专业课程设计报告

题 目： 直流电机的调速与控制

课 程： 单片机技术及Proteus仿真

学生姓名：

学生学号：

年 级：

专 业：

班 级：

指导教师：

机械与电气工程学院制

2018年3月

**目录**

[1.设计任务及要求分析 1](#_Toc360719960)

[2.控制方案设计 2](#_Toc360719961)

[2.1用分立元件构成直流电机的驱动电路 2](#_Toc360719962)

[2.2用电机驱动芯片来构成直流电机的驱动电路 3](#_Toc360719965)

[2.3.方案选择 3](#_Toc360719966)

[2.4.总体方案设计 3](#_Toc360719967)

[3.硬件电路设计 4](#_Toc360719968)

[3.1单片机最小工作系统 4](#_Toc360719969)

[3.2数码管显示电路 4](#_Toc360719972)

[3.3直流电机驱动电路 5](#_Toc360719975)

[3.4按键控制电路 6](#_Toc360719978)

[4.软件设计 6](#_Toc360719979)

[4.1主程序设计 6](#_Toc360719980)

[4.2定时器0控制产生PWM波形 7](#_Toc360719980)

[4.3外部中断按键扫描程序 8](#_Toc360719983)

[5.仿真分析 9](#_Toc360719986)

[5.1软件仿真 10](#_Toc360719987)

[5.2运用proteus软件联合仿真 10](#_Toc360719988)

[6.总结 14](#_Toc360719986)

**直流电机的调速与控制**

# 1.设计任务及要求分析

设计任务：

设计一个直流伺服电机电机控制系统，控制功能要求实现电机的启动、停止控制、正转、反转、加速、减速共六种功能，为了能实现人机交互，需要用到按键和数码管。

具体要求如下：K0为启动/停止控制、K1正转、K2反转；K3加速，K4减速，用3个发光二极管显示状态：正转时红灯亮，反转时黄灯亮，不转时绿灯亮。并利用4位LED数码管显示电机转速。

要求分析：

按键K1用来控制电机反转，若电机正在停止或正转状态，当按下k1时，电机开始反转；若电机处于反转状态，电机不响应按键命令。

按键K2用来控制电机正转，若电机正在停止或反转状态，当按下k2时，电机开始正转；若电机处于正转状态，电机不响应按键命令。

按键K3用来控制电机加速，当电机处于运行状态时，如果电机尚未达到最大转速，当按下按键K3时，电机会加速，每按一次，电机加速一个节拍，若需要一直加速，则需要多次按下按键K3，当电机到达最大速度时，电机将保持在最大速度运行，不再响应按键命令。

按键K4用来控制电机减速，当电机处于运行状态时，如果电机尚未达到最小转速，即停止状态，当按下按键K4时，电机会减速，每按一次，电机减速一个节拍，若需要一直减速，则需要多次按下按键K4，当电机到达最小速度时，即电机处于停止状态，电机将保持停止，不再响应按键命令。

为了知道电机是否按控制要求工作，需要知道电机的转速，这里用到了4位数码管来显示，最高位显示转动方向，正转显示“0”，反转显示“-”。后3位用来显示转速的数值大小。

此外，设置三个发光二极管，来显示电机的运行状态。当电机停止时，绿灯亮，当电机正转时，红灯亮，当电机反转时，黄灯亮。

# 2. 控制方案设计

因为直流电机的工作电压和电流与控制电路的电压和电流不匹配，所以需要设计接口电路，用来把控制电路的指令转化为电机的工作信号，这种接口电路称为直流电机的驱动电路。直流电机的驱动电路有多种不同的设计方案，这里着重介绍了用分立元件构成的驱动电路和用集成芯片构成的驱动电路。

**2.1 用分立元件构成直流电机的驱动电路**

利用晶体管，电阻等分立元件，来组成直流电机的全桥驱动电路。因为电机的工作电流较大，而一个三极管的电流放大能力有限，所以采用达林顿组成全桥（H桥）驱动电路控制电机，以满足电机工作的需要。如图1所示。图中的Q1~Q4是功率三极管，其中Q1和Q2组成一个桥臂，其中Q3和Q4组成一个桥臂。每个功率三极管旁边有一个续流二极管。当Q1和Q4打开时，直流电机的控制电流从A流向B，此时直流电机正转；当Q2和Q3打开时，直流电机的控制电流从B流向A，此时直流电机反转。这样，通过Q1~Q4的控制就可以控制直流电机的转向了。

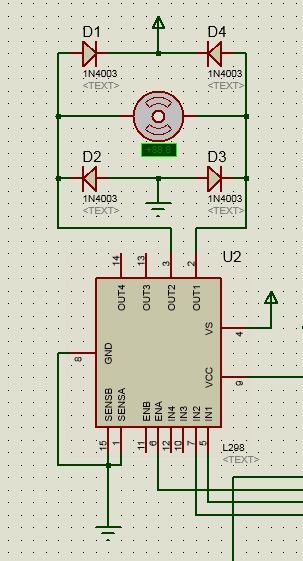


**图1 直流电机全桥控制原理图**

**2.2用电机驱动芯片来构成直流电机的驱动电路**

这里所用的驱动芯片是L298，L298起到方大电流的作用。L298有4路输入，4路输出，2个使能端。用L298构成的驱动电路如图2所示。

其工作原理为：L298芯片中，IN1对应OU1,IN2对应OUT2，IN3对应OU3,IN4对应OUT4，使能端ENA对应IN1和IN2，使能端ENB对应IN3和IN4。在使能端为高电平的状态下，当IN1输入高电平IN2输入低电平时，电机正转，当IN1输入低电平IN2输入高电平时，电机反转，当IN1和IN2输入电平相等时，电机不转。如果使能端为低电平，那么电机不受控制，而处于停止状态。



**图2 直流电机驱动电路图**

L298的工作电压范围在7V~50V之间，输出电流可达2A，能够满足直流电机的控制需要。

**2.3.方案选择**

以上所给的两种直流电机的驱动方案都能满足要求，现在来对它们进行分析比较，选出合适的方案。

用分立元件构成的直流电机驱动电路，其优点是：电路由分立元件组成，通过分析电路，能直观的了解驱动电路的工作原理，对于理论学习是非常有帮助的。其缺点是：电路过于复杂，在制作实物时容易出错，另一方面，分立元件过多，其可靠性很难保证。

用集成芯片构成的直流电机驱动电路，其优点是，电路简单，元件减少，电路的可靠性高，制作实物时相对较容易。其缺点是，对于初学者很难知道期内的工作原理，只能简单的用。

通过上面的分析，我决定选用集成芯片L298来构成直流电机的驱动电路，这样实现起来较为容易。

2.4**.总体方案设计**

控制系统的核心控制芯片是AT89C51单片机，单片机作为整个系统的大脑，协调各个部件的工作。首先，当按键发出的控制指令时，单片机利用外部中断采集控制指令，并对控制指令进行处理，之后输出信号控制直流电机的驱动电路。与此同时，单片机要计算直流电机的转速，并且要把电机的转速在数码管上显示出来。直流电机的调速是通过单片机控制输出PWM的占空比实现的，为了提高单片机的工作效率，对按键的扫描采用中断方式。其原理框图如图3：

AT89C51

直流电机

数码管显示电路

驱动电路

独立按键

输入与门

**图3 控制系统原理框图**

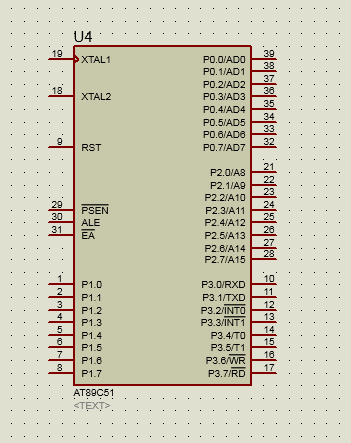
# 3.硬件电路设计

**3.1单片机最小工作系统**

单片机最小工作系统指单片机能正常工作的最少配置，包括电源、时钟电路、复位电路和一块51单片机。

**（1）89C51**

AT89C51是一种带4K字节FLASH字节存储器（FPEROM—Flash Programmable and Erasable Read Only Memory）的低电压、高性能CMOS 8位微处理器，俗称单片机。如图4所示。

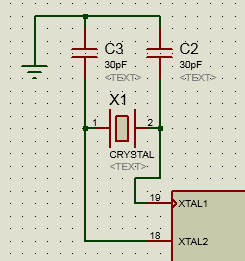


**图4 AT89C51**

**（2）时钟电路**

MCS-51单片机的时钟有两种工作方式：一种是内部振荡方式，它是利用单片机内部的振荡电路，产生时钟信号，这种方式单片机的时钟引脚上接石英晶体和振荡电容；另外一种是外部振荡方式，它是把外部已有的时钟引入单片机。下面来介绍较为常用的内部振荡方式：

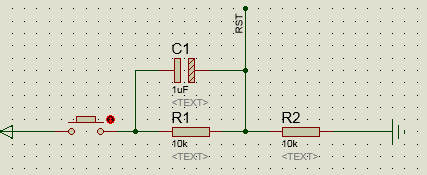
如图5，单片机内部有一个用于构成振荡器的高增益反相放大器，它的输入端为XTAL1引脚，输出端为XTAL2引脚，分别是8051单片机的19脚和18脚，其频率范围为fOSC=0~12MHZ。



**图5 内部振荡方式时钟电路图**

**（3）复位电路**

往单片机的复位引脚上输入24个时钟周期以上的高电平，即执行复位操作。在此次课设中我用的是按键复位，电路如图6所示，系统在运行时，按下复位按键，就能在RST引脚产生一段时间的高电平，使系统复位。此处，取12MHZ的晶振，电容取30pF。



**图6 按键复位电路**

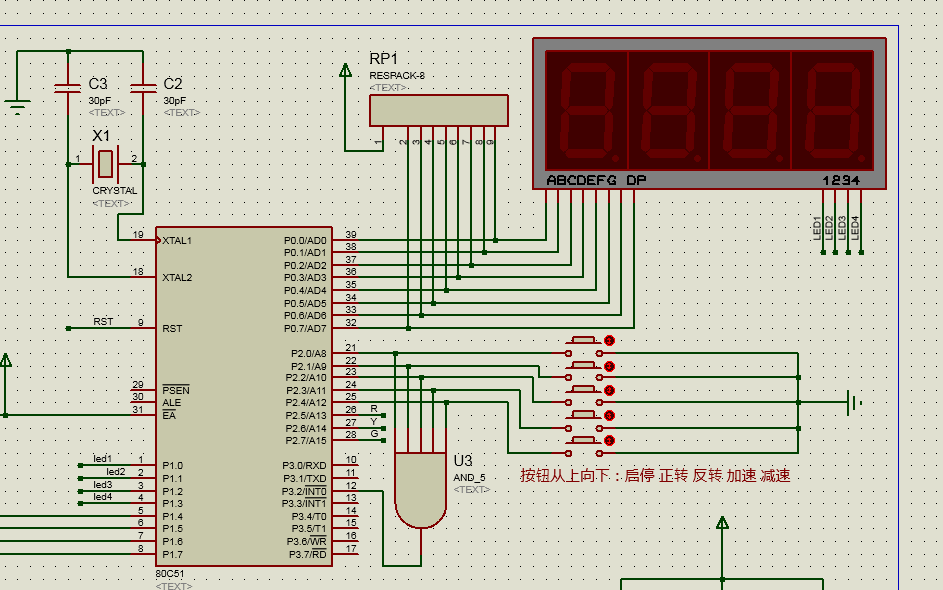
**3.2数码管显示电路**

LED数码管是由a、b、c、d、e、f、g、h这8段发光二极管组成的“8”字形显示器件。LED数码管分为共阳极和共阴极两种，如图7所示，如果每段的阳极接在一起则为共阳极LED，如果每段的阴极接在一起则为共阴极LED。此次课设中，我所用的数码管为共阴极的。



**图7 LED数码管**

这里用到了4位数码管，如图8所示，段选端接单片机的P0口，因为P0口内部没有上拉电阻，为高阻态，不能正常的输出高/低电平，所以这里需要上拉电阻。为了电路的简化，这里使用了排阻。数码管的位选端接单片机的P1.0~P1.3，当输出低电平时，表示该位数码管被选中。

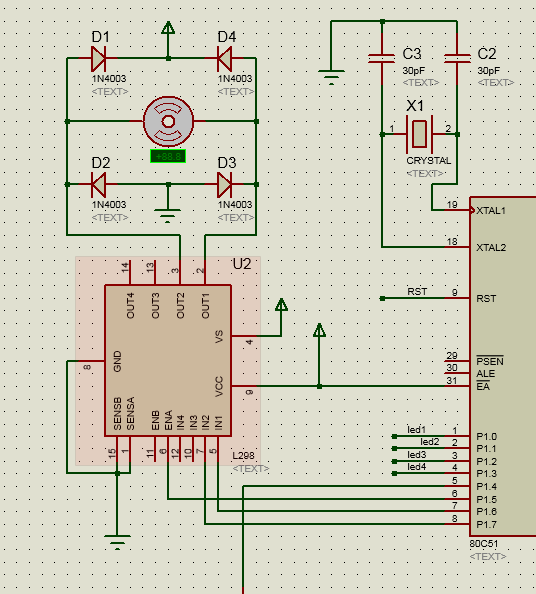


**图8 数码管电路**

**3.3直流电机驱动电路**

L298是一款高电压、高电流型的全桥驱动电路，输入为TTL电平，用来驱动如继电器、直流电机或步进电机等感性负载。两个独立的使能端，可以独立的控制两组四个输入。

直流电机的驱动电路由L298构成，51单片机通过P1.5~P1.7口控制L298的使能端口和输入端口，而驱动芯片则输出被放大的控制信号，用以控制电机运行。驱动电路如图9所示，图中所加4个二极管是起到保护作用。



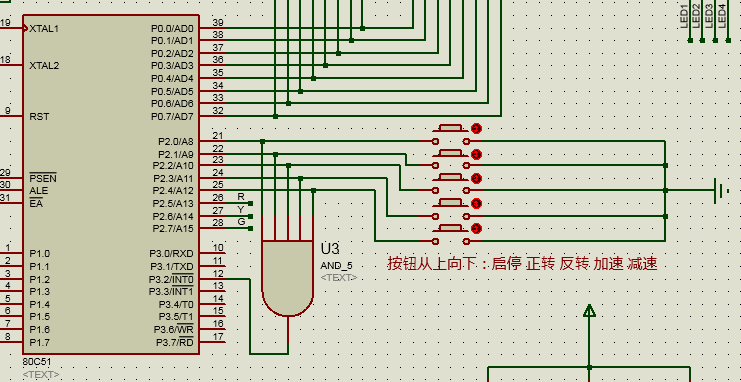
**图9 直流电机驱动电路**

**3.4按键控制电路**

单片机检测按键的原理是：单片机的I/O口既可作为输出也可以作为输入使用，当检测按键时用的是它的输入功能，我们把按键的一端接地，另一端与单片机的某个I/O口相连，开始时先给该I/O口赋一个高电平，然后让单片机不断的检测该I/O口是否变为低电平，当按键闭合时，即相当于该I/O口通过按键与地相连，变成低电平，程序一旦检测到I/O变为低电平则说明按键被按下，然后执行相应的指令。

这种按键在按下和松开时都会有抖动现象，可以采用硬件或者软件的方式来消除抖动，为了节约成本，采用软件延时的方式来消除抖动的影响。

为了提高单片机的工作效率，本次设计将5个按键通过一个与门连至单片机的外部中断输入口。当有按键按下时，单片机停下正在做的工作，转而执行按键扫描程序，然后执行相应指令。按键控制电路如图10所示：

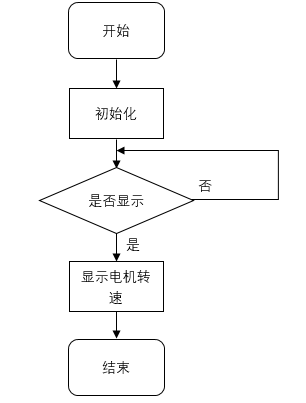


**图10 按键控制电路**

# 4.软件设计

**4.1 主程序设计**

当程序开始运行时，不断采集直流电机的转速并显示。其流程图如图11所示：



**图11 主程序流程图**

**4.2定时器0控制产生PWM波形**

直流电机调速一般运用PWM原理来进行。对于直流电机来说，如果加在电枢两端的电压为脉动电压，可以看出，在T不变的情况下，改变*t*1和*t*2的宽度，电压将发生变化。



则



式中，是常数。

利用PWM调速的原理，也就是通过改变占空比来改变加在电机两端的电压，从而改变电机的转速，其流程图如图12所示。



**图12 PWM波形控制流程图**

直流电机在进行调速时，通过改变变量num的值实现。加速时，num的值增加，而TH0的初值则相应的减小，也就是一个周期中，高电平相应的时间延长，低电平的时间相应减小速度增加。减速时num的值减小，也就是TH0的初值增加，一个周期中，高电平的时间减小，低电平的时间增加，速度降低。

**4.3按键扫描程序**

当发生外部中断时，单片机进入外部中断程序，开始扫描按键，先扫描KEY0按键，看是否按下了起/停键，如果按下了该键，则执行起/停程序；如果没有则判断KEY1键，看是否按下了该键，若果按下了，则执行正转程序；若果没有判断KEY2键，若该键按下，则执行反转程序；若没有则检测KEY3键是否按下，若按下，则执行加速程序；若没有则检测KEY4键，若该键按下，则执行减速程序，否则退出中断程序。按键显示程序流程图如图13所示。在这段中断程序中，扫描按键，并执行相应的指令。因为把按键扫描程序放到外部中断程序中，单片机只有在接收到外部中断的申请时，才去扫描按键，在无按键按下时，不必浪费时间去扫描按键，提高了CPU的效率。



**图13 外部中断按键扫描程序**

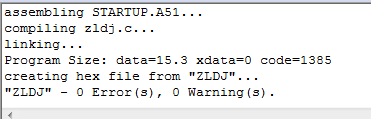
# 5.仿真分析

**5.1 Proteus简介**

Proteus开发平台不仅能够胜任PCB电路板设计制作的工作，同时还拥有进行交互式的仿真测试的功能。Proteus软件是目前用户最为青睐的仿真软件，它不仅能够完成单一电路以及混合电路的仿真与设计，同时能够很好的和市面上常见的微型控制器系统兼容匹配。它完美的完成了从设计原理图→仿真分析→代码调试→功能测试→形成PCB的完整过程。经过多年的发展与完善，如今无论是在性能上还是功能上都越来越值得广大用户的信赖。

**5.2软件仿真**

这部分仿真主要是解决程序中的语法错误。由于程序代码较长，在书写过程中难免出现语法错误和函数参数类型方面的错误，我选择使用keil4软件编写程序，通过这道程序的仿真，能基本解决这方面的错误，为下一步应用proteus软件仿真做好准备。其仿真结果见图14：



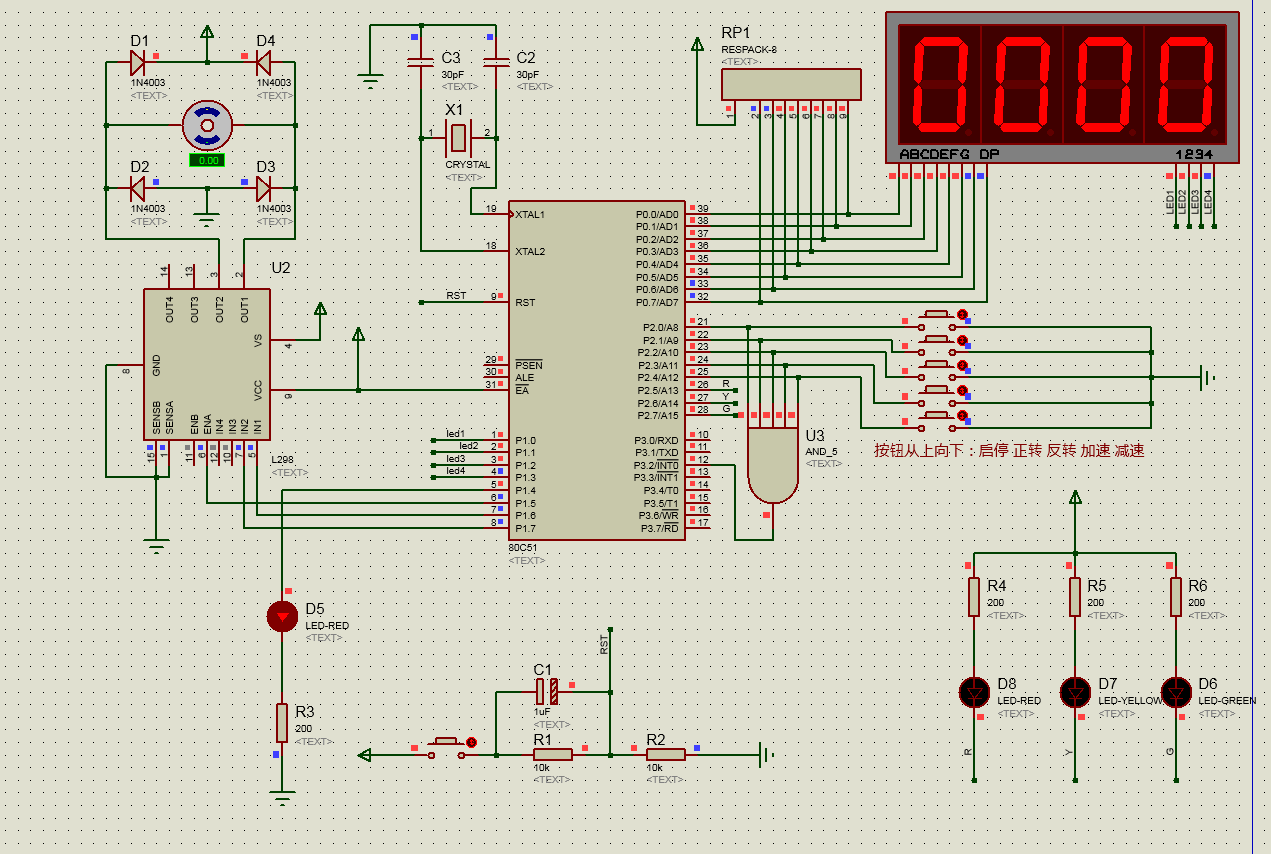
**图14 源程序仿真结果**

由图可知，程序已无语法错误，编译成功，并且生成hex文件，利用该文件可以与proteus软件联合仿真，进一步检查程序的语法错误。

**5.3运用proteus软件联合仿真**

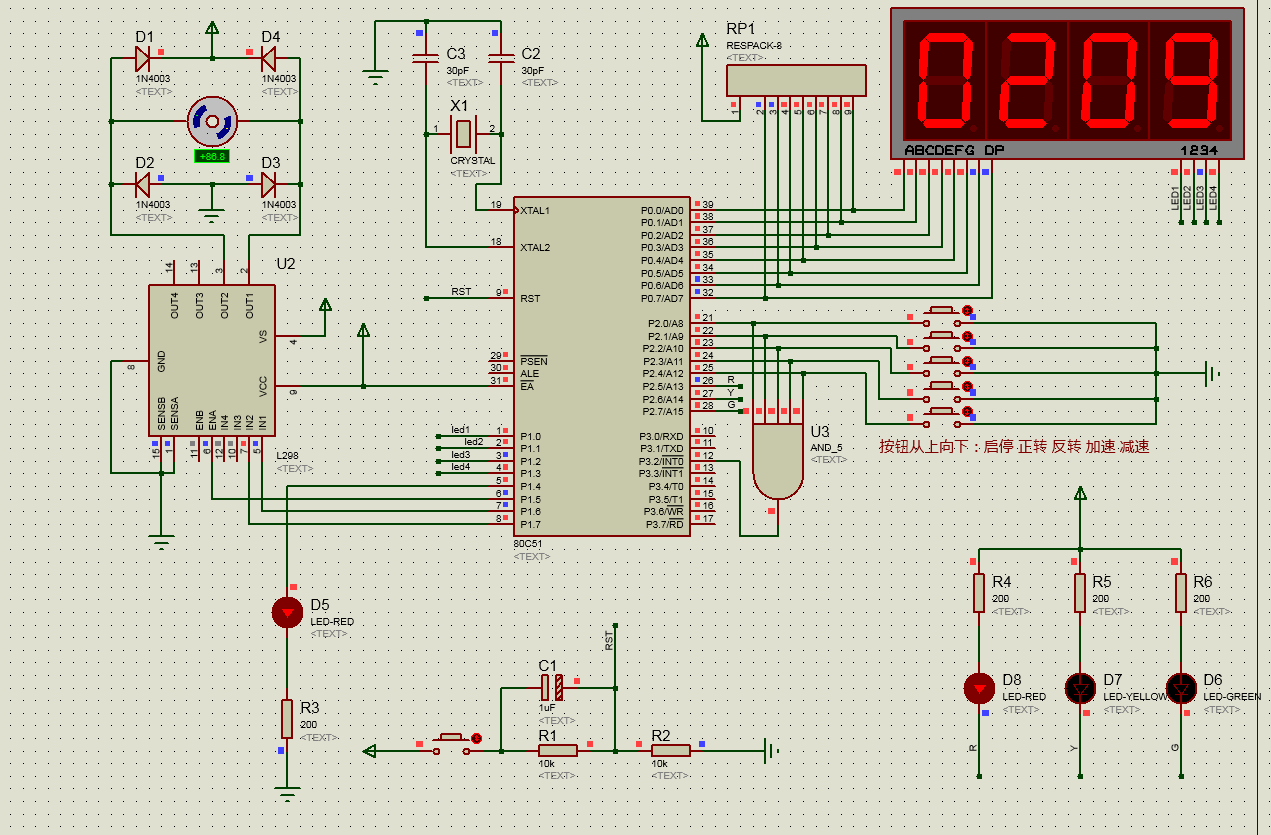
首先在proteus软件中画好硬件电路，并检查硬件电路是否满足要求。在确定硬件功能正常后，载入编译源程序生成的hex文件，在proteus软件中仿真运行，通过电路中各元件引脚显示的电平状态，检查软件逻辑上是否存在问题。再确认没有问题之后，开始进入仿真的环节。

点击左下角运行之后，电机显示0转速，数码管也显示速度为0000，右下角显示电机状态的LED灯全灭。如图15：



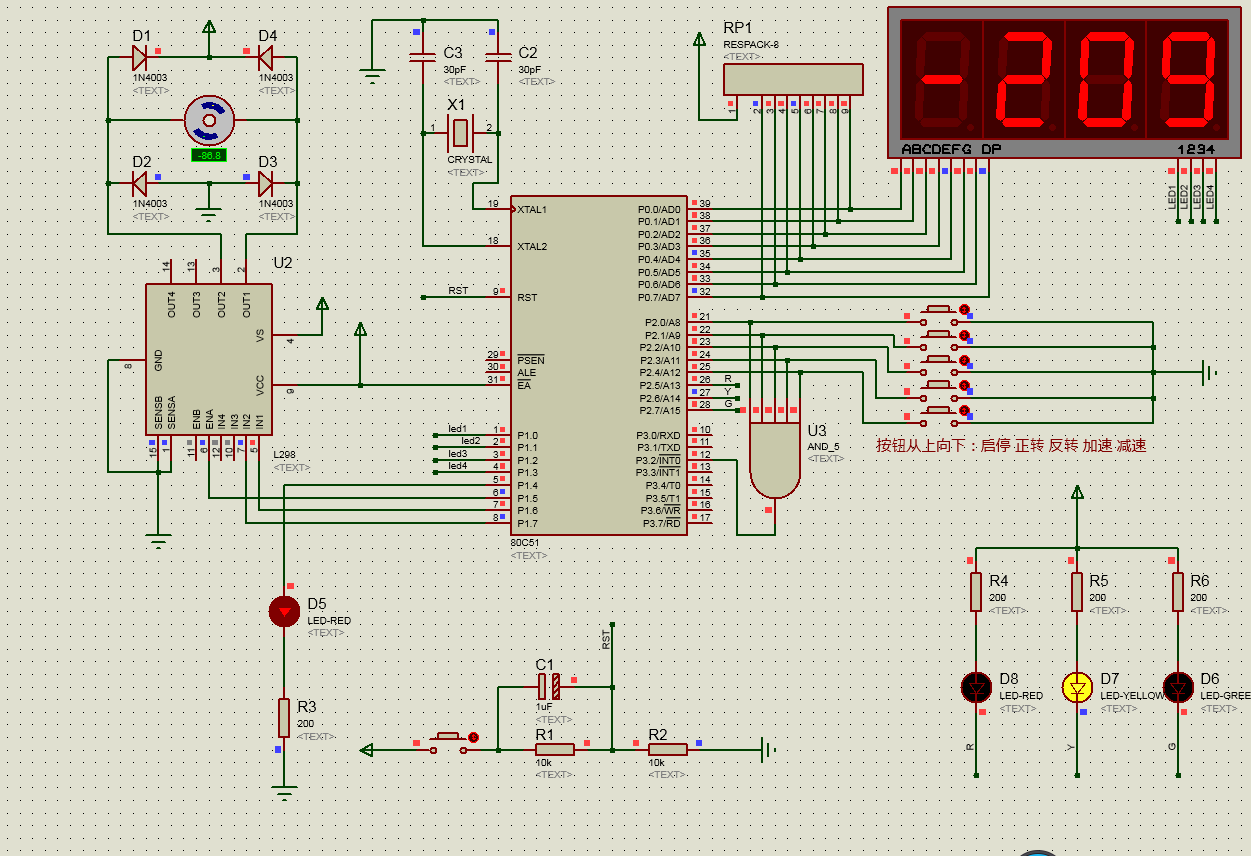
**图15 开始运行**

点击启动停止按钮，电机开始正转，并维持一定的速度，速度显示在数码管上为0209，右下角红灯亮，表示电机处在正转状态，如图16：



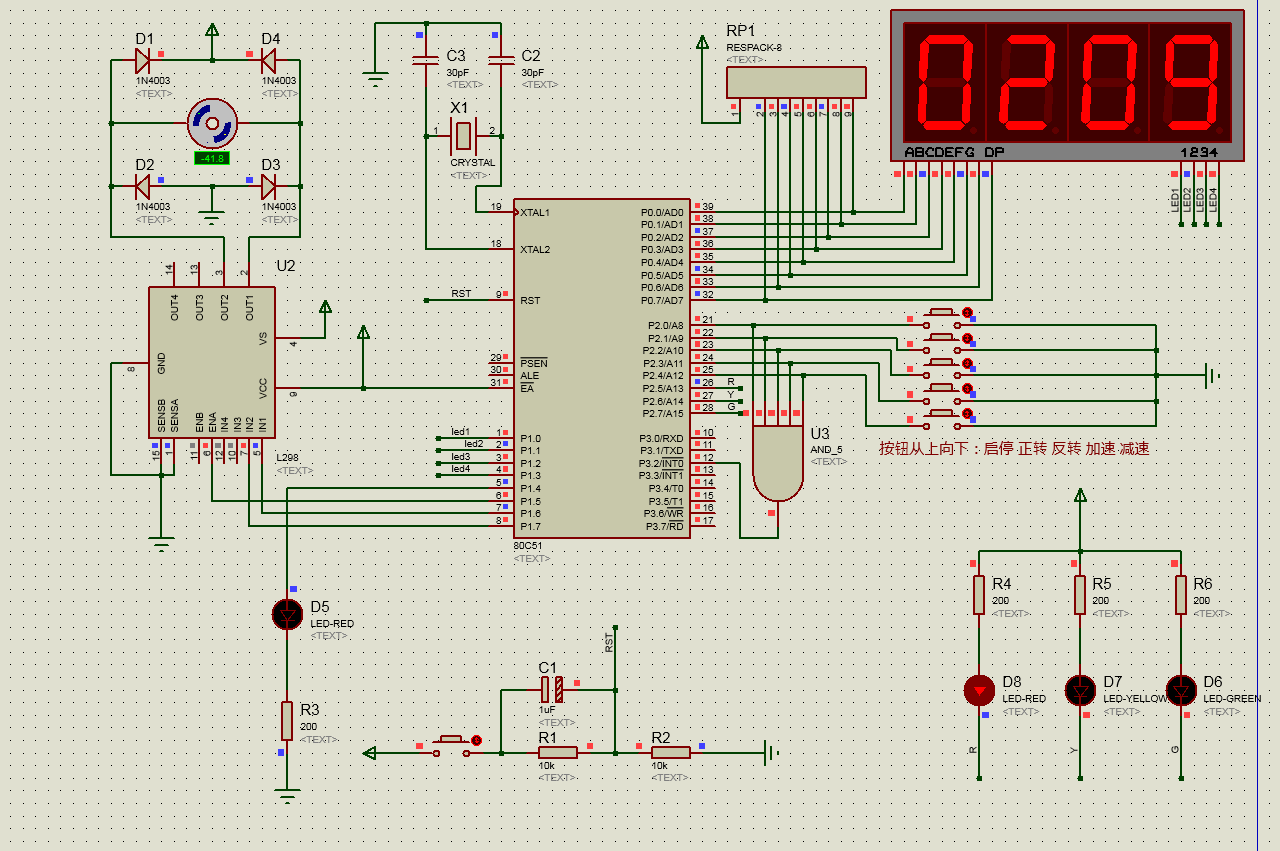
**图16 点击启动停止按钮后画面**

点击反转按钮，电机开始反转，并维持一定的速度，速度显示在数码管上为-209，右下角黄灯亮，表示电机处在反转状态，如图17：



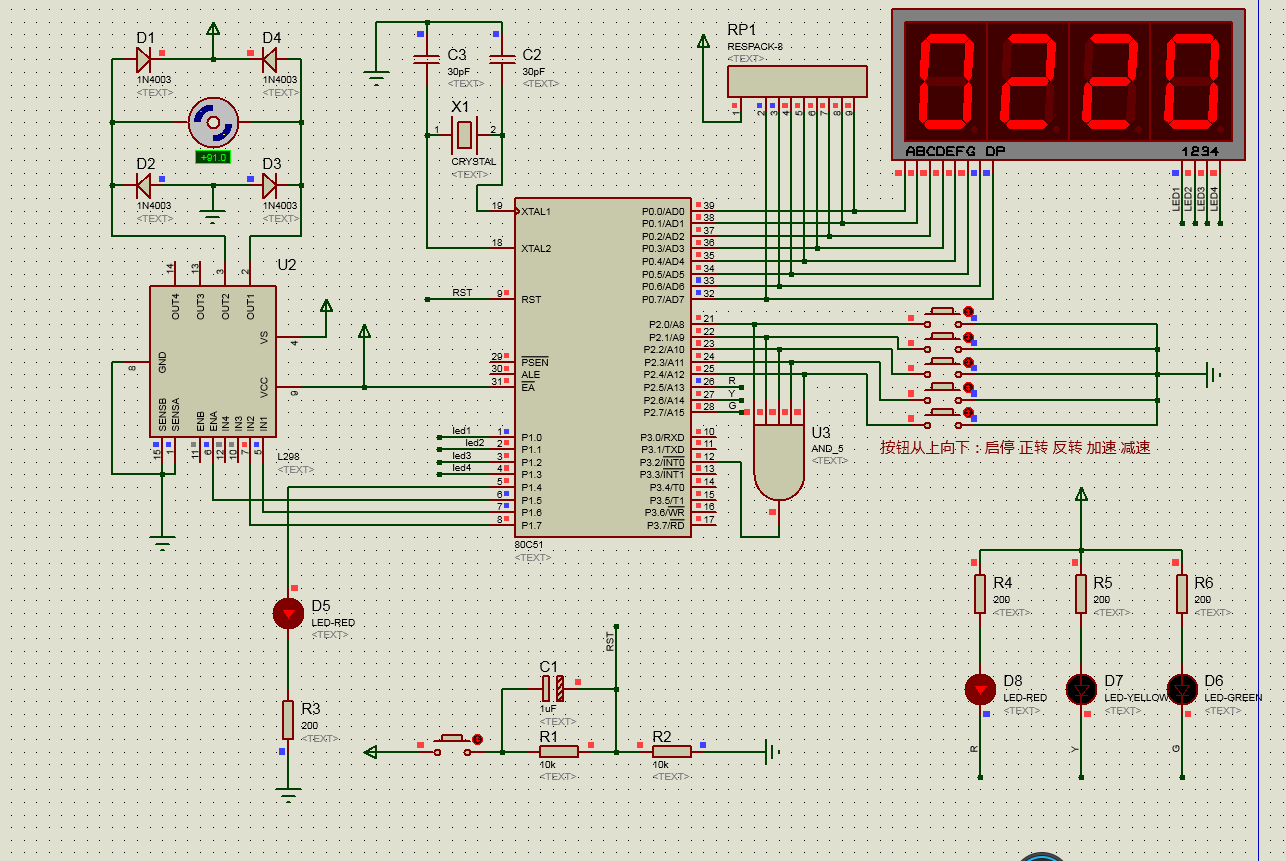
**图17 点击反转按钮后画面**

点击正转按钮，电机开始正转，并维持一定的速度，速度显示在数码管上为0209，右下角红灯亮，表示电机处在正转状态，如图18：



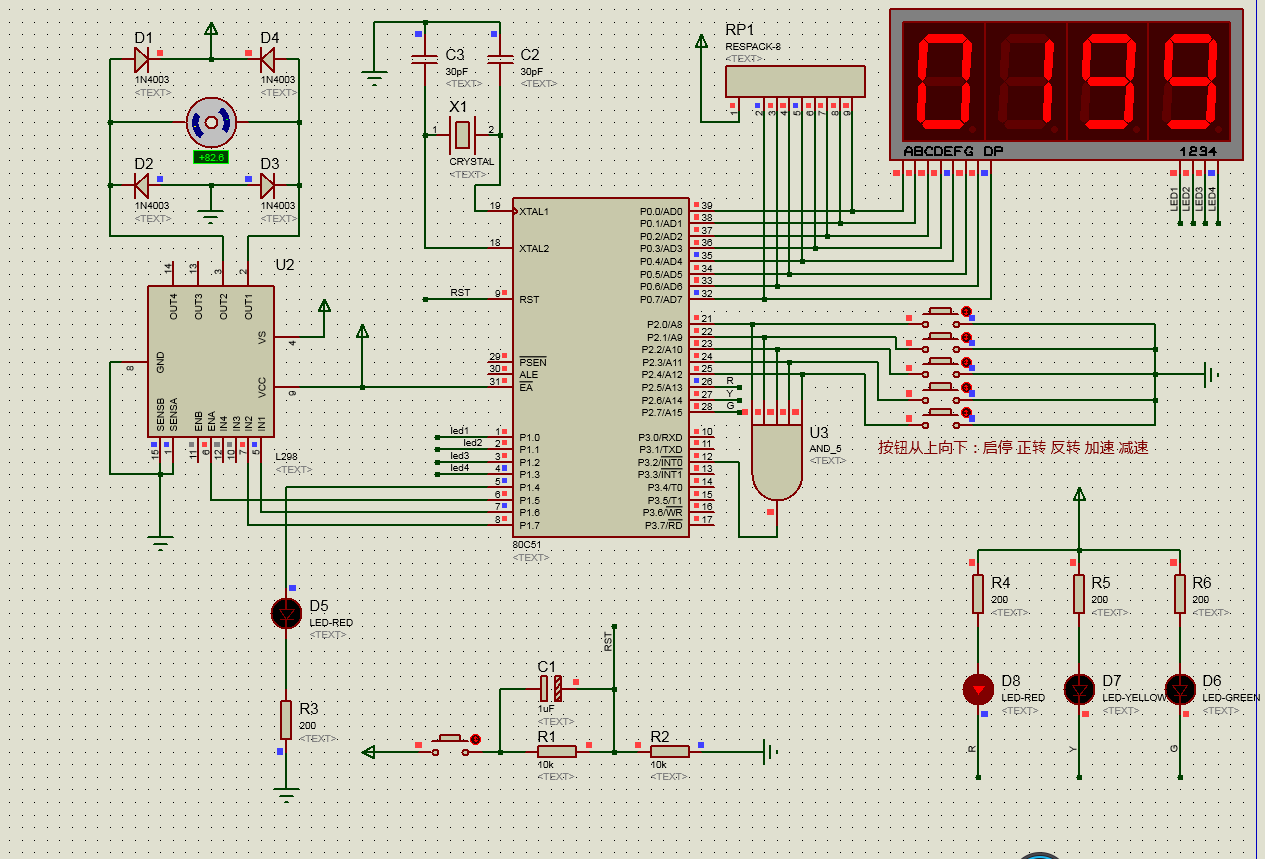
**图18 点击正转按钮后画面**

点击加速按钮，电机开始正转加速，速度显示在数码管上为0220，右下角红灯亮，表示电机处在正转状态，如图19：



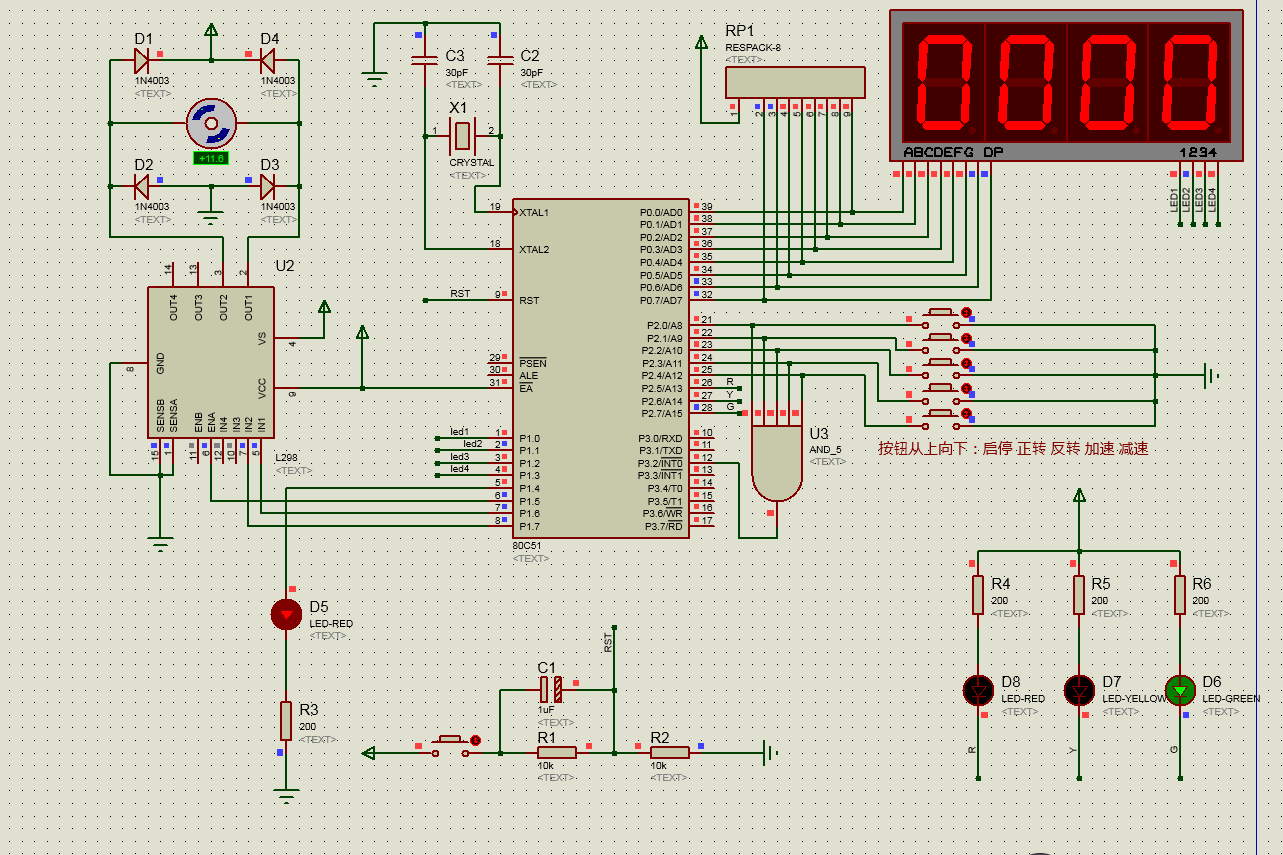
**图19 点击加速后画面**

点击两次减速按钮，电机开始正转减速，速度显示在数码管上为0199，右下角红灯亮，表示电机处在正转状态，如图20：



**图20 点击两次减速按钮后画面**

点击启动停止按钮，电机停止转动，速度显示在数码管上为0000，右下角绿灯亮，表示电机处在停止状态，如图21：



**图21 点击启动停止按钮后画面**

# 6 总结

通过此次课程设计，使我更加扎实的掌握了有关单片机方面的知识，在设计过程中虽然遇到了一些问题，但经过一次又一次的思考，一遍又一遍的检查终于找出了原因所在，也暴露出了前期我在这方面的知识欠缺和经验不足。实践出真知，通过亲自动手制作，使我们掌握的知识不再是纸上谈兵。过而能改，善莫大焉。在课程设计过程中，我们不断发现错误，不断改正，不断领悟，不断获取。最终的检测调试环节，本身就是在践行“过而能改，善莫大焉”的知行观。这次课程设计终于顺利完成了，在设计中遇到了很多问题，最后在老师的指导下，终于游逆而解。在今后社会的发展和学习实践过程中，一定要不懈努力，不能遇到问题就想到要退缩，一定要不厌其烦的发现问题所在，然后一一进行解决，只有这样，才能成功的做成想做的事，才能在今后的道路上劈荆斩棘，而不是知难而退，那样永远不可能收获成功，收获喜悦，也永远不可能得到社会及他人对你的认可！

**参考文献**

[1] 孟访. 基于PIC单片机的无刷直流电机控制系统研究[D].安徽理工大学,2017.

[2] 杨天. 移动机器人用无刷直流电机控制系统的研究[D].安徽工业大学,2017.

[3] 范守高. 电动车用无刷直流电机控制系统研究[D].江苏大学,2017.

[4] 党璇. 基于STM32的无刷直流电机控制系统研究[D].安徽工业大学,2017.

[5] 凡庆. 电动试验车轮毂电机控制器研究[D].长安大学,2017.

[6] 黄亚军. 电动车用无刷直流电机矢量控制系统研究[D].扬州大学,2017.

[7] 李美艳.基于单片机的直流电机调速系统的设计[J].电子设计工程,2016,24(22):158-160.

[8] 吴松. 串励直流电机新型控制系统研究[D].扬州大学,2016.

[9] 魏星. 用于燃料电池汽车双向DC/DC变换器研究[D].广西大学,2016.

[10] 菅一凡. 基于CAN总线的直流无刷轮毂电机控制的研究[D].天津科技大学,2016.

[11] 傅涛. 电动汽车用大功率无刷直流电机控制关键技术研究[D].天津大学,2016.

[12] 张艳昆,李伟,王崇哲,罗军,孙庆国,刘铁军. 一种双层结构空气式恒温槽总体设计[A]. 中国气象学会.第32届中国气象学会年会S16 地基遥感观测技术与应用[C].中国气象学会:,2015:4.

[13] 何彭. 基于STM32的无刷直流电机控制系统研究[D].武汉理工大学,2015.

[14] 杨玲. 微型电动汽车用无刷直流电机控制系统的研究[D].辽宁工业大学,2015.

[15] 石江涛. 基于STC单片机无刷直流电机控制系统的设计[D].太原科技大学,2014.

[16] 林继铭. 内燃—直线发电集成动力系统的性能分析与优化[D].南京理工大学,2014.

[17] 朱海舟. 无刷直流电机转速控制系统的研究与设计[D].长沙理工大学,2012.

[18] 赵文才. 机车牵引四象限变流器研究[D].华中科技大学,2011.

附录：完整的源程序

#include"reg52.h"

#define uchar unsigned char

#define uint unsigned int

sbit IN1=P1^6;

sbit IN2=P1^7;

sbit ENA=P1^5;

sbit KEY0=P2^0;

sbit KEY1=P2^1;

sbit KEY2=P2^2;

sbit KEY3=P2^3;

sbit KEY4=P2^4;

sbit LEDR=P2^5;

sbit LEDY=P2^6;

sbit LEDG=P2^7;

sbit WE1=P1^0;

sbit WE2=P1^1;

sbit WE3=P1^2;

sbit WE4=P1^3;

sbit led=P1^4;

bit flag=0;

bit sign=1;

uint num=0;

uchar code table[]={

0x3f,0x06,0x5b,0x4f,0x66,

0x6d,0x7d,0x07,0x7f,0x6f};

void delay(uint n)

{

uint i,j;

for(i=n;i>0;i--)

for(j=0;j<110;j++);

}

void initial()

{

IN1=0;

IN2=0;

ENA=0;

num=0;

TMOD=0x01;

TL0=0xff;

TH0=0xff;

EA=1;

EX0=1;

ET0=1;

IT0=1;

TR0=1;

}

void display(bit flag,uint n)

{

uint v;

uchar qian,bai,shi,ge;

v=(uint)(n\*535.0/255);

qian=v/1000;

bai=(v-1000\*qian)/100;

shi=v%100/10;

ge=v%10;

WE1=0;

if(flag==1)

P0=table[0];

else

P0=0x40;

delay(10);

WE1=1;

WE2=0;

P0=table[bai];

delay(10);

WE2=1;

WE3=0;

P0=table[shi];

delay(10);

WE3=1;

WE4=0;

P0=table[ge];

delay(10);

WE4=1;

}

void main()

{

initial();

while(1)

{

display(sign,num);

}

}

void int0() interrupt 0

{

if(KEY0==0)

{

delay(10);

if(KEY0==0)

{

flag=~flag;

if(flag==0)

{

num=0;

P2=0xff;

ENA=0;

IN1=0;

IN2=0;

LEDG=0;

sign=1;

}

else

{

P2=0xff;

num=100;

IN1=0;

IN2=1;

LEDR=0;

sign=1;

}

while(!KEY0);

}

}

if(KEY1==0)

{

delay(10);

if(KEY1==0)

{

P2=0xff;

//ENA=1;

IN1=0;

IN2=1;

LEDR=0;

sign=1;

}

while(!KEY1);

}

if(KEY2==0)

{

delay(10);

if(KEY2==0)

{

P2=0xff;

//ENA=1;

IN1=1;

IN2=0;

LEDY=0;

sign=0;

}

while(!KEY2);

}

if(KEY3==0)

{

delay(10);

if(KEY3==0)

{

num+=5;

if(num>=255)

num=255;

}

while(!KEY3);

}

if(KEY4==0)

{

delay(10);

if(KEY4==0)

{

num-=5;

if(num<=0)

num=0;

}

while(!KEY4);

}

}

void itt0() interrupt 1

{

ENA=~ENA;

TL0=0xff; //定时器重装初值

led=~led;

if(ENA==1)

TH0=255-num; //定时器高8位装初值

else

TH0=num;

}

|  |  |  |
| --- | --- | --- |
| **指导教师评语** | | |
|  | | |
| 成绩（60%） |  | 指导教师签字：  年 月 日 |
| **答辩过程及评价** | | |
|  | | |
| 成绩（40%） |  | 答辩小组签字：  年 月 日 |
| **院综合意见** | | |