

中華民國第 54 屆中小學科學展覽會

作品說明書

高中組 生活與應用科學科

最佳創意獎

040812

智慧型節水供水系統

學校名稱：臺北市立內湖高級工業職業學校

作者： 職三 王聖閔 職三 朱詠誠 職二 許智堯	指導老師： 張俊雄 張志成
---	-----------------------------

關鍵詞：電磁閥、微控制器、即熱式電熱水器

摘要

冬天能洗個舒服且讓身心放鬆的「熱水澡」是很幸福的，可是一般我們在淋浴時，都會先開啟水龍頭讓水慢慢變熱，但你是否想過，在冷水變成熱水之前的那些冷水去哪呢？好像是流進排水溝全都浪費掉了。目前很少有任何一款市售的產品能夠瞬間就加熱到適當的溫度，這也就是「即熱式電熱水器」的缺點。為了節省大自然的水資源，一個嶄新的「智慧型節水供水系統」應運而生。在此，我們建構一個以單晶片為核心的微電腦控制系統，使其能有效地控制管路中的電磁閥，再加上溫度感測裝置能讓水溫達到我們所設定的溫度之後再行出水，未達溫度的冷水也能儲存回流之後再使用，這樣不僅可以解決熱效能不彰的問題，又毋需忍受那濕冷的感覺，也不會造成水資源的浪費。

壹、研究動機與目的

冬天使用熱水來洗澡，幾乎是每一個人都會做的事。但很少有人分得清楚「即熱式電熱水器」與「儲熱式電熱水器」有何不同？顧名思義，後者必須有一大小固定的容器來儲存熱水，如此一來將耗損不少的電能來加熱。為了有效節省電力能源，多數採用即用即熱的方式，可是當我們把水龍頭打開之後，它需要慢慢地加熱到一定溫度才行。目前很少有任何一款市售的產品能夠瞬間就加熱到適當的溫度，在這之前所流出來的冷水就浪費掉了，這也是即熱式電熱水器的缺點。因此，如果我們能讓水溫達到我們所設定的溫度之後再行出水，未達溫度的冷水也能儲存回流之後再加熱，讓使用者能夠直接享用熱水，這樣不僅可以解決熱效能不彰的問題，又毋需忍受那濕冷的感覺，也不會造成水資源的浪費，真是一舉數得。

為了達到這個目的，我們將高工三年所學習到的理論與實習充分發揮運用。首先，透過『冷凍空調實習』課程所學習到的知識與技術，完成一個基本的「加熱供水系統」。然後，再結合資訊科『專題製作實習』課程的概念，建立一個以單晶片為核心的微電腦控制系統，使其能有效控制管路中的電磁閥。在此我們選用 Arduino 公司所開發之 Mega 2560 微電腦控制平台，加上溫度感測裝置，將特定功能需求之程式碼(參考高三『資訊實務』課程所學到的 C 語言)寫入微控器的晶片中，當溫度達到我們所設定的數值之後才啟動電磁閥讓熱水流出，而未達溫度的冷水儲存後再抽回使用，一個嶄新的「智慧型節水供水系統」應運而生。

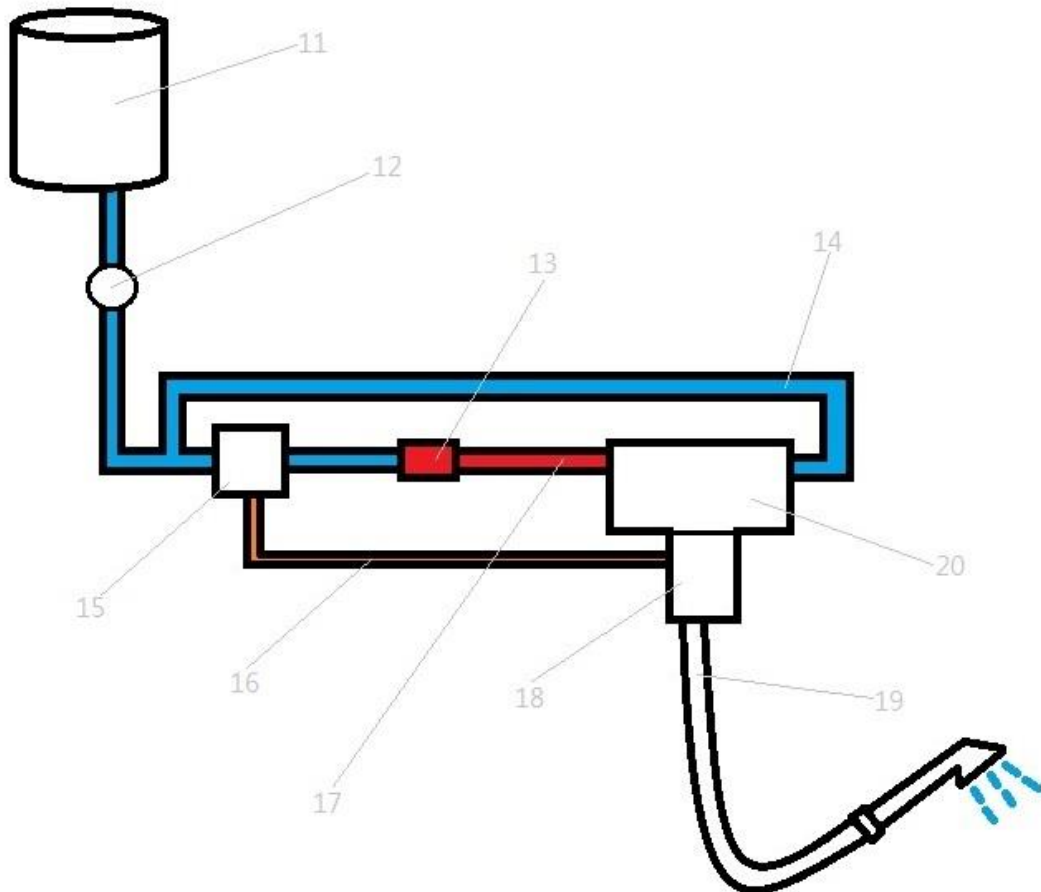
貳、研究器材與設備

名 稱	規 格	數 量	備 註
個人筆記型電腦	Mac OS, Intel i7, 16GB RAM	1	
微控制器開發版	Arduino Mega 2560	1	
七段顯示器	0.65"	3	
帶燈按鈕	有段	1	
方形按鈕	無段	2	
電熱水器	220V	1	即熱式
管徑轉接頭	6分轉4分	4	
管徑轉接頭	4分轉3分	4	
銅管	3分	些許	
銅T型接頭	3分	2	
銅由令	3分	10	
電磁閥	3分 24V	2	
繼電器	5V	1	
熱水鐵管	6分	2	
感溫棒	防水型	1	
電阻	4.7K Ω	1	
電阻	220 Ω	24	
水龍頭	6分	1	
電源適配器	110V 轉 24V	1	

參、研究過程或方法

一、主架構設計

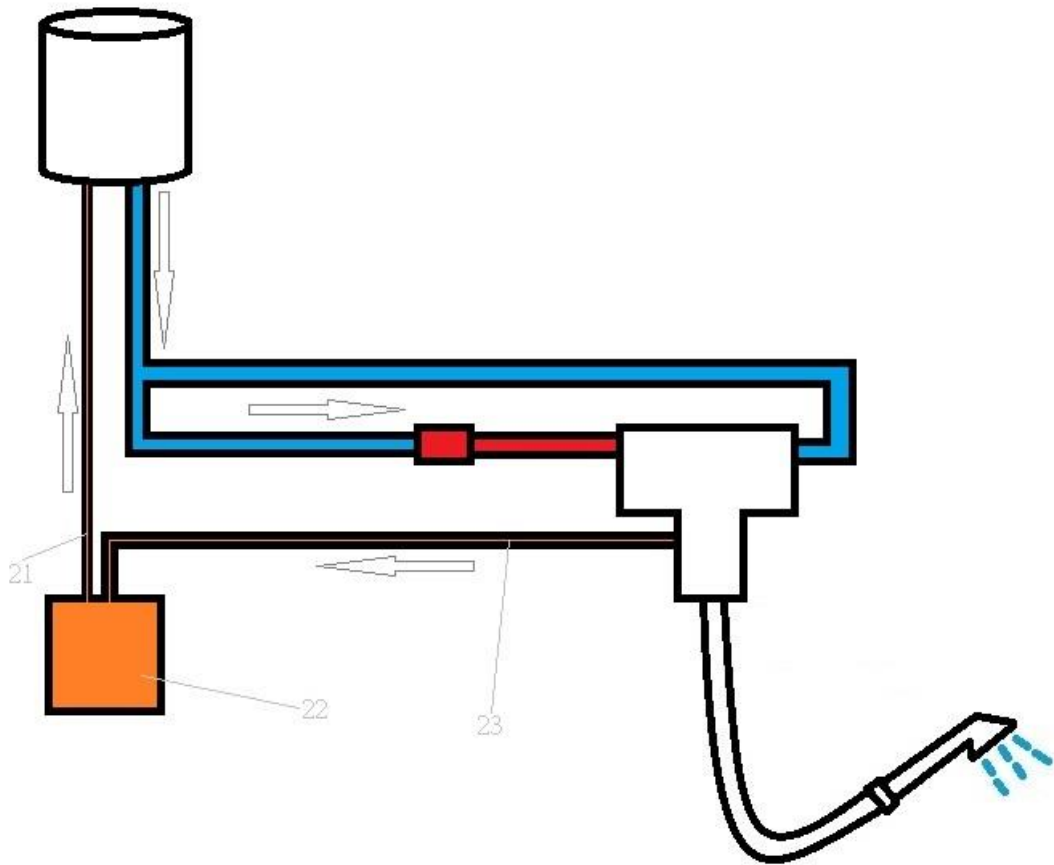
一般大樓或公寓的加熱供水方式是從水塔經由水表，再到熱水器加熱後從水龍頭出水。而在我們改良的「智慧型節水供水系統」中，是在水塔之後、熱水器之前加裝了一個蓄水槽，收集從出水處回流來的水，並再次流向熱水器繼續加熱，完整的系統結構，如圖一。



圖一 系統結構(一)

11：水塔	16：回流管
12：水表	17：熱水管
13：熱水器	18：管路控制電磁閥
14：冷水管	19：出水管
15：蓄水槽	20：水龍頭〈或電動冷熱水調節閥〉

不過，為了作品展示的整體性及製作的方便，我們將系統稍作簡化，省去了水表，並將一些管線的路徑做了修改。先把原本的蓄水槽拿掉，回流的冷水直接回流到一個獨立的蓄水槽，而這個獨立蓄水槽內的水，會經由抽水馬達抽回水塔之中，修改後之系統結構，如圖二。電腦模擬設計，如圖三。



圖二 系統結構(二)

21：抽水馬達將蓄水槽裡的水抽回水塔的管線路徑

22：蓄水槽

23：回流管

※ 箭頭方向為管路內水流的方向

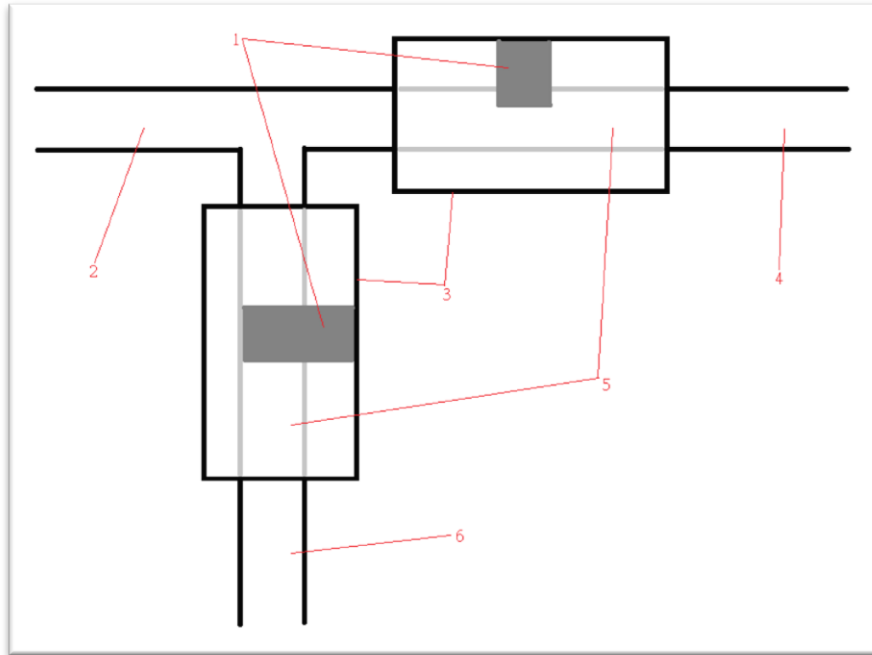


圖三 電腦模擬設計圖

二、電磁閥之管路控制

我們的加熱供水系統，是把傳統浴室的水龍頭換成能調整水流量的三通閥，只要在調整處裝上伺服馬達，並將伺服馬達連結至 **Arduino** 控制板，然後在調整水流量的三通閥出水口處，再加上一個控制出水或是讓水回流至蓄水槽的管路控制電磁閥，即可完成。不過，此次研究重點在於讓冷水回流達到節水的效用，而非冷熱水的比例控制。因此，冷熱水的調整沿用原本浴室用的水龍頭，至於電動控制則期待後續的研究再與系統做整合。

管路控制的運作方式為：先在水龍頭出水口處裝設兩個電磁閥，兩個電磁閥及電磁閥的電源，分別連接到繼電器的常開接點、常閉接點及共同端點。使得原本兩個相同的電磁閥，分別成了常開電磁閥以及常閉電磁閥。常開電磁閥接至出水管，一般情況為不通；常閉電磁閥接至回流管，一般情況則是暢通。這樣的裝配方式可以使得未達額定溫度的水，流向常閉電磁閥的回流管，待溫度到達之後，控制板會切換繼電器，使得常開繼電器得以導通，水就可以從出水管流出，如圖四。

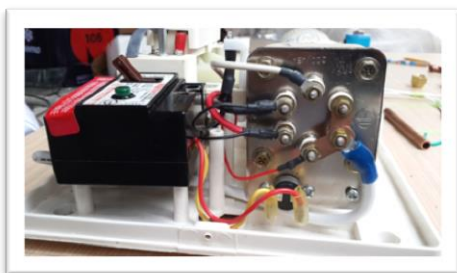


圖四 電磁閥之管路控制示意圖

- | | |
|-------------|-----------|
| 1：電磁閥內管路控制閘 | 4：回流管 |
| 2：給水來源 | 5：電磁閥內部管路 |
| 3：管路控制電磁閥 | 6：出水管 |

三、加熱裝置

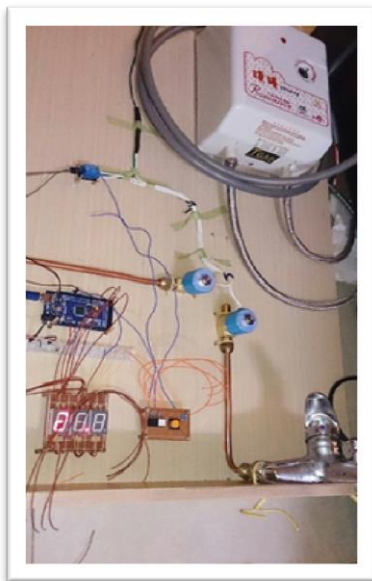
加熱裝置方面有數個可行的方案，分別為瓦斯熱水器、自製電熱絲以及電熱水器。選擇瓦斯熱水器能夠讓此次研究的結果更為接近實際狀況，但要連接瓦斯桶，處理不良可能造成瓦斯外洩，因此較不安全且不方便搬運；自製電熱絲雖然對於系統建構來說比較方便，體積也比較小，但功率較小、效率較差且較不安全；電熱水器跟瓦斯熱水器都是家庭中常用的加熱裝置，不過電熱水器不會像瓦斯熱水器有瓦斯外洩的風險，用來啟動的電也比較好處理。經討論後，決定用電熱水器較為適當，如圖五。



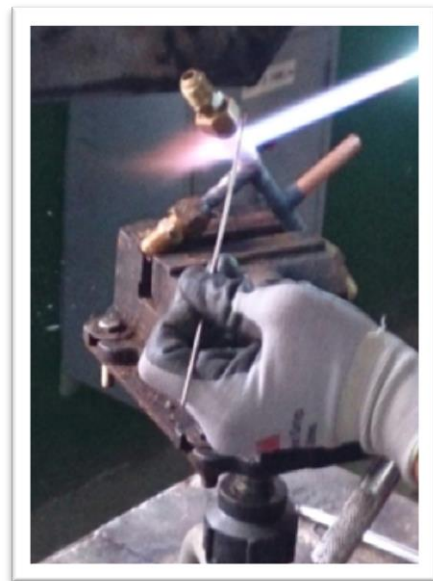
圖五 電熱水器內部構造

四、智慧型節水供水系統展示板

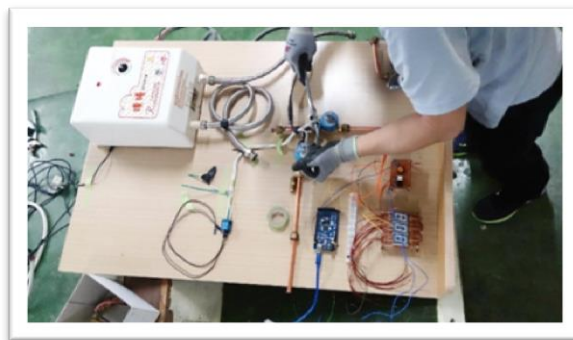
在此次研究中，我們計畫將整套系統模擬並展示在一塊板子上，如圖六。主要包含有：電熱水器、控制回流或出水用的電磁閥、水龍頭、Arduino 控制板、控制電磁閥的繼電器，以及切換冷熱水模式的開關。製作過程如下：第一，先焊接製作連通管，如圖七。第二，連接外部管線，如圖八。而蓄水槽及模擬的水塔不會在展示板上，將置於旁邊或後面，利用水管把連接起來。另外，展示板內需要通水的管路部分，將會用銅管加以連接。



圖六 智慧型節水供水系統展示板



圖七 焊接製作連通管



圖八 連接外部管線

五、微電腦控制平台

微控制器 Arduino Mega 2560 (如圖九) 是整個系統的核心，它是創用 CC (Creative Commons) 開放的電路設計，使用類似 Java 與 C 語言的開發環境。可簡單的與各式各樣的電子元件連接，官方網站也提供大量的函式庫以供參考，電源部份則由 USB 介面來提供，也可以獨立運行使用，建議電源輸入 7~12V。Arduino 微控制器 Mega 2560 的電氣特

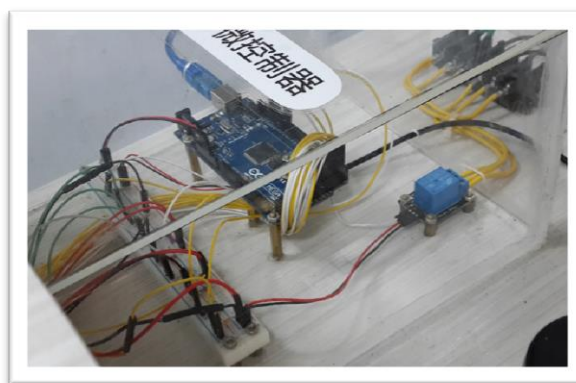
性規格如表(一)：

表(一)

工作電壓	5V
電源建議輸入電壓	7-12V
電源極限輸入電壓	6-20V
數位 I/O Pins	54 (其中提供 15 個 PWM 輸出)
類比輸入 Pins	16
DC Current per I/O Pin	40 mA
DC Current for 3.3V	50 mA
Flash Memory	256 KB
SRAM	8 KB
EEPROM	4 KB
Clock Speed	16 MHz



圖九 Arduino Mega 2560 微控制器



圖十 微電腦控制平台

將溫度感測裝置與 Arduino 微控制器的輸入端連接，用以感測目前管線內之水溫。在輸出端的地方，連接七段顯示器，用來顯示預設（使用按鍵開關來作設定）的溫度值。連接完成之後的微電腦控制平台，如圖十。

六、程式碼撰寫

經過四個版本的修改之後，將最有效率及最佳使用體驗的核心程式，寫入 Arduino 微控制器內，程式碼如下：

```
//匯入程式庫  
#include <OneWire.h>  
#include <DallasTemperature.h>  
#include <Wire.h>
```

```

#define ONE_WIRE_BUS 9//定義 9 接腳為感溫棒傳值之匯流排
OneWire oneWire(ONE_WIRE_BUS);//使用程式庫之函示取得感溫值
DallasTemperature sensors(&oneWire); //使用程式庫之函示轉換感溫值

int relay = 10;//繼電器控制接腳
int bled = 11,rb = 12;//切換開關及指示燈之接腳
int plus = 6,minus = 7;//加 , 減按鈕之接腳

int hot = 30;//預設定義熱水溫度
byte seven[10][7] = {
    { 0,0,0,0,0,0,1 }, // = 0
    { 1,0,0,1,1,1,1 }, // = 1
    { 0,0,1,0,0,1,0 }, // = 2
    { 0,0,0,0,1,1,0 }, // = 3
    { 1,0,0,1,1,0,0 }, // = 4
    { 0,1,0,0,1,0,0 }, // = 5
    { 0,1,0,0,0,0,0 }, // = 6
    { 0,0,0,1,1,1,1 }, // = 7
    { 0,0,0,0,0,0,0 }, // = 8
    { 0,0,0,1,1,0,0 } // = 9
}; //共陽七段顯示器使用陣列

//初始化
void setup(){
    //定義接腳輸出輸入
    pinMode(rb,INPUT);
    pinMode(plus,INPUT);
    pinMode(minus,INPUT);
    pinMode(bled,OUTPUT);
    pinMode(led,OUTPUT);
    pinMode(relay,OUTPUT);
    for(int i=22;i<43;i++){
        pinMode(i,OUTPUT);
    }
    //讓感溫棒進行初始化
    sensors.begin();

```

```

}
//重複執行動作
void loop(){
    //主按鈕按下，指示燈亮
    if(digitalRead(rb) == LOW){
        digitalWrite(bled,HIGH);
        sensors.requestTemperatures();//呼叫感溫棒傳回數值
        temp(sensors.getTempCByIndex(0));//轉換數值，呼叫七段顯示器顯示目前溫度
        if(digitalRead(plus)==HIGH){ //預設數值加
            hot = plus_click(hot);
        }
        else if(digitalRead(minus)==HIGH){ //預設數值減
            hot = minus_click(hot);
        }
        //判斷是否到達指定溫度，控制繼電器
        if(sensors.getTempCByIndex(0) >= hot){
            digitalWrite(relay,LOW);
        }
        else{
            digitalWrite(relay,HIGH);
        }
        delay(200);//延遲 0.2 秒
    }
    else if(digitalRead(plus)==HIGH){ //預設數值加
        plus_click(hot);
    }
    else if(digitalRead(minus)==HIGH){ //預設數值減
        minus_click(hot);
    }
    else{//持續讓繼電器不動作，七段顯示器繼續顯示設定溫度
        buttonpm(hot);
        digitalWrite(bled,LOW);
        digitalWrite(relay,LOW);
    }
}
}

```

//預設數值加

```
int plus_click(int hot){  
    if(hot < 99){  
        hot++;  
        buttonpm(hot);  
        delay(200);  
        return hot;  
    }  
}
```

//預設數值減

```
int minus_click(int hot){  
    if(hot >= 1){  
        hot--;  
        buttonpm(hot);  
        delay(200);  
        return hot;  
    }  
}
```

//分割設定溫度數字進行顯示

```
void buttonpm(int d){  
    one((d/10)%10);  
    two(d%10);  
    three(0);  
}
```

//分割感溫數字進行顯示

```
void temp(float d){  
    one((int)(d/10) % 10);  
    two((int)d%10);  
    three((int)(d*10)%10);  
}
```

//控制第一顆七段顯示器 (即十位數)

```
int one(int d){  
    int j=0;  
    for(int i=22;i<29;i++){  
        digitalWrite(i,seven[d][j]);
```

```
        j++;  
    }  
}  
  
//控制第二顆七段顯示器 ( 即個位數 )
```

```
int two(int d){  
    int j=0;  
    for(int i=29;i<36;i++){  
        digitalWrite(i,seven[d][j]);  
        j++;  
    }  
}
```

```
//控制第三顆七段顯示器 ( 即小數點後一位數 )
```

```
int three(int d){  
    int j=0;  
    for(int i=36;i<43;i++){  
        digitalWrite(i,seven[d][j]);  
        j++;  
    }  
}
```

肆、研究結果

「智慧型節水供水系統」之溫度控制面板，如圖十一。展示板上之『電源』啟動開關，是為了控制熱水模式（ECO 模式）或冷水模式。當旋轉鑰匙開關啟動電源之後，按下橘黃色有段式按鈕、指示燈亮，熱水模式立即被啟動。控制電路上之繼電器（Relay）開始動作，使得電磁閥轉為閉合狀態，同時改變水流的方向。經由電磁閥的控制，若熱水溫度未達預設指定溫度（初始值為 30 度）之前，冷水將會從回流管流入蓄水槽；只要當熱水到達指定溫度後，出水電磁閥開啟，熱水就會從水龍頭正常流出。當按鈕彈起、指示燈滅，再度轉變為冷水模式，此時控制電路的繼電器不動作，回流電磁閥關閉、出水電磁閥開啟，冷水直接從水龍頭流出。另外，“+”與“-”按鈕之作用為溫度設定之用，可以輕鬆調整自己喜歡的熱水溫度，範圍為 0~99。至於顯示器在按鈕未動作時，則顯示預設的熱水溫度，當按鈕一啟動，則顯示目前熱水管內水的溫度。

設置切換開關的理由是，當我們只使用熱水時（冷水管不出水），且管內溫度已到達設定溫度的情況之下，出水管電磁閥開啟，回流電磁閥關閉。若此時水龍頭卻轉往冷水，會導致囤積在銅管內的熱水逐漸降溫，將使得溫度感測棒傳回的數值達不到預設值，而再次進行加熱，如此一來反而達不到節能的效果。



圖十一 溫度控制面板



圖十二 管線水流方向示意圖

「智慧型節水供水系統」之管線水流方向示意圖，如圖十二。管線中的水是從較高處的水箱流進電熱水器，經電熱水器加熱後連接到主銅管，再由兩個電磁閥開關來決定接下來兩支副銅管的走向。其中，兩個電磁閥都接上繼電器（Relay）進行一打開一閉合的控制，若出水電磁閥導通，且水龍頭位於開啟位置，水就會從水龍頭出來；如果回流電磁閥導通，水就會回流至低處的蓄水槽。實驗結果證明，本系統能讓水溫達到我們所設定的溫度之後再行出水，未達溫度的冷水也能儲存回流之後再加熱，讓使用者能夠直接享用熱水，這樣不僅可以解決熱效能不彰的問題，又毋需忍受那濕冷的感覺，也不會造成水資源的浪費。

究竟我們的「智慧型節水供水系統」能省下多少的水費？反之，要將回流至低處蓄水槽的水重新抽回水塔之中，又需要耗費多少的電費？兩者的比例是否能夠符合成本效益呢？以下的討論將詳細說明：假設我們採用 12 公升的蓄水槽來儲水，使用 1/4HP 的抽水馬達來抽水、且其抽水量為每分鐘 180 公升、馬達所消耗的功率為每小時 186.5W (= 0.25HPx746W/HP)。因此欲抽完 12 公升蓄水槽內的儲水只要 4 秒鐘 (=1/900 小時)，而它所消耗的電量為 $0.1865\text{KW}/900=2.072\times 10^{-4}$ 度電。已知目前台灣電價為 2.1 元/度，所需之電費為 4.35×10^{-4} 元。又目前水費的計價方式乃每 1 公噸為 1 度，且台北自來水價為 5 元/度，因此節省 12 公升的水可以省下 6×10^{-2} 元的水費。換言之，所消耗的電費與所節省的水費之間的比例約為 1:138，從成本效益的觀點來看，本「智慧型節水供水系統」的確是值得推廣的。再者，根據 102

年台北自來水事業統計年報，大台北地區家庭用戶共有 1,476,793 戶，每戶每天若能節省 12 公升的洗澡用水，一年將可以省下 6,468,353 公噸的自來水，數據說明了對於大自然水資源的節省確實有很大的幫助。

伍、討論

在微電腦控制平台中所使用之溫度感測裝置，原本計畫是希望採用紅外線感溫裝置，由於硬體取得不易及精確度不高，只好作罷。其次，管線上的開關，原本打算使用電動調節閥：它可以自動調節冷水與熱水的比例，而達到恆溫的目的；但是我們的管線控制模式主要是做非比例調節，因此改用電磁閥來替代它。另外，在焊接管線時，中心雙公頭連接管比較不容易抓住施力點，而造成銅管損毀、甚至導致多次重覆焊接。最後，因為各種元件管徑不一，需要大量的轉接頭：熱水器（8/8 英吋）連接銅管（3/8 英寸）需經過 8/8 轉 6/8 再轉 3/8 三個轉接頭，而水龍頭 8/8 英吋連接到電磁閥 3/8 英吋也需 8/8 轉 6/8 再轉 3/8 三個轉接頭。上述之種種問題，最終也都找到解決的方法、或替代方案，雖然過程不算很順利，但基本上這次的研究結果是成功的。

陸、結論

實驗結果證明，我們的「智慧型節水供水系統」能讓水溫達到我們所設定的溫度之後再行出水，未達溫度的冷水也能儲存回流之後再使用，不會造成水資源的浪費。再者，經過詳細精算之後，系統所消耗的電費與所節省的水費之間的比例約為 **1:138**，從成本效益的觀點來看更是值得推廣的。另外，根據 **102** 年台北自來水事業統計年報，大台北地區家庭用戶共有 **1,476,793** 戶，每戶每天若能節省 **12** 公升的洗澡用水，一年將可以省下 **6,468,353** 公噸的自來水，數據說明了對於大自然水資源的節省確實有很大的幫助。

這次的研究將水回流再加熱，形成節能的運作，只是最基本的設計，假如要更節能、節水的話，後續的研究還有很多的改進空間。例如：首先，可以將水龍頭更換成電動調節閥，再配合紅外線感溫裝置，會更容易控制，真正落實智慧化、自動化的目標。其次，水塔容量、高度、管線路徑之長短，都是影響節水效能的重要因素，加熱速度越慢則節能效果會越顯著，也更能提高熱水器的效能。最後，可以善加利用高位置的水塔到屋內之間的水位能，來進行水力發電，直接供應節水系統所需之電力，或甚至可以儲存之後再提供其它家電來使用。

柒、參考資料及其他

- 一、孫駿榮、吳明展、盧聰勇 (2012)。最簡單的互動設計Arduino 一試就上手(第二版)。臺北市：碁峯資訊。
- 二、孫駿榮譯 (2013)。Arduino互動設計入門與應用。臺北市：碁峯資訊。
- 三、蔡明志 (2012)。樂在C語言(第二版)。臺北市：碁峰資訊。
- 四、許祺清、陳聰明 (2006)。冷凍空調原理。臺北市：弘揚圖書。
- 五、溫度感測 DS18B20 教學，取自：<http://yehnan.blogspot.tw/2013/01/arduinods18b20.html>。
- 六、七段顯示器範例，取自：<http://yehnan.blogspot.tw/2012/02/arduinoseven-segment-display.html>。
- 七、台北自來水事業處編印 (中華民國103年5月出版)。一〇二年台北自來水事業統計年報。ISSN 1812-3104。

【評語】 040812

1. 主題關心水資源和從日常生活需求切入。
2. 宜有相近且較廣用作品當對照組比較。
3. 宜多參考文獻中已有設計再求創新。