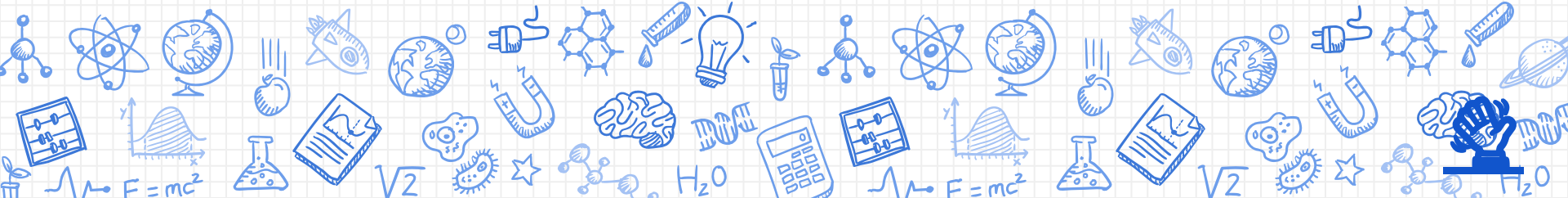
4060小車系統整合

桃園市陽明高中莊志忠

1. 4060小車材料
2. 4060小車組裝
3. 4060小車控制(blockly)
4. App 控制程式(App Inventor2)
5. App 控制程式(1071)

4060小車材料



4060 小車

- 3DP(三D列印機)版本, 搭載兩顆N20減速馬達, 250rpm(6V, 300轉), 全金屬齒輪, 體積小, 扭力大, 耐用不易磨損。
- 主要控制核心為Arduino Nano。
- 動力來源為 18650 鋰離子電池兩顆串連, 經由兩塊降壓板轉化成 5V、3.3V兩種電壓。

可適用課程: 4060小車基本款

(1)3D繪圖設計
(2)2D繪圖設計
(3)Scratch程式設計
(4)NKNUScratch馬達與感測器
(5)IDE程式設計
(6)4060循跡避障追熱智慧自走車

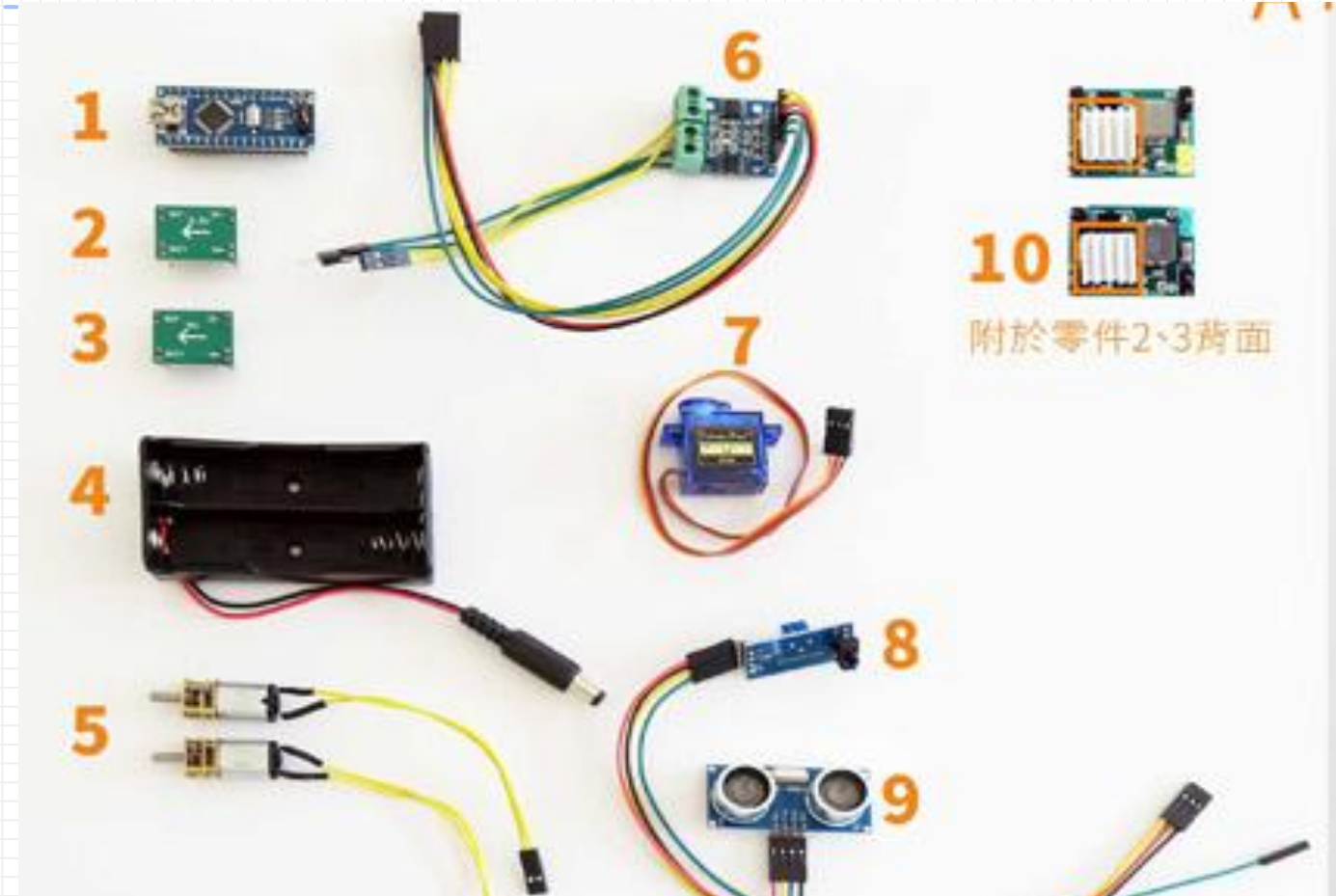
P.S. 車身檔案可至課程圖書館下載
或至雲端教室申請列印

4060 小車

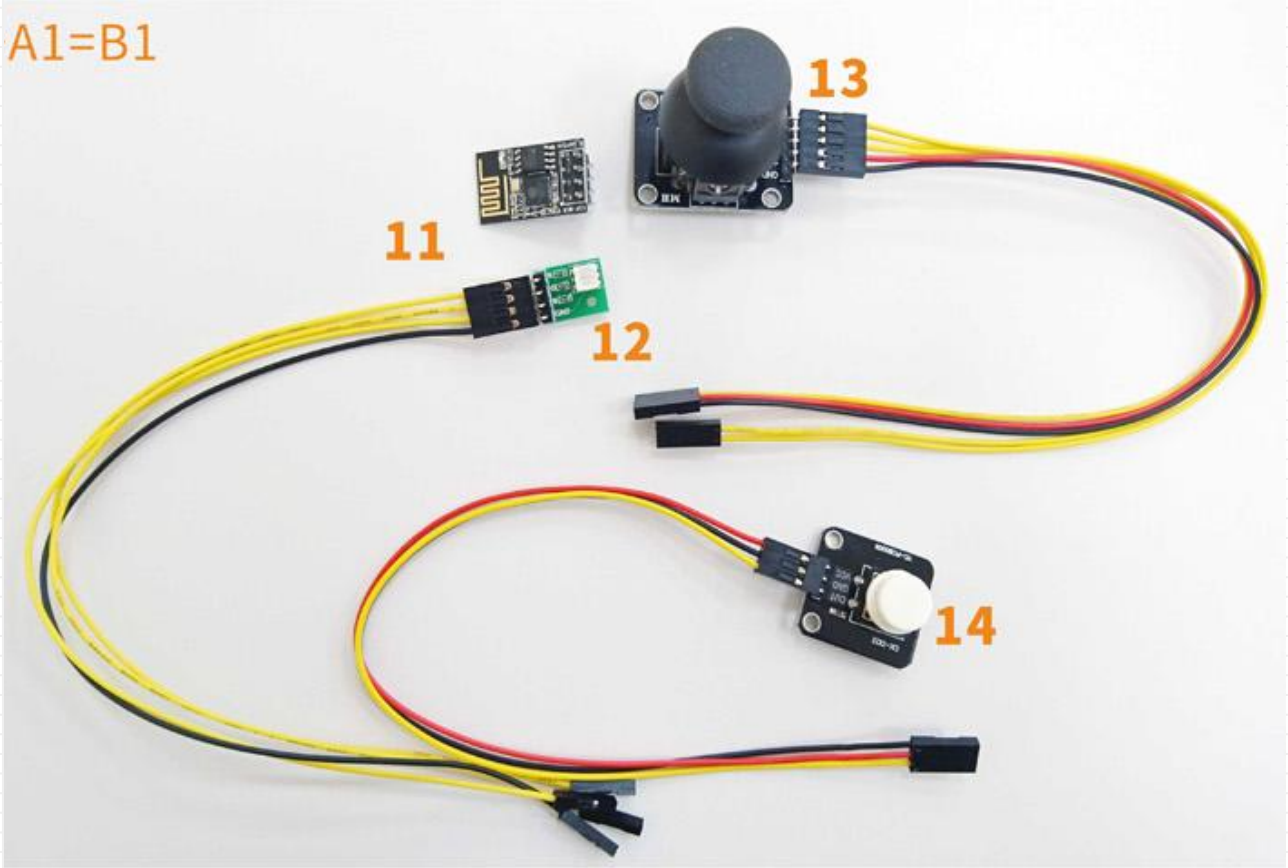
- B1 版零件清單

	零件模組品名
1	Arduino nano 開發板
2	DC-DC MP1584 電源模組　3.3V
3	DC-DC MP1584 電源模組 5V
4	電池盒 18650　2 節(含插頭)
5	減速馬達 GA12-N20 6V-300RPM+線組
6	直流步進電機驅動板 L9110S+線組
7	伺服馬達 SG90 9G
8	尋跡傳感器 TCRT5000 感測器+線組
9	超音波感測器 HC-SR04+線組
10	散熱片 9*9*5mm
11	ESP8266 ESP-01S WIFI 模組
12	RGB 全彩 LED 模組+線組
13	雙軸按鍵搖杆+線組
14	按鍵模組+線組

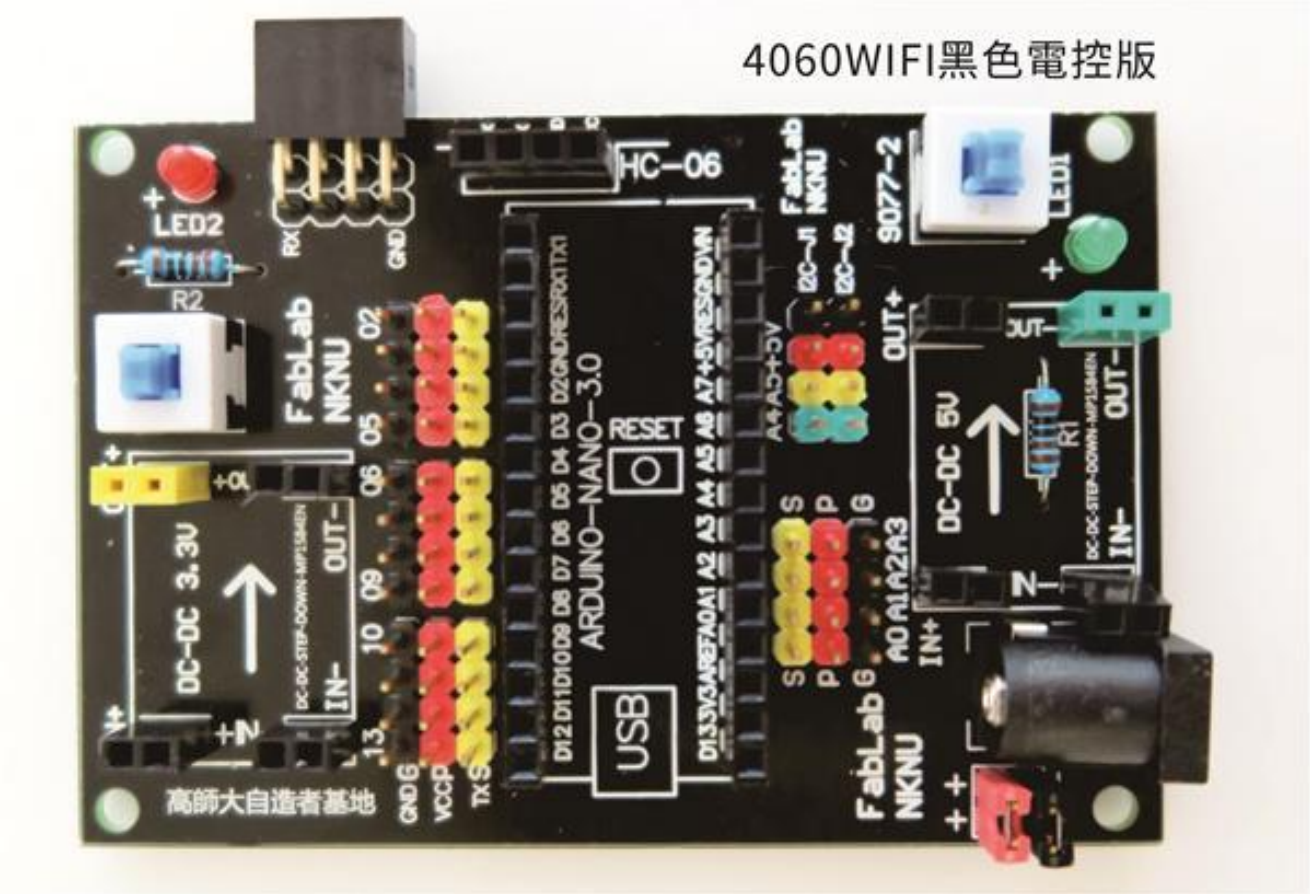
4060 零件圖片(1-10)



4060 零件圖片(11-14)

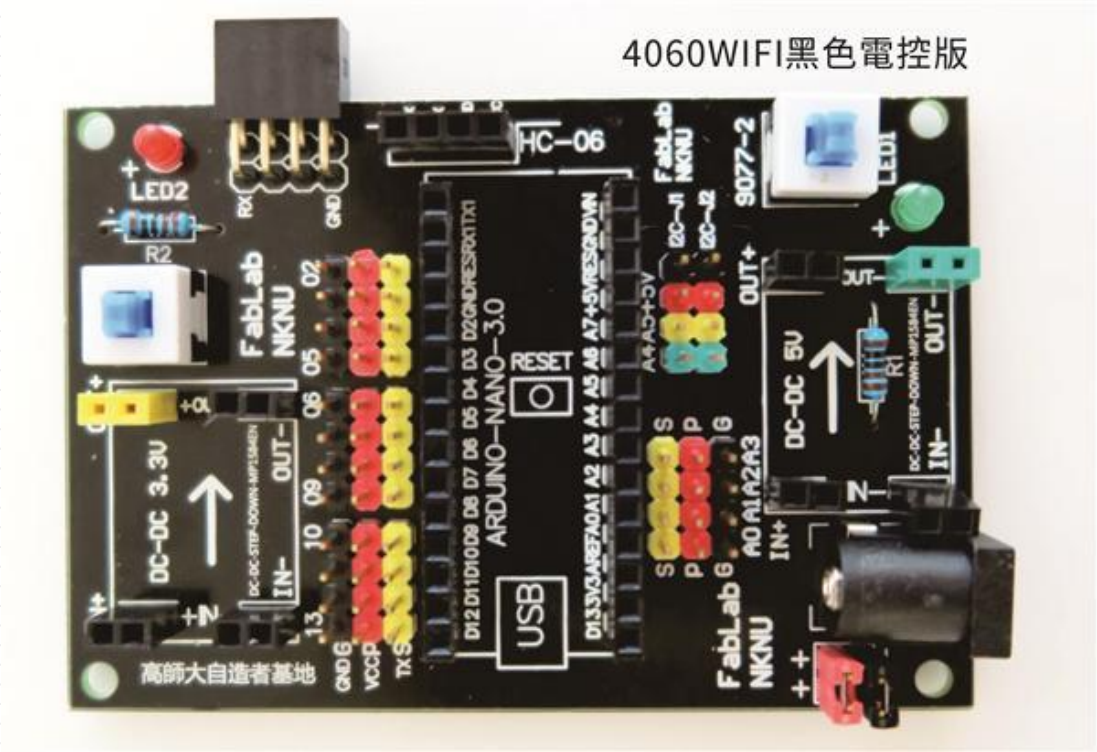


4060 WIFI 黑色電控板



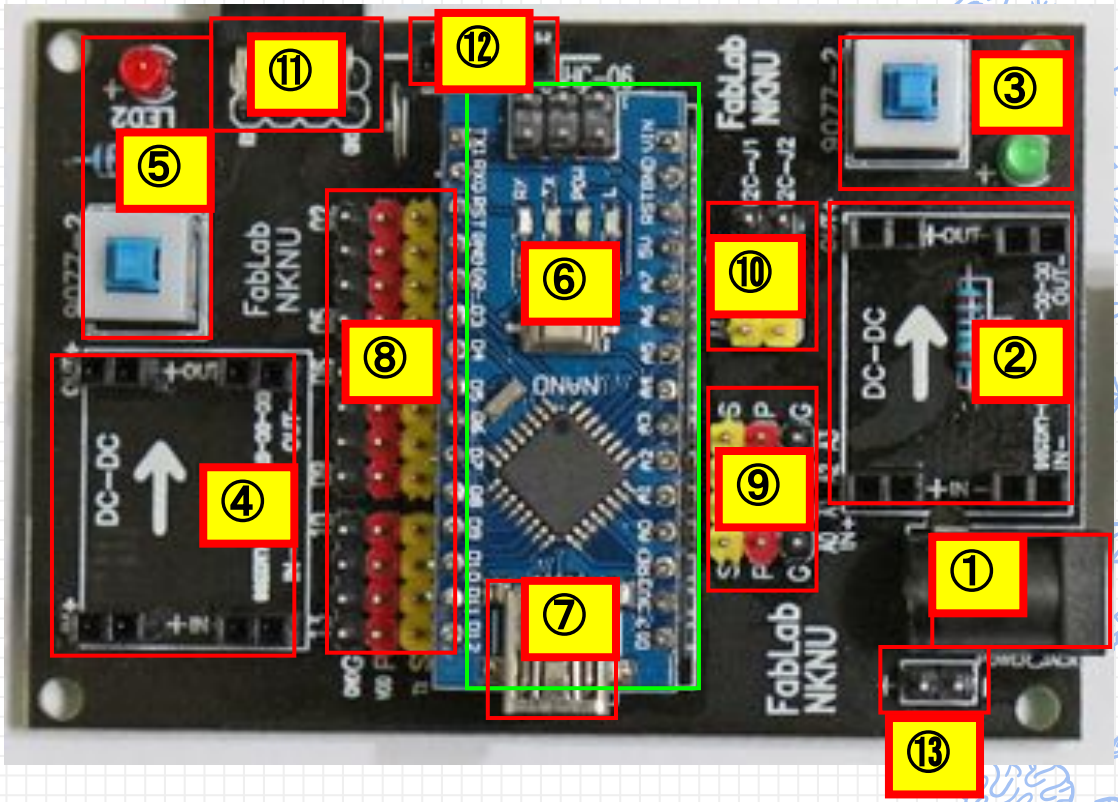
4060 WIFI 黑色電控板

- Nano I/O 擴充 + 5V 電源擴充+WIFI 擴充 + 藍芽擴充 + I2C 擴充
- WIFI 模組電源



4060 WIFI 黑色電控板

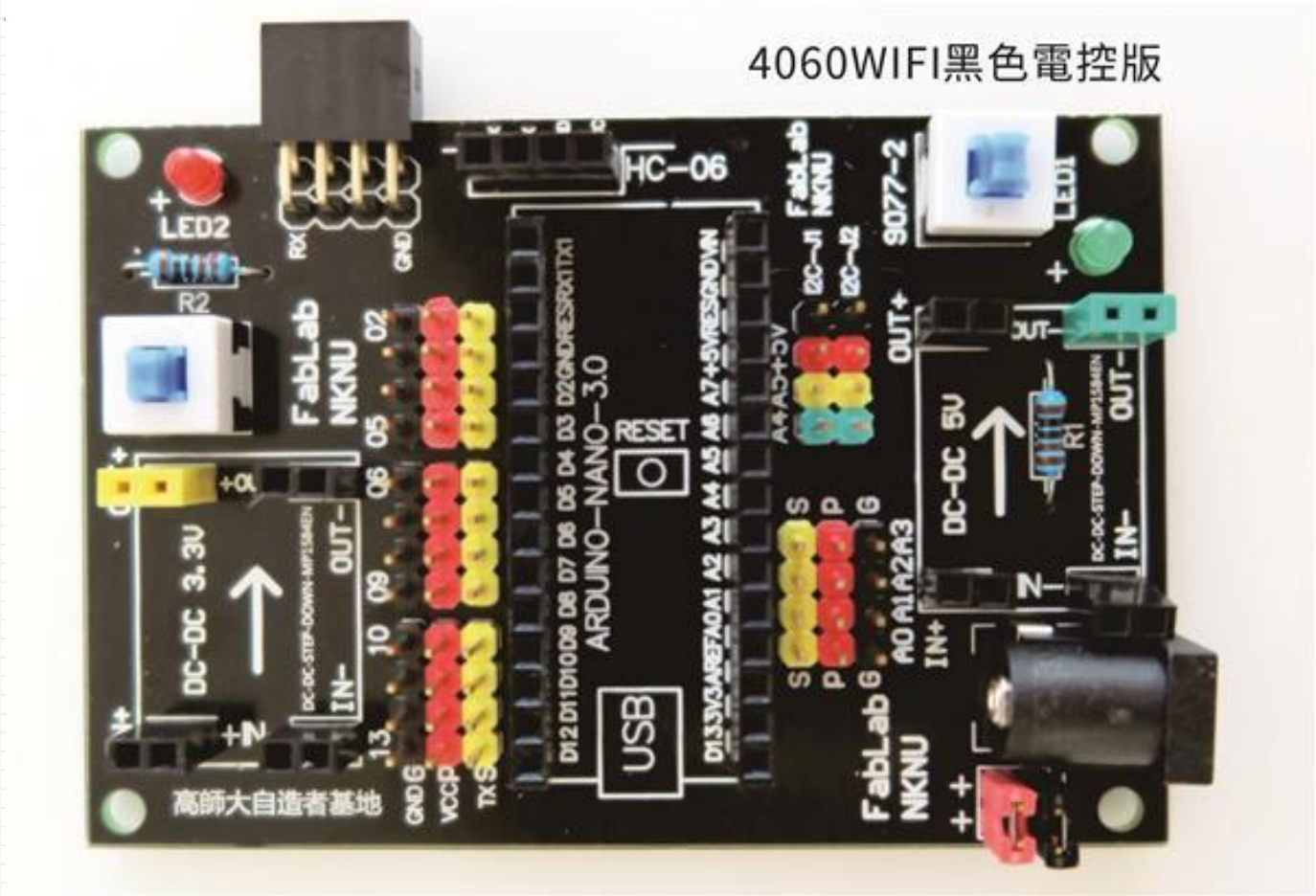
1). 2.1mm DC插座(7~12V)
2). 5V 降壓板
3). 5V 開關
4). 3.3V 降壓板
5). 3.3V(WiFi電源)開關
6). Nano
7). mini USB 接頭
8). 數位I/O&5V供電擴充接口
9). 類比I/O&5V供電擴充接口
10). I2C擴充接口
11). WiFi接口
12). 藍芽接口
13). Vin接口

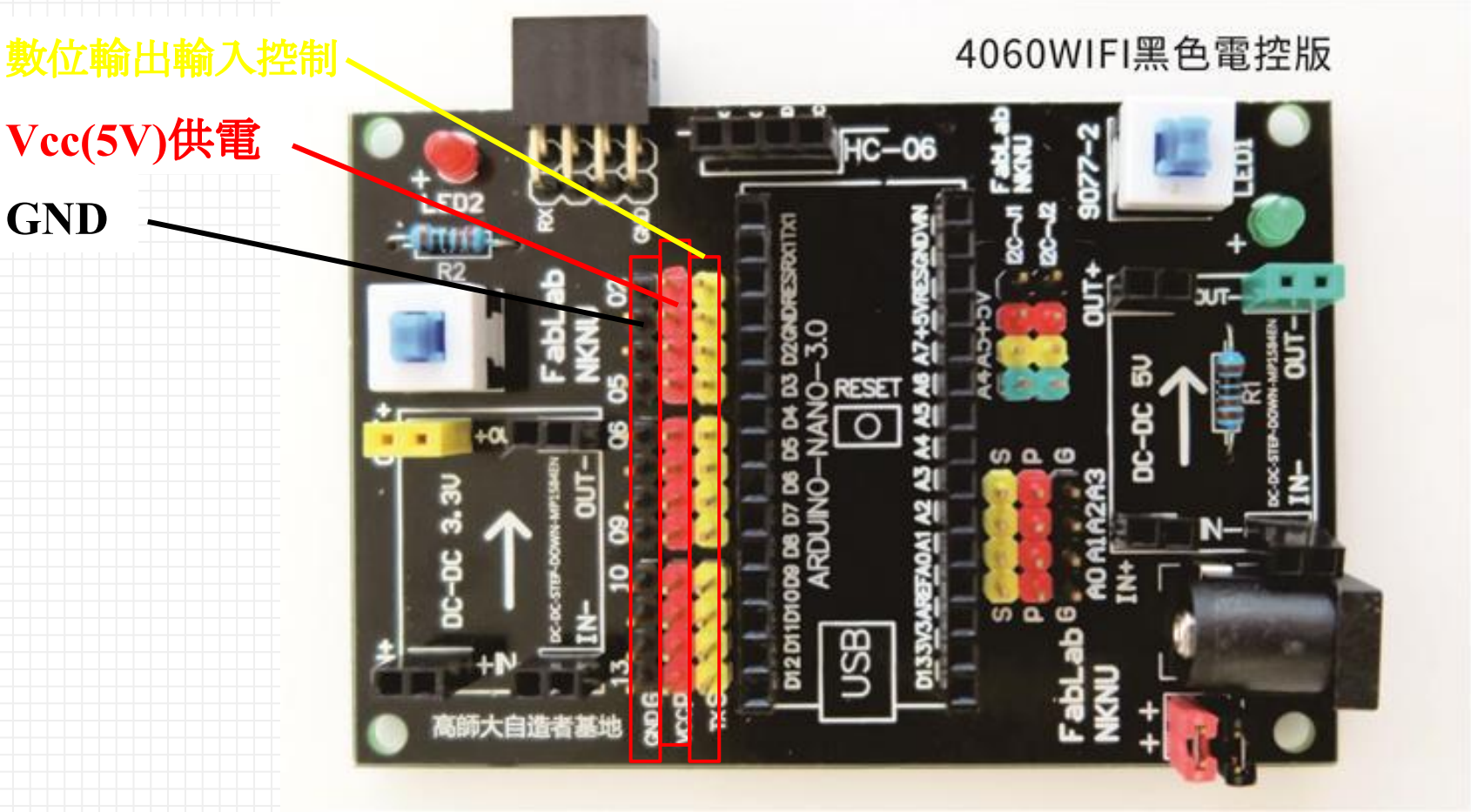


紅:供電 Vcc(5V)

黑:供電 GND

黃:訊號/控制

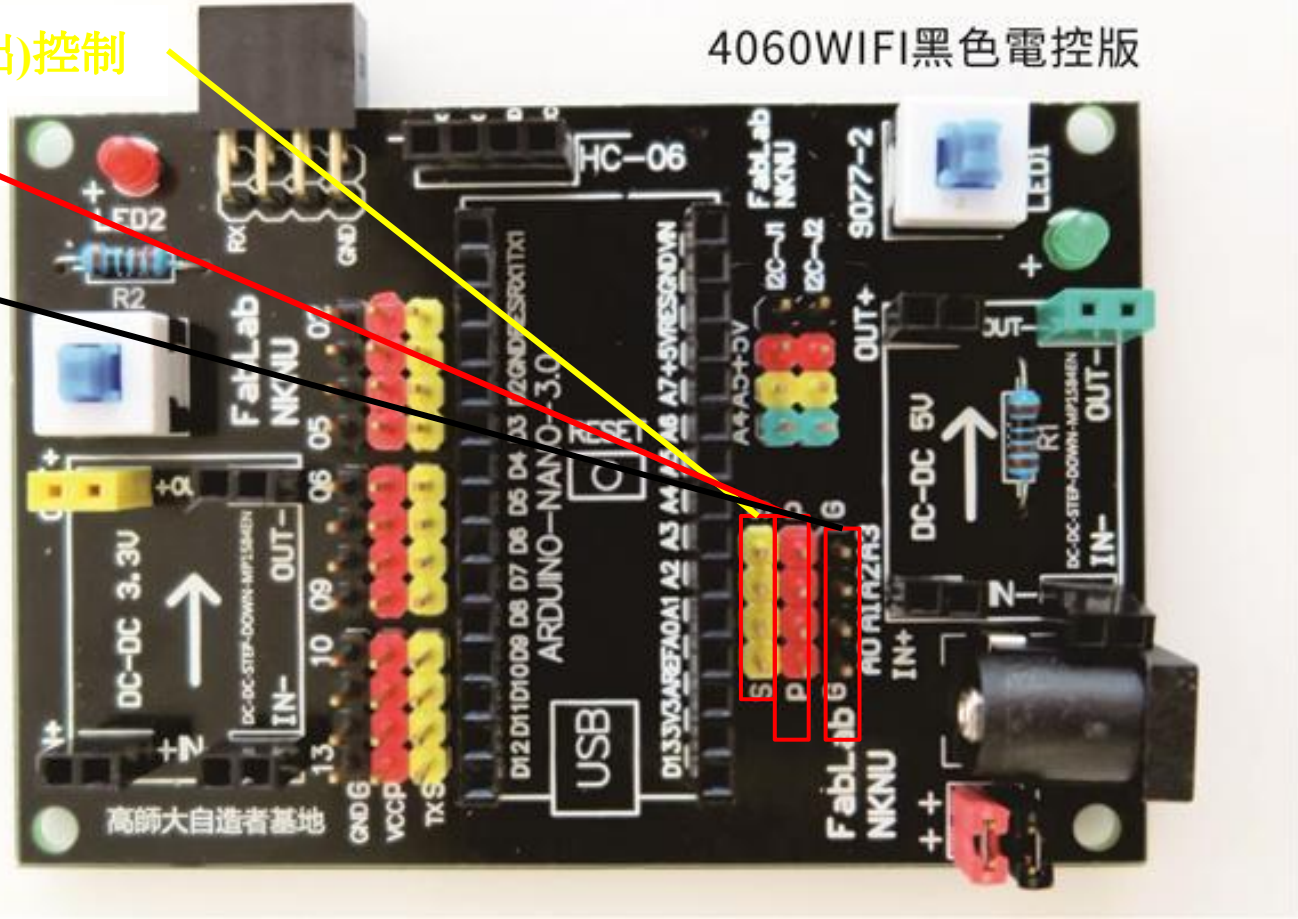



數位輸出輸入控制
Vcc(5V)供電
GND
4060WIFI黑色電控版
HC-06
ARDUINO-NANO-3.0
RESET
USB
DC-DC 3.3V
DC-DC 5V
FabLab NKNU
高師大自造者基地

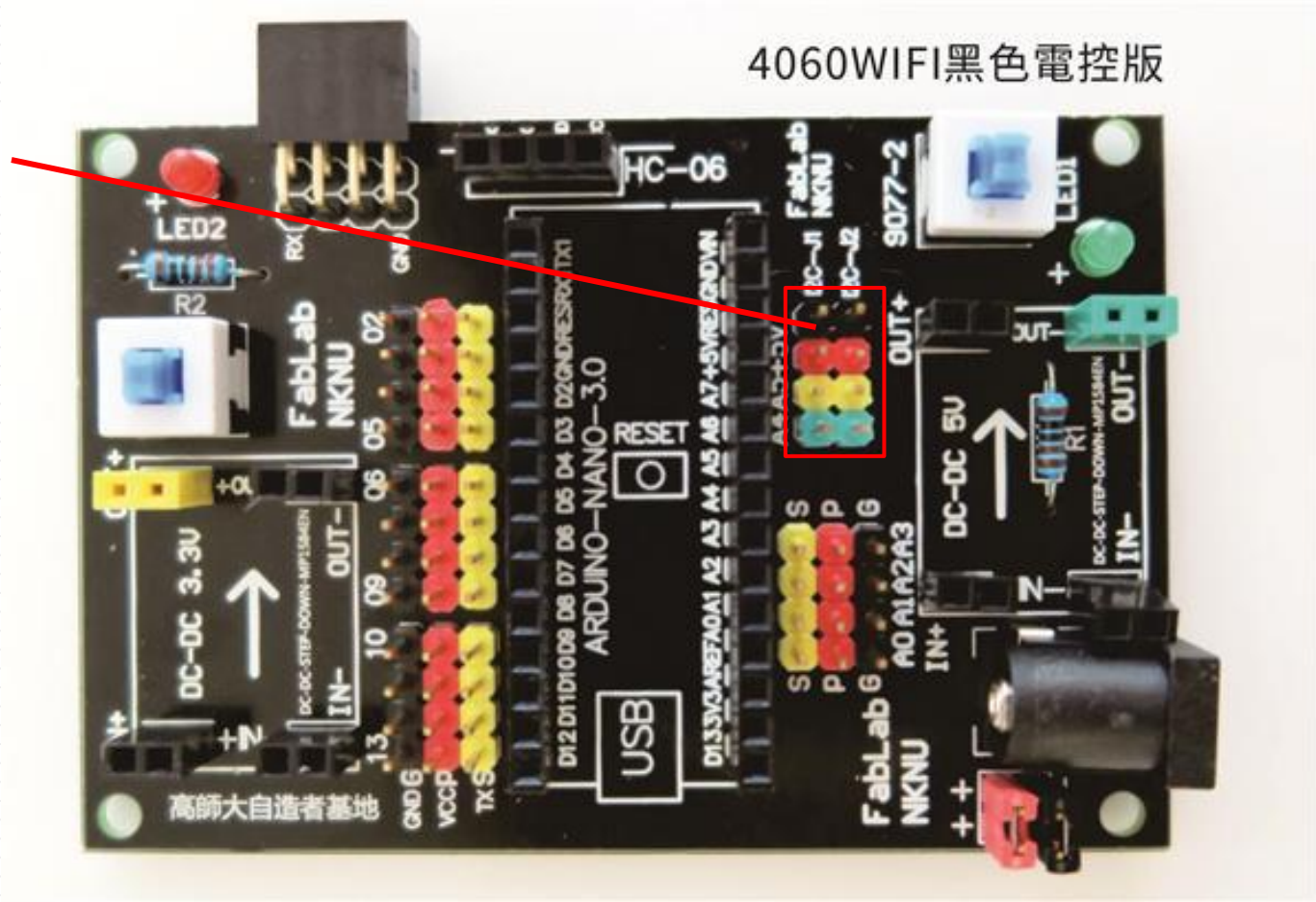
類比輸入(數位輸出)控制

Vcc(5V)供電

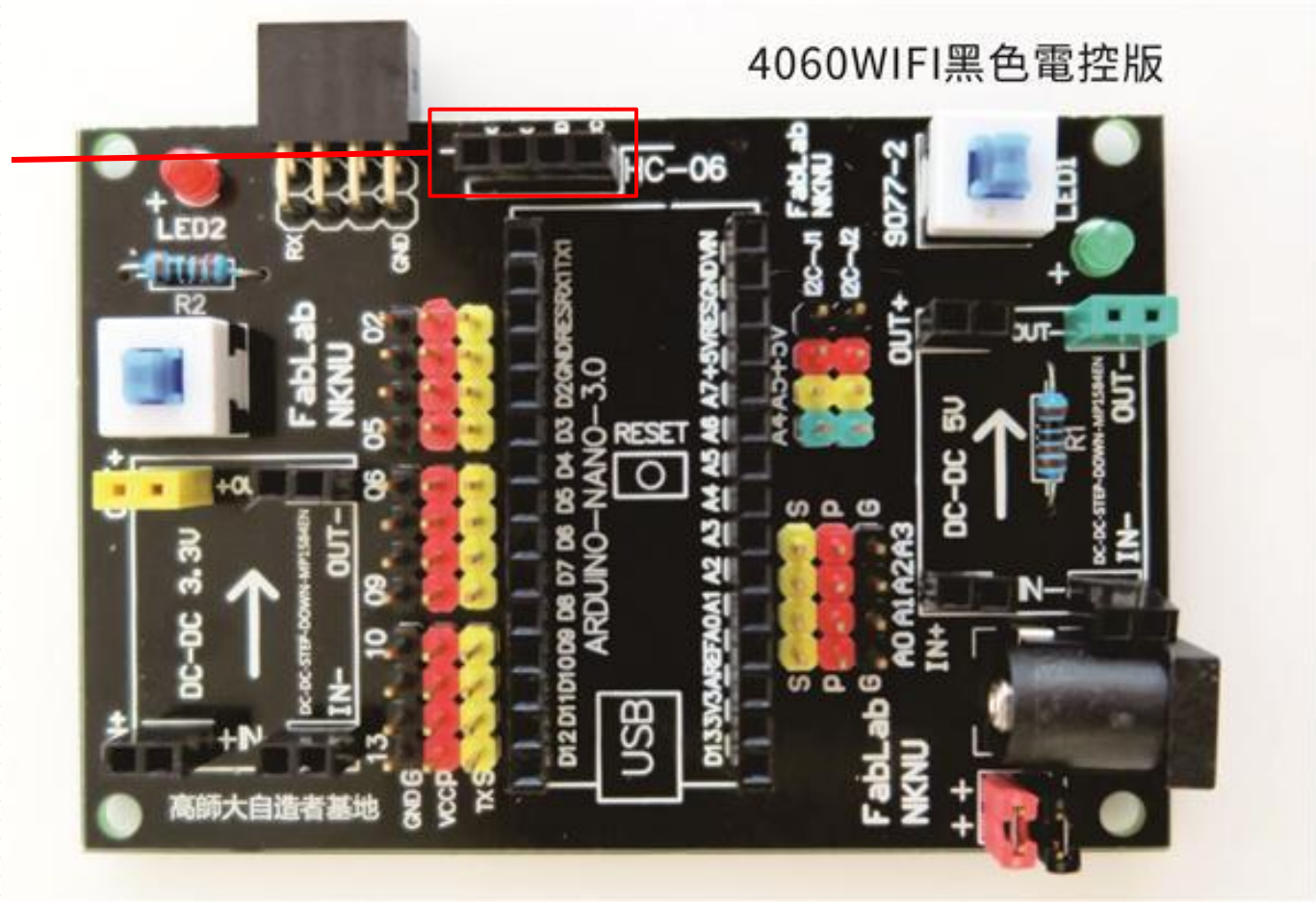
GND



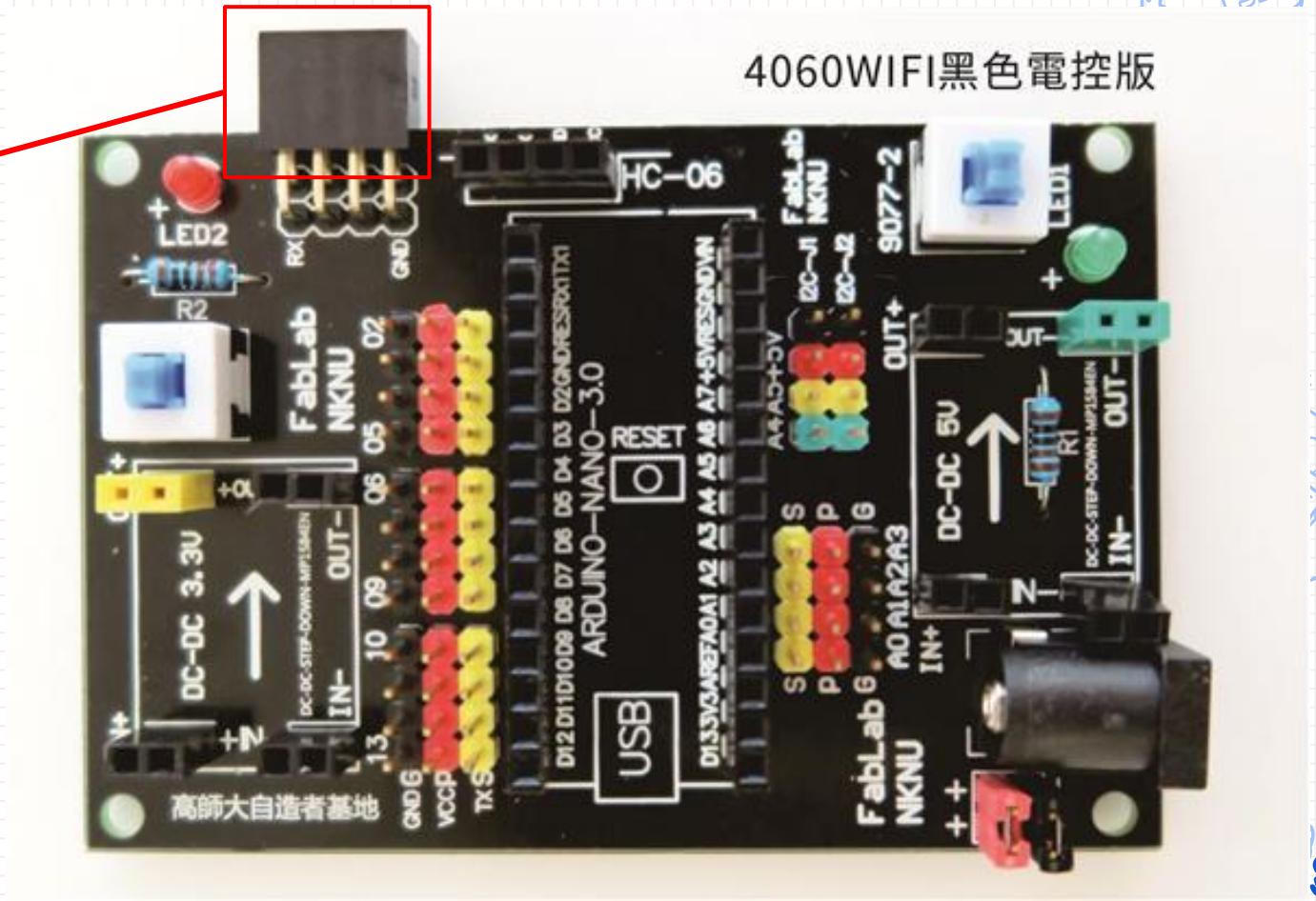
2 組 I2C


4060WIFI黑色電控版
HC-06
LED2
R2
LED1
9077-2
FabLab
NKNU
RESET
ARDUINO-NANO-3.0
USB
DC-DC 3.3V
DC-DC 5V
IN+
IN-
OUT+
OUT-
高師大自造者基地

藍牙


4060WIFI黑色電控版
HC-06
LED2
R2
LED1
9077-2
FabLab
NKNU
ARDUINO-NANO-3.0
RESET
USB
DC-DC 3.3V
DC-DC 5V
IN+
IN-
OUT+
OUT-
高師大自造者基地

WiFi
(ESP-01)


4060WIFI黑色電控版
HC-06
LED2
R2
RX
GND
FabLab
NKNU
RESET
ARDUINO-NANO-3.0
USB
LED1
9077-2
DC-DC 3.3V
DC-DC 5V
IN+
IN-
OUT+
OUT-
高師大自造者基地

供電系統

- 18650 鋰電池(平頭, 2600mhA)
- 雙槽 18650鋰電池充電器

供電系統

- 18650 鋰電池(平頭, 2600mhA)
- 行動電源盒

18650 鋰電池

18650 鋰電池

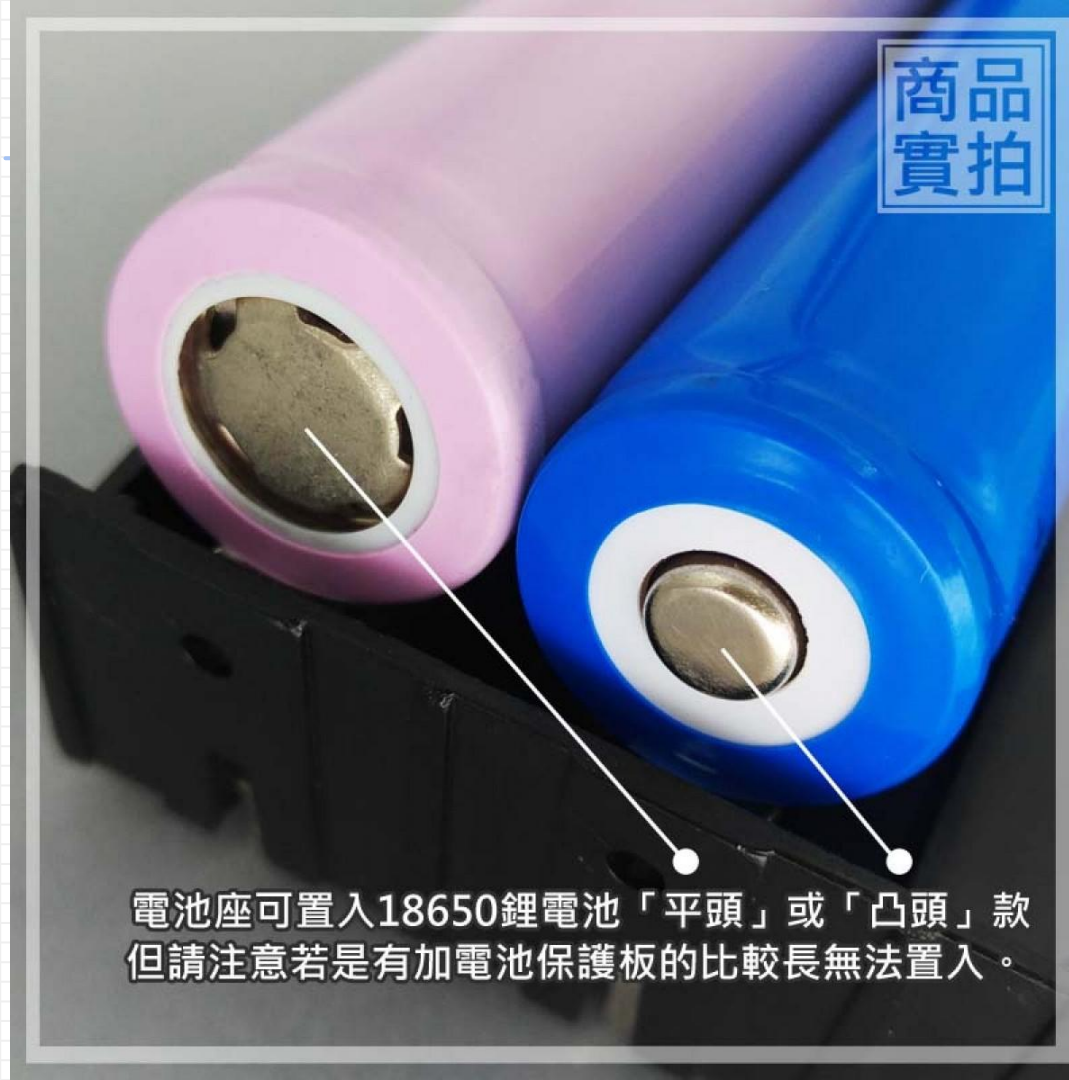
電壓:3.7V

容量:3250 mAh(電流325mA，可供電10小時)(電流32.5mA，可供電100小時)

)

凸頭(加工焊接)

平頭



平頭　　尖頭　　保護板

平頭電池

一般適用於有馬達運轉之小風扇，或是供電需求之18650電源儲存盒

尖頭電池

一般適用於LED、T6、L2、手電筒、探照燈等及其他照明設備產品

加保護板

此款電池增加晶片保護，控制電流穩定度，防止過充、過放、電流短路

Q 如何分辨？

加保護板　無保護板

加保護板

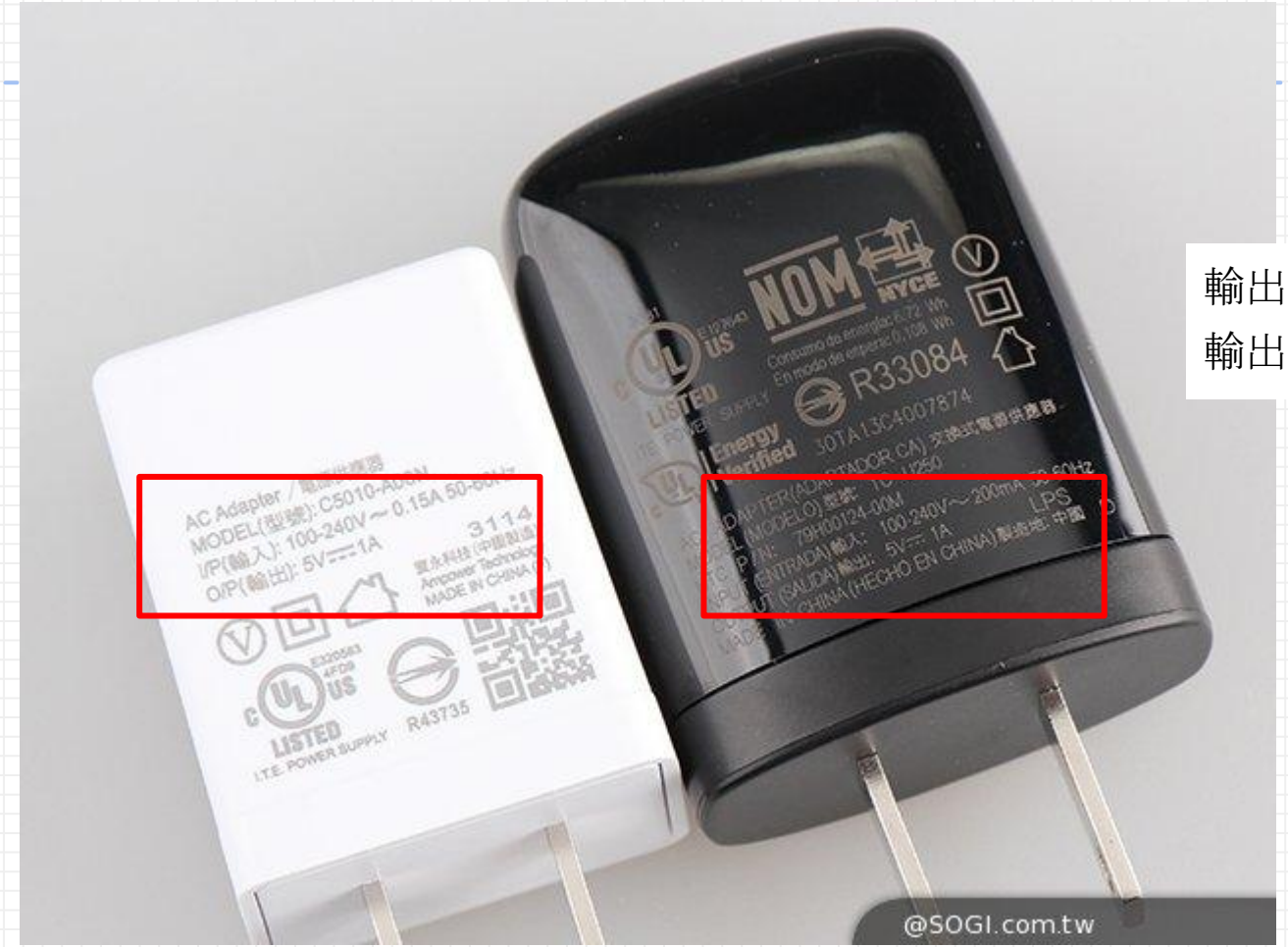
無保護板

A 加保護板的電池負極端
會有一圈黑色的墊片喔！

釹鐵硼強力磁鐵

直徑:4~6mm

厚度:1.5~2mm



輸出:5V，1A

輸出:5V，1000mA

輸出:5V，2A

輸出:5V, 1.67A

輸出:5V, 2A


XIAOMI 电源适配器
型号:MDY-08-EV
执行标准： GB17625.1-2012
GB4943.1-2011 GB9254-2008
输入:100-240VAC,50/60Hz,0.35A
输出:5V⎓2.0A 2A4171258089798
惠州比亚迪电子有限公司 制造

輸出:5V, 2A

弹片是正极 +，弹簧为负极 -

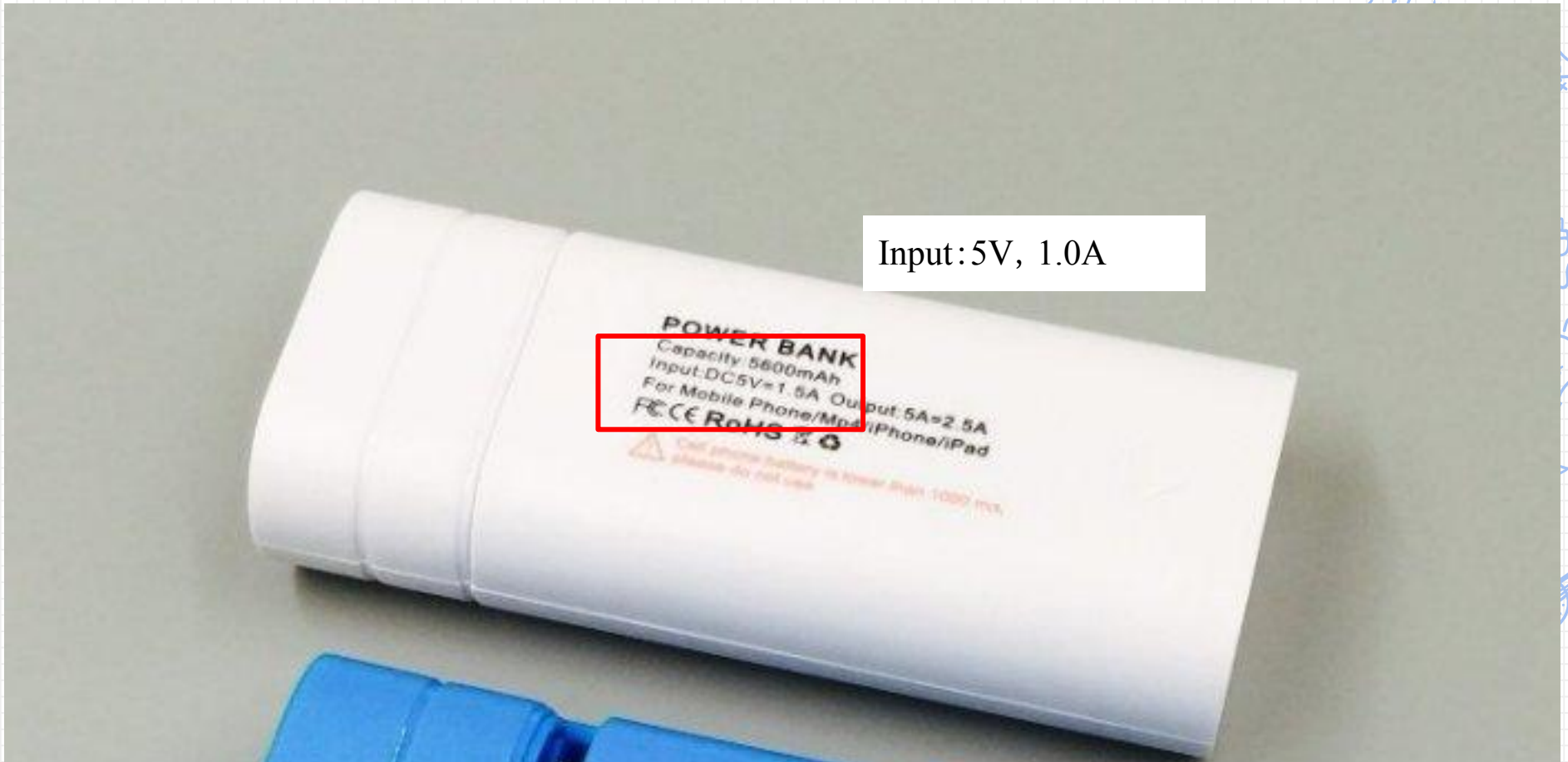

实物拍摄 盗图必究

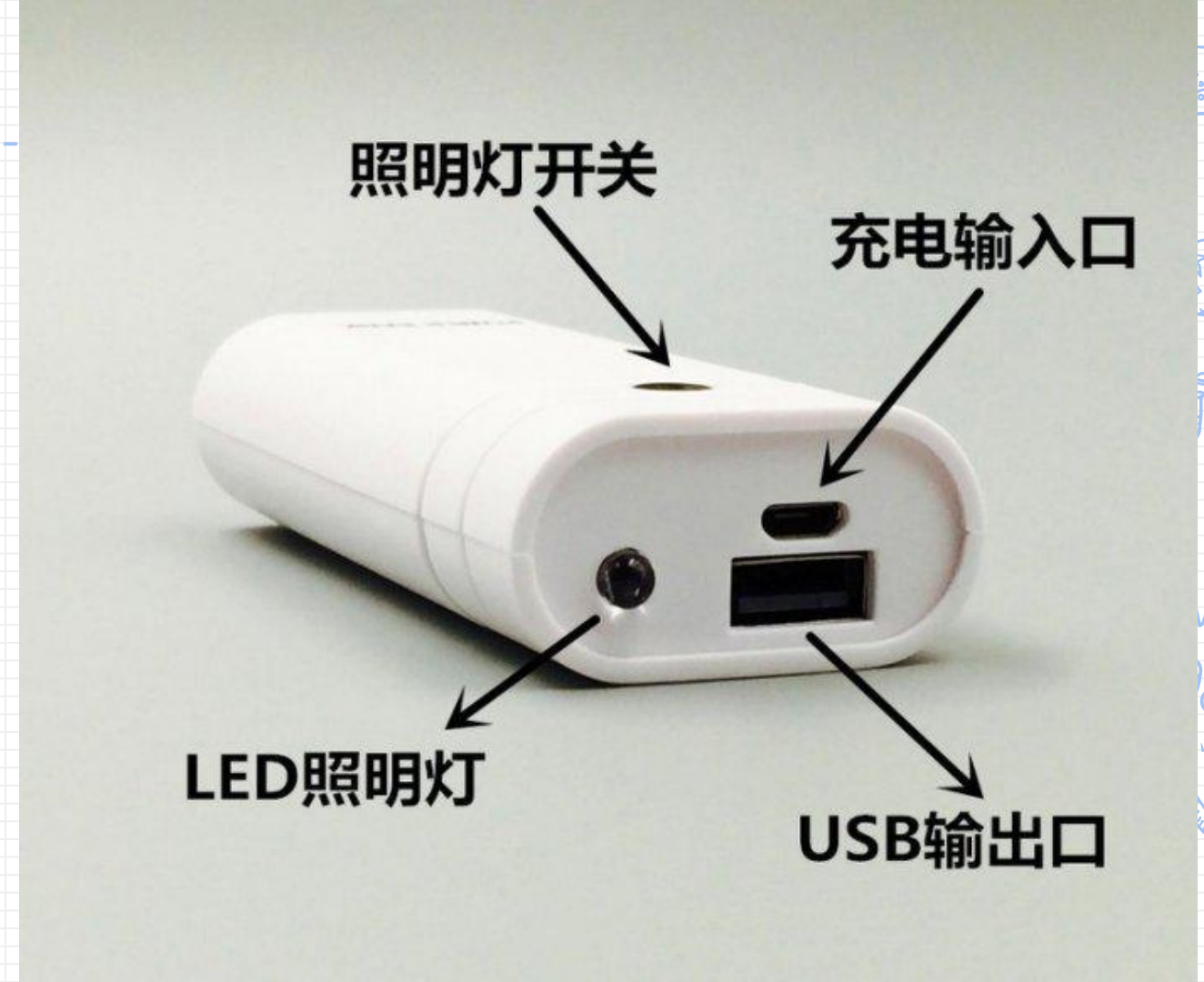
產品規格

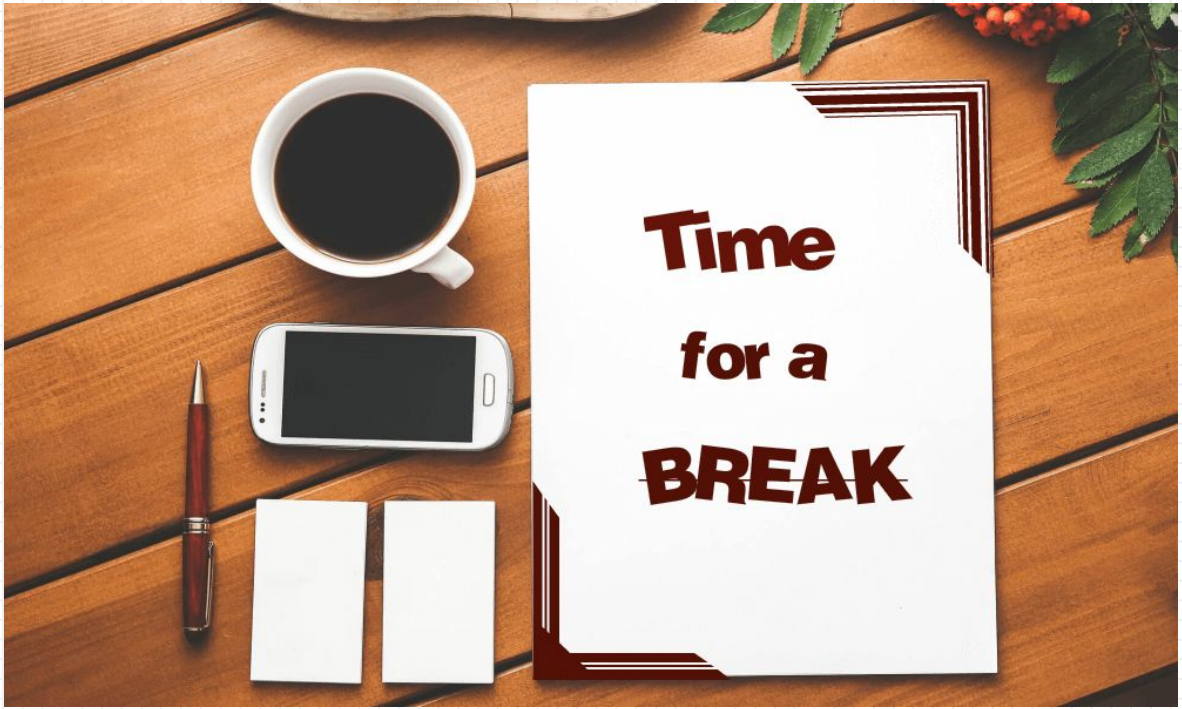
- 輸入電壓:MicroUSB 5.0V±0.5V
- 輸入電流:850mA
- 輸出電壓:USB 5.0V±0.5V
- 輸出電流:1200mA 實測=1.25A(MAX)
- 尺寸:100 * 43 * 23 mm
- 保護功能: 短路、過流、過充、過放、欠壓保護

操作說明(切記一定要先裝好正常的18650電池再充電)

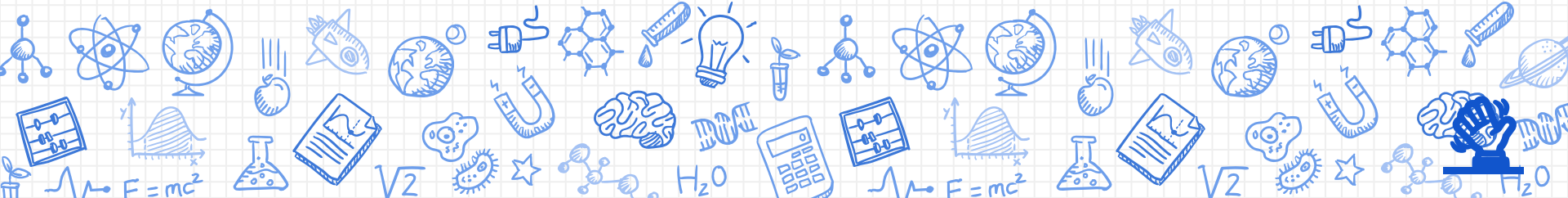
- 充電說明:1~4顆藍燈會依當時充電時的電量顯示, 分別為25% - 50% - 75% - 100%, 被充飽後4顆藍燈恆亮
- 放電過程:亮藍燈, 當下顯示幾顆燈電量分別為25% - 50% - 75% - 100%
- 帶手電筒功能(按1下開關亮燈-高亮)
- 可裝兩節平頭18650電池(注意是平頭與凸頭電池都能裝)


Input：5V，1.0A
POWER BANK
Capacity:5600mAh
Input:DC5V=1.5A Output:5A=2.5A
For Mobile Phone/Mp4/iPhone/iPad
RoHS


照明灯开关
充电输入口
LED照明灯
USB输出口


Time
for a
BREAK

4060小車組裝



馬達轉向測試

- N20馬達輪軸朝外, 凹洞朝上, 訂出左右輪(圖為右輪)

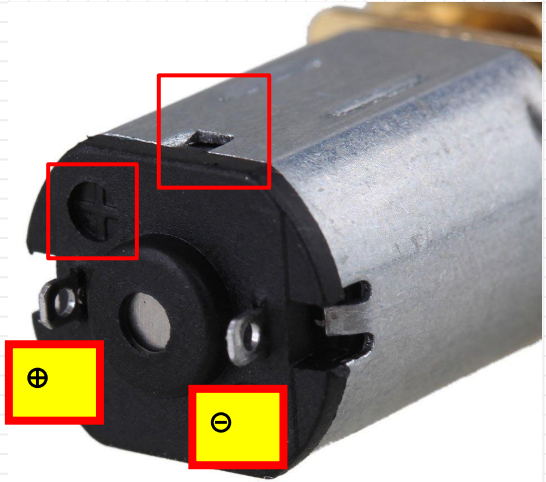
凹洞朝上　　輪軸朝外

左輪

右輪

馬達轉向

- 訂出馬達正負端



馬達轉向

- 以自焊 USB 電源接頭測試馬達轉向

USB(-)　USB(+)

USB(+)　USB(-)

後退　前進

直流減速馬達N20

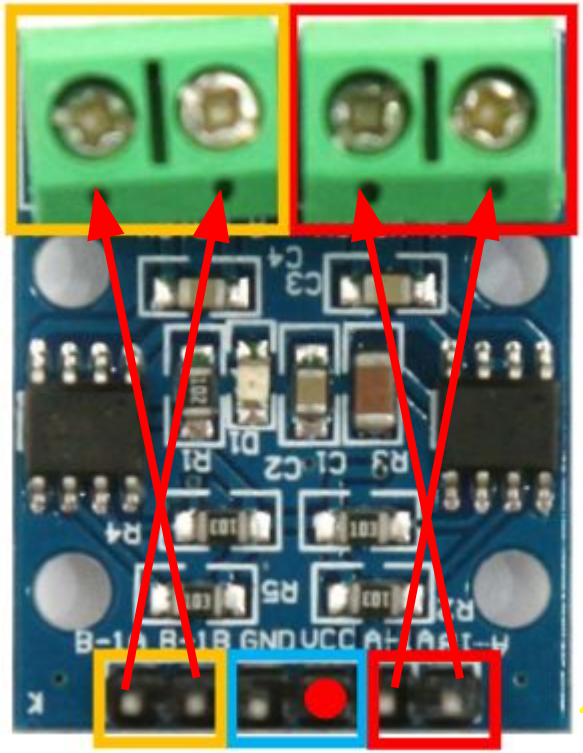
- 轉向測試

	左輪(+)	左輪(-)	右輪(+)	右輪(-)	運動
電源	-	+	+	-	前進
電源	+	-	-	+	後退
電源	+	+	+	+	停止
電源	-	-	-	-	停止

馬達接線

右輪(+) 右輪(-) 左輪(-) 左輪(+)

- 左輪
  - AIB↔D2↔左輪(-)
  - AIA↔D3~↔左輪(+)
- 右輪
  - BIB↔D4↔右輪(+)
  - BIA↔D5~↔右輪(-)



輪胎

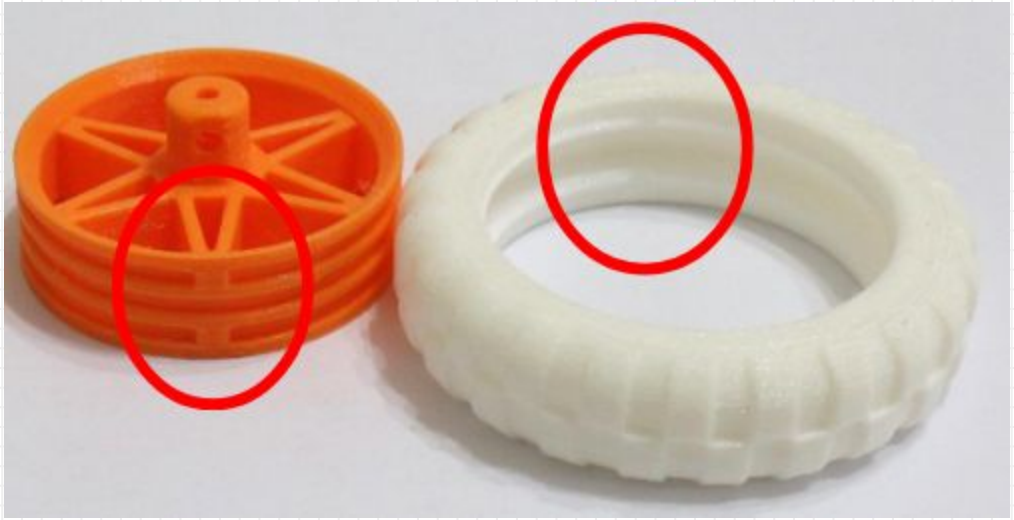
左輪　右輪

內側

外側

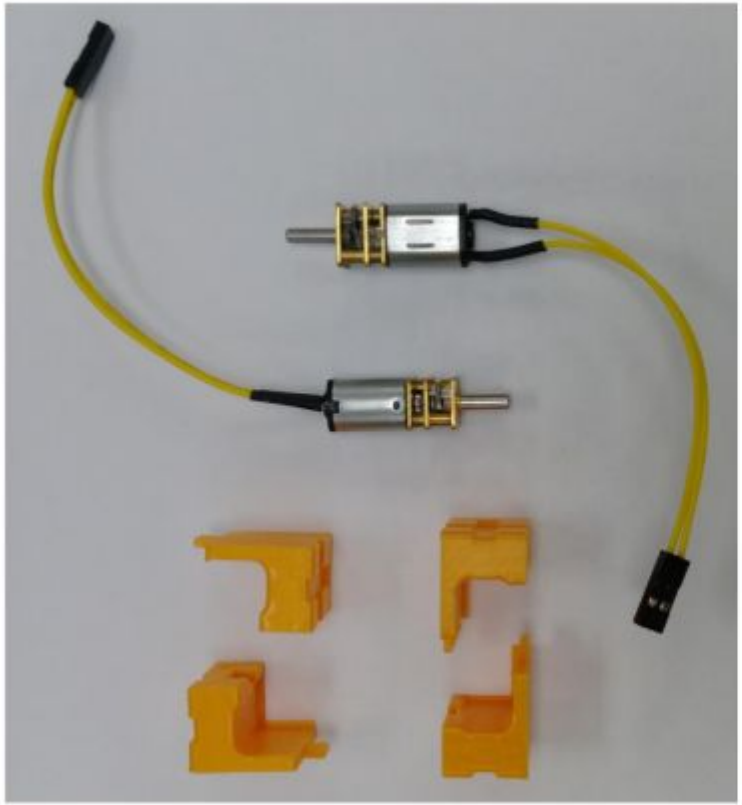
注意胎紋方向不同

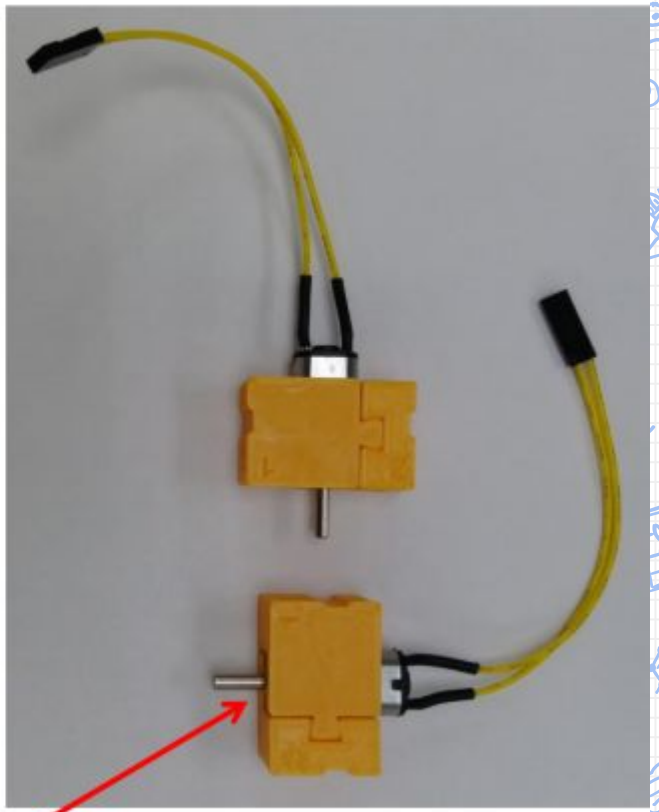
輪胎



輪框與輪胎的溝槽要配合

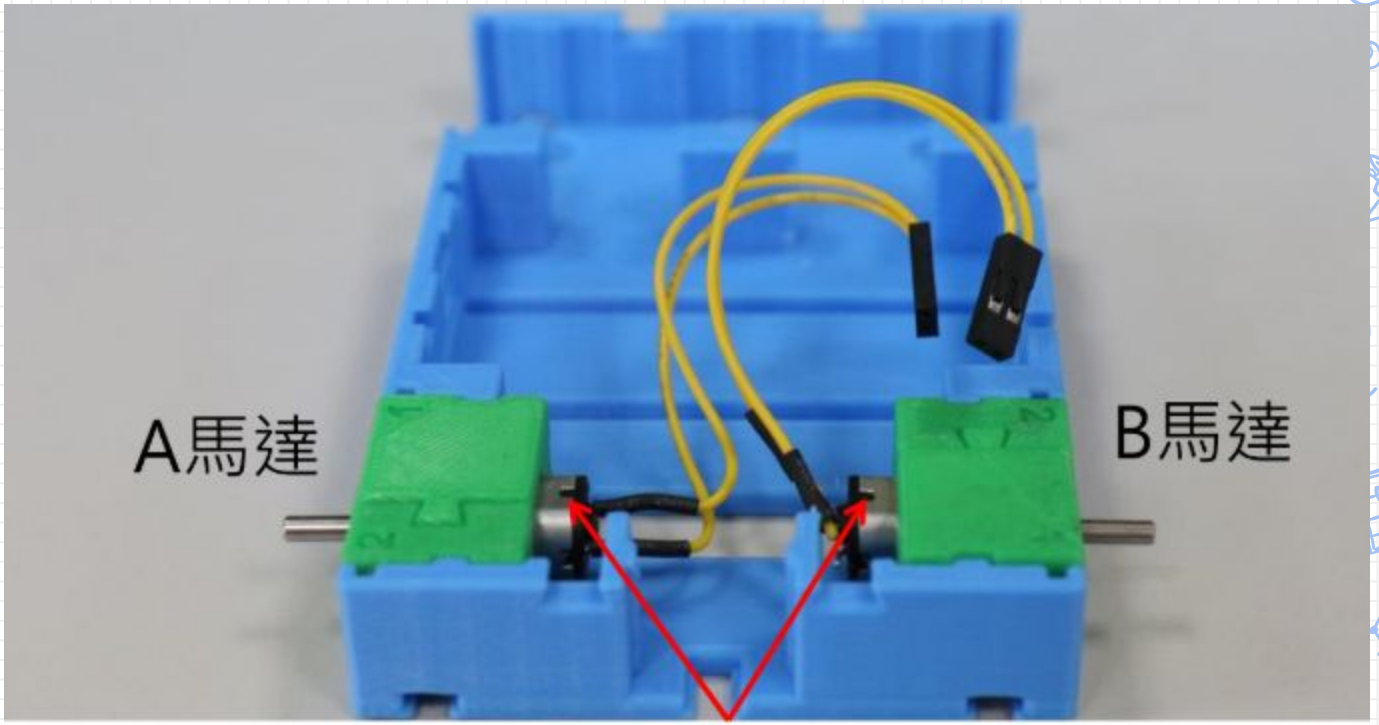
N20馬達





注意銅片不外漏

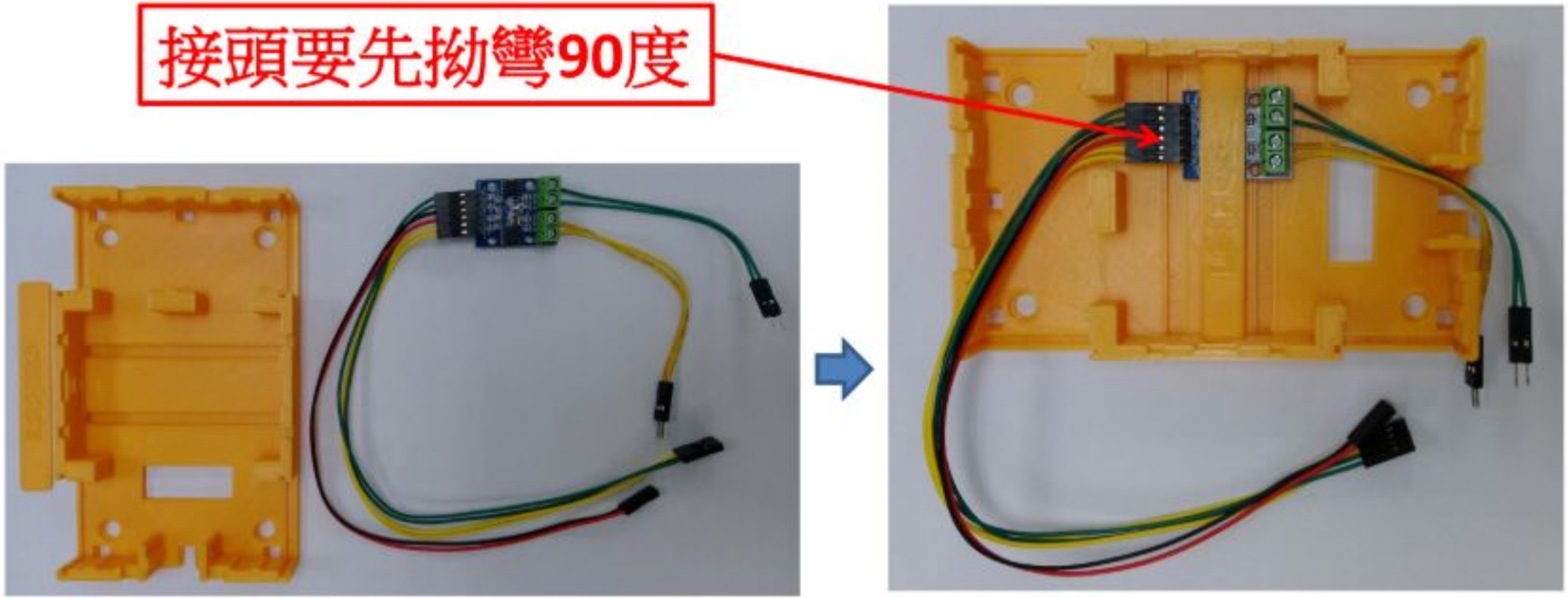
N20馬達



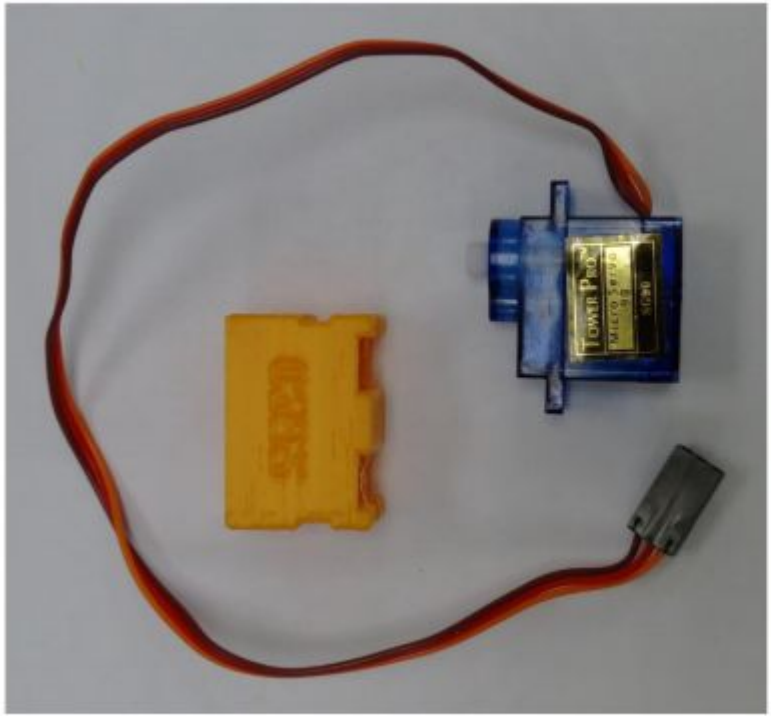
黑色凹槽朝上

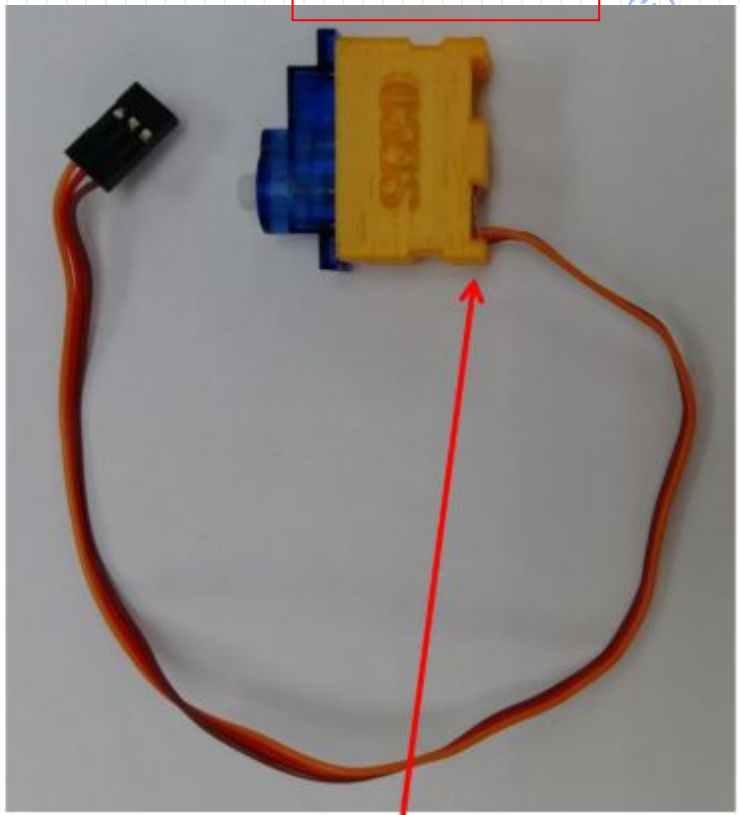
L9110S 馬達驅動板

❖

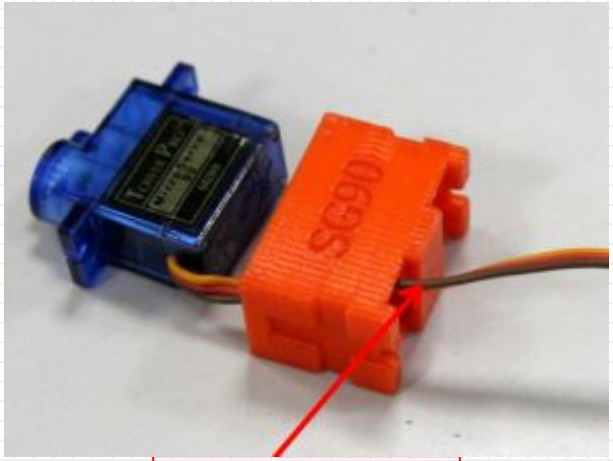


SG90 伺服馬達





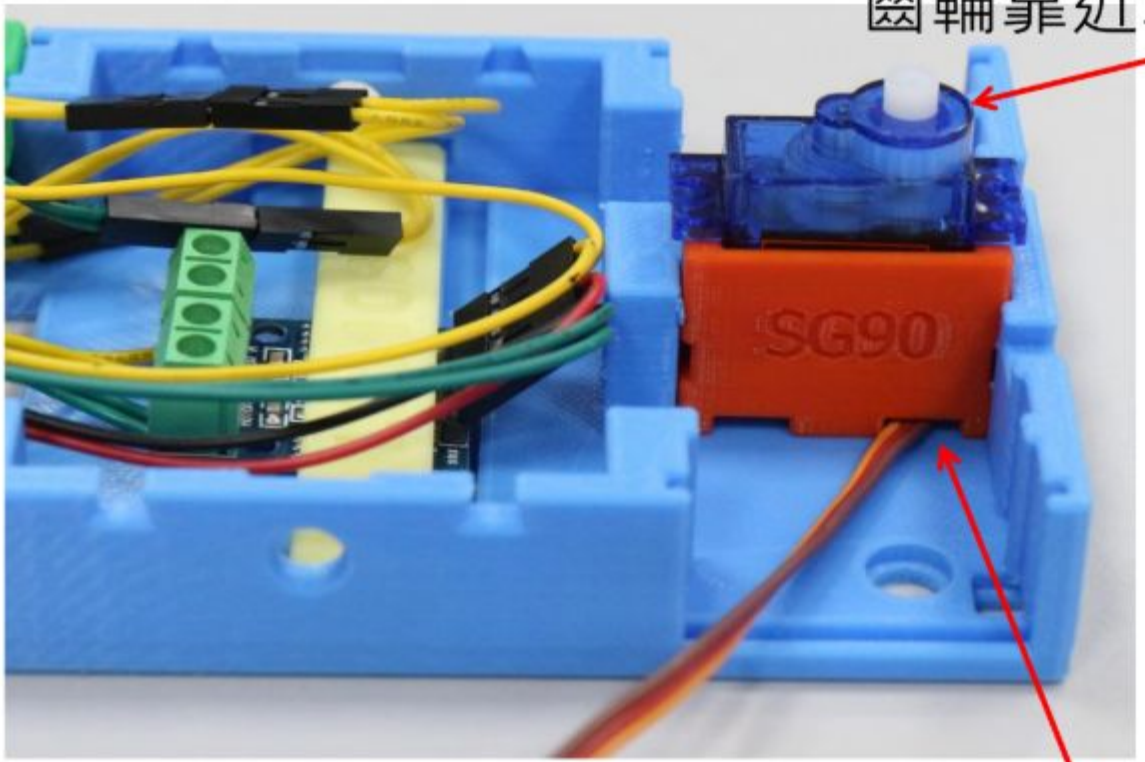
SG90 伺服馬達



線先穿過外殼

注意線從底部穿出

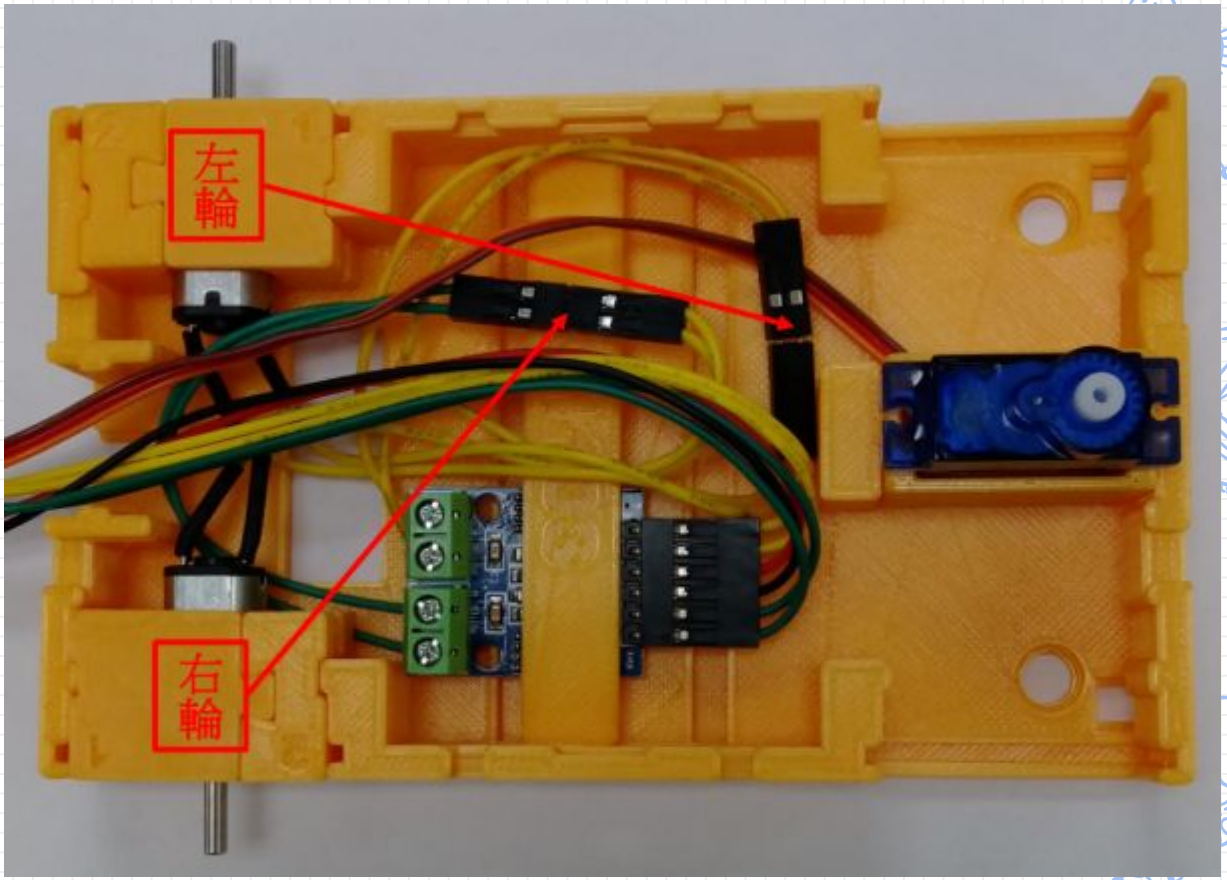
SG90 伺服馬達



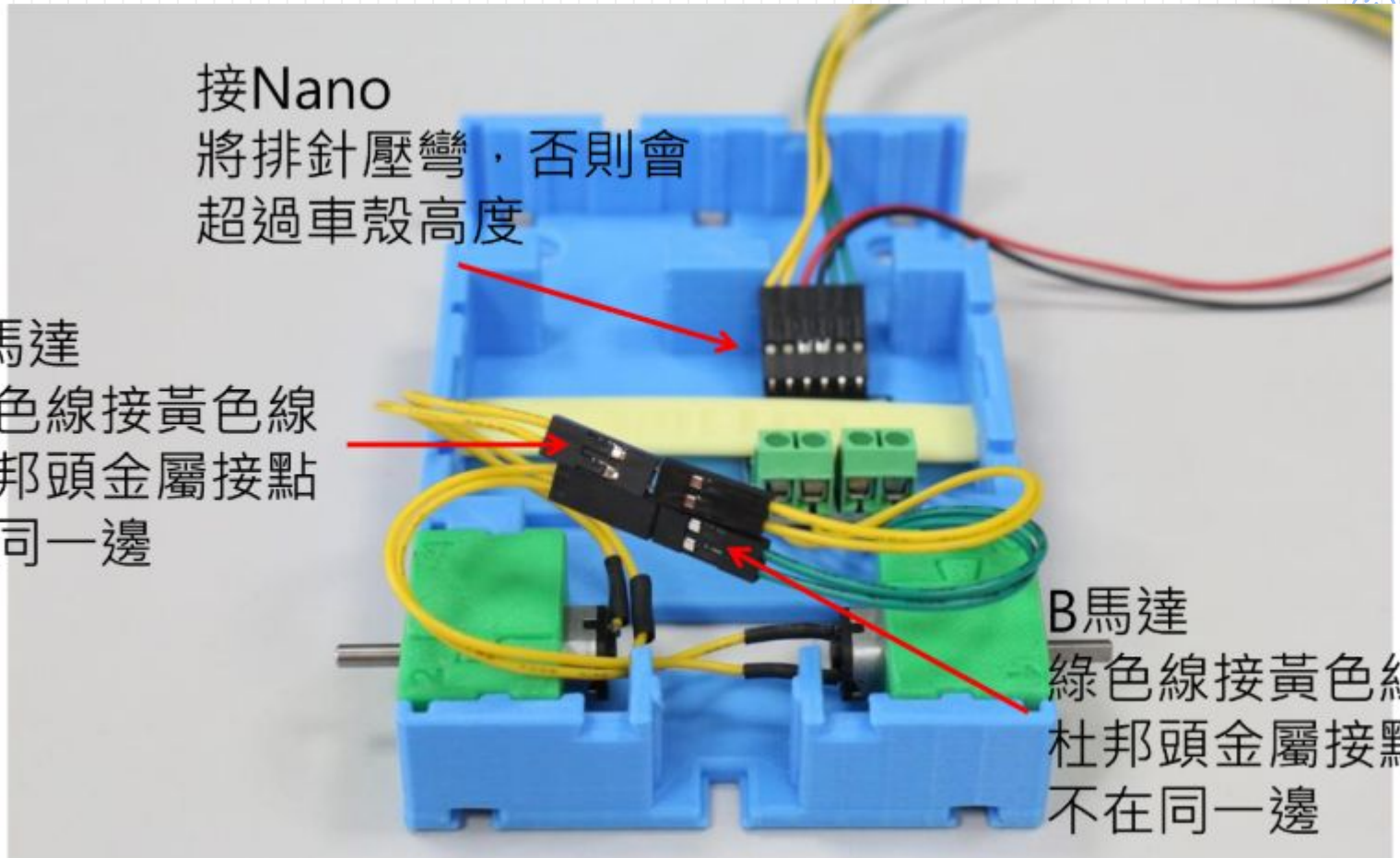
馬達接線

❖

L9110S
黃色接左輪
綠色接右輪



馬達接線


接Nano
將排針壓彎，否則會
超過車殼高度
A馬達
黃色線接黃色線
杜邦頭金屬接點
在同一邊
B馬達
綠色線接黃色線
杜邦頭金屬接點
不在同一邊

馬達接線

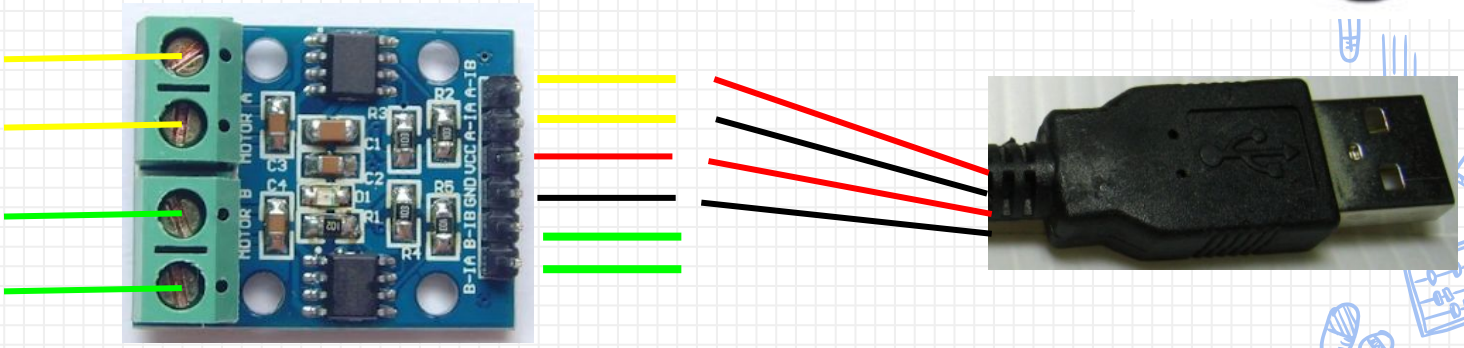
- 將左右輪N-20馬達與L9110S馬達驅動板接線後，與這裡需先停下來進行左右輪馬達轉向測試。

左馬達轉向測試

將L9110S與自製USB接頭接線如下, 觀察左輪馬達應前進。
若未前進, 調整(交換)N-20馬達與L9110S之接線接腳。

左輪馬達

右輪馬達

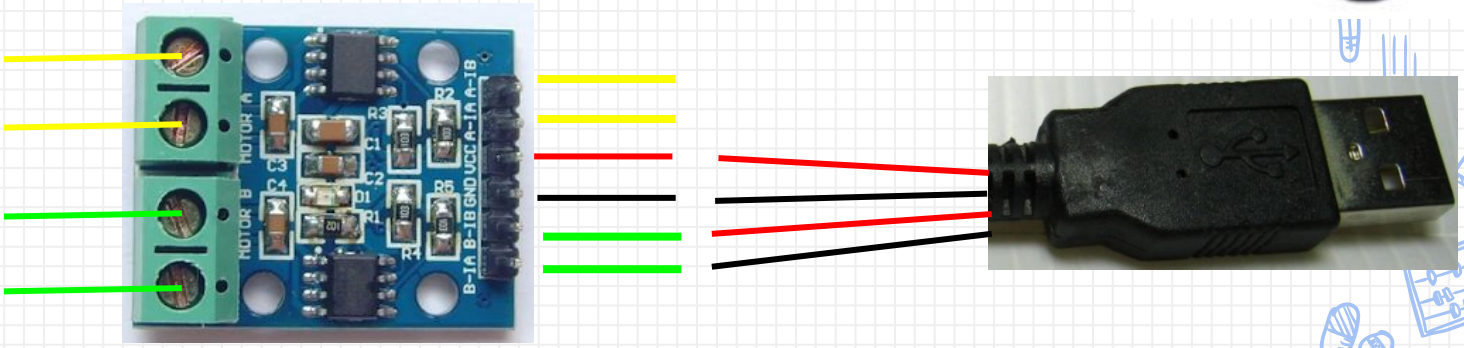


右馬達轉向測試

將L9110S與自製USB接頭接線如下, 觀察右輪馬達應前進。
若未前進, 調整(交換)N-20馬達與L9110S之接線接腳。

左輪馬達

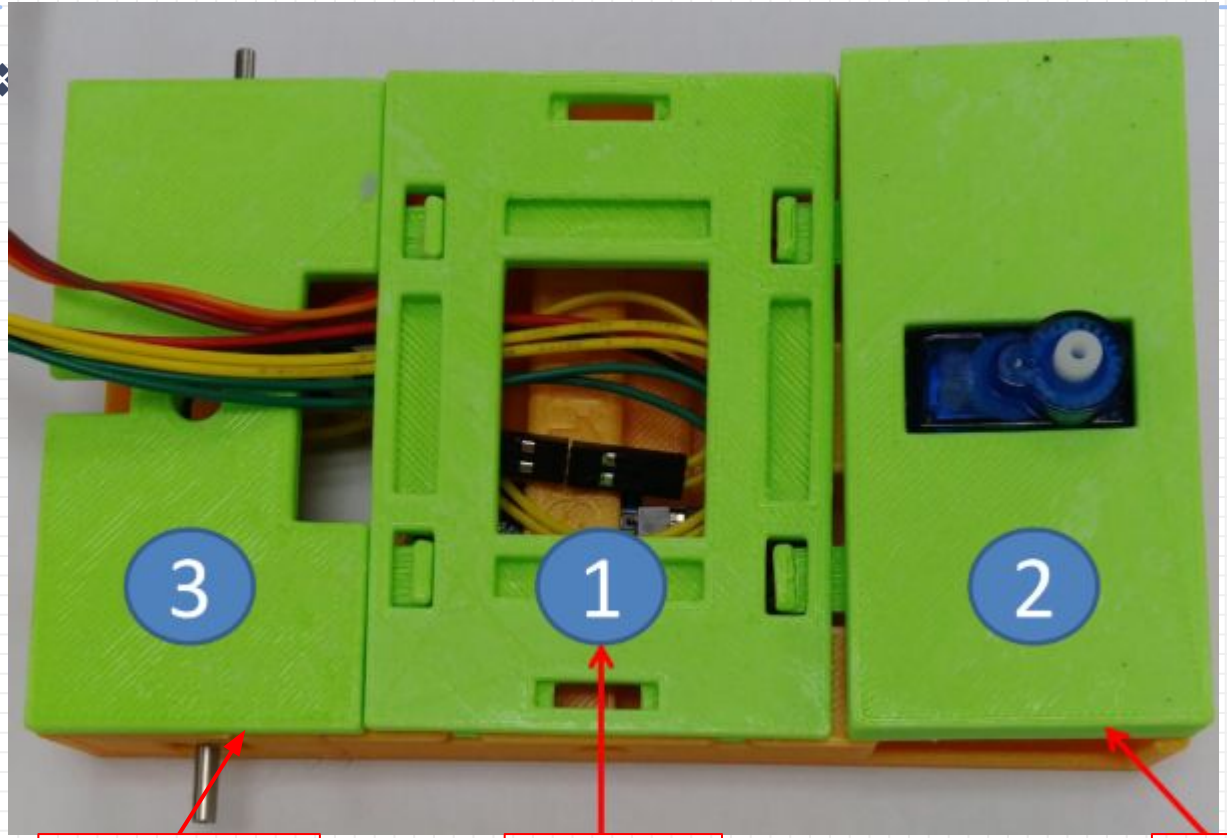
右輪馬達



馬達接線

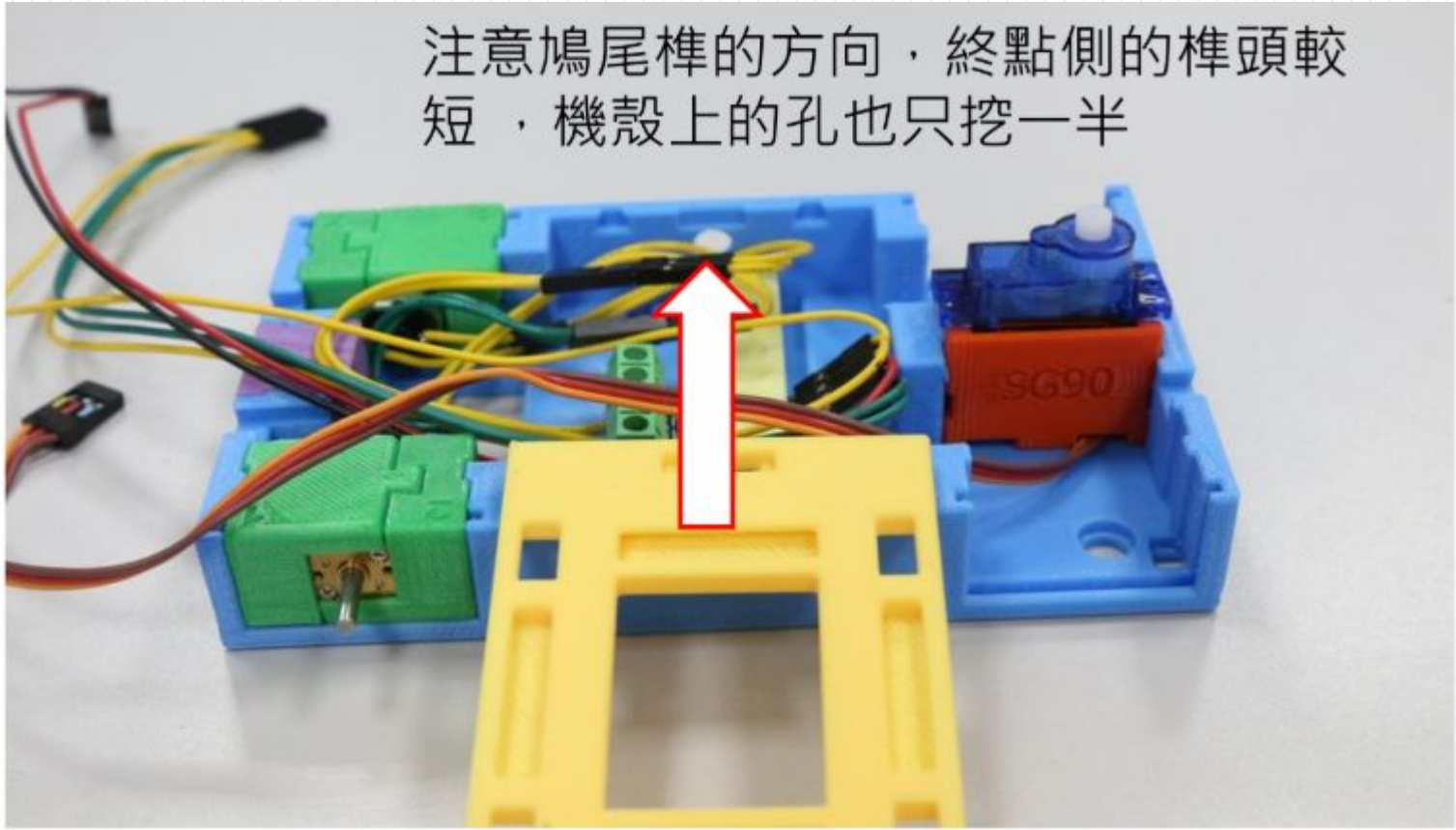
- 請確定左右輪馬達轉向測試與接線正確再往下。

蓋板

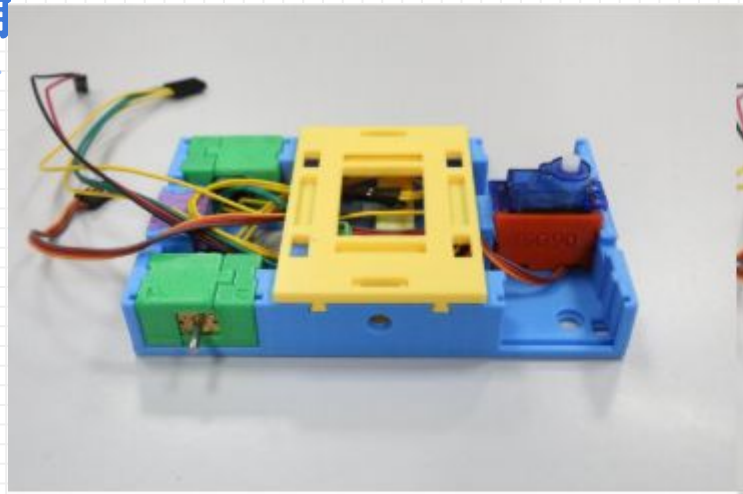


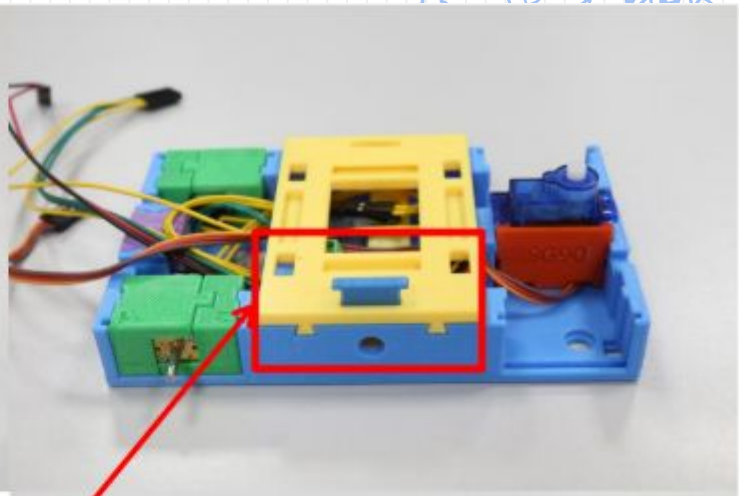
傾斜再装入　組装順序　傾斜再装入

中蓋板

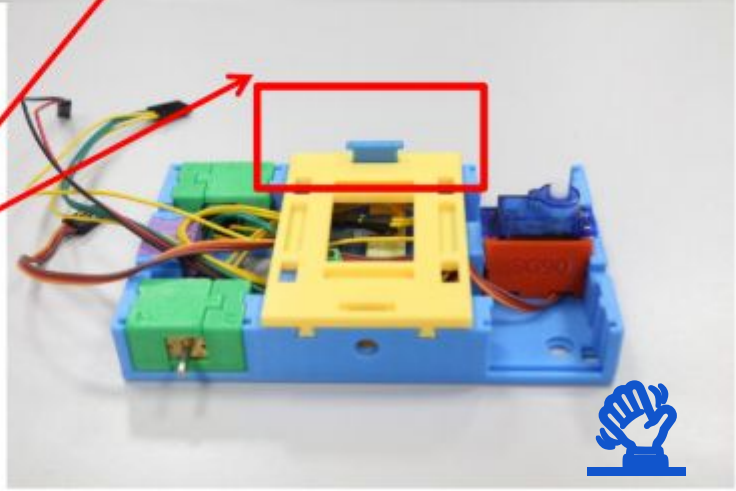


中蓋板插銷



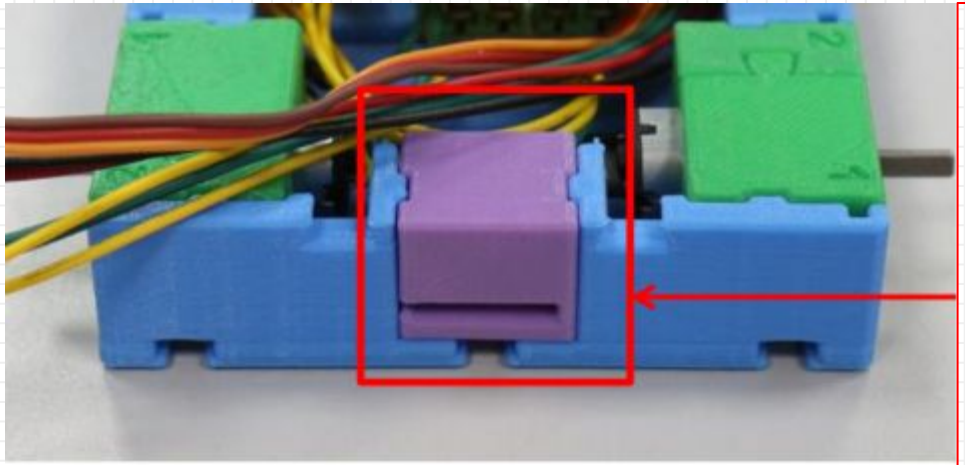


用途：
1.擋住電池
2.裝在右上圖的位置
可以固定中蓋板
3.也可以不裝



中央循線感應器外殼

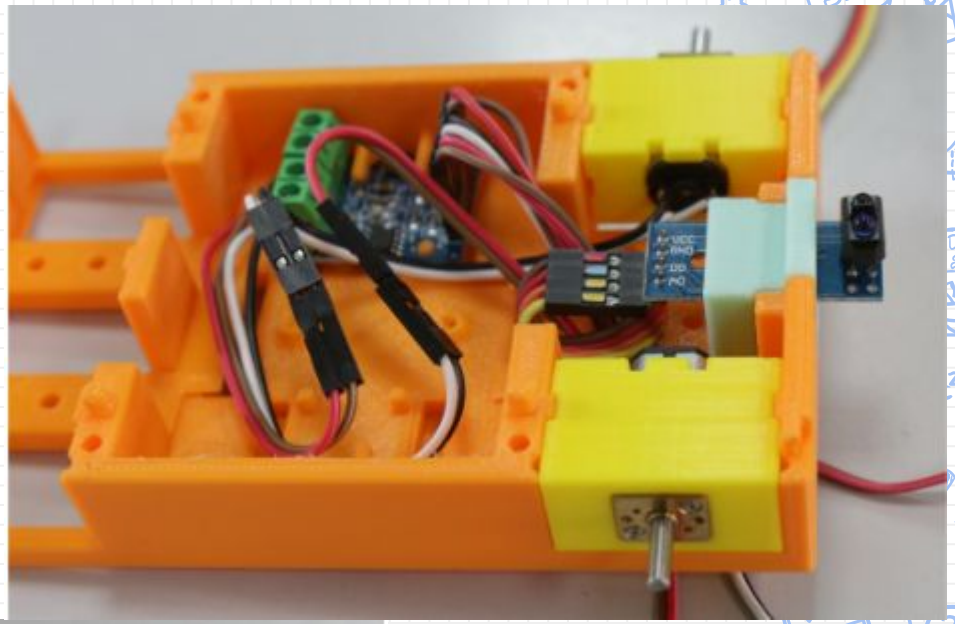
❖

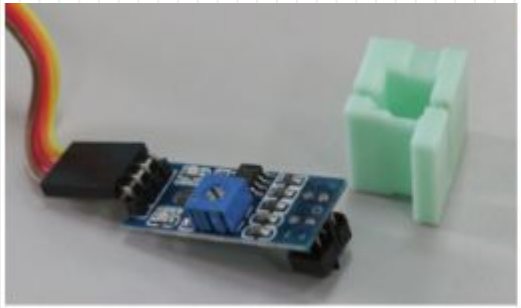


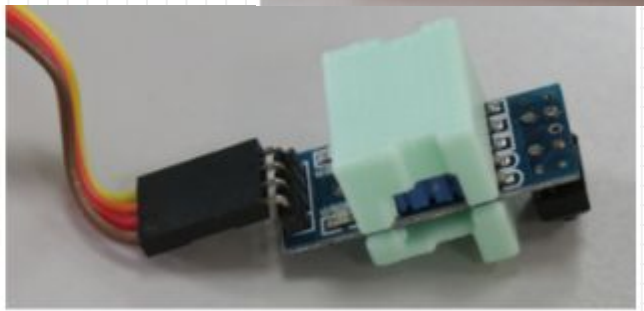
- 外緣切齊，開口在下方
- 不一定要裝，裝上可以讓上方蓋板牢固一些
- 不裝也可以，有印的話就裝上去，免得遺失

循線感測器

- 可不裝



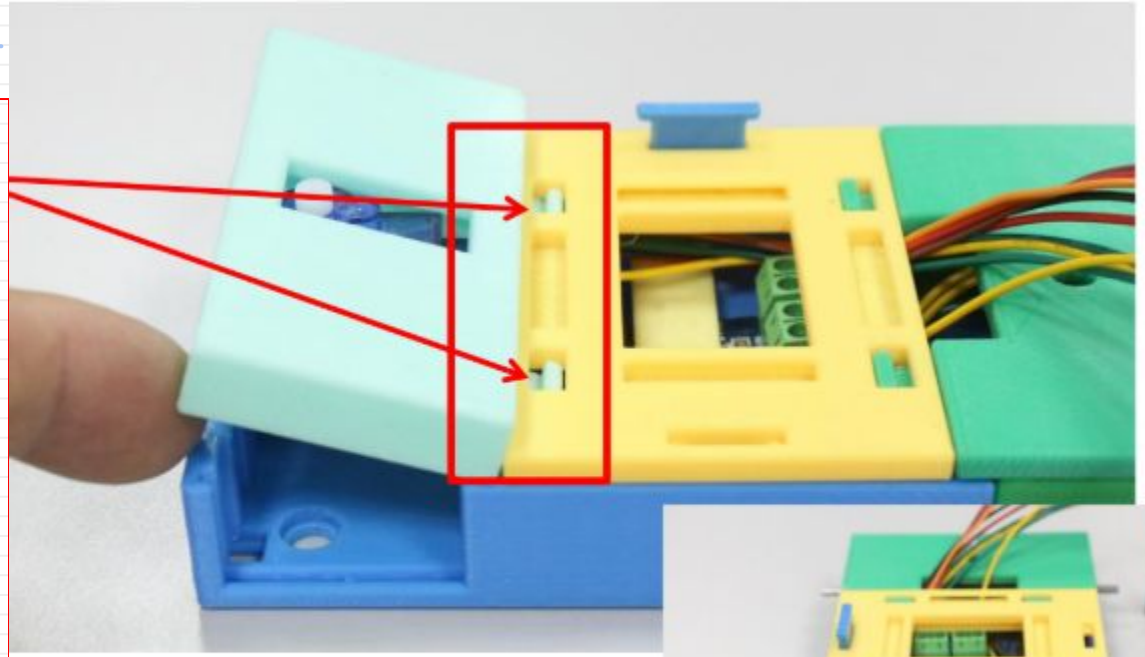




SG90蓋板

步驟

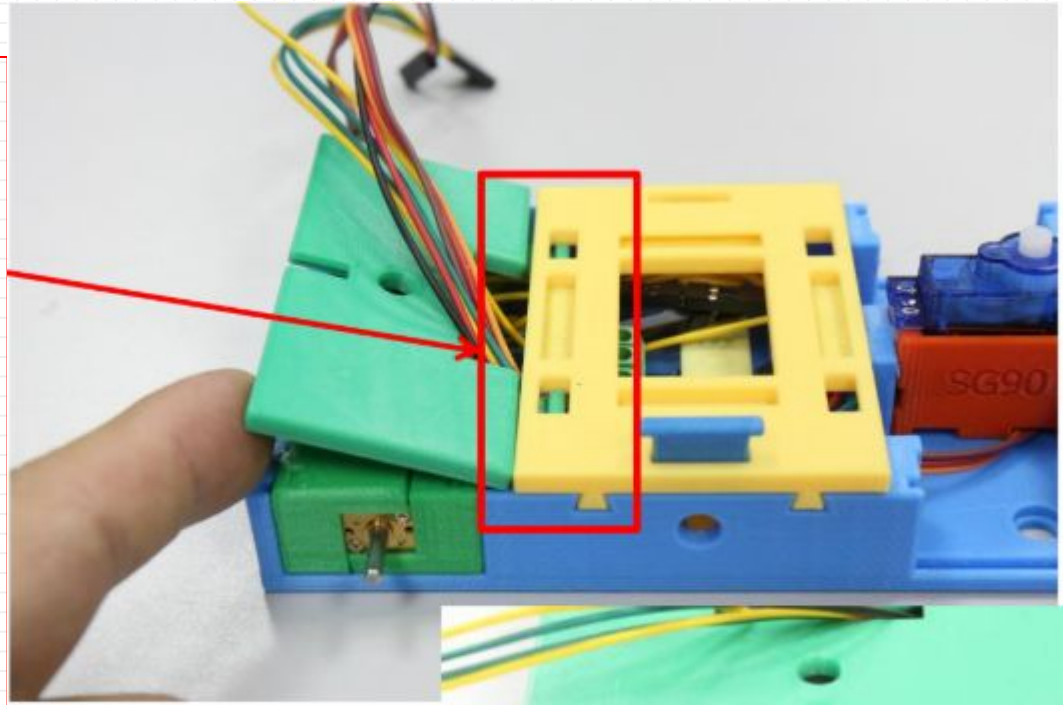
1. 卡榫勾住中蓋板溝槽
2. 往外拉、往下壓
3. 將鳩尾榫進入機殼凹槽時往裡推



N20蓋板

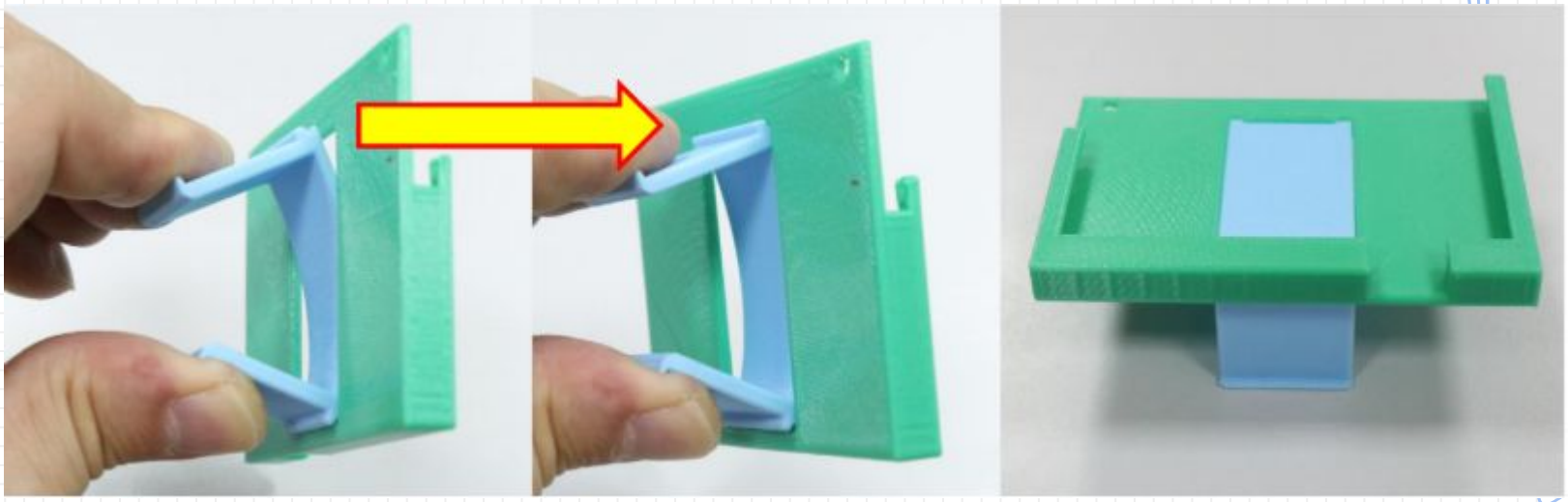
步驟

1. 將線整好集中在中間凹槽。
2. 卡榫勾住中蓋板溝槽
3. 往外拉、往下壓
4. 將鳩尾榫進入機殼凹槽時往裡推



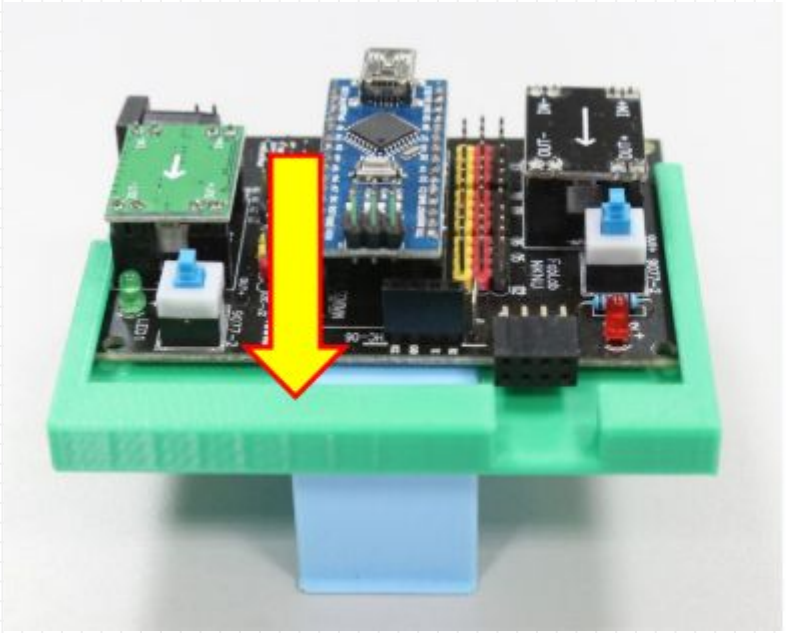
控制板座

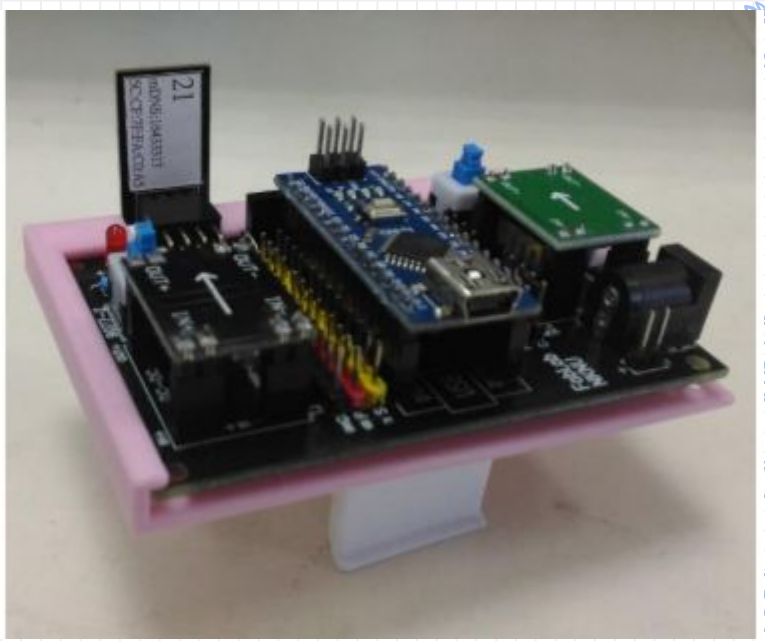
- 將控制板座壓彎穿過底座中間，對準兩邊卡榫後再放開，就可卡住。



WIFI 黑色電控板

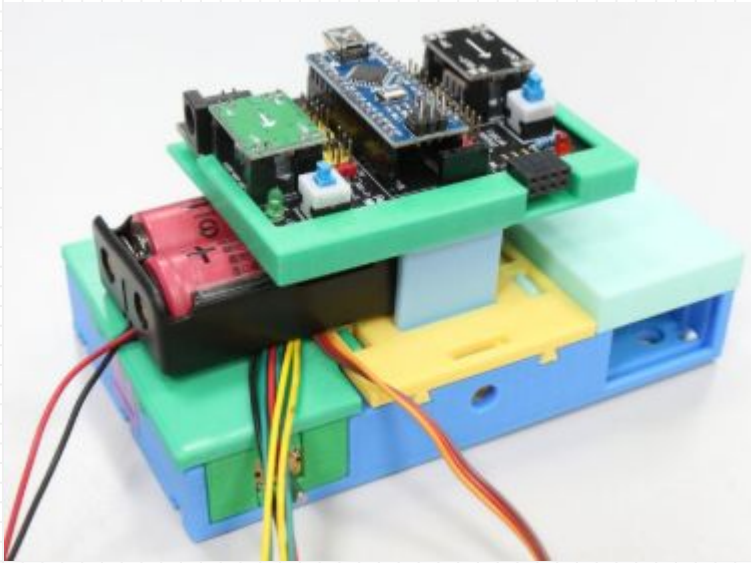
- 先裝控制板，再裝 ESP-01 WIFI 無線網卡。

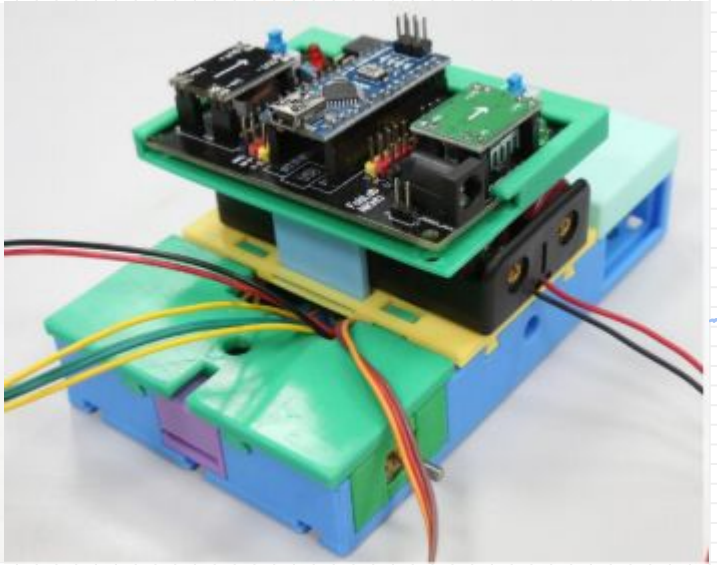




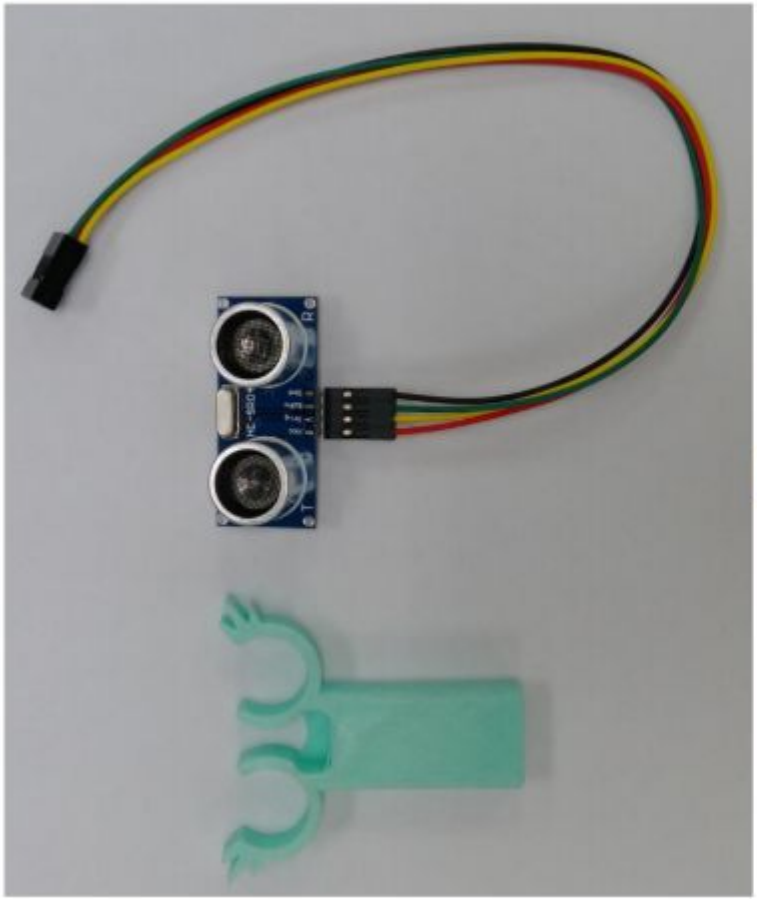
組合控制板、車身與電池

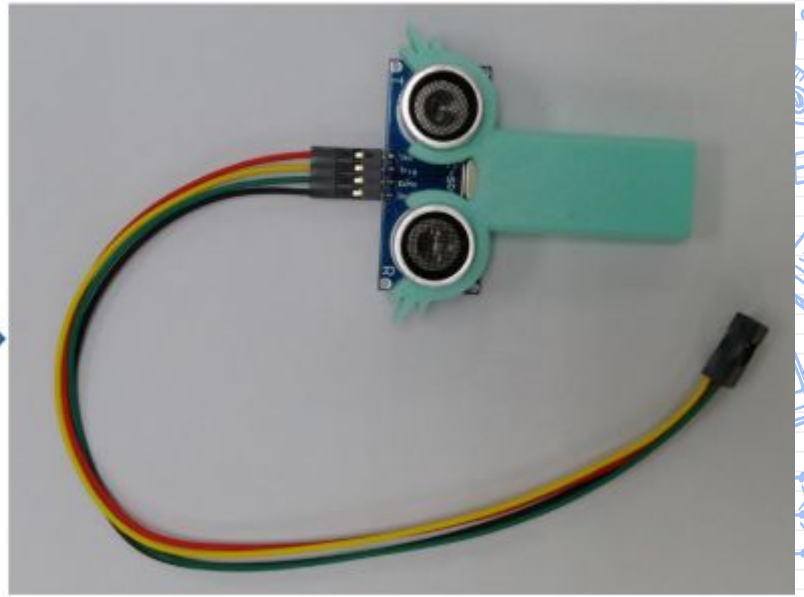
❖





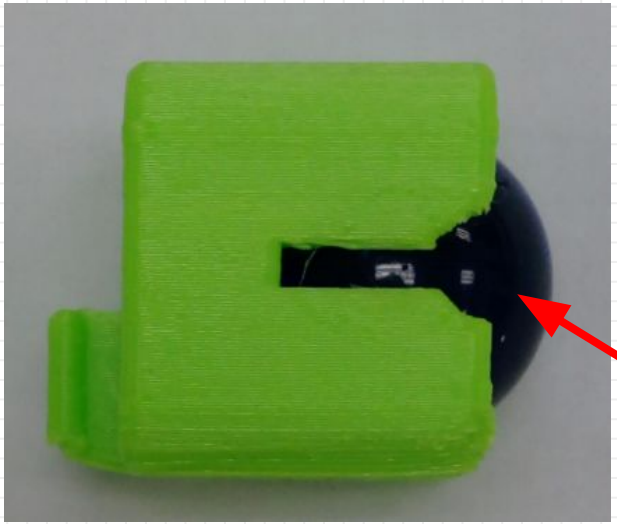
HC-SR04 超音波模組





惰輪

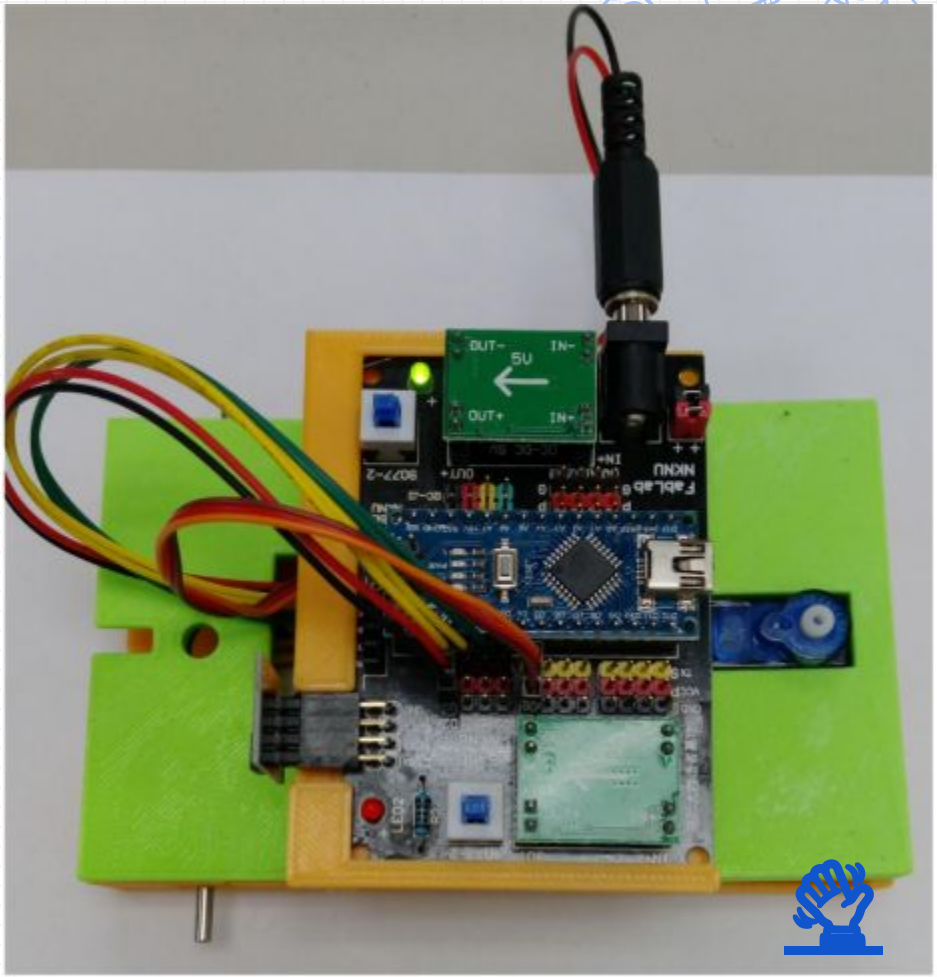
- 萬向輪



小彈珠
文具店售

電路接線

- L9110S
  - 左輪黃色　D2,D3~
  - 右輪綠色　D4,D5~
  - Vcc 紅色　5V 紅色針腳
  - GND 黑色　GND 黑色針腳
- SG90
  - 訊號橘色　D6
  - Vcc 紅色　5V 紅色針腳
  - GND 黑色　GND 黑色針腳

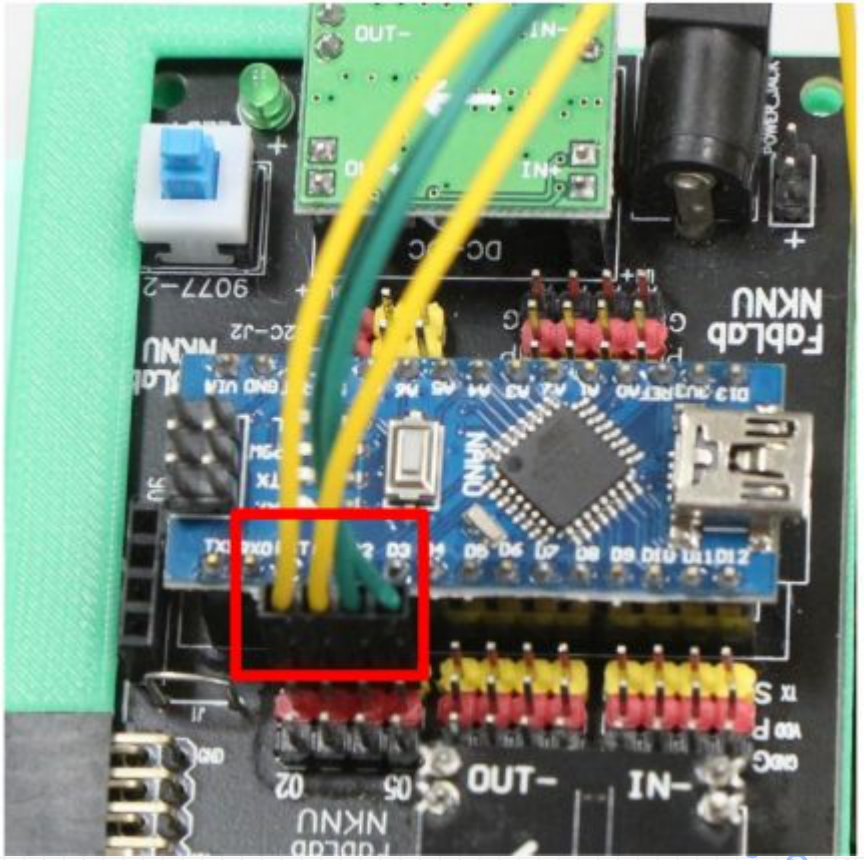


接線 - 馬達控制模組訊號線

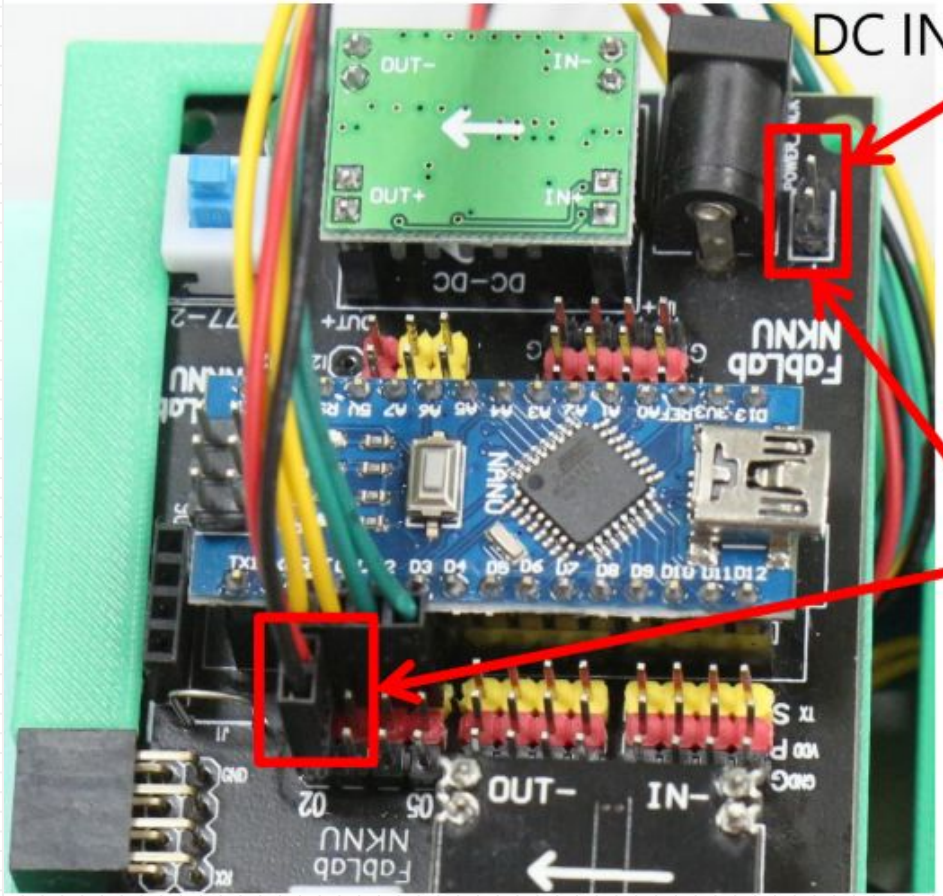
- 順序:

  黃→綠

  D2→D5



接線 - 馬達控制模組電源



可接在數位腳或類比腳的VCC與Gnd腳位(5V)
若要求高電壓高速度，也可接在DC IN，無論接在哪裡，都要**注意正負極不能接反**

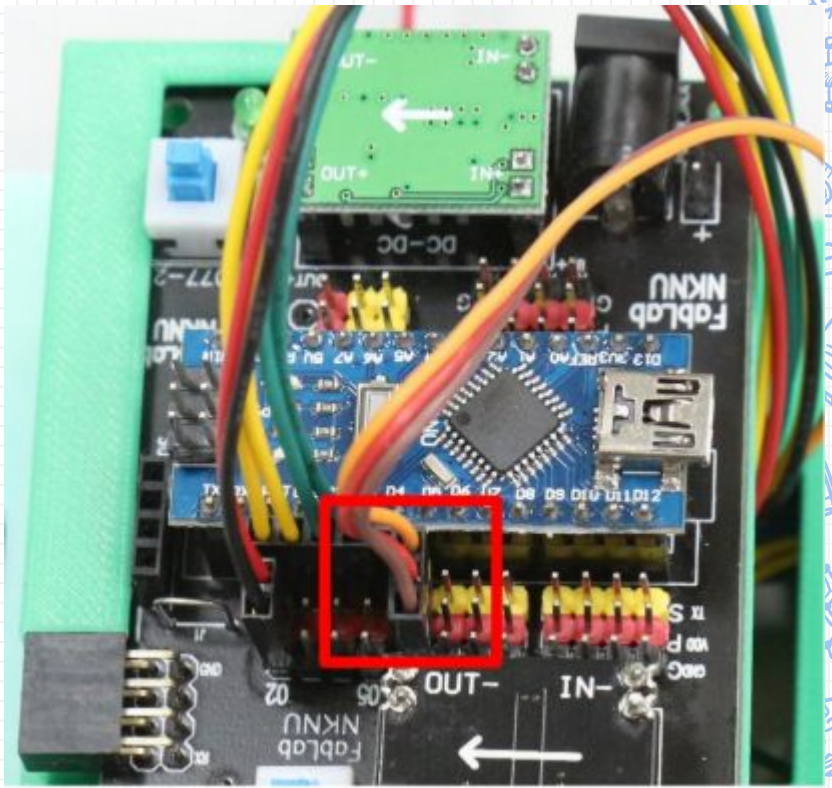
接線 - SG-90

- 接數位腳位6, 不可接反

橘→黃色訊號腳

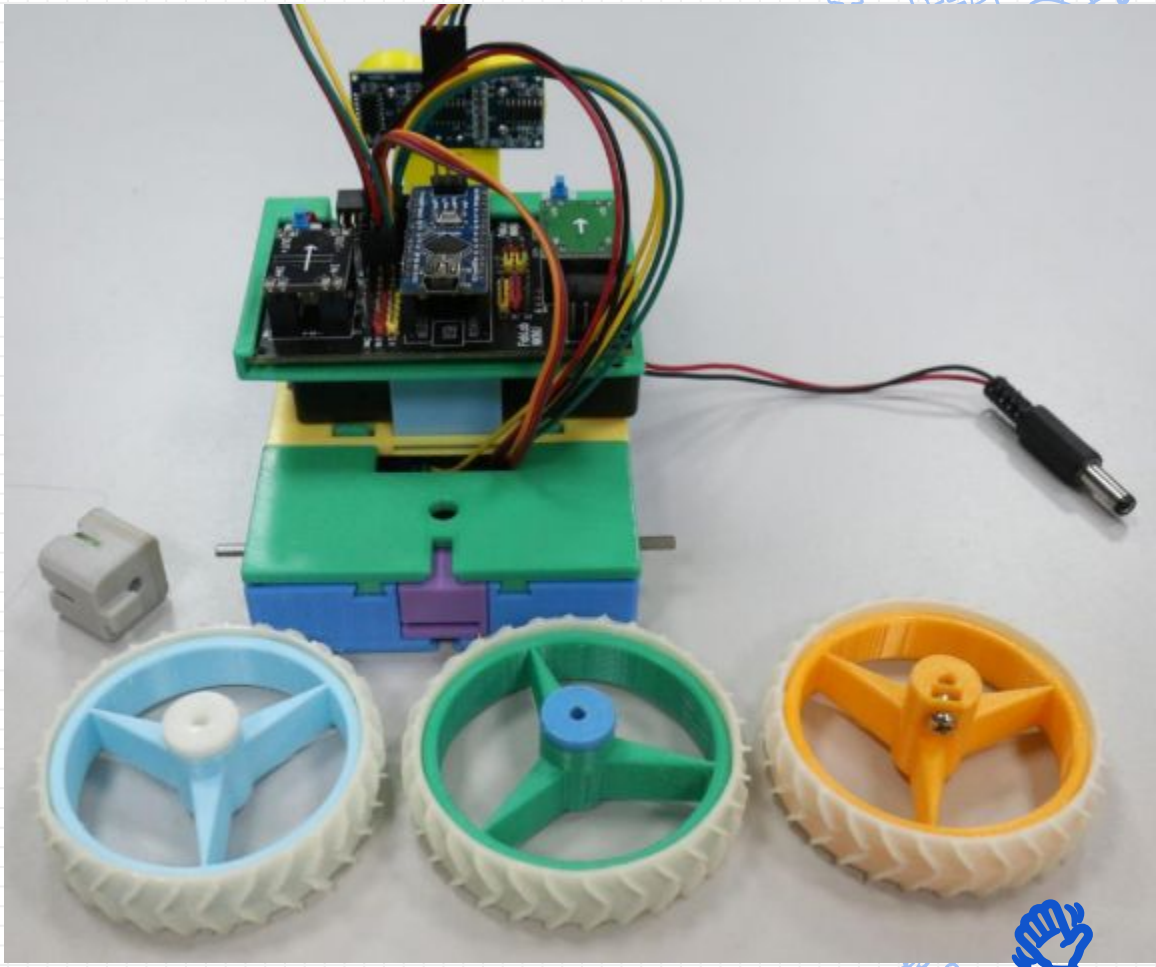
紅→紅色Vcc腳

棕→黑色GND腳



車輪

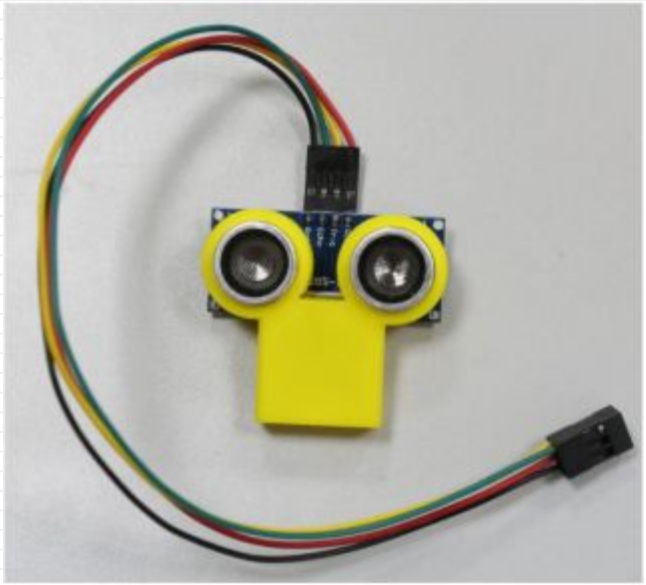
- 輪框軸心不要摩擦到車輪，要留一點空隙

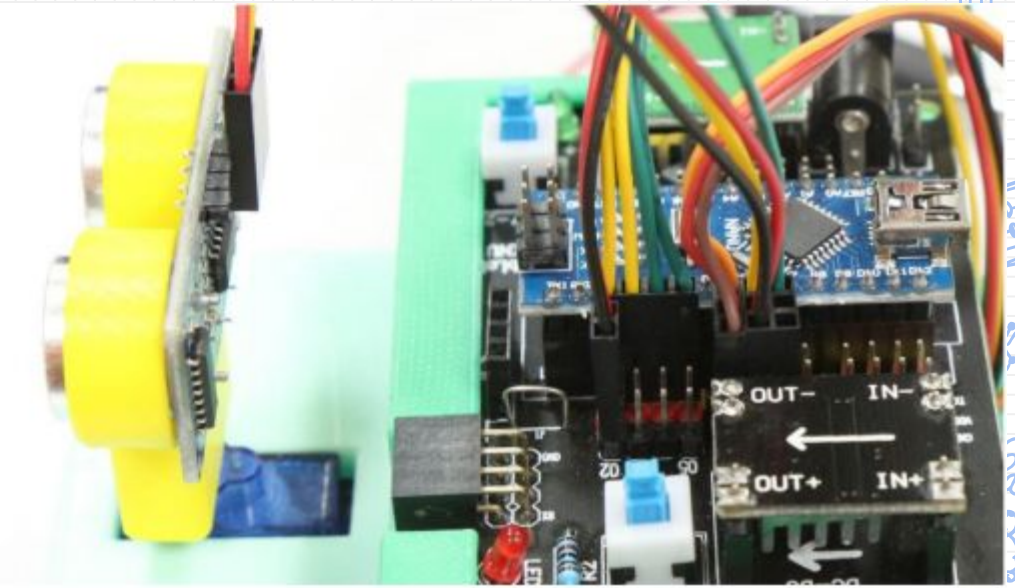


惰輪

接線 - HCSR04超音波

- Trig(黃)接7, Echo(綠)接8
- 伺服馬達先歸零(轉到90度), 再裝超音波感測器





超音波

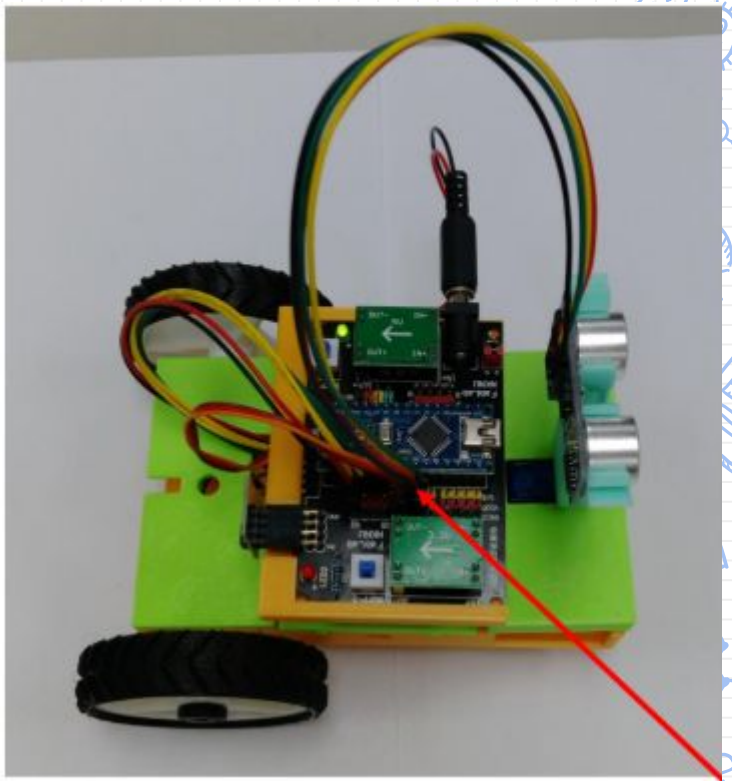
- HCSR04超音波

  Trig　　D7

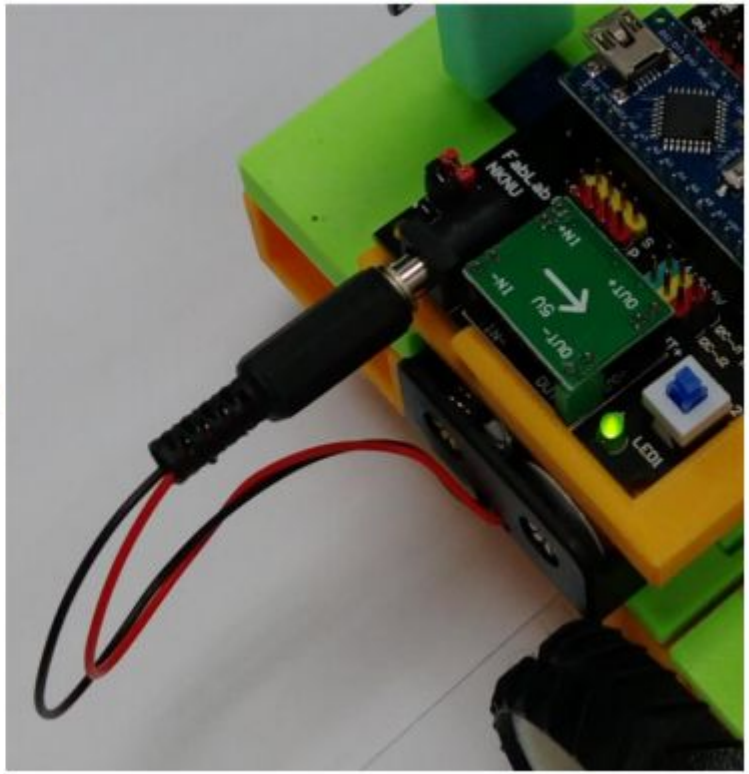
  Echo　　D8

  Vcc 紅色　　5V 紅色針腳

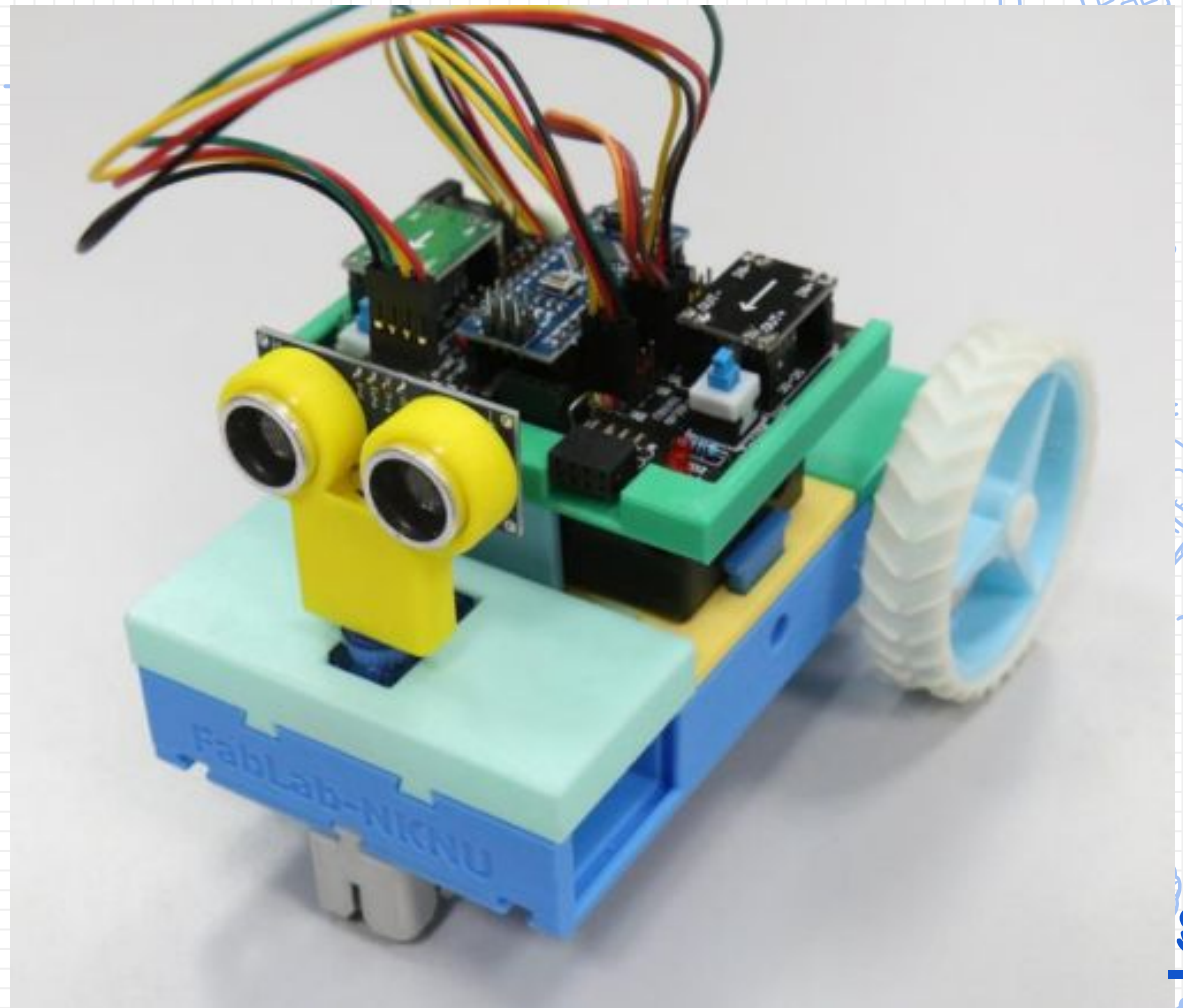
  GND 黑色　　GND 黑色針腳



組裝電源



完成圖



USB 接頭類型

USB 的接頭以外型區分可分為Type-A、Mini-A、Micro-A、Type-B、Mini-B、Micro-B、Type-C等型態, 以及 USB 3.0 Type-B 與 USB 3.0 Micro-B。目前市售的行動裝置除iPhone/iPad外大都採用Micro-B 插頭(又稱microUSB)。USB Type-C針腳數量最多, 且無須對應特定接頭方向, 正、反插都能使用。

USB 接頭類型

Type-A 公

Type-A 公

Type-A 公

mini-A 公

Type-A 公

Type-B 公

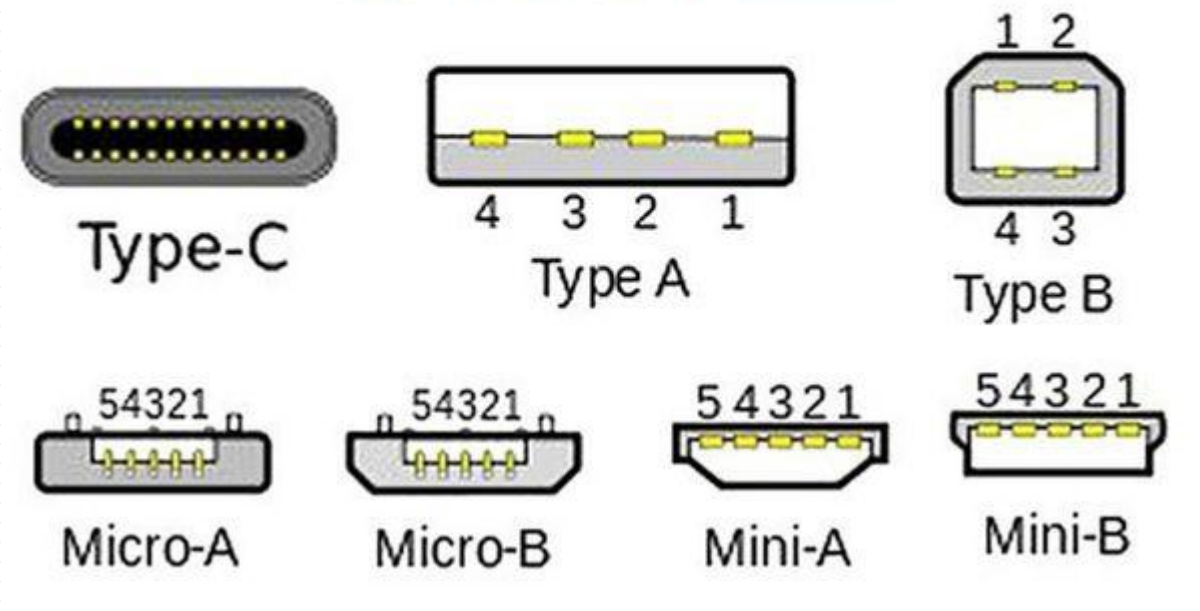
Type-A 公

micro-B 公

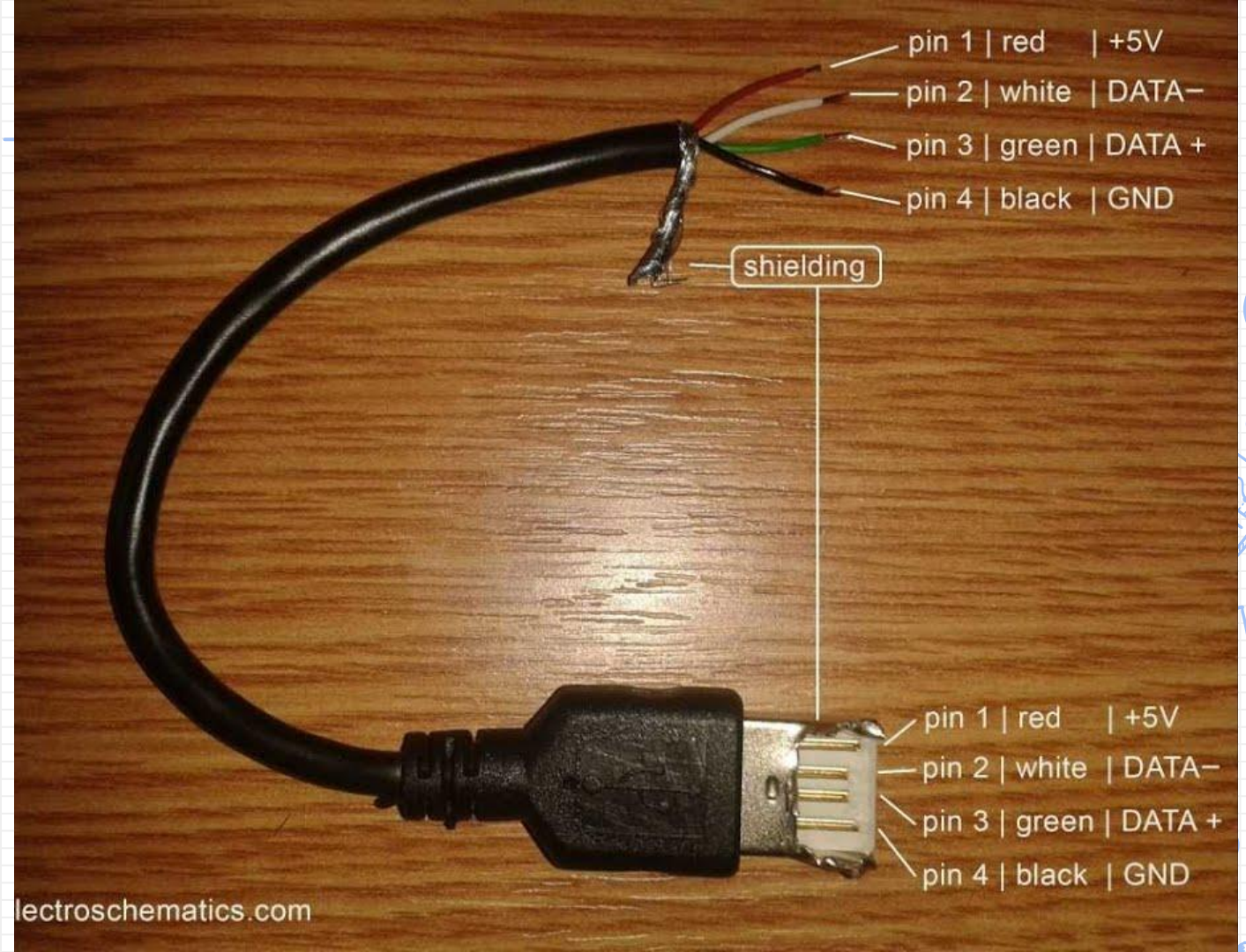
Type-A 公

Type-C 公

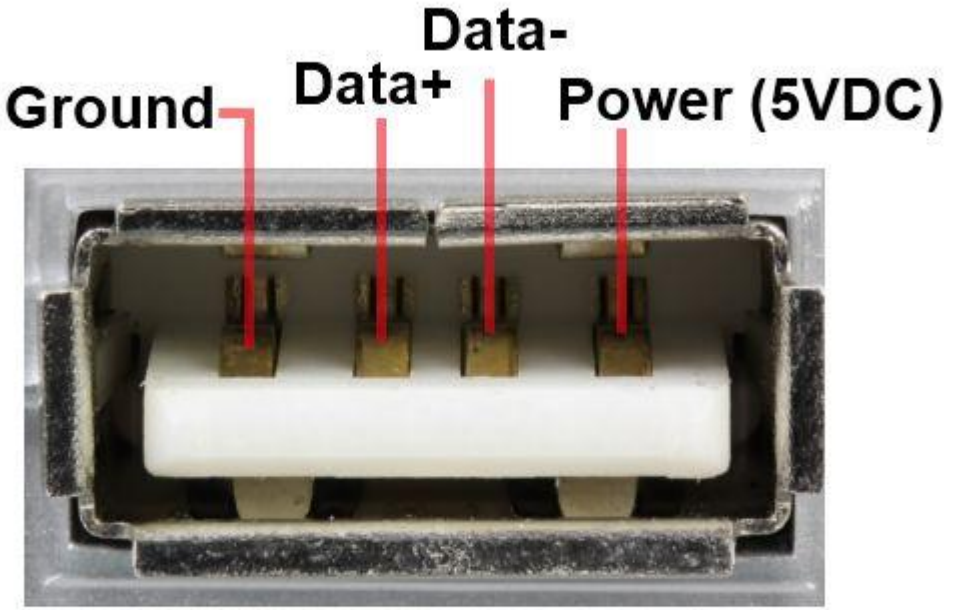
USB 接頭腳位


Type-C
4 3 2 1
Type A
1 2
4 3
Type B
54321
Micro-A
54321
Micro-B
54321
Mini-A
54321
Mini-B

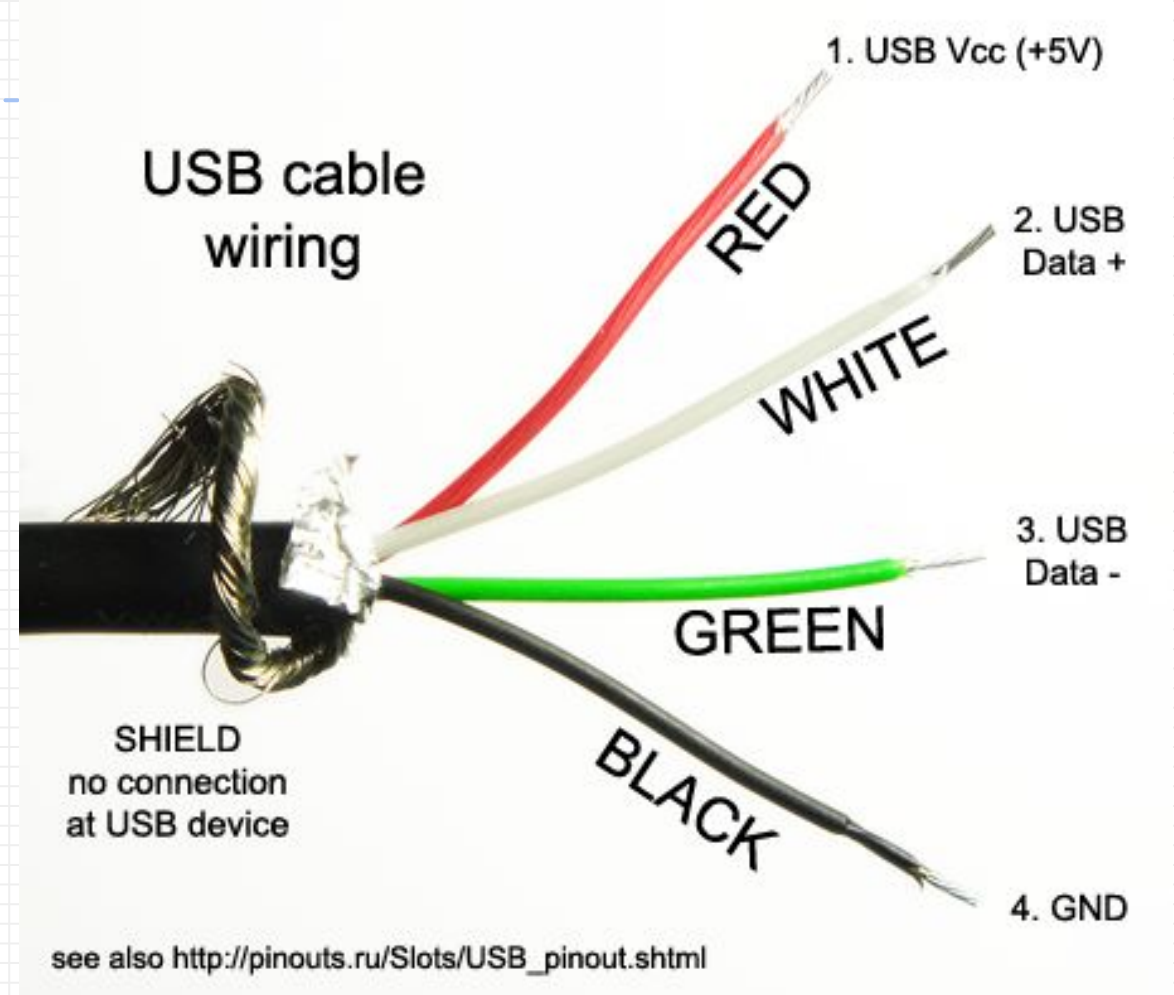
Type-A 公


pin 1 | red | +5V
pin 2 | white | DATA−
pin 3 | green | DATA +
pin 4 | black | GND
shielding
pin 1 | red | +5V
pin 2 | white | DATA−
pin 3 | green | DATA +
pin 4 | black | GND
lectroschematics.com

Type-A 公



USB 線


USB cable
wiring
1. USB Vcc (+5V)
RED
2. USB
Data +
WHITE
3. USB
Data -
GREEN
SHIELD
no connection
at USB device
BLACK
4. GND
see also http://pinouts.ru/Slots/USB_pinout.shtml

USB 電源接頭

- 自焊 USB 電源接

A型 USB接口（公口）

A型 USB接口（母口）

B型 USB接口（公口）

B型 USB接口（母口）

USB引脚定义：

引脚	功能	颜色	备注
1	V Bus	红	电源正5V
2	Data-	白	数据-
3	Data+	绿	数据+
4	GND	黑	地

A型 Mini-USB接口（公口）

B型 Mini-USB接口（公口）

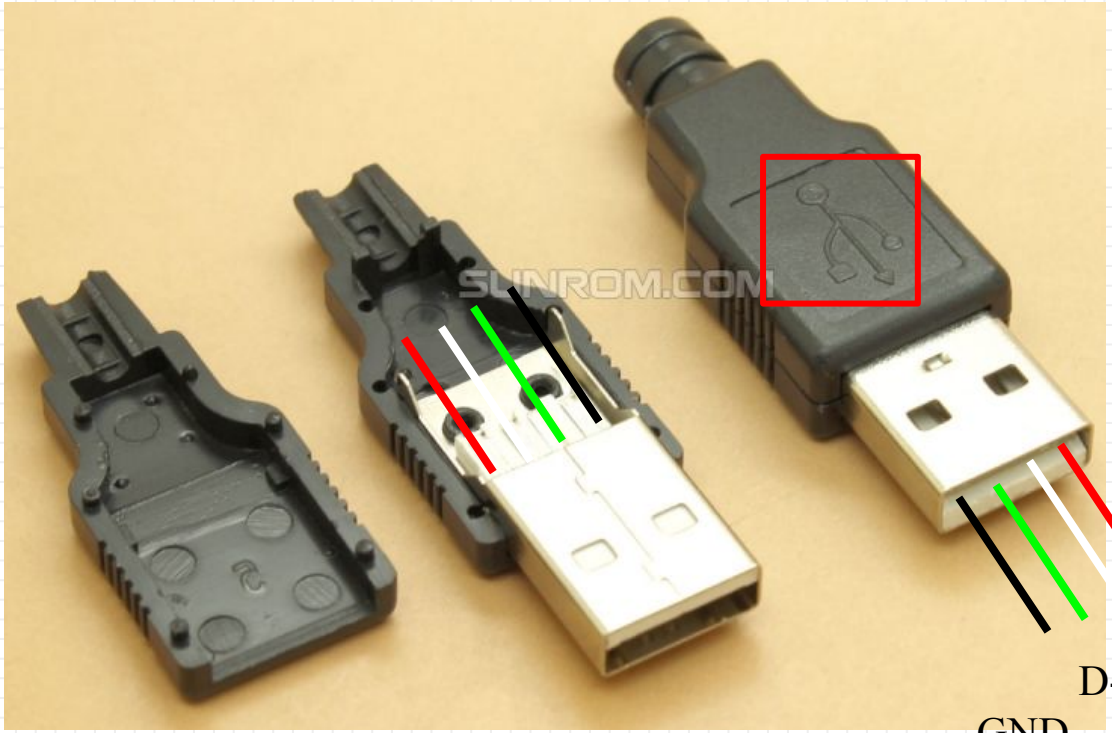
Mini-USB引脚定义：

引脚	功能	颜色	备注
1	V Bus	红	电源正5V
2	Data-	白	数据-
3	Data+	绿	数据+
4	ID		A型:与地相连
			B型:不接地（空）
5	GND	黑	地

Mini-USB接口（母口）通用

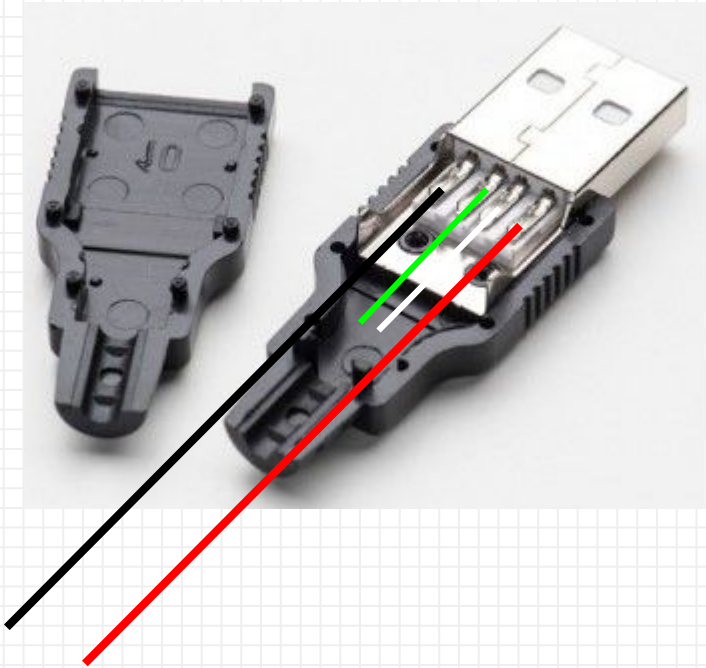
自焊 USB 電源接頭

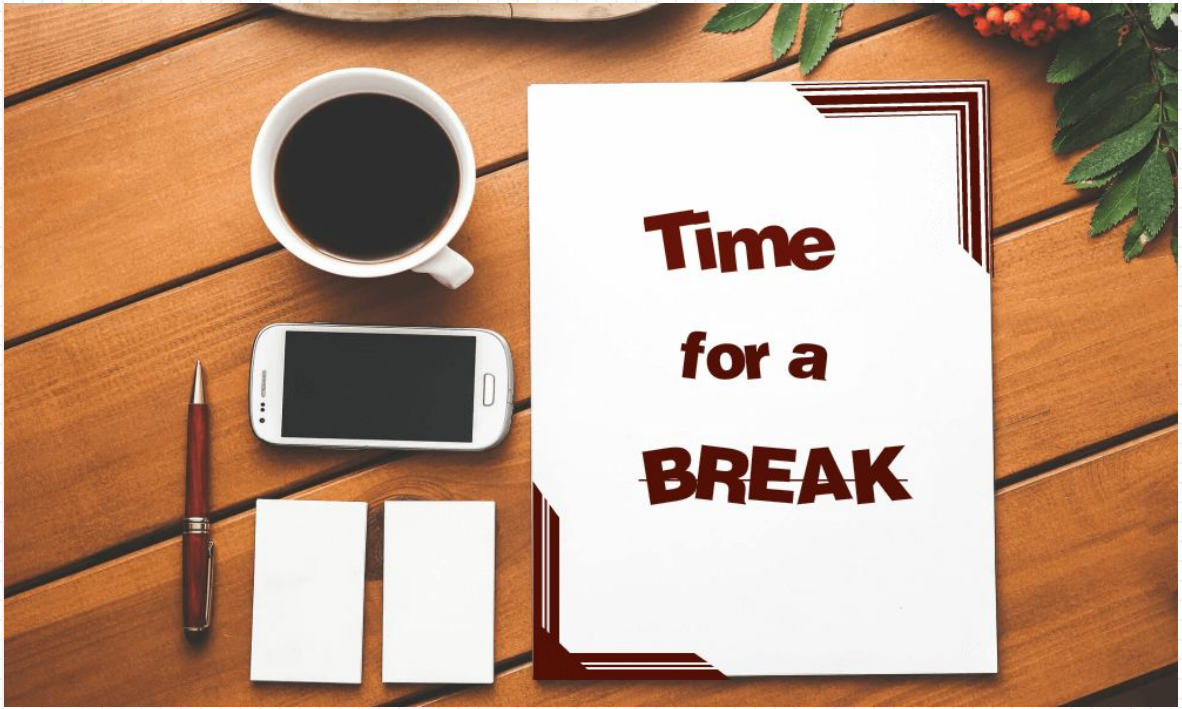
USB 電源接頭



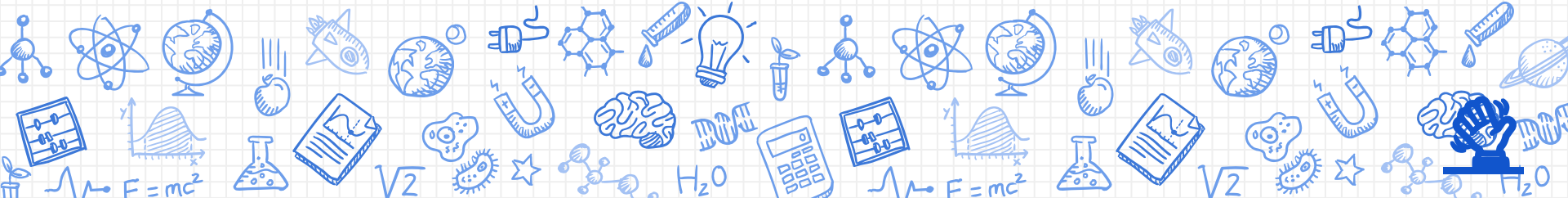
USB 電源接頭

- 自焊 USB 電源接頭



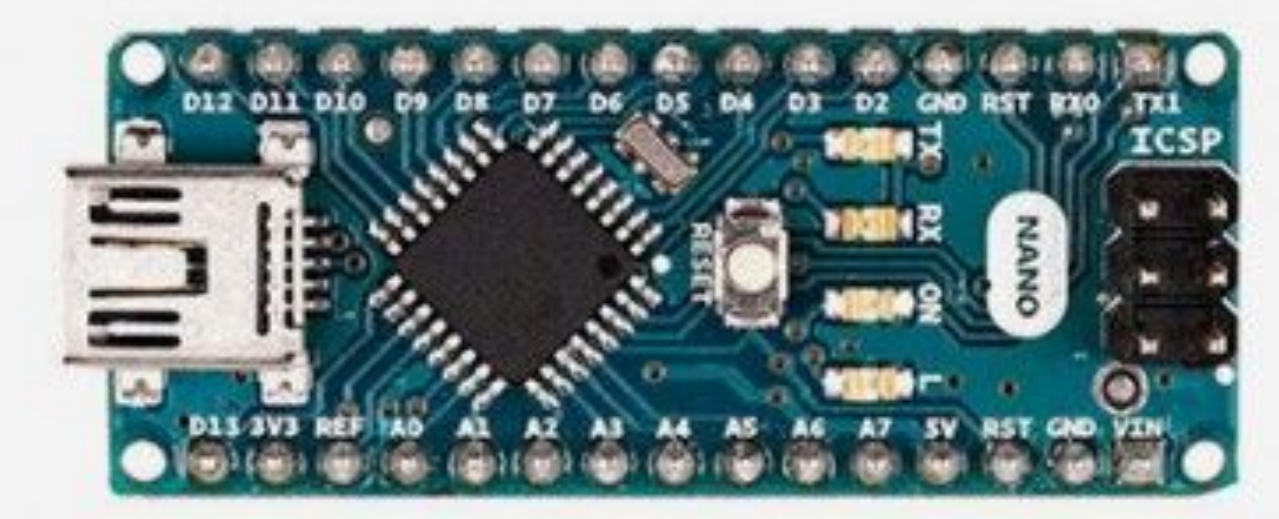

Time
for a
BREAK

4060小車控制



NANO

- NANO


D12 D11 D10 D9 D8 D7 D6 D5 D4 D3 D2 GND RST RX0 TX1
ICSP
TX
RX
ON
NANO
RESET
D13 3V3 REF A0 A1 A2 A3 A4 A5 A6 A7 5V RST GND VIN

NANO

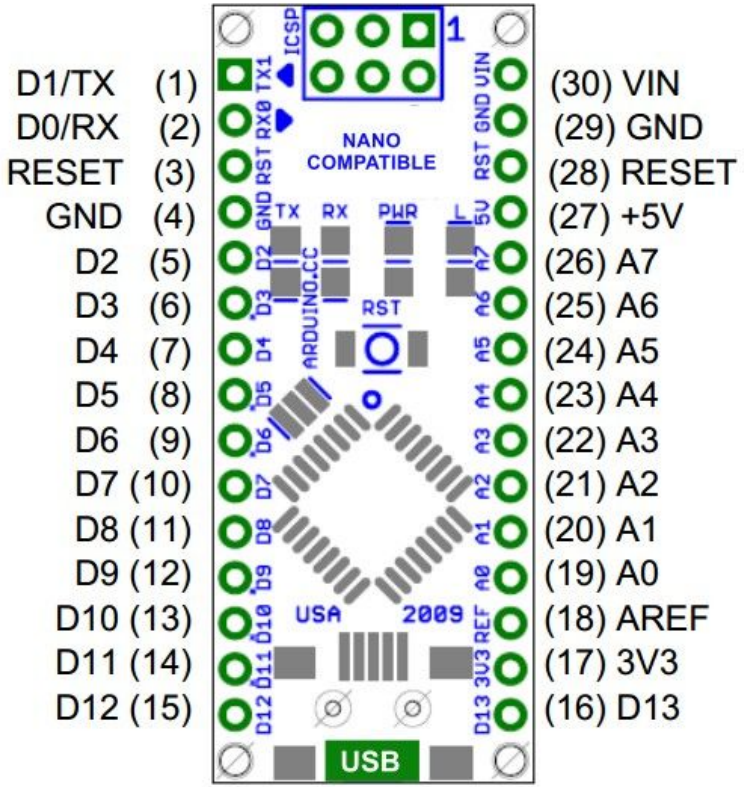
- Microcontroller：ATmega328P
- Architecture：AVR
- Operating Voltage：5V
- Input Voltage：7-12 V
- Digital I/O Pins：14 (of which 6 provide PWM output)
- PWM Digital I/O Pins：6 (3, 5, 6, 9, 10, 11)
- Analog Input Pins：8 (A0~A7, 其中A0~A5也可當數位腳位(D14~19用)
- UART：1。
- PCB Size：18 x 45 mm

NANO

- DC Current per I/O Pin : 40 mA
- Flash Memory : 32 KB (ATmega328P) of which 2 KB used by bootloader
- SRAM : 2 KB (ATmega328P)
- EEPROM : 1 KB (ATmega328P)
- Clock Speed : 16 MHz
- LED_BUILTIN : 13

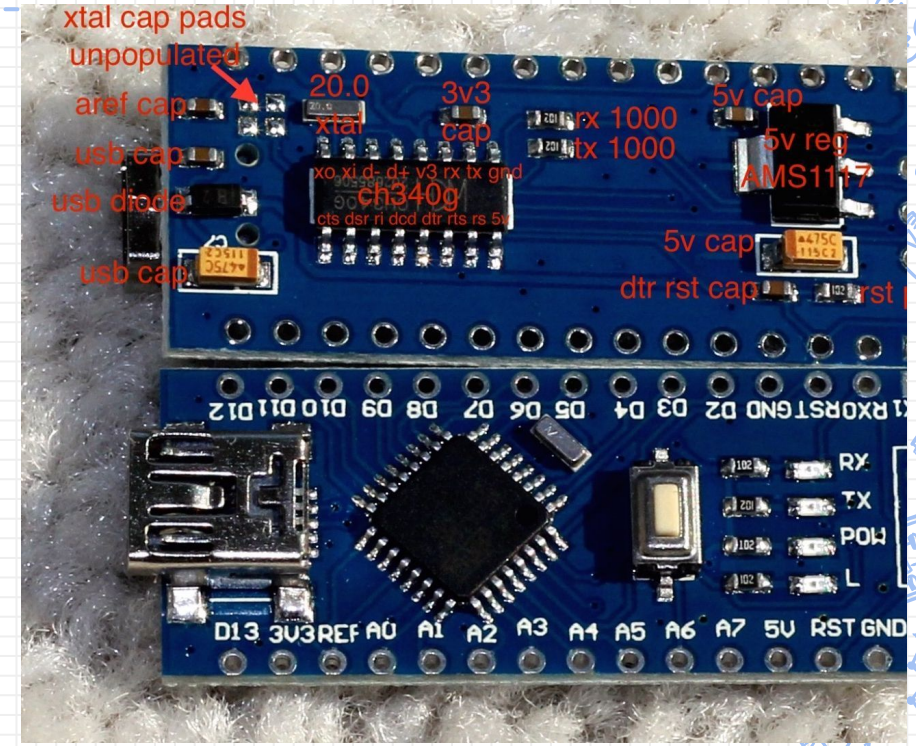
NANO 正面

- 數位輸出/輸入
- 類比輸入
- 5V
- 3.3V
- GND
- TX/RX
- 內建LED
- Vin
- mini-B USB 接口



NANO 反面

- 5V 電壓調節IC(AMS1117)
- USB晶片(CH-340G)



NANO MIC CLONE

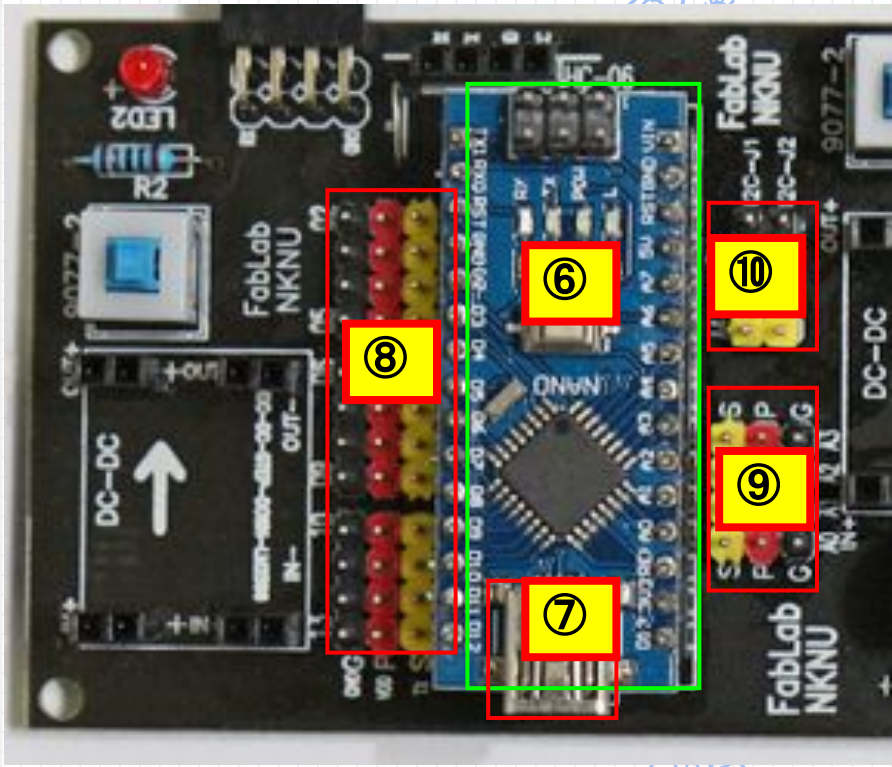
- 中國CLONE的NANO, 使用的CH340G USB連接晶片, 部分WINDOWS OS未提供驅動, 使用時要另外安裝CH340系列的驅動程式(上網搜尋[CH341SER.EXE]安裝)。
- 參考下載點:http://www.wch.cn/download/CH341SER_EXE.html

NANO USB 連接線

- A-USB
- mini-B USB

4060 WIFI 黑色電控板

- 8). 數位I/O+5V供電擴充接口
  :D2~D13, PWM:3,5,6,9,10,11
- 9). 類比I/O+5V供電擴充接口:A0~A3
- 10). I2C擴充接口:
  A4(SDA), A5(SCL)

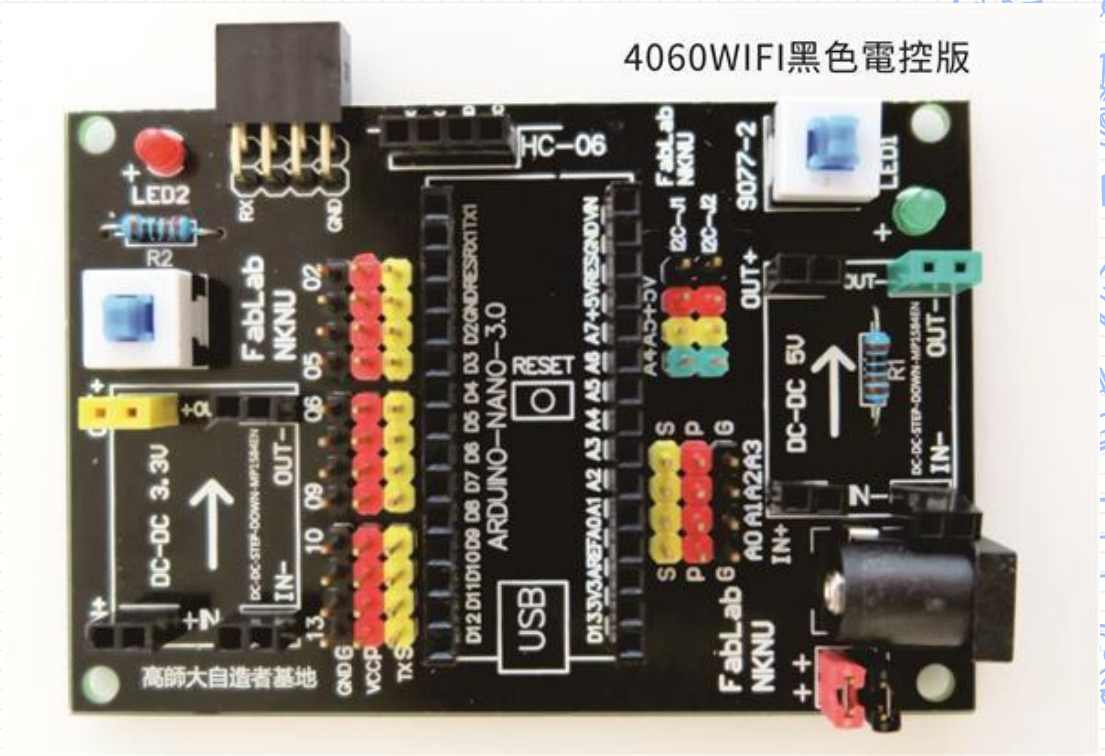


4060 WIFI 黑色電控板

- 紅:供電 Vcc(5V)

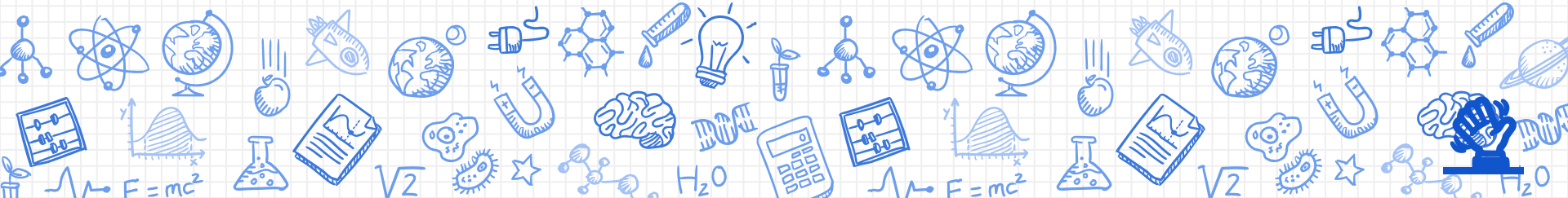
  黑:供電 GND

  黃:訊號



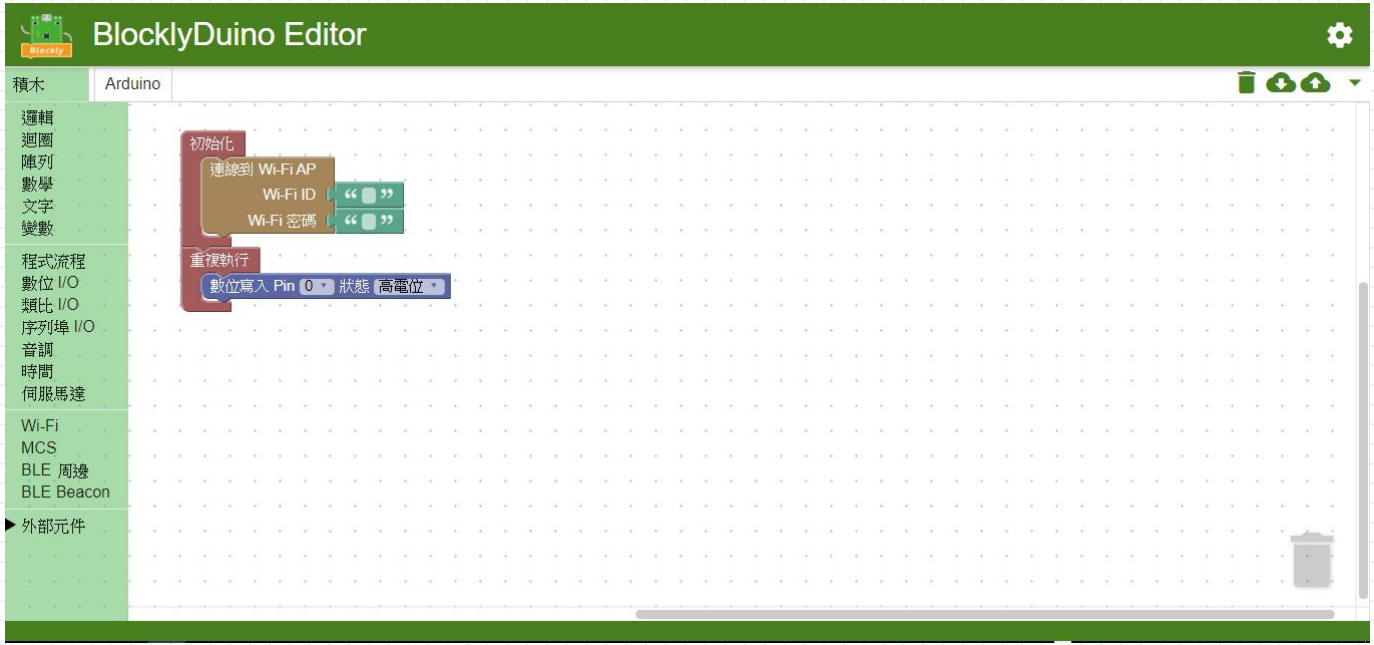
Blockly

積木程式設計環境



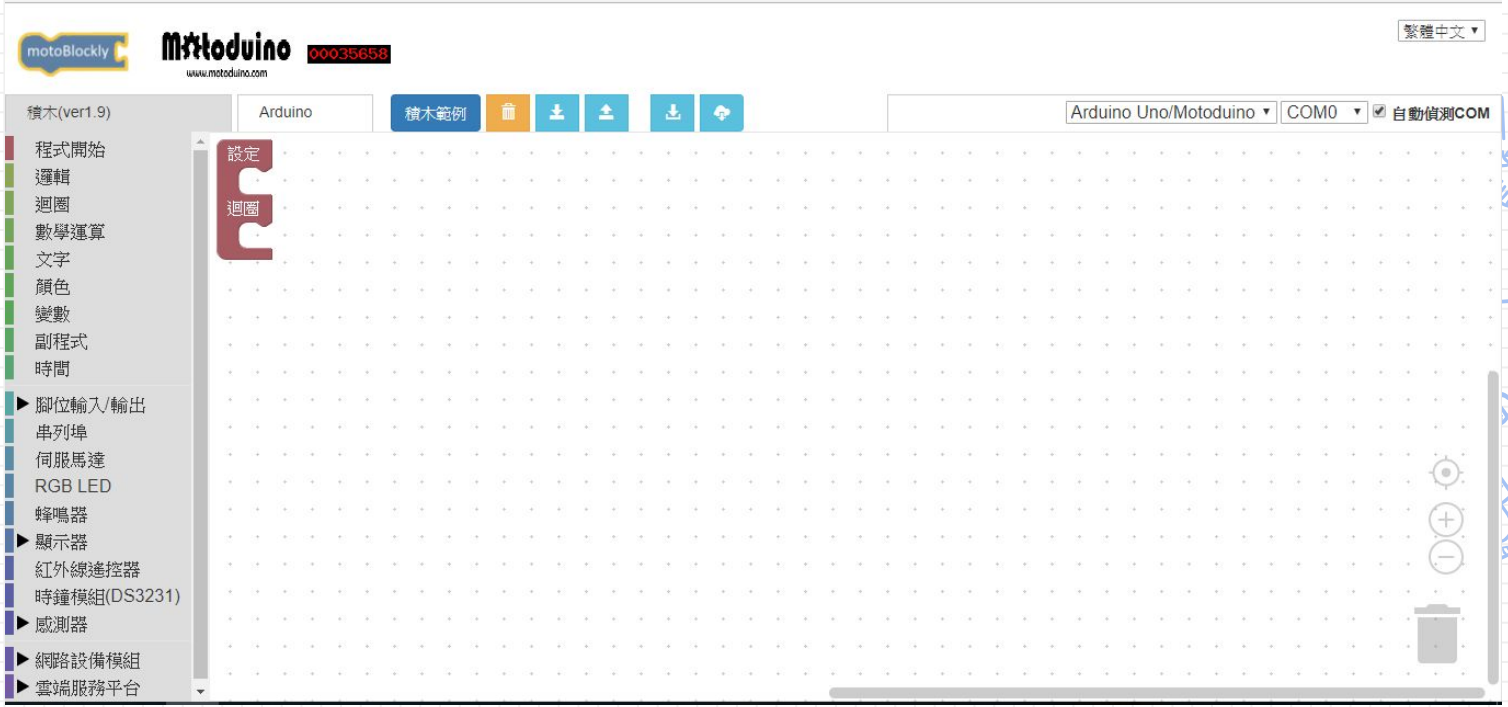
BlocklyDuino-for-LinkIt

- https://mediatek-labs.github.io/BlocklyDuino-for-LinkIt/



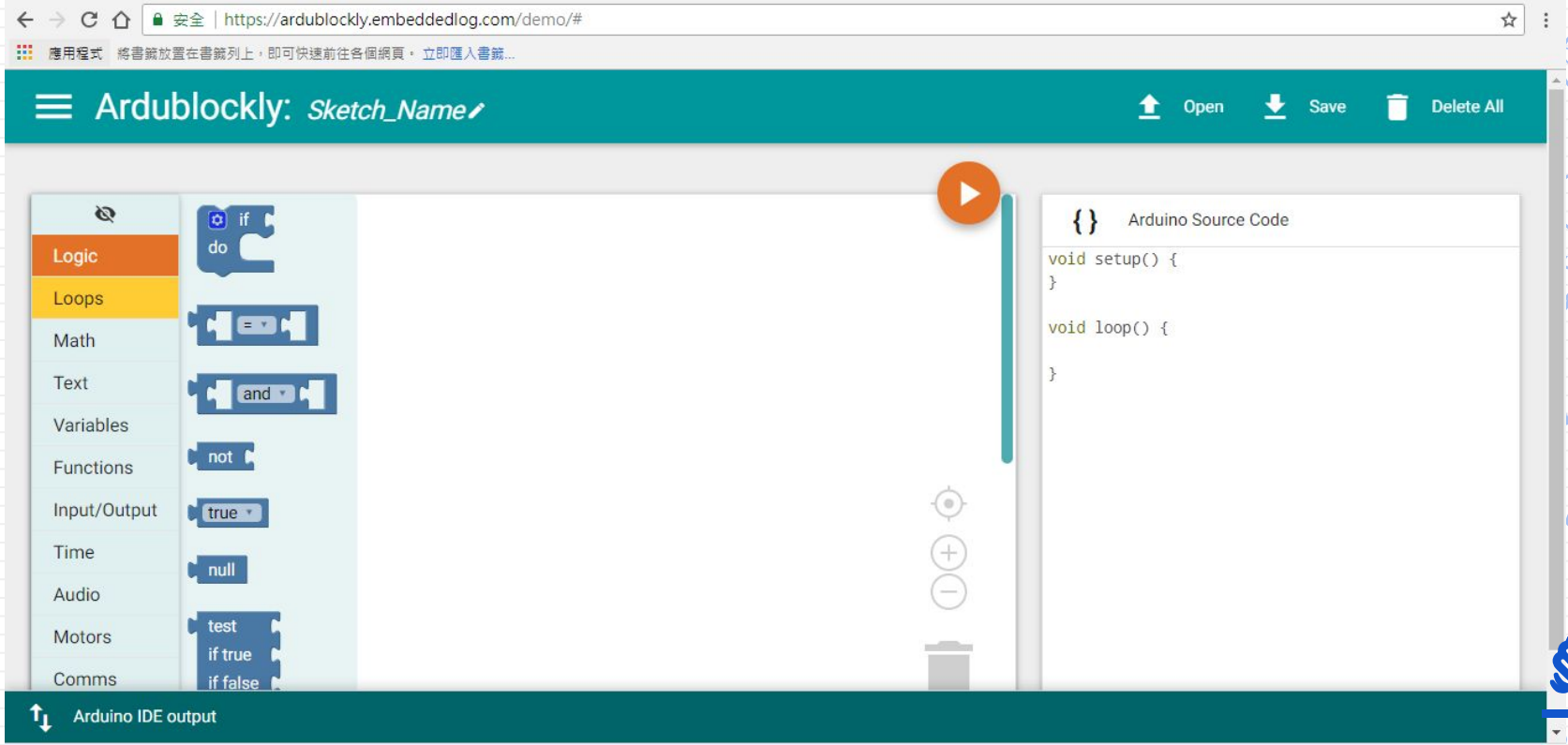
motoBlockly

- https://www.motoblockly.com
- Ver 1.9



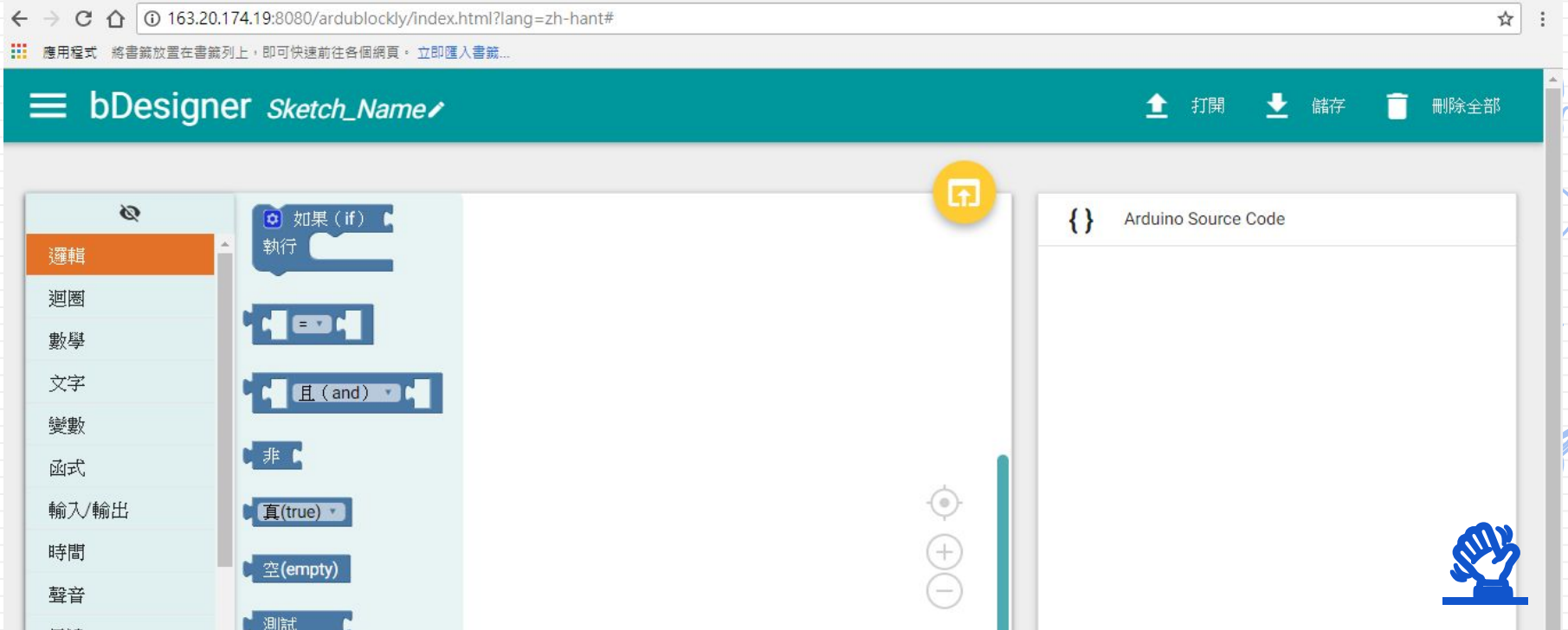
Ardublockly

- https://ardublockly.embeddedlog.com/demo/#



bDesigner(blockly)

- http://163.20.174.19:8080/ardublockly/index.html?lang=zh-hant#

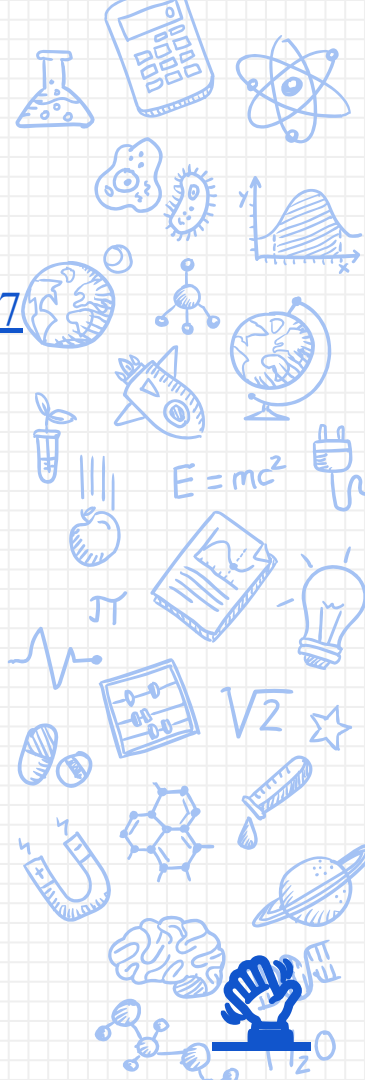


bDesigner(blockly)

- https://bdesigner1.webnode.tw/bdesigner-blockly/
- bDesigner 以 Ardublockly 為基礎, 建置了一套可以編譯nodeMCU的blockly, 這套積木軟體有C語言積木跟簡易積木兩種, 讓你可以用C語言積木, 自己編譯出網路功能, 也可以讓你用幾個簡易積木, 就能做出網路功能, 當然, 你也可以混合使用, 以做出適合你的程式, 目前放在台北縣樹林國小網站, 大家可以前往使用

bDesigner(blockly)

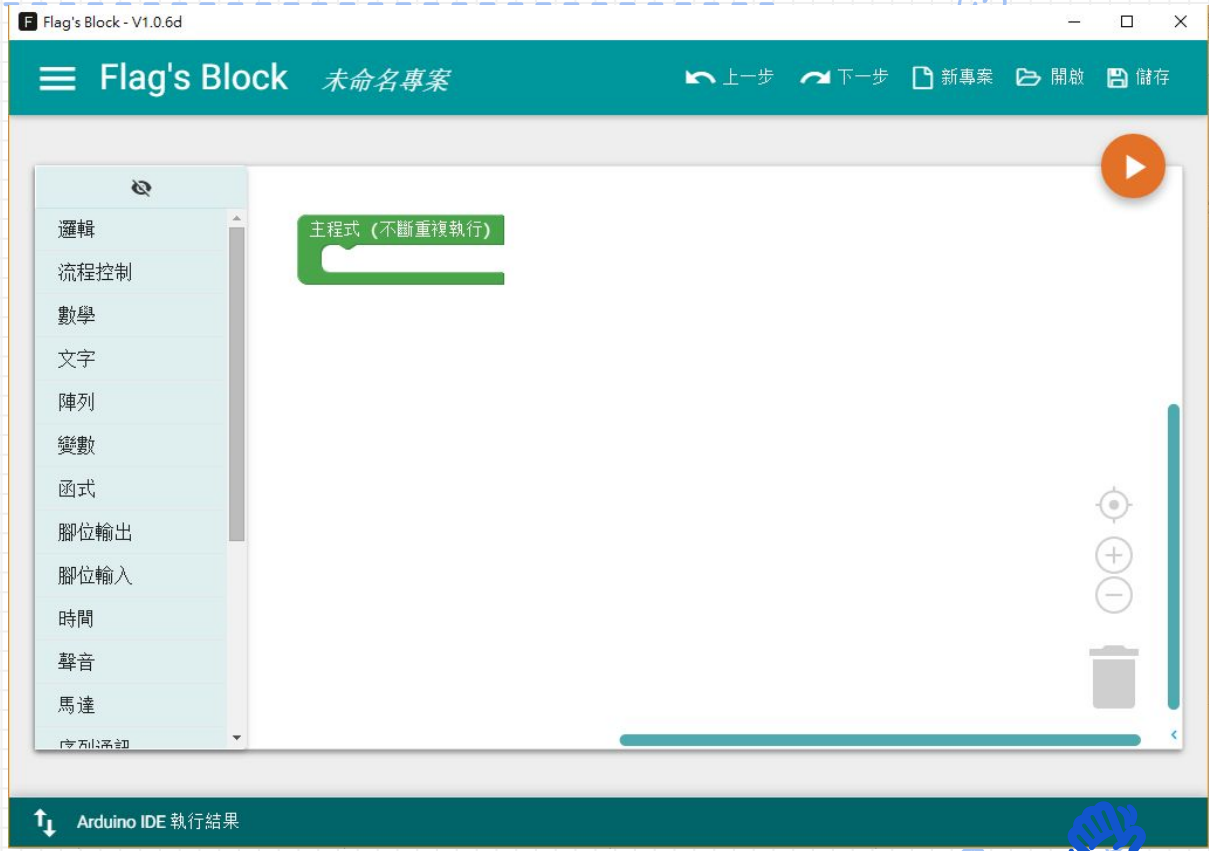
- 下載：bDesigner(blockly)0.1版，bDesigner 0.1.zip
  https://drive.google.com/file/d/1XwrgroIy7oA3G7K9i4VBKOgpvVYw8a7K/view?usp=sharing



Flags Block

- 旗標科技
- FlagsBlock_v1.1.3b

  FlagsBlock.zip



Flags Block

- 旗標科技
- 下載:FlagsBlock.zip

  http://www.flag.com.tw/book/FM606_reg.asp

  http://www.flag.com.tw/book/FM604_reg.asp

  http://www.flag.com.tw/download.asp?FM606A
- FlagsBlock.zip,

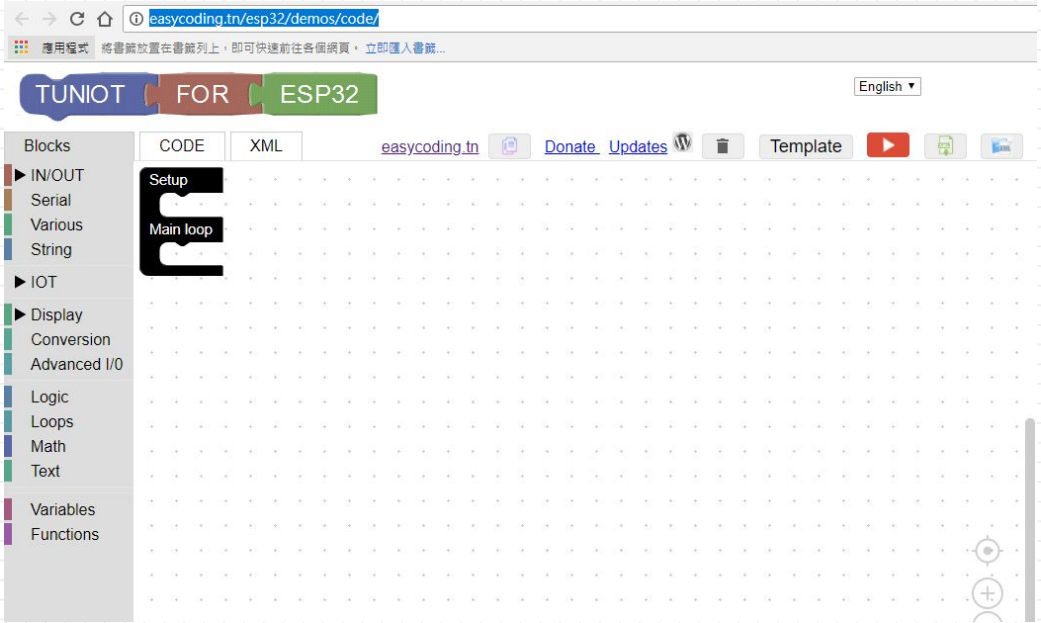
  FlagsBlock_v1.0.6D.exe

TUNIOT

- TUNIOT For ESP8266
- http://easycoding.tn/tuniot/demos/code/

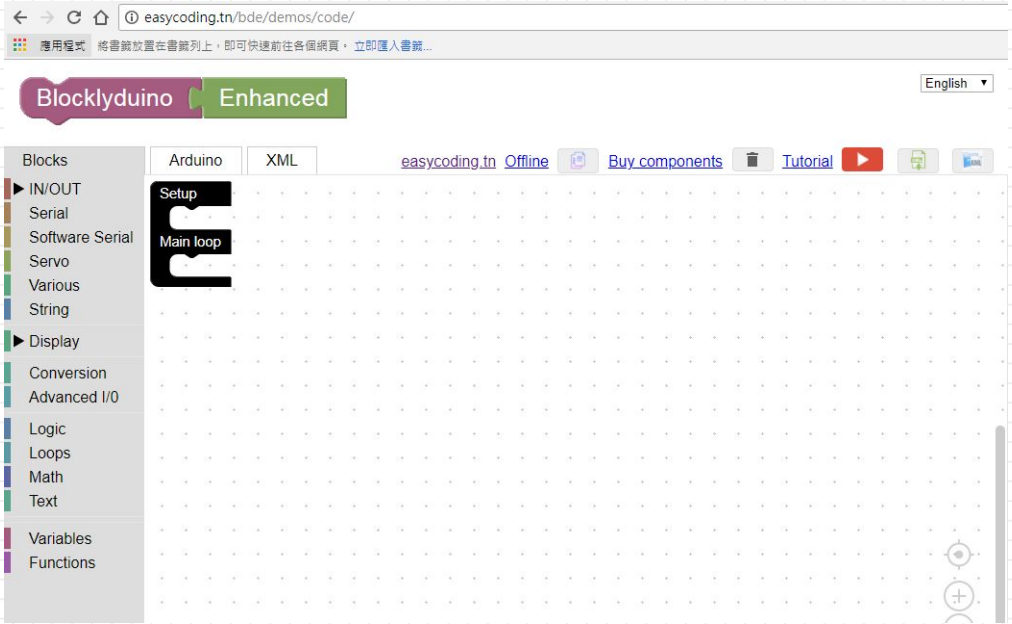
TUNIOT

- TUNIOT For ESP32
- http://easycoding.tn/esp32/demos/code/



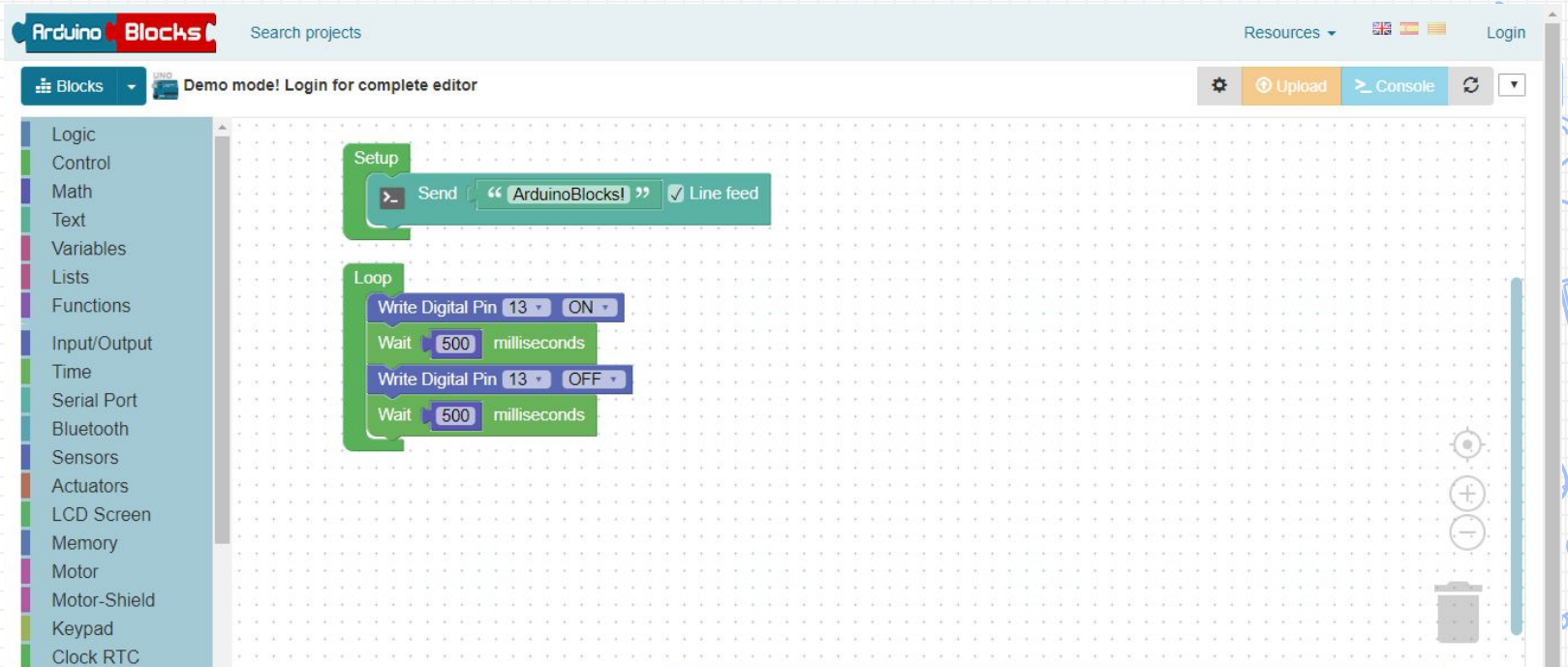
Blocklyduino Enhanced

- Blocklyduino Enhanced
- http://easycoding.tn/bde/demos/code/



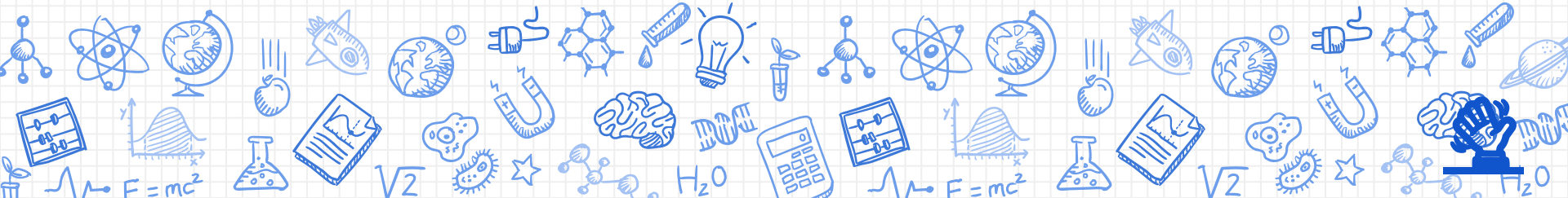
ArduinoBlock

- http://www.arduinoblocks.com/
- http://www.arduinoblocks.com/web/project/editordemo



Arduino IDE

免安裝USB開發環境



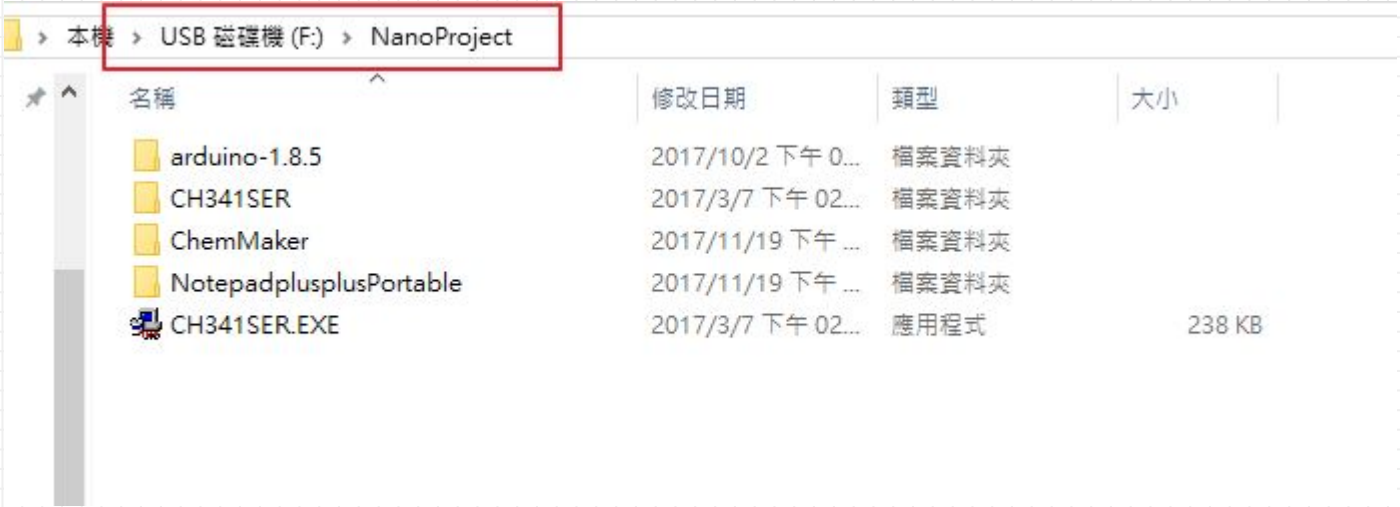
軟體開發環境

- 下載：https://goo.gl/SRMckf

  NKNUProject.zip

軟體開發環境

- 解壓縮 NKNUProject.zip

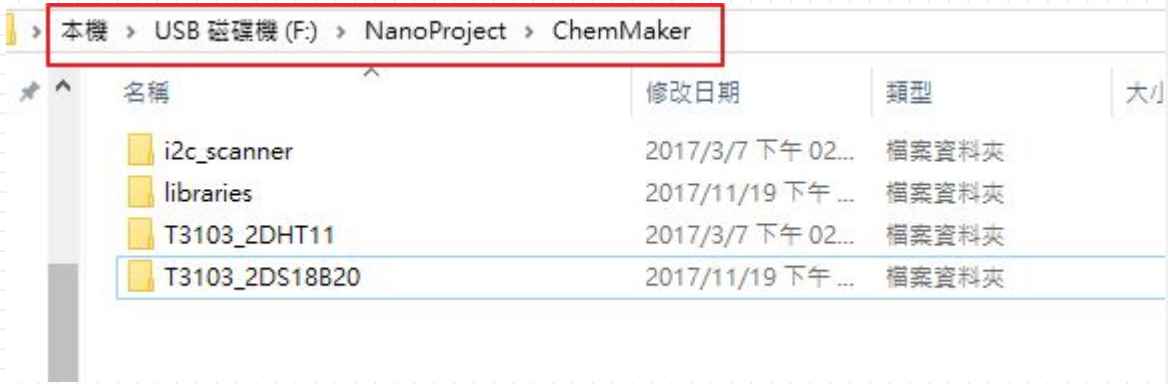


軟體開發環境

- (1) 資料夾 arduino-1.8.5：由https://www.arduino.cc/en/main/software 下載之zip檔。實際應用時，打開資料夾後，即可快速方便的使用，免去繁雜耗時的安裝過程，又可隨時更新版本。
- (2) 資料夾CH341SER，CH341SER.EXE：CH340/CH341的USB晶片WINDOWS驅動程式。筆電作業系統為Windows 10以前版本，需先安裝此驅動程式，否則系統無法抓到控制板，將無法進行程式燒錄，後續應用也無法進行了。

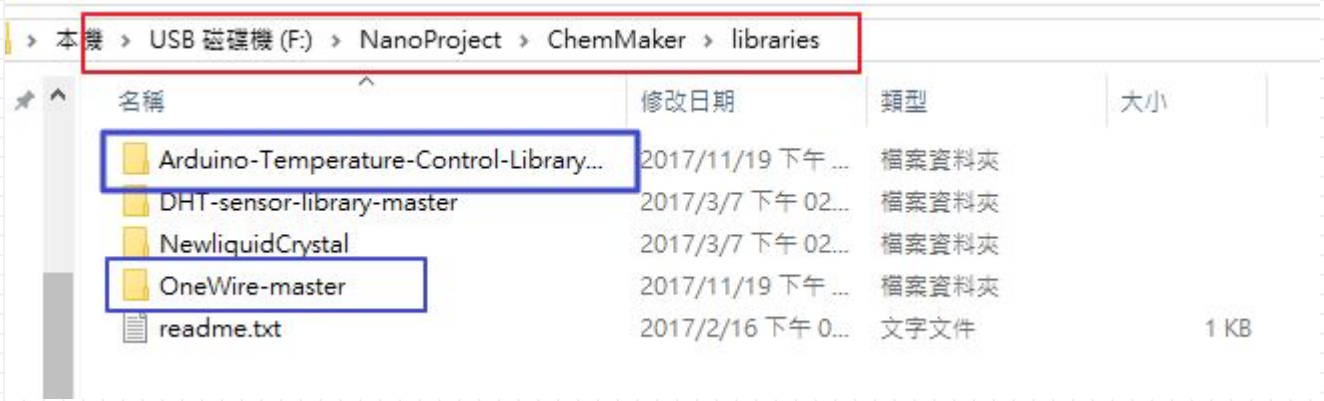
軟體開發環境

- (3) 資料夾MyMaker: 自行設計之相關程式與感測器函示庫存放於此資料夾。



軟體開發環境

- 資料夾 MyMaker\libraries：網路下載感測器函式庫，解壓縮後，事先置放於此，省去繁雜耗時的安裝。



軟體開發環境

- (4) 資料夾NotepadplusplusPortable：一款好用的程式碼編輯器程式，也可用於一般的文字編輯(取代Windows的記事本程式)

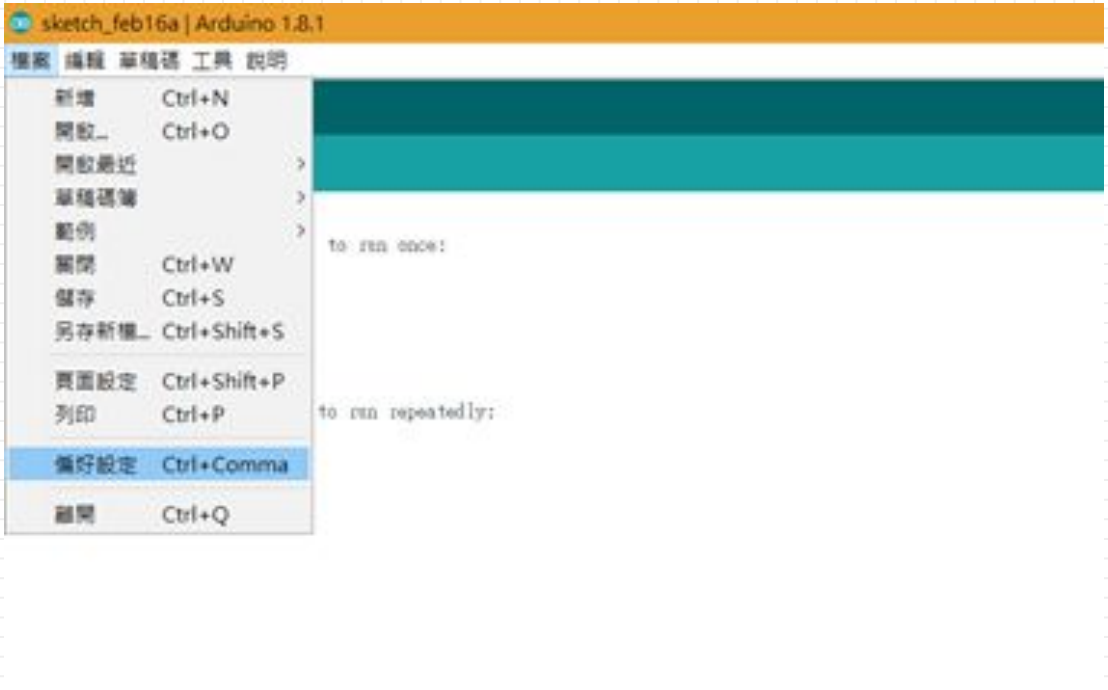
軟體開發環境

- 安裝 CH340 USB晶片驅動程式

  執行CH341SER\CH341SER\SETUP.EXE安裝
- 控制板以 USB 傳輸線接上電腦

設定電腦與控制板正常連線

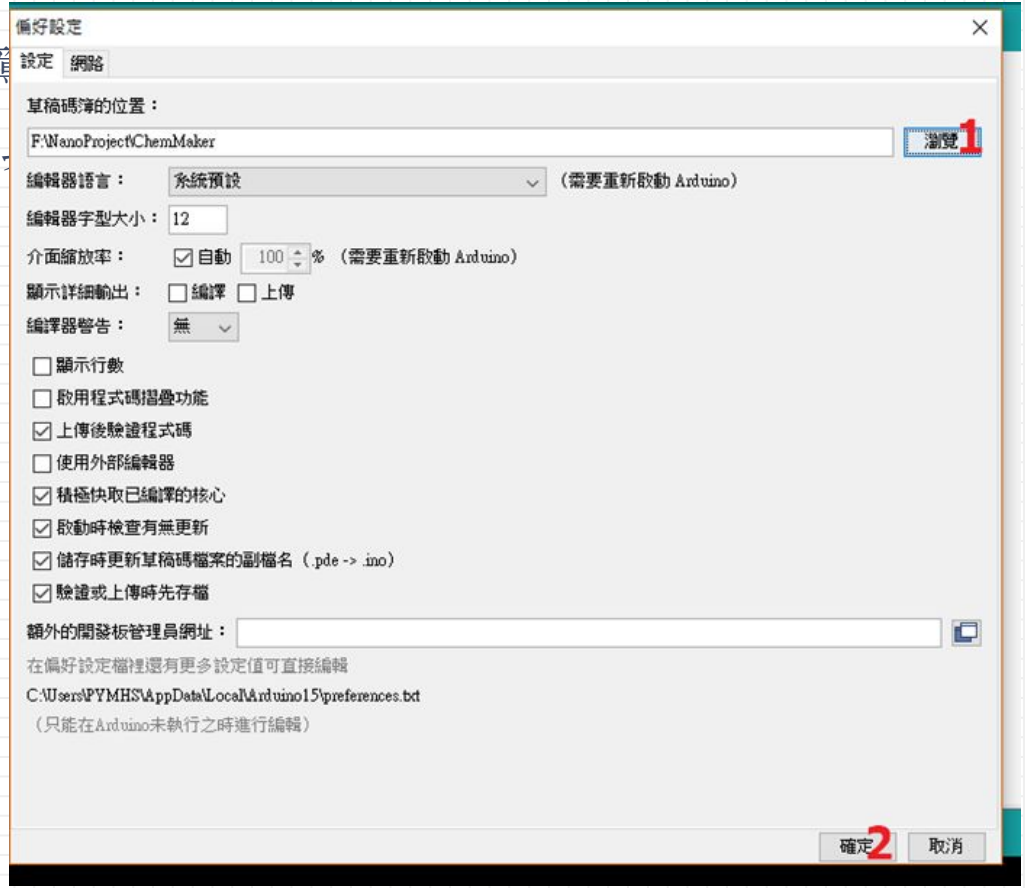
- 滑鼠雙按arduino-1.8.5/arduino.exe，開啟Arduino IDE介面
- 選檔案-偏好設定



設定電腦與控制板正常連線

- 於草稿碼簿

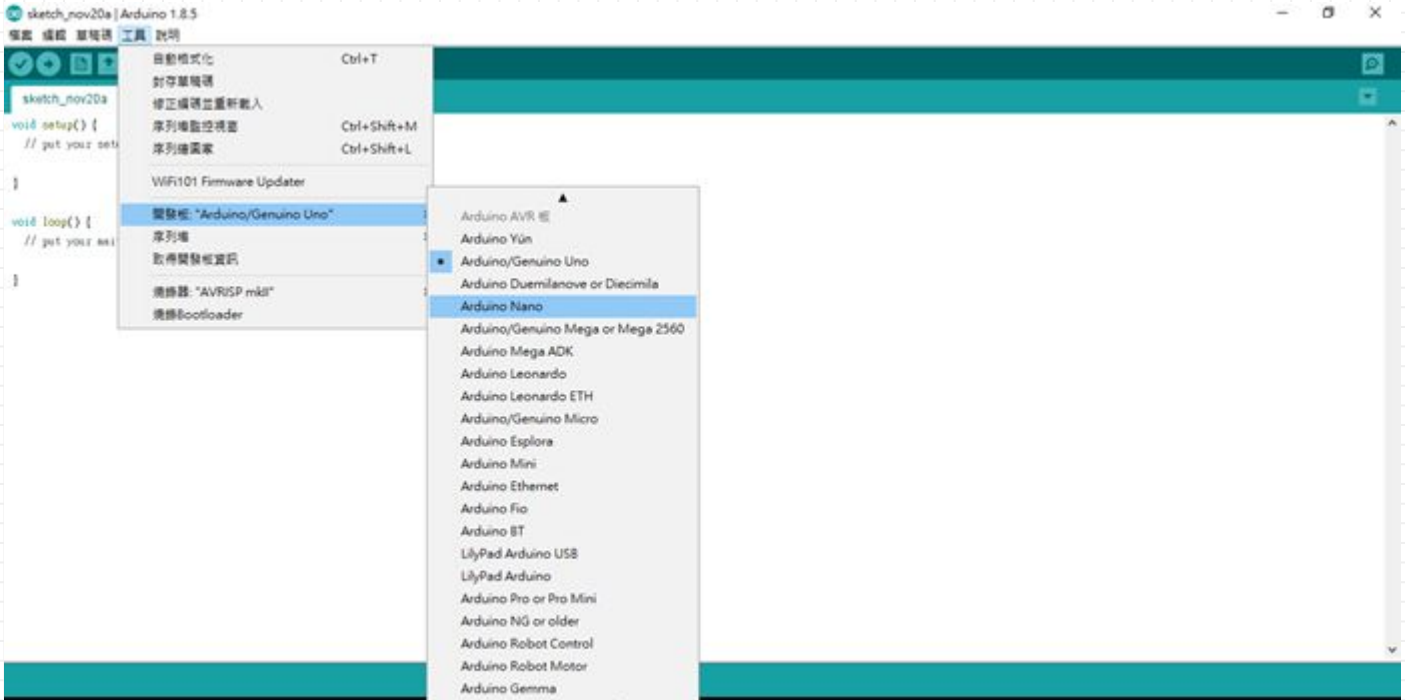
  瀏覽-選隨



設定控制板類型與USB通訊埠

- 選工具-開發板-選 Arduino Nano

  若使用UNO控制板者，請選Arduino/Genuino UNO)



設定控制板類型與USB通訊埠

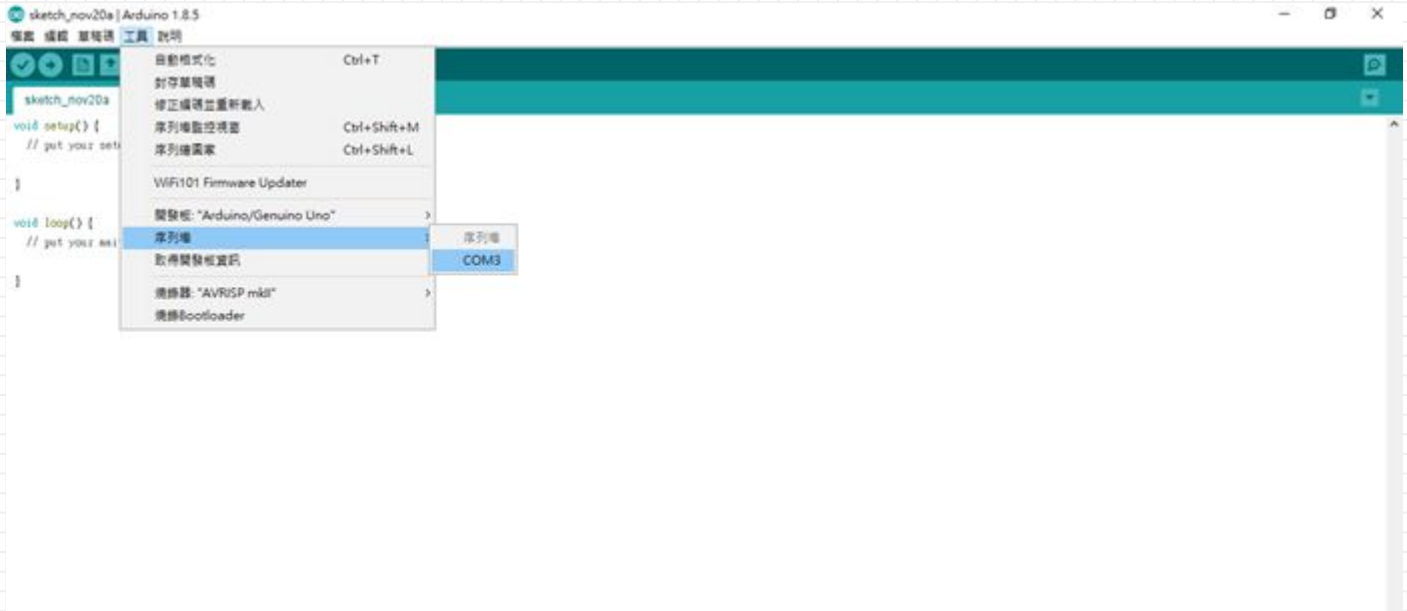
- 選工具-處理器-選 ATmega328P(Old Bootloader)

  若使用UNO控制板者，請選Arduino/Genuino UNO)

設定控制板類型與USB通訊埠

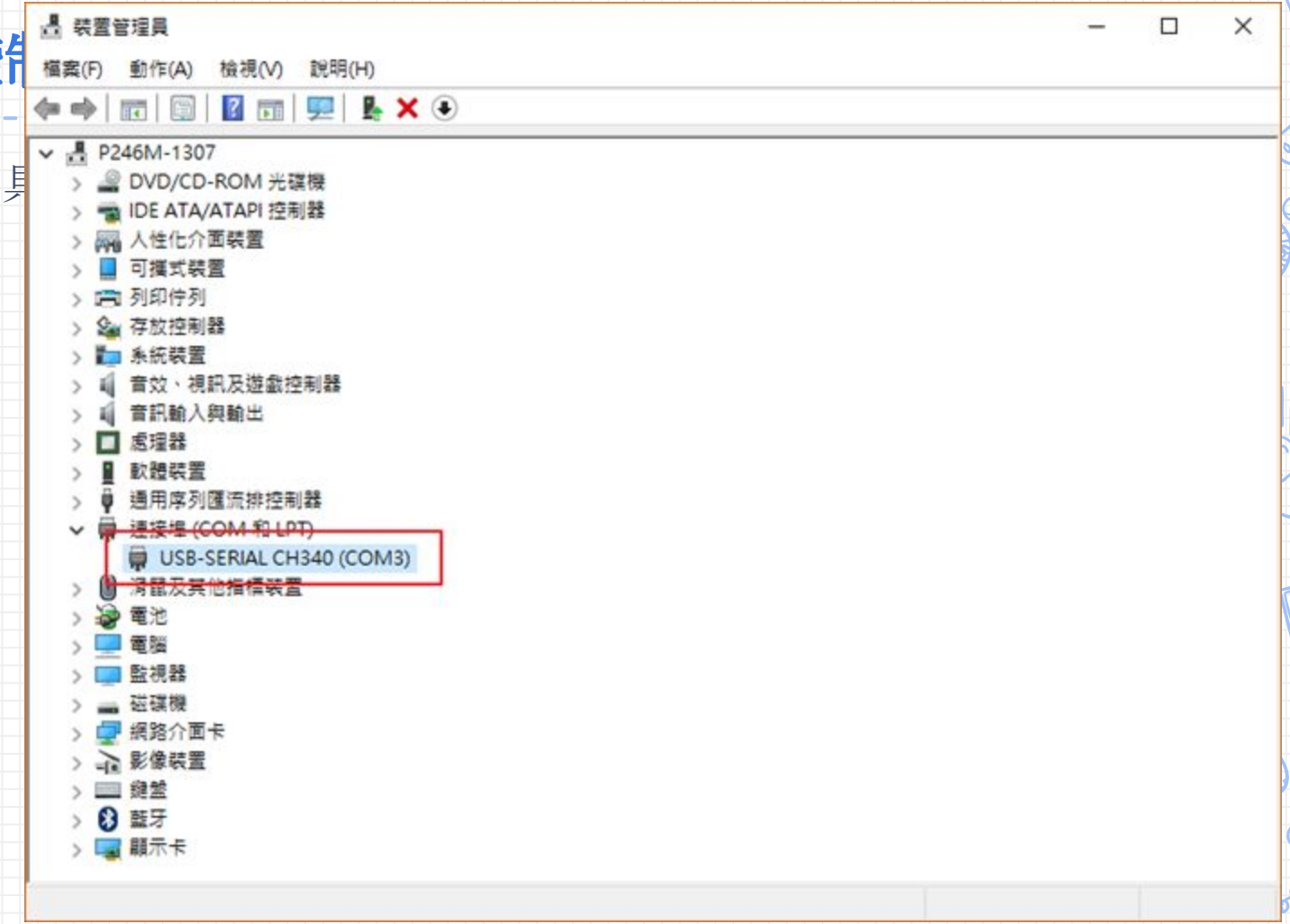
- 選工具-序列埠-選 COM3 (隨電腦環境而異)

  一般COM1、COM2會供電腦使用，通常會是COM3以上。實際使用的序列埠可由控制台-裝置管理員查詢。



設定控制

- 選工具


裝置管理員
檔案(F) 動作(A) 檢視(V) 說明(H)
P246M-1307
DVD/CD-ROM 光碟機
IDE ATA/ATAPI 控制器
人性化介面裝置
可攜式裝置
列印佇列
存放控制器
系統裝置
音效、視訊及遊戲控制器
音訊輸入與輸出
處理器
軟體裝置
通用序列匯流排控制器
連接埠 (COM 和 LPT)
USB-SERIAL CH340 (COM3)
滑鼠及其他指標裝置
電池
電腦
監視器
磁碟機
網路介面卡
影像裝置
鍵盤
藍牙
顯示卡

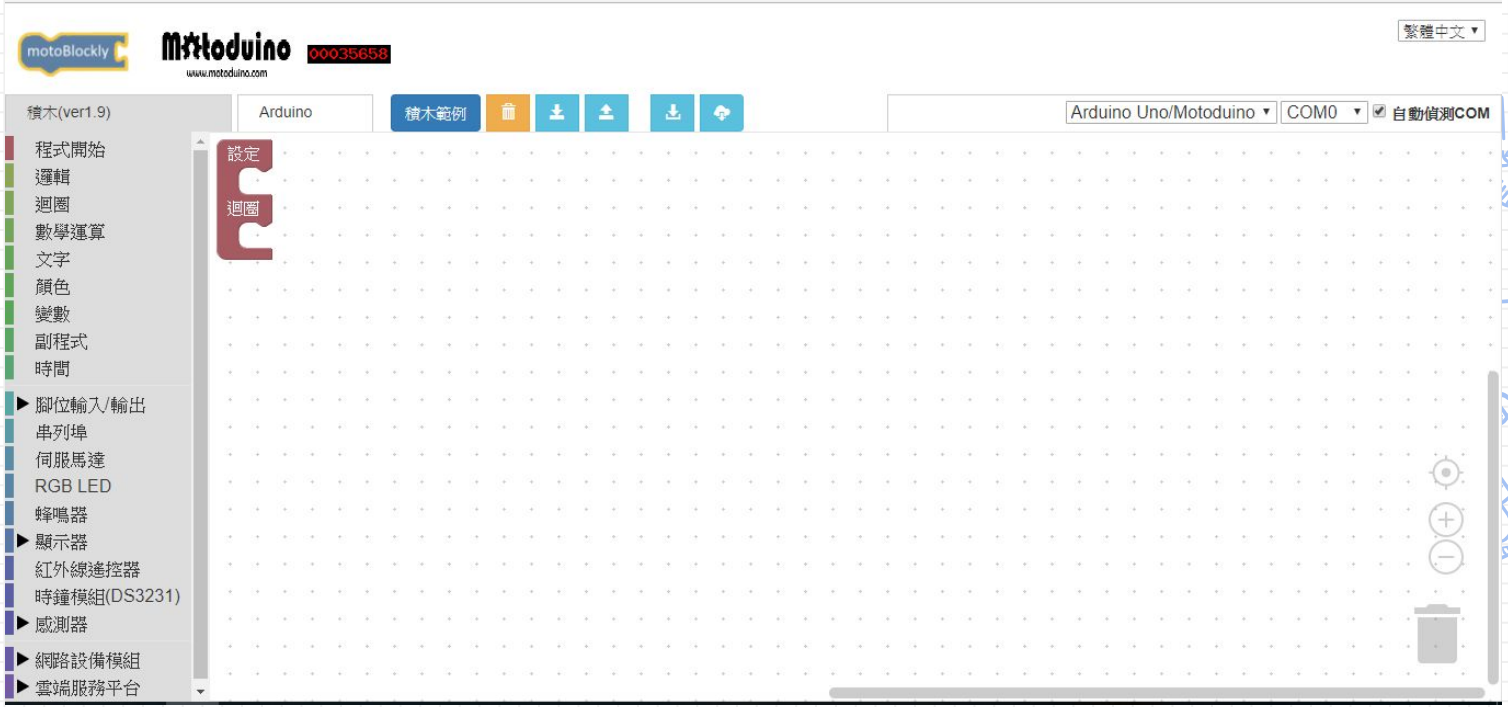
測試正常連線

- 選檔案-範例-01 Basics-Blink

  上傳
- 確認控制板內建 LED 正確閃爍

motoBlockly

- https://www.motoblockly.com
- Ver 1.9



motoBlockly

- https://www.motoblockly.com

內建 LED 閃爍

- blockly 熱身
- 01_LED13_blink.xml
- 可將積木存檔(.xml)

內建 LED 閃爍

加入序列埠

- 顯示訊息用
- 互動控制用
- 除錯用
- 02_LED13_blink_serial.xml

內建 LED 閃爍

加入序列埠

- 顯示訊息用
- 互動控制用
- 除錯用
- 03_LED13_blink_serialControl.xml

RGB LED 控制

- 數位輸出

RGB LED 控制

- 類比輸出

按鈕控制

- 數位輸入

Joy Stick 搖桿控制

- 類比輸入

聲音輸出控制

❖

光敏電阻控制

- 類比輸入

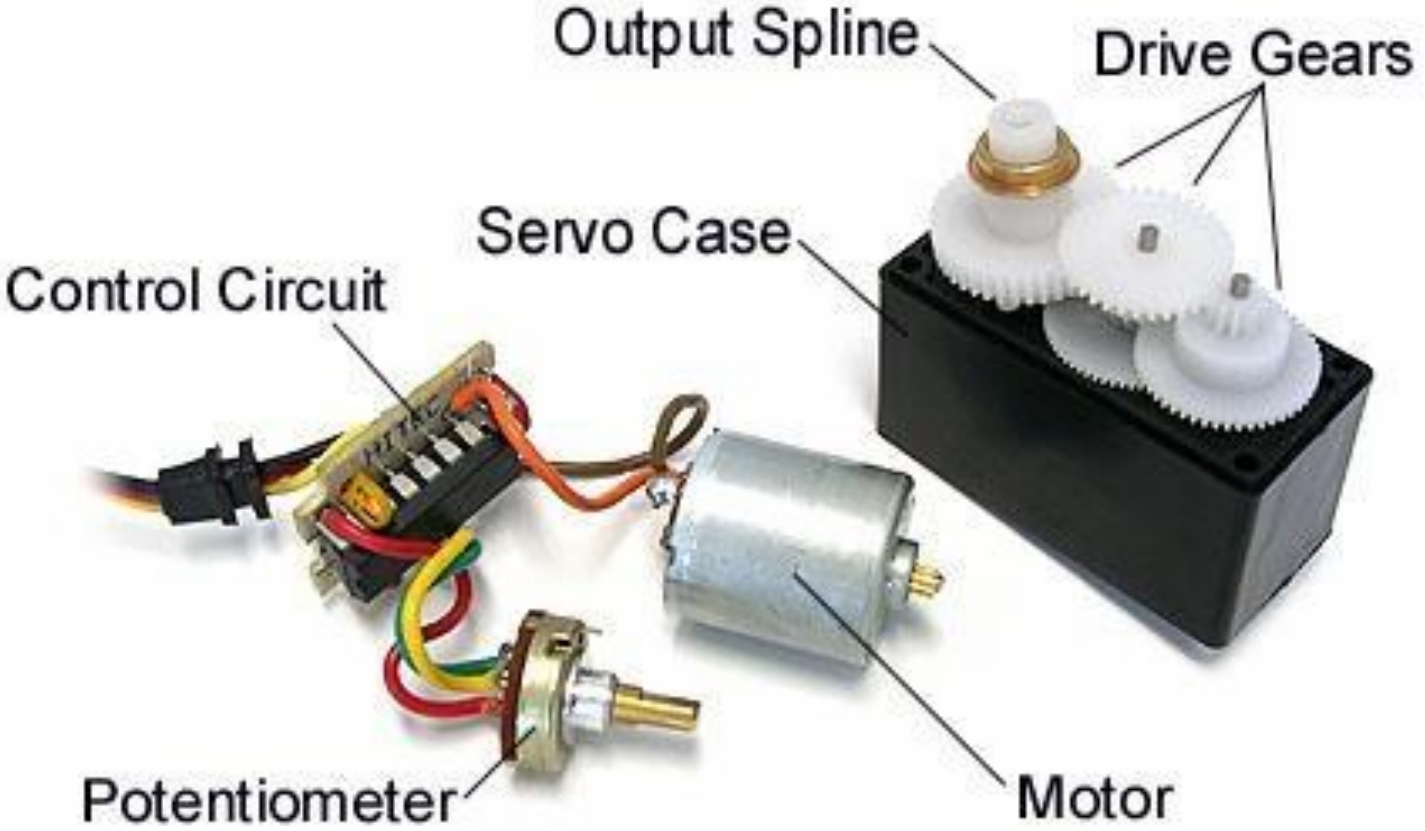
伺服馬達(SG90)

- SG90
- 運轉速度:0.1秒/60度。 旋轉角度:0度～180度。
- 使用前, 先「歸零(90度角)」。
- 橘黃色為訊號線, 紅色接正極, 棕色接地。工作電壓: 約5V。
- 伺服馬達轉到指定角度後, 會固定在定點;若強制轉開, 可能導致馬達齒輪組損壞。

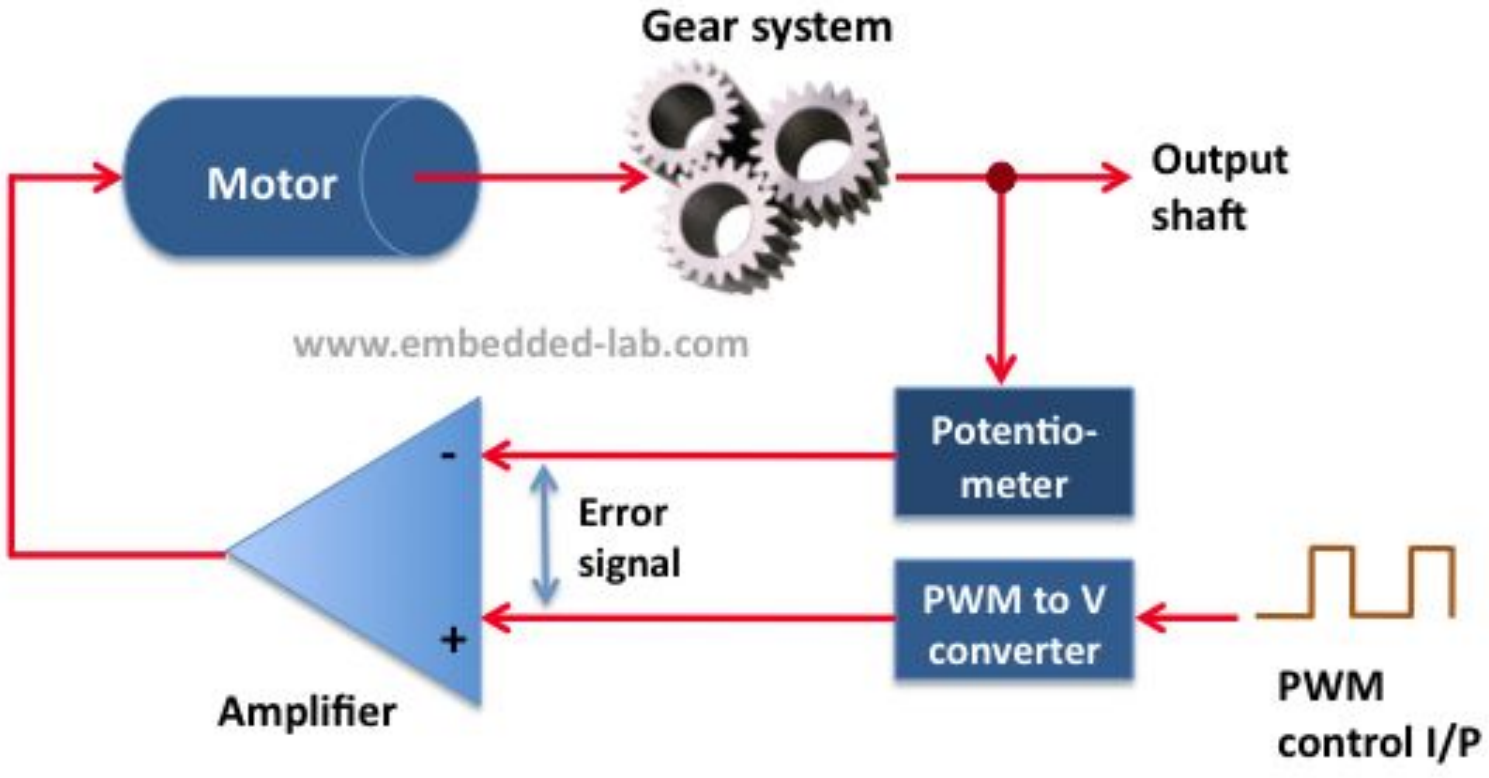
- 重量:9g
- 尺寸:23*12.2*29mm
- 工作電壓:4.8V
- 轉矩:1.8kg-cm, 當工作電壓為4.8V時
- 運轉速度:0.1秒/60度，當工作電壓為4.8V時
- 脈衝寬度範圍:500~2400μs
- 死頻帶寬度(dead band width):10μs

- 伺服馬達(servo motor)，因常用於遙控模型飛機，所以又常稱為RC伺服機(RC Servo，Radio Control Servo，Remote Control Servo)、伺服馬達舵機。
- 伺服馬達裡含有直流馬達、齒輪箱、軸柄、以及控制電路，我們可透過訊號控制軸柄的停止角度，大概都是0到180度，但不同廠牌型號會有不同的範圍;經由齒輪箱降速後，變成適當可用的轉速，並且提供更高的轉矩(扭轉力)。
- 另外還有能連續轉動的伺服馬達，有些出廠時便能連續轉動，有些則是玩家自己動手改造，這種馬達透過訊號可控制轉動的速度。

Servo parts

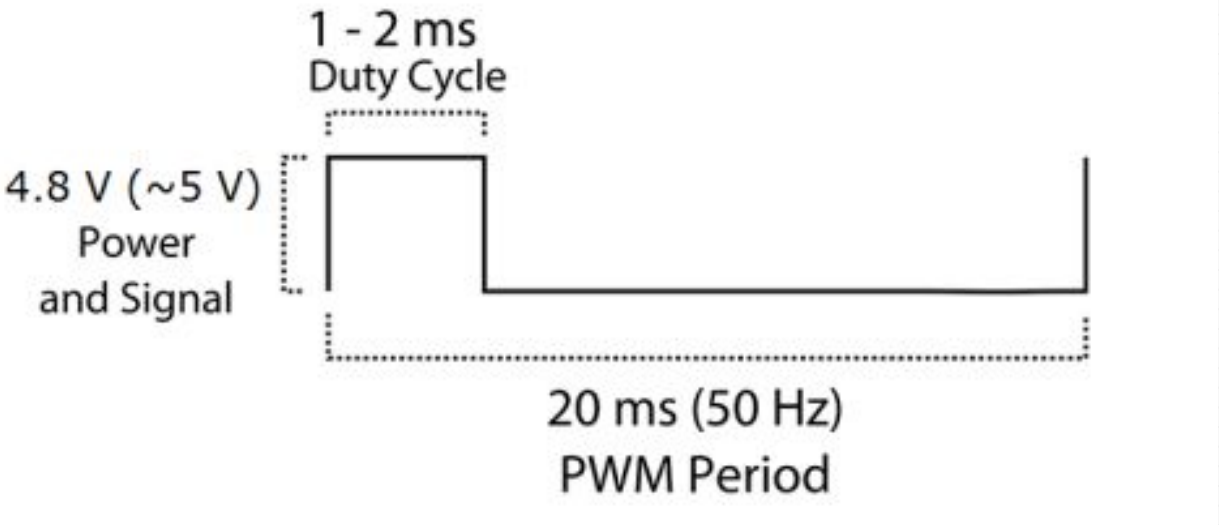

Output Spline
Drive Gears
Servo Case
Control Circuit
Potentiometer
Motor

Principle of a servomotor

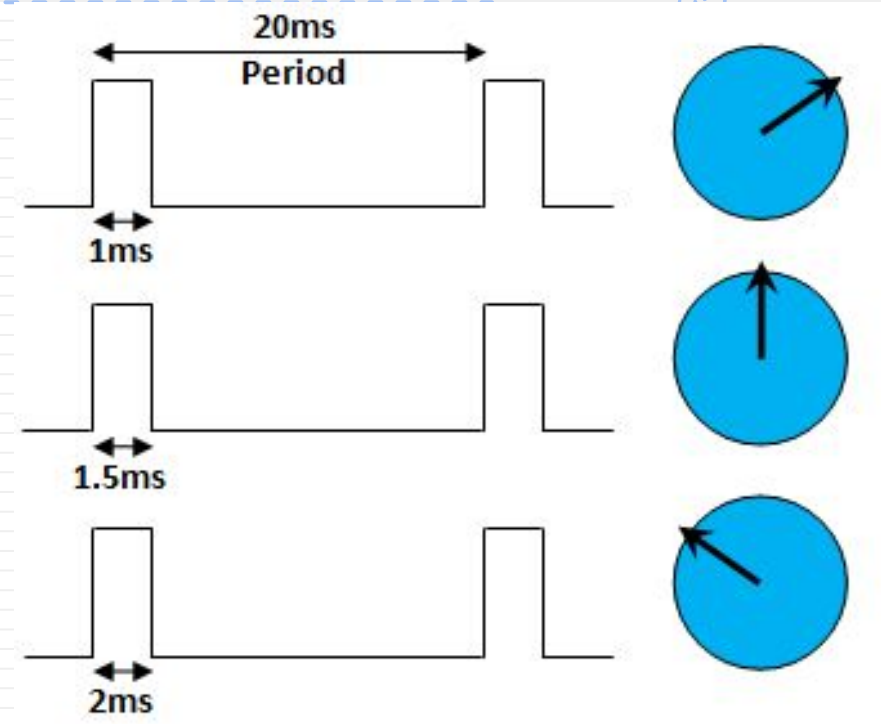


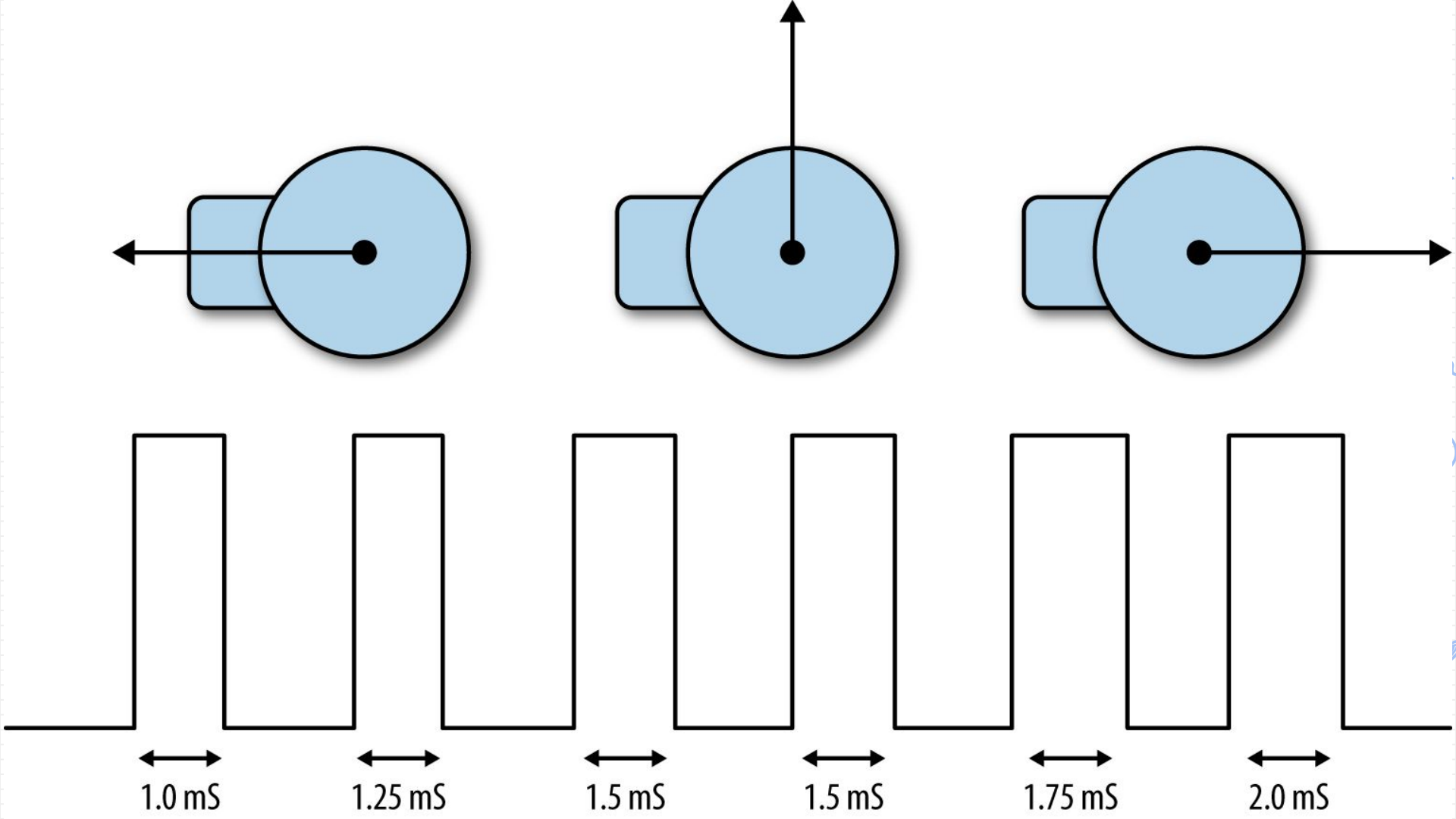
- 一般伺服馬達有三條線，電源（紅色）、接地（黑或棕色）、訊號線（白、黃、橘、藍，甚至是黑色）。透過訊號線傳送PWM脈波來控制軸柄的停止位置旋轉角度，這個訊號脈波必須每秒重複50次（也就是50Hz），而脈衝持續時間長短便代表了馬達該將軸柄轉到什麼位置，範圍從1.0ms到2.0ms（millisecond，毫秒，千分之一秒），若想置中則是1.5ms；也可將1.0ms當做角度0度，那麼1.5ms會是90度，2.0ms則是轉到底180度。注意，也有可能反過來。

PWM脈波



Pulse width (ms)	Angular position (° CCW)
0.7	0 (min)
1.1	45
1.5	90
1.9	135
2.3	180 (max)



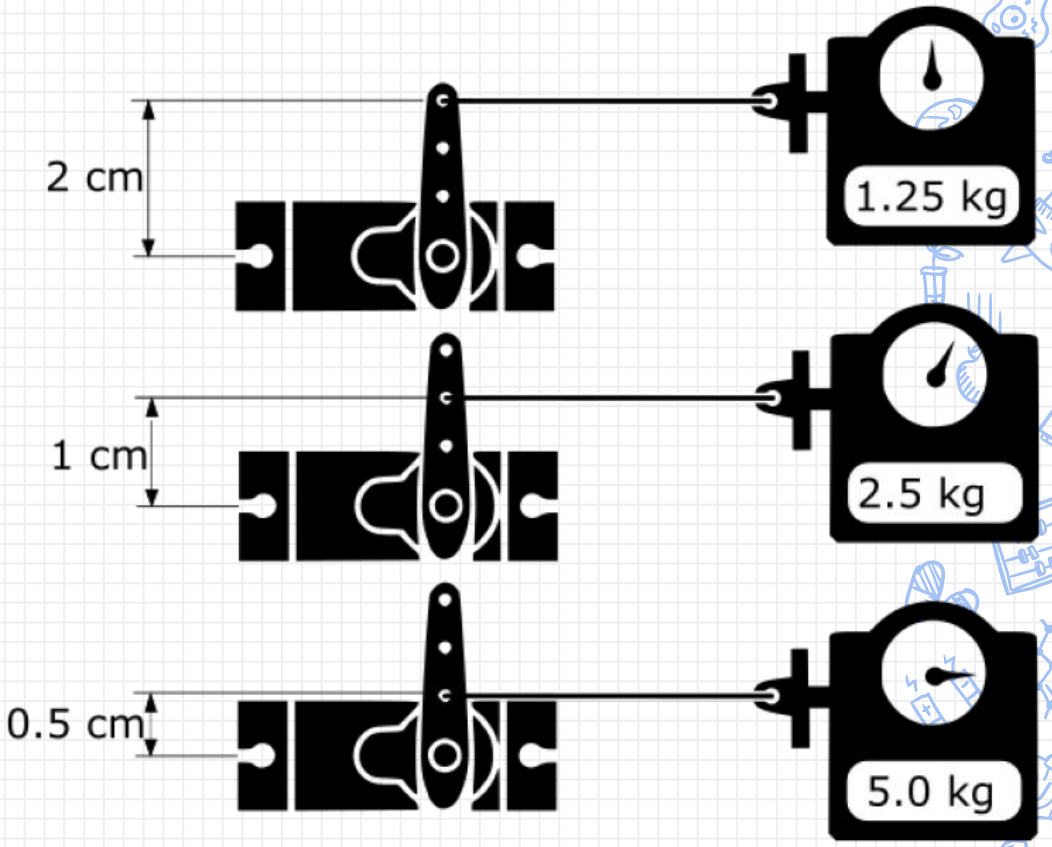

1.0 mS
1.25 mS
1.5 mS
1.5 mS
1.75 mS
2.0 mS

Torque(轉矩)

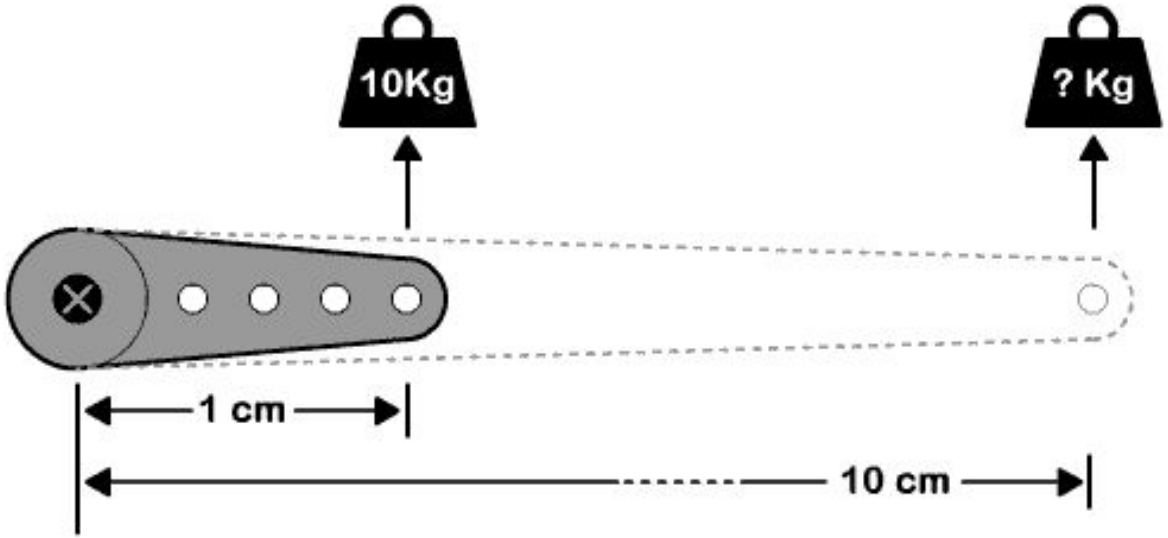
20 kg-cm or 270 oz-in mean

2.5kg/cm rated servo

2.5 kg-cm



10 kg-cm



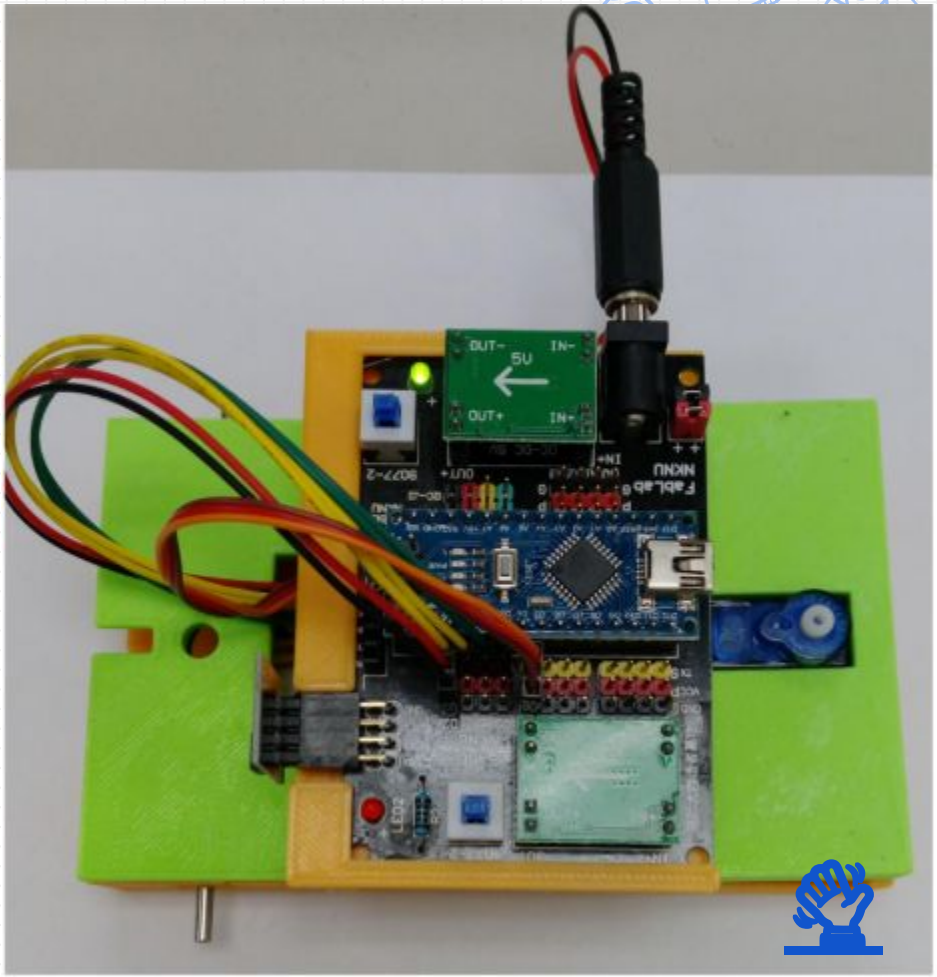
伺服馬達(SG90)

- SG90

  訊號橘色　　D6

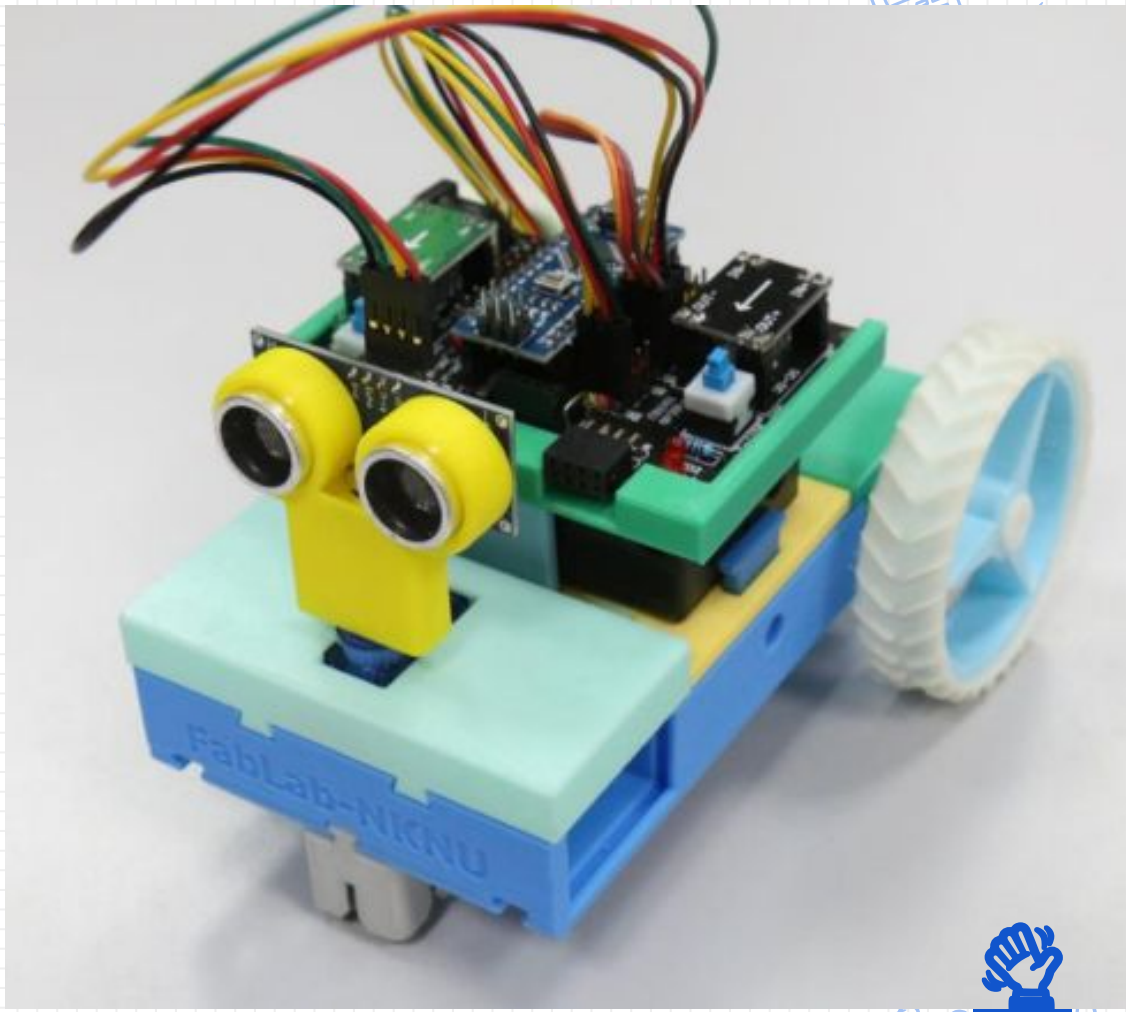
  Vcc 紅色　　5V 紅色針腳

  GND 黑色　　GND 黑色針腳



伺服馬達(SG90)定位

- 取下超音波模組
- 定位至 90度
- 接上超音波模組



伺服馬達(SG90)

- 90度定位
- 04_SG90_set90.xml

伺服馬達(SG90)

- 左右搖擺
- 05_SG90_Scan.xml

超音波

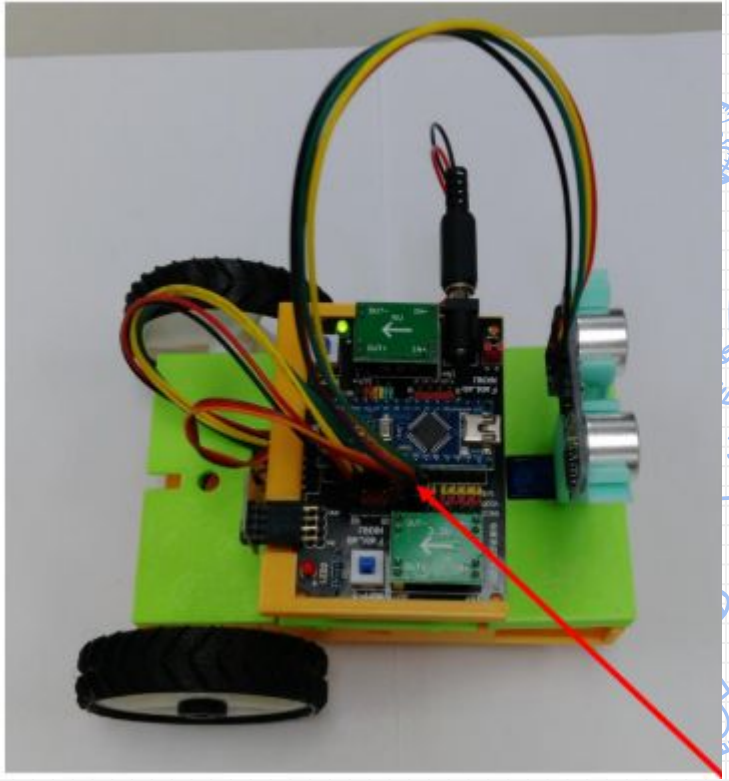
- HCSR04超音波

  Trig　　D7

  Echo　　D8

  Vcc 紅色　　5V 紅色針腳

  GND 黑色　　GND 黑色針腳



超音波

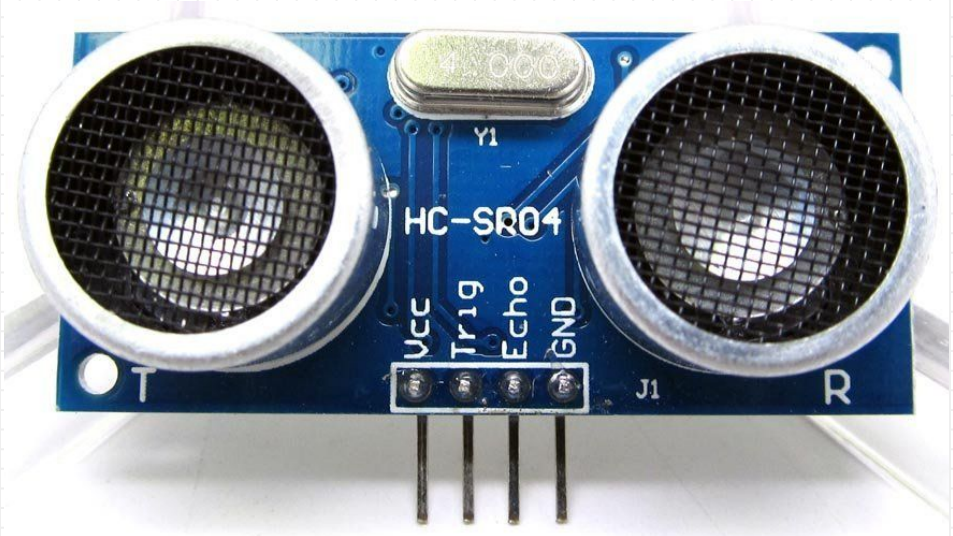
- 超音波是指超出人耳所能聽到的頻率 20KHz 的音波, 人類的聽覺一般只能聽到 20Hz~20KHz 左右的音域, 但某些動物如狗, 海豚或蝙蝠等, 可以聽到 20KHz 以上的超音波, 例如海豚利用超音波傳遞溝通訊息; 而蝙蝠則利用超音波在飛行中定位以避開障礙物
- 超音波於 19 世紀被發現, 並於第一次世界大戰首次應用於潛水艇之偵測, 如今在工業, 醫學, 與農業上都有廣泛之應用, 例如非破壞性檢測中的孔隙偵測, 孕婦姙娠掃描, 人體組織斷層掃描, 物件清洗, 以及蔬果品質檢測等等

超音波

- 在空氣中傳遞的超音波其頻率在 20K~200KHz 之間, 頻率越高, 衰減程度越大, 可傳播的距離越短。
- 一般常用的超音波模組, 其頻率通常是 38K, 40K, 或 42KHz 這三種。

HC-SR04

- HC-SR04 超音波模組使用的是 40KHz 的超音波，可探測距離為 2cm~400cm。
- 有 Vcc (5V)、Gnd、Trig、Echo 四隻接腳

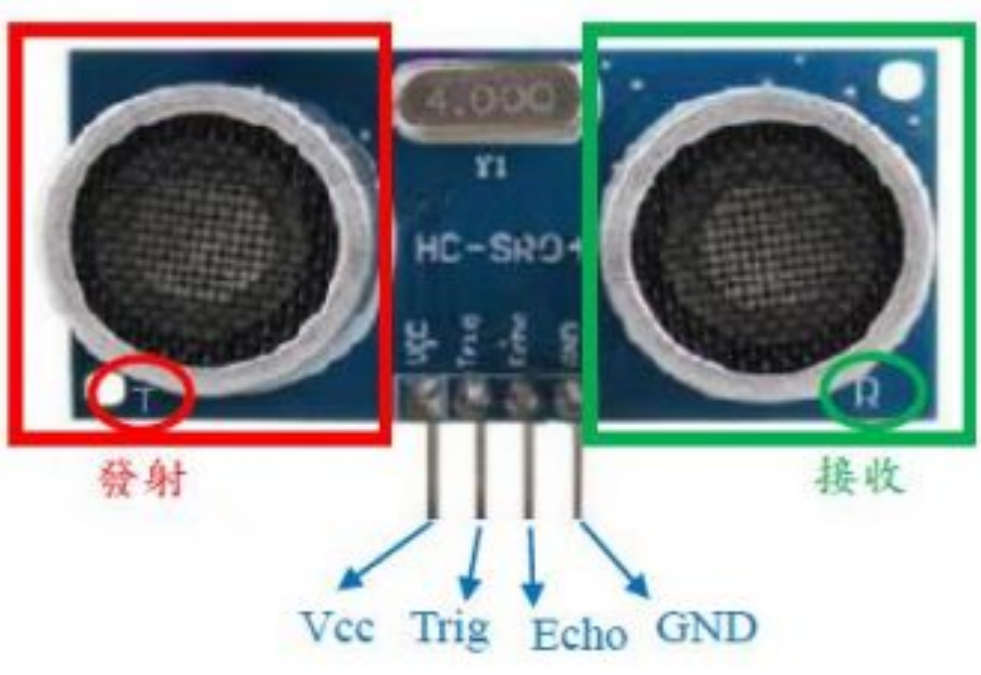


HC-SR04 規格

- 電源:DC5V/2mA
- 輸出電位(1/ 0):5V/ 0V
- 精度:3mm
- 距離範圍:2 ~ 450cm
- 有效的角度:<15度
- 觸發輸入信號:10uS TTL pulse
- ECHO輸出信號: Input TTL lever signal and the range in proportion
- 尺寸:45(L) x 20(W) x 15(H)mm

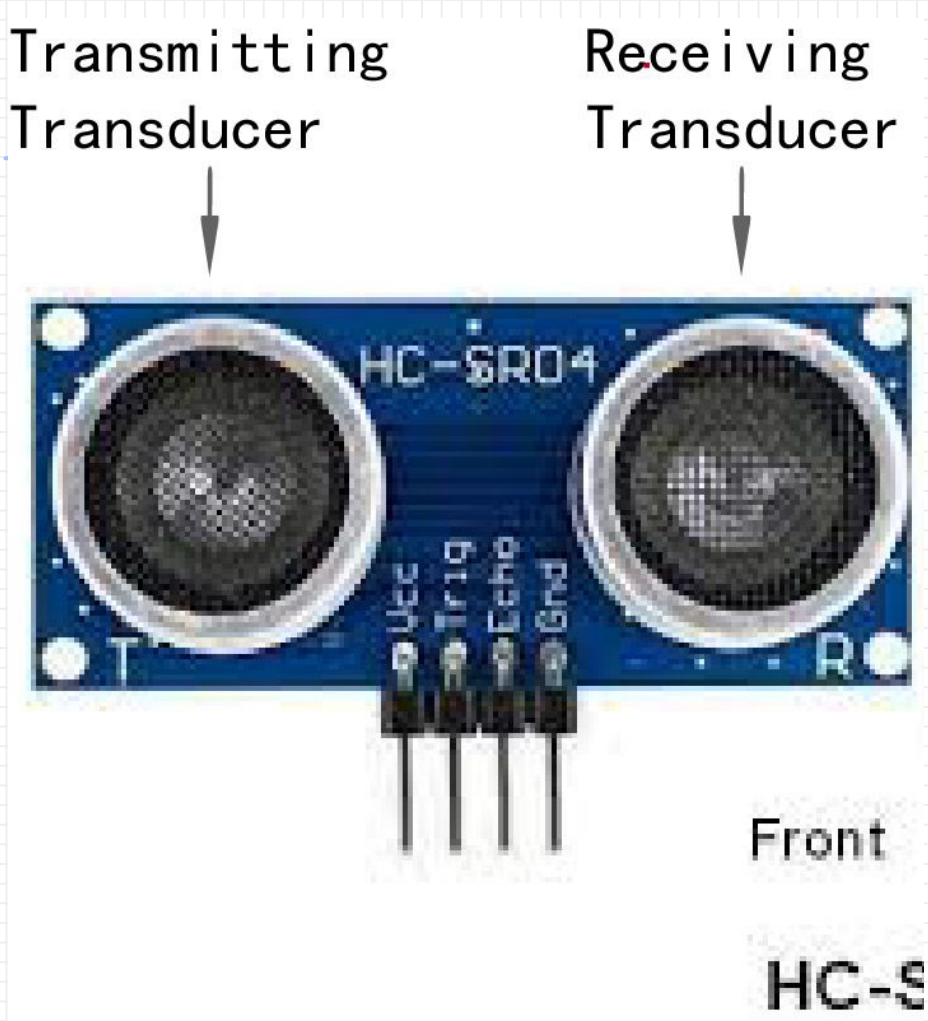
HC-SR04

- HC-SR04



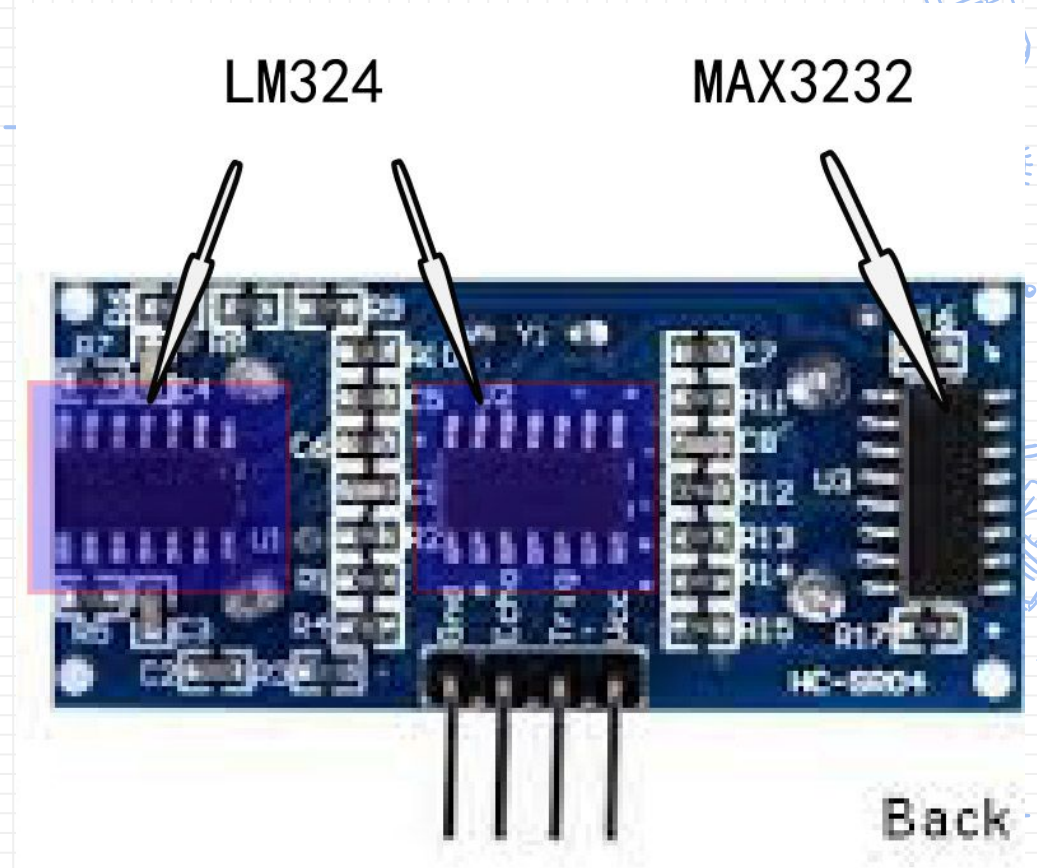
HC-SR04

- HC-SR04



HC-SR04

- HC-SR04



SR04

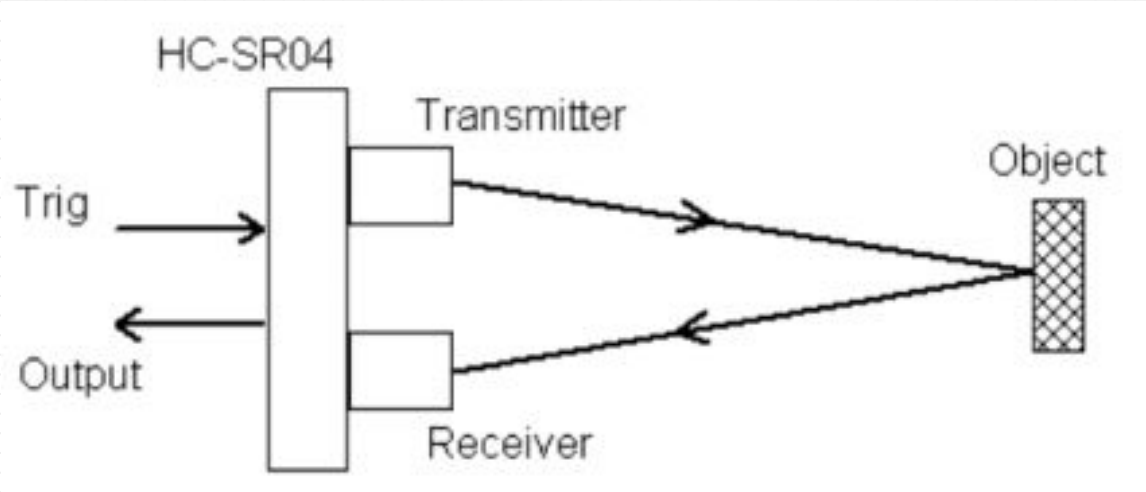
HC-SR04

- 測距

  06_HCSR04_DistanceMeter.xml

HC-SR04 模組測量距離原理

- 超音波模組測量距離的方法, 利用超音波碰到物體會反射的特性, 計算發射與接收的時間差, 計算出距離

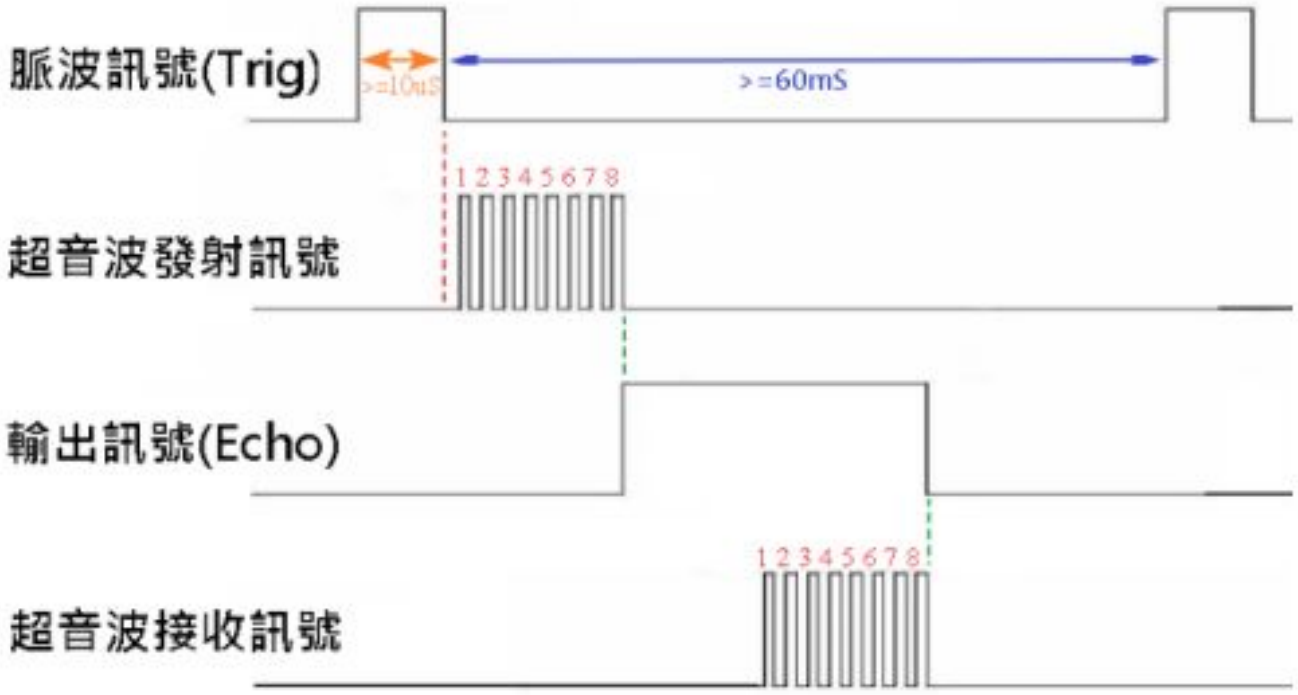


HC-SR04 模組工作原理

- 採用I/O觸發測距
- 給 Trig 一個 10 us 寬度的脈波, 啟動模組, 模組就會發射 8 個 40k Hz 的聲波, Echo 會由低電位變成高電位, 直到接收到反射回來的超音波, Echo 會由高電位變回低電位。
- 要特別注意的地方是被測物體最好大於 0.5 平方公尺, 而 Trig 時間間隔最好大於 60ms, 避免 Trig 與 Echo 互相干擾。

HC-SR04 模組工作原理

- 採用I/O觸發測距



測距應用

- 使用超音波量測距離必須知道音波的傳遞速度，這主要取決於大氣的密度，而溫度又是影響空氣密度的主要因素。音速與溫度的關係如下：

  音速=331.5 m/s + 0.6*攝氏溫度

  在常溫 20 度時，音速是 331.5+0.6*20=343.5 m/s，大約是 344 m/s。

測距應用

- 在常溫 20 度時, 音速是 344 m/s。
- 344 m/s = _____ cm/us(公分/微秒)。
- 聲波前進 1 cm 需時_____ us。

測距應用

- 超音波測距的原理是利用一個超音波發射器與一個接收器組成的模組來量測音波從發射到收到反射波的時間，乘以音速即可得到音波往返的距離，除以 2 即得與反射物體間的距離。

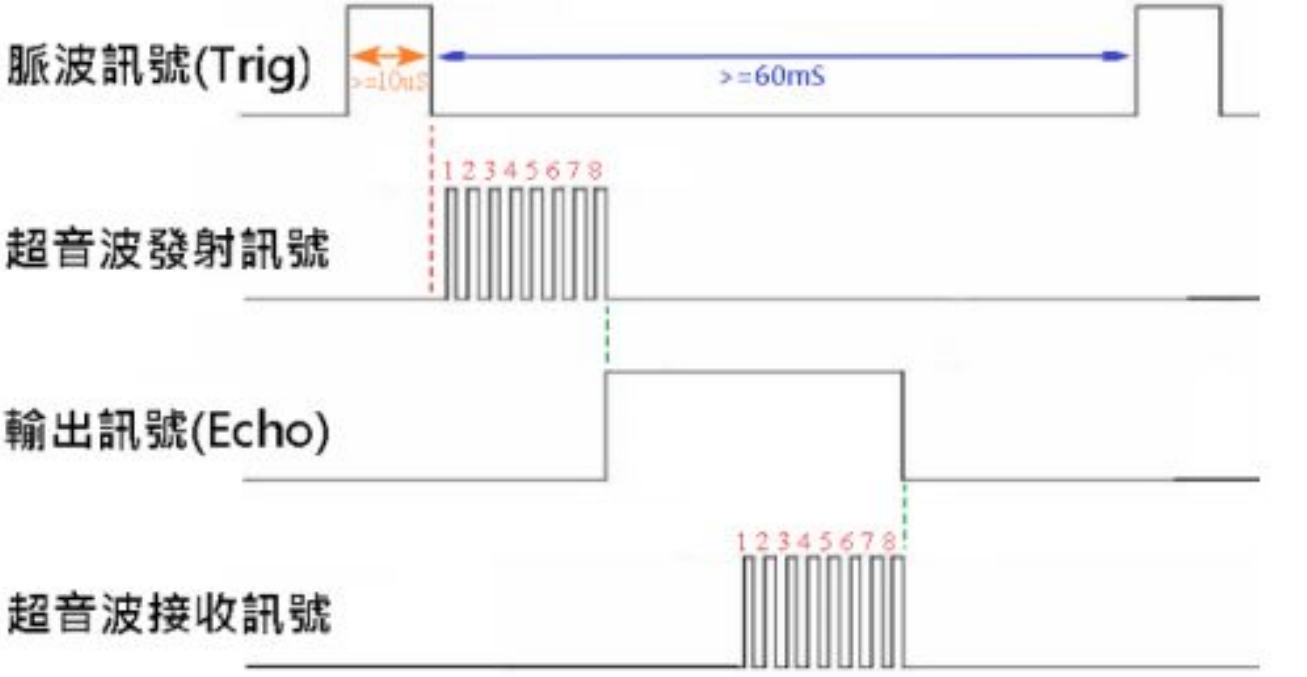
測距應用

- 在常溫 20 度下, 在常溫 20 度時, 音速是 344 m/s。
- 344 m/s = $344*10^{-4}$ = 0.0344 cm/us。
- 音波前進 1 公分約需時 29 微秒。計算如下 :

  0.0344 cm/ 1 us = 1 cm / (1/0.0344) us = 1 cm / 29 us

測距應用

- HC-SR04 的 Echo 腳輸出的高位準脈衝, 其時長即是超音波往返所花的時間(duration)(us)。



測距應用

- HC-SR04 的 Echo 腳輸出的高位準脈衝, 其時長即是超音波往返所花的時間(duration)(us)。
- 距離 = 速率 * 時間

  distance = (duration/2) * 0.0344 cm/us

  = (duration/2) * 1/29 cm = (duration/2)/29 cm

  = (duration/58 cm)
- 1 cm / 29 us = distance / (duration/2)

  distance = (duration/2)/29 = duration/58 cm

測距應用

- HC-SR04 的 Echo 腳輸出的高位準脈衝, 其時長即是超音波往返所花的時間, 用 Arduino 的 pulseIn() 函數量取 Echo 的脈衝時長(單位:微秒), 再除以 58, 即可得到與反射超音波的障礙物之間的距離是幾公分了。
- duration = pulseIn()

  distance = (duration/2) * 0.0344 cm

  = duration/58 cm

  = (duration/2)/29 cm

語法

- pinMode(trigPin, OUTPUT);

  pinMode(echoPin, INPUT);
- digitalWrite(trigPin, HIGH);

  delayMicroseconds(10); //sustain at least 10us HIGH pulse

  digitalWrite(trigPin, LOW);
- pulseIn(echoPin, HIGH);

pulseIn()

- pulseIn(pin, value)

  pulseIn(pin, value, timeout)

- 此函數有 2 或 3 個參數:

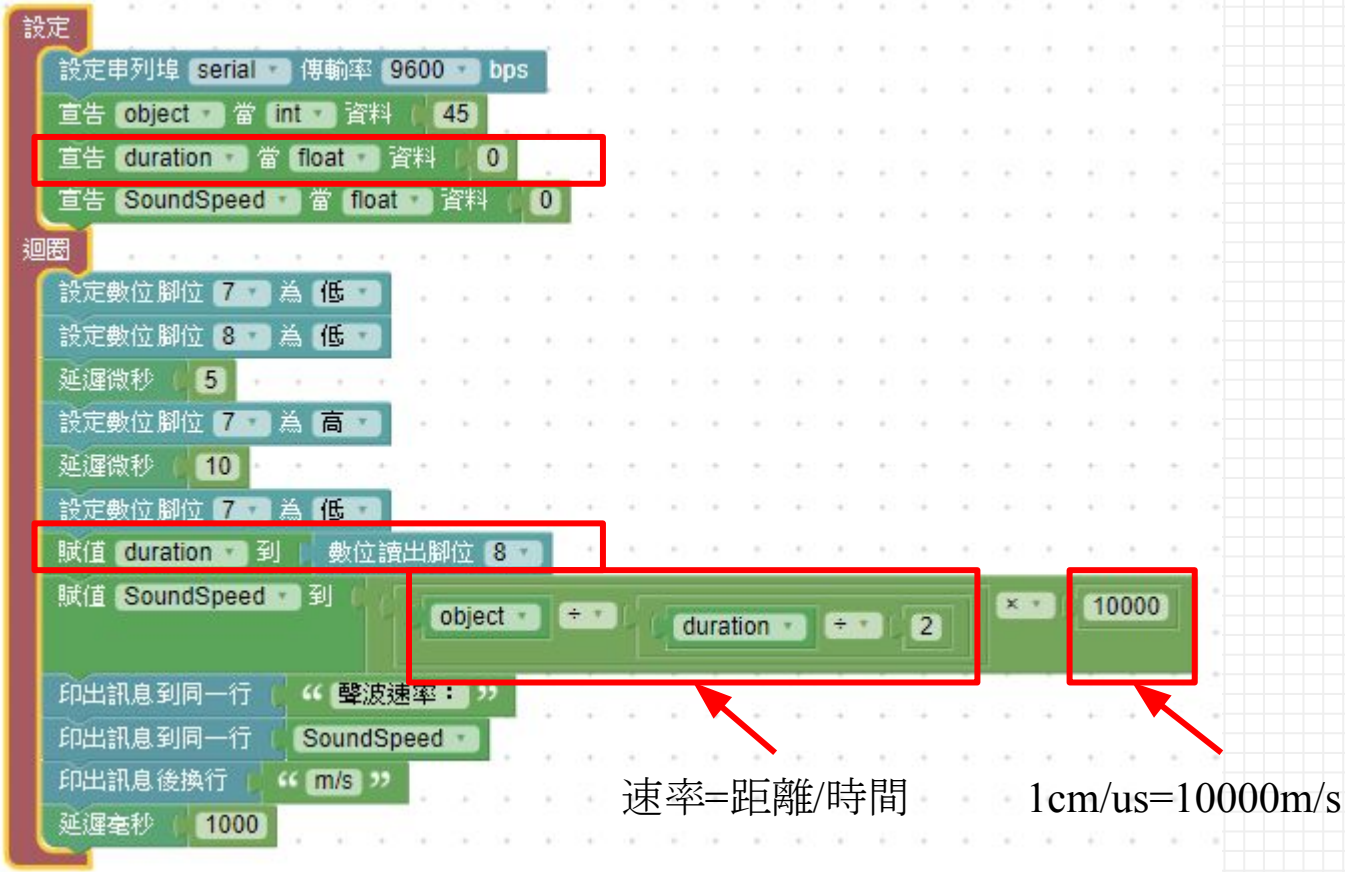
  第一參數 pin 是 Arduino 接 HC-SR04 的 Echo 腳之 DIO 數位腳位編號。第二參數 value 是要偵測的位準 (LOW/HIGH), 對於 HC-SR04 而言要用 HIGH。第三參數 timeout 是等候脈衝出現的延遲時間 (單位是微秒), 預設是 1 秒 (即 1000000us), 如果在 timeout 時間內未偵測到脈衝, 則回傳 0。

HC-SR04

- 測聲波速率
- 無現成積木可用，自造。以數位輸入積木讀入聲波行走時間，拉好積木後，再於 Arduino IDE 修改程式碼。

  07_HCSR04_SoundSpeed.xml

HC-SR04



速率=距離/時間

1cm/us=10000m/s

HC-SR04

```
10 {
11   Serial.begin(9600);
12   pinMode(7, OUTPUT);
13 // 建立完成 Blockly 後，增修程式碼
14 //  pinMode(8, OUTPUT);
15 // 建立完成 Blockly 後，增修程式碼
16   pinMode(8, INPUT);
17     object = 45;
```

HC-SR04

```
30      delayMicroseconds(10);
31      digitalWrite(7,LOW);
32  // 建立完成 Blockly 後，增修程式碼
33      duration = pulseIn(8,HIGH);
34  // 建立完成 Blockly 後，增修程式碼
35      SoundSpeed = (object / (duration / 2)) * 10000;
36      Serial.print("聲波速率：");
37      Serial.print(SoundSpeed);
```

HC-SR04

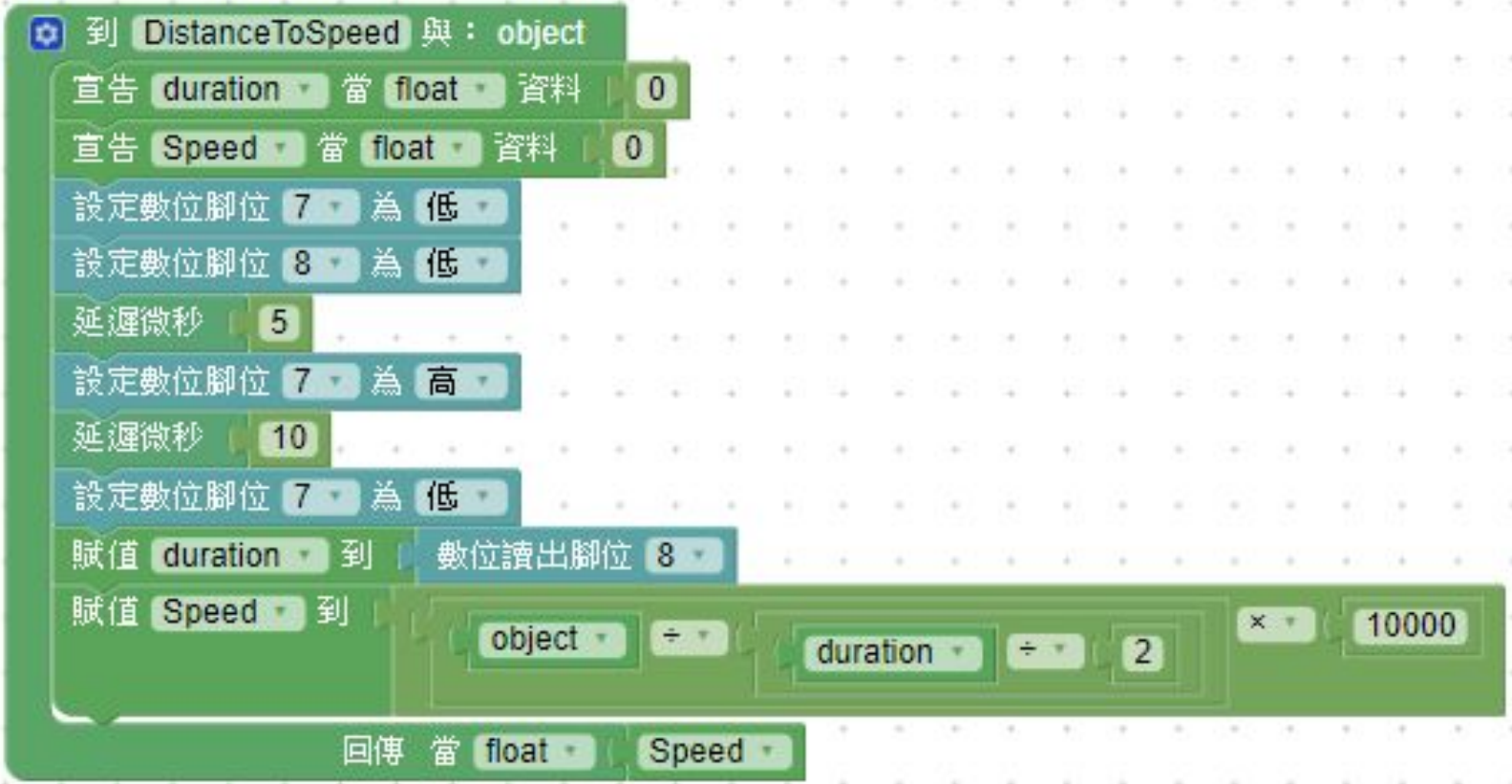
- 測聲波速率, 自訂函式
- 拉好積木後, 須於 Arduino IDE 改程式碼

  08_HCSR04_SoundSpeed_Function.xml

HC-SR04

- 
設定
設定串列埠 serial 傳輸率 9600 bps
宣告 SoundSpeed 當 float 資料 0
迴圈
賦值 SoundSpeed 到 呼叫 DistanceToSpeed 與：
object 45
印出訊息到同一行 “聲波速率：”
印出訊息到同一行 SoundSpeed
印出訊息後換行 “m/s”
延遲毫秒 1000

HC-SR04


到 DistanceToSpeed 與： object
宣告 duration 當 float 資料 0
宣告 Speed 當 float 資料 0
設定數位腳位 7 為 低
設定數位腳位 8 為 低
延遲微秒 5
設定數位腳位 7 為 高
延遲微秒 10
設定數位腳位 7 為 低
賦值 duration 到 數位讀出腳位 8
賦值 Speed 到
object ÷ duration ÷ 2 × 10000
回傳 當 float Speed

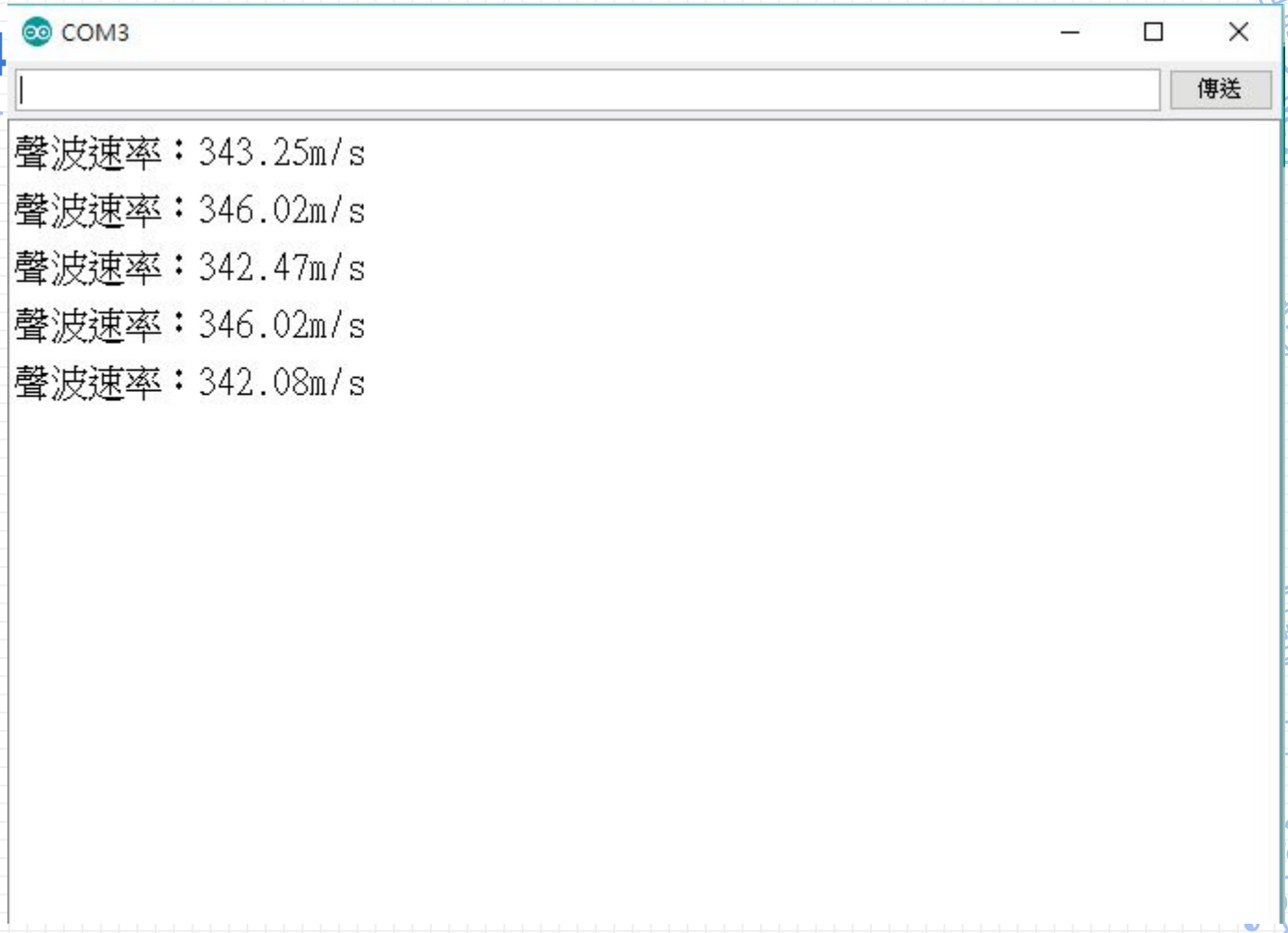
HC-SR04

```
16    delayMicroseconds(10);
17    digitalWrite(7,LOW);
18  // 建立完成 Blockly 後，增修程式碼
19    duration = pulseIn(8,HIGH);
20  // 建立完成 Blockly 後，增修程式碼
21    Speed = (object / (duration / 2)) * 10000;
22    return Speed;
23  }
```

HC-SR04

```
27    Serial.begin(9600);
28      SoundSpeed = 0;
29
30    pinMode(7, OUTPUT);
31  // 建立完成 Blockly 後，增修程式碼
32  //   pinMode(8, OUTPUT);
33  // 建立完成 Blockly 後，增修程式碼
34    pinMode(8, INPUT);
35  }
36
```

HC-SR04

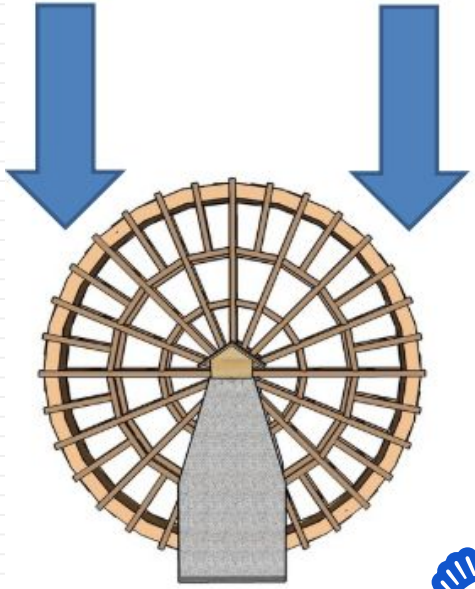

COM3
傳送
聲波速率：343.25m/s
聲波速率：346.02m/s
聲波速率：342.47m/s
聲波速率：346.02m/s
聲波速率：342.08m/s

直流減速馬達 N20

- N20, 採購時需指定電壓與轉速, 例如:6V、300rpm。
- N20兩端電位相等時, 不會轉動。電位差距越大, 轉速越快。

直流減速馬達N20

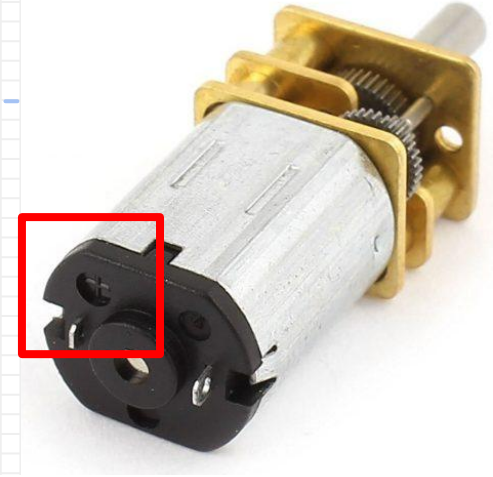
- 利用接線兩端電位差控制轉速與轉動方向
- 兩端電位差距越大，轉速越快
- 兩端電位相等，不轉動
- 將高低電位供電方向相反，轉動方向相反

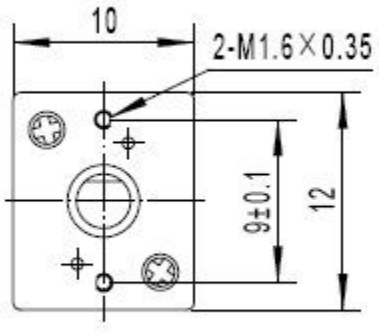


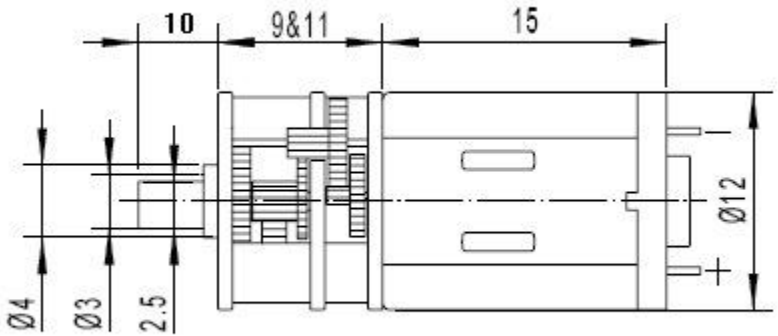
直流減速馬達N20

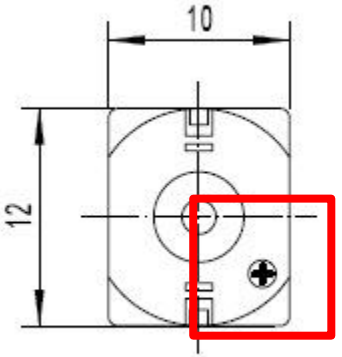
型號:JGA12-N20				參數表								
型號	電壓		空載		最大效率點				堵轉		減速器	
	Voltage		Noload		At Max.Effciency				Stall		Speedreducer	
	使用範圍	額定值	轉速	電流	轉速	電流	扭矩	功率	扭矩	電流	減速比	長度
Model	Using	The	Speed	Current	Speed	Current	Torque	Power	Torque	Current	Redction	(L)Size
	Rangge	Rated	r/min	ma	r/min	ma	kg.cm	W	kg.cm	A	Ratio	mm
JGA12-N20-30	3-12V	6V	500	40	375	160	0.081	0.35	0.4	0.55	30	10
JGA12-N20-50	3-12V	6V	300	40	225	155	0.14	0.35	0.7	0.55	50	10
JGA12-N20-100	3-12V	6V	150	40	120	155	0.35	0.35	1.75	0.55	100	10
JGA12-N20-298	3-12V	6V	50	40	40	140	1.1	0.35	5.5	0.55	298	10

直流減速馬達N20









直流減速馬達N20

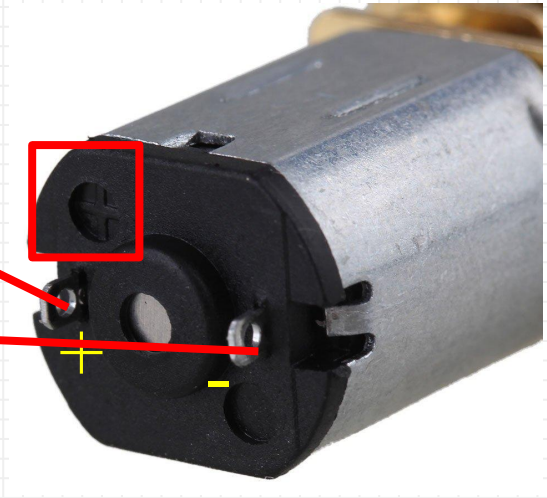
- 自焊 USB 電源接頭
- 轉向測試

直流減速馬達N20

- 接線

USB(-)　USB(+)

USB(+)　USB(-)



後退　前進

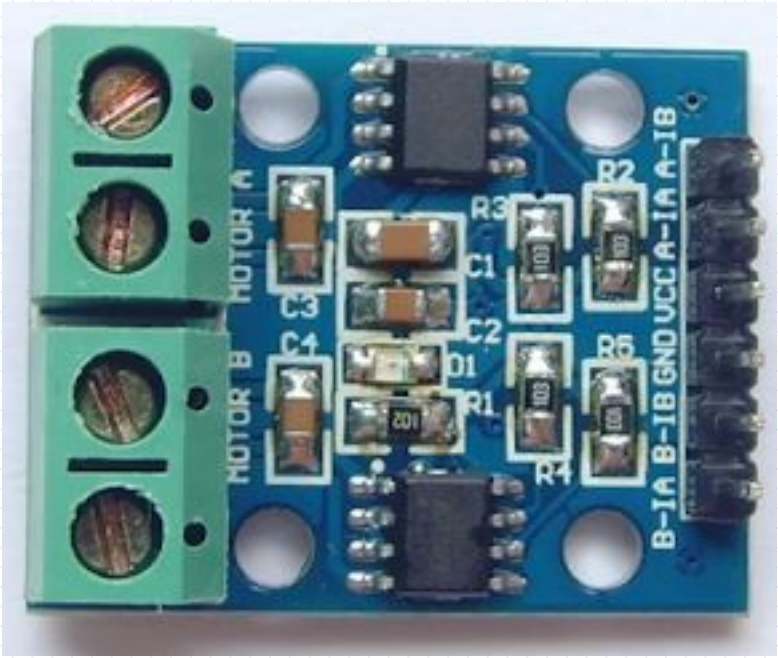
直流減速馬達N20

- 轉向測試

	左輪(+)	左輪(-)	右輪(+)	右輪(-)	運動
電源	-	+	+	-	前進
電源	+	-	-	+	後退
電源	+	+	+	+	停止
電源	-	-	-	-	停止

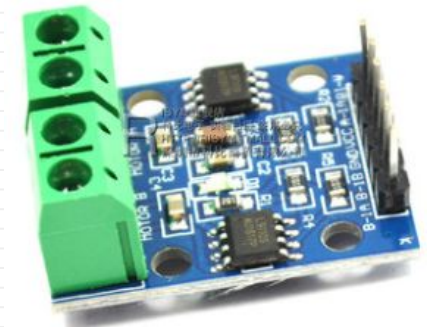
L9110S

- L9110S 直流馬達驅動模塊



L9110S

- 雙L9110S芯片的電機驅動
- 模塊供電電壓:2.5-12V
- 適合的電機範圍:電機工作電壓2.5v-12V之間, 最大工作電流0.8A, 目前市面上的智能小車電壓和電流都在此範圍內。
- 可以同時驅動2個直流電機, 或者1個4線2相式步進電機。



L9110S

- 控制方法:

① 2 個馬達都以 PWM 腳位(3~,5~,6~,9~,10~,11)控制。

② 一組控制方向(數位)、一組控制轉速(類比)

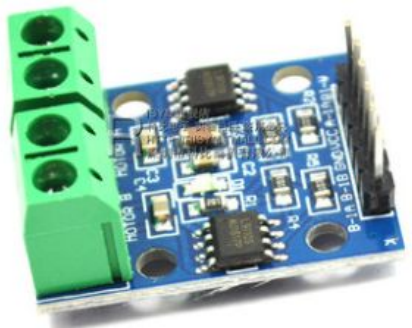
A-IA、B-IA:控制方向, 數位輸出。

A-IB、B-IB:控制轉速, PWM類比輸出。

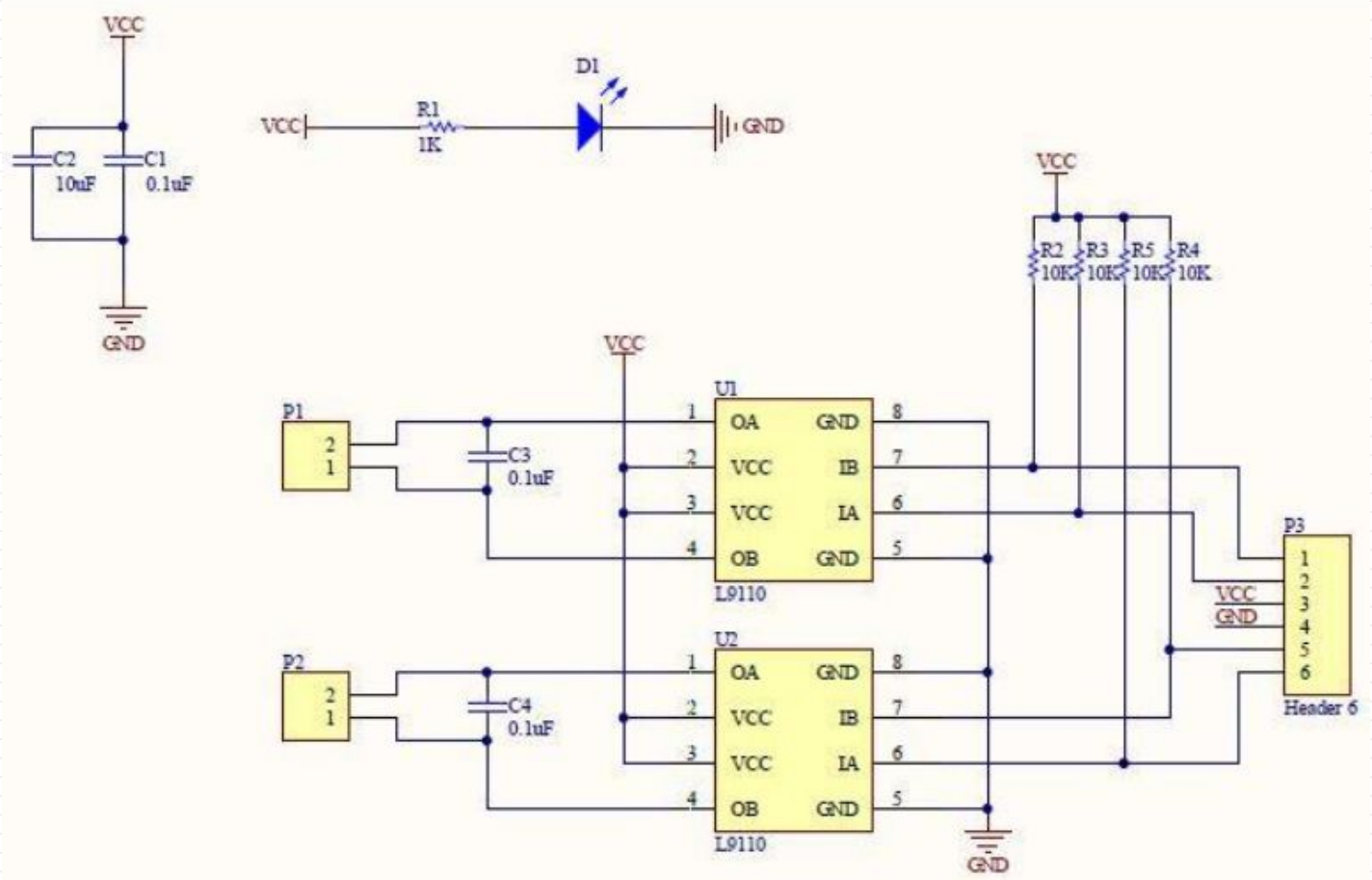
或

A-IA、B-IA:控制方向, 數位輸出。

A-IB、B-IB:控制轉速, PWM類比輸出。

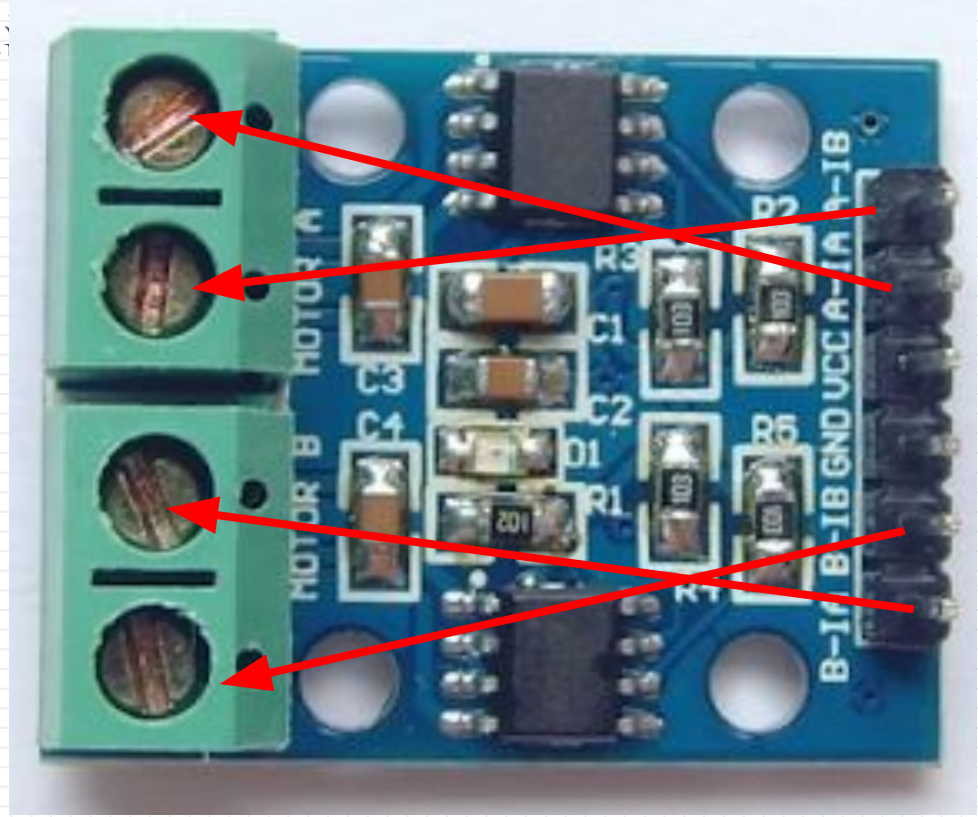


L9110S 電路圖


VCC
C2
10uF
C1
0.1uF
GND
D1
VCC
R1
1K
GND
VCC
R2
10K
R3
10K
R5
10K
R4
10K
VCC
P1
2
1
C3
0.1uF
U1
1
OA
GND
8
2
VCC
IB
7
3
VCC
IA
6
4
OB
GND
5
L9110
P2
2
1
C4
0.1uF
U2
1
OA
GND
8
2
VCC
IB
7
3
VCC
IA
6
4
OB
GND
5
L9110
GND
P3
1
2
3
4
5
6
VCC
GND
Header 6

L9110S

- L9110S 直



馬達

- 接線

	D3~	D2	D4	D5~	
	AIA	AIB	BIB	BIA	
	左輪(+)	左輪(-)	右輪(+)	右輪(-)	運動
電源	-	+	+	-	前進
電源	+	-	-	+	後退
電源	+	+	+	+	停止
電源	-	-	-	-	停止

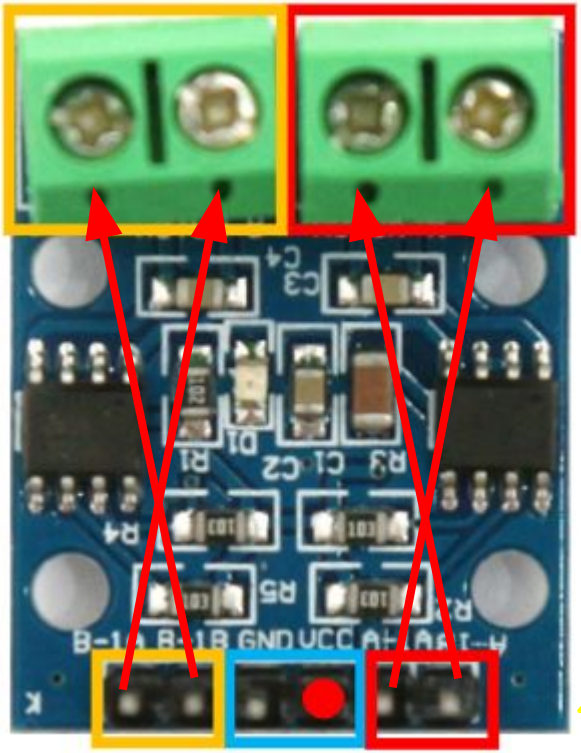
馬達控制板

右輪(+) 右輪(-) 左輪(-) 左輪(+)

- 左輪
  - AIB↔D2↔左輪(-)
  - AIA↔D3~↔左輪(+)
- 右輪
  - BIB↔D4↔右輪(+)
  - BIA↔D5~↔右輪(-)

馬達B

馬達A



馬達B訊號(4,5~)

馬達A訊號(2,3~)

D5 D4 電源 D3 D2

單輪控制

- 左輪測試
- 前進 10 秒,

  停 2 秒,

  後退 10 秒

  Motor_L_ForwardBack.xml

單輪控制

- 右輪測試
- 前進 10 秒,

  停 2 秒,

  後退 10 秒

  Motor_R_ForwardBack.xml

單輪控制(自訂函式)

- 左輪測試
- 自訂函式, 可控制轉動速率

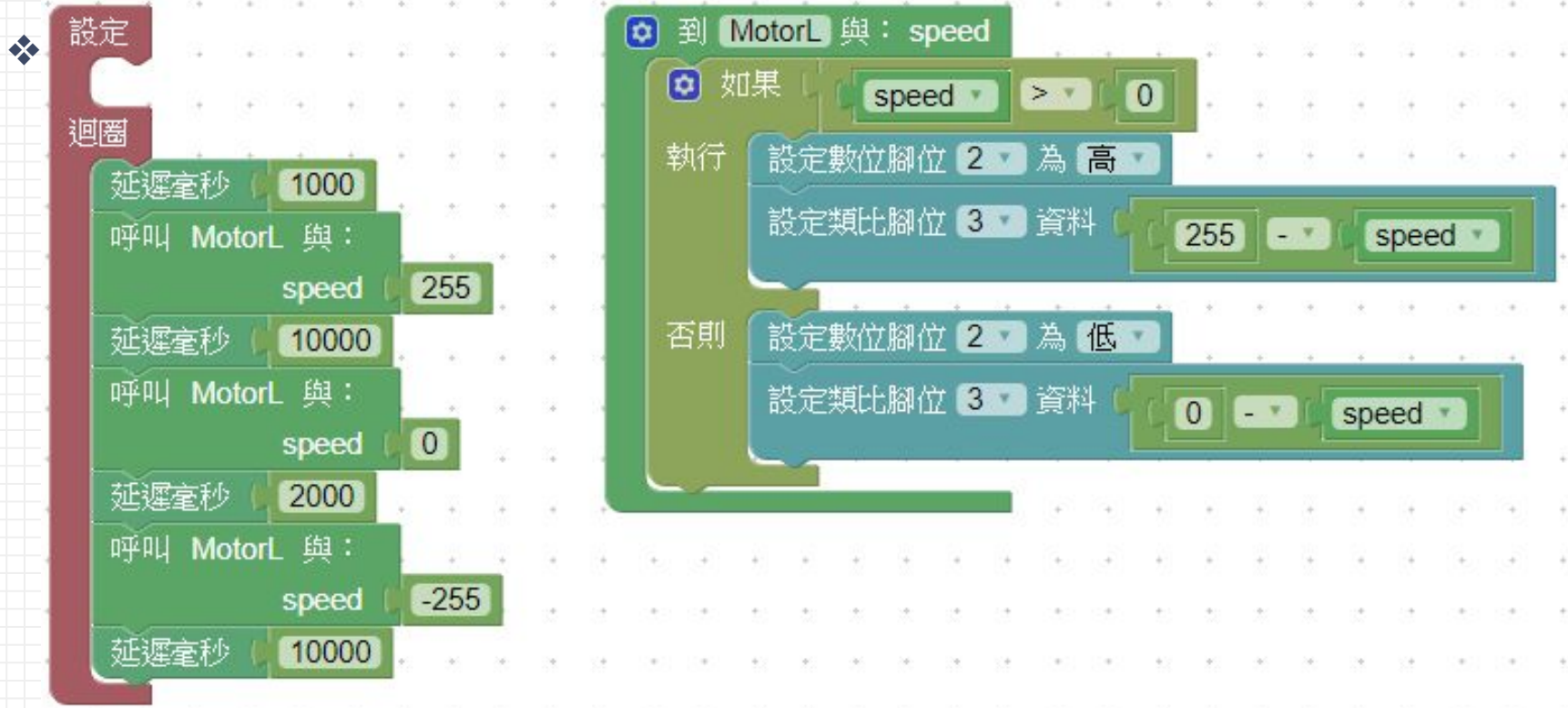
  前進 10 秒,

  停 2 秒,

  後退 10 秒

  Motor_L_ForwardBack_Function.xml

單輪控制


設定
迴圈
延遲毫秒 1000
呼叫 MotorL 與：
speed 255
延遲毫秒 10000
呼叫 MotorL 與：
speed 0
延遲毫秒 2000
呼叫 MotorL 與：
speed -255
延遲毫秒 10000
到 MotorL 與： speed
如果 speed > 0
執行
設定數位腳位 2 為 高
設定類比腳位 3 資料 255 - speed
否則
設定數位腳位 2 為 低
設定類比腳位 3 資料 0 - speed

單輪控制

- 右輪測試
- 自訂函式, 可控制轉動速率

  前進 10 秒,

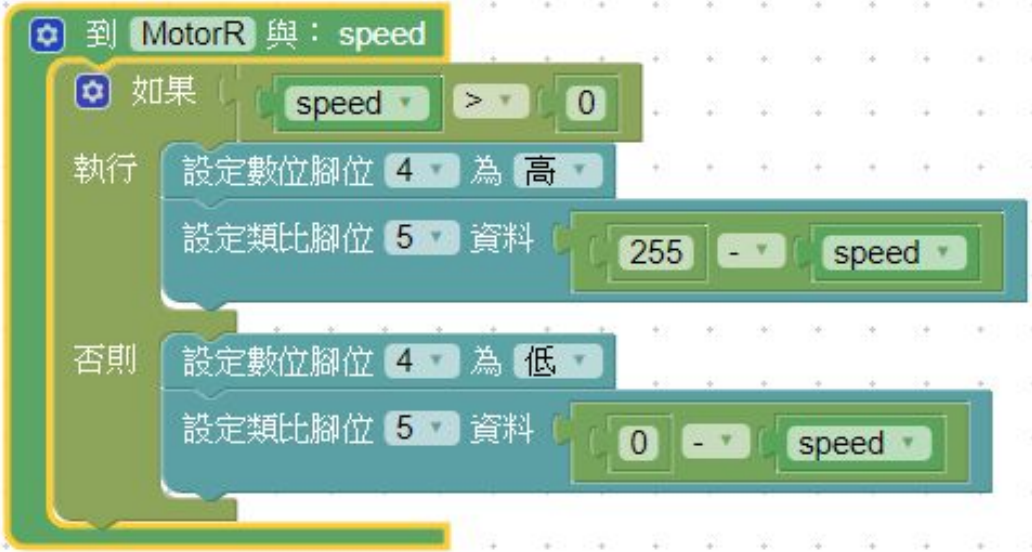
  停 2 秒,

  後退 10 秒

  Motor_R_ForwardBack_Function.xml

單輪控制


設定
迴圈
延遲毫秒 1000
呼叫 MotorR 與：
speed 255
延遲毫秒 10000
呼叫 MotorR 與：
speed 0
延遲毫秒 2000
呼叫 MotorR 與：
speed -255
延遲毫秒 10000


到 MotorR 與： speed
如果 speed > 0
執行
設定數位腳位 4 為 高
設定類比腳位 5 資料 255 - speed
否則
設定數位腳位 4 為 低
設定類比腳位 5 資料 0 - speed

單輪控制

- 左輪轉速測試
- 自訂函式, 可控制轉動速率

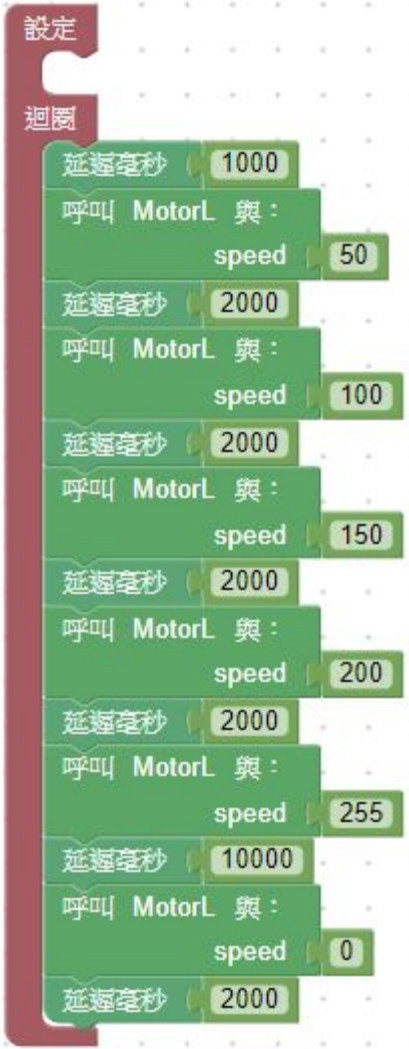
  轉速由 PWM 50 遞增 50, 漸漸增加至最大 255,

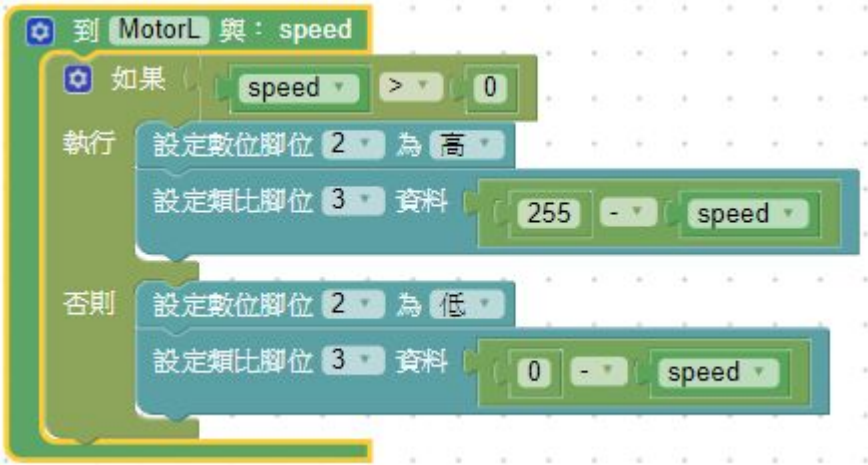
  停 2 秒,

  Motor_L_SpeedTest.xml

單輪控制

-


設定
迴圈
延遲毫秒 1000
呼叫 MotorL 與：
speed 50
延遲毫秒 2000
呼叫 MotorL 與：
speed 100
延遲毫秒 2000
呼叫 MotorL 與：
speed 150
延遲毫秒 2000
呼叫 MotorL 與：
speed 200
延遲毫秒 2000
呼叫 MotorL 與：
speed 255
延遲毫秒 10000
呼叫 MotorL 與：
speed 0
延遲毫秒 2000


到 MotorL 與： speed
如果 speed > 0
執行
設定數位腳位 2 為 高
設定類比腳位 3 資料 255 - speed
否則
設定數位腳位 2 為 低
設定類比腳位 3 資料 0 - speed

單輪控制

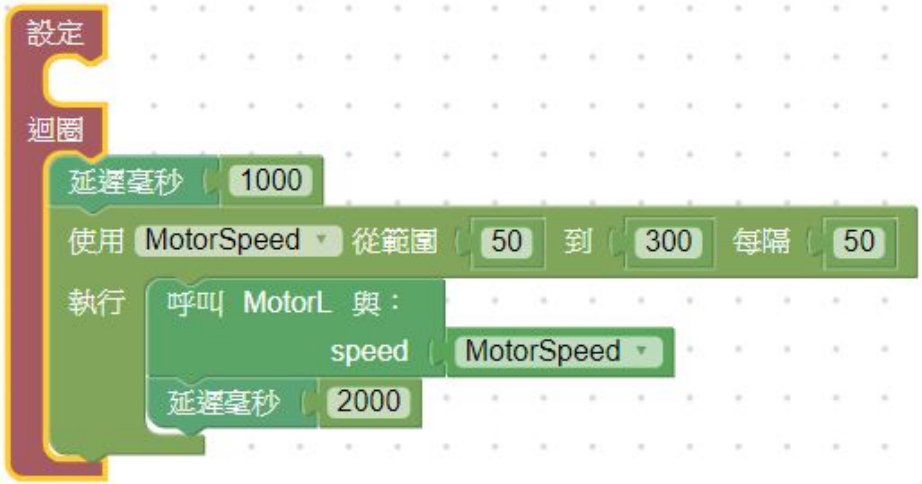
- 左輪轉速測試, 使用 for 迴圈
- 自訂函式, 可控制轉動速率

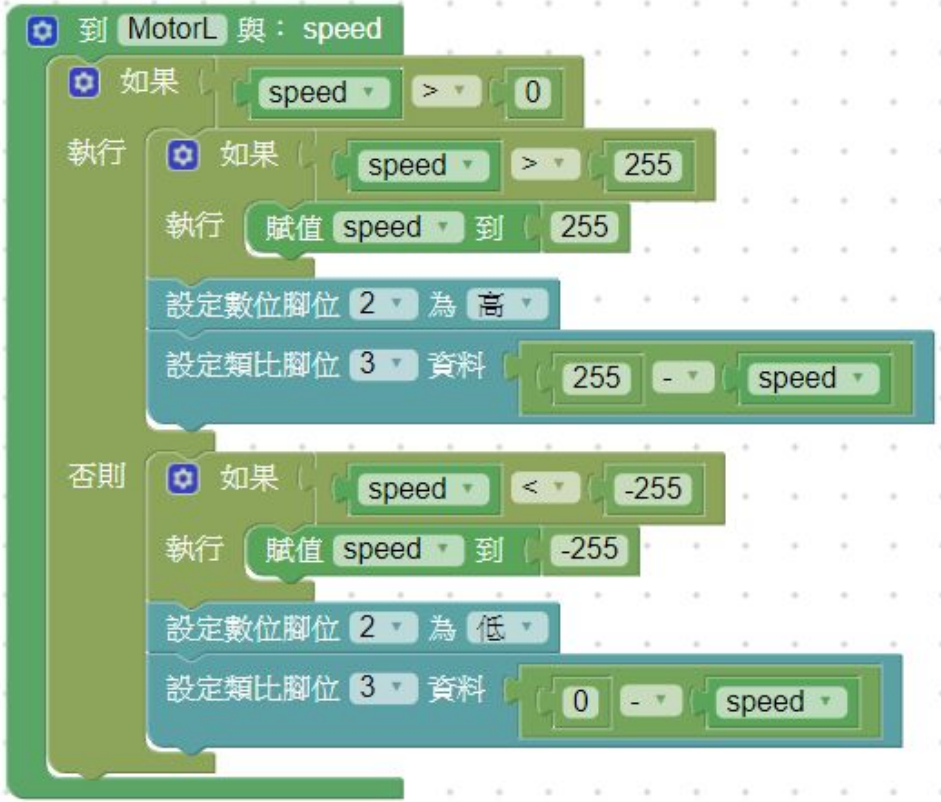
  轉速由 PWM 50 遞增 50, 漸漸增加至最大 255,

  停 2 秒,

  Motor_L_SpeedTest_ForLoop.xml

單輪控制


設定
迴圈
延遲毫秒 1000
使用 MotorSpeed 從範圍 50 到 300 每隔 50
執行
呼叫 MotorL 與：
speed MotorSpeed
延遲毫秒 2000


到 MotorL 與： speed
如果 speed > 0
執行
如果 speed > 255
執行 賦值 speed 到 255
設定數位腳位 2 為 高
設定類比腳位 3 資料 255 - speed
否則
如果 speed < -255
執行 賦值 speed 到 -255
設定數位腳位 2 為 低
設定類比腳位 3 資料 0 - speed

自走車控制

- 自走車前進、後退、左轉、右轉測試
- 單一自訂函式控制前進、後退、左轉、右轉、停止及二輪功率

  Motor_Move_Function.xml

自走車控制

- 全圖控程式


設定
重複
GO
SpeedL
SpeedR
延遲毫秒


MotorL
MotorR
speed
如果
執行
否則
設定數位腳位
設定類比腳位
資料


GO
SpeedL, SpeedR
MotorL
MotorR
speed

自走車控制

❖


到 MotorL 與：speed
如果 speed > 0
執行
設定數位腳位 2 為 高
設定類比腳位 3 資料 255 - speed
否則
設定數位腳位 2 為 低
設定類比腳位 3 資料 0 - speed


到 MotorR 與：speed
如果 speed > 0
執行
設定數位腳位 4 為 高
設定類比腳位 5 資料 255 - speed
否則
設定數位腳位 4 為 低
設定類比腳位 5 資料 0 - speed

自走車控制


到 GO 與： SpeedL, SpeedR
呼叫 MotorL 與：
speed SpeedL
呼叫 MotorR 與：
speed SpeedR

自走車控制

❖


設定
迴圈
呼叫 GO 與：
SpeedL 255
SpeedR 255
延遲毫秒 5000
呼叫 GO 與：
SpeedL 0
SpeedR 0
延遲毫秒 2000
呼叫 GO 與：
SpeedL -255
SpeedR -255
延遲毫秒 5000
呼叫 GO 與：
SpeedL 0
SpeedR 0
延遲毫秒 2000


呼叫 GO 與：
SpeedL 0
SpeedR 255
延遲毫秒 5000
呼叫 GO 與：
SpeedL 0
SpeedR 0
延遲毫秒 2000
呼叫 GO 與：
SpeedL 255
SpeedR 0
延遲毫秒 5000
呼叫 GO 與：
SpeedL 0
SpeedR 0
延遲毫秒 2000

按鍵自走車控制

- 按鍵控制自走車前進、後退、左轉、右轉
- 單一自訂函式控制前進、後退、左轉、右轉、停止及二輪功率

  Motor_Move_SerialControl.xml

按鍵自走車控制

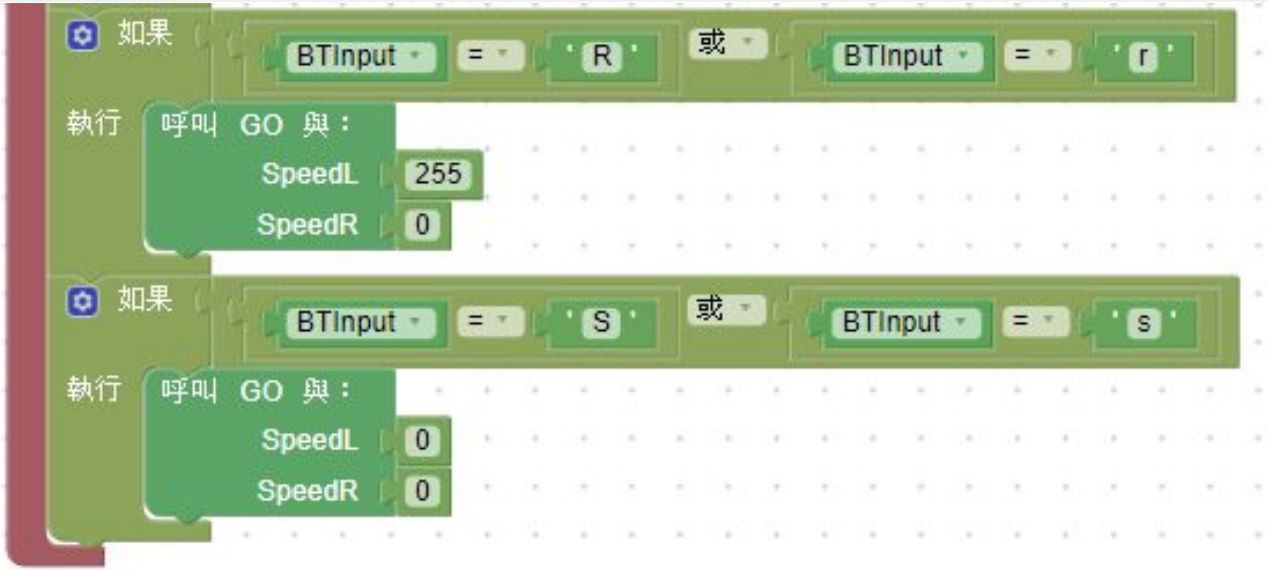

設定
設定串列埠 serial 傳輸率 9600 bps
宣告 BTInput 當 char 資料 ' '
印出訊息後換行 "***按鍵控制自走車***"
印出訊息後換行 "'F'or'f' ---> 前進..."
印出訊息後換行 "'B'or'b' ---> 後退..."
印出訊息後換行 "'L'or'lf' ---> 左轉..."
印出訊息後換行 "'R'or'r' ---> 右轉..."
印出訊息後換行 "'S'or's' ---> 停止..."

按鍵自走車控制

❖


迴圈
如果 串列埠有效資料? > 0
執行 賦值 BTInput 到 串列埠輸入
如果 BTInput = 'F' 或 BTInput = 'f'
執行 呼叫 GO 與：
SpeedL 255
SpeedR 255
如果 BTInput = 'B' 或 BTInput = 'b'
執行 呼叫 GO 與：
SpeedL -255
SpeedR -255
如果 BTInput = 'L' 或 BTInput = 'l'
執行 呼叫 GO 與：
SpeedL 0
SpeedR 255

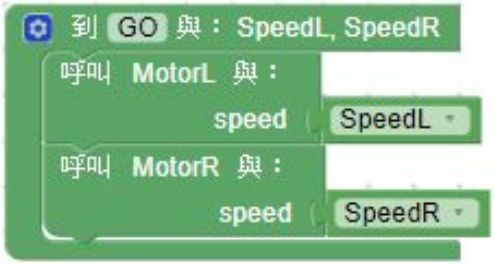
按鍵自走車控制


如果
BTInput = 'R' 或 BTInput = 'r'
執行
呼叫 GO 與：
SpeedL 255
SpeedR 0
如果
BTInput = 'S' 或 BTInput = 's'
執行
呼叫 GO 與：
SpeedL 0
SpeedR 0

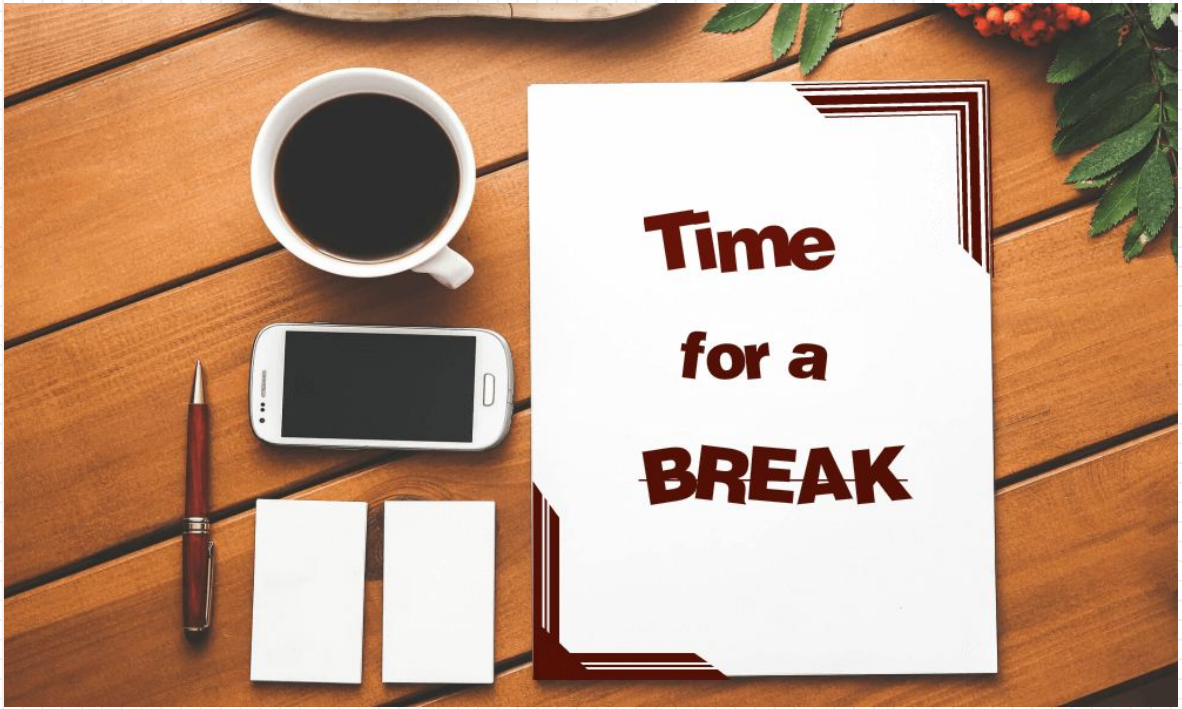
按鍵自

❖

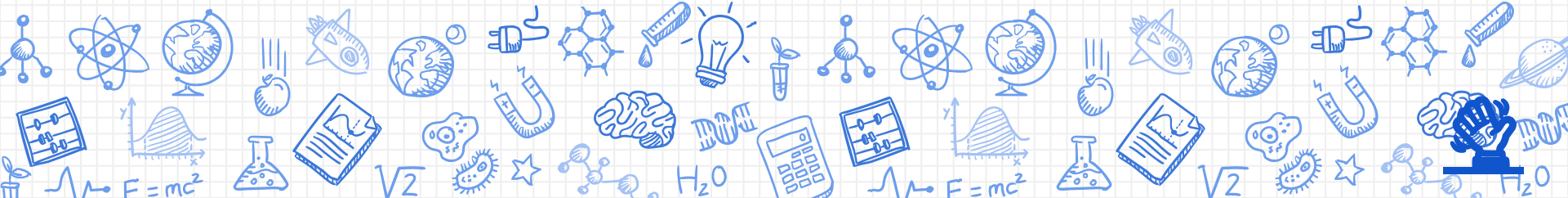

到 MotorL 與： speed
如果 speed > 0
執行
設定數位腳位 2 為 高
設定類比腳位 3 資料 255 - speed
否則
設定數位腳位 2 為 低
設定類比腳位 3 資料 0 - speed


到 GO 與： SpeedL, SpeedR
呼叫 MotorL 與：
speed SpeedL
呼叫 MotorR 與：
speed SpeedR


到 MotorR 與： speed
如果 speed > 0
執行
設定數位腳位 4 為 高
設定類比腳位 5 資料 255 - speed
否則
設定數位腳位 4 為 低
設定類比腳位 5 資料 0 - speed


Time
for a
BREAK

手機藍芽自走車



手機控制自走車

- Arduino 端:接收手機 App 傳送來的特定指令, 並依指定動作。
- 手機端:發送特定指令給 Arduino, 使依特定指定動作。
- 橋樑:通訊元件

  藍芽

  WIFI

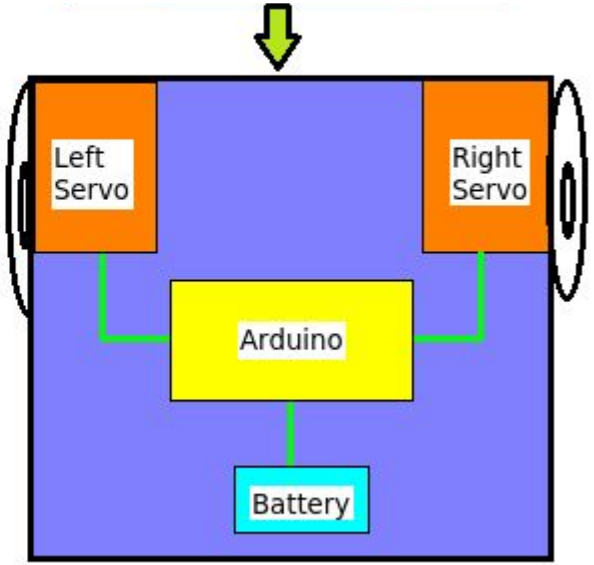
  RF(433M RF)

手機控制自走車

手機 App 程式

藍芽
WIFI

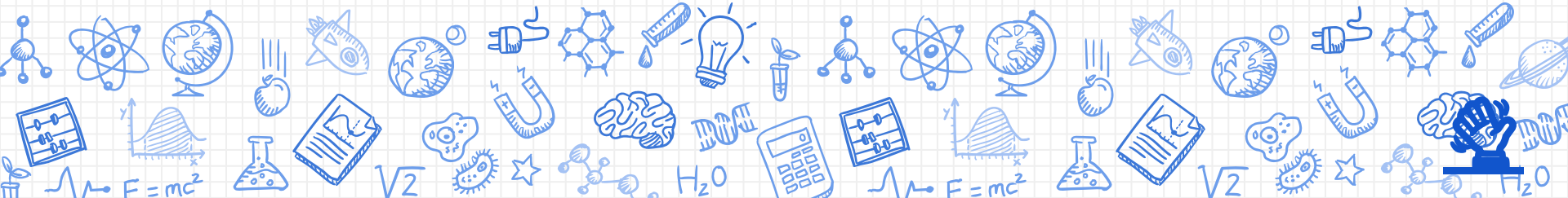
藍芽模組(HC-05,HC-06)
WIFI模組(ESP-01,ESP-12)



Arduino 程式

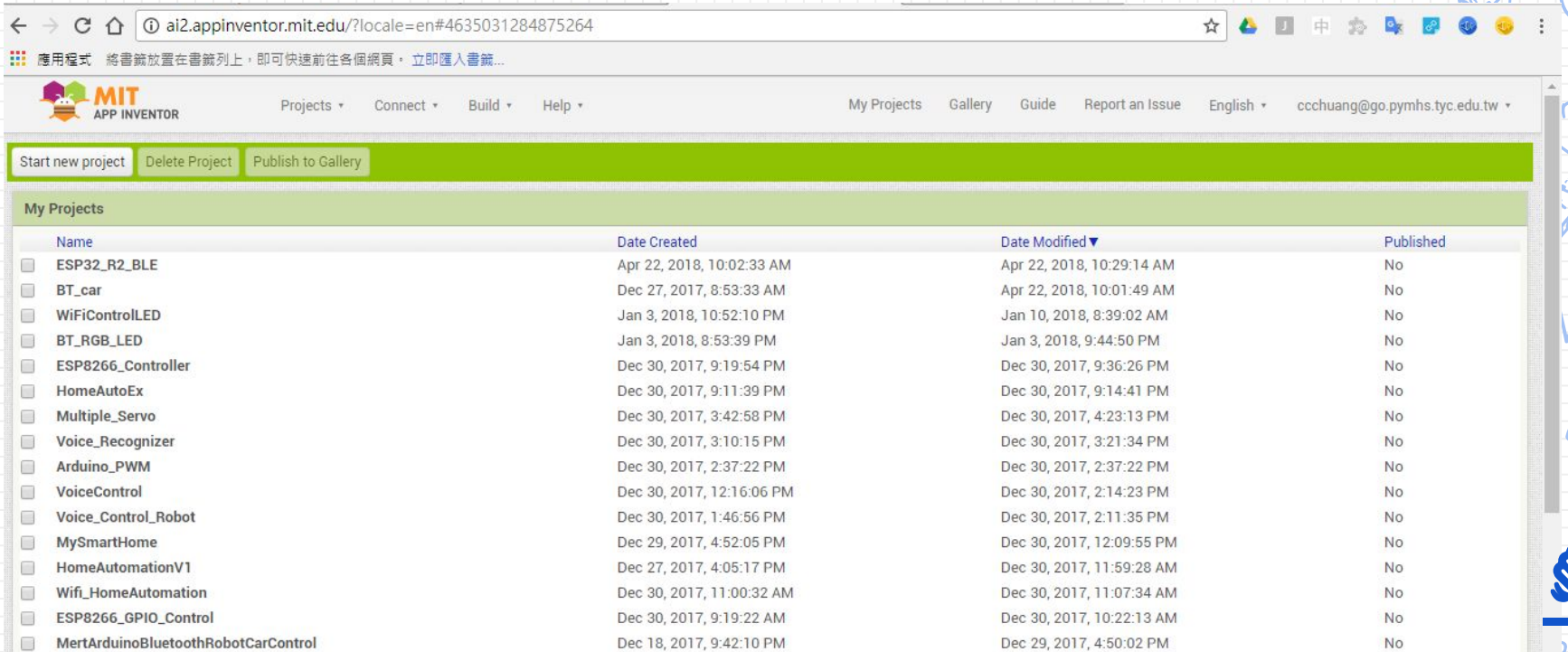
App 控制程式

App Inventor2



App Inventor 2

- http://ai2.appinventor.mit.edu/
- Android App 視覺化程式設計



Name	Date Created	Date Modified ▼	Published
ESP32_R2_BLE	Apr 22, 2018, 10:02:33 AM	Apr 22, 2018, 10:29:14 AM	No
BT_car	Dec 27, 2017, 8:53:33 AM	Apr 22, 2018, 10:01:49 AM	No
WiFiControlLED	Jan 3, 2018, 10:52:10 PM	Jan 10, 2018, 8:39:02 AM	No
BT_RGB_LED	Jan 3, 2018, 8:53:39 PM	Jan 3, 2018, 9:44:50 PM	No
ESP8266_Controller	Dec 30, 2017, 9:19:54 PM	Dec 30, 2017, 9:36:26 PM	No
HomeAutoEx	Dec 30, 2017, 9:11:39 PM	Dec 30, 2017, 9:14:41 PM	No
Multiple_Servo	Dec 30, 2017, 3:42:58 PM	Dec 30, 2017, 4:23:13 PM	No
Voice_Recognizer	Dec 30, 2017, 3:10:15 PM	Dec 30, 2017, 3:21:34 PM	No
Arduino_PWM	Dec 30, 2017, 2:37:22 PM	Dec 30, 2017, 2:37:22 PM	No
VoiceControl	Dec 30, 2017, 12:16:06 PM	Dec 30, 2017, 2:14:23 PM	No
Voice_Control_Robot	Dec 30, 2017, 1:46:56 PM	Dec 30, 2017, 2:11:35 PM	No
MySmartHome	Dec 29, 2017, 4:52:05 PM	Dec 30, 2017, 12:09:55 PM	No
HomeAutomationV1	Dec 27, 2017, 4:05:17 PM	Dec 30, 2017, 11:59:28 AM	No
Wifi_HomeAutomation	Dec 30, 2017, 11:00:32 AM	Dec 30, 2017, 11:07:34 AM	No
ESP8266_GPIO_Control	Dec 30, 2017, 9:19:22 AM	Dec 30, 2017, 10:22:13 AM	No
MertArduinoBluetoothRobotCarControl	Dec 18, 2017, 9:42:10 PM	Dec 29, 2017, 4:50:02 PM	No

離線版

- App Inventor 2 Ultimate
- https://sourceforge.net/projects/ai2u/

  安裝版:

  AI2U 64bit v4.3.exe

  AI2U 32bit v4.3.exe

  Portable:

  AI2U4364.zip

  AI2U4332.zip

離線版

- 執行安裝 App Inventor 2 Ultimate
- 安裝完成後點選桌面 App Inventor 2 Ultimate，會自動執行App Inventor 2伺服器、Build伺服器與AI Starter。
- 使用瀏覽器瀏覽 http://127.0.0.1:8888/

手機控制 0406 小車

- 手機透過藍芽控制 0406 小車
- 2A1B

  手機端程式(Android)

  控制板端程式(Arduino)

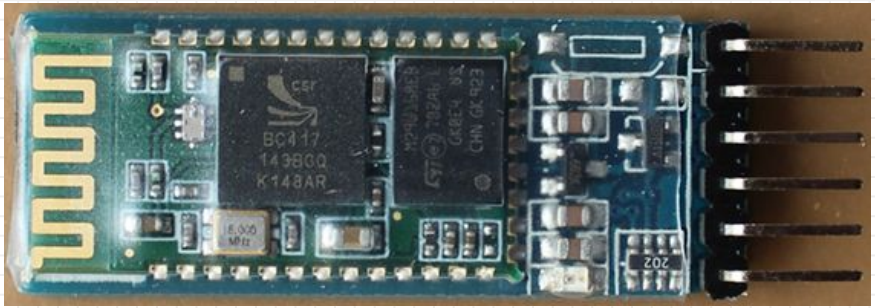
  藍芽

HC-05 藍芽模組

- 藍芽協定: BLUETOOTH V2.0 + EDR (ENHANCED DATA RATE)
- 工作頻率: 2.4~2.48GHZ, ISM BAND
- 傳輸距離: 空曠地有效距離 10 公尺
- 介面: UART
- 輸入電壓: 3.3V ~ 4.2V
- 工作溫度: -20℃ ~ +75℃

HC-05 藍芽模組

- 原始模組使用工作電壓為 3.3V ~ 4.2V, 因為 ARDUINO 一般是 5V 的, 如果要跟 ARDUINO 連接, 得利用 LDO REGULATOR (LOW DROP OUT REGULATOR) 轉換電壓, 對一般使用者會有些困難。
- 所幸市面上可以買到帶底板的模組。這類帶底板的模組已經幫你做好了 5V 與 3.3V 電壓轉換, 而且還拉出引腳方便連接線路



HC-05 藍芽設定

- 執行 Arduino IDE → 開啟 HC05_AT.ino → 上傳 → 去電
- 卸除 USB 接線

  HC-05 RX ↔ D11

  HC-05 TX ↔ D10

  HC-05 Vcc ↔ Vcc(5V)

  HC-05 GND ↔ GND

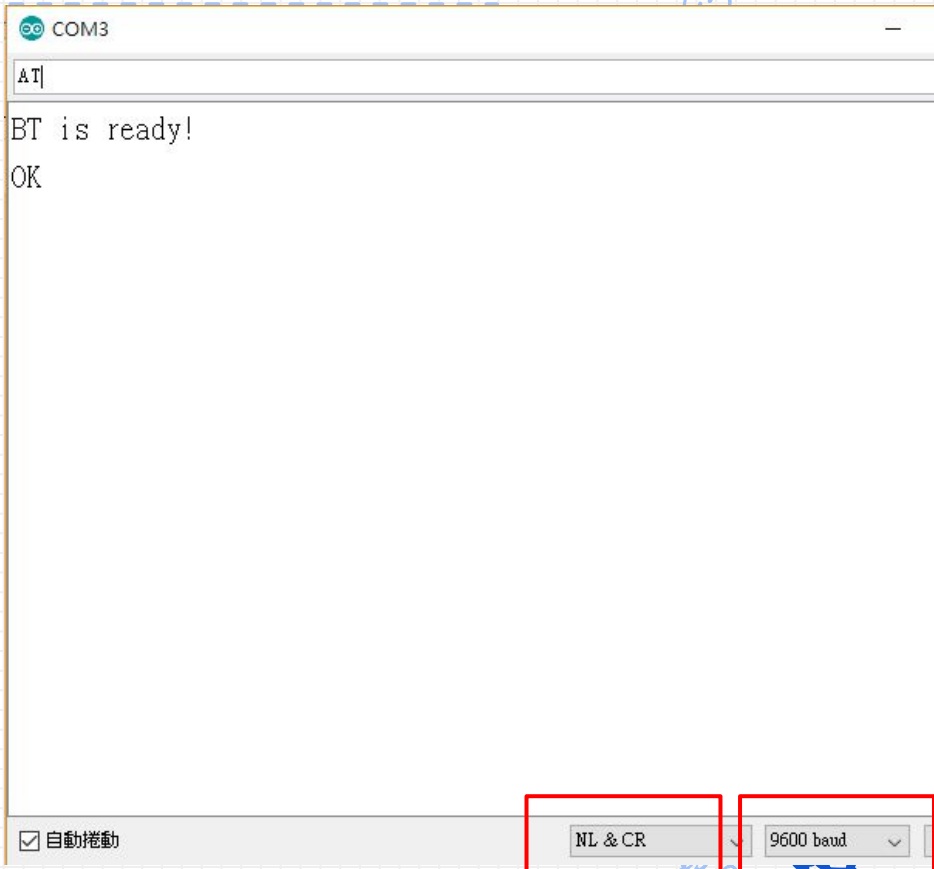
藍芽設定

- 藍芽進入 AT 模式:

  按住 HC-05藍芽晶片中的小按鈕, USB 再接上電腦, 直到指示燈呈現慢閃狀態再放開小按鈕。此即進入 AT 模式, 可利用 AT 命令進行設定

藍芽設定

- 開啟Arduino IDE 序列埠監控視窗
- 請把沒有行結尾改成 NL & CR
- baudrate:9600



藍芽設定

- 測試是否成功進入 AT mode
- 輸入 AT → 傳送

  回應:OK

藍芽設定

- 更改藍芽名稱

  AT+NAME=藍芽名稱

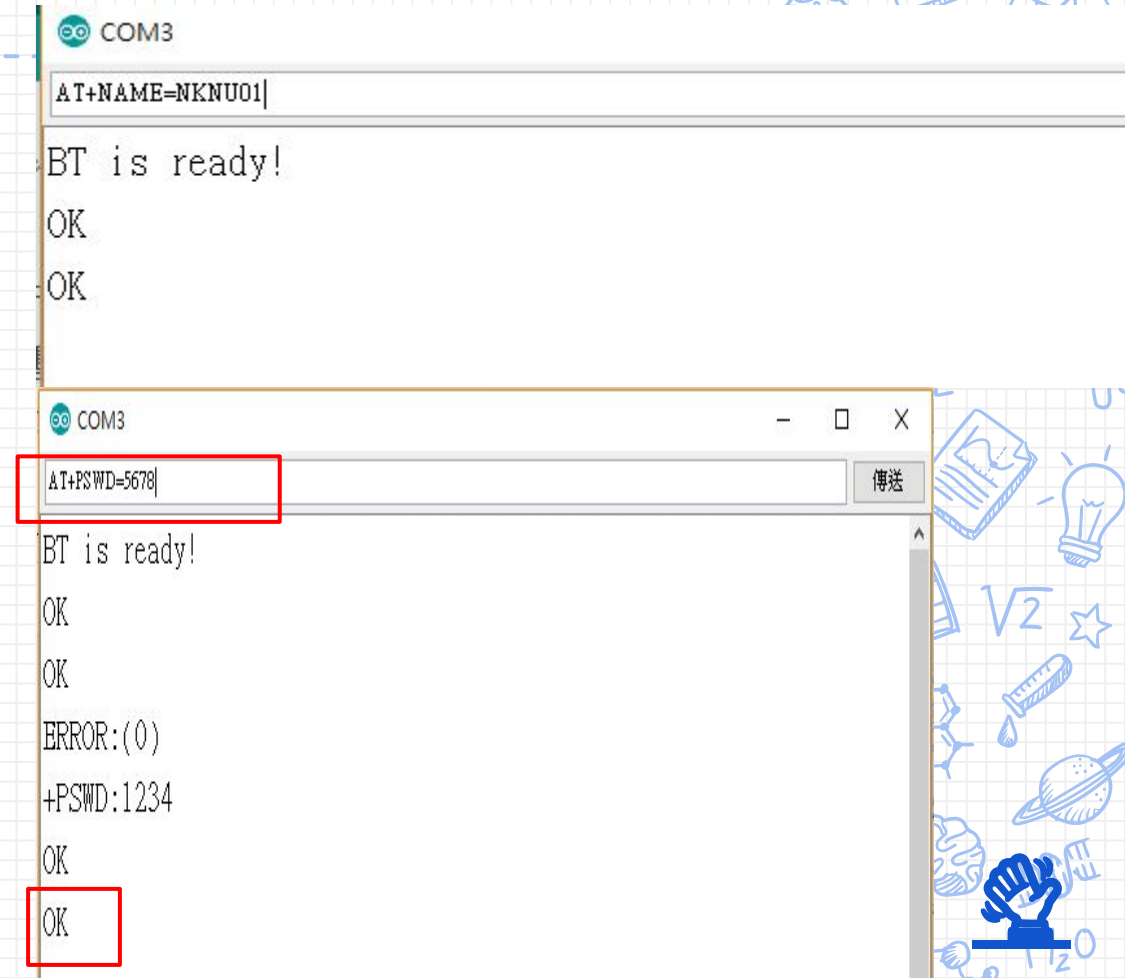
  回應:OK

- 更改藍芽配對密碼

  AT+PWSD=新密碼

  回應:OK

  預設:1234, 建議不修改



藍芽設定

- AT+ADDR?

  查詢藍芽 address

- AT+VERSION?

  查詢藍芽韌體版本

- AT+PWSD?

  查詢配對密碼

- AT+NAME?

  查詢藍芽名稱

藍芽設定

- AT+ADDR?

  查詢藍芽 address

- AT+VERSION?

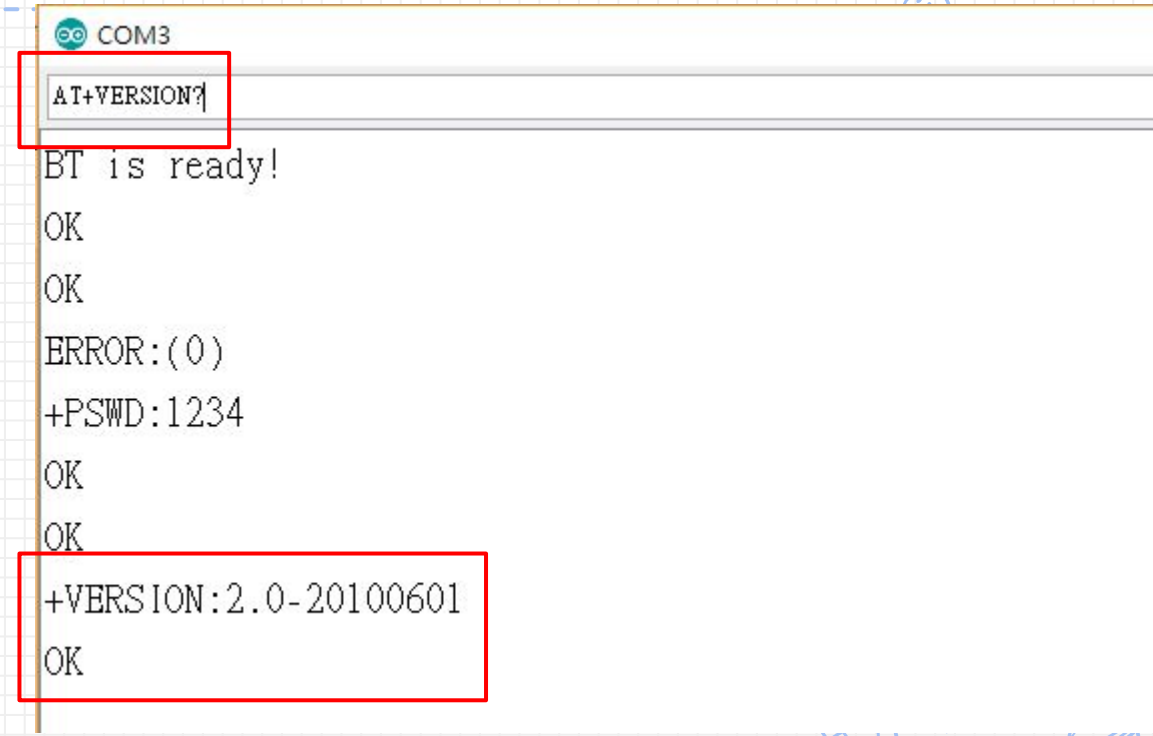
  查詢藍芽韌體版本

- AT+PWSD?

  查詢配對密碼

- AT+NAME?

  查詢藍芽名稱



藍芽設定

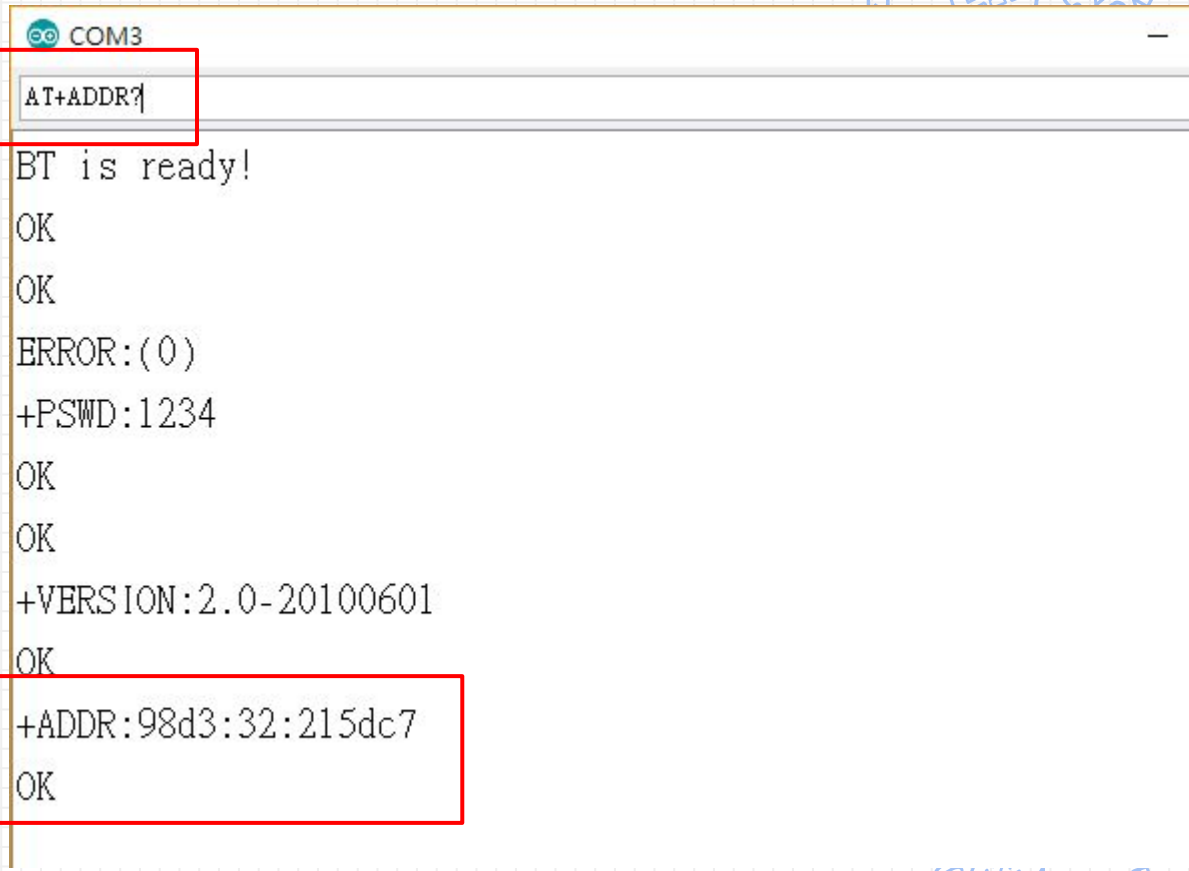
- AT+ADDR?

  查詢藍芽 address
- AT+VERSION?

  查詢藍芽韌體版本
- AT+PWSD?

  查詢配對密碼
- AT+NAME?

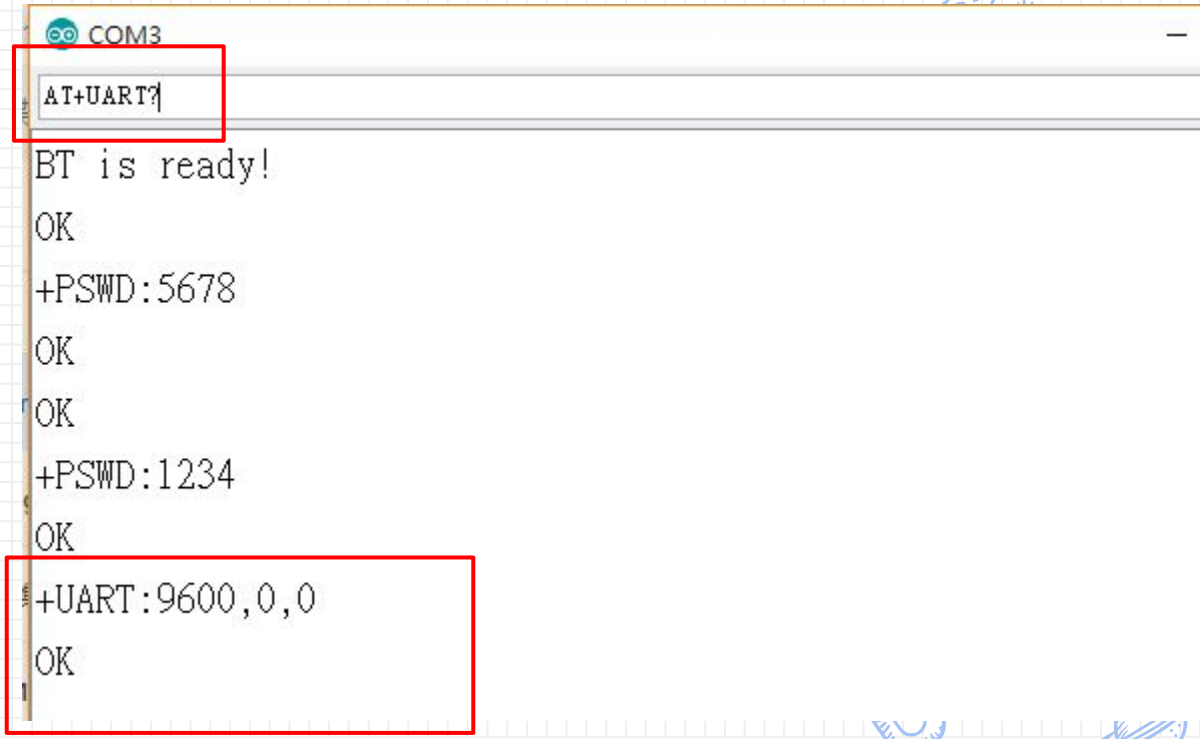
  查詢藍芽名稱



HC-05 藍芽設定

- AT+UART?

  查詢藍芽 baudrate



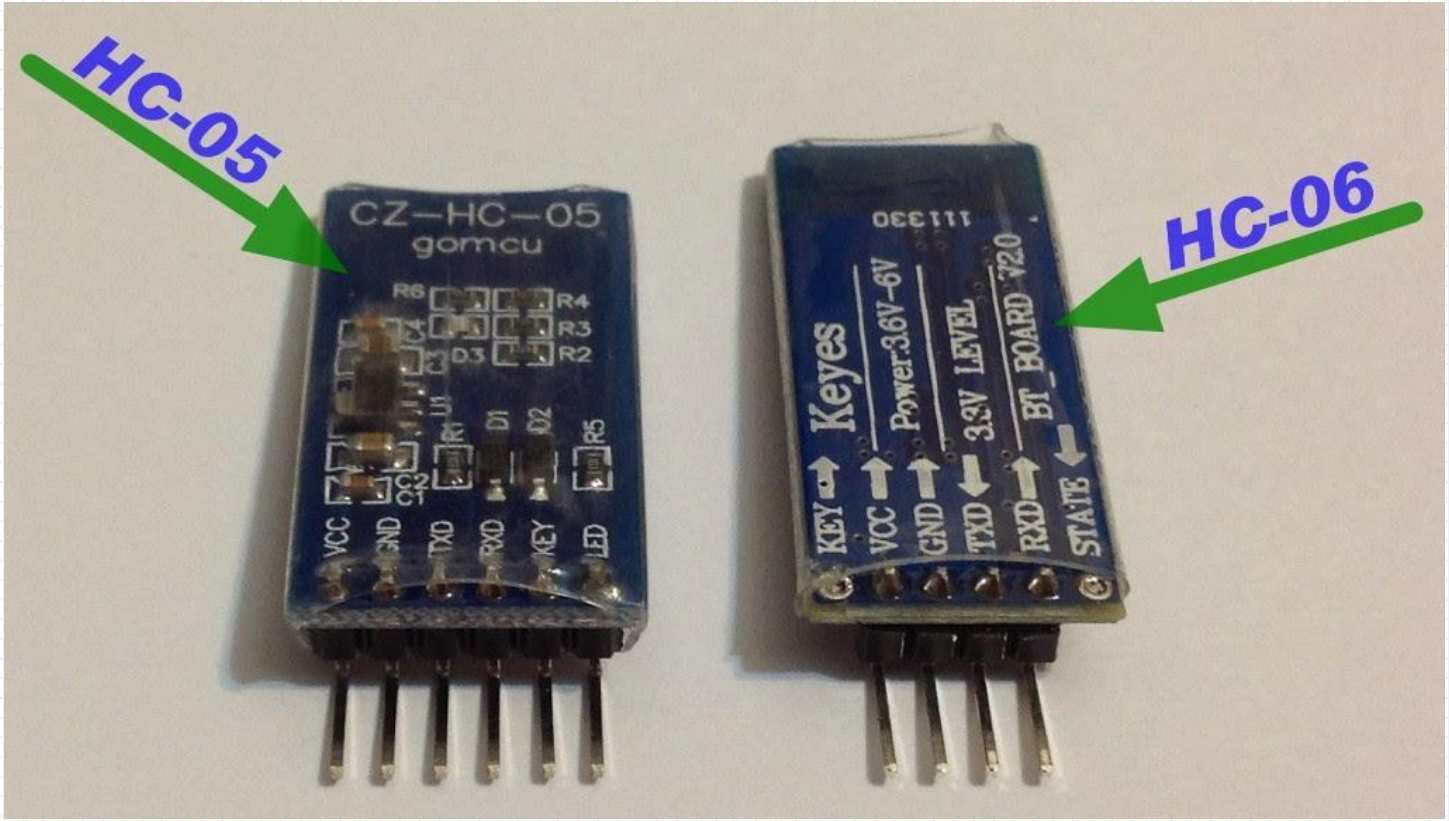
HC-05 藍芽設定

- 設定完畢, USB 重新上電, 進入工作模式

  快閃:待連線狀態

  慢閃:配對成功(連線)狀態

HC-06 藍芽設定


HC-05
CZ—HC—05
gomcu
VCC
GND
TXD
RXD
KEY
LED
HC-06
KEY
VCC
GND
TXD
RXD
STATE
Keyes
Power 3.6V-6V
3.3V LEVEL
BT_BOARD V2.0

HC-06 藍芽設定

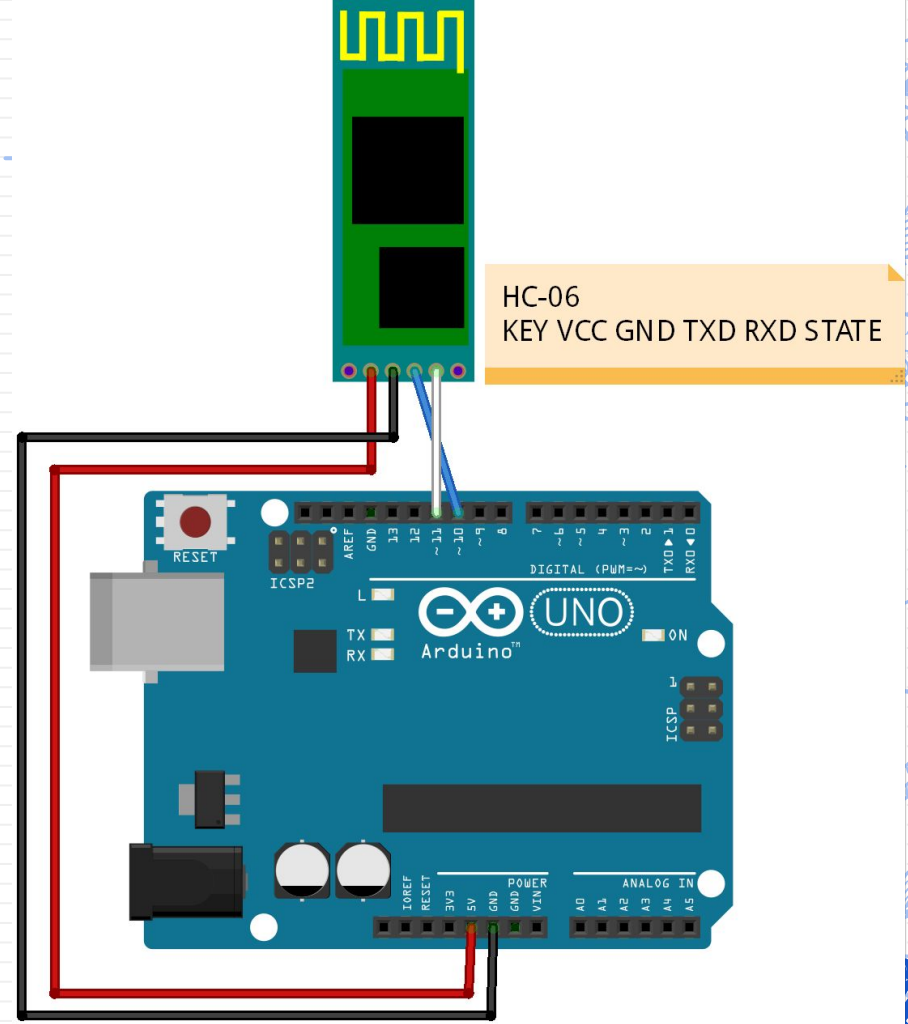
- 以 Arduino 與 HC-06 連線

  HC-06 VCC → Arduino 5V

  HC-06 GND → Arduino GND

  HC-06 TXD → Arduino pin 10

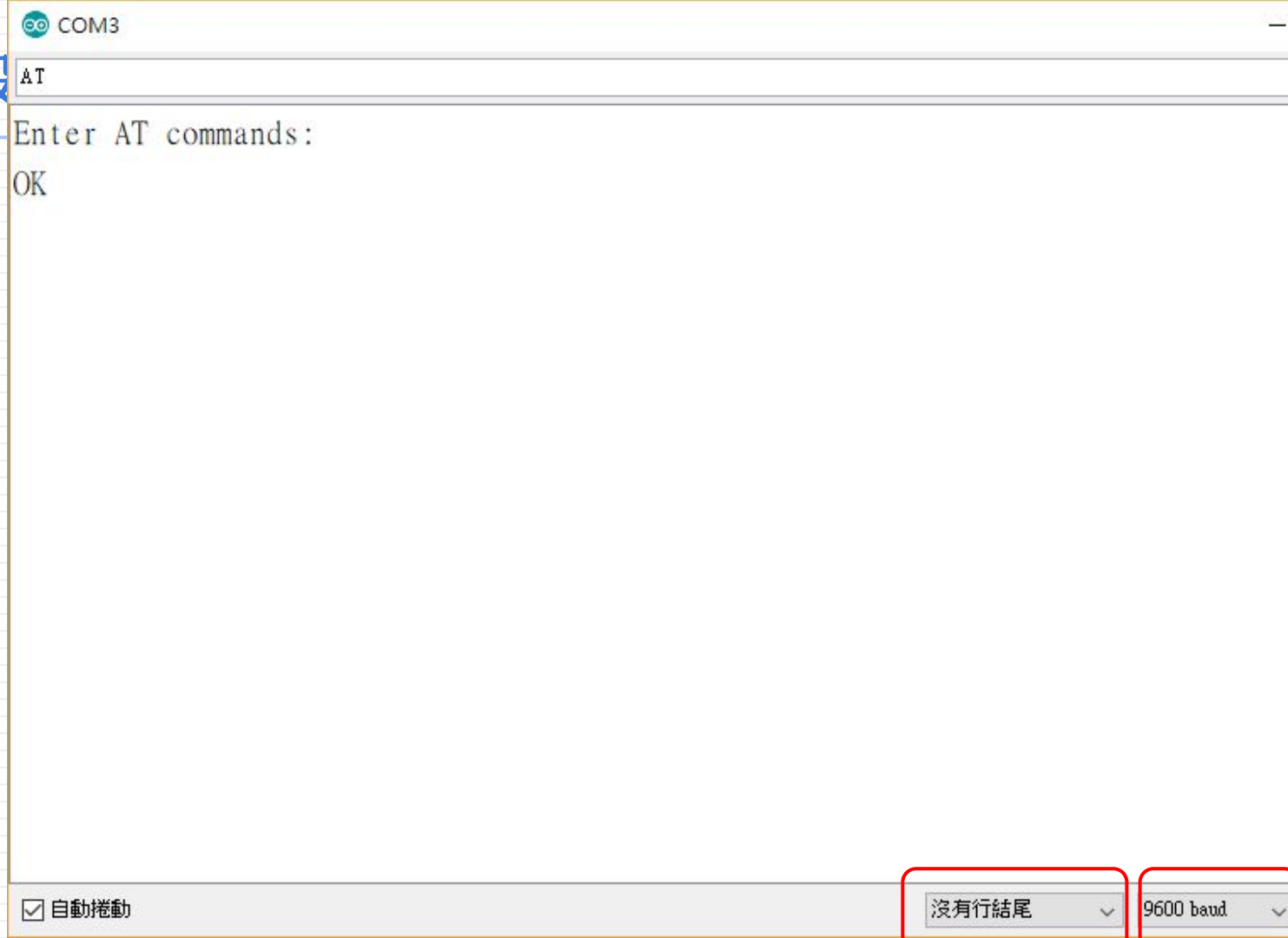
  HC-06 RXD → Arduino pin 11



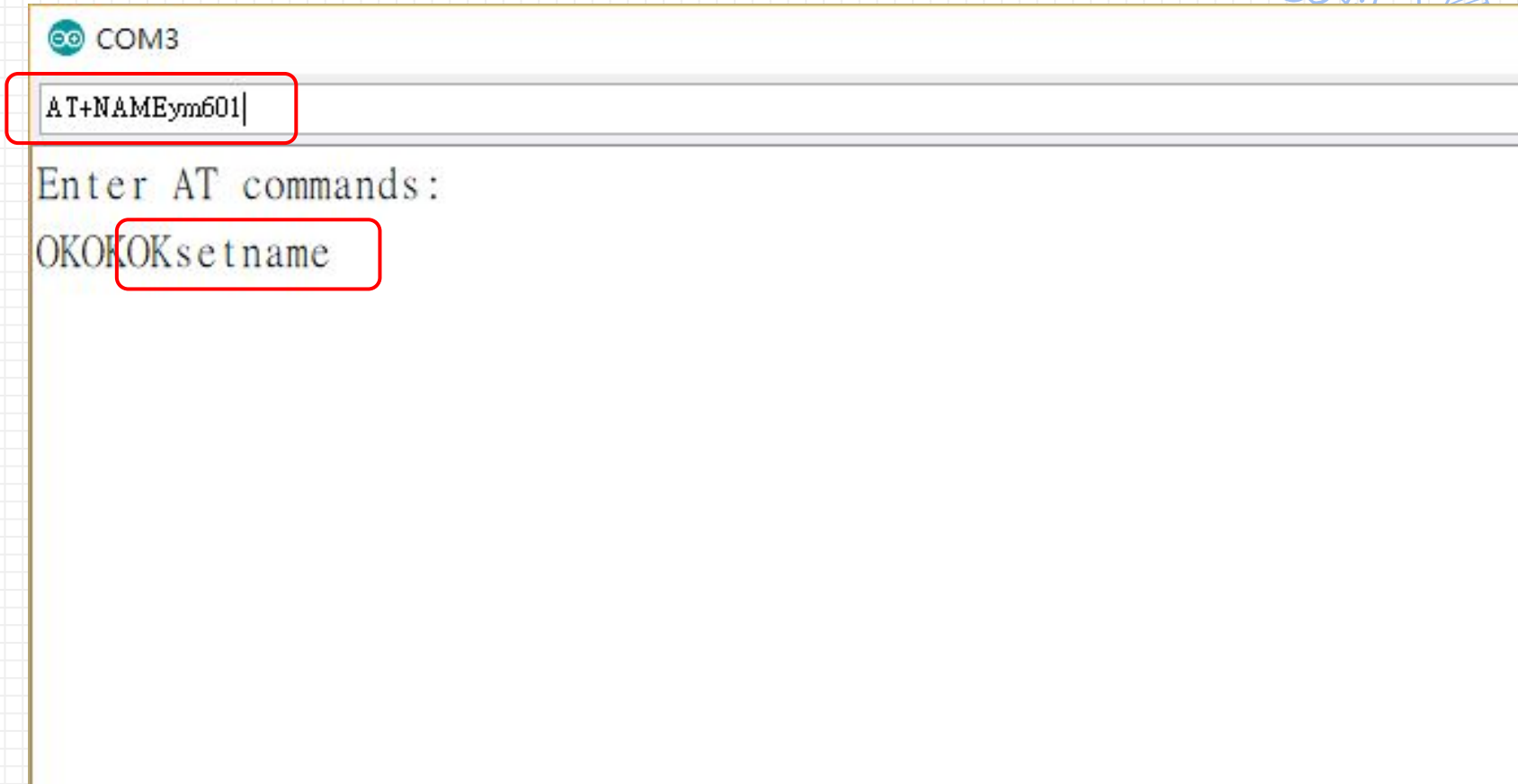
HC-06 藍芽設定程式

```
#include <SoftwareSerial.h>
SoftwareSerial BTSerial(10, 11); // RX | TX
void setup()
{
  Serial.begin(9600);
  Serial.println("Enter AT commands:");
  BTSerial.begin(9600);  // HC-06 current bound rate (default 9600)
}
void loop()
{
  // Keep reading from HC-06 and send to Arduino Serial Monitor
  if (BTSerial.available())
    Serial.write(BTSerial.read());
  // Keep reading from Arduino Serial Monitor and send to HC-06
```

HC-06 藍芽設



HC-06 藍芽設定



HC-06 的 AT command

- HC-06 的 AT command 只有簡單的幾項, 而且不像 HC-05 必須按 Enter 鍵送出「\r\n」才會執行並回應。所以, 我們一按完「AT」兩個按鍵, 它馬上就回應「OK」了。

HC-06 的 AT command

- AT：測試，回應「OK」

  AT+VERSION：回應韌體的版本。

  AT+NAMExyz：將裝置名稱改為「xyz」。

  AT+PIN1234：將連線密碼換為「1234」。

  AT+BAUD4：將 baud rate 換為 9600。

  AT+BAUD5：將 baud rate 換為 19200

  AT+BAUD6：將 baud rate 換為 38400

  AT+BAUD7：將 baud rate 換為 57600

HC-06 的 AT command

- Command Reply Comment

  AT OK Communications test

  AT+VERSION OKlinvorV1.8 Firmware version.

  AT+NAMEmyBTmodule OKsetname Sets the modules name to “myBTmodule”

  AT+PIN6789 OKsetPIN Set the PIN to 6789

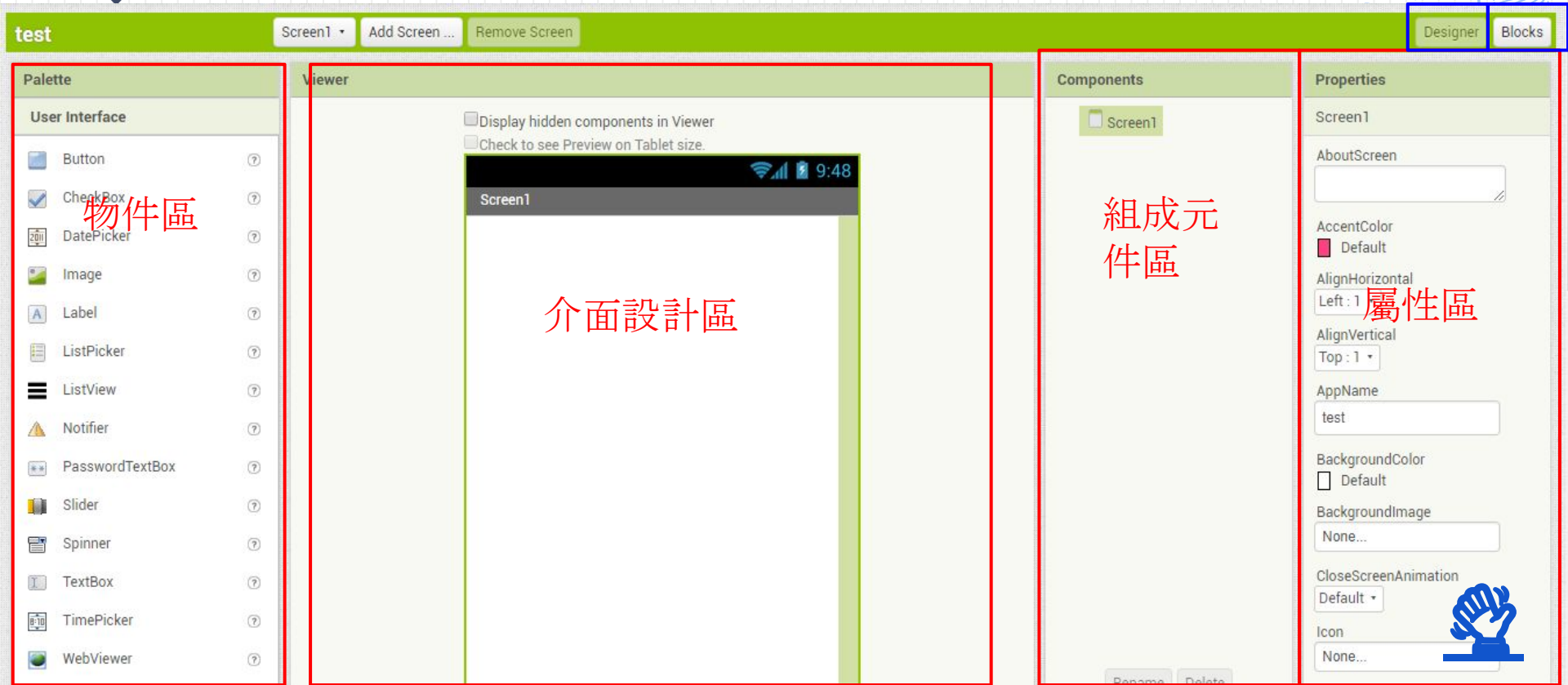
  AT+BAUD1 OK1200 Sets the baud rate to 1200

  AT+BAUD2 OK2400 Sets the baud rate to 2400

  AT+BAUD3 OK4800 Sets the baud rate to 4800

  AT+BAUD4 OK9600 Sets the baud rate to 9600

App 程式設計


test
Screen1
Add Screen ...
Remove Screen
Designer
Blocks
Palette
User Interface
Button
CheckBox
DatePicker
Image
Label
ListPicker
ListView
Notifier
PasswordTextBox
Slider
Spinner
TextBox
TimePicker
WebViewer
物件區
Viewer
Display hidden components in Viewer
Check to see Preview on Tablet size.
9:48
Screen1
介面設計區
Components
Screen1
組成元件區
Properties
Screen1
AboutScreen
AccentColor
Default
AlignHorizontal
Left : 1
AlignVertical
Top : 1
AppName
test
BackgroundColor
Default
BackgroundImage
None...
CloseScreenAnimation
Default
Icon
None...
屬性區

App 程式設計

- Designer：App 使用者介面設計

  調色板(palette)區：有常用的物件, 分類呈現

  介面設計區：手機 App 顯示頁面

  組成元件區：介面設計區所使用的物件

  屬性區：介面設計區所使用的個別物件屬性

- Blocks：積木程式設計

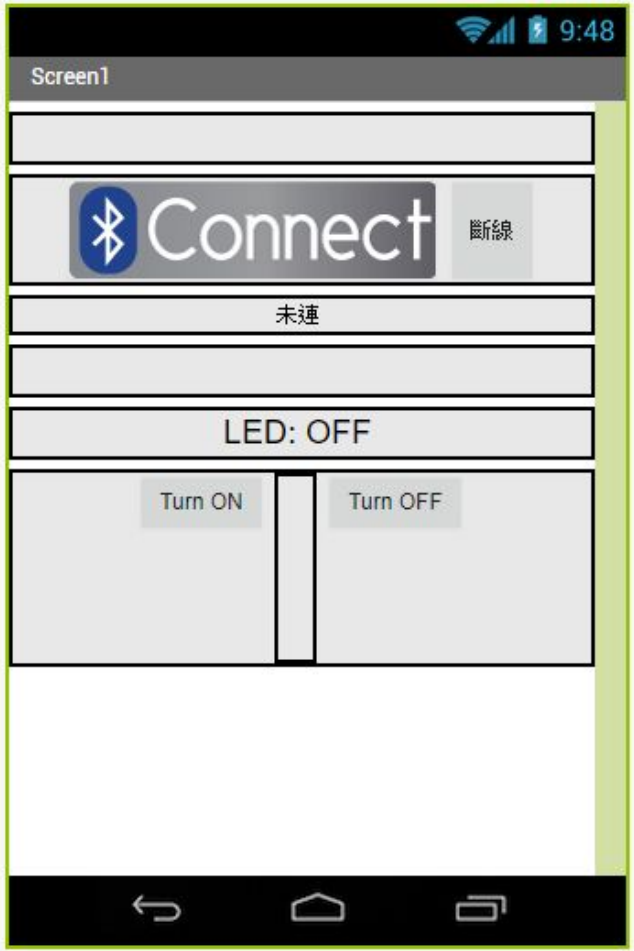
App 程式設計 初體驗

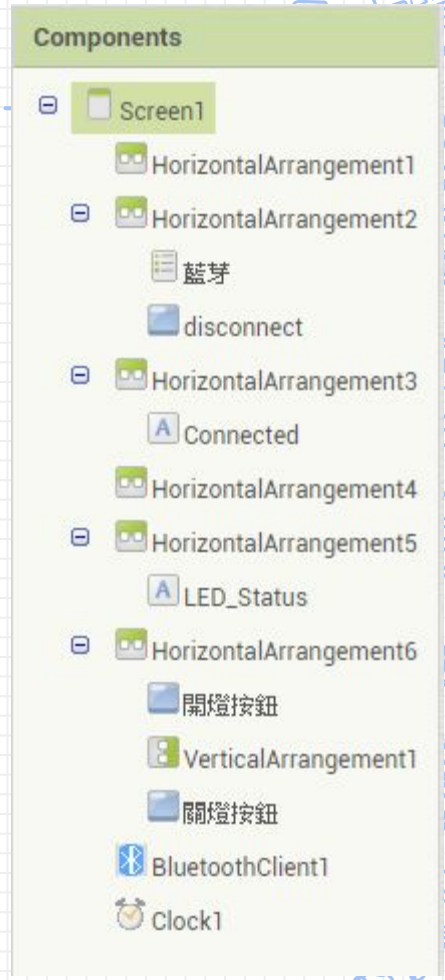
- 控制 LED 燈 App
- BT_Led.aia

  BT_Led.apk

App 程式設計 初

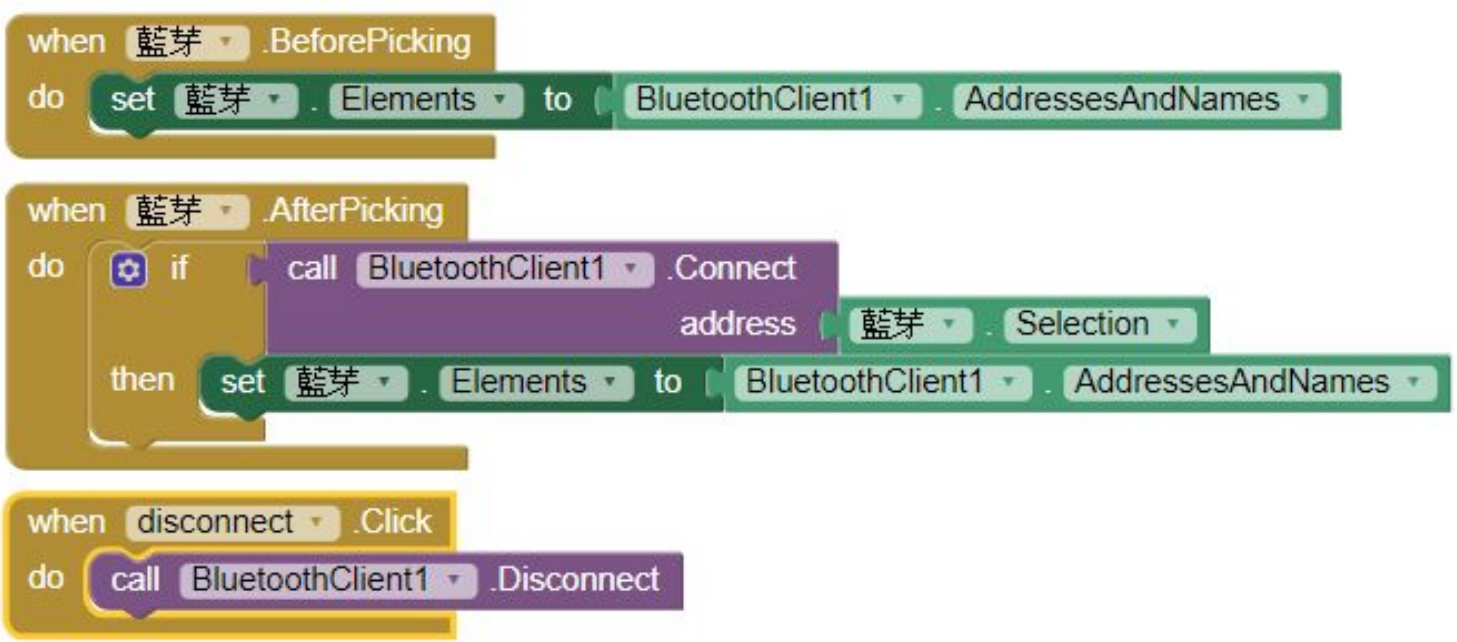
- App UI 設計





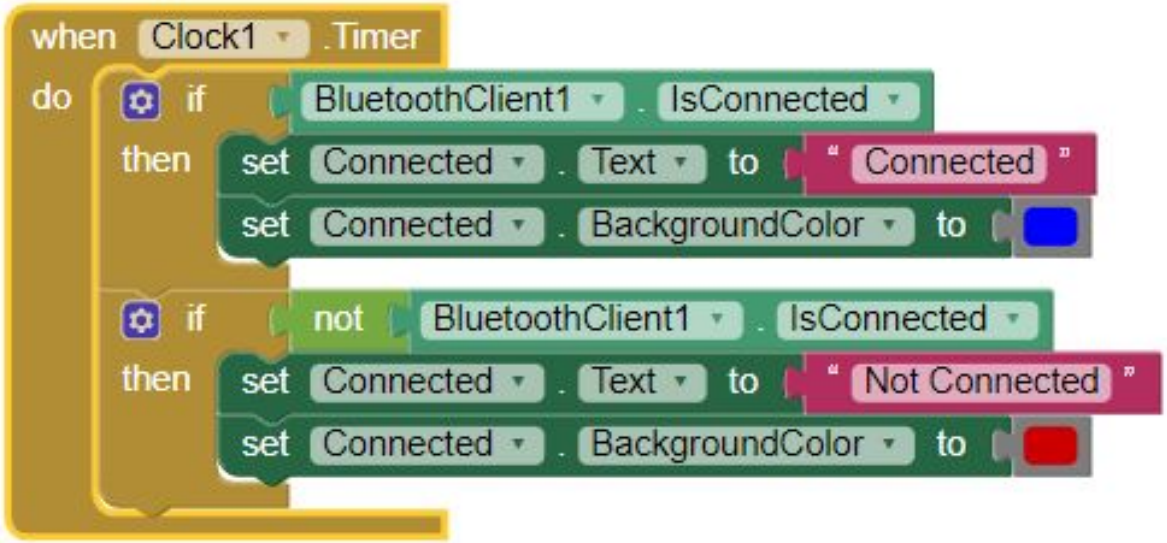
App 程式設計 初體驗

- 積木程式設計

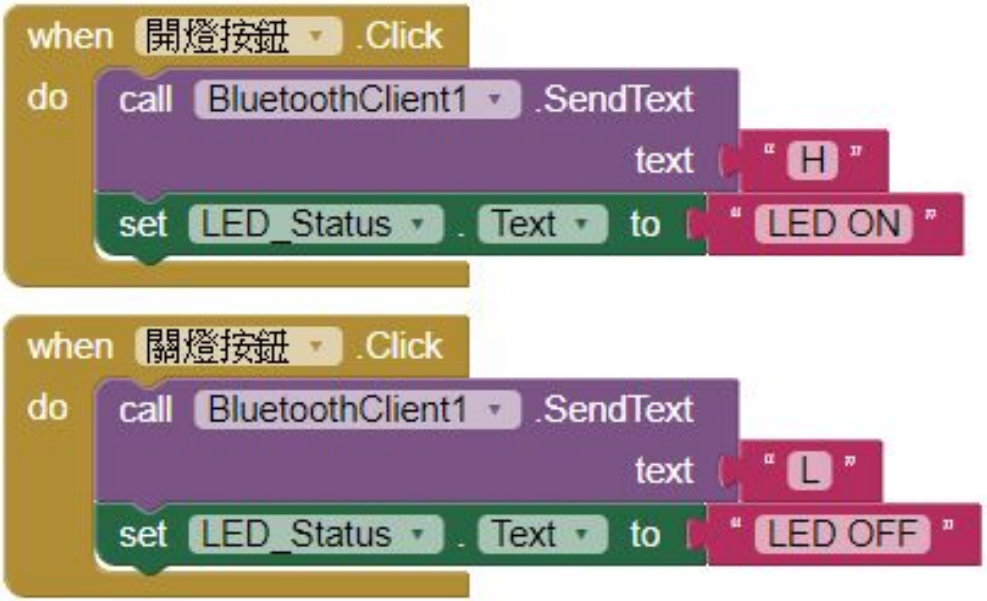


App 程式設計 初體驗

❖

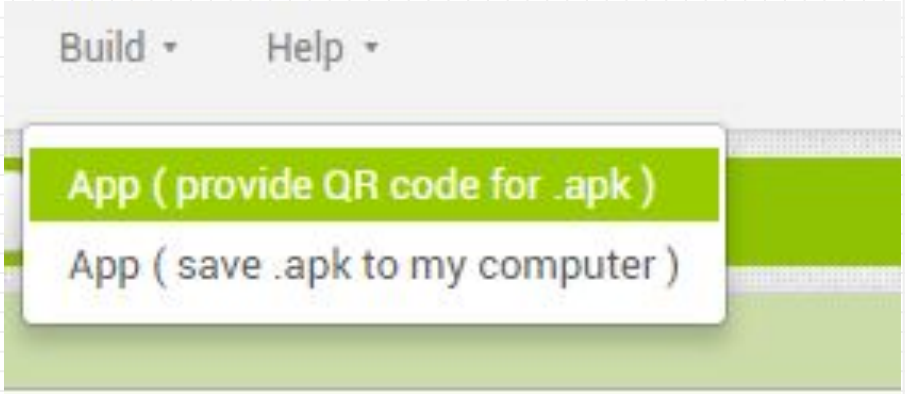

when Clock1 .Timer
do
if BluetoothClient1 . IsConnected
then
set Connected . Text to " Connected "
set Connected . BackgroundColor to
if not BluetoothClient1 . IsConnected
then
set Connected . Text to " Not Connected "
set Connected . BackgroundColor to

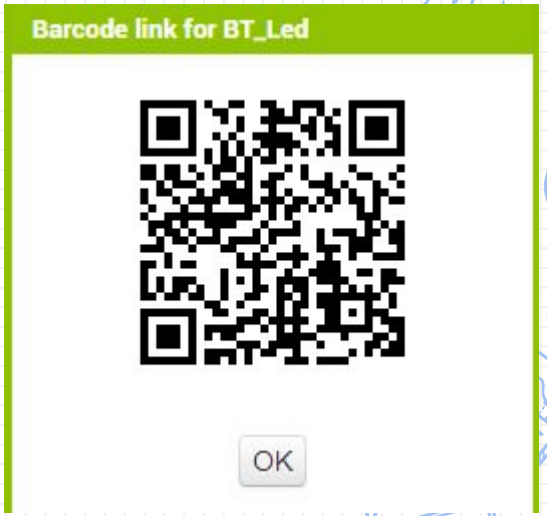
App 程式 初體驗

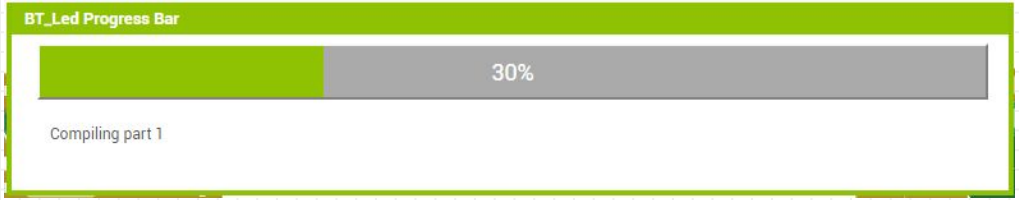

when 開燈按鈕 .Click
do call BluetoothClient1 .SendText
text " H "
set LED_Status . Text to " LED ON "
when 關燈按鈕 .Click
do call BluetoothClient1 .SendText
text " L "
set LED_Status . Text to " LED OFF "

App 程式下載

- 打包下載手機 App(.apk)

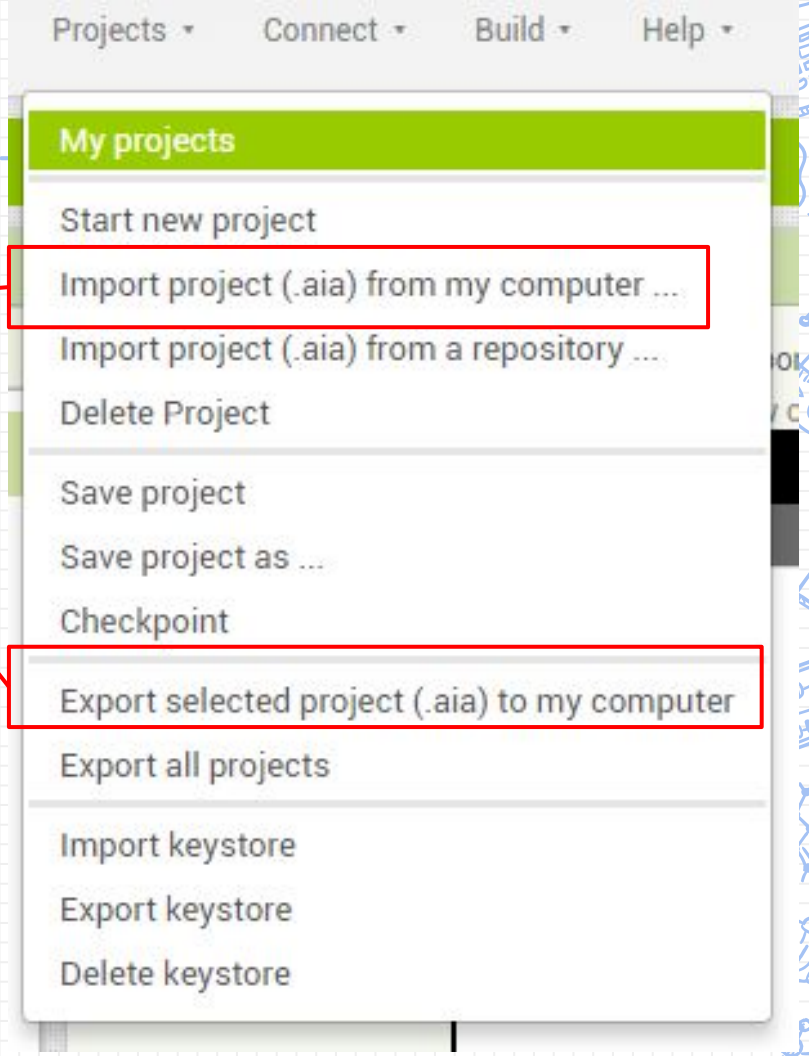






App 程式下載

- 儲存專案檔(.aia)
- 匯入專案檔(.aia)



Arduino 端程式

- Arduino 端程式
- LED_BTControl.xml

藍芽自走車

- Arduino 端程式
- Android App

Arduino 端程式

- 慢轉
- RobotCar_BTControl_SlowTurn.xml

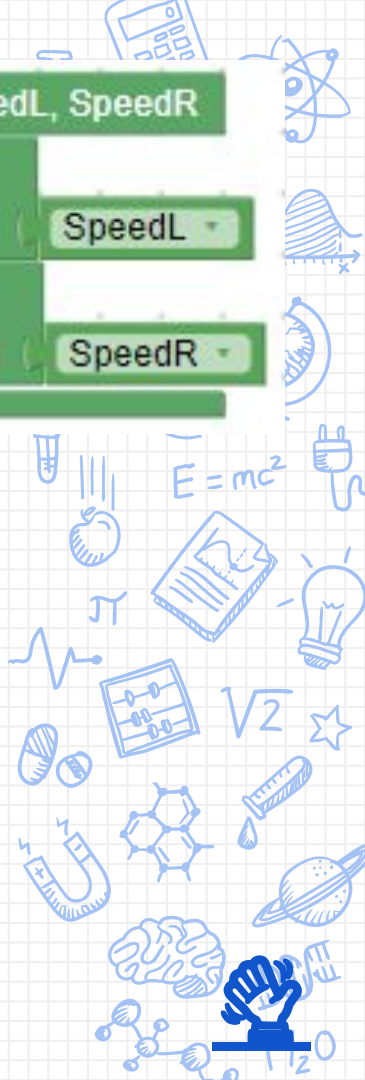
慢


設定
設定串列埠 serial 傳輸率 9600 bps
宣告 BTInput 當 char 資料 ' '
迴圈
如果 串列埠有效資料? > 0
執行 賦值 BTInput 到 串列埠輸入
如果 BTInput = 'F'
執行 呼叫 GO 與：
SpeedL 255
SpeedR 255
如果 BTInput = 'B'
執行 呼叫 GO 與：
SpeedL -255
SpeedR -255


如果 BTInput = 'L'
執行 呼叫 GO 與：
SpeedL 0
SpeedR 255
如果 BTInput = 'R'
執行 呼叫 GO 與：
SpeedL 255
SpeedR 0
如果 BTInput = 'S'
執行 呼叫 GO 與：
SpeedL 0
SpeedR 0

```
到 MotorL 與： speed
  如果 speed > 0
  執行
    設定數位腳位 2 為 高
    設定類比腳位 3 資料 255 - speed
  否則
    設定數位腳位 2 為 低
    設定類比腳位 3 資料 0 - speed
到 MotorR 與： speed
  如果 speed > 0
  執行
    設定數位腳位 4 為 高
    設定類比腳位 5 資料 255 - speed
  否則
    設定數位腳位 4 為 低
    設定類比腳位 5 資料 0 - speed
```

```
到 GO 與： SpeedL, SpeedR
  呼叫 MotorL 與：
    speed SpeedL
  呼叫 MotorR 與：
    speed SpeedR
```



Arduino 端程式

- 快轉
- RobotCar_BTControl_FastTurn.xml

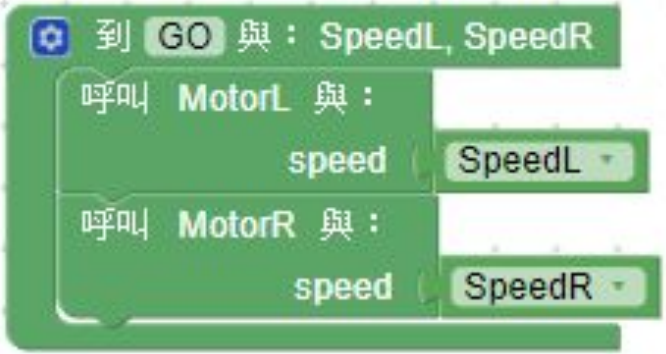
快轉(1)

❖


設定
設定串列埠 serial 傳輸率 9600 bps
宣告 BTInput 當 char 資料 ' '
迴圈
如果 串列埠有效資料? > 0
執行 賦值 BTInput 到 串列埠輸入
如果 BTInput = 'F'
執行 呼叫 GO 與：
SpeedL 255
SpeedR 255
如果 BTInput = 'B'
執行 呼叫 GO 與：
SpeedL -255
SpeedR -255


如果 BTInput = 'L'
執行 呼叫 GO 與：
SpeedL -255
SpeedR 255
如果 BTInput = 'R'
執行 呼叫 GO 與：
SpeedL 255
SpeedR -255
如果 BTInput = 'S'
執行 呼叫 GO 與：
SpeedL 0
SpeedR 0


如果 BTInput = '1'
執行 呼叫 GO 與：
SpeedL -150
SpeedR 150
如果 BTInput = '2'
執行 呼叫 GO 與：
SpeedL 150
SpeedR -150

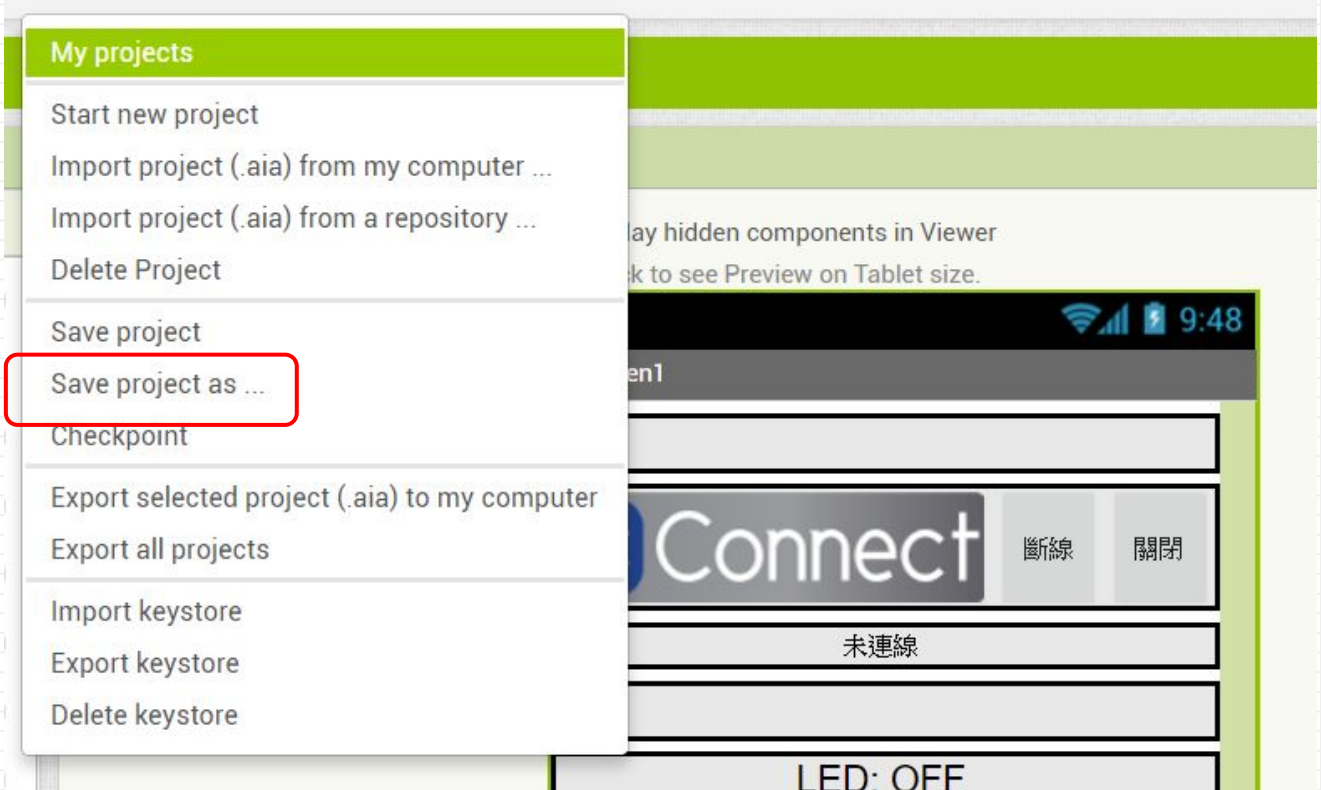

到 GO 與： SpeedL, SpeedR
呼叫 MotorL 與：
speed SpeedL
呼叫 MotorR 與：
speed SpeedR


到 MotorL 與： speed
如果 speed > 0
執行 設定數位腳位 2 為 高
設定類比腳位 3 資料 255 - speed
否則 設定數位腳位 2 為 低
設定類比腳位 3 資料 0 - speed


到 MotorR 與： speed
如果 speed > 0
執行 設定數位腳位 4 為 高
設定類比腳位 5 資料 255 - speed
否則 設定數位腳位 4 為 低
設定類比腳位 5 資料 0 - speed

Android App

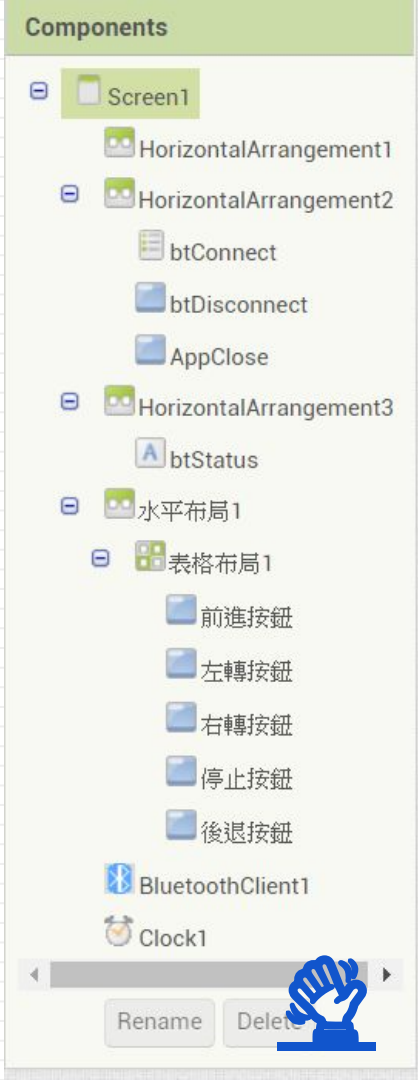
- App Inventor 2



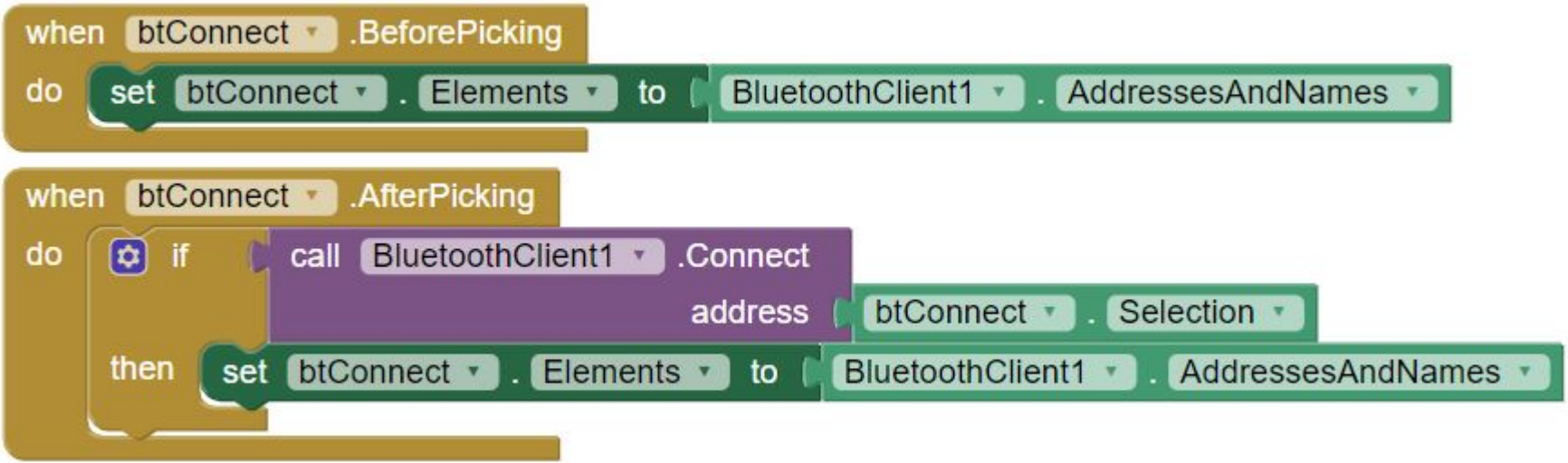
自走車控制 App

- BT_car.aia

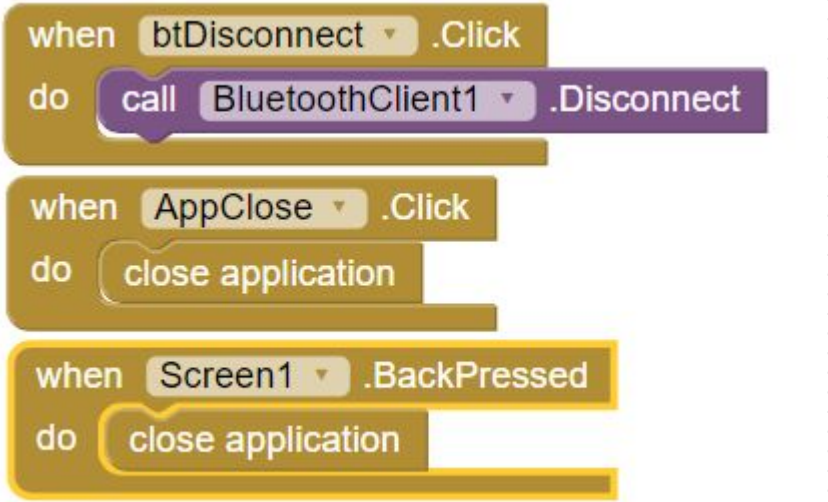
  BT_car.apk



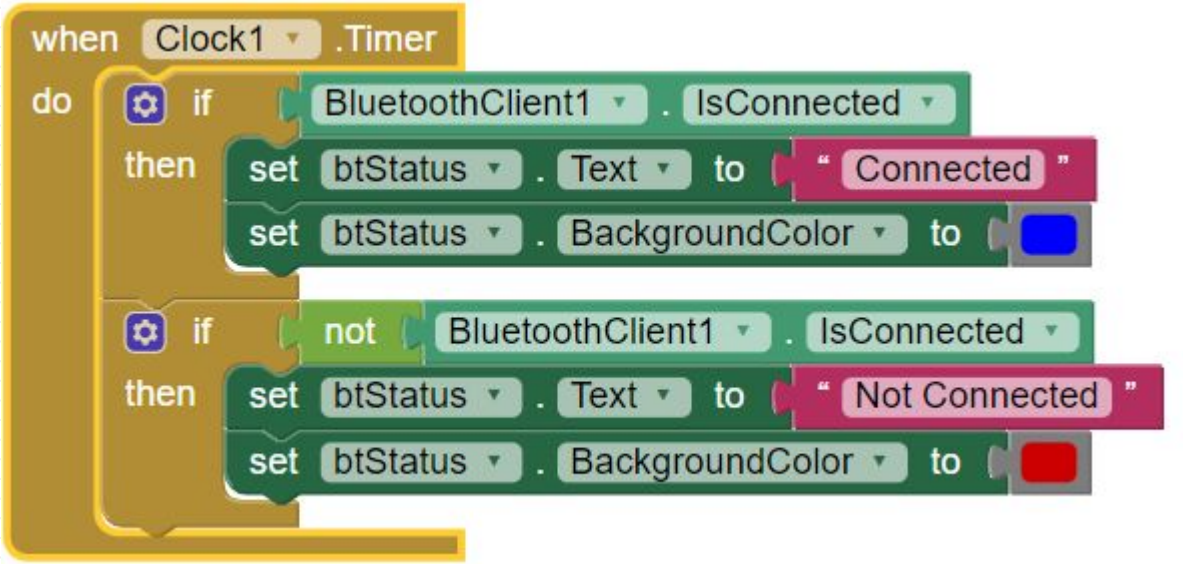
自走車控制 App


when btConnect .BeforePicking
do set btConnect . Elements to BluetoothClient1 . AddressesAndNames
when btConnect .AfterPicking
do if call BluetoothClient1 .Connect
address btConnect . Selection
then set btConnect . Elements to BluetoothClient1 . AddressesAndNames

自走車控制 App

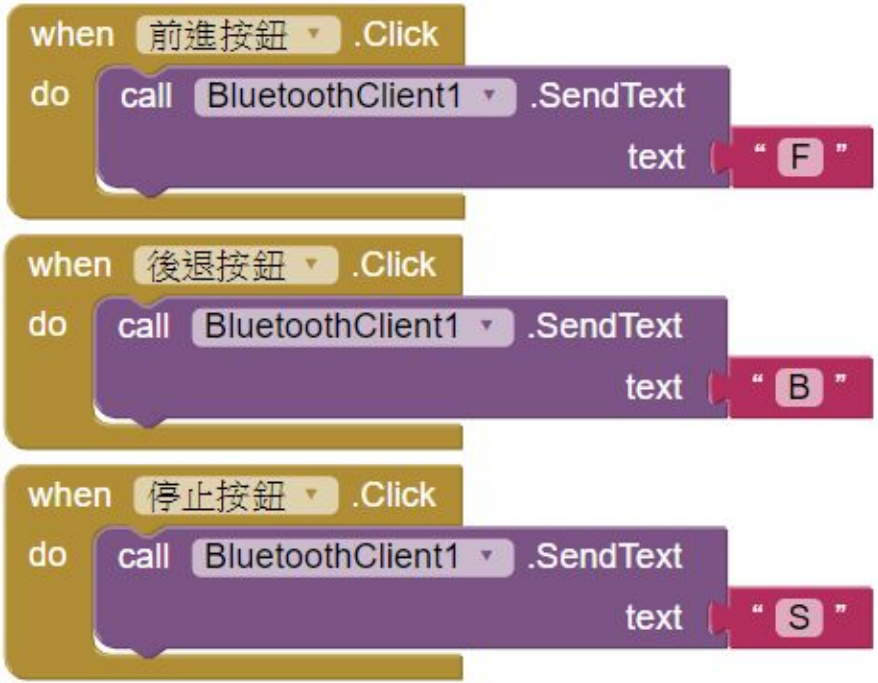

when btDisconnect .Click
do call BluetoothClient1 .Disconnect
when AppClose .Click
do close application
when Screen1 .BackPressed
do close application

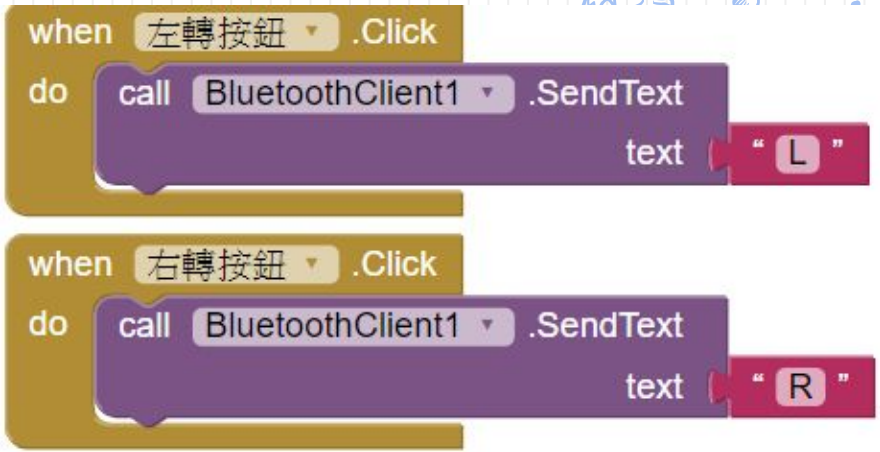
自走車控制 App


when Clock1 .Timer
do if BluetoothClient1 . IsConnected
then set btStatus . Text to " Connected "
set btStatus . BackgroundColor to
if not BluetoothClient1 . IsConnected
then set btStatus . Text to " Not Connected "
set btStatus . BackgroundColor to

自走車控制 App

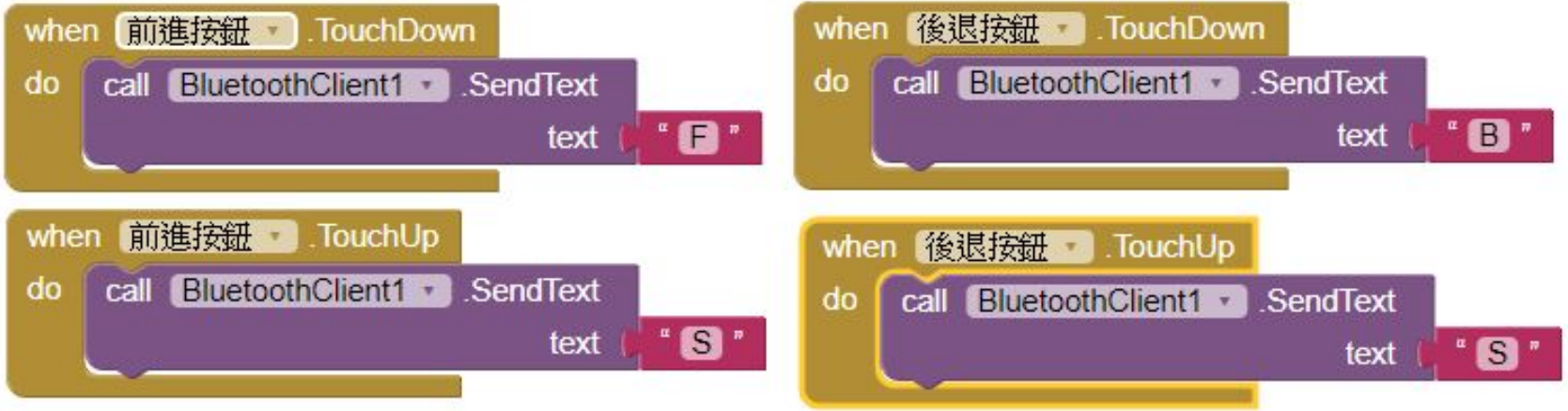
❖


when 前進按鈕 .Click
do call BluetoothClient1 .SendText
text " F "
when 後退按鈕 .Click
do call BluetoothClient1 .SendText
text " B "
when 停止按鈕 .Click
do call BluetoothClient1 .SendText
text " S "


when 左轉按鈕 .Click
do call BluetoothClient1 .SendText
text " L "
when 右轉按鈕 .Click
do call BluetoothClient1 .SendText
text " R "

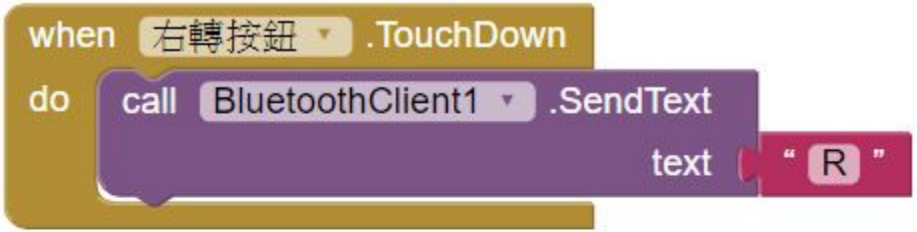
自走車控制 App

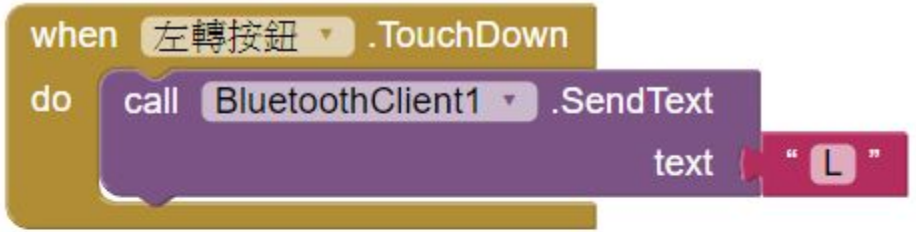
- 另一種控制方式

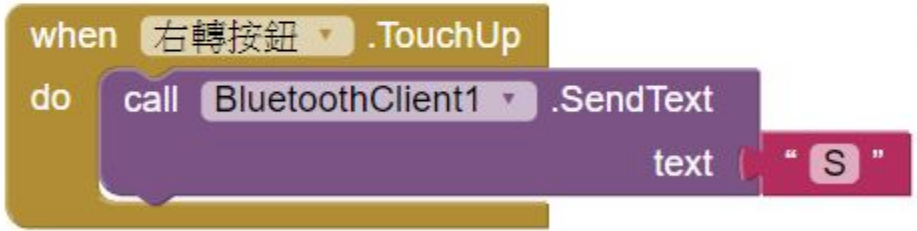


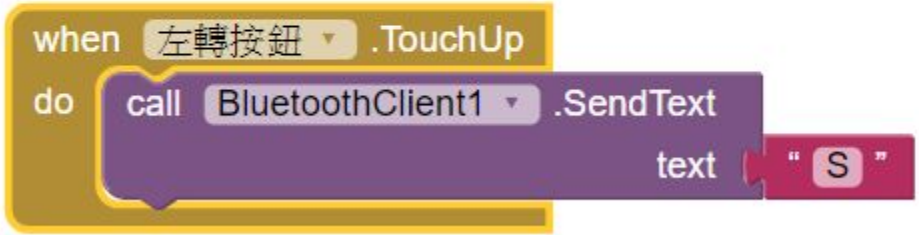
自走車控制 App

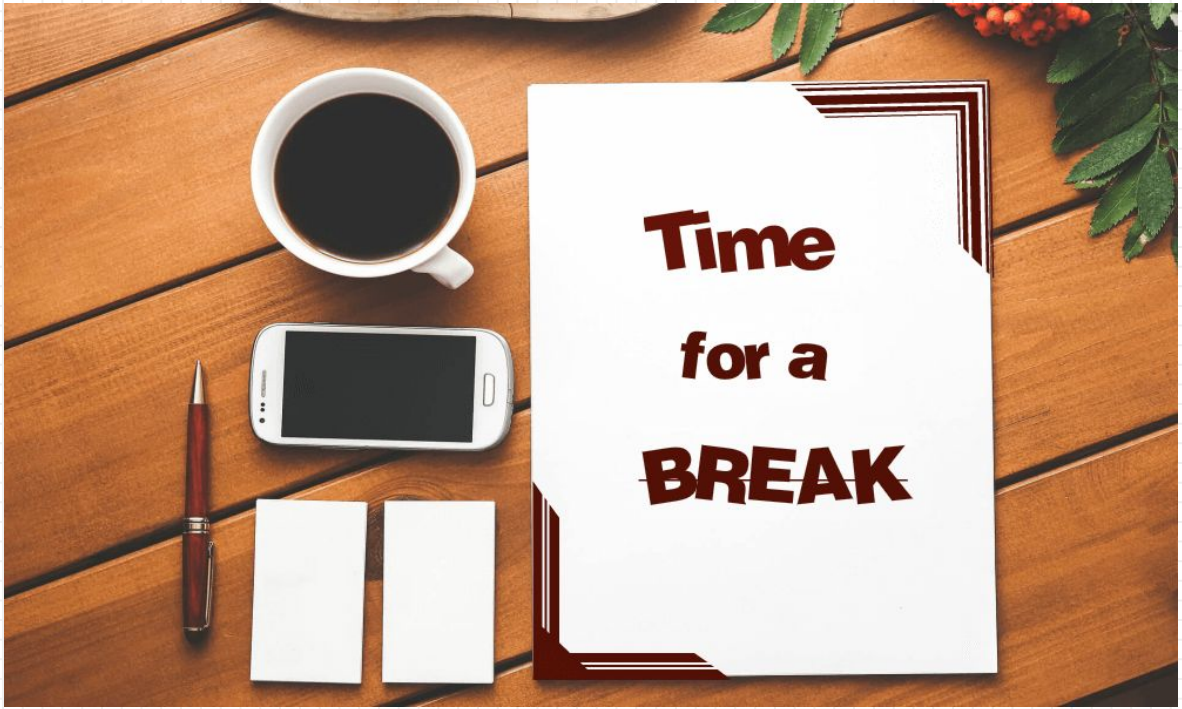
- 另一種控制方式









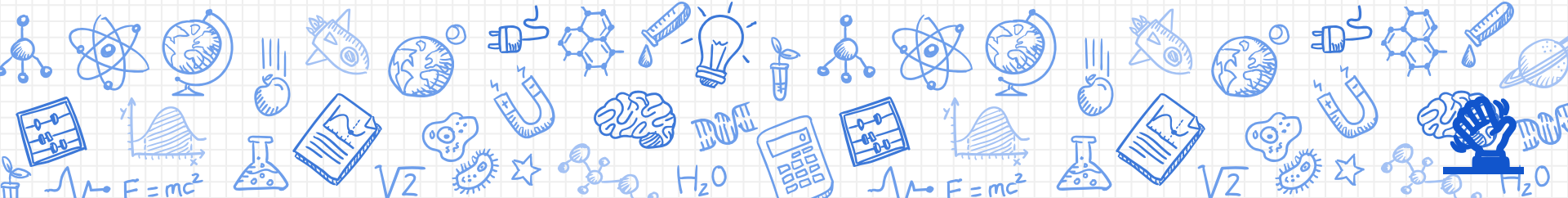

Time
for a
BREAK

交流

- 歡迎交流
- chyhchong@gmail.com

  0988087024

手機藍芽自走車



Arduino 端程式

- Arduino 端程式
- LED_BTControl.xml

App Inventor 2

- http://ai2.appinventor.mit.edu/
- Android App 視覺化程式設計

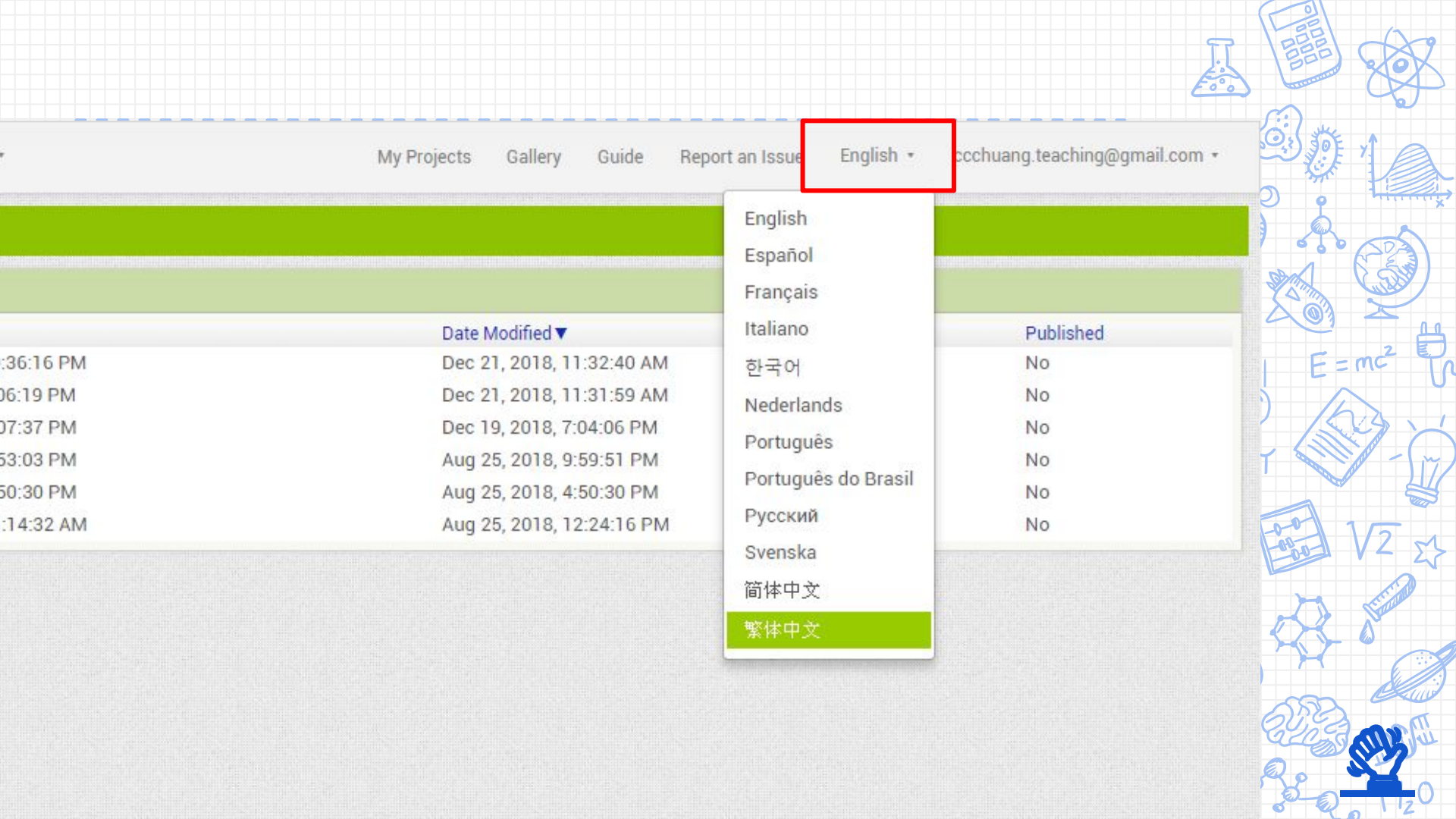
使用 Google 帳戶登入

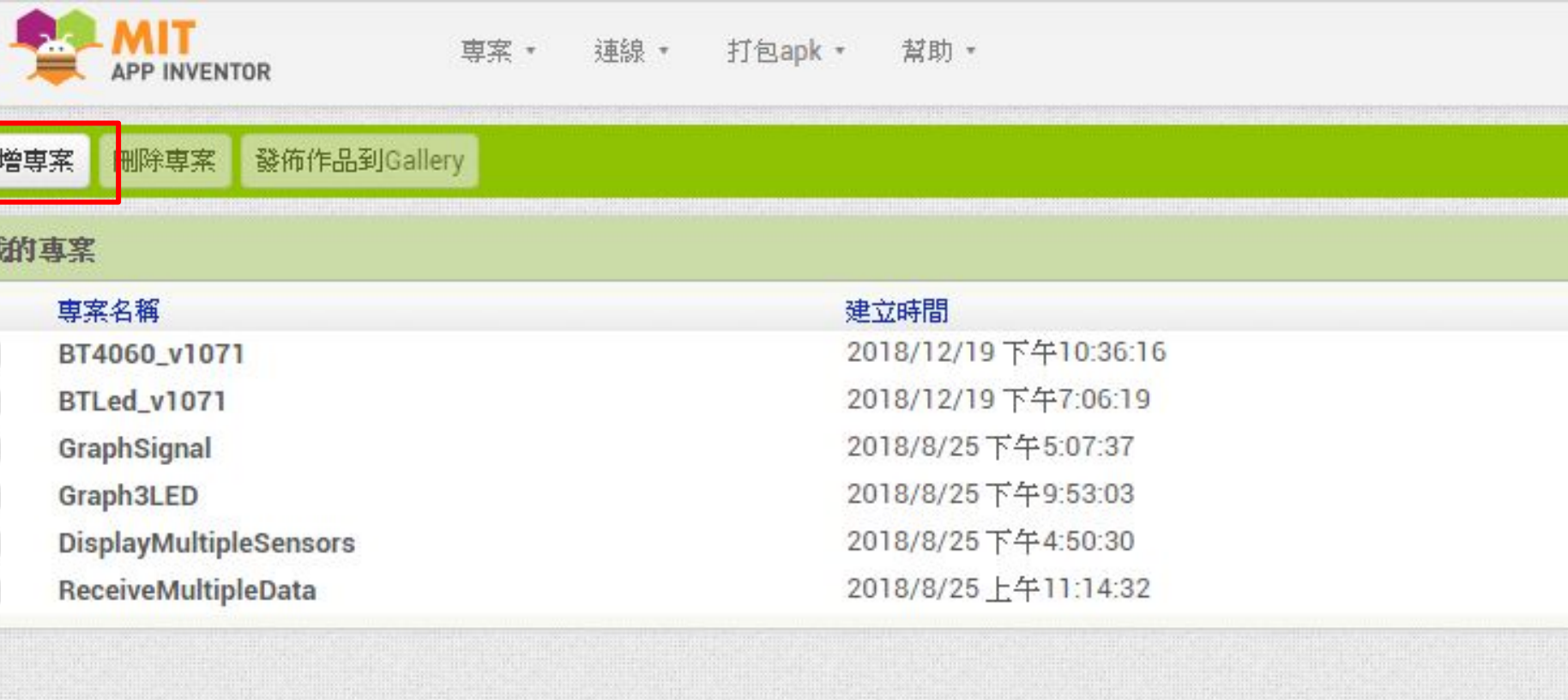
選擇帳戶

以繼續使用「mit.edu」

志忠	莊志忠 ccchuang@go.pymhs.tyc.edu.tw	未登入
c	chyh chong chyhchong@gmail.com	未登入
閔鈞	莊閔鈞 mc.mark.chuang@gmail.com	未登入
志忠	莊志忠 ccchuang.teaching@gmail.com	未登入

使用其他帳戶

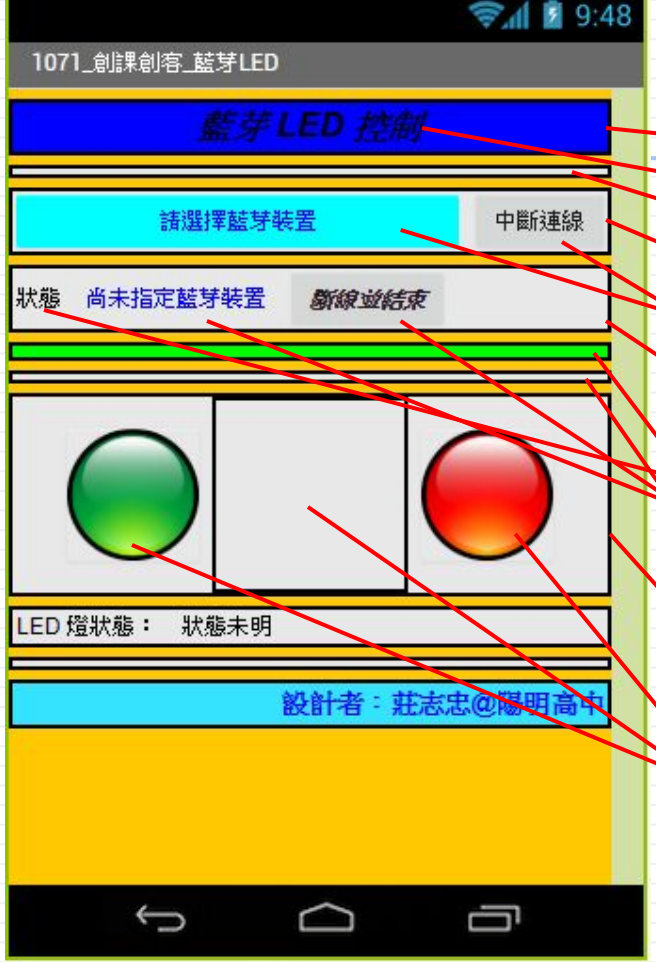

My Projects
Gallery
Guide
Report an Issue
English
cchuang.teaching@gmail.com
English
Español
Français
Italiano
한국어
Nederlands
Português
Português do Brasil
Русский
Svenska
简体中文
繁体中文
Date Modified
Published
:36:16 PM
Dec 21, 2018, 11:32:40 AM
No
06:19 PM
Dec 21, 2018, 11:31:59 AM
No
07:37 PM
Dec 19, 2018, 7:04:06 PM
No
53:03 PM
Aug 25, 2018, 9:59:51 PM
No
50:30 PM
Aug 25, 2018, 4:50:30 PM
No
:14:32 AM
Aug 25, 2018, 12:24:16 PM
No

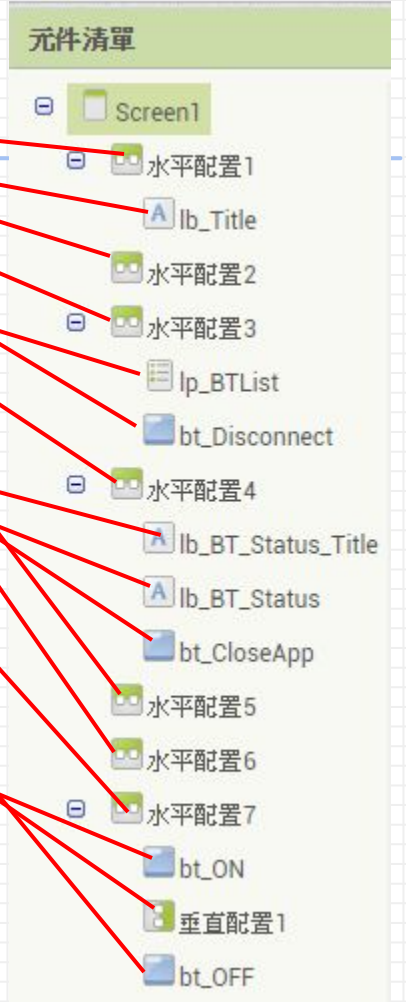

MIT
APP INVENTOR
專案
連線
打包apk
幫助
新增專案
刪除專案
發佈作品到Gallery
我的專案
專案名稱
建立時間
BT4060_v1071
2018/12/19 下午10:36:16
BTLed_v1071
2018/12/19 下午7:06:19
GraphSignal
2018/8/25 下午5:07:37
Graph3LED
2018/8/25 下午9:53:03
DisplayMultipleSensors
2018/8/25 下午4:50:30
ReceiveMultipleData
2018/8/25 上午11:14:32


新增專案...
專案名稱:
BTLed_1071
取消
確定

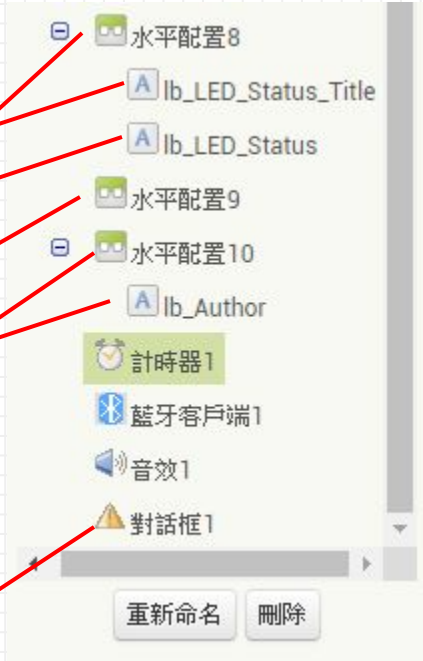

MIT APP INVENTOR
專案
連線
打包apk
幫助
我的專案
Gallery
指南
回報問題
繁體中文
ccchuang.teaching@gmail.com
BTLed_1072
Screen1
新增螢幕
刪除螢幕
畫面編排
程式設計
元件面板
使用者介面
按鈕
複選盒
日期選擇器
圖像
標籤
清單選擇器
清單顯示器
對話框
密碼輸入盒
滑桿
下拉式選單
文字輸入盒
物件區
工作面板
顯示隱藏元件
以平板電腦尺寸預覽
9:48
Screen1
介面設計區
元件清單
Screen1
組成元件區
元件屬性
Screen1
應用說明
突顯顏色
預設
屬性區
水平對齊
靠左：1
垂直對齊
靠上：1
App名稱
BTLed_1072
背景顏色
預設
背景圖片
無...
關閉螢幕動畫
預設效果


MIT APP INVENTOR
專案
連線
打包apk
幫助
我的專案
Gallery
指南
回報問題
繁體中文
ccchuang.teaching@gmail.com
BTLed_1072
Screen1
新增螢幕
刪除螢幕
畫面編排
程式設計
方塊
內件方塊
流程控制
邏輯
數學
文字
清單
顏色
變數
程序
Screen1
任意元件
物件
工作面板
視覺化程式設計
0
0
顯示警告


9:48
1071_創課創客_藍芽LED
藍芽 LED 控制
請選擇藍芽裝置
中斷連線
狀態
尚未指定藍芽裝置
斷線並結束
LED 燈狀態：
狀態未明
設計者：莊志忠@陽明高中
Non-visible components
計時器1
藍牙客戶端1
音效1
對話框1


元件清單
Screen1
水平配置1
lb_Title
水平配置2
水平配置3
lp_BTList
bt_Disconnect
水平配置4
lb_BT_Status_Title
lb_BT_Status
bt_CloseApp
水平配置5
水平配置6
水平配置7
bt_ON
垂直配置1
bt_OFF


9:48
1071_創課創客_藍芽LED
藍芽 LED 控制
請選擇藍芽裝置
中斷連線
狀態 尚未指定藍芽裝置
斷線並結束
LED燈狀態： 狀態未明
設計者：莊志忠@陽明高中
Non-visible components
計時器1
藍牙客戶端1
音效1
對話框1


水平配置8
lb_LED_Status_Title
lb_LED_Status
水平配置9
水平配置10
lb_Author
計時器1
藍牙客戶端1
音效1
對話框1
重新命名
刪除

素材

off.png

on.png

led.jpg

上傳文件...

元件屬性

Screen1

應用說明

突顯顏色
預設

水平對齊
靠左 : 1

垂直對齊
靠上 : 1

App名稱
BTLed_v1071

背景顏色
橙色

背景圖片
無...

關閉螢幕動畫
預設效果

圖示
led.jpg...

開啟螢幕動畫
預設效果

基本色

基本色
預設

PrimaryColorDark
預設

螢幕方向
未指定方向

允許捲動

以JSON格式顯示清單

狀態欄顯示

視窗大小
固定大小

Theme
Classic

標題
1071_創課創客_藍芽LED

標題顯示

TutorialURL

版本編號
1

版本編號
1

版本名稱
1.0

元件屬性

水平配置1

水平對齊
靠左 : 1

垂直對齊
置中 : 2

背景顏色
預設

高度
自動...

寬度
填滿...

圖像
無...

可見性


元件屬性
lb_Title
背景顏色
藍色
粗體
斜體
字體大小
20.0
字形
預設字體
HTML格式
具有外邊距
高度
自動...
寬度
填滿...
文字
藍芽 LED 控制
文字對齊
置中 : 1
文字對齊
置中 : 1
文字顏色
預設
可見性
元件屬性
水平配置2
水平對齊
靠左 : 1
垂直對齊
靠上 : 1
背景顏色
預設
高度
3像素...
寬度
填滿...
圖像
無...
可見性
元件屬性
水平配置3
水平對齊
靠左 : 1
垂直對齊
靠上 : 1
背景顏色
預設
高度
自動...
寬度
填滿...
圖像
無...
可見性
元件屬性
lp_BTList
背景顏色
青色
元素字串
啟用
粗體
斜體
字體大小
14.0
字形
預設字體
高度
自動...
寬度
填滿...
寬度
填滿...
圖像
無...
項目背景顏色
預設
項目文字顏色
預設
選中項
形狀
預設
顯示互動效果
顯示搜尋框
文字
請選擇藍芽裝置
文字對齊
置中 : 1
文字顏色
藍色
標題
標題
可見性

元件屬性

bt_Disconnect

背景顏色
預設

啟用

粗體

斜體

字體大小
14.0

字形
預設字體

高度
自動...

寬度
自動...

圖像
無...

形狀
預設

顯示互動效果

顯示互動效果

文字
中斷連線

文字對齊
置中 : 1

文字顏色
預設

可見性

元件屬性

水平配置4

水平對齊
靠左 : 1

垂直對齊
置中 : 2

背景顏色
預設

高度
自動...

寬度
填滿...

圖像
無...

可見性

元件屬性

lb_BT_Status_Title

背景顏色
透明

粗體

斜體

字體大小
14.0

字形
預設字體

HTML格式

具有外邊距

高度
自動...

寬度
自動...

文字
狀態

文字對齊
靠左 : 0

文字顏色
預設

可見性

元件屬性

lb_BT_Status

背景顏色
透明

粗體

斜體

字體大小
14.0

字形
預設字體

HTML格式

具有外邊距

高度
自動...

寬度
自動...

文字
尚未指定藍芽裝置

文字對齊
靠左 : 0

文字顏色
藍色

可見性

元件屬性

bt_CloseApp

背景顏色
預設

啟用

粗體

斜體

字體大小
14.0

字形
預設字體

高度
自動...

寬度
自動...

圖像
無...

形狀
預設

顯示互動效果

文字
斷線並結束

文字對齊
置中 : 1

文字顏色
預設

可見性

元件屬性

水平配置5

水平對齊
靠左 : 1

垂直對齊
靠上 : 1

背景顏色
綠色

高度
6像素...

寬度
填滿...

圖像
無...

可見性

元件屬性

水平配置6

水平對齊
靠左 : 1

垂直對齊
靠上 : 1

背景顏色
預設

高度
3像素...

寬度
填滿...

圖像
無...

可見性

元件屬性

水平配置7

水平對齊
置中 : 3

垂直對齊
置中 : 2

背景顏色
預設

高度
自動...

寬度
填滿...

圖像
無...

可見性

元件屬性

bt_ON

背景顏色
預設

啟用

粗體

斜體

字體大小
14.0

字形
預設字體

高度
70像素...

寬度
70像素...

圖像
on.png...

形狀
預設

顯示互動效果

文字

文字對齊
置中 : 1

文字顏色
預設

可見性

元件屬性

垂直配置1

水平對齊
靠左 : 1

垂直對齊
靠上 : 1

背景顏色
預設

高度
自動...

寬度
自動...

圖像
無...

可見性

元件屬性

bt_OFF

背景顏色
預設

啟用

粗體

斜體

字體大小
14.0

字形
預設字體

高度
70像素...

寬度
70像素...

圖像
off.png...

形狀
預設

顯示互動效果

文字

文字對齊
置中 : 1

文字顏色
預設

可見性

元件屬性

水平配置8

水平對齊
靠左 : 1

垂直對齊
置中 : 2

背景顏色
預設

高度
自動...

寬度
填滿...

圖像
無...

可見性

元件屬性

lb_LED_Status_Title

背景顏色
透明

粗體

斜體

字體大小
14.0

字形
預設字體

HTML格式

具有外邊距

高度
自動...

寬度
自動...

文字
LED 燈狀態：

文字對齊
靠左 : 0

文字顏色
預設

可見性


元件屬性
lb_LED_Status
背景顏色
透明
粗體
斜體
字體大小
14.0
字形
預設字體
HTML格式
具有外邊距
高度
自動...
寬度
自動...
文字
狀態未明
文字對齊
靠左 : 0
文字顏色
預設
可見性
元件屬性
水平配置9
水平對齊
靠左 : 1
垂直對齊
靠上 : 1
背景顏色
預設
高度
3像素...
寬度
填滿...
圖像
無...
可見性
元件屬性
水平配置10
水平對齊
靠右 : 2
垂直對齊
置中 : 2
背景顏色
Custom...
高度
自動...
寬度
填滿...
圖像
無...
可見性
元件屬性
lb_Author
背景顏色
透明
粗體
斜體
字體大小
16.0
字形
預設字體
HTML格式
具有外邊距
高度
自動...
寬度
自動...
文字
設計者：莊志忠@陽明高中
文字對齊
靠右 : 2
文字顏色
藍色
可見性

元件屬性

計時器1

持續計時

啟用計時

計時間隔

1000

元件屬性

藍牙客戶端1

字元編碼

UTF-8

分隔符號位元組碼

0

高位元優先

啟用安全連線

元件屬性

音效1

最小間隔（ms）

500

來源

無...

元件屬性

對話框1

背景顏色

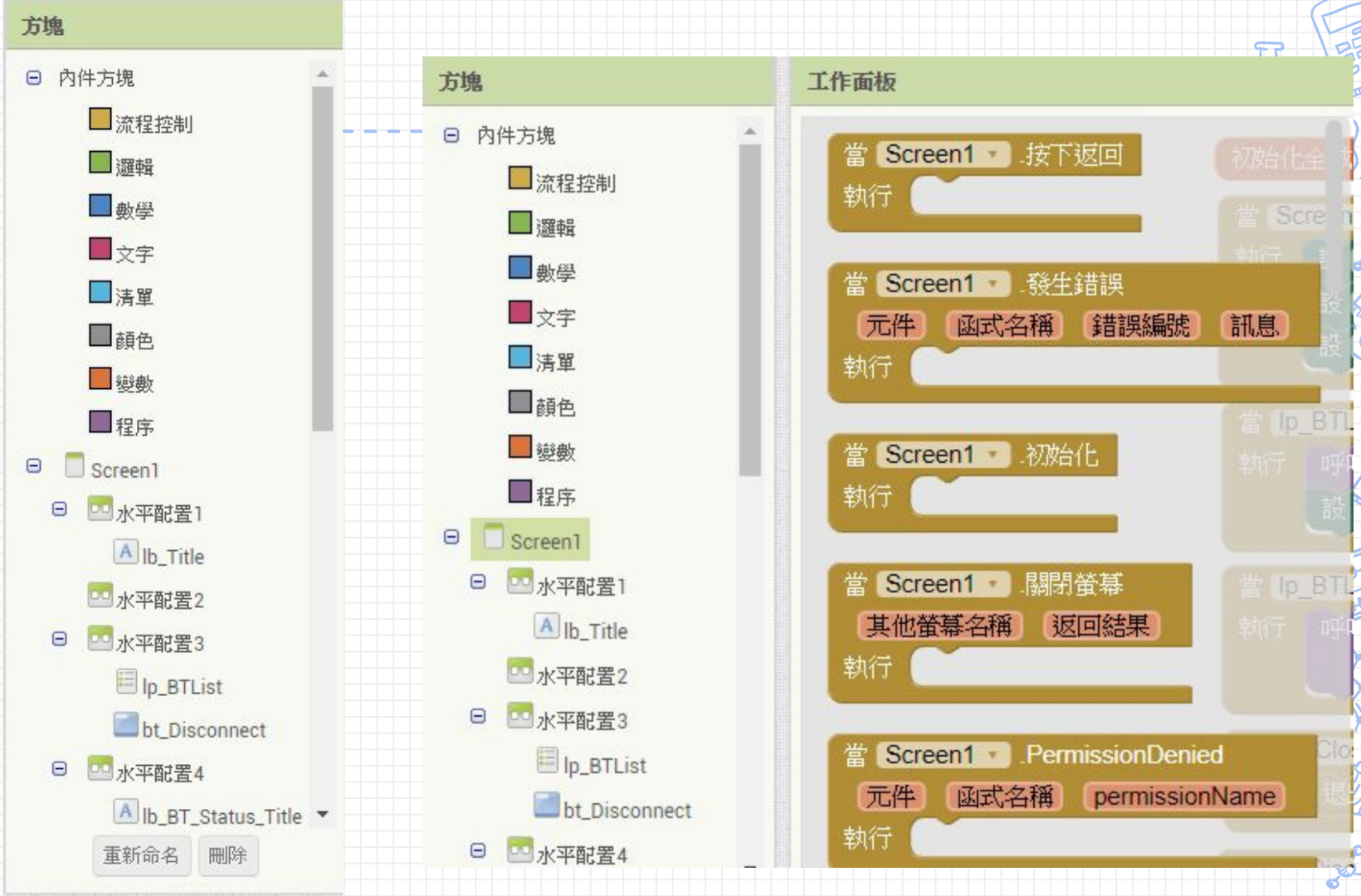
預設

顯示時間長度

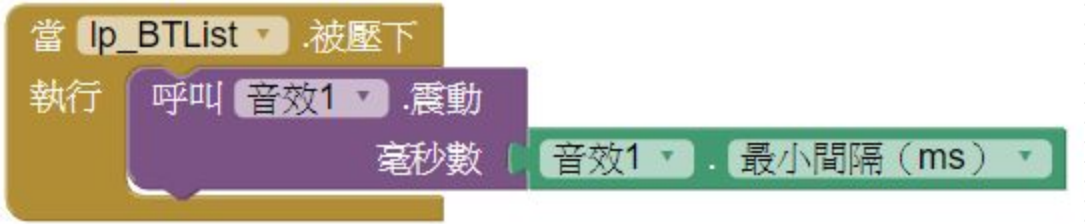
顯示時間長

文字顏色

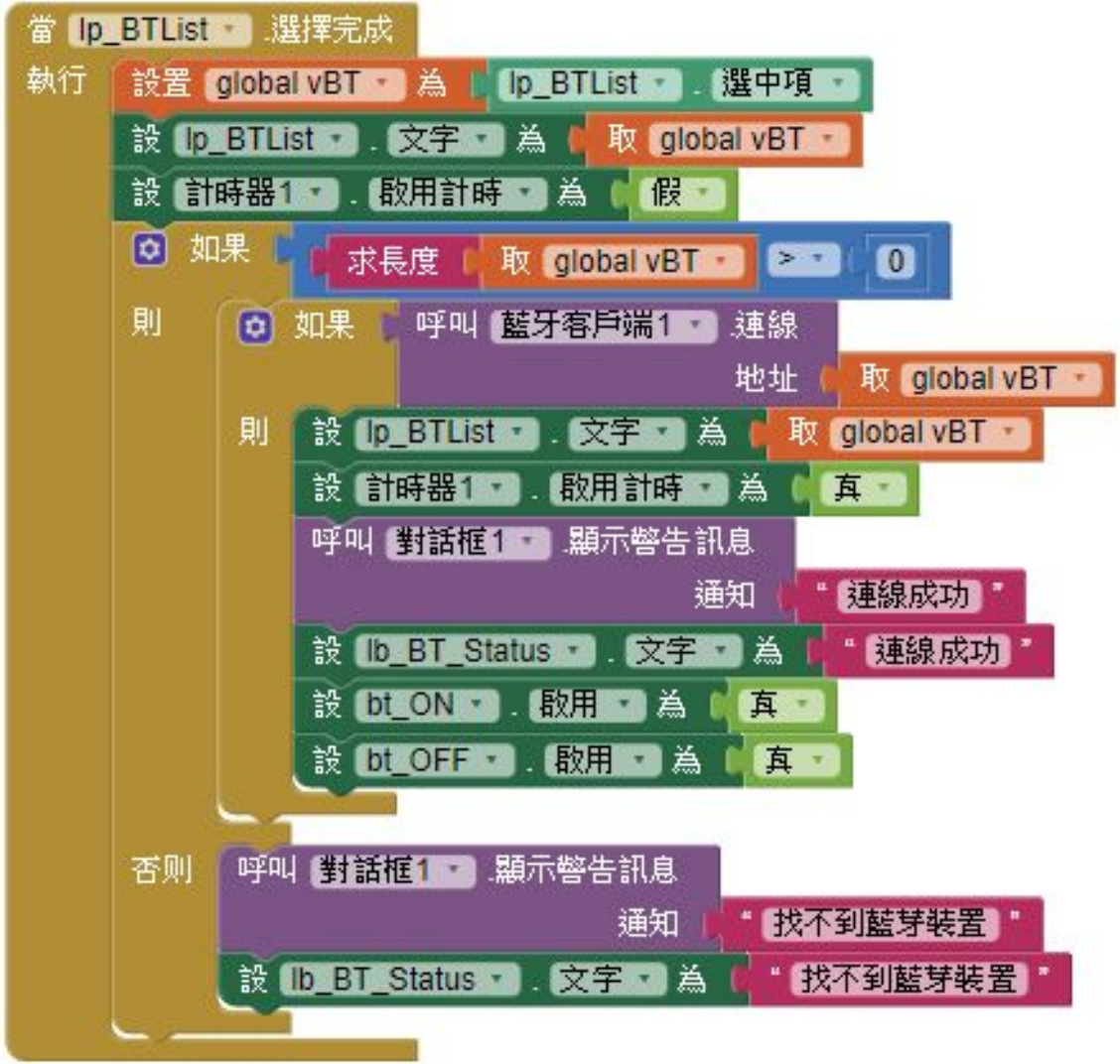
預設


方塊
內件方塊
流程控制
邏輯
數學
文字
清單
顏色
變數
程序
Screen1
水平配置1
lb_Title
水平配置2
水平配置3
lp_BTList
bt_Disconnect
水平配置4
lb_BT_Status_Title
重新命名
刪除
方塊
內件方塊
流程控制
邏輯
數學
文字
清單
顏色
變數
程序
Screen1
水平配置1
lb_Title
水平配置2
水平配置3
lp_BTList
bt_Disconnect
水平配置4
工作面板
當 Screen1 .按下返回
執行
當 Screen1 .發生錯誤
元件 函式名稱 錯誤編號 訊息
執行
當 Screen1 .初始化
執行
當 Screen1 .關閉螢幕
其他螢幕名稱 返回結果
執行
當 Screen1 .PermissionDenied
元件 函式名稱 permissionName
執行


初始化全域變數 vBT 為 " "
當 Screen1 .初始化
執行
設 計時器1 . 啟用計時 為 假
設 bt_ON . 啟用 為 假
設 bt_OFF . 啟用 為 假


當 lp_BTList .被壓下
執行
呼叫 音效1 .震動
毫秒數 音效1 . 最小間隔（ms）

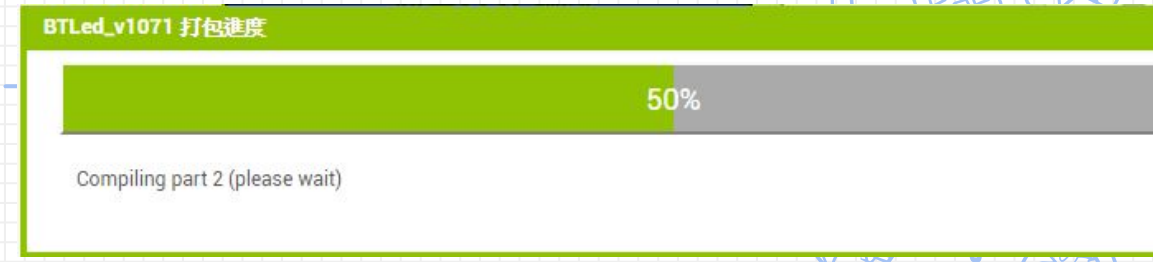

當 lp_BTList .準備選擇
執行
呼叫 藍牙客戶端1 .斷開連線
設 lp_BTList . 元素 為 藍牙客戶端1 . 地址及名稱


當 lp_BTList 選擇完成
執行
設置 global vBT 為 lp_BTList . 選中項
設 lp_BTList . 文字 為 取 global vBT
設 計時器1 . 啟用計時 為 假
如果 求長度 取 global vBT > 0
則
如果 呼叫 藍牙客戶端1 連線
地址 取 global vBT
則
設 lp_BTList . 文字 為 取 global vBT
設 計時器1 . 啟用計時 為 真
呼叫 對話框1 顯示警告訊息
通知 " 連線成功 "
設 lb_BT_Status . 文字 為 " 連線成功 "
設 bt_ON . 啟用 為 真
設 bt_OFF . 啟用 為 真
否則
呼叫 對話框1 顯示警告訊息
通知 " 找不到藍芽裝置 "
設 lb_BT_Status . 文字 為 " 找不到藍芽裝置 "


當 bt_ON .被點選
執行
呼叫 音效1 .震動
毫秒數 音效1 . 最小間隔 (ms)
設 lb_LED_Status . 文字 為 " 開啟 "
呼叫 藍牙客戶端1 .發送文字
文字 " H "
當 bt_OFF .被點選
執行
呼叫 音效1 .震動
毫秒數 音效1 . 最小間隔 (ms)
設 lb_LED_Status . 文字 為 " 關閉 "
呼叫 藍牙客戶端1 .發送文字
文字 " L "


當 bt_CloseApp .被點選
執行 退出程式
當 bt_Disconnect .被點選
執行 呼叫 音效1 .震動
毫秒數 音效1 . 最小間隔（ms）
設 計時器1 . 啟用計時 為 假
呼叫 對話框1 .顯示警告訊息
通知 " 使用者斷線 "
呼叫 藍牙客戶端1 .斷開連線
設 lb_BT_Status . 文字 為 " 使用者斷線 "
設 lp_BTList . 文字 為 " 請選擇藍芽裝置 "
設 lb_LED_Status . 文字 為 " 狀態未明 "

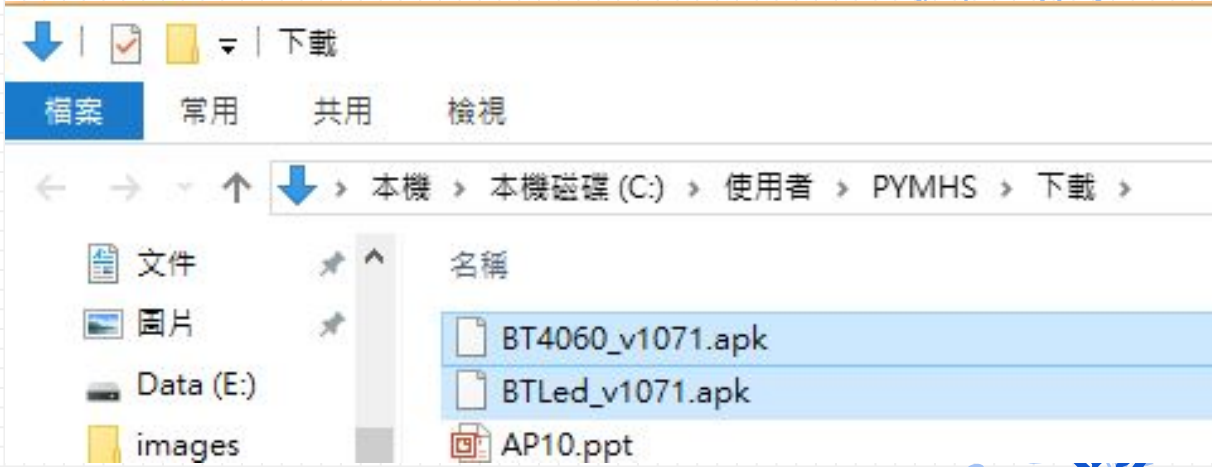
打包apk - QR Code



打包apk - 存為 apk


打包apk
幫助
打包apk並顯示二維條碼
打包apk並下載到電腦


BTLed_v1071 打包進度
50%
Compiling part 2 (please wait)


下載
檔案
常用
共用
檢視
本機 › 本機磁碟 (C:) › 使用者 › PYMHS › 下載 ›
文件
圖片
Data (E:)
images
名稱
BT4060_v1071.apk
BTLed_v1071.apk
AP10.ppt

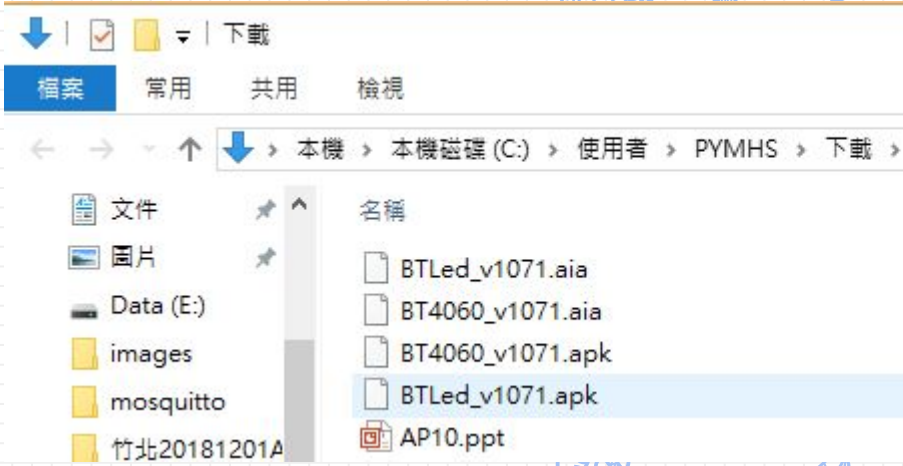
另存專案


專案
連線
我的專案
新增專案
匯入專案(.aia)
匯入範例專案(.aia)
刪除專案
儲存專案
另存專案
檢查點
導出專案(.aia)
導出所有專案
上傳金鑰
下載金鑰
刪除金鑰


專案BTLed_v1071另存為
新名稱:
BTCar
取消
確定

專案導出


專案
連線
我的專案
新增專案
匯入專案(.aia)
匯入範例專案(.aia)
刪除專案
儲存專案
另存專案
檢查點
導出專案(.aia)
導出所有專案
上傳金鑰
下載金鑰
刪除金鑰

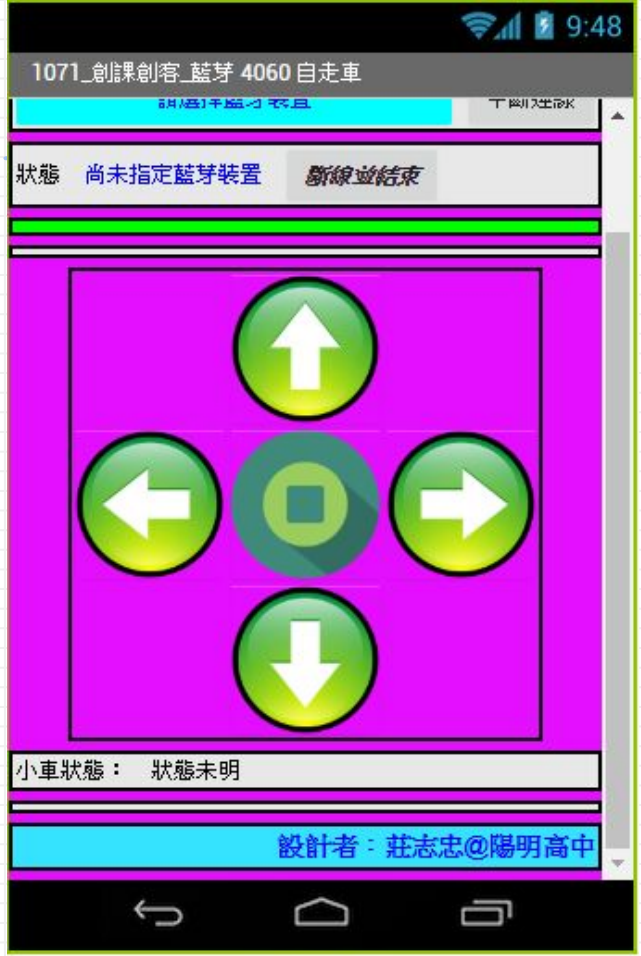

下載
檔案
常用
共用
檢視
本機 › 本機磁碟 (C:) › 使用者 › PYMHS › 下載 ›
文件
圖片
Data (E:)
images
mosquitto
竹北20181201A
名稱
BTLed_v1071.aia
BT4060_v1071.aia
BT4060_v1071.apk
BTLed_v1071.apk
AP10.ppt

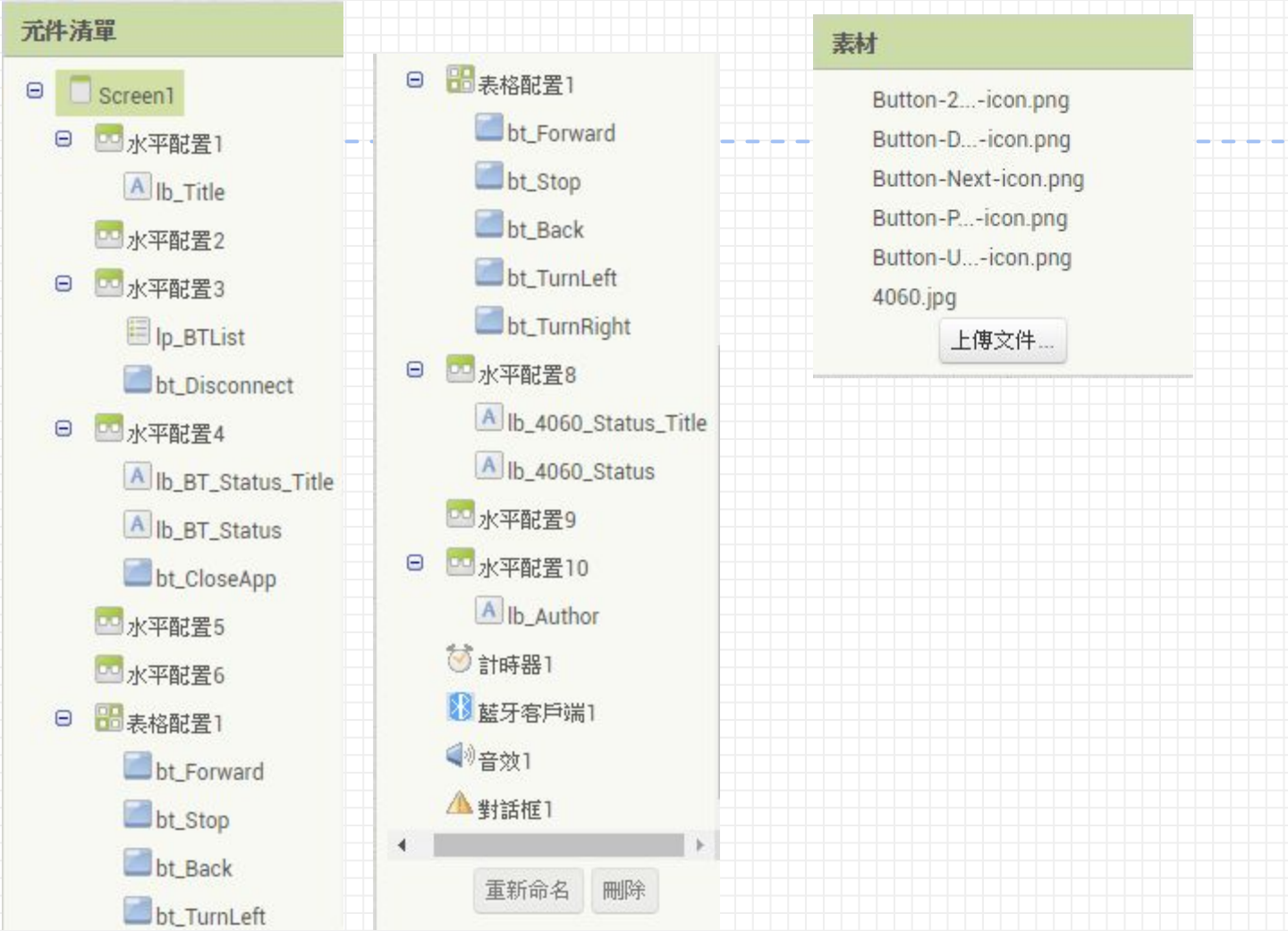
專案導入

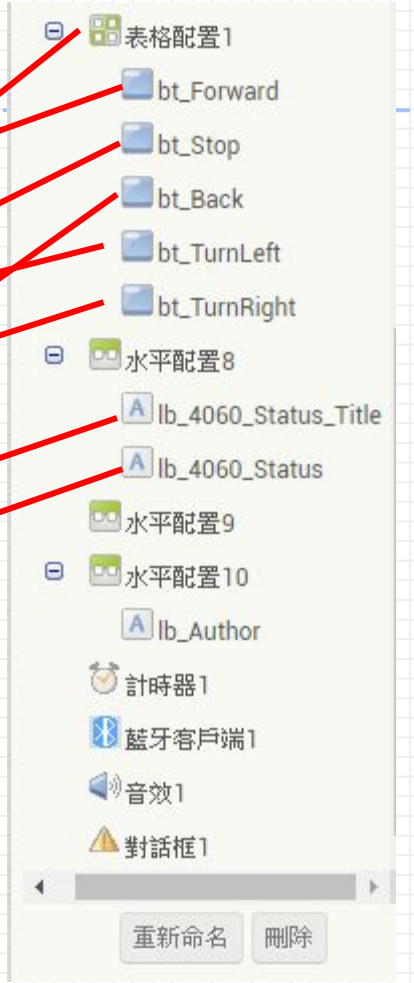

專案
連線
我的專案
新增專案
匯入專案(.aia)
匯入範例專案(.aia)
刪除專案
儲存專案
另存專案
檢查點
導出專案(.aia)
導出所有專案
上傳金鑰


上傳專案...
選擇檔案
未選擇任何檔案
取消
確定


9:48
1071_創課創客_藍芽 4060 自走車
藍芽 4060 小車控制
請選擇藍芽裝置
中斷連線
狀態
尚未指定藍芽裝置
斷線並結束
非可視元件
計時器1
藍牙客戶端1
音效1
對話框1


9:48
1071_創課創客_藍芽 4060 自走車
狀態
尚未指定藍芽裝置
斷線並結束
小車狀態：
狀態未明
設計者：莊志忠@陽明高中
非可視元件
計時器1
藍牙客戶端1
音效1
對話框1


元件清單
Screen1
水平配置1
lb_Title
水平配置2
水平配置3
lp_BTList
bt_Disconnect
水平配置4
lb_BT_Status_Title
lb_BT_Status
bt_CloseApp
水平配置5
水平配置6
表格配置1
bt_Forward
bt_Stop
bt_Back
bt_TurnLeft
表格配置1
bt_Forward
bt_Stop
bt_Back
bt_TurnLeft
水平配置8
lb_4060_Status_Title
lb_4060_Status
水平配置9
水平配置10
lb_Author
計時器1
藍牙客戶端1
音效1
對話框1
重新命名
刪除
素材
Button-2...-icon.png
Button-D...-icon.png
Button-Next-icon.png
Button-P...-icon.png
Button-U...-icon.png
4060.jpg
上傳文件...



非可視元件

計時器1 藍牙客戶端1 音效1 對話框1

元件屬性

Screen1

應用說明

突顯顏色
預設

水平對齊
置中 : 3

垂直對齊
靠上 : 1

App名稱
BT4060_v1071

背景顏色
Custom...

背景圖片
無...

關閉螢幕動畫
預設效果

圖示
4060.jpg...

開啟螢幕動畫
預設效果

基本色
預設

基本色
預設

PrimaryColorDark
預設

螢幕方向
未指定方向

允許捲動

以JSON格式顯示清單

狀態欄顯示

視窗大小
固定大小

Theme
Classic

標題
1071_創課創客_藍芽 4060

標題顯示

TutorialURL

版本編號
1

元件屬性

表格配置1

列數
3

高度
自動...

寬度
自動...

行數
3

可見性

元件屬性

bt_Forward

背景顏色
預設

啟用

粗體

斜體

字體大小
14.0

字形
預設字體

高度
80像素...

寬度
80像素...

圖像
Button-Upload-icon.png...

形狀
預設

顯示互動效果

顯示互動效果

文字

文字對齊
置中 : 1

文字顏色
預設

可見性

元件屬性

bt_Stop

背景顏色
預設

啟用

粗體

斜體

字體大小
14.0

字形
預設字體

高度
80像素...

寬度
80像素...

圖像
Button-2-stop-icon.png...

形狀
預設

顯示互動效果

文字

文字對齊
置中 : 1

文字顏色
預設

可見性


元件屬性
bt_Back
背景顏色
預設
啟用
粗體
斜體
字體大小
14.0
字形
預設字體
高度
80像素...
寬度
80像素...
圖像
Button-Download-icon.png.
形狀
預設
顯示互動效果
文字
文字對齊
置中 : 1
文字顏色
預設
可見性
元件屬性
bt_TurnLeft
背景顏色
預設
啟用
粗體
斜體
字體大小
14.0
字形
預設字體
高度
80像素...
寬度
80像素...
圖像
Button-Previous-icon.png...
形狀
預設
顯示互動效果
文字
文字對齊
置中 : 1
文字顏色
預設
可見性
元件屬性
bt_TurnRight
背景顏色
預設
啟用
粗體
斜體
字體大小
14.0
字形
預設字體
高度
80像素...
寬度
80像素...
圖像
Button-Next-icon.png...
形狀
預設
顯示互動效果
文字
文字對齊
置中 : 1
文字顏色
預設
可見性

元件屬性

lb_4060_Status_Title

背景顏色
透明

粗體

斜體

字體大小
14.0

字形
預設字體

HTML格式

具有外邊距

高度
自動...

寬度
自動...

文字
小車狀態：

文字對齊
靠左 : 0

文字顏色
預設

可見性

元件屬性

lb_4060_Status

背景顏色
透明

粗體

斜體

字體大小
14.0

字形
預設字體

HTML格式

具有外邊距

高度
自動...

寬度
自動...

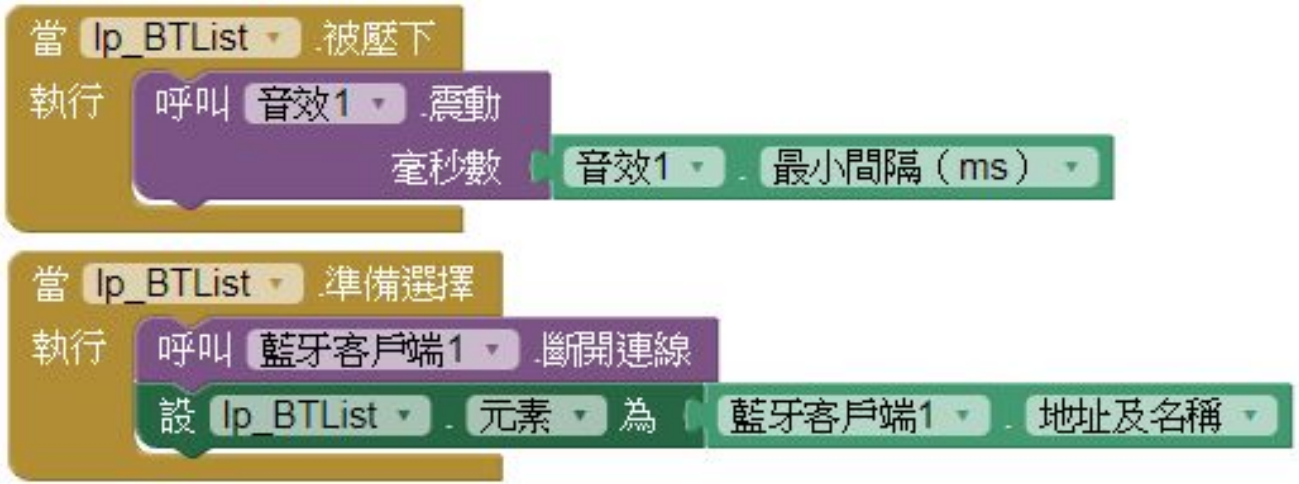
文字
狀態未明

文字對齊
靠左 : 0

文字顏色
預設

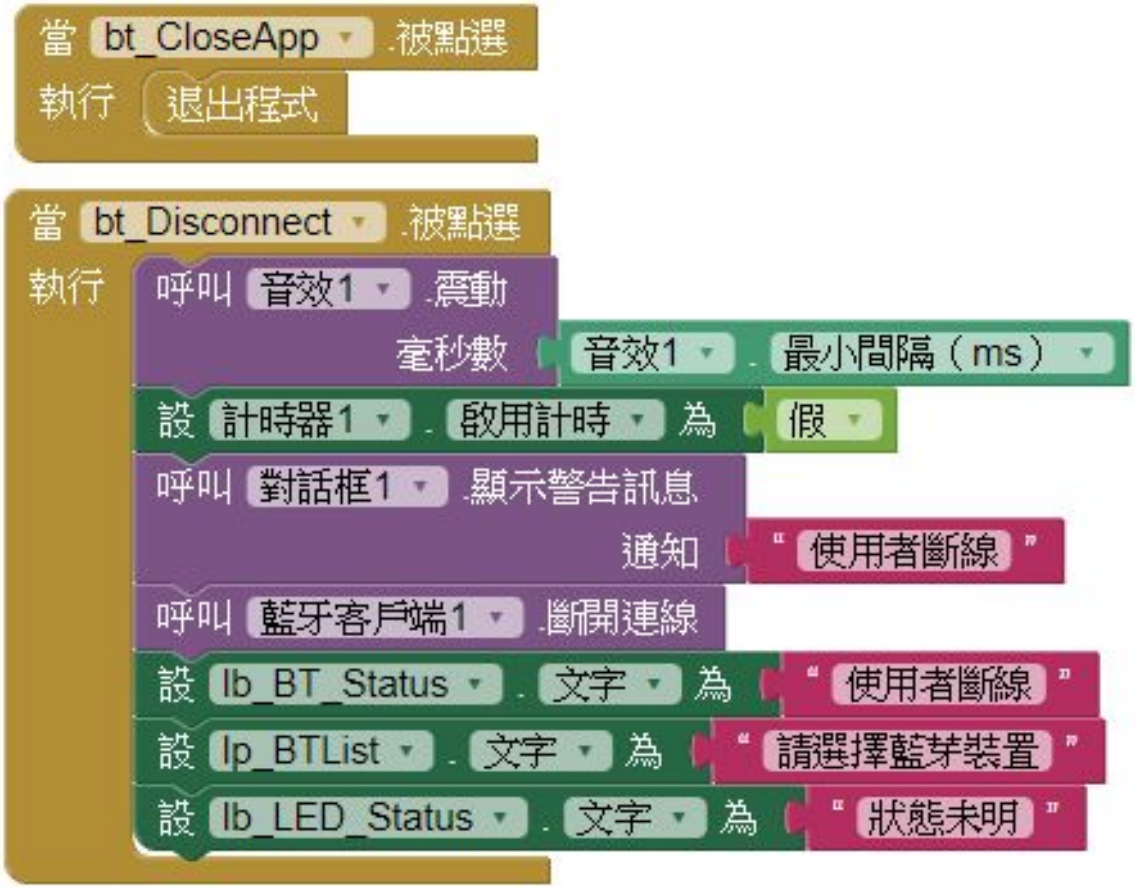
可見性


初始化全域變數 vBT 為 ""
當 Screen1 .初始化
執行
設 計時器1 . 啟用計時 為 假
設 bt_Forward . 啟用 為 假
設 bt_Back . 啟用 為 假
設 bt_TurnLeft . 啟用 為 假
設 bt_TurnRight . 啟用 為 假
設 bt_Stop . 啟用 為 假


當 lp_BTList .被壓下
執行 呼叫 音效1 .震動
毫秒數 音效1 . 最小間隔（ms）
當 lp_BTList .準備選擇
執行 呼叫 藍牙客戶端1 .斷開連線
設 lp_BTList . 元素 為 藍牙客戶端1 . 地址及名稱


當 lp_BTList 選擇完成
執行
設置 global vBT 為 lp_BTList 選中項
設 lp_BTList 文字 為 取 global vBT
設 計時器1 啟用計時 為 假
如果 求長度 取 global vBT > 0
則
如果 呼叫 藍牙客戶端1 連線
地址 取 global vBT
則
設 lp_BTList 文字 為 取 global vBT
設 計時器1 啟用計時 為 真
設 bt_Forward 啟用 為 真
設 bt_Back 啟用 為 真
設 bt_TurnLeft 啟用 為 真
設 bt_TurnRight 啟用 為 真
設 bt_Stop 啟用 為 真
呼叫 對話框1 顯示警告訊息


呼叫 對話框1 .顯示警告訊息
通知 " 連線成功 "
設 lb_BT_Status . 文字 為 " 連線成功 "
否則
呼叫 對話框1 .顯示警告訊息
通知 " 找不到藍芽裝置 "
設 lb_BT_Status . 文字 為 " 找不到藍芽裝置 "


當 bt_CloseApp .被點選
執行 退出程式
當 bt_Disconnect .被點選
執行 呼叫 音效1 .震動
毫秒數 音效1 . 最小間隔（ms）
設 計時器1 . 啟用計時 為 假
呼叫 對話框1 .顯示警告訊息
通知 " 使用者斷線 "
呼叫 藍牙客戶端1 .斷開連線
設 lb_BT_Status . 文字 為 " 使用者斷線 "
設 lp_BTList . 文字 為 " 請選擇藍芽裝置 "
設 lb_LED_Status . 文字 為 " 狀態未明 "


當 bt_Forward .被壓下
執行
呼叫 音效1 .震動
毫秒數 音效1 . 最小間隔（ms）
設 lb_4060_Status . 文字 為 " 前進 "
呼叫 藍牙客戶端1 .發送文字
文字 " F "
當 bt_Forward .被鬆開
執行
設 lb_4060_Status . 文字 為 " 停止 "
呼叫 藍牙客戶端1 .發送文字
文字 " S "


當 bt_Back .被壓下
執行
呼叫 音效1 .震動
毫秒數 音效1 . 最小間隔（ms）
設 lb_4060_Status . 文字 為 " 後退 "
呼叫 藍牙客戶端1 .發送文字
文字 " B "
當 bt_Back .被鬆開
執行
設 lb_4060_Status . 文字 為 " 停止 "
呼叫 藍牙客戶端1 .發送文字
文字 " S "


當 bt_Back .被壓下
執行
呼叫 音效1 .震動
毫秒數 音效1 . 最小間隔（ms）
設 lb_4060_Status . 文字 為 " 後退 "
呼叫 藍牙客戶端1 .發送文字
文字 " B "
當 bt_Back .被鬆開
執行
設 lb_4060_Status . 文字 為 " 停止 "
呼叫 藍牙客戶端1 .發送文字
文字 " S "


當 bt_TurnLeft .被壓下
執行
呼叫 音效1 .震動
毫秒數 音效1 . 最小間隔（ms）
設 lb_4060_Status . 文字 為 " 左轉 "
呼叫 藍牙客戶端1 .發送文字
文字 " L "
當 bt_TurnLeft .被鬆開
執行
設 lb_4060_Status . 文字 為 " 停止 "
呼叫 藍牙客戶端1 .發送文字
文字 " S "


當 bt_TurnRight .被壓下
執行
呼叫 音效1 .震動
毫秒數 音效1 . 最小間隔（ms）
設 lb_4060_Status . 文字 為 " 右轉 "
呼叫 藍牙客戶端1 .發送文字
文字 " R "
當 bt_TurnRight .被鬆開
執行
設 lb_4060_Status . 文字 為 " 停止 "
呼叫 藍牙客戶端1 .發送文字
文字 " S "

Arduino 端程式

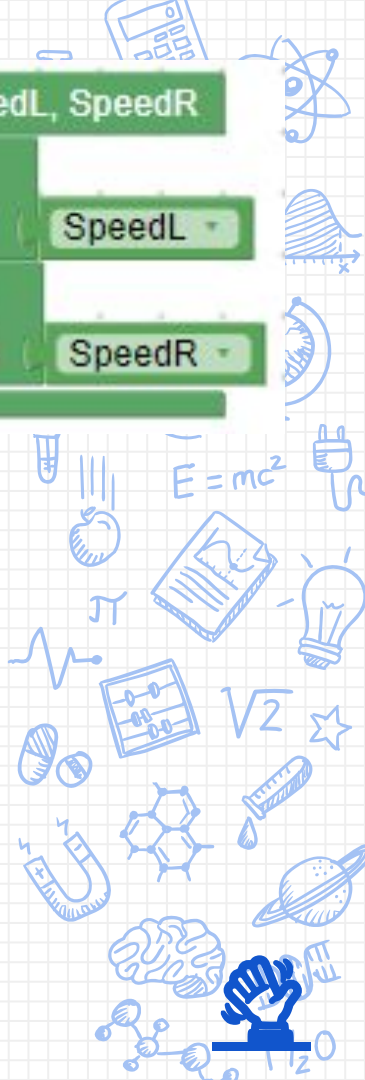
- 慢轉
- RobotCar_BTControl_SlowTurn.xml

慢


設定
設定串列埠 serial 傳輸率 9600 bps
宣告 BTInput 當 char 資料 ' '
迴圈
如果 串列埠有效資料? > 0
執行 賦值 BTInput 到 串列埠輸入
如果 BTInput = 'F'
執行 呼叫 GO 與：
SpeedL 255
SpeedR 255
如果 BTInput = 'B'
執行 呼叫 GO 與：
SpeedL -255
SpeedR -255
如果 BTInput = 'L'
執行 呼叫 GO 與：
SpeedL 0
SpeedR 255
如果 BTInput = 'R'
執行 呼叫 GO 與：
SpeedL 255
SpeedR 0
如果 BTInput = 'S'
執行 呼叫 GO 與：
SpeedL 0
SpeedR 0

```
到 MotorL 與： speed
  如果 speed > 0
  執行
    設定數位腳位 2 為 高
    設定類比腳位 3 資料 255 - speed
  否則
    設定數位腳位 2 為 低
    設定類比腳位 3 資料 0 - speed
到 MotorR 與： speed
  如果 speed > 0
  執行
    設定數位腳位 4 為 高
    設定類比腳位 5 資料 255 - speed
  否則
    設定數位腳位 4 為 低
    設定類比腳位 5 資料 0 - speed
```

```
到 GO 與： SpeedL, SpeedR
  呼叫 MotorL 與：
    speed SpeedL
  呼叫 MotorR 與：
    speed SpeedR
```



Arduino 端程式

- 快轉
- RobotCar_BTControl_FastTurn.xml

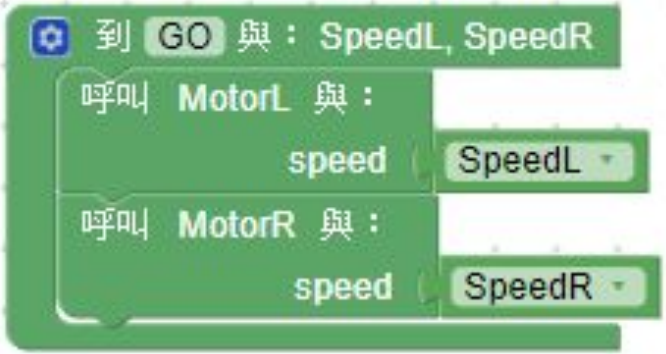
快轉

❖


設定
設定串列埠 serial 傳輸率 9600 bps
宣告 BTInput 當 char 資料 ' '
迴圈
如果 串列埠有效資料? > 0
執行 賦值 BTInput 到 串列埠輸入
如果 BTInput = 'F'
執行 呼叫 GO 與：
SpeedL 255
SpeedR 255
如果 BTInput = 'B'
執行 呼叫 GO 與：
SpeedL -255
SpeedR -255


如果 BTInput = 'L'
執行 呼叫 GO 與：
SpeedL -255
SpeedR 255
如果 BTInput = 'R'
執行 呼叫 GO 與：
SpeedL 255
SpeedR -255
如果 BTInput = 'S'
執行 呼叫 GO 與：
SpeedL 0
SpeedR 0


如果 BTInput = '1'
執行 呼叫 GO 與：
SpeedL -150
SpeedR 150
如果 BTInput = '2'
執行 呼叫 GO 與：
SpeedL 150
SpeedR -150


到 GO 與： SpeedL, SpeedR
呼叫 MotorL 與：
speed SpeedL
呼叫 MotorR 與：
speed SpeedR


到 MotorL 與： speed
如果 speed > 0
執行 設定數位腳位 2 為 高
設定類比腳位 3 資料 255 - speed
否則 設定數位腳位 2 為 低
設定類比腳位 3 資料 0 - speed


到 MotorR 與： speed
如果 speed > 0
執行 設定數位腳位 4 為 高
設定類比腳位 5 資料 255 - speed
否則 設定數位腳位 4 為 低
設定類比腳位 5 資料 0 - speed