MHS3200_4 软件源程序

```
#include "SetVal.h"

/**********全局变量声明******/
uint8_t SetOK_Flag;//检测是否波动旋钮和设置标志位

/*
*****************************************************************
  *  函数原型：  void Check_Set(void)
  *  功        能：  检测设置
*****************************************************************
*/
void Check_Set(void)
{
	if(EC11A_Knob != 0)
	{
		SetOK_Flag = 1;//检测到波动旋钮，等待退出设置模式
	}
	if(SetOK_Flag == 1)
	{
		if(SetMode_Option == 0)//在设定好后
		{
			Set_Speeds(&Speed,&Set_Speed,&Ctrl_Speed);//比较临时速度，不同就将设置值赋值
			Set_Speeds(&Speed,&Set_Speed,&Display_Speed);//比较临时速度，不同就将设置值赋值
			Set_Speeds(&Speed,&Set_Speed,&Speed);//比较临时速度,不同就将设置值赋值

			Set_Times(&Time,&Set_Time,&Rel_Time);//比较临时时间，不同就将设置值赋值
			Set_Times(&Time,&Set_Time,&Ctrl_Time);//比较临时时间，不同就将设置值赋值
			Set_Times(&Time,&Set_Time,&Time);//比较临时时间，不同就将设置值赋值

			if(Temp != Set_Temp)//比较临时温度和设定温度不一样
			{
				Ctrl_Temp = Set_Temp;//将设置温度赋值给控制温度
				Temp = Ctrl_Temp;//将设置温度赋值给临时温度
			}

			if(Temp_ADDMode == 1 && Rel_Temp > Ctrl_Temp)//在加热模式下，显示温度大于控制温度
			{
				Temp_ADDMode = 0;//重新判断
			}
			else if(Temp_ADDMode == 2 && Rel_Temp < Ctrl_Temp)//在降温模式下，显示温度小于控制温度
			{
				Temp_ADDMode = 0;//重新判断
			}
			else if(Temp_ADDMode == 3)//在温度状态下
```

```
            {
                Temp_ADDMode = 0;//重新判断
            }

            SetOK_Flag = 0;
        }
    }
}

/*
*****************************************************************
 * 函数原型： void Set_Val(uint8_t flag,uint8_t Work_Option,uint8_t SetMode_Option)
 * 功      能： 设置数值
 * 输      入: flag ：0是加 1是减   Work_Option：工位 SetMode_Option：设置模式
 * 参      数： uint8_t flag,uint8_t Work_Option,uint8_t SetMode_Option
*****************************************************************
*/
void Set_Val(uint8_t flag,uint8_t Work_Option,uint8_t SetMode_Option)
{
    if(flag == 0)//加
    {
        switch(Work_Option)//工位
        {
            case 1:switch(SetMode_Option)//模式
                   {
                        case 1:Set_Speed.L1 = Set_Speed.L1 + 10;//速度加10
                               Set_Speed.L1 = (Set_Speed.L1 < 50) ? 50 :
Set_Speed.L1;//小于50时从50开始加
                               break;
                        case 2:Set_Temp = Set_Temp + 10;//温度加1℃
                               break;
                        case 3:Set_Time.L1 = Set_Time.L1 + 60;//时间加一分钟
                               break;
                   }break;
            case 2:switch(SetMode_Option)//模式
                   {
                        case 1:Set_Speed.L2 = Set_Speed.L2 + 10;//速度加10
                                Set_Speed.L2 = (Set_Speed.L2 < 50) ? 50 :
Set_Speed.L2;//小于50时从50开始加
                               break;
                        case 2:Set_Temp = Set_Temp + 10;//温度加1℃
                               break;
                        case 3:Set_Time.L2 = Set_Time.L2 + 60;//时间加一分钟
                               break;
                   }break;
            case 3:switch(SetMode_Option)//模式
                   {
                        case 1:Set_Speed.L3 = Set_Speed.L3 + 10;//速度加10
                                Set_Speed.L3 = (Set_Speed.L3 < 50) ? 50 :
Set_Speed.L3;//小于50时从50开始加
```

```
                                break;
                        case 2:Set_Temp = Set_Temp + 10;//温度加 1℃
                                break;
                        case 3:Set_Time.L3 = Set_Time.L3 + 60;//时间加一分钟
                                break;
                }break;
        case 4:switch(SetMode_Option)//模式
                {
                        case 1:Set_Speed.L4 = Set_Speed.L4 + 10;//速度加 10
                                  Set_Speed.L4 = (Set_Speed.L4 < 50) ? 50 :
Set_Speed.L4;//小于 50 时从 50 开始加
                                break;
                        case 2:Set_Temp = Set_Temp + 10;//温度加 1℃
                                break;
                        case 3:Set_Time.L4 = Set_Time.L4 + 60;//时间加一分钟
                                break;
                }break;
        case 5:switch(SetMode_Option)//模式
                {
                        case 1:Set_Speed.L5 = Set_Speed.L5 + 10;//速度加 10
                                  Set_Speed.L5 = (Set_Speed.L5 < 50) ? 50 :
Set_Speed.L5;//小于 50 时从 50 开始加
                                break;
                        case 2:Set_Temp = Set_Temp + 10;//温度加 1℃
                                break;
                        case 3:Set_Time.L5 = Set_Time.L5 + 60;//时间加一分钟
                                break;
                }break;
        case 6:switch(SetMode_Option)//模式
                {
                        case 1:Set_Speed.L6 = Set_Speed.L6 + 10;//速度加 10
                                  Set_Speed.L6 = (Set_Speed.L6 < 50) ? 50 :
Set_Speed.L6;//小于 50 时从 50 开始加
                                break;
                        case 2:Set_Temp = Set_Temp + 10;//温度加 1℃
                                break;
                        case 3:Set_Time.L6 = Set_Time.L6 + 60;//时间加一分钟

                                break;
                }break;
        case 7:switch(SetMode_Option)//模式
                {
                        case 1:Set_Speed.L7 = Set_Speed.L7 + 10;//速度加 10
                                  Set_Speed.L7 = (Set_Speed.L7 < 50) ? 50 :
Set_Speed.L7;//小于 50 时从 50 开始加
                                break;
                        case 2:Set_Temp = Set_Temp + 10;//温度加 1℃

                                break;
                        case 3:Set_Time.L7 = Set_Time.L7 + 60;//时间加一分钟
```

```
                                                break;
                                    }break;
                        case 8:switch(SetMode_Option)//模式
                                    {
                                          case 1:Set_Speed.L8 = Set_Speed.L8 + 10;//速度加 10
                                                 Set_Speed.L8    =    (Set_Speed.L8    <    50)    ?    50    :
Set_Speed.L8;//小于 50 时从 50 开始加
                                                break;
                                          case 2:Set_Temp = Set_Temp + 10;//温度加 1℃
                                                break;
                                          case 3:Set_Time.L8 = Set_Time.L8 + 60;//时间加一分钟
                                                break;
                                    }break;
                  }
            }
            if(flag == 1)
            {
                  switch(Work_Option)//工位
                  {
                        case 1:switch(SetMode_Option)//模式
                                    {
                                          case 1:Set_Speed.L1 = Set_Speed.L1 - 10;//速度减 10
                                                break;
                                          case 2:Set_Temp = Set_Temp - 10;//温度减 1℃
                                                break;
                                          case 3:Set_Time.L1 = Set_Time.L1 - 60;//时间减一分钟
                                                 break;
                                    }break;
                        case 2:switch(SetMode_Option)//模式
                                    {
                                          case 1:Set_Speed.L2 = Set_Speed.L2 - 10;//速度减 10
                                                break;
                                          case 2:Set_Temp = Set_Temp - 10;//温度减 1℃
                                                break;
                                          case 3:Set_Time.L2 = Set_Time.L2 - 60;//时间减一分钟
                                                 break;
                                    }break;
                        case 3:switch(SetMode_Option)//模式
                                    {
                                          case 1:Set_Speed.L3 = Set_Speed.L3 - 10;//速度减 10
                                                break;
                                          case 2:Set_Temp = Set_Temp - 10;//温度减 1℃
                                                break;
                                          case 3:Set_Time.L3 = Set_Time.L3 - 60;//时间减一分钟
                                                 break;
                                    }break;
                        case 4:switch(SetMode_Option)//模式
                                    {
                                          case 1:Set_Speed.L4 = Set_Speed.L4 - 10;//速度减 10
                                                break;
                        case 2:Set_Temp = Set_Temp - 10;//温度减 1℃
                               break;
                        case 3:Set_Time.L4 = Set_Time.L4 - 60;//时间减一分钟
                                break;
                   }break;
             case 5:switch(SetMode_Option)//模式
                   {
                        case 1:Set_Speed.L5 = Set_Speed.L5 - 10;//速度减 10
                               break;
                        case 2:Set_Temp = Set_Temp - 10;//温度减 1℃
                               break;
                        case 3:Set_Time.L5 = Set_Time.L5 - 60;//时间减一分钟
                                break;
                   }break;
             case 6:switch(SetMode_Option)//模式
                   {
                        case 1:Set_Speed.L6 = Set_Speed.L6 - 10;//速度减 10
                               break;
                        case 2:Set_Temp = Set_Temp - 10;//温度减 1℃
                               break;
                        case 3:Set_Time.L6 = Set_Time.L6 - 60;//时间减一分钟
                                break;
                   }break;
             case 7:switch(SetMode_Option)//模式
                   {
                        case 1:Set_Speed.L7 = Set_Speed.L7 - 10;//速度减 10
                               break;
                        case 2:Set_Temp = Set_Temp - 10;//温度减 1℃
                               break;
                        case 3:Set_Time.L7 = Set_Time.L7 - 60;//时间减一分钟
                                break;
                   }break;
             case 8:switch(SetMode_Option)//模式
                   {
                        case 1:Set_Speed.L8 = Set_Speed.L8 - 10;//速度减 10
                               break;
                        case 2:Set_Temp = Set_Temp - 10;//温度减 1℃
                               break;
                        case 3:Set_Time.L8 = Set_Time.L8 - 60;//时间减一分钟
                                break;
                   }break;
         }
     }
     Set_Speed.L1 = (Set_Speed.L1 > 1500) ? 1500 : Set_Speed.L1;//速度不超过 1500 转
     Set_Speed.L1 = (Set_Speed.L1 < 50) ? 0 : Set_Speed.L1;//速度设置小于 50 转时清零
     Set_Time.L1 = (Set_Time.L1 > 86400) ? 86400 : Set_Time.L1;//时间最多设定 23 小时 59 分钟
     Set_Time.L1 = (Set_Time.L1 < 60) ? 0 : Set_Time.L1;//时间小于 1 分钟不设定
     SetTime_State.L1    = (Set_Time.L1 < 60) ? 0 : 1;//判断是否设置了时间
```

```
    Set_Speed.L2 = (Set_Speed.L2 > 1500) ? 1500 : Set_Speed.L2;//速度不超过 1500 转
    Set_Speed.L2 = (Set_Speed.L2 < 50) ? 0 : Set_Speed.L2;//速度设置小于 50 转时清零
    Set_Time.L2 = (Set_Time.L2 > 86400) ? 86400：Set_Time.L2;//时间最多设定 23 小时 59 分钟
    Set_Time.L2 = (Set_Time.L2 < 60) ? 0 : Set_Time.L2;//时间小于 1 分钟不设定
    SetTime_State.L2    = (Set_Time.L2 < 60) ? 0 : 1;//判断是否设置了时间

    Set_Speed.L3 = (Set_Speed.L3 > 1500) ? 1500 : Set_Speed.L3;//速度不超过 1500 转
    Set_Speed.L3 = (Set_Speed.L3 < 50) ? 0 : Set_Speed.L3;//速度设置小于 50 转时清零
    Set_Time.L3 = (Set_Time.L3 > 86400) ? 86400：Set_Time.L3;//时间最多设定 23 小时 59 分钟
    Set_Time.L3 = (Set_Time.L3 < 60) ? 0 : Set_Time.L3;//时间小于 1 分钟不设定
    SetTime_State.L3    = (Set_Time.L3 < 60) ? 0 : 1;//判断是否设置了时间

    Set_Speed.L4 = (Set_Speed.L4 > 1500) ? 1500 : Set_Speed.L4;//速度不超过 1500 转
    Set_Speed.L4 = (Set_Speed.L4 < 50) ? 0 : Set_Speed.L4;//速度设置小于 50 转时清零
    Set_Time.L4 = (Set_Time.L4 > 86400) ? 86400：Set_Time.L4;//时间最多设定 23 小时 59 分钟
    Set_Time.L4 = (Set_Time.L4 < 60) ? 0 : Set_Time.L4;//时间小于 1 分钟不设定
    SetTime_State.L4    = (Set_Time.L4 < 60) ? 0 : 1;//判断是否设置了时间

    Set_Speed.L5 = (Set_Speed.L5 > 1500) ? 1500 : Set_Speed.L5;//速度不超过 1500 转
    Set_Speed.L5 = (Set_Speed.L5 < 50) ? 0 : Set_Speed.L5;//速度设置小于 50 转时清零
    Set_Time.L5 = (Set_Time.L5 > 86400) ? 86400：Set_Time.L5;//时间最多设定 23 小时 59 分钟
    Set_Time.L5 = (Set_Time.L5 < 60) ? 0 : Set_Time.L5;//时间小于 1 分钟不设定
    SetTime_State.L5    = (Set_Time.L5 < 60) ? 0 : 1;//判断是否设置了时间

    Set_Speed.L6 = (Set_Speed.L6 > 1500) ? 1500 : Set_Speed.L6;//速度不超过 1500 转
    Set_Speed.L6 = (Set_Speed.L6 < 50) ? 0 : Set_Speed.L6;//速度设置小于 50 转时清零
    Set_Time.L6 = (Set_Time.L6 > 86400) ? 86400：Set_Time.L6;//时间最多设定 23 小时 59 分钟
    Set_Time.L6 = (Set_Time.L6 < 60) ? 0 : Set_Time.L6;//时间小于 1 分钟不设定
    SetTime_State.L6    = (Set_Time.L6 < 60) ? 0 : 1;//判断是否设置了时间

    Set_Speed.L7 = (Set_Speed.L7 > 1500) ? 1500 : Set_Speed.L7;//速度不超过 1500 转
    Set_Speed.L7 = (Set_Speed.L7 < 50) ? 0 : Set_Speed.L7;//速度设置小于 50 转时清零
    Set_Time.L7 = (Set_Time.L7 > 86400) ? 86400：Set_Time.L7;//时间最多设定 23 小时 59 分钟
    Set_Time.L7 = (Set_Time.L7 < 60) ? 0 : Set_Time.L7;//时间小于 1 分钟不设定
    SetTime_State.L7    = (Set_Time.L7 < 60) ? 0 : 1;//判断是否设置了时间

    Set_Speed.L8 = (Set_Speed.L8 > 1500) ? 1500 : Set_Speed.L8;//速度不超过 1500 转
    Set_Speed.L8 = (Set_Speed.L8 < 50) ? 0 : Set_Speed.L8;//速度设置小于 50 转时清零
    Set_Time.L8 = (Set_Time.L8 > 86400) ? 86400：Set_Time.L8;//时间最多设定 23 小时 59 分钟
    Set_Time.L8 = (Set_Time.L8 < 60) ? 0 : Set_Time.L8;//时间小于 1 分钟不设定
    SetTime_State.L8    = (Set_Time.L8 < 60) ? 0 : 1;//判断是否设置了时间

    Set_Temp = (Set_Temp > 1200) ? 1200 : Set_Temp;//温度不超过 120℃
    Set_Temp = (Set_Temp < 10) ? 0 : Set_Temp;//温度设置小于 1℃时清零
    Temp_State = (Set_Temp < 10) ? 0 : 1;//判断是否设置了温度

    if(SetMode_Option!=0)//如果在设置模式中转动旋钮
    {
        Twinkle_Time = 6000;//闪烁显示 6S
        EC11A_Knob = 1;//检测是不是在旋动旋钮
        Work_All = 0;//退出同步模式
    }
}

/*
******************************************************************
 * 函数原型：  void SetALL_int(int Val,_Work_Num_ *Work_Num)
 * 功      能：  将结构图中的参数赋值-int 型
 * 输      入:  Val  赋予的值    Work_Num：结构体，要用&号连接
 * 参      数：  int Val,_Work_Num_ *Work_Num
******************************************************************
*/
void SetALL_int(int Val,_Work_Num_ *Work_Num)
{
    Work_Num->L1 = Val;
    Work_Num->L2 = Val;
    Work_Num->L3 = Val;
    Work_Num->L4 = Val;
    Work_Num->L5 = Val;
    Work_Num->L6 = Val;
    Work_Num->L7 = Val;
    Work_Num->L8 = Val;
}

/*
******************************************************************
 * 函数原型：  void SetALL_int8(uint8_t Val,_Work_Num_Flag *Work_Num)
 * 功      能：  将结构图中的参数赋值-uint8_t 型
 * 输      入:  Val  赋予的值    Work_Num：结构体，要用&号连接
 * 参      数：  uint8_t Val,_Work_Num_Flag *Work_Num
******************************************************************
*/
void SetALL_int8(uint8_t Val,_Work_Num_Flag *Work_Num)
{
    Work_Num->L1 = Val;
    Work_Num->L2 = Val;
    Work_Num->L3 = Val;
    Work_Num->L4 = Val;
    Work_Num->L5 = Val;
    Work_Num->L6 = Val;
    Work_Num->L7 = Val;
    Work_Num->L8 = Val;
}
```

```
/*
*************************************************************************
 * 函数原型：  void SetALL_int32(uint32_t Val,_Work_Num_long *Work_Num)
 * 功    能：  将结构图中的参数赋值-uint32_t 型
 * 输    入:  Val 赋予的值   Work_Num：结构体，要用&号连接
 * 参    数：  uint32_t Val,_Work_Num_long *Work_Num
*************************************************************************
*/
void SetALL_int32(uint32_t Val,_Work_Num_long *Work_Num)
{
    Work_Num->L1 = Val;
    Work_Num->L2 = Val;
    Work_Num->L3 = Val;
    Work_Num->L4 = Val;
    Work_Num->L5 = Val;
    Work_Num->L6 = Val;
    Work_Num->L7 = Val;
    Work_Num->L8 = Val;
}

/*
*************************************************************************
 * 函数原型：  void SetALL_TimeOver(_Work_Num_long *Work_Num1,_Work_Num_long
*Work_Num)
 * 功    能：  //将两个结构体变量的参数对应赋值，用于结束时间复原
 * 输    入:  Work_Num1 结构体，要用&号连接   Work_Num：结构体，要用&号连接
 * 参    数：  _Work_Num_long *Work_Num1,_Work_Num_long *Work_Num
*************************************************************************
*/
void SetALL_TimeOver(_Work_Num_long *Work_Num1,_Work_Num_long *Work_Num)
{
    Work_Num1->L1 = Work_Num->L1;
    Work_Num1->L2 = Work_Num->L2;
    Work_Num1->L3 = Work_Num->L3;
    Work_Num1->L4 = Work_Num->L4;
    Work_Num1->L5 = Work_Num->L5;
    Work_Num1->L6 = Work_Num->L6;
    Work_Num1->L7 = Work_Num->L7;
    Work_Num1->L8 = Work_Num->L8;
}

/*
*************************************************************************
 * 函数原型：  void SetALL_SpeedOver(_Work_Num_ *Work_Num1,_Work_Num_
*Work_Num)
 * 功    能：  将两个结构体变量的参数对应赋值，用于结束时间速度复原
 * 输    入:  Work_Num1 结构体，要用&号连接   Work_Num：结构体，要用&号连接
 * 参    数：  _Work_Num_long *Work_Num1,_Work_Num_long *Work_Num
*************************************************************************
```

```
*/
void SetALL_SpeedOver(_Work_Num_ *Work_Num1,_Work_Num_ *Work_Num)
{
    Work_Num1->L1 = Work_Num->L1;
    Work_Num1->L2 = Work_Num->L2;
    Work_Num1->L3 = Work_Num->L3;
    Work_Num1->L4 = Work_Num->L4;
    Work_Num1->L5 = Work_Num->L5;
    Work_Num1->L6 = Work_Num->L6;
    Work_Num1->L7 = Work_Num->L7;
    Work_Num1->L8 = Work_Num->L8;
}
/*
****************************************************************
 * 函数原型：  void Speed_ALL(uint8_t work,_Work_Num_ *Work_Num,_Work_Num_ *Work_Num1)
 * 功    能：  同步功能，将所有工位的速度同步
 * 输    入:  work 工位号 Work_Num 结构体，要用&号连接  Work_Num1：结构体，要用&号连接
 * 参    数：  uint8_t work,_Work_Num_ *Work_Num,_Work_Num_ *Work_Num1
****************************************************************
*/
void Speed_ALL(uint8_t work,_Work_Num_ *Work_Num,_Work_Num_ *Work_Num1)
{
    switch(work)
    {
        case 1: Work_Num->L1 = Work_Num1->L1;
                Work_Num->L2 = Work_Num1->L1;
                Work_Num->L3 = Work_Num1->L1;
                Work_Num->L4 = Work_Num1->L1;
                Work_Num->L5 = Work_Num1->L1;
                Work_Num->L6 = Work_Num1->L1;
                Work_Num->L7 = Work_Num1->L1;
                Work_Num->L8 = Work_Num1->L1;
                break;
        case 2:Work_Num->L2 = Work_Num1->L2;
                Work_Num->L1 = Work_Num1->L2;
                Work_Num->L3 = Work_Num1->L2;
                Work_Num->L4 = Work_Num1->L2;
                Work_Num->L5 = Work_Num1->L2;
                Work_Num->L6 = Work_Num1->L2;
                Work_Num->L7 = Work_Num1->L2;
                Work_Num->L8 = Work_Num1->L2;
                break;
        case 3:Work_Num->L3 = Work_Num1->L3;
                Work_Num->L1 = Work_Num1->L3;
                Work_Num->L2 = Work_Num1->L3;
                Work_Num->L4 = Work_Num1->L3;
                Work_Num->L5 = Work_Num1->L3;
                Work_Num->L6 = Work_Num1->L3;
```

```
                    Work_Num->L7 = Work_Num1->L3;
                    Work_Num->L8 = Work_Num1->L3;
                    break;
        case 4:Work_Num->L4 = Work_Num1->L4;
                    Work_Num->L1 = Work_Num1->L4;
                    Work_Num->L2 = Work_Num1->L4;
                    Work_Num->L3 = Work_Num1->L4;
                    Work_Num->L5 = Work_Num1->L4;
                    Work_Num->L6 = Work_Num1->L4;
                    Work_Num->L7 = Work_Num1->L4;
                    Work_Num->L8 = Work_Num1->L4;
                    break;
        case 5:
                    Work_Num->L5 = Work_Num1->L5;
                    Work_Num->L1 = Work_Num1->L5;
                    Work_Num->L2 = Work_Num1->L5;
                    Work_Num->L3 = Work_Num1->L5;
                    Work_Num->L4 = Work_Num1->L5;
                    Work_Num->L6 = Work_Num1->L5;
                    Work_Num->L7 = Work_Num1->L5;
                    Work_Num->L8 = Work_Num1->L5;
                    break;
        case 6:Work_Num->L6 = Work_Num1->L6;
                    Work_Num->L1 = Work_Num1->L6;
                    Work_Num->L2 = Work_Num1->L6;
                    Work_Num->L3 = Work_Num1->L6;
                    Work_Num->L4 = Work_Num1->L6;
                    Work_Num->L5 = Work_Num1->L6;
                    Work_Num->L7 = Work_Num1->L6;
                    Work_Num->L8 = Work_Num1->L6;
                    break;
        case 7:Work_Num->L7 = Work_Num1->L7;
                    Work_Num->L1 = Work_Num1->L7;
                    Work_Num->L2 = Work_Num1->L7;
                    Work_Num->L3 = Work_Num1->L7;
                    Work_Num->L4 = Work_Num1->L7;
                    Work_Num->L5 = Work_Num1->L7;
                    Work_Num->L6 = Work_Num1->L7;
                    Work_Num->L8 = Work_Num1->L7;
                    break;
        case 8:Work_Num->L8 = Work_Num1->L8;
                    Work_Num->L1 = Work_Num1->L8;
                    Work_Num->L2 = Work_Num1->L8;
                    Work_Num->L3 = Work_Num1->L8;
                    Work_Num->L4 = Work_Num1->L8;
                    Work_Num->L5 = Work_Num1->L8;
                    Work_Num->L6 = Work_Num1->L8;
                    Work_Num->L7 = Work_Num1->L8;
                    break;
    }
```

```
}

/*
******************************************************************
 * 函数原型： void Time_ALL(uint8_t work,_Work_Num_long *Work_Num,_Work_Num_long
*Work_Num1)
 * 功      能： 同步功能，将所有工位的时间同步
 * 输      入: work 工位号 Work_Num 结构体，要用&号连接  Work_Num1：结构体，要
用&号连接
 * 参      数： uint8_t work,_Work_Num_long *Work_Num,_Work_Num_long *Work_Num1
******************************************************************
*/
void Time_ALL(uint8_t work,_Work_Num_long *Work_Num,_Work_Num_long *Work_Num1)
{
	switch(work)
	{
		case 1: Work_Num->L1 = Work_Num1->L1;
				Work_Num->L2 = Work_Num1->L1;
				Work_Num->L3 = Work_Num1->L1;
				Work_Num->L4 = Work_Num1->L1;
				Work_Num->L5 = Work_Num1->L1;
				Work_Num->L6 = Work_Num1->L1;
				Work_Num->L7 = Work_Num1->L1;
				Work_Num->L8 = Work_Num1->L1;
				break;
		case 2:Work_Num->L2 = Work_Num1->L2;
				Work_Num->L1 = Work_Num1->L2;
				Work_Num->L3 = Work_Num1->L2;
				Work_Num->L4 = Work_Num1->L2;
				Work_Num->L5 = Work_Num1->L2;
				Work_Num->L6 = Work_Num1->L2;
				Work_Num->L7 = Work_Num1->L2;
				Work_Num->L8 = Work_Num1->L2;
				break;
		case 3:Work_Num->L3 = Work_Num1->L3;
				Work_Num->L1 = Work_Num1->L3;
				Work_Num->L2 = Work_Num1->L3;
				Work_Num->L4 = Work_Num1->L3;
				Work_Num->L5 = Work_Num1->L3;
				Work_Num->L6 = Work_Num1->L3;
				Work_Num->L7 = Work_Num1->L3;
				Work_Num->L8 = Work_Num1->L3;
				break;
		case 4:Work_Num->L4 = Work_Num1->L4;
				Work_Num->L1 = Work_Num1->L4;
				Work_Num->L2 = Work_Num1->L4;
				Work_Num->L3 = Work_Num1->L4;
				Work_Num->L5 = Work_Num1->L4;
				Work_Num->L6 = Work_Num1->L4;
				Work_Num->L7 = Work_Num1->L4;
```

```
                Work_Num->L8 = Work_Num1->L4;
                break;
        case 5:
                Work_Num->L5 = Work_Num1->L5;
                Work_Num->L1 = Work_Num1->L5;
                Work_Num->L2 = Work_Num1->L5;
                Work_Num->L3 = Work_Num1->L5;
                Work_Num->L4 = Work_Num1->L5;
                Work_Num->L6 = Work_Num1->L5;
                Work_Num->L7 = Work_Num1->L5;
                Work_Num->L8 = Work_Num1->L5;
                break;
        case 6:Work_Num->L6 = Work_Num1->L6;
                Work_Num->L1 = Work_Num1->L6;
                Work_Num->L2 = Work_Num1->L6;
                Work_Num->L3 = Work_Num1->L6;
                Work_Num->L4 = Work_Num1->L6;
                Work_Num->L5 = Work_Num1->L6;
                Work_Num->L7 = Work_Num1->L6;
                Work_Num->L8 = Work_Num1->L6;
                break;
        case 7:Work_Num->L7 = Work_Num1->L7;
                Work_Num->L1 = Work_Num1->L7;
                Work_Num->L2 = Work_Num1->L7;
                Work_Num->L3 = Work_Num1->L7;
                Work_Num->L4 = Work_Num1->L7;
                Work_Num->L5 = Work_Num1->L7;
                Work_Num->L6 = Work_Num1->L7;
                Work_Num->L8 = Work_Num1->L7;
                break;
        case 8:Work_Num->L8 = Work_Num1->L8;
                Work_Num->L1 = Work_Num1->L8;
                Work_Num->L2 = Work_Num1->L8;
                Work_Num->L3 = Work_Num1->L8;
                Work_Num->L4 = Work_Num1->L8;
                Work_Num->L5 = Work_Num1->L8;
                Work_Num->L6 = Work_Num1->L8;
                Work_Num->L7 = Work_Num1->L8;
                break;
    }
}

/*
******************************************************************
 * 函数原型：  void Flag_ALL(uint8_t work,_Work_Num_Flag *Work_Num,_Work_Num_Flag
*Work_Num1)
 * 功    能：  同步功能，将所有工位的 flag 同步
 * 输    入:  work 工位号 Work_Num 结构体，要用&号连接  Work_Num1：结构体，要
用&号连接
 * 参    数：  uint8_t work,_Work_Num_Flag *Work_Num,_Work_Num_Flag *Work_Num1
```

```
**************************************************************************
*/
void Flag_ALL(uint8_t work,_Work_Num_Flag *Work_Num,_Work_Num_Flag *Work_Num1)
{
        switch(work)
        {
                case 1: Work_Num->L1 = Work_Num1->L1;
                                Work_Num->L2 = Work_Num1->L1;
                                Work_Num->L3 = Work_Num1->L1;
                                Work_Num->L4 = Work_Num1->L1;
                                Work_Num->L5 = Work_Num1->L1;
                                Work_Num->L6 = Work_Num1->L1;
                                Work_Num->L7 = Work_Num1->L1;
                                Work_Num->L8 = Work_Num1->L1;
                                break;
                case 2:Work_Num->L2 = Work_Num1->L2;
                                Work_Num->L1 = Work_Num1->L2;
                                Work_Num->L3 = Work_Num1->L2;
                                Work_Num->L4 = Work_Num1->L2;
                                Work_Num->L5 = Work_Num1->L2;
                                Work_Num->L6 = Work_Num1->L2;
                                Work_Num->L7 = Work_Num1->L2;
                                Work_Num->L8 = Work_Num1->L2;
                                break;
                case 3:Work_Num->L3 = Work_Num1->L3;
                                Work_Num->L1 = Work_Num1->L3;
                                Work_Num->L2 = Work_Num1->L3;
                                Work_Num->L4 = Work_Num1->L3;
                                Work_Num->L5 = Work_Num1->L3;
                                Work_Num->L6 = Work_Num1->L3;
                                Work_Num->L7 = Work_Num1->L3;
                                Work_Num->L8 = Work_Num1->L3;
                                break;
                case 4:Work_Num->L4 = Work_Num1->L4;
                                Work_Num->L1 = Work_Num1->L4;
                                Work_Num->L2 = Work_Num1->L4;
                                Work_Num->L3 = Work_Num1->L4;
                                Work_Num->L5 = Work_Num1->L4;
                                Work_Num->L6 = Work_Num1->L4;
                                Work_Num->L7 = Work_Num1->L4;
                                Work_Num->L8 = Work_Num1->L4;
                                break;
                case 5:
                                Work_Num->L5 = Work_Num1->L5;
                                Work_Num->L1 = Work_Num1->L5;
                                Work_Num->L2 = Work_Num1->L5;
                                Work_Num->L3 = Work_Num1->L5;
                                Work_Num->L4 = Work_Num1->L5;
                                Work_Num->L6 = Work_Num1->L5;
                                Work_Num->L7 = Work_Num1->L5;
                Work_Num->L8 = Work_Num1->L5;
                break;
        case 6:Work_Num->L6 = Work_Num1->L6;
                Work_Num->L1 = Work_Num1->L6;
                Work_Num->L2 = Work_Num1->L6;
                Work_Num->L3 = Work_Num1->L6;
                Work_Num->L4 = Work_Num1->L6;
                Work_Num->L5 = Work_Num1->L6;
                Work_Num->L7 = Work_Num1->L6;
                Work_Num->L8 = Work_Num1->L6;
                break;
        case 7:Work_Num->L7 = Work_Num1->L7;
                Work_Num->L1 = Work_Num1->L7;
                Work_Num->L2 = Work_Num1->L7;
                Work_Num->L3 = Work_Num1->L7;
                Work_Num->L4 = Work_Num1->L7;
                Work_Num->L5 = Work_Num1->L7;
                Work_Num->L6 = Work_Num1->L7;
                Work_Num->L8 = Work_Num1->L7;
                break;
        case 8:Work_Num->L8 = Work_Num1->L8;
                Work_Num->L1 = Work_Num1->L8;
                Work_Num->L2 = Work_Num1->L8;
                Work_Num->L3 = Work_Num1->L8;
                Work_Num->L4 = Work_Num1->L8;
                Work_Num->L5 = Work_Num1->L8;
                Work_Num->L6 = Work_Num1->L8;
                Work_Num->L7 = Work_Num1->L8;
                break;
    }
}

/*
*****************************************************************
 * 函数原型： void Set_Speeds(_Work_Num_ *Work_Num,_Work_Num_
*Work_Num1,_Work_Num_ *Work_Num2)
 * 功    能： 判断设置速度数值是否改变
 * 输    入: Work_Num 临时存储的速度，要用&号连接 Work_Num1：设置的速度，要
用&号连接 Work_Num2：要赋值的速度，要用&号连接
 * 参    数： _Work_Num_ *Work_Num,_Work_Num_ *Work_Num1,_Work_Num_
*Work_Num2
*****************************************************************
*/
void Set_Speeds(_Work_Num_ *Work_Num,_Work_Num_ *Work_Num1,_Work_Num_
*Work_Num2)
{
    if(Work_Num->L1 != Work_Num1->L1)
    {
        Work_Num2->L1 = Work_Num1->L1;
        if(Speed_ADDMode.L1  !=  0)//假如工位只有在启动并且设置了速度的情况下不等于
```

```
0，不在未处理模式下
            Speed_ADDMode.L1 = 0;//进入未处理，判断加速还是减速
    }
    if(Work_Num->L2 != Work_Num1->L2)
    {
        Work_Num2->L2 = Work_Num1->L2;
        if(Speed_ADDMode.L2  !=  0)//假如工位只有在启动并且设置了速度的情况下不等于
0，不在未处理模式下
                Speed_ADDMode.L2 = 0;//进入未处理，判断加速还是减速
    }
    if(Work_Num->L3 != Work_Num1->L3)
    {
         Work_Num2->L3 = Work_Num1->L3;
         if(Speed_ADDMode.L3 != 0)//假如工位只有在启动并且设置了速度的情况下不等于
0，不在未处理模式下
            Speed_ADDMode.L3 = 0;//进入未处理，判断加速还是减速
    }
    if(Work_Num->L4 != Work_Num1->L4)
    {
        Work_Num2->L4 = Work_Num1->L4;
        if(Speed_ADDMode.L4  !=  0)//假如工位只有在启动并且设置了速度的情况下不等于
0，不在未处理模式下
                Speed_ADDMode.L4 = 0;//进入未处理，判断加速还是减速
    }
    if(Work_Num->L5 != Work_Num1->L5)
    {
        Work_Num2->L5 = Work_Num1->L5;
        if(Speed_ADDMode.L5  !=  0)//假如工位只有在启动并且设置了速度的情况下不等于
0，不在未处理模式下
                Speed_ADDMode.L5 = 0;//进入未处理，判断加速还是减速
    }
    if(Work_Num->L6 != Work_Num1->L6)
    {
         Work_Num2->L6 = Work_Num1->L6;
        if(Speed_ADDMode.L6  !=  0)//假如工位只有在启动并且设置了速度的情况下不等于
0，不在未处理模式下
                Speed_ADDMode.L6 = 0;//进入未处理，判断加速还是减速
    }
    if(Work_Num->L7 != Work_Num1->L7)
    {
         Work_Num2->L7 = Work_Num1->L7;
         if(Speed_ADDMode.L7 != 0)//假如工位只有在启动并且设置了速度的情况下不等于
0，不在未处理模式下
                Speed_ADDMode.L7 = 0;//进入未处理，判断加速还是减速
    }
    if(Work_Num->L8 != Work_Num1->L8)
    {
        Work_Num2->L8 = Work_Num1->L8;
        if(Speed_ADDMode.L8  !=  0)//假如工位只有在启动并且设置了速度的情况下不等于
0，不在未处理模式下
```

```
                    Speed_ADDMode.L8 = 0;//进入未处理，判断加速还是减速
    }
}

/*
*****************************************************************
 * 函数原型： void Set_Times(_Work_Num_long *Work_Num,_Work_Num_long *Work_Num1,_Work_Num_long *Work_Num2)
 * 功    能： 判断设置时间数值是否改变
 * 输    入: Work_Num 临时存储的时间，要用&号连接 Work_Num1：设置的时间，要用&号连接 Work_Num2：要赋值的时间，要用&号连接
 * 参    数 ： _Work_Num_long *Work_Num,_Work_Num_long *Work_Num1,_Work_Num_long *Work_Num2
*****************************************************************
*/
void Set_Times(_Work_Num_long *Work_Num,_Work_Num_long *Work_Num1,_Work_Num_long *Work_Num2)
{
    if(Work_Num->L1 != Work_Num1->L1)
    {
        Work_Num2->L1 = Work_Num1->L1;
        RelTime_State.L1 = SetTime_State.L1;//同步时间状态
        DownTime_Over.L1 = 0;
    }
    if(Work_Num->L2 != Work_Num1->L2)
    {
        Work_Num2->L2 = Work_Num1->L2;
        RelTime_State.L2 = SetTime_State.L2;//同步时间状态
        DownTime_Over.L2 = 0;
    }
    if(Work_Num->L3 != Work_Num1->L3)
    {
        Work_Num2->L3 = Work_Num1->L3;
         RelTime_State.L3 = SetTime_State.L3;//同步时间状态
        DownTime_Over.L3 = 0;
    }
    if(Work_Num->L4 != Work_Num1->L4)
    {
        Work_Num2->L4 = Work_Num1->L4;
        RelTime_State.L4 = SetTime_State.L4;//同步时间状态
        DownTime_Over.L4 = 0;
    }
    if(Work_Num->L5 != Work_Num1->L5)
    {
        Work_Num2->L5 = Work_Num1->L5;
        RelTime_State.L5 = SetTime_State.L5;//同步时间状态
        DownTime_Over.L5 = 0;
    }
    if(Work_Num->L6 != Work_Num1->L6)
    {
            Work_Num2->L6 = Work_Num1->L6;
            RelTime_State.L6 = SetTime_State.L6;//同步时间状态
            DownTime_Over.L6 = 0;
    }
    if(Work_Num->L7 != Work_Num1->L7)
    {
            Work_Num2->L7 = Work_Num1->L7;
            RelTime_State.L7 = SetTime_State.L7;//同步时间状态
            DownTime_Over.L7 = 0;
    }
    if(Work_Num->L8 != Work_Num1->L8)
    {
            Work_Num2->L8 = Work_Num1->L8;
            RelTime_State.L8 = SetTime_State.L8;//同步时间状态
            DownTime_Over.L8 = 0;
    }
}

#include "Show.h"

/**********全局变量声明******/
uint16_t Twinkle_Time;//闪烁时间



/**********局部变量声明******/
uint8_t seg15_Flag;//用于显示屏实际速度最后一位中间的斜杠个其他的相差太远
uint8_t Speed_ShowFlag,Temp_ShowFlag,Time_ShowFlag;//速度、温度、时间显示的标志位 0:
常亮 1：熄灭
uint8_t Work_All_Time;//全部工位设置时闪烁的工位时间
uint8_t Run_Flag;//运行闪烁图标

/*
******************************************************************
 * 函数原型：   void Check_ShowFlag(uint16_t dT)
 * 功      能：   闪烁检测
 * 输      入:   dT:执行周期
 * 参      数：   uint16_t dT
******************************************************************
*/
void Check_ShowFlag(uint16_t dT)
{
        if(Work_All)//进入全家设置
                Work_All_Time = ~Work_All_Time;//选中工位下闪烁
        if(Run_Status)//运行时
                Run_Flag = ~Run_Flag;//运行时闪烁
        if(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 && EC11A_Knob==0)//闪烁和没有操作旋钮时
        {
                Twinkle_Time -= dT;//闪烁计时
                if(SetMode_Option == 1)//设置速度
                {
                    Speed_ShowFlag = ~Speed_ShowFlag;//速度闪烁
                    Temp_ShowFlag = 0;//温度常亮
                    Time_ShowFlag = 0;//时间常亮
                }
                else if(SetMode_Option == 2)//设置温度
                {
                    Speed_ShowFlag = 0;//速度常亮
                    Temp_ShowFlag = ~Temp_ShowFlag;//温度闪烁
                    Time_ShowFlag = 0;//时间常亮
                }
                else if(SetMode_Option == 3)//设置时间
                {
                    Speed_ShowFlag = 0;//速度常亮
                    Temp_ShowFlag = 0;//温度常亮
                    Time_ShowFlag = ~Time_ShowFlag;//时间闪烁
                }
                if(Twinkle_Time == 0)//如果闪烁结束
                {
                    SetMode_Option = 0;//模式选择清零
                }
        }
}

/*
*****************************************************************
 * 函数原型：  void LCD_Light(short LCD_Status)
 * 功     能：  打开和关闭背光显示
 * 输     入:  LCD_Status：1、打开背光  0：关闭背光
 * 参     数：  short LCD_Status
*****************************************************************
*/
void LCD_Light(short LCD_Status)
{
        switch(LCD_Status)
        {
                case 0 :
                    HAL_GPIO_WritePin(LED_KEY_GPIO_Port,                LED_KEY_Pin,
GPIO_PIN_SET);//拉高背光电平信号
                    #if(Integration_TYPE == 0)//设置成四联时
                    __HAL_TIM_SET_COMPARE(&htim4, TIM_CHANNEL_3, 0);//不输出 pwm
                    #elif(Integration_TYPE == 1)//设置成六联时
                    __HAL_TIM_SET_COMPARE(&htim4, TIM_CHANNEL_3, 0);//不输出 pwm
                    #elif(Integration_TYPE == 2)//设置成八联时
                    __HAL_TIM_SET_COMPARE(&htim8, TIM_CHANNEL_1, 0);//不输出 pwm
                    #endif
                    break ;
                case 1 :
                    HAL_GPIO_WritePin(LED_KEY_GPIO_Port, LED_KEY_Pin, GPIO_PIN_RESET);//拉低背光电平信号
                    #if(Integration_TYPE == 0)//设置成四联时
                    __HAL_TIM_SET_COMPARE(&htim4, TIM_CHANNEL_3, 15);//不输出 pwm
                    #elif(Integration_TYPE == 1)//设置成六联时
                    __HAL_TIM_SET_COMPARE(&htim4, TIM_CHANNEL_3, 7);//不输出 pwm
                    #elif(Integration_TYPE == 2)//设置成八联时
                    __HAL_TIM_SET_COMPARE(&htim8, TIM_CHANNEL_1, 15);//不输出 pwm
                    #endif
                    break ;
        }
}

/*
****************************************************************
 *  函数原型：  void Display_Speed_Lift(uint16_t Speed_L1,uint16_t Speed_L7)
 *  功      能：  写最左边的设定速度
 *  输      入:   Speed_L1 左边最上面的速度  Speed_L7 左边最上面的速度
 *  参      数：  uint16_t Speed_L1,uint16_t Speed_L7
****************************************************************
*/
void Display_Speed_Lift(uint16_t Speed_L1,uint16_t Speed_L3,uint16_t Speed_L5,uint16_t Speed_L7)
{
        uint8_t seg21,seg22,seg23,seg24,seg25,seg26,seg27,seg28,seg29,seg30;
        seg21=0;seg22=0;seg23=0;seg24=0;seg25=0;seg26=0;seg27=0;seg28=0;seg29=0;seg30=0;
        uint8_t Val;//用于百十个取出来的数字

        /***********L1 千位***********/
        if(Speed_L1 > 999)//大于 999 时
        {
                seg21&=0x3f;seg21|=0x80;//显示 1
        }
        else//小于 999 时
        {
                seg21&=0x3f;seg21|=0xc0;//显示 0
        }

        /***********L1 百位***********/
        if(Speed_L1 > 99)//大于 99 时
        {
                Val=Speed_L1/100;//取出百位的数字
                if(Speed_L1 > 999)//加入大于 999 时
                    Val=Val%10;//取出百位的数字
                switch(Val)
                {
                    case 0:seg23&=0xbf;seg22|=0xc0;seg23|=0x80;seg24|=0xc0;seg25|=0x80;//数字 0
                        break;
                    case
1:seg22&=0x3f;seg23&=0x3f;seg24&=0x3f;seg25&=0x3f;seg24|=0x80;seg25|=0x80;//数字 1
                        break;
                    case
2:seg22&=0x3f;seg23&=0x3f;seg24&=0x3f;seg25&=0x3f;seg22|=0x40;seg23|=0xc0;seg24|=0xc
0;//数字 2
                        break;
                    case
3:seg22&=0x3f;seg23&=0x3f;seg24&=0x3f;seg25&=0x3f;seg23|=0xc0;seg24|=0xc0;seg25|=0x8
0;//数字 3
                        break;
                    case
4:seg22&=0x3f;seg23&=0x3f;seg24&=0x3f;seg25&=0x3f;seg22|=0x80;seg23|=0x40;seg24|=0x8
0;seg25|=0x80;//数字 4
                        break;
                    case
5:seg22&=0x3f;seg23&=0x3f;seg24&=0x3f;seg25&=0x3f;seg22|=0x80;seg23|=0xc0;seg24|=0x4
0;seg25|=0x80;//数字 5
                        break;
                    case
6:seg22&=0x3f;seg23&=0x3f;seg24&=0x3f;seg25&=0x3f;seg22|=0xc0;seg23|=0xc0;seg24|=0x4
0;seg25|=0x80;//数字 6
                        break;
                    case
7:seg22&=0x3f;seg23&=0x3f;seg24&=0x3f;seg25&=0x3f;seg22|=0x00;seg23|=0x80;seg24|=0x8
0;seg25|=0x80;//数字 7
                        break;
                    case
8:seg22&=0x3f;seg23&=0x3f;seg24&=0x3f;seg25&=0x3f;seg22|=0xc0;seg23|=0xc0;seg24|=0xc
0;seg25|=0x80;//数字 8
                        break;
                    case
9:seg22&=0x3f;seg23&=0x3f;seg24&=0x3f;seg25&=0x3f;seg22|=0x80;seg23|=0xc0;seg24|=0xc
0;seg25|=0x80;//数字 9
                        break;
                    default:
                        break;
                }
        }
        else
        {
                seg23&=0xbf;seg22|=0xc0;seg23|=0x80;seg24|=0xc0;seg25|=0x80;//数字 0
        }

        /***********L1 十位***********/
        if(Speed_L1 > 9)//大于 9 时
        {
                Val=Speed_L1/10;//取出十位的数字
                if(Speed_L1 > 99)//大于 99 时
                    Val=Val%10;//取出十位的数字
                switch(Val)
                {
                    case 0:seg27&=0xbf;seg26|=0xc0;seg27|=0x80;seg28|=0xc0;seg29|=0x40;//数字 0
                        break;
                    case
1:seg26&=0x3f;seg27&=0x3f;seg28&=0x3f;seg29&=0x3f;seg28|=0x80;seg29|=0x40;//数字 1
                        break;
                    case
2:seg26&=0x3f;seg27&=0x3f;seg28&=0x3f;seg29&=0x3f;seg26|=0x40;seg27|=0xc0;seg28|=0xc
0;//数字 2
                        break;
                    case
3:seg26&=0x3f;seg27&=0x3f;seg28&=0x3f;seg29&=0x3f;seg27|=0xc0;seg28|=0xc0;seg29|=0x4
0;//数字 3
                        break;
                    case
4:seg26&=0x3f;seg27&=0x3f;seg28&=0x3f;seg29&=0x3f;seg26|=0x80;seg27|=0x40;seg28|=0x8
0;seg29|=0x40;//数字 4
                        break;
                    case
5:seg26&=0x3f;seg27&=0x3f;seg28&=0x3f;seg29&=0x3f;seg26|=0x80;seg27|=0xc0;seg28|=0x4
0;seg29|=0x40;//数字 5
                        break;
                    case
6:seg26&=0x3f;seg27&=0x3f;seg28&=0x3f;seg29&=0x3f;seg26|=0xc0;seg27|=0xc0;seg28|=0x4
0;seg29|=0x40;//数字 6
                        break;
                    case
7:seg26&=0x3f;seg27&=0x3f;seg28&=0x3f;seg29&=0x3f;seg26|=0x00;seg27|=0x80;seg28|=0x8
0;seg29|=0x40;//数字 7
                        break;
                    case
8:seg26&=0x3f;seg27&=0x3f;seg28&=0x3f;seg29&=0x3f;seg26|=0xc0;seg27|=0xc0;seg28|=0xc
0;seg29|=0x40;//数字 8
                        break;
                    case
9:seg26&=0x3f;seg27&=0x3f;seg28&=0x3f;seg29&=0x3f;seg26|=0x80;seg27|=0xc0;seg28|=0xc
0;seg29|=0x40;//数字 9
                        break;
                    default:
                        break;
                }
        }
        else
        {
                seg27&=0xbf;seg26|=0xc0;seg27|=0x80;seg28|=0xc0;seg29|=0x40;//数字 0
        }
        /***********L1 个位***********/
        Val=Speed_L1%10;//取出个位的数字
        if(Val > 4)//大于 4 时
        {
                seg29&=0xef;seg30&=0x3f;seg29|=0x80;seg30|=0x80;//显示 5
        }
        else//小于 5 时
        {
                seg29&=0xef;seg30&=0x3f;seg29|=0x00;seg30|=0xc0;//显示 0
        }

        /***********L3 千位***********/
        if(Speed_L3 > 999)//大于 999 时
        {
                seg21&=0xCf;seg21|=0x20;//显示 1

        }
        else//小于 999 时
        {
                seg21|=0x30;//显示 0
        }

        /***********L3 个百位***********/
        if(Speed_L3 > 99)//大于 99 时
        {
                Val=Speed_L3/100;//取出百位
                if(Speed_L3 > 999)//大于 999 时
                    Val=Val%10;//去除百位
                switch(Val)
                {
                    case 0:seg23&=0xef;seg22|=0x30;seg23|=0x20;seg24|=0x30;seg25|=0x20;//数字 0
                        break;
                    case
1:seg22&=0xCf;seg23&=0xCf;seg24&=0xCf;seg25&=0xCf;seg24|=0x20;seg25|=0x20;//数字 1
                        break;
                    case
2:seg22&=0xCf;seg23&=0xCf;seg24&=0xCf;seg25&=0xCf;seg22|=0x10;seg23|=0x30;seg24|=0x
30;//数字 2
                        break;
                    case
3:seg22&=0xCf;seg23&=0xCf;seg24&=0xCf;seg25&=0xCf;seg23|=0x30;seg24|=0x30;seg25|=0x
20;//数字 3
                        break;
                    case
4:seg22&=0xCf;seg23&=0xCf;seg24&=0xCf;seg25&=0xCf;seg22|=0x20;seg23|=0x10;seg24|=0x
20;seg25|=0x20;//数字 4
                        break;
                    case
5:seg22&=0xCf;seg23&=0xCf;seg24&=0xCf;seg25&=0xCf;seg22|=0x20;seg23|=0x30;seg24|=0x
```

```
10;seg25|=0x20;//数字 5
                break;
            case
6:seg22&=0xCf;seg23&=0xCf;seg24&=0xCf;seg25&=0xCf;seg22|=0x30;seg23|=0x30;seg24|=0x
10;seg25|=0x20;//数字 6
                break;
            case
7:seg22&=0xCf;seg23&=0xCf;seg24&=0xCf;seg25&=0xCf;seg22|=0x00;seg23|=0x20;seg24|=0x
20;seg25|=0x20;//数字 7
                break;
            case
8:seg22&=0xCf;seg23&=0xCf;seg24&=0xCf;seg25&=0xCf;seg22|=0x30;seg23|=0x30;seg24|=0x
30;seg25|=0x20;//数字 8
                break;
            case
9:seg22&=0xCf;seg23&=0xCf;seg24&=0xCf;seg25&=0xCf;seg22|=0x20;seg23|=0x30;seg24|=0x
30;seg25|=0x20;//数字 9
                break;
            default:
                break;
        }
    }
    else
    {
        seg23&=0xef;seg22|=0x30;seg23|=0x20;seg24|=0x30;seg25|=0x20;//数字 0
    }

    /***********L3 个十位***********/
    if(Speed_L3 > 9)//大于 9 时
    {
        Val=Speed_L3/10;//取出十位
        if(Speed_L3 > 99)//大于 99 时
            Val=Val%10;//取出十位
        switch(Val)
        {
            case 0:seg27&=0xef;seg26|=0x30;seg27|=0x20;seg28|=0x30;seg29|=0x10;//数字 0
                break;
            case
1:seg26&=0xCf;seg27&=0xCf;seg28&=0xCf;seg29&=0xCf;seg28|=0x20;seg29|=0x10;//数字 1
                break;
            case
2:seg26&=0xCf;seg27&=0xCf;seg28&=0xCf;seg29&=0xCf;seg26|=0x10;seg27|=0x30;seg28|=0x
30;//数字 2
                break;
            case
3:seg26&=0xCf;seg27&=0xCf;seg28&=0xCf;seg29&=0xCf;seg27|=0x30;seg28|=0x30;seg29|=0x
10;//数字 3
                break;
            case
4:seg26&=0xCf;seg27&=0xCf;seg28&=0xCf;seg29&=0xCf;seg26|=0x20;seg27|=0x10;seg28|=0x
```

```
20;seg29|=0x10;//数字 4
                break;
            case
5:seg26&=0xCf;seg27&=0xCf;seg28&=0xCf;seg29&=0xCf;seg26|=0x20;seg27|=0x30;seg28|=0x
10;seg29|=0x10;//数字 5
                break;
            case
6:seg26&=0xCf;seg27&=0xCf;seg28&=0xCf;seg29&=0xCf;seg26|=0x30;seg27|=0x30;seg28|=0x
10;seg29|=0x10;//数字 6
                break;
            case
7:seg26&=0xCf;seg27&=0xCf;seg28&=0xCf;seg29&=0xCf;seg26|=0x00;seg27|=0x20;seg28|=0x
20;seg29|=0x10;//数字 7
                break;
            case
8:seg26&=0xCf;seg27&=0xCf;seg28&=0xCf;seg29&=0xCf;seg26|=0x30;seg27|=0x30;seg28|=0x
30;seg29|=0x10;//数字 8
                break;
            case
9:seg26&=0xCf;seg27&=0xCf;seg28&=0xCf;seg29&=0xCf;seg26|=0x20;seg27|=0x30;seg28|=0x
30;seg29|=0x10;//数字 9
                break;
            default:
                break;
        }
    }
    else
    {
        seg27&=0xef;seg26|=0x30;seg27|=0x20;seg28|=0x30;seg29|=0x10;//数字 0
    }

    /***********L3 个位***********/
    Val=Speed_L3%10;//去除十位
    if(Val > 4)//大于 4 时
    {
        seg29&=0xdf;seg30&=0xcf;seg29|=0x20;seg30|=0x20;//显示 5
    }
    else
    {
        seg29&=0xdf;seg30&=0xcf;seg29|=0x00;seg30|=0x30;//显示 0
    }

   /***********L5 千位***********/
    if(Speed_L5 > 999)//大于 999 时
    {
        seg21|=0x08;seg21&=0xfb;//显示 1
    }
    else//小于 999 时
    {
        seg21|=0x0c;//显示 0
    }

    /***********L5 百位***********/
    if(Speed_L5 > 99)//大于 99 时
    {
        Val=Speed_L5/100;//取出百位
        if(Speed_L5 > 999)//大于 999 时
            Val=Val%10;//去除百位
        switch(Val)
        {
            case 0:seg23&=0xfb;seg22|=0x0c;seg23|=0x08;seg24|=0x0c;seg25|=0x08;//数字 0
                break;
            case
1:seg22&=0xf3;seg23&=0xf3;seg24&=0xf3;seg25&=0xf3;seg24|=0x08;seg25|=0x08;//数字 1
                break;
            case
2:seg22&=0xf3;seg23&=0xf3;seg24&=0xf3;seg25&=0xf3;seg22|=0x04;seg23|=0x0c;seg24|=0x0
c;//数字 2
                break;
            case
3:seg22&=0xf3;seg23&=0xf3;seg24&=0xf3;seg25&=0xf3;seg23|=0x0c;seg24|=0x0c;seg25|=0x0
8;//数字 3
                break;
            case
4:seg22&=0xf3;seg23&=0xf3;seg24&=0xf3;seg25&=0xf3;seg22|=0x08;seg23|=0x04;seg24|=0x0
8;seg25|=0x08;//数字 4
                break;
            case
5:seg22&=0xf3;seg23&=0xf3;seg24&=0xf3;seg25&=0xf3;seg22|=0x08;seg23|=0x0c;seg24|=0x0
4;seg25|=0x08;//数字 5
                break;
            case
6:seg22&=0xf3;seg23&=0xf3;seg24&=0xf3;seg25&=0xf3;seg22|=0x0c;seg23|=0x0c;seg24|=0x0
4;seg25|=0x08;//数字 6
                break;
            case
7:seg22&=0xf3;seg23&=0xf3;seg24&=0xf3;seg25&=0xf3;seg22|=0x00;seg23|=0x08;seg24|=0x0
8;seg25|=0x08;//数字 7
                break;
            case
8:seg22&=0xf3;seg23&=0xf3;seg24&=0xf3;seg25&=0xf3;seg22|=0x0c;seg23|=0x0c;seg24|=0x0
c;seg25|=0x08;//数字 8
                break;
            case
9:seg22&=0xf3;seg23&=0xf3;seg24&=0xf3;seg25&=0xf3;seg22|=0x08;seg23|=0x0c;seg24|=0x0
c;seg25|=0x08;//数字 9
                break;
            default:
                break;
        }
}
else
{
        seg23&=0xfb;seg22|=0x0c;seg23|=0x08;seg24|=0x0c;seg25|=0x08;//数字 0
}

/***********L5 十位***********/
if(Speed_L5 > 9)//大于 9 时
{
        Val=Speed_L5/10;//取出十位
        if(Speed_L5 > 99)//大于 99 时
            Val=Val%10;//取出十位
        switch(Val)
        {
            case 0:seg27&=0xfb;seg26|=0x0c;seg27|=0x08;seg28|=0x0c;seg29|=0x04;//数字 0
                break;
            case
1:seg26&=0xf3;seg27&=0xf3;seg28&=0xf3;seg29&=0xf3;seg28|=0x08;seg29|=0x04;//数字 1
                break;
            case
2:seg26&=0xf3;seg27&=0xf3;seg28&=0xf3;seg29&=0xf3;seg26|=0x04;seg27|=0x0c;seg28|=0x0
c;//数字 2
                break;
            case
3:seg26&=0xf3;seg27&=0xf3;seg28&=0xf3;seg29&=0xf3;seg27|=0x0c;seg28|=0x0c;seg29|=0x0
4;//数字 3
                break;
            case
4:seg26&=0xf3;seg27&=0xf3;seg28&=0xf3;seg29&=0xf3;seg26|=0x08;seg27|=0x04;seg28|=0x0
8;seg29|=0x04;//数字 4
                break;
            case
5:seg26&=0xf3;seg27&=0xf3;seg28&=0xf3;seg29&=0xf3;seg26|=0x08;seg27|=0x0c;seg28|=0x0
4;seg29|=0x04;//数字 5
                break;
            case
6:seg26&=0xf3;seg27&=0xf3;seg28&=0xf3;seg29&=0xf3;seg26|=0x0c;seg27|=0x0c;seg28|=0x0
4;seg29|=0x04;//数字 6
                break;
            case
7:seg26&=0xf3;seg27&=0xf3;seg28&=0xf3;seg29&=0xf3;seg26|=0x00;seg27|=0x08;seg28|=0x0
8;seg29|=0x04;//数字 7
                break;
            case
8:seg26&=0xf3;seg27&=0xf3;seg28&=0xf3;seg29&=0xf3;seg26|=0x0c;seg27|=0x0c;seg28|=0x0
c;seg29|=0x04;//数字 8
                break;
            case
9:seg26&=0xf3;seg27&=0xf3;seg28&=0xf3;seg29&=0xf3;seg26|=0x08;seg27|=0x0c;seg28|=0x0
c;seg29|=0x04;//数字 9
```

```
                break;
            default:
                break;
        }
    }
    else
    {
        seg27&=0xfb;seg26|=0x0c;seg27|=0x08;seg28|=0x0c;seg29|=0x04;//数字 0
    }

    /***********L5 个位***********/
    Val=Speed_L5%10;//去除十位
    if(Val > 4)//大于 4 时
    {
        seg29&=0xf7;seg30&=0xf3;seg29|=0x08;seg30|=0x08;//显示 5
    }
    else
    {
        seg29&=0xf7;seg30&=0xf3;seg29|=0x00;seg30|=0x0c;//显示 0
    }

    /***********L7 千位***********/
    if(Speed_L7 > 999)//大于 999 时
    {
        seg21&=0xfe;seg21|=0x02;//显示 1
    }
    else//小于 999 时
    {
        seg21|=0x03;//显示 0
    }

    /***********L7 百位***********/
    if(Speed_L7 > 99)//大于 99 时
    {
        Val=Speed_L7/100;//取出百位
        if(Speed_L7 > 999)//大于 999 时
            Val=Val%10;//去除百位
        switch(Val)
        {
            case 0:seg23&=0xfe;seg22|=0x03;seg23|=0x02;seg24|=0x03;seg25|=0x02;//数字 0
                break;
            case
1:seg22&=0xfc;seg23&=0xfc;seg24&=0xfc;seg25&=0xfc;seg24|=0x02;seg25|=0x02;//数字 1
                break;
            case
2:seg22&=0xfc;seg23&=0xfc;seg24&=0xfc;seg25&=0xfc;seg22|=0x01;seg23|=0x03;seg24|=0x03
;//数字 2
                break;
            case
3:seg22&=0xfc;seg23&=0xfc;seg24&=0xfc;seg25&=0xfc;seg23|=0x03;seg24|=0x03;seg25|=0x02
```

```
;//数字 3
                        break;
                    case
4:seg22&=0xfc;seg23&=0xfc;seg24&=0xfc;seg25&=0xfc;seg22|=0x02;seg23|=0x01;seg24|=0x02
;seg25|=0x02;//数字 4
                        break;
                    case
5:seg22&=0xfc;seg23&=0xfc;seg24&=0xfc;seg25&=0xfc;seg22|=0x02;seg23|=0x03;seg24|=0x01
;seg25|=0x02;//数字 5
                        break;
                    case
6:seg22&=0xfc;seg23&=0xfc;seg24&=0xfc;seg25&=0xfc;seg22|=0x03;seg23|=0x03;seg24|=0x01
;seg25|=0x02;//数字 6
                        break;
                    case
7:seg22&=0xfc;seg23&=0xfc;seg24&=0xfc;seg25&=0xfc;seg22|=0x00;seg23|=0x02;seg24|=0x02
;seg25|=0x02;//数字 7
                        break;
                    case
8:seg22&=0xfc;seg23&=0xfc;seg24&=0xfc;seg25&=0xfc;seg22|=0x03;seg23|=0x03;seg24|=0x03
;seg25|=0x02;//数字 8
                        break;
                    case
9:seg22&=0xfc;seg23&=0xfc;seg24&=0xfc;seg25&=0xfc;seg22|=0x02;seg23|=0x03;seg24|=0x03
;seg25|=0x02;//数字 9
                        break;
                    default:
                        break;
                }
            }
        else
        {
            seg23&=0xfe;seg22|=0x03;seg23|=0x02;seg24|=0x03;seg25|=0x02;//数字 0
        }

        /***********L7 十位***********/
        if(Speed_L7 > 9)//大于 9 时
        {
            Val=Speed_L7/10;//取出十位
            if(Speed_L7 > 99)//大于 99 时
                Val=Val%10;//取出十位
            switch(Val)
            {
                case 0:seg27&=0xfe;seg26|=0x03;seg27|=0x02;seg28|=0x03;seg29|=0x01;//数字 0
                        break;
                    case
1:seg26&=0xfc;seg27&=0xfc;seg28&=0xfc;seg29&=0xfc;seg28|=0x02;seg29|=0x01;//数字 1
                        break;
                    case
2:seg26&=0xfc;seg27&=0xfc;seg28&=0xfc;seg29&=0xfc;seg26|=0x01;seg27|=0x03;seg28|=0x03
```

```
;//数字 2
                break;
            case
3:seg26&=0xfc;seg27&=0xfc;seg28&=0xfc;seg29&=0xfc;seg27|=0x03;seg28|=0x03;seg29|=0x01
;//数字 3
                break;
            case
4:seg26&=0xfc;seg27&=0xfc;seg28&=0xfc;seg29&=0xfc;seg26|=0x02;seg27|=0x01;seg28|=0x02
;seg29|=0x01;//数字 4
                break;
            case
5:seg26&=0xfc;seg27&=0xfc;seg28&=0xfc;seg29&=0xfc;seg26|=0x02;seg27|=0x03;seg28|=0x01
;seg29|=0x01;//数字 5
                break;
            case
6:seg26&=0xfc;seg27&=0xfc;seg28&=0xfc;seg29&=0xfc;seg26|=0x03;seg27|=0x03;seg28|=0x01
;seg29|=0x01;//数字 6
                break;
            case
7:seg26&=0xfc;seg27&=0xfc;seg28&=0xfc;seg29&=0xfc;seg26|=0x00;seg27|=0x02;seg28|=0x02
;seg29|=0x01;//数字 7
                break;
            case
8:seg26&=0xfc;seg27&=0xfc;seg28&=0xfc;seg29&=0xfc;seg26|=0x03;seg27|=0x03;seg28|=0x03
;seg29|=0x01;//数字 8
                break;
            case
9:seg26&=0xfc;seg27&=0xfc;seg28&=0xfc;seg29&=0xfc;seg26|=0x02;seg27|=0x03;seg28|=0x03
;seg29|=0x01;//数字 9
                break;
            default:
                break;
        }
    }
    else
    {
        seg27&=0xfe;seg26|=0x03;seg27|=0x02;seg28|=0x03;seg29|=0x01;//数字 0
    }

    /***********L7 个位***********/
    Val=Speed_L7%10;//去除十位
    if(Val > 4)//大于 4 时
    {
        seg29&=0xfd;seg30&=0xfc;seg29|=0x02;seg30|=0x02;//显示 5
    }
    else
    {
        seg29&=0xfd;seg30&=0xfc;seg29|=0x00;seg30|=0x03;//显示 0
    }
```

```
	if(Work_All)//在全局设置时
	{
		seg25&=0xAA;seg25|=0x55;//L1 和 L3 和 L5 和 L7 下面的下划线
		if(Work_All_Time)//选中工位 1 和熄灭下划线时
		{
			switch(Work_Option)
			{
				case 1:seg25&=0xAA;seg25|=0x15;//L1 下划线熄灭
					break;
				case 3:seg25&=0xAA;seg25|=0x45;//L3 下划线熄灭
					break;
				case 5:seg25&=0xAA;seg25|=0x51;//L5 下划线熄灭
					break;
				case 7:seg25&=0xAA;seg25|=0x54;//L7 下划线熄灭
					break;
				default:seg25&=0xAA;seg25|=0x55;//全部亮
					break;
			}
		}
	}
	else
	{
		switch(Work_Option)//工位号选择
		{
			case 1:seg25&=0xAA;seg25|=0x40;//显示 L1 下面的下划线
				break;
			case 3:seg25&=0xAA;seg25|=0x10;//显示 L3 下面的下划线
				break;
			case 5:seg25&=0xAA;seg25|=0x04;//显示 L5 下面的下划线
				break;
			case 7:seg25&=0xAA;seg25|=0x01;//显示 L7 下面的下划线
				break;
			default:
				break;
		}
	}

	#if(Integration_TYPE == 0)//设置为四联时 L3 和 L5 不显示

seg21&=0xC3;seg22&=0xC3;seg23&=0xC3;seg24&=0xC3;seg25&=0xC3;seg26&=0xC3;seg27
&=0xC3;seg28&=0xC3;seg29&=0xC3;seg30&=0xC3;
	#elif(Integration_TYPE == 1)//设置为六联时 L7 不显示

seg21&=0xFC;seg22&=0xFC;seg23&=0xFC;seg24&=0xFC;seg25&=0xFC;seg26&=0xFC;seg27
&=0xFC;seg28&=0xFC;seg29&=0xFC;seg30&=0xFC;
	#endif

	Write_Addr_Dat_N(40, seg21,1);
	Write_Addr_Dat_N(42, seg22,1);
	Write_Addr_Dat_N(44, seg23,1);
```

```
    Write_Addr_Dat_N(46, seg24,1);
    Write_Addr_Dat_N(48, seg25,1);
    Write_Addr_Dat_N(50, seg26,1);
    Write_Addr_Dat_N(52, seg27,1);
    Write_Addr_Dat_N(54, seg28,1);
    Write_Addr_Dat_N(56, seg29,1);
    Write_Addr_Dat_N(58, seg30,1);
}

/*
******************************************************************
 * 函数原型：  void Display_Speed_Right(uint16_t Speed_L2,uint16_t Speed_L8)
 * 功     能：  写最右边的设定速度
 * 输     入:   Speed_L2 右边最上面的速度  Speed_L8 右边最上面的速度
 * 参     数：  uint16_t Speed_L2,uint16_t Speed_L8
******************************************************************
*/
void Display_Speed_Right(uint16_t Speed_L2,uint16_t Speed_L4,uint16_t Speed_L6,uint16_t Speed_L8)
{

    uint8_t seg1,seg2,seg3,seg4,seg5,seg6,seg7,seg8,seg9,seg10;
    seg1=0;seg2=0;seg3=0;seg4=0;seg5=0;seg6=0;seg7=0;seg8=0;seg9=0;seg10=0;
    uint8_t Val;//用于百十个取出来的数字

    /***********L2 千位***********/
    if(Speed_L2 > 999)//大于 999 时
    {
        seg2&=0xdf;seg1|=0x80;//显示 1
    }
    else//小于 999 时
    {
        seg1|=0xc0;//显示 0
    }

    /***********L2 百位***********/
    if(Speed_L2 > 99)//大于 99 时
    {
        Val=Speed_L2/100;//取出百位
        if(Speed_L2 > 999)//大于 999 时
            Val=Val%10;//去除百位
        switch(Val)
        {
            case  0:seg2&=0x3f;seg3&=0xbf;seg2|=0xc0;seg3|=0x80;seg4|=0xc0;seg5|=0x80;//数字 0
                break;
            case 1:seg2&=0x3f;seg3&=0x3f;seg4&=0x3f;seg5&=0x3f;seg4|=0x80;seg5|=0x80;//数字 1
                break;
            case
```

```
2:seg2&=0x3f;seg3&=0x3f;seg4&=0x3f;seg5&=0x3f;seg2|=0x40;seg3|=0xc0;seg4|=0xc0;//数字
2
                break;
            case
3:seg2&=0x3f;seg3&=0x3f;seg4&=0x3f;seg5&=0x3f;seg3|=0xc0;seg4|=0xc0;seg5|=0x80;//数字
3
                break;
            case
4:seg2&=0x3f;seg3&=0x3f;seg4&=0x3f;seg5&=0x3f;seg2|=0x80;seg3|=0x40;seg4|=0x80;seg5|=
0x80;//数字 4
                break;
            case
5:seg2&=0x3f;seg3&=0x3f;seg4&=0x3f;seg5&=0x3f;seg2|=0x80;seg3|=0xc0;seg4|=0x40;seg5|=
0x80;//数字 5
                break;
            case
6:seg2&=0x3f;seg3&=0x3f;seg4&=0x3f;seg5&=0x3f;seg2|=0xc0;seg3|=0xc0;seg4|=0x40;seg5|=0
x80;//数字 6
                break;
            case
7:seg2&=0x3f;seg3&=0x3f;seg4&=0x3f;seg5&=0x3f;seg2|=0x00;seg3|=0x80;seg4|=0x80;seg5|=
0x80;//数字 7
                break;
            case
8:seg2&=0x3f;seg3&=0x3f;seg4&=0x3f;seg5&=0x3f;seg2|=0xc0;seg3|=0xc0;seg4|=0xc0;seg5|=0
x80;//数字 8
                break;
            case
9:seg2&=0x3f;seg3&=0x3f;seg4&=0x3f;seg5&=0x3f;seg2|=0x80;seg3|=0xc0;seg4|=0xc0;seg5|=0
x80;//数字 9
                break;
            default:
                break;
        }
    }
    else
    {
        seg3&=0xbf;seg2|=0xc0;seg3|=0x80;seg4|=0xc0;seg5|=0x80;//数字 0
    }

    /***********L2 十位***********/
    if(Speed_L2 > 9)//大于 9 时
    {
        Val=Speed_L2/10;//取出十位
        if(Speed_L2 > 99)//大于 99 时
            Val=Val%10;//取出十位
        switch(Val)
        {
            case 0:seg7&=0xbf;seg6|=0xc0;seg7|=0x80;seg8|=0xc0;seg9|=0x40;//数字 0
                break;
                case
1:seg6&=0x3f;seg7&=0x3f;seg8&=0x3f;seg9&=0x3f;seg8|=0x80;seg9|=0x40;//数字 1
                    break;
                case
2:seg6&=0x3f;seg7&=0x3f;seg8&=0x3f;seg9&=0x3f;seg6|=0x40;seg7|=0xc0;seg8|=0xc0;//数字
2
                    break;
                case
3:seg6&=0x3f;seg7&=0x3f;seg8&=0x3f;seg9&=0x3f;seg7|=0xc0;seg8|=0xc0;seg9|=0x40;//数字
3
                    break;
                case
4:seg6&=0x3f;seg7&=0x3f;seg8&=0x3f;seg9&=0x3f;seg6|=0x80;seg7|=0x40;seg8|=0x80;seg9|=
0x40;//数字 4
                    break;
                case
5:seg6&=0x3f;seg7&=0x3f;seg8&=0x3f;seg9&=0x3f;seg6|=0x80;seg7|=0xc0;seg8|=0x40;seg9|=
0x40;//数字 5
                    break;
                case
6:seg6&=0x3f;seg7&=0x3f;seg8&=0x3f;seg9&=0x3f;seg6|=0xc0;seg7|=0xc0;seg8|=0x40;seg9|=0
x40;//数字 6
                    break;
                case
7:seg6&=0x3f;seg7&=0x3f;seg8&=0x3f;seg9&=0x3f;seg6|=0x00;seg7|=0x80;seg8|=0x80;seg9|=
0x40;//数字 7
                    break;
                case
8:seg6&=0x3f;seg7&=0x3f;seg8&=0x3f;seg9&=0x3f;seg6|=0xc0;seg7|=0xc0;seg8|=0xc0;seg9|=0
x40;//数字 8
                    break;
                case
9:seg6&=0x3f;seg7&=0x3f;seg8&=0x3f;seg9&=0x3f;seg6|=0x80;seg7|=0xc0;seg8|=0xc0;seg9|=0
x40;//数字 9
                    break;
                default:
                    break;
            }
        }
        else
        {
            seg7&=0xbf;seg6|=0xc0;seg7|=0x80;seg8|=0xc0;seg9|=0x40;//数字 0
        }

        /***********L2 个位***********/
        Val=Speed_L2%10;//取出个位
        if(Val > 4)//大于 4 时
        {
            seg9&=0x7f;seg10&=0x3f;seg9|=0x80;seg10|=0x80;//显示 5
        }
```

```
    else//小于 4 时
    {
        seg9&=0x7f;seg10&=0x3f;seg9|=0x00;seg10|=0xc0;//显示 0
    }

    /***********L4 千位***********/
    if(Speed_L4 > 999)//大于 999 时
    {
        seg1&=0xcf;seg1|=0x20;//显示 1
    }
    else//小于 999 时
    {
        seg1|=0x30;//显示 0
    }

    /***********L4 百位***********/
    if(Speed_L4 > 99)//大于 99 时
    {
        Val=Speed_L4/100;//取出百位
        if(Speed_L4 > 999)//大于 999 时
            Val=Val%10;//去除百位
        switch(Val)
        {
            case
0:seg2&=0xCf;seg3&=0xCf;seg2|=0x30;seg3|=0x20;seg4|=0x30;seg5|=0x20;//数字 0
                break;
            case
1:seg2&=0xCf;seg3&=0xCf;seg4&=0xCf;seg5&=0xCf;seg4|=0x20;seg5|=0x20;//数字 1
                break;
            case
2:seg2&=0xCf;seg3&=0xCf;seg4&=0xCf;seg5&=0xCf;seg2|=0x10;seg3|=0x30;seg4|=0x30;//  数
字 2
                break;
            case
3:seg2&=0xCf;seg3&=0xCf;seg4&=0xCf;seg5&=0xCf;seg3|=0x30;seg4|=0x30;seg5|=0x20;//  数
字 3
                break;
            case
4:seg2&=0xCf;seg3&=0xCf;seg4&=0xCf;seg5&=0xCf;seg2|=0x20;seg3|=0x10;seg4|=0x20;seg5|
=0x20;//数字 4
                break;
            case
5:seg2&=0xCf;seg3&=0xCf;seg4&=0xCf;seg5&=0xCf;seg2|=0x20;seg3|=0x30;seg4|=0x10;seg5|
=0x20;//数字 5
                break;
            case
6:seg2&=0xCf;seg3&=0xCf;seg4&=0xCf;seg5&=0xCf;seg2|=0x30;seg3|=0x30;seg4|=0x10;seg5|
=0x20;//数字 6
                break;
            case
```

```
7:seg2&=0xCf;seg3&=0xCf;seg4&=0xCf;seg5&=0xCf;seg2|=0x00;seg3|=0x20;seg4|=0x20;seg5|
=0x20;//数字 7
                break;
            case
8:seg2&=0xCf;seg3&=0xCf;seg4&=0xCf;seg5&=0xCf;seg2|=0x30;seg3|=0x30;seg4|=0x30;seg5|
=0x20;//数字 8
                break;
            case
9:seg2&=0xCf;seg3&=0xCf;seg4&=0xCf;seg5&=0xCf;seg2|=0x20;seg3|=0x30;seg4|=0x30;seg5|
=0x20;//数字 9
                break;
            default:
                break;
        }
    }
    else
    {
        seg3&=0xef;seg2|=0x30;seg3|=0x20;seg4|=0x30;seg5|=0x20;//数字 0
    }

    /***********L4 十位***********/
    if(Speed_L4 > 9)//大于 9 时
    {
        Val=Speed_L4/10;//取出十位
        if(Speed_L4 > 99)//大于 99 时
            Val=Val%10;//取出十位
        switch(Val)
        {
            case 0:seg7&=0xef;seg6|=0x30;seg7|=0x20;seg8|=0x30;seg9|=0x10;//数字 0
                break;
            case
1:seg6&=0xCf;seg7&=0xCf;seg8&=0xCf;seg9&=0xCf;seg8|=0x20;seg9|=0x10;//数字 1
                break;
            case
2:seg6&=0xCf;seg7&=0xCf;seg8&=0xCf;seg9&=0xCf;seg6|=0x10;seg7|=0x30;seg8|=0x30;//  数
字 2
                break;
            case
3:seg6&=0xCf;seg7&=0xCf;seg8&=0xCf;seg9&=0xCf;seg7|=0x30;seg8|=0x30;seg9|=0x10;//  数
字 3
                break;
            case
4:seg6&=0xCf;seg7&=0xCf;seg8&=0xCf;seg9&=0xCf;seg6|=0x20;seg7|=0x10;seg8|=0x20;seg9|
=0x10;//数字 4
                break;
            case
5:seg6&=0xCf;seg7&=0xCf;seg8&=0xCf;seg9&=0xCf;seg6|=0x20;seg7|=0x30;seg8|=0x10;seg9|
=0x10;//数字 5
                break;
            case
```

```
6:seg6&=0xCf;seg7&=0xCf;seg8&=0xCf;seg9&=0xCf;seg6|=0x30;seg7|=0x30;seg8|=0x10;seg9|
=0x10;//数字 6
                break;
            case
7:seg6&=0xCf;seg7&=0xCf;seg8&=0xCf;seg9&=0xCf;seg6|=0x00;seg7|=0x20;seg8|=0x20;seg9|
=0x10;//数字 7
                break;
            case
8:seg6&=0xCf;seg7&=0xCf;seg8&=0xCf;seg9&=0xCf;seg6|=0x30;seg7|=0x30;seg8|=0x30;seg9|
=0x10;//数字 8
                break;
            case
9:seg6&=0xCf;seg7&=0xCf;seg8&=0xCf;seg9&=0xCf;seg6|=0x20;seg7|=0x30;seg8|=0x30;seg9|
=0x10;//数字 9
                break;
            default:
                break;
        }
    }
    else
    {
        seg7&=0xef;seg6|=0x30;seg7|=0x20;seg8|=0x30;seg9|=0x10;//数字 0
    }

    /***********L4 个位***********/
    Val=Speed_L4%10;//取出个位
    if(Val > 4)//大于 4 时
    {
        seg9&=0xdf;seg10&=0xcf;seg9|=0x20;seg10|=0x20;//显示 5
    }
    else//小于 4 时
    {
        seg9&=0xdf;seg10&=0xcf;seg9|=0x00;seg10|=0x30;//显示 0
    }

    /***********L6 千位***********/
    if(Speed_L6 > 999)//大于 999 时
    {
        seg1|=0x08;seg1&=0xfb;//显示 1
    }
    else//小于 999 时
    {
        seg1|=0x0c;//显示 0
    }

    /***********L6 百位***********/
    if(Speed_L6 > 99)//大于 99 时
    {
        Val=Speed_L6/100;//取出百位
        if(Speed_L6 > 999)//大于 999 时
```

```
            Val=Val%10;//去除百位
        switch(Val)
        {
            case 0:seg3&=0xfb;seg2|=0x0c;seg3|=0x08;seg4|=0x0c;seg5|=0x08;//数字 0
                break;
            case
1:seg2&=0xf3;seg3&=0xf3;seg4&=0xf3;seg5&=0xf3;seg4|=0x08;seg5|=0x08;//数字 1
                break;
            case
2:seg2&=0xf3;seg3&=0xf3;seg4&=0xf3;seg5&=0xf3;seg2|=0x04;seg3|=0x0c;seg4|=0x0c;//数字
2
                break;
            case
3:seg2&=0xf3;seg3&=0xf3;seg4&=0xf3;seg5&=0xf3;seg3|=0x0c;seg4|=0x0c;seg5|=0x08;//数字
3
                break;
            case
4:seg2&=0xf3;seg3&=0xf3;seg4&=0xf3;seg5&=0xf3;seg2|=0x08;seg3|=0x04;seg4|=0x08;seg5|=
0x08;//数字 4
                break;
            case
5:seg2&=0xf3;seg3&=0xf3;seg4&=0xf3;seg5&=0xf3;seg2|=0x08;seg3|=0x0c;seg4|=0x04;seg5|=
0x08;//数字 5
                break;
            case
6:seg2&=0xf3;seg3&=0xf3;seg4&=0xf3;seg5&=0xf3;seg2|=0x0c;seg3|=0x0c;seg4|=0x04;seg5|=0
x08;//数字 6
                break;
            case
7:seg2&=0xf3;seg3&=0xf3;seg4&=0xf3;seg5&=0xf3;seg2|=0x00;seg3|=0x08;seg4|=0x08;seg5|=
0x08;//数字 7
                break;
            case
8:seg2&=0xf3;seg3&=0xf3;seg4&=0xf3;seg5&=0xf3;seg2|=0x0c;seg3|=0x0c;seg4|=0x0c;seg5|=0
x08;//数字 8
                break;
            case
9:seg2&=0xf3;seg3&=0xf3;seg4&=0xf3;seg5&=0xf3;seg2|=0x08;seg3|=0x0c;seg4|=0x0c;seg5|=0
x08;//数字 9
                break;
            default:
                break;
        }
    }
    else
    {
        seg3&=0xfb;seg2|=0x0c;seg3|=0x08;seg4|=0x0c;seg5|=0x08;//数字 0
    }

    /***********L6 十位***********/
    if(Speed_L6 > 9)//大于 9 时
    {
        Val=Speed_L6/10;//取出十位
        if(Speed_L6 > 99)//大于 99 时
            Val=Val%10;//取出十位
        switch(Val)
        {
            case 0: seg7&=0xfb;seg6|=0x0c;seg7|=0x08;seg8|=0x0c;seg9|=0x04;//数字 0
                break;
            case                                                                   1:
seg6&=0xf3;seg7&=0xf3;seg8&=0xf3;seg9&=0xf3;seg8|=0x08;seg9|=0x04;//数字 1
                break;
            case                                                                   2:
seg6&=0xf3;seg7&=0xf3;seg8&=0xf3;seg9&=0xf3;seg6|=0x04;seg7|=0x0c;seg8|=0x0c;//数字 2
                break;
            case                                                                   3:
seg6&=0xf3;seg7&=0xf3;seg8&=0xf3;seg9&=0xf3;seg7|=0x0c;seg8|=0x0c;seg9|=0x04;//数字 3
                break;
            case                                                                   4:
seg6&=0xf3;seg7&=0xf3;seg8&=0xf3;seg9&=0xf3;seg6|=0x08;seg7|=0x04;seg8|=0x08;seg9|=0x
04;//数字 4
                break;
            case                                                                   5:
seg6&=0xf3;seg7&=0xf3;seg8&=0xf3;seg9&=0xf3;seg6|=0x08;seg7|=0x0c;seg8|=0x04;seg9|=0x
04;//数字 5
                break;
            case                                                                   6:
seg6&=0xf3;seg7&=0xf3;seg8&=0xf3;seg9&=0xf3;seg6|=0x0c;seg7|=0x0c;seg8|=0x04;seg9|=0x
04;//数字 6
                break;
            case                                                                   7:
seg6&=0xf3;seg7&=0xf3;seg8&=0xf3;seg9&=0xf3;seg6|=0x00;seg7|=0x08;seg8|=0x08;seg9|=0x
04;//数字 7
                break;
            case                                                                   8:
seg6&=0xf3;seg7&=0xf3;seg8&=0xf3;seg9&=0xf3;seg6|=0x0c;seg7|=0x0c;seg8|=0x0c;seg9|=0x
04;//数字 8
                break;
            case                                                                   9:
seg6&=0xf3;seg7&=0xf3;seg8&=0xf3;seg9&=0xf3;seg6|=0x08;seg7|=0x0c;seg8|=0x0c;seg9|=0x
04;//数字 9
                break;
            default:
                break;
        }
    }
    else
    {
        seg7&=0xfb;seg6|=0x0c;seg7|=0x08;seg8|=0x0c;seg9|=0x04;//数字 0
    }
```

```
     /***********L6 个位***********/
     Val=Speed_L6%10;//取出个位
     if(Val > 4)//大于 4 时
     {
          seg9&=0xf7;seg10&=0xf3;seg9|=0x08;seg10|=0x08;//显示 5
     }
     else//小于 4 时
     {
          seg9&=0xf7;seg10&=0xf3;seg9|=0x00;seg10|=0x0c;//显示 0
     }

     /***********L8 千位***********/
     if(Speed_L8 > 999)//大于 999 时
     {
          seg1&=0xfe;seg1|=0x02;//显示 1
     }
     else//小于 999 时
     {
          seg1|=0x03;//显示 0
     }

     /***********L8 百位***********/
     if(Speed_L8 > 99)//大于 99 时
     {
          Val=Speed_L8/100;//取出百位
          if(Speed_L8 > 999)//大于 999 时
               Val=Val%10;//取出百位
          switch(Val)
          {
               case 0:seg3&=0xfe;seg2|=0x03;seg3|=0x02;seg4|=0x03;seg5|=0x02;//数字 0
                    break;
               case
1:seg2&=0xfc;seg3&=0xfc;seg4&=0xfc;seg5&=0xfc;seg4|=0x02;seg5|=0x02;//数字 1
                    break;
               case
2:seg2&=0xfc;seg3&=0xfc;seg4&=0xfc;seg5&=0xfc;seg2|=0x01;seg3|=0x03;seg4|=0x03;//数字 2
                    break;
               case
3:seg2&=0xfc;seg3&=0xfc;seg4&=0xfc;seg5&=0xfc;seg3|=0x03;seg4|=0x03;seg5|=0x02;//数字 3
                    break;
               case
4:seg2&=0xfc;seg3&=0xfc;seg4&=0xfc;seg5&=0xfc;seg2|=0x02;seg3|=0x01;seg4|=0x02;seg5|=0
x02;//数字 4
                    break;
               case
5:seg2&=0xfc;seg3&=0xfc;seg4&=0xfc;seg5&=0xfc;seg2|=0x02;seg3|=0x03;seg4|=0x01;seg5|=0
x02;//数字 5
                    break;
               case
```

```
6:seg2&=0xfc;seg3&=0xfc;seg4&=0xfc;seg5&=0xfc;seg2|=0x03;seg3|=0x03;seg4|=0x01;seg5|=0
x02;//数字 6
                break;
            case
7:seg2&=0xfc;seg3&=0xfc;seg4&=0xfc;seg5&=0xfc;seg2|=0x00;seg3|=0x02;seg4|=0x02;seg5|=0
x02;//数字 7
                break;
            case
8:seg2&=0xfc;seg3&=0xfc;seg4&=0xfc;seg5&=0xfc;seg2|=0x03;seg3|=0x03;seg4|=0x03;seg5|=0
x02;//数字 8
                break;
            case
9:seg2&=0xfc;seg3&=0xfc;seg4&=0xfc;seg5&=0xfc;seg2|=0x02;seg3|=0x03;seg4|=0x03;seg5|=0
x02;//数字 9
                break;
            default:
                break;
        }
    }
    else
    {
        seg3&=0xfe;seg2|=0x03;seg3|=0x02;seg4|=0x03;seg5|=0x02;//数字 0
    }

    /***********L8 十位***********/
    if(Speed_L8 > 9)//大于 9 时
    {
        Val=Speed_L8/10;//取出十位
        if(Speed_L8 > 99)//大于 99 时
            Val=Val%10;//取出十位
        switch(Val)
        {
            case 0:seg7&=0xfe;seg6|=0x03;seg7|=0x02;seg8|=0x03;seg9|=0x01;//数字 0
                break;
            case
1:seg6&=0xfc;seg7&=0xfc;seg8&=0xfc;seg9&=0xfc;seg8|=0x02;seg9|=0x01;//数字 1
                break;
            case
2:seg6&=0xfc;seg7&=0xfc;seg8&=0xfc;seg9&=0xfc;seg6|=0x01;seg7|=0x03;seg8|=0x03;//数字 2
                break;
            case
3:seg6&=0xfc;seg7&=0xfc;seg8&=0xfc;seg9&=0xfc;seg7|=0x03;seg8|=0x03;seg9|=0x01;//数字 3
                break;
            case
4:seg6&=0xfc;seg7&=0xfc;seg8&=0xfc;seg9&=0xfc;seg6|=0x02;seg7|=0x01;seg8|=0x02;seg9|=0
x01;//数字 4
                break;
            case
5:seg6&=0xfc;seg7&=0xfc;seg8&=0xfc;seg9&=0xfc;seg6|=0x02;seg7|=0x03;seg8|=0x01;seg9|=0
x01;//数字 5
```

```
                break;
            case
6:seg6&=0xfc;seg7&=0xfc;seg8&=0xfc;seg9&=0xfc;seg6|=0x03;seg7|=0x03;seg8|=0x01;seg9|=0
x01;//数字 6
                break;
            case
7:seg6&=0xfc;seg7&=0xfc;seg8&=0xfc;seg9&=0xfc;seg6|=0x00;seg7|=0x02;seg8|=0x02;seg9|=0
x01;//数字 7
                break;
            case
8:seg6&=0xfc;seg7&=0xfc;seg8&=0xfc;seg9&=0xfc;seg6|=0x03;seg7|=0x03;seg8|=0x03;seg9|=0
x01;//数字 8
                break;
            case
9:seg6&=0xfc;seg7&=0xfc;seg8&=0xfc;seg9&=0xfc;seg6|=0x02;seg7|=0x03;seg8|=0x03;seg9|=0
x01;//数字 9
                break;
            default:
                break;
        }
    }
    else
    {
        seg7&=0xfe;seg6|=0x03;seg7|=0x02;seg8|=0x03;seg9|=0x01;//数字 0
    }

    /***********L8 个位***********/
    Val=Speed_L8%10;//取出个位
    if(Val > 4)//大于 4 时
    {
        seg9&=0xfd;seg10&=0xfc;seg9|=0x02;seg10|=0x02;//显示 5
    }
    else//小于 4 时
    {
        seg9&=0xfd;seg10&=0xfc;seg9|=0x00;seg10|=0x03;//显示 0
    }

    if(Work_All)//在全局设置时
    {
        seg5&=0xAA;seg5|=0x55;//L2 和 L4 和 L6 和 L8 下划线
        if(Work_All_Time)//选中工位 1 和熄灭下划线时
        {
            switch(Work_Option)
            {
                case 2:seg5&=0xAA;seg5|=0x15;//L2 下划线熄灭
                    break;
                case 4:seg5&=0xAA;seg5|=0x45;//L4 下划线熄灭
                    break;
                case 6:seg5&=0xAA;seg5|=0x51;//L6 下划线熄灭
                    break;
                case 8:seg5&=0xAA;seg5|=0x54;//L8 下划线熄灭
                    break;
                default:seg5&=0xAA;seg5|=0x55;//全部亮
                    break;
            }
        }
    }
    else
    {
        switch(Work_Option)//工位号选择
        {
            case 2:seg5&=0xAA;seg5|=0x40;//L2 下划线
                break;
            case 4:seg5&=0xAA;seg5|=0x10;//L4 下划线
                break;
            case 6:seg5&=0xAA;seg5|=0x04;//L6 下划线
                break;
            case 8:seg5&=0xAA;seg5|=0x01;//L8 下划线
                break;
            default:
                break;
        }
    }
    #if(Integration_TYPE == 0)//设置为四联时 L2 和 L6 不显示

seg1&=0xC3;seg2&=0xC3;seg3&=0xC3;seg4&=0xC3;seg5&=0xC3;seg6&=0xC3;seg7&=0xC3;
seg8&=0xC3;seg9&=0xC3;seg10&=0xC3;
    #elif(Integration_TYPE == 1)//设置为六联时 L8 不显示

seg1&=0xFC;seg2&=0xFC;seg3&=0xFC;seg4&=0xFC;seg5&=0xFC;seg6&=0xFC;seg7&=0xFC
;seg8&=0xFC;seg9&=0xFC;seg10&=0xFC;
    #endif
    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);
    Write_Addr_Dat_N(16, seg9,1);
    Write_Addr_Dat_N(18, seg10,1);
}

/*
**************************************************************
 * 函数原型：void Display_RelVal(uint16_t dis_rel_rpm,int dis_rel_temp,uint32_t dis_rel_time)
 * 功      能： 显示实际数值和中间图标
 * 输      入: dis_rel_rpm：实际速度 dis_rel_temp：实际温度 dis_rel_time：实际时间
 * 参      数： uint16_t dis_rel_rpm,int dis_rel_temp,uint32_t dis_rel_time
```

```
*****************************************************************
*/
void Display_RelVal(uint16_t dis_rel_rpm,int dis_rel_temp,uint32_t dis_rel_time)
{
    uint8_t seg31,seg32,seg33,seg34,seg35,seg36,seg37,seg38,seg39,seg40;
    seg31=0;seg32=0;seg33=0;seg34=0;seg35=0;seg36=0;seg37=0;seg38=0;seg39=0;seg40=0;
    uint8_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

    /***********rel_rpm 千位***********/
    if(dis_rel_rpm > 999)//大于 999 时
    {
        seg32&=0xF8;seg33&=0xF0;seg33|=0x80;//显示 1
    }
    else//小于 999 时
    {
        seg32&=0xF8;seg33&=0xF0;seg33|=0x80;seg32 |= 0x80;//显示 0
    }

    /***********rel_rpm 百位***********/
    if(dis_rel_rpm > 99)//大于 99 时
    {
        Val=dis_rel_rpm/100;//取出百位
        if(dis_rel_rpm > 999)//大于 999 时
            Val=Val%10;//取出百位
        switch(Val)
        {
            case
0:seg34&=0x1f;seg35&=0x1f;seg36&=0x1f;seg34|=0xc0;seg35|=0xa0;seg36|=0xc0;//数字 0
                break;
            case
1:seg34&=0x1f;seg35&=0x1f;seg36&=0x1f;seg34|=0x00;seg35|=0x00;seg36|=0xc0;//数字 1
                break;
            case
2:seg34&=0x1f;seg35&=0x1f;seg36&=0x1f;seg34|=0x40;seg35|=0xe0;seg36|=0x80;//数字 2
                break;
            case
3:seg34&=0x1f;seg35&=0x1f;seg36&=0x1f;seg34|=0x00;seg35|=0xe0;seg36|=0xc0;//数字 3
                break;
            case
4:seg34&=0x1f;seg35&=0x1f;seg36&=0x1f;seg34|=0x80;seg35|=0x40;seg36|=0xc0;//数字 4
                break;
            case
5:seg34&=0x1f;seg35&=0x1f;seg36&=0x1f;seg34|=0x80;seg35|=0xe0;seg36|=0x40;//数字 5
                break;
            case
6:seg34&=0x1f;seg35&=0x1f;seg36&=0x1f;seg34|=0xc0;seg35|=0xe0;seg36|=0x40;//数字 6
                break;
            case
7:seg34&=0x1f;seg35&=0x1f;seg36&=0x1f;seg34|=0x00;seg35|=0x80;seg36|=0xc0;//数字 7
```

```
                break;
            case
8:seg34&=0x1f;seg35&=0x1f;seg36&=0x1f;seg34|=0xc0;seg35|=0xe0;seg36|=0xc0;//数字 8
                break;
            case
9:seg34&=0x1f;seg35&=0x1f;seg36&=0x1f;seg34|=0x80;seg35|=0xe0;seg36|=0xc0;//数字 9
                break;
            default:
                break;
        }
    }
    else
    {
        seg34&=0x1f;seg35&=0x1f;seg36&=0x1f;seg34|=0xc0;seg35|=0xa0;seg36|=0xc0;// 数
字 0
    }

    /***********rel_rpm 十位***********/
    if(dis_rel_rpm > 9)//大于 9 时
    {
        Val=dis_rel_rpm/10;//取出十位
        if(dis_rel_rpm>99)//大于 99 时
            Val=Val%10;//取出十位
        switch(Val)
        {
            case
0:seg37&=0x1f;seg38&=0x1f;seg39&=0x1f;seg37|=0x60;seg38|=0xa0;seg39|=0xc0;//数字 0
                break;
            case
1:seg37&=0x1f;seg38&=0x1f;seg39&=0x1f;seg37|=0x00;seg38|=0x00;seg39|=0xc0;//数字 1
                break;
            case
2:seg37&=0x1f;seg38&=0x1f;seg39&=0x1f;seg37|=0x20;seg38|=0xe0;seg39|=0x80;//数字 2
                break;
            case
3:seg37&=0x1f;seg38&=0x1f;seg39&=0x1f;seg37|=0x00;seg38|=0xe0;seg39|=0xc0;//数字 3
                break;
            case
4:seg37&=0x1f;seg38&=0x1f;seg39&=0x1f;seg37|=0x40;seg38|=0x40;seg39|=0xc0;//数字 4
                break;
            case
5:seg37&=0x1f;seg38&=0x1f;seg39&=0x1f;seg37|=0x40;seg38|=0xe0;seg39|=0x40;//数字 5
                break;
            case
6:seg37&=0x1f;seg38&=0x1f;seg39&=0x1f;seg37|=0x60;seg38|=0xe0;seg39|=0x40;//数字 6
                break;
            case
7:seg37&=0x1f;seg38&=0x1f;seg39&=0x1f;seg37|=0x00;seg38|=0x80;seg39|=0xc0;//数字 7
                break;
            case
```

```
8:seg37&=0x1f;seg38&=0x1f;seg39&=0x1f;seg37|=0x60;seg38|=0xe0;seg39|=0xc0;//数字 8
                break;
            case
9:seg37&=0x1f;seg38&=0x1f;seg39&=0x1f;seg37|=0x40;seg38|=0xe0;seg39|=0xc0;//数字 9
                break;
            default:
                break;
        }
    }
    else
    {
        seg37&=0x1f;seg38&=0x1f;seg39&=0x1f;seg37|=0x60;seg38|=0xa0;seg39|=0xc0;//  数
字 0
    }

    /***********rel_rpm 个位***********/
    Val=dis_rel_rpm%10;//取出个位
    if(Val > 4)//大于 4 时
    {
        seg40&=0x3f;seg40|=0x80;//显示 5
        seg15_Flag = 1;
    }
    else//小于 4 时
    {
        seg40&=0x3f;seg40|=0xc0;//显示 0
        seg15_Flag = 0;
    }
//     seg31&=0x8f;seg31|=0x30;//显示电机反转向的图标
    seg31&=0x8f;seg31|=0x60;//显示电机正转向的图标
    if(Run_Status==1 &&Run_Flag > 1 && Set_Speed.L0 > 0 && Rel_Speed.L0)    //运行时
    {
        seg31&=0x8f;//电机图标消失
    }
    /***********rel_Temp 千位***********/
    if(dis_rel_temp > 999)//大于 999 时
    {
        seg32&=0x87;seg33&=0x8F;seg33|=0x50;//显示 1
    }
    else//小于 999 时
    {
        seg32&=0x87;seg33&=0x8F;seg32|=0x78;seg33|=0x50;//显示 0
    }

    /***********rel_Temp 百位***********/
    if(dis_rel_temp > 99)//大于 99 时
    {
        Val = dis_rel_temp / 100;//取出百位
        if(dis_rel_temp > 999)//大于 999 时
            Val = Val % 10;//取出百位
        switch(Val)
        {
            case
0:seg34&=0xC7;seg35&=0xE7;seg36&=0xCF;seg34|=0x38;seg35|=0x08;seg36|=0x30;//数字 0
                break ;
            case
1:seg34&=0xC7;seg35&=0xE7;seg36&=0xCF;seg34|=0x00;seg35|=0x00;seg36|=0x30;//数字 1
                break ;
            case
2:seg34&=0xC7;seg35&=0xE7;seg36&=0xCF;seg34|=0x00;seg34|=0x28;seg35|=0x18;seg36|=0x
20;//数字 2
                break ;
            case
3:seg34&=0xC7;seg35&=0xE7;seg36&=0xCF;seg34|=0x00;seg34|=0x20;seg35|=0x18;seg36|=0x
30;//数字 3
                break ;
            case
4:seg34&=0xC7;seg35&=0xE7;seg36&=0xCF;seg34|=0x00;seg34|=0x10;seg35|=0x10;seg36|=0x
30;//数字 4
                break ;
            case
5:seg34&=0xC7;seg35&=0xE7;seg36&=0xCF;seg34|=0x00;seg34|=0x30;seg35|=0x18;seg36|=0x
10;//数字 5
                break ;
            case
6:seg34&=0xC7;seg35&=0xE7;seg36&=0xCF;seg34|=0x00;seg34|=0x38;seg35|=0x18;seg36|=0x
10;//数字 6
                break ;
            case
7:seg34&=0xC7;seg35&=0xE7;seg36&=0xCF;seg34|=0x00;seg34|=0x20;seg35|=0x00;seg36|=0x
30;//数字 7
                break ;
            case
8:seg34&=0xC7;seg35&=0xE7;seg36&=0xCF;seg34|=0x00;seg34|=0x38;seg35|=0x18;seg36|=0x
30;//数字 8
                break ;
            case
9:seg34&=0xC7;seg35&=0xE7;seg36&=0xCF;seg34|=0x00;seg34|=0x30;seg35|=0x18;seg36|=0x
30;//数字 9
                break ;
            default:
                break;
        }
    }
    else
    {
        seg34&=0xC7;seg35&=0xE7;seg36&=0xCF;seg34|=0x38;seg35|=0x08;seg36|=0x30;//
数字 0
    }

    /***********rel_Temp 十位***********/
    if(dis_rel_temp > 9)//大于 9 时
    {
        Val = dis_rel_temp / 10;//取出十位
        if(dis_rel_temp > 99)//大于 99 时
            Val = Val % 10;//取出十位
        switch(Val)
        {
            case 0:seg36&=0xF7;seg37&=0xE7;seg38&=0xE7;seg39&=0xD7;seg36|=0x08;seg37|=0x18;seg38|=0x10;seg39|=0x28;//数字 0
                break;
            case 1:seg36&=0xF7;seg37&=0xE7;seg38&=0xE7;seg39&=0xD7;seg36|=0x00;seg37|=0x00;seg38|=0x00;seg39|=0x28;//数字 1
                break;
            case 2:seg36&=0xF7;seg37&=0xE7;seg38&=0xE7;seg39&=0xD7;seg36|=0x08;seg37|=0x08;seg38|=0x18;seg39|=0x20;//数字 2
                break;
            case 3:seg36&=0xF7;seg37&=0xE7;seg38&=0xE7;seg39&=0xD7;seg36|=0x00;seg37|=0x08;seg38|=0x18;seg39|=0x28;//数字 3
                break;
            case 4:seg36&=0xF7;seg37&=0xE7;seg38&=0xE7;seg39&=0xD7;seg36|=0x00;seg37|=0x10;seg38|=0x08;seg39|=0x28;//数字 4
                break;
            case 5:seg36&=0xF7;seg37&=0xE7;seg38&=0xE7;seg39&=0xD7;seg36|=0x00;seg37|=0x18;seg38|=0x18;seg39|=0x08;//数字 5
                break;
            case 6:seg36&=0xF7;seg37&=0xE7;seg38&=0xE7;seg39&=0xD7;seg36|=0x08;seg37|=0x18;seg38|=0x18;seg39|=0x08;//数字 6
                break;
            case 7:seg36&=0xF7;seg37&=0xE7;seg38&=0xE7;seg39&=0xD7;seg36|=0x00;seg37|=0x00;seg38|=0x10;seg39|=0x28;//数字 7
                break;
            case 8:seg36&=0xF7;seg37&=0xE7;seg38&=0xE7;seg39&=0xD7;seg36|=0x08;seg37|=0x18;seg38|=0x18;seg39|=0x28;//数字 8
                break;
            case 9:seg36&=0xF7;seg37&=0xE7;seg38&=0xE7;seg39&=0xD7;seg36|=0x00;seg37|=0x18;seg38|=0x18;seg39|=0x28;//数字 9
                break;
            default:
                break;
        }
        }
        else
        {
seg36&=0xF7;seg37&=0xE7;seg38&=0xE7;seg39&=0xD7;seg36|=0x08;seg37|=0x18;seg38|=0x
10;seg39|=0x28;//数字 0
        }

        /***********rel_Temp 个位***********/
        Val = dis_rel_temp % 10;//取出个位
        if(Val > 4)//大于 4 时
        {
                seg39&=0xEF;seg40&=0xCF;seg39|=0x10;seg40|=0x20;//显示 5
        }
        else//小于 4 时
        {
                seg39&=0xEF;seg40&=0xCF;seg39|=0x00;seg40|=0x30;//显示 0
        }
        seg40|=0x08;//温度的小数点
        seg31&=0xF3;seg31|=0x0C;//温度图标
        if(Run_Status == 1 && Temp_State == 1 && Run_Flag > 1)//运行时
        {
                seg31&=0xF3;//温度图标消失
        }

        /***********时间计算**********/
        if(Time_unit == 1)
        {
                SH=dis_rel_time/3600/10;//计算十位单位的小时数
                H=dis_rel_time/3600%10;//计算个位单位的小时数
                SM=dis_rel_time%3600/60/10;//计算十分位单位的分钟数
                M=dis_rel_time%3600/60%10;//计算个分位单位的分钟数
        }
        else
        {
                SH=dis_rel_time%3600/60/10;//计算十位单位的分钟数
                H=dis_rel_time%3600/60%10;//计算个位单位的分钟数
                SM=dis_rel_time%60/10;//计算十分位单位的秒钟数
                M=dis_rel_time%60%10;//计算十分位单位的秒钟数
        }
        /***********rel_time 十小时位***********/
        switch(SH%10)
        {
                case 0:seg32&=0xf8;seg33&=0xf0;seg32|=0x05;seg33|=0x0f;//数字 0
                        break;
                case 1:seg32&=0xf8;seg33&=0xf0;seg32|=0x00;seg33|=0x06;//数字 1
                        break;
                case 2:seg32&=0xf8;seg33&=0xf0;seg32|=0x03;seg33|=0x0d;//数字 2
                        break;
                case 3:seg32&=0xf8;seg33&=0xf0;seg32|=0x02;seg33|=0x0f;//数字 3
                break;
            case 4:seg32&=0xf8;seg33&=0xf0;seg32|=0x06;seg33|=0x06;//数字 4
                break;
            case 5:seg32&=0xf8;seg33&=0xf0;seg32|=0x06;seg33|=0x0b;//数字 5
                break;
            case 6:seg32&=0xf8;seg33&=0xf0;seg32|=0x07;seg33|=0x0b;//数字 6
                break;
            case 7:seg32&=0xf8;seg33&=0xf0;seg32|=0x00;seg33|=0x0e;//数字 7
                break;
            case 8:seg32&=0xf8;seg33&=0xf0;seg32|=0x07;seg33|=0x0f;//数字 8
                break;
            case 9:seg32&=0xf8;seg33&=0xf0;seg32|=0x06;seg33|=0x0f;//数字 9
                break;
            default:
                break;
        }

        /***********rel_time 小时位***********/
        switch(H%10)
        {
            case
0:seg34&=0xf8;seg35&=0xf8;seg36&=0xfe;seg34|=0x07;seg35|=0x05;seg36|=0x01;//数字 0
                break;
            case
1:seg34&=0xf8;seg35&=0xf8;seg36&=0xfe;seg34|=0x00;seg35|=0x04;seg36|=0x01;//数字 1
                break;
            case
2:seg34&=0xf8;seg35&=0xf8;seg36&=0xfe;seg34|=0x05;seg35|=0x07;seg36|=0x00;//数字 2
                break;
            case
3:seg34&=0xf8;seg35&=0xf8;seg36&=0xfe;seg34|=0x04;seg35|=0x07;seg36|=0x01;//数字 3
                break;
            case
4:seg34&=0xf8;seg35&=0xf8;seg36&=0xfe;seg34|=0x02;seg35|=0x06;seg36|=0x01;//数字 4
                break;
            case
5:seg34&=0xf8;seg35&=0xf8;seg36&=0xfe;seg34|=0x06;seg35|=0x03;seg36|=0x01;//数字 5
                break;
            case
6:seg34&=0xf8;seg35&=0xf8;seg36&=0xfe;seg34|=0x07;seg35|=0x03;seg36|=0x01;//数字 6
                break;
            case
7:seg34&=0xf8;seg35&=0xf8;seg36&=0xfe;seg34|=0x04;seg35|=0x04;seg36|=0x01;//数字 7
                break;
            case
8:seg34&=0xf8;seg35&=0xf8;seg36&=0xfe;seg34|=0x07;seg35|=0x07;seg36|=0x01;//数字 8
                break;
            case
9:seg34&=0xf8;seg35&=0xf8;seg36&=0xfe;seg34|=0x06;seg35|=0x07;seg36|=0x01;//数字 9
                break;
            default:
                  break;
      }

      /***********rel_time 十分位***********/
      switch(SM%10)
      {
            case
0:seg36&=0xfb;seg37&=0xf8;seg38&=0xf8;seg36|=0x04;seg37|=0x05;seg38|=0x07;//数字 0
                  break;
            case
1:seg36&=0xfb;seg37&=0xf8;seg38&=0xf8;seg36|=0x00;seg37|=0x00;seg38|=0x06;//数字 1
                  break;
            case
2:seg36&=0xfb;seg37&=0xf8;seg38&=0xf8;seg36|=0x00;seg37|=0x07;seg38|=0x05;//数字 2
                  break;
            case
3:seg36&=0xfb;seg37&=0xf8;seg38&=0xf8;seg36|=0x00;seg37|=0x06;seg38|=0x07;//数字 3
                  break;
            case
4:seg36&=0xfb;seg37&=0xf8;seg38&=0xf8;seg36|=0x04;seg37|=0x02;seg38|=0x06;//数字 4
                  break;
            case
5:seg36&=0xfb;seg37&=0xf8;seg38&=0xf8;seg36|=0x04;seg37|=0x06;seg38|=0x03;//数字 5
                  break;
            case
6:seg36&=0xfb;seg37&=0xf8;seg38&=0xf8;seg36|=0x04;seg37|=0x07;seg38|=0x03;//数字 6
                  break;
            case
7:seg36&=0xfb;seg37&=0xf8;seg38&=0xf8;seg36|=0x00;seg37|=0x04;seg38|=0x06;//数字 7
                  break;
            case
8:seg36&=0xfb;seg37&=0xf8;seg38&=0xf8;seg36|=0x04;seg37|=0x07;seg38|=0x07;//数字 8
                  break;
            case
9:seg36&=0xfb;seg37&=0xf8;seg38&=0xf8;seg36|=0x04;seg37|=0x06;seg38|=0x07;//数字 9
                  break;
            default:
                  break;
      }

      /***********rel_time 分位***********/
      switch(M%10)
      {
            case
0:seg31&=0xfe;seg39&=0xf8;seg40&=0xf8;seg31|=0x01;seg39|=0x07;seg40|=0x05;//数字 0
                  break;
            case
1:seg31&=0xfe;seg39&=0xf8;seg40&=0xf8;seg31|=0x01;seg39|=0x00;seg40|=0x04;//数字 1
                  break;
```

```
        case
2:seg31&=0xfe;seg39&=0xf8;seg40&=0xf8;seg31|=0x00;seg39|=0x05;seg40|=0x07;//数字 2
            break;
        case
3:seg31&=0xfe;seg39&=0xf8;seg40&=0xf8;seg31|=0x01;seg39|=0x04;seg40|=0x07;//数字 3
            break;
        case
4:seg31&=0xfe;seg39&=0xf8;seg40&=0xf8;seg31|=0x01;seg39|=0x02;seg40|=0x06;//数字 4
            break;
        case
5:seg31&=0xfe;seg39&=0xf8;seg40&=0xf8;seg31|=0x01;seg39|=0x06;seg40|=0x03;//数字 5
            break;
        case
6:seg31&=0xfe;seg39&=0xf8;seg40&=0xf8;seg31|=0x01;seg39|=0x07;seg40|=0x03;//数字 6
            break;
        case
7:seg31&=0xfe;seg39&=0xf8;seg40&=0xf8;seg31|=0x01;seg39|=0x04;seg40|=0x04;//数字 7
            break;
        case
8:seg31&=0xfe;seg39&=0xf8;seg40&=0xf8;seg31|=0x01;seg39|=0x07;seg40|=0x07;//数字 8
            break;
        case
9:seg31&=0xfe;seg39&=0xf8;seg40&=0xf8;seg31|=0x01;seg39|=0x06;seg40|=0x07;//数字 9
            break;
        default:
            break;
    }
    seg31&=0xFD;seg31|=0x02;//时间图标
    seg36|=0x02;//时间冒号
    if(Run_Status == 1 && Run_Flag > 1 && DownTime_Over.L0 == 0 && RelTime_State.L0
== 1)//运行时
    {
        seg36&=0xfd;//时间冒号消失
        seg31&=0xFD;//时间图标消失
    }
    if(RelTime_State.L0 == 0)//没有设置时间
    {
        //实际时间显示“----”

seg31&=0xfe;seg32&=0xf8;seg32|=0x02;seg33&=0xf0;seg34&=0xf8;seg35&=0xf8;seg35|=0x02
;seg36&=0xf8;seg37&=0xf8;seg37|=0x02;seg38&=0xf8;seg39&=0xf8;seg40&=0xf8;seg40|=0x0
2;
    }
    #if(Temp_TYPE == 0)//设置成四联普通款时

seg31&=0xF3;seg32&=0x87;seg33&=0x8F;seg34&=0xC7;seg35&=0xE7;seg36&=0xC7;seg37&
=0xE7;seg38&=0xE7;seg39&=0xC7;seg40&=0xC7;//隐藏温度的图标和数码管
    #endif
    Write_Addr_Dat_N(60, seg31,1);
    Write_Addr_Dat_N(62, seg32,1);
```

```
      Write_Addr_Dat_N(64, seg33,1);
      Write_Addr_Dat_N(66, seg34,1);
      Write_Addr_Dat_N(68, seg35,1);
      Write_Addr_Dat_N(70, seg36,1);
      Write_Addr_Dat_N(72, seg37,1);
      Write_Addr_Dat_N(74, seg38,1);
      Write_Addr_Dat_N(76, seg39,1);
      Write_Addr_Dat_N(78, seg40,1);
}
```