

# LC3100\_CGF\_8 软件源程序

```

#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 <= 8;i++)
    {
        Param.P_Param[i][0] = 300;//时间
        Param.P_Param[i][1] = 2300;//转速
        Param.P_Param[i][2] = 500;//安全温度
        Param.P_Param[i][3] = 100;//离心率
    }
    #if(Type == 0)
    /*****CGF*****/
    Param.P_Param[1][0] = 840;//时间
    Param.P_Param[1][1] = 2450;//转速
    Param.P_Param[2][1] = 2150;//转速
    Param.P_Param[3][1] = 2450;//转速
    Param.P_Param[4][1] = 2690;//转速
    /*****PrP*****/
    Param.P_Param[5][0] = 300;//时间
    Param.P_Param[5][1] = 3930;//转速
    /*****APrF*****/
    Param.P_Param[6][0] = 720;//时间
    Param.P_Param[6][1] = 1300;//转速
    /*****IPrF*****/
    Param.P_Param[7][0] = 300;//时间
    Param.P_Param[7][1] = 680;//转速
    #elif(Type == 1)
    /*****CGF*****/
    Param.P_Param[1][0] = 840;//时间
    Param.P_Param[1][1] = 2350;//转速
    Param.P_Param[2][1] = 2070;//转速
    Param.P_Param[3][1] = 2350;//转速
    Param.P_Param[4][1] = 2590;//转速
    /*****PrP*****/
    Param.P_Param[5][0] = 300;//时间

```

```

Param.P_Param[5][1] = 3780;//转速
/*****APrF*****/
Param.P_Param[6][0] = 720;//时间
Param.P_Param[6][1] = 1250;//转速
/*****IPrF*****/
Param.P_Param[7][0] = 300;//时间
Param.P_Param[7][1] = 640;//转速
#endif

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();
        Time.Set = Param.P_Param[PMODE.Option][0];//时间
        SafeTemp.Set = Param.P_Param[PMODE.Option][2];//安全温度
        if(Speed.Unit)
            Speed.Set = Param.P_Param[PMODE.Option][3];//离心率
        else
            Speed.Set = Param.P_Param[PMODE.Option][1];//转速
        SetOK_Flag = 1;
        Save_Param_En = 1;
    }
    else
    {
        Time.Set = Param.P_Param[PMODE.Option][0];//时间

```

---

```

    SafeTemp.Set = Param.P_Param[PMode.Option][2];//安全温度
    if(Speed.Unit)
        Speed.Set = Param.P_Param[PMode.Option][3];//离心率
    else
        Speed.Set = Param.P_Param[PMode.Option][1];//转速
    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 "SetVal.h"

/*****全局变量声明*****/
uint8_t SetOK_Flag;//检测是否按下按键

/*
*****
* 函数原型: void Check_Set(float dT)
* 功    能: 检测设置
*****
*/

```

```

void Check_Set(float dT)
{
    if(Key_Status)
    {
        SetOK_Flag = 1;//检测到波动旋钮，等待退出设置模式
    }
    if(SetOK_Flag)
    {
        if(!sys.SetMode_Option)//在设定好后
        {
            if(Speed.Ctrl != Speed.Set)//判断控制速度和设定速度是不是不一样
            {
                Speed.Ctrl = Speed.Set;//把设定速度赋值给控制速度
                if(Speed.Unit == 0)
                    Param.P_Param[PMode.Option][1] = Speed.Set;//转速
                else
                    Param.P_Param[PMode.Option][3] = Speed.Set;//离心率
            }
            if(Time.Rel != Time.Set)//实际时间不等于设定时间
            {
                Time.Rel = Time.Set;//把设定时间赋值给控制时间
                Param.P_Param[PMode.Option][0] = Time.Set;//时间
            }
            Param.P_Param[PMode.Option][2] = SafeTemp.Set;//安全温度
            Save_Param_En = 1;//保存
            SetOK_Flag = 0;
        }
    }
}

#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.Stop_Cnt++;//每 50ms 进入
if(Speed.Stop_Cnt >= 10)//0.5s 发现没出发输入捕获
{
    Speed.Rel = 0;//将速度清零
    Speed.Stop_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);//读取编码器的
    值
    if(Capture1 > Capture2)//假如读到的值大于上一次的值
        Capture = Capture1 - Capture2;//得出当前的值
    else
        Capture = Capture1 + (0xFFFF - Capture2);//如果小于上次的值, 0xFFFF-上一次的值
    if(Capture < 100)//过滤小于 100 的
        return;
    rel = 60000000 / (Capture * 12);//用 1S/编码器的值*一圈的脉冲
    Capture2 = Capture1;//将读到的值赋值给上一次的值
    Speed.Rel = rel;
    Speed.Stop_Cnt = 0;//将检测速度的时间清零
}

/*
*****
* 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 "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
uint8_t Lock_ShowFlag,TimeIcn_ShowFlag,CGF_ShowFlag,SafeTemp_ShowFlag;//开盖图标和
时间图标闪烁和 CGF 闪烁和安全温度图标闪烁

/*
*****
* 函数原型: static void Time_Twinkle(float dT)
* 功 能: 时间图标闪烁
* 调 用: 内部调用
*****
*/
static void Time_Twinkle(float dT)
{
    static float T;
    if(sys.Run_Status)
    {
        T += dT;
        if(T >= 0.5f)
        {
            TimeIcn_ShowFlag = ~TimeIcn_ShowFlag;//时间图标闪烁;
            T = 0;
        }
    }
    else
    {
        TimeIcn_ShowFlag = 0;//显示时间图标
    }
}

/*
*****
* 函数原型: void Check_Lock(float dT)
* 功 能: 开盖图标闪烁检测
*****
*/
void Check_Lock(float dT)
{
    static float T;
    if(sys.Lock_On == 0)
    {
        Lock_ShowFlag = 0;
        return;
    }
    T += dT;
    if(T >= 0.5f)

```

---

```

    {
        Lock_ShowFlag = ~Lock_ShowFlag;//开盖图标闪烁;
        T = 0;
    }
    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 和电磁锁 2 都
闭合时
        sys.Lock_On = 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)
    {
        Speed.Icon_Step = 0;
        T = 0;
        return;
    }
    T += dT;
    if(T >= 0.5)
    {
        Speed.Icon_Step++;
        if(Speed.Icon_Step > 3)
            Speed.Icon_Step = 1;
        T = 0;
    }
}

/*
*****
* 函数原型: static void Check_Safe_Mode(float dT)
* 功    能: 安全温度图标闪烁检测
* 调    用: 内部调用
*****
*/
static void Check_Safe_Mode(float dT)
{
    static float T;
    if(SafeTemp.Twinkleg)
    {
        T += dT;
        if(T >= 0.5f)

```



---

```

        {
            SafeTemp_ShowFlag = ~SafeTemp_ShowFlag;//安全温度图标闪烁
            T = 0;
        }
    }
else
{
    if(sys.SetMode_Option == 3)
        SafeTemp_ShowFlag = 1;
    else
        SafeTemp_ShowFlag = 0;
}
}

/*
*****
* 函数原型: static void Check_ShowFlag(float dT)
* 功    能: 闪烁检测
* 输    入: dT:执行周期
* 参    数: float dT
* 调    用: 内部调用
*****
*/
static void Check_ShowFlag(float dT)
{
    static float T;
    if(sys.SetMode_Option == 0)//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
    {
        T += dT;
        if(T >= 0.5f)
        {
            Twinkle_Time -= 0.5;//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
    {
        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
    }
    T = 0;
}
}
else
{
    Speed_ShowFlag = 0;//常亮
    Temp_ShowFlag = 0;//常亮
    Time_ShowFlag = 0;//常亮
    T = 0;
}
}

/*
*****
* 函数原型: void Twinkle(float dT)
* 功    能: 闪烁函数
*****
*/
void Twinkle(float dT)
{
    Time_Twinkle(dT);//时间图标闪烁
    Check_Lock(dT);//开盖图标闪烁检测
    Check_Safe_Mode(dT);//安全温度图标闪烁检测
    Start_SpeedRun(dT);//转速动画
    Check_ShowFlag(dT);//闪烁检测
}

/*
*****
* 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;

```

## LC3100\_CGF\_8 软件 V1.0

```
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
```

```
{
```

```
    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)//设置时间闪烁
```

```
{
```

```
    /******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)
```

```
    {
```

## LC3100\_CGF\_8 软件 V1.0

```

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(dis_set_time > 3599)//如果设定时间大于 59.59 分钟时
{
    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;

```

## LC3100\_CGF\_8 软件 V1.0

```

        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;
    }

```

## LC3100\_CGF\_8 软件 V1.0

```

        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(dis_set_time > 3599)//如果设定时间大于 59.59 分钟时
    {
        seg9&=0xF7;seg9|=0x08;//显示 min
    }
    else
    {
        seg7&=0x7F;seg7|=0x80;//显示 sec
    }

    /*****关闭盖子图标*****/
    if(Lock_ShowFlag == 0)//盖子底部(因为关闭不会触发报警, 所以没有加闪烁)
    {
        seg9&=0xFD;seg9|=0x02;//显示盖子底部
    }
    else
    {
        seg9&=0xFD;seg9|=0x00;//不显示盖子底部
    }
    if(sys.Lid_State == 0)
    {
        seg9&=0xFE;seg9|=0x01;//关闭盖子图标
    }
    else
    {
        seg9&=0xFE;seg9|=0x00;//关闭盖子图标不显示
    }

    /*****

    /*****时间图标*****/
    if(TimeIcn_ShowFlag == 0)
    {
        seg9&=0xFB;seg9|=0x04;//显示
    }
    else

```



---

```

{
    seg9&=0xFB;seg9|=0x00;//不显示
}

/*****/
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(sys.Run_Mode == 0)//正常模式下
    {
        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
            {

```

## LC3100\_CGF\_8 软件 V1.0

---

```

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;
    }
}

```

## LC3100\_CGF\_8 软件 V1.0

```

        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*****/
/*****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)
    {

```

## LC3100\_CGF\_8 软件 V1.0

```

        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;
    }
}

```

## LC3100\_CGF\_8 软件 V1.0

```

        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
}

```

## LC3100\_CGF\_8 软件 V1.0

```

/*****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 99 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;

```

## LC3100\_CGF\_8 软件 V1.0

```

        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;
    }
    /*****xg&rpm*****/
    if(Speed.Unit)//速度单位
    {
        seg19&=0x7F;seg19|=0x80;//xg
    }
    else
    {
        seg17&=0x7F;seg17|=0x80;//rpm
    }
}
else if(sys.Run_Mode == 1)//在 CGF 模式下，设定显示 “----” 和实际速度显示 “CGF”
{
    seg20&=0x7F;seg20|=0x00;//不显示
    seg20&=0x8F;seg21&=0x0F;seg20|=0x00;seg21|=0x00;//显示 “-”
    seg15&=0x8F;seg14&=0x0F;seg15|=0x00;seg14|=0x00;//显示 “-”
    seg17&=0x8F;seg16&=0x0F;seg17|=0x00;seg16|=0x00;//显示 “-”
    seg19&=0x8F;seg18&=0x0F;seg19|=0x00;seg18|=0x00;//显示 “-”

    seg12&=0xF0;seg13&=0xF1;seg12|=0x00;seg13|=0x00;//显示 “-”
    seg21&=0xF0;seg20&=0xF1;seg21|=0x0F;seg20|=0x00;//显示 “C”
    seg14&=0xF0;seg15&=0xF1;seg14|=0x0F;seg15|=0x0C;//显示 “G”
    seg16&=0xF0;seg17&=0xF1;seg16|=0x07;seg17|=0x04;//显示 “F”
    seg18&=0xF0;seg19&=0xF1;seg18|=0x00;seg19|=0x00;//显示 “ ”
    seg19&=0x7F;seg19|=0x00;//不显示 “xg”
    seg17&=0x7F;seg17|=0x00;//不显示 “rpm”
}
else if(sys.Run_Mode == 2)//在 PRP 模式下，设定显示 “----” 和实际速度显示 “PrP”
{
    seg20&=0x7F;seg20|=0x00;//不显示
    seg20&=0x8F;seg21&=0x0F;seg20|=0x00;seg21|=0x00;//显示 “-”
    seg15&=0x8F;seg14&=0x0F;seg15|=0x00;seg14|=0x00;//显示 “-”
    seg17&=0x8F;seg16&=0x0F;seg17|=0x00;seg16|=0x00;//显示 “-”
    seg19&=0x8F;seg18&=0x0F;seg19|=0x00;seg18|=0x00;//显示 “-”

    seg12&=0xF0;seg13&=0xF1;seg12|=0x00;seg13|=0x00;//显示 “-”
    seg21&=0xF0;seg20&=0xF1;seg21|=0x07;seg20|=0x06;//显示 “P”
    seg14&=0xF0;seg15&=0xF1;seg14|=0x06;seg15|=0x04;//显示 “r”
}

```



## LC3100\_CGF\_8 软件 V1.0

```

seg16&=0xF0;seg17&=0xF1;seg16|=0x07;seg17|=0x06;//显示 “P”
seg18&=0xF0;seg19&=0xF1;seg18|=0x00;seg19|=0x00;//显示 “ ”
seg19&=0x7F;seg19|=0x00;//不显示 “xg”
seg17&=0x7F;seg17|=0x00;//不显示 “rpm”
}
else if(sys.Run_Mode == 3)//在 APRF 模式下，设定显示 “----” 和实际速度显示 “APRF”
{
    seg20&=0x7F;seg20|=0x00;//不显示
    seg20&=0x8F;seg21&=0x0F;seg20|=0x00;seg21|=0x00;//显示 “-”
    seg15&=0x8F;seg14&=0x0F;seg15|=0x00;seg14|=0x00;//显示 “-”
    seg17&=0x8F;seg16&=0x0F;seg17|=0x00;seg16|=0x00;//显示 “-”
    seg19&=0x8F;seg18&=0x0F;seg19|=0x00;seg18|=0x00;//显示 “-”

    seg12&=0xF0;seg13&=0xF1;seg12|=0x00;seg13|=0x00;//显示 “-”
    seg21&=0xF0;seg20&=0xF1;seg21|=0x07;seg20|=0x0E;//显示 “A”
    seg14&=0xF0;seg15&=0xF1;seg14|=0x07;seg15|=0x06;//显示 “P”
    seg16&=0xF0;seg17&=0xF1;seg16|=0x06;seg17|=0x04;//显示 “r”
    seg18&=0xF0;seg19&=0xF1;seg18|=0x07;seg19|=0x04;//显示 “F”
    seg19&=0x7F;seg19|=0x00;//不显示 “xg”
    seg17&=0x7F;seg17|=0x00;//不显示 “rpm”
}
else if(sys.Run_Mode == 4)//在 IPRF 模式下，设定显示 “----” 和实际速度显示 “IPRF”
{
    seg20&=0x7F;seg20|=0x00;//不显示
    seg20&=0x8F;seg21&=0x0F;seg20|=0x00;seg21|=0x00;//显示 “-”
    seg15&=0x8F;seg14&=0x0F;seg15|=0x00;seg14|=0x00;//显示 “-”
    seg17&=0x8F;seg16&=0x0F;seg17|=0x00;seg16|=0x00;//显示 “-”
    seg19&=0x8F;seg18&=0x0F;seg19|=0x00;seg18|=0x00;//显示 “-”

    seg12&=0xF0;seg13&=0xF1;seg12|=0x00;seg13|=0x00;//显示 “-”
    seg21&=0xF0;seg20&=0xF1;seg21|=0x06;seg20|=0x00;//显示 “I”
    seg14&=0xF0;seg15&=0xF1;seg14|=0x07;seg15|=0x06;//显示 “P”
    seg16&=0xF0;seg17&=0xF1;seg16|=0x06;seg17|=0x04;//显示 “r”
    seg18&=0xF0;seg19&=0xF1;seg18|=0x07;seg19|=0x04;//显示 “F”
    seg19&=0x7F;seg19|=0x00;//不显示 “xg”
    seg17&=0x7F;seg17|=0x00;//不显示 “rpm”
}

/***** 盖子打开 *****/
if(sys.Lid_State && (Lock_ShowFlag == 0))
{
    seg10&=0xFE;seg10|=0x01;//开盖
}
else
{
    seg10&=0xFE;seg10|=0x00;//开盖不显示
}

/*****/

```

```

/*****转速图标*****/
switch(Speed.Icon_Step)
{
    case 0:seg10&=0xF1;seg11&=0xF1;seg10|=0x0E;seg11|=0x0E;//常亮
        break;
    case 1:seg11&=0xF1;seg11|=0x06;seg10&=0xF1;seg10|=0x0C;//S5&S6&S8&S9
        break;
    case 2:seg10&=0xF1;seg11&=0xF1;seg10|=0x0A;seg11|=0x0A;//S10&S5&S7&S8
        break;
    case 3:seg11&=0xF1;seg11|=0x0C;seg10&=0xF1;seg10|=0x06;//S6&S7&S9&S10
        break;
    default:
        break;
}

/*****

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

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)
* 功    能: 显示安全温度和记忆模式
* 输    入: temp: 安全温度
* 参    数: uint16_t temp
*****
*/
void Display_RunMode(uint16_t temp)
{
    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;

```

## LC3100\_CGF\_8 软件 V1.0

```

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

seg28&=0x0F;seg27&=0x8F;seg28|=0x00;seg27|=0x00;//C 不显示
seg26&=0x0F;seg25&=0x8F;seg26|=0x00;seg25|=0x00;//G 不显示
seg24&=0x0F;seg23&=0x8F;seg24|=0x00;seg23|=0x00;//F 不显示
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(float dT)
* 功    能： 速度显示处理
*****
*/
float Speed_Xg;
void Deal_Speed(float dT)
{
    if(sys.Run_Status == 1)//启动的情况下
    {
        if(Speed.Unit)
        {
            Speed_Xg = 11.18f * ((float)Speed.Rel/1000.0f) * ((float)Speed.Rel/1000.0f) *
9.6f;//计算当前显示的离心力速度
        }
        else
        {
            Speed_Xg = Speed.Rel;//显示速度
        }

        if(Speed.ADDMode == 0)//在电机控制中，速度未处理
        {

```

---

```

Speed.Display_Rel = 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
}
else if(Speed.ADDMode==1) // 在进入加速模式下
{
    Speed.New = Speed_Xg; // 记录当前速度
    if(Speed.New > Speed.Display_Rel) // 当前速度大于显示速度
    {
        if(Speed.Display_Rel < Speed.New)
            Speed.Display_Rel += 1; // 显示当前速度
    }
    else // 当前速度小于上一次速度
    {
        Speed.Display_Rel = Speed.Display_Rel; // 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
    }
    if(sys.Motor_Stop == 0)
    {
        if(Speed.Display_Rel >= Speed.Ctrl) // 实际速度大于等于控制速度
        {
            Speed.ADDMode = 3; // 进入稳定模式
            return;
        }
    }
}
else if(Speed.ADDMode==2) // 在进入加速模式下
{
    Speed.New = Speed_Xg; // 记录当前速度

    if(Speed.New < Speed.Display_Rel) // 当前速度小于上一次速度
    {
        if(Speed.Display_Rel > Speed.New)
            Speed.Display_Rel -= 1; // 显示当前速度
    }
    else // 当前速度大于上一次速度
    {
        Speed.Display_Rel = Speed.Display_Rel; // 显示上一次速度，不让速度大于
        当前速度。呈现下降速度的现象
    }
    if(sys.Motor_Stop == 0)
    {
        if(Speed.Display_Rel <= Speed.Ctrl) // 实际速度小于等于控制速度
        {
            Speed.ADDMode = 3; // 进入稳定模式
            return;
        }
    }
}
}

```

---

```

        else if(Speed.ADDMode == 3)//速度稳定模式下
        {
            Speed.Display_Rel = Speed.Ctrl;//显示控制速度
        }
    }
    else
    {
        Speed.Display_Rel = 0;//没开始转速显示为 0
    }
}

/*
*****
* 函数原型: void Show_Display(void)
* 功    能: 显示屏幕内容
*****
*/
void Show_Display(void)
{
    Time.Display_Rel = Time.Rel;
    Time.Display_Set = Time.Set;
    Speed.Display_Set = Speed.Set;
    if(sys.SetMode_Option == 3)
    {
        SafeTemp.Display = SafeTemp.Set;
    }
    else
    {
        SafeTemp.Display = SafeTemp.Rel;
    }

    Display_Time(Time.Display_Set,Time.Display_Rel);
    Display_Speed(Speed.Display_Set,Speed.Display_Rel);
    Display_RunMode(SafeTemp.Display);
}
#include "Drv_Flash.h"

/*****用法*****/
//Flash_Write((uint8_t *)&Param,sizeof(Param));
//Flash_Read((uint8_t *)&Param,sizeof(Param));
/*
*****
* 函数原型: uint8_t Flash_Write(uint8_t *addr, uint16_t len)
* 功    能: 写入 Flash
* 输    入: addr 需要写入结构体的地址, len 结构体长度
* 输    出: 写入是否成功
* 参    数: uint8_t *addr, uint16_t len
*****
*/

```



```

uint8_t Flash_Write(uint8_t *addr, uint16_t len)
{
    uint16_t FlashStatus;//定义写入 Flash 状态
    FLASH_EraseInitTypeDef My_Flash;// 声明 FLASH_EraseInitTypeDef 结构体为 My_Flash

    HAL_FLASH_Unlock();//解锁 Flash

    My_Flash.TypeErase = FLASH_TYPEERASE_PAGES;//标明 Flash 执行页面只做擦除操作

    My_Flash.PageAddress = PARAMFLASH_BASE_ADDRESS;//声明要擦除的地址
    My_Flash.NbPages = 1;//说明要擦除的页数，此参数必须是 Min_Data = 1 和 Max_Data =(最大页数-初始页的值)之间的值

    uint32_t PageError = 0;//设置 PageError,如果出现错误这个变量会被设置为出错的 FLASH 地址

    FlashStatus = HAL_FLASHEx_Erase(&My_Flash, &PageError);//调用擦除函数（擦除 Flash）
    if(FlashStatus != HAL_OK)
        return 0;

    for(uint16_t i=0; i<len; i=i+2)
    {
        uint16_t temp;//临时存储数值
        if(i+1 <= len-1)
            temp = (uint16_t)(addr[i+1]<<8) + addr[i];
        else
            temp = 0xff00 + addr[i];
        //对 Flash 进行烧写，FLASH_TYPEPROGRAM_HALFWORD 声明操作的 Flash 地址的 16 位的，此外还有 32 位跟 64 位的操作，自行翻查 HAL 库的定义即可
        FlashStatus = HAL_FLASH_Program(FLASH_TYPEPROGRAM_HALFWORD, PARAMFLASH_BASE_ADDRESS+i, temp);
        if (FlashStatus != HAL_OK)
            return 0;
    }
    HAL_FLASH_Lock();//锁住 Flash
    return 1;
}

/*
*****
* 函数原型：uint8_t Flash_Read(uint8_t *addr, uint16_t len)
* 功    能：读取 Flash
* 输    入：addr 需要写入结构体的地址，len 结构体长度
* 输    出：读取是否成功
* 参    数：uint8_t *addr, uint16_t len
*****
*/
uint8_t Flash_Read(uint8_t *addr, uint16_t len)

```

---

```
{
    for(uint16_t i=0; i<len; i=i+2)
    {
        uint16_t temp;
        if(i+1 <= len-1)
        {
            temp = (*(__IO uint16_t*)(PARAMFLASH_BASE_ADDRESS+i));/*(__IO
uint16_t *)是读取该地址的参数值,其值为 16 位数据,一次读取两个字节
            addr[i] = BYTE0(temp);
            addr[i+1] = BYTE1(temp);
        }
        else
        {
            temp = (*(__IO uint16_t*)(PARAMFLASH_BASE_ADDRESS+i));
            addr[i] = BYTE0(temp);
        }
    }
    return 1;
}
```