上海仰邦软件科技有限公司

Font_Controller_ProtocolV2.5

仅供内部使用

2011-4-7

Version list:

Version	Date	Author	Comment, Description
1.0	2011-01-12	Qin ZhongLi	1. 字库卡协议初定
2.0	2011-04-27	Qin ZhongLi	1. 修改颜色转义字符定义，颜色参数用 ascii 码表示。 2. 修改实例信息，使实例信息能够在屏上显示
2.1	2011-5-5	Qin ZhongLi	1. 修改强制开关机命令，去掉掉电保存参数。 2. 修改定时开关机命令，增加定时时段 3. 增加取消定时开关机命令 4. 增加了对帧头的说明 5. 调整系统时间校时命令，把星期参数移动到秒后发送。
2.2	2011-5-12	Qin ZhongLi	1. 增加写多文件命令 2. 增加结束写多文件命令
2.3	2011-5-19	Qin ZhongLi	1. 删除随机显示方式 2. 去掉包头数据中多余设备类型的判断 3. 修改锁定/解锁节目命令中锁定状态的定义
2.4	2011-6-29	Qin ZhongLi	1. 增加设置动态区命令 2. 区域数据格式中增加动态区域编号的说明
2.5	2011-7-07	Qin ZhongLi	1. 删除动态节目的说明。 2. 增加动态区设置的说明

目录

1. 快速开发导读 4
2. 概述 5
2.1 功能描述 5
2.2 通讯方式 5
2.3 术语和缩略语 5
2.4 协议说明 5
3. 标准通讯格式 6
3.1 协议的分层 6
3.2 数据流向示意图 6
3.3 字符转义 7
4. 包头数据格式 8
5. 数据域定义 9
5.1 请求与答复 9
5.1.1 请求信息 9
5.1.2 答复信息 9
5.1.3 单包发送和分包发送 9
6. 错误状态 CmdError 10
7. ACK 和 NACK 11
7.1 ACK 11
7.2 NACK 11
8. 命令定义 12
8.1 格式化 12
8.2 删除文件 12
8.3 开始写文件 12
8.4 写文件 13
8.5 写多文件命令 14
8.6 结束写多文件命令 14
8.7 PING 命令 14
8.8 波特率自适应 15
8.9 系统复位命令 15
8.10 系统时间校正命令 15
8.11 查询当前固件 16
8.12 激活固件 16

8.13 强制开关机 17
8.14 定时开关机 17
8.15 取消定时开关机 17
8.16 设置亮度 18
8.17 锁定/解锁节目 18
8.18 设置动态区 19
9. 节目文件格式定义 20
9.1 区域数据格式 20
9.2 多分区节目文件实例 22
10. 图文区内容编辑 23
10.1 转义字符定义 23
10.2 文本内容编辑 23
10.3 时间内容编辑 24
附录 25
1. CRC16 校验算法 25

Font_Controller_ ProtocolV1.0

1. 快速开发导读

如果你想尽快开发出一个能简单控制的软件，建议按以下流程阅读协议，并进行软件开发。

1) 阅读协议说明，熟悉协议的相关规定。
2) 阅读标准通讯格式章节，熟悉协议的结构、格式。由于协议采用了分层的方式，并对各个层次分开描述，建议用户将协议各层串联起来看，以形成一个整体认识。
3) 如果只开发实时信息刷新功能，请阅读设置动态区和区域数据格式两个章节，后面几点可以略过
4) 阅读节目文件定义章节，该部分对节目的分类，节目文件格式的定义进行了说明，并附有一个多分区节目文件实例，建议详细阅读这部分，能够修改或制作自己的节目文件。
5) 阅读图文区内容编辑章节，该部分教你编辑自己的节目内容，掌握编辑方法后你可以编辑自己的节目内容并制作节目文件。
6) 阅读开始写文件命令和写文件命令，掌握节目文件的发送，特别要注意文件单包发送和多包发送的区别，如果需要连续发送多个节目文件，还需阅读写多文件命令和结束写多文件命令。
7) 完成以上五点，你已经可以编辑并发布自己的节目文件了，如果想对控制器进行更灵活的控制，可以阅读命令定义里的其他命令定义，以实现更多功能。

2. 概述

2.1 功能描述

类型：双色屏

扫描方式：1/4，1/8，1/16 扫，用命令可以修改。

标准字库：16/24/32 点阵字库

数据类型：字库方式

播放方式：节目顺序播放/定长播放可选

通讯方式：RS232/TTL、RS485/TTL、GPRS，波特率 9600/57600 可设，波特率自适应

外部闪存：512Kbyte/1Mbyte/2Mbyte/可选。

节目个数：64 个

区域个数：同时支持 5 个图文区，5 个动态区

动态区刷新率：>=0.5s

2.2 通讯方式

1) RS232/485 波特率：9600/57600, 无校验, 8 位数据, 1 停止位

2) GPRS /RF 波特率：9600/57600, 无校验, 8 位数据, 1 停止位

2.3 术语和缩略语

名称	说明
MSB	高位字节（Most Significant Byte）
LSB	低位字节（Least Significant Byte ）
CRC16	16 位的 CRC 校验，校验算法参考附录
CHK	CRC 校验值

2.4 协议说明

- 本文档中十六进制数据表示为 0x？？，如 0x7E。
- 本文档中涉及到的多字节参数，均以先低字节(LSB)后高字节(MSB)顺序发送，但是对于文件名和控制器名称等字符串参数，发送时按顺序发送，如“P123”则先发送‘P’，最后发送‘3’。
- 本文档中提及的数据长度，如无特别说明，皆是以字节（byte）为单位。
- 本文档中提及的时间相关的参数均采用 BCD 码
- 本文档中提及的颜色属性，均用 1Byte 来表示，其中，Bit0 表示红，bit1 表示绿，bit2 表示蓝，对于每一个 Bit，0 表示灭，1 表示亮 。
- 本文档中所有偏移量、块地址等参数如无特殊说明，均以 0 开始计算 。
- 本文档中区域的坐标定义按照左上角为坐标原点。横、纵坐标分别向右、向下延伸。
- 本文档中提及的“读取”和“写入”都是指上位机对控制器的动作
- 本文档中提及的保留字全部默认发送 0x00。

3. 标准通讯格式

协议结构如下：

帧头 0xA5 (8byte)	包头数据 (14byte)	数据域 (Nbyte)	包校验 (2byte)	帧尾 0x5A (1byte)

以下为协议中各项数据的说明：

1. 帧头由 8 个字节的 0xA5 组成，帧尾由一个字节的 0x5A 组成。帧头采用 8 个帧头，是为了防止 0xA5 丢失导致数据接收错误。在接收数据时，只要接收到一个 0xA5 就可认为接收到了帧头，然后等待下一个不是 0xA5 的数据，该数据为该帧的第一个有效数据。
2. 包头数据包含本包数据的一些属性，其定义参考包头数据格式定义。
3. 数据域为用户协议层数据，参考数据域定义。
4. 包校验为包头数据和数据域的 CRC16 校验值。CRC16 校验算法参考附录。

3.1 协议的分层

协议采用分层模式，分为协议层和物理传输层两层，其中数据域属于协议层数据。物理传输层又分为 PHY0 和 PHY1 两层，其中 PHY1 层数据结构如下：

包头数据(14byte)	数据域(Nbyte)	包校验(2byte)

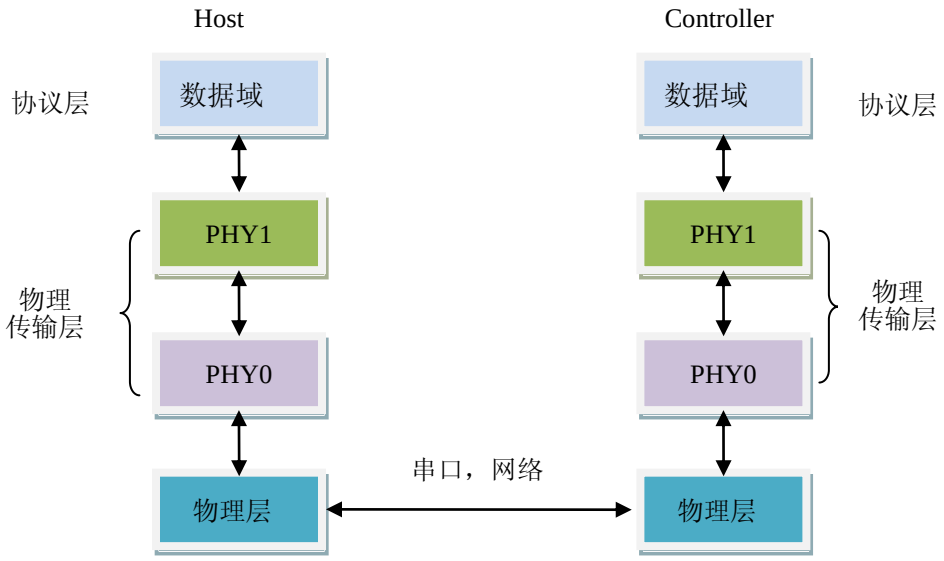
PHY1 层实现数据域的封包操作，它为数据域增加包头，并计算包数据的校验值。

PHY0 层数据结构为：

帧头(8byte)	PHY1 层数据(Nbyte)	帧尾(1byte)

PHY0 层为 PHY1 层数据增加帧头和帧尾，并对 PHY1 层数据进行转义（参考字符转义）。

3.2 数据流向示意图



在发送端，协议层数据先提交到 PHY1 层，对数据域进行封包操作。然后 PHY1 层报数据提交到 PHY0 层，对 PHY1 层数据进行字符转义并增加帧头帧尾，最后数据经过物理底层发送出去。

在接收端，控制器将物理底层接收到的数据发送到 PHY0 层，PHY0 层去除帧头帧尾，并对数据进行反转义，然后将数据提交到 PHY1 层。PHY1 层将判断包数据的正确性，并去除包头和包校验值，向协议层提交有用数据。

3.3 字符转义

- 封帧中遇到 0xA5，则将之转义为 0xA6，0x02；如遇到 0xA6，则将之转义为 0xA6，0x01 。
- 封帧中遇到 0x5A，则将之转义为 0x5B，0x02；如遇到 0x5B，则将之转义为 0x5B，0x01 。
- 解帧过程如果遇到连续两个字节为 0xA6, 0x02 ,则反转义为 0xA5 。
- 解帧过程如果遇到连续两个字节为 0xA6, 0x01 ,则反转义为 0xA6 。
- 解帧过程如果遇到连续两个字节为 0x5B, 0x02 ,则反转义为 0x5A 。
- 解帧过程如果遇到连续两个字节为 0x5B, 0x01 ,则反转义为 0x5B 。

注意：封帧过程中，所涉及校验的数据皆是转义之前的数据，所涉及的数据长度皆是转义之前的数据长度。

4. 包头数据格式

包头数据包含数据来源和流向，以及设备类型和协议版本等信息，它与数据域、包校验一起组成 PHY1 层数据包。

参数	数据长度	默认值	描述
目标地址(DstAddr)	2	0x0001	目标地址。 在 PHY 层，广播地址定义如下： 0xFFFF 为广播地址 1，此种模式下，控制器不返回数据，其可用于广播校时等命令。 0xFFFE 为广播地址 2，此种模式主要用于广播设置屏参，控制器需返回数据。在返回的数据帧中，地址也应为 0XFFFE。 0x8000~0xDFFF 地址为保留地址，对于物理层类型为 TCP/IP 或 GPRS 这种不需要处理 DstAddr 的，可将其目标地址设置为这个范围中的一个，默认设置地址为 0x8000。
源地址 (SrcAddr)	2	0x8000	源地址，几个特殊地址定义如下： PC 客户端软件从 0x8000 开始，范围为 0x8000~0xDFFF，用来代表不同客户端软件； 0xE000~0xFFFE 为保留地址
保留 (Reverse)	6	0x00	此处保留，将来可用于用户权限定义
设备类型 (DeviceType)	1	0x51	用于区分网络中不同的设备类型，定义如下： 0xFE——设备类型通配符 0x51——BX-4K1 0x52——BX-4K2
协议版本号 (ProtocolVersion)	1	0x01	协议版本号，用于区分控制卡使用的协议
数据域长度 (DataLenth)	2		数据域长度

5. 数据域定义

5.1 请求与答复

信息(Message)可分为请求(Request)和答复(Response)两种，请求是指从上位机到控制器(LED Controller)的信息，答复是指从控制器到上位机的回复。所有的数据通讯必须由上位机来发起。通讯过程中广播通讯控制器不回复，点对点通讯时可配置控制器是否回复，配置为不回复的通讯必须采用单包模式（参考单包发送和分包发送）。配置为有回复的通讯，如果在超时时间(Timeout Value) 之内没有收到回复，上位机将产生超时错误(Timeout Error)。

5.1.1 请求信息

请求信息是 PC 机软件到控制器的信息，其格式如下：

参数	数据长度	默认值	描述
CmdGroup	1		命令分组编号
Cmd	1		命令编号
Response	1		是否要求控制器回复。 0x01——控制器必须回复 0x02——控制器不必回复
DataLen	2		数据长度
Data	N		发送的数据

5.1.2 答复信息

答复信息为控制器到 PC 机软件的信息，其格式如下：

参数	数据长度	默认值	描述
CmdGroup	1		命令分组编号
Cmd	1		命令编号
CmdError	1		命令处理状态
DataLen	2		数据长度
Data	N		发送的数据

5.1.3 单包发送和分包发送

由于控制器数据接收缓存为 1024byte，所以当数据域数据大小大于 1024byte 时采用分包发送，小于 1024byte 时采用单包发送。采用分包发送时控制器可以不回复，但是这种做法不推荐，因为无法保证数据的正确接收。无论采用单包发送还是分包发送，在写文件命令前，必须先发送开始写文件命令（参考写文件命令）。

6. 错误状态CmdError

编号	名称	描述
0	ERR_NO	No Error
1	ERR_OUTOFGROUP	Command Group Error
2	ERR_NOCMD	Command Not Found
3	ERR_BUSY	The Controller is busy now
4	ERR_MEMORYVOLUME	Out of the Memory Volume
5	ERR_CHECKSUM	CRC16 Checksum Error
6	ERR_FILENOTEXIST	File Not Exist
7	ERR_FLASH	Flash Access Error
8	ERR_FILE_DOWNLOAD	File Download Error
9	ERR_FILE_NAME	Filename Error
10	ERR_FILE_TYPE	File type Error
11	ERR_FILE_CRC16	File CRC16 Error
12	ERR_FONT_NOT_EXIST	Font Library Not Exist
13	ERR_FIRMWARE_TYPE	Firmware Type Error (Check the controller type)
14	ERR_DATE_TIME_FORMAT	Date Time format error
15	ERR_FILE_EXIST	File Exist for File overwrite
16	ERR_FILE_BLOCK_NUM	File block number error

7. ACK和NACK

ACK 和 NACK 常用于不需要返回额外数据的命令的回复，比如 PING 命令时，控制器返回 ACK，表明控制器在线。

7.1 ACK

参数	数据长度	默认值	描述
CmdGroup	1	0xA0	命令分组编号
Cmd	1	0x00	命令编号
CmdError	1		命令处理状态
DataLen	2	0	数据长度

7.2 NACK

参数	数据长度	默认值	描述
CmdGroup	1	0xA0	命令分组编号
Cmd	1	0x01	命令编号
CmdError	1		命令处理状态
DataLen	2	0	数据长度

8. 命令定义

8.1 格式化

参数	数据长度	默认值	描述
CmdGroup	1	0XA1	命令分组编号
Cmd	1	0x00	命令编号
Response	1	0x01	是否要求控制器回复。 0x01——控制器必须回复 0x02——控制器不必回复
DataLen	2	0	数据长度

格式化命令会删除控制器中的所有文件，包括节目文件和字库文件，如果只希望删除节目文件而保留字库文件，该命令需慎用。命令回复为 **ACK** 或 **NACK**。

格式化指令实例：

A5 A5 A5 A5 A5 A5 A5 A5 01 00 00 80 00 00 00 00 00 00 51 01 05 00 A1 00 01 00 00 FF FF 5A

8.2 删除文件

参数	数据长度	默认值	描述
CmdGroup	1	0XA1	命令分组编号
Cmd	1	0x01	命令编号
Response	1	0x01	是否要求控制器回复。 0x01——控制器必须回复 0x02——控制器不必回复
DataLen	2		数据长度
FileNumber	2		需要删除文件总数， 如果该值为 0 则删除所有节目文件，后面的文件名数据不发送。
FileName1	4		文件名 1
FileName2	4		文件名 2
⋮	⋮	⋮	⋮
FileNameN	4		文件名 N

删除文件命令可以删除控制器中任意文件，包括节目文件和字库文件。如果需要删除所有的节目文件，则将 FileNumber 置为 0 即可。命令回复为 **ACK** 或 **NACK**。

删除文件指令实例：

A5 A5 A5 A5 A5 A5 A5 A5 01 00 00 80 00 00 00 00 00 00 51 01 07 00 A1 01 01 02 00 00 00 FF FF 5A

8.3 开始写文件

参数	数据长度	默认值	描述
CmdGroup	1	0XA1	命令分组编号
Cmd	1	0x05	命令编号
Response	1	0x01	是否要求控制器回复。 0x01——控制器必须回复 0x02——控制器不必回复

DataLen	2	9	数据长度
OverWrite	1		文件覆盖方式： 0x00——若文件系统中已经存在该文件，则不再下发该文件，返回 ERR_FILE_EXIST 状态。 0x01——若文件系统中已经存在该文件，则直接覆盖该文件。
FileName	4		文件名
FileLength	4		文件长度

控制器接收到开始写文件命令后，首先检查文件系统剩余容量，若剩余容量不足，则返回 ERR_MEMORYVOLUME 状态。返回值为 **ACK** 或 **NACK**。

开始写文件指令实例：

A5 A5 A5 A5 A5 A5 A5 A5 01 00 00 80 00 00 00 00 00 00 51 01 0E 00 A1 05 01 09 00 01 50 30 30 31 51 00 00 00 FF FF 5A

8.4 写文件

在发送该命令前请先发送开始写文件命令，以判断是否有足够空间存储该文件。该命令用于发送文件数据到控制器，如果文件大小不超出控制器接收缓冲区(1024Byte)的大小，可以单包发送。如果已经超出接收缓冲区(1024Byte)大小，则必须分包发送，分包大小可以自行定义。

参数	数据长度	默认值	描述
CmdGroup	1	0XA1	命令分组编号
Cmd	1	0x06	命令编号
Response	1	0x01	是否要求控制器回复。 0x01——控制器必须回复 0x02——控制器不必回复
DataLen	2		数据长度
FileName	4		文件名
LastBlockFlag	1	0x00	用于标志是否最后一包， 0x00——不是最后一包 0x01——最后一包
BlockNum	2		包号，如果是单包发送，则默认为 0x00
BlockLen	2		包长，若是单包发送，此处为文件长度
BlockAddr	4		本包数据在文件中的起始位置，如果是单包发送，此处默认为 0。
Data	N		文件包数据

控制器接收玩最后一个 Block 后，需对整个文件进行 CRC16 校验，如校验错误则返回 ERR_FILE_CRC16 错误。控制器接收完一个文件后，如校验出现错误(ERR_FILE_CRC16, ERR_FIRMWARE_TYPE)，控制器需自动删除此文件。如 BlockAddr 出错，即 BlockAddr 不连续则返回 ERR_FILE_BLOCK_NUM 错误状态和即将要写的块地址。返回值为：

参数	数据长度	默认值	描述
CmdGroup	1	0xA1	命令分组编号
Cmd	1	0x06	命令编号
CmdError	1		命令处理状态
DataLen	2		数据长度

BlockAddr	4		包地址 如果当前包成功接收则返回下一包的文件地址，否则返回当前包的文件地址

写文件指令实例，该实例分区为(0,0,16,16),显示内容为”012345QRSTTV’b”：

A5 A5 A5 A5 A5 A5 A5 A5 01 00 00 80 00 00 00 00 00 00 51 01 63 00 A1 06 01 5c 00 50 30 30 31 01 01 00 51 00 00 00 00 00 00 50 30 30 31 51 00 00 00 01 00 00 01 20 11 03 14 20 11 03 16 01 00 00 01 32 00 00 00 00 00 00 00 00 10 00 10 00 02 00 00 00 00 00 00 01 01 02 00 00 04 03 17 00 00 00 5c 46 45 30 30 30 5c 43 01 30 31 32 33 34 35 51 52 53 54 54 56 60 62 ff ff FF FF 5A

8.5 写多文件命令

参数	数据长度	默认值	描述
CmdGroup	1	0XA1	命令分组编号
Cmd	1	0x07	命令编号
Response	1	0x01	是否要求控制器回复。 0x01——控制器必须回复 0x02——控制器不必回复
DataLen	2	0	数据长度

在连续发送多个文件时，可以先发送写多文件命令，此时屏幕关闭，等待接收文件，在所有文件发送完毕后发送结束写多文件命令。结束等待。

指令实例：

A5 A5 A5 A5 A5 A5 A5 A5 01 00 00 80 00 00 00 00 00 00 51 01 05 00 A1 07 01 00 00 FF FF 5A

8.6 结束写多文件命令

参数	数据长度	默认值	描述
CmdGroup	1	0XA1	命令分组编号
Cmd	1	0x08	命令编号
Response	1	0x01	是否要求控制器回复。 0x01——控制器必须回复 0x02——控制器不必回复
DataLen	2	0	数据长度

指令实例：

A5 A5 A5 A5 A5 A5 A5 A5 01 00 00 80 00 00 00 00 00 00 51 01 05 00 A1 08 01 00 00 FF FF 5A

8.7 PING命令

参数	数据长度	默认值	描述
CmdGroup	1	0XA2	命令分组编号
Cmd	1	0x00	命令编号
Response	1	0x01	是否要求控制器回复。 0x01——控制器必须回复 0x02——控制器不必回复
DataLen	2		数据长度

返回值为 **ACK** 或 **NACK**。

PING 命令实例：

A5 A5 A5 A5 A5 A5 A5 A5 01 00 00 80 00 00 00 00 00 00 51 01 05 00 A2 00 01 00 00 FF FF 5A

8.8 波特率自适应

对于当前的控制器，只支持 9600bps 和 57600bps 两种波特率。所以，对于波特率自适应，我们这里采用简单的轮询模式。其过程如下：

a. 上位机使用 9600 波特率向控制器发出 Ping 命令 。

b. 如上位机在 2S 内收到正确的回复，则说明控制器的波特率即为 9600 。

c. 如上位机 2S 内没有收到正确的回复信息，则改用 57600 重复 a,b 。

d. 如上位机在 2S 内收到正确的回复，则说明控制器的波特率为 57600 。

e. 如上位机 2S 内没有收到正确的回复信息，则重复 a,b,c,d 。

f. 如 a,b,c,d 仍失败，则说明通讯失败，此时需提醒客户检查通讯线路 。

8.9 系统复位命令

参数	数据长度	默认值	描述
CmdGroup	1	0XA2	命令分组编号
Cmd	1	0x01	命令编号
Response	1	0x01	是否要求控制器回复。 0x01——控制器必须回复 0x02——控制器不必回复
DataLen	2		数据长度

控制器接接收到系统复位命令后，擦除除 Firmware 文件外的所有数据文件，然后复位 CPU。命令返回 **ACK** 或 **NACK**。

复位命令实例：

A5 A5 A5 A5 A5 A5 A5 A5 01 00 00 80 00 00 00 00 00 00 51 01 05 00 A2 01 01 00 00 FF FF 5A

8.10 系统时间校正命令

参数	数据长度	默认值	描述
CmdGroup	1	0XA2	命令分组编号
Cmd	1	0x03	命令编号
Response	1	0x01	是否要求控制器回复。 0x01——控制器必须回复 0x02——控制器不必回复
DataLen	2	8	数据长度
SystemTime	8		控制器时间，发送顺序如下（时间表示均采用 BCD 码）： 年(2)+月(1)+日(1) +时(1)+分(1)+秒(1) +星期(1) 注： 1. 年采用两个字节表示，如 2011 年表示为 0x20,0x11。先发送低位字节，后发送高位字节 2. 星期用 1~7 来表示，其中 7 表示星期日

返回 **ACK** 或 **NACK**。

校时命令实例：

A5 A5 A5 A5 A5 A5 A5 A5 01 00 00 80 00 00 00 00 00 00 51 01 0D 00 A2 03 01 08 00 11 20 04 06 12 43 23 03 FF FF 5A

8.11 查询当前固件

参数	数据长度	默认值	描述
CmdGroup	1	0XA2	命令分组编号
Cmd	1	0x04	命令编号
Response	1	0x01	是否要求控制器回复。 0x01——控制器必须回复 0x02——控制器不必回复
DataLen	2	0	数据长度

返回值：

参数	数据长度	默认值	描述
CmdGroup	1	0xA2	命令分组编号
Cmd	1	0x04	命令编号
CmdError	1		命令处理状态
DataLen	2		数据长度
FirmwareName	4		当前固件名
FirmwareVersion	4		固件版本号，如 1001 表示版本号为 1001。
FirmwareDateTime	19		Firmware生成日期与时间 例“2010-03-02 21:06:00“

查询固件命令实例：

A5 A5 A5 A5 A5 A5 A5 A5 01 00 00 80 00 00 00 00 00 00 51 01 05 00 A2 04 01 00 00 FF FF 5A

8.12 激活固件

固件文件下载到控制器以后，必须激活固件才能完成升级过程。

参数	数据长度	默认值	描述
CmdGroup	1	0XA2	命令分组编号
Cmd	1	0x05	命令编号
Response	1	0x01	是否要求控制器回复。 0x01——控制器必须回复 0x02——控制器不必回复
DataLen	2	4	数据长度
FirmwareName	4		固件文件名

回复为 **ACK** 或 **NACK**。如果此 Firmware 文件不存在，则返回 ERR_FILENOTEXIST 错误。如果此文件存在，则需先对此文件进行校验，如校验失败则返回 ERR_FILE_CRC16 错误。如激活过程中产生错误，控制器将继续保持原先激活的 Firmware。

激活固件命令实例：

A5 A5 A5 A5 A5 A5 A5 A5 01 00 00 80 00 00 00 00 00 00 51 01 09 00 A2 05 01 04 00 46 30 30 31 FF FF 5A

8.13 强制开关机

参数	数据长度	默认值	描述
CmdGroup	1	0XA3	命令分组编号
Cmd	1	0x00	命令编号
Response	1	0x01	是否要求控制器回复。 0x01——控制器必须回复 0x02——控制器不必回复
DataLen	2	0x01	数据长度
OnOffFlag	1		开关状态 0x01——开机 0x02——关机

返回 **ACK** 或 **NACK**。强制开关机命令的优先级高于定时开关机。如果发送强制开机命令，则在下一个定时开关机关机时段前都为开机状态，如果发送强制关机命令，则在下一个定时开关机开机时段前都为关机状态。

强制关机命令实例：

A5 A5 A5 A5 A5 A5 A5 A5 01 00 00 80 00 00 00 00 00 00 51 01 06 00 A3 00 01 01 00 02 FF FF 5A

8.14 定时开关机

参数	数据长度	默认值	描述
CmdGroup	1	0XA3	命令分组编号
Cmd	1	0x01	命令编号
Response	1	0x01	是否要求控制器回复。 0x01——控制器必须回复 0x02——控制器不必回复
DataLen	2		数据长度
TimerNum	1		定时器组数(最大只支持 3 组)
Timer1On	2		第一组定时开机时间，BCD 码表示，格式如下： 时（1）+分（1）
Timer1Off	2		第一组定时关机时间，BCD 码表示，格式如下： 时（1）+分（1）
……	……	……	……
Timer(n) On	2		第 n 组定时关机时间，BCD 码表示，格式如下： 时（1）+分（1）
Timer(n) Off	2		第 n 组定时关机时间，BCD 码表示，格式如下： 时（1）+分（1）

返回 **ACK** 或 **NACK**。定时开关机一旦设置，则一直有效直到取消定时开关机为止。

注意！关机时间一定要比开机时间大，不支持跨天设置。时间的发送顺序为先发小时后发分钟

定时开关机指令实例：

A5 A5 A5 A5 A5 A5 A5 A5 01 00 00 80 00 00 00 00 00 00 51 01 12 00 A3 01 01 0d 00 03 08 30 12 30 13 30 17 30 20 30 23 59 FF FF 5A

8.15 取消定时开关机

参数	数据长度	默认值	描述

CmdGroup	1	0XA3	命令分组编号
Cmd	1	0x08	命令编号
Response	1	0x01	是否要求控制器回复。 0x01——控制器必须回复 0x02——控制器不必回复
DataLen	2		数据长度

返回 **ACK** 或 **NACK**。

指令实例：

A5 A5 A5 A5 A5 A5 A5 A5 01 00 00 80 00 00 00 00 00 00 51 01 05 00 A3 08 01 01 00 FF FF 5A

8.16 设置亮度

该命令包含强制设置亮度和定时设置亮度功能。

参数	数据长度	默认值	描述
CmdGroup	1	0XA3	命令分组编号
Cmd	1	0x02	命令编号
Response	1	0x01	是否要求控制器回复。 0x01——控制器必须回复 0x02——控制器不必回复
DataLen	2		数据长度
BrightnessType	1		亮度调节方式 0x01——强制调节亮度 0x02——定时调节亮度 注： 若设置为强制调节方式，则定时器组和亮度数据不必发送； 若设置为定时调节方式，则 CurrentBrightness 默认为 0。
CurrentBrightness	1		当前亮度，强制调节亮度时有效。 亮度值为 0-15 共 16 级。亮度值为 15 时亮度最高
BrightnessValue	48		亮度值列表，用于定时调亮 把一天分为 48 个时段，每 30 分钟为一个时段，此处 48 个字节为每个时段对应的亮度值

返回 **ACK** 或 **NACK**。

强制设置亮度实例：

A5 A5 A5 A5 A5 A5 A5 A5 01 00 00 80 00 00 00 00 00 00 51 01 07 00 A3 02 01 02 00 01 0f FF FF 5A

8.17 锁定/解锁节目

被锁定的节目将一直播放直至解锁，该命令常用于节目的的选择播放。

参数	数据长度	默认值	描述
CmdGroup	1	0XA3	命令分组编号
Cmd	1	0x04	命令编号
Response	1	0x01	是否要求控制器回复。 0x01——控制器必须回复 0x02——控制器不必回复

DataLen	2	6	数据长度
StoreMode	1	0x00	锁定状态保存方式： 0x00——掉电不保存 0x01——掉电保存
LockFlag	1	0x00	锁定状态 0x00——解锁状态 0x01——锁定状态
ProgramFileName	4		节目文件名

如果需锁定的节目不存在，则返回 ERR_FILENOTEXIST 错误状态，返回 **ACK** 或 **NACK**。

锁定节目实例：

A5 A5 A5 A5 A5 A5 A5 A5 01 00 00 80 00 00 00 00 00 00 51 01 0B 00 A3 04 01 06 00 00 01 50 30 30 31 FF FF 5A

8.18 设置动态区

控制卡支持 5 个动态区，每个动态区数据量不能超过 1024byte，可以单独更新一个动态区，也可同时更新所有动态区。

参数	数据长度	默认值	描述
CmdGroup	1	0XA3	命令分组编号
Cmd	1	0x06	命令编号
Response	1	0x01	是否要求控制器回复。 0x01——控制器必须回复 0x02——控制器不必回复
DataLen	2		数据长度
DeleteAreaNum	1		要删除除的区域个数。 注意：如果该值为 0xFF，则删除所有动态区数据。
DeleteAreaId	N		需要删除的区域 ID 号
AreaNum	1		区域个数
AreaDataLen0	2		区域 0 数据长度
AreaData0	N		区域 0 数据，参考区域数据格式
⋮	⋮	⋮	⋮
AreaDataLenN	2		区域 N 数据长度
AreaDataN	N		区域 N 数据

返回 **ACK** 或 **NACK**。

设置动态区实例：(该实例有两个动态区，将屏参设为 192*32，显示内容为“012345QRSTTV’B”)

A5 A5 A5 A5 01 00 00 80 FF FF FF FF FF FF 51 06 6f 00 A3 06 01 6a 00 00 02 32 00 00 00 00 00 00 08 00 20 00 01 ff ff ff ff ff ff 01 02 02 00 00 04 03 17 00 00 00 5c 46 45 30 30 30 5c 43 01 30 31 32 33 34 35 51 52 53 54 54 56 60 62 32 00 00 0a 00 00 00 08 00 20 00 02 ff ff ff ff ff ff 01 02 02 00 00 04 03 17 00 00 00 5c 46 45 30 30 30 5c 43 01 30 31 32 33 34 35 51 52 53 54 54 56 60 62 FF FF 5A

9. 节目文件格式定义

参数	数据长度	默认值	描述
FileType	1	0x00	文件类型
FileName	4	PXXX	文件名 XXX 为文件编号，ASCII 码表示 文件名为字符串，发送按顺序发送，如“P123”则先发送‘P’，最后发送‘3’。
FileLen	4		文件长度
Reverse	1	0x00	保留字
DisplayType	2		节目播放方式 0——顺序播放 其他——定长播放的时间，单位为秒
PlayTimes	1		节目重复播放次数
ProgramLife	8		节目生命周期，发送顺序为： 起始年(2)+起始月(1)+起始日(1)+ 结束年(2)+结束月(1)+结束日(1) 注： 1. 时间均采用 BCD 码的方式 2. 年范围为 0x1900—0x2099，0xffff 为永久有效，先发送 LSB，后发送 MSB
ProgramWeek	1		节目的星期属性 1. Bit0 为1表示一周中的每一天都播放。 2. Bit0为0时，需判断bit1-bit7的来决定每天播放，bit1-bit7依次表示周一到周日。 3. 比特为0表示禁止播放，为1表示播放。
Reverse	2	0x00	保留字
AreaNum	1		区域个数
AreaDataLen0	4		区域 0 数据长度
AreaData0	N		区域 0 数据，参考区域数据格式
⋮	⋮	⋮	⋮
AreaDataLenN	4		区域 N 数据长度
AreaDataN	N		区域 N 数据，参考区域数据格式
CHK	2		整个节目文件的 CRC16 校验值

9.1 区域数据格式

参数	数据长度	默认值	描述
AreaType	1	0x00	区域类型
AreaX	2		区域 X 坐标，以字节(8 个像素点)为单位
AreaY	2		区域 Y 坐标，以像素点为单位
AreaWidth	2		区域宽度，以字节(8 个像素点)为单位
AreaHeight	2		区域高度，以像素点为单位
DynamicAreaLoc	1	0xFF	动态区域编号 注意：该参数只对动态区有效，其他区域为默认

			值，动态区必须统一编号，编号从 0 开始递增。
Reverse	6	0x00	保留字
FontCode	1	0x01	数据编码格式 0x01——GB2312
SingleLine	1	0x02	是否单行显示 0x01——单行显示 0x02——多行显示
NewLine	1		是否自动换行 0x01——不自动换行，显示数据在换行时必须插入换行符 0x02——自动换行，显示内容不需要换行符，但是只能使用统一的中文字体和英文字体
DisplayMode	1		显示方式，其定义如下： 0x01——静止显示 0x02——快速打出 0x03——向左移动 0x04——向右移动 0x05——向上移动 0x06——向下移动
ExitMode	1		退出方式
Speed	1		显示速度，定义如下： 0x00——4 字/秒 0x01——3 字/秒 0x02—— 2.5 字/秒 0x03 ——2 字/秒 0x04 ——4/3 字/秒 0x05——1 字/秒 0x06 —— 4/5 字/秒 0x07——4/6 字/秒 0x08——4/7 字/秒 0x09 ——4/8 字/秒
StayTime	1		显示停留时间，单位为 0.5s
DataLen	4		数据长度（包括换行，颜色等转义参数）
Data	N		显示数据参考图文区内容编辑

9.2 多分区节目文件实例

以下给出的实例为：屏大小为 1536*32

区域位置和大小：(184,0, 8,32)(176,0, 8,32)(168,0, 8,32) (160,0, 8,32) (152,0, 8,32)

A5 A5 A5 A5 A5 A5 A5 A5 01 00 00 80 00 00 00 00 00 00 51 01 3b 01 A1 06 01 34 01 50 30 30 31 01 01 00 29 01 00 00 00 00

//文件开始

00 50 30 30 31 29 01 00 00 01 00 00 01 20 11 03 14 20 11 03 16 01 00 00 05

32 00 00 00//区域 0 据长度

//区域 0 ，显示字串“全天 20 小时重”。

00 b8 00 00 00 08 00 20 00 02 00 00 00 00 00 00 01 01 02 00 00 00 00 17 00 00 00 5c 46 4f 30 30 30 5c 43 01 C8 AB CC EC A3 B2 A3 B0 D0 A1 CA B1 D6 D8

32 00 00 00 //区域 1 长度

//区域 1，显示字串“全天 20 小时重”。

00 b0 00 00 00 08 00 20 00 02 00 00 00 00 00 00 01 01 02 00 00 00 00 17 00 00 00 5c 46 4f 30 30 30 5c 43 01 C8 AB CC EC A3 B2 A3 B0 D0 A1 CA B1 D6 D8

32 00 00 00

//区域 2，显示字串“全天 20 小时重”。

00 a8 00 00 00 08 00 20 00 02 00 00 00 00 00 00 01 01 02 00 00 00 00 17 00 00 00 5c 46 4f 30 30 30 5c 43 01 C8 AB CC EC A3 B2 A3 B0 D0 A1 CA B1 D6 D8

32 00 00 00

//区域 3，显示字串“全天 20 小时重”。

00 a0 00 00 00 08 00 20 00 02 00 00 00 00 00 00 01 01 02 00 00 00 00 17 00 00 00 5c 46 4f 30 30 30 5c 43 01 C8 AB CC EC A3 B2 A3 B0 D0 A1 CA B1 D6 D8

32 00 00 00

//区域 4，显示字串“全天 20 小时重”。

00 98 00 00 00 08 00 20 00 02 00 00 00 00 00 00 01 01 02 00 00 00 00 17 00 00 00 5c 46 4f 30 30 30 5c 43 01 C8 AB CC EC A3 B2 A3 B0 D0 A1 CA B1 D6 D8

ff ff //文件校验值，文件结束

FF FF 5A

10. 图文区内容编辑

字库卡由于采用字符编码的方式，可以忽略不同区域类型数据的组织方式，每个区域都可以采用相同编辑的方式。编辑区域内容时将采用一些转义字符，以构成内容的显示样式。在显示文字前必须先定义文字使用的字库或图库名、文字颜色。如果文字样式没有改变，文字将按照之前设置的方式显示。在需要更改文字样式时，也要先定义文字使用的字库或图库名、文字颜色。编辑时间区时，时间参数也采用转义字符的方式，用户可以根据需要编辑出自己想要的各种显示样式。

10.1 转义字符定义

编号	转义字符	描述
1	“\FXXXX”	字库文件名，在编辑显示文字前，必须先定义文字使用哪种字库。”XXXX”为文字使用的字库文件名，字库文件名的定义： O000——16 点阵中文字库 O001——24 点阵中文字库 O002——32 点阵中文字库 E000——16 点阵英文字库 E001——24 点阵英文字库 E002——32 点阵英文字库
3	“\CX”	文字颜色，”X”为颜色编码，其定义如下： 0x31——表示红色 0x32——表示绿色 0x33——表示黄色
4	“\DX”	插入日期或时间的格式，X 定义如下： ‘S’: 秒 ‘M’: 分 ‘H’: 时 ‘D’: 日 ‘L’: 月 ‘Y’: 年 ‘W’:星期
8	“\n”	换行符

10.2 文本内容编辑

图文区可以插入文字、时间参数数据，下面将举例说明图文区的显示样式和数据格式：

“\FO000\C1 上海仰邦\FE000onbon\n\FO001\C2 欢迎你”,

字串中 O000 为 16 点阵中文字库名，E000 为 16 点阵英文字库名，O001 为 24 号中文字库，\C1 为红色参数，\C2 为绿色参数。则该字串将显示如下：

值得注意的是内容开始时必须设置所使用的中文或英文字库以及文字颜色，文字将按照此规则显示直至参数改变。

下面给出测试指令，该测试指令需将屏参设为 192*32，显示内容为“012345QRSTTV’B”:

先发送开始写文件命令：

A5 A5 A5 A5 A5 A5 A5 A5 01 00 00 80 00 00 00 00 00 00 51 01 0E 00 A1 05 01 09 00 01 50 30 30 31 51 00 00 00 FF FF 5A

然后发送写文件命令：

A5 A5 A5 A5 A5 A5 A5 A5 01 00 00 80 00 00 00 00 00 00 51 01 63 00 A1 06 01 5c 00 50 30 30 31 01 01 00 51 00 00 00 00 00 00 50 30 30 31 51 00 00 00 01 00 00 01 ff ff ff ff ff ff ff ff 01 00 00 01 32 00 00 00 00 00 00 00 00 10 00 10 00 02 00 00 00 00 00 00 01 01 02 00 00 04 03 17 00 00 00 5c 46 45 30 30 30 5c 43 31 30 31 32 33 34 35 51 52 53 54 54 56 60 62 ff ff FF FF 5A

10.3 时间内容编辑

通过插入时间参数的方式，客户可以编辑任意显示格式的时间，如以下字串:

“\FO000\C1 今天是\FE000\DY 年\DL 月\DD 日\n\C3\DH 时\DM 分\DS 秒”

\C3 为显示黄色，以上字串将显示为以下格式：

下面给出测试指令，该测试指令需将屏参设为 192*32，显示内容为“年-月-秒”

先发送开始写文件命令：

A5 A5 A5 A5 A5 A5 A5 A5 01 00 00 80 00 00 00 00 00 00 51 01 0E 00 A1 05 01 09 00 01 50 30 30 31 51 00 00 00 FF FF 5A

然后发送写文件命令：

A5 A5 A5 A5 A5 A5 A5 A5 01 00 00 80 00 00 00 00 00 00 51 01 63 00 A1 06 01 5c 00 50 30 30 31 01 01 00 51 00 00 00 00 00 00 50 30 30 31 51 00 00 00 01 00 00 01 ff ff ff ff ff ff ff ff 01 00 00 01 32 00 00 00 00 00 00 00 00 10 00 20 00 02 00 00 00 00 00 00 01 01 02 00 00 04 03 17 00 00 00 5c 46 45 30 30 30 5c 43 31 5c 44 59 2d 5c 44 4c 2d 5c 44 53 20 20 20 ff ff FF FF 5A

附录

1. CRC16校验算法

For the calculation of the CRC-16 the following polynomial is used:

X^16 + X ^15 + X^2 + 1 = (x + 1) * (X^15 + x + 1).

For this polynomial efficient calculation via a table is possible. Below the algorithm is given in C:

```
#define CRC(crc,byte) (((crc) >> 8 ) ^ tabel[((crc) ^ (unsigned int) (byte)) & 0XFF])
static const uint16  tabel[256] = {
 0X0000, 0XC0C1, 0XC181, 0X0140, 0XC301, 0X03C0, 0X0280, 0XC241,
 0XC601, 0X06C0, 0X0780, 0XC741, 0X0500, 0XC5C1, 0XC481, 0X0440,
 0XCC01, 0X0CC0, 0X0D80, 0XCD41, 0X0F00, 0XCFC1, 0XCE81, 0X0E40,
 0X0A00, 0XCAC1, 0XCB81, 0X0B40, 0XC901, 0X09C0, 0X0880, 0XC841,
 0XD801, 0X18C0, 0X1980, 0XD941, 0X1B00, 0XDBC1, 0XDA81, 0X1A40,
 0X1E00, 0XDEC1, 0XDF81, 0X1F40, 0XDD01, 0X1DC0, 0X1C80, 0XDC41,
 0X1400, 0XD4C1, 0XD581, 0X1540, 0XD701, 0X17C0, 0X1680, 0XD641,
 0XD201, 0X12C0, 0X1380, 0XD341, 0X1100, 0XD1C1, 0XD081, 0X1040,
 0XF001, 0X30C0, 0X3180, 0XF141, 0X3300, 0XF3C1, 0XF281, 0X3240,
 0X3600, 0XF6C1, 0XF781, 0X3740, 0XF501, 0X35C0, 0X3480, 0XF441,
 0X3C00, 0XFCC1, 0XFD81, 0X3D40, 0XFF01, 0X3FC0, 0X3E80, 0XFE41,
 0XFA01, 0X3AC0, 0X3B80, 0XFB41, 0X3900, 0XF9C1, 0XF881, 0X3840,
 0X2800, 0XE8C1, 0XE981, 0X2940, 0XEB01, 0X2BC0, 0X2A80, 0XEA41,
 0XEE01, 0X2EC0, 0X2F80, 0XEF41, 0X2D00, 0XEDC1, 0XEC81, 0X2C40,
 0XE401, 0X24C0, 0X2580, 0XE541, 0X2700, 0XE7C1, 0XE681, 0X2640,
 0X2200, 0XE2C1, 0XE381, 0X2340, 0XE101, 0X21C0, 0X2080, 0XE041,
 0XA001, 0X60C0, 0X6180, 0XA141, 0X6300, 0XA3C1, 0XA281, 0X6240,
 0X6600, 0XA6C1, 0XA781, 0X6740, 0XA501, 0X65C0, 0X6480, 0XA441,
 0X6C00, 0XACC1, 0XAD81, 0X6D40, 0XAF01, 0X6FC0, 0X6E80, 0XAE41,
 0XAA01, 0X6AC0, 0X6B80, 0XAB41, 0X6900, 0XA9C1, 0XA881, 0X6840,
 0X7800, 0XB8C1, 0XB981, 0X7940, 0XBB01, 0X7BC0, 0X7A80, 0XBA41,
 0XBE01, 0X7EC0, 0X7F80, 0XBF41, 0X7D00, 0XBDC1, 0XBC81, 0X7C40,
 0XB401, 0X74C0, 0X7580, 0XB541, 0X7700, 0XB7C1, 0XB681, 0X7640,
 0X7200, 0XB2C1, 0XB381, 0X7340, 0XB101, 0X71C0, 0X7080, 0XB041,
 0X5000, 0X90C1, 0X9181, 0X5140, 0X9301, 0X53C0, 0X5280, 0X9241,
 0X9601, 0X56C0, 0X5780, 0X9741, 0X5500, 0X95C1, 0X9481, 0X5440,
```

```
  0X9C01, 0X5CC0, 0X5D80, 0X9D41, 0X5F00, 0X9FC1, 0X9E81, 0X5E40,
  0X5A00, 0X9AC1, 0X9B81, 0X5B40, 0X9901, 0X59C0, 0X5880, 0X9841,
  0X8801, 0X48C0, 0X4980, 0X8941, 0X4B00, 0X8BC1, 0X8A81, 0X4A40,
  0X4E00, 0X8EC1, 0X8F81, 0X4F40, 0X8D01, 0X4DC0, 0X4C80, 0X8C41,
  0X4400, 0X84C1, 0X8581, 0X4540, 0X8701, 0X47C0, 0X4680, 0X8641,
  0X8201, 0X42C0, 0X4380, 0X8341, 0X4100, 0X81C1, 0X8081, 0X4040
};
unsigned short CalcCRC(data, size)
char *data; int size;
{
int i;
 unsigned short crc = 0;
for (i = 0; i < size; i++) {
 crc = CRC(crc, data[i]);
}
return crc;
 }
```