

中華民國第 58 屆中小學科學展覽會

作品說明書

高級中等學校組 電腦與資訊學科

第二名

052509

結合數據統計之感應式水龍頭

學校名稱：高雄市私立復華高級中學

作者： 職二 郭力瑋 職二 葉威廷	指導老師： 蔡宗勳
---------------------------------	------------------

關鍵詞：物聯網、節能、流量統計

摘要

水龍頭讓人們方便、快速、有效率的使用水資源，無處不有，隨著環保意識抬頭，水資源的運用出現了許多的議題。雖無法開源，但能節流。以下是本作品功能：

- 運用紅外線感測器偵測距離進行開關水龍頭，自動出水
- 將網頁寫入晶片，無須額外架站就能觀看、統計流量
- 利用流量計進行統計及控制水量，達到精準的水量控制
- 採分散式架構，各個水龍頭都是獨立的系統，不因其中一座故障而影響其它運作
- 將水龍頭分成不同模式，根據不同情境，以達到彈性用水
- 無須更改管線的採包覆式架構，比起常見的感應式水龍頭瑣碎的安裝過程，可直接裝置在傳統水龍頭上，節省安裝過程
- 運用課堂所學物聯網、3D 建模、程式設計，發揮學以致用的精神

研究動機

水是維持生命所必備的物質，地球僅只有 1%的水可使用[1]，在一般人的認知下台灣的降雨量十分充沛，每年平均都有約 2000 毫升的降雨量，但是台灣因為地狹人稠，山勢陡峭，雨勢集中，所以台灣是一個缺水國。如何節省水資源是台灣一個重要的議題[2]。

我們在高二的課程中接觸了物聯網、3D 列印、以及程式設計的課程，我們利用所學習的技術，以及相關的開發板，來製作一個感應式水龍頭，以達到節省水資源的效果，再透過許多的實驗以及數據分析水資源的使用量，希望我們利用在課程上所做出的作品能夠實際應用在生活上。

研究目的

- (1). 透過不同的旋轉角度來製作不同模式出水量的水龍頭。
- (2). 配合程式來控制旋轉角度，達到幫助使用者省水的效果。
- (3). 統計水龍頭的用水量，配合物聯網技術了解水龍頭用水情形。
- (4). 利用課程學習到的技術，來實現學以致用的精神。
- (5). 製作出讓使用者方便管理的系統，讓使用者統一管理。
- (6). 探討與一般水龍頭所使用的水量差距，計算所節省的水量。

研究設備及器材

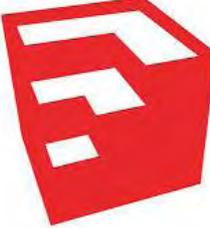
 <p>3D 印表機</p>	 <p>NodeMCU</p>	 <p>伺服馬達</p>	 <p>紅外線感測器</p>
 <p>PLA 環保 3D 列印線</p>	 <p>Sketch Up</p>	 <p>筆記型電腦</p>	 <p>K24 流量計</p>
 <p>止洩膠帶</p>	 <p>水龍頭</p>	 <p>Arduino IDE</p>	 <p>流量計</p>
 <p>LED 燈(RGB)</p>	 <p>量杯</p>	 <p>JavaScript</p>	

表 1 使用材料

研究過程與方法

系統架構

本研究架構以 **Arduino IDE** 撰寫程式[5]來控制伺服馬達的轉動角度，並將伺服馬達掛載在水龍頭的旋鈕上，藉由轉動角度的不同來控制水龍頭的流量;我們經過實驗後，以霍爾水流量計來計算水流量、並以 **3D** 列印技術來實作模型，將其包覆在傳統的水龍頭上面，並透過相關的電子元件來完成結合數據統計之感應式水龍頭。系統架構如下圖所示:



圖 1 系統架構圖

為了能夠完成上述實驗作品，我們的研究過程分為三個部分，分別為電路設計、3D 建模實作、以及控制程式設計。

電路控制:

電路控制部分，我們使用了伺服馬達來控制水龍頭的旋轉角度;NodeMCU V3(ESP-12E)開發板來作為控制的核心;紅外線感測器來偵測感測區域的靠近距離來判斷是否供水以及 RGB LED 來幫助我們識別目前的狀態。分別介紹如下:

1. 伺服馬達:[3]

伺服馬達(MG996)是利用 PWM 控制轉動角度，其控制的角度為 0~180 度，而控制的原理與機構是利用內部的減速齒輪與可變電阻組裝再一起，並在可變電阻上的齒輪加上擋柱，輸入對應角度的 PWM，即可達成動作要求。正因為可以透過控制其旋轉角度，因此可以用來控制水龍頭水閥的開啟程度，藉以控制出水量。



圖 2 伺服馬達

2. ESP8266(物聯網晶片):[7]

ESP8266 是一款 WiFi 模組，具有低功耗率、體積小、價格低廉的優點，透過 arduino IDE 工具以及大量的函式庫我們可以利用 C 語言實作出物聯網的應用設計。ESP8266 有很多版本，我們採用整合 ESP-12E 的 NodeMCU 開發版，因為此開發版引出大量控制腳位，方便我們可以接上各種感測器。它內建的 SPIFFS 檔案系統可以儲存靜態資料，方便我們儲存用水的紀錄(.json 格式)。



圖 3 ESP8266

3. 紅外線感測器:[4]

是一種距離測量傳感器組件，由一個 PSD (位置靈敏感測器)、IRED(紅外發光二極管) 和信號處理電路組成。採用三角測量法：檢測距離不易受到各種物體反射率或對環境的溫度和持續操作時間的影響。因不易受到干擾，所以這種感測器也可以用來作為接近感測器。我們使用的感測器型號為 GP2Y0A41SK0F，它的有效偵測距離為 4cm~30cm



圖 4 紅外線感測器

實驗(一):

為了能夠精確的測量感測距離，我們設計了第一個實驗，實驗方法是逐步測試紅外線感測器接收各種距離所對應的回傳電壓值。以下是程式碼及流程圖：

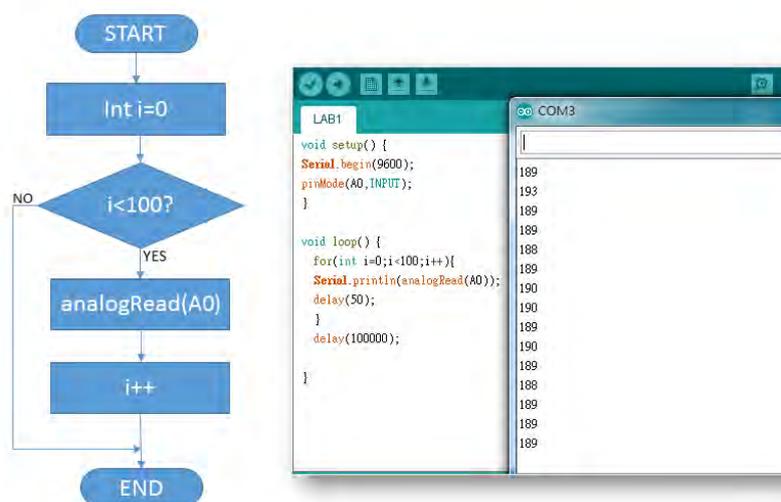


圖 5 距離感測程式碼及流程圖

透過上述程式流程，我們針對 4cm~30cm 每隔 1cm 進行一次實驗，每隔 50ms 取得回傳值，並取 100 次數值後取平均值，作為該次距離實驗的參考值。經過反覆實驗後我們得到大量數據如附錄 1 紅外線感測器測試數據，最後整理出下列平均數值：

距離	1cm	2cm	3cm	4cm	5cm	6cm	7cm	8cm	9cm	10cm
平均回傳值	NA	NA	NA	762.53	663.11	579.51	500.83	454.76	392.46	362.68
距離	11cm	12cm	13cm	14cm	15cm	16cm	17cm	18cm	19cm	20cm
平均回傳值	331.36	304.48	282.88	270.35	248.94	234.68	222.68	210.5	196.84	189.05
距離	21cm	22cm	23cm	24cm	25cm	26cm	27cm	28cm	29cm	30cm
平均回傳值	178.43	172.02	166.4	160.6	154.85	148.85	140.95	139.98	133.37	132.4

表 2 電壓與距離回傳值

根據上表我們繪出回傳電壓與距離的關係圖如下圖：

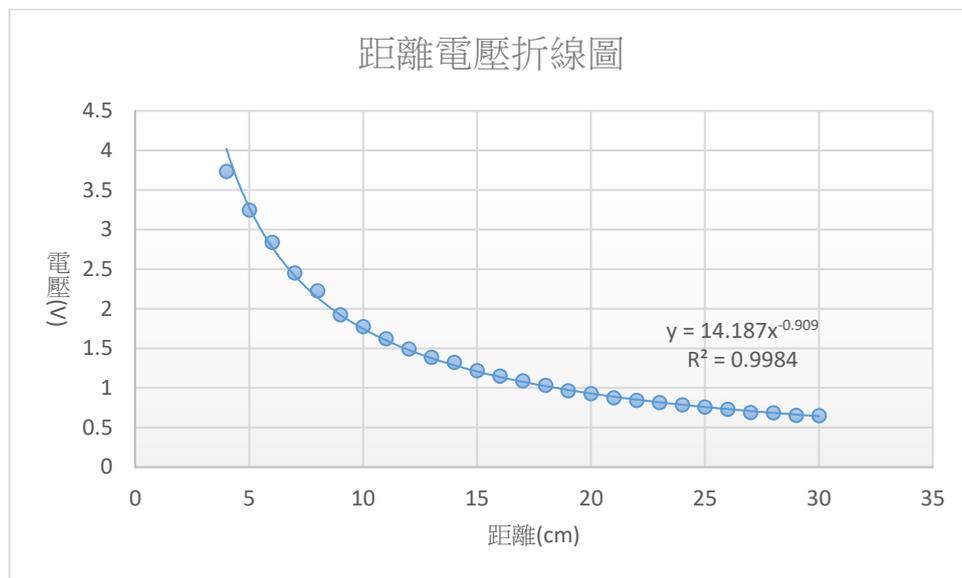


圖 6 距離電壓折線圖

我們得到距離與回傳值的關係方程式：

$$y = 14.187x^{-0.909}$$

其中 x 為距離， y 為偵測回傳值。我們可以利用此方程式來計算感應區是否有手靠近。

4. 電路組裝:[8]

利用 fritzing 電路設計軟體繪製了我們各部件的電路圖，NodeMCU、紅外線感測器、伺馬達，再將這些零件的電路線給接起來，繪製成我們需要的電路圖。

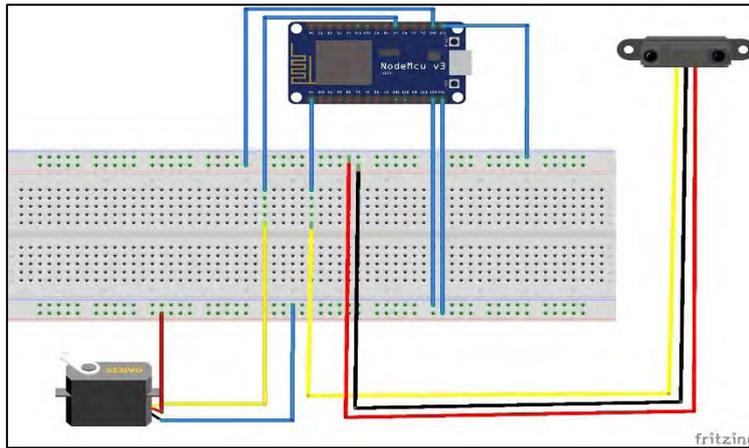


圖 7 電路圖

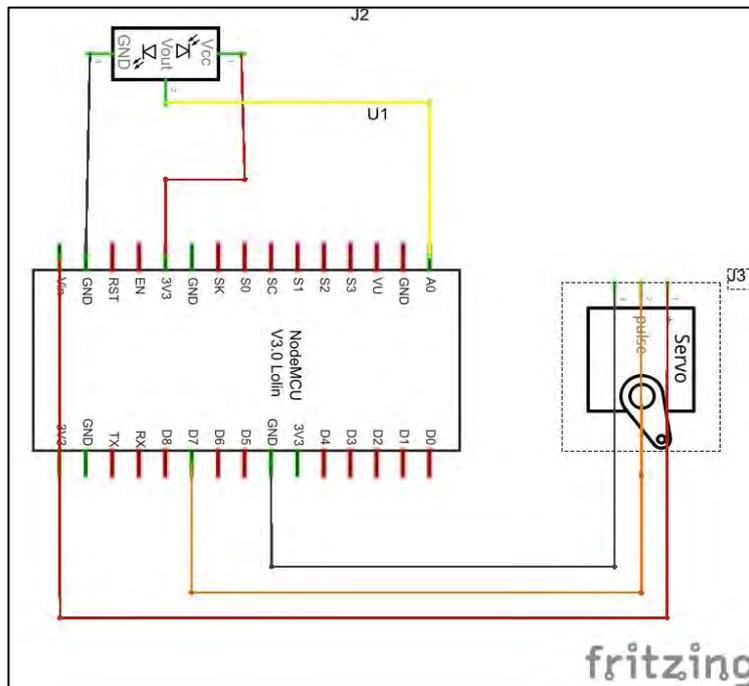


圖 8 電路圖

3D 列印:[6]



因為本作品需要能夠容納現有之水龍頭，因此 3D 建模是一大難題。我們透過游標卡尺量測水龍頭與相關電子元件的實際尺寸及細節，並透過 3D 建模軟體繪出模型，再透過 3D 印表機印出部分細節，來觀察印出的元件與水龍頭的誤差為多少，透過測試修正法來修正公差數據，最後實際印出模型的原型。

圖 9 零件測量

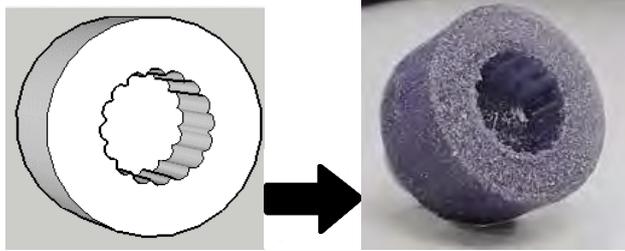


圖 10 旋鈕圖

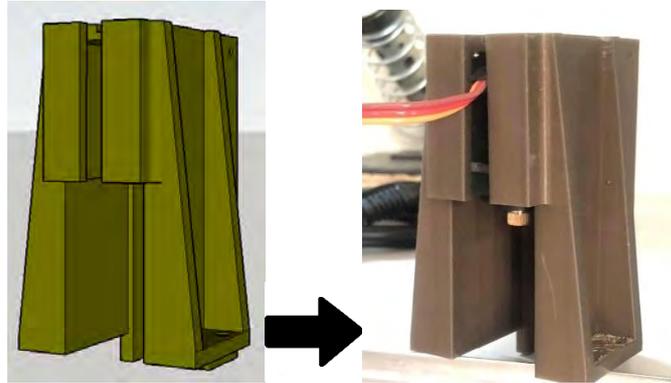


圖 11 伺服馬達基座及轉接頭安裝

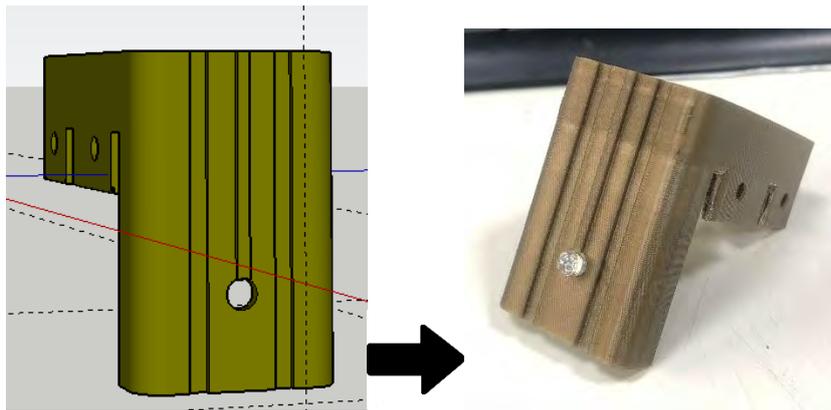


圖 12 判斷水龍頭的使用狀態的狀態面板

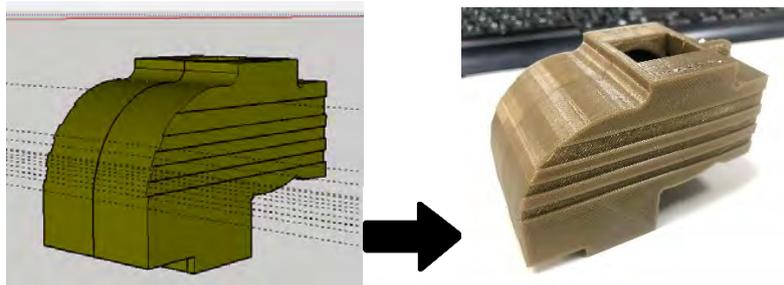


圖 13 水龍頭外殼

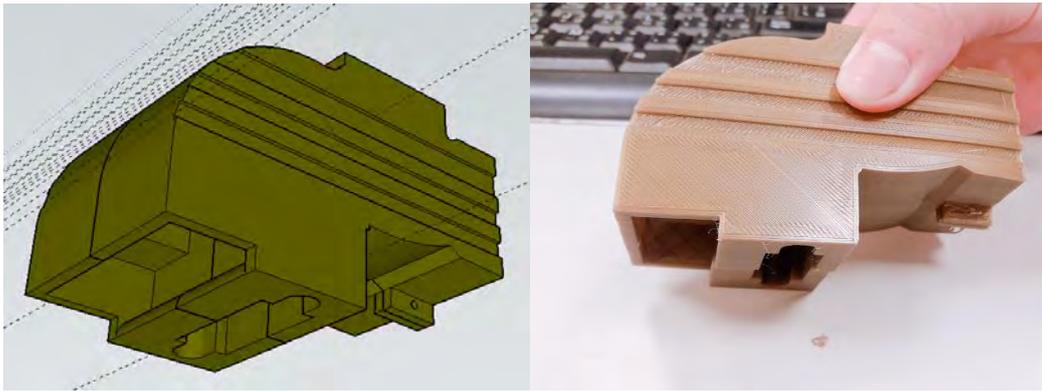


圖 14 外線感測器模組安裝位置

建模完成後，我們實際將傳統的水龍頭安裝包覆，並裝上紅外線感測器，如下圖所示：

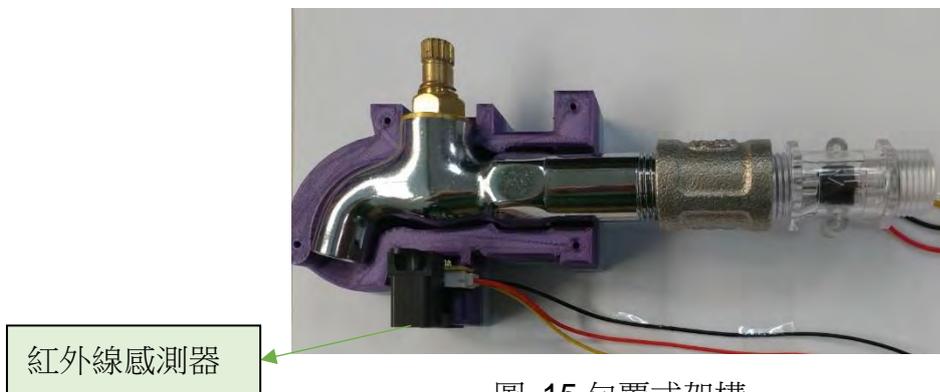


圖 15 包覆式架構

最後我們將感測器焊接到 NodeMCU 的對應腳位上，並且安裝伺服馬達以及馬達旋鈕轉接器，如下圖所示

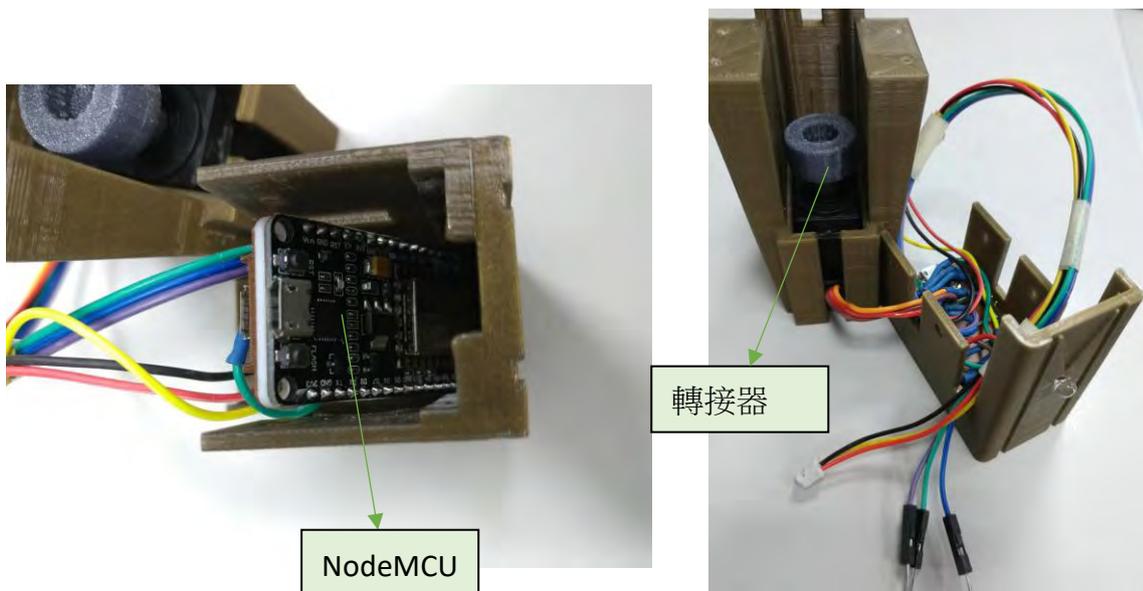


圖 16 各零件安裝位置

程式控制設計:

1. 網頁架設: 將網頁寫在 ESP8266 的 SPIFFS 上, 方便我們進行遠端控制。

```
type: 'line',
xValueType: 'dateTime',
yValueFormatString: '### c.c',
xValueFormatString: 'hh:mm:ss',
showInLegend: true,
name: '水龍頭1',
dataPoints: dataPoints1
}];

$("#m1").click(function(e){
$.get("http://" + ip + "/m1");
});
$("#m2").click(function(e){
$.get("http://" + ip + "/m2");
});
$("#m3").click(function(e){
$.get("http://" + ip + "/m3");
});
$("#m4").click(function(e){
$.get("http://" + ip + "/m4");
});
$("#m5").click(function(e){
$.get("http://" + ip + "/m5");
});
$("#m6").click(function(e){
$.get("http://" + ip + "/m6");
});

function toggleDataSeries(e) {
if (typeof(e.dataSeries.visible) === 'undefined' || e.dataSeries.visible) {
e.dataSeries.visible = false;
}
else {
e.dataSeries.visible = true;
}
}
```

```
//掛載檔案系統 //靜態檔案存取
if (!SPIFFS.begin())
{
// Serious problem
Serial.println("SPIFFS Mount failed"); //server.on("/", handleRoot);
} else {
Serial.println("SPIFFS Mount succesfull"); //網頁回應
}
server.on("/report1",report1);
server.on("/report2",report2);
server.onNotFound(handleNotFound);
if (MDNS.begin("esp8266")) {
Serial.println("MDNS responder started");
server.on("/hello",Hello);
}
server.on("/m1",M1);
//啟動server
server.on("/m2",M2);
server.begin();
server.on("/m3",M3);
server.on("/m4",M4);
server.on("/qm",QM);
//啟動UDP request
UDP.begin(UdpPort);//UDP connection.
Serial.print("starting UDP at Local Port:");
Serial.println(UDP.localPort());
Serial.println("HTTP server started");
//DNS lookup
if(!WiFi.hostByName(NTPServerName,timeServerIP)){
Serial.println("DNS lookup fail.");
}
Serial.print("Time server IP:");
Serial.println(timeServerIP);
```

圖 17 網頁伺服器程式

2. 紅外線的感測:根據實驗(一)計算出的方程式，我們可以透過下列程式將計算出的距離回傳到 NodeMCU 上:

```
176 sensorValue = 0.0;
177 for(int i=0;i<20;i++){
178     sensorValue += analogRead(sensorIR);
179 }
180 sensorValue/=20;
181 sensorValue *=0.0049;
182 cm = 13*pow(sensorValue, -0.909);
183
184 Serial.print("cm: ");
185 Serial.println(cm);
```

圖 18 紅外線程式

3. 伺服馬達的轉動: 接收到 NodeNCU 的指令後水龍頭會開啟預設的模式，並在手離開後關閉水龍頭。

實驗(二):

我們先透過實際實驗測試水龍頭旋轉至各角度的出水量，以便評估不同模式應該對應的旋轉角度。實驗數據如下:

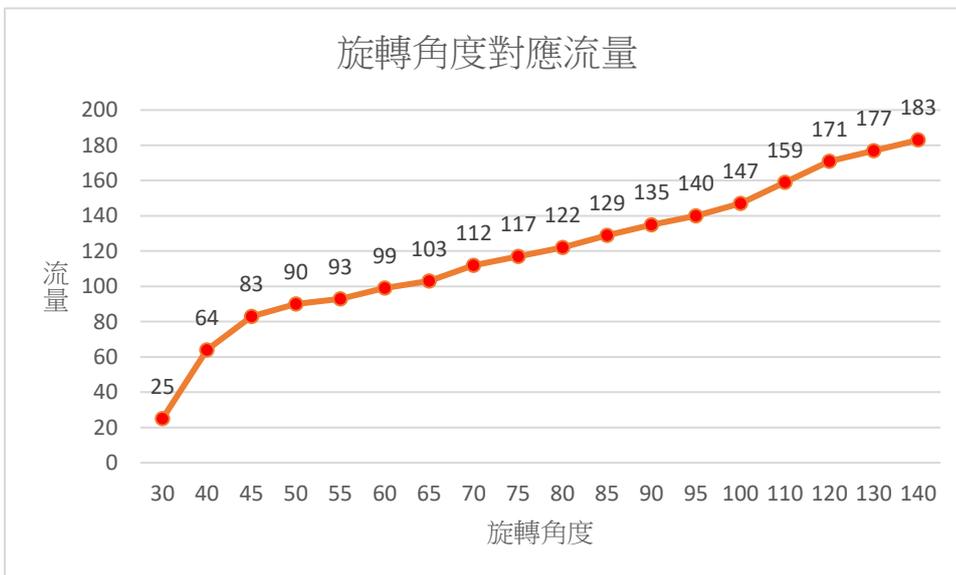


圖 19 旋轉角度對應流量折線圖

```
221 //open gate
222 void TurnOn(){
223     Serial.print("trunOn--");
224     if(!myservo.attached())
225         myservo.attach(servoPin);
226     Serial.print("Servo attach.angle=");
227     Serial.println(pos);
228     isTurnOn=1;
229     switch(usingMode){
230     case 1:
231         feq = 45;
232         break;
233     case 2:
234         feq = 60;
235         break;
236     case 3:
237         feq = 30;
238         break;
239     }
240     if(ServoTime==0)
241         ServoTime=millis();
242     if(pos!=180)
243         myservo.write(pos);
```

圖 19 控制伺服馬達程式

透過上面關係圖，我們在各樓層安裝實驗水龍頭，並透過程式控制不同旋轉角度對應出水量的實驗數據，程式邏輯如右圖:我們發現水龍頭裝在不同樓層的出水量會因為水壓不同而造成出水量不同。實驗數據如下圖:

樓層\旋轉角度	模式 1(60 度圖)	模式 2(90 度)	模式 3(140 度)	模式 4(180 度)
1 樓	110.9ml	128.5ml	222.2ml	268.8ml
2 樓	95.4ml	128.5ml	200.8ml	222.2ml
3 樓	84.4ml	110.3ml	150.4ml	180.6ml

表 3 各樓層對應角度水量表

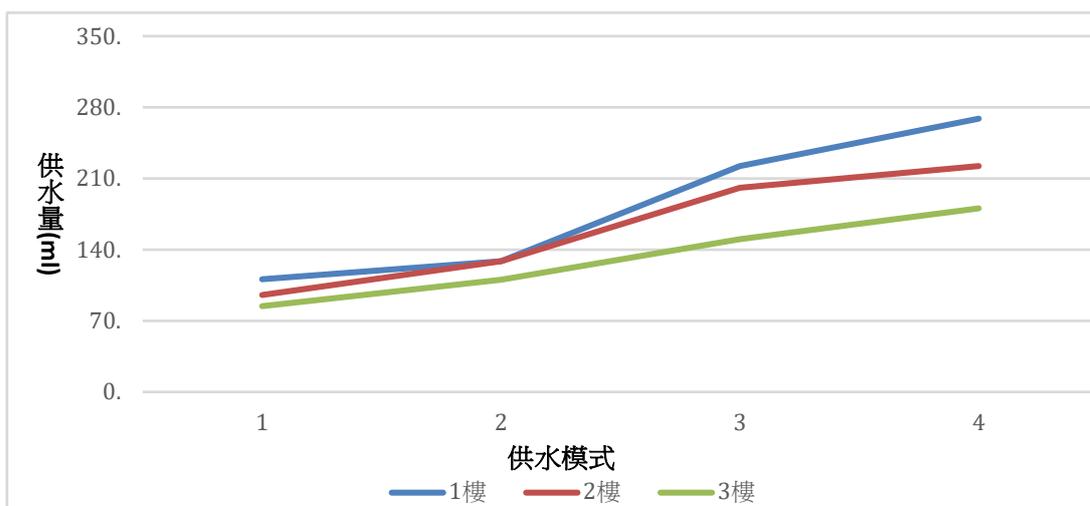


圖 20 各樓層角度水量折線圖

因此單靠旋轉角度去控制水流量並無法客觀且精確的提供不同的用水情境以及詳實的記錄用水量。因此我們必須加裝另外一種裝置，以提高用水的精準度。因此我們加入了霍爾水流計，水流計是透過安裝在磁性渦輪葉片(四葉片)配合霍爾感測器去偵測水流量，當水流經葉片讓它旋轉時，霍爾感測器就會感測到葉片上的磁力產生高電位訊號。因此當渦輪轉動一圈我們會偵測到四次高電位訊號。我們利用霍爾感測器感測到的頻率來計算在同秒速下不同頻率的水量，進而轉換算出不同的頻率代表的每秒流速為何。希望可以透過這些數據運用到伺服馬達上，讓馬達能提供更準確的給予水量。

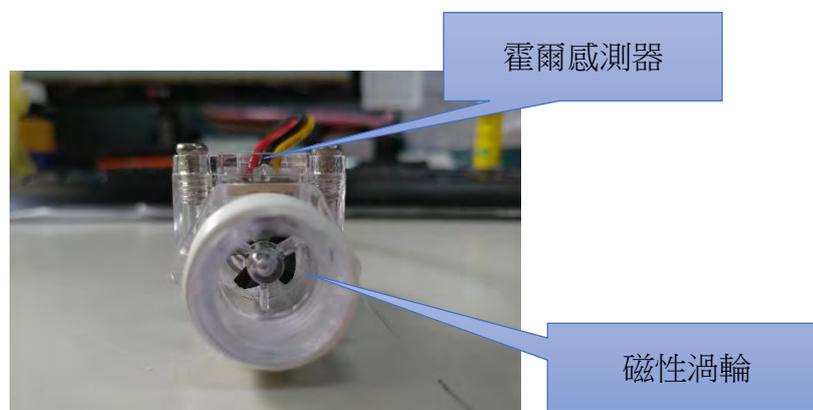
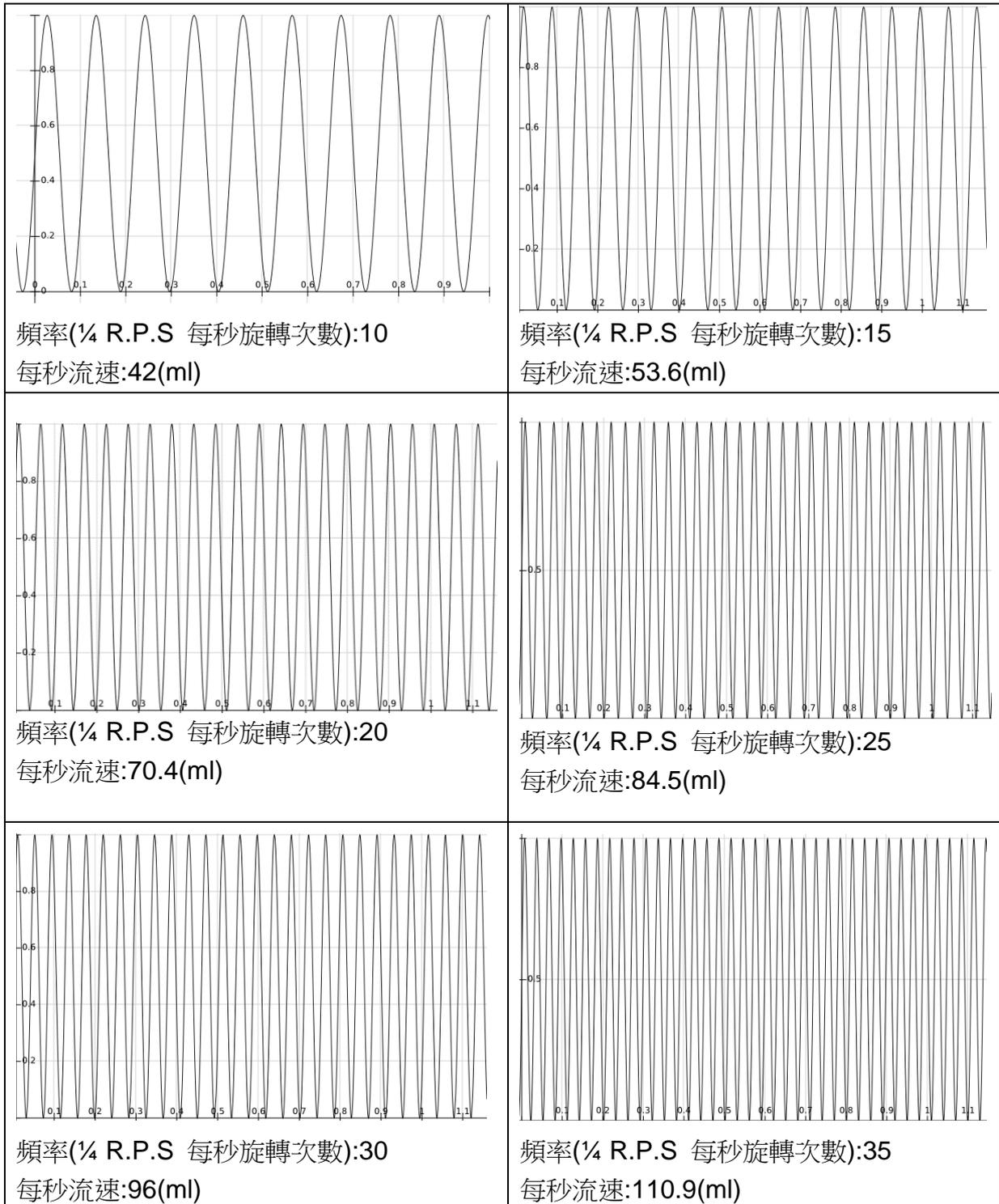
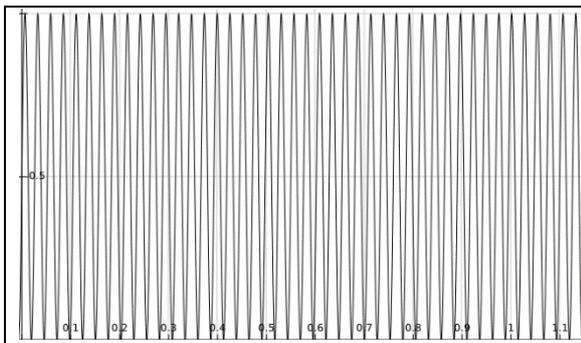


圖 21 霍爾水流計

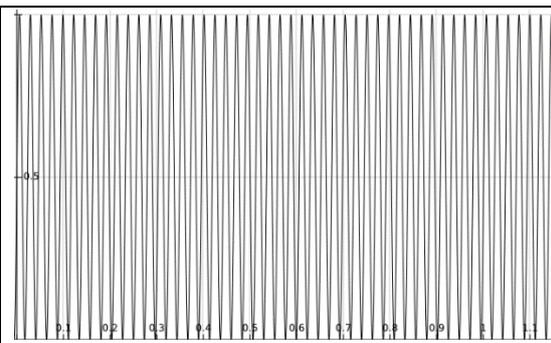
實驗(三):

流量計的原理是透過霍爾感測器搭配磁力葉片的轉動計算它所感應到得電流脈衝，只要每經過一次霍爾感測器就會發送電流頻率。 $\frac{1}{4}$ R.P.S 是每秒旋轉圈數，因為磁力葉片為 4 片，所以利用頻率除以 4 之後就能得到每秒的圈數因此我們可以透過固定頻率來測得對應得水流量，以下各頻率對應水量的數據，可以看出越密集的頻率代表水流量越大。

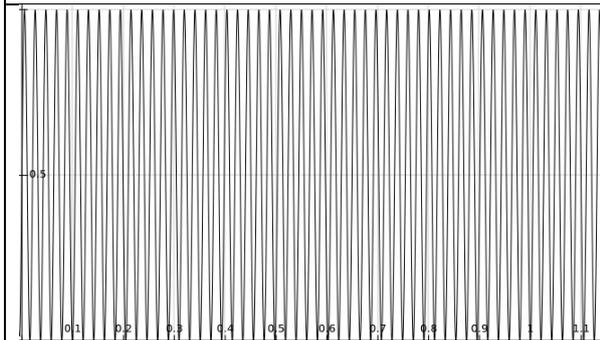




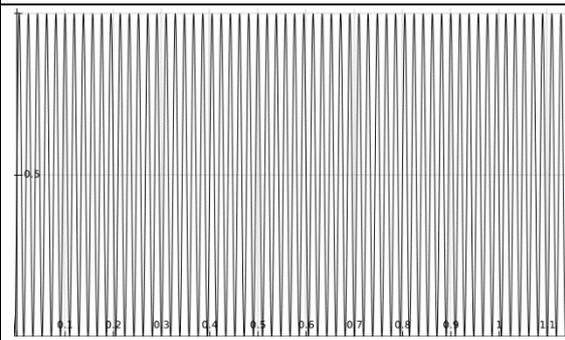
頻率(¼ R.P.S 每秒旋轉次數):40
 每秒流速:128.5(ml)



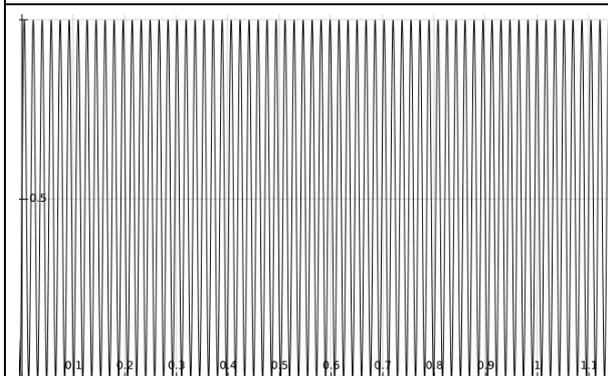
頻率(¼ R.P.S 每秒旋轉次數):45
 每秒流速:138.1(ML)



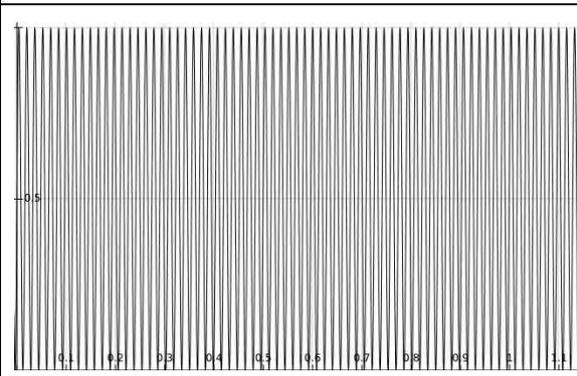
頻率(¼ R.P.S 每秒旋轉次數):50
 每秒流速:158.7(ML)



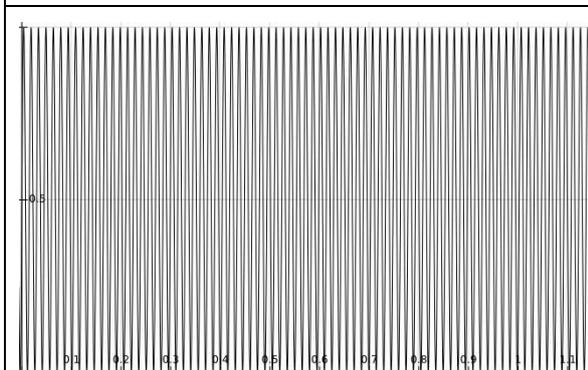
頻率(¼ R.P.S 每秒旋轉次數):55
 每秒流速:170.1(ML)



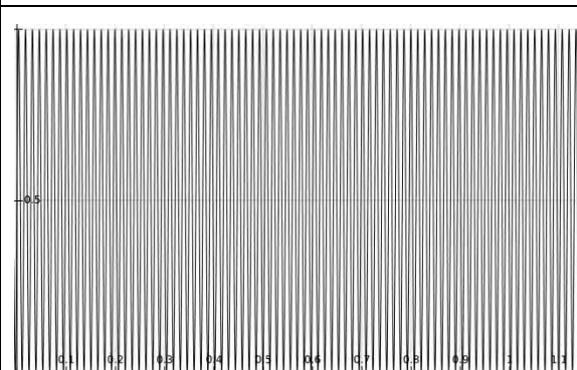
頻率(¼ R.P.S 每秒旋轉次數):60
 每秒流速:188.7(ML)



頻率(¼ R.P.S 每秒旋轉次數):65
 每秒流速:200.8(ML)



頻率(¼ R.P.S 每秒旋轉次數):70
 每秒流速:222.2(ML)



頻率(¼ R.P.S 每秒旋轉次數):75
 每秒流速:243.9(ML)

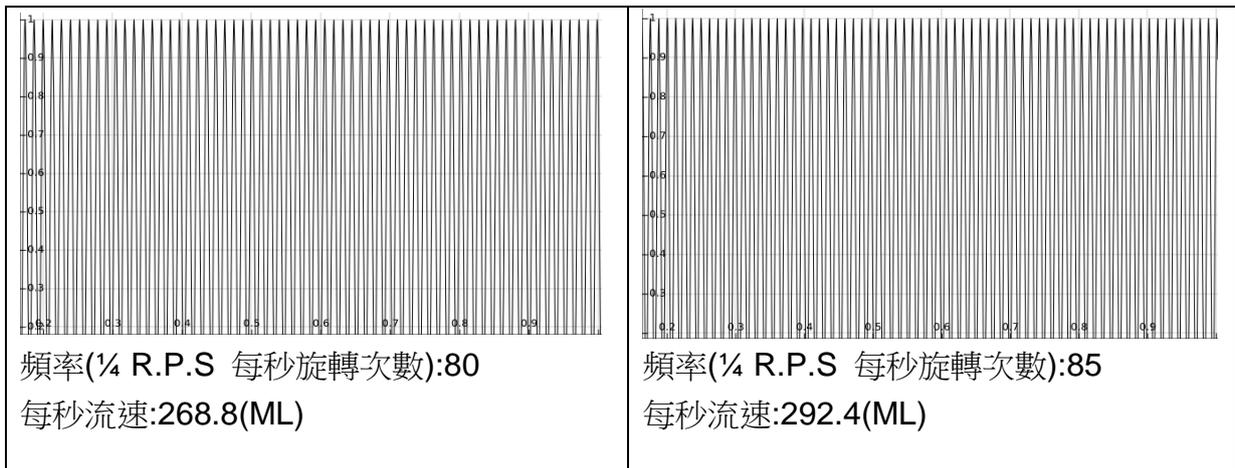


圖 22 頻率波型圖

根據上面的數據，我們可以整理出數據如下表：

頻率(¼ R.P.S)	10	15	20	25	30	35	40	45
流速(ml/s)	42	53.6	70.4	84.5	96	110.9	128.5	138.1
頻率(¼ R.P.S)	50	55	60	65	70	75	80	85
流速(ml/s)	158.7	170.1	188.7	200.8	222.2	243.9	268.8	292.4

表 4 各頻率對應流速表

根據上表我們可以繪出頻率與流量的關係圖

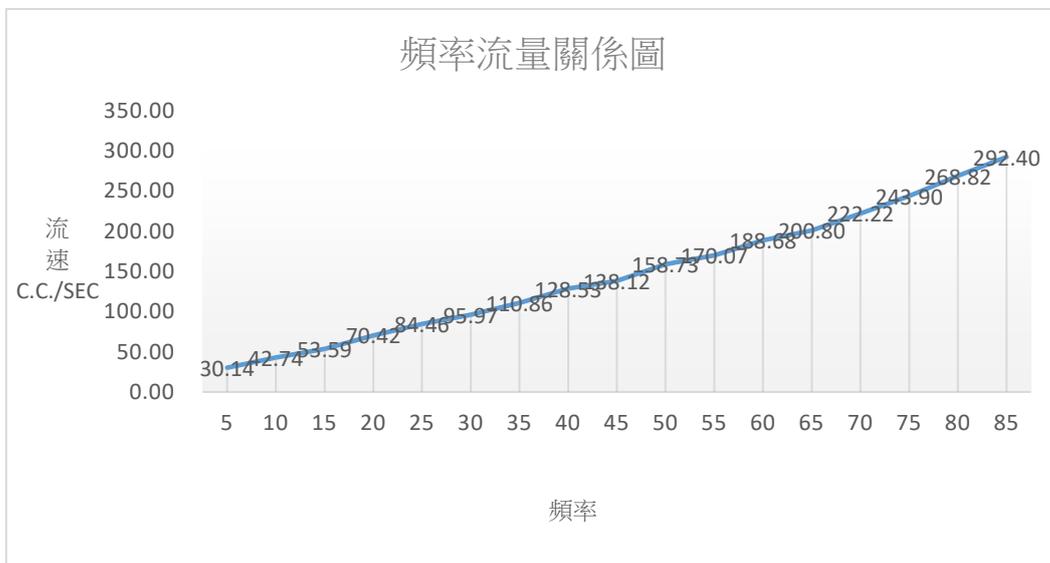


圖 23 流量關係圖表

然而上圖並非一個直線方程式，我們希望可以找出一條直線方程式可以跟上述實驗數據點的相關性最高，我們又稱之為迴歸線。我們可以透過最小平方差法求得這條近似方程式。

最小平方差法:

假設線性迴歸線的方程式為

$$\hat{Y} = \alpha + \beta x$$

我們可以得知在這條線上的任何一個點 (x_i, \hat{y}_i) 滿足

$$\hat{y}_i = \alpha + \beta * x_i$$

如下圖所示:

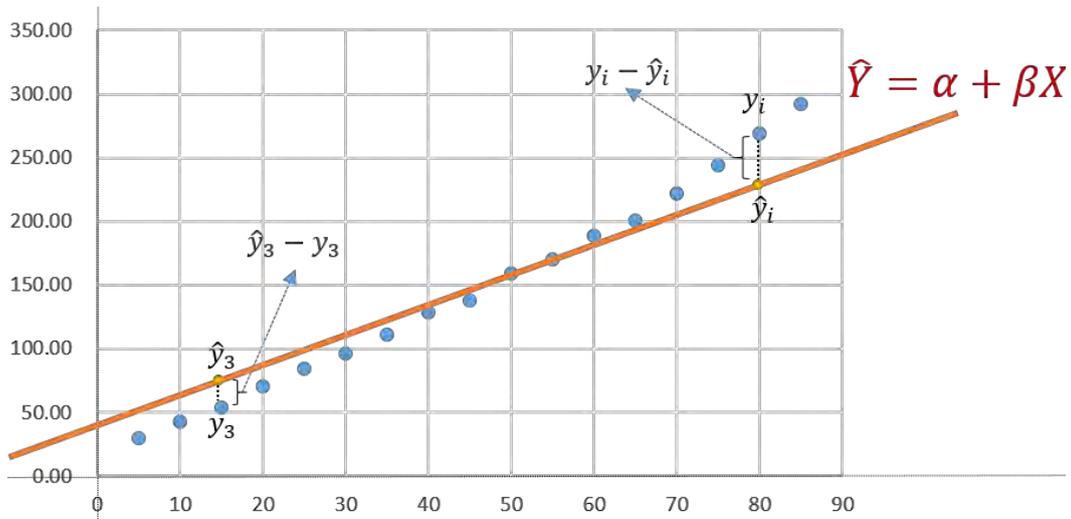


圖 24 最小平方差與數據關係圖

而我們希望我們得到的數據點 (x_i, y_i) 距離直線方程式越近越好，換句話說我們希望 $y_i - \hat{y}_i$ 越小越好，然而如上圖所示， $y_i - \hat{y}_i$ 不一定是正值，因此可以改寫為

$$\sum_{i=1}^N (Y_i - \hat{Y})^2 \dots\dots\dots(1)$$

越小越好，因此我們的目的便是去求得一條迴歸方程式使的上式為最小值。

將 $\hat{y}_i = \alpha + \beta x_i$ 帶入上述方程式我們得到

$$\sum_{i=1}^N (Y_i - \alpha - \beta X_i)^2$$

我們知道一個二次方程式的一階微分等於零的點即是極值點，因此我們針對 α 作微分得到

$$\frac{\partial \sum_{i=1}^N (Y_i - \alpha - \beta X_i)^2}{\partial \alpha} = 0$$

$$\sum_{i=1}^N 2(Y_i - \alpha - \beta X_i) * (-1) = 0$$

$$\sum_{i=1}^N (Y_i - \alpha - \beta X_i) = 0$$

$$\begin{aligned} \sum_{i=1}^N Y_i - \sum_{i=1}^N \alpha - \beta \sum_{i=1}^N X_i &= 0 \\ \sum_{i=1}^N Y_i - N * \alpha - \beta \sum_{i=1}^N X_i &= 0 \\ \frac{(\sum_{i=1}^N Y_i - N * \alpha - \beta \sum_{i=1}^N X_i)}{N} &= 0 \\ \left(\frac{\sum_{i=1}^N Y_i}{N} - \alpha - \frac{\beta \sum_{i=1}^N X_i}{N} \right) &= 0 \end{aligned}$$

其中 $\frac{\beta \sum_{i=1}^N X_i}{N} = \beta \bar{X}$, $\frac{\sum_{i=1}^N Y_i}{N} = \bar{Y}$, 因此我們得到:

$$\bar{Y} - \alpha - \beta \bar{X} = 0$$

$$\bar{Y} = \alpha + \beta \bar{X} \dots\dots\dots(2)$$

由上述方程式可以得知點 (\bar{X}, \bar{Y}) 必定在 $\hat{Y}_i = \alpha + \beta X_i$ 上

我們可以得到點 (\bar{X}, \bar{Y}) 和 $\hat{Y}_i = \alpha + \beta X_i$ 上的任一點可以得出斜率

$$\beta = \frac{(\hat{Y}_i - \bar{Y})}{(X_i - \bar{X})}$$

其中 $(\hat{Y}_i - \bar{Y}) = (Y_i - \bar{Y}) - (Y_i - \hat{Y}_i)$ 因此可以得到

$$\beta = \frac{(Y_i - \bar{Y}) - (Y_i - \hat{Y}_i)}{(X_i - \bar{X})}$$

$$(X_i - \bar{X}) \beta = (Y_i - \bar{Y}) - (Y_i - \hat{Y}_i)$$

$$(Y_i - \hat{Y}_i) = (Y_i - \bar{Y}) - (X_i - \bar{X}) \beta \dots\dots\dots(3)$$

由式(1)及式(3)我們可以得到

$$\sum_{i=1}^N (Y_i - \hat{Y}_i)^2 = \sum_{i=1}^N ((Y_i - \bar{Y}) - \beta(X_i - \bar{X}))^2 \dots\dots\dots(4)$$

將式(4)對 β 進行一次微分:

$$\sum_{i=1}^N 2 \left((Y_i - \bar{Y}) - \beta(X_i - \bar{X}) \right) (X_i - \bar{X}) = 0$$

$$\sum_{i=1}^N \left((Y_i - \bar{Y})(X_i - \bar{X}) - \beta(X_i - \bar{X})(X_i - \bar{X}) \right) = 0$$

$$\sum_{i=1}^N (Y_i - \bar{Y})(X_i - \bar{X}) - \beta \sum_{i=1}^N (X_i - \bar{X})(X_i - \bar{X}) = 0$$

$$\sum_{i=1}^N (Y_i - \bar{Y})(X_i - \bar{X}) = \beta \sum_{i=1}^N (X_i - \bar{X})(X_i - \bar{X})$$

$$\beta = \frac{\sum_{i=1}^N (Y_i - \bar{Y})(X_i - \bar{X})}{\sum_{i=1}^N (X_i - \bar{X})^2} \dots\dots\dots (5)$$

由式(2)我們可以得到:

$$\alpha = \bar{y} - \beta\bar{x} \dots\dots\dots (6)$$

透過式(5)，我們將實測所得的數據代入，可以得到下表:

x_i (頻率)	y_i (水流量)	$x_i - \bar{x}$	$y_i - \bar{y}$	$(Y_i - \bar{Y})(X_i - \bar{X})$	$(x_i - \bar{x})^2$	$(y_i - \bar{y})^2$
5	30.13864	-40.0	-116.9	4677.9	1600	13676.61262
10	42.73504	-35.0	-104.4	3652.3	1225	10889.05721
15	53.59057	-30.0	-93.5	2804.9	900	8741.337495
20	70.42254	-25.0	-76.7	1916.6	625	5877.23903
25	84.45946	-20.0	-62.6	1252.5	400	3922.044529
30	95.96929	-15.0	-51.1	766.7	225	2612.886185
35	110.8647	-10.0	-36.2	362.2	100	1311.956727
40	128.5347	-5.0	-18.6	92.8	25	344.139002
45	138.1215	0.0	-9.0	0.0	0	80.35582858
50	158.7302	5.0	11.6	58.2	25	135.5936934
55	170.068	10.0	23.0	229.8	100	528.1879053
60	188.6792	15.0	41.6	623.9	225	1730.023989
65	200.8032	20.0	53.7	1074.4	400	2885.572452
70	222.2222	25.0	75.1	1878.4	625	5645.498741
75	243.9024	30.0	96.8	2904.5	900	9373.483236
80	268.8172	35.0	121.7	4260.6	1225	14818.56201
85	292.3977	40.0	145.3	5812.5	1600	21115.56938
$\bar{x} = 45.00 \quad \bar{y} = 147.09$				$\Sigma = 32368.1$	$\Sigma = 10200.0$	$\Sigma = 103688.1$

表 5 數據表

由式(4)， $\beta = \frac{\sum_{i=1}^N (Y_i - \bar{Y})(X_i - \bar{X})}{\sum_{i=1}^N (X_i - \bar{X})^2} = \frac{32368.1}{10200.0} = 3.1733$

由式(3)， $\bar{y} = \alpha + \beta\bar{x}$ ， $\alpha = \bar{y} - \beta\bar{x}$ ， $\alpha = 4.2915$

因此我們得到:

最小平方差方程式: $y = 3.1733x + 4.2915$

$$\text{相關係數 } R^2 = \left(\frac{\sum_{i=1}^n (x-\bar{x})(y-\bar{y})}{\sqrt{\sum_{i=1}^n (x-\bar{x})^2} \sqrt{\sum_{i=1}^n (y-\bar{y})^2}} \right)^2 = \frac{32368.1}{100.99 \times 322.00} = 0.995$$

其中相關係數 0.995 代表這個函數與實測數據之間的相關性非常高。

程式修正:

求出線性回歸方程式後，接著我們完成程式控制的部分，主要的程式架構如下圖:

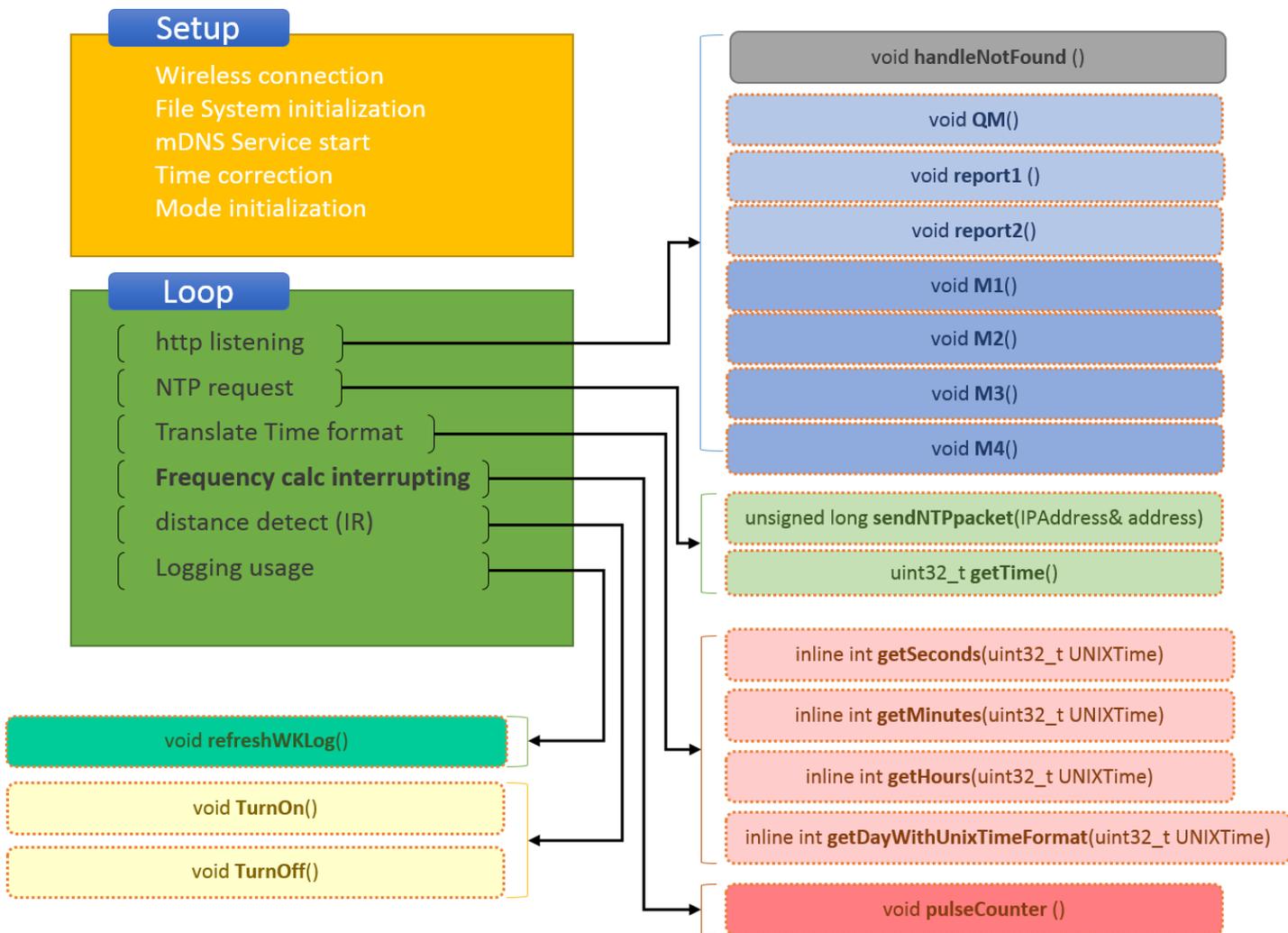


圖 25 程式架構圖

圖 26 是程式部份的函數以及對應的功能，我們需要利用到頻率來計算我們的出水量，計算頻率的工作採用到分時多工，因此我們需要利用到中斷機制，當我們的霍爾感測器葉片轉動時會先將目前工作停止，先記錄累積次數再返回繼續執行目前工作，圖 26 中的 Frequency Calc Interrupting 利用 pulseCounter() 函數每收到一次得電流脈衝就會累加每過一秒就會歸

零來計算出每秒的頻率

其中 **http listening** 是監聽網頁的服務，檢查需不需要更改模式透過 **javascripts** 顯示相關網頁並且利用 **NTP request** 取到 **NTP** 的網路時間格式 **Translate Time format** 來計算目前的時間。

程式的流程圖如下圖所示：

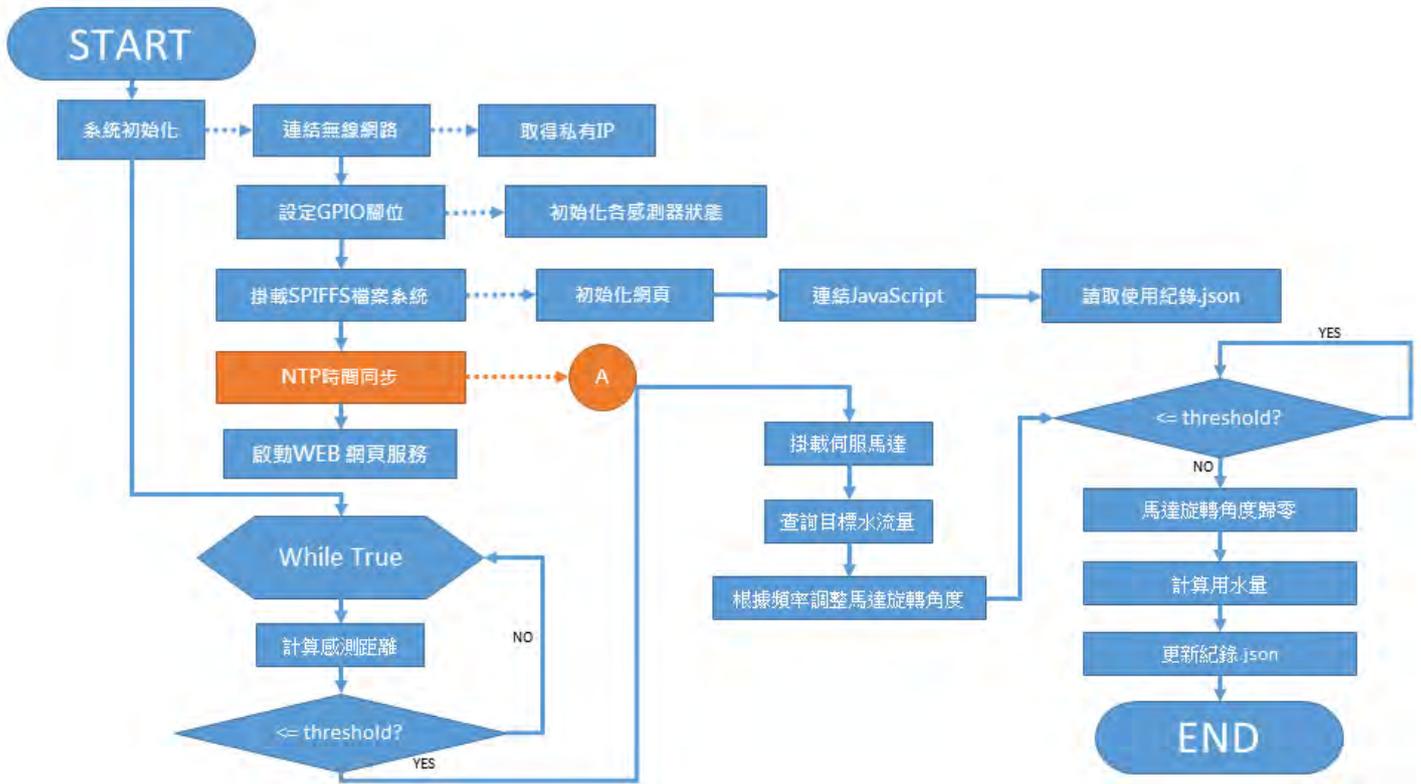


圖 26 系統初始化架構圖

水龍頭接上線路後，我們會先進行初始化設定，初始化設定要先連接到我們的無線網路，因為考慮到資訊安全方面的問題，我們使用私有的網路來設定。連接到網路後我們要接上 **GPIO** 的腳位，讓每個元件都初始化。接著掛載系統檔案，並連接到外部 **CDN** 下載需要用到的 **JavaScripts**，最後讀取 **SPIFFS** 檔案系統內的用水紀錄檔。由於我們需要紀錄特定時間點內的用水量，因此我們需要使用 **NTP** 通訊協定去取得目前的時間。最後再啟動我們的網頁伺服器。

當系統初始化完之後我們就會運用迴圈開始偵測是否有物體接近，當達到我們的門檻值時我們會將伺服馬達掛載起來，並查詢目前設定模式，根據頻率去設定旋轉角度，最後再用迴圈判斷是否要關閉水龍頭，

右圖為透過解析 NTP 封包後，擷取出時間戳章的位元資料，並且將它包裝成陣列，呼叫遠端 NTP 伺服器，以取得目前的正確時間。

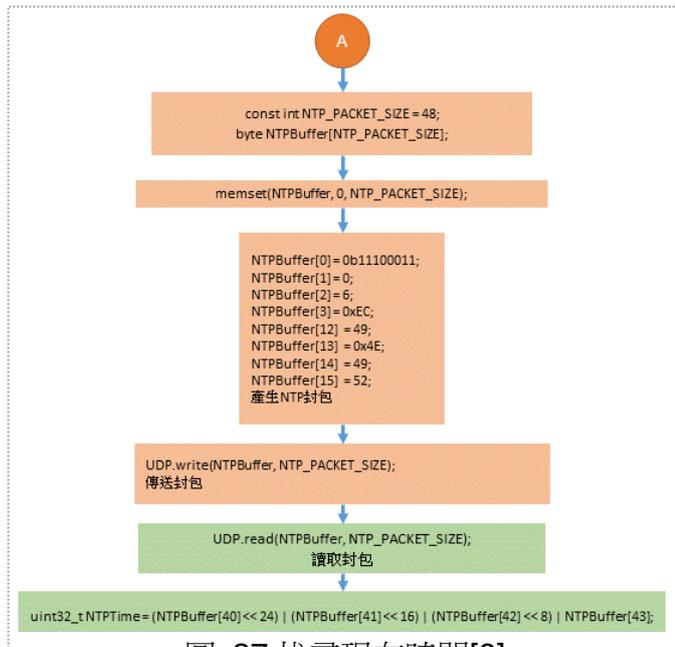


圖 27 找尋現在時間[9]

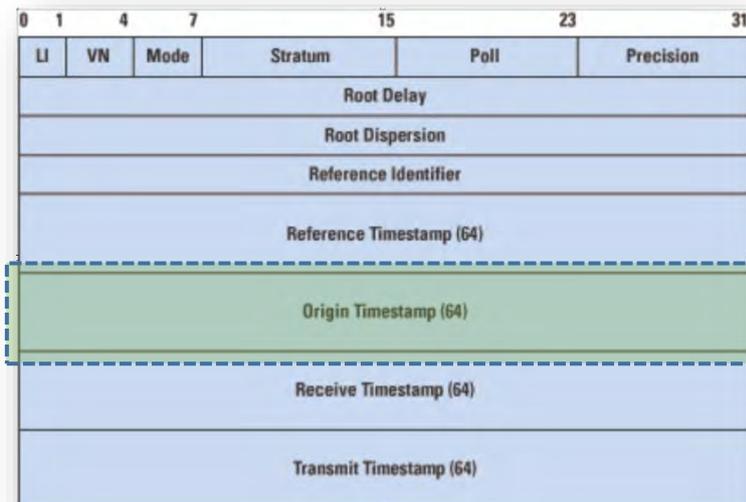


圖 28 NTP 封包

初始化完畢後，程式邊開始檢查是否需要開啟水龍頭，流程圖如下：

首先，先設定每隔一分鐘呼叫一次 NTP request，當時間到達時便傳送 NTP 封包，當得到 NTP 封包後，我們再擷取第 40~43Byte 內的內容，即為目前的時間。

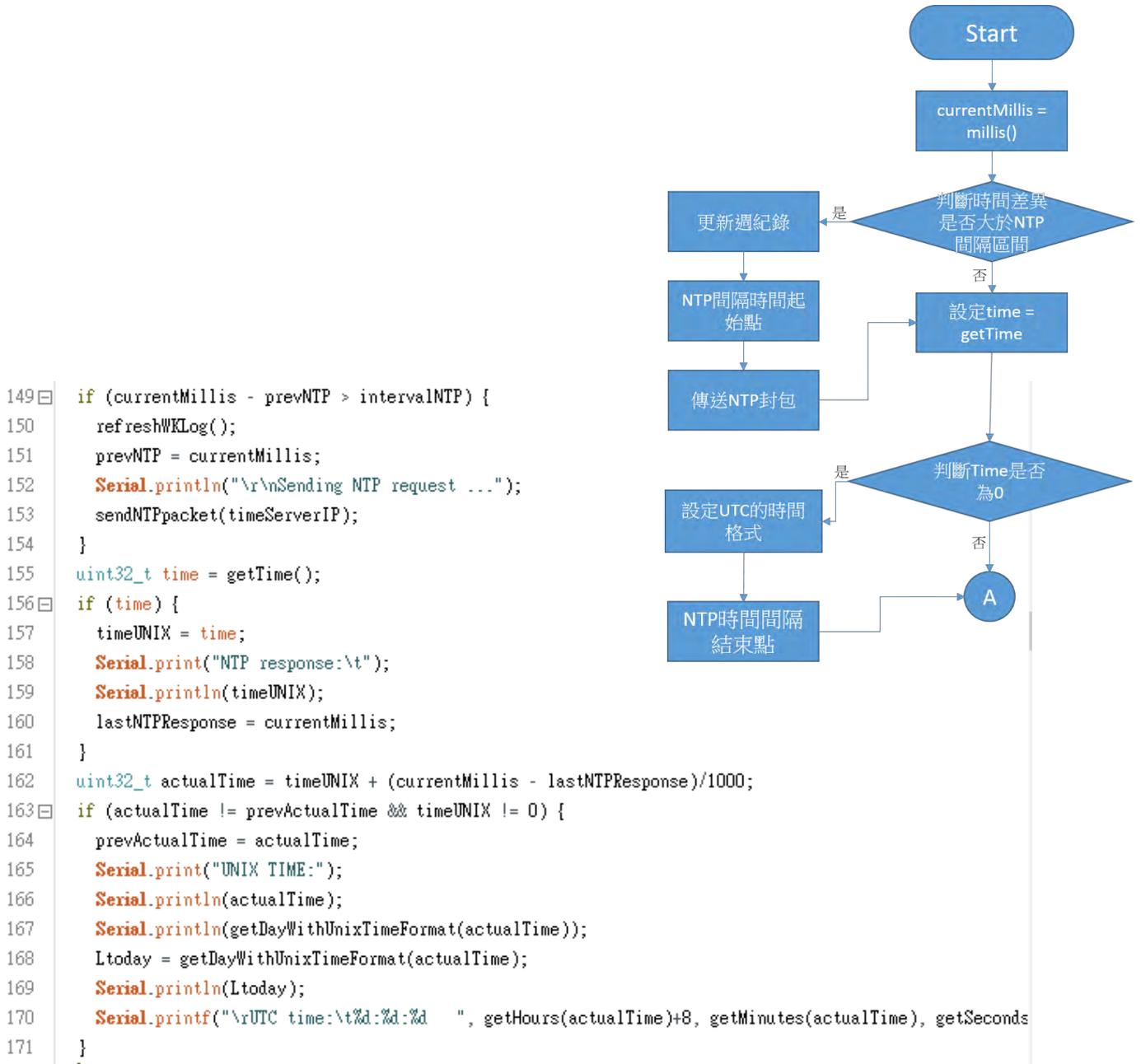


圖 29 取得 NTP 程式碼與流程圖

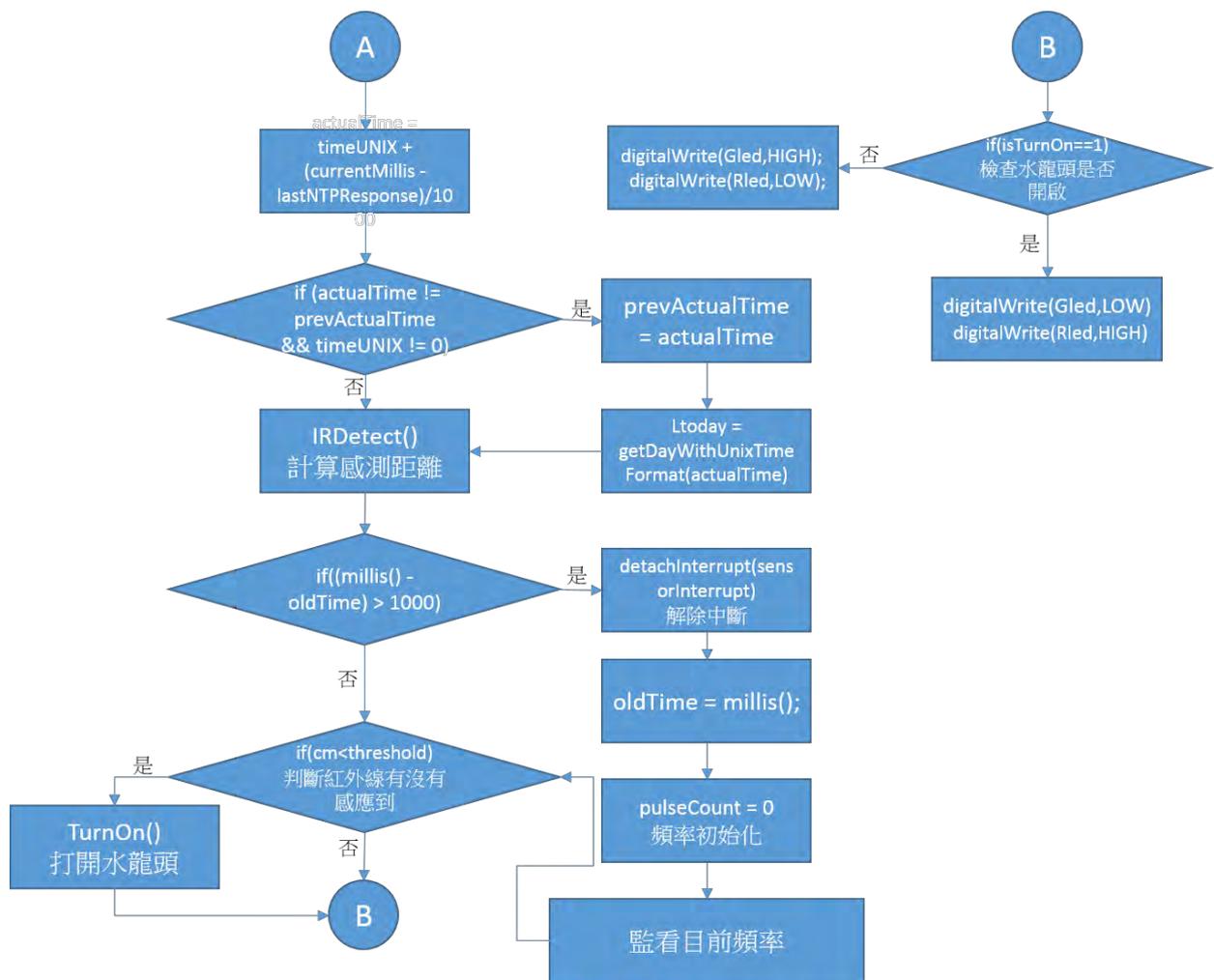


圖 30 判斷水龍頭開啟流程圖

上圖為感測是否有物體靠近的程式流程圖，程式每隔一秒偵測一次目前頻率，並同時偵測是否有物體靠近感測區內，判斷是否應該控制伺服馬達轉動水龍頭供水，並且改變 LED 的狀態，感測區為偵測物體時為藍燈，當偵測到物體後轉為紅燈並給水。

研究結果

系統架構圖:



圖 31 系統架構圖

每一個作品都是採分散式獨立架構，也就是說每個水龍頭都是獨立的伺服器，能夠個別存取水流量，其中網頁介面是儲存於 NodeMCU 的 SPIFFS 上，因此不會因為其中一座故障而影響到其他水龍頭的運作。讓我們能隨時查詢各個水龍頭的流量、統計水量以及設定不同的流水模式，這樣就能在因不同場所使用的流量不同而制定方案。

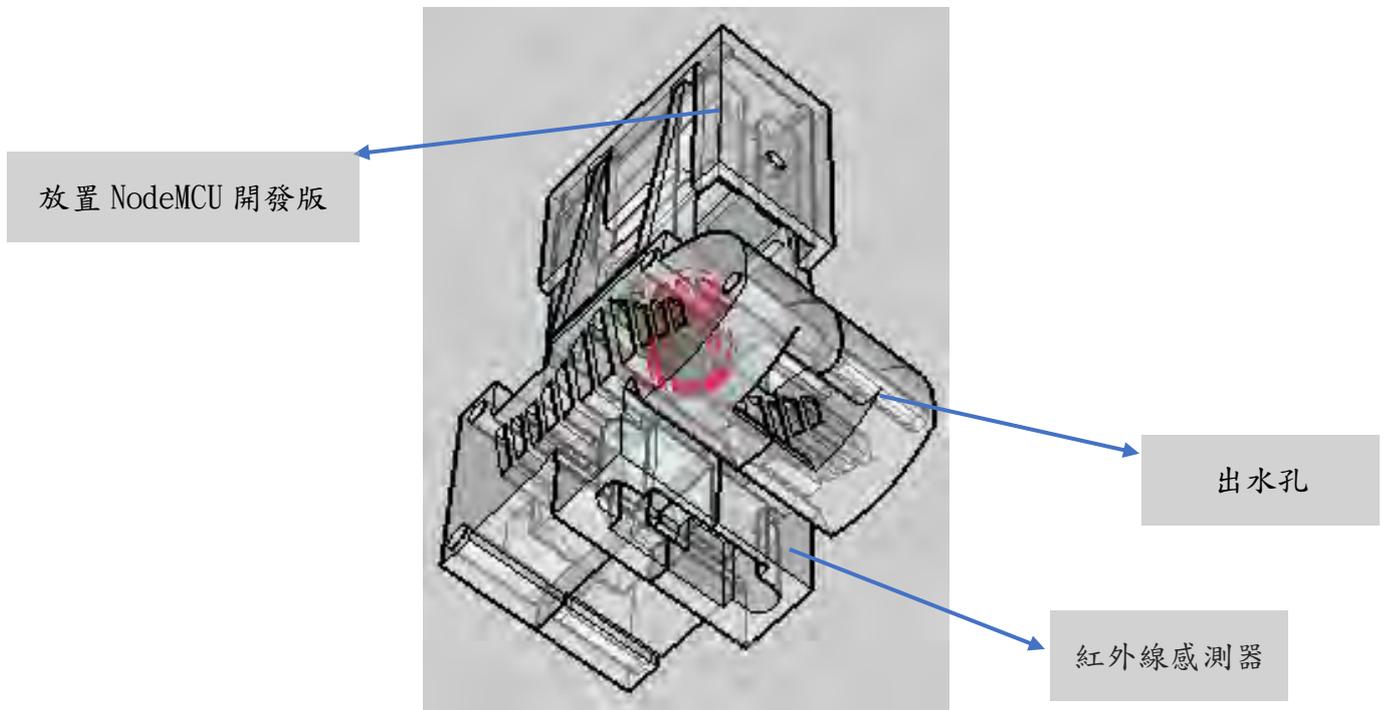


圖 32 3D 建模透視圖

另外因為使用了 3D 建模去容納現存之水龍頭，因此完全無需更改管路架構，只需要裝上並接上電源即可使用。



圖 33 產品完成及實際使用圖

因 esp8266 可用來作成一個小型的網頁伺服器，因此我們在上面寫了一個控制介面，畫面如下：



圖 34 模式按鈕

上面六個模式是根據錯誤的經驗以及修正錯誤後，結合上面使用的流量計做出更精準的控制。

分別為：

一般模式: 頻率 45，流速為 138ml/sec，一般非限水情況下皆使用此種模式。

清潔模式: 頻率 60 流速為 188ml/sec，掃地時間使用此種模式。

節水模式: 頻率 30，流速為 96ml/sec，旱季或者特定情況下使用該種模式。

禁用模式: 伺服馬達不動作，周末假日限定部分水龍頭使用此種模式。避免在沒人監督使用下造成過度浪費。

輸入水龍頭所屬的 IP 位址，即可看到目前水龍頭所在的位置、即時的使用量。

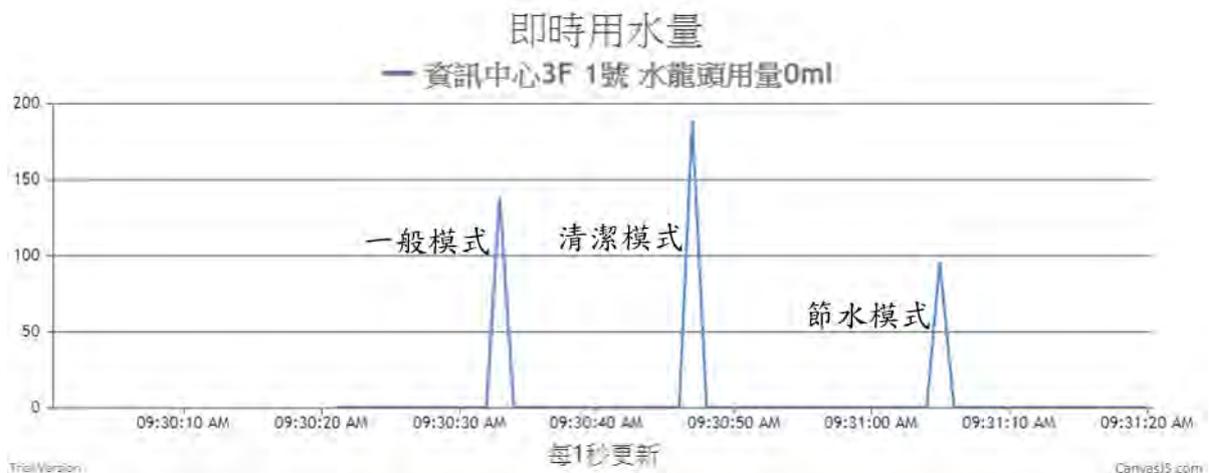


圖 35 即時用水量統計

除了即時用量外，也可以查詢該週用量。



圖 36 每週用水量統計

網頁採用響應式網頁技術，因此無論是在筆電上、平板上或手機上，皆可自動調整並同步顯示。



圖 37 不同裝置皆可同步顯示使用量

結果與討論

1. 以自動感應的方式打開，既方便又能透過流量控制節省水資源，非常符合現代的環保理念。
2. 與市面常見的感應式水龍頭的差異就是使用了物聯網技術配合網頁監控功能，讓我們更方便於網路上進行管理。
3. 只要透過手持裝置或電腦即能夠了解用水量，進而微幅調整用水策略，有益於各公家單位，學校在水量控管上的功能
4. 配合現代水龍頭的外型，利用課堂上學習的 3D 建模軟體(Sketch up)建置可以包圍在水龍頭的模組，讓安裝感應式水龍頭的過程簡易化，使日後使用者能快速簡易安裝。

5. 可以指定不同模式來控制不同的出水量，旱季可設定為節水模式，提供最低可用之水量，清潔模式可加大水流量增加用水效率，其餘時間使用一般模式，提供固定穩定之水量。
6. 最重要的是採用分散式架構，讓水龍頭能獨立運行，不會因為其中一座故障而影響到其他的運作。且不需後端資料庫，可直接將資料儲存在水龍頭的晶片上。

結論

本作品結合了現代環保意識以及物聯網上的使用達到節省水資源的效果，利用3D列印可以讓本作品可以更加方便的被使用例如：在不同外型的水龍頭我們可以先測量水龍頭大小然後再修改建模圖就可以不必更改任何管線，以及本作品利用到分散式架構每個作品的一個獨立的作品不會因一個作品當機而導致其他作品受到影響。運用了大量的數據去分析，找出最精確的水量，以達到環保節能的概念。雖然中間經過了許多波折，實驗也失敗了很多次，但這實驗讓我們了解到自己不足的地方，利用這些經驗加上不斷的實驗，最終得出了我們的實驗結果。

未來展望

1. 加裝 NFC 感測器，可利用感應式裝置控制水龍頭的使用狀態
2. 增加互動模式，根據感測距離給予不同的水量
3. 透過流量計，可進行高壓流體的偵測，判斷是否有洩漏
4. 增加防水方面的措施，讓水龍頭不易進水
5. 結合大數據的分析，依不同地區所擁有的水量進行設定
6. 將水龍頭的設定簡單化，讓使用者更方便於使用

參考文獻

[1]瑪姬·布拉克、珍奈特·金(2012)。水資源地圖。出版社：聯經出版公司

[2]柯安妮(2005)。台灣的水權問題與水權交易市場制度之研究

[3]維基百科全書官網(2018)。伺服馬達，取自

<https://zh.wikipedia.org/wiki/%E4%BC%BA%E6%9C%8D%E9%A6%AC%E9%81%94>

[4]紅外線距離電壓圖。紅外線距離感測器，取自 http://images.100y.com.tw/pdf_file/46-

Sharp-GP2Y0A41SK0F.pdf

[5]曹永忠、許智誠、蔡英德(2016)。Arduino 程式教學(基本語法篇)。

出版社：渥瑪數位有限公司

[6]James Floyd Kelly(2014)。3D 列印無限可能：從打造自己的 3D 印表機到輸出個性化 3D 物件。出版社：博碩文化股份有限公司

[7] 維基百科全書官網(2018)。NodeMCU。取自 <https://zh.wikipedia.org/wiki/NodeMCU>

[8] Cooper Maa, Fritzing – PCB 印刷電路板的設計工具。取自

<http://coopermaa2nd.blogspot.com/2011/01/fritzing-pcb.html>

[9] 小狐狸事務所，使用 Arduino IDE 開發 ESP8266 應用 (三)：從 NTP 伺服器取得網路時間。取自 <http://yhhuang1966.blogspot.com/2017/09/arduino-ide-esp8266-ntp.html>

[10] 維基百科全書官網(2018)。最小平方法

<https://zh.wikipedia.org/wiki/%E6%9C%80%E5%B0%8F%E4%BA%8C%E4%B9%98%E6%B3%95>

[11]維基百科全書官網(2018)。微積分學

<https://zh.wikipedia.org/wiki/%E5%BE%AE%E7%A7%AF%E5%88%86%E5%AD%A6>

附錄 1 紅外線感測器測試數據

	1cm	2cm	3cm	4cm	5cm	6cm	7cm	8cm	9cm	10cm
數據 1	552	761	765	769	670	580	501	454	393	362
數據 2	550	762	769	765	665	579	499	454	380	363
數據 3	550	758	764	765	662	579	499	459	392	367
數據 4	550	761	769	765	669	581	500	453	391	363
數據 5	552	761	744	766	663	579	500	458	392	362
數據 6	550	760	769	765	664	579	500	454	393	363
數據 7	550	761	768	766	664	579	500	457	393	363
數據 8	552	762	768	765	664	579	501	430	396	363
數據 9	550	760	769	766	664	578	500	456	394	363
數據 10	550	761	761	764	664	598	500	453	392	362
數據 11	552	753	769	768	663	583	501	460	393	363
數據 12	552	757	746	768	664	579	506	453	392	363
數據 13	552	758	768	768	663	579	500	456	393	363
數據 14	552	759	769	768	663	577	505	453	391	362
數據 15	550	761	765	769	665	579	499	454	392	362
數據 16	552	752	766	766	663	581	500	457	384	363
數據 17	550	760	766	766	665	579	502	454	391	362
數據 18	552	758	765	768	663	579	500	447	392	364
數據 19	552	760	766	768	665	580	500	453	403	367
數據 20	552	757	768	768	666	579	499	454	396	364
數據 21	550	758	766	768	663	583	499	454	392	364
數據 22	549	757	766	766	662	580	476	453	392	363
數據 23	554	740	766	764	664	579	498	454	394	365
數據 24	550	756	769	768	665	579	499	463	392	365
數據 25	552	757	769	741	662	579	501	453	392	362
數據 26	552	757	769	766	664	579	504	454	391	363
數據 27	550	757	769	768	661	581	499	460	393	362
數據 28	579	754	769	765	663	556	500	460	390	363
數據 29	552	759	770	768	663	579	501	454	392	368
數據 30	553	759	769	764	663	580	502	454	377	362
數據 31	553	760	770	766	662	580	500	456	392	364
數據 32	552	757	769	745	663	572	499	460	392	362
數據 33	550	760	769	768	667	580	499	454	392	362
數據 34	550	756	770	766	634	579	500	457	391	362

數據 35	554	760	770	766	665	580	499	460	392	363
數據 36	528	758	763	766	666	580	499	454	396	363
數據 37	554	759	768	756	663	580	509	454	392	363
數據 38	550	757	739	653	663	580	499	453	392	364
數據 39	552	761	769	745	663	579	501	454	393	362
數據 40	549	756	672	764	663	579	506	453	391	363
數據 41	552	761	766	765	663	579	505	471	417	364
數據 42	549	759	769	766	665	582	499	452	392	362
數據 43	550	761	761	765	669	581	500	454	393	363
數據 44	550	642	769	765	663	580	500	457	394	363
數據 45	550	760	748	765	664	582	499	454	392	363
數據 46	541	760	768	766	664	579	501	456	392	362
數據 47	552	761	771	765	664	579	499	471	391	363
數據 48	550	748	764	765	662	579	506	454	393	363
數據 49	553	760	769	766	663	579	500	457	392	362
數據 50	552	757	754	768	664	582	500	460	393	363
數據 51	552	756	769	769	664	580	500	460	392	363
數據 52	552	761	766	768	669	579	501	453	392	361
數據 53	552	752	765	769	663	579	499	454	394	339
數據 54	552	760	766	768	668	579	506	453	400	362
數據 55	550	760	766	768	662	579	501	456	391	366
數據 56	552	760	766	768	666	579	509	454	392	363
數據 57	552	760	766	768	663	580	500	458	392	363
數據 58	554	757	766	766	669	579	500	454	393	362
數據 59	552	760	765	769	662	579	499	454	368	361
數據 60	552	738	766	769	664	580	499	456	401	365
數據 61	552	761	766	765	669	579	499	454	401	362
數據 62	552	760	769	770	664	580	501	454	392	362
數據 63	549	759	769	764	664	580	506	454	392	355
數據 64	550	757	768	769	663	579	500	458	392	363
數據 65	553	753	769	747	665	579	498	454	391	362
數據 66	552	760	771	768	663	579	500	454	392	364
數據 67	550	761	769	769	663	603	500	452	392	363
數據 68	549	760	769	766	664	581	506	454	392	362
數據 69	550	756	769	766	658	580	502	456	391	363
數據 70	548	759	770	761	664	580	500	454	400	363
數據 71	550	756	768	766	664	580	505	456	394	363
數據 72	555	759	769	746	664	580	499	454	392	363

數據 73	549	760	768	766	681	579	499	454	396	363
數據 74	553	759	754	768	663	577	500	456	391	363
數據 75	570	757	759	764	664	568	500	457	392	363
數據 76	553	762	768	768	664	582	507	454	394	363
數據 77	548	761	768	757	663	579	498	454	391	362
數據 78	550	758	765	764	664	579	506	454	392	361
數據 79	556	766	769	765	664	579	495	456	392	363
數據 80	553	760	741	765	663	580	506	453	392	363
數據 81	550	757	769	765	646	578	506	454	392	363
數據 82	552	761	768	765	664	579	501	459	398	362
數據 83	550	758	769	766	662	582	505	454	391	362
數據 84	552	760	769	766	664	579	501	425	392	367
數據 85	552	759	758	764	605	579	500	453	392	363
數據 86	550	752	770	764	669	578	500	456	392	363
數據 87	550	742	761	625	666	580	499	453	394	362
數據 88	552	760	768	768	663	578	501	456	392	364
數據 89	547	756	766	768	663	579	499	457	391	364
數據 90	552	761	763	769	669	579	499	456	392	364
數據 91	550	748	636	769	663	580	500	454	391	363
數據 92	553	762	766	768	663	579	499	453	393	367
數據 93	528	649	766	770	663	580	507	453	392	363
數據 94	552	757	766	768	662	578	500	456	392	361
數據 95	550	761	766	768	662	578	507	454	392	363
數據 96	552	758	766	766	664	579	500	456	396	366
數據 97	542	760	765	768	663	581	499	454	393	362
數據 98	553	740	766	744	663	579	500	453	391	361
數據 99	552	760	768	766	663	579	507	453	393	363
數據 100	552	761	770	768	664	579	501	454	391	357
平均值	551.07	755.59	763.49	762.53	663.11	579.51	500.83	454.76	392.46	362.68

	11cm	12cm	13cm	14cm	15cm	16cm	17cm	18cm	19cm	20cm
數據 1	333	303	283	272	246	233	223	209	196	189
數據 2	332	304	283	270	246	234	223	210	197	190
數據 3	327	304	282	270	246	234	222	210	196	190
數據 4	327	302	282	271	254	234	225	209	200	190
數據 5	334	305	282	270	246	237	223	210	195	185
數據 6	333	309	283	273	254	234	223	209	197	194
數據 7	334	306	284	271	246	234	222	209	196	192

【評語】 052509

本研究實作水龍頭出水量控制器，完成度高。運用紅外線感測器偵測物理距離進行水龍頭之開關，並利用旋轉角度與流量計進行統計及控制水量，製作不同出水量的水龍頭模式，最後使用電流頻率判斷旋轉角度，並配合程式控制馬達來控制旋轉角度，達到節省水資源之目的。此外，配合物聯網技術，透過統計水龍頭用水量了解水龍頭用水情形，亦可遠端控制水龍頭。

考慮因素詳盡。於測量旋轉角度與出水量此一關係之實驗過程中，不忘考慮因樓層高度而造成的水壓不同，進而改變出水量之問題，並在此之上提出流量計這一實際解法，可提升對於出水量管控之精確度。該作品兼具理論與實用性。

摘要

1. 紅外線感測器偵測距離進行開關水閥
2. 分成不同模式，根據不同情境使用
3. 流量計統計及控制水量
4. 無須更改管線的包覆式架構
5. 將網頁寫入晶片，無須額外架站就能觀看、統計流量
6. 採每個作品都是獨立伺服器的分散式架構

動機

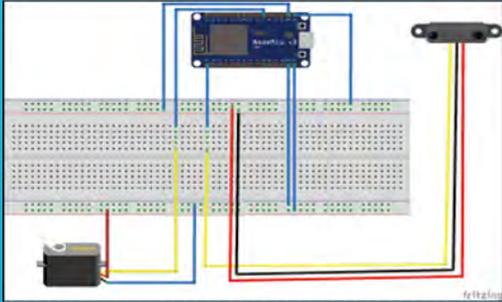
1. 環保節能
2. 避免接觸式感染
3. 課程所學實際運用在生活上

目的

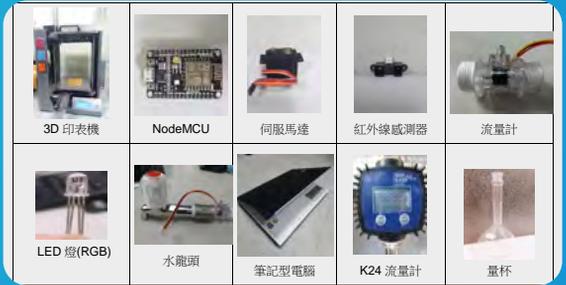
1. 透過伺服馬達旋轉控制出水
2. 流量計配合物聯網技術了解用水情形
3. 製作出方便統一管理的系統

研究過程與方法

電路設計



設備與器材



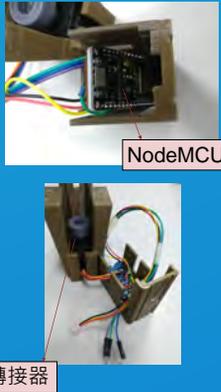
3D列印



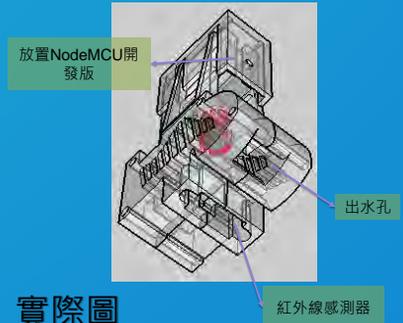
包覆式架構



零件安裝部位



3D透視圖

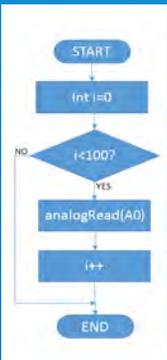


實際圖

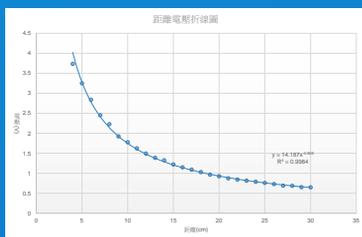


實驗一

紅外線感測器校正



距離	1cm	2cm	3cm	4cm	5cm	6cm	7cm	8cm	9cm	10cm
平均回傳值	NA	NA	NA	762.53	663.11	579.51	500.83	454.76	392.46	362.68
距離	11cm	12cm	13cm	14cm	15cm	16cm	17cm	18cm	19cm	20cm
平均回傳值	331.36	304.48	282.88	270.35	248.94	234.68	222.68	210.5	196.84	189.05
距離	21cm	22cm	23cm	24cm	25cm	26cm	27cm	28cm	29cm	30cm
平均回傳值	178.43	172.02	166.4	160.6	154.85	148.85	140.95	139.98	133.37	132.4



```

181 sensorValue = 0.0;
182 for(int i=0;i<20;i++){
183   sensorValue += analogRead(sensorIR);
184 }
185 sensorValue/=20;
186 sensorValue *=0.0049;
187 cm = 14.187*pow(sensorValue, -0.909);
188
189 Serial.print("cm: ");
190 Serial.println(cm);
  
```

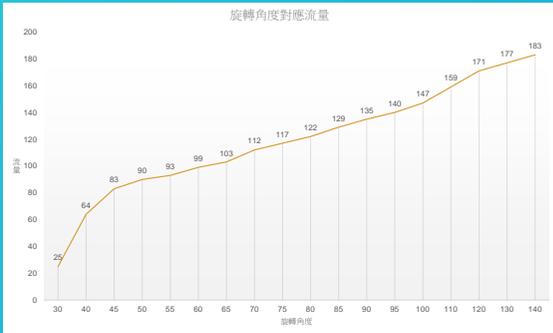
距離與回傳值的關係方程式:

$$y = 14.187x^{-0.909}$$

其中x為距離，y為偵測回傳值。我們可以利用此方程式來計算感應區是否有手靠近

實驗二

伺服馬達旋轉角度對應出水量



各角度對應水量

樓層/旋轉角度	模式1(60度圖)	模式2(90度)	模式3(140度)	模式4(180度)
1樓	110.9ml	128.5ml	222.2ml	268.8ml
2樓	95.4ml	128.5ml	200.8ml	222.2ml
3樓	84.4ml	110.3ml	150.4ml	180.6ml

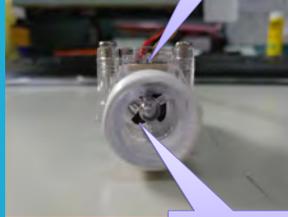
各樓層角度對應水量

實驗三

霍爾流量計修正水量統計模型

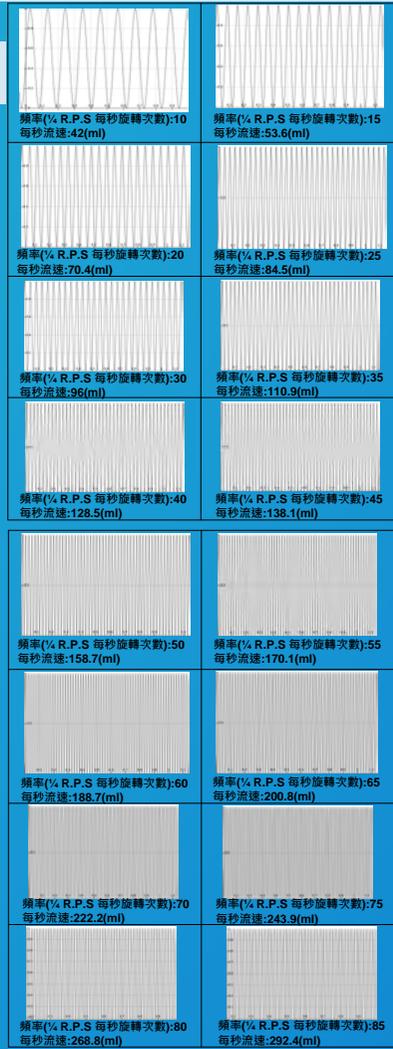
單靠旋轉角度去控制水流量並無法客觀且精確的提供不同的用水情境，所以我們使用流量計透過霍爾感測器來計算水量

我們利用流量計所發送的電流脈衝除以時間得到的頻率將它製作成圖表



霍爾感測器

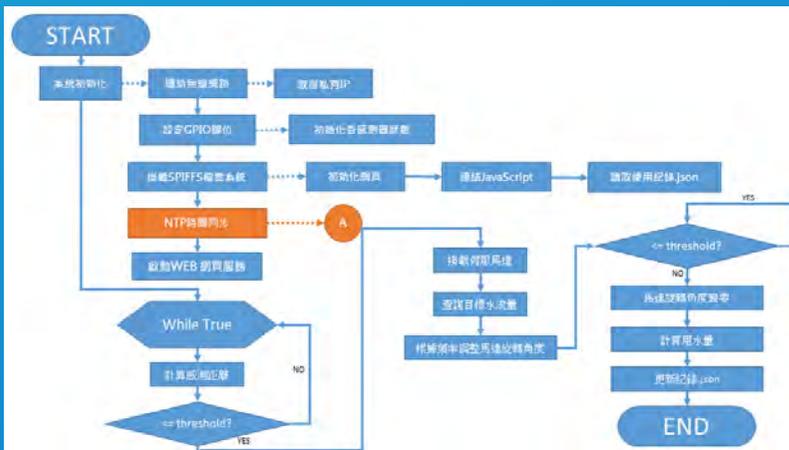
磁性渦輪



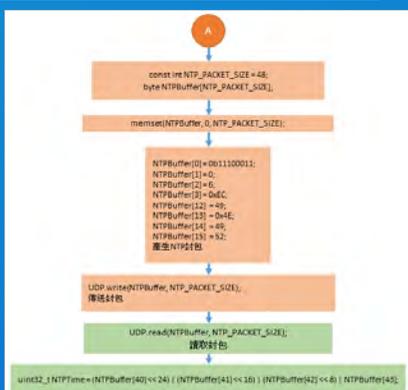
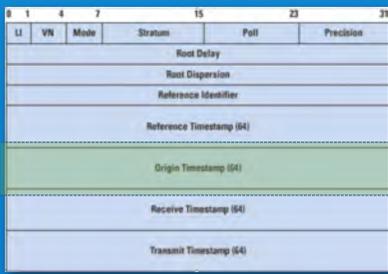
系統架構



程式流程圖

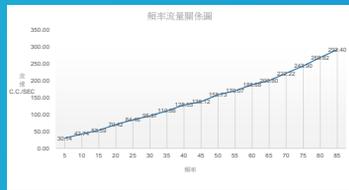


NTP協定取得時間



頻率(¼ R.P.S)	10	15	20	25	30	35	40	45
流速(ml/s)	42	53.6	70.4	84.5	96	110.9	128.5	138.1
頻率(¼ R.P.S)	50	55	60	65	70	75	80	85
流速(ml/s)	158.7	170.1	188.7	200.8	222.2	243.9	268.8	292.4

各頻率對應流速表



流量關係圖

流量關係圖並非一個直線方程式，我們希望可以找出一條直線方程式可以跟上上述實驗數據點的相关性最高

最小平方差法

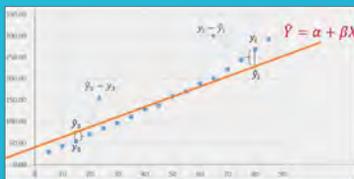
假設線性迴歸線的方程式為

$$\hat{Y} = \alpha + \beta x$$

我們可以得知在這條線上的任何一個點 (x_i, \hat{y}_i) 滿足

$$\hat{y}_i = \alpha + \beta * x_i$$

如下圖所示:



而我們希望我們得到的數據點 (x_i, y_i) 距離直線方程式越近越好，換句話說我們希望 $y_i - \hat{y}_i$ 越小越好，然而如上圖所示 $y_i - \hat{y}_i$ 不一定是正值，因此可以改寫為

$$\sum_{i=1}^N (Y_i - \hat{Y})^2 \dots\dots\dots (1)$$

$$\sum_{i=1}^N (Y_i - \alpha - \beta X_i)^2$$

由下述方程式可以得知點 (\bar{X}, \bar{Y}) 必定在 $\hat{Y}_i = \alpha + \beta X_i$ 上

$$\bar{Y} = \alpha + \beta \bar{X} \dots\dots\dots (2)$$

$$\beta = \frac{(\bar{Y} - \bar{Y})}{(\bar{X} - \bar{X})}$$

$$(Y_i - \hat{Y}_i) = (Y_i - \bar{Y}) - (X_i - \bar{X})\beta \dots\dots\dots (3)$$

$$\sum_{i=1}^N (Y_i - \hat{Y}_i)^2 = \sum_{i=1}^N ((Y_i - \bar{Y}) - \beta(X_i - \bar{X}))^2 \dots\dots\dots (4)$$

$$\beta = \frac{\sum_{i=1}^N (Y_i - \bar{Y})(X_i - \bar{X})}{\sum_{i=1}^N (X_i - \bar{X})^2} \dots\dots\dots (5)$$

$$\alpha = \bar{y} - \beta \bar{x} \dots\dots\dots (6)$$

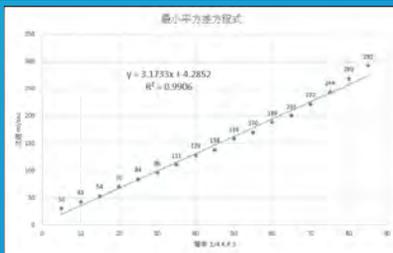
透過式(5)，我們將實測所得的數據代入，可以得到下表:

x_i (頻率)	y_i (水流量)	$x_i - \bar{x}$	$y_i - \bar{y}$	$(Y_i - \bar{Y})(X_i - \bar{X})$	$(x_i - \bar{x})^2$	$(y_i - \bar{y})^2$
5	30.13864	-40.0	-116.9	4677.9	1600	13676.61262
10	42.73504	-35.0	-104.4	3652.3	1225	10889.05721
15	53.59057	-30.0	-93.5	2804.9	900	8741.337495
20	70.42254	-25.0	-76.7	1916.6	625	5877.23903
25	84.45946	-20.0	-62.6	1252.5	400	3922.044529
30	95.96929	-15.0	-51.1	766.7	225	2612.886185
35	110.8647	-10.0	-36.2	362.2	100	1311.956727
40	128.5347	-5.0	-18.6	92.8	25	344.139002
45	138.1215	0.0	-9.0	0.0	0	80.35582858
50	158.7302	5.0	11.6	58.2	25	135.5936934
55	170.068	10.0	23.0	229.8	100	528.1879053
60	188.6792	15.0	41.6	623.9	225	1730.023989
65	200.8032	20.0	53.7	1074.4	400	2885.572452
70	222.2222	25.0	75.1	1878.4	625	5645.498741
75	243.9024	30.0	96.8	2904.5	900	9373.483236
80	268.8172	35.0	121.7	4260.6	1225	14818.56201
85	292.3977	40.0	145.3	5812.5	1600	21115.56938
$\bar{x} = 45.00$	$\bar{y} = 147.09$			$\sum = 32368.1$	$\sum = 10200.0$	$\sum = 103688.1$

最小平方差方程式: $y = 3.1733x + 4.2915$

$$\text{相關係數 } R^2 = \left(\frac{\sum_{i=1}^N (x_i - \bar{x})(y_i - \bar{y})}{\sqrt{\sum_{i=1}^N (x_i - \bar{x})^2} \sqrt{\sum_{i=1}^N (y_i - \bar{y})^2}} \right)^2 = \frac{32368.1}{100.99 * 322.00} = 0.995$$

其中相關係數0.995代表這個函數與實測數據之間的相关性非常高



線性迴歸方程式

實驗四 K24流量計校正

Time(sec)	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
流量計水量(ML)	199	408	631	845	1044	1258	1438	1590	1818	2017	2217	2406	2605	2819	3076	3246
實際水量	207	452	679	915	1158	1359	1575	1735	2005	2230	2444	2669	2900	3130	3444	3616
誤差值	8	44	48	70	114	101	137	145	187	213	227	263	295	311	368	370
間距	36	4	22	44	-13	36	8	42	26	14	36	32	16	57	2	

透過比較實際水量以及流量計水量的誤差值並利用線性迴歸，我們找出了K24水流量計的修正方程式。

$$y = 225.017x' - 6.2 \dots\dots\dots (1)$$

$$y' = 201.98x' + 12.179 \dots\dots\dots (2)$$

先利用最小平方差法求出上面的 y 與 y'

$$x' = \frac{y' - 12.179}{201.98}$$

$$y = 225.17 \left(\frac{y' - 12.179}{201.98} \right) - 6.2$$

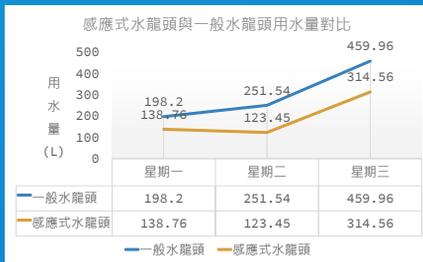
再將 x' 代入(1)

$$y = 1.114(y' - 12.179) - 6.2 = 1.114y' - 7.367$$

實驗五 效能分析

	星期一	星期二	星期三
一般水龍頭	198.2	251.54	459.96
感應式水龍頭	138.76	123.45	314.56

我們分別實測感應式水龍頭(節水模式)及一般水龍頭加裝K24流量計的用水量，關係如上表所示，可以發現節水模式可以大幅度減少用水量。



結論

1. 以自動感應的方式打開，既方便又能透過流量控制節省水資源，非常符合現代的環保理念。
2. 與市面常見的感應式水龍頭的差異就是使用了物聯網技術配合網頁監控功能，讓我們更方便於網路上進行管理。
3. 只要透過手持裝置或電腦即能夠了解用水量，進而微幅調整用水策略，有益於各公家單位，學校在水量控管上的功能。
4. 配合現代水龍頭的外型，利用課堂上學習的3D建模軟體(Sketch up)建置可以包覆在水龍頭的模組，讓安裝感應式水龍頭的過程簡易化，使日後使用者能快速簡易安裝。
5. 最重要的是採用分散式架構，讓水龍頭能獨立運行，不會因為其中一座故障而影響到其他的運作。且不需後端資料庫，可直接將資料儲存在水龍頭的晶片上。

未來展望

1. 透過流量計，可進行高壓流體的偵測，判斷是否有洩漏
2. 增加NFC模式，辨認使用者身份
3. 結合大數據的分析，依不同地區所擁有的水量進行設定
4. 將水龍頭的設定簡單化，讓使用者更方便於使用