

運用 Arduino 並結合 RFID 與藍芽智慧水杯墊

陳榮靜
朝陽科技大學
資管系

e-mail :
rungching@gmail.com

陳明家
朝陽科技大學
資管系

e-mail :
ming40324@gmail.com

摘要

現代人因忙碌導致每日飲水量不足，銀髮長者更是要注意。有鑑於部分老年人可能會有不能喝太多水的疾病，例如：腎臟功能不好的老人。所以本研究設計一個運用智慧型水杯墊，可以有效解決在每日飲水量的控管並透過藍芽傳送，了解長者飲水狀況。主要技術在於重量感測與藍芽傳送並結合 RFID 標籤識別，再經由智慧型手機分析去提醒用戶該喝水了。我們以每日飲水的加總去判斷，是否飲用不足或者是過量，運用這個設備可以讓忙碌的人或長者提醒自己，達到每日建議的飲水量。

關鍵詞：Arduino、藍芽、RFID、智慧型手機。

Abstract

Due to elderly habits take daily water insufficiently, they used to more attention to intake water. Some of the elderly have the disease that forbid them to drink too much water, for example; poor kidney function disease. This paper proposes an intelligent design system to effectively resolve the daily intake water control using system. Bluetooth is utilized to transfer the information of the intake water from the elderly. The system calculates whether the total daily intake water they take is inefficient or over. The system uses weight sensor, Bluetooth and RFID technology. Smartphone is utilized to remind the elderly whether elderly whether they are recommended to drink more water.

Keywords:Arduino, Bluetooth, RFID, SmartPhone.

1. 前言

如今，高齡化社會趨勢日漸成長，人們的

生活品質相對提高，對於長者的健康也更加重視，提早預防老年出現的一些疾病，加上現今族群不規律的生活，常常忘記自身水分是否充足。我們提出智慧杯墊，在杯上貼上 RFID TAG 由重量感測器所測量到的數值藉由藍芽傳送到智慧型手機，立即顯示該杯子所有者與日常飲水量，經由這些水量加以分析，達到是否正常或超量飲水，同時也可以運用到銀髮老人的飲水監控。

本研究使用的 Arduino 開發板作為開發平台，它可以接上藍芽棒、重量感測器、RFID Reader 來作出智慧杯墊，藍芽消耗功率低，傳輸可靠性非常高，可以做為智慧型手機與杯墊溝通的橋樑，配合 APP，演算法計算出日常飲水量，判斷是否超標，實現飲水量的監測。

本論文接下來主要對硬體與軟體做論述，章節安排如下：第二節相關研究將對於 Arduino 領域與 Arduino 電子秤及所需零件進行說明，第三節系統架構將介紹系統的架構與運作。系統實現，將介紹本研究運用相關開發實作，陸續完成實驗的過程，第四節實驗結果分析實作所產生的數據，第五節將對於本研究進行未來展望與結論。

2. 相關研究

Arduino 開發版是的開源的主機板，配合各種感測器進行有趣的搭配：如陽光感測器加上 Arduino 就可以做一個小氣象觀測站，土壤感測器加上濕度感測器可以做為植物園的應用，層出不窮的創意引起創客 (Maker) 的注意，成本不高，每個人都可以是開發者，本篇使用了重量感測器 (Load Cell)、藍芽棒、RFID Reader，做出一個簡易的水杯秤。

Android Studio 是一套由 Google 所推出的平台開發程式，在 2013 年推出，可以供開發者免費使用，網路也有眾多範例可以讓開發者應用，SDK 部分也整合的相當完整，是一套比 App Inventor 功能更齊全的軟體。可以模擬眾

多 Android 機種，讓開發者兼容各類機型。

2.1 硬體

Arduino 是一塊開放原始碼發展的主機板(圖 1)，擁有豐富的 I/O 介面，Java、C 語言的開發環境，讓使用者可以輕易上手。Arduino 上有開關、數位訊號、類比訊號、感測器插槽、步進馬達等，延展的擴充性，可讓創客(Maker)在這平台上自由發揮。

Arduino UNO R3 在硬體上與其他的 Arduino UNO 無異，它是改用 ATmega328，所需提供電量為 5V，使用的軟體依舊是官方的 Arduino IDE 進行編譯及燒錄[1]。

藍芽模組 HC-06 來作為和手機端溝通的工具，供電要求只要 3-5V 即可驅動。主控端或從端模式，出廠前就設定好，不能更改；市面上販售的通常是「從端」模式[2]。本研究將採用 HC-06，如圖 2 所示。



圖 1 Arduino UNO R3 主機板

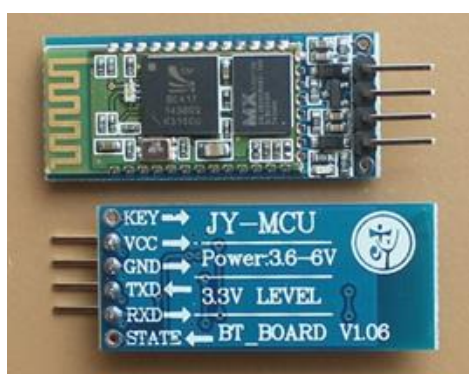


圖 2 Bluetooth 藍牙棒

電子秤感測模組所採用的是 HX711 晶片，為高精度電子秤的 24 位元 A/D 轉換晶片。特色是擁有穩壓電源、鐘振盪器等其他同類型所需的外圍電路，反應速度快，不易受到其他干擾，提高了整機的性能和可靠性[3]。本研究將採用 HX711 作為重量感測的原件。



圖 3 電子秤感測模組

本論文採用兩個圓形的壓克力板來作為秤的底及待測面，中間連接的金屬條為應變梁，應變梁中間封膠是電阻應變片，如圖 4 所示。受重量的作用應變片發生形變導致電阻值發生變化，該變化即為重量的數值。本篇可感測的重量值上限為 5KG。

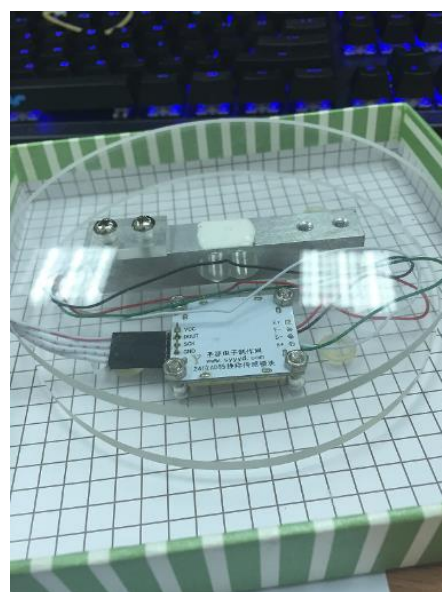


圖 4 5KG 電子秤

Arduino 擴充板是一個結合各種感測器的 Arduino 擴充板，上面有個麵包板可以接上許多杜邦線，同時還擁有許多插槽和感測器[4]。

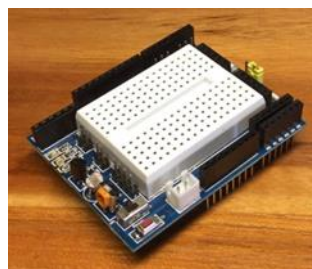


圖 5 Arduino 擴充板

RC522 RFID 感測器是應用於 13.56MHz 非接觸式通信中高集成度的讀寫卡晶片，是 NXP 公司針對「三表」應用推出的一款低電壓、低成本、體積小的非接觸式讀寫卡芯片，是智能儀表可攜式手持設備研發的較好選擇[5]。標籤採用被動標籤，不必使用電源，只要黏貼在杯底就可以了[6]。

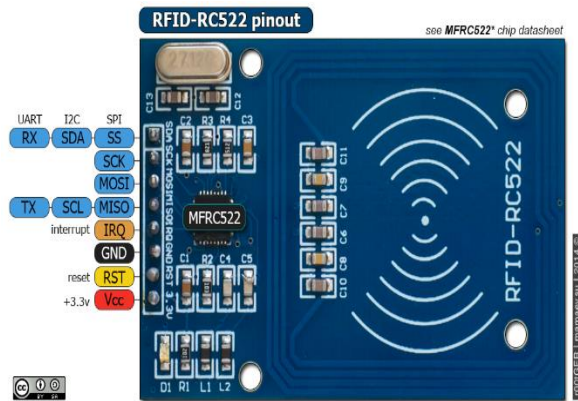


圖 6 RFID Reader RC522

2.2 軟體

使用 Arduino 所需要的開發環境、編譯器、燒錄程式、開發工具、Arduino 驅動程式，在 Arduino 官網上一應俱全，可在多平台上開發：MAC、Windows、Linux 等。因為 Arduino 本身是開源原始碼，在官網或討論區都有相關程式碼，所運用語言類似 C/C++，具備文字編輯介面，工具欄，並且開發程式 Arduino IDE 提供驗證程式碼功能，如果可以運行就可以直接燒錄到 Arduino 板子了。每個檔案我們稱為草稿簿 (Sketch)，腳本都寫在文字編輯視窗裡，序列埠監控視窗可以看出每當在量測重量時可以顯示當時重量。在文字編輯視窗裡分為設定腳位端、硬體端與控制端，本次使用藍芽與手機溝通，讓手機代替序列埠監控視窗的重量顯示。

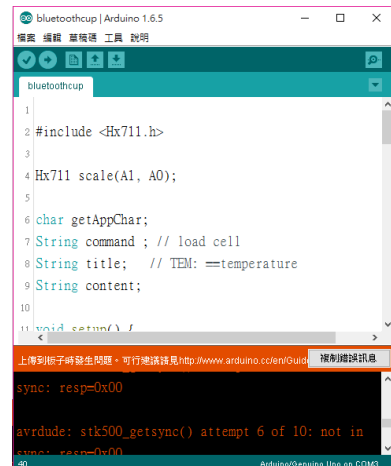


圖 7 Arduino IDE 操作畫面

在手機的 APP 端開發，我們使用由 Google 所提供的 Android Studio 軟體來撰寫程式，有比於以往 App Inventor、eclipse，更加的相容性且穩定，搭配上目前主流的 Android 5.0.1 平台來作為開發基準，並以 Google 推出的 Nexus 6P 作為開發實機測試[7]。

Android Studio 是一個完全免費的軟體，可以提供開發者在各種 Android 平台上進行開發，並且提供模擬器與最新最齊全的 Android 版本，每一期間都會有新的 SDK 來提供用戶安裝，是一個非常優秀的 APP IDE 軟體。

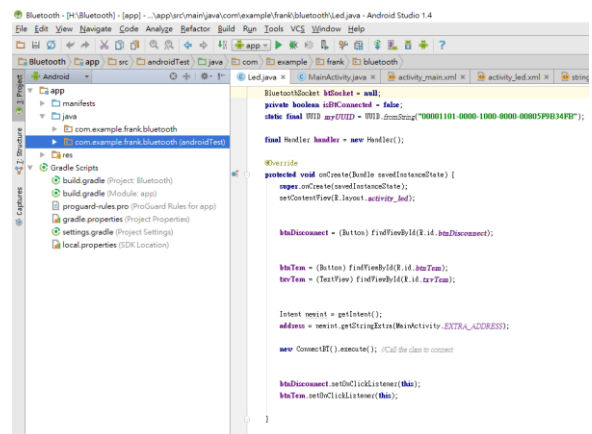


圖 8 Android Studio 操作畫面

3. 系統架構與實現

傳統的量秤可能無法應付精準的簡易醫學測量，人們對於量測結果準確度要求越來越高，如果誤差太大，可能會造成配量失準。傳統量秤體積較為龐大，誤差也較不精準，這都是我們所要克服的困難。

本論文中的藍芽電子秤分為軟體及硬體，整個系統架構圖如圖 9 所示。

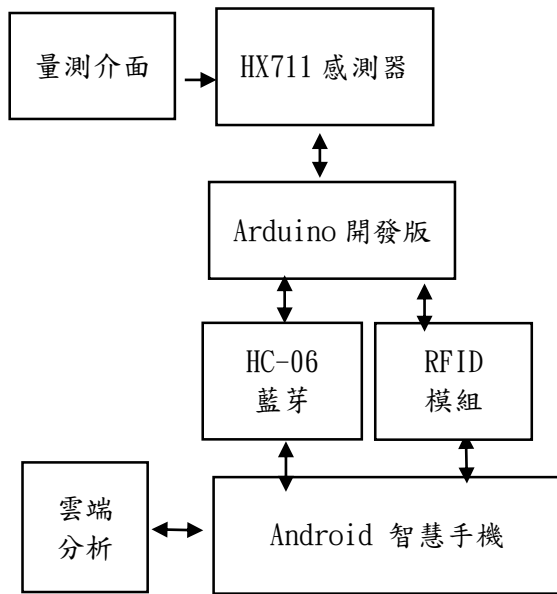


圖 9. 智慧杯墊軟硬體架構圖

貼有 RFID 標籤的水杯放置於杯墊上，經由 RFID 識別到，再經由 HX711 秤傳感器感測到的變化，將傳送到 Arduino 開發板上，利用藍芽將所量測到的數據傳送到手機分析並且可以傳送數據到雲端，進行歷史資料分析。

本文系統實現是基於硬體設計與軟體開發兩部分進行。

3.1 硬體設計

硬體設計的過程如下：是使用 HX711 24 位元 A/D 轉換晶片來當作重量感測晶片，可以看到圖 10 示意圖，應變梁處於中空狀態，在重力作用下，會導致應變梁上面的產生形變，引起電阻發生變化，送出信號。特別注意的一點，應變片有方向性，如果裝反會有「負重量」出現，本實驗可測得最大重量為 5KG，此重量能滿足一般水杯含水的量測。



圖 10 Arduino 與 秤連接圖

3.2 軟體開發

本文所研究的電子秤是採用藍芽的方式與智慧型手機作為溝通的媒介，採用低功率藍芽模組達到省電功效，使用藍芽發送由重量感測器量測到的重量數據發送到手機 APP 端接收。藍芽電子秤軟體部分分為兩端，使用 Arduino IDE 將程式碼燒錄到 Arduino UNO R3 裡，另外一端是 Android 程式，安裝在於智慧型手機，提供顯示量測的重量與判斷。

Arduino 軟體端是用於燒錄撰寫的程式碼，將程式碼寫入 Arduino UNO R3 中，並有下判斷，使得感測器得以成功運作[8][9]。手機 APP 端採用 Google 所提供的 Android Studio 進行編寫，設計了藍芽(連接/斷開)、RFID(是否啟用判斷)、RFID 標籤識別、重量顯示、手機水量數據資料庫。

首先，我們先將管理者所設定使用者生理參數的 RFID 標籤讓使用者貼在自己的杯底下，再將之放到電子秤上，取得「空杯」重量，使用者再將開水加入到杯子，放到磅秤上開始記錄飲水量，此時演算法就會扣掉空杯重量，取得真正的飲水量，倘若使用者突然又放上水杯，此時就會有 5 分鐘的緩衝期間，避免太頻繁的量測，造成測量失準。

Android Studio 與 Arduino 去發送硬體端發出的信號，再由 APP 端接收解析訊號，進而達成軟硬體搭配，水杯先經校正儲存，杯子重量及對應 RFID 標籤的使用者 ID。

為了計算出一天所喝的飲水量，我們設計了一個演算法，設 $V = \{V_1, V_2, \dots, V_n\}$ 為每一次喝完水之後所量測的水量且為一個向量，設 N 為建議飲水量， N 的值是以個人資料對照表 2 所取得的建議值，其中個人資料包含 AGE(年齡)、S(性別)，首先透過參考表和年齡以及性別去取得建議飲水量，接著計算一天的飲水量，最後在對比建議飲水量，即可得到飲水狀況的評估結果，演算法如下表所示：

表 1 水杯測量演算法虛擬碼

Function WaterDrinking	
1.	Input:P, Age //輸入體重、年齡
2.	set i=1 //次數 1
3.	if age<18 //年齡小於 18 歲
4.	R, age, sex→N //參照表年齡、性別
5.	if age>18 //年齡大於 18 歲
6.	N=P*30 //個人每日所需飲水量

```

7. While  $V_{max}$  //次數最大值
8.   if  $V_i - V_{i+1} > 0$  //水差額大於 0
9.     Sum=sum+( $V_i - V_2$ ) //開始加總水量
10.    i=i+1
11.   if  $V_i - V_{i+1} < 0$ 
12.     i=i+1 //水差額小於 0
13. if sum>N*1.1 //當 sum 大於 N 的 10%
14.   S=過量
15. if sum<N*0.9//當 sum 小於 N 的 10%
16.   S=不足
17. if sum<=N*1.1 and sum>N*0.9
    //介於 N 中間
18.   S=適量
19. Output S //輸出狀態
20. end

```

首先輸入 P(體重)，在我們的演算法中提到 $P*30$ 會等於 N，因為人體每日飲水量是依體重乘上 30 的 C.C. 為建議的飲水量，如果在年齡小於 18 歲，依照我們的採用查詢表 2 的方式，畢竟年紀過小的小孩飲水量不能太多。

V_{max} 為最後一筆的次數，當到達最後一筆次數便會做加總，我們將取得所有的差額值來當作日飲水量，但是有一點必須注意的是，當差值為正數，將不予已加入計算，因為這可能是使用者加入的新水量，這樣會導致量測數據失準。

最後在演算法，將加總結果使用三種級距去做狀態的分析，第一是當差額加總大於 N 的 10%，屬於「過量」飲水，第二是當差額加總小於 N 的 10%「不足」，第三種狀態則是不屬於在上述兩種狀態內，屬於「適量」。

日飲水量公式： 30 C.C./1KG ，也就是說體重 70 公斤的人，每天需要補充 2100 C.C 的水，超過 3000 C.C 即為過量飲水[10]。

實驗流程：

Step 1:

先放置已設定好參數的 RFID TAG 並貼好在杯子下，將其放置於杯墊上測量。

Step 2:

RFID Reader 識別到 TAG 參數(使用者的體重輸入到 TAG，杯子的重量)。

Step 3:

經由 APP 提示，開啟手機藍芽進行與秤的連線。

Step 4:

經由演算法所算出的重量，取得當前飲水量。

Step 5:

將所取得的飲水差額加總即為當日飲水量。

Step 6:

飲水總量將紀錄於手機內的資料庫，每日會進行統計，推播消息給使用者知道是否已達本日飲水量。

表 2 針對不同年齡的每日飲水量[11]

Age	Boys/Men	Girls/Women
0-6 months	0.7 L/day	0.7 L/day
7-12 months	0.8 L/day	0.8 L/day
1-3 years	1.3 L/day	1.3 L/day
4-8 years	1.7 L/day	1.7 L/day
9-13 years	2.4 L/day	2.1 L/day
14-18 years	3.3 L/day	2.4 L/day
19-70+ years	3.7 L/day	2.7 L/day

表 2 這是由 Bin Dai 等所提出的建議飲水量，它是以年紀、性別作為區分，在演算法中以年齡判斷低於 18 歲以下的使用者，我們將用查表方式直接判斷飲水量。

由以上演算法推算出適合使用者的飲水量是否充足。符合飲水量畫面呈現「笑臉」做為示意，如果您的飲水量是 2100 C.C，我們允許的範圍為 $\pm 10\%$ ，都是 OK 的，反之超量就是「哭臉」。然而符合飲水量我們還在下面做貼心提示，如：本日飲水量稍嫌不足喔，還差 500c.c 請再喝點水！多喝水沒事，沒事多喝水 [12]！

4. 實驗結果

首先，初步實驗我們蒐集了兩位長者(以下稱為長者 A、長者 B)的資料作為本次實驗的測試，透過我們的智慧水杯墊計算出需要的飲水量。在表 3 我們先蒐集用戶的一些生理指數及特殊病史，再去計算出推薦飲水量。

在此將設定好的 RFID 貼片貼在杯子上，將杯子放置於杯墊上，手機經由 APP 連接藍芽，按下 Connect，搜尋到杯墊的設備，此時 RFID 識別到資料庫裡的 RFID TAG 參數，進入量測介面此時按下「開始測量」，畫面即會出現透過重量感測器所識別的重量，經由演算法推算是否已達飲水量，狀態將用圖案表示，更輕易的了解，在於每次測量結束後便會儲存該數值，直到當日結束即會推算本日是否達到該有的飲水量。

表 3 長者個人基本資料部分樣本

	長者 A	長者 B
體重	70KG	50KG
特殊病史	無	腎臟病

表 4 杯墊偵測水量資料部分樣本

Date	Tag ID	飲水量
2016/01/01 AM 08:00	NA001	250c. c
2016/01/01 AM 09:30	NA001	500c. c
2016/01/01 PM 12:00	NA001	200c. c

表 5 長者個人基本資料部分樣本

	長者 A	長者 B
飲水量	1150c. c	800c. c
	700c. c	600c. c
	250c. c	450c. c
總計水量	2100c. c	1850c. c
是否超量	適量	過量

在表 4 呈現每次長者放置水杯，智慧杯墊會記錄飲用的水量在我們的資料庫裡，並藉由 RFID Reader 識別這是哪位長者的杯子。

在表 5 我們可以看到長者 A 的飲水量是符合所推薦的飲水量，反觀長者 B 是已經超量的狀態，而且長者 B 患有腎臟病，是不允許過量喝水。對於一些嚴重腎臟病患者和水腫患者，他們是需要限制水的攝入量的。因為腎臟的嚴重受損會導致水裡中的鈉滯留，多喝水將會加重水腫、惡化病情，所以他們需要限制水量 [13]。

透過本實驗我們了解到，運用 Arduino 可以開發出生活上方便的小物，再透過藍芽與智慧型手機連線，雖然秤會有小許的誤差 ($\pm 0.2g$)，這都是可以在接受範圍內，因為是電流及溫度或者是秤的品質無法達到專業級量秤，但已經足以讓本實驗獲得成效。

5. 結論及發展

本論文提出使用智慧水杯方式讓長者在無意間量測到一天所飲用的水量，透過數據分析去統計歷史水量，可以提供醫療方面的資訊，讓有特殊病症的患者更加控管飲水量。

在這物聯網的時代，我們可以使用這種開發板進行一些小實驗改善我們的生活，現在人都擁有一支智慧型手機，將兩者結合一起，搭配網路或感測器，就可以開發出本實驗的一個實物來改善生活品質，成本花費不高，所獲得的成就感也很大。任何實驗，數據是非常重要的，利用演算法來推算出每日飲水量，即便秤的精準有些誤差，但只要將這些情況設定在一個允許的範圍內，都是可以接受的。利用藍芽與手機的溝通，讓這些資料在手機上分析，利用圖案的方式呈現出來，就可以達到自我監督的效果，簡單明瞭。

未來我們將對於秤的重量感測器，找尋是否有更小更精準的零件，來縮小秤的尺寸，並針對飲水這部分來給使用者更多建議，如幾歲的人或 BMI 多少的人適合的飲水量、結合雲端技術做出歷史資料統計，用大數據分析去規畫個人整個飲水狀況，讓健康漸入佳境。對於每次的飲水狀況有很多環境變量的可能，例如使用者裝滿途中忘記放置到杯墊上量秤，導致沒回傳到資料庫內等，這些都是造成失準的可能。

參考文獻

- [1] M. Suresh, P. Saravana Kumar and Dr. T. V. P. Sundararajan "IoT Based Airport Parking System," *Coimbatore*, pp. 1-5, 2015.
- [2] HC-06 藍芽, <http://swf.com.tw/?p=693>。
- [3] HX711, <http://nxtmarket.info/item/16515045079>。
- [4] Arduino 擴充板, <http://software.gotop.com.tw/hardware/Item.aspx?id=A2014042309>。
- [5] RFID Reader Spec, <http://bbs.wandianlian.com/forum.php?mod=viewthread&tid=72>。
- [6] Dequan Liu, Long hao and Zehua Gao "A Practical Internet of Things System for Mine Application," *Energy Procedia*, Vol. 11, pp. 3336-3340, 2011.

- [7] Nexus 6P Spec , <http://chinese.engadget.com/2015/11/23/nexus-6p-review/> 。
- [8] 趙英傑，**超圖解 Arduino 互動設計入門 第2版**，旗標出版社，2014/04/24。
- [9] 葉難，「Arduino 輕鬆入門 範例分析與實作」，博碩出版社，2014/02/27。
- [10] 飲水量算法，<http://hospital.kingnet.com.tw/essay/essay.html?pid=24138> 。
- [11] Bin Dai, Rung-Ching Chen, Shun-Zhi Zhu and Chung-Yi Huang “Recommendation System for Daily Water Intake – Design and Realization Based on Fuzzy System,” *Dalian*, pp. 1-6, 2015.
- [12] 林杰樑，**水是百藥之王**，元氣齋出版社，1992/08/15。
- [13] 水分補充夠不夠，<http://myshare.url.com.tw/show/431853> 。