

說在前面

其實在我們的日常生活中，IO控制是相當普遍的，只是我們往往忽略了它們的存在。本文就要帶您一同探討這些已經深入我們日常生活的 IO 控制，並學習 IO 控制的一些技巧與方法。

預期可以學到的相關知識

一、IO 控制系統的架構規劃

從控制器與感測器的結合、到與 PC 端連線、並記錄溫度變化的歷史資料，您可以習得整體的系統架構規劃。

二、熱水瓶的溫度控制的原理

探討熱水瓶的各種加熱模式，及其控制的相關原理。

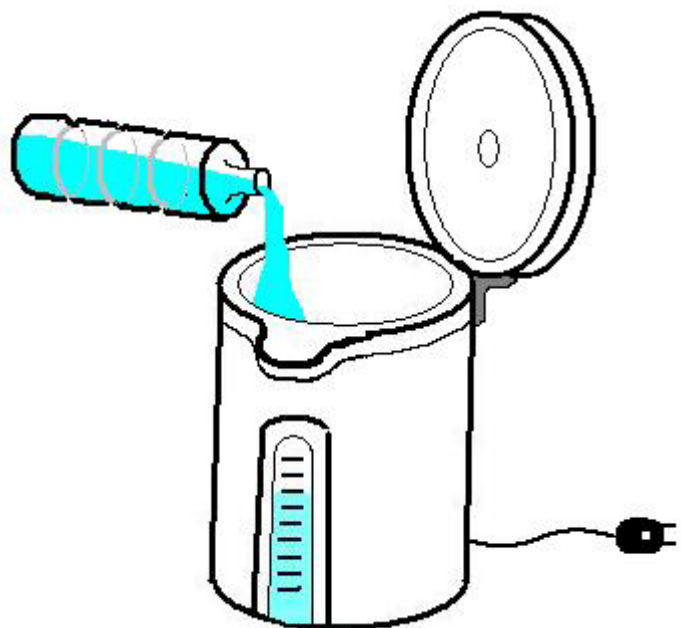
三、MSComm 元件的用法

MSComm是在Visual Basic當中，通訊界面的必用元件，您可以在本文章中學到使用它的方法與技巧。

四、Excle 圖表的製作與應用

溫度變化的歷史資料，我們可以透過Excle強大的試算能力，來進行分析的工作。

接下來，就請各位細細品嚐，由我們為您精心調配的 IO 控制知性饗宴吧！



旗威科技有限公司

保溫的真相 -- 熱水瓶篇

保溫知多少

爲了配合多元化的消費者，家用的電熱水瓶功能，可是一加再加，從只要會煮沸和保溫，到現在功能是越來越多，三段保溫、五段定時，要泡咖啡、綠茶、牛奶、泡麵，通通都可以交給熱水瓶一手包辦。這些新增的功能不但貼心，也更能打動消費者的心。

對於熱水瓶保溫的功能我們是肯定的，讓我們對於電熱水IO的設定不禁想一探究竟，失去密封瓶身的熱水瓶，到底該怎麼去保持瓶內水的溫度?我們知道，熱水瓶在良好的狀態下，密封的保溫空間再加上保溫的運作，兩個因素加起來，讓瓶內溫度不會產生太大的改變，但是如果我們干擾外在的環境，也就是不讓熱水瓶瓶身密閉，單憑保溫動作的情形下，熱水瓶又該如何讓瓶內溫度持續保持呢?它又會產生那些動作呢?一起來實驗看看吧!

量測工具的準備

1、電熱水瓶：

此次實驗對象爲一般家用電熱水瓶，容量3L可加熱到100度，加熱後可設定98度的保溫模式；或是保持90度的省電保溫模式，另外也可以選擇60度的保溫設定另外面版上也有顯示電熱水瓶的溫度，不過此一溫度顯示以5度爲一個指標，所以我們將會以自行準備的溫度計來得知熱水瓶度的變化。當中爲了避免溫度下降過快，所以我們僅將熱水瓶口微開約四分之一。



[圖一]圖中顯示量測用的熱水瓶，而右上角倒掛的是溫度計，目前顯示熱水瓶的溫度值爲88.2度

2、溫度計： 量測用溫度計可量測範圍由+120 度到零下 20 度。



[圖二]此為量測用的溫度計，上方是不鏽鋼的感測棒，通電後中間螢幕將會顯示感測棒所感應到的溫度值

3、電腦： 我們用電腦做資料的接收，爲了取得電熱水瓶每一秒的溫度變化，因此我們捨棄人工記錄熱水瓶面版顯示的資料，另外準備用Microsoft Visual Basic 6.0，也就是VB寫一個小程式，來接收溫度計接收到的資料，再讓電腦把接受到的溫度回傳到電腦並儲存下來供我們分析。

4、UR485： 溫度計和電腦的溝通，我們用 UR485 來做橋樑。

5、DC24V 電源供應器： 提供溫度計的電源。



[圖三]左方即 DC24V 電源供應器，右邊是 UR485 轉接盒

量測溫度資料

在準備好上述工具後，讓我們一步一步來觀測囉！

1、 儲存資料的格式

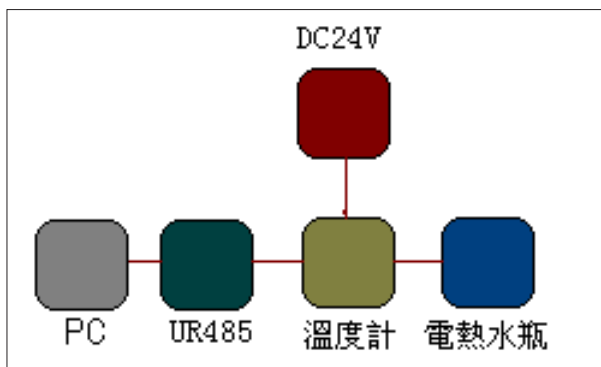
在程式裡我們會透過UR485送出詢問溫度計目前接收到的值為何，再傳送回來到電腦做儲存，當然也可以用人工記錄的方式，只是會比較耗費時間。目前是以文字檔(.txt)的狀態來儲存資料。另外，為了增加溫度資料的取樣性以確保分析數據的參考性，在程式中我們每隔一秒就會去抓取溫度計的值，同時也將抓取資料的時間記錄下來。程式的部份是參考「RS485 講座的mscomm的應用」下去改寫，對此部份有興趣的可以自行前往旗威網站參考相關內容。

2、 預計取樣範圍

通訊部份完成後(也就是程式部份)，接下來就可以開始進行量測的動作，當我們將熱水瓶加滿水後，設定保溫98度觀看電熱水瓶在沸騰時，其溫度狀況為何?還有保溫狀況又是如何?我們也以同樣的方式來監測看看，如果設定保溫90度模式下，電熱水瓶溫度的改變狀況會是怎樣?保溫的狀況會改變嗎?

3、 實際量測的方式

以下是我們實際量測時的動作。



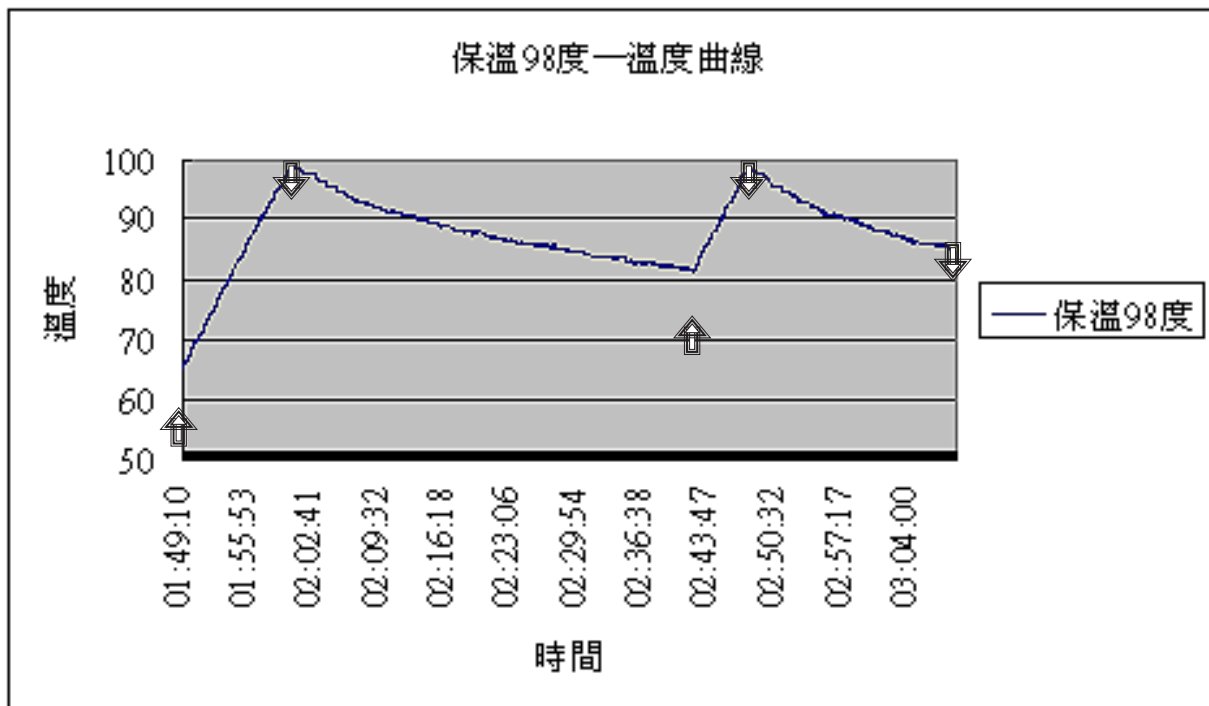
[圖四]整個量測工具的串接順序

個人電腦與 UR485 我們會用一條 USB2.0 傳輸線來連接，而 UR485 的另一端就接到溫度計上面，然後把溫度計與 DC24V 接上，再把溫度計的感應棒放入電熱水瓶裡面，最後再將 DC24V 電源供應器的電源插上，即可開啓電腦(PC)內的程式來執行溫度的量測

量測結果

1、加水後設定 98 度保溫模式

在加入 3L 的水後，我們讓熱水瓶進行煮沸的動作，同時間我們也設定煮沸後保溫 98 度，此時水溫為 65.3 度。在經過 11 分後水溫達到最高 98.7 度，沸騰過後熱水瓶進入我們預設的保溫模式，持續保溫了 41 分後溫度降到 81.5 度，此時熱水瓶由保溫模式切入煮沸模式，再經過 5 分後第二次煮沸，此時溫度為 98.4 度，之後繼續切入保溫模式，22 分後溫度值降低到 84.9 度後我們停止量測，收到資料筆數 4030 筆。



[表一]熱水瓶保溫 98 度溫度曲線圖

圖中五個箭頭所指的溫度，由左到右分別為，起始溫度：65.3 度，第一次沸騰：98.7 度，保溫轉再沸騰：81.5 度，第二次沸騰：98.4 度，結束溫度：84.9 度。

第一條上昇曲線：花費 11 分 36 秒，上昇 33.4 度，平均每分鐘上昇 2.879 度

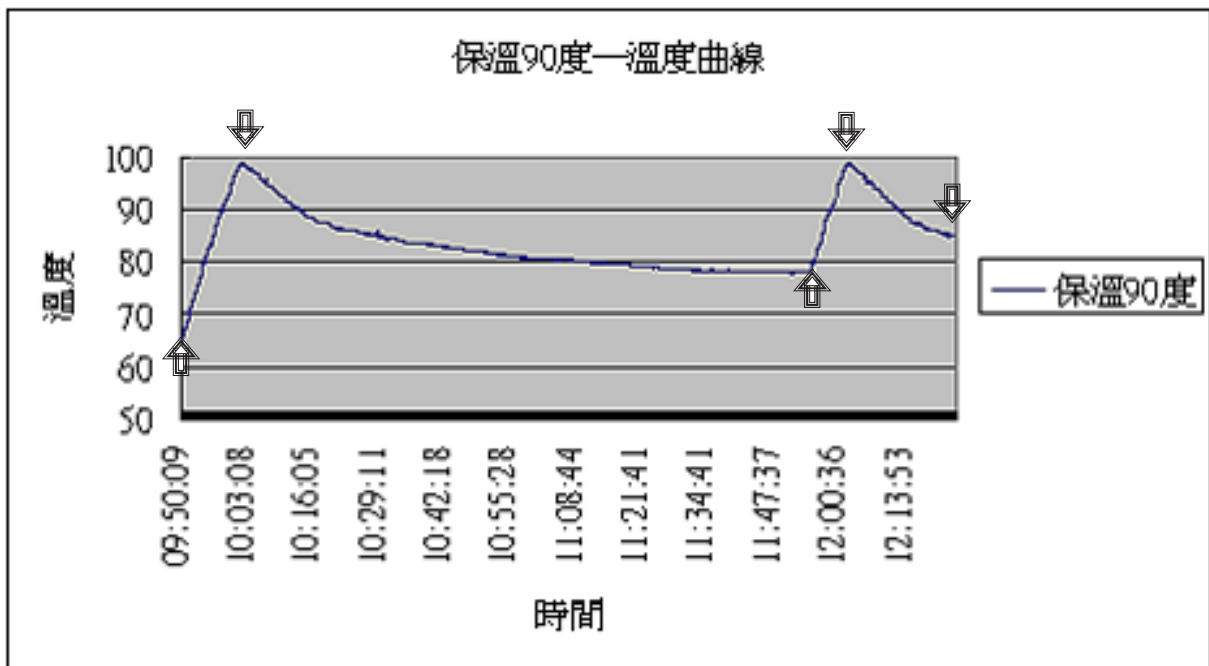
第二條下降曲線：花費 41 分 24 秒，下降 17.2 度，平均每分鐘下降 0.415 度

第三條上昇曲線：花費 05 分 57 秒，上昇 16.9 度，平均每分鐘上昇 2.840 度

第四條下降曲線：花費 22 分 28 秒，下降 13.5 度，平均每分鐘下降 0.6 度

2、加水後設定 90 度保溫模式

一樣在加入3L的水後，我們讓熱水瓶進行煮沸的動作，但是我們這次設定煮沸後保溫 90度。在經過12分後水溫達到最高98.9度，沸騰過後熱水瓶進入我們預設的保溫模式，持續保溫了1小時52分後溫度降到77.8度，此時熱水瓶由保溫模式切入煮沸模式，再經過7分後達到第二次煮沸，此時溫度為98.6度，之後繼續切入保溫模式，22分後溫度值降低到84.9度後我們同樣停止量測，收到資料筆數 7806 筆。



[表二]熱水瓶保溫 90 度下的溫度曲線圖

圖中五個箭頭所指的溫度，由左到右分別為，起始溫度：65.3 度，第一次沸騰：98.9 度，保溫轉再沸騰：77.8 度，第二次沸騰：98.6 度，結束溫度：84.9 度

第一條上昇曲線：花費 12 分 19 秒，上昇 33.6 度，平均每分鐘上昇 2.728 度

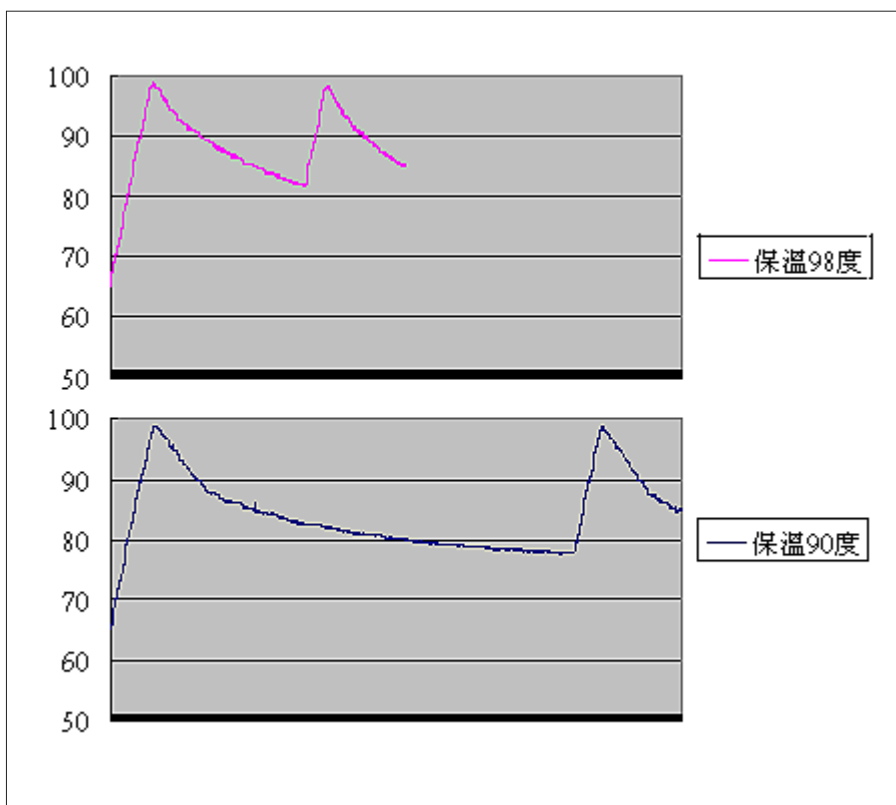
第二條下降曲線：花費 1 小時 52 分 38 秒，下降 20.8 度，平均每分鐘下降 0.188 度

第三條上昇曲線：花費 7 分 55 秒，上昇 20.8 度，平均每分鐘上昇 2.627 度

第四條下降曲線：花費 22 分 17 秒，下降 13.7 度，平均每分鐘下降 0.614 度

數據會說話

由這個實驗我們知道，失去密封空間的熱水瓶，瓶內的溫度是不會一直保持在我們設定的溫度狀態下，當散熱效果高於熱水瓶所能提供的保溫動作時，溫度就會持續呈現下降的曲線。另外，當熱水瓶進入保溫模式後，在某一個觸發點下，它是會轉進沸騰模式，而不是靜止在保溫動作下，平時因為有密封瓶身協助，所以我們都只看的到熱水瓶靜止在保溫下，但事實上，熱水瓶內的IO控制器在保溫動作後，仍然一直在偵測熱水瓶內的水溫，而一但發現溫度降低到一個範圍，控制器不再讓熱水瓶繼續保溫，而是進入到再沸騰模式，避免溫度超過預定的範圍。



[表三]熱水平保溫 98 度與 90 度時的溫度曲線

由圖中我們可以明確看出來，保溫 90 度在第一次沸騰後，經過溫度驟降才趨於平緩。而保溫 98 度一次循環所終止的點，約在圖表的正中央，此時保溫 90 度確仍然在保溫狀態中

另外，在設定保溫90度的圖中我們可以明顯的看到，在到達沸點過後，溫度曲線在剛開始有一度下降相當快速，而後在約88度時雖然仍在持續下降，但是曲線開始趨於平緩。這段曲線的出現表示，熱水瓶在煮沸轉保溫後有一段時間是讓溫度自動下降，所以溫度會下降比較快速，之後為不讓溫度下降過快，所以熱水瓶在此時才真正啟動保溫狀態。同樣的情形在保溫98度時也是有出現的，可是出現的時間較為短暫，也代表著保溫動作出現的比較早，所以我們會比較難觀察出來。

再來，當我們把兩張曲線圖放在一起比較時發現，加溫時的曲線是幾乎相似的，而且上昇相當的穩定。但是沸點過後，就開始出現很大的差異了。設定保溫90度的溫度曲線，除了開始的一小段較為抖峭，之後就漸緩下來了。反觀保溫98度的溫度曲線，沸騰過後的下陷曲線，漸緩的現象不僅不明顯，而且再不久之後就再度轉進加熱狀態了。從圖表裡我們不難發現，同樣的動作，設定保溫90度經過第一次循環時，設定保溫98度已經循環第二次了。整個差距是在進入保溫時段後所拉出來的。保溫90度的保溫時間整整高出1小時8分，也就是說大多的時間熱水瓶都保持在保溫狀態下，相比之下；保溫98度時，電熱水瓶的加熱次數比起保溫90度可是相當頻繁的。換而言之，在保溫98度的情形下，即使在保溫狀況良好，為了保持瓶內水溫在我們的設定下，比起保溫90度，保溫98度時的熱水瓶IO會的切入保溫狀態是相當的頻繁，也難怪保溫90度是省電模式。

保溫滿意標準

說了那麼多，這樣大家對熱水瓶的溫度控制有沒有比較了解呢!?在保溫98度的情況下會轉沸騰的設定點約在81.5度，與預計相差了16.5度了。在保溫90度的情況下，轉成再沸騰的設定點是在77.8度，當中的溫度差為12.2度。理論上這些保溫動作，本來應該由熱水瓶密封的瓶身來維護的，畢竟只靠保溫動作是不夠的。所以選一個良好的密封瓶身可是會讓我們省下不少電力的，下次在挑選熱水瓶時，就知道該以什麼做依據了吧!

程式觀看與註解

先前提到過，我們的程式是用VB6.0完成的，提供程式的製作過程給大家參考一下。首先我們先開啓VB6.0，接下來在點選『檔案』→『建立新專案』→『標準執行檔』，就可以開啓一個專案區來撰寫程式。

由於程式部份將使用到引用一些元件，所以在開起一個VB的新專案後，先到視窗上方的『專案』選項找到『設定使用元件』，並且將下列兩個物件模組打開，不然有些元件可是會無法使用。

Microsoft Comm Control 6.0--程式使用 MSCOMM

Microsoft Scripting Runtime--程式使用FileSystemObject物件

Option Explicit

Dim buffer\$	' 接收資料暫存區
Dim com As Integer	' 連接埠與 ID 值的的宣告
Dim fs As New FileSystemObject	' 建立 fs 爲 FileSystemObject 物件
Dim txtf As TextStream	' 建立 txtf 爲 TextStream 物件
Dim kkk As Integer	' 宣告 kkk 爲一整數

Private Sub Command1_Click()	' 指令輸出
kkk = Val(Text2)	' 將 Text2 的值轉成數質再給 kkk
If Timer1.Enabled = False Then Timer1.Enabled = True	' 啓動 Timer 定時抓取接收值
Call ask_data	' 開始詢問資料
End Sub	

Sub ask_data()	' 送出詢問資料
Call ComPortOpen	' 開啓連接埠
MSComm1.Output = Chr(8)	' 送出詢問指令
Do	' 詢問狀態後等待接收
DoEvents	
buffer\$ = buffer\$ + MSComm1.Input	' 抓取 MSComm 接收到的資料
Loop Until Right(buffer\$, 1) = vbCrLf Or Len(buffer\$) > 10	
' 當暫存區右邊值爲爲 vbCrLf 或長度大於 10 就抓取資料	
Text1 = Mid(buffer\$, 4, 5) & vbCrLf	' 資料僅顯示從第四個位置開始取五位
buffer\$ = ""	' 清空暫存
Text3 = Mid(Time, 3, 9) & vbCrLf	' 時間抓取由第三個位置開始取九位

```

Callfile_open                                '進行資料儲存
If (MSComm1.PortOpen = True) Then MSComm1.PortOpen = False
                                                '關閉連接埠
End Sub
-----
Private Sub Command2_Click()                  '關閉整個程式
Timer1.Enabled=False                         '關閉Timer
End
End Sub
-----
Private Sub Form_Load()
buffer$=""                                    '清空字串暫存區
Text1=""                                     '清空訊息視窗
Timer1.Enabled=False                         '關閉Timer
Timer1.Interval = 1000                      '設定Timer 啟動時間間隔=1S
com = 1                                       '預設 com=1
End Sub
-----
Private Sub Option1_Click()                  '選擇接收資料的 com
com = 1
End Sub
-----
Private Sub Option2_Click()
com = 2
End Sub
-----
Private Sub Option3_Click()
com = 3
End Sub
-----
Private Sub Option4_Click()
com = 4
End Sub
-----
Sub ComPortOpen()                            '連接埠啟動設定
If (MSComm1.PortOpen = True) Then MSComm1.PortOpen = False
'先將 MSComm 關閉再進行設定
MSComm1.CommPort = com                       '預設 com=1
MSComm1.Settings = "9600,n,8,1"

```

```
'連線速度 9600baud、無同位檢查、資料位元 8bit、停止位元 1bit
MSComm1.InputLen = 0
'告訴控制項當使用Input時，讀取整個暫存區
If (MSComm1.PortOpen = False) Then MSComm1.PortOpen = True      '開啓序列連接埠
End Sub

-----

Sub ComPortClose()      '關閉連接埠
If (MSComm1.PortOpen = True) Then MSComm1.PortOpen = False
End Sub

-----

Sub file_open()      '儲存接收到的溫度和時間資料
If fs.FileExists("e:\temp.txt") Then
Set txtf = fs.OpenTextFile("e:\temp.txt", ForAppending, True)
'如果e:\temp.txt已存在則開啓，唯寫且資料由檔案最後面開始寫入
Else
Set txtf = fs.CreateTextFile("e:\temp.txt")      '如果此檔案不存在，則新建一個
End If
txtf.WriteText1.Text      '將text1(溫度)的資料寫入e:\temp.txt
txtf.Close      '關閉檔案
If fs.FileExists("e:\time.txt") Then
Set txtf = fs.OpenTextFile("e:\time.txt", ForAppending, True)
'如果e:\time.txt已存在則開啓，唯寫且資料由檔案最後面開始寫入
Else
Set txtf = fs.CreateTextFile("e:\time.txt")      '如果此檔案不存在，則新建一個
End If
txtf.WriteText3.Text      '將text3(時間)的資料寫入e:\time.txt
txtf.Close      '關閉檔案
Call ComPortClose      '關閉連接埠
End Sub

-----

PrivateSub Timer1_Timer()      '設定Timer
If Text2 > 0 Then Text2 = Text2 - 1      '設定抓取時間並到數
If (Text2 > 0) Then
Call ask_data      '當抓取時間設定後，開始詢問資料
Else
Timer1.Enabled = False      '關閉Timer
End If
End Sub

-----
```

物件說明

MSComm1：進行連接埠傳送的控制

Timer1：設定抓取時間

Text1：顯示抓取資料

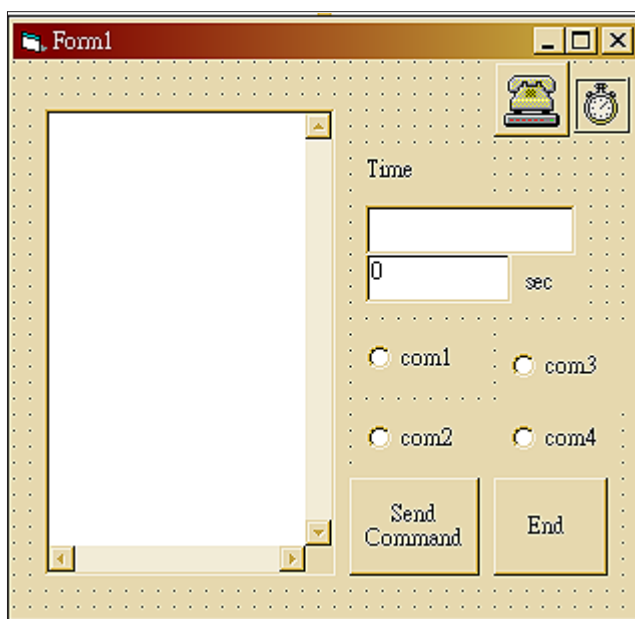
Text2：輸入預計接收時間，開始後
會進行到數動作

Text3：顯示時間

Option1~4：選取com

Command1：開始抓取資料

Command2：結束資料抓取動作



[圖四]開啓一個新專案後，點選『檢視』→『物件』即可利用左方的工具箱來配置物件，如果左方沒有出現工具箱，請到『檢視』→『工具箱』來將它開啓

EXCEL 圖表製作

在資料抓取結束後我們會發現自己電腦E槽底下會多出兩個文字檔(.txt)，這是我們在程式中所設定的，當然也可以指定到別的地方，這兩個檔分別為temp.txt和time.txt，第一份檔案記錄抓取到的溫度，另一份則是記錄時間。接下來我們就可以把資料放進Excel裡面來觀察了，當然您也可以直接把資料存進Excel裡，不過這部份就留給有興趣的您來更改程式碼，現在我們仍然使用最簡單的複製、貼上來完成這次的實驗。

我們先將時間的資料全部複製，接下來開起一份EXCEL檔，將資料貼到EXCEL內的第一個欄位上，溫度的資料則貼到第二個欄位內，接下來就可以開始製作圖表了。

動作一： 插入→圖表

接下來就會有圖表精靈，分四個步驟協助我們完成圖表的繪製。

動作二： 圖表精靈 - 步驟4 之 1 —圖表類型

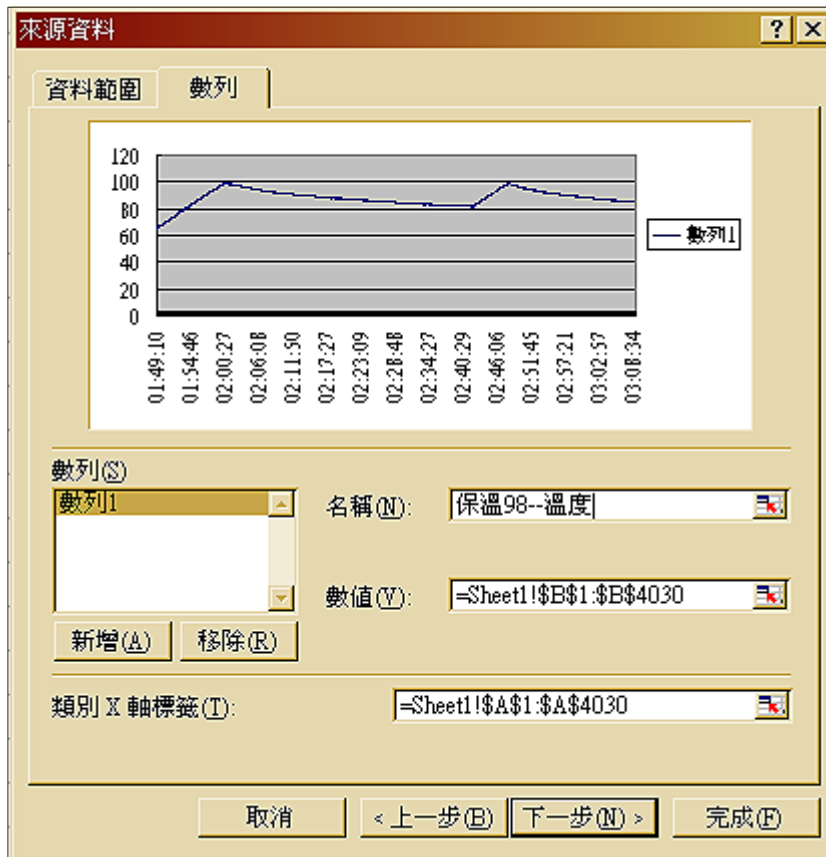
這邊有標準類型和自訂類型兩種，我們選『標準類型』，圖表類型選『折線圖』，副圖表類型選『折線圖』，然後按下一步。



[圖五]在『副圖表類型』的選項，可依個人喜好習慣來選擇，文章中的圖表是用左上角的『折線圖』所繪製而成

動作三：圖表精靈 - 步驟 4 之 2 — 圖表資料來源

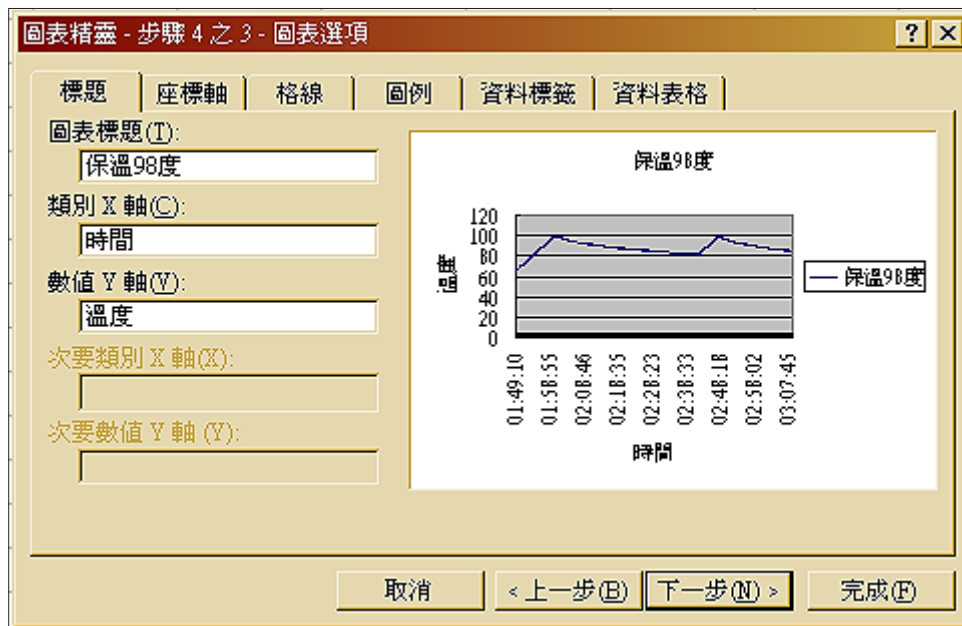
這裡有資料範圍和數列兩個選項，請選『數列』，數列部份如無內容則按新增，如果已有顯示”數列 1”，就不需再新增，直接將『名稱』、『數值』和『類別 X 軸標籤』填上，我們填入的內容如圖，填入後按下一步。



[圖六]圖中填入的『名稱』是代表曲線的名稱，而『數值』內的選項，是直接選取我們貼上的溫度資料，『類別 X 軸標籤』也是直接選取我們貼上的時間資料

動作四：圖表精靈 - 步驟 4 之 3 → 圖表選項 → 標題 — 圖表標題

這部份的選項有表題、座標軸、格線、圖例、資料標籤和資料表格，請選『標題』並輸入『圖表標題』、『類別 X 軸』和『數值 Y 軸』這三項的定義，輸入後就可以按下一步了。



【圖七】此項是要輸入圖表的名稱和 XY 軸代表的意義

動作五：圖表精靈 - 步驟 4 之 4 → 圖表位置

這邊有新的工作表或工作表中的物件，我們選『工作表中的物件』，內容就看您圖表想放在那一區域了，按下完成後您就可以在您選擇的圖表位置上看到我們製作的圖表了。



【圖八】圖表位置一樣可以依個人喜好來選擇，由於我們的資料放在 sheet1(工作表)，所以我們也將繪製成的圖型放在同一個區域內

參考資料

RS485 講座—mscomm的應用	www.chipware.com.tw
Visual Basic 6.0教本—王國榮著	旗標出版
Visual Basic 6.0實戰講座— 王國榮著	旗標出版
Visual Basic 6.0學習範本— 黃世陽 / 吳明哲編	松崗出版
8051 單晶片 - 徹底研究實習篇 - 林伸茂著	旗標出版
CD-JSV30T 使用說明書	象印牌微電腦電動給水熱水瓶
台象家電資訊網	www.zojirushi.com.tw
產品使用資訊	www.chipware.com.tw
旗威論壇	chipware.myvnc.com

旗威相關產品資訊

TEMP_PT	TEMP_PT100溫度計	\$6,000
UIR485	USB2.0-RS485 轉接盒(USB2.0 傳輸線)	\$2,499
PS24V	DC24V 直流電源供應器 -50W	\$ 700

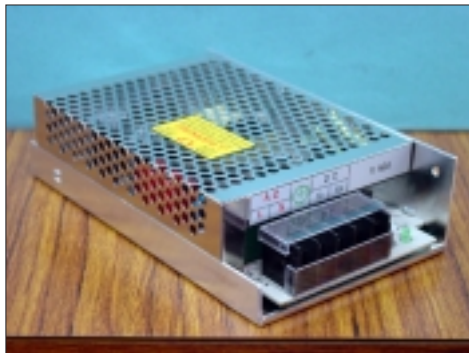


[產品一]TEMP_PT100

此溫度計量測值可由 -20 度到 +120 度的範圍，除了可將瞬間溫度表示在面版上外，還可以透過 UIR485 來傳輸資料，其不銹鋼的量測棒更是為此溫度計加分不少

**[產品二]UIR485**

USB 2.0 轉 RS485 的轉換界面，輸出入點多達 32 個，傳統的 PC 透過此轉換板可和 31 台具備 RS485 界面的週邊設備進行連線

**[產品三]PS24V 系列**

工業用的直流電壓都是以 DC24V 為標準，而 PS24V 系列可以適用 AC110V 及 AC220V 兩種交流電壓輸入，提供可調電阻以修正輸出的電壓，有 50W 及 100W 兩種功率規格可供選擇

如果您需要做相同的實驗時，您需事先準備：

1. 電熱水瓶
2. 個人電腦(有 USB 介面)
3. 溫度量測計(耐溫 +100 度以上且具接收 RS485 串列通訊功能)
4. Microsoft Visual Basic 6.0
5. Microsoft Office Excel
6. USB2.0 與 RS485 轉接器
7. RS485 專用線材
8. USB2.0 專用傳輸線
9. 直流電源供應器