WBS3100 软件源程序

```
#include "Drv_Beep.h"

/**********全局变量**********/
float Beep_Time;//蜂鸣器响的时间
float Beep_Flash;//蜂鸣器响的次数

/*
*********************************************************************
 * 函数原型：  void Buzzer_Status(float dT)
 * 功      能：  蜂鸣器的状态检测
 * 输      入:  dT:执行周期
 * 参      数：  uint16_t dT
*********************************************************************
*/
void Buzzer_Status(float dT)
{
    static float BT;
    if(Beep_Time <= 0 && Beep_Flash <= 0)//蜂鸣器的时间小于等于 0 时
    {
        Beep_OFF;//关闭蜂鸣器
        return;
    }
    if(Beep_Time)
    {
        Beep_ON;//打开蜂鸣器
        Beep_Time -= dT;//蜂鸣器响的时间--
    }
    if(Beep_Flash)
    {
        BT =   BT + dT;//周期++
        if(BT < 0.2)//如果小于 0.2s 时
        {
            Beep_ON;//蜂鸣器响
        }
        else if(BT >= 0.2 && BT < 0.3)//在 0.2 和 0.3s 之间时
        {
            Beep_OFF;//关闭蜂鸣器
        }
        else if(BT >= 0.3)//大于等于 0.3 时
        {
            Beep_Flash--;//次数--
            BT = 0;//周期清零
        }
    }
}
#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;
}
#include "Drv_Heat.h"

/*
*****************************************************************
 * 函数原型：void Temp_Init(void)
 * 功　　能：温度初始化
*****************************************************************
*/
void Temp_Init(void)
{
    HAL_TIM_PWM_Start(&htim1, TIM_CHANNEL_1);//开启 tim1 通道 1 的 PWM
}
#include "Drv_HT1623.h"

/*
*****************************************************************
 * 函数原型：static void delay(uint16_t time)
 * 功　　能：us 延时
 * 输　　入: time ： 时间
 * 参　　数：uint16_t time
 * 调　　用：内部调用
*****************************************************************
```

```
*/
static void delay(uint16_t time)
{
    unsigned char a;
    for(a = 100; a > 0; a--);
}

/*
*********************************************************************
 * 函数原型：static void write_mode(unsigned char MODE)
 * 功    能：写入模式,数据 or 命令
 * 输    入: MODE ：数据 or 命令
 * 参    数：unsigned char MODE
 * 调    用：内部调用
*********************************************************************
*/
static void write_mode(unsigned char MODE)
{
    delay(10);
    Clr_1625_Wr;//RW = 0;
    delay(10);
    Set_1625_Dat;//DA = 1;
    Set_1625_Wr;//RW = 1;
    delay(10);
    Clr_1625_Wr;//RW = 0;
    delay(10);
    Clr_1625_Dat;
    delay(10);//DA = 0;
    Set_1625_Wr;//RW = 1;
    delay(10);
    Clr_1625_Wr;//RW = 0;
    delay(10);
    if (0 == MODE)
    {
        Clr_1625_Dat;//DA = 0;
    }
    else
    {
        Set_1625_Dat;//DA = 1;
    }
    delay(10);
    Set_1625_Wr;//RW = 1;
    delay(10);
}

/*
*********************************************************************
 * 函数原型：static void write_command(unsigned char Cbyte)
 * 功    能：LCD 命令写入函数
 * 输    入: Cbyte：控制命令字
```

```
 * 参      数：unsigned char Cbyte
 * 调      用：内部调用
**********************************************************************
*/
static void write_command(unsigned char Cbyte)
{
      unsigned char i = 0;
      for (i = 0; i < 8; i++)
      {
            Clr_1625_Wr;
            //Delay_us(10);
            if ((Cbyte >> (7 - i)) & 0x01)
            {
                  Set_1625_Dat;
            }
            else
            {
                  Clr_1625_Dat;
            }
            delay(10);
            Set_1625_Wr;
            delay(10);
      }
      Clr_1625_Wr;
      delay(10);
      Clr_1625_Dat;
      Set_1625_Wr;
      delay(10);
}

/*
**********************************************************************
 * 函数原型：static void write_address(unsigned char Abyte)
 * 功      能：LCD 地址写入函数
 * 输      入: Abyte：地址
 * 参      数：unsigned char Abyte
 * 调      用：内部调用
**********************************************************************
*/
static void write_address(unsigned char Abyte)
{
      unsigned char i = 0;
      Abyte = Abyte << 1;
      for (i = 0; i < 6; i++)
      {
            Clr_1625_Wr;
            if ((Abyte >> (6 - i)) & 0x01)
            {
                  Set_1625_Dat;
            }
```

```
        else
        {
            Clr_1625_Dat;
        }
        delay(10);
        Set_1625_Wr;
        delay(10);
    }
}

/*
*********************************************************************
 * 函数原型：static void write_data_8bit(unsigned char Dbyte)
 * 功    能：LCD 8bit 数据写入函数
 * 输    入: Dbyte：数据
 * 参    数：unsigned char Dbyte
 * 调    用：内部调用
*********************************************************************
*/
static void write_data_8bit(unsigned char Dbyte)
{
    int i = 0;
    for (i = 0; i < 8; i++)
    {
        Clr_1625_Wr;
        if ((Dbyte >> (7 - i)) & 0x01)
        {
            Set_1625_Dat;
        }
        else
        {
            Clr_1625_Dat;
        }
        delay(10);
        Set_1625_Wr;
        delay(10);
    }
}

/*
*********************************************************************
 * 函数原型：void write_data_4bit(unsigned char Dbyte)
 * 功    能：LCD 4bit 数据写入函数
 * 输    入: Dbyte：数据
 * 参    数：unsigned char Dbyte
 * 调    用：内部调用
*********************************************************************
*/
void write_data_4bit(unsigned char Dbyte)
{
```

```
    int i = 0;
    for (i = 0; i < 4; i++)
    {
        Clr_1625_Wr;
        if ((Dbyte >> (3 - i)) & 0x01)
        {
            Set_1625_Dat;
        }
        else
        {
            Clr_1625_Dat;
        }
        delay(10);
        Set_1625_Wr;
        delay(10);
    }
}

/*
*********************************************************************
 * 函数原型：void Lcd_Init(void)
 * 功　　能：LCD 初始化，对 lcd 自身做初始化设置
*********************************************************************
*/
void Lcd_Init(void)
{
    Set_1625_Cs;
    Set_1625_Wr;
    Set_1625_Dat;
    delay(500);
    Clr_1625_Cs;//CS = 0;
    delay(10);
    write_mode(0);//命令模式
    write_command(0x01);//Enable System
    write_command(0x03);//Enable Bias
    write_command(0x04);//Disable Timer
    write_command(0x05);//Disable WDT
    write_command(0x08);//Tone OFF
    write_command(0x18);//on-chip RC 震荡
    write_command(0x29);//1/4Duty 1/3Bias
    write_command(0x80);//Disable IRQ
    write_command(0x40);//Tone Frequency 4kHZ
    write_command(0xE3);//Normal Mode
    Set_1625_Cs;//CS = 1;

    HAL_TIM_PWM_Start(&htim1, TIM_CHANNEL_2);
    __HAL_TIM_SET_COMPARE(&htim1, TIM_CHANNEL_2, 17);//不输出 pwm

    Lcd_All();
    HAL_Delay(1000);
```

```
    Lcd_Clr();
}

/*
*********************************************************************
 * 函数原型：void Lcd_Clr(void)
 * 功      能：LCD 清屏函数
*********************************************************************
*/
void Lcd_Clr(void)
{
    Write_Addr_Dat_N(0x0, 0x00, 60);
}

/*
*********************************************************************
 * 函数原型：void Lcd_All(void)
 * 功      能：LCD 全显示函数
*********************************************************************
*/
void Lcd_All(void)
{
    Write_Addr_Dat_N(0x0, 0xff, 60);
}

/*
*********************************************************************
 * 函数原型：void Write_Addr_Dat_N(unsigned char _addr, unsigned char _dat, unsigned char
n)
 * 功      能：屏幕显示
 * 输      入: _addr：地址   char _dat：数据 n：个数
 * 参      数：unsigned char _addr, unsigned char _dat, unsigned char n
*********************************************************************
*/
void Write_Addr_Dat_N(unsigned char _addr, unsigned char _dat, unsigned char n)
{
    unsigned char i = 0;
    Clr_1625_Cs;//CS = 0;
    delay(10);
    write_mode(1);
    write_address(_addr);
    for (i = 0; i < n; i++)
    {
        write_data_8bit(_dat);
    }
    Set_1625_Cs;//CS = 1;
}
#include "Drv_Key.h"

/**********全局变量声明******/
```

```
uint8_t Key_Status;//按键按下标志

/**********局部变量声明******/
float Key_Cnt1,Key_Cnt2,Key_Cnt3,Key_Cnt4;//按下时间
uint8_t Key_Flag1,Key_Flag2,Key_Flag3,Key_Flag4;//按键按下标志
uint8_t LongPress1,LongPress2,LongPress3,LongPress4;//按键长按标志

/*
**********************************************************************
 * 函数原型：void Check_Press(float dT)
 * 功      能：检测按键按下状态-500ms
**********************************************************************
*/
void Check_Press(float dT)
{
    if(Key_Status)//按键按下
        Key_Status -= dT;//倒计时
}

/*
**********************************************************************
 * 函数原型：void Key_Scan(float dT)
 * 功      能：按键扫描
**********************************************************************
*/
void Key_Scan(float dT)
{
    /**************************************MENU                                键
*****************************************/
    if(Key1)//按下按键
    {
        if(Protect || (sys.Run_Status == 0 && Speed.Display_Rel != 0))
        {
            return;
        }
        if(LongPress1 == 0)//没有长按过
        {
            Key_Cnt1 += dT;//按下时间++
            Key_Flag1 = 1;//按键按下标志置一
        }
    }

    if(Key_Flag1)//按键被按下
    {
        if(!Key1)//抬起按键
        {
            if(Key_Cnt1 > 0.05f && Key_Cnt1 < 1.5)//按键时间大于 0.05S 小于 1.5S 是单击
            {
                #if(Type == 0)
                sys.SetMode_Option++;
```

```
                if(sys.SetMode_Option > 2)
                    sys.SetMode_Option = 0;
                #elif(Type == 1)
                sys.SetMode_Option++;
                if(sys.SetMode_Option > 3)
                    sys.SetMode_Option = 0;
                #endif
                Twinkle_Time = 6.0f;//闪烁时间 6S
                Beep_Time = 0.1;//蜂鸣器响 0.1S
            }
            Key_Flag1 = 0;//按键事件结束，等待下一次按下
            LongPress1 = 0;//长按标志清零
            Key_Cnt1 = 0;//按钮计数清零
        }
        if(Key_Cnt1 > 1.5 && Key_Cnt1 < 3.0)//按键时间大于 1.5S 小于 3S 表示长按
        {
            if(LongPress1 == 0)//如果没有一直一直长按着
            {
                LongPress1 = 1;//长按标志置一
            }
        }
    }

    /************************************ 减 键
************************************/
    if(Key2)//按下按键
    {
        if(Protect)
        {
            return;
        }
        Key_Cnt2 += dT;//按下时间++
        Key_Flag2 = 1;//按键按下标志置一
    }
    if(Key_Flag2 == 1)//按键被按下
    {
        if(!Key2)//抬起按键
        {
            if(Key_Cnt2 > 0.05f && Key_Cnt2 < 1.5)//按键时间大于 0.05S 小于 1.5S 是单击
            {
                if(sys.SetMode_Option == 1)
                {
                    Temp.Set += 10;//温度加 1℃
                    if(Temp.Set > Temp_MAX)//如果大于 Temp_MAX℃
                        Temp.Set = Temp_MAX;//设定温度为 Temp_MAX℃
                    Key_Status = 2.0f;//操作时常亮 2S
                    Twinkle_Time = 6.0f;//闪烁时间 6S
                }
                else if(sys.SetMode_Option == 2)
                {
```

```
                #if(Type == 0)
                Time.Set += 60;//时间加 1 分钟
                if(Time.Set > Time_MAX)//如果时间大于 Time_Max
                    Time.Set = Time_MAX;//设定时间等于 Time_Max
                Key_Status = 2.0f;//操作时常亮 2S
                Twinkle_Time = 6.0f;//闪烁时间 6S
                #elif(Type == 1)
                if(!Speed.Set)
                    Speed.Set += 100;//速度加 100
                else
                    Speed.Set += 50;//速度加 50
                if(Speed.Set > Speed_MAX)//如果速度大于 Speed_MAX
                    Speed.Set = Speed_MAX;//设定速度等于 Speed_MAX
                Key_Status = 2.0f;//操作时常亮 2S
                Twinkle_Time = 6.0f;//闪烁时间 6S
                #endif
            }
            else if(sys.SetMode_Option == 3)
            {
                Time.Set += 60;//时间加 1 分钟
                if(Time.Set > Time_MAX)//如果时间大于 Time_Max
                    Time.Set = Time_MAX;//设定时间等于 Time_Max
                Key_Status = 2.0f;//操作时常亮 2S
                Twinkle_Time = 6.0f;//闪烁时间 6S
            }
        }
        Key_Flag2 = 0;//按键事件结束，等待下一次按下
        Key_Cnt2 = 0;//按钮计数清零
    }
    if(Key_Cnt2 > 1.9 && Key_Cnt2 < 2.1)//按键时间大于 1.9S 小于 2.1S 表示长按
    {
        if(sys.SetMode_Option == 1)
        {
            Temp.Set += 100;//温度加 10℃
            if(Temp.Set > Temp_MAX)//如果大于 Temp_MAX℃
                Temp.Set = Temp_MAX;//设定温度为 Temp_MAX℃
            Key_Status = 2.0f;//操作时常亮 2S
            Twinkle_Time = 6.0f;//闪烁时间 6S
        }
        else if(sys.SetMode_Option == 2)
        {
            #if(Type == 0)
            Time.Set += 1800;//时间加 30 分钟
            if(Time.Set > Time_MAX)//如果时间大于 Time_Max
                Time.Set = Time_MAX;//设定时间等于 Time_Max
            Key_Status = 2.0f;//操作时常亮 2S
            Twinkle_Time = 6.0f;//闪烁时间 6S
            #elif(Type == 1)
            Speed.Set += 100;//速度加 100
            if(Speed.Set > Speed_MAX)//如果速度大于 Speed_MAX
```

```
                    Speed.Set = Speed_MAX;//设定速度等于 Speed_MAX
                Key_Status = 2.0f;//操作时常亮 2S
                Twinkle_Time = 6.0f;//闪烁时间 6S
                #endif
            }
            else if(sys.SetMode_Option == 3)
            {
                Time.Set += 60;//时间加 1 分钟
                if(Time.Set > Time_MAX)//如果时间大于 Time_Max
                    Time.Set = Time_MAX;//设定时间等于 Time_Max
                Key_Status = 2.0f;//操作时常亮 2S
                Twinkle_Time = 6.0f;//闪烁时间 6S
            }
            Key_Flag2 = 0;//按键事件结束，等待下一次按下
            Key_Cnt2 = 1.5;//按钮计数清零
        }
    }

    /************************************ 加 键
****************************************/
    if(Key3)//按下按键
    {
        if(Protect)
        {
            return;
        }
        Key_Cnt3 += dT;//按下时间++
        Key_Flag3 = 1;//按键按下标志置一
    }
    if(Key_Flag3 == 1)//按键被按下
    {
        if(!Key3)//抬起按键
        {
            if(Key_Cnt3 > 0.05f && Key_Cnt3 < 1.5)//按键时间大于 0.05S 小于 1.5S 是单击
            {
                if(sys.SetMode_Option == 1)
                {
                    if(Temp.Set)
                        Temp.Set -= 10;//温度减 1℃
                    if(Temp.Set <= Temp_MIN)//如果小于等于 Temp_MIN℃
                        Temp.Set = Temp_MIN;//设定温度为 Temp_MIN℃
                    Key_Status = 2.0f;//操作时常亮 2S
                    Twinkle_Time = 6.0f;//闪烁时间 6S
                }
                else if(sys.SetMode_Option == 2)
                {
                    #if(Type == 0)
                    if(Time.Set)
                        Time.Set -= 60;//时间减 1 分钟
                    if(Time.Set <= Time_MIN)//如果时间小于等于 Time_MIN
```

```
                    Time.Set = Time_MIN;//设定时间等于 Time_MIN
                Key_Status = 2.0f;//操作时常亮 2S
                Twinkle_Time = 6.0f;//闪烁时间 6S
                #elif(Type == 1)
                if(Speed.Set)
                    Speed.Set -= 50;//速度减 50
                if(Speed.Set < Speed_MIN)//如果速度小于 Speed_MIN
                    Speed.Set = 0;//设定速度等于 0
                Key_Status = 2.0f;//操作时常亮 2S
                Twinkle_Time = 6.0f;//闪烁时间 6S
                #endif
            }
            else if(sys.SetMode_Option == 3)
            {
                if(Time.Set)
                    Time.Set -= 60;//时间减 1 分钟
                if(Time.Set <= Time_MIN)//如果时间小于等于 Time_MIN
                    Time.Set = Time_MIN;//设定时间等于 Time_MIN
                Key_Status = 2.0f;//操作时常亮 2S
                Twinkle_Time = 6.0f;//闪烁时间 6S
            }
        }
        Key_Flag3 = 0;//按键事件结束，等待下一次按下
        Key_Cnt3 = 0;//按钮计数清零
    }
    if(Key_Cnt3 > 1.9 && Key_Cnt3 < 2.1)//按键时间大于 1.9S 小于 2.1S 表示长按
    {
        if(sys.SetMode_Option == 1)
        {
            if(Temp.Set>100)//温度大于 100
                Temp.Set -= 100;//温度减 10℃
            else if(Temp.Set > 0 && Temp.Set<=100)
                Temp.Set -= 10;//温度减 1℃
            if(Temp.Set <= Temp_MIN)//如果小于等于 Temp_MIN℃
                Temp.Set = Temp_MIN;//设定温度为 Temp_MIN℃
            Key_Status = 2.0f;//操作时常亮 2S
            Twinkle_Time = 6.0f;//闪烁时间 6S
        }
        else if(sys.SetMode_Option == 2)
        {
            #if(Type == 0)
            if(Time.Set>1800)
                Time.Set -= 1800;//时间减 30 分钟
            else if(Time.Set > 600 && Time.Set <= 1800)
                Time.Set -= 600;//时间减 10 分钟
            else if(Time.Set >= 60 && Time.Set <= 600)
                Time.Set -= 60;//时间减 1 分钟
            if(Time.Set <= Time_MIN)//如果时间小于等于 Time_MIN
                Time.Set = Time_MIN;//设定时间等于 Time_MIN
            Key_Status = 2.0f;//操作时常亮 2S
```

```
                    Twinkle_Time = 6.0f;//闪烁时间 6S
                    #elif(Type == 1)
                    if(Speed.Set > 100)
                        Speed.Set -= 100;//速度减 100
                    else if(Speed.Set > 50 && Speed.Set <= 100)
                        Speed.Set -= 50;//速度减 50
                    if(Speed.Set < Speed_MIN)//如果速度小于 Speed_MIN
                        Speed.Set = 0;//设定速度等于 0
                    Key_Status = 2.0f;//操作时常亮 2S
                    Twinkle_Time = 6.0f;//闪烁时间 6S
                    #endif
                }
                else if(sys.SetMode_Option == 3)
                {
                    if(Time.Set>1800)
                        Time.Set -= 1800;//时间减 30 分钟
                    else if(Time.Set > 600 && Time.Set <= 1800)
                        Time.Set -= 600;//时间减 10 分钟
                    else if(Time.Set >= 60 && Time.Set <= 600)
                        Time.Set -= 60;//时间减 1 分钟
                    if(Time.Set <= Time_MIN)//如果时间小于等于 Time_MIN
                        Time.Set = Time_MIN;//设定时间等于 Time_MIN
                    Key_Status = 2.0f;//操作时常亮 2S
                    Twinkle_Time = 6.0f;//闪烁时间 6S
                }
                Key_Flag3 = 0;//按键事件结束，等待下一次按下
                Key_Cnt3 = 1.5;//按钮计数清零
            }
        }

        /**************************************Start                                    键
**************************************/
        if(Key4)//按下按键
        {
            if(Protect)
            {
                return;
            }
            if(LongPress4 == 0)//没有长按过
            {
                Key_Cnt4 += dT;//按下时间++
                Key_Flag4 = 1;//按键按下标志置一
            }
        }
        if(Key_Flag4)//按键被按下
        {
            if(!Key4)//抬起按键
            {
                if(Key_Cnt4 > 0.05f && Key_Cnt4 < 1.5)//按键时间大于 0.05S 小于 1.5S 是单击
                {
```

```
                    if(sys.Run_Status)
                    {
                        sys.Run_Status = 0;
                        Speed.DisplayMode = 2;//降速显示
                        Beep_Time = 0.1;//蜂鸣器响 0.1S
                    }
                    else
                    {
                        if(Speed.Set || Time.Set || Temp.Set)//如果 3 个功能都不工作，不开启系
统
                        {
                            sys.Run_Status = 1;
                            Speed_Val.Integral = 25;
                            Judge_TempMode();//温度显示模式检测
                            Judge_SpeedMode();//速度显示模式检测
                            sys.SetMode_Option = 0;
                            Beep_Time = 0.1;//蜂鸣器响 0.1S
                        }
                    }
                }
                Key_Flag4 = 0;//按键事件结束，等待下一次按下
                LongPress4 = 0;//长按标志清零
                Key_Cnt4 = 0;//按钮计数清零
            }
            if(Key_Cnt4 > 1.5 && Key_Cnt4 < 3.0)//按键时间大于 1.5S 小于 3S 表示长按
            {
                if(LongPress4 == 0)//如果没有一直一直长按着
                {
                    LongPress4 = 1;//长按标志置一
                }
            }
        }
}
#include "Drv_Motor.h"

/*
**********************************************************************
 * 函数原型：void Motor_Init(void)
 * 功      能：电机初始化
**********************************************************************
*/
void Motor_Init(void)
{
    HAL_TIM_PWM_Start(&htim1, TIM_CHANNEL_3);//开启 tim1 通道三
}
#include "Drv_NTC.h"

/*********局部变量**********/
const uint16_t R100K_TAB[401] = //R25=100K B25=3950K -25-150
{
```

```
58,//-48
63,//-47
67,//-46
72,//-45
77,//-44
83,//-43
89,//-42
95,//-41
102,//-40
109,//-39
116,//-38
124,//-37
132,//-36
141,//-35
150,//-34
160,//-33
170,//-32
181,//-31
192,//-30
204,//-29
217,//-28
230,//-27
244,//-26
259,//-25
274,//-24
290,//-23
306,//-22
324,//-21
342,//-20
361,//-19
381,//-18
401,//-17
423,//-16
445,//-15
468,//-14
492,//-13
517,//-12
543,//-11
570,//-10
597,//-9
626,//-8
655,//-7
685,//-6
717,//-5
749,//-4
782,//-3
815,//-2
850,//-1
886,//0
922,//1
```

959,//2
997,//3
103,//4
107,//5
1115,//6
1156,//7
1198,//8
1240,//9
1288,//10
1320,//11
1340,//12
1360,//13
1395,//14
1430,//15
1480,//16
1540,//17
1590,//18
1620,//19
1668,//20
1712,//21
1774,//22
1820,//23
1870,//24
1904,//25
1976,//26
2000,//27
2030,//28
2080,//29
2120,//30
2170,//31
2220,//32
2265,//33
2325,//34
2365,//35
2400,//36
2450,//37
2490,//38
2530,//39
2570,//40
2590,//41
2647,//42
2680,//43
2721,//44
2753,//45
2784,//46
2820,//47
2850,//48
2888,//49
2911,//50
2950,//51

```
2977,//52
3007,//53
3034,//54
3050,//55
3080,//56
3105,//57
3130,//58
3154,//59
3181,//60
3218,//61
3230,//62
3267,//63
3295,//64
3313,//65
3338,//66
3348,//67
3366,//68
3388,//69
3409,//70
3425,//71
3450,//72
3461,//73
3486,//74
3495,//75
3505,//76
3528,//77
3545,//78
3555,//79
3565,//80
3574,//81
3581,//82
3597,//83
3607,//84
3620,//85
3631,//86
3641,//87
3651,//88
3664,//89
3677,//90
3682,//91
3694,//92
3707,//93
3718,//94
3727,//95
3732,//96
3741,//97
3751,//98
3756,//99
3764,//100
3795,
```

3803,
3812,
3820,
3827,
3835,
3842,
3849,
3856,
3862,
3869,
3875,
3881,
3887,
3893,
3898,
3903,
3909,
3914,
3919,
3923,
3928,
3932,
3937,
3941,
3945,
3949,
3953,
3956,
3960,
3964,
3967,
3970,
3973,
3977,
3980,
3983,
3985,
3988,
3991,
3993,
3996,
3998,
4001,
4003,
4005,
4008,
4010,
4012,
4014,
4016,

4018,
4020,
4022,
4023,
4025,
4027,
4028,
4030,
4031,
4033,
4034,
4036,
4037,
4039,
4040,
4041,
4042,
4044,
4045,
4046,
4047,
4048,
4049,
4050,
4051,
4052,
4053,
4054,
4055,
4056,
4057,
4058,
4058,
4059,
4060,
4061,
4061,
4062,
4063,
4064,
4064,
4065,
4066,
4066,
4067,
4067,
4068,
4069,
4069,
4070,

4070,
4071,
4071,
4072,
4072,
4073,
4073,
4074,
4074,
4074,
4075,
4075,
4076,
4076,
4076,
4077,
4077,
4077,
4078,
4078,
4079,
4079,
4079,
4079,
4080,
4080,
4080,
4081,
4081,
4081,
4081,
4082,
4082,
4082,
4082,
4083,
4083,
4083,
4083,
4084,
4084,
4084,
4084,
4084,
4085,
4085,
4085,
4085,
4085,
4086,

4086,
4086,
4086,
4086,
4086,
4087,
4087,
4087,
4087,
4087,
4087,
4087,
4088,
4088,
4088,
4088,
4088,
4088,
4088,
4088,
4088,
4089,
4089,
4089,
4089,
4089,
4089,
4089,
4089,
4089,
4090,
4090,
4090,
4090,
4090,
4090,
4090,
4090,
4090,
4090,
4091,
4091,
4091,
4091,
4091,
4091,
4091,
4091,
4091,
4091,

4091,
4091,
4091,
4091,
4092,
4092,
4092,
4092,
4092,
4092,
4092,
4092,
4092,
4092,
4092,
4092,
4092,
4092,
4092,
4092,
4092,
4092,
4092,
4092,
4093,
4093,
4093,
4093,
4093,
4093,
4093,
4093,
4093,
4093,
4093,
4093,
4093,
4093,
4093,
4093,
4093,
4093,
4093,
4093,
4093,
4093,
4093,
4093,
4093,
4093,
4093,
4093,
4093,
4093,
4093,
4093,
4093,

```
    4093
};

/**********全局变量**********/
#define AD_LEN 2//DMA 获取长度
uint16_t ADC_Val[AD_LEN];//adc 的值  0:防干烧。  1：ad 值;
uint16_t ADC_Val1,ADC_Val2;

/*
*********************************************************************
 * 函数原型：int Filter_ADC(void)
 * 功      能：滑动平均值滤波
 * 输      出：滤波后的值
*********************************************************************
*/
#define N 50//采集 100 次
int ADCvalue_Buf[N];//用于储存采集到的 adc 值
int i = 0;
int Filter_ADC(void)
{
    uint8_t count;
    long sum = 0;

    ADCvalue_Buf[i++] = ADC_Val[0];//将 adc 的值储存

    if (i == N)//加入读了 100 组就从新开始
    {
        i = 0;
    }
    for (count = 0; count < N; count++)
    {
        sum += ADCvalue_Buf[count];//100 组相加
    }
    if(ADCvalue_Buf[N-1] == 0)//如果没有读到 100 组就用第一次读到的数
        return ADCvalue_Buf[0];
    else//读到 100 组后
        return (int)(sum / N);//输出平均值
}

/*
*********************************************************************
 * 函数原型：int Filter_ADC1(void)
 * 功      能：滑动平均值滤波
 * 输      出：滤波后的值
*********************************************************************
*/
int ADCvalue_Buf1[N];//用于储存采集到的 adc 值
int j = 0;
int Filter_ADC1(void)
{
```

```
    uint8_t count;
    long sum = 0;

    ADCvalue_Buf1[j++] = ADC_Val[1];//将 adc 的值储存

    if(j == N)//加入读了 100 组就从新开始
    {
        j = 0;
    }
    for(count = 0; count < N; count++)
    {
        sum += ADCvalue_Buf1[count];//100 组相加
    }
    if(ADCvalue_Buf1[N-1] == 0)//如果没有读到 100 组就用第一次读到的数
        return ADCvalue_Buf1[0];
    else//读到 100 组后
        return (int)(sum / N);//输出平均值
}

/*
*********************************************************************
 * 函数原型：uint16_t Get_Ntc_Temp(uint16_t value_adc)
 * 功    能：计算出 Ntc 的温度
 * 输    入: value_adc:adc 读到的值
 * 参    数：uint16_t value_adc
*********************************************************************
*/
#define SHORT_CIRCUIT_THRESHOLD 15
#define OPEN_CIRCUIT_THRESHOLD 4096
uint16_t index_l, index_r ;
uint16_t Get_Ntc_Temp(uint16_t value_adc)
{
    uint16_t R100k_Tab_Size = 401;
    int temp = 0;
    if(value_adc <= SHORT_CIRCUIT_THRESHOLD)
    {
        return 1;
    }
    else if(value_adc >= OPEN_CIRCUIT_THRESHOLD)
    {
        return 2;
    }
    else if(value_adc < R100K_TAB[0])
    {
        return 3;
    }

    else if(value_adc > R100K_TAB[R100k_Tab_Size - 1])
    {
        return 4;
```

```
    }

    index_l = 0;
    index_r = R100k_Tab_Size - 1;
    for(; index_r - index_l > 1;)
    {
        if((value_adc <= R100K_TAB[index_r]) && (value_adc > R100K_TAB[(index_l + index_r) % 2 == 0 ? (index_l + index_r) / 2 : (index_l + index_r) / 2 ]))
        {
            index_l = (index_l + index_r) % 2 == 0 ? (index_l + index_r) / 2 : (index_l + index_r) / 2 ;
        }
        else
        {
            index_r = (index_l + index_r) / 2;
        }
    }
    if(R100K_TAB[index_l] == value_adc)
    {
        temp = (((int)index_l) - 48) * 10; //rate *10
    }
    else if(R100K_TAB[index_r] == value_adc)
    {
        temp = (((int)index_r) - 48) * 10; //rate *10
    }
    else
    {
        if(R100K_TAB[index_r] - R100K_TAB[index_l] == 0)
        {
            temp = (((int)index_l) - 48) * 10; //rate *10
        }
        else
        {
            temp = (((int)index_l) - 48) * 10 + ((value_adc - R100K_TAB[index_l] ) * 100 + 5) / 10 / (R100K_TAB[index_r] - R100K_TAB[index_l]);
        }
    }

    /******************温度补偿**********************/
    return temp;
}

/*
*********************************************************************
 * 函数原型：uint16_t Get_ADCVal(uint16_t temp)
 * 功    能：计算当前温度的 adc 值
 * 输    入：temp：当前温度
 * 输    出：ADC 值
 * 参    数：uint16_t temp
*********************************************************************
```

```
*/

uint16_t Get_ADCVal(uint16_t temp)
{
	uint16_t adc,adc1;
	float val2;
	uint16_t val3,val1;

	val3 = (temp/10)+48;
	val2 = (float)(temp%10)/10;

	val1 = ((temp+10)/10)+47;

	adc = R100K_TAB[val3];
	adc1 = R100K_TAB[val1];
	return adc+((adc1-adc)*val2);
}

/*
**********************************************************************
 * 函数原型：void ADCDMA_Init(void)
 * 功　　能：ADC 和 DMA 的初始化
**********************************************************************
*/
void ADCDMA_Init(void)
{
	HAL_TIM_Base_Start_IT(&htim15);//开启 TIM3 的定时，用于刷新
	HAL_ADC_Start_DMA(&hadc,(uint32_t *)ADC_Val, AD_LEN);//用 DMA 获取 adc 值
	HAL_ADCEx_Calibration_Start(&hadc);
	for(uint8_t i=0;i<10;i++)
	{
		Temp.Alarm	= Get_Ntc_Temp(ADC_Val[0]);//计算温度
		Temp.Rel = Get_Ntc_Temp(ADC_Val[1]);//计算温度
		HAL_Delay(10);
	}
}

/*
**********************************************************************
 * 函数原型：	void Read_Temp(float dT)
 * 功　　能：	读取温度-10ms
**********************************************************************
*/
uint16_t RelTemp_Old,AlarmTemp_Old;
void Read_Temp(float dT)
{
	static float T;
	T += dT;
	ADC_Val1 = Filter_ADC();//滤波获取 adc 的滑动平均值
	ADC_Val2 = Filter_ADC1();//滤波获取 adc 的滑动平均值
```

```
    if(T >= 1.0f)//1S
    {
        RelTemp_Old = Temp.Rel;
        AlarmTemp_Old = Temp.Alarm;
        Temp.Alarm = Get_Ntc_Temp(ADC_Val1);//计算温度
        Temp.Rel = Get_Ntc_Temp(ADC_Val2);//计算温度
        Temp.Rel_Speed = safe_div((float)Temp.Rel - (float)RelTemp_Old,2.0f,0);//计算实际温度速度
        Temp.Alarm_Speed = safe_div((float)Temp.Alarm - (float)AlarmTemp_Old,2.0f,0);//计算报警温度速度
        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;

    Param.type = Type;//保存类型
    Param.Speed = 100;//转速 100
    Param.Temp = 500;//温度 50℃
    Param.Time = 0;//时间常动

    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.type != Type)
    {
        Param_Reset();
        Speed.Set = Param.Speed;//将 Flash 中的速度赋值
        Temp.Set = Param.Temp;//将 Flash 中的温度赋值
        Time.Set = Param.Time;//将 Flash 中的时间赋值
        SetOK_Flag = 1;
        Save_Param_En = 1;
    }
    else
    {
        Speed.Set = Param.Speed;//将 Flash 中的速度赋值
        Temp.Set = Param.Temp;//将 Flash 中的温度赋值
        Time.Set = Param.Time;//将 Flash 中的时间赋值
        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;//检测到设置，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)
            {
                Speed.Ctrl = Speed.Set;
                Param.Speed = Speed.Set;
            }
            if(Temp.Ctrl != Temp.Set)
            {
                Temp.Ctrl = Temp.Set;
                Param.Temp = Temp.Set;
            }
            if(Time.Ctrl != Time.Set)
            {
                Time.Ctrl = Time.Set;
                Time.Rel = Time.Set;
                Param.Time = Time.Set;
            }
            if(sys.Run_Status && Speed.Ctrl)
            {
```

```
                    Judge_SpeedMode();//速度显示模式检测
                }
                if(sys.Run_Status && Temp.Ctrl)
                {
                    Judge_TempMode();//温度显示模式检测
                }
                Save_Param_En = 1;//保存
                SetOK_Flag = 0;
            }
        }
}
#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 Protect_ShowFlag;//防干烧图标
uint8_t Tab[] = {0x77,0x24,0x5D,0x6D,0x2E,0x6B,0x7B,0x25,0x7F,0x6F};//0·9
uint8_t Tab1[] = {0x77,0x12,0x5D,0x5B,0x3A,0x6B,0x6F,0x52,0x7F,0x7B};//0·9
uint8_t Tab2[] = {0xEE,0x24,0xBA,0xB6,0x74,0xD6,0xDE,0xA4,0xFE,0xF6};//0·9

/*
**********************************************************************
 * 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
    {
        if(T == 0)
        {
            T += dT;
            Twinkle_Time -= 0.5f;//闪烁计时
```

```
                if(Twinkle_Time == 0)//如果闪烁结束
                {
                    sys.SetMode_Option = 0;//模式选择清零
                    Speed_ShowFlag = 0;//常亮
                    Temp_ShowFlag = 0;//常亮
                    Time_ShowFlag = 0;//常亮
                }
                if(sys.SetMode_Option == 1)//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
                }
                else if(sys.SetMode_Option == 2)//设置速度
                {
                    #if(Type == 0)
                    Time_ShowFlag = ~Time_ShowFlag;//时间闪烁
                    Speed_ShowFlag = 0;//速度常亮
                    Temp_ShowFlag = 0;//温度常亮
                    #elif(Type == 1)
                    Time_ShowFlag = 0;//时间常亮
                    Speed_ShowFlag = ~Speed_ShowFlag;//速度闪烁
                    Temp_ShowFlag = 0;//温度常亮
                    #endif
                }
                else if(sys.SetMode_Option == 3)//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
            {
                T += dT;
                if(T >=0.5f)
                    T = 0;
            }
        }
        else
        {
            Speed_ShowFlag = 0;//常亮
            Temp_ShowFlag = 0;//常亮
            Time_ShowFlag = 0;//常亮
            T = 0;
        }
}

/*
*********************************************************************
```

```
 * 函数原型：static void Start_SpeedRun(float dT)
 * 功      能：转速动画
 * 调      用：内部调用
**********************************************************************
*/
static void Start_SpeedRun(float dT)
{
    static float T;
    if(!sys.Run_Status || !Speed.Ctrl)
    {
        Speed.IcnStep = 0;
        T = 0;
        return;
    }
    T += dT;
    if(T >= 0.5)
    {
        Speed.IcnStep++;
        if(Speed.IcnStep > 3)
            Speed.IcnStep = 1;
        T = 0;
    }
}

/*
**********************************************************************
 * 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)
        {
            Time.Icn = ~Time.Icn;//时间图标闪烁;
            T = 0;
        }
    }
    else
    {
        Time.Icn = 0;//显示时间图标
    }
}

/*
```

```
*************************************************************************
 * 函数原型：static void Temp_Twinkle(float dT)
 * 功      能：温度图标闪烁
 * 调      用：内部调用
*************************************************************************
*/
static void Temp_Twinkle(float dT)
{
    static float T;
    if(sys.Run_Status)
    {
        T += dT;
        if(T >= 0.5f)
        {
            Temp.Icn = ~Temp.Icn;//温度图标闪烁;
            T = 0;
        }
    }
    else
    {
        Temp.Icn = 0;//显示温度图标
    }
}

/*
*************************************************************************
 * 函数原型：static void Protect_Twinkle(float dT)
 * 功      能：防干烧图标闪烁
 * 调      用：内部调用
*************************************************************************
*/
static void Protect_Twinkle(float dT)
{
    static float T;
    if(Protect)
    {
        T += dT;
        if(T >= 0.5f)
        {
            Protect_ShowFlag = ~Protect_ShowFlag;//温度图标闪烁;
            T = 0;
        }
    }
    else
    {
        Protect_ShowFlag = 0;//显示温度图标
    }
}

/*
```

```
*****************************************************************
 * 函数原型：void Twinkle(float dT)
 * 功      能：闪烁函数
*****************************************************************
*/
void Twinkle(float dT)
{
    Check_ShowFlag(dT);//时间图标闪烁
    Start_SpeedRun(dT);//转速动画
    Time_Twinkle(dT);//时间图标闪烁
    Temp_Twinkle(dT);//温度图标闪烁
    Protect_Twinkle(dT);//防干烧图标闪烁
}

/*
*****************************************************************
 * 函数原型：void Display_RelVal(uint16_t val1, uint32_t val2, uint16_t val3)
 * 功      能：显示实际的值
 * 输      入: val1：显示左边的数值;val2：显示中间的数值;val3：显示右边的数值
 * 参      数：uint16_t val1, uint32_t val2, uint16_t val3
*****************************************************************
*/
void Display_RelVal(uint16_t val1, uint32_t val2, uint16_t val3)
{
    uint8_t seg1,seg2,seg3,seg4,seg5,seg6,seg7,seg8;
    seg1=0;seg2=0;seg3=0;seg4=0;seg5=0;seg6=0;seg7=0;seg8=0;
    uint16_t Val;//用于百十个取出来的数字

    /*****************屏幕设定左边数值******************/
    /***********val1 百位***********/
    if(val1 > 999)//大于 999 时
    {
        Val=val1/1000;//除以 1000
        if(val1 > 9)//如果大于 9，那么说明除以 1000 后是两位数
            Val=Val%10;//除以 10 取余，得出千位上的数
        seg1&=0x80;seg1|=Tab[Val];
    }
    else
    {
        seg1&=0x80;seg1|=0x00;
    }

    /***********val1 十位***********/
    if(val1 > 99)//大于 99 时
    {
        Val=val1/100;//除以 100
        if(val1 > 9)//如果大于 9，那么说明除以 100 后是两位数
            Val=Val%10;//除以 10 取余，得出百位上的数
        seg2&=0x80;seg2|=Tab[Val];
    }
```

```
    else
    {
        seg2&=0x80;seg2|=0x00;
    }

    /***********val1 个位***********/
    if(val1 > 9)//大于 9 时
    {
        Val=val1/10;//除以 10
        if(val1 > 9)//如果大于 9，那么说明除以 10 后是两位数
            Val=Val%10;//除以 10 取余，得出十位上的数
        seg3&=0x80;seg3|=Tab[Val];
    }
    else
    {
        seg3&=0x80;seg3|=Tab[0];
    }

    /***********val1 小数位***********/
    Val=val1%10;//取出个位
    seg4&=0x80;seg4|=Tab[Val];
    seg4&=0x7F;seg4|=0x80;//显示温度的小数点

#if(Type == 0)
    uint8_t SH,H,SM,M;//时间的单位取值
    if(val2 && Time.Set != 0)
    {
        SH=val2/3600/10;//计算十位单位的小时数
        H=val2/3600%10;//计算个位单位的小时数
        SM=val2%3600/60/10;//计算十分位单位的分钟数
        M=val2%3600/60%10;//计算个分位单位的分钟数
        seg5&=0x80;seg5|=Tab[SH];
        seg6&=0x80;seg6|=Tab[H];
        seg7&=0x80;seg7|=Tab[SM];
        seg8&=0x80;seg8|=Tab[M];
    }
    else
    {
        seg5&=0x80;seg5|=0x08;
        seg6&=0x80;seg6|=0x08;
        seg7&=0x80;seg7|=0x08;
        seg8&=0x80;seg8|=0x08;
    }
    /*****************时间冒号******************/
    seg7&=0x7f;seg7|=0x80;
#elif(Type == 1)
    /*****************屏幕设定右边数值******************/
    /***********val3 千位***********/
    if(val3 > 999)//大于 999 时
    {
```

```
        Val=val3/1000;//除以 1000
        if(val3 > 9)//如果大于 9，那么说明除以 1000 后是两位数
            Val=Val%10;//除以 10 取余，得出千位上的数
        seg5&=0x80;seg5|=Tab[Val];
    }
    else
    {
        seg5&=0x80;seg5|=0x00;
    }

    /***********val3 百位***********/
    if(val3 > 99)//大于 99 时
    {
        Val=val3/100;//除以 100
        if(val3 > 9)//如果大于 9，那么说明除以 100 后是两位数
            Val=Val%10;//除以 10 取余，得出百位上的数
        seg6&=0x80;seg6|=Tab[Val];
    }
    else
    {
        seg6&=0x80;seg6|=0x00;
    }

    /***********val3 十位***********/
    if(val3 > 9)//大于 9 时
    {
        Val=val3/10;//除以 10
        if(val3 > 9)//如果大于 9，那么说明除以 10 后是两位数
            Val=Val%10;//除以 10 取余，得出十位上的数
        seg7&=0x80;seg7|=Tab[Val];
    }
    else
    {
        seg7&=0x80;seg7|=0x00;
    }

    /***********val3 个位***********/
    Val=val3%10;//取出个位
    seg8&=0x80;seg8|=Tab[Val];

    /*****************转速图标******************/
    switch(Speed.IcnStep)
    {
        case 0:seg5&=0x7F;seg5|=0x80;seg6&=0x7F;seg6|=0x80;seg8&=0x7F;seg8|=0x80;//常
亮
            break;
        case 1:seg5&=0x7F;seg5|=0x80;seg6&=0x7F;seg6|=0x80;seg8&=0x7F;seg8|=0x00;//常
亮
            break;
        case 2:seg5&=0x7F;seg5|=0x80;seg6&=0x7F;seg6|=0x00;seg8&=0x7F;seg8|=0x80;//常
```

```
亮
                break;
        case 3:seg5&=0x7F;seg5|=0x00;seg6&=0x7F;seg6|=0x80;seg8&=0x7F;seg8|=0x80;//常
亮
                break;
        default:
                break;
    }
#endif

    /******************温度图标*******************/
    if(!Temp.Icn || !Temp.Ctrl)
    {
        seg2&=0x7F;seg2|=0x80;seg3&=0x7F;seg3|=0x80;
    }
    else
    {
        seg2&=0x7F;seg2|=0x00;seg3&=0x7F;seg3|=0x00;
    }

    /******************时间图标*******************/
    if(!Time.Icn || !Time.Rel)
    {
        seg1&=0x7f;seg1|=0x80;
    }
    else
    {
        seg1&=0x7f;seg1|=0x00;
    }

    if(Protect)
    {
        if(Protect_ShowFlag)
        {
            seg1 = 0xFF;
            seg2 = 0xFF;
            seg3 = 0xFF;
            seg4 = 0xFF;
            seg5 = 0xFF;
            seg6 = 0xFF;
            seg7 = 0xFF;
            seg8 = 0xFF;
        }
        else
        {
            seg1 = 0x00;
            seg2 = 0x00;
            seg3 = 0x00;
            seg4 = 0x00;
            seg5 = 0x00;
```

```
                seg6 = 0x00;
                seg7 = 0x00;
                seg8 = 0x00;
            }
        }
        Write_Addr_Dat_N(0,seg1,1);
        Write_Addr_Dat_N(2,seg2,1);
        Write_Addr_Dat_N(4,seg3,1);
        Write_Addr_Dat_N(6,seg4,1);
        Write_Addr_Dat_N(8,seg5,1);
        Write_Addr_Dat_N(10,seg6,1);
        Write_Addr_Dat_N(12,seg7,1);
        Write_Addr_Dat_N(14,seg8,1);
}

/*
*********************************************************************
 * 函数原型：void Display_SetVal(uint16_t val1, uint32_t val2, uint16_t val3)
 * 功      能：显示设定的值
 * 输      入: val1：显示左边的数值;val2：显示右边的数值;val3：显示右边的数值
 * 参      数：uint16_t val1, uint32_t val2, uint16_t val3
*********************************************************************
*/
void Display_SetVal(uint16_t val1, uint32_t val2, uint16_t val3)
{
    uint8_t seg1,seg2,seg3,seg4,seg5,seg6,seg7,seg8,seg9,seg10,seg11,seg12;

seg1=0;seg2=0;seg3=0;seg4=0;seg5=0;seg6=0;seg7=0;seg8=0;seg9=0;seg10=0;seg11=0;seg12=0;
    uint16_t Val;//用于百十个取出来的数字
    uint8_t SH,H,SM,M;//时间的单位取值

    /*****************屏幕设定左边数值******************/
    /***********val1 百位***********/
    seg3&=0x7F;seg3|=0x80;//温度单位图标
    seg4&=0xFE;seg4|=0x01;//温度小数点
    if(!Temp_ShowFlag)
    {
        if(val1)
        {
            if(val1 > 999)//大于 999 时
            {
                Val=val1/1000;//除以 1000
                if(val1 > 9)//如果大于 9，那么说明除以 1000 后是两位数
                    Val=Val%10;//除以 10 取余，得出千位上的数
                seg1&=0x80;seg1|=Tab1[Val];
            }
            else
            {
                seg1&=0x80;seg1|=0x00;
            }
```

```
            /***********val1 十位***********/
            if(val1 > 99)//大于 99 时
            {
                Val=val1/100;//除以 100
                if(val1 > 9)//如果大于 9，那么说明除以 100 后是两位数
                    Val=Val%10;//除以 10 取余，得出百位上的数
                seg2&=0x80;seg2|=Tab1[Val];
            }
            else
            {
                seg2&=0x80;seg2|=0x00;
            }


            /***********val1 个位***********/
            if(val1 > 9)//大于 9 时
            {
                Val=val1/10;//除以 10
                if(val1 > 9)//如果大于 9，那么说明除以 10 后是两位数
                    Val=Val%10;//除以 10 取余，得出十位上的数
                seg3&=0x80;seg3|=Tab1[Val];
            }
            else
            {
                seg3&=0x80;seg3|=Tab1[0];
            }

            /***********val1 小数位***********/
            Val=val1%10;//取出个位
            seg4&=0x01;seg4|=Tab2[Val];
        }
        else//显示“----”
        {
            seg1&=0x80;seg1|=0x08;
            seg2&=0x80;seg2|=0x08;
            seg3&=0x80;seg3|=0x08;
            seg4&=0x80;seg4|=0x10;
        }
    }
    else//闪烁不显示
    {
        seg1&=0x80;seg1|=0x00;
        seg2&=0x80;seg2|=0x00;
        seg3&=0x80;seg3|=0x00;
        seg4&=0x00;seg4|=0x00;
    }

#if(Type == 0)
    /*****************屏幕设定右边数值******************/
```

```
    if(!Time_ShowFlag)
    {
        if(val2)
        {
            SH=val2/3600/10;//计算十位单位的小时数
            H=val2/3600%10;//计算个位单位的小时数
            SM=val2%3600/60/10;//计算十分位单位的分钟数
            M=val2%3600/60%10;//计算个分位单位的分钟数
            seg9&=0x80;seg9|=Tab1[SH];
            seg10&=0x80;seg10|=Tab1[H];
            seg11&=0x80;seg11|=Tab1[SM];
            seg12&=0x80;seg12|=Tab1[M];
        }
        else
        {
            seg9&=0x80;seg9|=0x08;//显示‘-’
            seg10&=0x80;seg10|=0x08;//显示‘-’
            seg11&=0x80;seg11|=0x08;//显示‘-’
            seg12&=0x80;seg12|=0x08;//显示‘-’
        }
    }
    else
    {
        seg9&=0x80;seg9|=0x00;
        seg10&=0x80;seg10|=0x00;
        seg11&=0x80;seg11|=0x00;
        seg12&=0x80;seg12|=0x00;
    }
    /*****************时间冒号******************/
    seg10&=0x7f;seg10|=0x80;
    /*****************时间单位******************/
    seg12&=0x7f;seg12|=0x80;//显示 min

#elif(Type == 1)
    /*****************屏幕设定中间数值******************/
    if(Time.Set || sys.SetMode_Option == 3)//设定时间大于 0，在设定时间和闪烁下
    {
        if(!Time_ShowFlag)
        {
            if(Time.Set)//假如设定时间大于 0
            {
                SH=val2/3600/10;//计算十位单位的小时数
                H=val2/3600%10;//计算个位单位的小时数
                SM=val2%3600/60/10;//计算十分位单位的分钟数
                M=val2%3600/60%10;//计算个分位单位的分钟数
                seg5&=0x80;seg5|=Tab1[SH];
                seg6&=0x80;seg6|=Tab1[H];
                seg7&=0x80;seg7|=Tab1[SM];
                seg8&=0x80;seg8|=Tab1[M];
            }
```

```
            else
            {
                seg5&=0x80;seg5|=0x08;//显示‘-’
                seg6&=0x80;seg6|=0x08;//显示‘-’
                seg7&=0x80;seg7|=0x08;//显示‘-’
                seg8&=0x80;seg8|=0x08;//显示‘-’
            }
        }
        else//设定时间等于 0
        {
            seg5&=0x80;seg5|=0x00;
            seg6&=0x80;seg6|=0x00;
            seg7&=0x80;seg7|=0x00;
            seg8&=0x80;seg8|=0x00;
        }
    }
    else//设定时间等于 0
    {
        seg5&=0x80;seg5|=0x00;
        seg6&=0x80;seg6|=0x00;
        seg7&=0x80;seg7|=0x00;
        seg8&=0x80;seg8|=0x00;
    }

    if(Time.Set > 0 || sys.SetMode_Option == 3)
    {
        /******************时间冒号******************/
        seg6&=0x7f;seg6|=0x80;
        /******************时间单位******************/
        // seg8&=0x7f;seg8|=0x80;//显示 sec
        seg7&=0x7f;seg7|=0x80;//显示 min
    }
    else
    {
        seg6&=0x7f;seg6|=0x00;
        seg7&=0x7f;seg7|=0x00;
    }

    /******************屏幕设定右边数值******************/
    if(!Speed_ShowFlag)
    {
        if(Speed.Set)
        {
            /***********val3 千位***********/
            if(val3 > 999)//大于 999 时
            {
                Val=val3/1000;//除以 1000
                if(val3 > 9)//如果大于 9，那么说明除以 1000 后是两位数
                    Val=Val%10;//除以 10 取余，得出千位上的数
                seg9&=0x80;seg9|=Tab1[Val];
```

```
            }
            else
            {
                seg9&=0x80;seg9|=0x00;
            }

            /***********val3 百位***********/
            if(val3 > 99)//大于 99 时
            {
                Val=val3/100;//除以 100
                if(val3 > 9)//如果大于 9，那么说明除以 100 后是两位数
                    Val=Val%10;//除以 10 取余，得出百位上的数
                seg10&=0x80;seg10|=Tab1[Val];
            }
            else
            {
                seg10&=0x80;seg10|=0x00;
            }

            /***********val3 十位***********/
            if(val3 > 9)//大于 9 时
            {
                Val=val3/10;//除以 10
                if(val3 > 9)//如果大于 9，那么说明除以 10 后是两位数
                    Val=Val%10;//除以 10 取余，得出十位上的数
                seg11&=0x80;seg11|=Tab1[Val];
            }
            else
            {
                seg11&=0x80;seg11|=0x00;
            }

            /***********val3 个位***********/
            Val=val3%10;//取出个位
            seg12&=0x80;seg12|=Tab1[Val];
        }
        else
        {
            seg9&=0x80;seg9|=0x08;
            seg10&=0x80;seg10|=0x08;
            seg11&=0x80;seg11|=0x08;
            seg12&=0x80;seg12|=0x08;
        }
    }
    else
    {
        seg9&=0x80;seg9|=0x00;
        seg10&=0x80;seg10|=0x00;
        seg11&=0x80;seg11|=0x00;
        seg12&=0x80;seg12|=0x00;
```

```
    }
    /******************速度单位******************/
    seg11&=0x7F;seg11|=0x80;
#endif
    if(Protect)
    {
        if(Protect_ShowFlag)
        {
            seg1 = 0xFF;
            seg2 = 0xFF;
            seg3 = 0xFF;
            seg4 = 0xFF;
            seg5 = 0xFF;
            seg6 = 0xFF;
            seg7 = 0xFF;
            seg8 = 0xFF;
            seg9 = 0xFF;
            seg10 = 0xFF;
            seg11 = 0xFF;
            seg12 = 0xFF;
        }
        else
        {
            seg1 = 0x00;
            seg2 = 0x00;
            seg3 = 0x00;
            seg4 = 0x00;
            seg5 = 0x00;
            seg6 = 0x00;
            seg7 = 0x00;
            seg8 = 0x00;
            seg9 = 0x00;
            seg10 = 0x00;
            seg11 = 0x00;
            seg12 = 0x00;
        }
    }

    Write_Addr_Dat_N(38,seg1,1);
    Write_Addr_Dat_N(36,seg2,1);
    Write_Addr_Dat_N(34,seg3,1);
    Write_Addr_Dat_N(32,seg4,1);
    Write_Addr_Dat_N(30,seg5,1);
    Write_Addr_Dat_N(28,seg6,1);
    Write_Addr_Dat_N(26,seg7,1);
    Write_Addr_Dat_N(24,seg8,1);
    Write_Addr_Dat_N(22,seg9,1);
    Write_Addr_Dat_N(20,seg10,1);
    Write_Addr_Dat_N(18,seg11,1);
    Write_Addr_Dat_N(16,seg12,1);
```

```
}

/*
*********************************************************************
 * 函数原型：void Judge_SpeedMode(void)
 * 功      能：判断速度显示的模式
*********************************************************************
*/
void Judge_SpeedMode(void)
{
    if(!Speed.Set)
    {
        Speed.DisplayMode = 0;
    }
    else if(Speed.Set > Speed.Rel)//升速
    {
        Speed.DisplayMode = 1;
    }
    else//降速
    {
        Speed.DisplayMode = 2;
    }
}

/*
*********************************************************************
 * 函数原型：void Judge_TempMode(void)
 * 功      能：判断温度显示的模式
*********************************************************************
*/
void Judge_TempMode(void)
{
    if(!Temp.Ctrl)
        Temp.DisplayMode = 0;
    else if(Temp.Ctrl > Temp.Rel)
        Temp.DisplayMode = 1;//升温
    else if(Temp.Ctrl < Temp.Rel)
        Temp.DisplayMode = 2;//保温
}

/*
*********************************************************************
 * 函数原型：void Deal_Speed(float dT)
 * 功      能：速度显示处理
*********************************************************************
*/
void Deal_Speed(float dT)
{
    switch(Speed.DisplayMode)
    {
```

```
            case 0:Speed.Display_Rel = 0;
                break;
            case 1://升速时
                    if(Speed.Display_Rel < Speed.Rel && !sys.SetMode_Option)
                        Speed.Display_Rel ++;
                    if(Speed.Ctrl == 1200 && Speed.Rel > 1140)
                        Speed.Display_Rel ++;
                    if(Speed.Display_Rel >= Speed.Ctrl)
                        Speed.DisplayMode = 3;
                break;
            case 2://降速时
                    if(Speed.Display_Rel > Speed.Rel && !sys.SetMode_Option)
                        Speed.Display_Rel --;
                    if(sys.Run_Status && (Speed.Display_Rel <= Speed.Ctrl))
                        Speed.DisplayMode = 3;
                break;
            case 3: Speed.Display_Rel = Speed.Ctrl;
                break;
        }
}

/*
**********************************************************************
 * 函数原型：void Deal_Temp(float dT)
 * 功      能：温度显示处理
**********************************************************************
*/
void Deal_Temp(float dT)
{
    if(sys.Run_Status && !Display_Res && Temp.Set != 0)//系统开启，加热功能开启
    {
        if(Temp.DisplayMode == 0)
        {
            Temp.Display_Rel = Temp.Rel;
        }
        else if(Temp.DisplayMode == 1)
        {
            if(Temp.Display_Rel <= Temp.Rel)
            {
                Temp.Display_Rel ++;
            }
            if(Temp.Display_Rel >= Temp.Ctrl)
            {
                Temp.DisplayMode = 3;
            }
        }
        else if(Temp.DisplayMode == 2)
        {
            if(Temp.Display_Rel >= Temp.Rel)
            {
```

```
                    Temp.Display_Rel --;
                }
                if(Temp.Display_Rel <= Temp.Ctrl)
                {
                    Temp.DisplayMode = 3;
                }
            }
            else if(Temp.DisplayMode == 3)
            {
                Temp.Display_Rel = Temp.Ctrl;
            }
        }
        else
        {
            Temp.Display_Rel = Temp.Rel;
            Temp.DisplayMode = 0;
        }
}

/*
**********************************************************************
 * 函数原型：void Show_Display(void)
 * 功      能：显示屏幕内容
**********************************************************************
*/
void Show_Display(void)
{
    Temp.Display_Set = Temp.Set;
    Speed.Display_Set = Speed.Set;

#if(Type == 0)
    Time.Display_Set = Time.Set;
    if(sys.Run_Status)
        Time.Display_Rel = Time.Rel+59;
    else
        Time.Display_Rel = Time.Rel;
#elif(Type == 1)
    if(sys.Run_Status && !sys.SetMode_Option)
        Time.Display_Set = Time.Rel+59;
    else
        Time.Display_Set = Time.Set;
#endif
    if(Temp.Display_Rel > 1000)
        Temp.Display_Rel = 1000;
    Display_RelVal(Temp.Display_Rel,Time.Display_Rel,Speed.Display_Rel);
    Display_SetVal(Temp.Display_Set,Time.Display_Set,Speed.Display_Set);
}
```