

1. 简介

HD35-RS 是 RS485 接口 3.5 寸 TFT 彩色显示模块，自带 ASCII 字库和中文字库。模块带 64M bit 的内存，可存储 BMP 图片文件，通过 RS485 接口与 MCU 连接，实现彩色图形信息显示。

(1) 集成 4 个 ASCII 字库和 4 个汉字字库，字库类型：

ASCII 字库：8x16，12x24，16x32,32x64

GB2312 中文字库：16x16，24x24，32x32,64x64

(2) 内置内存容量：64Mbit

字库占用 16Mbit，剩余空间 48Mbit 用于存储图片数据。

(3) 集成绘图功能，向 LCDs 简单地发送命令和坐标参数，可实现点，直线，方形，圆形，按键，窗体，编辑框显示。

2. 规格说明

序号	项目	规格标准值	单位	备注
1	显示点阵数	480*RGB*320	Dots	
2	LCD 尺寸	3.5	inch	
3	LCM 外型尺寸		mm	
4	动态显示区	46(L)*72(W)	mm	
5	像素尺寸	0.15(L)*0.15(W)	mm	
6	像素成份	a-Si TFT		
7	LCD 模式	65 k TFT		16 位色彩
8	视角	12 o'clock		
9	背光	白色 led		
10	模块供电	6~15	V	

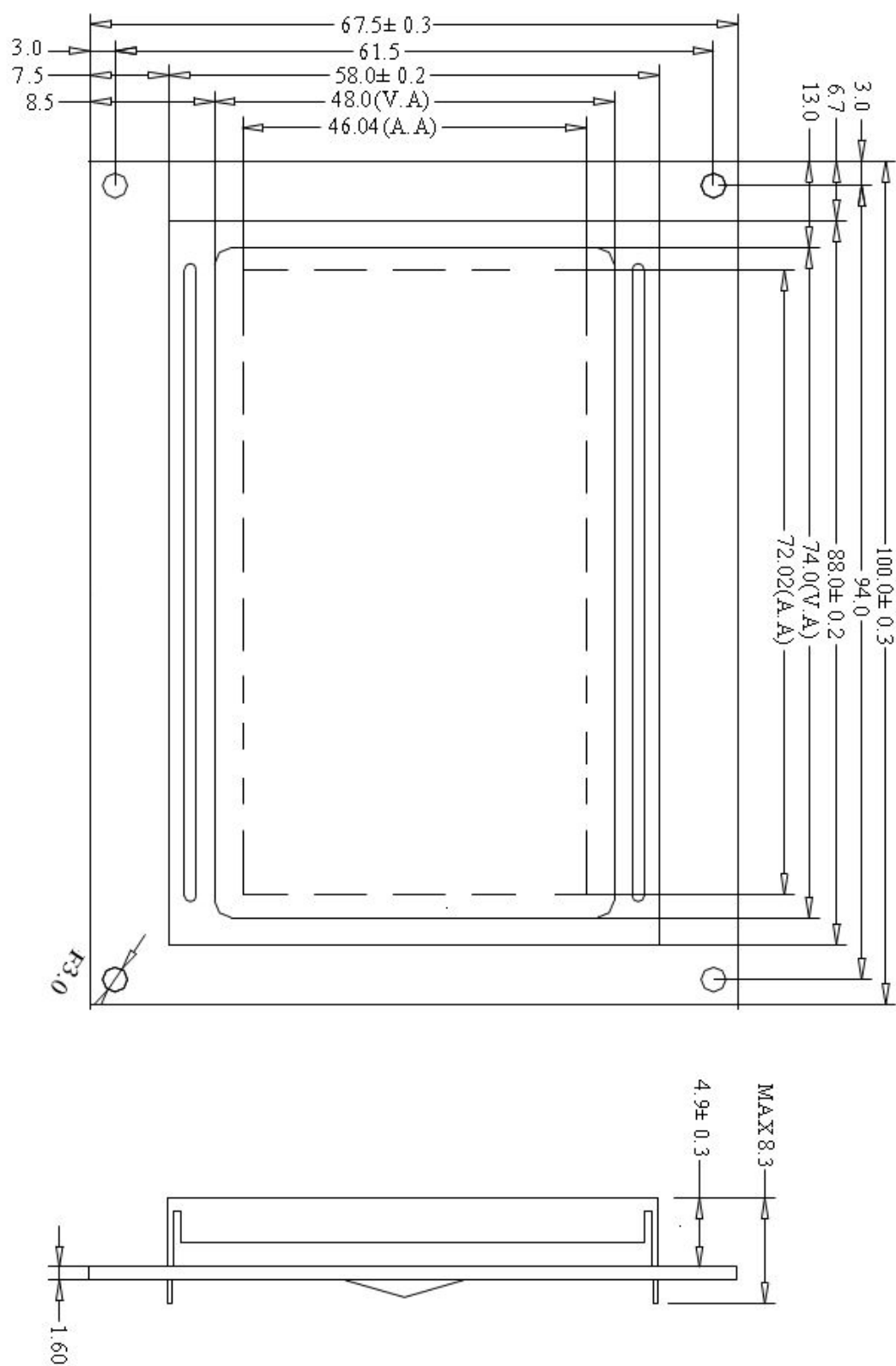
产品最大额定值：

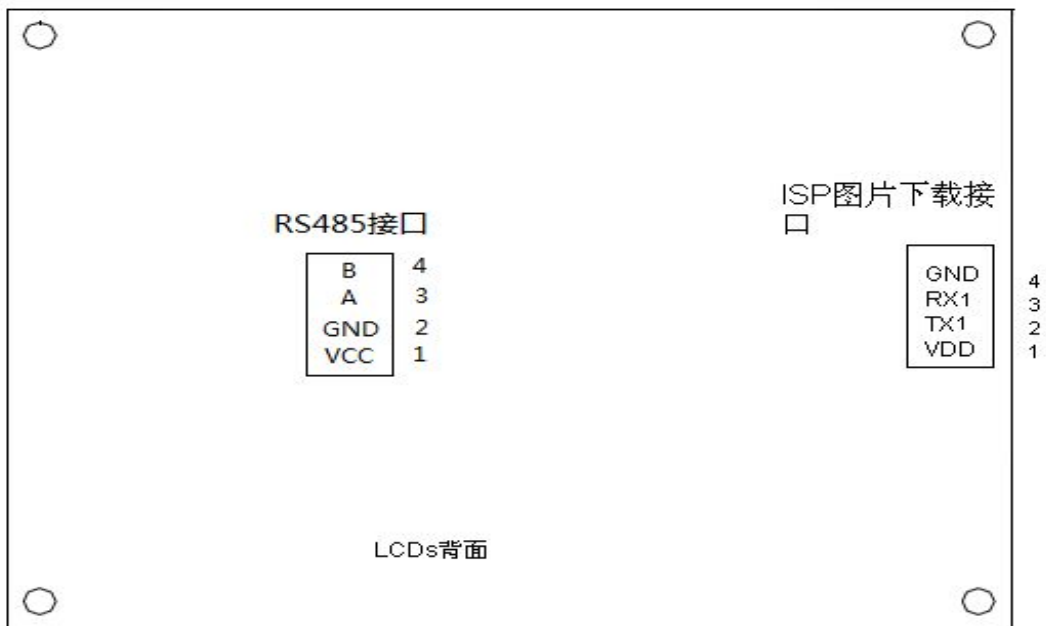
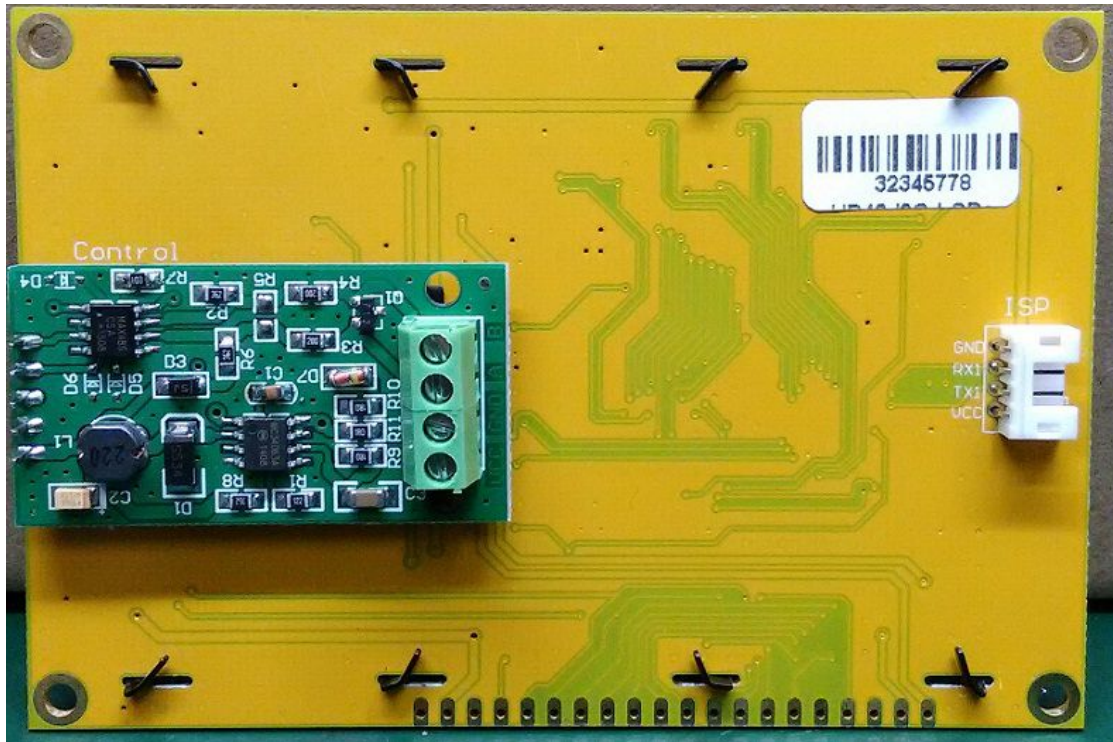
项目	符号	最小值	典型值	最大值	单位	备注
操作温度	Top	-20	-	+70	℃	
存储温度	Tst	-30	-	+80	℃	
供电电压	VDD	-0.3	-	15	V	
输入电压	Vin	-0.3	-	VDD	V	根据输入电压确定

产品电气特性：

项目	符号	最小值	典型值	最大值	单位	备注
工作电压	VDD	6	12	15	V	6~15V 供电
工作电流	I	50	70	80	mA	背光开
	I	40	50	70		背光关

3. 结构图与引脚





RS485 控制接口说明:

接口序号	接口名称
1	VCC 电源
2	GND 地
3	A RS485 A 接口
4	B RS485 B 接口

注: RS485 控制口默认的通讯速率为: 9600 KBPS

ISP 图片下载接口说明：

接口序号	接口名称
1	VDD 电源
2	TX1 串行发送接口
3	RX1 串行接收接口
4	GND 地

注：ISP 图片下载串口的默认通讯波特率为：256000 KBPS

4. 控制命令格式和指令集

通过 RS485 接口向 LCDs 模块发送命令和数据是通过数据包的格式发送，数据包格式由命令头（0xAA），设备地址（Address）数据长度（Lengh），命令（Command），数据（Data），校验和（Checksum）组成。

4.1. 控制指令格式

0xAA	Address	Lengh	Command	Data	Checksum
------	---------	-------	---------	------	----------

Lengh 的数值为整个数据包长度减 1，Checksum 为整个数据包的数值相加和。

Address 设备地址（范围 0~255）

Lengh = 整个数据包长度-1

Data 发送给模块的数据，长度最长为 24byte

Checksum = 0xAA+Address+Lengh+Command+Data

4.2. 返回数据指令格式

0xAA	Address	Lengh	Status	Command	Checksum
------	---------	-------	--------	---------	----------

Lengh 的数值为整个数据包长度减 1，Checksum 为整个数据包的数值相加和。

Lengh = 整个数据包长度-1

Status =0 执行命令有误，Status =1 成功执行命令

Command 返回状态的命令

Checksum = 0xAA+Address+Lengh+ Status+ Command

4.3. 控制命令集

功 能 function	数据长度 Lengh	命令 Command	数据 Data	返回数据
清 整 个 显示屏	5	0x80	Data1 = 0x00	Lengh=5 成功： Status =1 出错： Status =0
设 置 背 景色	6	0x81	Data1:背景色高八位。 Data2:背景色低八位。	Lengh=5 成功： Status =1 出错： Status =0
设 置 前 景色	6	0x82	Data1: 字符颜色高八位。 Data2: 字符颜色低八位。	Lengh=5 成功： Status =1 出错： Status =0
设 置 背 光亮度	5	0x83	Data1: 背光亮度，范围 0x00 关背光 0x01~0x09 九级可调背 光	Lengh=5 成功： Status =1 出错： Status =0

ASCII 字符显示	10	0x85	Data1: 字符左上角 X 轴坐标高八位 Data2: 字符左上角 X 轴坐标低八位 Data3: 字符左上角 Y 轴坐标高八位 Data4: 字符左上角 Y 轴坐标低八位 Data5: 高 4 位字符显示方式, 低 4 位字符类型 Data6: ASCII 码	Lenth=5 成功: Status =1 出错: Status =0
ASCII 字符串显示	>=10 <=28	0x86	Data1: 左上角 X 轴坐标高八位 Data2: 左上角 X 轴坐标低八位 Data3: 左上角 Y 轴坐标高八位 Data4: 左上角 Y 轴坐标低八位 Data5: 高 4 位字符显示方式, 低 4 位字符类型 Data6~DataN: 要显示的 ASCII 数据	Lenth=5 成功: Status =1 出错: Status =0
汉字显示	11	0x87	Data1: 字符左上角 X 轴坐标高八位 Data2: 字符左上角 X 轴坐标低八位 Data3: 字符左上角 Y 轴坐标高八位 Data4: 字符左上角 Y 轴坐标低八位 Data5: 高 4 位字符显示方式, 低 4 位字符类型 Data6: 二级汉字库 GB 码高八位 Data7: 二级汉字库 GB 码低八位	Lenth=5 成功: Status =1 出错: Status =0
汉字字符串显示	>=10 <=28	0x88	Data1: 左上角 X 轴坐标高八位 Data2: 左上角 X 轴坐标低八位 Data3: 左上角 Y 轴坐标高八位 Data4: 左上角 Y 轴坐标	Lenth=5 成功: Status =1 出错: Status =0

			低八位 Data5: 高 4 位字符显示方式, 低 4 位字符类型 Data6~DataN: 要显示的中文 GB 码数据 中文 GB 码的顺序按照先高字节后低字节	
图片显示	12	0x89	Data1: 图标左上角 X 轴坐标高八位 Data2: 图标左上角 X 轴坐标低八位 Data3: 图标左上角 Y 轴坐标高八位 Data4: 图标左上角 Y 轴坐标低八位 Data5~ Data8: 图标地址共 4 个字节, 顺序按照先高字节后低字节	Lenth=5 成功: Status =1 出错: Status =0
画点	8	0x8A	Data1: 点 X 轴坐标高八位 Data2: 点 X 轴坐标低八位 Data3: 点 Y 轴坐标高八位 Data4: 点 Y 轴坐标低八位	Lenth=5 成功: Status =1 出错: Status =0
画任意直线	12	0x8B	Data1: 直线起点 X 轴坐标高八位 Data2: 直线起点 X 轴坐标低八位 Data3: 直线起点 Y 轴坐标高八位 Data4: 直线起点 Y 轴坐标低八位 Data5: 直线终点 X 轴坐标高八位 Data6: 直线终点 X 轴坐标低八位 Data7: 直线终点 Y 轴坐标高八位 Data8: 直线终点 Y 轴坐标低八位	Lenth=5 成功: Status =1 出错: Status =0
画矩形	12	0x8C	Data1: 左上角 X 轴坐标	Lenth=5

框			高八位 Data2: 左上角 X 轴坐标 低八位 Data3: 左上角 Y 轴坐标 高八位 Data4: 左上角 Y 轴坐标 低八位 Data5: 矩形长度高八位 Data6: 矩形长度低八位 Data7: 矩形高度高八位 Data8: 矩形高度低八位	成功: Status =0x59 出错: Status =0x4E
画实心 矩形	12	0x8D	Data1: 左上角 X 轴坐标 高八位 Data2: 左上角 X 轴坐标 低八位 Data3: 左上角 Y 轴坐标 高八位 Data4: 左上角 Y 轴坐标 低八位 Data5: 矩形长度高八位 Data6: 矩形长度低八位 Data7: 矩形高度高八位 Data8: 矩形高度低八位	Lenth=5 成功: Status =0x59 出错: Status =0x4E
画圆	10	0x8E	Data1: 圆心 X 轴坐标高 八位 Data2: 圆心 X 轴坐标低 八位 Data3: 圆心 Y 轴坐标高 八位 Data4: 圆心 Y 轴坐标低 八位 Data5: 圆半径高八位 Data6: 圆半径低八位	Lenth=5 成功: Status =0x59 出错: Status =0x4E
画实心 圆	10	0x8F	Data1: 圆心 X 轴坐标高 八位 Data2: 圆心 X 轴坐标低 八位 Data3: 圆心 Y 轴坐标高 八位 Data4: 圆心 Y 轴坐标低 八位 Data5: 圆半径高八位 Data6: 圆半径低八位	Lenth=5 成功: Status =1 出错: Status =0

画窗体 box	13	0x9a	Data1: 窗体起点 X 轴坐标高八位 Data2: 窗体起点 X 轴坐标低八位 Data3: 窗体起点 Y 轴坐标高八位 Data4: 窗体起点 Y 轴坐标低八位 Data5: 窗体终点 X 轴坐标高八位 Data6: 窗体终点 X 轴坐标低八位 Data7: 窗体终点 Y 轴坐标高八位 Data8: 窗体终点 Y 轴坐标低八位 Data9: 显示模式	Lengh=5 成功: Status =1 出错: Status =0
画按钮 button	13	0x9b	Data1: 按钮起点 X 轴坐标高八位 Data2: 按钮起点 X 轴坐标低八位 Data3: 按钮起点 Y 轴坐标高八位 Data4: 按钮起点 Y 轴坐标低八位 Data5: 按钮终点 X 轴坐标高八位 Data6: 按钮终点 X 轴坐标低八位 Data7: 按钮终点 Y 轴坐标高八位 Data8: 按钮终点 Y 轴坐标低八位 Data9: 显示模式	Lengh=5 成功: Status =1 出错: Status =0
画 ASCII 字符串 编辑栏 edit	>=13 <=28	0x9c	Data1: 编辑栏起点 X 轴坐标高八位 Data2: 编辑栏起点 X 轴坐标低八位 Data3: 编辑栏起点 Y 轴坐标高八位 Data4: 编辑栏起点 Y 轴坐标低八位 Data5: 编辑栏终点 X 轴坐标高八位	Lengh=5 成功: Status =1 出错: Status =0

			Data6: 编辑栏终点 X 轴坐标低八位 Data7: 编辑栏终点 Y 轴坐标高八位 Data8: 编辑栏终点 Y 轴坐标低八位 Data9: bit7~bit4 字符串长度 bit2~bit0 字符类型 Data6~DataN: 要显示的 ASCII 数据	
画汉字串编辑栏 edit	>=13 <=28	0x9d	Data1: 编辑栏起点 X 轴坐标高八位 Data2: 编辑栏起点 X 轴坐标低八位 Data3: 编辑栏起点 Y 轴坐标高八位 Data4: 编辑栏起点 Y 轴坐标低八位 Data5: 编辑栏终点 X 轴坐标高八位 Data6: 编辑栏终点 X 轴坐标低八位 Data7: 编辑栏终点 Y 轴坐标高八位 Data8: 编辑栏终点 Y 轴坐标低八位 Data5: bit7~bit4 字符串长度 bit2~bit0 字符类型 Data6~DataN: 要显示的汉字数据	Lengh=5 成功: Status =1 出错: Status =0
十进制数字显示	10	0x9e	Data1: 数字左上角 X 轴坐标高八位 Data2: 数字左上角 X 轴坐标低八位 Data3: 数字左上角 Y 轴坐标高八位 Data4: 数字左上角 Y 轴坐标低八位	Lengh=5 成功: Status =1 出错: Status =0

			Data5: 高 4 位数字显示方式，低 4 位数字显示大小 Data6: 要显示十进制数字（范围 0~255）	
--	--	--	--	--

4.4 控制命令详述

4.4.1 清整个显示屏（0x80）

把整个显示屏设为背景色，背景色默认为蓝色。

发送数据：

0xAA address 0x05 0x80 0x00 CheckSum

注：

address 设备地址（范围 0~255）

操作成功返回：

0xAA address 0x05 0x01 0x80 CheckSum

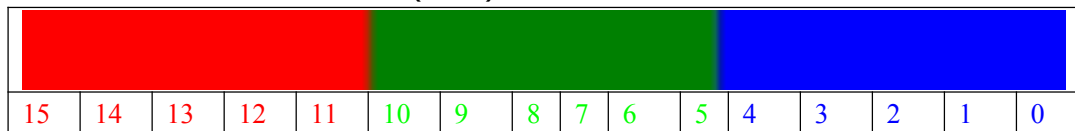
操作出错返回：

0xAA address 0x05 0x00 0x80 CheckSum

4.4.2 设置背景色（0x81）

设置在 LCD 上显示的背景色，主要与清整个显示屏命令配合使用，默认颜色为蓝色。

本 LCD 模块采用 RGB565 彩色模式，一个像素占两个字节，其中：第一个字节的前 5 位用来表示 R(Red)，第一个字节的后三位+第二个字节的前三位用来表示 G(Green)，第二个字节的后 5 位用来表示 B(Blue)。如：



常用颜色的 RGB565 数值：

颜色	RGB565 值
红	0xF800
蓝	0x001F
绿	0x07E0
白	0xFFFF
黑	0x0000

发送数据：

0xAA address 0x06 0x81 ColorH ColorL CheckSum

注：

address 设备地址（范围 0~255）

ColorH 背景颜色高 8 位

ColorL 背景颜色低 8 位

CheckSum: 校验和

操作成功返回：

0xAA address 0x05 0x01 0x81 CheckSum

操作出错返回：

0xAA address 0x05 0x00 0x81 CheckSum

4.4.3 设置前景颜色（0x82）

设置在 LCD 上显示的字符颜色

发送数据：

0xAA address 0x06 0x82 ColorH ColorL CheckSum

注：

address 设备地址（范围 0~255）

ColorH 前景颜色高 8 位

ColorL 前景颜色低 8 位

CheckSum: 校验和

操作成功返回：

0xAA address 0x05 0x01 0x82 CheckSum

操作出错返回：

0xAA address 0x05 0x00 0x82 CheckSum

4.4.4 设置背光亮度（0x83）

设置 LCD 模块背光亮度。

发送数据：

0xAA address 0x05 0x83 BLValue CheckSum

注：

address 设备地址（范围 0~255）

BLValue 背光亮度设置值，范围从 1~9。

亮度	数值
最暗	1
可调范围	1~9
最高	9
关闭背光	0x00

CheckSum: 校验和

操作成功返回：

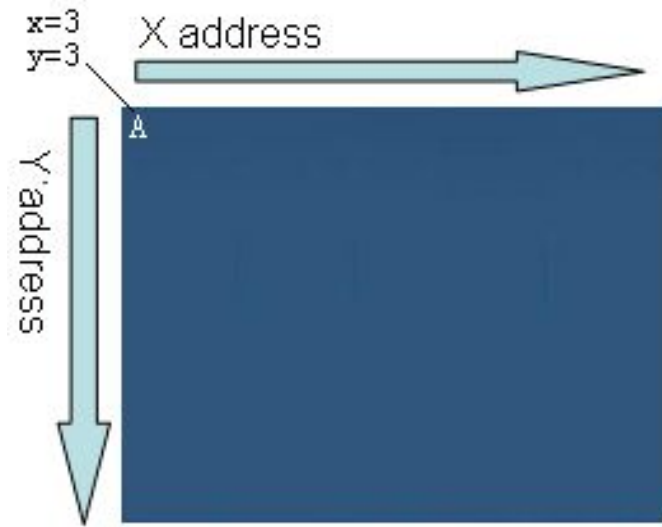
0xAA address 0x05 0x01 0x83 CheckSum

操作出错返回：

0xAA address 0x05 0x00 0x83 CheckSum

4.4.5 ASCII 字符显示（0x85）

在屏幕上显示 ASCII 字符，先是指定要显示的 X 轴坐标和 Y 轴坐标，然后发送要显示字符的 ASCII 码。如下图示，要显示的字符坐标是，X=3，Y=3。要显示字符是 ‘A’。



发送数据：

0xAA address 0x0A 0x85 XH XL YH YL FT ASCII CheckSum

注：

address 设备地址（范围 0~255）

FT：

高 4 位：0xf 透明写字符 0x0 带背景写字符

低 4 位：ASCII 显示字库类型

字库尺寸	数值
8x16	1
12x24	2
16x32	3
32x64	0

XH: X 轴地址高 8 位

XL: X 轴地址低 8 位

YH: Y 轴地址高 8 位

YL: Y 轴地址低 8 位

ASCII: ASCII 码

CheckSum: 校验和

操作成功返回：

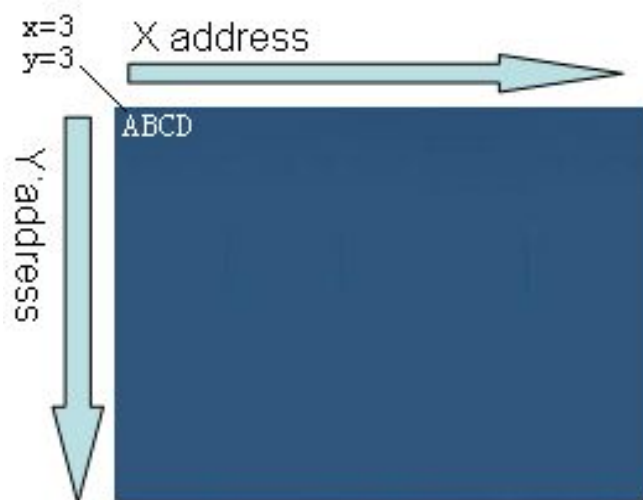
0xAA address 0x05 0x01 0x85 CheckSum

操作出错返回：

0xAA address 0x05 0x00 0x85 CheckSum

4.4.7 ASCII 字符串显示 (0x86)

在屏幕上显示 ASCII 字符串，先是指定要显示的 X 轴坐标和 Y 轴坐标，然后发送要显示字符串的 ASCII 码。如下图示，要显示的字符坐标是，X=3，Y=3。要显示字符串是‘ABCD’。



发送数据：

0xAA address StringLen 0x86 XH XL YH YL FT ASCIIString CheckSum

注：

address 设备地址（范围 0~255）

StringLen: 数据长度（范围 10~28）

FT:

高 4 位：0xf 透明写字符 0x0 带背景写字符

低 4 位：ASCII 显示字库类型

字库尺寸	数值
8x16	1
12x24	2
16x32	3
32x64	0

XH: X 轴地址高 8 位

XL: X 轴地址低 8 位

YH: Y 轴地址高 8 位

YL: Y 轴地址低 8 位

ASCIIString: ASCII 码字符串

CheckSum: 校验和

操作成功返回：

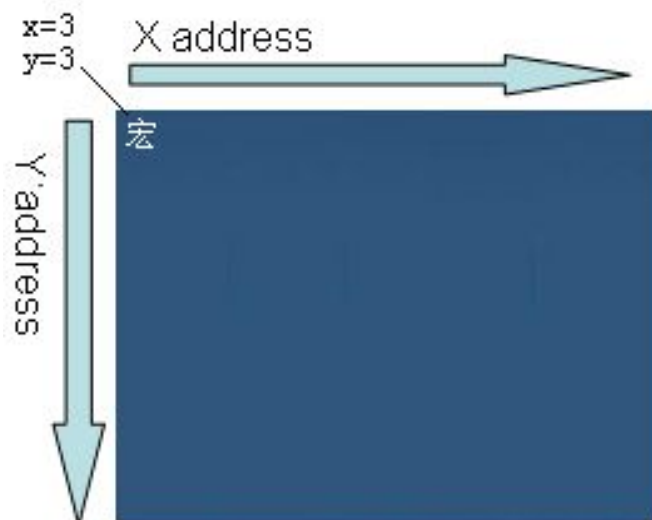
0xAA address 0x05 0x01 0x86 CheckSum

操作出错返回：

0xAA address 0x05 0x00 0x86 CheckSum

4.4.8 汉字显示 (0x87)

在屏幕上显示汉字字符，先是指定要显示的 X 轴坐标和 Y 轴坐标，然后发送要显示汉字的区位码。如下图示，要显示的字符坐标是，X=3，Y=3。要显示字符是‘宏’。



发送数据：

0xAA address 0x0B 0x87 XH XL YH YL FT HZH HZL CheckSum

注：

address 设备地址（范围 0~255）

FT：

高 4 位：0xf 透明写字符 0x0 带背景写字符

低 4 位：汉字显示字库类型

字库尺寸	数值
16x16	4
24x24	5
32x32	6
64x64	7

XH: X 轴地址高 8 位

XL: X 轴地址低 8 位

YH: Y 轴地址高 8 位

YL: Y 轴地址低 8 位

HZH: 汉字区位码高 8 位

HZL: 汉字区位码低 8 位

CheckSum: 校验和

操作成功返回：

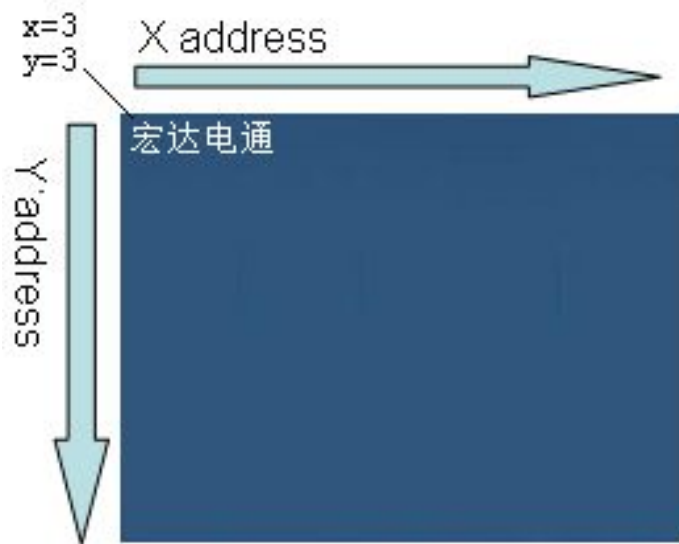
0xAA address 0x05 0x01 0x87 CheckSum

操作出错返回：

0xAA address 0x05 0x00 0x87 CheckSum

4.4.9 汉字字符串显示 (0x88)

在屏幕上显示汉字字符串，先是指定要显示的 X 轴坐标和 Y 轴坐标，然后发送要显示汉字串的区位码。如下图所示，要显示的字符坐标是，X=3，Y=3。要显示字符是‘宏达电通’。



发送数据：

0xAA address StringLen 0x88 XH XL YH YL FT HZString CheckSum

注：

address 设备地址（范围 0~255）

StringLen: 数据长度（范围 11~28）

FT:

高 4 位: 0xf 透明写字符 0x0 带背景写字符

低 4 位: 汉字显示字库类型

字库尺寸	数值
16x16	4
24x24	5
32x32	6
64x64	7

XH: X 轴地址高 8 位

XL: X 轴地址低 8 位

YH: Y 轴地址高 8 位

YL: Y 轴地址低 8 位

HZString: 汉字区位码字符串

CheckSum: 校验和

操作成功返回：

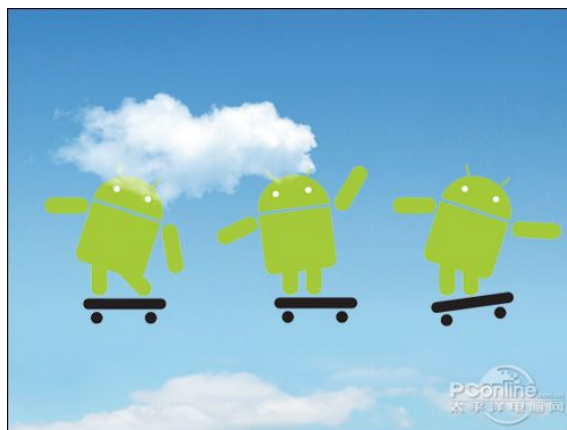
0xAA address 0x05 0x01 0x88 CheckSum

操作出错返回：

0xAA address 0x05 0x00 0x88 CheckSum

4.4.10 图片显示 (0x89)

在屏幕上显示图标，先是指定要显示的 X 轴坐标和 Y 轴坐标，然后发送要显示图标在 flash 的存储地址。如下图示，要显示的字符坐标是，X=0，Y=0。



发送数据：

0xAA address 0x0C 0x89 XH XL YH YL Icon3 Icon2 Icon1 Icon0 CheckSum

注：

address 设备地址（范围 0~255）

XH: X 轴地址高 8 位

XL: X 轴地址低 8 位

YH: Y 轴地址高 8 位

YL: Y 轴地址低 8 位

Icon3: icon 地址 24~31 位

Icon2: icon 地址 16~23 位

Icon1: icon 地址 8~15 位

Icon0: icon 地址 0~7 位

CheckSum: 校验和

操作成功返回：

0xAA address 0x05 0x01 0x89 CheckSum

操作出错返回：

0xAA address 0x05 0x00 0x89 CheckSum

4.4.11 画点 (0x8A)

在 LCD 任意区域根据前景色画一点

发送数据：

0xAA address 0x08 0x8A XH XL YH YL CheckSum

注：

address 设备地址（范围 0~255）

XH: X 轴地址高 8 位

XL: X 轴地址低 8 位

YH: Y 轴地址高 8 位

YL: Y 轴地址低 8 位

Checksum: 校验和

操作成功返回:

0xAA address 0x05 0x01 0x8A CheckSum

操作出错返回:

0xAA address 0x05 0x00 0x8A CheckSum

4.4.12 画任意线 (0x8B)

在 LCD 任意区域画线

发送数据:

0xAA address 0x0C 0x8B XSH XSL YSH YSL XEH XEL YEH YEL CheckSum

注:

address 设备地址 (范围 0~255)

XSH: 直线 X 轴起始地址高 8 位

XSL: 直线 X 轴起始地址低 8 位

YSH: 直线 Y 轴起始地址高 8 位

YSL: 直线 Y 轴起始地址低 8 位

XEH: 直线 X 轴结束地址高 8 位

XEL: 直线 X 轴结束地址低 8 位

YEH: 直线 Y 轴结束地址高 8 位

YEL: 直线 Y 轴结束地址低 8 位

Checksum: 校验和

操作成功返回:

0xAA address 0x05 0x01 0x8B CheckSum

操作出错返回:

0xAA address 0x05 0x00 0x8B CheckSum

4.4.13 画矩形框 (0x8C)

在 LCD 任意区域画矩形框

发送数据:

0xAA address 0x0C 0x8C XH XL YH YL WH WL HH HL CheckSum

注:

address 设备地址 (范围 0~255)

XH: X 轴地址高 8 位

XL: X 轴地址低 8 位

YH: Y 轴地址高 8 位

YL: Y 轴地址低 8 位

WH: 矩形框宽度高 8 位

WL: 矩形框宽度低 8 位

HH: 矩形框高度高 8 位

HL: 矩形框高度低 8 位

Checksum: 校验和

操作成功返回:

0xAA address 0x05 0x01 0x8C CheckSum

操作出错返回：

0xAA address 0x05 0x00 0x8C CheckSum

4.4.14 画实心矩形（0x8D）

在 LCD 任意区域画矩形框

发送数据：

0xAA address 0x0C 0x8D XH XL YH YL WH WL HH HL CheckSum

注：

address 设备地址（范围 0~255）

XH: X 轴地址高 8 位

XL: X 轴地址低 8 位

YH: Y 轴地址高 8 位

YL: Y 轴地址低 8 位

WH: 矩形宽度高 8 位

WL: 矩形宽度低 8 位

HH: 矩形高度高 8 位

HL: 矩形高度低 8 位

CheckSum: 校验和

操作成功返回：

0xAA address 0x05 0x01 0x8D CheckSum

操作出错返回：

0xAA address 0x05 0x00 0x8D CheckSum

4.4.15 画圆形（0x8E）

在 LCD 任意区域画圆形框

发送数据：

0xAA address 0x0A 0x8E XH XL YH YL RH RL CheckSum

注：

address 设备地址（范围 0~255）

XH: 圆心 X 轴地址高 8 位

XL: 圆心 X 轴地址低 8 位

YH: 圆心 Y 轴地址高 8 位

YL: 圆心 Y 轴地址低 8 位

RH: 圆形半径高 8 位

RL: 圆形半径低 8 位

CheckSum: 校验和

操作成功返回：

0xAA address 0x05 0x01 0x8E CheckSum

操作出错返回：

0xAA address 0x05 0x00 0x8E CheckSum

4.4.16 画实心圆形 (0x8F)

在 LCD 任意区域画实心圆形

发送数据:

0xAA address 0x0A 0x8F XH XL YH YL RH RL CheckSum

注:

address 设备地址 (范围 0~255)

XH: 圆心 X 轴地址高 8 位

XL: 圆心 X 轴地址低 8 位

YH: 圆心 Y 轴地址高 8 位

YL: 圆心 Y 轴地址低 8 位

RH: 实心圆形半径高 8 位

RL: 实心圆形半径低 8 位

CheckSum: 校验和

操作成功返回:

0xAA address 0x05 0x01 0x8F CheckSum

操作出错返回:

0xAA address 0x05 0x00 0x8F CheckSum

4.4.17 画窗体 box (0x9A)

在 LCD 任意区域画窗体

发送数据:

0xAA address 0x0D 0x9A XSH XSL YSH YSL XEH XEL YEH YEL MODE CheckSum

注:

address 设备地址 (范围 0~255)

XSH: 窗体 X 轴起始地址高 8 位

XSL: 窗体 X 轴起始地址低 8 位

YSH: 窗体 Y 轴起始地址高 8 位

YSL: 窗体 Y 轴起始地址低 8 位

XEH: 窗体 X 轴结束地址高 8 位

XEL: 窗体 X 轴结束地址低 8 位

YEH: 窗体 Y 轴结束地址高 8 位

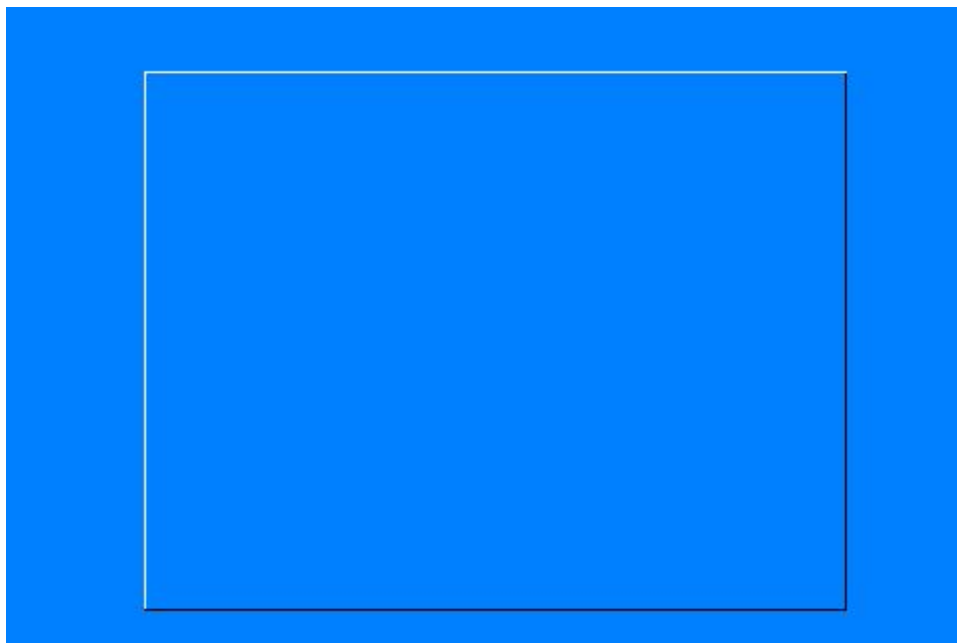
YEL: 窗体 Y 轴结束地址低 8 位

MODE:

0x00 窗体上凸

0xff 窗体下凹

CheckSum: 校验和



操作成功返回：

0xAA address 0x05 0x01 0x9A CheckSum

操作出错返回：

0xAA address 0x05 0x00 0x9A CheckSum

4.4.18 画按键 button (0x9B)

在 LCD 任意区域画按键

发送数据：

0xAA address 0x0D 0x9B XSH XSL YSH YSL XEH XEL YEH YEL MODE CheckSum

注：

address 设备地址（范围 0~255）

XSH：按键 X 轴起始地址高 8 位

XSL：按键 X 轴起始地址低 8 位

YSH：按键 Y 轴起始地址高 8 位

YSL：按键 Y 轴起始地址低 8 位

XEH：按键 X 轴结束地址高 8 位

XEL：按键 X 轴结束地址低 8 位

YEH：按键 Y 轴结束地址高 8 位

YEL：按键 Y 轴结束地址低 8 位

MODE:

0x00 按键上凸

0xff 按键下凹

CheckSum: 校验和



操作成功返回：
0xAA address 0x05 0x01 0x9B CheckSum
操作出错返回：
0xAA address 0x05 0x00 0x9B CheckSum

4.4.19 画 ASCII 字符显示编辑栏 edit (0x9C)

在 LCD 任意区域画编辑栏
发送数据：

0xAA address StringLen 0x9C XSH XSL YSH YSL XEH XEL YEH YEL FT ASCIIString
Checksum

注：
address 设备地址（范围 0~255）
StringLen: 数据长度（范围 14~28）
XSH: 编辑栏 X 轴起始地址高 8 位
XSL: 编辑栏 X 轴起始地址低 8 位
YSH: 编辑栏 Y 轴起始地址高 8 位
YSL: 编辑栏 Y 轴起始地址低 8 位
XEH: 编辑栏 X 轴结束地址高 8 位
XEL: 编辑栏 X 轴结束地址低 8 位
YEH: 编辑栏 Y 轴结束地址高 8 位
YEL: 编辑栏键 Y 轴结束地址低 8 位
FT:

bit7~bit4 字符串长度
bit2~bit0 字库尺寸类型

字库尺寸	数值
------	----

8x16	1
12x24	2
16x32	3
32x64	0

ASCIIString:

ASCII 码字符串



操作成功返回:

0xAA address 0x05 0x01 0x9C CheckSum

操作出错返回:

0xAA address 0x05 0x00 0x9C CheckSum

4.4.20 画汉字字符显示编辑栏 edit (0x9D)

在 LCD 任意区域画编辑栏

发送数据:

0xAA address StringLen 0x9D XSH XSL YSH YSL XEH XEL YEH YEL FT HZString
Checksum

注:

address 设备地址 (范围 0~255)

StringLen: 数据长度 (范围 15~28)

XSH: 编辑栏 X 轴起始地址高 8 位

XSL: 编辑栏 X 轴起始地址低 8 位

YSH: 编辑栏 Y 轴起始地址高 8 位
 YSL: 编辑栏 Y 轴起始地址低 8 位
 XEH: 编辑栏 X 轴结束地址高 8 位
 XEL: 编辑栏 X 轴结束地址低 8 位
 YEH: 编辑栏 Y 轴结束地址高 8 位
 YEL: 编辑栏键 Y 轴结束地址低 8 位
 FT:

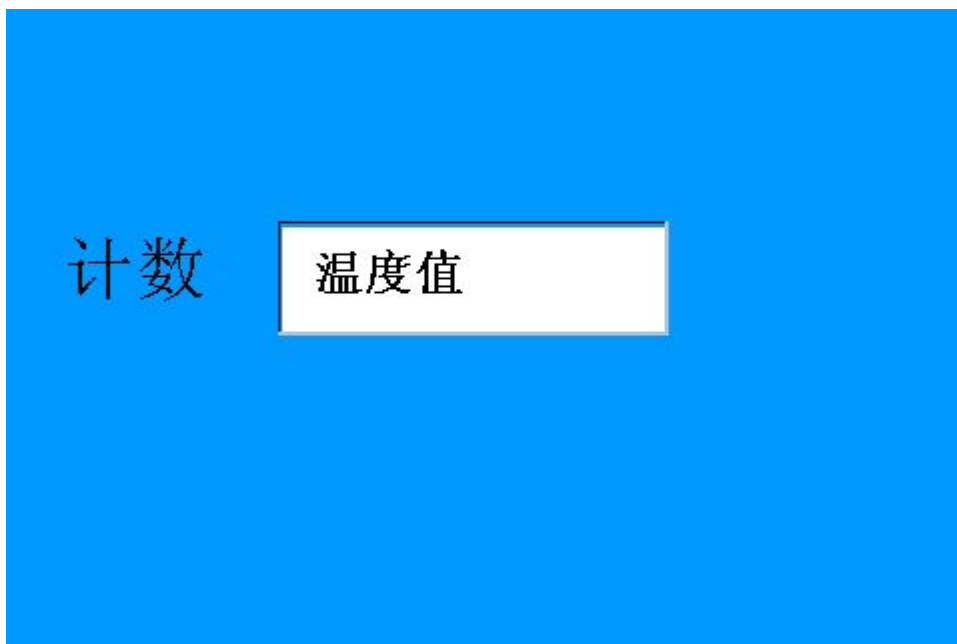
bit7~bit4 字符串长度

bit2~bit0 字库尺寸类型

字库尺寸	数值
16x16	4
24x24	5
32x32	6
64x64	7

HZString:

汉字字符串



操作成功返回:

0xAA address 0x05 0x01 0x9D CheckSum

操作出错返回:

0xAA address 0x05 0x00 0x9D CheckSum

4.4.21 十进制数字显示 (0x9E)

在屏幕上指定 X 轴坐标和 Y 轴坐标显示十进制数字

发送数据:

0xAA address 0x0A 0x9E XH XL YH YL FT Number CheckSum

注：

address 设备地址（范围 0~255）

FT：

高 4 位：0xf 透明写字符 0x0 带背景写字符

低 4 位：数字显示字库类型

字库尺寸	数值
8x16	1
12x24	2
16x32	3
32x64	0

XH：X 轴地址高 8 位

XL：X 轴地址低 8 位

YH：Y 轴地址高 8 位

YL：Y 轴地址低 8 位

Number ：十进制数字

Checksum： 校验和

操作成功返回：

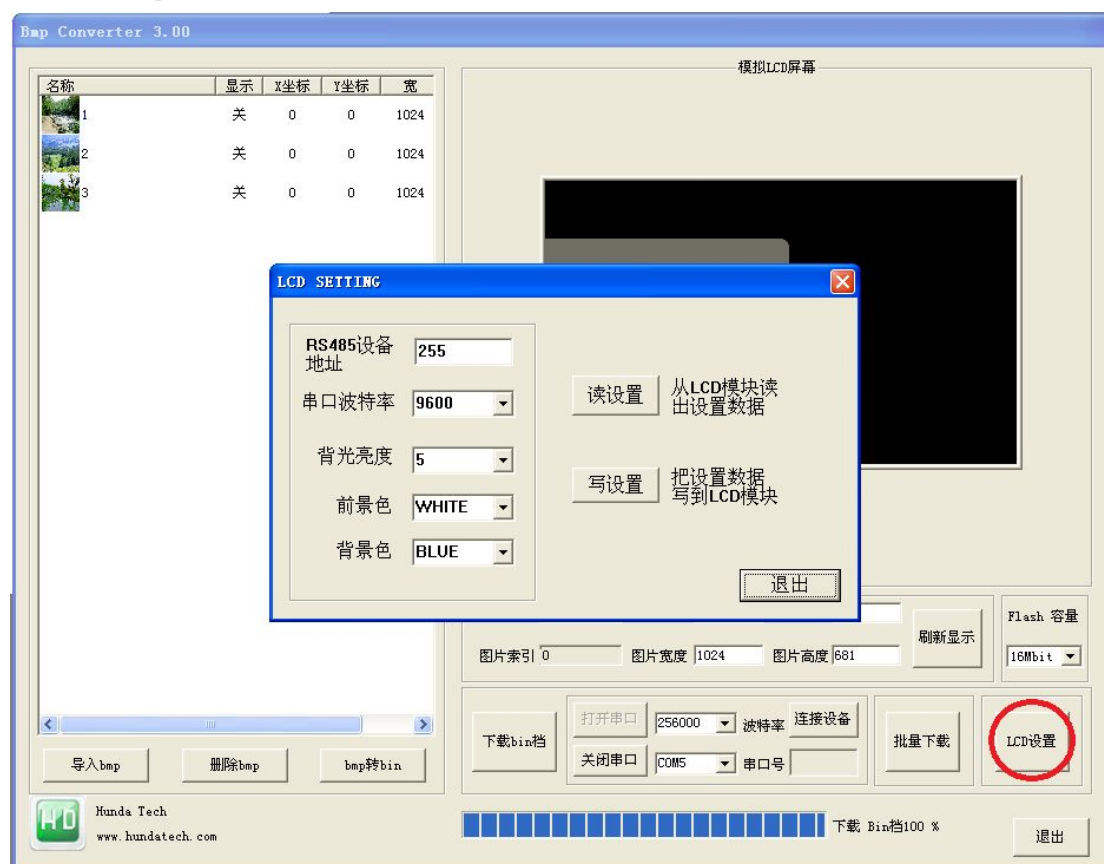
0xAA address 0x05 0x01 0x9E CheckSum

操作出错返回：

0xAA address 0x05 0x00 0x9E CheckSum

5.修改液晶显示模块默认设置

5.1 使用 BmpConverter 通过串口链接液晶模块



设置说明:

- (1) RS485 设备地址 (范围 0~255, 在 RS485 网络里面, 设备地址不能相同)
- (2) 串口波特率 (默认波特率 9600)
- (3) 背光亮度 (范围 0~9, 0 为关背光, 9 亮度最高。默认值为 5)
- (4) 前景颜色 (默认为白色)
- (5) 背景颜色 (默认为蓝色)

*注意事项: 前景颜色和背景颜色不能相同, 会导致分不出显示。

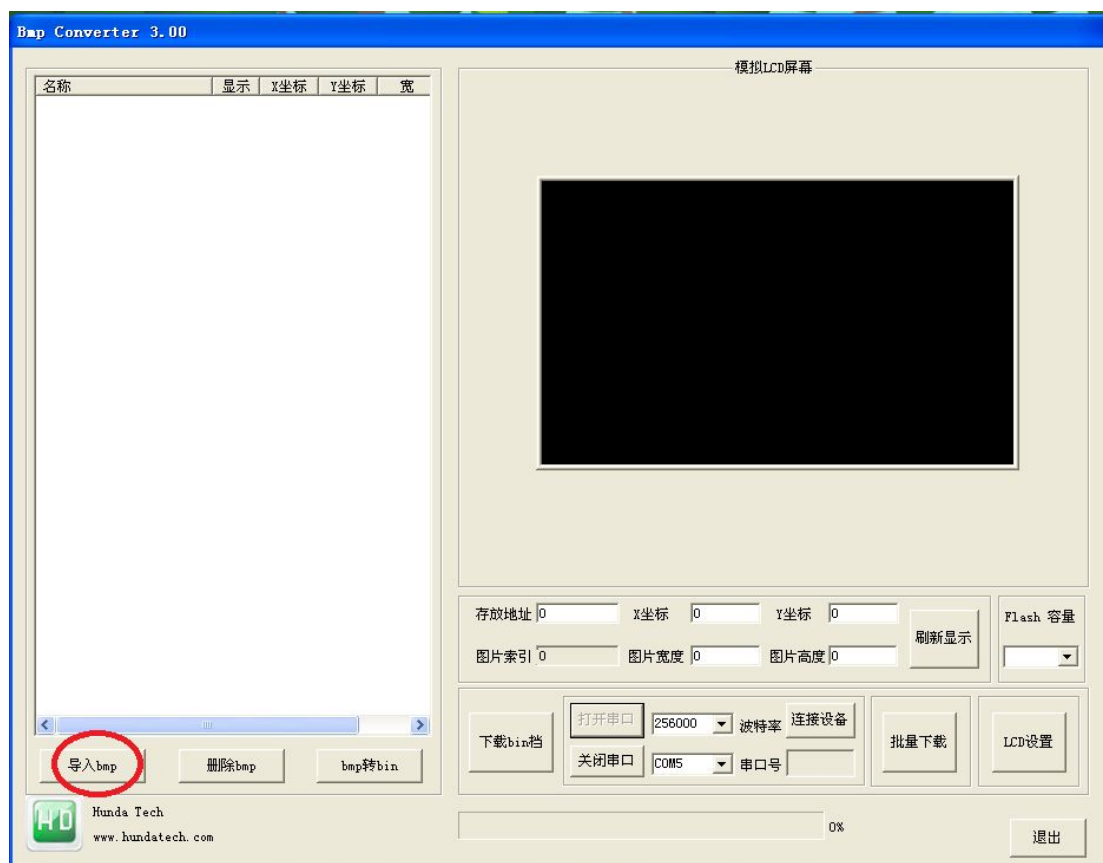
修改默认设置后按“写设置”按键, 把默认设置写到液晶模块。

6. 使用 BmpConverter 做 UI 设计和图片下载:

6.1 开始选择 LCD 尺寸:

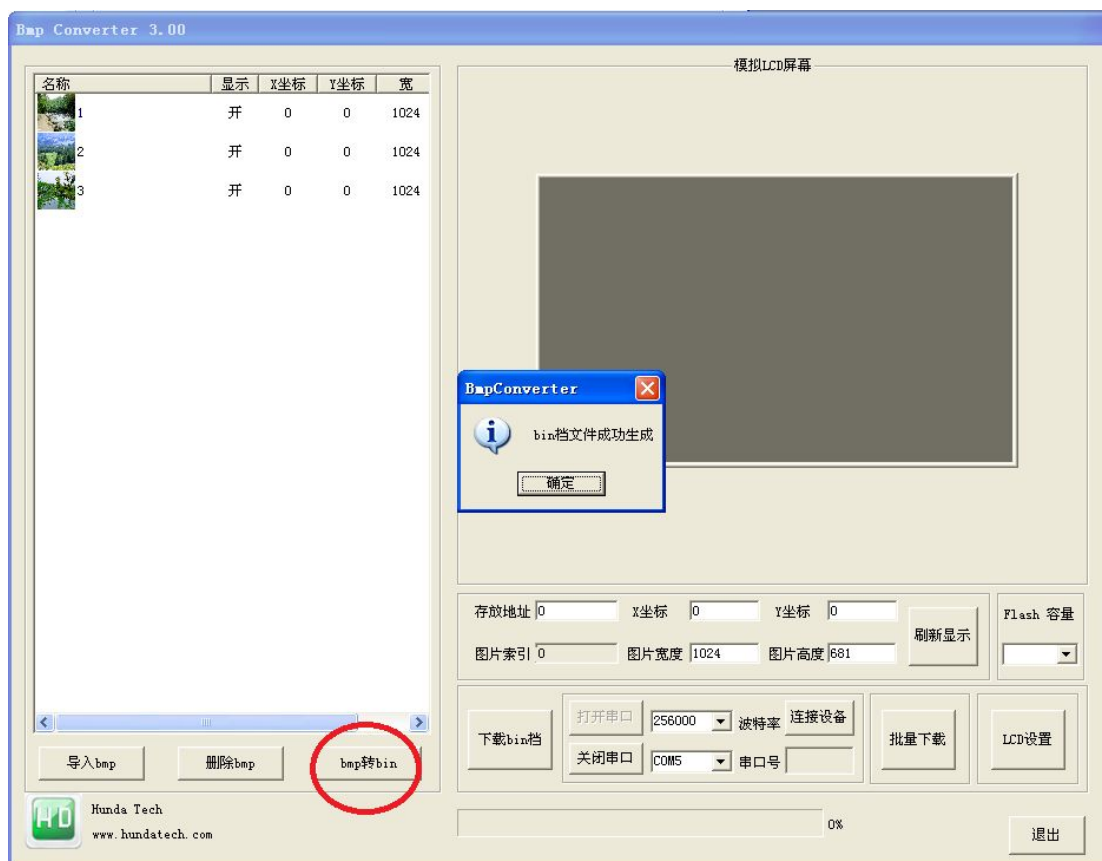
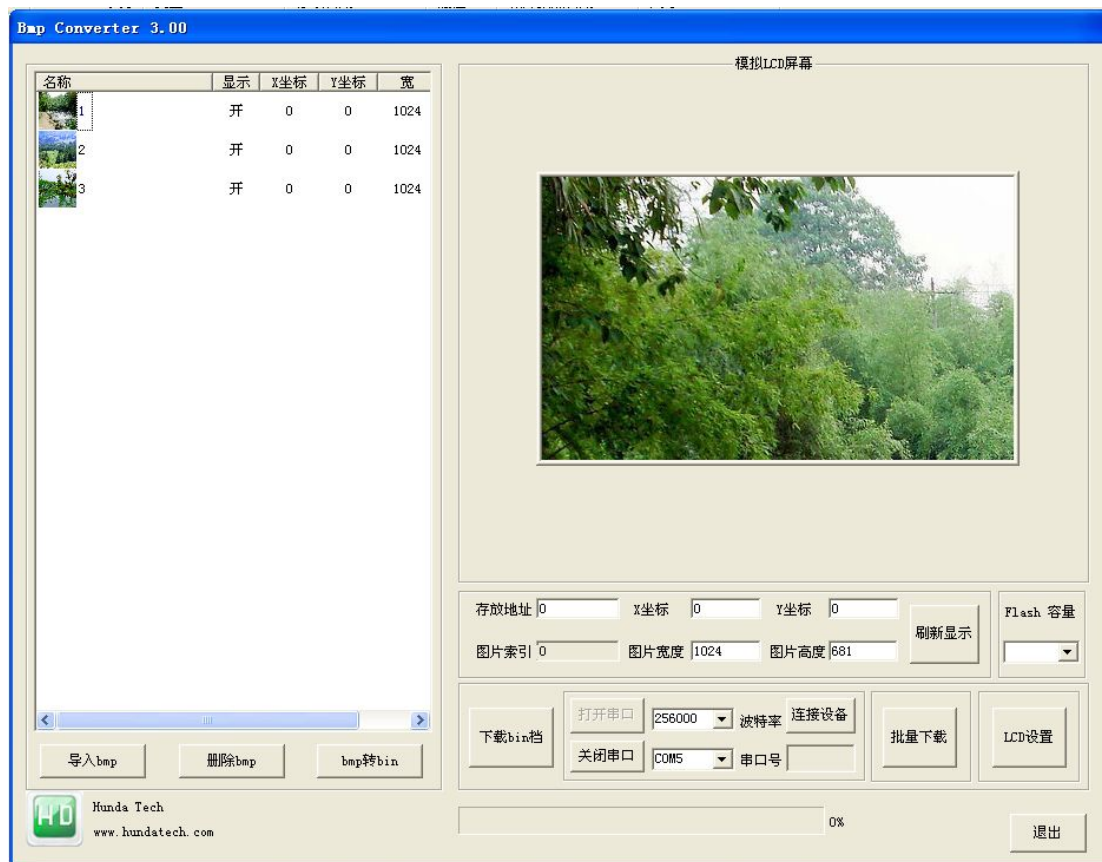


6.2.按导入 bmp,导入 bmp 图片:

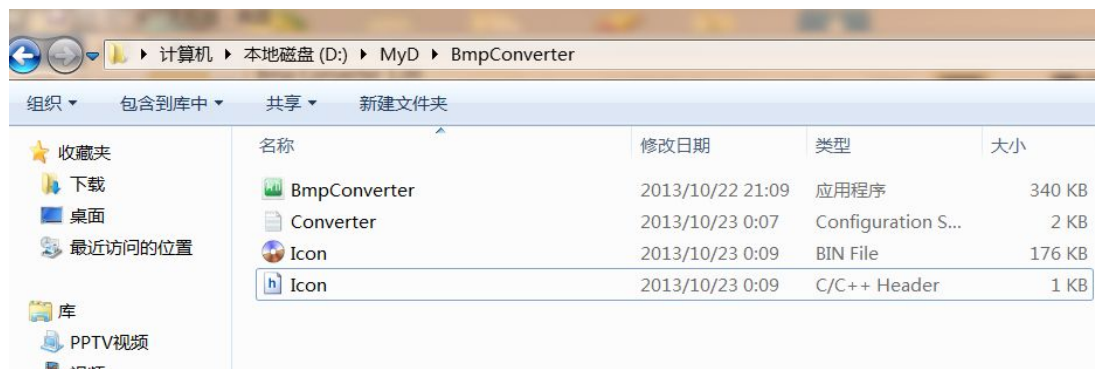


我们可以通过选择左边的图片后，在模拟 lcd 屏里拖动选择的图片，记录对应图片的存放位置和图片长宽。

6.3.按 bmp 转 bin 按键，把图片转成 bin 档和.h 文件：

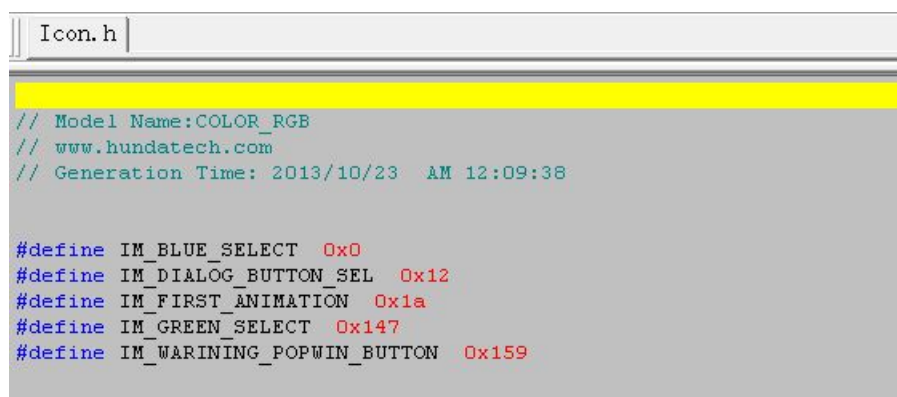


6.4.生成 bin 档：Icon.bin 和 Icon.h:

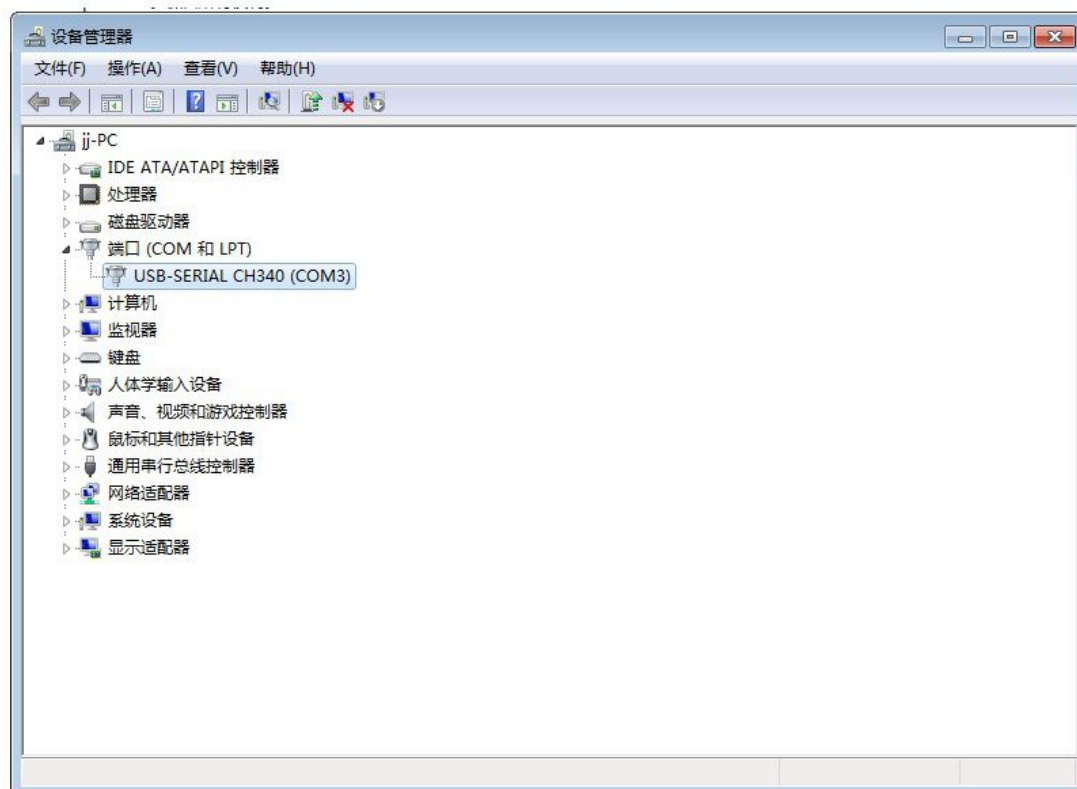


Icon.bin 是相应的图片数据。

Icon.h 里记录 bmp 存储地址，通过该地址让 bmp 在 LCD 液晶模块上显示图片：

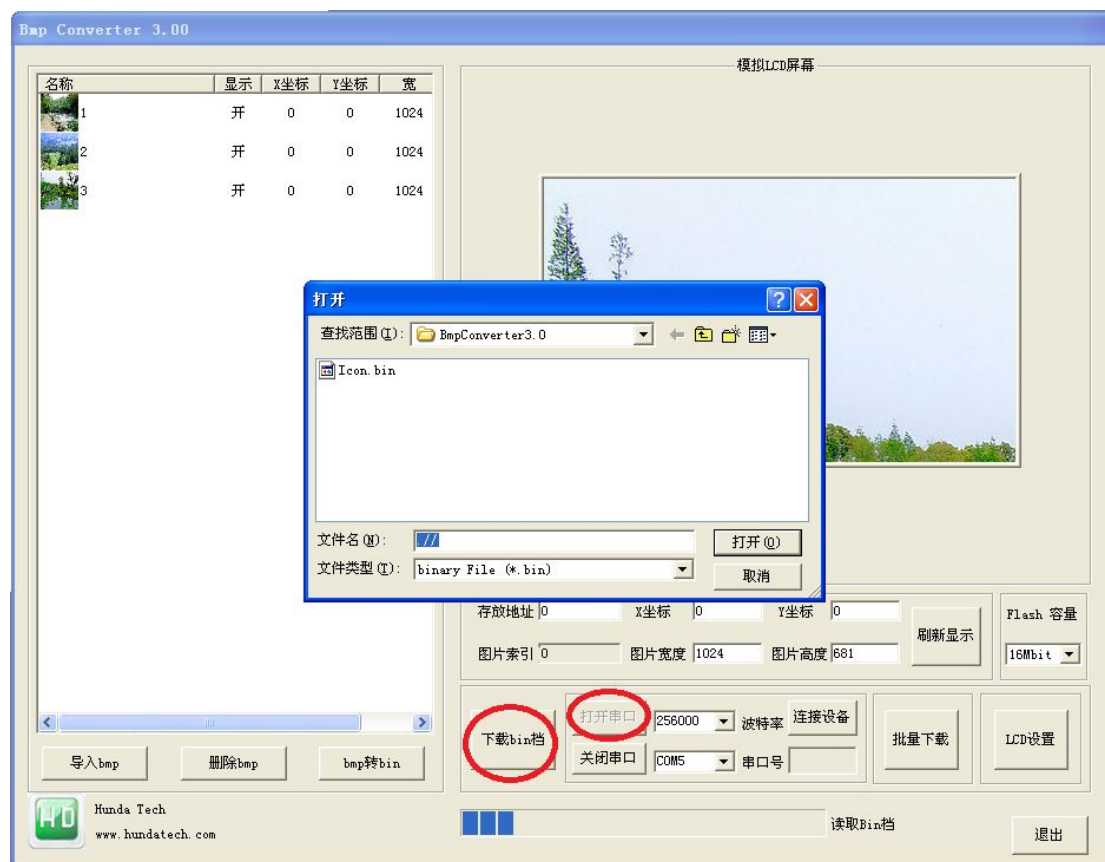


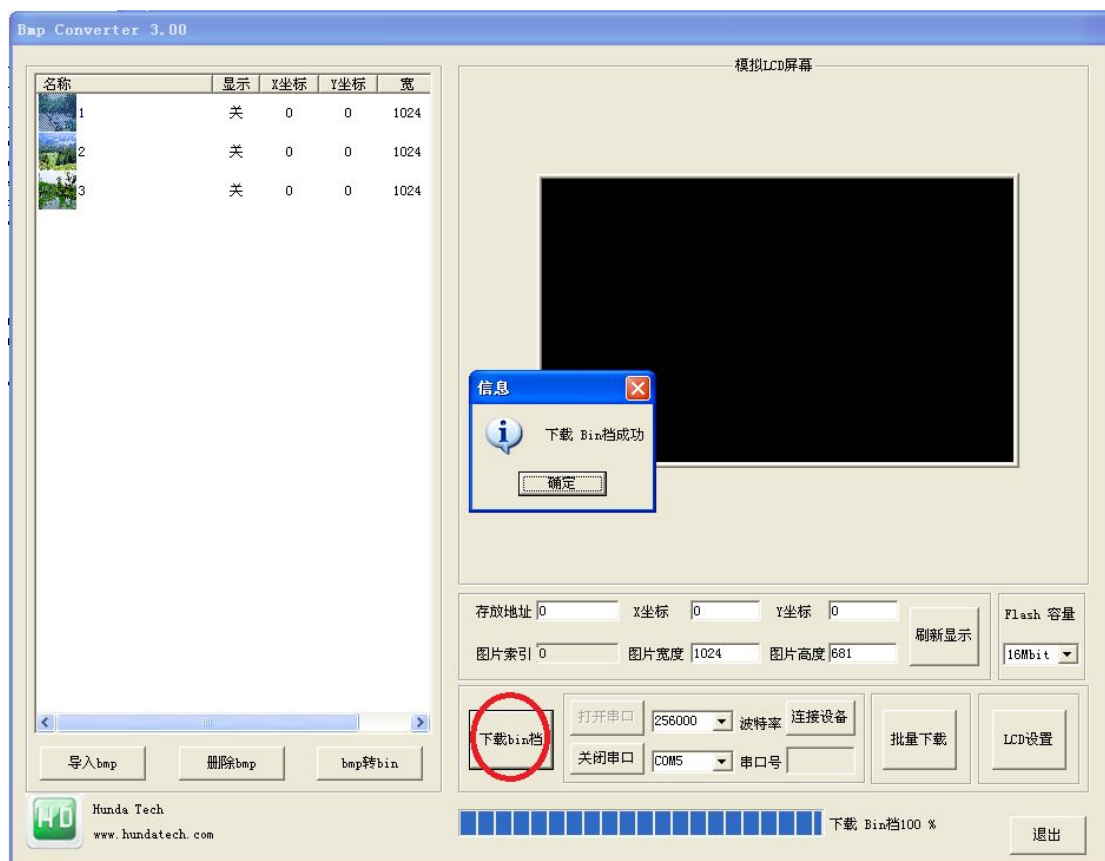
6.5 通过 USB 下载线，LCD 液晶模块链接 PC 后记录对应的串口号：



按打开串口按键，链接 LCD 液晶模块：

6.6.按下载 bin 档按键，下载 bin 档到 LCD 液晶模块的内存中。





注意事项:

ISP 图片下载串口的默认通讯波特率为: 256000 KBPS

7.开发流程

7.1 整理素材

BmpConverter 只支持 BMP 图片。

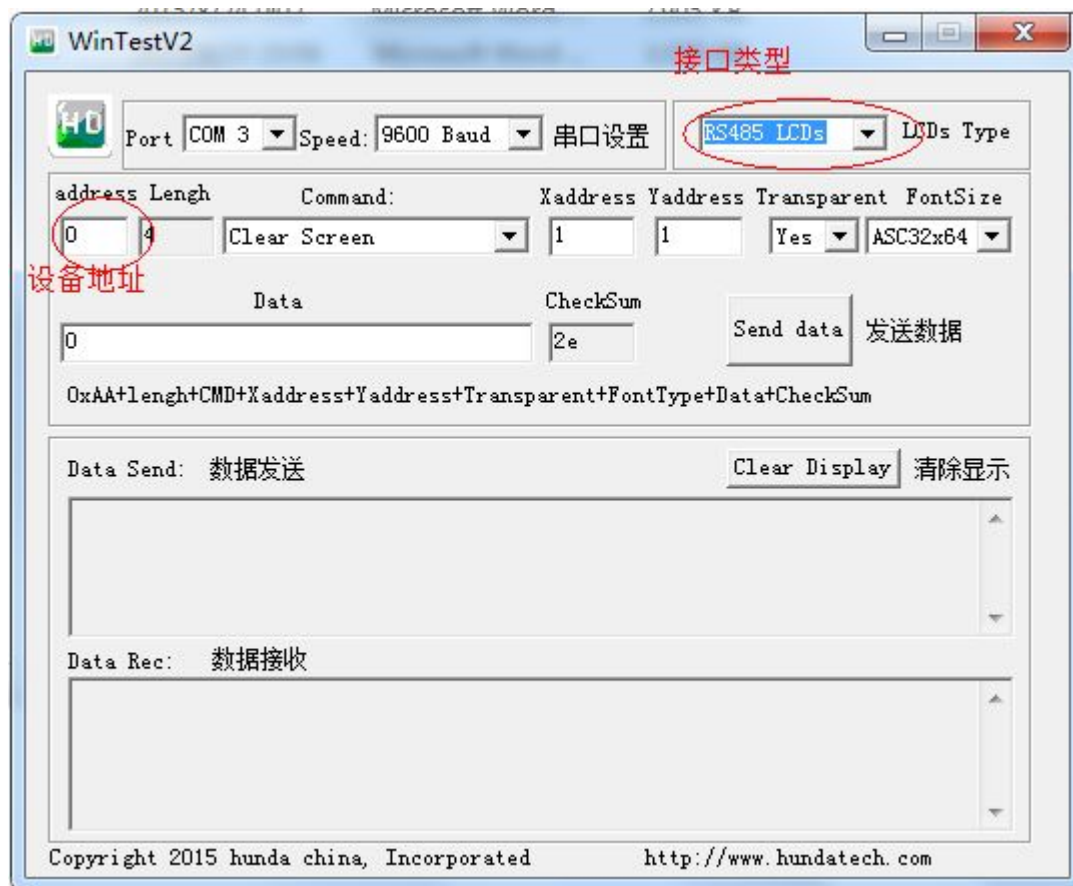
7.2 使用 BmpConverter 把 BMP 图片转为 BIN 档。

7.3 使用串口把图片 BIN 下载到显示模块的 flash 中。

7.4 上电后就可与单片机通讯，并显示彩色图片。



8.使用 Wintest 软件进行测试



注意事项:

1. 串口波特率默认为 9600
2. 选择 lcd 接口类型为 RS485_LCDs
3. RS485 设备地址范围 0~255, 在 RS485 网络内设备地址不能相同。

附录部分

附录 1: ASCII 码表

H/L	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0																
1																
2		!	"	#	\$	%	&	'	(	)	*	+	,	-	.	/
3	0	1	2	3	4	5	6	7	8	9	:	;	<	=	>	?
4	@	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
5	P	Q	R	S	T	U	V	W	X	Y	Z	[	\	]	^	_
6	`	a	b	c	d	e	f	g	h	i	j	k	l	m	n	o
7	p	q	r	s	t	u	v	w	x	y	z	{		}	~	Δ

16*8 半宽字符表

附录 2: 汉字码址表

GB 2312 中文字型碼表

A1A0		.	*	-	-	#	%	-	~		...	'	"
A1B0	"	()	< >	《 》	[]								
A1C0	± × ÷ :	△ ∇	Σ Π	∪ ∩	≡ ≍	√ ∞	∫ ∂	≤ ≥	∞ ∴	∠			
A1D0	⌒ ⊙	f g	= ≈	≈ ∞	∞ ≠	↳ ↲	≤ ≤	≥ ≥	∞ ∴	:			
A1E0	. : †	‡ §	¶ Ⓢ	° ™	\$ %	© ®	‰ №	§	No ☆ ★				
A1F0	● ○	◎ ◊	◆ ◇	□ ■	▲ △	※	→ ↑	↓ ↓	▬ ▮				
A2A0													
A2B0	1.	2.	3.	4.	5.	6.	7.	8.	9.	10.	11.	12.	13.
A2C0	16.	17.	18.	19.	20.	(1)	(2)	(3)	(4)	(5)	(6)	(7)	(8)
A2D0	(9)	(10)	(11)	(12)	(13)	(14)	(15)	(16)	(17)	(18)	(19)	(20)	(21)
A2E0	(22)	(23)	(24)	(25)	(26)	(27)	(28)	(29)	(30)	(31)	(32)	(33)	(34)
A2F0	(35)	(36)	(37)	(38)	(39)	(40)	(41)	(42)	(43)	(44)	(45)	(46)	(47)
A3A0	I	II	III	IV	V	VI	VII	VIII	IX	X	XI	XII	
A3B0	!	"	#	¥	%	&	'	()	*	+	-	.	/
A3C0	@	A	B	C	D	E	F	G	H	I	J	K	L
A3D0	P	Q	R	S	T	U	V	W	X	Y	Z	[\]	^
A3E0	~	a	b	c	d	e	f	g	h	i	j	k	l
A3F0	p	q	r	s	t	u	v	w	x	y	z	()	—
A4A0	あ	い	う	え	お	か	が	き	こ	く			
A4B0	け	こ	ご	さ	し	じ	す	せ	そ	ぞ	た		
A4C0	ち	つ	つ	て	で	と	な	に	ね	の	は		
A4D0	ば	ひ	び	ぶ	ぶ	へ	べ	ば	ば	ま	み		
A4E0	な	ね	も	や	ゆ	よ	ら	り	る	ろ	わ		
A4F0	ぬ	を	ん										
A5A0	ア	イ	ウ	エ	オ	カ	ガ	キ	ク				
A5B0	ケ	コ	サ	シ	ス	セ	ソ	タ					
A5C0	チ	ツ	テ	ト	ニ	ネ	ノ	ハ					
A5D0	バ	ビ	ブ	ヘ	ベ	バ	ボ	マ					
A5E0	メ	モ	ヤ	ユ	ヨ	ラ	リ	ロ					
A5F0	ワ	ヰ	ヴ	カ	ケ								
A6A0	A	B	Γ	Δ	E	Z	H	Θ	I	K	Λ	M	N
A6B0	Π	P	Σ	T	Υ	Φ	X	Ψ	Ω				
A6C0	~	α	β	γ	δ	ε	ζ	η	θ	ι	κ	λ	μ
A6D0	ν	π	ρ	σ	τ	υ	φ	χ	ψ	ω			
A6E0													
A6F0													
A7A0	A	B	В	Г	Д	E	Ж	З	И	П	К	Л	М
A7B0	О	П	Р	С	Т	У	Ф	Х	Ц	Ч	Ш	Щ	Ъ
A7C0	Ы	Э											
A7D0	a	б	в	г	д	e	ж	з	и	п	к	л	м
A7E0	о	п	р	с	т	у	ф	х	ц	ч	ш	щ	ъ
A7F0	ы	я											
A8A0	ā	â	ã	ä	å	ē	ê	ë	ì	í	î	ï	ð
A8B0	ō	ŏ	ú	û	ü	ù	ū	û	ü	ē	ē	ē	ē
A8C0	G	g											
A8D0	く	丁	止	オ	尸	尸	ち	ム	丫	ぞ	せ	ろ	フ
A8E0	么	ヌ	マ	ラ	尤	ル	儿	メ	口				
A8F0													
A9A0													
A9B0	Г	г	Г	г	Г	г	Г	г	Г	г	Г	г	Г
A9C0	Г	г	Г	г	Г	г	Г	г	Г	г	Г	г	Г
A9D0	Г	г	Г	г	Г	г	Г	г	Г	г	Г	г	Г
A9E0	Г	г	Г	г	Г	г	Г	г	Г	г	Г	г	Г
A9F0													
B0A0	啊	阿	埃	挨	哎	唉	哀	絕	癌	萬	矮	艾	碍
B0B0	氣	鞍	安	俺	按	暗	岸	案	肮	昂	盍	凹	熬
B0C0	做	傲	奧	懊	澳	捌	扒	叭	吧	筈	八	葩	拔

B0D0	把班椰	把班椰
B0E0	把班椰	把班椰
B0F0	把班椰	把班椰
B1A0	把班椰	把班椰
B1B0	把班椰	把班椰
B1C0	把班椰	把班椰
B1D0	把班椰	把班椰
B1E0	把班椰	把班椰
B1F0	把班椰	把班椰
B2A0	把班椰	把班椰
B2B0	把班椰	把班椰
B2C0	把班椰	把班椰
B2D0	把班椰	把班椰
B2E0	把班椰	把班椰
B2F0	把班椰	把班椰
B3A0	把班椰	把班椰
B3B0	把班椰	把班椰
B3C0	把班椰	把班椰
B3D0	把班椰	把班椰
B3E0	把班椰	把班椰
B3F0	把班椰	把班椰
B4A0	把班椰	把班椰
B4B0	把班椰	把班椰
B4C0	把班椰	把班椰
B4D0	把班椰	把班椰
B4E0	把班椰	把班椰
B4F0	把班椰	把班椰
B5A0	把班椰	把班椰
B5B0	把班椰	把班椰
B5C0	把班椰	把班椰
B5D0	把班椰	把班椰
B5E0	把班椰	把班椰
B5F0	把班椰	把班椰
B6A0	把班椰	把班椰
B6B0	把班椰	把班椰
B6C0	把班椰	把班椰
B6D0	把班椰	把班椰
B6E0	把班椰	把班椰
B6F0	把班椰	把班椰
B7A0	把班椰	把班椰
B7B0	把班椰	把班椰
B7C0	把班椰	把班椰
B7D0	把班椰	把班椰
B7E0	把班椰	把班椰
B7F0	把班椰	把班椰
B8A0	把班椰	把班椰
B8B0	把班椰	把班椰
B8C0	把班椰	把班椰
B8D0	把班椰	把班椰
B8E0	把班椰	把班椰
B8F0	把班椰	把班椰
B9A0	把班椰	把班椰
B9B0	把班椰	把班椰
B9C0	把班椰	把班椰
B9D0	把班椰	把班椰
B9E0	把班椰	把班椰
B9F0	把班椰	把班椰
BAA0	把班椰	把班椰
BAB0	把班椰	把班椰

BAC0 豪盒亨吼
BAD0 话焕恍秒火
BAE0 话焕恍秒火
BAF0 话焕恍秒火
BB00 话焕恍秒火
BBC0 话焕恍秒火
BBD0 话焕恍秒火
BBE0 话焕恍秒火
BBF0 话焕恍秒火
BCA0 及祭夹监拣
BCB0 及祭夹监拣
BCD0 及祭夹监拣
BCE0 及祭夹监拣
BCF0 及祭夹监拣
BDA0 及祭夹监拣
BDB0 及祭夹监拣
BDC0 及祭夹监拣
BDD0 及祭夹监拣
BDE0 及祭夹监拣
BDF0 及祭夹监拣
BEA0 及祭夹监拣
BEB0 及祭夹监拣
BEC0 及祭夹监拣
BED0 及祭夹监拣
BEE0 及祭夹监拣
BEF0 及祭夹监拣
BFA0 及祭夹监拣
BFB0 及祭夹监拣
BFC0 及祭夹监拣
BFD0 及祭夹监拣
BFE0 及祭夹监拣
BFF0 及祭夹监拣
COA0 及祭夹监拣
COB0 及祭夹监拣
COC0 及祭夹监拣
COD0 及祭夹监拣
COE0 及祭夹监拣
COF0 及祭夹监拣
C1A0 及祭夹监拣
C1B0 及祭夹监拣
C1C0 及祭夹监拣
C1D0 及祭夹监拣
C1E0 及祭夹监拣
C1F0 及祭夹监拣
C2A0 及祭夹监拣
C2B0 及祭夹监拣
C2C0 及祭夹监拣
C2D0 及祭夹监拣
C2E0 及祭夹监拣
C2F0 及祭夹监拣
C3A0 及祭夹监拣
C3B0 及祭夹监拣
C3C0 及祭夹监拣
C3D0 及祭夹监拣
C3E0 及祭夹监拣
C3F0 及祭夹监拣
C4A0 及祭夹监拣

C4B0 及祭夹监拣
C4C0 及祭夹监拣
C4D0 及祭夹监拣
C4E0 及祭夹监拣
C5A0 及祭夹监拣
C5B0 及祭夹监拣
C5C0 及祭夹监拣
C5D0 及祭夹监拣
C5E0 及祭夹监拣
C5F0 及祭夹监拣
C6A0 及祭夹监拣
C6B0 及祭夹监拣
C6C0 及祭夹监拣
C6D0 及祭夹监拣
C6E0 及祭夹监拣
C6F0 及祭夹监拣
C7A0 及祭夹监拣
C7B0 及祭夹监拣
C7C0 及祭夹监拣
C7D0 及祭夹监拣
C7E0 及祭夹监拣
C7F0 及祭夹监拣
C8A0 及祭夹监拣
C8B0 及祭夹监拣
C8C0 及祭夹监拣
C8D0 及祭夹监拣
C8E0 及祭夹监拣
C8F0 及祭夹监拣
C9A0 及祭夹监拣
C9B0 及祭夹监拣
C9C0 及祭夹监拣
C9D0 及祭夹监拣
C9E0 及祭夹监拣
C9F0 及祭夹监拣
CAA0 及祭夹监拣
CAB0 及祭夹监拣
CAC0 及祭夹监拣
CAD0 及祭夹监拣
CAE0 及祭夹监拣
CAF0 及祭夹监拣
CBA0 及祭夹监拣
CBB0 及祭夹监拣
CBC0 及祭夹监拣
CBD0 及祭夹监拣
CBE0 及祭夹监拣
CBF0 及祭夹监拣
CCA0 及祭夹监拣
CCB0 及祭夹监拣
CCC0 及祭夹监拣
CCD0 及祭夹监拣
CCE0 及祭夹监拣
CCF0 及祭夹监拣
CDA0 及祭夹监拣
CDB0 及祭夹监拣
CDC0 及祭夹监拣
CDD0 及祭夹监拣
CDE0 及祭夹监拣
CDF0 及祭夹监拣

[illegible][illegible]

[illegible]