模块化矩阵 (VM 系列)

串口通讯协议

文件名	矩阵板卡串口协议
作者	
版本	V2.0
日期	2020-02-17
审核	

串口通信协议格式

数据格式

Baud rate: 9600

Data bits:8

Parity:None

Stop bits:1

协议格式

命令头 (2 字节)	命令码 (1 字节)	命令体长度 (1 字节)	命令体 (CMD_LENGTH 字节)	校验和 (1 字节)	命令尾 (2 字节)
CMD_HEAD	CMD_INDEX	CMD_LENGTH	CMD_BODY	CMD_CHECKSUM	CMD_TAIL
{} 0x7B 0x7B					}} 0x7D 0x7D

注:

1. 命令的所有数据为 16 进制;
2. 每条命令以 7b7b 开头, 7d7d 结尾;
3. 命令体长度: 命令体的字节数;
4. 校验和为: 除去校验和这 1 字节, 某一条命令中其他所有数据字节的总加和的低 8 位;

举例: 7B 7B 01 02 01 01 F5 7D 7D

7B 7B: Command head

01: Command index(Change routing command)

02: Command body length: the length of the command body

01 01: Command body

F5: Checksum. $7B+7B+01+02+01+01+7D+7D=1F5$, 8-bit checksum is F5.

7D 7D: Command tail

主板通信协议列表

0x01: 切换通道（普通）

命令头 (2byte)	命令索引 (1byte)	命令长度 (1byte)	命令体 (CMD_LENGTH byte)		校验和 (1byte)	命令尾 (2byte)
{{	0x01	0x02	输入通道	输出通道		}}
0x7b 0x7b	0x01	0x02				0x7d 0x7d

注:

假设矩阵最大规模值为 M，取值 8 或 16

- 当 M=8 时：输出通道每个 bit 表示是否要输出到该 bit 对应的输出 HDMI 端口；为 1 时要输出到该 bit 对应的输出 HDMI 端口
当 M=16 时：输出通道取值为 0-15 分别表示第 1-16 号输出端口，或取值 0xFF 表示所有输出口均切换到输入通道值表示的输入口
- 输入通道为 0~7 表示 HDMI 输入口 1~8，或 0~15 表示 HDMI 输入口 1~16
- 通过 0x95 命令获取矩阵的最大规模值；

举例:

当 M=8 时：把输入 2 切换到输出 1 当 M=16 时：把输入 2 切换到输出 2	把输入 4 切换到所有输出
串口命令：7B 7B 01 02 01 01 F5 7D 7D	串口命令：7B 7B 01 02 03 FF F5 7D 7D
7b 7b: 命令头 01: 命令字（通道切换命令） 02: 名字长度 02 01: 命令体 F5: Check-sum 7d 7d: 命令尾	7b 7b: 命令头 01: 命令字（通道切换命令） 02: 名字长度 03 ff: 命令体 F5: Check-sum 7d 7d: 命令尾
返回：7B 7B 01 02 01 01 F5 7D 7D	返回：7B 7B 01 02 03 FF F5 7D 7D

当设备接收命令成功后，会把该命令原封不动的返回。

0x94: Route 切换指令（拼接时使用）

命令头 (2 字节)	命令码 (1 字节)	命令体长度 (1 字节)	命令体 (CMD_LENGTH 字节)	校验和 (1 字节)	命令尾 (2 字节)
CMD_HEAD	CMD_INDEX	CMD_LENGTH	CMD_BODY	CMD_CHECKSUM	CMD_TAIL
{{					}}
7B 7B	94	3+N	见命令体格式		7D 7D

命令体格式: Mode(1Byte) + Input (1Byte) + Output (1Byte) + [可选数据]

Route 切换指令

- Mode: 0 – 可选数据长度为 0 字节，即无可选数据；
 1 – 可选数据长度为 2N 字节，为 Input-Output 的数据对（此时 Mode 后面的 Input 和 Output 也为数据对）；
 2 – 可选数据长度为 N 字节，数据值的某一个 bit 值为 1 时表示 outputX 需要切换，为 0 则不需要；
 0xFF - 可选数据长度为 0 字节，即无可选数据；
- Input: 表示 Input 端口；
- Output: 当 Mode=0 时，表示 Output 端口；
 当 Mode=1 时，Input 和 Output 组成一个 route 信息对，可选数据为 N 组 route 信息对；
 当 Mode=2 时，Output 和可选数据组合成 (N+1)*8 个 bit，第 X 个 bit 值为 1 表示 Output-X 需要切换至 Input；
 当 Mode=0xFF 时，Output 值不使用；

命令返回说明:

- 返回当前矩阵的 Route 信息；

举例:

- Mode 0, 设置单个 Route 信息: 7B 7B 94 03 00 01 00 88 7D 7D //Input1->Output0
- Mode 1, 设置多个 Route 信息: 7B 7B 94 05 01 02 01 07 07 xx 7D 7D //Input2->Output1, Input7->Output7
- Mode 2, 设置多个 Route 信息（输入相同）: 7B 7B 94 04 02 01 00 F0 xx 7D 7D //Input1->Output8,9,10,11
- Mode 0xFF, 设置 Allset 的 Route 信息: 7B 7B 94 03 FF 01 00 87 7D 7D //Input1->All Output
- 其他 Mode 值保留未使用

0x95： 矩阵规模

	命令头 (2 字节)	命令码 (1 字节)	命令体长度 (1 字节)	命令体 (CMD_LENGTH 字节)	校验和 (1 字节)	命令尾 (2 字节)
	CMD_HEAD	CMD_INDEX	CMD_LENGTH	CMD_BODY	CMD_CHECKSUM	CMD_TAIL
	{{					}}
	7B 7B	95	3 或 4	见命令体格式		7D 7D
矩 阵 规 模	命令体格式： Mode(1Byte) + OutputNUM (1Byte) + InputNUM (1Byte) +MaxNUM (1Byte)					
	1. Mode: 0 – 读取矩阵规模 1 – 设置矩阵规模					
	2. OutputNUM: 表示输出端口个数; 3. InputNUM: 表示输入端口个数; 4. MaxNUM : 输入/输出最大的端口个数					
	命令返回说明: 1. 返回当前矩阵的规模信息;					
	举例: 1. 读取矩阵规模: 7B 7B 95 03 00 00 00 xx 7D 7D //读取矩阵规模 2. 设置矩阵规模: 7B 7B 95 03 01 08 08 xx 7D 7D //设置矩阵规模 8x8					

0x96：板卡软件版本

命令头 (2 字节)	命令码 (1 字节)	命令体长度 (1 字节)	命令体 (CMD_LENGTH 字节)	校验和 (1 字节)	命令尾 (2 字节)
CMD_HEAD	CMD_INDEX	CMD_LENGTH	CMD_BODY	CMD_CHECKSUM	CMD_TAIL
{} 7B 7B	96	9	见命令体格式		}} 7D 7D
<p>命令体格式： Address(1Byte) + Ctrl(1Byte) + yearH(1Byte)+ yearL(1Byte)+ month(1Byte)+ day(1Byte)+ hour(1Byte)+ minute(1Byte)+ second(1Byte)</p> <p>1. Address: 板卡地址，在读取输入输出板卡时使用，其他情况默认为 0</p> <p>2. Ctrl: 0-读取主板软件版本 1-读取按键板软件版本 2-读取输出板卡软件版本 3-读取输出板卡-KDMCU 软件版本信息 4-读取输入板卡软件版本信息 其他-保留未使用</p> <p>3. yearH: 年，如 0x20</p> <p>4. yearL: 年，如 0x16</p> <p>5. month: 月，如 0x10</p> <p>6. day: 日，如 0x01</p> <p>7. hour: 时，如 0x08</p> <p>8. minute: 分，如 0x30</p> <p>9. second: 秒，如 0x00</p> <p>上述举例值，合起来为软件编译时间：2016 年 10 月 1 日，08 时 30 分 00 秒</p> <p>命令返回说明：</p> <p>1. 返回当前板卡软件的版本信息（编译时间）：</p> <p>举例：</p> <p>1. 读取软件版本：7B 7B 96 00 00 00 00 00 00 00 00 00 xx 7D 7D //读取主板软件版本 返回数据举例：7B 7B 96 00 00 20 16 10 01 08 30 00 xx 7D 7D //主板软件版本：2016 年 10 月 1 日，08 时 30 分 00 秒</p>					

板卡软件版本

0x97： 拼接模式保存

拼接 模式 保存	命令头 (2 字节)	命令码 (1 字节)	命令体长度 (1 字节)	命令体 (CMD_LENGTH 字节)	校验和 (1 字节)	命令尾 (2 字节)
	CMD_HEAD	CMD_INDEX	CMD_LENGTH	CMD_BODY	CMD_CHECKSUM	CMD_TAIL
	{{					}}
	7B 7B	97	1	Mode		7D 7D
	Mode: 0-6					

0x98: 拼接模式调用

拼接模式调用	命令头 (2 字节)	命令码 (1 字节)	命令体长度 (1 字节)	命令体 (CMD_LENGTH 字节)	校验和 (1 字节)	命令尾 (2 字节)
	CMD_HEAD	CMD_INDEX	CMD_LENGTH	CMD_BODY	CMD_CHECKSUM	CMD_TAIL
	{{					}}
	7B 7B	98	1	mode		7D 7D
	Mode: 0-6					

0x99: 拼接模式个数查询

拼接模式调用	命令头 (2 字节)	命令码 (1 字节)	命令体长度 (1 字节)	命令体 (CMD_LENGTH 字节)	校验和 (1 字节)	命令尾 (2 字节)
	CMD_HEAD	CMD_INDEX	CMD_LENGTH	CMD_BODY	CMD_CHECKSUM	CMD_TAIL
	{{					}}
	7B 7B	99	1	Mode_NUM		7D 7D
	Mode_NUM: VM1616 返回 7, VM88 返回 7					

0x02: 场景保存

把当前的输入输出对应关系保存为某一个场景

命令头 (2byte)	命令索引 (1byte)	命令长度 (1byte)	命令体 (CMD_LENGTH byte)	校验和 (1byte)	命令尾 (2byte)
{{	0x02	0x01	场景序列号		}}
0x7b 0x7b	0x02	0x01			0x7d 0x7d

注:

(1) 系统支持 8 种场景;

举例:

把当前的输入输出对应关系保存为模式 1	把当前的输入输出对应关系保存为模式 8
串口命令: 7B 7B 02 01 00 F3 7D 7D	串口命令: 7B 7B 02 01 07 FA 7D 7D
7b 7b: 命令头 02: 命令字 (通道切换命令) 01: 名字长度 00: 命令体, 表示场景 1 (0~7 分别表示场景 1~8) F3: Check-sum 7d 7d: 命令尾	7b 7b: 命令头 01: 命令字 (通道切换命令) 01: 名字长度 0,7: 命令体, 表示场景 1 (0~7 分别表示场景 1~8) FA: Check-sum 7d 7d: 命令尾
返回: 不返回命令	返回: 不返回命令

0x03: 场景调用

把当前的输入输出对应关系设置成某一场景已保存的输入输出对应关系

命令头 (2byte)	命令索引 (1byte)	命令长度 (1byte)	命令体 (CMD_LENGTH byte)	校验和 (1byte)	命令尾 (2byte)
{{	0x03	0x01	场景序列号		}}
0x7b 0x7b	0x03	0x01			0x7d 0x7d

注:

(1) 系统支持 8 种场景;

举例:

把当前的输入输出对应关系设置成场景 1 已保存的的输入输出对应关系	把当前的输入输出对应关系设置成场景 8 已保存的的输入输出对应关系
串口命令: 7B 7B 03 01 00 F4 7D 7D	串口命令: 7B 7B 03 01 07 FB 7D 7D
7b 7b: 命令头 03: 命令字 (通道切换命令) 01: 名字长度 00: 命令体, 表示场景 1 (0~7 分别表示场景 1~8) F4: Check-sum 7d 7d: 命令尾	7b 7b: 命令头 03: 命令字 (通道切换命令) 01: 名字长度 07: 命令体, 表示场景 1 (0~7 分别表示场景 1~8) FB: Check-sum 7d 7d: 命令尾
返回: 7B 7B 11 04 00 01 02 03 0B 7D 7D (返回命令的具体意思请查命令 0x11)	返回: 7B 7B 11 04 00 01 02 03 0B 7D 7D (返回命令的具体意思请查命令 0x11)

0x12:让矩阵汇报当前的输入输出对应关系的命令 (0x12): 该命令用于让矩阵返回当前的输入输出对应关系, 矩阵将通过 0x11 命令返回:

命令头 (2byte)	命令索引 (1byte)	命令长度 (1byte)	命令体 (CMD_LENGTH byte)	校验和 (1byte)	命令尾 (2byte)
{{	0x12	1	0x00		}}
0x7b 0x7b	0x12	1	0x00		0x7d 0x7d

举例:

7B 7B 12 01 00 03 7D 7D, 矩阵收到改命令后将会通过 0x11 命令返回矩阵当前的输入输出对应关系;

0x11:汇报矩阵当前输入输出对应关系的指令（0x11）

命令头 (2byte)	命令索引 (1byte)	命令长度 (1byte)	命令体 (CMD_LENGTH byte)					校验和 (1byte)	命令尾 (2byte)
{{	0x11		输出 1 对 应的输入	输出 2 对 应 的 输入	输出 3 对 应 的 输入	输出 4 对 应 的 输入			}}
0x7b 0x7b	0x11								0x7d 0x7d

注：

- (1) 该命令的长度取决于矩阵的输出通道数；
- (2) 命令体依次表示输出 1~8 的输入通道，0~7 表示输入 1~8

举例：

4X4 矩阵：当前的输入输出配置关系为 1-1,2-2,3-3,4-4	4X4 矩阵：当前的输入输出配置关系为 1-1,1-2,1-3,1-4
串口命令：7B 7B 11 04 00 01 02 03 0B 7D 7D	串口命令：7B 7B 11 04 00 00 00 00 05 7D 7D
7b 7b: 命令头 11: 命令字（通道切换命令） 04: 名字长度 00 01 02 03: 命令体。改返回指令表示输入 1 到输出 1， 输入 2 到输出 2，输入 3 到输出 3，输入 4 到输出 4， 0B: Check-sum 7d 7d: 命令尾	7b 7b: 命令头 11: 命令字（通道切换命令） 04: 名字长度 00 00 00 00: 命令体。改返回指令表示输入 1 到输出 1， 输入 1 到输出 2，输入 1 到输出 3，输入 1 到输出 4， 05: Check-sum 7d 7d: 命令尾
返回：无返回	返回：无返回

0xAA：恢复出厂设置

命令头 (2byte)	命令索引 (1byte)	命令长度 (1byte)	命令体 (CMD_LENGTH byte)		校验和 (1byte)	命令尾 (2byte)
{{	0xAA	0x02	0x01	0x01	0x9c	}}
0x7b 0x7b	0xAA	0x02	0x01	0x01	0x9c	0x7d 0x7d

设备接收到该命令后会恢复工厂设置，并返回矩阵恢复工厂设置后的输入输出对应关系的指令（0x11）

输入输出板卡协议列表

0xFF：返回命令执行错误结果

1. 若发送命令成功处理，则设置命令的返回数据与发送的数据相同，查询命令的返回数据与查询内容相关
2. 若发送命令**处理失败**，则统一返回命令码 0xFF，命令体长度为 3 个字节，命令体内容为发送命令的**目标地址+命令码+操作类型**（操作类型为设置-0x00 或查询-0x01）

命令头 (2 字节)	命令码 (1 字节)	命令体长度 (1 字节)	命令体 (CMD_LENGTH 字节)	校验和 (1 字节)	命令尾 (2 字节)
CMD_HEAD	CMD_INDEX	CMD_LENGTH	CMD_BODY	CMD_CHECKSUM	CMD_TAIL
{{					}}
7B 7B	FF	03	目标地址+操作码+操作类型		7D 7D

处理失败返回命令说明：

1. 返回数据的**命令码**统一为 0xFF；
2. 返回数据的**命令体长度**统一固定为 0x03，即命令体长度固定为 3 字节；
3. 返回数据的**命令体内容**为**目标地址+命令码+操作类型**；
4. 操作码为 0x00 代表发送的设置命令处理失败，或操作码为 0x01 代表发送是查询命令处理失败。

0x80: OSD 背景透明

命令头 (2 字节)	命令码 (1 字节)	命令体长度 (1 字节)	命令体 (CMD_LENGTH 字节)	校验和 (1 字节)	命令尾 (2 字节)
CMD_HEAD	CMD_INDEX	CMD_LENGTH	CMD_BODY	CMD_CHECKSUM	CMD_TAIL
{} 7B 7B	80	03	见命令体格式		}} 7D 7D
<p>命令体格式: Address(1Byte) + Ctrl(1Byte) + Transparent(1Byte)</p> <p>1. Address: 目标地址</p> <p>2. Ctrl 值的 bit7: 1-表示查询 OSD 背景是否透明 (此时该命令除 Ctrl 外, 其他的参数可以是任意值), 0-表示设置 OSD 背景透明与否;</p> <p>Ctrl 值的 Bit6: 1-表示板卡地址为输入板卡; 0-表示板卡地址为输出板卡;</p> <p>Ctrl 值的 bit[5,0]: 保留未使用;</p> <p>3. Transparent: 1-OSD 背景透明; 0-OSD 背景不透明;</p> <p>命令返回说明:</p> <p>1. 与发送的命令格式相同;</p> <p>2. 设置命令成功时, 返回数据与发送数据相同; 设置失败时, 返回 7B 7B FF 03 add 80 00 xx 7D 7D</p> <p>3. 查询命令成功时, 返回数据与查询内容相关; 查询失败时, 返回 7B 7B FF 03 add 80 01 xx 7D 7D</p> <p>举例:</p> <p>1. 设置 OSD 背景透明: 7B 7B 80 03 00 00 01 74 7D 7D 设置成功, 返回命令为: 7B 7B 80 03 00 00 01 74 7D 7D 设置失败, 返回命令为: 7B 7B FF 03 00 80 00 72 7D 7D</p>					

OSD 透明度

0x81: OSD 显示位置

命令头 (2 字节)	命令码 (1 字节)	命令体长度 (1 字节)	命令体 (CMD_LENGTH 字节)	校验和 (1 字节)	命令尾 (2 字节)
CMD_HEAD	CMD_INDEX	CMD_LENGTH	CMD_BODY	CMD_CHECKSUM	CMD_TAIL
{} 7B 7B	81	06	见命令体格式		}} 7D 7D
<p>命令体格式: Address(1Byte) + Ctrl(1Byte) + xH(1Byte) + xL(1Byte) + yH(1Byte) + yL(1Byte)</p> <ol style="list-style-type: none"> Address: 目标地址 Ctrl 值的 bit7: 1-表示查询文字窗口的位置信息 (此时该命令除 Ctrl 外, 其他的参数可以是任意值), 0-表示设置文字窗口的位置; Ctrl 值的 Bit6: 1-表示板卡地址为输入板卡; 0-表示板卡地址为输出板卡; Ctrl 值的 bit5: 保留未使用; Ctrl 值的 bit4: 1 表示操作的是窗口的左上角坐标, 此时 Ctrl 值的 bit[3,0]值不起作用, 0 表示操作的是窗口的整体位置, 此时与 Ctrl 值的 bit[3,0]配合使用; Ctrl 值的 bit[3,2]: 00-垂直靠上, 01-垂直居中, 10-垂直靠下, 11-保留未使用 Ctrl 值的 bit[1,0]: 00-水平靠左, 01-水平居中, 10-水平靠右, 11-保留未使用 xH 为 x 坐标值的高八位 (Ctrl:bit6 为 1 时); xL 为 x 坐标值的低八位 (Ctrl:bit6 为 1 时); yH 为 y 坐标值的高八位 (Ctrl:bit6 为 1 时); yL 为 y 坐标值的低八位 (Ctrl:bit6 为 1 时); <p>命令返回说明:</p> <ol style="list-style-type: none"> 与发送的命令格式相同; xH/xL/yH/yL 值的含义与 Ctrl 的值相关。 设置命令成功时, 返回数据与发送数据相同; 设置失败时, 返回 7B 7B FF 03 add 81 00 xx 7D 7D 查询命令成功时, 返回数据与查询内容相关; 查询失败时, 返回 7B 7B FF 03 add 81 01 xx 7D 7D <p>举例 (假设目标地址均为 0x00) :</p> <ol style="list-style-type: none"> 设置 OSD 窗口左上角坐标为(80,60): 7B 7B 81 06 00 40 00 50 00 3C 43 7D 7D 设置成功, 返回与设置命令命令相同: 7B 7B 81 06 00 40 00 50 00 3C 43 7D 7D 设置失败, 返回命令为: 7B 7B FF 03 00 81 00 73 7D 7D 查询 OSD 窗口左上角坐标: 7B 7B 81 06 00 C0 00 00 00 00 36 7D 7D 查询成功, 返回命令结果为: 7B 7B 81 06 00 C0 00 50 00 3C C3 7D 7D(假设已设置左上角坐标为(80,60)) 查询失败, 返回命令结果为: 7B 7B FF 03 00 81 01 74 7D 7D 设置 OSD 窗口水平和垂直居中: 7B 7B 81 06 00 05 07 80 04 38 3F 7D 7D (假设显示器分辨率为 1920*1080) 设置成功, 返回命令结果(相同): 7B 7B 81 06 00 05 07 80 04 38 3F 7D 7D 设置失败, 返回命令结果: 7B 7B FF 03 00 81 00 73 7D 7D 查询 OSD 窗口的水平和垂直居中情况: 7B 7B 81 06 00 80 00 00 00 00 F7 7D 7D 查询成功, 返回命令为: 7B 7B 81 06 00 85 07 80 04 38 BF 7D 7D (假设已经设置显示器分辨率为 1920*1080 了) 查询失败, 返回命令为: 7B 7B FF 03 00 81 01 74 7D 7D 					

OSD 显示位置

0x82: OSD 字体信息

命令头 (2 字节)	命令码 (1 字节)	命令体长度 (1 字节)	命令体 (CMD_LENGTH 字节)	校验和 (1 字节)	命令尾 (2 字节)
CMD_HEAD	CMD_INDEX	CMD_LENGTH	CMD_BODY	CMD_CHECKSUM	CMD_TAIL
{} 7B 7B	82	04	见命令体格式		}} 7D 7D
<p>命令体格式: Address(1Byte) + Ctrl(1Byte) + Type(1Byte) + Size(1Byte)</p> <p>1. Address: 目标地址</p> <p>2. Ctrl 值的 bit7: 1-表示查询当前的字体信息（此时该命令的其他参数可以是任意值）， 0-表示设置字体信息； Ctrl 值的 Bit6: 1-表示板卡地址为输入板卡； 0-表示板卡地址为输出板卡； Ctrl 值的 bit5: 保留未使用； Ctrl 值的 bit 4: 1-颜色反转, 0-不反转 Ctrl 值的 bit 3: 1-斜体, 0-正常 Ctrl 值的 bit 2: 1-下划线, 0-无下划线 Ctrl 值的 bit 1: 1-加黑, 0-不加黑 Ctrl 值的 bit 0: 1-加粗, 0-不加粗</p> <p>3. Type 值表示字体类型: 0x00-黑体、0x 01-楷体、0x 02 宋体、0x 03-仿宋、其他值保留使用</p> <p>4. Size 值表示字体大小: 0x00-32*32 字体、0x01-48*48 字体、0x02-64*64 字体、0x03-80*80、0x04-96*96、其他值保留使用</p> <p>命令返回说明:</p> <p>1. 与发送的命令格式相同；</p> <p>2. 设置命令成功时，返回数据与发送数据相同；设置失败时，返回 7B 7B FF 03 add 82 00 xx 7D 7D</p> <p>3. 查询命令成功时，返回数据与查询内容相关；查询失败时，返回 7B 7B FF 03 add 82 01 xx 7D 7D</p> <p>举例（假设目标地址均为 0x00）:</p> <p>1. 设置 OSD 字体为黑体、斜体、64*64 的字体: 7B 7B 82 04 00 08 00 01 7F 7D 7D 设置成功，返回命令为: 7B 7B 82 04 00 08 00 01 7F 7D 7D 设置失败，返回命令为: 7B 7B FF 03 00 82 00 73 7D 7D</p>					

OSD
字
体
信
息

0x83: OSD 颜色信息

命令头 (2 字节)	命令码 (1 字节)	命令体长度 (1 字节)	命令体 (CMD_LENGTH 字节)	校验和 (1 字节)	命令尾 (2 字节)
CMD_HEAD	CMD_INDEX	CMD_LENGTH	CMD_BODY	CMD_CHECKSUM	CMD_TAIL
{} 7B 7B	83	05	见命令体格式		}} 7D 7D
<p>命令体格式: Address(1Byte) + Ctrl(1Byte) + R(1Byte) + G(1Byte) + B(1Byte)</p> <ol style="list-style-type: none"> Address: 目标地址 Ctrl 值的 bit7: 1-表示查询当前的颜色信息（此时该命令除 Ctrl 外，其他的参数可以是任意值），0-表示设置颜色信息； Ctrl 值的 Bit6: 1-表示板卡地址为输入板卡；0-表示板卡地址为输出板卡； Ctrl 值的 bit[5,2]: 保留未使用； Ctrl 值的 bit1: 1-表示 RGB 参数为十进制值，0-表示 RGB 参数为十六进制值； Ctrl 值的 bit0: 1-设置背景色，0-设置前景色（文字颜色）； R 为 RGB 中的 Red 值，数的进制与 Ctrl 的 bit1 的值有关； G 为 RGB 中的 Green 值，数的进制与 Ctrl 的 bit1 的值有关； B 为 RGB 中的 Blue 值，数的进制与 Ctrl 的 bit1 的值有关； <p>命令返回说明:</p> <ol style="list-style-type: none"> 与发送的命令格式相同； RGB 值为颜色值，其进制值与 Ctrl 的 bit1 相关，其表示背景色或前景色与 Ctrl 的 bit0 相关。 设置命令成功时，返回数据与发送数据相同；设置失败时，返回 7B 7B FF 03 add 83 00 xx 7D 7D 查询命令成功时，返回数据与查询内容相关；查询失败时，返回 7B 7B FF 03 add 83 01 xx 7D 7D <p>举例:</p> <ol style="list-style-type: none"> 设置 OSD 字体的前景色（文字颜色为(255,255,255)）：7B 7B 83 05 00 00 FF FF FF 75 7D 7D 设置成功，返回命令为：7B 7B 83 05 00 00 FF FF FF 75 7D 7D 设置失败，返回命令为：7B 7B FF 03 00 83 00 75 7D 7D 设置 OSD 字体的前景色（文字颜色为(80,80,80)）：7B 7B 83 05 00 02 80 80 80 FA 7D 7D 设置成功，返回命令为：7B 7B 83 05 00 02 80 80 80 FA 7D 7D 设置失败，返回命令为：7B 7B FF 03 00 83 00 75 7D 7D 					

OSD 颜色信息

0x84: OSD 文字滚动

命令头 (2 字节)	命令码 (1 字节)	命令体长度 (1 字节)	命令体 (CMD_LENGTH 字节)	校验和 (1 字节)	命令尾 (2 字节)
CMD_HEAD	CMD_INDEX	CMD_LENGTH	CMD_BODY	CMD_CHECKSUM	CMD_TAIL
{} 7B 7B	84	03	见命令体格式		}} 7D 7D
<p>命令体格式: Address(1Byte) + Ctrl(1Byte) + Flag(1Byte)</p> <ol style="list-style-type: none"> Address: 目标地址 Ctrl 值的 bit7: 1-表示查询是否是文字滚动; 0-表示设置文字滚动。 Ctrl 值的 Bit6: 1-表示板卡地址为输入板卡; 0-表示板卡地址为输出板卡; Ctrl 值的 bit[5,0]: 保留未使用; Flag 值的: 1-文字滚动, 0-文字不滚动; <p>命令返回说明:</p> <ol style="list-style-type: none"> 与发送的命令格式相同; 设置命令成功时, 返回数据与发送数据相同; 设置失败时, 返回 7B 7B FF 03 add 84 00 xx 7D 7D 查询命令成功时, 返回数据与查询内容相关; 查询失败时, 返回 7B 7B FF 03 add 84 01 xx 7D 7D <p>举例:</p> <ol style="list-style-type: none"> 设置文字滚动为打开: 7B 7B 84 03 00 00 01 78 7D 7D 设置成功, 返回命令为: 7B 7B 84 03 00 00 01 78 7D 7D 设置失败, 返回命令为: 7B 7B FF 03 00 84 00 76 7D 7D 					

OSD 文字滚动

0x85: OSD 文字内容

命令头 (2 字节)	命令码 (1 字节)	命令体长度 (1 字节)	命令体 (CMD_LENGTH 字节)	校验和 (1 字节)	命令尾 (2 字节)
CMD_HEAD	CMD_INDEX	CMD_LENGTH	CMD_BODY	CMD_CHECKSUM	CMD_TAIL
{} 7B 7B	85	(04+N)	见命令体格式		}} 7D 7D
<p>命令体格式: Address(1Byte) + Ctrl(1Byte) + Line(1Byte) + Column(1Byte) + String(N Byte)</p> <ol style="list-style-type: none"> Address: 目标地址 Ctrl 值的 bit7: 1-表示查询当前行字符内容, 0-表示设置当前行字符内容; Ctrl 值的 Bit6: 1-表示板卡地址为输入板卡; 0-表示板卡地址为输出板卡; Ctrl 值的 bit[5,3]: 保留未使用; Ctrl 值的 bit2: 1-表示将快速添加字符串 (可能显示不完整), 0-全面显示字符串 (可能耗时长); Ctrl 值的 bit1: 1-表示将某行的(Line,Column)位置的写入内容之前的部分擦除为空白, 0-不擦除; Ctrl 值的 bit0: 1-表示将某行的(Line,Column)位置的写入内容之后的部分擦除为空白, 0-不擦除; Line 为需要修改的行; Column 为需要修改的列 (查询时 Column 值无影响, 建议为 0); String 为需要写入的文字串的编码值,长度为任意的 N 个字节 (查询时 String 长度无影响, 建议为 0); <p>命令返回说明:</p> <ol style="list-style-type: none"> 与发送的命令格式相同; 设置命令成功时, 返回数据与发送数据相同; 设置失败时, 返回 7B 7B FF 03 add 85 00 xx 7D 7D 查询命令成功时, 返回数据与查询内容相关; 查询失败时, 返回 7B 7B FF 03 add 85 01 xx 7D 7D <p>举例:</p> <ol style="list-style-type: none"> 设置第一行第一列开始的文字为“1234”: 7B 7B 85 08 00 03 00 00 31 32 33 34 4A 7D 7D 设置成功, 返回命令为: 7B 7B 85 08 00 03 00 00 31 32 33 34 4A 7D 7D 设置失败, 返回命令为: 7B 7B FF 03 00 85 00 76 7E 7D 					

OSD 文字内容

0x86: OSD 不透明度

命令头 (2 字节)	命令码 (1 字节)	命令体长度 (1 字节)	命令体 (CMD_LENGTH 字节)	校验和 (1 字节)	命令尾 (2 字节)
CMD_HEAD	CMD_INDEX	CMD_LENGTH	CMD_BODY	CMD_CHECKSUM	CMD_TAIL
{{					}}
7B 7B	86	03	见命令体格式		7D 7D

命令体格式: Address(1Byte) + Ctrl(1Byte) + Alpha (1Byte)

- Address: 目标地址
- Ctrl 值的 bit7: 1-表示**查询** OSD 的不透明度 (此时该命令除 Ctrl 外, 其他的参数可以是任意值),
0-表示**设置** OSD 的不透明度;
Ctrl 值的 Bit6: 1-表示板卡地址为输入板卡;
0-表示板卡地址为输出板卡;
Ctrl 值的 bit[5,0]: 保留未使用;
- Alpha** 为需要设置的不透明度值;

命令返回说明:

- 与发送的命令格式相同;
- 设置命令成功时, 返回数据与发送数据相同; 设置失败时, 返回 7B 7B FF 03 **add 86 00** xx 7D 7D
- 查询命令成功时, 返回数据与查询内容相关; 查询失败时, 返回 7B 7B FF 03 **add 86 01** xx 7D 7D

举例:

- 设置 OSD 不透明度为 128: 7B 7B **86 03 00 00 80** F9 7D 7D
设置成功, 返回命令为: 7B 7B **86 03 00 00 80** F9 7D 7D
设置失败, 返回命令为: 7B 7B **FF 03 00 86 00** 78 7D 7D

OSD
透
明
度

0x87: OSD 开关

命令头 (2 字节)	命令码 (1 字节)	命令体长度 (1 字节)	命令体 (CMD_LENGTH 字节)	校验和 (1 字节)	命令尾 (2 字节)
CMD_HEAD	CMD_INDEX	CMD_LENGTH	CMD_BODY	CMD_CHECKSUM	CMD_TAIL
{{					}}
7B 7B	87	03	见命令体格式		7D 7D

命令体格式: Address(1Byte) + Ctrl(1Byte) + Flag(1Byte)

- Address: 目标地址
- Ctrl 值的 bit7: 1-表示**查询** OSD 的开关状态,
0-表示**设置** OSD 的开关状态;
Ctrl 值的 Bit6: 1-表示板卡地址为输入板卡;
0-表示板卡地址为输出板卡;
Ctrl 值的 bit[5,0]: 保留未使用;
- Flag 值的: 1-打开, 0-关闭;

命令返回说明:

- 与发送的命令格式相同;
- 设置命令成功时, 返回数据与发送数据相同; 设置失败时, 返回 7B 7B FF 03 **xx 87 00** xx 7D 7D
- 查询命令成功时, 返回数据与查询内容相关; 查询失败时, 返回 7B 7B FF 03 **xx 87 01** xx 7D 7D

举例:

- 设置 OSD 为打开: 7B 7B **87 03 00 00 01** 7B 7D 7D
设置成功, 返回命令为: 7B 7B **87 03 00 00 01** 7B 7D 7D
设置失败, 返回命令为: 7B 7B **FF 03 00 87 00** 79 7D 7D

O
S
D
开
关

0x88: OSD 预设

命令头 (2 字节)	命令码 (1 字节)	命令体长度 (1 字节)	命令体 (CMD_LENGTH 字节)	校验和 (1 字节)	命令尾 (2 字节)
CMD_HEAD	CMD_INDEX	CMD_LENGTH	CMD_BODY	CMD_CHECKSUM	CMD_TAIL
{} 7B 7B	88	03	见命令体格式		}} 7D 7D
命令体格式: Address(1Byte) + Ctrl(1Byte) + PreNUM (1Byte) 1. Address: 目标地址 2. Ctrl 值的 bit7: 1-表示读取 OSD 预设并显示, 0-表示 存储 当前的显示内容为 OSD 预设; Ctrl 值的 Bit6: 1-表示板卡地址为输入板卡; 0-表示板卡地址为输出板卡; Ctrl 值的 bit[5,0]: 保留未使用; 3. PreNUM:预设号码: 如果该号码已经被存储了, 则覆盖之前的该预设号码的存储内容; 命令返回说明: 1. 与发送的命令格式相同; 2. 设置命令成功时, 返回数据与发送数据相同; 设置失败时, 返回 7B 7B FF 03 add 88 00 xx 7D 7D 3. 查询命令成功时, 返回数据与查询内容相关; 查询失败时, 返回 7B 7B FF 03 add 88 01 xx 7D 7D 举例: 1. 存储当前 OSD 显示到预设 1 号码: 7B 7B 88 03 00 80 01 7D 7D 7D 存储成功, 返回命令为: 7B 7B 88 03 00 80 01 7D 7D 7D 存储失败, 返回命令为: 7B 7B FF 03 00 88 00 7A 7D 7D					

O S D 预 设

0x89: OSD 尺寸

命令头 (2 字节)	命令码 (1 字节)	命令体长度 (1 字节)	命令体 (CMD_LENGTH 字节)	校验和 (1 字节)	命令尾 (2 字节)
CMD_HEAD	CMD_INDEX	CMD_LENGTH	CMD_BODY	CMD_CHECKSUM	CMD_TAIL
{} 7B 7B	89	04	见命令体格式		}} 7D 7D
<p>命令体格式: Address(1Byte) + Ctrl(1Byte) + Width (1Byte) + Height (1Byte)</p> <p>1. Address: 目标地址</p> <p>2. Ctrl 值的 bit7: 1-表示读取 OSD 尺寸, 0-表示设置 OSD 尺寸; Ctrl 值的 Bit6: 1-表示板卡地址为输入板卡; 0-表示板卡地址为输出板卡; Ctrl 值的 bit[5,0]: 保留未使用;</p> <p>3. Width: OSD 宽度, 单位为 16 像素, 应确保 Width*16 小于等于 1920;</p> <p>4. Height: OSD 高度, 单位字体行数;</p> <p>命令返回说明:</p> <p>1. 与发送的命令格式相同;</p> <p>2. 设置命令成功时, 返回数据与发送数据相同; 设置失败时, 返回 7B 7B FF 03 add 89 00 xx 7D 7D</p> <p>3. 查询命令成功时, 返回数据与查询内容相关; 查询失败时, 返回 7B 7B FF 03 add 89 01 xx 7D 7D</p> <p>举例:</p> <p>1. 读取 (查询) OSD 尺寸: 7B 7B 89 04 00 80 00 00 FD 7D 7D 读取 (查询) 成功, 返回命令为: 7B 7B 89 04 00 80 20 02 1F 7D 7D 读取 (查询) 失败, 返回命令为: 7B 7B FF 04 00 89 00 7C 7D 7D`</p> <p>2. 设置 OSD 尺寸: 7B 7B 89 04 00 00 20 02 9F 7D 7D 设置成功, 返回命令为: 7B 7B 89 04 00 00 20 02 9F 7D 7D 设置失败, 返回命令为: 7B 7B FF 03 00 89 01 7B 7D 7D</p>					

O
S
D
尺
寸

0x90: 设置屏幕进入拼接

命令头 (2 字节)	命令码 (1 字节)	命令体长度 (1 字节)	命令体 (CMD_LENGTH 字节)	校验和 (1 字节)	命令尾 (2 字节)
CMD_HEAD	CMD_INDEX	CMD_LENGTH	CMD_BODY	CMD_CHECKSUM	CMD_TAIL
{} 7B 7B	90	06 或 09	见命令体格式		}} 7D 7D
<p>命令体格式: Address(1Byte) + Ctrl(1Byte) + Line + Column + P + Q + 【bSyncLock + H-Adjust + V-Adjust】 (各个变量均为一个字节)</p> <p>1. Address: 目标地址</p> <p>2. Ctrl 值的 bit7: 1-表示读取拼接参数, 0-表示设置拼接; Ctrl 值的 Bit6: 1-表示板卡地址为输入板卡; 0-表示板卡地址为输出板卡; Ctrl 值的 bit[5,0]: 保留未使用;</p> <p>3. Line:拼接屏幕一共的行数</p> <p>4. Column: 拼接屏幕一共的列数</p> <p>5. P: 当前屏幕所在的行数</p> <p>6. Q: 当前屏幕所在的列数</p> <p>7. bSyncLock: 同步锁开关, 1-开; 0-关</p> <p>8. H-Adjust: 水平方向调整值</p> <p>9. V-Adjust: 垂直方向调整值</p> <p>命令返回说明:</p> <p>1. 与发送的命令格式相同;</p> <p>2. 设置命令成功时, 返回数据与发送数据相同; 设置失败时, 返回 7B 7B FF 03 add 90 00 xx 7D 7D</p> <p>3. 查询命令成功时, 返回数据与查询内容相关; 查询失败时, 返回 7B 7B FF 03 add 90 01 xx 7D 7D</p> <p>4. 当 Line 和 Column 参数值均为 1 时, 此时为设置退出拼接</p> <p>举例:</p> <p>1. 设置 2*2 的拼接, 当前屏幕为 (1,1) : 7B 7B 90 06 00 00 02 02 01 01 8C 7D 7D 设置成功, 返回命令为: 7B 7B 90 06 00 00 02 02 01 01 8C 7D 7D 设置失败, 返回命令为: 7B 7B FF 03 00 90 00 82 7D 7D</p>					

OSD 预 设

0x91: 拼接屏幕的边沿调整

命令头 (2 字节)	命令码 (1 字节)	命令体长度 (1 字节)	命令体 (CMD_LENGTH 字节)	校验和 (1 字节)	命令尾 (2 字节)
CMD_HEAD	CMD_INDEX	CMD_LENGTH	CMD_BODY	CMD_CHECKSUM	CMD_TAIL
{} 7B 7B	91	6	见命令体格式		}} 7D 7D
<p>命令体格式: Address(1Byte) + Ctrl(1Byte) + Left + Right + Top + Bottom (各个变量均为一个字节)</p> <p>1. Address: 目标地址</p> <p>2. Ctrl 值的 bit7: 1-表示读取当前屏幕的边沿调整参数, 0-表示设置当前屏幕的边沿调整;</p> <p>Ctrl 值的 Bit6: 1-表示板卡地址为输入板卡; 0-表示板卡地址为输出板卡;</p> <p>Ctrl 值的 bit[5,0]: 保留未使用;</p> <p>3. Left: 左边沿的值, 单位像素</p> <p>4. Right: 右边沿的值, 单位像素 (目前该参数不使用, 表示与 Left 值相同)</p> <p>5. Top: 上边沿的值, 单位像素</p> <p>6. Bottom: 下边沿的值, 单位像素 (目前该参数不使用, 表示与 Top 值相同)</p> <p>命令返回说明:</p> <p>1. 与发送的命令格式相同;</p> <p>2. 设置命令成功时, 返回数据与发送数据相同; 设置失败时, 返回 7B 7B FF 03 add 91 00 xx 7D 7D</p> <p>3. 查询命令成功时, 返回数据与查询内容相关; 查询失败时, 返回 7B 7B FF 03 add 91 01 xx 7D 7D</p> <p>举例:</p> <p>1. 设置边沿调整参数: 7B 7B 91 06 00 00 00 00 00 00 89 7D 7D 设置成功, 返回命令为: 7B 7B 91 06 00 00 00 00 00 00 89 7D 7D 设置失败, 返回命令为: 7B 7B FF 03 00 91 00 7B 7D 7D</p>					

O S D 预 设

0x92: 拼接 SyncLock 命令

命令头 (2 字节)	命令码 (1 字节)	命令体长度 (1 字节)	命令体 (CMD_LENGTH 字节)	校验和 (1 字节)	命令尾 (2 字节)
CMD_HEAD	CMD_INDEX	CMD_LENGTH	CMD_BODY	CMD_CHECKSUM	CMD_TAIL
{} 7B 7B	92	3	见命令体格式		}} 7D 7D
<p>命令体格式: Address(1Byte) + Ctrl(1Byte) + Flag (1Byte)</p> <p>1. Address: 目标地址</p> <p>2. Ctrl 值的 bit7: 1-表示读取拼接 Lock 的开关值, 0-表示设置拼接 Lock 的开关值;</p> <p>Ctrl 值的 Bit6: 1-表示板卡地址为输入板卡; 0-表示板卡地址为输出板卡;</p> <p>Ctrl 值的 bit[5,0]: 保留未使用;</p> <p>3. Flag: 1-SyncLock; 0-Remove SyncLock</p> <p>命令返回说明:</p> <p>1. 与发送的命令格式相同;</p> <p>2. 设置命令成功时, 返回数据与发送数据相同; 设置失败时, 返回 7B 7B FF 03 add 91 00 xx 7D 7D</p> <p>3. 查询命令成功时, 返回数据与查询内容相关; 查询失败时, 返回 7B 7B FF 03 add 91 01 xx 7D 7D</p> <p>举例:</p> <p>1. 设置边沿调整参数: 7B 7B 91 06 00 00 00 00 00 00 89 7D 7D 设置成功, 返回命令为: 7B 7B 91 06 00 00 00 00 00 00 89 7D 7D 设置失败, 返回命令为: 7B 7B FF 03 00 91 00 7B 7D 7D</p>					

O S D 预 设

0x93: 拼接数据存取命令

命令头 (2 字节)	命令码 (1 字节)	命令体长度 (1 字节)	命令体 (CMD_LENGTH 字节)	校验和 (1 字节)	命令尾 (2 字节)
CMD_HEAD	CMD_INDEX	CMD_LENGTH	CMD_BODY	CMD_CHECKSUM	CMD_TAIL
{{					}}
7B 7B	93	3+4*N	见命令体格式		7D 7D

命令体格式: Address(1Byte) + Ctrl(1Byte) + Data (3Byte+4*N Byte, 其中 N 的值为拼接墙屏幕的有效个数)

- Address: 目标地址 (不使用, 主板处理)
- Ctrl 值的 bit7: 1-表示读取拼接数据,
0-表示存储拼接数据;
Ctrl 值的 bit[6,0]: 保留未使用;
- Data: 拼接数据, 前面为 1 个 TVWALL_HEAD (3 字节), 后面为 N 个 TVWALL_UNIT (4 字节), 具体数据结构如下:

```

typedef struct __TVWALL_HEAD__
{
    unsigned char u8NUM_Valid; // 拼接墙屏幕的有效个数 (有效个数 <= 行数 * 列数)
    unsigned char u8NUM_Line; // 拼接墙屏幕的行数
    unsigned char u8NUM_Column; // 拼接墙屏幕的列数
} TVWALL_HEAD;

typedef struct __TVWALL_UNIT__
{
    unsigned char bFlag; // bit0- bCombine; bit1- bSyncLock
    unsigned char u8FirstCombineID; // M*N 个拼接屏幕的第一个屏幕的索引 ID
    unsigned char u8OutputPort; // 输出端口
    unsigned char u8AdjustValue; // 高4位为水平边缘调整值, 低4位为垂直边缘调整值
} TVWALL_UNIT;
    
```

命令返回说明:

- 与发送的命令格式相同;
- 当返回参数中 TVWALL_HEAD 结构体的值均为 0 时, 说明主板未保存过有效数据

OSD 预设

0xB0: 设置、读取端子类型

命令头 (2 字节)	命令码 (1 字节)	命令体长度 (1 字节)	命令体 (CMD_LENGTH 字节)	校验和 (1 字节)	命令尾 (2 字节)
CMD_HEAD	CMD_INDEX	CMD_LENGTH	CMD_BODY	CMD_CHECKSUM	CMD_TAIL
7B 7B	B0	03	见命令体格式		7D 7D

命令体格式: Address(1Byte) + Ctrl(1Byte) + OutputType (1Byte)

Address: 板卡地址

Ctrl:

bit7: 1-表示**读取**端子类型;
0-表示**设置**端子类型;

Bit6: 1-表示板卡地址为**输入板卡**;
0-表示板卡地址为**输出板卡**;

bit5: 1-表示操作板卡的**输入端子类型**
0-表示操作板卡的**输出端子类型**

bit[4,0]: 保留未使用;

OutputType: 端子类型, 值如下: (命令发送端发送读取命名时, 接收端会忽略该字节的值)

- 0, // FIBER (只可以读取, 不可以设置)
- 1, // SDI (只可以读取, 不可以设置)
- 2, // HDBST (只可以读取, 不可以设置)
- 3, // DVI
- 4, // CVBS
- 5, // YPbPr
- 6, // VGA
- 7, // HDMI
- 8, // HDMI 直通 (只可以读取, 不可以设置)

命令返回说明:

与发送的命令格式相同;

设置命令成功时, 返回数据与发送数据相同; 设置失败时, 返回 7B 7B FF 03 **add B0 00** xx 7D 7D

查询命令成功时, 返回数据与查询内容相关; 查询失败时, 返回 7B 7B FF 03 **add B0 01** xx 7D 7D

举例:

读取 (查询) 输出板卡的输出端子类型: 7B 7B **B0 03 00 80 00** 23 7D 7D

读取 (查询) 成功, 返回命令为: 7B 7B **B0 03 00 80 07** 24 7D 7D (读取结果为 HDMI)

读取 (查询) 失败, 返回命令为: 7B 7B **FF 03 00 B0 01** A3 7D 7D

设置输出板卡的输出端子类型: 7B 7B **B0 03 00 00** 07 A4 7D 7D (设置为 HDMI)

设置成功, 返回命令为: 7B 7B **B0 03 00 00** 07 A4 7D 7D

设置失败, 返回命令为: 7B 7B **FF 03 00 B0 01** A3 7D 7D

设置读取输出板的输出端子类型

0xB1: 设置、读取分辨率

命令头 (2 字节)	命令码 (1 字节)	命令体长度 (1 字节)	命令体 (CMD_LENGTH 字节)	校验和 (1 字节)	命令尾 (2 字节)
CMD_HEAD	CMD_INDEX	CMD_LENGTH	CMD_BODY	CMD_CHECKSUM	CMD_TAIL
{{					}}
7B 7B	B1	03	见命令体格式		7D 7D
<p>命令体格式: Address(1Byte) + Ctrl(1Byte) + OutputResolution (1Byte)</p> <p>Address: 板卡地址</p> <p>Ctrl:</p> <p>bit7: 1-表示读取分辨率; 0-表示设置分辨率;</p> <p>Bit6: 1-表示板卡地址为输入板卡; 0-表示板卡地址为输出板卡;</p> <p>bit5: 1-表示操作板卡的输入分辨率 0-表示操作板卡的输出分辨率</p> <p>bit[4,0]: 保留未使用;</p> <p>OutputResolution: 输出分辨率, 值如下: (命令发送端发送读取命名时, 接收端会忽略该字节的值)</p> <p>当为输入板卡, 且为输入分辨率时:</p> <p>此情况, 该协议未实现; 使用其他 index=0xB6 实现;</p> <p>当为输入板卡, 且为输出分辨率时:</p> <p>当为输出板卡, 且为输入分辨率时:</p> <p>0: "1024x768@60HZ", 1: "1280x720@60HZ", 2: "1280x1024@60HZ", //3: "1366x768@60HZ", //4: "1400x1050@60HZ", //5: "1440x900@60HZ", 3: "1600x1200@60HZ", 4: "1680x1050@60HZ", 5: "1920x1080@50HZ", 6"1920x1080@60HZ", //10: "1920x1200@60HZ",</p> <p>当为输出板卡, 且为输出分辨率时:</p> <p>当端子类型为 VGA/HDMI/DVI/HDBST 时:</p> <p>0: EM_OUTPUT_MODE_1024X768_60,// 0 1: EM_OUTPUT_MODE_1280X720_60,// 1 2: EM_OUTPUT_MODE_1280X1024_60,// 2 3: EM_OUTPUT_MODE_1360X768_60,// 3 4: EM_OUTPUT_MODE_1600X1200_60,// 4 5: EM_OUTPUT_MODE_1680X1050_60,// 5 6: EM_OUTPUT_MODE_1920X1080_30,// 6 7: EM_OUTPUT_MODE_1920X1080_60,// 7</p> <p>当端子类型为 CVBS 时:</p> <p>0: NTSC</p>					

设置读取输出板的输出端子类型

串口通讯协议

1: PAL

当端子类型为 YPbPr 时:

0:720P 60HZ

1:1080P60HZ

当端子类型为 SDI 时:

0:720P 60HZ

1:1980P 30HZ

2:1920P 60HZ

当端子类型为 HDMI 直通输出板卡时:

此情况, 该协议未实现; 使用其他 index=0xB7 实现返回分辨率数据;

命令返回说明:

与发送的命令格式相同;

设置命令成功时, 返回数据与发送数据相同; 设置失败时, 返回 7B 7B FF 03 **add** B1 00 xx 7D 7D

查询命令成功时, 返回数据与查询内容相关; 查询失败时, 返回 7B 7B FF 03 **add** B1 01 xx 7D 7D

举例:

读取 (查询) 输出板卡的输出分辨率: 7B 7B **B1** 03 **00** 80 00 24 7D 7D

读取 (查询) 成功, 返回命令为: 7B 7B **B1** 03 **00** 80 01 25 7D 7D

读取 (查询) 失败, 返回命令为: 7B 7B **FF** 03 **00** B1 00 A3 7D 7D

设置输出板卡的输出分辨率: 7B 7B **B1** 03 **00** 00 01 A5 7D 7D

设置成功, 返回命令为: 7B 7B **B1** 03 **00** 00 01 A5 7D 7D

设置失败, 返回命令为: 7B 7B **FF** 03 **00** B1 01 A4 7D 7D

0xB6: 设置、读取分辨率 (仅输入板卡&&输入分辨率)

命令头 (2 字节)	命令码 (1 字节)	命令体长度 (1 字节)	命令体 (CMD_LENGTH 字节)	校验和 (1 字节)	命令尾 (2 字节)
CMD_HEAD	CMD_INDEX	CMD_LENGTH	CMD_BODY	CMD_CHECKSUM	CMD_TAIL
7B 7B	B1	08	见命令体格式		7D 7D
<p>命令体格式: Address(1Byte) + Ctrl(1Byte) + Hori_H + Hori_L + Vert_H + Vert_L + bProgressive + Frequency</p> <p>Address: 板卡地址</p> <p>Ctrl:</p> <p>bit7: 1-表示读取分辨率; 0-表示设置分辨率;</p> <p>Bit6: 1-表示板卡地址为输入板卡; 0-表示板卡地址为输出板卡;</p> <p>bit5: 1-表示操作板卡的输入分辨率 0-表示操作板卡的输出分辨率</p> <p>bit[4,0]: 保留未使用;</p> <p>OutputResolution: 输出分辨率, 值如下: (命令发送端发送读取命名时, 接收端会忽略该字节的值)</p> <p>当为输入板卡, 且为输入分辨率时:</p> <p>Hori_H: 水平分辨率的高 8 位值 (1Byte); Hori_L: 水平分辨率的高 8 位值 (1Byte); Vert_H: 垂直分辨率的高 8 位值 (1Byte); Vert_L: 垂直分辨率的高 8 位值 (1Byte);</p> <p>bProgressive: 1-逐行; 0-隔行 (interlace)</p> <p>Frequency: 频率;</p> <p>当为输入板卡, 且为输出分辨率时:</p> <p>当为输出板卡, 且为输入分辨率时:</p> <p>当为输出板卡 (非 HDMI 直通板卡), 且为输出分辨率时:</p> <p>此情况, 该协议未实现; 使用其他 index=0xB1 实现;</p> <p>命令返回说明:</p> <p>与发送的命令格式相同;</p> <p>设置命令成功时, 返回数据与发送数据相同; 设置失败时, 返回 7B 7B FF 03 add B6 00 xx 7D 7D</p> <p>查询命令成功时, 返回数据与查询内容相关; 查询失败时, 返回 7B 7B FF 03 add B6 01 xx 7D 7D</p> <p>举例:</p>					

设置读取输出板的输出端子类型

0xB2:设置和读取输出板颜色设置:

命令头 (2 字节)	命令码 (1 字节)	命令体长度 (1 字节)	命令体 (CMD_LENGTH 字节)	校验和 (1 字节)	命令尾 (2 字节)
CMD_HEAD	CMD_INDEX	CMD_LENGTH	CMD_BODY	CMD_CHECKSUM	CMD_TAIL
{} 7B 7B	B2	03	见命令体格式		}} 7D 7D
<p>命令体格式: Address(1Byte) + Ctrl(1Byte) + Brightness (1Byte)</p> <p>Address: 目标地址</p> <p>Ctrl:</p> <p>bit[7]: 1-读取 0-设置</p> <p>Bit6: 1-表示板卡地址为输入板卡; 0-表示板卡地址为输出板卡;</p> <p>bit[5,4]:</p> <p>00:加 1 01:减 1 10: 绝对值设置 11:复位到出厂设置</p> <p>bit[3,0]:</p> <p>0000: 亮度 0001: 对比度 0010: 饱和度 0011: 锐度 0100: 色调 0101: R-Gain 0110: G-Gain 0111: B-Gain 1000: R-Offset 1001: G-Offset 1010: B-Offset</p> <p>颜色:</p> <p>命令发送端发送读取命名时, 接收端会忽略该字节的值;</p> <p>命令返回说明:</p> <p>与发送的命令格式相同;</p> <p>设置命令成功时, 返回数据与发送数据相同; 设置失败时, 返回 7B 7B FF 03 add B2 00 xx 7D 7D 查询命令成功时, 返回数据与查询内容相关; 查询失败时, 返回 7B 7B FF 03 add B2 01 xx 7D 7D</p> <p>举例:</p> <p>待定</p> <p>读取 (查询) 亮度: 7B 7B B2 03 00 80 00 24 7D 7D 读取 (查询) 成功, 返回命令为: 7B 7B B2 03 00 80 01 26 7D 7D 读取 (查询) 失败, 返回命令为: 7B 7B FF 03 00 B1 00 A4 7D 7D`</p> <p>设置亮度: 7B 7B B2 03 00 00 01 A6 7D 7D 设置成功, 返回命令为: 7B 7B B2 03 00 00 01 A6 7D 7D 设置失败, 返回命令为: 7B 7B FF 03 B2 01 A5 7D 7D</p>					

设置和读取输出板颜色设置

0xB3:设置和读取输出板位置:

命令头 (2 字节)	命令码 (1 字节)	命令体长度 (1 字节)	命令体 (CMD_LENGTH 字节)	校验和 (1 字节)	命令尾 (2 字节)
CMD_HEAD	CMD_INDEX	CMD_LENGTH	CMD_BODY	CMD_CHECKSUM	CMD_TAIL
{} 7B 7B	 B3	 03	 见命令体格式		 }} 7D 7D
<p>命令体格式: Address(1Byte) + Ctrl(1Byte) + Value (1Byte)</p> <p>Address: 目标地址</p> <p>Ctrl:</p> <p>bit[7]:</p> <ul style="list-style-type: none"> 1: 读取 0: 设置 <p>Bit6: 1-表示板卡地址为输入板卡; 0-表示板卡地址为输出板卡;</p> <p>bit[5,4]:</p> <ul style="list-style-type: none"> 00:加 1 01:减 1 10: 绝对值设置 11: 复位到出厂设置 <p>bit[3,0]:</p> <ul style="list-style-type: none"> 0000: HPostion 0001: VPostion 0010: HSize 0011: VSize <p>颜色:</p> <p>命令发送端发送读取命名时, 接收端会忽略该字节的值;</p> <p>命令返回说明:</p> <p>与发送的命令格式相同;</p> <p>设置命令成功时, 返回数据与发送数据相同; 设置失败时, 返回 7B 7B FF 03 add B3 00 xx 7D 7D</p> <p>查询命令成功时, 返回数据与查询内容相关; 查询失败时, 返回 7B 7B FF 03 add B3 01 xx 7D 7D</p> <p>举例:</p> <p>待定</p> <p>读取 (查询) 亮度: 7B 7B B2 03 00 80 00 24 7D 7D</p> <p>读取 (查询) 成功, 返回命令为: 7B 7B B2 03 00 80 01 26 7D 7D</p> <p>读取 (查询) 失败, 返回命令为: 7B 7B FF 03 00 B1 00 A4 7D 7D`</p> <p>设置亮度: 7B 7B B2 03 00 00 01 A6 7D 7D</p> <p>设置成功, 返回命令为: 7B 7B B2 03 00 00 01 A6 7D 7D</p> <p>设置失败, 返回命令为: 7B 7B FF 03 B2 01 A5 7D 7D</p>					

设置和读取输出板位置

0xB4: 无缝切换

命令头 (2 字节)	命令码 (1 字节)	命令体长度 (1 字节)	命令体 (CMD_LENGTH 字节)	校验和 (1 字节)	命令尾 (2 字节)
CMD_HEAD	CMD_INDEX	CMD_LENGTH	CMD_BODY	CMD_CHECKSUM	CMD_TAIL
{{					}}
7B 7B	B4	01	见命令体格式		7D 7D
<p>命令体格式: Address(1Byte) Address: 目标地址 Ctrl: bit[7]: 保留未使用 Bit6: 1-表示板卡地址为输入板卡; 0-表示板卡地址为输出板卡; bit[5,0]: 保留未使用</p> <p>输出板卡收到该命令后执行无缝切换动作（主板必须在切换 crosspoint 前发送该命令，然后延时 50-150MS） 然后切换 CrossPoint。</p> <p>命令返回说明: 与发送的命令格式相同; 设置命令成功时，返回数据与发送数据相同；设置失败时，返回 7B 7B FF 03 add B3 00 xx 7D 7D 查询命令成功时，返回数据与查询内容相关；查询失败时，返回 7B 7B FF 03 add B3 01 xx 7D 7D</p> <p>举例: 待定 读取（查询）亮度：7B 7B B2 03 00 80 00 24 7D 7D 读取（查询）成功，返回命令为：7B 7B B2 03 00 80 01 26 7D 7D 读取（查询）失败，返回命令为：7B 7B FF 03 00 B1 00 A4 7D 7D` 设置亮度：7B 7B B2 03 00 00 01 A6 7D 7D 设置成功，返回命令为：7B 7B B2 03 00 00 01 A6 7D 7D 设置失败，返回命令为：7B 7B FF 03 B2 01 A5 7D 7D</p>					

无缝切换

0xB5: VGA AutoConfig(仅 VGA 输入有效):

命令头 (2 字节)	命令码 (1 字节)	命令体长度 (1 字节)	命令体 (CMD_LENGTH 字节)	校验和 (1 字节)	命令尾 (2 字节)
CMD_HEAD	CMD_INDEX	CMD_LENGTH	CMD_BODY	CMD_CHECKSUM	CMD_TAIL
{} 7B 7B	B5	03	见命令体格式		}} 7D 7D
<p>命令体格式: Address(1Byte) + Ctrl(1Byte) + Flag (1Byte)</p> <p>Address: 目标地址</p> <p>Ctrl: bit7: 1: 读取 0: 设置</p> <p>bit6: 1-表示板卡地址为输入板卡; 0-表示板卡地址为输出板卡;</p> <p>bit[5,0]: 保留未使用</p> <p>Flag: 0-OFF, 1-ON</p> <p>命令返回说明: 与发送的命令格式相同; 设置命令成功时, 返回数据与发送数据相同; 设置失败时, 返回 7B 7B FF 03 add B5 00 xx 7D 7D 查询命令成功时, 返回数据与查询内容相关; 查询失败时, 返回 7B 7B FF 03 add B5 01 xx 7D 7D</p> <p>举例: 7B 7B B5 03 00 00 01 xx 7D 7D</p>					

设置和读取输出板位置

0xB7: 设置、读取信号数据

命令头 (2 字节)	命令码 (1 字节)	命令体长度 (1 字节)	命令体 (CMD_LENGTH 字节)	校验和 (1 字节)	命令尾 (2 字节)
CMD_HEAD	CMD_INDEX	CMD_LENGTH	CMD_BODY	CMD_CHECKSUM	CMD_TAIL
{} 7B 7B	 B7	 07	 见命令体格式		 }} 7D 7D
<p>命令体格式 1: Address(1Byte) + Ctrl(1Byte) + DataValid + InType + OutType + InResolution + OutResolution 命令体格式 2: Address(1Byte) + Ctrl(1Byte) + Hori_H + Hori_L + Vert_H + Vert_L + P_Frequency</p> <p>Address: 板卡地址</p> <p>Ctrl: bit7: 1-表示读取信号数据; 0-表示设置信号数据; bit6: 1-表示板卡地址为输入板卡; //目前不支持输入板卡 0-表示板卡地址为输出板卡; //目前只支持输出板卡 bit5: 0-使用命令体格式 1, 1-使用命令体格式 2 (命令体格式 2 仅用于输入板卡的输入分辨率) bit[4,0]: 保留未使用;</p> <p>命令体格式 1 参数说明如下:</p> <p>DataValid: bit[7-4]: 保留未使用 bit3: 0-InType 数据无效 (无效时不操作), 1-InType 数据有效 bit2: 0-OutType 数据无效 (无效时不操作), 1-OutType 数据有效 bit1: 0-InResolution 数据无效 (无效时不操作), 1-InResolution 数据有效 bit0: 0-OutResolution 数据无效 (无效时不操作), 1-OutResolution 数据有效</p> <p>InType: 输入信号的端子类型 OutType: 输出信号的端子类型</p> <p>0, // FIBER 1, // SDI 2, // HDBST 3, // DVI 4, // CVBS 5, // YPbPr 6, // VGA 7, // HDMI 8, // HDMI 直通</p> <p>InResolution: 输入信号的分辨率 OutResolution: 输出信号的分辨率</p> <p>1. 输入板卡, 不支持;</p> <p>2. 输出板卡的输入分辨率, 取值如下:</p> <p>0: "1024x768@60HZ", 1: "1280x720@60HZ", 2: "1280x1024@60HZ", //3: "1366x768@60HZ", //4: "1400x1050@60HZ", //5: "1440x900@60HZ", 3: "1600x1200@60HZ",</p>					

设置读取输出板的输出端子类型

4: "1680x1050@60HZ",

5: "1920x1080@50HZ",

6: "1920x1080@60HZ",

//10: "1920x1200@60HZ",

3. 输出板卡的输出分辨率，取值如下：

当端子类型为 VGA/HDMI/DVI/HDBST 时：

- 0: EM_OUTPUT_MODE_1024X768_60,// 0
- 1: EM_OUTPUT_MODE_1280X720_60,// 1
- 2: EM_OUTPUT_MODE_1280X1024_60,// 2
- 3: EM_OUTPUT_MODE_1360X768_60,// 3
- 4: EM_OUTPUT_MODE_1600X1200_60,// 4
- 5: EM_OUTPUT_MODE_1680X1050_60,// 5
- 6: EM_OUTPUT_MODE_1920X1080_30,// 6
- 7: EM_OUTPUT_MODE_1920X1080_60,// 7

当端子类型为 CVBS 时：

- 0: NTSC
- 1: PAL

当端子类型为 YPbPr 时：

- 0:720P 60HZ
- 1:1080P60HZ

当端子类型为 SDI 时：

- 0:720P 60HZ
- 1:1980P 30HZ
- 2:1920P 60HZ

当端子类型为 HDMI 直通板卡时：

可以使用 **该条指令的格式 2** 进行返回分辨率数据；

命令体格式 2 的说明如下：（可用于 HDMI 直通输出板卡的分辨率）

Hori_H: 水平分辨率的高 8 位值（1Byte）；

Hori_L: 水平分辨率的高 8 位值（1Byte）；

Vert_H: 垂直分辨率的高 8 位值（1Byte）；

Vert_L: 垂直分辨率的高 8 位值（1Byte）；

P_Frequency 的 bit8: 表示 bProgressive -> 1-逐行； 0-隔行（interlace）

P_Frequency 的 bit[7:0]: 表示 Frequency -> 频率值（取值 0x00-0x7F）；

命令返回说明：

与发送的命令格式相同；

设置命令成功时，返回数据与发送数据相同；设置失败时，返回 7B 7B FF 03 **add B7** 00 xx 7D 7D

查询命令成功时，返回数据与查询内容相关；查询失败时，返回 7B 7B FF 03 **add B7** 01 xx 7D 7D

举例：

0xB8: EDID 操作

命令头 (2 字节)	命令码 (1 字节)	命令体长度 (1 字节)	命令体 (CMD_LENGTH 字节)	校验和 (1 字节)	命令尾 (2 字节)
CMD_HEAD	CMD_INDEX	CMD_LENGTH	CMD_BODY	CMD_CHECKSUM	CMD_TAIL
{{					}}
7B 7B	B8	0x13	见命令体格式		7D 7D

命令体格式: Address(1Byte) + Ctrl(1Byte) + Offset (1Byte) + EDID(16Bytes)

Address: 目标地址

Ctrl: bit7: 1: 读取
0: 设置

bit6: 1-表示板卡地址为输入板卡;
0-表示板卡地址为输出板卡;

bit[5,0]: 保留未使用

Offset: 本条命令中的 EDID 内容在整个 EDID 中的偏移量

EDID: 本条命令中的 EDID 内容

NOTE:

1. 输入板卡只可以设置 EDID, 读取 EDID 则返回错误
2. 输出板卡只可以读取 EDID, 设置 EDID 则返回错误

命令返回说明:

与发送的命令格式相同;

设置命令成功时, 返回数据与发送数据相同; 设置失败时, 返回 7B 7B FF 03 add B8 00 xx 7D 7D

查询命令成功时, 返回数据与查询内容相关; 查询失败时, 返回 7B 7B FF 03 add B8 01 xx 7D 7D

举例:

设置和读取端口的 EDID

0xB9: 直通板卡设置输出类型

命令头 (2 字节)	命令码 (1 字节)	命令体长度 (1 字节)	命令体 (CMD_LENGTH 字节)	校验和 (1 字节)	命令尾 (2 字节)
CMD_HEAD	CMD_INDEX	CMD_LENGTH	CMD_BODY	CMD_CHECKSUM	CMD_TAIL
7B 7B	B9	03	见命令体格式		7D 7D
<p>命令体格式: Address(1Byte) + Ctrl(1Byte) + Value (1Byte)</p> <p>Address: 目标地址</p> <p>Ctrl: bit7: 1: 读取 0: 设置</p> <p>bit6: 1-表示板卡地址为输入板卡 (暂时不支持该功能); 0-表示板卡地址为输出板卡;</p> <p>bit[5,0]: 保留未使用</p> <p>Value: 0-DVI (暂时无效) 1- HDMI 2- HDMI-HDCP</p> <p>NOTE:</p> <ol style="list-style-type: none"> 1. 输入板卡则返回错误 2. 输出板卡可读取可设置 <p>命令返回说明:</p> <p>与发送的命令格式相同;</p> <p>设置命令成功时, 返回数据与发送数据相同; 设置失败时, 返回 7B 7B FF 03 add B9 00 xx 7D 7D</p> <p>查询命令成功时, 返回数据与查询内容相关; 查询失败时, 返回 7B 7B FF 03 add B9 01 xx 7D 7D</p> <p>举例:</p>					

直通板卡设置输出类型