

## 一、设计目的

《单片机原理及应用》课程设计是一项重要的实践性教育环节，是学生在校期间必须接受的一项工程训练。在课程设计过程中，在教师指导下，应用工程的方法，通过一个简单课题的设计练习，可使学生初步体验单片机应用系统的设计过程、设计要求、完成的工作内容和具体的设计方法，了解必须提交的各项工程文件，也达到巩固、充实和综合运用所学知识解决实际问题的目的。

通过课程设计，应该能加强学生如下能力的培养：

- (1) 独立工作能力和创造力；
- (2) 综合运用专业及基础知识，解决实际工程技术问题的能力；
- (3) 查阅图书资料、产品手册和各种工具书的能力；
- (4) 工程绘图的能力；
- (5) 编写技术报告和编写技术资料的能力。

## 二、设计要求

### 1 总体要求

- (1) 独立完成设计任务
- (2) 绘制系统硬件总框图
- (3) 绘制系统原理电路图
- (4) 制定编写设计方案，编制软件框图，完成详细完整的程序清单和注释
- (5) 制定编写调试方案，编写用户操作使用说明书
- (6) 写出设计工作小结。对在完成以上文件过程所进行的有关步骤如设计思想、指标论证、方案确定、参数计算、元器件选择、原理分析等作出说明，并对所完成的设计作出评价，对自己整个设计工作中经验教训，总结收获和今后研究方向。

### 2 具体要求

本次工程实践的校内部分主要以单片机为基础，进行单片机软件编程，目的

是为了提高学生的软件编程和系统设计能力，整个设计系统包括两个部分，硬件及软件部分，硬件部分已经制作成功，学生只需要掌握其原理和焊接相应的元器件，掌握元器件的辨别和元器件的作用以及应用场所即可，另外对所焊接的电路还需要进行仔细的检查，判断是否有焊接错误的地方或者短路的地方，对出现的异常情况要能够根据现象判别原因，并具备解决问题的能力，从而切实提高学生的硬件电子电路的分析、判断能力。

软件编程是本次课程设计的重要环节。在为期两周的工程实践中，将占据主要时间，完成的软件编程任务主要包括以下几点：

- 1)、熟悉 Keil C51编程平台及相关编程软件
- 2)、编写、调试蜂鸣器、继电器动作、方波程序并进行软硬件联调
- 3)、编写、调试 LED流水灯（循环显示）程序并进行软硬件联调
- 4)、编写、调试键盘扫描子程序并进行软硬件联调
- 5)、编写、调试数码管动态扫描程序并进行软硬件联调
- 6)、基于交通灯的多功能控制设计

其中前五个内容是后两个内容的基础，主要是编制一些子程序，为后继的整个系统设计打下基础。

交通灯控制是在五个课题中选做的一个设计，是一个简单的单片机编程设计，来实现智能交通灯的多功能作用。

### 三、硬件电路的设计及描述

#### (一)硬件总框图及原理图

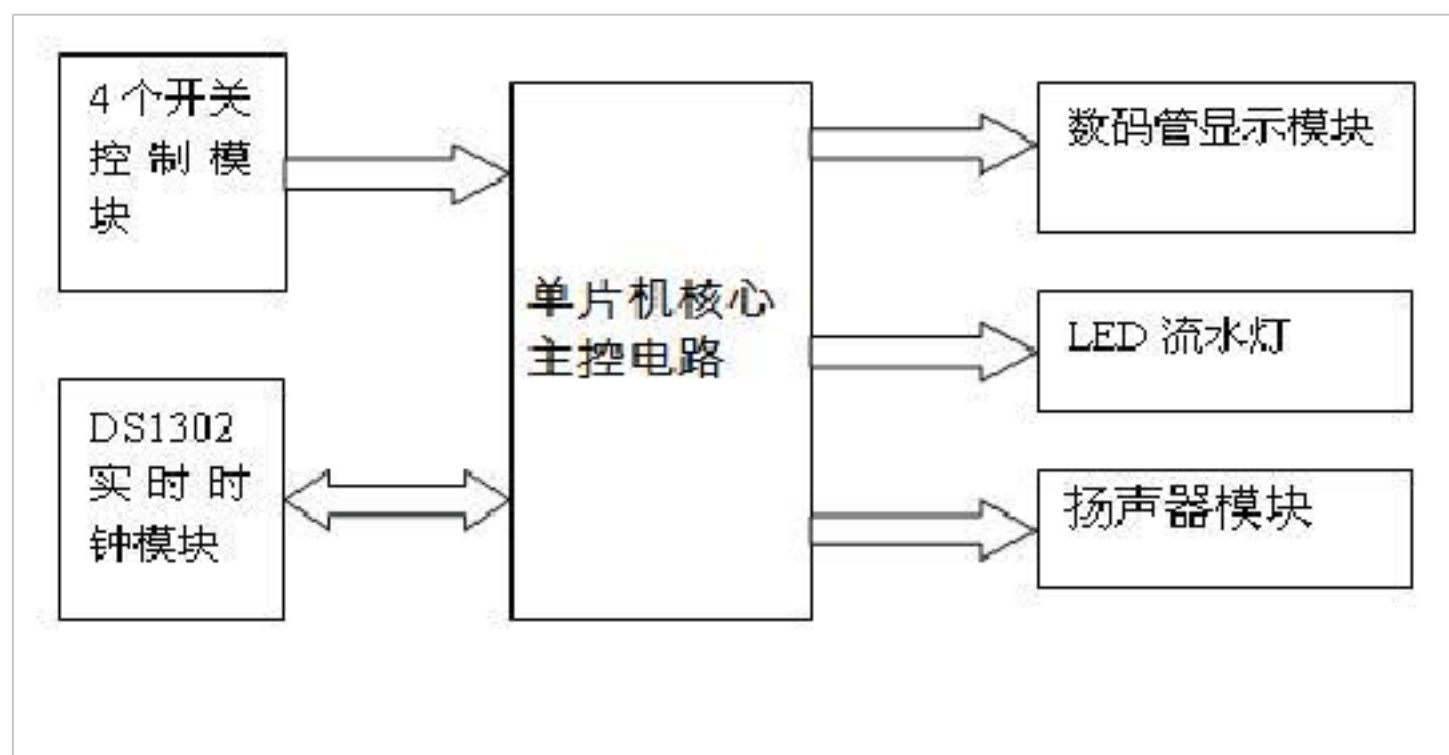


图 1 硬件总框图



存储器，且允许在系统内改写或用编程器编程。另外，AT89C52的指令系统和引脚与 80C52 完全兼容，所以，近几年 AT89C52单片机应用极为广泛。

AT89C52单片机除了含有 8KB的 Flash 存储器外，片内还有 256 B 的 RAM，32 条 I/O 口线，3 个 16 位定时/计算器、6 个中断源、1 个全双工串行口等。同时，与 80C52 一样，有空闲和掉电两种节电运行方式。

AT89C52单片机的结构与 8051 单片机的结构基本相同，不同之处有 CHMOS工艺的节能运行方式、增加了一个 16 位的定时器 T2、片内 RAM 由 128 B 增加为 256 B、ROM 类型属于 Flash 存储器。

AT89C52引脚如图 7 所示：

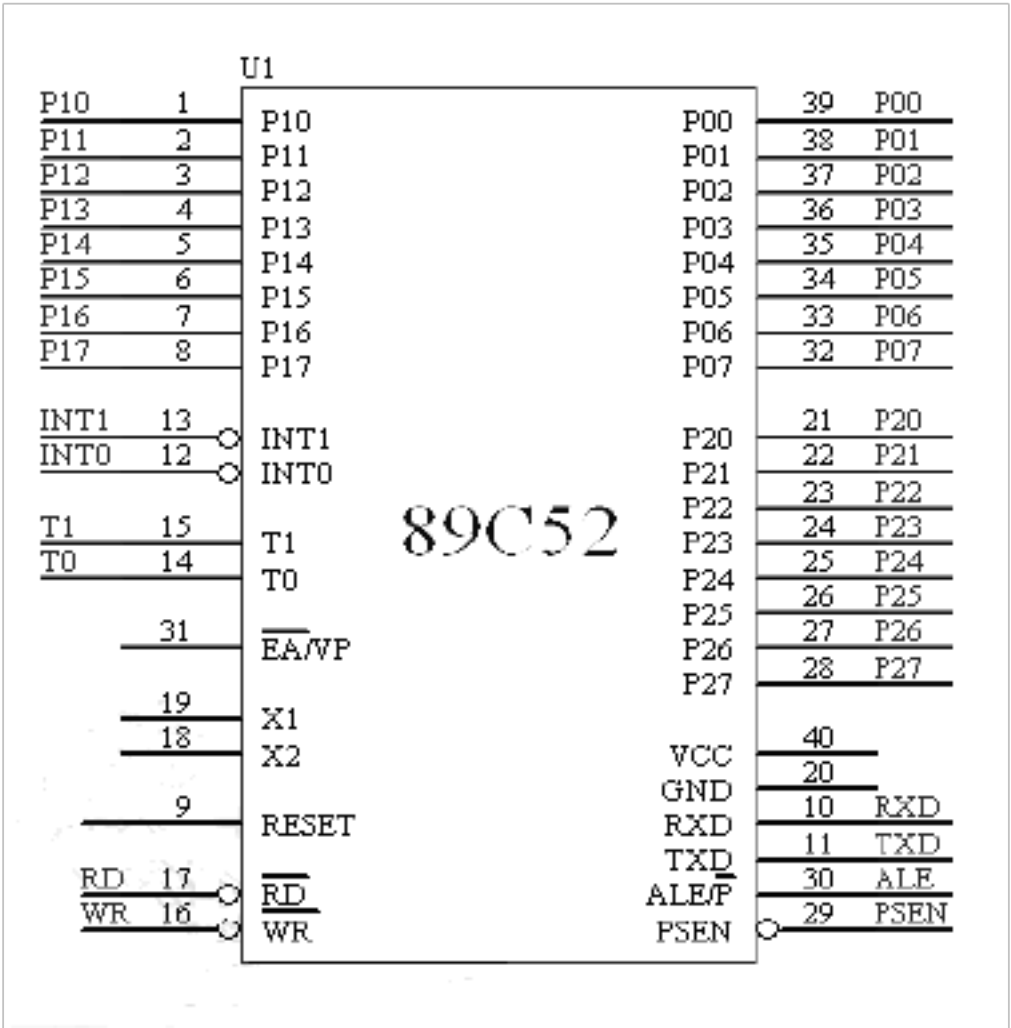


图 7 AT89C52引脚

AT89C52单片机为 40 脚双列直插封装的 8 位通用微处理器，采用工业标准的 C51 内核，在内部功能及管脚排布上与通用的 8xc52 相同，其主要用于会聚调整时的功能控制。功能包括对会聚主 IC 内部寄存器、数据 RAM 及外部接口等功能部件的初始化，会聚调整控制，会聚测试图控制，红外遥控信号 IR 的接收解码及与主板 CPU 通信等。

主要管脚有：XTAL1(19 脚)和 XTAL2(18 脚)为振荡器输入输出端口，

外接 12MHz 晶振。RST/Vpd(9 脚) 为复位输入端口，外接电阻电容组成的复位电路。VCC(40 脚) 和 VSS(20 脚) 为供电端口，分别接+5V 电源的正负端。P0~P3 为可编程通用 I/O 脚，其功能用途由软件定义，在本设计中，P0 端口（32~39 脚）被定义为 N1 功能控制端口，分别与 N1 的相应功能管脚相连接，13 脚定义为 IR 输入端，10 脚和 11 脚定义为 I2C 总线控制端口，分别连接 N1 的 SDAS(18 脚) 和 SCLS(19 脚) 端口，12 脚、27 脚及 28 脚定义为握手信号功能端口，连接主板 CPU 的相应功能端，用于当前制式的检测。

## 2) DS1302 实时时钟模块

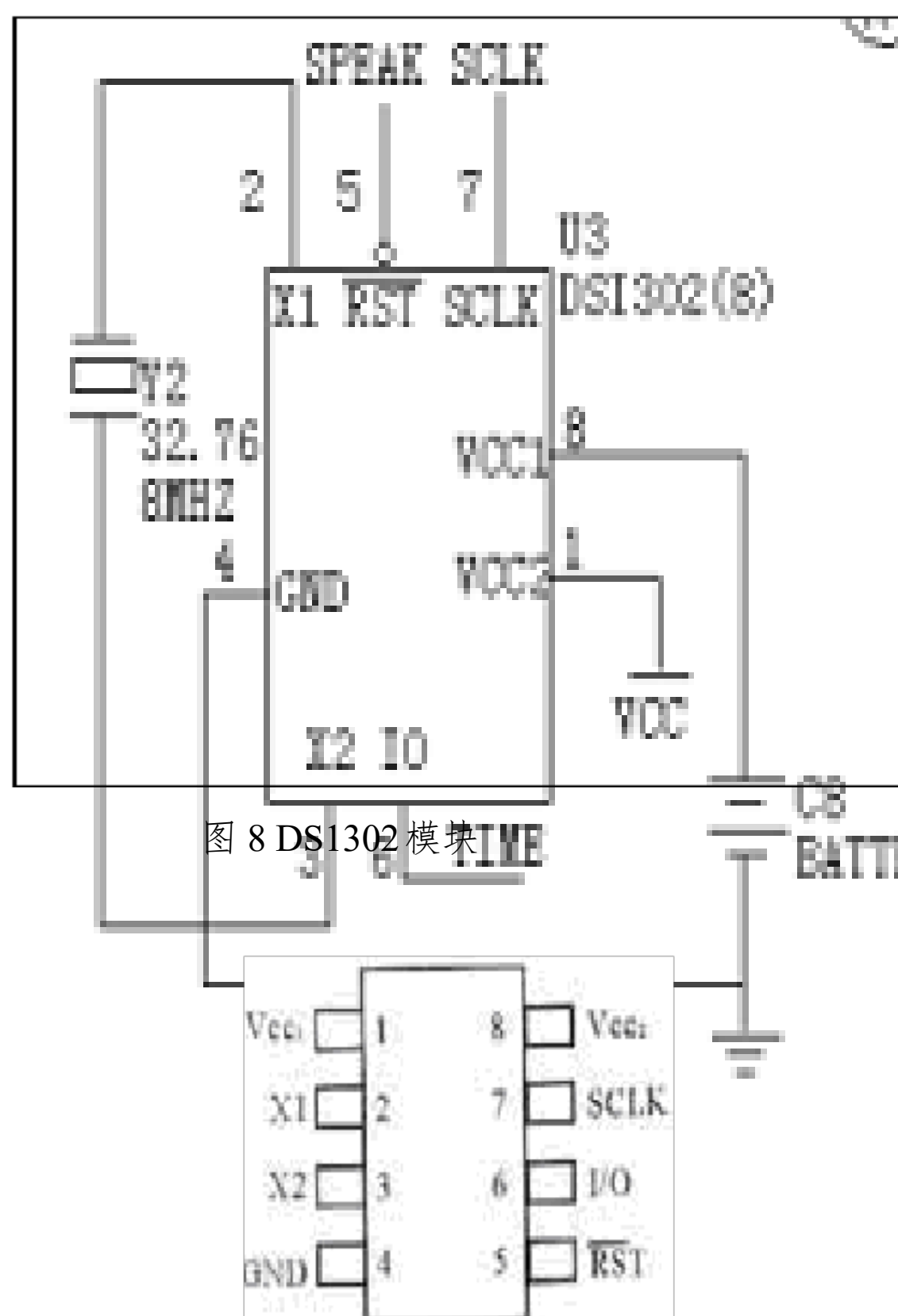


图 8 DS1302 模块

图 9 DS1302 的引脚图

其中 Vcc1 为后备电源，VCC2 为主电源。在主电源关闭的情况下，也能保持

时钟的连续运行。DS1302由 Vcc1 或 Vcc2 两者中的较大者供电。当 Vcc2 大于 Vcc1 + 10mV 时，Vcc2 给 DS1302 供电。当 Vcc2 小于 Vcc1 时，DS1302 由 Vcc1 供电。X1 和 X2 是振荡源，外接晶振。RST 是复位/片选线，通过把 RST 输入驱动置高电平来启动所有的数据传送。

### ① 概述

现在流行的串行时钟电路很多，如 DS1302、DS1307、PCF8485 等。这些电路的接口简单、价格低廉、使用方便，被广泛地采用。本文介绍的实时时钟电路 DS1302 是 DALLAS 公司的一种具有涓细电流充电能力的电路，主要特点是采用串行数据传输，可为掉电保护电源提供可编程的充电功能，并且可以关闭充电功能。采用普通晶振。

### ② 引脚功能及结构

图 8 为 DS1302 的引脚排列，其中 Vcc1 为后备电源，VCC2 为主电源。在主电源关闭的情况下，也能保持时钟的连续运行。DS1302 由 Vcc1 或 Vcc2 两者中的较大者供电。当 Vcc2 大于 Vcc1 + 10mV 时，Vcc2 给 DS1302 供电。当 Vcc2 小于 Vcc1 时，DS1302 由 Vcc1 供电。X1 和 X2 是振荡源，外接晶振。RST 是复位/片选线，通过把 RST 输入驱动置高电平来启动所有的数据传送。RST 输入有两种功能：首先，RST 接通控制逻辑，允许地址/命令序列送入移位寄存器；其次，RST 提供终止单字节或多字节数据的传送手段。当 RST 为高电平时，所有的数据传送被初始化，允许对 DS1302 进行操作。如果在传送过程中 RST 置为低电平，则会终止此次数据传送，I/O 引脚变为高阻态。上电运行时，在  $V_{cc} \geq V_{cc1}$  之前，RST 必须保持低电平。只有在 SCLK 为低电平时，才能将 RST 置为高电。

### ③ DS1302 的控制字节

控制字节的最高有效位(位 7)必须是逻辑 1，如果它为 0，则不能把数据写入 DS1302 中，位 6 如果为 0，则表示存取日历时钟数据，为 1 表示存取 RAM 数据；位 5 至位 1 指示操作单元的地址；最低有效位(位 0)如为 0 表示要进行写操作，为 1 表示进行读操作，控制字节总是从最低位开始输出。

### ④ DS1302 与 CPU 的连接

实际上,在调试程序时可以不加电容器,只加一个 的晶振即可。只是选择晶振时,不同的晶振,误差也较大。另外,还可以在上面的电路中加入 DS18B20 同时显示实时温度。只要占用 CPU 一个口线即可。LCD 还可以换成 LED 还可以使用北京卫信杰科技发展有限公司生产的 10 位多功能 8 段液晶显示模块 LCM101,内含看门狗(WDT)时钟发生器及两种频率的蜂鸣器驱动电路,并有内置显示 RAM,可显示任意字段笔划,具有 3—4 线串行接口,可与任何单片机、IC 接口。功耗低,显示状态时电流为  $2\mu\text{A}$  (典型值),省电模式时小于  $1\mu\text{A}$ ,工作电压为 $\sim$ ,显示清晰。

#### ⑤ 总结

DS1302 存在时钟精度不高,易受环境影响,出现时钟混乱等缺点。DS1302 可以用于数据记录,特别是对某些具有特殊意义的数据点的记录,能实现数据与出现该数据的时间同时记录。这种记录对长时间的连续测控系统结果的分析及对异常数据出现的原因的查找具有重要意义。传统的数据记录方式是隔时采样或定时采样,没有具体的时间记录,因此,只能记录数据而无法准确记录其出现的时间;若采用单片机计时,一方面需要采用计数器,占用硬件资源,另一方面需要设置中断、查询等,同样耗费单片机的资源,而且,某些测控系统可能不允许。

DS1302 慢速充电时钟芯片包括时时钟/日历和 31B 的静态 RAM,经过一个简单的串行接口与单片机通信。DS1302 与单片机的通信仅需 3 根线,即 RST(复位线)、I/O(数据线)、SCLK(串行时钟线)。数据可按每次 1B 或多达 31B 的形式传送到时钟/RAM 或从中送出。

DS1302 在任何数据传送时必须先初始化,把 RST 脚置为高电平,然后把 8 位地址和命令字装入移位寄存器,数据在 SCLK 的上升沿被输入。无论是读周期还是写周期,开始八位指定 40 个寄存器中哪个将被访问到。在开始的 8 个时钟周期,把命令字节装入移位寄存器之后,另外的时钟周期在读操作时输出数据,在写操作时写入数据。时钟脉冲的个数在单字节方式下为 8 加 8,在多字节方式下为 8 加字节数,最大可达 248 字节数。

DS1302 的 RST 是复位/片选线,通过把 RST 输入驱动置高电平来启动所有的



数据传送。**RST**输入有两种功能：首先，**RST**接通控制逻辑，允许地址/命令序列送入移位寄存器；其次，**RST**提供终止单字节或多字节数据的传送手段。当 **RST** 为高电平时，所有的数据传送被初始化，允许对 **DS1302** 进行操作。

如果在传送过程中 **RST**置为低电平，则会终止此次数据传送，**I/O** 引脚变为高阻态。上电运行时，在  $V_{cc} \geq$  之前，**RST**必须保持低电平。只有在 **SCLK**为低电平时，才能将 **RST**置为高电平。**I/O** 为串行数据输入输出端 (双向)。**SCLK**始终是输入端。

3) 数码管显示电路

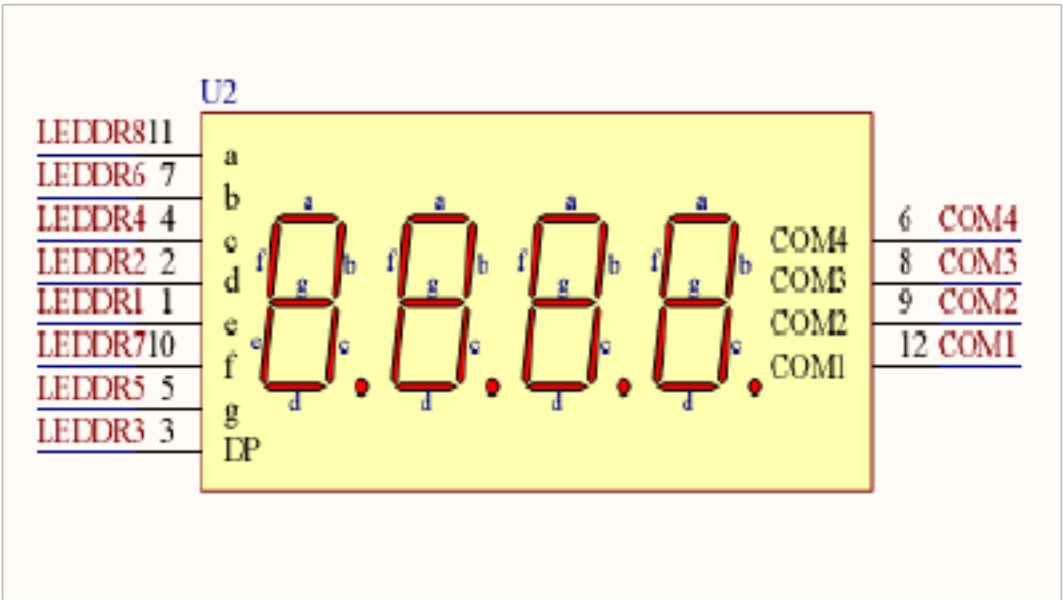


图 6 数码管显示电路

数码显示电路设计中采用四位共阳极扫描型数码管即其公共端接正极，扫描型是指数码管的段选都是并联的。其位选位控制哪一位的数码管亮,在设计中由单片机 **AT89C52**的 **P1** 口控制其位选，以达到动态显示的效果，再通过 **P0** 口，控制其段选以显示相应的数值。

在其位选控制部分,采用了一个 **9012** 型三极管,要求当 **P1** 口输出低电平时，位选成功。动态扫描显示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**是各自独立地受 **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，所以我们可以自行决定何时显示哪一位了。

轮流控制各个显示器的端，使各个显示器轮流点亮。在轮流点亮扫描过程中，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。

） LED流水灯循环显示

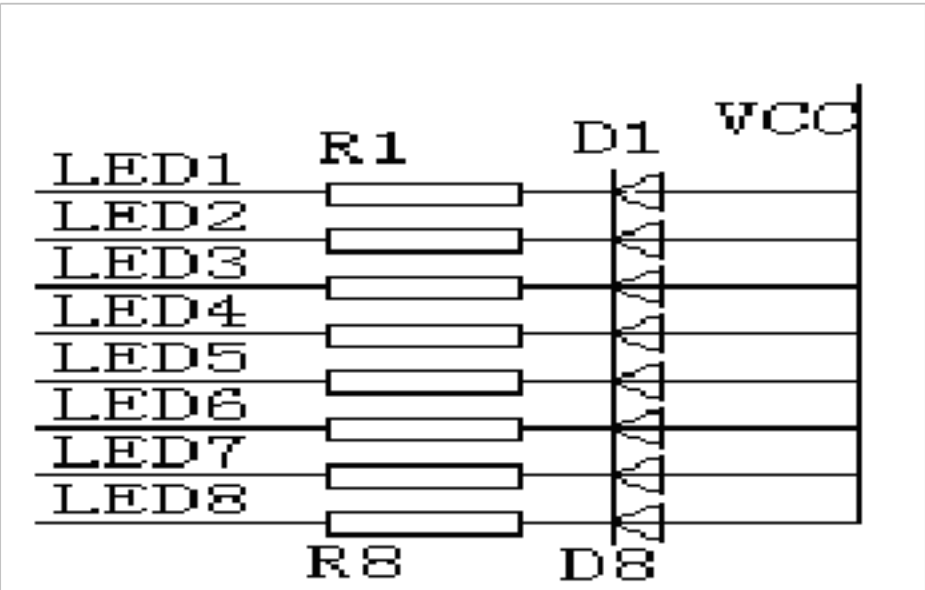


图 3 循环显示电路

要实现此功能，最重要的是设法将 P2 口的引脚变为低电平时，相应的 LED 灯管就会发光。通过修改程序可使得 IO 口电平依次变为低电平后按设定的时间再变为高电平，这样就可以实现流水灯效果。改变延时时间，可以改变流水灯的流动速度。从上面中文表示看来实现单片机流水灯很简单，但是我们不能说你变低，它就变低了。

因为单片机只能接受二进制的 1、0 机器代码。为了让单片机工作，只能将程序写为二进制代码交给其执行。这里的“程序语言”主要有汇编语言和 C 语言两种；在这里我们将“程序语言”通过编译器产生单片机的二进制代码的。前面说到，要想使 LED1 变亮，只需将对应的单片机引脚电平变为低电平就可以了。

流水灯的现象是：P20 低电平 LED1亮、延时、P20 高，P21 低电平 LED2亮、延时、P21 高，P22 低电平 LED3亮、延时、P22 高，P23 低电平 LED4亮、延时、P23 高，P24 低电平 LED5亮、延时、P24 高，P25 低电平 LED6亮、延时、P25 高，P26 低电平 LED7亮、延时、P26 高，P27 低电平 LED8亮、延时、P27 高，返回到开始、程序结束。

#### ）键盘扫描模块

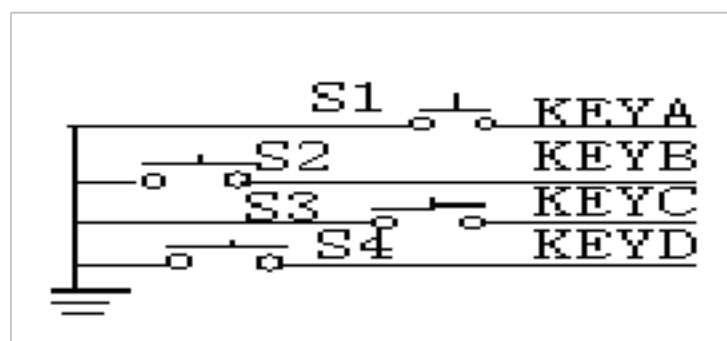


图 4 按键输入电路

本模块四个开关分别用于控制流水灯控制电路，时间显示电路，音乐蜂鸣器电路，交通灯四部分功能。当按下其中一个开关时，该部分电路导通，给单片机发出地址信号，实现其中一项功能的运做，如图 4 所示。

**KEYA**键：其功能是该键按下时，流水灯程序工作；

**KEYB**键：其功能是该键按下时，音频程序工作；

**KEYC**键：其功能是该键按下时，时间日期程序工作；

**KEYD**键：其功能是该键按下时，交通灯程序工作。

#### 6) 蜂鸣器电路

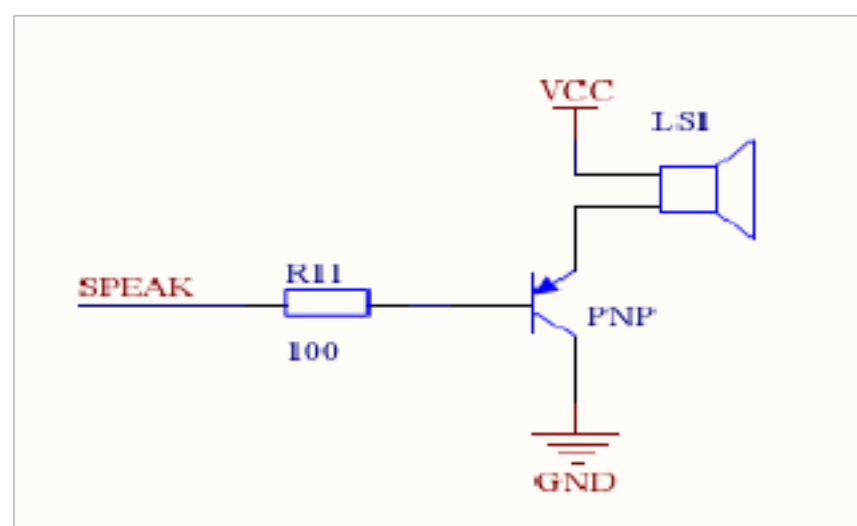


图 5 蜂鸣器电路

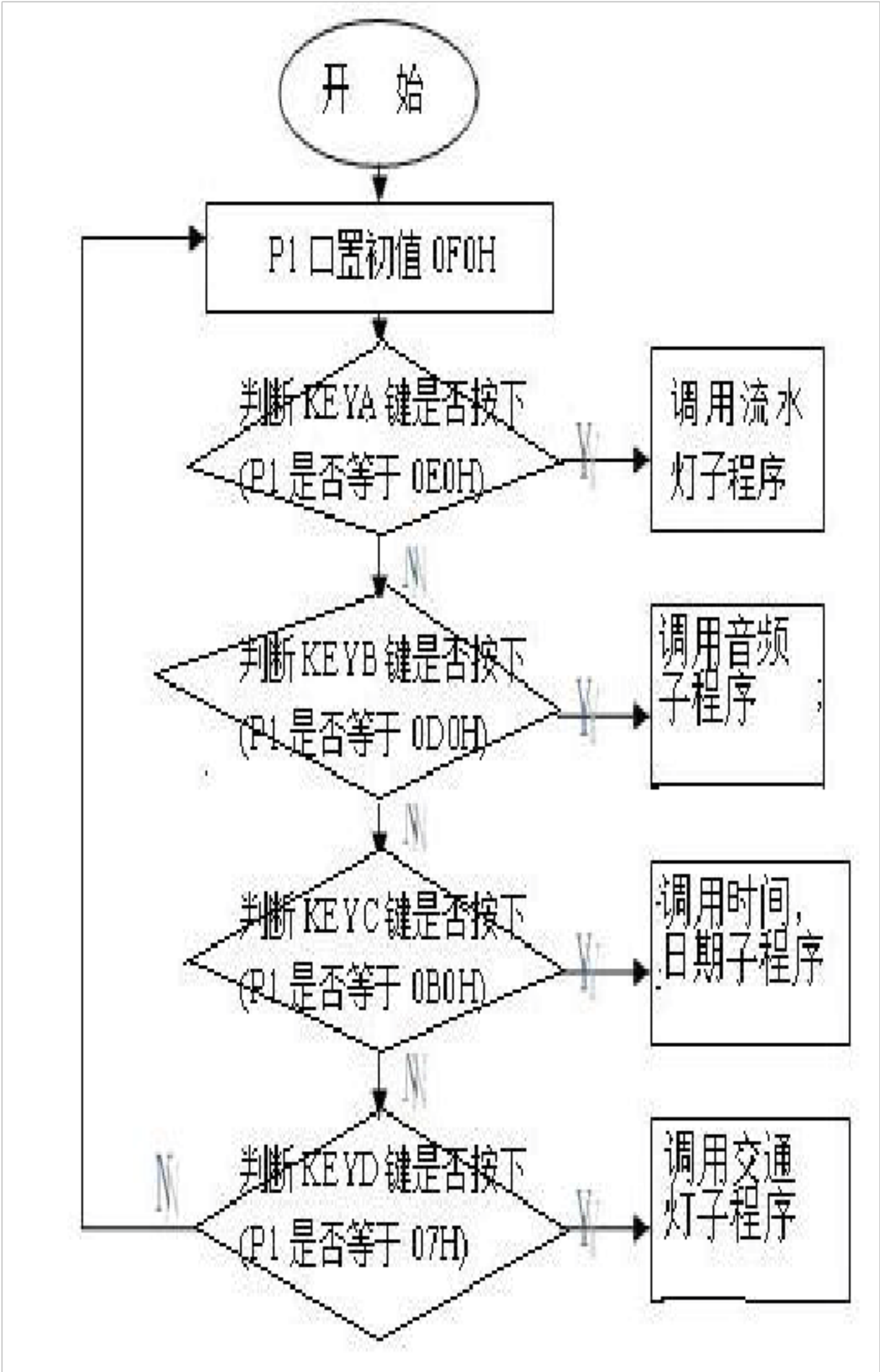
可选择一只蜂鸣器来实现这一功能。压电式蜂鸣器（ ）工作时约需 10mA 的驱动电流，并设计一个相应的驱动及控制电路。

蜂鸣器单元电路是通过一个 PNP 三极管来放大驱动蜂鸣器，蜂鸣器的正极接到+5V 电源上面，蜂鸣器的负极接到三极管的发射极，三极管的基极经过限流电阻后由单片机的控制。

当输出高电平时，三极管截止，蜂鸣器的两引脚间的直流电压接近于 0V，没有电流流过线圈，蜂鸣器不发声；当输出低电平时，三极管导通，使蜂鸣器的两个引脚间获得将接近+5V的直流电压，这样蜂鸣器的电流形成回路，发出声音。因此，我们可以通过程序控制 IO 口输出的电平来控制蜂鸣器是否发出声音，实现各种可能音响的产生。

## 四、软件设计流程

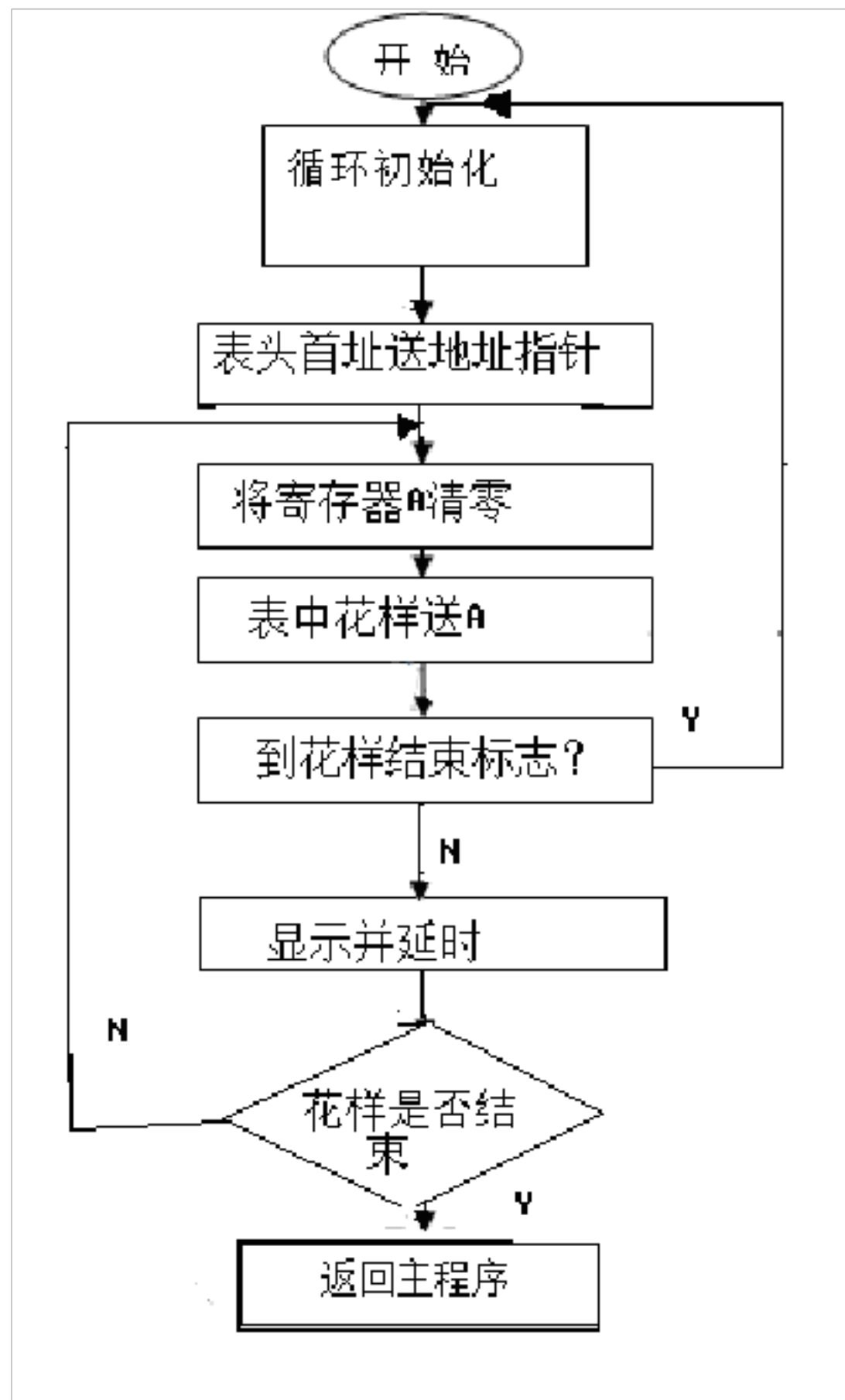
### 一)主程序流程图

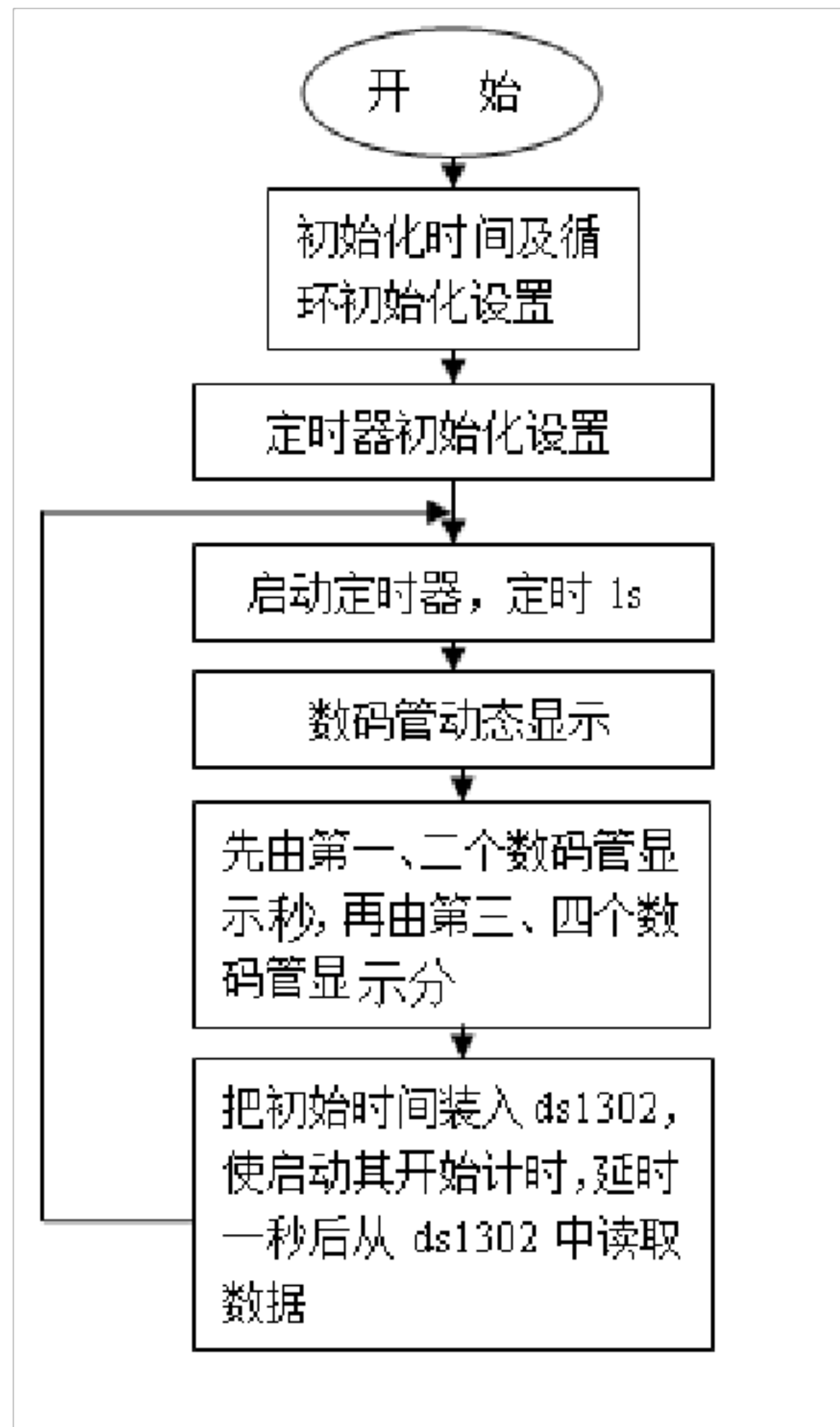


主程序流程图

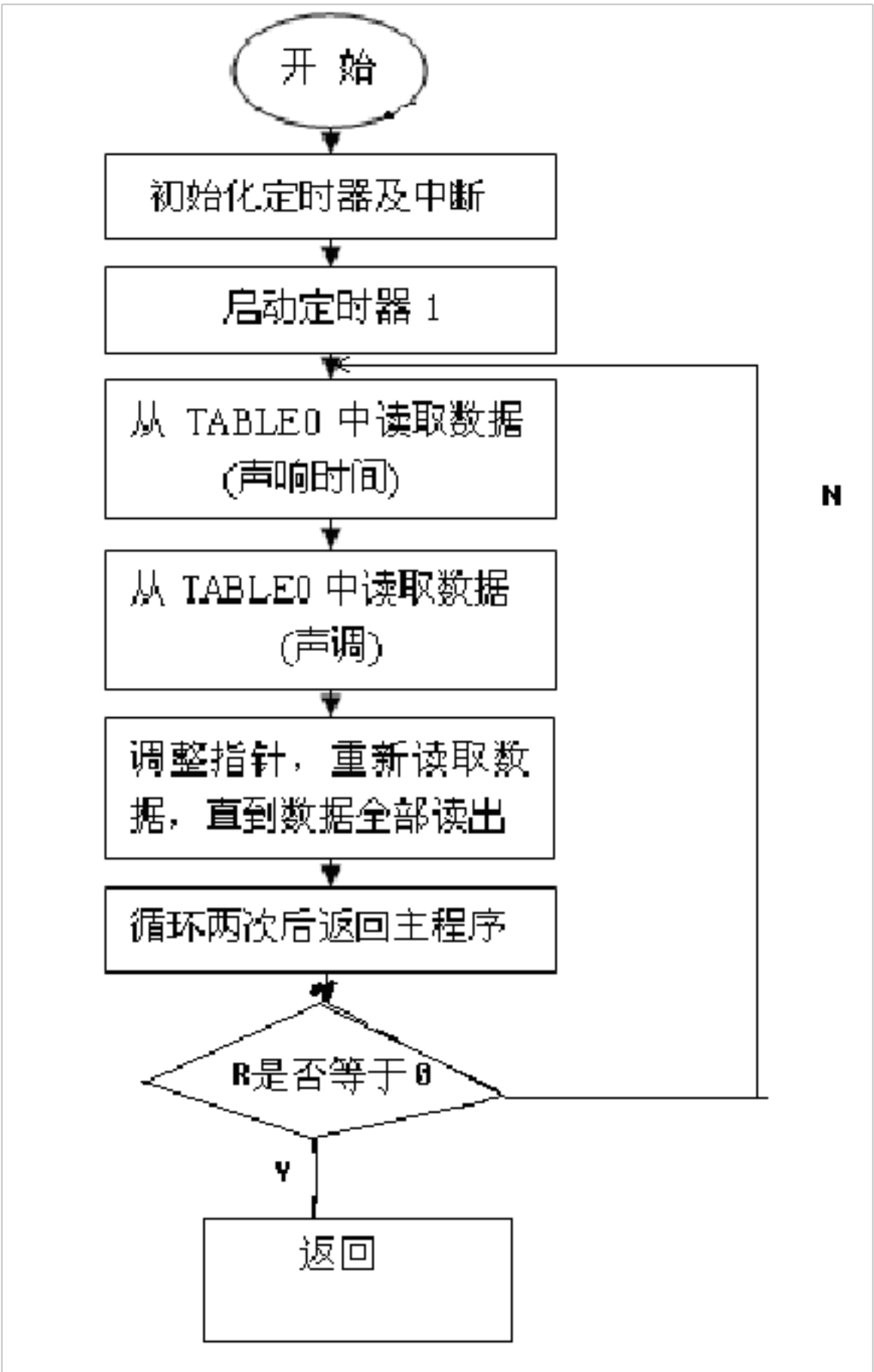
(二) 各模块流程图

流水灯程序流程图





利用蜂鸣器的音乐流程图



智能交通灯



以上内容仅为本文档的试下载部分，为可阅读页数的一半内容。如要下载或阅读全文，请访问：<https://d.book118.com/125113324322011224>