

VM6100 软件源程序

VM6100 软件 V1.0

```

#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 0.2s 时
        {
            Beep_Flash--;//次数--
            BT = 0;//周期清零
        }
    }
}

#include "Drv_TM1640.h"

/*
*****
* 函数原型: void TM1640_Start(void)

```

```

* 功    能：TM1640 开始函数
*****
*/
void TM1640_Start(void)
{
    TM1640_SCL_H;
    TM1640_SDA_H;
    Delay_us(10);
    TM1640_SDA_L;
    Delay_us(10);
    TM1640_SCL_L;
    Delay_us(10);
}

/*
*****
* 函数原型：void TM1640_Stop(void)
* 功    能：TM1640 结束函数
*****
*/
void TM1640_Stop(void)
{
    TM1640_SDA_L;
    TM1640_SCL_H;
    Delay_us(10);
    TM1640_SDA_H;
    Delay_us(10);
}

/*
*****
* 函数原型：void TM1640_Write_Byte(uint8_t date)
* 功    能：写入一个 byte
* 输    入：date：写入的数据
* 参    数：uint8_t date
*****
*/
void TM1640_Write_Byte(uint8_t date)
{
    uint8_t i;
    uint8_t Temp;
    Temp=date;
    TM1640_SCL_L;
    TM1640_SDA_L;
    for(i=0;i<8;i++)
    {
        TM1640_SCL_L;
        Delay_us(2);
        if(Temp&0x01)
        {

```

```

        TM1640_SDA_H;
        Delay_us(10);
    }
    else
    {
        TM1640_SDA_L;
        Delay_us(10);
    }
    TM1640_SCL_H;
    Delay_us(1);
    Temp = Temp >> 1;
}
TM1640_SCL_L;
TM1640_SDA_L;
}

/*
*****
* 函数原型: void TM1640_Write_Com(uint8_t date)
* 功    能: 发送命令字
* 输    入: date: 写入的命令字
* 参    数: uint8_t date
*****
*/
void TM1640_Write_Com(uint8_t date)
{
    TM1640_Start();
    TM1640_Write_Byte(date);
    TM1640_Stop();
}

/*
*****
* 函数原型: void TM1640_Write_DATA(uint8_t add,uint8_t DATA)
* 功    能: 指定地址写入数据
* 输    入: add: 地址值, DATA: 写入的数据
* 参    数: uint8_t add,uint8_t DATA
*****
*/
void TM1640_Write_DATA(uint8_t add,uint8_t DATA)//add 到 15
{
    TM1640_Write_Com(0x44);
    TM1640_Start();
    TM1640_Write_Byte(0xc0|add);
    TM1640_Write_Byte(DATA);
    TM1640_Stop();
}

/*
*****

```

```

* 函数原型: void TM1640_Clr(void)
* 功    能: TM1640 清屏函数
*****
*/
void TM1640_Clr(void)
{
    uint8_t i;
    TM1640_Write_Com(0x40);//连续地址模式
    TM1640_Start();
    TM1640_Write_Byte(0xc0);
    for(i=0;i<16;i++)
        TM1640_Write_Byte(0x00);
    TM1640_Stop();
}

/*
*****
* 函数原型: void TM1640_All(void)
* 功    能: TM1640 全显示函数
*****
*/
void TM1640_All(void)
{
    uint8_t i;
    TM1640_Write_Com(0x40);//连续地址模式
    TM1640_Start();
    TM1640_Write_Byte(0xc0);
    for(i=0;i<16;i++)
        TM1640_Write_Byte(0xFF);
    TM1640_Stop();
}

/*
*****
* 函数原型: void TM1640_Init(void)
* 功    能: TM1640 初始化
*****
*/
void TM1640_Init(void)
{
    TM1640_All();
    TM1640_Write_Com(0x8b);//亮度选择(默认最低) 0x88,0x89,0x8a,0x8b,0x8c,x8d,x8e,8x8f
    x0X(这里 X 代表十六进制的任意值

/*1/16,2/16,4/16,10/16,11/16,12/16,13/16,14/16, 关灯*/
    TM1640_Clr();
}
#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_EC11A.h"

/*****结构体*****/
_EC11A_EC11A[2];//旋钮参数

/*****全局变量声明*****/
float Key_Status;//按键按下标志

/*
*****
* 函数原型: void EC11A_Init(void)
* 功    能: EC11A 初始化定时器
*****
*/
void EC11A_Init(void)
{
    /*****EC11A_1*****/
    EC11A[0].EXTI_Pin = EC1A_Pin;//EC11A 旋钮中断引脚
    EC11A[0].EC11A_Pin = EC1B_Pin;//EC11A 旋钮输入引脚

```

VM6100 软件 V1.0

```

EC11A[0].EC11A_GPIO = EC1B_GPIO_Port;//EC11A 旋钮输入 GPIO 端口

EC11A[0].Key_Pin = KEY1_Pin;//EC11A 按键输入引脚
EC11A[0].Key_GPIO = KEY1_GPIO_Port;//EC11A 按键输入 GPIO 端口

EC11A[0].Tim = &EC11A_Tim_1;//定时器选择
EC11A[0].EC11A_Fast = EC11A_FastSpeed;//判断旋转速度阈值

/*****EC11A_2*****/
EC11A[1].EXTI_Pin = EC2A_Pin;//EC11A 旋钮中断引脚
EC11A[1].EC11A_Pin = EC2B_Pin;//EC11A 旋钮输入引脚
EC11A[1].EC11A_GPIO = EC2B_GPIO_Port;//EC11A 旋钮输入 GPIO 端口

EC11A[1].Key_Pin = KEY2_Pin;//EC11A 按键输入引脚
EC11A[1].Key_GPIO = KEY2_GPIO_Port;//EC11A 按键输入 GPIO 端口

EC11A[1].Tim = &EC11A_Tim_2;//定时器选择
EC11A[1].EC11A_Fast = EC11A_FastSpeed;//判断旋转速度阈值
}

/*
*****
* 函数原型：void EC11A_Speed(float dT)
* 功    能：EC11A 旋钮速度计算
*****
*/
void EC11A_Speed(float dT)
{
    /*****EC11A_1*****/
    EC11A[0].EC11A_Speed = EC11A[0].EC11A_Cnt*60/20;//一秒检测一次。转一圈 20 个反
    馈，一分钟的速度
    EC11A[0].EC11A_Cnt = 0;//将检测到的计数清零

    /*****EC11A_2*****/
    EC11A[1].EC11A_Speed = EC11A[1].EC11A_Cnt*60/20;//一秒检测一次。转一圈 20 个反
    馈，一分钟的速度
    EC11A[1].EC11A_Cnt = 0;//将检测到的计数清零
}

/*
*****
* 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(EC11A[0].EC11A_Knob)//旋钮 0 旋转
        EC11A[0].EC11A_Knob -= dT;//倒计时

```



```

    if(EC11A[1].EC11A_Knob)//旋钮 1 旋转
        EC11A[1].EC11A_Knob -= dT;//倒计时
}

/*
*****
* 函数原型: void EC11AKey_Scan(float dT)
* 功    能: EC11A 按键扫描
*****
*/
void EC11AKey_Scan(float dT)
{
    /*****EC11A_1*****/
    if(HAL_GPIO_ReadPin(EC11A[0].Key_GPIO,EC11A[0].Key_Pin) == KEY_DOWN)//按下
    按键
    {
        if(EC11A[0].LongPress == 0)//没有长按过
        {
            EC11A[0].Key_Cnt += dT;//按下时间++
            EC11A[0].Key_Flag = 1;//按键按下标志置一
        }
    }
    if(EC11A[0].Key_Flag == 1)//按键被按下
    {
        if(HAL_GPIO_ReadPin(EC11A[0].Key_GPIO,EC11A[0].Key_Pin) == KEY_UP)//抬
        起按键
        {
            if(EC11A[0].Key_Cnt > 0.1 && EC11A[0].Key_Cnt < 1.5)//小于 1.5S 是单击
            {
                if(sys.Run_Status == 0)
                {
                    sys.Run_Status = 1;//启动
                    Speed.ADDMode = 0;//进入判断速度显示处理
                    Speed_Val.Integral = 27;//电机起步的 PWM
                    SpeedSet_Flag=TimeSet_Flag=1;//进入设置
                    Speed_Twinkle_Time = Time_Twinkle_Time = 0;//关闭闪烁
                }
                else
                {
                    Speed.ADDMode = 2;//进入降速显示
                    sys.Motor_Stop = 1;//检测电机
                    SpeedSet_Flag=TimeSet_Flag=1;//进入设置
                    Speed_Twinkle_Time = Time_Twinkle_Time = 0;//关闭闪烁
                }
                Beep_Time = 0.1;//蜂鸣器响 0.1S
            }
            EC11A[0].Key_Flag = 0;//按键事件结束，等待下一次按下
            EC11A[0].LongPress = 0;//长按标志清零
            EC11A[0].Key_Cnt = 0;//按钮计数清零
        }
    }
}

```

if(EC11A[0].Key_Cnt > 1.5 && EC11A[0].Key_Cnt < 3)//按键时间大于 1.5S 小于 3S
表示长按

```
{
    if(EC11A[0].LongPress == 0)//如果没有一直一直长按着
    {
        EC11A[0].LongPress = 1;//长按标志置一
    }
}
```

/*****EC11A_2*****/

if(HAL_GPIO_ReadPin(EC11A[1].Key_GPIO,EC11A[1].Key_Pin) == KEY_DOWN)//按下
按键

```
{
    if(EC11A[1].LongPress == 0)//没有长按过
    {
        EC11A[1].Key_Cnt += dT;//按下时间++
        EC11A[1].Key_Flag = 1;//按键按下标志置一
    }
}
if(EC11A[1].Key_Flag == 1)//按键被按下
{
```

if(HAL_GPIO_ReadPin(EC11A[1].Key_GPIO,EC11A[1].Key_Pin) == KEY_UP)//抬
起按键

```
{
    if(EC11A[1].Key_Cnt > 0.1 && EC11A[1].Key_Cnt < 1.5)//小于 1.5S 是单击
    {
        if(sys.Run_Status == 0)
        {
            sys.Run_Status = 1;//启动
            Speed.ADDMode = 0;//进入判断速度显示处理
            Speed_Val.Integral = 27;//电机起步的 PWM
            SpeedSet_Flag=TimeSet_Flag=1;//进入设置
            Speed_Twinkle_Time = Time_Twinkle_Time = 0;//关闭闪烁
        }
        else
        {
            Speed.ADDMode = 2;//进入降速显示
            sys.Motor_Stop = 1;//检测电机
            SpeedSet_Flag=TimeSet_Flag=1;//进入设置
            Speed_Twinkle_Time = Time_Twinkle_Time = 0;//关闭闪烁
        }
        Beep_Time = 0.1;//蜂鸣器响 0.1S
    }
    EC11A[1].Key_Flag = 0;//按键事件结束，等待下一次按下
    EC11A[1].LongPress = 0;//长按标志清零
    EC11A[1].Key_Cnt = 0;//按钮计数清零
}
```

if(EC11A[1].Key_Cnt > 1.5 && EC11A[1].Key_Cnt < 3)//按键时间大于 1.5S 小于 3S

表示长按

```

{
    if(EC11A[1].LongPress == 0)//如果没有一直一直长按着
    {
        EC11A[1].LongPress = 1;//长按标志置一
    }
}
}

/*
*****
* 函数原型: void HAL_GPIO_EXTI_Callback(uint16_t GPIO_Pin)
* 功    能: 外部中断
*****
*/
void HAL_GPIO_EXTI_Callback(uint16_t GPIO_Pin)
{
    UNUSED(GPIO_Pin);

    /*****EC11A_1*****/
    if(GPIO_Pin == EC11A[0].EXTI_Pin)//A 上升沿触发外部中断
    {
        HAL_TIM_Base_Start_IT(EC11A[0].Tim);//开始定时器
        while(EC11A[0].TIM_Cnt <= 2)//定时器一个周期 1ms, 计时 2ms 内看看 A 有没有电
        跳变
        {
            if(GPIO_Pin == EC11A[0].EXTI_Pin)//在 2ms 内, 检测到电平变化
            {
                HAL_TIM_Base_Stop_IT(EC11A[0].Tim);//停止定时器
                EC11A[0].TIM_Cnt = 0;//清除 TIM 计数
                EC11A[0].EC11A_Cnt++; //旋钮计数
                EC11A[0].EC11A_Knob = 2.0f;//在旋转旋钮时
                if(HAL_GPIO_ReadPin(EC11A[0].EC11A_GPIO, EC11A[0].EC11A_Pin) ==
                0)//加
                {
                    if(EC11A[0].EC11A_Speed < EC11A[0].EC11A_Fast)//如果慢慢旋转
                        Speed.Set += 100;//转速加 100
                    else
                        Speed.Set += 200;//转速加 200
                    Speed.Set = (Speed.Set > Speed_MAX) ? Speed_MAX : Speed.Set;//速
                    度不超过最快值
                    Speed_Twinkle_Time = 2.0f;//速度闪烁 2S
                    break;
                }
            }
            else if(HAL_GPIO_ReadPin(EC11A[0].EC11A_GPIO, EC11A[0].EC11A_Pin)
            == 1)//减
            {
                if(EC11A[0].EC11A_Speed < EC11A[0].EC11A_Fast)//如果慢慢旋转

```

VM6100 软件 V1.0

```

        Speed.Set -= 100;//转速减 100
    else
        Speed.Set -= 200;//转速减 200
    Speed.Set = (Speed.Set < Speed_MIN) ? Speed_MIN : Speed.Set;//速度
    不小于最慢值

    Speed_Twinkle_Time = 2.0f;//速度闪烁 2S
    break;
}
break;
}
}
HAL_TIM_Base_Stop_IT(EC11A[0].Tim);//停止定时器
EC11A[0].TIM_Cnt = 0;//清除 TIM 计数
}

/*****EC11A_2*****/
if(GPIO_Pin == EC11A[1].EXTI_Pin)//A 上升沿触发外部中断
{
    HAL_TIM_Base_Start_IT(EC11A[1].Tim);//开始定时器
    while(EC11A[1].TIM_Cnt <= 2)//定时器一个周期 1ms, 计时 2ms 内看看 A 有没有电
    跳变
    {
        if(GPIO_Pin == EC11A[1].EXTI_Pin)//在 2ms 内, 检测到电平变化
        {
            HAL_TIM_Base_Stop_IT(EC11A[1].Tim);//停止定时器
            EC11A[1].TIM_Cnt = 0;//清除 TIM 计数
            EC11A[1].EC11A_Cnt++;//旋钮计数
            EC11A[1].EC11A_Knob = 2.0f;//在旋转旋钮时
            if(HAL_GPIO_ReadPin(EC11A[1].EC11A_GPIO, EC11A[1].EC11A_Pin) ==
            0)//加
            {
                if(EC11A[1].EC11A_Speed < EC11A[1].EC11A_Fast)//如果慢慢旋转
                    Time.Set += 60;//设定时间加 60S
                else
                    Time.Set += 600;//设定时间加 600S
                Time.Set = (Time.Set > Time_MAX) ? Time_MAX : Time.Set;//时间最
                多设定 24 小时

                Time_Twinkle_Time = 2.0f;//时间闪烁 2S
                break;
            }
            else if(HAL_GPIO_ReadPin(EC11A[1].EC11A_GPIO, EC11A[1].EC11A_Pin)
            == 1)//减
            {
                if(EC11A[1].EC11A_Speed < EC11A[1].EC11A_Fast)//如果慢慢旋转
                    Time.Set -= 60;//设定时间减 60S
                else
                    Time.Set -= 600;//设定时间减 600S
                Time.Set = (Time.Set < Time_MIN) ? 0 : Time.Set;//时间小于 1 分钟不
                设定, 设定时间为 0 (常动)
                Time_Twinkle_Time = 2.0f;//时间闪烁 2S
            }
        }
    }
}

```

```

        break;
    }
    break;
}
}
HAL_TIM_Base_Stop_IT(EC11A[1].Tim);//停止定时器
EC11A[1].TIM_Cnt = 0;//清除 TIM 计数
}
}

/*
*****
* 函数原型: void HAL_TIM_PeriodElapsedCallback(TIM_HandleTypeDef *htim)
* 功    能: 定时器计数中断
*****
*/
void HAL_TIM_PeriodElapsedCallback(TIM_HandleTypeDef *htim)
{
    if(htim->Instance == EC11A_Tim_1.Instance)
    {
        EC11A[0].TIM_Cnt++;
    }

    if(htim->Instance == EC11A_Tim_2.Instance)
    {
        EC11A[1].TIM_Cnt++;
    }
}
#include "Show.h"

/*****全局变量声明*****/
float Speed_Twinkle_Time,Time_Twinkle_Time;//速度闪烁时间和时间闪烁时间

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

/*
*****
* 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 Speed_T,Time_T;
    if(Speed_Twinkle_Time && !EC11A[0].EC11A_Knob)//速度闪烁和没有操作按键时

```

```

{
    Speed_T += dT;
    if(Speed_T >= 0.5f)
    {
        Speed_Twinkle_Time -= 0.5; //闪烁计时
        Speed_ShowFlag = ~Speed_ShowFlag; //速度闪烁
        if(Speed_Twinkle_Time == 0) //如果闪烁结束
        {
            SpeedSet_Flag = 1; //进入速度设定
            Beep_Time = 0.1; //蜂鸣器响 0.1S
        }
        Speed_T = 0;
    }
}
else
{
    Speed_ShowFlag = 0; //常亮
    Speed_T = 0;
}

if(Time_Twinkle_Time && !EC11A[1].EC11A_Knob) //时间闪烁和没有操作按键时
{
    Time_T += dT;
    if(Time_T >= 0.5f)
    {
        Time_Twinkle_Time -= 0.5; //闪烁计时
        Time_ShowFlag = ~Time_ShowFlag; //时间闪烁
        if(Time_Twinkle_Time == 0) //如果闪烁结束
        {
            TimeSet_Flag = 1; //进入时间设定
            Beep_Time = 0.1; //蜂鸣器响 0.1S
        }
        Time_T = 0;
    }
}
else
{
    Time_ShowFlag = 0; //常亮
    Time_T = 0;
}
}

/*
*****
* 函数原型: void Twinkle(float dT)
* 功    能: 闪烁函数
*****
*/
void Twinkle(float dT)
{

```

```

    Check_ShowFlag(dT);//闪烁检测
}

/*
*****
* 函数原型： void Display_Screen(int16_t dis_set_speed,int16_t dis_rel_speed,int32_t
dis_set_time,int32_t dis_rel_time)
* 功 能： 显示屏幕内容
* 输 入： dis_set_speed: 设定转速， dis_rel_speed: 实际转速， dis_set_time: 设定时间，
dis_rel_time: 实际时间
* 参 数： int16_t dis_set_speed,int16_t dis_rel_speed,int32_t dis_set_time,int32_t
dis_rel_time
*****
*/
void Display_Screen(int16_t dis_set_speed,int16_t dis_rel_speed,int32_t dis_set_time,int32_t
dis_rel_time)
{
    uint8_t seg[16] = {0};
//    uint8_t
seg0,seg1,seg2,seg3,seg4,seg5,seg6,seg7,seg8,seg9,seg10,seg11,seg12,seg13,seg14,seg15;
//
    seg0=0;seg1=0;seg2=0;seg3=0;seg4=0;seg5=0;seg6=0;seg7=0;seg8=0;seg9=0;seg10=0;seg1
1=0;seg12=0;seg13=0;seg14=0;seg15=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

    /*****dis_set_speed 千位*****/
    if(!Speed_ShowFlag)//闪烁显示
    {
        if(dis_set_speed > 999)//大于 999 时
        {
            Val=dis_set_speed/1000;//取出千位
            if(dis_set_speed > 9999)//大于 9999 时
                Val=Val%10;//取出千位
            switch(Val)
            {
                case
0:seg[0]&=0xFE;seg[1]&=0xFE;seg[3]&=0xFE;seg[5]&=0xFE;seg[6]&=0xFE;seg[7]&=0xFE;

seg[0]=0x01;seg[1]=0x01;seg[3]=0x01;seg[5]=0x01;seg[6]=0x01;seg[7]=0x01;//数字 0
                break;
                case 1:seg[3]&=0xFE;seg[5]&=0xFE;
                    seg[3]=0x01;seg[5]=0x01;//数字 1
                break;
                case
2:seg[0]&=0xFE;seg[1]&=0xFE;seg[4]&=0xFE;seg[5]&=0xFE;seg[7]&=0xFE;

seg[0]=0x01;seg[1]=0x01;seg[4]=0x01;seg[5]=0x01;seg[7]=0x01;//数字 2
                break;
                case

```

VM6100 软件 V1.0

```

3:seg[1]&=0xFE;seg[3]&=0xFE;seg[4]&=0xFE;seg[5]&=0xFE;seg[7]&=0xFE;

seg[1]=0x01;seg[3]=0x01;seg[4]=0x01;seg[5]=0x01;seg[7]=0x01;//数字 3
    break;
    case 4:seg[3]&=0xFE;seg[4]&=0xFE;seg[5]&=0xFE;seg[6]&=0xFE;
        seg[3]=0x01;seg[4]=0x01;seg[5]=0x01;seg[6]=0x01;//数字 4
        break;
    case
5:seg[1]&=0xFE;seg[3]&=0xFE;seg[4]&=0xFE;seg[6]&=0xFE;seg[7]&=0xFE;

seg[1]=0x01;seg[3]=0x01;seg[4]=0x01;seg[6]=0x01;seg[7]=0x01;//数字 5
    break;
    case
6:seg[0]&=0xFE;seg[1]&=0xFE;seg[3]&=0xFE;seg[4]&=0xFE;seg[6]&=0xFE;seg[7]&=0xFE;

seg[0]=0x01;seg[1]=0x01;seg[3]=0x01;seg[4]=0x01;seg[6]=0x01;seg[7]=0x01;//数字 6
    break;
    case 7:seg[3]&=0xFE;seg[5]&=0xFE;seg[7]&=0xFE;
        seg[3]=0x01;seg[5]=0x01;seg[7]=0x01;//数字 7
        break;
    case
8:seg[0]&=0xFE;seg[1]&=0xFE;seg[3]&=0xFE;seg[4]&=0xFE;seg[5]&=0xFE;seg[6]&=0xFE;seg[7]&=0xFE;

seg[0]=0x01;seg[1]=0x01;seg[3]=0x01;seg[4]=0x01;seg[5]=0x01;seg[6]=0x01;seg[7]=0x01;//
数字 8
    break;
    case
9:seg[1]&=0xFE;seg[3]&=0xFE;seg[4]&=0xFE;seg[5]&=0xFE;seg[6]&=0xFE;seg[7]&=0xFE;

seg[1]=0x01;seg[3]=0x01;seg[4]=0x01;seg[5]=0x01;seg[6]=0x01;seg[7]=0x01;//数字 9
    break;
    default:
        break;
    }
}
else
{

seg[0]&=0xFE;seg[1]&=0xFE;seg[3]&=0xFE;seg[4]&=0xFE;seg[5]&=0xFE;seg[6]&=0xFE;seg[7]&=0xFE;

seg[0]=0x00;seg[1]=0x00;seg[3]=0x00;seg[4]=0x00;seg[5]=0x00;seg[6]=0x00;seg[7]=0x00;//
不显示
    }

    /*****dis_set_speed 百位*****/
    if(dis_set_speed > 99)//大于 99 时
    {
        Val=dis_set_speed/100;//取出百位
    }

```

```

if(dis_set_speed > 999)//大于 999 时
    Val=Val%10;//取出百位
switch(Val)
{
    case
0:seg[0]&=0xFD;seg[1]&=0xFD;seg[3]&=0xFD;seg[5]&=0xFD;seg[6]&=0xFD;seg[7]&=0xFD;

seg[0]=0x02;seg[1]=0x02;seg[3]=0x02;seg[5]=0x02;seg[6]=0x02;seg[7]=0x02;//数字 0
    break;
    case 1:seg[3]&=0xFD;seg[5]&=0xFD;
        seg[3]=0x02;seg[5]=0x02;//数字 1
    break;
    case
2:seg[0]&=0xFD;seg[1]&=0xFD;seg[4]&=0xFD;seg[5]&=0xFD;seg[7]&=0xFD;

seg[0]=0x02;seg[1]=0x02;seg[4]=0x02;seg[5]=0x02;seg[7]=0x02;//数字 2
    break;
    case
3:seg[1]&=0xFD;seg[3]&=0xFD;seg[4]&=0xFD;seg[5]&=0xFD;seg[7]&=0xFD;

seg[1]=0x02;seg[3]=0x02;seg[4]=0x02;seg[5]=0x02;seg[7]=0x02;//数字 3
    break;
    case 4:seg[3]&=0xFD;seg[4]&=0xFD;seg[5]&=0xFD;seg[6]&=0xFD;
        seg[3]=0x02;seg[4]=0x02;seg[5]=0x02;seg[6]=0x02;//数字 4
    break;
    case
5:seg[1]&=0xFD;seg[3]&=0xFD;seg[4]&=0xFD;seg[6]&=0xFD;seg[7]&=0xFD;

seg[1]=0x02;seg[3]=0x02;seg[4]=0x02;seg[6]=0x02;seg[7]=0x02;//数字 5
    break;
    case
6:seg[0]&=0xFD;seg[1]&=0xFD;seg[3]&=0xFD;seg[4]&=0xFD;seg[6]&=0xFD;seg[7]&=0xFD;

seg[0]=0x02;seg[1]=0x02;seg[3]=0x02;seg[4]=0x02;seg[6]=0x02;seg[7]=0x02;//数字 6
    break;
    case 7:seg[3]&=0xFD;seg[5]&=0xFD;seg[7]&=0xFD;
        seg[3]=0x02;seg[5]=0x02;seg[7]=0x02;//数字 7
    break;
    case
8:seg[0]&=0xFD;seg[1]&=0xFD;seg[3]&=0xFD;seg[4]&=0xFD;seg[5]&=0xFD;seg[6]&=0xFD;s
eg[7]&=0xFD;

seg[0]=0x02;seg[1]=0x02;seg[3]=0x02;seg[4]=0x02;seg[5]=0x02;seg[6]=0x02;seg[7]=0x02;//
数字 8
    break;
    case
9:seg[1]&=0xFD;seg[3]&=0xFD;seg[4]&=0xFD;seg[5]&=0xFD;seg[6]&=0xFD;seg[7]&=0xFD;

seg[1]=0x02;seg[3]=0x02;seg[4]=0x02;seg[5]=0x02;seg[6]=0x02;seg[7]=0x02;//数字 9
    break;

```

```

        default:
            break;
    }
}
else
{

seg[0]&=0xFD;seg[1]&=0xFD;seg[3]&=0xFD;seg[4]&=0xFD;seg[5]&=0xFD;seg[6]&=0xFD;seg[7]&=0xFD;

seg[0]=0x00;seg[1]=0x00;seg[3]=0x00;seg[4]=0x00;seg[5]=0x00;seg[6]=0x00;seg[7]=0x00;//
不显示
}

/*****dis_set_speed 十位*****/
if(dis_set_speed > 9)//大于 9 时
{
    Val=dis_set_speed/10;//取出十位
    if(dis_set_speed > 99)//大于 99 时
        Val=Val%10;//取出十位
    switch(Val)
    {
        case
0:seg[0]&=0xFB;seg[1]&=0xFB;seg[3]&=0xFB;seg[5]&=0xFB;seg[6]&=0xFB;seg[7]&=0xFB;

seg[0]=0x04;seg[1]=0x04;seg[3]=0x04;seg[5]=0x04;seg[6]=0x04;seg[7]=0x04;//数字 0
        break;
        case 1:seg[3]&=0xFB;seg[5]&=0xFB;
            seg[3]=0x04;seg[5]=0x04;//数字 1
        break;
        case
2:seg[0]&=0xFB;seg[1]&=0xFB;seg[4]&=0xFB;seg[5]&=0xFB;seg[7]&=0xFB;

seg[0]=0x04;seg[1]=0x04;seg[4]=0x04;seg[5]=0x04;seg[7]=0x04;//数字 2
        break;
        case
3:seg[1]&=0xFB;seg[3]&=0xFB;seg[4]&=0xFB;seg[5]&=0xFB;seg[7]&=0xFB;

seg[1]=0x04;seg[3]=0x04;seg[4]=0x04;seg[5]=0x04;seg[7]=0x04;//数字 3
        break;
        case 4:seg[3]&=0xFB;seg[4]&=0xFB;seg[5]&=0xFB;seg[6]&=0xFB;
            seg[3]=0x04;seg[4]=0x04;seg[5]=0x04;seg[6]=0x04;//数字 4
        break;
        case
5:seg[1]&=0xFB;seg[3]&=0xFB;seg[4]&=0xFB;seg[6]&=0xFB;seg[7]&=0xFB;

seg[1]=0x04;seg[3]=0x04;seg[4]=0x04;seg[6]=0x04;seg[7]=0x04;//数字 5
        break;
        case
6:seg[0]&=0xFB;seg[1]&=0xFB;seg[3]&=0xFB;seg[4]&=0xFB;seg[6]&=0xFB;seg[7]&=0xFB;

```

```

seg[0]=0x04;seg[1]=0x04;seg[3]=0x04;seg[4]=0x04;seg[6]=0x04;seg[7]=0x04;//数字 6
    break;
case 7:seg[3]&=0xFB;seg[5]&=0xFB;seg[7]&=0xFB;
    seg[3]=0x04;seg[5]=0x04;seg[7]=0x04;//数字 7
    break;
case
8:seg[0]&=0xFB;seg[1]&=0xFB;seg[3]&=0xFB;seg[4]&=0xFB;seg[5]&=0xFB;seg[6]&=0xFB;s
eg[7]&=0xFB;

seg[0]=0x04;seg[1]=0x04;seg[3]=0x04;seg[4]=0x04;seg[5]=0x04;seg[6]=0x04;seg[7]=0x04;//
数字 8
    break;
case
9:seg[1]&=0xFB;seg[3]&=0xFB;seg[4]&=0xFB;seg[5]&=0xFB;seg[6]&=0xFB;seg[7]&=0xFB;

seg[1]=0x04;seg[3]=0x04;seg[4]=0x04;seg[5]=0x04;seg[6]=0x04;seg[7]=0x04;//数字 9
    break;
default:
    break;
    }
}
else
{

seg[0]&=0xFB;seg[1]&=0xFB;seg[3]&=0xFB;seg[4]&=0xFB;seg[5]&=0xFB;seg[6]&=0xFB;seg
[7]&=0xFB;

seg[0]=0x00;seg[1]=0x00;seg[3]=0x00;seg[4]=0x00;seg[5]=0x00;seg[6]=0x00;seg[7]=0x00;//
不显示
    }

    /*****dis_set_speed 个位*****/
    Val=dis_set_speed%10;//取出个位
    switch(Val)
    {
        case
0:seg[0]&=0xF7;seg[1]&=0xF7;seg[3]&=0xF7;seg[5]&=0xF7;seg[6]&=0xF7;seg[7]&=0xF7;

seg[0]=0x08;seg[1]=0x08;seg[3]=0x08;seg[5]=0x08;seg[6]=0x08;seg[7]=0x08;//数字 0
        break;
        case 1:seg[3]&=0xF7;seg[5]&=0xF7;
            seg[3]=0x08;seg[5]=0x08;//数字 1
            break;
        case
2:seg[0]&=0xF7;seg[1]&=0xF7;seg[4]&=0xF7;seg[5]&=0xF7;seg[7]&=0xF7;
            seg[0]=0x08;seg[1]=0x08;seg[4]=0x08;seg[5]=0x08;seg[7]=0x08;// 数
字 2
            break;
        case

```

VM6100 软件 V1.0

```

3:seg[1]&=0xF7;seg[3]&=0xF7;seg[4]&=0xF7;seg[5]&=0xF7;seg[7]&=0xF7;
    seg[1]=0x08;seg[3]=0x08;seg[4]=0x08;seg[5]=0x08;seg[7]=0x08;// 数字 3
        break;
    case 4:seg[3]&=0xF7;seg[4]&=0xF7;seg[5]&=0xF7;seg[6]&=0xF7;
        seg[3]=0x08;seg[4]=0x08;seg[5]=0x08;seg[6]=0x08;//数字 4
        break;
    case
5:seg[1]&=0xF7;seg[3]&=0xF7;seg[4]&=0xF7;seg[6]&=0xF7;seg[7]&=0xF7;
    seg[1]=0x08;seg[3]=0x08;seg[4]=0x08;seg[6]=0x08;seg[7]=0x08;// 数字 5
        break;
    case
6:seg[0]&=0xF7;seg[1]&=0xF7;seg[3]&=0xF7;seg[4]&=0xF7;seg[6]&=0xF7;seg[7]&=0xF7;
    seg[0]=0x08;seg[1]=0x08;seg[3]=0x08;seg[4]=0x08;seg[6]=0x08;seg[7]=0x08;//数字 6
        break;
    case 7:seg[3]&=0xF7;seg[5]&=0xF7;seg[7]&=0xF7;
        seg[3]=0x08;seg[5]=0x08;seg[7]=0x08;//数字 7
        break;
    case
8:seg[0]&=0xF7;seg[1]&=0xF7;seg[3]&=0xF7;seg[4]&=0xF7;seg[5]&=0xF7;seg[6]&=0xF7;seg[7]&=0xF7;
    seg[0]=0x08;seg[1]=0x08;seg[3]=0x08;seg[4]=0x08;seg[5]=0x08;seg[6]=0x08;seg[7]=0x08;//
    数字 8
        break;
    case
9:seg[1]&=0xF7;seg[3]&=0xF7;seg[4]&=0xF7;seg[5]&=0xF7;seg[6]&=0xF7;seg[7]&=0xF7;
    seg[1]=0x08;seg[3]=0x08;seg[4]=0x08;seg[5]=0x08;seg[6]=0x08;seg[7]=0x08;//数字 9
        break;
    default:
        break;
    }
}
else
{

seg[0]&=0xF0;seg[1]&=0xF0;seg[3]&=0xF0;seg[4]&=0xF0;seg[5]&=0xF0;seg[6]&=0xF0;seg[7]
&=0xF0;

seg[0]=0x00;seg[1]=0x00;seg[3]=0x00;seg[4]=0x00;seg[5]=0x00;seg[6]=0x00;seg[7]=0x00;//
不显示
    }

    /*****dis_rel_speed 千位*****/
    if(dis_rel_speed > 999)//大于 999 时
    {
        Val=dis_rel_speed/1000;//取出千位
    }
}

```

```

    if(dis_rel_speed > 9999)//大于 9999 时
        Val=Val%10;//取出千位
    switch(Val)
    {
        case
0:seg[0]&=0xEF;seg[1]&=0xEF;seg[3]&=0xEF;seg[5]&=0xEF;seg[6]&=0xEF;seg[7]&=0xEF;

seg[0]=0x10;seg[1]=0x10;seg[3]=0x10;seg[5]=0x10;seg[6]=0x10;seg[7]=0x10;//数字 0
        break;
        case 1:seg[3]&=0xEF;seg[5]&=0xEF;
            seg[3]=0x10;seg[5]=0x10;//数字 1
        break;
        case
2:seg[0]&=0xEF;seg[1]&=0xEF;seg[4]&=0xEF;seg[5]&=0xEF;seg[7]&=0xEF;
            seg[0]=0x10;seg[1]=0x10;seg[4]=0x10;seg[5]=0x10;seg[7]=0x10;// 数
字 2
        break;
        case
3:seg[1]&=0xEF;seg[3]&=0xEF;seg[4]&=0xEF;seg[5]&=0xEF;seg[7]&=0xEF;
            seg[1]=0x10;seg[3]=0x10;seg[4]=0x10;seg[5]=0x10;seg[7]=0x10;// 数
字 3
        break;
        case 4:seg[3]&=0xEF;seg[4]&=0xEF;seg[5]&=0xEF;seg[6]&=0xEF;
            seg[3]=0x10;seg[4]=0x10;seg[5]=0x10;seg[6]=0x10;//数字 4
        break;
        case
5:seg[1]&=0xEF;seg[3]&=0xEF;seg[4]&=0xEF;seg[6]&=0xEF;seg[7]&=0xEF;
            seg[1]=0x10;seg[3]=0x10;seg[4]=0x10;seg[6]=0x10;seg[7]=0x10;// 数
字 5
        break;
        case
6:seg[0]&=0xEF;seg[1]&=0xEF;seg[3]&=0xEF;seg[4]&=0xEF;seg[6]&=0xEF;seg[7]&=0xEF;

seg[0]=0x10;seg[1]=0x10;seg[3]=0x10;seg[4]=0x10;seg[6]=0x10;seg[7]=0x10;//数字 6
        break;
        case 7:seg[3]&=0xEF;seg[5]&=0xEF;seg[7]&=0xEF;
            seg[3]=0x10;seg[5]=0x10;seg[7]=0x10;//数字 7
        break;
        case
8:seg[0]&=0xEF;seg[1]&=0xEF;seg[3]&=0xEF;seg[4]&=0xEF;seg[5]&=0xEF;seg[6]&=0xEF;se
g[7]&=0xEF;

seg[0]=0x10;seg[1]=0x10;seg[3]=0x10;seg[4]=0x10;seg[5]=0x10;seg[6]=0x10;seg[7]=0x10;//
数字 8
        break;
        case
9:seg[1]&=0xEF;seg[3]&=0xEF;seg[4]&=0xEF;seg[5]&=0xEF;seg[6]&=0xEF;seg[7]&=0xEF;

seg[1]=0x10;seg[3]=0x10;seg[4]=0x10;seg[5]=0x10;seg[6]=0x10;seg[7]=0x10;//数字 9
        break;

```

```

        default:
            break;
    }
}
else
{

seg[0]&=0xEF;seg[1]&=0xEF;seg[3]&=0xEF;seg[4]&=0xEF;seg[5]&=0xEF;seg[6]&=0xEF;seg[
7]&=0xEF;

seg[0]=0x00;seg[1]=0x00;seg[3]=0x00;seg[4]=0x00;seg[5]=0x00;seg[6]=0x00;seg[7]=0x00;//
不显示
}

/*****dis_rel_speed 百位*****/
if(dis_rel_speed > 99)//大于 99 时
{
    Val=dis_rel_speed/100;//取出百位
    if(dis_rel_speed > 999)//大于 999 时
        Val=Val%10;//取出百位
    switch(Val)
    {
        case
0:seg[0]&=0xDF;seg[1]&=0xDF;seg[3]&=0xDF;seg[5]&=0xDF;seg[6]&=0xDF;seg[7]&=0xDF;

seg[0]=0x20;seg[1]=0x20;seg[3]=0x20;seg[5]=0x20;seg[6]=0x20;seg[7]=0x20;//数字 0
        break;
        case 1:seg[3]&=0xDF;seg[5]&=0xDF;
            seg[3]=0x20;seg[5]=0x20;//数字 1
        break;
        case
2:seg[0]&=0xDF;seg[1]&=0xDF;seg[4]&=0xDF;seg[5]&=0xDF;seg[7]&=0xDF;
            seg[0]=0x20;seg[1]=0x20;seg[4]=0x20;seg[5]=0x20;seg[7]=0x20;// 数
字 2
        break;
        case
3:seg[1]&=0xDF;seg[3]&=0xDF;seg[4]&=0xDF;seg[5]&=0xDF;seg[7]&=0xDF;
            seg[1]=0x20;seg[3]=0x20;seg[4]=0x20;seg[5]=0x20;seg[7]=0x20;// 数
字 3
        break;
        case 4:seg[3]&=0xDF;seg[4]&=0xDF;seg[5]&=0xDF;seg[6]&=0xDF;
            seg[3]=0x20;seg[4]=0x20;seg[5]=0x20;seg[6]=0x20;//数字 4
        break;
        case
5:seg[1]&=0xDF;seg[3]&=0xDF;seg[4]&=0xDF;seg[6]&=0xDF;seg[7]&=0xDF;
            seg[1]=0x20;seg[3]=0x20;seg[4]=0x20;seg[6]=0x20;seg[7]=0x20;// 数
字 5
        break;
        case
6:seg[0]&=0xDF;seg[1]&=0xDF;seg[3]&=0xDF;seg[4]&=0xDF;seg[6]&=0xDF;seg[7]&=0xDF;

```

```

seg[0]=0x20;seg[1]=0x20;seg[3]=0x20;seg[4]=0x20;seg[6]=0x20;seg[7]=0x20;//数字 6
    break;
    case 7:seg[3]&=0xDF;seg[5]&=0xDF;seg[7]&=0xDF;
        seg[3]=0x20;seg[5]=0x20;seg[7]=0x20;//数字 7
        break;
    case
8:seg[0]&=0xDF;seg[1]&=0xDF;seg[3]&=0xDF;seg[4]&=0xDF;seg[5]&=0xDF;seg[6]&=0xDF;seg[7]&=0xDF;

seg[0]=0x20;seg[1]=0x20;seg[3]=0x20;seg[4]=0x20;seg[5]=0x20;seg[6]=0x20;seg[7]=0x20;//
数字 8
    break;
    case
9:seg[1]&=0xDF;seg[3]&=0xDF;seg[4]&=0xDF;seg[5]&=0xDF;seg[6]&=0xDF;seg[7]&=0xDF;

seg[1]=0x20;seg[3]=0x20;seg[4]=0x20;seg[5]=0x20;seg[6]=0x20;seg[7]=0x20;//数字 9
    break;
    default:
        break;
}
}
else
{

seg[0]&=0xDF;seg[1]&=0xDF;seg[3]&=0xDF;seg[4]&=0xDF;seg[5]&=0xDF;seg[6]&=0xDF;seg[7]&=0xDF;

seg[0]=0x00;seg[1]=0x00;seg[3]=0x00;seg[4]=0x00;seg[5]=0x00;seg[6]=0x00;seg[7]=0x00;//
不显示
}
/*****dis_rel_speed 十位*****/
if(dis_rel_speed > 9)//大于 9 时
{
    Val=dis_rel_speed/10;//取出十位
    if(dis_rel_speed > 99)//大于 99 时
        Val=Val%10;//取出十位
    switch(Val)
    {
        case
0:seg[0]&=0xBF;seg[1]&=0xBF;seg[3]&=0xBF;seg[5]&=0xBF;seg[6]&=0xBF;seg[7]&=0xBF;

seg[0]=0x40;seg[1]=0x40;seg[3]=0x40;seg[5]=0x40;seg[6]=0x40;seg[7]=0x40;//数字 0
        break;
        case 1:seg[3]&=0xBF;seg[5]&=0xBF;
            seg[3]=0x40;seg[5]=0x40;//数字 1
            break;
        case
2:seg[0]&=0xBF;seg[1]&=0xBF;seg[4]&=0xBF;seg[5]&=0xBF;seg[7]&=0xBF;
            seg[0]=0x40;seg[1]=0x40;seg[4]=0x40;seg[5]=0x40;seg[7]=0x40;// 数

```

字 2

```

        break;
    case
3:seg[1]&=0xBF;seg[3]&=0xBF;seg[4]&=0xBF;seg[5]&=0xBF;seg[7]&=0xBF;
    seg[1]=0x40;seg[3]=0x40;seg[4]=0x40;seg[5]=0x40;seg[7]=0x40;// 数

```

字 3

```

        break;
    case 4:seg[3]&=0xBF;seg[4]&=0xBF;seg[5]&=0xBF;seg[6]&=0xBF;
    seg[3]=0x40;seg[4]=0x40;seg[5]=0x40;seg[6]=0x40;//数字 4
        break;
    case
5:seg[1]&=0xBF;seg[3]&=0xBF;seg[4]&=0xBF;seg[6]&=0xBF;seg[7]&=0xBF;
    seg[1]=0x40;seg[3]=0x40;seg[4]=0x40;seg[6]=0x40;seg[7]=0x40;// 数

```

字 5

```

        break;
    case
6:seg[0]&=0xBF;seg[1]&=0xBF;seg[3]&=0xBF;seg[4]&=0xBF;seg[6]&=0xBF;seg[7]&=0xBF;

seg[0]=0x40;seg[1]=0x40;seg[3]=0x40;seg[4]=0x40;seg[6]=0x40;seg[7]=0x40;//数字 6
        break;
    case 7:seg[3]&=0xBF;seg[5]&=0xBF;seg[7]&=0xBF;
    seg[3]=0x40;seg[5]=0x40;seg[7]=0x40;//数字 7
        break;
    case
8:seg[0]&=0xBF;seg[1]&=0xBF;seg[3]&=0xBF;seg[4]&=0xBF;seg[5]&=0xBF;seg[6]&=0xBF;s
eg[7]&=0xBF;

```

```

seg[0]=0x40;seg[1]=0x40;seg[3]=0x40;seg[4]=0x40;seg[5]=0x40;seg[6]=0x40;seg[7]=0x40;//
数字 8

```

```

        break;
    case
9:seg[1]&=0xBF;seg[3]&=0xBF;seg[4]&=0xBF;seg[5]&=0xBF;seg[6]&=0xBF;seg[7]&=0xBF;

seg[1]=0x40;seg[3]=0x40;seg[4]=0x40;seg[5]=0x40;seg[6]=0x40;seg[7]=0x40;//数字 9
        break;
    default:
        break;
    }
}
else
{

```

```

seg[0]&=0xBF;seg[1]&=0xBF;seg[3]&=0xBF;seg[4]&=0xBF;seg[5]&=0xBF;seg[6]&=0xBF;seg
[7]&=0xBF;

```

```

seg[0]=0x00;seg[1]=0x00;seg[3]=0x00;seg[4]=0x00;seg[5]=0x00;seg[6]=0x00;seg[7]=0x00;//
不显示

```

```

    }
    /*****dis_rel_speed 个位*****/
    Val=dis_rel_speed%10;//取出个位

```



```

switch(Val)
{
    case
0:seg[0]&=0x7F;seg[1]&=0x7F;seg[3]&=0x7F;seg[5]&=0x7F;seg[6]&=0x7F;seg[7]&=0x7F;

seg[0]=0x80;seg[1]=0x80;seg[3]=0x80;seg[5]=0x80;seg[6]=0x80;seg[7]=0x80;//数字 0
        break;
    case 1:seg[3]&=0x7F;seg[5]&=0x7F;
        seg[3]=0x80;seg[5]=0x80;//数字 1
        break;
    case 2:seg[0]&=0x7F;seg[1]&=0x7F;seg[4]&=0x7F;seg[5]&=0x7F;seg[7]&=0x7F;
        seg[0]=0x80;seg[1]=0x80;seg[4]=0x80;seg[5]=0x80;seg[7]=0x80;//数字 2
        break;
    case 3:seg[1]&=0x7F;seg[3]&=0x7F;seg[4]&=0x7F;seg[5]&=0x7F;seg[7]&=0x7F;
        seg[1]=0x80;seg[3]=0x80;seg[4]=0x80;seg[5]=0x80;seg[7]=0x80;//数字 3
        break;
    case 4:seg[3]&=0x7F;seg[4]&=0x7F;seg[5]&=0x7F;seg[6]&=0x7F;
        seg[3]=0x80;seg[4]=0x80;seg[5]=0x80;seg[6]=0x80;//数字 4
        break;
    case 5:seg[1]&=0x7F;seg[3]&=0x7F;seg[4]&=0x7F;seg[6]&=0x7F;seg[7]&=0x7F;
        seg[1]=0x80;seg[3]=0x80;seg[4]=0x80;seg[6]=0x80;seg[7]=0x80;//数字 5
        break;
    case
6:seg[0]&=0x7F;seg[1]&=0x7F;seg[3]&=0x7F;seg[4]&=0x7F;seg[6]&=0x7F;seg[7]&=0x7F;

seg[0]=0x80;seg[1]=0x80;seg[3]=0x80;seg[4]=0x80;seg[6]=0x80;seg[7]=0x80;//数字 6
        break;
    case 7:seg[3]&=0x7F;seg[5]&=0x7F;seg[7]&=0x7F;
        seg[3]=0x80;seg[5]=0x80;seg[7]=0x80;//数字 7
        break;
    case
8:seg[0]&=0x7F;seg[1]&=0x7F;seg[3]&=0x7F;seg[4]&=0x7F;seg[5]&=0x7F;seg[6]&=0x7F;seg[
7]&=0x7F;

seg[0]=0x80;seg[1]=0x80;seg[3]=0x80;seg[4]=0x80;seg[5]=0x80;seg[6]=0x80;seg[7]=0x80;//
数字 8
        break;
    case
9:seg[1]&=0x7F;seg[3]&=0x7F;seg[4]&=0x7F;seg[5]&=0x7F;seg[6]&=0x7F;seg[7]&=0x7F;

seg[1]=0x80;seg[3]=0x80;seg[4]=0x80;seg[5]=0x80;seg[6]=0x80;seg[7]=0x80;//数字 9
        break;
    default:
        break;
}

if(!Time_ShowFlag)//设定时间闪烁
{
    if(dis_set_time)//设定时间大于 0
    {

```

VM6100 软件 V1.0

```

/*****dis_set_time*****/
SH=dis_set_time/3600/10;//计算十位单位的小时数
H=dis_set_time/3600%10;//计算个位单位的小时数
SM=dis_set_time%3600/60/10;//计算十分位单位的分钟数
M=dis_set_time%3600/60%10;//计算个分位单位的分钟数

switch(SH)
{
    case
0:seg[8]&=0xFE;seg[9]&=0xFE;seg[11]&=0xFE;seg[13]&=0xFE;seg[14]&=0xFE;seg[15]&=0xFE;
E;

seg[8]=0x01;seg[9]=0x01;seg[11]=0x01;seg[13]=0x01;seg[14]=0x01;seg[15]=0x01;//数字 0
    break;
    case 1:seg[11]&=0xFE;seg[13]&=0xFE;
        seg[11]=0x01;seg[13]=0x01;//数字 1
    break;
    case
2:seg[8]&=0xFE;seg[9]&=0xFE;seg[12]&=0xFE;seg[13]&=0xFE;seg[15]&=0xFE;

seg[8]=0x01;seg[9]=0x01;seg[12]=0x01;seg[13]=0x01;seg[15]=0x01;//数字 2
    break;
    case
3:seg[9]&=0xFE;seg[11]&=0xFE;seg[12]&=0xFE;seg[13]&=0xFE;seg[15]&=0xFE;

seg[9]=0x01;seg[11]=0x01;seg[12]=0x01;seg[13]=0x01;seg[15]=0x01;//数字 3
    break;
    case 4:seg[11]&=0xFE;seg[12]&=0xFE;seg[13]&=0xFE;seg[14]&=0xFE;
        seg[11]=0x01;seg[12]=0x01;seg[13]=0x01;seg[14]=0x01;//数字 4
    break;
    case
5:seg[9]&=0xFE;seg[11]&=0xFE;seg[12]&=0xFE;seg[14]&=0xFE;seg[15]&=0xFE;

seg[9]=0x01;seg[11]=0x01;seg[12]=0x01;seg[14]=0x01;seg[15]=0x01;//数字 5
    break;
    case
6:seg[8]&=0xFE;seg[9]&=0xFE;seg[11]&=0xFE;seg[12]&=0xFE;seg[14]&=0xFE;seg[15]&=0xFE;
E;

seg[8]=0x01;seg[9]=0x01;seg[11]=0x01;seg[12]=0x01;seg[14]=0x01;seg[15]=0x01;//数字 6
    break;
    case 7:seg[11]&=0xFE;seg[13]&=0xFE;seg[15]&=0xFE;
        seg[11]=0x01;seg[13]=0x01;seg[15]=0x01;//数字 7
    break;
    case
8:seg[8]&=0xFE;seg[9]&=0xFE;seg[11]&=0xFE;seg[12]&=0xFE;seg[13]&=0xFE;seg[14]&=0xFE;
E;seg[15]&=0xFE;

seg[8]=0x01;seg[9]=0x01;seg[11]=0x01;seg[12]=0x01;seg[13]=0x01;seg[14]=0x01;seg[15]=0
x01;//数字 8

```

```

        break;
    case
9:seg[9]&=0xFE;seg[11]&=0xFE;seg[12]&=0xFE;seg[13]&=0xFE;seg[14]&=0xFE;seg[15]&=0x
FE;

seg[9]=0x01;seg[11]=0x01;seg[12]=0x01;seg[13]=0x01;seg[14]=0x01;seg[15]=0x01;//数字 9
        break;
    default:
        break;
    }
    switch(H)
    {
        case
0:seg[8]&=0xFD;seg[9]&=0xFD;seg[11]&=0xFD;seg[13]&=0xFD;seg[14]&=0xFD;seg[15]&=0x
FD;

seg[8]=0x02;seg[9]=0x02;seg[11]=0x02;seg[13]=0x02;seg[14]=0x02;seg[15]=0x02;//数字 0
        break;
        case 1:seg[11]&=0xFD;seg[13]&=0xFD;
            seg[11]=0x02;seg[13]=0x02;//数字 1
            break;
        case
2:seg[8]&=0xFD;seg[9]&=0xFD;seg[12]&=0xFD;seg[13]&=0xFD;seg[15]&=0xFD;

seg[8]=0x02;seg[9]=0x02;seg[12]=0x02;seg[13]=0x02;seg[15]=0x02;//数字 2
        break;
        case
3:seg[9]&=0xFD;seg[11]&=0xFD;seg[12]&=0xFD;seg[13]&=0xFD;seg[15]&=0xFD;

seg[9]=0x02;seg[11]=0x02;seg[12]=0x02;seg[13]=0x02;seg[15]=0x02;//数字 3
        break;
        case 4:seg[11]&=0xFD;seg[12]&=0xFD;seg[13]&=0xFD;seg[14]&=0xFD;
            seg[11]=0x02;seg[12]=0x02;seg[13]=0x02;seg[14]=0x02;//数字 4
            break;
        case
5:seg[9]&=0xFD;seg[11]&=0xFD;seg[12]&=0xFD;seg[14]&=0xFD;seg[15]&=0xFD;

seg[9]=0x02;seg[11]=0x02;seg[12]=0x02;seg[14]=0x02;seg[15]=0x02;//数字 5
        break;
        case
6:seg[8]&=0xFD;seg[9]&=0xFD;seg[11]&=0xFD;seg[12]&=0xFD;seg[14]&=0xFD;seg[15]&=0x
FD;

seg[8]=0x02;seg[9]=0x02;seg[11]=0x02;seg[12]=0x02;seg[14]=0x02;seg[15]=0x02;//数字 6
        break;
        case 7:seg[11]&=0xFD;seg[13]&=0xFD;seg[15]&=0xFD;
            seg[11]=0x02;seg[13]=0x02;seg[15]=0x02;//数字 7
            break;
        case
8:seg[8]&=0xFD;seg[9]&=0xFD;seg[11]&=0xFD;seg[12]&=0xFD;seg[13]&=0xFD;seg[14]&=0x

```

```

FD;seg[15]&=0xFD;

seg[8]=0x02;seg[9]=0x02;seg[11]=0x02;seg[12]=0x02;seg[13]=0x02;seg[14]=0x02;seg[15]=0
x02;//数字 8
        break;
        case
9:seg[9]&=0xFD;seg[11]&=0xFD;seg[12]&=0xFD;seg[13]&=0xFD;seg[14]&=0xFD;seg[15]&=0
xFD;

seg[9]=0x02;seg[11]=0x02;seg[12]=0x02;seg[13]=0x02;seg[14]=0x02;seg[15]=0x02;//数字 9
        break;
        default:
        break;
    }
    switch(SM)
    {
        case
0:seg[8]&=0xFB;seg[9]&=0xFB;seg[11]&=0xFB;seg[13]&=0xFB;seg[14]&=0xFB;seg[15]&=0x
FB;

seg[8]=0x04;seg[9]=0x04;seg[11]=0x04;seg[13]=0x04;seg[14]=0x04;seg[15]=0x04;//数字 0
        break;
        case 1:seg[11]&=0xFB;seg[13]&=0xFB;
            seg[11]=0x04;seg[13]=0x04;//数字 1
            break;
        case
2:seg[8]&=0xFB;seg[9]&=0xFB;seg[12]&=0xFB;seg[13]&=0xFB;seg[15]&=0xFB;

seg[8]=0x04;seg[9]=0x04;seg[12]=0x04;seg[13]=0x04;seg[15]=0x04;//数字 2
        break;
        case
3:seg[9]&=0xFB;seg[11]&=0xFB;seg[12]&=0xFB;seg[13]&=0xFB;seg[15]&=0xFB;

seg[9]=0x04;seg[11]=0x04;seg[12]=0x04;seg[13]=0x04;seg[15]=0x04;//数字 3
        break;
        case 4:seg[11]&=0xFB;seg[12]&=0xFB;seg[13]&=0xFB;seg[14]&=0xFB;
            seg[11]=0x04;seg[12]=0x04;seg[13]=0x04;seg[14]=0x04;//数字 4
            break;
        case
5:seg[9]&=0xFB;seg[11]&=0xFB;seg[12]&=0xFB;seg[14]&=0xFB;seg[15]&=0xFB;

seg[9]=0x04;seg[11]=0x04;seg[12]=0x04;seg[14]=0x04;seg[15]=0x04;//数字 5
        break;
        case
6:seg[8]&=0xFB;seg[9]&=0xFB;seg[11]&=0xFB;seg[12]&=0xFB;seg[14]&=0xFB;seg[15]&=0x
FB;

seg[8]=0x04;seg[9]=0x04;seg[11]=0x04;seg[12]=0x04;seg[14]=0x04;seg[15]=0x04;//数字 6
        break;
        case 7:seg[11]&=0xFB;seg[13]&=0xFB;seg[15]&=0xFB;

```

```

        seg[11]=0x04;seg[13]=0x04;seg[15]=0x04;//数字 7
        break;
    case
8:seg[8]&=0xFB;seg[9]&=0xFB;seg[11]&=0xFB;seg[12]&=0xFB;seg[13]&=0xFB;seg[14]&=0x
FB;seg[15]&=0xFB;

seg[8]=0x04;seg[9]=0x04;seg[11]=0x04;seg[12]=0x04;seg[13]=0x04;seg[14]=0x04;seg[15]=0
x04;//数字 8

        break;
    case
9:seg[9]&=0xFB;seg[11]&=0xFB;seg[12]&=0xFB;seg[13]&=0xFB;seg[14]&=0xFB;seg[15]&=0
xFB;

seg[9]=0x04;seg[11]=0x04;seg[12]=0x04;seg[13]=0x04;seg[14]=0x04;seg[15]=0x04;//数字 9
        break;
    default:
        break;
}
switch(M)
{
    case
0:seg[8]&=0xF7;seg[9]&=0xF7;seg[11]&=0xF7;seg[13]&=0xF7;seg[14]&=0xF7;seg[15]&=0xF7
;

seg[8]=0x08;seg[9]=0x08;seg[11]=0x08;seg[13]=0x08;seg[14]=0x08;seg[15]=0x08;//数字 0
        break;
    case 1:seg[11]&=0xF7;seg[13]&=0xF7;
        seg[11]=0x08;seg[13]=0x08;//数字 1
        break;
    case
2:seg[8]&=0xF7;seg[9]&=0xF7;seg[12]&=0xF7;seg[13]&=0xF7;seg[15]&=0xF7;

seg[8]=0x08;seg[9]=0x08;seg[12]=0x08;seg[13]=0x08;seg[15]=0x08;//数字 2
        break;
    case
3:seg[9]&=0xF7;seg[11]&=0xF7;seg[12]&=0xF7;seg[13]&=0xF7;seg[15]&=0xF7;

seg[9]=0x08;seg[11]=0x08;seg[12]=0x08;seg[13]=0x08;seg[15]=0x08;//数字 3
        break;
    case 4:seg[11]&=0xF7;seg[12]&=0xF7;seg[13]&=0xF7;seg[14]&=0xF7;
        seg[11]=0x08;seg[12]=0x08;seg[13]=0x08;seg[14]=0x08;//数字 4
        break;
    case
5:seg[9]&=0xF7;seg[11]&=0xF7;seg[12]&=0xF7;seg[14]&=0xF7;seg[15]&=0xF7;

seg[9]=0x08;seg[11]=0x08;seg[12]=0x08;seg[14]=0x08;seg[15]=0x08;//数字 5
        break;
    case
6:seg[8]&=0xF7;seg[9]&=0xF7;seg[11]&=0xF7;seg[12]&=0xF7;seg[14]&=0xF7;seg[15]&=0xF7
;

```

```

seg[8]=0x08;seg[9]=0x08;seg[11]=0x08;seg[12]=0x08;seg[14]=0x08;seg[15]=0x08;//数字 6
    break;
    case 7:seg[11]&=0xF7;seg[13]&=0xF7;seg[15]&=0xF7;
        seg[11]=0x08;seg[13]=0x08;seg[15]=0x08;//数字 7
        break;
    case
8:seg[8]&=0xF7;seg[9]&=0xF7;seg[11]&=0xF7;seg[12]&=0xF7;seg[13]&=0xF7;seg[14]&=0xF7
;seg[15]&=0xF7;

seg[8]=0x08;seg[9]=0x08;seg[11]=0x08;seg[12]=0x08;seg[13]=0x08;seg[14]=0x08;seg[15]=0
x08;//数字 8

    break;
    case
9:seg[9]&=0xF7;seg[11]&=0xF7;seg[12]&=0xF7;seg[13]&=0xF7;seg[14]&=0xF7;seg[15]&=0xF
7;

seg[9]=0x08;seg[11]=0x08;seg[12]=0x08;seg[13]=0x08;seg[14]=0x08;seg[15]=0x08;//数字 9
    break;
    default:
        break;
}
}
else
{

seg[8]&=0xF0;seg[9]&=0xF0;seg[11]&=0xF0;seg[12]&=0xF0;seg[13]&=0xF0;seg[14]&=0xF0;s
eg[15]&=0xF0;

seg[8]=0x00;seg[9]=0x00;seg[11]=0x00;seg[12]=0x0F;seg[13]=0x00;seg[14]=0x00;seg[15]=0
x00;//显示 “--”
    }
}
else
{

seg[8]&=0xF0;seg[9]&=0xF0;seg[11]&=0xF0;seg[12]&=0xF0;seg[13]&=0xF0;seg[14]&=0xF0;s
eg[15]&=0xF0;

seg[8]=0x00;seg[9]=0x00;seg[11]=0x00;seg[12]=0x00;seg[13]=0x00;seg[14]=0x00;seg[15]=0
x00;
    }

    /*****dis_rel_time*****/
    if(Time.Rel || dis_set_time)//显示时间表示为 0
    {
        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
    }

```

```

switch(SH)
{
    case
0:seg[8]&=0xEF;seg[9]&=0xEF;seg[11]&=0xEF;seg[13]&=0xEF;seg[14]&=0xEF;seg[15]&=0x
EF;

seg[8]=0x10;seg[9]=0x10;seg[11]=0x10;seg[13]=0x10;seg[14]=0x10;seg[15]=0x10;//数字 0
    break;
    case 1:seg[11]&=0xEF;seg[13]&=0xEF;
        seg[11]=0x10;seg[13]=0x10;//数字 1
    break;
    case
2:seg[8]&=0xEF;seg[9]&=0xEF;seg[12]&=0xEF;seg[13]&=0xEF;seg[15]&=0xEF;

seg[8]=0x10;seg[9]=0x10;seg[12]=0x10;seg[13]=0x10;seg[15]=0x10;//数字 2
    break;
    case
3:seg[9]&=0xEF;seg[11]&=0xEF;seg[12]&=0xEF;seg[13]&=0xEF;seg[15]&=0xEF;

seg[9]=0x10;seg[11]=0x10;seg[12]=0x10;seg[13]=0x10;seg[15]=0x10;//数字 3
    break;
    case 4:seg[11]&=0xEF;seg[12]&=0xEF;seg[13]&=0xEF;seg[14]&=0xEF;
        seg[11]=0x10;seg[12]=0x10;seg[13]=0x10;seg[14]=0x10;//数字 4
    break;
    case
5:seg[9]&=0xEF;seg[11]&=0xEF;seg[12]&=0xEF;seg[14]&=0xEF;seg[15]&=0xEF;

seg[9]=0x10;seg[11]=0x10;seg[12]=0x10;seg[14]=0x10;seg[15]=0x10;//数字 5
    break;
    case
6:seg[8]&=0xEF;seg[9]&=0xEF;seg[11]&=0xEF;seg[12]&=0xEF;seg[14]&=0xEF;seg[15]&=0x
EF;

seg[8]=0x10;seg[9]=0x10;seg[11]=0x10;seg[12]=0x10;seg[14]=0x10;seg[15]=0x10;//数字 6
    break;
    case 7:seg[11]&=0xEF;seg[13]&=0xEF;seg[15]&=0xEF;
        seg[11]=0x10;seg[13]=0x10;seg[15]=0x10;//数字 7
    break;
    case
8:seg[8]&=0xEF;seg[9]&=0xEF;seg[11]&=0xEF;seg[12]&=0xEF;seg[13]&=0xEF;seg[14]&=0x
EF;seg[15]&=0xEF;

seg[8]=0x10;seg[9]=0x10;seg[11]=0x10;seg[12]=0x10;seg[13]=0x10;seg[14]=0x10;seg[15]=0
x10;//数字 8
    break;
    case
9:seg[9]&=0xEF;seg[11]&=0xEF;seg[12]&=0xEF;seg[13]&=0xEF;seg[14]&=0xEF;seg[15]&=0x
EF;

```

VM6100 软件 V1.0

```

seg[9]=0x10;seg[11]=0x10;seg[12]=0x10;seg[13]=0x10;seg[14]=0x10;seg[15]=0x10;//数字 9
    break;
    default:
        break;
}
switch(H)
{
    case
0:seg[8]&=0xDF;seg[9]&=0xDF;seg[11]&=0xDF;seg[13]&=0xDF;seg[14]&=0xDF;seg[15]&=0x
DF;

seg[8]=0x20;seg[9]=0x20;seg[11]=0x20;seg[13]=0x20;seg[14]=0x20;seg[15]=0x20;//数字 0
    break;
    case 1:seg[11]&=0xDF;seg[13]&=0xDF;
        seg[11]=0x20;seg[13]=0x20;//数字 1
        break;
    case
2:seg[8]&=0xDF;seg[9]&=0xDF;seg[12]&=0xDF;seg[13]&=0xDF;seg[15]&=0xDF;

seg[8]=0x20;seg[9]=0x20;seg[12]=0x20;seg[13]=0x20;seg[15]=0x20;//数字 2
    break;
    case
3:seg[9]&=0xDF;seg[11]&=0xDF;seg[12]&=0xDF;seg[13]&=0xDF;seg[15]&=0xDF;

seg[9]=0x20;seg[11]=0x20;seg[12]=0x20;seg[13]=0x20;seg[15]=0x20;//数字 3
    break;
    case 4:seg[11]&=0xDF;seg[12]&=0xDF;seg[13]&=0xDF;seg[14]&=0xDF;
        seg[11]=0x20;seg[12]=0x20;seg[13]=0x20;seg[14]=0x20;//数字 4
        break;
    case
5:seg[9]&=0xDF;seg[11]&=0xDF;seg[12]&=0xDF;seg[14]&=0xDF;seg[15]&=0xDF;

seg[9]=0x20;seg[11]=0x20;seg[12]=0x20;seg[14]=0x20;seg[15]=0x20;//数字 5
    break;
    case
6:seg[8]&=0xDF;seg[9]&=0xDF;seg[11]&=0xDF;seg[12]&=0xDF;seg[14]&=0xDF;seg[15]&=0x
DF;

seg[8]=0x20;seg[9]=0x20;seg[11]=0x20;seg[12]=0x20;seg[14]=0x20;seg[15]=0x20;//数字 6
    break;
    case 7:seg[11]&=0xDF;seg[13]&=0xDF;seg[15]&=0xDF;
        seg[11]=0x20;seg[13]=0x20;seg[15]=0x20;//数字 7
        break;
    case
8:seg[8]&=0xDF;seg[9]&=0xDF;seg[11]&=0xDF;seg[12]&=0xDF;seg[13]&=0xDF;seg[14]&=0x
DF;seg[15]&=0xDF;

seg[8]=0x20;seg[9]=0x20;seg[11]=0x20;seg[12]=0x20;seg[13]=0x20;seg[14]=0x20;seg[15]=0
x20;//数字 8
    break;

```

```

        case
9:seg[9]&=0xDF;seg[11]&=0xDF;seg[12]&=0xDF;seg[13]&=0xDF;seg[14]&=0xDF;seg[15]&=0
xDF;

seg[9]=0x20;seg[11]=0x20;seg[12]=0x20;seg[13]=0x20;seg[14]=0x20;seg[15]=0x20;//数字 9
        break;
        default:
        break;
    }
    switch(SM)
    {
        case
0:seg[8]&=0xBF;seg[9]&=0xBF;seg[11]&=0xBF;seg[13]&=0xBF;seg[14]&=0xBF;seg[15]&=0x
BF;

seg[8]=0x40;seg[9]=0x40;seg[11]=0x40;seg[13]=0x40;seg[14]=0x40;seg[15]=0x40;//数字 0
        break;
        case 1:seg[11]&=0xBF;seg[13]&=0xBF;
            seg[11]=0x40;seg[13]=0x40;//数字 1
        break;
        case
2:seg[8]&=0xBF;seg[9]&=0xBF;seg[12]&=0xBF;seg[13]&=0xBF;seg[15]&=0xBF;

seg[8]=0x40;seg[9]=0x40;seg[12]=0x40;seg[13]=0x40;seg[15]=0x40;//数字 2
        break;
        case
3:seg[9]&=0xBF;seg[11]&=0xBF;seg[12]&=0xBF;seg[13]&=0xBF;seg[15]&=0xBF;

seg[9]=0x40;seg[11]=0x40;seg[12]=0x40;seg[13]=0x40;seg[15]=0x40;//数字 3
        break;
        case 4:seg[11]&=0xBF;seg[12]&=0xBF;seg[13]&=0xBF;seg[14]&=0xBF;
            seg[11]=0x40;seg[12]=0x40;seg[13]=0x40;seg[14]=0x40;//数字 4
        break;
        case
5:seg[9]&=0xBF;seg[11]&=0xBF;seg[12]&=0xBF;seg[14]&=0xBF;seg[15]&=0xBF;

seg[9]=0x40;seg[11]=0x40;seg[12]=0x40;seg[14]=0x40;seg[15]=0x40;//数字 5
        break;
        case
6:seg[8]&=0xBF;seg[9]&=0xBF;seg[11]&=0xBF;seg[12]&=0xBF;seg[14]&=0xBF;seg[15]&=0x
BF;

seg[8]=0x40;seg[9]=0x40;seg[11]=0x40;seg[12]=0x40;seg[14]=0x40;seg[15]=0x40;//数字 6
        break;
        case 7:seg[11]&=0xBF;seg[13]&=0xBF;seg[15]&=0xBF;
            seg[11]=0x40;seg[13]=0x40;seg[15]=0x40;//数字 7
        break;
        case
8:seg[8]&=0xBF;seg[9]&=0xBF;seg[11]&=0xBF;seg[12]&=0xBF;seg[13]&=0xBF;seg[14]&=0x
BF;seg[15]&=0xBF;

```

```

seg[8]=0x40;seg[9]=0x40;seg[11]=0x40;seg[12]=0x40;seg[13]=0x40;seg[14]=0x40;seg[15]=0
x40;//数字 8
        break;
        case
9:seg[9]&=0xBF;seg[11]&=0xBF;seg[12]&=0xBF;seg[13]&=0xBF;seg[14]&=0xBF;seg[15]&=0
xBF;

seg[9]=0x40;seg[11]=0x40;seg[12]=0x40;seg[13]=0x40;seg[14]=0x40;seg[15]=0x40;//数字 9
        break;
        default:
            break;
    }
    switch(M)
    {
        case
0:seg[8]&=0x7F;seg[9]&=0x7F;seg[11]&=0x7F;seg[13]&=0x7F;seg[14]&=0x7F;seg[15]&=0x7F
;

seg[8]=0x80;seg[9]=0x80;seg[11]=0x80;seg[13]=0x80;seg[14]=0x80;seg[15]=0x80;//数字 0
        break;
        case 1:seg[11]&=0x7F;seg[13]&=0x7F;
            seg[11]=0x80;seg[13]=0x80;//数字 1
            break;
        case
2:seg[8]&=0x7F;seg[9]&=0x7F;seg[12]&=0x7F;seg[13]&=0x7F;seg[15]&=0x7F;

seg[8]=0x80;seg[9]=0x80;seg[12]=0x80;seg[13]=0x80;seg[15]=0x80;//数字 2
        break;
        case
3:seg[9]&=0x7F;seg[11]&=0x7F;seg[12]&=0x7F;seg[13]&=0x7F;seg[15]&=0x7F;

seg[9]=0x80;seg[11]=0x80;seg[12]=0x80;seg[13]=0x80;seg[15]=0x80;//数字 3
        break;
        case 4:seg[11]&=0x7F;seg[12]&=0x7F;seg[13]&=0x7F;seg[14]&=0x7F;
            seg[11]=0x80;seg[12]=0x80;seg[13]=0x80;seg[14]=0x80;//数字 4
            break;
        case
5:seg[9]&=0x7F;seg[11]&=0x7F;seg[12]&=0x7F;seg[14]&=0x7F;seg[15]&=0x7F;

seg[9]=0x80;seg[11]=0x80;seg[12]=0x80;seg[14]=0x80;seg[15]=0x80;//数字 5
        break;
        case
6:seg[8]&=0x7F;seg[9]&=0x7F;seg[11]&=0x7F;seg[12]&=0x7F;seg[14]&=0x7F;seg[15]&=0x7F
;

seg[8]=0x80;seg[9]=0x80;seg[11]=0x80;seg[12]=0x80;seg[14]=0x80;seg[15]=0x80;//数字 6
        break;
        case 7:seg[11]&=0x7F;seg[13]&=0x7F;seg[15]&=0x7F;
            seg[11]=0x80;seg[13]=0x80;seg[15]=0x80;//数字 7

```

```

        break;
    case
8:seg[8]&=0x7F;seg[9]&=0x7F;seg[11]&=0x7F;seg[12]&=0x7F;seg[13]&=0x7F;seg[14]&=0x7F
;seg[15]&=0x7F;

seg[8]=0x80;seg[9]=0x80;seg[11]=0x80;seg[12]=0x80;seg[13]=0x80;seg[14]=0x80;seg[15]=0
x80;//数字 8
        break;
    case
9:seg[9]&=0x7F;seg[11]&=0x7F;seg[12]&=0x7F;seg[13]&=0x7F;seg[14]&=0x7F;seg[15]&=0x7
F;

seg[9]=0x80;seg[11]=0x80;seg[12]=0x80;seg[13]=0x80;seg[14]=0x80;seg[15]=0x80;//数字 9
        break;
    default:
        break;
    }
}
else
{

seg[8]&=0x0F;seg[9]&=0x0F;seg[11]&=0x0F;seg[12]&=0x0F;seg[13]&=0x0F;seg[14]&=0x0F;s
eg[15]&=0x0F;

seg[8]=0x00;seg[9]=0x00;seg[11]=0x00;seg[12]=0xF0;seg[13]=0x00;seg[14]=0x00;seg[15]=0
x00;//显示 “-- --”
}

/*****设定时间冒号*****/
seg[10]&=0xDF;seg[10]=0x20;
/*****实际时间冒号*****/
seg[10]&=0xFD;seg[10]=0x02;

TM1640_Write_DATA(cnt, seg[cnt]);
cnt++;
if(cnt >= 16)
    cnt= 0;
// TM1640_Write_DATA(1, seg[1]);
// TM1640_Write_DATA(2, seg[2]);
// TM1640_Write_DATA(3, seg[3]);
// TM1640_Write_DATA(4, seg[4]);
// TM1640_Write_DATA(5, seg[5]);
// TM1640_Write_DATA(6, seg[6]);
// TM1640_Write_DATA(7, seg[7]);
// TM1640_Write_DATA(8, seg[8]);
// TM1640_Write_DATA(9, seg[9]);
// TM1640_Write_DATA(10, seg[10]);
// TM1640_Write_DATA(11, seg[11]);

```

```

// TM1640_Write_DATA(12, seg[12]);
// TM1640_Write_DATA(13, seg[13]);
// TM1640_Write_DATA(14, seg[14]);
// TM1640_Write_DATA(15, seg[15]);
}

/*
*****
* 函数原型: void Deal_Speed(float dT)
* 功    能: 速度显示处理
*****
*/

int16_t Speed_CalVal;//计算后的速度
void Deal_Speed(float dT)
{
    if(sys.Run_Status)//启动的情况下
    {
        if(Speed.ADDMode==0)//在电机控制中, 速度未处理
        {
            if(Speed.Ctrl >= Speed.Display_Rel)//控制速度大于实际速度
            {
                Speed.ADDMode = 1;//进入加速模式下
            }
            else if(Speed.Ctrl < Speed.Display_Rel)//控制速度小于实际速度
            {
                Speed.ADDMode = 2;//进入减速模式下
            }
            Speed_Arg.Kp = 20 * 0.001f;
            Speed_Arg.Ki = 44 * 0.001f;
        }
        if(Speed.ADDMode==1)//在进入加速模式下
        {
            Speed_Arg.Kp = 20 * 0.001f;
            Speed_Arg.Ki = 44 * 0.001f;
            if(Speed.Rel > 2000)
                Speed_CalVal = 2000 + ((Speed.Rel - 2000) * 2);
            else
                Speed_CalVal = Speed.Rel;
            Speed.New = Speed_CalVal;//记录当前速度

            if(Speed.New > Speed.Display_Rel)//当前速度大于显示速度
            {
                if(Speed.Display_Rel < Speed.New)
                    Speed.Display_Rel+=1;//显示当前速度
            }
            else//当前速度小于上一次速度
            {
                Speed.Display_Rel = Speed.Display_Rel;//显示上一次速度, 不让速度小于
                当前速度。呈现攀升速度的现象
            }
        }
    }
}

```

```

    if(sys.Motor_Stop == 0)
    {
        if(Speed.Display_Rel >= Speed.Ctrl)//实际速度大于等于控制速度
        {
            Speed.ADDMode = 3;//进入稳定模式
            return;
        }
    }
}
if(Speed.ADDMode == 2)//速度下降模式下
{
    if(Speed.Rel > 2000)
        Speed_CalVal = 2000 + ((Speed.Rel - 2000) * 2);
    else
        Speed_CalVal = Speed.Rel;
    Speed.New = Speed_CalVal;//记录当前速度

    if(Speed.New < Speed.Display_Rel)//当前速度小于上一次速度
    {
        if(Speed.Display_Rel > Speed.New)
            Speed.Display_Rel -= 1;//显示当前速度
    }
    else//当前速度大于上一次速度
    {
        Speed.Display_Rel = Speed.Display_Rel;//显示上一次速度，不让速度大于
        当前速度。呈现下降速度的现象
    }
    if(sys.Motor_Stop == 0)
    {
        if(Speed.Display_Rel <= Speed.Ctrl)//实际速度小于等于控制速度
        {
            Speed.ADDMode = 3;//进入稳定模式
            return;
        }
    }
}
else if(Speed.ADDMode == 3)//速度稳定模式下
{
    Speed.Display_Rel = Speed.Ctrl;//显示控制速度
    Speed_Arg.Kp = 5 * 0.001f;
    Speed_Arg.Ki = 10 * 0.001f;
}
}
else
{
    Speed.Display_Rel = 0;//实际速度显示为零
    Speed.New = 0;//现在的速度清零
    Speed.ADDMode = 0;//清除显示处理
}
}

```

```
/*
*****
* 函数原型: void Show_Display(void)
* 功    能: 显示屏幕内容
*****
*/
void Show_Display(void)
{
    Speed.Display_Set = Speed.Set;//显示设定速度
    Time.Display_Set = Time.Set;//显示设定时间
    Time.Display_Rel = Time.Rel + 59;//显示实际时间+59s

    Display_Screen(Speed.Display_Set,Speed.Display_Rel,Time.Display_Set,Time.Display_Rel);
}
```