

电子设计自学手册

(2012年版)

天津工业大学

电子与信息工程学院

前 言

《电子设计自学手册》是由电子与信息工程学院团委发起，以学校教改项目《基于学科竞赛提高学生实践能力的方法和实践研究》为依托，整合学院电子设计教学部、专业社团等资源编写的一本学生辅导用书。编写这本手册的初衷是想总结提炼多年来学生开展电子设计的实践经验并加以推广，让更多的学生分享经验、从中受益，进而达到增强学生的专业兴趣、强化学生的电子设计热情、扩大学生参与电子设计的覆盖面以及提高学生实践能力的目的。本手册在内容上尽量做到理论知识和典型作品介绍相结合，以方便学生理解掌握。本手册由于是学生经验的总结和提炼，采用了较多的学生语言，它在追求方便学生理解的同时，也显现了诸如概念不够严谨、内容不够完整等不足。我们真心地希望本手册能给学生带去些许帮助，真诚地希望能得到更多老师和学生的帮助，进一步完善优化自学手册。

目 录

| | |
|--------------------------------|----|
| 第一章 模拟电子技术基本知识 | 5 |
| 1.1 电阻: | 5 |
| 1.2 电容: | 6 |
| 1.3 场效应管: | 8 |
| 1.4 二极管: | 9 |
| 1.5 三极管 | 11 |
| 1.6 整流电路: | 12 |
| 1.7 滤波电路 | 12 |
| 1.8 稳压电路: | 13 |
| 第二章 数字电子技术基本知识 | 16 |
| 2.1 电平特性 | 16 |
| 2.2 二进制与十六进制 | 16 |
| 2.3 数字逻辑运算: | 16 |
| 2.4 编码器 | 21 |
| 2.5 译码器 | 21 |
| 2.6 加法器 | 23 |
| 第三章 常用工具介绍 | 24 |
| 3.1 工具概述 | 24 |
| 3.2 电烙铁的介绍与使用 | 24 |
| 3.3 吸锡器的介绍与使用 | 26 |
| 3.4 数字万用表的介绍与使用 | 27 |
| 第四章 C 语言程序设计基础 | 29 |
| 4.1 C 语言简介 | 29 |
| 4.2 C 语言数据类型 | 29 |
| 4.3 输入与输出函数 | 31 |
| 4.4 关系运算符 | 32 |
| 4.5 逻辑运算符 | 33 |
| 4.6 if 条件选择语句 | 34 |
| 4.7 switch...case 条件选择语句 | 35 |

| | |
|---------------------------------------|----|
| 4.8 for 循环语句 | 36 |
| 4.9 while 及 do...while 循环语句..... | 37 |
| 4.10 函数..... | 39 |
| 第五章 单片机基本知识与例程分析 | 41 |
| 5.1 单片机概述 | 41 |
| 5.2Keil 软件介绍..... | 41 |
| 5.3 用单片机点亮一个 LED..... | 41 |
| 5.4 数码管应用..... | 46 |
| 5.5 键盘检测 | 47 |
| 第六章 单片机常用外围器件 | 50 |
| 6.1 蜂鸣器..... | 50 |
| 6.2 继电器..... | 50 |
| 6.3 直流电机..... | 51 |
| 6.4 步进电机..... | 51 |
| 6.5AD 与 DA | 52 |
| 第七章 常用软件及其功能..... | 54 |
| 7.1 Multisim..... | 54 |
| 7.2 Keil..... | 55 |
| 7.3 Protues | 55 |
| 7.4 Protel99 SE/Altium Designer | 56 |
| 7.5 其他软件..... | 57 |
| 第八章 优秀作品选集 | 59 |
| 第一篇光控自动窗帘 | 59 |
| 第二篇 LED 照明灯的智能控制..... | 69 |
| 第三篇 多功能数字钟的设计 | 77 |
| 第四篇 智能测距避障小车的设计 | 85 |
| 第五篇 宽带直流放大器的设计 | 92 |

第一章 模拟电子技术基本知识

本章主要介绍电子设计中常用到的模拟电子元器件以及模拟电子电路的基本知识。由于编者知识浅薄，读者请参阅教材《模拟电子技术基础》。

1.1 电阻

1.1.1 电阻的分类

按阻值特性：固定电阻、可调电阻、特种电阻（敏感电阻）

按制造材料：碳膜电阻、金属膜电阻、线绕电阻，无感电阻，薄膜电阻等。

按安装方式：插件电阻、贴片电阻

按功能：负载电阻，采样电阻，分流电阻，保护电阻等

1.1.2 常用电阻图片



图 1.1 光敏电阻



图 1.2 贴片电阻



图 1.3 金属氧化膜电阻

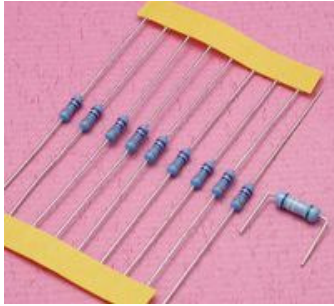


图 1.4 金属膜电阻



图 1.5 碳膜电阻

1.1.3 识别色环电阻的阻值：

一个电阻色环由 4 部分组成（不包括精密电阻）。四个色环的其中第一、二环分别代表阻值的前两位数；第三环代表 10 的幂；第四环代表误差。

熟记第一、二环每种颜色所代表的数。可这样记忆：

棕=1 红=2， 橙=3， 黄=4， 绿=5， 蓝=6， 紫=7， 灰=8， 白=9， 黑=0。

彩虹的颜色分布：红橙黄绿蓝靛（diàn）紫，去掉靛，后面添上灰白黑，前面加上棕，对应数字 1 开始。

从数量级来看，总体上可把它们划分为三个大的等级，即：金、黑、棕色是欧姆级的；红是千欧级，橙、黄色是十千欧级的；绿是兆欧级、蓝色则是十兆欧级的。这样划分一下也好记忆。所以要先看第三环颜色（倒数第 2 个颜色），才能准确。

第四环颜色所代表的误差：金色为 5%；银色为 10%；无色为 20%。

举例说明：

四个色环颜色为：黄橙红金

读法：前三颜色对应的数字为 432，金为 5%，所以阻值为 $43 \times 10^2 = 4300 = 4.3\text{K}\Omega$ ，误差为 5%。

1.1.4 电阻器的极限参数：

额定电压：当实际电压超过额定电压时，即便满足功率要求，电阻器也会被击穿损坏。

额定功率：所选电阻器的额定功率应大于实际承受功率的两倍以上才能保证电阻器在电路中长期工作的可靠性。

1.2 电容

电容（或电容量， **Capacitance**）指的是在给定电位差下的电荷储藏量；记为 C ，国际单位是法拉（F）。一般来说，电荷在电场中会受力而移动，当导体之间有了介质，则阻碍了电荷移动而使得电荷累积在导体上，造成电荷的累积储存，最常见的例子就是两片平行金属板。

1.2.1 电容符号



图 1.6 电容符号

1.2.2 单位及转换

1 法拉(F)= 1000 毫法(mF)=1000000 微法(μ F)

1 微法(μ F)= 1000 纳法(nF)= 1000000 皮法(pF)。

1.2.3 定义式

$C=Q/U$; Q 为电荷, U 为电压。

多电容器并联计算公式: $C=C_1+C_2+C_3+\dots+C_n$

多电容器串联计算公式: $1/C=1/C_1+1/C_2+\dots+1/C_n$

1.2.4 电容的分类

按照安装方式: 插件电容、贴片电容。

按电路中电容起的作用: 耦合电容、滤波电容、高频消振电容、旁路电容等。

按照功能: 云母电容、高频瓷介电容、低频瓷介电容、空气介质可变电容器、薄膜介质微调电容器、独石电容等。

按照结构分三大类: 固定电容器、可变电容器和微调电容器。

1.2.5 常用电容图片

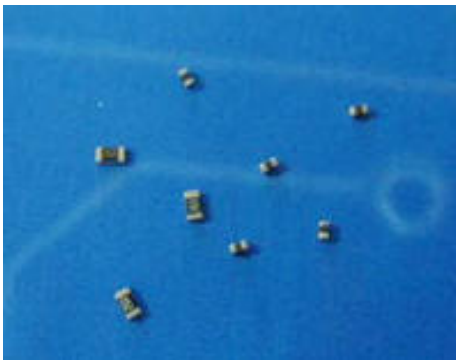


图 1.7 贴片电容

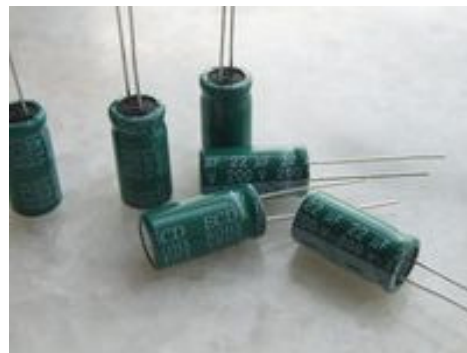


图 1.8 插件电容



图 1.9 电解电容



图 1.10 玻璃釉电容



图 1.11 云母电容

1.2.6 电容的作用

电容在电路中的作用: 具有隔断直流、连通交流、阻止低频的特性, 广泛应用在耦合、

隔直、旁路、滤波、调谐、能量转换和自动控制等。

1、滤波电容：它接在直流电压的正负极之间，以滤除直流电源中不需要的交流成分，使直流电平滑，通常采用大容量的电解电容，并接其它类型的小容量电容以滤除高频交流电。

2、退耦电容：并接于放大电路的电源正负极之间，防止由电源内阻形成的正反馈而引起的寄生振荡。

3、旁路电容：在交直流信号的电路中，将电容并接在电阻两端或由电路的某点跨接到公共电位上，为交流信号或脉冲信号设置一条通路，避免交流信号成分因通过电阻产生压降衰减。

4、耦合电容：在交流信号处理电路中，用于连接信号源和信号处理电路或者作为两放大器的级间连接，用于隔断直流，让交流信号通过，使前后级放大电路的直流工作点互不影响。

1.2.7 电容的主要技术指标：

常用固定式电容的直流工作电压系列为：6.3V，10V，16V，25V，40V，63V，100V，160V，250V，400V。耐压级别常见的有 0.2、I、II、III、IV、V、VI 等七个等级，对应不同的容许误差。

1.3 场效应管

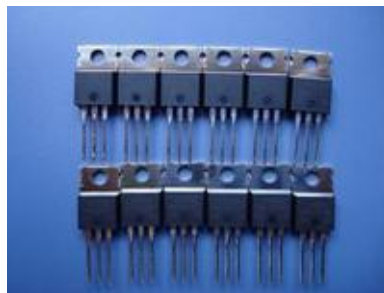


图 1.12 场效应管

场效应晶体管（Field Effect Transistor 缩写(FET)）简称场效应管。由多数载流子参与导电，也称为单极型晶体管。它属于电压控制型半导体器件。具有输入电阻高（ $10^8 \sim 10^9 \Omega$ ）、噪声小、功耗低、动态范围大、易于集成、没有二次击穿现象、安全工作区域宽等优点。

主要作用：

1. 场效应管可应用于信号放大。由于场效应管放大器的输入阻抗很高，因此耦合电容可以容量较小，不必使用电解电容器。

2. 场效应管很高的输入阻抗非常适合作阻抗变换。常用于多级放大器的输入级作阻抗变换。

3. 场效应管可以用作可变电阻。

4. 场效应管可以方便地用作恒流源。

5. 场效应管可以用作电子开关。

分类：结型场效应管的分类：结型场效应管有两种结构形式，它们是 N 沟道结型场效应管和 P 沟道结型场效应管。

1.4 二极管



图 1.13 直插式二极管

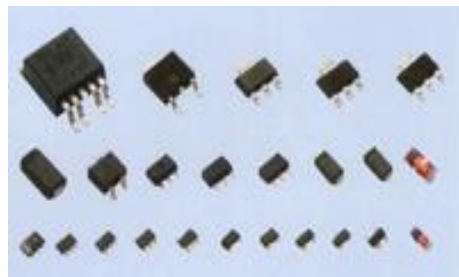


图 1.14 贴片式二极管

二极管又称晶体二极管，简称二极管(diode)，另外，还有早期的真空电子二极管；它是一种具有单向传导电流的电子器件。在半导体二极管内部有一个 PN 结两个引线端子，这种电子器件按照外加电压的方向，具备单向电流的转导性。一般来讲，晶体二极管是一个由 p 型半导体和 n 型半导体烧结形成的 p-n 结界面。在其界面的两侧形成空间电荷层，构成自建电场。当外加电压等于零时，由于 p-n 结两边载流子的浓度差引起扩散电流和由自建电场引起的漂移电流相等而处于电平衡状态，这也是常态下的二极管特性。

1.4.1 二极管特性

二极管最重要的特性就是单方向导电性。在电路中，电流只能从二极管的正极流入，负极流出。

正向特性：在电子电路中，将二极管的正极接在高电位端，负极接在低电位端，二极管就会导通，这种连接方式，称为正向偏置。必须说明，当加在二极管两端的正向电压很小时，二极管仍然不能导通，流过二极管的正向电流十分微弱。只有当正向电压达到某一数值（这一数值称为“门坎电压”，又称“死区电压”，锗管约为 0.1V，硅管约为 0.5V）以后，二极管才能真正导通。导通后二极管两端的电压基本上保持不变（锗管约为 0.3V，硅管约为 0.7V），

称为二极管的“正向压降”。

反向特性：在电子电路中，二极管的正极接在低电位端，负极接在高电位端，此时二极管中几乎没有电流流过，此时二极管处于截止状态，这种连接方式，称为反向偏置。二极管处于反向偏置时，仍然会有微弱的反向电流流过二极管，称为漏电流。当二极管两端的反向电压增大到某一数值，反向电流会急剧增大，二极管将失去单方向导电特性，这种状态称为二极管的击穿。

1.4.2 二极管根据用途分类

检波用二极管、整流用二极管、限幅用二极管、调制用二极管、混频用二极管、放大用二极管、开关用二极管、稳压二极管、雪崩二极管等。

1.4.3 二极管的特性曲线

与 PN 结一样，二极管具有单向导电性。硅二极管典型伏安特性曲线（图）。在二极管加有正向电压，当电压值较小时，电流极小；当电压超过 0.6V 时，电流开始按指数规律增大，通常称此为二极管的开启电压；当电压达到约 0.7V 时，二极管处于完全导通状态，通常称此电压为二极管的导通电压，用符号 U_D 表示。

对于锗二极管，开启电压为 0.2V，导通电压 U_D 约为 0.3V。

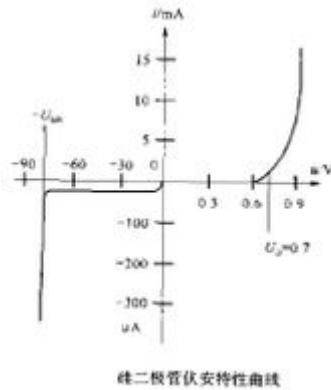


图 1.15 二极管伏安特性曲线

在二极管加有反向电压，当电压值较小时，电流极小，其电流值为反向饱和电流 I_S 。当反向电压超过某个值时，电流开始急剧增大，称之为反向击穿，称此电压为二极管的反向击穿电压，用符号 U_{BR} 表示。不同型号的二极管的击穿电压 U_{BR} 值差别很大，从几十伏到几千伏。

1.4.4 发光二极管：

发光二极管简称为 LED。由镓 (Ga) 与砷 (AS)、磷 (P) 的化合物制成的二极管，当电子与空穴复合时能辐射出可见光，因而可以用来制成发光二极管。在电路及仪器中作为指示灯，或者组成文字或数字显示。磷砷化镓二极管发红光，磷化镓二极管发绿光，碳化硅二极管发黄光。

发光二极管正负引脚判别：长正短负。

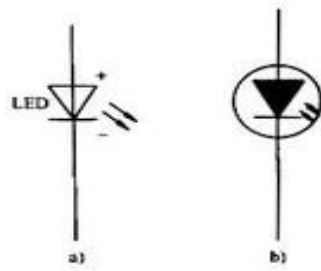


图 4-21 发光二极管的
电路图形符号
a) 新图形符号 b) 旧图形符号

图 1.16 发光二极管符号



图 1.17 发光二极管实物

1.5 三极管



图 1.18 三极管实物

1.5.1 三极管的分类

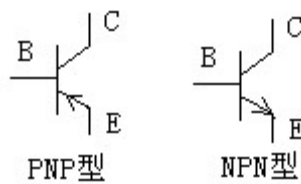


图 1.19 三极管分类

半导体三极管也称为晶体三极管，可以说它是电子电路中最重要器件。它最主要的功

能是电流放大和开关作用。三极管顾名思义具有三个电极。二极管是由一个 PN 结构成的，而三极管由两个 PN 结构成，共用的一个电极成为三极管的基极(用字母 b 表示)。其他的两个电极成为集电极(用字母 c 表示)和发射极(用字母 e 表示)。由于不同的组合方式，形成了一种是 NPN 型的三极管，另一种是 PNP 型的三极管。三极管的种类很多，并且不同型号各有不同的用途。三极管大都是塑料封装或金属封装，常见三极管的外观，有一个箭头的电极是发射极，箭头朝外的是 NPN 型三极管，而箭头朝内的是 PNP 型。实际上箭头所指的方向是电流的方向。

当发射极和集电极之间的电压处于在放大区内时，较小的基极电流的变化引起集电极电流成比例的较大变化，这就是三极管最基本的作用——电流放大作用，三极管其他的作用都是由此而来。

1.6 整流电路

整流电路 (rectifying circuit) 把交流电转换为直流能的电路。大多数整流电路由变压器、整流主电路和滤波器等组成。它在直流电动机的调速、发电机的励磁调节、电解、电镀等领域得到广泛应用。整流电路通常由主电路、滤波器和变压器组成。

电源电路中的整流电路主要有半波整流电路、全波整流电路和桥式整流三种

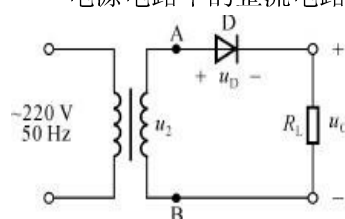


图 1.20 半波整流电路

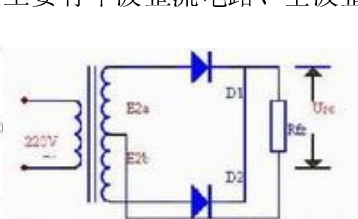


图 1.21 全波整流电路

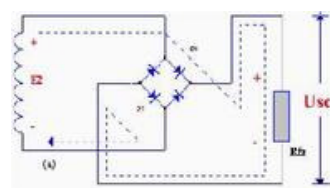


图 1.22 桥式整流电路

1.7 滤波电路

(1) 滤波电路常用于滤去整流输出电压中的纹波，一般由电抗元件组成，如在负载电阻两端并联电容器 C，或与负载串联电感器 L，以及由电容，电感组成而成的各种复式滤波电路。

(2) 常用的滤波电路有无源滤波和有源滤波两大类

(3) 电路作用：滤波电路尽可能减小脉动的直流电压中的交流成分，保留其直流成分，使输出电压纹波系数降低，波形变得比较平滑

(4) 当允许信号中较高频率的成分通过滤波器时，这种滤波器叫做高通滤波器。当允许信号中较低频率的成分通过滤波器时，这种滤波器叫做低通滤波器。当只允许信号中某个频率范围内的成分通过滤波器时，这种滤波器叫做带通滤波器。理想滤波器的行为特性通常用幅度-频率特性图描述，也叫做滤波器电路的幅频特性。理想滤波器的幅频特性如图所示。

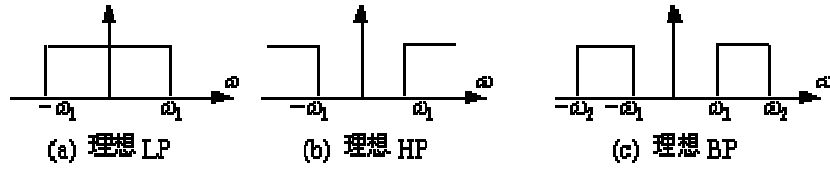


图 1.23 理想滤波器幅频特性

(5) 简单一阶低通有源滤波器

一阶低通滤波器的电路如左图所示，其幅频特性见右图，图中虚线为理想的情况，实线为实际的情况。特点是电路简单，阻带衰减太慢，选择性较差。

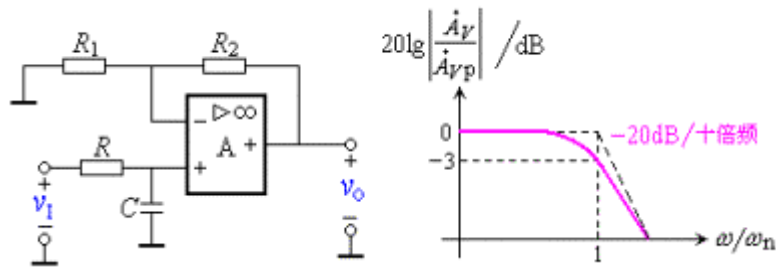


图 1.24 一阶低通滤波器及其幅频特性

当 $f=0$ 时，电容器可视为开路，通带内的增益为

$$A_p = 1 + \frac{R_2}{R_1}$$

一阶低通滤波器的传递函数如下：

$$A(s) = \frac{V_o(s)}{V_i(s)} = \frac{A_p}{1 + \left(\frac{s}{\omega_0}\right)}$$

其中， $\omega_0 = \frac{1}{RC}$ ， $S=j\omega$

该传递函数式的样子与一阶 RC 低通环节的增益频率表达式差不多，只是缺少通带增益 A_p 这一项。

1.8 稳压电路

利用电路的调整作用使输出电压稳定的过程称为稳压。在输入电压、负载、环境温度、电路参数等发生变化时仍能保持输出电压恒定的电路。这种电路能提供稳定的直流电源。



图 1.25 三端稳压器件

直流稳压电路的结构：调整元件、基准电压电路、取样电路、比较放大电路

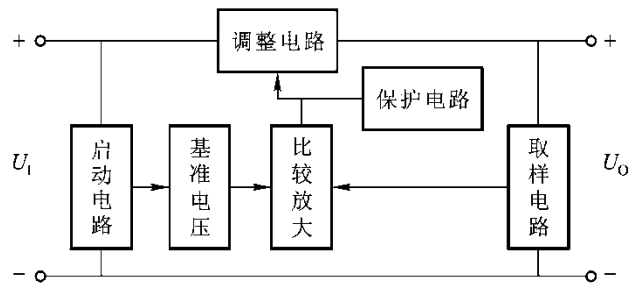


图 1.26 直流稳压电路的结构

1.8.1 稳压电路分类

按调整管与负载的接法分：并联型稳压电路、串联型稳压电路

按调整管的工作状态分：线性稳压电路、开关稳压电路

按线性集成稳压电路分：三端固定输出、三端可调输出

1.8.2 串联型稳压电路的工作原理：

a、电路组成、各部分的作用和稳压工作原理：

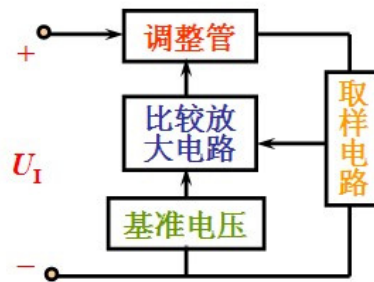


图 1.27 串联型稳压电路结构

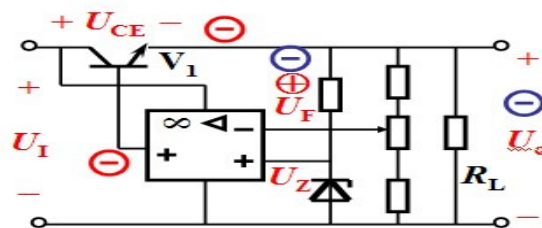


图 1.28 串联型稳压电路

自动调节 U_{CE} ，使输出电压稳定；
 电路引入了深度负反馈，所以运放工作于放大状态；

b、输出电压的计算：

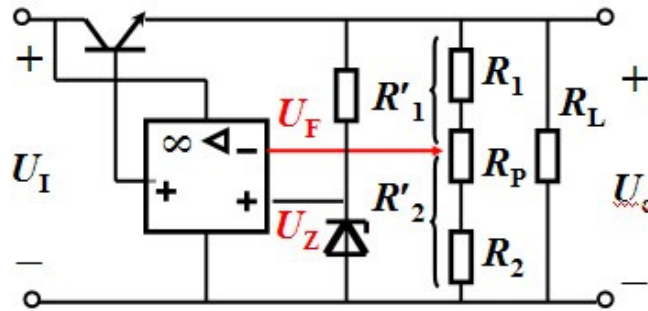


图 1.29 串联型稳压电路

$$U_F = \frac{U_O R'_2}{R_1 + R_2 + R_p} = U_Z \quad U_O = \frac{R_1 + R_2 + R_p}{R'_2} U_Z$$

$$U_{Omin} = \frac{R_1 + R_2 + R_p}{R_2 + R_p} U_Z \quad U_{Omax} = \frac{R_1 + R_2 + R_p}{R_2} U_Z$$

第二章 数字电子技术基本知识

2.1 电平特性

关于高电平和低电平的概念：电位指绝对电压的大小，电平指一定的电压范围。高电平和低电平在数字电路中分别表示两段电压范围。

例：电路中规定高电平为 $\geq 3V$ ，低电平为 $\leq 0.7V$ 。

TTL 电路中通常规定高电平的额定值为 3V。

从 2V 到 5V 都算高电平；低电平的额定值为 0.3V。

从 0V 到 0.8V 都算做低电平。

2.2 二进制与十六进制

二进制：只用 0 和 1 表示

十六进制：用 0、1、2、……A、B……F 表示的数。

表 2.1 十进制、二进制、十六进制数转换表

| 十进制 | 二进制 | 十六进制 | 十进制 | 二进制 | 十六进制 |
|-----|------|------|-----|------|------|
| 0 | 0000 | 0 | 8 | 1000 | 8 |
| 1 | 0001 | 1 | 9 | 1001 | 9 |
| 2 | 0010 | 2 | 10 | 1010 | A |
| 3 | 0011 | 3 | 11 | 1011 | B |
| 4 | 0100 | 4 | 12 | 1100 | C |
| 5 | 0101 | 5 | 13 | 1101 | D |
| 6 | 0110 | 6 | 14 | 1110 | E |
| 7 | 0111 | 7 | 15 | 1111 | F |

2.3 数字逻辑运算：

在二值逻辑函数中，最基本的逻辑运算有与（AND）、或（OR）、非（NOT）三种逻辑运算

2.3.1 与运算

也叫逻辑乘或逻辑与，即当所有的条件都满足时，事件才会发生，缺一不可。

如图 2.1 所示电路，两个串联的开关控制一盏灯就是与逻辑事例，只有开关 A、B 同时闭合时灯才会亮。

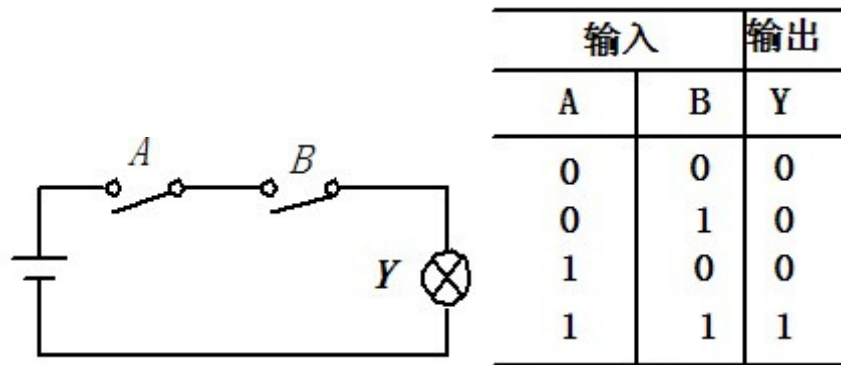


图 2.1 与门电路及其输入与输出关系

这种与逻辑可以写成下面的表达式：

$Y = A \cdot B$ 称为与逻辑式，这种运算称为与运算。

从真值表中可知，其逻辑规律服从“有 0 出 0，全 1 才出 1”

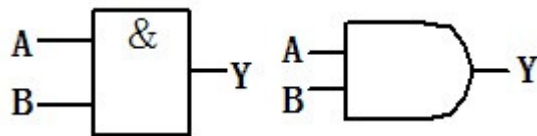


图 2.2 与门逻辑符号

2.3.2 或运算：

或运算也叫逻辑加或逻辑或，即当其中一个条件满足时，事件就会发生，即有一即可。

如图所示电路，两个并联的开关控制一盏灯就是或逻辑事例，只要开关 A、B 有一个闭合时灯就会亮。

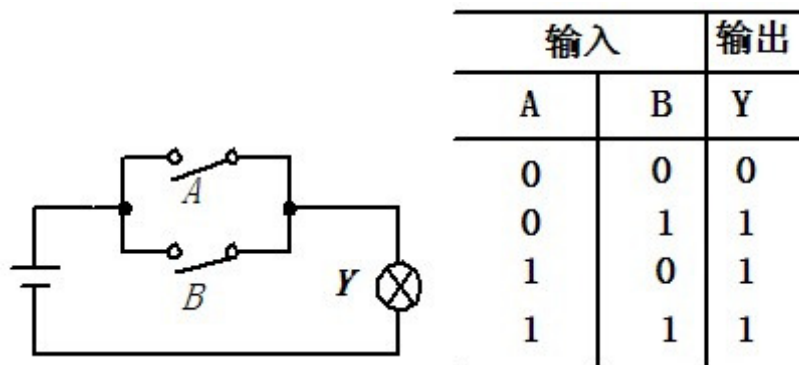


图 2.3 或门电路及其输入与输出关系

用与前面相同的逻辑赋值同样也可得到其真值表如表所示，其逻辑规律服从“有 1 出 1，全 0 才出 0”。

其逻辑式为： $Y = A + B$

上式说明：当逻辑变量 A、B 有一个为 1 时，逻辑函数输出 Y 就为 1。只有 A、B 全为 0，Y 才为 0。

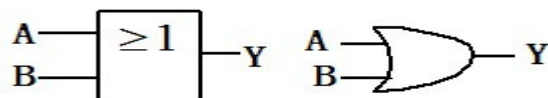


图 2.4 或门逻辑符号

2.3.3 非逻辑运算

条件具备时，事件不发生；条件不具备时，事件发生，这种因果关系叫做逻辑非，也称逻辑求反

如图所示电路，一个开关控制一盏灯就是非逻辑事例，当开关 A 闭合时灯就会不亮。

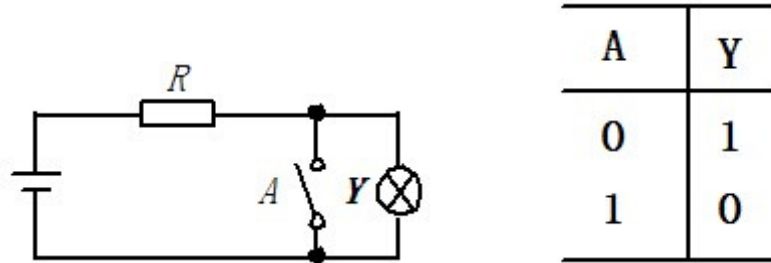


图 2.4 非门逻辑符号

用与前面相同的逻辑赋值同样也可得到其真值表如表所示。

非逻辑运算也叫逻辑非或非运算、反相运算，即输出变量是输入变量的相反状态。其逻辑式为 $Y = \bar{A}$ 或 $Y = A'$ 。

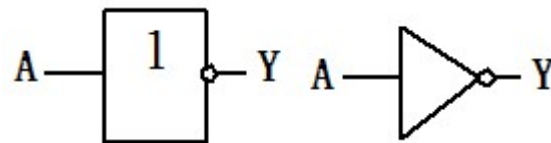


图 2.5 非门逻辑符号

以上为最基本的三种逻辑运算，除此之外，还有下面的由基本逻辑运算组合出来的逻辑运算

2.3.4 与非 (NAND) 逻辑运算

与非运算是先与运算后非运算的组合。以二变量为例，布尔代数表达式为： $Y = (AB)'$

| 输入 | | 输出 |
|----|---|----|
| A | B | Y |
| 0 | 0 | 1 |
| 0 | 1 | 1 |
| 1 | 0 | 1 |
| 1 | 1 | 0 |

图 2.6 与非门逻辑运算

其逻辑规律服从“有 0 出 1，全 1 才出 0”

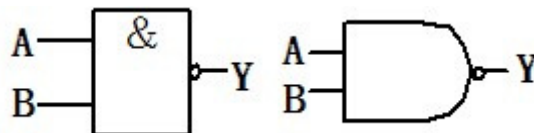


图 2.7 与非门逻辑符号

2.3.5 或非 (NOR) 运算

或非运算是先或运算后非运算的组合。以二变量 A、B 为例，布尔代数表达式为：

$$Y = (A + B)'$$

| 输入 | | 输出 |
|----|---|----|
| A | B | Y |
| 0 | 0 | 1 |
| 0 | 1 | 0 |
| 1 | 0 | 0 |
| 1 | 1 | 0 |

图 2.8 或非门逻辑运算

或非逻辑规律服从有“1”出“0”全“0”出“1”

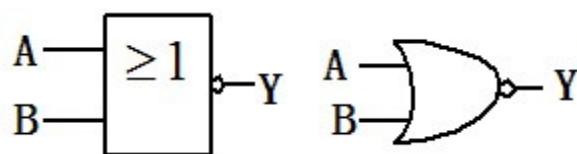


图 2.9 或非门逻辑符号

2.3.6 与或非运算

与或非运算是“先与后或非再非”三种运算的组合。以四变量为例，逻辑表达式为：

$$Y = (AB + CD)'$$

上式说明，当输入变量 A、B 同时为 1 或 C、D 同时为 1 时，输出 Y 才等于 0。与或非运算是先或运算后非运算的组合。在工程应用中，与或非运算由与或非门电路来实现，其真值表见书 P22 表 2.2.6 所示，逻辑符号如图所示

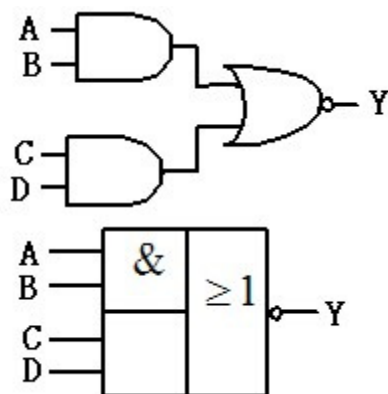


图 2.10 与或非门逻辑符号

2.3.7 异或运算

其布尔表达式（逻辑函数式）为 $Y = A \oplus B = AB' + A'B$

符号“ \oplus ”表示异或运算，即两个输入逻辑变量取值不同时 Y=1，即不同为“1”相同为“0”，异或运算用异或门电路来实现

| 输入 | | 输出 |
|----|---|----|
| A | B | Y |
| 0 | 0 | 0 |
| 0 | 1 | 1 |
| 1 | 0 | 1 |
| 1 | 1 | 0 |

图 2.11 异或逻辑运算

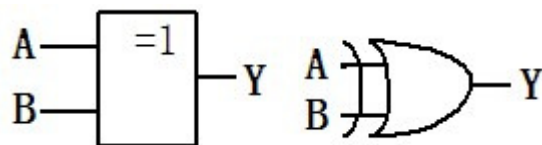


图 2.12 异或逻辑符号

2.3.8 同或运算：

$$Y = A \odot B = (A \oplus B)' = AB + A'B'$$

符号“ \odot ”表示同或运算，即两个输入变量值相同时 $Y=1$ ，即相同为“1”不同为“0”。同或运算用同或门电路来实现，它等价于异或门输出加非门。

| 输入 | | 输出 |
|----|---|----|
| A | B | Y |
| 0 | 0 | 1 |
| 0 | 1 | 0 |
| 1 | 0 | 0 |
| 1 | 1 | 1 |

图 2.13 同或逻辑运算

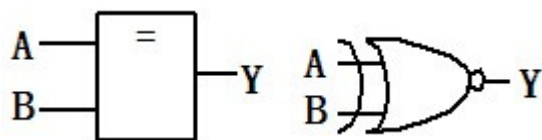


图 2.14 同或逻辑符号

2.4 编码器:

编码器 (encoder) 是将信号 (如比特流) 或数据进行编制、转换为可用以通讯、传输和存储的信号形式的设备。

按照工作原理编码器可分为增量式和绝对式两类。

增量式:

增量式编码器是将位移转换成周期性的电信号, 再把这个电信号转变成计数脉冲, 用脉冲的个数表示位移的大小。

绝对式:

绝对式编码器的每一个位置对应一个确定的数字码, 因此它的示值只与测量的起始和终止位置有关, 而与测量的中间过程无关。

2.5 译码器

译码器是组合逻辑电路的一个重要的器件, 其可以分为两类: 变量译码和显示译码。变量译码一般是一种较少输入变为较多输出的器件, 一般分为 2^n 译码和 8421BCD 码译码两类。显示译码主要解决二进制数显示成对应的十、或十六进制数的转换功能, 一般其可分为驱动 LED 和驱动 LCD 两类。

译码是编码的逆过程, 在编码时, 每一种二进制代码, 都赋予了特定的含义, 即都表示了一个确定的信号或者对象。把代码状态的特定含义“翻译”出来的过程叫做译码, 实现译码操作的电路称为译码器。或者说, 译码器是可以将输入二进制代码的状态翻译成输出信号, 以表示其原来含义的电路。

举例应用: 3-8 译码器

①3-8 译码器工作原理如下:

当一个选通端 ($G1$) 为高电平, 另两个选通端 ($\overline{G2A}$ 和 $\overline{G2B}$) 为低电平时, 可将地址端 (A、B、C) 的二进制编码在一个对应的输出端以低电平译出。

②3-8 译码器的作用:

利用 $G1$ 、 $\overline{G2A}$ 和 $\overline{G2B}$ 可级联扩展成 24 线译码器; 若外接一个反相器还可级联扩展成 32 线译码器。

若将选通端中的一个作为数据输入端时, 74LS138 还可作数据分配器

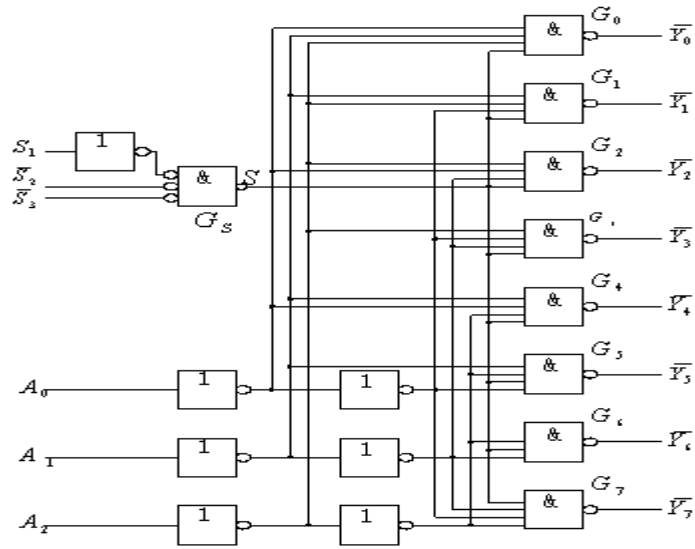
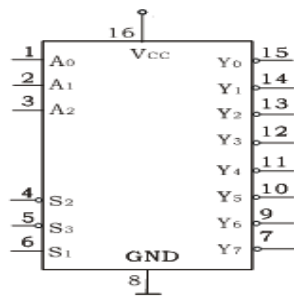


图 2.15 3-8 译码器内部电路



3-8译码器引脚排列

表 2.1 3-8 译码器的引脚分配

| | | | | | | | |
|----|----|----|----|----|----|-----|-----|
| A0 | A1 | A2 | S2 | S3 | S1 | Y0 | Y1 |
| 1 | 2 | 3 | 4 | 5 | 6 | 15 | 14 |
| Y2 | Y3 | Y4 | Y5 | Y6 | Y7 | GND | VCC |
| 13 | 12 | 11 | 10 | 9 | 7 | 8 | 16 |

表 2.2 3-8 译码器的功能表

| 输 入 | | | | | 输 出 | | | | | | | |
|-------|-------------------------|-------|-------|-------|-------------|-------------|-------------|-------------|-------------|-------------|-------------|-------------|
| S_1 | $\bar{S}_2 + \bar{S}_3$ | A_2 | A_1 | A_0 | \bar{Y}_0 | \bar{Y}_1 | \bar{Y}_2 | \bar{Y}_3 | \bar{Y}_4 | \bar{Y}_5 | \bar{Y}_6 | \bar{Y}_7 |
| 1 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |
| 1 | 0 | 0 | 0 | 1 | 1 | 0 | 1 | 1 | 1 | 1 | 1 | 1 |

| | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 1 | 0 | 0 | 1 | 0 | 1 | 1 | 0 | 1 | 1 | 1 | 1 | 1 |
| 1 | 0 | 0 | 1 | 1 | 1 | 1 | 1 | 0 | 1 | 1 | 1 | 1 |
| 1 | 0 | 1 | 0 | 0 | 1 | 1 | 1 | 1 | 0 | 1 | 1 | 1 |
| 1 | 0 | 1 | 0 | 1 | 1 | 1 | 1 | 1 | 1 | 0 | 1 | 1 |
| 1 | 0 | 1 | 1 | 0 | 1 | 1 | 1 | 1 | 1 | 1 | 0 | 1 |
| 1 | 0 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 0 |
| 0 | × | × | × | × | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |
| × | 1 | × | × | × | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |

无论从逻辑图还是功能表我们都可以看到 3-8 译码器的八个输出管脚,任何时刻要么全为高电平 1—芯片处于不工作状态,要么只有一个为低电平 0,其余 7 个输出管脚全为高电平 1。如果出现两个输出管脚在同一个时间为 0 的情况,说明该芯片已经损坏。

2.6 加法器

加法器是产生数的和的装置。加数和被加数为输入,和数与进位为输出的装置为半加器。若加数、被加数与低位的进位数为输入,而和数与进位为输出则为全加器。常用作计算机算术逻辑部件,执行逻辑操作、移位与指令调用。在电子学中,加法器是一种数位电路,其可进行数字的加法计算。在现代的电脑中,加法器存在于算术逻辑单元 (ALU) 之中。加法器可以用来表示各种数值,如:BCD、加三码,主要的加法器是以二进制作运算。由于负数可用二的补数来表示,所以减器也就不那么必要。

第三章 常用工具介绍

本章主要介绍我们作为电子爱好者或者研发者常用到的一些硬件工具,这些工具在实际的开发和调试中是必不可少的,作为初学者刚进入电子设计领域时,学习使用这些工具是非常重要的,同时也是必经的环节。

3.1 工具概述

常用的硬件工具主要包括制作工具与调试工具两大类。制作工具我们常用到的有电烙铁、吸锡器、镊子、小电钻、改锥、胶枪等;调试工具我们常用的有电源、数字万用表、示波器、信号发生器等,除此之外我们在一些特别的应用中(如通信、放大、高频电路)常会用到一些专用的仪器设备,如频谱仪、网络分析仪、高频信号发生器、数字电感/电容表、交流毫伏表等,在用漆包线制作小电感时还会用到电感测试仪等。

3.2 电烙铁的介绍与使用

3.2.1 什么是电路铁

电烙铁是电子制作和电器维修必工具,主要用途是焊接电子元件及导线,按结构可分为内热式电烙铁和外热式电烙铁,按功能可分为焊接用电烙铁和吸锡用电烙铁,根据用途不同又分为大功率电烙铁和小功率电烙铁。

3.2.2 电烙铁分类及构成

(1) 外热式电烙铁

由烙铁头、烙铁芯、外壳、木柄、电源引线、插头等部分组成。由于烙铁头安装在烙铁芯里面,故称为外热式电烙铁。烙铁芯是电烙铁的关键部件,它是将电热丝平行地绕制在一根空心瓷管上构成,中间的云母片绝缘,并引出两根导线与 220V 交流电源连接。外热式电烙铁的规格很多,常用的有 25W、45W、75W、100W 等,功率越大烙铁头的温度也就越高。

(2) 内热式电烙铁

由手柄、连接杆、弹簧夹、烙铁芯、烙铁头组成。由于烙铁芯安装在烙铁头里面,因而发热快,热利用率高,因此,称为内热式电烙铁。

(3) 恒温电烙铁

由于恒温电烙铁头内,装有带磁铁式的温度控制器,控制通电时间而实现温控,即给电

烙铁通电时，烙铁的温度上升，当达到预定的温度时，因强磁体传感器达到了居里点而磁性消失，从而使磁芯触点断开，这时便停止向电烙铁供电；当温度低于强磁体传感器的居里点时，强磁体便恢复磁性，并吸动磁芯开关中的永久磁铁，使控制开关的触点接通，继续向电烙铁供电。如此循环往复，便达到了控制温度的目的。

(4) 吸锡电烙铁

吸锡电烙铁是将活塞式吸锡器与电烙铁溶为一体的拆焊工具。它具有使用方便、灵活、适用范围宽等特点。这种吸锡电烙铁的不足之处是每次只能对一个焊点进行拆焊。

3.2.3 电烙铁使用功率选择

(1) 焊接集成电路、晶体管及受热易损元器件时，应选用 20W 内热式或 25W 的外热式电烙铁。

(2) 焊接导线及同轴电缆时，应先用 45W~75W 外热式电烙铁，或 50W 内热式电烙铁。

(3) 焊接较大的元器件时，如行输出变压器的引线脚、大电解电容器的引线脚，金属底盘接地焊片等，应选用 100W 以上的电烙铁。

3.2.4 电烙铁头的使用方法

新烙铁使用前，应用细砂纸将烙铁头打光亮，通电烧热，蘸上松香后用烙铁头刃面接触焊锡丝，使烙铁头上均匀地镀上一层锡。电烙铁要用 220V 交流电源，使用时要特别注意安全。应认真做到以下几点：电烙铁插头最好使用三极插头。要使外壳妥善接地。使用前，应认真检查电源插头、电源线有无损坏。并检查烙铁头是否松动。电烙铁使用中，不能用力敲击。要防止跌落。烙铁头上焊锡过多时，可用布擦掉。不可乱甩，以防烫伤他人。焊接过程中，烙铁不能到处乱放。不焊时，应放在烙铁架上。注意电源线不可搭在烙铁头上，以防烫坏绝缘层而发生事故。使用结束后，应及时切断电源，拔下电源插头。冷却后，再将电烙铁收回工具箱。

3.2.5 电烙铁的使用步骤

1、选用合适的焊锡，应选用焊接电子元件用的低熔点焊锡丝。

2、助焊剂，用 25%的松香溶解在 75%的酒精（重量比）中作为助焊剂。

3、电烙铁使用前要上锡，具体方法是：将电烙铁烧热，待刚刚能熔化焊锡时，涂上助焊剂，再用焊锡均匀地涂在烙铁头上，使烙铁头均匀的吃上一层锡。

4、焊接方法，把焊盘和元件的引脚用细砂纸打磨干净，涂上助焊剂。用烙铁头沾取适量焊锡，接触焊点，待焊点上的焊锡全部熔化并浸没元件引线头后，电烙铁头沿着元器件的引脚轻轻往上一提离开焊点。

5、焊接时间不宜过长，否则容易烫坏元件，必要时可用镊子夹住管脚帮助散热。

6、焊点应呈正弦波峰形状，表面应光亮圆滑，无锡刺，锡量适中。

7、焊接完成后，要用酒精把线路板上残余的助焊剂清洗干净，以防炭化后的助焊剂影响电路正常工作。

8、集成电路应最后焊接，电烙铁要可靠接地，或断电后利用余热焊接。或者使用集成电路专用插座，焊好插座后再把集成电路插上去。

9、电烙铁应放在烙铁架上。

备注：焊咀,烙铁咀,电烙铁头均属于烙铁头市场上的叫法

3.2.6 铁电烙铁使用注意事项

1、应使用两相三线插座作为电源，并将接地线可靠接地或接零。

2、电源线不应拉得太长，不得卷曲，不得使用强度低的塑料线作为电源线。

3、应根据焊条形状和尺寸选用合适瓦数和焊头的电烙铁。如焊接大件，一般不宜使用瓦数小的电烙铁，否则，会造成虚焊；焊接一般电子线路，应使用 45 瓦以下的小烙铁，以防烫坏元件。

4、焊接大件时，应使用电烙铁将焊件加热到一定温度后施焊。

5、焊接前应将焊件表面的污垢或氧化膜除去，并抹上松香或焊油。

6、焊接过程中暂时不使用电烙铁时，应将其置于专用的支架上，避免烫坏导线或其他物件；电烙铁的放置地点应远离易燃物，以防引起电气火灾。

7、焊接结束时应断开电烙铁电源。

3.3 吸锡器的介绍与使用

吸锡器分为手动式和电动式两种，手动式价格低廉，我们通常使用的较多，本节讲述手动式吸锡器的使用方法及注意事项。

3.3.1 吸锡器的介绍

吸锡器是一种修理电器用的工具，收集拆卸焊盘电子元件时融化的焊锡。有手动，电动两种。维修拆卸零件需要使用吸锡器，尤其是大规模集成电路，更为难拆，拆不好容易破坏印制电路板，造成不必要的损失。简单的吸锡器是手动式的，且大部分是塑料制品，它的头部由于常常接触高温，因此通常都采用耐高温塑料制成。

3.3.2 吸锡器的使用步骤

(1) 先把吸锡器活塞向下压至卡住。

(2) 用电烙铁加热焊点至焊料熔化。

(3) 移开电烙铁的同时，迅速把吸锡器咀贴上焊点，并按动吸锡器按钮。

(4) 一次吸不干净，可重复操作多次。

3.3.3 吸锡器的使用技巧

(1) 要确保吸锡器活塞密封良好。通电前，用手指堵住吸锡器头的小孔，按下按钮，如活塞不易弹出到位，说明密封是好的。

(2) 吸锡器头的孔径有不同尺寸，要选择合适的规格使用。

(3) 吸锡器头用旧后，要适时更换新的。

(4) 接触焊点以前，每次都蘸一点松香，改善焊锡的流动性。

(5) 头部接触焊点的时间稍长些，当焊锡融化后，以焊点针脚为中心，手向外按顺时针方向画一个圆圈之后，再按动吸锡器按钮。

3.4 数字万用表的介绍与使用

数字万用表分很多型号，其性能差异也很大，我们实验室配有的绝大多数为 DT9205 型数字万用表，本节将介绍 DT9205 型数字万用表的使用。

3.4.1 数字万用表的介绍

数字万用表在电子设计与制作中主要用于测量电阻、电压、电流、三极管部分参数与极性、二极管部分参数与极性等。

3.4.2 数字万用表各种功能的使用方法

1、电压的测量

(1) 直流电压的测量，如电池、随身听电源等。首先将黑表笔插进“com”孔，红表笔插进“VΩ”。把旋钮选到比估计值大的量程（注意：表盘上的数值均为最大量程，“V—”表示直流电压档，“V~”表示交流电压档，“A”是电流档），接着把表笔接电源或电池两端；保持接触稳定。数值可以直接从显示屏上读取，若显示为“1.”，则表明量程太小，那么就要加大量程后再测量工业电器。如果在数值左边出现“-”，则表明表笔极性与实际电源极性相反，此时红表笔接的是负极。

(2) 交流电压的测量。表笔插孔与直流电压的测量一样，不过应该将旋钮打到交流档“V~”处所需的量程即可。交流电压无正负之分，测量方法跟前面相同。无论测交流还是直流电压，都应注意人身安全，不要随便用手触摸表笔的金属部分。

2、电流的测量

(1) 直流电流的测量。先将黑表笔插入“COM”孔。若测量大于 200mA 的电流，则要将红表笔插入“10A”插孔并将旋钮打到直流“10A”档；若测量小于 200mA 的电流，则将红表笔插入“200mA”插孔，将旋钮打到直流 200mA 以内的合适量程。调整好后，就可以测量了。将万用表串进电路中，保持稳定，即可读数。若显示为“1.”，那么就要加大量程；如果在数值左边出现“-”，则表明电流从黑表笔流进万用表。

(2) 交流电流的测量。测量方法与 1 相同，不过档位应该打到交流档位，电流测量完毕后应将红笔插回“VΩ”孔，若忘记这一步而直接测电压，你的表或电源会在“一缕青烟中上云霄”——报废！

3、电阻的测量

将表笔插进“COM”和“VΩ”孔中，把旋钮打旋到“Ω”中所需的量程，用表笔接在电阻两端金属部位，测量中可以用手接触电阻，但不要把手同时接触电阻两端，这样会影响测量精确度的（人体是电阻很大但是有限大的导体）。读数时，要保持表笔和电阻有良好的接触；

注意单位：在“200”档时单位是“ Ω ”，在“2K”到“200K”档时单位为“ $K\Omega$ ”，“2M”以上的单位是“ $M\Omega$ ”。

4、二极管的测量

数字万用表可以测量发光二极管，整流二极管等各种二极管，测量时，表笔位置与电压测量一样，将旋钮旋到“二极管/蜂鸣器”档；用红表笔接二极管的正极，黑表笔接负极，这时会显示二极管的正向压降。肖特基二极管的压降是 0.2V 左右，普通硅整流管（1N4000、1N5400 系列等）约为 0.7V，发光二极管约为 1.8~2.3V。调换表笔，显示屏显示“1.”则为正常，因为二极管的反向电阻很大，否则此管已被击穿。

5、三极管的测量

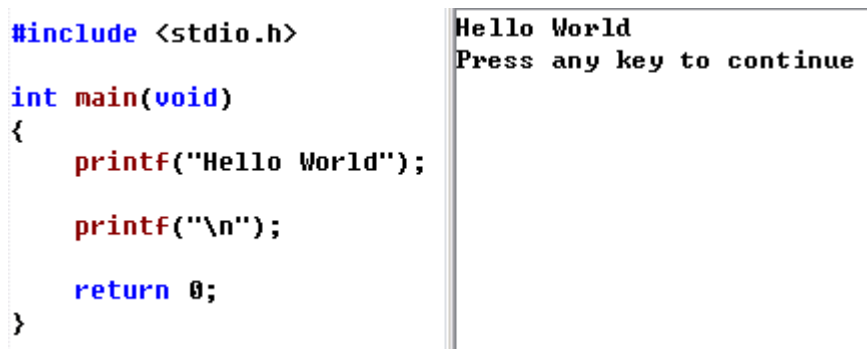
表笔插位同上；其原理同二极管。先假定 A 脚为基极，用黑表笔与该脚相接，红表笔与其他两脚分别接触其他两脚；若两次读数均为 0.7V 左右，然后再用红笔接 A 脚，黑笔接触其他两脚，若均显示“1”，则 A 脚为基极，否则需要重新测量，且此管为 PNP 管。那么集电极和发射极如何判断呢？数字表不能像指针表那样利用指针摆幅来判断，那怎么办呢？我们可以利用“hFE”档来判断：先将档位打到“hFE”档，可以看到档位旁有一排小插孔，分为 PNP 和 NPN 管的测量。前面已经判断出管型，将基极插入对应管型“b”孔，其余两脚分别插入“c”，“e”孔，此时可以读取数值，即 β 值；再固定基极，其余两脚对调；比较两次读数，读数较大的管脚位置与表面“c”，“e”相对应。

第四章 C 语言程序设计基础

本章节希望通过通俗易懂的语言描述，让读者对 C 语言程序设计基础知识有一个大体的认识，因为这仅仅只是一个入门教程。通过本章节，希望读者能明白什么是 C 语言，它有什么作用，并学会简单的 C 程序设计。

本章节的所有程序开发环境是 Visual C++ 6.0，操作系统是 Windows XP SP3。先通过经典的 Hello World 小程序，让读者对 C 语言有个简单的认识。图 4.1 左半部分是一个 C 语言程序，右半部分是这个程序运行的结果。

例 1:



```
#include <stdio.h>

int main(void)
{
    printf("Hello World");

    printf("\n");

    return 0;
}
```

```
Hello World
Press any key to continue
```

图 4.1 hello world 例程

4.1 C 语言简介

C 语言是一种计算机程序设计语言。它既具有高级语言的特点，又具有汇编语言的特点。

就是说，我们可以通过计算机编写 C 语言程序，以实现各种不同的功能，比如数学运算、工业控制等等。举个例子，比如有一道不定方程的数学题，答案有很多个，通过笔算根本算不出来，如果我们将这道题用 C 语言把它描述出来，并通过计算机运算，很快就能得出答案。后面会讲解更多的实例，让读者体会到 C 语言的强大功能。

4.2 C 语言数据类型

C 语言的数据类型有：整型、实型(浮点型)、字符型、数组类型、指针类型、结构体类型、共用体类型等。这几种类型数据的长度和范围随处理器的类型和 C 语言编译环境的不同而有所差异。下面介绍几种常用的数据类型：

4.2.1 整型 (int)

整型就是整数，比如 -16, -1, 0, 1, 8 都是整型数据，在 C 语言中通过关键字 `int` 声明一个整型变量，比如 `int a;` 表示定义变量 `a` 是一个整型变量，它的值只能是整数，不能

是小数。

例 2:

```
#include <stdio.h>

int main(void)
{
    int a;
    a = 10;

    printf("a = %d",a);

    printf("\n");

    return 0;
}
```

```
a = 10
Press any key to continue
```

在例 2 中，通过关键字 `int` 定义了整型变量 `a`，然后给 `a` 赋值 10，结果 `a` 的值是 10，最后输出 `a` 的值，从右面的结果可以看到 `a` 的值是 10。

4.2.2 字符型 (char)

字符型用关键字 `char` 定义字符变量，大多数系统采用 ASCII 字符集。包括以下内容：

字母：A~Z, a~z

数字：0~9

专门符号：29 个：! " # & ' () * 等

空格符：空格、水平制表符、换行等

不能显示的字符：空(null)字符(以 '`\0`' 表示)、警告(以 '`\a`' 表示)、退格(以 '`\b`' 表示)、回车(以 '`\r`' 表示)等

例 3:

```
#include <stdio.h>

int main(void)
{
    char a;
    a = 'X';

    printf("a = %c",a);

    printf("\n");

    return 0;
}
```

```
a = X
Press any key to continue
```

在例 3 中，通过关键字 `char` 定义了字符型变量 `a`，然后给 `a` 赋值大写字母 X，结果 `a` 的值是字母 X，最后输出 `a` 的值，从右面的结果可以看到 `a` 的值是字母 X。

4.2.3 实型 (也称浮点型) (float)

实型也称浮点型，由关键字 `float` 定义，表示小数，并且小数点后有 6 位有效数字。

例 4

```

#include <stdio.h>

int main(void)
{
    float a;
    a = 1;

    printf("a = %f",a);
    printf("\n");

    return 0;
}

```

```

a = 1.000000
Press any key to continue

```

在例 4 中，通过关键字 `float` 定义了实型变量 `a`，然后给 `a` 赋值 1，但从右面的结果可以看，`a` 的值并不是 1，而是 1.000000，这就是实型数据（带小数点），虽然赋给 `a` 的值是整数 1，但由于 `a` 是实型变量，所以最后输出结果中 `a` 的值是 1.000000。

以上是最基本、最常用的三种数据类型，由于是入门教程，篇幅有限，所以其它的数据类型请读者查阅相关书籍及资料，自行学习。

4.3 输入与输出函数

C 语文中标准输入/输出函数有 `printf`、`fprintf`、`scanf`、`fscanf`、`putchar`、`puts`、`fputc`、`fputs`、`getchar`、`gets`、`fgetc`、`fgets`。其中 `printf` 和 `scanf` 相对较简单，以下主要介绍这两个函数的使用方法。

标准输入函数 `scanf` 可以实现从键盘输入的功能，而标准输出函数 `printf` 可以将信息输出的计算机显示器上。`printf` 和 `scanf` 后面括号中的“%d”表示以整型变量输入或者输出，对应的字符型和浮点型分别是“%c”、“%f”，引号内其他字符按原样输出，“\n”表示换行(换行的意思就是结束当前行，跳到下一行)。`printf` 函数在前面的实例中已经出现过，通过下面的实例介绍 `scanf` 和 `printf` 函数的用法。例 5 中通过 `scanf` 函数输入 `a` 的值，然后通过 `printf` 函数将 `a` 的值输出，结果见上右图。

例 5

```

#include <stdio.h>

int main(void)
{
    int a;

    printf("Please input a: ");
    scanf("%d",&a);

    printf("a = %d",a);

    printf("\n");

    return 0;
}

```

```

Please input a: 100
a = 100
Press any key to continue

```

4.4 关系运算符

C 语言中的关系运算符有以下 13 种：

| | | | |
|----|---------|----|---------|
| + | ： 正号运算符 | = | ： 赋值运算符 |
| - | ： 负号运算符 | == | ： 等于运算符 |
| * | ： 乘法运算符 | >= | ： 大于等于 |
| / | ： 除法运算符 | <= | ： 小于等于 |
| % | ： 求余运算符 | - | ： 减法运算符 |
| + | ： 加法运算符 | -- | ： 自减运算符 |
| ++ | ： 自加运算符 | | |

C 语言中的运算符有优先级之分，所谓优先级就是指运算顺序，比如 $21+4/2-2*(4+2)$ 这个算式中括号中的 $(4+2)$ 最先运算，然后再运算除法和乘法，最后再运算加法和减法，基本上按照“先算括号，再算乘除，最后算加减”的原则。

重点介绍自增(++)、自减(--)、== 运算符：

作用是使变量的值加 1 或减 1

++i, --i: 在使用 i 之前，先使 i 的值加（减）1

i++, i--: 在使用 i 之后，使 i 的值加（减）1

例如：若 $a=5$; $b=++a$; 等价于 $a=a+1$; $b=a$;

$b=a++$; 等价于 $b=a$; $a=a+1$;

注意：++和--的运算对象只能是变量，不能是常量。例如 $5++$ 、 $(a+2)++$ 不合法，尽量不采用引起歧义的写法： $i+++j$ 等价于 $(i++)+j$

== 运算符：在 C 语言中如果表示两个数相同，应该用两个等号表示，如“ $a==b$ ”表示 a 和 b 相等，而一个等号表示赋值，这点是初学者很容易犯的错误。

例 6

```
#include <stdio.h>

int main(void)
{
    int a,b,c,d;
    int result1,result2;

    a = 47;b = 38;c = 20,d = 12;

    result1 = a+b-c*d;
    result2 = (a-d)*3-b;

    printf("%d\n",result1);
    printf("%d\n",result2);

    return 0;
}
```

```
-155
67
Press any key to continue
```


例 6 中实现了两个最基本的运算，结果见上右图。

4.5 逻辑运算符

C 语言中的逻辑运算符有三种：**&&**(与逻辑)、**||**(或逻辑)、**!**(非逻辑)

4.4.1 与逻辑(&&)

```
if(a == 1 && b == 2) c=1;
```

上面语句表示的意思是：只有当 a 的值为 1 且 b 的值为 2 时，才把 1 赋值给 c，也就是说，只有当与逻辑(&&)的左边和右边两个条件都满足时才执行相应的操作，否则不执行。

或逻辑(||)

```
if(a == 1 || b == 2) c=1;
```

上面语句表示的意思是：当 a 的值为 1 或者 b 的值为 2 时，就把 1 赋值给 c，也就是说，当或逻辑(||)的左边和右边的条件只要有一个满足就执行相应的操作，只有当两个条件都不满足时才不执行。

非逻辑(!)

```
a = 1;
```

```
if(!(a==1)) b=1;
```

上面语句表示的意思是：“如果非(a 等于 1)，则把 1 赋值给 b”，括号内的“a==1”是成立的，但在其前面加了一个“非”，那么“!(a==1)”就变成假条件了，也就是不成立了，所以“b=1;”不会被执行的。通俗的说就是“在条件为真的表达式前面加上非，条件就变成假了；在条件为假的表达式前面加上非，条件就变成真了”。

例 7

```
#include <stdio.h>
int main(void)
{
    int a1,b1,c1,a2,b2,c2,c3;
    a1 = 1; b1 = 2;
    a2 = 3; b2 = 4;
    c1 = 100;c2 = 200;c3 = 300;
    if(a1==1 && b1==2)
        c1 = 10;
    if(a2==2 || b2==4)
        c2 = 20;
    if(!(a1==1))
        c3 = 30;

    printf("%d\n",c1);
    printf("%d\n",c2);
    printf("%d\n",c3);

    return 0;
}
```

```
10
20
300
Press any key to continue
```

读者根据前面的介绍以及程序运行的结果，自己分析输出结果的由来。

4.6 if 条件选择语句

if 语句的基本格式如下：

```
if(表达式) 语句 1  
[ else 语句 2 ]
```

其表示意思是：如果括号内的表达式的值为真，则执行语句 1；如果有 else 语句，当 if 括号内的表达式值为假时，则执行语句 2。

注：[]内表示可选语句，可以有也可以没有。

最常用的 3 种 if 语句形式：

(1) if(表达式) 语句 1 (没有 else 子句)

(2) if(表达式) 语句 1
 else 语句 2 (有 else 子句)

(3) if(表达式 1) 语句 1
 else if(表达式 2) 语句 2
 else if(表达式 3) 语句 3

 else if(表达式 m)语句 m
 else 语句 m+1
(在 else 部分又嵌套了多层的 if 语句)

在 if 语句中又包含一个或多个 if 语句称为 if 语句的嵌套如下：

```
if(表达式 1)  
{  
    if(表达式 2) 语句 1  
}  
else 语句 2
```

例 8

```

#include <stdio.h>

int main(void)
{
    int a1,b1,c1,a2,b2,c2,c3;
    a1 = 1; b1 = 2;a2 = 3; b2 = 4;
    /* 单个if使用方法 */
    if(a1 == 1)
        printf("a1 = %d\n",a1);

    /* if...else...使用方法 */
    if(a1 == 2)
        printf("a1 = %d\n",a1);
    else
        printf("a1 is not equal 2\n");

    /* if...else if...else使用方法 */
    if(a2 == 1)
        printf("a2 = 1\n");
    else if(a2 == 2)
        printf("a2 = 2\n");
    else if(a2 == 3)
        printf("a2 = 3\n");
    else
        printf("a2 is not exist \n");

    return 0;
}

```

```

a1 = 1
a1 is not equal 2
a2 = 3
Press any key to continue

```

if 语句是很常用的语句,是 C 语言中最基本的条件选择语句,输出结果请读者自行分析,实例 7 中出现的“/**/”表示注释,即对程序的一个解释说明,还有另外一种注释方法是“//”,不同的是“//”只能注释单行,而“/**/”可以注释多行。注释部分不会被编译,更不会被运行,仅仅是为了让别人对你写的程序更好的理解,建议读者写程序时最好写上注释,以增加程序的可读性。

4.7 switch...case 条件选择语句

第二种条件选择语句是 witch...case 语句,可以实现多分支选择结构,也是很常用的,其基本格式如下:

```

Switch(表达式) //表达式为整数类型(包括字符型)
{
    case 常量 1 : 语句 1;break;
    case 常量 2 : 语句 2;break;
        ⋮      ⋮      ⋮
    case 常量 n : 语句 n;break;
    default      : 语句 n+1;break;
}

```

```
}
```

其表示的意思是：如果某个 case 后面的常量和表达式相等，则执行该 case 后的语句，而其它 case 和 default 语句后面的语句均不会被执行；如果所有 case 后面的常量和表达式均不相等，则执行 default 后面的语句，且其它 case 后面的语句均不会被执行。

注意：常量 1~常量 n 必须都不一样。

例 9

成绩的等级规定如下：A 等为 85 分以上，B 等为 70~84 分，C 等为 60~69 分，D 等为 60 分以下。假如某同学的成绩等级为 B，求该同学的成绩范围是多少？

```
#include<stdio.h>
int main(void)
{
    char grade='B';
    switch(grade)
    {
        case 'A':printf("Your score is range 85 to 100 !\n");break;
        case 'B':printf("Your score is range 70 to 84 !\n");break;
        case 'C':printf("Your score is range 60 to 69 !\n");break;
        case 'D':printf("Your score is range 0 to 59 !\n");break;
        default :printf("Error input!\a\n");break;
    }
    printf("\n");
    return 0;
}
```

```
Your score is range 70 to 84 !
Press any key to continue_
```

例 9 运行结果如上右图，switch...case 是很常用也是很重要的条件选择结构语句，读者一定要好好体会和理解。

4.8 for 循环语句

一般形式如下：

```
for(表达式 1; 表达式 2; 表达式 3)
```

```
{
    循环体语句;
}
```

for 语句的功能就是循环作用，其执行顺序为：先计算表达式 1，再判断表达式 2，如果表达式 2 的判断结果为真，则执行“循环体语句”，执行完“循环体语句”后继续执行表达式 3，再判断表达式 2，如果为真继续执行“循环体语句”，如果为假则结束整个 for 循环语句，如此循环执行；如果表达式 2 的判断结果为假，则不执行“循环体语句”，并结束整个 for 循环语句。

注意：表达式 1、表达式 2、表达式 3 都可以省略，但是分号不能省。

例 10

用 for 语句求 100 个自然数的和，即： $s=1+2+3+\dots+100$ 。

```
#include <stdio.h>

int main(void)
{
    int i,sum=0;

    for(i=0;i<101;i++)
    {
        sum = sum + i;
    }
    printf("sum = %d\n",sum);

    return 0;
}
```

```
sum = 5050
Press any key to continue
```

通过例 10，读者就知道 C 语言是如何轻而易举实现数学运行的了，如果是一些没有规律的计算题，用 C 语言就可以轻松计算出来了！

4.9 while 及 do...while 循环语句

while 语句的一般形式如下：

```
while(表达式)
```

```
循环体语句;
```

(1) 表达式部分一般为关系表达式或逻辑表达式，也可以是 C 语言其他类型的合法表达式，用来控制“循环体语句”是否执行。

(2) “循环体语句”部分称为内嵌语句，可以是基本语句、控制语句，也可以是复合语句，是循环重复执行的部分，当语句多于一句时，必须加上大括号“{}”，详细说明见实例 10。

(3) do---while 语句

do---while 语句的特点：先无条件地执行“循环体语句”，然后判断循环条件是否成立，如果成立，则继续执行“循环体语句”，否则结束整个循环结构。

一般形式如下：

```
do
{
    循环体语句;
} while (表达式); //注意此处的分号一定不要忘了
```

while、do...while 以及 for 循环结构是三种最基本的、也是最常用的循环结构，具体用哪一种结构，视实际情况而定，一般情况下这三种结构都可以实现相同的功能。读者无比把这三种循环结构学会，使用时做到得心应手。

例 11

分别用 while 和 do...while 求 100 个自然数的和，即： $s=1+2+3+\dots+100$ 。

```
#include <stdio.h>

int main(void)
{
    int i=0,sum=0;

    while(i<100)
    {
        i++;
        sum += i;
    }
    /* 以下写法也可以，请理解自加
       运算符(++在前和在后的区别
    while(i++ < 100)
        sum += i;
    */

    printf("Sum = %d\n",sum);

    return 0;
}

Sum = 5050
Press any key to continue
```

```
#include <stdio.h>

int main(void)
{
    int i=0,sum=0;
    do
    {
        sum += ++i;
        //sum += i++;注意这两句的区别
    }while(i < 100);

    printf("Sum = %d\n",sum);

    return 0;
}

Sum = 5050
Press any key to continue
```

以上程序均实现了题目要求，三种循环结构有时也能互相转化，但有些情况是不能相互转化的，当你程序写的多了，自然就知道什么时候用什么样的结构了。

4.10 函数

C 语言中函数并不是我们在数学中学过的函数，函数的存在为我们写程序提供了方便，主要用在多次重复操作的过程中，先看下面的实例。

例 12

```
#include <stdio.h>

int main(void)
{
    int a,b,c,d,e,f;
    a=1,b=2,c=3,d=4;

    e = (a+b)*0.01+a*100+b;
    f = (c+d)*0.01+c*100+d;

    printf("e = %d\n",e);
    printf("f = %d\n",f);
    return 0;
}

e = 102
f = 304
Press any key to continue
```

```
#include <stdio.h>

int MyAdd(int x,int y); //函数声明

int main(void)
{
    int a,b,c,d,e,f;
    a=1,b=2,c=3,d=4;

    e = MyAdd(a,b);
    f = MyAdd(c,d);

    printf("e = %d\n",e);
    printf("f = %d\n",f);
    return 0;
}
// 定义MyAdd函数
int MyAdd(int x,int y)
{
    int sum;
    sum = (x+y)*0.01+x*100+y;
    return sum;
}

e = 102
f = 304
Press any key to continue
```

例 12 中，第一个程序没有使用函数，第二个函数使用了函数，最后结果是一样的，这里只举了两个数做运算的情况，可能还不能完全体现出使用函数的方便之处，但如果有 1000 个甚至更多的算式，例 5 中通过 scanf 函数输入 a 的值，然后通过 printf 函数将 a 的值输出，结果见上右图。

使用函数就会达到事半功倍的效果。也就是说，同样的操作我们不用重复编写，只需要把它写成一个函数，然后在需要时调用它就可以了。下面解释以上程序中出现的语句。

“`e=MaAdd(a,b);`”和“`f=MyAdd(c,d);`”两条语句中，调用了函数 `MyAdd`，其中的 `a,b,c,d` 分别称为实参(实际参数)，而在定义 `MyAdd` 函数时的 `x` 和 `y` 则称为形参(形式参数)，通过调用函数使 `a(c)` 和 `x`、`b(d)` 和 `y` 建立了一一对应的关系，将 `a(c)` 和 `b(d)` 的值传递给了 `x` 和 `y`，在 `MyAdd` 函数中的最后一句“`return sum;`”表示将运算结果返回给调用该函数处，此处为变量 `e(f)`，因此 `e(f)` 便得到了运算后的结果，请读者细心体会。

C 语言的知识远不止这些，此处所介绍的只是入门的基本知识，想要学好 C 语言还得花更多的时间去学习和理解。譬如，指针没有讲解，指针被称为 C 语言的精髓，这里没有提及是因为指针比较难，并不属于入门的知识，也不是三言两语就能够说清楚的，所以这里就不讲解指针的知识，对于初学者，先把前面讲的知识掌握好，再深入学习为好。

最后给读者推荐学习 C 语言经典著作，分别是 C 程序设计语言(第 2 版·新版)(`The C Programming Language`)、C 和指针、C 专家编程、C 陷阱与缺陷、C 语言深度解剖、高质量 C++C 编程指南，这几本都是非常不错的 C 语言书籍，希望读者好好学习，天天向上！

第五章 单片机基本知识与例程分析

本章主要介绍单片机的基本知识以及简单的例程分析。介绍了单片机的最小系统电路以及如何用单片机点亮一个发光二极管。

5.1、单片机概述

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。

常用单片机: 8051、AVR、PIC、MSP430、S08、9S12、ARM 等。

5.2、Keil 软件介绍

Keil C51 是美国 Keil Software 公司出品的 51 系列兼容单片机 C 语言软件开发系统,与汇编相比, C 语言在功能上、结构性、可读性、可维护性上有明显的优势,因而易学易用。Keil 提供了包括 C 编译器、宏汇编、连接器、库管理和一个功能强大的仿真调试器等在内的完整开发方案,通过一个集成开发环境(uVision)将这些部分组合在一起。运行 Keil 软件需要 WIN98、NT、WIN2000、WINXP 等操作系统。如果你使用 C 语言编程,那么 Keil 几乎就是你的不二之选,即使不使用 C 语言而仅用汇编语言编程,其方便易用的集成环境、强大的软件仿真调试工具也会令你事半功倍。

5.3 用单片机点亮一个 LED

5.3.1 单片机最小系统框图设计

对 51 系列单片机来说,单片机要正常工作,必须具有五个基本电路,也称五个工作条件: 1、电源电路, 2、时钟电路, 3、复位电路, 4、程序存储器选择电路, 5、外围电路。因此,单片机最小系统一般应该包括单片机、晶振电路、复位电路、外围电路等,如下面的框图所示:

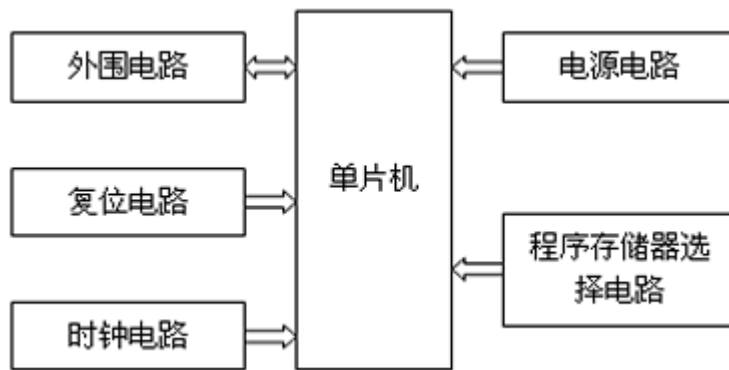


图 5.1 单片机最小系统框图

5.3.2 单片机最小系统原理图设计

1. 电源电路设计

单片机芯片的第 40 脚为正电源引脚 VCC，一般外接+5V 电压。第 20 脚为接地引脚 GND，常见电源电路设计如图 2 所示。

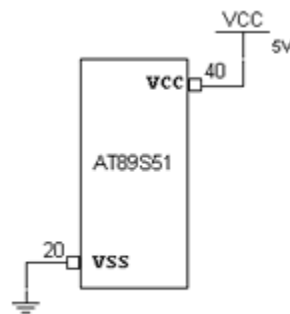


图 2 单片机电源电路

2. 时钟电路设计

单片机是一种时序电路，必须要有时钟信号才能正常工作。单片机芯片的 18 脚 (XTAL2)、19 脚 (XTAL1) 分别为片内反向放大器的输出端和输入端，只要在 18 脚 (XTAL2) 和 19 脚 (XTAL1) 之间接上一个晶振，再加上 2 个 30PF 的瓷片电容即可构成单片机所需的时钟电路。常见的时钟电路如下。

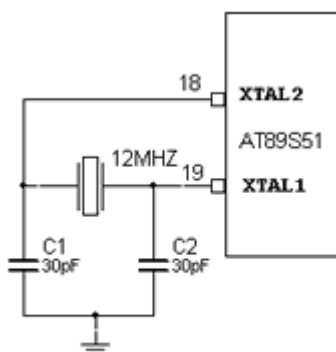


图3 时钟电路

注意：当采用外部时钟时，19脚（XTAL1）接地，18脚（XTAL2）接外部时钟信号，本文就不详细讨论了。

3. 复位电路的设计

单片机芯片的第9脚 RST（Reset）是复位信号输入端。单片机系统在开机时或在工作中因干扰而使程序失控，或工作中程序处于某种死循环状态等情况下都需要复位。复位的作用是使中央处理器 CPU 以及其他功能部件都恢复到一个确定的初始状态，并从这个状态开始工作。MCS-51 系列单片机的复位靠外部电路实现，信号从 RST 引脚输入，高电平有效，只要保持 RST 引脚高电平 2 个机器周期，单片机就能正常复位。常见的复位电路有上电复位电路和按键复位电路二种：

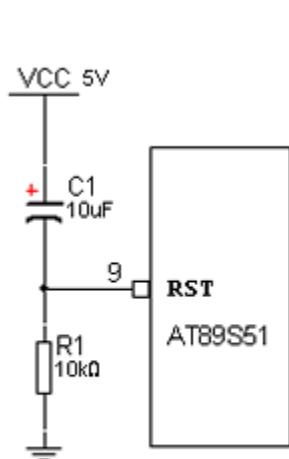


图 4 上电复位电路

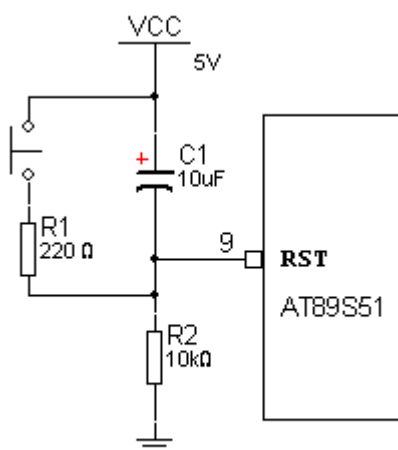


图 5 按键复位电路

4. 程序存储器选择电路

单片机芯片的第 31 脚 (EA) 为内部与外部程序存储器选择输入端。当 EA 引脚接高电平时, CPU 先访问片内 4KB 的程序存储器, 执行内部程序存储器中的指令, 当程序计数器超过 0FFFH 时, 将自动转向片外程序存储器, 既是从 1000H 地址单元开始执行指令; 当 EA 引脚接低电平时, 不管片内是否有程序存储器, CPU 只访问片外程序存储器。

AT89S51 内部有 4KB 的程序存储器, 所以根据该脚的引脚功能, 只要将该脚接上高电平, 才能先从片内程序存储器开始取指令。

常见的程序存储器选择电路就是将第 31 脚直接接到正电源上。

5. 外围电路的设计

外围电路的设计主要依据项目要实现的功能, 本项目要实现的功能是用单片机控制一个 LED, 先来分析下面的 LED 工作原理图:

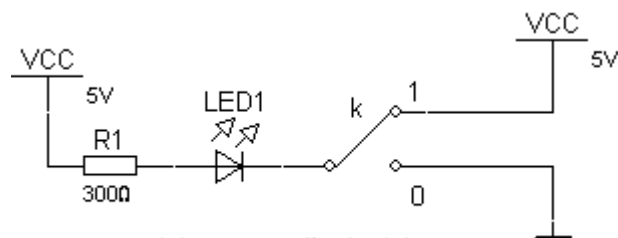


图 6 LED工作原理图

如果用“1”表示高电平，“0”表示低电平。当开关 K 接上高电平，既是 K=1 时，LED1 不亮；当开关 K 接上低电平，既是 K=0 时，LED1 亮。LED1 的亮与灭完全受开关 K 的控制。

如果把开关 K 换成单片机的第 1 个引脚，电路设计如下:

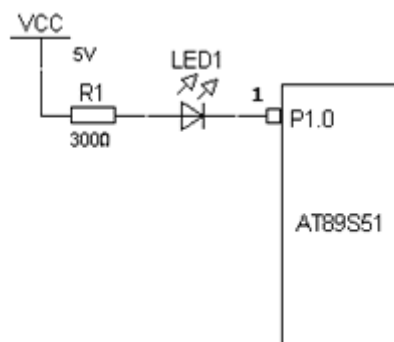


图 7 单片机外围电路设计

单片机的主要控制功能是通过单片机的 I/O 口按不同时序输出不同的高低电平控制外

部的电路实现特定的功能。依据单片机的 P0、P1、P2、P3 口的功能特点和单片机最小系统要实现简单的点亮一个 LED 的功能，我们采用 P1.0 引脚作为控制一个 LED 的端口，依据单片机的引脚功能和单片机工作的条件，综合上面的设计电路，设计的 AT89S51 单片机最小应用系统如下图 8 所示，电路原理图设计如下：

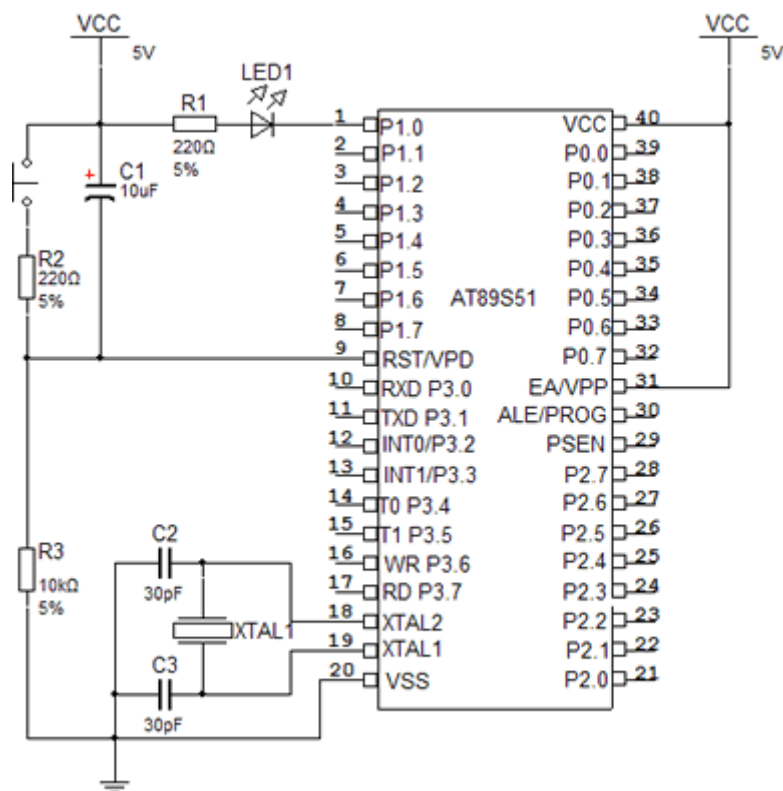


图 8 单片机最小系统图

5.3.3 单片机最小系统程序编写

当单片机最小应用系统的硬件电路设计完毕后，单片机最小应用系统的设计工作还没有完成，因为单片机产品需要硬件+软件共同支持才能正常工作，这里提到的软件主要是指用户编写的源程序。这也是区别于传统的电子产品的地方。

本系统要实现的主要功能是点亮一个 LED，根据外围电路的设计，只需要通过指令控制单片机的第一个引脚输出低电平，就可以使第一个 LED 发光。

采用汇编语言编程序如下：

ORG 0000H ; 起始伪指令 ORG, 指示随后的指令代码从 0000H 地址单元开始存放。

LJMP START ; 跳转到标号 START 处去执行。

ORG 0030H ; 起始伪指令 ORG, 指示随后的指令代码从 0030H 地址单元开始存放。

START: CLR P1.0; 第一个引脚输出低电平, 点亮 LED。

END ; 结束伪指令, 说明程序到此结束。

采用 C 语言编程如下:

```
#include<reg52.h>           //包含头文件
sbit led1=P1^0;             //定义变量 led1 为 P1.0, 也就是单片机的第一个引脚。
void main()                 //主程序开始, C 语言唯一的一个主程序。
{
    led1=0;                 //让单片机的第一个引脚输出低电平, 点亮 LED。
}
```

5.4 数码管应用

数码管的使用方法与发光二极管没什么区别, 只是把七或八只发光二极管组合在一个模块上组成了个 8 字和小数点, 用以显示数字。为了减少管脚, 把各个发光管的其中同一个极接在一起作为共用点, 因此就产生了共阳极和共阴极数码之说。共阳极就是把各个发光管的正极接在一起, 而共阴极就刚好相反, 如图 9 所示。

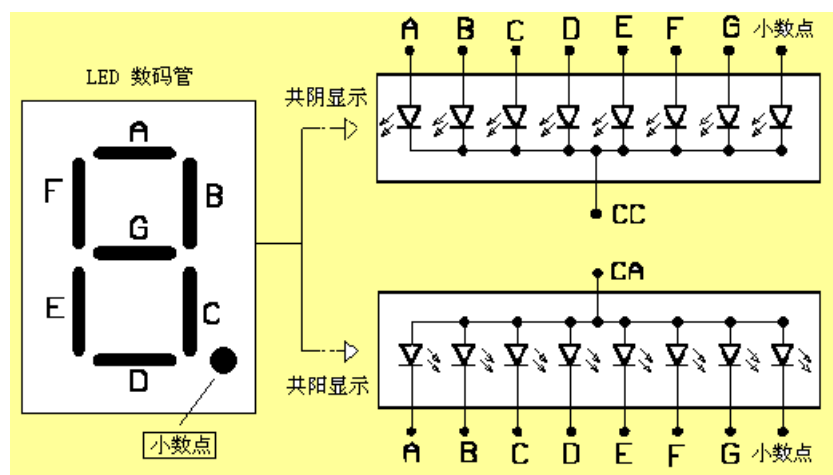


图 9 数码管电路图

一般来说大部分的逻辑 IC 的吸收电流要强于输出电流。因此, 大家都爱使用共阴极的数码管, 因为可选的 IC 多些。

共阴和共阳结构的 LED 数码管显示器各笔划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 hgfedcba 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。例如，对于共阴 LED 数码管显示器，当公共阴极接地（为零电平），而阳极 hgfedcba 各段为 0111011 时，数码管显示器显示“P”字符，即对于共阴极 LED 数码管显示器，“P”字符的字形码是 73H。如果是共阳 LED 数码管显示器，公共阳极接高电平，显示“P”字符的字形代码应为 10001100（8CH）。这里必须注意的是：很多产品为方便接线，常不按规则的办法去对应字段与位的关系，这个时候字形码就必须根据接线来自行设计了。

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到接口电路，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。

动态扫描就是指我们采用分时的办法，轮流控制各个显示器的 COM 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。

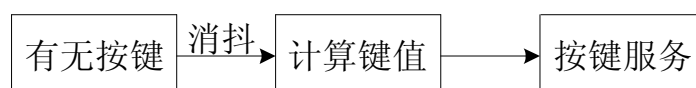
其接口电路是把所有显示器的 8 个笔划段 a-h 同名端连在一起，而每一个显示器的公共极 COM 是各自独立地受 I/O 线控制。CPU 向字段输出口送出字形码时，所有显示器接收到相同的字形码，但究竟是那个显示器亮，则取决于 COM 端，而这一端是由 I/O 控制的，所以我们就能自行决定何时显示哪一位了。

5.5 键盘检测

5.5.1 键盘的分类

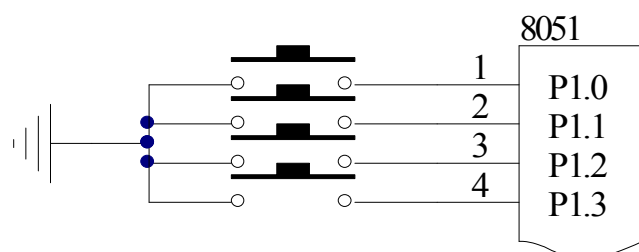
键盘分编码键盘和非编码键盘。键盘上闭合键的识别由专用硬件编码器实现，并产生键编号或键值的称为编码键盘，如 BCD 码键盘、ASCLL 码键盘等；而靠软件来识别的称为非编码键盘；在单片机组成的测控系统及智能化仪器中，用的最多的是非编码键盘。

非编码键盘有分为：独立式非编码键盘和行列式非编码键盘（矩阵键盘）。无论是何种按键，其功能实现都是分为三个步骤：



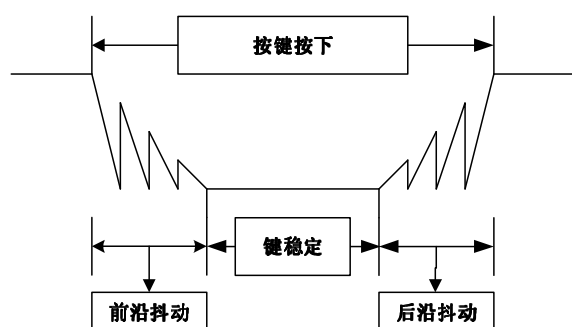
5.5.2 独立按键

1. 独立按键原理



单片机中除了 P0 接口,其它 I/O 接口内部都有上拉电阻。上图所示 P1.0-P1.3 接了 4 个按键,当没有按下按键时对应的 I/O 接口为高电平。当按下按键时,对应管脚会变为低电平。因此只要检测到对应管脚是否为低电平就知道此按键是否被按下。

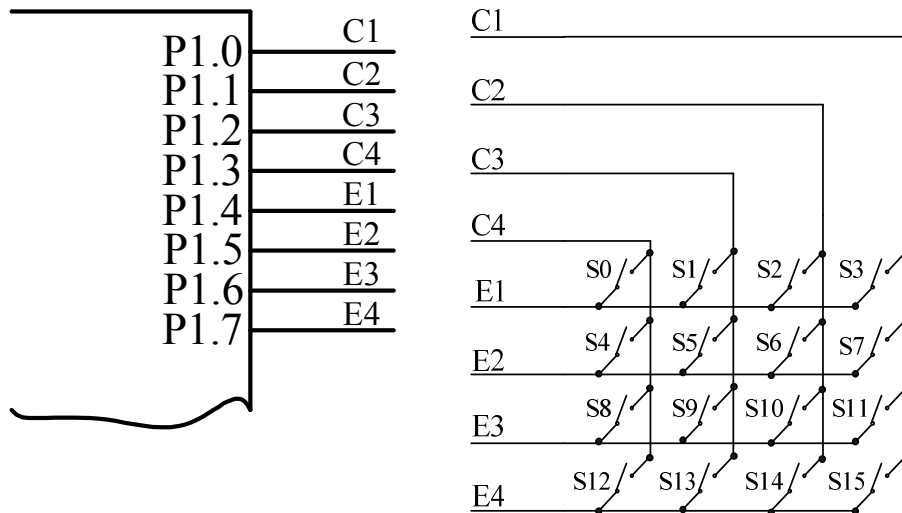
2. 按键消抖



当某个按键被按下,由于其触点的机械特性,会抖动约 20ms,才会稳定下来,如图所示的波形。如果在此期间去检测,很可能产生误判,所以检测到有键被按下并延时 20ms 避开抖动期,然后才去检测是哪个被按下。另外手松开也有抖动,所以很有可能在刚才判断时有按键,等延时 20ms 之后,就没有按键了。为了提高效率,应在延时 20ms 后再次判断一下,是否有键被按下。

5.5.3 矩阵按键

1. 矩阵按键的原理



在矩阵式键盘中，每条水平线和垂直线在交叉处不直接连通，而是通过一个按键加以连接。这样，一个端口（如 P1 口）就可以构成 $4 \times 4 = 16$ 个按键，比之直接将端口线用于键盘多出了一倍，而且线数越多，区别越明显，比如再多加一条线就可以构成 20 键的键盘，而直接用端口线则只能多出一键（9 键）。由此可见，在需要的键数比较多时，采用矩阵法来做键盘是合理的。矩阵式结构的键盘显然比独立键盘要复杂一些。

第六章 单片机常用外围器件

6.1 蜂鸣器

蜂鸣器是一种一体化结构的电子讯响器，采用直流电压供电，广泛应用于计算机、打印机、复印机、报警器、电子玩具、汽车电子设备、电话机、定时器等电子产品中作发声器件。蜂鸣器主要分为压电式蜂鸣器和电磁式蜂鸣器两种类型。

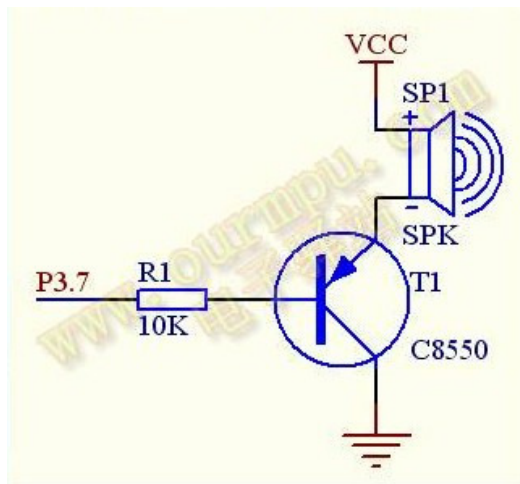


图 6.1 蜂鸣器电路图



图 6.2 蜂鸣器实物图

6.2 继电器

继电器是一种电控制器件。它具有控制系统（又称输入回路）和被控制系统（又称输出回路）之间的互动关系。通常应用于自动化的控制电路中，它实际上是用小电流去控制大电

流运作的一种“自动开关”。故在电路中起着自动调节、安全保护、转换电路等作用。

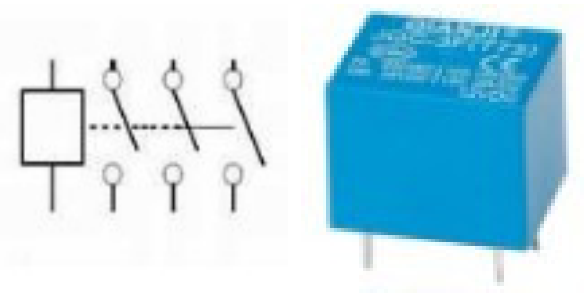


图 6.3 继电器表示符号及实物图

6.3 直流电机

电机是指依据电磁感应定律实现电能转换或传递的一种电磁装置。电动机也俗称马达)，在电路中用字母“M”（旧标准用“D”）表示。它的主要作用是产生驱动转矩，作为用电器或各种机械的动力源。发电机在电路中用字母“G”表示。它的主要作用是利用机械能转化为电能，目前最常用的是，利用热能、水能等推动发电机转子来发电。

6.4 步进电机

步进电机是将电脉冲信号转变为角位移或线位移的开环控制元步进电机件。在非超载的情况下，电机的转速、停止的位置只取决于脉冲信号的频率和脉冲数，而不受负载变化的影响，当步进驱动器接收到一个脉冲信号，它就驱动步进电机按设定的方向转动一个固定的角度，称为“步距角”，它的旋转是以固定的角度一步一步运行的。可以通过控制脉冲个数来控制角位移量，从而达到准确定位的目的；同时可以通过控制脉冲频率来控制电机转动的速度和加速度，从而达到调速的目的。



6.5 AD 与 DA

6.5.1 什么是 A/D、D/A 转换

随着数字技术，特别是信息技术的飞速发展与普及，在现代控制、通信及检测等领域，为了提高系统的性能指标，对信号的处理广泛采用了数字计算机技术。由于系统的实际对象往往都是一些模拟量（如温度、压力、位移、图像等），要使计算机或数字仪表能识别、处理这些信号，必须首先将这些模拟信号转换成数字信号；而经计算机分析、处理后输出的数字量也往往需要将其转换为相应模拟信号才能为执行机构所接受。这样，就需要一种能在模拟信号与数字信号之间起桥梁作用的电路--模数和数模转换器。

将模拟信号转换成数字信号的电路，称为模数转换器（简称 A/D 转换器或 ADC , analog to digital converter）；将数字信号转换为模拟信号的电路称为数模转换器（简称 D/A 转换器或 DAC, digital to analog converter）；A/D 转换器和 A/D 转换器为确保系统处理结果的精确度，A/D 转换器和 D/A 转换器必须具有足够的转换精度；如果要实现快速变化信号的实时控制与检测，A/D 与 D/A 转换器还要求具有较高的转换速度。转换精度与转换速度是衡量 A/D 与 D/A 转换器的重要技术指标。随着集成技术的发展，现已研制和生产出许多单片的和混合集成型的 A/D 和 D/A 转换器，它们具有愈来愈先进的技术指标。

6.5.2 D/A 和 A/D 转换器的相关性能参数

D/A 转换器是把数字量转换成模拟量的线性电路器件，已做成集成芯片。由于实现这种转换的原理和电路结构及工艺技术有所不同，因而出现各种各样的 D/A 转换器。目前，国外市场已有上百种产品出售，他们在转换速度、转换精度、分辨率以及使用价值上都各具特色。衡量一个 D/A 转换器的性能的主要参数有：

(1) 分辨率

是指 d/a 转换器能够转换的二进制数的位数，位数多分辨率也就越高。

(2) 转换时间

指数字量输入到完成转换，输出达到最终值并稳定为止所需的时间。电流型 D/A 转换较快，一般在几 ns 到几百 ns 之间。电压型 d/a 转换较慢，取决于运算放大器的响应时间。

(3) 精度

指 D/A 转换器实际输出电压与理论值之间的误差，一般采用数字量的最低有效位作为衡量单位。

(4) 线性度

当数字量变化时，D/A 转换器输出的模拟量按比例关系变化的程度。理想的 D/A 转换器是线性的，但是实际上是有误差的，模拟输出偏离理想输出的最大值称为线性误差。

A/D 转换器的功能是把模拟量转换成数字量。由于实现这种转换的工作原理和采用工艺技术不同，因此生产出种类繁多的 A/D 转换芯片。A/D 转换器按分辨率分为 4 位、6 位、8 位、10 位、14 位、16 位和 bcd 码的 3 1/2 位、5 1/2 位等。按照转换速度可分为超高速（转换时间 $\leq 330\text{ns}$ ），次超高速（ $330\sim 3.3\mu\text{s}$ ），高速（转换时间 $3.3\sim 333\mu\text{s}$ ），低速（转换时间 $> 330\mu\text{s}$ ）等。A/D 转换器按照转换原理可分为直接 A/D 转换器和间接 A/D 转换器。所谓直接 A/D 转换器，是把模拟信号直接转换成数字信号，如逐次逼近型，并联比较型等。其中逐次逼近型 A/D 转换器，易于用集成工艺实现，且能达到较高的分辨率和速度，故目前集成化 A/D 芯片采用逐次逼近型者多；间接 A/D 转换器是先把模拟量转换成中间量，然后再转换成数字量，如电压/时间转换型（积分型），电压/频率转换型，电压/脉宽转换型等。其中积分型 A/D 转换器电路简单，抗干扰能力强，切能作到高分辨率，但转换速度较慢。有些转换器还将多路开关、基准电压源、时钟电路、译码器和转换电路集成在一个芯片内，已超出了单纯 A/D 转换功能，使用十分方便。

第七章 常用软件及其功能

本章主要介绍电子设计中常用到的几款软件。如果只是模拟电子电路和数字电子电路的仿真，我们首先想到的是 Multisim。如果是带有单片机的仿真我们使用 Protues。单片机软件编程我们会使用 Keil。原理图的绘制我们会用到 Protell 或者 Altium Designer。由于篇幅所限，我们这里只是简单介绍各个软件，软件的下载、安装与学习请从网上搜索相关资源。

7.1 Multisim

Multisim 是美国国家仪器（NI）有限公司推出的以 Windows 为基础的仿真工具，适用于板级的模拟/数字电路板的设计工作。它包含了电路原理图的图形输入、电路硬件描述语言输入方式，具有丰富的仿真分析能力。



最新版：Multisim11

7.1.1 可进行仿真的内容

1. 器件建模及仿真；
2. 电路的构建及仿真；
3. 系统的组成及仿真；
4. 仪表仪器原理及制造仿真。
5. 可以进行建模及仿真的器件：模拟器件（二极管，三极管，功率管等）、数字器件（74 系列，COMS 系列，PLD，CPLD 等）、FPGA 器件。

技巧： Multisim 的元件库中有小部分常用器件没有包含，如三极管 S9018、S9014 等，读者有兴趣可以查阅相关资料在 Multisim 中导入 S9018、S9014 的仿真模型，从而在该环境中实现对 S9018 等器件的仿真，通常我们的更常用的方法是采用性能相近的器件来代替，从而实现仿真。

7.1.2 Multisim 的特点

- (1) 通过直观的电路图捕捉环境，轻松设计电路
- (2) 通过交互式 SPICE 仿真，迅速了解电路行为
- (3) 借助高级电路分析，理解基本设计特征
- (4) 通过一个工具链，无缝地集成电路设计和虚拟测试
- (5) 通过改进、整合设计流程，减少建模错误并缩短上市时间

7.2 Keil

Keil C51 是美国 Keil Software 公司出品的 51 系列兼容单片机 C 语言软件开发系统，与汇编相比，C 语言在功能上、结构性、可读性、可维护性上有明显的优势，因而易学易用。Keil 提供了包括 C 编译器、宏汇编、连接器、库管理和一个功能强大的仿真调试器等在内的完整开发方案，通过一个集成开发环境 (uVision) 将这



些部分组合在一起。运行 Keil 软件需要 WIN98、NT、WIN2000、WINXP 等操作系统。如果你使用 C 语言编程，那么 Keil 几乎就是你的不二之选，即使不使用 C 语言而仅用汇编语言编程，其方便易用的集成环境、强大的软件仿真调试工具也会令你事半功倍。

最新版: uvision4

Keil 除了 Keil C51 用于开发单片机以外，还有 Keil MDK 可用于开发 ARM，读者有兴趣将来可以尝试使用该软件进行 ARM 的开发。

7.3 Protues

Protues 软件是英国 Labcenter electronics 公司出版的 EDA 工具软件。它不仅具有其它 EDA 工具软件的仿真功能，还能仿真单片机及外围器件。它是目前最好的仿真单片机及外围器件的工具。虽然目前国内推广刚起步，但已受到单片机爱好者、从事单片机教学的教师、致力于单片机开发应用的科技工作者的青睐。



Proteus 是世界上著名的 EDA 工具(仿真软件)，从原理图布图、代码调试到单片机与外围电路协同仿真，一键切换到 PCB 设计，真正

实现了从概念到产品的完整设计。是目前世界上唯一将电路仿真软件、PCB 设计软件和虚拟模型仿真软件三合一的设计平台，其处理器模型支持 8051、HC11、PIC10/12/16/18/24/30/DsPIC33、AVR、ARM、8086 和 MSP430 等，2010 年即将增加 Cortex 和 DSP 系列处理器，并持续增加其他系列处理器模型。在编译方面，它也支持 IAR、Keil 和 MPLAB 等多种编译。

最新版本：Proteus 7.10

Protues 的特点

支持当前的主流单片机，如 51 系列、AVR 系列、PIC12 系列、PIC16 系列、PIC18 系列、Z80 系列、HC11 系列、68000 系列等。

1. 提供软件调试功能。
2. 提供丰富的外围接口器件及其仿真。RAM，ROM，键盘，马达，LED，LCD，AD/DA，部分 SPI 器件，部分 IIC 器件。这样很接近实际。在训练学生时，可以选择不同的方案，这样更利于培养学生。
3. 提供丰富的虚拟仪器，利用虚拟仪器在仿真过程中可以测量外围电路的特性，培养学生实际硬件的调试能力。
4. 具有强大的原理图绘制功

7.4 Protel99 SE/Altium Designer

我们在学习制作一些简单的低频电路板时往往使用万能板进行手工焊接，而实际的一些生产应用中需要制作成印刷电路板（PCB），此时 Protel 是我们不可缺少的工具，也是作为硬件工程师必须熟练掌握的软件。

PROTEL 是 Altium 公司在 80 年代末推出的 EDA 软件，在电子行业的 CAD 软件中，它当之无愧地排在众多 EDA 软件的前面，是电子设计者的首选软件，它较早就在国内开始使用，在国内的普及率也最高，有些高校的电子专业还专门开设了课程来学习它，几乎所有的电子公司都要用到它，许多大公司在招聘电子设计人才时在其条件栏上常会写着要求会使用 PROTEL。

Prote199 SE

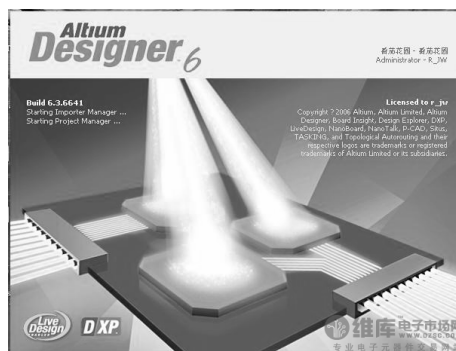
早期的 PROTEL 主要作为印制板自动布线工具使用，运行在 DOS 环境，对硬件的要求很低，在无硬盘 286 机的 1M 内存下就能运行，



但它的功能也较少，只有电路原理图绘制与印制板设计功能，其印制板自动布线的布通率也低，而现今的 PROTEL 已发展到 DXP 2004，是个庞大的 EDA 软件，完全安装有 200 多 M，它工作在 WINDOWS95 环境下，是个完整的板级全方位电子设计系统，它包含了电路原理图绘制、模拟电路与数字电路混合信号仿真、多层印制电路板设计（包含印制电路板自动布线）、可编程逻辑器件设计、图表生成、电子表格生成、支持宏操作等功能，并具有 Client/Server（客户/服务器）体系结构，同时还兼容一些其它设计软件的文件格式，如 ORCAD，PSPICE，EXCEL 等，其多层印制线路板的自动布线可实现高密度 PCB 的 100%布通率。在国内 PROTEL 软件较易买到，有关 PROTEL 软件和使用说明的书也有很多，这为它的普及提供了基础。想更多地了解 PROTEL 的软件功能或者下载 PROTEL99 的试用版，可以在 INTERNET 上。

Altium Designer

2005 年年底，Protel 软件的原厂商 Altium 公司推出了 Protel 系列的最新高端版本 Altium Designer 6.0。Altium Designer 6.0，它是完全一体化电子产品开发系统的一个新版本，也是业界第一款也是唯一一种完整的板级设计解决方案。Altium Designer 是业界首例将设计流程、集成化 PCB 设计、可编程器件（如 FPGA）设计和基于处理器设计的嵌入式软件开发功能整合在一起的产品，一种同时进行 PCB 和 FPGA 设计以及嵌入式设计的解决方案，具有将设计方案从概念转变为最终成品所需的全部功能。



Altium Designer 是业界首例将设计流程、集成化 PCB 设计、可编程器件（如 FPGA）设计和基于处理器设计的嵌入式软件开发功能整合在一起的产品，一种同时进行 PCB 和 FPGA 设计以及嵌入式设计的解决方案，具有将设计方案从概念转变为最终成品所需的全部功能。

这款最新高端版本 Altium Designer 6.除了全面继承包括 99SE，Protel2004 在内的先一系列版本的功能和优点以外，还增加了许多改进和很多高端功能。Altium Designer 6.0 拓宽了板级设计的传统界限，全面集成了 FPGA 设计功能和 SOPC 设计实现功能，从而允许工程师能将系统设计中的 FPGA 与 PCB 设计以及嵌入式设计集成在一起。

7.5 其他软件

前几节介绍的是一些必须熟练掌握的软件，除此之外，在实际开发中或者将来工作中还有一些软件经常用到也非常实用，在此粗略列举部分，感兴趣的读者可结合自身开发经验查阅相关资料。

- IAR (IAR Embedded Workbench IDE): 用于 MSP430、ARM、AVR 等系列单片机的开发;
- MATLAB: 主要用于图像处理、数学建模, 具有强大的仿真和建模功能;
- VB/VC6.0: 用于编写上位机程序, VB 容易上手, 实际做项目时经常会需要编写应用程序, 建议有精力的读者去学习 VB 或者 VC6.0 的 MFC 类库。
- QT: Nokia 旗下的一款跨平台 C++ 图形用户界面应用程序开发框架, 支持的嵌入式 Linux 平台, Windows CE;
- LabVIEW: 虚拟仪器原理及仿真, LabVIEW 使用的是图形化编辑语言 G 编写程序, 产生的程序是框图的形式, 可产生独立运行的可执行文件;
- QuartusII: Altera 公司的一款 EDA 开发软件, 专业课 EDA 中会学习;
- ISE: Xilinx 公司的一款 EDA 开发软件, Xilinx FPGA 的开发环境;
- CCS: TI 公司 DSP 开发环境;

第八章 优秀作品选集

本章内容主要介绍往年电子设计的部分优秀论文，以供大家参考。前两篇是模拟和数字电路类，没有接触过单片机和 C 语言的同学通过自学模电和数电就能完全看懂，继而就可以发挥自己的创造力，设计出更为优秀的作品。后面三篇是利用了单片机作为主控芯片，要求读者学习过单片机和 C 语言的基础知识，这部分同学们可以在学校安排的课程进度之前提前学习。

第一篇 光控自动窗帘

摘要：光控自动窗帘是利用自然光线自动控制窗帘开闭的家用自动控制装置。当夜晚来临时，变暗的光线通过控制电路将窗帘闭上；天亮后，光线又通过控制电路将窗帘拉开。控制电路中对于电动机的停止是通过单稳态电路的延时控制部分来实现的，由于 IC1 与 IC2 组成的电路是单稳态电路，它的暂态延时时间是由 RP1、RP2 与 C1、C4 来决定的。SB、SB2 分别组成手动开、闭控制电路。

关键词：光控；自动；窗帘

1 作品功能及总体方案

1.1 功能描述

简单来说，该装置是在窗户玻璃和窗帘之间安装的一种感光器。当光线达到一定程度时，便能使窗帘自动拉开或闭合，保证室内始终处于适宜的光亮环境，为我们的生活提供方便，为实现智能家居做出贡献。

光控自动窗帘要实现三个功能：①天亮后，光线又通过控制电路将窗帘拉开；②天黑后，变暗的光线通过控制电路将窗帘闭上；③独立于光控回路的手动电路，此时电路只由手动控制，不受光线因素的影响。

1.2 系统组成及工作过程

(1)555 时基电路

(本例采用 NE555 时基电路，故只说明 NE555 的相关内容)

555 时基电路是一种集模拟与数字电路特点于一身的集成电路，能以简单的方式和数字

集成电路直接连接，中间不需转换。

555 时基电路可以组成 R—S 触发器（即双稳态触发器），用作控制电路的开关；还可以组成单稳态触发器，用来做定时、延时或脉冲整形；也可以组成施密特触发器，在多种控制电路中得到应用。

555 时基电路有着独特而优异的性能，其独特的优点表现在组成电路简单，工作电压范围宽（2V-18V），工作性能稳定可靠，能以两种方式输出一定的控制功率，可以直接驱动继电器或扬声器等功率元件，价格便宜。

附表 1 555 的真值表

注：“*”表示为任意电平。

该表列出的引出端真值表，显然与一般数字电路不同，其触发端和阈值端输入不一定是逻辑电平，可以是模拟电平，因此该电路兼有模拟与数字的特色。

Ne555 的各个引脚功能如下：

1 脚：外接电源负端 VSS 或接地，一般情况下接地。

8 脚：外接电源 VCC，双极型时基电路 VCC 的范围是 4.5 ~ 16V，CMOS 型时基电路 VCC 的范围为 3 ~ 18V。一般用 5V。

3 脚：输出端 V_o

2 脚： \overline{TL} 低触发端

6 脚：TH 高触发端

4 脚： $\overline{R_D}$ 是直接清零端。当 $\overline{R_D}$ 端接低电平，则时基电路不工作，此时不论 \overline{TL} 、TH 处于何电平，时基电路输出为“0”，该端不用时应接高电平。

5 脚：VC 为控制电压端。若此端外接电压，则可改变内部两个比较器的基准电压，当该端不用时，应将该端串入一只 0.01 μ F 电容接地，以防引入干扰。

7 脚：放电端。该端与放电管集电极相连，用做定时器时电容的放电。

555 时基电路的内部机构如图 1-1。

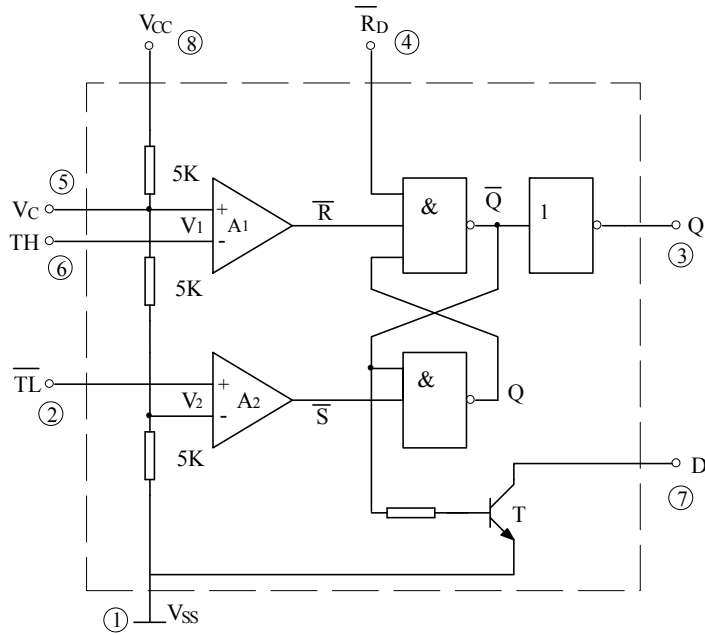


图 1-1

在使用 555 时基电路时还需注意以下几点：

①由于 555 内部的比较器能影响 20ns 的脉冲，为了有效地防止外部干扰所造成的误动作，所以 555 时基电路的⑤对地应加接 0.01uf 的去耦电容并应安装在引脚最近处。

②触发脉冲的作用可使②脚背驱动至充分接地，使下比较器的存储时间长达 10us 这种作用限制了单稳态脉冲的宽度最小不小于 10us。

③复位端到输出的延迟时间典型值为 0.47us，所以典型的复位脉冲宽度必须大于等于 0.3us。

④555 时基电路的②脚是集电极开始输出端，这种集电极开路的输出方式有着特殊的功能，它使门电路的并接成为可能。555 时基电路与 TTL(双极型的)电路相容，使用在高速时必须注意：由于时基电路内部的延迟作用，②脚的电流转换比③的电压转换慢，也就是说，它们的翻转有微小的不同步，时间差在 30ns 以内。

放电电流：555 时基电路在用作单稳态电路或多谐振荡器时，由于外接电容要随时进行充放电，当电路在进入电容放电过程时，由电容通过 555 时基电路内的放电管 Td 进行放电，放电电流就是电容放电时流过放电管的电流。显然，放电电流的大小与放电管 Td 的额定电流有关，放电电流用符号 I_{dis} 表示。

对于双极型电路（例如 NE555）它的定时精度一般为 1%。

(2)单稳态电路的基本知识

所谓“单稳态”电路，就是该电路只有一个稳定状态，另一个状态为不稳定状态或称为暂稳定状态。这种电路在平时处于稳定状态，当需要时用一个触发信号去触发它，使它翻转进入暂稳态，经过一段时间后电路又自动恢复到稳态。

单稳态电路有以下特点：一是电路处于稳定态时，若没有外加的触发信号，则电路一直处于稳定状态而不会自动改变其状态；二是暂稳态的暂稳时间可以根据需要预先设定；三是到达预定的暂态时间后，电路可不加任何信号而自动恢复到稳态。

单稳态电路的组成特点是将一个具有时间常数的 RC 电路接至 555 时基电路的②、⑥(或②⑦)脚，RC 电路的一端接工作电源，另一端接地；将触发信号加至②、⑥或②脚。

由 555 时基电路组成的单稳态电路的工作过程是这样的：先取该电路两种输出状态中的一种状态（例如输出低电平状态）作为电路的稳态，然后用输入控制脉冲的方法去启动这个电路，使输出端从原来的（低电平）状态转变为另一种（高电平）状态，即进入暂稳态。在电路翻转的同时，电源通过电阻 R_t 向定时电容 C_t 充电，当 C_t 充电电压上升到 555 时基电路的阈值电压 $2V_{cc}/3$ 时，电路翻转， $2V_{cc}/3$ 由高电平恢复到低电平的状态。

单稳态电路按启动方式分两种结构：人工启动式单稳态电路，脉冲启动式单稳态电路。

图 1-2 为人工启动式单稳态电路的结构图。

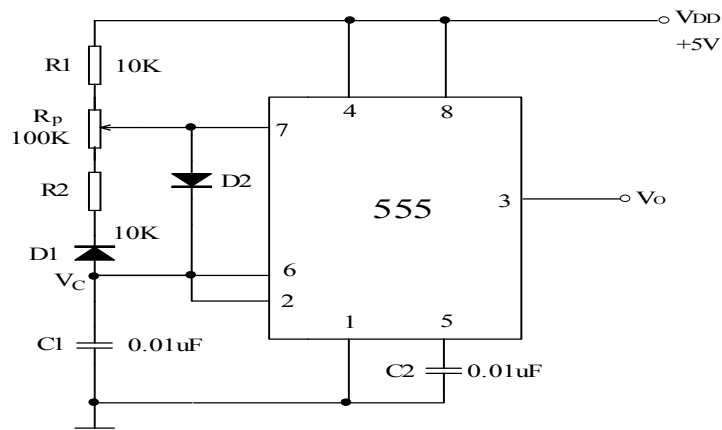


图 1-2

图 1-3 为脉冲启动式单态电路的结构图。

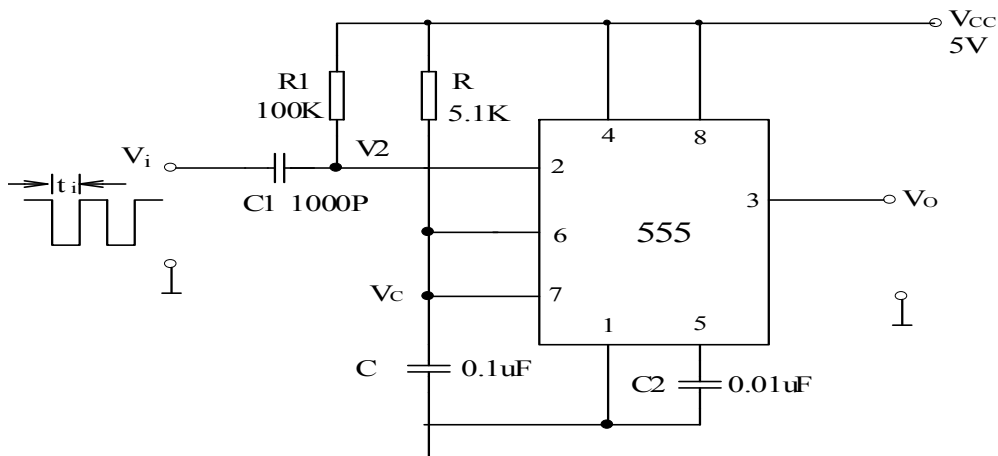


图 1-3

本例采用的是脉冲启动式单稳态电路。

脉冲启动式单稳态电路与人工启动式单稳态电路的不同之处是：将原电路（人工启动式单稳态电路）中的②、⑥脚接在一起接入 RC 电路中，改为将⑥、⑦脚（阈值端 TH 和放电端 DIS）连接在一起接入 RC 电路，另外将②脚单独用作启动端。电路的启动过程是将一个负向的脉冲加至启动端②脚，因此称之为脉冲启动式单稳态电路。

脉冲启动式单稳态电路两种工作状态的转换过程分析：①稳态：在这种单稳态电路中，它的启动端②脚通常总是在接通电源时就被接上高电平（而不是由 0 上升到 $1V_{CC}/3$ 以上）。在这种情况下，555 时基电路内的输出端③脚一定是输出低电平的。另一方面，由于在这时 555 时基电路内的放电管处于导通状态。电容 C_t 不可能进入充电状态，因此电路的状态一直保持不变，即处于稳态。在电路的稳态下，②脚加高电平，（本例是通过三极管实现的）

⑥、⑦脚则由于放电管的导通， C_t 不能充电而保持为低电平，输出端③脚输出低电平。

②暂稳态：当对②脚输入一个负向的脉冲，而且脉冲的幅值低于 $1V_{CC}/3$ 时，电路翻转，输出端③脚由低电平变为高电平。与此同时，555 时基电路内的放电管截止，电源通过 R_t 向 C_t 充电，电路进入暂稳态。

电路进入暂稳态后，输出端③脚输出高电平，⑥、⑦脚变为低电平，电容 C_t 进入充电状态。随着时间的推移，电容 C_t 上的电压逐渐上升，当电压上升到 $2V_{CC}/3$ 以上时，电路翻转，输出端③脚由高电平变为低电平，电路恢复到稳态。

暂稳态时间的长短与 R_t 和 C_t 的数值成正比，即： $t = 1.1R_t C_t$

在脉冲启动式单稳态电路，当电路被触发进入暂稳态后，如果对触发端②再加入触发脉冲，电路不会发生任何变化。因此在暂稳态时间内的触发脉冲是对电路无效的脉冲。如图 1-4 所示。

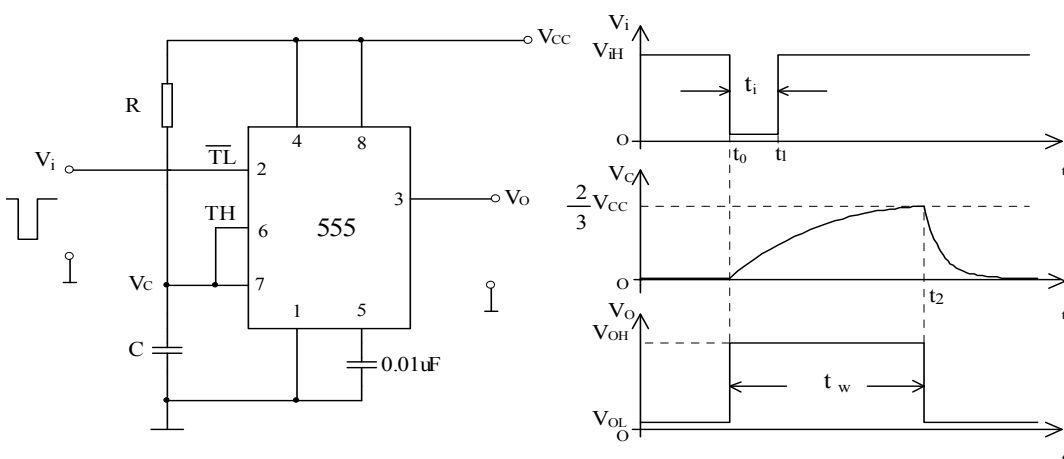


图 1-4

2 硬件设计

2.1 硬件电路总体设计

光控自动窗帘是利用自然光线自动控制窗帘开闭的家用自动控制装置。当夜晚来临时变暗的光线通过控制电路将窗帘上；天亮后，光线又通过控制电路将窗帘拉开。

控制电路中对于电动机的停止是通过单稳态电路的延时控制部分来实现的，由于 IC1 与 IC2 组成的电路是单稳态电路，它的暂态延时时间是由 RP1, RP2 与 C1, C4 来决定的。

SB1, SB2 分别组成手动开，闭控制电路。

3 光控自动窗帘电路工作原理分析：

该光控自动窗帘电路由光控电路，单稳态触发器电路和电源电路组成。电路中，光控电路由光敏晶体管 V1，晶体管 V2 和有关外围元件组成，时基集成电路 IC1 和 IC2 均接成单稳态电路，控制电动机 M 的正转与反转；按原电路图，电源电路由降压电容器 C6，整流二极管 VD1 和 VD2，稳压二极管 VS 和滤波电容器 C5 组成，交流 220V 电压经 C6 降压，VD1 和 VD2 整流，VS 稳压和 C5 滤波后，产生+12 电压，供给 IC1、IC2 和光控电路。但此处，电源电路我们采用了 6V 直流电源供电，省去了交变直的电路。

清晨，光敏晶体管 V1 在自然光线的照射下导通，使 IC1 的 2 脚和晶体管 V2 的基极为低电平，V2 截止 IC1 内部的单稳态触发器动作（变为暂态），其 3 脚输出高电平，IC2 的 2 脚为高电平，3 脚为低电平，使电动机 M 正转，将窗帘拉开。当窗帘拉开到位后，IC1 又由暂态变为稳态，其 3 脚恢复低电平，使电动机 M 停转。

傍晚光线暗淡时，V1 截止，IC1 因 2 脚为高电平而不能被触发，其 3 脚为低电平，V2 因基极变为高电平而导通，其集电极电压下降，使 IC2 的 2 脚产生低电平触发电压，IC2 内部的单稳态触发器翻转，由稳态变为暂态，3 脚输出高电平，使电动机 M 反转，窗帘开始闭合。窗帘闭合到位后，单稳态触发器恢复为稳态，IC2 的 3 脚变为低电平，电动机 M 停转。

S1、S2 为手动开关。采用手动控制时，应将 S3 关断，使光控电路失效。按动 S1 时，IC1 触发翻转，电动机 M 正转，将窗帘拉开；按动 S2 时，IC2 触发翻转，电动机 M 反转，将窗帘闭合。调节电位器 RP1 和 RP2，可以改变电动机的运转时间。

4 用到的元器件：

R1、R2 均选用 1/4W 碳膜电阻器；

R3 选用 1/2W 金属膜电阻器。

RP1、RP2 均选用小型膜式线性电位器。

C1、C4 均选用耐压值为 25V 的铝电解电容器；

C2、C3 选用涤纶电容器或独石电容器。

1 选用 3DU 系列的光敏晶体管；V2 选用 S9014、58050 等型号的硅 NPN 型晶体管。

IC1 和 IC2 选用 NE555 时基集成电路。

S1, S2 均选用微型按钮开关；S3 选用自锁动断型按钮。

电机选用的是额定电压为 6v 的小型直流电动机。

5 实验及结果

5.1 焊接

实物图如图 5-1 所示。

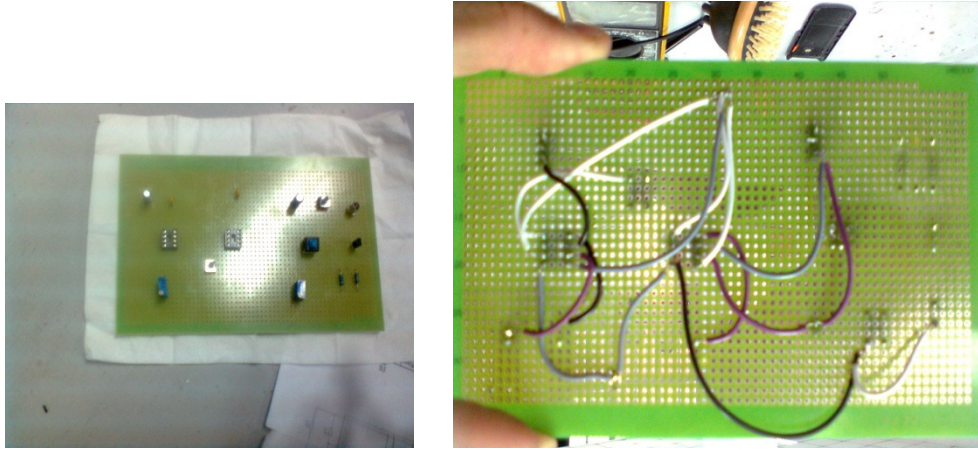


图 5-1

5.2 电路的调试与运行

对于时间常数的确定，我们先做了仿真，用 multisim10 软件很好地实现了预计的功能，并仿真出了各参量的数值。仿真结果如图 5-2, 5-3, 5-4, 5-5, 5-6 所示

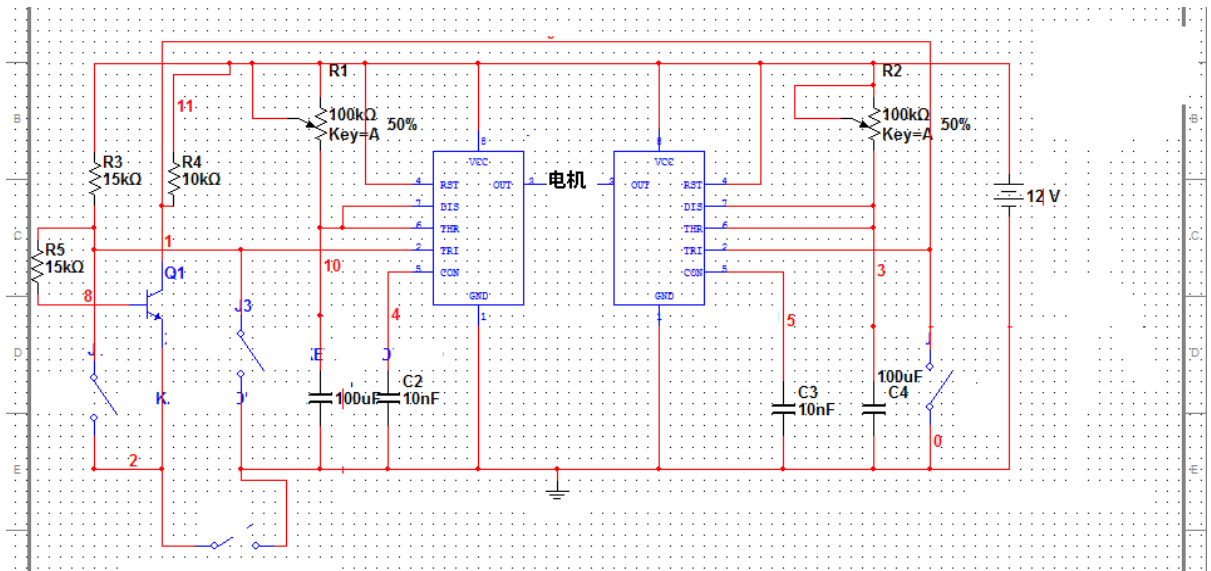


图 5-2

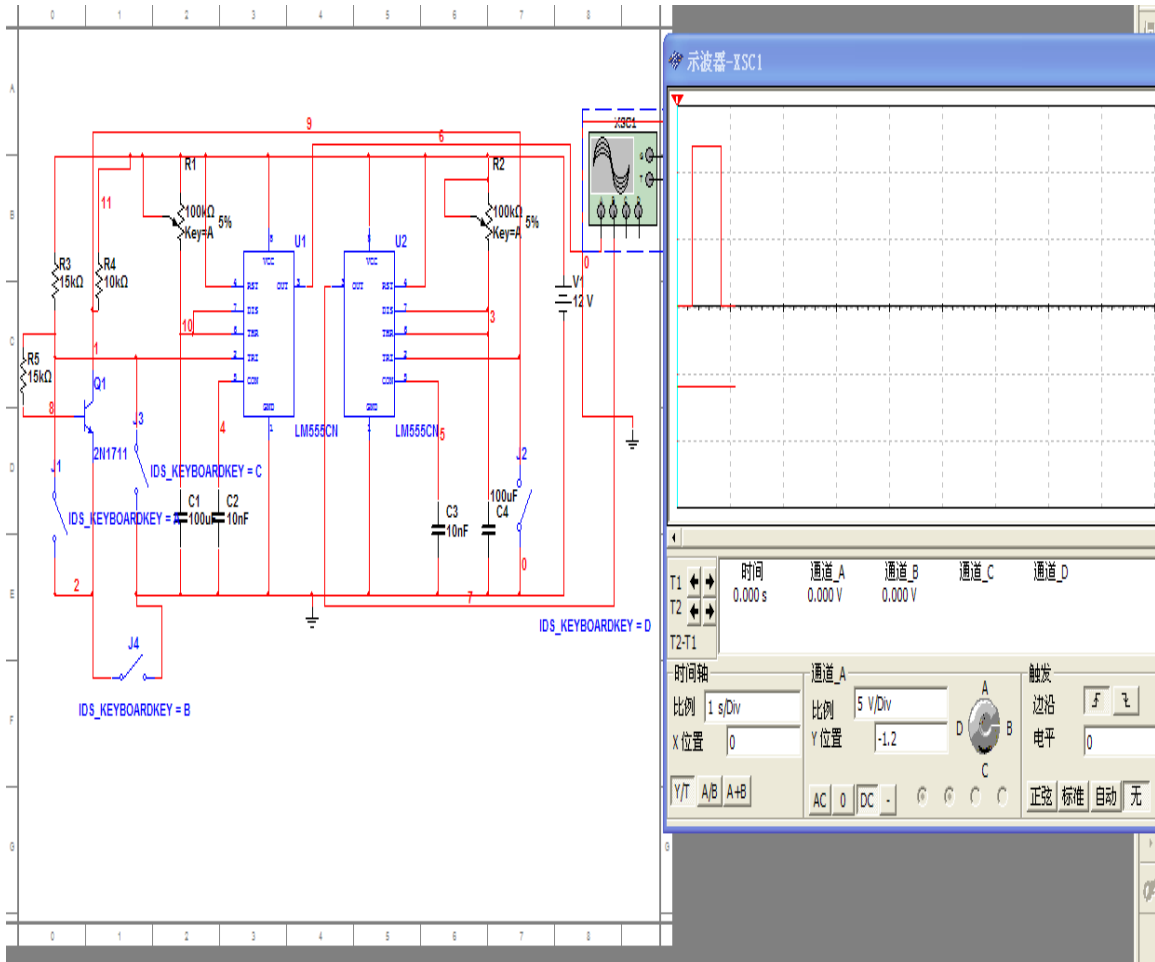


图 5-3

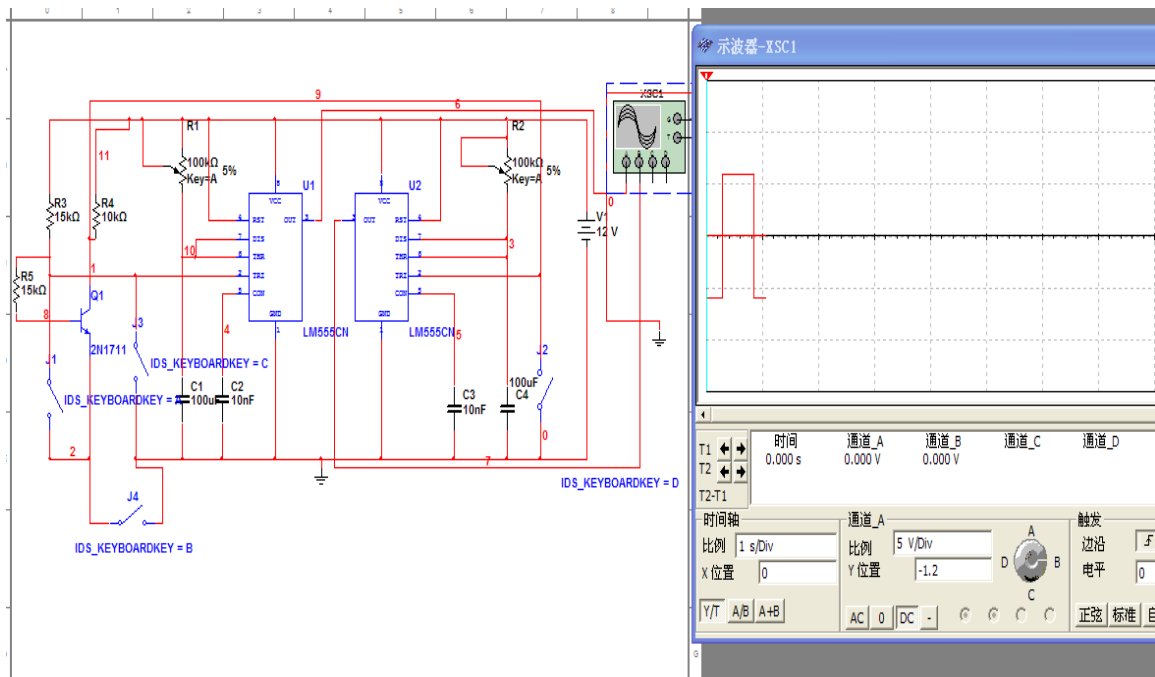


图 5-4

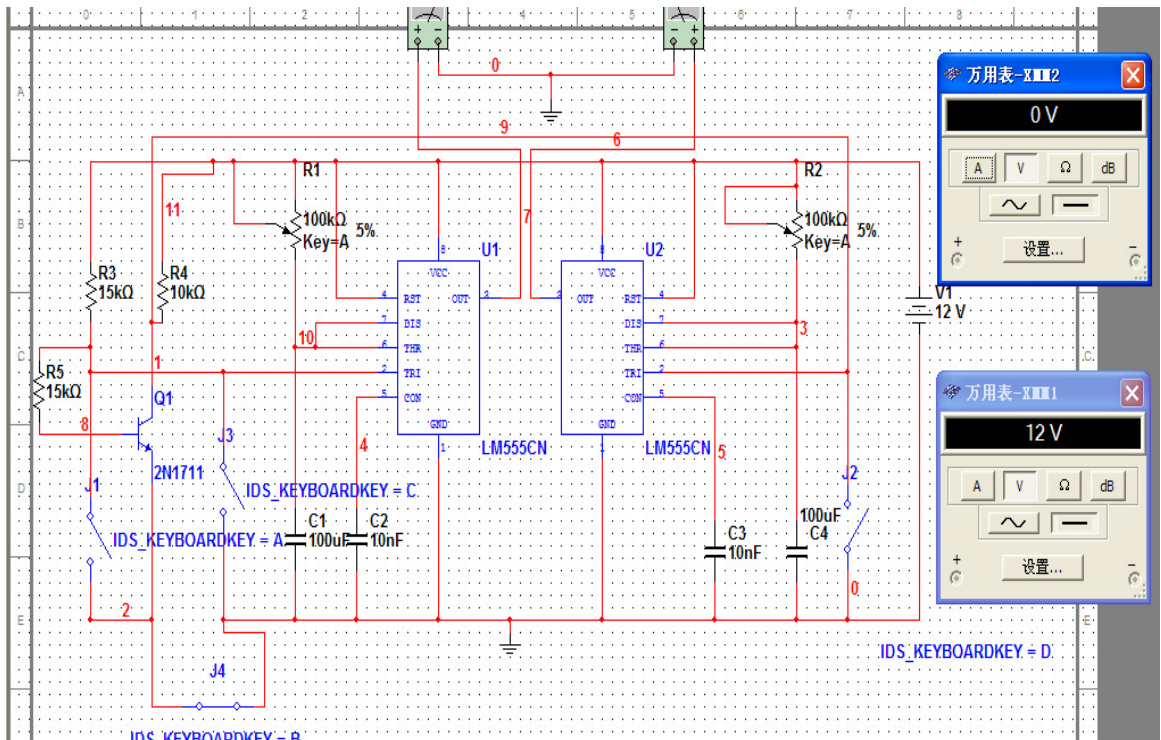


图 5-5

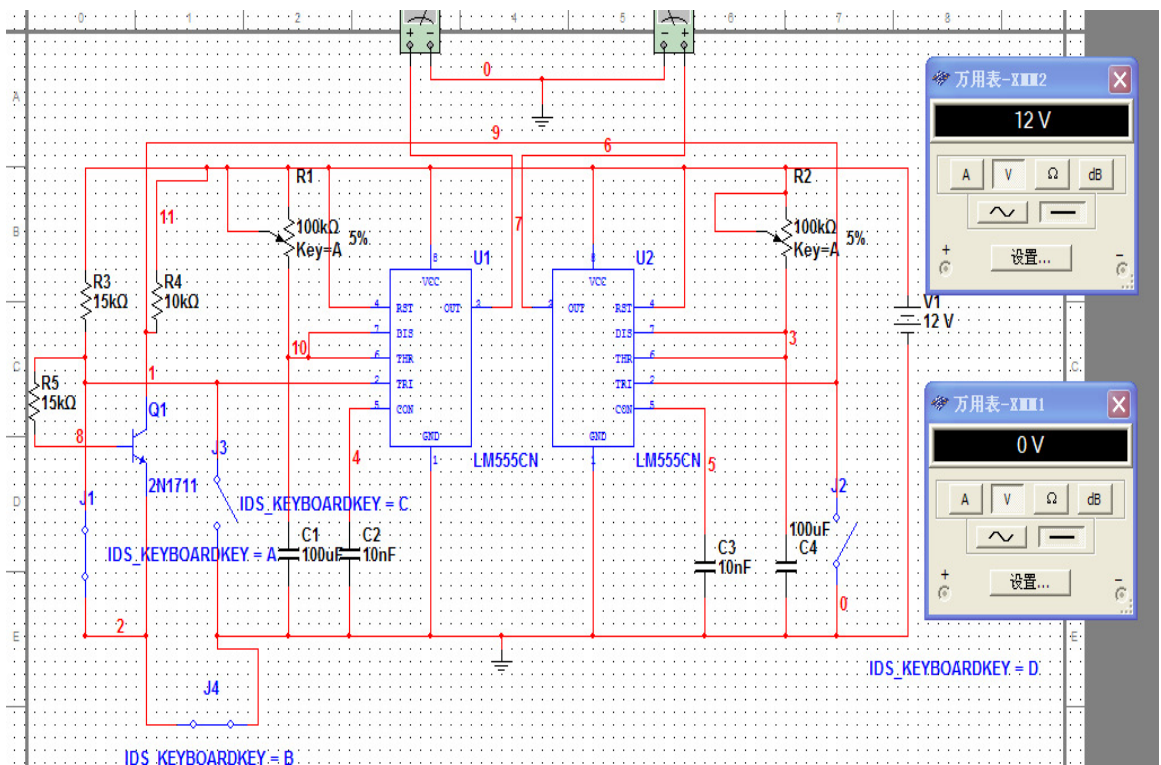


图 5-6

之后我们反复调节了电路板上 R_t 、 C_t 的值并用示波器观察了各参量的波形。（注：在调光控功能时我们用到了手电筒）最后实现了光控、手动等功能。至此，电路调试运行完成。

6 需要说明的几点

1. 电源：本例我们采用的是 6v 直流电源，由于电池提供。
2. 光敏电阻的安装：置于室外，减小室内灯光对其光控功能的影响；室外安装时，设计巧妙装置：接收太阳光，阻挡例如车灯光线等人工光，从而减少人为因素对光控功能的影响。
3. 单稳态电路定时精度问题：由通常的充电定时改为放电定时，能克服大容量电解电容的漏电影响定时精度的问题。⑤脚接入一只 1K 的电阻可延长定时时间。
4. 此作品的缺点：阴天时，光线强度不够，是光控功能不能很好实现；要实现上述电路的功能，必须要一直供电，这样就会增大电能的损耗。
5. 前景：
 - ①手动控制设计成遥控装置或配合定时器应运于智能家居。
 - ②直接利用太阳能为电路供电，是电路达到零损耗。
 - ③配合一些隔音材，将其做成兼具隔音功能的光控自动窗帘。

第二篇 LED 照明灯的智能控制

摘要：本设计的主要目的是改变传统的人为控制 LED 照明灯，而采用智能控制，当光线变强或变弱时，LED 可自动关闭或打开、也可根据个人需要调节光线强度，当电路中出现故障时，如过压、过流、开路，电路都可以自动保护并恢复。更加节能、环保，效率高达 75%以上。

关键词：大功率 LED 灯、继电器、光敏电阻、LM324 运放、三端稳压、可恢复式过流保险丝、PMOS 管

1 设计要求

LED 照明灯的智能控制不用人为干涉便可实现自动控制灯的亮灭，保护电路并可自动恢复。整体的设计要求主要体现在功能实现方面。

1.1 基本功能要求：

1. 主要性能

输入电压：24V+10%

2. 主要技术要求和指标

- (1) 负载 3 个 LED 灯，能够正常工作，输出平均电流 350+20%毫安。
- (2) 输出过流保护并自恢复。
- (3) 输出开路保护并自恢复。
- (4) 输入过压保护并自恢复。
- (5) 效率 70%以上。

1.2 扩展功能要求：

- (1) 当光线较强时关闭 LED 灯，当光线较弱时打开 LED 灯。
- (2) 环境温度过热保护，当温度高于 60 度时关闭 LED 灯，当温度低于 55 度时重新打开 LED 灯。
- (3) 输出电流可调，调整档 200+20%毫安，300+20%毫安，350+8%毫安。
- (4) 负载 5 个 LED 灯，输出（最大）平均电流 350+8%毫安。
- (5) 效率 75%以上。

2 总体设计

2.1 系统组成及工作原理

本作品主要用到硬件设施，分过压、过流、开路、光控以及电流可调等五个模块综合来实现 LED 照明灯的智能控制。

2.2 系统组成如图

系统组成如图如图 1 所示。



图1 系统组成

2.3 过压保护

过压保护电路如图2所示。

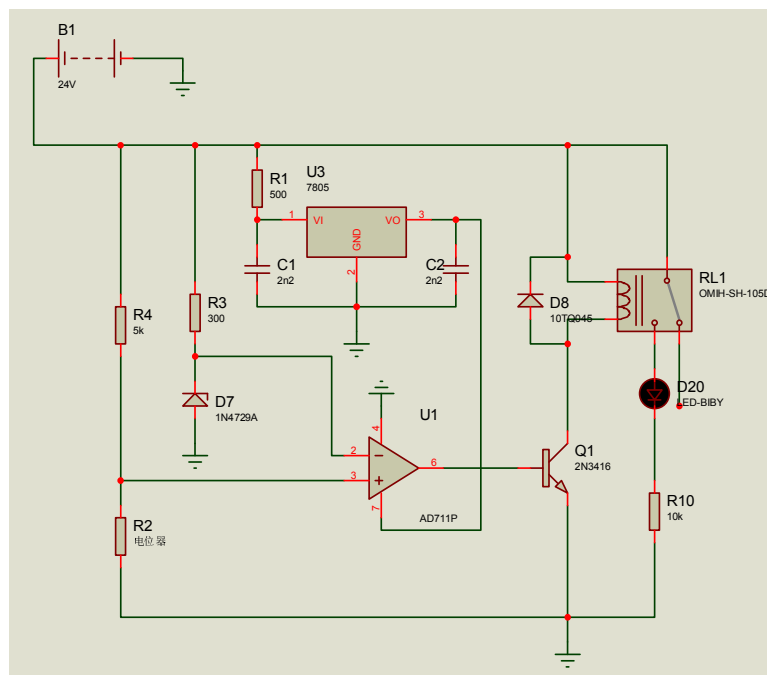


图2 过压保护电路

通过电压比较，利用三极管的导通性做开关作用，连接继电器，当电压过高时，三极管的基极为高电平，此时三极管导通，继电器工作，开关由常闭到常开，指示灯亮，提示过压。当电压恢复正常时，继电器可自动恢复。

由于此集成运放用的是 LM324，所以采用三端稳压 7805 将 24v 电源转化成正五伏，并将负端接地。

2.4 过流保护

方案一：使用自恢复式过流保险丝元件，当电流达到一定值后，再有微弱的电流增加都会引起电阻值的迅速增大，可达无穷，等同开路，当电流正常时，温度降低，阻值变小，仅有几欧，利用其阻值随电流可变原理来保护电路。

方案二：原理如下图，当电流正常时，R1、Q2、D3 支路导通，当电流过大时，R1 右端电位（三极管 b 极）变高，Q1 导通，继电器工作，开关由常闭到常开，G 端为高电平，Q2 不导通，从而保护 LED。如图 3 所示。

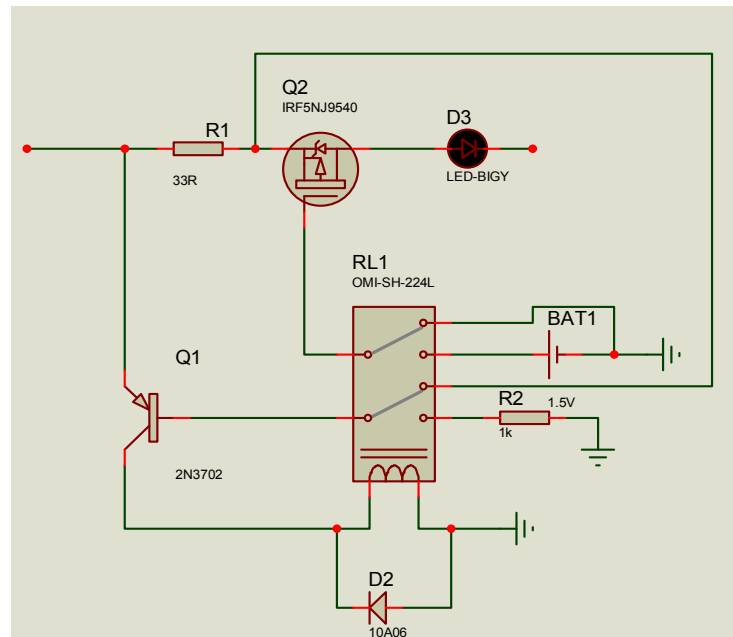


图 3 过流保护电路

两方案相比，第一个方案以更加简单、经济、实惠的方法就实现了此功能，而第二个方案在实际中较难实现，所以这里选用方案一。

2.5 开路保护

开路保护电路如图 4 所示。

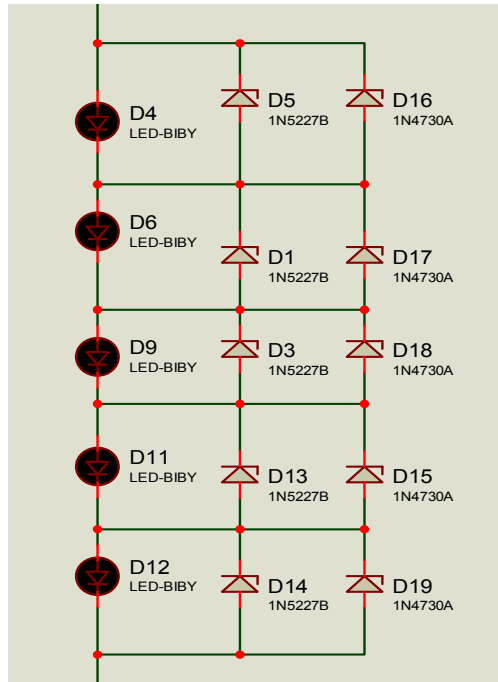


图4 开路保护电路

将五个 LED 串联，采用恒流驱动原理，以保证流过每个 LED 的电流都相同，从而使每个灯的亮度都相同，同时又将每个 LED 并联两个稳压二极管以保护 LED 灯，当电路中一个 LED 损坏或开路时稳压二极管被击穿，形成通路，使其他 LED 还能正常工作。

在此方案中，考虑到稳压二极管的耐电流值为 230mA 左右，采用同时并两个二极管以分流，防止热击穿损坏二极管。

2.6 光控：

光控电路如图 5 所示。

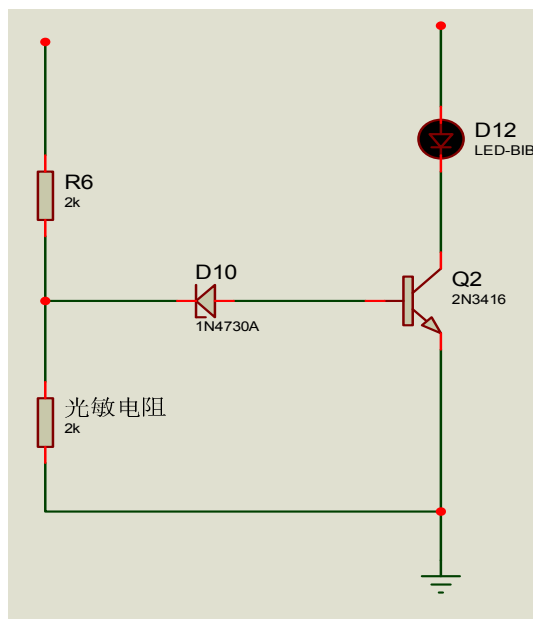


图5 光控电路

利用光敏电阻的特性：随外界光线增强，其阻值变小。

当外界光线变弱时，光敏电阻阻值变大，上端点位较高，稳压二极管被击穿，三极管导通，LED 点亮。

由于光敏电阻在光强条件下阻值基本为零，在完全黑暗的条件下可达无穷，当天黑时光敏电阻上端电位基本达到二十四伏，易损坏稳压二极管。改进方案如图 6。

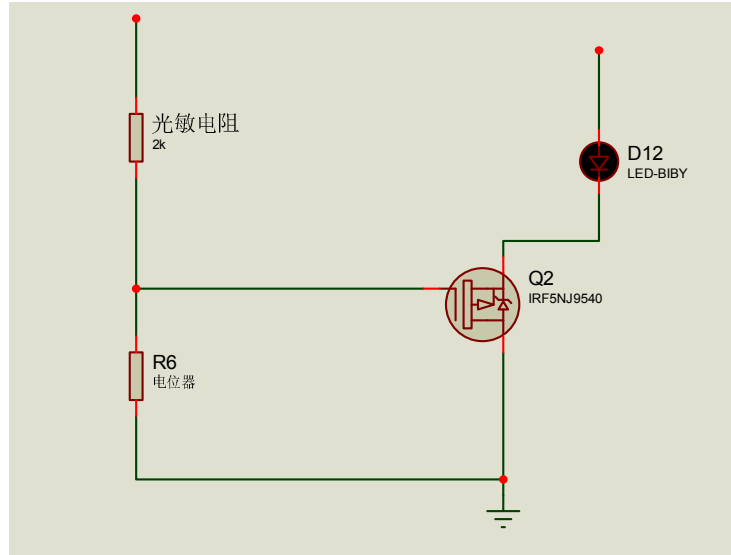


图 6 光敏电阻电路

当外界光强时，光敏电阻的阻值变小，G 端电位变高，Q2 截止，灯灭；

当外界光弱时，光敏电阻的阻值变大，G 端电位变低，Q2 导通，灯亮。

2.7 电流可调：

方案一：

电流可调电路 1 如图 7 所示。

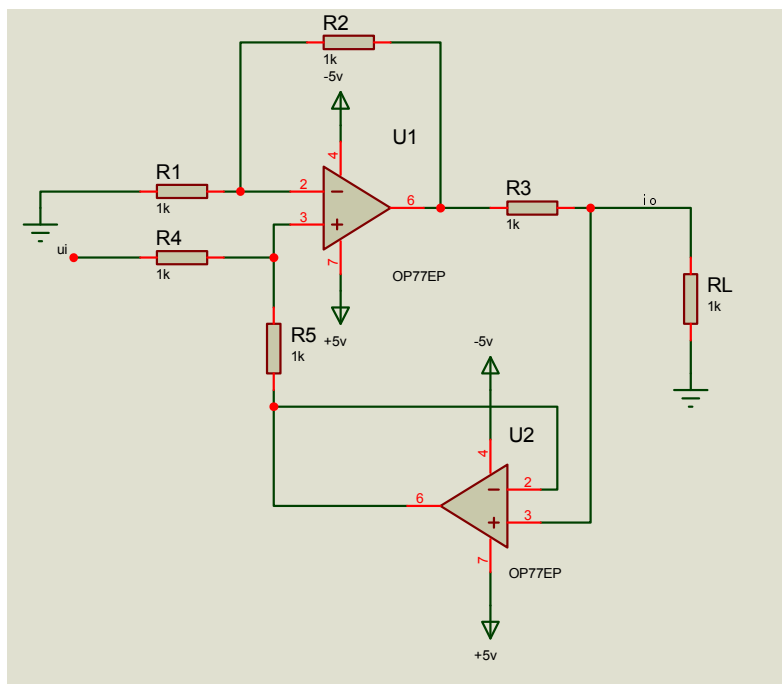


图 7 电流可调电路 1

在电路中增加 V/I 变换，可实现输出电流的稳定可调。

方案二：

电流可调电路 2 如图 8 所示。

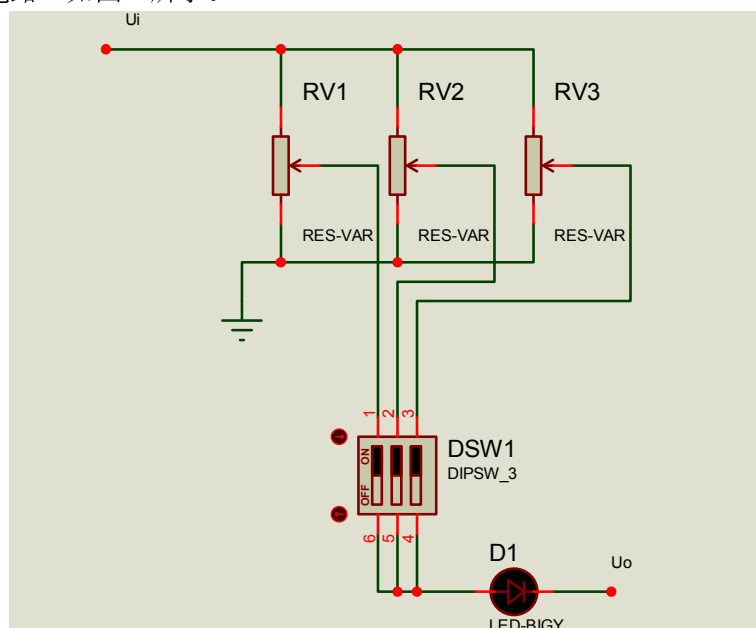


图 8 电流可调电路 2

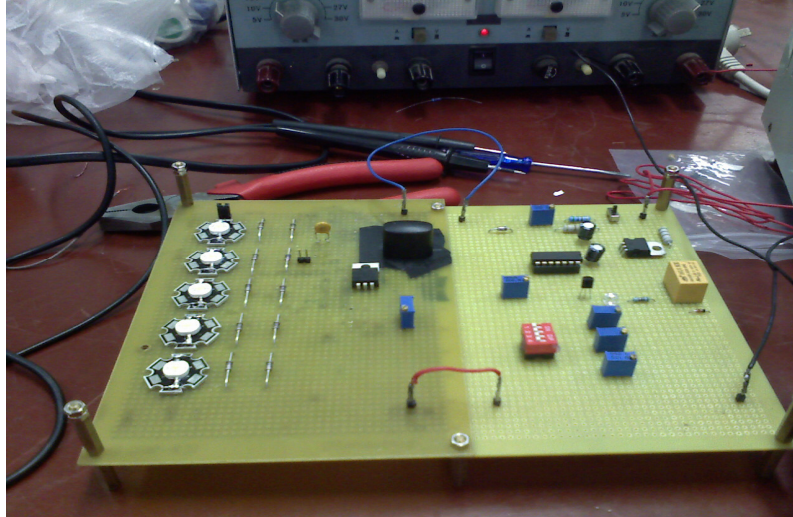
通过与灯分压式并联直接调节电流

由于方案一的 V/I 变换只适用于小功率的电路中，在实际电路中难以实现，所以选用第二个方案直接进行调节。

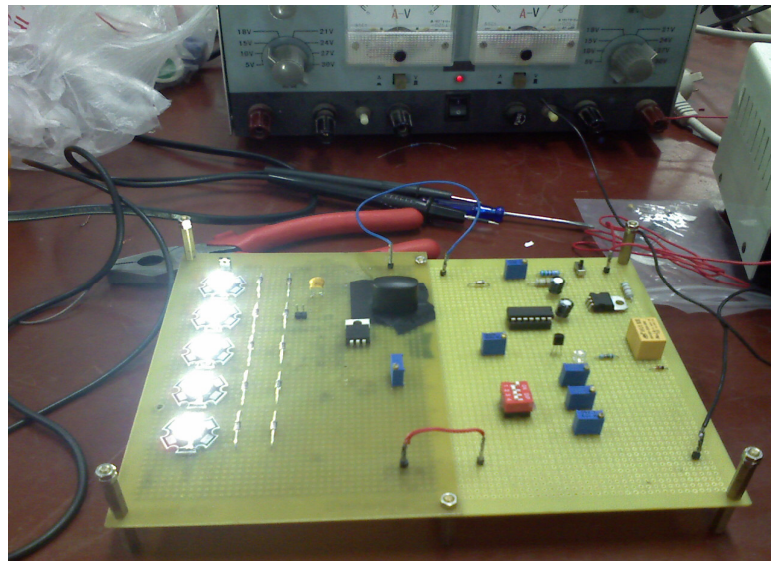
3 调试及结果

3.1 实际电路如图：

电流可调：未拨到任何挡时，LED 灯不亮：



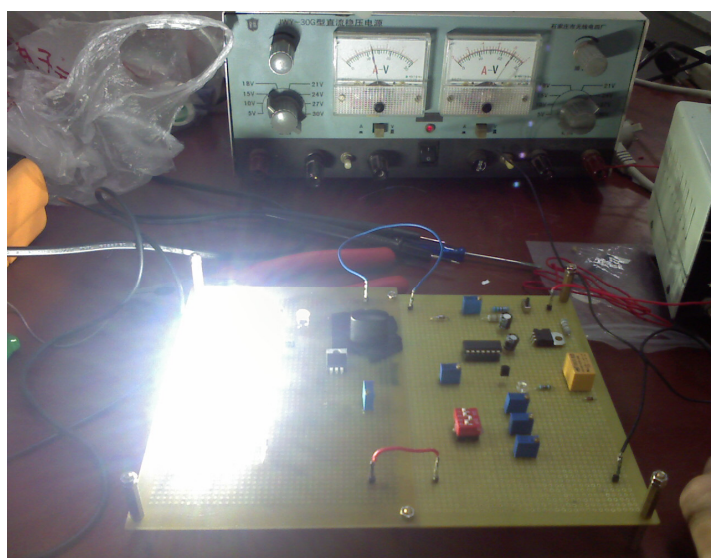
拨至 1 挡，LED 灯亮度较微弱：



拨至 2 档，灯的亮度增强：



拨至 3，灯的亮度最强：



第三篇 多功能数字钟的设计

摘要：本系统是以 51 系列单片机 89C52 作为控制器，主要由数码管显示电路，测温电路，DS1302 电路，声音传感止闹电路四部分组成。最主要的特点是利用了硬件软件化的思想，能够用软件来完成的尽量用软件完成。使电路简单化，系统稳定性大大提高。优点主要是实用性强，在完成各种功能模块的前提下，尽量降低成本。

关键词：多功能数字钟；89C52；DS1302

1. 任务

设计制作一个 24 小时制多功能数字钟。该数字钟可实现时间设定和定时闹钟控制。

2. 要求

基本要求：

具有时间（时：分：秒）显示和设置功能。

发挥部分。

总体方案如图 1 所示。

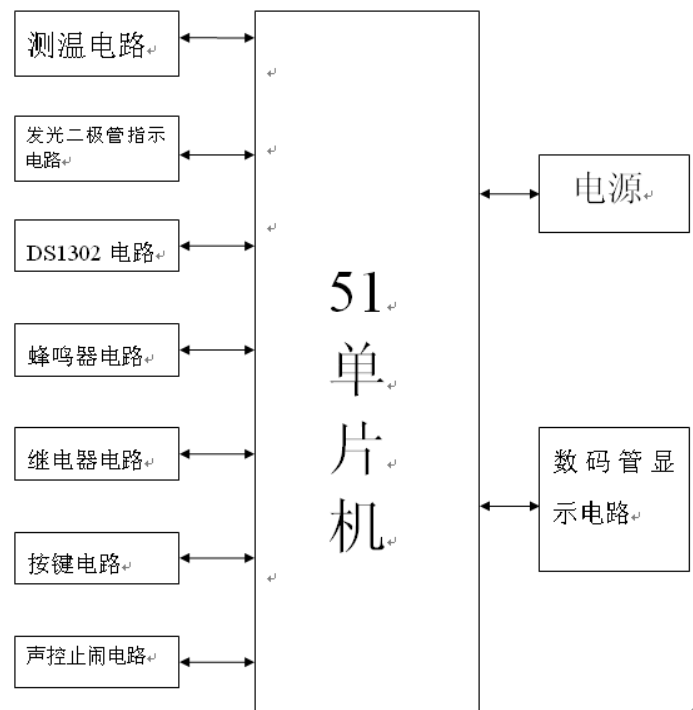


图 1 系统总体方案

系统利用 51 单片机作为控制模块，并采用了一个 5V 电源给系统供电，数码管是主要的显示模块，并用发光二极管来作为指示，单片机把声控止闹电路，按键电路，DS1302，测温电路的信息收集起来，并进行综合处理，把数据传输到数码管显示出来，实现了时间、日期、温度的显示。并对蜂鸣器，继电器，发光二极管进行控制，硬件资源利用十分充足，几乎利用了 51 单片机的所有输入/输出接口。

3. 硬件电路分析与理论分析

3.1 数码显示电路

本显示电路应用了 8 个数码管，共占用 16 个单片机引脚。P0 和 P2 接数码显示电路，P2 接口用来控制 8 个数码管。当要显示其中一个数码管时，相应的 P2 接口为低电平，其他 P2 接口为高电平。P0 用来输出字段码，本电路用的是动态显示方式，采用软件译码，大大减少了硬件电路的复杂性，给我们焊接电路带来方便。如图 2 所示。

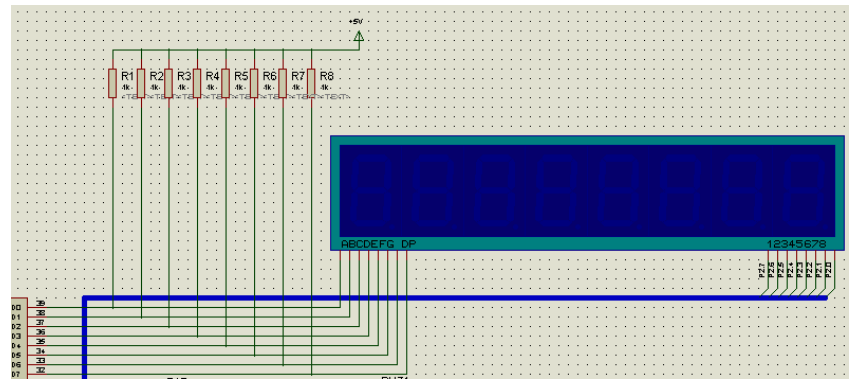


图 2 数码显示电路

3.2 二极管指示电路

本系统采用 4 个发光二极管 (H1, H2, H3, H4) 对系统的工作状态进行指示，并组成闹钟和作息时间的控制部分，电路把二极管和 1k 的电阻串联连接，接上 +5V 的电压，然后分别接在 P1.0, P1.2, P1.3, P1.4 上，单片机对其进行控制，当输出高电平时，二极管灭；输出低电平时二极管亮。如图 3 所示。

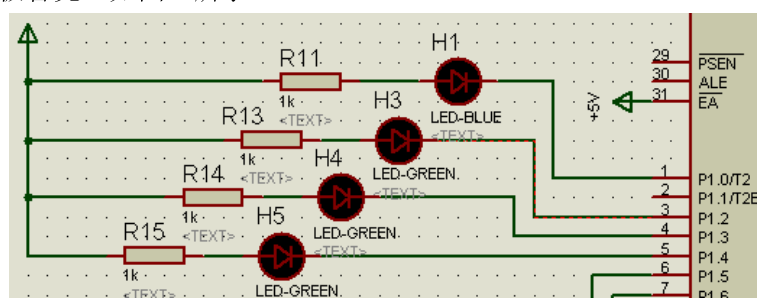


图 3 二极管指示电路

3.3 DS1302 电路

DS1302 为系统提供实时时钟，DS1302 的 SCLK, I/O, \overline{RST} , 分别接在单片机的 P1.5, P1.6, P1.7 口。x1, x2 接 32.768kHz 的晶振，GND 接地，Vcc1 接 3V 备用电源，当系统

断电时，备用电源给 DS1302 供电，使 DS1302 继续运行，因此在没有主电源的时候能保存时间信息以及相关数据。DS1302 由 Vcc1 和 Vcc2 中两者较大的电源供电。P1.7 接 DS1302 的复位引脚，低电平有效。I/O 口为数据输入/输出引脚，具有三态功能，通过此引脚单片机和 DS1302 之间进行信息交流，SCLK 为串行时钟输入引脚。如图 4 所示。

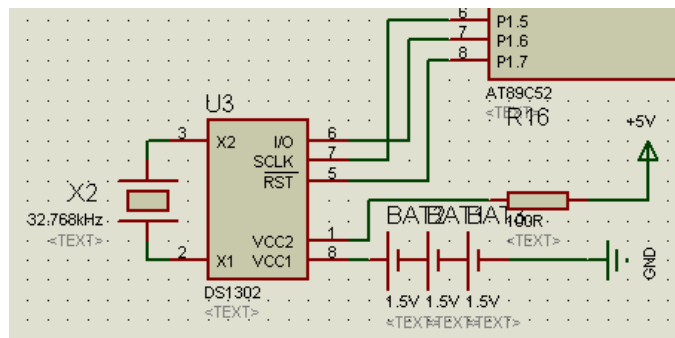


图 4 实时时钟电路

3.4 蜂鸣器电路

蜂鸣器电路如图所示。使用三极管 8550 驱动蜂鸣器。并用单片机的 P3.0 口对其进行控制，当输出高电平时，三极管 8550 截止，蜂鸣器停止鸣叫，当输出低电平时，三极管 8550 导通，蜂鸣器鸣叫。如图 5 所示。

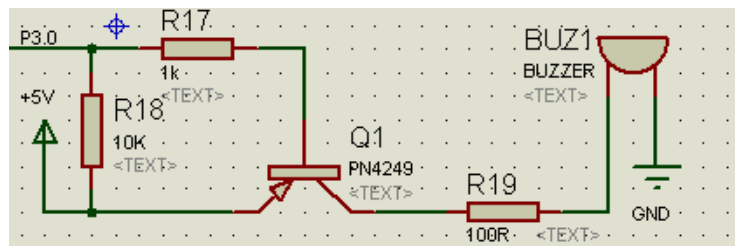


图 5 蜂鸣器电路

3.5 继电器电路

此继电器电路用三极管 8550 进行驱动，并用一个发光二极管来指示继电器的工作情况，电路如图所示，用单片机的 P3.1 口对继电器进行控制。当输出高电平时，三极管 8550 截止，因此继电器处于断开状态，同时二极管灭。当输出低电平时，三极管 8550 导通，继电器闭合，同时二极管亮。在继电器由闭合到断开的瞬间，由于线圈中的电流不能突变，将在线圈中产生下正上负的感应电压，使晶体管集电极承受很高的电压，有可能损坏驱动管 8550。为此在继电器线圈两端并联一个续流二极管 1N4148，使线圈两端的感应电压被箝位在 0.7V 左右。正常工作时，线圈上的电压上正下负，二极管 1N4148 截止，对电路没有影响。如图 6 所示。

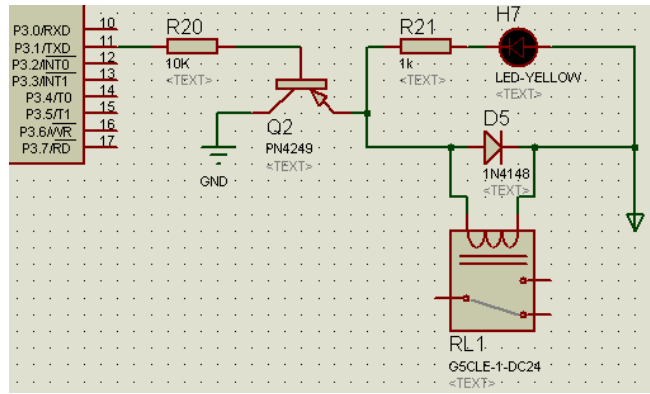


图6 继电器电路

3.6 键盘电路

本系统采用独立式按键方式，电路简单明了。使用五个按键，分别接在单片机的P3.2~P3.6接口，当其中一个按键被按下时，则相应的接口输入低电平，当系统扫描到时，系统就运行相应的程序。如图7所示。

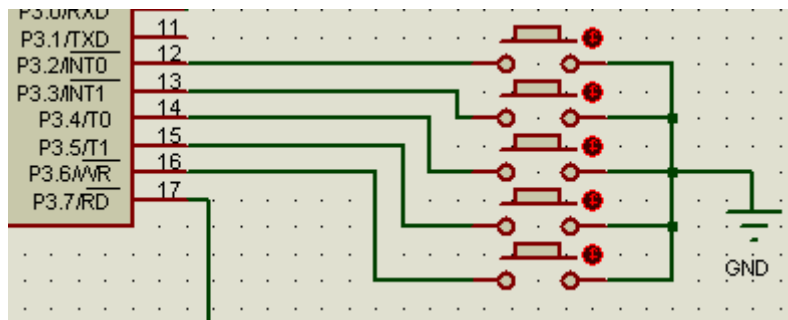


图7 键盘电路

3.7 测温电路

本次设计用DS18B20作为温度传感器来测量温度，电路十分简明，只需要一个4.7K电阻和一个DS18B20就可以组成电路，只需占用一个I/O接口。给我们焊接电路带来方便。DS18B20测量温度比较准确，精度一般可以达到0.5度，可信度高，完全满足我们生活中的需要。如图8所示。

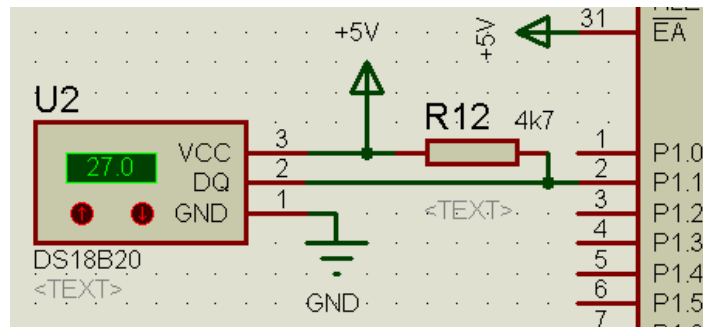


图8 测温电路

3.8 声控止闹电路

电路如图 9 所示，电路接于单片机的 P3.7 口，MIC 用于收集声音信号。用电容 104 把直流部分过滤掉，交流部分经过电容 104，导致三极管 9014 的基极电流 I 减少，基极电流 I 的微小变化经过 9014 放大后，使三极管 9014 集电极电流 I 明显减少，集电极上的电阻两端的电压减少，因此三极管 9014 的 CE 之间的电压升高，当 CE 之间的电压到达高电平时，高电平经过方相器反相后变为低电平，因此 P3.7 口上为低电平，当单片机扫描到 P3.7 口上为低电平时，随即进行止闹操作。在没有声音信号时，三极管 9014 的 CE 之间电压为低电平，经过 CD4069 反相后为高电平，因为 P3.7 是低电平有效，所以不会进行止闹操作。

同时我们可以通过按键开关 K3 来控制 MIC 的灵敏度，当把开关拨向下边时，三极管 9014 集电极上接的是 3.3M 的电阻，使电阻两端电压变高，9014 的 CE 两端的电压相对偏低，CE 不易到达高电平，所以灵敏度变低。当把开关拨向上边时，接在集电极上的是 1M 电阻，同样的道理，9014 的 CE 之间电压相对变高，因此声控电路的灵敏度变高。这种设计非常人性化。

有时我们不需要用声控止闹功能，这时我们可以通过按键 K2 来对声控电路进行开关控制，当开关 K2 按下时，则单片机和声控止闹电路断开，P3.7 接口直接加+5V 高电平，这时声控止闹功能失效，当 K2 没有按下时，则单片机和声控止闹电路连接，可以使用声控止闹功能。

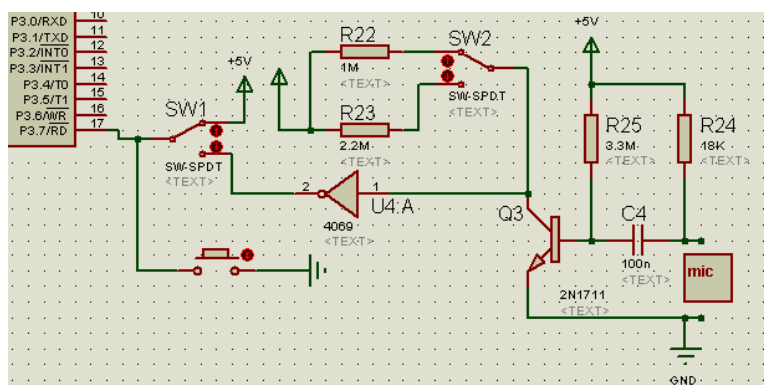


图 9 止闹电路

4. 软件系统设计

系统主程序流程如图 10 所示。

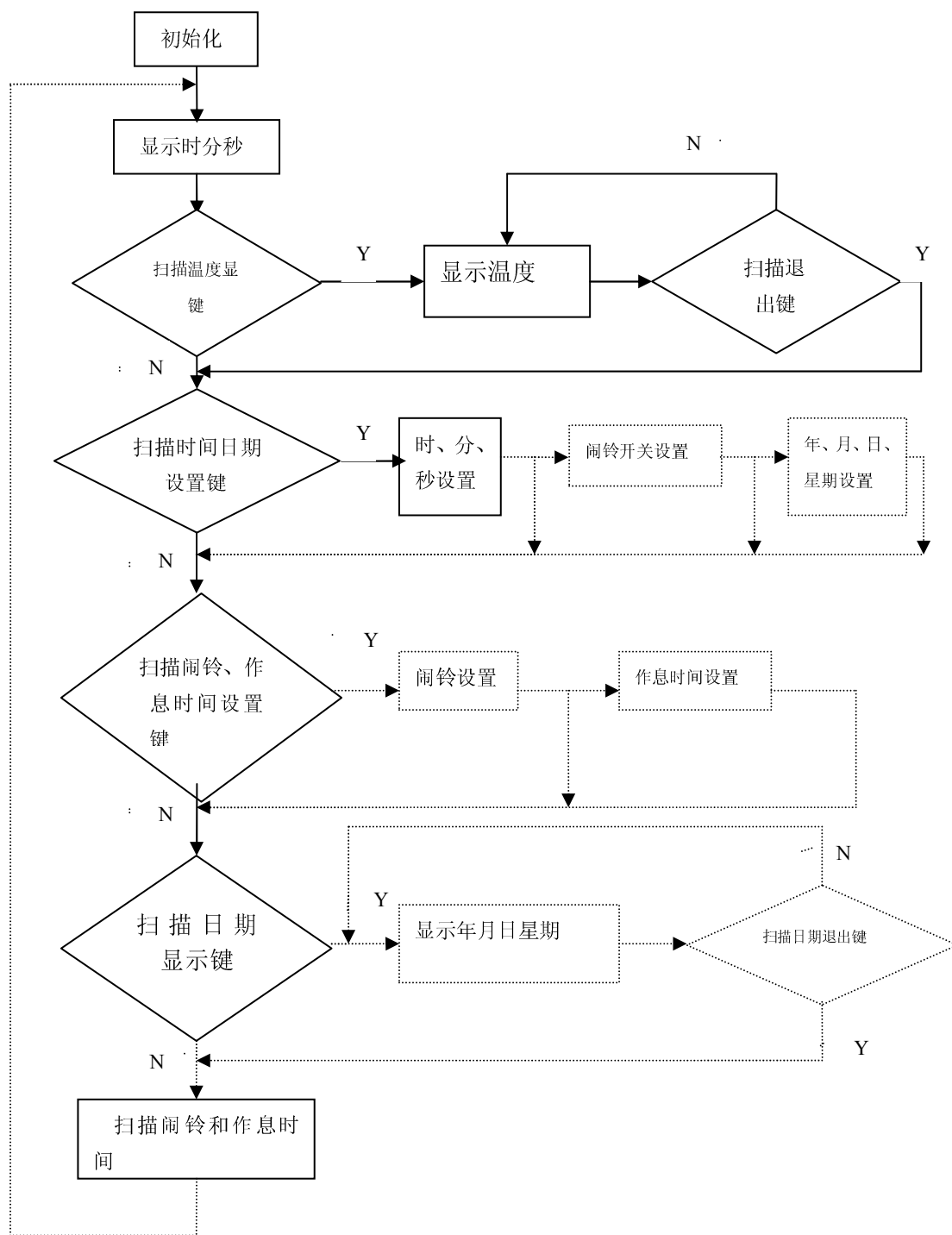


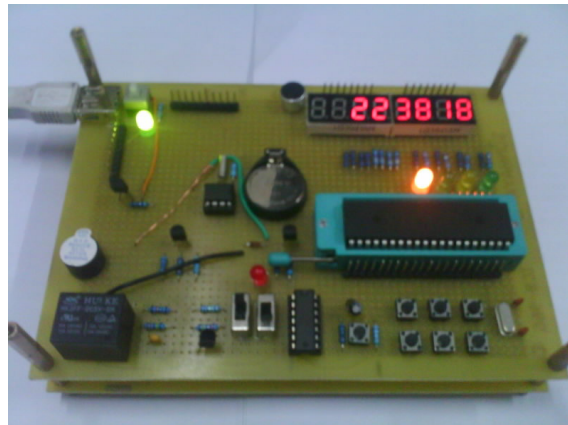
图 10 主程序流程图

程序从 main 函数开始执行，首先进行初始化，对单片机各端口以及各全局变量进行赋初值，然后开始显示时间，并对按键和闹钟进行扫描。如果扫描到闹钟时间到了，就进行声光显示；如果扫描到有按键按下，则开始运行相应的程序。例如：扫描到时间日期设置键按下，则系统进入时分秒设置状态，设置好以后可以选择退出，也可以选择继续对年月日或者闹铃开关进行设置。

5. 测试及结果

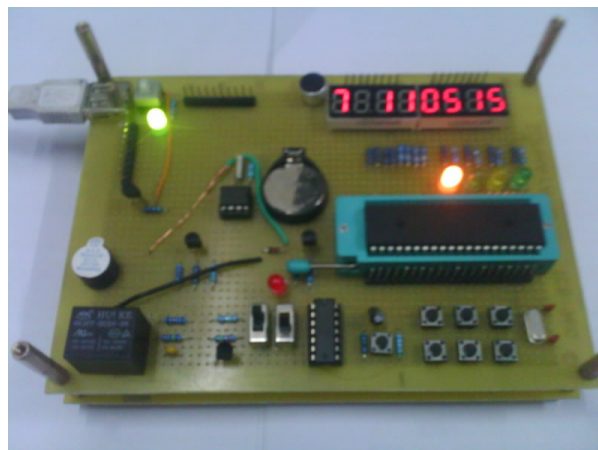
5.1 时间显示

打开数字钟电源开关，接通电源，系统开始工作，H6 亮，表示电源接通，H1 亮表示闹铃处于开启状态，此时 8 位数码管的前 6 位显示（从右边起依次为 1~8 位）时间，此时时间为 22 点 38 分 18 秒。如下图所示。



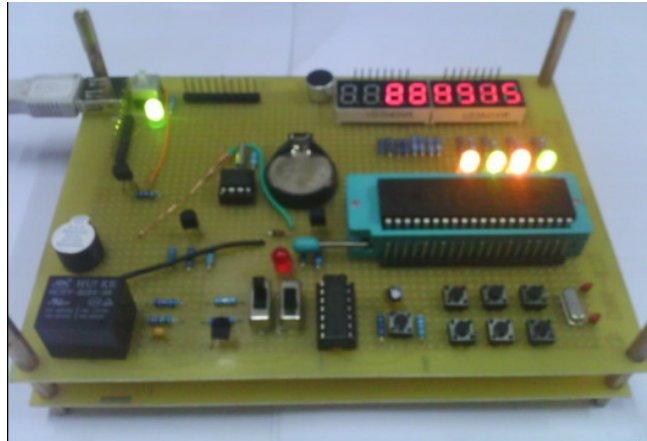
5.2 日历显示

在时、分、秒时间模式下，按 inc (S4) 键，则进入日历显示模式，这时 8 位数码显示管会显示年、月、日和星期，此时显示的日期为 2011 年 5 月 15 日星期日。如下图。



5.3 闹铃及声控止闹测试

在正常时间显示状态下，按 ringc (S1) 键，则进入闹铃时间调整状态，测试时线调好闹铃，闹铃时间到则进行声光提示，如图，H2, H3, H4 亮，蜂鸣器响，此时按下 S6 或对 MIC 发出声响即可止闹。



5.4 温度显示测试

在正常时间显示状态下，按 quit (S2) 键，则进入温度显示状态，如下图所示，此时温度为 25 度。再按 quit 键，则推出温度显示状态，正常显示时间。



第四篇 智能测距避障小车的设计

摘要：本设计的主要目的：一小车为汽车模型，通过超声波测距以直观的显示（LCD 显示）告知驾驶员周围的障碍物的距离，解除了驾驶员泊车或启动车辆时前后探视的困扰。提高驾驶的安全性。本系统主要由单片机模块、显示模块、键盘模块、超声波发射、接收模块、电机驱动模块等五部分组成。单片机模块采用的是 STC89C52 单片机，超声波模块 US-100 的 Trig 发射超声波 ECHO 端接收反射回来的超声波，通过计算时间差和超声波传播速度的乘积的二分之一得到小车距离障碍物的距离。本设计可以实现三个功能，即：输入距离障碍物得距离（距障碍物达到此距离时小车自动停止）、显示距离障碍物的距离、自动避障。输入功能可以自行输入小车所认为的最小安全距离，此输入距离的输入和改变在 Lcd 显示出来。当小车的前端距离某障碍物过近时，左右电机转速改变实现小车拐弯，绕过障碍物。Lcd 时刻显示小车后端距离最近障碍物的距离，并随小车的移动示数发生改变。

关键词：stc52 单片机；us-100；测距；避障

1 设计要求

智能测距避障小车的设计是自驾车测距避障的一个模型。用于倒车时司机掌握障碍物的距离，便于司机及时了解周围的安全情况，增强司机驾车的安全性。整体的设计要求主要体现在结构和功能以下几个方面。

结构要求：

- (1) 该系统由一个带两个直流电机的小车模型和若干模块组成；
- (2) 通过超声波发射、接收方式进行数据测量。

功能要求：

(1) 人为任意输入最小距离，信息确认后，小车开始运动。当小车的距离障碍物的距离达到此值时小车自动停止。前端障碍物过近时实现左右端电机转速改变，并在液晶屏上显示距离信息。

(2) 当小车的距离障碍物的距离达到原输入距离时小车自动停止。

(3) 前端障碍物过近时实现左右端电机转速改变。

2 总体设计

2.1 系统组成及工作原理

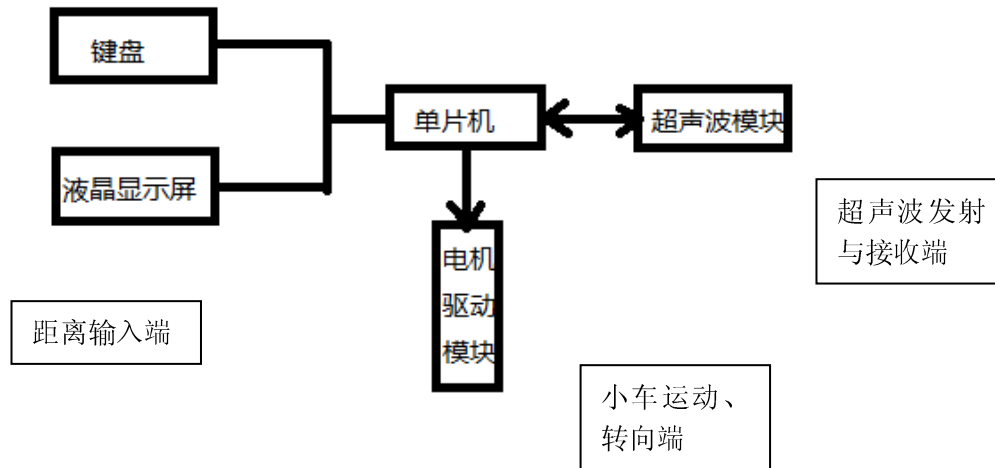


图 1 系统组成框图

距离输入端：

主要由键盘、stc52 单片机、1602 液晶显示屏等部分组成。键盘共有三个按键，对应着实现输入设计的三个功能，完成操作指令的输入；然后将按键信息传入单片机，单片机根据按键信息来控制数据在液晶屏上的显示。

超声波发射与接收端：

主要由 stc52 单片机、超声波发射与接收 US-100 模块等部分组成。US-100 模块共有 5 个引脚，一引脚接 VCC，两引脚接地，另外两个引脚接单片机实现单片机和模块的相互控制。

小车运动与转向端：

由单片机、电机驱动模块、电机组成。电机驱动模块以 L298N 为主体芯片，加一些外围电路，同时实现两个电机的驱动。

3 硬件设计

3.1 硬件组成

距离输入端单片机为核心连接键盘、Lcd；超声波发射与接收端以单片机为核心连接超声波模块。

3.2 单片机核心模块

我们在整体设计中选用的是 STC89C52 单片机，它是一种带 8K 字节闪烁可编程可擦除只读存储器（Flash Programable and Erasable Read Only Memory）的低电压，高性能 CMOS8 的微处理器。该器件采用 ATMEL 高密度非易失存储器制造技术制造，与工业标准的 MCS-51 指令集和输出管脚相兼容。

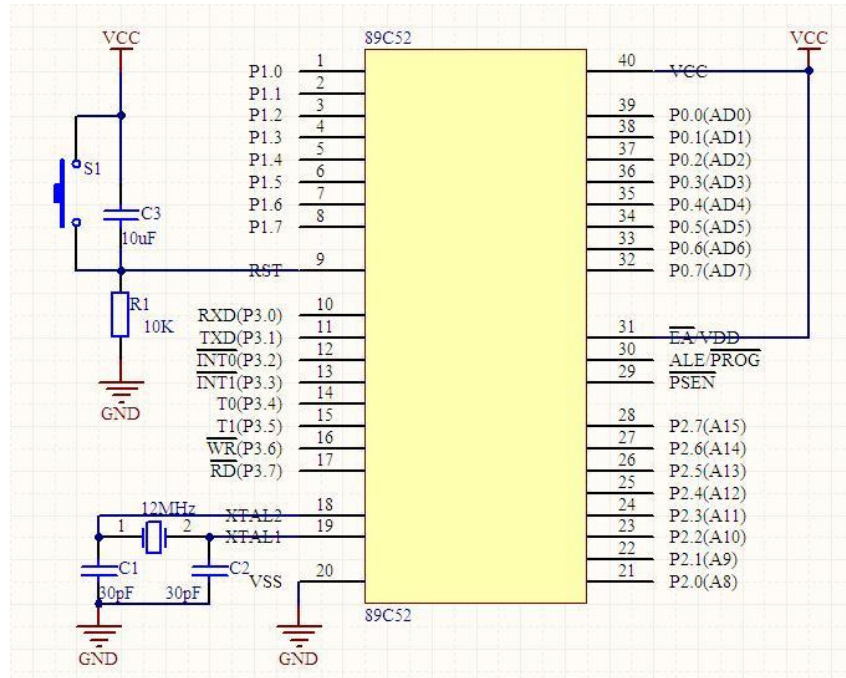


图2 STC89C52 单片机的最小系统

3.3 数据输入端电路

数据输入端的电路主要由 5 部分组成，分别为单片机电路、LCD 电路、键盘电路、电源电路等。

LCD 电路如图 3 所示。

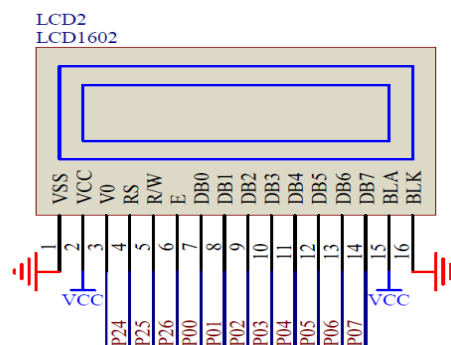


图3 LCD 电路

键盘电路：由三个触控式开关组成，与单片机的 P3 口连接，按键闭合前相应的引脚为高电平，闭合时变为低电平，松开按键后恢复为高电平。键盘电路如图 4 所示。

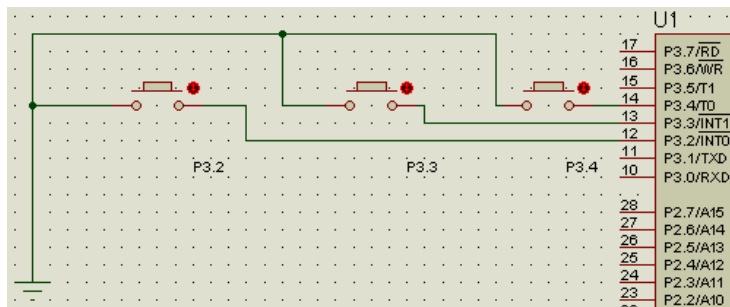


图4 键盘电路

电源电路：负责各个模块电路的电源供电，采用 9v 电源经 L7805 整压供电。电源电路如图 5 所示。

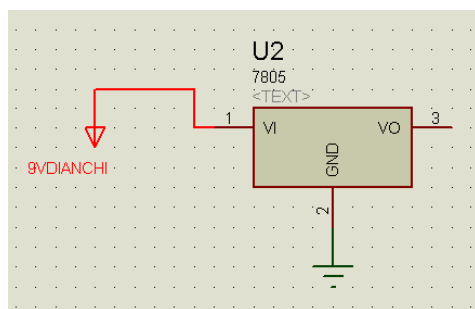


图5 电源电路

US-100 模块：该模块可实现 0—4.5m 非接触测距功能，拥有 2.4—5.5v 的宽电压输入范围，静态功耗低于 2mA，自带温度传感器对测距结果进行校正，工作稳定。

US-100 的主要特点：

(1) 只要在 Trig 引脚输入 10us 以上的高电平，系统便发出 8 个 40KHz 超声波脉冲，然后检测回波信号，当检测到回波信号后，模块还要对温度进行测量，然后根据当前温度对测距结果进行矫正，矫正后的结果通过 Echo 管脚输出；

(2) 在此模式下，模块将距离值转化为 340m/s 时的时间值的二倍，通过 Echo 端输出一高电平，可根据此电平的持续时间来计算距离值；

US-100 的连接电路如图 6 所示。

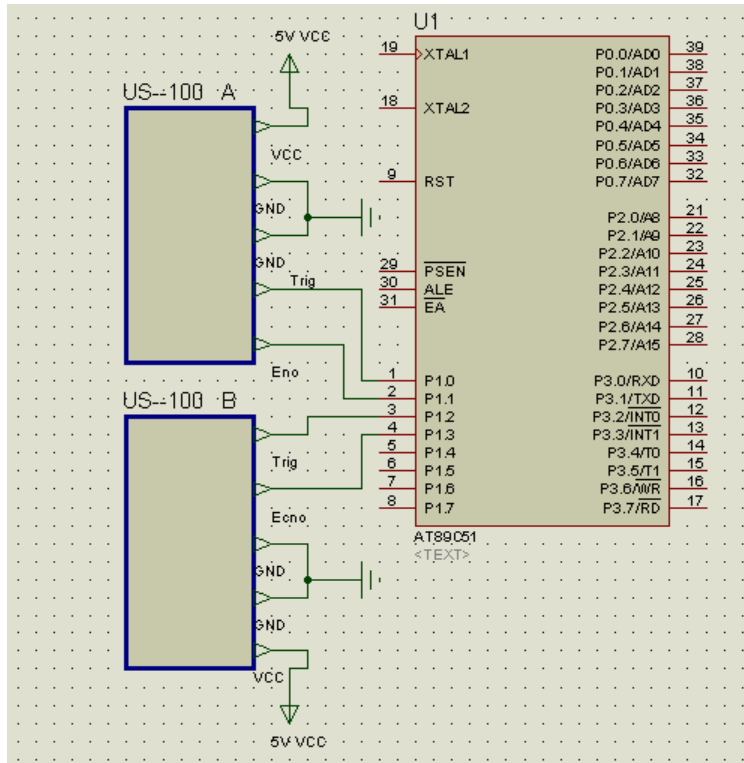


图6 US-100 模块接口电路

3.4 电机驱动电路

电机驱动的电路主要由4部分组成，分别为单片机电路、电源电路、电机模块等。

电源电路：负责各个模块电路的电源供电，采用电池直接供电。输入、输出电路如图7、图8所示。

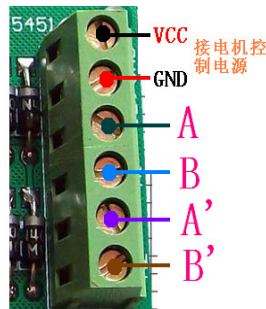


图7 驱动模块输出端

为提高驱动模块的驱动能力，VCC 直接接9v 电池。A、B 接左边电机，A'、B' 接右边电机。

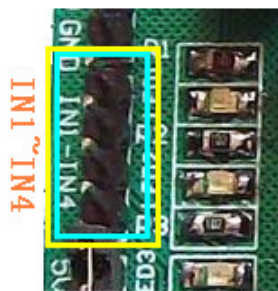


图8 驱动模块输入端

IN1-IN4 为输入端，IN1、IN2 为左电机控制输入端，IN3、IN4 为右电机控制输入端。左、右电机的控制如表 1 和 2 所示。

表 1 左电机控制表

| 电机 | 旋转方式 | 控制端 IN1 | 控制端 IN2 | 控制端 IN3 | 控制端 IN4 | 输入PWM信号改 | |
|----|------|------------|------------|------------|------------|----------|----------|
| | | | | | | 变脉宽可调速 | |
| | | | | | | 调速端 A | 调速端 B |
| M1 | 正转 | 高 | 低 | / | / | 高 | / |

表 2 右电机控制表

| 电机 | 旋转方式 | 控制端 IN1 | 控制端 IN2 | 控制端 IN3 | 控制端 IN4 | 输入PWM信号改 | |
|----|------|------------|------------|------------|------------|----------|----------|
| | | | | | | 变脉宽可调速 | |
| | | | | | | 调速端 A | 调速端 B |
| M1 | 正转 | 高 | 低 | / | / | 高 | / |

4 软件设计

4.1 软件结构

整体程序设计主要分为 NRF905 的初始化、发送函数、接收函数、键值读取、数码管显示等 5 个模块。

(1) 发送和接收部分，主要利用单片机的 I/O 口来模拟 SPI 时序，分别构成 SPI 的读写函数，来执行向缓冲区写入发送信息和读取接收到的信息并写入缓冲区；

(2) 在键值读取部分，主要通过不断扫描单片机与键盘相连的 P1 口的电平变化，从而确定按键的键值，执行相应的功能操作。

4.2 主程序设计

程序的全局变量为 `keyval`，在系统工作过程中，单片机不断扫描 P1 口的电平变化，并将相应的键值信息赋给变量 `keyval`，之后按照相应的指令执行。

在程序的开始定义一个空的数组，当有按键被按下时，就将相应的十六进制数据写入这个空的数组，确认之后便将数组中的数据发送给接收显示终端。如果在确认之前需要更改选项，新的十六进制数据则将数组中原来的数据覆盖，等待发送。

第五篇 宽带直流放大器的设计

摘要：本作品基于宽带放大器的设计，由前级放大模块，增益控制模块，后级功率放大模块，A/D 转换模块和电源模块组成。前级放大压控增益放大器 AD603 放大输入信号，输出倍数可调的电压。再由以 OPA690 和继电器构成的增益控制模块进行电压增益控制。后级功率放大模块由电流反馈放大器 THS3091 来实现。其中功率放大模块采用运放并联设计，此设计是为了增大驱动电流，增强带负载能力。

本设计采用三级放大，增强整个电路的驱动能力，进行合理的级联和阻抗匹配，加入后级驱动，全面提高了增益带宽积和输出电压幅度。应用单片机和数字处理技术，使得结果更加可靠有效。电路增加去耦，滤波等措施，以减少放大器的噪声及抑制自激振荡。经实际电路检测，本设计符合题目的设计要求。

关键词：宽带放大器；AD603；可控增益；单片机

1 作品功能及总体方案

1.1 功能描述

- (1) 系统电压增益 $A_v \geq 40\text{dB}$ ，输入电压有效值 $V_i \leq 20\text{mV}$ 。 A_v 可在 $0 \sim 40\text{dB}$ 范围内手动连续调节。
- (2) 最大输出电压正弦波有效值 $V_o \geq 2\text{V}$ ，输出信号波形无明显失真。
- (3) 3dB 通频带 $0 \sim 5\text{MHz}$ ；
- (4) 放大器的输入电阻 $\geq 50\Omega$ ，负载电阻 $(50 \pm 2)\Omega$ 。

1.2 系统组成及工作过程

系统总体框图如图 1 所示。

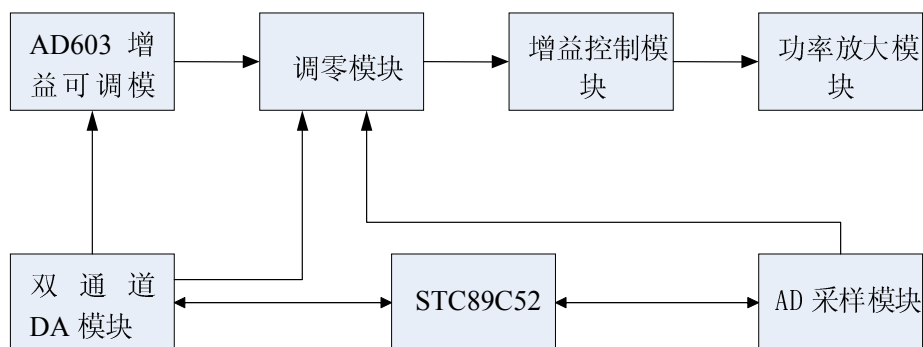


图 1 系统组成框图

2 硬件设计

2.1 前级放大模块

放大部分采用可变增益宽带放大器 AD603, AD603 是一款低噪声, 精密控制的可变增益放大器, 温度稳定性高, 最大增益误差为 0.5dB, 它有多种选择模式, 当带宽为 90MHz 时, 增益范围为: -10 dB-----+30dB; 当带宽为 9MHz 时, 增益范围为 10 dB---50 dB, 当带宽为 30MHz 时, 增益范围为 0 dB---40 dB。可根据引脚 V_{out} 和 FDBK 的连接方式的变化来选择。此种方式下输入控制电压 U 的范围为 -0.5---+0.5V, 增益随电压成线性关系。增益和控制电压的关系为: $Gain(dB)=40*U+G_0$, 5 脚和 7 脚间接 3k 反馈电阻, 使 $G_0=21.58dB$ 。其电路如图 2 所示。

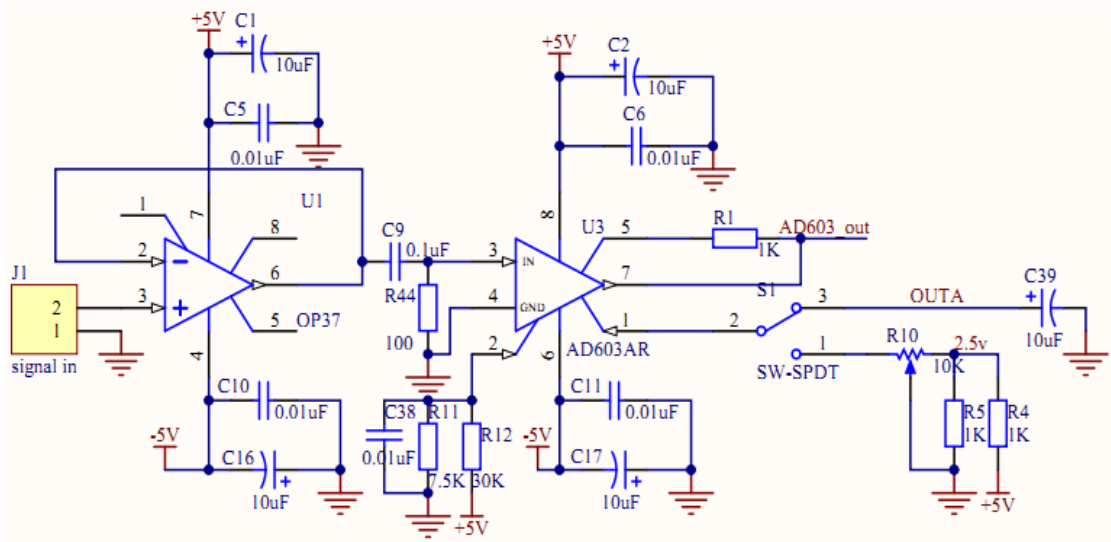


图 2 AD603 前级放大电路

2.2 AD 采样前端电路

由于 AD603 最大有 30mV 的输出漂移, 因此在电路设计时我们必须要对其直流漂移进行调零处理, 以免影响直流信号的放大。如图 2 所示, 经 AD 采集并经单片机处理, 测出当输入电压为零时, 输出端存在的直流漂移电压, 再由 DA 输出与漂移电压大小成比例, 极性相反的电压反馈回信号输入端, 以调节输入端的零偏。此处我们选择 TI 公司的 TLV5638 作为调零用 DA。其电路如图 3 所示。

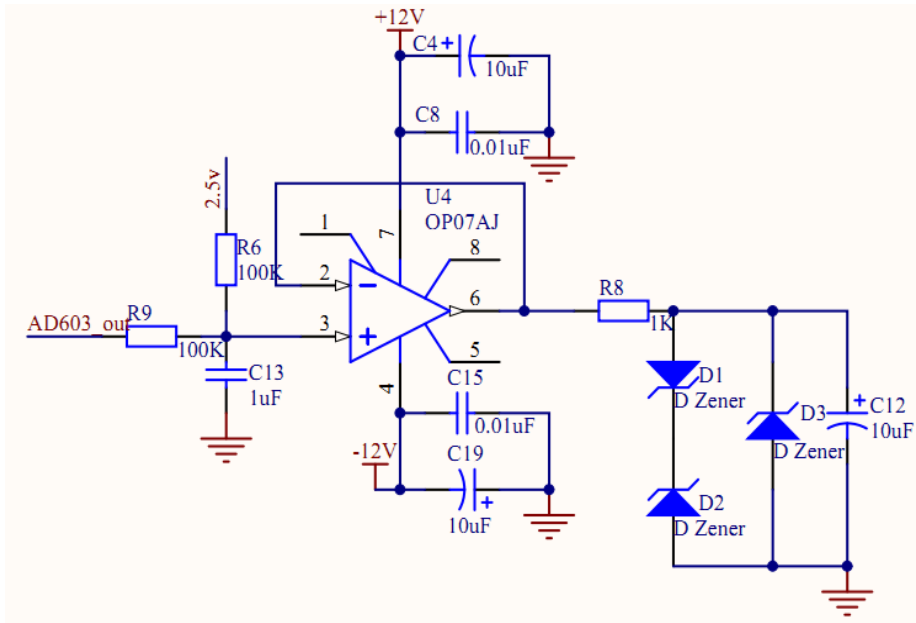


图 3 AD 采样前端电路

2.3 抑制直流零点漂移

调零电路由 OPA690 构成一个加法器。由 D/A 转换器输出调节电压加在 OPA690 输入端，对 AD603 输出的直流偏置电压进行校正。OPA690 第二路采用加法方式输入 -3V，调节双通道 D/A 转换器 TLV5638 输出的单极性电压转换为双极性。其电路如图 4 所示。

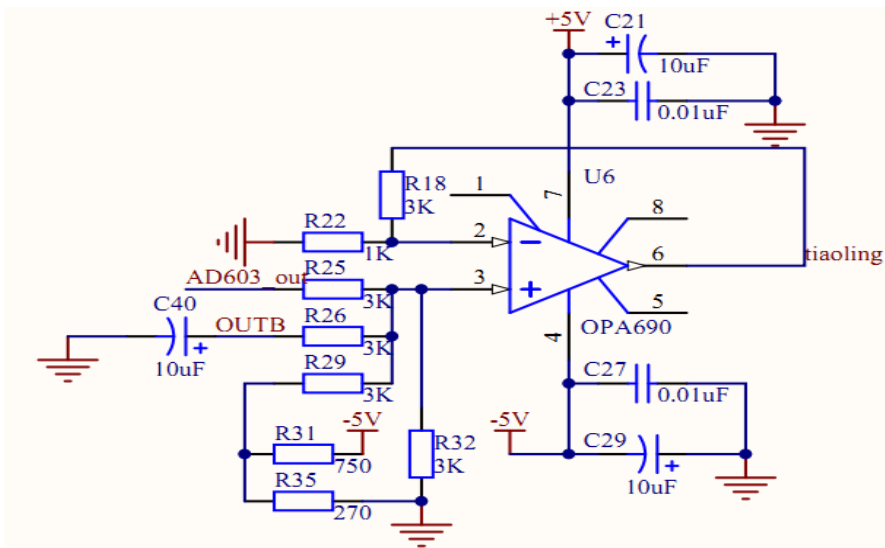


图 4 调零电路

2.4 滤波电路

滤波电路采用最平坦的巴特沃斯 5MHz 无源 LC 低通滤波器。其电路如图 5 所示。

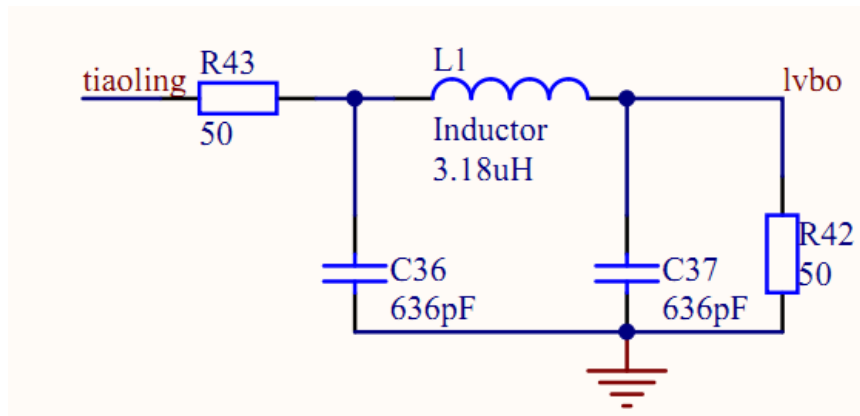


图 5 5MHz 无源低通滤波器

2.5 程控放大电路

程控放大电路由 OPA690 构成增益为 1 和 5 两个挡，后面为一个继电器切换衰减网络，调节电路的增益。其电路如图 6 所示。

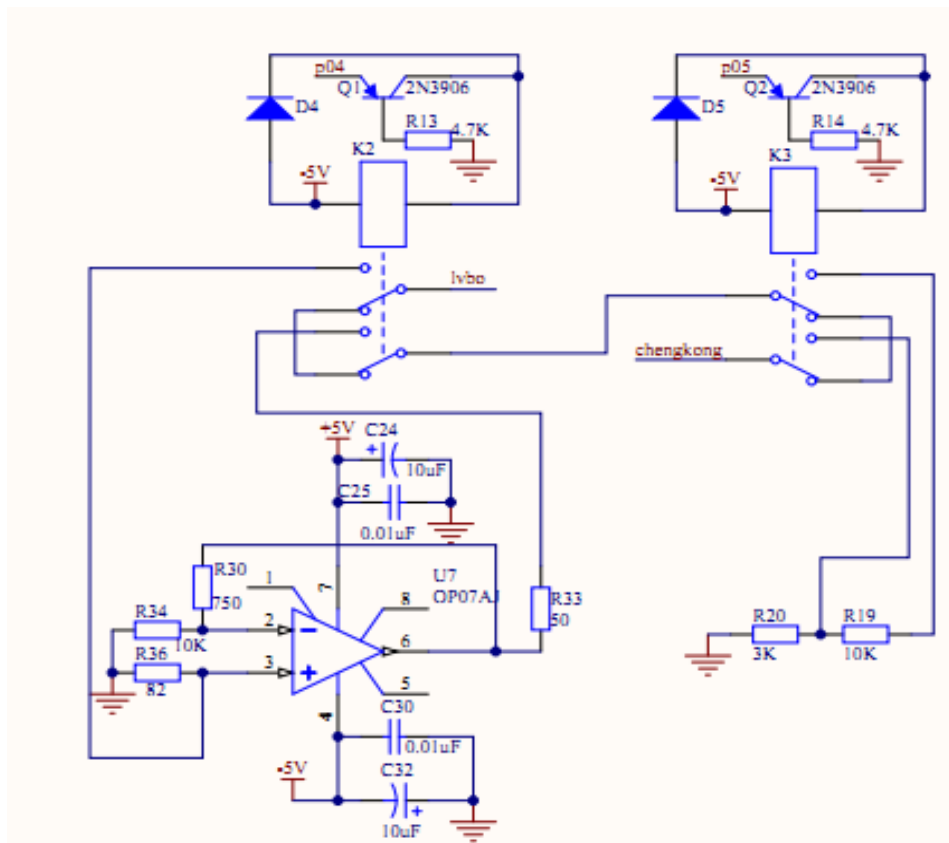


图 6 程控放大电路

2.6 后级功率放大电路

采用电流反馈型运放 THS3091 做 5 倍功率放大。THS3091 具有 7300V/us 的摆率，带宽不小于 200MHz，其最大输出电流为 250mA，本放大器采用两片 THS3091 并联，每片 THS3091 为 50Ω 负载提供一半电流，提高输出功率。其电路如图 7 所示。

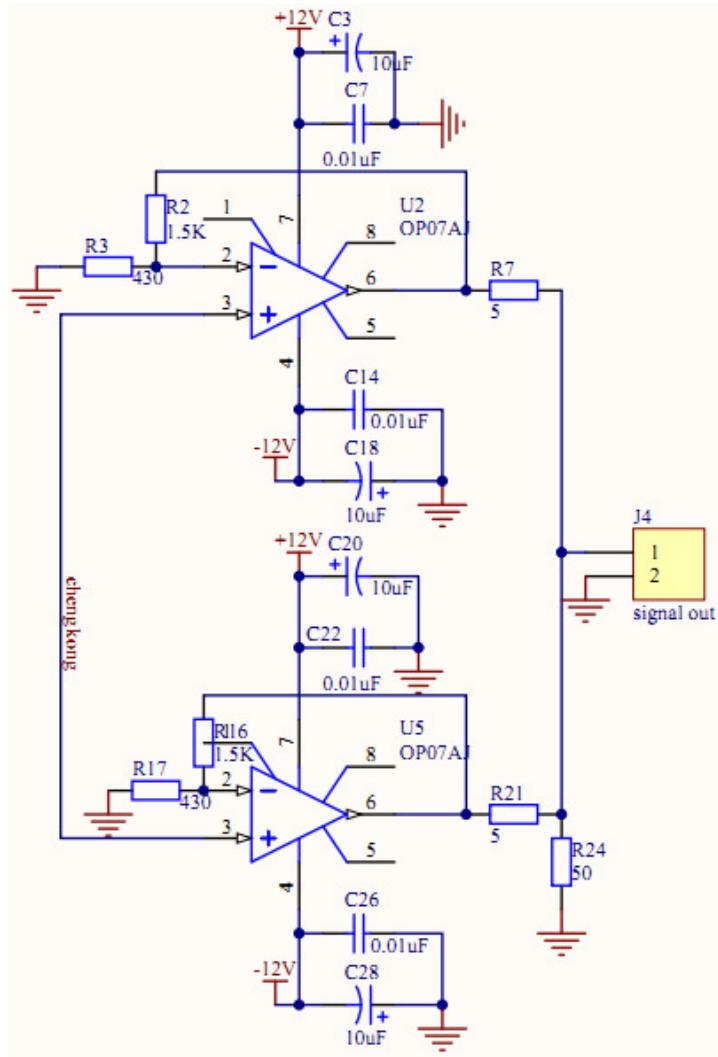


图7 功率放大电路

3 软件设计

本系统软件设计部分基于 STC89C52 单片机，主要完成增益控制、直流零点自动校准功能、按键处理等功能。主程序流程图如图 8 所示。

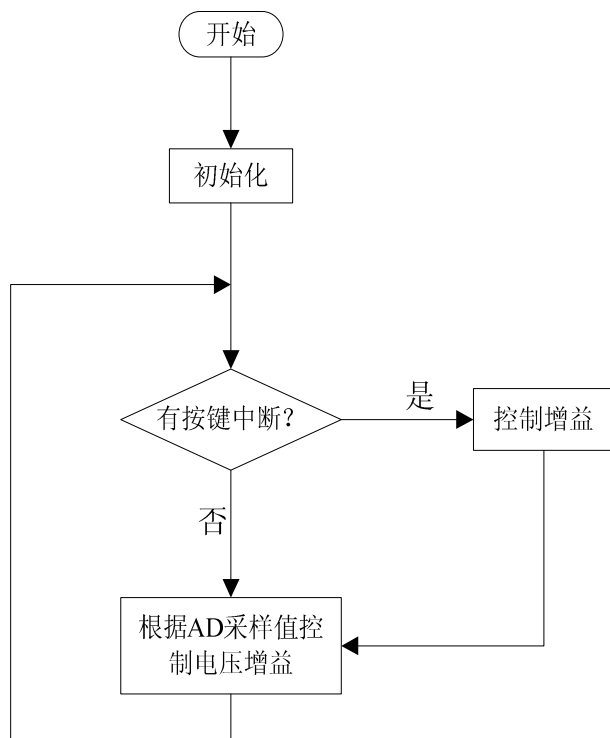


图8 主程序流程图

4 系统测试

测试方法：用函数发生器产生频率 1MHz，峰峰值分别为 20mV，50mV 正弦波送入进行测量。测试数据如表 1 所示。

表 1 测试数据

| 频率 | 输入信号峰峰值 | 输出信号峰峰值 | 增益 | 频率 | 输入信号峰峰值 | 输出信号峰峰值 | 增益 |
|----|---------|---------|--------|----|---------|---------|--------|
| 1M | 20mV | 6.7V | 50.4dB | 1M | 50mV | 8.22V | 44.3dB |
| 2M | 20mV | 6.14V | 49.7dB | 2M | 50mV | 8.08V | 44.1dB |
| 3M | 20mV | 6.08V | 49.6dB | 3M | 50mV | 7.84V | 43.9dB |
| 4M | 20mV | 6V | 49.5dB | 4M | 50mV | 7.61V | 43.6dB |
| 5M | 20mV | 4V | 46dB | 5M | 50mV | 4.56V | 39.1dB |
| 6M | 20mV | 3V | 43.5dB | 6M | 50mV | 4.04V | 38.1dB |

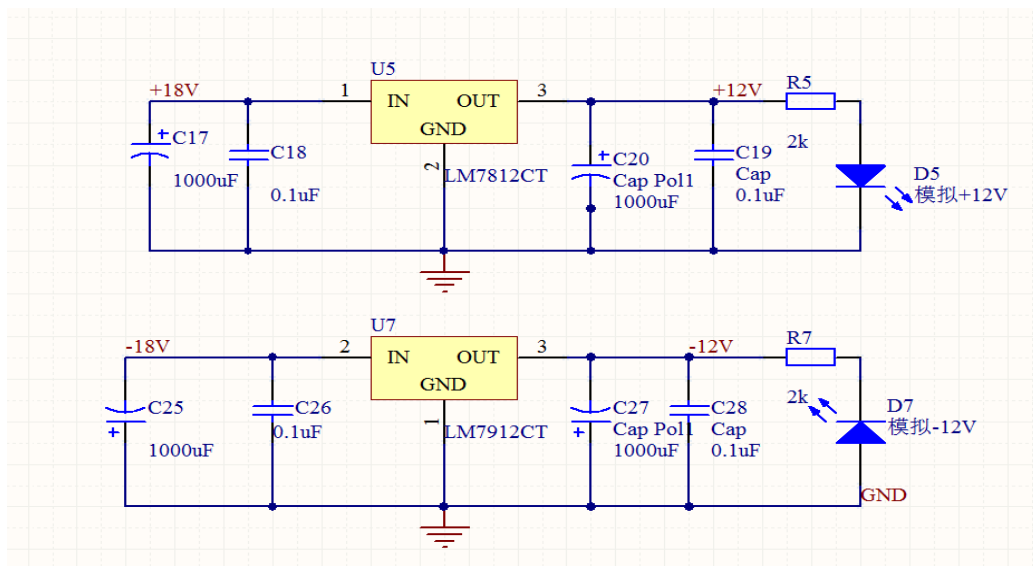
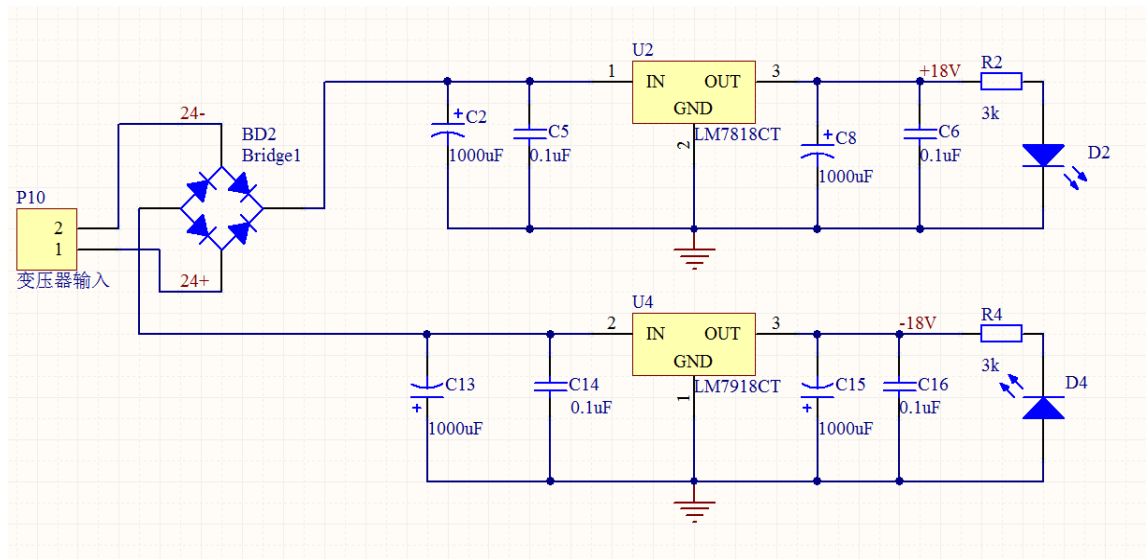
测试结果分析：从数据可以看出，信号增益程控可调，最大增益，最小输入信号幅度均达到题目指标要求。最大输出电压正弦波峰峰值 $V_o \geq 2V$ ，输出信号波形无明显失真。

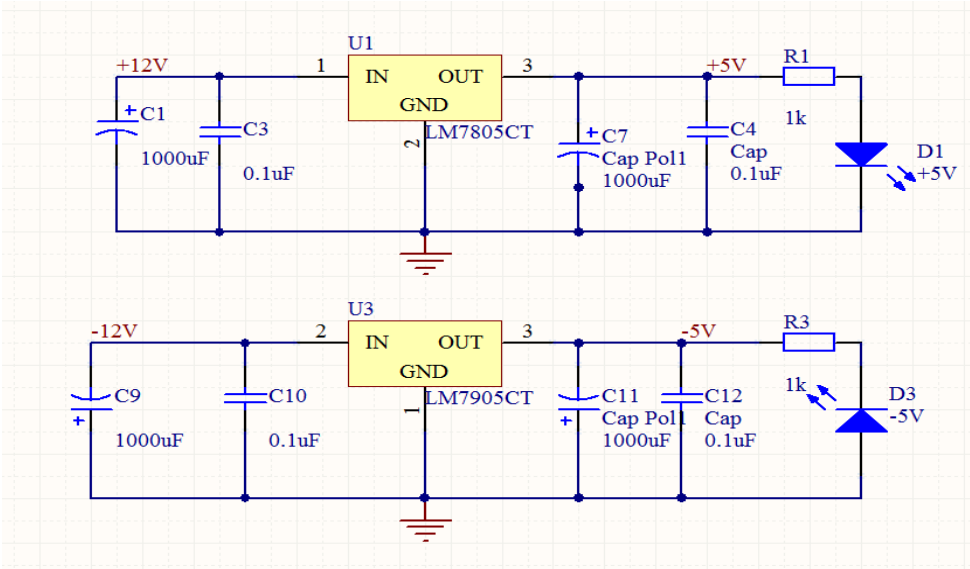
5 总结

本系统由可变增益放大，功率放大，单片机控制及自动直流偏移调零等模块组成。第一级可变增益放大模块采用可变增益放大器 AD603 实现从 0 到 40dB 可变增益放大；第二级固定增益放大模块采用宽带运放 OPA690 实现 3 倍固定增益放大兼调零作用，第三级功率放大采用两路 THS3901 并联配扩流的方式分别对信号进行功率放大，再进行功率合成，从而实现题目要求的 2V 有效值输出。本设计对压控增益器件和宽带高速运放进行合理的级联和匹配，同时加入自动直流偏移调零电路，全面提高了系统增益带宽积，增强了稳定性，抑制了直流零点漂移。

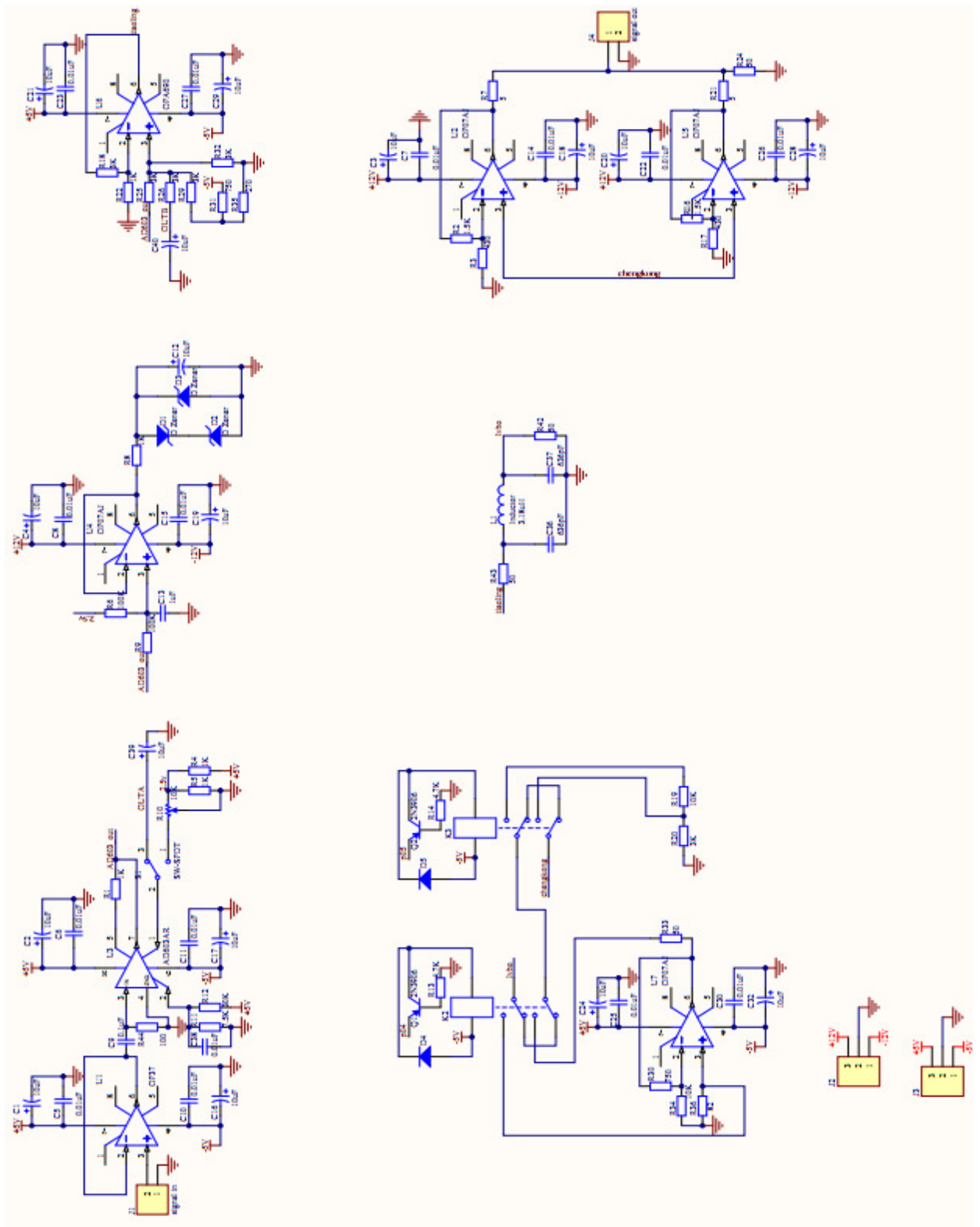
附录

1 直流稳压电源原理图





2 整体电路设计



3 AD/DA 模块

