基于单片机的数控直流稳压电源

一、引言

（1）题目要求：

利用 LM317三端稳压器，设计制作一个数控稳压电源，要求：

1、输出电压： 2-15V，步进 0.1V，纹波≤ 10mV;

2、输出电流 0.5A；

3、输出电压值由数码管显示，由“ + ”、“ - ”键分别控制输出电压的步进

（2） 概况：

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．但均存在以下问题： 输出电压是通过粗调 （波段开关)及细调 (电 位器)来调节。 这样，当输出电压需要精确输出，或需要在一个小范围内改变时（如 1.02~1．03V)，困难就较大。 另外，随着使用时间的增加， 波段开关及电位器难免接触不良， 对输出会有影响。 常常通过硬件对过载进行限流或截流型保护， 电路构成复杂， 稳压精度也 不高。本文设计了一种以单片机为核心的智能化高精度简易直流电源，克服了传统直流电压 源的缺点，具有很高的应用价值。

二、系统设计

（1） 方案论证：

方案：采用单片机控制此方案采用 AT89C51单片机作为整机的控制单元，通过改变输入数字量来改变输出电压值。 这里主要利用单片机程控输出数字信号，经过 D /A 转换器 (DA0832) 输出模拟量，然后使用运算放大器把电

流转换成电压， 在通过三段稳压器 LM317使得输出电压和输出电流达到稳压的目的。

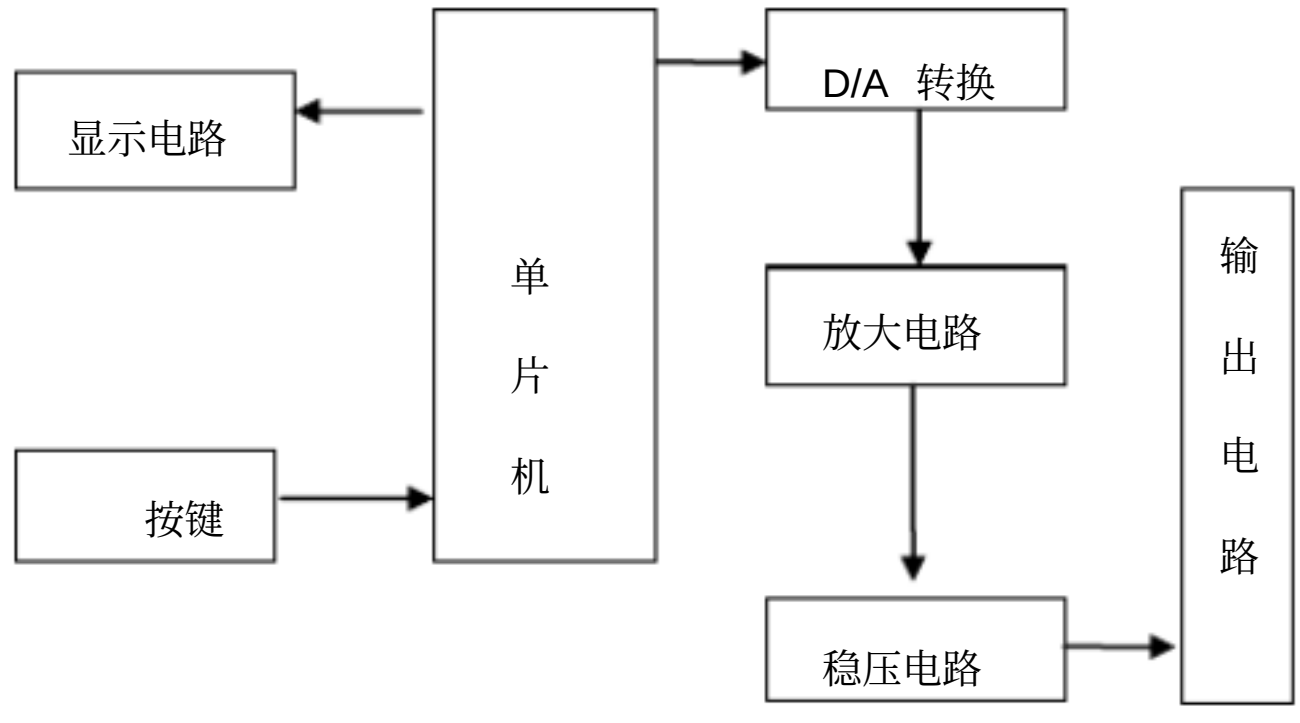
方案论证：

1、输出模块： 使用运算放大器做前级的运算放大器，由于运算放大器具有很大的电源电压抑制比，可以减少输出端的纹波电压。使用 LM317做电流稳压器，把电流稳定到 0.5A。

2、数控模块： 采用 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。

3、显示模块： 本来准备使用液晶显示，可是想想我们的层次不够，液晶现实的额程序不会写，只能退而其次，选择使用单片机通过锁存器控制 8 段 LED数码管直接显示，这样可以精确的显示输出电压。

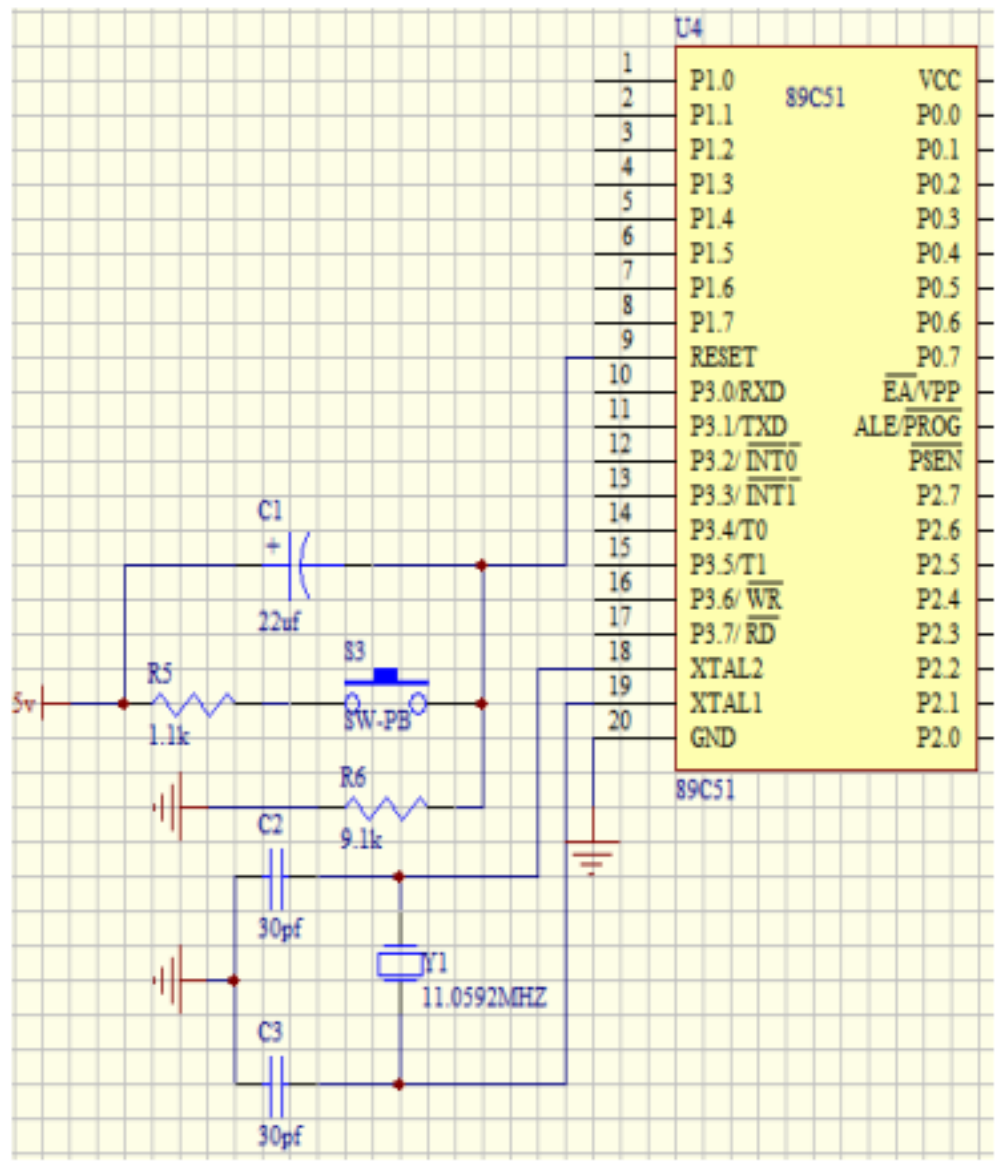
（2） 系统结构：



系统结构设计图如上图所示。 该系统主要由单片机最小控制系统、 显示电路、独立按键、 D/A 转换电路、放大电路和稳压电路组成。单片机设定预输出值，并可以通过独立键盘改变单片机的预设值。然后通过 DAC0832 转化为模拟量，再经过运算放大和稳压稳流电路最后输出预设电压值，通过 LED 显示能够直观的看到预设值。 因为器材原因， 我们设计的稳压电源采用的是外部稳压器提供的电源。这样虽然算不上是一个完整的数控直流稳压电源，但是，除了这点，我们设计的电源基本已经复合要求。

（3） 硬件设计：

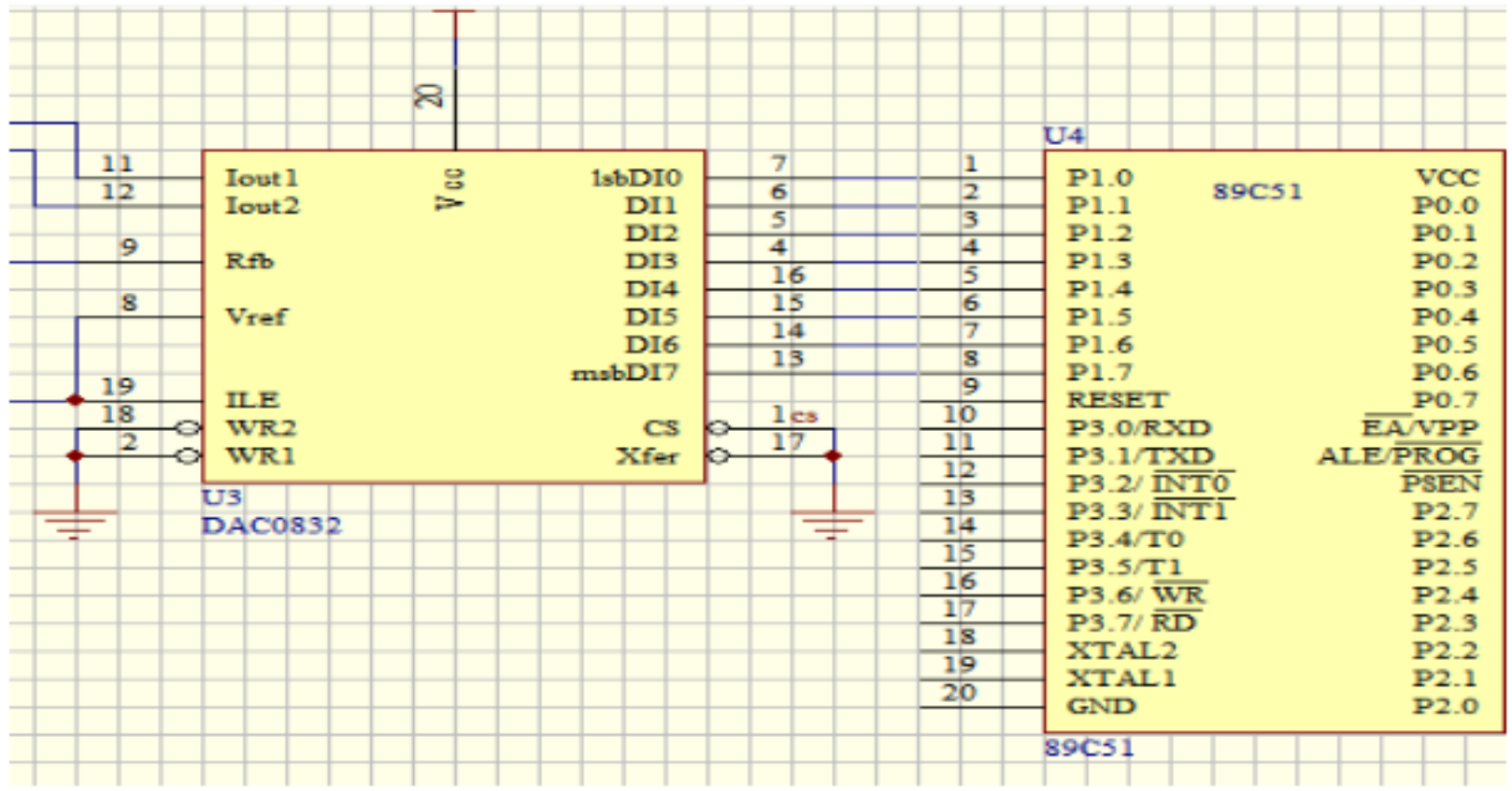
1 、最小系统控制电路设计： 最小控制系统由 STC单片机、晶振、独立键盘和复位电路等组成。如下图所示。



AT89C51 的管脚排列如上图所示， 9 管脚接复位电路， 18、19 管脚为晶振的两个输入端， 20 管脚接地， 40 管脚接 +5V。

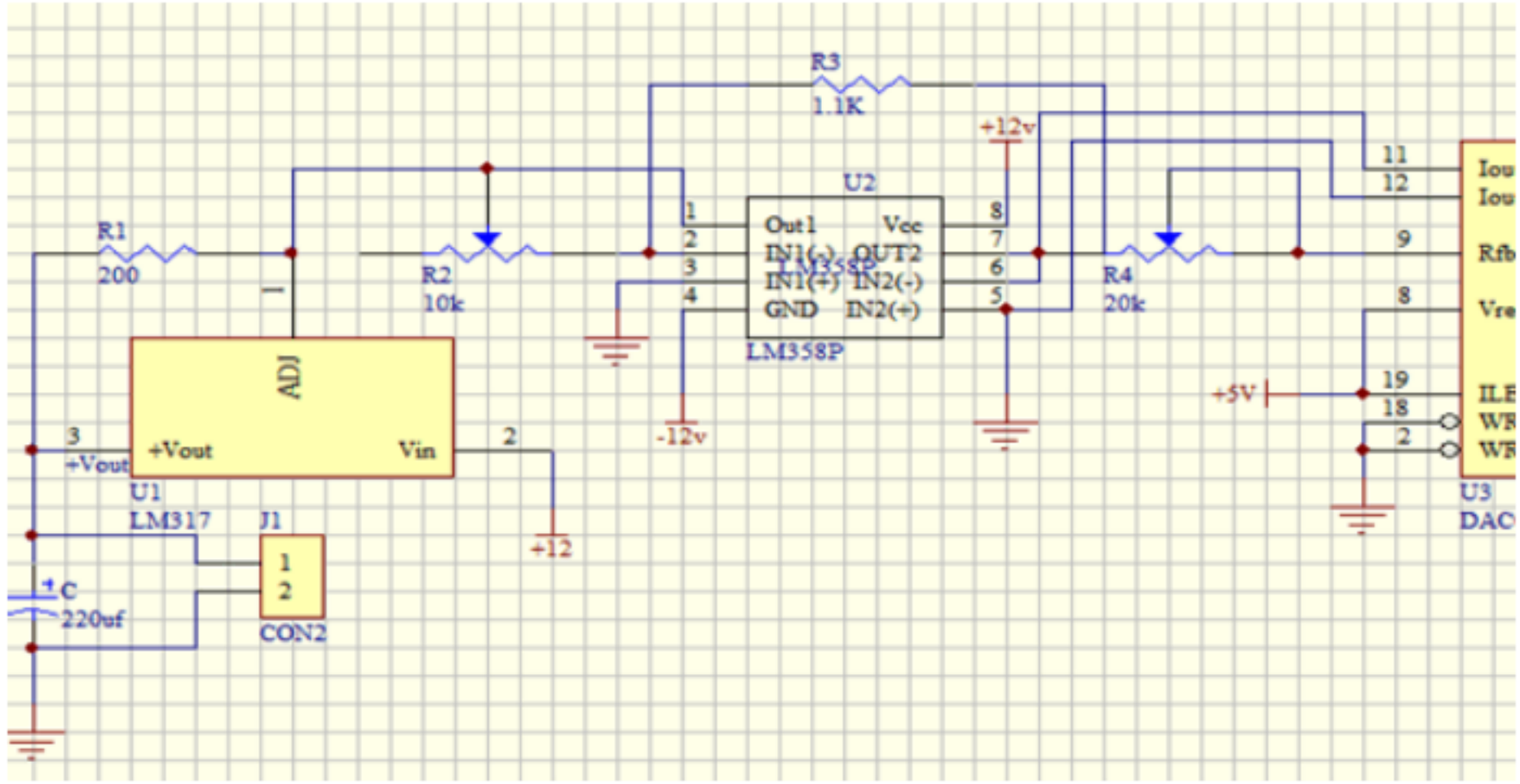
晶振 Y1 和两个电容 C2、C3 构成自激震荡，连接到单片机的 X1 和 X2 端，电解电容 C4、电阻 R5 和按键 S5 构成复位电路，连接到单片机的复位端。当按键 S5 按下后，复位端通过 R5 与+5V 电源接通，电容迅速放电，使 RST 管脚为高电平；当复位按键 S5 弹起后， +5V 电源通过 R6 对电容 C4 重新充电， RST 管脚出现复位正脉冲。

2、 D/A 转换电路设计：



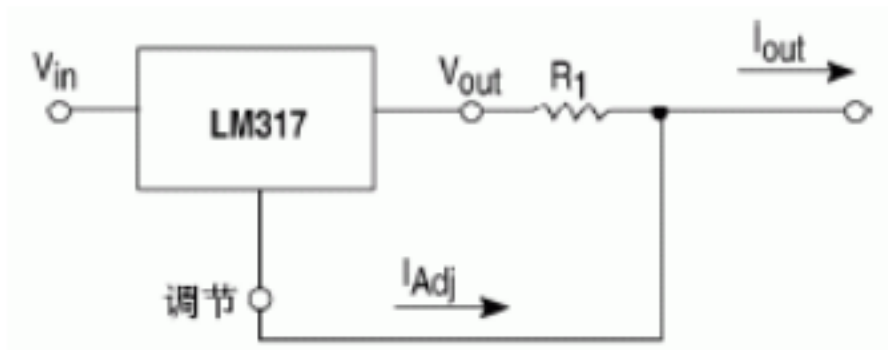
如上图所示，DA0832的 8 位数据线 D0~D7与单片机的 P1 口连接，1 管脚（CS）和 17 管脚（Xfer ）接地， 8 管脚(Vref) 的参考电压为 5V，则 LSB=5V/2^8=0.02V，即最小分表率为 0.02V。11 管脚（ Iout1 ）和 12 管脚（ Iout2 ）为电流输出端。

3、放大电路与稳压稳流电路设计：

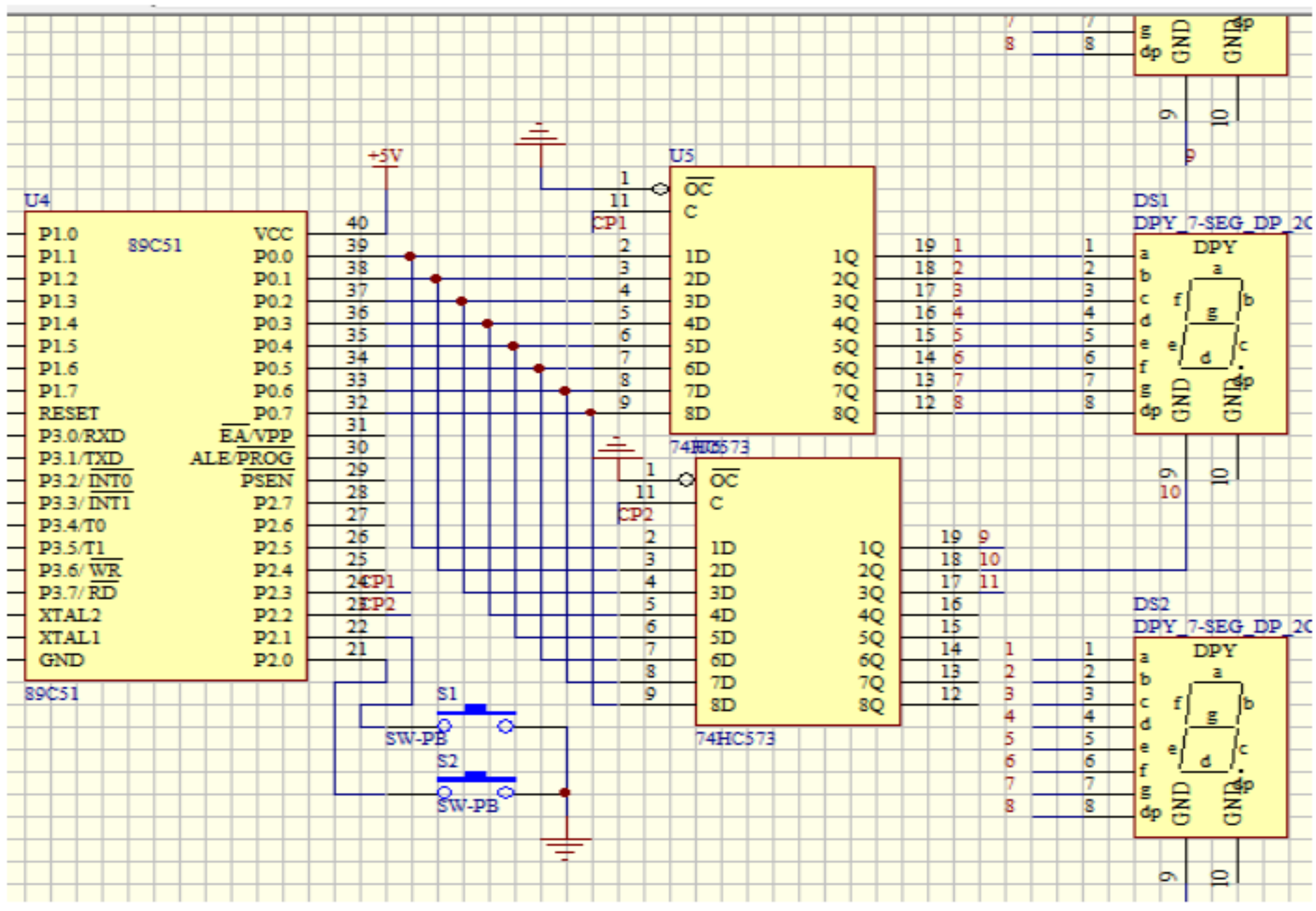


如上图所示， 本设计中将 DAC0832的 Iout2 接地，采用 Iout1 输出，然后接运算放大 LM358P将输出电流转化为电压。 经过 LM358P转化后的电压值也为 5V。为了达到与单片机预设电压范围 2~15V同步，输出端电压需要经过二级放大。 第一级不放大，直接将 D/A 输出的电流转化为电压，第二级放大，放大倍数 n =R2/R1=5.5K/1.1K=5。这里的 R2由于找不到 5.5K 的电阻，所以用电位器代替。

因为 DA0832转换后的电压的范围为 0~5V, 即 DA0832的 8 位输入端全为高电平 1 时，输出电压为 5V，输入端全为低电平 0 时，输出电压为 0V，且呈线性变化。为此为了使输出与 LED显示同步，必须经过放大倍数 n =5 的二级放大。再经过运放放大后的电压已经复合要求， 可是电流却没有复合要求， 这就要用到了三段稳压器 LM317。在这里， LM317 作为电流稳压器，其应用电路如下图所示，其中 $I_{out} = \left(\frac{V_{ref}}{R_1}\right) + I_{Adj} = \frac{1.25V}{R_1}$，所以 R1 的值应该为 2.5 Ω。可是，我们在实验室能找到的最小电阻是 200Ω，这还是远远大于 2.5 Ω。所以我们的输出电流才 6ma。这里还要说的是，本来我们采用的运算放大器是 Lm324，可是，因为我的不小心，在测试运放放大的时候，把芯片烧坏了。并且我们手头没有多余的芯片，幸亏和我们做同一方案的同学有运放 Lm358，所以我们也采用了 Lm358。



4、显示模块设计：

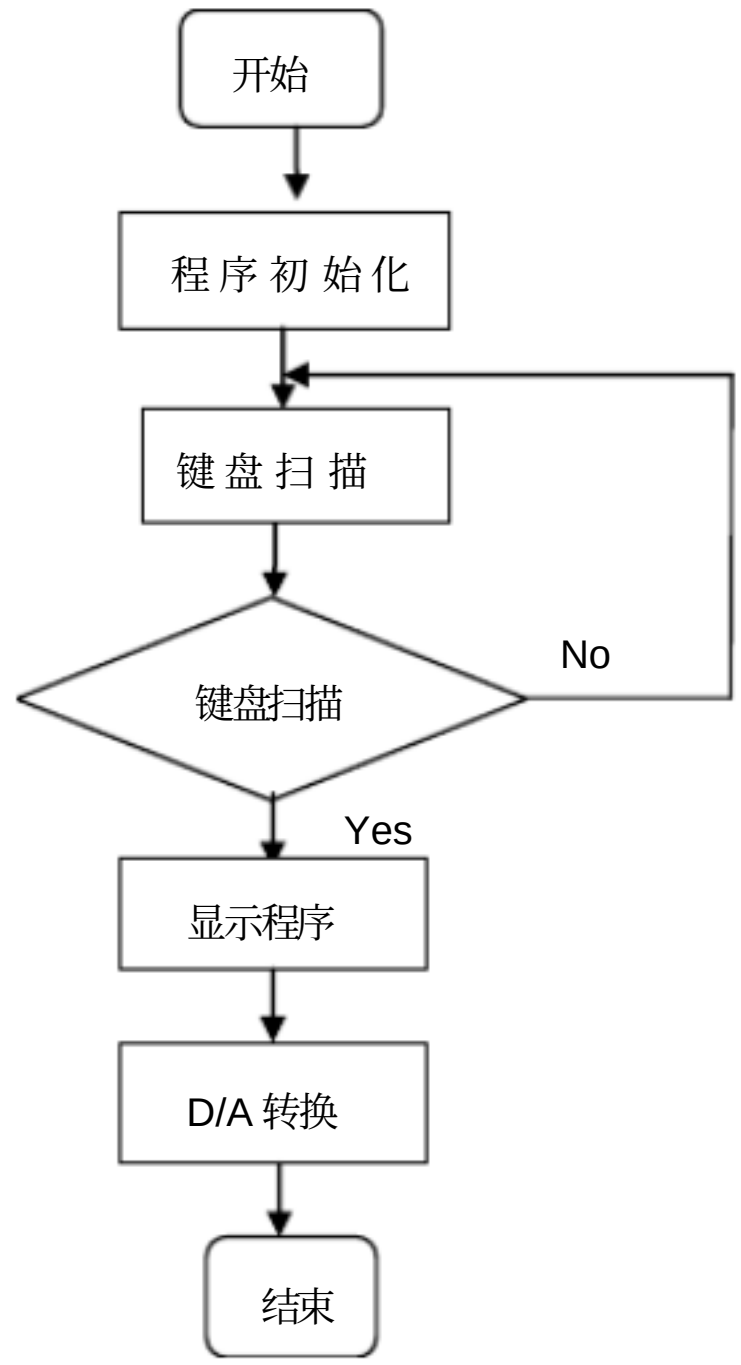


如上图所示，显示部分采用数码管　sr410561k，锁存器 74HC573 数码管段码 A~DP 接锁存器 1 的 Q0~Q7，数码管的位选　1~3 接锁存器 2 的 Q0~Q3。P0 接锁存器 1、2 的 D0~D7。锁存器 1 的 LE 接单片机 P2^2，锁存器 2 的 LE 接单片机 P2^3。数码管的 a~dp 接锁存器 1 的 Q0~Q7，数码管的位选　1~3 分别接锁存器 2 的 Q0~Q3。在使用数码管的过程中，我们发现数码管的位选直接接到单片机的 P2 口上，会使数码管的亮度不够。现在我们有　2 种方法解决。第一，接上拉电阻，经计算，　200Ω 左右的电阻可使数码管达到最亮，为了保险起见，可以使用 400Ω 的电阻。但当时我们手头刚好没有　400Ω 的电阻，所以我们采用了第二种方法，把数码管的位选接锁存器上。

（4）　软件设计：

程序流程图设计：

程序设计流程图下图所示。　程序开始以后，首先程序初始化，显示 LED 预设的初始电压值。然后进行按键检测，如果没有按键按下，　LED 显示的电压不变；如果有按键按下，确认当前　LED 的调整值。接着启动　D/A 转换，将转换后的模拟量送给系统最终输出端。



程序代码： 在附录

（5） 系统调试：

显示模块调试 ：算出数码管的段码， 位选，使数码管能正确的显示预设值。

按键模块调试：消除抖动，使我们按一下按键的加、减键时，能实现显示程序的步进 0.1 。

放大稳压电路调试：为了使输出电压和显示模块对应，我们要调节放大电路的方法倍数。假使显示的电压为 11.3v ，那么因为三端稳压器的自带电压为 1.25v ，所以放大电路输出电压因为 11.3-12.5 ≈10v，所以一级放大的输出电

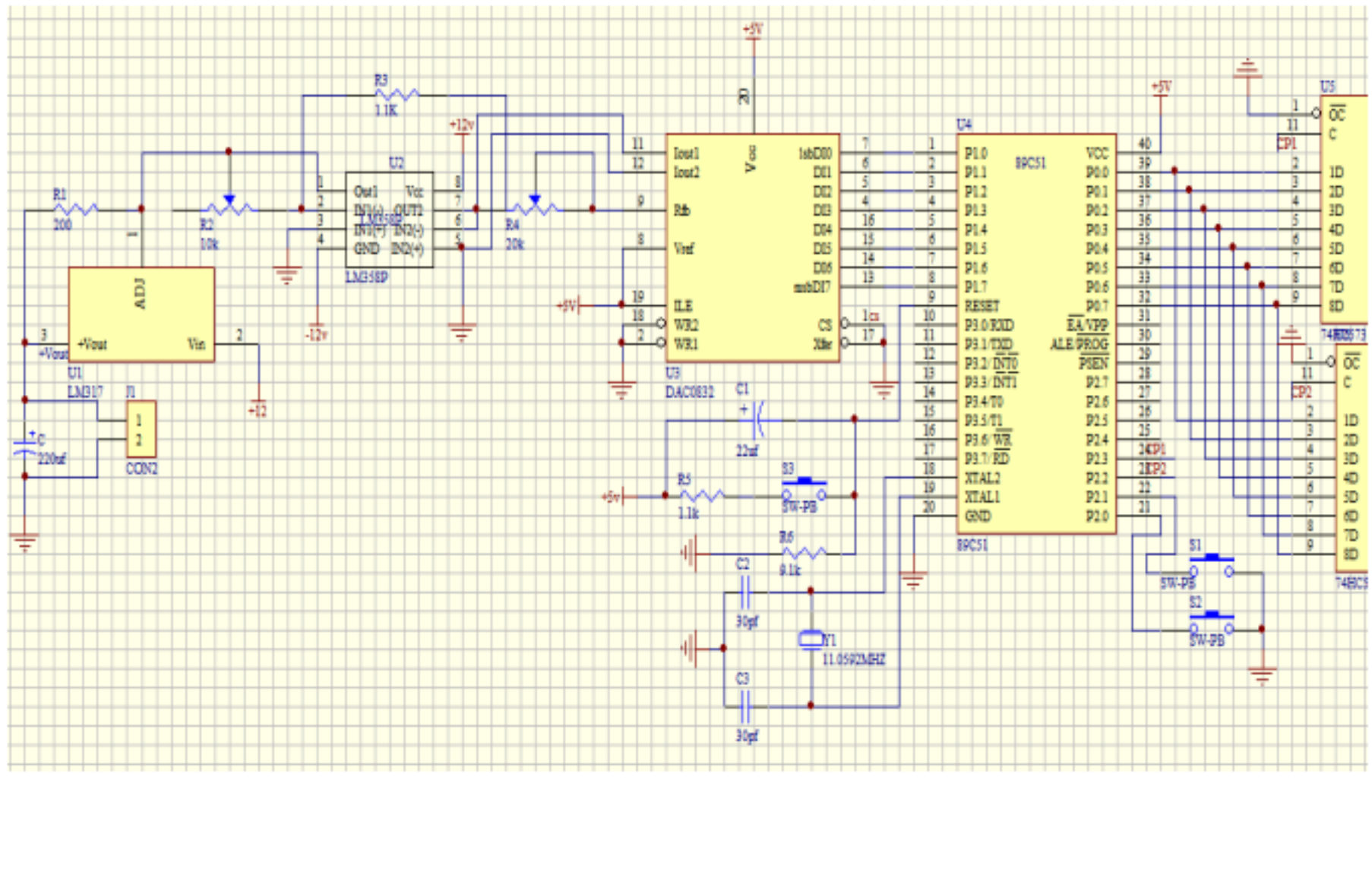
压应为 -2v，二级放大的电压应为 10v。稳流方面，因为器材的原因，我们只能把电流稳定在 6ma。

（6） 系统测试：

各个模块连接起来后， 因为电路的改变， 可能会改变输出值的大小， 所以我们要进行整体的测试：先测试放大电路的第一级放大，然后调整 LM358P和DAC0832连接的那个电位器，使输出电压再次达到预想值。再调整第二级放大，把放大倍数再次调为 5 倍。 把程序下载到硬件电路， 测试最后输出值， 是否为我们的预想值

三、总结

附录：



程序代码：

```
#include<reg51.h>

#define uint unsigned int

#define uchar unsigned char

#define DAC0832_PORT P1
```

```
sbit duanxuan=P2^6;

sbit weixuan=P2^5;

sbit cs=P2^2;

sbit wr1=P2^3;

sbit S1=P2^0;// 加

sbit S2=P2^1;// 减

uchar num=20;

uchar code table[]={0x03,0x9f,0x25,0x0d,0x99,0x49,0x41,0x1f,0x01,0x09};

void delay(uint z) //  延时  z ms 子程序

{

        uint x,y;

        for(x=z;x>0;x--)

                for(y=110;y>0;y--);

}



init()// 初始化子函数

{

        P1=num;

}



uchar keyscan()// 键盘扫描程序

{

        if(S1==0)

        {

        delay(10);          //键盘按键消抖

        if(S1==0)

        {

                if(num==150)

                {

                        num=20;
```

```
        }
        else
        {
            num++;
        }
    }
    while(!S1);              //松手检测
  }

  if(S2==0)
 {
        delay(10);
        if(S2==0)
        {
            if(num==20)
            {
                num=150;
            }
            else
            {
                num--;
            }
    }
    while(!S2);              //松手检测
            }
            return(num);
}

void display()//  显示程序
{
```

```
duanxuan=1;
P0=table[num/100];// 十位
duanxuan=0;
weixuan=1;
P0=0x80;
weixuan=0;
delay(1);



duanxuan=1;
P0=((table[num%100/10])&0xfe);//   个位
duanxuan=0;
weixuan=1;
P0=0x40;
weixuan=0;
delay(1);


duanxuan=1;
P0=table[num%10];// 小数
duanxuan=0;
weixuan=1;
P0=0x20;
weixuan=0;
        delay(1);


}
uchar dazh( uchar n)//D/A  转换子程序
{       cs=0;选定芯片
        wr1=0; 允许写入
```

```
        n=num-13; 输出电压值
        DAC0832_PORT=n; // 把 n 送给給 DA

}

// 主程序 //
void    main()
{
        init();
        while(1)
        {       keyscan();
                display();
                dazh();

        }
}
```