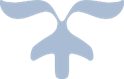


LCD\_V1.0与上位机通讯协议

(ASCII码 通信协议)

20181106

上海韦乐海茨医药有限公司



目录

[一、通讯原理 4](#_Toc529262757)

[二、液晶显示原理 4](#_Toc529262758)

[三、 通讯数据格式 5](#_Toc529262759)

[四、通讯协议详细说明 6](#_Toc529262760)

[1、上位机发送模板指令的具体协议和下位机响应说明如下： 6](#_Toc529262761)

[（1）上位机发送模板指令（ASCII格式）（指令代码T（0x54）） 6](#_Toc529262762)

[（2）上位机响应通讯信息格式如下： 7](#_Toc529262763)

[（3）以下是单条模板发送举例： 7](#_Toc529262764)

[（4）以下是模板打包发送举例： 7](#_Toc529262765)

[2、上位机发送默认模板数据指令的具体协议和下位机响应说明如下： 8](#_Toc529262766)

[（1）上位机发送模板指令（ASCII格式）（指令代码M（0x4D）） 8](#_Toc529262767)

[（2）上位机响应通讯信息格式如下： 9](#_Toc529262768)

[（3）以下是单条默认模板数据发送举例 9](#_Toc529262769)

[（4）以下是默认模板数据打包发送举例： 9](#_Toc529262770)

[3、上位机发送数据指令的具体协议和下位机响应说明如下： 10](#_Toc529262771)

[（1）以下是单条数据发送举例： 10](#_Toc529262772)

[（2）以下是数据打包发送举例： 10](#_Toc529262773)

[4、上位机发送保存数据指令的具体协议和下位机响应说明如下： 11](#_Toc529262774)

[（1）上位机发送保存数据指令（ASCII格式）（指令代码S（0x53）） 11](#_Toc529262775)

[（2）上位机响应通讯信息格式如下： 12](#_Toc529262776)

[5、上位机发送特殊指令的具体协议和下位机响应说明如下： 13](#_Toc529262777)

[（1）特殊指令1 14](#_Toc529262778)

[（2）特殊指令2 14](#_Toc529262779)

[（3）特殊指令3 14](#_Toc529262780)

[（4）特殊指令4 14](#_Toc529262781)

[（5）特殊指令5 15](#_Toc529262782)

[（6）特殊指令6 15](#_Toc529262783)

[（7）特殊指令10 15](#_Toc529262784)

# 一、通讯原理

下位机和上位机（PC）通讯使用RS485自定义通讯模式，PC为主机，单片机为从机，由主机发送指令数据，从机执行并回应。数据校验使用标准CRC16-CCITT校验协议。

# 二、液晶显示原理

液晶屏显示一个字符正常需要知道这个字符显示的大小（字号），显示X轴起始位置，显示Y轴起始位置，背景颜色，字体颜色，如果一条指令需要显示多个字符还需要知道显示的字节宽度，如果每次上位机发送数据显示都要包含这些元素，特别是要显示多个不同颜色不同位置的字符时，通讯的数据量将比较大，因此构建模板的目的是将相同背景颜色，字体颜色的一串字符使用模板在显示数据通讯前就定义好格式，通讯过程中只需要将显示的数据的ASCII码发送过去就可以完成显示。

本通讯协议主要分为五个部分，分别如下：

1. 第一部分是模板指令（‘T’指令），主要是发送模板，将正常通讯时需要的字体大小，位置，颜色等保护在模板里面，只在产品使用前发送一次，此模板会保存在单片机的eeprom中，每次开机后都会先执行模板检测和下载，没有模板将不能显示任何信息。
2. 第二部分是默认模板数据指令（‘M’指令），因为模板只是包含一些显示规则，只有模板上电之后显示屏是不会显示任何字符的，因此需要构建默认模板数据，默认模板数据就是液晶屏每次上电后显示的固定信息，比如每次上电都需要显示公司名称，那么公司名称就可以做成默认模板数据存储，如果不需要更改公司名称的话，这部分数据只是需要在产品使用前发送一次。默认模板数据也会保存在单片机的eeprom中，在产品使用前必须拥有默认，没有默认模板数据也不能显示任何信息，如果全屏都没有任何信息需要在开机后显示，请将默认模板数据全部写入“空格”，空格是不显示的字符。
3. 第三部分是数据指令（包括‘D’指令和‘S’指令），当液晶屏已经正确写入模板和默认模板数据之后，通讯的时候就可以只发送字符的ASCII就可以在相应位置显示你想要显示的信息，D指令发送数据显示完就不会再显示，此部分数据是不存储的数据，D指令发送显示完就丢弃。
4. 第四部分是保存数据指令（‘S’指令），S指令是多条数据显示指令，发送的时候先显示第一条，然后根据通讯指令自动显示第二条或者手动按键后显示第二条，第三条类似。S指令是暂存指令，除非掉电或上位机更改，否则会暂存在单片机的缓存中，掉电不保持。
5. 第四部分是特殊指令（‘C’指令），这部分指令是由上位机控制下位机的执行一些特殊的行为，比如在使用过程中更换模板和默认模板数据，需要发特殊指令刷新模板和默认模板数据（否则需要断电重启），擦除eeprom内部的模板和默认模板数据等，还可以控制LED灯的显示和显示方式，S指令数据是否循环显示和循环显示方式等，具体后面详细说明。

# 通讯数据格式

1、上位机发送通讯数据格式如下，具体通讯协议需要根据相应命令细分，在后面说明：

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 通讯包  起始（2byte） | | 485地址(1byte) | 指令代码（1byte） | 通讯命令1（1byte） | 通讯命令  2（1byte） | 数据长度（大端模式）  (2byte) | 数据  (N byte) | CRC16\_CCITT校验（大端模式）(2 byte) | 通讯包结束  （2byte） | |
| 0X01 | 0X58 |  |  |  |  | 有效数据的长度（2个字节，高位先发送）（data size） | Data | （2个字节，高位先发送） | 0X0D | 0X0A |

2、下位机响应通讯信息格式如下：

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 通讯包  起始（2byte） | | 485地址(1byte) | 指令代码（1byte） | 数据长度（大端模式）  （2byte） | 数据是否正确接收  （1byte） | 当前液晶屏显示页面状态（1 byte） | CRC16\_CCITT校验（大端模式）  (2 byte) | 通讯包结束  （2byte） | |
| 0X01 | 0X58 |  |  | 有效数据的长度（2个字节，高位先发送）（data size） |  |  | （2个字节，高位先发送） | 0X0D | 0X0A |

# 四、通讯协议详细说明

## 1、上位机发送模板指令的具体协议和下位机响应说明如下：

### （1）上位机发送模板指令（ASCII格式）（指令代码T（0x54））

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 通讯包  起始（2byte） | | 485地址(1byte) | 模板指令（1byte） | 模板总数（1byte） | 模板序号（1byte） | 数据长度（大端模式）  (2byte) | 数据  (N byte) | CRC16\_CCITT校验（大端模式）(2 byte) | 通讯包结束  （2byte） | |
| 0X01 | 0X58 |  | ‘T’ |  |  | 有效数据的长度（2个字节，高位先发送）（data size） | Data | （2个字节，高位先发送） | 0X0D | 0X0A |

使用说明：液晶屏模板数据长度是固定的，模板通讯可以使用单条指令通讯，也可以使用数据包一次性发送所有的模板指令。因为模板指令序号是从0开始的，模板总数却是一个具体的数量，所以当模板总数!=模板序号时是单条指令模式，当模板总数==模板序号时是指令包模式，使用指令包模式时，协议数据部分要解析成单条指令。

模板指令数据段代表的意义如下：

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| 通讯数据字节数(2byte) | 字号(1byte) | 显示字节宽度(1byte) | 显示模式(1byte) | X起始地址(2byte)  （大端模式） | Y起始地址(2byte)  （大端模式） | 背景颜色(2byte)  （大端模式） | 字体颜色(2byte)  （大端模式） |
|  | 液晶屏显示字体的大小 | 模板需要显示的字节数（汉字一个字是两个字节） | 目前在模板这个字节没有意义，因为在发送数据时带了这个命令 | 液晶显示X轴显示的起始地址 | 液晶显示Y轴显示的起始地址 | 当前指令显示的背景颜色 | 当前指令显示的字体颜色 |

### （2）上位机响应通讯信息格式如下：

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 数据包起始，两个字节 | | 数据长度（大端模式） | 数据 | CRC16\_CCITT校验（大端模式） | 数据包结束，两个字节 | |
| 0X01 | 0X58 | 有效数据的长度（2个字节，高位先发送）（data size） | Data | （2个字节，高位先发送） | 0X0D | 0X0A |

### （3）以下是单条模板发送举例：

a、单条模板指令发送，注意模板总数和模板序号不允许相等：

01 58 03 54 09 00 00 0B 20 14 00 00 00 00 00 00 1F FF FF 4E CC 0D 0A

01 58 03 54 09 01 00 0B 20 04 01 00 00 00 40 C6 18 01 CF 4C 40 0D 0A

01 58 03 54 09 02 00 0B 20 0E 01 00 60 00 40 C6 18 01 CF E6 38 0D 0A

01 58 03 54 09 03 00 0B 20 04 01 00 00 00 80 C6 18 01 CF 89 3A 0D 0A

01 58 03 54 09 04 00 0B 20 08 01 00 60 00 80 C6 18 01 CF AD 58 0D 0A

01 58 03 54 09 05 00 0B 20 02 01 00 E0 00 80 C6 18 01 CF E9 01 0D 0A

01 58 03 54 09 06 00 0B 20 06 01 00 00 00 C0 C6 18 01 CF C0 1C 0D 0A

01 58 03 54 09 07 00 0B 20 08 01 00 60 00 C0 C6 18 01 CF 31 93 0D 0A

01 58 03 54 09 08 00 0B 20 02 01 00 E0 00 C0 C6 18 01 CF 95 C6 0D 0A

1. 单条模板指令回复举例如下，当前没有液晶屏数据显示时当前显示状态为00（第八字节）：

01 58 03 54 00 02 01 00 31 6C 0D 0A

### （4）以下是模板打包发送举例：

a、模板打包指令，注意模板总数和模板序号必须相等，打包需要注意通讯数据长度是所有数据的长度，同时包里面包含每条模板指令的数据，每条指令是从命令序号开始到字体颜色结束，每条指令包含的指令单独的数据长度一定要是正确的，否则解析包的时候会出错：

01 58 03 54 09 09 00 7E 00 00 0B 20 14 00 00 00 00 00 00 1F FF FF 01 00 0B 20 04 01 00 00 00 40 C6 18 01 CF 02 00 0B 20 0E 01 00 60 00 40 C6 18 01 CF 03 00 0B 20 04 01 00 00 00 80 C6 18 01 CF 04 00 0B 20 08 01 00 60 00 80 C6 18 01 CF 05 00 0B 20 02 01 00 E0 00 80 C6 18 01 CF 06 00 0B 20 06 01 00 00 00 C0 C6 18 01 CF 07 00 0B 20 08 01 00 60 00 C0 C6 18 01 CF 08 00 0B 20 02 01 00 E0 00 C0 C6 18 01 CF C1 2E 0D 0A

b、模板打包发送下位机回复举例如下：

01 58 03 54 00 02 01 00 31 6C 0D 0A

## 2、上位机发送默认模板数据指令的具体协议和下位机响应说明如下：

### （1）上位机发送模板指令（ASCII格式）（指令代码M（0x4D））

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 通讯包  起始（2byte） | | 485地址(1byte) | 默认模板数据指令（1byte） | 默认模板数据总数（1byte） | 默认模板数据序号（1byte） | 数据长度（大端模式）  (2byte) | 数据  (N byte) | CRC16\_CCITT校验（大端模式）(2 byte) | 通讯包结束  （2byte） | |
| 0X01 | 0X58 |  | ‘M’ |  |  | 有效数据的长度（2个字节，高位先发送）（data size） | Data | （2个字节，高位先发送） | 0X0D | 0X0A |

使用说明：液晶屏默认模板数据长度不是固定的，默认模板数据通讯可以使用单条指令通讯，也可以使用数据包一次性发送所有的默认模板数据指令。因为默认模板数据指令序号是从0开始的，默认模板数据总数却是一个具体的数量，所以当默认模板数据总数!=默认模板数据序号时是单条指令模式，当默认模板数据总数==默认模板数据序号时是指令包模式，使用指令包模式时，协议数据部分要解析成单条指令。

默认模板数据指令数据段代表的意义如下：

|  |  |
| --- | --- |
| 字体显示模式（1byte） | 具体需要显示的字符（ASCII码，中文是ASCII内码）（N字节） |
| 字体显示模式0X00代表显示带背景颜色显示，0X01代表不带背景颜色显示（这种模式会使字体叠加，从而出现乱码），因此最好所有的字体显示模式都是要0X00，0X01只是预留这样的能力 | 根据模板设置的字节宽度填充相应的字符，需要往左往右格式化由上位机控制，空格（0X20）是不显示的格式化填充 |

### （2）上位机响应通讯信息格式如下：

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 数据包起始，两个字节 | | 数据长度（大端模式） | 数据 | CRC16\_CCITT校验（大端模式） | 数据包结束，两个字节 | |
| 0X01 | 0X58 | 有效数据的长度（2个字节，高位先发送）（data size） | Data | （2个字节，高位先发送） | 0X0D | 0X0A |

### （3）以下是单条默认模板数据发送举例

注意：有多少条模板指令，就有多少条默认模板数据指令，不显示的地方加空格填充：

A、单条默认模板数据指令发送，注意默认模板数据总数和默认模板数据序号不允许相等：

01 58 03 4D 09 00 00 15 00 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20 6F 4F 0D 0A

01 58 03 4D 09 01 00 05 01 B2 A1 C7 F8 2D 9E 0D 0A

01 58 03 4D 09 02 00 0F 01 20 20 20 20 20 20 20 20 20 20 20 20 20 20 E6 02 0D 0A

01 58 03 4D 09 03 00 05 01 BF F2 BA C5 05 DB 0D 0A

01 58 03 4D 09 04 00 09 01 20 20 20 20 20 20 20 20 3D 9E 0D 0A

01 58 03 4D 09 05 00 03 01 BA C5 45 70 0D 0A

01 58 03 4D 09 06 00 07 01 D2 D1 B7 D6 BC F0 A8 C2 0D 0A

01 58 03 4D 09 07 00 09 01 20 20 20 20 20 20 20 20 38 01 0D 0A

01 58 03 4D 09 08 00 03 01 B4 FC 8F 46 0D 0A

B、单条默认模板数据指令回复举例如下，当前没有液晶屏数据显示时当前显示状态为00（第八字节）：

1. 58 03 4D 00 02 01 00 9D 4A 0D 0A

### （4）以下是默认模板数据打包发送举例：

A、默认模板数据打包指令，注意默认模板数据总数和默认模板数据序号必须相等，打包需要注意通讯数据长度是所有数据的长度，同时包里面包含每条默认模板数据指令的数据，每条指令打包是从命令序号开始到显示数据结束，每条指令包含的指令单独的数据长度一定要是正确的，否则解析包的时候会出错：

01 58 03 4D 09 09 00 68 00 00 15 00 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20 01 00 05 01 B2 A1 C7 F8 02 00 0F 01 20 20 20 20 20 20 20 20 20 20 20 20 20 20 03 00 05 01 BF F2 BA C5 04 00 09 01 20 20 20 20 20 20 20 20 05 00 03 01 BA C5 06 00 07 01 D2 D1 B7 D6 BC F0 07 00 09 01 20 20 20 20 20 20 20 20 08 00 03 01 B4 FC 3A FB 0D 0A

B、默认模板数据打包发送下位机回复举例如下：

01 58 03 4D 00 02 01 00 9D 4A 0D 0A

## 3、上位机发送数据指令的具体协议和下位机响应说明如下：

因为数据指令协议和默认数据指令协议基本一致，只是默认模板数据是要在产品使用前发送完全自动保存，数据指令是产品使用过程中随时发送，通俗来说就是默认模板数据是固定显示部分，数据是可变显示部分。由于两个协议基本上是一样的，只是指令代码不一样，数据指令代码是‘D’(0X44),保存数据指令代码是‘S’以下直接给出举例。

### （1）以下是单条数据发送举例：

A、单条数据指令发送，注意单条发送数据数据总量位没有使用（第五个字节没有使用，默认希望为0X00）：

01 58 03 44 00 02 00 0F 00 20 CE F7 31 32 36 C7 F8 20 20 20 20 20 20 2B FA 0D 0A

01 58 03 44 00 04 00 09 00 20 20 20 35 38 36 20 20 40 24 0D 0A

01 58 03 44 00 07 00 09 00 20 20 31 33 35 39 38 20 48 77 0D 0A

B、单条数据指令回复举例如下，当前没有液晶屏数据显示时当前显示状态为00（第八字节）：

01 58 03 44 00 02 01 00 35 36 0D 0A

### （2）以下是数据打包发送举例：

A、数据打包指令，注意数据总数和数据序号必须相等，打包需要注意通讯数据长度是所有数据的长度，同时包里面包含每条数据指令的数据，每条指令打包是从命令序号开始到显示数据结束，每条指令包含指令单独的数据长度并且一定要是正确的，否则解析包的时候会出错：

2条指令组合

01 58 03 44 02 02 00 1E 02 00 0F 01 20 B6 AB 35 36 C7 F8 20 20 20 20 20 20 20 04 00 09 01 20 20 20 35 38 36 20 20 9F D7 0D 0A

3条指令组合

01 58 03 44 03 03 00 2A 02 00 0F 01 20 B6 AB 35 36 C7 F8 20 20 20 20 20 20 20 04 00 09 01 20 20 20 35 38 36 20 20 07 00 09 01 20 20 31 33 35 39 38 20 89 9B 0D 0A

B、默认模板数据打包发送下位机回复举例如下：

01 58 03 44 00 02 01 00 35 36 0D 0A

## 4、上位机发送保存数据指令的具体协议和下位机响应说明如下：

### （1）上位机发送保存数据指令（ASCII格式）（指令代码S（0x53））

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 通讯包  起始（2byte） | | 485地址(1byte) | 保存数据指令（1byte） | 保存数据总数和当前保存序号（1byte） | 数据显示总数（1byte） | 数据长度（大端模式）  (2byte) | 数据  (Nbyte) | CRC16\_CCITT校验（大端模式）(2 byte) | 通讯包结束  （2byte） | |
| 0X01 | 0X58 |  | ‘S’ | 这个字节需要拆分，前四位代表保存数据总数，后四位代表保存序号，最大保存15条数据 | 显示数据总数跟D指令的数据总数是一样的，和保存数据总数不是同一个东西 | 有效数据的长度（2个字节，高位先发送）（data size） | Data | （2个字节，高位先发送） | 0X0D | 0X0A |

使用说明：保存数据指令数据长度不是固定的，保存数据通讯只有包指令通讯模式，注意当前保存序号是从1开始的，有多少条要保存的数据，最大序号就是多少，保存数据总数却是一个具体的数量，保存数据需要按序号从小到大发送，序号为1的数据包发送完液晶显示屏直接显示当前包数据，其他包数据需要按特殊指令显示，当保存数据总数!= 当前保存序号时是最后一包保存数据指令包，只有收到最后一包数据时，才可以按照特殊指令定义的方式显示后续保存数据指令包数据。注意:保存数据指令包当中的数据段跟D指令包中的数据段是完全一样的。

保存数据指令数据段代表的意义如下：

|  |  |
| --- | --- |
| 字体显示模式（1byte） | 具体需要显示的字符（ASCII码，中文是ASCII内码）（N字节） |
| 字体显示模式0X00代表显示带背景颜色显示，0X01代表不带背景颜色显示（这种模式会使字体叠加，从而出现乱码），因此最好所有的字体显示模式都是要0X00，0X01只是预留这样的能力 | 根据模板设置的字节宽度填充相应的字符，需要往左往右格式化由上位机控制，空格（0X20）是不显示的格式化填充 |

### （2）上位机响应通讯信息格式如下：

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 数据包起始，两个字节 | | 数据长度（大端模式） | 数据 | CRC16\_CCITT校验（大端模式） | 数据包结束，两个字节 | |
| 0X01 | 0X58 | 有效数据的长度（2个字节，高位先发送）（data size） | Data | （2个字节，高位先发送） | 0X0D | 0X0A |

#### （3）以下是保存数据打包指令发送举例：

保存数据打包指令，注意保存数据总数和当前保存序号各占半个字节，最大可以发15条保存指令包，打包需要注意通讯数据长度是所有数据的长度，同时包里面包含每条数据指令的数据，每条指令打包是从命令序号开始到显示数据结束，每条指令包含指令单独的数据长度并且一定要是正确的，否则解析包的时候会出错：

A、3条显示指令组合,同时保存三条，这个是保存显示第一条（第五个字节31代表总共有三条保存指令，本次发送的指令包是第一条指令，第六个字节代表本次发送的指令包有三条数据要显示，注意区分这两个定义）

01 58 03 53 31 03 00 2A 02 00 0F 00 20 B6 AB 35 36 C7 F8 20 20 20 20 20 20 20 04 00 09 00 20 20 32 35 33 36 20 20 07 00 09 00 20 20 20 36 32 37 30 20 B9 9B 0D 0A

B、3条指令组合,保存显示第二条同时保存三条，这个是保存显示第二条（第五个字节32代表总共有三条保存指令，本次发送的指令包是第二条指令，第六个字节代表本次发送的指令包有三条数据要显示，注意区分这两个定义）

01 58 03 53 32 03 00 2A 02 00 0F 00 20 B6 AB 35 36 C7 F8 20 20 20 20 20 20 20 04 00 09 00 20 20 20 35 38 36 20 20 07 00 09 00 20 20 31 33 35 39 38 20 14 33 0D 0A

C、3条指令组合,保存显示第三条同时保存三条，这个是保存显示第三条（第五个字节33代表总共有三条保存指令，本次发送的指令包是第三条指令，第六个字节代表本次发送的指令包有三条数据要显示，注意区分这两个定义）

01 58 03 53 33 03 00 2A 02 00 0F 00 20 CE F7 32 35 C7 F8 20 20 20 20 20 20 20 04 00 09 00 20 20 20 35 38 36 20 20 07 00 09 00 20 20 31 33 31 35 38 20 C0 8A 0D 0A

（4）保存数据指令打包发送下位机回复举例如下：

1. 8 03 53 00 02 01 01 46 99 0D 0A

## 5、上位机发送特殊指令的具体协议和下位机响应说明如下：

特殊指令特殊对待，指令发送只需要保持基本格式一致（红色字体），其他根据特殊要求定义

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 通讯包起始（2byte） | | 485地址(1byte) | 特殊指令（1byte） | 特殊指令主分类（1byte） | 根据特殊指令改变（1byte） | 数据长度（大端模式）  (2byte) | 数据 | CRC16\_CCITT校验（大端模式）(2 byte) | 通讯包结束  （2byte） | |
| 0X01 | 0X58 |  | ‘C’ |  |  | 有效数据的长度（2个字节，高位先发送）（data size） | Data | （2个字节，高位先发送） | 0X0D | 0X0A |

### （1）特殊指令1

默认模板刷屏重启，使用过程中更新模板和默认模板需要发这条指令，否则需要断电重启

01 58 03 43 01 00 00 00 79 07 0D 0A

### （2）特殊指令2

清除eeprom里的模板和默认模板数据，注意数据段位0，但是数据长度还是要发两个00 00 字节

01 58 03 43 02 00 00 00 E2 DB 0D 0A

### （3）特殊指令3

RGB三色LED灯自由控制点亮和写灭，第六个字节每个位控制LED 三色灯中的一个灯，最低位控制R（红灯），第二位控制G（绿灯），第三位控制B（蓝灯），点亮置位1，熄灭复位0，由上位机自由控制，下位机不做自动熄灭处理。

01 58 03 43 03 01 00 00 A3 5F 0D 0A //LED R亮

01 58 03 43 03 02 00 00 FA 0F 0D 0A //LED G亮

01 58 03 43 03 04 00 00 48 AF 0D 0A //LED B亮

01 58 03 43 03 07 00 00 11 FF 0D 0A //LED RGB亮

### （4）特殊指令4

LED灯跟随保存数据指令显示而改变，第六个字节为0x00关闭跟随，为0x01时开启跟随，开启跟随需要带跟随LED的条数，用数据段表示，注意跟随条数必须和保存数据指令发送的总保存数据条数相当，否则会出错。

01 58 03 43 04 00 00 02 01 02 FC 34 0D 0A //配置跟随LED RGB,00关闭跟随，数据段没有意义

01 58 03 43 04 01 00 02 01 02 56 65 0D 0A //配置跟随LED RGB,01开启跟随，跟随两条，第一条亮红灯，第二条亮绿灯

01 58 03 43 04 01 00 03 01 02 07 59 60 0D 0A //配置跟随LED RGB,01开启跟随，跟随三条，第一条亮红灯，第二条亮绿灯第三条亮白灯（三色灯都亮的组合颜色）

### （5）特殊指令5

保存数据指令是否自动翻屏显示， 第六个字节为0x00停止自动翻屏，为0x01时开启开启自动翻屏，开启自动翻屏时需要使用数据段设置自动翻屏的时间，使用两个字节设置时间，单片机最小时间片是5ms，所以上位机设置的时间若是以1ms为单位时需要除以5ms，高位在前，地位在后，下位机获取的设置时间 = ((H byte)\*256 + (L byte）)\* 5ms

01 58 03 43 05 00 00 02 01 02 B9 94 0D 0A //停止自动翻屏

01 58 03 43 05 01 00 02 00 C8 58 F2 0D 0A //开启自动翻屏

### （6）特殊指令6

保存数据切换循环模式，此模式设置是为了使数据1-3,3-1不断循环显示，关闭循环模式数据从1-3显示完之后就会停留在第三个页面，循环模式加自动翻屏就能实现一直翻屏显示。 第六个字节为0x00停止切换循环模式，为0x01时开启切换循环模式

01 58 03 43 06 00 00 00 28 2A 0D 0A //停止切换循环模式

01 58 03 43 06 01 00 00 1F 1A 0D 0A //开启切换循环模式

### （7）特殊指令10

查询当前液晶屏显示页面，当有保存数据指令显示多页时才有意义，这个指令只是为了提供一个单独的指令查询当前显示页面，其实在每条指令回复的时候都带当前显示页面。

01 58 03 43 10 01 00 00 23 24 0D 0A //查询当前液晶屏显示页面

数据回复举例：

01 58 03 43 00 02 01 01 42 C3 0D 0A