

LC3100 源程序

```

#include "Show.h"

/*****全局变量声明*****/
float Twinkle_Time;//闪烁时间

/*****局部变量声明*****/
uint8_t Time_ShowFlag,Speed_ShowFlag,Temp_ShowFlag;//时间、速度、温度显示的标志位 0:
常亮 1: 熄灭

/*
*****
* 函数原型: void Show_Xg(float dT)
* 功    能: 显示离心力
*****
*/
void Show_Xg(float dT)
{
    float R;
    if(Speed_Mode == 0)
        return;
    switch(Speed_Mode)
    {
        case 1:R = 11;//15ml*8
            break;
        case 2:R = 11.6;//15ml*12
            break;
        case 3:R = 12.2;//10ml*18
            break;
        case 4:R = 12.1;//10ml*24
            break;
        case 5:R = 12;//50ml*6
            break;
        default:
            break;
    }
    Display_Xg = (uint32_t)(11.18f * (((float)Set_Speed/1000) * ((float)Set_Speed/1000)) *
(float)R);
}

/*
*****
* 函数原型: void Start_CircleRun(float dT)
* 功    能: 外框跑圈动画
*****
*/
void Start_CircleRun(float dT)
{
    static float T;
    if(Circle_Run == 0)
    {

```

```

        PModeIcon_Step = 0;
        T = 0;
        return;
    }
    T += dT;
    if(T >= 0.5)
    {
        PModeIcon_Step++;
        if(PModeIcon_Step > 18)
            PModeIcon_Step = 7;
        T = 0;
    }
}

/*
*****
* 函数原型: void Start_SpeedRun(float dT)
* 功    能: 转速动画
*****
*/
void Start_SpeedRun(float dT)
{
    static float T;
    if(sys.Run_Status == 0)
    {
        SpeedIcon_Step = 0;
        T = 0;
        return;
    }
    T += dT;
    if(T >= 0.5)
    {
        SpeedIcon_Step++;
        if(SpeedIcon_Step > 6)
            SpeedIcon_Step = 1;
        T = 0;
    }
}

/*
*****
* 函数原型: void Check_Safe_Mode(float dT)
* 功    能: 安全温度图标闪烁检测
*****
*/
void Check_Safe_Mode(float dT)
{
    if(Safe_Temp_Twinkleg)
    {
        Safe_Mode = ~Safe_Mode;//安全温度图标闪烁
    }
}

```

```

    }
    else
    {
        if(sys.Run_Status == 0)
        {
            if(Safe_Set_Temp == 500)
                Safe_Mode = 0;
            else
                Safe_Mode = 1;
        }
    }
}

/*
*****
* 函数原型: void Check_ShowFlag(float dT)
* 功    能: 闪烁检测
* 输    入: dT:执行周期
* 参    数: float dT
*****
*/
void Check_ShowFlag(float dT)
{
    if(sys.SetMode_Option == 0 || Speed_Mode)//如果没在设置选项中, 则都点亮, 不闪烁
    {
        Speed_ShowFlag = 0;//常亮
        Temp_ShowFlag = 0;//常亮
        Time_ShowFlag = 0;//常亮
        Twinkle_Time = 0;//闪烁计时清零
        return;
    }
    if(Twinkle_Time && Key_Status==0)//闪烁和没有操作按键时
    {
        Twinkle_Time -= dT;//闪烁计时
        if(sys.SetMode_Option == 1)//设置时间
        {
            Time_ShowFlag = ~Time_ShowFlag;//时间闪烁
            Speed_ShowFlag = 0;//速度常亮
            Temp_ShowFlag = 0;//温度常亮
        }
        else if(sys.SetMode_Option == 2)//设置速度
        {
            Time_ShowFlag = 0;//时间常亮
            Speed_ShowFlag = ~Speed_ShowFlag;//速度闪烁
            Temp_ShowFlag = 0;//温度常亮
        }
        else if(sys.SetMode_Option == 3)//设置温度
        {
            Safe_Mode = 1;//点亮"SAFE"图标
            Speed_ShowFlag = 0;//速度常亮

```

```

        Time_ShowFlag = 0;//时间常亮
        Temp_ShowFlag = ~Temp_ShowFlag;//温度闪烁
    }
    if(Twinkle_Time == 0)//如果闪烁结束
    {
        sys.SetMode_Option = 0;//模式选择清零
    }
}
else
{
    Speed_ShowFlag = 0;//常亮
    Temp_ShowFlag = 0;//常亮
    Time_ShowFlag = 0;//常亮
}
}

/*
*****
* 函数原型: void Display_Time(int32_t dis_set_time,int32_t dis_rel_time)
* 功    能: 显示时间
* 输    入: dis_set_time 设定时间  dis_rel_time 实际时间
* 参    数: int32_t dis_set_time,int32_t dis_rel_time
*****
*/
void Display_Time(int32_t dis_set_time,int32_t dis_rel_time)
{
    uint8_t seg1,seg2,seg3,seg4,seg5,seg6,seg7,seg8,seg9;
    seg1=0;seg2=0;seg3=0;seg4=0;seg5=0;seg6=0;seg7=0;seg8=0;seg9=0;
    uint8_t SH,H,SM,M;//时间的单位取值

    /*****设定时间计算*****/
    if(dis_set_time > 3599)//如果设定时间大于 59.59 分钟时
        Time_Unit=1;//单位变成分
    else
        Time_Unit=0;//不然就是秒

    if(Time_Unit == 1)
    {
        SH=dis_set_time/3600/10;//计算十位单位的小时数
        H=dis_set_time/3600%10;//计算个位单位的小时数
        SM=dis_set_time%3600/60/10;//计算十分位单位的分钟数
        M=dis_set_time%3600/60%10;//计算个分位单位的分钟数
    }
    else
    {
        SH=dis_set_time%3600/60/10;//计算十位单位的分钟数
        H=dis_set_time%3600/60%10;//计算个位单位的分钟数
        SM=dis_set_time%60/10;//计算十分位单位的秒钟数
        M=dis_set_time%60%10;//计算十分位单位的秒钟数
    }
}

```

```

if(Time_ShowFlag == 0)
{
    /******set_time 十小时位******/
    switch(SH)
    {
        case 0:seg2&=0x0F;seg3&=0x8F;seg2|=0xF0;seg3|=0x50;//数字 0
            break;
        case 1:seg2&=0x0F;seg3&=0x8F;seg2|=0x00;seg3|=0x50;//数字 1
            break;
        case 2:seg2&=0x0F;seg3&=0x8F;seg2|=0xD0;seg3|=0x30;//数字 2
            break;
        case 3:seg2&=0x0F;seg3&=0x8F;seg2|=0x90;seg3|=0x70;//数字 3
            break;
        case 4:seg2&=0x0F;seg3&=0x8F;seg2|=0x20;seg3|=0x70;//数字 4
            break;
        case 5:seg2&=0x0F;seg3&=0x8F;seg2|=0xB0;seg3|=0x60;//数字 5
            break;
        case 6:seg2&=0x0F;seg3&=0x8F;seg2|=0xF0;seg3|=0x60;//数字 6
            break;
        case 7:seg2&=0x0F;seg3&=0x8F;seg2|=0x10;seg3|=0x50;//数字 7
            break;
        case 8:seg2&=0x0F;seg3&=0x8F;seg2|=0xF0;seg3|=0x70;//数字 8
            break;
        case 9:seg2&=0x0F;seg3&=0x8F;seg2|=0xB0;seg3|=0x70;//数字 9
            break;
        default:
            break;
    }
    /******set_time 小时位******/
    switch(H)
    {
        case 0:seg4&=0x0F;seg5&=0x8F;seg4|=0xF0;seg5|=0x50;//数字 0
            break;
        case 1:seg4&=0x0F;seg5&=0x8F;seg4|=0x00;seg5|=0x50;//数字 1
            break;
        case 2:seg4&=0x0F;seg5&=0x8F;seg4|=0xD0;seg5|=0x30;//数字 2
            break;
        case 3:seg4&=0x0F;seg5&=0x8F;seg4|=0x90;seg5|=0x70;//数字 3
            break;
        case 4:seg4&=0x0F;seg5&=0x8F;seg4|=0x20;seg5|=0x70;//数字 4
            break;
        case 5:seg4&=0x0F;seg5&=0x8F;seg4|=0xB0;seg5|=0x60;//数字 5
            break;
        case 6:seg4&=0x0F;seg5&=0x8F;seg4|=0xF0;seg5|=0x60;//数字 6
            break;
        case 7:seg4&=0x0F;seg5&=0x8F;seg4|=0x10;seg5|=0x50;//数字 7
            break;
        case 8:seg4&=0x0F;seg5&=0x8F;seg4|=0xF0;seg5|=0x70;//数字 8
            break;
        case 9:seg4&=0x0F;seg5&=0x8F;seg4|=0xB0;seg5|=0x70;//数字 9

```

```

        break;
    default:
        break;
}

/*****set_time 十分位*****/
switch(SM%10)
{
    case 0:seg6&=0x0F;seg7&=0x8F;seg6|=0xF0;seg7|=0x50;//数字 0
        break;
    case 1:seg6&=0x0F;seg7&=0x8F;seg6|=0x00;seg7|=0x50;//数字 1
        break;
    case 2:seg6&=0x0F;seg7&=0x8F;seg6|=0xD0;seg7|=0x30;//数字 2
        break;
    case 3:seg6&=0x0F;seg7&=0x8F;seg6|=0x90;seg7|=0x70;//数字 3
        break;
    case 4:seg6&=0x0F;seg7&=0x8F;seg6|=0x20;seg7|=0x70;//数字 4
        break;
    case 5:seg6&=0x0F;seg7&=0x8F;seg6|=0xB0;seg7|=0x60;//数字 5
        break;
    case 6:seg6&=0x0F;seg7&=0x8F;seg6|=0xF0;seg7|=0x60;//数字 6
        break;
    case 7:seg6&=0x0F;seg7&=0x8F;seg6|=0x10;seg7|=0x50;//数字 7
        break;
    case 8:seg6&=0x0F;seg7&=0x8F;seg6|=0xF0;seg7|=0x70;//数字 8
        break;
    case 9:seg6&=0x0F;seg7&=0x8F;seg6|=0xB0;seg7|=0x70;//数字 9
        break;
    default:
        break;
}

/*****set_time 分位*****/
switch(M%10)
{
    case 0:seg8&=0x0F;seg9&=0x8F;seg8|=0xF0;seg9|=0x50;//数字 0
        break;
    case 1:seg8&=0x0F;seg9&=0x8F;seg8|=0x00;seg9|=0x50;//数字 1
        break;
    case 2:seg8&=0x0F;seg9&=0x8F;seg8|=0xD0;seg9|=0x30;//数字 2
        break;
    case 3:seg8&=0x0F;seg9&=0x8F;seg8|=0x90;seg9|=0x70;//数字 3
        break;
    case 4:seg8&=0x0F;seg9&=0x8F;seg8|=0x20;seg9|=0x70;//数字 4
        break;
    case 5:seg8&=0x0F;seg9&=0x8F;seg8|=0xB0;seg9|=0x60;//数字 5
        break;
    case 6:seg8&=0x0F;seg9&=0x8F;seg8|=0xF0;seg9|=0x60;//数字 6
        break;
    case 7:seg8&=0x0F;seg9&=0x8F;seg8|=0x10;seg9|=0x50;//数字 7

```

```

        break;
    case 8:seg8&=0x0F;seg9&=0x8F;seg8|=0xF0;seg9|=0x70;//数字 8
        break;
    case 9:seg8&=0x0F;seg9&=0x8F;seg8|=0xB0;seg9|=0x70;//数字 9
        break;
    default:
        break;
    }
}
else
{
    seg2&=0x0F;seg3&=0x8F;seg2|=0x00;seg3|=0x00;//不显示
    seg4&=0x0F;seg5&=0x8F;seg4|=0x00;seg5|=0x00;//不显示
    seg6&=0x0F;seg7&=0x8F;seg6|=0x00;seg7|=0x00;//不显示
    seg8&=0x0F;seg9&=0x8F;seg8|=0x00;seg9|=0x00;//不显示
}
/*****实际时间计算*****/
if(Time_Unit == 1)
{
    SH=dis_rel_time/3600/10;//计算十位单位的小时数
    H=dis_rel_time/3600%10;//计算个位单位的小时数
    SM=dis_rel_time%3600/60/10;//计算十分位单位的分钟数
    M=dis_rel_time%3600/60%10;//计算个分位单位的分钟数
}
else
{
    SH=dis_rel_time%3600/60/10;//计算十位单位的分钟数
    H=dis_rel_time%3600/60%10;//计算个位单位的分钟数
    SM=dis_rel_time%60/10;//计算十分位单位的秒钟数
    M=dis_rel_time%60%10;//计算十分位单位的秒钟数
}
/*****dis_rel_time 十小时位*****/
switch(SH)
{
    case 0:seg1&=0xF0;seg2&=0xF1;seg1|=0x0F;seg2|=0x0A;//数字 0
        break;
    case 1:seg1&=0xF0;seg2&=0xF1;seg1|=0x00;seg2|=0x0A;//数字 1
        break;
    case 2:seg1&=0xF0;seg2&=0xF1;seg1|=0x0D;seg2|=0x06;//数字 2
        break;
    case 3:seg1&=0xF0;seg2&=0xF1;seg1|=0x09;seg2|=0x0E;//数字 3
        break;
    case 4:seg1&=0xF0;seg2&=0xF1;seg1|=0x02;seg2|=0x0E;//数字 4
        break;
    case 5:seg1&=0xF0;seg2&=0xF1;seg1|=0x0B;seg2|=0x0D;//数字 5
        break;
    case 6:seg1&=0xF0;seg2&=0xF1;seg1|=0x0F;seg2|=0x0D;//数字 6
        break;
    case 7:seg1&=0xF0;seg2&=0xF1;seg1|=0x01;seg2|=0x0A;//数字 7
        break;
}

```



```

        case 8:seg1&=0xF0;seg2&=0xF1;seg1|=0x0F;seg2|=0x0E;//数字 8
            break;
        case 9:seg1&=0xF0;seg2&=0xF1;seg1|=0x0B;seg2|=0x0E;//数字 9
            break;
        default:
            break;
    }

    /*****dis_rel_time 小时位*****/
    switch(H)
    {
        case 0:seg3&=0xF0;seg4&=0xF1;seg3|=0x0F;seg4|=0x0A;//数字 0
            break;
        case 1:seg3&=0xF0;seg4&=0xF1;seg3|=0x00;seg4|=0x0A;//数字 1
            break;
        case 2:seg3&=0xF0;seg4&=0xF1;seg3|=0x0D;seg4|=0x06;//数字 2
            break;
        case 3:seg3&=0xF0;seg4&=0xF1;seg3|=0x09;seg4|=0x0E;//数字 3
            break;
        case 4:seg3&=0xF0;seg4&=0xF1;seg3|=0x02;seg4|=0x0E;//数字 4
            break;
        case 5:seg3&=0xF0;seg4&=0xF1;seg3|=0x0B;seg4|=0x0D;//数字 5
            break;
        case 6:seg3&=0xF0;seg4&=0xF1;seg3|=0x0F;seg4|=0x0D;//数字 6
            break;
        case 7:seg3&=0xF0;seg4&=0xF1;seg3|=0x01;seg4|=0x0A;//数字 7
            break;
        case 8:seg3&=0xF0;seg4&=0xF1;seg3|=0x0F;seg4|=0x0E;//数字 8
            break;
        case 9:seg3&=0xF0;seg4&=0xF1;seg3|=0x0B;seg4|=0x0E;//数字 9
            break;
        default:
            break;
    }

    /*****dis_rel_time 十分位*****/
    switch(SM%10)
    {
        case 0:seg5&=0xF0;seg6&=0xF1;seg5|=0x0F;seg6|=0x0A;//数字 0
            break;
        case 1:seg5&=0xF0;seg6&=0xF1;seg5|=0x00;seg6|=0x0A;//数字 1
            break;
        case 2:seg5&=0xF0;seg6&=0xF1;seg5|=0x0D;seg6|=0x06;//数字 2
            break;
        case 3:seg5&=0xF0;seg6&=0xF1;seg5|=0x09;seg6|=0x0E;//数字 3
            break;
        case 4:seg5&=0xF0;seg6&=0xF1;seg5|=0x02;seg6|=0x0E;//数字 4
            break;
        case 5:seg5&=0xF0;seg6&=0xF1;seg5|=0x0B;seg6|=0x0D;//数字 5
            break;
    }

```

```

        case 6:seg5&=0xF0;seg6&=0xF1;seg5|=0x0F;seg6|=0x0D;//数字 6
            break;
        case 7:seg5&=0xF0;seg6&=0xF1;seg5|=0x01;seg6|=0x0A;//数字 7
            break;
        case 8:seg5&=0xF0;seg6&=0xF1;seg5|=0x0F;seg6|=0x0E;//数字 8
            break;
        case 9:seg5&=0xF0;seg6&=0xF1;seg5|=0x0B;seg6|=0x0E;//数字 9
            break;
        default:
            break;
    }

    /*****dis_rel_time 分位*****/
    switch(M%10)
    {
        case 0:seg7&=0xF0;seg8&=0xF1;seg7|=0x0F;seg8|=0x0A;//数字 0
            break;
        case 1:seg7&=0xF0;seg8&=0xF1;seg7|=0x00;seg8|=0x0A;//数字 1
            break;
        case 2:seg7&=0xF0;seg8&=0xF1;seg7|=0x0D;seg8|=0x06;//数字 2
            break;
        case 3:seg7&=0xF0;seg8&=0xF1;seg7|=0x09;seg8|=0x0E;//数字 3
            break;
        case 4:seg7&=0xF0;seg8&=0xF1;seg7|=0x02;seg8|=0x0E;//数字 4
            break;
        case 5:seg7&=0xF0;seg8&=0xF1;seg7|=0x0B;seg8|=0x0D;//数字 5
            break;
        case 6:seg7&=0xF0;seg8&=0xF1;seg7|=0x0F;seg8|=0x0D;//数字 6
            break;
        case 7:seg7&=0xF0;seg8&=0xF1;seg7|=0x01;seg8|=0x0A;//数字 7
            break;
        case 8:seg7&=0xF0;seg8&=0xF1;seg7|=0x0F;seg8|=0x0E;//数字 8
            break;
        case 9:seg7&=0xF0;seg8&=0xF1;seg7|=0x0B;seg8|=0x0E;//数字 9
            break;
        default:
            break;
    }

    /*****时间冒号图标*****/
    seg4&=0xFE; seg4|=0x01;//实际时间冒号
    seg5&=0x7F; seg5|=0x80;//设定时间冒号

    if(Time_Unit==1)//时间单位切换
    {
        seg9&=0xF7;seg9|=0x08;//显示 min
    }
    else
    {
        seg7&=0x7F;seg7|=0x80;//显示 sec
    }

```

```

}

seg9&=0xFD;seg9|=0x02;
/*****关闭盖子图标*****/
if(Lid_State == 0)
{
    seg9&=0xFE;seg9|=0x01;
}
/*****/

/*****时间图标*****/
seg9&=0xFB;seg9|=0x04;

/*****/

Write_Addr_Dat_N(0, seg1,1);//SEG29
Write_Addr_Dat_N(2, seg2,1);//SEG28
Write_Addr_Dat_N(4, seg3,1);//SEG27
Write_Addr_Dat_N(6, seg4,1);//SEG26
Write_Addr_Dat_N(8, seg5,1);//SEG25
Write_Addr_Dat_N(10, seg6,1);//SEG24
Write_Addr_Dat_N(12, seg7,1);//SEG23
Write_Addr_Dat_N(14, seg8,1);//SEG22
Write_Addr_Dat_N(16, seg9,1);//SEG21
}

/*
*****
* 函数原型: void Display_Speed(int16_t dis_set_speed,int16_t dis_rel_speed)
* 功    能: 显示转速
* 输    入: dis_set_speed 设定转速  dis_rel_speed 实际转速
* 参    数: int16_t dis_set_speed,int16_t dis_rel_speed
*****
*/
void Display_Speed(int16_t dis_set_speed,int16_t dis_rel_speed)
{
    uint8_t seg10,seg11,seg12,seg13,seg14,seg15,seg16,seg17,seg18,seg19,seg20,seg21;
    seg10=0;seg11=0;seg12=0;seg13=0;seg14=0;seg15=0;seg16=0;seg17=0;seg18=0;seg19=0;seg20=0;seg21=0;
    uint16_t Val;//用于百十个取出来的数字

    /*****设定转速计算*****/
    if(Speed_ShowFlag == 0)
    {
        /*****dis_set_speed 千位*****/
        if(dis_set_speed > 9999)//大于 9999 时
        {
            seg20&=0x7F;seg20|=0x80;//显示 1
        }
    }

```

```

else//小于 9999 时
{
    seg20&=0x7F;seg20|=0x00;//不显示
}

/*****dis_set_speed 千位*****/
if(dis_set_speed > 999)//大于 999 时
{
    Val=dis_set_speed/1000;//取出千位
    if(dis_set_speed > 9999)//大于 9999 时
        Val=Val%10;//取出千位
    switch(Val)
    {
        case 0:seg20&=0x8F;seg21&=0x0F;seg20|=0x50;seg21|=0xF0;//数字 0
            break;
        case 1:seg20&=0x8F;seg21&=0x0F;seg20|=0x50;seg21|=0x00;//数字 1
            break;
        case 2:seg20&=0x8F;seg21&=0x0F;seg20|=0x30;seg21|=0xD0;//数字 2
            break;
        case 3:seg20&=0x8F;seg21&=0x0F;seg20|=0x70;seg21|=0x90;//数字 3
            break;
        case 4:seg20&=0x8F;seg21&=0x0F;seg20|=0x70;seg21|=0x20;//数字 4
            break;
        case 5:seg20&=0x8F;seg21&=0x0F;seg20|=0x60;seg21|=0xB0;//数字 5
            break;
        case 6:seg20&=0x8F;seg21&=0x0F;seg20|=0x60;seg21|=0xF0;//数字 6
            break;
        case 7:seg20&=0x8F;seg21&=0x0F;seg20|=0x50;seg21|=0x10;//数字 7
            break;
        case 8:seg20&=0x8F;seg21&=0x0F;seg20|=0x70;seg21|=0xF0;//数字 8
            break;
        case 9:seg20&=0x8F;seg21&=0x0F;seg20|=0x70;seg21|=0xB0;//数字 9
            break;
        default:
            break;
    }
}
else
{
    seg20&=0x8F;seg21&=0x0F;seg20|=0x00;seg21|=0x00;//不显示
}

/*****dis_set_speed 百位*****/
if(dis_set_speed > 99)//大于 99 时
{
    Val=dis_set_speed/100;//取出百位
    if(dis_set_speed > 999)//大于 999 时
        Val=Val%10;//取出百位
    switch(Val)
    {

```

```

        case 0:seg15&=0x8F;seg14&=0x0F;seg15|=0x50;seg14|=0xF0;//数字 0
            break;
        case 1:seg15&=0x8F;seg14&=0x0F;seg15|=0x50;seg14|=0x00;//数字 1
            break;
        case 2:seg15&=0x8F;seg14&=0x0F;seg15|=0x30;seg14|=0xD0;//数字 2
            break;
        case 3:seg15&=0x8F;seg14&=0x0F;seg15|=0x70;seg14|=0x90;//数字 3
            break;
        case 4:seg15&=0x8F;seg14&=0x0F;seg15|=0x70;seg14|=0x20;//数字 4
            break;
        case 5:seg15&=0x8F;seg14&=0x0F;seg15|=0x60;seg14|=0xB0;//数字 5
            break;
        case 6:seg15&=0x8F;seg14&=0x0F;seg15|=0x60;seg14|=0xF0;//数字 6
            break;
        case 7:seg15&=0x8F;seg14&=0x0F;seg15|=0x50;seg14|=0x10;//数字 7
            break;
        case 8:seg15&=0x8F;seg14&=0x0F;seg15|=0x70;seg14|=0xF0;//数字 8
            break;
        case 9:seg15&=0x8F;seg14&=0x0F;seg15|=0x70;seg14|=0xB0;//数字 9
            break;
        default:
            break;
    }
}
else
{
    seg15&=0x8F;seg14&=0x0F;seg15|=0x00;seg14|=0x00;//不显示
}
/*****dis_set_speed 十位*****/
if(dis_set_speed > 9)//大于 9 时
{
    Val=dis_set_speed/10;//取出十位
    if(dis_set_speed > 99)//大于 99 时
        Val=Val%10;//取出十位
    switch(Val)
    {
        case 0:seg17&=0x8F;seg16&=0x0F;seg17|=0x50;seg16|=0xF0;//数字 0
            break;
        case 1:seg17&=0x8F;seg16&=0x0F;seg17|=0x50;seg16|=0x00;//数字 1
            break;
        case 2:seg17&=0x8F;seg16&=0x0F;seg17|=0x30;seg16|=0xD0;//数字 2
            break;
        case 3:seg17&=0x8F;seg16&=0x0F;seg17|=0x70;seg16|=0x90;//数字 3
            break;
        case 4:seg17&=0x8F;seg16&=0x0F;seg17|=0x70;seg16|=0x20;//数字 4
            break;
        case 5:seg17&=0x8F;seg16&=0x0F;seg17|=0x60;seg16|=0xB0;//数字 5
            break;
        case 6:seg17&=0x8F;seg16&=0x0F;seg17|=0x60;seg16|=0xF0;//数字 6
            break;
    }
}

```

```

        case 7:seg17&=0x8F;seg16&=0x0F;seg17|=0x50;seg16|=0x10;//数字 7
            break;
        case 8:seg17&=0x8F;seg16&=0x0F;seg17|=0x70;seg16|=0xF0;//数字 8
            break;
        case 9:seg17&=0x8F;seg16&=0x0F;seg17|=0x70;seg16|=0xB0;//数字 9
            break;
        default:
            break;
    }
}
else
{
    seg17&=0x8F;seg16&=0x0F;seg17|=0x00;seg16|=0x00;//不显示
}
/*****dis_set_speed 个位*****/
Val=dis_set_speed%10;//取出个位
switch(Val)
{
    case 0:seg19&=0x8F;seg18&=0x0F;seg19|=0x50;seg18|=0xF0;//数字 0
        break;
    case 1:seg19&=0x8F;seg18&=0x0F;seg19|=0x50;seg18|=0x00;//数字 1
        break;
    case 2:seg19&=0x8F;seg18&=0x0F;seg19|=0x30;seg18|=0xD0;//数字 2
        break;
    case 3:seg19&=0x8F;seg18&=0x0F;seg19|=0x70;seg18|=0x90;//数字 3
        break;
    case 4:seg19&=0x8F;seg18&=0x0F;seg19|=0x70;seg18|=0x20;//数字 4
        break;
    case 5:seg19&=0x8F;seg18&=0x0F;seg19|=0x60;seg18|=0xB0;//数字 5
        break;
    case 6:seg19&=0x8F;seg18&=0x0F;seg19|=0x60;seg18|=0xF0;//数字 6
        break;
    case 7:seg19&=0x8F;seg18&=0x0F;seg19|=0x50;seg18|=0x10;//数字 7
        break;
    case 8:seg19&=0x8F;seg18&=0x0F;seg19|=0x70;seg18|=0xF0;//数字 8
        break;
    case 9:seg19&=0x8F;seg18&=0x0F;seg19|=0x70;seg18|=0xB0;//数字 9
        break;
    default:
        break;
}
}
else
{
    seg20&=0x7F;seg20|=0x00;//不显示
    seg20&=0x8F;seg21&=0x0F;seg20|=0x00;seg21|=0x00;//不显示
    seg15&=0x8F;seg14&=0x0F;seg15|=0x00;seg14|=0x00;//不显示
    seg17&=0x8F;seg16&=0x0F;seg17|=0x00;seg16|=0x00;//不显示
    seg19&=0x8F;seg18&=0x0F;seg19|=0x00;seg18|=0x00;//不显示
}
}

```

```

/*****实际转速计算*****/
if(Speed_Mode == 0)
{
    /*****dis_rel_speed 万位*****/
    if(dis_rel_speed > 9999)//大于 9999 时
    {
        Val=dis_rel_speed/10000;//取出万位
        switch(Val)
        {
            case 0:seg12&=0xF0;seg13&=0xF1;seg12|=0x0F;seg13|=0x0A;//数字 0
                break;
            case 1:seg12&=0xF0;seg13&=0xF1;seg12|=0x00;seg13|=0x0A;//数字 1
                break;
            case 2:seg12&=0xF0;seg13&=0xF1;seg12|=0x0D;seg13|=0x06;//数字 2
                break;
            case 3:seg12&=0xF0;seg13&=0xF1;seg12|=0x09;seg13|=0x0E;//数字 3
                break;
            case 4:seg12&=0xF0;seg13&=0xF1;seg12|=0x02;seg13|=0x0E;//数字 4
                break;
            case 5:seg12&=0xF0;seg13&=0xF1;seg12|=0x0B;seg13|=0x0C;//数字 5
                break;
            case 6:seg12&=0xF0;seg13&=0xF1;seg12|=0x0F;seg13|=0x0C;//数字 6
                break;
            case 7:seg12&=0xF0;seg13&=0xF1;seg12|=0x01;seg13|=0x0A;//数字 7
                break;
            case 8:seg12&=0xF0;seg13&=0xF1;seg12|=0x0F;seg13|=0x0E;//数字 8
                break;
            case 9:seg12&=0xF0;seg13&=0xF1;seg12|=0x0B;seg13|=0x0E;//数字 9
                break;
            default:
                break;
        }
    }
    else//小于 9999 时
    {
        seg12&=0xF0;seg13&=0xF1;seg12|=0x00;seg13|=0x00;//不显示
    }
    /*****dis_rel_speed 千位*****/
    if(dis_rel_speed > 999)//大于 999 时
    {
        Val=dis_rel_speed/1000;//取出千位
        if(dis_rel_speed > 9999)//大于 9999 时
            Val=Val%10;//取出千位
        switch(Val)
        {
            case 0:seg21&=0xF0;seg20&=0xF1;seg21|=0x0F;seg20|=0x0A;//数字 0
                break;
            case 1:seg21&=0xF0;seg20&=0xF1;seg21|=0x00;seg20|=0x0A;//数字 1
                break;
            case 2:seg21&=0xF0;seg20&=0xF1;seg21|=0x0D;seg20|=0x06;//数字 2

```

```

        break;
    case 3:seg21&=0xF0;seg20&=0xF1;seg21|=0x09;seg20|=0x0E;//数字 3
        break;
    case 4:seg21&=0xF0;seg20&=0xF1;seg21|=0x02;seg20|=0x0E;//数字 4
        break;
    case 5:seg21&=0xF0;seg20&=0xF1;seg21|=0x0B;seg20|=0x0C;//数字 5
        break;
    case 6:seg21&=0xF0;seg20&=0xF1;seg21|=0x0F;seg20|=0x0C;//数字 6
        break;
    case 7:seg21&=0xF0;seg20&=0xF1;seg21|=0x01;seg20|=0x0A;//数字 7
        break;
    case 8:seg21&=0xF0;seg20&=0xF1;seg21|=0x0F;seg20|=0x0E;//数字 8
        break;
    case 9:seg21&=0xF0;seg20&=0xF1;seg21|=0x0B;seg20|=0x0E;//数字 9
        break;
    default:
        break;
    }
}
else//小于 999 时
{
    seg21&=0xF0;seg20&=0xF1;seg21|=0x00;seg20|=0x00;//不显示
}
/*****dis_rel_speed 百位*****/
if(dis_rel_speed > 99)//大于 99 时
{
    Val=dis_rel_speed/100;//取出百位
    if(dis_rel_speed > 999)//大于 999 时
        Val=Val%10;//取出百位
    switch(Val)
    {
        case 0:seg14&=0xF0;seg15&=0xF1;seg14|=0x0F;seg15|=0x0A;//数字 0
            break;
        case 1:seg14&=0xF0;seg15&=0xF1;seg14|=0x00;seg15|=0x0A;//数字 1
            break;
        case 2:seg14&=0xF0;seg15&=0xF1;seg14|=0x0D;seg15|=0x06;//数字 2
            break;
        case 3:seg14&=0xF0;seg15&=0xF1;seg14|=0x09;seg15|=0x0E;//数字 3
            break;
        case 4:seg14&=0xF0;seg15&=0xF1;seg14|=0x02;seg15|=0x0E;//数字 4
            break;
        case 5:seg14&=0xF0;seg15&=0xF1;seg14|=0x0B;seg15|=0x0C;//数字 5
            break;
        case 6:seg14&=0xF0;seg15&=0xF1;seg14|=0x0F;seg15|=0x0C;//数字 6
            break;
        case 7:seg14&=0xF0;seg15&=0xF1;seg14|=0x01;seg15|=0x0A;//数字 7
            break;
        case 8:seg14&=0xF0;seg15&=0xF1;seg14|=0x0F;seg15|=0x0E;//数字 8
            break;
        case 9:seg14&=0xF0;seg15&=0xF1;seg14|=0x0B;seg15|=0x0E;//数字 9
    }
}

```



```

        break;
    default:
        break;
    }
}
else//小于 99 时
{
    seg14&=0xF0;seg15&=0xF1;seg14|=0x00;seg15|=0x00;//不显示
}
/*****dis_rel_speed 十位*****/
if(dis_rel_speed > 9)//大于 9 时
{
    Val=dis_rel_speed/10;//取出十位
    if(dis_rel_speed > 99)//大于 999 时
        Val=Val%10;//取出十位
    switch(Val)
    {
        case 0:seg16&=0xF0;seg17&=0xF1;seg16|=0x0F;seg17|=0x0A;//数字 0
            break;
        case 1:seg16&=0xF0;seg17&=0xF1;seg16|=0x00;seg17|=0x0A;//数字 1
            break;
        case 2:seg16&=0xF0;seg17&=0xF1;seg16|=0x0D;seg17|=0x06;//数字 2
            break;
        case 3:seg16&=0xF0;seg17&=0xF1;seg16|=0x09;seg17|=0x0E;//数字 3
            break;
        case 4:seg16&=0xF0;seg17&=0xF1;seg16|=0x02;seg17|=0x0E;//数字 4
            break;
        case 5:seg16&=0xF0;seg17&=0xF1;seg16|=0x0B;seg17|=0x0C;//数字 5
            break;
        case 6:seg16&=0xF0;seg17&=0xF1;seg16|=0x0F;seg17|=0x0C;//数字 6
            break;
        case 7:seg16&=0xF0;seg17&=0xF1;seg16|=0x01;seg17|=0x0A;//数字 7
            break;
        case 8:seg16&=0xF0;seg17&=0xF1;seg16|=0x0F;seg17|=0x0E;//数字 8
            break;
        case 9:seg16&=0xF0;seg17&=0xF1;seg16|=0x0B;seg17|=0x0E;//数字 9
            break;
        default:
            break;
    }
}
else//小于 9 时
{
    seg16&=0xF0;seg17&=0xF1;seg16|=0x00;seg17|=0x00;//不显示
}
/*****dis_rel_speed 个位*****/
Val=dis_rel_speed%10;//取出个位
switch(Val)
{
    case 0:seg18&=0xF0;seg19&=0xF1;seg18|=0x0F;seg19|=0x0A;//数字 0

```

```

        break;
    case 1:seg18&=0xF0;seg19&=0xF1;seg18|=0x00;seg19|=0x0A;//数字 1
        break;
    case 2:seg18&=0xF0;seg19&=0xF1;seg18|=0x0D;seg19|=0x06;//数字 2
        break;
    case 3:seg18&=0xF0;seg19&=0xF1;seg18|=0x09;seg19|=0x0E;//数字 3
        break;
    case 4:seg18&=0xF0;seg19&=0xF1;seg18|=0x02;seg19|=0x0E;//数字 4
        break;
    case 5:seg18&=0xF0;seg19&=0xF1;seg18|=0x0B;seg19|=0x0C;//数字 5
        break;
    case 6:seg18&=0xF0;seg19&=0xF1;seg18|=0x0F;seg19|=0x0C;//数字 6
        break;
    case 7:seg18&=0xF0;seg19&=0xF1;seg18|=0x01;seg19|=0x0A;//数字 7
        break;
    case 8:seg18&=0xF0;seg19&=0xF1;seg18|=0x0F;seg19|=0x0E;//数字 8
        break;
    case 9:seg18&=0xF0;seg19&=0xF1;seg18|=0x0B;seg19|=0x0E;//数字 9
        break;
    default:
        break;
    }
}
else/*离心力显示模式*/
{
    seg12&=0xF0;seg13&=0xF1;seg12|=0x00;seg13|=0x00;//不显示
    seg21&=0xF0;seg20&=0xF1;seg21|=0x00;seg20|=0x00;//不显示
    seg14&=0xF0;seg15&=0xF1;seg14|=0x07;seg15|=0x06;//数字 P
    seg16&=0xF0;seg17&=0xF1;seg16|=0x00;seg17|=0x04;//显示"-"
    switch(Speed_Mode)
    {
        case 0:seg18&=0xF0;seg19&=0xF1;seg18|=0x0F;seg19|=0x0A;//数字 0
            break;
        case 1:seg18&=0xF0;seg19&=0xF1;seg18|=0x00;seg19|=0x0A;//数字 1
            break;
        case 2:seg18&=0xF0;seg19&=0xF1;seg18|=0x0D;seg19|=0x06;//数字 2
            break;
        case 3:seg18&=0xF0;seg19&=0xF1;seg18|=0x09;seg19|=0x0E;//数字 3
            break;
        case 4:seg18&=0xF0;seg19&=0xF1;seg18|=0x02;seg19|=0x0E;//数字 4
            break;
        case 5:seg18&=0xF0;seg19&=0xF1;seg18|=0x0B;seg19|=0x0C;//数字 5
            break;
        case 6:seg18&=0xF0;seg19&=0xF1;seg18|=0x0F;seg19|=0x0C;//数字 6
            break;
        case 7:seg18&=0xF0;seg19&=0xF1;seg18|=0x01;seg19|=0x0A;//数字 7
            break;
        case 8:seg18&=0xF0;seg19&=0xF1;seg18|=0x0F;seg19|=0x0E;//数字 8
            break;
        case 9:seg18&=0xF0;seg19&=0xF1;seg18|=0x0B;seg19|=0x0E;//数字 9
    }
}

```

```

        break;
    default:
        break;
    }
}

/***** 盖子打开 *****/
if(Lid_State == 1)
{
    seg10&=0xFE;seg10|=0x01;//开盖
}
/*****

/***** 转速图标 *****/
switch(SpeedIcon_Step)
{
    case 0:seg10&=0xF1;seg11&=0xF1;seg10|=0x0E;seg11|=0x0E;//常亮
        break;
    case 1:seg11&=0xF1;seg11|=0x06;//S8&S9
        break;
    case 2:seg11&=0xF1;seg11|=0x0C;//S9&S10
        break;
    case 3:seg10&=0xF1;seg11&=0xF1;seg10|=0x08;seg11|=0x08;//S10&S5
        break;
    case 4:seg10&=0xF1;seg10|=0x0C;//S5&S6
        break;
    case 5:seg10&=0xF1;seg10|=0x06;//S6&S7
        break;
    case 6:seg10&=0xF1;seg11&=0xF1;seg10|=0x02;seg11|=0x02;//S7&S8
        break;
    default:
        break;
}

/*****

/***** °C&°F 图标 *****/
// seg12&=0xEF;seg12|=0x10;//华氏度
seg13&=0xEF;seg13|=0x10;//摄氏度

/*****

/***** xg&rpm *****/
if(Speed_Unit)
{
    seg19&=0x7F;seg19|=0x80;//xg
}
else
{
    seg17&=0x7F;seg17|=0x80;//rpm
}

```

```

}
/*****/

Write_Addr_Dat_N(18, seg10,1);//SEG20
Write_Addr_Dat_N(20, seg11,1);//SEG19
Write_Addr_Dat_N(22, seg12,1);//SEG18
Write_Addr_Dat_N(24, seg13,1);//SEG17
Write_Addr_Dat_N(26, seg14,1);//SEG16
Write_Addr_Dat_N(28, seg15,1);//SEG15
Write_Addr_Dat_N(30, seg16,1);//SEG14
Write_Addr_Dat_N(32, seg17,1);//SEG13
Write_Addr_Dat_N(34, seg18,1);//SEG12
Write_Addr_Dat_N(36, seg19,1);//SEG11
Write_Addr_Dat_N(38, seg20,1);//SEG10
Write_Addr_Dat_N(40, seg21,1);//SEG9
}

/*
*****
* 函数原型: void Display_RunMode(uint16_t temp,uint8_t P)
* 功    能: 显示安全温度和记忆模式
* 输    入: temp 安全温度  P 记忆模式
* 参    数: uint16_t temp,uint8_t P
*****
*/
void Display_RunMode(uint16_t temp,uint8_t P)
{
    uint8_t seg22,seg23,seg24,seg25,seg26,seg27,seg28,seg29;
    seg22=0;seg23=0;seg24=0;seg25=0;seg26=0;seg27=0;seg28=0;seg29=0;
    uint16_t Val;//用于百十个取出来的数字
    if(Temp_ShowFlag == 0)
    {
        /*****temp 千位*****/
        if(temp > 999)//大于 999 时
        {
            Val=temp/1000;//取出千位
            switch(Val)
            {
                case 0:seg29&=0xF0;seg28&=0xF8;seg29|=0x0F;seg28|=0x05;//数字 0
                    break;
                case 1:seg29&=0xF0;seg28&=0xF8;seg29|=0x00;seg28|=0x05;//数字 1
                    break;
                case 2:seg29&=0xF0;seg28&=0xF8;seg29|=0x0D;seg28|=0x03;//数字 2
                    break;
                case 3:seg29&=0xF0;seg28&=0xF8;seg29|=0x09;seg28|=0x07;//数字 3
                    break;
                case 4:seg29&=0xF0;seg28&=0xF8;seg29|=0x02;seg28|=0x07;//数字 4
                    break;
                case 5:seg29&=0xF0;seg28&=0xF8;seg29|=0x0B;seg28|=0x06;//数字 5
                    break;
            }
        }
    }
}

```

```

        case 6:seg29&=0xF0;seg28&=0xF8;seg29|=0x0F;seg28|=0x06;//数字 6
            break;
        case 7:seg29&=0xF0;seg28&=0xF8;seg29|=0x01;seg28|=0x05;//数字 7
            break;
        case 8:seg29&=0xF0;seg28&=0xF8;seg29|=0x0F;seg28|=0x07;//数字 8
            break;
        case 9:seg29&=0xF0;seg28&=0xF8;seg29|=0x0B;seg28|=0x07;//数字 9
            break;
        default:
            break;
    }
}
else//小于 999 时
{
    seg29&=0xF0;seg28&=0xF8;seg29|=0x00;seg28|=0x00;//不显示
}

/*****temp 百位*****/
if(temp > 99)//大于 99 时
{
    Val=temp/100;//取出百位
    if(temp > 999)//大于 999 时
        Val=Val%10;//取出百位
    switch(Val)
    {
        case 0:seg27&=0xF0;seg26&=0xF8;seg27|=0x0F;seg26|=0x05;//数字 0
            break;
        case 1:seg27&=0xF0;seg26&=0xF8;seg27|=0x00;seg26|=0x05;//数字 1
            break;
        case 2:seg27&=0xF0;seg26&=0xF8;seg27|=0x0D;seg26|=0x03;//数字 2
            break;
        case 3:seg27&=0xF0;seg26&=0xF8;seg27|=0x09;seg26|=0x07;//数字 3
            break;
        case 4:seg27&=0xF0;seg26&=0xF8;seg27|=0x02;seg26|=0x07;//数字 4
            break;
        case 5:seg27&=0xF0;seg26&=0xF8;seg27|=0x0B;seg26|=0x06;//数字 5
            break;
        case 6:seg27&=0xF0;seg26&=0xF8;seg27|=0x0F;seg26|=0x06;//数字 6
            break;
        case 7:seg27&=0xF0;seg26&=0xF8;seg27|=0x01;seg26|=0x05;//数字 7
            break;
        case 8:seg27&=0xF0;seg26&=0xF8;seg27|=0x0F;seg26|=0x07;//数字 8
            break;
        case 9:seg27&=0xF0;seg26&=0xF8;seg27|=0x0B;seg26|=0x07;//数字 9
            break;
        default:
            break;
    }
}
else//小于 99 时

```

```

{
    seg27&=0xF0;seg26&=0xF8;seg27|=0x00;seg26|=0x00;//不显示
}

/*****temp 十位*****/
if(temp > 9)//大于 9 时
{
    Val=temp/10;//取出十位
    if(temp > 99)//大于 999 时
        Val=Val%10;//取出十位
    switch(Val)
    {
        case 0:seg25&=0xF0;seg24&=0xF8;seg25|=0x0F;seg24|=0x05;//数字 0
            break;
        case 1:seg25&=0xF0;seg24&=0xF8;seg25|=0x00;seg24|=0x05;//数字 1
            break;
        case 2:seg25&=0xF0;seg24&=0xF8;seg25|=0x0D;seg24|=0x03;//数字 2
            break;
        case 3:seg25&=0xF0;seg24&=0xF8;seg25|=0x09;seg24|=0x07;//数字 3
            break;
        case 4:seg25&=0xF0;seg24&=0xF8;seg25|=0x02;seg24|=0x07;//数字 4
            break;
        case 5:seg25&=0xF0;seg24&=0xF8;seg25|=0x0B;seg24|=0x06;//数字 5
            break;
        case 6:seg25&=0xF0;seg24&=0xF8;seg25|=0x0F;seg24|=0x06;//数字 6
            break;
        case 7:seg25&=0xF0;seg24&=0xF8;seg25|=0x01;seg24|=0x05;//数字 7
            break;
        case 8:seg25&=0xF0;seg24&=0xF8;seg25|=0x0F;seg24|=0x07;//数字 8
            break;
        case 9:seg25&=0xF0;seg24&=0xF8;seg25|=0x0B;seg24|=0x07;//数字 9
            break;
        default:
            break;
    }
}
else//小于 9 时
{
    seg25&=0xF0;seg24&=0xF8;seg25|=0x0F;seg24|=0x05;//数字 0
}

/*****temp 个位*****/
Val=temp%10;//取出个位
switch(Val)
{
    case 0:seg23&=0xF0;seg22&=0xF8;seg23|=0x0F;seg22|=0x05;//数字 0
        break;
    case 1:seg23&=0xF0;seg22&=0xF8;seg23|=0x00;seg22|=0x05;//数字 1
        break;
    case 2:seg23&=0xF0;seg22&=0xF8;seg23|=0x0D;seg22|=0x03;//数字 2

```

```

        break;
    case 3:seg23&=0xF0;seg22&=0xF8;seg23|=0x09;seg22|=0x07;//数字 3
        break;
    case 4:seg23&=0xF0;seg22&=0xF8;seg23|=0x02;seg22|=0x07;//数字 4
        break;
    case 5:seg23&=0xF0;seg22&=0xF8;seg23|=0x0B;seg22|=0x06;//数字 5
        break;
    case 6:seg23&=0xF0;seg22&=0xF8;seg23|=0x0F;seg22|=0x06;//数字 6
        break;
    case 7:seg23&=0xF0;seg22&=0xF8;seg23|=0x01;seg22|=0x05;//数字 7
        break;
    case 8:seg23&=0xF0;seg22&=0xF8;seg23|=0x0F;seg22|=0x07;//数字 8
        break;
    case 9:seg23&=0xF0;seg22&=0xF8;seg23|=0x0B;seg22|=0x07;//数字 9
        break;
    default:
        break;
    }
}
else
{
    seg25&=0xF0;seg24&=0xF8;seg25|=0x00;seg24|=0x00;//不显示
    seg27&=0xF0;seg26&=0xF8;seg27|=0x00;seg26|=0x00;//不显示
    seg29&=0xF0;seg28&=0xF8;seg29|=0x00;seg28|=0x00;//不显示
    seg23&=0xF0;seg22&=0xF8;seg23|=0x00;seg22|=0x00;//不显示
}

/*****温度小数点*****/
seg24&=0xF7;seg24|=0x08;

/*****显示"SAFE"*****/
if(Safe_Mode)
{
    seg22&=0x7F;seg22|=0x80;//显示"SAFE"
}
else
{
    seg22&=0x7F;seg22|=0x00;//不显示
}

/*****P 模式*****/
switch(PModeIcon_Step)
{
    case
0:seg29&=0x0F;seg27&=0x7F;seg26&=0xF7;seg29|=0xF0;seg27|=0x80;seg26|=0x08;

seg25&=0x7F;seg23&=0x7F;seg22&=0x87;seg25|=0x80;seg23|=0x80;seg22|=0x78;// 外框 全部
亮

        break;
    case 1:seg29&=0x0F;seg29|=0x10;//L1

```

```

        break;
    case 2:seg26&=0xF7;seg26|=0x08;seg29&=0x0F;seg29|=0x10;//L1-L2
        break;
    case
3:seg26&=0xF7;seg26|=0x08;seg29&=0x0F;seg29|=0x10;seg22&=0x87;seg22|=0x08;//L1-L3
        break;
    case
4:seg26&=0xF7;seg26|=0x08;seg29&=0x0F;seg29|=0x10;seg22&=0x87;seg22|=0x18;//L1-L4
        break;
    case
5:seg26&=0xF7;seg26|=0x08;seg29&=0x0F;seg29|=0x10;seg22&=0x87;seg22|=0x38;//L1-L5
        break;
    case
6:seg26&=0xF7;seg26|=0x08;seg29&=0x0F;seg29|=0x10;seg22&=0x87;seg22|=0x78;//L1-L6
        break;
    case
7:seg26&=0xF7;seg22&=0x87;seg23&=0x7F;seg26|=0x08;seg22|=0x78;seg23|=0x80;//L2-L7
        break;
    case
8:seg22&=0x87;seg23&=0x7F;seg25&=0x7F;seg22|=0x78;seg23|=0x80;seg25|=0x80;//L3-L8
        break;
    case
9:seg22&=0x87;seg23&=0x7F;seg25&=0x7F;seg27&=0x7F;seg22|=0x70;seg23|=0x80;seg25|=0
x80;seg27|=0x80;//L4-L9
        break;
    case
10:seg22&=0x87;seg23&=0x7F;seg25&=0x7F;seg27&=0x7F;seg29&=0x0F;seg22|=0x60;seg23|
=0x80;seg25|=0x80;seg27|=0x80;seg29|=0x80;//L5-L10
        break;
    case
11:seg22&=0x87;seg23&=0x7F;seg25&=0x7F;seg27&=0x7F;seg29&=0x0F;seg22|=0x40;seg23|
=0x80;seg25|=0x80;seg27|=0x80;seg29|=0xC0;//L6-L11
        break;
    case
12:seg23&=0x7F;seg25&=0x7F;seg27&=0x7F;seg29&=0x0F;seg23|=0x80;seg25|=0x80;seg27|=
0x80;seg29|=0xE0;//L7-L12
        break;
    case
13:seg29&=0x0F;seg25&=0x7F;seg27&=0x7F;seg25|=0x80;seg27|=0x80;seg29|=0xF0;//L8-L1
        break;
    case
14:seg26&=0xF7;seg29&=0x0F;seg27&=0x7F;seg27|=0x80;seg29|=0xF0;seg26|=0x08;//L9-L2
        break;
    case
15:seg22&=0x87;seg26&=0xF7;seg29&=0x0F;seg29|=0xF0;seg26|=0x08;seg22|=0x08;//L10-L3
        break;
    case
16:seg22&=0x87;seg26&=0xF7;seg29&=0x0F;seg29|=0x70;seg26|=0x08;seg22|=0x18;//L11-L4
        break;
    case

```



```

17:seg22&=0x87;seg26&=0xF7;seg29&=0x0F;seg29|=0x30;seg26|=0x08;seg22|=0x38;//L12-L5
    break;
    case
18:seg22&=0x87;seg26&=0xF7;seg29&=0x0F;seg29|=0x10;seg26|=0x08;seg22|=0x78;//L1-L6
    break;
    default:
        break;
}
if(Show_Circle)
{
    seg28&=0x0F;seg27&=0x8F;seg28|=0x70;seg27|=0x30;//P
    seg25&=0x8F;seg25|=0x20;//"-
    switch(P)
    {
        case 0:seg23&=0x8F;seg24&=0x0F;seg23|=0x50;seg24|=0xF0;//数字 0
            break;
        case 1:seg23&=0x8F;seg24&=0x0F;seg23|=0x50;seg24|=0x00;//数字 1
            break;
        case 2:seg23&=0x8F;seg24&=0x0F;seg23|=0x30;seg24|=0xD0;//数字 2
            break;
        case 3:seg23&=0x8F;seg24&=0x0F;seg23|=0x70;seg24|=0x90;//数字 3
            break;
        case 4:seg23&=0x8F;seg24&=0x0F;seg23|=0x70;seg24|=0x20;//数字 4
            break;
        case 5:seg23&=0x8F;seg24&=0x0F;seg23|=0x60;seg24|=0xB0;//数字 5
            break;
        case 6:seg23&=0x8F;seg24&=0x0F;seg23|=0x60;seg24|=0xF0;//数字 6
            break;
        case 7:seg23&=0x8F;seg24&=0x0F;seg23|=0x50;seg24|=0x10;//数字 7
            break;
        case 8:seg23&=0x8F;seg24&=0x0F;seg23|=0x70;seg24|=0xF0;//数字 8
            break;
        case 9:seg23&=0x8F;seg24&=0x0F;seg23|=0x70;seg24|=0xB0;//数字 9
            break;
        default:
            break;
    }
}
else
{
    seg28&=0x0F;seg27&=0x8F;seg28|=0x00;seg27|=0x00;//P
    seg25&=0x8F;seg25|=0x00;//"-
    seg23&=0x8F;seg24&=0x0F;seg23|=0x00;seg24|=0x00;
    seg29&=0x0F;seg27&=0x7F;seg26&=0xF7;seg29|=0x00;seg27|=0x00;seg26|=0x00;
    seg25&=0x7F;seg23&=0x7F;seg22&=0x87;seg25|=0x00;seg23|=0x00;seg22|=0x00;//
    外框全部灭
}
/*****/
Write_Addr_Dat_N(42, seg22,1);//SEG8
Write_Addr_Dat_N(44, seg23,1);//SEG7

```

```

Write_Addr_Dat_N(46, seg24,1);//SEG6
Write_Addr_Dat_N(48, seg25,1);//SEG5
Write_Addr_Dat_N(50, seg26,1);//SEG4
Write_Addr_Dat_N(52, seg27,1);//SEG3
Write_Addr_Dat_N(54, seg28,1);//SEG2
Write_Addr_Dat_N(56, seg29,1);//SEG1
}

/*
*****
* 函数原型: void Deal_Speed(void)
* 功 能: 速度显示处理
*****
*/
void Deal_Speed(void)
{
    if(sys.Run_Status == 1)//启动的情况下
    {
        if(Speed_ADDMode == 0)//在电机控制中, 速度未处理
        {
            Display_RelSpeed = 0;
            Speed_New = 0;//现在的速度清零
            Speed_Last = 0;//之前的速度清零
            Speed_ADDMode = 1;//进入加速模式下
        }
        if(Speed_ADDMode == 1)//在进入加速模式下
        {
            if(Rel_Speed >= Ctrl_Speed)//实际速度大于等于控制速度
            {
                Speed_ADDMode = 3;//进入稳定模式
                return;
            }
            Speed_New = Rel_Speed;//记录当前速度
            if(Speed_New > Speed_Last)//当前速度大于上一次速度
                Display_RelSpeed = Speed_New;//显示当前速度
            else//当前速度小于上一次速度
            {
                Display_RelSpeed = Speed_Last;//显示上一次速度, 不让速度小于当前速度。呈现攀升速度的现象
                Speed_New = Speed_Last;//将上一次速度赋值给当前速度
            }
            Speed_Last = Speed_New;//将当前速度保存
        }
        else if(Speed_ADDMode == 3)//速度稳定模式下
        {
            Display_RelSpeed = Ctrl_Speed;//显示控制速度
        }
    }
    else
    {

```

```

        Speed_ADDMode = 0;
    }
}

/*
*****
* 函数原型: void Show_Display(void)
* 功    能: 显示屏幕内容
*****
*/
void Show_Display(void)
{
    Display_SetTime = Set_Time;
    Display_CtrlTime = Ctrl_Time;
    Display_Time(Display_SetTime, Display_CtrlTime);
    if(Speed_Mode)//显示离心力
    {
        Display_SetSpeed = Display_Xg;
    }
    else
    {
        Display_SetSpeed = Set_Speed;
    }

    Deal_Speed();
    Display_Speed(Display_SetSpeed, Display_RelSpeed);

    if(sys.SetMode_Option == 3)
    {
        Display_SafeTemp = Safe_Set_Temp;
    }
    else
    {
        Display_SafeTemp = Safe_Rel_Temp;
    }
    Display_RunMode(Display_SafeTemp, PMode_Option);
}
#include "Speed.h"

/*
*****
* 函数原型: void Encoder_Init(void)
* 功    能: 编码器初始化
*****
*/
void Encoder_Init(void)
{
    HAL_TIM_IC_Start_IT(&htim1, TIM_CHANNEL_1);//motor1 输入捕获
}

```

```

/*
*****
* 函数原型: void Check_Speed(float dT)
* 功    能: 检测速度是否停止-0.05s
*****
*/
void Check_Speed(float dT)
{
    Speed_Cnt++;//每 50ms 进入
    if(Speed_Cnt >= 10)//0.5s 发现没出发输入捕获
    {
        Rel_Speed = 0;//将速度清零
        Speed_Cnt = 0;//计数清零
    }
}

/*
*****
* 函数原型: void TIM1CaptureChannel1Callback(void)
* 功    能: Tim1 通道 1 的输入捕获回调函数
*****
*/
uint32_t Capture, Capture1, Capture2;
uint32_t rel;
void TIM1CaptureChannel1Callback(void)
{
    Capture1 = __HAL_TIM_GET_COMPARE(&htim1, TIM_CHANNEL_1);
    if(Capture1 > Capture2)
        Capture = Capture1 - Capture2;
    else
        Capture = Capture1 + (0xFFFF - Capture2);
    if(Capture < 100)
        return;
    rel = 60000000 / (Capture * 9);
    Capture2 = Capture1;
    Rel_Speed = rel;
    Speed_Cnt = 0;
}

/*
*****
* 函数原型: void HAL_TIM_IC_CaptureCallback(TIM_HandleTypeDef *htim)
* 功    能: TIM_IC 回调函数
*****
*/
void HAL_TIM_IC_CaptureCallback(TIM_HandleTypeDef *htim)
{
    if(htim->Instance == TIM1)
    {

```

```

        if(htim->Channel==HAL_TIM_ACTIVE_CHANNEL_1)
        {
            TIM1CaptureChannel1Callback();
        }
    }
}
#include "Ctrl_Motor.h"

/*
*****
* 函数原型: void Motor_Ctrl(float dT)
* 功    能: 电机控制
*****
*/
void Motor_Ctrl(float dT)
{
    if(sys.Run_Status == 1)//启动
    {
        if((HAL_GPIO_ReadPin(UC_IN1_GPIO_Port,UC_IN1_Pin)==
1)&&(HAL_GPIO_ReadPin(UC_IN2_GPIO_Port,UC_IN2_Pin)== 1))//电磁锁 1 和 2 闭合时
        {
            if(Ctrl_Speed && ((DownTime_Over == 0)|| (Ctrl_Time)) && (Safe_Temp_Flag
== 0))//速度大于 0 和定时器没有结束
            {
                COLD_ON;//打开风扇

                HAL_GPIO_WritePin(BREAKEZ_GPIO_Port,BREAKEZ_Pin,GPIO_PIN_SET);//高电 平
                不刹车, 低电平刹车
                PID_Speed(Ctrl_Speed,Rel_Speed,&Speed_Arg,&Speed_Val);//电机 PID 控
                制
                PWM = Speed_Val.Out;//pid 输出
            }
            else
            {
                sys.Motor_Stop = 1;//检测电机
            }
        }
        else
        {
            sys.Run_Status = 0;//不启动
        }
    }
    else
    {
        COLD_OFF;//关闭风扇
        PWM = 0;//pwm 不输出
        Speed_Val.SumError = 0;//防止关闭再打开时速度一下子就冲到之前的速度
        HAL_GPIO_WritePin(BREAKEZ_GPIO_Port,BREAKEZ_Pin,GPIO_PIN_RESET);//
        高电平不刹车, 低电平刹车
    }
}

```

```

}

/*
*****
* 函数原型: void Check_MotorStop(float dT)
* 功    能: 检测电机是否停止, 停止后开盖
*****
*/
void Check_MotorStop(float dT)
{
    static float T;
    if(sys.Motor_Stop)
    {
        if(Rel_Speed == 0)
        {
            T += dT;
            if(T>2)
            {
                Lock1_Status = 1;//电磁锁 1 打开
                Lock2_Status = 1;//电磁锁 2 打开
                sys.Run_Status = 0;//关闭
                sys.Motor_Stop = 0;//电机已经停止
                T = 0;
            }
        }
    }
}
#include "Param.h"

/*****结构体*****/
struct _Save_Param_ Param;//原始数据

/*****全局变量声明*****/
uint8_t Save_Param_En;

/*
*****
* 函数原型: void Param_Reset(void)
* 功    能: 初始化硬件中的参数
*****
*/
void Param_Reset(void)
{
    Param.Flash_Check_Start = FLASH_CHECK_START;
    for(uint8_t i=0;i <= 9;i++)
    {
        Param.P_Param[i][0] = 300;//时间
        Param.P_Param[i][1] = 4000;//转速
        Param.P_Param[i][2] = 500;//安全温度
    }
    Param.Flash_Check_End = FLASH_CHECK_END;
}

```

```

}

/*
*****
* 函数原型: void Param_Save(void)
* 功    能: 保存硬件中的参数
*****
*/
void Param_Save(void)
{
    Flash_Write((uint8_t *)&Param, sizeof(Param));
}

/*
*****
* 函数原型: void Param_Read(void)
* 功    能: 读取硬件中的参数, 判断是否更新
*****
*/
void Param_Read(void)
{
    Flash_Read((uint8_t *)&Param, sizeof(Param));

    //板子从未初始化
    if(Param.Flash_Check_Start != FLASH_CHECK_START || Param.Flash_Check_End !=
FLASH_CHECK_END)
    {
        Param_Reset();
        Set_Time = Param.P_Param[PMode_Option][0];//时间
        Set_Speed = Param.P_Param[PMode_Option][1];//转速
        Safe_Set_Temp = Param.P_Param[PMode_Option][2];//安全温度
        SetOK_Flag = 1;
        Save_Param_En = 1;
    }
    else
    {
        Set_Time = Param.P_Param[PMode_Option][0];//时间
        Set_Speed = Param.P_Param[PMode_Option][1];//转速
        Safe_Set_Temp = Param.P_Param[PMode_Option][2];//安全温度
        SetOK_Flag = 1;
    }

    //保存参数
    if(Save_Param_En)
    {
        Save_Param_En = 0;
        Param_Save();
    }
}

```

```

/*
*****
* 函数原型: void Param_Save_Overtime(float dT)
* 功    能: 保存标志位置 1, 0.5s 后保存
*****
*/
void Param_Save_Overtime(float dT)
{
    static float time;

    if(Save_Param_En)
    {
        time += dT;

        if(time >= 0.5f)
        {
            Param_Save();
            Save_Param_En = 0;
        }
    }
    else
        time = 0;
}
#include "Ctrl_Scheduler.h"

uint16_t  T_cnt_1ms=0,
           T_cnt_6ms=0,
           T_cnt_10ms=0,
           T_cnt_12ms=0,
           T_cnt_20ms=0,
           T_cnt_50ms=0,
           T_cnt_100ms=0,
           T_cnt_200ms=0,
           T_cnt_500ms=0,
           T_cnt_1S=0;

void Loop_Check(void)
{
    T_cnt_1ms++;
    T_cnt_6ms++;
    T_cnt_10ms++;
    T_cnt_12ms++;
    T_cnt_20ms++;
    T_cnt_50ms++;
    T_cnt_100ms++;
    T_cnt_200ms++;
    T_cnt_500ms++;
    T_cnt_1S++;

    Sys_Loop();
}

```



```
}

static void Loop_1ms(void)//1ms 执行一次
{

}

static void Loop_6ms(void)//6ms 执行一次
{

}

static void Loop_10ms(void)//10ms 执行一次
{
    Key_Scan(0.01f);//矩阵按键扫描
    Show_Xg(0.01f);//显示离心力

    Read_Temp(0.01f);//读取温度
    Check_Set(0.01f);//检测设置
}

static void Loop_12ms(void)//12ms 执行一次
{

}

static void Loop_20ms(void)//20ms 执行一次
{
    Ctrl_Lock(0.02f);//电磁铁控制
}

static void Loop_50ms(void)//50ms 执行一次
{
    Motor_Ctrl(0.05f);//控制速度
    Check_Speed(0.05f);//速度静止检测
}

static void Loop_100ms(void)//100ms 执行一次
{
    Buzzer_Status(0.1f);//蜂鸣器的状态检测
    Start_CircleRun(0.1f);//外框跑圈动画
    Start_SpeedRun(0.1f);//转速动画
    Cheak_TimeDown(0.1f);//时间倒计时检测
    Param_Save_Overtime(0.1f);//保存标志位置
}

static void Loop_200ms(void)//200ms 执行一次
{
    Check_MotorStop(0.2f);//检测电机是否停止，停止后开盖
    Check_Safe_Temp(0.2f);//检测安全温度
```

```
}

static void Loop_500ms(void)//500ms 执行一次
{
    Check_ShowFlag(0.5f);//闪烁检测
    Check_Press(0.5f);//检测按键按下状态
    Check_Safe_Mode(0.5f);//安全温度图标闪烁检测
}

static void Loop_1S(void)//1S 执行一次
{

}

void Sys_Loop(void)
{
    if(T_cnt_1ms >= 1) {
        Loop_1ms();
        T_cnt_1ms = 0;
    }
    if(T_cnt_6ms >= 6) {
        Loop_6ms();
        T_cnt_6ms = 0;
    }
    if(T_cnt_10ms >= 10) {
        Loop_10ms();
        T_cnt_10ms = 0;
    }
    if(T_cnt_12ms >= 12) {
        Loop_12ms();
        T_cnt_12ms = 0;
    }
    if(T_cnt_20ms >= 20) {
        Loop_20ms();
        T_cnt_20ms = 0;
    }
    if(T_cnt_50ms >= 50) {
        Loop_50ms();
        T_cnt_50ms = 0;
    }
    if(T_cnt_100ms >= 100) {
        Loop_100ms();
        T_cnt_100ms = 0;
    }
    if(T_cnt_200ms >= 200) {
        Loop_200ms();
        T_cnt_200ms = 0;
    }
    if(T_cnt_500ms >= 500) {
        Loop_500ms();
    }
}
```

```
        T_cnt_500ms = 0;
    }
    if(T_cnt_1S >= 1000) {
        Loop_1S();
        T_cnt_1S = 0;
    }
}
```