

设备管理说明文档

在停车系统的设备端有如下的设备：A9核心控制板，STM32驱动板，STM32闸机板。本文档主要是针对后两者的设备管理信息进行说明。设备管理信息主要分为两部分，第一部分为设置信息，第二部分为获取设备状态信息。

1. 设备设置信息：
设备的设置信息分为以下几个部分：

设置报警温度		设置报警温度上限值，当箱体温度到达该设置的上限值后会，上报温度过高警报
设置补光		设置补光灯打开时的环境光值，可以根据实际需要对比光灯的阈值进行设置操作
设置补光时间		设置补光灯出发后，补光灯亮起的时间，单位：秒
设置氛围灯模式		设置等待是的氛围灯模式，设置开闸时的氛围灯模式，设置关闸时氛围灯模式
设置氛围灯时间		设置氛围灯的亮起时间，单位：秒
设置散热风扇		设置散热风扇的状态，打开和关闭，也可以使用PWM调制，可以调节风速的大小
设置开闸速度		设置闸机开启速度，分为三个档次：高速（1.5s），中速（2.5s），慢速（3.5s）
设置反弹敏感度		设置闸机遇到阻力反弹的敏感度，分为三个档次：高，中等，一般
设置设备编号		设置设备编号，该设备编号有现场布置时进行编号，方便以后进行调试，维护和升级

注：

2. 设备状态信息：

获取报警温度		获取报警温度上限值，当箱体温度到达该设置的上限值后会，上报温度过高警报
获取当前温度		获取箱体内当前温度值
获取当前环境光值		获取当前的环境光的数值
获取补光		获取补光灯打开时的环境光值，可以根据实际需要对比光灯的阈值进行设置操作
获取补光时间		获取当前设置补光灯出发后，补光灯亮起的时间，单位：秒
获取氛围灯模式		获取等待是的氛围灯模式，设置开闸时的氛围灯模式，设置关闸时氛围灯模式
获取氛围灯时间		获取补光灯在打开状态是的闪烁时长，单位：秒
获取散热风扇状态		获取散热风扇的状态，打开和关闭，也可以使用PWM调制，可以调节风速的大小
获取开闸速度		获取闸机开启速度，分为三个档次：高速（1.5s），中速（2.5s），慢速（3.5s）
获取反弹敏感度		获取闸机遇到阻力反弹的敏感度，分为三个档次：高，中等，一般
获取设备编号		获取设备编号，该设备编号有现场布置时进行编号，方便以后进行调试，维护和升级

注：

3. 数据结构设计:

数据结构也可以分为设置和获取两个部分来说:

1). 设置数据结构

```
#pragma pack(1)
typedef struct sAtmosphereMode{
    byte devState1:2;    bit0 - bit1:等待时氛围灯模式
    byte devState2:2;    bit2 - bit3:开闸时氛围灯模式
    byte devState3:2;    bit4 - bit5:关闸时氛围灯模式
    byte rev:2;          bit6 - bit7:保留位:
}ATMOSPHEREMODE;

typedef struct {
    byte SetAlarmTemperture;           //设置报警温度
    byte SetEnvermentLightValueL;      //设置补光环境光值低8位
    byte SetEnvermentLightValueH;      //设置补光环境光值高8位
    byte SetEnvermentLightTime;        //设置补光时间长度
    ATMOSPHEREMODE SetAtmosphereMode;  //设置氛围灯模式
    byte SetAtmosphereOpenTime;        //设置相关氛围灯开启时长
    byte SetFanValue;                  //设置风扇值
    byte SetOpenSpeed;                 //设置闸机开启速度
    byte SetBounceSensitivity;         //设置反弹敏感度
    byte SetDeviceIDPart1;             //设置设备ID的第一部分
    byte SetDeviceIDPart2;             //设置设备ID的第二部分
    byte SetDeviceIDPart3;             //设置设备ID的第三部分
    byte SetDeviceIDPart4;             //设置设备ID的第四部分
}SETINFO,*pSETINFO;
```

2). 获取状态信息数据结构:

```
#pragma pack(1)
typedef struct sAtmosphereMode{
    byte devState1:2;    //bit0 - bit1:等待时氛围灯模式
    byte devState2:2;    //bit2 - bit3:开闸时氛围灯模式
    byte devState3:2;    //bit4 - bit5:关闸时氛围灯模式
    byte rev:2;          //bit6 - bit7:保留位:
}ATMOSPHEREMODE;

typedef struct {
    byte GetAlarmTemperture;           //获取报警温度
    byte GetCurrentTemperture;         //获取当前温度
    byte GetEnvermentLightValueL;      //获取设置环境光值得低8位
    byte GetEnvermentLightValueH;      //获取设置环境光值得高8位
    byte GetCurrentEnvLightValueL;     //获取环境光值当前的低8位
    byte GetCurrentEnvLightValueH;     //获取环境光值得高8位
    byte GetEnvermentLightTime;        //获取当前补光时间长度
    ATMOSPHEREMODE GetAtmosphereMode;  //获取当前氛围灯的设置
    byte GetAtmosphereTime;            //获取氛围灯设置的时间
    byte GetFanValue;                  //获取风扇的设置信息
    byte GetOpenSpeed;                 //获取当前的开闸速度
    byte GetBounceSensitivity;         //获取反弹敏感度
    byte GetDeviceIDPart1;             //获取设备ID的第一部分
    byte GetDeviceIDPart2;             //获取设备ID的第二部分
    byte GetDeviceIDPart3;             //获取设备ID的第三部分
    byte GetDeviceIDPart4;             //获取设备ID的第四部分
}GETINFO,*pGETINFO;
```

4. 补充信息:

参阅文档: 《STM32核心板与A9通讯协议文档v1.1》

5. 版本信息:

v1.0 2017-11-24 Writeen By Bert.Zhang