

Cute51 单片机实验板电子表仿真程序

作者：合肥工业大学 杨铸

网址：www.200yi.com 电邮：fairyfar@msn.com

2006.12 — 2007.01

Scripted by FairyFar.

1. 关于本文

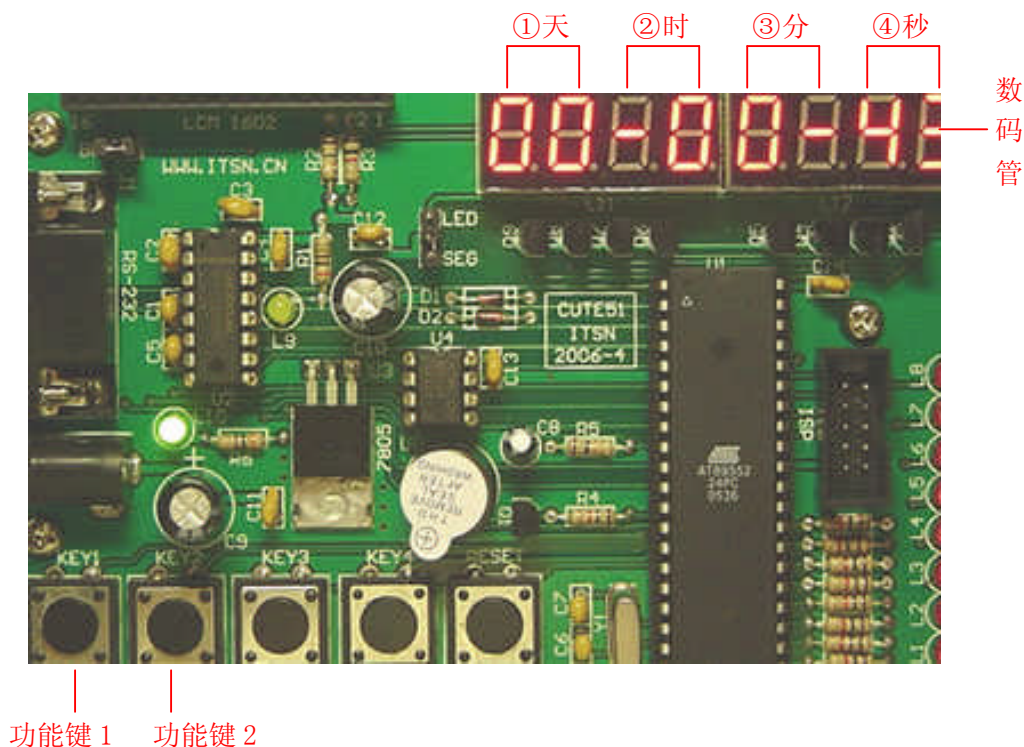
本文是关于 ATMEL 公司 AT89S52 单片机的应用实例的说明文档，Cute51 单片机实验板由中国信息技术资源网提供。

您可以无偿使用附件源代码，但出于对原作者劳动的尊重，请保留版权声明区之内容。

2. 开发平台

2.1 Cute51 单片机实验板

先给出 Cute51 单片机实验板的实体图及本案例对输入输出单元的定义：



2.2 案例实现的功能

1、走时状态下（即显示当前时间），数码管显示当前日期和时间，包括：天、小时、分、秒。

2、时间设定。在走时状态下，第 1 次按“功能键 1”则④闪烁，表示当前是设置走时秒，此时每按“功能键 2”一次则走时秒加一；第 2 次按“功能键 1”则③闪烁，表示当前是设置走时分，此时每按“功能键 2”一次则走时分加一；第 3 次按“功能键 1”则②闪烁，表示当前是设置走时小时，此时每按“功能键 2”一次则走时小时（24 小时制）加一；第 4 次按“功能键 1”则①闪烁，表示当前是设置走时天，此时每按“功能键 2”一次则走时天

加一；第 5 次按“功能键 1”则数码管显示“beep on”或“beep off”，表示当前是设定是否每到整点（7:00~23:00）响铃一下，此时每按“功能键 2”一次则数码管显示内容翻转一次；第 6 次按“功能键 1”则退出到走时状态。

3、闹铃功能。在走时状态下，第 1 次按“功能键 2”则④闪烁，表示当前是设置闹铃持续分钟数（1~59 对应闹铃持续时间 1~59 分，0 表示不闹铃。），此时每按“功能键 1”一次则闹铃持续分钟数加一；第 2 次按“功能键 2”则③闪烁，表示当前是设置闹铃分，此时每按“功能键 1”一次则闹铃分钟加一；第 3 次按“功能键 2”则②闪烁，表示当前是设置闹铃小时，此时每按“功能键 1”一次则闹铃小时加一；第 4 次按“功能键 2”则①闪烁，表示当前是设置是否闹铃每天都响（0 表示每天都闹铃，1~31 表示只在每月的该天闹铃。如果闹铃持续分钟数在之前已经被设置为 0，则此处设置任何值都不闹铃。），此时每按“功能键 1”一次则闹铃天加一；第 5 次按“功能键 2”则退出到走时状态。

4、在闹铃响铃的过程中，可以按“功能键 1”或“功能键 2”终止响铃。

2.3 Keil μ Vision3

Keil μ Vision3 是由 Keil 公司开发的 51 内核单片机软件集成开发环境。本案例程序在 Keil μ Vision3 中调试通过。并通过 ISP 下载软件加载到 Cute51 单片机实验板。

2.4 代码导读

源代码已经给出了较为详细的注释，所以我们不准备详细介绍源代码。只是提出几个需要注意的地方：

1、功能键操作逻辑关系；

2、闹铃“零点问题”的解决，假设闹铃时间设定为 23:50，闹铃持续时间设为 20 分钟，那么这里就有一个过零点的判断问题；

3、引入走时时间补偿系数，是为了消除走时的积累误差。测试表明（与国家天文台授时中心校准），通过加入时间补偿系数，该程序能够达到精度为： ± 1 秒/周。

3. 附件：汇编源代码

```

;***** 版 权 声 明 *****
;程序名称：电子表仿真程序。
;版权申明：(S) Scripted By FairyFar. ChuZhou • China.
;作者网站：http://www.200yi.com
;开源程序，但出于对原作者劳动的尊重，请保留版权声明区之内容。
;***** 版 权 声 明 *****

;*****
;实用环境：使用于 Cute51 单片机实验板（AT89S52）。
;开发时间：2006.12—2007.01
;功能说明：
;①、日期与时间显示与设定；
;②、可指定开始及持续时间的闹铃功能；
;③、闹铃时，可被随时终止；
;④、可设定的整点响铃功能（7:00—23:00）。
;*****
;变量定义区
;R0，确定延时长度。
;R3、R4，控制数码管闪烁时间系数。
;R5，数码管点灭标志，R5=0 则当前灭（用于控制闪烁功能）。
;R6，存放功能选择码。
MySecond EQU 30H ;秒
MyMinute EQU 31H ;分
MyHour EQU 32H ;小时
MyDay EQU 33H ;天
MyWeek EQU 34H ;周（预留）

```

```

EnBeep      EQU      35H      ;允许整点响铃标志位
CLK_fun     EQU      36H      ;正在设置闹钟功能选择码(为0,则表示当前不是在进行闹铃设置。)

HowMinute   EQU      37H      ;闹铃持续分钟数
CLKMinute   EQU      38H      ;闹铃开始分
CLKHour     EQU      39H      ;闹铃开始小时
CLKDay      EQU      3AH      ;闹铃开始天
CLKWeek     EQU      3BH      ;闹铃开始周(预留)

EndSecond   EQU      3CH      ;(预留)
EndMinute   EQU      3DH      ;闹铃截止分
EndHour     EQU      3EH      ;闹铃截止小时
EndDay      EQU      3FH      ;闹铃截止天
EndWeek     EQU      40H      ;闹铃截止周(预留)
Bell_Flag   EQU      41H      ;通知外部,1表示现在闹铃正在响铃。
Bell_int    EQU      42H      ;0表示闹铃被临时关闭

Loop_R9     EQU      43H      ;响铃时,用于延时时内循环。
Loop_R10    EQU      44H      ;响铃时,用于延时时外循环。
;*****

;*****
ORG         0000H
AJMP       MAIN
ORG         0003H
LJMP      PLUSVAL      ;转到 INTO(KEY2,设置数值键)中断服务程序
ORG         0013H
LJMP      SETING       ;转到 INT1(按键1,功能选择键)中断服务程序
ORG         001BH
LJMP      SERVE        ;转到 T1 中断服务程序,2。

;主程序
MAIN:      MOV      IP,#00001000B      ;T1具有高优先级
          MOV      TMOD,#10H          ;设置 T0 工作于模式 1, T1 模式 1。
          MOV      20H,#20D          ;中断次数
          CLR      A
          MOV      MySecond,A          ;秒、分、小时、天初始化。
          MOV      MyMinute,A
          MOV      MyHour,#08D
          MOV      MyDay,#01H
          MOV      EnBeep,#01H        ;允许整点响铃!

          MOV      CLK_fun,A          ;正在设置闹钟功能选择码
          MOV      HowMinute,#05D     ;闹铃持续分钟数
          MOV      CLKMinute,#30D     ;分
          MOV      CLKHour,#07D       ;小时
          MOV      CLKDay,A           ;天

          MOV      EndMinute,#35D     ;闹铃截止时间初始化
          MOV      EndHour,#07D
          MOV      EndDay,A

          MOV      Bell_int,#01H      ;0表示闹铃被临时关闭
          MOV      Bell_Flag,#00H;1则通知外部,现在闹铃正在响铃。

          SETB     ET1                ;允许 T1 中断
          SETB     EX0                ;允许 INTO 中断
          SETB     EX1                ;允许 INT1 中断
          SETB     EA                ;允许 CPU 中断
          MOV      TH1,#3CH
          MOV      TL1,#0B0H
          SETB     TR1                ;启动定时器 T1
          MOV      R6,#00H            ;功能选择码初始值
          MOV      R5,#00H            ;灭灯
;*****

;显示程序
DISP:     MOV      A,R6
          JNZ     TABLEJ30
          MOV     A,CLK_fun

```

```

JNZ     TABLEJ30      ;当前正在设置时间或闹铃，则不进行闹铃判断。

MOV     A, EndMinute
CLR     CY
SUBB   A, CLKMinute
JZ      TABLEJ31      ;CLKMinute=EndMinute，则不闹铃。
MOV     A, CLKDay
JZ      Judge_Hour     ;CLKDay=0时，每天都判断是否闹铃。

MOV     A, EndDay
CLR     CY
SUBB   A, CLKDay
JNZ     J_MeC_Day      ;EndDay<>CLKDay，则转到 MyDay 与 CLKDay 比较。

MOV     A, MyDay
CLR     CY
SUBB   A, CLKDay
JZ      Judge_Hour     ;CLKDay=MyDay，把判断依据扔给小时。
LJMP   NotBell

;*****
;跳转表（解决长转移问题，下同。）
TABLEJ30: LJMP B_DISP
;*****
J_MeC_Day:  MOV     A, MyDay
            CLR     CY
            SUBB   A, CLKDay
            JNZ     JJ0001      ;MyDay<>CLKDay，则转到 EndDay 与 MyDay 比较。

            MOV     A, MyHour
            CLR     CY
            SUBB   A, CLKHour
            JZ      JJ0002      ;MyDay=CLKDay，且 MyHour=CLKHour，则比较 MyMinute 与
            CLKMinute.
            LJMP   NotBell

JJ0002:    MOV     A, MyMinute
            CLR     CY
            SUBB   A, CLKMinute
            JC      TABLEJ31
            LJMP   BellStart

JJ0001:    MOV     A, MyDay
            CLR     CY
            SUBB   A, EndDay
            JNZ     TABLEJ31  ;MyDay<>CLKDay，且 MyDay<>EndDay，则不响铃。

            MOV     A, MyHour
            CLR     CY
            SUBB   A, EndHour
            JZ      JJ0003      ;MyHour=EndHour，则比较 MyMinte 和 EndMinute。
            LJMP   NotBell

JJ0003:    MOV     A, MyMinute
            SUBB   A, EndMinute
            JC      BellStart
            LJMP   NotBell

;*****
;小时算法
Judge_Hour: MOV     A, EndHour      ;算法同“天”
            CLR     CY
            SUBB   A, CLKHour
            JNZ     J_MeC_Hour     ;EndHour<>CLKHour，则转到 MyHour 与 CLKHour 比较。
            MOV     A, MyHour
            CLR     CY
            SUBB   A, CLKHour
            JZ      Judge_Minute   ;把判断依据扔给分
            LJMP   NotBell

```

```

J_MeC_Hour:  MOV     A, MyHour
              CLR     CY
              SUBB    A, CLKHour
              JNZ     JJ0004          ;MyHour<>CLKHour, 则转到 MyHour 与 EndHour 比较。

              MOV     A, MyMinute    ;MyHour=CLKHour, 则进行 MyMinute 与 CLKMinute 比较。
              CLR     CY
              SUBB    A, CLKMinute
              JC      NotBell
              LJMP    BellStart
;*****
;跳转表
TABLEJ0:  LJMP  Disp_Clock
TABLEJ31: LJMP  NotBell
TABLEJ33: LJMP  DISP
;*****
JJ0004:    MOV     A, EndHour
              CLR     CY
              SUBB    A, MyHour
              JNZ     NotBell        ;MyHour<>EndHour, 则不响铃。

              MOV     A, MyMinute    ;MyHour=EndHour, 则进行 MyMinute 与 EndMinute 比较。
              SUBB    A, EndMinute
              JC      BellStart
              LJMP    NotBell
;*****
;分算法
Judge_Minute: MOV    A, MyMinute
              CLR     CY
              SUBB    A, CLKMinute
              JC      NotBell

              MOV     A, EndMinute
              CLR     CY
              SUBB    A, MyMinute
              JC      NotBell
              JZ      NotBell

BellStart:  MOV     A, Bell_int
              JZ      B_DISP
              MOV     Bell_Flag, #01H

              CLR     P3.4           ;开
              MOV     R0, #100D
              CALL    DELYxM
              SETB    P3.4           ;关
              MOV     R0, #100D
              CALL    DELYxM
              CLR     P3.4           ;开
              MOV     R0, #100D
              CALL    DELYxM
              SETB    P3.4           ;关
              JMP     B_DISP

NotBell:    MOV     Bell_int, #01H
              MOV     Bell_Flag, #00H

;*****
;闹铃之后的显示程序
B_DISP:     MOV     A, Bell_Flag      ;判断闹铃是否正在响
              JZ      Loop_R11
              MOV     A, Bell_int
              JZ      Loop_R11
              MOV     A, R6
              JNZ     Loop_R11

Loop_R12:   MOV     Loop_R10, #15D    ;闹铃时延时的外循环
              MOV     Loop_R9, #250D  ;闹铃时延时的内循环

```

```

Loop_R11:   MOV     A, CLK_fun
            JNZ     TABLEJ0           ;不为零表示当前正在进行闹铃设置, 则转移到显示闹铃时
间程序。

            MOV     A, CLK_fun
            JNZ     TABLEJ33         ;再次判断(重要!), 只要 CLK_fun<>0 程序就不向下走。

            CJNE    R6, #05H, Temp_1  ;如果 R6<>05H, 则程序继续推进。
            LJMP    SetEnBeep         ;如果 R6=05H, 则正在进行整点响铃设置, 转移到显示 BEEP

On/OFF。
Temp_1:     MOV     A, R6
            JZ      FlashSe           ;如果 R6=00H, 则不闪烁。

            MOV     R3, #8D
            MOV     R4, #250D         ;闪烁时间系数

LOOPR4:     CJNE    R6, #01H, FlashSe  ;不是设置秒, 则秒管不闪!
FlashLED:   CJNE    R5, #00H, FlashMi  ;当前数码管是灭还是亮?
FlashSe:    MOV     A, MySecond
            MOV     B, #10D           ;显示秒
            DIV    AB                 ;累加器 A 中内容除以 B 中内容, 商放到 A 中, 余数送到 B
中。

            MOV     DPTR, #TABLE
            MOVC    A, @A+DPTR
            MOV     P0, A
            CLR     P2.1
            NOP
            NOP
            SETB   P2.1
            MOV     A, B
            MOVC    A, @A+DPTR
            MOV     P0, A
            CLR     P2.0
            NOP
            NOP
            SETB   P2.0

FlashMi:    CJNE    R6, #02H, FlashMi  ;不是设置分, 则分管不闪!
            CJNE    R5, #00H, FlashHo
            MOV     A, MyMinute       ;显示分
            MOV     B, #10D
            DIV    AB
            MOV     DPTR, #TABLE
            MOVC    A, @A+DPTR
            MOV     P0, A
            CLR     P2.3
            NOP
            NOP
            SETB   P2.3
            MOV     A, B
            MOVC    A, @A+DPTR
            MOV     P0, A
            CLR     P2.2
            NOP
            NOP
            SETB   P2.2

FlashHo:    CJNE    R6, #03H, FlashHo  ;不是设置小时, 则小时管不闪!
            CJNE    R5, #00H, FlashDa
            MOV     A, MyHour         ;显示小时
            MOV     B, #10D
            DIV    AB
            MOV     DPTR, #TABLE
            MOVC    A, @A+DPTR
            MOV     P0, A
            CLR     P2.5
            NOP
            NOP
            SETB   P2.5
            MOV     A, B
            MOVC    A, @A+DPTR
    
```

```

MOV      P0, A
CLR      P2.4
NOP
NOP
SETB     P2.4

FlashDa: CJNE     R6, #04H, FlashDa ;不是设置天, 则天管不闪!
          CJNE     R5, #00H, FlashEnd
          MOV      A, MyDay ;显示天
          MOV      B, #10D
          DIV      AB
          MOV      DPTR, #TABLE
          MOVC     A, @A+DPTR
          MOV      P0, A
          CLR      P2.7
          NOP
          NOP
          SETB     P2.7
          MOV      A, B
          MOVC     A, @A+DPTR
          MOV      P0, A
          CLR      P2.6
          NOP
          NOP
          SETB     P2.6

FlashEnd: MOV      A, Bell_Flag ;判断闹铃是否正在响
          JZ       BeepDudge
          MOV      A, Bell_int
          JZ       BeepDudge
          MOV      A, R6
          JNZ      BeepDudge

          DJNZ     Loop_R9, TABLEJ41
          DJNZ     Loop_R10, TABLEJ42

;整点响铃程序 (闪烁结束)
BeepDudge: MOV      A, MyHour ;07~23 点允许响铃
          CLR      CY
          SUBB     A, #07D
          JC       UnBell

          MOV      A, MyMinute ;每逢整点响铃
          CJNE     A, #00D, UnBell ;秒和分同时为 0, 才响铃。
          MOV      A, MySecond
          CJNE     A, #00D, UnBell
          MOV      A, EnBeep ;允许响铃吗?
          JZ       UnBell

          CLR      P3.4 ;启动蜂鸣器
          MOV      R0, #250D ;延时 250ms
          CALL     DELYxM
          SETB     P3.4 ;关闭蜂鸣器

UnBell:   MOV      A, R6
          JZ       TABLEJ1 ;如果 R6=00H, 则不闪烁。到跳转表去。

          CJNE     R6, #05H, Temp_2
          LJMP     SetEnBeep ;(重要) 如果 R6=05H, 则正在进行整点响铃设置, 转移
到显示 BEEP On/OFF。

Temp_2:   DJNZ     R4, TABLEJ2 ;到跳转表去
          DJNZ     R3, TABLEJ3 ;到跳转表去

SetR5:    CJNE     R5, #00H, SetR5 ;对灭灯进行设置
          MOV      R5, #01H
          LJMP     DISP
          MOV      R5, #00H
          LJMP     DISP

```

```

;*****
;跳转表
TABLEJ1: LJMP DISP
TABLEJ2: LJMP FlashLED
TABLEJ3: LJMP LOOPR4
TABLEJ41: LJMP Loop_R11
TABLEJ42: LJMP Loop_R12
;*****
;是否正在进行整点响铃设置
SetEnBeep:  MOV     A, EnBeep
            JNZ     DISPOn
            MOV     P0, #10001110B      ;F。显示 OFF
            CLR     P2.1
            SETB    P2.1

            MOV     P0, #10001110B      ;F
            CLR     P2.0
            SETB    P2.0

            MOV     P0, #11000000B      ;0
            CLR     P2.2
            SETB    P2.2
            JMP     DISPbeep

DISPOn:     MOV     P0, #11000000B      ;0。显示 On
            CLR     P2.1
            SETB    P2.1

            MOV     P0, #11001000B      ;n
            CLR     P2.0
            SETB    P2.0

DISPbeep:   MOV     P0, #10001100B      ;BEEP
            CLR     P2.4
            SETB    P2.4

            MOV     P0, #10000110B
            CLR     P2.5
            SETB    P2.5

            MOV     P0, #10000110B
            CLR     P2.6
            SETB    P2.6

            MOV     P0, #10000000B
            CLR     P2.7
            SETB    P2.7
            LJMP    DISP
;*****
Disp_Clock: MOV     R3, #8D
CLKLOOPR4:  MOV     R4, #250D           ;闪烁时间系数
CLKFlashLED: MOV    A, CLK_fun
            CJNE    A, #01H, CLKFlashEn ;不是设置闹铃持续时间, 则秒管不闪!
            CJNE    R5, #00H, CLKFlashMi ;当前数码管是灭还是亮?
CLKFlashEn: MOV     A, HowMinute      ;显示闹铃持续分钟数
            MOV     B, #10D
            DIV     AB                 ;累加器 A 中内容除以 B 中内容, 商放到 A 中, 余数
送到 B 中。
            MOV     DPTR, #TABLE
            MOVC    A, @A+DPTR
            MOV     P0, A
            CLR     P2.1
            NOP
            NOP
            SETB    P2.1
            MOV     A, B
            MOV     DPTR, #TABLED
            MOVC    A, @A+DPTR
            MOV     P0, A

```



```

                CLR      P2. 0
                NOP
                NOP
                SETB     P2. 0

                MOV      A, CLK_fun           ;显示闹铃分
                CJNE     A, #02H, CLKFlashMi  ;不是设置闹铃分，则分管不闪！
                CJNE     R5, #00H, CLKFlashHo

CLKFlashMi:    MOV      A, CLKMinute         ;显示闹铃分
                MOV      B, #10D
                DIV      AB
                MOV      DPTR, #TABLE
                MOVC     A, @A+DPTR
                MOV      P0, A
                CLR      P2. 3
                NOP
                NOP
                SETB     P2. 3
                MOV      A, B
                MOV      DPTR, #TABLED
                MOVC     A, @A+DPTR
                MOV      P0, A
                CLR      P2. 2
                NOP
                NOP
                SETB     P2. 2

                MOV      A, CLK_fun           ;显示闹铃小时
                CJNE     A, #03H, CLKFlashHo  ;不是设置闹铃小时，则小时管不闪！
                CJNE     R5, #00H, CLKFlashDa

CLKFlashHo:    MOV      A, CLKHour          ;显示闹铃小时
                MOV      B, #10D
                DIV      AB
                MOV      DPTR, #TABLE
                MOVC     A, @A+DPTR
                MOV      P0, A
                CLR      P2. 5
                NOP
                NOP
                SETB     P2. 5
                MOV      A, B
                MOV      DPTR, #TABLED
                MOVC     A, @A+DPTR
                MOV      P0, A
                CLR      P2. 4
                NOP
                NOP
                SETB     P2. 4

                MOV      A, CLK_fun           ;显示闹铃天
                CJNE     A, #04H, CLKFlashDa  ;不是设置闹铃天，则天管不闪！
                CJNE     R5, #00H, CLKFlashEnd

CLKFlashDa:    MOV      A, CLKDay           ;显示天
                MOV      B, #10D
                DIV      AB
                MOV      DPTR, #TABLE
                MOVC     A, @A+DPTR
                MOV      P0, A
                CLR      P2. 7
                NOP
                NOP
                SETB     P2. 7
                MOV      A, B
                MOV      DPTR, #TABLED
                MOVC     A, @A+DPTR
                MOV      P0, A
                CLR      P2. 6
                NOP
                NOP
    
```

```

                SETB    P2.6

CLKFlashEnd:  DJNZ    R4, TABLEJ10      ;到跳转表去
                DJNZ    R3, TABLEJ11      ;到跳转表去

                MOV     R3, #8D
                MOV     R4, #250D          ;重设闪烁时间系数（冗余代码，但很有效!）

                CJNE    R5, #00H, CLKSetR5 ;对灭灯进行设置
                MOV     R5, #01H
                LJMP    DISP
CLKSetR5:     MOV     R5, #00H
                LJMP    DISP

;*****
;跳转表
TABLEJ10: LJMP CLKFlashLED
TABLEJ11: LJMP CLKLOOPR4
;*****
;T1 中断服务程序区
SERVE:        PUSH    PSW                ;2 个周期指令，下同。
                PUSH    ACC                ;2
                MOV     TH1, #3CH          ;重新赋初值。1
                MOV     TL1, #(0B0H+10D)  ;（主要时间补偿）加了时间补偿。1。
                DJNZ    20H, RETUNT        ;2
                MOV     20H, #20D         ;2

                INC     MySecond          ;秒加 1
                MOV     A, MySecond
                CJNE    A, #60D, RETUNT
                MOV     MySecond, #00H

                INC     MyMinute          ;分加 1
                MOV     A, MyMinute
                CJNE    A, #60D, RETUNT
                MOV     MyMinute, #00H

                CLR     ET1                ;禁止 T1 中断
                MOV     R0, #200D          ;（次要时间补偿）每小时扣除 200ms，这是一个计算与实
验综合得到的数值。
                CALL    DELYxM
                SETB    ET1                ;允许 T1 中断

                INC     MyHour             ;小时加 1
                MOV     A, MyHour
                CJNE    A, #24D, RETUNT
                MOV     MyHour, #00H

                INC     MyDay              ;天加 1
                MOV     A, MyDay
                CJNE    A, #32D, RETUNT
                MOV     MyDay, #01H
RETUNT:       POP     ACC                  ;2
                POP     PSW                ;2
                RETI                       ;2
;*****
;加 1 键中断服务（INT0, KEY2）（也作为闹钟设置选择键）
PLUSVAL:     PUSH    PSW
                PUSH    ACC
                CLR     EX0                ;关中断
                MOV     R0, #200D          ;延时 200ms
                CALL    DELYxM

;重要程序区*****
                MOV     A, Bell_Flag      ;只有当前闹铃正在响铃时才允许中断之！
                JZ      Temp_01
                MOV     A, Bell_int      ;当前是闹铃时间，且闹铃被用户临时关闭，则允许功能键。
                JZ      Temp_01
                MOV     Bell_int, #00H    ;临时关闭闹铃

```

```

                LJMP    RETUNP
;重要程序区*****
Temp_01:      MOV     A, R6
              JZ      CLKfunSet      ;R6=00H时才允许准备设置闹铃功能

              CJNE   R6, #01H, SetMi  ;秒
              INC    MySecond
              MOV    A, MySecond
              CJNE   A, #60D, TABLEJ20
              MOV    MySecond, 00H

SetMi:        CJNE   R6, #02H, SetHo  ;分
              INC    MyMinute
              MOV    A, MyMinute
              CJNE   A, #60D, TABLEJ20
              MOV    MyMinute, 00H

SetHo:        CJNE   R6, #03H, SetDa  ;小时
              INC    MyHour
              MOV    A, MyHour
              CJNE   A, #24D, RETUNP
              MOV    MyHour, 00H

SetDa:        CJNE   R6, #04H, SetEBe ;天
              INC    MyDay
              MOV    A, MyDay
              CJNE   A, #32D, RETUNP
              MOV    MyDay, #01H

SetEBe:       CJNE   R6, #05H, RETUNP ;设置是否整点响铃
              MOV    A, EnBeep
              JZ     SETBeep
              MOV    EnBeep, #00H    ;禁止响铃
              JMP    RETUNP

SETBeep:      MOV    EnBeep, #01H    ;允许响铃
              JMP    RETUNP

;设置闹铃功能选择码
CLKfunSet:    INC    CLK_fun
              MOV    A, CLK_fun      ;设置闹钟功能选择码加1
              CJNE   A, #05H, CalEndTime
              MOV    CLK_fun, 00H
              JMP    CalEndTime

;*****
;跳转表
TABLEJ20: LJMP RETUNP
;*****
;由闹铃开始时间计算→闹铃截止时间
CalEndTime:   MOV    Bell_int, #01H

              MOV    A, HowMinute    ;计算闹铃截止分
              ADD    A, CLKMinute
              MOV    EndMinute, A
              MOV    A, #59D
              CLR    CY
              SUBB  A, EndMinute
              JNC   CalEndHo         ;分未溢出，则直接转移到设置小时。CY=0
              MOV    A, EndMinute
              CLR    CY
              SUBB  A, #60D
              MOV    EndMinute, A
              SETB  CY               ;分溢出标志。CY=1

CalEndHo:     MOV    A, CLKHour      ;计算闹铃截止小时
              ADDC  A, #00H
              MOV    EndHour, A
              MOV    A, #23D
              CLR    CY

```

```

SUBB    A, EndHour
JNC     CalEndDa      ;小时未溢出，则直接转移到设置天。CY=0
MOV     EndHour, #00H
SETB    CY            ;小时溢出标志。CY=1

CalEndDa:  MOV     A, CLKDay
           MOV     EndDay, A
           JZ      RETUNP      ;天为 0 则直接退出
           ADDC   A, #00H
           MOV     EndDay, A
           MOV     A, #31D
           CLR     CY
           SUBB   A, EndDay
           JNC     RETUNP      ;天未溢出，则直接退出。
           MOV     EndDay, #01H

RETUNP:   SETB    EX0
           POP     ACC
           POP     PSW
           RETI

;*****
;功能选择键中断服务 (INT1, KEY1) (也用于设置闹铃时间)
SETING:   PUSH    PSW
           PUSH    ACC
           CLR     EX1        ;关中断
           MOV     R0, #200D   ;延时 200ms
           CALL   DELYxM

           MOV     A, CLK_fun   ;CLK_fun<>0 表示当前正在进行闹铃设置，转移。
           JNZ    CLK_Seting

           INC     R6            ;功能码加 1
           CJNE   R6, #06H, RETUNS
           MOV     R6, 00H
           MOV     Bell_int, #01H ;设置时间后自动将闹铃临时关闭取消 (人性化设计)
           JMP     RETUNS

CLK_Seting: MOV    A, CLK_fun
            CJNE   A, #01H, SetCLKMi ;允许闹铃标志位 (占用秒位)
            INC    HowMinute        ;HowMinute 为零则禁止闹铃，否则其存放的是闹铃持续分
            钟数。

            MOV    A, HowMinute
            CJNE   A, #60D, RETUNS
            MOV    HowMinute, 00H

SetCLKMi:  MOV    A, CLK_fun
            CJNE   A, #02H, SetCLKHo ;闹铃分钟加 1
            INC    CLKMinute
            MOV    A, CLKMinute
            CJNE   A, #60D, RETUNS
            MOV    CLKMinute, 00H

SetCLKHo:  MOV    A, CLK_fun
            CJNE   A, #03H, SetCLKDa ;闹铃小时加 1
            INC    CLKHour
            MOV    A, CLKHour
            CJNE   A, #24D, RETUNS
            MOV    CLKHour, 00H

SetCLKDa:  MOV    A, CLK_fun
            CJNE   A, #04H, RETUNS   ;闹铃天加 1 (天如果为 0，则表示闹铃每天有效，否则只
            1 天有效。)

            INC    CLKDay
            MOV    A, CLKDay
            CJNE   A, #32D, RETUNS
            MOV    CLKDay, 00H

RETUNS:    SETB    EX1
            POP     ACC

```

```

                POP     PSW
                RETI
;*****
;延时 x ms 通用程序 (x 由 R0 传入)
DELYxM:        MOV     R1, #250
LOOP:          NOP
                NOP
                DJNZ  R1, LOOP
                DJNZ  R0, DELYxM
                RET
;*****
;定义 8 段码
TABLED:        DB      40H, 79H, 24H, 30H, 19H, 12H, 02H, 78H, 00H, 10H
TABLE:         DB      0c0H, 0f9H, 0a4H, 0b0H, 99H, 92H, 82H, 0f8H, 80H, 90H
;              '0'  '1'  '2'  '3'  '4'  '5'  '6'  '7'  '8'  '9'
                END
    
```