AN008 — 实现一个实时时钟

相关器件

本应用笔记适用于下列器件：

C8051F000、C8051F001、C8051F002、C8051F005、C8051F006、C8051F007、C8051F010、C8051F011 和 C8051F012。

引言

本应用笔记的目的是提供一个如何为 C8051F00x 或 C8051F01x 器件增加实时时钟（RTC）功能的例子。本笔记的最后给出了示例代码。

关键点

- 在系统时钟使用高频率的内部振荡器时，外部振荡器可用于驱动实时时钟所用的晶体。
- 系统时钟可取自内部或外部振荡器，可以切换时钟源而不影响 RTC 的精度。
- RTC 使用定时器 2，该定时器被配置为在一个外部输入的下降沿加 1。
- 比较器 0 用于将外部晶体的波形转变为方波。

概述

在很多嵌入式应用中都要用到实时时钟，例如：实时时钟用于记录一个事件发生的时间、一个压力传感器被激活的时间或读取 ADC 转换值的时间等。目前市场上已有现成的时钟器件，这些时钟器件包含一个小的晶体时间基准和简单的逻辑电路，具有与 I2C、SPI 或微控制器并行口连接的标准接口。本应用笔记介绍如何用一个 C8051Fxxx 器件、一个 32KHz 的钟表晶体和几个无源元件来实现一个廉价的实时时钟功能。

由于这种 RTC 所需的资源和 CPU 开销很小，该功能可以很容易地加到已有的基于 8051 的设计中。

在本例的设计中，一个 32KHz 的钟表晶体接到 C8051 器件的外部振荡器。晶体振荡器的输出信号经过一个内部比较器的调理后作为某一个定时器的输入。在本例中该定时器被配置为自动重装载方式，以十分之一秒为周期循环产生中断。定时器的中断服务程序更新秒、分、时和日的一系列计数器。

硬件说明

图 1 给出了本设计的硬件原理图。本设计使用一个 32KHz 的钟表晶体作为 RTC 的时间基准。该晶体接在 C8051 器件的 XTAL1 和 XTAL2 引脚之间。注意当 CPU 内核使用内部振荡器时，仍然可以允许外部振荡器的晶体驱动器。

Silicon Laboratories Inc.
4635 Boston Lane
Austin, TX 78735
Email: mcuinfo@silabs.com
Internet: www.silabs.com

新华龙电子有限公司
深圳市福田区华强北路现代之窗大厦 A 座 13F C 室(518013)
电话：0755-83645240 83645242 83645244 83645251
电邮：shenzhen@xhl.com.cn
网址：www.xhl.com.cn

XTAL2 上的输出馈送到片内比较器（比较器 0）的（+）输入端。XTAL2 信号经过一个低通滤波后送给比较器的（-）输入端，作为检测振荡信号状态变化的直流偏置电平。该滤波器的转折频率（R=1MΩ，C=0.022 μF）远小于振荡频率。

片内比较器的输出被送到一个外部通用 I/O 引脚（CP0，由交叉开关决定）并连到定时器 2 的输入（T2，也由交叉开关决定）。定时器 2 在每次检测到 T2 输入端产生下降沿时加 1。

定时器 2 被配置为 16 位自动重装载方式，每计 3200 个数或每隔十分之一秒产生一个中断。定时器 2 的中断服务程序更新秒、分、时和日的一系列计数器。

该 RTC 实现的缺省配置是假定 CPU 系统时钟（SYSCLK）取自高速内部振荡器。当系统时钟切换到外部 32KHz 时钟源时（例如为节省功耗），定时器 2 被软件切换到使用 SYSCLK 作为时间基准。RTC 中断服务程序内部的时钟切换同步机制确保不损失精度。

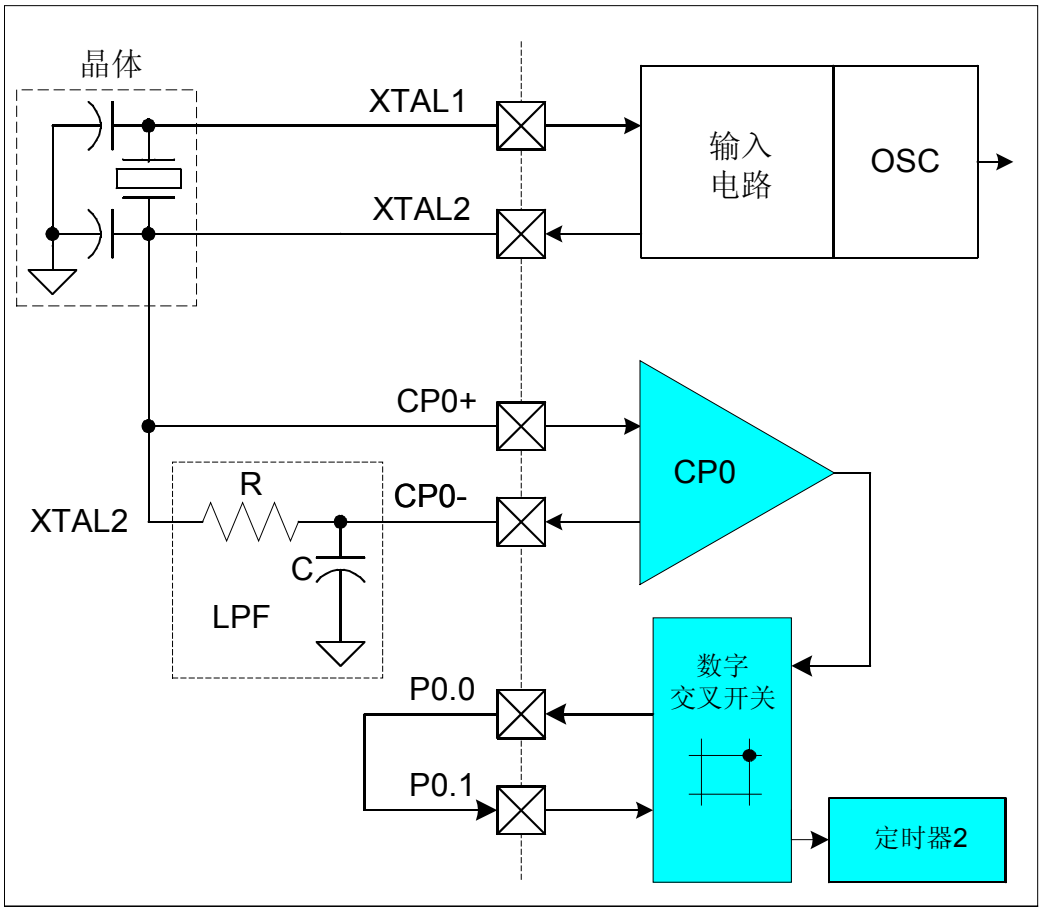

图 1. 连接图

交叉开关配置

内部数字外设与通用 I/O 引脚之间的连接由交叉开关控制。在本设计中，交叉开关将 CP0 输出和 T2 输入分别接到通用 I/O 引脚 P0.0 和 P0.1。需要引起注意的是，如果具有较高交叉开关优先权的外设被允许，则所用的特定端口引脚将发生变化（见 AN001）。交叉开关的设置由下面的程序段完成：

```
; 允许 CP0 输出
mov   XBR0, #80h
; 允许 T2 输入
;mov   XBR1, #20h
; 允许交叉开关和弱上拉
mov   XBR2, #40h
```

振荡器配置

关于配置外部振荡器的详细介绍请参考 AN002。下面的语句配置并允许使用外部 32KHz 晶体的外部振荡器：

```
; 允许外部振荡器工作于‘晶体’方式；
; 对于 32KHz 的晶体，XFCN = 001
mov   OSCXCN, #61h
```

一旦配置完成，必须在允许定时器之前检查外部振荡器是否稳定。当晶体处于稳定运行状态时，XTLVLD 位（OSCXN.7）置 1。在允许定时器 2 之前，用软件查询 XTLVLD 位：

```
; 等待外部振荡器稳定
WAIT:
mov   ACC, OSCXN
jnb     ACC.7, WAIT
; 允许定时器 2
setb  TR2
```

比较器配置

比较器 0 的初始化包含设置正/负回差电压和比较器允许。比较器的回差电压可以在比较器控制寄存器 CPT0CN 中设置。由于 XTAL2 信号电平的范围很宽（500 mV 到 3 V），CP0 的回差电压应设为较高的值以提高抗干扰能力。下面的语句设置回差电压值并允许比较器：

```
; 设置 CP0 回差电压值为 10mV/ 10mV。
mov  CPT0CN, #0Ah
; 允许 CP0
orl   CPT0CN, #80h
```

定时器配置

当 CPU 系统时钟（SYSCLK）取自高频率的内部振荡器时，定时器 2 被设置为自动重装载方式并对外部信号 T2 的下降沿计数。下面的语句用于设置定时器 2：

```
mov    T2CON, #02h
```

我们必须为定时器 2 设置初值和重载值。初值在定时器 2 被允许前装入；重载值保存在 RCAP2H（高字节）和 RCAP2L（低字节）中，在定时器发生溢出后被自动装入。初值和重载值完全相同，由所要求的实时时钟精度决定。本设计实现的精度是十分之一秒，因此定时器 2 被设置为每隔十分之一秒或每计 3200 个数（32KHz 的时间基准）发生一次溢出。我们将 COUNT 值设置为 3200，用下面的语句设置 RCAP2 寄存器中的重载值：

```
; 设置 T2 重载值的高字节
mov    RCAP2H, #HIGH(-COUNT)

; 设置 T2 重载值的低字节
mov    RCAP2L, #LOW(-COUNT)
```

当定时器 2 溢出时，它将被重新装载，以使其在下一次计满 3200 个数时溢出。程序每隔十分之一秒将转向定时器 2 中断服务程序，更新实时时钟所用到的计数器。由于中断服务程序很短并且每隔十分之一秒才调用一次，因此对 CPU 的占用率是很低的。

完成对定时器 2 的配置后，必须用下面的语句允许它的中断：

```
; 允许定时器 2 中断
setb    ET2
```

在所有定时器的其它设置都完成以后，通过设置运行位来允许定时器 2：

```
; 启动定时器 2
setb    TR2
```

系统时钟切换

本 RTC 示例的缺省配置是假定 CPU 系统时钟(SYSCLK)取自高速内部振荡器。如果 SYSCLK 取自外部振荡器（为了节省功耗），则必须将定时器 2 的设置改变为使用 SYSCLK 作为时间基准，因为要使 T2 输入信号被正确地检测到，其最大频率不能超过 SYSCLK/4。

改变系统时钟的操作过程如下：

1. 停止定时器（TR2 = ‘0’）。
2. 改变定时器时间基准。
3. 改变 SYSCLK 时间基准。
4. 加入校正因子到定时器的计数器。
5. 启动定时器（TR2 = ‘1’）。

为了保证不丢失外部时钟边沿，应在 RTC 的中断服务程序中更新 SYSCLK。

可以通过将 SET_EXT_OSC（切换到外部振荡器）或 SET_INT_OSC（切换到内部振荡器）设置为‘1’来切换系统时钟。在定时器 2 的中断服务程序中，这些位被用作标志位，允许改变系统时钟而不牺牲 RTC 的精度。本报告最后的软件说明对此有详细介绍。

软件说明

本节包含软件流程的说明。程序清单从第 7 页开始。

主函数

MAIN 函数用于配置交叉开关、外部振荡器、比较器和定时器。首先我们通过允许外部振荡器和设置功率因子位对外部晶体进行设置。

然后装入上述的交叉开关设置值和 CP0 的设置值，并在装入后允许这两个部件。用软件查询 XTLVLD 位以保证在晶体达到稳定之前不启动定时器 2。当硬件将 XTLVLD 位置‘1’后，程序跳出 WAIT 循环。在 MAIN 函数的最后，RTC_INIT 函数被调用，定时器 2 被允许，中断总允许位被置 1。

RTC 初始化函数

RTC_INIT 函数用于复位计数器值和配置定时器 2。该函数可用于对 RTC 复位。在对计数器的值清 0 后，定时器 2 的初值被设置为在配置一节中所描述的 COUNT 值。该 COUNT 值还被装入到重载寄存器（RCAP2H & RCAP2L）。然后将定时器 2 设置为在外部输入信号的边沿加 1 并允许定时器 2 中断。

定时器中断服务程序

定时器 2 的中断服务程序在每次定时器 2 溢出时被调用（每隔十分之一秒一次）。当中断服务程序被调用时，它首先清除定时器 2 的中断标志（TF2）。然后中断服务程序从十分之一秒计数器开始检查所有计数器的溢出情况。如果十分之一秒计数器当前值为 9，则它被复位到 0，然后检查秒计数器是否溢出。类似地，如果秒计数器当前值为 59，则它被复位到 0，然后检查分计数器。用同样的方式检查时和日计数器。计数器加 1 后，检查振荡器选择位（SET_EXT_OSC 或 SET_INT_OSC）。

振荡器选择

如果 SET_EXT_OSC 位被置 1，则中断服务程序将其清 0 并跳到 EXT_OSC 标号处。首先检查 OSCICN，如果系统时钟已经使用了外部振荡器，则中断服务程序返回。否则禁止定时器 2，以免在切换系统时钟时产生计数错误。按定时器 2 的输入时钟为系统时钟除以 1 来设置 CKCON。然后将定时器 2 设置为对系统时钟加 1 计数，并修改定时器 2 的计数器寄存器以补偿 SYSCLK 切换过程中丢失的计数脉冲。在系统时钟切换与定时器重新启动之间，定时器 2 丢失 5 个计数脉冲。校正值 EXT_COR 被设置为 5，在系统时钟被切换到外部振荡器之前，该值被加到定时器 2 寄存器。完成时钟切换后，定时器 2 被重新启动，中断服务程序返回。

如果 SET_INT_OSC 位被置 1，则中断服务程序将其清 0 并跳到 INT_OSC 标号处。首先检查 OSCICN 保证系统时钟不在使用内部振荡器。如果没有使用内部振荡器，则禁止定时器 2 以便进行时钟切换。选择内部振荡器作为系统时钟，然后将校正值 INT_COR 加到定时器 2 寄存器。在这种情况下，切换时丢失了 3 个计数脉冲。INT_COR 被设置为 3，并被加到定时器 2。选择外部输入引脚作为定时器 2 的输入，重新启动定时器 2。然后中断服务程序返回，等待下一次溢出。

计数器访问

可以调用 SAVE 例程来访问十分之一秒/秒/分/等计数器。SAVE 例程首先将定时器 2 的中断标志保存到进位位，然后禁止定时器 2 中断以保证在保存期间不会产生中断。此处禁止中断并不影响计时，因为在 SAVE 例程的最后又重新允许该中断。如果在 SAVE 例程执行期间产生了一个中断，则该中断将在定时器 2 中断被重新允许后立即得到服务。在 ET2 被清除后，每个计数器都被保存起来（TENTHS 存入 STORE_T，SECONDS 存入 STORE_S，依此类推）。最后恢复中断标志，函数返回到调用程序。

软件举例

```
;-------------------------------------------------------------------------------------------------------
;    CYGNAL INTEGRATED PRODUCTS, INC.

;    文件名                    ：RTC_1.asm
;    目标器件                  ：C8051F0xx
;    说明                      ：用软件实现实时时钟
;    作者                      ：JS

;    软件实现实时时钟，用 32KHz 晶体振荡器。
;    本程序使用晶体驱动器 XTAL2  驱动比较器 0。比较器的正输入端接 XTAL2，
;    负输入端是 XTAL2 的平均值。该平均值是用低通滤波器产生的。
;    比较器 0 的输出接到定时器 2 的输入（T2）。

;    定时器 2 被配置为自动冲装载方式，由连接到比较器 0 输出端的外部输入引脚触发。
;
;    本程序作如下假设：
;    (1)    一个外部振荡器连接在 XTAL1 和 XTAL2 之间
;    (2)    一个低通均值滤波器连接在 XTAL2 和 CP0-之间
;    (3)    XTAL2 连接到 CP0+
;    (4)    CP0 输出通过交叉开关分配的端口引脚连接到定时器 2 输入
;
;    对于 32KHz 的晶体，低通滤波器由一个 0.022uF 的电容和一个 1M 欧姆的电阻构成
;-------------------------------------------------------------------------------------------------------


;-------------------------------------------------------------------------------------------------------
; 等价定义
;-------------------------------------------------------------------------------------------------------

    $MOD8F000

; 计数值：该值用于定义每次溢出后装入到定时器 2 的初值
; 该计数值为 3200，表示定时器每计 3200 个脉冲溢出一次。
; 与 32KHz 的晶体配合使用，意味着定时器每隔十分之一秒溢出一次。

    COUNT       EQU     3200d           ; 计数值

; 系统时钟切换补偿因子，在系统时钟改变后用于更新定时器 2。

    EXT_COR     EQU     5d
    INT_COR     EQU     3d
```

```
;-------------------------------------------------------------------------------------------------------
; 变量
;-------------------------------------------------------------------------------------------------------

DSEG

    org 30h

    TENTHS:         DS      1               ; 十分之一秒计数值
    SECONDS:        DS      1               ; 秒计数值
    MINUTES:        DS      1               ; 分计数值
    HOURS:          DS      1               ; 时计数值
    DAYS:           DS      1               ; 日计数值

    STORE_T:        DS      1               ; 十分之一秒的存储字节，SAVE 程序使用
    STORE_S:        DS      1               ; 秒的存储字节
    STORE_M:        DS      1               ; 分
    STORE_H:        DS      1               ; 时
    STORE_D:        DS      1               ; 日

BSEG

    org     20h

    SET_EXT_OSC:  DBIT      1               ; 将系统时钟切换到外部振荡器的标志
    SET_INT_OSC:  DBIT      1               ; 将系统时钟切换到内部振荡器的标志

;-------------------------------------------------------------------------------------------------------
;复位和中断向量
;-------------------------------------------------------------------------------------------------------

CSEG

; 复位向量

    org     00h
    ljmp    MAIN

; 定时器 2 ISR 向量

    org     2Bh
    ljmp    T2_ISR                          ; 转到定时器 2 ISR
```

```
;-----------------------------------------------------------------------------------------------------------
; MAIN PROGRAM
;-----------------------------------------------------------------------------------------------------------

     org     0B3h

MAIN:

     mov     OSCXCN, #61h                    ; 允许外部振荡器
     mov     WDTCN, #0DEh                    ; 禁止看门狗定时器
     mov     WDTCN, #0Adh

     ; 设置交叉开关
     mov     XBR0, #80h                      ; 允许 CP0 输出
     mov     XBR1, #20h                      ; 允许 T2 输入
     mov     XBR2,       #40h                ; 允许交叉开关

     ; 初始化比较器 0
     mov     CPT0CN, #08h                    ; 设正回差电压为 10mV
     orl     CPT0CN, #02h                    ; 设负回差电压为 10mV
     orl     CPT0CN, #80h                    ; 允许 CP0
     acall   RTC_INIT                        ; 初始化 RTC 和定时器 2

WAIT:
     mov     ACC, OSCXCN                     ; 通过检查 OSCXCN 中的 XTLVLD 位
                                             ; 等待外部振荡器稳定
     jnb     ACC.7, Wait
     setb    TR2                             ; 启动定时器 2（启动 RTC）
     setb    EA                              ; 全局中断允许
     jmp     $                               ; 原地跳转

;-----------------------------------------------------------------------------------------------------------
;     初始化子程序
;-----------------------------------------------------------------------------------------------------------

RTC_INIT:
     ; 清 0 所有计数器
     mov     TENTHS, #0
     mov     SECONDS, #0
     mov     MINUTES, #0
     mov     HOURS, #0
     mov     DAYS, #0
```

```
    ; 初始化定时器 2 为自动重装载方式，对外部输入 T2 的下降沿计数

    mov   TH2, #HIGH(-COUNT)          ; 设置定时器 2 的初值
    mov   TL2, #LOW(-COUNT)

    mov   RCAP2H, #HIGH(-COUNT)       ; 设置定时器 2 的重载值
    mov   RCAP2L,  #LOW(-COUNT)

    mov   T2CON, #02h                 ; 配置定时器 2 对外部输入引脚（T2）的下降沿
                                      ; 加 1 计数
    setb  ET2                         ; 允许定时器 2 中断

    ret

;-------------------------------------------------------------------------------------------------------------------
; 定时器 2 ISR
;-------------------------------------------------------------------------------------------------------------------

T2_ISR:

    clr   TF2                         ; 清除溢出中断标志
    push  PSW                         ; 保存 PSW（进位标志）
    push  ACC                         ; 保存 ACC

    ; 检查溢出

    mov   A, TENTHS
    cjne  A, #9d, INC_TEN             ; 如果十分之一秒计数值小于 9，转去加 1
    mov   TENTHS, #0                  ; 如果十分之一秒计数值 = 9，清 0 后检查秒
    mov   A, SECONDS
    cjne  A, #59d, INC_SEC            ; 如果秒计数值小于 59，转去加 1
    mov   SECONDS, #0                 ; 如果秒计数值 = 59，清 0 后检查分
    mov   A, MINUTES
    cjne  A, #59d, INC_MIN            ; 如果分计数值小于 59，转去加 1
    mov   MINUTES, #0                 ; 如果分计数值 = 59，清 0 后检查小时
    mov   A, HOURS
    cjne  A, #23d, INC_HOUR           ; 如果时计数值小于 23，转去加 1
    mov   HOURS, #0                   ; 如果时计数值 = 23 清 0 后检查日计数值
    inc   DAYS                        ; DAYS 在计到 255 后归 0
    jmp   CHECK_OSC                   ; 转去检查振荡器的切换请求
```

```
;计数器加 1 ----------------------------------------------------------------------------------------------------

INC_TEN:
    inc     TENTHS                  ; 十分之一秒计数器加 1
    jmp     CHECK_OSC               ; 转去检查振荡器切换请求

INC_SEC:
    inc     SECONDS                 ; 秒计数器加 1
    jmp     CHECK_OSC               ; 转去检查振荡器切换请求

INC_MIN:
    inc     MINUTES                 ; 分计数器加 1
    jmp     CHECK_OSC               ; 转去检查振荡器切换请求

INC_HOUR:
    inc     HOURS                   ; 时计数器加 1
    jmp     CHECK_OSC               ; 转去检查振荡器切换请求

;振荡器切换 ----------------------------------------------------------------------------------------------

CHECK_OSC:
    jbc     SET_EXT_OSC, EXT_OSC    ; 检查外部振荡器选择
    jbc     SET_INT_OSC, INT_OSC    ; 检查内部振荡器选择
    jmp     END_ISR                 ; 退出 ISR


EXT_OSC:                            ; 系统时钟切换到外部振荡器
    mov     ACC, OSCICN             ; 检查当前系统时钟
    jb        ACC.3, END_ISR        ; 如果已经使用外部振荡器则返回

    orl     CKCON, #20h             ; 选择系统时钟为定时器 2 时钟源（系统时钟/1）

    clr     T2CON.2                 ; 在时钟切换期间禁止定时器 2
    clr     T2CON.1                 ; 选择 SYSCLK 作为定时器 2 输入

    mov     A, #LOW(EXT_COR)        ; 将校正值装入到累加器
    add     A, TL2                  ; 将校正值加到定时器 2 寄存器
    mov     TL2, A                  ; 保存更新后的定时器 2 初值

    orl     OSCICN, #08h            ; 设置外部振荡器为系统时钟
    setb    T2CON.2                 ; 完成时钟切换后允许定时器 2
    jmp     END_ISR                 ; 中断返回
```

```
INT_OSC:                                ; 将系统时钟切换到内部振荡器

    mov   ACC, OSCICN                   ; 检查当前系统时钟
    jnb   ACC.3, END_ISR                ; 如果已经使用内部振荡器则返回

    clr   T2CON.2                       ; 在时钟切换期间禁止定时器 2
    anl   OSCICN, #0f7h                 ; 设置内部振荡器为系统时钟

    mov   A, #LOW(INT_COR)              ; 将校正值装入到累加器
    add   A, TL2                        ; 将校正值加到定时器 2 寄存器
    mov   TL2, A                        ; 保存更新后的定时器 2 初值

    setb  T2CON.1                       ; 选择外部定时器 2 输入
    setb  T2CON.2                       ; 完成时钟切换后允许定时器 2

    jmp   END_ISR                       ; 中断返回

END_ISR:

    pop   ACC                           ; 恢复 ACC
    pop   PSW                           ; 恢复 PSW
    reti

;-------------------------------------------------------------------------------------------------------------
; 计数器保存子程序
;-------------------------------------------------------------------------------------------------------------

SAVE:

    mov   C, ET2                        ; 将 ET2 保存到进位标志

    clr   ET2                           ; 保存期间禁止定时器 2

    mov   STORE_T, TENTHS               ; 复制所有计数器
    mov   STORE_S, SECONDS
    mov   STORE_M, MINUTES
    mov   STORE_H, HOURS
    mov   STORE_D, DAYS
    mov   ET2, C                        ; 恢复 ET2
    ret

;-----------------------------------------------------------------------------

END
```