

数字时钟

福星电子网

提供单片机学习板，开发板，最小系统板；超声波测距实验应用板，

各类器件仪表，详情请访问网站 <http://www.fxdzw.com>

一 摘要

单片计算机即单片微型计算机。(Single-Chip Microcomputer),是集CPU, RAM, ROM, 定时, 计数和多种接口于一体的微控制器。他体积小, 成本低, 功能强, 广泛应用于智能产品和工业自动化上。而51单片机是各单片机中最为典型和最有代表性的一种。这次毕业设计通过对它的学习, 应用, 从而达到学习、设计、开发软、硬的能力。

二 说明

系统由AT89C51、LED 数码管、按键、发光二极管等部分构成, 能实现时间的调整、定时时间的设定, 输出等功能。系统的功能选择由SB0、SB1、SB2、SB3、SB4 完成。其中SB0

为时间校对, 定时器调整功能键, 按SB 0 进入调整状态。SB1 为功能切换键。第一轮按动

SB1 依次进入一路、二路、三路定时时间设置提示程序, 按SB3 进入各路定时调整状态。定

时时间到, 二极管发亮。到了关断时间后灭掉。如果不进入继续按SB1 键, 依次进入时间*i* 年*i* 位校对、*i* 月*i* 位校对、*i* 日*i* 位校对、*i* 时*i* 位校对、*i* 分*i* 位校对、*i* 秒*i* 位校对状态。不管是进入那种状态, 按动SB2 皆可以使被调整位进行不进位增量加1 变化。各

预置量设置完成后, 系统将所有的设置存入RAM 中, 按SB1 退出调整状态。上电后, 系统自

动进入计时状态, 起始于*i* 00*i* 时*i* 00*i* 分。SB4 为年月日显示转换键, 可使原来显示时分秒转换显示年月日。

三、电路原理分析

1. 显示原理

电原理图见附图1。由6 个共阴极管的数码管组成时、分、秒的显示。P0 口的8 条数据线P0.0 至P0.7 分别与两个CD4511 译码的ABCD 口相接, P2 口的 P2.0 至P2.2 分别通过电阻

R10 至R13 与VT1 至VT3 的基极相连接。这样通过P0 口送出一个存储单元的高位、低位BCD

显示代码, 通过P2 口送出扫描选通代码轮流点亮LED1 至LED6, 就会将要显示的数据在数

码管中显示出来。从P0 口输出的代码是BCD 码, 从P2 口输出的就是位选码。这是扫描显示

原理。

。

2 键盘及读数原理

键盘是人与微机打交道的主要设备, 按键的读取容易引起误动作。可采用软件去抖动的方法处理, 软件的触点在闭合和断开的时候会产生抖动, 这时触点的逻辑电平是不稳定的, 如不采取妥善处理的话, 将引起按键命令错误或重复执行, 在这里采用软件延时的方法来避开抖动, 延时时间20ms.

3 连击功能的实现

按下某键时, 对应的功能键解释程序得到执行, 如操作者没有释放按键, 则对应的功能会反复执行, 好象连续执行, 在这里我们采用软件延时250ms, 当按键没释放则

执行下一条对应程序。利用连击功能，能实现快速调时操作。

四、程序设计思想和相关指令介绍

本系统的主程序主要完成时间显示和定时输出判断功能。而年月日显示和各时间单元进位，时间设定时，调定时间设定时等功能全部在中断服务程序中完成。

1. 数据与代码转换。

由前述可知，从P2口输出位选码，从P0口输出段选码，LED就会显示出数字来。但P0口的输出的数据是要BCD码，各存储单元存储的是二进制数，也就是和要显示出的字符表达

的含义是不一致的。可见，将要显示的存储单元的数据直接送到P0口去驱动LED数码管显

示是不能正确表达的，必须在系统内部将要显示的数据经过BCD码行转换后，将各个单元数

据的段选代码送入P0口，给CD4511译码后去驱动数码管显示。

具体转换过程如下：

我们先将要显示的数据装入累加器A中，再将A中的数据转换成高低两位的BCD码，再放回A中，然后将A中的值输出。如：有一个单元存储了45这样一位数，则需转换成四位的BCD码：(0100)(0101)然后放入A中。A中BCD码，高位四位代表 j_4 低四位代

表 j_5 同时送给两个译码器中，译码后 j_4j_5 字就在两个LED中显示出来。

2. 计时功能的实现与中断服务程序

时间的运行依靠定时中断子程序对时钟单元数值进位调整来实现的。计数器T0打开后，进入计时，满100毫秒后，重装定时。中断一次，满一秒后秒进位，满60秒后即1分钟，分钟单元进位，60分到了后，时单元进位，24小时满后，天单元进位。这样然后根据进率，

得到年、月、日、时、分、秒存储单元的值，并经译码后，通过扫描程序送LED中显示出来，

实现时钟计时功能。累加是用指令INC来实现的。

进入中断服务程序以后，执行PUSH PSW和PUSH A将程序状态寄存器PSW的内容和累加器A中的数据保存起来，这便是所谓的 j 保护现场 j 。以保护现场和恢复现场时存取关键数据的存储区叫做堆栈。在软件的控制之下，堆栈可在片内RAM中的任一区间设定，而堆栈

的数据存取与一般的RAM存取又有区别，对它的操作，要遵循 j 后进先出 j 的原则。

3 时间控制功能与比较指令

系统的另一功能就是实现对执行设备的定时开关控制，其主要控制思想是这样的：先将执行设备开启的时间和关闭时间置入RAM某一单元，在计时主程序当中执行几条比较指令，如果当前计时时间与执行设备的设定开启时间相等，就执行一条CLR指令，将对应的那路P3置为高电位，开启；如果当前计时时间与执行设备设定的关闭时间相等，就执行SETB

对应的P3置低电位，二极管截止。实现此控制功能用到的比较指令为CJNE A, #direct, rel, 其转移条件是累加器A中的值与立即数不等则转移。

参考文献

- 1、谢自美，《电子线路设计、实验、测试》武汉：华中理工大学出版社，2000
- 2、何书森、何华斌《实用数字电路原理与设计速成》福州：福建科学技术出版社，2000.6
- 3、白驹衍，《单片计算机及应用》北京：电子工业出版社，1999.2

五：程序

```

SEC EQU 32H ;秒 即时时间 \伪指令
MIN EQU 31H ;分
HOUR EQU 30H ;时
DAY EQU 35H ;日
MON EQU 34H ;月
YEAR EQU 33H ;年
MIN_1 EQU 41H ;分 定时器1 路、开存储单元
HOUR_1 EQU 42H ;时
DAY_1 EQU 43H ;
MON_1 EQU 44H ;
YEAR_1 EQU 45H ;
MIN_11 EQU 40H ;分 定时器1 路、关存储单元
HOUR_11 EQU 46H ;时
DAY_11 EQU 47H ;日
MON_11 EQU 48H ;月
YEAR_11 EQU 49H ;年
;*****
ORG 0000H
ljmp MAIN
ORG 0003H ;中断转换显示年月日、INT0 (SB4
键)
LJMP SHOW
ORG 000BH ;计数中断 T0、方式1
LJMP TIME
ORG 0013H
LJMP CHANGE; 调整时间、定时、INT1 (SB0 键)
;-----主程序
ORG 0030H
MAIN:
;-----初始化付值
MOV YEAR , #02
MOV MON , #05
MOV DAY , #01
MOV HOUR , #00
MOV MIN , #00
MOV SEC , #00
CLR 40H ;定时单元1 路清零
CLR 41H
CLR 42H
CLR 43H
CLR 44H
CLR 45H
CLR 46H
CLR 47H
CLR 48H
CLR 49H
;-----开中断
MOV TMOD , #01H ;计数、模式1、T0
MOV TL0 , #0B0H ;100SM 计数定时
MOV TH0 , #3CH ;
clr p3.0
MOV 20H , #0AH ;10 次*100SM
SETB PT0 ;T0 为最高级
SETB TR0 ;允许计数
SETB ET0 ;允许T0 中断
SETB EX0 ;允许INT0 中断
SETB EX1 ;允许INT1 中断
SETB EA ;开总中断
;-----显示、定时器启动判断
LOOP:
MOV R1 , #30H ; 存储单元
MOV R4 , #01H ; 位选通
MOV R3 , #03H ; 三组显示
NEXT:
MOV A , @R1 ;
MOV B , #10 ;将存储单元转换成两高低两组的
BCD 码
DIV AB
SWAP A
ORL A , B
MOV P0 , A ;输出
MOV P2 , R4
INC R1 ;下一单元
MOV A , R4 ;
RL A ;位移
MOV R4 , A
LCALL DE5SM ;延时0.5SM
DJNZ R3 , NEXT ;全扫描显示一遍
;-----判断定时输出(只编写了一路)
CJNE R7 , #88H,LOOP ;是8 则开, 否则、定时已
关、转
;-----开
MOV A , YEAR
CJNE A , YEAR_1 , LOOP_1 ;年比较, 不等转关
MOV A , MON
CJNE A , MON_1 , LOOP_1
MOV A , DAY

```

```

CJNE A , DAY_1,LOOP_1
MOV A, HOUR
CJNE A, HOUR_1,LOOP_1
MOV A, MIN
CJNE A, MIN_1, LOOP_1
CPL P3.0
;-----关
LOOP_1:
MOV A, YEAR
CJNE A, YEAR_11, LOOP;年比较
MOV A, MON
CJNE A, MON_11, LOOP
MOV A, DAY
CJNE A , DAY_11,LOOP
MOV A, HOUR
CJNE A, HOUR_11,LOOP
MOV A, MIN
CJNE A, MIN_11, LOOP
CPL P3.0
LJMP LOOP
;-----年月日显示中断子程序
SHOW:
PUSH PSW
push ACC
PUSH B
PUSH 01H
PUSH 02H
PUSH 03H
PUSH 04H
MOV R2, #0FFH ;中断扫描次数
TURN: MOV R1 , #33H
MOV R4 , #01H
MOV R3 , #03H
NEXT_1:
MOV A, @R1
MOV B , #10
DIV AB
SWAP A
ORL A, B
MOV P0, A
MOV P2, R4
INC R1
RL A
MOV R4 ,A
LCALL DE5SM
DJNZ R3, NEXT_1
DJNZ R2, TURN ;反复显示一定时间后返回
POP 04H
POP 03H
POP 02H
POP 01H
POP B
POP ACC
POP PSW
RETI
;----计数中断服务子程序
TIME:
PUSH PSW
PUSH ACC
PUSH B
PUSH 06H
MOV TH0 , #3CH;重装计数
MOV TL0 , #0BH;
DJNZ 20H, OUT ;转到中断跳出pop 程序
MOV 20H, #0AH ; 重装: 100*10=1000
;----进位程序
INC SEC
MOV R6, SEC ;
CJNE R6, #60, OUT;比较
MOV SEC , #00 ;
INC MIN
MOV R6, MIN
CJNE R6, #60, OUT
MOV MIN , #00
INC HOUR
MOV R6 , HOUR
CJNE R6 , #25 , OUT
MOV HOUR ,#00
INC DAY
MOV R5, MON
CJNE R5, #1, MON_22;是否1 月、不是转2 月
MOV R5, DAY
CJNE R5, #32, OUT ; 本月是否益出
INC MON
MOV DAY,#1
LJMP OUT
OUT:
POP 06H

```

```

POP B
POP ACC
POP PSW
RETI
MON_22:
MOV R5, MON
CJNE R5, #2, MON_33;是否2 月、不是转3 月
MOV A, YEAR ;判断是否瑞年
MOV B, #4
DIV AB
MOV A, B
JNZ OUT_1;不是则转（A 不为零则转）
MOV R5, DAY
CJNE R5, #30, OUT;如是瑞年、判断是否到29 天
INC MON
MOV DAY, #1
LJMP OUT
OUT_1:
MOV R5, DAY
CJNE R5, #29, OUT ;平年二月判断
INC MON
MOV DAY, #1
LJMP OUT
MON_33:
MOV R5, MON
CJNE R5, #3, MON_44
MOV R5, DAY
CJNE R5, #32, OUT
INC MON
MOV DAY, #1
LJMP OUT
MON_44:
MOV R5, MON
CJNE R5, #4, MON_55
MOV R5, DAY
CJNE R5, #31, OUT
INC MON
MOV DAY, #1
LJMP OUT
MON_55:
MOV R5, MON
CJNE R5, #5, MON_66
MOV R5, DAY
CJNE R5, #32, OUT
INC MON
MOV DAY, #1
LJMP OUT
MON_66:
MOV R5, MON
CJNE R5, #6, MON_77
MOV R5, DAY
CJNE R5, #31, OUT
INC MON
MOV DAY, #1
LJMP OUT
MON_77:
MOV R5, MON
CJNE R5, #7, MON_88
MOV R5, DAY
CJNE R5, #32, L1
INC MON
MOV DAY, #1
L1: LJMP OUT
MON_88:
MOV R5, MON
CJNE R5, #8, MON_99
MOV R5, DAY
CJNE R5, #32, L2
INC MON
MOV DAY, #1
L2: LJMP OUT
MON_99:
MOV R5, MON
CJNE R5, #9, MON_00
MOV R5, DAY
CJNE R5, #31, L3
INC MON
MOV DAY, #1
L3: LJMP OUT
MON_00:
MOV R5, MON
CJNE R5, #10, MON_AA
MOV R5, DAY
CJNE R5, #32, L4
INC MON
MOV DAY, #1
L4: LJMP OUT
MON_AA:

```

```
MOV R5, MON
CJNE R5,#11, MON_BB
MOV R5, DAY
CJNE R5,#31, L5
INC MON
MOV DAY, #1
L5: LJMP OUT
MON_BB:
MOV R5, DAY
CJNE R5, #32, L6
INC YEAR
MOV MON, #1
MOV DAY, #1
L6: LJMP OUT
;-----按SB2定时器年单元加1 子程序
SB3_2: LJMP SHOW_2 ;二路没编返回
SB3_3: LJMP SHOW_3 ;三路没编返回
SB3_1:
MOV A, YEAR_1 ; 调时年单元
MOV B, #10
DIV AB
SWAP A
ORL A, B
MOV P0, A
MOV P2, #01H
LCALL READ
LCALL DE250SM
CJNE A, 01H, SB3_1
CJNE A, #0FBH, KEY2_7 ;按SB2 转年调整
LJMP MON_111 ;按SB1 往下调月单元
KEY2_7: CJNE A, #0FDH, SB3_1
INC YEAR_1 ; 1 路年单元加1
MOV R5, YEAR_1
CJNE R5, #09, SB3_1 ;溢出
MOV YEAR_1, #00H
AJMP SB3_1 ;
;-----月单元加1 子程序
MON_111:
MOV A, MON_1 ; 调时月单元显示
MOV B, #10
DIV AB
SWAP A
ORL A, B
MOV P0, A
MOV P2, #01H
LCALL READ
LCALL DE250SM
MOV P2, #02H
LCALL READ
LCALL DE250SM
CJNE A, 01H, MON_111
CJNE A, #0FBH, KEY2_8 ;按SB2 转月调整
LJMP DAY_111
KEY2_8:
CJNE A, #0FDH, MON_111
INC MON_1 ;1 路月单元加1
MOV R5, MON_1
CJNE R5, #13, MON_111 ;溢出
MOV MON_1, #01H
AJMP MON_111 ; 转到月显
;_-----日单元加1 子程序
DAY_111:
MOV A, DAY_1 ; 调时日单元显示提示
MOV B, #10
DIV AB
SWAP A
ORL A, B
MOV P0, A
MOV P2, #04H
LCALL READ
LCALL DE250SM
CJNE A, 01H, DAY_111
CJNE A, #0FBH, KEY2_9 ;按SB2 转日调整
LJMP HOUR_111
KEY2_9: CJNE A, #0FDH, DAY_111
INC DAY_1 ;1 组日单元加1
MOV R5, DAY_1
CJNE R5, #32, DAY_111 ;溢出
MOV DAY_1, #01H
AJMP DAY_111 ; 转到日显
;-----按SB2 时单元加1 子程序
HOUR_111:
MOV A, HOUR_1 ; 调时时单元显示提示
MOV B, #10
DIV AB
SWAP A
ORL A, B
MOV P0, A
MOV P2, #01H
LCALL READ
LCALL DE250SM
```

```
CJNE A, 01H, HOUR_111
CJNE A, #0FBH, KEY2_10; 按SB2 转时调整
LJMP MIN_111
KEY2_10: CJNE A, #0FDH, HOUR_111
INC HOUR_1
MOV R5, HOUR_1
CJNE R5, #24, HOUR_111; 溢出
MOV HOUR_1, #00H
AJMP HOUR_111; 转到时显
;-----分单元加1 子程序
MIN_111:
MOV A, MIN_1; 调时分单元、并显示提示
MOV B, #10
DIV AB
SWAP A
ORL A, B
MOV P0, A
MOV P2, #02H
LCALL READ
LCALL DE250SM
CJNE A, 01H, MIN_111
CJNE A, #0FBH, KEY2_11; 按SB2 转分调整
AJMP OFF_CH; 按SB3 往下调定时: 关单元
KEY2_11: CJNE A, #0FDH, MIN_111
INC MIN_1; 1 路分单元加1
MOV R5, MIN_1
CJNE R5, #60, MIN_111; 溢出
MOV MIN_1, #00H
AJMP MIN_111; 转到分显
年单元调整
OFF_CH: MOV A, YEAR_11; 调时年单元
MOV B, #10
DIV AB
SWAP A
ORL A, B
MOV P0, A
MOV P2, #01H
LCALL READ
LCALL DE250SM
CJNE A, 01H, OFF_CH
CJNE A, #0FBH, KEY2_F7; 按SB2 转年调整
LJMP MON_OFF; 按SB1 往下调月单元
KEY2_F7: CJNE A, #0FDH, OFF_CH
INC YEAR_11; 1 路年单元加1

MOV R5, YEAR_11
CJNE R5, #09, OFF_CH; 溢出
MOV YEAR_11, #00H
AJMP OFF_CH;
;-----月单元加1 子程序
MON_OFF:
MOV A, MON_11; 调时月单元显示
MOV B, #10
DIV AB
SWAP A
ORL A, B
MOV P0, A
MOV P2, #02H
LCALL READ
LCALL DE250SM
CJNE A, 01H, MON_OFF
CJNE A, #0FBH, KEY2_F8; 按SB2 转月调整
LJMP DAY_OFF
KEY2_F8:
CJNE A, #0FDH, MON_OFF
INC MON_11; 1 路月单元加1
MOV R5, MON_11
CJNE R5, #13, MON_OFF; 溢出
MOV MON_11, #01H
AJMP MON_OFF; 转到月显
;-----日单元加1 子程序
DAY_OFF:
MOV A, DAY_11; 调时日单元显示提示
MOV B, #10
DIV AB
SWAP A
ORL A, B
MOV P0, A
MOV P2, #04H
LCALL READ
LCALL DE250SM
CJNE A, 01H, DAY_OFF
CJNE A, #0FBH, KEY2_F9; 按SB2 转日调整
LJMP HOUR_OFF
KEY2_F9: CJNE A, #0FDH, DAY_OFF
INC DAY_11; 1 组日单元加1
MOV R5, DAY_11
CJNE R5, #32, DAY_OFF; 溢出
MOV DAY_11, #01H
```



```

AJMP DAY_OFF ; 转到日显
;-----按SB2 时单元加1 子程序
HOUR_OFF:
MOV A , HOUR_11 ; 调时时单元显示提示
MOV B ,#10
DIV AB
SWAP A
ORL A,B
MOV P0, A
MOV P2, #01H
LCALL READ
LCALL DE250SM
CJNE A, 01H,HOUR_OFF
CJNE A, #0FBH, KEY2_F10 ; 按SB2 转时调整
LJMP MIN_OFF
KEY2_F10:CJNE A,#0FDH,HOUR_OFF
INC HOUR_11
MOV R5,HOUR_11
CJNE R5,#24,HOUR_OFF;溢出
MOV HOUR_11, #00H
AJMP HOUR_OFF ; 转到时显
;-----分单元加1 子程序
MIN_OFF:
MOV A , MIN_11 ; 调时分单元、并显示提示
MOV B ,#10
DIV AB
SWAP A
ORL A,B
MOV P0, A
MOV P2, #02H
LCALL READ
LCALL DE250SM
CJNE A, 01H,MIN_OFF
CJNE A, #0FBH, KEY2_F11 ;按SB2 转分调整
LJMP ON_1 ;按SB3 往下调定时：开与关
KEY2_F11: CJNE A, #0FDH, MIN_OFF
INC MIN_11 ;1 路分单元加1
MOV R5, MIN_11
CJNE R5,#60,MIN_OFF;溢出
MOV MIN_11, #00H
LJMP MIN_OFF ; 转到分显
;-----开、关定时
ON_1:CJNE A, #0FBH, MIN_OFF
K1: MOV A, #88H
    
```

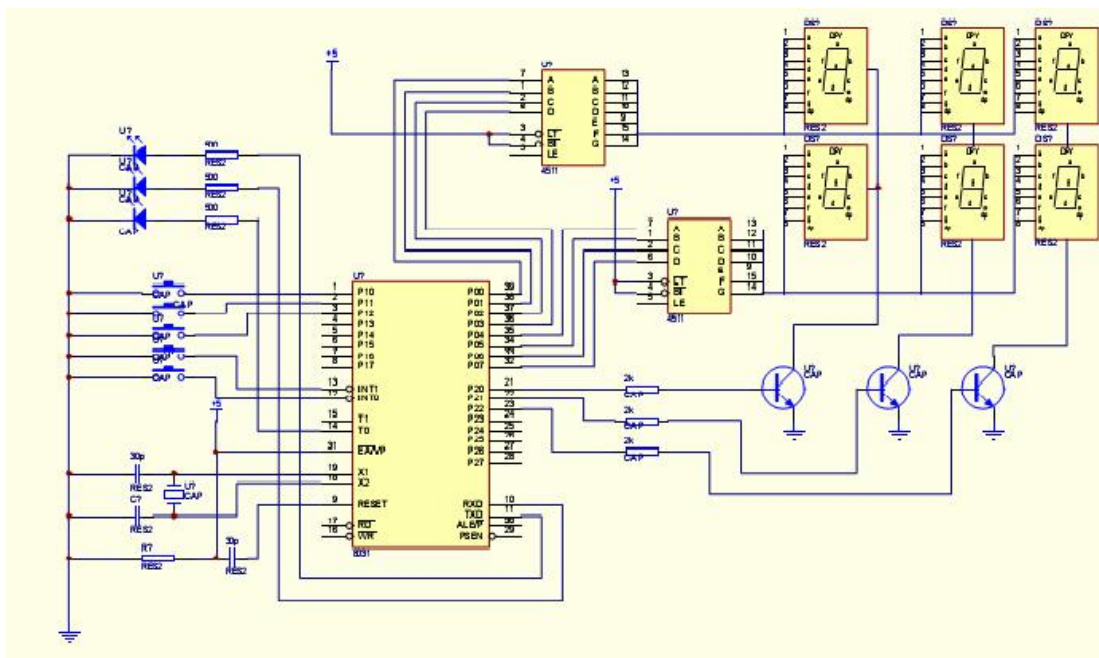
```

MOV R7, A
MOV P0, A
MOV P2, #0FFH;三组都显示开
LCALL READ
LCALL DE250SM
CJNE A, 01H, ON_1;去抖后比较
CJNE A, #0FBH,KEY2_12 ;按SB2 转关
LJMP OUT_A ;按SB3 调出、处于开状态
KEY2_12: CJNE A, #0FDH, K1
K2: MOV A, #00H; 显示0 关
MOV R7, A
MOV P0, A
MOV P2, #0FFH;
LCALL READ
LCALL DE250SM
CJNE A, 01H, K2;去抖后比较
CJNE A, #0FBH,KEY2_13 ;按SB2 转开
LJMP OUT_A ;SB3 调出、处关状态
KEY2_13:
CJNE A, #0FDH, K2 ; 比较按了没
LJMP K1 ; 按了SB2、转开
OUT_A:
POP 00H
POP B
POP ACC
POP PSW
RETI
;_-----读取按键程序
READ:MOV A , P1;读取按键
MOV R1, A
LCALL DE10MS
MOV A, P1
RET
;_----延时程序
DE5SM:
PUSH 01H
MOV R1, #0FFH
DJNZ R1,$
POP 01H
RET
DE10MS: PUSH 04H
PUSH 05H
MOV R4, #0AH
DI1: MOV R5, #0FFH
    
```

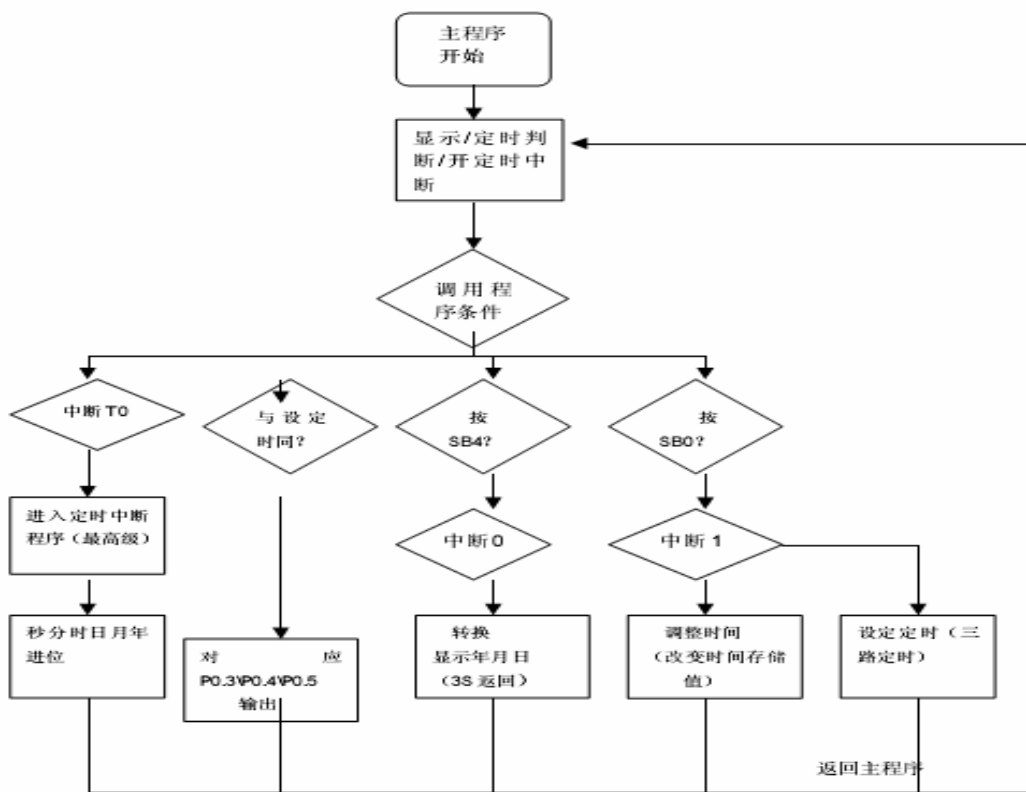
```
dl2: DJNZ R5,$
DJNZ R4,dl1
POP 05H
POP 04H
RET
DE250SM:PUSH 02H
PUSH 00H
MOV R0, #0FFH
DEL:MOV R2, #0FFH
DJNZ R2,$
DJNZ R0, DEL
POP 00H
POP 02H
RET
;---调整时间进位程序
MIN_AD:
INC MIN
MOV R6, MIN
CJNE R6, #60, OU1
MOV MIN, #00
OU1: RET
HOUR_AD:
INC HOUR
MOV R6, HOUR
CJNE R6, #25, OU2
MOV HOUR, #00
OU2: RET
DAY_AD:
INC DAY
MOV R6, DAY
CJNE R6, #32, OU3 ; 是否溢出
MOV DAY, #01H
OU3: RET
MON_AD:
INC MON
MOV R6, MON
CJNE R6, #13, OU4 ; 是否溢出
MOV MON, #01H
OU4: RET
YEAR_AD:
INC YEAR
MOV R6, YEAR
CJNE R6, #09, OU5;是否溢出
MOV YEAR, #00H
OU5: RET
END
```

六：附录

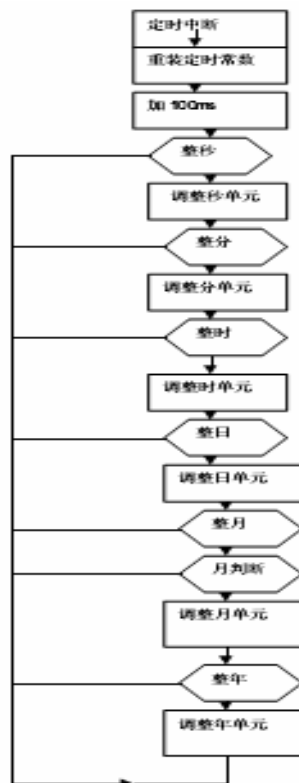
实验设计电路图 1



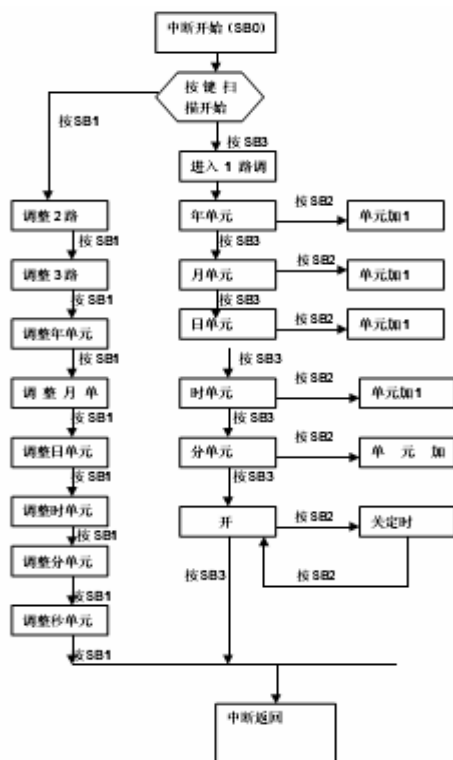
流程图 1: 实验主程序流程图



流程图 2: 定时中断程序流程图



流程图 3：调时功能流程图



七：实验心得

学了两周的课程设计，有很多的心得体会，有关于单片机方面的，更多的是关于人与人之间关系方面的。

在这期间，我得到了很多同学的帮助。我本人对单片机也并不是很熟悉，学的东西好像它是它，我是我似的，理论联系不了实际。以前的汇编语言没学好，一开始的程序这块儿就要令我束手无策了。后来请教我们班的一个男生，每次跟他一起到实验室调试程序，看他边做边给我讲解。最后在计算机上调试成功，后来自己又抽空做了些拓展，完成了本程序。

后来，我发现自己对单片机也有了很大兴趣，想暑假回家以后自己去买一些东西来做，再补一补汇编语言。

最后说明自己对这门课程的感受，课堂教学考虑到大多数同学的需求，主要强调“基本”——基本知识、基本理论、基本方法、基本技能。而这次设计正是为我们提供了一个深入学习、探索的机会，成为课堂教学的有益补充。

基于汇编语言的 单片机的设计

设计名称：电子时钟设计

班级：

学号：

姓名：

指导教师：