

MSComm 的應用 (四)

上一篇文章提到接收程式用 Timer 的寫法，接著要介紹 Do Loop 的用法，一般來說，Do Loop 是用在已確定回應指令的場合，加入一些適當的判斷，就可以讓接收程式收到完整且正確的資料，然而如果使用不當，也很容易造成系統的死當，我們現在就來看看用 Do Loop 應該怎樣寫。

在寫 Do Loop 用法時，我們需要一個可以接收指令並產生回應指令的裝置來試驗，因此筆者利用 DIO-I 控制板做為一個題材，先介紹一下 DIO-I 控制板的指令用法：

控制端詢問 DIO-I 的狀態值

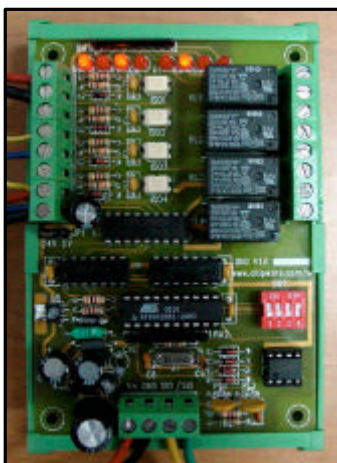
命令下達方式：`CHR(ID) + "?" + CHR(13) + CHR(10)`

DIO-I 控制板會回應

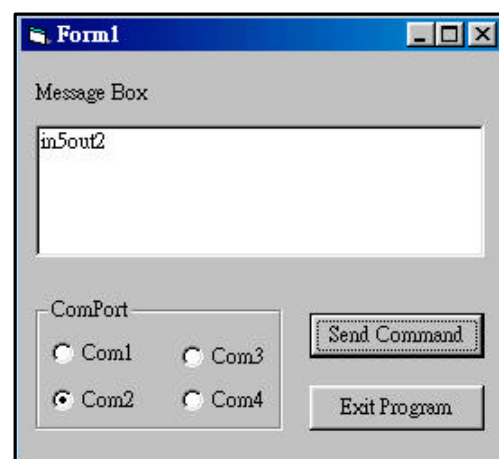
`"in" + ASC1 + "out" + ASC2 + CHR(13) + CHR(10)`

ASC1 與 ASC2 是一個 ASCII 字元，介於 "0"-"9" 與 "A"-"F" 之間，分別代表四個輸出入 bit 的加權狀態值。

上面這一小段是從 DIO-I 控制板的說明書中所節錄出來的，當我們跟 DIO-I 傳送詢問的指令時，DIO-I 就會回應當時的狀況值。比方說現在 DIO-I 的 ID 值是 1，其中輸入點 1 與輸入點 3 有訊號進來，而 RELAY 輸出點 2 正在啟動，那麼我們必須傳送 `CHR(1) + "?" + CHR(13) + CHR(10)` 這四個 Bytes 的碼給 DIO-I 控制板，DIO-I 控制板接收到指令後會回應 `"in5out2" + CHR(13) + CHR(10)`。接著我們來看看這一段程式應該怎麼寫。



〔圖 1〕 DIO-I 控制板的狀態
連線時 IN1 與 IN3 有信號進入，
且 RLY2 正在啟動的狀態。



〔圖 2〕 DIO-I 控制板回應的狀況
程式傳送 `CHR(1) + "?" + CHR(13) + CHR(10)`
後控制板的回應狀況。

DIO_connection.vbp

```
Option Explicit
Dim buffer$          ' 接收資料暫存區
Dim com As Integer  ' 連接埠與 ID 值的宣告
'-----
Private Sub Command1_Click() ' 指令輸出

    Call ComPortOpen      ' 開啟連接埠

    Text1 = ""           ' 清除訊息視窗

    MSComm1.Output = Chr(1) ' 送出指令
    MSComm1.Output = "?"
    MSComm1.Output = vbCrLf

    Do                  ' 詢問狀態後等待接收
        DoEvents
        buffer$ = buffer$ + MSComm1.Input
    Loop Until Len(buffer$) > 8

    Text1 = buffer$
    buffer$ = ""

    Call ComPortClose    ' 關閉連接埠

End Sub
'-----
Private Sub Command2_Click()
    End
End Sub
'-----
Private Sub Form_Load()
    buffer$ = "" ' 表單啟動時將暫存清空
    Text1 = ""
    com = 1
End Sub
'-----
Private Sub Option1_Click() ' 設定連接埠
    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
    ' 先將 MSComm 關閉再進行設定
    If (MSComm1.PortOpen = True) Then MSComm1.PortOpen = False
    ' 預設為 COM 1
    MSComm1.CommPort = com
End Sub
```

```

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

```

在整個接收程式中並沒有用到 Timer，這樣的用法是最簡潔也是最快速的方法，當資料滿足接收的條件時，就立刻將資料顯示在訊息視窗中，接下來筆者將程式中一些重點的部分整理出來，這個部分要仔細想想，不要看過就算了唷！

1. DoEvents 一定要加。

DoEvents 是將程式中 Do Loop 的控制權還給系統的指令，當程式在執行 Do Loop 的時候，如果沒有離開迴圈的設定時，程式會把系統所有的資源都鎖在這個迴圈裡，導致系統非常嚴重的延遲情形，甚至是呈現死當的狀況，因此撰寫程式時一定要把 DoEvents 加進來，避開程式產生無窮迴圈的危險。

2. Loop Until ... 的條件撰寫，儘量不要使用絕對式的寫法。

何謂絕對式的條件判斷？這個很難形容，用一行程式來寫您就明瞭了。

```
Loop Until Len (string$) = 4
```

上面這樣的寫法，就是絕對式條件判斷的寫法。為什麼不可以這樣寫呢？因為串列通訊的輸入訊號固然很不容易受到電氣的影響，也不太容易衰減，但是這並不代表傳輸資料的過程一定不會有錯誤訊息的產生，如果此刻 string\$ 的長度，因系統執行其它動作的延遲而只收到 3 個 Bytes 的資料，或是因電氣雜訊而多收了一個 Byte 的資料，那麼這個迴圈就會一直停在這裡等待條件成立，巧的是它左等右等就是不成立，因為寫法是用絕對式的。

要避開這樣的缺點，有兩種補救的方法：第一種就是像程式中所寫的一樣，改成 “ > ” 的寫法；如有絕對必要寫到 “ = ” 的用法時，請利用 OR 多加入一些判斷(如計時計數判斷)，這樣一來，所有設定條件都不成立的機率就會降低。不過最好的程式寫法還是要用第一種方法會比較好。

到此為止，您對 Do Loop 的用法應該也很熟悉了，換個方式思考一下，使用 Timer 與使用 Do Loop 的最大差異在哪裡？筆者將兩者的差異整理成下面的列表：

	Timer	Do Loop
運作原理	強制接收	已知回應指令接收
接收條件	定時觸發	滿足接收條件則接收
使用限制	時間間隔必須大於傳送指令所需花費的時間	1. 加入 DoEvents 指令 2. 避免絕對式條件判斷

您對 MSComm 的傳送與接收有完整的概念了嗎？接下來的文章，要進入 RS485 的另一個領域囉！趕快消化吸收一下吧！

參考資料：

1. DIO-I 控制板 / 旗威科技 – 介紹主題
2. UIR485 轉接盒 / 旗威科技 – 連線裝置
3. 『8051 單晶片徹底研究』系列叢書 / 旗標出版社 – 程式連線相關介紹