单片机实验六实验报告

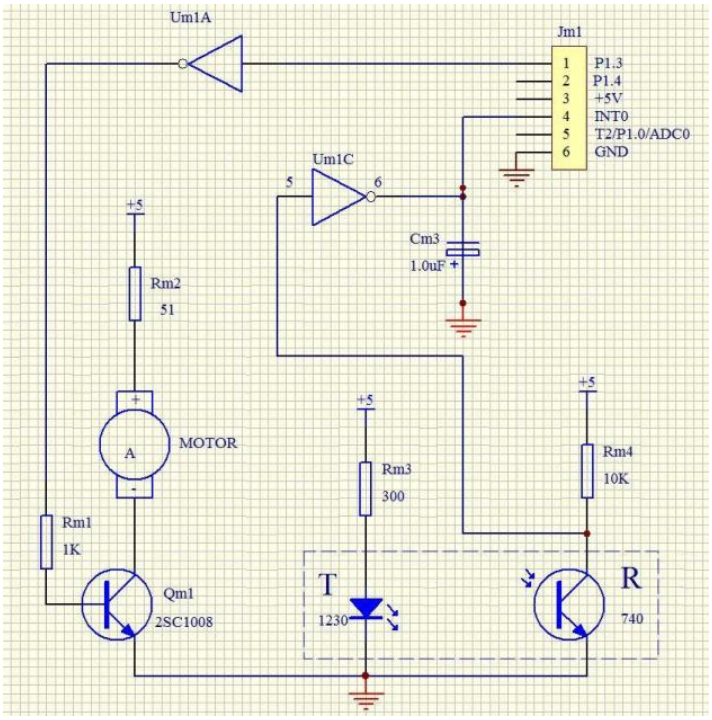
一、实验目的&实验要求

- 掌握点阵式液晶显示屏的原理和控制方法，掌握点阵字符的显示方法。
- 掌握脉宽调制调速的原理与方法，学习频率/周期测量的方法，了解闭环控制的原理。

二、实验内容

- 在液晶显示屏上显示出直流电机的：当前转速、低目标转速、高目标转速。
- 固定向 P1.1 输出 0，然后测量每秒钟电机转动的次数，将其显示在数码管，每秒钟刷新一次即可。
- 使用脉宽调制的方法，动态调整向 P1.1 输出的内容，使得电机转速能够稳定在一个预定值附近，同时实时显示当前转速。
- 根据输入修改电机得目标转速值，设置两个转速目标值：低转速和高转速。
- 每隔一秒钟读取两个开关的状态，如果 S1 按下，动态调整输出，使得电机转速能够稳定到低转速目标值附近，如果 S2 按下，动态调整输出，使得电机转速能够稳定到高转速 目标值附近。交替显示目标值和当前转速值。

三、实验原理



-

对于直流电机来说，其转速由输入电压决定，因此具有平滑调速的效果；相比而言，交流电机的转速由交流电频率和电机结构决定，难以改变速度。当然，交流电机构造简单，没有换向器，所以容易制造高转速、高电压、大电流、大容量的电机；而直流电机一般用在负荷小，但要求转速连续可调的场合。

- 电机转速就是一秒钟之内 INT0 的中断个数。

● 脉宽调制（Pulse Width Modulation，PWM）是一种能够通过开关量输出达到模拟量输出效果的方法。使用 PWM 可以实现频率调制、电压调制等效果，并且需要的外围器件较少，特别适合于单片机控制领域。这里只关心通过 PWM 实现电压调制，从而控制直流电机转速的效果。也称作脉宽调制调速。

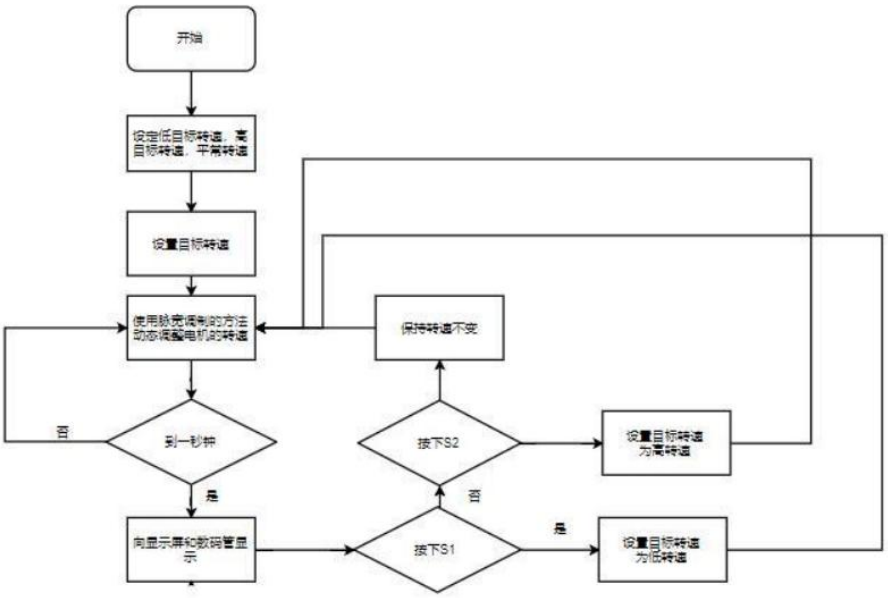
● 使用单片机实现 PWM，就是根据预定的占空比 δ 来输出 0 和 1，这里 δ 就是控制变量。最简单的办法就是以某个时间单位（如 0.1ms，相当于 10kHz）为基准，在前 N 段输出 1，后 M-N 段输出 0，总体的占空比就是 N/M。这种方法由于 0 和 1 分布不均匀，所以要求基准频率要足够高，否则会出现颠簸现象。要达到更稳定的效果，可以采用累加进位法如果将总的周期内的 0 和 1 均匀分散开。设置一个累加变量 x，每次加 N，若结果大于 M，则输出 1，并减去 M；否则输出 0。这样整体的占空比也是 N/M。在实验中取 M=256 可以使程序更加简单。

● 另外，由于本实验板的设计，输出 0 使电机工作。因此对于本实验，上面所说的 0 和 1 要翻转过来用。在本实验板中，电机每转动一次，与之相连的偏心轮将遮挡光电对管一次，因此会产生一个脉冲，送到 INT0。要测量转速，既可以测量相邻两次中断之间的时间；也可以测量一秒种之内发生的中断次数。显然，后一种方法更加简单。

● 进行转速控制时，涉及到三个变量：预期转速，实际转速和控制变量。这里控制变量就是占空比。我们并不能够预先精确知道某个控制变量的值会导致多少的实际转速，因为这里有很多内部和外部因素起作用（如摩擦力，惯性等），但可以确定就是随着控制变量的增加，实际转速会增加。反馈控制的基本原理就是根据实际结果与预期结果之间的差值，来调节控制变量的值。当实际转速高于预期转速时，我们需要减少控制变量，以降低速度；反之则需要调高控制变量。

● 本实验的转速控制可以使用简单的比例控制算法，也就是当转速 S 大于预定值时，将输出 0 的个数减少；当转速小于预定值时，将输出 0 的个数增加。改变值正比于测量出的差值。

四、实验流程图



五、代码实现

```
#include <reg52.h>
#include <intrins.h>
#define uchar unsigned char
#define uint unsigned int
//数码管初始化
sfr P4=0xC0;//初始化 P4 端口地址
sfr P4SW=0xBB;//P4SW 默认值 驱动
sbit sclk=P4^4;//模拟串口时钟
sbit sdata=P4^5;//模拟串口数据
//液晶屏初始化
sbit CS1=P1^7;
sbit CS2=P1^6;
sbit E=P3^3;//使能端
sbit RW=P3^4;//读写操作
sbit RS=P3^5;//寄存器选择
sbit RES=P1^5;//复位 低电平有效
sbit BUSY=P2^7;//当前为运行状态（忙状态位）
//直流电机初始化
sbit swh1=P3^6;//开关 S1
sbit swh2=P3^7;//开关 S2
sbit motor=P1^1;//从 p1.1 输出
//设置码值表，用于存放在液晶屏上显示的字符和数字
uchar code zima[20][32]=
{
0x00,0x00,0xC0,0xE0,0x30,0x10,0x08,0x08,0x08,0x08,0x08,0x18,0x30,0xE0
,
0xC0,0x00,
0x00,0x00,0x07,0x0F,0x18,0x10,0x20,0x20,0x20,0x20,0x20,0x10,0x18,0x0F
,
0x07,0x00,///*"0"*0/0x00,0x00,0x00,0x10,0x10,0x10,0x10,0xF0,0xF8,0x00,
0x00,0x00,0x00,0x00,
0x00,0x00,
0x00,0x00,0x00,0x20,0x20,0x20,0x20,0x3F,0x3F,0x20,0x20,0x20,0x20,0x00
,
0x00,0x00,//*"1"*1/
0x00,0x00,0x60,0x50,0x10,0x08,0x08,0x08,0x08,0x08,0x08,0x98,0xF0,0x70
,
0x00,0x00,
0x00,0x00,0x20,0x30,0x28,0x28,0x24,0x24,0x22,0x22,0x21,0x20,0x30,0x18
,
0x00,0x00,//*"2"*2/
0x00,0x00,0x30,0x30,0x08,0x08,0x88,0x88,0x88,0x88,0x58,0x70,0x30,0x00
,
0x00,0x00,
```

0x00, 0x00, 0x18, 0x18, 0x20, 0x20, 0x20, 0x20, 0x20, 0x20, 0x31, 0x11, 0x1F, 0x0E
,
0x00, 0x00, //*"3"*3/
0x00, 0x00, 0x00, 0x00, 0x00, 0x80, 0x40, 0x20, 0x10, 0xF0, 0xF8, 0xF8, 0x00, 0x00
,
0x00, 0x00,
0x00, 0x04, 0x06, 0x05, 0x05, 0x04, 0x24, 0x24, 0x24, 0x3F, 0x3F, 0x3F, 0x24, 0x24
,
0x24, 0x00, //*"4"*4/
0x00, 0x00, 0x00, 0xC0, 0x38, 0x88, 0x88, 0x88, 0x88, 0x88, 0x88, 0x88, 0x08, 0x08
,
0x00, 0x00,
0x00, 0x00, 0x18, 0x29, 0x21, 0x20, 0x20, 0x20, 0x20, 0x20, 0x30, 0x11, 0x1F, 0x0E
,
0x00, 0x00, //*"5"*5/
0x00, 0x00, 0x80, 0xE0, 0x30, 0x10, 0x98, 0x88, 0x88, 0x88, 0x88, 0x88, 0x98, 0x10
,
0x00, 0x00,
0x00, 0x00, 0x07, 0x0F, 0x19, 0x31, 0x20, 0x20, 0x20, 0x20, 0x20, 0x20, 0x11, 0x1F
,
0x0E, 0x00, //*"6"*6/
0x00, 0x00, 0x30, 0x18, 0x08, 0x08, 0x08, 0x08, 0x08, 0x88, 0x48, 0x28, 0x18, 0x08
,
0x00, 0x00,
0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x38, 0x3E, 0x01, 0x00, 0x00, 0x00, 0x00, 0x00
,
0x00, 0x00, //*"7"*7/
0x00, 0x00, 0x70, 0x70, 0xD8, 0x88, 0x88, 0x08, 0x08, 0x08, 0x08, 0x98, 0x70, 0x70
,
0x00, 0x00,
0x00, 0x0C, 0x1E, 0x12, 0x21, 0x21, 0x20, 0x21, 0x21, 0x21, 0x23, 0x12, 0x1E, 0x0C
,
0x00, 0x00, //*"8"*8/
0x00, 0xE0, 0xF0, 0x10, 0x08, 0x08, 0x08, 0x08, 0x08, 0x08, 0x18, 0x10, 0xF0, 0xC0
,
0x00, 0x00,
0x00, 0x00, 0x11, 0x33, 0x22, 0x22, 0x22, 0x22, 0x22, 0x32, 0x11, 0x1D, 0x0F, 0x03
,
0x00, 0x00, //*"9"*9/
0x08, 0x08, 0x0A, 0xEA, 0xAA, 0xAA, 0xAA, 0xFF, 0xA9, 0xA9, 0xA9, 0xE9, 0x08, 0x08
,
0x08, 0x00,
0x40, 0x40, 0x48, 0x4B, 0x4A, 0x4A, 0x4A, 0x7F, 0x4A, 0x4A, 0x4A, 0x4B, 0x48, 0x40
,

```
0x40,0x00,//*"?"*10/
0x40,0x40,0x40,0xDF,0x55,0x55,0x55,0xD5,0x55,0x55,0x55,0xDF,0x40,0x40
,
0x40,0x00,
0x40,0x40,0x40,0x57,0x55,0x55,0x55,0x7F,0x55,0x55,0x55,0x57,0x50,0x40
,
0x40,0x00,//*"?"*11/
0x00,0x00,0x00,0x00,0x00,0x00,0xC0,0xC0,0xC0,0xC0,0x00,0x00,0x00,0x00
,
0x00,0x00,
0x00,0x00,0x00,0x00,0x00,0x00,0x30,0x30,0x30,0x30,0x00,0x00,0x00,0x00
,
0x00,0x00,//*":"*12/0x00,0x04,0x04,0xE4,0x24,0x24,0x24,0x3F,0x24,0x24,
0x24,0xE4,0x04,0x04,
0x00,0x00,
0x00,0x00,0x80,0x43,0x31,0x0F,0x01,0x01,0x01,0x3F,0x41,0x43,0x40,0x40
,
0x70,0x00,//*"?"*13/
};
uchar tab[15]=
{0xC0,0xF9,0xA4,0xB0,0x99,0x92,0x82,0x0F8,0x80,0x90};//0-9
uchar tspeed=0;//脉冲计数
uchar cspeed=0;//当前转速
uchar xspeed=130;//预定转速
uchar speedUp = 160;//最高转速
uchar speedLow =100;//最低转速
uchar t1_cnt=0; //1s=50ms*20
//占空比设置
int N=50;
int M=256;
int X=0;
void send_byte(uchar dat ,uchar cs1,uchar cs2);//向液晶屏送 8 位数据
void send_all(uint page,uint lie,uint offset);//写液晶屏，显示相应的
字
void init();//数码管和中断初始化
void clearscreen();//清屏
void init_yejing();//初始化液晶屏
void sendbyte(uchar ch);//数码管显示一个数
void display(uchar n);//数码管显示
void delay1();//短延时
void delay2();//长延时
void delay(uint x){//延时空操作
while(x--) ;
}
```

```
void main()
{
init();//数码管和中断初始化
init_yejing();//初始化液晶屏
motor=0;//p1.1 输出口
while(1)
{
clearscreen();
send_all(1,3,speedLow/100);//最低值百位
send_all(1,4,(speedLow/10)%10);//最低值十位
send_all(1,5,speedLow%10);//最低值个位
send_all(3,3,cspeed/100);//当前值百位
send_all(3,4,(cspeed/10)%10);//当前值十位
send_all(3,5,cspeed%10);//当前值个位
send_all(5,3,speedUp/100);//最高值百位
send_all(5,4,(speedUp/10)%10);//最高值十位
send_all(5,5,speedUp%10);//最高值个位
delay1();//短延时
display(cspeed);//数码管显示 delay(50000);
}
}
//数码管和中断初始化
void init()
{
P4SW=0x30;//P4SW 寄存器值设置为 0x30
IT0=1;//1 跳变沿触发方式，0 电平触发方式
EA=1;//中断使能
EX0=1;//INT0 的中断允许位 为 1 允许
ET1=1;//T1 中断允许位 为 1 时允许响应
ET0=1;//T0 中断允许位 为 1 时允许响应
TMOD=0x11; //16 位寄存器，模式 1
TH1=0x3C;
TL1=0xB0; //50ms:65536-50000=15536
TH0=0xFF;
TL0=0x9C; //0.1ms:65536-100=65436
TR0=1;//将计时器 0 置高，定时器 0 启动，允许计数，调到中断地址
TR1=1;//1
}
//外部中断 INT0,转一圈加一，通过测量一秒钟之内发生的中断次数，来测量转速
//单片机中 51 系列的有 0 1 2 3 4 等几个中断, 中断号与中断事件是绑定的,不能随便设置,对应的中断向量会指向这个函数入口地址
void ex_int0() interrupt 0//interrupt 表明当前是一个中断函数,不需要被主函数直接或间接调用.interrupt 后的数字表明是中断号几
```

```
{
tspeed++;//脉冲计数加一，表示转了一圈，用来测速
}
//计时器中断 0，用来调整占空比（比较快速）
//动态改变 P1.1 的输出，使得宏观上输出有效（
0）的比例就是预定的控制变量
//控制变量变高，转速就随之变大
void t0_int() interrupt 1 ///0.1ms
{
TH0=0xFF;
TL0=0x9C;
//累加进位法，将总的周期内的 0,1 均匀分散开
X+=N;
//设置一个累加变量 X，每次加 N
//若结果大于 M，则输出 0，并减去 M
if(X>M)
{
motor=0;
X-=M;
}
//否则输出 1
elsemotor=1;
//这样使得整体的占空比为 N/M
}
//计时器中断 1，用来调整转速
void t1_int() interrupt 3 ///50ms
{
//计数 20 次使得每 1s 中断测量当前转速
if(++t1_cnt<20)
{
TH1=0x3C;
TL1=0xB0;
if(swh1==0)//S1 按下
{
xspeed = speedLow;//预定转速改为最低转速
}
if(swh2==0)//S2 按下
{
xspeed = speedUp;//预定转速改为最高转速
}
return;
}
t1_cnt=0;//计数二十次以得到 1s
cspeed=tspeed;//当前转速修改位脉冲计数得出的转速
```

```
tspeed=0;//脉冲计数清零
//与预定转速相比，通过修改当前转速的值来稳定转速
if(cspeed>xspeed) N--;//降低转速
if(cspeed<xspeed) N++;//提高转速
}
//液晶屏初始化
void init_yejing()
{
send_byte(192,1,1);//设置起始行，规定了显示屏上最顶一行所对应的显
示存储器的行地址，默认格式的最高两位是 1，所以是在 192 的基础上加
send_byte(63,1,1);//打开显示开关显示开关设置，默认格式为 0011111D，
D 为 1 时候显示，为 0 不显示
}
//送 8 位数
void send_byte(uchar dat,uchar cs1,uchar cs2)
{
P2=0xff;
CS1=cs1; CS2=cs2;
RS=0; RW=1; E=1;//读状态字
while(BUSY) ;//判断是否在忙，忙则等待
//送数据或和控制字
E=0;
RS=!(cs1&&cs2),RW=0;//写指令代码
P2=dat;
E=1;//在 E 的上升沿写显示数据
delay(3);E=0;//总线释放
CS1=CS2=0;
}
//写液晶屏，显示相应字
void send_all(uint page,uint lie,uint offset)
{
uint i,j,k=0;
for(i=0;i<2;++i)
{
send_byte(184+i+page,1,1);//page=0xb8|page;//选择页面 184-页
面地址设置，也就是 X 的设置，默认格式的高五位是 10111，所以是在 184 的基
础
上加
send_byte(64+lie*16-(lie>3)*64,1,1);//选择列号，也就是 Y 的设置，
默认格式中最高两位是 01，所以是在 64 的基础上加，Y 是自动加一的
for(j=0;j<16;++j)
send_byte(zima[offset][k++],lie<4,lie>=4);//送数
}
}
```

```
//清屏
void clearscreen()
{
int i,j;
for(i=0;i<8;++i)
{
send_byte(184+i,1,1);
send_byte(64,1,1);
for(j=0;j<64;++j)
{
send_byte(0x00,0,1);
send_byte(0x00,1,0);
}
}
}
//数码管显示 1 个数
void sendbyte(uchar ch)
{
uchar shape,c;
shape=tab[ch];//取出段码表中的数码管对应数字
for(c=0;c<8;c++)
{
sclk=0;//P4.4 串口时钟 当有上升沿，会将数据传过去
sdata=shape & 0x80;//P4.5 串口数据 串口的要求是数据一位一位的
传 所以在这里还是通过移位 做逻辑与 取一位数据
sclk=1;//制造上升沿，先传一位数据
shape <<= 1;//左移一
}
}
//数码管显示
void display(uchar n){
sendbyte(n%10);
sendbyte((n/10)%10);
sendbyte(n/100);
}
//短延时
void delay1()
{
int i,j;
for(i=0;i<1000;i++)
for(j=0;j<500;j++);
}
//长延时
void delay2()
```

```
{
int i,j;
for(i=0;i<1000;i++)
for(j=0;j<1000;j++);
}
```

六、思考题

1. 讨论脉宽调速和电压调速的区别、优缺点和应用范围。

区别：脉宽调速和电压调速的不同之处在于电压调速指的是调节直流电机端电压来改变转速，供电是连续的，可以做到无级调速，而脉宽调速指的是通过开关量输出达到模拟量输出效果，宏观看是一个匀速的过程，而微观看是非匀速的过程，脉宽调速通过改变信号的占空比来调节宏观上的输出电压进而改变转速。

优缺点：电压调速优点在于可以实现从使动电压到额定电压范围内的无级调速，调速范围大，点击运行平稳，缺点是需要外围器件多（如调压器等）且调压过程中转差功率损耗大，效率较低。脉宽调速的优点在于需要外围器件较少，效率高，缺点是调速范围没有电压调速大，当直流电机转速低时噪音较大。

应用范围：电压调速适用于对调速范围和平稳运行有特殊要求，调速精度要求不高，且对效率影响不大的情况下。脉宽调速适用于对噪音要求不高且采用电压调速对效率影响较大的情况下。此外，在交流电机中， PWM 也具有调节频率的作用。

2. 说明程序原理中累加进位法的正确性。

我们将整个周期分为 M 份，那么累加进位法在一个周期中需要累加并判断 M 次。又因为每一次判断前累加变量 X 都需要加 N，因此如果不进行减法操作，一周期后 X 的值为 N×M。因为累加进位法判断的逻辑为： X 大于等于 M 则减去 M，否则不减，所以我们可以将该过程转换为除法，即在一个周期中，X 减去 M 的操作发生了 N×M÷M=N 次。又因为每次 X 的值大于等于 M 时都会使电机停转，所以电机停转的次数为一周期中 X 减去 M 的次数，即 N 次，因此整体的占空比为 N/M

3. 计算转速测量的最大可能误差，讨论减少误差的办法。

①在使用定时器中断时，我们会对某些变量进行初始化，这些机器指令执行的周期的时间不会算在在中断计时中。同时，电机转动 1 周触发 1 次中断，本实验是通过对 1s 触发的中断进行计数来间接得到转速。我们可以在电机转动 1 圈时，多设置几次中断。

②缩短累加进位法中断的计时间隔，以及在计秒中断中每个小循环都添加对于当前速度的监测并随之更改 N 的值。

③可以增加齿轮齿数来减少误差。