

# HB-F 多区域 LED 图文控制器通信协议

(V1.00)

## 一，数据通信方式规程

控制器与计算机之间通信物理接口：标准 RS232 串行通信口；

通信速率可选：9600bps、38400bps；

异步通讯数据帧结构定义如下：

起始位	数据位	校验位	停止位
1bits	8bits	无校验	1bits

## 二，帧结构及协议约定

计算机将显示屏信息发送给控制器，控制器根据相应的信息对显示屏进行操作。计算机向控制器发送信息的帧头为 2 字节（0X4842），控制器向计算机发送信息的帧头为 2 字节（0x4153），帧校验信息为 2 字节（CRC）。2 字节及 2 字节以上的数据结构均为高位在前低位在后。帧数据格式中所定义的帧长度含帧头信息和 CRC 校验。

在以下协议各帧中的【设备号】为 2 字节，当设为 0XFFFF 时为广播模式（广播模式下只验证卡类型不验证屏号，所有显示屏都接受上位机控制，数据接受完毕不返回信息）；

在以下协议中各帧中的【功能字】为 2 字节，高字节为命令号，低字节为卡类型号，关于命令号各帧都有注明，关于卡类型号：取值 1：表示 A 卡，取值 2：表示 B 卡，取值 3：表示 C 卡；

在以下协议中各帧中涉及时间信息的，均用十六进制表示。

### 三，通讯应答规定

计算机向控制器发送操作信息，控制器接收后须在 1 秒之内给出应答，若在 1 秒内无应答或应答错误，计算机继续重发一次信息，信息重发若超过 1 次，认为此次通讯出现故障。

### 四，命令号编码表

命令类别	命令号（16 进制）	功能描述
控制类	0x10	控制器连接
	0x11	设置控制器设备号
	0x12	复位控制器
	0x13	设定显示屏亮度
	0x14	设定时间段亮度控制
	0x15	校准控制器时间

	0x16	手动开/关屏幕
	0x17	设定自动开/关屏时间
	0x18	设定播放天数
	0x19	读取控制器运行状态
图片类	0x20	发送区域图片信息
动画类	0x30	发送区域动画信息
时间类	0x40	发送区域模拟时钟信息
	0x41	发送区域数字时钟信息
	0x42	发送区域日期信息

## 五，控制类协议内容

### 5.1 控制器连接（计算机→控制器）

帧数据格式如下：

帧头(2Byte)	帧长度(2Byte)	设备号（2 Byte）	功能字（2Byte）	备用（6Byte）	CRC（2Byte）
0x4842	16	0~65535	命令号：0x10	留用扩展	循环冗余校验

应答帧格式如下：（控制器→计算机）

帧头 (2Byte)	帧长度 (2Byte)	设备号 (2 Byte)	功能字 (2Byte)	屏行列数 (2Byte)	卡类型号 (1Byte)	CRC (2Byte)
0x4153	13	0~65535	命令号：0x10	高字节存行数 低字节存列数	1：A 型卡 2：B 型卡 3：C 型卡	循环冗余校验

注：屏行列数以字节为单位

## 5.2 设定控制器设备号（计算机→控制器）

帧数据格式如下：

帧头(2Byte)	帧长度(2Byte)	编号（2 Byte）	功能字（2Byte）	备用（6Byte）	CRC（2Byte）
0x4842	16	0~65535	命令号： 0x11	留用扩展	循环冗余校验

应答帧格式如下：（控制器→计算机）

帧头(2Byte)	帧长度(2Byte)	设备号（2 Byte）	功能字（2Byte）	标志（1Byte）	CRC（2Byte）
0x4153	11	0~65535	命令号： 0x11	1： 设置完成 0： 设置失败	循环冗余校验

## 5.3 复位控制器（计算机→控制器）

帧数据格式如下：

帧头(2Byte)	帧长度(2Byte)	设备号（2 Byte）	功能字（2Byte）	备用（6Byte）	CRC（2Byte）
0x4842	16	0~65535	命令号： 0x12	留用扩展	循环冗余校验

应答帧格式如下：（控制器→计算机）

帧头(2Byte)	帧长度(2Byte)	设备号（2 Byte）	功能字（2Byte）	标志（1Byte）	CRC（2Byte）
0x4153	11	0~65535	命令号：0x12	1：复位完成 0：复位失败	循环冗余校验

#### 5.4 设定显示屏亮度（计算机→控制器）

帧数据格式如下：

帧头 (2Byte)	帧长度 (2Byte)	设备号 (2 Byte)	功能字 (2Byte)	亮度值 (2 Byte)	备用 (4 Byte)	CRC (2Byte)
0x4842	16	0~65535	命令号：0x13	范围：0~7	留用扩展	循环冗余校验

注：亮度默认值为 4，最亮为 7，最暗为 0，存放在亮度值低字节中，高字节留用

应答帧格式如下：（控制器→计算机）

帧头(2Byte)	帧长度(2Byte)	设备号（2 Byte）	功能字（2Byte）	标志（1Byte）	CRC（2Byte）
0x4153	11	0~65535	命令号：0x13	1：设置完成 0：设置失败	循环冗余校验

### 5.5 设定时间段控制亮度（计算机→控制器）

帧数据格式如下：

帧头 (2Byte)	帧长度 (2Byte)	设备号 (2Byte)	功能字 (2Byte)	亮度值 (2Byte)	开始时间 (2Byte)	结束时间 (2Byte)	CRC (2Byte)
0x4842	16	0~65535	命令号： 0x14	0~7	依次为： 时，分	依次为： 时，分	循环 冗余校验

注：亮度默认值为 4，最亮为 7，最暗为 0，存放在亮度值低字节中，高字节留用

应答帧格式如下：（控制器→计算机）

帧头(2Byte)	帧长度(2Byte)	设备号 (2 Byte)	功能字 (2Byte)	标志 (1Byte)	CRC (2Byte)
0x4153	11	0~65535	命令号： 0x14	1: 设置完成 0: 设置失败	循环冗余校验

### 5.6 校准控制器时间（计算机→控制器）

帧数据格式如下：

帧头(2Byte)	帧长度(2Byte)	设备号 (2 Byte)	功能字 (2Byte)	校准时间 (7 Byte)	CRC (2Byte)
-----------	------------	--------------	-------------	---------------	-------------

0x4842	17	0~65535	命令号：0x15	依次为：年月日时分秒和星期	循环冗余校验
--------	----	---------	----------	---------------	--------

应答帧格式如下：（控制器→计算机）

帧头(2Byte)	帧长度(2Byte)	设备号（2 Byte）	功能字（2Byte）	标志（1Byte）	CRC（2Byte）
0x4153	10	0~65535	命令号：0x15	1：校准完成 0：校准失败	循环冗余校验

## 5.7 手动开/关屏幕（计算机→控制器）

帧数据格式如下：

帧头 (2Byte)	帧长度 (2Byte)	设备号 (2 Byte)	功能字 (2Byte)	开关标志 (1Byte)	备用 (5Byte)	CRC（2Byte）
0x4842	16	0~255	命令号：0x16	1：开，0：关	留用扩展	循环校验

应答帧格式如下：（控制器→计算机）



帧头(2Byte)	帧长度(2Byte)	设备号 (2 Byte)	功能字 (2Byte)	标志 (1Byte)	CRC (2Byte)
0x4153	11	0~65535	命令号: 0x16	1: 操作完成 0: 操作失败	循环冗余校验

## 5.8 设定自动开/关屏幕时间（计算机→控制器）

帧数据格式如下：

帧头 (2Byte)	帧长度 (2Byte)	设备号 (2Byte)	功能字 (2Byte)	开关标志 (1Byte)	开始时间 (3Byte)	结束时间 (3Byte)	CRC (2Byte)
0x4842	17	0~65535	命令号: 0x17	1: 开 0: 关	依次为: 时, 分, 秒	依次为: 时, 分, 秒	循环 冗余校验

应答帧格式如下：（控制器→计算机）

帧头(2Byte)	帧长度(2Byte)	设备号 (2 Byte)	功能字 (2Byte)	标志 (1Byte)	CRC (2Byte)
0x4153	11	0~65535	命令号: 0x17	1: 操作完成 0: 操作失败	循环冗余校验

### 5.9 设置播放天数（计算机→控制器）

帧数据格式如下：

帧头 (2Byte)	帧长度 (2Byte)	设备号 (2 Byte)	功能字 (2Byte)	设置天数 (1Byte)	备用 (5Byte)	CRC (2Byte)
0x4842	16	0~65535	命令号：0x18	0~255	留用扩展	循环校验

应答帧格式如下：（控制器→计算机）

帧头(2Byte)	帧长度(2Byte)	设备号 (2 Byte)	功能字 (2Byte)	标志 (1Byte)	CRC (2Byte)
0x4153	11	0~65535	命令号：0x18	1：设置完成 0：设置失败	循环冗余校验

### 5.10 读取控制器运行状态（计算机→控制器）

帧数据格式如下：

帧头(2Byte)	帧长度(2Byte)	编号 (2 Byte)	功能字 (2Byte)	备用 (6 Byte)	CRC (2Byte)
0x4842	16	0~65535	命令号：0x19	留用扩展	循环冗余校验

应答帧格式如下：（控制器→计算机）

帧头(2Byte)	帧长度(2Byte)	设备号（2 Byte）	功能字（2Byte）	状态字(1Byte)	屏行列数（2Byte）
0x4153	42	0~65535	命令号：0x19	D7-D3 保留 (1 为真，0 为假)	高字节存行数 低字节存列数

播放天数(2Byte)	开屏时间（2Byte）	关屏时间（2Byte）	亮度参数（2Byte）
高字节存设定天数 低字节存剩余天数	高字节存小时 低字节存分钟	高字节存小时 低字节存分钟	高字节存亮度范围（0-7） 低字节存当前亮度值

当前播放幕序号（8Byte）	当前时间（7 Byte）	备用（6 Byte）	CRC（2Byte）
分别存放 8 个分区的当前 播放屏幕序号；	依次为：年，月，日，时， 分秒（BCD 码）	留用扩展	循环冗余校验

注：状态字：D2:是否处于测试状态；D1:是否启用限时播放；D0:是否在播放时间段；屏行列数以字节为单位；

说明：在以下协议中，所述的【区域大小】、【区域坐标】包含的各参数均以点阵为单位，区域宽、高以及坐标 X、Y 等参数，均为 16 位整型数，取值范围 0~65535。

## 六，图片类协议内容

### 6.1 发送区域图片信息（计算机→控制器）

帧数据格式如下：

帧头 (2Byte)	帧长度 (2Byte)	设备号 (2 Byte)	功能字 (2Byte)	图片幕号 (1 Byte)	区域号 (1 Byte)	区域优先 (1 Byte)	区域大小 (4 Byte)
0x4842	38+X	0~65535	命令：0x20	0~255	0~255	0~255	依次为宽，高

区域坐标(4 Byte)	备用 (10 Byte)	播放模式 (7 Byte)	图片数据 (X Byte)	CRC (2Byte)
依次为 X,Y	留用扩展	依次为： 1. 进入模式 (0~15) 2. 进入速度 (0~15) 3. 停留时间 (0~127) 4. 退出模式 (0~15) 5. 退出速度 (0~15)	X=宽度*高度*1 Bbyte	循环冗余校验

		6. 跟随前屏 (0~1) 7. 颜色标志 (0:红;1:绿;2:黄)		
--	--	--	--	--

注：X 为一幕图片的字节数；区域优先级取值越低，该区域的数据越先被刷新，如果两个区域出现重叠，关于重叠区域的显示，优先级高的那个区域会覆盖优先级的区域。

应答帧格式如下：（控制器→计算机）

帧头(2Byte)	帧长度(2Byte)	设备号 (2 Byte)	功能字 (2Byte)	标志 (1Byte)	CRC (2Byte)
0x4153	11	0~65535	命令号：0x20	1：接收完成 0：接收失败	循环冗余校验

## 七，动画类协议内容

### 7.1 发送区域动画信息（计算机→控制器）

帧数据格式如下：

帧头 (2Byte)	帧长度 (2Byte)	设备号 (2 Byte)	功能字 (2Byte)	区域号 (1 Byte)	区域优先级 (1 Byte)	区域大小 (4 Byte)	区域坐标 (4 Byte)
0x4842	32+X	0~65535	命令：0x30	0~255	0~255	依次为宽，高	依次为 X,Y

动画帧号（2 Byte）	播放速度（1 Byte）	备用（9 Byte）	动画数据（X Byte）	CRC（2Byte）
0~65535	0~255	留用扩展	1 帧动画的字节数	循环冗余校验

应答帧格式如下：（控制器→计算机）

帧头(2Byte)	帧长度(2Byte)	设备号（2 Byte）	功能字（2Byte）	标志（1Byte）	CRC（2Byte）
0x4153	11	0~65535	命令号：0x30	1：接收完成 0：接收失败	循环冗余校验

## 八，时间类协议内容

### 8.1 发送区域模拟时钟信息（计算机→控制器）

帧数据格式如下：

帧头 (2Byte)	帧长度 (2Byte)	设备号 (2 Byte)	功能字 (2Byte)	区域号 (1 Byte)	区域优先级 (1 Byte)	区域大小 (4 Byte)	区域坐标 (4 Byte)
0x4842	36+X	0~65535	命令：0x40	0~255	0~255	依次为宽，高	依次为 X,Y

表盘坐标 (4 Byte)	表针颜色 (3 Byte)	备用 (9 Byte)	模拟表盘数据 (X Byte)	CRC (2Byte)
依次为 X,Y	依次为：时，分，秒	留用扩展	$X = (\text{区域宽} * \text{区域高}) / 8$	循环冗余校验

注：表针颜色 1：红色； 2：绿色； 3：黄色；

应答帧格式如下：（控制器→计算机）

帧头(2Byte)	帧长度(2Byte)	设备号 (2 Byte)	功能字 (2Byte)	标志 (1Byte)	CRC (2Byte)
0x4153	10	0~65535	命令号： 0x40	1：接收完成 0：接收失败	循环冗余校验

## 8.2 发送区域数字时钟信息（计算机→控制器）

帧数据格式如下：

帧头	帧长度	设备号	功能字	区域号	区域优先级	区域大小	区域坐标
(2Byte)	(2Byte)	(2 Byte)	(2Byte)	(1 Byte)	(1 Byte)	(4 Byte)	(4 Byte)
0x4842	30+A+B	0~255	命令： 0x41	0~255	0~255	依次为宽，高	依次为 X,Y

时、分、秒 字库 (A Byte)	0~9 字符字库 (B Byte)	备用 (10 Byte)	CRC (2Byte)
A 为三个汉字的字节数	B 为 10 个字符字库的字节数	留用扩展	循环冗余校验

应答帧格式如下：（控制器→计算机）

帧头 (2Byte)	帧长度 (2Byte)	设备号 (2 Byte)	功能字 (2Byte)	标志 (1Byte)	CRC (2Byte)
0x4153	11	0~65535	命令：0x41	1: 接收完成 0: 接收失败	循环冗余校验

## 8.2 发送区域日期信息（计算机→控制器）

帧数据格式如下：

帧头 (2Byte)	帧长度 (2Byte)	设备号 (2 Byte)	功能字 (2Byte)	区域号 (1 Byte)	区域优先级 (1 Byte)	区域大小 (4 Byte)	区域坐标 (4 Byte)
0x4842	30+A+B	0~65535	命令：0x42	0~255	0~255	依次为宽，高	依次为 X,Y



年、月、日 字库 (A Byte)	0~9 字符字库 (B Byte)	备用 (10 Byte)	CRC (2Byte)
A 为三个汉字的字节数	B 为 10 个字符字库的字节数	留用扩展	循环冗余校验

应答帧格式如下：（控制器→计算机）

帧头 (2Byte)	帧长度 (2Byte)	设备号 (2 Byte)	功能字 (2Byte)	标志 (1Byte)	CRC (2Byte)
0x4153	11	0~65535	命令号: 0x42	1: 接收完成 0: 接收失败	循环冗余校验

## 附：CRC16 校验程序

```
#define xcrc(crc,cp) (crctab[(((crc>>8)&0xff)^(cp&0xff))^(crc<<8))] //校验计算公式
```

```
unsigned int code crctab[256] =
```

```
{
```

```
0x0000, 0x1021, 0x2042, 0x3063, 0x4084, 0x50A5, 0x60C6, 0x70E7,
```

0x8108, 0x9129, 0xA14A, 0xB16B, 0xC18C, 0xD1AD, 0xE1CE, 0xF1EF,  
0x1231, 0x0210, 0x3273, 0x2252, 0x52B5, 0x4294, 0x72F7, 0x62D6,  
0x9339, 0x8318, 0xB37B, 0xA35A, 0xD3BD, 0xC39C, 0xF3FF, 0xE3DE,  
0x2462, 0x3443, 0x0420, 0x1401, 0x64E6, 0x74C7, 0x44A4, 0x5485,  
0xA56A, 0xB54B, 0x8528, 0x9509, 0xE5EE, 0xF5CF, 0xC5AC, 0xD58D,  
0x3653, 0x2672, 0x1611, 0x0630, 0x76D7, 0x66F6, 0x5695, 0x46B4,  
0xB75B, 0xA77A, 0x9719, 0x8738, 0xF7DF, 0xE7FE, 0xD79D, 0xC7BC,  
0x48C4, 0x58E5, 0x6886, 0x78A7, 0x0840, 0x1861, 0x2802, 0x3823,  
0xC9CC, 0xD9ED, 0xE98E, 0xF9AF, 0x8948, 0x9969, 0xA90A, 0xB92B,  
0x5AF5, 0x4AD4, 0x7AB7, 0x6A96, 0x1A71, 0x0A50, 0x3A33, 0x2A12,  
0xDBFD, 0xCBDC, 0xFBBF, 0xEB9E, 0x9B79, 0x8B58, 0xBB3B, 0xAB1A,  
0x6CA6, 0x7C87, 0x4CE4, 0x5CC5, 0x2C22, 0x3C03, 0x0C60, 0x1C41,  
0xEDAE, 0xFD8F, 0xCDEC, 0xDDCD, 0xAD2A, 0xBD0B, 0x8D68, 0x9D49,

0x7E97, 0x6EB6, 0x5ED5, 0x4EF4, 0x3E13, 0x2E32, 0x1E51, 0x0E70,  
0xFF9F, 0xEFBE, 0xDFDD, 0xCFFC, 0xBF1B, 0xAF3A, 0x9F59, 0x8F78,  
0x9188, 0x81A9, 0xB1CA, 0xA1EB, 0xD10C, 0xC12D, 0xF14E, 0xE16F,  
0x1080, 0x00A1, 0x30C2, 0x20E3, 0x5004, 0x4025, 0x7046, 0x6067,  
0x83B9, 0x9398, 0xA3FB, 0xB3DA, 0xC33D, 0xD31C, 0xE37F, 0xF35E,  
0x02B1, 0x1290, 0x22F3, 0x32D2, 0x4235, 0x5214, 0x6277, 0x7256,  
0xB5EA, 0xA5CB, 0x95A8, 0x8589, 0xF56E, 0xE54F, 0xD52C, 0xC50D,  
0x34E2, 0x24C3, 0x14A0, 0x0481, 0x7466, 0x6447, 0x5424, 0x4405,  
0xA7DB, 0xB7FA, 0x8799, 0x97B8, 0xE75F, 0xF77E, 0xC71D, 0xD73C,  
0x26D3, 0x36F2, 0x0691, 0x16B0, 0x6657, 0x7676, 0x4615, 0x5634,  
0xD94C, 0xC96D, 0xF90E, 0xE92F, 0x99C8, 0x89E9, 0xB98A, 0xA9AB,  
0x5844, 0x4865, 0x7806, 0x6827, 0x18C0, 0x08E1, 0x3882, 0x28A3,  
0xCB7D, 0xDB5C, 0xEB3F, 0xFB1E, 0x8BF9, 0x9BD8, 0xABBB, 0xBB9A,

```

0x4A75, 0x5A54, 0x6A37, 0x7A16, 0x0AF1, 0x1AD0, 0x2AB3, 0x3A92,
0xFD2E, 0xED0F, 0xDD6C, 0xCD4D, 0xBDAA, 0xAD8B, 0x9DE8, 0x8DC9,
0x7C26, 0x6C07, 0x5C64, 0x4C45, 0x3CA2, 0x2C83, 0x1CE0, 0x0CC1,
0xEF1F, 0xFF3E, 0xCF5D, 0xDF7C, 0xAF9B, 0xBFBA, 0x8FD9, 0x9FF8,
0x6E17, 0x7E36, 0x4E55, 0x5E74, 0x2E93, 0x3EB2, 0x0ED1, 0x1EF0
};

```

```

/*****加 CRC 校验字*****/

void ConCRC(uchar n)          //n 为不包括校验字的串总长度
{
    unsigned int crc16;
    uchar i;                  //变量

```

```

    crc16=0;

    for(i=0;i<n;i++)                //n 个数据的 CRC 校验

        crc16=xcrc(crc16,U1_T_buf[i]);

    U1_T_buf[i]=(unsigned char)(crc16>>8);

    U1_T_buf[i+1]=(unsigned char)(crc16&0xff);

}

```

/\*\*\*\*\*CRC 数据验证\*\*\*\*\*/

```

uchar DataCRC(uchar count)        //接收数据总长度

{
    unsigned int crc16;

    uchar i;

    crc16=0;

```

```
for(i=0;i<(count-2);i++)  
  
crc16=xcrc(crc16,U1_R_buf[i]);  
  
if((U1_R_buf[count-2]==(unsigned char)(crc16>>8))  
  
    && (U1_R_buf[count-1]==(unsigned char)(crc16&0xff)))  
  
    return 0;                                //校验正确，返回 0  
  
else    return 1;                            //校验错误，返回 1  
  
}
```

**2009.08.26**