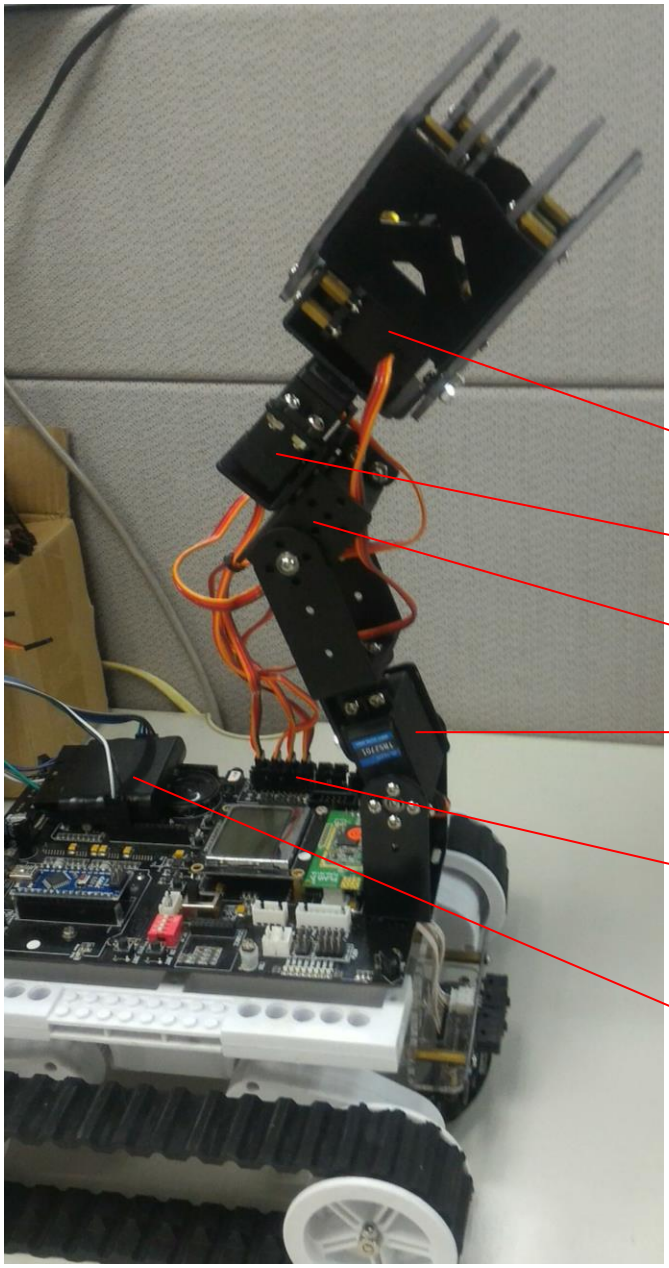
iArm 機械手臂 使用手冊

iArm 快速上手

請開啟**書籤目錄**以方便瀏覽
(例如在 Reader 中按左側的)

iArm 為 **iTank** 專用的 **4 自由度機械手臂**，須固定在 iTank 上使用。另外，套件中還附有 **PS2 手把** (PS2 遙控器) 與 **PS2 無線接收器**，以便用它來靈活地遙控 iArm 及 iTank。

iArm 上有 4 個舵機 (Servo)，由下到上編號為 0~3，各舵機的電線要分別插在 iTank 的 Sv0~Sv3 插座上，然後即可經由 iTank 來控制舵機的旋轉角度。





舵機 3 (手爪開合)

舵機 2 (轉動手爪)

舵機 1 (前伸後縮手臂)

舵機 0 (前伸後縮手臂)

Sv0~Sv3 插座

PS2 手把

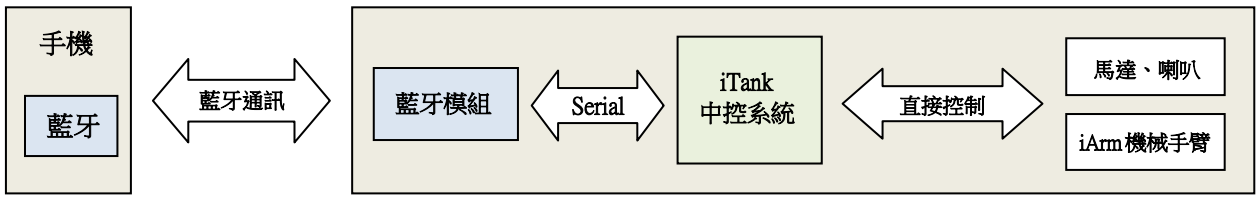
PS2 無線接收器



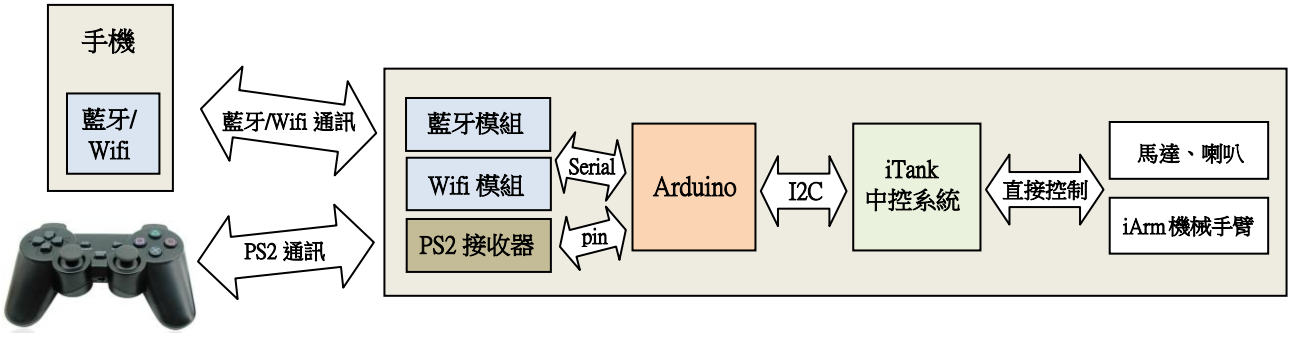
由於手臂的 4 個舵機是插在 iTank 的 Sv0~Sv3 插座上，因此實際上是透過 iTank 來控制手臂。換句話說，就是要對 iTank 下指令來控制手臂，也因此操作 iArm 的方式其實和操作 iTank 是差不多的。

若要遙控 iArm，一般有 2 種方式：

1. 在 iTank 上安裝**藍牙模組**，然後即可透過藍牙來遙控 (無線傳送 iTank 指令做控制)。例如在手機上撰寫 App 經藍牙來無線遙控 iTank 及 iArm：



2. 在 iTank 上加裝 **Arduino UNO** 或 **Nano**，然後撰寫 Arduino 程式來控制 iTank 及 iArm。此方式較為靈活，除了可經由藍牙 (或 Wifi) 來遙控外，還可使用 PS2 手把來遙控。



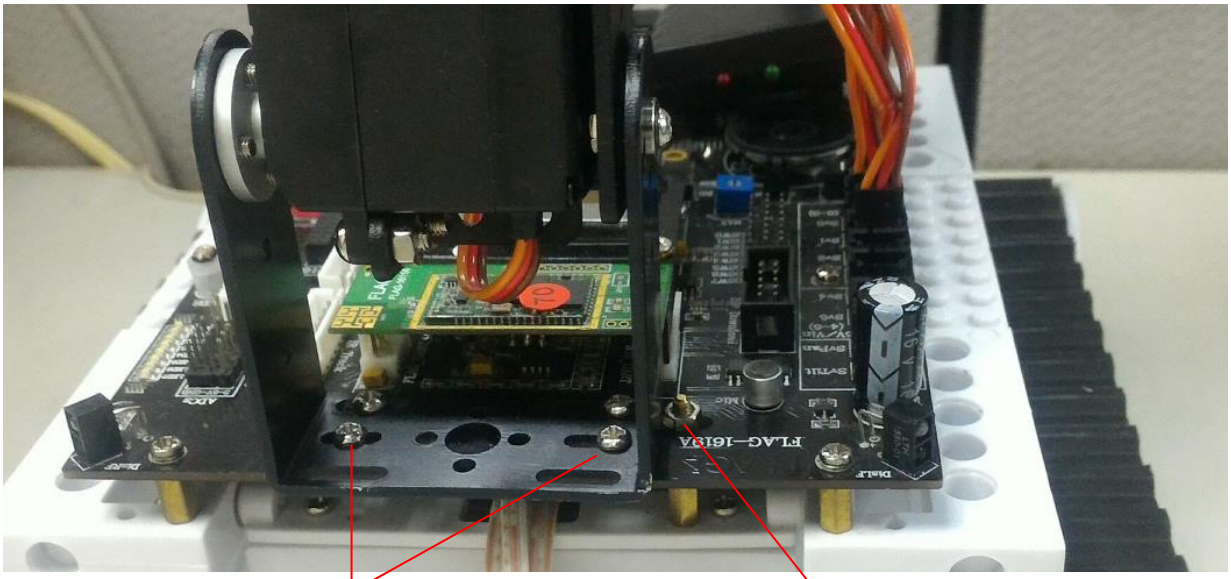
若要使用藍牙傳輸，須先將 iTank 裝上 FLAG-1611A 藍牙模組。
若要使用 Wifi 傳輸，須先將 iTank 裝上 FLAG-N006 Wifi 模組。
若要使用 PS2 手把，須先將 iTank 裝上 PS2 接收器。

3. 以上 2 種方法的操作細節，詳見後文各相關單元的說明。一般來說，使用 PS2 手把比較能夠靈活操作，若使用手機做遙控，限於硬體功能，在操作上會比較不方便。

請注意，**由於 iArm 屬機械設備，請務必小心使用**。在開機、關機、及操作時都要和 iArm 保持適當距離，頭、手、身體都不要進入 iArm 的活動範圍內，以避免受傷。

組裝

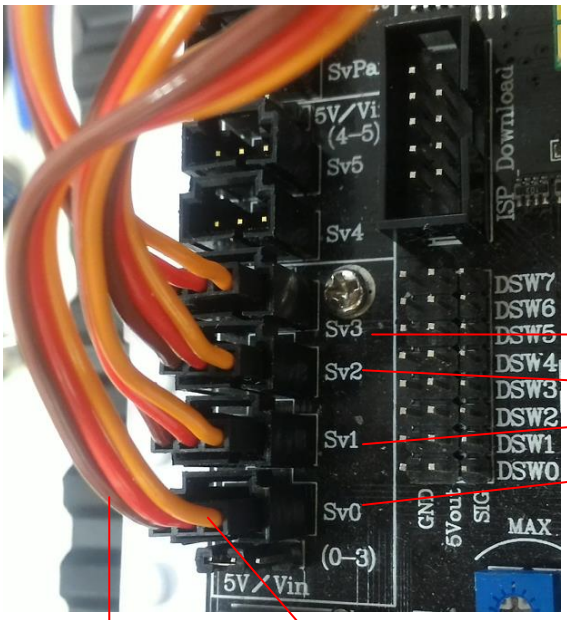
1. 請先將 iTank 上的雲台及固定 iTank 底座的旋轉螺帽都拆除，然後換上 iArm：



1 將 iArm 固定於底座：請於 4 個角落鎖上 4 個螺絲，必要時可將 iTank 底座掀起來會比較好鎖

2 將固定 iTank 底座的旋轉螺帽換成一般的螺帽

2. 將 iArm 的 4 條舵機線路，由下到上依序插在 iTank 的 Sv0~Sv3 插槽上：



最上面(手爪)的舵機 3 插 Sv3

舵機 2 插 Sv2

舵機 1 插 Sv1

最底下的舵機 0 插 Sv0

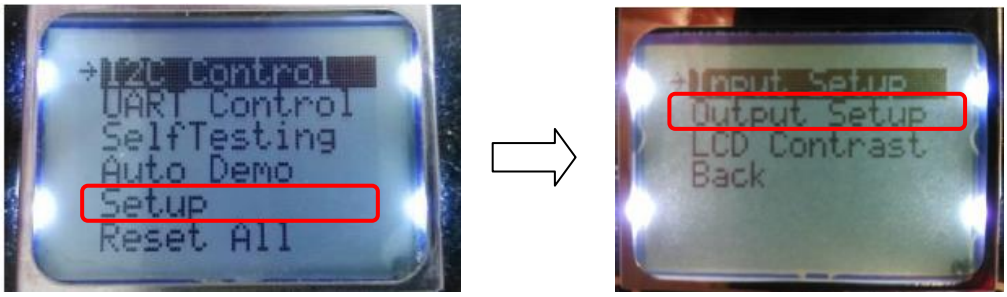
咖啡色線在外側，橘色線在內側，不可插反！

助記水果：**百香果** (外棕內橘)

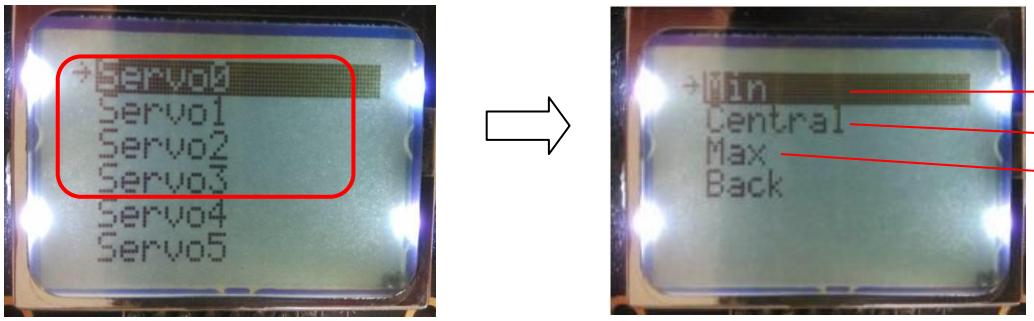
調校 4 個舵機的轉動範圍

iArm 舵機的轉動範圍為 270 度，但在 iTank 的指令中仍是用 0~180 度來控制旋轉角度。我們可依實際需要，利用 iTank 系統的 Servo 調校功能，來調整個每舵機最小(Min, 0 度)、中間(Central, 90 度)、及最大(Max, 180 度) 角度的實際旋轉位置。

1. 啟動 iTank 後，請在 LCD 選單中選取 **Setup**，再選 **Output Setup**



2. 接著再選 **Servo PMW**，然後分別針對 Servo0~Servo3 調校其 Min、Central、Max 角度：



本機械手臂舵機的可轉動角度為 270 度，比一般舵機的 180 要大。在調整時，可視需要將佔空比調到底於預設的 4.0%(Min) 或高於預設的 11.0%(Max)。

4 個舵機的旋轉方向如下 (請注意舵機 0 和 1 的方向是相反的)：
**以下都是以『站在車後向前看』的觀點來描述

舵機	Min (0 度)	Max (180 度)	備註
0	後上	前下	手臂
1	前下	後上	手肘
2	逆時針轉	順時針轉	手腕
3	張開	閉合	手爪

3. 相關操作細節, 請參考『**iTank Menu 系統 (LCD 功能表) 操作手冊.pdf**』 (可在本套件光碟的【iTank 參考資料】資料夾中找到) 的第 **5.2.2** 單元。底下為部份摘要：

5.2.2 Servo PMW (Servo 校準)：本控制板提供了 8 組 Servo (伺服馬達, 或稱舵機) 的輸出接口, 可用來控制 8 個 Servo, 其中有 2 個出廠即已用來連接雲台 Servo。Servo 的驅動基本上必須要有指定電壓的電源, 搭配 50Hz 左右的 PWM 訊號來控制它轉動到目標角度或速度。雖說 Servo 轉動的角度與 PWM duty cycle 變化率之間的關聯非常的接近線性, 但因每個 Servo 接受的 PWM duty cycle 範圍皆有所不同, 為了克服此差異性、並確保所安裝之 Servo 均可正確地受到控制, 因此我們設計了此功能, 以供使用者設定各 Servo 於最小角 (Min degree, 通常是 0 度)、中央角 (Central degree, 通常是 90 度)、與最大角 (Max degree, 通常是 180 度) 時所需提供之 PWM Duty cycle。即使並非一般的 180 度 Servo, 亦可用此功能來設定三個標準角度或速度下的 PWM 輸出強度, 以對 Servo 的活動範圍進行準確地定位。0 度下的 PWM duty cycle 不一定要小於 180 度的 PWM duty cycle, 90 度也不一定要在轉動範圍的正中央; 即方向、角度、與中央角左右的對稱性均可由使用者自行定義。選取此功能後, 必須依照如下的步驟進行設定

Step1. 選取所要調校的 Servo, 進入 Step2 開始對該 Servo 進行設定; 若選取 **Back**, 則回到上一層選單, 結束 Servo 的設定程序

Step2. 選取本次要調整的角度, 此時 AVR 會依原本設定給該角度的 Duty cycle 比值進行輸出控制, 將它轉到對應的角度 (速度), 同時進入 Step3; 若選取 **Back**, 則回到 Step1 重新選定要進行設定的 Servo

Step3. 使用者可重複利用 K0 鈕來增加 PWM 的 Duty cycle 比值、或是利用 K1 鈕來減少 PWM 的 Duty cycle 比值, 每次按鈕更動的 Duty cycle 比值只佔一個 PWM 週期的 0.05%, 且調整範圍僅能介於 0~100%, 藉以將 Servo 調整到所要的角度或速度。調整過程中, LCD 的 Line2 將顯示目前 PWM Duty cycle 的比例值：於左半列顯示 "PWM DC:", 右半邊 7 個位元空間則顯示 3 位整數、1 個小數點、2 位小數, 以及 1 個單位符號 "%"

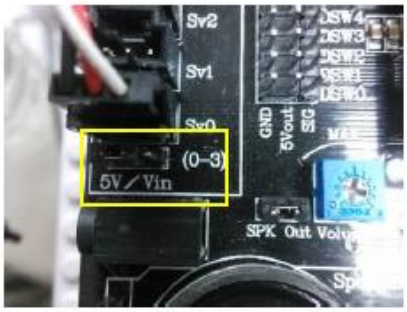
Step4. 把 Servo 調整到目標角度 (或不再繼續調整) 後，只要按下 K3 鈕，即結束針對該選取角度的設定，進入 Step5

Step5. 進入確認選單，若在此選取 **Confirm**，即可儲存新的設定值；若選取 **Back**，將放棄本設定值。完成本步驟後回到 Step2

以下就以 Servo0 為例，展示每個 Servo 的完整調校過程



一開始要先選擇想校準的 Servo



請記得先將 Servo 連接到對應的接口上，並且
利用跳線器調整好正確的電源電壓

5.2.3.1 Servo0 (調校 Servo0 的兩端極限值與中間值)：雖然 Menu 選單中的順序是 Min、Central、Max，但建議調整的順序還是從 Min (最小角或 0 度角) 先設定，然後是 Max (最大角或 180 度角)，最後再設定 Central (中央角或 90 度角)



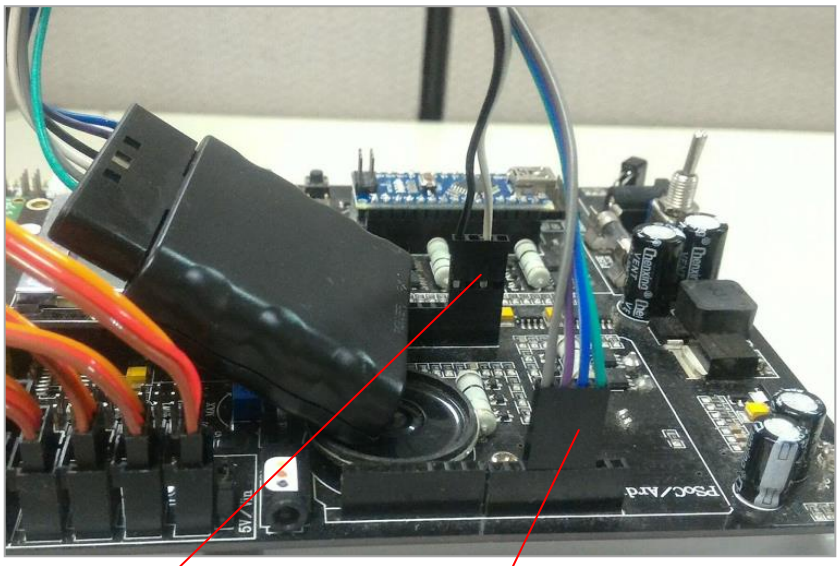
此處可設定 Servo 順時針與逆時針轉動時的極限角度;
即使 Servo 本身可以轉動更大角度,也可以利用此功能
來予以設限,以滿足使用上的需要。而中央角度(90 度角)
也可以自行定義,從中央角度到兩端極限角度之間的轉角
不一定要對稱

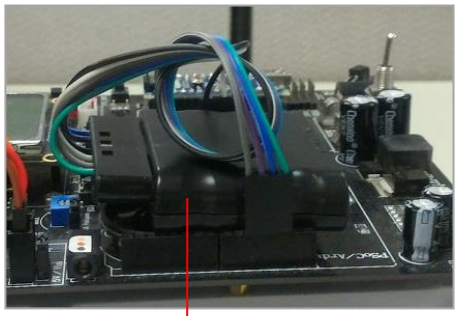
5.2.3.1.1 Min (最小角度)：設定數值上的 0 度，也就是實際轉動範圍最小角度的 PWM 輸出，預設 Duty cycle 比值為 4%，可利用 K0 與 K1 來變更，過程中 LCD 的 Line2 將顯示目前的 Duty cycle 比值；最後用 K3 結束設定，進入下一層選單。此

更多說明請參考『**iTank Menu 系統 (LCD 功能表) 操作手冊.pdf**』(可在本套件光碟的【iTank 參考資料】資料夾中找到) 的第 **5.2.2** 單元。

使用 PS2 手把遙控 iArm 及 iTank

1. 請先將 **PS2 接收器**的線路接在 iTank 的 Arduino Shield 插槽中：



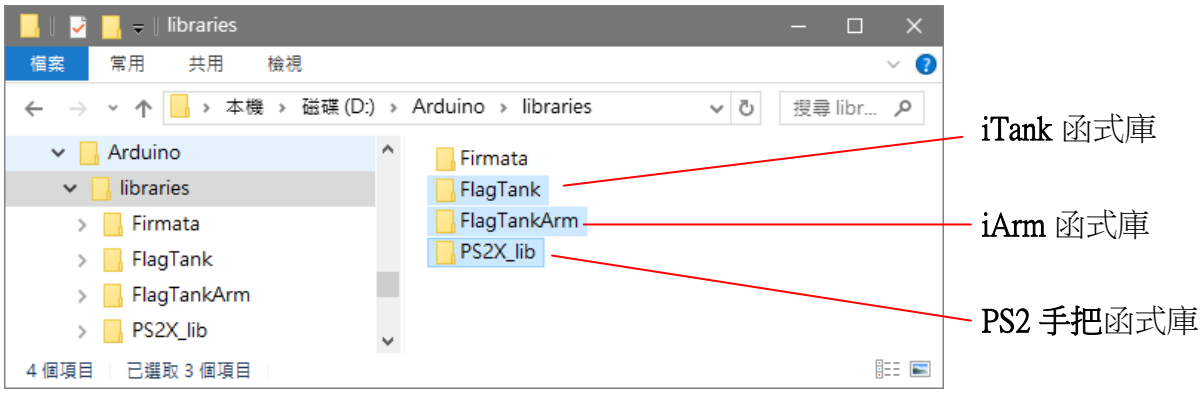


黑白 2 條線接在 Gnd 及 3V 插槽 (插在最靠右邊)

灰紫藍綠 4 條線接在 D10~D13 插槽 (插在最靠右邊但留 2 孔)

線插好後可將 PS2 接收器安置在 2 排線之間

2. 如果您有已燒好 **iArm 預錄程式**的 Nano 或 Uno，可直接跳到步驟 4。將套件光碟中【Arduino 函式庫】資料夾下的 **FlagTank**、**FlagTankArm**、及 **PS2X_lib** 資料夾，全部複製到 Arduino **草稿碼簿**資料夾的 **libraries** 資料夾下：



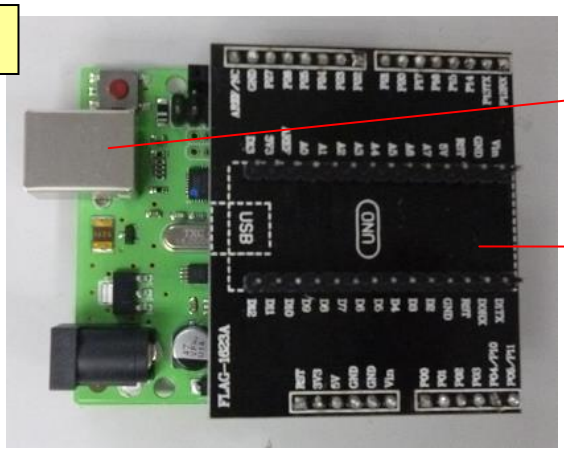
可在 Arduino 中執行『**檔案/偏好設定**』命令來查看**草稿碼簿**資料夾的位置。



3. 在 Arduino IED 中執行『**檔案/範例/FlagTankArm/iArm_Preload**』命令開啟 **iArm 預錄程式**，然後編譯並上傳到 UNO/Nano 中。

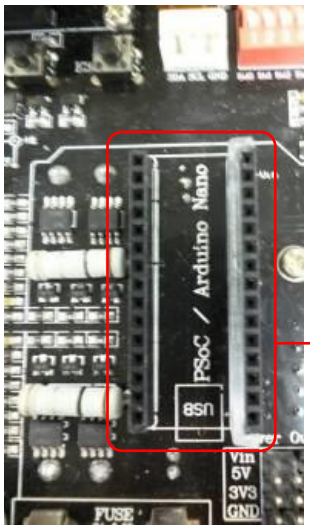
4. 將 Arduino UNO/Nano 插到 iTank 上。底下分別介紹 UNO 及 Nano 的插法：

Uno

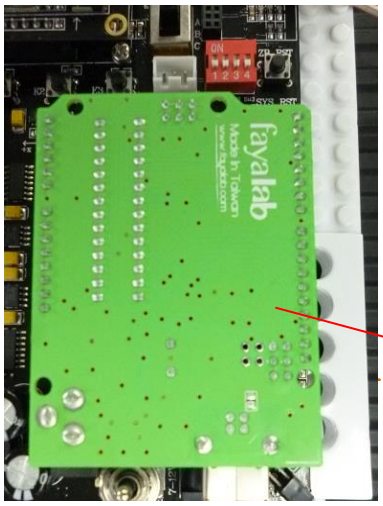


1.將 Arduino **UNO** 與 PC 的 USB 連接線拔除

2.插上轉接板

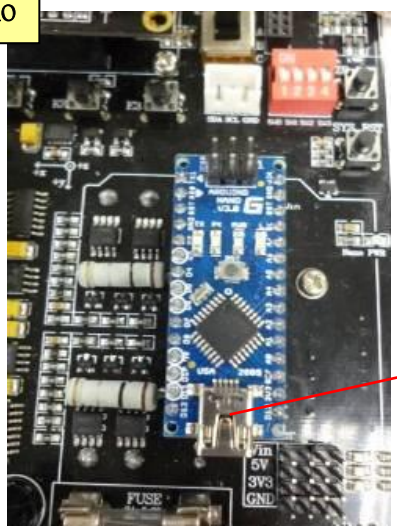


3.找到 iTank 上的插槽



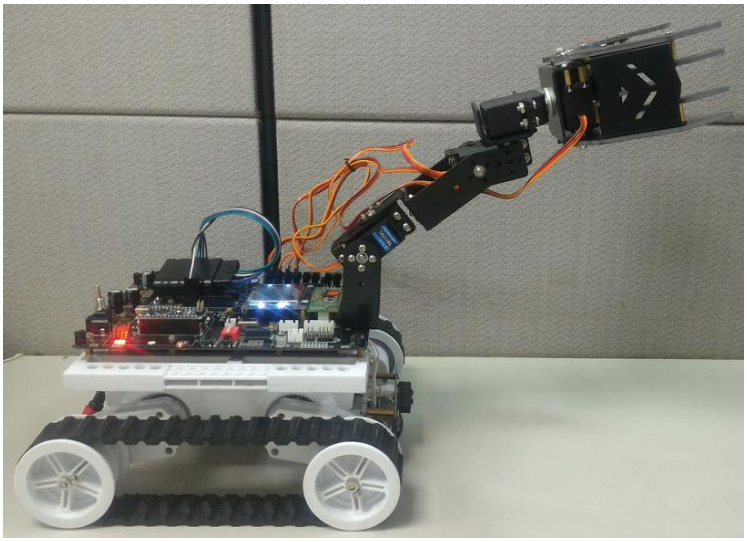
4.將轉接板上的針腳插入 iTank 插槽並注意方向（**USB 插頭方向朝車子後方**）

Nano



使用 Nano 時可直接插入 iTank，但須注意方向（**USB 插頭方向朝車子後方**）。

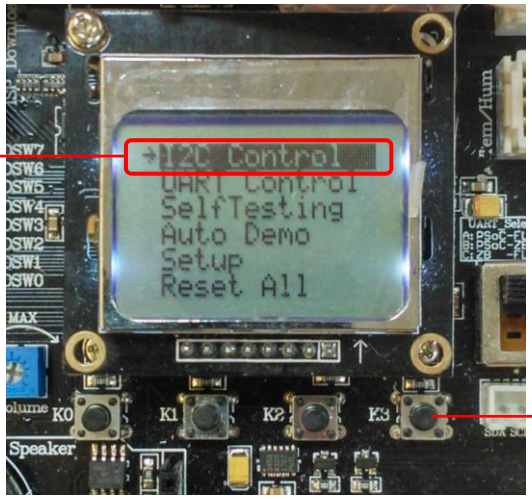
5. 請將 iTank 前方保持淨空，然後開啟 iTank 電源，此時 iArm 手臂會向前上方伸展 (各舵機轉到 iTank 預設的 90 度位置)：



6. 在 LCD 中選擇 **I2C Control** 並按二下 K3 鈕，此時 iArm 會轉成 Z 字型，表示 iTank 已由 Arduino 預錄程式掌控。

1. 確認目前反白項目為 I2C Control (如不是，請按 K0、K1 移動)

NOTE：按 K0 可將反白游標向上移動；按 K1 則是將反白游標向下移動；按 K3 表示選取/確認。



2.按下 K3 鈕

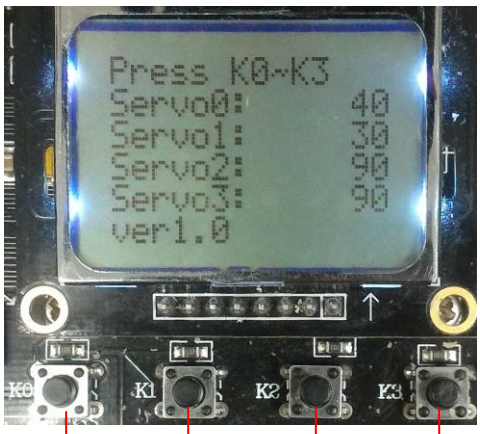


3.再按 K3 鈕確認



出現初始化訊息，稍待幾秒後，畫面會閃爍並消失

7. 接著可在 LCD 中觀看 4 個舵機的角度，並先試用 K0~K3 鈕分別控制舵機 0~3 的旋轉（每次放開再按可切換正轉/反轉）：

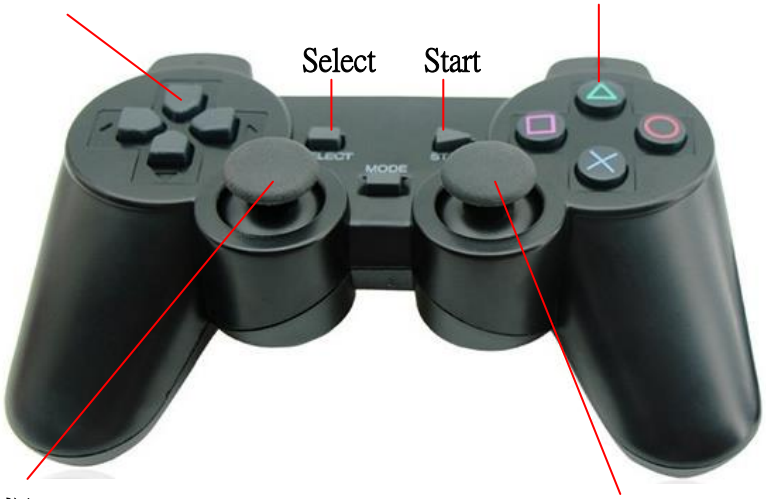


K0~K3 鈕分別控制舵機 0~3 的旋轉

8. 開啟 PS2 手把的開關(在前側右邊，見下頁圖)，即可控制 iArm 轉動或 iTank 開車：

左十字鈕：
垂直二鈕-手臂(舵機 0) 前伸與後縮
水平二鈕-手肘(舵機 1) 前伸與後縮

右十字鈕：
垂直二鈕-手腕(舵機 2) 左右旋轉
水平二鈕-手爪(舵機 3) 張開與閉合



左搖桿 (L3)：
按一下左搖桿可開啟(或關閉)行車功能，開啟行車功能時才可用搖桿控制行車。
(左、右搖桿都可用來控制行車)

右搖桿 (R3)：
按一下右搖桿可將手臂回到 Z 字預設姿勢

開啟電源後若手把上的 LED 燈未亮，可能是在**省電模式**中，此時按一下 START 鈕即可回復正常，若仍未亮則可能是沒電了。

行車功能預設為關閉，以避免不小心碰到搖桿而造成 iTank 意外的移動。

9. PS2 手把的進階操作：

- 左十字鈕可同時按住 2 鈕，讓舵機 0 和 1 同時轉動，例如同時按左+上、或右+下。
- 按住 L2 或 R2 鈕，再轉動手臂或開車，可以減慢速度。
- 按住 L2 鈕再按 R1 鈕，可切換爪子的張開、閉合。
- 按住 R2 鈕再按 L1 鈕，可切換手臂的前伸、後縮。



10. 錄製及播放姿勢（會**永久儲存**在 Arduino 的 EEPROM 記憶體中）：

- 按住 **SELECT** 鈕及 **L2**(或 **R2**) 鈕，再按一下**右搖桿**(R3 鈕)，可將目前的姿勢儲存為**預設姿勢**。
- 按住 **SELECT** 鈕，再按**左/右搖桿**或**左/右十字鈕**的任一按鈕，可將目前姿勢儲存在該按鈕上。未來只要按住 **L1** 或 **R1** 鈕不放，再按有儲存姿勢的按鈕，就會播放所儲存的姿勢。

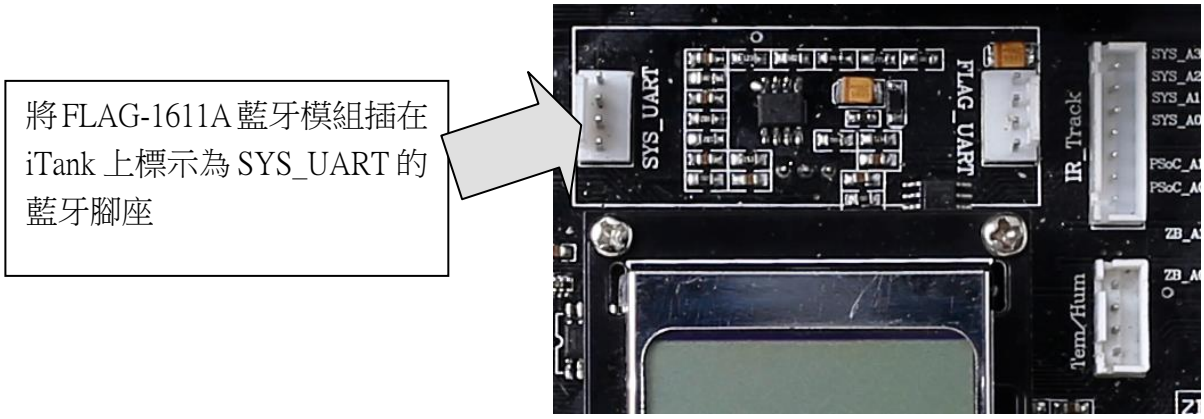
用手機經藍牙遙控 iArm 及 iTank

1. 請先由 Android 手機的 Play 商店安裝**旗標藍牙機械手臂車** App。

另外也可改用套件光碟中的 iTankArm.apk 來安裝：請自行將 apk 傳到手機中，並開啟手機系統設定中的【安全性/不明的來源(允許安裝來源不明的程式)】選項，然後進行安裝。

2. 若要直接操控 iTank 而不使用 Arduino，請如下安裝**旗標藍牙模組** (FLAG-1611)：

   A. 將 FLAG-1611A 藍牙模組插在 iTank 上，針腳端要**朝左**。

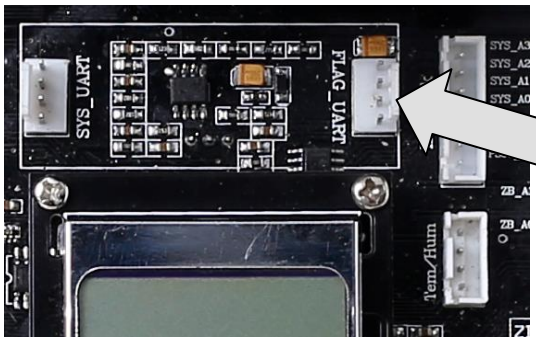


   B. 開啟 iTank，然後在 LCD 選單中選 UART Control，然後按二下 K3 再按 K1 鈕 (選單會陸續顯示 START、K1->NO、然後變空白)，即可進入 iTank 的 UART 控制模式。在此模式下，手機 (或其他裝置) 經藍牙傳送到 FLAG-1611A 的指令，會再經 UART 傳送給 iTank 執行，而 iTank 經 UART 傳出的資料，也會由 FLAG-1611A 轉送給手機。



3. 若要用藍牙並**透過 Arduino** (須先上傳**預錄程式**或您自行撰寫的程式) 來操控 iTank, 則請改為如下安裝：

A. 將 FLAG-1611A 藍牙模組插在 iTank 上, 針腳端要**朝右**。然後將 DIP 推到 A 的位置 (會連通藍牙模組及 Arduino), 再插上 Arduino UNO (或 Nano)。



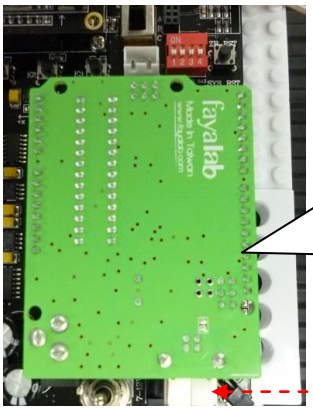
iTank 上標示為 FLAG_UART 的藍牙腳座，可接到 Arduino 腳座的 0/1 腳（RX/TX）。因此 Arduino 可利用 Serial 函式庫接收手機送來的資料、也可主動傳送資料到手機。



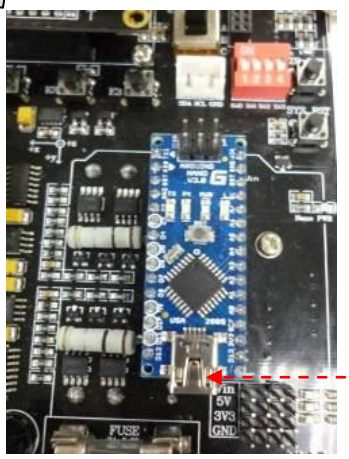
1. 將藍牙模組插入右邊的 FLAG_UART 腳座



2. 將 LCD 旁的 UART Select 開關扳至 A。



插上 Arduino **UNO** 及**轉接板**並注意方向 (USB 插頭方向朝車子後方)。



若使用 **Nano** 可直接插入 iTank (USB 插頭方向朝車子後方)。

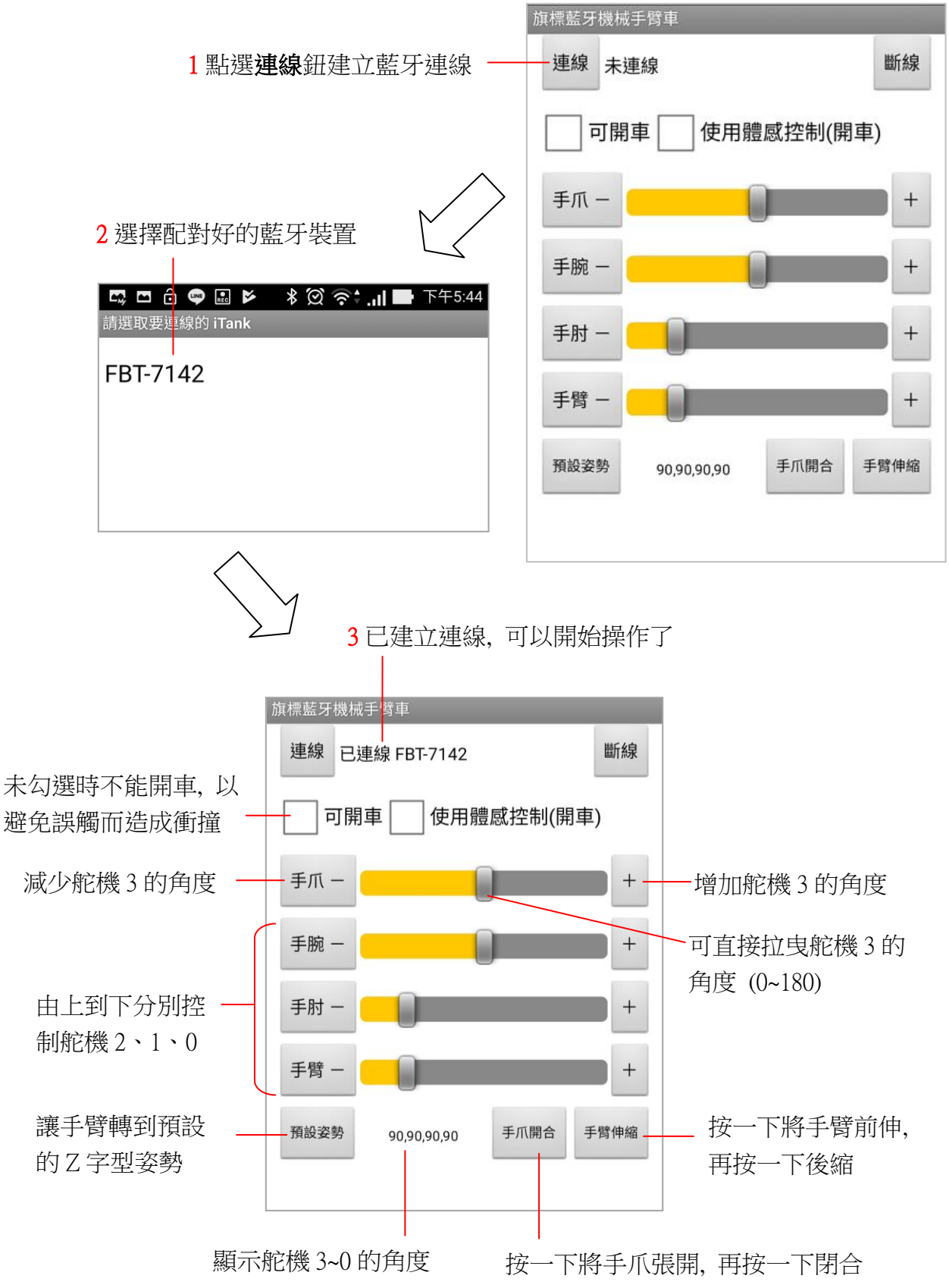
B. 開啟 iTank，然後在 LCD 選單中選 **I2C Control**，再按二下 K3 鈕，此時 iArm 會轉成 Z 字型，同時 LCD 上會顯示各舵機的角度，表示 iTank 已由 Arduino 預錄程式掌控。在此模式下，手機 (或其他裝置) 經藍牙傳送到 FLAG-1611A 的指令，會經 UART 傳送給 Arduino 解析，然後 Arduino 再經 I2C 對 iTank 進行控制。



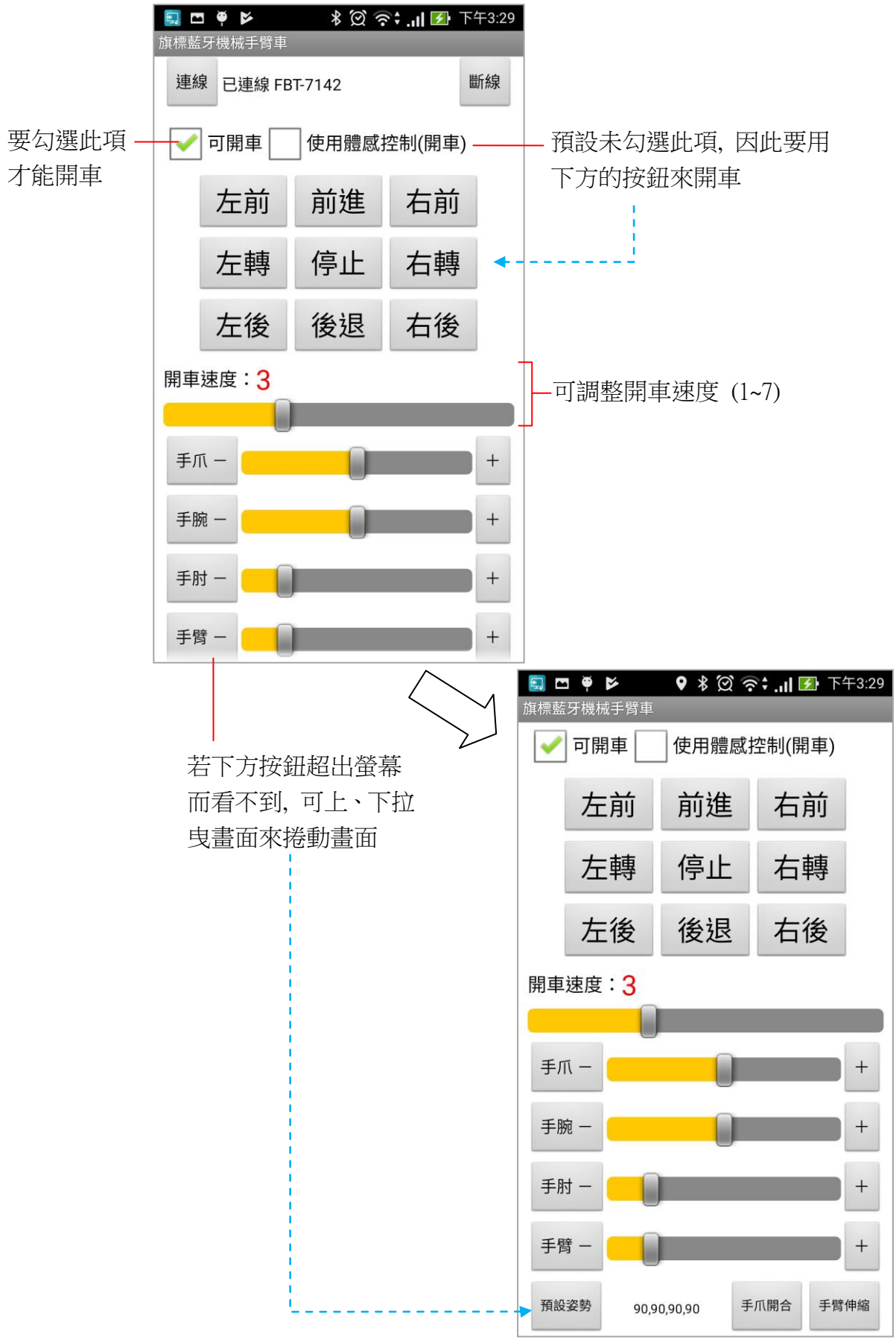
4. 在手機中先進入系統的**設定**畫面，然後開啟藍牙功能，並和 iTank 上的 FLAG-1611A 藍牙模組進行配對(只需做一次即可，配對前請務必開啟 iTank 並完成步驟 2 或 3)：



5. 然後開啟『**Flag iTankArm 藍牙機械手臂車**』App：



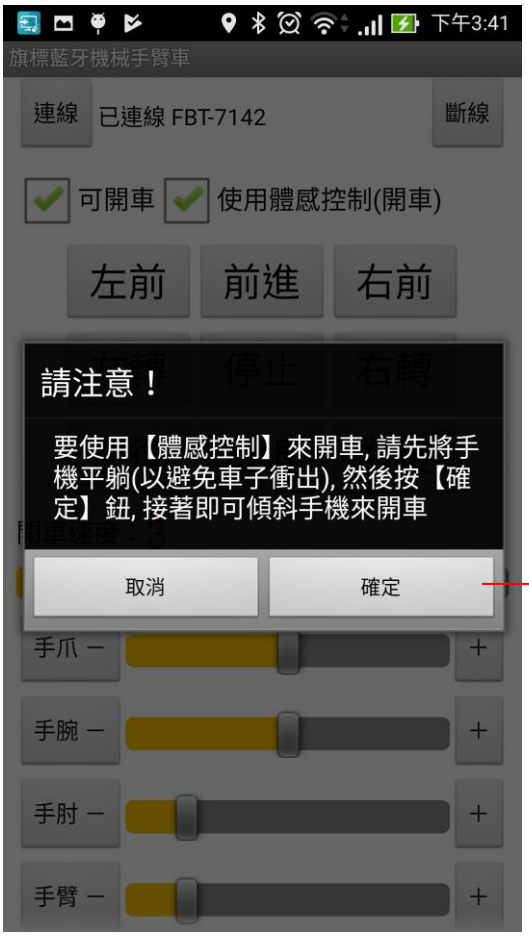
〇使用方向按鈕來開車：


下午3:29
旗標藍牙機械手臂車
連線
已連線 FBT-7142
斷線
可開車
使用體感控制(開車)
左前
前進
右前
左轉
停止
右轉
左後
後退
右後
開車速度：3
手爪 －
＋
手腕 －
＋
手肘 －
＋
手臂 －
＋
預設姿勢
90,90,90,90
手爪開合
手臂伸縮
要勾選此項
才能開車
預設未勾選此項，因此要用
下方的按鈕來開車
可調整開車速度（1~7）
若下方按鈕超出螢幕
而看不到，可上、下拉
曳畫面來捲動畫面

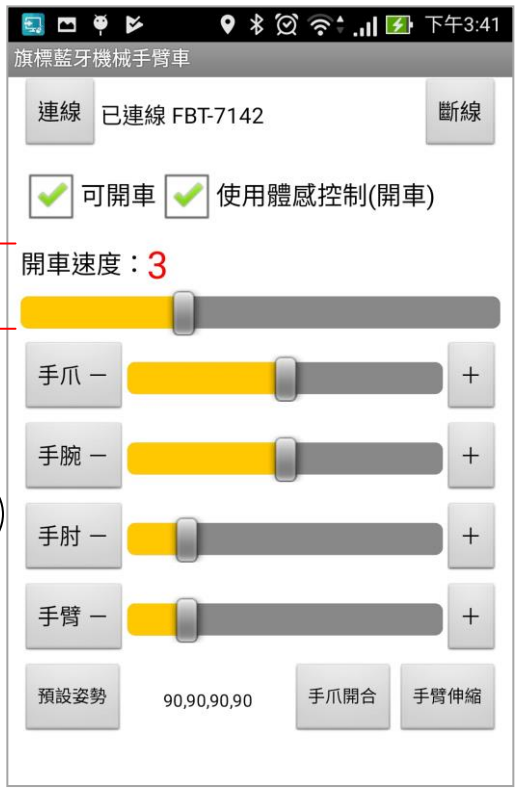
○使用體感控制來開車：



勾選此項即可開啟**體感控制**開車功能



會出現提示交談窗，請先將手機平放(以免車子衝出)，再按**確定**鈕



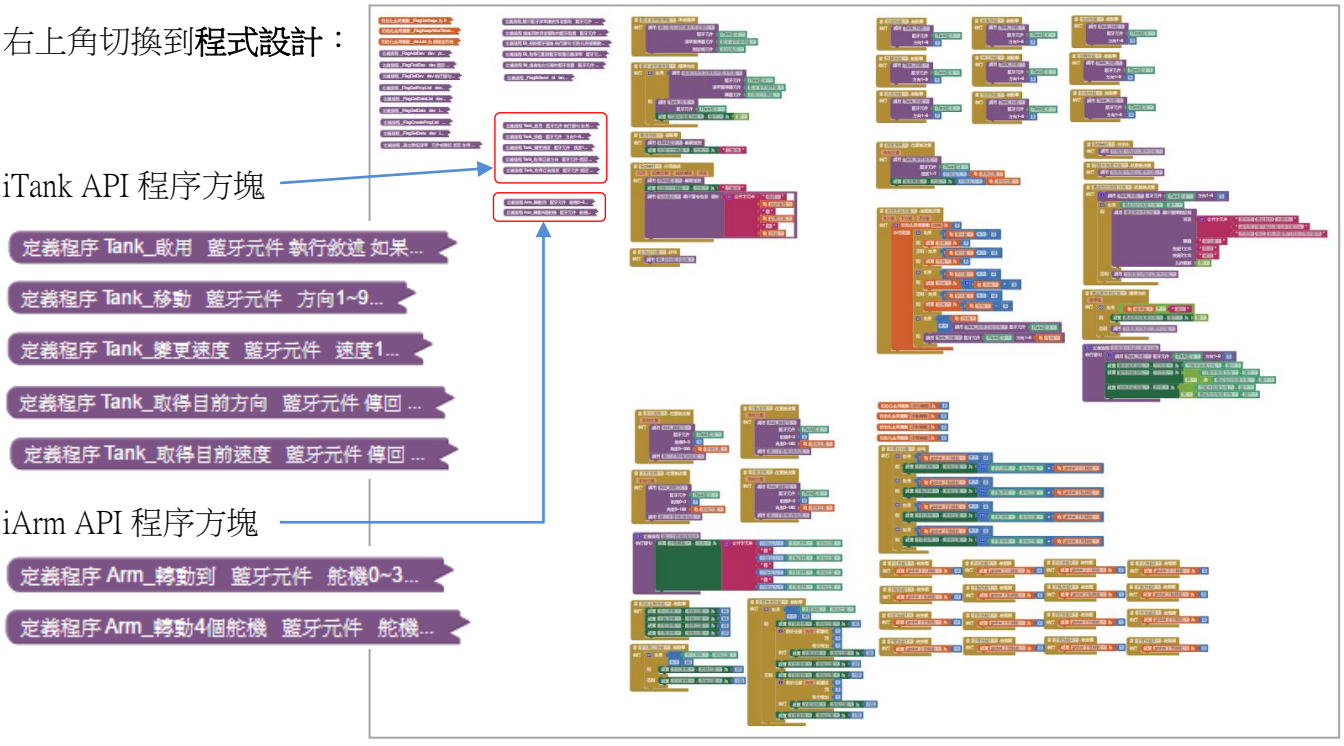
可調整開車速度（1~7）

可以前/後/左/右傾斜手機來開車了

本程式是使用 AI2 (App Invertor 2) 所製作的，若想要自行修改或撰寫新程式，可先登入 http://ai2.appinventor.mit.edu 網站 (若無帳號可免費申請)，然後匯入套件光碟中的 FlagTankArm.aia 專案，即可進行更改 (請先將 AI2 介面**切換到繁體中文**，才能正確編譯)，或利用背包將其中的方塊複製到其他專案中使用。



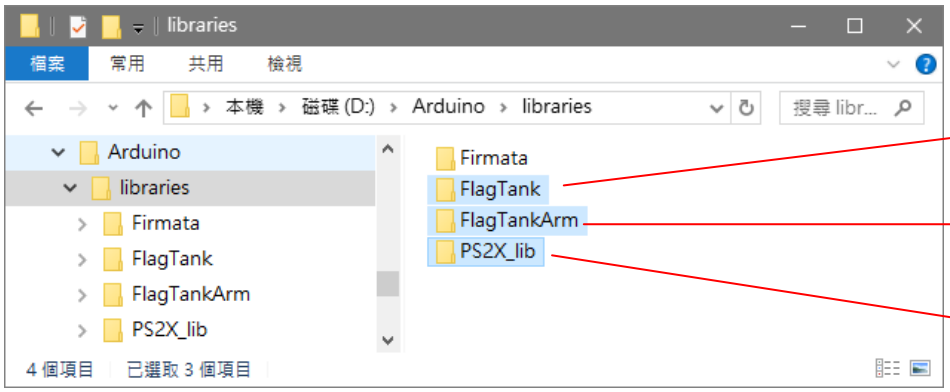
右上角切換到**程式設計**：



自行撰寫 Arduino 程式控制 iArm 及 iTank

若想要加強 **iArm 預錄程式**的功能，或是要撰寫新的程式來控制 iArm，可在 Arduino IDE 中執行『**檔案/範例/FlagTankArm/iArm_Preload**』命令開啟 **iArm 預錄程式**來修改使用，或是先開啟功能較單純的『**檔案/範例/FlagTankArm/iArm_UsePS2**』範例來學習。

底下我們就來介紹 iArm_UsePS2 程式的寫法，請先依照第 8 頁步驟 2 的說明安裝程式所需的 3 個函式庫： **FlagTank**、**FlagTankArm**、及 **PS2X_lib** (將套件光碟中的資料庫複製到 Arduino **草稿碼簿**資料夾的 **libraries** 資料夾下)：



接著在 Arduino IED 中執行『**檔案/範例/FlagTankArm/ iArm_UsePS2**』命令開啟範例檔：

```
#include <FlagTank.h>      // 引用 iTank 函式庫
#include <FlagTankArm.h> // 引用 iArm 函式庫
#include <PS2X_lib.h>      // 引用 ps2 函式庫

PS2X ps2x;                 // 建立 PS2 手把物件

void setup() {
  Serial.begin(9600);      // 初始化序列通訊 (供 PS2 手把偵錯使用)
  iTank.begin();           // 初始化 iTank 並等待就緒
  ps2_config();            // 初始化 PS2 手把
  iArm.turnToDefault();    // 先將手臂轉到預設姿勢
}
```

引用 3 個函式庫的表頭檔

定義 PS2 手把物件，以後都是用此物件來偵測 PS2 的按鈕狀態

iTank 物件會自動建好，因此可直接執行其 begin()做初始化

呼叫此函式來初始化 PS2 手把 (此函式定義在程式的最後)

iArm 物件也會自動建好(且不需要初始化)，呼叫此方法可將手臂轉到 Z 字形的預設姿勢

在 setup() 中做好初始化之後，就可在 loop() 中不斷偵測 PS2 各按鈕的狀態，然後對 iArm 做對應的控制：

利用 ps2x 物件的 **read_gamepad()** 方法
來讀取 PS2 各按鈕的狀態

```
void loop() {
  delay(50);           // 暫停一下
  ps2x.read_gamepad();  // 讀取 PS2 的狀態

  // 偵測左十字鍵的4個按鈕，如果按著就增減對應舵機的角度
  if(ps2x.Button(PSB_PAD_UP))    { iArm.turn(0, 1); } // 如果按著向上鈕就將舵機0加1
  if(ps2x.Button(PSB_PAD_DOWN)) { iArm.turn(0,-1); } // 如果按著向下鈕就將舵機0減1
  if(ps2x.Button(PSB_PAD_RIGHT)){ iArm.turn(1,-1); } // 如果按著向右鈕就將舵機1減1
  if(ps2x.Button(PSB_PAD_LEFT)) { iArm.turn(1, 1); } // 如果按著向左鈕就將舵機1加1

  // 偵測右十字鍵的4個按鈕，如果按著就增減對應舵機的角度
  if(ps2x.Button(PSB_GREEN)) { iArm.turn(2,-1); }   // 如果按著綠色鈕就將舵機2減1
  if(ps2x.Button(PSB_BLUE))  { iArm.turn(2, 1); }   // 如果按著藍色鈕就將舵機2加1
  if(ps2x.Button(PSB_RED))   { iArm.turn(3, 1); }   // 如果按著紅色鈕就將舵機3加1
  if(ps2x.Button(PSB_PINK))  { iArm.turn(3,-1); }   // 如果按著粉紅鈕就將舵機3減1

  if(ps2x.ButtonPressed(PSB_R3)) {    // 如果按下 R3 鈕（要從未按下變成按下才算）
    iArm.turnToDefault();             // 就將手臂轉到預設姿勢
  }
}
```

接著呼叫 ps2x 的 **Button()** 方法來檢查各按鈕是否**被按著**，若是則轉動對應的舵機

呼叫 ps2x 的 **ButtonPressed()** 方法來檢查 R3 鈕是否**被按下過**
(由未按下變成按下)，若是則將手臂轉動到 Z 字形的預設姿勢

請注意，這裡都是用 ps2x、iArm 物件來操作 PS2 手把及 iArm 手臂。若要操作 iTank，則可使用 iTank 物件。

程式最後面是 PS2 的初始化函式 ps2_config()，只須直接使用即可。若初始化發生錯誤（例如 PS2 接收器未正確接到 iTank 上），則會由 Serial 輸出錯誤訊息，可借此訊息找出問題所在。

```
// PS2 初始化函式，若看不懂可直接複製使用
// 若發生錯誤會將相關訊息由 Serial 送出
void ps2_config(){
  int ps2_error = 0;
  byte ps2_type = 0;
```

(函式後面的部份省略…)

Arduino iArm 函式庫 使用說明

由於 iArm 是安裝於 iTank 上, 因此 iArm 函式庫也必須配合 iTank 函式庫來運作。iArm 函式庫已預先定義好了 iArm 物件, 且不需做初始化即可直接使用。

iArm 在啟動時會假設各舵機的角度均為 90 度 (iTank 開機時會自動將各舵機轉到 90 度), 接著我們可用以下幾個常用的方法來操作手臂：

- iArm.**turn**(舵機編號, **正反轉**多少度)：
  例如 iArm.turn(2, -5) 即可將舵機 2 反轉 5 度。
  若轉動超過允許的範圍(0~180), 則只會轉到範圍的邊界 (0 或 180 度)。
  傳回值：true 表示成功, false 表示舵機編號錯誤或角度超出範圍。

- iArm.**turnTo**(舵機編號, **轉到**多少度)：
  例如 iArm.turnTo(2,50) 即可將舵機 2 轉到 50 度。
  若指定角度超過允許的範圍(0~180), 則不會轉動。
  傳回值：true 表示成功, false 表示舵機編號錯誤或角度超出範圍。

- iArm.**read**(要**讀取角度**的舵機編號)：
  例如 iArm.read(2) 會傳回舵機 2 的目前角度。

- iArm.**turnToDefault**()：
  將手臂轉到預設姿勢 (舵機 0~3 轉到 30、20、90、90 度, 手臂會呈 Z 字形)
  預設姿勢可以變更, 詳見後文說明。

進階方法

- 如果希望同時指定 4 個角度, 讓各舵機一次轉到定位, 同樣可用 **turnTo**( )方法但傳入不同數量的參數：

```
void turnTo(byte deg0, byte deg1, byte deg2, byte deg3);        // 轉動 4 個舵機到指定角度
void turnTo(byte deg[]);                                        // 轉動 4 個舵機到指定角度 (以陣列指定角度)
```

例如：

```
iArm.turnTo(50,   60, 70, 80);
byte aDeg[] = { 50,   60, 70, 80 };   // 宣告 byte 陣列並指定初值
iArm.turnTo(aDeg);
```

- 以上一次轉動 4 個舵機的功能, 也可以用慢動作來轉, 就是分解成較小的動作, 每 0.1 秒動一次。例如若分解為 10 個小動作, 則會以慢動作在 1 秒 (=10 個 0.1 秒) 內完成。

```
void show(byte deg0, byte deg1, byte deg2, byte deg3, byte tick);   // 慢動作轉到指定角度
void show(byte deg[], byte tick);                  // 慢動作轉到指定角度 (以陣列指定角度)
void showDefault(byte tick);                       // 慢動作轉到舵機預設姿勢
```

以上的 tick 參數就代表要用幾個 0.1 秒小動作來完成 (1tick =100ms=0.1sec), 例如：

```
iArm.show(50, 60, 70, 80, 5);        // 慢動作於 0.5 秒 轉到定位
byte aDeg[] = { 50, 60, 70, 80 };    // 宣告 byte 陣列並指定初值
iArm. show (aDeg, 10);               // 慢動作於 1 秒轉到定位
iArm. showDefault(20);               // 慢動作於 2 秒轉到預設姿勢
```

- 如果想要變更預設姿勢, 可使用 setDefault() 方法：

```
void setDefault(byte deg0, byte deg1, byte deg2, byte deg3);   // 設定舵機預設姿勢的角度
void setDefault(byte deg[]);                  // 設定舵機預設姿勢的角度 (以陣列指定)
void setDefault();                            // 將目前角度設定為舵機預設姿勢
```

例如：

```
iArm. setDefault (50, 60, 70, 80);
byte aDeg[] = { 50, 60, 70, 80 };
iArm. setDefault (aDeg);
iArm. setDefault ();        // 將目前姿勢設定為舵機預設姿勢
```

- 舵機允許的角度範圍預設是 0~180 度, 但也可以更改。另外, 4 顆舵機預設要連接到 iTank 的 Sv0~Sv3 插槽, 但同樣可以用程式變更 (需配合實際接線)：

```
void setRange(byte Servo, byte degMin, byte degMax);        // 必須 degMin < degMax
                                                            // 且 degMax<= 180
void setServoPin(byte pin0, byte pin1, byte pin2, byte pin3);   //設定 4 顆 Servo 的連接腳位
```

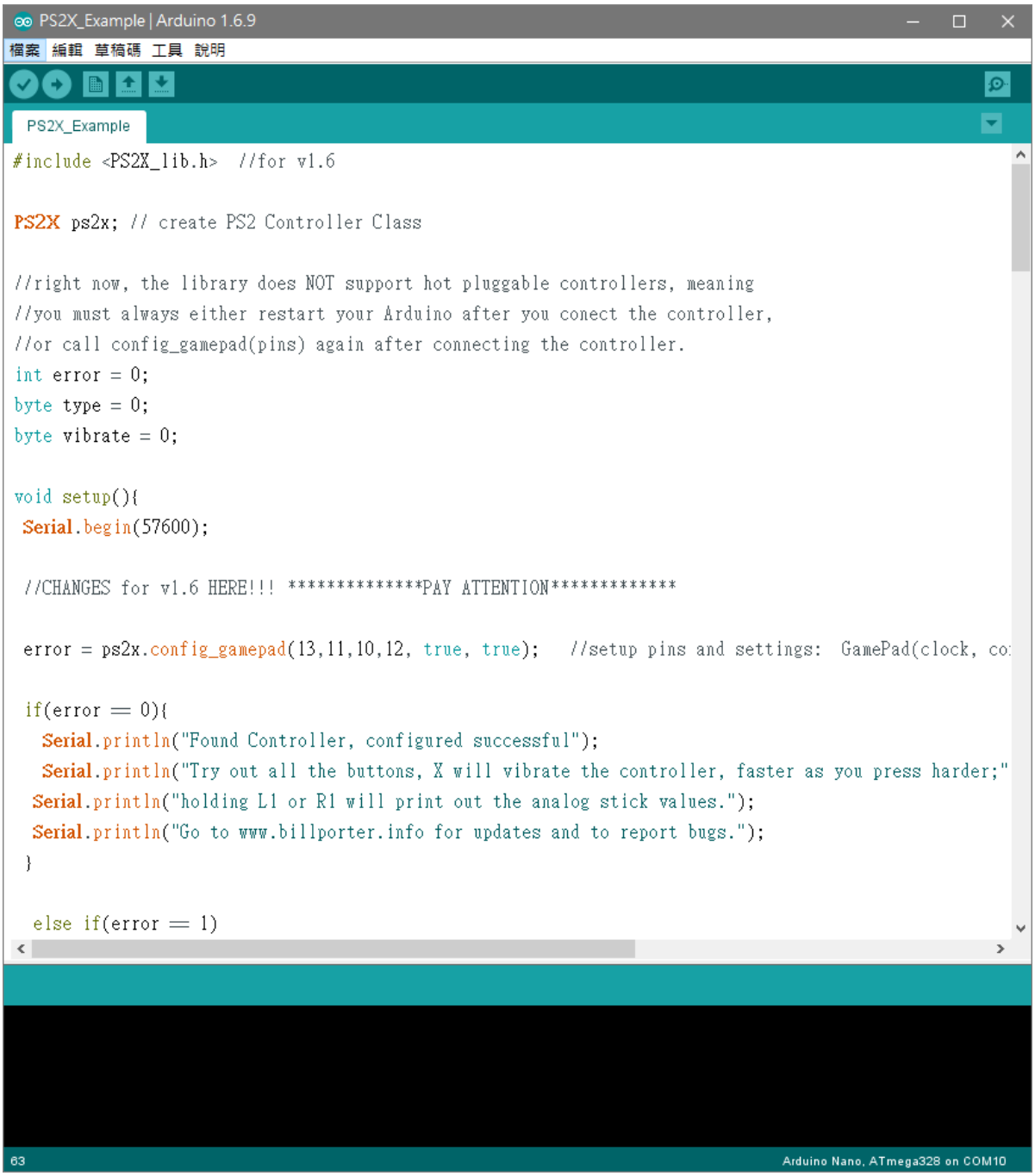
例如：

```
iArm.setRange(2, 20, 160);      // 設定舵機 2 的允許角度範圍為 20~160 度
iArm.setServoPin(4, 5, 6, 7);   // 設定舵機 0~3 是分別接到 iTank 的 Sv4~Sv7
```

Arduino PS2 手把函式庫 使用說明

iArm 使用的 PS2 手把函式庫 PS2X_lib 為網路上的高手所開發，相關版權及使用說明參見：
http://www.billporter.info/2010/06/05/playstation-2-controller-arduino-library-v1-0/

此函式庫在使用上並不複雜，簡單的應用可參考前面 **iArm_UsePS2** 範例的解說。進階的操作則可參考函式庫所附的範例，必要時也可直接複製其內的程式碼來使用。在 Arduino IDE 中執行『**檔案/範例/PS2X_lib/Examples/ PS2X_Example**』即可開啟此範例：



iArm 舵機規格

舵機 0~2 為 **TSB2701**, 舵機 3 (用於手爪) 為 **TBSN-K15**：

型號	TSB2701	TBSN-K15
特點	安靜	耐操
重量	60 g	
尺寸	40×20×40.5 mm	
旋轉速度	0.16sec / 60° (7.2 V)	
工作電壓	**5~7.2 V**	
空載電流	100 mA	
可控角度	**0~270°**	
齒輪材質	**金屬齒輪**	
線長	30 cm	
堵轉扭矩 (5V)	13 kg.cm	
堵轉扭矩 (7.2V)	15 kg.cm	
PWM 脈寬範圍	0.5 ms ~ 2.5 ms	
中間位置脈寬	1.5 ms	

當爪子在抓取物品時, 由於舵機壓力變大可能會發出細微聲音, 此為一般舵機都會有的正常現象。

用手臂抓舉越重的東西會越耗電, 因此建議抓取如空寶特瓶(約 30g)、空鋁罐(約 15g) 等較輕物品, 以延長電池的續航力。