摘 要

在工程问题中，频率是一种最基本的参数，并与其他许多电参量的测量方案和测量结果都有着十分密切的关系。由于频率信号抗干扰能力强、易于传输，可以获得较高的测量精度。因此，频率的测量就显得尤为重要，测频方法的研究越来越受到重视。

本设计由波形发生器模拟输入方波，方波的频率范围是 0~9999Hz，可手动改变被测信号的频率。AT89C52 单片机作为控制器，使用其内部的定时计数器计算方波的频率，使用四位数码管显示被测频率大小，并外接 24C02 芯片用于频率的存储和查询。当被测信号频率大于量程时，会触发报警灯亮起，使用查询功能时，会触发查询灯亮起。总体来看，本设计硬件系统简单且便于软件调试，达到了简易测量频率的需求。

关键词：单片机；频率计；Protues；24C02

目 录

1 绪论 1
2 方案设计 1
2.1 设计目的 1
2.2 设计内容 2
2.3 方案总体阐释 2
3 系统硬件设计 3
3.1 晶振、复位电路 3
3.2 计数显示电路 4
3.3 存储查询电路 5
3.4 指示灯电路 6
4 系统软件设计 6
4.1 频率测算 7
4.2 数码管显示 8
4.3 存储查询 8
5 系统仿真 9
5.1 Protues 简介 9
5.2 仿真调试 9
6 心得体会 11
参考文献 12
附件 13
附件一：硬件原理图 13
附件二：仿真原理图 14
附件三：程序清单 15

1 绪论

在电子技术中，频率是一种计算单位时间内的信号变化的数值的仪器，是最基本的参数之一，由于频率信号抗干扰能力强、易于传输，可以获得较高的测量精度，并且与许多电参量的测量方案、测量结果都有十分密切的关系。因此，频率的测量就显得更为重要。

频率计作为测量仪器的一种，常称为电子计数器，它的基本功能是测量信号的频率和周期频率计的应用范围很广，它不仅应用于一般的简单仪器测量，而且还广泛应用于教学、科研、高精度仪器测量、工业控制等其它领域。频率计也是一种应用较广泛的电子测量仪器。随着微电子技术和计算机技术的迅速发展，特别是单片微机的出现和发展，使传统的电子侧量仪器在原理、功能、精度及自动化水平等方面都发生了巨大的变化，形成一种完全突破传统概念的新一代测量仪器。频率计广泛采用了高速集成电路和大规模集成电路，使仪器在小型化、耗电、可靠性等方面都发生了重大的变化。

市场上的频率计产品很多，但基本上都是采用专用计数芯片和数字逻辑电路组成，由于这些芯片本身的工作频率不高，从而限制了产品工作频率的提高，远不能满足在一些特殊的场合需要测量很高的频率的要求，而且测量精度也受到芯片本身极大的限制。从 80 年代单片机引入我国至今，单片机已广泛地应用于电子设计中，使频率计智能化水平在广度和深度上产生了质的飞跃，数字化也成为了电子设计的必由之路。运用 51 系列单片机和高速计数器的组合设计频率计，并采用适当的算法取代传统电路，不仅能克服传统频率计结构复杂、稳定性差、精度不高的弊端，而且频率计性能也将大幅提高，可实现精度较高、等精度和宽范围频率计的要求。随着单片机技术的不断发展，单片机能实现更加灵活的逻辑控制功能，具有很强的数据处理能力，可以用单片机通过软件设计直接用十进制数字显示被测信号频率。频率计是电子测试、自动化控制等设备中不可或缺的重要模块。在电子工程、资源勘探、仪器仪表等相关应用中，频率计是工程技术人员必不可少的测量工具，频率测量也是电子测量技术中最基本最常见的测量之一。不少物理量的测量，如转速、振动频率等的测量都涉及到或可以转化为频率的测量。

目前，市场上有各种多功能、高精度、高频率的数字频率计，但价格不菲。为适应实际工作的需要，本次课程设计给出了一种较小规模的基于 51 系列单片机（AT89C52）的频率计的设计方案，不但切实可行，而且体积小、设计简单、成本低、便于调试、提供存储查询功能，大大降低了设计成本和实现复杂度。

2 方案设计

2.1 设计目的

1) 掌握单片机应用系统的设计方法和步骤。
2) 掌握单片机应用系统硬件设计技术，进一步掌握电子设计软件 Protel99 或 Altium Design 软件的使用。
3) 掌握单片机应用系统程序编写与调试技术，利用软件 Protues 进行软硬件的联机仿真调试。

2.2 设计内容

本设计以单片机作为控制单元，利用波形发生器模拟频率信号输入，外接 24C02 作为存储器，设计出嵌入式产品，以解决实际问题。设计过程包括系统方案确定，原理图设计，软硬件联机仿真调试。

此频率计实现了以下功能：

1) 可测范围：0Hz-9999Hz；（任务书要求 10Hz-999Hz）
2) 用 LED 显示测量结果，测量结果可以存储，可供查询；
3) 可手动改变被测信号的频率。

2.3 方案总体阐释

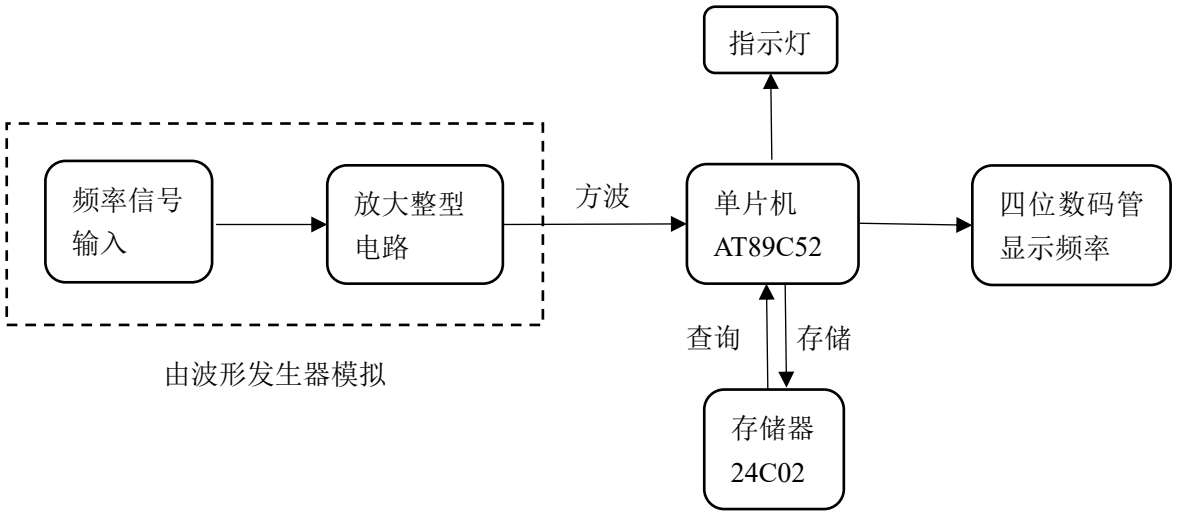


图 1 频率计设计框图

本方案采用的是单片机 AT89C52 作为控制器，它作为整个系统的枢纽，系统的显式输入是经过整形后的方波信号。系统还存在一隐式输入，即 24C02，串行 E2PROM 是基于 I2C-BUS 的存储器件，它用于存储所测得的频率信号，便于查询使用。两种输入存在某种关系，即输入的方波信号可以进过单片机计算出频率后存储到存储器中。存储器中的频率数据可以直接进过显示电路显示到数码管上，而方波信号的频率需先经过测算才能得以显示。

关于输入方波频率的计算是由单片机内部定时计数器实现的，是一种软件实现，将在后续章节介绍。计算得到的频率数据，一方面可由存储按键控制是否存储到存储器，这由用户自己决定，另一方面将直接显示在数码管上。本设计采用的是常用的四位数码管，要使频率能正常显示急需要硬件连接相关知识，也需要软件编程。

图 1 中的指示灯有两个，一是红色的超量程报警灯，当所测的频率大于 9999Hz 时此灯会亮起提示用户，此频率该频率计不可测，测得的值是不准确的，此时，数码管四位将全部消隐。二是黄色的查询灯，当按下查询按钮（分查询加和查询减）时此灯亮起，提示用户当前处于查询状态，数码管上显示的频率数值不是当前信号下的频率值。

在此方案中，原理图部分由 AD 软件绘制，详见附件一：硬件原理图。软件部分的 C 语言程序的编

译、调试由 keil 软件实现。具体程序包括 C 语言的主程序，其中包含近乎所有需要的程序，另一部分是外接的存储器 2402 的.h 文件和.c 文件，附件三：程序清单贴出了 C 语言主程序和部分必要的 2402C 程序。另一方面，Keil 软件可生成了.hex 文件供联机调试时使用。硬件的联机调试由 Protues 软件实现，Proteus 软件所搭建的仿真原理图是基于 AD 绘制的硬件原理图搭建的，此图可在附件二：仿真原理图中详细查看。

上述所有描述，将在后续章节逐一详细介绍。

3 系统硬件设计

3.1 晶振、复位电路

晶振电路和复位电路是单片机最基本的必要电路，结构形式较为固定，如图 2 所示。

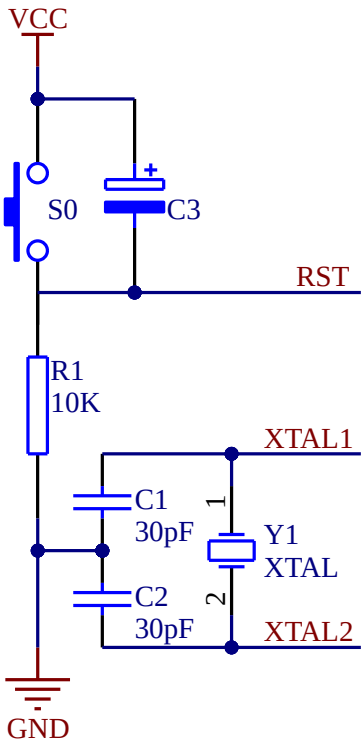


图 2 晶振、复位电路原理图

晶振，全称是石英晶体振荡器，是一种高精度和高稳定度的振荡器。通过一定的外接电路来，可以生成频率和峰值稳定的正弦波。而单片机在运行的时候，需要一个脉冲信号，作为自己执行指令的触发信号，可以简单的想象为：单片机收到一个脉冲，就执行一次或多次指令。由于晶体自身的特性致使这两个频率的距离相当的接近，在这个极窄的频率范围内，晶振等效为一个电感，所以只要晶振的两端并联上合适的电容它就会组成并联谐振电路。这个并联谐振电路加到一个负反馈电路中就可以构成正弦波振荡电路，由于晶振等效为电感的频率范围很窄，所以即使其他元件的参数变化很大，这个振荡器的频率也不会有很大的变化。晶振电路都是在一个反相放大器的两端接入晶振，再有两个电容分别接入到晶振的两端，另一个电容则接地，这两个电容串联的电容量就等于负载电容。

复位电路是一种用来使电路恢复到起始状态的电路设备，它的操作原理与计算器有着异曲同工之妙，只是启动原理和手段有所不同。复位电路，就是利用它把电路恢复到起始状态。就像计算器的清零按钮的作用一样，以便回到原始状态，重新进行计算。单片机的复位有上电复位、手动复位，本设计采用的是手动复位，复位按钮是图 2 中的 S0 按键。

在单片机的使用手册上可以查到对应引脚，按照原理图硬件相连即可。

3.2 计数显示电路

计数显示电路主要突出在显示部分，计数功能主要是由软件实现，后续将会介绍。

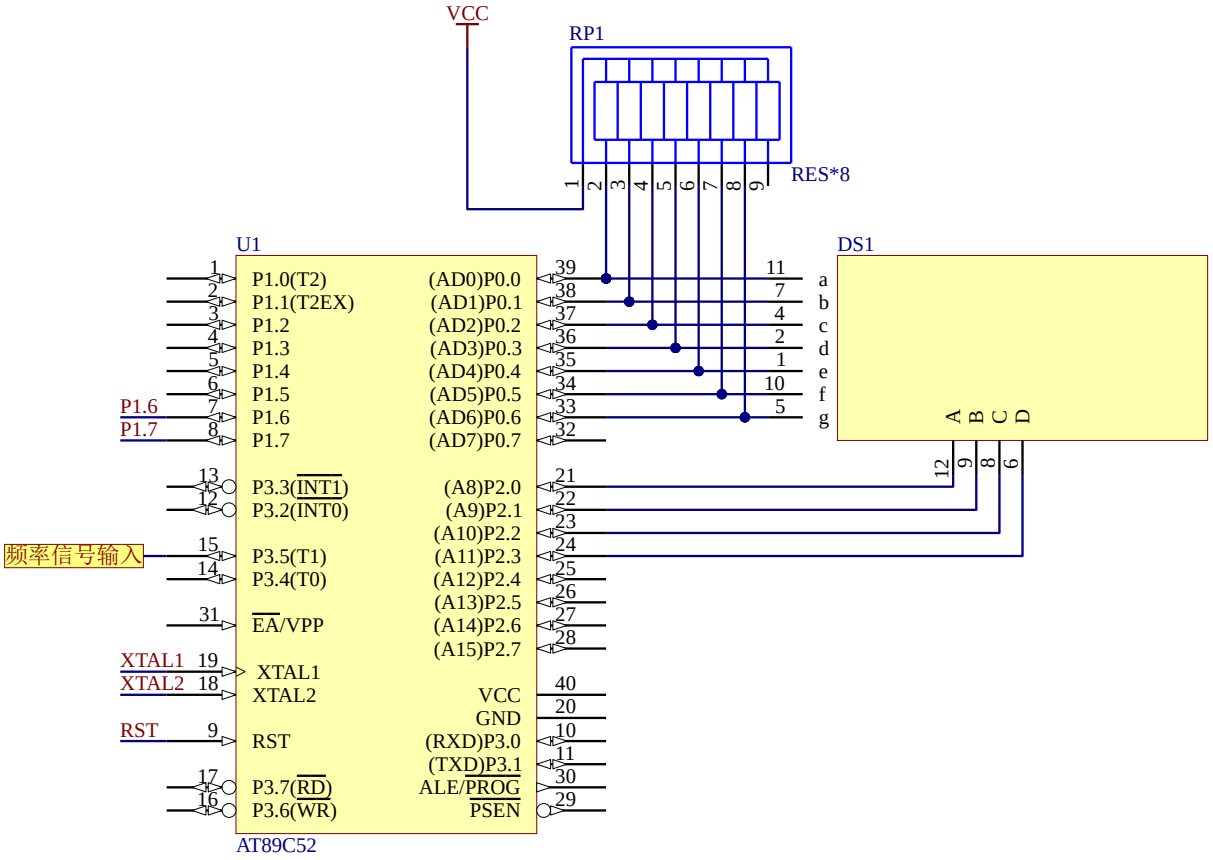


图 3 计数显示电路原理图

如图 3 所示在硬件连接上，共阳极数码管的 a、b、c、d、e、f、g 连接到单片机的 P0 口，一定要将频率数据送到 P0 口才可能正常显示，另外需要注意的是，A、B、C、D 四个引脚代表由高到低四位，比如 B 脚是高电平则此时显示第二位即十进制的百位，且 P0 口的数据必须是原始数据的第二位对应的管码。

这是因为在多位 LED 显示时，为了简化硬件电路，通常将所有位的段选线相应地并联在一起，由一个 8 位 I/O 口控制，形成段选线的多路复用。而各位的共阳极或共阴极分别由相应的 I/O 线控制，实现各位的分时选通。图 3 所示为一个 4 位 7 段 LED 动态显示器电路原理图。其中段选线占用一个 8 位 I/O 口，而位选线占用个 4 位 I/O 口。由于各位的段选线并联，管码的输出对各位来说都是相同的。因此，同一时刻，如果各位选都处于选通状态的话，4 位 LED 将显示相同的字符。若要各位 LED 能够显示出与本位相应的显示字符，就必须采用扫描显示方式，即在某一时刻，只让某一位的位选线处于选通

状态，而其他各位的位选线处于关闭状态，同时，段选线上输出相应位要显示字符的段码。这样同一时刻，4 位 LED 中只有选通的那一位显示字符，而其他三位则是熄灭的。同样，在下一时刻，只让下一位的位选线处于选通状态，而其他各位的位选线处于关闭状态，同时，在段选线上输出相应位将要显示字符的段码，则同一时刻，只有选通位显示出相应的字符，而其他各位则是熄灭的。如此循环下去，就可以使各位显示出将要显示的字符，虽然这些字符是在不同时刻出现的，而同一时刻，只有以为显示，其他各位熄灭，但由于 LED 显示器的余辉和人眼的视觉暂留作用，只要每位显示间隔足够短，则可造成多为同时亮的假象，达到同时显示的目的。

LED 不同位显示的时间间隔不能太短，因为发光二极管从导通到发光有一定的延时，导通时间太短，发光太弱人眼无法看清。但也不能太长，因为毕竟要受限于临界闪烁频率，而且时间越长，占用 CPU 时间也越多。另外，显示位增多，也将占用大量的 CPU 时间，因此动态显示实质是以牺牲 CPU 时间来换取元件的减小。

此频率计共阳极数码管的管码如表 1 所示。

显示	消隐	0	1	2	3	4	5	6	7	8	9
管码	0xff	0xc0	0xf9	0xa4	0xb0	0x99	0x92	0x82	0xf8	0x80	0x90

表 1 共阳极数码管管码

1.3 存储查询电路

为保证掉电数据不丢失，故外接 24C02C 作为外部存储器而不使用单片机内部的存储器。

AT24C02 是美国 ATMEL 公司的低功耗 CMOS 串行 EEPROM，它是内含 256×8 位存储空间，具有工作电压宽（2.5～5.5V）、擦写次数多（大于 10000 次）、写入速度快（小于 10ms）等特点。AT24C02 的 1、2、3 脚是三条地址线，用于确定芯片的硬件地址。在 AT89C51 试验开发板上它们都接地，第 8 脚和第 4 脚分别为正、负电源。第 5 脚 SDA 为串行数据输入/输出，数据通过这条双向 I2C 总线串行传送，在 AT89C51 试验开发板上和单片机的 P3.5 连接。第 6 脚 SCL 为串行时钟输入线，在 AT89C51 试验开发板上和单片机的 P3.6 连接。SDA 和 SCL 都需要和正电源间各接一个 5.1K 的电阻上拉。第 7 脚需要接地。

24C02 中带有片内地址寄存器。每写入或读出一个数据字节后，该地址寄存器自动加 1，以实现对下一个存储单元的读写。所有字节均以单一操作方式读取。为降低总的写入时间，一次操作可写入多达 8 个字节的数据。

欲执行存储功能，只需按存储按键 S1 即可，单片机会将当前显示的频率存储到存储器 24C02 中。存储多个数据只需多次按存储键即可。查询键分为两类，按照存储地址分为查询加和查询减。用查询功能时，一定是先按下查询减，否则显示的一定是 5535，这是因为在未写入数据时，每个单元默认存储的是 0xff。所以当查询过程中遇到 5535 这个特殊数字时一定要甄别其是当前频率还是查询到了未写入数据的单元。

该功能的硬件连接如图 4 所示。

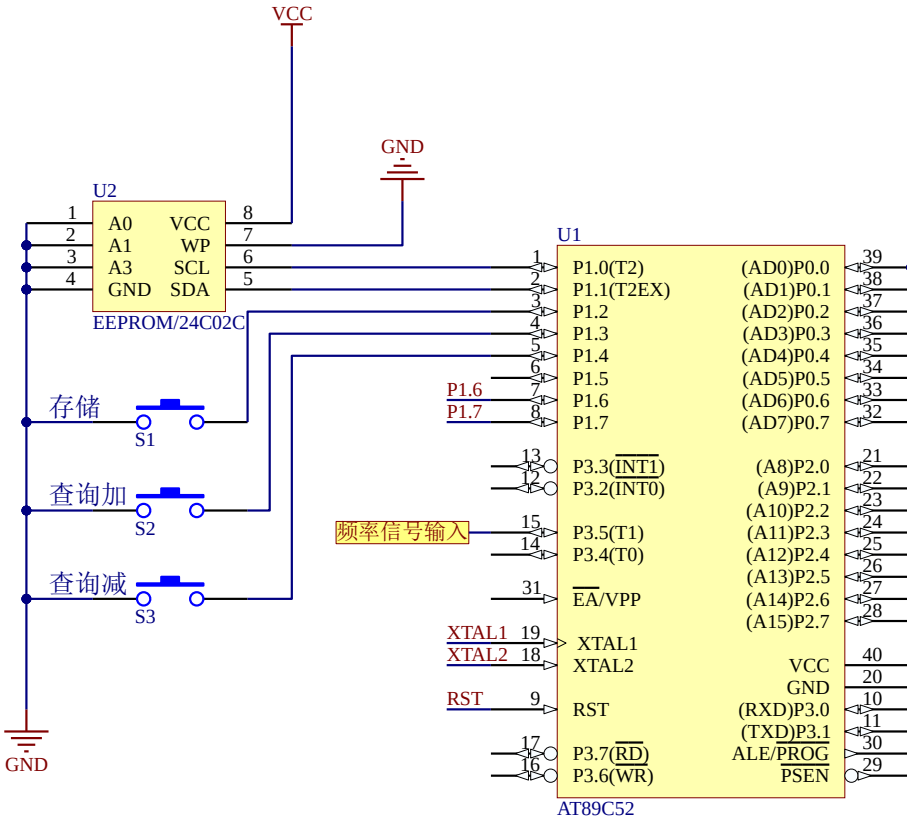


图 4 存储查询电路原理图

3.4 指示灯电路

该部分电路比较简单，此处不做过多说明，电路原理图见附件一：硬件原理图，在原理图中发光二极管的显示，低电平有效。

对于超量程报警灯，使用 if 判断语句，当频率值大于 9999 将其连接的 P1.6 置为 0 即可，默认情况下是 1。对于查询灯，与之类似，与之连接的 P1.7 默认为 1，在使用查询功能时将其置 0 即可。

4 系统软件设计

51 单片机的编程语言常用的有二种，一种是汇编语言，一种是 C 语言。汇编语言的机器代码生成效率很高但可读性却并不强，复杂一点的程序就更是难读懂，而 C 语言在大多数情况下其机器代码生成效率和汇编语言相当，但可读性和可移植性却远远超过汇编语言，而且 C 语言还可以嵌入汇编来解决高时效性的代码编写问题。因此在频率计的程序设计过程中使用的 C 语言，使用 C 语言肯定要使用到 C 编译器，以便把写好的 C 程序编译为机器码，这样单片机才能执行编写好的程序。Keil 是众多单片机应用开发软件中优秀的软件之一，它支持众多不同公司的 MCS51 架构的芯片，它集编辑，编译，仿真等于一体，同时还支持，PLM，汇编和 C 语言的程序设计，它的界面和常用的微软 VC++的界面相似，界面友好，易学易用，在调试程序，软件仿真方面也有很强大的功能。

4.1 频率测算

频率的测量实际上就是在 1s 时间内对信号进行计数，计数值就是信号频率。用单片机设计频率计通常采用两种办法，第一种方法是使用单片机自带的计数器对输入脉冲进行计数；第二种方法是单片机外部使用计数器对脉冲信号进行计数，计数值再由单片机读取。第一种方法的好处是设计出的频率计系统结构和程序编写简单，成本低廉，不需要外部计数器，直接利用所给的单片机最小系统就可以实现。这种方法的缺陷是受限于单片机计数的晶振频率，输入的时钟频率通常是单片机晶振频率的几分之一甚至是几十分之一，在本次设计使用的 AT89C52 单片机，由于检测一个由“1”到“0”的跳变需要两个机器周期，前一个机器周期测出“1”，后一个周期测出“0”。故输入时钟信号的最高频率不得超过单片机晶振频率的二十四分之一。第二种方法的好处是输入的时钟信号频率可以不受单片机晶振频率的限制，可以对相对较高频率进行测量，但缺点是成本比第一种方法高，设计出来的系统结构和程序也比较复杂。由于成本有限，本次设计中采用第一种方法，因此输入的时钟信号最高频率不得高于 12MHz/24=500KHz。对外部脉冲的占空比无特殊要求。

本设计使用的是单片机内部的 T0、T1 定时计数器，T0 作为定时器，T1 作为计数器。程序详见附件三：程序清单，体现在中断服务程序中，此处给出程序流程图如图 5 所示。

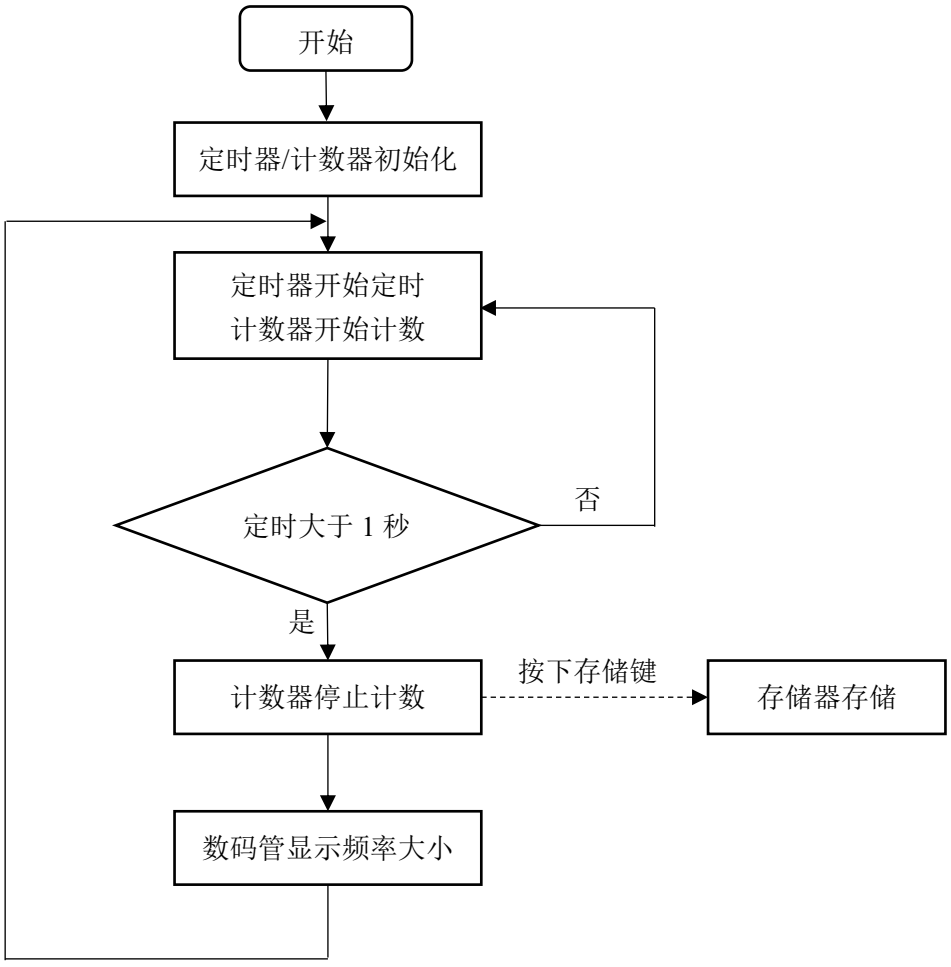


图 5 频率计算程序流程图

4.2 数码管显示

数码管显示部分的硬件部分 3.2 章节已经做了说明。译码部分即将数字译成数码管管码的依据是表 1，程序方面，由于 C 语言数组的索引是从 0 开始的。所以我们可以构建一个长度为 10 的数组，索引对应其管码，详见附件三：程序清单中的主程序中的 disp 函数。

数码管显示部分的程序流程图如图 6 所示。

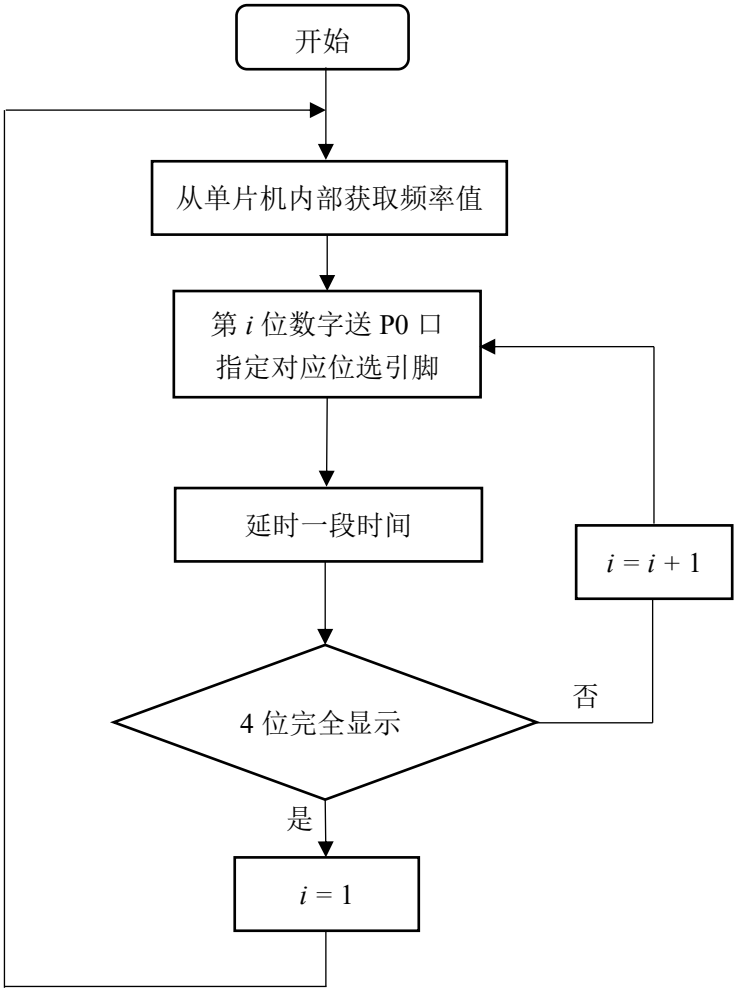


图 6 数码管显示程序流程图

4.3 存储查询

存储查询功能的实现需要借助外部存储器 24C02，它的 API 包含在 2402.c 中，由于篇幅限制，附件中贴出的仅为本设计所用到的函数。WriteSet 需要指定写入数据的值和欲写入的地址，即可将数据写入到外部存储器 24C02 中，而 ReadSet 需要指定欲读取数据的地址即可将数据读出，接下来调用主程序中已封装好的显示函数 disp，即可将频率数据显示在数码管。需要注意的是地址需要连续地选取，一来避免资源浪费，二来在查询时不容易误导用户。程序清单中正是这样做的。

在主程序中使用了选择语句，选择某按键按下时需要执行的功能，三个按键的功能分别是存储、查

询加、查询减，查询加。其中查询加和减指的是地址的加和减，利用了 ReadSet 函数。

5 系统仿真

5.1 Protues 简介

系统的联机调试利用的是 Protues 软件，Proteus 是英国 Labcenter 公司开发的电路分析与实物仿真软件。它运行于 Windows 操作系统上，可以仿真、分析(SPICE)各种模拟器件和集成电路。Proteus 与其它单片机仿真软件不同的是，它不仅能仿真单片机 CPU 的工作情况，也能仿真单片机外围电路或没有单片机参与的其它电路的工作情况。因此在仿真和程序调试时，关心的不再是某些语句执行时单片机寄存器和存储器内容的改变，而是从工程的角度直接看程序运行和电路工作的过程和结果。

该软件的特点是：

① 实现了单片机仿真和 SPICE 电路仿真相结合。具有模拟电路仿真、数字电路仿真、单片机及其外围电路组成的系统的仿真、RS232 动态仿真、I2C 调试器、SPI 调试器、键盘和 LCD 系统仿真的功能；有各种虚拟仪器，如示波器、逻辑分析仪、信号发生器等。

② 支持主流单片机系统的仿真。目前支持的单片机类型有：68000 系列、8051 系列、AVR 系列、PIC12 系列、PIC16 系列、PIC18 系列、Z80 系列、HC11 系列以及各种外围芯片。

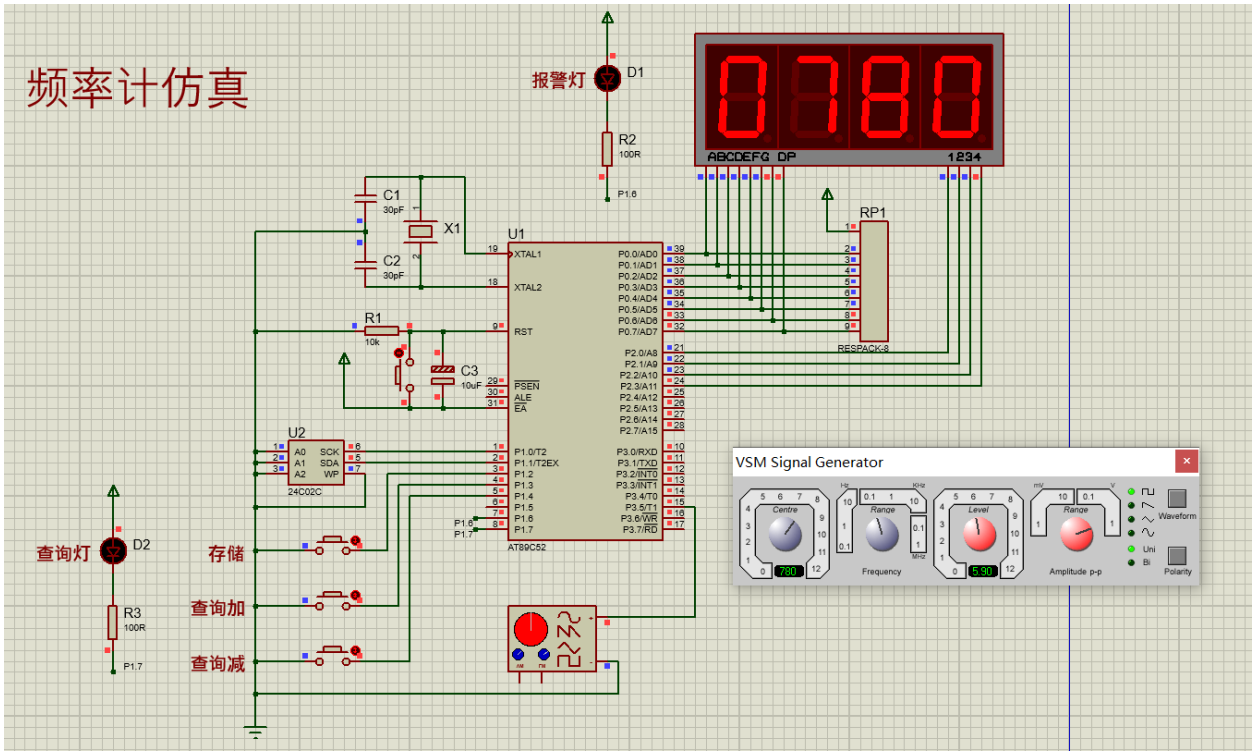
③ 提供软件调试功能。在硬件仿真系统中具有全速、单步、设置断点等调试功能，同时可以观察各个变量、寄存器等的当前状态，因此在该软件仿真系统中，也必须具有这些功能；同时支持第三方的软件编译和调试环境，如 Keil 等软件。

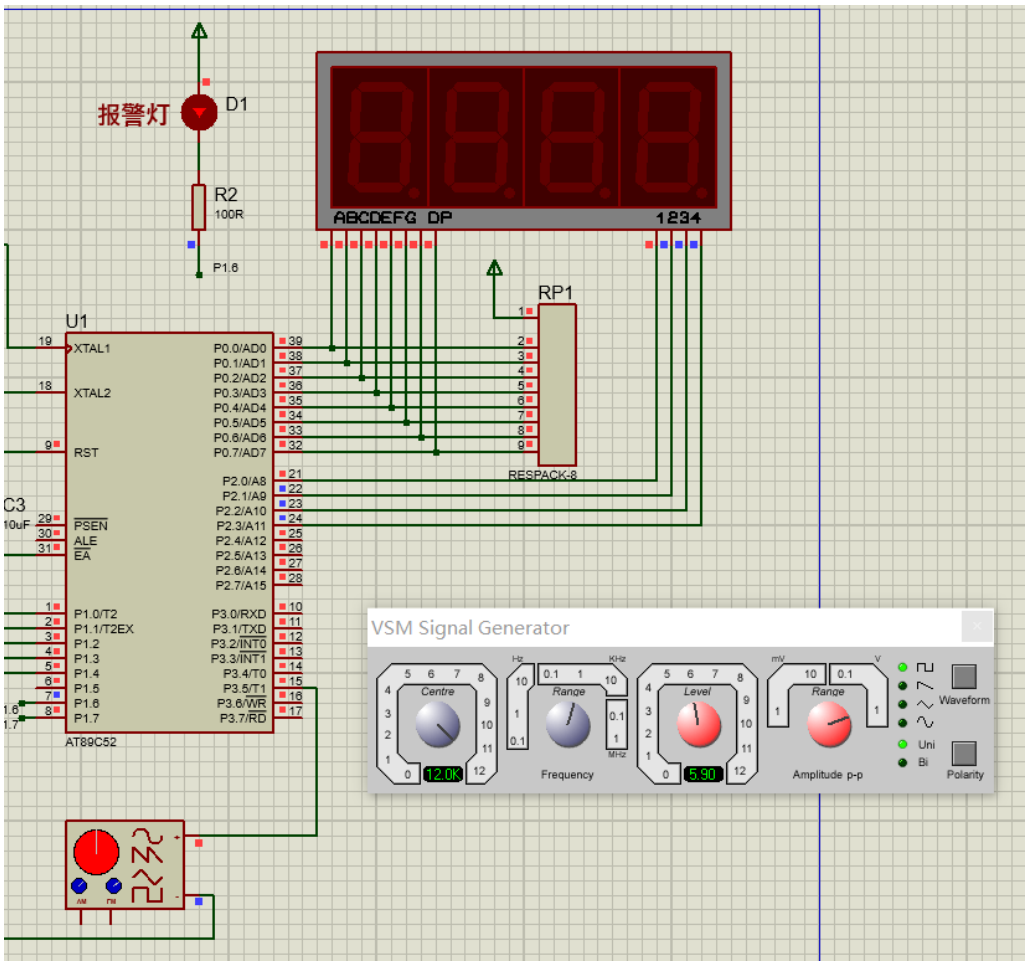
④ 具有强大的原理图绘制功能。总之，该软件是一款集单片机和 SPICE 分析于一身的仿真软件，功能极其强大。

5.2 仿真调试

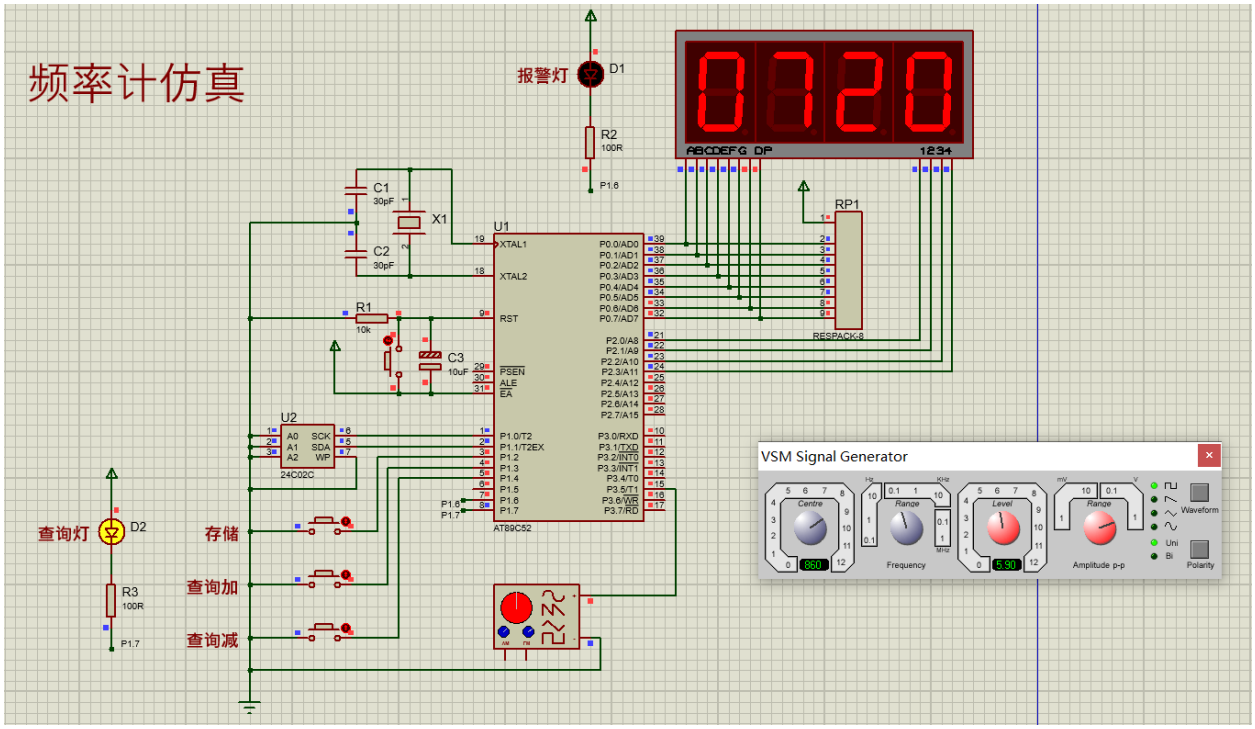
原理图参见附件二：仿真原理图，此处贴出仿真调试的截图，如图所示。在图中利用的是波形发生器模拟待测信号，可以手动调整旋钮以改变频率的大小，图中的信号频率波形发生器指定的频率是 780Hz，在数码管上也显示的是 780Hz。经过多测试验，可以得出在 0~9999Hz 范围内的频率都是可以正确显示的，且误差最多相差不过 2Hz。

当我们将频率设置为超过 9999Hz，图中所示的为 12.0kHz，可见报警灯（红色）亮起，且数码管呈消隐状态，提示用户超出量程了。当使用查询功能是查询灯（黄色）亮起，提示用户此时数码管显示的频率不是当前待测信号的频率。查询模式下需要先按下查询减，显示的是最近一次保存的频率数据，以此类推。





如下图所示，图中当前信号的频率是 860Hz，我们使用了查询功能，查询灯被点亮，按下查询减后显示的是 720Hz，这恰好是上一次的频率值。经过多次实验，发现这一功能是可以完美执行的。



整个软硬件完美运行，实现了任务书的要求，并且扩展了将频率上限提高了一位。缺陷在于并未设计放大整形电路，不能直接测量任意信号的频率。即便任务书没有提及，但这使得本频率计要想广泛使用，必须加一个放大整形器件，或者搭建相应电路。此频率计作为实验室测量较为标准的方波的频率还是着实有效的。

6 心得体会

本次课程设计因为疫情影响不能在学校实验室做，使得同学与同学，老师与老师交流起来不是很方便。我上网查找了许多资料后一开始感觉无从下手，后来渐渐的有了思路。经过这次训练，我对开发单片机应用产品的流程有了大致了解，为以后工作打下了坚实的基础。

本次课程设计，我对书本上学到的理论知识进一步复习加深理解，特别是定时器和计数器的使用，单片机中 T0 和 T0 功能强大，值得学习者深入学习。另外 AD 软件的使用熟练度也加深了，使用过程中有些元件很难找到，我尝试着自己动手画封装，最终完成了原理图的绘制。

在写程序方面一定要注意代码的可读性，添加必要的注释，使用缩进能大大提高代码的美感。而且需要注意一些命名规范，汉语拼音式的命名就显得不那么舒适。另一方面就是要使用分支代替嵌套，嵌套层数多了也会对代码可读性造成影响。

Protues 软件可以提供虚拟实验环境，大大降低了学习成本，而且不用担心损坏硬件。但是，仿真软件毕竟是理想化的，很难与实际情况作对比，做实物时会遇到的一些问题，仿真软件是模拟不出的。所以仿真到实物还有一段路要走，后续回到学校应该注意这方面情况，多在实物上练习使用单片机，加强对单片机的理解和使用。

参考文献

[1] 李彦秀.基于 STC8A 系列单片机的高精度频率计设计[J].工业控制计算机,2019,32(6):125-126

[2] 王玉香,张喜红.基于 Protues 的频率计教学案例设计[J].高师理科学刊,2019,39(11):95-98

[3] 赵迎会,周万顺,陈威志.基于 51 单片机的数字频率计的设计与实现[J].内江科技,2020(5):54-55

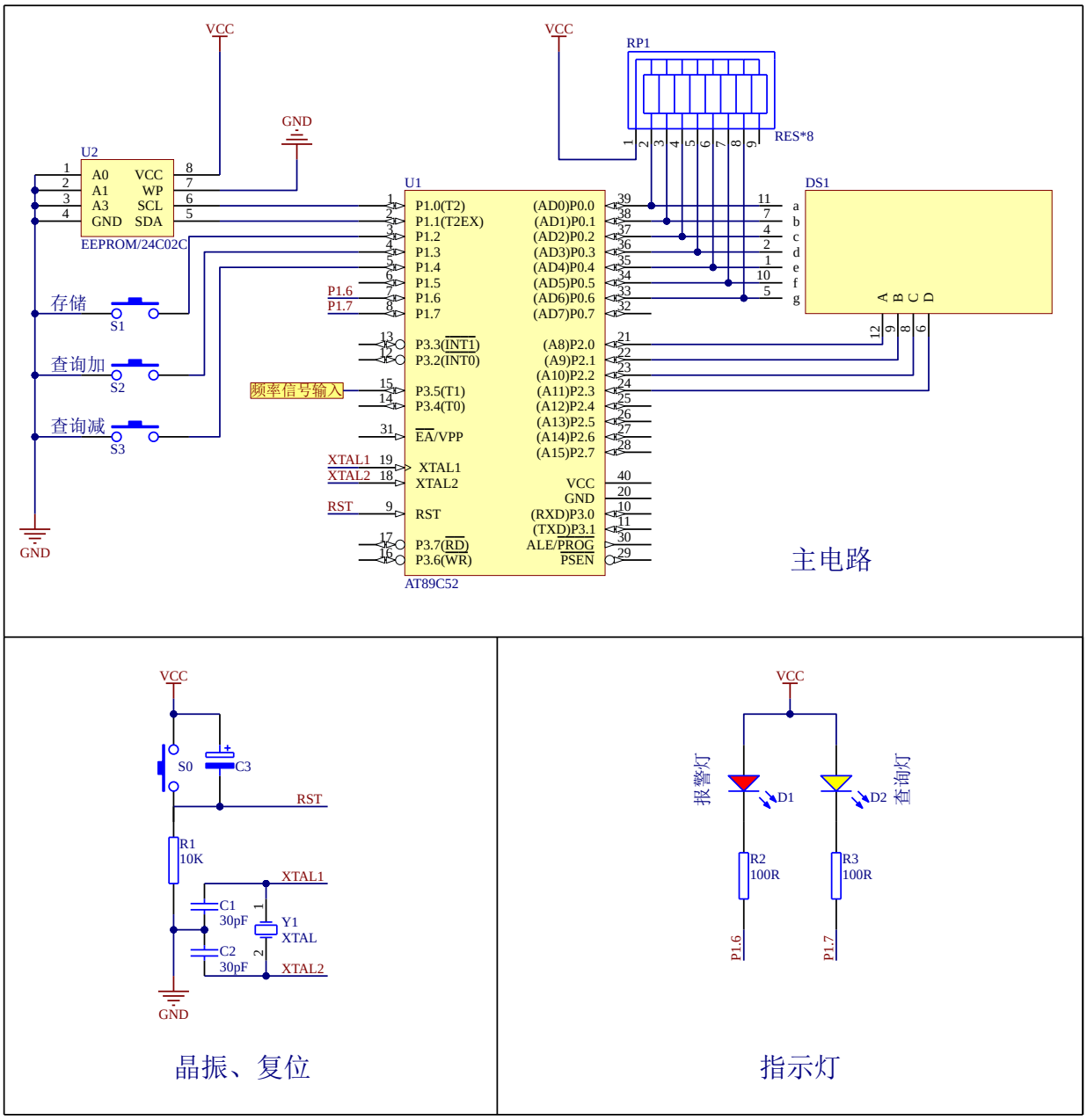
[4] 蔡美琴.MCS-51 系列单片机系统及应用[M].高等教育出版社,2004,2.

[5] 康华光.皱筹彬.电子技术基础(数字部分)[M].高等教育出版社,2000

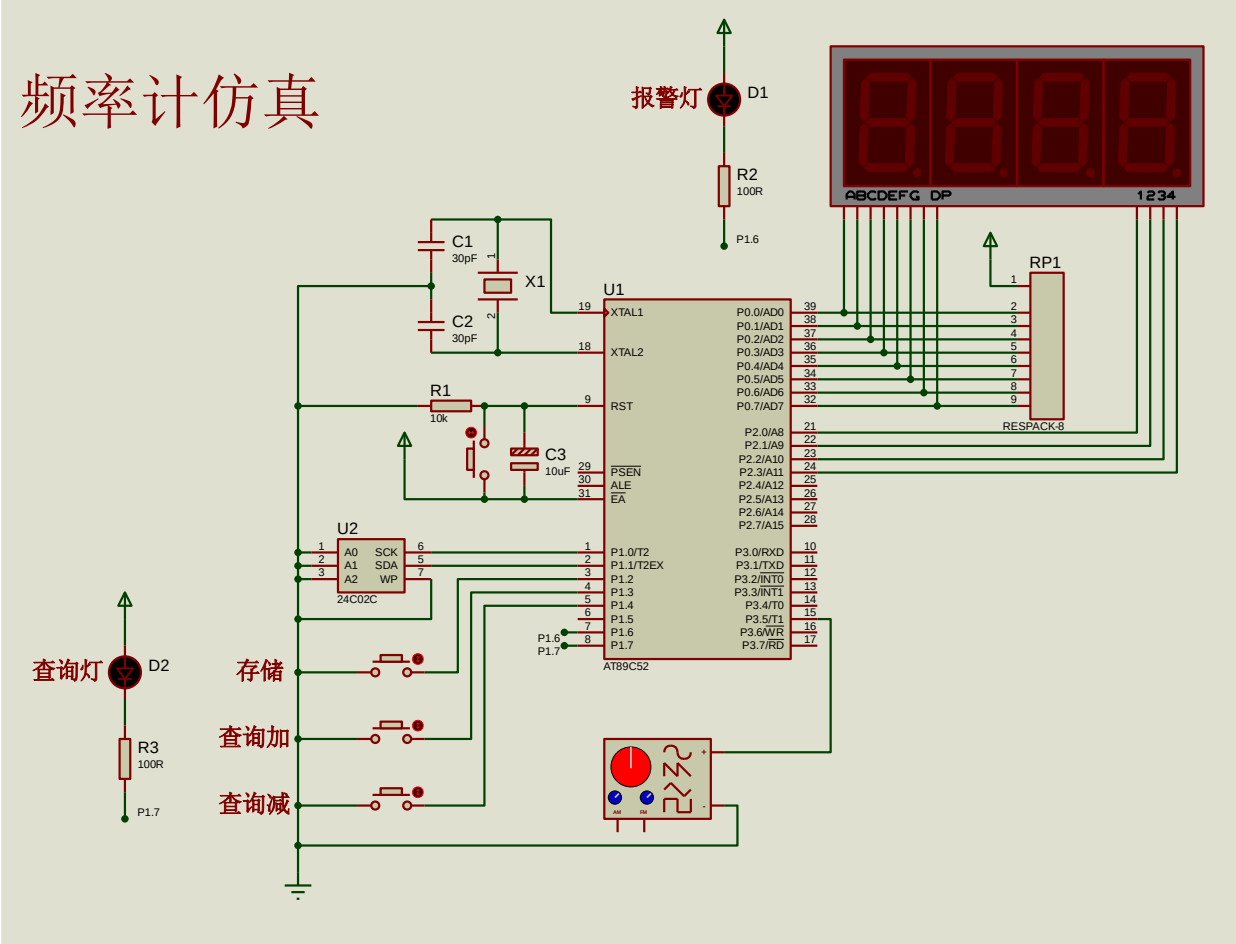
[6] 纪宗南.单片机外围器件实用手册[M].北京航空航天大学出版社,1999.

附件

附件一：硬件原理图


VCC
GND
U2
A0
A1
A3
GND
VCC
WP
SCL
SDA
EEPROM/24C02C
存储
查询加
查询减
S1
S2
S3
U1
P1.0(T2)
P1.1(T2EX)
P1.2
P1.3
P1.4
P1.5
P1.6
P1.7
P3.3(INT1)
P3.2(INT0)
频率信号输入
P3.5(T1)
P3.4(T0)
EA/VPP
XTAL1
XTAL2
RST
P3.7(RD)
P3.6(WR)
(AD0)P0.0
(AD1)P0.1
(AD2)P0.2
(AD3)P0.3
(AD4)P0.4
(AD5)P0.5
(AD6)P0.6
(AD7)P0.7
(A8)P2.0
(A9)P2.1
(A10)P2.2
(A11)P2.3
(A12)P2.4
(A13)P2.5
(A14)P2.6
(A15)P2.7
VCC
GND
(RXD)P3.0
(TXD)P3.1
ALE/PROG
PSEN
AT89C52
RP1
RES*8
DS1
主电路
S0
C3
R1
10K
C1
30pF
C2
30pF
Y1
XTAL
晶振、复位
报警灯
查询灯
D1
D2
R2
100R
R3
100R
指示灯

附件二：仿真原理图


频率计仿真
报警灯
D1
R2
100R
P1.6
ABCDEFG DP
1234
RP1
RESPACK-8
C1
30pF
X1
C2
30pF
U1
XTAL1
XTAL2
RST
PSEN
ALE
EA
P0.0/AD0
P0.1/AD1
P0.2/AD2
P0.3/AD3
P0.4/AD4
P0.5/AD5
P0.6/AD6
P0.7/AD7
P2.0/A8
P2.1/A9
P2.2/A10
P2.3/A11
P2.4/A12
P2.5/A13
P2.6/A14
P2.7/A15
P1.0/T2
P1.1/T2EX
P1.2
P1.3
P1.4
P1.5
P1.6
P1.7
P3.0/RXD
P3.1/TXD
P3.2/INT0
P3.3/INT1
P3.4/T0
P3.5/T1
P3.6/WR
P3.7/RD
AT89C52
R1
10k
C3
10uF
U2
A0
A1
A2
SCK
SDA
WP
24C02C
查询灯
D2
R3
100R
P1.7
存储
查询加
查询减

附件三：程序清单

主程序 main.c

```
1     #include <reg52.h>
2     #include <intrins.h>
3     #include "2402.h"
4
5     #define uchar unsigned char
6     #define uint unsigned int
7
8     //时间(s)，地址，频率
9     uint second， addr， freq;
10    bit sear;                    //逻辑值，表示是否正在查询
11
12    //存储的频率数组，将频率按位转换成数码管管码
13    uchar freqs[4];
14
15    //共阳极数码管码管码
16    uchar code segca[] =
17    {0xc0， 0xf9， 0xa4， 0xb0， 0x99，
18       0x92， 0x82， 0xf8， 0x80， 0x90};
19
20    sbit wled = P1^6;          //超量程报警灯
21    sbit sled = P1^7;          //查询灯
22
23    //声明数码管显示位的控制位
24    sbit w1 = P2^0;
25    sbit w2 = P2^1;
26    sbit w3 = P2^2;
27    sbit w4 = P2^3;
28
29    //声明三个按键
30    sbit key1 = P1^2;          //存储键
31    sbit key2 = P1^3;          //查询加
32    sbit key3 = P1^4;          //查询减
33
34    //声明函数
35    void delay(uchar ms);
36    void key();
37    void disp();
38    void read();
39    void timer0();
40
41    /************************************************
```

```
42      函数功能：主函数
43      ************************************************/
44      void main()
45      {
46          TMOD = 0x51;        //定时器设置
47          EA = 1;             //总中断
48          ET0 = 1;
49          TR0 = 1;
50          ET1 = 1;
51          TR1 = 1;
52          while(1)
53          {
54              sled = ~sear;   //查询灯灭
55              if(sear == 0)
56                  disp();
57              else
58                  read();
59              key();
60          }
61      }
62
63      /************************************************
64      函数功能：延时一段时间
65      入口参数：ms
66      ************************************************/
67      void delay(uchar ms)
68      {
69          uchar i ;
70          while(ms--)
71          {
72              for(i = 0; i < 250; i++);
73          }
74      }
75
76      /************************************************
77      函数功能：响应按键
78      ************************************************/
79      void key()
80      {
81          if(key1 == 0)          //保持
82          {
83              delay(5);
84              sear = 0;
85              WriteSet(addr * 2,  freq / 256);       //存高位
```

```
86          WriteSet(addr * 2 + 1,  freq % 256);      //存低位
87          addr++;
88          if(addr == 100) addr = 0;
89          while(!key1);
90       }
91       if(key2 == 0)          //查询加
92       {
93          delay(5);
94          sear = 1;
95          if(addr < 100) addr++;
96          while(!key2);
97       }
98       if(key3 == 0)          //查询减
99       {
100         delay(5);
101         sear = 1;
102         if(addr > 0) addr--;
103         while(!key3);
104      }
105   }
106
107   /*****************************************************
108   函数功能：在数码管上显示频率
109   *****************************************************/
110   void disp()
111   {
112      P0 = freqs[0];
113      w1 = 1;
114      delay(1);
115      w1 = 0;
116      P0 = freqs[1];
117      w2 = 1;
118      delay(1);
119      w2 = 0;
120      P0 = freqs[2];
121      w3 = 1;
122      delay(1);
123      w3 = 0;
124      P0 = freqs[3];
125      w4 = 1;
126      delay(1);
127      w4 = 0;
128   }
129
```

```
130   /*************************************************
131   函数功能：读取所存储的数据
132   *************************************************/
133   void read()
134   {
135      freq = ReadSet(addr * 2) * 256
136         + ReadSet(addr * 2 + 1);
137      freqs[0] = segca[freq % 10000 / 1000];
138      freqs[1] = segca[freq % 1000 / 100];
139      freqs[2] = segca[freq % 100 / 10];
140      freqs[3] = segca[freq % 10];
141      P0 = freqs[0];
142      w1 = 1;
143      delay(1);
144      w1 = 0;
145      P0 = freqs[1];
146      w2 = 1;
147      delay(1);
148      w2 = 0;
149      P0 = freqs[2];
150      w3 = 1;
151      delay(1);
152      w3 = 0;
153      P0 = freqs[3];
154      w4 = 1;
155      delay(1);
156      w4 = 0;
157   }
158
159   /*************************************************
160   函数功能：定时器 0 中断服务程序
161   *************************************************/
162   void timer0() interrupt 1
163   {
164      TH0 = (65536-50000) / 256;
165      TL0 = (65536-50000) % 256;
166      second++;
167      if(second >= 20)   //1 秒到
168      {
169         second = 0;
170         freq = TH1*256 + TL1;     //从计数器中读出频率
171         TH1 = 0;
172         TL1 = 0;
173
```

```
174          wled = 1;       //超量程报警灯灭
175          freqs[0] = segca[freq % 10000 / 1000];
176          freqs[1] = segca[freq % 1000 / 100];
177          freqs[2] = segca[freq % 100/10];
178          freqs[3] = segca[freq % 10];
179          if(freq > 9999)
180          {
181             wled = 0;    //超量程报警灯亮
182             freqs[0] = 0xff;
183             freqs[1] = 0xff;
184             freqs[2] = 0xff;
185             freqs[3] = 0xff;
186       }
187       }
188    }
```

所需的 2402.c 部分程序

```
1      /*************************************************
2      函数功能：向 AT24Cxx 中的指定地址写入数据
3      入口参数：add (储存指定的地址)；dat(储存待写入的数据)
4      *************************************************/
5      void WriteSet(unsigned char addr， unsigned char dat)
6      //在指定地址 addr 处写入数据 WriteCurrent
7      {
8         start();                  //开始数据传递
9         WriteCurrent(OP_WRITE);   //选择要操作的 AT24Cxx 芯片，
10                                  //告知要对其写入数据
11        WriteCurrent(addr);       //写入指定地址
12        WriteCurrent(dat);        //向当前地址写入数据
13        stop();          //停止数据传递
14        delaynms(4);     //1 个字节的写入周期为 1ms， 最好延时 1ms 以
       上
15     }
16
17     /*************************************************
18     函数功能：从 AT24Cxx 中的指定地址读取数据
19     入口参数：set_addr
20     出口参数：x
```

```
21      ************************************************/
22      unsigned char ReadSet(unsigned char set_addr)
23      //在指定地址读取
24      {
25          start();                        //开始数据传递
26          WriteCurrent(OP_WRITE);         //选择要操作的 AT24Cxx 芯片
27                                          //告知要对其写入数据
28          WriteCurrent(set_addr);         //写入指定地址
29          return(ReadCurrent());          //从指定地址读出数据并返回
30      }
```