

上海仰邦软件科技有限公司

6th 动态区域用户手册

Copyright

All rights reserved. No part of this publication may be reproduced in any form by print, photo print, microfilm or any other means without written permission by ONBON.

©2009-2018Onbon



Version list:

1.0	2013-04-17	张文忠	初始版本
1.1	2013-04-23	张文忠	更新了包头数据格式，以避免出现非对齐访问
1.2	2013-04-27	张文忠	动态区域的页数更新为 2 个字节表示，以便可以存储更多的页 修改了动态区域 ImmePlay 参数的定义 为避免过多转义，动态区域命令字修改为 0xA7
1.3	2014-10-21	覃忠利	增加 5q 系列卡类型 增加 5q 系列数据编码格式
1.4	2015-03-06	张文忠	抽取了 5E 和 5Q 的共性，去除了部分无关紧要的信息 删除了更新页信息和删除页信息命令
1.5	2016-08-01	段志园	对颜色属性由 1 字节调整到 4 字节 对更新动态区域信息命令做出调整，前部分为多个区域的参数，后部分为多个区域的数据
1.6	2016-10-28	段志园	为方便上位机开发，对数据偏移 DataOffset 做出调整 6Q 仅支持网口，通讯方式说明中去除串口通讯 对动态区域个数限制
1.7	2018-03-12	chenyd	添加其它命令(ping 命令 、 亮度调节命令 、 亮度查询命令)
1.8	2018-5-17	QINZL	1. 包头数据增加 6E 系列型号
1.9	2018-8-16	张继力	2. 附录增加 6e 点阵编码说明
2.0	2018-10-24	zhangwz	增加图文动态区语音
2.0	2018-10-26	Chenjd	精简动态区域的 ImmePlay 参数，删除其为 2 时的功能
2.0	2018-10-26	Chenjd	1、 增加独立语音命令 2、添加 6EX 的型号 3、添加 ImmePlay 和 RelateAllPro 参数的详细说明 4、添加 AreaDataLen 参数说明 5、备注删除动态区个数为 0xFF，DeleteAreald 不需要发送
2.0	2019-09-30	Chenjd	将独立语音命令修改为语音队列命令

目录

1. 概述	5
1.1 功能描述	5
1.2 通讯方式	5
1.3 动态区域概况	5
1.4 术语和缩略语	5
1.5 协议说明	5
2. 标准通讯格式	7
2.1 协议的分层	7
2.2 数据流向示意图	7
2.3 字符转义	8
3. 网络通讯	8
3.1 网络连接模式	8
3.1.1 单机直连	9
3.1.2 以太网连接	9
3.1.3 跨 Internet/Intranet 连接	9
3.2 TCP 与 UDP 端口号	10
4. 包头数据格式	11
5. 数据域定义	12
5.1 请求与答复	12
5.1.1 Request 信息格式	12
5.1.2 Response 信息格式	12
5.2 Status 与 Error	12
5.2.1 Status 寄存器定义	12
5.2.2 Error 寄存器定义	12
5.3 ACK 与 NACK	13
5.3.1 ACK 答复	13
5.3.2 NACK 答复	13
5.4 动态区域相关命令	13
5.4.1 更新动态区域信息	13
5.4.2 删除动态区域区域信息	19
5.4.3 语音队列命令	21
5.4.4 区域边框属性	23

5.5	其它命令.....	23
5.5.1	Ping 命令	23
5.5.2	亮度调节命令.....	24
5.5.3	亮度值查询命令	25
附录 1	CRC16 校验算法.....	27
附录 2	点阵编码规则	29



6th 动态区域用户手册

1. 概述

1.1 功能描述

类型：全彩(6Q)

扫描方式：静态，1/2，1/4，1/8，1/16 扫

数据类型：点阵方式

播放方式：节目顺序播放/锁定播放可选

通讯方式：Ethernet

节目个数：6Q 系列 1000 个

区域个数：同时支持 16 个图文区，32 个动态区

1.2 通讯方式

- 1) Ethernet 10/100M 自适应

1.3 动态区域概况

- 1) 动态区域可与异步节目同时播放，也可以单独播放
- 2) 控制卡支持动态区域 32 个,每个动态区域最大容量 1M
- 3) 动态区域信息不能掉电保存，但可以无限次更新

1.4 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

1.5 协议说明

- ◆ 本文中十六进制数据表示为 0x??，如 0x7E。
- ◆ 本文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
- ◆ 本文中提及的时间相关的参数均采用 BCD 码

- ◆ 本文中提及的颜色属性，均用 4Byte 来表示，其中，Byte0 表示红，Byte1 表示绿，Byte2 表示蓝，Byte3 保留
- ◆ 本文中所有偏移量、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。
- ◆ 本文中提及的“读取”和“写入”都是指上位机对控制器的动作
- ◆ 本文中提及的保留字全部默认发送 0x00。



2. 标准通讯格式

协议结构如下：

帧头 0xA5 (8byte)	包头数据 (16byte)	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。

2.1 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(16byte)	数据域(Nbyte)	包校验(2byte)
--------------	------------	------------

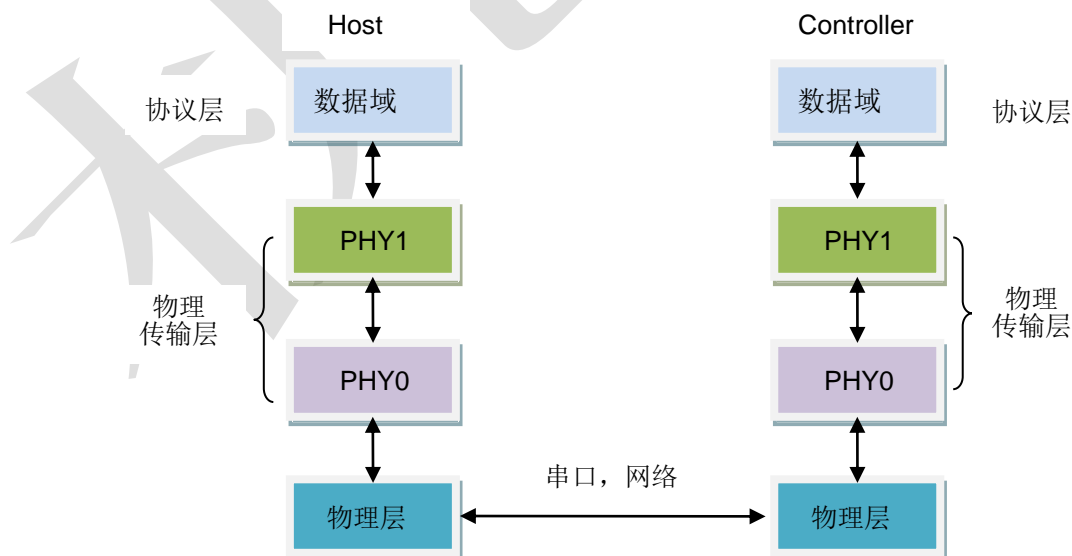
PHY1 层实现数据域的封包操作，它为数据域增加包头，并计算包数据的校验值。

PHY0 层数据结构为：

帧头(8byte)	PHY1 层数据(N + 18byte)	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](#)）。

2.2 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。

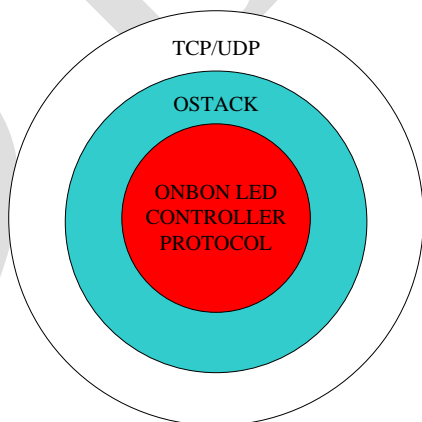
2.3 字符转义

- ◆ 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。

3. 网络通讯

网络通讯部分的通讯协议格式与串口通讯部分完全相同，不同之处仅在于，网络通讯最底层采用 UDP 或 TCP 协议。因此，此处只针对单用于以太网通讯的相关协议进行描述。

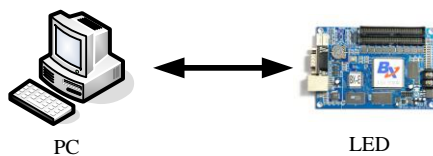


3.1 网络连接模式

为了方便用户的使用，我们将控制器与 PC 之间的连接，分为单机直连、通过以太网连接、通过 Internet 连接三种模式。

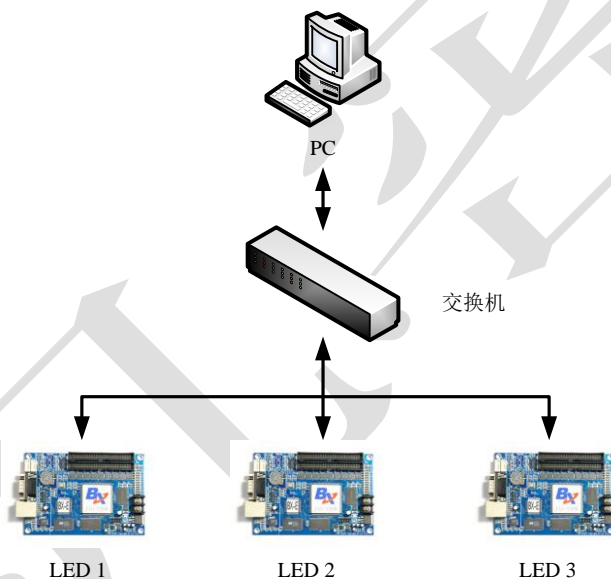
3.1.1 单机直连

单机直连是指，PC 和 LED 控制器之间不经过任何交换机或路由而直接连接。在这种连接模式下，为了减少参数设置上的过多操作，PC 与控制器之间全部采用广播地址来进行通讯，而无须对控制器进行 IP 地址的设置。



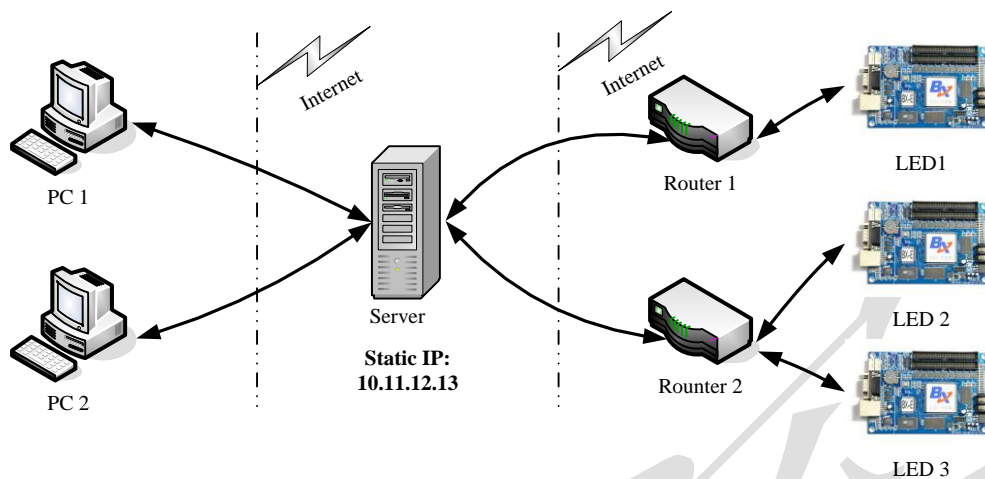
3.1.2 以太网连接

以太网连接，是指 PC 与控制器处于同一以太网内部。在这种连接模式下，控制器可以支持 DHCP 和手动设置 IP（静态 IP）两种模式。



3.1.3 跨Internet/Intranet连接

跨 Internet/Intranet 连接，是指 PC 软件和 LED 控制器之间通过 Internet/Intranet 进行连接。由于，在 Internet 上为每一个控制器分配一个固定的 IP 地址成本太高（基本上不可能实现）。因此，在此我们采用了一种基于 Server 的连接模式，具体连接模式如下：



- ◆ 架设一台具有固定 IP 的服务器（Server）
- ◆ 各控制器定时向服务器发送自身的 IP 地址和端口号
- ◆ PC 软件定时从服务器上下载各控制器的 IP 地址和端口号列表
- ◆ PC 软件通过 IP 与端口列表对 LED 控制器进行访问

注：在此种模式下，Server 和 PC 可以是同一台机器

3.2 TCP与UDP端口号

TCP 端口号，默认值为 5005，UDP 端口号为 5007。

4. 包头数据格式

Size (byte)	Name	Description
2	DstAddr	目标地址，即目标屏幕的屏号
2	SrcAddr	源地址 PC 客户端软件默认为 0x8000
1	ProtocolVer	动态区域协议版本号为 0xf0
1	Reserved	保留字
2	DeviceType	设备类型，用于区分网络中不同类型的设备，具体如下： BX-6QL 0x0066 BX-6Q1 0x0166 BX-6Q2 0x0266 BX-6Q3 0x0366 BX-6Q2L 0x0466 BX-6Q3L 0x0566 BX-6E1 0X0174 BX-6E2 0X0274 BX-6E3 0X0374 BX-6E1X 0X0474 BX-6E2X 0X0574
4	Reserved	保留字
4	DataLen	数据域的长度(仅针对数据域，不包括包头和包校验字节)

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 PC 软件到 LED 控制器的命令, 答复是指从控制器到 PC 的回复。

所有的数据通讯必须由 PC 来发起。

5.1.1 Request信息格式

参数	数据长度	默认值	描述
RtnReq	1	0x01	是否要求返回状态 注: 对于有些命令, 此字段必须为 0x01, 如: Ping 命令
CmdGroup	1	0x00	命令组
Cmd	1	0x00	命令编号
Data	N		发送的数据

5.1.2 Response信息格式

参数	数据长度	默认值	描述
Reserved	1	0x00	保留字
CmdGroup	1	0x00	命令组
Cmd	1	0x00	命令编号
Status	2		控制器状态寄存器
Error	2		错误状态寄存器
DataLen	2	N	数据长度, 只包括 DATA, 而不包括 Status 和 Error
Data	N		发送的数据

5.2 Status与Error

5.2.1 Status寄存器定义

位置	参数	是否可重置	描述
Bit0	NACK	N	0 –Request 被正常处理 1 –有错误存在
Bit2– Bit15	Reserved		

5.2.2 Error寄存器定义

Error Number	Name	Description
0	ERR_NO	No Error
1	ERR_OUTOFGROUP	命令组错误
2	ERR_NOCMD	此命令不存在
3	ERR_BUSY	控制器忙
4	ERR_MEMORYVOLUME	存储器容量越界
5	ERR_CHECKSUM	数据包 CRC 校验错误
6	ERR_FILENOTEXIST	此文件不存在

7	ERR_FLASH	Flash 访问错误
8	ERR_FILE_DOWNLOAD	文件下载错误
9	ERR_FILE_NAME	文件名错误
10	ERR_FILE_TYPE	文件类型错误
11	ERR_FILE_CRC16	文件校验错误
12	ERR_FONT_NOT_EXIST	字库文件不存在
13	ERR_FIRMWARE_TYPE	Firmware 与控制器类型不匹配
14	ERR_DATE_TIME_FORMAT	日期时间格式错误
15	ERR_FILE_EXIST	此文件已存在
16	ERR_FILE_BLOCK_NUM	文件 Block 号错误
17	ERR_CONTROLLER_TYPE	控制器类型不匹配
18	ERR_SCREEN_PARA	控制器参数越界或错误
19	ERR_CONTROLLER_ID	控制器 ID 错误

5.3 ACK与NACK

5.3.1 ACK答复

当一个请求信息被正常处理，没有发生任何错误，且不需要向 PC 回复任何附加内容时，需返回 ACK。其格式如下表所示：

参数	数据长度	默认值	描述
Reserved	1	0x00	保留字
CmdGroup	1	0xA0	命令组
Cmd	1	0x00	命令编号
Status	2	Bit0 = 0	控制器状态
Error	2		错误状态寄存器
DataLen	2	0x00	数据长度

5.3.2 NACK答复

参数	数据长度	默认值	描述
Reserved	1	0x00	保留字
CmdGroup	1	0xA0	命令组
Cmd	1	0x01	命令编号
Status	2	Bit0 = 1	控制器状态
Error	2		错误状态寄存器
DataLen	2	0x00	数据长度

5.4 动态区域相关命令

5.4.1 更新动态区域信息

参数	数据长度	默认值	描述
RtnReq	1	0x01	要求返回
CmdGroup	1	0xA7	命令组
Cmd	1	0x00	命令编号
Reserved	2	0x00	保留字

AreaNum	2	0x01	要更新区域的个数
AreaDataLen	4		区域数据长度，即下面紧挨着的一个区域数据的总长度(不包含自身的 4 字节，从参数 AreaId 到参数 DataLen)
AreaId	1	0x00	区域序号，从 0 开始
RunMode	1	0x00	动态区运行模式 0— 动态区数据循环显示。 1— 动态区数据显示完成后静止显示最后一页数据。 2— 动态区数据循环显示，超过设定时间后数据仍未更新时不再显示 3— 动态区数据循环显示，超过设定时间后数据仍未更新时显示 Logo 信息,Logo 信息即为动态区域的最后一页信息 4— 动态区数据顺序显示，显示完最后一页后就不再显示
Timeout	2		动态区数据超时时间，单位为秒
RelateAllPro	1		当该字节为 1 时，所有异步节目播放时都允许播放该动态区域； (异步节目和动态区一起播放，我们默认为不重叠的区域，重叠情况下，某些特技组合会导致花屏。所以，要保证区域没有重叠。)
RelateProNum	2	N	为 0 时，由接下来的规则来决定动态区域关联了多少个异步节目一旦关联了某个异步节目，则当该异步节目播放时允许播放该动态区域，否则，不允许播放该动态区域 以下的节目编号根据 RelateProNum 的值来确定，当该值为 0 时不发送
RelateProSerial0	2		动态区域关联的第 0 个异步节目的编号
.....			
RelateProSerialN-1	2		动态区域关联的第 N-1 个异步节目的编号
ImmePlay	1		是否覆盖 该字节为 0 时，该动态区域与异步节目一起播放，该模式下， 注意：异步节目和动态区不能重叠，重叠情况下，某些特技组合会导致花屏。所以，要保证区域没有重叠。 该字节为 1 时，异步节目停止播

			放，仅播放该动态区域 注意： 当该字节为 0 时，RelateAllPro 到 RelateProSerialN-1 的参数才有效，否则无效 当该参数为 1 时，由于不与异步节目同时播放，为控制该动态区域能及时结束，可选择 RunMode 参数为 2 或 4，当然也可通过删除该区域来实现
Reserved	4	0x00	保留字节
AreaType	1	0x10	区域类型
AreaX	2		区域左上角横坐标(Top Left)，单位 Pixel
AreaY	2		区域左上角纵坐标(Top Left)，单位 Pixel
AreaWidth	2		区域宽度，单位 Pixel
AreaHeight	2		区域高度，单位 Pixel
AreaFrame	N		区域边框属性
BackGroundFlag	1		是否有背景、中间层背景 BIT0 为 1 时为最内层背景 BIT1 为 1 时为中间层背景
Transparency	1		透明度(参考背景区域)
AreaEqual	1	0x00	前景、背景区域大小是否相同 0x00: 相同 0x01: 不同
SoundFlag	1	0x00	是否使能语音播放 0 表示不使能语音 1 表示播放下文中 SoundData 部分内容
SoundPerson	1	0x00	发音人 该值范围是 0-5，共 6 种选择 只有 SoundFlag（是否使能语音播放）为 1 时才发送该字节，否则不发送 该值默认为 0
SoundVolum	1	0x05	音量 该值范围是 0~10，共 11 种，0 表示静音 只有 SoundFlag（是否使能语音播放）为 1 时才发送该字节，否则不发送 该值默认为 5
SoundSpeed	1	0x05	语速 该值范围是 0~10，共 11 种

			只有 SoundFlag （是否使能语音播放）为 1 时才发送该字节，否则不发送
			该值默认为 5
SoundDataMode	1	0x00	SoundData 的编码格式： 该值意义如下： 0x00 GB2312 0x01 GBK 0x02 BIG5 0x03 UNICODE
			只有 SoundFlag （是否使能语音播放）为 1 时才发送该字节，否则不发送
SoundReplayTimes	4	0x00000000	重播次数 该值为 0，表示播放 1 次 该值为 1，表示播放 2 次 该值为 0xffffffff，表示播放无限次
			只有 SoundFlag （是否使能语音播放）为 1 时才发送该字节，否则不发送
			该值默认为 0
SoundReplayDelay	4	0x00000000	重播时间间隔 该值表示两次播放语音的时间间隔，单位为 10ms
			只有 SoundFlag （是否使能语音播放）为 1 时才发送该字节，否则不发送
			该值默认为 0
SoundReservedParaLen	1	0x03	语音参数保留参数长度
Soundnumdeal	1	0	0: 自动判断 1: 数字作号码处理 2: 数字作数值处理 只有当 SoundFlag 为 1 且 SoundReservedParaLen 不为 0 才发送此参数
Soundlanguages	1	0	0: 自动判断语种 1: 阿拉伯数字、度量单位、特殊符号等合成为中文 2: 阿拉伯数字、度量单位、特殊符号等合成为英文 只有当 SoundFlag 为 1 且 SoundReservedParaLen 不为 0

			才发送此参数 (目前只支持中英文)
Soundwordstyle	1	0	0: 自动判断发音方式 1: 字母发音方式 2: 单词发音方式 只有当 SoundFlag 为 1 且 SoundReservedParaLen 不为 0 才发送此参数
SoundDataLen	4		语音数据长度
			只有 SoundFlag (是否使能语音 播放) 为 1 时才发送该字节, 否 则不发送
SoundData	N		语音数据
			只有 SoundFlag (是否使能语音 播放) 为 1 时才发送该字节, 否 则不发送
Reserved	5		保留字
以下为参数数据			
PageNum	2	0x0001 ~ 0xFFFF	数据页数, 此参数不能为 0
PageDataLen	4		每页的数据长度
PageStyle	1	0x00	数据页类型 0x00—文本、图片
DisplayMode	1		显示方式 0x00—随机显示 0x01—静止显示 0x02—快速打出 0x03—向左移动 0x04—向左连移 0x05—向上移动 0x06—向上连移 0x07—闪烁 0x08—飘雪 0x09—冒泡 0x0a—中间移出 0x0b—左右移入 0x0c—左右交叉移入 0x0d—上下交叉移入 0x0e—画卷闭合 0x0f—画卷打开 0x10—向左拉伸 0x11—向右拉伸 0x12—向上拉伸 0x13—向下拉伸 0x14—向左镭射 0x15—向右镭射 0x16—向上镭射 0x17—向下镭射

0x18 –左右交叉拉幕

0x19 –上下交叉拉幕

0x1a –分散左拉

0x1b –水平百页

0x1c –垂直百页

0x1d –向左拉幕

0x1e –向右拉幕

0x1f –向上拉幕

0x20 –向下拉幕

0x21 –左右闭合

0x22 –左右对开

0x23 –上下闭合

0x24 –上下对开

0x25 –向右移动

0x26 –向右连移

0x27 –向下移动

0x28 –向下连移

0x29 –45 度左旋

0x2a –180 度左旋

0x2b –90 度左旋

0x2c –45 度右旋

0x2d –180 度右旋

0x2e –90 度右旋

0x2f –菱形打开

0x30 –菱形闭合

ClearMode	1	退出方式/清屏方式
Speed	1	速度等级
StayTime	2	停留时间，单位为 10ms
RepeatTime	1	重复次数
ValidLen	2	此字段只在左移、右移方式下有效，默认值为区域宽度。 注：此字段使用时一定要和区域宽度进行一下比较，如果此字段大于区域宽度，则使用区域宽度
Reserved	9	保留字
DataOffset	4	点阵数据在该文件中的偏移量。 注：该字段从 0 开始，即第一个区域第一页偏移量为 0，其后区域在前一区域基础上累加
DataLen	4	本页数据的数据长度(不包括自身)
文本、图片数据，该部分放入所有参数结束后		
PicPageData	N	区域点阵数据，参考 点阵编码规则

5.4.2 删除动态区域区域信息

参数	数据长度	默认值	描述
RtnReq	1	0x01	要求返回
CmdGroup	1	0XA7	命令组
Cmd	1	0x01	命令编号
Reserved	2	0x00	保留字
DeleteAreaNum	1		要删除的区域个数 注意：如果该值为 0xFF，则删除所有动态区数据; 如果该值为 0x00，则不删除区域(如果是 0xFF，则 DeleteAreald 不需要发送)
DeleteAreald	N		需要删除的区域 ID 号 如果要删除的区域个数为 0，则该项不发送

5.4.3 更新语音队列命令

此命令更新整个队列，可以发送多条语音。并会删除控制器已经存在的语音队列。

参数	数据长度	默认值	描述
RtnReq	1	0x01	要求返回
CmdGroup	1	0XA7	命令组
Cmd	1	0x02	命令编号
Reserved	2	0x00	保留字节长度
StoreFlag	1	1	该值为 1 表示需要存储到 FLASH 中，掉电信息不丢失 该值为 0 表示需要存储到 RAM 中，掉电信息丢失
VoiceNum	1	1	更新语音队列个数

单个语音队列格式

VoiceID	1	1	语音队列 ID，从 0 开始
Datalen	4	0	数据长度（从 SoundPerson 到 SoundData 结束）
SoundPerson	1	0x00	发音人 该值范围是 0-5，共 6 种选择 该值默认为 0
SoundVolum	1	0x05	音量 该值范围是 0~10，共 11 种，0 表示静音 该值默认为 5
SoundSpeed	1	0x05	语速

			该值范围是 0~10，共 11 种 该值默认为 5
SoundDataMode	1	0x00	SoundData 的编码格式： 该值意义如下： 0x00 GB2312 0x01 GBK 0x02 BIG5 0x03 UNICODE
SoundReplayTimes	4	0x00000000	重播次数 该值为 0，表示播放 1 次 该值为 1，表示播放 2 次 该值为 0xffffffff，表示播放无限次 该值默认为 0
SoundReplayDelay	4	0x00000000	重播时间间隔 该值表示两次播放语音的时间间隔，单位为 10ms 该值默认为 0
SoundReservedParaLen	4	3	语音参数保留参数长度
Soundnumdeal	1	0	0: 自动判断 1: 数字作号码处理 2: 数字作数值处理 只有当 SoundReservedParaLen 不为 0 才发送此参数
Soundlanguages	1	0	0: 自动判断语种 1: 阿拉伯数字、度量单位、特殊符号等合成为中文 2: 阿拉伯数字、度量单位、特殊符号等合成为英文 只有当 SoundReservedParaLen 不为 0 才发送此参数
Soundwordstyle	1	0	0: 自动判断发音方式 1: 字母发音方式

		2: 单词发音方式
		只有当 SoundReservedParaLen 不为 0 才发送此参数
SoundDataLen	4	语音数据长度
SoundData	N	语音数据

5.4.4 插入队列命令

参数	数据长度	默认值	描述
RtnReq	1	0x01	要求返回
CmdGroup	1	0xA7	命令组
Cmd	1	0x07	命令编号
Reserved	2	0x00	保留字节
StoreFlag	1	0	该值为 1 表示需要存储到 FLASH 中，掉电信息不丢失
			该值为 0 表示需要存储到 RAM 中，掉电信息丢失
VoiceFlg	1	1	语音属性
			0: 此条语音从头插入队列，且停止当前正在播放的语音
			1: 此条语音从头插入队列，不停止当前播报的语音
			2: 此条语音从尾插入队列
Reserved	3	1	保留字节
Datelen	4	0	数据长度（从 SoundPerson 到 SoundData 结束）
SoundPerson	1	0x00	发音人
			该值范围是 0-5，共 6 种选择
			该值默认为 0
SoundVolum	1	0x05	音量
			该值范围是 0~10，共 11 种，0 表示静音
			该值默认为 5
SoundSpeed	1	0x05	语速
			该值范围是 0~10，共 11 种
			该值默认为 5

SoundDataMode	1	0x00	SoundData 的编码格式： 该值意义如下： 0x00 GB2312 0x01 GBK 0x02 BIG5 0x03 UNICODE
SoundReplayTimes	4	0x00000000	重播次数 该值为 0，表示播放 1 次 该值为 1，表示播放 2 次 该值为 0xffffffff，表示播放无限次 该值默认为 0
SoundReplayDelay	4	0x00000000	重播时间间隔 该值表示两次播放语音的时间间隔，单位为 10ms 该值默认为 0
SoundReservedParaLen	4	3	语音参数保留参数长度
Soundnumdeal	1	0	0：自动判断 1：数字作号码处理 2：数字作数值处理 只有当 SoundReservedParaLen 不为 0 才发送此参数
Soundlanguages	1	0	0：自动判断语种 1：阿拉伯数字、度量单位、特殊符号等合成为中文 2：阿拉伯数字、度量单位、特殊符号等合成为英文 只有当 SoundReservedParaLen 不为 0 才发送此参数
Soundwordstyle	1	0	0：自动判断发音方式 1：字母发音方式 2：单词发音方式 只有当 SoundReservedParaLen 不为 0 才

		发送此参数
SoundDataLen	4	语音数据长度
SoundData	N	语音数据

5.4.5 区域边框属性

参数	数据长度	默认值	描述
区域边框属性 (Area Frame)			
AreaFFlag	1	0x00	区域边框标志位 注：如果此字段为 0x00，则以下区域边框属性不发送
AreaFDispStyle	1	0x00	边框显示方式： 0x00 –闪烁 0x01 –顺时针转动 0x02 –逆时针转动 0x03 –闪烁加顺时针转动 0x04 –闪烁加逆时针转动 0x05 –红绿交替闪烁 0x06 –红绿交替转动 0x07 –静止打出
AreaFDispSpeed	1	0x01	边框显示速度
AreaFMoveStep	1	0x01	边框移动步长，单位为点，此参数范围为 1~16
AreaDLength	1		边框组员长度
AreaFWidth	1	0x01	边框组元宽度，此参数范围为 1~8 注：边框组元的长度固定为 16
AreaFBackup	8	0x00	备用字
AreaFUnitData	N		边框组员数据，格式同图文区

5.5 其它命令

5.5.1 Ping命令

参数	数据长度	默认值	描述
RtnReq	1	0x01	要求返回
CmdGroup	1	0xA2	命令组
Cmd	1	0x00	命令编号
Reserved	2	0x00	保留字

命令示例:

```
a5 a5 a5 a5 a5 a5 a5 a5 fe ff 00 80 f0 00 fe ff 00 00 00 00 05 00 00 00 01 A2 00 00 00 C6
86 5a
```


注意：该命令采用**统配符**，不针对特定控制器；可用于**简单判断控制器是否在线**(收到回包即可)

5.5.2 亮度调节命令

参数	数据长度	默认值	描述
RtnReq	1	0x01	要求返回
CmdGroup	1	0XA2	命令组
Cmd	1	0x07	命令编号
DataLen	2	0x31	数据长度
BrightnessMode	1		亮度调节模式 0x00---手动调节亮度 0x01---定时调亮(数据表域见下表格式，从 0 点开始，每半个小时一个亮度设定值，共 48 个) 注意：若为手动调 HalfHourValue0 至 HalfHourValue47 为一个值即可
HalfHourValue0	1	0x0f	00:00-----00:29 的亮度值,0x00----0x0f
HalfHourValue1	1	0x0f	00:30-----00:59 的亮度值,0x00----0x0f
.....			
HalfHourValue47	1	0x0f	23:30-----23:59 的亮度值,0x00----0x0f

当**支持亮度传感器**时其命令格式如下

参数	数据长度	默认值	描述
RtnReq	1	0x01	要求返回
CmdGroup	1	0XA2	命令组
Cmd	1	0x07	命令编号
DataLen	2	0x31	数据长度
BrightnessMode	1		亮度调节模式 0x02---默认调亮(需外接亮度传感器，此时控制器根据默认的亮度表值进行调亮) 0x03---自动调亮(需外接亮度传感器，此时控制器根据下述的亮度表值进行调亮)
Value0	2	0---65535	亮度等级 1 的环境亮度值
Value1	2	0---65535	亮度等级 2 的环境亮度值
.....			
Value15	2	0---65535	亮度等级 16 的环境亮度值
Value_0_level	1	0---15	亮度值 1 的亮度等级
Value_1_level	1	0---15	亮度值 2 的亮度等级
.....			
Value_15_level	1	0---15	亮度值 16 的亮度等级

建议支持**亮度传感器**时默认亮度表

参数	数据长度	默认值	描述
Value0	2	10(10 进制)	亮度等级 1 的环境亮度值

Value1	2	24(10 进制)	亮度等级 2 的环境亮度值
Value2	2	60(10 进制)	亮度等级 3 的环境亮度值
Value3	2	148(10 进制)	亮度等级 4 的环境亮度值
Value4	2	280(10 进制)	亮度等级 5 的环境亮度值
Value5	2	437(10 进制)	亮度等级 6 的环境亮度值
Value6	2	550(10 进制)	亮度等级 7 的环境亮度值
Value7	2	693(10 进制)	亮度等级 8 的环境亮度值
Value8	2	873(10 进制)	亮度等级 9 的环境亮度值
Value9	2	1100(10 进制)	亮度等级 10 的环境亮度值
Value10	2	1793(10 进制)	亮度等级 11 的环境亮度值
Value11	2	3336(10 进制)	亮度等级 12 的环境亮度值
Value12	2	6207(10 进制)	亮度等级 13 的环境亮度值
Value13	2	11551(10 进制)	亮度等级 14 的环境亮度值
Value14	2	21495(10 进制)	亮度等级 15 的环境亮度值
Value15	2	40000(10 进制)	亮度等级 16 的环境亮度值
Value_0_level	1	0(10 进制)	亮度值 1 的亮度等级
Value_1_level	1	1(10 进制)	亮度值 2 的亮度等级
Value_2_level	1	2(10 进制)	亮度值 3 的亮度等级
Value_3_level	1	3(10 进制)	亮度值 4 的亮度等级
Value_4_level	1	4(10 进制)	亮度值 5 的亮度等级
Value_5_level	1	5(10 进制)	亮度值 6 的亮度等级
Value_6_level	1	6(10 进制)	亮度值 7 的亮度等级
Value_7_level	1	7(10 进制)	亮度值 8 的亮度等级
Value_8_level	1	8(10 进制)	亮度值 9 的亮度等级
Value_9_level	1	9(10 进制)	亮度值 10 的亮度等级
Value_10_level	1	10(10 进制)	亮度值 11 的亮度等级
Value_11_level	1	11(10 进制)	亮度值 12 的亮度等级
Value_12_level	1	12(10 进制)	亮度值 13 的亮度等级
Value_13_level	1	13(10 进制)	亮度值 14 的亮度等级
Value_14_level	1	14(10 进制)	亮度值 15 的亮度等级
Value_15_level	1	15(10 进制)	亮度值 16 的亮度等级

5.5.3 亮度值查询命令

参数	数据长度	默认值	描述
RtnReq	1	0x01	要求返回
CmdGroup	1	0XA2	命令组
Cmd	1	0x19	命令编号
DataLen	2	0x0	数据长度

返回格式如下

参数	数据长度	默认值	描述
RtnReq	1	0x01	要求返回
CmdGroup	1	0XA2	命令组
Cmd	1	0x99	命令编号
Staus	2		控制器状态
Error	2		错误状态寄存器
DataLen	2	0x07	数据长度

BX128E002

CurrentBrightValue	4	当前亮度值
BrightnessMode	1	亮度调节模式
		注意：手动调亮、定时调亮时， CurrentBrightnessValue 字段无效
CurrentBrightness	1	当前亮度等级



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])
```

```
unsigned short 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  
};
```

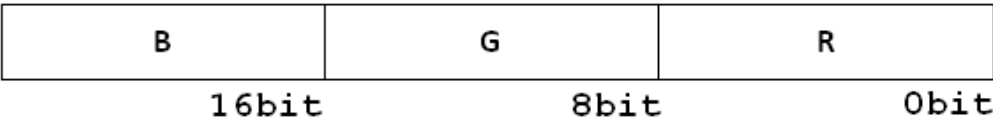
```
unsigned short  
CalcCRC(data, size)  
char    *data;  
int      size;  
{  
    int i;  
    unsigned short crc = 0;  
  
    for (i = 0; i < size; i++) {  
        crc = CRC(crc, data[i]);  
    }  
    return crc;  
}
```

附录2 点阵编码规则

1. BX-6Q 系列动态区编码规则

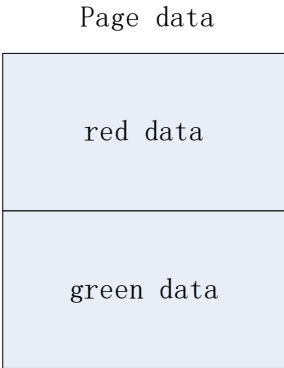
BX-6Q 系列采用 RGB888 模式，即一个像素用两个字节共 24bit 表示。像素由左到右，由上到下依次编码。

RGB888 编码格式如下：



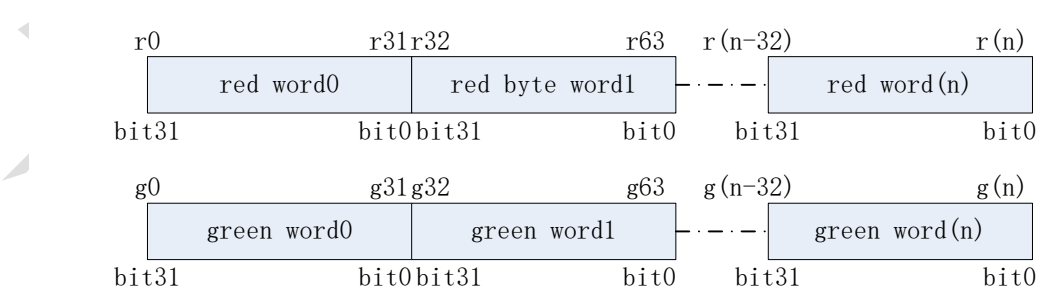
2. BX-6E 系列动态区编码规则

6 代协议将红绿数据分离。即先对红像素编码，再对绿像素编码。对于每一页数据，其组织方式为先红数据，后绿数据。如果是单色，则只有红数据。



编码时不是按照字节（byte）编码，而是按照字（word）编码。从区域的左上角开始，依次向右编码，到达区域最右端结束，然后再开始下一行的编码，依此循环。

如下图所示：



假定区域位于屏幕的左上角，编码时左上角的第一个红像素点（坐标为（0,0））对应第一个 word 的 bit 31,第二个红像素点（坐标为（1,0）），对应第一个 word 的 bit 30；以此类推。

当区域的起始坐标不是 32 的整数倍时，编码时需要对该区域最左边像素所对应的字节靠高位以 1 进行填充，例如某区域左上角起始坐标为（35,0），则该区域第一个 word 对应的二进制编码

应该是 111****。同理，当区域的结束坐标不是 32 的整数倍时，编码时需要对该区域最右边像素所对应的 word 靠低位以 1 进行填充。

