

# 中華民國第 62 屆中小學科學展覽會

## 作品說明書

---

高級中等學校組 工程學(一)科

佳作

052308

當藍芽碰上 Wi-Fi—自組裝遠端操控智能船與  
無線傳輸技術監測環境水域之應用

學校名稱：高雄市立高雄女子高級中學

作者：	指導老師：
高二 林靖萍	呂雲瑞
高二 許昱唐	張函郁
高一 洪愛佳	

關鍵詞：藍芽、Wi-Fi、遠端操控

# 摘要

物聯網為未來發展的核心與趨勢，因此我們自行撰寫程式碼並結合 Arduino(嵌入式硬體平台)和酸鹼值與溫濕度感測器，組裝一台可遠端操控的智能船，搭載無線傳輸技術(藍芽、Wi-Fi)，應用於檢測水域中水質酸鹼值、水溫、氣溫及環境濕度。

再以智慧型行動裝置(手機、平板或筆電)，將本研究裝置—智慧遠端遙控船，實際於高雄地區生態池和湖泊中進行環境分析。經由量測數據相互比對後，準確獲取選定地點水域中的 pH 值與溫(濕)度。本研究具有操控便利、自由規畫路徑、水域障礙閃避與距離回傳等功能，視野範圍內自行手動控制，依據偵測任務需求或行駛環境狀態，進行航行控制的水面監測。提升監測工作的安全性，操作介面簡易便利，系統穩定成本低廉。

## 壹、前言

### 一、研究動機

生為港都的女兒，我們自小便和四周的水域有許多淵源。那可能是像〈河川證據〉裡寫的，查覺到「我之中也有河」，讓我們急於探索我們能力所能及的，對於生活中的水域有更多的了解和認知。

同時我們也了解台灣南部主要的河川都屬於嚴重污染的情況，因此我們將從生活科技課及特色選修課程、彈性課程等課程，學習到機電整合的這些概念，結合 Arduino 和 IoT 的概念開發出一台能以智慧型行動裝置遠端操控的智慧遠端遙控船，用以探測家鄉甚至全球可被觀測河流水域中的水質酸鹼值 (pH Value)、水溫及濕度。且利用其方便組裝和方便操作的特性，降低傳統政府部門檢測時造成的危險問題。

在探索水域的同時，我們也期許可以透過科技的運用，及當前物聯網(IoT)的技術，了解未來無人駕駛船隻的可能性和潛力。結合環境、科技、鄉土人文等領域，我們對於這座生長所在城市的未來和展望充滿期待，也希望能透過自己的研究和努力，付出一份心力。



圖(一) 台灣河川污染情況

(資料來源: [http://library.taiwanschoolnet.org/cyberfair2009/water/02/02\\_1.htm](http://library.taiwanschoolnet.org/cyberfair2009/water/02/02_1.htm))

## 二、研究目的

- (一) 結合 IoT( Internet of Things , 物聯網)概念，開發出以智慧型行動裝置遠端遙控船  
利用 Arduino 及各類傳感器，開發出一台方便組裝、方便操作的遠端遙控船。
- (二)以智慧型行動裝置連接船體及各嵌入式系統物件。
- (三)自製遙控船運用於河川湖泊水域之任意點水質酸鹼值、水溫及濕度檢測。  
探索自製無人載具及專業無人載具的差別及未來可能性。
- (四)比對藍芽及 Wi-Fi 兩者訊號傳輸系統之優缺。
- (五)利用 Blynk、PushingBox 與 google 表單，將遠端數據上傳至雲端監控。

## 三、文獻探討

### (一) Webduino 板及 Arduino 板的比較和運用

同為當作開發板的電路板，兩者皆用來進行計算機技術的嵌入，以完成操作系統的開發及應用。開發板的組成，主要分成輸入端、輸出端、處理器、存儲單元、通路等硬體。

Webduino 板與 Arduino 板的主要差別，為使用藍芽及 Wi-fi 為控制傳輸上網。除此之外，使用 Webduino 還可不受限於硬體燒路上傳及使用 C 系列語言編撰指令。本次研究希望借助大眾較熟悉的工具，以達到演示效果。同時，藍芽使用範圍受限於 10 公尺，也有利於進行較初步及未來發展性的研究。鑑於以上理由，我們選擇使用 Arduino 進行作業。



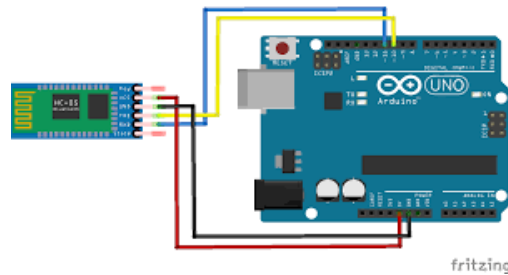
圖(二) Arduino UNO 面板

## (二)ESP8266 Wi-Fi IoT 控制晶片 及 WeMos D1 開發板

Arduino WeMos D1 為一物聯網控制面板，雖然接腳較其它控制面板少，但十分淺顯易懂，且已內建 Wi-Fi 晶片(ESP 8266)。ESP8266 是一款成本較低的 Wi-Fi 微晶片，它內置 TCP/IP 網路軟件以及微控制器功能。

## (三)HC-05 & HC-06 藍牙序列埠通訊模組

Arduino 使用藍芽進行傳輸，而藍芽裝置可分為發出端（主控）及接收端（從端）。為了方便作業，我們使用藍牙序列埠通訊模組來進行作業。市面上，常見 HC-05（主從一體）及 HC-06（主端或從端）兩種型號。考慮操作簡易性及普及性，也因為我們需要使用的 AT 指令只有少數（HC-05 可使用 30 多種指令），我們選擇使用的是 HC-06（從端）模組。



圖(三) Arduino 連接 HC-06

## (四)pH 值感測模組

pH 值為表示溶液酸鹼值的單位，為溶液中氫離子濃度的指數計算而成。因溶液中氫離子及氫氧根離子濃度有相關變動關係，故 pH 值也與 pOH 值固定相加數值為 14。此感測模組主要藉由電極於未知溶液中時產生的電位差，與電極於標準溶液中時產生的電位差做比較。因此，使用前須將其浸在 pH 值為 7 的緩衝溶液中進行校正，以避免之後量測時產生的誤差。本次研究使用該模組，以測量水質的酸鹼度，可了解水域生態及生物適居環境，希望可助未來相關無人載具研究使用相關感測模組。



圖(四) pH 感測器模組

(資料來源:<https://www.electronicclinic.com/ph-meter-arduino-ph-meter-calibration-diymore-ph-sensor-arduino-code/>)

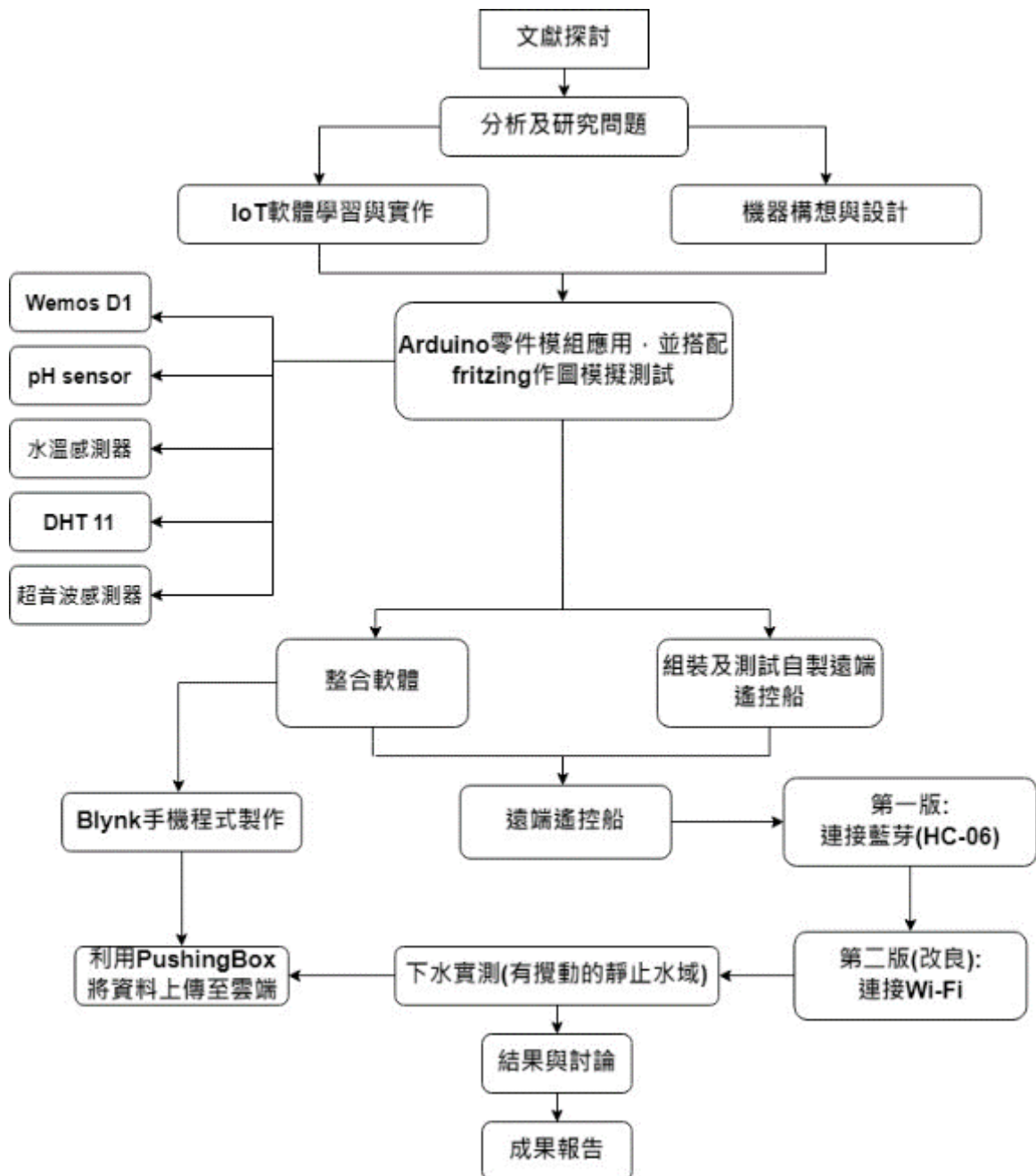
## 貳、研究設備及器材

表(一) 硬體設備

			
船體模型	380 高速直流馬達	TB6612FNG 馬達片	Arduino UNO
			
Arduino nano	HC-06 藍芽	WeMos D1 R1	pH 值感應電極
			
DHT22 溫溼度測量器	DHT11 溫溼度測量器	DS18B20 水溫感測器	TCRT500 紅外線感測器
			
HC-SR04 超音波感測器	DHT22 溫溼度測量器	HC-05 藍芽	太陽能行動電源

## 參、研究過程或方法

### 一、本研究架構



圖(五) 本研究架構流程圖

### 二、研究步驟簡述及計畫

- (一) 研究計畫擬定
- (二) 文獻參考及水域機件資料查詢
- (三) 實際設計及製作
- (四) 檢視製作成果
- (五) 確定檢測地點與方法
- (六) 實際檢測及取得數據
- (七) 整合數據並產出結論及報告



### 三、結合 IoT 概念，開發出以智慧型行動裝置遠端遙控船

#### (一)訊號傳送系統

在參照了文獻後，我們發現 Wi-Fi 及藍芽兩種訊號傳輸系統各有其優缺點，故我們做了兩版的船隻來做測試。除了訊號傳輸系統，以及其配合的開發板(Wi-Fi 系統配合 WeMosD1 板、藍芽系統配合 Arduino Nano 板)，其餘的硬件、軟件皆相同。

#### (二)船體主要的材料選定及用途

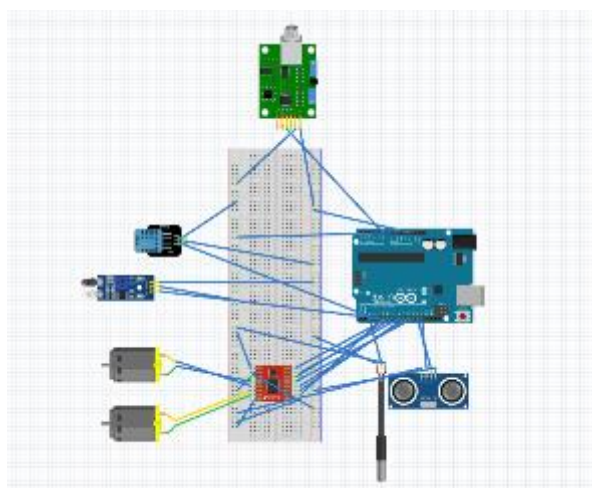
表(二) 船體主要材料

 <b>WeMos D1</b>	 <b>SR-04</b>	 <b>Arduino Nano</b>
<b>規格：</b> 電壓：2.5V~13.5V <b>用途：</b> 驅動直流馬達	<b>規格：</b> 可使用 3.3/5V <b>用途：</b> 遠端遙控及數據傳遞	<b>規格：</b> 微控制器 ATmega328 <b>用途：</b> 將指令傳遞給各個感測器
 <b>HC-06 藍芽</b>	 <b>80 高速直流馬達</b>	 <b>TB6612FNG 馬達片</b>
<b>規格：</b> 通訊距離 10 公尺 <b>用途：</b> 遠端遙控及數據傳遞	<b>規格：</b> 電壓：3V~7.4V <b>用途：</b> 以強大的馬力帶動遙控船航行	<b>規格：</b> 電壓：2.5V~13.5V <b>用途：</b> 驅動直流馬達
 <b>DS18B20 水溫感測器</b>	 <b>pH 值感應電極</b>	 <b>DHT11 溫溼度測量器</b>
<b>規格：</b> 3.0~5.5V，測量範圍 -55°C ~ +125°C <b>用途：</b> 為數位溫度感測器，在本實驗中主要測量水溫	<b>規格：</b> 檢測濃度範圍，pH 0~14 <b>用途：</b> 檢測液體酸鹼值	<b>規格：</b> 與一 8-bit 晶片結合，電壓為 3~5V <b>用途：</b> 同時測量周遭空氣環境之溫度與濕度

#### (三)電路設計及實作

在本研究中，我們的主要設計方向遵照感測器、馬達、WiFi 等裝置所需接腳原理製作，如：

1. 接地 (GND)
2. 接入電壓 (VCC、3.3V)
3. 正負極
4. 串聯及並聯



圖(六) 本研究線路配置

接電路時，除了使用杜邦線（分為雙公、公母、雙母）以外，我們大量使用焊錫槍及絕緣膠帶等工具，以穩固電路接線。以絕緣膠帶包裹裸露電線除為電路安全外，也為避免電路短路，造成不必要的設備損害。包裹時，須注意保持膠帶平整及電線完整包覆，並以兩層完整包覆以確保電路安全。我們也有手工將電線接上杜邦頭以方便拔除及接線。



圖(七) 自製杜邦頭

在進行電路設計及製作時，接腳等資訊同步於程式中，使系統正常運作。焊接時，先將兩條需焊接之電線緊緊纏繞，並以錫條沾取助焊劑使電線上塗上一層，在後續焊接時電線不致脫落；以焊槍將錫熔化後使其滴落在並凝固在電線上成火山錐形狀為最佳。此步驟需注意的點為在焊接時不可與其他條線路搞混，桌面盡量保持整潔，後再以熱縮套放置在以焊好的電路上，以工業用高溫熱風槍使其成形在電路上，電線不會裸露在外，達到安全性。



圖(八) 高溫熱風槍



#### 四、以智慧型行動裝置連接船體及各嵌入式系統物件。

##### (一)船體設計與實作

###### 1. 製作第一版(藍芽版)船體

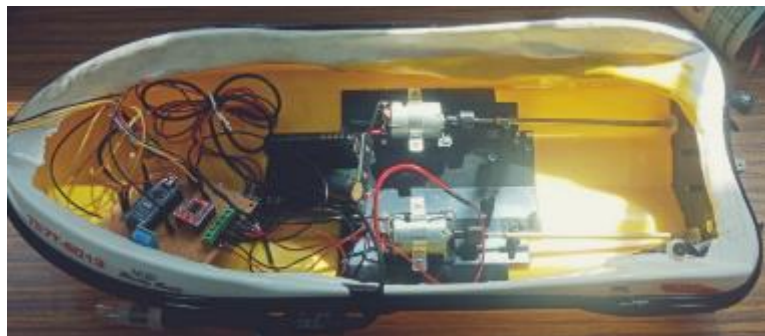
目的: 使用技術較複雜的 Wi-Fi 前先利用藍芽測試並比較其與 Wi-Fi 之效能差異

- (1) 循序確保船體與各個結構（具體順序為：Arduino nano 板、pH 值感測器、避障器、溫溼度感測器、水溫感測器、太陽能板、馬達、藍芽）連結後可單獨運作，並於校正後，取得數據。



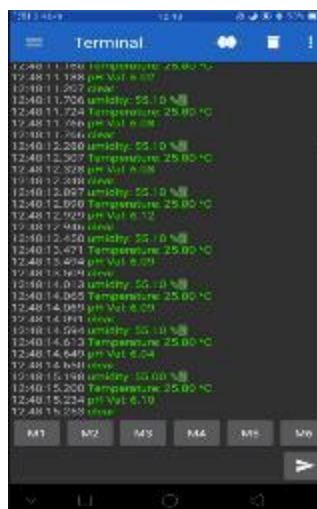
圖(九) 檢測太陽能板輸出電壓

- (2) 將所有感測器的程式與硬體結合使用，並可取得數據。



圖(十) 船體最終內部模型

- (3) 將馬達的硬體與程式結合上述步驟成果，並於水槽內測試，確保可使船體前進、後退、停止轉動、左轉、右轉。



圖(十一) 藍芽船操作介面

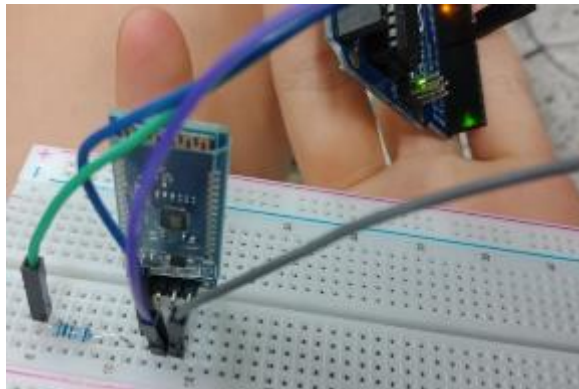
(4) 連接藍芽設備，確保船可正確接收指令運作，且接收數據可完整記錄。

(5) 具體確認步驟

循序確保船體與各個結構連結後可單獨運作，並於校正後，取得數據。以下為校正過程:

#### (i) 藍芽校正測試並上傳至 Arduino Nano 板

Arduino Nano 板的測試較簡單，首先於電腦中下載 Arduino IDE，此軟體為編寫程式處，並將草稿碼上傳至板子。再來，安裝 CH341SER(USB to COM PORT)，為連接電腦與 Arduino Uno 板硬體所需軟體。接著，下載所需程式庫（可於過程中隨時下載並增加）。最後，實際硬體連接，並上傳程式碼至板子。



圖(十二) 單獨確認藍牙執行狀況

#### (ii) pH 值感應電極的操作過程

除了須先了解其運作原理之外，也需因不同型號，而更改程式碼，才能開始使用。將電極放進裝滿蒸餾水（使用正常 pH 計確保其為 pH 值等於七）的燒杯中，並校正程式碼中的數值，以減少少部分的誤差。我們也數次測試不同酸鹼值的緩衝溶液，以確保校正成功及日後的使用。



圖(十三) 校正 pH 質感應電極

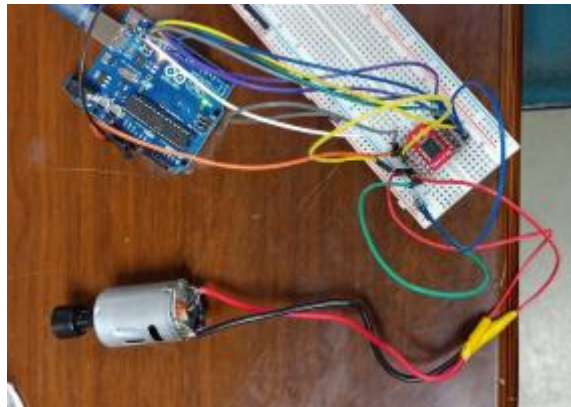
#### (iii) 溫溼度感測器的部分

我們原先是使用 DHT22 測試及組裝，但是其在運作過程有些複雜，並且佔據過多腳位，因此我們隨後便換成了 DHT11 進行測試。在測試過程中，發現數據總是要等到我們用手緊握感測器後，才会有有效且可信的數據出現在監視序列埠中，讓我們花了一些時間在研

究其運作的方法和改善情況的方法。後來我們發現，是檢測器的部分有些脫離感應的部件，導致測試數據無法準確轉換成數字顯示。這個問題在我們將其進行固定及換新後，即改善。

#### (iv) 馬達的部分

一開始直接連接乾電池（1.5 伏特）時無法轉動，後來發現是伏特不夠，我們改用 3.7 伏特的鋰電池連結，單個馬達便可正常運轉。再來將兩個馬達串聯時（此時馬達已安裝在船體上連接螺旋槳），容易發生其中一個轉動較慢的狀況，因此我們學會如何調整螺旋槳及螺帽的鬆緊，才能使兩顆馬達轉速相同，方便控制方向及前進。在之後，我們便固定使用電池盒及太陽能板驅動馬達。



圖(十四) 馬達單獨測試

#### (6) 藍芽版船體之程式

<p>第一步驟： 定義</p>	<pre> #include &lt;Wire.h&gt; #include &lt;SoftwareSerial.h&gt; #include &lt;Adafruit_GFX.h&gt; #include &lt;Adafruit_SSD1306.h&gt; #include &lt;SimpleTimer.h&gt;  //===== //Setting for humidity //===== #include "DHT.h" //溫溼度 #define DHTPIN 3 #define DHTTYPE DHT11 DHT dht(DHTPIN, DHTTYPE);  //===== //Setting for Obstacle //===== int LED = 13; // Use the onboard Uno LED int isObstaclePin = 4; // This is our input int isObstacle = HIGH; // HIGH means no obstacle         </pre>	<p>#include 表呼叫函式庫  wire.h 通訊用函式庫  SoftwareSerial.h 藍芽、序列埠用函式庫  Adafruit_SSD1306.h  Adafruit_GFX.h 核心函式庫  SimpleTimer.h 計時函式庫</p> <p>DHT.h 溫溼度計函式庫  #define 表定義腳位  DHT 表溫溼度計使用硬體</p> <p>設定避障器使用腳位及電流反應設定</p> <p>設定 pH 計計算及校正算式  宣告 pH 值使用的變數為零  宣告使用陣列所需大小</p> <p>設定馬達片/馬達所需腳位  設定正反轉</p>
---------------------	---	---

	<pre> //===== //Setting for PH Detector //===== float calibration_value = 21.34 + 0.25; int phuel = 0; unsigned long int oxgval; int buffer_arr[18], temp;  float ph_act;  //===== //Setting for PH Motor //===== #include &lt;SparkFun_TMG12.h&gt; #define AIM1 7 // 控制輸入A1 #define AIN2 6 // 控制輸入A2 #define BIN1 9 // 控制輸入B1 #define BIN2 10 // 控制輸入B2 #define PWA 5 #define PWB 11 #define STRY 8 // 方向：控制板Arduino的1腳 const int offsetA = 1; // 正旋轉設定A，逆旋轉為1假-1- const int offsetB = 1; // 正旋轉設定B，逆旋轉為1假-1-  Motor motor1 = Motor(AIN1, AIN2, PWA, offsetA, STRY); Motor motor2 = Motor(BIN1, BIN2, PWB, offsetB, STRY);  //===== //Setting for Bluetooth //===== SoftwareSerial BT(12,13); </pre>	<p>設定馬達一號及二號</p> <p>設定藍芽腳位</p>
<p>第二步驟： 初始化</p>	<pre> void setup() {   Wire.begin();   Serial.begin(9600);   //display.begin(SSD1306_SWITCHCAPVCC, 0x3C);   //display.clearDisplay();   //display.setTextColor(WHITE);    //timer.setInterval(500L, display_pHValue);    Serial.println("DHTxx test!");   dht.begin(); //初始化DHT    BT.begin(9600);    pinMode(LED, OUTPUT);   pinMode(isObstaclePin, INPUT); } </pre>	<p>撰寫 setup()，開啟時先進行運轉前設定(類似積木塊式程式編寫器的 Green Flag 被按下時所進行的動作)</p> <p>Display 相關程式可使結果出現在連接的螢幕上</p> <p>初始化溫溼度計及設定初始速率</p> <p>設定避障器輸出入</p>
<p>第三步驟： 輸入</p>	<pre> void MotorControl(char direct) {   switch(direct)   {     case 'w':       forward(motor1, motor2, 500); // 驅動雙馬達半速前進       Serial.println(direct);       delay(1000); // 延遲1秒       break;      case 's':       back(motor1, motor2, 500); // 驅動雙馬達半速後退       Serial.println(direct);       delay(1000); // 延遲1秒       break;      case 'l':       left(motor1, motor2, 500); // 左轉       Serial.println(direct);       delay(1000);       break;      case 'r':       right(motor1, motor2, 500); // 右轉       Serial.println(direct);       delay(1000);       break;      }   brake(motor1, motor2); // 停止馬達轉速 } </pre>	<p>控制馬達</p> <p>w:前進 s:後退 a:左轉 d:右轉</p> <p>半速為：500</p>

<p>第三步驟： 輸入</p>	<pre> char ReadBT() {   char cmd="x";   if(BT.available())   {     cmd=char(BT.read());     return cmd;   }   else   {     return cmd;   } } </pre>	<p>呼叫藍牙，並使藍芽回傳數值</p>
<p>第四步驟： 回傳</p>	<pre> void SerialData(float humidity, float temperature, float p_uset, bool detect_obstacle) {   //顯示到串列埠   Serial.print(" Humidity: ");   Serial.print(humidity);   Serial.print(" RH ");   Serial.print(" Temperature: ");   Serial.print(temperature);   Serial.print(" °C ");   Serial.print(" p Us: ");   Serial.print(p_uset);   Serial.println(p_uset);    if(detect_obstacle)   {     Serial.println("Warning obstacle!");   }   else   {     Serial.println("Clear");   } }  void BTData(float humidity, float temperature, float p_uset, bool detect_obstacle) {   //顯示到藍牙埠   BT.print(" Humidity: ");   BT.print(humidity);   BT.print(" RH ");   BT.print(" Temperature: ");   BT.print(temperature);   BT.print(" °C ");   BT.print(" p Us: ");   BT.print(p_uset);   BT.println(p_uset);   if(detect_obstacle)   {     BT.println("Warning obstacle!");   }   else   {     BT.println("Clear");   } } </pre>	<p>使 Arduino 內建序列埠上的數值能回傳至手機手機上</p>

## 2. 製作第二版(Wi-Fi 版)船體

### 目的: 增進船隻遙控距離與傳輸速度

使用 WeMos D1 為主要開發版(已經嵌入 Wi-Fi ESP8266)了。接下來依循上述藍芽版的步驟將感測器與 WeMos D1 結合。並利用 Blynk、PushingBox

- (1) 將各個硬體設備與 WeMos D1 結合，相較於第一版我們多加上了水溫感測器與超音波感測器，合成後分別測試各個硬體運作是否正常





圖(十五)第二版船隻內部結構圖

(2) 為確保船隻防水，我們先塗上多層防水漆，並使用黑色塑膠板蓋之其上，之後再噴上多層防水漆，並於尾端綁上尼龍繩，避免緊急狀況發生。



圖(十六)本船隻最終樣貌



圖(十七)本船隻 3-D 立體圖



### (3) 撰寫(Wi-Fi 版)船體之程式

<p>第一步驟： 定義</p>	<pre> #include "ESP8266WiFi.h" #include "DHT.h" //含入DHT溫濕度 #include &lt;BlynkSimpleEsp8266.h&gt; #include &lt;Wire.h&gt; #include &lt;SoftwareSerial.h&gt; #include &lt;Adafruit_GFX.h&gt; #include &lt;Adafruit_SSD1306.h&gt; #include &lt;SimpleTimer.h&gt; #include &lt;SparkFun_TB6612.h&gt; #include &lt;Wire.h&gt; #include &lt;SoftwareSerial.h&gt; #include &lt;Adafruit_GFX.h&gt; #include &lt;Adafruit_SSD1306.h&gt; #include &lt;SimpleTimer.h&gt; #include &lt;SparkFun_TB6612.h&gt;  // Template ID, Device Name and Auth Token are provided by the Blynk.Cloud // See the Device Info tab, or Template settings #define BLYNK_TEMPLATE_ID "TMPL05118QX0" #define BLYNK_DEVICE_NAME "Quickstart Device" #define BLYNK_AUTH_TOKEN "Gg5gbD-MWRJSUQL4bQdGd-_MOJB_tdm"  #define DHTPIN 5 //DHT11接於GPIO5 #define DHTTYPE DHT11 //選擇型號DHT11 DHT dht(DHTPIN, DHTTYPE, 15); //建立DHT物件         </pre>	<p>定義各個程式 並啟動 Blynk 放上 Blynk 啟動碼</p>
<p>第二步驟： 連接 Wi-Fi</p>	<pre> // Comment this out to disable prints and save space #define BLYNK_PRINT Serial  float humidity, temperature; #include &lt;SPS2600Pi.h&gt; #include &lt;BlynkSimpleEsp8266.h&gt;  char auth[] = "Gg5gbD-MWRJSUQL4bQdGd-_MOJB_tdm";  // Your Wi-Fi credentials. // Set password to "" for open networks. char ssid[] = "so_suptel"; char pass[] = "1q5@725";  BlynkTimer timer;  // This function is called every time the Virtual Pin 8 state changes. BLYNK_WRITE(V8) {   // Set incoming value from pin 8 to a variable   int value = param.asInt();    // Update state   Blynk.virtualWrite(1, value); }  // This function is called every time the device is connected to the Blynk.Cloud BLYNK_CONNECTED() {   // Change Web Link banner message to "Congratulations!"   Blynk.setProperty("v3", "htmlPageBanner", "https://static-image.my32bit.digitaloceanspaces.com/general/via/congratulations.png");   Blynk.setProperty("v3", "htmlPageV1", "https://static-image.my32bit.digitaloceanspaces.com/general/via/congratulations.png");   Blynk.setProperty("v3", "html", "https://docs.blynk.io/en/getting-started/what-do-i-need-to-blynk-from-a-raspberry-pi-device-as-a-server"); }  // This function sends Arduino's active every second to Virtual Pin 1.         </pre>	<p>輸入密碼後連上 Wi-Fi 並透過 Blynk 連接</p>

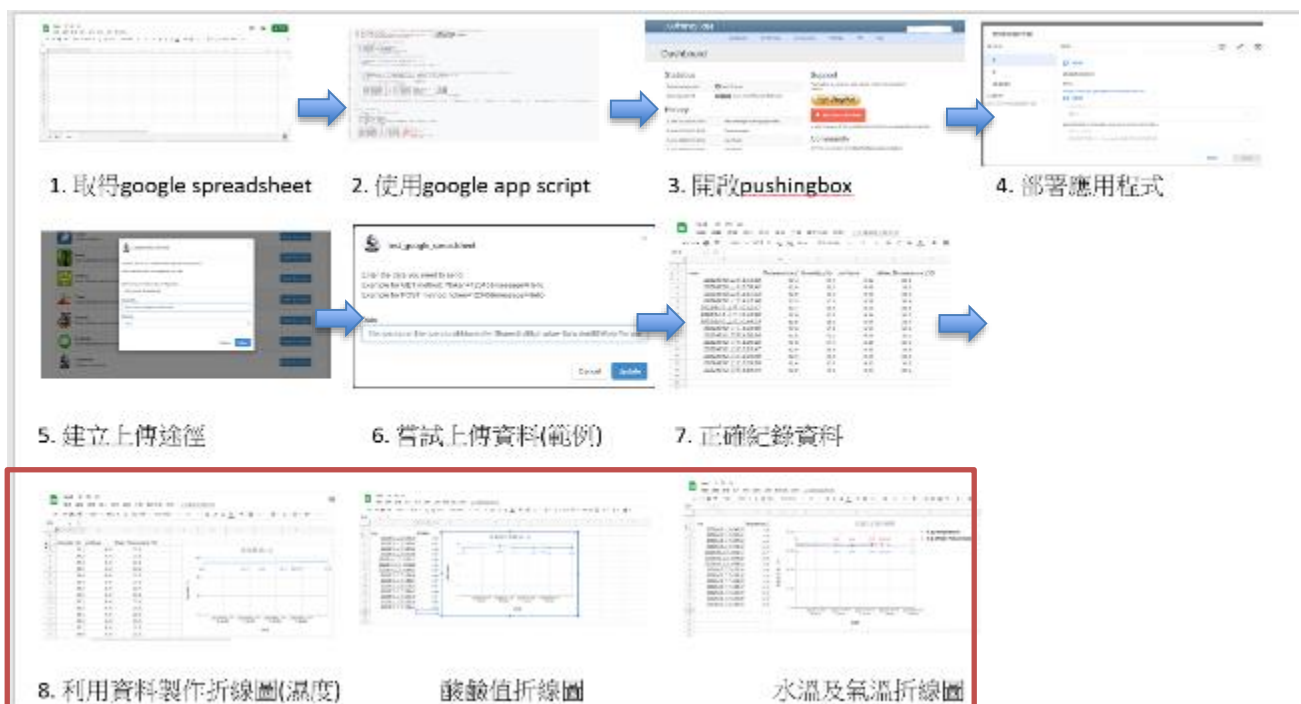
<p>第三步驟： 輸入</p>	<pre>float PHDetectC() {     for(int i=0;i&lt;10;i++)     {         buffer_arr[i]=analogRead(A0);         //delay(30);     }     for(int i=0;i&lt;9;i++)     {         for(int j=i+1;j&lt;10;j++)         {             if(buffer_arr[i]&gt;buffer_arr[j])             {                 temp=buffer_arr[i];                 buffer_arr[i]=buffer_arr[j];                 buffer_arr[j]=temp;             }         }     }     avgval=0;     for(int i=2;i&lt;8;i++)     {         avgval+=buffer_arr[i];     }      float volt=(float)avgval*5.0/1024/6;     ph_act = -5.78 * volt + calibration.value;      return ph_act; }</pre>	<p>設定將 pH 檢測計的檢測到的數值輸入陣列</p> <p>經過計算後，取得正確的數值並將其返回(return)</p>
<p>第三步驟： 輸入</p>	<pre>//===== //Setting for 超音波感測器 //===== int trigPin = 11; // 超音波感測器 Trig 腳接 Arduino pin 11 int echoPin = 12; // 超音波感測器 Echo 腳接 Arduino pin 12 long duration, cm; // 宣告計算距離時，需要用的兩個變數</pre>	<p>設定超一波感測器的運作</p>
<p>第三步驟： 輸入</p>	<pre>void MotorControl(char direct) {     switch(direct)     {         case 'w':             forward(motor1, motor2, 500); // 驅動雙馬達半速前進             Serial.println(direct);             delay(1000); // 持續1秒             break;          case 's':             back(motor1, motor2, -500); // 驅動雙馬達半速後退             Serial.println(direct);             delay(1000); // 持續1秒             break;          case 'a':             left(motor1, motor2, 500); // 左轉             Serial.println(direct);             delay(1000);             break;          case 'd':             right(motor1, motor2, 500); // 右轉             Serial.println(direct);             delay(1000);             break;      }     brake(motor1, motor2); // 停止兩個馬達 }</pre>	<p>控制馬達</p> <p>w:前進 s:後退 a:左轉 d:右轉</p> <p>半速為：500</p>

<p>第四步驟： 執行</p>	<pre>void loop() {   //timer.run();   float ph_act=PHDetect();    float humidity = dht.readHumidity(); //取得濕度   float temperature = dht.readTemperature(); //取得溫度C    bool detect_obstacle=DetectObstacle();    char keyboard_read_serial=Serial.read();   char keyboard_read_bt=ReadBT();   //Serial.println(keyboard_read_bt);   MotorControl(keyboard_read_bt);    //SerialShow(humidity,temperature,ph_act,detect_obstacle);   BtShow(humidity,temperature,ph_act,detect_obstacle);    delay(500); }</pre>	<p>撰寫 loop()</p> <p>引入 DHT11、TCRT5000、TB6612FNG、HC-06、pH sensor 的程式</p> <p>宣告所需變數</p> <p>宣告藍芽需傳回濕度、溫度、pH值、前方有無障礙阻擋</p>
<p>第五步驟： 回傳</p>	<pre>void SerialShow(float humidity, float temperature, float ph_act,bool detect_obstacle) {   //顯示在藍芽裝置裡   Serial.print("humidity: ");   Serial.print(humidity);   Serial.println(" %");   Serial.print("Temperature: ");   Serial.print(temperature);   Serial.println(" °C ");   Serial.print("pH Val: ");   Serial.println(ph_act);    if(detect_obstacle)   {     Serial.println("Meeting obstacle!");   }   else   {     Serial.println("clear");   } }</pre>	<p>使 Arduino 內建序列埠上的數值能回傳至手機手機上</p>

(4) 連接 Blynk 並製作使用者介面



(5) 利用 PushingBox 將資料上傳至雲端



## 肆、研究結果

### 一、結合 IoT 概念，開發出以智慧型行動裝置遠端遙控船

將 Arduino 的各個結構先確認可以單獨操作後，我們將所有設備連接至遙控船上使其能正常運作，再將程式碼以及模組本身連接再一起形成一個整體以驅動。

我們以高雄大學的生態池為訓練場地，在岸邊水深較淺處確定馬達已經能夠正常運作及遙控船不會沉下去。



圖(十八) 高雄大學生態池

加裝超音波感測器來監測船體，是否太過接近障礙物，在實際操作時除了輸入指令的人員，也有另一人員負責監測船體目前位置及狀況，以免船體發生任何意外。

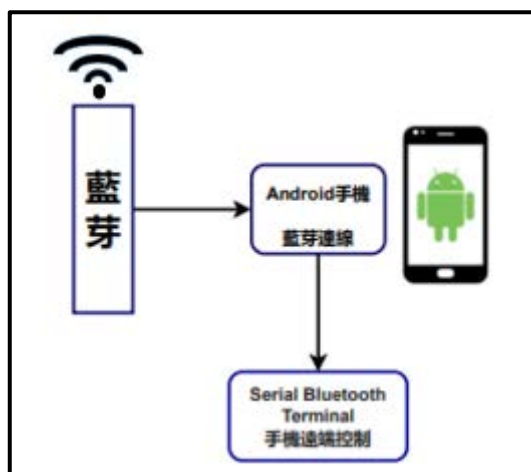


圖(十九) 船體與馬達執行成功

### 二、以智慧型行動裝置連接船體及各嵌入式系統物件。

- ◆ 第一版(藍芽版)使用 HC-06 藍芽透過智慧型行動裝置遠端操控遙控船





圖(二十) 藍芽連接及傳送示意圖

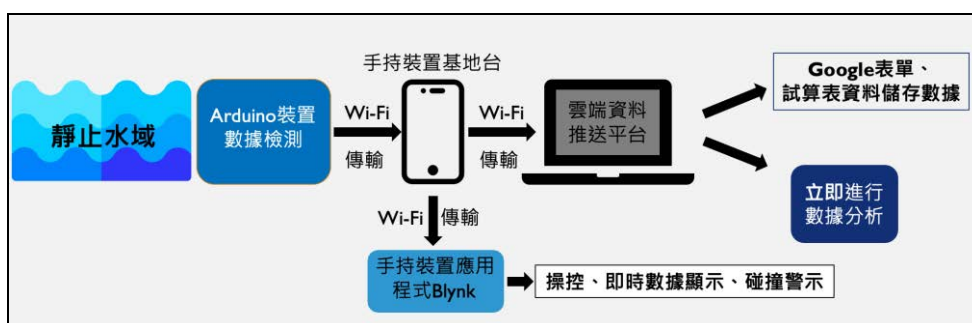
使用現存之手機程式 Serial Bluetooth Terminal，利用 HC-06 藍芽連接智慧型行動裝置，在程式輸入進藍芽後，於手機端輸入指令給藍芽，再使藍芽輸出指令給遙控船（輸入 w、a、s、d 分別使會船前進、左轉、後退、右轉），從而使船驅動。



圖(二十一) Serial Bluetooth Terminal

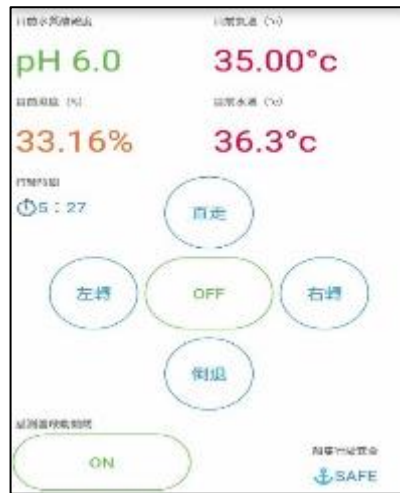
### ◆ 製作第二版(Wi-Fi 版)船體

1.我們使用 WeMos D1 連接智慧型行動裝置，以遠端操控遙控船



圖(二十二) Wi-Fi 傳送示意圖





圖(二十三) 操控介面

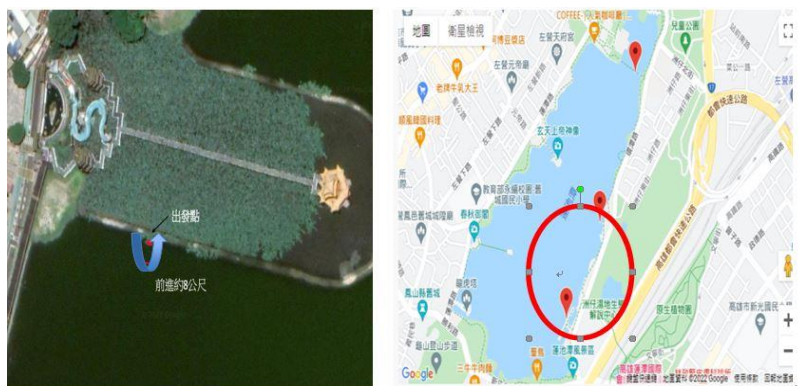
2.將資料立即藉由 PushingBox 傳回雲端，並進行數據分析

Time	Temperature (°C)	Humidity (%)	pHValue	Water Temperature (°C)
2022/6/10 上午 8:43:28	32.4	55.1	9.41	33.2
2022/6/10 上午 8:50:42	32.4	55.1	6.09	33.2
2022/6/10 上午 8:57:33	32.6	55.1	6.09	33.3
2022/6/10 上午 9:12:46	32.5	55.1	6.09	33.4
2022/6/10 上午 10:32:17	32.7	55.1	6.09	33.5
2022/6/10 上午 10:45:26	32.9	55.1	6.09	33.6
2022/6/10 上午 10:49:03	32.8	55.1	6.09	33.7
2022/6/12 下午 4:23:59	32.6	55.1	6.09	33.4
2022/6/12 下午 4:56:34	32.5	55.1	6.09	33.3
2022/6/12 下午 5:09:25	32.6	55.1	6.09	33.2
2022/6/12 下午 5:25:47	32.5	55.1	6.09	33.4
2022/6/12 下午 5:34:19	32.3	55.1	6.09	33.3
2022/6/12 下午 5:39:29	32.4	55.1	6.09	33.2
2022/6/12 下午 5:59:44	32.2	55.1	6.09	33.2

圖(二十四) 數據 Google 表格

### 三、自製遙控船運用於河川湖泊水域之任意點水質酸鹼值、水溫及濕度檢測。




本實驗自製船體，選擇於高雄市蓮池潭為監測地點，圖(二十一)為觀測地點，蓮池潭為大型人工湖，湖面面積約 42 公頃且水流平穩，故為理想實測地點。我們將自製遠端遙控船放入水中進行定點水域酸鹼值 (pH Value)、水溫及濕度測量。



圖(二十五) 蓮池潭地圖及測量範圍(紅框處)

將船體放入水中，再確保馬達運作正常後，即開始進入定點監測，我們以岸邊為出發點。因為藍牙偵測極限為 10 公尺，故我們使船大約前進 8 公尺(目測)後即迴轉回岸邊，來回 5~6 趟，每趟 8~10 分鐘，約 1 個小時左右完成測量，如下表(三)所示。

表(三) 實地遙控測量

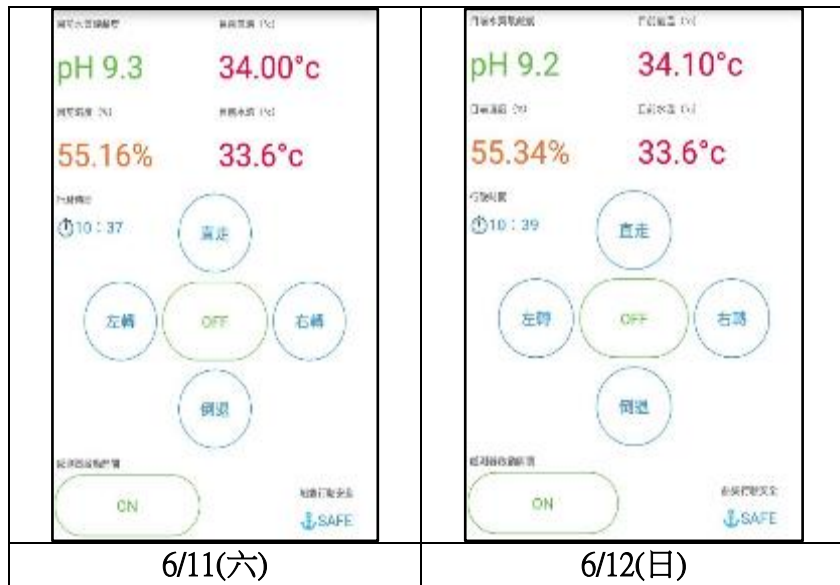
		
施放船體	啟動馬達	迴轉歸岸

在遙控船船移動的同時，我們時時刻刻在偵測水體的酸鹼值、水溫及濕度等，以下為我們檢測出的數值回傳回手機的畫面，個別連續兩天施測(藍芽和 Wi-Fi 船)如下表(四)和(五):

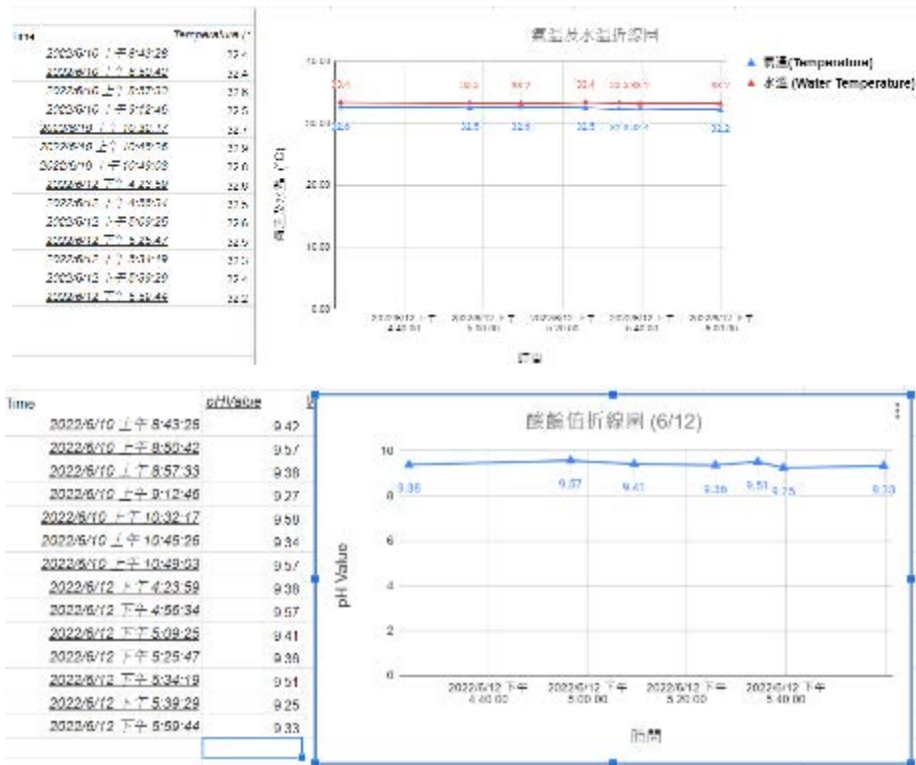
表(四) 手機藍芽操控介面

	
3/11(六)	3/12(日)

表(五)Wi-Fi 版手機操作介面



將監控得的數據上傳 PushingBox，回傳至 Excel 立即進行數據分析



圖(二十六、二十七) 監控數據折線圖

自製遙控船將其測量數值藉由 Wi-Fi 回傳至手機，測量顯示：

1. 水域酸鹼值 (pH Value) 範圍:9.3~9.6
2. 水域溫度範圍:33.16~34.1 度
3. 空氣濕度範圍:50.6~56.6%

表(六) 高雄市環保局水域酸鹼(pH 值)

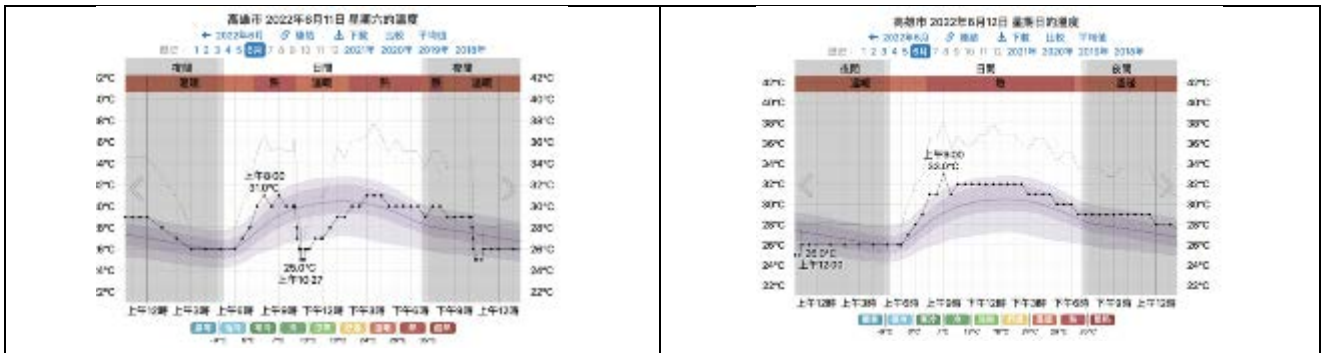
河川名稱	監測站名稱	距匯流口距離(公里)	監測站位置(地址)	監測站座標	
				經度	緯度
蓮池潭	遊艇碼頭	-	左營區環潭路	120.296650	22.680234
	人行拱橋	-	左營區環潭路	120.299560	22.686730
	龍船船塢	-	左營區環潭路	120.298067	22.682793

測站名稱	採樣日期	採樣時間	河川污染指數 RPI	是否符合 戊類	水溫 °C	氧離子濃度指數 pH值
人行拱橋	2021/06/03	11:34	--	否	33.6	9.4
龍船船塢	2021/06/03	11:24	--	否	33.4	9.7
遊艇碼頭	2021/06/03	11:15	--	否	33.6	9.6

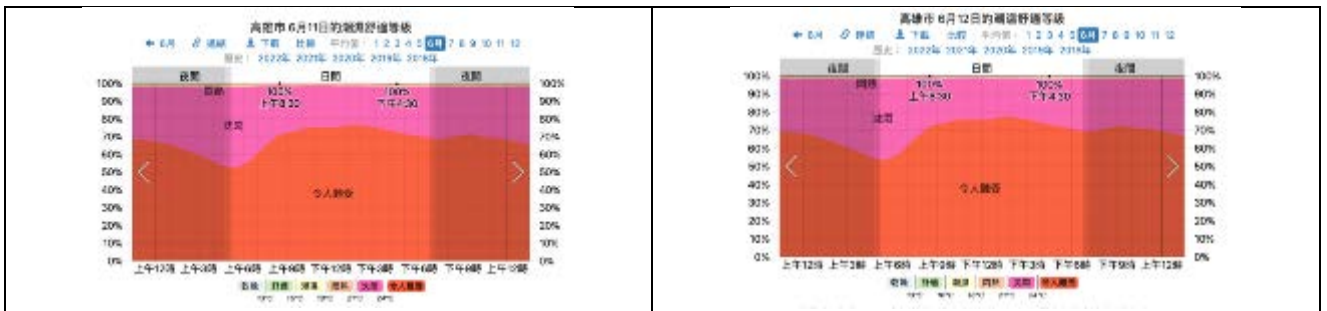
圖(二十八) 高雄市環保局水域酸鹼(pH 值)

資料來源:<https://lab.ksepb.kcg.gov.tw/kaqm/tw/b0401a.aspx>

表(七) 中央氣象局資訊(空氣溫度與相對濕度)



資料來源:<https://reurl.cc/Qjg8zq>



資料來源:<https://reurl.cc/Ep0Okv>



將自行測量數值與高市環保局與中央氣象局監控數值進行比較，由下表(九)所示，若考慮到其他天氣因素(量測地點、時間差異、日照強度、風(雲)量等因素)，我們可歸納兩個測量方式(監測站和自製遙控船)的差異不大。因此本研究的自製藍牙遙控船應可替代日常水域的測量方式。

表(八) 量測項目相互比對值

測量項目	6/11(六)	監測站資料	6/12(日)	監測站資料
水域酸鹼(pH 值)	9.2~9.5	9.4	9.3~9.5	9.6
水溫(°C)	33.1°C	33.6 °C	32.9 °C	33.6 °C
空氣濕度(%)	55.16~56.8%	52%	55.16~57.10%	50%

## 伍、討論

### 一、硬體設備

在製作的過程中，為了改變船體的裝載及機件安排，我們進行了多次的測試，在比較太陽能及其穩定性後，最終決定使用開發出行動電源與二次能源提供能源使船隻正常運作。。



圖(二十九) 太陽能/行動電源藍芽遙控船

### 二、程式碼的撰寫困難

在設計程式碼時，因為有結合很多硬體設備的原因，因此我們是先編寫使各個硬體運作的程式，再慢慢結合在一起。再結合在一起的過程中，也須依循編寫程式的原則，除了確保正確運作之外，也是為了減少上傳的時間及使用的記憶體，增加智慧藍芽遠端遙控船的運作效率。

### 三、製作船隻的初衷

除了可以盡自己的努力，偵測環境週遭水域，也藉由我們試圖使用一般五金行或是網路可購到的器材及機件的嘗試，搭配未來趨勢-IoT 及 Arduino 的應用，了解如何自製運輸工具進行科學研究及探討進行科普研究的相關可能性，並試圖降低傳統檢測造成的不便與危險。

### 四、水質檢測的地點

在靜止水域的選擇上，我們選擇已經有政府檢測數據，且水面可進入的蓮池潭，除了希望藉由先前的數據應證我們檢測設備的正確外，也是因為蓮池潭為大型人工湖(或可應用於生態池、灌溉埤塘、養殖魚塭等)，湖面面積約 42 公頃且水流平穩，故為理想實測地點。



圖(二十八) 蓮池潭實際測量

## 陸、結論

### 一、結合 IoT( Internet of Things，物聯網)概念，開發出以智慧型行動裝置遠端遙控船

本研究最終使用 WeMos D1，自製了杜邦線和進行焊錫，結合不同感測器，設計一艘可以於智慧型行動裝置遠端操控的藍芽操控船，除了方便操作，可自行組裝，素材容易取得外，兼備科技運用與環境關懷的機械，促進傳統檢測方式的改良，大幅提升傳統測量時的安全性。

### 二、以智慧型行動裝置連接船體及各嵌入式系統物件。

自行撰寫程式碼並以智慧型行動裝置(手機、平板或筆電)遠端操控船體，應用於檢測水域中水質酸鹼值(pH Value)、水域溫度、氣溫及濕度，藉由 Wi-Fi 回傳技術，成功打造出一臺智慧遠端遙控船。



### 三、自製遙控船運用於河川湖泊水域之任意點水質酸鹼值、水溫及濕度檢測。

將其實際放置於環境中附近的水域，進行水域酸鹼值（pH Value）、水溫及濕度的檢測，實際於高雄地區生態池和湖泊中進行環境分析。經由量測數據相互比對後，準確獲取選定地點水域中的 pH 值與溫濕度。

本研究具有操控便利、自主路徑規劃、障礙閃避、障礙距離測量等功能，在目測範圍內由岸上人員遠距控制，或自動依據任務需求及行駛環境狀態，進行航行控制的水面監測。其優點為降低人為作業風險，提升檢測工作的安全性，操作更為簡易便利，一般民眾也可直接自行測量。我們還將其利用程式控制的方式，得以實現無人裝置於水上的應用及檢測，提供未來無人載具之研究及製作上之參考及試驗示範。

### 四、比較Wi-Fi 與藍芽效能差異

本實驗先後做出兩版船隻，第一版為藍牙版，第二版為 Wi-Fi 版。

(一)傳輸速度：藍芽傳輸速度 720kbps(千位元/秒)，Wi-Fi 則為 11mbps(百萬位元/秒)，速度上已有明顯差距。但藍芽技術可取代纜線，以點對點的方式連接不同裝置。WiFi 主要用來連接整個網路。

(二)安全性比較：藍芽的安全性比 WiFi 來得高，因為藍芽的傳輸距離就比 Wi-Fi(802.11b)短，此外藍芽還提供兩層密碼保護，而 WiFi 的安全風險則是一旦有人取得部分存取權限，就能進入整個網路。

(三)不同裝置連接：藍芽單一裝置可同時連上七台裝置，因此可以輕鬆連上別的裝置，或裝置之間作切換。WiFi 則較為複雜，需要具備一般實體網路的管理能力才行。

(四)裝置耗電量：藍芽的功率需求比 WiFi 低，因此裝置體積也較小，適用於消費電子產品(智慧型手機或平板)。

(五)硬體相容性：藍芽與 WiFi 都共用相同的頻率，因此有可能會相互干擾，基於一些技術因素藍芽比較容易被 WiFi 干擾到，WiFi 反倒不會。

若傳輸速度不是問題的話，藍芽很適合用來連結單一裝置，最適合的應用包括在 9-18 公尺的範圍內共享同步化、將手機當作數據機，甚至作軟體程式的相互無線連結。藍芽並不能取代所有的連接線，例如 USB 連結；它也不適合連接高頻寬的裝置，如外接磁碟機、數位攝影機或電腦等。藍芽簡易好用，而 WiFi 則適合用來當作與電腦之間的連接以及無線上網。

表(九) Wi-Fi 與藍芽效能差異

	藍芽	Wi-Fi
型號	HC-06	ESP 8266
傳輸距離	10 公尺內	60~100公尺
最高傳輸速度	24 Mbit/s	300 Mbit/s

因此後續使用者可搭配欲測量環境水域的特性(水面靜止循環或開放流動)，水域面積寬廣大小(大型水庫或小型埤塘)，將此技術應用於大學生態池、養殖魚塢、灌溉埤塘或湖泊水庫或水域河流的水質監測。

## 5、 本研究系統的特點

**1.傳輸訊號方面：**有別於藍牙與ZigBee，我們使用傳輸速度最快、普及應用度高的Wi-Fi作為傳輸工具，以提升航行範圍與傳輸速度。

**2.船體設計方面：**我們開發出一台基礎版船體，並以駕駛平穩、沉浮穩定方向為主要考量。

**3.硬體設施方面：**本計畫參考環保署之測量數據，選取最常測量的三樣數據，水質酸鹼值（pH

Value）、水溫及濕度，進行系統開發並結合Arduino安裝pH值感應電極、DS18B20水溫感測模組、DHT11 溫溼度測量器。

**4.安全防護方面：**

(1)防水系統：本船體以塑膠蓋將硬體設備單獨放置，密封後噴上防水噴霧，使防水系統最大化。

(2)運行時保護系統：為避免船隻運行中碰撞障礙物，本船隻安裝

HC-SR04

超音波感測器與紅外線避障感測器。除此之外，為避免最糟糕情況，本船隻於船身綁上尼龍繩降低損失。

**5.能源供給方面：**本計劃在開發設計船時嘗試使用太陽能板，惟效能轉換不佳且不小心觸水後又易引發安全問題，因此本正式船體中打算全面使用可循環利用的二次能源（8顆 3.7 V 鋰電池）與行動電源，且經過計算與測試船隻至少可保持運行2小時以上。

6.控制方面：本船體以智慧型行動裝置(手機、平板或筆電)遠端操控船體，並藉由Wi-Fi回傳，將數據回傳至行動裝置端。

6、 本研究成本

表(十) Wi-Fi 船隻成本表

器材	數量	價格
WeMos D1	1 個	126
pH 值感應電極	1 個	780
DHT11 溫溼度測量器	1 個	40
HC-SR04 超音波測距	1 個	43
DS18B20 水溫感測模組	1 個	231
TB6612FNG 馬達片	1 個	40
TCRT5000 紅外線感測器	1 個	40
杜邦線	1 包	20
船體	1 組	300
共 11 項		1620

表(十一) 藍芽船隻成本表

器材	數量	價格
<b>Arduino nano</b>	<b>1個</b>	<b>200</b>
<b>pH值感應電極</b>	<b>1個</b>	<b>780</b>
<b>DHT11 溫溼度測量器</b>	<b>1個</b>	<b>40</b>
<b>DHT22 溫溼度測量器</b>	<b>1個</b>	<b>45</b>
<b>HC-06 藍芽</b>	<b>1個</b>	<b>100</b>
<b>HC-05 藍芽</b>	<b>1個</b>	<b>100</b>
<b>TB6612FNG馬達片</b>	<b>1個</b>	<b>40</b>
太陽能板	1個	400
TCRT5000紅外線感測器	1個	40
杜邦線	1包	20
船體	1組	500
共11項		2265

## 七、對未來研究的期許

我們藉由較不同系統裝置和探測器的使用，了解無人載具的可行性與未來可進行研究的方向。在無人智能船舶，則需考慮的很大一點困難為訊號資料的傳送，這次因為距離及在水上進行實驗，因此較無大礙，但若是進行水下的探測或到更遠的距離，勢必要考慮影像的傳遞或信號的強弱等問題。

在功能方面，除了簡化其在運作時的自行控制及船體運轉的能力，也希望讓其在環境水質的檢測上有更進一步的深入，例如水質採樣或搭配機器手臂。未來可應用在於海洋環境研究上，如水域生物覓食紀錄、水波傳遞能量、海浪強度與潮汐觀測等。

## 柒、參考資料及其他

### 一、github 程式庫

- (一) <https://github.com/esp8266/Arduino/blob/master/libraries/Wire/Wire.h>
- (二) <https://github.com/esp8266/Basic/blob/master/libraries/espsoftwareserial/SoftwareSerial.h>
- (三) [https://github.com/adafruit/Adafruit\\_SSD1306](https://github.com/adafruit/Adafruit_SSD1306)
- (四) <https://github.com/adafruit/Adafruit-GFX-Library>
- (五) <https://github.com/jfturcot/SimpleTimer>

### 二、參考文獻

- 1.趙英傑，超圖解 Arduino 互動設計入門(第二版)，旗標出版股份有限公司，2015。
- 2.葉難，Arduino 輕鬆入門：範例分析與實作設計，博碩文化出版，2014。
- 3.Mark Geddes，CAVEDU 教育團隊-曾吉弘譯，Arduino 自作專案大百科，碁峰資訊，2015。
- 4.楊明豐，Arduino 最佳入門與應用：打造互動設計輕鬆學，碁峰資訊，2015。
- 5.M. T. Lazarescu, "Design of a WSN platform for long-term environmental monitoring for IoT applications," IEEE Journal on emerging and selected topics in circuits and systems, vol. 3, pp. 45-54, 2013.
- 6.A. S. Ali, Z. Zanzinger, D. Debose, and B. Stephens, "Open Source Building Science Sensors (OSBSS): A low-cost Arduino-based platform for long-term indoor environmental data collection," Building and Environment, vol. 100, pp. 114-126, 2016

### 三、網路資料

- 1.行政院環保署，環保法規水污染防治法，Available： <http://goo.gl/jZtUNr>， Accessed：2016。
- 2.行政院環保署，環境資源資料庫 重要河川污染情形， Available： <http://goo.gl/AFoHkE>， Accessed：2016。
- 3 台灣物聯科技，Arduino PH 值檢測 PH 值+溫度器感測器模組， Available： <https://goo.gl/EaHVkc>， Accessed：2016。

## 【評語】 052308

1. 作品利用物聯網概念，組裝一台可遠端操控的智能船，可用於檢測水質酸鹼值、水溫、氣溫及環境濕度。
2. 文中提到水域自主規劃路徑與障礙閃避的功能，針對實際運作原理和情況缺乏說明。
3. 若針對環境監測，尤其是水域，每一次量測建議在同樣的地點，此時，以 GPS 進行船體定位和導航（每次依循同個路徑在固定地點收集數據）相比遙控更具有效益。
4. 第 5 頁中所列紅外線感測器的用途為？超音波避障在水域中用途不大，多為岸邊停泊時需要。但若為手機人在旁遙控，有無超音波提供訊息的效益有限。
5. 作品可多加強無人船動力系統及控制系統如何整合的說明。
6. 作品可加強說明無人船採測水質酸鹼值及水域溫度的機制。

## 作品簡報



當藍芽碰上 WI-FI —  
自組裝遠端操控智能船與無線傳輸  
技術監測環境水域之應用

科別：工程學科(一)(含電子、電機、機械)

組別：高級中等學校組

# 當藍芽碰上 Wi-Fi—自組裝遠端操控智能船與無線傳輸技術監測環境水域之應用

## 壹、船隻特色

本船隻俯視圖



本船隻前視圖



安全性、便利性、  
無人機概念

1. 障礙物距離偵測搭配
2. 方邊操作、使用簡易
3. 設有防水保護系統
4. 節省時間並提升效率

價格低廉、廣泛運用

1. 本Wi-Fi版船僅需1620元
2. 可運用於任意點之靜止水域檢測

船隻  
特點

Wi-Fi 擴大遙控距離

Wi-Fi版船體遙控範圍  
介於60~100公尺

結合多組感測器

結合pH值感應電極、DS18B20水溫感測  
模組、DHT11 溫溼度測量器。

遠端遙控與快速測量

簡易遠端操作並快速檢測

本技術應用於生態池、灌溉埤塘、養殖魚塭，並以Wi-Fi增加資料傳送距離與操控範圍。

## 貳、研究目的

- (一) 結合IoT(Internet of Things, 物聯網)概念, 開發出智慧型行動裝置遠端遙控船
- (二) 以智慧型行動裝置藉由Wi-Fi與藍芽連接船體及各嵌入式系統物件。
- (三) 自製遙控船運用於河川湖泊水域之任意點水質酸鹼值、水溫及濕度檢測。
- (四) 比對藍芽及Wi-Fi兩者訊號傳輸系統之優缺。
- (五) 利用Blynk、PushingBox與google表單, 將遠端數據上傳至雲端監控。

### 本船隻內部結構圖

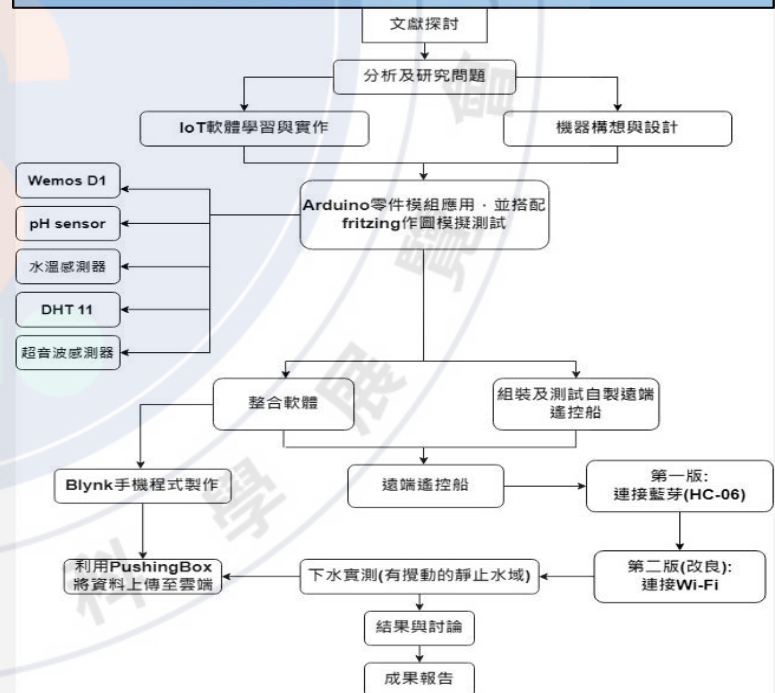


在內部加上塑膠保護套避免水沾濕儀器

### 本船隻側視圖



### 本實驗設計架構圖





# 第一版：藍芽版船隻

## 參、研究過程及方法

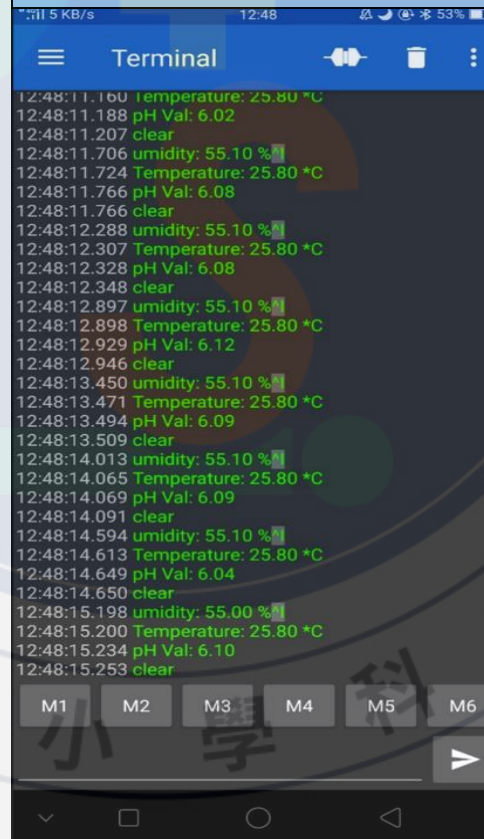
使用藍芽連接進行遙控及數據監測，並且在使用較複雜的Wi-Fi前測試並做比較

- 確認各個結構可單獨運作將軟體及硬體整合，並實際放置於生態池中取進行測量
- ✓ 依序：Nano 板、pH值感測器、避障器、溫溼度感測器、馬達、藍芽
- ✓ Terminal 操作：前進 (w)、後退 (s)、左轉 (a)、右轉 (d)

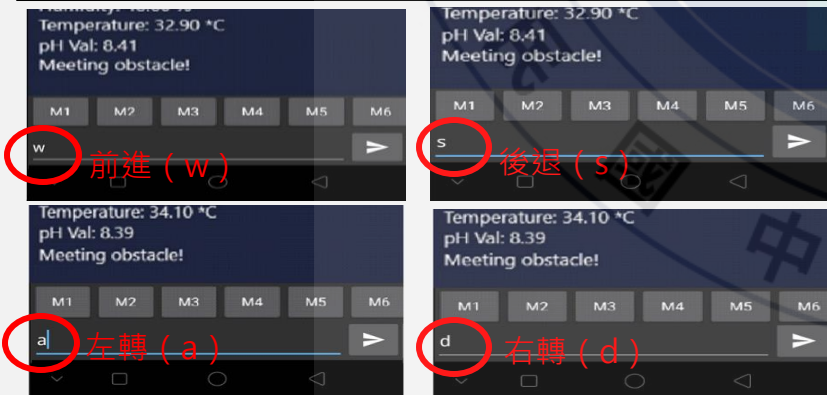
### 藍芽版船隻內部構造圖



### 操作介面



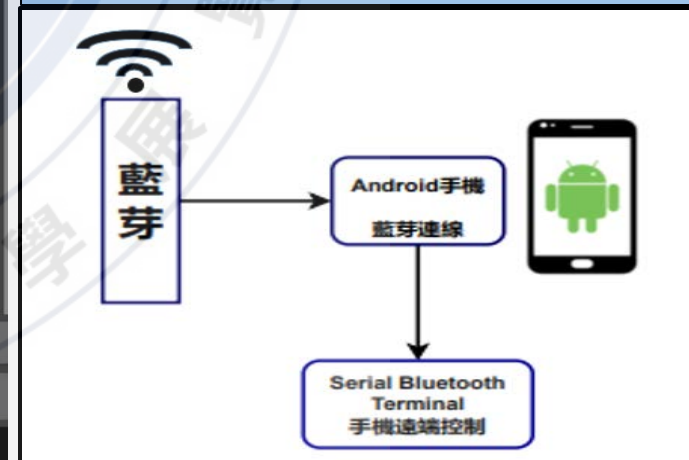
### Terminal 操作：控制方向



### 藍芽：問題、缺點

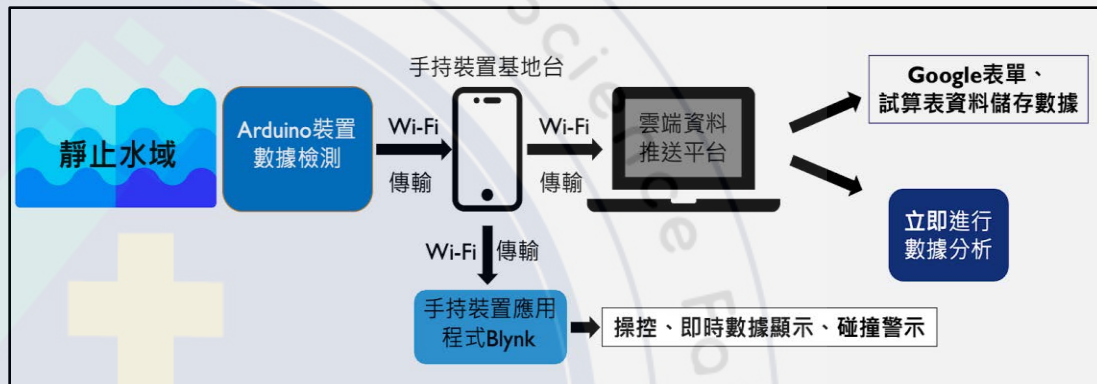
1. 傳輸距離僅能10公尺之內
2. 安全性較高
3. 訊號不易遭干擾

### 藍芽連接及傳送示意圖

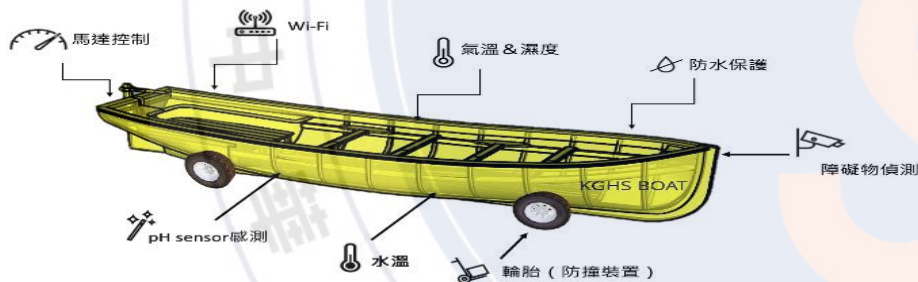


目的: 改良並提升船隻遙控距離與傳輸速度，新增水溫感測及超音波測距避障功能

使用WeMos D1為主要開發板 (含Wi-Fi ESP8266)。依循上述藍芽版的步驟將感測器與WeMos D1結合。並利用Blynk、PushingBox將測得數據同步回傳至雲端並上傳至excel進行數據分析



### 本船隻3-D立體圖



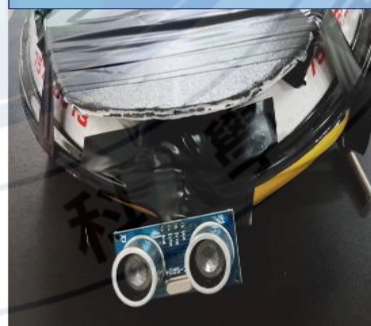
### 本船隻側視圖



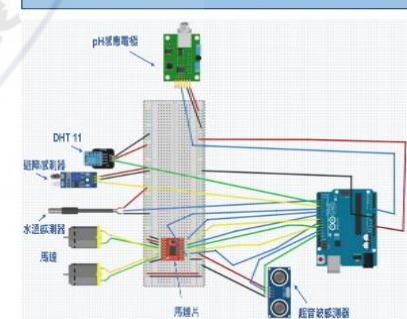
### 本船隻內部構造說明



### 本船隻前視圖



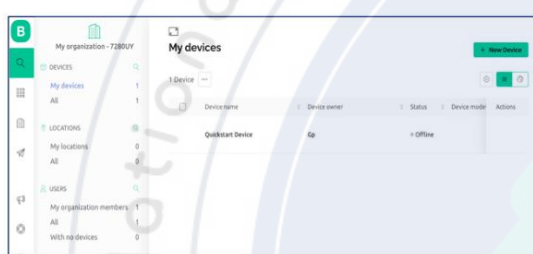
### 本船隻線路配置圖



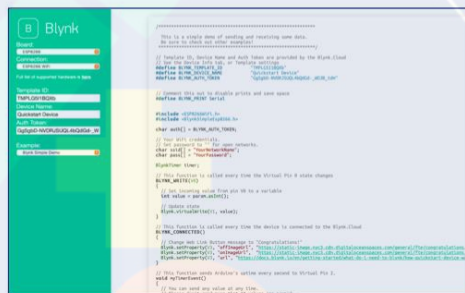


## 一、Blynk的部分

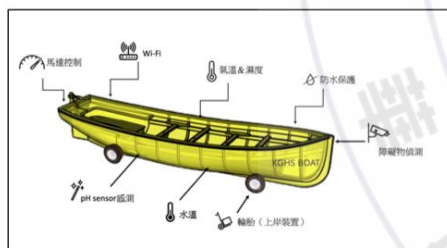
目的：連接新版Blynk並上傳結合Arduino程式，製作更簡易的使用者操作介面。  
藉由Blynk可立即透過監視窗獲取測量數值



步驟一：建立設備



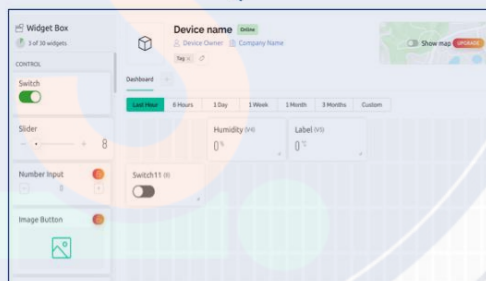
步驟二：使用新版程式碼



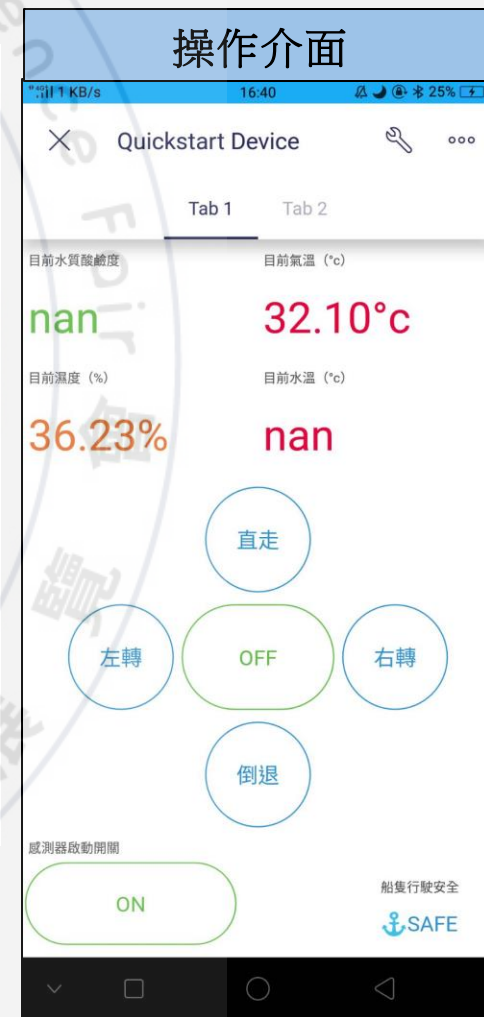
整體功能圖！



步驟四：完成！



步驟三：創建使用者介面



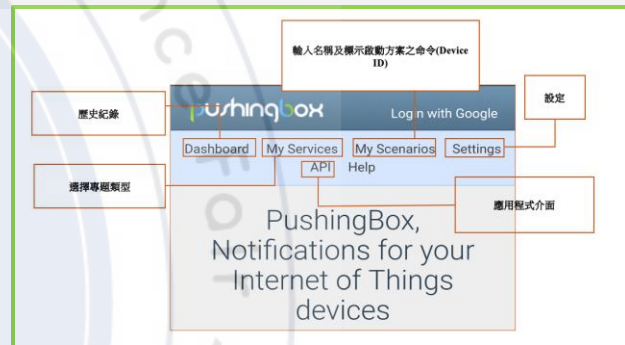
我們利用Blynk作為連接ESP8266與行動裝置的媒介，並解製作使用者操作介面用以控制馬達及感測器運作與偵測。

### 二、PushingBox的部分

目的：利用PushingBox將即時測得的水域環境數值轉傳至Google表單並回傳至excel進行大數據分析與歷史資料統整儲存



- 1. 取得google spreadsheet
- 2. 使用google app script
- 3. 開啟pushingbox
- 4. 部署應用程式



PushingBox操作介面



- 5. 建立上傳途徑
- 6. 嘗試上傳資料(範例)
- 7. 正確紀錄資料



8. 利用資料製作折線圖(濕度) 酸鹼值折線圖 水溫及氣溫折線圖

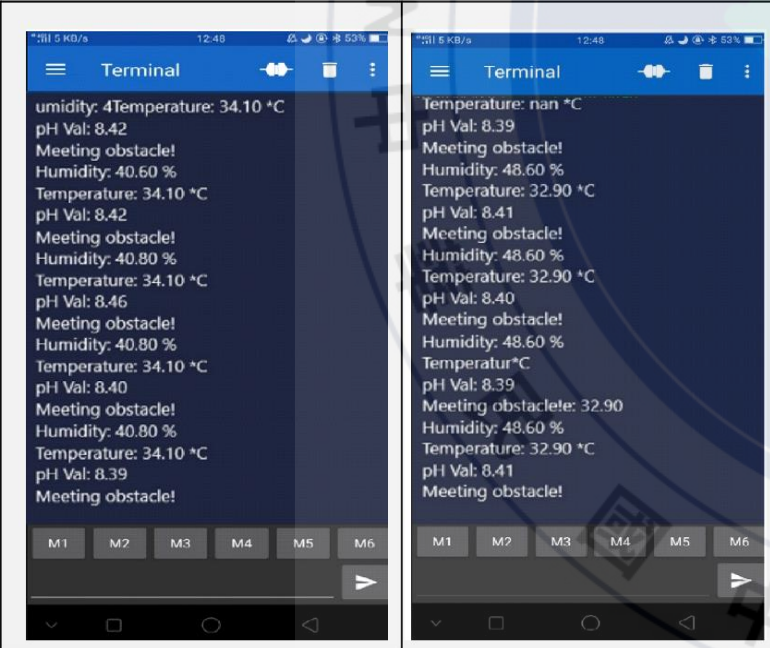
# 肆、研究結果：實際操作過程

## 自製遙控船運用於各靜止水域之任意點水質酸鹼值、水溫及濕度檢測

我們分別於3月和6月在高雄市蓮池潭與高雄某大學的生態池分別施放測量。

### 藍芽版監控畫面

前進大約8公尺(目測)後即迴轉回岸邊，來回5~6趟，每趟8~10分鐘，約1個小時左右完成測量。

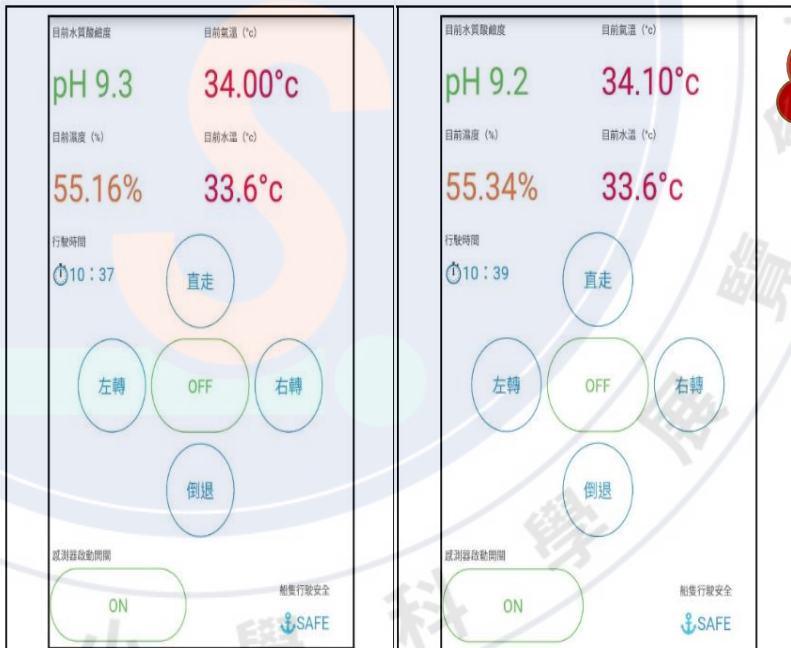


3/11(五)

3/12(六)

### Wi-Fi版監控畫面

前進大約70公尺(目測)後即迴轉回岸邊，來回4~5趟，每趟10~15分鐘，約1個小時左右完成測量。

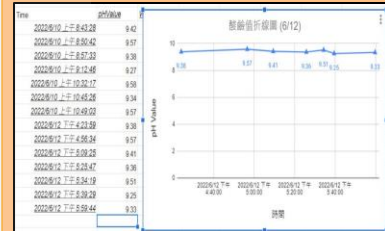


6/11(六)

6/12(日)



**Wi-Fi版船隻**  
可將監控得的數據上傳至PushingBox並回傳Excel，立即進行數據分析。





# 實際操作步驟介紹

Youtube

關鍵字收尋：

自組裝遠端操控智能船與無線傳輸技術監測環境水域



船隻運作以及實測方法



在眾多的測試中我們隨機取兩天的數據以作比較分析

PowerDirector



分別是6/11及6/12



打開程式 Blynk IoT

我們利用智慧型行動裝置傳送信號至船來操控船的行進

PowerDirector



選擇我們專為此船隻設計的專案

我們利用智慧型行動裝置傳送信號至船來操控船的行進

PowerDirector



以簡單易懂之指示操控船隻

以 WeMos D1 Wi-Fi 板作為訊號傳輸的工具

PowerDirector



以 WeMos D1 Wi-Fi 板作為訊號傳輸的工具

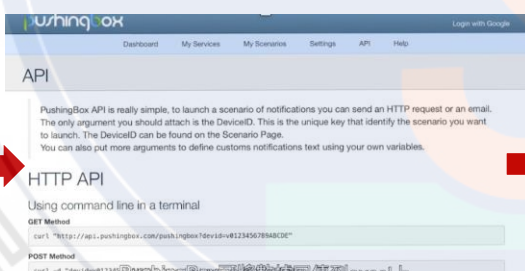
PowerDirector



使用 Blynk 及 PushingBox 程式去操控及蒐集船所測得之數據



使用 Blynk 及 PushingBox 程式去操控及蒐集船所測得之數據



## HTTP API

Using command line in a terminal

GET Method

```
curl -H "deviceid:1234567890" http://api.pushingbox.com/pushingbox?deviceid=1234567890
```

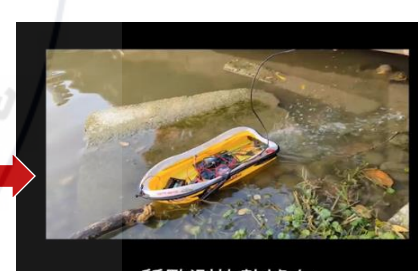
POST Method

```
curl -H "deviceid:1234567890" http://api.pushingbox.com/pushingbox?deviceid=1234567890
```

PushingBox 可將數據回傳到 excel 上



PushingBox 可將數據回傳到 excel 上



所監測的數據有



水溼酸鹼值、氣溫、水溫、濕度、與障礙物距離等



高雄大學生態池

PowerDirector



蓮池潭

PowerDirector



學校游泳池

PowerDirector



# 肆、研究結果：與政府單位網站數據比較

量測項目相互比對值

測量項目	6/11(六)	監測站資料	6/12(日)	監測站資料
水域酸鹼(pH值)	9.2~9.5	9.4	9.3~9.5	9.6
水溫(°C)	33.1°C	33.6°C	32.9°C	33.6°C
空氣濕度(%)	55.1~56.8%	52%	55.1~57.1%	50%

自製遙控船數據：  
 水域酸鹼值範圍:9.3~9.6  
 水域溫度範圍:33.1~34.1度  
 空氣濕度範圍:50.6~56.6%

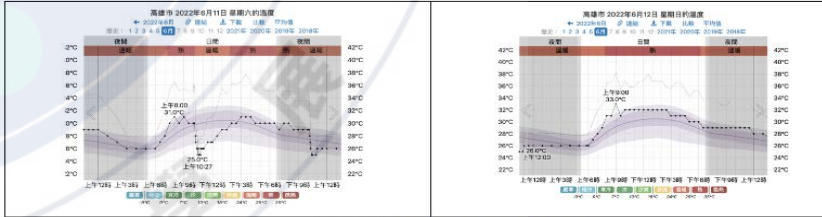
可歸納兩個測量方式(監測站和自製遙控船)差異不大  
 本研究自製船舶可應用於靜止或水流變化小水域之偵測

## 高雄市環保局水域酸鹼(pH值)

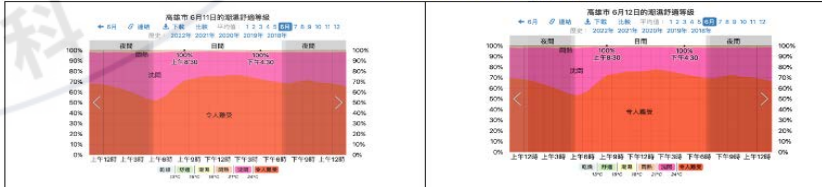
監測年月：中華民國 110 年 06 月，湖泊/河川別： 湖泊

測站名稱	採樣日期	採樣時間	河川污染指數 RPI	是否符合 戊類	水溫 °C	氫離子濃度指數 pH值	溶氧量 mg/L	生化需氧量 mg/L	化學需氧量 mg/L	懸浮固體 mg/L	氨氮 mg/L
人行拱橋	2021/06/03	11:34	--	否	33.6	9.4	7.6	9.4	34.2	27.2	0.13
龍船船塢	2021/06/03	11:24	--	否	33.4	9.7	7.8	8.4	32.2	21.8	0.20
遊艇碼頭	2021/06/03	11:15	--	否	33.6	9.6	8.0	8.2	29.4	18.4	0.09

## 中央氣象局空氣溫度與相對濕度



資料來源: <https://reurl.cc/Oiq8zq>





# 伍、討論

## 一、驅動能源

行動電源+鋰電池 (可充電耗能低)

## 二、程式碼的撰寫

藍芽模組 → Wi-Fi功能性 → 不同水域範圍

## 三、製作船隻的初衷

科普+自製運輸工具 → 減少檢測危險不便

## 四、水質檢測的地點

比對政府單位網站+水流平穩 → 準確度高

# 陸、結論

## 一、智慧型行動裝置遠端操控的藍芽操控船

第二版 Wi-Fi 船成品



## 二、實際應用



- 偵測項目
1. pH值
  2. 水溫
  3. 氣溫
  4. 濕度

可應用於生態池、灌溉埤塘、養殖魚塢 可建立大數據資料庫

## 三、比較Wi-Fi與藍芽效能差異

項目	藍芽	Wi-Fi
型號	HC-06	ESP 8266
傳輸距離	10 公尺內	60~100公尺
最高傳輸速度	24 Mbit/s	300 Mbit/s

## 四、研究系統特點

1. 傳輸訊號方面：速度最快、普及高的Wi-Fi
2. 船體設計方面：駕駛平穩、沉浮穩定方向。
4. 安全防護方面：(1)防水系統 (2)運行保護系統
5. 能源供給方面：二次能源與行動電源運行2hr
6. 控制方面：智慧型行動裝置藉由Wi-Fi遠端操控船體，同時數據回傳

## 五、成本表

第一版:藍芽船		
器材	數量	價格
Arduino nano	1個	200
pH值感應電極	1個	780
DHT11 溫溼度測量器	1個	40
DHT22 溫溼度測量器	1個	45
HC-06 藍芽	1個	100
TB6612FNG馬達片	1個	40
太陽能板	1個	400
TCRT5000紅外線感測器	1個	40
杜邦線	1包	20
船體	1組	500
共11項		2265

第二版:Wi-Fi船		
器材	數量	價格
WeMos D1	1個	126
pH值感應電極	1個	780
DHT11 溫溼度測量器	1個	40
HC-SR04超音波測距	1個	43
DS18B20 水溫感測模組	1個	231
TB6612FNG馬達片	1個	40
TCRT5000紅外線感測器	1個	40
杜邦線	1包	20
船體	1組	300
共11項		1620

## 柒、參考資料

- 1.趙英傑，超圖解 Arduino 互動設計入門(第二版)，旗標出版股份有限公司，2015。
- 2.葉難，Arduino 輕鬆入門：範例分析與實作設計，博碩文化出版，2014。
- 3.Mark Geddes，CAVEDU 教育團隊-曾吉弘譯，Arduino 自作專案大百科，基峰資訊，2015。
- 4.楊明豐，Arduino 最佳入門與應用：打造互動設計輕鬆學，基峰資訊，2015。
- 5.M. T. Lazarescu, "Design of a WSN platform for long-term environmental monitoring for IoT applications," IEEE Journal on emerging and selected topics in circuits and systems, vol. 3, pp. 45-54, 2013.
- 6.A. S. Ali, Z. Zanzinger, D. Debose, and B. Stephens, "Open Source Building Science Sensors (OSBSS): A low-cost Arduino-based platform for long-term indoor environmental data collection," Building and Environment, vol. 100, pp. 114-126, 2016