

一、项目参加人员、负责内容以及技术特长：

主要人员	负责内容	技术特长
	自制稳压电源的设计，撰写论文 电路精度的调节，	数字电路设计，应用软件
	键盘、数控部分、数字显示的设计， 单片机编程	单片机编程以及电路设计
	D/A 转换以及精度，输出电路的设计	模拟电路设计

二、项目背景

数控直流稳压电源是电子技术常用的设备之一，广泛的应用于教学、科研等领域。传统的多功能直流稳压电源功能简单、难控制、可靠性低、干扰大、精度低且体积大、复杂度高。普通直流稳压电源品种很多，在家用电器和其他各类电子设备中，通常都需要电压稳定的直流电源供电。但在实际生活中，都是由 220V 的交流电网供电。这就需要通过变压、整流、滤波、稳压电路将交流电转换成稳定的直流电。滤波器用于滤去整流输出电压中的纹波，一般传统电路由滤波扼流圈和电容器组成，若由晶体管滤波器来替代，则可缩小直流电源的体积，减轻其重量，且晶体管滤波直流电源不需直流稳压器就能用作家用电器的电源，这既降低了家用电器的成本，又缩小了其体积，使家用电器小型化。

电源技术尤其是数控电源技术是一门实践性很强的工程技术，服务于各行各业。电力电子技术是电能的最佳应用技术之一。当今电源技术融合了电气、电子、系统集成、控制理论、材料等诸多学科领域。随着计算机和通讯技术发展而来的现代信息技术革命，给电力电子技术提供了广阔的发展前景，同时也给电源提出了更高的要求。随着数控电源在电子装置中的普遍使用，普通电源在工作时产生的误差，会影响整个系统的精确度。电源在使用时会造成很多不良后果，世界各国纷纷对电源产品提出了不同要求并制定了一系列的产品精度标准。只有满足产品标准，才能够进入市场。

随着经济全球化的发展，满足国际标准的产品才能获得进出的通行证。数控电源是从 80 年代才真正的发展起来的，期间系统的电力电子理论开始建立。这些理论为其后来的发展提供了一个良好的基础。在以后的一段时间里，数控电源技术有了长足的发展。但其产品存在数控程度达不到要求、分辨率不高、功率密度比较低、可靠性较差的缺点。因此数控电源主要的发展方向，是针对上述缺点不断加以改善。单片机技术及电压转换模块的出现为精确数控电源的发展提供了有利的条件。新的变换技术和控制理论不断发展，各种类型专用集成电路、数字信号处理器件的研制应用，到 90 年代，已出现了数控精度达到 0.05V 的数控电源，功率密度达到每立方英寸 50W 的数控电源。目前在电力电子器件方面，几乎都为旋纽开关调节电压，调节精度不高，而且经常跳变，使用麻烦。随着人们生活水平的不断提高,数字化控制无疑是人们追求的目标之一,它所给人带来的方便也是不可否定的,其中数控制直流稳压电源就是一个很好的典型例子。但人们对它的要求也越来越高,要为现代人工作、科研、生活提供更好的更方便的设施，就需要从数字电子技术入手，一切向数字化和智能化方向发展。

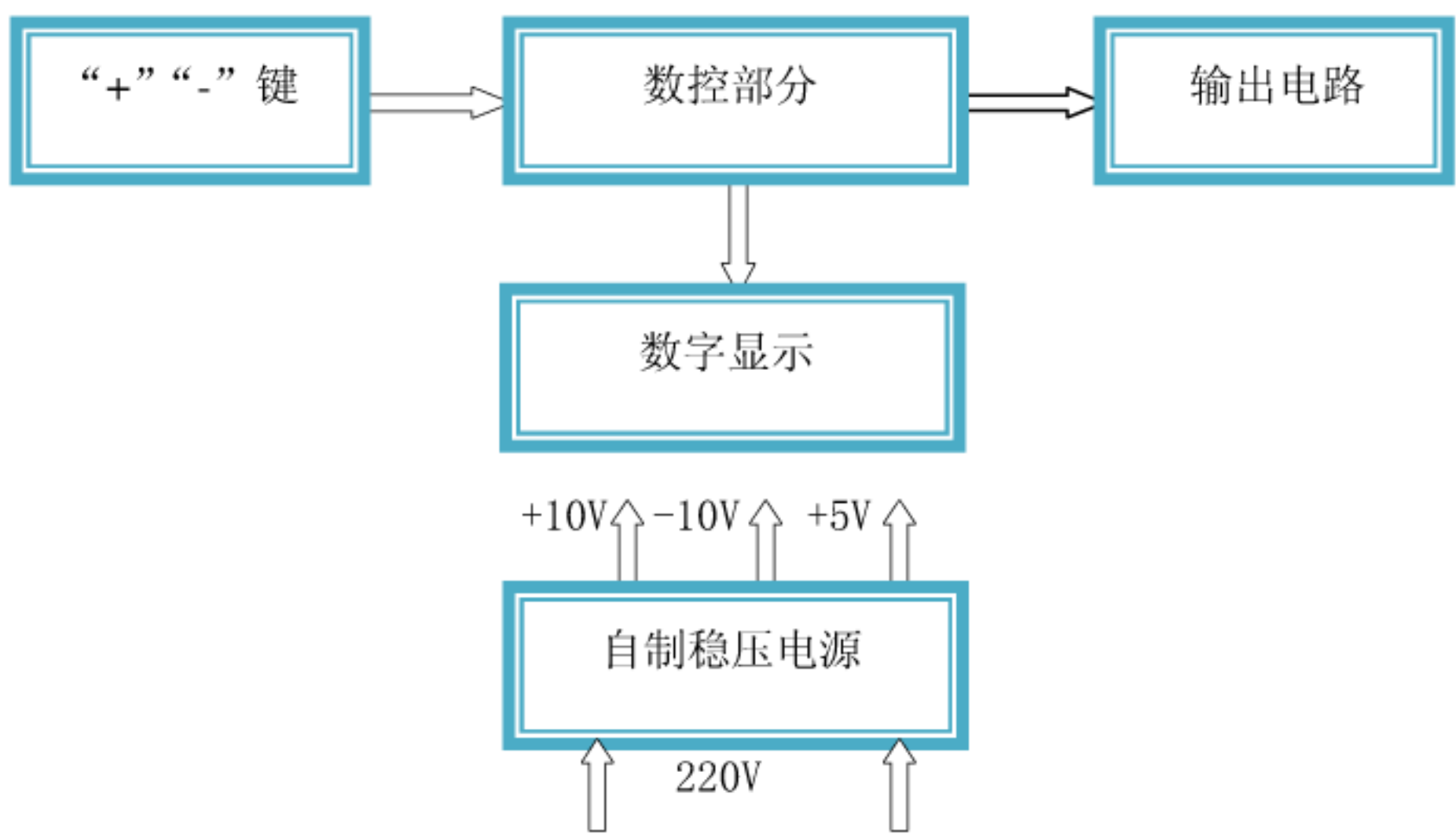
数字化智能电源模块是针对传统智能电源模块的不足提出的，数字化能够减少生产过程中的不确定因素和人为参与的环节数，有效地解决电源模块中诸如可靠性、智能化和产品一致性等工程问题，极大地提高生产效率和产品的可维护性。

### 三、主要研究内容

#### 一、设计任务

设计出有一定输出电压范围和功能的数控电源。

其原理示意图如下：



#### 二、设计要求

##### 1. 基本要求

- (1) 输出电压：范围0~+9.9V，步进0.1V，纹波不大于10mV；
- (2) 输出电流：500mA；
- (3) 输出电压值由数码管显示；
- (4) 由“+”、“-”两键分别控制输出电压步进增减；
- (5) 为实现上述几部件工作，自制一稳压直流电源，输出±10V，+5V。

##### 2. 发挥部分

- (1) 输出电压可预置在0~9.9V之间的任意一个值；
- (2) 用自动扫描代替人工按键，实现输出电压变化（步进0.1V不变）；
- (3) 扩展输出电压种类（比如三角波等）。

### 四、总体思路与研究方案

#### 1. 总体思路



随着时代的发展，数字电子技术已经普及到我们生活、工作、科研等各个领域，本文将介绍一种数控直流电源，本电源由电源电路、显示电路、控制电路、数模转换电路四部分组成。准确说就是电源电路提供各个芯片电源、数码管、放大器所需电压，显示电路用于显示电源输出电压的大小，同时分析了数字技术和模拟技术相互转换的概念。与传统的稳压电源相比具有操作方便，电源稳定性高以及其输出电压大小采用数码显示的特点。

数控电压源是最常用的仪器设备，在科研及实验中都是必不可少的。目前所使用的直流可调电源中，几乎都为旋纽开关调节电压，调节精度不高，而且经常跳变，使用麻烦。利用数控电压源，可以达到每步 0.1V 的精度，输出电压范围 0.1V 至 9.9V，电流可以达到 0.02A。针对以上问题，本课题设计了一种以单片机为核心的数控式高精度简易直流电源的设计，该电源采用数字调节、闭环实时监控、输出精度高，特别适用于各种有较高精度要求的场合。其设计方法是由单片机通过 D/A，控制驱动模块输出一个稳定电压，同时稳压方法采用三端可调稳压管进行调整，输出电压通过电阻反馈给运放，与设定值进行比较，若有偏差则调整输出。工作过程中，单片机输出驱动 LED 显示，通过键盘可设置和调整电压值。该电路具有设计简单，应用广泛，精度较高等特点。

本文介绍的简易数控直流稳压电源，其中输出电压的调节是通过“+”和“-”两个按键来操作的；步进电压精确到 0.1V 去控制可逆计数器分别作加，减计数；可逆计数器的二进制数字输出分两路运行，一路用于驱动数字显示电路，精确显示当前输出电压值；另一路进入数模转换电路（D/A 转换电路）；数模转换电路将数字量按比例，转换成模拟电压，然后经过射极跟随器的控制，调整输出级，使输出稳定直流电压。

## 2. 实施方案

2.1 目前数控电压源已广泛使用，要实现目标其方案比较多，主要有以下几种方案：

2.1.1 方案一：采用单片机的数控电压源的设计

采用常用的 AT89C51 单片机作为控制器，P0 口和 DAC1232 的数据口直接相连，DA 的各个端口连接后接 P3.4，和接单片机的端，让 DA 工作在单缓冲方式下。DA 的 8 脚接参考电压，DA 的基准电压接-10V 电源，所以在 DAC 的 8 脚输出电压的分辨率为约等于 0.1V，也就是说 DA 输入数据端每增加 1，电压增加 0.1V。通过运放 OP07 将 DA 的输出电流转化为电压，再通过运放 OP07 将电压反相并放大输出电压并稳压，最后通过示波器观察其波纹，其硬件框图如图 2.1 所示：

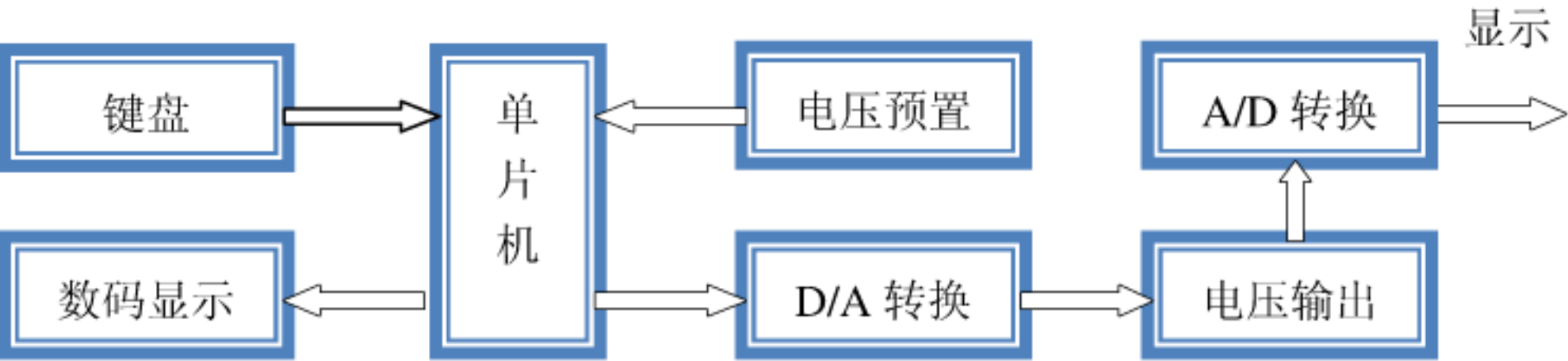


图 2.1 方案一硬件框图

2.1.2 方案二：采用调整管的双计数器的数控电压源的设计

此方案采用传统的调整管方案，主要特点在于使用一套双计数器完成系统的控制功能，其中二进制计数器的输出经过 D/A 变换后去控制误差放大的基准电压，以控制输出步进。十进制计数器通过译码后数码管显示输出电压值，为了使系统工作正常，必须保证双十计数器同步工作。其硬件框图如图 2.2 所示：

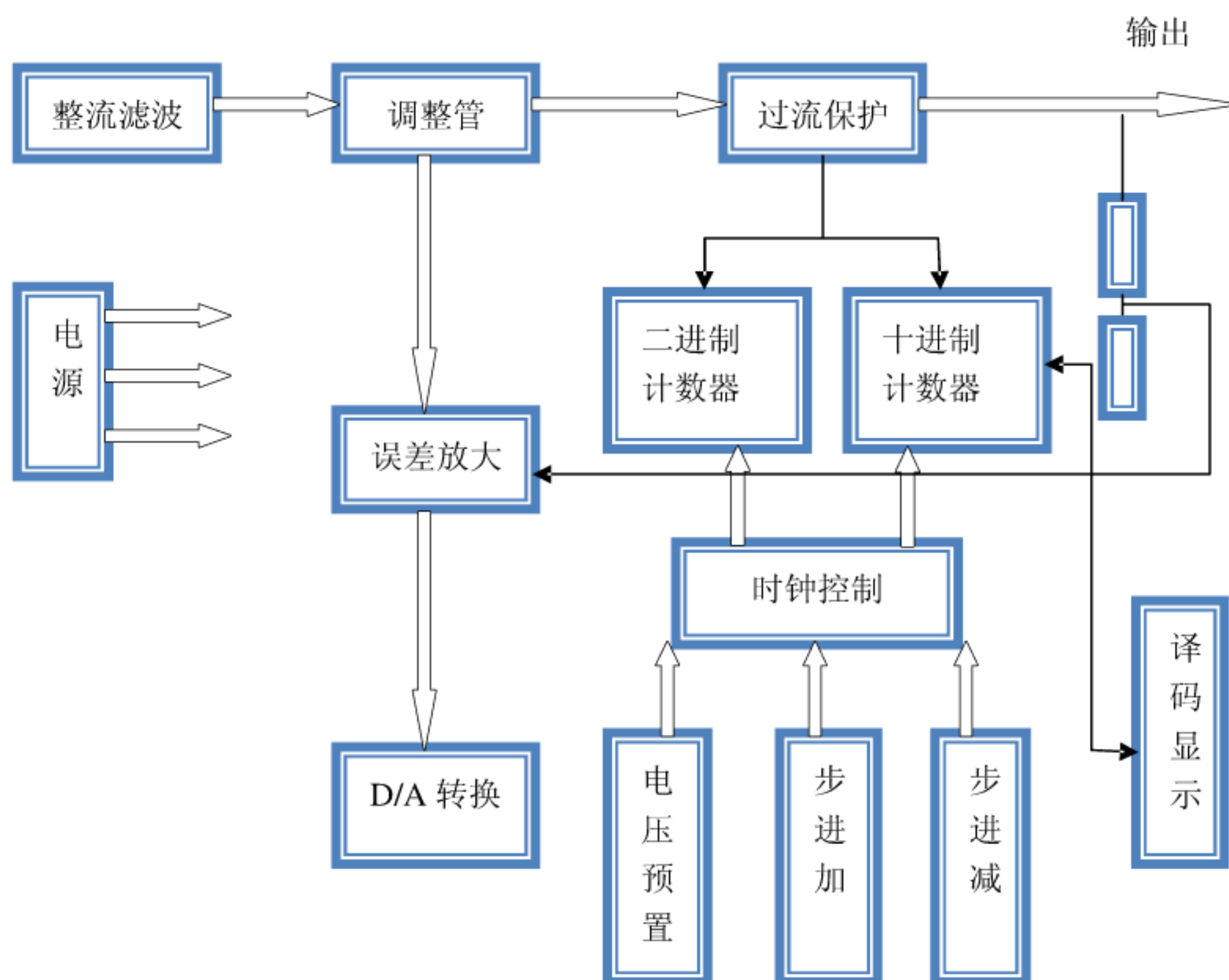


图 2.2 方案二硬件框图

### 2.1.3 方案三：采用调整管的十进制计数器的数控电压源的设计

此方案不同于方案之二处在于使用一套十进制计数器，一方面完成电压的译码显示，另一方面其作为 EPROM 的地址输入，而由 EPROM 的输出经 D/A 变换后控制误差放大的基准电压来实现输出步进，只使用了一套计数器，回避了方案二中必须保证双计数器同步的问题，但由于控制数据烧录在 EPROM 中，使系统设计灵活性降低。其硬件框图如图 2.3 所示

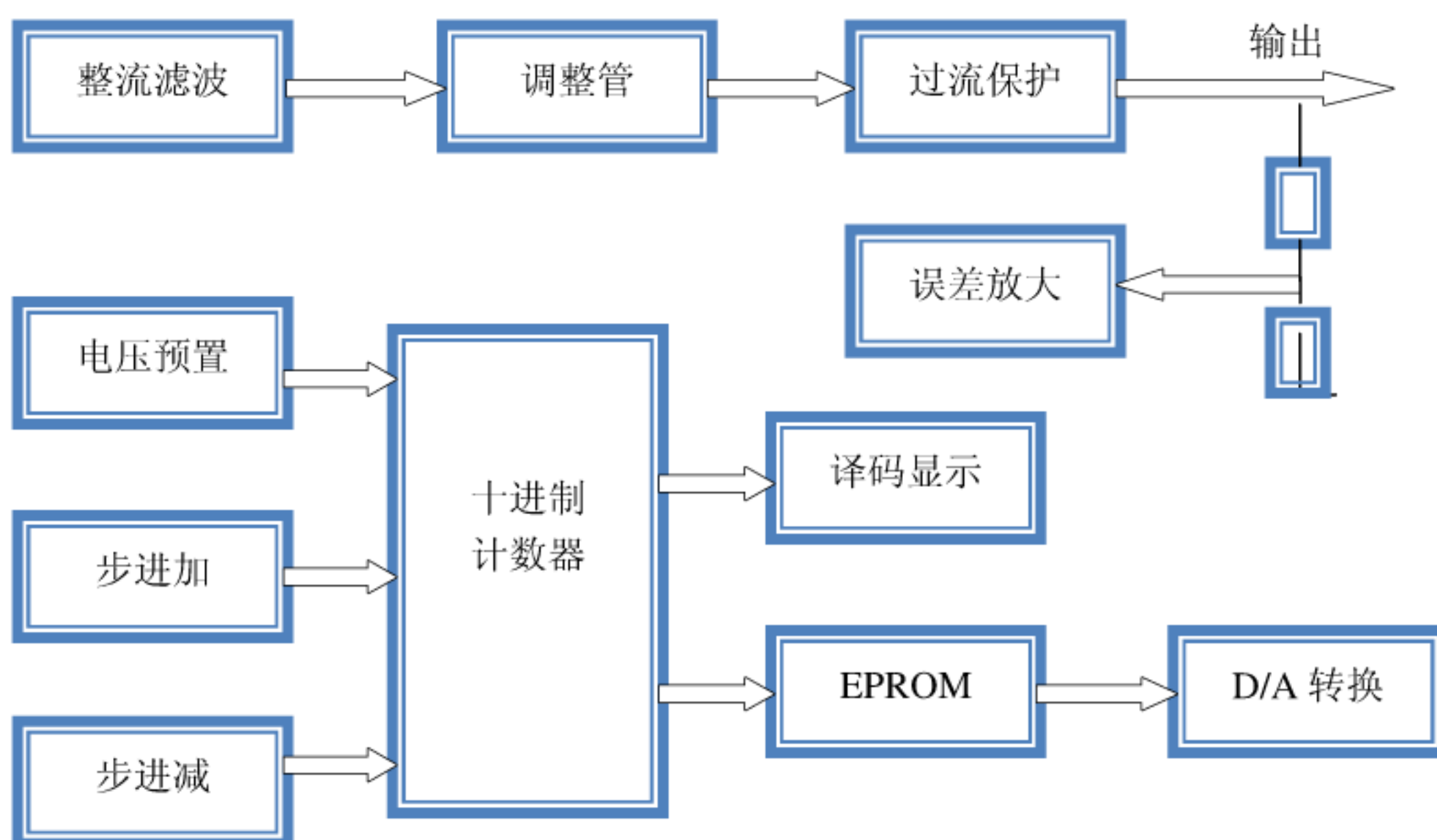


图 2.3 方案三硬件框图



## 2.2 方案比较

### 2.2.1 数控部分的比较

方案二、三中采用中、小规模器件实现系统的数控部分，使用的芯片很多，造成控制电路内部接口信号繁琐，中间相互关联多，抗干扰能力差。在方案一中采用了 AT89C51 单片机完成整个数控部分的功能，同时，AT89C51 作为一个智能化的可编程器件，便于系统功能的扩展。

### 2.2.2 输出部分的比较

方案二、三中采用线性调压电源，以改变其基准电压的方式使输出步进增加或减少，这不能不考虑整流滤波后的纹波对输出的影响，而方案一中使用运算放大器放大电压，由于运算放大器具有很大的电源电压抑制化，可以大大减少输出端的纹波电压。

### 2.2.3 显示部分的比较

方案二、三中的显示输出是对电压的量化值直接进行译码显示输出，显示值为 D/A 变化输入量，由于 D/A 变换与功率驱动电路引入的误差，显示值与电源实际输出值之间可能出现较大偏差，而方案一中采用三位一体的数码管直接对电压值进行显示。

总之，方案一的优点是具有精度高，使用方便，硬件电路简单等特点，它使用了单片机，使得进一步扩展功能较为方便；方案二、三的优点是电路结构简单，其缺点是使用比较复杂，精度没有那么高。

考虑到各种因素，本设计采用方案一。

## 2.3 自制稳压电源的设计

在自制稳压电源设计中，我们采用 TPAN-2P3S 变压器将 220V 交流电压降压处理，再通过 2W005G 桥式整流电路得到正向电压，然后通过 7805、7810 三段稳压器电路，得到+5V、+10V 的直流电源，再在+10V 的基础上加一方向器即可得到需要的-10V 直流电压。其硬件框图如图 2.4 所示

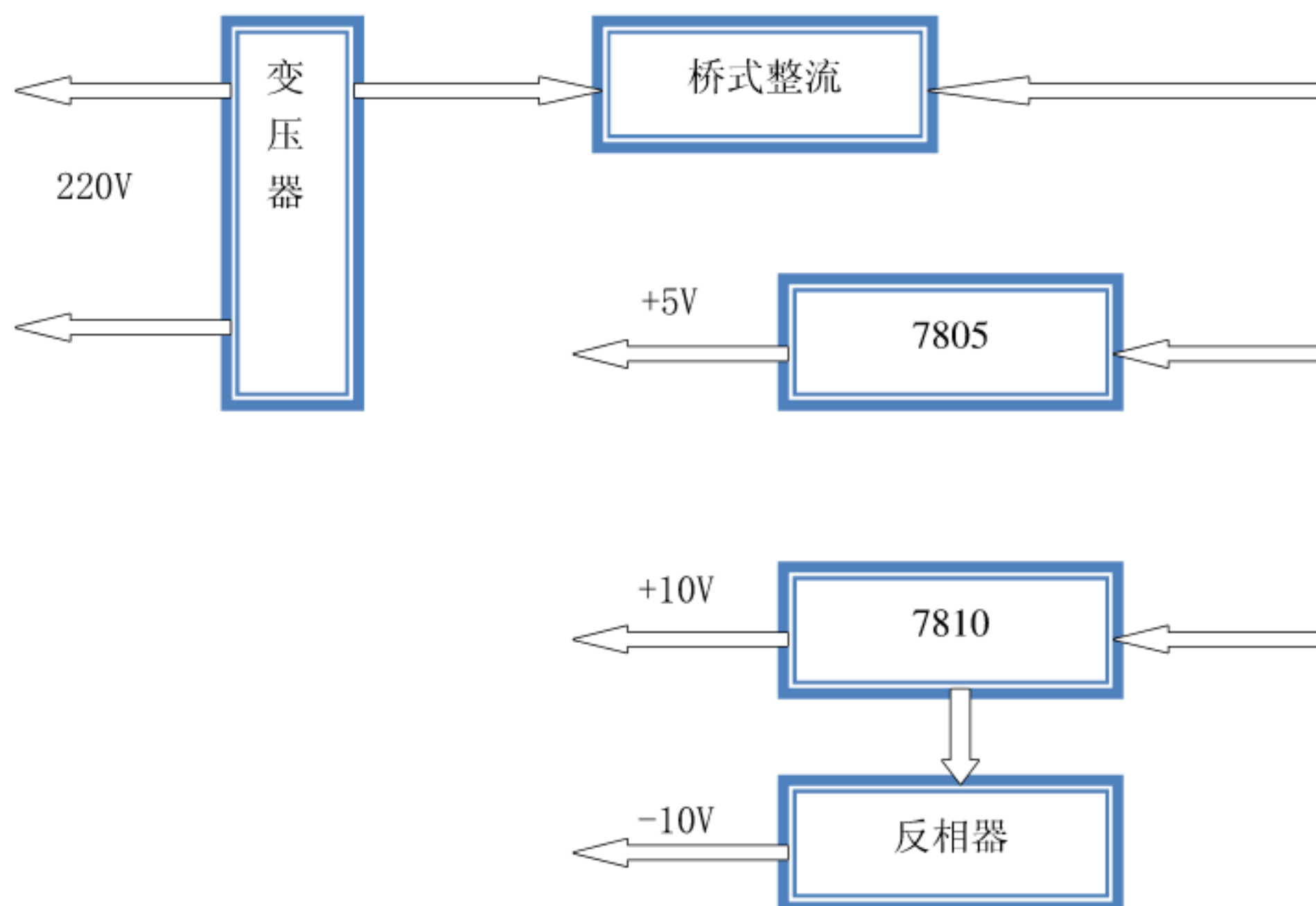


图 2.4 自制稳压电源硬件框图

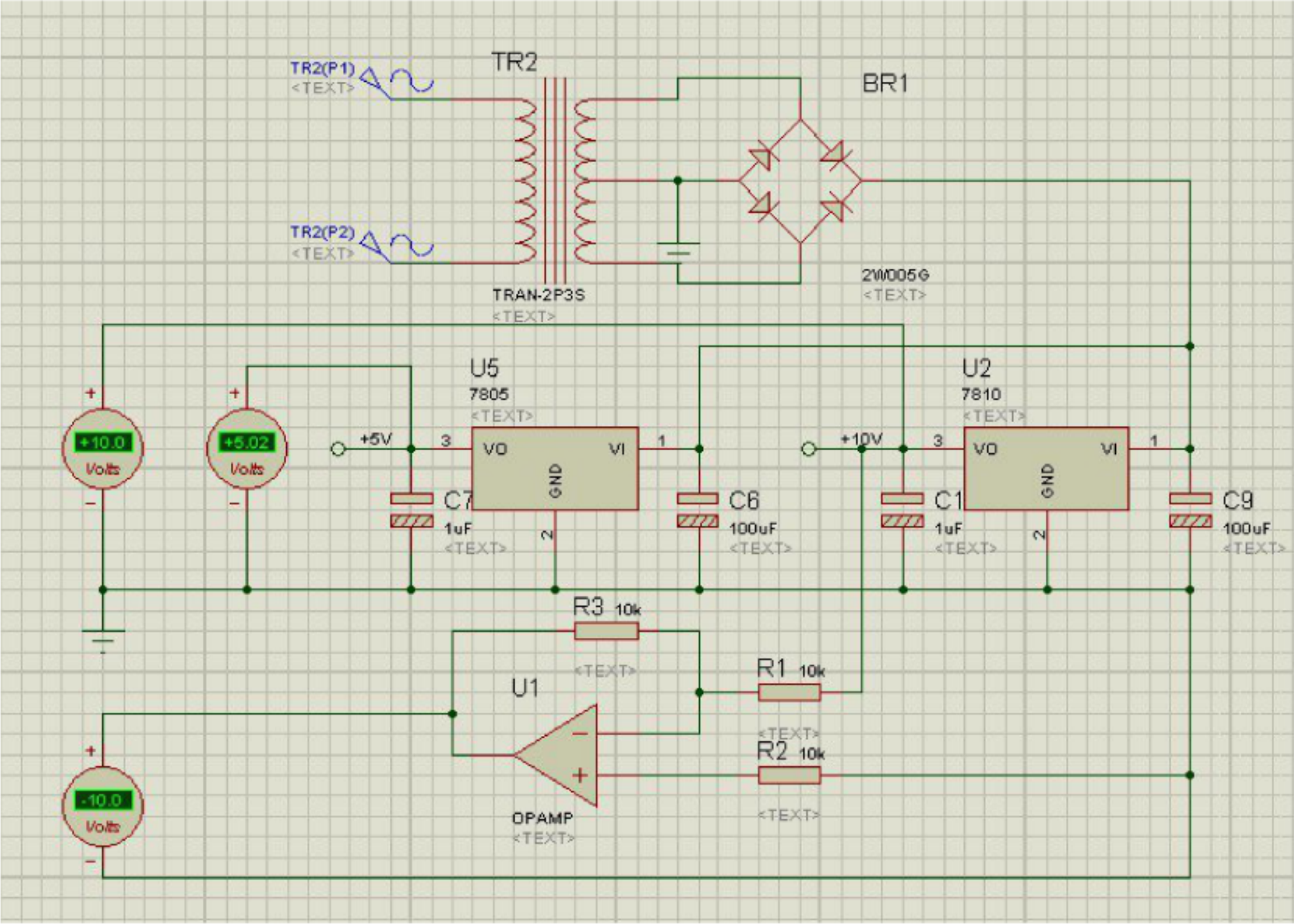
## 2.4 仿真软件的应用

在设计过程中，基于单片机的应用，我们采用 Proteus 仿真软件，并用 keil 进行 C 语言编程，从而实现数控稳压电源的设计。



# 五、主要研究结果

## 1. 自制稳压电源电路仿真图



## 2. 数控稳压电源电路部分

### 2.1 数据测量结果

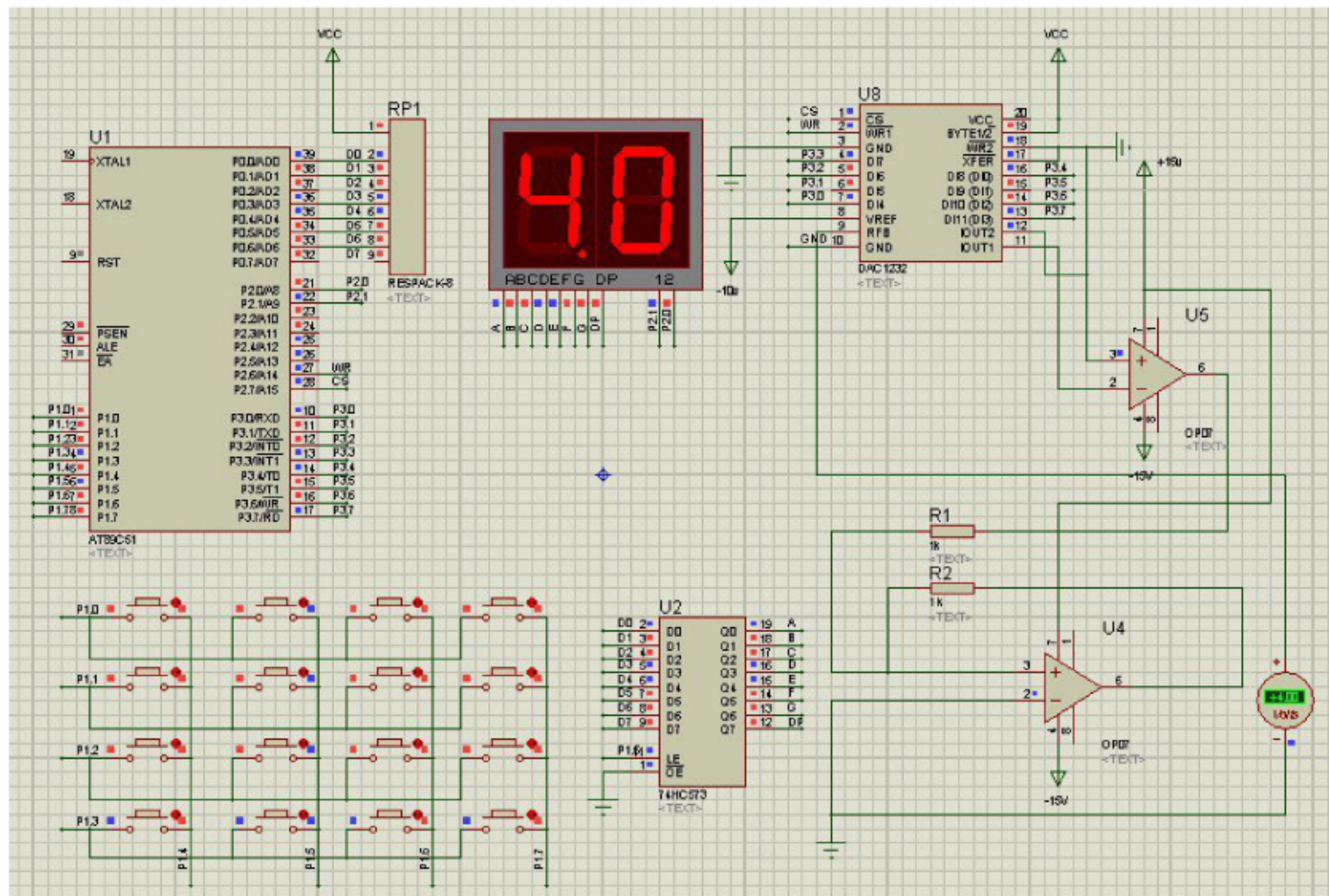
真实值	0.09	0.20	0.27	0.39	0.47	0.59	0.67	0.78	0.90	0.98	1.10
显示值	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9	1.0	1.1
误差率	0.100	0.000	0.100	0.025	0.060	0.016	0.043	0.025	0.000	0.020	0.000
真实值	1.17	1.29	1.37	1.49	1.57	1.68	1.80	1.88	2.00	2.08	2.19
显示值	1.2	1.3	1.4	1.5	1.6	1.7	1.8	1.9	2.0	2.1	2.2
误差率	0.025	0.007	0.021	0.006	0.018	0.012	0.000	0.011	0.000	0.009	0.004
真实值	2.27	2.39	2.51	2.59	2.71	2.78	2.90	2.98	3.10	3.18	3.29
显示值	2.3	2.4	2.5	2.6	2.7	2.8	2.9	3.0	3.1	3.2	3.3
误差率	0.013	0.004	0.004	0.004	0.004	0.007	0.000	0.007	0.000	0.006	0.003
真实值	3.41	3.49	3.61	3.68	3.80	3.88	4.00	4.08	4.19	4.31	4.39
显示值	3.4	3.5	3.6	3.7	3.8	3.9	4.0	4.1	4.2	4.3	4.4
误差率	0.003	0.003	0.003	0.005	0.000	0.005	0.000	0.005	0.002	0.002	0.002



真实值	4.51	4.59	4.70	4.78	4.90	5.02	5.10	5.21	5.29	5.41	5.49
显示值	4.5	4.6	4.7	4.8	4.9	5.0	5.1	5.2	5.3	5.4	5.5
误差率	0.002	0.002	0.000	0.004	0.000	0.004	0.000	0.002	0.002	0.002	0.002
真实值	5.61	5.69	5.80	5.92	6.00	6.12	6.19	6.31	6.39	6.51	6.59
显示值	5.6	5.7	5.8	5.9	6.0	6.1	6.2	6.3	6.4	6.5	6.6
误差率	0.002	0.002	0.000	0.003	0.000	0.003	0.001	0.001	0.001	0.001	0.001
真实值	6.70	6.82	6.90	7.02	7.10	7.21	7.29	7.41	7.53	7.61	7.72
显示值	6.7	6.8	6.9	7.0	7.1	7.2	7.3	7.4	7.5	7.6	7.7
误差率	0.000	0.003	0.000	0.003	0.000	0.001	0.001	0.001	0.004	0.001	0.002
真实值	7.80	7.92	8.00	8.12	8.20	8.31	8.43	8.51	8.63	8.70	8.82
显示值	7.8	7.9	8.0	8.1	8.2	8.3	8.4	8.5	8.6	8.7	8.8
误差率	0.000	0.002	0.000	0.002	0.000	0.001	0.003	0.001	0.003	0.000	0.002
真实值	8.90	9.02	9.10	9.21	9.33	9.41	9.53	9.61	9.72	9.80	9.92
显示值	8.9	9.0	9.1	9.2	9.3	9.4	9.5	9.6	9.7	9.8	9.9
误差率	0.000	0.002	0.000	0.001	0.003	0.001	0.003	0.001	0.002	0.000	0.002

备注：误差率 = （显示值 - 真实值） / 真实值

## 2.2 电路仿真图



备注：4\*4 矩阵键盘代表内容如下表 2.1 所示



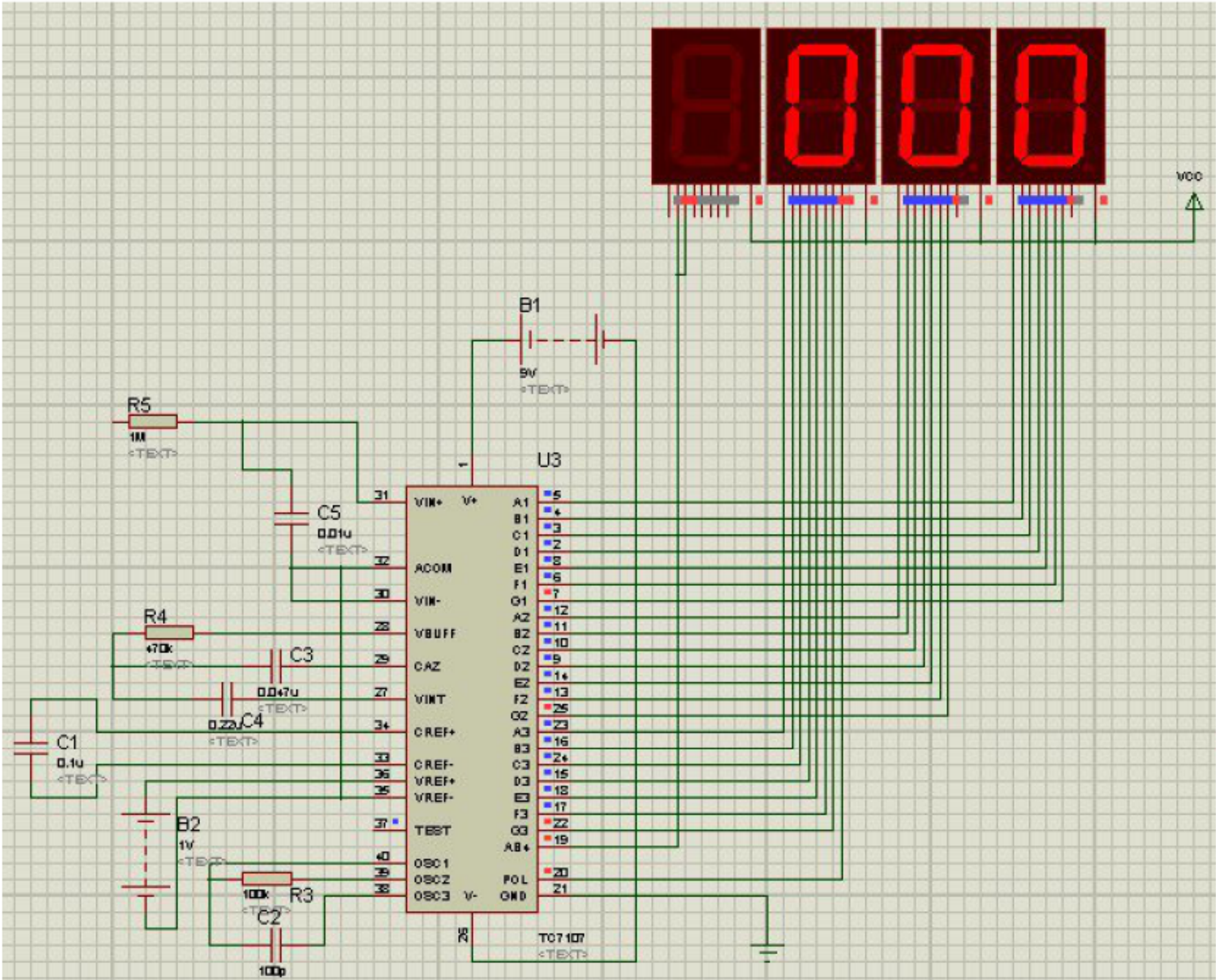
表 2.1 4\*4 矩阵键盘代表内容

7	8	9	C1c
4	5	6	-
1	2	3	+
0	Enter	+1	-1

使用说明：按下第一个数字代表小数点前一位数值，按下第二个数字表示小数点后一位数值，Enter 可以预置某一个电压值，C1c 表示清零，± 表示增加 0.1V 或减去 0.1V，±1 代表增加 1V 或者减去 1V。

3. 自制数字电压表测量电压

在测量电压时，结合设计要求，设计了如下数字电压表

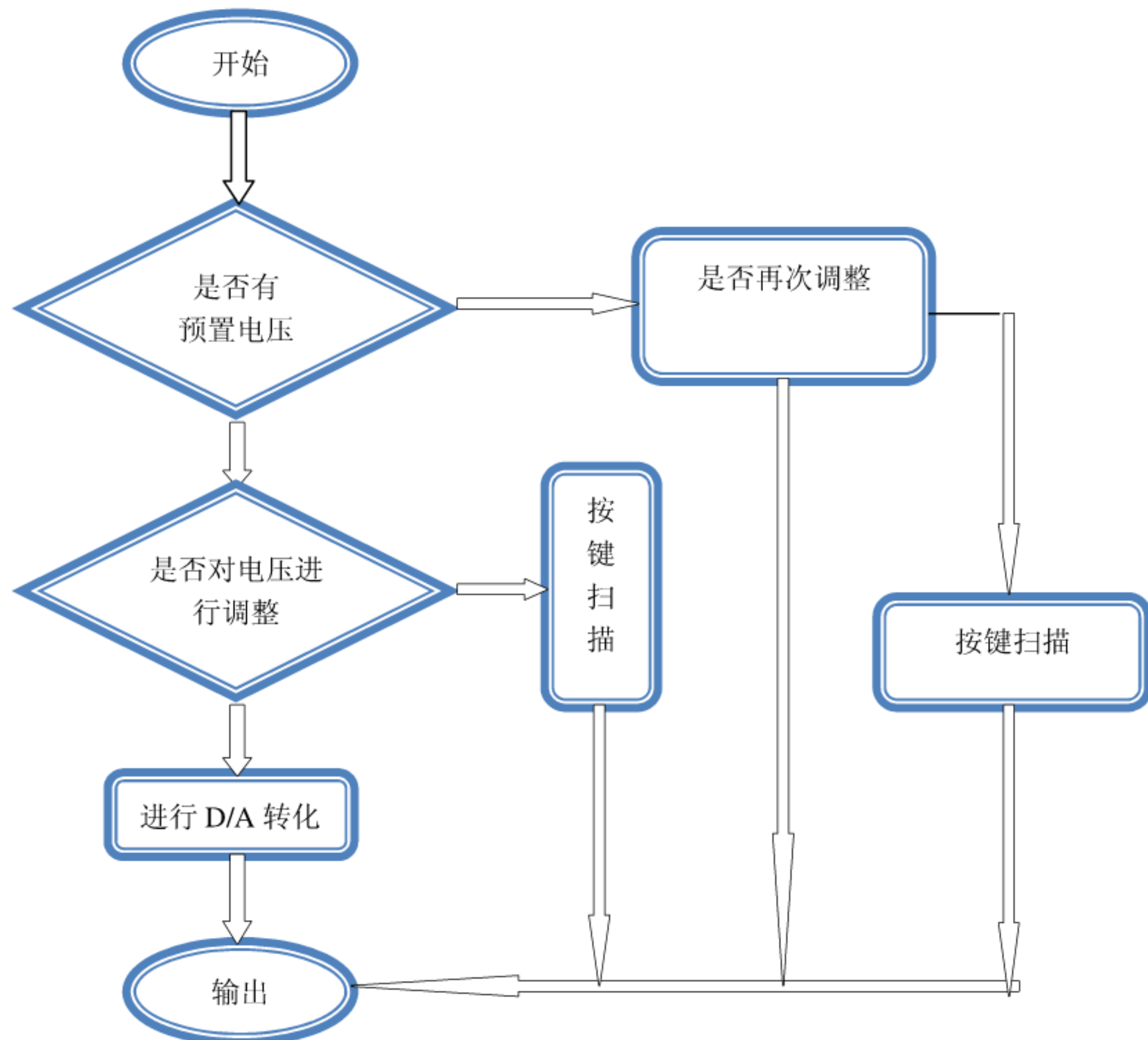




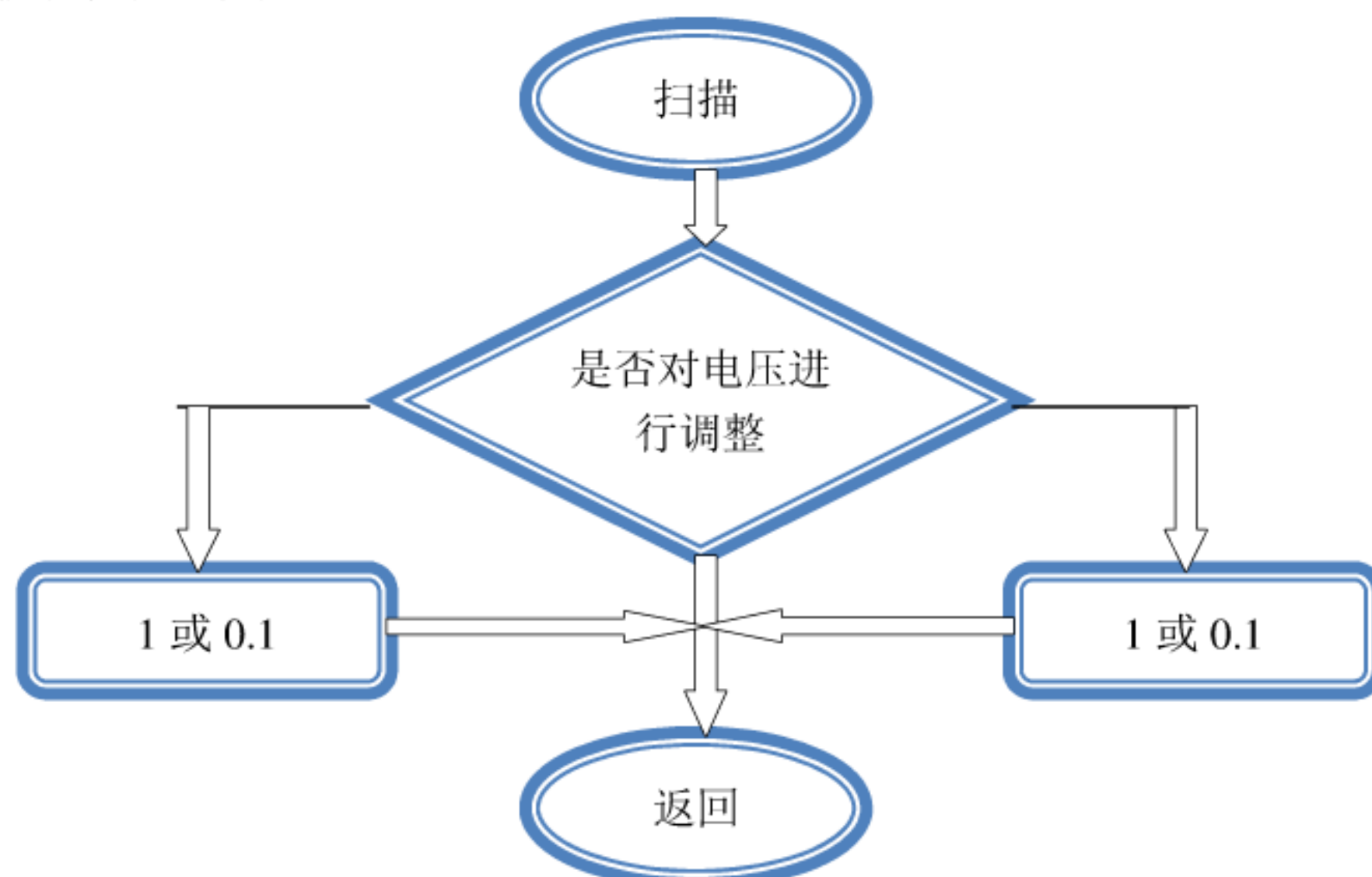
## 2. 单片机控制系统

### 2.1 程序流程图

#### 2.1.1 主流程图



#### 2.1.2 按键扫描流程图





## 2.2 程序代码

```
#include<reg51.h>
#include<intrins.h>
#define uint unsigned int
#define uchar unsigned char
uchar code table[]={0x3f,0x06,0x5b,0x4f,0x66,0x6d,0x7d,0x07,0x7f,0x6f,0x77};
uchar data tab[2];
uchar ws[2]={0,0};
uchar code table2[]={0xbf,0x86,0xdb,0xcf,0xe6,0xed,0xfd,0x87,0xff,0xef,0xf7};
uchar code tablewei[]={0x0e,0x0d};

sbit dula=P1^5;
sbit dawr=P2^6;//DA 写数据段
sbit dacs=P2^7;//DA 始能段

uchar num1,f,f1,f2,f3,f4,f5,f6,keynum;
//f 为数字键标志位;f1 为 DA 转换标志位;f2,f4 分别为+,-0.1 标志位;f3 为清零标志位
void delay(uint z)//延时子程序
{
    uint x,y;
    for(x=z;x>0;x--)
        for(y=110;y>0;y--);
}
void display(uint number)//显示 LED
{
    uchar i,j;
    tab[0]=number%10;
    tab[1]=number/10;
    for(i=0,j=0;i<2;i++,j++)
    {
        dula=1;
        P2=tablewei[j];
        if(i==0)
            P0=table[tab[i]];
        else
            P0=table2[tab[i]];
        dula=0;
        delay(5);
    }
}
keyscan();//键盘扫描
```



```

{
    uchar num,tt;
    P1=0xfe;
    tt=P1;
    tt=tt&0xf0;
    while(tt!=0xf0) // 检查按键是否被摁下
    {
        delay(50);
        tt=P1;
        tt=tt&0xf0;
        while(tt!=0xf0) // 检查按键是否被摁下
        {
            tt=P1;
            switch(tt)
            {
                case 0xee: num=7,f=1;
                break;
                case 0xde: num=8,f=1;
                break;
                case 0xbe: num=9,f=1;
                break;
                case 0x7e: num='0';f3=1;//清零按键
                break;
                default:break;
            }
            while(tt!=0xf0)
            {
                tt=P1;
                tt=tt&0xf0;
            }
        }
    }

    P1=0xfd;
    tt=P1;
    tt=tt&0xf0;
    while(tt!=0xf0) // 检查按键是否被摁下
    {
        delay(50);
        tt=P1;
        tt=tt&0xf0;
        while(tt!=0xf0) // 检查按键是否被摁下
        {

```



```

        tt=P1;
        switch(tt)
        {
            case 0xed: num=4;f=1;
            break;
            case 0xdd: num=5;f=1;
            break;
            case 0xbd: num=6;f=1;
            break;
            case 0x7d: num=14;f4=1;//减 0.1 按键
            break;
            default:break;
        }
        while(tt!=0xf0)
        {
            tt=P1;
            tt=tt&0xf0;
        }
    }

    P1=0xfb;
    tt=P1;
    tt=tt&0xf0;
    while(tt!=0xf0) // 检查按键是否被摁下
    {
        delay(50);
        tt=P1;
        tt=tt&0xf0;
        while(tt!=0xf0) // 检查按键是否被摁下
        {
            tt=P1;
            switch(tt)
            {
                case 0xeb: num=1;f=1;
                break;
                case 0xdb: num=2;f=1;
                break;
                case 0xbb: num=3;f=1;
                break;
                case 0x7b: num='+';f2=1;//加 0.1 按键
                break;
                default:break;
            }
        }
    }
}

```



```

        }
        while(tt!=0xf0)
        {
            tt=P1;
            tt=tt&0xf0;
        }
    }

    P1=0xf7;
    tt=P1;
    tt=tt&0xf0;
    while(tt!=0xf0) // 检查按键是否被摁下
    {
        delay(50);
        tt=P1;
        tt=tt&0xf0;
        while(tt!=0xf0) // 检查按键是否被摁下
        {
            tt=P1;
            switch(tt)
            {
                case 0xe7: num=0;f=1;
                break;
                case 0xd7: num=10;f1=1;//确认 DA 转换按键
                break;
                case 0xb7: num=11;f5=1;
                break;
                case 0x77: num=12;f6=1;
                break;
                default:break;
            }
        }
        while(tt!=0xf0)
        {
            tt=P1;
            tt=tt&0xf0;
        }
    }

}
return(num);

```

```
}
```

```
void jishu()//预置电压值
```

```
{
```

```
    keynum=keyscan();
```

```
    if(f==1)
```

```
    {
```

```
        f=0;
```

```
        num1=num1*10+keynum;
```

```
    }
```

```
}
```

```
void change()//DA 转换
```

```
{
```

```
    if(f1==1)
```

```
    {
```

```
        P3=(uchar)(256*num1/100);
```

```
        delay(100);
```

```
    }
```

```
}
```

```
void up_down_clear()//加減 0.1 和清零
```

```
{
```

```
    if(f2==1)
```

```
    {
```

```
        f2=0;
```

```
        num1++;
```

```
    }
```

```
    if(f4==1)
```

```
    {
```

```
        f4=0;
```

```
        num1--;
```

```
    }
```

```
    if(f3==1)
```

```
    {
```

```
        f3=0;
```

```
        num1=0;
```

```
    }
```

```
}
```

```
void up_down_1()//加減 1
```

```
{
```

```
    if(f5==1)
```

```
    {
```



```

        f5=0;
        num1=num1+10;
    }
    if(f6==1)
    {
        f6=0;
        num1=num1-10;
    }
}
void main()
{
    dawr=0;
    dacs=0;
    P3=0;
    while(1)
    {

        jishu();
        up_down_clear();
        up_down_1();
        display(num1);
        change();

    }
}

```

## 六、存在的问题及今后努力方向

数控电源是一个原理和操作都比较简单的电子系统，数控电源的输出电压调整范围为 0.0V 到 9.9V，在这个范围内，以 0.1V 的步进值，可以提供任一想要得到的电压，为负载供电，并且操作简单，使用方便。

本次设计在完成了题目的要求中仍有不足之处，其中主要的是没有达到高精度的设计要求。其主要原因是设计条件不够，在选择元器件上没有能够很好的选择。在设计的过程中，遇到过一些问题，给我们的感觉很不顺手，看似很简单的电路，要动手把它给设计出来，是很难的一件事，主要原因是我们没有经常动手设计过电路以及在设计过程中用到的知识我学得不是很扎实，还有资料的查找也是一大难题，对单片机的功能没有完全掌握，程序编写不够熟练，这就要求我们在以后的学习和工作中，应该注意到这一点，更重要的是我们要学会把从书本中学到的知识和实际的电路联系起来，这不论是对我们以后的就业还是学习，都会起到很大的促进和帮助。不过通过我们的努力，或到实验室进行试验和查阅各种资料，最后都得到了解决。通过这个设计，让我们将所学过的知识综合起来考虑、分析，使我对专业知识的掌握更加深入，应用起来也更加得心应手。同时，通过本次电子设计，巩固了我们学习过的专业知识，也使我们把理论与实践从真正意义上相结合了起来；考验了我们借助互联网络搜集、查阅相关文献资料，和组织材料的综合能力；从中可以自我测验，认识到自己哪方面有欠缺、不足，以便于在日后的工作中得以改进、提高。

评审组意见：

评审组组长：

年      月      日