C 系列彩屏智能终端使用说明书

（V1.1.6）

感谢您关注和使用金鹏彩屏系列液晶显示器产品，欢迎您提出意见和建议，我们将竭诚为您服务、让您满意。您可以浏览 **http://www.gptlcm.cn** 了解最新的产品与应用信息，或拨打热线电话 **0758－2317156** 及向 **syl@gptlcm.cn** 邮箱发 **E-mail** 获取具体的技术咨询与服务。

金鹏实业有限公司

Golden Palm Industry Co., Ltd.

地址：肇庆市建设四路 13 号天宁广场 E 幢 13、14、15 楼 邮编：526040 E-mail: syl@gptlcm.cn
业务联系：0758－2317976/2317972 技术支持：0758－2317156 传真：0758－2317023

目录

地址：肇庆市建设四路 13 号天宁广场 E 幢 13、14、15 楼 邮编：526040 E-mail: syl@gptlcm.cn
业务联系：0758－2317976/2317972 技术支持：0758－2317156 传真：0758－2317023

一、产品简介

肇庆金鹏电子集团有限公司的彩屏智能型彩屏终端，是在汲取了众多客户要求和建议的基础上，采用 32 位 ARM 处理器 + FPGA 双核控制架构开发的一款高性能、低功耗、易使用的 64K 色的 TFT 真彩显示器，可以直接和具有 UART 串行接口的 MCU（如 51 单片机、AVR、PIC、DSP、ARM、工控机等）连接。用户只需通过串口向终端发命令，便可完成相应的操作。

本产品的主要特点如下：

- 处理器
  - 采用 32 位 ARM 处理器 + FPGA 双核控制架构，加强图像处理功能。
- 存储容量
  - 1GBit Flash 存储容量，储存一百多张 16bit 真彩色图片；
- 接口特性
  - 图片下载接口：全速 USB，速度 600KB/ S；
  - 通讯接口：3.3V RS232 或 TTL/CMOS 电平。
- PC 软件功能
  - 强大的 IDE 编译下载环境，可视化窗口、界面美观大方；
  - IDE 集成了大量工控行业图标、按钮、3D 视图等矢量图，降低了美工难度；
  - 支持新建多个页面，编译后软件自动生成每个页面的驱动函数；
  - 支持 PC 软件与 HMI 同步显示，具有单步调试等功能；
  - 支持二进制文件烧录，量产更快更安全。工程编译后 IDE 将自动生成工程二进制文件。
- 硬件特性
  - 16 位真彩色 RGB 显示（65536 色）；
  - 支持最高分辨率位 800*600；
  - 内置标准 8*12、8*16、12*24、16*32 的 ASIC 字库，12*12、16*16、24*24GBK 字库和 32*32 的 GB2312 字库；
  - 支持光标显示；
  - 支持画点、圆、直线、矩形等 GUI 功能；
  - 图片下载支持格式 JPG、BMP、JPEG、WMF、PNG 和 GIF；
  - 支持任意位置图片显示、图片裁剪、区域图像更新；
  - 支持 GIF 动画显示；
  - 255 级可控背光控制；
  - 波特率范围 1200-115200bps。
- 电源

  3.5~4.3 寸供电电压为 5V，5.6 寸以上供电为 7.6V~24V
- 工作温度
  - -20℃～70℃

二、产品选型表

尺寸	型号	分辨率	比例	工作电压			工作电流（标准电压）		
				最小	标准	最大	最小	标准	最大
3.5	OCM320240T350-1C	320*240	4:3	4.5V	5V	5.5V	110mA	215mA	240mA
4.3	OCM480272T430-1C	480*272	16:9	4.5V	5V	5.5V	130mA	280mA	330mA
5.6	OCM640480T560-1C	640*480	4:3	5V	9V	24V	140mA	290mA	310mA
7.0	OCM800480T700-1C	800*480	16:9	7.6V	9V	24V	170mA	420mA	460mA
8.0	OCM800600T800-1C	800*600	4:3	7.6V	9V	24V	190mA	435mA	460mA
10.4	OCM800600T104-1C	800*600	16:9	7.6V	9V	24V	220mA	450mA	480mA

其中，全部 C 系列产品均可选配触摸屏、实时时钟；5.6 寸、7.0 寸、8.0 寸、10.4 寸均可选配实时时钟。请在采购的时候向我们的销售人员说明。

三、极限参数

3.1 温度参数

参数	符号	最小值	最大值	单位
工作温度	Top	-20	+70	℃
存储温度	Tst	-30	+80	℃

3.2 电气参数

类别		符号	最小值	典型值	最大值	单位
输入电压	TTL 高电平	Vih	2.1	--	5.5	V
	TTL 低电平	Vil	-	--	0.9	V
输出电压	TTL 高电平	Voh	--	3.3	--	V
	TTL 低电平	Vol	--	--	0.1	V

四、通讯接口定义及方式设置

4.1 通讯接口定义

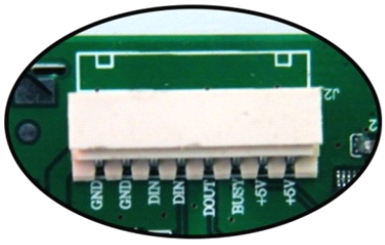


图 4.1 指令通讯端口实物图

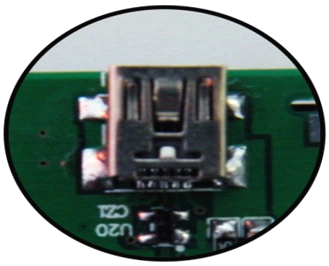


图 4.2 图片下载端口实物图

表 4.1 指令通讯端口定义

	编号	名称	类型	说明
	1.2	VCC	输入	电源
	3	BUSY	输出	忙信号输出(通常可悬空)，3.3V CMOS 电平
	4	DOUT	输出	发送(TXD)引脚，与用户 MCU 的 RXD 相连
	5.6	DIN	输入	接收(RXD)引脚，与用户 MCU 的 TXD 相连
	7.8	GND	输入	电源地
备注：BUSY=1，表示处于忙状态无法接收新的指令；BUSY=0，表示处于空闲状态。由于内部有 6K 的指令缓冲区，一般应用中用户可以无需判断 busy 信号。				

表 4.2 图片下载端口定义

	名称	说明
	USB 接口	图片下载接口，高速 USB2.0 标准

注意：本 USB 不支持热插拔。必须先断开彩屏的 VCC 电源，再插拔 USB。

图 4.3 标配通讯电缆线图

4.2 通讯方式及波特率设置

出厂前默认 RS232 电平，若用户需要 TTL/CMOS 通讯方式，需将下图 4.4 所示的焊点（J6）

短路。出厂前波特率默认 9600bps，用户可以通过上位机进行其它波特率值设置。实际操作中，若用户忘记了之前所设的波特率，可以短接下图 4.5 所示的焊点(J5)，然后重启屏，此时屏将恢复出厂前设置的 9600bps。新的波特率值设置后断电不丢失。

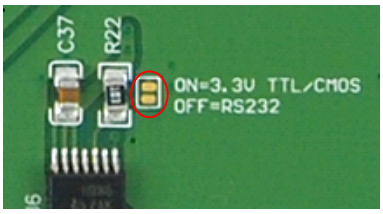


图 4.4 通讯方式设置

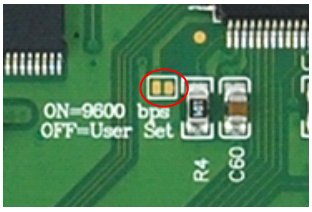


图 4.5 通讯波特率恢复

五、指令集表

5.1 指令格式

一条完整的用户命令格式如表 5.1 所示。如果指令参数多于一个字节,统一以 MSB 方式(高字节在前,低字节在后)进行通信，每帧最大的数目是 1024 字节(包含帧头和帧尾)，指令数据均以十六进制表示。

表 5.1 指令帧的格式

指令	EE	XX	XX	FF FC FF FF
说明	帧头	指令	指令参数(最长 1018 字节)	帧尾

举例说明，在坐标（100，50）处显示一幅指定的图片，如下图 6.1 所示。

图 6.1 指定位置显示图片

用户 MCU 串口发送命令如下：

发送命令：EE 【32 00 64 00 32 00 02 00】FF FC FF FF

命令解析：EE 代表帧头；

32 代表区域图片显示指令；

00 64 00 32 代表（x,y)坐标（100，50），高字节在前；

00 02 代表图片的编号（上位机对所有图片自动进行ID 分配）；

00 表示无颜色需过滤

FF FC FF FF 代表帧尾。

5.2 颜色格式

支持 2^16=65536 种颜色（简称 65K 色），其 RGB 高低字节分配如表 6.2 所示。

表 5.2 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。

5.3 指令列表

指令列表内容分 2 部分：指令参数定义和驱动函数。每一条指令都有对应的驱动函数，驱动库源码文件为 hmi_driver.c 和 hmi_driver.h，该部分函数的具体实现可在所配套的金鹏终端开发软件里的工程窗口-代码文件里找到。使用驱动库时，用户必须仔细阅读 hmi_driver.c 文件包含的“使用必读”内容。

表 6.3 指令列表

类别	指令	指令参数	说明
握手命令	0x00	无	HMI 接收握手命令后发送 55 给主机,以示握手成功
			驱动函数：void SetHandShake ( );
握手命令 1	0x04	无	HMI 接收握手命令后返回 EE 55 FF FC FF FF 给主机,以示握手成功
设置前景色	0x41	Fcolor	Fcolor(2 字节) 前景色用于点、线、圆、图形和文字的颜色指定
			驱动函数：void SetFcolor(uint16 fcolor);
设置背景色	0x42	Bcolor	Bcolor(2 字节) 背景色用于清屏和文字的背景色颜色指定
			驱动函数：void SetBcolor(uint16 bcolor);
取屏幕点设置前景色/背景色	0XA3	Mode +X+Y	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 Mode(1 字节) : 0x00 取当前显示屏幕(X,Y)处颜色作为前景色 0x01 取当前显示屏幕(X,Y)处颜色作为背景色
			驱动函数：void ColorPicker(uint8 mode, uint16 x,uint16 y);
清 屏	0x01	无	清屏颜色取决于背景色调色板设置
			驱动函数：void GUI_CleanScreen( );
设置文字行列间距	0x43	X_W+ Y_W	X_W(1 字节) 以点为单位的列间距，取值 00~3F Y_W(1 字节) 以点为单位的行间距，取值 00~3F
			驱动函数：void SetTextSpace(uint8 x_w, uint8 y_w);
设置文本框	0x45	Enable+Length+Width	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 Length（2 字节）:文本框限制框长度 Width（2 字节）:文本框限制框宽度 驱动函数：DisText_Region(uint16 length,uint16 width);
设置过滤颜色	0x44	FillColor_Down + FillColor_UP	FillColor_Dwon(2 字节):滤除颜色的下限值 FillColor_UP(2 字节)： 滤除颜色的上限值 备注：当两者相同时，只滤除一种颜色 驱 动 函 数 ： void SetFilterColor(uint16 fillcolor_down, uint16 fillcolor_up);
文本显示	0x20	*X+Y*+Back+Font+Strings	文本显示设置 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（背景色，1 字节） 0x01:背景色显示 0x00：背景色不显示 Font（字库编码,1 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） 备注：文字字体颜色与前景色一致，底色为背景色 驱动函数：void DisText(uint16 x, uint16 y,uint8 back,uint8 font,uchar *strings);
光标显示	0x21	Enable+X+Y+Length+Width	光标开闭 Enable(1 字节) 0x00：关闭 0x01：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 Length（1 字节）:光标长度 Width （1 字节）：光标宽度 驱动函数：void DisCursor(uint8 enable,uint16 x, uint16 y,uint16 length,uint16 width);
图片显示	0x31	Image_ID+MaskEn	全屏整幅图显示 Image _ID（图片编号 ,2 字节） MaskEn(1 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，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。
			驱动函数： void DisFull_Image(uint16 image_id,uint8 masken);
	0x32	X+Y+ Image_ID+MaskEn	区域图片显示 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 (图片编号,2 字节) MaskEn(1 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
			驱动函数： void DisArea_Image(uint16x,uint16y,uint16 image_id,uint8 masken);
图片剪切	0x33	X+Y+Image_ID+Image_X+Image_Y+Image_L+Image_W+MaskEn	指定位置显示剪切后的图片 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): 图片编号 Image_X(2 字节): 图片内部 X 坐标 Image_Y(2 字节): 图片内部 Y 坐标 Image_L(2 字节): 剪切长度 Image_W(2 字节): 剪切宽度 MaskEn(1 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
			驱动函数： void DisCut_Image(uint16 x,uint16 y, uint16 image_id, uint16 image_x, uint16 image_y, uint16 image_l, uint16 image_w, uint8 masken);
动画显示	0x80	X+Y+FlashImgae_ID+Enable+Playnum	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) : 图片编号 Enable(1 字节) 0x00：关闭动画播放; 0x01: 开启动画播放 PlayNum(1 字节) 0x00：重复播放; 0x01~0xFF：播放指定次数后停止 备注：播放停止后，HMI 上传 EE 02 FF FC FF FF 表示动画播放结束，动画格式只支持*gif 格式。一个画面只支持一个动画播放，不支持透明的 gif 格式播放

			驱动函数：void DisFlashImgae(uint16 x,uint16 y,uint16 flashimage_id, uint8 enable, uint8 playnum);
画点	0x50	X+Y	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 备注：颜色值取决于前景色调色板设置
			驱动函数：void GUI_Dot(uint16 x,uint16 y);
画线	0x51	X_0 +Y_0+X_1+Y_1	X_0（2 字节）:以点为单位的直线 X 轴起点坐标值 Y_0（2 字节）:以点为单位的直线 Y 轴起点坐标值 X_1（2 字节）:以点为单位的直线 X 轴终点坐标值 Y_1（2 字节）:以点为单位的直线 Y 轴终点坐标值 备注：颜色值取决于前景色调色板设置
			驱动函数：void GUI_Line(uint16 x0, uint16 y0, uint16 x1, uint16 y1);
绘制折线	0x63	Mode+(X,Y)$_0$+(X,Y)$_1$…+(X,Y)n	Mode(1 字节): 0x00 连接线颜色为前景色; 0x01 连接线颜色为背景色 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 备注：将指定的多个坐标点连接起来
			驱动函数：void GUI_ConDots(uint8 mode,uint8 *pDot);
画空心圆	0x52	X_0+Y_0+R	X_0（2 字节）:以点为单位的圆心 X 坐标值 Y_0（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 备注：颜色值取决于前景色调色板设置
			驱动函数：void GUI_Circle(uint16 x0, uint16 y0, uint16 r);
画实心圆	0x53	X_0+Y_0+R	X_0（2 字节）:以点为单位的圆心 X 坐标值 Y_0（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 备注：颜色值取决于前景色调色板设置
			驱动函数：void GUI_CircleFill(uint16 x0, uint16 y0, uint16 r);
画圆弧	0x67	X_0+Y_0+R +SA+EA	X0（2 字节）:以点为单位的圆心 X 坐标值 Y0（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 备注：钟表 15 刻钟方向为 0 度，顺时针方向计算；颜色值取决于前景色调色板设置
			驱动函数：void GUI_Arc (uint16 x0,uint16 y0, uint16 r,uint16 sa, uint16 ea);
画空心矩形	0x54	X_0+Y_0+X_1+Y_1	X_0(2 字节）:以点为单位的空心矩形左上角 X 坐标值

			Y_0 (2 字节) :以点为单位的空心矩形左上角 Y 坐标值 X_1 (2 字节) :以点为单位的空心矩形右下角 X 坐标值 Y_1 (2 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调色板设置
			驱动函数：void GUI_Rectangle(uint16 x0, uint16 y0, uint16 x1,uint16 y1);
画实心矩形	0x55	$X_0+Y_0+X_1+Y_1$	X_0(2 字节) :以点为单位的实心矩形左上角 X 坐标值 Y_0 (2 字节) :以点为单位的实心矩形左上角 Y 坐标值 X_1 (2 字节) :以点为单位的实心矩形右下角 X 坐标值 Y_1 (2 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调色板设置
			驱动函数：void GUI_RectangleFill(uint16 x0, uint16 y0, uint16 x1,uint16 y1);
画空心椭圆	0x56	$X_0+Y_0+X_1+Y_1$	X0(2 字节):以点为单位的空心椭圆最左端 X 坐标值 Y0(2 字节):以点为单位的空心椭圆最上端 Y 坐标值 X1(2 字节):以点为单位的空心椭圆最右端 X 坐标值 Y1(2 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调色板设置
			驱动函数：void GUI_Ellipse (uint16 x0, uint16 y0, uint16 x1,uint16 y1);
画实心椭圆	0x57	$X_0+Y_0+X_1+Y_1$	X0(2 字节):以点为单位的实心椭圆最左端 X 坐标值 Y0(2 字节):以点为单位的实心椭圆最上端 Y 坐标值 X1(2 字节):以点为单位的实心椭圆最右端 X 坐标值 Y1(2 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调色板设置
			驱动函数：void GUI_EllipseFill (uint16 x0, uint16 y0, uint16 x1,uint16 y1);
多边形填充	0x64	X+Y+Color	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 (X,Y)代表区域填充的种子点位置 Color(2 字节):填充颜色 备注：用于凸多边形的填充，填充区域的颜色不能与边界一致
			驱动函数：void GUI_PolygonFill(uint16 x,uint16 y, uint16 color);
背光调节	0x60	Light_level	255 级别背光亮度调节； 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 字节)：鸣叫时间长度为 10ms*Time
			驱动函数： void SetBuzzer(uint8 Time);
注意：5.6 寸、7 寸、8 寸可选配 RTC。所有尺寸产品均可选配触摸屏。如果需要用到这些功能，请在采购时向我们的销售人员说明。			
触摸屏控制	0x70	Cmd	Cmd(1 字节),Bit0~Bit3 功能如下：

			Bit0：1 表示触摸屏打开，0 表示触摸屏关闭； Bit1：1 表示蜂鸣器自动响， 0 表示不响； BIT3~BIT2 ： 00：表示触摸屏被按下时立刻上传坐标 01：表示触摸屏被按下直至释放后才上传坐标 10：触摸屏按下时每 100ms 上传一次坐标，直到触摸屏释放 11： 表示触摸屏被按下和松开时均上传一次坐标 触摸坐标上传格式： 触摸屏被按下时 上传格式:EE 01 X 坐标 Y 坐标 FF FC FF FF 触摸屏松开时 上传格式 EE 03 X 坐标 Y 坐标 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)
触摸屏体验	0x72	无	触摸屏校准
			驱动函数：void SetTouchScreen_Adj (void)
	0x73	无	用户按下触摸后，屏对应坐标处显示一个红色实心圆，测试用
			驱动函数：void TestTouchScreen (void)
RTC 时钟设置	0x81	Sec+Min+Hour+Day +Week+Mon+Year	时钟参数设定 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 备注：各 1 个字节，以 BCD 码表示，星期天设置为 0x00
			驱动函数:SetRTCTime(uint8 sec,uint8 min,uint8 hour,uint8 day,uint8 week,uint8 mon,uint year)
读取 RTC 时钟	0x82	无	数据输出格式：帧头+ 0xF7+Year +Mon +Week +Day +Hour +Min + Sec + 帧尾 备注：各 1 个字节，以 BCD 码表示
			驱动函数：ReadRTCTime(void)
RTC 显示设置	0x85	Enable+Dis_Mode+Text_Mode +Color +Xpoint+Ypoint	RTC 显示设置 Enable(1 字节) 0x00：RTC 关闭 0x01：RTC 开启 Dis_Mode(1 字节) 0x00 :格式 HH:MM:SS 0x01 :格式 20XX-MM-DD HH:MM:SS Text_Mode(1 字节) 0x00： 0 号字体 0x01： 1 号字体 0x02： 2 号字体 0x03： 3 号字体

			Color (2 字节) : 显示颜色 Xpoint(2 字节) : 显示 X 坐标 Ypoint(2 字节): 显示 Y 坐标 驱动函数:SetRTCEn(uint8 enble, uint8 Dis_Mode, uint8 Text_Mode , uint16 Color, uint16 Xpoint, uint16 Ypoint);
设置波特率	0xA0	Baudset	Baudset(单位 bps,1 字节), 波特率编序: 0x00: 1200 0x01: 2400 0x02: 4800 0x03: 9600 0x04: 19200 0x05: 38400 0x06: 57600 0x07: 115200 驱动函数：void SetCommBps(uint8 Baudset);
写图层操作	0xA1	Layer	Layer(1 字节)：写入的图层 0x00: 写入图层 0（显存 0） 0x01: 写入图层 1（显存 1） 0x02: 写入图层 2（显存 2） 0x03: 写入图层 3（显存 3） 注：4.3 寸和 3.5 寸屏仅支持双图层 驱动函数：void WriteLayer(uint8 layer)
切换图层显示	0xA2	Layer	Layer(1 字节)：当前显示的图层 0x00: 切换图层 0 显示 0x01: 切换图层 1 显示 0x02: 切换图层 2 显示 0x03: 切换图层 3 显示 注：4.3 寸和 3.5 寸屏仅支持双图层 驱动函数：void DisplyLayer(uint8 layer)
旋转屏幕	0x74	Cmd	Cmd ：旋转方向 0x00 正常屏幕方向 0x01 上下翻转 0x02 左右翻转 0x03 180 度旋转

5.4 指令使用详解

5.4.1 握手命令

用户发送：EE【00】FF FC FF FF

设备返回： 55

握手命令主要用于判读 HMI 是否上电初始化完毕，通信是否正常。用户发送指令参数后，HMI 返回 55 命令，表示连机成功。实际使用中，该命令可以忽略。

5.4.2 握手命令 1

用户发送：EE【04】FF FC FF FF

设备返回： EE 55 FF FC FF FF

5.4.3 设置前/背景色

命令格式： 设置前景色：EE【41 Fcolor】FF FC FF FF

设置背景色：EE【42 Bcolor】FF FC FF FF

参数说明：Fcolor、Bcolor (2 个字节)分别是前景色和背景色 RGB 数值。

前景色用于指定点、线、圆、图形和文字的颜色，背景色用于刷屏和文字背景色颜色指定，如图 5.1 所示。

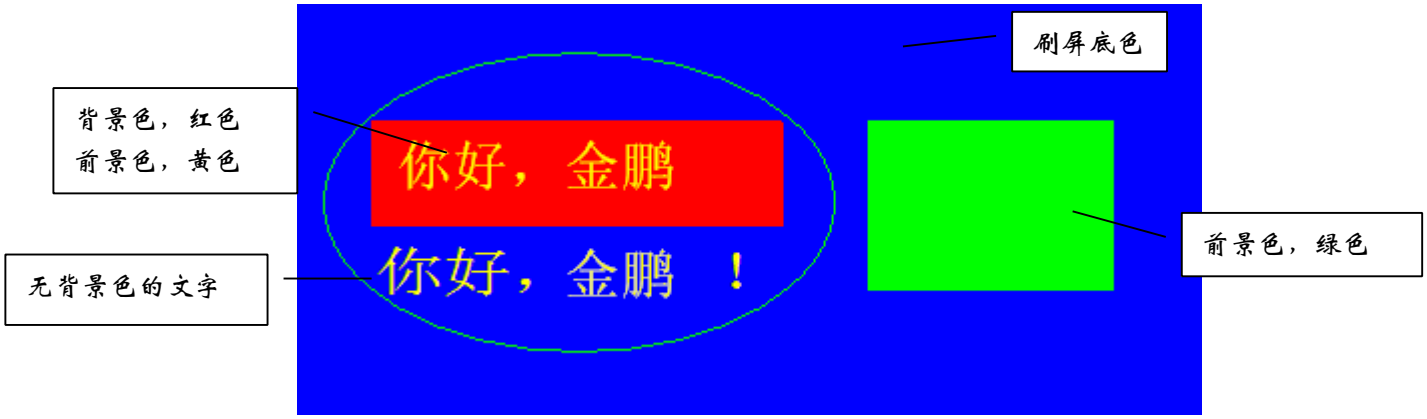


图 5.1 背景前景色说明

5.4.4 取屏幕点设置前景色/背景色

命令格式：EE【A3 Mode X Y】FF FC FF FF

参数说明：Mode(1 字节) ：

0x00：取当前显示屏幕(X,Y)处颜色作为前景色

0x01：取当前显示屏幕(X,Y)处颜色作为背景色

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；

该指令主要用于提取当前屏幕某点处的像素(RGB)值作为前景色或背景色。例如用户如需获取图 6.3 所示坐标(100,50)处的像素值作为背景色，参考程序如下所示。

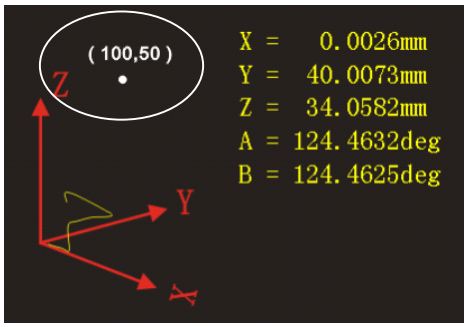


图 5.2 提取屏幕某点像素值

参考程序：

```
{
 ColorPicker(1, 100,50);                    //获取屏幕(100,50)像素值作为背景色；
 DisText(50, 50, 1, 6, "你好，金鹏！"); //在(50,50)写入字符串，文本背景色为上述获取值
}
```

5.4.5 清屏

命令格式：EE【01】FF FC FF FF

参数说明：无参数

该命令用于实现指定颜色清屏，清屏的颜色取决于背景色调色板的设置。若用户未进行背景色设置，清屏默认为蓝色。

5.4.6 设置文字行间距和列间距

命令格式：EE【43 X_W Y_W】FF FC FF FF

参数说明：X_W(1 字节) :以点为单位的行间距，取值范围 00~1F；

Y_W(1 字节) :以点为单位的列间距，取值范围 00~1F。

该命令用于设置文字之间的行列距，方便自动换行显示。

5.4.7 设置文本框限制框

命令格式：EE【45 Enable Length Width】FF FC FF FF

参数说明： Enable(1 字节)：打开/关闭文本框限制使能

Length (2 字节): 文本框长度；

Width (2 字节): 文本框的高度。

当设置了文本框的限制后，文字在限定框里面会自动换行，超出限定框的文本不会被显示。

5.4.8 设置过滤颜色

命令格式：EE【44 FillColor_Down FillColor_UP】FF FC FF FF

参数说明： FillColor_Down (2 字节): 过滤颜色的下限值，即最小值；

FillColor_UP (2 字节): 过滤颜色的上限值，即最大值。

该命令主要实现指定某一范围颜色的过滤，当两者相同时，只滤除一种颜色。

过滤原理：设定过滤上限和下限值后，当采集到图片中的某一像素值正好满足过滤范围，该点就会屏蔽禁止写入显存。设置过滤色前后对比图如 5.3 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, 65535);           //设置过滤值 65535，白色 RGB 值为 65535
   DisArea_Image(258,68, 2, 1);            // (258,68)处显示过滤后的蝴蝶图案
}
```

图 5.3 设置过滤色前后对比图

5.4.9 文本显示

命令格式：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（背景色，1 字节）

0x01:背景色显示 0x00：背景色不显示

Font（字库编码,1 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。

该命令用于实现在屏幕任意位置显示指定的文本。汉字字库有 GBK 和 GB2312，ASCII 字库不能显示汉字。在实际操作中，用户确定文本的前景色、背景色、字库编码后，直接输入字符串即可，设备将会自动换行或字符与汉字匹配切换显示。

5.4.10 光标显示

命令格式： EE【21 Enable X Y Length Width】FF FC FF FF

参数说明： Enable(1 字节) ： 0x00：关闭 ， 0x01：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

Length (1 字节) :光标长度

Width (1 字节) :光标宽度

用户通过该命令控制光标的闪烁和关闭。例如用户在 24*24 的汉字尾缀显示 1 个长度 16、宽度 8 的光标，参考程序和效果图如下所示。

程序参考代码：

```
{
  SetBcolor(31);                  // 设置蓝色背景色
  GUI_CleanScreen( );             // 背景清屏蓝色
  SetFcolor(65516);               // 设置文字前景色
  DisText(23, 24, 0, 6, "为客户提供免费的美工服务");   //14 个 24*24 大小汉字
  DisCursor(1,359,40,16,8);   //光标闪烁使能，在(359,40)处显示长度 16 宽度 8 的光标
}
```

图 5.4 光标参数说明

5.4.11 图片显示

1. 全屏整幅图显示

命令格式：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)

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，下载的图片分辨率不能超过当前屏幕的分辨率，否则无法去显示

2. 区域图片显示

命令格式： 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 字节) ： 图片编号

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

该命令用于实现图片任意位置显示，图片位置不能超越屏幕边界。

5.4.12 图片剪切

命令格式： EE【33 X Y Image_ID Image_X Image_Y Image_L Image_W 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)：图片编号

Image_X (2 字节)： 图片内部 X 坐标

Image_Y (2 字节)： 图片内部 Y 坐标

Image_L (2 字节) ： 剪切长度

Image_W（2 字节）：剪切宽度

MaskEn（1 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

该命令用于实现在指定的坐标处显示被剪切的图片。用户可以对储存在 Flash 中的任意图片进行裁剪。例如实现电梯 0-60 的楼层显示，用户只需对图 6.8 所示的相应数字进行剪切组合，同时过滤掉蓝色背景。除此之外，剪切命令还在数据采集、图片移动、进度条和仪表显示等场合使用非常广泛。例如实现一只小鸟在画面中飞行，在实际操作过程中，小鸟每飞行到一个新的坐标位置，用户必须剪切原始背景图来更新前一个坐标位置的背景。一般来讲，局部剪切与全屏更新相比，耗时更短，画面显示更流畅。

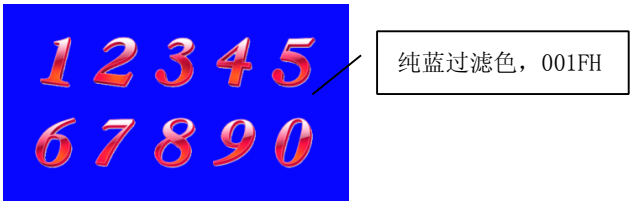


图 5.5 数字图片

图片剪切的各参数定义如图 6.9 描述所示,图片内部坐标(X,Y)的原点为其左上角。

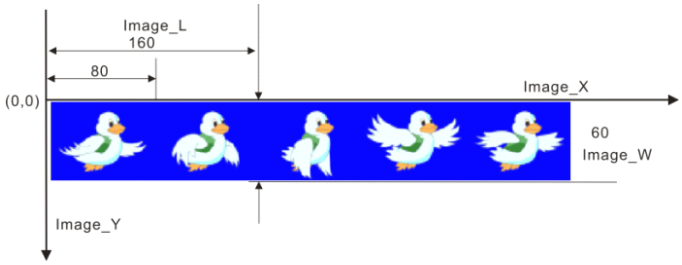


图 5.6 图片剪切参数说明

5.4.13 动画显示

命令格式：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 字节)

0x00：关闭动画播放；　0x01：开启动画播放

PlayNum(1 字节)

0x00：重复播放；　0x01~0xFF：播放指定次数后停止

备注：播放停止后，HMI 返回 EE 02 FF FC FF FF 表示动画播放结束，动画只支持*gif 格式，不支持两个以上的 GIF 动画同时播放，不支持透明的 GIF 播放。

该命令用于实现任意位置*gif 动画的显示，其它动画格式暂不支持。

5.4.14 画点

命令格式：　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

该命令主要实现在屏幕的任意位置画点操作，点的颜色值取决前景色的设置。

5.4.15 画线

命令格式： EE【 51 X_0 Y_0 X_1 Y_1 】FF FC FF FF

参数说明： X_0 (2 字节) :以点为单位的直线 X 轴起点坐标值

Y_0 (2 字节) :以点为单位的直线 Y 轴起点坐标值

X_1 (2 字节) :以点为单位的直线 X 轴终点坐标值

Y_1 (2 字节) :以点为单位的直线 Y 轴终点坐标值

该命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 6.4 所示。

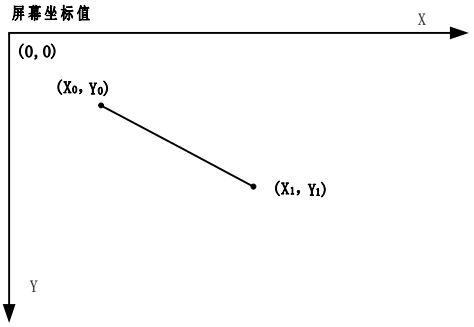


图 5.7 画线参数说明

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 6.13

程序参考代码：

```
{
  SetBcolor(31);                    //设置背景色蓝色
  GUI_CleanScreen( );               //清屏
  SetFcolor(65523);                 //设置线的前景色为黄色
  GUI_Rectangle(20, 10, 420, 70);   //画空心矩形，左上角坐标(20,10),右下角(420,70)
  GUI_Line(20, 30, 420, 30);        //画从(20,30)到(420,30)的直线
  GUI_Line(20, 50, 420, 50);        //画从(20,50)到(420,50)的直线
  GUI_Line(61, 10, 61, 70);         //画从(61,10)到(61,70)的直线
  GUI_Line(211, 10, 211, 70);       //画从(211,10)到(211,70)的直线
  DisText(30, 13, 0, 4, "编号");    //写入 12*12 汉字
  DisText(38, 32, 0, 4, "1");       //写入 12*12 字符
  DisText(38, 52, 0, 4, "2");       //写入 12*12 字符
}
```

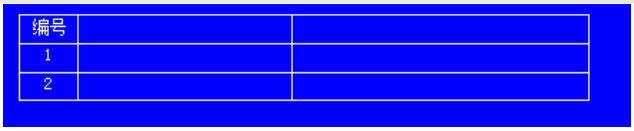


图 5.8 画线显示效果图

5.4.16 绘制折线

命令格式： EE【 63 Mode X_0 Y_0 X_1 Y_1 … X_n Y_n】FF FC FF FF

参数说明： Mode(1 字节)

0x00 连接线颜色为前景色； 0x01 连接线颜色为背景色

X_n (2 字节) :以点为单位的直线 X 轴起点坐标值

Y_n (2 字节) :以点为单位的直线 Y 轴起点坐标值

该命令主要实现将指定的多个坐标点连接起来。例如要实现如图 6.14 所示的折线和六边形形状，程序如下所示。

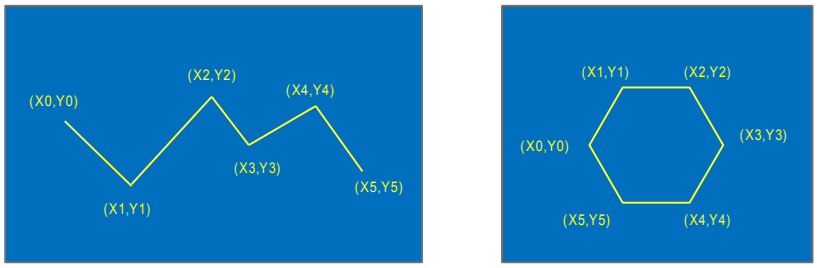


图 5.9 画折线效果图

程序参考代码：

```
unsigned short usPointXY0[] ={ x0,y0,x1,y1,x2,y2,x3,y3,x4,y4,x5,y5}
unsigned short usPointXY1[] ={ x0,y0,x1,y1,x2,y2,x3,y3,x4,y4,x5,y5,x0,y0}
{
  SetBcolor(31);                    //设置背景色蓝色
  GUI_CleanScreen( );               //清屏
  SetFcolor(65523);                 //设置线的前景色为黄色
  GUI_ConDots(0, usPointXY0);       //绘制(x0,y0)到(x5,y5)的折线
  GUI_ConDots(0, usPointXY1); //绘制六边形，首尾相连
}
```

注意：每帧最大的数目是 1024 字节(包含帧头和帧尾)，因此一次所包含的这点数不能超 254.

5.4.17 画空心圆

命令格式： EE【 52 X_0 Y_0 R 】FF FC FF FF

参数说明： X_0 (2 字节) : 以点为单位的圆心 X 坐标值

Y_0 (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

该命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。参数定于如图 6.5 所示。

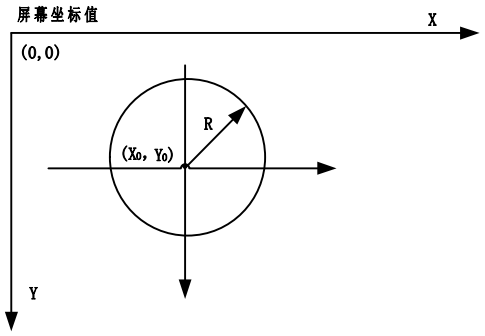


图 5.10 画空心圆参数说明

5.4.18 画实心圆

命令格式： EE【 53 X_0 Y_0 R 】FF FC FF FF

参数说明： X_0 (2 字节)：以点为单位的圆心 X 坐标值

Y_0 (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

该命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。

5.4.19 画圆弧

命令格式： EE【 67 X_0 Y_0 R SA EA】FF FC FF FF

参数说明： X_0 (2 字节)：以点为单位的圆心 X 坐标值

Y_0 (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

该命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。圆弧起始角度参考坐标如下 6.16 所示。

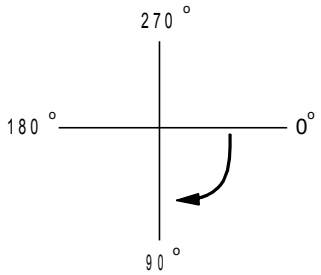


图 5.11 圆弧起始角度参考图

5.4.20 画空心矩形

命令格式： EE【 54 X_0 Y_0 X_1 Y_1 】FF FC FF FF

参数说明： X_0 (2 字节)：以点为单位的空心矩形左上角 X 坐标值

Y_0 (2 字节)：以点为单位的空心矩形左上角 Y 坐标值

X_1 (2 字节)：以点为单位的空心矩形左上角 X 坐标值

Y_1 (2 字节)：以点为单位的空心矩形左上角 Y 坐标值

该命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。参数定于如图 6.6 所示。

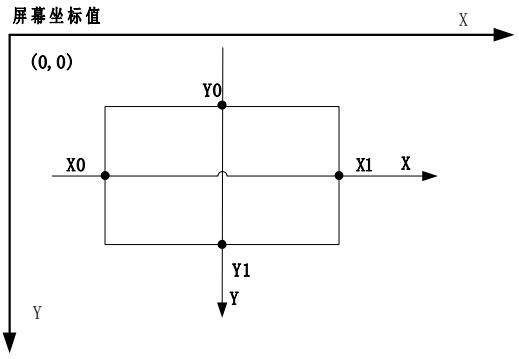


图 5.12 画空心矩形参数说明

5.4.21 画实心矩形

命令格式： EE【 55 X_0 Y_0 X_1 Y_1 】FF FC FF FF

参数说明： X_0 (2 字节) :以点为单位的实心矩形左上角 X 坐标值

Y_0 (2 字节) :以点为单位的实心矩形左上角 Y 坐标值

X_1 (2 字节) :以点为单位的实心矩形左上角 X 坐标值

Y_1 (2 字节) :以点为单位的实心矩形左上角 Y 坐标值

该命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。

5.4.22 画空心椭圆

命令格式：EE【 56 X_0 Y_0 X_1 Y_1 】FF FC FF FF

参数说明： X_0 (2 字节) :以点为单位的空心椭圆最左端 X 坐标值

Y_0 (2 字节) :以点为单位的空心椭圆最上端 Y 坐标值

X_1 (2 字节) :以点为单位的空心椭圆最右端 X 坐标值

Y_1 (2 字节) :以点为单位的空心椭圆最下端 Y 坐标值

该命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 6.7 所示。

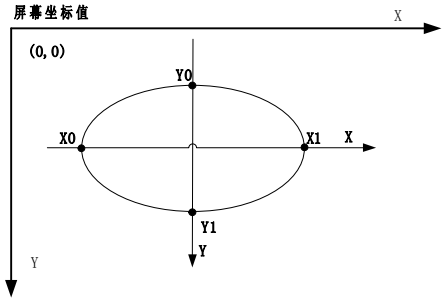


图 5.13 画空心椭圆参数说明

5.4.23 画实心椭圆

命令格式：EE【 57 X_0 Y_0 X_1 Y_1 】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

该命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。

5.4.24 区域填充

命令格式：EE【 64 X Y Color 】FF FC FF FF

参数说明： X (2 字节) :以点为单位的实心椭圆最左端 X 坐标值

Y (2 字节) :以点为单位的实心椭圆最上端 Y 坐标值

(X,Y)代表填充区域中任一个种子位置。

Color(2 字节) ： 填充的颜色值

该指令主要用于凸多边形区域的填充。通常该指令需要与绘制折线指令配合使用，而且

被填充的颜色不能与边界颜色值相同。例如用户要实现如图 6.19 所示指针图，分如下步骤操作：(1) 绘制多边形; (2)确定种子位置; (3)指定填充区域的颜色值

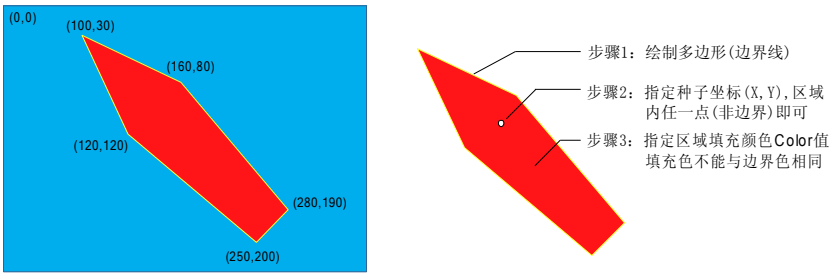


图 5.14 区域填充示意图

程序参考代码：

```
{
  SetBcolor(1662);                    // 设置淡蓝背景色
  GUI_CleanScreen( );                 // 清屏
  SetFcolor(65523);                   // 设置多边形的前景色为黄色
  GUI_ConDots(0,100,30,160,80,280,190,250,200,120,120,100,30); //绘制多边形
  GUI_AreaFill(110,100,63488);        // 执行区域填充，填充色红色
}
```

5.4.25 背光调节

命令格式： EE【 60 Light_level 】FF FC FF FF

参数说明： Light_level (1 字节) : 255 级背光调节

该命令主要用于液晶背光亮度的调节，取值范围 00H～FF H。00H 表示背光最亮，FFH 表示背光关闭。

5.4.26 蜂鸣器控制

命令格式： EE【 61 Time 】FF FC FF FF

参数说明： Enable (1 字节)

0x00：关闭蜂鸣 ，0x01：开启蜂鸣

Time (1 字节)：鸣叫时间长度为 10ms*Time。

该命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。蜂鸣器大多场合是配合触摸屏使用，当有触摸动作时，蜂鸣器讯响一声，此时建议 Time 值设为 3-5。

5.4.27 触摸屏控制

命令格式： EE【 70 Cmd 】FF FC FF FF

参数说明： Cmd (1 字节)

BIT3-BIT2 ：

00：表示触摸屏被按下时立刻上传坐标

01：表示触摸屏被按下直至释放后才上传坐标

10：触摸屏按下时每 100ms 上传一次坐标，直到触摸屏释放

11： 表示触摸屏被按下和松开时均上传一次坐标

该命令包含了触摸使能、开闭蜂鸣器和数据上传方式。

触摸屏被按下时的上传格式：EE 01 X 坐标 Y 坐标 FF FC FF FF

触摸屏松开时的上传格式 EE 03 X 坐标 Y 坐标 FF FC FF FF

X 坐标 Y 坐标均为 2 个字节，高字节在前例如，用户按下屏幕(50,100)的位置后，HMI 对外串口输出数组：EE 01 【00 32 00 64】 FF FC FF FF。用户 MCU 通过判断坐标值大小，即可确定当前触摸点范围。HMI 内部针对触摸点进行了多次采样和运算，用户 MCU 无需再进行第二次运算。

5.4.28 触摸屏校准

命令格式： EE【 72 】FF FC FF FF

参数说明： 无

该命令用于触摸屏的校准。所有产品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 5.14 所示，点击 3 次不同参考点后，设备将提示校准成功，否则需要重新校准。用户也可以通过我们提供的 PC 软件进行触摸重新校准。校准完成后，屏会返回 EE 【04】 FF FC FF FF。

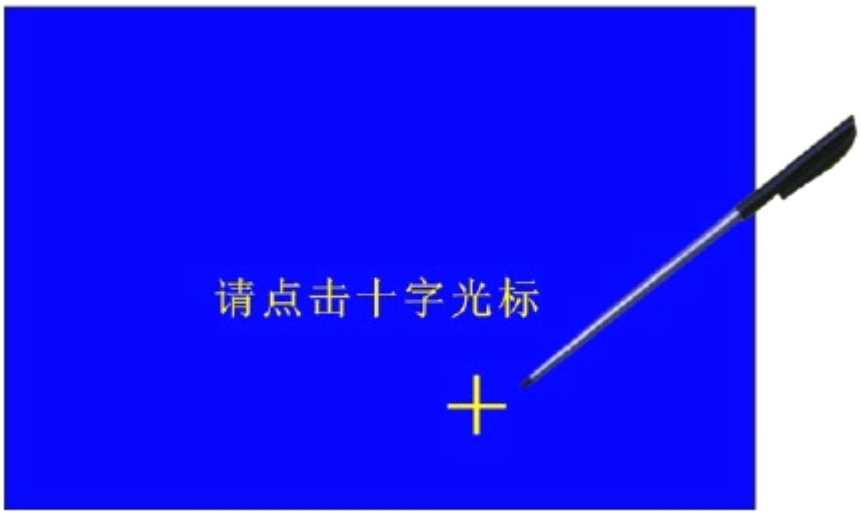


图 5.14 触摸屏校准图

5.4.29 触摸屏体验

命令格式： EE【 73 】FF FC FF FF

参数说明： 无

该命令属于测试命令。如图 6.22 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。HMI 与 PC 串口联机后，用户可以点击软件工具栏的“体验触摸”来感受触摸的方便和精准。

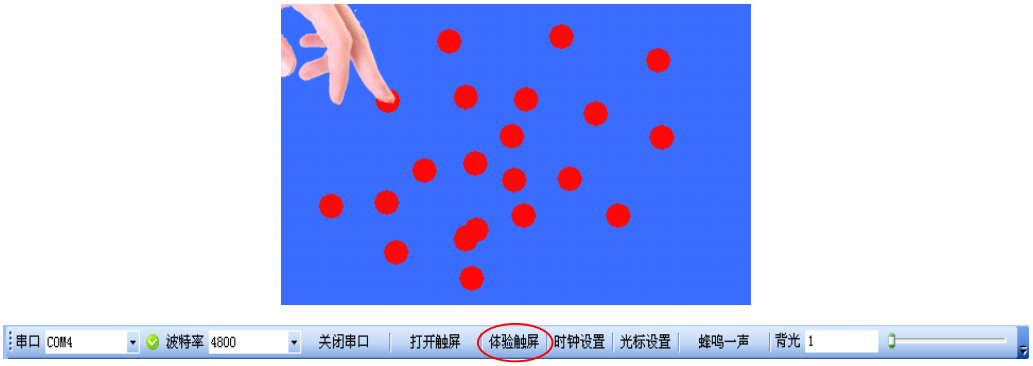


图 5.15 触摸体验效果图

5.4.30 RTC 属性设定

命令格式： EE【 85 Enable DisMode TextMode Color Xpoint Ypoint 】FF FC FF FF

参数说明： Enable (1 字节)

0x00： RTC 关闭， 0x01： RTC 使能

DisMode(1 字节)

0x00 :显示格式 HH:MM:SS

0x01 :显示格式 20XX-MM-DD HH:MM:SS

TextMode(1 字节)

0x00: 0 号字体 0x01: 1 号字体

0x02: 2 号字体 0x03: 3 号字体

Color (2 字节) : 显示颜色

Xpoint (2 字节): 以点为单位的 X 坐标值

Ypoint(2 字节) : 以点为单位的 Y 坐标值

该命令主要用于 RTC 时钟参数设定，通过设定对应的参数实现不同时钟格式、字体的显示。RTC 相关指令建议用户直接采用上位机软件进行设置，PC 设置参考界面如图 6.23 所示。

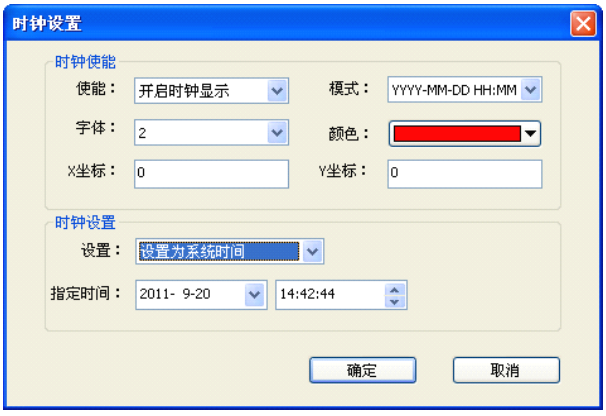


图 5.16 RTC 设置参考图

5.4.31 RTC 时钟设置

命令格式： 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
备注：各 1 个字节，以 BCD 码表示，星期天设置为 0x00

该命令用于当前时间的设定。

5.4.32 读取 RTC 时钟

命令格式： EE【 82 】FF FC FF FF
参数说明：无
该命令用于获取当前时间，数据输出格式：帧头+ 0xF7+Year +Mon +Week +Day +Hour +Min + Sec + 帧尾，各 1 个字节，以 BCD 码表示。

5.4.33 设置波特率

命令格式： EE【 A0 Baudset 】FF FC FF FF
参数说明： Baudset(1 字节)：波特率编序
0x00：1200bps　　0x01：2400
0x02：4800　　0x03：9600
0x04：19200　　0x05：38400
0x06：57600　　0x07：115200

该命令主要用于波特率的配置，范围为 1200-15200bps。新的波特率值存在 HMI 存储器中，并断电保存。用户也可以直接通过上位机的“工具→设置波特率”配置新的波特率，如图 6.24 所示。

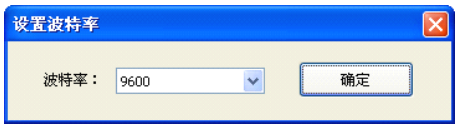


图 5.17 波特率设置图

5.4.34 写图层操作

命令格式： EE【 A1 Layer】FF FC FF FF
参数说明： Layer （1 字节）
0x00：写图层 0，即显存区 0　0x01：写图层 1，即显存区 1
0x02：写图层 2，即显存区 2　0x03：写图层 3，即显存区 3

这里图层的定义可以简单理解为显存区。常见的单色屏由于点阵数目较少，只有一个显存缓冲区，用户永远只能对当前的显存区进行操作，而真彩屏分辨高、数据传输量大，用户如果对当前显存区进行太多的操作，即便全部操作在较短时间内完成，由于画面色差过渡的缘故，人眼还是可以感察到略微的屏闪或刷屏动作。若需解决此问题，用户可调用写图层 1 指令，然后把下一帧显示的内容全部写入显存区 1，最后调用切换图层 1 显示指令，此时人眼将不会感察到任何闪屏或刷屏滞留，因为在写显存区 1 的过程中，屏幕一直会显示图层 0 的内容，直至接收到切换图层显示命令为止。默认图层 0 写入和显示。

程序参考代码：

```
{
    screen0( );          // 显示当画面 0 的内容
    WriteLayer(1);   // 设置写图层 1 操作
    screen1( );           // 写入画面 1 的内容
    DisplyLayer(1);   // 切换图层，显示当前画面 1 的内容
    …….
    /*****同样，用户也可以从图层 1 切换到图层 0 显示*****/
    WriteLayer(0);   // 设置写图层 0 操作
    screen0( );           // 写入画面 0 的内容
    DisplyLayer(0);   // 切换图层，显示当前画面 0 的内容
 }
```

5.4.35 切换图层显示

命令格式： EE【 A2 Layer】FF FC FF FF

参数说明： Layer（1 字节）

0x00：显示图层 0，即显存 0 00x01：显示图层 1，即显存 1

0x02：显示图层 2，即显存 2 00x03：显示图层 3，即显存 3

该命令主要用于选择当前显示的图层，通常与写图层操作前后呼应使用。

5.4.36 旋转屏幕：

命令格式： EE【74 cmd】FF FC FF FF

参数说明： Cmd（1 字节）

0x00 正常屏幕方向

0x01 上下翻转

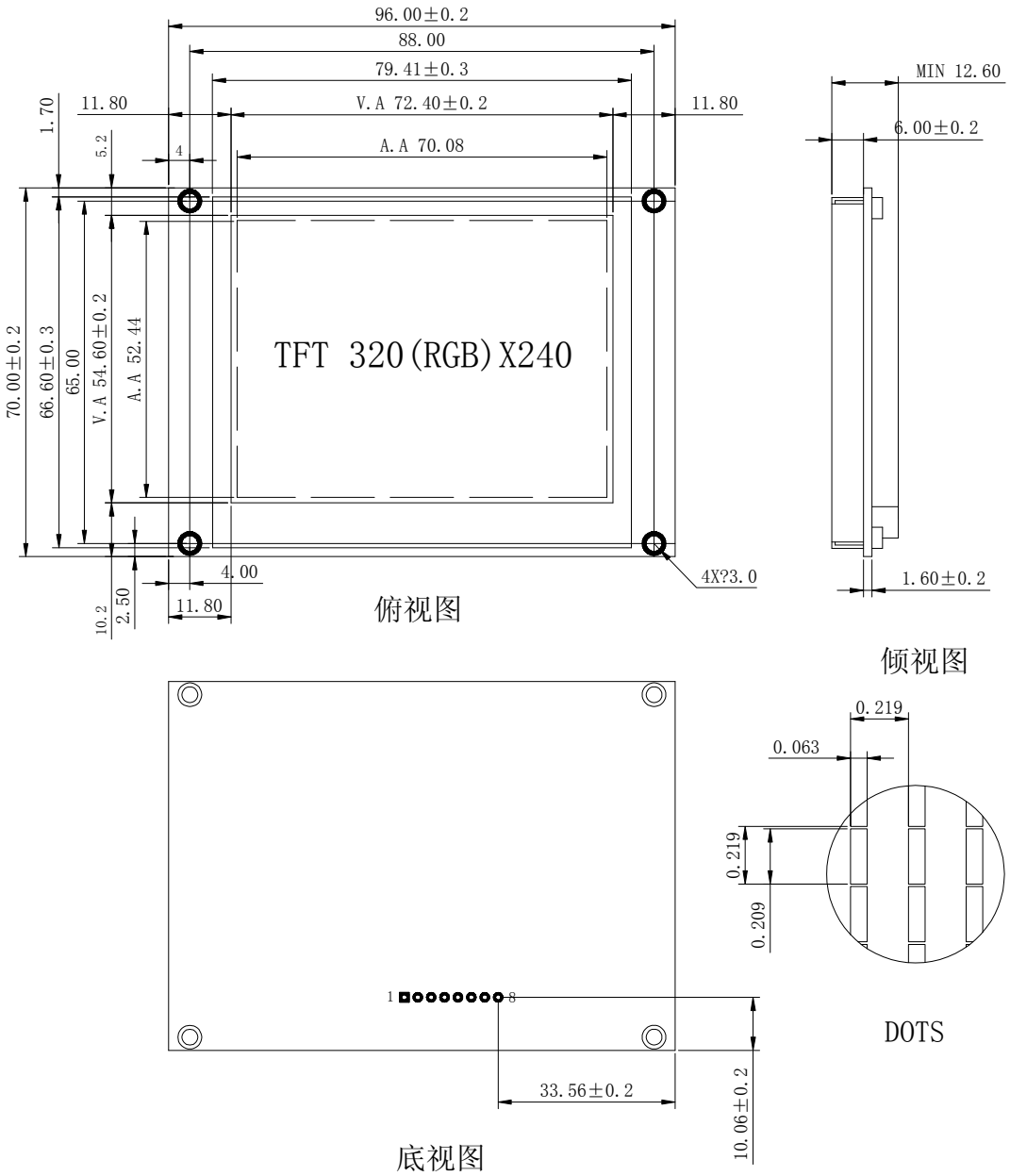
0x02 左右翻转

0x03 180 度旋转

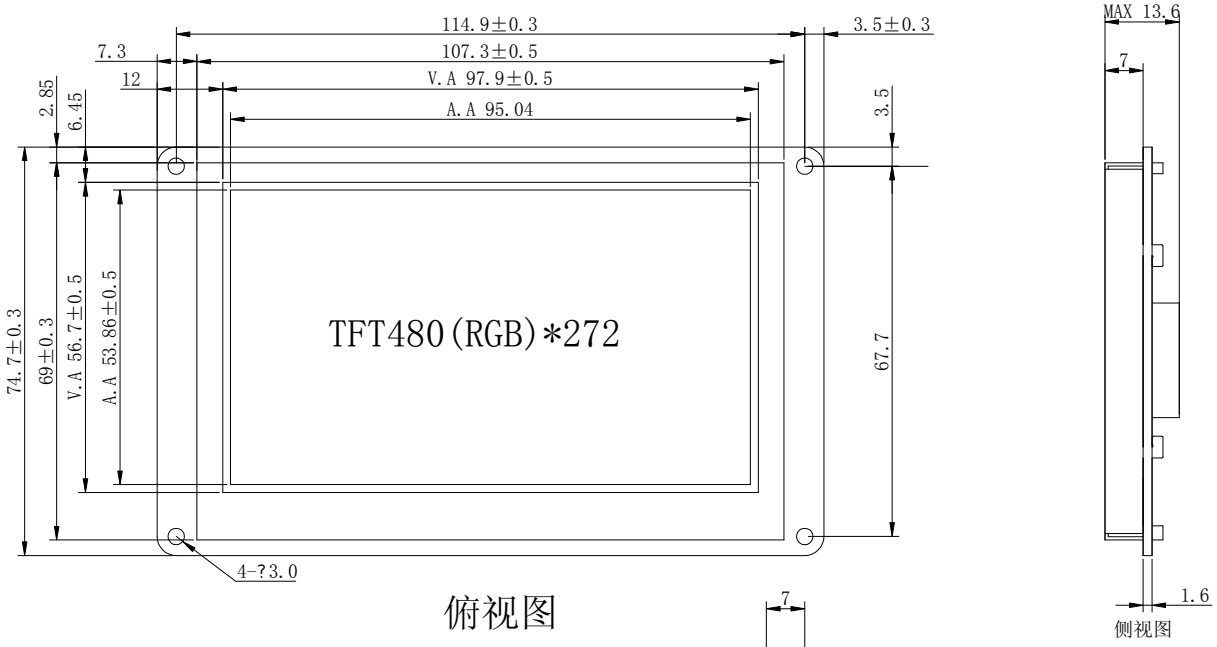
该命令用于控制屏幕旋转的方向

六、模块外型尺寸图

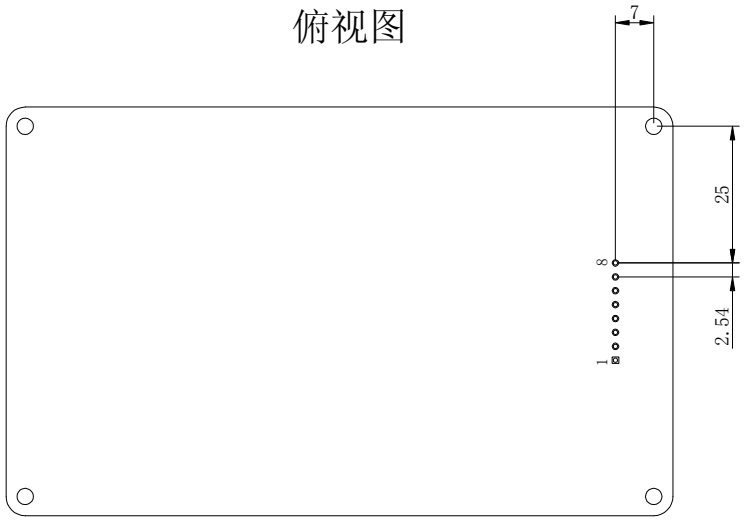
OCM320240T350-1C



OCM480272T430-1C

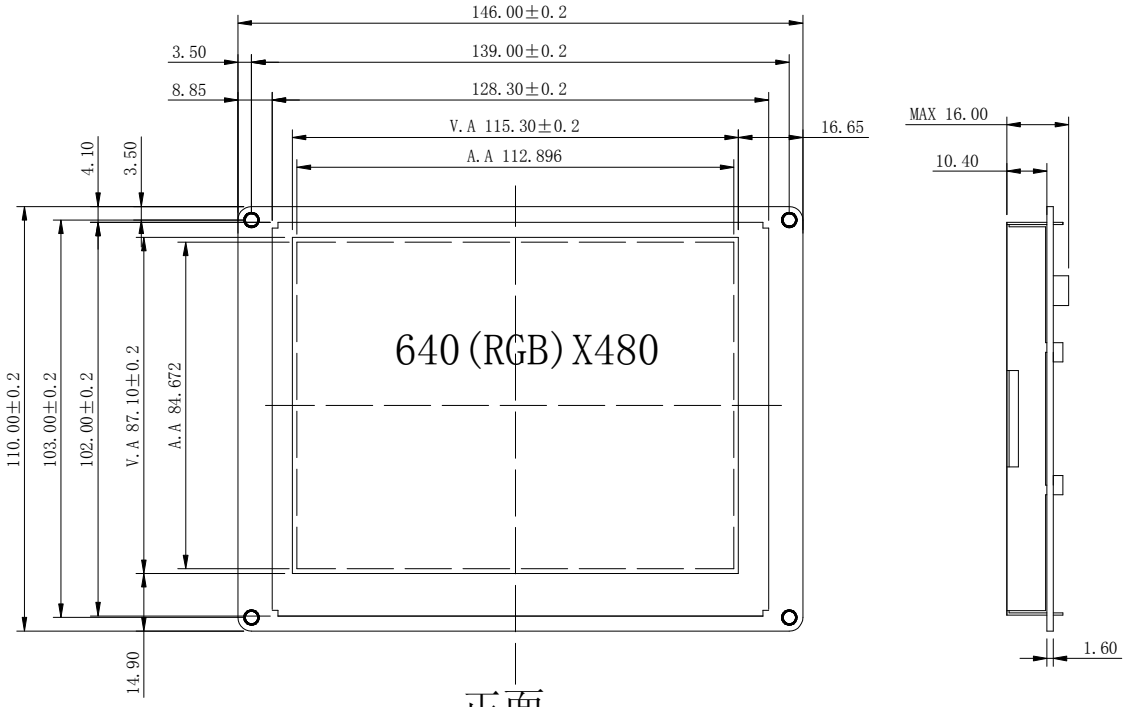


俯视图

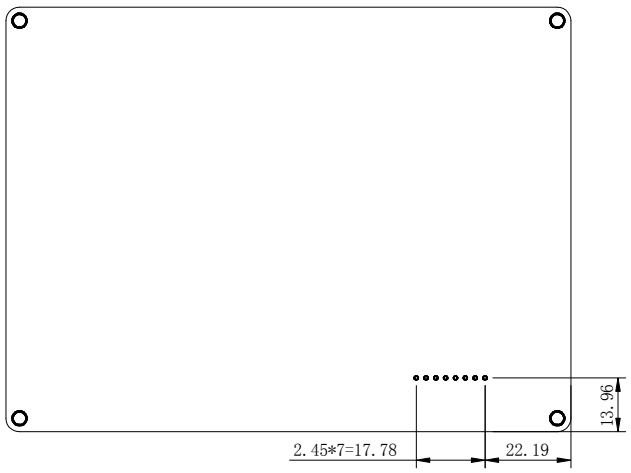


底视图

OCM640480T560-1C

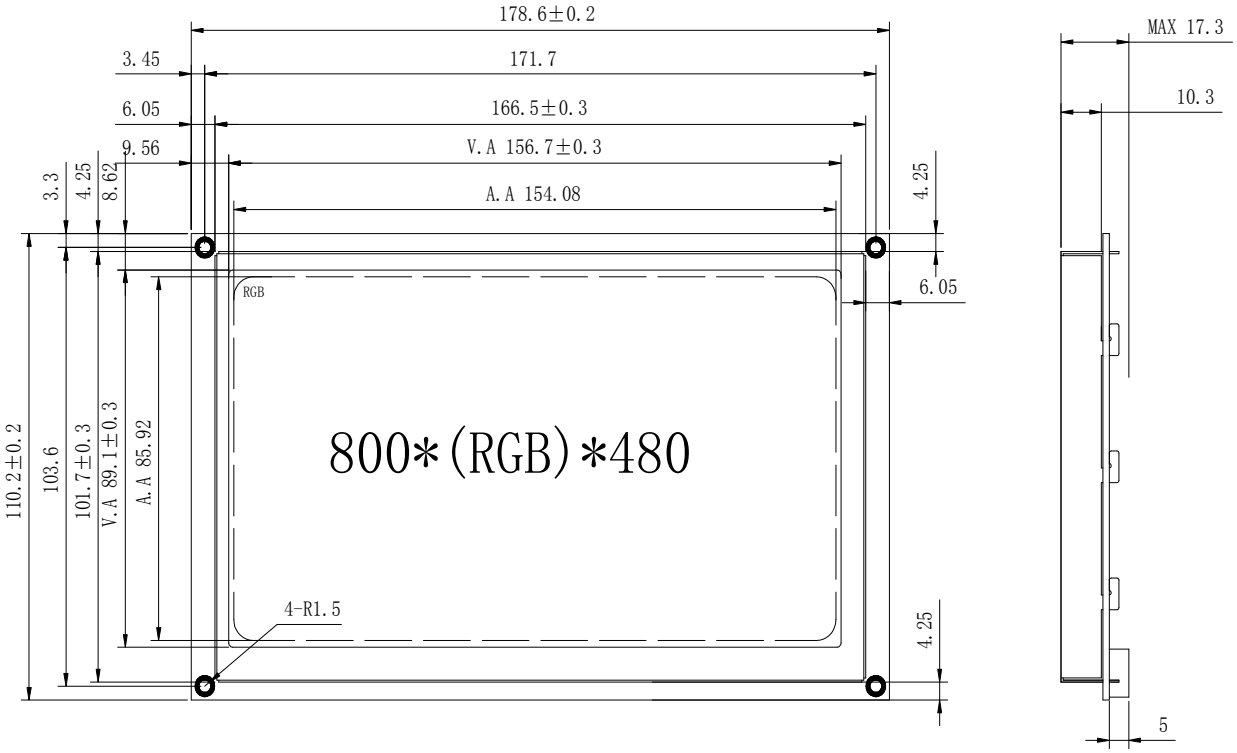


正面

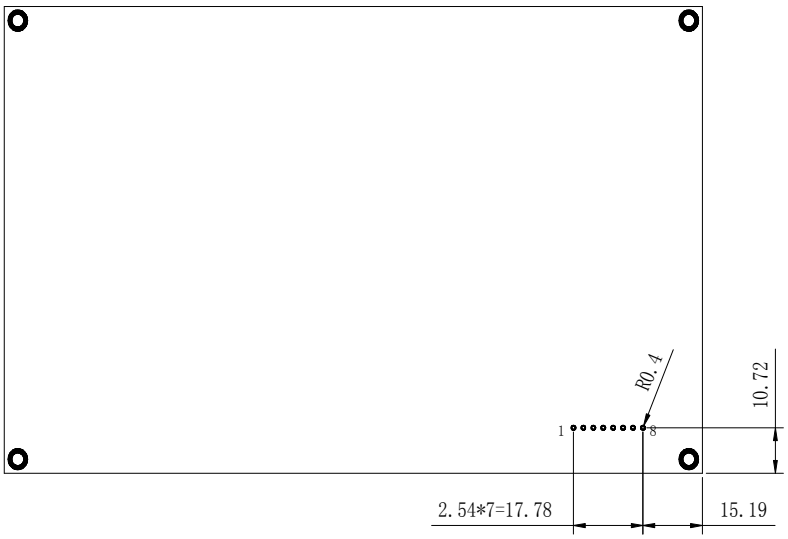


背面

OCM800480T7001C

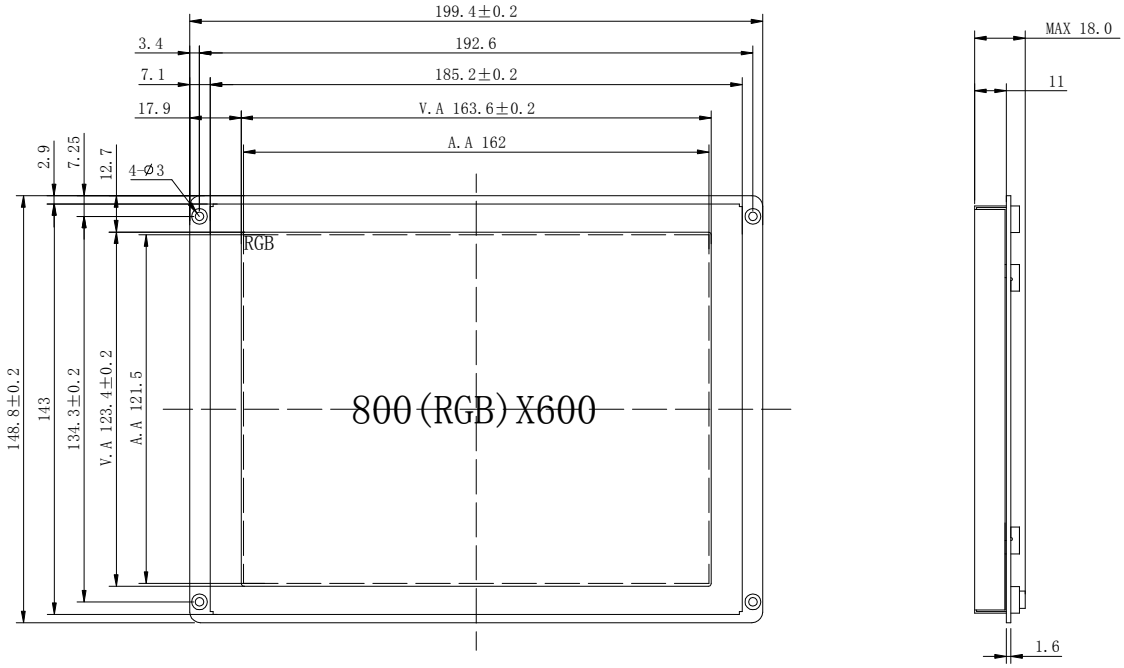


正面

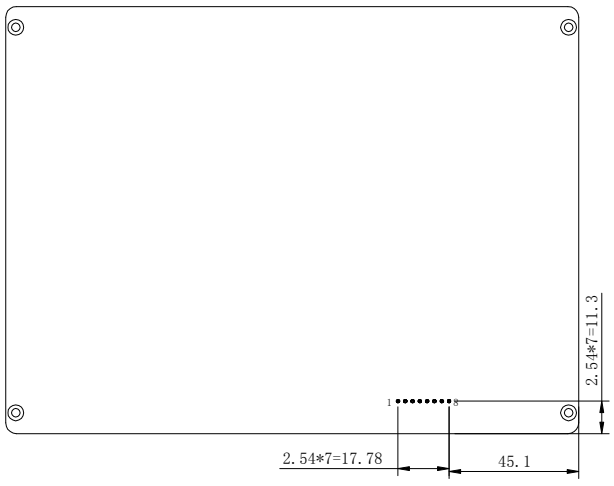


背面

OCM800600T800-1C

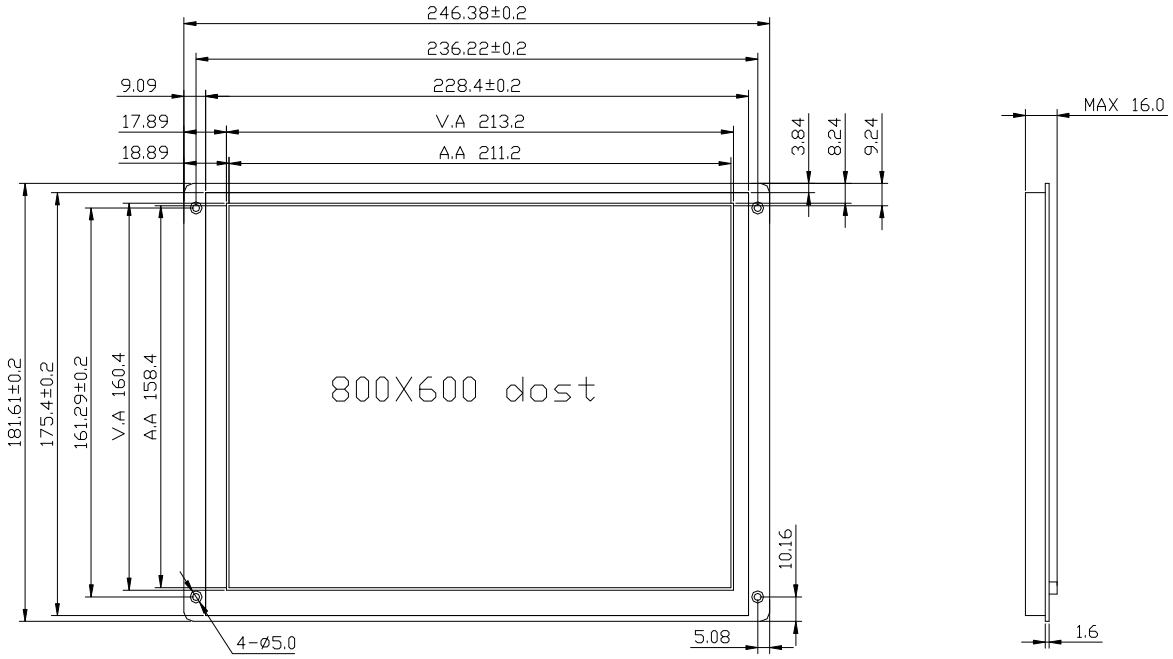


正面

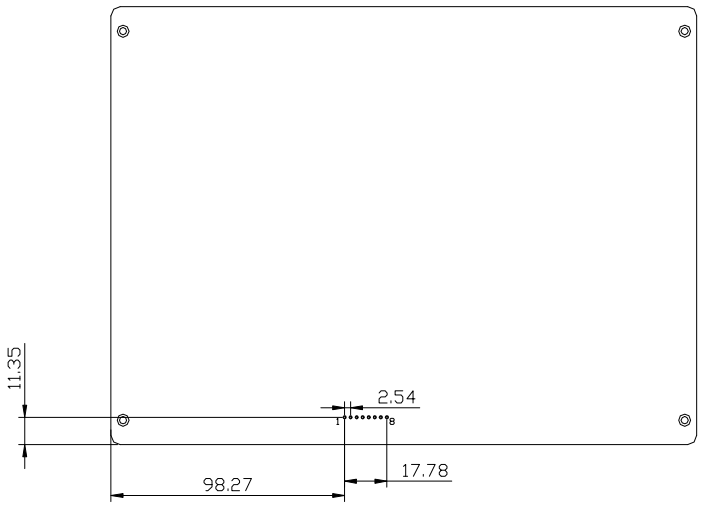


背面

OCM800600T104-1C



正面



背面