系統的誕生(二)

DIO-I 控制板核心程式介紹

要開始使用 DIO-I 控制板之前，我們先來瞭解一下整個 DIO-I 控制板的核心程式是如何運作的，底下是 DIO-I 控制核心的完整程式碼，筆者將程式分解為數個方塊，詳細地為各位介紹該方塊的功能及其使用上的限制：

```
;
;---DIO PROGRAM BY ASSEMBLER---
;
MS10      EQU     9217        ;10mS Timer 10mS 中斷的運算值
B_RATE    EQU     FDH         ;Baud_Rate 鮑率的設定(9600 bps)
; 計時器高低位元組的設定值
TMH       EQU     (65536-MS10) / 256          ;Timer high byte setting
TML       EQU     (65536-MS10) .MOD. 256      ;Timer low byte setting
; 換行指令的 ASCII 值
CR        EQU     0DH
LF        EQU     0AH
;
RLY0      REG     P1.7        ;Relay setting  輸出點的對應腳位
RLY1      REG     P1.6
RLY2      REG     P1.5
RLY3      REG     P1.4
DRIVE     REG     P3.7        ;Serial drive setting  串列通訊驅動腳位
;
; 狀態判斷位元 ( KEY = 1,  狀態 OK；KEY = 0,  錯誤狀態)
CHK_KEY   REG     29H.0       ;KEY=1 ← OK, KEY=0 ← ERROR
;
; 動作判斷位元 ( MOTION = 1,  輸出點動作；MOTION = 0,  資料傳送)
MOTION    REG     29H.1       ;MOTION=1 ← RELAY WORK, MOTION=0 ← DATA SEND
; 暫存器設定
ID        EQU     2AH         ; ID 值暫存器
INTR_CNT  EQU     2BH         ; 內部計時暫存器
DG_IN     EQU     2CH         ; 輸入點狀態暫存器
DG_OUT    EQU     2DH         ; 輸出點狀態暫存器
I_CNT     EQU     2EH         ; 串列輸入字數計算器
O_CNT     EQU     2FH         ; 串列輸出字數計算器
I_BUF     EQU     40H         ; 串列輸入字串暫存器 ( 40H-4FH )16 BYTES
O_BUF     EQU     30H         ; 串列輸出字串暫存器 ( 30H-3FH )16 BYTES
STACK     EQU     50H         ; 堆疊起點宣告
```

這一段是 DIO-I 的系統參數宣告，在以上所列的參數，有幾個重要需要特別強調：

1. 程式中需要用到位元定址的部份，必須將暫存器宣告在 20H~29H 之間。

```
DG_IN     EQU     2CH
DG_IN.0 → 2CH.0   (OK)
```


旗威科技有限公司　地址：高雄市三民區昌裕街 18-1 號　網址：http://www.chipware.com.tw
技術專線：07-395-5152　技術支援傳真：07-395-5155　E-mail：chipware@chipware.com.tw

```
DG_OUT  EQU     3CH
```

DG_OUT.1 → 3CH.1　(ERROR)

2. 程式中設定為串列輸入的暫存器，必須設定在最後，避免因接受了不正確的字串長度而改變了其它設定值。我們看看 I_BUF 的設定值是 40H，接下來就是 STACK 的 50H，其中並不會影響到其它的設定，如果我們改成 I_BUF 是 30H，O_BUF 是 40H，那麼當我們收到的字串是 17 個 BYTES 時，那麼 O_BUF 的第一個位元組很可能就會被更改或是產生錯誤，這是在軟體撰寫之初必須考慮進去的。

3. STACK 的宣告必須在這裡宣告，儘量避免在主程式中直接輸入數值，以免將來程式堆疊不夠時，修改會變得很麻煩，而常用且會經常性更改的變數，也盡量在這裡宣告，如 TMH、TML、CR、LF……等等，這都有助於日後改寫程式時的除錯與修正。

以上的參數設定值，筆者建議除非必要否則不要隨意修改，以免造成系統的誤動作或當機。

```
;
;---FOR FUNCTION SETTING---
;
X1          REG         DG_IN.0         ; 輸入腳位設定
X2          REG         DG_IN.1         ; X1   X4 分別代表 IN_1   IN4
X3          REG         DG_IN.2
X4          REG         DG_IN.3
Y1          REG         DG_OUT.0        ; 輸出腳位設定
Y2          REG         DG_OUT.1        ; Y1   Y4 分別代表 OUT_1   OUT4
Y3          REG         DG_OUT.2
Y4          REG         DG_OUT.3
T1          EQU         28H             ; 計時器的設定
T2          EQU         27H             ; T1 及 T2 都是以秒為單位的計時器
;
RESULT      REG         29H.2           ; 判斷用暫存位元 1
OLD_R       REG         29H.3           ; 判斷用暫存位元 2
```

接下來的這一段是提供給使用者應用的設定，這裡包含了四個輸入及四個輸出設定，還有兩個每秒一數的計時器可以使用，而 RESULT 是判斷結果的暫存位元，OLD_R 則是存放前一次判斷結果的暫存位元。如果您所需要的系統有更多的需要，可以從這裡增加您所需要的變數。

```
;
;---PROGRAM STARTER---
;
        ORG     0000H       ;MAIN PROGRAM
        JMP     START
        ORG     0003H
        RETI
        ORG     000BH       ;TIMER0 INTERRUPT PROGRAM
        JMP     T0_ISR
```


旗威科技有限公司　地址：高雄市三民區昌裕街 18-1 號　網址：http://www.chipware.com.tw
技術專線：07-395-5152　技術支援傳真：07-395-5155　E-mail：chipware@chipware.com.tw

```
;
        ORG     0013H
        RETI
        ORG     001BH
        RETI
        ORG     0023H     ;SERIAL INTERRUPT PROGRAM
        JMP     SI0_ISR
```

這一段是設定程式的起點，主程式的起點是從 0000H 開始，而外部觸發、定時中斷、及串列中斷的起點可參考 MCS-51 的介紹，在此我們只用到 Timer0 的中斷及串列中斷，起點分別為 T0_ISR 及 SI0_ISR。筆者會建議您在撰寫程式時都能把這一段完整地寫出來，不管您是否需要用到中斷，如此一來當您的程式需要擴充時，會有很大的彈性空間可以應用。

```
;
;---MAIN PROGRAM---
;
        ORG     0030H
START   MOV     R2,#00H
$       DJNZ    R2,$            ; 系統延遲一小段時間
        MOV     SP,#STACK       ; 堆疊宣告
        MOV     P1,#FFH         ; 清除所有的狀態
        MOV     P3,#FFH
        MOV     INTR_CNT,#00H
        MOV     I_CNT,#00H
        MOV     O_CNT,#00H
        MOV     DG_IN,#00H
        MOV     DG_OUT,#00H
;FOR FUNCTION 這裡是可以修改的
        MOV     T1,#00H
        MOV     T2,#00H
        CLR     OLD_R
        CLR     RESULT
;CLR IO BUFFER
        MOV     R1,#I_BUF
        MOV     R7,#0AH
$1      MOV     @R1,#FFH
        INC     R1
        DJNZ    R7,$1
        MOV     R0,#O_BUF
        MOV     R7,#0AH
$2      MOV     @R0,#00H
        INC     R0
        DJNZ    R7,$2
;
        CALL    READ_ID         ; 儲存 ID 設定
        CALL    SET_ISR         ; 啟動中斷
;
$LOOP   MOV     A,ID            ; 如果 ID 為 0 則不做任何動作
        CJNE    A,#0,$FUNC
        JMP     $LOOP
$FUNC   CALL    FUNCTION        ; 如果 ID 不為 0 則執行 FUNCTION 副程式的動作
        JMP     $FUNC
```

這一個方塊便是主程式的內容：一開始進入系統時，先讓系統延遲一小段時間以達到穩定的狀態，接著宣告堆疊並清空所有暫存器內的資料。這些動作都做好了之後，便要儲存 DIO-I

旗威科技有限公司　地址：高雄市三民區昌裕街 18-1 號　網址：http://www.chipware.com.tw
技術專線：07-395-5152　技術支援傳真：07-395-5155　E-mail：chipware@chipware.com.tw

的 ID 值，並啟動 10mS 定時中斷及串列中斷。最後進入一個迴圈讓系統反覆執行您所設計的程式，同時在定時中斷裡加入所有和時間有關的動作。

在這裡所必須注意的是關於 FUNCTION 副程式的部份，當 FUNCTION 的動作裡和串列中斷所宣告的動作有所衝突時，FUNCTION 的優先權應該是比較高的，因為 FUNCTION 算是主程式的一部份，如果強制宣告串列中斷的優先權大於主程式時，很可能會造成系統的不穩定或當機，因此當您的 FUNCTION 副程式已經在控制輸出點的同時，請不要再以遠端控制指令強制系統執行串列所指定的動作，這樣一定會造成兩個動作打架的，而不管是誰打贏，您的程式設計都算輸了。

同樣地，這一段程式除了提供給 FUNCTION 用的變數外，也儘量不要隨意修改，除非必要。

```
;
;---NORMAL CONTROL---
; IF X1 = TRUE & X2 = TRUE THEN Y1 = ON 5 SEC
; C1=1 → X1 & X2 = 1
; C1=0 → X1 & X2 = 0
;
FUNCTION
;INPUT
      MOV       C,X1
      ANL       C,X2
      MOV       RESULT,C
;
;CHECK STATE
      MOV       C,RESULT
      JNC       $1          ;RESULT=FALSE
;RESULT=TRUE
      MOV C,OLD_R
      JC        $1          ;OLD=TRUE
;OLD=FALSE
      MOV       T1,#5
;
;OUTPUT
$1    MOV       A,T1
      JNZ       $ON         ;T1>0
;T1=0
      CLR       Y1
      JMP       $END
$ON
      SETB      Y1
$END
      MOV       C,RESULT
      MOV       OLD_R,C
      RET
```

FUNCTION 副程式是一小段 PLC 的寫法介紹，動作是當 X1 及 X2 同時有輸入時，Y1 就啟動五秒。而動作的順序是先讀入輸入狀態，再進行狀態的判斷，最後才將判斷好的動作輸出，並更新暫存器的設定值。

旗威科技有限公司　地址：高雄市三民區昌裕街 18-1 號　網址：http://www.chipware.com.tw
技術專線：07-395-5152　技術支援傳真：07-395-5155　E-mail：chipware@chipware.com.tw

```
;
;---ID SETTING; SWITCH STATE---
;
READ_ID
        MOV     A,P1
        CPL     A
        ANL     A,#0FH
        CJNE    A,#CR,$1
        MOV     A,#17
        JMP     $2
$1      CJNE    A,#LF,$2
        MOV     A,#16
$2      MOV     ID,A
        RET
```

讀取 ID 的副程式，可用的值為 0　15，其中 ID0 有特殊作用，請在規劃程式時避開這個 ID 不要使用。

```
;
;---SERIAL INTERRUPT PROGRAM---
;
SI0_ISR
        PUSH    A
        PUSH    PSW
;
        JB      RI,$RXD
        JMP     $1
;READ DATA
$RXD
        CLR     RI
;
        CALL    STR_OVER
        JNB     CHK_KEY,$1
$INPUT
        MOV     A,SBUF
        CALL    SAVE_INPUT_DATA
        CJNE    A,#LF,$1        ;A=LF → INPUT DATA END
        MOV     A,I_BUF         ;CHECK ID
        CJNE    A,ID,$1
        CALL    CHECK_COMMAND       ;CHECK COMMAND
        JB      MOTION,$RLY  ;MOTION=1 → RELAY WORK
        CALL    ENQUIRY         ;MOTION=0 → SEND DATA
JMP $C_BUF
$RLY
        CALL    OUTPUT
;CLEAR INPUT BUFFER
$C_BUF
        MOV     R4,#0AH
$LOOP
        MOV     R1,#I_BUF
        MOV     @R1,#00H
        INC     R1
        DJNZ    R4,$LOOP
        MOV     I_CNT,#00H
;
$1      JB      TI,$TXD
        JMP     $END
```


旗威科技有限公司　地址：高雄市三民區昌裕街 18-1 號　網址：http://www.chipware.com.tw
技術專線：07-395-5152　技術支援傳真：07-395-5155　E-mail：chipware@chipware.com.tw

```
;TRAMSMIT DATA
$TXD
        CLR     TI
        INC     O_CNT
        INC     R0
        MOV     A,O_CNT
        CJNE    A,#10,$NO_TX
$NO_TX
        JC      $SEND
        SETB    DRIVE
        JMP     $END
$SEND
        MOV     SBUF,@R0
;
$END
        POP     PSW
        POP     A
RETI
```

緊接著這一個方塊是串列中斷的設定，所有的輸出入指令都是由這裡設定的。在進入中斷之前，別忘了把累加器和程式狀態字元的數值先存起來，需要的話也將您所用到的暫存器值一併存起來，以免不小心改到系統的設定值。

```
;
;---CHECK INPUT DATA LENGTH---
;
STR_OVER
        MOV     A,I_CNT
        CJNE    A,#5,$WORDS
$WORDS
        JC      $OK
        MOV     I_CNT,#00H
        MOV     R1,#I_BUF
        CLR     CHK_KEY     ;CHK_KEY=0 → OVER LENGTH → I_CNT=0, R1=I_BUF+0
        JMP     $END
$OK
        SETB    CHK_KEY     ;CHK_KEY=1 → OK
$END
        RET
```

如果您擔心不正常串列輸入的字串會影響堆疊的值，那麼就加入上面這段 INPUT DATA 的長度判斷吧！

```
;
;---SAVE DATA TO BUFFER---
;
SAVE_INPUT_DATA
        CJNE    A,#0,$1
$1      JNC     $2
        JMP     $END
$2      CJNE    A,#32,$3
$3      JC      $4
        JMP     $END
$4      CJNE    A,#CR,$5
        JMP     $END
$5      CJNE    A,#LF,$6
        JMP     $END
```


旗威科技有限公司　地址：高雄市三民區昌裕街 18-1 號　網址：http://www.chipware.com.tw
技術專線：07-395-5152　技術支援傳真：07-395-5155　E-mail：chipware@chipware.com.tw

```
$6    MOV     R1,#I_BUF
      MOV     I_CNT,#0
$END
      CALL    UPPER
      MOV     @R1,A
      INC     R1
      INC     I_CNT
      RET
```

以上是將串列輸入的字串存到 O_BUF 中。

```
;
;---TO UPPER---
;
UPPER
      CJNE    A,#'a',$1
$1    JC      $END
      CJNE    A,#'z'+1,$2
$2    JNC     $END
      CLR     C
      SUBB    A,#20H
$END
      RET
```

你的程式有大小寫轉換的問題嗎？加入 TO UPPER 就可以讓所有的小寫都變成大寫囉。

```
;
;---CHECK INPUT COMMAND---
;
CHECK_COMMAND
      MOV     R1,#I_BUF
      INC     R1
      CJNE    @R1,#'O',$1
      INC     R1
      INC     R1
      CJNE    @R1,#CR,$ERR
      INC     R1
      CJNE    @R1,#LF,$ERR
      SETB    MOTION
      JMP     $END
$1    CJNE    @R1,#'?',$ERR
      INC     R1
      CJNE    @R1,#CR,$ERR
      INC     R1
      CJNE    @R1,#LF,$ERR
      CLR     MOTION
      JMP     $END
$ERR
      JMP     $END
$END
      RET
```

以上這一段程式是用來判斷串列輸入指令是否正確，並執行其對應的功能。

```
;
;---SAVE RELAY STATE TO DG_OUT---
;
OUTPUT
        MOV     A,I_BUF+2
;'0' =< ACC =< '9'
        CJNE    A,#'0',$1
$1      JC      $END
        CJNE    A,#('9'+1),$2
$2      JNC     $A
        CLR     C
        SUBB    A,#30H
        MOV     DG_OUT,A
        JMP     $END
;'A' =< ACC =< 'F'
$A      CJNE    A,#'A',$3
$3      JC      $END
        CJNE    A,#('F'+1),$4
$4      JNC     $END
        CLR     C
        SUBB    A,#55
        MOV     DG_OUT,A
$END
        RET
```

這裡的程式是用來儲存定時中斷時所得到輸出點狀態，寫法上比較需要注意的（I_BUF+2）這樣的用法，在程式中所有與變數有關的數值，都不可以出現絕對性的數值，而必須以相對性的寫法。如果您這裡寫成 MOV A,42H，那麼日後在修改時一定會很容易忽略這個值的修改，這是必須要小心的。

```
;
;---SAVE OUTPUT DATA TO O_BUF & SEND 1ST BYTE---
;
ENQUIRY
        MOV     R0,#O_BUF
        MOV     @R0,#'i'
        INC     R0
        MOV     @R0,#'n'
        INC     R0
        MOV     A,DG_IN
        CALL    BIN2ASC
        MOV     @R0,A
        INC     R0
        MOV     @R0,#'o'
```

```
        INC     R0
        MOV     @R0,#'u'
        INC     R0
        MOV     @R0,#'t'
        INC     R0
        MOV     A,DG_OUT
        CALL    BIN2ASC
        MOV     @R0,A
        INC     R0
        MOV     @R0,#CR
        INC     R0
        MOV     @R0,#LF
;
        CLR     DRIVE     ;ENABLE DRIVE
        MOV     R0,#O_BUF
        MOV     SBUF,@R0
        MOV     O_CNT,#00H
        RET
```

這一段是用來將串列輸出的值存入暫存器中，並啟動串列輸出。

```
;
;---TIMER INTERRUPT PROGRAM---
;
T0_ISR
        MOV     TH0,#TMH
        MOV     TL0,#TML
        PUSH    A
        PUSH    PSW
;SAVE INPUT STATE TO DG_IN
$input_state
        MOV     A,P3
        RR      A
        RR      A
        CPL     A
        ANL     A,#0FH
        MOV     DG_IN,A
;ID=0 -> DG_OUT=DG_IN
$id0_motion
        MOV     A,ID
        INC     A
        DJNZ    A,$out_state
        MOV     DG_OUT,DG_IN
;RELAY WORK BY DG_OUT DATA
$out_state
        MOV     A,DG_OUT
        JB      ACC.0,$ON_0
        SETB    RLY0
        JMP     $1
$ON_0
        CLR     RLY0
$1      JB      ACC.1,$ON_1
        SETB    RLY1
        JMP     $2
$ON_1
        CLR     RLY1
$2      JB      ACC.2,$ON_2
        SETB    RLY2
        JMP     $3
$ON_2
        CLR     RLY2
$3      JB      ACC.3,$ON_3
```

```
        SETB    RLY3
        JMP     $counter
$ON_3
        CLR     RLY3
;1 sec COUNTER
$counter
        INC     INTR_CNT
        MOV     A,INTR_CNT
        CJNE    A,#100,$T0_END
        MOV     INTR_CNT,#00H
;1S COUNTER
        MOV     A,T1
        JZ      $T2
        DEC     T1
$T2     MOV     A,T2
        JZ      $T0_END
        DEC     T2
;
$T0_END
        POP     PSW
        POP     A
        RETI
```

以上這一段是定時中斷的程式內容，所有的動作必須在 10mS 的時間內做完，否則會造成系統的時間 DELAY，產生錯誤的動作，而在定時中斷裡，包含了輸入狀態的讀取、輸出狀態的改變、及計時器的倒數功能。

```
;
;---INTERRUPT SETTING---
;
SET_ISR
        MOV     TMOD,#21H
        MOV     TCON,#50H
        MOV     TH0,#TMH
        MOV     TL0,#TML
        MOV     SCON,#50H
        MOV     TH1,#B_RATE
        SETB    ET0
        SETB    ES
        SETB    EA
        RET
```

這一段是設定中斷的副程式，一定要記得在主程式中啟動，不然所有的中斷都無法使用了。

旗威科技有限公司　地址：高雄市三民區昌裕街 18-1 號　網址：http://www.chipware.com.tw
技術專線：07-395-5152　技術支援傳真：07-395-5155　E-mail：chipware@chipware.com.tw

```
;
;---BINARY TO ASCII---
;
BIN2ASC
        CJNE    A,#9,$1
        JMP     $NUM
$1      JC      $NUM
        SUBB    A,#10
        ADD     A,#41H
        JMP     $OUT
$NUM
        ADD     A,#30H
$OUT
        RET
```

最後這個部份是將要輸出的二進位值轉換成 ASCII 的值

****

筆者將整個程式的重心放在 FUNCTION 的改寫上，其中要再次強調的有以下幾個重點：

1. FUNCTION 副程式中的動作，視為主程式的動作，比任何中斷的動作都具備更高的優先權，如果有需要利用遠端控制（串列指令）來改變輸出點的狀態時，一定要先將 FUNCTION 的功能關閉，比較好的方法是加入 RMT 的指令（遠端控制串列指令）來啟動或關閉 FUNCTION 的運作。

2. 由於定時中斷的時間是 10mS 的時間，因此 FUNCTION 的動作要盡量控制在 10mS 可以做完的範圍內，否則就要拆開成數個部份，以免影響動作的時間程序產生誤差。

3. FUNCTION 中所會用到的暫存器，必須在主程式中宣告，並在一啟動時將其清除，如有位元定址的需要，必須將暫存器的值定在 20H　2FH 這個範圍內，否則可能會造成程式無法運作或產生誤動作。

到此為止，您是否對 DIO-I 控制板的核心有了一些概念了呢？下一篇文章我們要正式將軟硬體結合起來囉。趕緊將這個核心程式消化一下吧！

旗威科技有限公司　地址：高雄市三民區昌裕街 18-1 號　網址：http://www.chipware.com.tw
技術專線：07-395-5152　技術支援傳真：07-395-5155　E-mail：chipware@chipware.com.tw