

微型计算机技术专业方向设计

任务书

题目名称：基于 MCS-51 单片机的频率计的设计

专业自动化 班级自动化 112 班

姓名都明枢 学号201128052

学校：青岛理工大学自动化学院

指导教师：史贺男

2013 年 12 月 9 日

课程设计任务书

课程名称：微型计算机技术

设计题目：基于 MCS-51 单片机的频率计的设计

系统硬件要求：

- 1、最小系统设计：AT89C51 单片机为本设计的控制器，包括外扩 ROM，RAM 各 32KB （其大小由设计者自己设计），系统时钟电路、复位电路等构成的最小系统；
- 2、频率计使用定时器的计数方式，记录频率的个数并显示；显示部分为 LED 动态显示设计；并有键盘设计；
- 3、接口电路的设计：设计者扩展一个并行接口（8155 或 8255），键盘设计由设计者根据需要设计键盘的数量，显示采用 LED 显示，显示电路也根据显示的内容设计；
- 4、有开机显示状态（如显示 88....）；

软件设计：

- 1) 主程序设计（包括初始化芯片，定时器，中断以及 SP 指针等）；
- 2) 各功能子程序设计，键盘子程序、\显示子程序设计，定时，中断程序等；)

其他要求：

- 1、每位同学独立完成本设计。
- 2、依据题目要求，提出系统设计方案。
- 3、设计系统电路原理图。
- 1、调试系统硬件电路、功能程序。
- 2、编制课程设计报告书并装订成册，报告书内容（按顺序）
 - (1) 报告书封面
 - (2) 课程设计任务书
 - (3) 系统设计方案的提出、分析
 - (4) 系统中典型电路的分析
 - (5) 系统软件结构框图
 - (6) 系统电路原理图
 - (7) 源程序
 - (8) 课设字数不少于 2000 字

成绩	
评语	

原创力文档
max.book118.com
预览与源文档一致,下载高清无水印

1、系统设计方案

本方案主要以 MCS-51 单片机为核心设计的频率计。首先，将被测信号送入到信号放大电路进行放大；然后，再将其放大后的信号送到波形整形电路进行整形，把被测得的正弦波或者三角波转变为方波。利用 MCS-51 单片机的计数器和定时器的功能对被测信号进行计数。编写相应的程序可以使 MCS-51 单片机自动调节测量的量程，并把测出的频率数据送到显示电路显示。

门控信号由 MCS-51 内部的计数定时器产生，由于单片机的计数频率上限较低，所以需对高频被测信号进行硬件预分频处理，MCS-51 则完成运算、控制及显示功能。由于使用了单片机，使整个系统具有极为灵活的可编程性，能方便地对系统进行功能扩展与改进。

2、系统中典型电路的分析

1. MCS-51 单片机及 8255 扩展

(1) MCS-51 单片机

单片机是一种集成在电路芯片，是采用超大规模集成电路技术把具有数据处理能力的中央处理器 CPU 随机存储器 RAM、只读存储器 ROM、多种 I/O 口和中断系统、定时器/计时器等功能（可能还包括显示驱动电路、脉宽调制电路、模拟多路转换器、A/D 转换器等电路）集成到一块硅片上构成的一个小而完善的计算机系统。

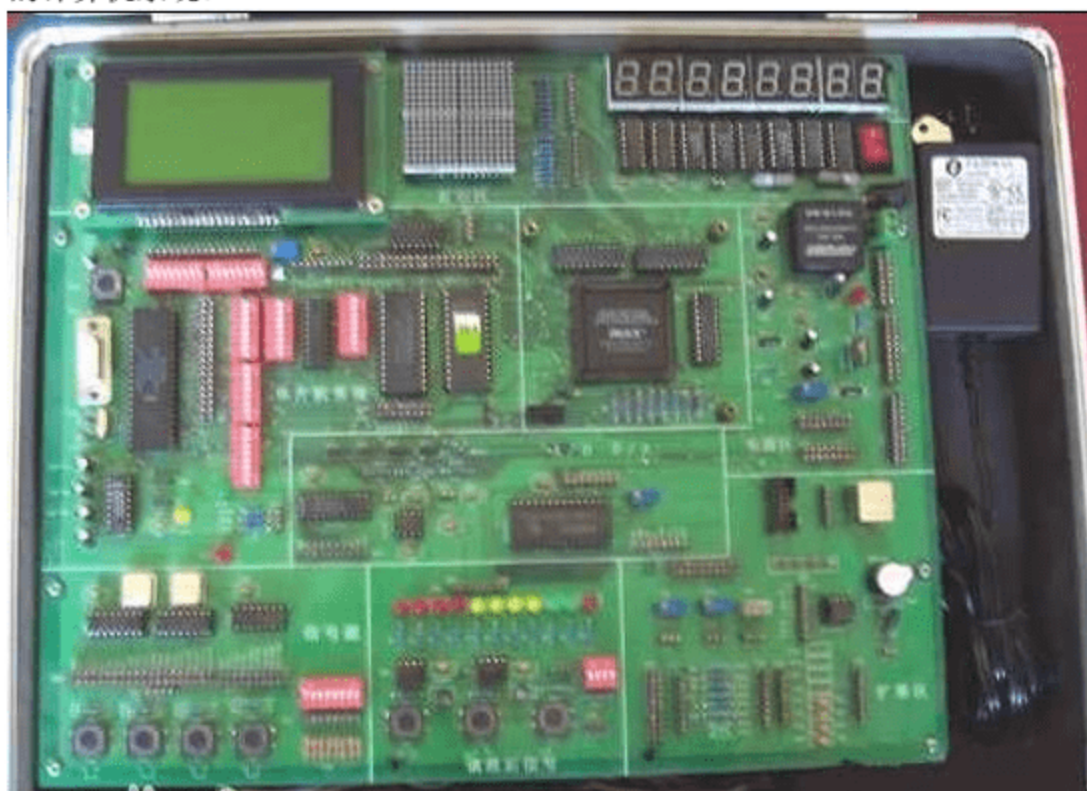


图 1 MCS-51 单片机

(2) 8255 简介

8255 是 Intel 公司生产的可编程并行 I/O 接口芯片，有 3 个 8 位并行 I/O

口。具有 3 个通道 3 种工作方式的可编程并行接口芯片（40 引脚）。其各口功能可由软件选择，使用灵活，通用性强。8255 可作为单片机与多种外设连接时的中间接口电路。

8255 作为主机与外设的连接芯片，必须提供与主机相连的 3 个总线接口，即数据线、地址线、控制线接口。同时必须具有与外设连接的接口 A、B、C 口。由于 8255 可编程，所以必须具有逻辑控制部分，因而 8255 内部结构分为 3 个部分：与 CPU 连接部分、与外设连接部分、控制部分。

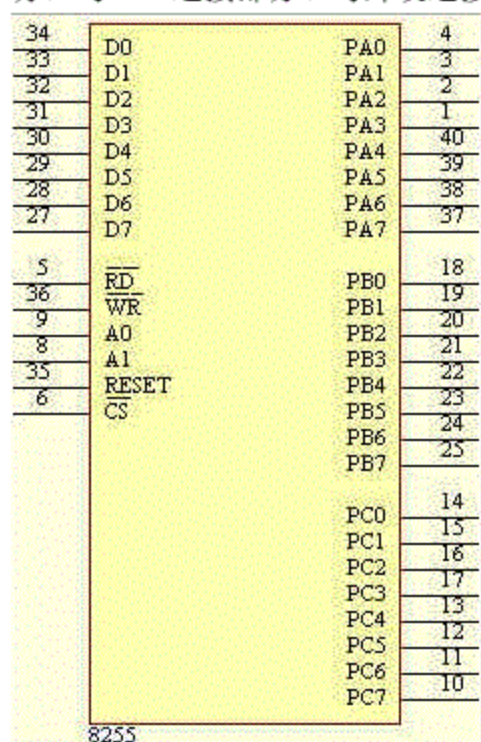


图 2 8255

2. 分频电路

本次设计采用的是脉冲定时测频法，由于考虑到单片机的定时计数器的计数能力有限，无法对高频进行测量，所以我们对待信号进行了分频处理，这样能提高测量频率的范围，还能相应的提高频率测量的精度。因此我们需要把待测信号进行分频。

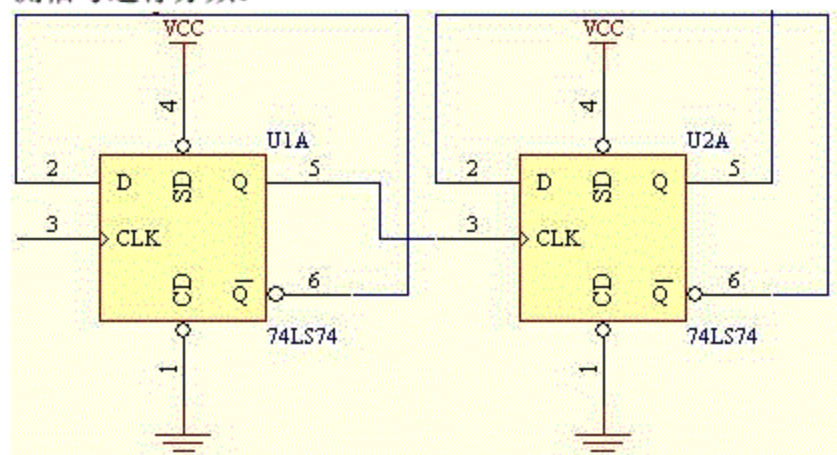


图3 分频电路

3. 显示电路

我们测量的频率最终要用八段 LED 数码管显示器（由8个发光二极管组成）显示出来。其中7个长条形的发光管排列成“日”字形，另一个圆点形的发光管在数码管显示器的右下角作为显示小数点用，它能显示各种数字及部分英文字母。LED 数码管显示器有两种形式：一种是8个发光二极管的阳极都连在一起的，称之为共阳极 LED 数码管显示器；另一种是8个发光二极管的阴极都连在一起的，称之为共阴极 LED 数码管显示器。共阴和共阳结构的 LED 数码管显示器各段名和安排位置是相同的。当二极管导通时，对应的笔划段发亮，由发亮的笔划段组合而显示的各种字符。8个笔划段 h g f e d c b a 对应于一个字节（8位）的 D7、D6、D5、D4、D3、D2、D1、D0，于是用8位二进制码就能表示欲显示字符的字形代码。

在单片机应用系统中，数码管显示器显示常用两种办法：静态显示和动态扫描显示。所谓静态显示，就是每一个数码管显示器都要占用单独的具有锁存功能的 I/O 接口用于笔划段字形代码。这样单片机只要把要显示的字形代码发送的接口电路，就不用管它了，直到要显示新的数据时，再发送新的字形码，因此，使用这种办法单片机中 CPU 的开销小，能供给单独锁存的 I/O 接口电路很多。在单片机系统中动态扫描显示接口是单片机中应用最为广泛的一种显示方式之一。其接口电路是把所有显示器的8个笔划段 a-h 同名端连在一起，而每一个显示器的公共极 COM 是各自独立的受 I/O 线控制。CPU 向字段输出口送出字形码时，所有显示器接收到相同的字形码，但究竟是那个显示器亮，则取决于 COM 端，而这一端是由 I/O 控制的，所以我们就能自行决定何时显示哪一位了。而所谓动态扫描就是指我们采用分时的方法，轮流控制各个显示器的 COM 端，使各个显示器轮流的点亮。在轮流点亮扫描的过程中，每位显示器的点亮时间是极为短暂的（约1ms），但由于人的视觉暂留现象及发光二极管的余辉效应，尽管实际上各位显示器并非同时点亮，但只要扫描的速度足够快，给人的印象就是一组稳定的显示数据，不会有闪烁感。

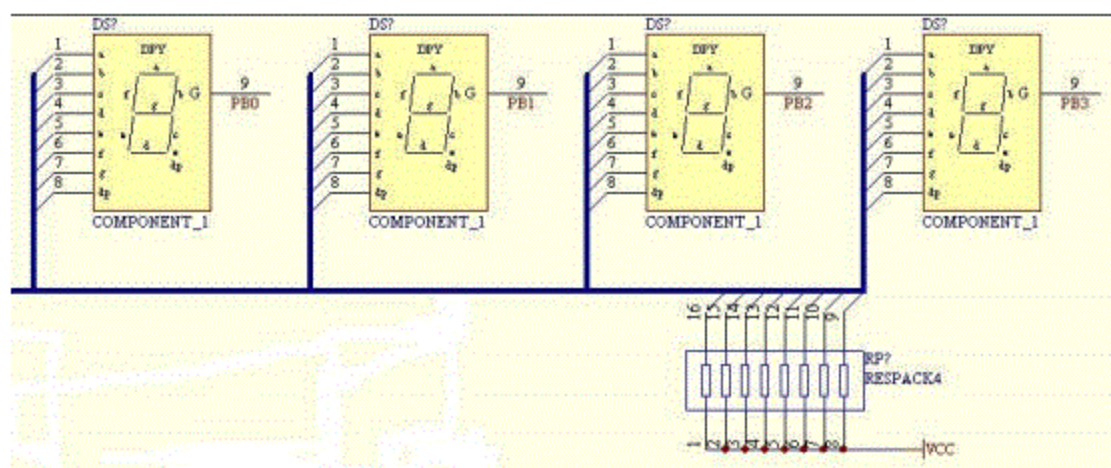


图4 显示电路

4. 指示灯电路

用来指示当前测得频率在在哪个范围段，还可以提示所测频率是否超出测试能力。

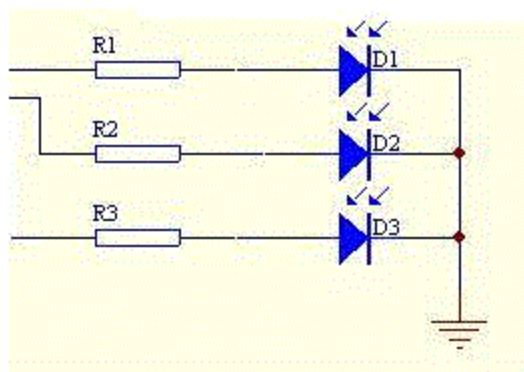


图5 指示电路

5. 报警电路

主要用在测试的频率已超出测试的范围就会报警，来提示用户，所测的电压范围已超出了1MHZ了。

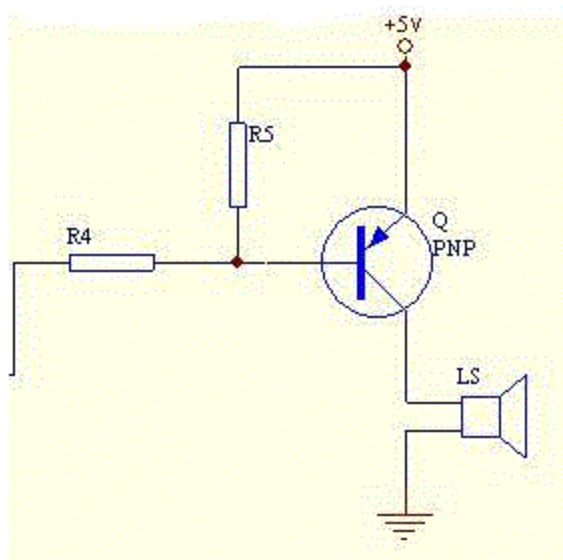


图6 报警电路

6. 按键电路

主要是提供给用户频率段的切换用的。

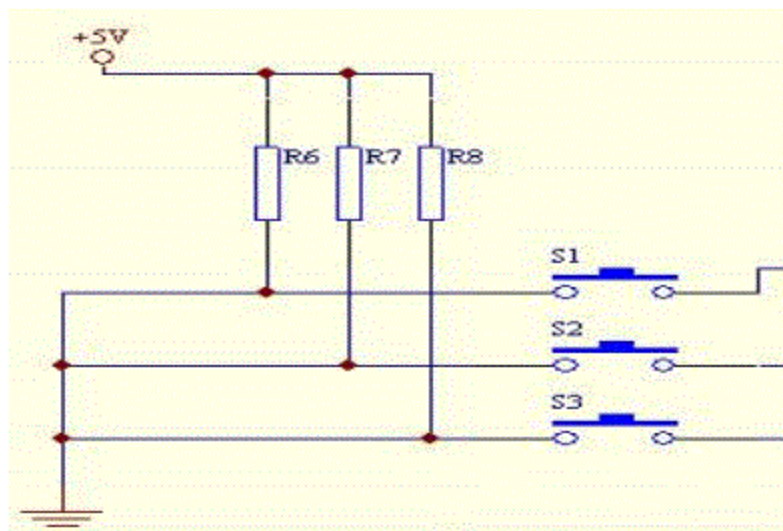


图7 按键电路

3、系统软件结构框图

1. 主程序结构框图

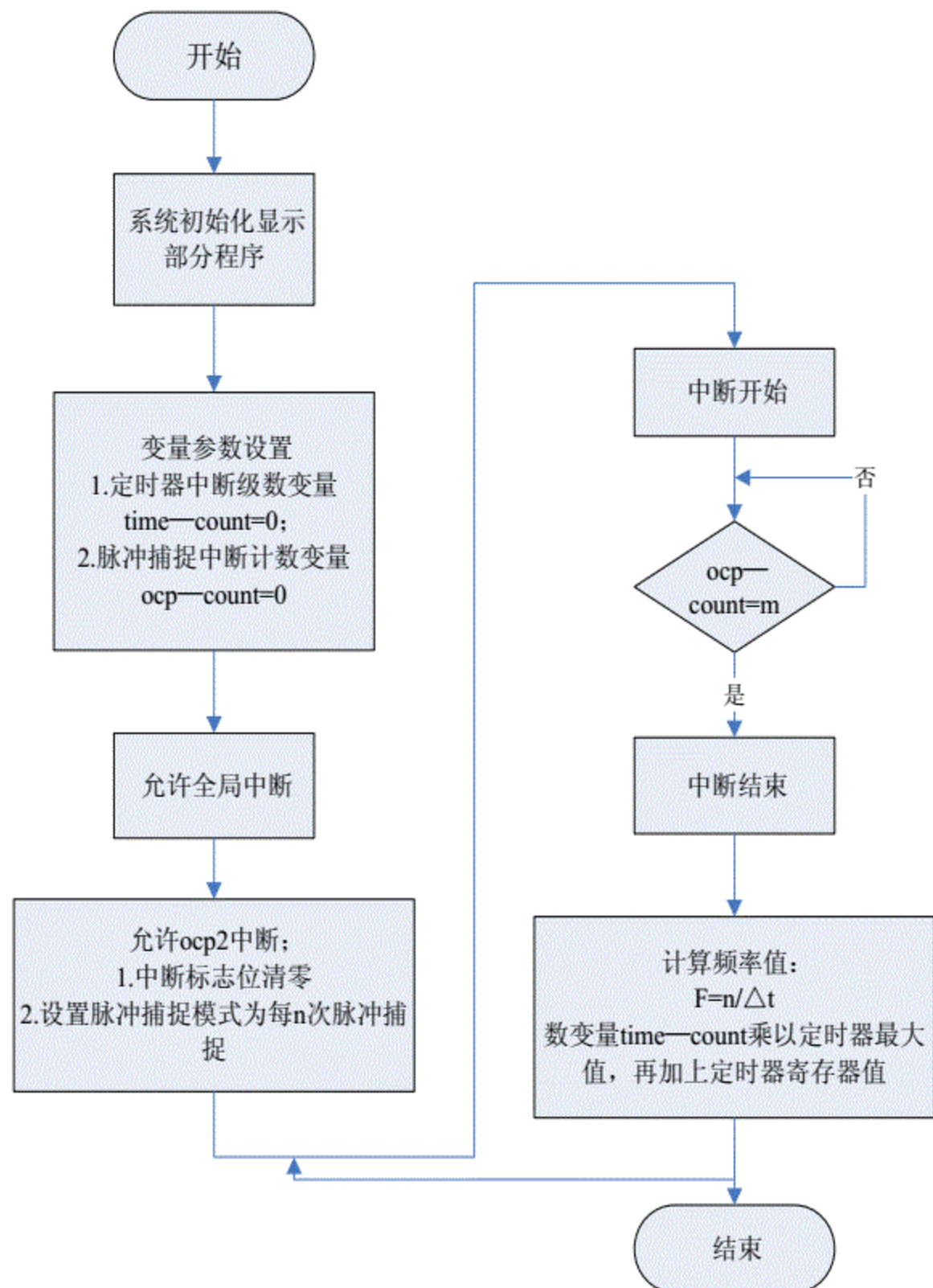


图8 主程序

2. 定时器结构框图

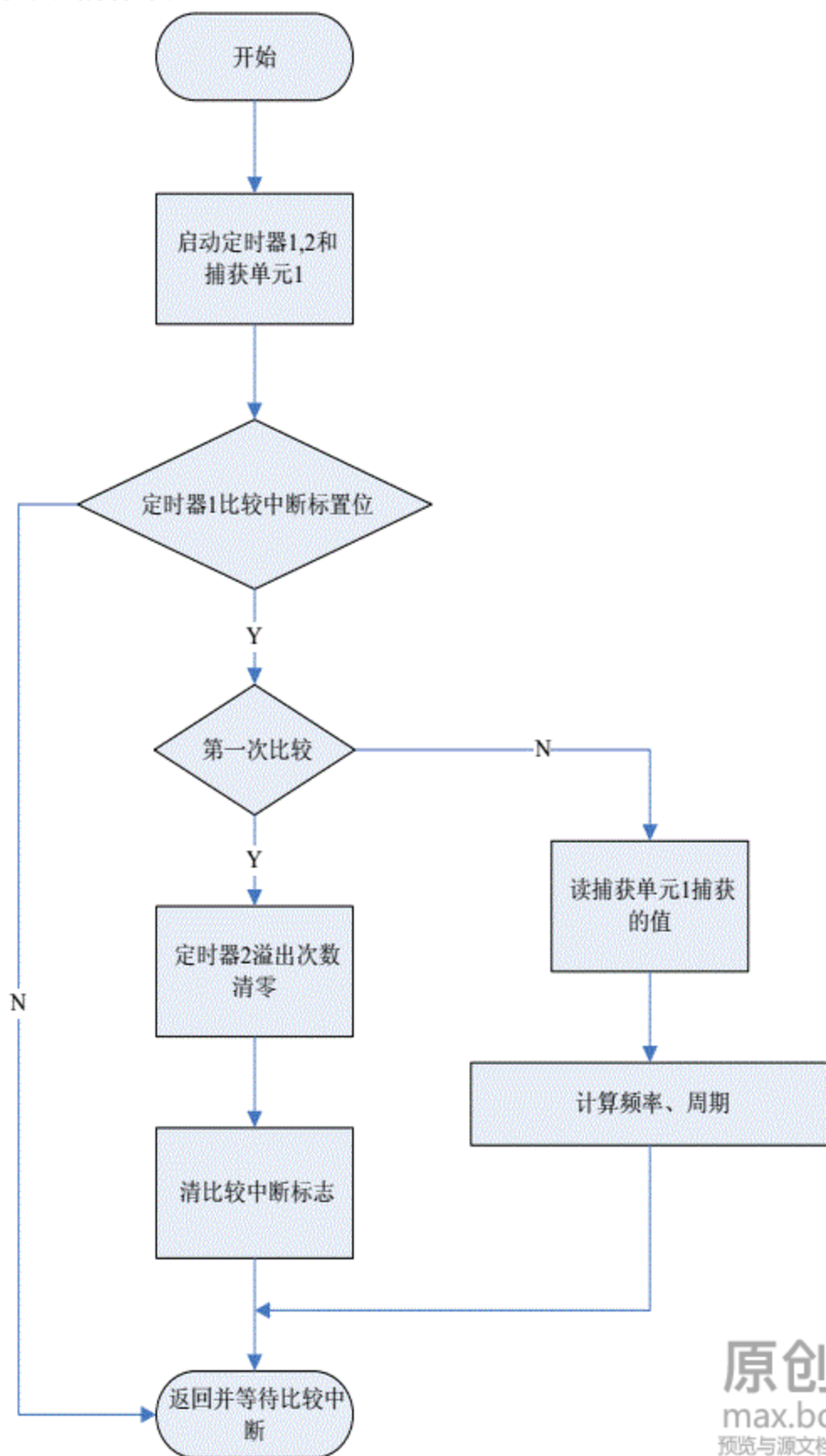


图9 定时器

4、软件的调试

单片机开发中除必要的硬件外，同样离不开软件，我们写的汇编语言源程序要变为 CPU 可以执行的机器码有两种方法，一种是手工汇编，另一种是机器汇编，目前已极少使用手工汇编的方法了。

KEIL 软件提供了丰富的库函数和功能强大的集成开发调试工具。因此我们使用了 KEIL 软件来进行软件的调试。调试结果语句没有错误。

5、课程设计总结

经过了两个周的时间，我们组七个人一起完成了这次的单片机课程设计。在这次课程设计中，我们每一个人都认真的思考过，经过我们的一起讨论之后，确定了本次实验设计的方案，然后我们进行了一下明确的分工，之后我们做了自己的部分，最后我们又在一起合在一起，修正了个别的错误。

在这次课程设计中，本来自己看了任务书感觉很困难但是最后经过我们的努力，我们完成了此次设计。这次在编程中，不像学习课程时写的那么简单了，它是将很多一部分的程序组合成一个大的程序。经过此次设计后发现组合程序不是那么随随便便的，是要讲究一些规则的。经过这次课程设计，我收获了很多很多：程序的编写得到了提高；与同学们团结协作；方案的整体设计；在遇到困难后冷静分析。

参考文献

- [1] 胡汉才编.《单片机原理及其接口技术》.清华大学出版社2010
- [2] 童诗白编.《模拟电子技术基础》.高等教育出版社2006
- [3] 阎石编.《数字电子技术基础》.高等教育出版社2006
- [4] 李华等编.《MCS-51系列单片机实用接口技术》.北京航空航天大学出版社1993
- [5] 徐惠民编.《单片微型计算机原理接口及其应用》.北京大学出版社

2000

附录

(1)、系统的电路原理图

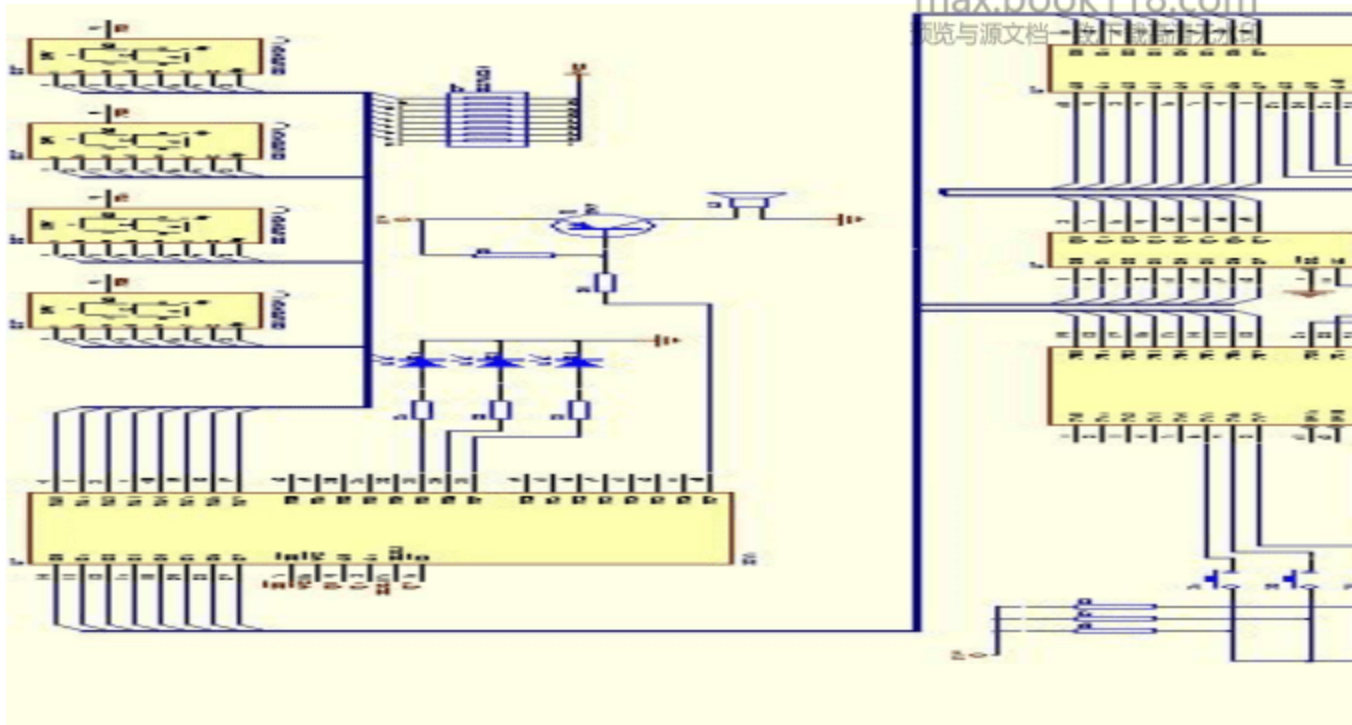


图10 主电路图

(2)、源程序

```
NUMBYT EQU 5DH
SLA EQU 5EH
MTD EQU 5FH
ORG 0000H
LJMP MAIN
ORG 000BH ;T0 中断入口
LJMP INTT0
ORG 001BH ;T1 中断入口
LJMP INTT1
ORG 0030H
MAIN:
MOV SP,#60H
MOV IE,#8AH ;开放 T0、T1 中断
MOV TMOD,#51H ;T0 定时, T1 计数
MOV TH0,#0DCH
MOV TL0,#00H ;定时 10ms
```

```

MOV 20H,#100H ;100*10ms=1s
MOV TH1,#00H
MOV TL1,#00H
MOV 21H,#0
MOV 22H,#0
MOV 23H,#0;存放采集到的频率
MOV R0,#03H;控制口地址送 R0
MOV A,#88H;方式控制字送 A
MOVX @R0,A;送 8255A 控制口
MOV R1,#01H
MOV A,R1
MOVX @R0,A
ACALL XIANSHI1
SETB TR1
SETB TR0

```

WAIT:

```

SJMP $;等待中断

```

XIANSHI1:

```

MOV DPTR,#8003H
MOV A,#00H
MOVX @DPTR,A
MOV DPTR,#TAB
MOV A,@R0
MOVC A,@A+DPTR
MOV DPTR,#8001H
MOVX @DPTR,A
ACALL DELAY
RET

```

INTT1:

```

INC 23H;计数器溢出则 23H 单元自增 1
RETI

```

INTT0: ;定时 10ms 产生中断

```

DJNZ 20H,NEXT1
CLR TR1
CLR TR0
MOV 22H,TH1;1s 时间到则采集数据
MOV 21H,TL1
ACALL DISPLAY
AJMP EXIT

```

NEXT1:

```

MOV TH0,#0DCH;继续定时
MOV TL0,#00H

```

EXIT:

```

RETI

```

DISPLAY:

MOV R0,#60H

MOV R1,#80H;对 60H--67H 单元清零

NEXT2:

MOV @R0,#0

INC R0

DJNZ R1,NEXT2

ZHUANHUAN: ;进制转换

MOV A,23H

MOV B,#0AH

DIV AB

MOV 24H,A ;存储第一位商

MOV A,B

MOV 30H,22H

ANL 30H,#0F0H

ADD A,30H

SWAP A

MOV B,#0AH

DIV AB

MOV 25H,A ;存储第二位商

MOV A,B

SWAP A

ANL 22H,#0FH

ADD A,22H

MOV B,#0AH

DIV AB

MOV 26H,A ;存储第三位商

MOV A,B

MOV 30H,21H

ANL 30H,#0F0H

ADD A,30H

SAWP A

MOV B,#0AH

DIV AB

MOV 27H,A ;存储第四位商

MOV A,B

SWAP A

ANL 21H,0FH

ADD A,21H

MOV B,#0AH

DIV AB

MOV 28H,A ;存储第五位商

MOV 50H,B ;存储十进制个位

MOV A,24H

```

SWAP    A
ADD     A,25H
MOV     B,#0AH
DIV     AB
MOV     24H,A    ;存储第一位商
MOV     A,B
SWAP    A
ADD     A,26H
MOV     B,#0AH
DIV     AB
MOV     25H,A    ;存储第二位商
MOV     A,B
SWAP    A
ADD     A,27H
MOV     B,#0AH
DIV     AB
MOV     26H,A    ;存储第三位商
MOV     A,B
SWAP    A
ADD     A,28H
MOV     B,#0AH
DIV     AB
MOV     27H,A    ;存储第四位商
MOV     51H,B    ;存储十位制数十位
MOV     A,24H
SWAP    A
ADD     A,25H
MOV     B,#0AH
DIV     AB
MOV     24H,A    ;存储第一位商
MOV     A,B
SWAP    A
ADD     A,26H
MOV     B,#0AH
DIV     AB
MOV     25H,A    ;存储第二位商
MOV     A,B
SWAP    A
ADD     A,27H
MOV     B,#0AH
DIV     AB
MOV     26H,A    ;存储第三位商
MOV     52H,B    ;存储十位制数百位
MOV     A,24H

```

```

    SWAP    A
    ADD     A,25H
    MOV     B,#0AH
    DIV     AB
    MOV     24H,A    ;存储第一位商
    MOV     A,B
    SWAP    A
    ADD     A,26H
    MOV     B,#0AH
    DIV     AB
    MOV     25H,A    ;存储第二位商
    MOV     53H,B    ;存储十进制数千位
    MOV     A,24H
    SWAP    A
    ADD     A,25H
    MOV     B,#0AH
    DIV     AB
    MOV     54H,B ; 存储十进制数万位
    MOV     55H,A ; 存储十进制数十万位
PINBI:    ;将高位的 0 屏蔽不显示
    MOV     R3,#0
    MOV     R0,#55H

ST2:
    MOV     A,@R0
    JZ      ST1
    AJMP    SHUMA

ST1:
    INC     R3
    DEC     R0
    AJMP    ST2

SHUMA:
    MOV     A,#6
    CLR     C
    SUBB    A,R3
    MOV     R2,A ; 将需要显示的位数存入 R2
    MOV     R0,#50H
    MOV     R1,#5FH
    MOV     DPTR,#TAB

NEXT3:
    MOV     A,@R0
    MOVC    A,@A+DPTR
    INC     R0
    INC     R1
    MOV     @R1,A

```

```

    DJNZ R2,NEXT3
    MOV MTD,#10H
    MOV NUMBYT,#09H
    MOV SLA,#70H
    LCALL WRNBYT
    RET
WRNBYT:
    PUSH PSW
WRNBYT1:
    MOV PSW,#18h
    CALL STA
    MOV A,SLA
    CALL WRB
    CALL CACK
    JB F0,WRNBYT
    MOV R0,#MTD
    MOV R5,NUMBYT
WRDA:
    MOV A,@R0
    LCALL CACK
    JB F0,WRNBYT1
    INC R0
    DJNZ R5, WRDA
    LCALL STOP
    POP PSW
    RET
WRB:
    MOV R7,#8 ;字节数据发送
WLP:
    RLC A
    JC WR1
    CLR P1.1
    SETB P1.0
    NOP
    NOP
    NOP
    NOP
    CLR P1.0
    DJNZ R7,WLP
    RET
WR1:
    SETB P1.1
    SETB P1.0
    NOP

```

```

NOP
NOP
NOP
CLR  P1.0
CLR  P1.1
DJNZ R7,WLP
RET
CACK:
SETB P1.1
SETB P1.0
NOP
NOP
MOV  C,P1.1
MOV  F0,C
CLR  P1.0
NOP
NOP
RET
STA:
SETB P1.1 ;发送起始位
SETB P1.0
NOP
NOP
NOP
NOP
NOP
CLR  P1.1
NOP
NOP
NOP
NOP
CLR  P1.0
RET
STOP:
CLR  P1.1;发送停止位
SETB P1.0
NOP
NOP
NOP
NOP
SETB P1.1
NOP
NOP
NOP
NOP

```

CLR P1.0

RETI

TAB:

DB 3FH,06H,5BH,4FH,66H

DB 6DH,7DH,07H,7FH,6FH

END