

# 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

- 1，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 2 字节 CRC 校验。
- 2，在以下协议各帧中的【设备编号】为 2 字节，当设备号为 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)；
- 3，在以下协议各帧中的【设备类别】为 1 字节，取值范围暂定 1~5，当取值为 1，表示 A 卡；取值为 2，表示 B 卡；取值为 3 表示 C 卡；取值为 4 表示 D 卡，取值为 5 表示 E 卡。
- 4，在以下协议中各帧中涉及时间信息的，均用十六进制表示。
5. 由于控制器串口通讯采用查询模式，计算机发信息前先发【控制器连接】数据帧，待接收到控制器应

答后，再发送信息；当信息发送完后，再发【断开控制器连接】数据帧，告知控制器信息发送结束。

### 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，命令号编码索引表

命令编码分为两大类：1.控制类，2.数据类，其中数据类，包括图片、动画、时间等数据。

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
	0x1A	设置屏幕宽高
	0x1B	数据发送起始帧
	0x1C	数据发送结束帧
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	发送区域时间信息

## 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)	设备类别 (1Byte)	设备编号 (2 Byte)	命令号 (1Byte)	留用扩展 (6Byte)	CRC (2Byte)
0x4842	16	1~5	0~65535	0x10	(默认 0x00)	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 (1Byte)	设备类别 (1Byte)	设备编号 (2 Byte)	命令号 (1Byte)	屏行列数 (2Byte)	CRC (2Byte)
0x4153	11	1~5	0~65535	0x10	依次为行数、列数	循环冗余校验

注：屏行列数以字节为单位，即屏行列点阵数除以 8

### 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)	设备类别 (1Byte)	设备编号 (2 Byte)	命令号 (1Byte)	留用扩展 (6Byte)	CRC (2Byte)
---------------	----------------	-----------------	------------------	----------------	-----------------	-------------

0x4842	16	1~5	0~65535	0x11	(默认 0x00)	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 (1Byte)	设备类别 (1Byte)	设备编号 (2 Byte)	命令号 (1Byte)	返回标志 (1Byte)	CRC (2Byte)
0x4153	10	1~5	0~65535	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)	设备类别 (1Byte)	设备编号 (2 Byte)	命令号 (1Byte)	留用扩展 (6Byte)	CRC (2Byte)
0x4842	16	1~5	0~65535	0x12	(默认 0x00)	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 (1Byte)	设备类别 (1Byte)	设备编号 (2 Byte)	命令号 (1Byte)	返回标志 (1Byte)	CRC (2Byte)
---------------	----------------	-----------------	------------------	----------------	-----------------	-------------

0x4153	10	1~5	0~65535	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)	设备类别 (1Byte)	设备编号 (2 Byte)	命令号 (1Byte)	亮度值 (1 Byte)	留用扩展 (5 Byte)	CRC (2Byte)
0x4842	16	1~5	0~65535	0x13	范围：0~7	(默认 0x00)	循环冗余校验

注：亮度默认值为 7，最亮为 7，最暗为 0。

应答帧格式如下：（控制器→计算机）

帧头 (2Byte)	帧长度 (1Byte)	设备类别 (1Byte)	设备编号 (2 Byte)	命令号 (1Byte)	返回标志 (1Byte)	CRC (2Byte)
0x4153	10	1~5	0~65535	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)	设备类别 (1Byte)	设备编号 (2Byte)	命令号 (1Byte)	亮度值 (1Byte)	开始时间 (2Byte)	结束时间 (2Byte)	留用扩展 (1Byte)	CRC (2Byte)
0x4842	16	1~5	0~65535	0x14	0~7	依次为： 时，分	依次为： 时，分	(默认 0x00)	循环 冗余校验

注：亮度默认值为 7，最亮为 7，最暗为 0。

应答帧格式如下：（控制器→计算机）

帧头 (2Byte)	帧长度 (1Byte)	设备类别 (1Byte)	设备编号 (2 Byte)	命令号 (1Byte)	返回标志 (1Byte)	CRC (2Byte)
0x4153	10	1~5	0~65535	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)	设备类别 (1Byte)	设备编号 (2 Byte)	命令号 (1Byte)	校准时间 (7 Byte)	CRC (2Byte)
---------------	----------------	-----------------	------------------	----------------	------------------	-------------

0x4842	17	1~5	0~65535	0x15	依次：年月日 星期时分秒	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 (1Byte)	设备类别 (1Byte)	设备编号 (2 Byte)	命令号 (1Byte)	返回标志 (1Byte)	CRC (2Byte)
0x4153	10	1~5	0~65535	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)	设备类别 (1Byte)	设备编号 (2 Byte)	命令号 (1Byte)	开关标志 (1Byte)	留用扩展 (5 Byte)	CRC (2Byte)
0x4842	16	1~5	0~65535	0x16	1: 开, 0: 关	(默认 0x00)	循环校验

应答帧格式如下：（控制器→计算机）

帧头 (2Byte)	帧长度 (1Byte)	设备类别 (1Byte)	设备编号 (2 Byte)	命令号 (1Byte)	返回标志 (1Byte)	CRC (2Byte)
---------------	----------------	-----------------	------------------	----------------	-----------------	-------------



0x4153	10	1~5	0~65535	0x16	1: 操作完成 0: 操作失败	循环校验
--------	----	-----	---------	------	--------------------	------

## 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)	设备类别 (1Byte)	设备编号 (2Byte)	命令号 (1Byte)	开关标志 (1Byte)	开始时间 (2Byte)	结束时间 (2Byte)	留用扩展 (1Byte)	CRC (2Byte)
0x4842	16	1~5	0~65535	0x17	1: 开 0: 关	依次为: 时, 分	依次为: 时, 分	(默认 0x00)	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 (1Byte)	设备类别 (1Byte)	设备编号 (2Byte)	命令号 (1Byte)	返回标志 (1Byte)	CRC (2Byte)
0x4153	10	1~5	0~65535	0x17	1: 操作完成 0: 操作失败	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)	设备类别 (1Byte)	设备编号 (2 Byte)	命令号 (1Byte)	设置天数 (2Byte)	留用扩展 (4Byte)	CRC (2Byte)
0x4842	16	1~5	0~65535	0x18	0~65535	(默认 0x00)	冗余校验

注：设置播放天数，默认 0xFFFF,为无限期播放,最大有效值为 9999。

应答帧格式如下：（控制器→计算机）

帧头 (2Byte)	帧长度 (1Byte)	设备类别 (1Byte)	设备编号 (2 Byte)	命令号 (1Byte)	返回标志 (1Byte)	CRC (2Byte)
0x4153	10	1~5	0~65535	0x18	1: 设置完成 0: 设置失败	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)	设备类别 (1Byte)	设备编号 (2 Byte)	命令号 (1Byte)	留用扩展 (6Byte)	CRC (2Byte)
0x4842	16	1~5	0~65535	0x19	(默认 0x00)	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 (1Byte)	设备类别 (1Byte)	设备编号 (2 Byte)	命令号 (1Byte)	状态字 (1Byte)	屏行列数 (2Byte)
0x4153	42	1~5	0~65535	0x19	D7-D3 保留 (1 为真, 0 为假)	依次:行数、列数

播放天数(4Byte)	开屏时间 (2Byte)	关屏时间 (2Byte)	亮度参数 (1Btye)
前 2 字节存设定天数 后 2 字节存剩余天数	依次为小时、分钟	依次为小时、分钟	当前亮度值 (0-7)

当前播放幕序号 (8Byte)	当前时间 (7 Byte)	留用扩展(6 Byte)	CRC (2Byte)
依次存放 8 个分区的当前 播放屏幕序号；顺序为： 从屏幕的左边到右边	依次为:年，月，星期，日， 时，分秒	(默认 0x00)	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，即屏行列点阵数除以 8。

### 5.11 设置屏幕宽高（计算机→控制器）

帧数据格式如下：

帧头 (2Byte)	帧长度 (2Byte)	设备类别 (1Byte)	设备编号 (2 Byte)	命令号 (1Byte)	屏幕宽高 (2Byte)	留用扩展 (4Byte)	CRC (2Byte)
0x4842	16	1~5	0~65535	0x1A	依次为宽高	(默认 0x00)	冗余校验

注：屏幕宽高以字节为单位，即屏幕宽高点阵数除以 8

应答帧格式如下：（控制器→计算机）

帧头 (2Byte)	帧长度 (1Byte)	设备类别 (1Byte)	设备编号 (2 Byte)	命令号 (1Byte)	返回标志 (1Byte)	CRC (2Byte)
0x4153	10	1~5	0~65535	0x1A	1: 设置完成 0: 设置失败	冗余校验

### 5.12 数据发送起始帧（计算机→控制器）

帧数据格式如下：

帧头 (2Byte)	帧长度 (2Byte)	设备类别 (1Byte)	设备编号 (2 Byte)	命令号 (1Byte)	留用扩展 (6Byte)	CRC (2Byte)
---------------	----------------	-----------------	------------------	----------------	-----------------	----------------

0x4842	16	1~5	0~65535	0x1B	(默认 0x00)	循环冗余校验
--------	----	-----	---------	------	-----------	--------

注：该帧作用是通知控制器，开始发送图片或动画或时间等类数据

应答帧格式如下：（控制器→计算机）

帧头 (2Byte)	帧长度 (1Byte)	设备类别 (1Byte)	设备编号 (2 Byte)	命令号 (1Byte)	返回标志 (1Byte)	CRC (2Byte)
0x4153	10	1~5	0~65535	0x1B	1: 操作完成 0: 操作失败	冗余校验

### 5.13 数据发送结束帧（计算机→控制器）

帧数据格式如下：

帧头 (2Byte)	帧长度 (2Byte)	设备类别 (1Byte)	设备编号 (2 Byte)	命令号 (1Byte)	留用扩展 (6Byte)	CRC (2Byte)
0x4842	16	1~5	0~65535	0x1C	(默认 0x00)	循环冗余校验

注：该帧作用是通知控制器，图片或动画或时间等类数据已发送结束

应答帧格式如下：（控制器→计算机）

帧头	帧长度	设备类别	设备编号	命令号	返回标志	CRC
----	-----	------	------	-----	------	-----

(2Byte)	(1Byte)	(1Byte)	(2 Byte)	(1Byte)	(1Byte)	(2Byte)
0x4153	10	1~5	0~65535	0x1C	1: 操作完成 0: 操作失败	冗余校验

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，传输及使用均为 8 的整倍数。

## 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)	设备类别 (1Byte)	设备编号 (2 Byte)	命令号 (1Byte)	同类型 区域编号 (1 Byte)	区域 优先级 (1 Byte)	区域坐标 (4 Byte)	区域大小 (4 Byte)
0x4842	40+X	1~5	0~65535	0x20	0~255	0~255	依次为 X,Y	依次为 宽，高

总幕数 (2 Byte)	当前幕号 (2 Byte)	备用 (9 Byte)	播放模式 (7 Byte)	图片数据 (X Byte)	CRC (2Byte)
0~65535	0~65535	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) 6. 跟随前屏 (0~1) 7. 颜色标志 (1:单色;2:双色;3:彩色)	X=宽度*高度*颜色标志 (字节按以前既定方式排 放)	循环 冗余校验

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 (1Byte)	设备编号 (2 Byte)	设备类别 (1Byte)	命令号 (1Byte)	返回标志 (1Byte)	CRC (2Byte)
0x4153	10	0~65535	1~5	0x20	1: 接收完成 0: 接收失败	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)	设备类别 (1Byte)	设备编号 (2 Byte)	命令号 (1Byte)	同类型 区域编号 (1 Byte)	区域 优先级 (1 Byte)	区域坐标 (4 Byte)	区域大小 (4 Byte)
0x4842	30+X	1~5	0~65535	0x30	0~255	0~255	依次为 X,Y	依次为宽, 高

动画总帧数 (2 Byte)	当前帧号 (2 Byte)	播放速度 (1 Byte)	备用 (5 Byte)	动画数据 (X Byte)	CRC (2Byte)
0~65535	0~65535	默认 10 (0~255)	留用扩展	1 帧动画的字节数	循环冗余校验

注：播放速度指每秒钟播放多少帧图片。

应答帧格式如下：（控制器→计算机）

帧头 (2Byte)	帧长度 (1Byte)	设备类别 (1Byte)	设备编号 (2 Byte)	命令号 (1Byte)	返回标志 (1Byte)	CRC (2Byte)
0x4153	10	1~5	0~65535	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)	设备类别 (1Byte)	设备编号 (2 Byte)	命令号 (1Byte)	同类型 区域编号 (1 Byte)	区域优先级 (1 Byte)	区域坐标 (4 Byte)	区域大小 (4 Byte)
0x4842	30+X	1~5	0~65535	0x40	0~255	0~255	依次为 X,Y	依次为宽，高

参数长度 (2 Byte)	表盘坐标 (4 Byte)	表针颜色 (3 Byte)	备用 (1 Byte)	模拟表盘数据 (X Byte)	CRC (2Byte)
包括：表盘坐标、表针颜色、备用字节、模拟表盘数等参数的字节总数	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，其它颜色待定；表盘坐标指表盘的圆心坐标

应答帧格式如下：（控制器→计算机）

帧头(2Byte)	帧长度	设备类别	设备编号	命令号	返回标志	CRC (2Byte)
-----------	-----	------	------	-----	------	-------------

	(1Byte)	(1Byte)	(2 Byte)	(1Byte)	(1Byte)	
0x4153	10	1~5	0~65535	0x40	1: 接收完成 0: 接收失败	循环冗余校验

## 8.2 发送区域时间信息（计算机→控制器）

帧数据格式如下：

帧头 (2Byte)	帧长度 (2Byte)	设备类别 (1Byte)	设备编号 (2 Byte)	命令号 (1Byte)	同类型 区域编号 (1 Byte)	区域优先级 (1 Byte)	区域坐标 (4 Byte)	区域大小 (4 Byte)
0x4842	40	1~5	0~65535	0x41	0~255	0~255	依次为 X,Y	依次为宽，高

参数长度 (2 Byte)	字符模式 (1 Byte)	行显模式 (1 Byte)	日期选择 (1 Byte)	星期选择 (1 Byte)	时间选择 (1 Byte)
包括：模式选择标志、备用字节、字库等参数的总字节数	0: 字符显示 1: 汉字显示	0: 单行显示 1: 多行显示	0: 不显示 1: 显示(年为 2 位) 2: 显示(年为 4 位)	0:不显示 1: 显示	0: 不显示 1: 显示

备用 (13 Byte)	“年”、“月”、“日”、“时”、“分”、“秒”、“星”、“期”、“一”、“二”、“三”、“四”、“五”、“六”等字的字库 (A Byte)	0~9 字符字库 (B Byte)	CRC (2Byte)
留用扩展	A 为 14 个汉字的占用的字节数	B 为 10 个字符字库的占用的字节数	循环冗余校验

注：年格式为 4 位 (20XX)，例如：2009 年；格式为 2 位 (XX)，例如：09 年。

应答帧格式如下：（控制器→计算机）

帧头 (2Byte)	帧长度 (1Byte)	设备类别 (1Byte)	设备编号 (2 Byte)	命令号 (1Byte)	返回标志 (1Byte)	CRC (2Byte)
0x4153	10	1~5	0~65535	0x41	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
}

```

**2009.09.03**