

## 自動日期計數器

### 傳統與突破

所謂資料庫 IC (DATA BANK) 其實只是結合 LCD 驅動器、輸入輸出埠 (I/O) 以及較一般通用型微控制器更大的記憶體的一種單晶片，它是專門設計用來開發電子記事本、萬年曆、計算機的 IC。與通用型微控制器最大的區別是功能著重於資料的處理，隨著人們對資訊需求的渴望，此類消費性產品如 PDA，在忙碌的 e 世代生活中，已成為一般人愈來愈有必要的裝備。國內義隆電子已經在 DATA BANK 投注大筆的心力，並且積極在教育界推廣，同時也是同業中唯一提供 OTP 型 IC 的業者，所謂 OTP (One Time Programming) 是指能夠燒錄一次的可程式 IC，這項技術使得即使是資金不很充裕的個人工作室，在程式開發初期，不必再花大筆的錢承受開光罩 (mask) 的風險，亦可縮短產品的交貨期。在開發工具方面，義隆提供了十分廉價的線上模擬器 (ICE)，雖說廉價但是功能可是一點都不陽春，只要是 ICE 該有的功能一樣都不少，比起國外開發的 ICE，義隆採用自行開發的 ASIC 當作核心控制器，能夠大幅降低成本，在維修上也方便許多。

本刊專欄的目的就是為了縮短資料庫 IC 設計者的開發時間，同時也提供初學者一個學習捷徑而設計的，目前國內使用 EM78 組合語言來開發程式的人已逐步增加，如果您過去曾學習過任何一種 EM78 程式，那真是恭喜您了，因為就本質上來說所有的 EM78 組合語言都是如出一撤，僅在 IC 的功能

特性上有些許差異，您可對照期技術手冊查閱，如果您未曾接觸過此類 IC 族群，還是可以花很短的時間就可上手，筆者建議學習 EM78 組合語言並非沒有理由的，想想看有多少人已經在使用 8051，不可諱言的它比較適合用在一些高單價產品開發，因為它的價格仍無法和一些國產的微控制器相抗衡，而且缺少看門狗 (Watch-dog) 計時器以及耗電問題，也使它顯得不夠完美，正好 EM78 系列微控制器可以彌補這些缺失，所以如果您還能多會一種微控制器語言，就可以大小通吃。

閒話不多說，還記得筆者過去曾在某技術雜誌上寫過一篇以 EM78860 設計萬年曆速算法，得到很大的迴響，因為開發一套 DATA BANK 的中間過程是十分繁瑣複雜的，如果每次開發不同的機種都要重寫一套相同的程式那不是很麻煩的事嗎？所以我們可以將一些常用的程式制定好統一的介面，然後將它們模組化之後就成了一個軟體資料庫，以後就直接拿來用了，就算是別人寫的，只要遵循輸入輸出程序，也可以直接拿來用，節省開發時間，因此筆者整理出一些 DATA BANK 經常會用到的壓箱工具，打算和讀者分享一些心得，當我們設計的模組愈來愈多時，相對地就可以縮短不少寫程式的時間了。

### 日期問題

本篇的開場白就先以一個最經常遇到的問題開始吧！任何一部 DATA BANK 都需要用到計時器的功能，這也

是最基本的功能，當時間走到換日線的時候，要將時間往後推進一天，這下麻煩就接踵而至了，因為日期的推進可能影響到「日」、「月」甚至是「年」的變化，中間會碰到大小月份的問題，每逢二月還要考慮閏年的問題，於是想起來就開始頭大了，筆者也覺得這個問題很囉唆，所以就順手將它給模組化了，如程式一所示，INC\_1DAY這個模組的用法是每呼叫一次，日期就會往後推一日，它會自動判斷大小月及閏年的問題，簡單地說，它是一個自動日期計數器，在程式中需要用到以下四組變數：

YEAR\_L：存放「年」的低位元組。

YEAR\_H：存放「年」的高位元組。

MONTH：存放「月」。

DAY：存放「日」。

我們預設單位「年」的存放範圍在1900~2099，所以它是以一組十六位元暫存器存放，「月」的範圍是1~12，「日」的範圍在1~31之間。假設目前是公元1999年12月31日，存在記憶體的情形是：

YEAR\_L = 0XCF      YEAR\_H = 0X07

MONTH = 0X0C      DAY = 0X1F

因為存放年份需要一組16位元的記憶體，所以採用兩個8位元的記憶體YEAR\_L、YEAR\_H完成，其餘的變數只需要用到8位元的記憶體就能做到。呼叫之後記憶體的內容變化：

YEAR\_L = 0XD0      YEAR\_H = 0X07

MONTH = 0X01      DAY = 0X01

日期往後推一天，成了2000年1月1日，好了！爾後只要碰到要計算日期的問題時，相信您已經知道該怎麼做了，模組化之後的確方便多了。為了程式能夠順利執行，需要用到以下一些暫存器變數，這些變數位址您可以自行修改後放到适合自己程式的位址去，在呼叫副程式模組之後，該變數位址即可釋放出來。

```
STATUS == 0X03
R10    == 0X10
R11    == 0X11
R12    == 0X12
R13    == 0X13
YEAR_L == 0X20
YEAR_H == 0X21
MONTH  == 0X22
DAY    == 0X23
```

#### 程式一：

;這個表格紀錄每月的最大天數

```
CALEN_MAXDAY:
    TBL
    RETL @31 ; 1月
    RETL @28 ; 2月
    RETL @31 ; 3月
    RETL @30 ; 4月
    RETL @31 ; 5月
    RETL @30 ; 6月
    RETL @31 ; 7月
    RETL @31 ; 8月
    RETL @30 ; 9月
    RETL @31 ; 10月
    RETL @30 ; 11月
    RETL @31 ; 12月
```

```
INC_1DAY      PROC
;*****
; 副程式模組：日期增加一天
;*****
; **先將 YEAR-1900 化作 8 位元變數
    MOV    A,@0X6C
    MOV    R10,A
    MOV    A,@0X07
    MOV    R11,A
; **以取 2 補數方式替代減法
    COM    R10
```

```

COM      R11
JZ       R10
JMP      $+2
INC      R11
MOV      A, YEAR_L
ADD      R10, A
JBC      STATUS, C
INC      R11
MOV      A, YEAR_H
ADD      R11, A
MOV      A, R10
MOV      R12, A
; **R12=0~199, 表示 1900~2099 年
; **由 R12 判斷是否為閏年
AND      A, @4-1
JBS      STATUS, Z
JMP      NOT_LEAP_YEAR
MOV      A, MONTH
XOR      A, @2
JBS      STATUS, Z
JMP      NOT_LEAP_YEAR
; **若為閏年且適逢 2 月, 總天數為 29 天
MOV      A, @29
MOV      R13, A
JMP      GET_MAXDAY
NOT_LEAP_YEAR:
; **非閏年直接查表
DECA    MONTH
CALL    CALEN_MAXDAY
MOV     R13, A
GET_MAXDAY:
INCA    DAY
SUB     A, R13
JBS     STATUS, C
JMP     INC_MONTH
INC     DAY
JMP     INC_DATE_OK
INC_MONTH:
MOV     A, @1
MOV     DAY, A
INCA    MONTH
SUB     A, @12
JBS     STATUS, C
JMP     INC_YEAR
INC     MONTH
JMP     INC_DATE_OK
INC_YEAR:
MOV     A, @1
MOV     MONTH, A
ADD     YEAR_L, A
JBC     STATUS, C
ADD     YEAR_H, A
MOV     A, @199
SUB     A, DATA
JBS     STATUS, C
JMP     INC_DATE_OK
MOV     A, @0X6C
MOV     YEAR_L, A

```

```

MOV      A, @0X07
MOV      YEAR_H, A
INC_DATE_OK:
RET
ENDP

```

要補充說明的是程式一執行時，如果日期已經累進到最大值的情況下，會回到最初的日期 1900/01/01。

程式二的作用剛好和程式一相反，它的目的是 DEC\_1DAY 每次被呼叫一次的時候，會將儲存在記憶體的日子遞減一日，如果日期已經是 1900 年 1 月 1 日時，在程式二執行後日期會設定為 2099 年 12 月 31 日。

## 程式二

```

DEC_1DAY  PROC
; *****
; 副程式模組：日期減少一天
; *****
MOV      A, @0X6C
MOV      R10, A
MOV      A, @0X07
MOV      R11, A
COM      R10
COM      R11
JZ       R10
JMP      $+2
INC      R11
MOV      A, YEAR_L
ADD      R10, A
JBC      STATUS, C
INC      R11
MOV      A, YEAR_H
ADD      R11, A
MOV      A, R10
; **年份存在 R10
MOV      R12, A
DEC      DAY
MOV      A, DAY
JBS      STATUS, Z
JMP      DEC_DATE_OK
DEC      MONTH
MOV      A, MONTH
JBS      STATUS, Z
JMP      R_GET_MAXDAY
MOV      A, DATA
JBS      STATUS, Z
JMP      DEC_YEAR
MOV      A, @0X33

```

```

        MOV     YEAR_L,A
        MOV     A,@OX08
        MOV     YEAR_H,A
        JMP     MAX_MONTH
DEC_YEAR:
; **減少一年
        MOV     A,@1
        SUB     YEAR_L,A
        JBS     STATUS,C
        SUB     YEAR_H,A
MAX_MONTH:
        MOV     A,@12
        MOV     MONTH,A
R_GET_MAXDAY:
; **由 R12 判斷是否為閏年
        MOV     A,R12
        AND     A,@4-1
        JBS     STATUS,Z
        JMP     R_NOT_LEAP
; **判斷閏年且為 2 月嗎?
        MOV     A,MONTH
        XOR     A,@2
        JBS     STATUS,Z
        JMP     R_NOT_LEAP
        MOV     A,@29
        MOV     DAY,A
        JMP     DEC_DATE_OK
R_NOT_LEAP:

```