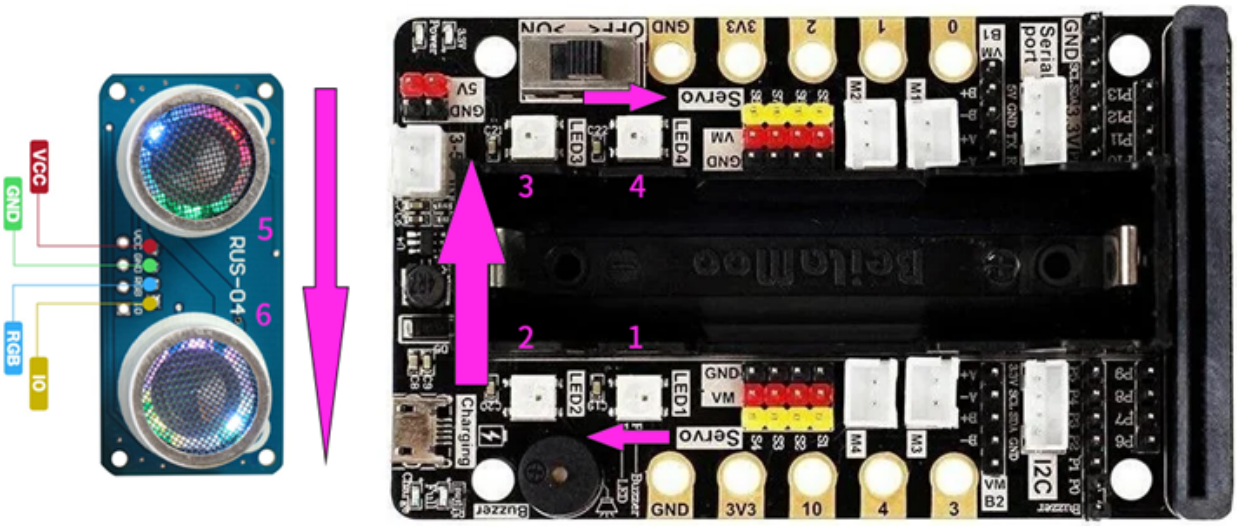
$$\frac{時間 \times 0.0346(cm/\mu s)}{2}$$



Stemhub:bit 超聲波

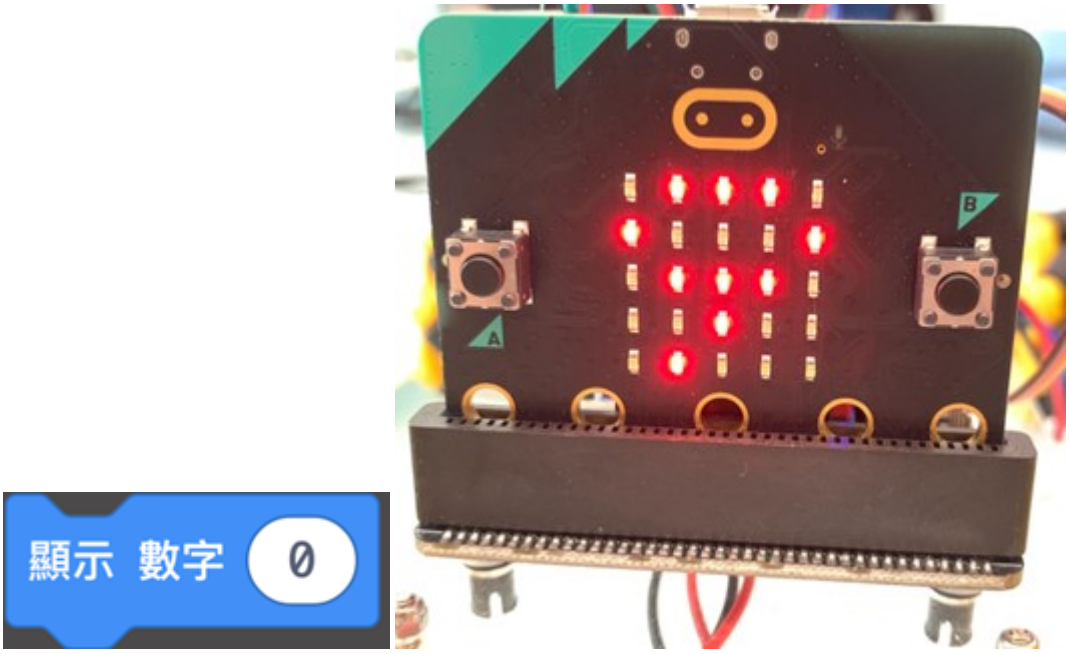
左
右
RUS-04
VCC GND RGB IO
VCC
GND
RGB
IO

練習 1:


VCC
GND
RGB
IO
RUS-04
5
6
3
4
2
1
Servo
LED1
LED2
LED3
LED4
GND
VM
Charging
Buzzer
Serial port
I2C
M1
M2
M3
M4
B1
B2
GND
3V3
10
4
3

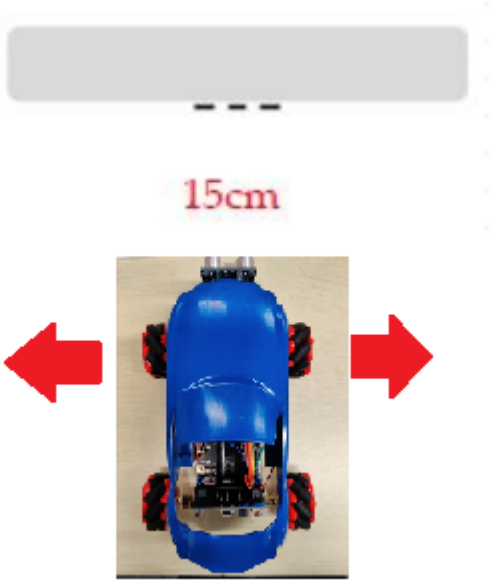
設置板載燈
第一個燈
顏色
紅
燈光亮度
0

練習 2: 設計程式令超聲波讀取的距離顯示在 micro:bit 板上

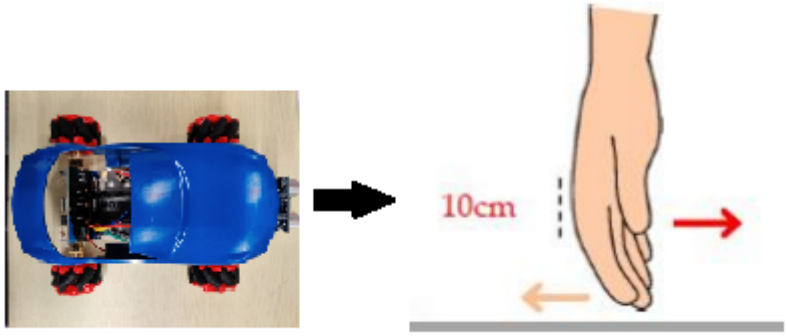


練習 3: 設計程式把超聲波讀取的距離轉Ⓕ成歌曲，Ⓕ顯示不同顏色的燈

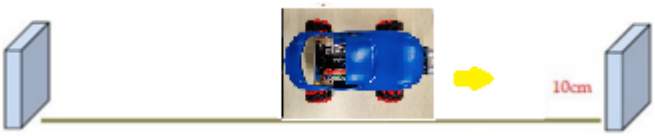
練習 4:



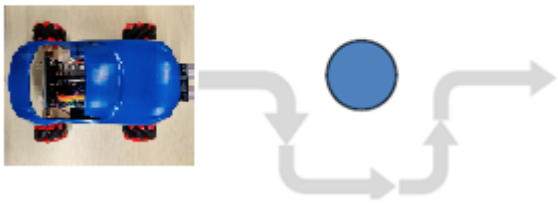
練習 5:



練習 6:

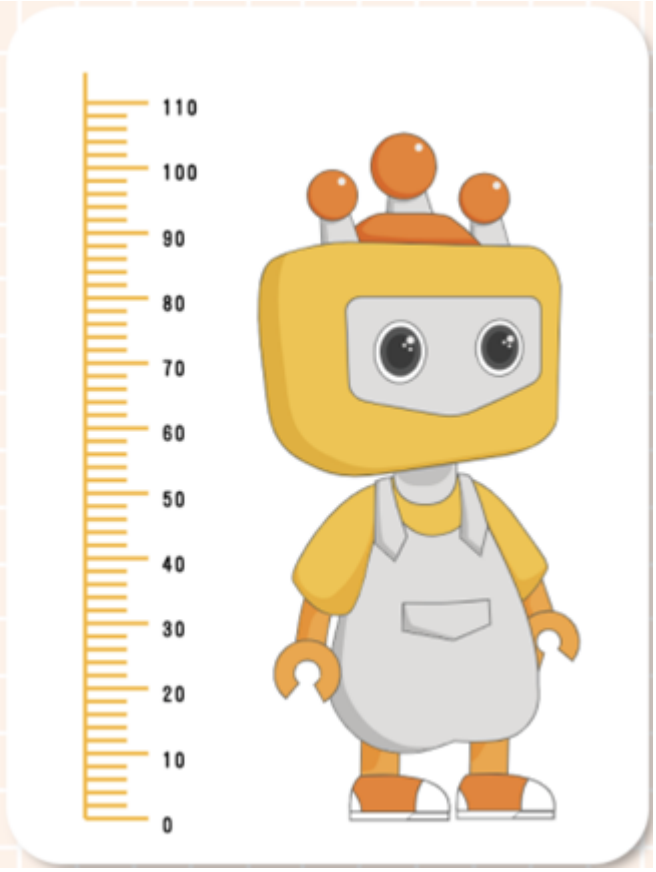


練習 7:



練習 8: 簡易版迷Ⓕ

思考



答案

練習 1:

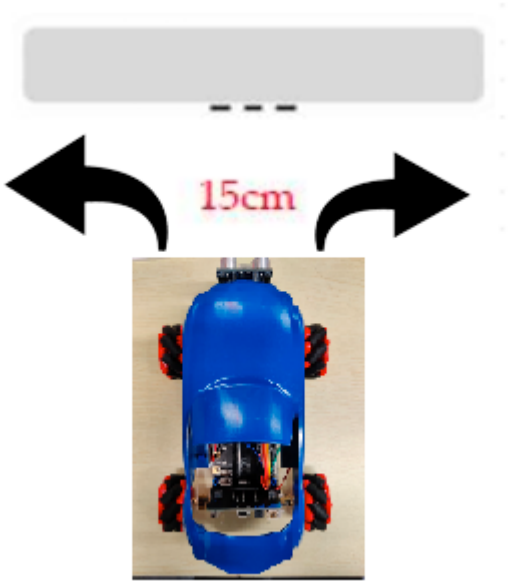
重複無限次
設置板載燈 第一個燈 顏色 藍 燈光亮度 255
暫停 100 毫秒
設置板載燈 第一個燈 顏色 藍 燈光亮度 0
設置板載燈 第二個燈 顏色 藍 燈光亮度 255
暫停 100 毫秒
設置板載燈 第二個燈 顏色 藍 燈光亮度 0
設置板載燈 第三個燈 顏色 藍 燈光亮度 255
暫停 100 毫秒
設置板載燈 第三個燈 顏色 藍 燈光亮度 0
設置板載燈 第四個燈 顏色 藍 燈光亮度 255
暫停 100 毫秒
設置板載燈 第四個燈 顏色 藍 燈光亮度 0
暫停 100 毫秒
設置超聲波燈 左邊 顏色 藍
暫停 100 毫秒
設置超聲波燈 右邊 顏色 藍

練習 2:

練習 3:

重複無限次
變數 距離 設為 RGB超聲波讀取距離(cm)
如果 距離 ≤ 4 那麼
演奏 音階 中音 C 持續 1 拍
設置超聲波燈 全部 顏色 紅
否則如果 距離 ≤ 8 那麼
演奏 音階 中音 D 持續 1 拍
設置超聲波燈 全部 顏色 橙
否則如果 距離 ≤ 12 那麼
演奏 音階 中音 E 持續 1 拍
設置超聲波燈 全部 顏色 黃
否則如果 距離 ≤ 16 那麼
演奏 音階 中音 F 持續 1 拍
設置超聲波燈 全部 顏色 綠
否則如果 距離 ≤ 20 那麼
演奏 音階 中音 G 持續 1 拍
設置超聲波燈 全部 顏色 藍
否則
設置超聲波燈 全部 顏色 黑
暫停 100 毫秒
變數 距離 設為 0

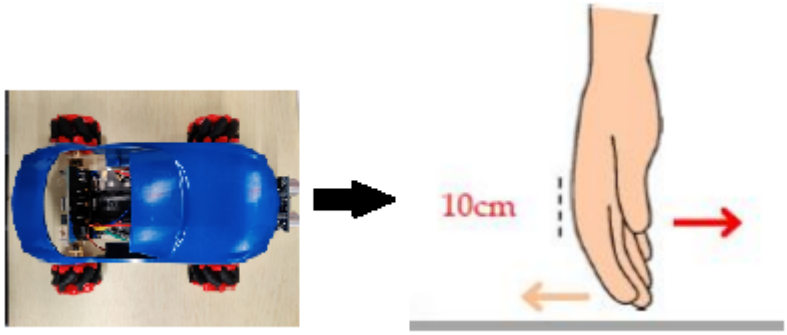
練習 4:


15cm

重複無限次
如果
RGB超聲波讀取距離(cm)
<
15
那麼
馬達 M1 速度 0 M2 速度 60
馬達 M3 速度 0 M4 速度 60
左轉
否則
馬達 M1 速度 60 M2 速度 60
馬達 M3 速度 60 M4 速度 60
直行

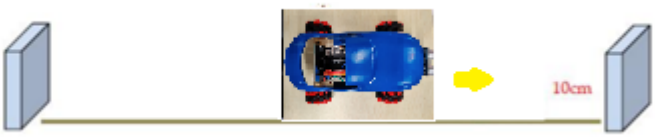
重複無限次
如果
RGB超聲波讀取距離(cm)
<
15
那麼
馬達 M1 速度 60 M2 速度 0
馬達 M3 速度 60 M4 速度 0
右轉
否則
馬達 M1 速度 60 M2 速度 60
馬達 M3 速度 60 M4 速度 60
直行

練習 5:


10cm

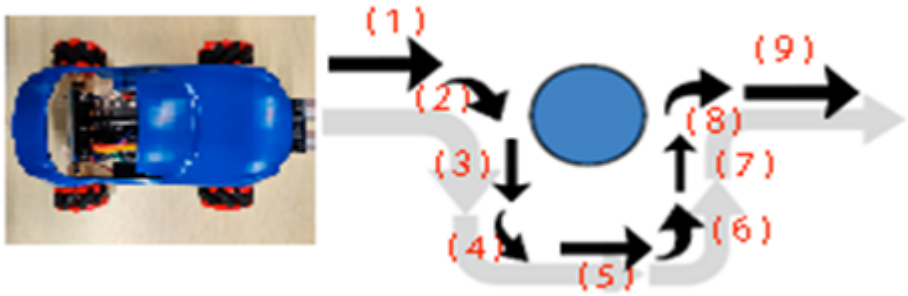
重複無限次
如果 RGB超聲波讀取距離(cm) > 10 那麼
馬達 M1 速度 60 M2 速度 60
馬達 M3 速度 60 M4 速度 60
直 行
否則如果 RGB超聲波讀取距離(cm) < 10 那麼
馬達 M1 速度 -60 M2 速度 -60
馬達 M3 速度 -60 M4 速度 -60
倒 退
否則如果 RGB超聲波讀取距離(cm) = 10 那麼
停止所有馬達
暫 停

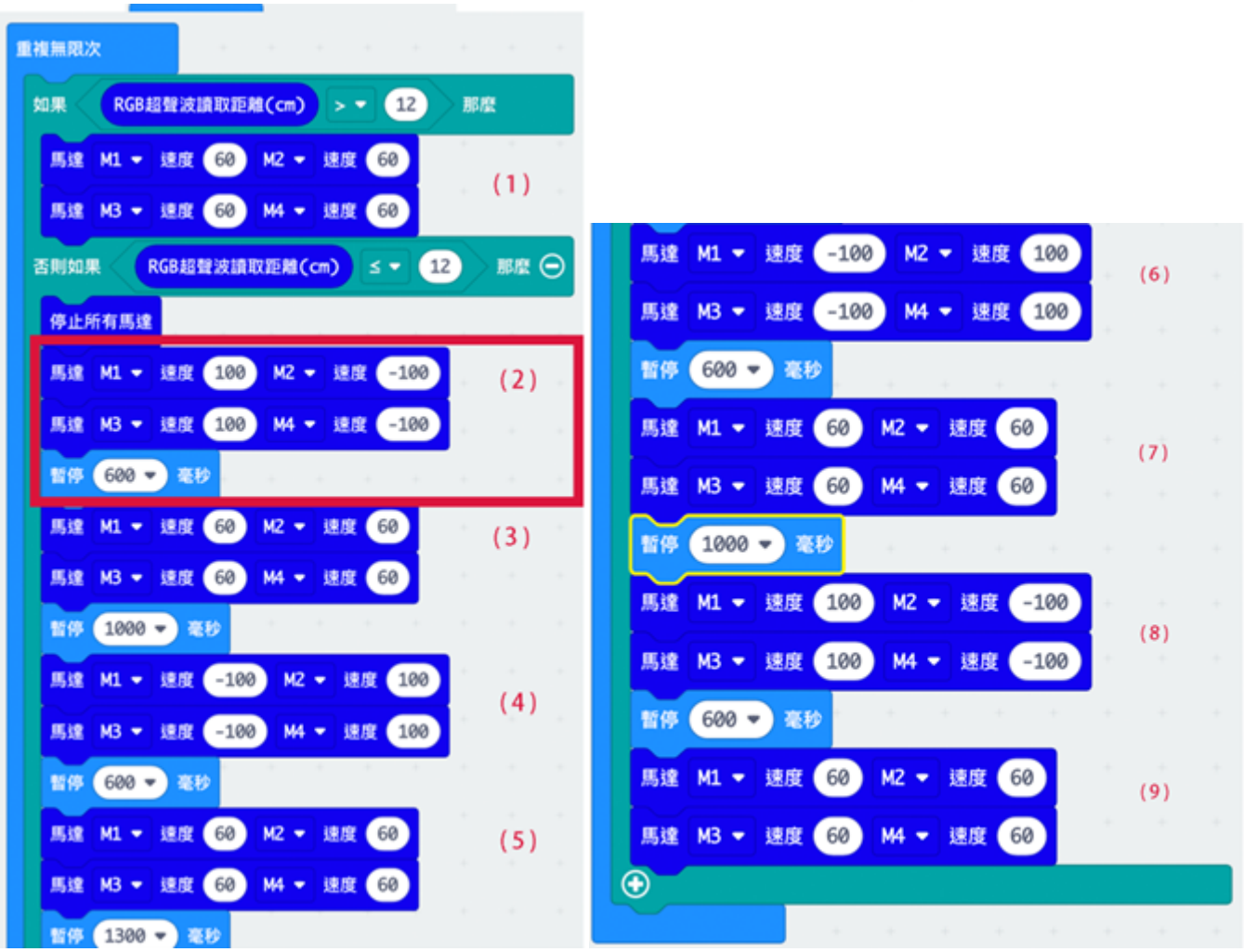
練習 6:


10cm

重複無限次
如果
RGB超聲波讀取距離(cm)
>
10
那麼
馬達 M1 速度 150 M2 速度 150
馬達 M3 速度 150 M4 速度 150
直 行
否則如果
RGB超聲波讀取距離(cm)
≤
10
那麼
停止所有馬達
馬達 M1 速度 100 M2 速度 -100
馬達 M3 速度 100 M4 速度 -100
向 右 旋 轉 180 度
暫停 2000 毫秒

練習 7:


(1)
(2)
(3)
(4)
(5)
(6)
(7)
(8)
(9)


重複無限次
如果 RGB超聲波讀取距離(cm) > 12 那麼
馬達 M1 速度 60 M2 速度 60
馬達 M3 速度 60 M4 速度 60
(1)
否則如果 RGB超聲波讀取距離(cm) ≤ 12 那麼
停止所有馬達
馬達 M1 速度 100 M2 速度 -100
馬達 M3 速度 100 M4 速度 -100
暫停 600 毫秒
(2)
馬達 M1 速度 60 M2 速度 60
馬達 M3 速度 60 M4 速度 60
暫停 1000 毫秒
(3)
馬達 M1 速度 -100 M2 速度 100
馬達 M3 速度 -100 M4 速度 100
暫停 600 毫秒
(4)
馬達 M1 速度 60 M2 速度 60
馬達 M3 速度 60 M4 速度 60
暫停 1300 毫秒
(5)
馬達 M1 速度 -100 M2 速度 100
馬達 M3 速度 -100 M4 速度 100
(6)
暫停 600 毫秒
馬達 M1 速度 60 M2 速度 60
馬達 M3 速度 60 M4 速度 60
(7)
暫停 1000 毫秒
馬達 M1 速度 100 M2 速度 -100
馬達 M3 速度 100 M4 速度 -100
(8)
暫停 600 毫秒
馬達 M1 速度 60 M2 速度 60
馬達 M3 速度 60 M4 速度 60
(9)

練習 8:

當啟動時
變數 detecteddistance 設為 12
設定感應障礙物的距離為12
重複無限次
如果 RGB超聲波讀取距離(cm) < detecteddistance 那麼
停止所有馬達
設置超聲波燈 全部 顏色 紅
遇到障礙物的指示燈
馬達 M1 速度 150 M2 速度 -150
馬達 M3 速度 0 M4 速度 0
車子向右轉
暫停 1000 毫秒
停止所有馬達
暫停 500 毫秒
如果 RGB超聲波讀取距離(cm) < detecteddistance 那麼
停止所有馬達
設置超聲波燈 全部 顏色 紅
遇到障礙物的指示燈
馬達 M1 速度 -150 M2 速度 150
馬達 M3 速度 0 M4 速度 0
車子向左轉
暫停 2000 毫秒
停止所有馬達
暫停 500 毫秒

如果 RGB超聲波讀取距離(cm) < detecteddistance
停止所有馬達
如果車子再遇上障礙物，車子停止
否則
設置超聲波燈 全部 顏色 綠
沒有遇到障礙物的指示燈
馬達 M1 速度 50 M2 速度 50
馬達 M3 速度 50 M4 速度 50
車子繼續直行

附Ⓕ:

數學
Stemhub:bit
Neopixel
進階
函式
陣列
文字
遊戲
圖像
引腳
更多
引腳
數位信號讀取 引腳 P0
數位信號寫入 引腳 P0 數字 0
類比信號讀取 引腳 P0
類比信號寫入 引腳 P0 數字 1023
對應 0
從低 0
到高 1023
至低 0
到高 4

數位信號寫入 引腳 P0 數字 0

數學
Stemhub:bit
Neopixel
進階
函式
陣列
文字
遊戲
圖像
引腳
更多
更多
當 引腳 P0 脈衝為 高
脈衝長度 (μs)
引腳 P0 脈波 高 持續時間 (微秒)
i2c 讀取數字 位置 0 格式 Int8LE 重複 false
i2c 寫入數字
位置 0
數值 0
格式 Int8LE

引腳 P0 脈波 高 持續時間 (微秒)

超聲波傳感器測距編程

定義函式 超聲波
數位信號寫入 引腳 P15 數字 0
等待 2 微秒
數位信號寫入 引腳 P15 數字 1
等待 10 微秒
數位信號寫入 引腳 P15 數字 0
變數 距離 設為 引腳 P14 脈波 高 持續時間（微秒） ÷ 58
顯示 數字 距離

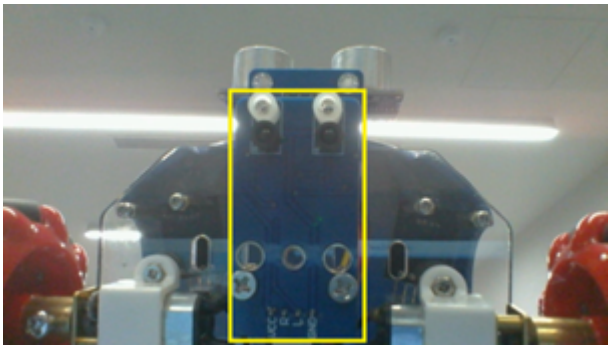
1.3.6 第七至八課

簡介

教學目標

巡Ⓕ行駛

- 紅外巡Ⓕ傳感器

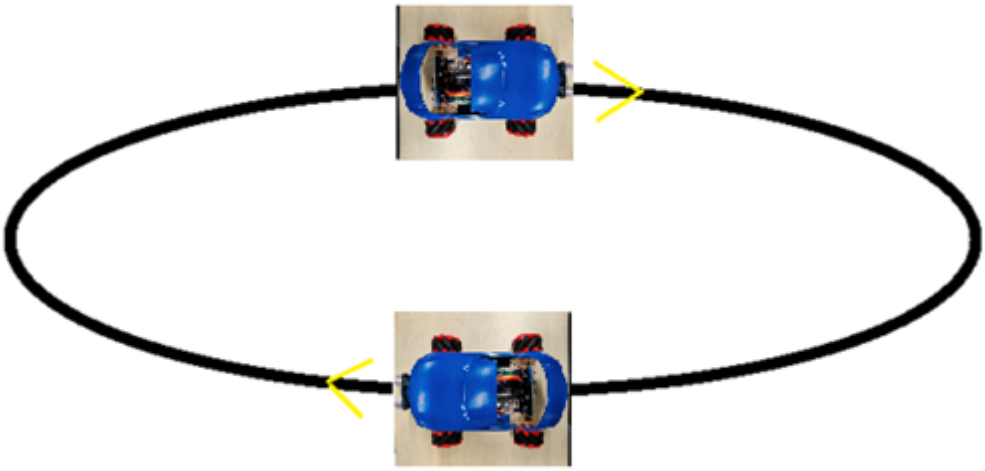


- 巡Ⓕ基本模組

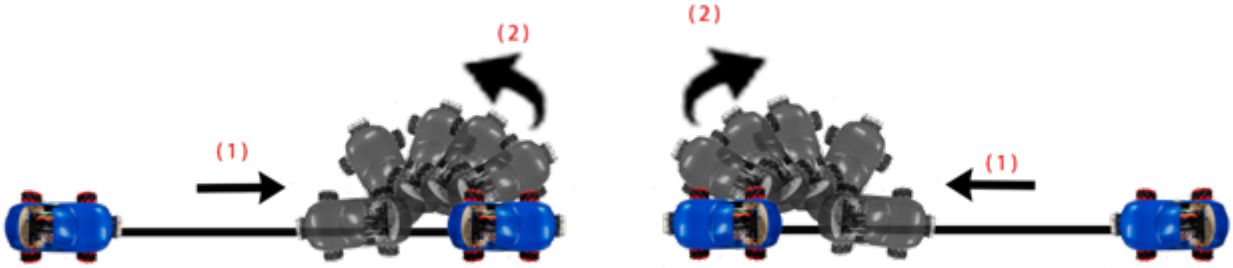
基本
輸入
音效
燈光
廣播
迴圈
邏輯
變數
數學
Stemhub:bit
Neopixel
進階
函式
陣列
文字
遊戲
圖像
引腳
更多
序列
控制
引腳
數位信號讀取 引腳 P0
數位信號寫入 引腳 P0 數字
類比信號讀取 引腳 P0
類比信號寫入 引腳 P0 數字
對應 0

變數 L 設為 數位信號讀取 引腳 P13
變數 R 設為 數位信號讀取 引腳 P14

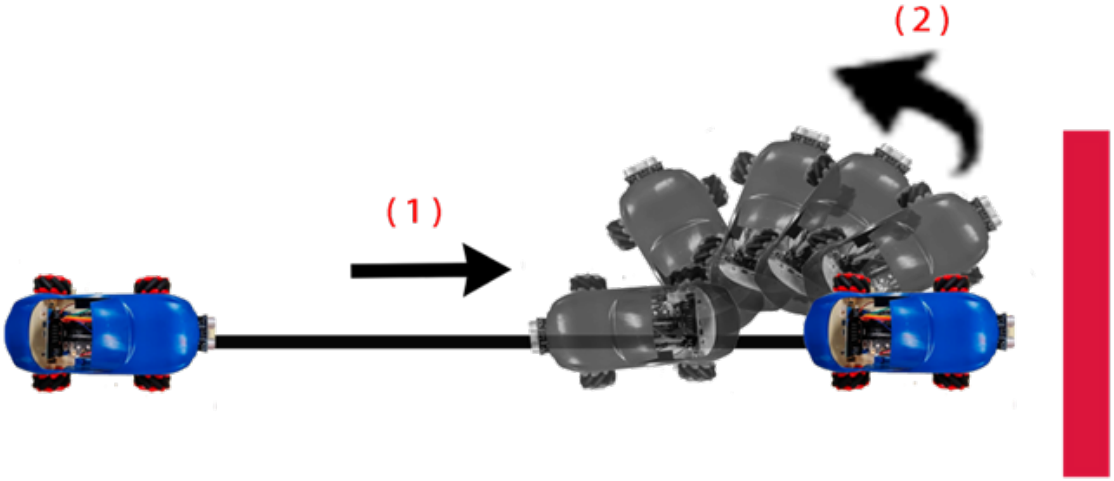
練習一: 設計程序令小車巡著黑色Ⓕ行駛



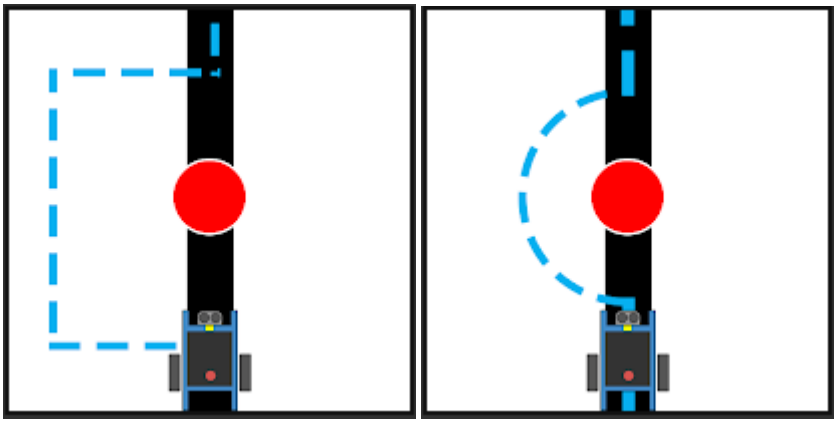
練習二:



練習三:



練習四: **迷Ⓕ** **



挑戰

- 建立變數

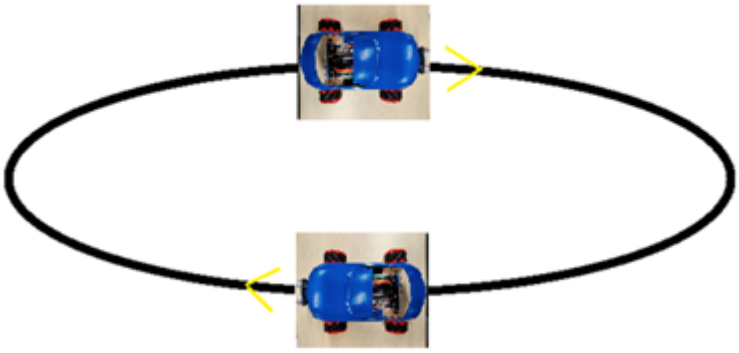
，想想如何在同時間儲存左右邊感應器的數位信號

- 建立變數

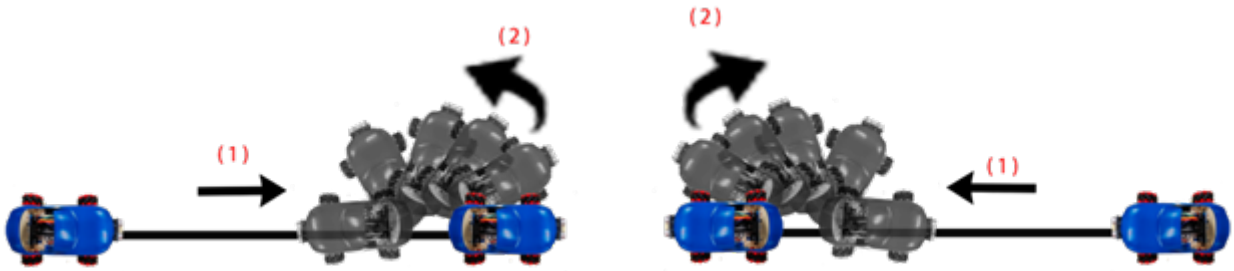
儲存出界時的信號，ⒻⒻ定出界後的轉彎方向

答案

練習一:

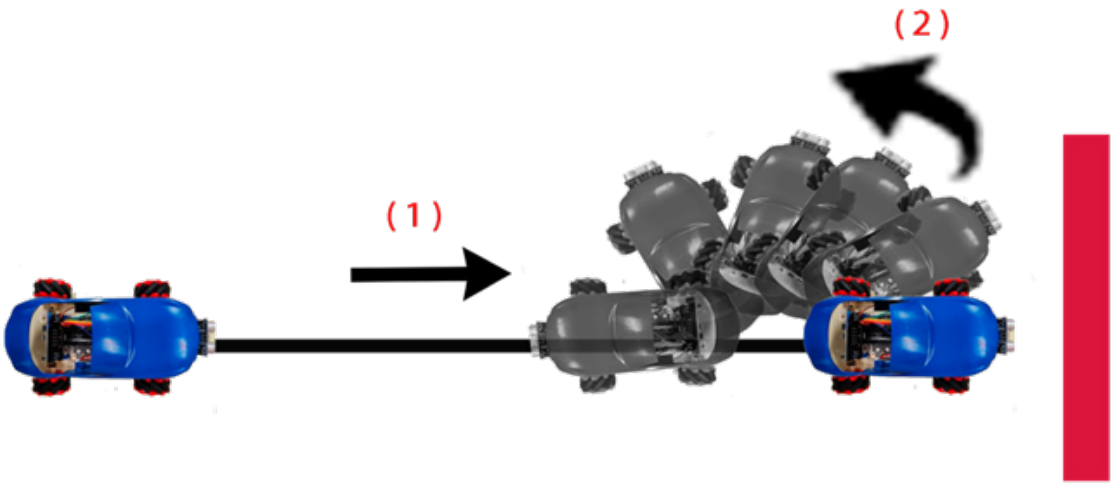


練習二:


(2)
(2)
(1)
(1)

重複無限次
變數 L 設為 數位信號讀取 引腳 P13
變數 R 設為 數位信號讀取 引腳 P14
設定R和L的數位型號為
P13 和 P14
如果 L = 0 且 R = 0 那麼
馬達 M1 速度 60 M2 速度 60
馬達 M3 速度 60 M4 速度 60
車子直行
否則如果 L = 1 且 R = 0 那麼
馬達 M1 速度 80 M2 速度 0
馬達 M3 速度 80 M4 速度 0
車子向右轉
否則如果 L = 0 且 R = 1 那麼
馬達 M1 速度 0 M2 速度 80
馬達 M3 速度 0 M4 速度 80
車子向左轉
否則如果 L = 1 且 R = 1 那麼
馬達 M1 速度 60 M2 速度 -40
馬達 M3 速度 60 M4 速度 -40
如果車子出界，
車子掉頭

練習三:


(2)
(1)

設定R和L的數位型號為
P13 和 P14
遇到障礙物的指示燈
距離為12
遇到障礙物，轉右
L：黑色
R：黑色
車子直行
L：白色
R：黑色
車子向右轉
L：黑色
R：白色
車子向左轉

L：白色
R：白色
車子向右旋轉180度

練習四:

距離為12
重複sensor函式
如果遇到障礙物，
呼叫ultrasonic函式
沒有遇到障礙物的指示燈
設定L和R的數位信號
為P12和P13
L：黑色
R：黑色
車子直行
L：白色
R：黑色
車子向右轉
L：白色
R：白色
車子向右旋轉180度
L：黑色
R：白色
車子向左轉

挑戰

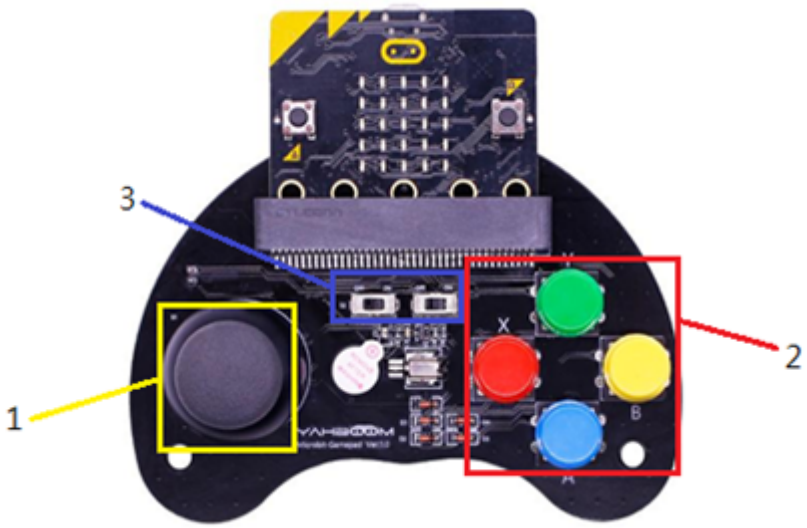
重複無限次
呼叫 line_follow
定義函式 line_follow
變數 BW_value 設為 數位信號讀取 引腳 P13 × 10
變數 BW_value 改變 數位信號讀取 引腳 P14
先儲存左邊讀取的數位信號，會得出00或10
把右邊讀取的數位信號加上，
如果是0（黑色）可得出00,10，是1（白色）則可得
如果 BW_value ≠ 11 那麼
馬達 M1 速度 80 M2 速度 80
馬達 M3 速度 80 M4 速度 80
變數 last_value 設為 BW_value
當車子沒有完全出界，車子前進
並儲存最後的數位信號
否則
如果 last_value = 1 那麼
如果車子完全出界，以最後的數位信號（1或10）決定轉彎方向
馬達 M1 速度 -40 M2 速度 80
馬達 M3 速度 -40 M4 速度 80
01則黑線在左，車子左轉
否則
馬達 M1 速度 80 M2 速度 -40
馬達 M3 速度 80 M4 速度 -40
10則黑線在右，車子右轉
暫停 10 毫秒
車子不斷短暫暫停以避免走太快出界

1.3.7 第九至十課

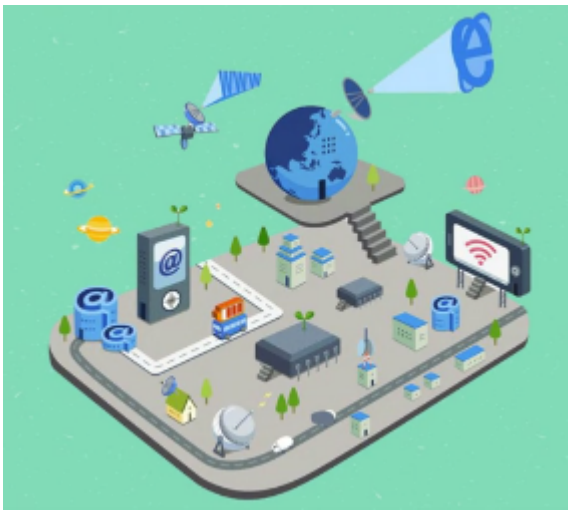
簡介

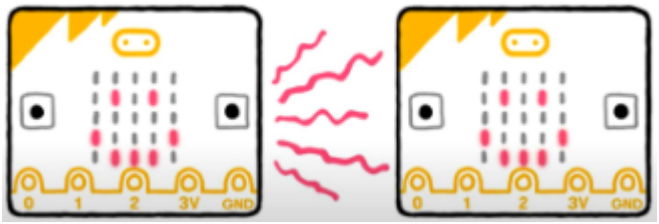
教學目標

Ⓕ控器



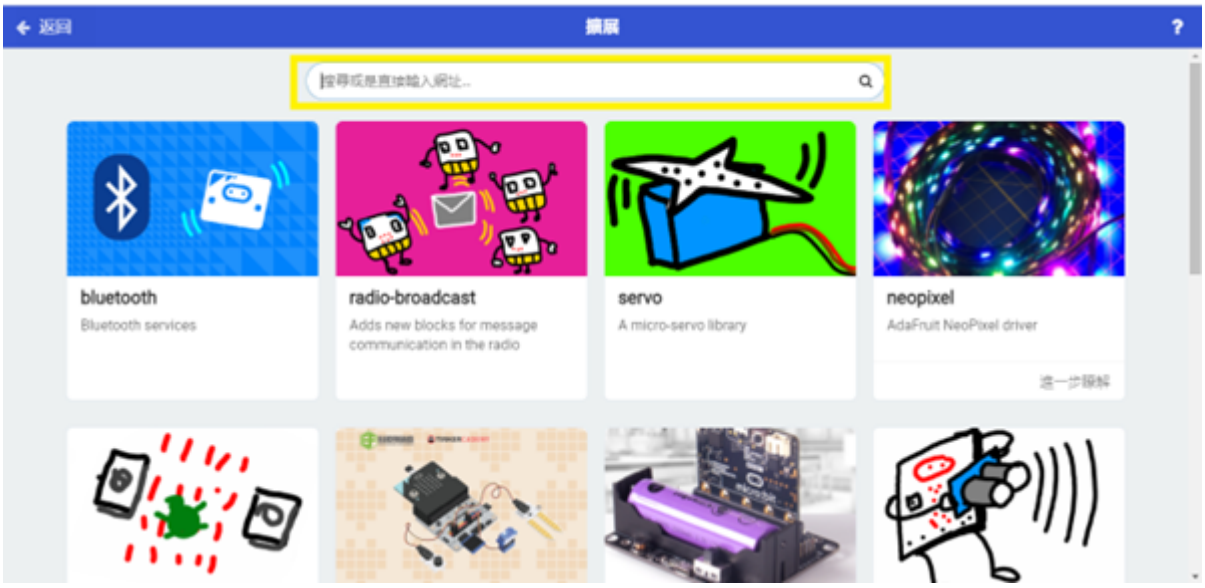
無線電與我們的日常生活

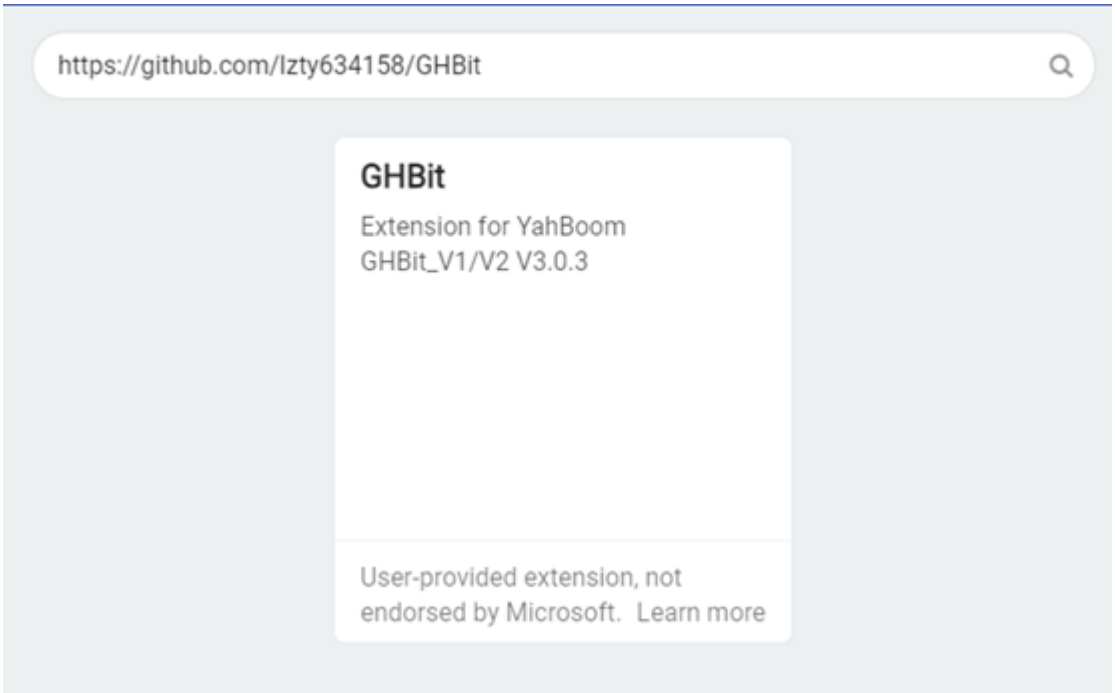




無線電的程式模組

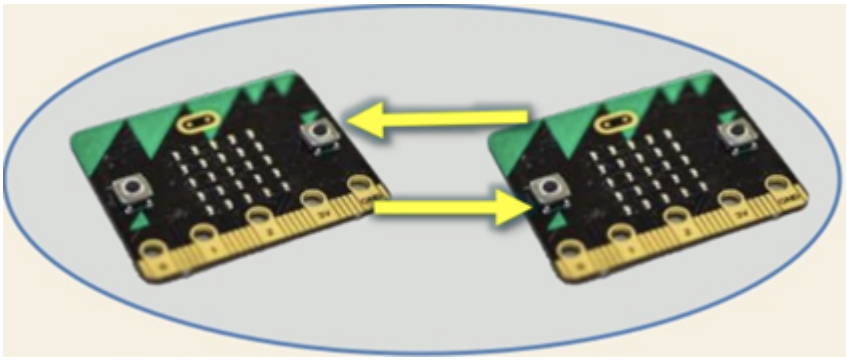
下載手制擴展包:





設立無Ⓕ電的積木: (在廣播分類中)

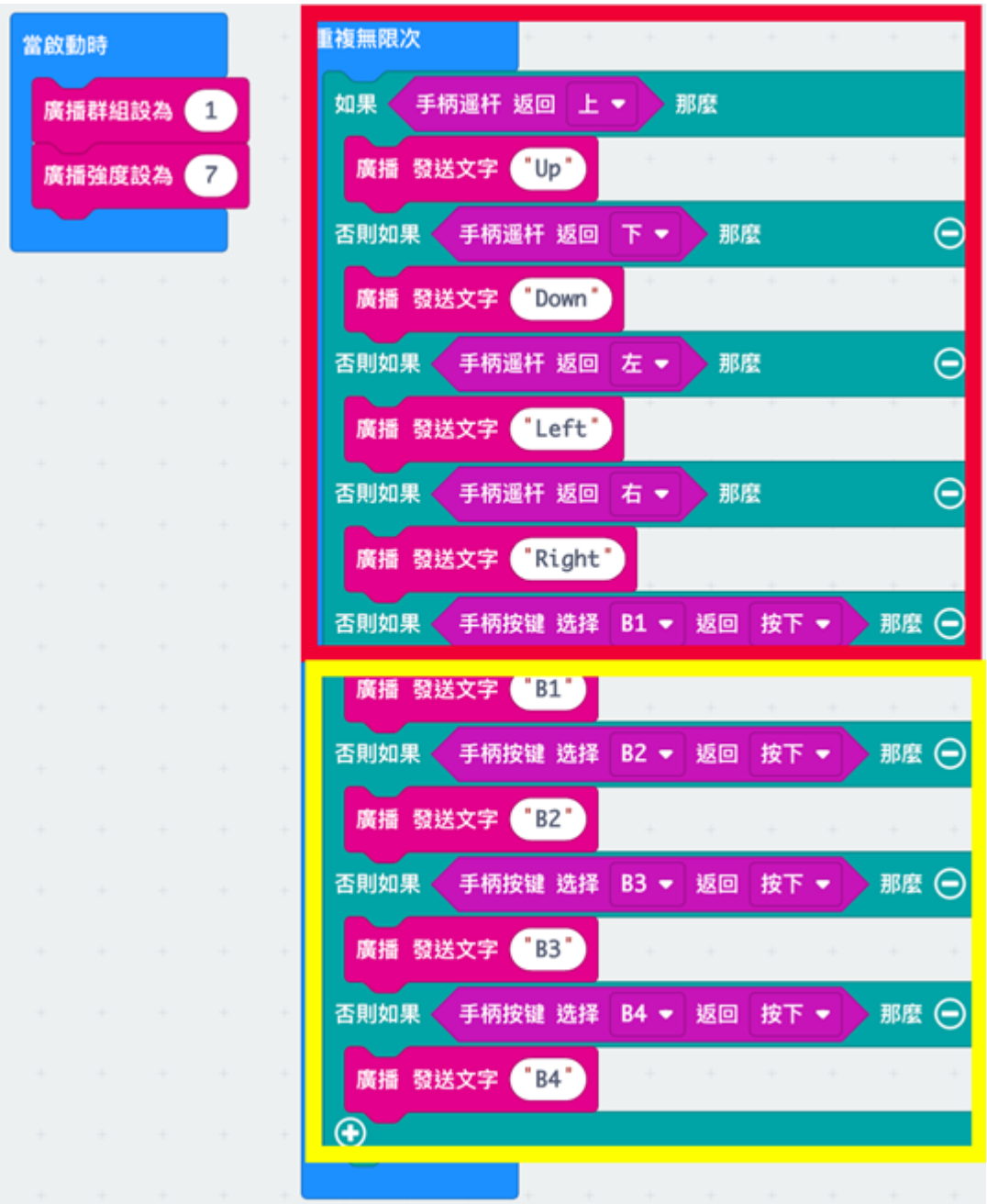
范例 1: 雙向無Ⓕ電



Ⓕ控器和小車的程式模組

手制的積木:

手制的編程:(GHBit)


當啟動時
廣播群組設為 1
廣播強度設為 7
重複無限次
如果 手柄遥杆 返回 上 那麼
廣播 發送文字 "Up"
否則如果 手柄遥杆 返回 下 那麼
廣播 發送文字 "Down"
否則如果 手柄遥杆 返回 左 那麼
廣播 發送文字 "Left"
否則如果 手柄遥杆 返回 右 那麼
廣播 發送文字 "Right"
否則如果 手柄按键 选择 B1 返回 按下 那麼
廣播 發送文字 "B1"
否則如果 手柄按键 选择 B2 返回 按下 那麼
廣播 發送文字 "B2"
否則如果 手柄按键 选择 B3 返回 按下 那麼
廣播 發送文字 "B3"
否則如果 手柄按键 选择 B4 返回 按下 那麼
廣播 發送文字 "B4"

小車的編程:

```
B1: 暫停行駛
B2: 顯示心型圖案
B3: 車子向左移
B4: 車子向右移
```

手制
當收到廣播文字 receivedString
如果 receivedString = "Up" 那麼
馬達 M1 速度 255 M2 速度 255
馬達 M3 速度 255 M4 速度 255
否則如果 receivedString = "Down" 那麼
馬達 M1 速度 -255 M2 速度 -255
馬達 M3 速度 -255 M4 速度 -255
否則如果 receivedString = "Left" 那麼
馬達 M1 速度 -85 M2 速度 255
馬達 M3 速度 85 M4 速度 255
否則如果 receivedString = "Right" 那麼
馬達 M1 速度 255 M2 速度 -85
馬達 M3 速度 255 M4 速度 85
當啟動時
廣播群組設為 1
廣播強度設為 7
否則如果 receivedString = "B1" 那麼
停止所有馬達
車子暫停行駛
否則如果 receivedString = "B2" 那麼
顯示 圖示
否則如果 receivedString = "B3" 那麼
馬達 M1 速度 -150 M2 速度 150
馬達 M3 速度 150 M4 速度 -150
車子左移
否則如果 receivedString = "B4" 那麼
馬達 M1 速度 150 M2 速度 -150
馬達 M3 速度 -150 M4 速度 150
車子右移
按鈕

練習 1:

練習 2:

答案

練習 1:

練習 2:

當啟動時
廣播群組設為 1
廣播強度設為 7
重複無限次
如果 手柄遥杆 返回 上 那麼
廣播 發送文字 "Up"
否則如果 手柄遥杆 返回 下 那麼
廣播 發送文字 "Down"
否則如果 手柄遥杆 返回 左 那麼
廣播 發送文字 "Left"
否則如果 手柄遥杆 返回 右 那麼
廣播 發送文字 "Right"
否則如果 手柄按键 选择 B1 返回 按下 那麼
廣播 發送文字 "B1"
否則如果 手柄按键 选择 B2 返回 按下 那麼
廣播 發送文字 "B2"
否則如果 手柄按键 选择 B3 返回 按下 那麼
廣播 發送文字 "B3"
否則如果 手柄按键 选择 B4 返回 按下 那麼
廣播 發送文字 "B4"
否則
廣播 發送文字 "Stop"

手制
當啟動時
廣播群組設為 1
廣播強度設為 7
定義函式 RGB_Off
設置超聲波燈 全部 顏色 黑
設置板載燈 全部燈 顏色 黑
定義函式 RGB_ON
設置超聲波燈 全部 顏色 藍
設置板載燈 第一個燈 顏色 紅
設置板載燈 第二個燈 顏色 綠
設置板載燈 第三個燈 顏色 黃
設置板載燈 第四個燈 顏色 紫
按鈕
當收到廣播文字 receivedString
如果 receivedString = "Up" 那麼
馬達 M1 速度 255 M2 速度 255
馬達 M3 速度 255 M4 速度 255
否則如果 receivedString = "Down" 那麼
馬達 M1 速度 -255 M2 速度 -255
馬達 M3 速度 -255 M4 速度 -255
否則如果 receivedString = "Left" 那麼
馬達 M1 速度 -85 M2 速度 255
馬達 M3 速度 85 M4 速度 255
否則如果 receivedString = "Right" 那麼
馬達 M1 速度 255 M2 速度 -85
馬達 M3 速度 255 M4 速度 85
否則如果 receivedString = "B1" 那麼
呼叫 RGB_ON
否則如果 receivedString = "B2" 那麼
舵機 S1 角度 0
否則如果 receivedString = "B3" 那麼
舵機 S1 角度 110
否則如果 receivedString = "B4" 那麼
馬達 M1 速度 150 M2 速度 -150
馬達 M3 速度 -150 M4 速度 150
否則如果 receivedString = "Stop" 那麼
停止所有馬達
呼叫 RGB_Off

1.4 Micro:bit M1 智能小車 _ 中級

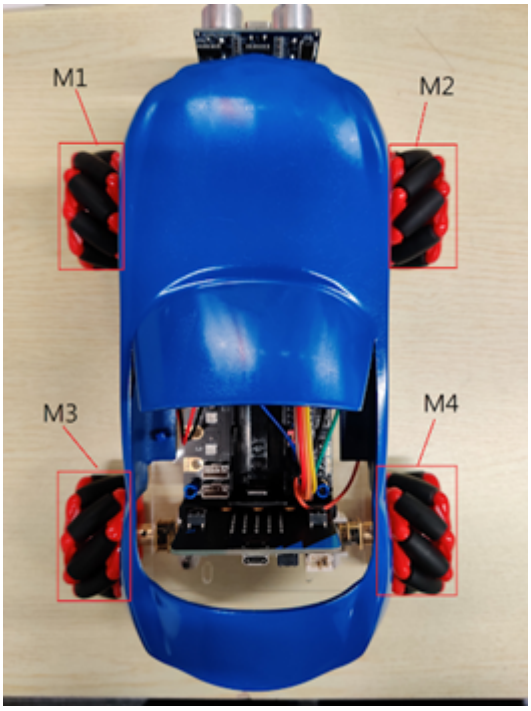
1.4.1 第一課

簡介

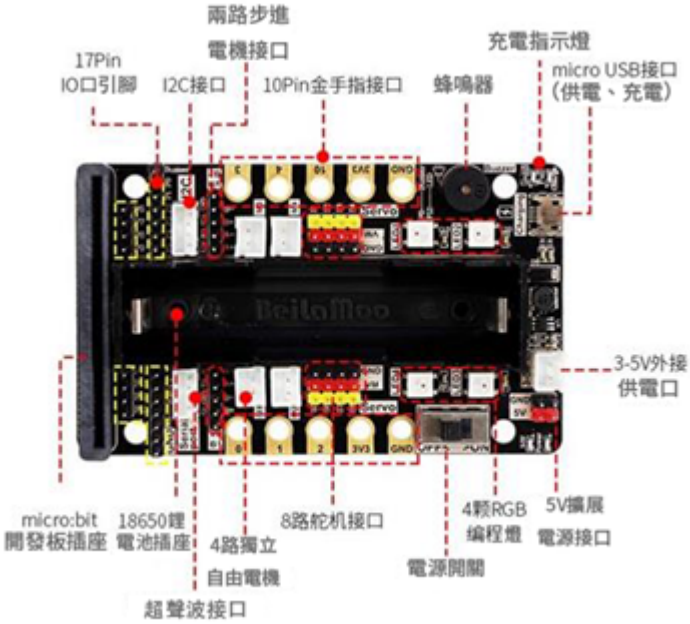
教學目標

重Ⓕ麥克納姆輪車子的結構

- 4 個馬達 (M1, M2, M3, M4)
- 4 個麥克納姆輪子
- Micro bit
- Micro bit 擴展板

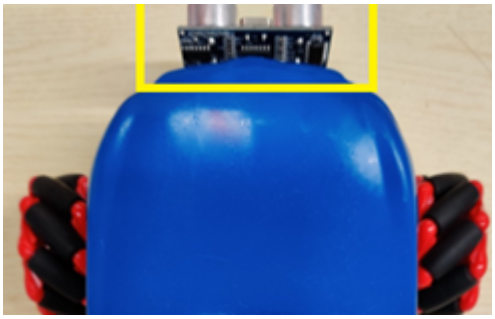


micro:bit 擴展板

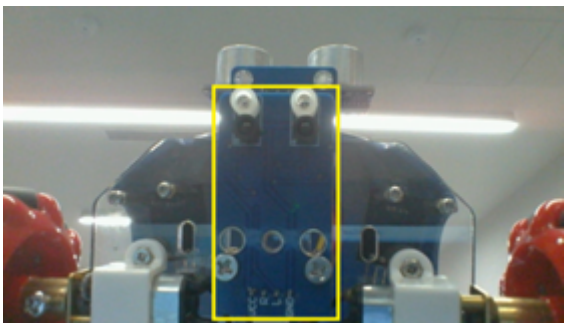


傳感器

- 超聲波傳感器



- 紅外巡Ⓕ傳感器



重Ⓕ進階積木模塊

函式

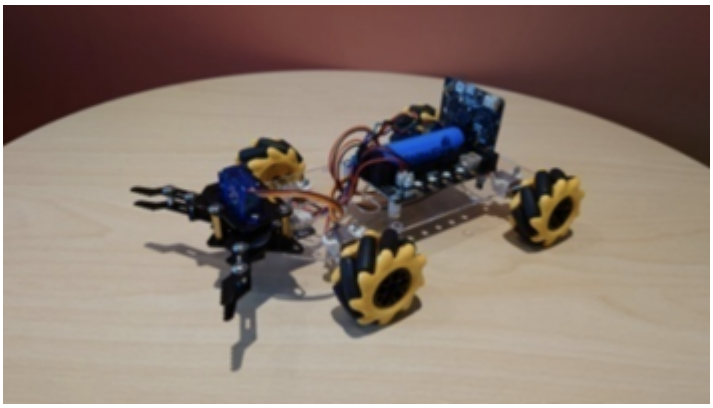
建立函式

在F框F填入函式名稱

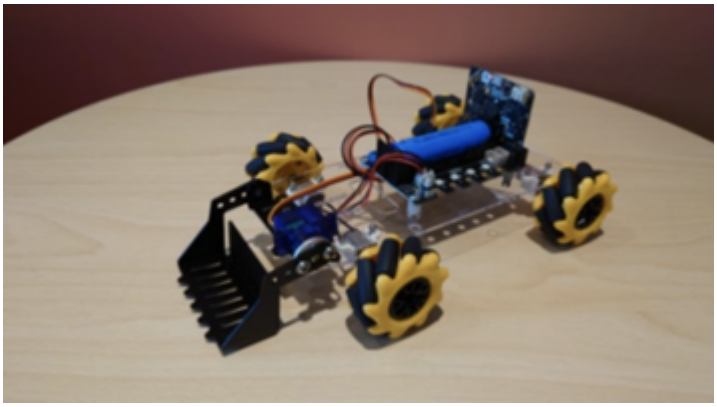
呼叫函式以運行函式中的程序

認識機械擴展工具

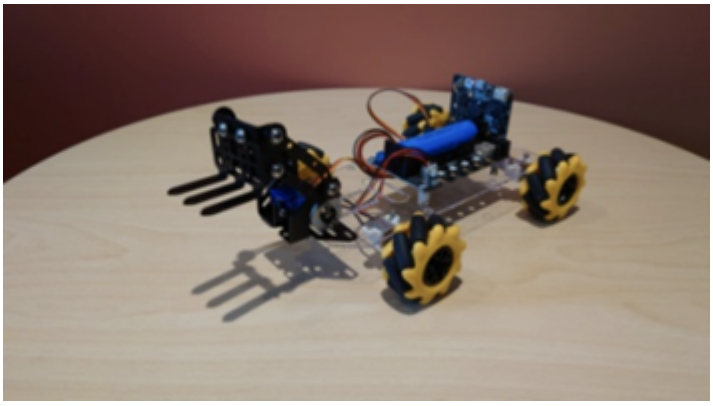
夾子 (Beetle)



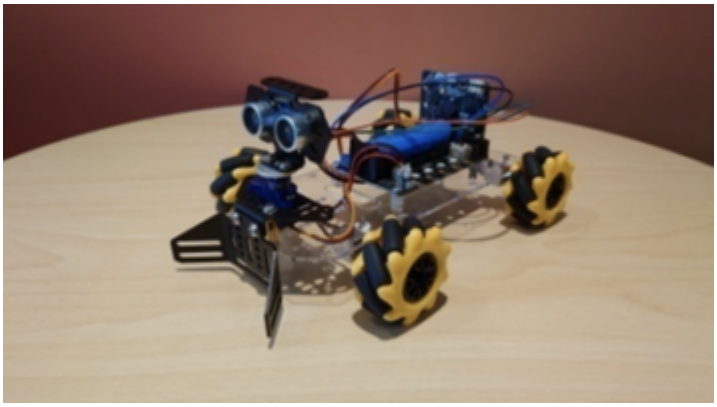
鏟子 (Loader)



叉子 (Forklift)

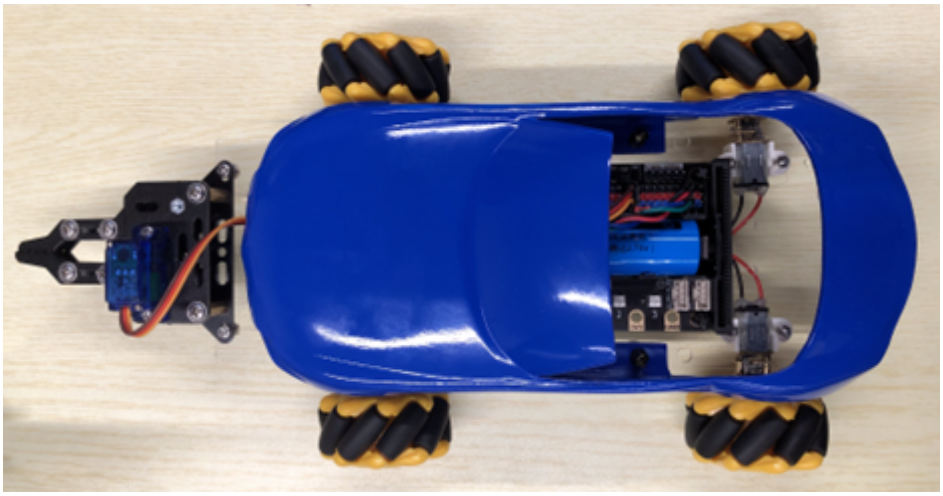


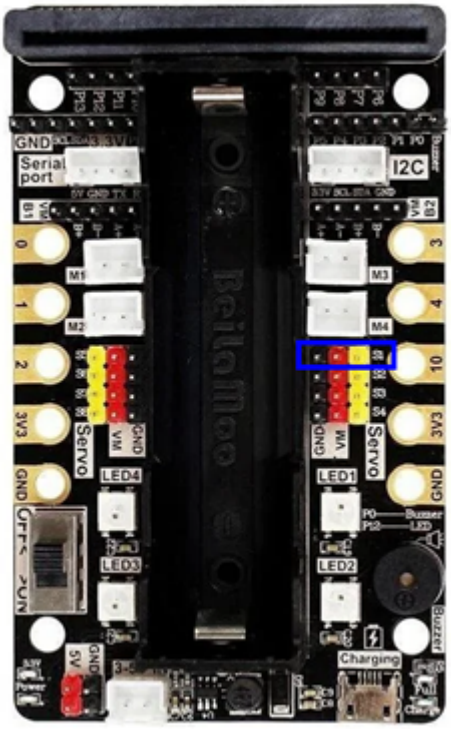
推土機配超聲波傳感器



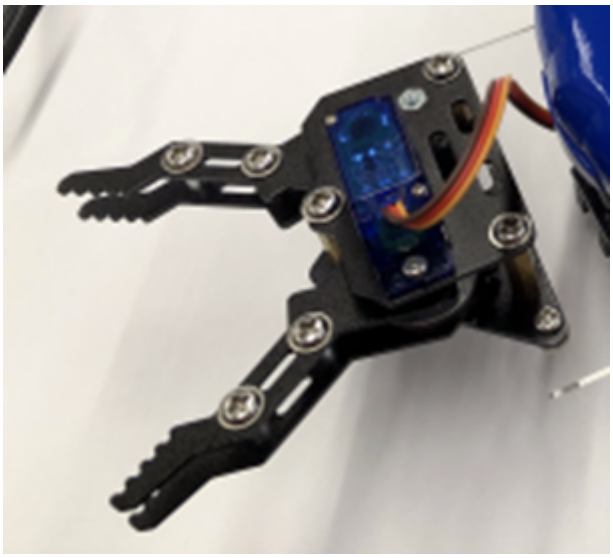
擴展工具包中所有的零件：

認識 Micro:bit 擴展工具——夾子

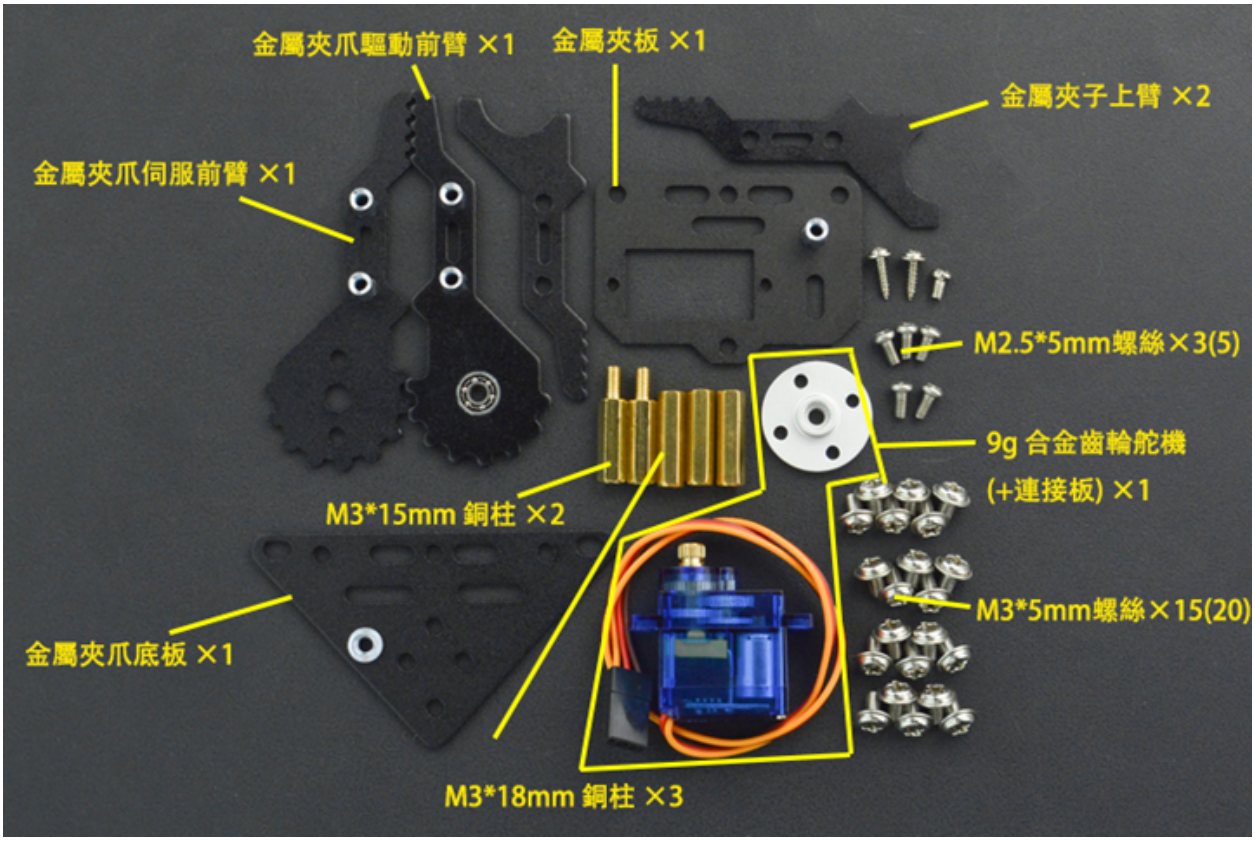


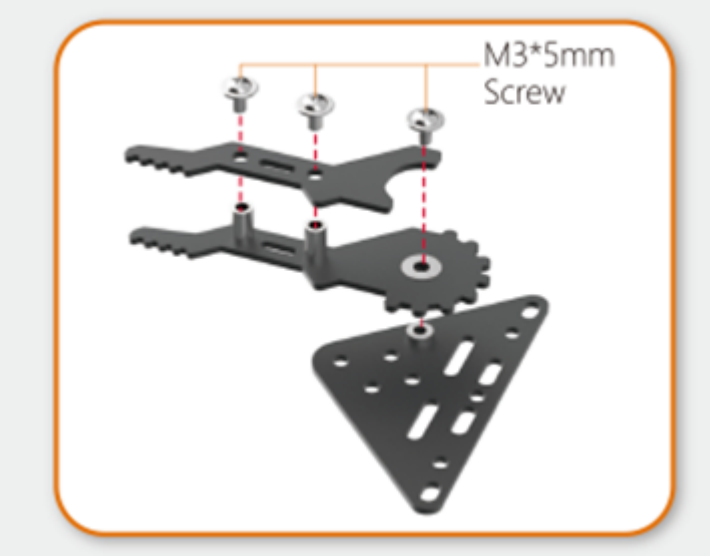


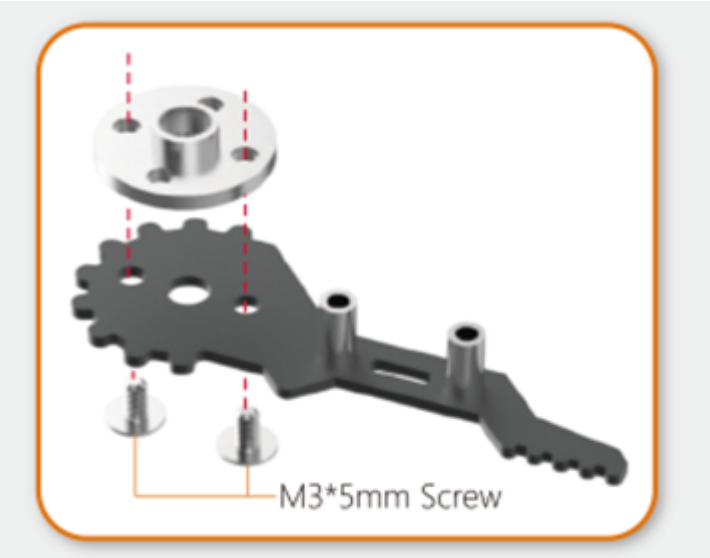
機械夾子的原理及功用

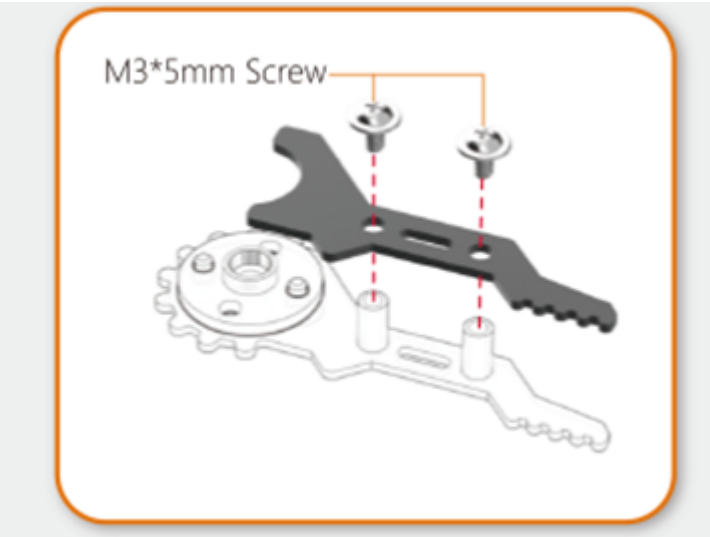


安裝夾子:

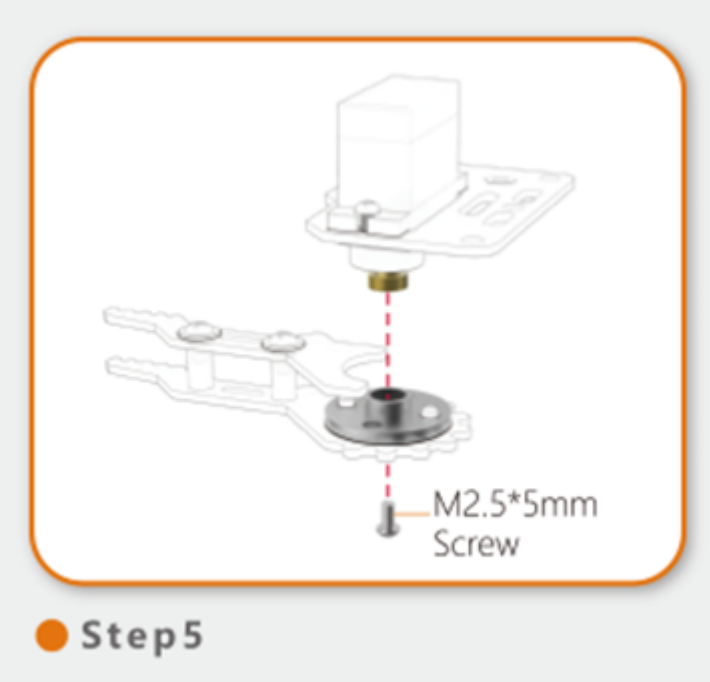

金屬夾爪驅動前臂 ×1
金屬夾板 ×1
金屬夾子上臂 ×2
金屬夾爪伺服前臂 ×1
M2.5*5mm螺絲×3(5)
9g 合金齒輪舵機
(+連接板) ×1
M3*15mm 銅柱 ×2
M3*5mm螺絲×15(20)
金屬夾爪底板 ×1
M3*18mm 銅柱 ×3

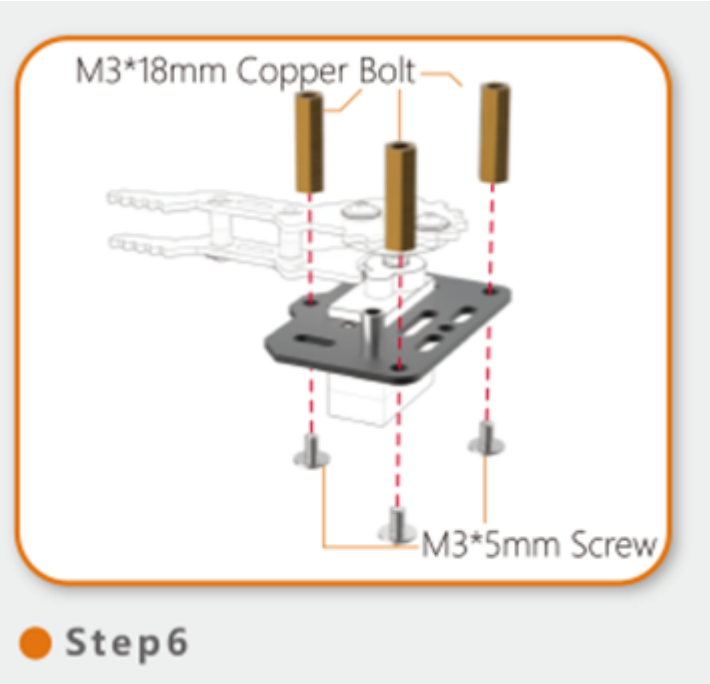

M3*5mm
Screw

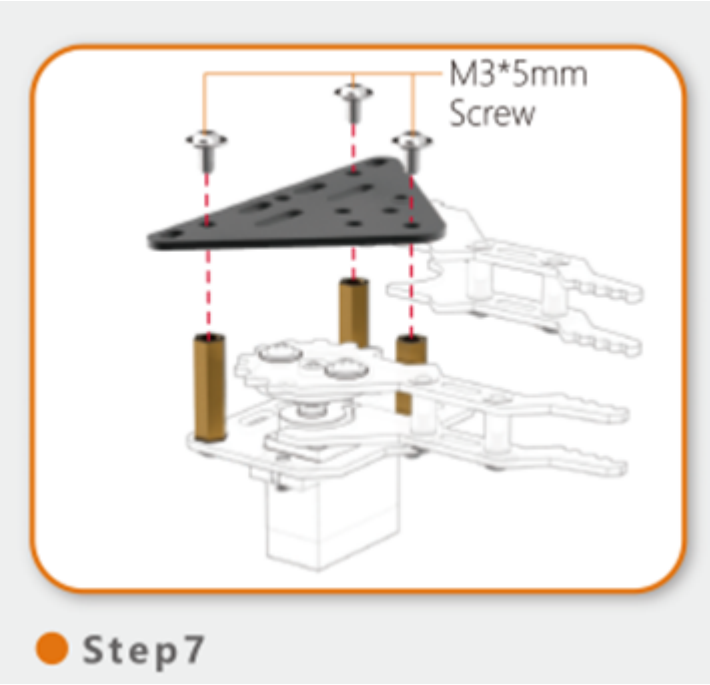

M3*5mm Screw

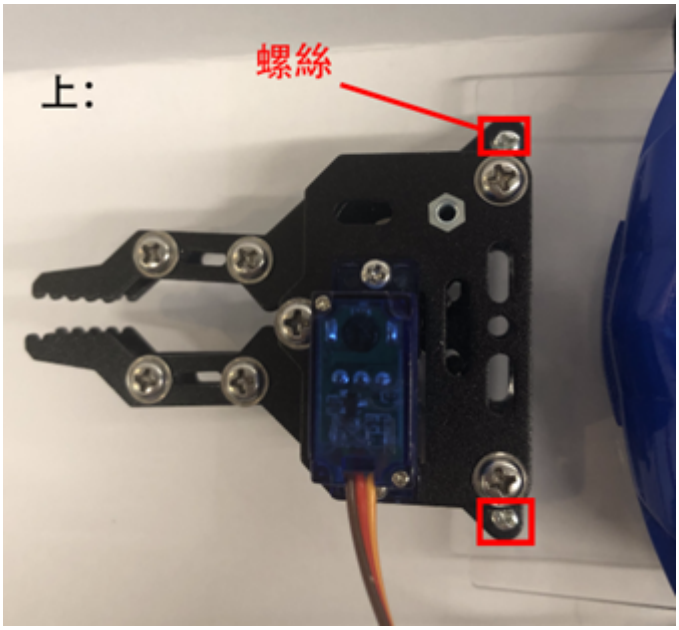

M3*5mm Screw

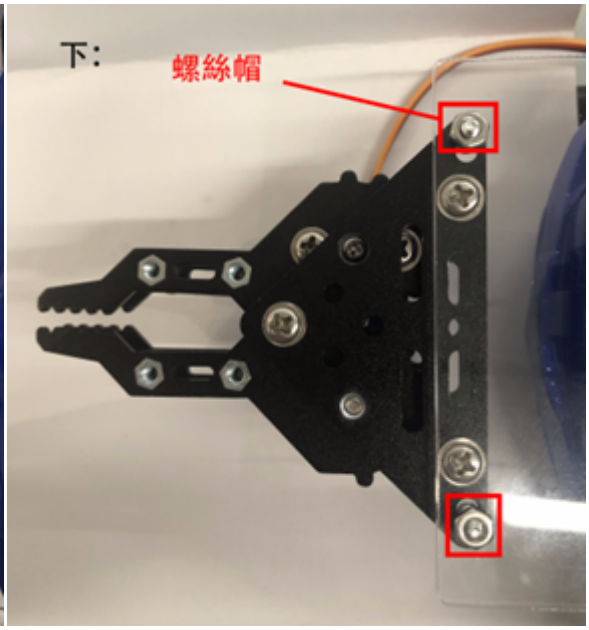
M2.5*5mm
Screw
Step4


M2.5*5mm
Screw
Step5


M3*18mm Copper Bolt
M3*5mm Screw
Step6

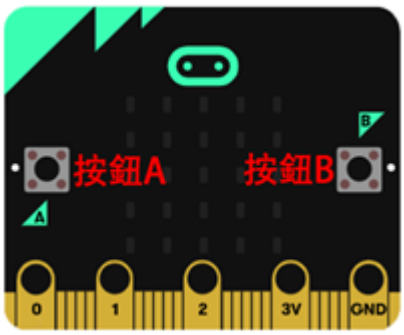

M3*5mm
Screw
Step7

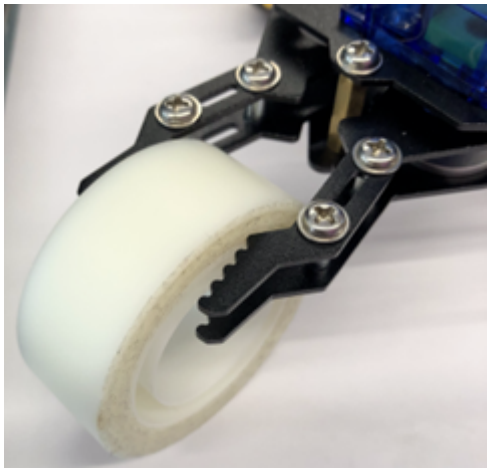




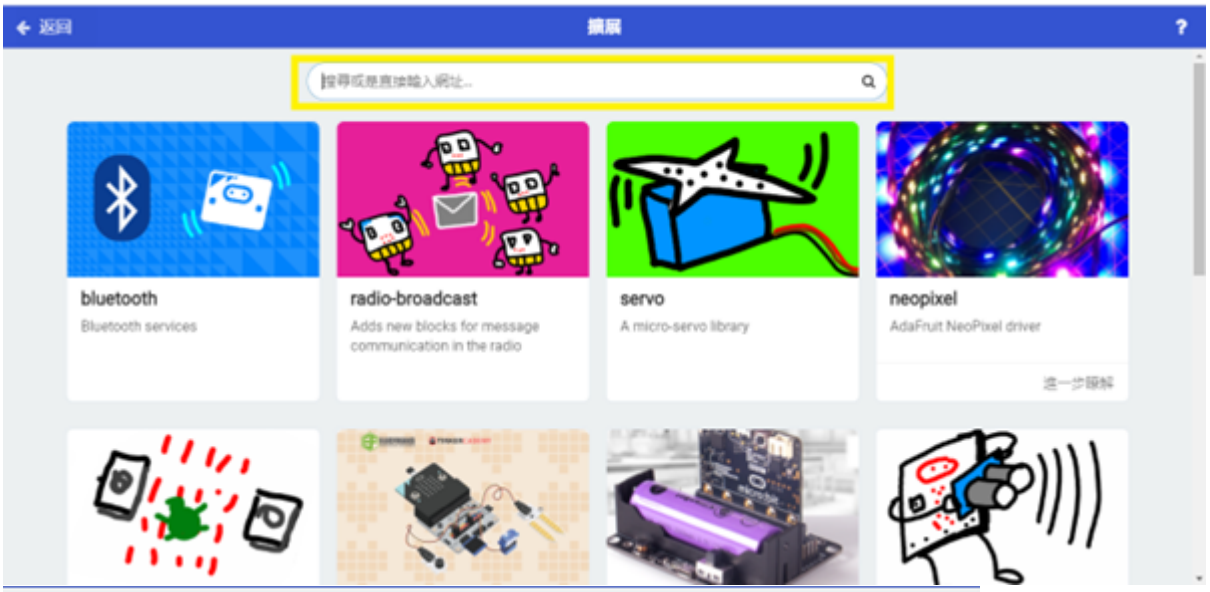
Stemhub:bit 舵機積木模塊

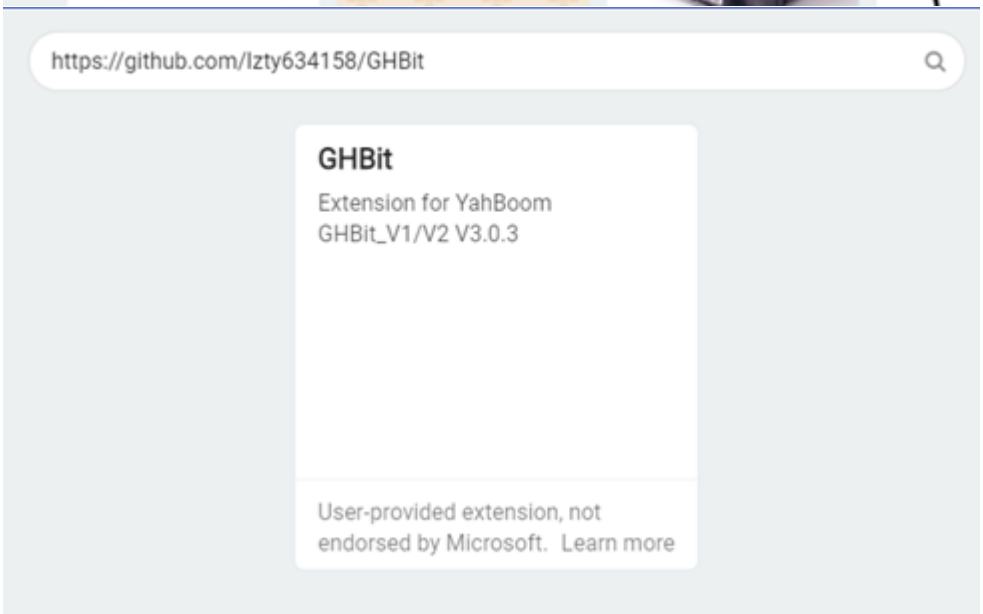
練習 1

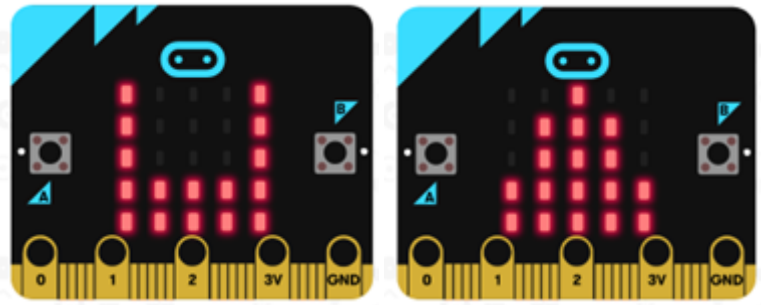




練習 2


返回
擴展
bluetooth
Bluetooth services
radio-broadcast
Adds new blocks for message communication in the radio
servo
A micro-servo library
neopixel
AdaFruit NeoPixel driver

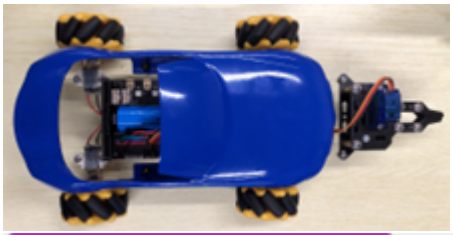

https://github.com/lzty634158/GHBit
GHBit
Extension for YahBoom
GHBit_V1/V2 V3.0.3
User-provided extension, not endorsed by Microsoft. Learn more

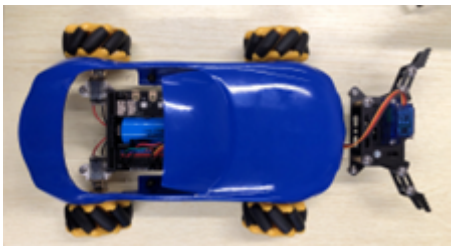


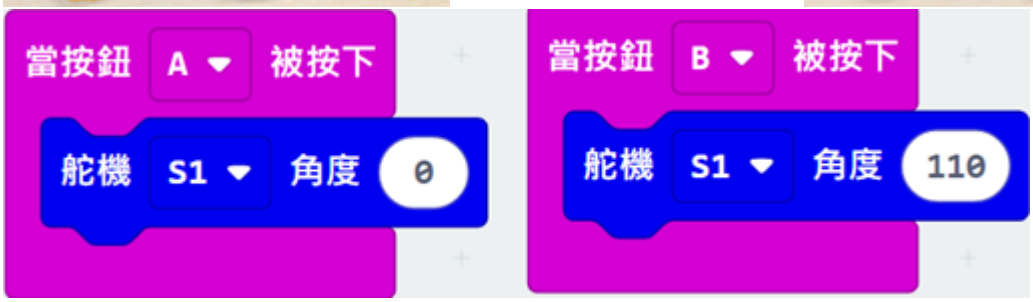
- 參考初階第十科，以接收文字作爲行動條件
- 運用基本 » 顯示指示燈積木模塊

答案

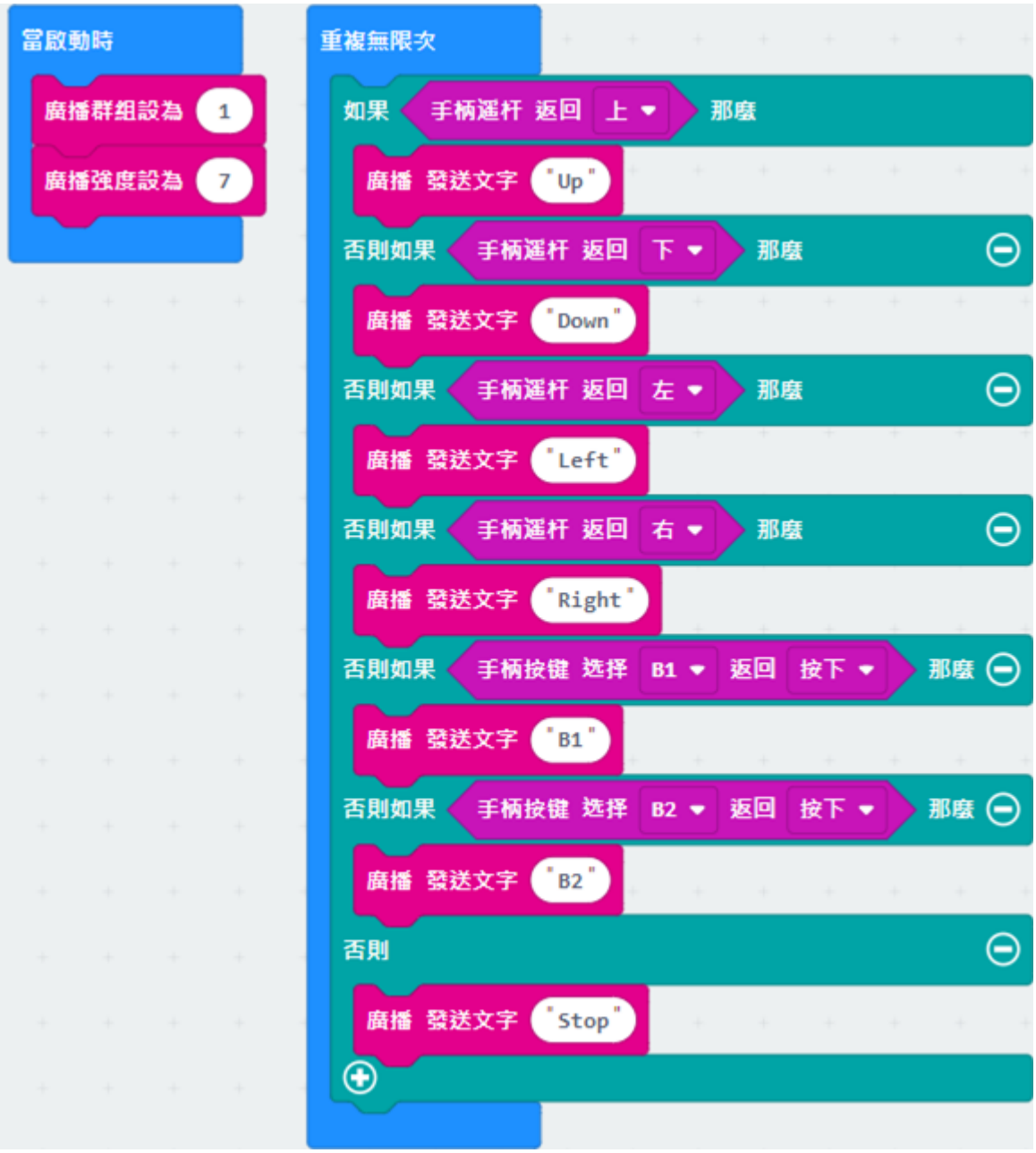
練習 1







練習 2


當啟動時
廣播群組設為 1
廣播強度設為 7
重複無限次
如果 手柄遥杆 返回 上 那麼
廣播 發送文字 "Up"
否則如果 手柄遥杆 返回 下 那麼
廣播 發送文字 "Down"
否則如果 手柄遥杆 返回 左 那麼
廣播 發送文字 "Left"
否則如果 手柄遥杆 返回 右 那麼
廣播 發送文字 "Right"
否則如果 手柄按键 选择 B1 返回 按下 那麼
廣播 發送文字 "B1"
否則如果 手柄按键 选择 B2 返回 按下 那麼
廣播 發送文字 "B2"
否則
廣播 發送文字 "Stop"

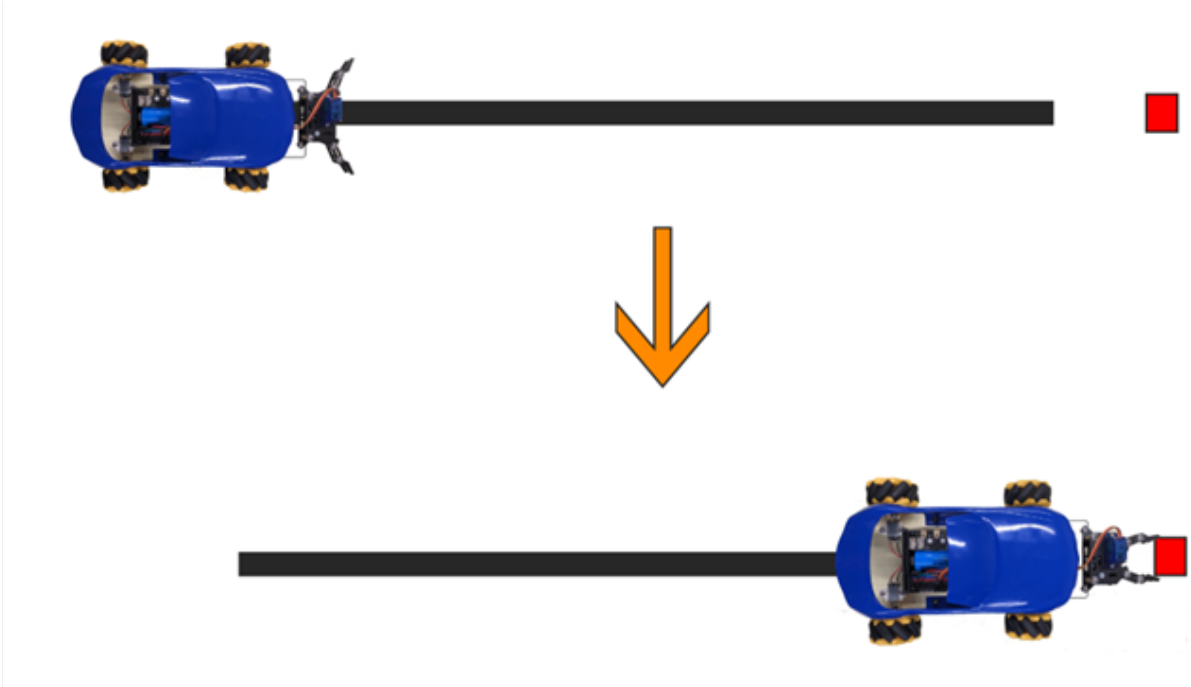
當啟動時
廣播群組設為 1
夾子合上並顯示合上符號
定義函式 Close
舵機 S1 角度 110
顯示 指示燈
夾子打開並顯示打開符號
定義函式 Open
舵機 S1 角度 0
顯示 指示燈
當收到廣播文字 receivedString
如果 receivedString = "Up" 那麼
馬達 M1 速度 255 M2 速度 255
馬達 M3 速度 255 M4 速度 255
否則如果 receivedString = "Down" 那麼
馬達 M1 速度 -255 M2 速度 -255
馬達 M3 速度 -255 M4 速度 -255
否則如果 receivedString = "Left" 那麼
馬達 M1 速度 -85 M2 速度 255
馬達 M3 速度 85 M4 速度 255
否則如果 receivedString = "Right" 那麼
馬達 M1 速度 255 M2 速度 -85
馬達 M3 速度 255 M4 速度 85
否則如果 receivedString = "B1" 那麼
呼叫 Open
否則如果 receivedString = "B2" 那麼
呼叫 Close
否則如果 receivedString = "Stop" 那麼
停止所有馬達

1.4.2 第二課

簡介

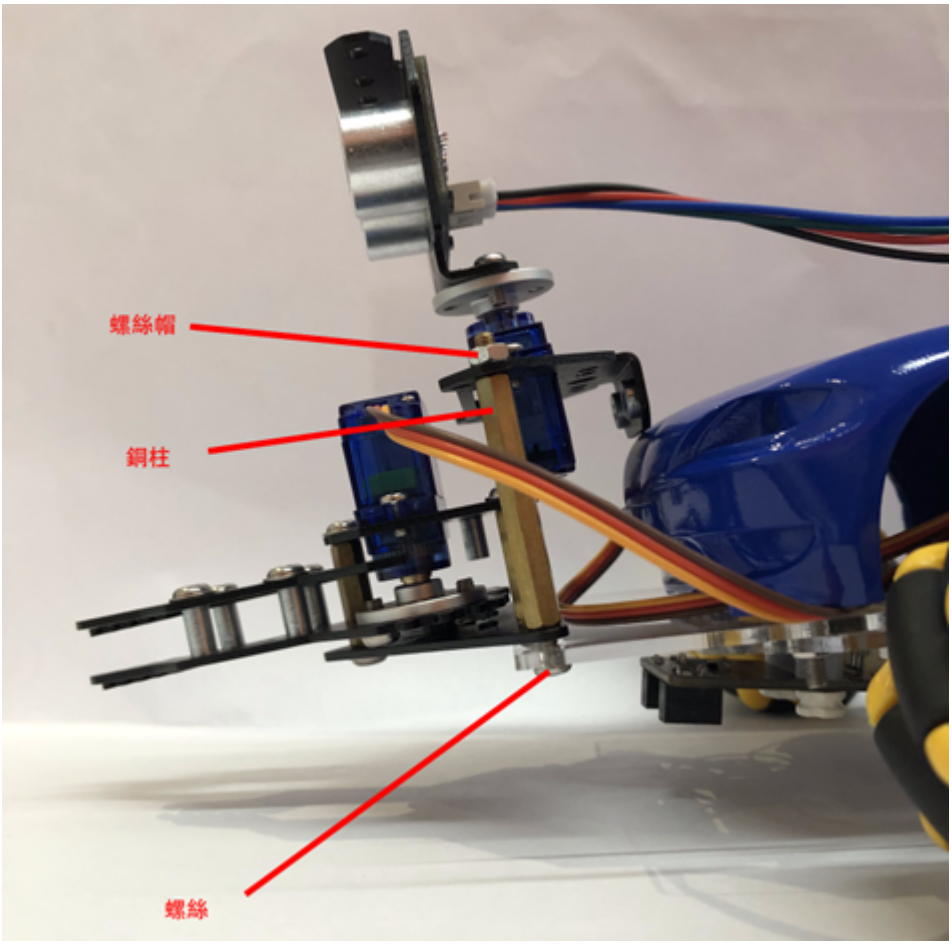
教學目標

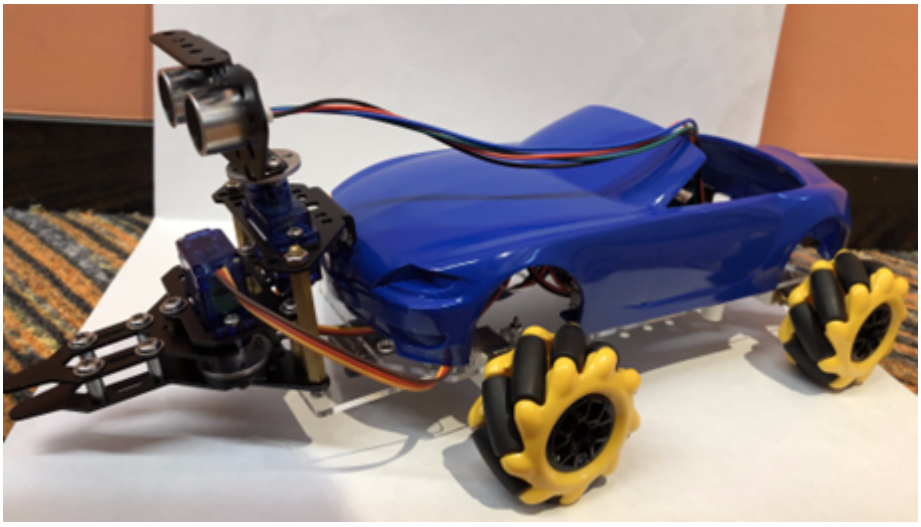
練習 1



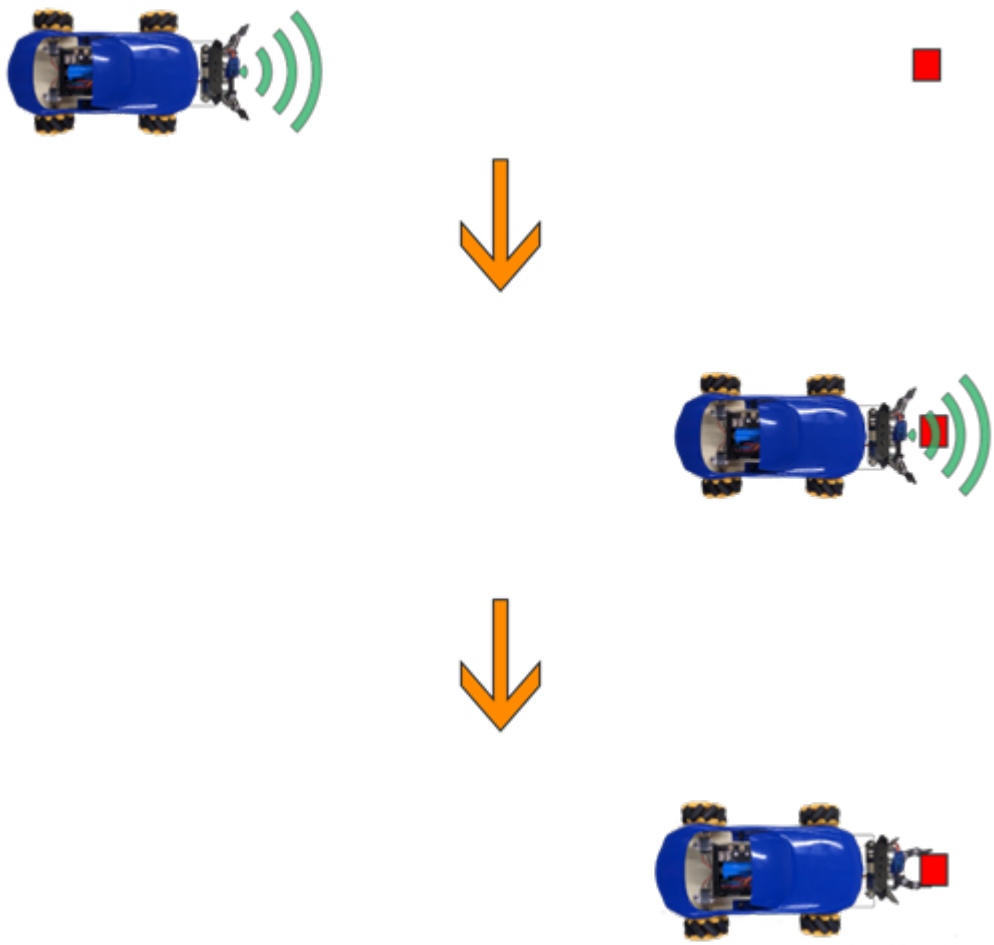
通用擴展安裝板	雲台變焦安裝板

M3*5mm
Screw

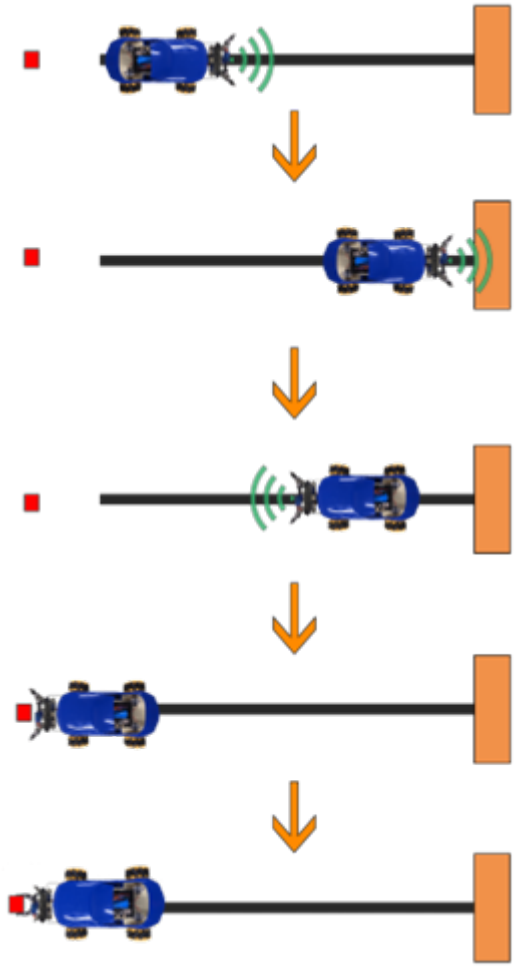

螺絲帽
銅柱
螺絲



練習 2



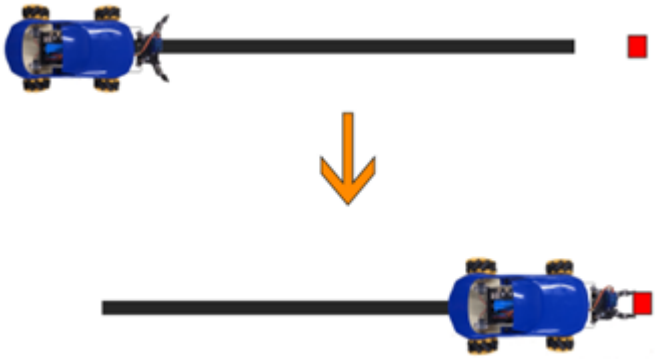
練習 3



- 建立變數“goal”記F目的地，F用於條件積木（以文字“start”和“end”分F代表起點和終點）
- 留意超聲波、巡F及“goal”的條件檢查次序

答案

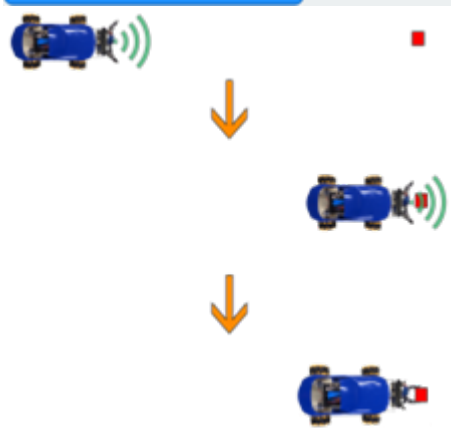
練習 1



當啟動時
舵機 S1 角度 0
預設夾子打開
重複無限次
變數 L 設為 數位信號讀取 引腳 P13
變數 R 設為 數位信號讀取 引腳 P14
呼叫 LineFollow
定義函式 LineFollow
如果 L = 0
馬達 M1 速度 60
馬達 M3 速度 60
否則如果 L =
馬達 M1 速度 80
馬達 M3 速度 80
否則如果 L =
馬達 M1 速度 0
馬達 M4 速度 0
否則如果 L =
舵機 S1 角度 110
停止所有馬達

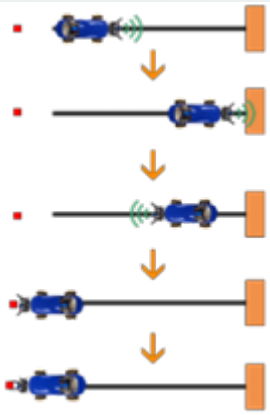
練習 2

重複無限次
如果
RGB超聲波讀取距離(cm)
>
4
那麼
舵機
S1
角度
0
夾子打開
暫停
100
毫秒
馬達
M1
速度
80
M2
速度
80
馬達
M1
速度
80
M2
速度
80
否則如果
RGB超聲波讀取距離(cm)
≤
4
那麼
停止所有馬達
舵機
S1
角度
110
夾子合上



練習 3

重複無限次
變數 L 設為 數位信號讀取 引腳 P13
變數 R 設為 數位信號讀取 引腳 P14
呼叫 Ultrasonic
當啟動時
變數 goal 設為 "end"
預設目標地為終點
定義函式 Ultrasonic
往終點的程序
如果 goal = "end" 那麼
如果 RGB超聲波讀取距離(cm) > 5 那麼
舵機 S1 角度 0
呼叫 LineFollower
暫停 200 毫秒
超聲波探測到物件
否則如果 RGB超聲波讀取距離(cm) ≤ 5 那麼
停止所有馬達
馬達 M1 速度 100 M2 速度 -100
馬達 M3 速度 100 M4 速度 -100
暫停 2000 毫秒
旋轉180度
變數 goal 設為 "start"
改爲前往起點
否則如果 goal = "start" 那麼
呼叫 LineFollower
回起點的程序
定義函式 LineFollower
如果 L = 0 且 R = 0 那麼
馬達 M1 速度 60 M2 速度 60
馬達 M3 速度 60 M4 速度 60
否則如果 L = 1 且 R = 0 那麼
馬達 M1 速度 80 M2 速度 0
馬達 M3 速度 80 M4 速度 0
否則如果 L = 0 且 R = 1 那麼
馬達 M1 速度 0 M2 速度 80
馬達 M3 速度 0 M4 速度 80
否則如果 L = 1 且 R = 1 那麼
如果 goal = "start" 那麼
當走出黑線
目標地為終點時
合上夾子
停止所有馬達
舵機 S1 角度 110
否則
停止所有馬達

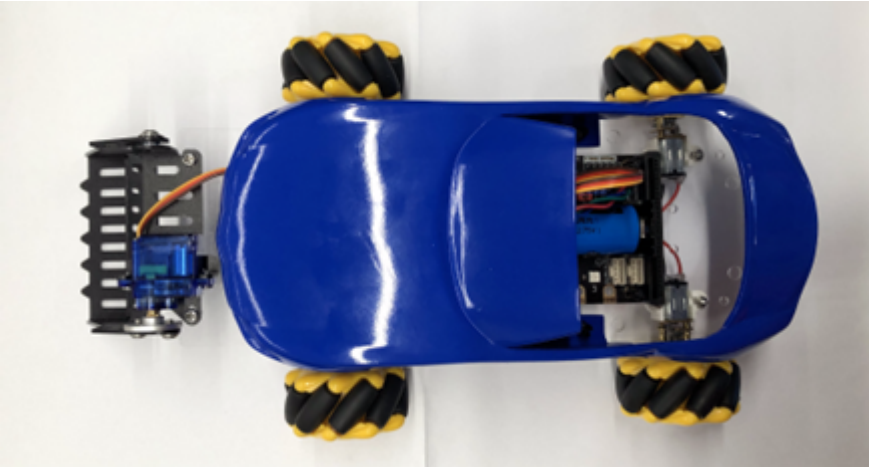


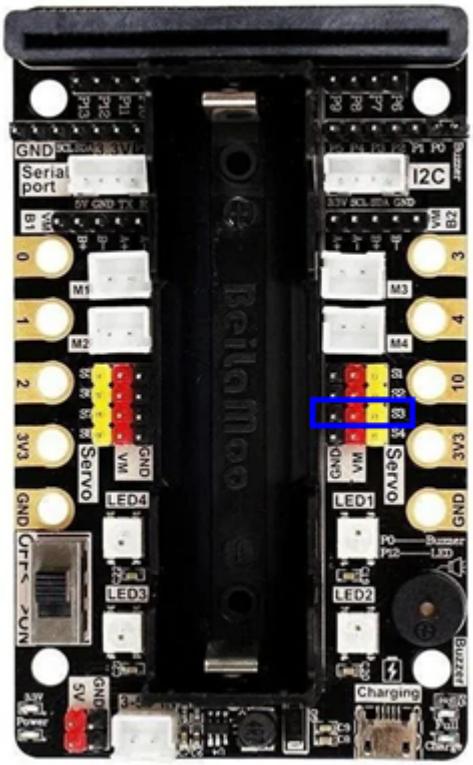
1.4.3 第三課

簡介

教學目標

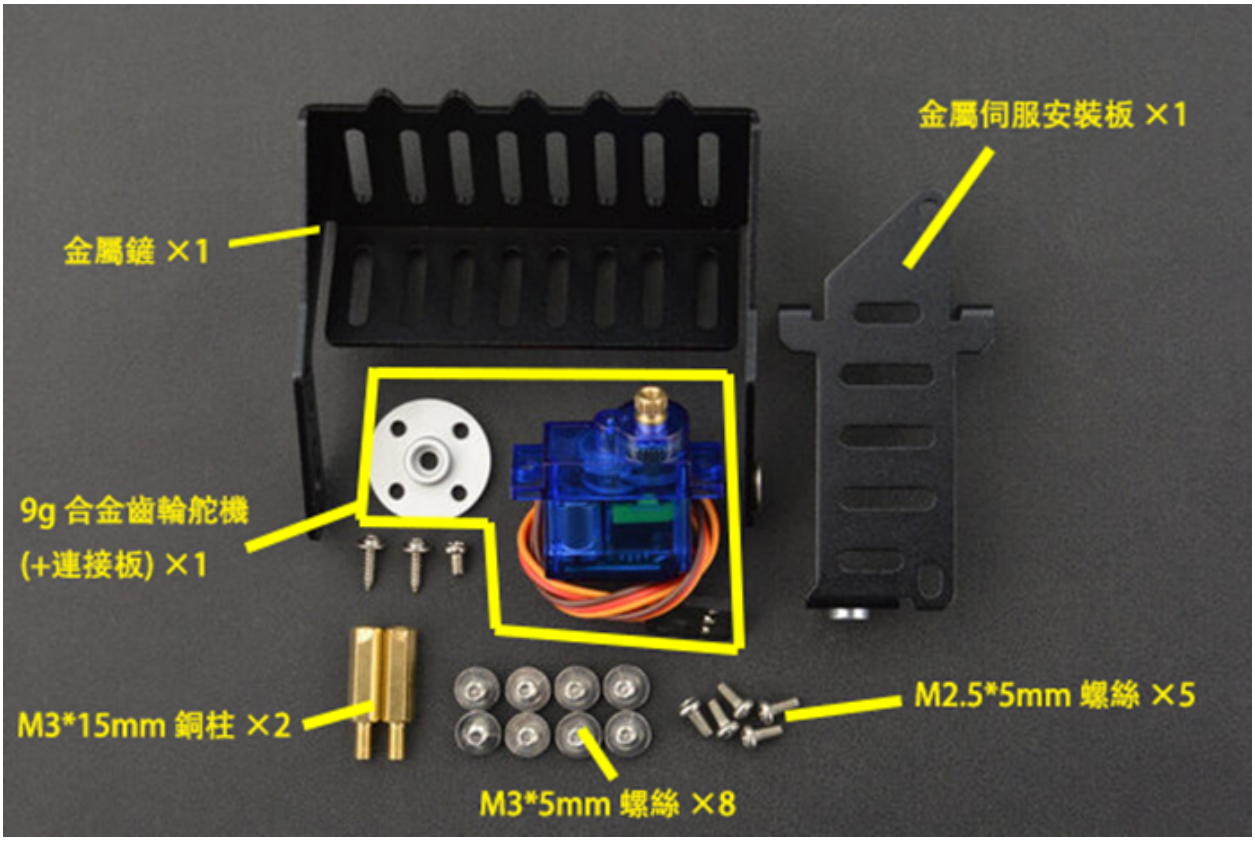
認識 Micro:bit 擴展工具——鏟子





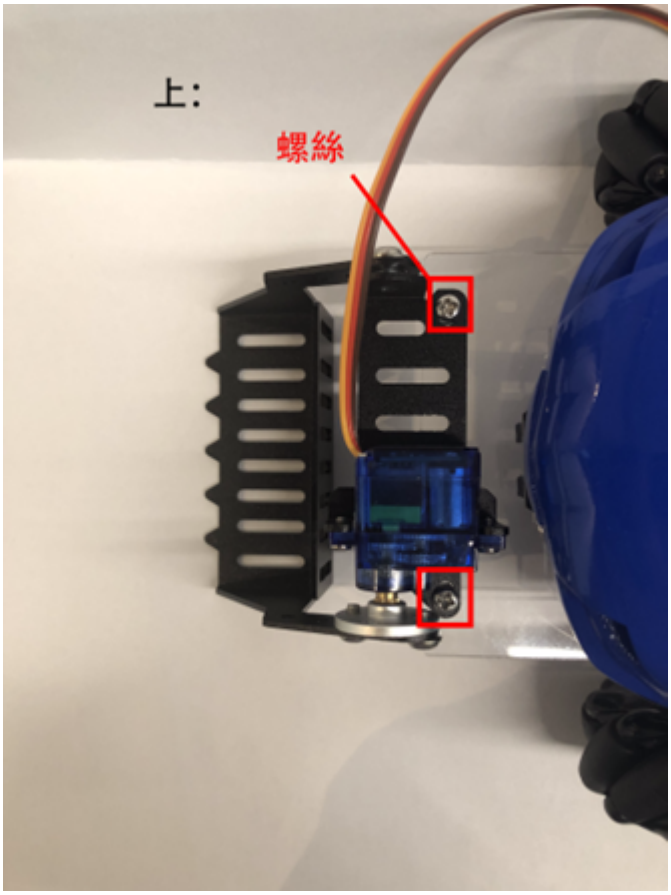
機械鏟子的原理及功用

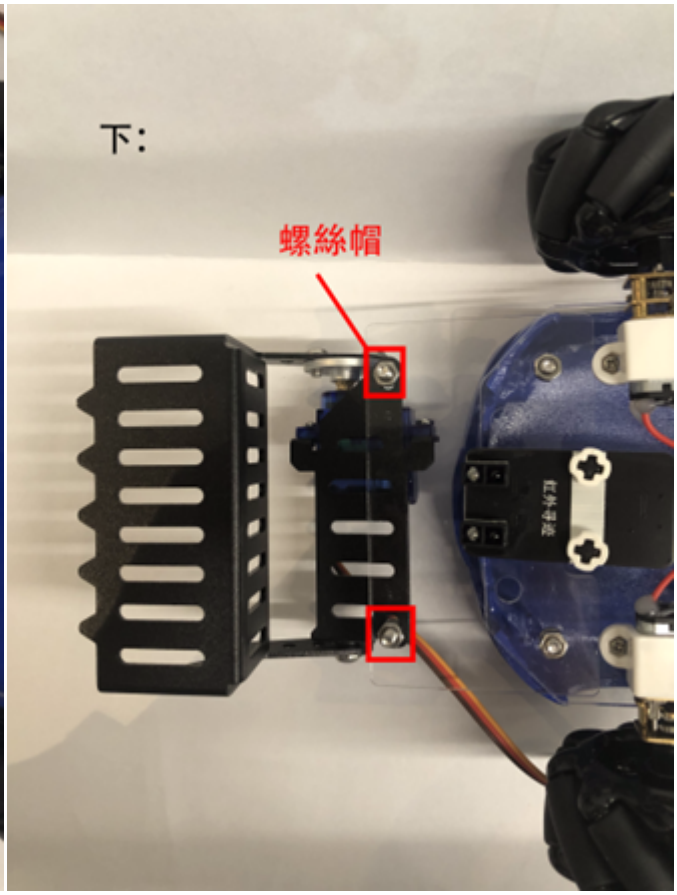
安裝鏟子


金屬伺服安裝板 ×1
金屬鏟 ×1
9g 合金齒輪舵機
(+連接板) ×1
M3*15mm 銅柱 ×2
M2.5*5mm 螺絲 ×5
M3*5mm 螺絲 ×8

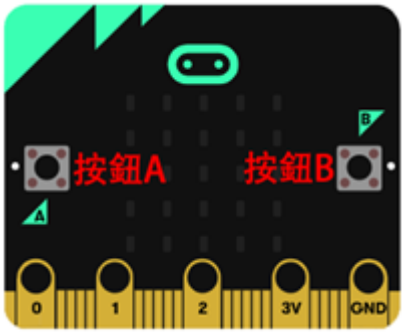
M2.5*5mm Screw
Step1

M3*5mm Screw
M3*5mm
Screw
Step2

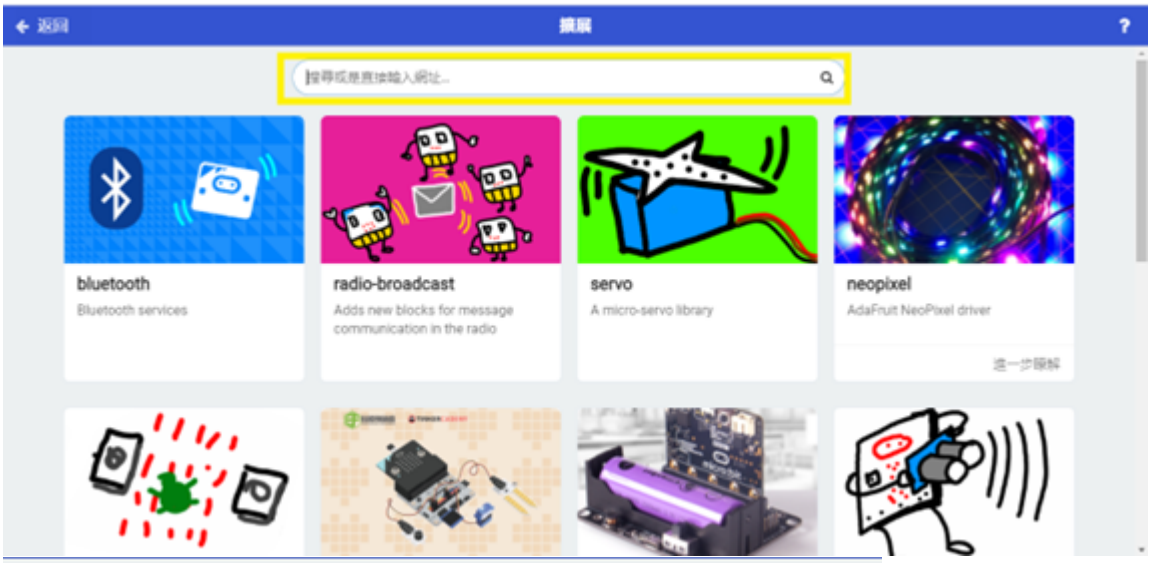

上:
螺絲

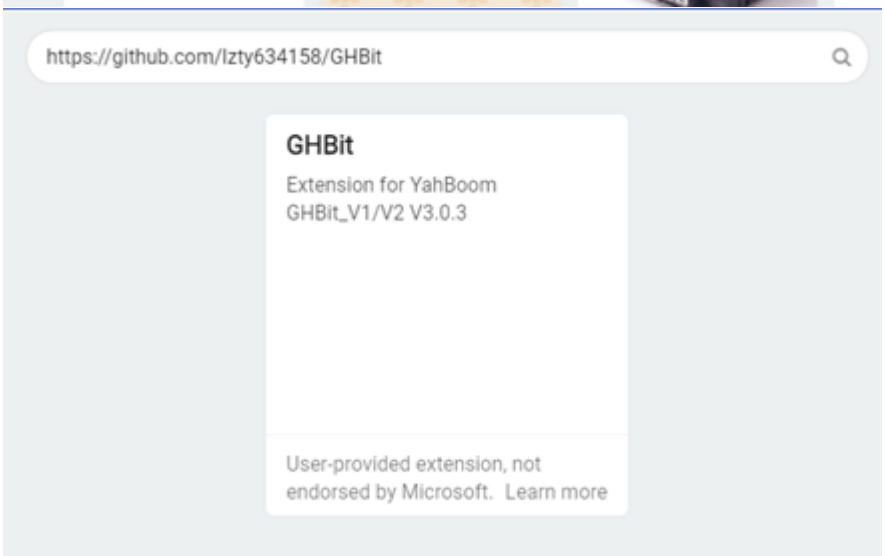

下:
螺絲帽

練習一


按鈕A
按鈕B
0
1
2
3V
GND

練習二


擴展
bluetooth
Bluetooth services
radio-broadcast
Adds new blocks for message communication in the radio
servo
A micro-servo library
neopixel
AdaFruit NeoPixel driver

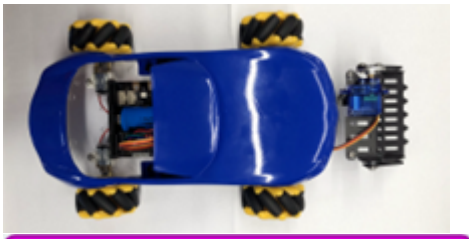

https://github.com/lzty634158/GHBit
GHBit
Extension for YahBoom
GHBit_V1/V2 V3.0.3
User-provided extension, not endorsed by Microsoft. Learn more

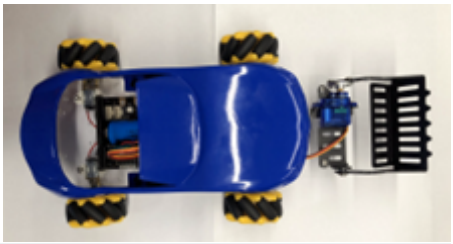
- B1 和 B4 按鈕操作車子左右水平移動
- B2 和 B3 按鈕操作鏟子鏟起放下
- 操縱杆控制車子前後移動及向左右轉
- Ⓕ控器待機時車子停下
- 鏟子鏟起和放下時，分Ⓕ播放 DoReMi 和 SoFaMi （或任何兩段不同旋律）

- 參考初階第十科，以接收文字作爲行動條件
- 運用音效 » 演奏旋律 ♫ …速度…(bpm) 積木模塊

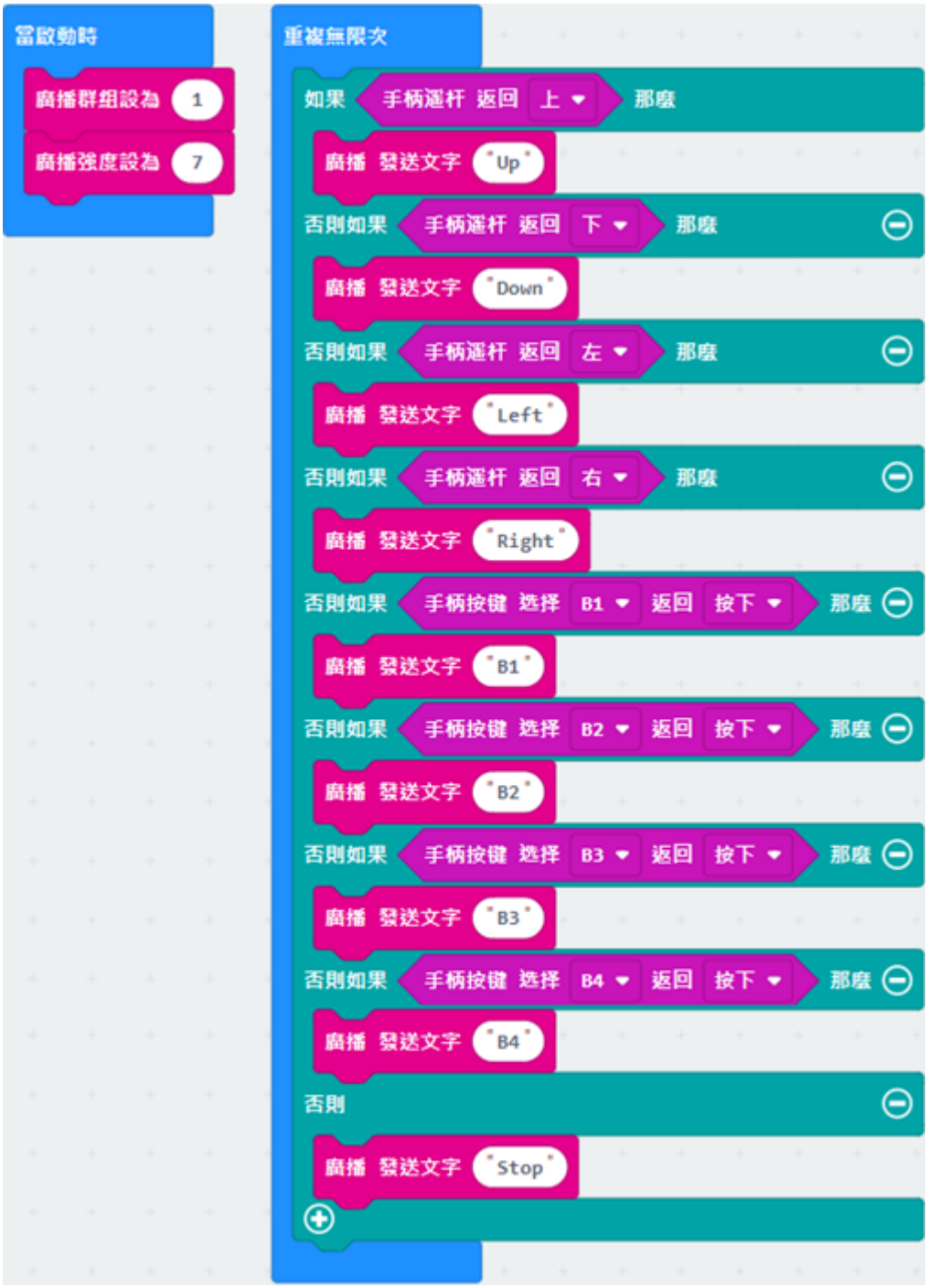
答案

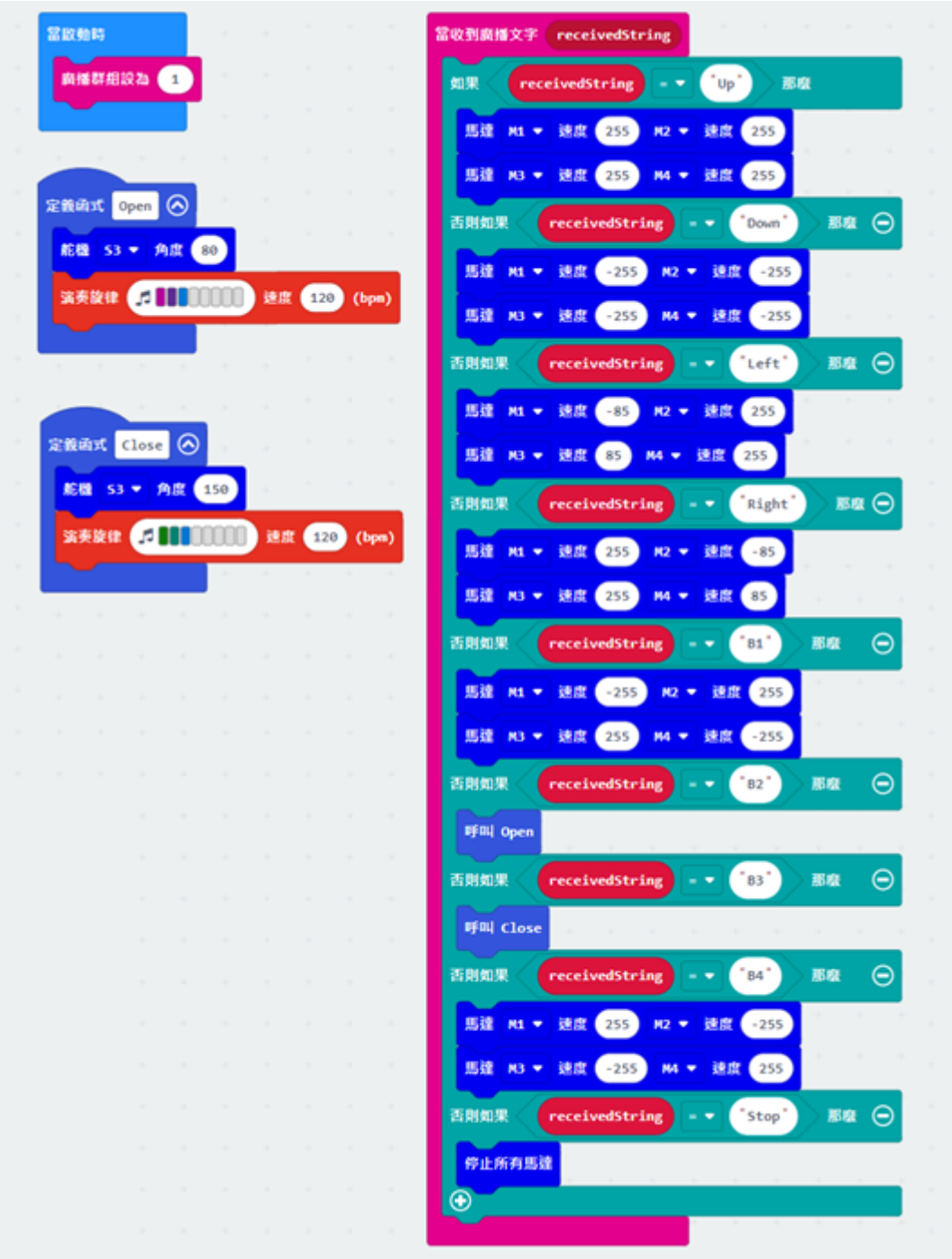
練習一





練習二


當啟動時
廣播群組設為 1
廣播強度設為 7
重複無限次
如果 手柄遥杆 返回 上 那麼
廣播 發送文字 "Up"
否則如果 手柄遥杆 返回 下 那麼
廣播 發送文字 "Down"
否則如果 手柄遥杆 返回 左 那麼
廣播 發送文字 "Left"
否則如果 手柄遥杆 返回 右 那麼
廣播 發送文字 "Right"
否則如果 手柄按键 选择 B1 返回 按下 那麼
廣播 發送文字 "B1"
否則如果 手柄按键 选择 B2 返回 按下 那麼
廣播 發送文字 "B2"
否則如果 手柄按键 选择 B3 返回 按下 那麼
廣播 發送文字 "B3"
否則如果 手柄按键 选择 B4 返回 按下 那麼
廣播 發送文字 "B4"
否則
廣播 發送文字 "Stop"

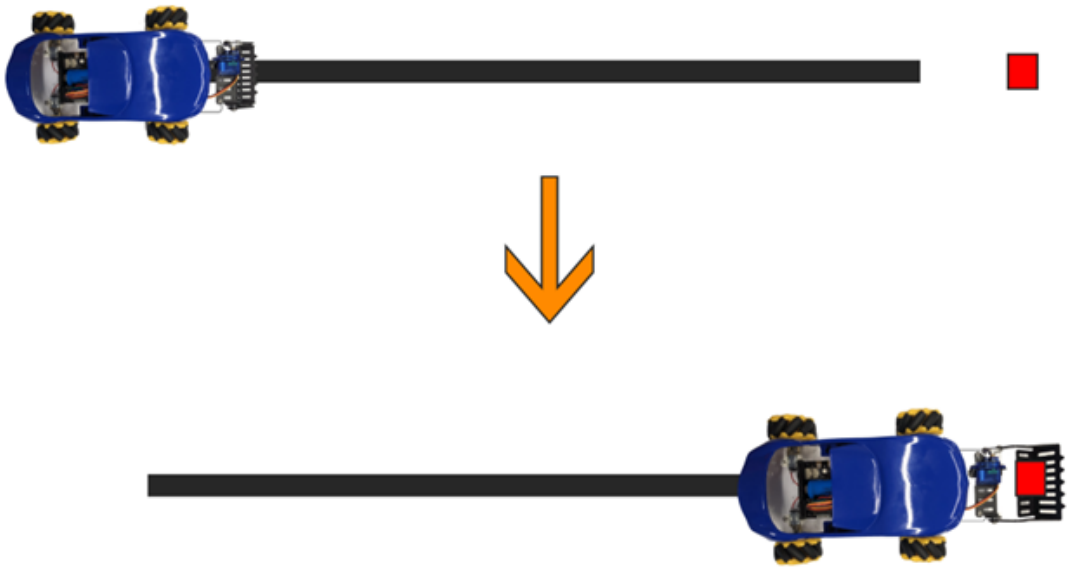

當啟動時
廣播群組設為 1
定義函式 Open
舵機 S3 角度 80
演奏旋律 速度 120 (bpm)
定義函式 Close
舵機 S3 角度 150
演奏旋律 速度 120 (bpm)
當收到廣播文字 receivedString
如果 receivedString = "Up" 那麼
馬達 M1 速度 255 M2 速度 255
馬達 M3 速度 255 M4 速度 255
否則如果 receivedString = "Down" 那麼
馬達 M1 速度 -255 M2 速度 -255
馬達 M3 速度 -255 M4 速度 -255
否則如果 receivedString = "Left" 那麼
馬達 M1 速度 -85 M2 速度 255
馬達 M3 速度 85 M4 速度 255
否則如果 receivedString = "Right" 那麼
馬達 M1 速度 255 M2 速度 -85
馬達 M3 速度 255 M4 速度 85
否則如果 receivedString = "B1" 那麼
馬達 M1 速度 -255 M2 速度 255
馬達 M3 速度 255 M4 速度 -255
否則如果 receivedString = "B2" 那麼
呼叫 Open
否則如果 receivedString = "B3" 那麼
呼叫 Close
否則如果 receivedString = "B4" 那麼
馬達 M1 速度 255 M2 速度 -255
馬達 M3 速度 -255 M4 速度 255
否則如果 receivedString = "Stop" 那麼
停止所有馬達

1.4.4 第四課

簡介

教學目標:

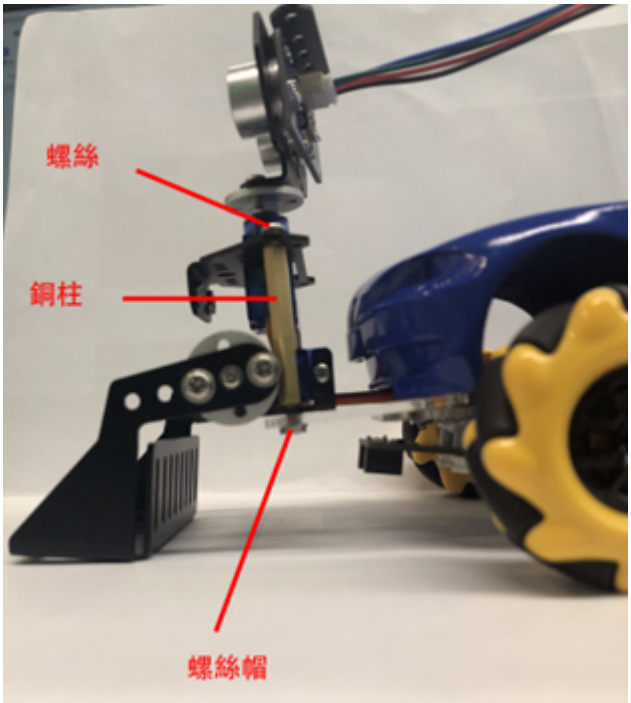
練習一

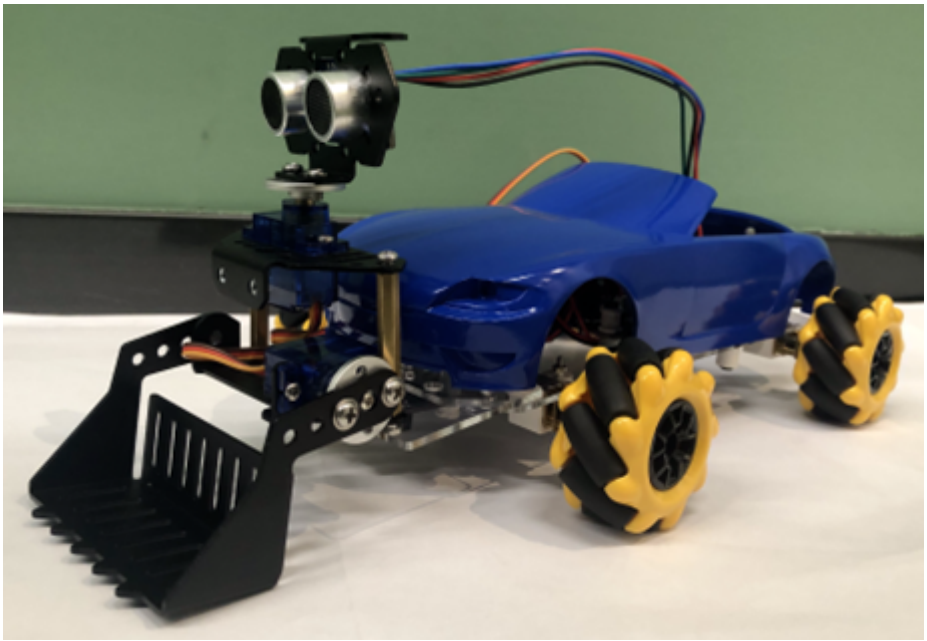


安裝超聲波傳感器:

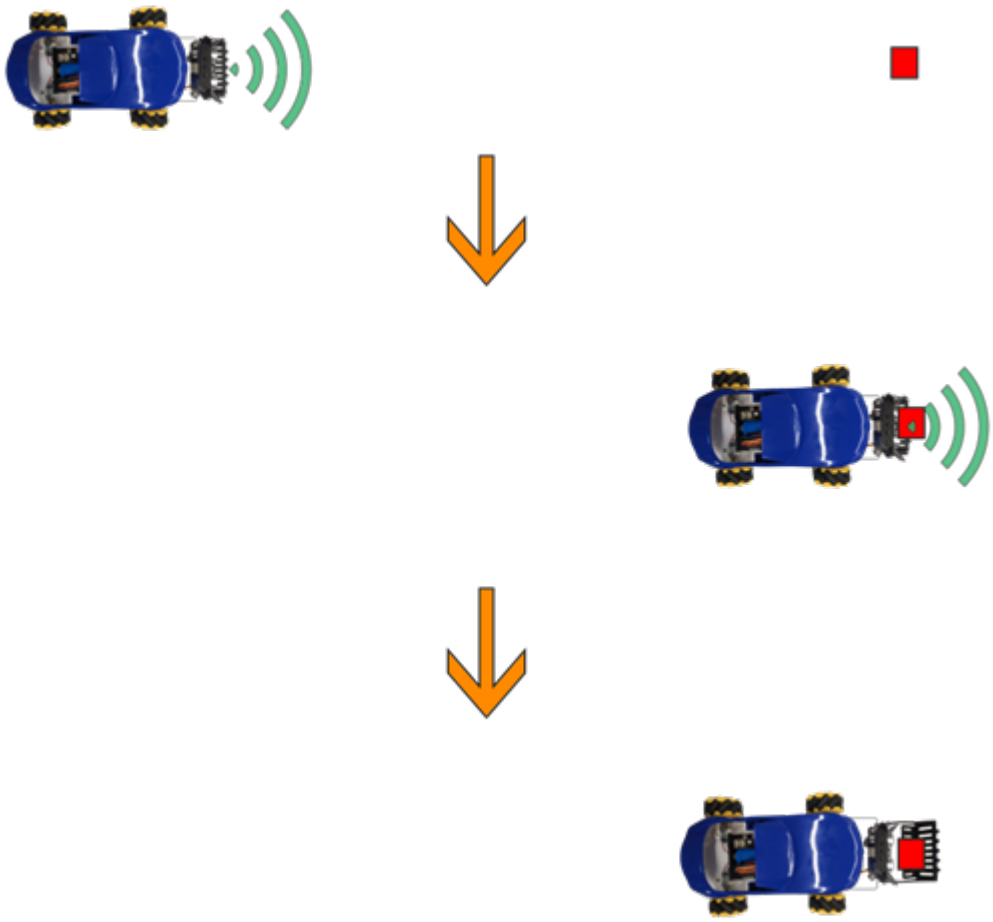
通用擴展安裝板	雲台變焦安裝板

M3*5mm
Screw

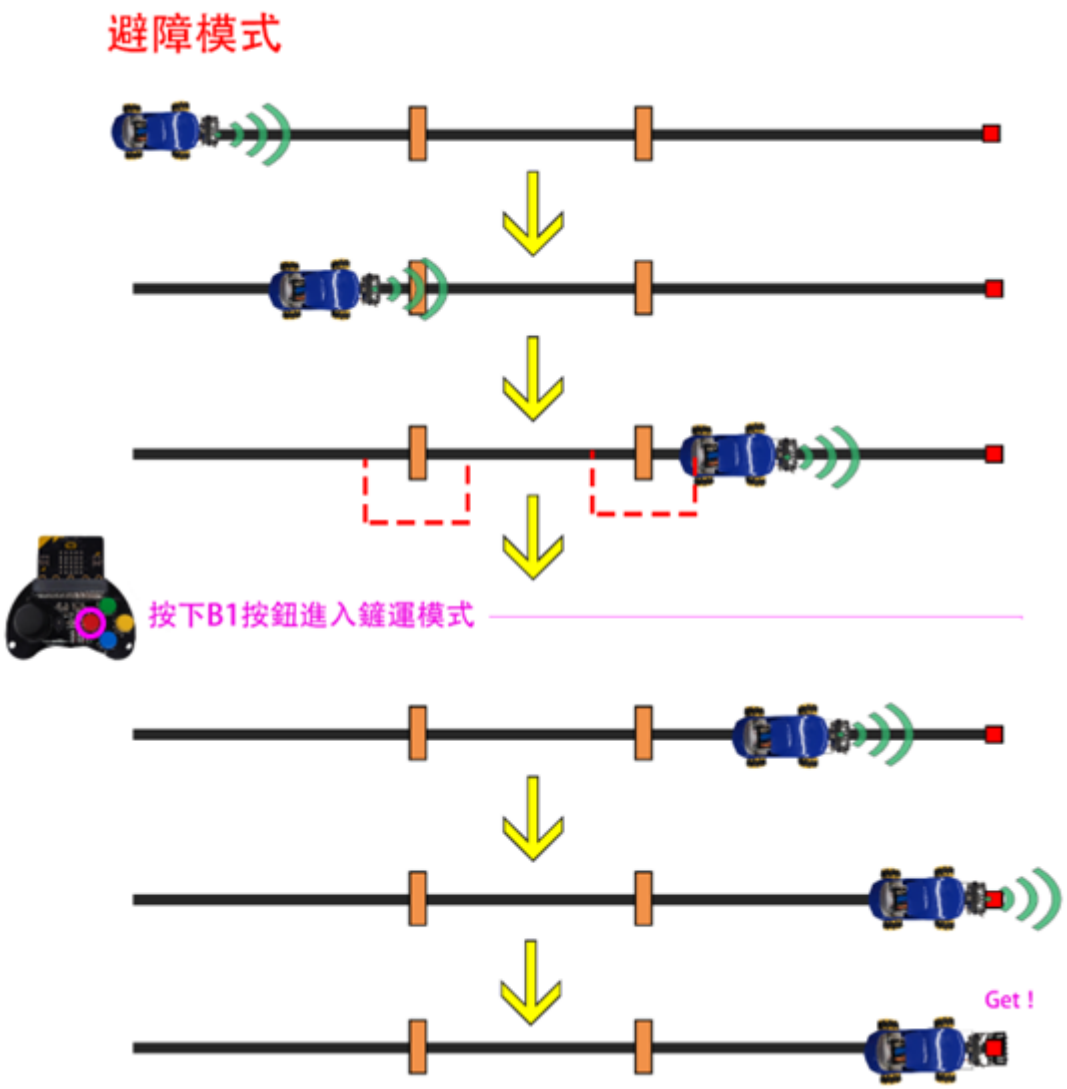

螺絲
銅柱
螺絲帽



練習二

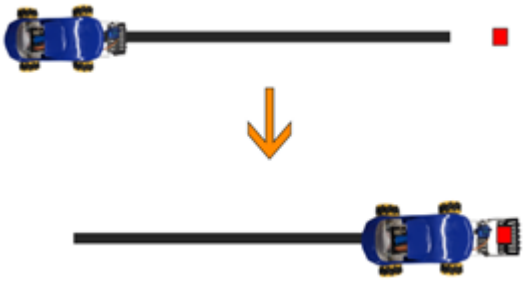


練習三



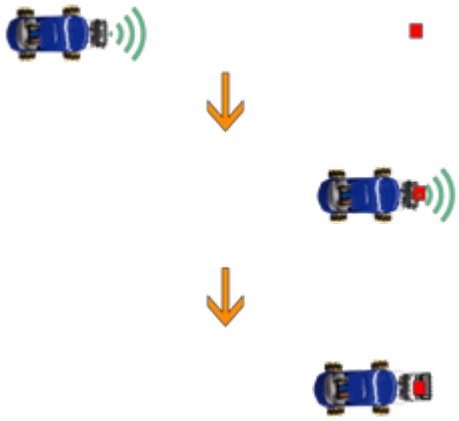
答案

練習一



當啟動時
舵機 S3 角度 150
放下鏈子
重複無限次
變數 L 設為 數位信號讀取 引腳 P13
變數 R 設為 數位信號讀取 引腳 P14
呼叫 LineFollow
定義函式 LineFollow
如果 L = 0
馬達 M1 速度 60 M2
馬達 M3 速度 60 M4
否則如果 L = 1
馬達 M1 速度 80 M2
馬達 M3 速度 80 M4
否則如果 L = 0
馬達 M1 速度 0 M2
馬達 M4 速度 0 M3
否則如果 L = 1
舵機 S3 角度 80
停止所有馬達

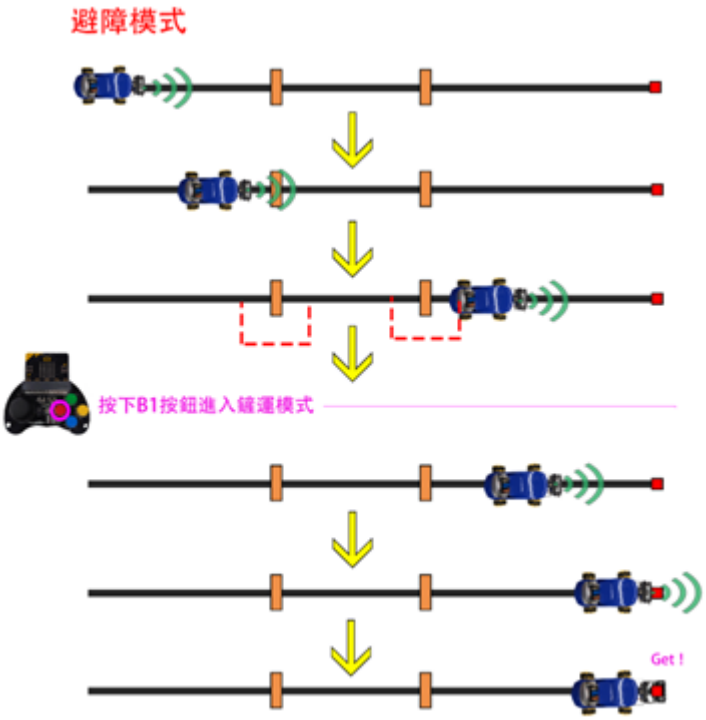
練習二

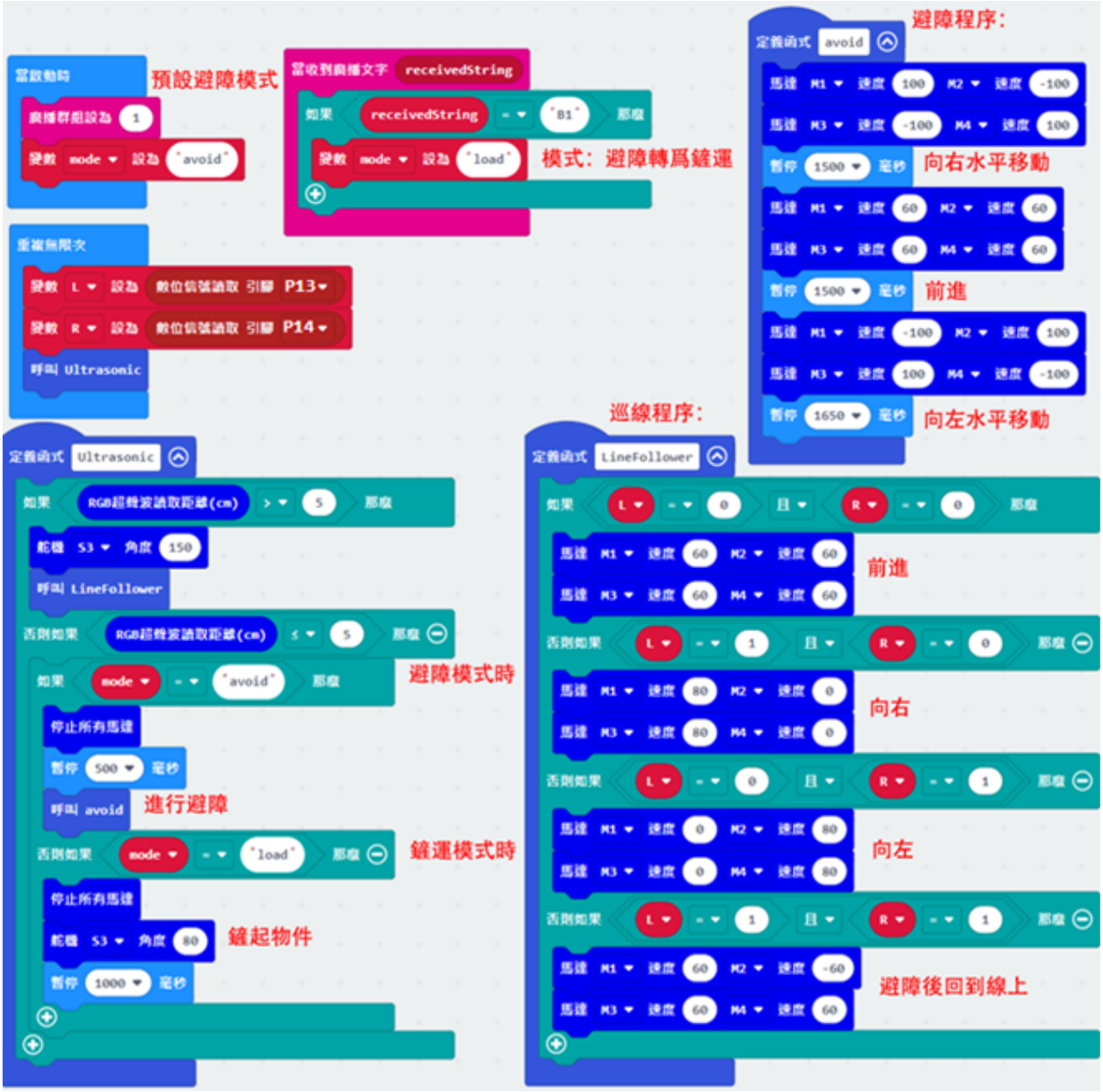


重複無限次
如果
RGB超聲波讀取距離(cm)
>
4
舵機
S3
角度
150
放下鏟子
暫停
100
毫秒
馬達
M1
速度
80
M2
速度
80
馬達
M1
速度
80
M2
速度
80
否則如果
RGB超聲波讀取距離(cm)
≤
4
停止所有馬達
舵機
S3
角度
80
鏟起

練習三

當啟動時
廣播群組設為 1
廣播強度設為 7
重複無限次
如果 手柄按键 选择 B1 返回 按下 那麼
廣播 發送文字 "B1"


避障模式
按下B1按鈕進入繼運模式
Get !

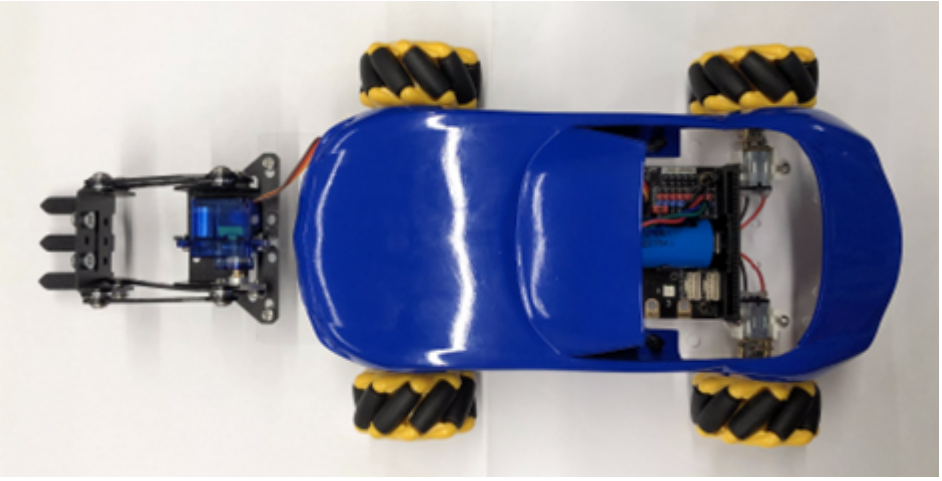

預設避障模式
模式：避障轉爲鏟運
避障程序：
向右水平移動
前進
向左水平移動
巡線程序：
避障模式時
進行避障
鏟運模式時
鏟起物件
前進
向右
向左
避障後回到線上

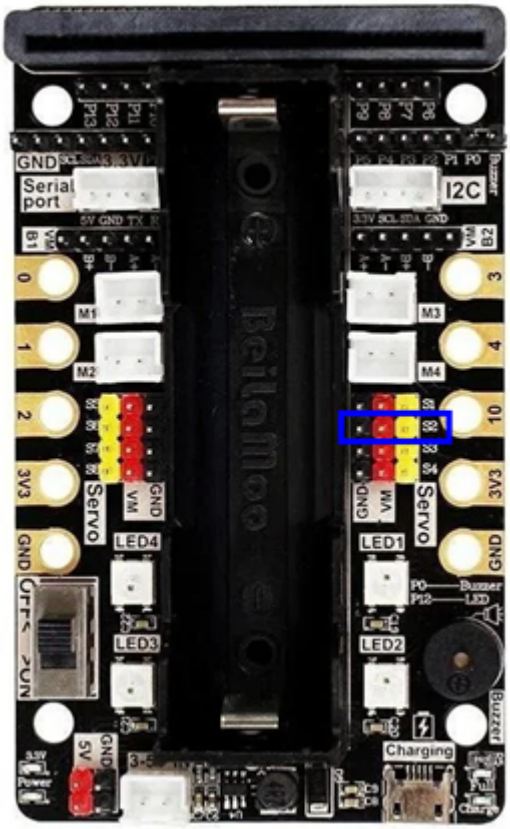
1.4.5 第五課

簡介

教學目標

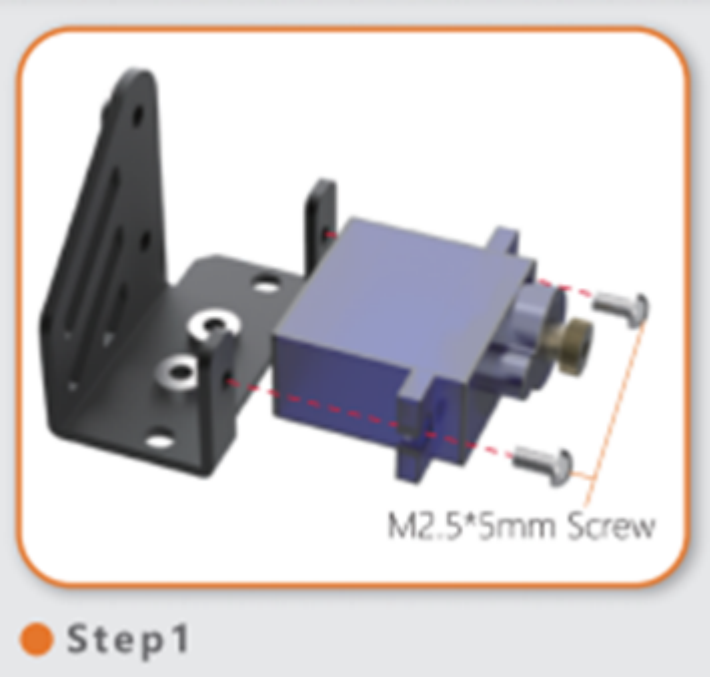
認識 Micro:bit 擴展工具——叉子

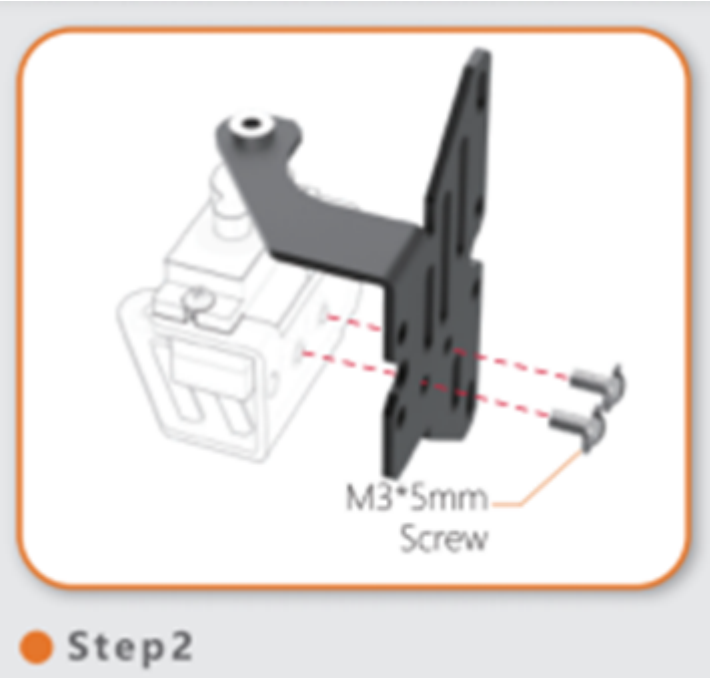




機械叉子的原理及功用

安裝叉車




M3*5mm
Screw
Step2

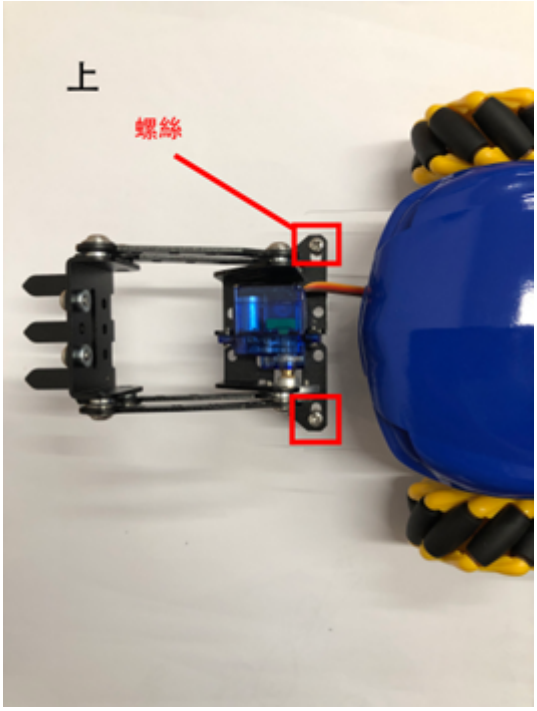
M3*5mm
Screw
Step3

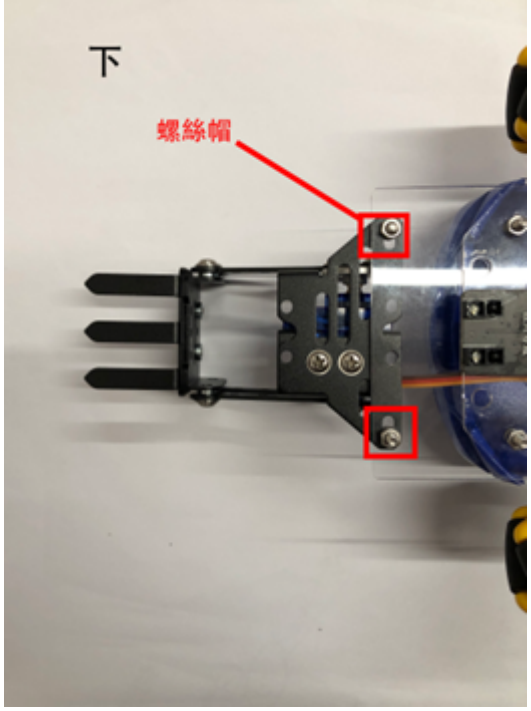
M3*5mm
Screw
Step4

M3*5mm
Screw
Step5

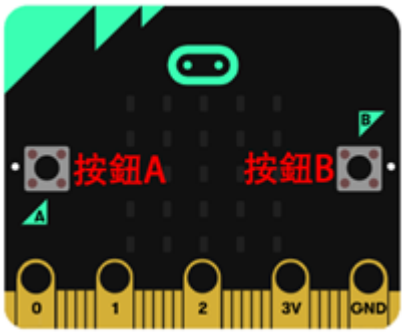
M2.5*5mm Screw
M3*5mm Screw
Step6

M3*5mm Screw
Step7





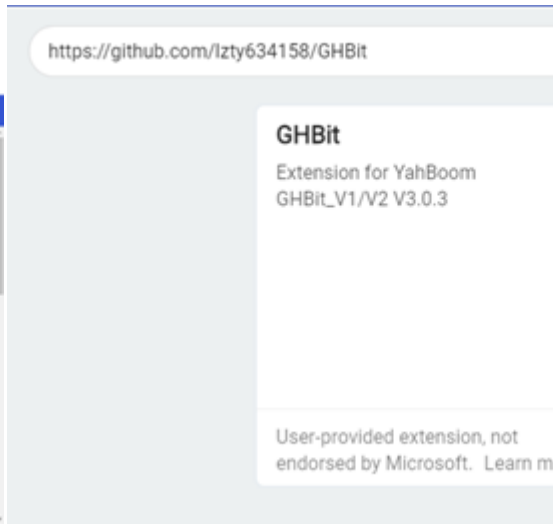
練習一



練習二

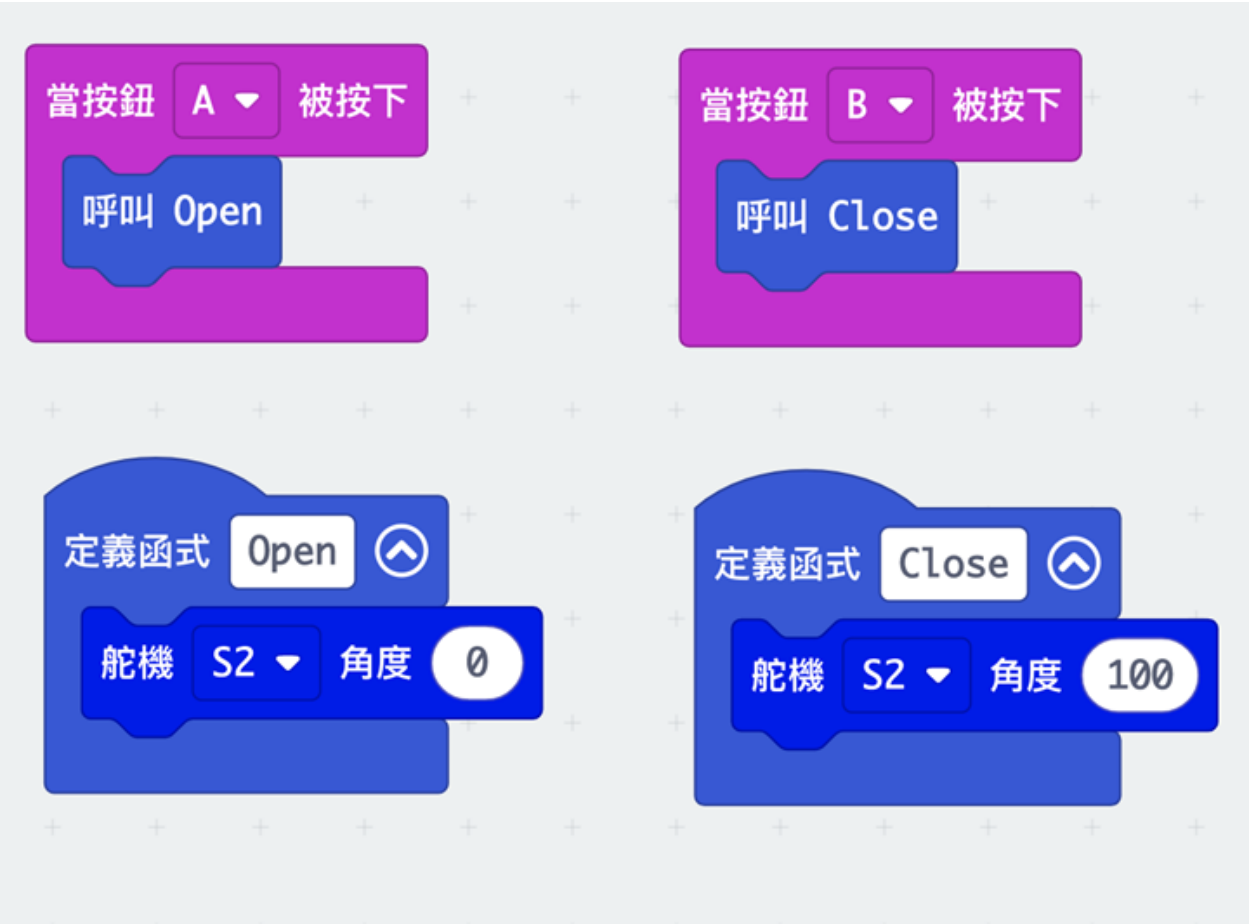
練習三

bluetooth
radio-broadcast
servo
neopixel

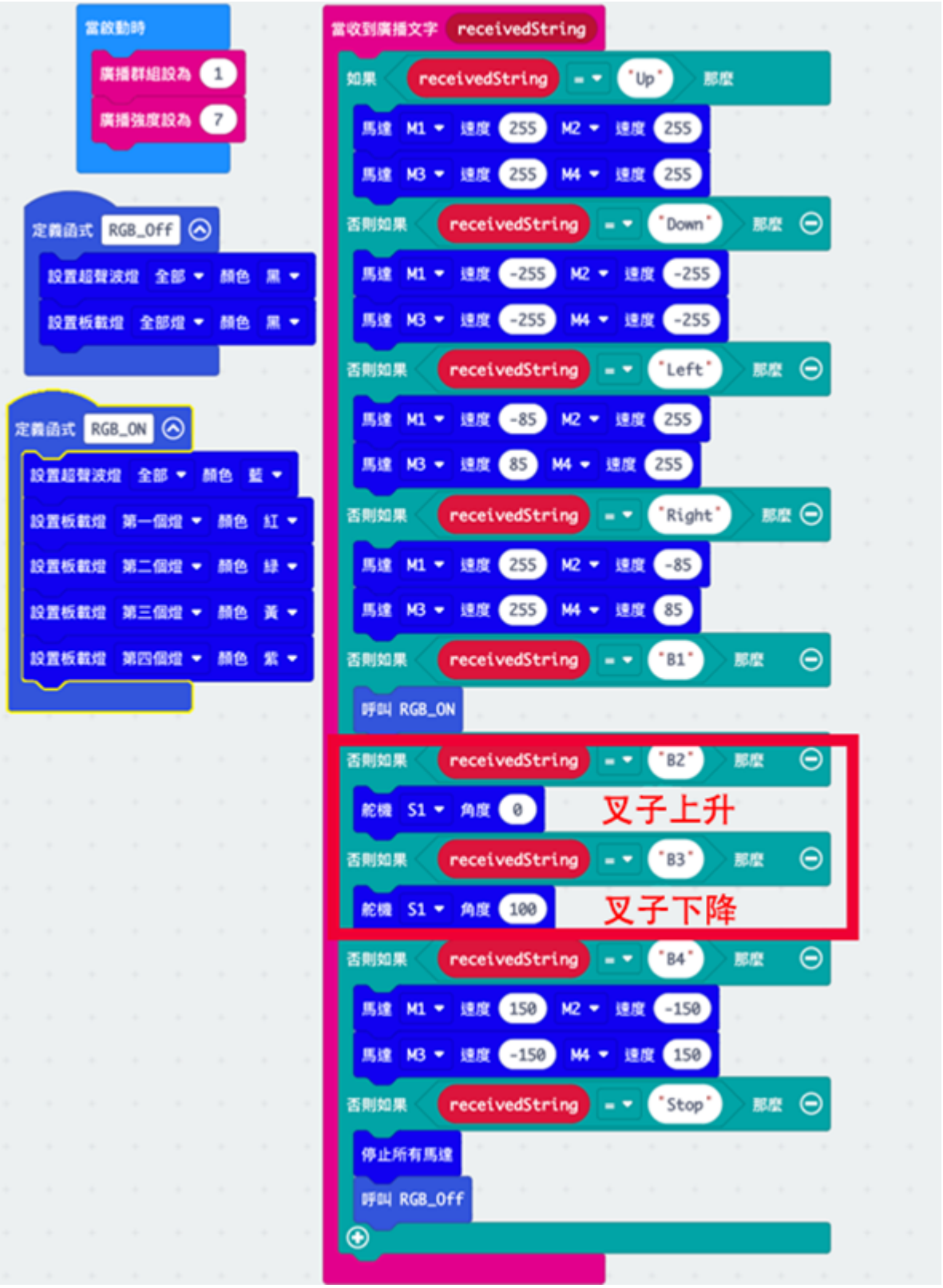

https://github.com/lzty634158/GHBit
GHBit
Extension for YahBoom
GHBit_V1/V2 V3.0.3
User-provided extension, not
endorsed by Microsoft. Learn m

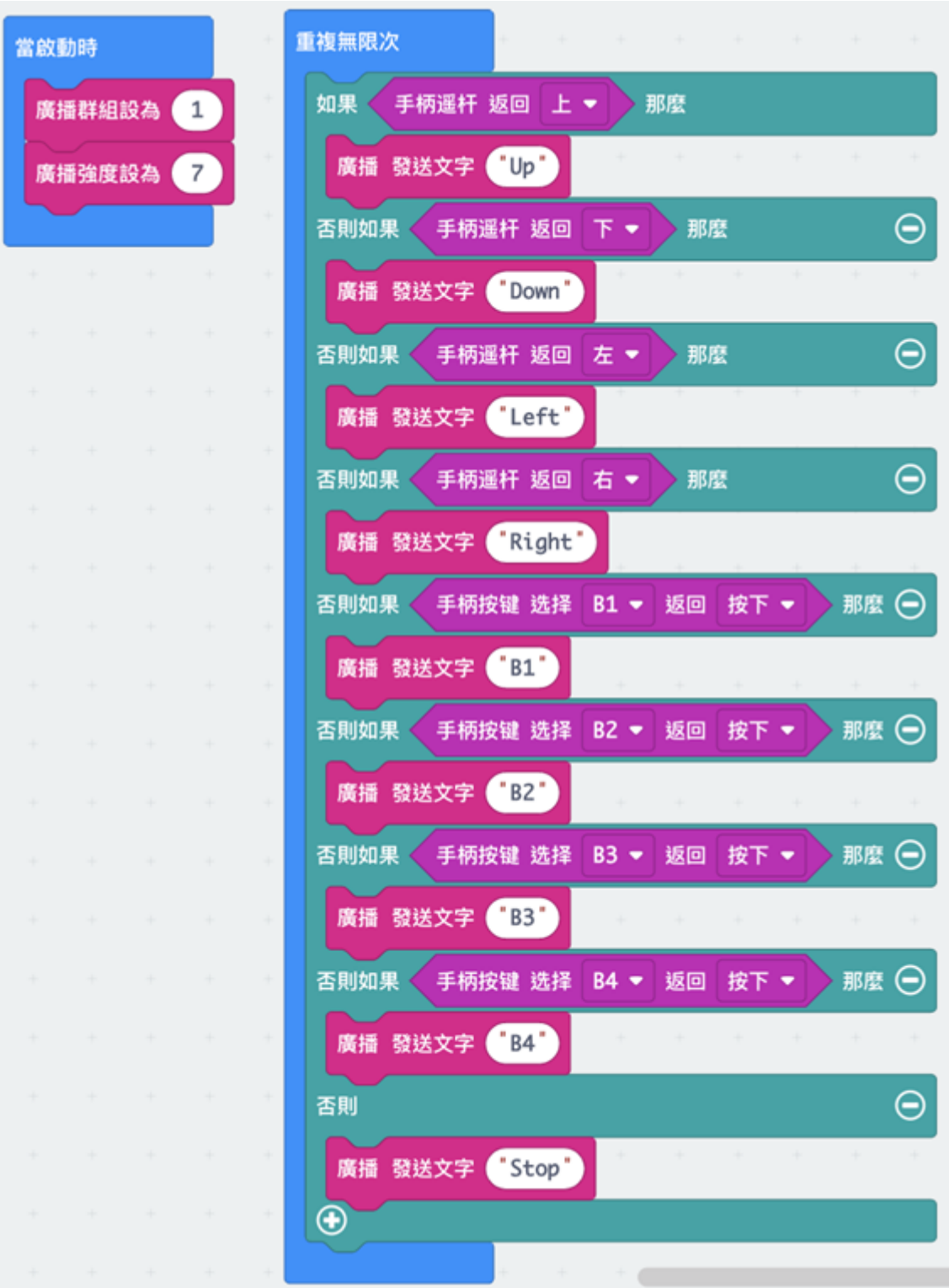
答案

練習一


當按鈕 A 被按下
呼叫 Open
當按鈕 B 被按下
呼叫 Close
定義函式 Open
舵機 S2 角度 0
定義函式 Close
舵機 S2 角度 100

練習三


當啟動時
廣播群組設為 1
廣播強度設為 7
定義函式 RGB_Off
設置超聲波燈 全部 顏色 黑
設置板載燈 全部燈 顏色 黑
定義函式 RGB_ON
設置超聲波燈 全部 顏色 藍
設置板載燈 第一個燈 顏色 紅
設置板載燈 第二個燈 顏色 綠
設置板載燈 第三個燈 顏色 黃
設置板載燈 第四個燈 顏色 紫
當收到廣播文字 receivedString
如果 receivedString = "Up" 那麼
馬達 M1 速度 255 M2 速度 255
馬達 M3 速度 255 M4 速度 255
否則如果 receivedString = "Down" 那麼
馬達 M1 速度 -255 M2 速度 -255
馬達 M3 速度 -255 M4 速度 -255
否則如果 receivedString = "Left" 那麼
馬達 M1 速度 -85 M2 速度 255
馬達 M3 速度 85 M4 速度 255
否則如果 receivedString = "Right" 那麼
馬達 M1 速度 255 M2 速度 -85
馬達 M3 速度 255 M4 速度 85
否則如果 receivedString = "B1" 那麼
呼叫 RGB_ON
否則如果 receivedString = "B2" 那麼
舵機 S1 角度 0
叉子上升
否則如果 receivedString = "B3" 那麼
舵機 S1 角度 100
叉子下降
否則如果 receivedString = "B4" 那麼
馬達 M1 速度 150 M2 速度 -150
馬達 M3 速度 -150 M4 速度 150
否則如果 receivedString = "Stop" 那麼
停止所有馬達
呼叫 RGB_Off


當啟動時
廣播群組設為 1
廣播強度設為 7
重複無限次
如果 手柄遥杆 返回 上 那麼
廣播 發送文字 "Up"
否則如果 手柄遥杆 返回 下 那麼
廣播 發送文字 "Down"
否則如果 手柄遥杆 返回 左 那麼
廣播 發送文字 "Left"
否則如果 手柄遥杆 返回 右 那麼
廣播 發送文字 "Right"
否則如果 手柄按键 选择 B1 返回 按下 那麼
廣播 發送文字 "B1"
否則如果 手柄按键 选择 B2 返回 按下 那麼
廣播 發送文字 "B2"
否則如果 手柄按键 选择 B3 返回 按下 那麼
廣播 發送文字 "B3"
否則如果 手柄按键 选择 B4 返回 按下 那麼
廣播 發送文字 "B4"
否則
廣播 發送文字 "Stop"

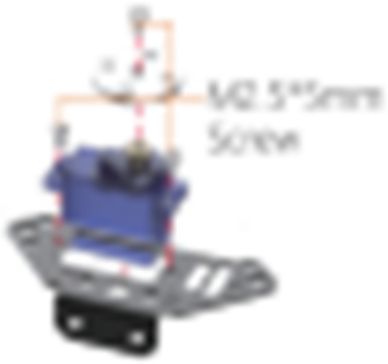
1.4.6 第六課

簡介

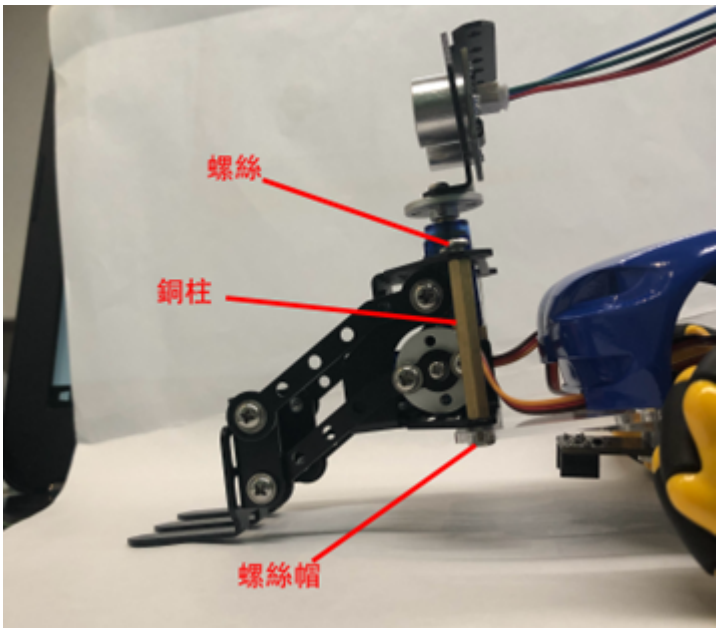
教學目標

叉車工具的運用

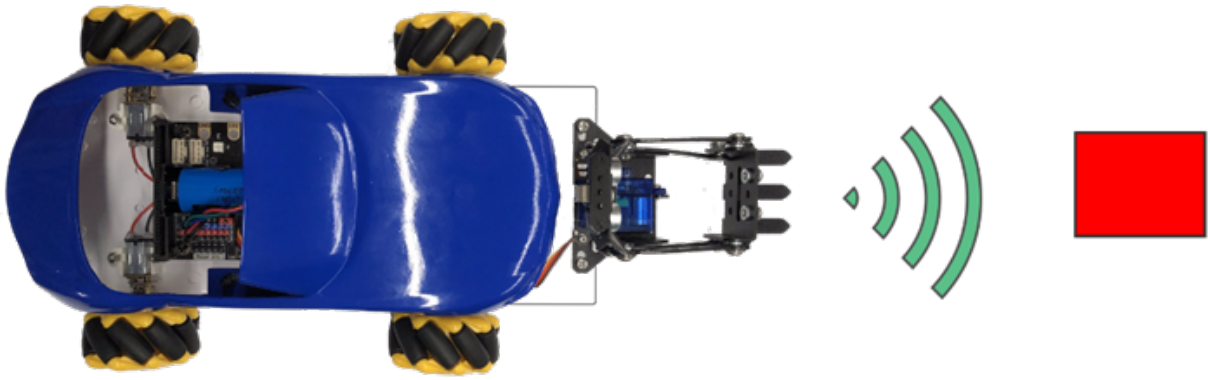
通用擴展安裝板	雲台變焦安裝板


M2.5*5mm
Screw

M3*5mm
Screw


螺絲
銅柱
螺絲帽

練習一



練習二

1.	沿黑🅕走
2.	直至感應器感測到白色，立刻暫停
3.	舉高貨物
4.	繼續沿著黑色🅕走
5.	直至感應器再次感測到白色，放下貨物

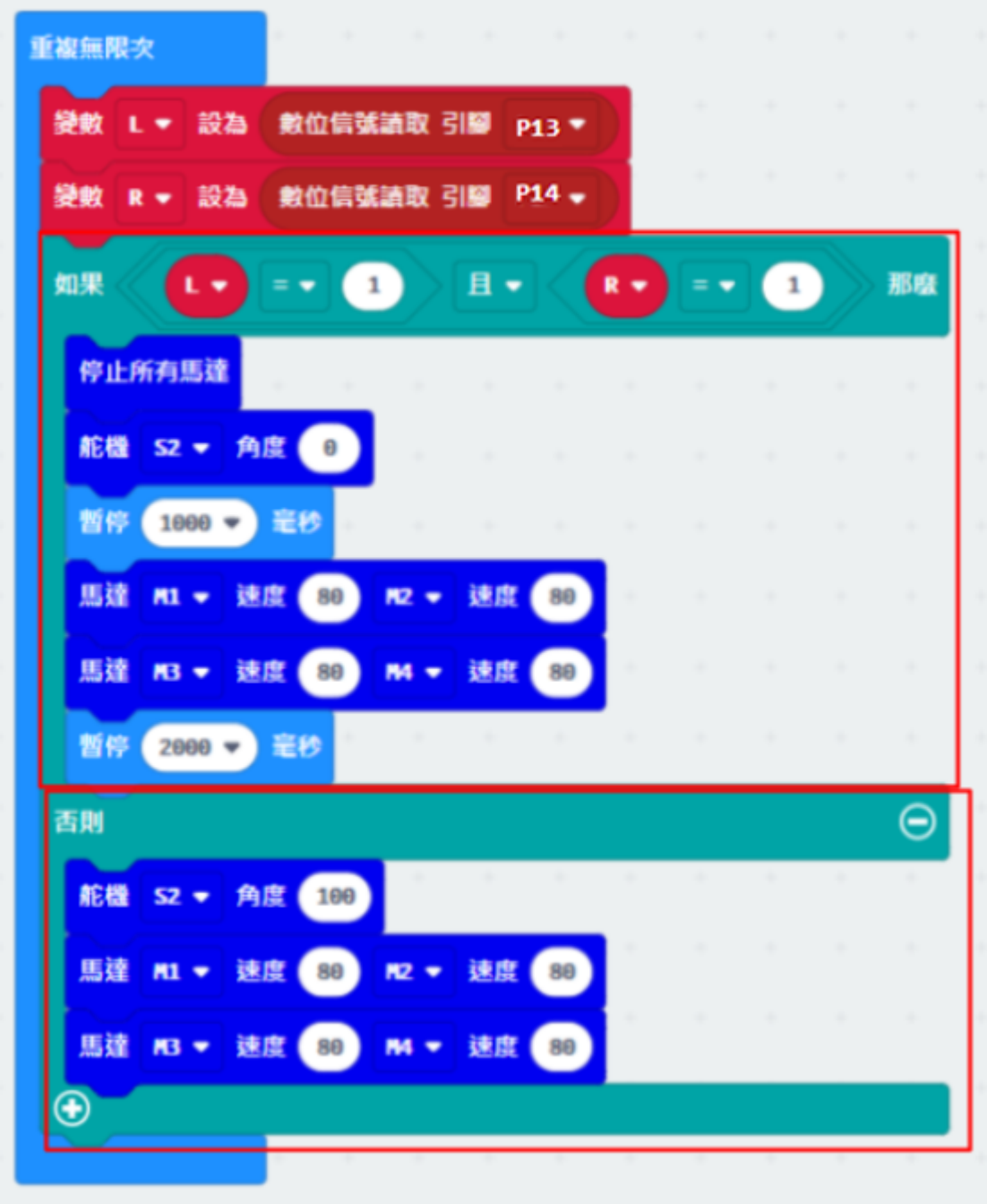
練習三

答案

練習一

重複無限次
舵機 S2 角度 100
如果 RGB超聲波讀取距離(cm) ≤ 12 那麼
停止所有馬達
舵機 S2 角度 10
暫停 1000 毫秒
馬達 M1 速度 100 M2 速度 100
馬達 M3 速度 100 M4 速度 100
否則
馬達 M1 速度 100 M2 速度 100
馬達 M3 速度 100 M4 速度 100
暫停 1000 毫秒

練習二


重複無限次
變數 L 設為 數位信號讀取 引腳 P13
變數 R 設為 數位信號讀取 引腳 P14
如果 L = 1 且 R = 1 那麼
停止所有馬達
舵機 S2 角度 0
暫停 1000 毫秒
馬達 M1 速度 80 M2 速度 80
馬達 M3 速度 80 M4 速度 80
暫停 2000 毫秒
否則
舵機 S2 角度 100
馬達 M1 速度 80 M2 速度 80
馬達 M3 速度 80 M4 速度 80

練習三

定義函式 ultrasonic
如果 RGB超聲波讀取距離(cm) ≤ 5 那麼
停止所有馬達
舵機 S2 角度 0
暫停 1000 毫秒
停車
否則
呼叫 linefollower
呼叫巡綫功能
重複無限次
呼叫 ultrasonic
當啟動時
舵機 S2 角度 80
初始化

巡線模組

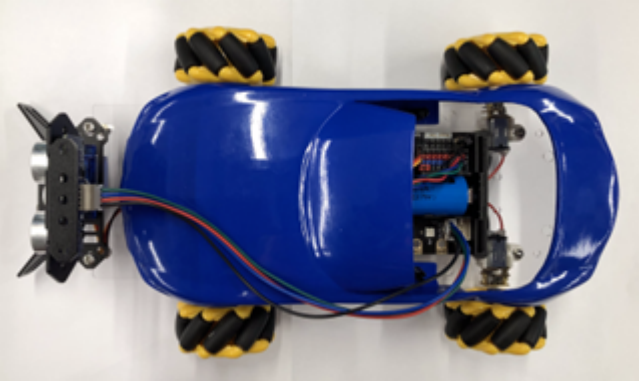
定義函式 linefollower
變數 L 設為 數位信號讀取 引腳 P13
變數 R 設為 數位信號讀取 引腳 P14
如果 L = 1 且 R = 1 那麼
馬達 M1 速度 -60 M2 速度 -60
馬達 M3 速度 -60 M4 速度 -60
倒退
否則如果 L = 1 且 R = 0 那麼
馬達 M1 速度 80 M2 速度 0
馬達 M3 速度 80 M4 速度 0
右轉
否則如果 L = 0 且 R = 1 那麼
馬達 M1 速度 0 M2 速度 80
馬達 M3 速度 0 M4 速度 80
左轉
否則
馬達 M1 速度 60 M2 速度 60
馬達 M3 速度 60 M4 速度 60
向前

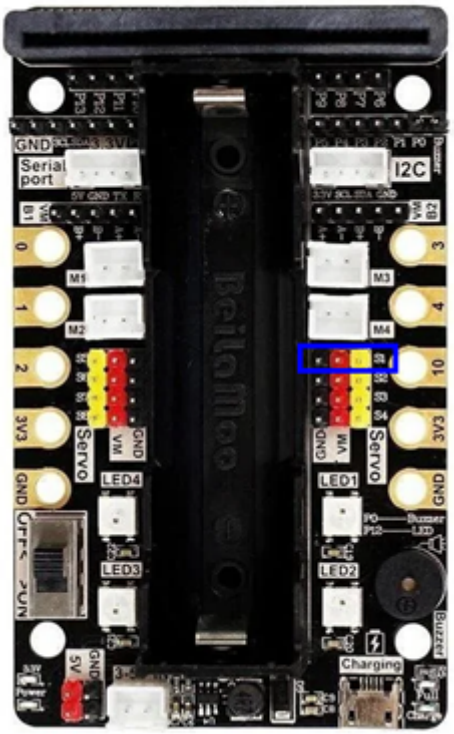
1.4.7 第七課

簡介

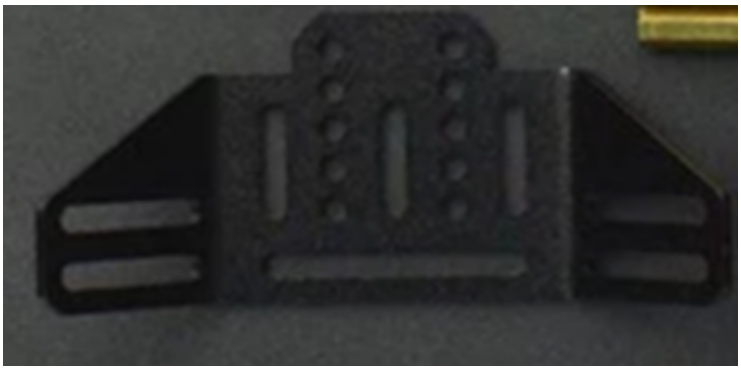
教學目標

認識 Micro:bit 擴展工具——推土機

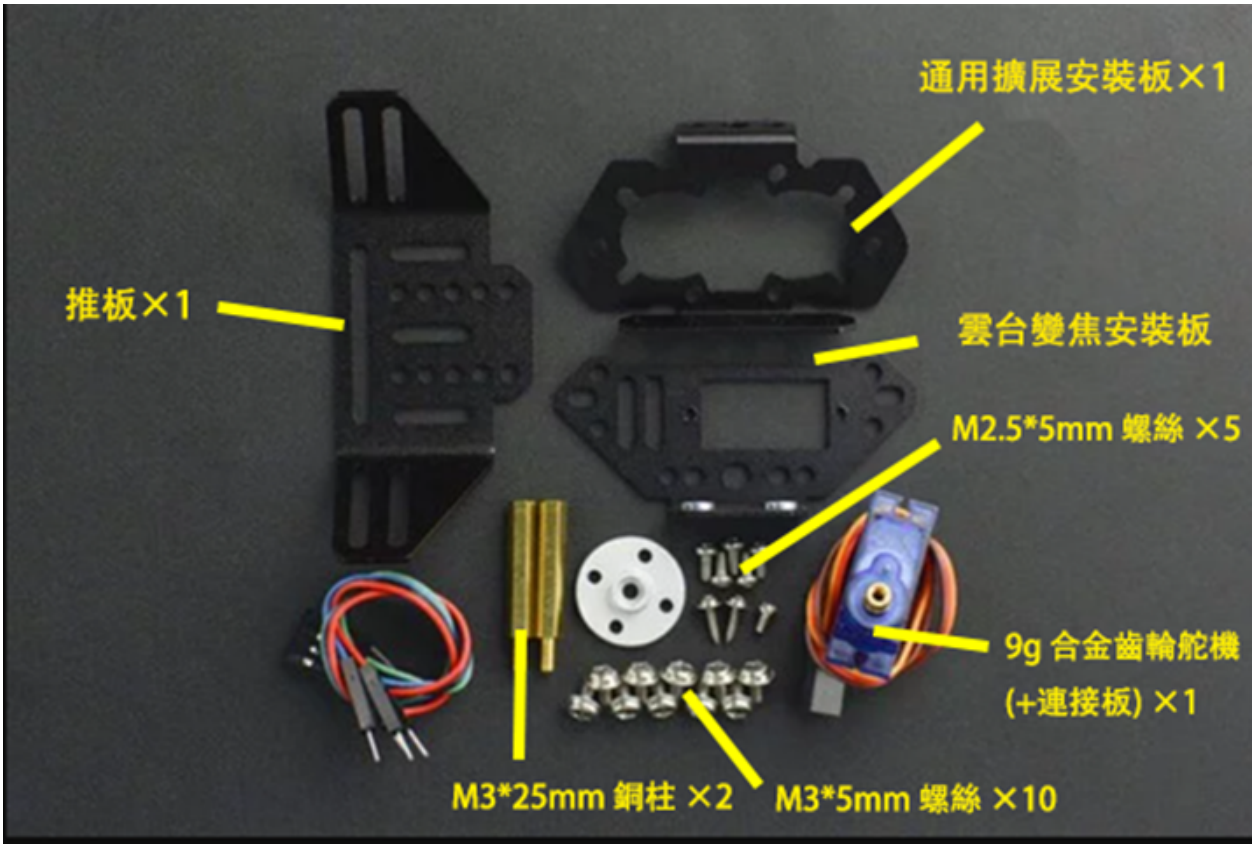




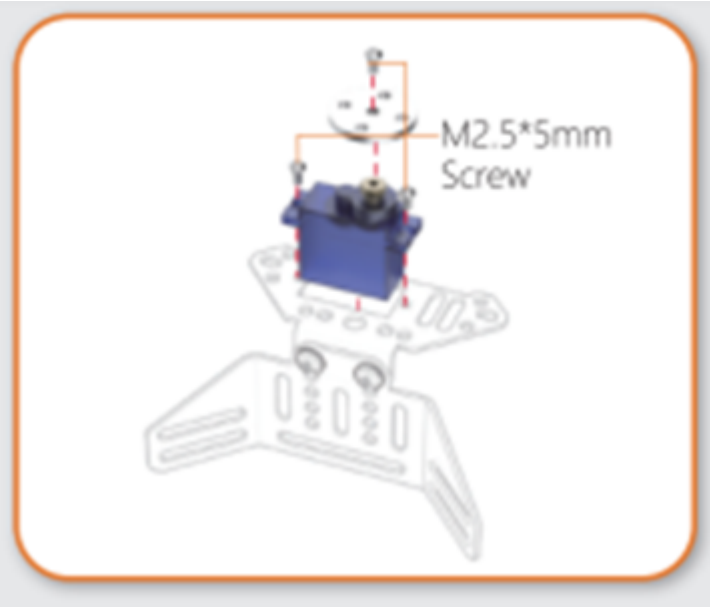
機械推土機的原理及功用



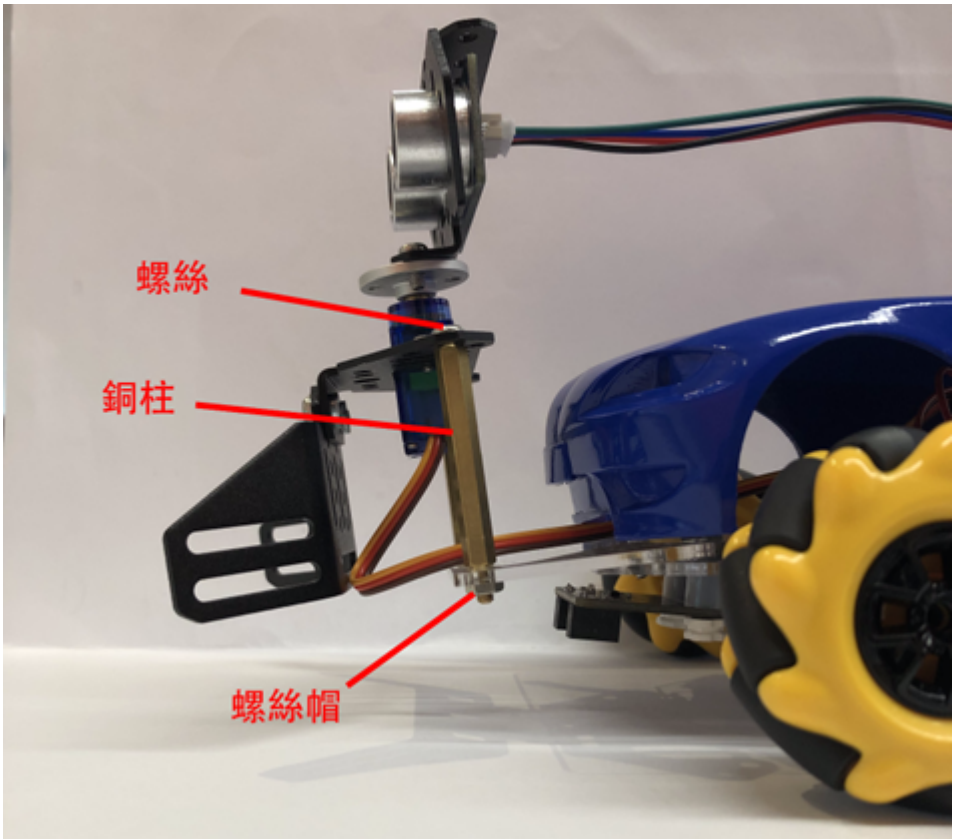
安裝推土機：

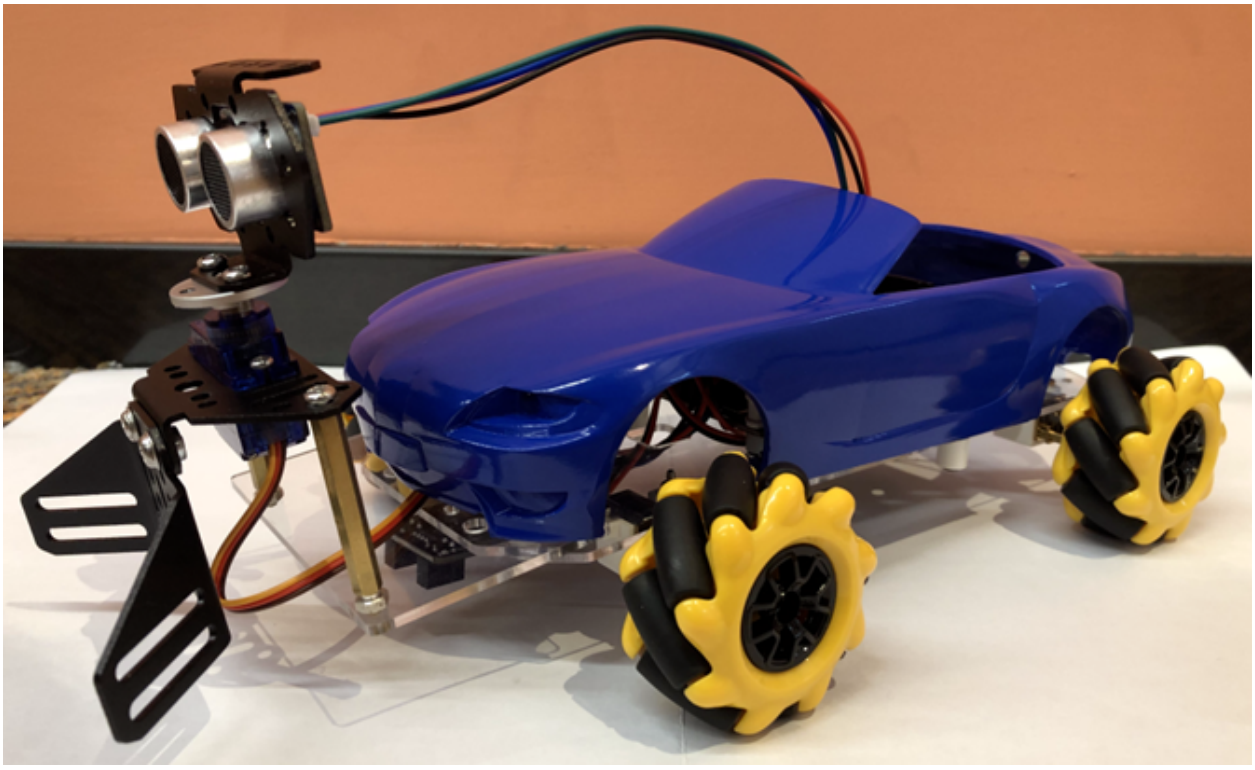

通用擴展安裝板×1
推板×1
雲台變焦安裝板
M2.5*5mm 螺絲 ×5
9g 合金齒輪舵機
(+連接板) ×1
M3*25mm 銅柱 ×2
M3*5mm 螺絲 ×10

M3*5mm Screw

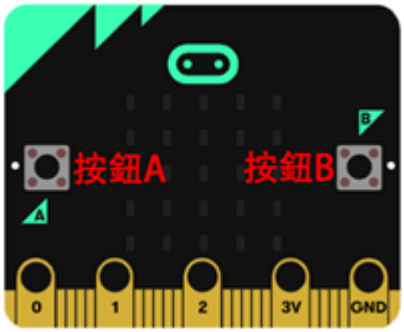

M2.5*5mm
Screw

M3*5mm
Screw


螺絲
銅柱
螺絲帽

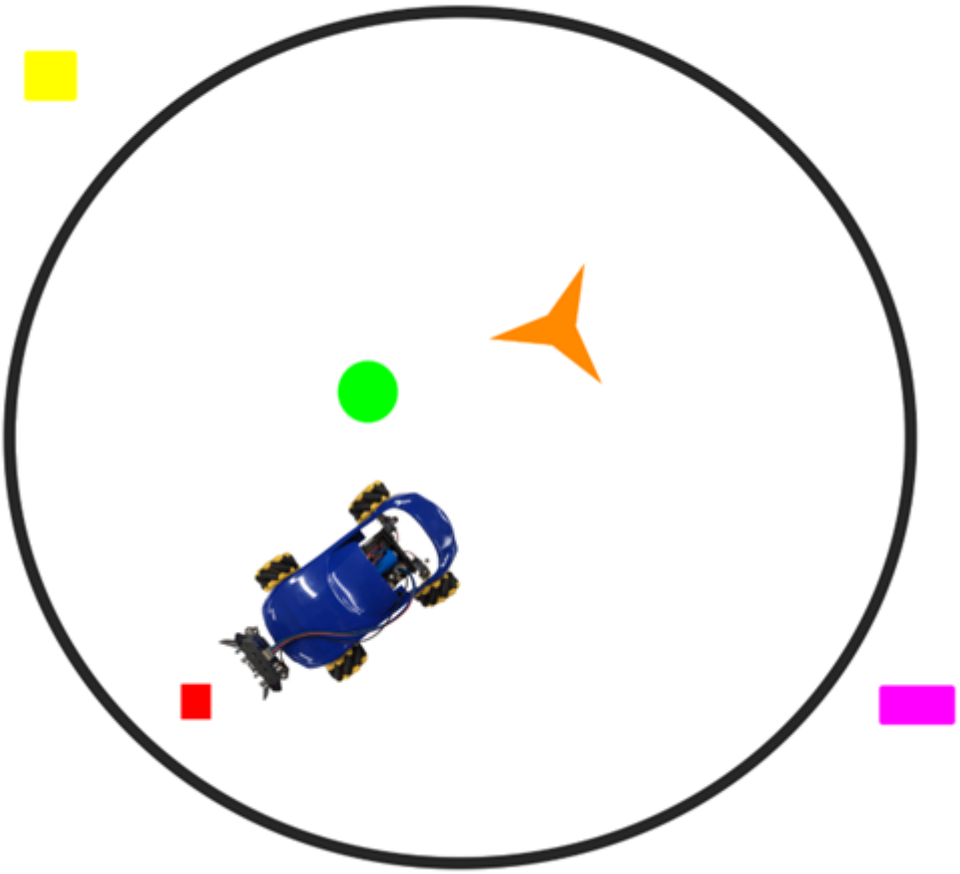


練習 1



- 先調節舵機的安裝角度，令舵機模塊角度數值 85 Ⓕ正面
- 只用舵機模塊 0-180 度無法令舵機超聲波傳感器望向正左方和正右方
- 需嘗試負數和 180 度以上的數值

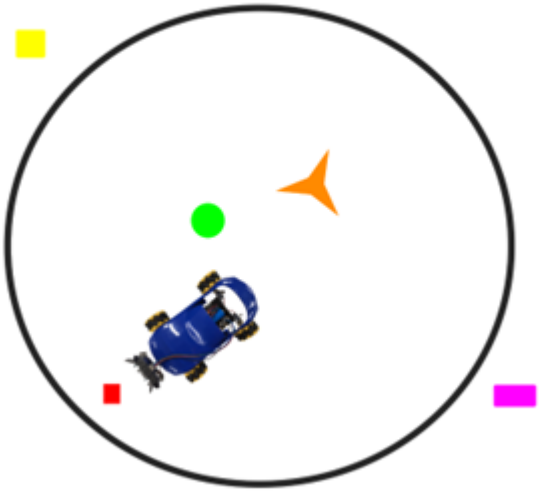
練習 2

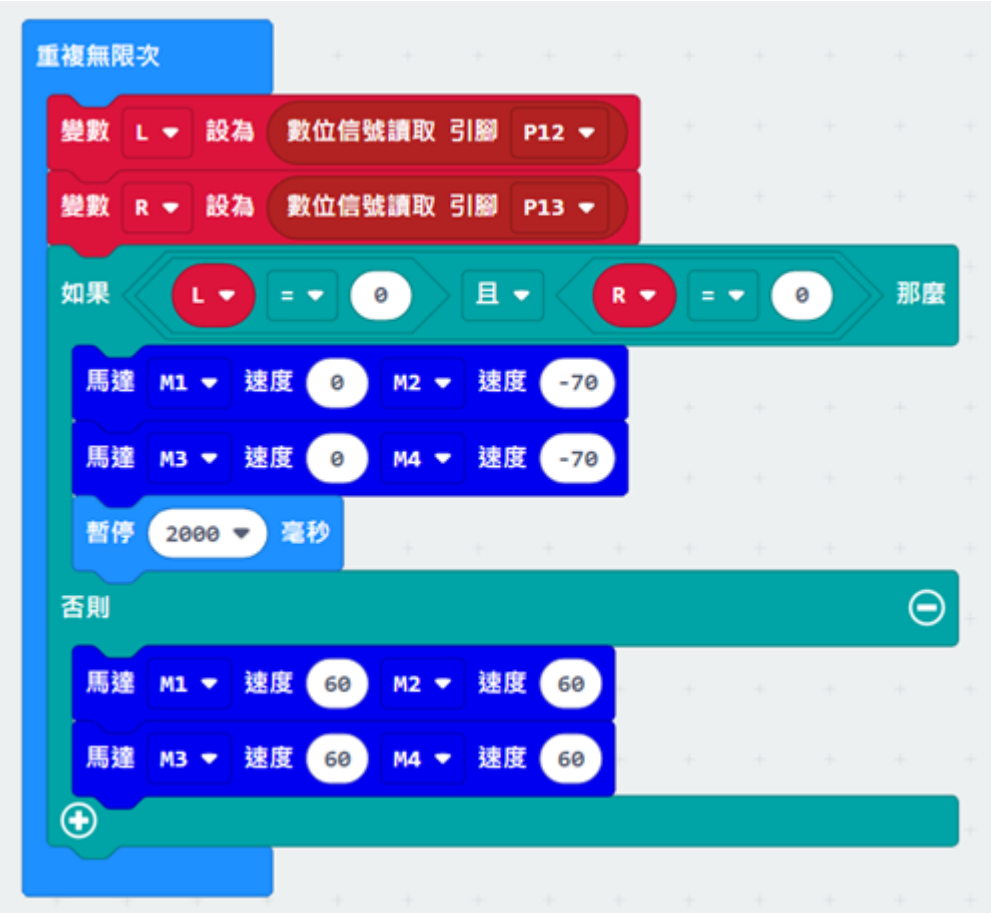


答案

練習 1

練習 2




重複無限次
變數 L 設為 數位信號讀取 引腳 P12
變數 R 設為 數位信號讀取 引腳 P13
如果 L = 0 且 R = 0 那麼
馬達 M1 速度 0 M2 速度 -70
馬達 M3 速度 0 M4 速度 -70
暫停 2000 毫秒
否則
馬達 M1 速度 60 M2 速度 60
馬達 M3 速度 60 M4 速度 60

1.4.8 第八課

簡介

教學目標

推土機工具的運用

練習一

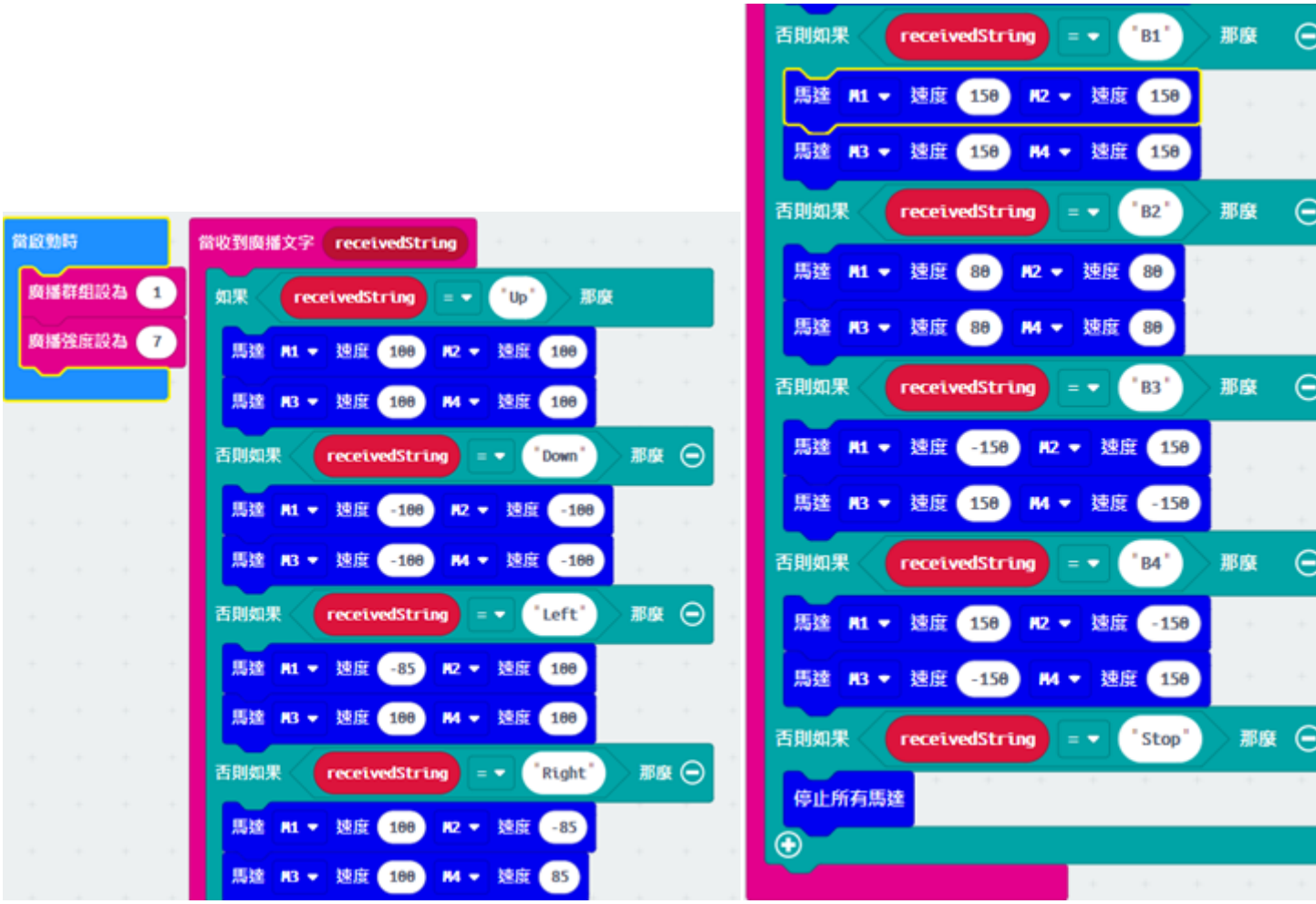
射龍門

足球比賽

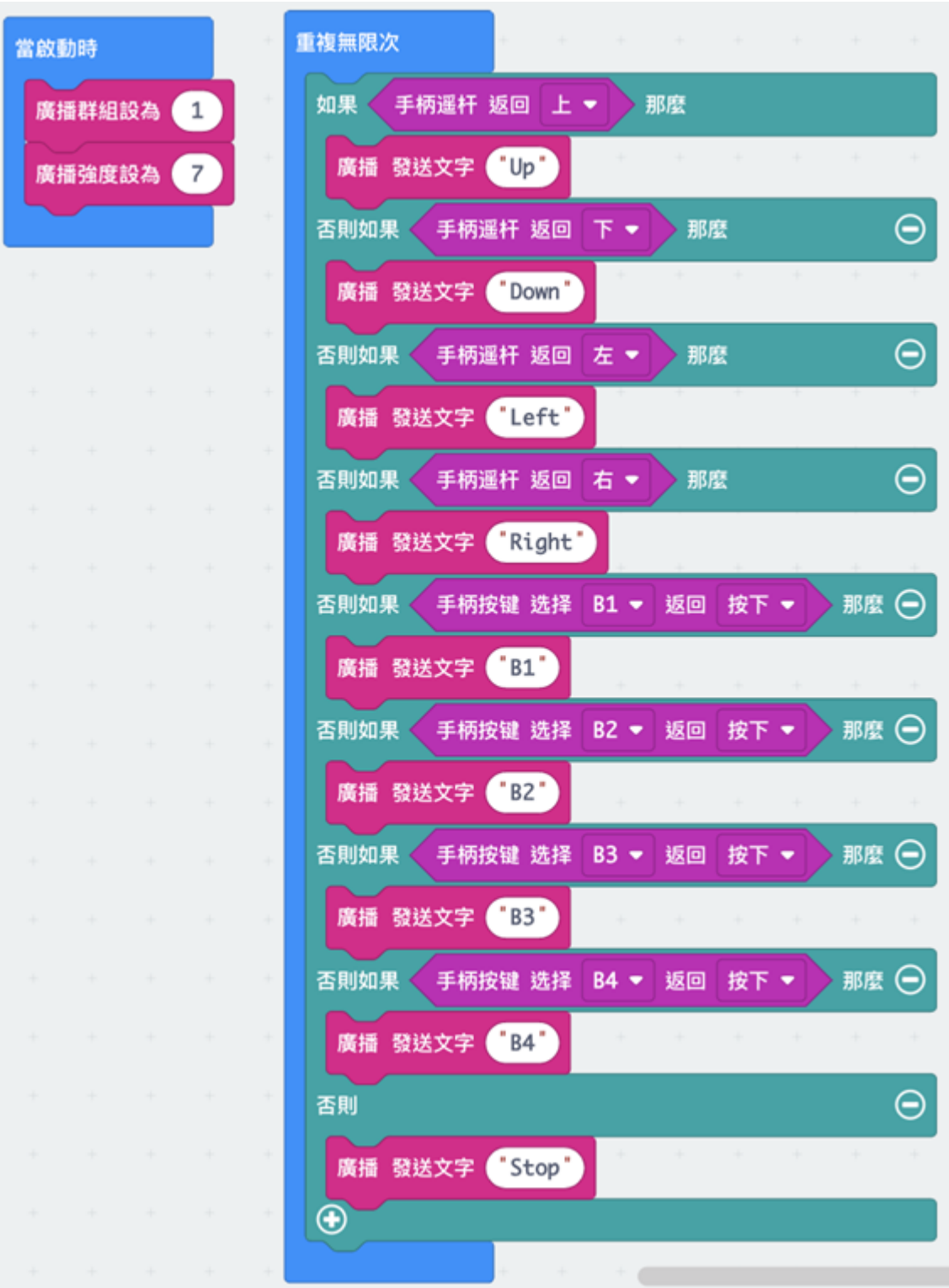
```
1.      車子或擴展工具不可超過龍門前的綫
2.      速度不可超過 150
3.      比賽途中不可以觸碰車子
```

答案

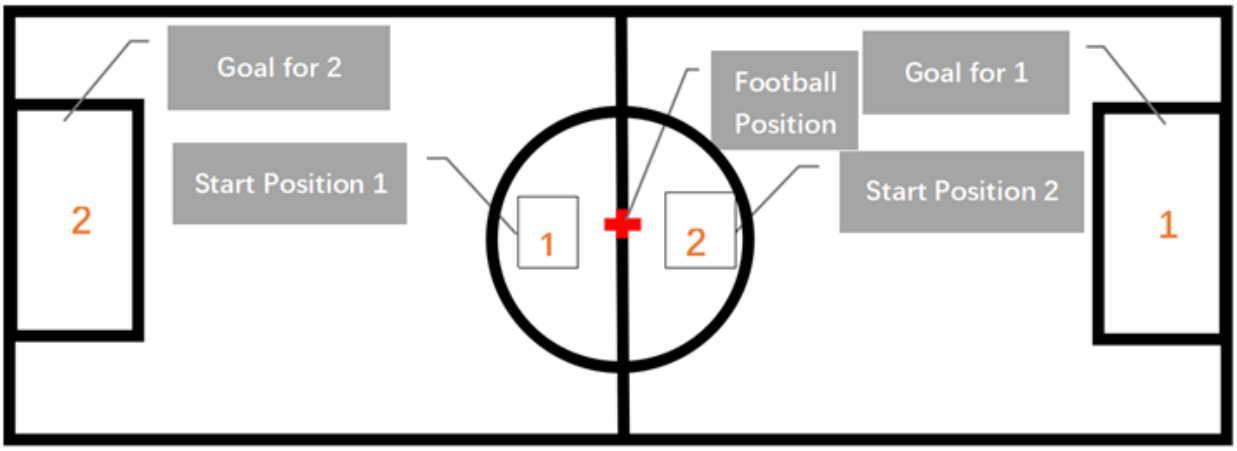
車子程式


當啟動時
廣播群組設為 1
廣播強度設為 7
當收到廣播文字 receivedString
如果 receivedString = "Up" 那麼
馬達 M1 速度 100 M2 速度 100
馬達 M3 速度 100 M4 速度 100
否則如果 receivedString = "Down" 那麼
馬達 M1 速度 -100 M2 速度 -100
馬達 M3 速度 -100 M4 速度 -100
否則如果 receivedString = "Left" 那麼
馬達 M1 速度 -85 M2 速度 100
馬達 M3 速度 100 M4 速度 100
否則如果 receivedString = "Right" 那麼
馬達 M1 速度 100 M2 速度 -85
馬達 M3 速度 100 M4 速度 85
否則如果 receivedString = "B1" 那麼
馬達 M1 速度 150 M2 速度 150
馬達 M3 速度 150 M4 速度 150
否則如果 receivedString = "B2" 那麼
馬達 M1 速度 80 M2 速度 80
馬達 M3 速度 80 M4 速度 80
否則如果 receivedString = "B3" 那麼
馬達 M1 速度 -150 M2 速度 150
馬達 M3 速度 150 M4 速度 -150
否則如果 receivedString = "B4" 那麼
馬達 M1 速度 150 M2 速度 -150
馬達 M3 速度 -150 M4 速度 150
否則如果 receivedString = "Stop" 那麼
停止所有馬達

Ⓕ控器程式


當啟動時
廣播群組設為 1
廣播強度設為 7
重複無限次
如果 手柄遥杆 返回 上 那麼
廣播 發送文字 "Up"
否則如果 手柄遥杆 返回 下 那麼
廣播 發送文字 "Down"
否則如果 手柄遥杆 返回 左 那麼
廣播 發送文字 "Left"
否則如果 手柄遥杆 返回 右 那麼
廣播 發送文字 "Right"
否則如果 手柄按键 选择 B1 返回 按下 那麼
廣播 發送文字 "B1"
否則如果 手柄按键 选择 B2 返回 按下 那麼
廣播 發送文字 "B2"
否則如果 手柄按键 选择 B3 返回 按下 那麼
廣播 發送文字 "B3"
否則如果 手柄按键 选择 B4 返回 按下 那麼
廣播 發送文字 "B4"
否則
廣播 發送文字 "Stop"

附Ⓕ: 足球場地



1.4.9 第九課

簡介

教學目標

可升降夾子

安裝升降支架:

練習 1

練習 2

- 夾子加上支架長度增加，Ⓕ置場地時預留足Ⓕ距離。
- 操作每一個工具時留意是否需要暫停。
- 建立變數“destination”記Ⓕ目的地，以目的地確定車子出界時的動作

多功能工程車（夾子和叉子）

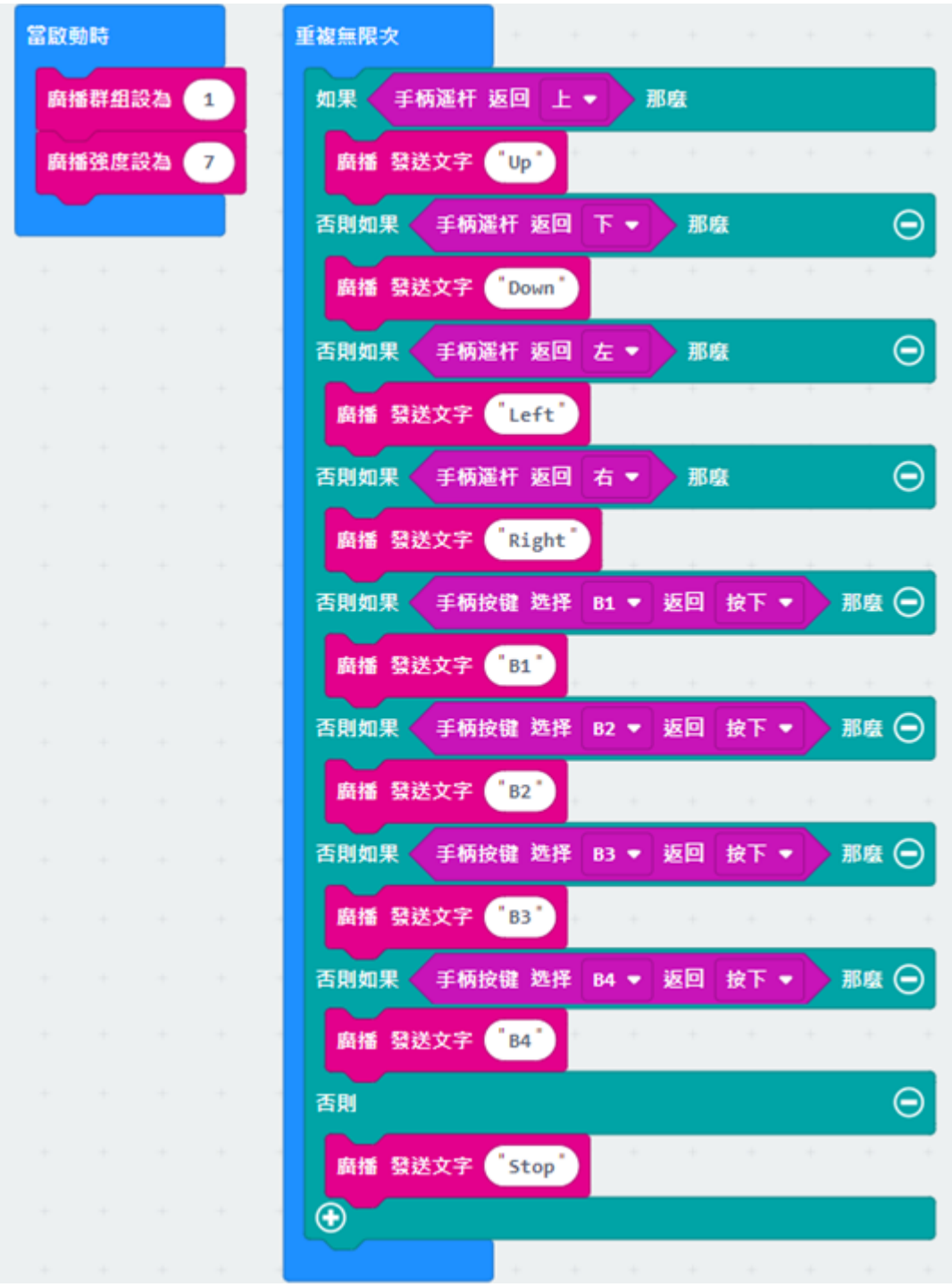
練習 3

- 叉車在車子前面，夾子在車子後面
- 直接改良練習 1 程序
- 建立變數記Ⓕ兩種工具使用狀態，以此狀態Ⓕ定工具的下個動作

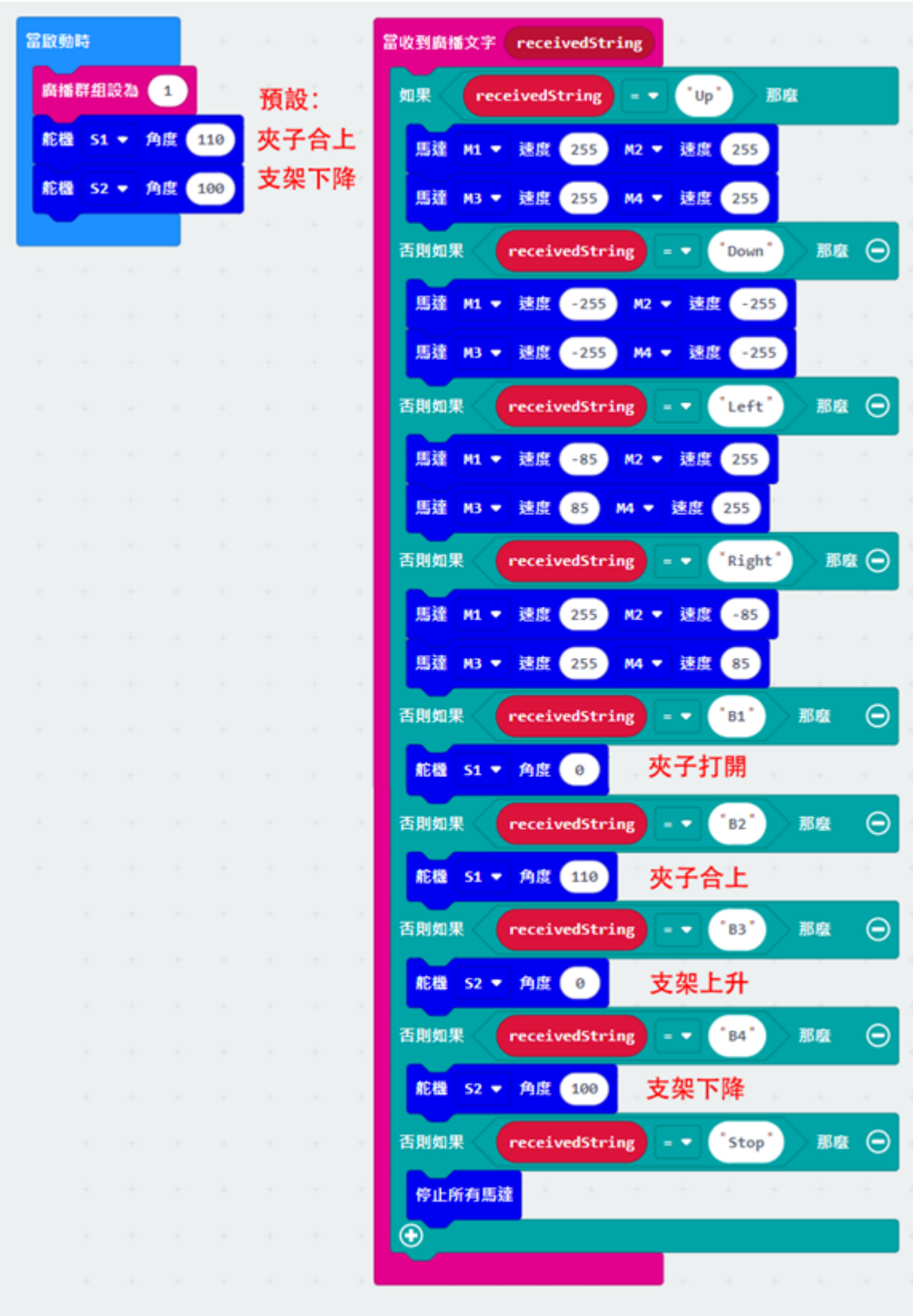
答案

練習 1

Ⓕ控器的程序


當啟動時
廣播群組設為 1
廣播強度設為 7
重複無限次
如果 手柄遥杆 返回 上 那麼
廣播 發送文字 "Up"
否則如果 手柄遥杆 返回 下 那麼
廣播 發送文字 "Down"
否則如果 手柄遥杆 返回 左 那麼
廣播 發送文字 "Left"
否則如果 手柄遥杆 返回 右 那麼
廣播 發送文字 "Right"
否則如果 手柄按键 选择 B1 返回 按下 那麼
廣播 發送文字 "B1"
否則如果 手柄按键 选择 B2 返回 按下 那麼
廣播 發送文字 "B2"
否則如果 手柄按键 选择 B3 返回 按下 那麼
廣播 發送文字 "B3"
否則如果 手柄按键 选择 B4 返回 按下 那麼
廣播 發送文字 "B4"
否則
廣播 發送文字 "Stop"

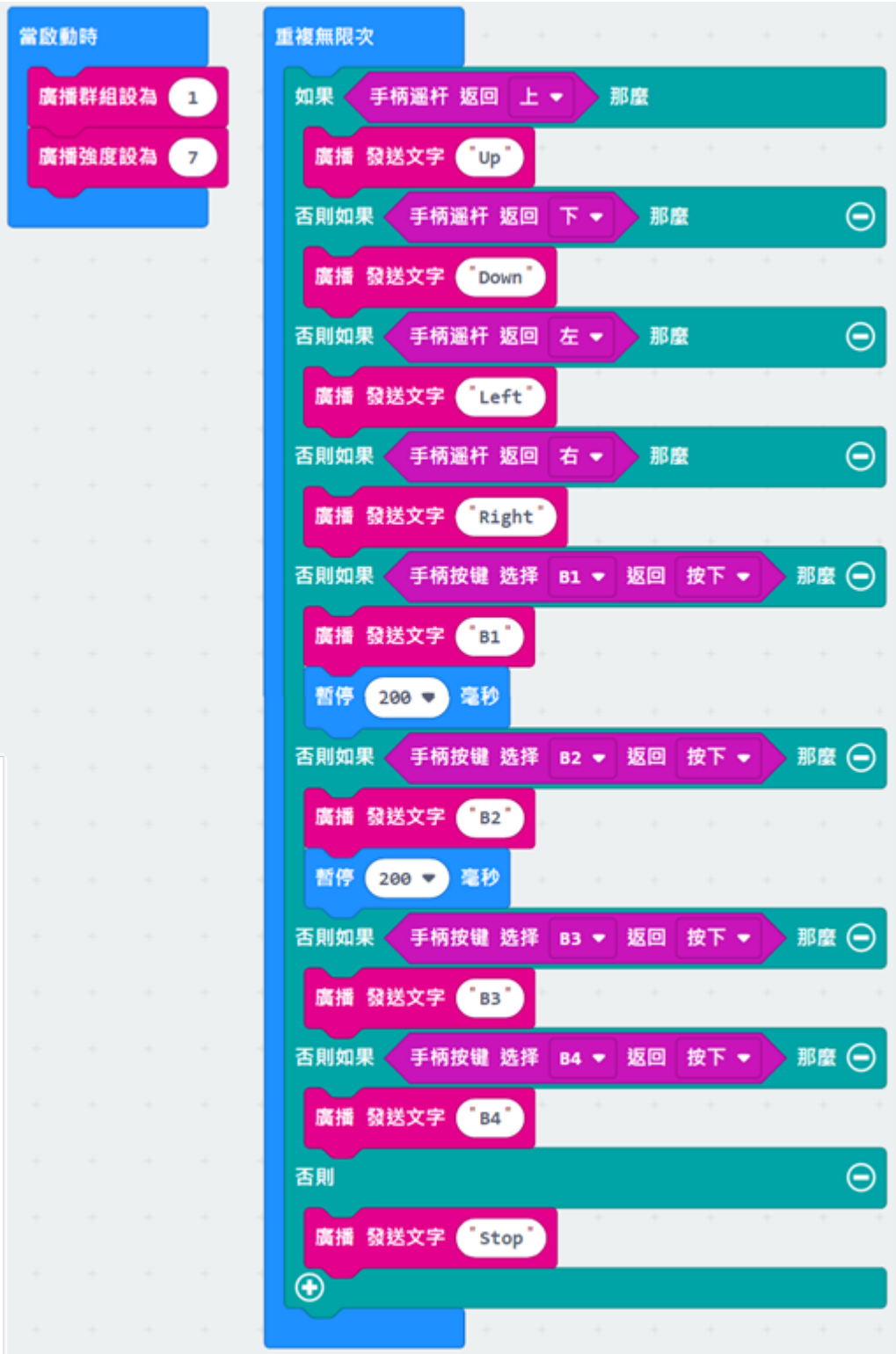
車子的程序


當啟動時
廣播群組設為 1
舵機 S1 角度 110
舵機 S2 角度 100
預設:
夾子合上
支架下降
當收到廣播文字 receivedString
如果 receivedString = "Up" 那麼
馬達 M1 速度 255 M2 速度 255
馬達 M3 速度 255 M4 速度 255
否則如果 receivedString = "Down" 那麼
馬達 M1 速度 -255 M2 速度 -255
馬達 M3 速度 -255 M4 速度 -255
否則如果 receivedString = "Left" 那麼
馬達 M1 速度 -85 M2 速度 255
馬達 M3 速度 85 M4 速度 255
否則如果 receivedString = "Right" 那麼
馬達 M1 速度 255 M2 速度 -85
馬達 M3 速度 255 M4 速度 85
否則如果 receivedString = "B1" 那麼
舵機 S1 角度 0
夾子打開
否則如果 receivedString = "B2" 那麼
舵機 S1 角度 110
夾子合上
否則如果 receivedString = "B3" 那麼
舵機 S2 角度 0
支架上升
否則如果 receivedString = "B4" 那麼
舵機 S2 角度 100
支架下降
否則如果 receivedString = "Stop" 那麼
停止所有馬達

練習 2

當啟動時
Servo S1 degree 0
Servo S2 degree 100
變數 destination 設為 "end"
預設:
夾子打開
支架降下
目標地: 終點
重複無限次
變數 L 設為 數位信號讀取 引腳 P13
變數 R 設為 數位信號讀取 引腳 P14
呼叫 LineFollow
定義函式 LineFollow
如果 L = 0 且 R = 0 那麼
Motor M1 speed 60 M2 speed 60
Motor M3 speed 60 M4 speed 60
否則如果 L = 1 且 R = 0 那麼
Motor M1 speed 80 M2 speed 0
Motor M3 speed 80 M4 speed 0
否則如果 L = 0 且 R = 1 那麼
Motor M1 speed 0 M2 speed 80
Motor M4 speed 0 M3 speed 80
否則如果 L = 1 且 R = 1 那麼
當目標地為終點時的程序
如果 destination = "end" 那麼
Motor Stop All
Servo S1 degree 110
暫停 500 毫秒
Servo S2 degree 0
Motor M1 speed 80 M2 speed -80
Motor M4 speed 80 M3 speed -80
暫停 1500 毫秒
變數 destination 設為 "start"
車子停下
夾子夾住物件
支架上升
180° 右轉
設目的地爲起點
當目標地為起點時的程序
否則如果 destination = "start" 那麼
Motor Stop All
Servo S1 degree 0
車子停下
放開物品

練習 3


當啟動時
廣播群組設為 1
廣播強度設為 7
重複無限次
如果 手柄遥杆 返回 上 那麼
廣播 發送文字 "Up"
否則如果 手柄遥杆 返回 下 那麼
廣播 發送文字 "Down"
否則如果 手柄遥杆 返回 左 那麼
廣播 發送文字 "Left"
否則如果 手柄遥杆 返回 右 那麼
廣播 發送文字 "Right"
否則如果 手柄按键 选择 B1 返回 按下 那麼
廣播 發送文字 "B1"
暫停 200 毫秒
否則如果 手柄按键 选择 B2 返回 按下 那麼
廣播 發送文字 "B2"
暫停 200 毫秒
否則如果 手柄按键 选择 B3 返回 按下 那麼
廣播 發送文字 "B3"
否則如果 手柄按键 选择 B4 返回 按下 那麼
廣播 發送文字 "B4"
否則
廣播 發送文字 "Stop"

車子的程序

當啟動時
廣播群組設為 1
舵機 S1 角度 110
舵機 S2 角度 100
變數 beetle_state 設為 "close"
變數 forklift_state 設為 "down"
預設:
夾子合上
支架降下
夾子狀態：合上
支架狀態：降下
夾子的程序
定義函式 beetle_control
如果 beetle_state = "close" 那麼
舵機 S1 角度 0
夾子打開
變數 beetle_state 設為 "open"
狀態：打開
否則如果 beetle_state = "open" 那麼
舵機 S1 角度 110
夾子合上
變數 beetle_state 設為 "close"
狀態：合上
支架的程序
定義函式 forklift_control
如果 forklift_state = "down" 那麼
舵機 S2 角度 0
支架上升
變數 forklift_state 設為 "up"
狀態：升起
否則如果 forklift_state = "up" 那麼
舵機 S2 角度 100
支架下降
變數 forklift_state 設為 "down"
狀態：降下
當收到廣播文字 receivedString
如果 receivedString = "Up" 那麼
馬達 M1 速度 255 M2 速度 255
馬達 M3 速度 255 M4 速度 255
否則如果 receivedString = "Down" 那麼
馬達 M1 速度 -255 M2 速度 -255
馬達 M3 速度 -255 M4 速度 -255
否則如果 receivedString = "Left" 那麼
馬達 M1 速度 -85 M2 速度 255
馬達 M3 速度 85 M4 速度 255
否則如果 receivedString = "Right" 那麼
馬達 M1 速度 255 M2 速度 -85
馬達 M3 速度 255 M4 速度 85
否則如果 receivedString = "B1" 那麼
呼叫 beetle_control
否則如果 receivedString = "B2" 那麼
呼叫 forklift_control
否則如果 receivedString = "B3" 那麼
馬達 M1 速度 80 M2 速度 -80
馬達 M3 速度 80 M4 速度 -80
向右自轉
否則如果 receivedString = "B4" 那麼
馬達 M1 速度 -80 M2 速度 80
馬達 M3 速度 -80 M4 速度 80
向左自轉
否則如果 receivedString = "Stop" 那麼
停止所有馬達

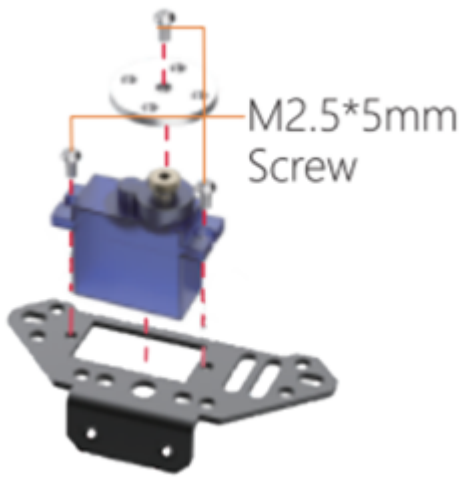
1.4.10 第十課

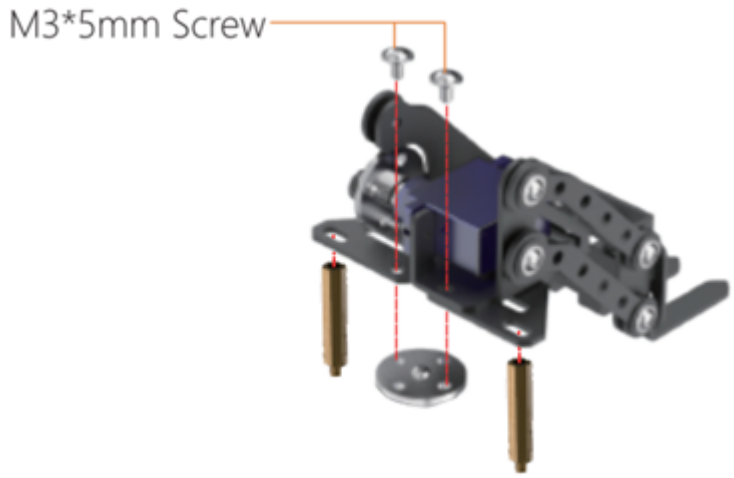
簡介

教學目標

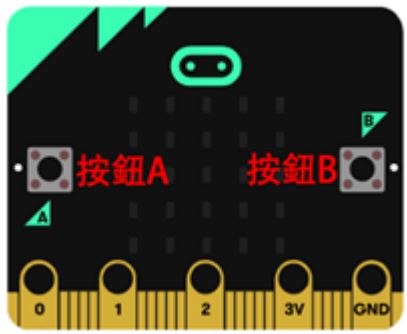
雙向可動式叉子

安裝步驟





練習 1

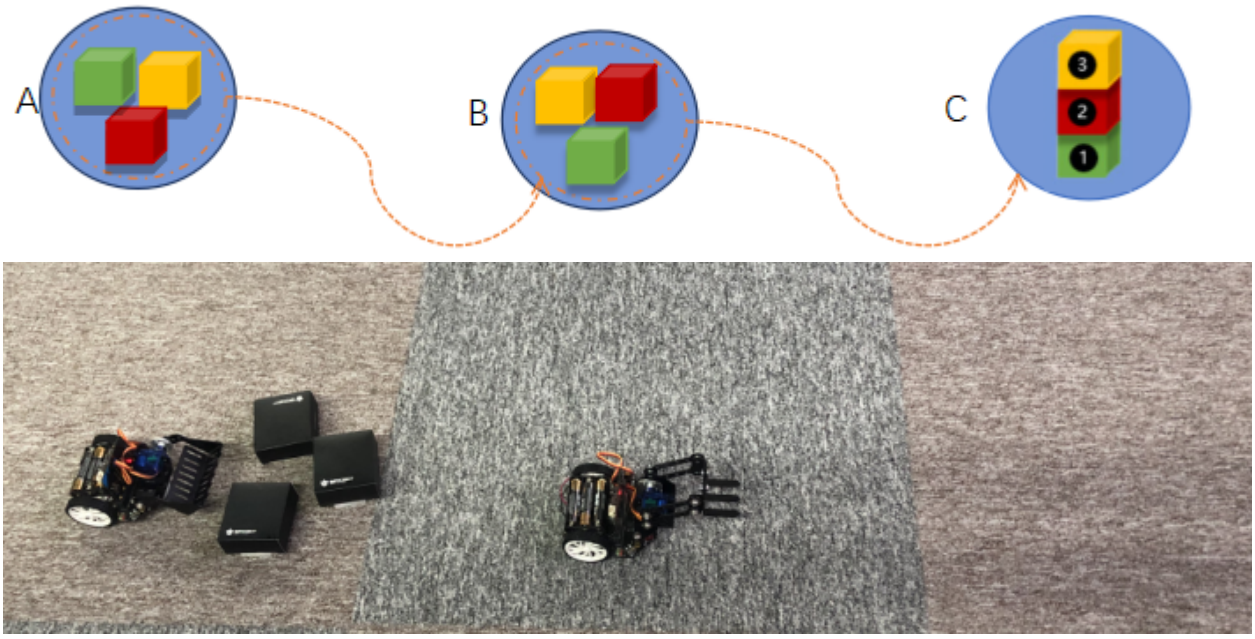


任務

車子接力賽

介紹

游戲規則

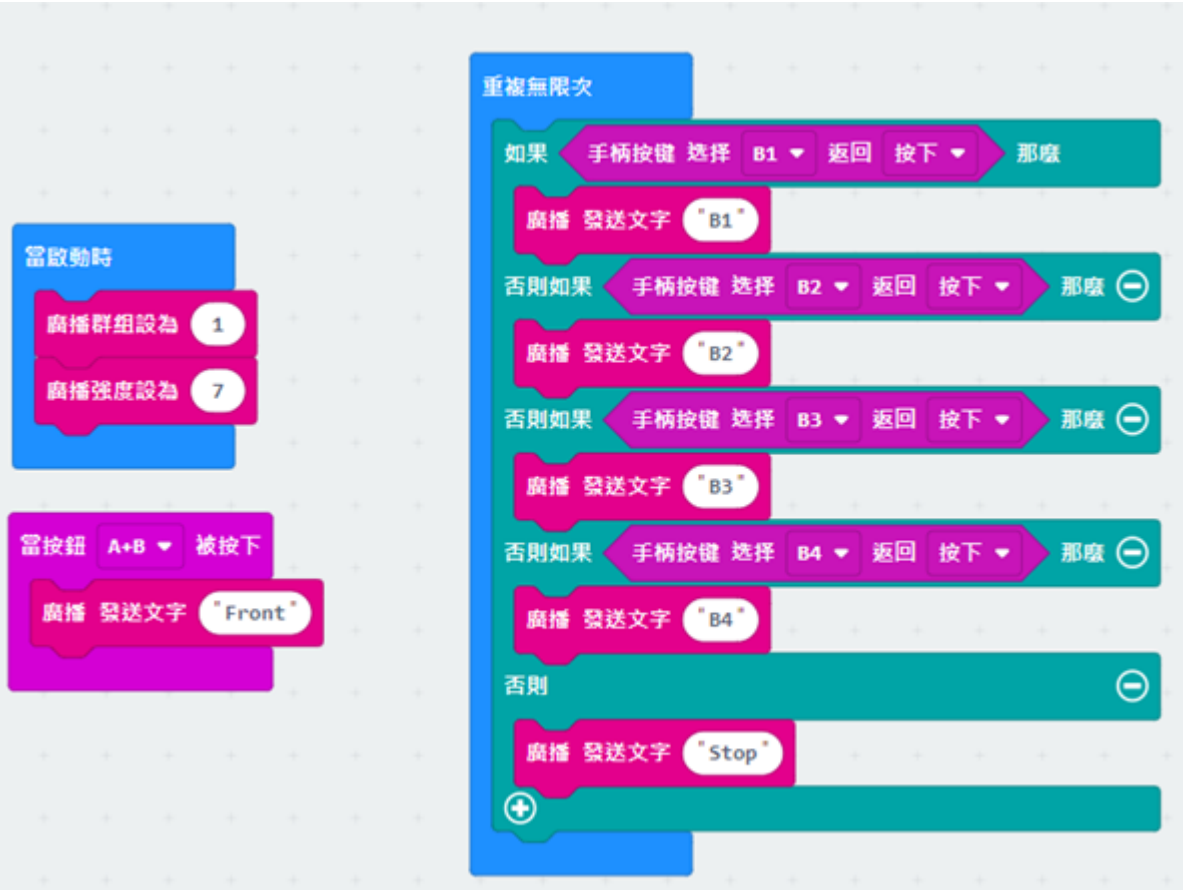


答案

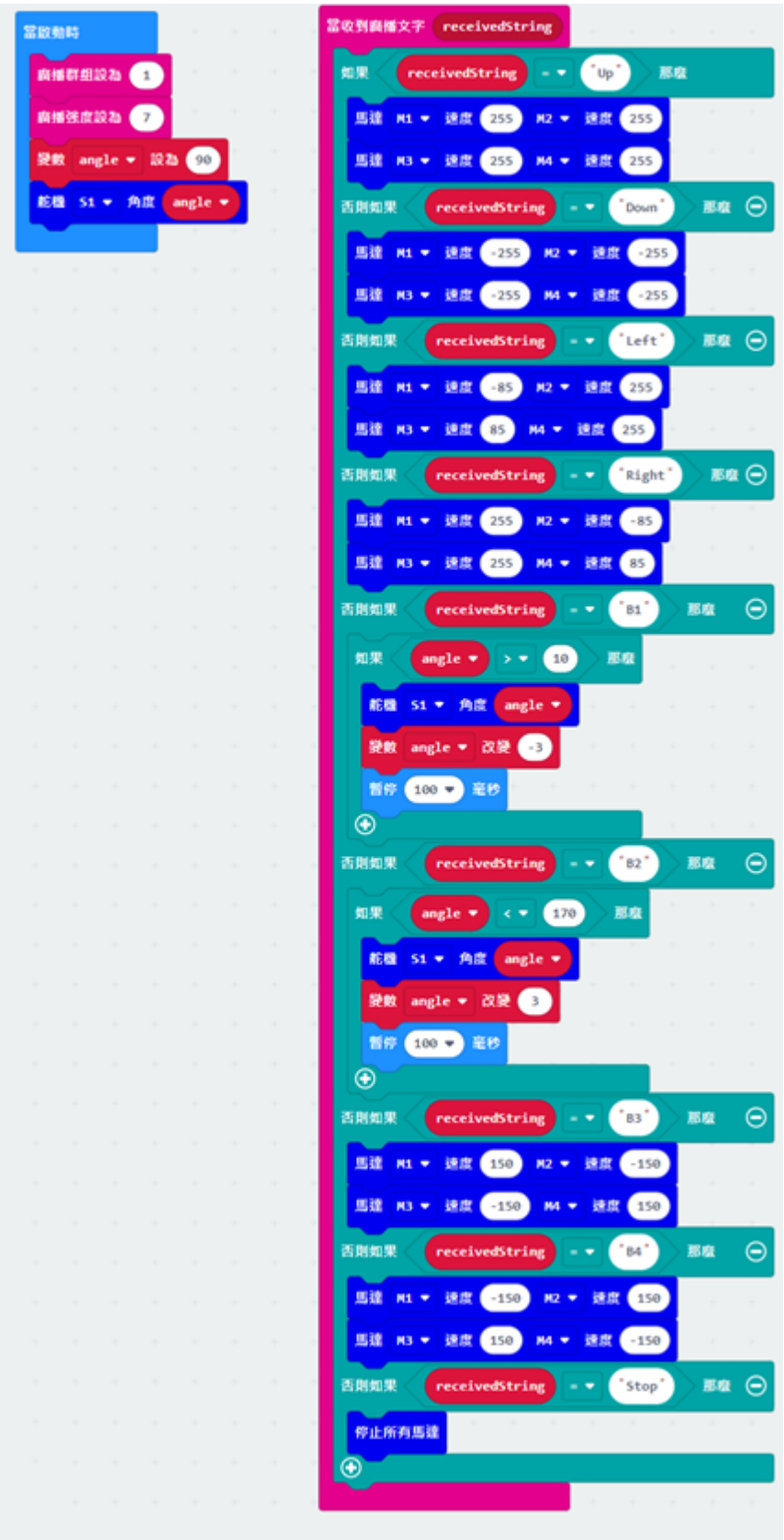
練習 1

車子的程序（叉子水平移動部分）

Ⓕ控器的程序

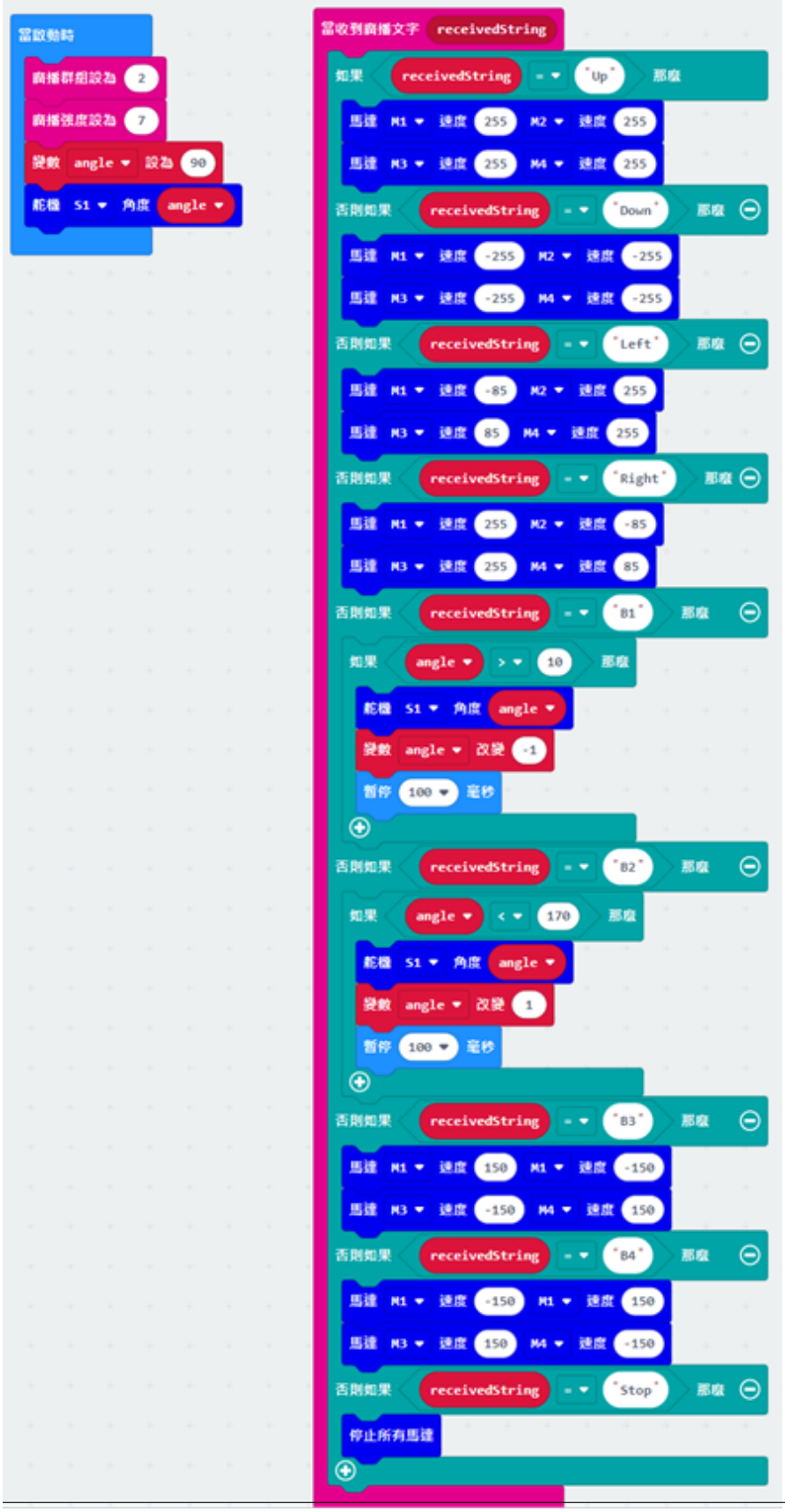

當啟動時
廣播群組設為 1
廣播強度設為 7
當按鈕 A+B 被按下
廣播 發送文字 "Front"
重複無限次
如果 手柄按键 选择 B1 返回 按下 那麼
廣播 發送文字 "B1"
否則如果 手柄按键 选择 B2 返回 按下 那麼
廣播 發送文字 "B2"
否則如果 手柄按键 选择 B3 返回 按下 那麼
廣播 發送文字 "B3"
否則如果 手柄按键 选择 B4 返回 按下 那麼
廣播 發送文字 "B4"
否則
廣播 發送文字 "Stop"

車子接力賽

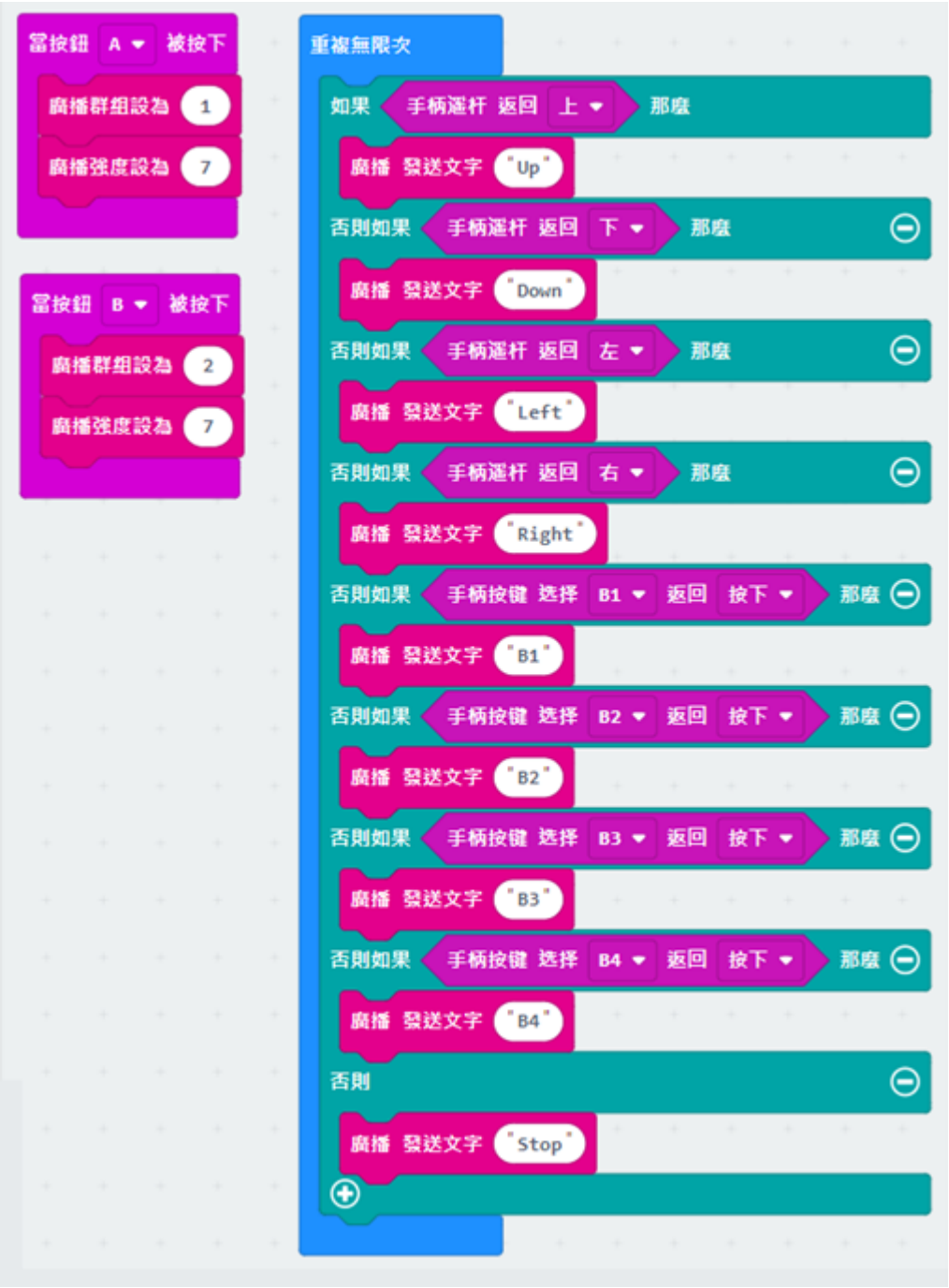

當啟動時
廣播群組設為 1
廣播強度設為 7
變數 angle 設為 90
舵機 S1 角度 angle
當收到廣播文字 receivedString
如果 receivedString = "Up" 那麼
馬達 M1 速度 255 M2 速度 255
馬達 M3 速度 255 M4 速度 255
否則如果 receivedString = "Down" 那麼
馬達 M1 速度 -255 M2 速度 -255
馬達 M3 速度 -255 M4 速度 -255
否則如果 receivedString = "Left" 那麼
馬達 M1 速度 -85 M2 速度 255
馬達 M3 速度 85 M4 速度 255
否則如果 receivedString = "Right" 那麼
馬達 M1 速度 255 M2 速度 -85
馬達 M3 速度 255 M4 速度 85
否則如果 receivedString = "B1" 那麼
如果 angle > 10 那麼
舵機 S1 角度 angle
變數 angle 改變 -3
暫停 100 毫秒
否則如果 receivedString = "B2" 那麼
如果 angle < 170 那麼
舵機 S1 角度 angle
變數 angle 改變 3
暫停 100 毫秒
否則如果 receivedString = "B3" 那麼
馬達 M1 速度 150 M2 速度 -150
馬達 M3 速度 -150 M4 速度 150
否則如果 receivedString = "B4" 那麼
馬達 M1 速度 -150 M2 速度 150
馬達 M3 速度 150 M4 速度 -150
否則如果 receivedString = "Stop" 那麼
停止所有馬達

鏟車

叉車


當啟動時
廣播群組設為 2
廣播強度設為 7
變數 angle 設為 90
舵機 S1 角度 angle
當收到廣播文字 receivedString
如果 receivedString = "Up" 那麼
馬達 M1 速度 255 M2 速度 255
馬達 M3 速度 255 M4 速度 255
否則如果 receivedString = "Down" 那麼
馬達 M1 速度 -255 M2 速度 -255
馬達 M3 速度 -255 M4 速度 -255
否則如果 receivedString = "Left" 那麼
馬達 M1 速度 -85 M2 速度 255
馬達 M3 速度 85 M4 速度 255
否則如果 receivedString = "Right" 那麼
馬達 M1 速度 255 M2 速度 -85
馬達 M3 速度 255 M4 速度 85
否則如果 receivedString = "B1" 那麼
如果 angle > 10 那麼
舵機 S1 角度 angle
變數 angle 改變 -1
暫停 100 毫秒
否則如果 receivedString = "B2" 那麼
如果 angle < 170 那麼
舵機 S1 角度 angle
變數 angle 改變 1
暫停 100 毫秒
否則如果 receivedString = "B3" 那麼
馬達 M1 速度 150 M1 速度 -150
馬達 M3 速度 -150 M4 速度 150
否則如果 receivedString = "B4" 那麼
馬達 M1 速度 -150 M1 速度 150
馬達 M3 速度 150 M4 速度 -150
否則如果 receivedString = "Stop" 那麼
停止所有馬達

Ⓕ控器


當按鈕 A 被按下
廣播群組設為 1
廣播強度設為 7
當按鈕 B 被按下
廣播群組設為 2
廣播強度設為 7
重複無限次
如果 手柄遥杆 返回 上 那麼
廣播 發送文字 "Up"
否則如果 手柄遥杆 返回 下 那麼
廣播 發送文字 "Down"
否則如果 手柄遥杆 返回 左 那麼
廣播 發送文字 "Left"
否則如果 手柄遥杆 返回 右 那麼
廣播 發送文字 "Right"
否則如果 手柄按键 选择 B1 返回 按下 那麼
廣播 發送文字 "B1"
否則如果 手柄按键 选择 B2 返回 按下 那麼
廣播 發送文字 "B2"
否則如果 手柄按键 选择 B3 返回 按下 那麼
廣播 發送文字 "B3"
否則如果 手柄按键 选择 B4 返回 按下 那麼
廣播 發送文字 "B4"
否則
廣播 發送文字 "Stop"

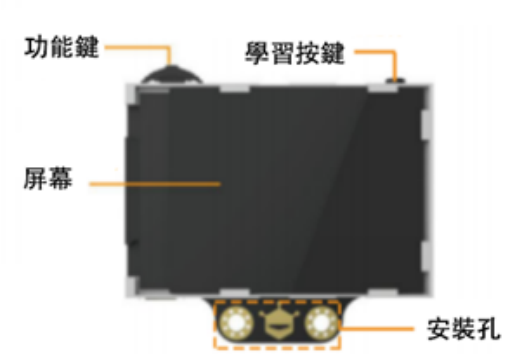
1.5 Micro:bit M1 智能小車 _ 高級

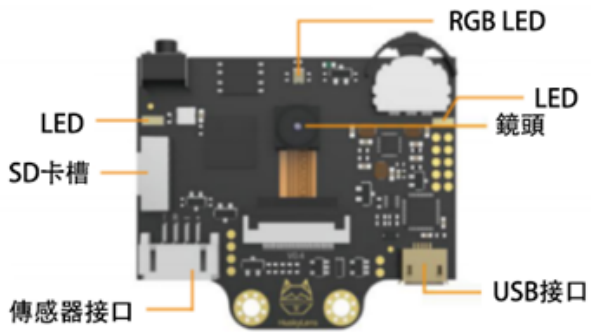
1.5.1 第一課

簡介

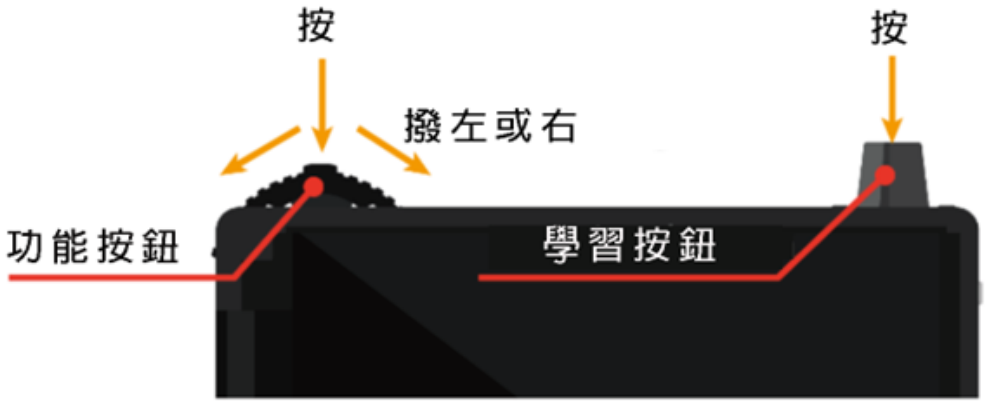
教學目標

認識 HuskyLens





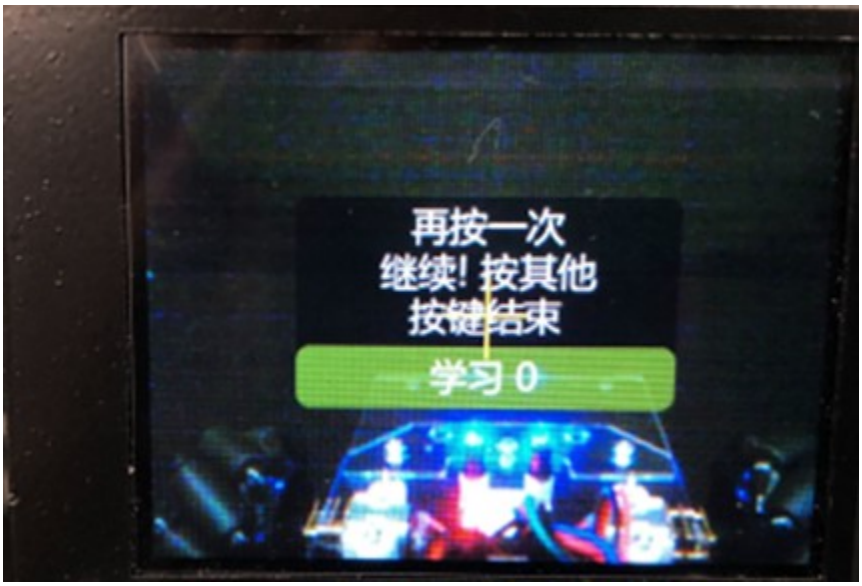
使用方法



- 向左或右撥功能按鈕能切F功能/模式
- 單擊功能按鈕能叫出功能菜單
- 長按功能按鈕後，您可以設置是否學習多個目標：
- 再按一次功能按鈕，可以拖移F動條至右啓動「學習多個」

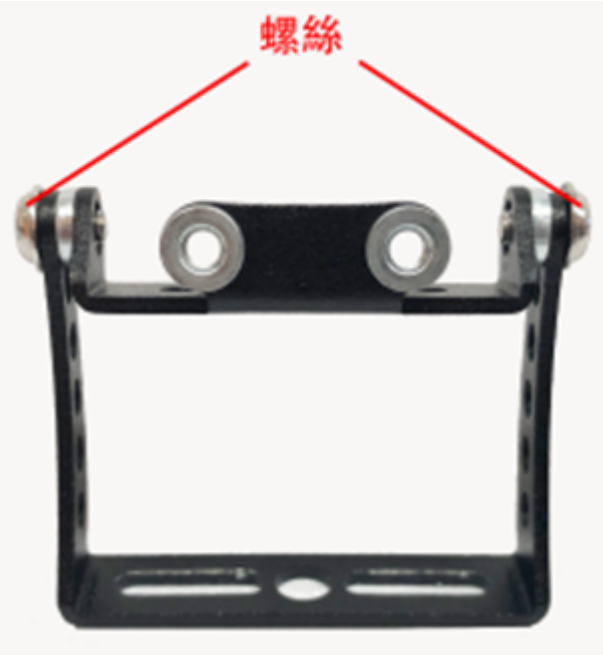
- 單擊學習按鈕能學習新的物件，物件的 ID 會顯示在螢幕上。
- 長按學習按鈕能從不同角度和距離學習識F新的物件。

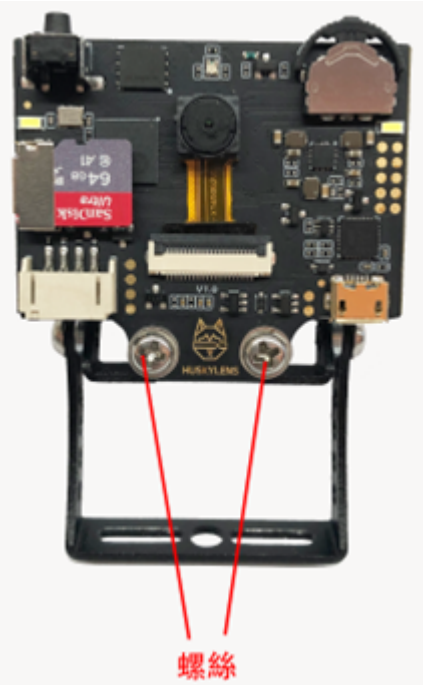
- 在每次學習目標後，需要在 4 秒內再次按學習按鈕一下以繼續學習（否則 4 秒後停止學習，必須先遺忘目標後再重新學習）

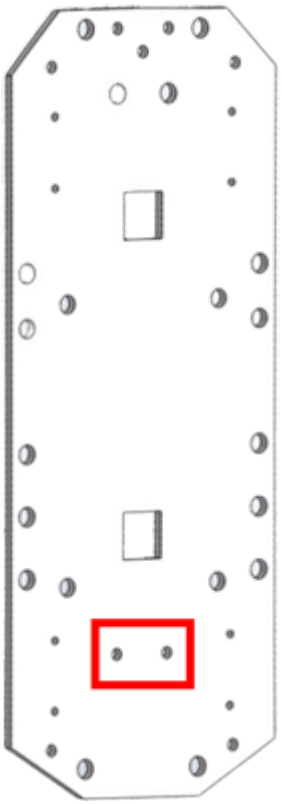


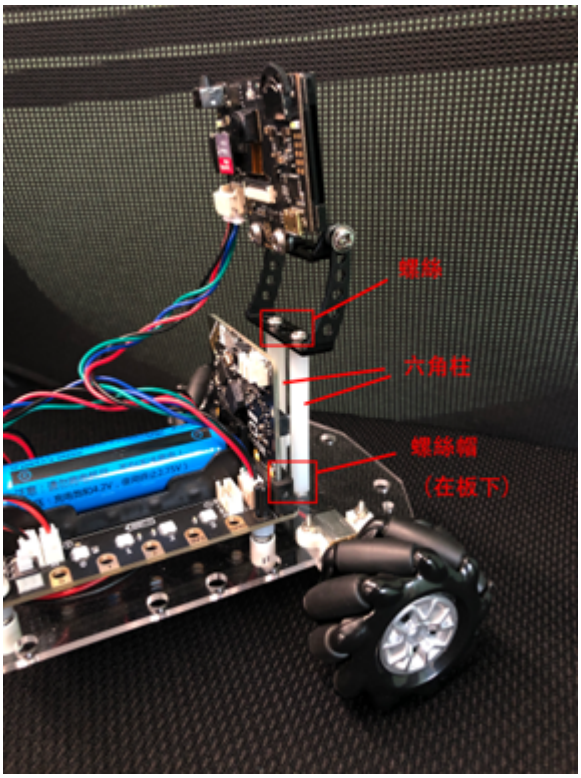
- 在停止學習後，再按兩次學習按鈕能Ⓕ遺忘已學習的物件

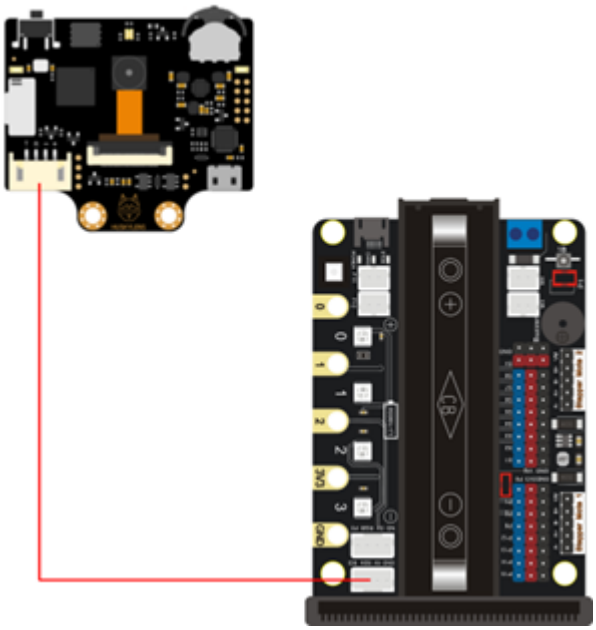
安裝 HuskyLens 到車子


螺絲

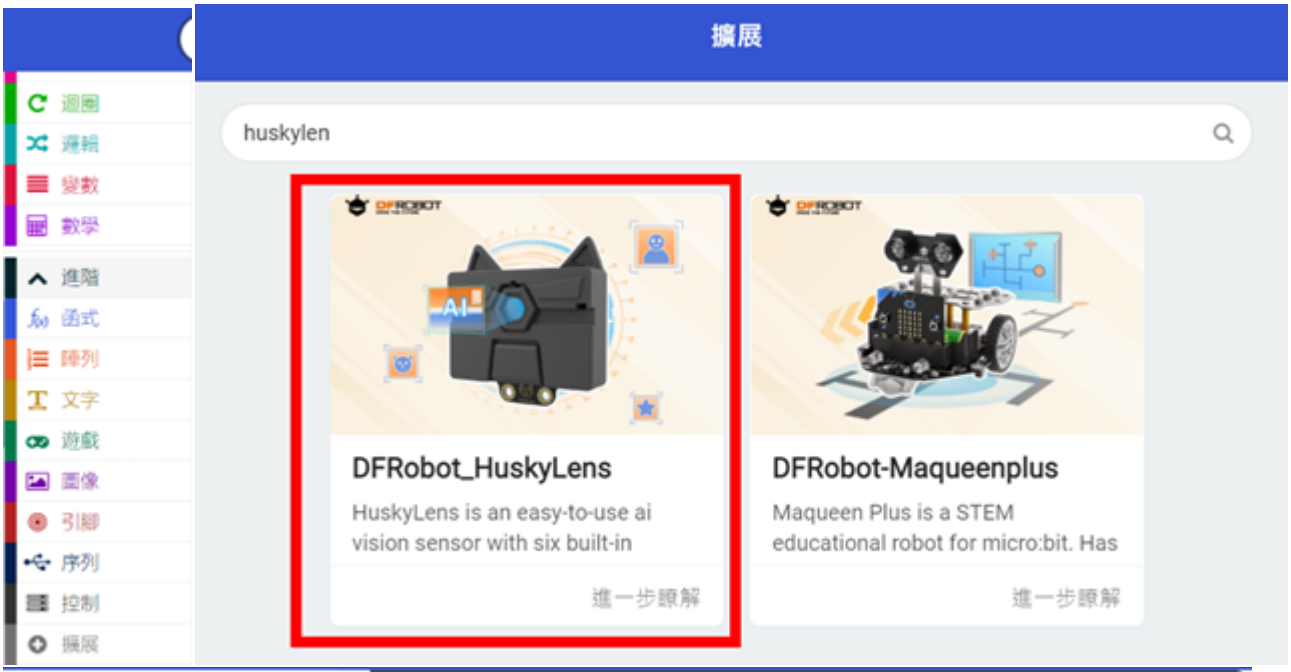

螺絲




螺絲
六角柱
螺絲帽
（在板下）

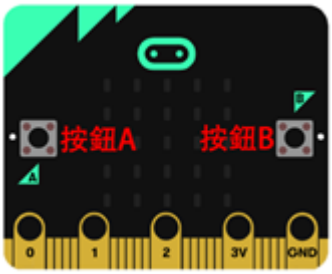


HuskyLens 擴展包



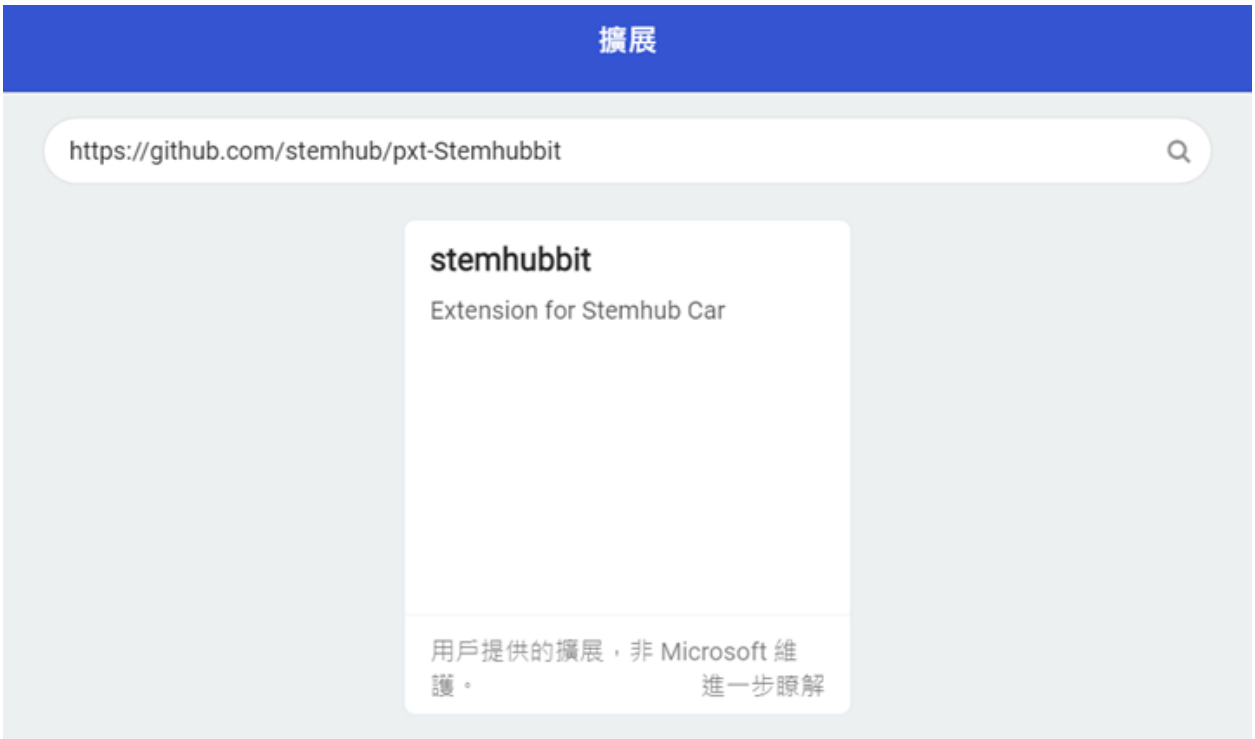
HuskyLens 積木模塊

練習 1




```
1) 程序開始時，自動切F到物體分類模式
2) 使用 HuskyLens 學習 5 種物件 (ID)
3) 按下 A 按鈕：顯示學習物件數量——(a)
4) 按下 B 按鈕：清除學習物件數據
5) 再按下 A 按鈕，顯示學習物件數量——(b)
```

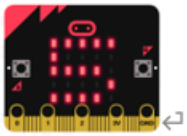
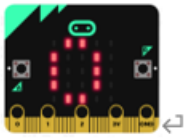
練習 2




```
1) 程序開始時，自動切🅕到物體分類模式
2) 使用 HuskyLens 學習數件物件
3) 如果有已學習物件在屏幕之內，Micro:bit 板顯示最接近屏幕中心的物件 ID，🅕將全部板載 LED 亮起🅕🅕色，
否則亮起紅色
```

答案

練習 1

位置	數字
(a)	
(b)	

當啟動時
HuskyLens通過I2C初始化直到成功
HuskyLens切換到 物體分類 算法直到成功
HuskyLens遺忘當前算法的所有學習數據
當按鈕 A 被按下
HuskyLens請求一次數據存入結果
顯示 數字 HuskyLens從結果中獲取已學ID總數
當按鈕 B 被按下
HuskyLens遺忘當前算法的所有學習數據

練習 2

當啟動時
HuskyLens通過I2C初始化直到成功
HuskyLens切換到 物體分類 算法直到成功
重複無限次
HuskyLens請求一次數據存入結果
如果 HuskyLens從結果中獲取 方框 是否在畫面中? 那麼
設置板載燈 全部燈 顏色 綠
顯示 數字 HuskyLens從結果中獲取靠近中心方框的 ID 參數
否則
設置板載燈 全部燈 顏色 紅
顯示 數字 HuskyLens從結果中獲取靠近中心方框的 ID 參數

1.5.2 第二課

簡介

教學目標

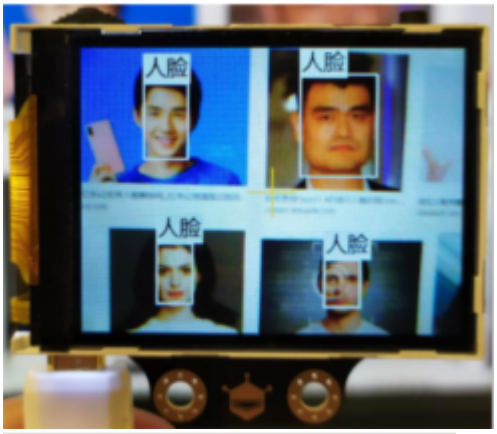
HuskyLens 人臉識Ⓕ功能

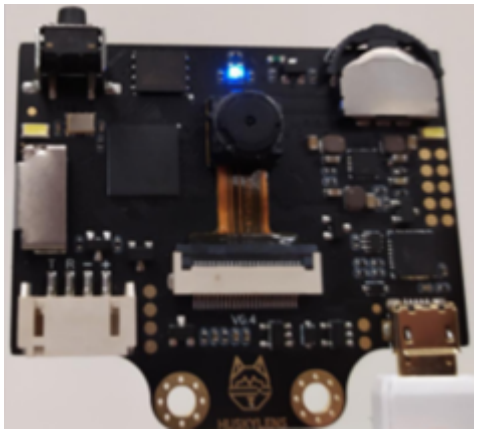
人臉識Ⓕ是什麼？

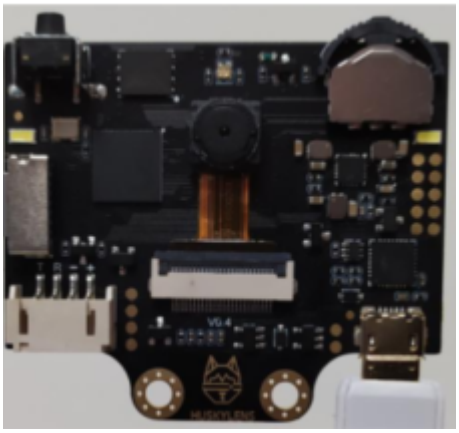
人臉識Ⓕ的應用

Husky lens 的人臉識Ⓕ

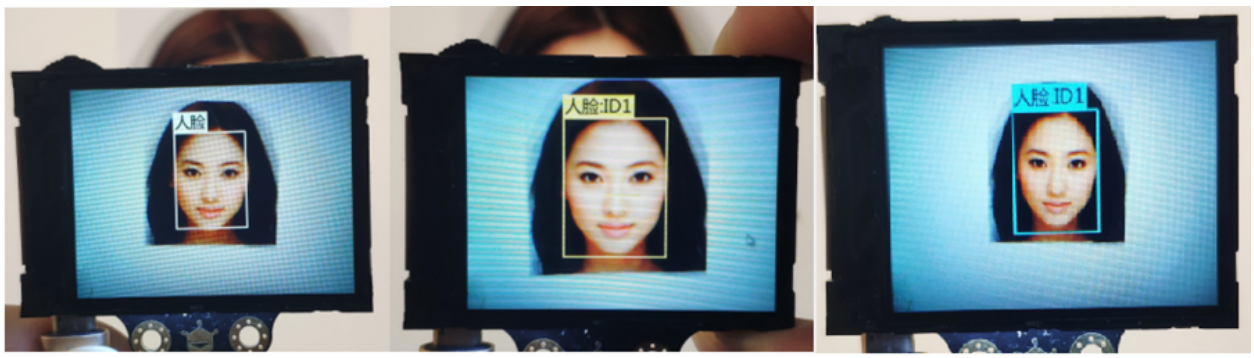
人臉探測:

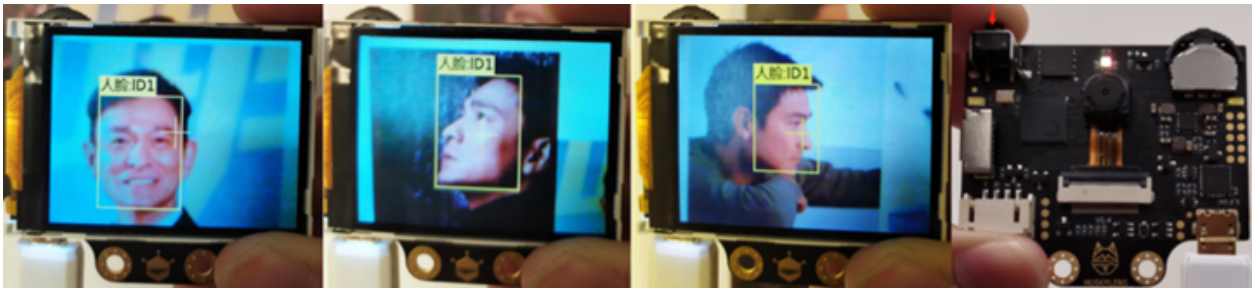






人臉學習:





人臉識F:

練習 1

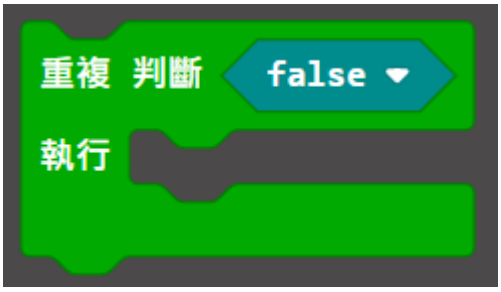
- 在 HuskyLens » 更多內有以 ID 更改名稱的積木

- F有名字的點子的話，可以嘗試以下名字：Peter, Sam, Mary
- 由於 HuskyLens 更改物件需時，一次性更改多個物件名稱有可能失敗
- 所以使用“重F無限次”

還原改變後的物件名稱

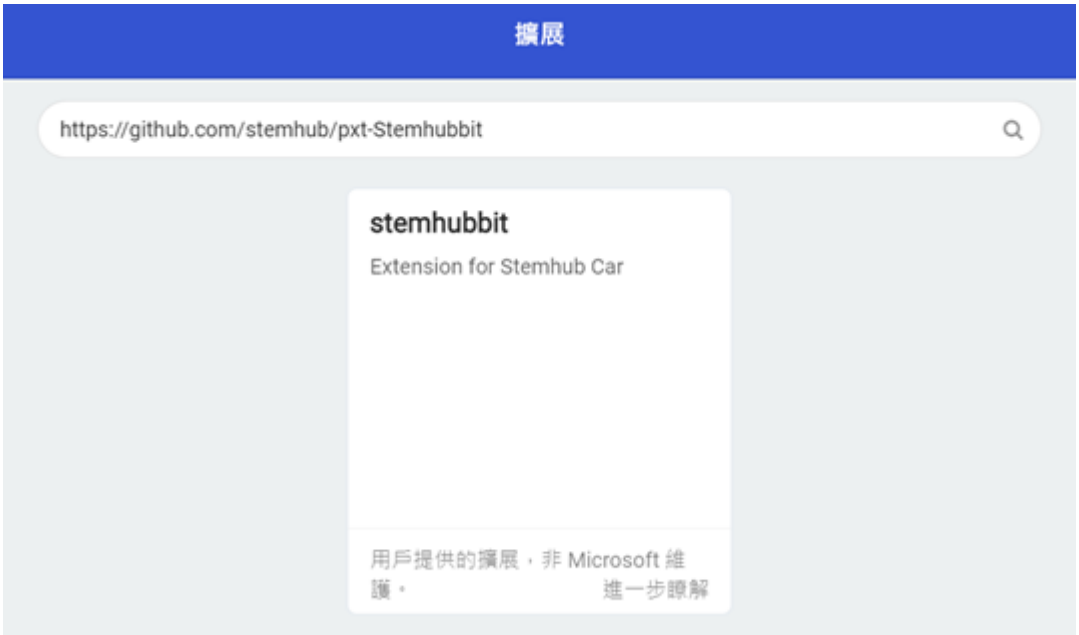
練習 2

- 建立變數”ID“以儲存每一個人臉 ID
- 使用Ⓕ圈 » 重Ⓕ判斷…執行積木將人臉名稱逐個改變

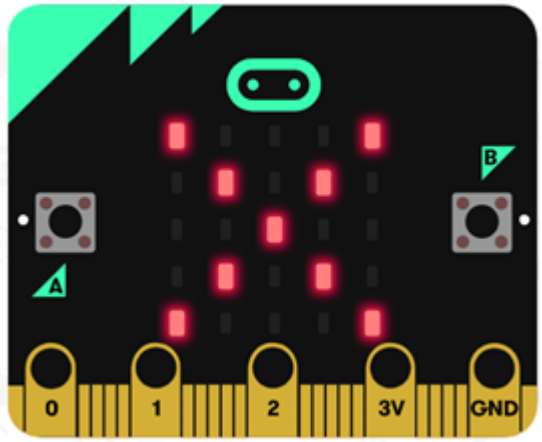


練習 3

- 假設 HuskyLens 一次只看到一張人臉
- 由於鏡頭的角度無法實時調較，因此將不同照片輪流放在車子前，或手持車子Ⓕ將鏡頭對住人臉效果較好。
- 控制車子馬達的積木在 Stemhub:bit 擴展包：https://github.com/stemhub/pxt-Stemhubbit


擴展
https://github.com/stemhub/pxt-Stemhubbit
stemhubbit
Extension for Stemhub Car
用戶提供的擴展，非 Microsoft 維護。
進一步瞭解

練習 4


B
A
0
1
2
3V
GND

答案

練習 1

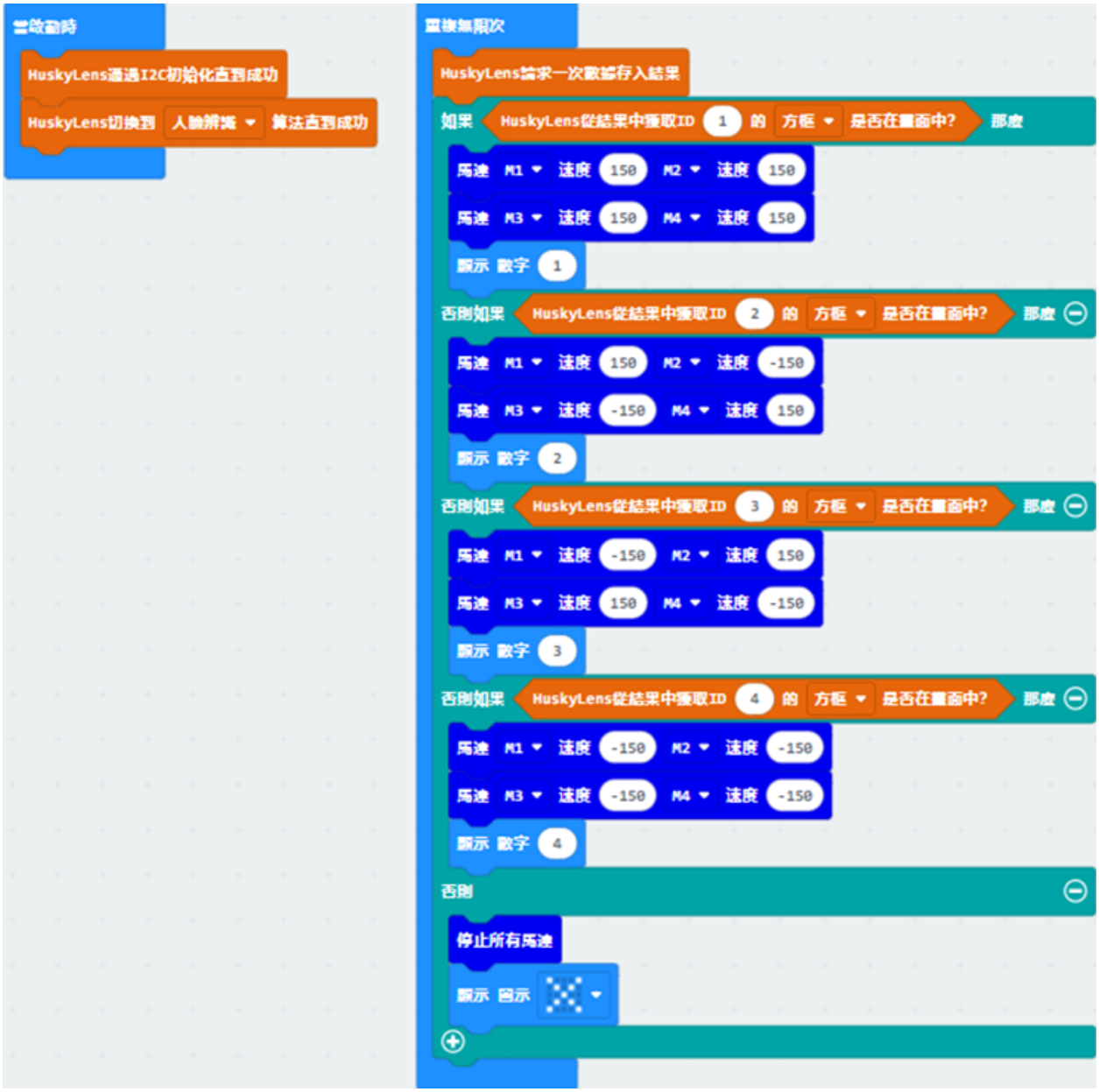
當啟動時
HuskyLens通過I2C初始化直到成功
HuskyLens切換到 人臉辨識 算法直到成功

重複無限次
HuskyLens設定當前算法ID 1 名稱 "Peter"
HuskyLens設定當前算法ID 2 名稱 "Sam"
HuskyLens設定當前算法ID 3 名稱 "Mary"

練習 2

當啟動時
HuskyLens通過I2C初始化直到成功
HuskyLens切換到 人臉辨識 算法直到成功
重複無限次
HuskyLens請求一次數據存入結果
變數 ID 設為 HuskyLens從結果中獲取已學ID總數
重複 判斷 ID > 0
執行
HuskyLens設定當前算法ID ID 名稱 " "
變數 ID 改變 -1
將ID不斷-1以改變
每個人臉ID的名稱

練習 3


當啟動時
HuskyLens通過I2C初始化直到成功
HuskyLens切換到 人臉辨識 算法直到成功
重複無限次
HuskyLens請求一次數據存入結果
如果 HuskyLens從結果中獲取ID 1 的 方框 是否在畫面中? 那麼
馬達 M1 速度 150 M2 速度 150
馬達 M3 速度 150 M4 速度 150
顯示 數字 1
否則如果 HuskyLens從結果中獲取ID 2 的 方框 是否在畫面中? 那麼
馬達 M1 速度 150 M2 速度 -150
馬達 M3 速度 -150 M4 速度 150
顯示 數字 2
否則如果 HuskyLens從結果中獲取ID 3 的 方框 是否在畫面中? 那麼
馬達 M1 速度 -150 M2 速度 150
馬達 M3 速度 150 M4 速度 -150
顯示 數字 3
否則如果 HuskyLens從結果中獲取ID 4 的 方框 是否在畫面中? 那麼
馬達 M1 速度 -150 M2 速度 -150
馬達 M3 速度 -150 M4 速度 -150
顯示 數字 4
否則
停止所有馬達
顯示 圖示

練習 4

當啟動時
HuskyLens通過I2C初始化直到成功
HuskyLens切換到 人臉辨識 算法直到成功

重複無限次
HuskyLens請求一次數據存入結果
如果 HuskyLens從結果中獲取 方框 是否在畫面中?
馬達 M1 速度 150 M2 速度 150
馬達 M3 速度 150 M4 速度 150
顯示 數字 HuskyLens從結果中獲取靠近中心方框的 ID
否則
停止所有馬達
顯示 圖示

1.5.3 第三課

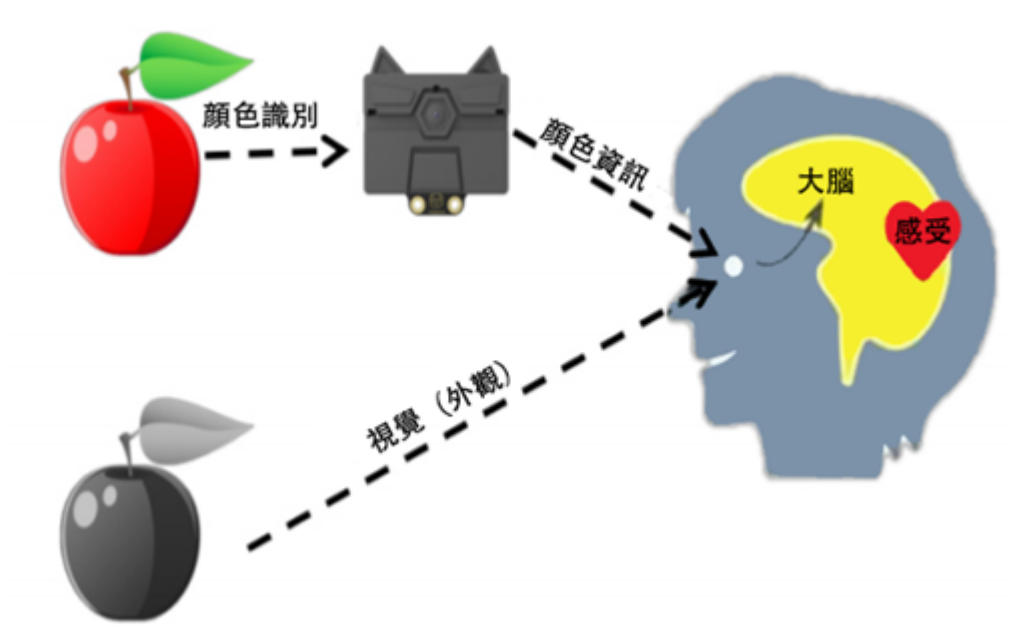
簡介

教學目標

HuskyLens Ⓕ色識Ⓕ功能

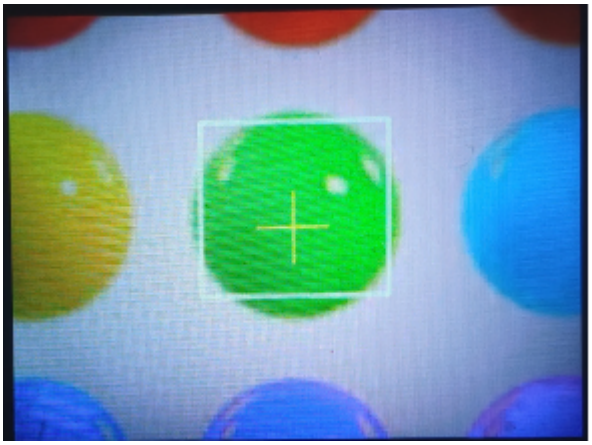
Ⓕ色識Ⓕ是什麼？

Ⓕ色識Ⓕ的應用

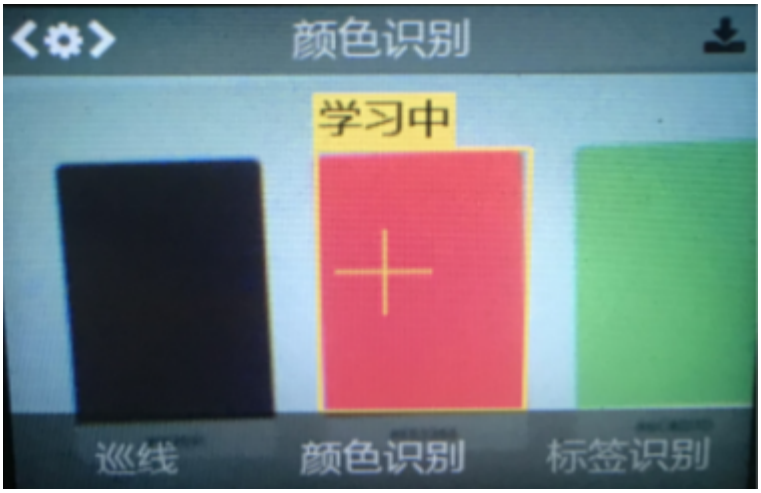


Huskylens 的Ⓕ色識Ⓕ

偵測Ⓕ色:



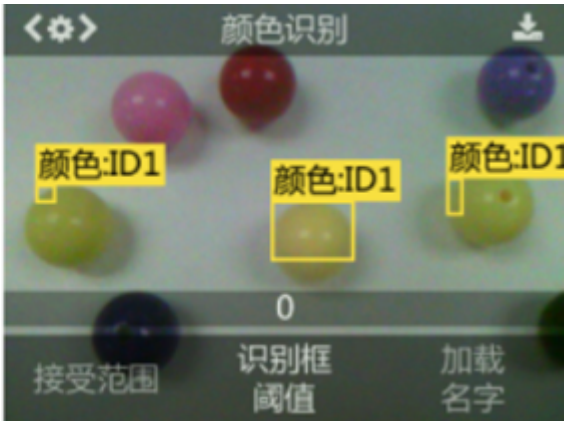
學習Ⓕ色:



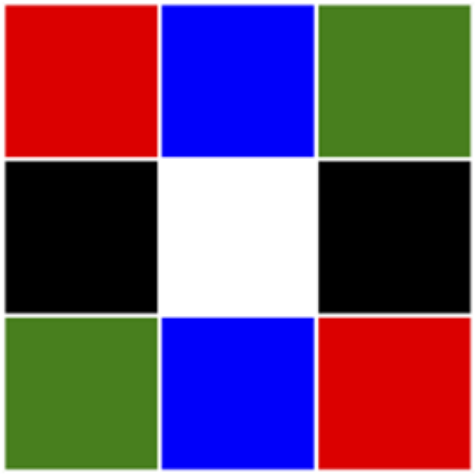
識ⒻⒻ色:

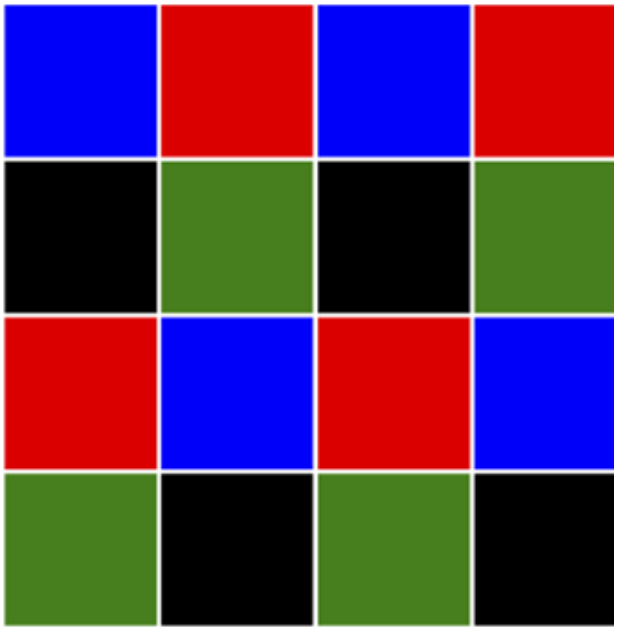
調節識Ⓕ框Ⓕ值

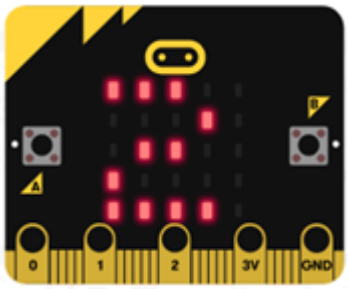
颜色识别
颜色:ID1
20
接受范围
识别框
阈值
加载
名字


颜色识别
颜色:ID1
颜色:ID1
颜色:ID1
0
接受范围
识别框
阈值
加载
名字

練習 1:

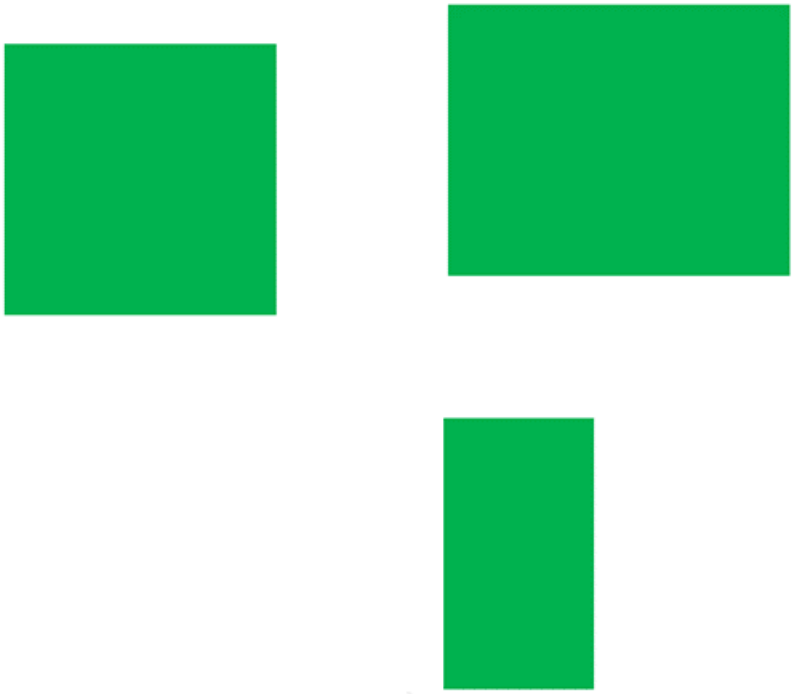





0
1
2
3V
GND

HuskyLens從結果中獲取ID
1
方框
總數

練習 2:

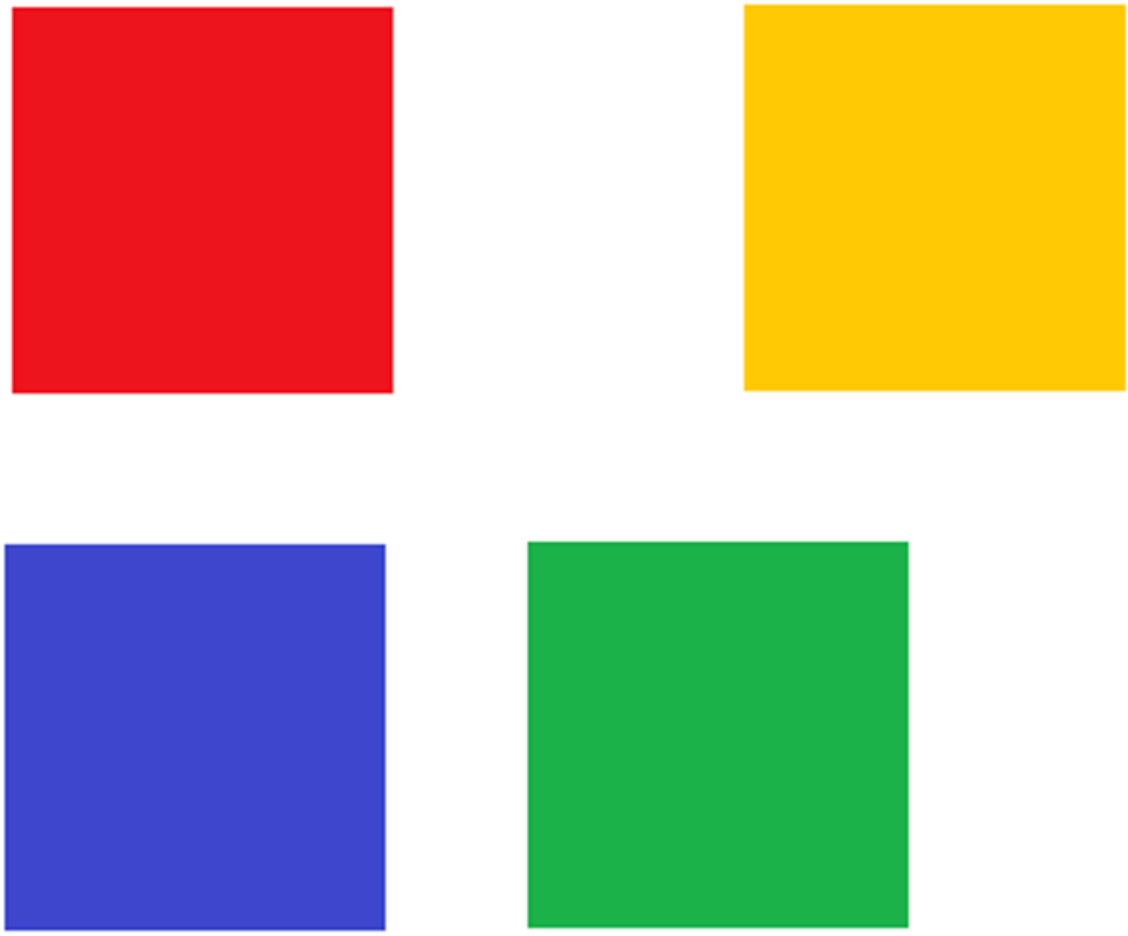


- 使用以下 Huskylens 積木獲取每一個色塊的高闊，面積 = (長) 高度 × 闊度

- 建立」number」儲存」HuskyLens 從結果中獲取 ID 1 方框總數」
- 建立」area」儲存增加後的面積，使用Ⓕ圈將色塊面積逐個增加

練習 3:

- ID 1 亮起紅色
- ID 2 亮起Ⓕ色
- ID 3 亮起藍色
- ID 4 亮起Ⓕ色



練習 4（挑戰）:

```
1）當 Huskylens 偵測到紅色最接近屏幕中心時，車子開始前進
2）直至 Huskylens 偵測到紫色格子最接近屏幕中心，車子短暫停下🅵開始後退
3）一直後退直至 Huskylens 再次偵測到紅色格子最接近屏幕中心，車子停下
```

- 先調校 Huskylens 角度望住紅色格子
- 建立變數」direction」記🅵車子移動方向，以🅵定車子動作
- 建立變數」finish」判定車子是否已回到紅色格子
- 未回到（" false "）播放中音 C 一秒，🅵將 finish 設爲 "true"
- finish 爲" true "時停止繼續播放音色

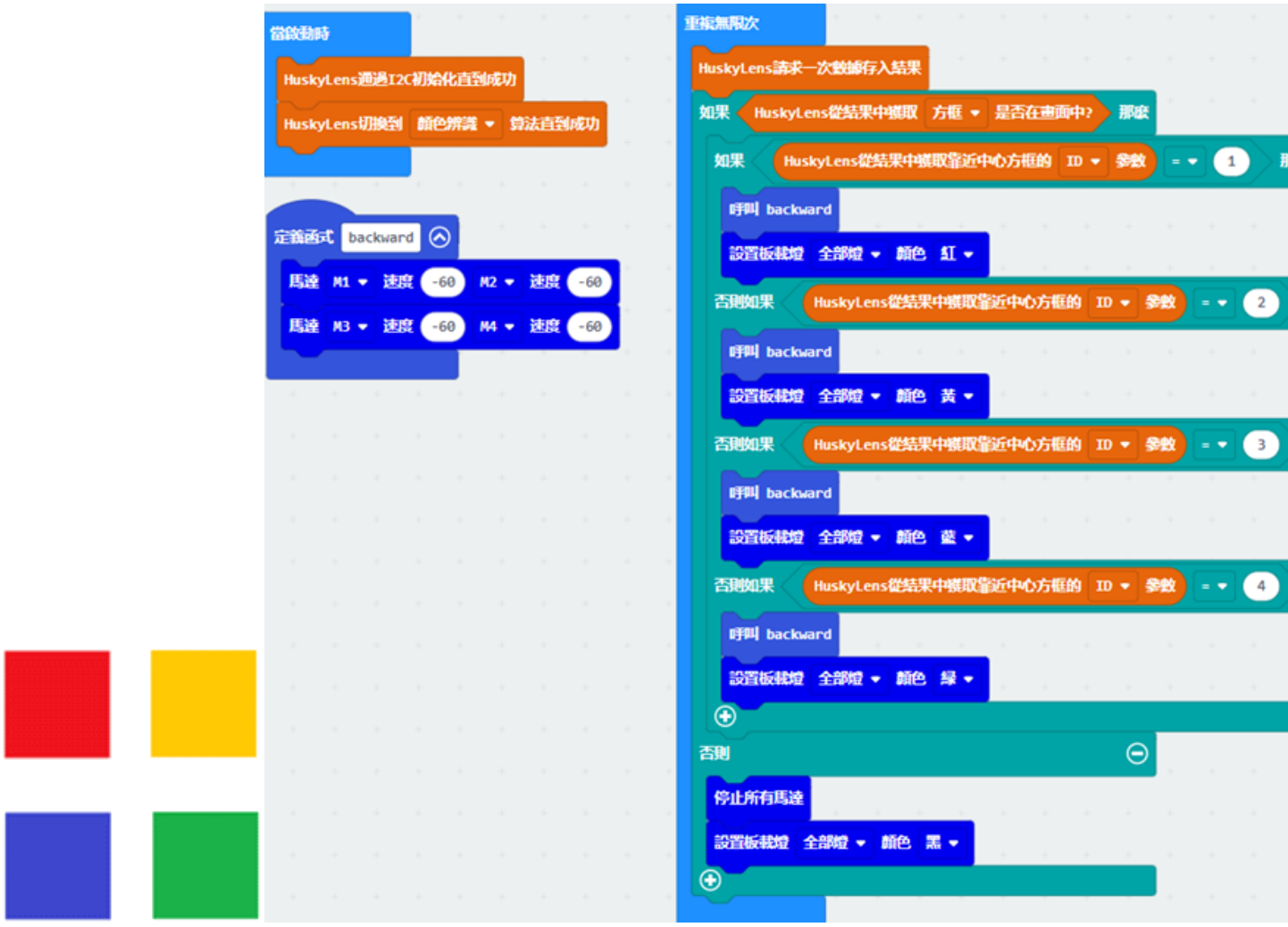
答案

練習 1

練習 2

當啟動時
HuskyLens通過I2C初始化直到成功
HuskyLens切換到 顏色辨識 算法直到成功
定義函式 totalArea
計算所有色塊總面積的程序
變數 height 設為 0
變數 width 設為 0
先初始化高和闊以能夠獲取子函式的高和闊度 (如不使用子函式則非必要)
呼叫 getHeight
呼叫 getWidth
將目前色塊總面積加上新一個色塊外面積
變數 area 設為 area + height x width
當按鈕 A 被按下
HuskyLens請求一次數據存入結果
變數 area 設為 0
變數 number 設為 HuskyLens從結果中擷取ID 1 方框 總數
重複 判斷 number > 0
執行
呼叫 totalArea
變數 number 改變 -1
將number不斷-1以得到每一個色塊的資訊
顯示 數字 area
定義函式 getHeight
獲得一個色塊的高度
變數 height 設為 HuskyLens從結果中擷取ID 1 第 number 個方框的參數 高度
定義函式 getWidth
獲得一個色塊的闊度
變數 width 設為 HuskyLens從結果中擷取ID 1 第 number 個方框的參數 寬度

練習 3


當啟動時
HuskyLens通過I2C初始化直到成功
HuskyLens切換到 顏色辨識 算法直到成功
定義函式 backward
馬達 M1 速度 -60 M2 速度 -60
馬達 M3 速度 -60 M4 速度 -60
重複無限次
HuskyLens請求一次數據存入結果
如果 HuskyLens從結果中選取 方框 是否在畫面中? 那麼
如果 HuskyLens從結果中選取靠近中心方框的 ID 參數 = 1
呼叫 backward
設置板載燈 全部燈 顏色 紅
否則如果 HuskyLens從結果中選取靠近中心方框的 ID 參數 = 2
呼叫 backward
設置板載燈 全部燈 顏色 黃
否則如果 HuskyLens從結果中選取靠近中心方框的 ID 參數 = 3
呼叫 backward
設置板載燈 全部燈 顏色 藍
否則如果 HuskyLens從結果中選取靠近中心方框的 ID 參數 = 4
呼叫 backward
設置板載燈 全部燈 顏色 綠
否則
停止所有馬達
設置板載燈 全部燈 顏色 黑

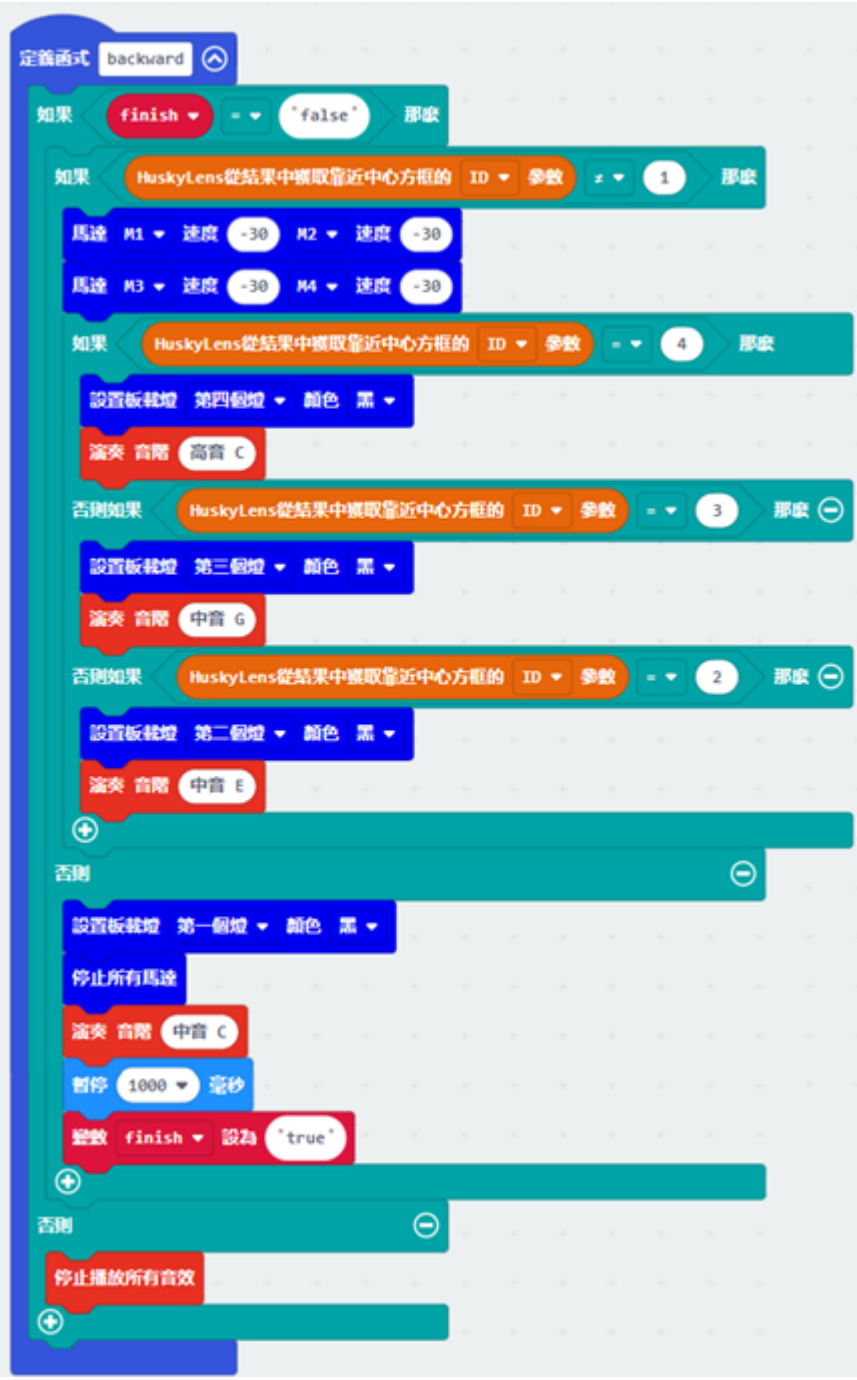
練習 4

當啟動時
停止播放所有音效
設置板載燈 全部燈 顏色 黑
HuskyLens通過I2C初始化直到成功
HuskyLens切換到 顏色辨識 算法直到成功
變數 direction 設為 "forward"
變數 finish 設為 "false"
重複無限次
HuskyLens請求一次數據存入結果
如果 HuskyLens從結果中獲取 方框 是否在畫面中? 那麼
如果 direction = "forward" 那麼
呼叫 forward
否則如果 direction = "backward" 那麼
呼叫 backward
否則
停止所有馬達
設置板載燈 全部燈 顏色 黑
停止播放所有音效

前進的程序

定義函式 forward
如果 HuskyLens從結果中獲取靠近中心方框的 ID 參數 ≠ 4 那麼
如果 HuskyLens從結果中獲取靠近中心方框的 ID 參數 = 1 那麼
馬達 M1 速度 30 M2 速度 30
馬達 M3 速度 30 M4 速度 30
設置板載燈 第一個燈 顏色 紅
演奏 音階 中音 C
否則如果 HuskyLens從結果中獲取靠近中心方框的 ID 參數 = 2 那麼
設置板載燈 第二個燈 顏色 黃
演奏 音階 中音 E
否則如果 HuskyLens從結果中獲取靠近中心方框的 ID 參數 = 3 那麼
設置板載燈 第三個燈 顏色 綠
演奏 音階 中音 G
否則
設置板載燈 第四個燈 顏色 紫
演奏 音階 高音 C
停止所有馬達
暫停 1000 毫秒
變數 direction 設為 "backward"

後退的程序


定義函式 backward
如果 finish = 'false' 那麼
如果 HuskyLens從結果中擷取靠近中心方框的 ID 參數 ≠ 1 那麼
馬達 M1 速度 -30 M2 速度 -30
馬達 M3 速度 -30 M4 速度 -30
如果 HuskyLens從結果中擷取靠近中心方框的 ID 參數 = 4 那麼
設置板載燈 第四個燈 顏色 黑
演奏 音階 高音 C
否則如果 HuskyLens從結果中擷取靠近中心方框的 ID 參數 = 3 那麼
設置板載燈 第三個燈 顏色 黑
演奏 音階 中音 G
否則如果 HuskyLens從結果中擷取靠近中心方框的 ID 參數 = 2 那麼
設置板載燈 第二個燈 顏色 黑
演奏 音階 中音 E
否則
設置板載燈 第一個燈 顏色 黑
停止所有馬達
演奏 音階 中音 C
暫停 1000 毫秒
變數 finish 設為 'true'
否則
停止播放所有音效

1.5.4 第四課

簡介

教學目標

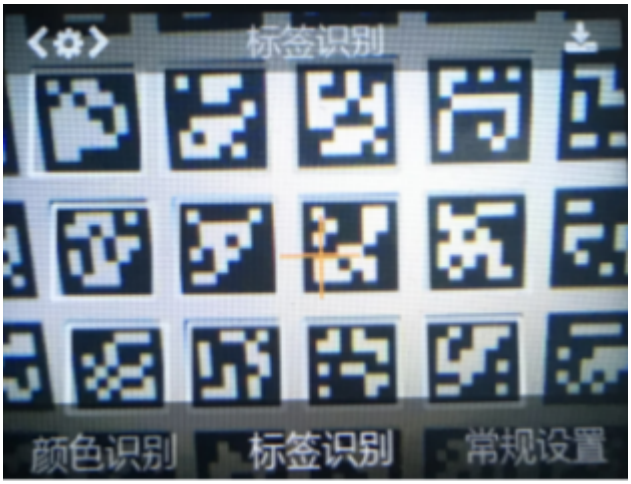
HuskyLens 標Ⓕ識Ⓕ功能

標Ⓕ識Ⓕ是什麼？

標簽識Ⓕ的應用

Huskylens 的標Ⓕ識Ⓕ

偵測標Ⓕ



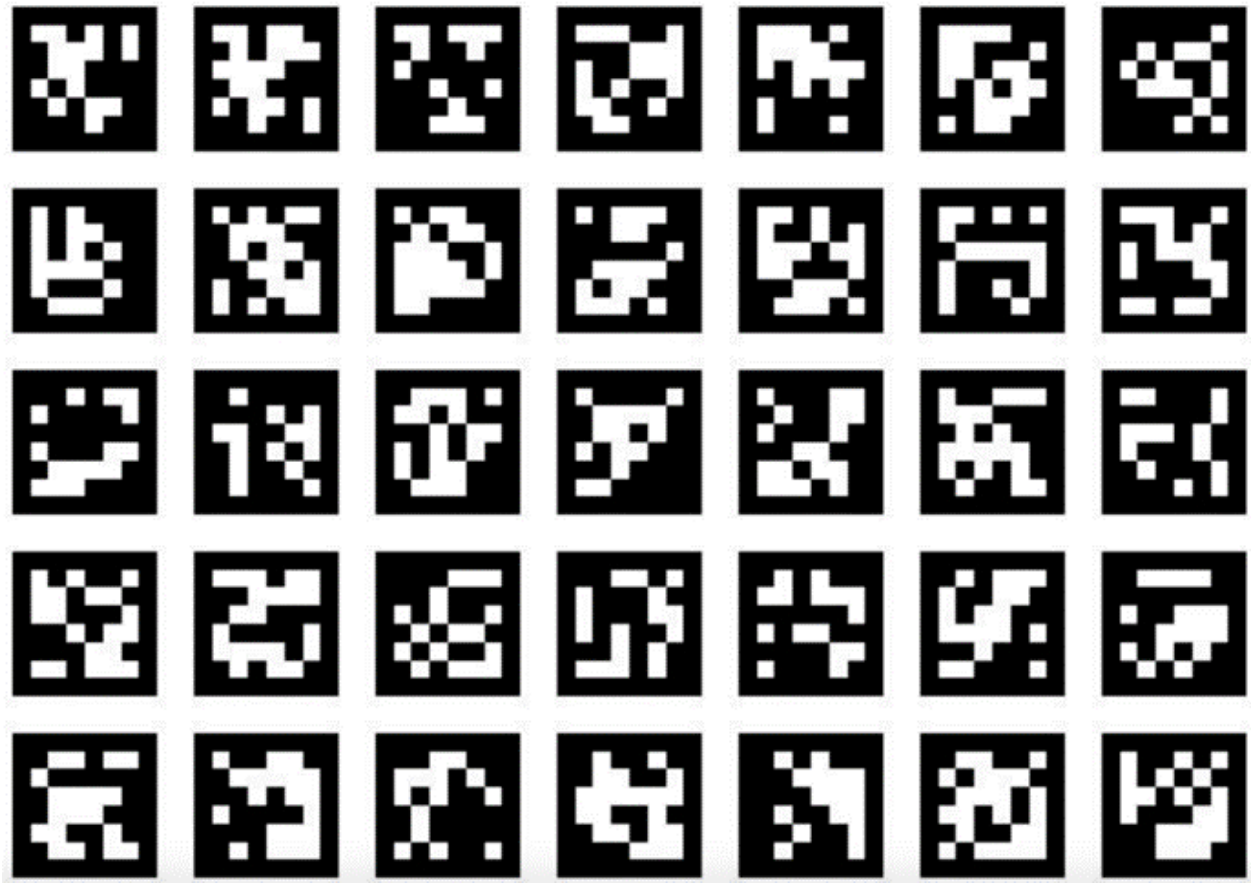
學習標Ⓕ

标签识别
标签:ID1
颜色识别
标签识别
常规设置

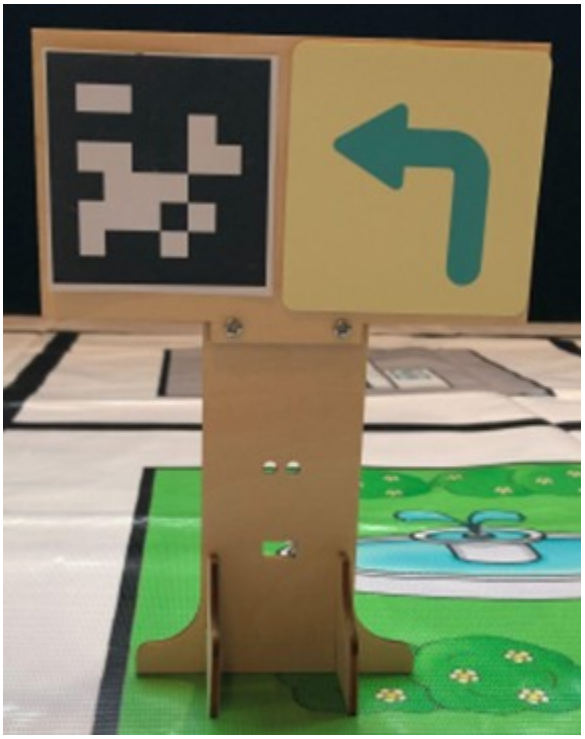
識Ⓕ標Ⓕ

标签识别
标签:ID1
标签:ID2
标签:ID3
颜色识别
标签识别
常规设置

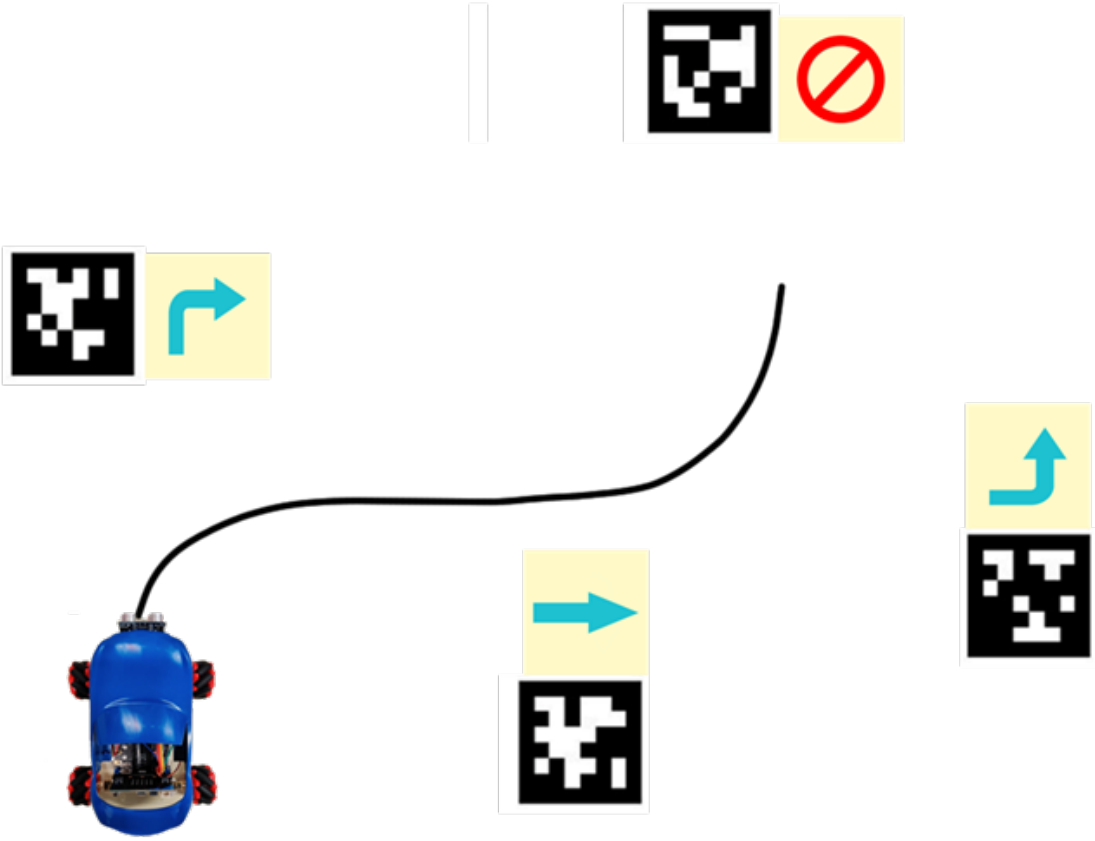
AprilTag



練習 1:



練習 2:

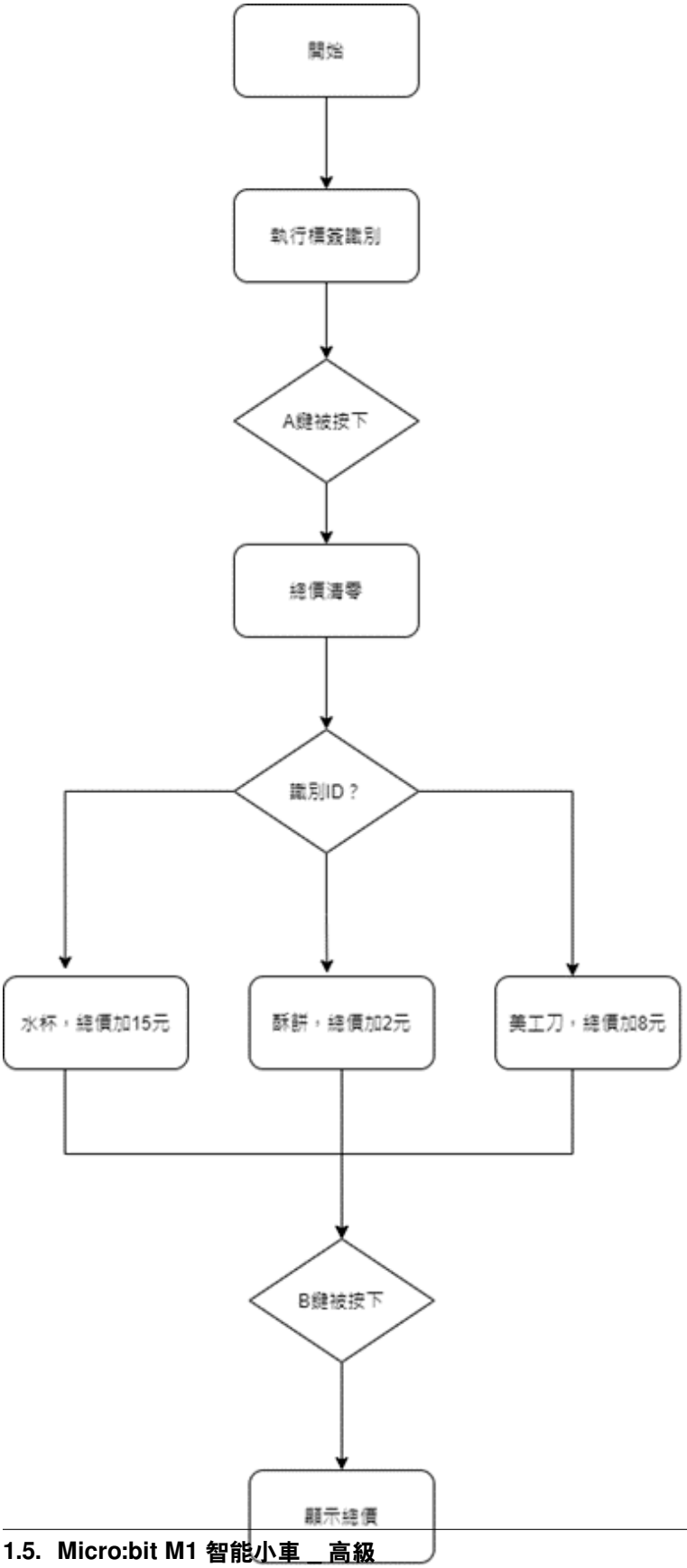


練習 3: 自助超市收銀機

任務一：識Ⓕ商品

任務二：開始與結束掃碼

任務三：商品結算

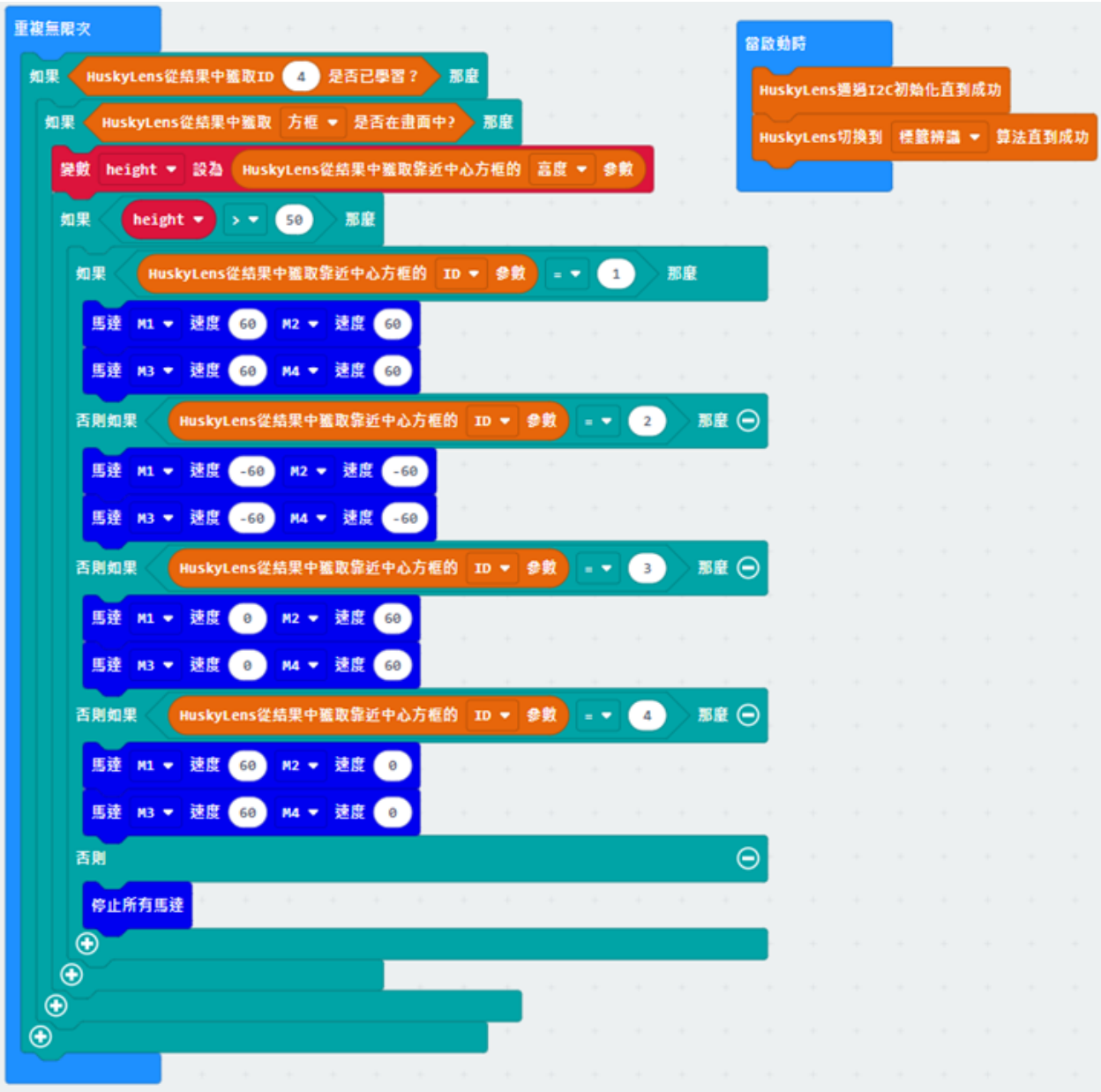

開始
執行標籤識別
A鍵被按下
總價清零
識別ID？
水杯，總價加15元
酥餅，總價加2元
美工刀，總價加8元
B鍵被按下
顯示總價

答案

練習 1

重複無限次
HuskyLens請求一次數據存入結果
如果 HuskyLens從結果中獲取ID 4 是否已學習？ 那麼
如果 HuskyLens從結果中獲取 方框 是否在畫面中? 那麼
如果 HuskyLens從結果中獲取靠近中心方框的 ID 參數 = 1 那麼
馬達 M1 速度 180 M2 速度 180
馬達 M3 速度 180 M4 速度 180
否則如果 HuskyLens從結果中獲取靠近中心方框的 ID 參數 = 2 那麼
馬達 M1 速度 -160 M2 速度 -160
馬達 M3 速度 -160 M4 速度 -160
否則如果 HuskyLens從結果中獲取靠近中心方框的 ID 參數 = 3 那麼
馬達 M1 速度 0 M2 速度 80
馬達 M3 速度 0 M4 速度 80
否則如果 HuskyLens從結果中獲取靠近中心方框的 ID 參數 = 4 那麼
馬達 M1 速度 80 M2 速度 0
馬達 M3 速度 80 M4 速度 0
否則
停止所有馬達
當啟動時
HuskyLens通過I2C初始化直到成功
HuskyLens切換到 標籤辨識 算法直到成功

練習 2


重複無限次
如果 HuskyLens從結果中獲取ID 4 是否已學習？ 那麼
如果 HuskyLens從結果中獲取 方框 是否在畫面中? 那麼
變數 height 設為 HuskyLens從結果中獲取靠近中心方框的 高度 參數
如果 height > 50 那麼
如果 HuskyLens從結果中獲取靠近中心方框的 ID 參數 = 1 那麼
馬達 M1 速度 60 M2 速度 60
馬達 M3 速度 60 M4 速度 60
否則如果 HuskyLens從結果中獲取靠近中心方框的 ID 參數 = 2 那麼
馬達 M1 速度 -60 M2 速度 -60
馬達 M3 速度 -60 M4 速度 -60
否則如果 HuskyLens從結果中獲取靠近中心方框的 ID 參數 = 3 那麼
馬達 M1 速度 0 M2 速度 60
馬達 M3 速度 0 M4 速度 60
否則如果 HuskyLens從結果中獲取靠近中心方框的 ID 參數 = 4 那麼
馬達 M1 速度 60 M2 速度 0
馬達 M3 速度 60 M4 速度 0
否則
停止所有馬達
當啟動時
HuskyLens通過I2C初始化直到成功
HuskyLens切換到 標籤辨識 算法直到成功

練習 3

重複無限次
如果 按鈕 A 被按下？ 那麼
變數 total 設為 0
變數 B pressed? 設為 "false"
建立變數儲存B鍵的狀態，
預設為沒有被按住 （false）
重複 判斷 B pressed? = "false"
執行 如果 按鈕 B 被按下？ 那麼
當變數狀態為false，
不停檢查B鍵有沒有被按下
HuskyLens請求一次數據存入結果
如果 HuskyLens從結果中獲取ID 1 的 方框 是否在畫面中？ 那麼
顯示 文字 "bottle"
變數 total 改變 15
判斷識別的數據結果中是否有先
前學習過的標籤，每個學習過的
標籤都有相應的ID
每次識別到商品時，變量總價會加
上該商品對應的售價
如果 HuskyLens從結果中獲取ID 2 的 方框 是否在畫面中？ 那麼
顯示 文字 "biscuit"
變數 total 改變 2
如果 HuskyLens從結果中獲取ID 3 的 方框 是否在畫面中？ 那麼
顯示 文字 "knife"
變數 total 改變 8
變數 B pressed? 設為 "true"
在按下B鍵後， 變數記錄B按鍵狀態為按下（true），跳出循環
顯示 數字 total
當啟動時
HuskyLens通過I2C初始化直到成功
HuskyLens切換到 標籤辨識 算法直到成功

1.5.5 第五課

簡介

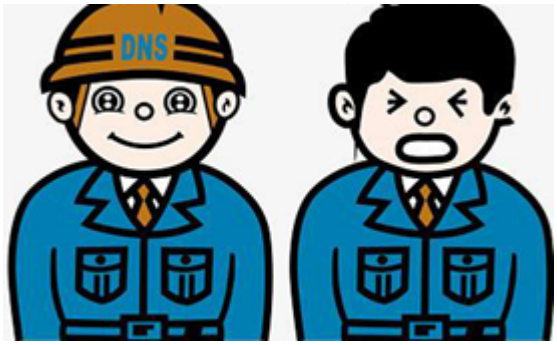
教學目標

物體分類

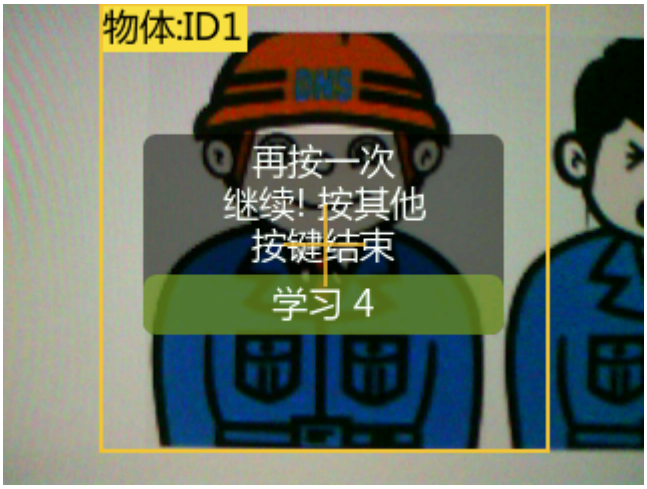
物體分類是什麼？

HuskyLens 物體分類與其他功能的差Ⓕ

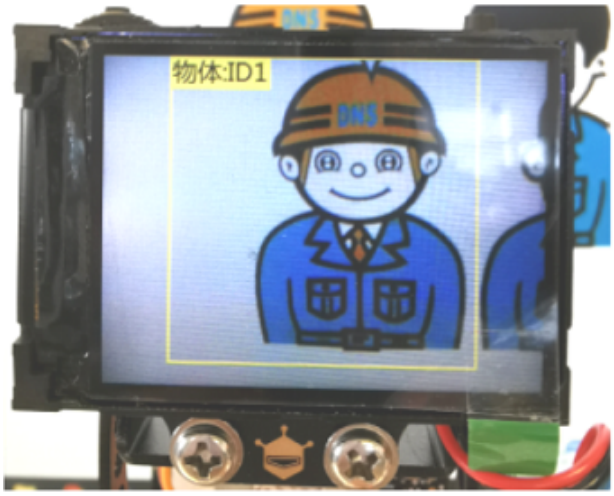
Huskylens 的物體分類



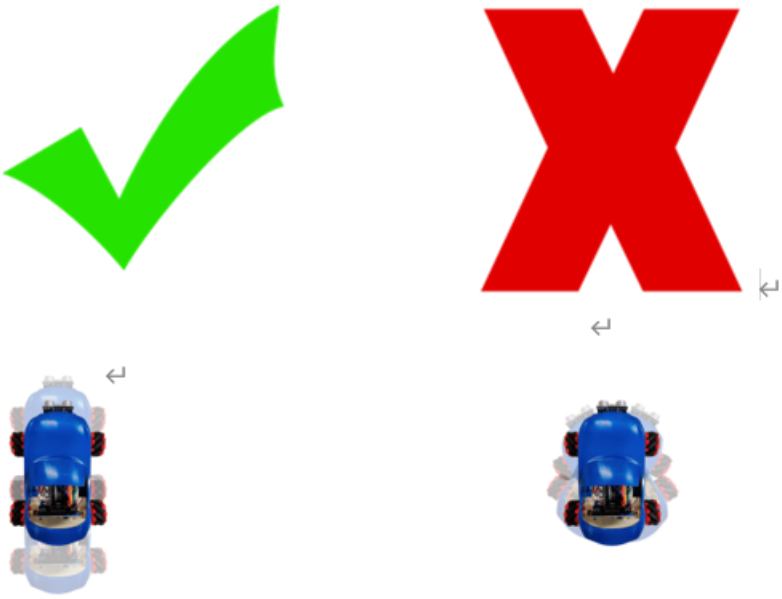
學習物體



識Ⓕ標Ⓕ物體



練習 1：圖案分類



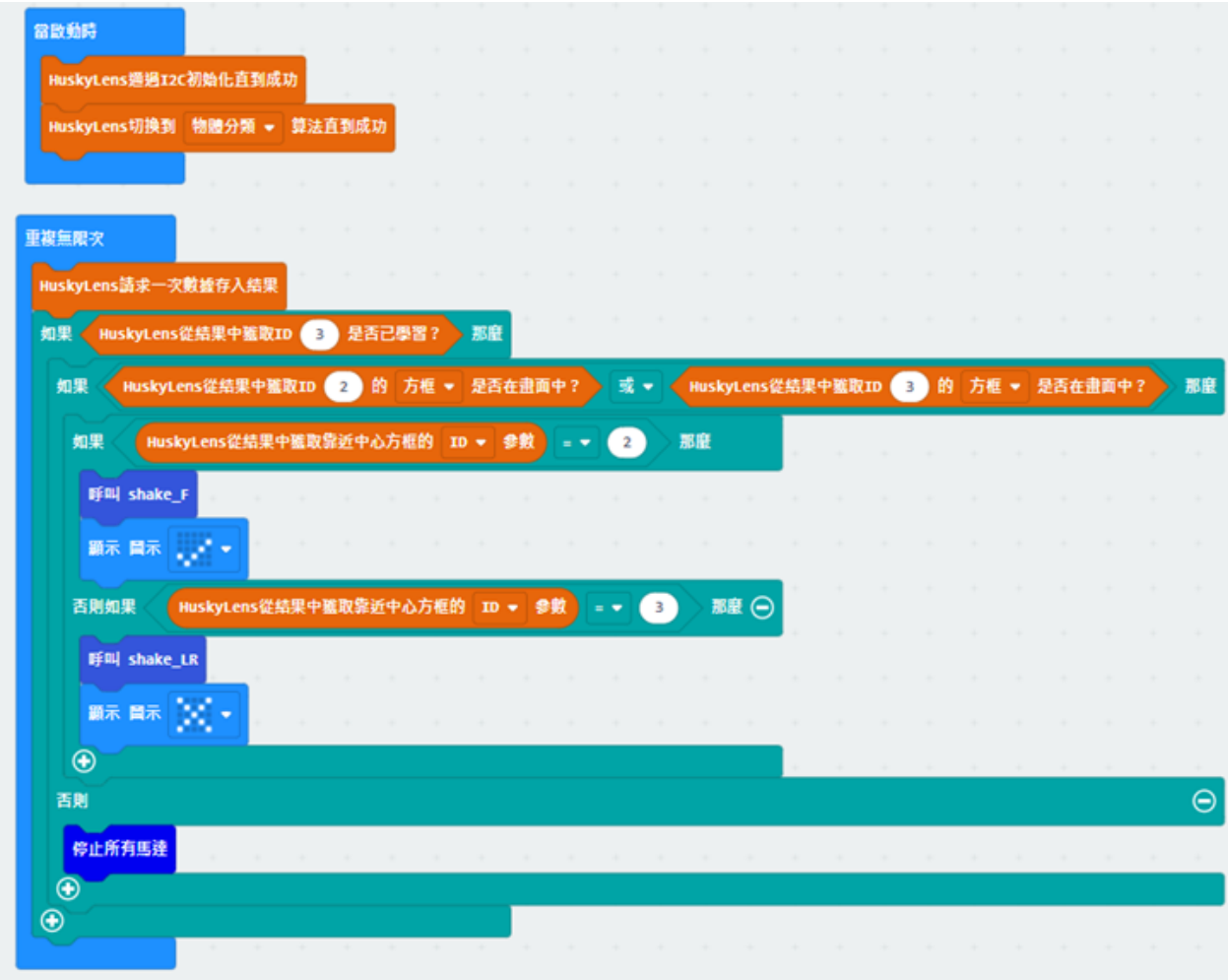
練習 2：位置識F

練習 3：巡Ⓕ移動

答案

練習 1

程序主體


當啟動時
HuskyLens通過I2C初始化直到成功
HuskyLens切換到 物體分類 算法直到成功
重複無限次
HuskyLens請求一次數據存入結果
如果 HuskyLens從結果中獲取ID 3 是否已學習？ 那麼
如果 HuskyLens從結果中獲取ID 2 的 方框 是否在畫面中？ 或 HuskyLens從結果中獲取ID 3 的 方框 是否在畫面中？ 那麼
如果 HuskyLens從結果中獲取靠近中心方框的 ID 參數 = 2 那麼
呼叫 shake_F
顯示 圖示
否則如果 HuskyLens從結果中獲取靠近中心方框的 ID 參數 = 3 那麼
呼叫 shake_LR
顯示 圖示
否則
停止所有馬達

車子向左右Ⓕ動以及向前後Ⓕ動的程序

練習 2

當啟動時
HuskyLens過過I2C初始化直到成功
HuskyLens切換到 物體分類 算法直到成功
重複無限次
HuskyLens請求一次數據存入結果
如果 HuskyLens從結果中獲取ID 4 是否已學習？ 那麼
物件指向左時
如果 HuskyLens從結果中獲取靠近中心方框的 ID 參數 = 2 那麼
暫停 100 毫秒
馬達 M1 速度 0 M2 速度 70
馬達 M3 速度 70 M4 速度 0
向左旋轉
停止所有馬達
物件向前
否則如果 HuskyLens從結果中獲取靠近中心方框的 ID 參數 = 3 那麼
停止所有馬達
車子停下
物件指向右時
否則如果 HuskyLens從結果中獲取靠近中心方框的 ID 參數 = 4 那麼
馬達 M1 速度 70 M2 速度 0
馬達 M3 速度 0 M4 速度 70
向右旋轉
否則
停止所有馬達

練習 3

當啟動時
HuskyLens通過I2C初始化直到成功
HuskyLens切換到 物體分類 算法直到成功
重複無限次
HuskyLens請求一次數據存入結果
如果 HuskyLens從結果中獲取ID 4 是否已學習？ 那麼
如果 HuskyLens從結果中獲取靠近中心方框的 ID 參數 = 2 那麼
馬達 M1 速度 70 M2 速度 70
馬達 M3 速度 70 M4 速度 70
否則如果 HuskyLens從結果中獲取靠近中心方框的 ID 參數 = 3 那麼
馬達 M1 速度 -40 M2 速度 60
馬達 M3 速度 40 M4 速度 60
否則如果 HuskyLens從結果中獲取靠近中心方框的 ID 參數 = 4 那麼
馬達 M1 速度 60 M2 速度 -40
馬達 M3 速度 60 M4 速度 40
否則
停止所有馬達
暫停 10 毫秒
綫條往前
車子前進
綫條往左上
車子往左回綫
綫條往右上
車子往右回綫
到達終點、探測不到綫條時車子停下

1.5.6 第六課

簡介

教學目標

循Ⓕ功能

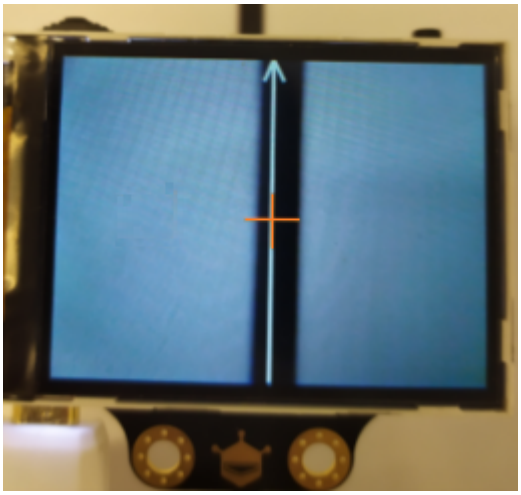
循Ⓕ功能是什麼？

循Ⓕ功能的應用

- 在商場或博物館等公衆場所用於導航
- 用於工業運輸，比起傳統輸送帶節省大量空間
- 餐廳運送食物至客人身邊

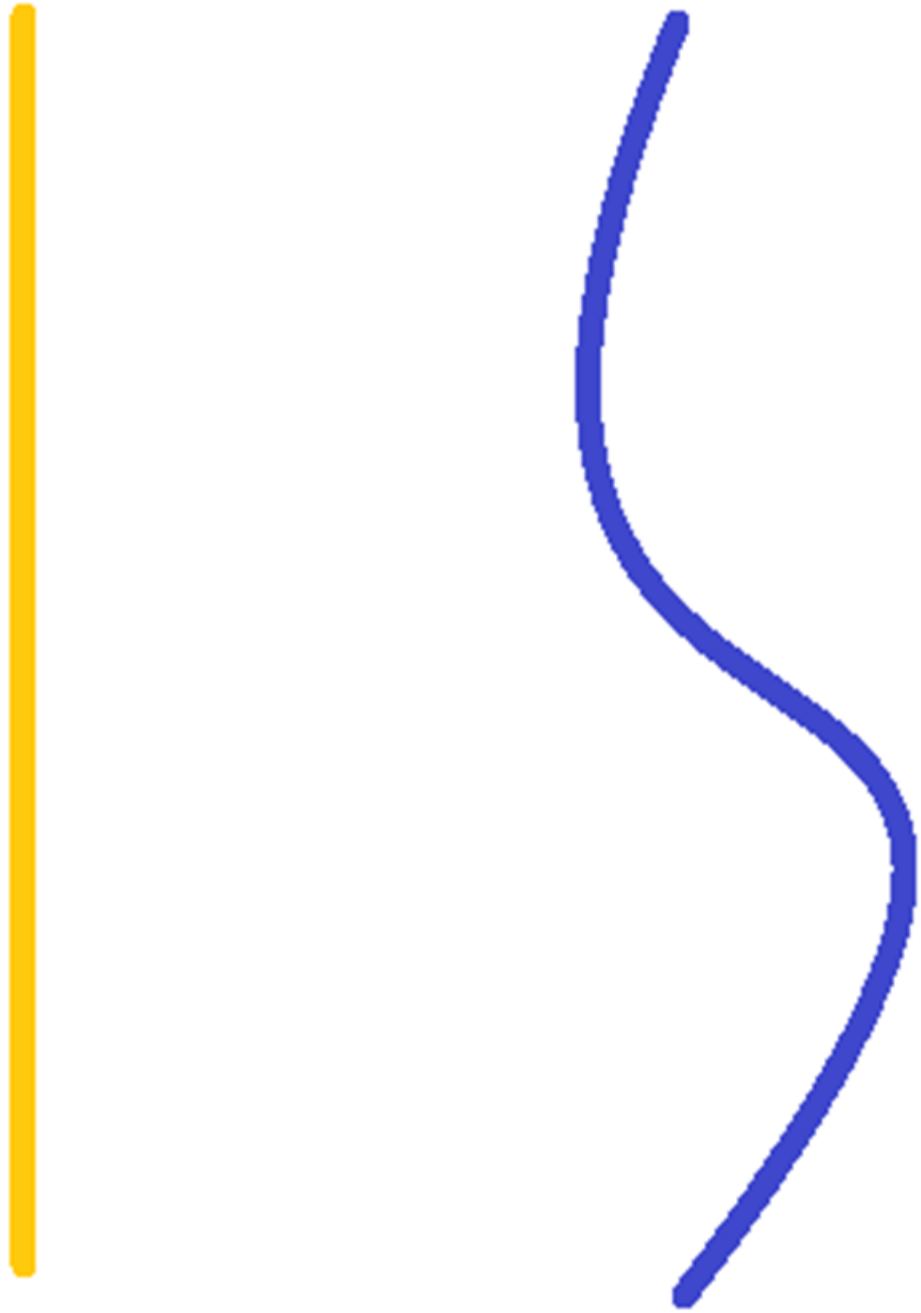
Huskylens 的循Ⓕ功能

學習Ⓕ條

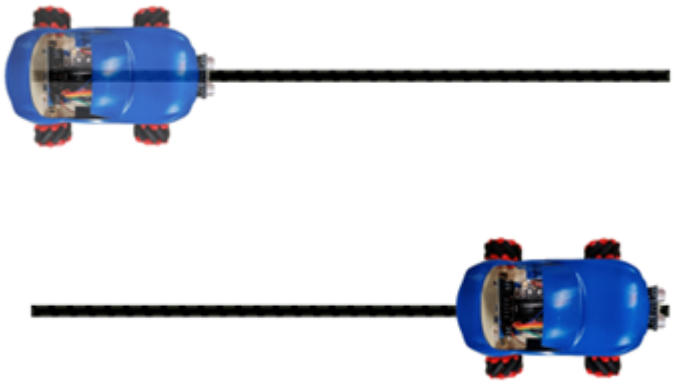


巡Ⓕ追踪

練習 1

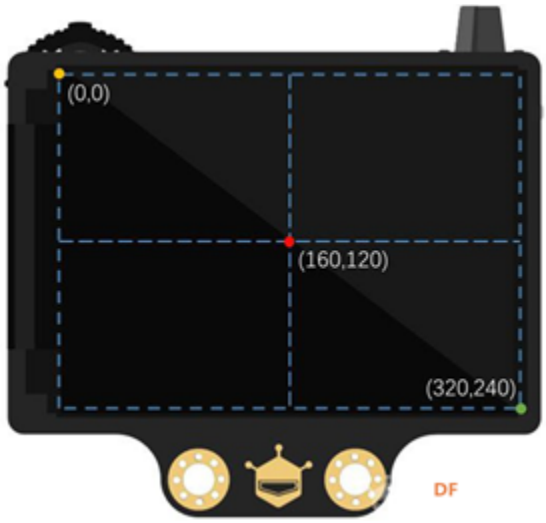


練習 2

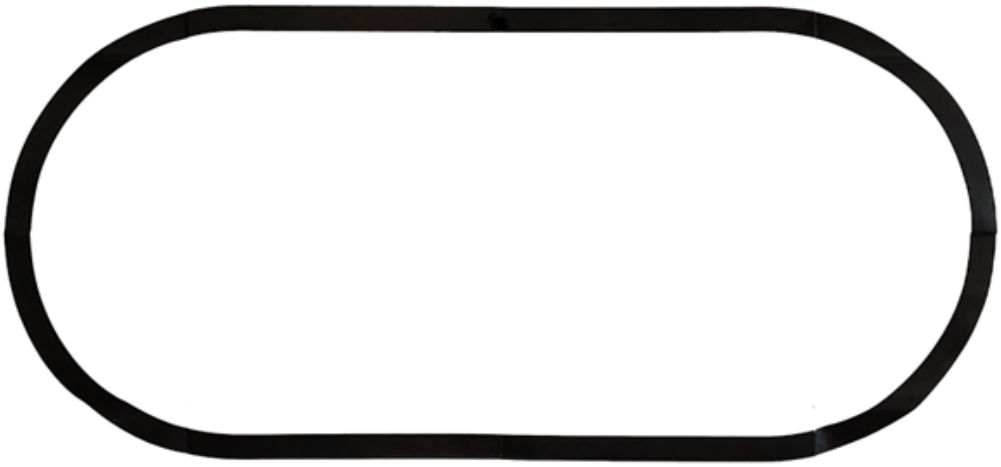


- 以不同的 X 起點Ⓕ定車子應該前進、轉左還是轉右

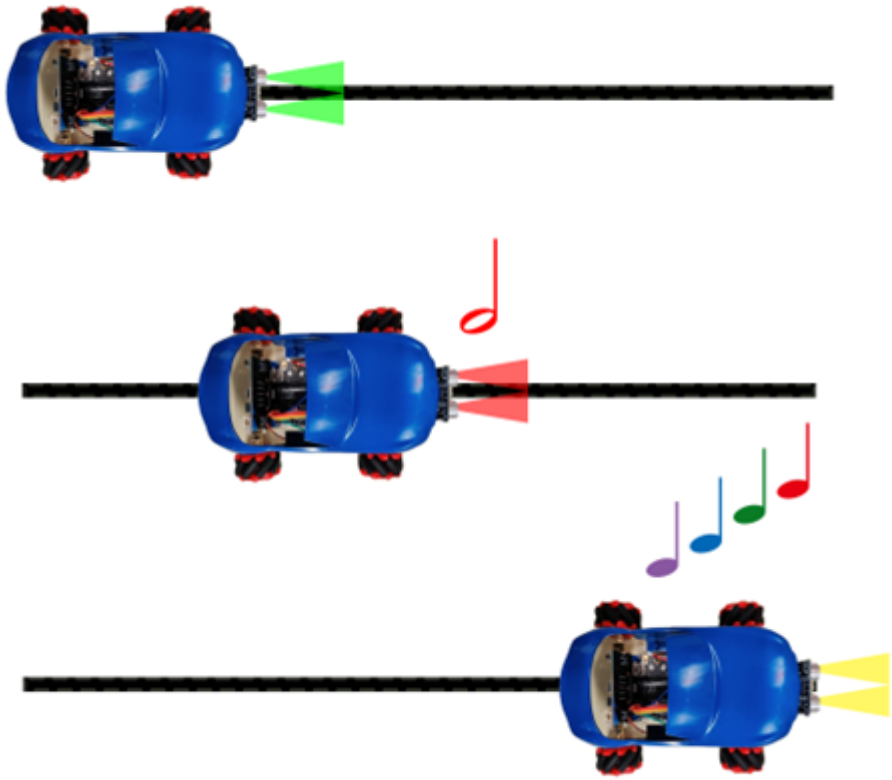
- X 坐標範圍：0 - 320
- Y 坐標範圍：0 - 240



練習 3



練習 4：巡Ⓕ自動車

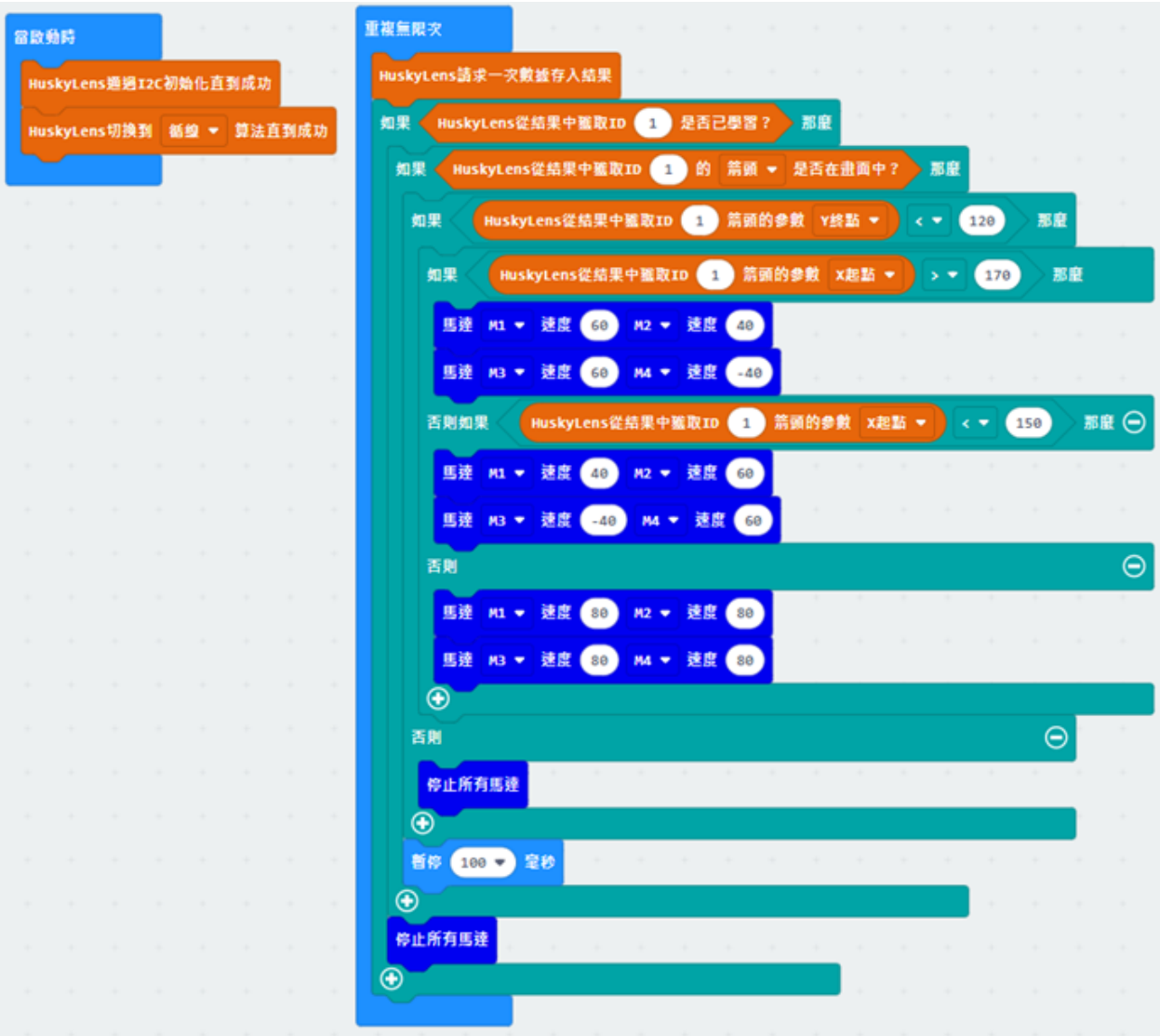


答案

練習 1

當啟動時
HuskyLens通過I2C初始化直到成功
HuskyLens切換到 循線 算法直到成功
重複無限次
HuskyLens請求一次數據存入結果
顯示 數字 HuskyLens從結果中獲取靠近中心箭頭的 ID 參數

練習 2


當啟動時
HuskyLens通過I2C初始化直到成功
HuskyLens切換到 循線 算法直到成功
重複無限次
HuskyLens請求一次數並存入結果
如果 HuskyLens從結果中獲取ID 1 是否已學習？ 那麼
如果 HuskyLens從結果中獲取ID 1 的 箭頭 是否在畫面中？ 那麼
如果 HuskyLens從結果中獲取ID 1 箭頭的參數 Y終點 < 120 那麼
如果 HuskyLens從結果中獲取ID 1 箭頭的參數 X起點 > 170 那麼
馬達 M1 速度 60 M2 速度 40
馬達 M3 速度 60 M4 速度 -40
否則如果 HuskyLens從結果中獲取ID 1 箭頭的參數 X起點 < 150 那麼
馬達 M1 速度 40 M2 速度 60
馬達 M3 速度 -40 M4 速度 60
否則
馬達 M1 速度 80 M2 速度 80
馬達 M3 速度 80 M4 速度 80
否則
停止所有馬達
暫停 100 毫秒
停止所有馬達

練習 3

程序

重複無限次
HuskyLens請求一次數據存入結果
如果 HuskyLens從結果中獲取ID 1 是否已學習？ 那麼
如果 HuskyLens從結果中獲取ID 1 的 箭頭 是否在畫面中？ 那麼
變數 X start 設為 HuskyLens從結果中獲取ID 1 箭頭的參數 X起點
呼叫 lineFollow
否則
停止所有馬達
定義函式 lineFollow
如果 HuskyLens從結果中獲取ID 1 箭頭的參數 Y終點 < 120 那麼
如果 X start ≥ 170 且 X start ≤ 190 那麼
馬達 M1 速度 50 M2 速度 -30
馬達 M3 速度 50 M4 速度 30
X起點在畫面偏右時,
車子向右慢轉
否則如果 X start > 190 那麼
馬達 M1 速度 60 M2 速度 -40
馬達 M3 速度 60 M4 速度 40
X起點在畫面右端時,
車子向右快轉
否則如果 X start ≤ 150 且 X start ≥ 130 那麼
馬達 M1 速度 -30 M2 速度 50
馬達 M3 速度 30 M4 速度 50
X起點在畫面偏左時,
車子向左慢轉
否則如果 X start < 130 那麼
馬達 M1 速度 -40 M2 速度 60
馬達 M3 速度 40 M4 速度 60
X起點在畫面左端時,
車子向左快轉
否則
馬達 M1 速度 80 M2 速度 80
馬達 M3 速度 80 M4 速度 80
X起點在畫面中間時,
車子前進
否則
停止所有馬達

練習 4

車子因探測到障礙或到達終點時，
停下的程序
當啟動時
HuskyLens通過I2C初始化直到成功
HuskyLens切換到 循線 算法直到成功
變數 played? 設為 "false"
預設未播放任何音效
定義函式 Stop
停止所有馬達
如果 RGB超聲波讀取距離(cm) ≤ 5 那麼
車子探測到障礙時
設置超聲波燈 全部 顏色 紅
如果 played? = "false" 那麼
變數為false時，播放一次音效
演奏 音階 低音 A# 持續 1 拍
變數 played? 設為 "true"
播完音效後，變數設爲true後
（不在播放）
如果 HuskyLens從結果中獲取ID 1 箭頭的參數 Y終點 ≥ 120 那麼
車子到達終點時
設置超聲波燈 全部 顏色 黃
（箭頭尖端在畫面下半）
如果 played? = "false" 那麼
演奏旋律 速度 120 (bpm)
變數 played? 設為 "true"

重複無限次
HuskyLens請求一次數據存入結果
如果 HuskyLens從結果中獲取ID 1 是否已學習？ 那麼
如果 HuskyLens從結果中獲取ID 1 的 箭頭 是否在畫面中？ 那麼
當車子未探測到障礙或未到
如果 HuskyLens從結果中獲取ID 1 箭頭的參數 Y終點 < 120 且
設置超聲波燈 全部 顏色 綠
變數 played? 設為 "false"
變數設定爲：音效未播放
如果 HuskyLens從結果中獲取ID 1 箭頭的參數 X起點 > 190 那麼
馬達 M1 速度 60 M2 速度 -40
車子往右走
馬達 M3 速度 60 M4 速度 40
否則如果 HuskyLens從結果中獲取ID 1 箭頭的參數 X起點 < 130 那麼
馬達 M1 速度 -40 M2 速度 60
車子往左走
馬達 M3 速度 40 M4 速度 60
否則

1.5.7 第七課

簡介

教學目標

HuskyLens 物件追Ⓕ功能（上）

物件追Ⓕ是什麼？

物件追Ⓕ的應用

- 事件偵測：可針對車流中的停止車輛、低速車輛、逆行車輛、行人及Ⓕ霧偵測等交通事件進行偵測，以便即時進行應變措施
- 指揮交通：透過偵測經過的車輛數量，測量停滯時間，Ⓕ將資訊提供給紅Ⓕ燈，以提升交叉路口行車管理效率

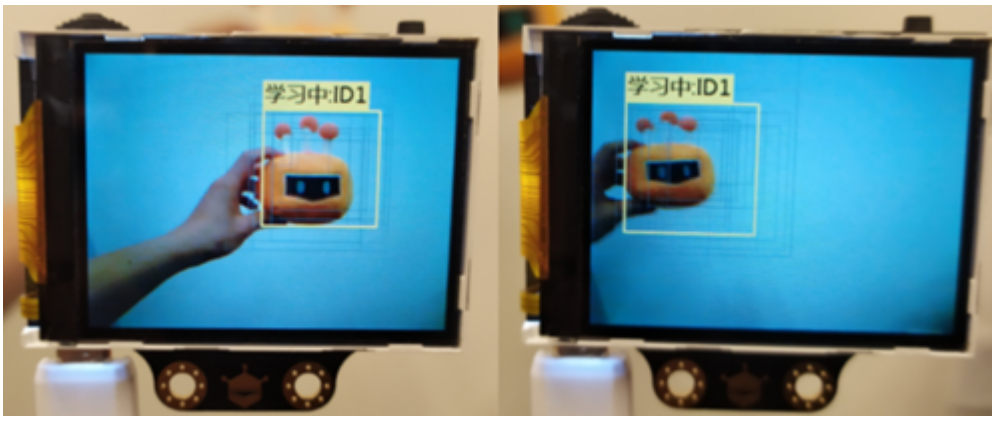
- 人機互動：透過追踪人類的動作，使機器學習識Ⓕ和理解人的動作、動作、人類
- Ⓕ擬實境：透過追踪與分析人體運動，在Ⓕ擬環境中模擬出Ⓕ擬角色的動作，可用於以捕捉玩家身體動作來進行的Ⓕ戲，如 Xbox Kinnect 和 VR Ⓕ戲

Huskylens 的物件追Ⓕ

而 HuskyLens 的巡Ⓕ追踪功能可以分爲學習物體和追踪物體兩個部分。

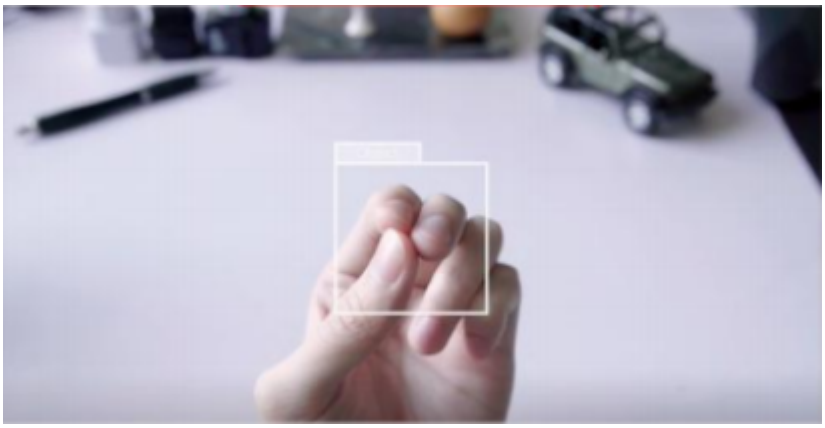
學習物體

追踪物體

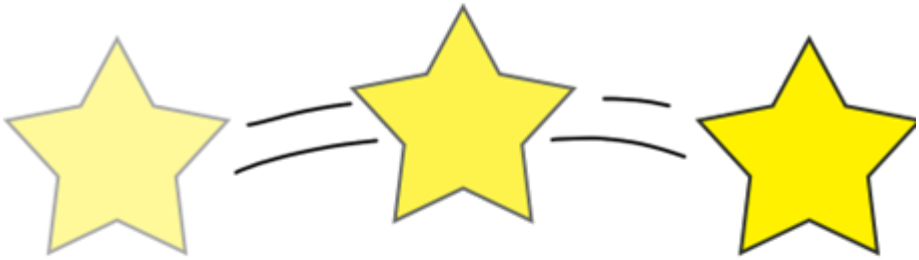


關閉同一時間進行追踪和學習的方法

練習 1



練習 2



- 使用“暫停”積木控制車子每次轉方向的時間

- 使用 X 中心代表物件位置

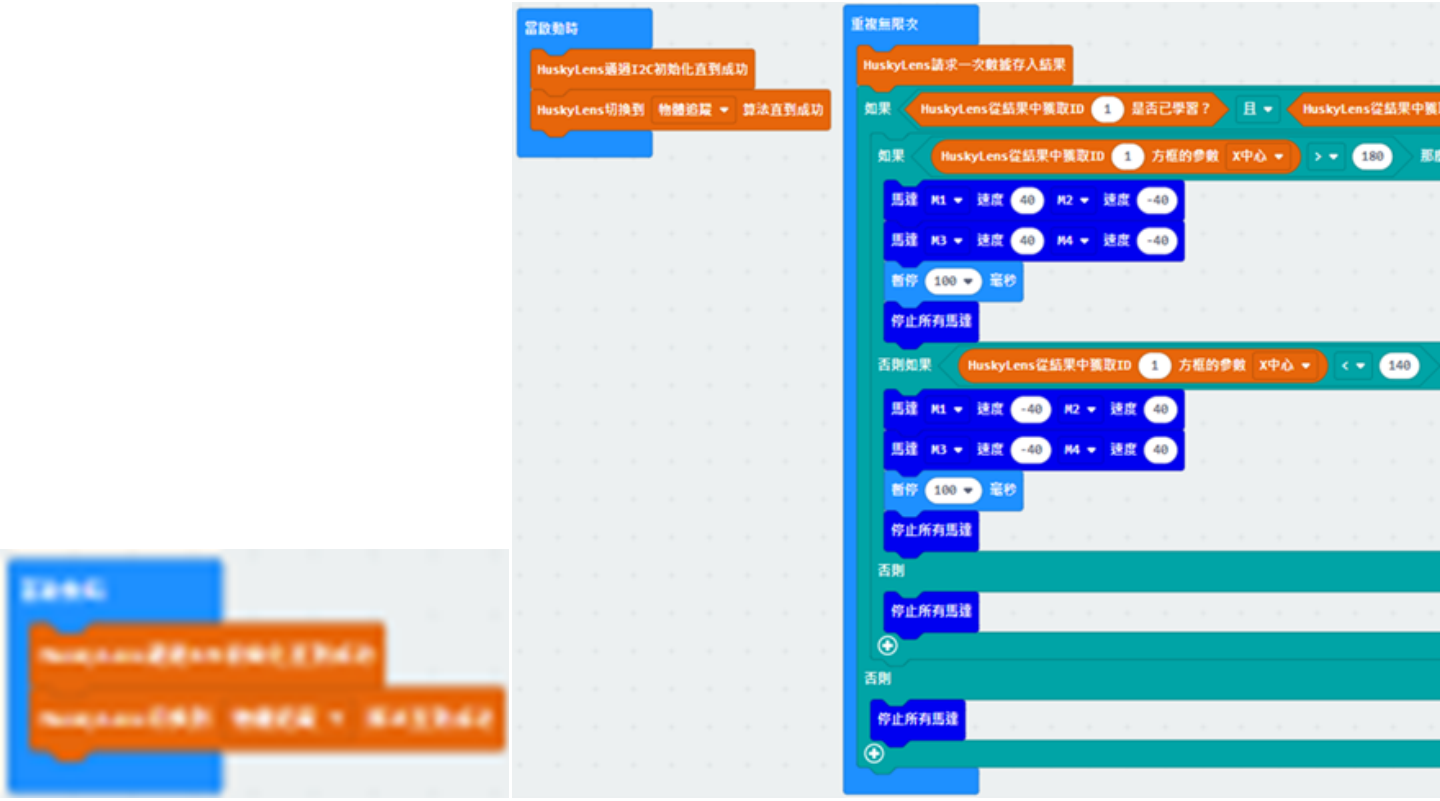
練習 3

- 我們平時如何靠眼睛知道自己是否靠近物件?
- HuskyLens 物件追F的學習框預設是 70*70 平方單位
- 在較遠的距離學習物件效果更佳

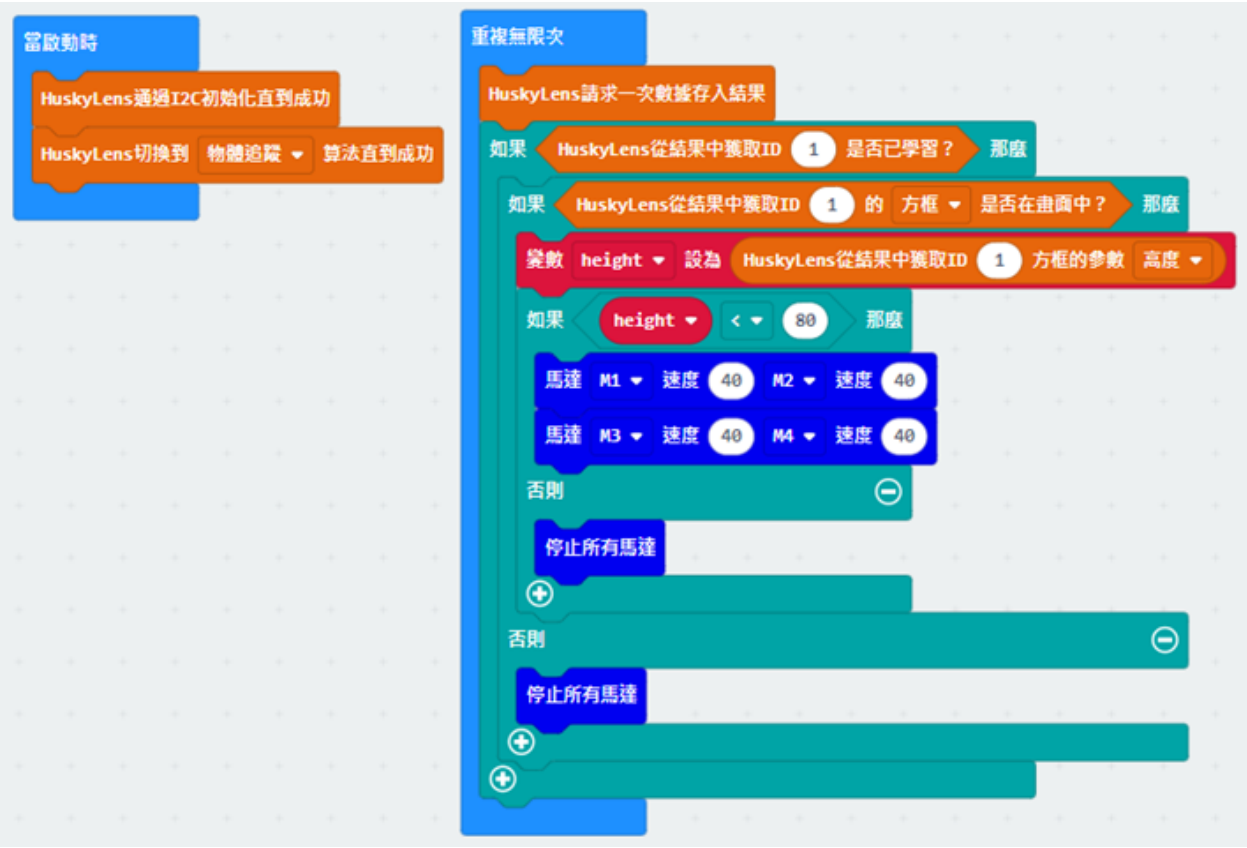
練習 4

答案

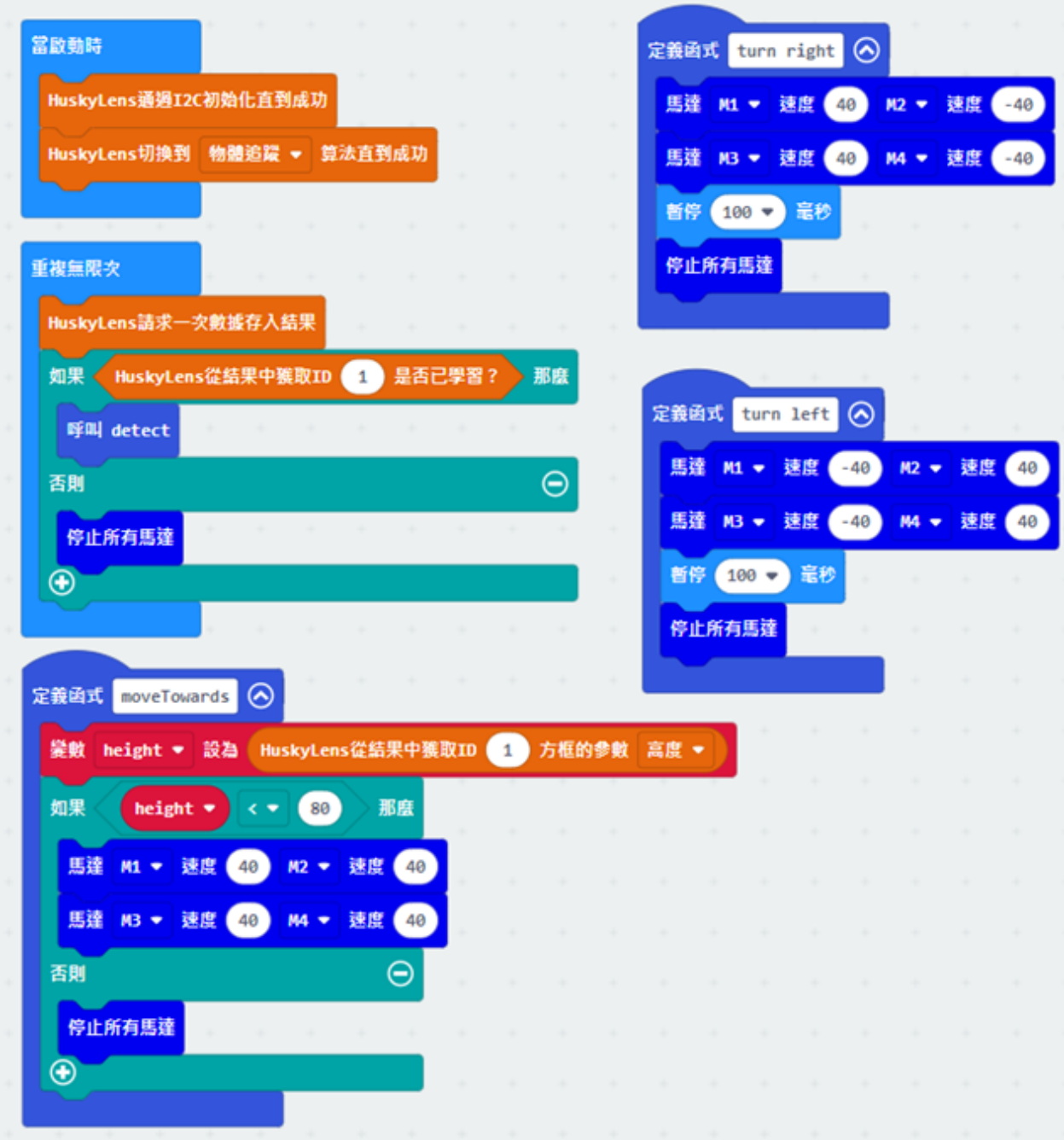
練習 2


當啟動時
HuskyLens通過I2C初始化直到成功
HuskyLens切換到 物體追蹤 算法直到成功
重複無限次
HuskyLens請求一次數據存入結果
如果 HuskyLens從結果中獲取ID 1 是否已學習？ 且
如果 HuskyLens從結果中獲取ID 1 方框的參數 X中心 > 180
馬達 M1 速度 40 M2 速度 -40
馬達 M3 速度 40 M4 速度 -40
暫停 100 毫秒
停止所有馬達
否則如果 HuskyLens從結果中獲取ID 1 方框的參數 X中心 < 140
馬達 M1 速度 -40 M2 速度 40
馬達 M3 速度 -40 M4 速度 40
暫停 100 毫秒
停止所有馬達
否則
停止所有馬達
否則
停止所有馬達

練習 3


當啟動時
HuskyLens通過I2C初始化直到成功
HuskyLens切換到 物體追蹤 算法直到成功
重複無限次
HuskyLens請求一次數據存入結果
如果 HuskyLens從結果中獲取ID 1 是否已學習？ 那麼
如果 HuskyLens從結果中獲取ID 1 的 方框 是否在畫面中？ 那麼
變數 height 設為 HuskyLens從結果中獲取ID 1 方框的參數 高度
如果 height < 80 那麼
馬達 M1 速度 40 M2 速度 40
馬達 M3 速度 40 M4 速度 40
否則
停止所有馬達
否則
停止所有馬達

練習 4


當啟動時
HuskyLens通過I2C初始化直到成功
HuskyLens切換到 物體追蹤 算法直到成功
重複無限次
HuskyLens請求一次數據存入結果
如果 HuskyLens從結果中獲取ID 1 是否已學習？ 那麼
呼叫 detect
否則
停止所有馬達
定義函式 turn right
馬達 M1 速度 40 M2 速度 -40
馬達 M3 速度 40 M4 速度 -40
暫停 100 毫秒
停止所有馬達
定義函式 turn left
馬達 M1 速度 -40 M2 速度 40
馬達 M3 速度 -40 M4 速度 40
暫停 100 毫秒
停止所有馬達
定義函式 moveTowards
變數 height 設為 HuskyLens從結果中獲取ID 1 方框的參數 高度
如果 height < 80 那麼
馬達 M1 速度 40 M2 速度 40
馬達 M3 速度 40 M4 速度 40
否則
停止所有馬達

定義函式 detect
如果 HuskyLens從結果中獲取ID 1 的 方框 是否在畫面中？ 那麼
如果 HuskyLens從結果中獲取ID 1 方框的參數 X中心 > 180 那麼
呼叫 turn right
否則如果 HuskyLens從結果中獲取ID 1 方框的參數 X中心 < 140 那麼
呼叫 turn left

1.5.8 第八課

簡介

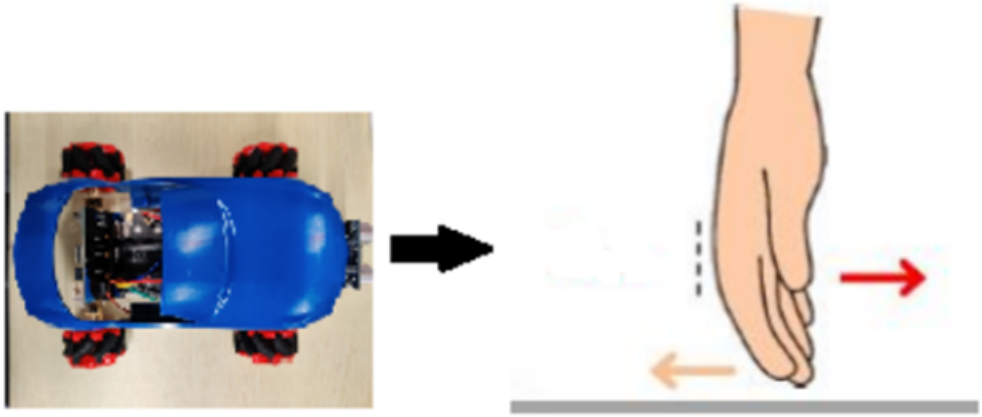
教學目標

物件追Ⓕ功能（下）

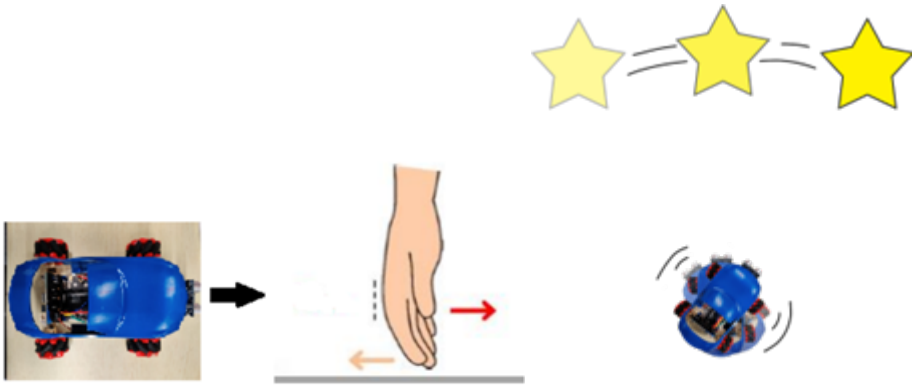
HuskyLens 物件追Ⓕ進階應用

探測障礙：HuskyLens 物件追Ⓕ v.s. 超聲波傳感器

練習 1



練習 2



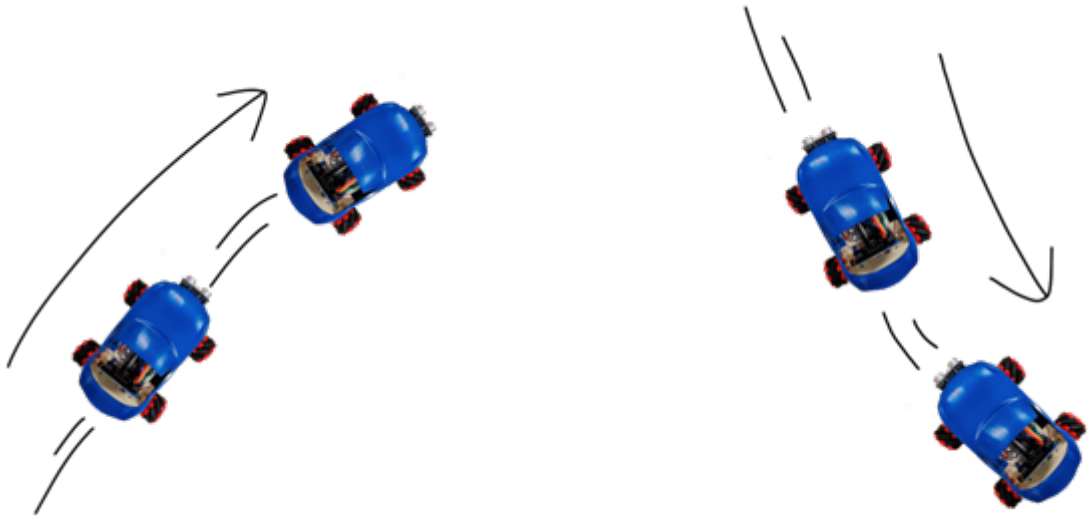
- 可參考第 6 課練習 2 將畫面劃分成三個區域
- 先使用調節車子方向，再繼續前進

練習 3

物件在不同位置時的車子動作

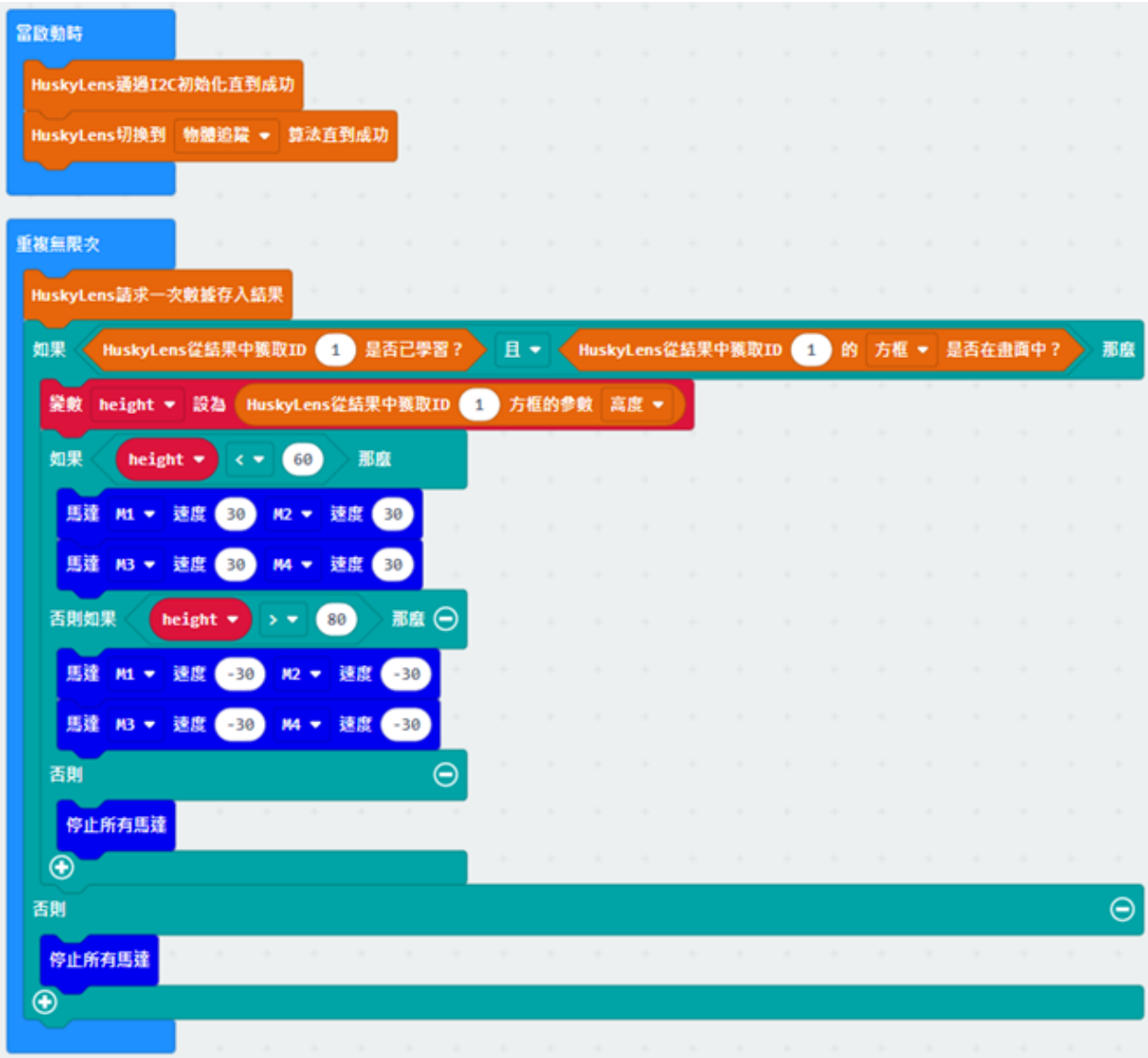
物件距離	物件在畫面位置			超聲波燈顏色
	靠左	置中	靠右	
遙遠				綠
適中				黃
靠近				紅

挑戰



答案

練習 1


當啟動時
HuskyLens通過I2C初始化直到成功
HuskyLens切換到 物體追蹤 算法直到成功
重複無限次
HuskyLens請求一次數據存入結果
如果 HuskyLens從結果中獲取ID 1 是否已學習？ 且 HuskyLens從結果中獲取ID 1 的 方框 是否在畫面中？ 那麼
變數 height 設為 HuskyLens從結果中獲取ID 1 方框的參數 高度
如果 height < 60 那麼
馬達 M1 速度 30 M2 速度 30
馬達 M3 速度 30 M4 速度 30
否則如果 height > 80 那麼
馬達 M1 速度 -30 M2 速度 -30
馬達 M3 速度 -30 M4 速度 -30
否則
停止所有馬達
否則
停止所有馬達

練習 2

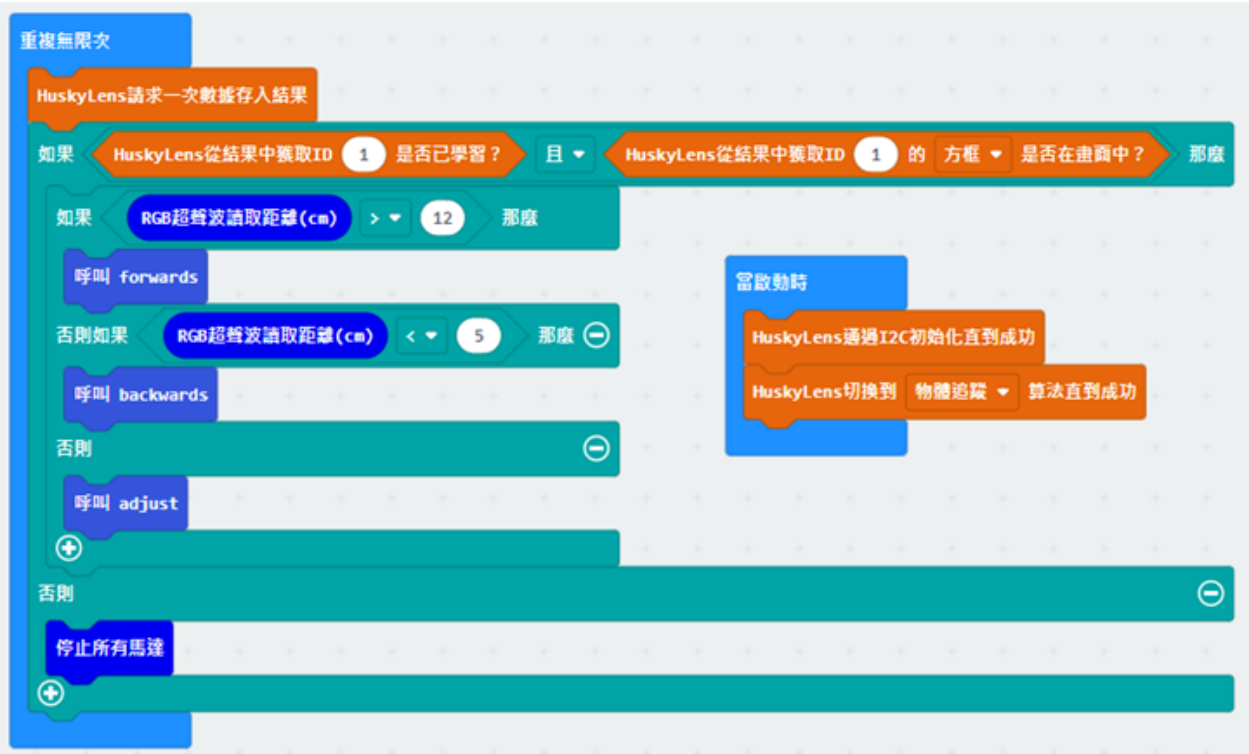
重複無限次
HuskyLens請求一次數據存入結果
如果 HuskyLens從結果中獲取ID 1 是否已學習？ 且 HuskyLens從結果中獲取ID 1 的 方框 是否在畫面中？ 那麼
如果 HuskyLens從結果中獲取ID 1 方框的參數 X中心 > 190 或 HuskyLens從結果中獲取ID 1 方框的參數 X中心 < 130 那麼
如果 HuskyLens從結果中獲取ID 1 方框的參數 X中心 > 190 那麼
馬達 M1 速度 40 M2 速度 -40
馬達 M3 速度 40 M4 速度 -40
否則
馬達 M1 速度 -40 M2 速度 40
馬達 M3 速度 -40 M4 速度 40
暫停 100 毫秒
停止所有馬達
否則
呼叫 ultrasonic
否則
停止所有馬達

定義函式 ultrasonic
如果 RGB超聲波讀取距離(cm) < 3 那麼
馬達 M1 速度 -40 M2 速度 -40
馬達 M3 速度 -40 M4 速度 -40
否則如果 RGB超聲波讀取距離(cm) > 12 那麼
馬達 M1 速度 40 M2 速度 40
馬達 M3 速度 40 M4 速度 40
否則
停止所有馬達
當啟動時
HuskyLens通過I2C初始化直到成功
HuskyLens切換到 物體追蹤 算法直到成功

練習 3

- 物件體積較大，F且只對物件其中一處的特徵進行學習。（確保傳感器能接收反彈回來的信號）
- 使用 HuskyLens 學習時必須要包含充足數量的角度及距離。（由於鏡頭在支架上的角度固定，而在不同角度、距離看見的物件樣子都存在分F，必須確保 HuskyLens 在不同位置都能辨識到物件）

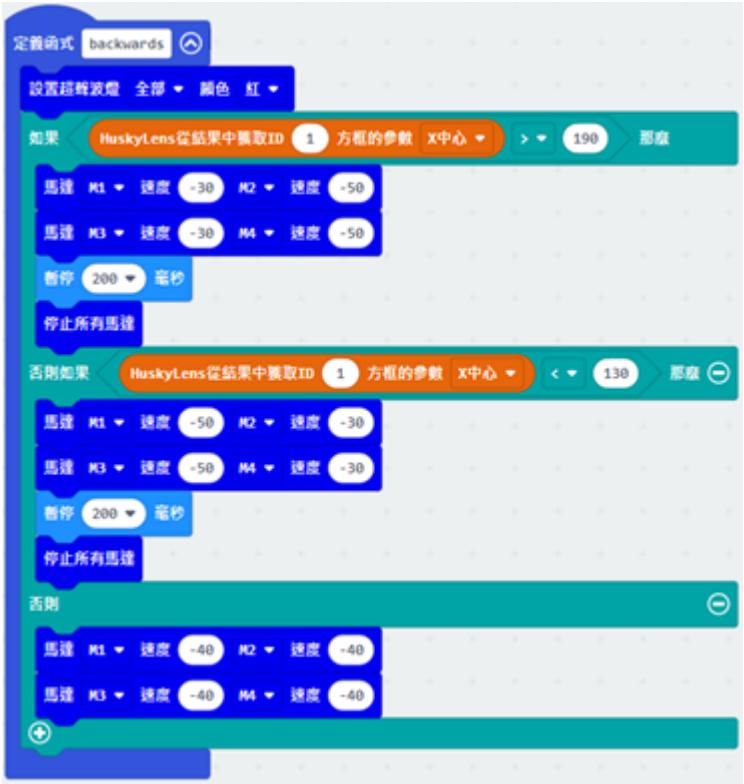
程序無限次執行



向前移動

定義函式 forwards
設置超聲波燈 全部 顏色 綠
如果 HuskyLens從結果中獲取ID 1 方框的參數 X中心 > 190 那麼
馬達 M1 速度 50 M2 速度 30
馬達 M3 速度 50 M4 速度 30
暫停 200 毫秒
停止所有馬達
否則如果 HuskyLens從結果中獲取ID 1 方框的參數 X中心 < 130 那麼
馬達 M1 速度 30 M2 速度 50
馬達 M3 速度 30 M4 速度 50
暫停 200 毫秒
停止所有馬達
否則
馬達 M1 速度 40 M2 速度 40
馬達 M3 速度 40 M4 速度 40

向後移動


定義函式 backwards
設置超級波燈 全部 顏色 紅
如果 HuskyLens從結果中獲取ID 1 方框的參數 X中心 > 190 那麼
馬達 M1 速度 -30 M2 速度 -50
馬達 M3 速度 -30 M4 速度 -50
暫停 200 毫秒
停止所有馬達
否則如果 HuskyLens從結果中獲取ID 1 方框的參數 X中心 < 130 那麼
馬達 M1 速度 -50 M2 速度 -30
馬達 M3 速度 -50 M4 速度 -30
暫停 200 毫秒
停止所有馬達
否則
馬達 M1 速度 -40 M2 速度 -40
馬達 M3 速度 -40 M4 速度 -40

原地移動

定義函式 adjust
設置超聲波燈 全部 顏色 黃
如果 HuskyLens從結果中獲取ID 1 方框的參數 X中心 > 190 那麼
馬達 M1 速度 40 M2 速度 -40
馬達 M3 速度 40 M4 速度 -40
暫停 200 毫秒
停止所有馬達
否則如果 HuskyLens從結果中獲取ID 1 方框的參數 X中心 < 130 那麼
馬達 M1 速度 -40 M2 速度 40
馬達 M3 速度 -40 M4 速度 40
暫停 200 毫秒
停止所有馬達
否則
停止所有馬達

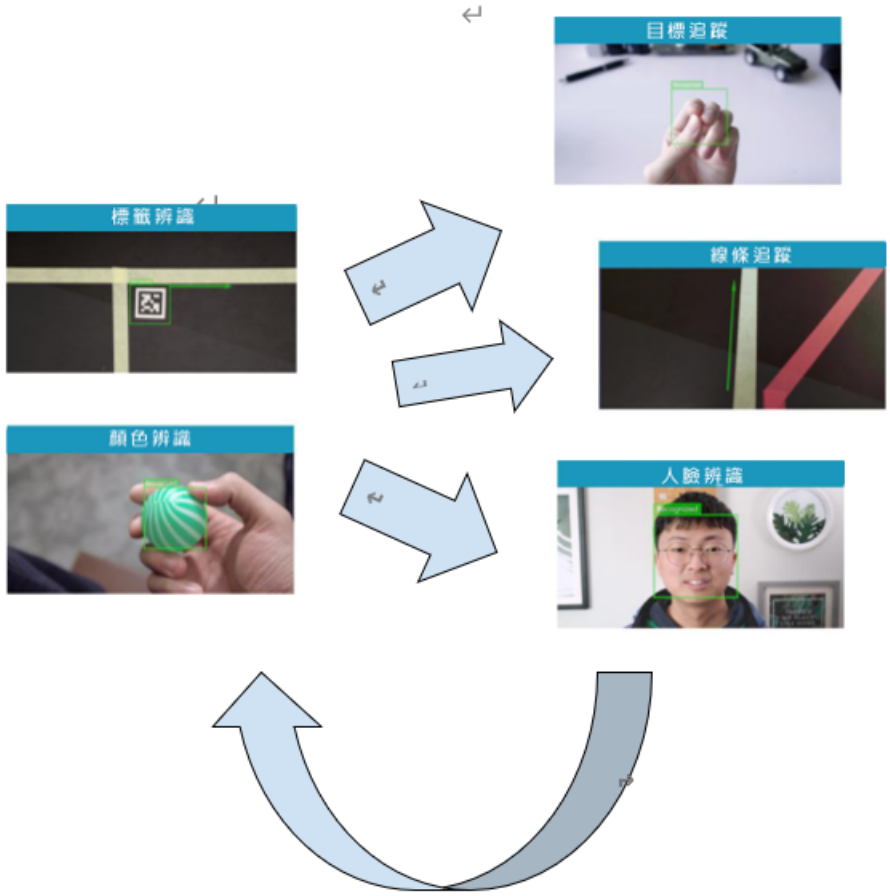
1.5.9 第九課

簡介

教學目標

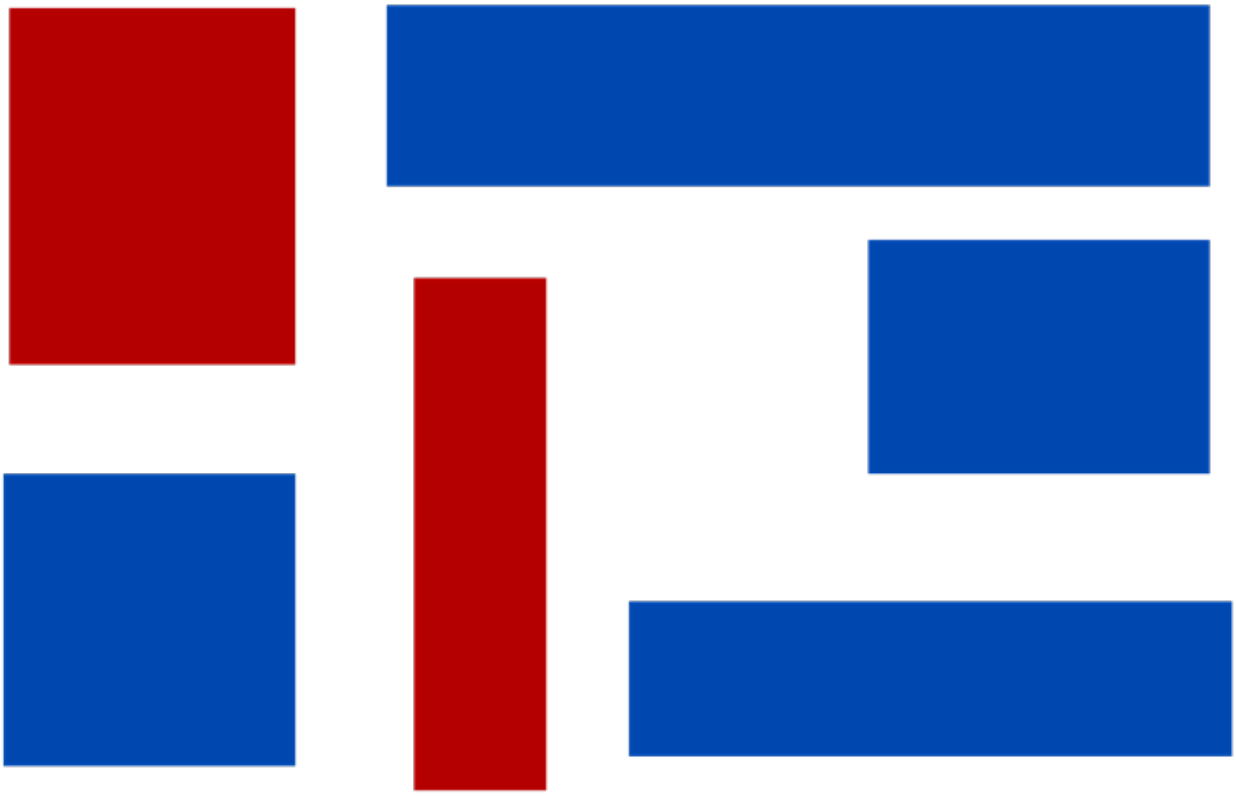
Micro:bit AI 智能小車

HuskyLens 進階應用

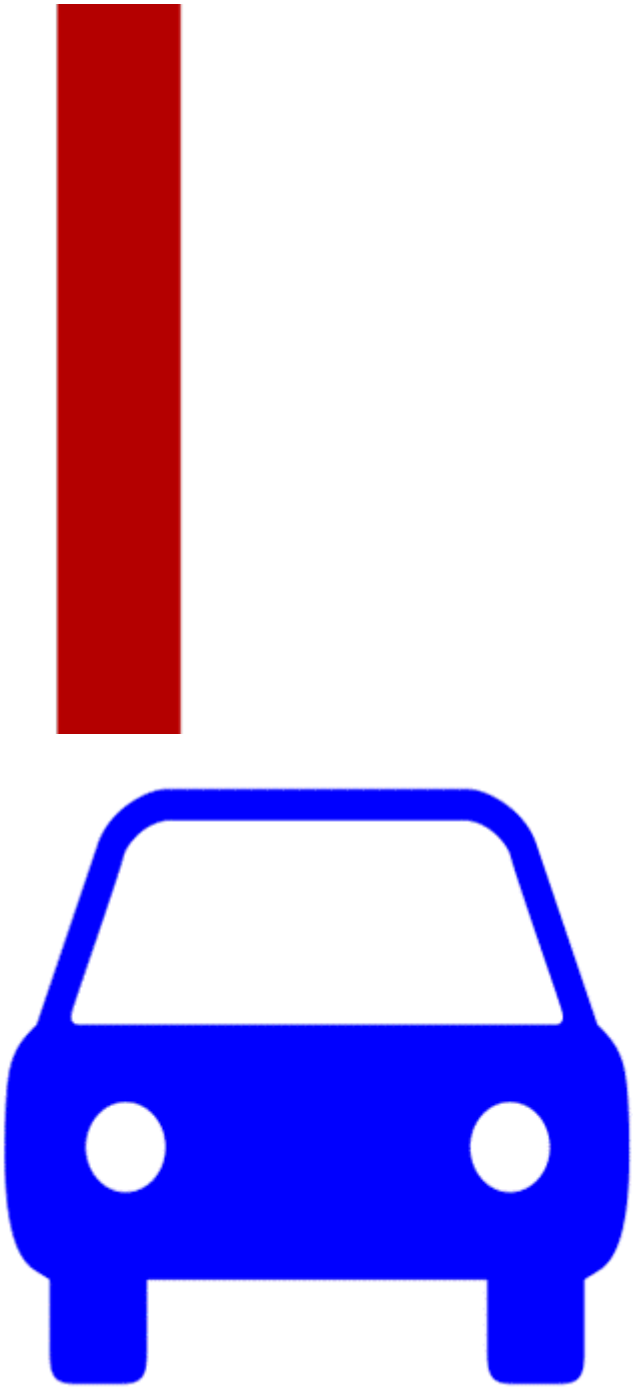


練習 1：功能組合：顏色辨認 + 巡F

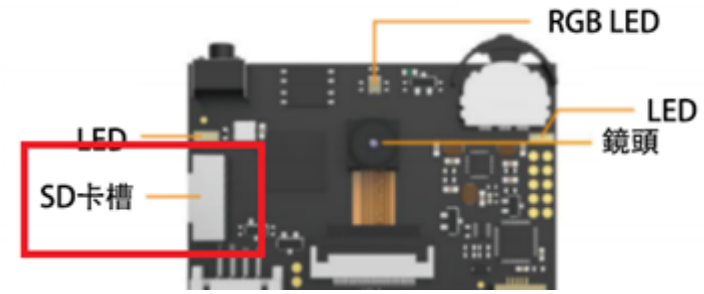
- 如巡F容易出界可使用暫停積木令車子每次移動後剎停
- 在按下 A 按鈕轉F到顏色辨認模式後發送兩次數據存入結果請求
- 第一次將數據進行初始化，第二次將數據更新化F新的學習結果
- 建立變數 id 儲存方塊數量最多的顏色 ID
- 建立變數 mode 儲存 HuskyLens 當下的模式



練習 2：以不同標🅵代表不同 HuskyLens 模式



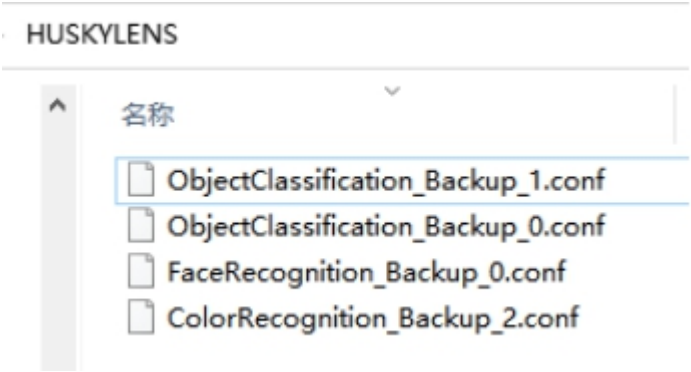
SD 卡保存/加載模型



方式一：在各算法功能的二級菜單目F下手動操作

1. 學習新物體：讓 HuskyLens 學習幾個新的物體，每個物體有不同的 ID 號。
2. 導出模型：長按“功能按鍵”，進入物體分類功能的二級菜單參數設置界面，然後向右撥動“功能按鍵”至選擇“導出到 SD 卡”(即保存模型到 SD 卡)。短按“功能按鍵”，左右撥動“功能按鍵”，選擇 0~4 任意一個即可(相當於選擇保存的位置，方便區分不同的模型，一個模式最多有 5 個模型)，然後短按“功能按鍵”即可保存模型。

3. 模型導出到 SD 卡後，用電腦讀取該 SD 卡，在“HUSKYLENS”文件夾下即可看到後綴F.conf 的模型數據文件，不同算法保存的文件名稱不同。如下圖所示：

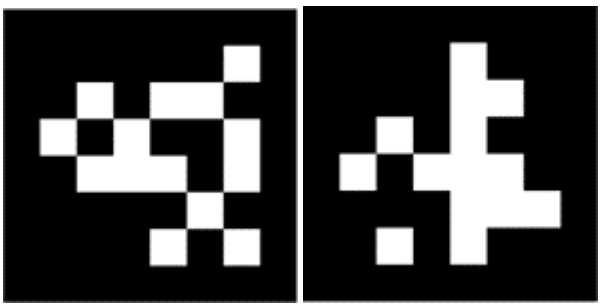


方式二：使用積木模塊操作

拍照、截屏保存到 SD 卡

練習 3

練習 4：先將物件進行分類，再切Ⓕ至相應模式 + 截屏



答案

練習 1

當按鈕 A 被按下
HuskyLens切換到 顏色辨識 算法直到成功
HuskyLens請求一次數據存入結果
初始化上次顏色學習結果
HuskyLens請求一次數據存入結果
儲存新的學習結果
顯示 數字 HuskyLens從結果中獲取已學ID總數
如果 HuskyLens從結果中獲取已學ID總數 = 2 那麼
如果 HuskyLens從結果中獲取ID 1 的 方框 是否在畫面中？ 且 HuskyLens從結果中獲取ID 2 的 方框 是否在畫面中？ 那麼
如果 HuskyLens從結果中獲取ID 1 方框 總數 > HuskyLens從結果中獲取ID 2 方框 總數 那麼
比較色塊數量
變數 id 設為 1
否則
變數 id 設為 2
暫停 3000 毫秒
如果 id ≠ 0 那麼
變數 mode 設為 "line mode"
HuskyLens切換到 循線 算法直到成功
當啟動時
變數 mode 設為 "color mode"
變數 id 設為 0
HuskyLens透過I2C初始化直到成功
HuskyLens切換到 顏色辨識 算法直到成功
（排除了沒有識別到任何顏色和兩種顏色方塊數量相約的情況）
當id是1或2時才切換至巡線模式

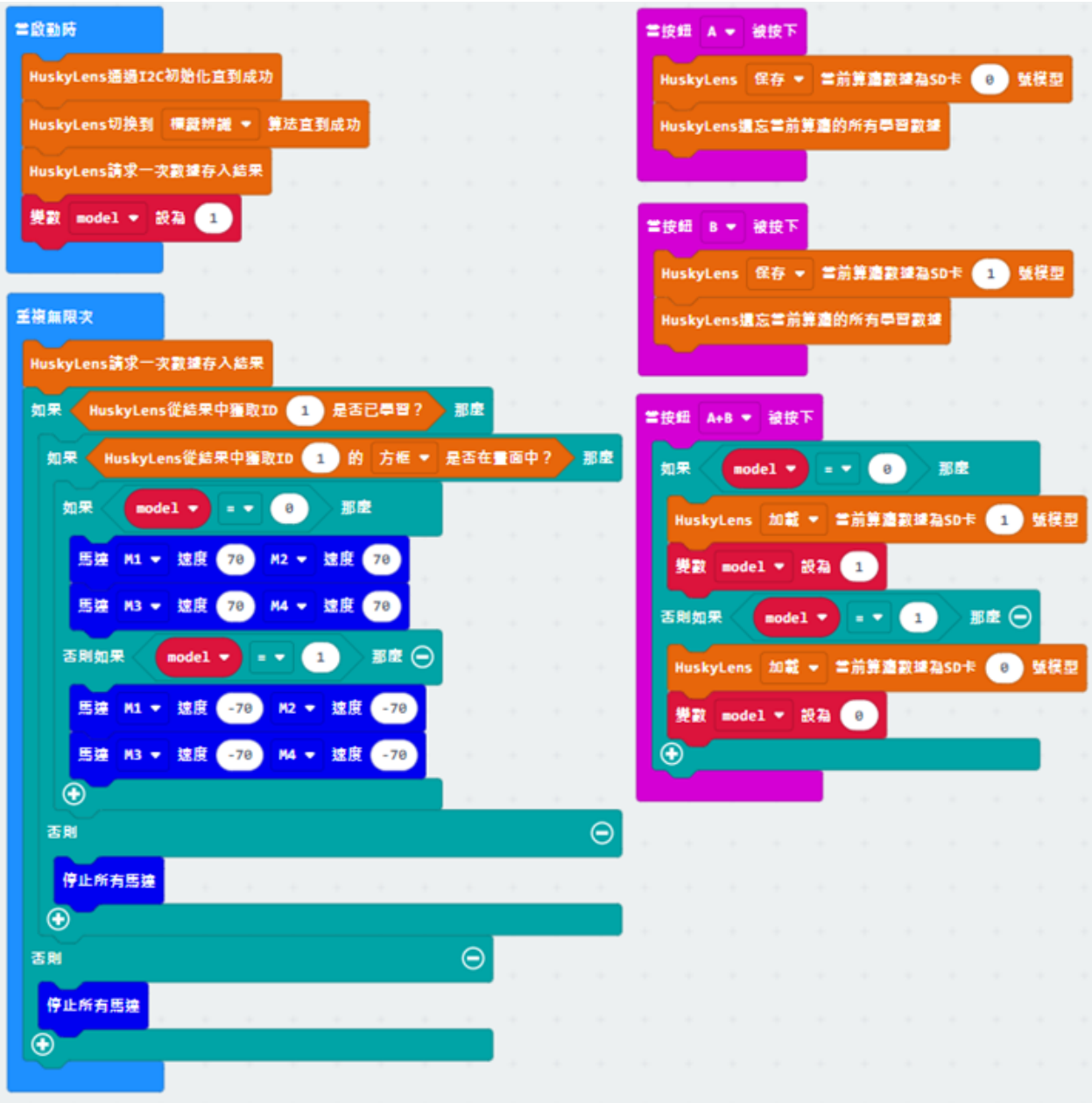
重複無限次
如果 mode = "line mode" 那麼
車子只在巡線模式時執行的程序
HuskyLens請求一次數據存入結果
如果 HuskyLens從結果中獲取ID id 箭頭的參數 Y終點 < 120 那麼
如果 HuskyLens從結果中獲取ID id 箭頭的參數 X起點 < 130 那麼
馬達 M1 速度 40 M2 速度 60
馬達 M3 速度 -40 M4 速度 60
否則如果 HuskyLens從結果中獲取ID id 箭頭的參數 X起點 > 190 那麼
馬達 M1 速度 60 M2 速度 40
馬達 M3 速度 60 M4 速度 -40
否則

練習 2

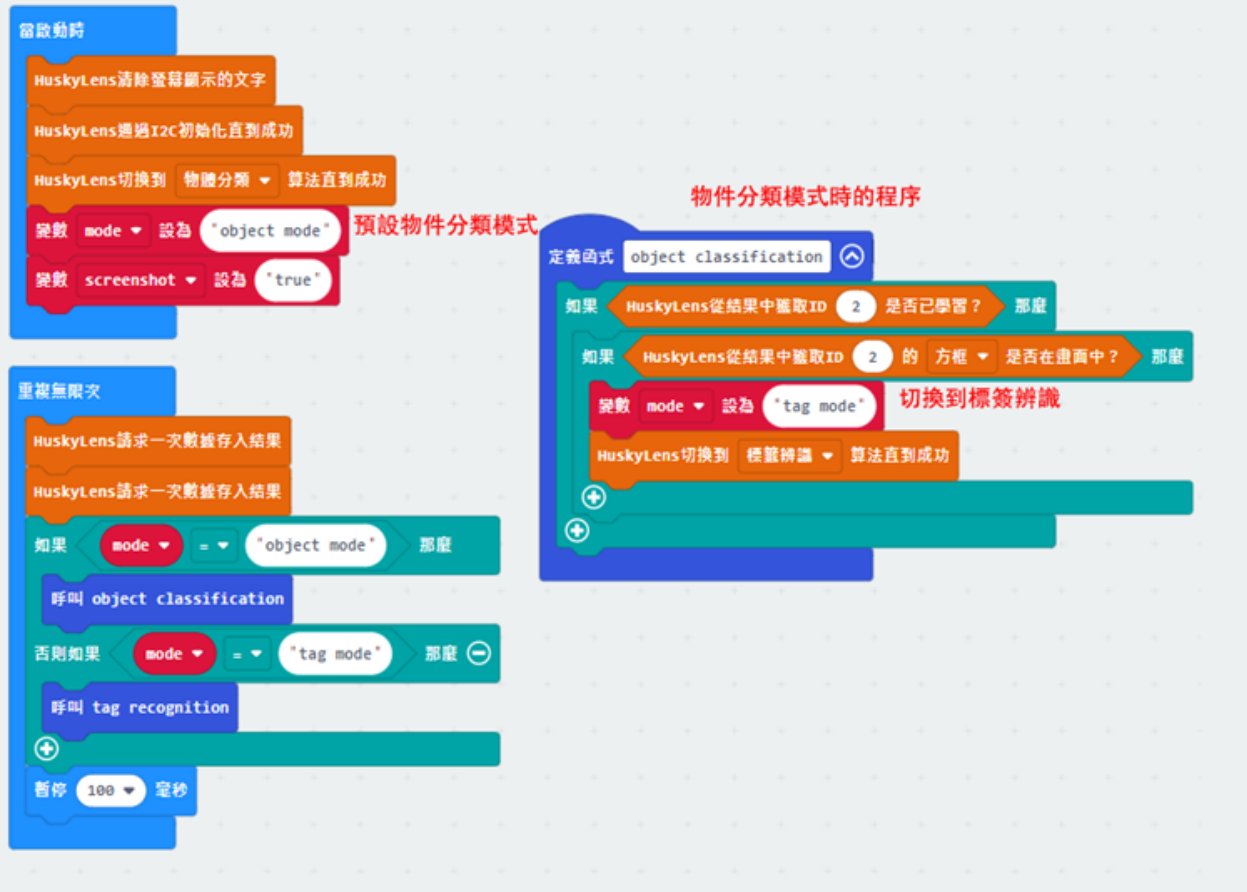
當啟動時
HuskyLens通過I2C初始化直到成功
HuskyLens切換到 標籤辨識 算法直到成功
HuskyLens請求一次數據存入結果
變數 mode 設為 0
當按鈕 B 被按下
停止所有馬達
變數 mode 設為 0
HuskyLens切換到 標籤辨識 算法直到成功
重複無限次
HuskyLens請求一次數據存入結果
呼叫 car_movement
當按鈕 A 被按下
如果 mode = 0 那麼
如果 HuskyLens從結果中獲取靠近中心方框的 ID 參數 = 1 那麼
變數 mode 設為 1
HuskyLens切換到 顏色辨識 算法直到成功
否則如果 HuskyLens從結果中獲取靠近中心方框的 ID 參數 = 2 那麼
變數 mode 設為 2
HuskyLens切換到 人臉辨識 算法直到成功
否則如果 HuskyLens從結果中獲取靠近中心方框的 ID 參數 = 3 那麼
變數 mode 設為 3
HuskyLens切換到 物體分類 算法直到成功
否則如果 HuskyLens從結果中獲取靠近中心方框的 ID 參數 = 4 那麼
變數 mode 設為 4
HuskyLens切換到 循線 算法直到成功

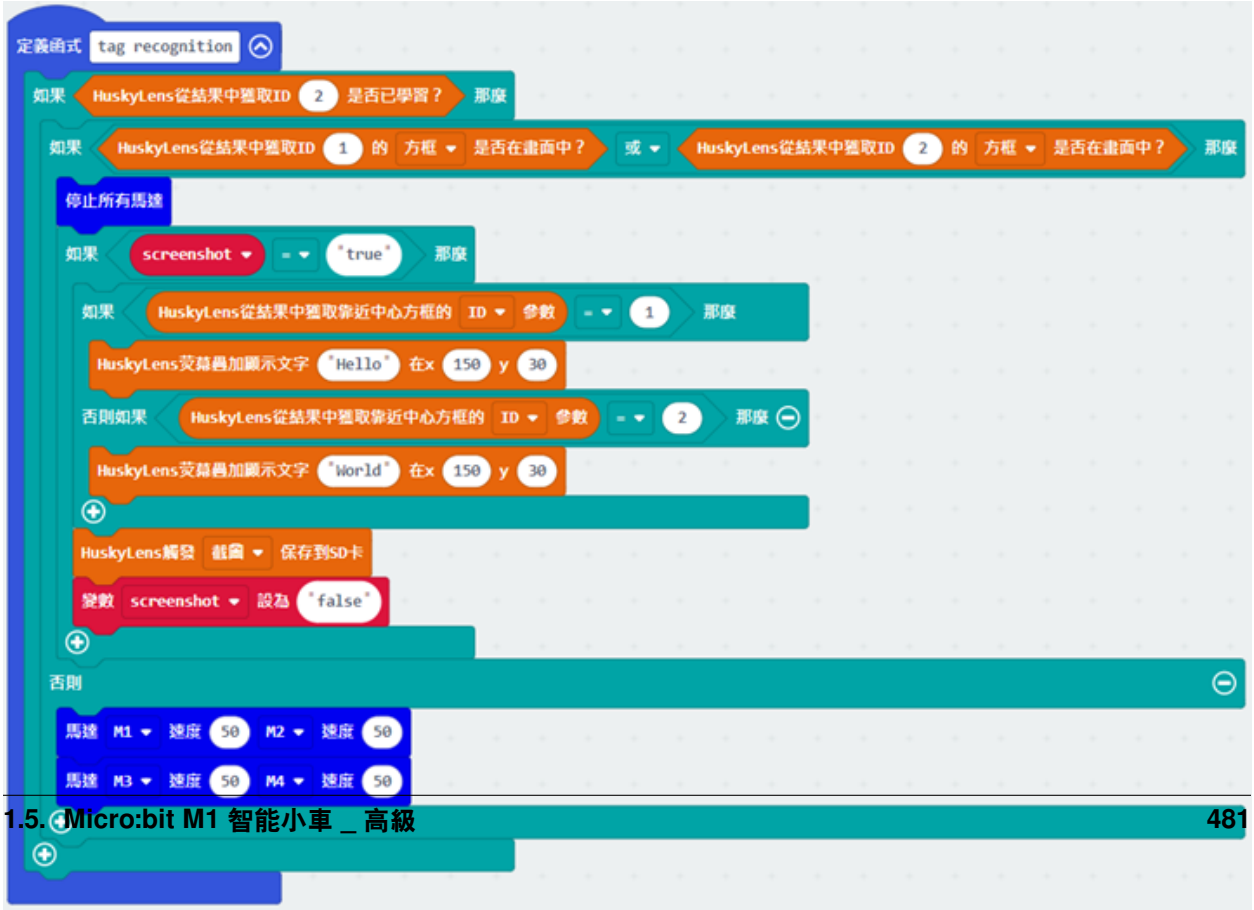
定義函式 car_movement
如果 mode ≠ 3 且 mode ≠ 4 那麼
如果 HuskyLens從結果中獲取ID 1 的 方框 是否在畫面中? 那麼
如果 mode = 1 那麼
顏色辨識
馬達 M1 速度 70 M2 速度 70
馬達 M3 速度 70 M4 速度 70
前進
否則如果 mode = 2 那麼
人臉辨識
馬達 M1 速度 -70 M2 速度 -70
馬達 M3 速度 -70 M4 速度 -70
後退
否則
停止所有馬達
否則如果 mode = 3 那麼
物件分類
如果 HuskyLens從結果中獲取ID 2 的 方框 是否在畫面中? 那麼
馬達 M1 速度 -70 M2 速度 70
馬達 M3 速度 70 M4 速度 -70
左移
否則
停止所有馬達
否則
循線
如果 HuskyLens從結果中獲取ID 1 的 箭頭 是否在畫面中? 那麼
馬達 M1 速度 70 M2 速度 -70
馬達 M3 速度 -70 M4 速度 70
右移
否則
停止所有馬達

練習 3


啟動時
HuskyLens通過I2C初始化直到成功
HuskyLens切換到 標籤辨識 算法直到成功
HuskyLens請求一次數據存入結果
變數 model 設為 1
重複無限次
HuskyLens請求一次數據存入結果
如果 HuskyLens從結果中獲取ID 1 是否已學習？ 那麼
如果 HuskyLens從結果中獲取ID 1 的 方框 是否在畫面中？ 那麼
如果 model = 0 那麼
馬達 M1 速度 70 M2 速度 70
馬達 M3 速度 70 M4 速度 70
否則如果 model = 1 那麼
馬達 M1 速度 -70 M2 速度 -70
馬達 M3 速度 -70 M4 速度 -70
否則
停止所有馬達
否則
停止所有馬達
按鈕 A 被按下
HuskyLens 保存 當前算法數據為SD卡 0 號模型
HuskyLens遺忘當前算法的所有學習數據
按鈕 B 被按下
HuskyLens 保存 當前算法數據為SD卡 1 號模型
HuskyLens遺忘當前算法的所有學習數據
按鈕 A+B 被按下
如果 model = 0 那麼
HuskyLens 加載 當前算法數據為SD卡 1 號模型
變數 model 設為 1
否則如果 model = 1 那麼
HuskyLens 加載 當前算法數據為SD卡 0 號模型
變數 model 設為 0

練習 4


當啟動時
HuskyLens清除螢幕顯示的文字
HuskyLens通過I2C初始化直到成功
HuskyLens切換到 物體分類 算法直到成功
變數 mode 設為 "object mode"
預設物件分類模式
變數 screenshot 設為 "true"
重複無限次
HuskyLens請求一次數據存入結果
HuskyLens請求一次數據存入結果
如果 mode = "object mode" 那麼
呼叫 object classification
否則如果 mode = "tag mode" 那麼
呼叫 tag recognition
暫停 100 毫秒
物件分類模式時的程序
定義函式 object classification
如果 HuskyLens從結果中獲取ID 2 是否已學習？ 那麼
如果 HuskyLens從結果中獲取ID 2 的 方框 是否在畫面中？ 那麼
變數 mode 設為 "tag mode"
切換到標籤辨識
HuskyLens切換到 標籤辨識 算法直到成功


定義函式 tag recognition
如果 HuskyLens從結果中獲取ID 2 是否已學習？ 那麼
如果 HuskyLens從結果中獲取ID 1 的 方框 是否在畫面中？ 或 HuskyLens從結果中獲取ID 2 的 方框 是否在畫面中？ 那麼
停止所有馬達
如果 screenshot = "true" 那麼
如果 HuskyLens從結果中獲取靠近中心方框的 ID 參數 = 1 那麼
HuskyLens螢幕疊加顯示文字 "Hello" 在x 150 y 30
否則如果 HuskyLens從結果中獲取靠近中心方框的 ID 參數 = 2 那麼
HuskyLens螢幕疊加顯示文字 "World" 在x 150 y 30
HuskyLens觸發 截圖 保存到SD卡
變數 screenshot 設為 "false"
否則
馬達 M1 速度 50 M2 速度 50
馬達 M3 速度 50 M4 速度 50

1.5.10 第十課

簡介

教學目標

總結 Micro:bit AI 智能小車

比賽一

- 人臉：micro:bit 板顯示笑容符號
- 顏色：亮起一個 RGB 板載燈或超聲波燈（共有六個，每個不同顏色）

```
1.      比賽F分數制，共有兩回合，以最高分回合的分數F最後分數
2.      當車子成功識F人臉，分數 +5
3.      每當車子成功識F一種顏色，分數 +2
```

(下页继续)

(繼續上一頁)

```
4.          如果所有 RGB 板載燈及超聲波燈成功亮起，分數額外加 +6
5.          車子如在限定時間內走不出終點，分數Ⓕ一半
```

比賽二

```
1.          比賽Ⓕ加分制，共有兩回合，以最高分回合的分數Ⓕ最後分數
2.          每當車子成功識Ⓕ一個標Ⓕ，分數 +2
3.          每當 HuskyLens 屏幕範圍的單字成功連成一個短句，分數額外 +5
4.          車子如在限定時間內走不出終點，分數Ⓕ一半
5.          最快完成的一組分數 +6，第二組則分數 +4，第三組則分數 +2
```