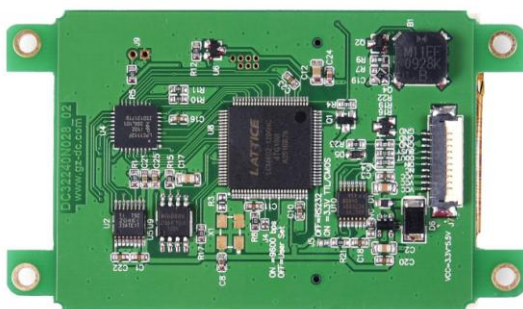


# 小尺寸 Nano 型串口屏数据手册

版本：V3.0



版本	描述	日期	审查
V0.9	编著文档	2012-05-18	李 勇
V1.99	完善和校准内容(初稿)	2012-12-15	李 勇
V2.0	修正部分参数，增加部分指令	2013-10-25	李 勇
V3.0	去掉 4.3/7.0 尺寸	2014-06-20	李 勇

## 小尺寸 Nano 型串口屏应用环境

- ☞ 全面替换 LED 数码管、12864 黑白屏，产品升级为高端 4.3-10.1 真彩屏显示；
- ☞ 支持全屏触控、替换机械按键，增强用户体验和产品档次；
- ☞ 工业标准生产、售价低于用户自主设计，满足大客户苛刻成本需求；
- ☞ 无需大肆修改 PCB，只需添加 1 个 232 或 TTL 串口接口电路；
- ☞ 所有文本、图片、曲线显示都只需 1 个串口指令；
- ☞ 任何 8051、AVR、PIC 和 M3 等单片机都可以驱动，不需修改核心程序；
- ☞ 强大的上位机 VisualTFT 编辑环境；
- ☞ 任何初学者 30 分钟学会开发，3 天内完成产品的程序升级；
- ☞ 按需定制指令、修改 PCB 尺寸图、添加 WIFI、用户 MCU 等；



## Nano 型串口屏架构图

系统处理器采用 Cortex-M0+高速 FPGA 双核设计，ARM 主要进行协议解析和图片下载，FPGA 主要实现 Nandflash 的图片读取和 TFT 控制显示。内部结构如图 A 所示。

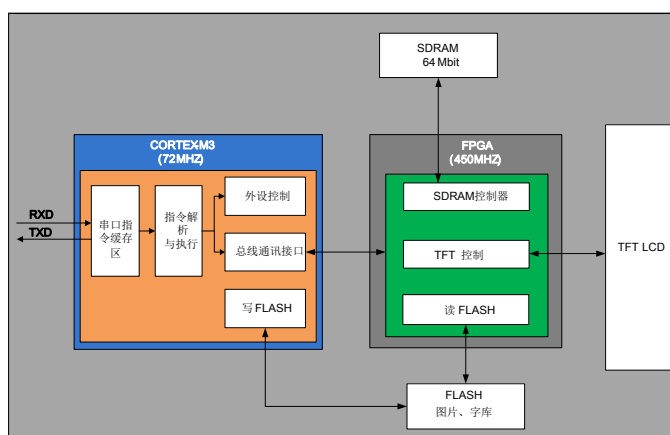


图 A Nano 串口屏内部结构图

系统在无指令接收的状态下，M3 处于空闲状态，所有大负荷的数据读取和显示刷新工作全部由 FPGA 纯硬件完成，该架构特点从某种程度上确保了系统的健壮性和抗干扰性。

由于 FPGA 内部是采用并行处理机制，所以 Nandflash 的数据读取、SDRAM 的写入和 RGB 数据输出全部在一个时钟节拍下执行，确保了系统图片刷新速度快，响应迅速等特点。

设备内部有 640 个字节的指令缓冲区，用户主机可无等待、一次性发送多条指令后退出串口程序。整个过程操作简单，程序代码量大大降低。

## 操作与创新—30 分钟完成界面设计

任何大彩串口屏用户只需要 3 个步骤，即可快速完成复杂的人机交互设计。

### 1. 准备美工素材。

安排美工人员将开机画面、文本背景、按钮图标和提示框等产品所需图片设计好。

### 2. 利用配套的 VisualTFT 软件进行画面配置和图片下载。

首先利用上位机 VisualTFT 软件，将预先设计好的美工图片进行界面排版，编译无误后通过 UART 将整个工程图片下载到串口屏内部存储器中。下载之前，上位机会对工程中的每个图片分配一个唯一的 ID 号。

3. 用户单片机监听和发送相应的串口指令控制画面显示。

工程图片下载到设备后，一旦按下屏幕某个按钮，设备就会上传的当前的坐标值。通过判断坐标值(X, Y)范围，用户就可以知道当前触摸点是否为有效触控区域。若为有效区域，单片机就可以发送图片显示指令来切换画面或控制外围设备动作。

对于无触摸产品，用户单片机无需监听坐标值上传的信息，只需发送相关指令进行画面切换和文本显示。整个过程操作简单，程序代码量大大降低。

VisualTFT 软件特性

VisualTFT 是我司自主创新开发的一款功能强大的串口屏开发调试软件。用户新建工程后，导入设计好的美工图片，然后对每个画面中的按钮和其它控件进行配置，模拟仿真正确后，最后将整个工程下载到串口屏中。设备与 PC 连接成功后，还可以进行同步和调试显示，如图 B 所示。

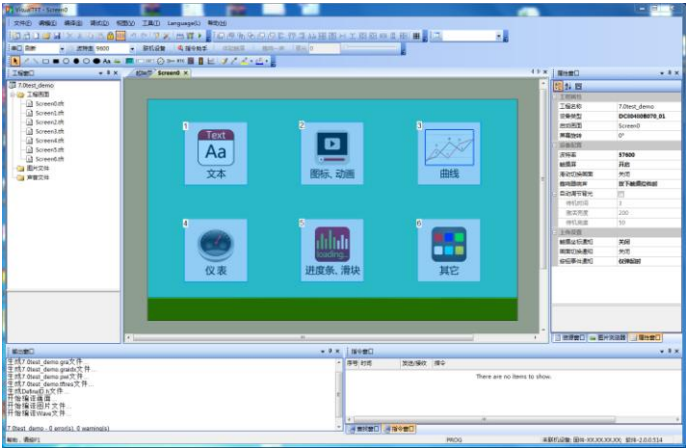


图 B VisualTFT 主界面

软件主要功能如下：

- VisualTFT IDE 环境操作人性化，界面美观大方；
- 软件集成了常用图标、按钮和键盘等矢量图，降低美工难度；
- 画面支持图片缩放、文字编辑和元素拷贝等常用操作功能；
- 工程编译后为每个画面、图片和控件分配唯一 ID 编号；
- 支持二进制文件烧录，量产更方便安全。
- 根据用户需求，定制特殊的 PC 软件功能；

产品特征

- 产品定位
  - Nano 系列主要应用单色/黑白屏升级、低成本和界面简单等工业应用场合；
- 型号和尺寸

	型号	尺寸	分辨率	备注
Nano 型	DC32240N024_01T/NW	2.4 "	320*240	T: 支持触摸 N: 无触摸
	DC32240N028_01T/NW	2.8 "	320*240	
	DC32240N030_01T/NW	3.0 "	400*240	

● 核心处理器单元

- 采用 32 位 Cortex-M0 内核 + 高速 FPGA 架构;
- 无操作系统, 纯硬件驱动显示, 上电即刻运行;

● 工作电压

- 标准 5.0V;

● 图片存储容量

- 内置标准 64MBit Flash, 支持任意大小图片存储, 累计全屏图片存储数量:

尺寸	2.4 "	2.8 "	3.0 "
数量 (张)	47	47	32

注: 存储器可扩展为 128Gbit 空间大小

● 通信接口

- 与用户主机 MCU 通信接口: RS232 或 3.3V CMOS/TTL UART;
- 图片下载接口: UART;

● 硬件功能特性

- 16 位真彩色 RGB 显示(64K 色);
- 内置标准 8\*16、12\*24、16\*32、32\*64 ASCII 字库, 16\*16 和 24\*24 GB2312 字库, 可自定义任意 windows 字库;
- 支持 BMP、JPEG、WMF、PNG 和 GIF 等图片格式下载;
- 支持任意位置文本和光标显示;
- 支持任意大小图片显示、图片裁剪、图片过滤和 GIF 动画显示;
- 支持画点、线、圆、矩形和多边形等常用 GUI 功能;
- 波特率设置范围 1200-1Mbps;
- 255 级背光调节模式;
- 不支持组态控件: 按钮控件、文本控件、进度条、仪表和滑块控件等;
- 按需定制, 满足用户一切需求。

● 关键参数指标

参数指标		2.4 "	2.8 "	3.0 "
单一颜色清屏(ms, 业界最快)		6.8	6.8	8.2
全屏图片更新时间(ms, 业界最快)		26	26	32
16*16 文本显示速度(ms, 业界最快)		0.55	0.55	0.55
功耗大小 (mA@5V)	背光最亮	95	100	110
	关闭背光	29	29	39

● LCD 特性

- 显示器类型: TFT LCD
- 背光灯管: LED
- 背光灯寿命 (平均): 大于 30,000 小时
- 对比度/亮度/视角: (选用常规参数、与屏级别相关)

● 触摸屏特性

- 触控类型: 4 线精密电阻式
- 触控方式: 连续式
- 透光率: 80%
- 触控次数: >80 万次

- 工作温度：-20℃～70℃ (与触摸板类型有关)
- 环境规格
  - 工作温度：-20℃～70℃ (与液晶屏类型有关)
  - 相对湿度：10%-90% @40℃,无凝露
  - 震动测试：10 to 25Hz(X,Y,Z 方向 2G 30 分钟)
  - ESD 标准：Air= ±8KV, Contact= ±4KV, class B (该参数与屏类型有关)
  - 上电即运行，满足复杂恶劣环境，365\*24\*7 小时不断电稳定工作
- 定制服务
  - 可定制 RS485、WIFI、Zigbee 或以太网等通信接口；
  - 定制 PCB 尺寸及厚度、添加用户电路、选用指定高亮度、特殊规格 TFT 屏；
  - 根据用户需要定制特殊指令或控件，降低主机开销；
  - 可定制电阻或电容触摸板；
  - 可提供图片美工及产品结构设计服务；
  - 一次性下单 500PCS 可免费进行定制，包含用户程序、PCB 贴标等；
  - 按需定制，满足用户一切需求；

## VisualTFT 软件特性

VisualTFT 是广州大彩光电科技有限公司([www.gz-dc.com](http://www.gz-dc.com))自主创新开发的一款功能强大的串口屏开发调试软件。用户新建工程后，导入设计好的美工图片，然后对每个画面中的按钮和其它控件进行配置，模拟仿真正确后，最后将整个工程下载到串口屏中。设备与 PC 连接成功后，还可以进行同步和调试显示，如图 D 所示。

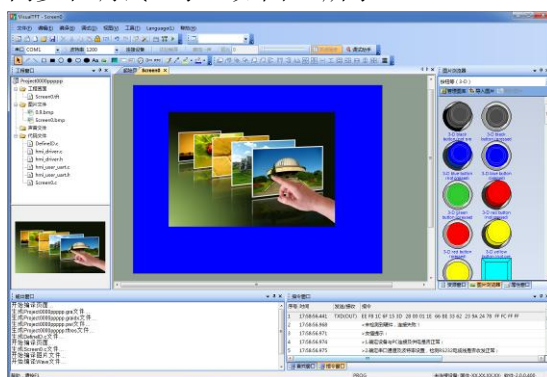


图 D VisualTFT 主界面

软件主要功能如下：

- VisualTFT IDE 环境操作人性化，界面美观大方；
- 软件集成了常用图标、按钮和键盘等矢量图，降低美工难度；
- 画面支持图片缩放、文字编辑和元素拷贝等常用操作功能；
- 工程编译后为每个画面、图片和控件分配唯一 ID，并自动生成驱动函数；
- 支持上位机与串口屏同步显示和调试功能；
- 内嵌“虚拟串口屏”模拟仿真器，节省开发周期；
- 支持二进制文件烧录，量产更方便安全。
- 根据用户需求，定制特殊的 PC 软件功能；

## 目 录

第 1 章 通信接口定义及设置 .....	1
1.1 通讯接口定义 .....	1
1.2 工程图片下载 .....	1
1.3 RS232 和 TTL 电平设置 .....	1
1.4 波特率恢复 .....	2
1.5 出厂清单和开发所需工具 .....	2
第 2 章 产品外观及机械尺寸图 .....	4
2.1 机械尺寸图 .....	4
第 3 章 小尺寸 Nano 串口屏指令集 .....	6
3.1 指令格式 .....	6
3.2 颜色格式 .....	7
3.3 指令列表 .....	8
第 4 章 Nano 指令集详述 .....	15
4.1 握手 .....	15
4.2 设置前/背景色 .....	15
4.3 清屏 .....	15
4.4 设置文字行列间距 .....	16
4.5 设置文本框 .....	16
4.6 设置图片过滤色 .....	16
4.7 文本显示 .....	17
4.8 光标显示 .....	17
4.9 全屏图片显示 .....	18
4.10 区域图片显示 .....	18
4.11 图片剪切 .....	19
4.12 动画显示 .....	19
4.13 前景色画点 .....	20
4.14 背景色画点（删除点） .....	20
4.15 画线 .....	20
4.16 将等间隔 X 坐标用前景色连接 .....	21
4.17 按照坐标偏移量用前景色连线 .....	21
4.18 将指定的坐标点用前景色连接 .....	21
4.19 将指定的坐标点用背景色连接 .....	22
4.20 按照坐标偏移量用背景色连线 .....	22
4.21 画空心圆 .....	22
4.22 画实心圆 .....	23
4.23 画圆弧 .....	23
4.24 画空心矩形 .....	23
4.25 画实心矩形/局部清屏 .....	24
4.26 画空心椭圆 .....	24
4.27 画实心椭圆 .....	24
4.28 背光调节 .....	24
4.29 自动屏保模式 .....	24

4.30	蜂鸣器控制 .....	25
4.31	配置触摸屏 .....	25
4.32	触摸屏校准 .....	26
4.33	触摸屏体验 .....	26
4.34	设置波特率 .....	27
4.35	锁定系统配置 .....	27
4.36	解除系统配置锁定 .....	28
<b>第 5 章 参考电路和程序 .....</b>		<b>29</b>
5.1	单片机与设备连接的参考图 .....	29
5.2	单片机控制设备的参考程序 .....	29
<b>第 6 章 产品储存与保养 .....</b>		<b>31</b>
6.1	处理保护膜 .....	31
6.2	焊接严防静电 .....	31
6.3	装配操作注意事项 .....	31
6.4	产品保养与存储 .....	31

## 第1章 通信接口定义及设置

设备对外只有 1 个 UART 接口，主要用于指令通讯和图片下载，如图 1.1 所示。

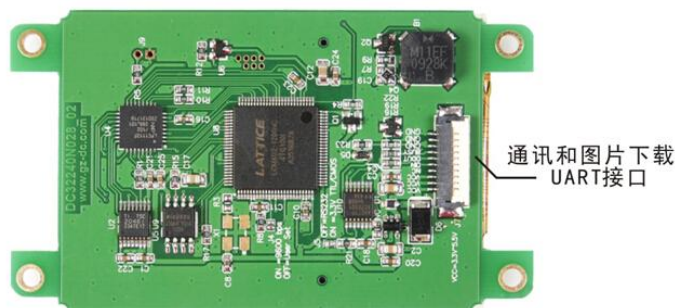


图 1.1 设备对外接口图

### 1.1 通讯接口定义

设备与主机通讯接口详细定义如表 1.1 所示。

表 1.1 2.4/2.8/3.0 寸通讯接口

	编号	名称	类型	说明
	1,2,3	VCC	输入	电源正(电压范围见产品特性)
	4	NC		空脚
	5	DIN	输入	设备数据接收引脚, 与用户 MCU 的 TXD 相连
	6	DOUT	输出	设备数据发送引脚, 与用户 MCU 的 RXD 相连
	7	BUSY	输出	设备忙信号输出(通常可悬空), 3.3/5V TTL/CMOS 电平, 连接用户主机任意 IO 口即可
	8,9,10	GND	输入	电源地

备注: BUSY=1, 表示设备处于忙状态无法接收新的指令; BUSY=0, 表示设备处于空闲状态。由于设备内部有 4K 字节的指令缓冲区, 大部分应用中用户无需使用 busy 信号。若用户主机有多余的 IO 资源, 建议使用该引脚。

### 1.2 工程图片下载

小尺寸 Nano 系列工程图片只能通过 UART 进行下载。产品调试时, 若工作波特率为 9600, 图片下载时, 建议先设置为 115200, 下载完毕后再设置回 9600。

**注意:** 电脑物理串口波特率最高为 115200, 不要轻易设置大于 115200, 否则只能采用波特率恢复操作方法来处理。若需要更高的 1M 串口下载速度, 请从我司购买高速 USB 串口调试板。

### 1.3 RS232 和 TTL 电平设置

设备出厂前均默认 RS232 电平, 若用户需要 TTL 通讯方式, 只需将 J5 焊盘短路。注意: 不同型号屏短点位置也不一样, 用户需要先找到通讯接口附近的白色丝印“ON= TTL/CMOS, OFF=RS232”, 然后在丝印附件会有一个 J5 短路点, 如图 1.2 所示。短路 J5, 屏幕就变成 TTL 电平, 断开就是 RS232。TTL 电平模式下兼容 3.3V 和 5V IO 输入输出。



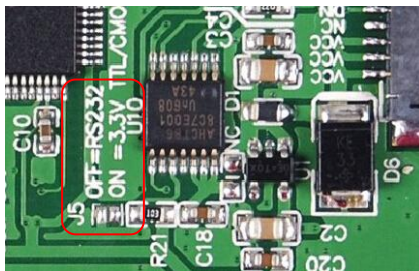


图 1.2 RS232 和 TTL 设置

## 1.4 波特率恢复

由于电脑硬件串口最高波特率为 115200, 实际操作中, 若用户误将波特率设置为 115200 以上, 可能导致 PC 与屏幕无法联机, 此时通过短接 J4 焊点即可恢复 9600。经济型的 PCB 丝印上有 1 个“ON=9600, OFF=UserSet”标识, 找到这个丝印后, 然后短路附近的 J4 点(镊子短路即可), 短路情况下重新上电, 此时波特率强制恢复到 9600bps。解除短路后, 就可以进行其它波特率设置, 如图 1.3 所示。

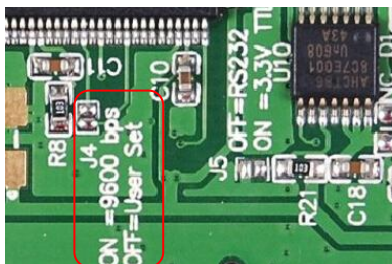


图 1.3 波特率恢复

## 1.5 出厂清单和开发所需工具

NANO 型系列工业串口屏开发所需工具清单如下, PC 联机设备如图 1.4 所示。

- (1) 电源适配器。建议电流 500mA 以上, 保证足够电流。
- (2) DB9 串口通讯电缆。用于与上位机软件进行联机、图片下载和调试等。
- (3) 串口调试板。将屏的 10PIN 通讯接口转换为单独电源和 DB9 接口, 方便供电和 PC 串口调试。



图 1.4 2.4/2.8/3.0 寸与 PC 联机通信

产品出厂时标配清单如表 1.1 所示。

表 1.1 产品出厂标配清单

类型	产品	数量	备注
标配清单	串口屏	1 个	--
	FPC1.0/10P 软线缆	1 条	连接屏和用户主控板
	串口调试板	--	购买样品时赠送，批量出货不提供
	10P 软排线端子	1 个	焊接用户主板上
选配	电源 5V	1 个	设备供电
	USB 转串口电缆线	1 条	若电脑无串口，需购买 USB 转串口线

## 第2章 产品外观及机械尺寸图

本章主要介绍常用尺寸的外观和机械尺寸图，其它最新尺寸图可以登录公司网站 [www.gz-dc.com](http://www.gz-dc.com) 进行查阅和下载。

### 2.1 机械尺寸图

各尺寸型号机械尺寸图如下所示。

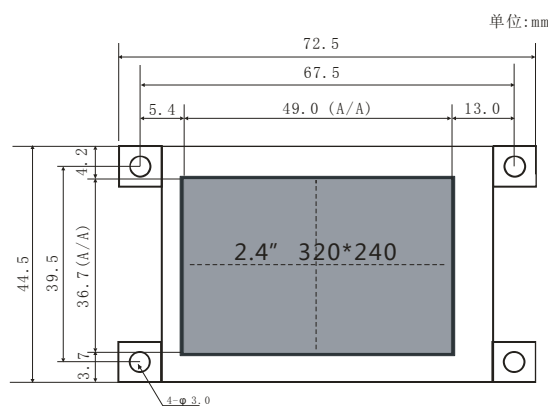


图 2.1 2.4 寸串口屏机械尺寸图

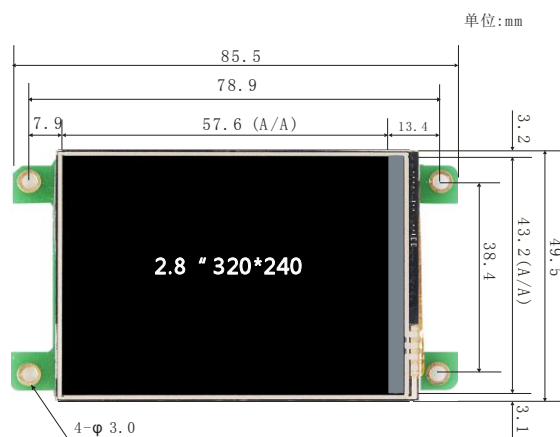


图 2.2 2.8 寸 DC32240N028\_01TW/NW 机械尺寸图

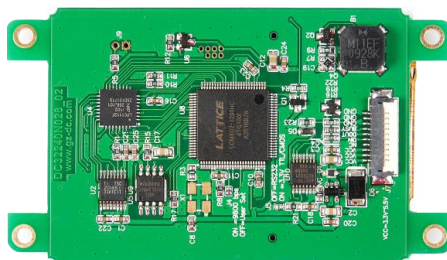


图 2.3 2.8 寸 DC32240N028\_01TW/NW 背面实物图

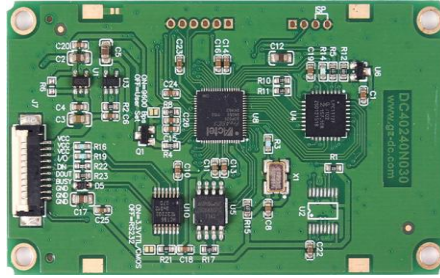


图 2.4 DC40240N030\_01TW/NW 串口屏背面实物图

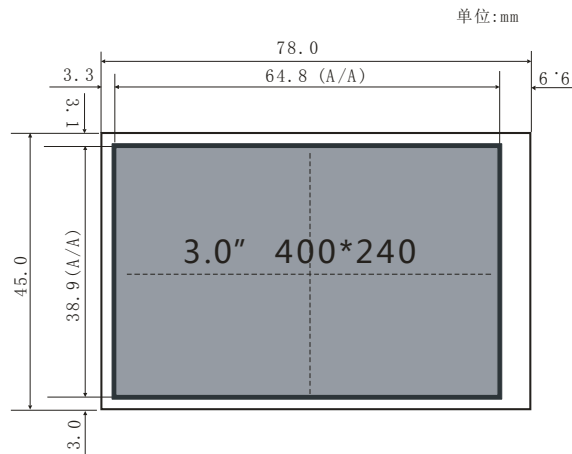


图 2.5 DC40240N030\_01TW/NW 串口屏机械尺寸图

第3章 小尺寸 Nano 串口屏指令集

3.1 指令格式

一条完整的串口指令帧格式如表 3.1 指令帧格式 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 3.1 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

(1) 若用户需要在如图 3.1 所示的坐标(100,50)处显示 ID 为 2 的图片，则主机发送的串口指令如下：

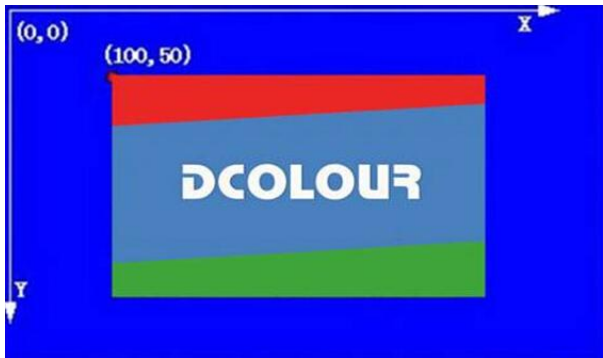


图 3.1 区域图片显示

用户主机发送命令：EE 【32 00 64 00 32 00 02 00】 FF FC FF FF

命令解析：EE 表示帧头；  
32 表示区域图片显示指令；  
00 64 00 32 代表显示坐标为 (100, 50)，高字节在前；  
00 02 代表图片的编号 (PC 在下载前会对所有图片进行编号)；  
00 代表不需要颜色过滤；  
FF FC FF FF 表示帧尾

(2) 若用户点击触摸图 3.2 中的“首页”按钮，则串口屏上传串口指令如下：



图 3.2 按钮控件 ID 值上传

设备主动上传指令：EE【01 01 90 01 C2】FF FC FF FF

指令解析：EE 表示帧头；  
01 表示按下触摸后立即上传坐标；  
01 90 01 C2 表示当前按下的触摸坐标值为(400,450)  
FF FC FF FF 表示帧尾。

用户主机串口接收到坐标值后就可以判断当前屏幕哪个区域被按下，然后再发送相应的指令去控制显示。

由于 Nano 型不支持组态功能，按下触摸后只能上传坐标值，所以用户需要自己程序判断坐标值的位置来确定当前屏幕按下的位置，而经济型和基本型支持组态功能，当用户按下某个按钮后，不仅可以设置屏幕自动切换到哪个画面，还可以设置设备只上传该按钮的 ID 值，无需计算坐标。

(3) 若用户需要在图 3.3 中当前电压显示数字 220，则用户主机发送指令：EE【20 02 3E 00 B3 01 02 32 32 30】FF FC FF FF。



图 3.3 文本显示

指令解析：  
EE 表示帧头；  
20 表示文本显示指令；  
02 3E 00 B3 表示显示坐标为 (574,179)；  
01 表示打开背景色；  
02 表示字体大小为 12\*24；  
32 32 30 表示数字 220 的 ASCII 码；  
FF FC FF FF 表示帧尾。

## 3.2 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 3.2 所示。

表 3.2 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=F800H，■ 纯蓝色=001FH

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。

### 3.3 指令列表

指令列表分为基本指令集和组态控件指令集，主要描述了各指令参数的定义和用户 MCU 驱动函数库，分别如表 3.3 所示。

表 3.3 Nano 型系列指令集

类别	指令	指令参数	说明
握手	0x04	无	设备接收握手命令后, 返回 55 给主机以示握手成功或在线状态。 返回格式: EE 55 FF FC FF FF
			驱动函数: void SetHandShake ( );
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
			驱动函数: void SetFcolor(uint16 fcolor);
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
			驱动函数: void SetBcolor(uint16 bcolor);
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
			驱动函数: void GUI_CleanScreen();
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
			驱动函数: void SetTextSpace(uint8 y_w, uint8 x_w);
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) 0x01: 打开文本框限制使能, 0x00: 关闭文本框限制使能 Width (2 个字节): 文本显示框的宽度 Hight (2 个字节): 文本显示框的高度
			驱动函数: DisText_Region(uint16 Width,uint16 Hight );
设置图片过滤色	0x44	FilterColor	图片中的颜色与过滤色值相同时不予显示 FillColor (2 个字节): 过滤色 RGB 值
			驱动函数: void SetFilterColor(uint16 FilterColor);
文本显示	0x20	X+Y+Back+Font+String	任意坐标处显示指定大小的文本内容 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 Back (背景色, 1 个字节) 0x01: 打开背景色显示 0x00: 关闭背景色显示 Font (字库编码, 1 个字节) 0x01: 8x16 点阵(ASCII) 0x02: 12x24 点阵(ASCII) 0x03: 16x32 点阵(ASCII)

			<p>0x08: 32 x 64 点阵(ASCII)</p> <p>0x07: 32 x 32 点阵(GB2312)</p> <p>0x0A:16X16 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>
			<p>驱动函数: void DisText(uint16 x, uint16 y,uint8 back,uint8 font,uchar *strings);</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>
			<p>驱动函数: void DisCursor(uint8 enable,uint16 x, uint16 y,uint16 Width,uint16 Hight);</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>
			<p>驱动函数: void DisFull_Image(uint16 image_id,uint8 masken);</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> <p>备注: 被过滤的颜色取决于过滤色的设置。</p>
			<p>驱动函数: void DisArea_Image(uint16x,uint16y,uint16 image_id,uint8 masken);</p>
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> <p>驱动函数: void DisCut_Image(uint16 x,uint16 y, uint16 image_id, uint16 image_x, uint16 image_y, uint16 Image_w, uint16 Image_h, uint8 masken);</p>
动画显示	0x80	X+Y+FlashImgae_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。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> <p>驱动函数: void DisFlashImgae(uint16 x,uint16 y,uint16 flashimage_id, uint8 enable, uint8 playnum);</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> <p>驱动函数: void GUI_FcolorDot(uint16 x,uint16 y);</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> <p>驱动函数: void GUI_BcolorDot(uint16 x,uint16 y);</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> <p>驱动函数: void GUI_Line(uint16 x0, uint16 y0, uint16 x1, uint16 y1);</p>
按照等间隔 X 坐标 用前景色连线	0x59	X <sub>0</sub> +Xspace+Y <sub>0</sub> +...+Y <sub>n</sub>	<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> <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</p> <p>void GUI_ConSpaceDots(uint16 Xspace, uint8 *pDot);</p>
按照坐标偏移量用前景色连线	0x75	$(X,Y)_0 + (X1_0,Y1_0) + \dots + (X_n,Y_n)$	<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> <p>void GUI_FcolorConOffsetDots(uint8 *pDot);</p>
按照坐标偏移量用背景色连线	0x76	$(X,Y)_0 + (X1_0,Y1_0) + \dots + (X_n,Y_n)$	<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> <p>void GUI_BcolorConOffsetDots(uint8 *pDot);</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> <p>驱动函数: void GUI_FcolorConDots(uint8 *pDot);</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> <p>驱动函数: void GUI_BcolorConDots(uint8 *pDot);</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> <p>驱动函数: void GUI_Circle(uint16 x0, uint16 y0, uint16 r);</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> <p>驱动函数: void GUI_CircleFill(uint16 x0, uint16 y0, uint16 r);</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>X0 (2 个字节): 以点为单位的圆心 X 坐标值</p> <p>Y0 (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> <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</p> <p>驱动函数: void GUI_Arc (uint16 x0,uint16 y0, uint16 r,uint16 sa, uint16 ea);</p>
画空心矩形/局部清屏	0x54	$X_0+Y_0+X_1+Y_1$	<p>任意位置画一个空心矩形, 也可用于局部清屏使用</p> <p>X0(2 个字节): 以点为单位的空心矩形左上角 X 坐标值</p> <p>Y0(2 个字节): 以点为单位的空心矩形左上角 Y 坐标值</p> <p>X1(2 个字节): 以点为单位的空心矩形右下角 X 坐标值</p> <p>Y1(2 个字节): 以点为单位的空心矩形右下角 Y 坐标值</p> <p>备注: 颜色值取决于前景色设置</p> <p>驱动函数: void GUI_Rectangle(uint16 x0, uint16 y0, uint16 x1,uint16 y1 );</p>
画实心矩形	0x55	$X_0+Y_0+X_1+Y_1$	<p>任意位置画一个实心矩形</p> <p>X0(2 个字节): 以点为单位的实心矩形左上角 X 坐标值</p> <p>Y0(2 个字节): 以点为单位的实心矩形左上角 Y 坐标值</p> <p>X1(2 个字节): 以点为单位的实心矩形右下角 X 坐标值</p> <p>Y1(2 个字节): 以点为单位的实心矩形右下角 Y 坐标值</p> <p>备注: 颜色值取决于前景色设置</p> <p>驱动函数: void GUI_RectangleFill(uint16 x0, uint16 y0, uint16 x1,uint16 y1 );</p>
画空心椭圆	0x56	$X_0+Y_0+X_1+Y_1$	<p>任意位置画一个空心椭圆</p> <p>X0(2 个字节): 以点为单位的空心椭圆最左端 X 坐标值</p> <p>Y0(2 个字节): 以点为单位的空心椭圆最上端 Y 坐标值</p> <p>X1(2 个字节): 以点为单位的空心椭圆最右端 X 坐标值</p> <p>Y1(2 个字节): 以点为单位的空心椭圆最下端 Y 坐标值</p> <p>说明: 颜色值取决于前景色设置</p> <p>驱动函数: void GUI_Ellipse (uint16 x0, uint16 y0, uint16 x1,uint16 y1 );</p>
画实心椭圆	0x57	$X_0+Y_0+X_1+Y_1$	<p>任意位置画一个实心椭圆</p> <p>X0(2 个字节): 以点为单位的实心椭圆最左端 X 坐标值</p> <p>Y0(2 个字节): 以点为单位的实心椭圆最上端 Y 坐标值</p> <p>X1(2 个字节): 以点为单位的实心椭圆最右端 X 坐标值</p> <p>Y1(2 个字节): 以点为单位的实心椭圆最下端 Y 坐标值</p>

			说明: 颜色值取决于前景色设置 驱动函数: void GUI_EllipseFill (uint16 x0, uint16 y0, uint16 x1,uint16 y1);
背光调节	0x60	Light_level	设置背光亮度值 0x00: 背光最亮 0xFF: 背光关闭 驱动函数: void SetBackLight(uint8 light_level);
蜂鸣器控制	0x61	Time	Time(1 个字节): 讯响时间, 单位为 10ms 驱动函数: void SetBuzzer(uint8 Time);
配置触摸屏	0x70	Cmd	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: 触摸上传方式 000: 表示按下触摸屏时才上传一次坐标 001: 表示触摸屏被按下直至释放后才上传一次坐标 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 011: 表示触摸屏被按下和释放时分别上传一次坐标 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; BIT7-BIT6: 保留 触摸上传格式: 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 坐标、Y 坐标均为 2 个字节, 高字节在前 驱动函数: void SetTouchScreen_Reg(uint8 cmd)
触摸屏校准	0x72	无	校准完毕后, 设备返回 EE 04 FF FC FF FF 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 驱动函数: void SetTouchScreen_Adj (void)
触摸屏体验	0x73	Enable	Enable: 使能信号 0x00: 关闭体验 0x01: 体验使能 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。 驱动函数: void TestTouchScreen (void)
设置波特率	0xA0	Baudset	Baudset(单位 bps,1 个字节), 波特率编序: 0x00: 1200      0x01: 2400      0x02: 4800 0x03: 9600      0x04: 19200      0x05: 38400 0x06: 57600      0x07: 115200      0x08: 1M 0x09: 2M      0x0A : 625000 驱动函数: void SetCommBps(uint8 Baudset);
屏幕旋转	0x74	Direct	设置屏幕横屏或竖屏显示

			Direct ( 1 个字节 ): 0x00 表示横屏显示 0x01 表示竖屏显示
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> <p>驱动函数:Locksysconfig(void)</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> <p>驱动函数:Unlocksysconfig(void)</p>

## 第4章 Nano 指令集详述

详细的指令集说明如下列章节所述。

### 4.1 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。

### 4.2 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 4.1 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 4.1 背景前景色说明

### 4.3 清屏

指令格式: EE 【01】 FF FC FF FF

参数说明: 无参数

该指令用于实现指定颜色清屏,清屏颜色取决背景色的设置。若用户未进行背景色设置,清屏色默认为蓝色。

销售咨询: 020-82186683-601

Email: hmi@gz-dc.com

欢迎登陆 [www.gz-dc.com](http://www.gz-dc.com) 了解更多...

广州大彩光电科技有限公司 版权所有

## 4.4 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
}
```

## 4.5 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。

## 4.6 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 4.2 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 4.2 设置过滤色前后对比图

## 4.7 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**错误! 未找到引用源。**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 4.3 文本显示效果图

## 4.8 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**错误! 未找到引用源。**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
}
```

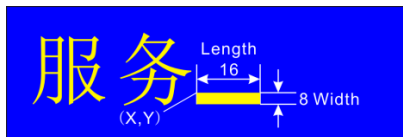


图 4.4 光标参数说明

## 4.9 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。

## 4.10 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, 如错误! 未找到引用源。所示, 参考程序如下所示。



图 4.5 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
}
```

## 4.11 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，剪切效果如错误！未找到引用源。所示。

## 4.12 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错误！未找到引用源。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 4.6 图片动画显示效果图

### 4.13 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。

### 4.14 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**错误! 未找到引用源。**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。

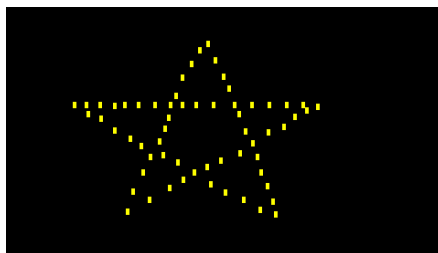


图 4.7 前景/背景色画点

### 4.15 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**错误! 未找到引用源。**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

销售咨询: 020-82186683-601

Email: hmi@gz-dc.com

欢迎登陆 [www.gz-dc.com](http://www.gz-dc.com) 了解更多...

广州大彩光电科技有限公司 版权所有

错误！未找到引用源。所示。

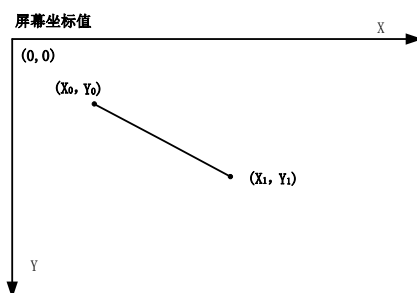


图 4.8 画线参数说明

#### 4.16 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 个字节): 以点为单位的 X 轴坐标值

Xspace(2 个字节): 以点为单位的 X 轴相邻两点固定的间隔值

Y<sub>n</sub> (2 个字节): 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。

#### 4.17 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。

#### 4.18 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。例如要实现错误！未找到引用源。所示的折线和六边形形状, 程序如下所示。

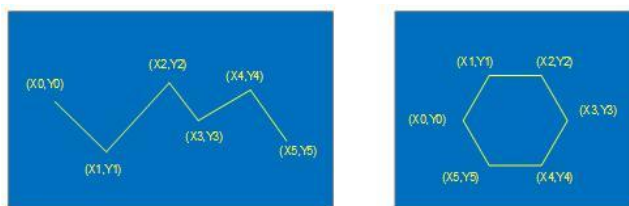


图 4.9 指定坐标点用前景色连接效果图

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
}
```

#### 4.19 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。

#### 4.20 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。

#### 4.21 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 个字节)：以点为单位的圆心 X 坐标值

Y<sub>0</sub> (2 个字节)：以点为单位的圆心 Y 坐标值

R (2 个字节)：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。参数说明如错误！未找到引用源。所示。

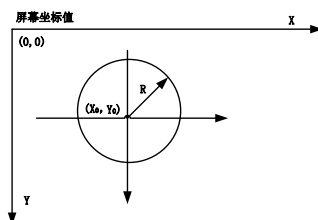


图 4.10 画空心圆参数说明

## 4.22 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。  
参数说明与错误! 未找到引用源。相同。

## 4.23 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, 参考坐标如错误! 未找到引用源。所示。

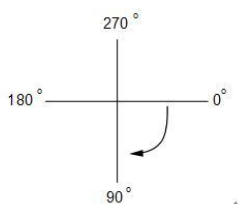


图 4.11 圆弧起始角度参考图

## 4.24 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。参数定义如错误! 未找到引用源。所示。

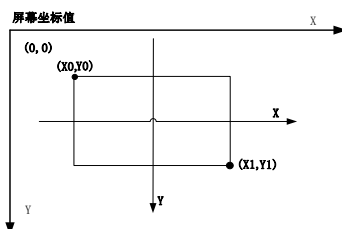


图 4.12 画空心矩形参数说明



## 4.25 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。参数定义与错误! 未找到引用源。相同。该功能还可以作为局部清屏使用。

## 4.26 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。参数定义说明如错误! 未找到引用源。所示。

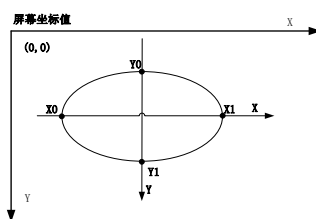


图 4.13 画空心椭圆参数说明

## 4.27 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<sub>0</sub> (2 字节): 以点为单位的实心椭圆最左端 X 坐标值  
Y<sub>0</sub> (2 字节): 以点为单位的实心椭圆最上端 Y 坐标值  
X<sub>1</sub> (2 字节): 以点为单位的实心椭圆最右端 X 坐标值  
Y<sub>1</sub> (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。

## 4.28 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。

## 4.29 自动屏保模式

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

销售咨询: 020-82186683-601

Email: hmi@gz-dc.com

欢迎登陆 [www.gz-dc.com](http://www.gz-dc.com) 了解更多...

广州大彩光电科技有限公司 版权所有

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

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。

### 4.30 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。

### 4.31 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

BIT5: 1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, 0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**错误! 未找到引用源。**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。



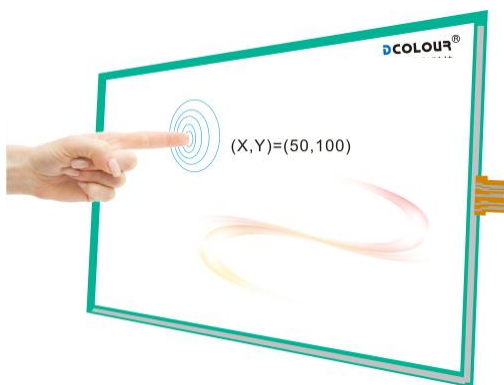


图 4.14 触摸屏工作介绍

### 4.32 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**错误！未找到引用源。**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。

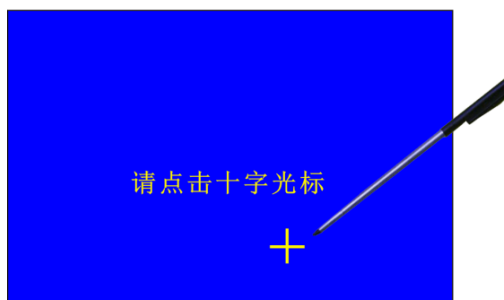


图 4.15 触摸屏校准示意图

### 4.33 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**错误！未找到引用源。**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。

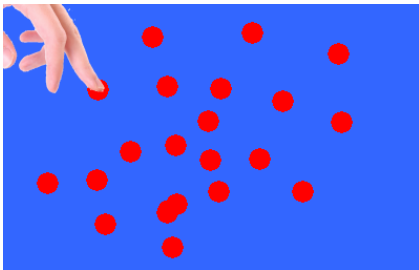


图 4.16 触摸体验效果图

4.34 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 : 218750	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。用户可以通过上位机 VisualTFT 的“调试助手”配置新的波特率，如错误！未找到引用源。所示。

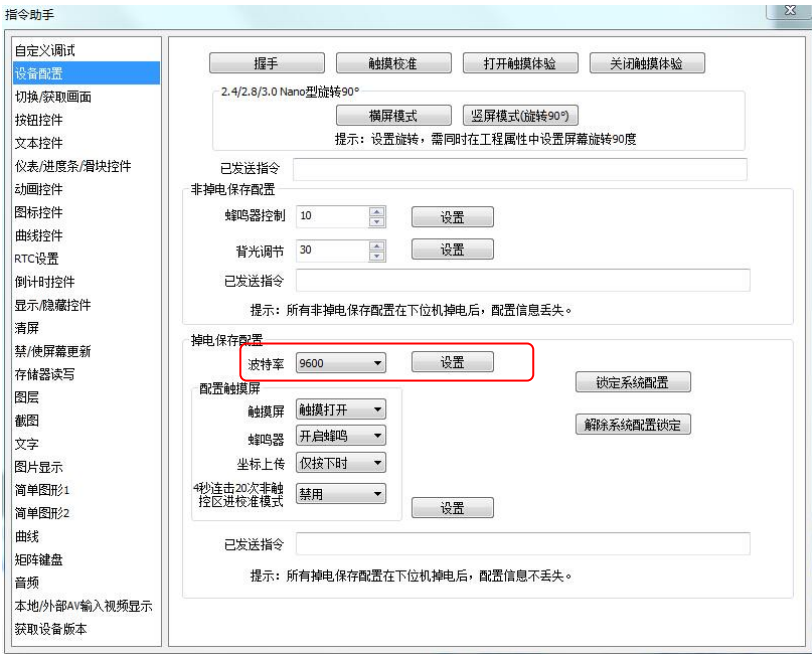


图 4.17 波特率设置

4.35 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。

#### 4.36 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。

## 第5章 参考电路和程序

### 5.1 单片机与设备连接的参考图

用户单片机与设备 RS232 通讯方式电路参考图 5.1 所示。若需要 3.3V TTL/COMS 串口电平通信，用户可以直接跳过 MAX232 电平转换芯片，同时将屏设置为 TTL 模式(默认 232 状态)。另外在 3.3V TTL/CMOS 模式下，设备可以容忍 5V 输入。

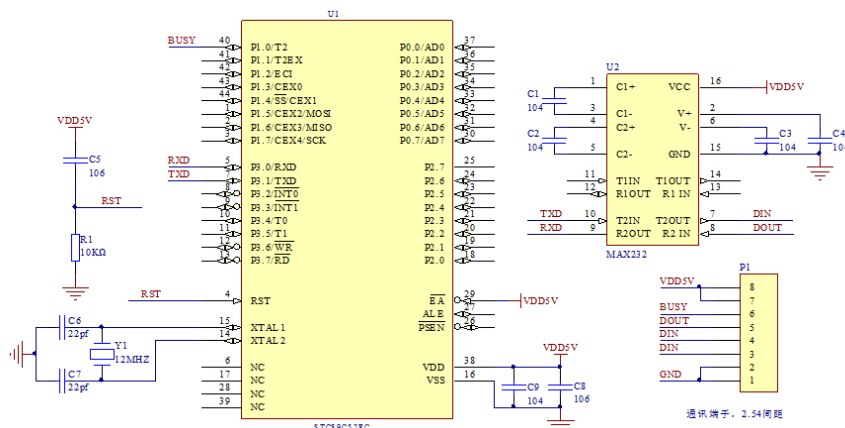


图 5.1 单片机与设备硬件连接参考图

### 5.2 单片机控制设备的参考程序

(1) 若用户按下如图 5.2 所示的画面 0 “文本显示”按钮后，需要切换到画面 1 显示，并在电压值栏写入“220V”，电流值栏显示 0~99 不断变化数据。



图 5.2 画面 0 和画面 1 内容显示

程序设计流程图如下

1. 判断触摸坐标是否落在文本显示区域内；
2. 发送图片显示指令，进行画面切换；
3. 发送文本指令，在指定位置写入文本。

参考程序如下：

```
if((TouchValueX > 554 && TouchValueX < 758) && (TouchValueY > 106 && TouchValueY < 158))
//表示文本显示按钮被按下，坐标值落在文本显示区域范围内
```

```
{
    DisFull_Image(1,0); //显示图片 ID=1 的全屏图片，即进入画面 1 显示界面
    uchar powerdata[3];
```

销售咨询：020-82186683-601

Email: hmi@gz-dc.com

欢迎登陆 [www.gz-dc.com](http://www.gz-dc.com) 了解更多...

广州大彩光电科技有限公司 版权所有

```
DisText(248, 128, 0, 6, "220V"); //文本显示操作，在坐标(248,128)处写入 12*24 的字符串“220V”
for( i=0;i<100;i++)           //文本显示操作
{
    powerdata[0] = 0x30+i/10;
    powerdata[1] = 0x30+i%10;
    powerdata[2] = '\0';
    DisText(253, 240, 0, 6, powerdata); //在坐标(253,240)处写入 12*24 的 0-99 变化数据
    delay_ms(20);
}
}
```

说明：由于 Nano 型不支持组态控件，所以需要用户程序判断当前触摸坐标范围，然后再发送图片显示指令切换到新的画面。

## 第6章 产品储存与保养

感谢您选择我司研发生产的串口屏系列产品，我们将竭诚为您提供全方位的技术支持及售后服务。液晶属于易碎精密器件，虽然出厂前做了各种可能性的保护，但为了您能够更好的使用本产品，仍然建议您在使用之前仔细阅读如下注意事项，以免造成不必要的损失。

### 6.1 处理保护膜

出厂前串口屏表面上都有一层保护膜，以防止运输、安装、调试过程中沾污显示屏。在剥离保护膜时候，可能会产生静电，需要做好防静电措施。正常使用时，建议去掉触摸屏上的保护膜，防止触摸敏感度下降。

### 6.2 焊接严防静电

串口屏有两处短路焊盘供用户设置选用，分别用于恢复出厂波特率设置和 RS232/TTL (COMS)电平选择设置。产品出厂前均默认设置为 RS232 电平，波特率 115200bps，用户如需更改设置，焊接前必须注意以下几点：

1. 不要用手随意触摸短路焊盘，以防止短路；
2. 用手触摸模块各个元件时，应该保持人体和模块在同一电位或者人体良好接地，以防人体静电损害产品；
3. 焊接使用的电烙铁和操作工具都必须良好接地，以防漏电事故；
4. 工作间湿度应该在 RH60%左右；
5. 烙铁头焊接温度要求：无铅焊接小于 320℃，有铅焊接小于 280℃；
6. 焊接时间小于 3~4 秒，不要使用酸性助焊剂；

### 6.3 装配操作注意事项

所有产品都是经过精心设计组装而成，用户请勿自行进行再加工、修整。用户装配操作时候，注意如下几点：

1. 显示屏放置妥当，切勿倾斜，以免影响可视范围；
2. 整体设备需固定，以减小震动。

### 6.4 产品保养与存储

由于显示屏长期暴露在外，容易沾染污垢、灰尘，需要不定期清理。推荐使用异丙基酒精或乙荃酒精清洁块处理。若某种原因需要长期储存，我们推荐几下方式：

1. 产品装入静电袋并封口；
2. 放置在暗处，避免强光照射；
3. 存储室温度-10℃~60℃；
4. 不能在表面压放任何物品。