

RS232-DMX512盒子协议命令表

V1.2

1: 模块串口上电时默认波特率是9600, 8位数据, 1停止位, 无奇偶效验 (9600 8N1), 可以通过串口命令修改串口速度, 以达到高速通讯, 详细见下面对应的说明。每次发送命令包时不要超过25mS, 否则模块会识别不符合数据, 超过时也会做超时处理。

2: 命令格式如下表, 均为十六进制格式, 例:C8 00 01 C0 01 00 00 FF FF, 这是全亮的命令。

3: 如果对十六制转换不太了解, 请下载一个进制转APP查一下。一般中控设备发送的都是十六进制的。

命令详解与示例

序号	命令	发送字数	说明	例子	返回值
1	CMD = 0xCB	9	更改串口波特率, 批量发送数据时必须要用高速! 设置好先返回值马上生效。 确认返回值后用户要马上切换对应的波特率。 断电后下次上电会恢复9600, 以免用户连不上。 D0=00是9600; D0=01是19200; D0=02是115200; D0=03是250000; D0=04是256000; D0=05是512000; D0=06是921600; D0=07是2000000 (2M); D0=08是4M	1、设置波特率为115200, d0=02即115200, C8 00 01 CB 00 00 00 00 02 2、设置波特率为250000, d0=03即250000, C8 00 01 CB 00 00 00 00 03	命令有效返回 (00 CMD 设置值 OK!) 错误返回 (Err 00 CMD) 效验错返回 (EF CMD Err 1 CRC计算结果+接收到的CRC)
2	CMD = 0xC0	9	设置全部通道调光值为xx, 同时发DMX+SPI, 也可独立发	全部通道为255 (开灯): C8 00 01 C0 00 09 00 00 FF 全部通道为 0 (关灯): C8 00 01 C0 00 09 00 00 00 只发DMX全部通道为127 (50%调光, 7F=127), 并打开CRC效验 :C8 00 01 C0 01 00 00 00 7F d0调光数值范围为: 00--FF.	命令有效返回 (OK! 00 CMD) 错误返回 (Err 00 CMD) 效验错返回 (EF CMD Err 1 CRC计算结果+接收到的CRC)
3	CMD = 0xC1	11	3通道RGB灯具时用 同时设置全部DMX+SPI的调光值 即通道: R (D0)=1、4、7... G (D1)=2、5、8... B (D2)=3、6、9...	RGB调光 (白色): C8 00 03 C1 00 00 00 00 FF FF FF RGB调光 (红色): C8 00 03 C1 00 00 00 00 FF 00 00	

4	CMD = 0xC3	12	<p>总调光 + RGB灯具， 共4通道 同时设置全部DMX+SPI的调光值 即通道：Dim(D0)是总调光 R(D1)=1、4、7... G(D2)=2、5、8... B(D3)=3、6、9...</p>	<p>总调光+R+G+B，（DMX+SPI所有域同时控） 下面为红色为00, 绿色为40, 蓝色为7F， 但受F0的总亮度调节RGB；</p> <p>C8 00 04 C3 00 00 00 00 F0 00 40 7F</p>
5	CMD = 0xC4	13	<p>总调光 + RGBW灯具， 共5通道 同时设置全部DMX+SPI的调光值 即通道：Dim(D0)是总调光 R(D1)=1、5、9... G(D2)=2、6、10... B(D3)=3、7、11... W(D4)=4、8、12...</p>	<p>总调光+R+G+B+W，（DMX+SPI所有域同时控） 下面为红色为FF, 蓝色为7F， 但受0E的总亮度调节RGBW；</p> <p>C8 00 05 C4 00 00 00 00 0E FF 00 7F</p>
6	CMD = 0xC5	9+	<p>批量发送通道MAX 512CH, DMX:CD=0, SPI:CD=1--12. 512CH时波特率最少要250000以上, 否则会超时。</p>	<p>DMX、SPI域可独立控制， 0F是DMX与SPI同时输出</p> <p>C8 02 00 C5 00 0F 00 00 CH1 CH2 CH3 CHn..... CHn <512</p>
7	CMD = 0xC7	9+	<p>类似THUR命令, 从第C个通道起发送n个d0数据</p>	<p>DMX、SPI域可独立控制， 0F是DMX与SPI同时输出</p> <p>从第 9个通道(08表示)， 发送 03个 [F0 FF 00]值的数</p> <p>C8 00 03 C7 00 0F 00 08 F0 FF 00</p> <p>00 08 是CRC_H+L表示从第几通道开始送数. 前面通道不影响（计算机中从00开始计算是1）； 00 03 是Len_H+L发送多少个通道</p>

8	CMD = 0xC8	9+	<p>从第C个通道起发送n个d0数据； 可设置每隔多少通道（Len_H）； 可设置多少台灯（Len_L）；</p> <p>适合摇头灯或PAR灯同型号灯同时设置多台灯的相同属性通道。</p>	<p>DMX、SPI域可独立控制，0F是DMX与SPI同时输出</p> <p>例：从第3[00 02]个通道发送04个数，隔03个通道，共发了03*04个 [F0 00 FF 7F] 数据 C8 03 04 C8 00 0F 00 02 F0 00 FF 7F</p>	
9	CMD = 0xD0	9	<p>停止自动 ART-NET转232， 当用0xD2打开自动向232口一直输出数据时，用0xD0关闭，否则一直输出</p>	<p>停止自动向232发送： C8 00 01 D0 00 00 00 00 00</p>	
10	CMD = 0xD1	9	<p>按需获取ART-NET转232数据， 发一次此命令，模块返回一次DMX数据到232输出， 用高波特率，否则超时</p>	<p>按需获取，模块只向串口传一次 读域00的DMX数据从00 01地址码开始，共0F个通道数： C8 00 01 D1 00 00 00 00 0F</p> <p>域范围0--8，一次读的个数MAX255，起始地址001--511</p>	
11	CMD = 0xD2	9	<p>打开自动 ART-NET转232， 打开自动向232口一直输出数据 打开后需要指令关闭 用高波特率，否则超时，一值刷新会占用SPI输出的刷新率！</p>	<p>连续向串口转发ART-NET接收到的DMX数据。 读域00的DMX数据从00 01地址码开始，共0F个数： C8 00 01 D2 00 00 00 00 0F</p> <p>读域01的DMX数据从00 09地址码开始，共1F个数： C8 00 09 D2 00 01 00 00 1F</p> <p>域范围0--8，一次读的个数MAX255，起始地址001--511</p>	
12	CMD = 0xD3	9	<p>使能DMX输出，DMX口输出信号</p>	<p>232命令控制DMX收发芯片，高底电平，切换为H即输出DMX模式： C8 00 01 D3 00 00 00 00 00</p>	<p>2021/9/11，定制/特定版本才有此功能指令</p>

13	CMD = 0xD4	9	<p>使能DMX输入，DMX口输入信号；</p> <p>当调用发送DMX数据0xCx指令时，如果要恢复DMX输入，请调用0xD4指令恢复DMX信号输入。</p>	<p>232命令控制DMX收发芯片，高底电平，切换为L即输入DMX信号转发到网络：</p> <p>C8 00 01 D4 00 00 00 00_00</p>	2021/9/12，定制/特定版本才有此功能指令
<p>CRC效验计算的例子程序, 只是简单效验.</p> <pre>uint_16 crc_count =0; //计算结果在此 uint_16 j; for(j=0; j<d_len; j++) //从第9个字节开始的数据开始相加 { crc_count =crc_count + RS232_buf[8+j]; //效验值是所有数据相加 }</pre>					