

类别	内容
关键词	组态功能串口屏指令集
摘要	大彩串口屏指令集包括组态指令集和基本指令集

## 修订历史

版本	日期	原因	编制	审查
V0.9	2012/05/18	编著文档	刘仁武	李勇
V1.0	2012/12/15	添加组态控件功能，完善功能描述	刘仁武	李勇
V2.0	2013/10/25	增添了动画、图标和曲线控件，修正一些错误参数	刘仁武	李勇
V3.0	2014/07/01	增添了中文输入法、按钮互斥、滑动控件、指令 CRC 校验	刘仁武	李勇
V4.0	2015/10/01	增添了下拉菜单、滑动选择控件、二维码控件、文本控件颜色改变、批量文本更新、新增 RS485、带外壳和电容屏系列	刘仁武	李勇
V4.1	2017/09/14	增添了数据记录控件指令集（常规记录添加与删除、选中某一行、对于某一行进行添加和删除记录等）以及将当前画面导出到 SD 卡指令、调节文本数据增量的指令、视频控件指令和网络设置指令	刘仁武	李勇
V4.2	2017/11/03	添加了相应指令的屏幕适用范围	刘仁武	李勇
V4.3	2018/06/01	添加获取设备版本、音频文件播放、停止、音量调节、本地/外部 AV 输入视频、显示/隐藏系统键盘、触摸屏灵敏度调节等指令	谢岢成	刘启鑫
V4.4	2018/6/09	添加读取图标控件帧 ID 值、读取选择控件选项值指令	谢岢成	李勇
V4.5	2018/7/17	添加使屏幕进入待机和激活状态指令	谢岢成	李勇
V4.6	2018/7/23	修正曲线控件显示指令的错误描述	谢岢成	李勇
V4.7	2018/8/16	增加设置图标位置、获取当前语言 ID、旋转控件增加小角度功能说明	刘启鑫	李勇
V4.8	2018/8/28	增加播放视频与音频、替换开机 logo、替换系统提示音的功能说明	林青田	刘启鑫
V4.9	2018/9/17	添加开机音乐和按键提示音	林青田	刘启鑫
V5.0	2018/9/27	增加播放音频指令 重新整理组态指令和基本指令 调整指令顺序，修改部分格式问题	林青田	刘启鑫
V5.1	2020/6/24	AV 视频输入指令注明为旧版本型号使用 添加使用动画效果切换画面的指令 新添加休眠、休眠唤醒、AV 视频信号切换、切换子画面、子画面窗口上下画面切换、绘图画布、基本绘图等指令。 修改视频音频指令适用于 M 型	林青田	刘启鑫



## 销售与服务

### 广州大彩光电科技有限公司

电话：020-82186683

传真：020-82187676

Email: hmi@gz-dc.com（公共服务）

网站：www.gz-dc.com

地址：广州高新技术产业开发区玉树工业园富康西街 8 号 C 栋 303 房

官网零售淘宝店：<https://gz-dc.taobao.com>

# 目录

1. 指令格式和指令集.....	8
1.1 指令格式.....	8
1.1.1 指令操作举例说明.....	8
1.2 颜色格式.....	14
1.3 组态指令集与基本指令集区别.....	14
1.4 如何快速熟悉指令集.....	14
1.5 组态指令集表.....	15
1.6 基本指令集表.....	33
2. 组态指令集详述.....	39
2.1 握手.....	39
2.2 复位报告.....	39
2.3 复位设备.....	39
2.4 获取设备版本.....	39
2.5 背光调节.....	39
2.6 自动屏保模式.....	39
2.7 蜂鸣器控制.....	40
2.8 配置触摸屏.....	40
2.9 触摸屏校准.....	41
2.10 触摸屏体验.....	41
2.11 设置波特率.....	41
2.12 切换画面.....	42
2.13 使用动画效果切换画面.....	42
2.14 读取画面.....	42
2.15 按钮控件ID值上传.....	43
2.16 设置按钮弹起或按下状态.....	46
2.17 对内指令实现按钮互斥和状态显示.....	47
2.18 对外指令输出.....	47
2.19 读取按钮控件状态.....	47
2.20 更新文本控件数值.....	48
2.21 清除文本控件内容.....	50
2.22 读取文本控件数值.....	50
2.23 设置光标焦点.....	50
2.24 设置文本控件闪烁.....	50
2.25 设置文本控件滚动速度.....	51
2.26 设置文本控件背景色.....	51
2.27 设置文本背景透明.....	51
2.28 设置文本控件前景色.....	51
2.29 格式化文本显示.....	52
2.30 文本控件数字增量调节指令.....	52
2.31 更新进度条控件数值.....	52
2.32 读取进度条控件值.....	53
2.33 滑动条控件上传格式.....	53

2.34	设置进度条的背景色.....	54
2.35	设置进度条的前景色.....	54
2.36	更新滑动条控件数值.....	54
2.37	读取滑动条控件值.....	54
2.38	RTC时钟设置.....	54
2.39	读取RTC时钟.....	55
2.40	更新仪表控件数值.....	55
2.41	读取仪表控件数值.....	55
2.42	动画控件显示.....	56
2.43	图标控件显示.....	57
2.44	读取图标控件数值.....	58
2.45	图标控件值上传.....	58
2.46	设置图标位置.....	58
2.47	批量更新控件数值.....	59
2.48	曲线控件显示.....	59
2.49	设置下拉菜单写入的文本控件.....	61
2.50	下拉菜单控件值上传.....	63
2.51	设置滑动选择控件值.....	64
2.52	读取滑动选择控件值.....	64
2.53	滑动选择控件值上传.....	64
2.54	更新二维码控件.....	66
2.55	启动定时器.....	67
2.56	设置定时器.....	67
2.57	启动定时器.....	67
2.58	暂停定时器.....	67
2.59	停止定时器.....	68
2.60	读取定时器.....	68
2.61	手动禁止/使能屏幕更新.....	68
2.62	显示/隐藏控件.....	68
2.63	RS485 屏地址设置.....	68
2.64	RS485 屏地址取消.....	68
2.65	锁定系统配置.....	69
2.66	解除系统配置锁定.....	69
2.67	告警事件触发.....	69
2.68	告警事件解除.....	69
2.69	数据记录控件-添加常规记录.....	70
2.70	数据记录控件-清除记录数据.....	70
2.71	数据记录控件-设置记录显示偏移.....	70
2.72	数据记录控件-获取当前记录数目.....	70
2.73	数据记录控件-读取某一行记录.....	70
2.74	数据记录控件-修改常规记录.....	71
2.75	数据记录控件-删除某一行记录.....	71
2.76	数据记录控件-插入常规记录.....	71
2.77	数据记录控件-选中某一行数据记录.....	71

2.78	数据记录控件-一次添加多条记录.....	71
2.79	数据记录控件-SD卡导出指令（CSV格式） .....	72
2.80	历史曲线设置采样值.....	72
2.81	历史曲线禁止/使能采样.....	72
2.82	历史曲线隐藏/显示通道.....	72
2.83	历史曲线设置时间长度（即采样点数） .....	72
2.84	历史曲线缩放到全屏.....	73
2.85	历史曲线设置缩放比例系数.....	73
2.86	历史曲线设置数值显示范围.....	73
2.87	历史曲线数据清空.....	73
2.88	旋转控件设置旋转角度.....	73
2.89	多语言切换指令.....	74
2.90	获取系统语言索引.....	74
2.91	视频相关指令(仅物联型、M型适用).....	74
2.91.1	添加视频路径.....	74
2.91.2	播放视频.....	74
2.91.3	视频暂停/恢复.....	75
2.91.4	视频停止.....	75
2.92	音频播放相关指令(仅物联型、F系列、M型适用).....	75
2.92.1	音频播放.....	75
2.92.2	停止音频播放.....	75
2.92.3	音频播放指令.....	75
2.92.4	暂停音频播放指令.....	75
2.92.5	恢复音频播放指令.....	75
2.92.6	停止音频播放指令.....	75
2.92.7	音频序列播放（适用于以数字命名的音频） .....	75
2.92.8	音量调节.....	75
2.93	矩阵键盘控制.....	76
2.94	写数据到FLASH.....	76
2.95	读取保存在FLASH中的数据 .....	76
2.96	将当前画面存储到TF卡 .....	77
2.97	保存无线网络设置.....	77
2.98	获取无线网络设置.....	77
2.99	获取网络状态.....	77
2.100	搜索无线网络.....	77
2.101	保存网络设置.....	78
2.102	获取网络设置.....	78
2.103	保存网络服务.....	78
2.104	获取网络服务.....	78
2.105	发送网络数据.....	78
2.106	发送网络数据（十六进制） .....	78
2.107	保存网络设置.....	79
2.108	本地/外部AV输入视频 .....	79
2.109	显示系统键盘.....	79

2.110	隐藏系统键盘.....	79
2.111	触摸屏灵敏度调节指令（仅用于电容屏）.....	79
2.112	进入待机状态.....	79
2.113	进入激活状态.....	80
2.114	进入休眠状态.....	80
2.115	退出休眠状态.....	80
2.116	切换AV输入信号.....	80
2.117	切换子画面.....	80
2.118	子画面窗口切换上一画面.....	80
2.119	子画面窗口切换下一画面.....	81
2.120	使能画布.....	81
2.121	基本绘图指令.....	81
3.	附录A基本指令集详述.....	82
3.1	设置前/背景色.....	82
3.2	切换画面时自动清除图层.....	82
3.3	设置文字行列间距.....	83
3.4	设置文本框.....	83
3.5	设置图片过滤色.....	83
3.6	文本显示.....	83
3.7	光标显示.....	84
3.8	全屏图片显示.....	85
3.9	区域图片显示.....	85
3.10	图片剪切.....	85
3.11	动画显示.....	86
3.12	前景色画点.....	86
3.13	背景色画点（删除点）.....	87
3.14	画线.....	87
3.15	将等间隔X坐标用前景色连接.....	87
3.16	按照坐标偏移量用前景色连线.....	88
3.17	将指定的坐标点用前景色连接.....	88
3.18	将指定的坐标点用背景色连接.....	88
3.19	按照坐标偏移量用背景色连线.....	88
3.20	画空心圆.....	89
3.21	画实心圆.....	89
3.22	画圆弧.....	89
3.23	画空心矩形.....	90
3.24	画实心矩形/局部清屏.....	90
3.25	画空心椭圆.....	90
3.26	画实心椭圆.....	90
3.27	清除图层.....	91
3.28	截取当前屏幕并保存在FLASH中.....	91
3.29	显示保存在FLASH中的截取画面.....	91
3.30	RTC模式设置.....	91
4.	声明与服务.....	93

1. 指令格式和指令集

1.1 指令格式

一条完整的串口指令帧格式如表 1.1 所示。如果指令参数大于 1 个字节，则高字节在前、低字节在后。指令的最大长度为 1024 字节(包含帧头和帧尾)，数值均为十六进制。串口格式：8 位数据位、1 位停止位、无效验位。

表 1.1 无 CRC 校验指令帧格式

指令	EE	XX	XX XX…XXX	FF FC FF FF
说明	帧头	指令	指令参数	帧尾

若用户需要指令支持 CRC 格式效验，则指令帧格如表 1.2 所示。CRC 的算法程序请在官网资料下载栏中下载。CRC 校验不包含帧头和帧尾。

表 1.2 带 CRC 校验的指令帧格式

指令	EE	XX	XX XX…XXX	CRC16	FF FC FF FF
说明	帧头	指令	指令参数	校验位	帧尾

1.1.1 指令操作举例说明

1. 切换画面。若用户需要在屏幕上显示如图 1-1 所示的画面，则用户主机发送的串口指令如下：



图 1-1 切换画面显示

单片机发送命令	EE 【B1 00 00 01】 FF FC FF FF
命令解析	EE 表示帧头
	B1 00 表示切换画面指令
	00 01 表示需要显示的目标画面 ID，2 个字节
	FF FC FF FF 表示帧尾
提示	每个画面的 ID 编号均由上位机编译后生成

按钮控件 ID 上传。若用户点击触摸图 1-2 中的“启动运行”按钮，假设当前画面 ID 为 2，按钮 ID 为 1，则串口屏上传串口指令如下：



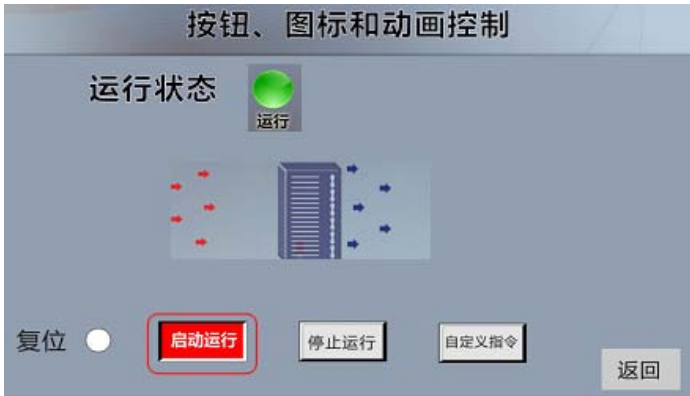


图 1-2 按钮控件 ID 值上传

屏幕上传指令	EE 【B1 11 00 02 00 01 10 01 01】 FF FC FF FF
命令解析	EE 表示帧头
	B1 11 表示上传的组态控件指令；
	00 02 00 01 表示当前画面 ID 为 2，控件 ID 为 1
	10 表示控件为按钮控件
	01 表示按钮控件属性为开关类型
	01 表示按钮状态由弹起变成按下
	FF FC FF FF 表示帧尾
提示	所有按钮控件的 ID 号均由上位机配置生成，也可自行修改
说明	用户单片机串口接收到以上指令数据就可以解析出当前哪个画面的哪个按钮被按下

文本显示。若用户需要在图 1-3 中当前电压显示数字 220，假设画面 ID 为 1，文本控件 ID 为 7，则用户主机发送指令如下：



图 1-3 文本控件显示

单片机发送指令	EE 【B1 10 00 01 00 07 32 32 30】 FF FC FF FF
命令解析	EE 表示帧头
	B1 10 表示发送的组态控件指令；
	00 01 00 07 表示画面 ID 为 1，控件 ID 为 7
	32 32 30 表示数字 220 的 ASCII 码；
	FF FC FF FF 表示帧尾
说明	由于字体的颜色、大小、位置都在 PC 上预先进行了配置，所以用户单片机只需针对 ID 号发送数据即可

系统键盘输入。若用户需要在图 1-4 中的机组号显示“1001 号”，假设画面 ID 为 1，文本控件为 1，则只需点击文本框处，然后在弹出的系统键盘中录入输入“1001 号”（需进行中英文的切换），最后点击确定，这样录入的文本将自动嵌入到文本框中显示，同时上传指令如下：



图 1-4 键盘录入参数显示

屏幕上传指令	EE 【B1 11 00 01 00 01 11 31 30 30 31 BA C5 00】 FF FC FF FF
命令解析	EE 表示帧头
	B1 11 表示接收的组态控件指令；
	00 01 00 01 表示画面 ID 为 1，控件 ID 为 1
	11 表示控件为文本控件
	31 30 30 31 表示数字 1001 的 ASCII 码
	BA C5 表示汉字“号”的内码
	00 表示字符结束
	FF FC FF FF 表示帧尾
说明	用户单片机接收到屏幕上传的 ASCII 码即可获取输入信息

自定义指令输出。用户可以设置按下某个按钮后，设备上传自己定义的数据串列。例图 1-5 所示，用户可以设置按下“自定义指令”按钮后，屏幕下发指令：FF 01 AA FF。

注意：自定义数据不能包含“FF FC FF FF”组合字符，否则会常规指令帧尾一致，导致执行异常。若需同时多条指令输出，每个指令之间需要分号隔开。

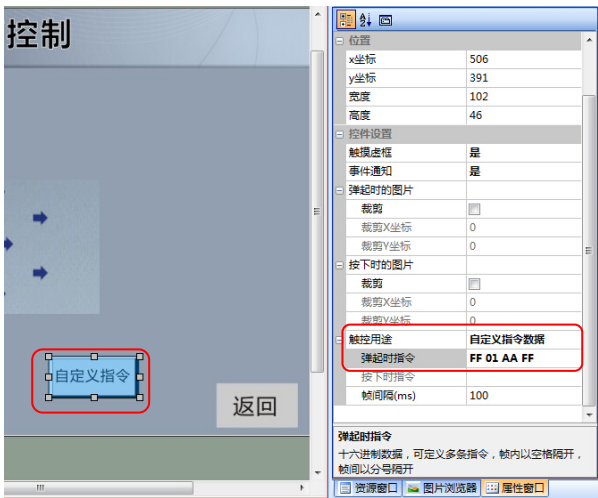


图 1-5 自定义指令输出

图标显示。若用户想实现某个状态图标变化显示，例如在运行时候，显示“运行图标”，停止时显示“停止图标”，复位时图标消失，异常时两个图标来回闪烁，此时可以使用图标控件来解决。

先用软件内置的图标生成器将所有状态图标生成一个 ICON 文件(ICON 文件包含了所有状态帧图)，然后单片机发送指令控制 ICON 的帧播放。例如图 1-6 所示，用户要将“停止图标”更换为“启动图标”，则发送指令如下：

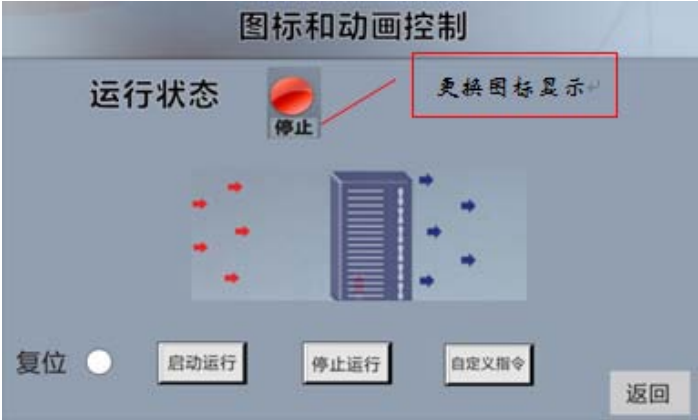


图 1-6 图标显示

单片机发送指令	EE 【 B1 23 00 03 00 01 01】 FF FC FF FF
命令解析	EE 表示帧头
	B1 23 表示图标控件指令；
	00 03 00 01 表示画面 ID 为 3，控件 ID 为 1
	01 代表“运行图标”的帧 ID
	FF FC FF FF 表示帧尾
说明	ICON 文件里面有 2 个图片帧，停止运行是帧 ID 0，启动运行是帧 ID 1，这样播放帧 1 就是显示运行图标，播放帧 0 就是显示停止图标。

仪表显示。若用户需要将图 1-7 中表盘 1 的指针转到度 5 处，假设画面 ID 为 4，仪表控件 ID 为 1，则用户单片机发送指令如下：

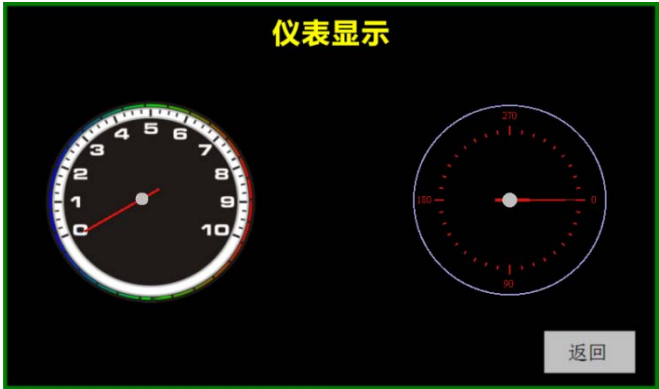


图 1-7 仪表控件显示

设备上传指令	EE【B1 10 00 04 00 01 00 00 00 32】FF FC FF FF
命令解析	EE 表示帧头
	B1 10 表示发送的组态控件指令
	00 04 00 01 表示画面 ID 为 4，控件 ID 为 1
	00 00 00 32 表示数值 50
	FF FC FF FF 表示帧尾
说明	由于 PC 预先设置了表盘的起始值 0，终止值 100，发送数值 50，指针正好指向刻度 5 方向处

曲线显示。若用户需要实现图 1-8 中曲线显示，用户主机只需发送 AD 采样序列值，设备将会自动进行缩放、平移推进显示。

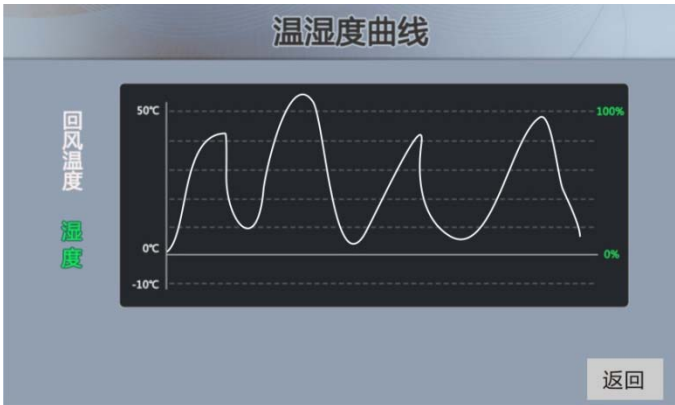


图 1-8 曲线控件显示

进度条和滑块显示。若需要实现图 1-9 所示的进度条滚动显示，用户主机只需发送显示的进度值即可。假设 PC 上配置的进度条起始值为 0，终止值为 100，画面 ID 为 3，进度条控件 ID 为 1，则用户需要将进度条显示在中央，发送指令如下：



图 1-9 进度和滑块显示

设备上传指令	EE【B1 10 00 03 00 01 00 00 00 32】FF FC FF FF
命令解析	EE 表示帧头
	B1 10 表示发送的组态控件指令
	00 03 00 01 表示画面 ID 为 3，控件 ID 为 1
	00 00 00 32 表示数值 50
	FF FC FF FF 表示帧尾
说明	由于 PC 预先设置了进度条的起始值 0，终止值 100，发送数值 50，进度条正好显示在中央位置

下拉菜单显示。点击下拉按钮，自动弹出下拉菜单，下拉菜单的选项值自动加载到文本框内显示，同时屏幕上传下拉菜单的 ID 和选项列数值到用户单片机。用户即可解析当前哪个下拉菜单、哪个选项被选中。如图 1-10 所示，用户点击下拉按钮后，弹出下拉菜单，用户点击 57.7V，数据自动加载到文本框内。

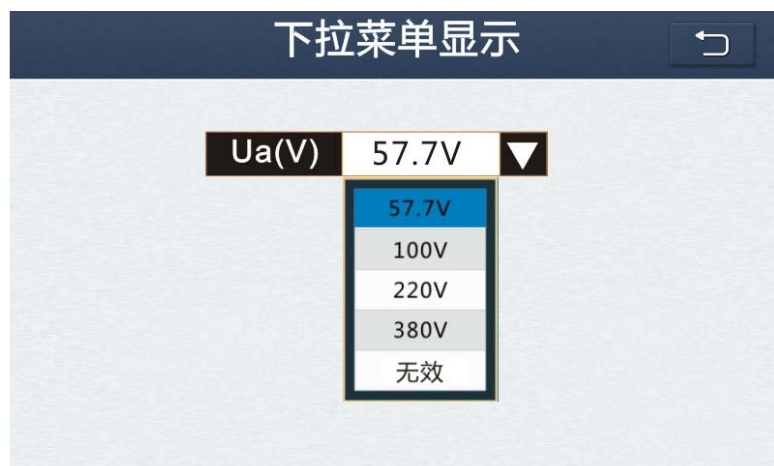


图 1-10 下拉菜单显示

选择菜单显示。用户预先在上位机填入选择项所有数据，设置显示的个数、两端大小的放大倍数，然后滑动参数列，数据就进行滚动循环显示。滚动停止后，屏幕上传当前控件的 ID 和选中的项数值给用户单片机。用户即可解析当前哪个选择菜单、哪个选项被选中，滑动选择控件参考图如图 1-11 所示。



图 1-11 滑动选择控件显示

二维码显示。用户单片机只需要发送扫描内容字符信息，屏幕自动将这些内容生成一个二维码图像，用户手机扫描即可识别。二维码的放大倍数、编码模式、显示颜色均可在上位机进行设置。例如用户串口发送字符“www.gz-dc.com”，屏幕自动生产对应的二维码，如图 1-12 所示。



图 1-12 二维码控件显示

## 1.2 颜色格式

设备共支持  $2^{16}=65536$  种颜色(简称 65K 色), RGB 为 565 格式, 其高低字节分配如表 1.3 所示。

表 1.3 RGB 颜色分配格式

位数 (Bit)	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
颜色分配	R					G					B					

举例说明: 纯红色 ■ 00H, 纯蓝色 ■ 00

注: 用户可以通过 visualTFT 上位机软件调试并获取期望的颜色和 RGB 值。

## 1.3 组态指令集与基本指令集区别

指令集分两部分: 组态控件指令集和基本指令集。

两者主要区别是: 基本指令集可以理解为最底层的指令集, 大部分操作必须包含坐标、颜色、字体等参数信息; 组态指令集直接是面向对象 ID 操作, 而这些对象的相关参数全部预先在上位机软件中进行了配置, 与图片一起下载到了屏的存储器中。

组态指令集满足 99% 的用户需求, 开发简单, 真正的“零”代码编程。除此之外, 在组态指令集无法满足的情况下, 用户还可以将基本指令和组态相结合, 完成所期望显示功能。

## 1.4 如何快速熟悉指令集

串口屏指令集众多, 用户可以利用配套的串口调试板进行屏与上位机联机调试。通过上位机内置的指令助手下发串口命令, 模拟用户单片机主机, 即可快速熟悉指令功能。屏幕与上位机 VisualTFT 联机和指令助手分别如图 1-13、图 1-14 所示。

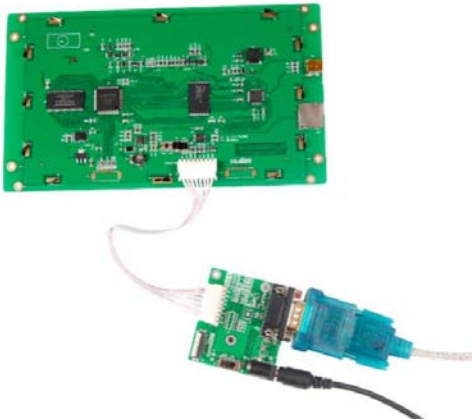


图 1-13 屏幕与上位机联机接线图

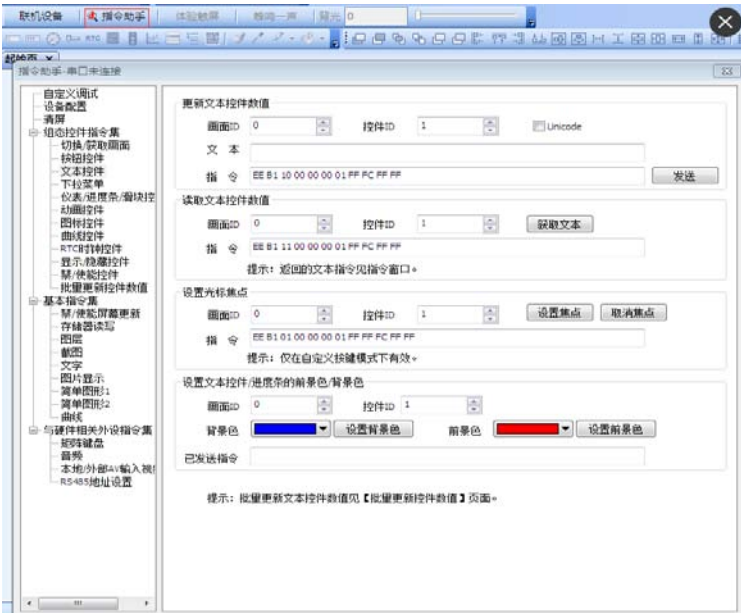


图 1-14 指令助手

1.5 组态指令集表

表 1.4 组态指令集

类别	指令	指令参数	说明
握手指令	0x04	无	握手指令主要用于判断设备是否上电初始化完毕、通信是否正常和是否在线状态等。发送指令后设备返回 55 表示握手成功。 返回指令格式:EE 55 FF FC FF FF
复位报告	0x07	无	一旦设备上电启动、意外重启或监控芯片复位，将立即上传相关数据，告知用户设备已被复位。用户主机检测到设备意外复位后，需控制程序重新从头初始化执行。
复位设备	0x07+0x35 +0x5A+0x5	无	主机在运行的过程中通过串口指令来复位设备。建议主机初始化设备时增加该指令，以便主机意外复位后，设备也跟着复位。



	3+0xA5		返回指令格式: EE 07 FF FC FF FF
获取设备版本	0xFE+0x01	无	<p>该指令主要用于读取屏幕的固件版本, 以 2. 22. 1025. 260 为例, 返回指令格式: EE FE 02 16 00 02 04 01 01 04 FF FC FF FF;</p> <p>参数说明: 02 是指数字十进制的 2;</p> <p>22 是指数字十进制的 16;</p> <p>00 02 是指屏幕的类型, 指商业型;</p> <p>04 01 是指数字十进制的 1025;</p> <p>01 04 是指数字十进制的 260;</p>
背光调节	0x60	Light_level	<p>设置背光亮度值</p> <p>0x00: 背光最亮 0xFF: 背光关闭</p>
自动屏保模式	0x77	Enable+BL_ON+BL_OFF +BL_ON_Time	<p>设置屏保模式的背光亮度值和时间值。一段时间无触摸动作, 屏幕自动降低亮度, 进入节能模式直至触摸被按下时唤醒。</p> <p>Enable(1 个字节): 使能信号</p> <p>0x00: 关闭省电模式 0x01: 开启省电模式</p> <p>BL_ON(1 个字节): 触摸激活后背光的亮度值</p> <p>BL_OFF(1 个字节): 进入省电模式后背光的亮度值</p> <p>BL_ON_Time(2 个字节): 在无触摸动作下, 多长时间后进入省电模式, 单位 1s</p> <p>提示: 只适合带触摸的型号, 不带 TP 产品需要程序控制背光</p>
蜂鸣器控制	0x61	Time	Time(1 个字节): 讯响时间, 单位为 10ms
配置触摸屏	0x70	Cmd	<p>Cmd(1 个字节): 配置参数</p> <p>BIT0: 1 表示触摸屏打开, 0 表示触摸屏关闭;</p> <p>BIT1: 1 表示触摸时蜂鸣器自动响, 0 表示不响;</p> <p>BIT4~BIT2: 触摸上传方式</p> <p>000: 表示按下触摸屏时才上传一次坐标</p> <p>001: 表示触摸屏被按下直至释放后才上传一次坐标</p> <p>010: 触摸屏一直被按下时, 每隔 100ms 定时上传坐标, 释放时也上传一次坐标</p> <p>011: 表示触摸屏被按下和释放时分别上传一次坐标</p> <p>BIT5: 1 表示在 4 秒内连续点击某个区域 20 下, 屏幕进入触摸校准模式, 0 表示禁止此功能;</p> <p>BIT7~BIT6: 保留</p> <p>触摸上传格式:</p> <p>按下时, 上传格式: EE 01 X Y FF FC FF FF</p> <p>释放时, 上传格式: EE 03 X Y FF FC FF FF</p> <p>X 坐标、Y 坐标均为 2 个字节, 高字节在前</p>
触摸屏校准	0x72	无	<p>校准完毕后, 设备返回 EE 04 FF FC FF FF</p> <p>或者在非触控区域某一点 4 秒内快速点击 20 下, 自动进入触摸校准模式, 校准完成后自动返回当前显示画面</p>
触摸屏体验	0x73	Enable	<p>Enable: 使能信号</p> <p>0x00: 关闭体验 0x01: 体验使能</p> <p>用户按下触摸后, 屏对应坐标处显示一个红色实心圆。用于测试触摸屏精确度。</p>



设置波特率	0xA0	Baudset	<p>Baudset(单位 bps,1 个字节), 波特率编序:</p> <p>0x00: 1200      0x01: 2400      0x02: 4800</p> <p>0x03: 9600      0x04: 19200      0x05: 38400</p> <p>0x06: 57600      0x07: 115200      0x08: 1M</p> <p>0x09: 2M      0x0A: 218750      0x0B: 437500</p> <p>0x0C: 875000      0x0D: 921800</p>
切换画面	0xB1+0x00	Screen_id	<p>从当前屏幕切换到目标画面显示</p> <p>Screen_id(2 个字节): 目标画面ID</p> <p>该指令主要实现切换画面显示</p>
使用动画效果切换画面	0xB1+0x05	<p>SCREEN_ID(2Byte)</p> <p>EFFECT(1Byte)</p> <p>AREA_EN(1Byte)</p> <p>AREA_LEFT(2Byte)</p> <p>AREA_RIGHT(2Byte)</p> <p>AREA_TOP(2Byte)</p> <p>AREA_BOTTOM(2Byte)</p>	<p>SCREEN_ID(2Byte): 目标画面 ID</p> <p>EFFECT(1Byte): 切换效果, 0 禁用效果, 1 从左到右 2 从右到左 3 从上到下 4 从下到上</p> <p>AREA_EN(1Byte): 0 全屏效果, 1 区域效果</p> <p>AREA_LEFT(2Byte) 区域左边距离, 像素</p> <p>AREA_RIGHT(2Byte) 区域右边距离</p> <p>AREA_TOP(2Byte) 区域上边距离</p> <p>AREA_BOTTOM(2Byte) 区域下边距离</p> <p>该指令主要实现使用动画效果切换画面</p>
读取画面	0xB1+0x01	无	<p>读取当前画面的 ID 值</p> <p>指令返回格式: EE B1 01 Screen_id FF FC FF FF</p> <p>Screen_id(2 个字节): 当前画面的编号</p> <p>该指令主要用于获取当前屏幕处于哪个画面显示</p>
按钮控件 ID 值上传	0xB1+0x11	Screen_id+Control_id+Control_type+Subtype+Status	<p>按下某个按钮后, 设备主动上传该按钮 ID 信息</p> <p>上传格式: EE 【B1 11 Screen_id Control_id Control_type Subtype Status】 FF FC FF FF</p> <p>Screen_id(2 个字节): 画面编号</p> <p>Control_id(2 个字节): 控件编号</p> <p>Control_type(1 个字节): 固定值 0x10, 表示为按钮控件类型</p> <p>Subtype(1 个字节): 按钮控件的子类型</p> <p>0x00: 画面切换, 表示当前按下的是画面切换按钮</p> <p>0x01: 开关类型, 表示当前按下的是开关按钮</p> <p>0x02: 自定义键值, 表示当前按下的是自定义键值按钮</p> <p>0x03: 自定义指令, 表示当前按下的是自定义指令按钮</p> <p>0x04: 弹出菜单, 表示弹出菜单按钮</p> <p>Status (1 个字节): 按钮状态</p> <p>0x00: 按钮由按下变成弹起状态</p> <p>0x01: 按钮从弹起变成按下状态</p> <p>该指令主要告诉用户当前哪个画面上第几个按钮被按下了</p>
自定义指令输出	无	无	<p>通过上位机自定义按下某个按钮控件后下发的指令, 也可多条指令同时输出, 多条指令之间使用分号隔开</p>
设置按钮弹起和按下状态	0xB1+0x10	Screen_id+Control_id+Status	<p>将某个按钮设置为弹起或按下的显示状态</p> <p>Screen_id(2 个字节): 画面编号</p> <p>Control_id(2 个字节): 控件编号</p>

			<p>Status (1 个字节): 按钮状态</p> <p>0x00: 按钮由按下变成弹起</p> <p>0x01: 按钮由弹起变成按下</p> <p>该指令主要用于改变按钮的显示状态</p>
读取按钮控件状态	0xB1+0x11	Screen_id+Control_id	<p>查询某个按钮当前是按下还是弹起状态</p> <p>Screen_id(2 个字节): 画面编号</p> <p>Control_id(2 个字节): 控件编号</p> <p>返回指令格式: EE B1 11 Screen_id Control_id Control_type Subtype Status FF FC FF FF</p> <p>Control_type(1 个字节): 固定值 0x10, 表示为按钮控件类型</p> <p>Subtype(1 个字节): 按钮控件的子类型</p> <p>0x00: 画面切换, 表示当前按下的是画面切换按钮</p> <p>0x01: 开关类型, 表示当前按下的是开关按钮</p> <p>0x02: 自定义键值, 表示当前按下的自定义键值按钮</p> <p>0x03: 自定义指令, 表示当前按下的自定义指令按钮</p> <p>0x04: 弹出菜单, 表示弹出菜单按钮</p> <p>Status (1 个字节): 0x00 弹起状态 0x01 按下状态</p> <p>注: 返回参数的定义同按钮控件值上传指令一致</p>
更新文本控件数值	0xB1+0x10	Screen_id+Control_id+Strings	<p>对指定的文本控件写入文本数据</p> <p>Screen_id(2 个字节): 画面编号</p> <p>Control_id(2 个字节): 控件编号</p> <p>Strings (不定长): 用户写入的文本数值;</p> <p>该指令主要用于实现文本数据显示</p>
清除文本控件内容	0xB1+0x10	Screen_id+Control_id	<p>清除文本控件内容</p> <p>Screen_id(2 个字节): 画面编号</p> <p>Control_id(2 个字节): 控件编号</p> <p>该指令主要用于清除文本控件内容</p>
读取文本控件数值	0xB1+0x11	Screen_id+Control_id	<p>读取某个文本控件当前显示的数值</p> <p>Screen_id(2 个字节): 画面编号</p> <p>Control_id(2 个字节): 控件编号</p> <p>返回指令格式: EE B1 11 Screen_id Control_id Control_type Strings FF FC FF FF</p> <p>返回参数: Control_type(1 个字节): 固定值 0x11, 表示为文本控件类型</p> <p>Strings (不定长): 当前显示的文本值, 文本后面附加 1 个 0x00 作为结束符</p>
设置文本控件闪烁周期	0xB1+0x15	Screen_id+Control_id+Cycle	<p>设置文本控件闪烁速度</p> <p>Screen_id(2 个字节): 画面编号</p> <p>Control_id(2 个字节): 控件编号</p> <p>Cycle (2 个字节): 闪烁周期(10 毫秒单位), 0 表示不闪烁</p>
设置文本控件滚动速度	0xB1+0x16	Screen_id+Control_id+Speed	<p>设置文本控件滚动速度 (方向朝左)</p> <p>Screen_id(2 个字节): 画面编号</p> <p>Control_id(2 个字节): 控件编号</p> <p>Speed (2 个字节): 文本滚动速度(每秒移动像素), 0 表示不</p>

			滚动
设置文本控件背景色	0xB1+0x18	Screen_id+Control_id+BK_Color	<p>设置文本控件背景色</p> <p>Screen_id(2 个字节): 画面编号</p> <p>Control_id(2 个字节): 控件编号</p> <p>BK_Color (2 个字节): 背景色 RGB 颜色值</p>
设置文本控件前景色	0xB1+0x19	Screen_id+Control_id+FORE_Color	<p>设置文本控件前景色</p> <p>Screen_id(2 个字节): 画面编号</p> <p>Control_id(2 个字节): 控件编号</p> <p>FORE_Color (2 个字节): 前景色 RGB 颜色值</p>
设置文本背景透明	0xB1+0x17	Screen_id+Control_id	<p>将带背景的文本变成背景透明显示</p> <p>Screen_id(2 个字节): 画面编号</p> <p>Control_id(2 个字节): 控件编号</p>
格式化文本显示	0xB1+0x07	Screen_id+Control_id+Sign+Fill_zero+Value	<p>按照相应的条件输入文本显示</p> <p>Screen_id(2 个字节): 画面编号</p> <p>Control_id(2 个字节): 控件编号</p> <p>Sign (1 个字节): 数据类型, 分别为:</p> <p>Unsigned int: 无符号整型, 0x00;</p> <p>Int: 有符号整型, 0x01;</p> <p>Float: 单精度浮点数, 0x02;</p> <p>Double: 双精度浮点数, 0x03;</p> <p>Fill_zero (1 个字节): 小数位数, 若选择不足为补 0 的话, 需要加 0x80;</p> <p>Value (4 个字节): 添加的数据, 用十六进制数表示;</p>
外部更新图片显示 (物联型不支持, 其他型号支持)	0x39	X+Y+Wight+Height	<p>指定图片显示的坐标和宽度, 然后发送 RGB 数据</p> <p>X(2 个字节): 以点为单位的 X 轴坐标值</p> <p>Y(2 个字节): 以点为单位的 Y 轴坐标值</p> <p>Wight (2 个字节): 设置文本范围宽度</p> <p>Height (2 个字节): 设置文本范围高度</p> <p>备注: 先发图片显示的位置和宽度, 然后接着发送图片的 BMP 数据即可, 图片 RGB 高字节在前。</p>
更新进度条控件数值	0xB1+0x10	Screen_id+Control_id+Progressvalue	<p>对指定的进度条控件写入显示的数据</p> <p>Screen_id(2 个字节): 画面编号</p> <p>Control_id(2 个字节): 控件编号</p> <p>Progressvalue (4 个字节): 用户写入的新进度条值</p> <p>该指令主要用于实现进度条的递增或递减</p>
读取进度条控件数值	0xB1+0x11	Screen_id+Control_id	<p>读取指定进度条控件的当前数值</p> <p>Screen_id(2 个字节): 画面编号</p> <p>Control_id(2 个字节): 控件编号</p> <p>返回指令格式: EE B1 11 Screen_id Control_id Control_type Progressvalue FF FC FF FF</p> <p>返回参数: Control_type (1 个字节): 固定值 0x12, 表示为进度条控件类型</p> <p>Progressvalue (4 个字节): 当前进度条值</p>

滑动条控件上传格式	无	无	<p>当拖动滑动条时，设备不断上传游标数值如下：</p> <p>上传格式：EE 【B1 11 Screen_id Control_id Control_type Slidervalue】 FF FC FF FF</p> <p>Screen_id(2 个字节)：画面编号</p> <p>Control_id(2 个字节)：控件编号</p> <p>Control_type(1 个字节)：固定值 0x13，表示为滑动条控件</p> <p>Slidervalue (4 个字节)：表示当前游标数值</p>
更新滑动条控件数值	0xB1+0x10	Screen_id+Control_id+Slidervalue	<p>对指定的滑动条控件写入显示的数据</p> <p>Screen_id(2 个字节)：画面编号</p> <p>Control_id(2 个字节)：控件编号</p> <p>Slidervalue(4 个字节)：用户写入的新滑动条值</p> <p>该指令主要用于控制滑动条游标的位置</p>
读取滑动条控件数值	0xB1+0x11	Screen_id+Control_id	<p>读取指定滑动条控件的当前游标数值</p> <p>Screen_id(2 个字节)：画面编号</p> <p>Control_id(2 个字节)：控件编号</p> <p>返回指令格式：EE B1 11 Screen_id Control_id Control_type Slidervalue FF FC FF FF</p> <p>返回参数：Control_type（1 个字节）：固定值 0x13，表示为滑动条控件类型</p> <p>Slidervalue（4 个字节）：当前显示的滑动条值</p>
更新仪表控件数值	0xB1+0x10	Screen_id+Control_id+Metervalue	<p>对指定的仪表控件写入显示的数据</p> <p>Screen_id(2 个字节)：画面编号</p> <p>Control_id(2 个字节)：控件编号</p> <p>Metervalue(4 个字节)：用户写入的新仪表值</p> <p>该指令主要实现仪表指针的转动</p>
读取仪表控件数值	0xB1+0x11	Screen_id+Control_id	<p>读取指定仪表控件的当前数值</p> <p>Screen_id(2 个字节)：画面编号</p> <p>Control_id(2 个字节)：控件编号</p> <p>返回指令格式：EE B1 11 Screen_id Control_id Control_type Metervalue FF FC FF FF</p> <p>返回参数：Control_type（1 个字节）：固定值 0x14，表示为仪表控件类型</p> <p>Metervalue（4 个字节）：当前显示的仪表值</p> <p>注：返回参数的定义同更新仪表控件数值一致</p>
设置进度条的背景色	0xB1+0x18	Screen_id+Control_id+Bk_Color	<p>设置进度条的背景色</p> <p>Screen_id(2 个字节)：画面编号</p> <p>Control_id(2 个字节)：控件编号</p> <p>Bk_Color（2 个字节）：背景色 RGB 颜色值</p>
设置进度条的前景色	0xB1+0x19	Screen_id+Control_id+Fore_Color	<p>设置进度条的前景色</p> <p>Screen_id(2 个字节)：画面编号</p> <p>Control_id(2 个字节)：控件编号</p> <p>Fore_Color（2 个字节）：前景色 RGB 颜色值</p>
批量更新控件数值	0xB1+0x12	Screen_id+Control_id0+Len0+ Strings0+	<p>批量更新文本、进度条、仪表、按钮和图标控件数值。</p> <p>批量更新的好处是一条指令可以把当前画面全部内容更新完</p>

		+...Control_idn+ Lenn+ Strings n	<p>毕，提高刷新速度和效率。</p> <p>Screen_id(2 个字节)：画面编号</p> <p>Control_id0 (2 个字节)：第 1 个控件的编号</p> <p>Len0 (2 个字节)：第 1 个控件的字节长度</p> <p>Strings0 (不定长)：第 1 个控件的数值，字节数以 Len0 长度而定</p> <p>Control_idn (2 个字节)：第 n 个控件的编号</p> <p>Lenn (2 个字节)：第 n 个控件的字节长度</p> <p>Strings n (不定长)：第 n 个控件的数值，字节数以 Lenn 长度而定</p>
设置光标焦点	0xB1+0x02	SCREEN_ID + CONTROL_ID + ENABLE	<p>在指定的文本控件上显示光标（仅适用自定义键盘的模式下）</p> <p>Screen_id(2 个字节)：画面编号</p> <p>Control_id(2 个字节)：控件编号</p> <p>Enable(1 个字节)：光标显示使能</p> <p>0x00：关闭显示； 0x01：开启显示</p>
手动禁止/使能 屏幕更新	0xB3	ENABLE	<p>手动禁止/使能屏幕更新</p> <p>Enable(1 个字节)：更新使能</p> <p>0x00：禁止更新； 0x01：使能更新</p> <p>该命令主要解决某一画面中实时动态更新的控件数目过多，导致屏幕更新速度慢的问题。</p> <p>使用方法：用户先发送禁止屏幕更新指令，然后发送整个画面中需要更新的内容，最后再使能屏幕更新。</p>
禁止/使能控件	0xB1+0x04	SCREEN_ID + CONTROL_ID + ENABLE	<p>禁止/使能控件</p> <p>Screen_id(2 个字节)：画面编号</p> <p>Control_id(2 个字节)：控件编号</p> <p>Enable(1 个字节)：控件使能</p> <p>0x00：禁止控件； 0x01：使能控件</p> <p>该命令主要用于某一画面中某些控件的禁止或者使能操作，可以根据用户需求进行某些控件的禁止或者使能操作。</p>
屏蔽/隐藏控件	0xB1+0x03	SCREEN_ID + CONTROL_ID + ENABLE	<p>将某个控件屏蔽或显示隐藏。</p> <p>Screen_id(2 个字节)：画面编号</p> <p>Control_id(2 个字节)：控件编号</p> <p>Enable(1 个字节)：屏蔽或隐藏使能</p> <p>0x00：屏蔽或隐藏控件； 0x01：屏蔽/隐藏解除</p> <p>该指令常用于某一时刻将指定按钮控件功能失效，也可用于将某个控件隐藏无法显示</p>
动画控件显示	0xB1+0x20	SCREEN_ID + CONTROL_ID	<p>启动动画播放</p> <p>Screen_id(2 个字节)：画面编号</p> <p>Control_id(2 个字节)：控件编号</p> <p>启动播放后，动画每次从帧头 0 开始播放，格式仅支持 GIF。</p>
	0xB1+0x21	SCREEN_ID + CONTROL_ID	<p>停止动画播放</p> <p>Screen_id(2 个字节)：画面编号</p> <p>Control_id(2 个字节)：控件编号</p> <p>执行停止播放后，下次将从帧头 0 开始播放</p>

	0xB1+0x22	SCREEN_ID + CONTROL_ID	<p>暂停动画播放</p> <p>Screen_id(2 个字节): 画面编号</p> <p>Control_id(2 个字节): 控件编号</p> <p>执行暂停后, 下次将从暂停帧开始继续播放</p>
	0xB1+0x23	SCREEN_ID + CONTROL_ID + FlashImgae_ID	<p>指定帧播放</p> <p>Screen_id(2 个字节): 画面编号</p> <p>Control_id(2 个字节): 控件编号</p> <p>FlashImgae_ID(1 个字节): 某一动画帧 ID</p> <p>指定从某一帧开始播放</p>
	0xB1+0x24	SCREEN_ID + CONTROL_ID	<p>播放上一帧</p> <p>Screen_id(2 个字节): 画面编号</p> <p>Control_id(2 个字节): 控件编号</p>
	0xB1+0x25	SCREEN_ID + CONTROL_ID	<p>播放下一帧</p> <p>Screen_id(2 个字节): 画面编号</p> <p>Control_id(2 个字节): 控件编号</p>
动画控件值 上传	0xB1+0x26	SCREEN_ID + CONTROL_ID+ Status + FlashImgae_ID	<p>按下或滑动某个动画控件时候, 设备上传的相关信息:</p> <p>Screen_id(2 个字节): 画面编号</p> <p>Control_id(2 个字节): 控件编号</p> <p>Status (1 个字节): 0x00 表示触摸按下, 0x01 表示弹起;</p> <p>FlashImgae_ID (1 个字节): 表示按下屏幕时刻显示的动画帧;</p> <p>备注: 用户可以在 PC 配置禁止/使能动画控件值上传。</p>
图标控件显示	0xB1+0x23	SCREEN_ID + CONTROL_ID + IconImgae_ID	<p>用户先使用上位机自带的图标生成器, 把所有不同状态的图片合成 1 个 ICON 文件, 然后主机指定某一图标帧显示。</p> <p>Screen_id(2 个字节): 画面编号</p> <p>Control_id(2 个字节): 控件编号</p> <p>IconImgae_ID (1 个字节): 某一图标帧 ID</p> <p>该指令常用于在画面同一个位置显示几种不同状态的图片。</p> <p>注: 若用户需要图标消失显示, 可以做一帧透明的 PNG 图片即可达到消失效果。</p>
图标控件值 上传	0xB1+0x26	SCREEN_ID + CONTROL_ID+ Status + IconImgae_ID	<p>按下或滑动某个图标控件时候, 设备上传的相关信息:</p> <p>Screen_id(2 个字节): 画面编号</p> <p>Control_id(2 个字节): 控件编号</p> <p>Status (1 个字节): 0x00 表示触摸按下, 0x01 表示弹起;</p> <p>IconImgae_ID (1 个字节): 表示按下屏幕时刻显示的图标帧;</p> <p>备注: 用户可以在 PC 配置禁止/使能图标控件值上传。</p>
设置图标位置	0xB1+0x28	Screen_id + Control_id + X + Y	<p>设置图标位置</p> <p>参数说明: Screen_id(2 个字节): 画面编号</p> <p>Control_id(2 个字节): 控件编号</p> <p>X (2 个字节): 显示的 X 坐标位置</p> <p>Y (2 个字节): 显示的 Y 坐标位置</p> <p>该指令主要用于动态调整图标控件显示的位置。</p>
曲线控件	0xB1+0x30	SCREEN_ID + CONTROL_ID + CHANNEL+COLOR	<p>添加指定数据通道</p> <p>Screen_id(2 个字节): 画面编号</p> <p>Control_id(2 个字节): 控件编号</p>

			CHANNEL(1 个字节): 共 8 个数据通道, 编号范围 (0~7) COLOR(2 个字节): 数据通道颜色
	0xB1+0x31	SCREEN_ID + CONTROL_ID+CHANNEL	删除指定数据通道 Screen_id(2 个字节): 画面编号 Control_id(2 个字节): 控件编号 CHANNEL(1 个字节): 共 8 个数据通道, 编号范围 (0~7)
	0xB1+0x32	SCREEN_ID + CONTROL_ID+ CHANNEL+DATA_LEN+DATA	指定通道末尾添加新数据 在指定数据通道末尾添加新数据, 当数据长度超过缓冲区长度时, 旧数据左移。 Screen_id(2 字节): 画面编号; Control_id(2 字节): 控件编号 CHANNEL(1 个字节): 共 8 个数据通道, 编号范围 (0~7) DATA_LEN(2 个字节): 数据长度 DATA : 不定长数据, 长度由 DATA_LEN 指定格式: 数据代表是 Y 轴方向的值, X 轴方向的会根据水平缩放系数自动递增, 例如当水平缩放系数为 1 的时候, 每插入一个点 X 轴自动加 1, 当水平缩放系数为 5 的时候, 每插入一个点 X 轴自动加 5
	0xB1+0x33	SCREEN_ID + CONTROL_ID +CHANNEL	清空指定通道的数据 Screen_id(2 字节): 画面编号; Control_id(2 字节): 控件编号 CHANNEL(1 个字节): 共 8 个数据通道, 编号范围 (0~7)
	0xB1+0x34	SCREEN_ID + CONTROL_ID+ XOFFSET + XMUL + YOFFSET+YMUL	指定垂直/水平的缩放/平移 Screen_id(2 个字节): 画面编号 Control_id(2 个字节): 控件编号 XOFFSET (2 个字节): 水平偏移数据点数, 左移为正, 右移为负 XMUL (2 个字节): 水平缩放系数, 单位 0.01 YOFFSET (2 个字节): 垂直偏移数值, 下移为正, 上移为负 YMUL (2 个字节): 垂直缩放系数, 单位 0.01 采样点与坐标点的计算公式: 第 N 个采样点的数值为 V X 坐标 = (N-XOFFSET)*XMUL*0.01 Y 坐标 = (V-YOFFSET)*YMUL*0.01
	0xB1+0x35	SCREEN_ID + CONTROL_ID+ CHANNEL+DATA_LEN+DATA	指定通道前端添加新数据 在指定数据通道最前端插入新数据, 当数据长度超过缓冲区长度时, 旧数据右移 Screen_id(2 字节): 画面编号 Control_id(2 字节): 控件编号 CHANNEL(1 个字节): 共 8 个数据通道, 编号范围 (0~7) DATA_LEN(2 个字节): 数据长度 DATA : 不定长数据, 长度由 DATA_LEN 指定
设置下拉菜单 控件	0xB1+0x13	SCREEN_ID + CONTROL_ID+ Enable+Textctrl_ID	设置弹出下拉菜单后, 菜单选项数据自动写入到哪个文本框中 Screen_id(2 个字节): 画面编号 Control_id(2 个字节): 控件编号 Enable (1 个字节): 菜单使能位, 00 为隐藏, 01 为显示, 点击菜单后, 自动隐藏 Textctrl_ID(2 个字节): 文本控件
下拉菜单控件值 上传	0xB1+0x14	SCREEN_ID + CONTROL_ID+0x1A+	菜单控件上传的数据帧格式。选中下拉菜单数据项后, 系统自动上传控件 ID 值和选中项值。

		Meundata_ID+ Status	<p>Screen_id(2 个字节): 画面编号</p> <p>Control_id(2 个字节): 控件编号</p> <p>Meundata_ID ( 1 个字节 ): 第几个选项数据</p> <p>Status ( 1 个字节 ): 触摸状态, 0 表示弹起; 1 表示按下</p> <p>下拉菜单控件值上传指令在弹起或者按下都会有相应的指令数据上传。</p>
设置滑动选择控件当前选项	0xB1+0x10	SCREEN_ID + CONTROL_ID+ Select_data_ID	<p>设置滑动选择控件当前显示的数据项</p> <p>Screen_id(2 个字节): 画面编号</p> <p>Control_id(2 个字节): 控件编号</p> <p>Select_data_ID ( 1 个字节 ): 第几个选项数据</p>
滑动选择控件值上传	0xB1+0x11	SCREEN_ID + CONTROL_ID+0x1B+ Select_data_ID	<p>滚动控件主要实现上下滑动选中所需数据, 选中后系统自动上传控件 ID 值和当前选中项。</p> <p>Screen_id(2 个字节): 画面编号</p> <p>Control_id(2 个字节): 控件编号</p> <p>Select_data_ID ( 1 个字节 ): 第几个选项数据</p>
更新二维码控件	0xB1+0x10	Screen_id+Control_id+ Strings	<p>生成指定显示内容的二维码图</p> <p>Screen_id(2 个字节): 画面编号</p> <p>Control_id(2 个字节): 控件编号</p> <p>Strings (不定长): 二维码字符内容;</p> <p>该指令主要用于实现二维码图显示, 指令格式与文本控件一样</p>
设置定时器	0xB1+0x40	Screen_id+Control_id+ timedata	<p>设置 RTC 定时器时间</p> <p>Screen_id(2 个字节): 画面编号</p> <p>Control_id(2 个字节): RTC 控件编号</p> <p>timedata (4 个字节): 定时器数值</p> <p>该指令只针对硬件支持 RTC 时钟的产品设备, 并预先上位机设置好时钟控件属性</p>
启动定时器	0xB1+0x41	Screen_id+Control_id	<p>启动定时器</p> <p>Screen_id(2 个字节): 画面编号</p> <p>Control_id(2 个字节): RTC 控件编号</p> <p>定时器时间结束, 屏上传事件通知, 上传格式:</p> <p>EE B1 43 Screen_idControl_id 17 FF FC FF FF</p>
暂停定时器	0xB1+0x44	Screen_id+Control_id	<p>暂停定时器</p> <p>Screen_id(2 个字节): 画面编号</p> <p>Control_id(2 个字节): RTC 控件编号</p> <p>暂停计时后, 发送启动定时器将继续计数</p>
停止定时器	0xB1+0x42	Screen_id+Control_id	<p>停止计时器</p> <p>Screen_id(2 个字节): 画面编号</p> <p>Control_id(2 个字节): RTC 控件编号</p> <p>停止计时器后, 定时数值将恢复原始设置值</p>
读取定时器	0xB1+0x45	Screen_id+Control_id	<p>读取定时器</p> <p>Screen_id(2 个字节): 画面编号</p> <p>Control_id(2 个字节): RTC 控件编号</p> <p>读取定时器后, 屏幕会返回相应的定时值</p> <p>返回格式: EE B1 45 Screen_idControl_id17 Strings FF FC FF</p>



			<p>FF</p> <p>返回参数: Screen_id(2 个字节): 画面编号</p> <p>Control_id(2 个字节): RTC 控件编号</p> <p>17(1 个字节): 0x17, 固定码</p> <p>Strings(4 个字节): 读取的定时值</p>
设置 485 屏地址 (只支持 485 接口的屏幕)	0xA5+0x01	Addr	<p>设置 485 屏地址</p> <p>Addr: 2 个字节, 表示屏地址, 地址从 1 开始, 0 为广播地址</p> <p>地址设置成功后, 所有指令均会添加地址信息, 例如:</p> <p>EE Addr1 Addr0 Screen_idScreen_id... FF FC FF FF</p>
取消 485 屏地址 (只支持 485 接口的屏幕)	0xA5+0x00	无	取消 485 地址信息
写数据到 FLASH	0x87	Addr + Data0... +Datan	<p>将数据保存在指定的 FLASH 地址中, 当做 EEPROM 使用</p> <p>Addr (4 个字节): 数据写入的起始地址</p> <p>Datan (1 个字节): 写入的数据</p> <p>存储空间为 128K 字节, 地址范围是 0~0x1FFFF</p> <p>写入成功后返回: EE 0C FF FC FF FF</p> <p>注意: 由于 Nandflash 有擦写寿命 (10 万次左右), 重要参数、需要反复擦写的核心数据不建议存放</p>
读取保存在 FLASH 中的数据	0x88	Addr + Length	<p>将写入随机或顺序存储器中的数据读出</p> <p>Addr (4 个字节): 数据读起始地址</p> <p>Length(2 个字节): 读取数据的长度, 单位为字节</p> <p>返回的数据格式为:</p> <p>EE 0B Data0 ... Datan FF FC FF FF</p>
截取当前屏幕并保存在 FLASH 中 (基本型、商业型、经济型支持, 物联网型不支持)	0x46	Image_ID	<p>将当前屏幕显示内容保存到设备 FLASH 中</p> <p>Image_ID (1 个字节): 用户自定义保存在存储器中的画面编号</p>
显示保存在 FLASH 中的截取画面 (基本型、商业型、经济型支持, 物联网型不支持)	0x47	Image_ID	<p>显示保存在设备 FLASH 中的截取画面</p> <p>Image_ID (1 个字节): 用户自定义保存在存储器中的画面编号</p>
锁定系统配置	0x08	0xA5+0x5A+0x5F+0xF5	<p>防止在系统运行过程中, 收到错误指令导致系统配置意外修改。一旦配置被锁定, 设备将无法接收外部串口命令进行修改, 直到系统被解除。</p> <p>配置参数包括: 波特率、触摸和矩阵键盘工作模式、自动背光调节参数。</p> <p>返回格式: EE 17 FF FC FF FF</p>
解除系统配置锁定	0x09	0xDE+0xED+0x13+0x31	<p>一旦解除系统配置锁定, 设备可以重新接收外部串口命令来修改配置参数。</p> <p>返回格式 EE 18 FF FC FF FF</p>

告警事件触发	0xB1+0x50	Screen_id+Control_id+Value+ (Sec+Min+Hour+Day+Week+Mon+Year)	<p>告警事件触发</p> <p>Screen_id(2 个字节): 画面编号</p> <p>Control_id(2 个字节): 数据记录控件编号</p> <p>Value(2 个字节): 告警事件 ID 号</p> <p>若客户选中了指令时间的话, 指令会添加以下参数:</p> <p>Sec(1 个字节): 秒, BCD 码表示</p> <p>Min(1 个字节): 分, BCD 码表示</p> <p>Hour(1 个字节): 时, BCD 码表示</p> <p>Day(1 个字节): 日, BCD 码表示</p> <p>Week(1 个字节): 星期, BCD 码表示</p> <p>Mon(1 个字节): 月, BCD 码表示</p> <p>Year(1 个字节): 年, BCD 码表示</p> <p>该指令是通过给屏幕传送相应的事件 ID, 使得相应的事件 ID 号转换成用户设置的告警事件输出</p>
告警事件解除	0xB1+0x51	Screen_id+Control_id+Value+ (Sec+Min+Hour+Day+Week+Mon+Year)	<p>告警事件解除</p> <p>Screen_id(2 个字节): 画面编号</p> <p>Control_id(2 个字节): 数据记录控件编号</p> <p>Value(2 个字节): 告警事件 ID 号</p> <p>若客户选中了指令时间的话, 指令会添加以下参数:</p> <p>Sec(1 个字节): 秒, BCD 码表示</p> <p>Min(1 个字节): 分, BCD 码表示</p> <p>Hour(1 个字节): 时, BCD 码表示</p> <p>Day(1 个字节): 日, BCD 码表示</p> <p>Week(1 个字节): 星期, BCD 码表示</p> <p>Mon(1 个字节): 月, BCD 码表示</p> <p>Year(1 个字节): 年, BCD 码表示</p> <p>该指令是通过给屏幕传送相应的事件 ID, 使得相应的事件 ID 号转换成用户设置的解除告警事件输出</p>
数据记录控件添加常规记录	0xB1+0x52	Screen_id+Control_id+Strings	<p>添加常规记录</p> <p>Screen_id(2 个字节): 画面编号</p> <p>Control_id(2 个字节): 数据记录控件编号</p> <p>Strings (不定长): 用户添加的字符串内容</p>
数据记录控件清除记录数据	0xB1+0x53	Screen_id+Control_id	<p>清除记录数据</p> <p>Screen_id(2 个字节): 画面编号</p> <p>Control_id(2 个字节): 数据记录控件编号</p>
数据记录控件设置记录显示偏移 (即滚动条位置)	0xB1+0x54	Screen_id+Control_id+Offset	<p>设置记录偏移 (即滚动条位置)</p> <p>Screen_id(2 个字节): 画面编号</p> <p>Control_id(2 个字节): 数据记录控件编号</p> <p>Offset (2 个字节): 所在行号 (即滚动条的位置)</p>
数据记录控件获取当前记录数目	0xB1+0x55	Screen_id+Control_id	<p>获取当前数据记录数目</p> <p>Screen_id(2 个字节): 画面编号</p> <p>Control_id(2 个字节): 数据记录控件编号</p> <p>返回格式: EE B1 55 Screen_idControl_id 1D Count FF FC FF FF</p>

数据记录控件读取某一行记录	0xB1+0x56	Screen_id+Control_id+ Position	读取某一行记录 Screen_id(2 个字节): 画面编号 Control_id(2 个字节): 数据记录控件编号 Position (2 个字节): 某一行的行号 返回格式: EE B1 56 Screen_idControl_id 1D Strings FF FC FF FF
数据记录控件修改常规记录	0xB1+0x57	Screen_id+Control_id+ Position+Strings	修改常规记录 Screen_id(2 个字节): 画面编号 Control_id(2 个字节): 数据记录控件编号 Position (2 个字节): 某一行的行号 Strings (不定长): 用户添加的字符串内容
数据记录控件删除常规记录	0xB1+0x58	Screen_id+Control_id+ Position	删除常规记录 Screen_id(2 个字节): 画面编号 Control_id(2 个字节): 数据记录控件编号 Position (2 个字节): 某一行的行号
数据记录控件插入常规记录	0xB1+0x59	Screen_id+Control_id+ Position+Strings	插入常规记录 Screen_id(2 个字节): 画面编号 Control_id(2 个字节): 数据记录控件编号 Position (2 个字节): 某一行的行号 Strings (不定长): 用户添加的字符串内容
数据记录控件选择一条记录	0xB1+0x5A	Screen_id+Control_id+ Offset	选择一条记录 Screen_id(2 个字节): 画面编号 Control_id(2 个字节): 数据记录控件编号 Offset (2 个字节): 所在行号
数据记录控件一次添加多条记录	0xB1+0x5B	Screen_id+Control_id+ Count+Record0_size+ Record0+Record1_size+ Record1+...	一次添加多条记录 Screen_id(2 个字节): 画面编号 Control_id(2 个字节): 数据记录控件编号 Count (2 个字节): 添加的记录条数 Recordx_size(2 个字节): 每一条记录对应的大小 Recordx (不定长): 每一条记录相应的数据
用 SD 卡导出数据记录控件的数据 (CSV 格式)	0xB1+0x5C	Screen_id+Control_id	SD 卡导出数据记录控件的数据 (CSV 格式) Screen_id(2 个字节): 画面编号 Control_id (2 个字节): 数据记录控件编号 返回格式: EE B1 5C Screen_idControl_id00 FF FC FF FF
历史曲线	0xB1+0x60	Screen_id+Control_id+ Value	历史曲线设置采样值 参数说明: Screen_id(2 个字节): 画面编号 Control_id (2 个字节): 数据记录控件编号 Value (不定长): 数据类型有: UINT8、INT8、UINT16、INT16、UINT32、INT32、FLOAT, 每个通道的数据按照设定的数据类型对应, 如果有两个通道, 数据类型是 UINT8, 相应添加两个数据类型为 UINT8 的数据 该指令用于添加历史曲线的采样值数据, 相应的数据添加到指令里面, 指令发给屏幕, 屏幕按照相应的数据值进行绘制曲线。
	0xB1+0x61	Screen_id+Control_id+	历史曲线禁止/使能采样

		Enable	<p>参数说明: Screen_id(2 个字节): 画面编号</p> <p>Control_id(2 个字节): 文本控件编号</p> <p>Channel(1 个字节): 通道 ID</p> <p>Show(1 个字节): 显示禁止/使能位</p> <p>0x00: 隐藏;</p> <p>0x01: 显示;</p> <p>该指令主要用于对历史曲线进行禁止/使能采样。如果用户需要对某一段数据不进行输入的话, 可以使用禁止采样指令实现; 如果用户需要重新赋予采样功能的话, 需要使用使能采样指令实现。</p>
	0xB1+0x62	Screen_id+Control_id+Show	<p>历史曲线隐藏/显示通道</p> <p>参数说明: Screen_id(2 个字节): 画面编号</p> <p>Control_id(2 个字节): 文本控件编号</p> <p>Channel(1 个字节): 通道 ID</p> <p>Show(1 个字节): 显示禁止/使能位</p> <p>0x00: 隐藏;</p> <p>0x01: 显示;</p> <p>该指令主要用于对历史曲线的某些通道进行显示/隐藏的操作。在用户界面中存在多种通道曲线, 用户需要对某个通道进行详细查看、分析的话, 可以使用指令隐藏相应通道的曲线, 在相应情况下, 也可以用指令来进行显示通道曲线。</p>
	0xB1+0x63	Screen_id+Control_id+00+Sample_Count	<p>历史曲线设置时间长度 (即采样点数)</p> <p>参数说明: Screen_id(2 个字节): 画面编号</p> <p>Control_id(2 个字节): 文本控件编号</p> <p>00 (1 个字节): 控制类型 ID 号, 0x00: 设置时间宽度</p> <p>Sample_Count (2 个字节): 屏显示的采样点数</p> <p>该指令主要用于设置历史曲线的时间长度, 即可以设置历史曲线的采样点数。</p>
	0xB1+0x63	Screen_id+Control_id+01	<p>历史曲线缩放到全屏</p> <p>参数说明: Screen_id(2 个字节): 画面编号</p> <p>Control_id(2 个字节): 文本控件编号</p> <p>01 (1 个字节): 控制类型 ID 号, 0x01: 缩放到全屏</p> <p>该指令主要用于设置历史曲线的缩放, 即可以设置历史曲线缩放比例达到全屏。</p>
	0xB1+0x63	Screen_id+Control_id+02+Zoom+Max_Zoom+Min_Zoom	<p>历史曲线设置缩放比例系数</p> <p>参数说明: Screen_id(2 个字节): 画面编号</p> <p>Control_id(2 个字节): 文本控件编号</p> <p>02 (1 个字节): 控制类型 ID 号, 0x02: 缩放比例系数</p> <p>Zoom (2 个字节): 缩放百分比</p> <p>Max_Zoom (2 个字节): 缩放限制, 一屏最多显示采样点数</p> <p>Min_Zoom (2 个字节): 缩放限制, 一屏最少显示采样点数</p> <p>该指令主要用于设置历史曲线按照用户设置的缩放比例系数进行显示, 使得绘制的曲线能够符合用户查看的角度。</p>
	0xB1+0x64	Screen_id+Control_id+	历史曲线设置数值显示范围

		Max+Min	<p>参数说明: Screen_id(2 个字节): 画面编号</p> <p>Control_id(2 个字节): 文本控件编号</p> <p>Max (4 个字节): 最大值</p> <p>Min (4 个字节): 最小值</p> <p>该指令主要用于设置历史曲线的最大值和最小值, 即设置历史曲线的显示范围。</p>
设置旋转控件的 旋转角度	0xB1+0x10	Value	<p>旋转控件设置旋转角度</p> <p>参数说明: Screen_id(2 个字节): 画面编号</p> <p>Control_id(2 个字节): 文本控件编号</p> <p>Value(2 个字节): 旋转角度, 旋转角度为 0~360 度; 同时旋转控件支持小角度旋转, 粒度 0.1 度, Value=实际角度*10+0x8000, 例如 5.3 度设置值 = 5.3*10+0x8000</p> <p>该指令主要用于设置旋转控件的旋转角度, 用户可用该指令实现指针按照旋转中心旋转设定的角度进行旋转。</p>
多语言切换指令	C1	Ui_lang + check	<p>多语言切换指令</p> <p>参数说明: Ui_lang(1 个字节):</p> <p>BIT3~BIT0: 标识语言索引, 范围: 0-9</p> <p>BIT6~BIT4: 保留, 默认 0</p> <p>BIT7: 标识系统键盘语言, 0: 中文, 1: 英文</p> <p>check (1 个字节): 用于校验, check = 0xC1 + Ui_lang, 取低字节。</p> <p>该指令一般用于对于多语言进行切换的指令, 用户可以用改指令来设置键盘的输出语言是中文还是英文。</p>
获取系统语言索引	C2	无	<p>指令格式: EE【C2】FF FC FF FF</p> <p>参数说明: 无</p> <p>该指令用于获取串口屏当前使用的语言索引;</p> <p>返回指令: EE【C2 Lang】FF FC FF FF</p> <p>返回参数: Lang (1 个字节):</p> <p>BIT3~BIT0: 标识语言索引, 范围: 0-9</p> <p>BIT6~BIT4: 保留, 默认 0</p> <p>BIT7: 标识系统键盘语言, 0: 中文, 1: 英文</p>
文本控件数字变量调节	0xB1+0x1A	Screen_id+Control_id+Option+Delta+Min_Limit+Max_Limit	<p>数字变量调节</p> <p>Screen_id(2 个字节): 画面编号</p> <p>Control_id (2 个字节): 文本控件编号</p> <p>Option (1 个字节): bit0-0 减少, 1 增加;</p> <p>bit1-0 不循环调节, 1 循环调节</p> <p>Delta (2 个字节): 增量</p> <p>Min_Limit (2 个字节): 最小值</p> <p>Max_Limit (2 个字节): 最大值</p>
RTC 时间设置 (需硬件支持)	0x81	Sec+Min+Hour+Day+Week+Mon+Year	<p>时间参数设定</p> <p>Sec: 秒设置      Min: 分设置;</p> <p>Hour: 小时设置      Day: 日设置</p>

			<p>Week: 星期设置      Mon: 月设置</p> <p>Year: 年设置</p> <p>建议用户直接使用时钟控件来完成 RTC 时间显示和校准。</p> <p>备注: 各 1 个字节, 以 BCD 码表示, 星期天设置为 0x00</p>
读取 RTC 时钟 (需硬件支持)	0x82	无	<p>数据返回格式: EE +0xF7 +Year +Mon +Week +Day +Hour +Min +Sec +FF FC FF FF</p> <p>备注: 各 1 个字节, 以 BCD 码表示</p> <p>建议用户直接使用时钟控件来完成 RTC 的显示和时间的校准。</p>
将当前画面存储到 TF 卡 (基本型、商业型、经济型支持, 物联网型不支持)	0x34	00	<p>将当前画面存储在 TF 卡</p> <p>相应的指令格式为: EE 34 00 FF FC FF FF</p> <p>发送相应指令后, 在 TF 卡中会自动生成相应的.bmp 文件。</p> <p>如果存储成功, 返回格式为: EE 10 FF FC FF FF</p> <p>如果存储失败, 返回格式为: EE 11 FF FC FF FF</p>
获取无线网络设置 (仅物联网型支持)	0xD0+0xA 1	无	<p>获取无线网络设置</p> <p>相应的指令格式为:EE D0 A1 FF FC FF FF</p> <p>返回格式: EE D0 A1 WIFI_GET_CFG FF FC FF FF</p>
获取网络状态 (仅物联网型支持)	0xD0+0xA 2	无	<p>获取网络状态</p> <p>相应的指令格式为:EE D0 A2 FF FC FF FF</p> <p>返回格式: EE D0 A2 WIFI_GET_STATE FF FC FF FF</p>
获取网络设置 (仅物联网型支持)	0xD0+0xA 5	无	<p>获取网络设置</p> <p>相应的指令格式为:EE D0 A5 FF FC FF FF</p> <p>返回格式: EE D0 A5 NETWORK_GET_CFG FF FC FF FF</p>
获取网络服务 (仅物联网型支持)	0xD0+0xA 7	无	<p>获取网络设置</p> <p>相应的指令格式为:EE D0 A7 FF FC FF FF</p> <p>返回格式: EE D0 A7 NETWORK_SERVICE_GET_CFG FF FC FF FF</p>
搜索无线网络 (仅物联网型支持)	0xD0+0xA 3	无	<p>搜索无线网络</p> <p>相应的指令格式为:EE D0 A3 FF FC FF FF</p> <p>返回格式: EE D0 A3 WIFI_SCAN_APLIST FF FC FF FF</p>
保存网络设置 (仅物联网型支持)	0xD0+0xA 4	NETWORK_SET_CFG	<p>保存网络设置</p> <p>相应的指令格式为: EE D0 A4 NETWORK_SET_CFG FF FC FF FF</p>
保存网络服务 (仅物联网型支持)	0xD0+0xA 6	NETWORK_SERVICE_SET	<p>保存网络服务</p> <p>相应的指令格式为: EE D0 A6 NETWORK_SERVICE_SET FF FC FF FF</p>
保存无线网络设置 (仅物联网型支持)	0xD0+0xA 0	WIFI_SET_CFG	<p>保存无线网络设置</p> <p>相应的指令格式为: EE D0 A0 WIFI_SET_CFG FF FC FF FF</p>
发送网络数据 (仅物联网型支持)	0xD0+0xA C	Count+Strings	<p>发送网络数据</p> <p>相应的指令格式为: EE D0 AC Count Strings FF FC FF FF</p> <p>参数格式: Count(2 个字节): 发送的字节数</p> <p>Strings (不定长): 用户选择发送的数据, 用软件定义的格式写入, 例如: 12, 写入为 31 32</p>

发送网络数据（十六进制，仅物联网型支持）	0xD0+0xA C	Count+Strings	<p>发送网络数据</p> <p>相应的指令格式为：EE D0 AC Count Strings（十六进制） FF FC FF FF</p> <p>参数格式：Count(2 个字节)：发送的字节数</p> <p>Strings（不定长）：用户选择发送的数据，用相应的格式写入，例如：12 13，写入为 12 13</p>
保存网络设置（仅物联网型支持）	0xD0+0xA A	无	<p>保存网络设置</p> <p>相应的指令格式为：EE D0 AA FF FC FF FF</p>
音频文件的播放	0x94	音频文件路径	<p>音频文件播放</p> <p>参数说明：音频文件路径（不定长）：用户设置的音频文件路径；该指令主要用于播放用户设置的音频文件，屏幕根据相应的文件路径播放音频文件。</p>
音频文件停止	0x95	无	<p>音频文件停止</p> <p>该指令主要用于使音频文件由播放状态变成停止</p>
音量调节	0x93	Value	<p>音量调节</p> <p>参数说明：Value（1 个字节）：音量值，范围为 0-100.</p> <p>该指令主要用于调节喇叭输出的音量大小。</p>
本地/外部 AV 输入视频	0x4B	Enable	<p>本地/外部 AV 输入视频</p> <p>参数说明：Enable（1 个字节）：本地/外部画面切换标志位</p> <p>0x00：切换到本地图片显示</p> <p>0x01：切换到 AV 视频输入显示</p> <p>该指令主要用于本地图片显示画面和 AV 视频输入显示画面的切换。</p>
显示/隐藏系统键盘	0x86	Show+x+y+type+Option+max_len	<p>显示/隐藏系统键盘</p> <p>参数说明：Show（1 个字节）：0 隐藏，1 显示；</p> <p>x（2 个字节）：键盘弹出的 x 坐标；</p> <p>y（2 个字节）：键盘弹出的 y 坐标；</p> <p>type（1 个字节）：0 小键盘，1 全键盘；</p> <p>Option（1 个字节）：0 正常字符，1 密码，2 时间设置；</p> <p>max_len（1 个字节）：键盘录入字符长度限制；</p> <p>该指令主要用于显示/隐藏系统键盘。</p>
触摸屏灵敏度调节指令（仅用于电容屏）	8A 5A A5	XX	<p>触摸屏灵敏度调节指令（仅用于电容屏）</p> <p>参数说明：XX（1 个字节）：XX 取值范围是 0~7，越高档位越灵敏，可以支持更厚的盖板</p> <p>该指令用于调节触摸屏的灵敏度，适用于电容屏。</p>
播放本地视频	设置本地视频路径	0xB1+0x74	<p>Screen_id+ Control_id+ Videos_Path</p> <p>设置本地视频路径指令：</p> <p>指令格式：EE【B1 74 Screen_id Control_id Videos_Path】 FF FC FF FF</p> <p>参数说明：Screen_id(2 个字节)：画面编号</p> <p>Control_id(2 个字节)：视频播放控件编号</p> <p>Videos_Path：本地视频路径（可以在指令助手中转换）</p>

	播放视频	0xB1+0x70	Screen_id+ Control_id+ Number	<p>播放指令：</p> <p>指令格式：EE【B1 70 Screen_id Control_id Number】FF FC FF FF</p> <p>参数说明：Screen_id(2 个字节)：画面编号</p> <p>Control_id(2 个字节)：视频播放控件编号</p> <p>Number (2 个字节)：视频循环播放的次数，0 为循环播放</p>
播放 U 盘中视频	设置 U 盘视频路径	0xB1+0x74	Screen_id+ Control_id+ Videos_Path	<p>设置 U 盘视频路径指令：</p> <p>指令格式：EE【B1 74 Screen_id Control_id Videos_Path】FF FC FF FF</p> <p>参数说明：Screen_id(2 个字节)：画面编号</p> <p>Control_id(2 个字节)：视频播放控件编号</p> <p>Videos_Path: U 盘中视频路径（可以在指令助手中转换）</p>
	播放视频	0xB1+0x70	Screen_id+ Control_id+ Number	<p>播放指令：</p> <p>指令格式：EE【B1 70 Screen_id Control_id Number】FF FC FF FF</p> <p>参数说明：Screen_id(2 个字节)：画面编号</p> <p>Control_id(2 个字节)：视频播放控件编号</p> <p>Number (2 个字节)：视频循环播放的次数，0 为循环播放</p>
播放 SD 卡中视频	设置 SD 卡视频路径	0xB1+0x74	Screen_id+ Control_id+ Videos_Path	<p>设置 SD 卡视频指令：</p> <p>指令格式：EE【B1 74 Screen_id Control_id Videos_Path】FF FC FF FF</p> <p>参数说明：Screen_id(2 个字节)：画面编号</p> <p>Control_id(2 个字节)：视频播放控件编号</p> <p>Videos_Path: SD 卡中视频路径（可以在指令助手中转换）</p>
	播放视频	0xB1+0x70	Screen_id+ Control_id+ Number	<p>播放指令：</p> <p>指令格式：EE【B1 70 Screen_id Control_id Number】FF FC FF FF</p> <p>参数说明：Screen_id(2 个字节)：画面编号</p> <p>Control_id(2 个字节)：视频播放控件编号</p> <p>Number (2 个字节)：视频循环播放的次数，0 为循环播放</p>
暂停/恢复播放视频		0xB1+0x72	Screen_id+ Control_id	<p>暂停/恢复播放指令：</p> <p>指令格式：EE【B1 72 screen_id control_id】FF FC FF FF</p> <p>参数说明：screen_id(2 个字节)：画面编号</p> <p>control_id(2 个字节)：视频播放控件编号</p>
停止播放视频		0xB1+0x71	Screen_id+ Control_id	<p>停止指令：</p> <p>指令格式：EE【B1 71 screen_id control_id】FF FC FF FF</p> <p>参数说明：screen_id(2 个字节)：画面编号</p> <p>control_id(2 个字节)：视频播放控件编号</p>
音频 ID 播放		0x90+0x01	音频 ID+播放次数	<p>音频播放指令：</p> <p>指令格式：EE【90 01 sounds_id time】FF FC FF FF</p> <p>参数说明：sounds_id(2 个字节)：音频 ID</p> <p>time (1 个字节)：播放次数（0 代表循环播放）</p>



音频暂停播放	0x90+0x02		音频暂停播放指令： 指令格式：EE【90 02】FF FC FF FF
音频恢复播放	0x90+0x03		音频恢复播放指令： 指令格式：EE【90 03】FF FC FF FF
音频停止播放	0x90+0x00		音频停止播放指令： 指令格式：EE【90 00】FF FC FF FF
音频序列播放	0x90+0x04+ 0x00+0x01	音频 ID+播放次数	音频序列播放指令： 指令格式：EE【90 04 00 01 sounds_id time】FF FC FF FF 参数说明：sounds_id(2 个字节)：音频序列 ID (用；分开序列 ID) time (1 个字节)：播放次数 (0 代表循环播放)
锁定系统配置	0x08+0xA5 +0x5A+0x5 F+0xF5	无	锁定系统配置： 该指令用于锁定系统配置 (即掉电保存的设置)，该指令成功配置的话，设备会返回 EE 17 FF FC FF FF。 返回指令格式：EE 17 FF FC FF FF
解除系统配置锁定	0x09+0xD E+0xED+0 x13+0x31	无	解除系统配置锁定： 该指令用于解除系统配置锁定的功能 (即掉电保存的设置)，该指令成功配置的话，设备会返回 EE 18 FF FC FF FF。 返回指令格式：EE 18 FF FC FF FF

## 1.6 基本指令集表

在组态指令集无法满足显示情况下，用户可以通过基本指令集来实现，基本指令集说明见附录 A。用户基本无需关心基本指令集，组态指令集可以满足 99% 的用户需求。基本指令集如下表 1.5 所示。

表 1.5 基本指令集

类别	指令	指令参数	说明
设置前景色	0x41	Fcolor	前景色用于点、线、圆、图形和文字的颜色指定 Fcolor(2 个字节)：RGB 颜色值
设置背景色	0x42	Bcolor	背景色用于清屏、文字底色和曲线背景等颜色的指定 Bcolor(2 个字节)：RGB 颜色值
清屏	0x01	无	按照指定的颜色进行清屏 备注：清屏颜色取决于背景色设置，默认为蓝色
设置文字行列间距	0x43	Y_W+ X_W	Y_W(1 个字节) 以点为单位的行间距，取值 00~3F X_W(1 个字节) 以点为单位的列间距，取值 00~3F
设置文本框	0x45	Enable+Width+Hight	限制文本显示区域，以便自动换行显示 Enable(1 个字节)

			<p>0x01: 打开文本框限制使能, 0x00: 关闭文本框限制使能</p> <p>Width (2 个字节): 文本显示框的宽度</p> <p>Hight (2 个字节): 文本显示框的高度</p>
设置图片过滤色	0x44	FilterColor	<p>图片中的颜色与过滤色值相同时不予显示</p> <p>FillColor (2 个字节): 过滤色 RGB 值</p>
文本显示	0x20	X+Y+Back+Font+String	<p>任意坐标处显示指定大小的文本内容</p> <p>X (2 个字节): 以点为单位的 X 轴坐标值</p> <p>Y (2 个字节): 以点为单位的 Y 轴坐标值</p> <p>Back (背景色, 1 个字节)</p> <p>0x01: 打开背景色显示 0x00: 关闭背景色显示</p> <p>Font (字体编码, 1 个字节)</p> <p>0x00: 8x12 点阵 (ASCII)</p> <p>0x01: 8x16 点阵 (ASCII)</p> <p>0x02: 12x24 点阵 (ASCII)</p> <p>0x03: 16x32 点阵 (ASCII)</p> <p>0x04 12 x 12 点阵 (GBK)</p> <p>0x05: 16 x 16 点阵 (GBK)</p> <p>0x06: 24 x 24 点阵 (GBK)</p> <p>0x07: 32 x 32 点阵 (GB2312)</p> <p>0x08: 32 x 64 点阵 (ASCII)</p> <p>0x09: 64 x 64 点阵 (GB2312)</p> <p>Strings: 用户写入的字符串 (高字节在前)</p> <p>备注: 文字字体颜色与前景色一致, 底色为背景色</p>
光标显示	0x21	Enable+X+Y+Width+Hight	<p>任意坐标处显示指定大小的光标</p> <p>Enable(1 个字节): 光标使能信号</p> <p>0x00: 关闭 0x01: 开启</p> <p>X(2 个字节): 以点为单位的 X 轴坐标值</p> <p>Y(2 个字节): 以点为单位的 Y 轴坐标值</p> <p>Width (1 个字节): 光标宽度</p> <p>Hight (1 个字节): 光标高度</p> <p>备注: 光标的颜色与当前光标区域起点颜色相反, 闪烁时间默认 1 秒</p>
全屏图片显示	0x31	Image_ID+MaskEn	<p>全屏显示某张图片, 起始位置固定 (0, 0) 坐标</p> <p>Image_ID (2 个字节): 图片编号</p> <p>MaskEn(1 个字节): 过滤使能</p> <p>0x00: 颜色不过滤; 0x01 执行颜色过滤</p> <p>备注: 被过滤的颜色取决于图片过滤色的设置, 下载的图片分辨率不能超过当前屏幕的分辨率, 否则不能显示。</p>
区域图片显示	0x32	X+Y+Image_ID+MaskEn	<p>任意坐标处显示某张图片</p> <p>X (2 个字节): 以点为单位的 X 轴坐标值</p> <p>Y (2 个字节): 以点为单位的 Y 轴坐标值</p> <p>Image_ID (2 个字节): 图片编号</p> <p>MaskEn(1 个字节): 过滤使能</p> <p>0x00: 颜色不过滤; 0x01 执行颜色过滤</p>

			备注：被过滤的颜色取决于过滤色的设置。
图片剪切显示	0x33	$X+Y+Image\_ID+Image\_X+Image\_Y+Image\_W+Image\_H+MaskEn$	<p>任意坐标处显示从某张图剪切过来的图片</p> <p>X(2 个字节):以点为单位的 X 轴坐标值</p> <p>Y(2 个字节):以点为单位的 Y 轴坐标值</p> <p>Image_ID (2 字节): 要剪切的图片编号</p> <p>Image_X (2 字节): 被剪切的图片起点 X 坐标</p> <p>Image_Y (2 字节): 被剪切的图片起点 Y 坐标</p> <p>Image_W (2 字节): 剪切的宽度</p> <p>Image_H (2 字节): 剪切的高度</p> <p>MaskEn(1 个字节): 过滤使能</p> <p>0x00:颜色不过滤 ;0x01 执行颜色过滤</p> <p>备注：被过滤的颜色取决于过滤色的设置</p>
动画显示	0x80	$X+Y+FlashImage\_ID+Enable+Playnum$	<p>任意坐标处显示 GIF 动画</p> <p>X (2 个字节): 以点为单位的 X 轴坐标值</p> <p>Y (2 个字节): 以点为单位的 Y 轴坐标值</p> <p>FlashImage_ID (2 个字节): 动画编号</p> <p>Enable(1 个字节): 播放使能</p> <p>0x00: 关闭动画播放; 0x01: 开启动画播放</p> <p>Playnum(1 个字节): 播放次数</p> <p>0x00: 重复播放; 0x01~0xFF: 播放指定次数后停止</p> <p>串口屏上传 EE 02 FF FC FF FF 表示动画播放结束</p> <p>备注：动画只支持*gif 格式，一个画面只支持一个动画播放。</p> <p>如果期望一个画面同时播放多个 gif 动画，以及动画暂停、停止、播放上/下帧等功能，请使用组态动画控件，参见如下组态指令表。</p>
前景色画点	0x50	$X+Y$	<p>在屏幕上显示一个点，点的颜色取决于前景色设置</p> <p>X (2 个字节):以点为单位的 X 轴坐标值</p> <p>Y (2 个字节):以点为单位的 Y 轴坐标值</p>
背景色画点 (删除点)	0x58	$X+Y$	<p>在屏幕上显示一个点，点的颜色取决于背景色设置</p> <p>X (2 个字节):以点为单位的 X 轴坐标值</p> <p>Y (2 个字节):以点为单位的 Y 轴坐标值</p> <p>备注：主要配合前景色画点使用，可用于清除前景色画的点</p>
画线	0x51	$X0+Y0+X1+Y1$	<p>将指定的两个坐标点连接起来</p> <p>X<sub>0</sub> (2 个字节):以点为单位的直线X轴起点坐标值</p> <p>Y<sub>0</sub> (2 个字节):以点为单位的直线Y轴起点坐标值</p> <p>X<sub>1</sub> (2 个字节):以点为单位的直线X轴终点坐标值</p> <p>Y<sub>1</sub> (2 个字节):以点为单位的直线Y轴终点坐标值</p> <p>备注：线的颜色值取决于前景色设置</p>
按照等间隔X坐标 用前景色连线	0x59	$X0+Xspace+Y0+...+Yn$	<p>将指定的多个等间隔 X 轴坐标点快速连接起来</p> <p>X (2 个字节): 以点为单位的 X 轴坐标值</p> <p>Xspace(2 个字节): 以点为单位的 X 轴间隔值，相邻前后点固定间距为 Xspace</p> <p>Y (2 个字节): 以点为单位的 Y 轴坐标值</p>

			备注: 由于该指令不需发送X坐标值, 指令发送时间节省一半, 绘制速度提高一倍。线的颜色值取决于前景色设置
按照坐标偏移量 用前景色连线	0x75	$(X,Y)_0 + (X1_0,Y1_0) + \dots + (Xn_0,Yn_0)$	<p>将指定的多个偏移量坐标点用前景色快速连接起来</p> <p>X (2 个字节): 以点为单位的 X 轴坐标值</p> <p>Y (2 个字节): 以点为单位的 Y 轴坐标值</p> <p>X1<sub>0</sub>(1 个字节): 以点为单位的 X 轴偏移量</p> <p>Y1<sub>0</sub>(1 个字节): 以点为单位的 Y 轴偏移量</p> <p>备注: (X,Y)为第1点绝对坐标, 后面的每一个点分别由前一个点的绝对坐标加上当前偏移量组成。偏移量的最高位为符号位, 0 代表正偏移量, 1 代表负偏移, 最大偏移量值为正负 127 个点。</p>
按照坐标偏移量 用背景色连线	0x76	$(X,Y)_0 + (X1_0,Y1_0) + \dots + (Xn_0,Yn_0)$	<p>将指定的多个偏移量坐标点用背景色快速连接起来</p> <p>X (2 个字节): 以点为单位的 X 轴坐标值</p> <p>Y (2 个字节): 以点为单位的 Y 轴坐标值</p> <p>X1<sub>0</sub>(1 个字节): 以点为单位的 X 轴偏移量</p> <p>Y1<sub>0</sub>(1 个字节): 以点为单位的 Y 轴偏移量</p> <p>备注: (X,Y)第1点为绝对坐标, 后面的每一个点分别由前一个点的绝对坐标加上当前偏移量组成。偏移量的最高位为符号位, 0 代表正偏移量, 1 代表负偏移, 最大偏移量值为正负 127 个点。</p>
将指定的坐标点 用前景色连线	0x68	$(X,Y)_0 + (X,Y)_1 \dots + (X,Y)_n$	<p>将指定的多个坐标点用前景色连接起来</p> <p>X (2 个字节): 以点为单位的 X 轴坐标值</p> <p>Y (2 个字节): 以点为单位的 Y 轴坐标值</p> <p>备注: 线的颜色值取决于前景色设置</p>
将指定的坐标点 用背景色连线	0x69	$(X,Y)_0 + (X,Y)_1 \dots + (X,Y)_n$	<p>将指定的多个坐标点用背景色连接起来</p> <p>X (2 个字节): 以点为单位的 X 轴坐标值</p> <p>Y (2 个字节): 以点为单位的 Y 轴坐标值</p> <p>备注: 线的颜色值取决于背景色设置</p>
画空心圆	0x52	$X_0 + Y_0 + R$	<p>任意坐标处画一个半径为 R 的空心圆</p> <p>X<sub>0</sub> (2 个字节): 以点为单位的圆心 X 坐标值</p> <p>Y<sub>0</sub> (2 个字节): 以点为单位的圆心 Y 坐标值</p> <p>R (2 个字节): 空心圆的半径</p> <p>备注: 颜色值取决于前景色设置</p>
画实心圆	0x53	$X_0 + Y_0 + R$	<p>任意坐标处画一个半径为 R 的实心圆</p> <p>X<sub>0</sub> (2 个字节): 以点为单位的圆心 X 坐标值</p> <p>Y<sub>0</sub> (2 个字节): 以点为单位的圆心 Y 坐标值</p> <p>R (2 个字节): 空心圆的半径</p> <p>备注: 颜色值取决于前景色设置</p>
画圆弧	0x67	$X_0 + Y_0 + R + EA + SA$	<p>任意坐标处画一个圆弧</p> <p>X<sub>0</sub> (2 个字节): 以点为单位的圆心 X 坐标值</p> <p>Y<sub>0</sub> (2 个字节): 以点为单位的圆心 Y 坐标值</p> <p>R (2 个字节): 圆的半径</p> <p>EA(2 个字节): 结束角度</p> <p>SA(2 个字节): 起始角度</p>

			备注：钟表3点方向为0度，逆时针计算；颜色值取决于前景色调色板设置
画空心矩形	0x54	$X_0+Y_0+X_1+Y_1$	任意位置画一个空心矩形，也可用于局部清屏使用 $X_0$ (2个字节):以点为单位的空心矩形左上角X坐标值 $Y_0$ (2个字节):以点为单位的空心矩形左上角Y坐标值 $X_1$ (2个字节):以点为单位的空心矩形右下角X坐标值 $Y_1$ (2个字节):以点为单位的空心矩形右下角Y坐标值 备注：颜色值取决于前景色设置
画实心矩形/局部清屏	0x55	$X_0+Y_0+X_1+Y_1$	任意位置画一个实心矩形 $X_0$ (2个字节):以点为单位的实心矩形左上角X坐标值 $Y_0$ (2个字节):以点为单位的实心矩形左上角Y坐标值 $X_1$ (2个字节):以点为单位的实心矩形右下角X坐标值 $Y_1$ (2个字节):以点为单位的实心矩形右下角Y坐标值 备注：颜色值取决于前景色设置
画空心椭圆	0x56	$X_0+Y_0+X_1+Y_1$	任意位置画一个空心椭圆 $X_0$ (2个字节):以点为单位的空心椭圆最左端 X 坐标值 $Y_0$ (2个字节):以点为单位的空心椭圆最上端 Y 坐标值 $X_1$ (2个字节):以点为单位的空心椭圆最右端 X 坐标值 $Y_1$ (2个字节):以点为单位的空心椭圆最下端 Y 坐标值 说明：颜色值取决于前景色设置
画实心椭圆	0x57	$X_0+Y_0+X_1+Y_1$	任意位置画一个实心椭圆 $X_0$ (2个字节):以点为单位的实心椭圆最左端 X 坐标值 $Y_0$ (2个字节):以点为单位的实心椭圆最上端 Y 坐标值 $X_1$ (2个字节):以点为单位的实心椭圆最右端 X 坐标值 $Y_1$ (2个字节):以点为单位的实心椭圆最下端 Y 坐标值 说明：颜色值取决于前景色设置
清除图层	0x05	Layer	清除某个图层内容 Layer(1个字节): 写入的图层（取值范围 0~1）
切换画面时自动清除当前图层	0x06	Enable	设置切换画面时是否需要自动清除当前用户层 Enable (1个字节): 使能信号 0x01: 自动清除图层 0x00: 禁止清除图层 默认切换画面时全部清除用户两个图层里面的全部内容。
RTC 显示设置 (需硬件支持)	0x85	Cmd+DisMode+Font +Color +X+Y	RTC 显示设置 Cmd (1个字节): 参数配置 BIT0: 使能信号 0: RTC 关闭 1: RTC 开启 BIT7-BIT1: 保留 DisMode(1个字节): 显示模式 0x00: 格式 HH:MM:SS 0x01: 格式 20XX-MM-DD HH:MM:SS Font (1个字节): 字体选择 0x00: 8x12 点阵 (ASCII) 0x01: 8x16 点阵 (ASCII) 0x02: 12x24 点阵 (ASCII) 0x03: 16x32 点阵 (ASCII) 0x04 12 x 12 点阵 (GBK) 0x05: 16 x 16 点阵 (GBK)



			<p>0x06: 24 x 24 点阵 (GBK)0x07: 32 x 32 点阵 (GB2312)</p> <p>0x08: 32 x 64 点阵 (ASCII)0x09: 64 x 64 点阵 (GB2312)</p> <p>Color (2 个字节): 显示颜色</p> <p>X(2 个字节): 以点为单位的 X 轴坐标值</p> <p>Y(2 个字节): 以点为单位的 Y 轴坐标值</p> <p>建议用户直接使用时钟控件来完成 RTC 时间显示和校准。</p>
--	--	--	--

## 2. 组态指令集详述

以下章节主要介绍组态指令集功能和用法，内容中涉及到的驱动函数库可在参考程序范例中查看。程序范例可以在网站中进行下载。

### 2.1 握手

指令格式：EE【04】FF FC FF FF

设备返回：EE 55 FF FC FF FF

握手指令主要用于判断设备是否上电初始化完毕、通信是否正常和是否在线状态等。发送指令后设备返回 55 表示握手成功。

### 2.2 复位报告

指令格式：无

设备返回：EE 07 FF FC FF FF

一旦设备上电启动、意外重启或监控芯片复位，将立即上传相关数据，告知用户设备已被复位。用户主机检测到设备意外复位后，需控制程序重新从头初始化执行。

### 2.3 复位设备

指令格式：EE【07 35 5A 53 A5】FF FC FF FF

设备返回：EE 07 FF FC FF FF

主机在运行的过程中通过串口指令来复位设备。建议主机初始化设备时增加该指令，以便主机意外复位后，设备也跟着复位。

### 2.4 获取设备版本

指令格式：EE【FE 01】FF FC FF FF

设备返回：EE FE 02 16 00 02 04 01 01 04 FF FC FF FF

参数说明：设备返回的是屏幕的固件版本号 2.22.1025.260，其中，指令的表示是十六进制的，相应转换如下：02 转换是 2,16 转换是 22,00 02 是指屏幕的类型号，指商业型，04 01 转换是 1025，01 04 转换是 260。其他类型的屏幕在读取相应的固件版本也按照这种方式获取版本号。

### 2.5 背光调节

指令格式：EE【60 Light\_level】FF FC FF FF

参数说明：Light\_level (1 个字节)：背光亮度值

该指令主要用于液晶背光亮度的调节，取值范围 00H~FFH。00H 表示背光最亮，FFH 表示背光最暗，共有 255 级亮度调节。

若屏幕一定时间内无操作动作，建议用户降低背光亮度至 30%左右，以提高背光寿命。

### 2.6 自动屏保模式

指令格式：EE【77 Enable BL\_ON BL\_OFF BL\_ON\_Time】FF FC FF FF

参数说明：Enable(1 个字节)：使能信号

0x00：关闭省电模式 0x01：开启省电模式

0x03：开启省电模式且开启背光通知

BL\_ON(1 个字节)：触摸激活后背光的亮度值

BL\_OFF(1 个字节)：进入屏保模式后背光的亮度值

BL\_ON\_Time(2 个字节): 无触摸动作时, 进入屏保模式的时间(单位: 1 秒)

该指令主要用于设置屏保模式被激活和进入屏保模式的背光亮度值。省电模式不仅可以延长液晶屏的背光寿命, 还可以降低液晶发光管对外散发的热量。

## 2.7 蜂鸣器控制

指令格式: EE 【 61 Time 】 FF FC FF FF

参数说明: Time (1 个字节): 蜂鸣器讯响的时间, 单位 10ms

该指令用于蜂鸣器的控制, 通过设定 Time 参数实现不同频率的讯响。一般触摸讯响时间 Time 设置为 100ms。

## 2.8 配置触摸屏

指令格式: EE 【 70 Cmd 】 FF FC FF FF

参数说明: Cmd(1个字节): 配置参数

BIT0: 1表示触摸屏打开, 0表示触摸屏关闭;

BIT1: 1表示触摸时蜂鸣器自动响, 0表示不响;

BIT4~BIT2: 触摸坐标值上传方式

000: 表示按下触摸屏时才上传1次坐标

001: 表示触摸屏被按下直至释放后上传1次坐标

010: 触摸一直被按下时, 每100ms上传1次坐标, 释放时也上传1次坐标

011: 表示触摸屏被按下和释放时均上传1次坐标

100: 表示关闭触摸坐标上传

BIT5: 0表示在4秒内连续点击某个非触控区域20下, 屏幕进入触摸校准模式, 1表示禁止此功能;

BIT7-BIT6: 保留

触摸坐标值上传格式:

按下时上传格式: EE 01 X Y FF FC FF FF

释放时上传格式: EE 03 X Y FF FC FF FF, X、Y均为2个字节, 高字节在前

该指令包含了触摸使能、开闭蜂鸣器和坐标值上传方式。如下图 2-1 所示, 若触摸上传格式配置为“000”, 用户按下屏幕(50,100)的位置后, 设备上传数据: EE 01【00 32 00 64】FF FC FF FF。用户主机通过判断接收到的坐标(X,Y)是否在有效触摸区域内即可确定当前触摸是否有效。设备自身对触摸压力值进行了多次采样和运算, 用户无需再进行二次运算。

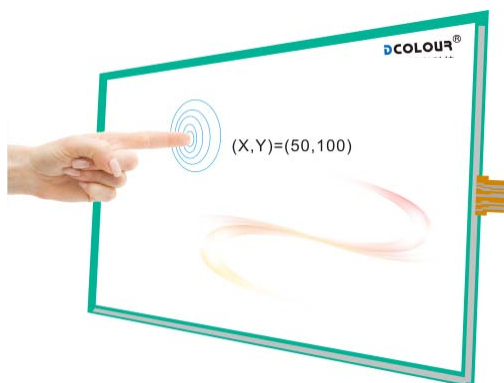


图 2-1 触摸屏工作介绍



## 2.9 触摸屏校准

指令格式：EE【 72 】FF FC FF FF

参数说明：无

该指令用于触摸屏的校准。设备出厂前均进行了校准，用户无需再次校准。发送校准命令后，根据屏幕的提示点击对应的光标，如图 2-2 所示。点击完毕后设备将会提示是否校准成功，否则需要重新校准。用户也可以通过上位机软件发送指令进行校准。

除此之外，用户在非触控区域某一点 4 秒内快速点击 20 下，系统将自动进入触摸校准模式，校准完成后自动返回当前显示画面。该功能比较适合现场进行触摸校准。

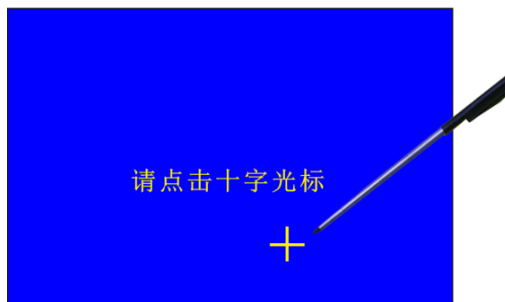


图 2-2 触摸屏校准示意图

## 2.10 触摸屏体验

指令格式：EE【 73 】FF FC FF FF

参数说明：无

该指令属于测试命令。如图 2-3 所示，用户按下触摸后将在对应坐标处显示一个红色的实心圆，方便用户直观地测试触摸屏的好坏及体验触摸值的精准。设备与 PC 连接成功后，用户可点击 VisualTFT 软件工具栏的“体验触摸”来体验触摸的灵敏度和准确度。

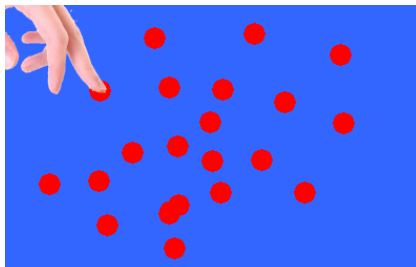


图 2-3 触摸体验效果图

## 2.11 设置波特率

指令格式：EE【A0 Baudset】FF FC FF FF

参数说明：Baudset(1 个字节)：波特率编序，单位 bps

0x00: 1200	0x01: 2400	0x02: 4800
0x03: 9600	0x04: 19200	0x05: 38400
0x06: 57600	0x07: 115200	0x08: 1M
0x09: 2M	0x0A: 2187500	0x0B: 437500
0x0C: 875000	0x0D: 921800	

该指令主要用于波特率的配置,范围为 1200-2Mbps。新的波特率值断电保存。用户可以直接通过上位机 VisualTFT 的“调试助手”配置新的波特率，如图 2-4 所示。



图 2-4 波特率设置

## 2.12 切换画面

指令格式：EE【B1 00 Screen\_id】FF FC FF FF

参数说明：Screen\_id(2 个字节)：画面编号

虽然通过按钮控件可设置按下某个按钮后自动切换到哪个画面,但有些场合也需要主机做出逻辑判断后去控制目标画面显示。

程序参考代码：

```
{SetScreen(2); // 切换到 Screen_id =2 的画面
}
```

## 2.13 使用动画效果切换画面

指令格式：EE【B1 05 SCREEN\_ID(2Byte) EFFECT(1Byte) AREA\_EN(1Byte) AREA\_LEFT(2Byte) AREA\_RIGHT(2Byte) AREA\_TOP(2Byte) AREA\_BOTTOM(2Byte)】FF FC FF FF

参数说明：SCREEN\_ID(2Byte):目标画面 ID

EFFECT(1Byte):切换效果, 0 禁用效果, 1 从左到右 2 从右到左 3 从上到下 4 从下到上

AREA\_EN(1Byte): 0 全屏效果, 1 区域效果

AREA\_LEFT(2Byte) 区域左边距离, 像素

AREA\_RIGHT(2Byte) 区域右边距离

AREA\_TOP(2Byte) 区域上边距离

AREA\_BOTTOM(2Byte) 区域下边距离

## 2.14 读取画面

指令格式：EE【B1 01】FF FC FF FF

参数说明：无

该指令主要用于获取当前画面的 ID 值。在可靠性应用中，主机通过发送该指令来确定画面是否切换成功。

指令返回格式：EE B1 01 Screen\_id FF FC FF FF

其中，Screen\_id(2 个字节)：当前画面的编号

## 2.15 按钮控件 ID 值上传

按钮控件有 5 种用途：切换画面、开关描述、自定义按键、自定义指令和弹出菜单。

(1) 切换画面。切换画面是指按下某个按钮后屏幕自动切换到另外一个画面显示。

例如图 2-5 所示，用户要实现点击“对话框”图标后切换到画面 Screen1 显示，则首先将整个按钮设置为触控区域，然后在属性窗口选择：触控用途→切换画面；目标画面→Screen1，最后运行“虚拟串口屏”进行效果验证。关于上位机详细操作见《大彩组态串口屏快速入门手册》。

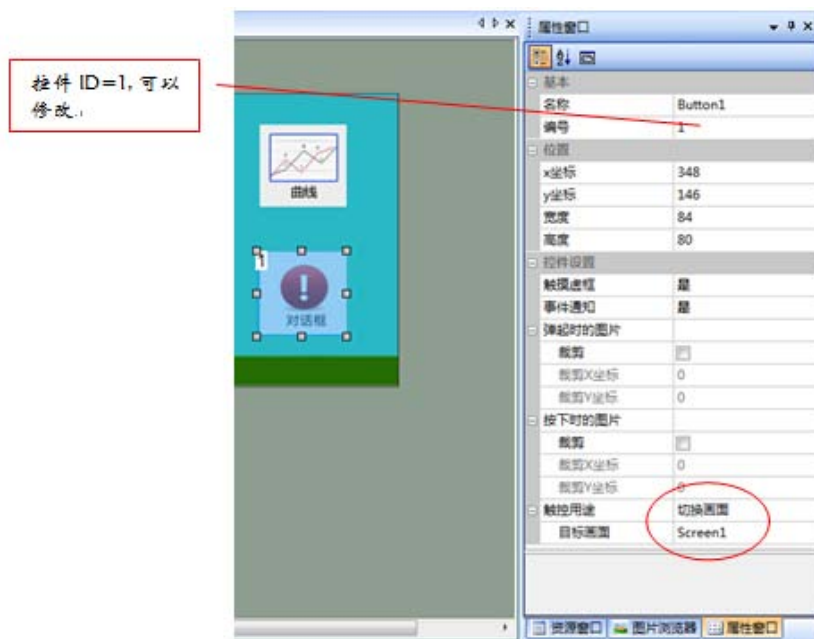


图 2-5 按钮控件-切换画面配置图

画面切换按钮上传的指令格式：

EE 【B1 11 Screen\_id Control\_id Control\_type Subtype Status】 FF FC FF FF

参数说明：Screen\_id(2 个字节)：画面 ID

Control\_id(2 个字节)：控件 ID 编号

Control\_type(1 个字节)：固定值 0x10，表示为按钮控件

Subtype(1 个字节)：固定值 0x00,表示当前按钮功能为切换画面

Status (1 个字节)：保留

开关描述。开关描述是指按钮作为一个按下或弹起的开关功能，共分为 4 种风格，内容如下：

瞬变。按下后，开关自动弹起，类似轻触开关功能；

开关。按下后，开关由弹起变成按下或由按下变成弹起，类似带锁开关功能；

置位。开关只能由弹起变成按下；

复位。开关只能由按下变成弹起；

例如图 2-6 中①②所示，用户需要将“停止运行”按钮作为一个开关功能，则首先将整个按钮设置为触控区域，然后在属性窗口选择：触控用途→开关描述；操作风格→开关；按下时的图片→按钮选中 UI，最后运行“虚拟串口屏”进行效果验证。

运行虚拟串口屏后，可以看到“停止运行”按钮按下去的效果，如所示。



图 2-6 按钮控件-开关类型配置图



图 2-7 运行虚拟串口屏查看按钮按下

开关类型的按钮控件上传格式：

EE 【B1 11 Screen\_id Control\_id Control\_type Subtype Status】 FF FC FF FF

参数说明：Screen\_id(2 个字节)：画面编号

Control\_id(2 个字节)：按钮控件 ID 编号

Control\_type(1 个字节)：固定值 0x10，表示为按钮控件

Subtype(1 个字节)：固定值 0x01,表示当前按钮功能为开关描述

Status (1 个字节)：按钮状态

0x00：按钮由按下变成弹起状态

0x01：按钮从弹起变成按下状态

用户单片机通过解析指令，即可判断当前哪个画面的哪个按钮被按下或弹起。

图中③④是扩展指令，表示按钮被按下的同时，还可以对内/对外输出额外指令，详细介绍可以参考本章节 4.5 和 4.6 所述。

**自定义按键。**自定义按键就是用户预先在背景图上设计好所需的键盘，然后逐个对每个键值进行匹配设置，例如有些按键是数字，有些是删除功能，有些是 Enter 功能等。设置完毕后这个键盘就可以与当前画面的文本控件进行绑定，也就是用户键盘输入的数值可以自动显示在文本框内。

例如图 2-8 所示，用户需要将“数字 9”按钮作为一个键值 9 使用，则首先将整个按钮设置为触控区域，然后在属性窗口选择：触控用途→自定义键值；类型→字符；字符→写入 9，最后运行“虚拟串口屏”进行仿真测试。另外，键值类型除了字符外，还可以选择“Enter”、“Clear”、“Backspace”、“Esc”或“Shift”特殊功能键。



图 2-8 按钮控件-自定义按键配置图

自定义按键按钮上传格式：

EE【B1 11 Screen\_id Control\_id Control\_type Subtype Key\_value Status】FF FC  
FF FF

参数说明：Screen\_id(2 个字节)：画面编号

Control\_id(2 个字节)：控件编号

Control\_type(1 个字节)：固定值 0x10，表示为按钮控件

Subtype(1 个字节)：固定值 0x02,表示当前按钮功能为自定义按键。

Key\_value(1 个字节)：用户自定义的键值，以 ASCII 码表示

Status (1 个字节)：保留

**自定义指令。**用户可以设置按下某个按钮后，设备上传自己定义的数据串列。例图 2-9 所示，用户可以设置按下“自定义指令”按钮后，屏幕下发指令： FF01 AA FF。

**提示：**自定义指令中不能包含 FF FC FF FF 组合字符，否则会与常规指令冲突引起错误。弹出菜单。该指令主要配合下拉菜单使用，详细使用可以参考本章节的设置下拉菜单内容。

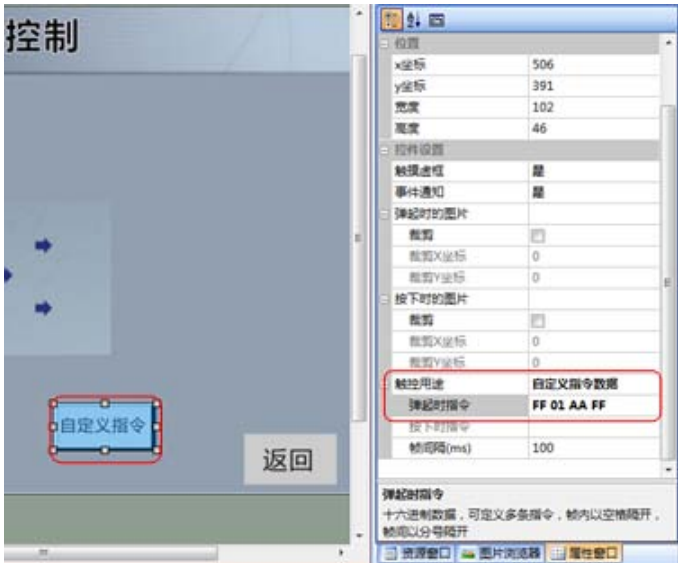


图 2-9 自定义指令数据

2.16 设置按钮弹起或按下状态

该指令主要实现将画面中某个按钮强行按下或弹起,也就是说用户除了点击触摸设置按钮按下或弹起外,还可以单片机发送命令来设置按钮状态。

指令格式: EE 【B1 10 Screen\_id Control\_id Status】 FF FC FF FF

参数说明: Screen\_id(2 个字节): 画面编号

Control\_id(2 个字节): 按钮控件编号

Status(1 个字节) : 按钮状态

0x00: 设置按钮由按下变成弹起状态

0x01: 设置按钮从弹起变成按下状态

该指令主要用于将“按下”的按钮变成“弹起”或将“弹起”的按钮变成“按下”状态。在某些场合中,按钮之间存在互斥性,当某个按钮被按下后,另外一个按钮必须弹起,则可以使用该指令来实现。

例如图 2-10 所示,用户按下“启动运行”按钮后,需要将“停止运行”按钮(画面 ID 为 2,控件 ID 为 4)由“按下”变成“弹起”状态,则单片机发送指令: EE B1 10 00 02 00 04 00 FF FC FF FF。

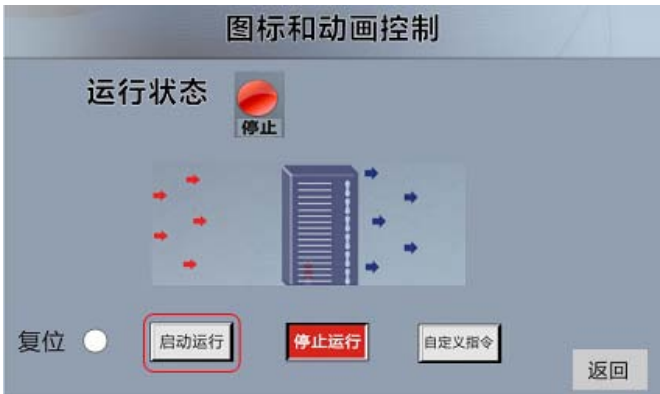


图 2-10 设置按钮按下和弹起状态

程序参考代码:

```
{
```



```
..... //检测到启动运行被按下
SetButtonValue(2,4,0); // 将画面 2，控制 ID 位 4 的停止运行按钮设置为弹起状态
}
```

## 2.17 对内指令实现按钮互斥和状态显示

所谓对内指令就是用户可以设置按下某个键后，屏幕内部还执行一些指令串，实现一些状态和逻辑显示，无需外部单片机参与，节省程序代码。

如图 2-11 所示，若用户按下“启动运行”按钮后，必须将“停止运行”按钮(画面 ID 为 4，控件 ID 为 2)进行弹起，则可以直接在 PC 中进行配置。属性窗口选择：对内指令--按下时→EE B1 10 00 04 00 02 01 FF FC FF FF，这样按下“启动运行”后，屏幕对内发送弹起“停止运行”的命令，实现按钮自动弹起，无需 4.4 章节所述外部单片机参与。

当然，除了可以配置按下“启动运行”，弹起“停止运行”外，还可以在对内指令框内输入多条其它的指令，例如启动 gif 动画，改变运行状态标志等，多条指令用分号隔开。



图 2-11 按钮互斥配置

## 2.18 对外指令输出

所谓对外指令就是用户可以设置按下某个键后，屏幕对外执行用户自定义的指令串，类似 4.3 章节按钮控件中的自定义指令。

## 2.19 读取按钮控件状态

指令格式：EE 【B1 11 Screen\_id Control\_id】 FF FC FF FF

参数说明：Screen\_id(2 个字节)：画面编号

Control\_id(2 个字节)：控件编号

该指令主要用于查询某个按钮当前是“按下”还是“弹起”状态。

返回指令格式：EE B1 11 Screen\_id Control\_id Control\_type Subtype Status FF FC FF FF

参数说明：Screen\_id(2 个字节)：画面编号

Control\_id(2 个字节)：控件编号

Control\_type(1 个字节)：固定值 0x10，表示为按钮控件

Subtype(1 个字节)：固定值 0x01，表示按钮为开关描述类型

Status (1 个字节): 按钮状态

0x00: 按钮为弹起状态

0x01: 按钮为按下状态

## 2.20 更新文本控件数值

文本更新方式有三种：用户主机输入、弹出系统键盘输入和自定义键盘输入。

1. 用户主机输入。屏幕上显示的数据来自于用户单片机输入。

指令格式：EE【B1 10 Screen\_id Control\_id Strings】FF FC FF FF

参数说明：Screen\_id(2 个字节)：画面编号

Control\_id(2 个字节)：控件编号

Strings（不定长）：用户写入的字符串

该指令主要用于文本显示。由于系统对所有文本变量分配了内存地址，当屏幕从其它界面再次返回到文本界面时，文本数据依然保存，无须再次重新刷新数据。

使用文本控件时，用户首先在上位机进行相关参数配置，譬如字体大小、前景色、背景色和文本输入方式等，如图 2-12 所示，然后主机只需要直接对相应 ID 写入变化的数据即可。



图 2-12 文本控件参数配置

弹出系统键盘输入。当点击文本框时，系统自动弹出内置键盘，用户输入完所需字符后点击确定，设备会将输入的字符显示在文本框内，同时将字符的 ASCII 码上传到主机。

例如，用户预先设置好文本控件，输入方式选择“弹出键盘输入”，如图 2-13 所示。点击文本框，屏幕将自动弹出系统小键盘，录入“123”后点击 Enter，此时输入的数字将自动显示到文本框内，同时上传录入的 ASCII 码，上传格式与下面即将介绍的“读取文本控件数值”返回格式一样。用户单片机解析上传的指令即可知道录入的数据。





图 2-13 弹出系统小键盘

若用户需要录入中文字符，只需将键盘类型选择为全键盘，然后点击键盘上的中英文切换键，如图 2-14 所示。

注意：因为显示区域的原因，只有 3.5 寸以上（不包括 3.5 寸）的尺寸才支持全键盘和输入法。



图 2-14 弹出系统全键盘和中英文输入

自定义按键输入。文本框内的数据来自于同一画面内的键盘输入。如图 2-15 所示，若用户需要实现密码输入，操作时先利用按钮控件的自定义键值功能，将每个按钮定义为对应的 ASCII 字符（参考图 2-8 所示），然后在密码框内放置一个文本控件，将属性输入方式设置为“自定义按键输入”。这样点击文本框后，界面将自动出现光标闪动，然后点击右边键盘输入，对应的数字就自动显示在文本框中，同时输入的字符 ASCII 码也将上传给主机。

提示：配置完毕后，建议运行“虚拟串口屏”，查看按下每个按钮后屏幕上传的信息。



图 2-15 自定义按键输入

## 2.21 清除文本控件内容

指令格式: EE 【B1 10 Screen\_id Control\_id】 FF FC FF FF

参数说明: Screen\_id(2 个字节): 画面编号

Control\_id(2 个字节): 控件编号

该指令主要用于清除当前文本控件的数值。

## 2.22 读取文本控件数值

指令格式: EE 【B1 11 Screen\_id Control\_id】 FF FC FF FF

参数说明: Screen\_id(2 个字节): 画面编号

Control\_id(2 个字节): 控件编号

该指令主要用于获取当前文本控件的数值。对于一些重要参数, 用户可以采用获取文本控件数值命令来重新校验。

返回指令格式: EE B1 11 Screen\_id Control\_id Control\_type Strings FF FC FF FF

返回参数: Screen\_id(2 个字节): 画面编号

Control\_id(2 个字节): 控件编号

Control\_type(1 个字节): 固定值 0x11, 表示为文本控件

Strings(不定长): 当前显示的文本值, 文本后面附加 1 个 0x00 作为结束符

## 2.23 设置光标焦点

指令格式: EE 【B1 02 Screen\_id Control\_id Enable】 FF FC FF FF

参数说明: Screen\_id(2 个字节): 画面编号

Control\_id(2 个字节): 控件编号

Enable (1 个字节): 0x00 表示关闭, 0x01 表示开启显示;

该指令主要将光标符号移植到指定的文本控件上显示, 更直观提醒用户录入文本。

提示: 仅在自定义键盘中有效。

## 2.24 设置文本控件闪烁

指令格式: EE 【B1 15 Screen\_id Control\_id Cycle】 FF FC FF FF

参数说明: Screen\_id(2 个字节): 画面编号

Control\_id(2 个字节): 控件编号

Cycle(2 个字节): 闪烁周期(10 毫秒单位), 0 表示不闪烁

该指令主要用于实现文本控件交替闪烁, Cycle 值为 0 时停止闪烁。

## 2.25 设置文本控件滚动速度

指令格式: EE 【B1 16 Screen\_id Control\_id Speed】 FF FC FF FF

参数说明: Screen\_id(2 个字节): 画面编号

Control\_id(2 个字节): 控件编号

Speed (2 个字节): 文本滚动速度(每秒移动像素), 0 表示不滚动

该指令主要用于实现文本滚动显示, 默认从右到左滚动, Speed 值为 0 表示停止滚动。

## 2.26 设置文本控件背景色

指令格式: EE 【B1 18 Screen\_id Control\_id BK\_Color】 FF FC FF FF

参数说明: Screen\_id(2 个字节): 画面编号

Control\_id(2 个字节): 控件编号

BK\_Color (2 个字节): 背景色 RGB 颜色值

该指令主要用于运行过程中添加文本控件背景显示, 实现文本被选中或突出的效果。

如何获取背景/前景 RGB 颜色值? 用户可以任意新建一个画面, 绘制一个矩形, 将矩形的颜色设置为自己所需颜色, 然后鼠标放置在矩形上, 软件左下角就会显示当前屏幕鼠标点的 RGB565 颜色值, 如所图 2-16 示。

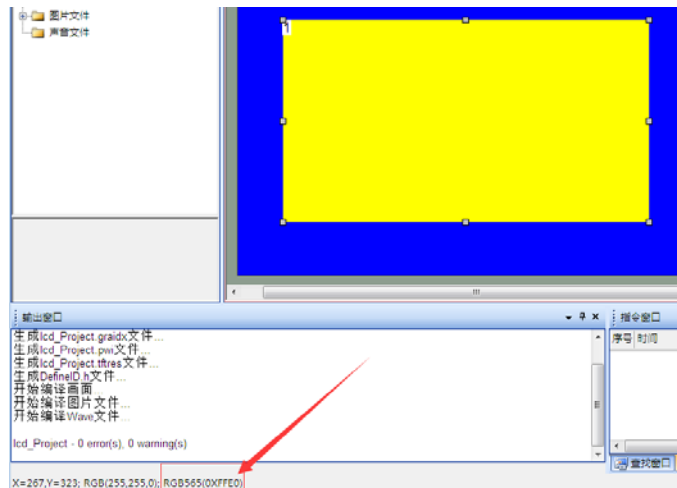


图 2-16 前景/背景色颜色提取

## 2.27 设置文本背景透明

该指令主要用于取消带背景文本控件显示,

指令格式: EE 【B1 17 Screen\_id Control\_id】 FF FC FF FF

参数说明: Screen\_id(2 个字节): 画面编号

Control\_id(2 个字节): 控件编号

## 2.28 设置文本控件前景色

该指令主要用于在运行过程中更换文本控件颜色显示, 实现一些告警、突出显示等。

指令格式: EE 【B1 19 Screen\_id Control\_id FORE\_Color】 FF FC FF FF

参数说明: Screen\_id(2 个字节): 画面编号

Control\_id(2 个字节): 控件编号

FORE\_Color (2 个字节): 前景色 RGB 颜色值

## 2.29 格式化文本显示

指令格式: EE 【B1 07 Screen\_id Control\_idSign Fill\_zero Value】 FF FC FF FF

参数说明: Screen\_id(2 个字节): 画面编号

Control\_id(2 个字节): 控件编号

Sign(1 个字节): 数据类型, 分别为:

Unsigned int: 无符号整型, 0x00;

Int: 有符号整型, 0x01;

Float: 单精度浮点数, 0x02;

Double: 双精度浮点数, 0x03;

Fill\_zero (1 个字节): 小数位数, 若选择不足为补 0 的话, 需要加 0x80;

Value (4 个字节): 添加的数据, 用十六进制数表示;

该指令主要用于按照相应的条件进行输入文本, 按照相应的限制条件来显示文本数据。

## 2.30 文本控件数字增量调节指令

指令格式: EE 【B1 1A Screen\_idControl\_idOptionDeltaMin\_LimitMax\_Limit】 FF FC FF FF

参数说明: Screen\_id(2 个字节): 画面编号

Control\_id(2 个字节): 文本控件编号

Option(1 个字节): bit0-0 减少, 1 增加; bit1-0 不循环调节, 1 循环调节。

增加不循环指令为 0x01; 增加循环指令为 0x03;

减少不循环指令为 0x00; 减少循环指令为 0x02;

Delta(2 个字节): 增量, 即每次调节的幅度;

Min\_Limit(2 个字节): 最小值;

Max\_Limit(2 个字节): 最大值;

该指令一般用于点击按钮控件时, 对文本控件显示数值进行递增或递减, 一般用作按钮控件的对内指令。如果按钮控件的操作风格设置为长按时, 可对文本控件进行持续递增或递减。

## 2.31 更新进度条控件数值

指令格式: EE 【B1 10 Screen\_id Control\_id Progressvalue】 FF FC FF FF

参数说明: Screen\_id(2 个字节): 画面编号

Control\_id(2 个字节): 控件编号

Progressvalue (4 个字节): 新的进度条值

该指令主要实现进度的递增或递减。

使用进度条控件时, 用户首先在上位机进行控件相关参数配置, 譬如前景图、背景图、数值最大值和最小值等, 如图 2-17 所示, 然后主机只需要对相应的 ID 写入进度条数值即可实现进度条滚动。

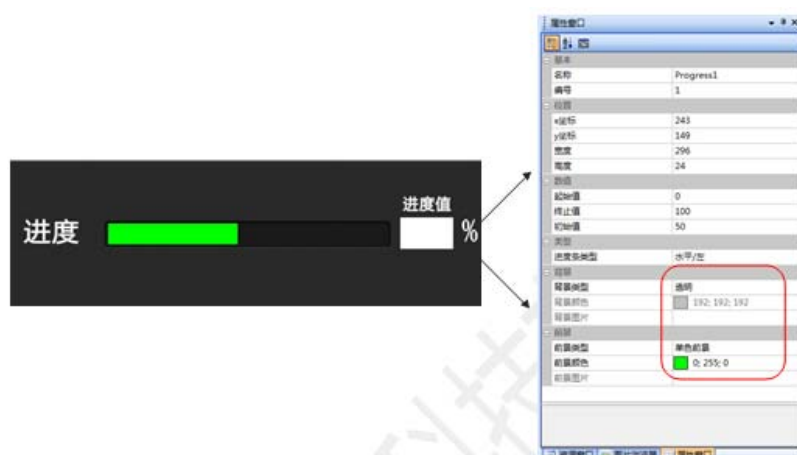


图 2-17 进度条控件配置信息

## 2.32 读取进度条控件值

指令格式: EE 【B1 11 Screen\_id Control\_id】 FF FC FF FF

参数说明: Screen\_id(2 个字节): 画面编号

Control\_id(2 个字节): 控件编号

该指令主要用于获取当前进度条的数值。

返回指令格式: EE B1 11【Screen\_id Control\_id Control\_type Progressvalue】 FF FC FF FF

返回参数: Screen\_id(2 个字节): 画面编号

Control\_id(2 个字节): 控件编号

Control\_type(1 个字节): 固定值 0x12, 表示为进度条控件

Progressvalue(4 个字节): 当前进度条数值

## 2.33 滑动条控件上传格式

使用滑块控件时, 用户首先在上位机进行控件相关参数配置, 譬如标尺长度、方向、游标图片、背景图和数值等, 如图 2-18 所示, 当用户拖动游标时, 设备将不断上传当前游标数值给主机, 通过判断游标数值即可知道当前滑块所在位置。

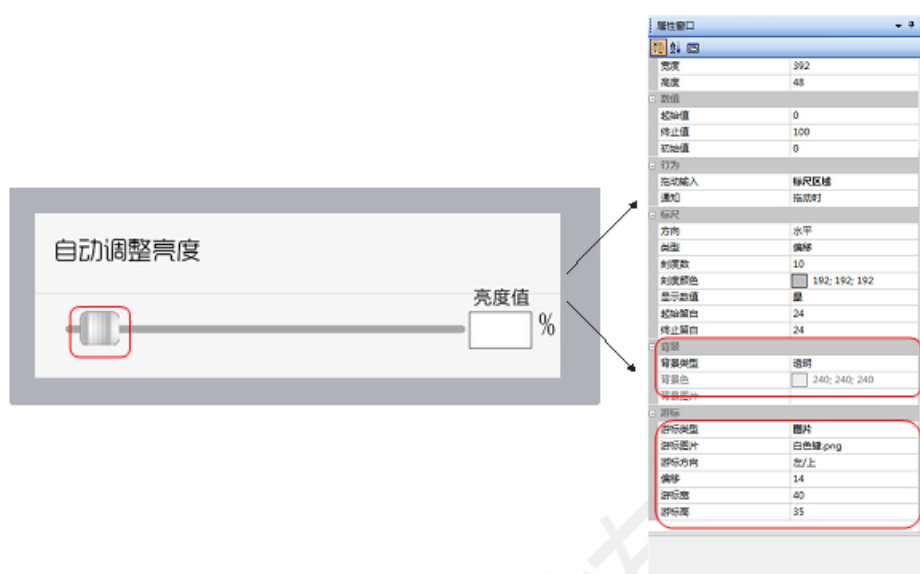


图 2-18 滑块控件配置图

滑块控件上传格式:

EE 【B1 11 Screen\_id Control\_id Control\_type Slidervalue】 FF FC FF FF

参数说明: Screen\_id(2 个字节): 画面编号

Control\_id(2 个字节): 控件编号

Control\_type(1 个字节): 固定值 0x13, 表示为滑块控件

Slidervalue (4 个字节): 表示当前游标数值

### 2.34 设置进度条的背景色

指令格式: EE 【B1 18 Screen\_id Control\_id Bk\_Color】 FF FC FF FF

参数说明: Screen\_id(2 个字节): 画面编号

Control\_id(2 个字节): 控件编号

Bk\_Color (2 个字节): 背景色 RGB 颜色值

该指令主要用于设置进度条的背景色, 用户可以通过指令修改进度条的背景色。

### 2.35 设置进度条的前景色

指令格式: EE 【B1 19 Screen\_id Control\_id Bk\_Color】 FF FC FF FF

参数说明: Screen\_id(2 个字节): 画面编号

Control\_id(2 个字节): 控件编号

Fore\_Color (2 个字节): 前景色 RGB 颜色值

该指令主要用于设置进度条的前景色, 用户可以通过指令修改进度条的前景色。

### 2.36 更新滑动条控件数值

指令格式: EE 【B1 10 Screen\_id Control\_id Slidervalue】 FF FC FF FF

参数说明: Screen\_id(2 个字节): 画面编号

Control\_id(2 个字节): 控件编号

Slidervalue (4 个字节): 新的游标数值

该指令主要用于控制滑块游标显示的位置。用户主机可以发送相应的指令控制游标强制在某一个位置显示。

### 2.37 读取滑动条控件值

指令格式: EE 【B1 11 Screen\_id Control\_id】 FF FC FF FF

参数说明: Screen\_id(2 个字节): 画面编号

Control\_id(2 个字节): 控件编号

该指令主要用于获取当前游标所在的数值。

返回指令格式: EE B1 11 【Screen\_id Control\_id Control\_type Slidervalue】 FF FC FF FF

返回参数: Screen\_id(2 个字节): 画面编号

Control\_id(2 个字节): 控件编号

Control\_type(1 个字节): 固定值 0x13, 表示为滑块控件

Slidervalue (4 个字节): 表示当前游标数值

### 2.38 RTC 时钟设置

指令格式: EE 【81 Sec Min Hour Day Week Mon Year】 FF FC FF FF

参数说明: Sec: 秒设置 Min: 分设置

Hour: 小时设置 Day: 日设置

Week: 星期设置 Mon: 月设置



Year: 年设置

以上参数均为 1 个字节，以 BCD 码表示，星期天设置为 0x00

该指令主要用于当前时间参数的设定，建议用户直接通过上位机软件进行设置。建议用户使用时钟控件来显示 RTC，可以直接点击触摸弹出键盘来校准当前时间。

2.39 读取 RTC 时钟

指令格式: EE 【82】 FF FC FF FF

参数说明: 无

该指令主要用于获取当前时间数值。数据上传格式: EE F7 Year Mon Week Day Hour Min Sec FF FC FF FF。以上参数均为 1 个字节，以 BCD 码表示

2.40 更新仪表控件数值

指令格式: EE 【B1 10 Screen\_id Control\_id Metervalue】 FF FC FF FF

参数说明: Screen\_id(2 个字节): 画面编号

Control\_id(2 个字节): 控件编号

Metervalue (4 个字节): 新的仪表值

使用仪表控件时，用户首先在上位机进行控件相关参数配置，譬如表盘、刻度、指针和数值等，如图 2-19 所示，然后主机只需要发送相应的数值就可以实现仪表指针的转动。

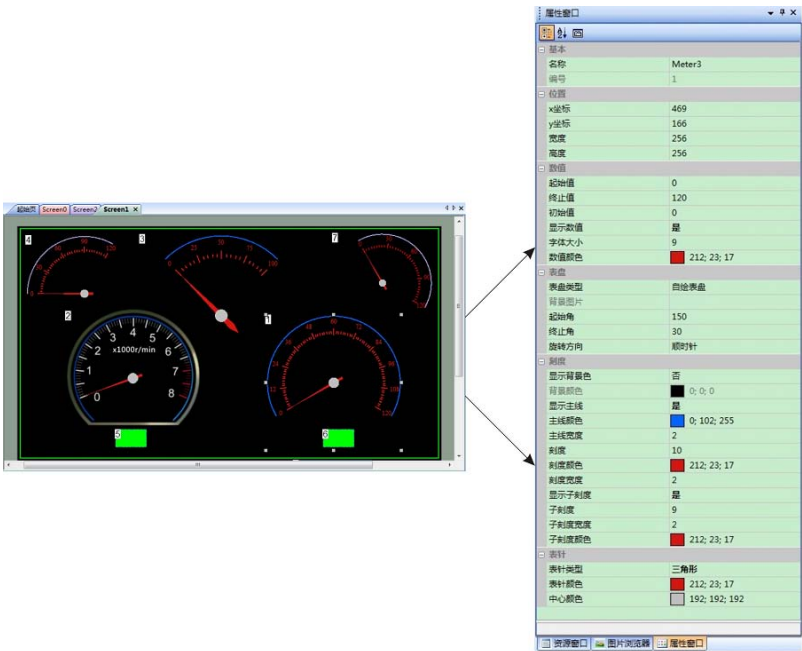


图 2-19 仪表控件配置信息

例如用户需要对第 0 个画面、控件 ID 为 4 的仪表控件写入数值 100，程序代码如下。  
程序参考代码:

```
{
.....
SetMeterValue (0,4,100) //对画面 0 的控件 ID 为 4 的仪表控件写入数值 100、
}
```

2.41 读取仪表控件数值

指令格式: EE 【B1 11 Screen\_id Control\_id】 FF FC FF FF

参数说明: Screen\_id(2 个字节): 画面编号

Control\_id(2 个字节): 控件编号

该指令主要用于获取当前仪表显示的数值。

返回指令格式: EE B1 11 Screen\_id Control\_id Control\_type Metervalue FF FC FF FF

返回参数: Screen\_id(2 个字节): 画面编号

Control\_id(2 个字节): 控件编号

Control\_type(1 个字节): 固定值 0x14, 表示为仪表控件类型

Metervalue (4 个字节): 当前显示的仪表值

## 2.42 动画控件显示

通过动画控件调用 gif 动画显示, 可在同一画面内支持多个 gif 显示, 用户发送相应的指令就可以控制动画播放、停止、暂停和上/下帧等功能, 指令介绍如表 2.1 所示。

表 2.1 动画控件指令表

功能	指令格式
启动动画播放	EE 【B1 20Screen_id Control_id】 FF FC FF FF
	Screen_id(2 个字节): 画面编号
	Control_id(2 个字节): 控件编号
	启动播放后, 动画每次从帧头 0 开始播放
停止动画播放	EE 【B1 21Screen_id Control_id】 FF FC FF FF
	Screen_id(2 个字节): 画面编号
	Control_id(2 个字节): 控件编号
	停止播放后, 下次将从帧头 0 开始播放
暂停动画播放	EE 【B1 22Screen_id Control_id】 FF FC FF FF
	Screen_id(2 个字节): 画面编号
	Control_id(2 个字节): 控件编号
	暂停后, 下次将从暂停帧开始继续播放
指定帧播放	EE 【B1 23Screen_id Control_idFlashImgae_ID】 FF FC FF FF
	Screen_id(2 个字节): 画面编号
	Control_id(2 个字节): 控件编号
	FlashImgae_ID(1 个字节): 某一动画面 ID 指定从某一帧开始播放
播放上一帧	EE 【B1 24Screen_id Control_id】 FF FC FF FF
	Screen_id(2 个字节): 画面编号
	Control_id(2 个字节): 控件编号
	显示当前帧的上一帧内容
播放下一帧	EE 【B1 25Screen_id Control_id】 FF FC FF FF
	Screen_id(2 个字节): 画面编号
	Control_id(2 个字节): 控件编号
	显示当前帧的下一帧内容
动画控件值上传	EE 【B1 26Screen_id Control_idStatusFlashImgae_ID】 FF FC FF FF
	按下某个动画控件时候, 设备上传的如下信息:
	Screen_id(2 个字节): 画面编号
	Control_id(2 个字节): 控件编号
	Status (1 个字节): 0x00 表示触摸按下, 0x01 表示弹起



	FlashImgae_ID (1 个字节): 表示按下屏幕时, 此刻显示的画面帧 ID 备注: 用户可以在 PC 软件配置禁止/使能动画控件值上传。
--	---

使用时, 用户首先通过动画控件调用 gif 动画进行显示, 然后设置是否需要触摸通知(也就是点击动画是否需要上传动画控件 ID 值)和自动播放, 如图 2-20 所示。动画播放的间隔时间自动从 gif 原文件中提取。

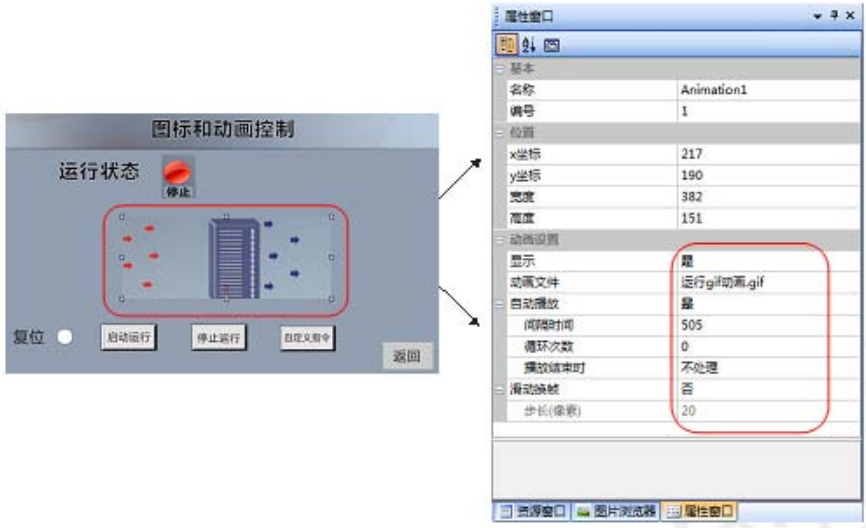


图 2-20 动画控件配置图

2.43 图标控件显示

指令格式: EE 【B1 23 Screen\_id Control\_id IconImgae\_ID】 FF FC FF FF

参数说明: Screen\_id(2 个字节): 画面编号

Control\_id(2 个字节): 控件编号

IconImgae\_ID (1 个字节): ICON 文件中某一图标帧 ID

该指令主要解决画面内同一位置不同状态图的切换或消失显示。

如何生成 ICON 图标? 使用前用户先点击 VisualTFT 软件的工具菜单, 选择→图标生成, 弹出如图 2-21 所示的画面, 然后将预先做好的 3 张 62x82 像素图片(停止、运行和透明 PNG)添加进去, 按照排列的顺序, 分别第 1 帧为停止帧、第 2 帧为运行和第 3 帧为透明图片, 最后点击生成图标。这样一个新的 ICON 文件就生成了, 里面包含了 3 帧图片。



图 2-21 图标生成工具

使用时, 用户首先通过图标控件调用新生成的 ICON 进行显示, 然后设置是否需要触摸通知(也就是点击图标控件是否需要上传 ID)和自动播放, 如图 2-22 所示。画面默认显示 ICON 的第 1 帧, 若需要切换状态图显示, 主机程序只需发送 ICON 编号和第几帧即可完成显示。透明帧其实就是一个空的 PNG 图片, 显示透明帧就可以达到图标消失或隐藏的效果。

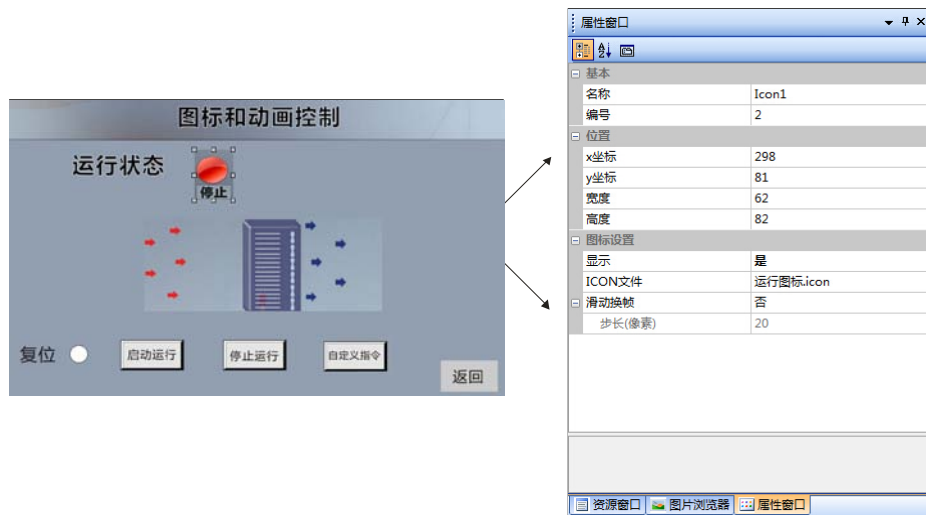


图 2-22 图标控件配置图

## 2.44 读取图标控件数值

指令格式: EE 【B1 11 Screen\_id Control\_id】 FF FC FF FF

参数说明: Screen\_id(2 个字节): 画面编号

Control\_id(2 个字节): 控件编号

该指令主要用于获取当前图标显示的帧 ID。

返回指令格式: EE B1 11 Screen\_id Control\_id Control\_type IconImgae\_ID FF FC FF FF

返回参数: Screen\_id(2 个字节): 画面编号

Control\_id(2 个字节): 控件编号

Control\_type(1 个字节): 固定值 0x16, 表示为图标控件类型

IconImgae\_ID (2 个字节): 当前图标显示的帧 ID

## 2.45 图标控件值上传

指令上传格式: EE 【B1 26 Screen\_id Control\_id Status IconImgae\_ID】 FF FC FF FF

参数说明: Screen\_id(2 个字节): 画面编号

Control\_id(2 个字节): 控件编号

Control\_type(1 个字节): 固定值 0x16, 表示为图标控件类型

IconImgae\_ID (1 个字节): ICON 文件中某一图片帧 ID

若在 PC 软件选择了图标控件触摸通知, 用户点击图标后, 设备将上传给图标控件的 ID 和当前图片帧的 ID。

## 2.46 设置图标位置

指令上传格式: EE 【B1 28 Screen\_id Control\_id X Y】 FF FC FF FF

参数说明: Screen\_id(2 个字节): 画面编号

Control\_id(2 个字节): 控件编号

X (2 个字节): 显示的 X 坐标位置

Y (2 个字节): 显示的 Y 坐标位置

该指令主要用于动态调整图标控件显示的位置。

2.47 批量更新控件数值

指令上传格式：EE 【B1 12 Screen\_id+Control\_id0+Len0+ Strings0+ +...Control\_idn+ Lenn+ Stringsnn】 FF FC FF FF

- 参数说明：Screen\_id(2 个字节)：画面编号
- Control\_id0 (2 个字节)：第 1 个控件的编号
- Len0 (2 个字节)：第 1 个控件的字节长度
- Strings0 (不定长)：第 1 个控件的数值，字节数以 Len0 长度而定
- Control\_idn (2 个字节)：第 n 个控件的编号
- Lenn (2 个字节)：第 n 个控件的字节长度
- Stringsnn (不定长)：第 n 个控件的数值，字节数以 Lenn 长度而定

批量更新控件指令解决了一个画面内有太多更新数据而刷新速度低的问题。使用批量更新不仅节省了指令传输的时间，而且在同一个时刻所有数据一起更新。

批量更新指令的测试，可以打开 VisualTFT 的《指令助手》，然后输入几个文本控件值进行验证测试，如图 2-23 所示。

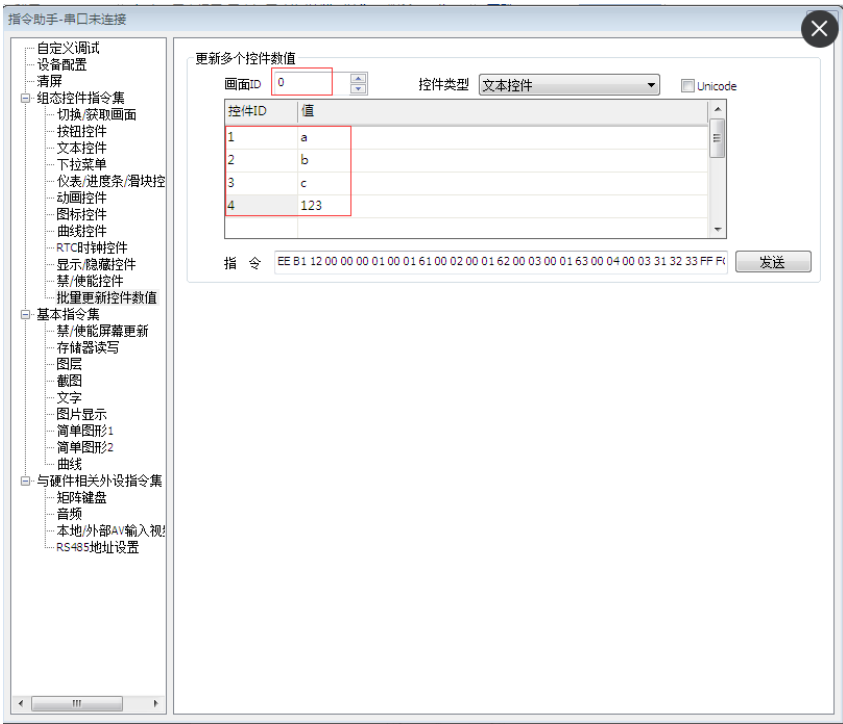


图 2-23 批量更新文本控件

2.48 曲线控件显示

若需要实现动态图表数据采集，自动左右平移，曲线控件将给用户带来极大便捷。指令介绍如表 2.2 所示，上位机软件配置图如图 2-24 所示。

建议：曲线控件的详细使用教程，可以登录 [www.gz-dc.com](http://www.gz-dc.com) 的资料下载栏进行下载。

表 2.2 曲线控件指令表

功能	指令格式
添加数据通道	EE 【B1 30 SCREEN_ID CONTROL_ID CHANNEL COLOR】 FF FC FF FF

	<p>指定数据通道和曲线颜色</p> <p>SCREEN_ID (2 个字节): 画面编号</p> <p>CONTROL_ID (2 个字节): 控件编号</p> <p>CHANNEL(1 个字节): 共 8 个数据通道, 编号范围 (0~7)</p> <p>COLOR(2 个字节): 数据通道颜色</p>
删除指定数据通道	<p>EE 【B1 31 SCREEN_IDCONTROL_ID CHANNEL】FF FC FF FF</p> <p>SCREEN_ID (2 个字节): 画面编号</p> <p>CONTROL_ID (2 个字节): 控件编号</p> <p>CHANNEL(1 个字节): 共 8 个数据通道, 编号范围 (0~7)</p>
指定垂直/水平的缩放/平移	<p>EE 【 B1 34 SCREEN_IDCONTROL_ID XOFFSET XMUL YOFFSET YMUL】FF FC FF FF</p> <p>指定垂直/水平的缩放/平移</p> <p>SCREEN_ID (2 个字节): 画面编号</p> <p>CONTROL_ID (2 个字节): 控件编号</p> <p>XOFFSET (2 个字节): 水平偏移数据点数, 左移为正, 右移为负</p> <p>XMUL (2 个字节): 水平缩放系数, 单位 0.01</p> <p>YOFFSET (2 个字节): 垂直偏移数值, 下移为正, 上移为负</p> <p>YMUL (2 个字节): 垂直缩放系数, 单位 0.01</p> <p>采样点与坐标点的计算公式: 第 N 个采样点的数值为 V</p> <p>X 坐标 = (N-XOFFSET)*XMUL*0.01</p> <p>Y 坐标 = (V-YOFFSET)*YMUL*0.01</p>
指定通道前端添加新数据	<p>EE 【B1 32 SCREEN_ID CONTROL_ID CHANNEL DATA_LEN DATA】FF FC FF FF</p> <p>在指定数据通道前端添加新数据, 当数据长度超过缓冲区长度时, 旧数据右移</p> <p>SCREEN_ID (2 个字节): 画面编号</p> <p>CONTROL_ID (2 个字节): 控件编号</p> <p>CHANNEL(1 个字节): 共 8 个数据通道, 编号范围 (0~7)</p> <p>DATA_LEN(2 个字节): 数据长度</p> <p>DATA : 不定长数据, 长度由 DATA_LEN 指定格式: 数据代表是 Y 轴方向的值, X 轴方向的会根据水平缩放系数自动递增, 例如当水平缩放系数为 1 的时候, 每插入一个点 X 轴自动加 1, 当水平缩放系数为 5 的时候, 每插入一个点 X 轴自动加 5</p>
指定通道末尾添加新数据	<p>EE 【B1 35 SCREEN_ID CONTROL_ID CHANNEL DATA_LEN DATA】FF FC FF FF</p> <p>在指定数据通道末尾插入新数据, 当数据长度超过缓冲区长度时, 旧数据左移</p> <p>SCREEN_ID (2 个字节): 画面编号</p> <p>CONTROL_ID (2 个字节): 控件编号</p> <p>CHANNEL(1 个字节): 共 8 个数据通道, 编号范围 (0~7)</p> <p>DATA_LEN(2 个字节): 数据长度</p> <p>DATA : 不定长数据, 长度由 DATA_LEN 指定</p>
清空指定通道的数据	EE 【B1 33 SCREEN_ID CONTROL_ID CHANNEL】FF FC FF FF

	清空指定通道的数据 SCREEN_ID (2 个字节): 画面编号 CONTROL_ID (2 个字节): 控件编号 CHANNEL(1 个字节): 共 8 个数据通道, 编号范围 (0~7)
--	---

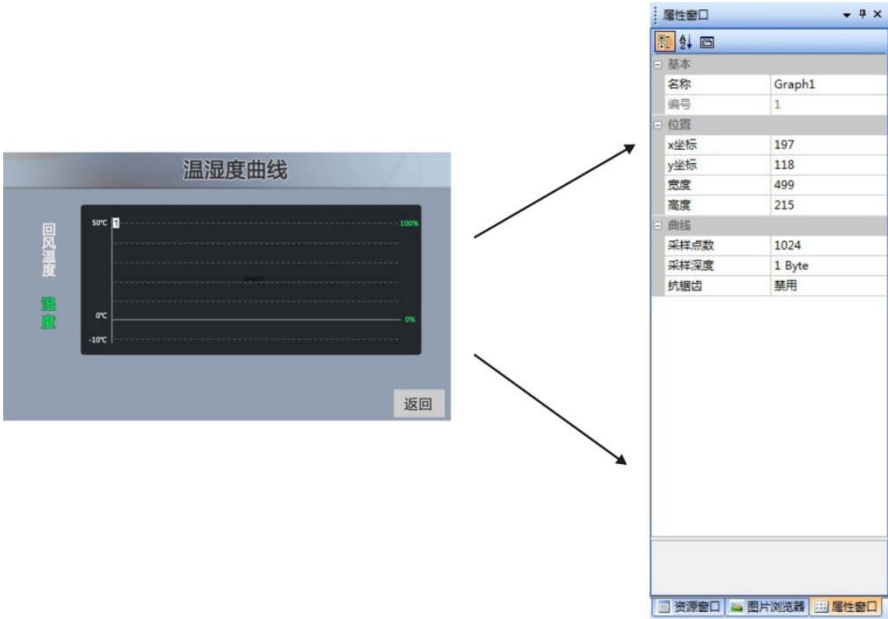


图 2-24 曲线控件配置图

2.49 设置下拉菜单写入的文本控件

指令格式: EE 【B1 13Screen\_id Control\_id Enable Textctrl\_ID】 FF FC FF FF

参数说明: Screen\_id(2 个字节): 画面编号

Control\_id(2 个字节): 控件编号

Enable (1 个字节): 菜单使能。1 显示菜单; 0 不显示菜单

Textctrl\_ID(2 个字节) : 需写入菜单数据的文本控件编号

该指令主要设置弹出下拉菜单后, 菜单选项数据写入到哪个文本控件中。

假设用户需要实现如下功能: 点击图 2-25 所示下拉按钮①, 弹出下拉菜单②, 用户选择 57.7V 数据项③, 此时该数据自动写入到文本控件 (文本框)中, 同时屏幕上传菜单控件 ID 和选择项编号给用户单片机, 则操作如下。

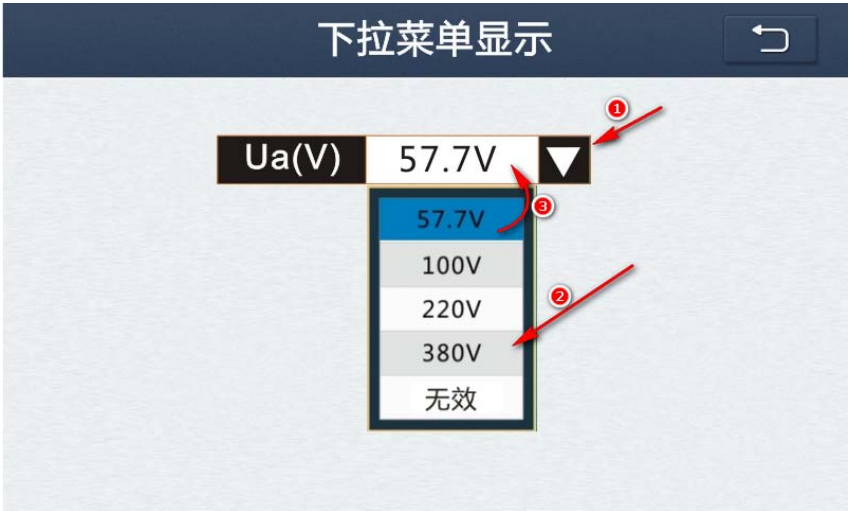


图 2-25 下拉菜单显示效果

操作步骤:

1. 准备 2 张菜单图片,如图 2-26 所示,选中和未选中的。若不需要按下去的选中效果,只需未选中的 1 张图片即可。

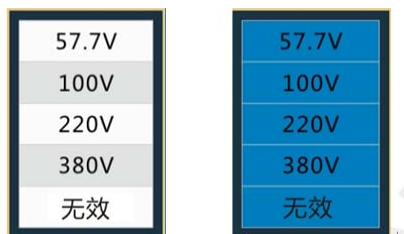


图 2-26 未选中(左)和选中(右)的下拉菜单图片

设置下拉菜单属性。按照图 2-27 所示,点击①处的下拉菜单控件,拖放在图中②位置处,在右边的属性窗口③处,将菜单外观设置为:自定义图片,然后分别在④、⑤处设置好弹起和按下的菜单图片,菜单项数⑥处设置为 5,表示有 5 个下拉项,最后在⑦处填入所有的下拉数据。数据与数据之前使用分开隔开,必须半角的分号。



图 2-27 设置下拉菜单属性

设置弹出菜单的按钮和写入的文本控件。如图 2-27 所示,在①处放入文本控件,控件 ID 为 2,然后在②处放置 1 个按钮控件,设置触控用途为弹出菜单,如③所示,最后在④、⑤处分别填入菜单控件 ID 和文本控件 ID 值。





图 2-28 设置下拉菜单按钮和文本控件

操作完毕。编译后运行虚拟串口屏，点击下拉按钮，即会弹出菜单，然后选择菜单项，数据加载到文本控件并上传 ID 信息，如图 2-29 所示。



图 2-29 下拉菜单上传指令

## 2.50 下拉菜单控件值上传

指令格式：EE 【B1 14 Screen\_idControl\_id1A Meundata\_ID Status】FF FC FF FF

参数说明：Screen\_id(2 个字节)：画面编号

Control\_id(2 个字节)：控件编号

0x1A(1 个字节)：固定数值，表示下拉菜单

Meundata\_ID(1 个字节)：下拉菜单中的第几个选项数据

Status(1 个字节)：触摸状态，0 表示此刻按钮弹起，1 表示按钮按下

当弹出下拉菜单，用户选中所需数据后，此刻屏幕会将当前的菜单控件 ID 和选中的数值项编号上传给用户单片机。用户通过解析串口数据，即可了解当前按下的数据。

### 2.51 设置滑动选择控件值

指令格式：EE 【B1 10 Screen\_idControl\_idSelect\_data\_ID】 FF FC FF FF

参数说明：Screen\_id(2 个字节)：画面编号

Control\_id(2 个字节)：控件编号

Select\_data\_ID (2 个字节)：滑动选择控件中的选项 ID

该指令主要用于设置滑动选择控件当前选项，即通过滑动选择控件的选项 ID 号来进行索引，这个功能可以用滑动选择控件的“状态显示”功能，即发送相应的索引 ID，滑动选择控件会显示当前数据项。

### 2.52 读取滑动选择控件值

指令格式：EE 【B1 11 Screen\_idControl\_id】 FF FC FF FF

参数说明：Screen\_id(2 个字节)：画面编号

Control\_id(2 个字节)：控件编号

该指令主要用于获取滑动选择控件当前选项。

返回指令格式：EE B1 11 Screen\_id Control\_id 1Bvalue FF FC FF FF

返回参数：Screen\_id(2 个字节)：画面编号

Control\_id(2 个字节)：控件编号

0x1B (1 个字节)：固定数值，表示滑动选择控件

value (1 个字节)：当前显示的滑动选择控件选中项的索引值

### 2.53 滑动选择控件值上传

指令格式：EE 【B111Screen\_idControl\_id1B Select\_data\_ID】 FF FC FF FF

参数说明：Screen\_id(2 个字节)：画面编号

Control\_id(2 个字节)：控件编号

0x1B (1 个字节)：固定数值，表示滑动选择控件

Select\_data\_ID (1 个字节)：选择菜单中的第几个选项数据

该指令主要实现上下滑动选择所需数据，选中后系统自动上传控件 ID 值和当前选中值。

假设用户需要实现图 2-30 所示亮度设置效果，通过手指上下滑动选择所需的数值，然后屏幕上传滑动选择控件 ID 和选择项编号给用户单片机，则操作如下。



图 2-30 滑动选择控件显示效果

操作步骤：

1. 如图 2-31 所示，点击软件菜单栏①处的滑动选择控件，在图②放置一个滑动选择控



- 件。
- 2. 在属性窗口③处设置字体大小为 ASCII16\*32，该大小是选中后显示的字体大小。
  - 3. 部分参数定义如下：
    - 中心颜色：表示被选中后数据显示颜色；
    - 两端颜色：表示最上和最下面数据显示颜色；
    - 两端缩放：表示最上和最下的字体大小与中间的百分比，50%表示两端字体为中间一半，0表示所有字体大小一样，不缩放。
    - 显示项数：固定为 3 和 5 两种。
    - 初始选项：表示默认选中的数据。
    - 候选数据：表示所有选择的数据。
  - 4. 在属性窗口进行④、⑤、⑥、⑦处参数设置。
  - 5. 在属性窗口⑧处设置候选数据：0;10;20;30;40;50;60;70;80;90。
  - 6. 操作完毕。运行虚拟串口屏，上下滑动时候，就可以看到屏幕上传的指令数据，如图 2-32 所示。



图 2-31 滑动选择控件属性设置

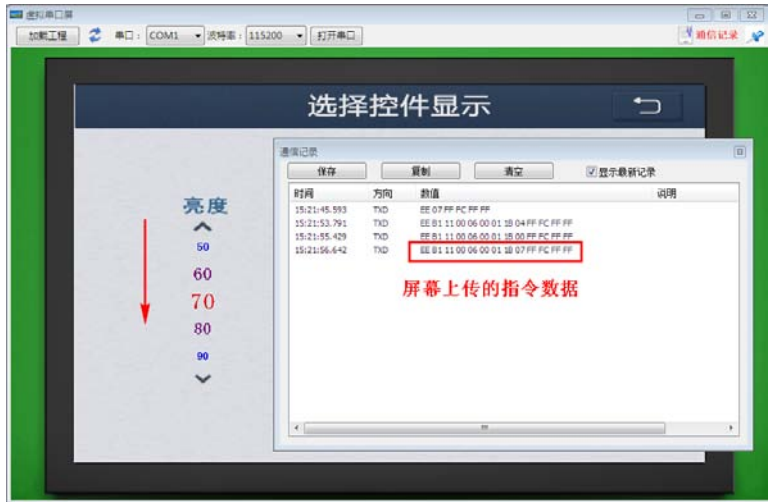


图 2-32 滑动选择控件上传指令数据

## 2.54 更新二维码控件

指令格式：EE【B1 10 Screen\_id Control\_id Strings】FF FC FF FF

参数说明：Screen\_id(2 个字节)：画面编号

Control\_id(2 个字节)：控件编号

Strings（不定长）：二维码字符内容，编码类型为 UTF8

**注意：**若发送的二维码内容有中文字符，必须转换为 UTF8 格式，用户需要熟悉 UTF8 编码格式，然后进行转换后发送。

该指令主要用于实现二维码图显示，指令格式与文本控件一样。用户需要显示什么内容的二维码，直接发送字符内容即可。

假设用户需要显示如图 2-33 所示的二维码，扫描内容：[www.gz-dc.com](http://www.gz-dc.com)，则操作如下：



图 2-33 二维码控件显示

操作步骤：

1. 如图 2-34 所示，点击导航①处的二维码控件图标，放置在画面②处区域。
2. 在属性窗口文本③处输入：[www.gz-dc.com](http://www.gz-dc.com)。
3. 操作完毕。运行虚拟串口屏即可显示图 2-33 所示二维码图像，用手机扫描即可准确识别。
4. 若用户单片机需要更新当前二维码内容，发送对应指令即可。假设图 2-34 中的画面 ID 为 5，二维码控件编号为 1，显示内容为 123，则发送指令如下：EE B1 10 00 05 00 01 31 32 33 FF FC FF FF，其中 31 32 33 表示 123 的 ASCII 字符。



图 2-34 二维码显示设置

## 2.55 启动定时器

指令格式: EE 【B1 41Screen\_id Control\_id】 FF FC FF FF

参数说明: Screen\_id(2 个字节): 画面编号

Control\_id(2 个字节): RTC 控件编号

该指令只针对硬件支持 RTC 的屏幕。使用时, 先点击导航栏中的 RTC 控件, 放置在画面右上角, 设置计数格式、计时长度和显示方式, 如图 2-35 所示, 然后用户单片机发送启动定时器指令, 时间就开始顺计时。

定时时间一到, 屏幕将上传事件通知, 上传格式: EE 【B1 43 Timedata 17】FF FC FF FF。

上传参数说明: Timedata (4 个字节): 用户设置的定时器时间数值, 高位在前。

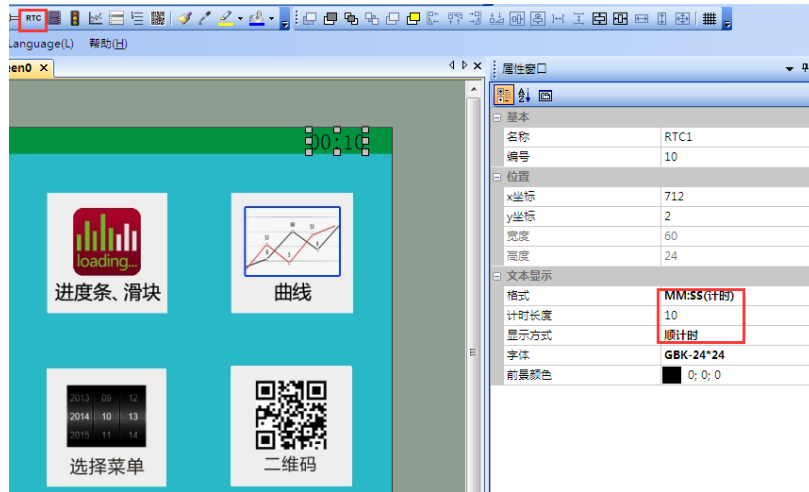


图 2-35 定时器设置

用户也可以使用设置定时器命令来修改定时参数。

## 2.56 设置定时器

指令格式: EE 【B1 40Screen\_id Control\_id Timedata】 FF FC FF FF

参数说明: Screen\_id(2 个字节): 画面编号

Control\_id(2 个字节): RTC 控件编号

Timedata (4 个字节): 定时器数值

该指令只针对硬件支持 RTC 的屏幕, 用于修改定时器时间参数。

## 2.57 启动定时器

指令格式: EE 【B1 41 Screen\_id Control\_id】 FF FC FF FF

参数说明: Screen\_id(2 个字节): 画面编号

Control\_id(2 个字节): RTC 控件编号

该指令只针对硬件支持 RTC 的屏幕, 用于启动定时器。

## 2.58 暂停定时器

指令格式: EE 【B1 44Screen\_id Control\_id】 FF FC FF FF

参数说明: Screen\_id(2 个字节): 画面编号

Control\_id(2 个字节): RTC 控件编号

该指令只针对硬件支持 RTC 的屏幕, 用于暂停计时器, 启动定时器后继续计时。

## 2.59 停止定时器

指令格式: EE 【B1 42Screen\_id Control\_id】 FF FC FF FF

参数说明: Screen\_id(2 个字节): 画面编号

Control\_id(2 个字节): RTC 控件编号

该指令只针对硬件支持 RTC 的屏幕, 用于停止定时器, 时间数值将恢复到初始值。

## 2.60 读取定时器

指令格式: EE 【B1 45 Screen\_id Control\_id】 FF FC FF FF

参数说明: Screen\_id(2 个字节): 画面编号

Control\_id(2 个字节): RTC 控件编号

该指令用于读取相应定时器的定时值, 发送该指令后, 屏幕会返回相应的定时值。

返回格式: EE B1 45 Screen\_id Control\_id 17 Strings FF FC FF FF

返回参数: Screen\_id(2 个字节): 画面编号

Control\_id(2 个字节): RTC 控件编号

17(1 个字节): 0x17, 固定码

Strings(4 个字节): 读取的定时值

## 2.61 手动禁止/使能屏幕更新

指令格式: EE 【B3 Enable】 FF FC FF FF

参数说明: Enable (1 个字节): 0x00 表示禁止更新, 0x01 表示使能更新;

该命令主要解决某一画面中实时动态更新的控件数目过多, 导致屏幕更新速度慢的问题。由于系统默认 100ms 自动刷新屏幕一次, 而有些场合用户数据量庞大, 系统自动更新无法满足时时性要求, 所以可以采用手动更新屏幕指令。

使用方法: 程序先发送禁止屏幕更新指令, 然后发送整个画面中需要更新的内容, 最后再使能屏幕更新, 这样更新的数据立刻显示在屏幕上。

## 2.62 显示/隐藏控件

指令格式: EE 【B1 03 Screen\_idControl\_id Enable】 FF FC FF FF

参数说明: Screen\_id(2 个字节): 画面编号

Control\_id(2 个字节): 控件编号

Enable (1 个字节): 0x00 表示隐藏控件, 0x01 表示显示控件;

该指令常用于某一时刻将指定按钮控件功能失效, 直至屏蔽被解除, 也可用于将某个动画/图标控件显示消失。

## 2.63 RS485 屏地址设置

指令格式: EE 【A501 Addr】 FF FC FF FF

参数说明: Addr (2 个字节): 地址信息

该指令仅支持 485 接口的屏幕。

该指令主要用于屏幕地址设置, 方便系统组网。地址设置成功后, 所有指令均包含地址信息, 例如清屏指令为: EE 01 FF FC FF FF, 加入地址后为 EE Addr1 Addr0 01 FF FC FF FF。

注意: 所有地址必须从 1 开始设置, 0 为广播地址, 所有屏都可以接收到广播信息。

## 2.64 RS485 屏地址取消

指令格式: EE 【00 00 A500】 FF FC FF FF

参数说明: 无

该指令仅支持 485 接口的屏幕。

该指令主要用于屏幕地址取消，指令中地址直接为广播地址。

## 2.65 锁定系统配置

指令格式：EE【08 A5 5A 5F F5】FF FC FF FF

参数说明：无

该指令防止在系统运行过程中，收到主机错误指令帧导致系统配置意外修改。一旦配置被锁定，设备将无法接收外部串口命令进行修改，直到锁定被解除。配置参数包括：波特率、触摸和矩阵键盘工作模式、自动背光调节参数，可以直接在 PC 的指令助手中进行配置。

## 2.66 解除系统配置锁定

指令格式：EE【09 DE ED 13 31】FF FC FF FF

参数说明：无

一旦解除系统配置锁定，设备可以重新接收外部串口命令来修改配置参数。用户可以直接在 PC 的指令助手中进行配置。

## 2.67 告警事件触发

指令格式：EE【B1 50 Screen\_idControl\_idValue (SecMinHourDayWeekMonYear)】FF FC FF FF

参数说明：Screen\_id(2 个字节)：画面编号

Control\_id(2 个字节)：数据记录控件编号

Value(2 个字节)：告警事件 ID 号

若客户选中了指令时间的话，指令会添加以下参数：

Sec(1 个字节)：秒，BCD 码表示

Min(1 个字节)：分，BCD 码表示

Hour(1 个字节)：时，BCD 码表示

Day(1 个字节)：日，BCD 码表示

Week(1 个字节)：星期，BCD 码表示

Mon(1 个字节)：月，BCD 码表示

Year(1 个字节)：年，BCD 码表示

该指令是通过给屏幕传送相应的事件 ID，使得相应的事件 ID 号转换成用户设置的告警事件输出。

## 2.68 告警事件解除

指令格式：EE【B1 51 Screen\_id Control\_idValue (Sec Min Hour Day Week Mon Year)】FF FC FF FF

参数说明：Screen\_id(2 个字节)：画面编号

Control\_id(2 个字节)：数据记录控件编号

Value(2 个字节)：告警事件 ID 号

若客户选中了指令时间的话，指令会添加以下参数：

Sec(1 个字节)：秒，BCD 码表示

Min(1 个字节)：分，BCD 码表示

Hour(1 个字节)：时，BCD 码表示

Day(1 个字节)：日，BCD 码表示

Week(1 个字节)：星期，BCD 码表示

Mon(1 个字节): 月, BCD 码表示

Year(1 个字节): 年, BCD 码表示

该指令是通过给屏幕传送相应的事件 ID, 使得相应的事件 ID 号转换成用户设置的解除告警事件输出。

## 2.69 数据记录控件-添加常规记录

指令格式: EE【B1 52Screen\_id Control\_id String】FF FC FF FF

参数说明: Screen\_id(2 个字节): 画面编号

Control\_id(2 个字节): 数据记录控件编号

String (不定长): 用户添加的字符串内容

该指令用于在数据记录控件中添加用户的数据, 用于显示一些重要的参数以及相应的数据表格, 用户用该指令可以实现添加相应用户数据到数据记录控件中去。

## 2.70 数据记录控件-清除记录数据

指令格式: EE【B1 53Screen\_idControl\_id】FF FC FF FF

参数说明: Screen\_id(2 个字节): 画面编号

Control\_id(2 个字节): 数据记录控件编号

该指令用于数据记录控件中, 清除全部记录数据。

## 2.71 数据记录控件-设置记录显示偏移

指令格式: EE【B1 54 Screen\_id Control\_id Offset】FF FC FF FF

参数说明: Screen\_id(2 个字节): 画面编号

Control\_id(2 个字节): 数据记录控件编号

Offset (2 个字节): 所在的行号

该指令是用于设置滚动条的位置, 即记录控件首行对应的行号。

## 2.72 数据记录控件-获取当前记录数目

指令格式: EE【B1 55 Screen\_id Control\_id】FF FC FF FF

参数说明: Screen\_id(2 个字节): 画面编号

Control\_id(2 个字节): 数据记录控件编号

该指令主要用于数据记录控件获取当前数据的记录数目

返回指令: EE【B1 55 Screen\_id Control\_id 1D Count】FF FC FF FF

返回参数: Screen\_id(2 个字节): 画面编号

Control\_id(2 个字节): 数据记录控件编号

1D (1 个字节): 0x1D, 固定值, 表示记录控件

Count (2 个字节): 当前记录数目

## 2.73 数据记录控件-读取某一行记录

指令格式: EE【B1 56 Screen\_id Control\_id Position】FF FC FF FF

参数说明: Screen\_id(2 个字节): 画面编号

Control\_id(2 个字节): 数据记录控件编号

Position (2 个字节): 某一行的行号

该指令用于数据记录控件读取某一行的记录, 会返回相应的数据指令。

返回指令: EE【B1 56 Screen\_idControl\_id 1D String】FF FC FF FF

返回参数: Screen\_id(2 个字节): 画面编号



Control\_id(2 个字节): 数据记录控件编号

1D (1 个字节): 0x1D, 固定码

String (不定长): 用户添加的字符串内容

## 2.74 数据记录控件-修改常规记录

指令格式: EE 【B1 57 Screen\_id Control\_id Position String】 FF FC FF FF

参数说明: Screen\_id(2 个字节): 画面编号

Control\_id(2 个字节): 数据记录控件编号

Position (2 个字节): 某一行的行号

String (不定长): 用户添加的字符串内容

该指令用于数据记录控件修改常规记录, 这个可以针对某一行进行相应的修改。

## 2.75 数据记录控件-删除某一行记录

指令格式: EE 【B1 58 Screen\_id Control\_id Position】 FF FC FF FF

参数说明: Screen\_id(2 个字节): 画面编号

Control\_id(2 个字节): 数据记录控件编号

Position (2 个字节): 某一行的行号

该指令用于数据记录控件中进行删除某一行记录, 可以方便用户删除某一行的记录。

## 2.76 数据记录控件-插入常规记录

指令格式: EE 【B1 59 Screen\_id Control\_id PositionString】 FF FC FF FF

参数说明: Screen\_id(2 个字节): 画面编号

Control\_id(2 个字节): 数据记录控件编号

Position (2 个字节): 某一行的行号

String (不定长): 用户添加的字符串内容

该指令用于数据记录控件中在某一行位置插入相应的常规记录, 此位置之后的记录整体后移。

## 2.77 数据记录控件-选中某一行数据记录

指令格式: EE 【B1 5A Screen\_id Control\_id Offset】 FF FC FF FF

参数说明: Screen\_id(2 个字节): 画面编号

Control\_id(2 个字节): 数据记录控件编号

Offset (2 个字节): 所在的行号

该指令用于数据记录控件中选中某一行的数据记录, 选中后会显示相应的选中颜色。

注意: 用户使用该指令须在数据记录控件有记录的行里面选择并且允许选择选项打开。

## 2.78 数据记录控件-一次添加多条记录

指令格式: EE 【B1 5B Screen\_id Control\_id Count Record0\_size Record0 Record1\_size Record1...】 FF FC FF FF

参数说明: Screen\_id(2 个字节): 画面编号

Control\_id(2 个字节): 数据记录控件编号

Count (2 个字节): 添加的记录条数

Recordx\_size(2 个字节): 每一条记录对应的大小, “;” 也算一个字节

Recordx (不定长): 每一条记录相应的数据, 以 Recordx\_size 长度为准

该指令主要可以实现数据记录控件中一次可以添加多条数据记录。

## 2.79 数据记录控件-SD 卡导出指令（CSV 格式）

指令格式：EE【B1 5C Screen\_id Control\_id】FF FC FF FF

参数说明：Screen\_id(2 个字节)：画面编号

Control\_id(2 个字节)：数据记录控件编号

该指令主要用于将数据记录控件以 CSV 格式导出到 SD 卡。

返回指令：EE【B1 5C Screen\_id Control\_id Status】FF FC FF FF

返回参数：Screen\_id(2 个字节)：画面编号

Control\_id(2 个字节)：数据记录控件编号

Status (1 个字节)：0x00 表示成功，其他表示失败

## 2.80 历史曲线设置采样值

指令格式：EE【B1 60 Screen\_id Control\_id Value】FF FC FF FF

参数说明：Screen\_id(2 个字节)：画面编号

Control\_id(2 个字节)：文本控件编号

Value (不定长):数据类型有: UINT8、INT8、UINT16、INT16、UINT32、INT32、FLOAT,每个通道的数据按照设定的数据类型对应,如果有两个通道,数据类型是 UINT8,相应添加两个数据类型为 UINT8 的数据

该指令用于添加历史曲线的采样值数据,相应的数据添加到指令里面,指令发给屏幕,屏幕按照相应的数据值进行绘制曲线。

## 2.81 历史曲线禁止/使能采样

指令格式：EE【B1 61 Screen\_id Control\_id Enable】FF FC FF FF

参数说明：Screen\_id(2 个字节)：画面编号

Control\_id(2 个字节)：文本控件编号

Enable(1 个字节)：禁止/使能位

0x00：禁止；

0x01：使能；

该指令主要用于对历史曲线进行禁止/使能采样。如果用户需要对某一段数据不进行输入的话,可以使用禁止采样指令实现;如果用户需要重新赋予采样功能的话,需要用使能采样指令实现。

## 2.82 历史曲线隐藏/显示通道

指令格式：EE【B1 62 Screen\_id Control\_idChannel Show】FF FC FF FF

参数说明：Screen\_id(2 个字节)：画面编号

Control\_id(2 个字节)：文本控件编号

Channel(1 个字节)：通道 ID

Show(1 个字节)：显示禁止/使能位

0x00：隐藏；

0x01：显示；

该指令主要用于对历史曲线的某些通道进行显示/隐藏的操作。在用户界面中存在多种通道曲线,用户需要对某个通道进行详细查看、分析的话,可以使用指令隐藏相应通道的曲线,在相应情况下,也可以用指令来进行显示通道曲线。

## 2.83 历史曲线设置时间长度（即采样点数）

指令格式：EE【B1 63 Screen\_id Control\_id00 Sample\_Count】FF FC FF FF



参数说明: Screen\_id(2 个字节): 画面编号

Control\_id(2 个字节): 文本控件编号

00 (1 个字节): 控制类型 ID 号, 0x00: 设置时间宽度

Sample\_Count (2 个字节): 屏显示的采样点数

该指令主要用于设置历史曲线的时间长度, 即可以设置历史曲线的采样点数。

## 2.84 历史曲线缩放到全屏

指令格式: EE 【B1 63 Screen\_id Control\_id 01】 FF FC FF FF

参数说明: Screen\_id(2 个字节): 画面编号

Control\_id(2 个字节): 文本控件编号

01 (1 个字节): 控制类型 ID 号, 0x01: 缩放到全屏

该指令主要用于设置历史曲线的缩放, 即可以设置历史曲线缩放比例达到全屏。

## 2.85 历史曲线设置缩放比例系数

指令格式: EE 【B1 63 Screen\_id Control\_id 02 ZoomMax\_ZoomMin\_Zoom】 FF FC FF FF

参数说明: Screen\_id(2 个字节): 画面编号

Control\_id(2 个字节): 文本控件编号

02 (1 个字节): 控制类型 ID 号, 0x02: 缩放比例系数

Zoom (2 个字节): 缩放百分比

Max\_Zoom (2 个字节): 缩放最大限制, 一屏最多显示采样点数

Min\_Zoom (2 个字节): 缩放最小限制, 一屏最少显示采样点数

该指令主要用于设置历史曲线按照用户设置的缩放比例系数进行显示, 使得绘制的曲线能够符合用户查看的角度。

## 2.86 历史曲线设置数值显示范围

指令格式: EE 【B1 64 Screen\_id Control\_id Max Min】 FF FC FF FF

参数说明: Screen\_id(2 个字节): 画面编号

Control\_id(2 个字节): 文本控件编号

Max (4 个字节): 最大值

Min (4 个字节): 最小值

该指令主要用于设置历史曲线的最大值和最小值, 即设置历史曲线的显示范围。

## 2.87 历史曲线数据清空

指令格式: EE B1 65 【Screen\_id Control\_id】 FF FC FF FF

参数说明: Screen\_id: 画面 ID (两个字节)

Control\_id: 控件 ID (两个字节)

该指令主要用于清空所有通道的数据, 但控件开启掉电存储后无法清空。

## 2.88 旋转控件设置旋转角度

指令格式: EE 【B1 10 Screen\_id Control\_id Value】 FF FC FF FF

参数说明: Screen\_id(2 个字节): 画面编号

Control\_id(2 个字节): 文本控件编号

Value(2 个字节): 旋转角度, 旋转角度为 0-360 度; 同时旋转控件支持小角度旋转, 粒度 0.1 度,  $Value = \text{实际角度} * 10 + 0x8000$ , 例如 5.3 度设置值 =  $5.3 * 10 + 0x8000$

该指令主要用于设置旋转控件的旋转角度, 用户可用该指令实现指针按照旋转中心旋转设定的角度进行旋转。

## 2.89 多语言切换指令

指令格式: EE 【C1 Lang Check】 FF FC FF FF

参数说明: Lang (1 个字节):

BIT3~BIT0: 标识语言索引, 范围: 0-9

BIT6~BIT4: 保留, 默认0

BIT7: 标识系统键盘语言, 0: 中文, 1: 英文

Check (1 个字节): 用于校验,  $Check = 0xC1 + Lang$ , 取低字节

该指令一般用于对于多语言进行切换的指令, 用户可以用改指令来设置键盘的输出语言是中文还是英文。

## 2.90 获取系统语言索引

指令格式: EE 【C2】 FF FC FF FF

参数说明: 无

该指令用于获取串口屏当前使用的语言索引;

返回指令: EE 【C2 Lang】 FF FC FF FF

返回参数: Lang (1 个字节):

BIT3~BIT0: 标识语言索引, 范围: 0-9

BIT6~BIT4: 保留, 默认0

BIT7: 标识系统键盘语言, 0: 中文, 1: 英文

## 2.91 视频相关指令(仅物联型、M 型适用)

### 2.91.1 添加视频路径

指令格式: EE 【B1 74 Screen\_id Control\_id Videos\_Path】 FF FC FF FF

参数说明: Screen\_id(2 个字节): 画面编号

Control\_id(2 个字节): 视频播放控件编号

Videos\_Path: 视频路径 (可以在指令助手转换)

该指令用于在播放前添加需要播放的视频路径。

注意: 播放 U 盘或 SD 卡里的视频请参考相关的技术文档。

### 2.91.2 播放视频

指令格式: EE 【B1 70 Screen\_id Control\_id Number】 FF FC FF FF

参数说明: Screen\_id(2 个字节): 画面编号

Control\_id(2 个字节): 视频播放控件编号

Number (2 个字节): 视频循环播放的次数, 0 为循环播放

该指令主要用于控制视频播放控件的播放视频, 可以设置相应的视频播放循环次数, 让视频按照相应的设置进行播放。(注: 视频播放控件必须先设置视频路径)

### 2.91.3 视频暂停/恢复

指令格式：EE【B1 72 Screen\_id Control\_id】FF FC FF FF

参数说明：Screen\_id(2 个字节)：画面编号

Control\_id(2 个字节)：视频播放控件编号

### 2.91.4 视频停止

指令格式：EE【B1 71 Screen\_id Control\_id】FF FC FF FF

参数说明：Screen\_id(2 个字节)：画面编号

Control\_id(2 个字节)：视频播放控件编号

## 2.92 音频播放相关指令(仅物联型、F 系列、M 型适用)

### 2.92.1 音频播放

指令格式：EE【94 Sounds\_Path】FF FC FF FF

参数说明：Sounds\_Path: 音频文件路径（参考指令助手）

该指令用于指定播放某路径下的音频文件。

注意：播放 U 盘或 SD 卡里的音频请参考相关的技术文档。

### 2.92.2 停止音频播放

指令格式：EE【95】FF FC FF FF

该指令用于停止播放的音频

### 2.92.3 音频播放指令

指令格式：EE【90 01 sounds\_id time】FF FC FF FF

参数说明：sounds\_id(2 个字节)：音频 ID（只支持数字 ID）

time（1 个字节）：播放次数（0 代表循环播放）

该指令用于播放以数字命名的音频文件。

### 2.92.4 暂停音频播放指令

指令格式：EE【90 02】FF FC FF FF

该指令用于暂停音频文件播放。

### 2.92.5 恢复音频播放指令

指令格式：EE【90 03】FF FC FF FF

该指令用于恢复音频文件播放。

### 2.92.6 停止音频播放指令

指令格式：EE【90 00】FF FC FF FF

该指令用于暂停音频文件播放。

### 2.92.7 音频序列播放（适用于以数字命名的音频）

音频序列播放指令：

指令格式：EE【90 04 00 01 sounds\_id time】FF FC FF FF

参数说明：sounds\_id(2 个字节)：音频 ID（每个 ID 用；隔开）

time（1 个字节）：播放次数（0 代表循环播放）

### 2.92.8 音量调节

指令格式：EE 【93 Value】 FF FC FF FF

参数说明：Value (1 个字节)：音量值，范围为 0-100。

该指令主要用于调节喇叭输出的音量大小。

## 2.93 矩阵键盘控制

指令格式：EE 【79cmd】 FF FC FF FF

参数说明：Cmd(1个字节)：配置参数

BIT0: 1表示矩阵键盘使能，0表示矩阵键盘关闭；

BIT1: 1表示按下键时蜂鸣器自动响，0表示不响；

BIT4~BIT2：矩阵键盘上传方式

000: 表示按下键盘时上传1次坐标

001: 表示键盘被按下直至释放后上传1次坐标

010: 键盘长久被按下时，每100ms上传1次坐标，释放时再上传1次坐标

011: 表示键盘被按下和释放时均上传1次坐标

BIT7-BIT5: 保留

设备支持外接 4\*4 矩阵键盘输入。按下键盘后，键值通过串口上传给主机。

键值编码的上传格式：

按下时上传格式：EE 12 K FF FC FF FF；

释放时上传格式：EE 13 K FF FC FF FF，K 为 1 个字节，代表上传的键值

键盘为 4\*4 矩阵键盘，键值编码范围为 0 到 15，其对应关系如下表 2.3 键码查询表所示。

表 2.3 键码查询表

列行	K6	K7	K8	K9
K1	0	1	2	3
K2	4	5	6	7
K3	8	9	10	11
K4	12	13	14	15

## 2.94 写数据到 FLASH

指令格式：EE 【87 Addr Data0 Data1 Data2...Datan】 FF FC FF FF

参数说明：Addr (4 个字节)：数据写入的起始地址

Datan(1 个字节)：写入的数据

存储空间为 128K 字节，地址范围是 0~0x1FFFF。

写入成功后，设备返回：EE 0C FF FC FF FF。

该指令主要将用户部分的一些数据保存在设备内的 Flash 中，当做 EEPROM 使用。

## 2.95 读取保存在 FLASH 中的数据

指令格式：EE 【88 Addr Length】 FF FC FF FF

参数说明：Addr (4 个字节)：数据读起始地址

Length (2 个字节)：读取数据的长度，单位为字节

返回的数据格式为：EE 0B Data0 Data1 Data2...Datan FF FC FF FF。

该指令主要用于将写入存储器中的数据读出。

## 2.96 将当前画面存储到 TF 卡

指令格式：EE 【34 00】 FF FC FF FF

参数说明：无

该指令支持基本型、商业型、经济型，物联型不支持。

该指令发送后，在 TF 卡中会自动生成相应的.bmp 文件。

如果存储成功，返回格式为：EE 10 FF FC FF FF

如果存储失败，返回格式为：EE 11 FF FC FF FF

## 2.97 保存无线网络设置

指令格式：EE 【D0 A0】 FF FC FF FF

参数说明：无

返回指令：EE D0 A0 WIFI\_SET\_CFG FF FC FF FF

返回参数：WIFI\_SET\_CFG（不定长）：保存无线网络设置

该指令仅支持物联型屏幕。

该指令主要用来保存无线网络设置，将用户的无线网络设置保存起来。无线网络设置包括：禁用、网卡模式、热点模式。

## 2.98 获取无线网络设置

指令格式：EE 【D0 A1】 FF FC FF FF

参数说明：无

返回指令：EE D0 A1 WIFI\_GET\_CFG FF FC FF FF

返回参数：WIFI\_GET\_CFG（不定长）：获取无线网络的设置

该指令仅支持物联型屏幕。

该指令主要用来获取无线网络的相应设置，设置之后，屏幕会返回相应的无线网络的参数指令。

## 2.99 获取网络状态

指令格式：EE 【D0 A2】 FF FC FF FF

参数说明：无

返回指令：EE D0 A2 WIFI\_GET\_STATE FF FC FF FF

返回参数：WIFI\_GET\_STATE（不定长）：获取网络状态

该指令仅支持物联型屏幕。

该指令主要用来获取网络状态，设置之后，网络状态分为无线网络连接、有线网络连接、远程服务器连接、客户连接数。当获取网络状态指令发送后，相应的网络状态会被选中。

## 2.100 搜索无线网络

指令格式：EE 【D0 A3】 FF FC FF FF

参数说明：无

返回指令：EE D0 A3 WIFI\_SCAN\_APLIST FF FC FF FF

返回参数：WIFI\_SCAN\_APLIST（不定长）：搜索无线网络

该指令仅支持物联型屏幕。

该指令主要用来搜索无线网络，搜索无线网络，可以让用户搜索到相应的 WIFI,进行无线的连接。

### 2.101 保存网络设置

指令格式：EE 【D0 A4】 FF FC FF FF

参数说明：无

返回指令：EE D0 A4 NETWORK\_SET\_CFG FF FC FF FF

返回参数：NETWORK\_SET\_CFG（不定长）：保存网络设置

该指令仅支持物联型屏幕。

该指令主要用来保存网络设置，将用户的网络设置保存起来。

### 2.102 获取网络设置

指令格式：EE 【D0 A5】 FF FC FF FF

参数说明：无

返回指令：EE D0 A5 NETWORK\_GET\_CFG FF FC FF FF

返回参数：NETWORK\_GET\_CFG（不定长）：获取网络设置

该指令仅支持物联型屏幕。

该指令主要用来获取网络设置，网络设置包括：IP 地址、子网掩码、默认网关、DNS 服务器。

### 2.103 保存网络服务

指令格式：EE 【D0 A6】 FF FC FF FF

参数说明：无

返回指令：EE D0 A6 NETWORK\_SERVICE\_SET FF FC FF FF

返回参数：NETWORK\_SERVICE\_SET（不定长）：保存网络服务

该指令仅支持物联型屏幕。

该指令主要用来保存网络设置，将用户的网络服务保存起来。网络服务包括：禁用、客户端、服务器。

### 2.104 获取网络服务

指令格式：EE 【D0 A7】 FF FC FF FF

参数说明：无

返回指令：EE D0 A7 NETWORK\_SERVICE\_GET\_CFG FF FC FF FF

返回参数：NETWORK\_SERVICE\_GET\_CFG（不定长）：获取网络服务

该指令仅支持物联型屏幕。

该指令主要用来获取网络服务，网络服务包括：禁用、客户端、服务器。

### 2.105 发送网络数据

指令格式：EE D0 AC Count Strings FF FC FF FF

参数格式：Count(2 个字节)：发送的字节数

Strings（不定长）：用户选择发送的数据，用软件定义的格式写入，例如：12，写入为 31 32

该指令仅支持物联型屏幕。

该指令主要用来发送网络数据，按照相应的 ASCII 码发送。

### 2.106 发送网络数据（十六进制）

指令格式：EE D0 AC Count Strings FF FC FF FF

参数格式：Count(2 个字节)：发送的字节数

Strings (不定长): 用户选择发送的数据, 用相应的格式写入, 例如: 12 13,  
写入为 12 13

该指令仅支持物联型屏幕。

该指令主要用来发送网络数据, 按照相应的格式发送。

### 2.107 保存网络设置

指令格式: EE 【D0 AA】 FF FC FF FF

参数说明: 无

该指令仅支持物联型屏幕。

该指令主要用于保存相应的网络设置。

### 2.108 本地/外部 AV 输入视频

指令格式: EE 【4B Enable】 FF FC FF FF

参数说明: Enable (1 个字节): 本地/外部画面切换标志位

0x00: 切换到本地图片显示

0x01: 切换到 AV 视频输入显示

该指令只适用于基本型、商业型等, 不适用于物联型和 F 系列, 指令主要用于本地图片显示画面和 AV 视频输入显示画面的切换。

显示画面和 AV 视频输入显示画面的切换。

### 2.109 显示系统键盘

指令格式: EE 【86 01 x y type Option max\_len】 FF FC FF FF

参数说明: 01 (1 个字节): 固定值, 显示键盘;

x (2 个字节): 键盘弹出的 x 坐标;

y (2 个字节): 键盘弹出的 y 坐标;

type (1 个字节): 0 小键盘, 1 全键盘;

Option (1 个字节): 0 正常字符, 1 密码, 2 时间设置;

max\_len (1 个字节): 键盘录入字符长度限制;

该指令主要用于显示/隐藏系统键盘;

键盘输入的值会通过串口输出, 指令格式 EE 【86 01 Strings】 FF FC FF FF。

Strings (不定长): 用户输入字符的 ascii 码, 例如键盘输入 123, 此处为  
31 32 33;

### 2.110 隐藏系统键盘

指令格式: EE 【86 00】 FF FC FF FF

参数说明: 无

该指令主要用于隐藏系统键盘;

### 2.111 触摸屏灵敏度调节指令 (仅用于电容屏)

指令格式: EE 【8A 5A A5 XX】 FF FC FF FF

参数说明: XX (1 个字节): XX 取值范围是 0-7, 越高档位越灵敏, 串口屏出厂默认是 0, 调整触摸灵敏度用于支持更厚的盖板。如无必要, 请勿随意调整灵敏度。

该指令用于调节触摸屏的灵敏度, 适用于电容屏。

### 2.112 进入待机状态

指令格式: EE 【60 00】 FF FC FF FF



参数说明：无

该指令主要用于使屏幕进入待机状态，从激活状态变成待机状态；

### 2.113 进入激活状态

指令格式：EE 【60 01】 FF FC FF FF

参数说明：无

该指令主要用于使屏幕进入激活状态，从待机状态变成激活状态；

### 2.114 进入休眠状态

指令格式：EE 【AA 02 AC】 FF FC FF FF

参数说明：无

该指令主要用于使屏幕进入休眠状态，从激活状态进入休眠状态，降低功耗。

注意：该指令仅适用于物联型，F 系列和 M 系列。

### 2.115 退出休眠状态

指令格式：EE 【AA 00 00 00 00 FF FF FF FF 00 00 00 00】 FF FC FF FF

参数说明：无

该指令主要用于使屏幕进入激活状态，从休眠状态进入激活状态。

注意：该指令仅适用于物联型，F 系列和 M 系列。

### 2.116 切换 AV 输入信号

指令格式：EE B1 77 【Screen\_id Control\_id AV\_Serial】 FF FC FF FF

参数说明：Screen\_id：画面 ID（两个字节）

Control\_id：控件 ID（两个字节）

AV\_Serial：输入路数（0/1 第一路输入/第二路输入）

该指令主要用于切换两路 AV 视频输入。

注意：该指令只适用于 M 型。

### 2.117 切换子画面

指令格式：EE B1 0A 【Screen\_id】 FF FC FF FF

参数说明：Screen\_id：画面 ID（两个字节）

该指令主要用于切换子窗口画面，切换至 PNG 全透图片需要使用。

注意：该指令只适用于物联型。

### 2.118 子画面窗口切换上一画面

指令格式：EE B1 80 S 【Screen\_id Control\_id】 FF FC FF FF

参数说明：Screen\_id：画面 ID（两个字节）

Control\_id：控件 ID（两个字节）

该指令主要用于清空所有通道的数据，但开启存储后无法清空。

## 2.119 子画面窗口切换下一画面

指令格式: EE B1 81 【Screen\_id Conrol\_id】 FF FC FF FF

参数说明: Screen\_id : 画面 ID (两个字节)

Conrol\_id: 控件 ID (两个字节)

该指令主要用于清空所有通道的数据, 但开启存储后无法清空。

## 2.120 使能画布

指令格式: EE 1E Enable X Y W H FF FC FF FF

参数说明: Enable : 0/1 使能或失能

X: X 轴起始坐标 (两个字节)

Y: Y 轴起始坐标 (两个字节)

W: 宽度 (两个字节)

H: 高度 (两个字节)

该指令主要用于设置画布的范围大小, 避免基本指令集和组态指令集冲突, 画布的范围显示基本指令集画的指令显示范围, 设置了画布之后, 再发送基本指令集的指令, 组态控件和基本指令集不冲突

## 2.121 基本绘图指令

指令格式: EE B1 10 【Screen\_id Conrol\_id Number Color X0 X1 Y0 Y1】 FF FC FF FF

参数说明: Screen\_id : 画面 ID (两个字节)

Conrol\_id: 控件 ID (两个字节)

Number: 绘图功能码 (01 直线, 02 多点连线, 03 空心矩形, 04 实心矩形, 05 空心椭圆(圆), 06 实心椭圆(圆), 07 图片裁剪)

X0 : 左上 X 轴坐标

X1 : 右下 X 轴坐标

Y0 : 左上 Y 轴坐标

Y1 : 右下 Y 轴坐标

该指令只在使用画布后生效, 用于绘制基本图形。

### 3. 附录 A 基本指令集详述

由于组态指令集可以满足 95% 的工程应用,而且操作简单,大部分配置都在 PC 中完成,因此我们并不推荐用户使用基本指令集。相对组态指令集,基本指令集属于最底层指令,灵活性更强,在组态指令无法完成的情况下,可以使用基本集指令来实现。详细的指令集说明如下列章节所述。

#### 3.1 设置前/背景色

指令格式: 设置前景色: EE 【41 Fcolor】 FF FC FF FF  
 设置背景色: EE 【42 Bcolor】 FF FC FF FF  
 设置前景和背景色: EE 【40 Fcolor Bcolor】 FF FC FF FF

参数说明: Fcolor、Bcolor (2 个字节)分别是前景色和背景色的 RGB 值  
 前景色主要用于指定文本、点、线和圆等显示的颜色,背景色主要用于清屏和文字底色颜色的指定。比如通过设置前/背景色的指令可完成显示图 3-1 的所示内容。

参考程序:

```
{
    SetBcolor(31);          //设置背景色为蓝色,用于指定清屏颜色
    GUI_CleanScreen( );    //清屏
    SetFcolor(65516);       //设置前景色为黄色,用户指定文本显示颜色
    SetBcolor(63488);       //设置背景色为红色,用于指定文本背景色(底色)
    DisText(50, 50, 1, 6, "你好, 色彩! "); //在坐标(50,50)写入字符串,有背景色
    DisText(50, 90, 0, 6, "你好, 串口屏! "); //在坐标(50,90)写入字符串,无背景色
    SetFcolor(2016);        //重新设置前景色为绿色
    GUI_RectangleFill(256, 57, 370, 116); //画实心矩形
}
```

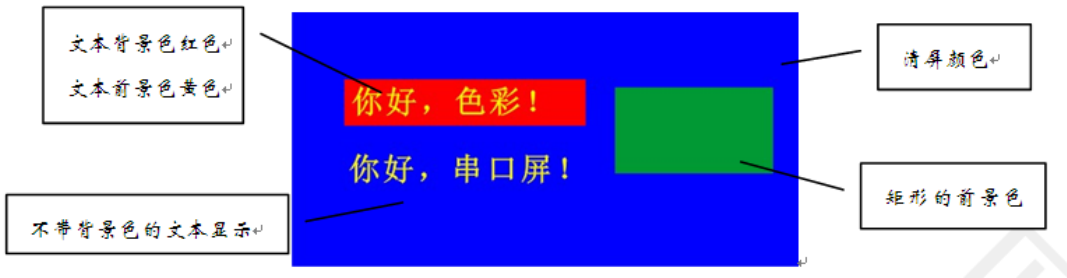


图 3-1 背景前景色说明

#### 3.2 切换画面时自动清除图层

指令格式: EE 【06 Enable】 FF FC FF FF  
 参数说明: Enable (1 个字节): 清除标志位  
 0x00: 不清除  
 0x01: 自动清楚

该指令用于实现切换画面时自动清除图层的操作,即切换画面时自动清除图层中的非控件图形(点、直线、矩形、圆、椭圆、文字、图片)。

### 3.3 设置文字行列间距

指令格式: EE 【43 Y\_W X\_W】 FF FC FF FF

参数说明: Y\_W(1 个字节) 是以点为单位的行间距, 取值范围 00~3F;

X\_W(1 个字节) 是以点为单位的列间距, 取值范围 00~3F。

该指令用于设置文字之间的行列距。如果文本只有一行, 行间距则为 0。例如在屏幕上显示

2 行 32\*32 的字符串, 设置行间距 24, 列间距 16, 程序如下所示。

参考程序:

```
{
    SetFcolor(65504);    //设置前景色为黄色, 指定文本显示颜色
    SetBcolor(63488);    //设置背景色为红色, 指定文本底色颜色
    SetTextSpace(24,16); //设置文本行间距为 24, 列间距为 16
    DisText(50, 51, 1, 7, "广州大彩科技工业串口屏"); //显示文本字符串
}
```

### 3.4 设置文本框

指令格式: EE 【45 Enable Width Hight】 FF FC FF FF

参数说明: Enable(1 个字节): 打开/关闭文本框限制使能

Width (2 字节): 文本框宽度;

Hight (2 字节): 文本框的高度。

设置文本框后, 文字将在限定框内自动换行显示。

### 3.5 设置图片过滤色

指令格式: EE 【44 FillColor】 FF FC FF FF

参数说明: FillColor (2 字节): 过滤色 RGB 值;

设定过滤色后, 当图片某一像素值正好与过滤色值相同, 该点就会被屏蔽, 无法在屏幕上显示。设置过滤色前后对比如图 3-2 所示。

参考程序:

```
{
    DisArea_Image(0, 0, 0, 0);    // (0,0)处显示草原背景图
    DisArea_Image(61,130, 1, 0);   // (61,130)处显示未过滤的蝴蝶图案
    SetFilterColor(65535);         //设置白色为过滤色, RGB 值为 65535
    DisArea_Image(258,68, 2, 1);   // (258,68)处显示过滤后的蝴蝶图案
}
```



图 3-2 设置过滤色前后对比图

### 3.6 文本显示

指令格式: EE 【20 X Y Back Font Strings】 FF FC FF FF

参数说明: X (2 字节): 以点为单位的 X 轴坐标值;

Y (2 字节):以点为单位的 Y 轴坐标值;

Back (1 个字节): 背景色使能

0x01:背景色显示 0x00: 背景色不显示

Font (字库编码,1 个字节)

0x00: 8x12 点阵 (ASCII)0x01: 8x16 点阵 (ASCII)

0x02: 12x24 点阵 (ASCII)0x03: 16x32 点阵 (ASCII)

0x04: 12 x 12 点阵 (GBK)0x05: 16 x 16 点阵 (GBK)

0x06: 24 x 24 点阵 (GBK)

0x07: 32 x 32 点阵 (GB2312)

0x08: 32 x 64 点阵 (ASCII)

0x09: 64 x 64 点阵 (GB2312)

Strings: 用户写入的字符串, 高字节在前。

该指令用于实现在屏幕任意位置显示指定的文本。**GBK** 包含了汉字及日韩等常用字库;  
**GB2312** 只包含汉字和字符;**ASCII** 不能显示汉字。在实际操作中, 用户确定文本的前景色、  
背景色、字库编码后, 可以连续写入汉字或字符串, 设备将会自动换行及中英文匹配显示。  
文本显示效果图 3-3 所示。

参考程序:

```
{
    SetFcolor(65504);    //设置文字前景色, 黄色
    DisText(46, 21, 0, 7, "工业串口屏 LCM 32*32"); //坐标(46,21)处显示文字, 字库为 7 号字体
}
```



图 3-3 文本显示效果图

### 3.7 光标显示

指令格式: EE 【21 Enable X Y Width Hight】 FF FC FF FF

参数说明: Enable(1 个字节) : 光标使能信号

0x00: 光标关闭, 0x01: 光标开启

X (2 字节):以点为单位的 X 轴坐标值

Y (2 字节):以点为单位的 Y 轴坐标值

Width (1 个字节) : 光标宽度

Hight (1 个字节) : 光标高度

该指令主要用来控制光标的闪烁和关闭。例如用户在 24\*24 的汉字尾缀显示宽度 16、高度 8 的光标, 效果图如图 3-4 所示。

程序参考代码:

```
{
    SetBcolor(31);    // 设置蓝色背景色
    GUI_CleanScreen( );    // 背景清屏蓝色
    DisCursor(1,359,40,16,8); //光标闪烁使能, 在(359,40)处显示宽度 16 高度 8 的光标
}
```

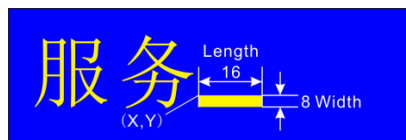


图 3-4 光标参数说明

### 3.8 全屏图片显示

指令格式: EE 【31 Image\_ID MaskEn】 FF FC FF FF

参数说明: Image\_ID (2 个字节): 图片编号

MaskEn (1 个字节): 过滤色使能

0x00:颜色不过滤; 0x01 执行颜色过滤

该指令主要显示全屏图片, 起始坐标固定 (0,0) 处, 被过滤的颜色取决于过滤色设置。若图片尺寸小于全屏大小, 执行该指令时候, 只显示实际大小的图片。

### 3.9 区域图片显示

指令格式: EE 【32 X Y Image\_ID MaskEn】 FF FC FF FF

参数说明: X (2 字节): 以点为单位的 X 轴坐标值

Y (2 字节): 以点为单位的 Y 轴坐标值

Image\_ID(2 个字节): 图片编号

MaskEn (1 个字节): 过滤色使能

0x00:颜色不过滤; 0x01 执行颜色过滤

该指令用于实现任意位置图片显示。用户需要注意起始坐标和图片长宽, 防止显示画面超出屏幕范围。例如用户要在屏幕 (100,50) 处显示一张 ID 为 2 的图片, 如图 3-5 所示, 参考程序如下所示。



图 3-5 区域图片显示

程序参考代码:

```
{
    DisFull_Image(0,0);           // 整屏图片显示
    DisArea_Image(100,50, 2, 0);  // 区域图片显示, 坐标 (100,50) 处显示 ID=2 的图片
}
```

### 3.10 图片剪切

指令格式: EE 【33 X Y Image\_ID Image\_X Image\_Y Image\_W Image\_H MaskEn】  
FF FC FF FF

参数说明: X (2 字节): 以点为单位的 X 轴坐标值

Y (2 字节): 以点为单位的 Y 轴坐标值

Image\_ID (2 字节): 要剪切的图片编号  
 Image\_X (2 字节): 被剪切的图片起点 X 坐标  
 Image\_Y (2 字节): 被剪切的图片起点 Y 坐标  
 Image\_W (2 字节): 剪切的宽度  
 Image\_H (2 字节): 剪切的高度  
 MaskEn (1 个字节) 0x00:颜色不过滤 ;0x01 执行颜色过滤  
 备注: 被过滤的颜色取决于过滤色的设置

该指令用于实现在屏幕任意坐标处显示被剪切过来的图片。用户可以对储存在 Flash 中的任意一张图片进行局部裁剪, 剪切效果如图 3-6 所示。

### 3.11 动画显示

指令格式: EE 【80 X Y FlashImage\_ID Enable PlayNum 】FF FC FF FF

参数说明: X (2 字节): 以点为单位的 X 轴坐标值

Y (2 字节): 以点为单位的 Y 轴坐标值

FlashImage\_ID (2 字节): 动画编号

Enable(1 个字节) : 使能信号

0x00: 关闭动画播放; 0x01: 开启动画播放

PlayNum(1 个字节)

0x00: 重复播放; 0x01~0xFF: 播放指定次数

播放停止后, 设备返回 EE 02 FF FC FF FF 表示动画播放结束。

该指令用于实现任意位置 gif 动画的显示。动画只支持 gif 格式, 不支持同一画面两个以上的 gif 动画同时播放。如果期望一个画面同时播放多个 gif 动画, 以及动画暂停、停止、播放上/下帧等功能, 请使用动画控件指令, 详情见组态控制指令表。图片动画显示效果如图 3-6 所示。

程序参考代码:

```
{
    DisFull_Image(0, 0);           // 全屏显示小孩图片, 编号为 0
    DisFlashImage(330,5, 1,1,0);   // 坐标(330,5)处插入奶牛 Flash 动画,重复播放
    DisCut_Image(343,137,0,95,30,92,116,0); // 从图片 0 坐标(95,30)处剪切大小 92 x116
                                         // 的图片, 放置在屏幕(343,137)处显示
}
```



图 3-6 图片动画显示效果图

### 3.12 前景色画点

指令格式: EE 【 50 X Y 】FF FC FF FF

参数说明: X (2 字节): 以点为单位的 X 轴坐标值

Y (2 字节): 以点为单位的 Y 轴坐标值



该指令主要实现在屏幕的任意位置画点，点的颜色值取决前景色的设置。

### 3.13 背景色画点（删除点）

指令格式：EE【 58 X Y 】FF FC FF FF

参数说明：X (2 字节)：以点为单位的 X 轴坐标值

Y (2 字节)：以点为单位的 Y 轴坐标值

该指令主要实现在屏幕的任意位置画点，点的颜色值取决背景色的设置。背景色画点通常配合前景色画点使用，可以用删除前景色画的点。

如图 3-7 所示，用户可以通过前景色画点在黑色背景图上显示一个黄色五角星图案，若要修改或删除此图案，可以通过调用背景色画点指令去覆盖同一坐标处前景色写入的点。当然，若用户对数据更新速度要求不太高，也可以先区域清屏，然后重新再绘制。

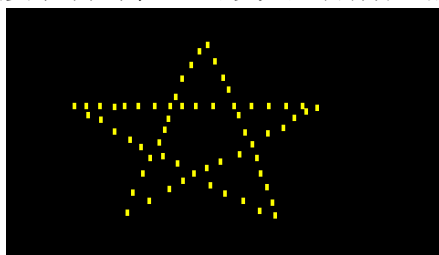


图 3-7 前景/背景色画点

### 3.14 画线

指令格式：EE【 51 X<sub>0</sub> Y<sub>0</sub> X<sub>1</sub> Y<sub>1</sub>】FF FC FF FF

参数说明：X<sub>0</sub> (2 字节)：以点为单位的直线X轴起点坐标值

Y<sub>0</sub> (2 字节)：以点为单位的直线Y轴起点坐标值

X<sub>1</sub> (2 字节)：以点为单位的直线X轴终点坐标值

Y<sub>1</sub> (2 字节)：以点为单位的直线Y轴终点坐标值

该指令主要实现在屏幕的任意两点之间画线，线的颜色值取决前景色的设置。参数说明如图 3-8 所示。例如通过调用画线指定实现一个简易表格，实际显示效果如图 3-8 所示。

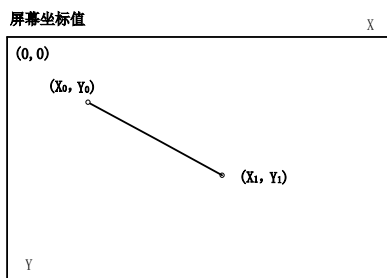


图 3-8 画线参数说明

### 3.15 将等间隔 X 坐标用前景色连接

指令格式：EE【 59 X<sub>0</sub> Xspace Y<sub>0</sub> Y<sub>1</sub> Y<sub>2</sub> ... Y<sub>n</sub>】FF FC FF FF

参数说明：X<sub>0</sub> (2 字节)：以点为单位的 X 轴坐标值

Xspace(2 字节)：以点为单位的 X 轴相邻两点固定的间隔值

Y<sub>n</sub> (2 字节)：以点为单位的 Y 轴坐标值

该指令主要实现快速绘制折线。由于 X 轴前后两点的距离都是固定的 Xspace，所以从第 2 个点开始指令参数都不需要 X 坐标值。对比前景/背景色绘制折线，速度提高了一倍。

### 3.16 按照坐标偏移量用前景色连线

指令格式: EE 【 75 X<sub>0</sub> Y<sub>0</sub> X<sub>10</sub> Y<sub>10</sub> X<sub>20</sub> Y<sub>20</sub> ... X<sub>no</sub> Y<sub>no</sub> 】 FF FC FF FF

参数说明: X<sub>0</sub> (2 个字节): 以点为单位的 X 轴坐标值

Y<sub>0</sub> (2 个字节): 以点为单位的 Y 轴坐标值

X<sub>no</sub> (1 个字节) : 以点为单位的 X 轴偏移量

Y<sub>no</sub> (1 个字节) : 以点为单位的 Y 轴偏移量

(X,Y)为第一点的绝对坐标, 后面的每一个点分别由前一个点的绝对坐标加上当前偏移量组成。偏移量的最高位为符号位, 0 代表正偏移量, 1 代表负偏移, 最大偏移量值为正负 127 个点。

该指令主要利用坐标偏移量将原来每个坐标的 4 个字节变为 2 个字节, 减少一半的指令参数, 达到绘制折线速度提高一倍。线的颜色由前景色设置决定。

### 3.17 将指定的坐标点用前景色连接

指令格式: EE 【68 X<sub>0</sub> Y<sub>0</sub> X<sub>1</sub> Y<sub>1</sub> ... X<sub>n</sub> Y<sub>n</sub>】 FF FC FF FF

参数说明: X<sub>n</sub> (2 字节): 以点为单位的直线X轴起点坐标值

Y<sub>n</sub> (2 字节): 以点为单位的直线Y轴起点坐标值

该指令主要实现用前景色将指定的多个坐标点连接起来。例如要实现图 3-9 所示的折线和六边形形状, 程序如下所示。

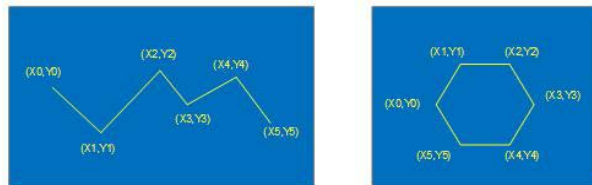


图 3-9 指定坐标点用前景色连接效果图

程序参考代码:

```
{
    SetFcolor(65523);           //设置线的前景色为黄色
    GUI_FcolorConDots ( x0,y0,x1,y1,x2,y2,x3,y3,x4,y4,x5,y5); // 绘制(x0,y0)到(x5,y5)的折线
    GUI_FcolorConDots (x0,y0,x1,y1,x2,y2,x3,y3,x4,y4,x5,y5,x0,y0); // 绘制六边形, 首尾相连
}
```

### 3.18 将指定的坐标点用背景色连接

指令格式: EE 【69 X<sub>0</sub> Y<sub>0</sub> X<sub>1</sub> Y<sub>1</sub> ... X<sub>n</sub> Y<sub>n</sub>】 FF FC FF FF

参数说明: X<sub>n</sub> (2 字节): 以点为单位的直线X轴起点坐标值

Y<sub>n</sub> (2 字节): 以点为单位的直线Y轴起点坐标值

该指令主要实现用背景色将指定的多个坐标点连接起来。背景色绘制折线通常配合前景色绘制折线使用, 可以用删除前景色绘制的折线。

### 3.19 按照坐标偏移量用背景色连线

指令格式: EE 【 76 X<sub>0</sub> Y<sub>0</sub> X<sub>10</sub> Y<sub>10</sub> X<sub>20</sub> Y<sub>20</sub> ... X<sub>no</sub> Y<sub>no</sub> 】 FF FC FF FF

参数说明: X<sub>0</sub> (2 个字节): 以点为单位的 X 轴坐标值

Y<sub>0</sub> (2 个字节): 以点为单位的 Y 轴坐标值

X<sub>no</sub> (1 个字节) : 以点为单位的 X 轴偏移量

Y<sub>no</sub> (1 个字节) : 以点为单位的 Y 轴偏移量

(X,Y)为第一点的绝对坐标，后面的每一个点分别由前一个点的绝对坐标加上当前偏移量组成。偏移量的最高位为符号位，0 代表正偏移量，1 代表负偏移，最大偏移量值为正负 127 个点。

该指令主要利用坐标偏移量将原来每个坐标的 4 个字节变为 2 个字节，减少一半的指令参数，达到绘制折线速度提高一倍。线的颜色由背景色设置决定。

### 3.20 画空心圆

指令格式：EE 【 52 X<sub>0</sub> Y<sub>0</sub> R 】FF FC FF FF

参数说明：X<sub>0</sub> (2 字节)：以点为单位的圆心X坐标值

Y<sub>0</sub> (2 字节)：以点为单位的圆心Y坐标值

R (2 字节)：空心圆的半径

该指令用于实现在指定的坐标处画一个半径 R 空心圆，圆的线条颜色取决前景色的设置。参数说明如图 3-10 所示。

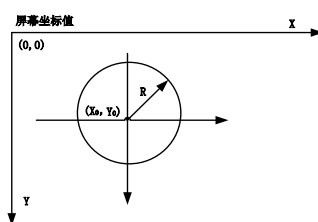


图 3-10 画空心圆参数说明

### 3.21 画实心圆

指令格式：EE 【 53 X<sub>0</sub> Y<sub>0</sub> R 】FF FC FF FF

参数说明：X<sub>0</sub> (2 字节)：以点为单位的圆心X坐标值

Y<sub>0</sub> (2 字节)：以点为单位的圆心Y坐标值

R (2 字节)：实心圆的半径

该指令用于实现在指定的坐标处画一个半径 R 实心圆，圆内填充色取决前景色的设置。参数说明与图 3-10 相同。

### 3.22 画圆弧

指令格式：EE 【 67 X<sub>0</sub> Y<sub>0</sub> R SA EA】FF FC FF FF

参数说明：X<sub>0</sub> (2 字节)：以点为单位的圆心X坐标值

Y<sub>0</sub> (2 字节)：以点为单位的圆心Y坐标值

R (2 字节)：圆的半径

SA(2 字节)：起始角度

EA(2 字节)：结束角度

该指令用于实现在指定的坐标处画一个半径 R 的圆弧，弧线颜色取决前景色的设置。钟表 3 点方向为起始角度 0 度，顺时针方向角度依次增加，参考坐标如图 3-11 所示。

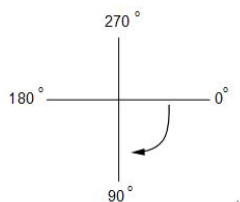


图 3-11 圆弧起始角度参考图

### 3.23 画空心矩形

指令格式: EE 【 54 X<sub>0</sub> Y<sub>0</sub> X<sub>1</sub> Y<sub>1</sub>】 FF FC FF FF

参数说明: X<sub>0</sub> (2 字节):以点为单位的空心矩形左上角X坐标值

Y<sub>0</sub> (2 字节):以点为单位的空心矩形左上角Y坐标值

X<sub>1</sub> (2 字节):以点为单位的空心矩形右下角X坐标值

Y<sub>1</sub> (2 字节):以点为单位的空心矩形右下角Y坐标值

该指令用于实现在屏幕任意位置画一个空心矩形,矩形边框颜色取决前景色的设置。参数定义如图 3-12 所示。

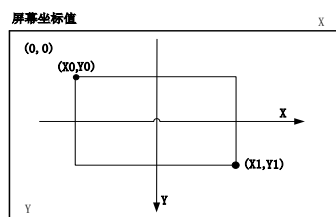


图 3-12 画空心矩形参数说明

### 3.24 画实心矩形/局部清屏

指令格式: EE 【 55 X<sub>0</sub> Y<sub>0</sub> X<sub>1</sub> Y<sub>1</sub>】 FF FC FF FF

参数说明: X<sub>0</sub> (2 字节):以点为单位的实心矩形左上角X坐标值

Y<sub>0</sub> (2 字节):以点为单位的实心矩形左上角Y坐标值

X<sub>1</sub> (2 字节):以点为单位的实心矩形右下角X坐标值

Y<sub>1</sub> (2 字节):以点为单位的实心矩形右下角Y坐标值

该指令用于实现在屏幕任意位置画一个实心矩形,矩形填充色取决前景色的设置。参数定义与图 3-12 相同。该功能还可以作为局部清屏使用。

### 3.25 画空心椭圆

指令格式: EE 【 56 X<sub>0</sub> Y<sub>0</sub> X<sub>1</sub> Y<sub>1</sub>】 FF FC FF FF

参数说明: X<sub>0</sub> (2 字节):以点为单位的空心椭圆最左端X坐标值

Y<sub>0</sub> (2 字节):以点为单位的空心椭圆最上端Y坐标值

X<sub>1</sub> (2 字节):以点为单位的空心椭圆最右端X坐标值

Y<sub>1</sub> (2 字节):以点为单位的空心椭圆最下端Y坐标值

该指令用于实现在屏幕任意位置画一个空心椭圆,椭圆边框颜色取决前景色的设置。参数定义说明如图 3-13 所示。

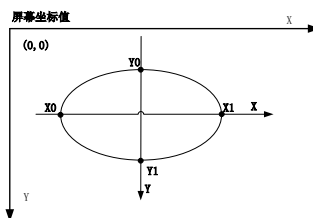


图 3-13 画空心椭圆参数说明

### 3.26 画实心椭圆

指令格式: EE 【 57 X<sub>0</sub> Y<sub>0</sub> X<sub>1</sub> Y<sub>1</sub>】 FF FC FF FF

参数说明:  $X_0$  (2 字节):以点为单位的实心椭圆最左端X坐标值

$Y_0$  (2 字节):以点为单位的实心椭圆最上端Y坐标值

$X_1$  (2 字节):以点为单位的实心椭圆最右端X坐标值

$Y_1$  (2 字节):以点为单位的实心椭圆最下端Y坐标值

该指令用于实现在屏幕任意位置画一个实心椭圆,椭圆填充色取决前景色的设置。

### 3.27 清除图层

指令格式: EE【05 Layer】FF FC FF FF

参数说明: Layer(1 个字节): 清除的图层

该指令主要用于清除指定的图层。

### 3.28 截取当前屏幕并保存在 FLASH 中

指令格式: EE【0x46 Image\_ID】FF FC FF FF

参数说明: Image\_ID (1 个字节): 用户自定义保存在存储器中的画面编号

该指令支持基本型、商业型、经济型, 物联型不支持。

该指令主要用于实现将当前屏幕显示内容保存在 Flash 中。Image\_ID 值可以任意设置, 与工程图片 ID 值不冲突。

### 3.29 显示保存在 FLASH 中的截取画面

指令格式: EE【0x47 Image\_ID】FF FC FF FF

参数说明: Image\_ID (1 个字节)

该指令支持基本型、商业型、经济型, 物联型不支持。

该指令主要用于显示保存在设备 FLASH 中的截取画面。

### 3.30 RTC 模式设置

指令格式: EE【85 Cmd DisMode TextMode Color Xpoint Ypoint 】FF FC FF FF

参数说明: Cmd (1 个字节): 参数配置

BIT0: 使能信号

0: RTC 关闭 1: RTC 开启

BIT7-BIT1: 保留

DisMode(1 个字节): 显示模式

0x00: 格式 HH:MM:SS

0x01: 格式 20XX-MM-DD HH:MM:SS

Font (1 个字节): 字体选择

0x00: 8x12 点阵 (ASCII) 0x01: 8x16 点阵 (ASCII)

0x02: 12x24 点阵 (ASCII) 0x03: 16x32 点阵 (ASCII)

0x04 12 x 12 点阵 (GBK) 0x05: 16 x 16 点阵 (GBK)

0x06: 24 x 24 点阵 (GBK) 0x07: 32 x 32 点阵 (GB2312)

0x08: 32 x 64 点阵 (ASCII) 0x09: 64 x 64 点阵 (GB2312)

Color (2 个字节): 显示颜色

X (2 个字节): 以点为单位的 X 轴坐标值

Y (2 个字节): 以点为单位的 Y 轴坐标值

该指令主要用于 RTC 显示参数的设定。通过设定对应的参数实现不同时钟格式、字体和位置的显示。RTC 相关设置指令建议用户直接采用上位机 VisualTFT 进行设置, 设置参

考界面如图 3-14 所示。

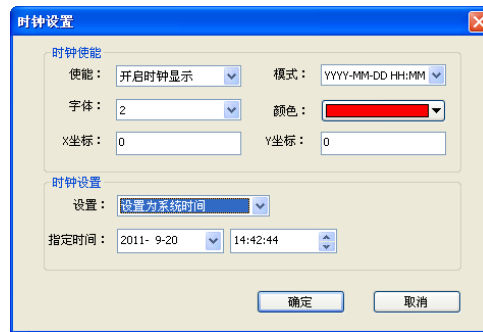


图 3-14RTC 设置参考图

## 4. 声明与服务

感谢您选用大彩系列产品，若您对文档有什么异议或疑问，欢迎随时与我们取得联系。  
电话：020-82186683-601， Email: [hmi@gz-dc.com](mailto:hmi@gz-dc.com)。当然若文档有什么错误或误解之处，欢迎给我们提出批评和建议，我们将及时纠正和改进。

