《单片机系统设计》实验报告

单片机实现直流电机速度测量和控制实验



**学生姓名 \_\_\_赵思蒙**

**学生学号 517021910935**

**学生班级 F1702113**

**任课教师 \_\_付庄\_\_\_**

**同组同学 陆以凡**

**实验日期 \_\_2020.11.2\_\_**

6.5实验五 单片机实现直流电机速度测量和控制实验

**一、实验目的**

1. 进一步熟悉单片机的C51编程，该实验中应用到C51的标准库函数printf，scanf等，结构体的定义和使用，函数参数值的传递和函数值返回，指针的定义和使用等；
2. 进一步熟悉单片机资源的C语言编程方法，该实验包括外部中断、定时器中断及中断优先级，串口通讯，LED显示等；
3. 熟悉速度测量的M法和T法，简单使用PID算法控制速度，该实验中采用T法，即通过定时器记录电机旋转一周所用的时间来计算速度。

**二、实验内容**

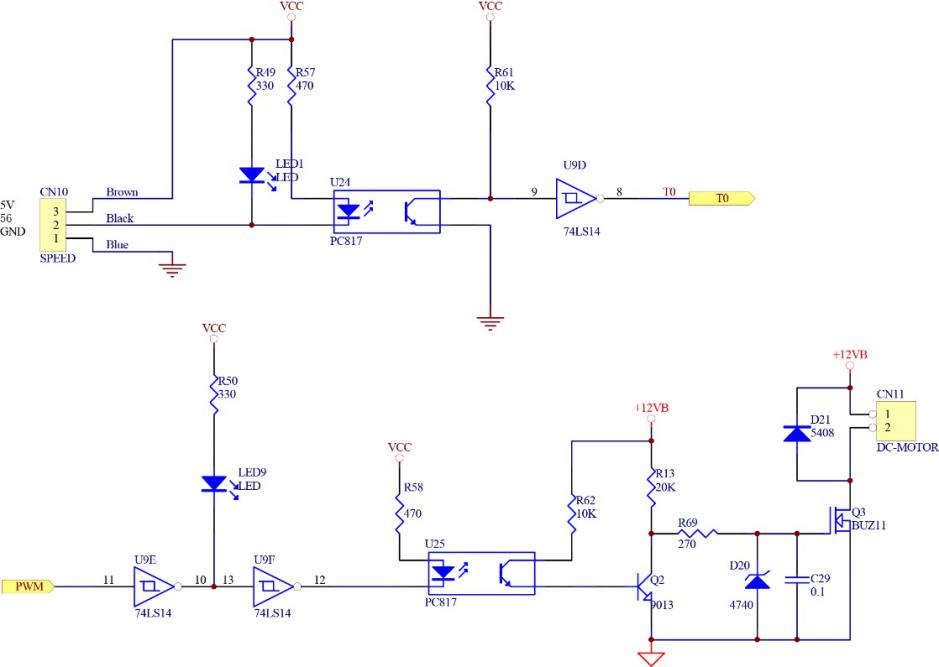
1. 速度测量和控制的范围是5转/s到50转/s；
2. 采用T法测量电机速度，每0.5秒进行一次调节PWM的占空比来调节电机速度，采用定时器0定时，定时时间0.4毫秒实现硬件定时，定时器采用中断方式；编码器信号接到单片机INT1引脚，电机每旋转一圈INT1中断一次；另定义两个软件计数器，分别用来记录INT1中断时定时器0的溢出次数和PWM的高电平定时器0的溢出次数；这样5转/s和50转/s的定时器溢出次数分别为400和40，PWM的载波周期（0.5s）内占空比的调节分辨率为1250；
3. 设定速度从上位机发送，printf函数输出“Please input the command speed!”,串口调试软件输入设定速度，单位转/s，单片机采用scanf函数接收指令速度，接收到后返回“The current command speed is xx!”,串口波特率采用单片机的定时器1实现；
4. 数码管的前两位显示设定速度，后两位显示当前速度；
5. 编写PID控制算法子函数，函数输入参量是指令速度和当前速度，函数返回值是PWM占空比的高电平计数值；
6. 外部中断INT1的中断函数中允许定时器0中断，即定时器0的中断优先级高于外部中断INT1。

三、实验电路及原理

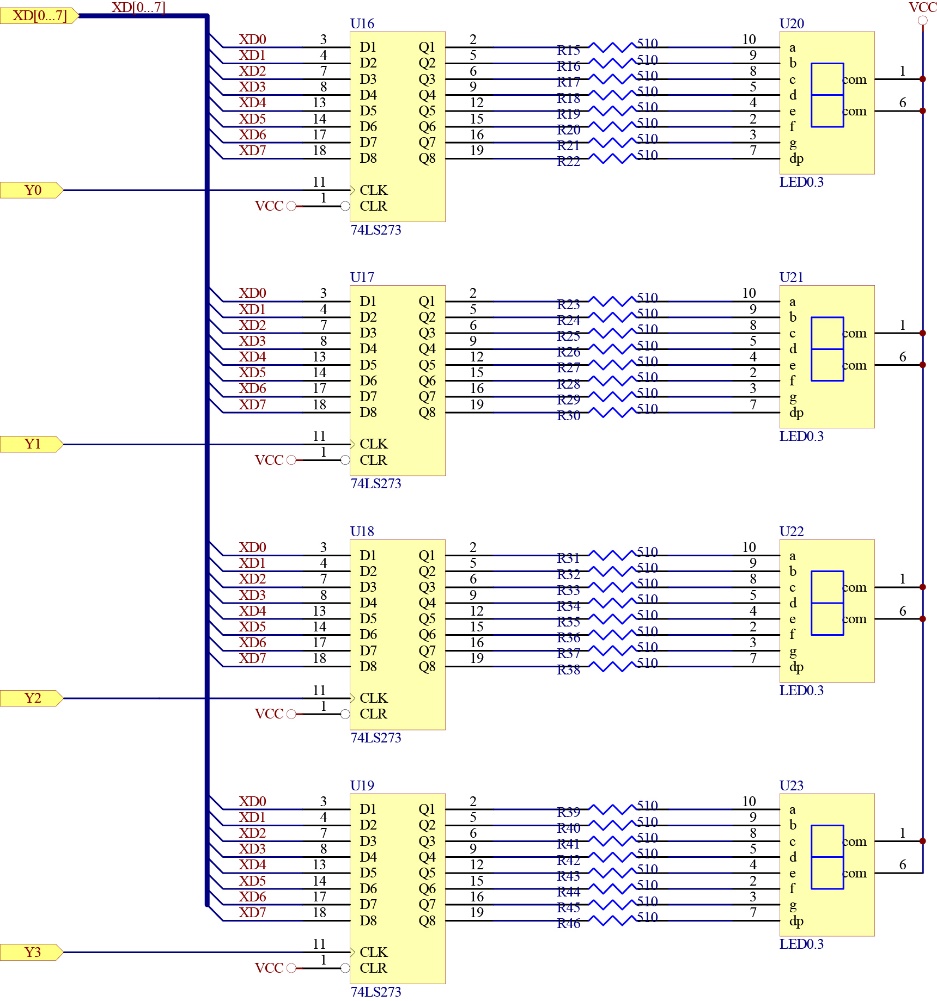
原理图：



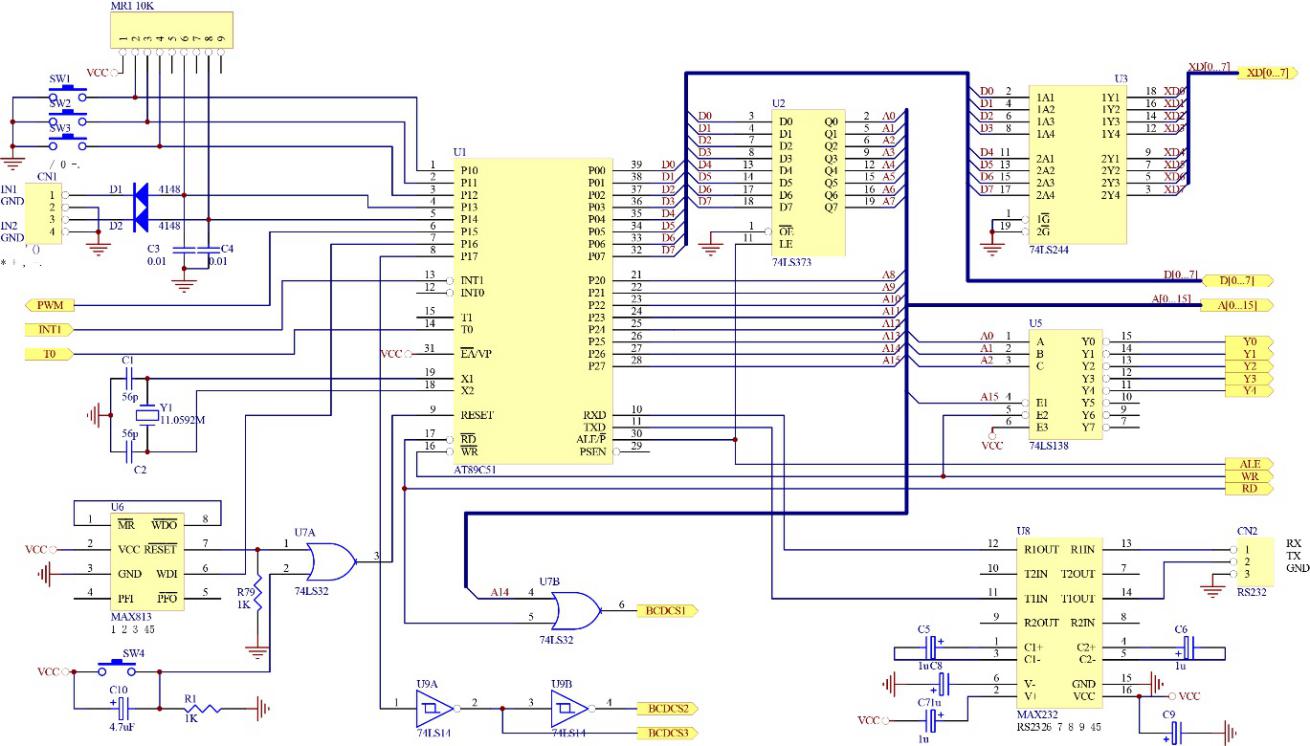
电路图：



转速测量、电机驱动模块

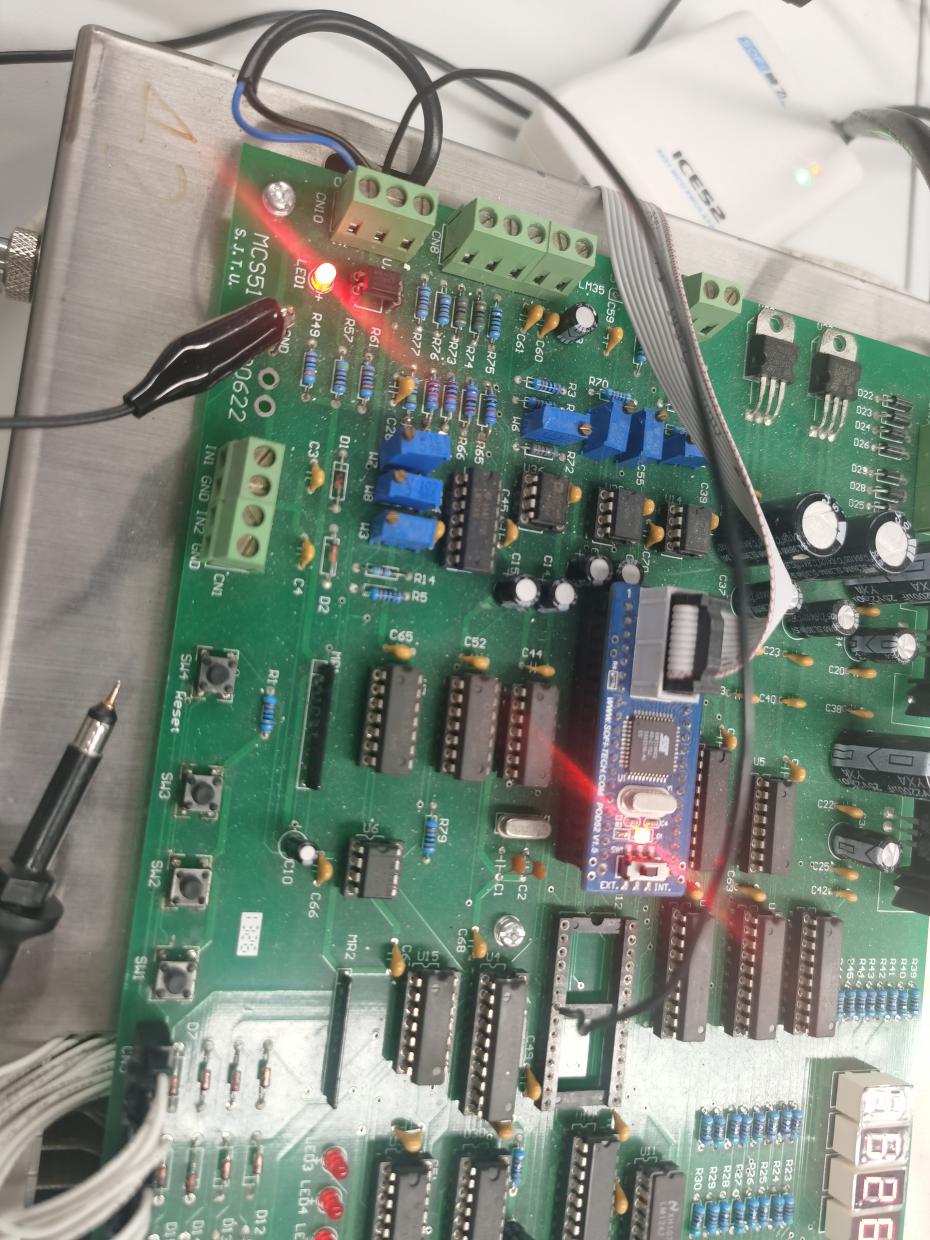


LED模块

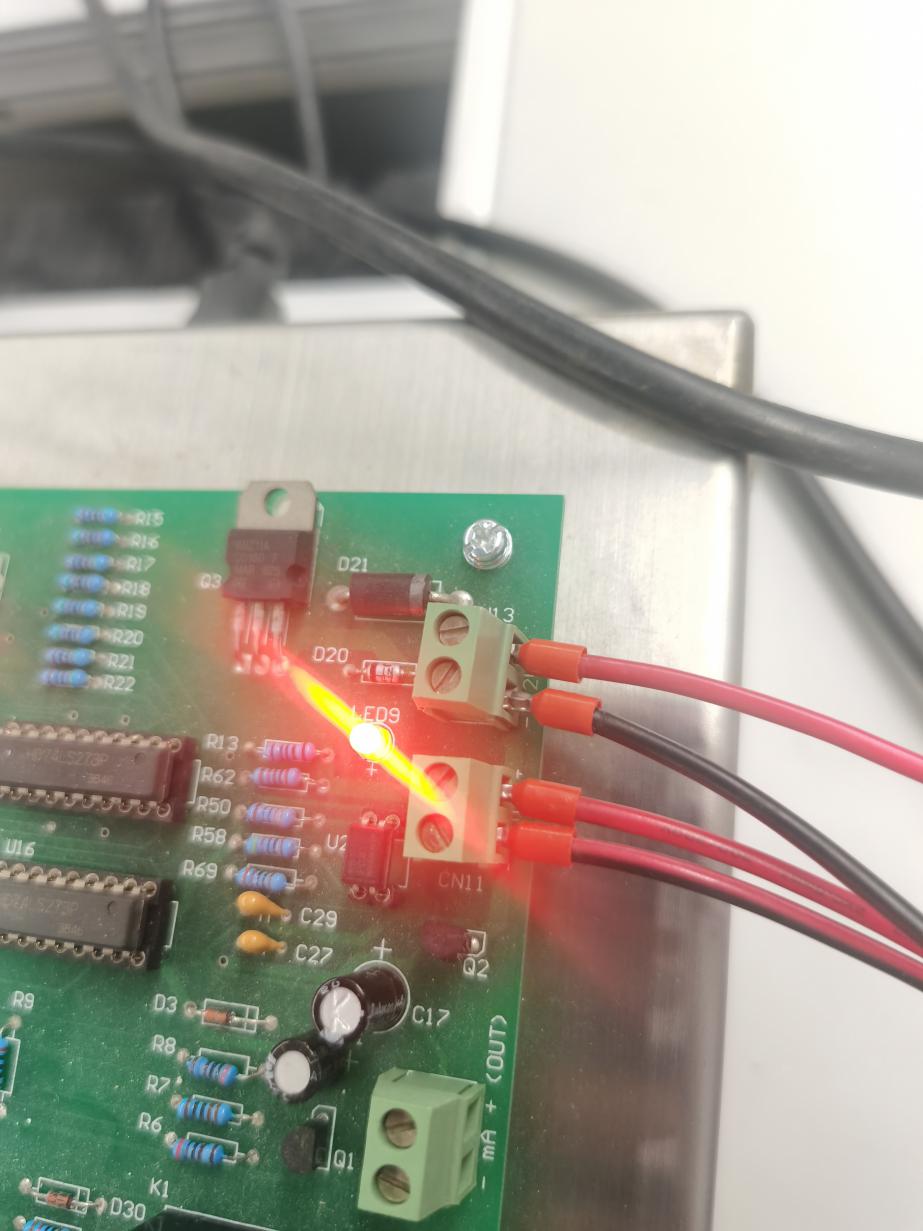


单片机模块

硬件与连线：



测量电路连线

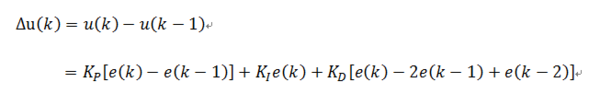


控制电路连线

1. 编程思路

PID算法简介：

增量式 PID：



比例P :    e(k)-e(k-1)   当前误差 - 上次误差

积分I :   e(i)     当前误差

微分D :  e(k) - 2e(k-1)+e(k-2)   当前误差 - 2\*上次误差 + 上上次误差

 增量式PID根据公式可以很好地看出，一旦确定了 KP、TI  、TD，只要使用前后三次测量值的偏差， 即可由公式求出控制增量。而得出的控制量。

▲u(k)对应的是近几次位置误差的增量，而不是对应与实际位置的偏差 没有误差累加

增量型 PID，是对位置型 PID 取增量，这时控制器输出的是相邻两次采样时刻所计算的位置值  
之差，得到的结果是增量，即在上一次的控制量的基础上需要增加（负值意味减少）控制量。

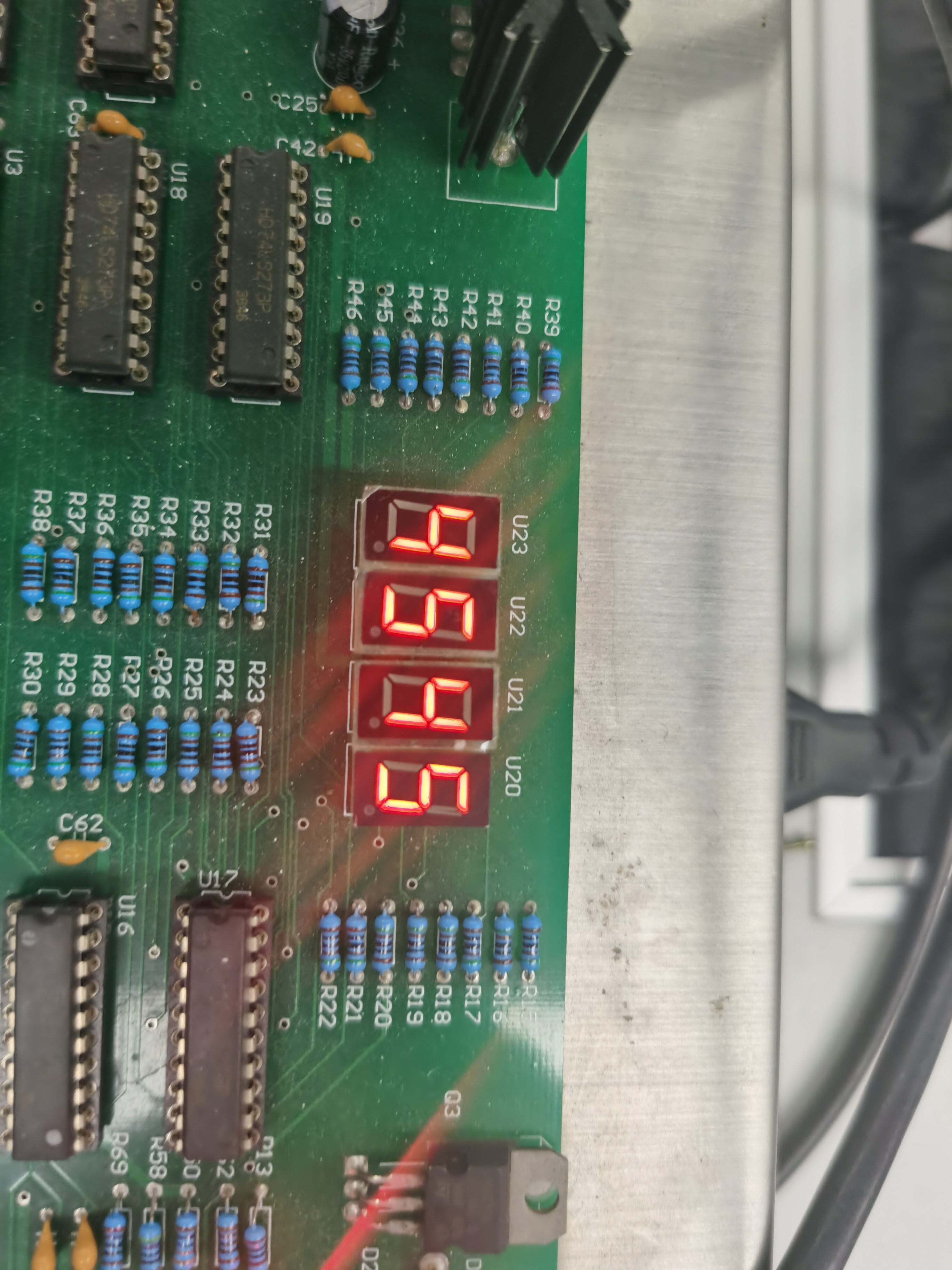
本次实验采用增量式PID算法进行控制，KP系数初始设置为40，KI环节系数初始设置为7，KD环节系数初始设置为2。

|  |  |
| --- | --- |
| 主程序 | 定时器0中断程序 |
|  |  |

（程序代码见附录）

1. 结果与反思

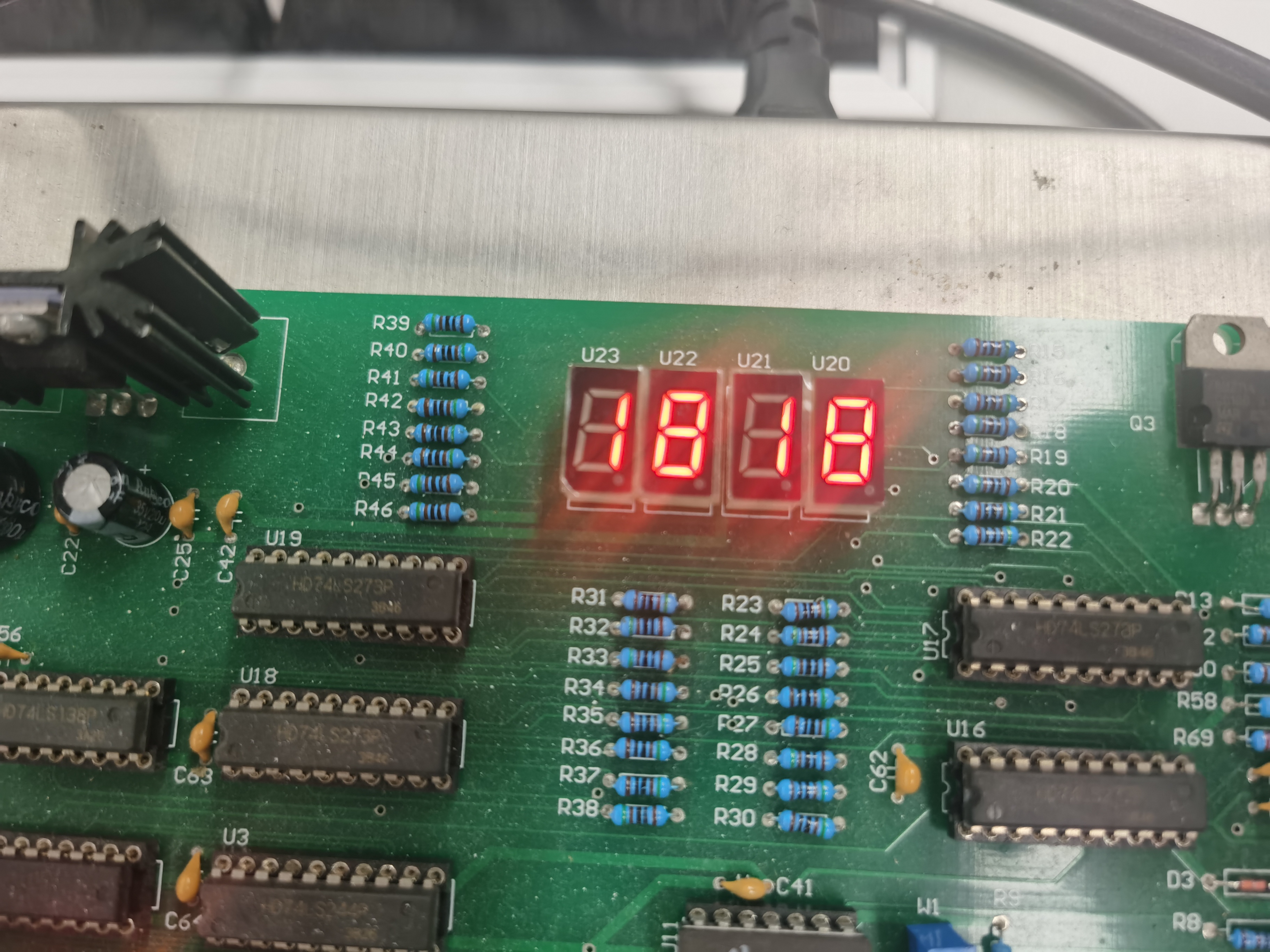
最终结果比较好的符合预期，PID效果较好。以下是效果图，（左边数字为设定转速，右侧数字为测得转速。）



45r/s



35r/s



18r/s

遇到的问题：

1. 响应速度过慢

发现是KI数值过小以及脉冲周期太长的问题。通过增加KI和将m从1250下调至600(即把周期从500ms下调至250 ms)增加了响应速度。

1. 振动过大

发现是KP设置过小导致，通过增加KP同时引入KD系数来减缓振动，最终成功使得示数变得稳定。

感谢老师的指导和同组同学的帮助。

代码如下：

#include<AT89X51.h>

#include<absacc.h>

#include<stdio.h>

//#define adc XBYTE[0xDFF8]

//#define dac XBYTE[0xEFFF]

#define uchar unsigned char

#define uint unsigned int

#define T0high 0xFE

#define T0low 0xB7 //65169~65536,398us

#define M 600

#define N 8

char code dx516[3] \_at\_ 0x003b;

uchar xdata \*ledadr\_tab[]={0x7FF8,0x7FF9,0x7FFA,0x7FFB};

uchar code tab[]={0xc0,0xf9,0xa4,0xb0,0x99,0x92,0x82,0xf8,0x80,0x90,0x88,0x83,0xc6,0xa1,0x86,0x8e};

uint data count\_t0=0,count\_pwm=0,m=0;

uchar data v\_input=0,v\_now=0;

bit ctrl\_flag=0;

uchar v\_calc(uint n);

void pid\_calc(uchar pid\_v\_ctrl,uchar pid\_v\_now);

void LEDfun(uchar y,uchar x,bit d);

void show0(uchar x);

void show1(uchar x);

void display(void);

struct PID

{

uchar KP;

uchar KI;

uchar KD;

}PID1={40,7,2};

void main(void)

{

uint i;

//LEDfun(3,0,0);LEDfun(2,0,0);LEDfun(1,0,1);LEDfun(0,0,0);

IE=0x8E;//EA=1;EX1=1;ET0=1;ET1=1

PT0=1;//T0 HIGH

PCON=0x00;

SCON=0x52;//mode 1;REN=1

TMOD=0x21;//T1 mode=2; T0 mode=1

TH1=0xFD;TL1=0xFD;//9600bps

TH0=T0high;TL0=T0low;

IT1=1;

TR1=1;TR0=1;

printf("Please input the command speed!(m/s)\n");

scanf("%bd",&v\_input);

printf("The current command speed is %bd!\n",v\_input);

P1\_5=1;

while(1)

{

if(P1\_0==0)//press SW1 to change v\_input

{

for(i=0;i<2000;++i);

if(P1\_0==0)

{

while(P1\_0==0);

printf("Please input the command speed!(m/s)\n");

scanf("%bd",&v\_input);

printf("The current command speed is %bd!\n",v\_input);

}

}

if(ctrl\_flag==1){pid\_calc(v\_input,v\_now);ctrl\_flag=0;}//calc m

}

}

void show0(uchar x)//low0~1

{

LEDfun(1,(x/10)%10,0);

LEDfun(0,x%10,0);

}

void show1(uchar x)//high2~3

{

LEDfun(3,(x/10)%10,0);

LEDfun(2,x%10,0);

}

void display(void)

{

show0(v\_now);

show1(v\_input);

}

void LEDfun(uchar y,uchar x,bit d) //共阳极 d=1表示需要小数点

{

uchar xdata \*led\_adr;

uint dd;

if(d==1)

{

dd=0x7F;

}

else

{

dd=0xFF;

}

led\_adr=ledadr\_tab[y];

\*led\_adr=tab[x]&dd;

}

void INT1\_int(void) interrupt 2

{

ET0=1;

v\_now=v\_calc(count\_t0);

//display();

count\_t0=0;

}

void T0\_int(void) interrupt 1

{

if(v\_input!=0)

{

count\_t0++;

count\_pwm++;

if(count\_pwm<m){P1\_5=1;}

else{P1\_5=0;}

if(count\_pwm==M){

count\_pwm=0;P1\_5=1;

//pid\_calc(v\_input,v\_now);

ctrl\_flag=1;

}

}

TH0=T0high;TL0=T0low;

}

void T1\_int(void) interrupt 3

{

}

uchar v\_calc(uint n)

{

return 2500/n;

}

void pid\_calc(uchar pid\_v\_ctrl,uchar pid\_v\_now)

{

static int err0=0,err1=0,err2=0;

uint p,i,d;

err0=pid\_v\_ctrl-pid\_v\_now;

p=err0-err1;

i=err0;

d=err0-2\*err1+err2;

m+=PID1.KP\*p+PID1.KI\*i+PID1.KD\*d;

err2=err1;err1=err0;

if(m<0){m=0;}

else if(m>M){m=M;}

printf("m=%d,v\_now=%bd\n",m,v\_now);

display();

}