

自动打铃器

摘要 文章介绍了一种以 8031 单片机为核心片外扩展 at93c46 电擦除可编程只读存储器并带有数码 LED 显示时、分、秒和以指示灯做标志的按键校时自动打铃器。

关键词 单片机 存储器 数码管

一、引言

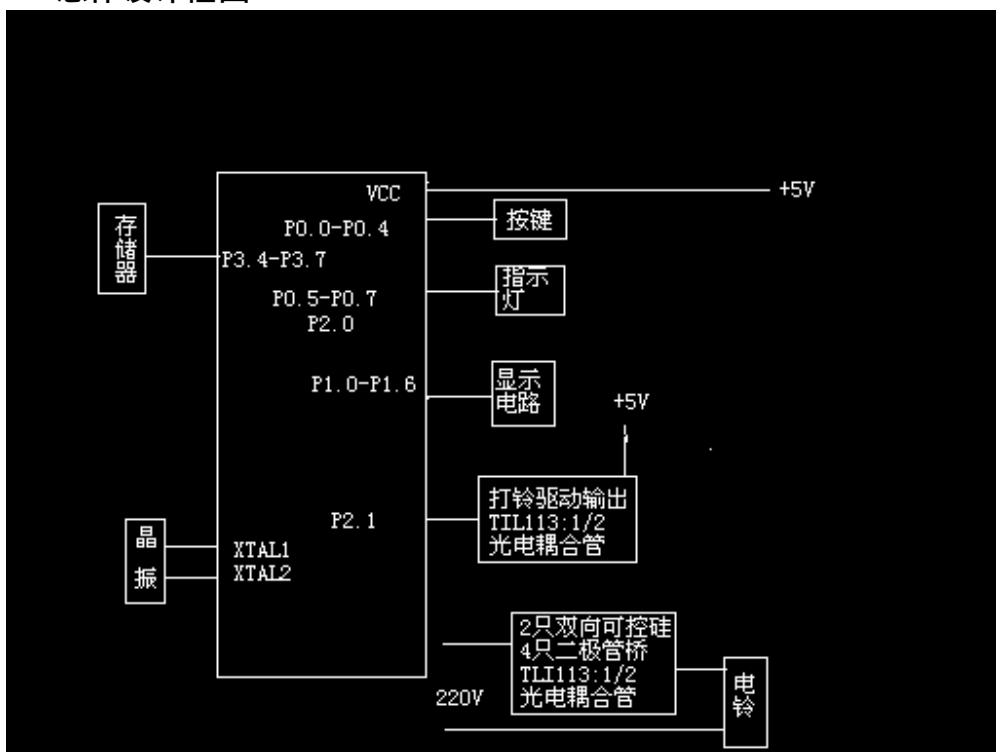
用单片机控制的自动打铃器，充分发挥单片机体积小，价格便宜，功耗低，可靠性好等特点。可用于学校作息，方便了广大师生。

二、总体设计方案

1. 设计思路

利用单片机及定时器设计的一个时钟，在每次秒加 1 的计时过程中，都与规定的时间作比较，如果相等就进行开关电铃，不等则返回。

2. 总体设计框图



总图 1

三、设计原理分析

1. 校时模式

按下 K1 (模式选择键)，将模式选择为校时模式 (模式指示灯黄灯亮)，K2, K3, K4 分别为 时钟加 1 键，分钟加 1 键，秒钟加 1 键，K5 无用。

2. 定时模式

按下 K1 (模式选择键)，将模式选择为定时模式 (模式指示灯绿灯亮)，K2 为定时器组选择 (本装置有 24 组定时器) 键，K3, K4 分别为定时器时钟加 1 键和定时器分钟加 1 键，

K5为定时器状态选择键,每按 K5一次,此组的定时器状态改变一次,并将此组数据存入 93C46 中。

3.工作模式

按下 K1(模式选择键),将模式选择为工作模式(模式指示灯红灯亮),K2, K3, K4, K5 无用。此模式为正常工作模式

4.存储器 93C46的相关介绍

存储器 AT93C46为 SPI接口,1脚为 CS,2脚为 CLK,3脚为 DI,4脚为 DO,5脚为 GND,6脚为 ORG,7脚为 NC,8脚为 VCC。单 5V电压, DIP8, 64*16(ORG为 HI) 或 128*8(ORG为 LO), 芯片及字的清除,时钟最高为 250KHZ。读(READ)操作:当下达 10XXXX指令后,将地址(XXXXXX) 的数据在 CLK=HI 时由 DO 输出。写(WRITE)操作:在写入之前,必须先下达写使能(EWEN) 指令,然后在下达 01XXXXX指令后,当 CLK为 HI 时,在把数据写入指定的地址(XXXXXX),而 DO=0 时,表示还在烧写,烧写完成后,DO 会转为 HI,写入动作完成后,必须再下达写禁止(ENDS) 指令。写使能(EWEN):下达 0011XXXX指令后,才可进行写入(WRITE)的数据清除。写禁止(ENDS):

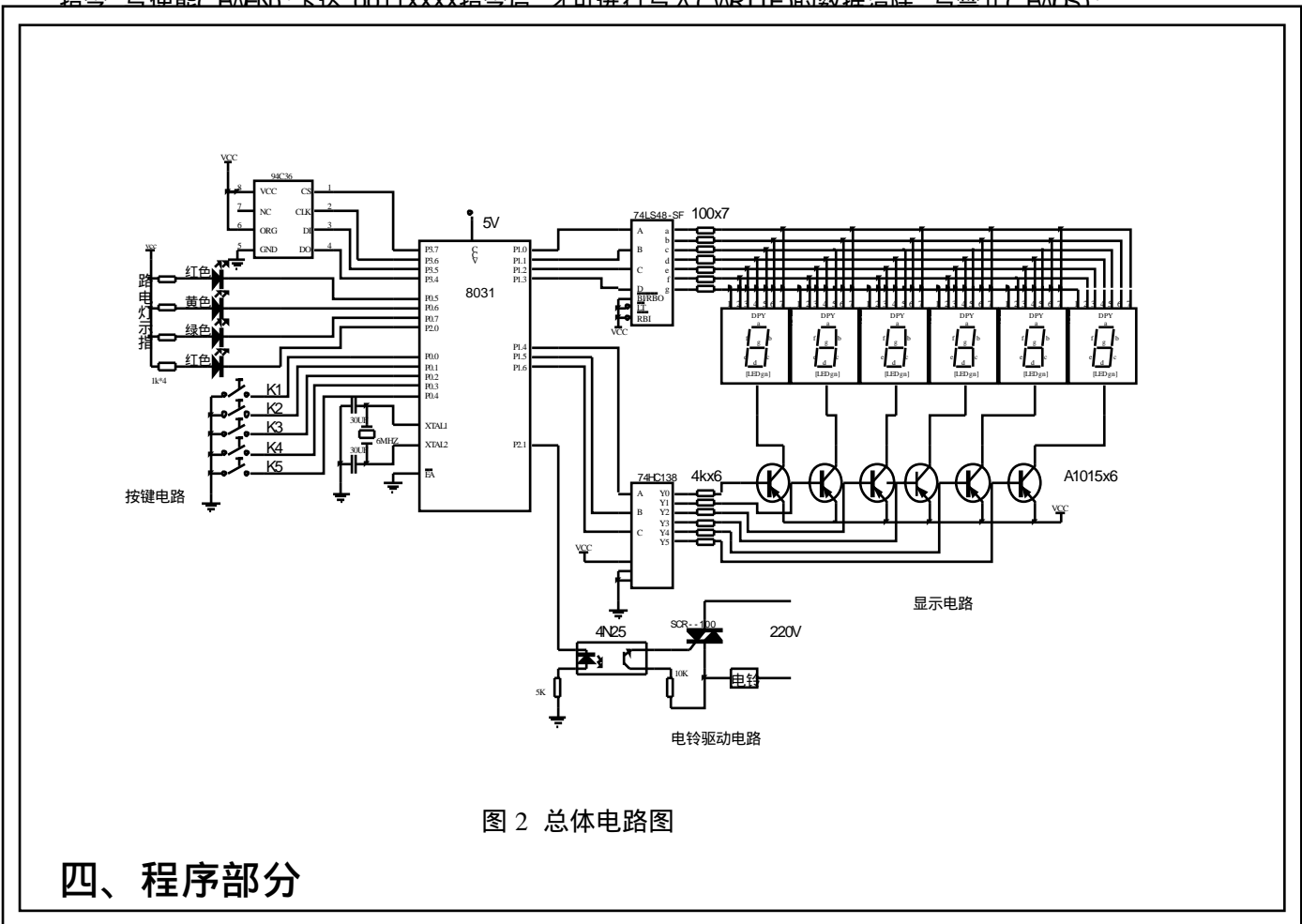


图 2 总体电路图

四、程序部分

1.程序分为五部分

1.初始化程序, 2.键值处理程序, 3.中断程序, 4.显示程序, 5.存储器 93C46读写程序

2.部分寄存器及按键说明

28H: 扫描指针寄存器

20H-25H: 显示缓存寄存器

R6: 时钟寄存器

R5: 分钟寄存器

- R4: 秒钟寄存器
- K1: 模式选择键
- K2: 时加 1(定时器组加 1) 键
- K3: 分加 1(定时器时加 1) 键
- K4: 秒加 1(定时器分加 1) 键
- K5: 定时器状态改变既存储键

3.程序流程图

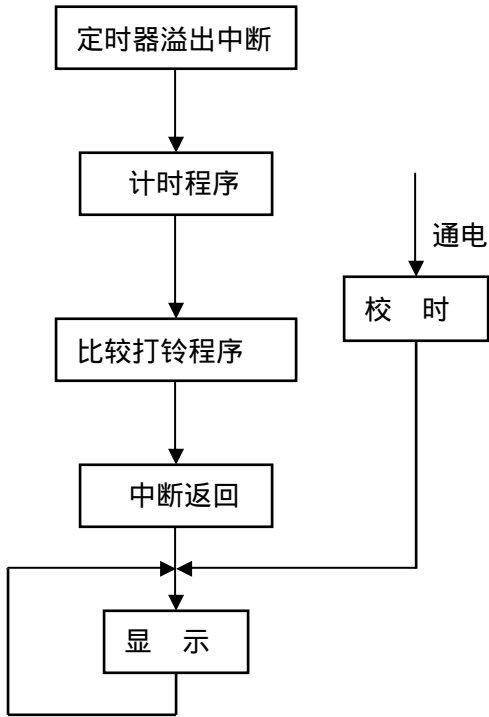


图 3 总体软件流程图

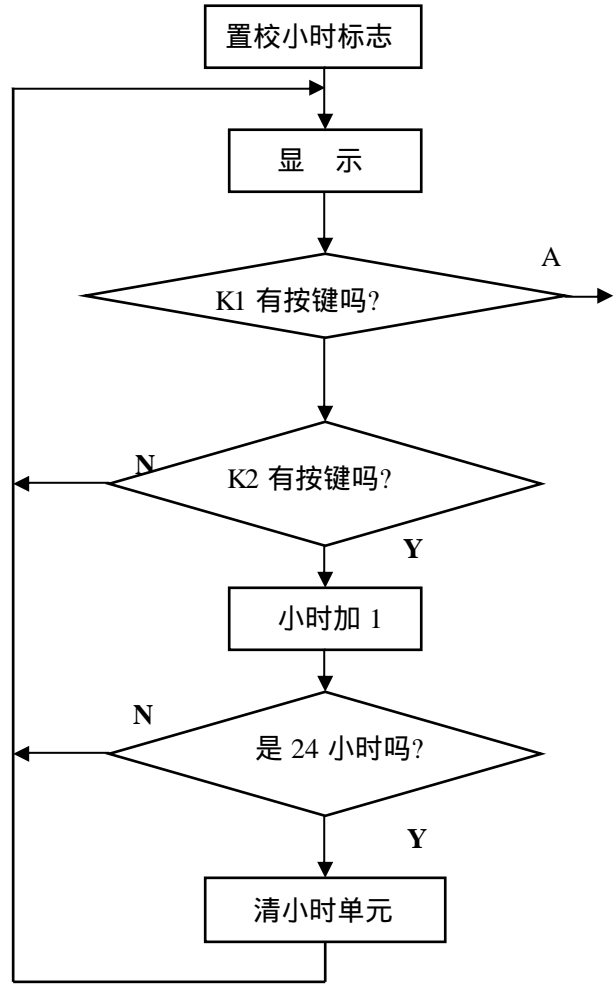


图 4 校时程序流程图

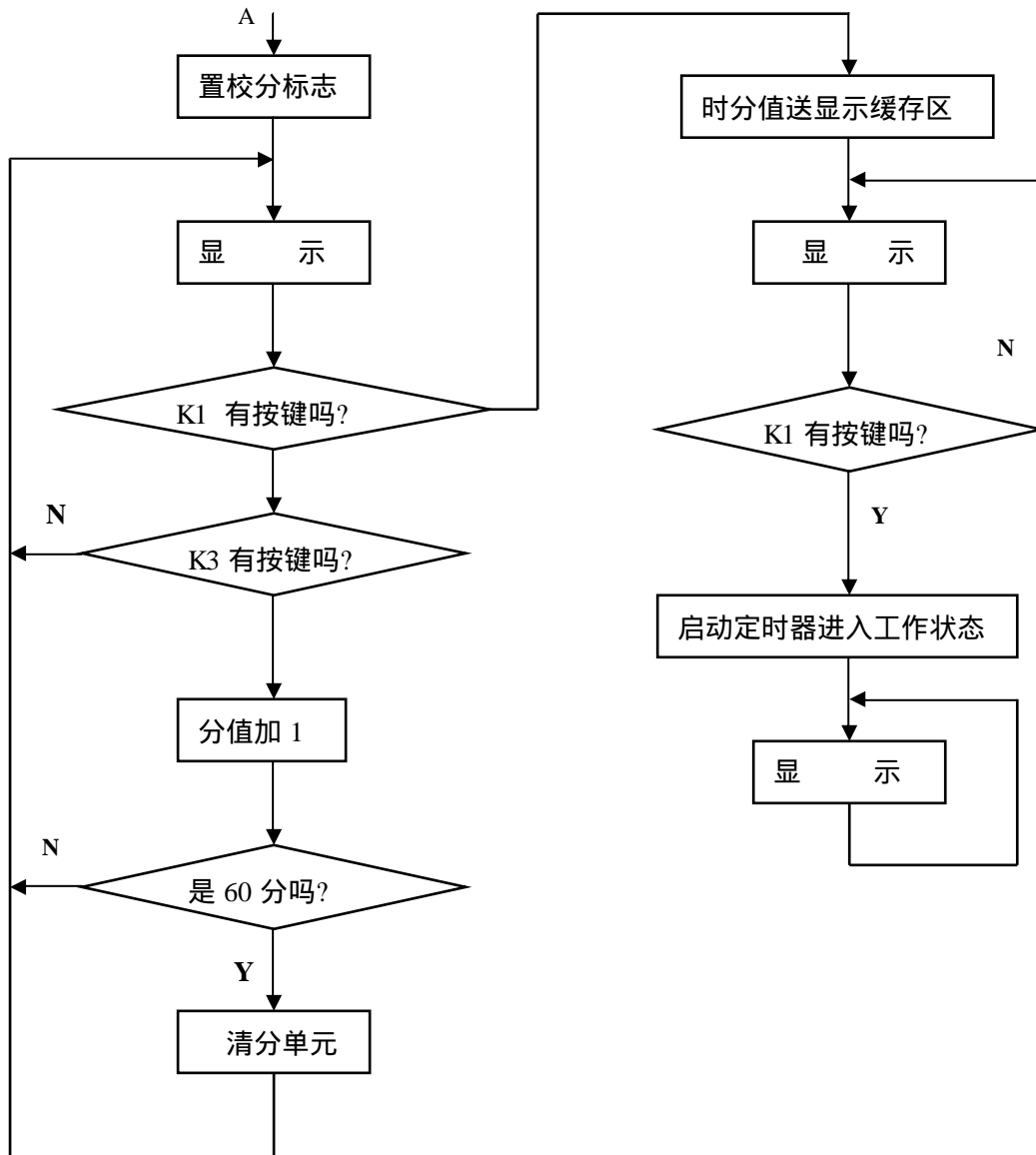


图 5 校分及显示流程图

4. 程序清单

1. 初始化程序：

```

READ    EQU    0      ; 读 93C46
WRITE   EQU    2      ; 写入 93C46
EVEN    EQU    4      ; 93C46写入使能
EVDS    EQU    6      ; 93C46写入禁止
ADR46   EQU    21H    ; 93C46地址
  
```

```

THIGH EQU    0F8H
TLOW EQU    5FH
TIMEN EQU    30          ; 打铃持续时间 ( 30秒 )
CS BIT     P3.7        ; 93C46 CS
CLK BIT     P3.6        ; 93C46 CLK
DI BIT     P3.5        ; 93C46 DI
DO BIT     P3.4        ; 93C46 DO
CRG      0000H        ; 主程序起始地址
AJMP     START        ; 跳至主程序
CRG      000BH        ; TIMER0中断起始地址
AJMP     TIM0         ; 跳至 TIMER0中断子程序
START:MOV SP,#60H     ; 设置堆栈在 60H
MOV      28H,#00      ; 显示寄存器初值为 00
MOV      R6,#12H      ; 时 寄存器值为 12H
MOV      R5,#00H      ; “分”寄存器值为 00H
MOV      R4,#00H      ; “秒”寄存器值为 00H
MOV      TMOD,#01H    ; 设 TIMER0为 MODE1
MOV      TH0,#THIGH   ; 计时中断为 4000微秒
MOV      TL0,#TLOW
MOV      IE,#82       ; TIMER中断使能
MOV      R2,#250      ; 中断 250次
SETB     TR0          ; 启动 TIMER0

2.键值处理程序：
LOOP:    SETB     P0.5    ; 工作模式
          CLR      P0.6
          CLR      P0.7
          JB      P0.0,N1 ; K1( 模式选择键 ) 按了? 不是则跳至 N1
          ACALL   DELAY   ; 消除抖动
          JNB     P0.0,$   ; K1( 模式选择键 ) 放开了?
          ACALL   DELAY   ; 消除抖动
          AJMP    SET      ; 转到定时模式 SET
N1:      MOV      2FH,R6   ; 将时钟数据载入 2FH中
          MOV      2EH,R5   ; 将分钟数据载入 2EH中
          MOV      ADR46,#00 ; 把 93C46地址置 0
          SETB    RS0       ; 将工作寄存器组选择 1区
LOOP1:   MOV      A,#READ   ; 读 93C46
          ACALL   T093c46
          MOV      A,R3      ; 取出定时器的状态
          ANL     A,#01H
          CJNE    A,#01H,N2 ; 定时器为关闭, 则跳转至 N2
          MOV      A,R4      ; 取出定时器分钟数据
          CJNE    A,2FH,N2   ; 定时器分钟数据与时钟分钟不同则转至 N2
          MOV      A,R5      ; 取出定时器时钟数据
          CJNE    A,2EH,N2   ; 定时器时钟数据与时钟时钟不同则转至 N2

```

```

        SETB   P2.1           ; 定时器时间到，打铃
        MOV    A,ADR46        ; 将 93C46地址载入 A中
N2:     CJNE   A,#1EH,N3     ; 到定时器数据表表尾？没到转至 N3
        MOV    ADR46,#00     ; 到表尾，置 93C46地址为表头
        AJMP   LOOP1         ; 转至 LOOP1
N3:     INC    ADR46         ; 93C46地址加 2
        INC    ADR46
        AJMP   LOOP1         ; 转至 LOOP1
SET:    CLR    P0.5          ; 校时模式
        SETB   P0.6
        CLR    P0.7

LOOPS:  JB    P0.0,N11       ; K1(模式选择键)按了？不是则跳至 N11
        ACALL  DELAY         ; 消除抖动
        JNB   P0.0,$         ; K1(模式选择键)放开了？
        ACALL  DELAY         ; 消除抖动
        AJMP   SETTIME      ; 转至 SETTIME
N11:    JB    P0.1,N13       ; K2(时钟加 1键)按了？不是则跳至 N13
        ACALL  DELAY         ; 消除抖动
        MOV    A,R6          ; 将时寄存器的值载入 A
        ADD   A,#01H        ; A内容加 1
        DA    A              ; 做十进制调整
        MOV    R6,A          ; 将 A的值存入时寄存器
        CJNE  A,#24H,N12    ; 是否超过 24时？不是则跳至 N12
        MOV    R6,#00H      ; 是则清除时寄存器的值为 00
N12:    JNB   P0.1,$         ; K2(时钟加 1键)放开了？
        ACALL  DELAY         ; 消除抖动
N13:    JB    P0.2,N15       ; K3(分钟加 1键)按了？不是则跳至 N15
        ACALL  DELAY         ; 消除抖动
        MOV    A,R5          ; 将分寄存器的值载入 A
        ADD   A,#01         ; A内容加 1
        DA    A              ; 做十进制调整
        MOV    R5,A          ; 将 A的值存入分寄存器
        CJNE  A,#60H,N14    ; 是否超过 60分？不是则跳至 N14
        MOV    R5,#00H      ; 是则清除分寄存器的值为 00
N14:    JNB   P0.2,$         ; K3(分钟加 1键)放开了？
        ACALL  DELAY         ; 消除抖动
N15:    JB    P0.3,N17       ; K4(秒钟加 1键)按了？不是则跳至 N17
        ACALL  DELAY         ; 消除抖动
        MOV    A,R4          ; 将秒寄存器的值载入 A
        ADD   A,#01         ; A内容加 1
        DA    A              ; 做十进制调整
        MOV    R4,A          ; 将 A的值存入秒寄存器
        CJNE  A,#60H,N16    ; 是否超过 60秒？不是则跳至 N16
        MOV    R4,#00       ; 是则清除分寄存器的值为 00

```

```

N16:  JNB  P0.3,$           ; K4( 秒钟加 1键 ) 放开了?
      ACALL  DELAY         ; 消除抖动
N17:  AJMP  LOOPS         ; 转至 LOOPS
SETTIME:CLR  P0.5        ; 定时模式
      CLR  P0.6
      SETB  P0.7
      MOV  ADR46,#00H
      MOV  A,#READ        ; 读 93C46地址为 ADR46中的数据
      ACALL  TC9346
LOOPB: ACALL  DISP         ; 调用 DISP
      JB   P0.0,N21       ; K1( 模式选择键 ) 按了? 不是则跳至 N21
      ACALL  DELAY         ; 消除抖动
      JNB  P0.0,$         ; K1( 模式选择键 ) 放开了?
      ACALL  DELAY         ; 消除抖动
      AJMP  LOOP          ; 转至 LOOP
N21:  JB    P0.1,N22      ; K2( 定时器组加 1键 ) 按了? 不是则跳至 N22
      ACALL  DELAY         ; 消除抖动
      MOV  A,ADR46        ; 将地址载入 A
      CLR  C               ; 清除进位位
      RRC  A               ; A右移 1位
      ADD  A,#01H         ; 地址加 1
      MOV  R6,A           ; 将 A存入 R6
      CLR  C               ; 清除进位位
      RLC  A               ; A左移 1位
      MOV  ADR46,A        ; 将 A中地址存入 ADR46
      MOV  A,#READ        ; 读 93C46地址为 ADR46中的数据
      ACALL  TC9346
      JNB  P0.1,$         ; K2( 定时器组加 1键 ) 放开了?
      ACALL  DELAY         ; 消除抖动
N22:  JB   P0.2,N24      ; K3( 定时器分钟加 1键 ) 按了? 不是则跳至 N24
      ACALL  DELAY         ; 消除抖动
      MOV  A,R4           ; 将定时器分寄存器的值载入 A
      ADD  A,#01         ; A内容加 1
      DA   A               ; 做十进制调整
      MOV  R4,A           ; 将 A的值存入定时器分寄存器
      CJNE A,#60H,N23    ; 是否超过 60分? 不是则跳至 N16
      MOV  R4,#00H       ; 是则清除分寄存器的值为 00
N23:  JNB  P0.2,$         ; K3( 定时器分钟加 1键 ) 放开了?
      ACALL  DELAY         ; 消除抖动
N24:  JB   P0.3,N26      ; K4( 定时器时钟加 1键 ) 按了? 不是则跳至 N26
      ACALL  DELAY         ; 消除抖动
      MOV  A,R5           ; 将定时器时寄存器的值载入 A
      ADD  A,#01H         ; A内容加 1
      DA   A               ; 做十进制调整

```

```

MOV    R5,A           ; 将 A的值存入定时器寄存器
CJNE  A,#24H,N25     ; 是否超过 24时? 不是则跳至 N25
MOV    R5,#00        ; 是则清除分寄存器的值为 00
N25:  JNB  P0.3,$     ; K4( 定时器时钟加 1键 ) 放开了?
      ACALL DELAY    ; 消除抖动
N26:  JB   P0.4,N27   ; K5( 改变定时器状态存储键 ) 按了? 不是则跳至 N27
      ACALL DELAY    ; 消除抖动
      INC  R3         ; 改变 R3 最低位的值 ( 表示定时器的状态, 1 表示开, 0
表示关 )
      MOV  A,#WRITE   ; 将 A的值写入 93C46地址为 ADR46中
      ACALL TO9346
      JNB  P0.4,$     ; K5( 改变定时器状态存储键 ) 放开了?
      ACALL DELAY    ; 消除抖动
N27:  AJMP LOOPB     ; 转至 LOOPB

```

3.读写 93C46程序:

```

TO9346: SETB  CS           ; CS=1
      SETB  DI           ; DI=1
      SETB  CLK         ; CLK=1
      ACALL DELAY
      CLR   CLK         ; CLK=0
      ACALL DELAY
      MOV  DPTR,#JPTBL46
      JMP  @A+DPTR ;
JPTBL46:AJMP SREAD      ; 读
      AJMP  SWRITE      ; 写
      AJMP  SENEN      ; 写使能
      AJMP  SENDS      ; 写禁止
SREAD:  MOV  A,ADR46    ; 地址载入 A
      ADD  A,#80H      ; 10XXXX读指令
      ACALL SDT46      ; 读入地址
      ACALL RDT46      ; 读定时器时钟数据
      MOV  R5,A        ; 存入 R5
      ACALL RDT46      ; 读定时器分钟数据
      MOV  R4,A        ; 存入 R4
      INC  ADR46      ; 地址加 1
      MOV  A,ADR46    ; 地址载入 A
      ADD  A,#80H      ; 10XXXX读指令
      ACALL SDT46      ; 读入地址
      ACALL RDT46      ; 读定时器状态数据
      ACALL RDT46
      DEC  ADR46
      MOV  R3,A        ; 存入 R3
      AJMP  EX9346

```

```

SWRITE:MOV A,ADR46      ; 写入地址载入 A
      ADD A,#40H        ; 01XXXX写指令
      ACALL SDT46       ; 载入地址
      MOV A,R5          ; 载入定时器时钟数据
      ACALL SDT46       ; 写入定时器时钟数据
      MOV A,R4          ; 载入定时器分钟数据
      ACALL SDT46       ; 写入定时器分钟数据
      INC ADR46         ; 地址加 1
      MOV A,ADR46       ; 写入地址载入 A
      ADD A,#40H        ; 01XXXX写指令
      ACALL SDT46       ; 载入地址
      MOV A,R3          ; 载入定时器状态数据
      DEC ADR46         ; 地址减 1
      ACALL ADR46       ; 写入定时器状态数据
      AJMP EX9346       ; 转至 EX9346
SEVEN:MOV A,#30H        ; 0011XXXX写使能指令
      ACALL SDT46
      AJMP EX9346
SEVDS:CLR A             ; 0000XXXX写禁止指令
      ACALL SDT46
EX9346:CLR CS           ; CS=0
      RET
SDT46:MOV R2,#8         ; 写入 8位数据
SD1:  RLC A
      MOV DI,C
      SETB CLK
      ACALL DELAY
      CLR CLK
      ACALL DELAY
      DJNZ R2,SD1
      RET
RDT46:MOV R2,#8         ; 读出 8位数据
RD1:  SETB CLK
      ACALL DELAY
      CLR CLK
      ACALL DELAY
      MOV C,DO
      RLC A
      DJNZ R2,RD1
      RET
DELAY:MOV R7,#0FH      ;
      DJNZ R7,$
      RET

```

4.中断程序：

```
TIM0:MOV TH0,#THIGH ;重设计时 4000微秒
      MOV TLO,#TLOW
      PUSH ACC ;将 A的值存入堆栈
      PUSH PSW ;将 PSW的值存入堆栈
      CLR RS0 ;设置工作寄存器为 0区
      DJNZ R2,X2 ;计时 1秒
      MOV R2,#250
      ACALL CLOCK ;调用计时子程序 CLOCK
      ACALL DISP ;调用显示子程序
X2: ACALL SCAN ;调用扫描子程序
     POP PSW
     POP ACC
     RETI
```

显示器驱动程序：

```
SCAN:MOV R0,#28H ;( 28H) 为扫描指针
      INC @R0 ;扫描指针加 1
      CJNE @R0,#6,X3 ;扫描完 6个显示器？不是跳至 X3
      MOV @R0,#00 ;是则扫描指针为 0
X3: MOV A,@R0 ;扫描指针载入 A
     ADD A,#20 ;A加常数 20H( 显示寄存器地址) =各时间显示区地址
     ;存入 R1=各时间显示地址
     MOV R1,A ;扫描指针存入 A
     MOV A,@R0 ;将 A高低 4位交换 ( P1高 4位为扫描值 ,
     SWAP A ;低 4位为显示数据值 )
     ORL A,@R1 ;扫描值 +显示值
     MOV R1,A ;输出至 P1
     RET
```

计时程序：

```
CLOCK:MOV A,R4 ; R4为秒寄存器
      ADD A,#01 ;加 1
      DA A ;做十进制调整
      MOV R4,A ;存入秒 寄存器
      CJNE A,#TIMEN,X4 ;是否超过打零连续时间？不是则跳至 X4
      CLR P2.0 ;关铃
X4: CJNE A,#60H,X5 ;是否超过 60秒？不是则跳至 X5
     MOV R4,#00 ;是则清除为 0
     MOV A,R5 ; R5为分寄存器
     ADD A,#01 ;加 1
     DA A ;做十进制调整
     MOV R5,A ;存入分寄存器
     CJNE A,#60H,X5 ;是否超过 60分？不是则跳至 X5
     MOV R5,#00 ;是则清除为 0
```

```

MOV A,R6           ; R6为时寄存器
ADD A,#01         ; 加 1
DA A              ; 做十进制调整
MOV R6,A          ; 存入时寄存器
CJNE A,#60H,X5   ; 是否超过 24时? 不是则跳至 X5
MOV R6,#00        ; 是则清除为 0
X5: RET

```

5 显示程序：

```

DISP:MOV R1,#20   ;( 20H) 为显示寄存器, R1=20H
MOV A,R6          ; 将时寄存器的内容存入 A
MOV B,#01H        ; 设 B累加器的值为 10H
DIV AB            ; A/B, 商(十位数)存入 A, 余数(个位数)存入 B
MOV @R1,B         ; 将 B的内容存入( 20H)
INC R1            ; R1=21H
MOV @R1,A         ; 将 A的内容存入( 21H)
INC R1            ; R1=22H
MOV A,R5          ; 将分寄存器的内容存入 A
MOV B,#01H        ; 设 B累加器的值为 10H
DIV AB            ; A/B, 商(十位数)存入 A, 余数(个位数)存入 B
MOV @R1,B         ; 将 B的内容存入( 22H)
INC R1            ; R1=23H
MOV @R1,A         ; 将 A的内容存入( 23H)
INC R1            ; R1=24H
MOV A,R4          ; 将秒寄存器的内容存入 A
MOV B,#01H        ; 设 B累加器的值为 10H
DIV AB            ; A/B, 商(十位数)存入 A, 余数(个位数)存入 B
MOV @R1,B         ; 将 B的内容存入( 24H)
INC R1            ; R1=25H
MOV @R1,A         ; 将 A的内容存入( 25H)
RET
END

```

五、总结与体会

经过近两周的努力,在老师和同学的帮助下,我基本上完成了设计任务.通过这次课程设计,我充分认识到了自学的重要性,以及学以致用道理.我在图书馆查阅了大量的资料,同时也认识到了图书馆的重要作用.在今后的学习过程中,应该多到图书馆看一些专业方面的书籍,以丰富自己的知识.也使我加深了对单片机知识的了解和应用和对线路 CAD 的理解和应用.由于知识水平的局限和时间的仓促,设计中可能会存在着一些不足,我真诚的接受老师和同学的批评和指正.

最后衷心感谢老师的悉心指导和同学们的热心帮助!

参考文献

[1] 李朝青.单片机原理及接口技术(修订版)北京:北京航空航天大学出版社,1998

- [2] 李广弟 .单片机基础 北京：北京航空航天大学出版社， 1992
- [3] 何立民 .单片机应用技术大全 .北京：北京航空航天大学出版社， 1994
- [4] 张毅刚 . 单片机原理及接口技术 .哈尔滨：哈尔滨工业大学出版社， 1990
- [5] 谭浩强 .单片机课程设计 . 北京：清华大学出版社， 1989