

# 中華民國第 56 屆中小學科學展覽會

## 作品說明書

---

國中組 生活與應用科學科

030803

能即時顯示用水量並具有物聯網功能的水流計  
量設備

學校名稱：臺北市立興雅國民中學

作者：  國二 賴亭瑋  國二 董廷倫  國二 李柏宸	指導老師：  劉奕漢  董增萊
---	-----------------------------

關鍵詞：節水、物聯網、水流計量設備

## 摘要

臺灣為世界排名第18位的缺水國家，面臨嚴重的缺水危機，有鑑於水資源的有限及寶貴，如何落實節約用水，是我們研究的初衷。我們利用Arduino為核心，配合水流計量器、OLED顯示幕及藍芽和WIFI模組，設計出能即時顯示用水量並具有物聯網功能的水流計量設備。每次用水時，都能讓使用者注意到用水狀況，且能記錄每次用水的時間及用量，並可透過藍芽或WIFI上傳至手機和雲端資料庫，具有簡易的物聯網功能；對於用水過多的行為，除螢幕反白及發聲警示外，更可控制電磁閥將水流截止，可避免忘記關水或是漏水的情形。此外累積一段時間後的雲端資料庫的大數據，可用來檢討用水習慣。此作品除了實用性，更有潛移默化的教育性，提醒改善不良用水習慣，從家庭落實節水教育。

## 壹、 研究動機

21世紀以來全球隨著氣候變遷、暖化效應，嚴重影響降雨量與分布；加上人口快速增長以及工業需求的急速擴張，使用的水量激增，世界各國均已面臨缺水的水資源危機。依據聯合國統計資料顯示，目前全球60億人口中，約有六分之一的民眾處於缺乏可用或衛生水資源的窘況。估計到2025年，全球將有30%的人口、50個國家面臨水資源匱乏的災難。人類過度開發危害到地球環境與氣候，水資源日漸枯竭的既成事實已成為全球人類必須共同面對的困境。

臺灣因夏季降雨豐沛，看似無缺水的危機，但卻是世界排名第18位的缺水國家。因為臺灣地形山陡流急，大部分的雨水都直接流入海洋；加上降雨過度集中、分布不均，臺灣每人每年所分配之降雨量僅達世界平均值七分之一。根據經濟部水利署統計，台灣每年浪費的水量約3,400億公升，相當1.5座石門水庫的蓄水量，但國人節約用水意識缺乏、用水習慣不良，加上水費便宜，造成人民缺乏危機意識，水資源持續浪費，所以應該更積極的節約用水。

水資源是有限及寶貴的，我們從自然與生活科技二下第7章課程學習到可以利用現代

科技建材來省水；也從自然與生活科技一下第8章的課程中，學到傳播科技已經對人類生活造成重大改變與影響，所以我們研究思考如何將兩者結合，設計具有物聯網功能的節水設備，來輔助落實省水。事實上，日常生活中的用水量不容易知道，所以很難控制用水，不容易培養節水意識。比如洗手用了多少水，洗澡用了多少水，都沒有量化的概念值。我們想要省水，就必須知道那些用水是被浪費掉的，就像想要省錢，第一件事就是記帳，才能知道哪些是不必要的開支。因此，我們討論後決定設計一種水流計量設備，讓使用者隨時打開水龍頭，就可以獲知每次的耗水量。就如同要省錢就要先記帳一樣，透過直觀顯示並紀錄統計用水量，針對每人用水習慣對症下藥來省水，才能當用則用、當省則省。



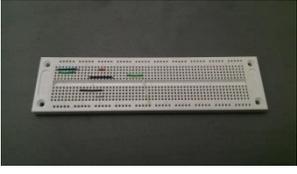
圖 1、世界各國降雨量分佈圖(取自經濟部水利署)

## 貳、 研究目的

本研究的目的是在於因應日漸缺乏的水資源，希望藉由能即時顯示用水量並具有物聯網功能的水流計量設備，提供分析了解個人和家戶用水習慣，進而檢討用水合理性，達成制約個人用水量，加強節水意識，落實節約用水與合理運用水資源的永續理念。

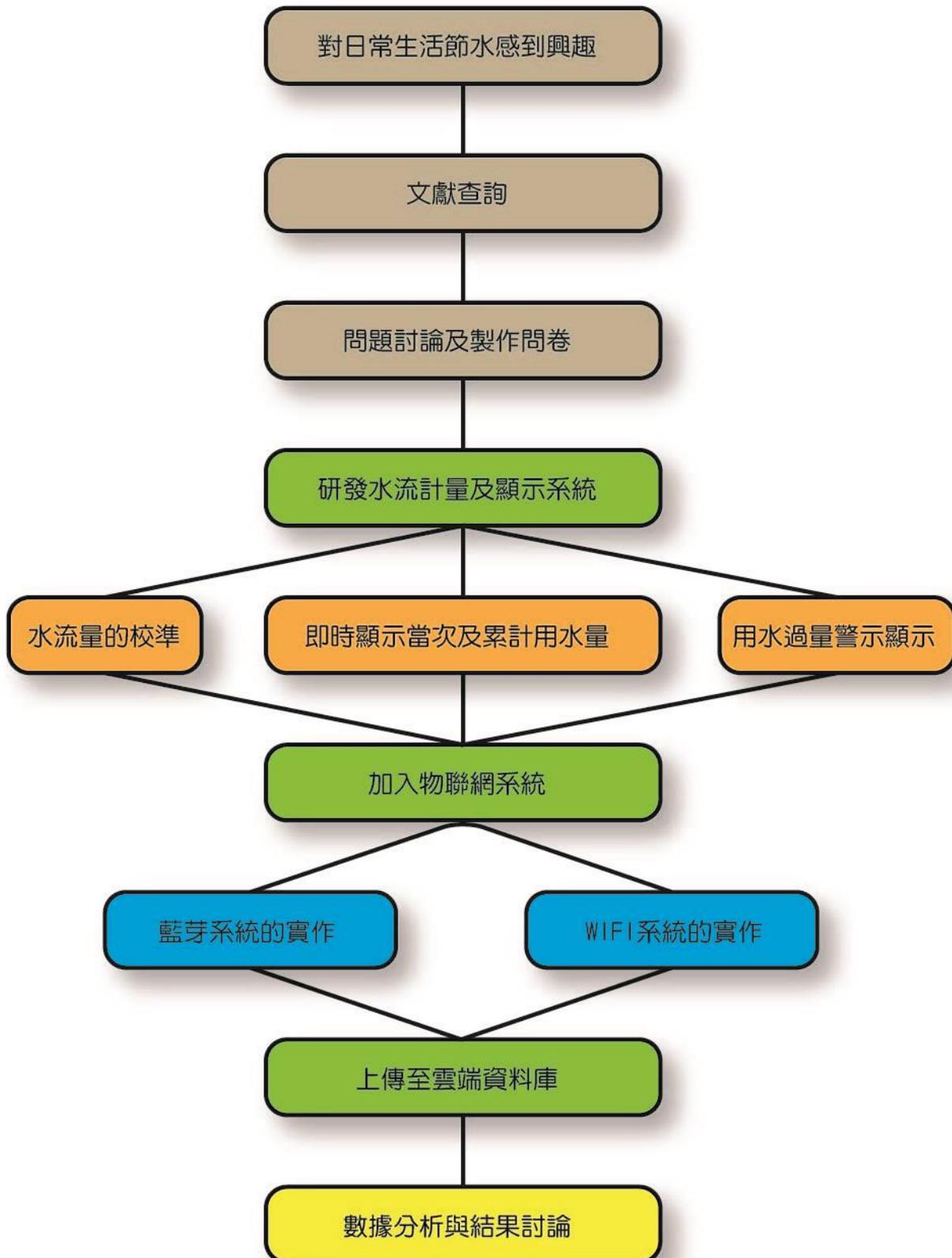
- 一、開發能即時顯示本次用水和累計用水量及用水過量警示的計量設備。
- 二、開發能透過藍芽或 WIFI 即時上傳雲端資料之 Arduino 軟體程式。
- 三、分析雲端資料，檢討用水習慣，喚起民眾節水意識，落實節水教育。
- 四、與現有類似的水流計量產品的比較。

## 參、 研究設備及器材

			
Arduino UNO 開發板	水力發電模組	可充電鋰電池	DC 升壓模組
			
具有霍爾元件的水流計量器	Arduino OLED 顯示幕	Arduino 藍芽傳送模組	Arduino WIFI 傳送模組
			
麵包板	手機或平板電腦	筆記型電腦	USB 轉 TTL 序列板
			
塑膠管材	金屬束帶	轉接頭	強力膠帶
			
蓮蓬頭管	高壓接管		

## 肆、 研究過程及方法

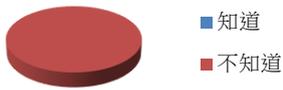
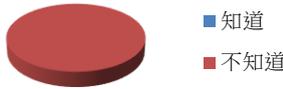
### 一、 思考與研究流程圖：

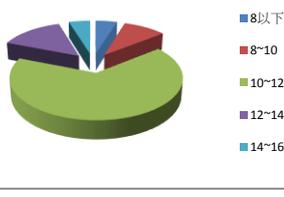


## 二、使用節水器材的現況調查與分析

我們進行初步的討論，發現我們其實對於自己家中用水情形幾乎不瞭解，要問過家長甚或查水費單據才知道家中用水度數；對於家中每個水龍頭和耗水設備每月的用水量更是完全沒概念，甚至有人表示 1 度水相當於 1000 公升也是第一次聽到。所以，我們初步構想是覺得要能節約用水，就應該有一個更直觀的計量設備，就像手錶隨時可看時間，提醒我們珍惜時光一樣；此計量設備能讓使用者注意到每次用水狀況，隨時提醒要珍惜水資源。

我們製作了 1 份簡單的問卷進行訪查(如附件 1)，詢問周遭同學親友是否瞭解家中每月家中平均用水量；各個水龍頭或用水設備的耗水量；如果有了計量設備，是否覺得對提醒節水有更直觀的幫助等。經過 1 周的調查，共發出 25 份問卷，回收有效問卷 21 份，我們發現原本不知道家中用水度數者有 17 人，佔了 81%；而民眾也都有省水意識，多少都有實際做過省水措施。以下為問卷的回收結果統計：

1	是否知道家中最近這期(二個月)用水量總共幾度？ <input type="checkbox"/> 知道。(4 人)(19%) <input type="checkbox"/> 不知道，查詢後才知道。(17 人)(81%)	 <p>知道 19% 不知道 81%</p>
2	家中哪個水龍頭或用水設備耗水量最多？ <input type="checkbox"/> 知道。(0 人)(0%) <input type="checkbox"/> 不知道。(21 人)(100%)	 <p>知道 不知道</p>
3	猜測哪個水龍頭或用水設備耗水量最多？ <input type="checkbox"/> 洗衣機。(5 人)(23.8%) <input type="checkbox"/> 廚房。(11 人)(52.4%) <input type="checkbox"/> 臥室蓮蓬頭。(3 人)(14.3%) <input type="checkbox"/> 馬桶。(2 人)(9.5%)	 <p>洗衣機 廚房 臥室蓮蓬頭 馬桶</p>
4	是否知道家中耗水量最大之水龍頭或用水設備，每月或每日耗水量？ <input type="checkbox"/> 知道。(0 人)(0%) <input type="checkbox"/> 不知道。(21 人)(100%)	 <p>知道 不知道</p>

5	<p>你的家庭有"實際做過"的省水方式(可複選)</p> <p><input type="checkbox"/>水龍頭裝節水墊片。(19 人)(90.5%)</p> <p><input type="checkbox"/>換裝節水標誌的設備或省水馬桶。(17 人)(80.9%)</p> <p><input type="checkbox"/>回收水再利用。(5 人)(23.8%)</p> <p><input type="checkbox"/>會影響原本用水習慣，裝了又拆。(15 人)(71.4%)</p> <p><input type="checkbox"/>太麻煩了，不想裝。(2 人)(9.5%)</p> <p><input type="checkbox"/>其他。(0 人)(0%)</p>	
6	<p>你們家庭有進行各項之節約措施嗎？</p> <p><input type="checkbox"/>有，加裝各項節水設施(包括馬桶廂內放置寶特瓶)。(19 人)(90.5%)</p> <p><input type="checkbox"/>有，父母(或學校、廣告等)有看到節水宣導，我有配合導正用水習慣。(2 人)(9.5%)</p> <p><input type="checkbox"/>沒有，因為不知道如何節水。(0 人)(0%)</p> <p><input type="checkbox"/>沒有，因為我家用水不多，不用省水。(0 人)(0%)</p>	
7	<p>如果免費安裝可以簡單測量家庭用水的設備，提醒每日用水量或水費，你覺得會讓你因此而更願意節約用水嗎？</p> <p><input type="checkbox"/>會。(19 人)(90.5%)</p> <p><input type="checkbox"/>不會。(2 人)(9.5%)</p>	
8	<p>平均每人兩個月間用水度數(水費單上用水度數/家中人數)</p> <p><input type="checkbox"/>8 度以下。(1 人)(4.7%)</p> <p><input type="checkbox"/>8~10 度。(2 人)(9.5%)</p> <p><input type="checkbox"/>10~12 度。(14 人)(66.7%)</p> <p><input type="checkbox"/>12~14 度。(3 人)(14.3%)</p> <p><input type="checkbox"/>14~16 度。(1 人)(4.7%)</p>	

就問卷結果分析，我們初步構想的直觀用水計量設備，是獲得大家肯定的，佔了 90.5%。如果有了此類設備，可以隨時提醒用水者注意到每次用水狀況，除了實用性，更有潛移默化的教育性，可以喚起民眾節水意識，讓用水者注意到自己耗水的不良用水習慣，從家庭用水漸漸改變，落實節水教育。因此我們就決定以此為目標來進行我們的科展實作。

### 三、Arduino 電路配置與 App Inventor 2 軟體應用

#### (一) Arduino 與周邊元件的電路配置

##### 1、Arduino UNO 結構圖：

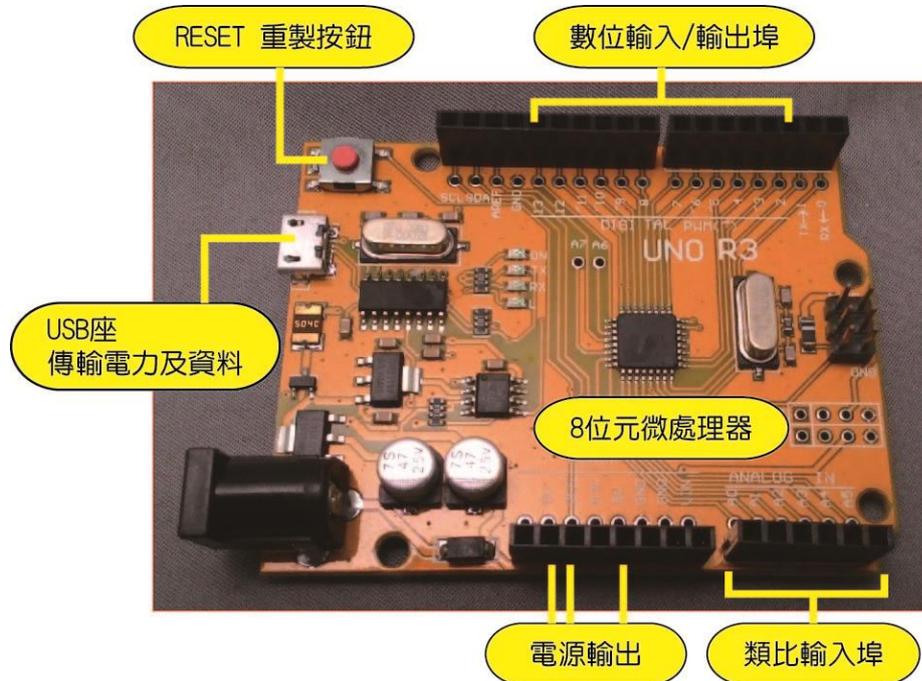
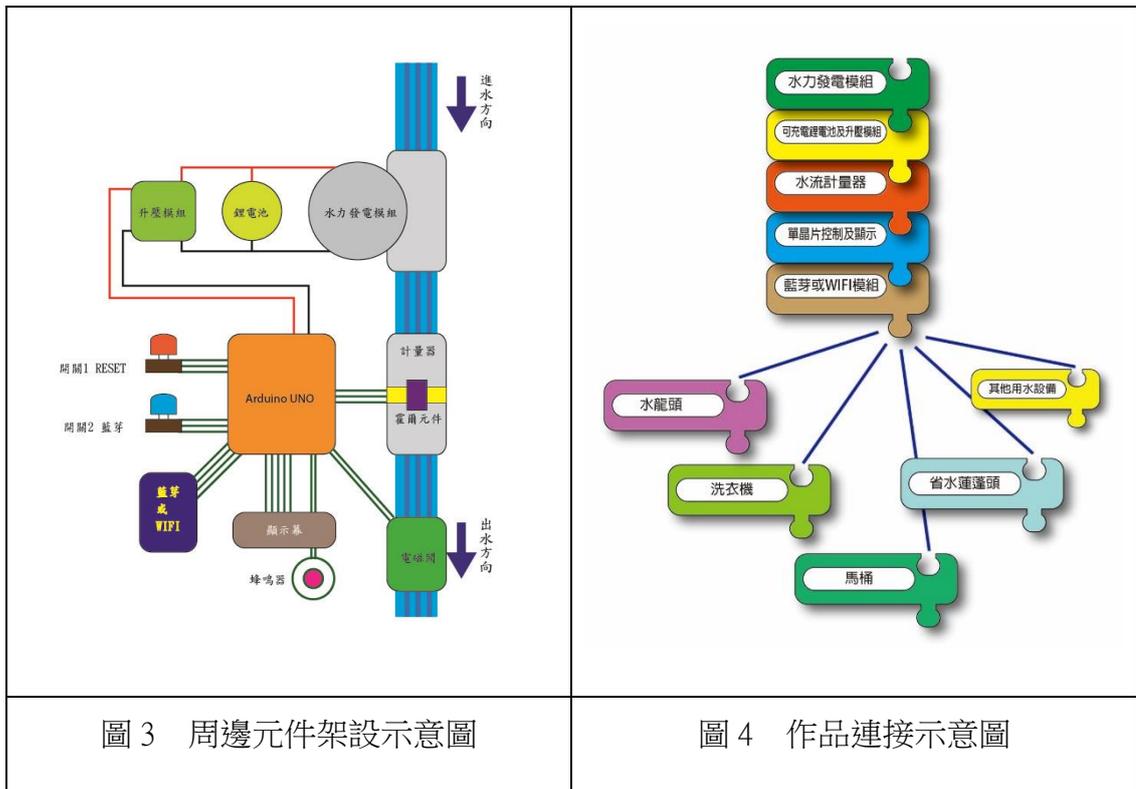


圖 2 Arduino UNO 結構圖

2、 周邊元件架設：電源部分由水力發電模組、DC 3.7V 可充電鋰電池及 DC 3.7V 升壓至 DC 5V 的升壓模組構成；將具有霍爾元件的水流計量器連接 Arduino 數位輸入埠，Arduino OLED 顯示幕連接數位輸出埠。水流通路則由水管連接水力發電模組、水流計量器及電磁閥組成。無線資料傳輸至雲端資料庫，則由藍芽或 WIFI 套件來完成。

(二) Arduino 控制指令撰寫：根據實驗設計需求，自行完成各元件電路的基本架設，參考 Arduino 書本及網路上範例，及老師和家長的協助，完成 Arduino 控制程式(詳如附件 2~4)。

(三) App Inventor 2 控制指令撰寫：主要參考 AppInventor 中文學習網內範例，完成 Android 系統的 APP 控制程式 (詳如附件 5、6)。



#### 四、水流計量器的校準量測

##### (一) 實驗目的：

確定水流計量器獲得的霍爾元件回傳數值與實際用水量間的比例。

##### (二) 實驗方法與步驟：

- 1、 撰寫 Arduino 程式，以第 2 腳為接收水流計量器霍爾元件的回傳值，當水流經過計量器中的渦輪時，每轉一圈霍爾元件就會回傳數值，我們可用程式來累加圈數；再來用固定的水量來實測累計圈數，依兩者間的比例關係，就可作為我們將來計量的基礎。
- 2、 以量筒測量出 600cc 的水量，先倒入大燒杯中，再將燒杯中的水倒入管路中流經水流計量器，並利用 Arduino 程式來累計圈數。
- 3、 以量筒測量出 1800cc 的水量，先倒入大寶特瓶中，再將大寶特瓶中的水倒入管路中流經水流計量器，並利用 Arduino 程式來累計圈數。
- 4、 接上自來水龍頭，在有水壓狀況下，相同時間(1 分鐘)不同流量，是否能準確測出用水量。
- 5、 將此比例作為我們後續量測水量的基準。

## 五、開發能即時顯示本次用水和累計用水量及用水過量警示的計量設備

### (一) 實驗目的：

開發能即時顯示本次用水和累計用水量及用水過量警示的計量設備。

### (二) 實驗方法與步驟

- 1、 撰寫 Arduino 程式，以水流計量器霍爾元件的回傳值換算出用水公升數。
- 2、 將現況用水數值即時在 OLED 顯示幕上顯示，下方並同時顯示累計公升數。
- 3、 設定警示值，當用水超過此值時，將用水數值反白，蜂鳴器並發出警告聲音，作為警示。
- 4、 設定強迫截止值，當用水超過此值時，啟動電磁閥，將水源關閉，可避免忘記關水或是漏水的情形。若是必要之大量用水，使用者須按鈕確認，電磁閥才會打開繼續供水。

## 六、以藍芽(Bluetooth)傳輸至手機後上傳雲端資料庫

我們最初選取藍芽做為資料無線傳輸時，是希望能直接傳送到雲端資料庫，後來發現我們可購買到的 Arduino 藍芽套件，只支援藍芽 2.1，不能直接傳到網路上，要最新的藍芽 4.2 技術，才有「IP 聯網」功能，可惜我們在市面上並沒有找到販售支援 Arduino 的藍芽 4.2 套件。

經過討論我們想到可以用手機或平板電腦來中繼，當藍芽連接 Arduino 和手機或平板傳送資料時，利用手機或平板聯網的功能，同步將資料傳到網路上。至於手機和平板端的 APP 程式，我們用麻省理工推廣的 MIT App Inventor 2 來製作，此種程式可以製作在 Android 系統上執行的 APP。

### (一) 實驗目的：

設計程式能將用水資料透過藍芽(Bluetooth)傳輸至 Android 系統手機或平板，再由手機或平板同步上傳至雲端資料庫。

### (二) 實驗方法與步驟：

- 1、 撰寫 Arduino 程式，連結藍芽(Bluetooth)套件 HC-05，並設定鮑率(Baud Rate)為 9600

bps。

- 2、將時間資料及用水數值，轉成 byte(0~255)值，每次用水資料分成 6 小筆資料傳送。例如 12/09 21:59 用水 12.34 公升，將資料分成 12, 9, 21, 59, 12, 34 (月/日/分/秒/小數前/小數後)6 小筆。
- 3、設定簡單傳送協定，避免傳送資料錯誤。傳送時手機先送出特定值”49”，當 Arduino 接收到，開始傳送第一筆特定資料”a”作為驗證用，手機確認第一筆資料正確後，繼續往後接收資料。
- 4、以麻省理工推廣的 MIT App Inventor 2 來製作 Android 系統手機或平板的接收端，並同時將資料上傳到 google 所提供的免費資料庫 TinyWebDB。

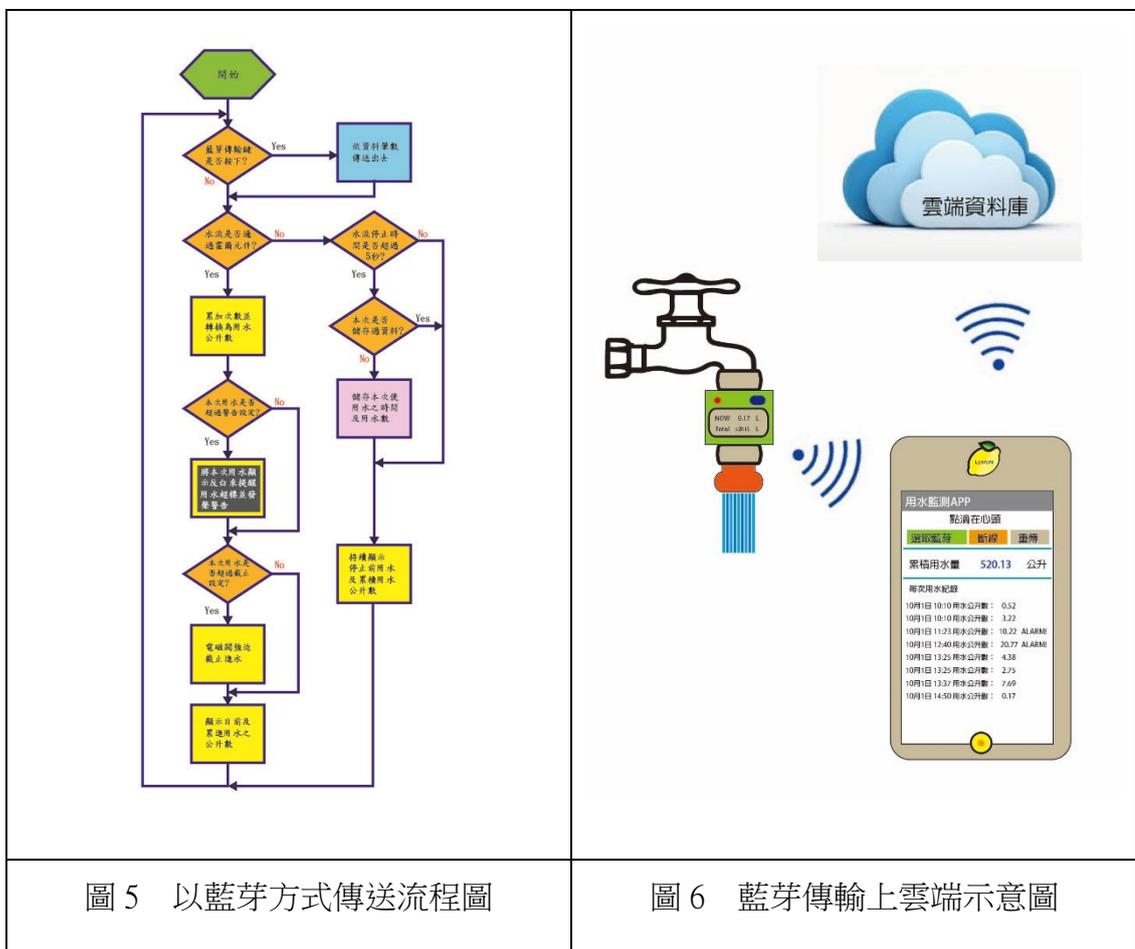


圖 6 藍芽傳輸上雲端示意圖

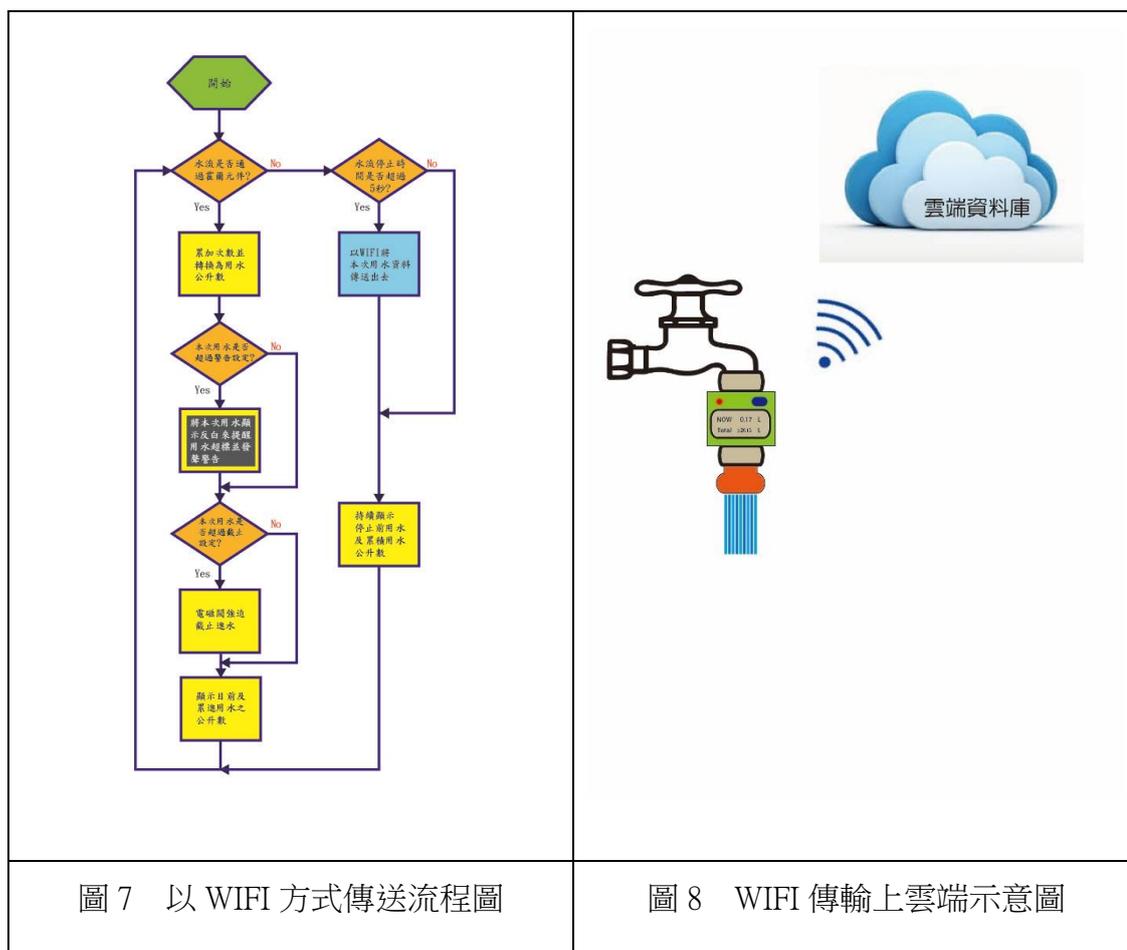
## 七、以 WIFI 模組直接上傳雲端資料庫

### (一) 實驗目的：

設計程式能將用水資料透過 WIFI 模組直接上傳至雲端資料庫。

(二) 實驗方法與步驟：

- 1、以 USB 轉 TTL 序列板連接 WIFI 晶片 ESP8266，設定鮑率(Baud Rate)為 9600 bps 以符合 Arduino Uno 傳送速率。
- 2、設定無線網路基地台 SSID (服務設定識別碼)及密碼，使可連結上無線網路。
- 3、撰寫 Arduino 程式，連接 WIFI 晶片 ESP8266。
- 4、將資料上傳到專門為物聯網設備提供的免費雲端資料庫 ThingSpeak: Internet Of Things ( <https://thingspeak.com/> )。



八、連接水龍頭及蓮蓬頭實測用水情形

(一) 實驗目的：

- 1、實測用水顯示及上傳至雲端資料庫情形。
- 2、觀察使用者在此設備下之用水量變化。
- 3、分析雲端資料，檢討用水不良習慣，喚起節水意識，落實節水教育。

(二) 實驗方法與步驟：

- 1、 將設備以防水塑膠袋裝好，連接至水龍頭下方及蓮蓬頭，並將 OLED 顯示幕配線加長，放置於明顯可見處，同樣以防水塑膠袋裝好。
- 2、 每次測試周期為五天，分別測試特定人一般日常行為的累計用水量，並限定使用次數為洗澡一次、刷牙二次及洗手五次。第一天為不顯示用水量，第二至五天為即時顯示用水量。
- 3、 將五日之用水狀況，利用雲端資料進行比較。



圖 9 連接於水龍頭下方



圖 10 改裝蓮蓬頭用水管



圖 11 將 OLED 顯示幕配線加長，以防水塑膠袋裝好

## 伍、 研究結果

### 一、水流計量器的校準量測

以下為重複測量 600cc 及 1800cc 水量，Arduino 所獲得的數值：

次數	第一次	第二次	第三次	第四次	第五次	第六次
600cc	201	198	203	200	201	200
1800cc	602	601	598	602	600	600

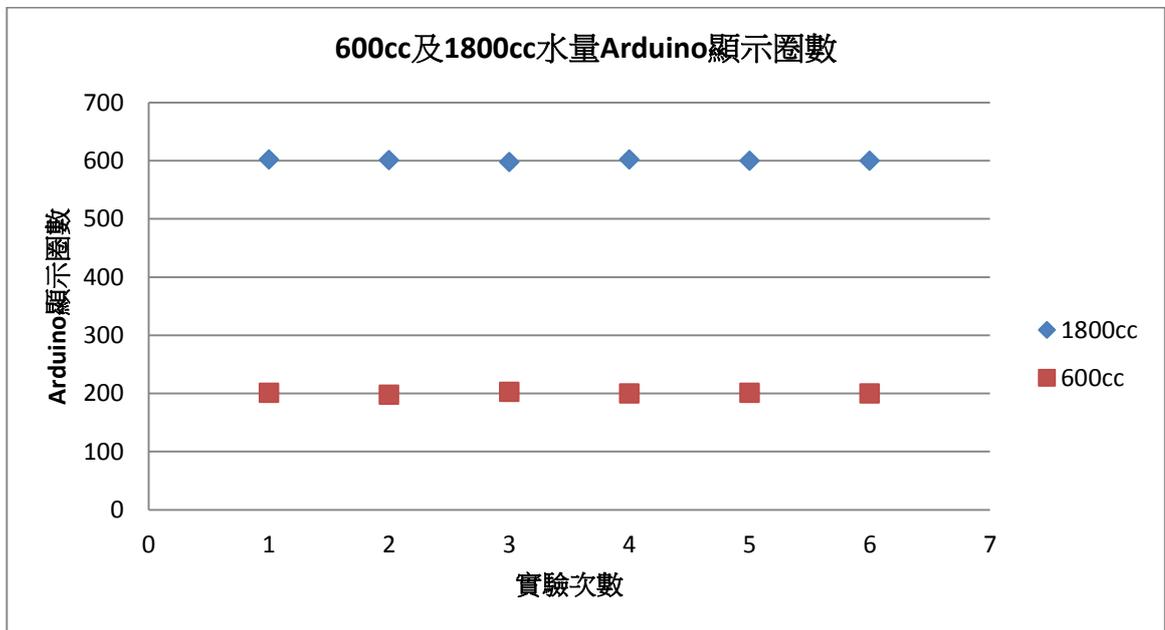


圖 12 600cc 及 1800cc 水量與 Arduino 顯示圈數之關係圖

以下為在有水壓狀況下，相同時間(1 分鐘)不同流量，Arduino 所獲得的數值：

實驗狀況	實際出水量	Arduino 顯示圈數
水龍頭"微開"1 分鐘	1350cc	452
水龍頭"半開"1 分鐘	3820cc	1275
水龍頭"全開"1 分鐘	9450cc	3160

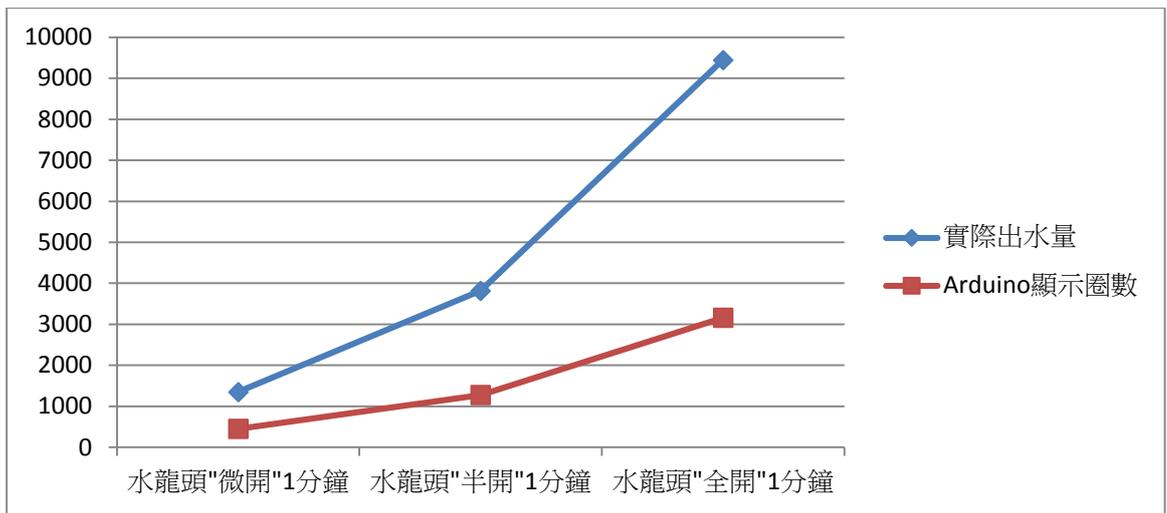


圖 13 不同水量出水量和流速與 Arduino 顯示圈數之關係圖

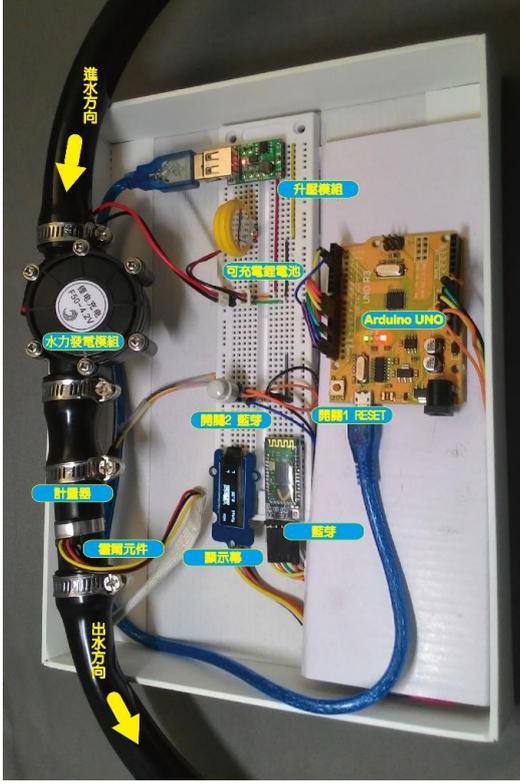
經由上述測量，我們發現用燒杯或大寶特瓶灌水時，每 600cc 約等於 200 轉。另外在有水壓狀態下，不同的出水量和流速，所量測出的數值，也約等於此比例，可知水流速及流量的變化，並不影響水流計量器的精準度，因此我們就以  $600\text{cc} = 200$  轉，作為我們量測水量的基準。



圖 14 水流計量器的圈數實測

## 二、開發能即時顯示本次用水和累計用水量及用水過量警示的計量設備

以前述  $600\text{cc} = 200$  轉的比例為基準，我們重新修正了程式，除了可即時顯示用水量，也可同時顯示到目前為止的累計用水量；我們還內設了 5 公升為用水過量警示，當次用水超過時，顯示數值就會反白，蜂鳴器也會發出聲音，提醒使用者注意用水。若是單次用水超過 100 公升，則會啟動電磁閥，將水源截止關閉，可避免忘記關水或是漏水的情形。若是必要之大量用水，使用者須按紐確認，電磁閥才會打開繼續供水。

	
<p>圖 15 作品組裝圖</p>	<p>圖 16 安裝於水龍頭出口</p>
<p>圖 17 單次用水超過 5 公升反白提醒</p>	

### 三、以藍芽(Bluetooth)傳輸至手機後上傳雲端資料庫

(一) 第一代作品: Arduino 和手機同時按鍵啟動傳送資料(第一代作品之研究成果曾於 2015 年全國能源科技創意實作競賽中發布，為敘明研究脈絡，在此簡單描述)

當需要資料傳送時，執行手機或平板 APP，選取藍芽裝置，選完後按下 Arduino 上資料傳送按鍵，雙方就會執行資料傳送。而手機或平板上的 APP，在接收資料的同時，也會將資料傳送到雲端資料庫紀錄。而且每次用水的時間、用水量和超過用水過量警示的資料等，也會在手機或平板以及雲端資料庫顯示。



圖 18 按鍵啟動傳送資料之 APP



圖 19 按鍵啟動傳送資料之 Arduino 組裝圖

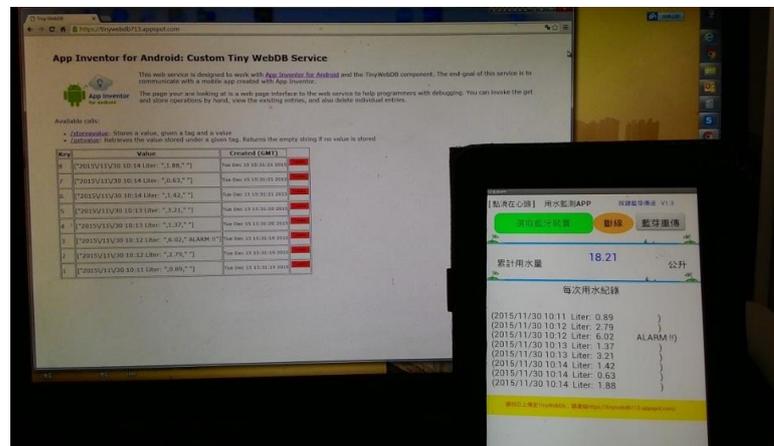


圖 20 將資料經藍芽傳送到手機，手機接收同時傳送到雲端資料庫

## (二) 第二代作品：手機單方面就能遙控接收資料

第一代藍芽在傳輸前，為了確保傳送資料正確，在傳送前要同時按鍵啟動 Arduino 和手機，我們覺得有些不便，希望能只用手機單方面就能遙控接收資料，因此我們改善了手機端的 MIT App Inventor 2 程式，在選取藍芽連接後，自動間隔時間送出“49”，其中任何一次 Arduino 接收到並傳送第一筆資料”a”，手機接收驗證正確後，雙方就繼續往後傳送接收資料，如此就不需要在 Arduino 端按鍵啟動。



圖 21 手機單方面就能遙控接收資料之 APP

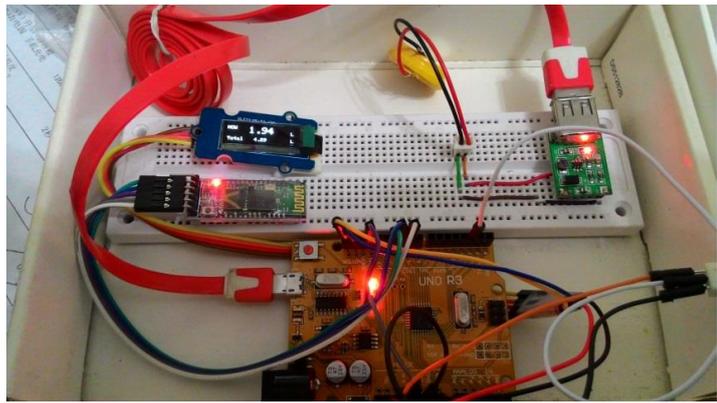


圖 22 手機單方面就能遙控接收資料之 Arduino 組裝圖 (Arduino 端不需要按鍵配合)

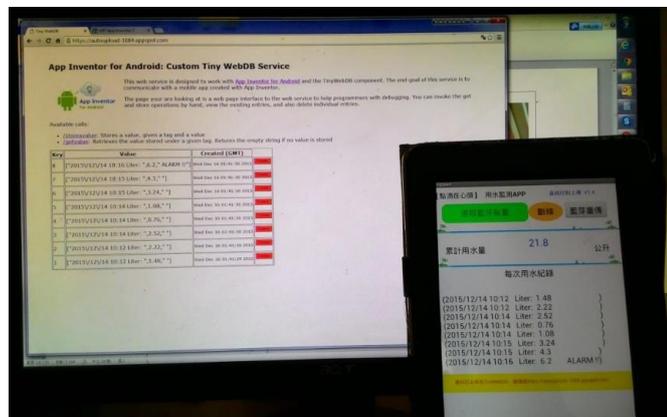


圖 23 手機能遙控將資料經藍芽傳送，接收同時傳送到雲端資料庫

#### 四、以 WIFI 模組直接上傳雲端資料庫

同前述實驗步驟獲得用水資料，當每次用水後 5 秒，Arduino 偵測到沒有繼續用水時，就會利用 WIFI 模組將本次用水資料上傳至免費雲端資料庫：ThingSpeak: Internet Of Things (<https://thingspeak.com/>)。與藍芽傳送資料的差別是，因為是單筆即時傳送，不是累積儲存多筆資料後再傳，因此可利用此雲端資料庫之寫入時間當作用水時間，傳送時就不用將時間資料加入。此外，因為是透過標準 Http 協定傳送，程式中我們也可省掉將資料拆成 byte 來傳送及編寫簡單傳送協定的部分。

另外此免費雲端資料庫：ThingSpeak，還可將資料傳出成 EXCEL 可讀的 csv 格式，方便我們整理及進一步分析資料。

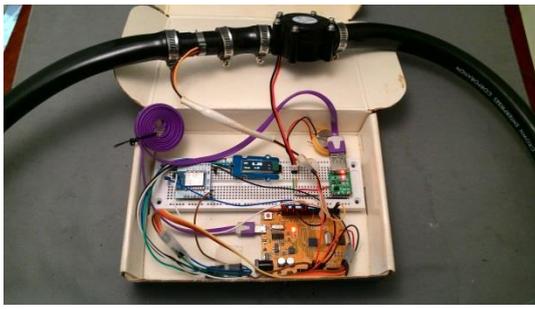


圖 24 WIFI 直接上傳雲端資料庫之 Arduino 組裝圖

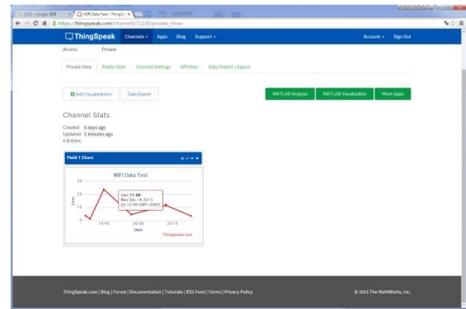


圖 25 上傳至 ThingSpeak 雲端資料庫

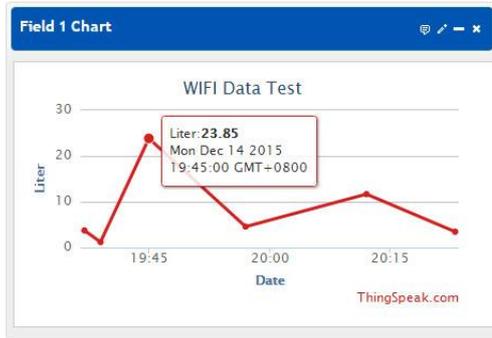


圖 26 局部放大圖(滑鼠點有詳細資料)

created_at	entry_id	field1
2015-12-14 11:57:00 UTC	1	3.75
2015-12-14 11:59:00 UTC	2	1.22
2015-12-14 11:55:00 UTC	3	23.85
2015-12-14 11:57:00 UTC	4	4.58
2015-12-14 12:12:00 UTC	5	11.68
2015-12-14 12:23:00 UTC	6	3.47

圖 27 雲端資料下載後由 EXCEL 讀取

## 五、連接水龍頭及蓮蓬頭實測用水情形

在已知安裝計量設備情形下，分別測試特定人洗澡(每日一次)、刷牙(每日二次)和洗手(每日五次)三種行為的累計用水量。第一天為不顯示用水量，第二至五天為即時顯示用水量，五天資料均透過 WIFI 自動上傳至雲端資料庫(ThingSpeak)，每次用水都可知道用水量及上傳時間，我們由雲端資料庫下載資料後，整理出總用水量及累計開關水龍頭的次數。

(一) 每日累計用水量(單位：公升)：

	第一天	第二天	第三天	第四天	第五天
測試人員 A	175.44	146.95	129.06	107.65	100.25
測試人員 B	213.25	165.09	141.75	128.01	119.38
測試人員 C	149.76	132.72	109.42	86.42	82.04

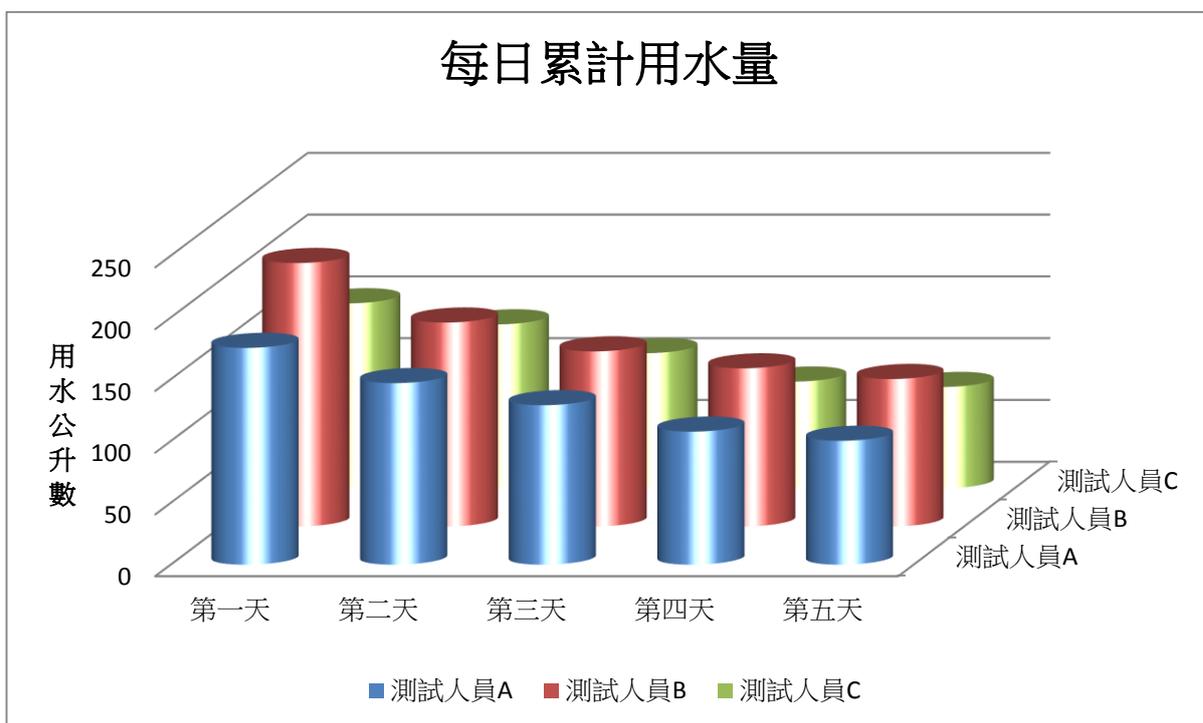


圖 28 測試人員每日累計用水量比較圖

(二) 每日累計開關水龍頭次數：

	第一天	第二天	第三天	第四天	第五天
測試人員 A	17	18	21	21	21
測試人員 B	17	20	21	21	21
測試人員 C	17	17	20	21	21

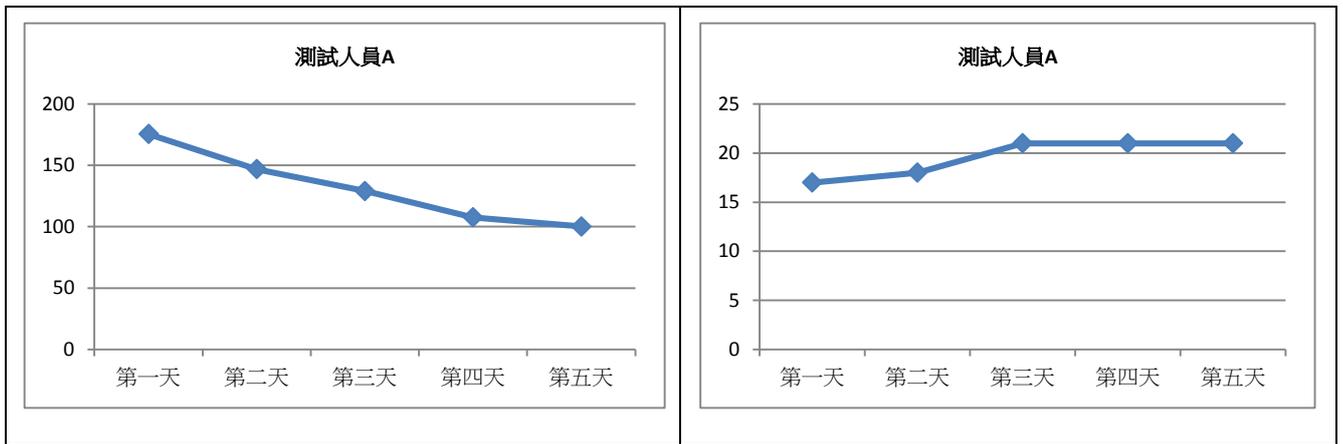


圖 29 測試人員 A 每日用水量與開關水龍頭次數比較圖

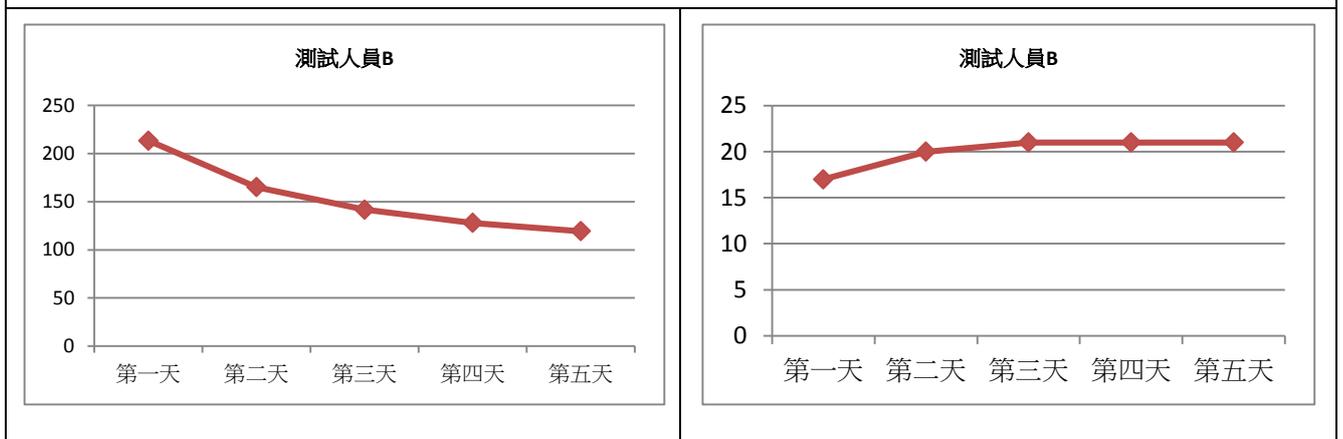


圖 30 測試人員 B 每日用水量與開關水龍頭次數比較圖

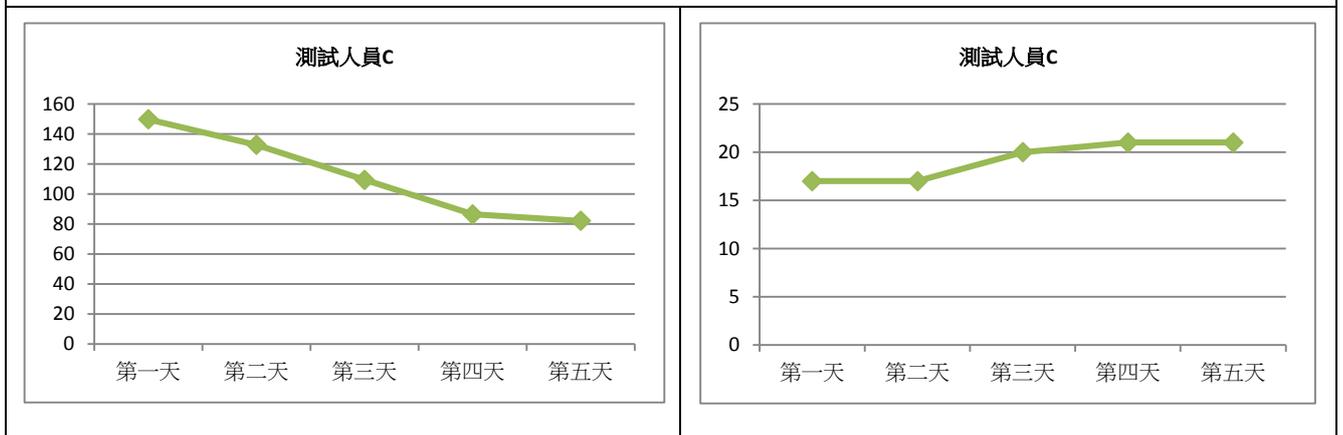


圖 31 測試人員 C 每日用水量與開關水龍頭次數比較圖

我們發現，即使在已知裝設計量器材的狀況下，有顯示用水量仍可對測試者心理產生壓力，達到節水的目的。受測者表示在用水時，隨著用水數字不斷跳動提醒外，因為知道用水情形是在被記錄中，因此會主動地開始省水。最明顯的就是增加開關水龍頭次數，以往沒有數值顯示及警示，往往就讓水龍頭保持開啟狀態，現在看到數值不斷增加，

就會主動關閉水龍頭。此外，在受測期間知道自己每次行為的用水公升數，**心理上也會希望自己下次相同行為時能用更少的水**。就實測的五天中，開關水龍頭的次數都增加了三次，而與第一日相較，到第五日時，測試人員分別省水了 75.19 公升、93.87 公升及 67.72 公升，省水比例分別達到 42.86%、44.02%及 45.22%。

若以上述數據為參考，每人每日保守估計約可節省約 70 公升用水，以每戶四人，每月 30 天估算，每月可節省  $70*4*30=8400$  公升，等於 8.4 度。水費計算方式若以台北市為參考，以一般水費 21-60 度單價 6.7 元計算，用水費可節省  $8.4*6.7=56.28$  元。水源保育與回饋費每度 0.5 元，可節省  $8.4*0.5=4.2$  元。代徵下水道費每度 5 元，可節省  $8.4*5=42$  元。**每月水費應繳總金額共可節省  $56.28+4.2+42=102.48$  元。**

## 六、與現有類似的水流計量產品的比較

本作品是我們努力研究並逐步改善的結果，以關鍵字搜尋並沒有發現相同的作品，唯一類似的產品是智慧水表(又稱電子水表)，由網路搜尋得知目前台北市政府選擇興隆公宅 272 戶示範，每戶補助裝設智慧水表約 3500 元，用戶可透過專屬帳號、密碼登入 App，了解每 10 分鐘用水量。

而我們的作品是用 Arduino 實驗版加上周邊模組，在尚未整合商品化前，硬體成本約 600 元。若以每一家庭八個用水設備估算(廚房、洗衣機、洗臉台\*2、蓮蓬頭\*2 及馬桶\*2)，即使全部安裝總費用約 4800 元。

本作品與智慧水表兩者比較如下：

- (一) 本作品可知道每個水龍頭或用水設備，每次用水的情形，而智慧水表只能顯示每 10 分鐘的家戶總用水數據。
- (二) 本作品可即時顯示用水量，可由「心理上提醒」使用人主動節水及檢討自己的用水行為，智慧水表並無此功能。
- (三) 本作品當單次用水超過設定警示值，顯示幕將數值反白，蜂鳴器並發出警告聲音，作為警示。若再超過強迫截止值，則會自動啟動電磁閥，將水源關閉，可避免忘記關水或是漏水的情形。智慧水表只能事後依用水量判斷是否曾發生忘記關水或漏水，但珍貴水資源早已經浪費。
- (四) 本作品可透過加總所有用水設備之用水量，和水費單上總用水量做比較，若有較大

差額，可推估在「水表後至水龍頭間」之管路有漏水情形，及早發現維修。而智慧水表對此漏水情形無法察覺顯示。

## 陸、 討論

- 一、依維基百科上的定義，「物聯網（Internet of Things，縮寫 IoT）是網際網路、傳統電信網等資訊承載體，讓所有能行使獨立功能的普通物體實作互聯互通的網路。」在我們此次具有物聯網功能的水流計量設備的實驗中，依現在可容易購買的套件來比較，藍芽模組和 WIFI 模組的價位差不多，均為 100 元上下，但 WIFI 模組可直接聯網，使用上較為簡易。
- 二、目前的實驗研究僅做到將用水資料傳遞上雲端資料庫，再來事後分析檢討。有了資料應該能做更多的事，將來也許能在資料庫端設計程式，使其能夠有回饋機制，例如每日累計用水量超過平均值，可主動發 EMAIL 或傳訊通知、比較資料後發現可能有管路漏水情形主動警示等。若蒐集的資料夠龐大，也能比較出家戶用水的用量和習慣，對於用量大的家庭建議換裝節水設備或調整用水習慣。
- 三、周遭每個人都有遇到或聽過馬桶故障卡住一直放水的情況，也常在各種場合看到水龍頭忘記關或不斷漏水的情形，若是安裝本作品即可自動將水源截止，避免水資源無端浪費。
- 四、一般常見的節水設備上所標示的節省百分比，並沒有完全代表事實。究竟省了多少水，除了出水量，還有使用時間的因素，並不是用了節水設備就能安心。例如：有些節水墊片號稱省水 30%，但如果使用者因為洗不乾淨或是心想已經用了節水墊片，造成使用時間拉長，其節水效果就大打折扣，甚至反而有可能浪費更多水。因此節水別再靠感覺，省水百分比不精確，實際的使用量數據才是比較的基礎。
- 五、我們認為與其盲目使用節水設備，不如讓使用者確實知道每次的用水量。而有了數據，也就有比較基礎。例如：為什麼爸爸洗碗用水比媽媽多？洗澡用水多少公升算是合理？換了省水洗衣機和省水馬桶，能省多少水？使用者在洗手、洗菜、洗澡的當下，透過設備的數字跳動，可以立即知道自己用了多少水，隨時提醒不要浪費，淺移默化中達到省水教育的目的。
- 六、即時顯示用水量就像手錶提醒時間一般，數字跳動不斷提醒使用人用水量；而每次用水都記錄下來，立刻提醒大家用水情形，不是等看到帳單才知道用了多少水。凡用水必留下痕跡，更讓人沒有僥倖的心理。本作品在硬體上雖無加裝節水功能，但可由「心理上

提醒」使用人主動節水及檢討自己的用水行為，例如：使用沐浴乳時，應先把水關掉；或是刷牙時用水杯裝水可以更省水.....等，若再配合各種節水設備，更可達到節水目的。

七、本作品與市售智慧水表相較，更具有立即性、容易觀察，對於水資源的節約與教育意義更有效益。

## 柒、 結論

一、我們以 Arduino 來製作能即時顯示用水量並具有物聯網功能的水流計量設備，目前在硬體的購買及聯網的支援上，WIFI 模組會較藍芽模組在使用上更為便利。

二、就雲端資料庫的應用上，免費資料庫中 ThingSpeak 較 google 所提供的 TinyWebDB，除可顯示圖表外，將資料直接轉出為 excel 可讀的檔案，在使用上也較為簡單。

三、本作品硬體成本約 600 元，以每戶四人小家庭估計，安裝後每月約可節省水費 100 元，以成本和效益相比，具有相當大的價值。若是因此而能避免馬桶故障造成的耗水情形或是提早發現管路漏水而進行修繕，更是達到避免水資源浪費及節水省錢的多重性效益。

四、要達到節水目的，除了硬體上的各種節水設備外，如果能再加裝本作品，邊用水邊觀察用水量，一有用水過度情形，可及時提醒修正用水行為，並可由心理上讓使用人主動節水及檢討自己的用水行為。

五、本作品相較於市售智慧水表，除了更具有立即性、更便利觀察，對於用戶的成本效益及用水資訊回饋性，都更勝一籌。數字的立即跳動提醒，可達到自我約制，對於水資源的節約與教育意義效益更高。

六、每天透過實際的數字化視覺感受用水變化，而非憑自己的感覺來用水或節水。讓省水教育不再是空泛的感覺和口號，而是在每次用水時都能讓使用者看到數據，實際的感受到水的流逝。透過智慧水流計量設備的隨時提醒，從家庭中培養珍惜水資源的節水觀念，是當前最重要的課題，也是本研究最大的成就與價值。

## 捌、 參考資料及其他

- [1]、 台灣自來水公司，2010，《98 年度永續報告書》，臺中：經濟部台灣自來水公司。
- [2]、 經濟部水利署，2005，《水資源白皮書》，第一版，臺中：經濟部水利署。
- [3]、 郭淑珍，2012，《大臺北地區飲用供水跨域治理問題之研究》，國立臺灣大學社會科學院政治學系碩士論文。
- [4]、 經濟部水利署，經濟部水利署網站，<http://www.wra.gov.tw/>。
- [5]、 趙英傑，2014，超圖解 Arduino 互動設計入門(第 2 版)，臺北：旗標出版股份有限公司。
- [6]、 CAVE 教育團隊，2012，Android 手機程式超簡單!! App Inventor[入門卷]，臺北：泰電電業股份有限公司。
- [7]、 App inventor TW 中文學習網，<http://www.appinventor.tw/>。
- [8]、 賴亭瑋、董廷倫、秦丕臻、李柏宸，2015，點滴在心頭，教育部 2015 年全國能源科技創意實作競賽。
- [9]、 智慧城市與物聯網。打造智慧水表聯網 創新市民智慧生活，  
[http://smartcity.org.tw/application\\_detail.php?id=35](http://smartcity.org.tw/application_detail.php?id=35)。
- [10]、 蘋果日報，北水處首推智慧水表 用水量隨時監看，  
<http://www.appledaily.com.tw/realtimenews/article/new/20151202/745038/>

## 附件 1 家庭耗水量調查問卷

### 家庭耗水量調查問卷

同學(或家長)您好，為了進行「生活節水」之研究，將針對家庭用水請教下列幾個問題，請您協助填寫並於6月18日前繳回。大家一起為節能減碳、推動節約用水盡一份心力，感謝您。

1. 你知道你們家最近這期(二個月)用水量總共幾度嗎?請參考水費單，或詢問家長。(1度等於1000公升)

水號 (Account Number)				收費年月	繳款日期	應繳總金額
大區	中區	戶號	樓	(Year/Month)	(Payment Date)	(Total Amount Due)
				104年04月	104/05/05	
基本資料				項目	金額	
水表號碼:				口徑/基本費		
本期指針:				基本費:		
上期指針:				水費:		
國家水回收料/新水處理費:				雜費:		
本期實際用水量:				污水下水道使用費:		
本期應繳水費:				水源保護與回饋費:		
上期應繳指針:				電子帳單回饋金(退還):		
分期應繳水費/戶數:				節水優惠獎勵金額:		
應繳水費數:				本期金額:		
用次計費期間:				C 逾期罰款:		
下次抄表/收費日期:				違約金:		
代繳帳號:				舊欠金額:		
				應繳總金額(元):		

知道，\_\_\_\_\_度。

本來不知道，查詢後知道，\_\_\_\_\_度。

2. 你知道你們家中哪個水龍頭或用水設備耗水量最多?(例如洗衣機、廚房水龍頭、主臥室蓮蓬頭、馬桶...等)

知道，是\_\_\_\_\_

不知道(也請猜一個)，可能是\_\_\_\_\_

3. 承上題，你知道該水龍頭或用水設備，每月或每日耗水量多少?(單位:公升)

知道，約\_\_\_\_\_公升/每\_\_\_\_\_ (請填月或日)

不知道。

4. 為了節約用水，你的家庭有"實際做過"的省水方式為何?(可複選)

水龍頭裝節水墊片  換裝節水標誌的設備或省水馬桶

回收水再利用  會影響原本用水習慣，裝了又拆

太麻煩了，不想裝  其他

5. 面對地球暖化、氣候嚴重變遷、降雨減少，全球逐漸缺乏水資源。臺灣今年也出現乾旱及分區停水，雖然臺北市未面臨停水危機，請教你們家庭有進行各項之節約措施嗎?

有，加裝各項節水設施(包括馬桶內放置寶特瓶)

有，父母(或學校、廣告等)有看到節水宣導，我有配合導正用水習慣

沒有，因為不知道如何節水

沒有，因為我家用水不多，不用省水

6. 面對水資源危機，水價調整在即。請問如果免費安裝可以簡單測量家庭用水的設備，提醒每日用水量或水費，你覺得會讓你因此而更願意節約用水嗎?

會

不會

個人資料:

1. 我姓 \_\_\_\_\_

2. 我家住在\_\_\_\_\_市(縣) \_\_\_\_\_區(鄉鎮)

3. 我的家庭用水人口數。\_\_\_\_\_個大人、\_\_\_\_\_個小孩

## 附件 2 Arduino 控制程式 第一代藍芽 (節錄主程式部分)

<pre> void loop() {  // 判斷是否藍芽傳輸開關是否按下 serialA=I2CBT.read();  boolean SW1 = digitalRead(SW); //讀入第 3 腳值 if (SW1 != lastStateSW) { // 如果和之前的藍芽傳輸開關值不 同...     delay(20); // 等待 20 毫秒 boolean SW2 = digitalRead(SW); // 再讀取一次藍芽傳輸開關值     if (SW1 == SW2) { // 如果確認兩次開關值一致，就 往下執行... lastStateSW = SW1; // 儲存開關的狀態 clickSW ++; // 增加訊號變化次數 Serial.print(lastStateSW); Serial.print(" , "); Serial.println(clickSW);     } }     if (clickSW == 2) { // 如果開關狀態改變兩次 clickSW = 0; Serial.print(" transfer"); Serial.print(" , "); Serial.println(clickSW); digitalWrite (ledPin, HIGH);  //for (int k=0 ; k&lt;countBT ; k++) { //傳 3 次(避免失誤) serialA=I2CBT.read(); //讀入手機傳送值，如等於 49 表示可 傳送 if (serialA == 49){ delay(100);     for(int j=0;j&lt;(dataQ);j++) { //將所有資料傳送 I2CBT.write(transferData[j]);         //delay(100); //延遲以利傳送(效用不 明)     } serialA=0; //將此值歸 0，以利下次收到 49 時辨別     }     delay(100); //延遲以利傳送 Serial.print("Begin Transfer, dataQ="); Serial.println(dataQ); digitalWrite (ledPin, LOW); } // 判斷是否水流量計量器(霍爾開關)是否有值輸入 boolean b1 = digitalRead(waterSensor); if (b1 != lastState) { delay(debounceDelay); boolean b2 = digitalRead(waterSensor); if (b1 == b2) { lastState = b1; click++; </pre>	<pre> } } if (click== 2) { timeCount = 0; //有在轉動，停止使用時間歸 0 flagA = 1; //設定旗標，在有轉動時為 1 click = 0; ledState = !ledState; digitalWrite (ledPin, ledState); waterCount++;     waterCountLiterTMP0 = waterCount/float(rotationCount);     waterCountLiterTMP1 = byte(waterCountLiterTMP0);     waterCountLiterTMP3= (100*waterCountLiterTMP0 - 100*waterCountLiterTMP1); //TMP3 為整數     waterCountLiterTMP2 = waterCountLiterTMP3; //同樣值，但轉為浮點小數。因為後面要除以 100 waterCountLiter = waterCountLiterTMP1 + (waterCountLiterTMP2/100); // 顯示幕顯示(略)     } else { timeCount++; if (timeCount&gt;= stopCount) { timeCount = 0; display.clearDisplay(); waterCountLiterTotal = waterCountLiterTotal + waterCountLiter; // 已經不轉動 5 秒，可將累計數值存入         if (flagA == 1){ transferData[dataQ] = month(); dataQ = dataQ + 1; transferData[dataQ] =day(); dataQ = dataQ + 1; transferData[dataQ]=hour(); dataQ = dataQ + 1; transferData[dataQ]=minute(); dataQ = dataQ + 1; transferData[dataQ]=byte(waterCountLiter); dataQ = dataQ + 1; transferData[dataQ]=waterCountLiterTMP2; transferData[1] = dataQ; dataQ = dataQ + 1; flagA=0; //將旗標歸 0，避免重複存入數值         }     } waterCountLiter = 0; waterCount = 0;     } } } } </pre>
--	--

### 附件3 Arduino 控制程式 第二代藍芽 (節錄主程式部分)

<pre> void loop() {   //(前略)   if (click== 2) {     timeCount = 0;          //有在轉動，停止使用時間歸 0     flagA = 1;              //設定旗標，在有轉動時為 1     click = 0;     ledState = !ledState;     digitalWrite (ledPin, ledState);     waterCount++;     waterCountLiterTMP0 = waterCount/float(rotationCount);     waterCountLiterTMP1 = byte(waterCountLiterTMP0);     waterCountLiterTMP3= (100*waterCountLiterTMP0 -     100*waterCountLiterTMP1); //TMP3 為整數     waterCountLiterTMP2 = waterCountLiterTMP3;     //同樣值，但轉為浮點小數。因為後面要除以 100     waterCountLiter = waterCountLiterTMP1 + (waterCountLiterTMP2/100);      // text display tests     display.clearDisplay();     display.setTextSize(1);     display.setTextColor(WHITE);     display.setCursor(0, 0);     display.print("NOW");      display.setTextSize(2);     display.setCursor(40, 0);     if (waterCountLiter&gt;= 5) {       display.setTextColor(BLACK, WHITE);     } else {       display.setTextColor(WHITE);     }     display.print(waterCountLiter);     display.setTextSize(1);     display.setCursor(120, 8);     display.setTextColor(WHITE);     display.print("L");      display.setTextSize(1);     display.setTextColor(WHITE);     display.setCursor(0, 24);     display.print("Total");     display.setTextSize(1);     display.setCursor(50, 24);     display.print(waterCountLiterTotal + waterCountLiter);     display.setTextSize(1);     display.setCursor(120, 24);     display.print("L");     display.display();     } else { </pre>	<pre> timeCount++; serialA=I2CBT.read();          //讀入手機傳送值，如等於 49 表示可     傳送     if (serialA == 49){       delay(100);       for(int j=0;j&lt;(dataQ);j++) { //將所有資料傳送         I2CBT.write(transferData[j]);       }       serialA=0;                //將此值歸 0，以利下次收到 49 時辨別     }     if (timeCount&gt;= stopCount) {       timeCount = 0;       display.clearDisplay();       waterCountLiterTotal = waterCountLiterTotal + waterCountLiter; //       已經不轉動 5 秒，可將累計數值存入       if (flagA == 1){          //如果旗標為         1，表示剛才有轉動，可將值存入         transferData[dataQ] = month();         dataQ = dataQ + 1;         transferData[dataQ] =day();         dataQ = dataQ + 1;         transferData[dataQ]=hour();         dataQ = dataQ + 1;         transferData[dataQ]=minute();         dataQ = dataQ + 1;         transferData[dataQ]=byte(waterCountLiter);         dataQ = dataQ + 1;         transferData[dataQ]=waterCountLiterTMP2;         Serial.println(dataQ);         transferData[1] = dataQ;         dataQ = dataQ + 1;         flagA=0;                //將旗標歸 0，避免重複存         入數值       }       waterCountLiter = 0;       waterCount = 0;     }   } } </pre>
--	--



# 附件 5 App inventor 2 按鍵接收藍芽資料

The image displays a complex set of App Inventor 2 code blocks for a Bluetooth data receiver application. The code is organized into several event-driven blocks:

- when Respond Click:** Initializes global variables: `global DataRVcount` to 1, `global DataOutcount` to 1, `global value_Total` to 0, and sets text and background color for `textBox1` and `textBox2`.
- when Screen1 Initialize:** Sets `BTList` to enabled, `textBox1` to disabled, `BTconnected` to enabled, and `clock1` timer to disabled.
- when BTList SelectedClicking:** Sets `BTList` elements to `BluetoothClient` addresses and names.
- when Disconnect Click:** Sets `BTList` to enabled, `textBox1` to disabled, `BTconnected` to disabled, `clock1` timer to disabled, and `textBox2` background color to white.
- when BTList AfterPicking:** Calls `BluetoothClient1` Connect with address `BTList` SelectedItem. Then sets `BTList` to disabled, `textBox1` to enabled, `BTconnected` to true, `clock1` timer to true, and `textBox2` background color to white.
- when Clock1 Timer:** Calls `Faucet01` OpenAll and `BluetoothClient1` Send1ByteNumber with number 0. It then checks for `BluetoothClient1` BytesAvailableToReceive. If true, it receives text and unsigned byte numbers. It uses a `while` loop to process `global DataRVcount` bytes, parsing them into `global value_Month`, `global value_Day`, `global value_Hour`, and `global value_Minute`. It also receives `global value_AD` and updates `global value_Total`. An alarm is triggered if `global value_AD` is 2. It stores values in `Faucet01` and `TimWeb01` and increments `global DataRVcount`. It updates `totalUseWater` text and checks for `global DataOutcount` bytes to send. If true, it receives `global DataOutTemp` and updates `global DataOutTemp`.
- Initialization blocks:** A series of `initialize global` blocks set up variables: `text`, `value_Day`, `value_Hour`, `value_Minute`, `value_Total`, `value_Month`, `text_Alarm`, `DataOutcount`, and `DataOutTemp`.

附件 6 App inventor 2 手機控制接收藍芽資料

The screenshot displays the code blocks for an App Inventor 2 application designed to receive Bluetooth data. The code is organized into several event-driven blocks:

- when Record1.Click**: Initializes global variables including `global DataRVcount`, `global DataOutcount`, `global value_Total`, `global uploadCount`, `global dataQ`, and sets the background color of `TextBox2`.
- when TinyWebDB1.ValueStored**: Sets `TextBox2` text to a URL and `global uploadCount` to 100, then calls `BluetoothClient1.SendByteNumber` with the value 43.
- when DTList.BeforePicking**: Sets `DTList.Elements` to `BluetoothClient1.AddressesAndNames`.
- when DTList.AfterPicking**: Checks if `BluetoothClient1.Connect` is successful. If yes, it sets `DTList.Enabled` to false, enables `TextBox1`, `Disconnect`, and `Clock1`, and sets `global uploadCount` to 0.
- when Screen1.Initialized**: Sets `DTList.Enabled` to true, `TextBox1.Enabled` to false, `Disconnect.Enabled` to false, `Clock1.TimerEnabled` to false, and initializes various global variables for date and time components.
- when Clock1.Timer**: A complex block that checks for data to receive. It uses `BluetoothClient1.BytesAvailableToReceive` and `BluetoothClient1.ReceiveText` to get data. It then processes the data to extract month, day, hour, minute, and second, and formats them into a date string (e.g., "2019/1/1 12:00:00"). It also calculates `global value_AD` and triggers an alarm if the value is greater than or equal to 4.
- when Faurest02.StoreValue**: Saves the `global DataRVcount` to the database.
- when TinyWebDB1.StoreValue**: Saves the `global DataRVcount` to the database.
- while test**: A loop that checks if `global DataOutcount` is less than `global dataQ`. If true, it retrieves the value from `Faurest02`, updates `TextBox1` with the data, and increments `global DataOutcount`.
- set global uploadCount**: Resets `global uploadCount` to 0 and increments it by 1.

## 【評語】 030803

1. 設計內容切合生活，提醒用水，節約資源。
2. 在利用檢測元件上有其貢獻。
3. 在受測者時間上似乎可以有延長的空間。
4. 討論使用者習慣的前後改變，也是可以深入的地方。