

串口指令集

首次发行日期 2012-10-31

☒ 暂订版规格书

☐ 正式版规格书

核准	确认	备注	
		版本	V1.6a
		Updated	2017-04-07

改版记录

版本	更新日期	页数	改版内容
V1.5	2015-11-12	44	添加二维码生成与显示功能
V1.5A	2016-02-25	无	把内部正负压 PWM 调节串口指令屏蔽掉，是固定正负压的 PWM 频率和占空比，防止客户调节产生白屏现象
	2016-02-25	42	增加不保存配置信息的背光调节指令
V1.6	2016-5-10	12	为增加通信的可靠性增加了指令集的 CRC 校验
V1.6a	2017-04-07		更正指引序号错误

目录

1. 串口说明.....	6
1.0 CRC16 校验码.....	6
1.1 串口的工作模式.....	8
1.2 数据帧结构.....	8
1.3 字节传送顺序.....	8
2. 指令速查表.....	9
3. 指令说明.....	13
3.0 CRC16 校验码(0x7c).....	13
3.1 握手.....	14
3.1.1. 握手指令(0x00).....	14
3.2 显示参数配置.....	14
3.2.1. 设置前景色和背景色(0x50).....	14
3.2.2. 设置字符的显示间距(0x51).....	15
3.2.3. 取色到背景色调色板(0x52).....	15
3.2.4. 取色到前景色调色板(0x53).....	16
3.2.5. 设置光标的模式(0x54).....	16
3.3 文本显示.....	17
3.3.1. 8x8 点阵 ASCII 字符(0x63).....	17
3.3.2. 16x16 点阵 GBK 字符(0x64).....	17
3.3.3. 32x32 点阵 GB2312 字符(0x65).....	18
3.3.4. 12x12 点阵 GBK 字符(0x7e).....	18
3.3.5. 24x24 点阵 GB2312 字符(0x7f).....	19
3.3.6. 任意字符(0x98).....	19
3.4 置点.....	22
3.4.1. 背景色置多个点(0x60).....	22
3.4.2. 前景色置多个点(0x61).....	22
3.4.3. 动态曲线(0x44).....	22
3.4.4. 直接显存操作(0x42).....	23
3.5 线段和多边形.....	24
3.5.1. 把指定的点用前景色线段连接(0x66).....	24
3.5.2. 把指定的点用背景色线段连接(0x6d).....	24
3.5.3. 显示多条同底垂直线(0x45).....	25
3.5.4. 显示折线(0x46).....	25
3.6 圆弧和圆域.....	26
3.6.1. 反色显示或填充圆弧或圆域(0x67).....	26
3.6.2. 反色显示或填充圆弧或圆域(0x67).....	27
3.7 矩形框.....	28
3.7.1. 前景色显示矩形框(0x69).....	28
3.7.2. 背景色显示矩形框(0x79).....	28
3.8 区域操作.....	29
3.8.1. 区域填充(0x74).....	29

3.8.2.	清屏(0x62).....	29
3.8.3.	背景色填充矩形区域(0x6a).....	29
3.8.4.	前景色填充矩形区域(0x6b).....	30
3.8.5.	反色矩形区域(0x6c).....	30
3.8.6.	矩形区域左环移(0x70).....	31
3.8.7.	矩形区域右环移(0x71).....	32
3.8.8.	矩形区域左移(0x72).....	32
3.8.9.	矩形区域右移(0x73).....	33
3.9	图和图标显示.....	33
3.9.1.	显示闪存里的一幅图片(0x40).....	33
3.9.2.	从闪存里的一幅图片中剪切一部分显示(0x41).....	34
3.9.3.	保存当前画面到闪存(0xe2).....	34
3.9.4.	显示多个闪存里的图标(0x99).....	35
3.10	自动执行指令.....	35
3.10.1.	关闭或打开自动执行指令(0x9a).....	35
3.11	闪存操作.....	36
3.11.1.	修改字库(0xf2).....	36
3.11.2.	写数据到闪存里的用户数据区(0x90).....	38
3.11.3.	从闪存里的用户数据区读数据(0x91).....	39
3.12	键盘操作.....	40
3.12.1.	键码上传(0x71).....	40
3.12.2.	配置键码(0xe5).....	41
3.13	触摸屏操作.....	41
3.13.1.	松开触摸屏时最后一次上传坐标(0x72).....	41
3.13.2.	按下触摸屏时上传坐标(0x73).....	41
3.13.3.	触摸屏校准(0xe4).....	42
3.13.4.	在触屏自动切换模式下自动上传预设键码(0x78).....	42
3.13.5.	在触控界面自动切换模式下自动上传预设键码(0x79).....	43
3.14	时间操作.....	43
3.14.1.	开启/关闭时间显示或读取时间(0x9b).....	43
3.14.2.	设置时间(0xe7).....	45
3.14.3.	闹钟设定(0xe8).....	45
3.15	参数配置.....	47
3.15.1.	参数配置(0xe0).....	47
3.16	蜂鸣器控制.....	51
3.16.1.	蜂鸣器控制(0x49).....	51
3.17	背光控制.....	51
3.17.1.	背光控制说明.....	51
3.17.2.	背光使能 POWER_EN 控制(0x6e).....	52
3.17.3.	触控背光亮度 PWM 控制(0x6e).....	52
3.17.4.	背光亮度 PWM 调节(0x6f).....	53
3.18	二维码生成与显示.....	54

3.18 二维码生成与显示(0x68).....	54
4. 配置文件介绍.....	55
4.0 系统配置文件(0x00 , 0x19).....	55
4.1 触控自动执行指令配置文件(0x1A).....	55
4.2 按键自动切换配置文件(0x1B).....	56
4.3 自动循环执行指令组配置文件(0x1C).....	56
4.4 图标配置文件(0x1D).....	57
4.5 触控自动切换配置文件(0x1E).....	58

串口说明

CRC16 校验码

为了提高串口通讯的可靠性，可开启 CRC16 校验码功能。开启该功能后，发送的串口指令均需要在数据段与帧尾之间添加两字节的 CRC16 校验码值(数据帧结构的数据块 4)后方可正常通讯。

CRC16 校验码值是由数据帧结构(Table1-1)的数据块 2、数据块 3 参与 CRC16 协议计算所得，高位在前，低位在后。同时串口返回的数据(客户端接到的串口数据)在数据帧结构的数据块 4 添加了校验码。

CRC16 举例：

如在显示屏的坐标值为(100,200)处显示 16*16 GBK16 码“123”的 0x64 指令时，未加 CRC16 校验码时发送：

AA 64 00 64 00 C8 31 32 33 CC 33 C3 3C 即可在(100,200)处显示 16*16 GBK 码字库的“123”。

如开启 CRC16 校验码后时,先对上面数据帧结构的数据块 2、3 的数据 64 00 64 00 C8 31 32 33 进行 CRC16 协议计算，计算后所得到的值为 2E75 的 CRC16 校验码，把此校验码放在数据帧结构的数据块 4 位置即得到开启 CRC16 校验码后需发送的指令：AA 64 00 64 00 C8 31 32 33 2E 75 CC 33 C3 3C。

以下是 CRC16 协议算法所得到的 CRC16 校验码函数，客户可参考此算法计算而得到相应的 CRC16 校验码。

```
static const unsigned char code aucCRCHi[] =
{
    0x00, 0xC1, 0x81, 0x40, 0x01, 0xC0, 0x80, 0x41, 0x01, 0xC0, 0x80, 0x41,
    0x00, 0xC1, 0x81, 0x40, 0x01, 0xC0, 0x80, 0x41, 0x00, 0xC1, 0x81, 0x40,
    0x00, 0xC1, 0x81, 0x40, 0x01, 0xC0, 0x80, 0x41, 0x01, 0xC0, 0x80, 0x41,
    0x00, 0xC1, 0x81, 0x40, 0x00, 0xC1, 0x81, 0x40, 0x01, 0xC0, 0x80, 0x41,
    0x00, 0xC1, 0x81, 0x40, 0x01, 0xC0, 0x80, 0x41, 0x01, 0xC0, 0x80, 0x41,
    0x00, 0xC1, 0x81, 0x40, 0x01, 0xC0, 0x80, 0x41, 0x00, 0xC1, 0x81, 0x40,
    0x00, 0xC1, 0x81, 0x40, 0x01, 0xC0, 0x80, 0x41, 0x00, 0xC1, 0x81, 0x40,
    0x01, 0xC0, 0x80, 0x41, 0x01, 0xC0, 0x80, 0x41, 0x00, 0xC1, 0x81, 0x40,
    0x00, 0xC1, 0x81, 0x40, 0x01, 0xC0, 0x80, 0x41, 0x01, 0xC0, 0x80, 0x41,
    0x00, 0xC1, 0x81, 0x40, 0x01, 0xC0, 0x80, 0x41, 0x00, 0xC1, 0x81, 0x40,
    0x00, 0xC1, 0x81, 0x40, 0x01, 0xC0, 0x80, 0x41, 0x01, 0xC0, 0x80, 0x41,
    0x00, 0xC1, 0x81, 0x40, 0x00, 0xC1, 0x81, 0x40, 0x01, 0xC0, 0x80, 0x41,
}
```

```

0x00, 0xC1, 0x81, 0x40, 0x01, 0xC0, 0x80, 0x41, 0x01, 0xC0, 0x80, 0x41,
0x00, 0xC1, 0x81, 0x40, 0x00, 0xC1, 0x81, 0x40, 0x01, 0xC0, 0x80, 0x41,
0x01, 0xC0, 0x80, 0x41, 0x00, 0xC1, 0x81, 0x40, 0x01, 0xC0, 0x80, 0x41,
0x00, 0xC1, 0x81, 0x40, 0x00, 0xC1, 0x81, 0x40, 0x01, 0xC0, 0x80, 0x41,
0x00, 0xC1, 0x81, 0x40, 0x01, 0xC0, 0x80, 0x41, 0x01, 0xC0, 0x80, 0x41,
0x00, 0xC1, 0x81, 0x40, 0x01, 0xC0, 0x80, 0x41, 0x00, 0xC1, 0x81, 0x40,
0x00, 0xC1, 0x81, 0x40, 0x01, 0xC0, 0x80, 0x41, 0x01, 0xC0, 0x80, 0x41,
0x00, 0xC1, 0x81, 0x40, 0x00, 0xC1, 0x81, 0x40, 0x01, 0xC0, 0x80, 0x41,
0x00, 0xC1, 0x81, 0x40, 0x01, 0xC0, 0x80, 0x41, 0x01, 0xC0, 0x80, 0x41,
0x00, 0xC1, 0x81, 0x40
};

static const unsigned char code aucCRCLo[] =
{
    0x00, 0xC0, 0xC1, 0x01, 0xC3, 0x03, 0x02, 0xC2, 0xC6, 0x06, 0x07, 0xC7,
    0x05, 0xC5, 0xC4, 0x04, 0xCC, 0x0C, 0x0D, 0xCD, 0x0F, 0xCF, 0xCE, 0x0E,
    0x0A, 0xCA, 0xCB, 0x0B, 0xC9, 0x09, 0x08, 0xC8, 0xD8, 0x18, 0x19, 0xD9,
    0x1B, 0xDB, 0xDA, 0x1A, 0x1E, 0xDE, 0xDF, 0x1F, 0xDD, 0x1D, 0x1C, 0xDC,
    0x14, 0xD4, 0xD5, 0x15, 0xD7, 0x17, 0x16, 0xD6, 0xD2, 0x12, 0x13, 0xD3,
    0x11, 0xD1, 0xD0, 0x10, 0xF0, 0x30, 0x31, 0xF1, 0x33, 0xF3, 0xF2, 0x32,
    0x36, 0xF6, 0xF7, 0x37, 0xF5, 0x35, 0x34, 0xF4, 0x3C, 0xFC, 0xFD, 0x3D,
    0xFF, 0x3F, 0x3E, 0xFE, 0xFA, 0x3A, 0x3B, 0xFB, 0x39, 0xF9, 0xF8, 0x38,
    0x28, 0xE8, 0xE9, 0x29, 0xEB, 0x2B, 0x2A, 0xEA, 0xEE, 0x2E, 0x2F, 0xEF,
    0x2D, 0xED, 0xEC, 0x2C, 0xE4, 0x24, 0x25, 0xE5, 0x27, 0xE7, 0xE6, 0x26,
    0x22, 0xE2, 0xE3, 0x23, 0xE1, 0x21, 0x20, 0xE0, 0xA0, 0x60, 0x61, 0xA1,
    0x63, 0xA3, 0xA2, 0x62, 0x66, 0xA6, 0xA7, 0x67, 0xA5, 0x65, 0x64, 0xA4,
    0x6C, 0xAC, 0xAD, 0x6D, 0xAF, 0x6F, 0x6E, 0xAE, 0xAA, 0x6A, 0x6B, 0xAB,
    0x69, 0xA9, 0xA8, 0x68, 0x78, 0xB8, 0xB9, 0x79, 0xBB, 0x7B, 0x7A, 0xBA,
    0xBE, 0x7E, 0x7F, 0xBF, 0x7D, 0xBD, 0xBC, 0x7C, 0xB4, 0x74, 0x75, 0xB5,
    0x77, 0xB7, 0xB6, 0x76, 0x72, 0xB2, 0xB3, 0x73, 0xB1, 0x71, 0x70, 0xB0,
    0x50, 0x90, 0x91, 0x51, 0x93, 0x53, 0x52, 0x92, 0x96, 0x56, 0x57, 0x97,
    0x55, 0x95, 0x94, 0x54, 0x9C, 0x5C, 0x5D, 0x9D, 0x5F, 0x9F, 0x9E, 0x5E,
    0x5A, 0x9A, 0x9B, 0x5B, 0x99, 0x59, 0x58, 0x98, 0x88, 0x48, 0x49, 0x89,
    0x4B, 0x8B, 0x8A, 0x4A, 0x4E, 0x8E, 0x8F, 0x4F, 0x8D, 0x4D, 0x4C, 0x8C,
    0x44, 0x84, 0x85, 0x45, 0x87, 0x47, 0x46, 0x86, 0x82, 0x42, 0x43, 0x83,
    0x41, 0x81, 0x80, 0x40
};

/* 输入:
    pucFrame—参与计算的数据段数据,即为 Table 1-1 结构的数据块 2、3.
    ucLen-----数据段的长度.

输出: 所计算得到的 CRC16 校验码 */
unsigned short usCalculateCrc16(unsigned char* pucFrame,unsigned char ucLen)
{
    unsigned char    ucCRCHi = 0xFF;

```

```

unsigned char    ucCRCLo = 0xFF;
unsigned char    iIndex;

while( ucLen-- )
{
    iIndex = ucCRCLo ^ *(pucFrame++);
    ucCRCLo = (unsigned char)(ucCRCHi ^ aucCRCHi[iIndex]);
    ucCRCHi = aucCRCLo[iIndex];
} return (unsigned short)(ucCRCHi << 8 | ucCRCLo);
}

```

串口的工作模式

本 TFT 驱动模组采用异步的全双工串口作为通信接口。每个数据传输为 10 位，包括 1 位起始位，8 位数据位和 1 位停止位。默认波特率为 115,200bps，用户可以使用 0xe0 指令修改此值。

数据帧结构

串口采用以帧为单位的通信方式。数据帧由 4 个数据块组成，如下表所示：

Table 1-1

数据块	1	2	3	4(开启 CRC 校码时)	5
说明	帧头	指令 (1 字节)	数据 (最大 248 字节)	CRC16 校验码(2 字节)	帧尾
	0xaa				0xcc 0x33 0xc3 0x3c

字节传送顺序

数据的传输字节顺序采用 big-ending 方式。也就是说，高字节在前，低字节在后，比如，要传输 0x6677，那么需要先传输 0x66，后传输 0x77。

指令速查表

Table 2- 1

类别	指令	指令参数	说明
CRC16 校验	0x7C	0100	开启 CRC16 校验
	0x7C	00 00 18 B0	关闭 CRC16 校验
握手	0x00	无或 00	查看配置和版本信息
显示参数配置	0x50	FColor+BColor	设置前景色和背景色
	0x51	XDistance+YDistance	设置字符的显示间距
	0x52	X+Y	取色到背景色调色板
	0x53	X+Y	取色到前景色调色板
	0x54	Cursor_En X Y C_Width C_Height	设置光标的模式
文本显示	0x63	X+Y+String	8x8 点阵 ASCII 字符
	0x64		16x16 点阵 GBK 字符
	0x65		32x32 点阵 GB2312 字符
	0x7e		12x12 点阵 GBK 字符
	0x7f		24x24 点阵 GB2312 字符
	0x98	X+Y+ID+Mode+Dot+FColor+BColor+String	任意字符
置点	0x60	(X0+Y0)+...+(Xn+Yn)	背景色置多个点
	0x61		前景色置多个点
	0x44	X+YStart+YEnd+BColor+(Y0+FColor0)+...+(Yn+ FColorN)	动态曲线
	0x42	Address+Color0+...+ColorN	直接显存操作
线段和多边形	0x66	(X0+Y0)+...+(Xn+Yn)	把指定点用前景色线段连接
	0x6d		把指定点用背景色线段

	0x45	$X+Y+HeightMax+Height0+...+HeightN$	连接 显示多条同底垂直线 (频谱)
	0x46	$X+XIncrement+Y0+...+Yn$	显示折线
圆弧和圆域	0x67	$(Type0+X0+Y0+r0)+...+(TypeN+Xn+Yn+rN)$	反色/显示 多个圆弧或圆域
矩形框	0x69	$(XStart0+YStart0+XEnd0+YEnd0)+...+$	前景色显示多个矩形框
	0x79	$(XStartN+YStartN+XendN+YEndN)$	背景色显示多个矩形框
区域操作	0x74	$X+Y+Color$	指定区域填充
	0x62	无	清屏
	0x6a	$(XStart0+YStart0+XEnd0+YEnd0)+...+$	背景色填充多个矩形区域
		$(XStartN+YStartN+XendN+YEndN)$	
	0x6b		前景色填充多个矩形区域
	0x6c		反色多个矩形区域
	0x70		多个矩形区域左环移
	0x71		多个矩形区域右环移
	0x72		多个矩形区域左移
	0x73		多个矩形区域右移
图片和图标显示	0x40	Pic_ID	显示保存在闪存的一幅图片
	0x41	$Pic_ID+XStart+YStart+XEnd+YEnd+X+Y$	从保存在闪存的一幅图片剪切一部分显示
	0xe2	Pic_ID	保存当前屏幕到闪存
	0x99	$(X0+Y0+IconID0)+...+(Xn+Yn+IconIDn)$	显示多个保存在闪存中的图标

自动执行指令	0x9a	0xff/PackID	关闭或打开自动执行指令
闪存操作	0xf2	0xf2+0x2f+0x5a+0xa5+ID	修改字库
	0x90	0x55+0xaa+0x5a+0xa5+Address+Data	写数据到闪存里的用户数据区
	0x91	Address+Length	从闪存里的用户数据区读数据
键盘操作	0x71	KeyCode	键码上传
	0xe5	0x55+0xaa+0x5a+0xa5+KeyCode	配置键码
触摸屏操作	0x72	TouchX+TouchY	触摸屏松开时上传坐标
	0x73		触摸屏按下时上传坐标
	0xe4	0x55+0xaa+0x5a+0xa5	触摸屏校准
	0x78	TouchCode	触摸屏松开时上传键码
	0x79	TouchCode	触摸屏按下时上传键码
时间操作	0x9b	Operation	开启/关闭时间显示或读取时间
	0xe7	0x55+0xaa+0x5a+0xa5+YY:MM:DD:HH:MM:SS	设置时间
	0xe8	Num Week Hour Minute Mode	闹钟设定
参数配置	0xe0	0x55+0xaa+0x5a+0xa5+TFT_ID+BaudrateID+Para1+(Para2)	参数配置
蜂鸣器控制	0x49	OnTime	蜂鸣器鸣响时间控制

背光控制	0x6E	无或 0x55+0xaa+0x5a+0xa5+ V_ON+V_OFF+ON_TIME	关闭或设置键控触控背光模式
	0x6F	无或 PWM_Duty 或 PWM_DivH+PWM_DivL	打开或 PWM 方式调节背光亮度或设定屏的 PWM 频率
LCD 参数 设定	0x5f	HS_LF+HS_PP+HS_RT+VS_LF+VS_PP+VS_RT+POS_CLK	设置 LCD 前肩后肩等参数

指令说明

CRC16 校验码(0x7c)

开启 CRC16 串口指令校验

发送: AA 7C 01 00 CC 33 C3 3C

接收: AA 7C 01 00 CC 33 C3 3C

串口屏上电默认为没有开启串口指令校验功能的，如果要开启串口指令校验功能的话，接下来发送的数据都需要指令校验的，否则不可通讯。如果要改回不开启指令校验功能需发送如下指令。

发送: AA 7C 00 00 18 B0 CC 33 C3 3C

接收: AA 7C 00 00 18 B0 CC 33 C3 3C

注：下所列出来的指令说明均未添加 CRC16 校验，如开启校验请自行添加 CRC16 校验值到帧尾前。

Table 3-0

参数	宽度（字节）	说明
EN	1	01 为开启
00	1	保留

握手

握手指令(0x00)

发送: 0xaa 0x00 0xcc 0x33 0xc3 0x3c

接收: 0xaa 0x00 'LCDV*.*' Param1 Param2 Param3 Param4 Pic_ID 0xcc 0x33 0xc3 0x3c

Table 3-1

参数	宽度 (字节)	说明
LCDV*.*	可变	*.*是为固件的版本
Param1	1	显示屏的类型 (见 Table 3-58)
Param2	1	当前使用的波特率 ID (见 Table 3-59)
Param3	1	触摸屏的配置模式 1 (见 Table 3-60)
Param4	1	触摸屏的配置模式 2 (见 Table 3-61)
Pic_ID	1	当前屏幕显示图片 ID

显示参数配置

设置前景色和背景色(0x50)

发送: 0xaa 0x50 FColor BColor 0xcc 0x33 0xc3 0x3c

接收: 无

颜色采用 16 位色, RGB565。

Table 3-2

参数	宽度（字节）	说明
FColor	2	前景色，复位默认值 0x0000
BColor	2	背景色，复位默认值 0xffff

设置字符的显示间距(0x51)

发送：0xaa 0x51 XDistance YDistance 0xcc 0x33 0xc3 0x3c

接收：无

Table 3-3

参数	宽度（字节）	说明
XDistance	1	x 方向的字符间距，取值范围 0x00-0xff，复位默认值是 0
YDistance	1	y 方向的字符间距，取值范围 0x00-0xff，复位默认值是 0

取色到背景色调色板(0x52)

发送：0xaa 0x52 X Y 0xcc 0x33 0xc3 0x3c

接收：无

Table 3-4

参数	宽度（字节）	说明
X	2	取色点 X 坐标
Y	2	取色点 Y 坐标

取色到前景色调色板(0x53)

发送：0xaa 0x53 X Y 0xcc 0x33 0xc3 0x3c

接收：无

Table 3-5

参数	宽度（字节）	说明
X	2	取色点 X 坐标
Y	2	取色点 Y 坐标

设置光标的模式(0x54)

发送：0xaa 0x54 Cursor_En X Y C_Width C_Height 0xcc 0x33 0xc3 0x3c

接收：无

Table 3-6

参数	宽度（字节）	说明
Cursor_En	1	0x00：光标关闭，复位默认值是 0x00。 0x01：光标打开。
X	2	光标左上角的横坐标
Y	2	光标左上角的纵坐标
C_Width	1	光标的宽度，设定值范围 0x00-0x1f，显示值为 0x01~0x20，即显示值比设定值多 1，复位默认值是 0x0f
C_Height	1	光标的高度，设定值范围 0x00-0x1f，，显示值为 0x01~0x20，即显示值比设定值多 1，复位默认值是 0x03

文本显示

8x8 点阵 ASCII 字符(0x63)

发送：0xaa 0x63 X Y String 0xcc 0x33 0xc3 0x3c

接收：无

说明：对应 0x1F 小字库。

Table 3-7

参数	宽度（字节）	说明
X	2	显示位置左上角的横坐标
Y	2	显示位置左上角的纵坐标
String	可变	需要显示的字符，字体颜色为前景色。

16x16 点阵 GBK 字符(0x64)

发送：0xaa 0x64 X Y String 0xcc 0x33 0xc3 0x3c

接收：无

说明：对应 0x21 大字库。

Table 3-8

参数	宽度（字节）	说明
X	2	显示位置左上角的横坐标
Y	2	显示位置左上角的纵坐标
String	可变	需要显示的字符，字体颜色为前景色。

32x32 点阵 GB2312 字符(0x65)

发送：0xaa 0x65 X Y String 0xcc 0x33 0xc3 0x3c

接收：无

说明：对应 0x23-0x24 大字库。

Table 3-9

参数	宽度（字节）	说明
X	2	显示位置左上角的横坐标
Y	2	显示位置左上角的纵坐标
String	可变	需要显示的字符，字体颜色为前景色。

12x12 点阵 GBK 字符(0x7e)

发送：0xaa 0x7e X Y String 0xcc 0x33 0xc3 0x3c

接收：无

说明：对应 0x20 大字库。

Table 3-10

参数	宽度（字节）	说明
X	2	显示位置左上角的横坐标
Y	2	显示位置左上角的纵坐标
String	可变	需要显示的字符，字体颜色为前景色。

24x24 点阵 GB2312 字符(0x7f)

发送: 0xaa 0x7f X Y String 0xcc 0x33 0xc3 0x3c

接收: 无

说明: 对应 0x22 大字库。

Table 3-11

参数	宽度 (字节)	说明
X	2	显示位置左上角的横坐标
Y	2	显示位置左上角的纵坐标
String	可变	需要显示的字符, 字体颜色为前景色。

任意字符(0x98)

发送: 0xaa 0x98 X Y Lib_ID Mode Dot Fcolor Bcolor String 0xcc 0x33 0xc3 0x3c

接收: 无

Table 3-12

参数	宽度 (字节)	说明
X	2	显示位置左上角的横坐标
Y	2	显示位置左上角的纵坐标
Lib_ID	1	字库的编号, 取值范围 0x00-0x3b, 对应于 0xf2 指令保存的字库位置
Mode	1	字符的显示方式和编码方式, 详见 Table 3-13
Dot	1	字符点阵大小, 详见 Table 3-14
Fcolor	2	字符区域颜色

Bcolor	2	非字符区域颜色
String	可变	需要显示的字符，不需要以 0 结束。

Table 3-13 字符的显示方式和编码方式

Mode 位	值	说明
7	1	字符区域用 FColor 显示
	0	字符区域不显示（配合.6='1'实现镂空字）
6	1	非字符区域用 BColor 显示
	0	非字符区域不显示（配合.7='1'实现透明字）
5:4		保留
3:0	0	ASCII 字符
	1	GB2312 简体中文字符
	2	GBK 简体中文字符
	3-4	保留
	5	UNICODE 编码
	3-15	保留

说明：点阵中的字符笔划为字符区域，点阵中的其它区域为非字符区域。

Table 3-14 字符点阵大小

Dot 值	字库类型，Mode 的低 4 位		
	0x00 或 0x05	0x01-0x02	
		ASCII 字符	非 ASCII 字符
0x00	8x8	6x12	12x12

0x01	6x12	8x16	16x16
0x02	8x16	12x24	24x24
0x03	12x24	16x32	32x32
0x04	16x32	20x40	40x40
0x05	20x40	24x48	48x48
0x06	24x48	28x56	56x56
0x07	28x56	32x64	64x64
0x08	32x64	--	--
--	--	--	--
0x11	40x80	--	--
0x12	48x96	--	--
0x13	56x112	--	--
0x14	64x128	--	--
--	--	--	--

说明：

1.当用户使用 0x98 指令时，在 0x01~0x02 编码方式下，如果用户选择的字库为任一小字库，系统会自动使用 0x01 复合 ASCII 字库来显示 ASCII 字符，字符大小与 0x01~0x02 编码方式下 Dot 对应的 ASCII 字符点阵大小一致。

2.当用户使用 0x98 指令，想要自己设计一个新的 ASCII 字库来显示 ASCII 字符，此时应选择 8bit 编码方式。比如想使用 64x128 的 ASCII 字符，可使用我司提供的字库生成软件生成一个字体 Size 为 128 的纯 ASCII 字库，下载到 0x02~0x18 的任意一个小字库中（如 0x02 小字库，超出 128K 部分将自动占用接下来的小字库），然后使用 0x98 指令，Lib_ID

为 0x02，Mode 低四位为 0x00，Dot 值为 0x14 来显示。

置点

背景色置多个点(0x60)

发送：0xaa 0x60 X0 Y0 ... Xn Yn 0xcc 0x33 0xc3 0x3c

接收：无

Table 3- 15

参数	宽度（字节）	说明
Xn	2	第 n 个点的显示位置的横坐标
Yn	2	第 n 个点的显示位置的纵坐标

前景色置多个点(0x61)

发送：0xaa 0x61 X0 Y0 ... Xn Yn 0xcc 0x33 0xc3 0x3c

接收：无

Table 3- 16

参数	宽度（字节）	说明
Xn	2	第 n 个点的显示位置的横坐标
Yn	2	第 n 个点的显示位置的纵坐标

动态曲线(0x44)

发送：0xaa 0x44 X YStart YEnd BColor (Y0 FColor0) ... (Yn FColorN) 0xcc 0x33 0xc3 0x3c

接收：无

说明：第一步，用 BColor 绘制直线(X, YStart)-(X, YEnd)，第二步，绘制后面给出的 n 个点，每个点的坐标是(X, Yn)，颜色是 FColorN。

Table 3- 17

参数	宽度（字节）	说明
X	2	垂直线操作的 x 坐标
YStart	2	垂直线操作的 y 轴起始坐标
YEnd	2	垂直线操作的 y 轴终止坐标
BColor	2	在第一步操作里，用于填充垂直线的颜色
Yn	2	第 n 个点的显示位置的坐标的 y 值
FColorN	2	第 n 个点的颜色

直接显存操作(0x42)

发送：0xaa 0x42 Address Color0 ... ColorN 0xcc 0x33 0xc3 0x3c

接收：无

说明：从给定的显存地址开始，写后面给出的 n 个点。如果屏的宽为 XSize,需要写的点的坐标为(X,Y)，那么显存的地址为 XSize * Y + X

Table 3- 18

参数	宽度（字节）	说明
Address	3	显存的地址
ColorN	2	第 n 个点的颜色

线段和多边形

把指定的点用前景色线段连接(0x66)

发送：0xaa 0x66 (X0 Y0) ... (Xn Yn) 0xcc 0x33 0xc3 0x3c

接收：无

Table 3- 19

参数	宽度（字节）	说明
Xn	2	第 n 个点的 x 轴坐标
Yn	2	第 n 个点的 y 轴坐标

把指定的点用背景色线段连接(0x6d)

发送：0xaa 0x6d (X0 Y0) ... (Xn Yn) 0xcc 0x33 0xc3 0x3c

接收：无

Table 3- 20

参数	宽度（字节）	说明
Xn	2	第 n 个点的 x 轴坐标
Yn	2	第 n 个点的 y 轴坐标

显示多条同底垂直线(0x45)

发送：0xaa 0x45 X Y HeightMax Height0 ... HeightN 0xcc 0x33 0xc3 0x3c

接收：无

Table 3- 21

参数	宽度（字节）	说明
X	2	垂直线起点的 X 轴坐标，每绘制一根垂直线，X 轴坐标加 1
Y	2	垂直线起点的 Y 轴坐标，每根垂直线起点的 Y 轴坐标相同
HeightMax	1 或者 3	垂直线的最大高度 如第一个字节为 0x00，最大高度为 3 个字节（仅后两个字节有效） 如第一个字节为非 0x00，最大高度为 1 字节
HeightN	1 或者 2	如最大高度为 1 字节，HeightN 为 1 字节 如最大高度为 3 字节，则 HeightN 为 2 字节（每两个字节作为一条垂直线的高度）。 HeightN 的取值为 0 ~ (HeightMax-1)，当 HeightN 等于 HeightMax 时则为清除此位置的垂直线

显示折线(0x46)

发送：0xaa 0x46 X XIncrement Y0 ... Yn 0xcc 0x33 0xc3 0x3c

接收：无

说明：绘制直线(X, Y(n-1)) ~ (X + XIncrement, Yn)，每完成一次以上操作，X 的值加上 XIncrement。

Table 3- 22

参数	宽度（字节）	说明
X	2	点的 x 轴坐标
XIncrement	1	每完成一次直线绘制操作后， X 坐标的增量
Yn	2	点的 y 轴坐标

圆弧和圆域

反色显示或填充圆弧或圆域(0x67)

发送：0xaa 0x67 (Type0 X0 Y0 r0) ... (TypeN Xn Yn rn) 0xcc 0x33 0xc3 0x3c

接收：无

Table 3- 23

参数	宽度（字节）	说明
TypeN	1	决定当前操作的类型，详见表 Table 3- 24
Xn	2	第 n 个圆的圆心的 x 轴坐标
Yn	2	第 n 个圆的圆心的 y 轴坐标
rn	1	第 n 个圆的半径

Table 3- 24 圆操作类型

TypeN	说明
0x00	把指定的圆弧反色（全圆弧）
0x01	前景色显示指定的圆弧（全圆弧）
0x02	把指定的圆域反色（全圆域）
0x03	用前景色填充圆域（全圆域）

反色显示或填充圆弧或圆域(0x67)

发送：0xaa 0x67 0x04 X Y r As Ae 0xcc 0x33 0xc3 0xc3c

接收：无

Table 3- 25

参数	宽度（字节）	说明
X	2	圆弧圆心的 x 轴坐标
Y	2	圆弧圆心的 y 轴坐标
r	2	圆弧的半径（取值范围 0x0000 ~ 0x00B5）
As	2	圆弧起始角度（0 ~ 720，单位 0.5 度）
Ae	2	圆弧结束角度（0 ~ 720，单位 0.5 度）

注：此指令仅为兼容指令，不太完善，半径超出 0x00B5 将直接返回，角度虽然表

示单位为 0.5 度，但最小画弧角度为 1 度，0.5 度将直接忽略。

矩形框

前景色显示矩形框(0x69)

发送: 0xaa 0x69 (XStart0 YStart0 XEnd0 YEnd0) ... (XStartN YStartN XEndN YEndN)

0xcc 0x33 0xc3 0x3c

接收: 无

Table 3- 26

参数	宽度（字节）	说明
XStartN	2	第 N 个矩形框左上角的 x 轴坐标
YStartN	2	第 N 个矩形框左上角的 y 轴坐标
XEndN	2	第 N 个矩形框右下角的 x 轴坐标
YEndN	2	第 N 个矩形框右下角的 y 轴坐标

背景色显示矩形框(0x79)

发送: 0xaa 0x79 (XStart0 YStart0 XEnd0 YEnd0) ...

(XStartN YStartN XEndN YEndN) 0xcc 0x33 0xc3 0x3c

接收: 无

Table 3-27

参数	宽度（字节）	说明
XStartN	2	第 N 个矩形框左上角的 x 轴坐标
YStartN	2	第 N 个矩形框左上角的 y 轴坐标
XEndN	2	第 N 个矩形框右下角的 x 轴坐标
YEndN	2	第 N 个矩形框右下角的 y 轴坐标

区域操作

区域填充(0x74)

发送: 0xaa 0x74 X Y Color 0xcc 0x33 0xc3 0x3c

接收: 无

说明: 仅填充凸多边形。填充区域应为纯色, 非纯色的填充区域将造成填充不完全达不到预期效果(小提示: .bmp 格式图像一般为纯色, 但.jpg 格式图像通常为杂色)。

Table 3- 28

参数	宽度(字节)	说明
X	2	区域填充种子点 X 轴坐标
Y	2	区域填充种子点 Y 轴坐标
Color	2	要填充的颜色

清屏(0x62)

发送: 0xaa 0x62 0xcc 0x33 0xc3 0x3c

接收: 无

说明: 用背景色来填充全屏。

背景色填充矩形区域(0x6a)

发送: 0xaa 0x6a (XStart0 YStart0 XEnd0 YEnd0) ... (XStartN YStartN XEndN YEndN)

0xcc 0x33 0xc3 0x3c

接收: 无

Table 3-29

参数	宽度（字节）	说明
XStartN	2	第 N 个矩形区域左上角的 X 轴坐标
YStartN	2	第 N 个矩形区域左上角的 Y 轴坐标
XEndN	2	第 N 个矩形区域右下角的 X 轴坐标
YEndN	2	第 N 个矩形区域右下角的 Y 轴坐标

前景色填充矩形区域(0x6b)

发送：0xaa 0x6b (XStart0 YStart0 XEnd0 YEnd0) ... (XStartN YStartN XEndN YEndN)

0xcc 0x33 0xc3 0x3c

接收：无

Table 3-30

参数	宽度（字节）	说明
XStartN	2	第 N 个矩形区域左上角的 X 轴坐标
YStartN	2	第 N 个矩形区域左上角的 Y 轴坐标
XEndN	2	第 N 个矩形区域右下角的 X 轴坐标
YEndN	2	第 N 个矩形区域右下角的 Y 轴坐标

反色矩形区域(0x6c)

发送：0xaa 0x6c (XStart0 YStart0 XEnd0 YEnd0) ... (XStartN YStartN XEndN YEndN)

0xcc 0x33 0xc3 0x3c

接收：无

Table 3- 31

参数	宽度（字节）	说明
XStartN	2	第 N 个矩形区域左上角的 X 轴坐标
YStartN	2	第 N 个矩形区域左上角的 Y 轴坐标
XEndN	2	第 N 个矩形区域右下角的 X 轴坐标
YEndN	2	第 N 个矩形区域右下角的 Y 轴坐标

矩形区域左环移(0x70)

发送：0xaa 0x70 (XStart0 YStart0 XEnd0 YEnd0 N0) ... (XStartN YStartN XEndN YendN NN)

0xcc 0x33 0xc3 0x3c

接收：无

Table 3- 32

参数	宽度（字节）	说明
XStartN	2	第 N 个矩形区域左上角的 X 轴坐标
YStartN	2	第 N 个矩形区域左上角的 Y 轴坐标
XEndN	2	第 N 个矩形区域右下角的 X 轴坐标
YEndN	2	第 N 个矩形区域右下角的 Y 轴坐标
NN	1	第 N 个矩形区域要移动的横向点数，取值 0x01-0x0f

矩形区域右环移(0x71)

发送：0xaa 0x71(XStart0 YStart0 XEnd0 YEnd0 N0) ... (XStartN YStartN XEndN YendN NN)

0xcc 0x33 0xc3 0x3c

接收：无

Table 3- 33

参数	宽度（字节）	说明
XStartN	2	第 N 个矩形区域左上角的 X 轴坐标
YStartN	2	第 N 个矩形区域左上角的 Y 轴坐标
XEndN	2	第 N 个矩形区域右下角的 X 轴坐标
YEndN	2	第 N 个矩形区域右下角的 Y 轴坐标
NN	1	第 N 个矩形区域要移动的横向点数，取值 0x01-0x0f

矩形区域左移(0x72)

发送：0xaa 0x72 (XStart0 YStart0 XEnd0 YEnd0 N0) ... (XStartN YStartN XEndN YendN NN)

0xcc 0x33 0xc3 0x3c

接收：无

Table 3- 34

参数	宽度（字节）	说明
XStartN	2	第 N 个矩形区域左上角的 X 轴坐标
YStartN	2	第 N 个矩形区域左上角的 Y 轴坐标
XEndN	2	第 N 个矩形区域右下角的 X 轴坐标

YEndN	2	第 N 个矩形区域右下角的 Y 轴坐标
NN	1	第 N 个矩形区域要移动的横向点数，取值 0x01-0x0f

矩形区域右移(0x73)

发送：0xaa 0x73 (XStart0 YStart0 XEnd0 YEnd0 N0) ... (XStartN YStartN XEndN YendN NN)

0xcc 0x33 0xc3 0x3c

接收：无

Table 3- 35

参数	宽度（字节）	说明
XStartN	2	第 N 个矩形区域左上角的 X 轴坐标
YStartN	2	第 N 个矩形区域左上角的 Y 轴坐标
XEndN	2	第 N 个矩形区域右下角的 X 轴坐标
YEndN	2	第 N 个矩形区域右下角的 Y 轴坐标
NN	1	第 N 个矩形区域要移动的横向点数，取值 0x01-0x0f

图和图标显示

显示闪存里的一幅图片(0x40)

发送：0xaa 0x40 Pic_ID 0xcc 0x33 0xc3 0x3c

接收：无

Table 3- 36

参数	宽度（字节）	说明
Pic_ID	1 或 2	保存在闪存的图片的编号(对应于 0xe2 指令)

从闪存里的一幅图片中剪切一部分显示(0x41)

发送：0xaa 0x41 Pic_ID XStart YStart XEnd YEnd X Y 0xcc 0x33 0xc3 0x3c

接收：无

Table 3- 37

参数	宽度（字节）	说明
Pic_ID	1 或 2	保存在闪存的图片的编号(对应于 0xe2 指令)
XStart	2	闪存图片中矩形区域左上角的 x 轴坐标
YStart	2	闪存图片中矩形区域左上角的 y 轴坐标
XEnd	2	闪存图片中矩形区域右下角的 x 轴坐标（XEnd >= XStart）
YEnd	2	闪存图片中矩形区域右下角的 y 轴坐标（YEnd >= YStart）
X	2	目标区域左上角的 x 轴坐标
Y	2	目标区域左上角的 y 轴坐标

保存当前画面到闪存(0xe2)

发送：0xaa 0xe2 Pic_ID 0xcc 0x33 0xc3 0x3c

接收：0xaa 0xe2 0x4f 0x4b 0xcc 0x33 0xc3 0x3c

Table 3- 38

参数	宽度（字节）	说明
Pic_ID	1 或 2	图片的编号（每幅图片占用的存储空间如下式计算：(分辨率 Ysize+63)%64*128KB，当前标准图片空间为 64MB，最大可扩展占用用户顺序 Flash 区 26MB，当图片占用用户区后，用户区相应位置应不再使用。

显示多个闪存里的图标(0x99)

发送：0xaa 0x99 (X0 Y0 IconID0)...(Xn Yn IconIDn) 0xcc 0x33 0xc3 0x3c

接收：无

Table 3- 39

参数	宽度（字节）	说明
Xn	2	目标区域左上角的 x 轴坐标
Yn	2	目标区域左上角的 y 轴坐标
IconIDn	2	图标编号(对应于 0x1d 配置文件)

注：如图片较小可使用本公司提供的图片拼合工具拼合成当前使用的分辨率的大图片存储，然后用 0x41 指令或 0x99 指令进行小图片显示，达到存储、显示多张小图片的目的。

自动执行指令

关闭或打开自动执行指令(0x9a)

关闭自动执行指令功能

发送：0xaa 0x9a 0xff 0xcc 0x33 0xc3 0x3c

接收：无

打开自动执行指令功能

发送：0xaa 0x9a PackID 0xcc 0x33 0xc3 0x3c

接收：无

Table 3-40

参数	宽度（字节）	说明
PackID	1	指令组编号，取值 0x00~0x0f 共 16 组，详见 0x1c 配置文件介绍

自动执行衍生指令

a. 开机 Demo 自动运行

发送：0xaa 0x9a DemoID 0xcc 0x33 0xc3 0x3c

接收：无

注：DemoID=0xe0+PackID，取值 0xe0~0xef。

b. 开机 Demo 自动运行关闭

发送：0xaa 0x9a 0xf0 0xcc 0x33 0xc3 0x3c

接收：无

闪存操作

修改字库(0xf2)

发送：0xaa 0xf2 0xf2 0x2f 0x5a 0xa5 Lib_ID 0xcc 0x33 0xc3 0x3c

接收：Please Tx Text_Lib ! 且当开启 CRC16 校验码时不返回校验码。

说明：发送完命令后，需要等待接收 “Please Tx Text_Lib !” 字符串，接收到以后，开始

发送字库数据，字库数据发送完毕后，需要等待接收 “One Text_Lib Saved OK !”

字符串，接收到该字符串后，字库已经保存到闪存中。

Table 3- 41

参数	宽度（字节）	说明
Lib_ID	1	字库编号。其中 0 到 0x1f 一共 32 个小字库，每个大小为 128KB，一共 4MB，0x20 到 0x3b 一共 28 个大字库，每个大小为 1MB，一共 28MB。若用户字库大小超过 1MB，可以通过使用多个连续的 1MB 字库组成一个大字库，下载下一个字库时应跳过上一个字库已经占用的编号（比如字库 0x23 为 2M，占用了 0x23 和 0x24 两个编号，这样在下载下一个字库时，则应以 0x25 编号开始而不能再使用 0x24，否则 0x23 字库的部分字会被覆盖引发错误显示）。系统保留了部分编号，详见表 Table 3- 42。

Table 3- 42 保留的字库编号

编号	说明
0x00	用于储存配置信息，已占用。
0x01	默认的 8x8 ASCII 字库位置，对应 0x63 指令。此字库为复合字库，同时包含其它几种分辨率的 ASCII 字库，用于兼容用户，一般不要修改，如需修改请联系厂家。
0x19	用于储存配置信息，已占用。
0x1A	用于储存键控、触控自动执行指令配置信息，可更新。
0x1B	用于储存键盘切换配置信息，可更新。
0x1C	用于储存自动循环执行指令配置信息，可更新。
0x1D	用于储存图标配置信息，可更新。
0x1E	用于储存触摸屏切换配置信息，可更新。
0x20	默认的 12x12 GBK 字库位置，对应 0x7e 指令。

0x21 默认的 16x16 GBK 字库位置，对应 0x64 指令。

0x22 默认的 24x24 GB2312 字库位置，对应 0x7f 指令。

0x23 默认的 32x32 GB2312 字库位置（1MB 内部分），对应 0x65 指令。

0x24 默认的 32x32 GB2312 字库位置（大于 1MB 部分），对应 0x65 指令。

写数据到闪存里的用户数据区(0x90)

发送：0xaa 0x90 0x55 0xaa 0x5a 0xa5 Address Data 0xcc 0x33 0xc3 0x3c

接收：0xaa 0x90 0x4f 0x4b 0xcc 0x33 0xc3 0x3c

说明：当正确接收到反馈指令后，表示数据已经成功保存到闪存中。用户数据区一共包含 239 个 Block，每个 Block 大小为 128KB。当写用户数据区的时候，如果写地址加上数据长度超过 128KB 的 Block 边界，比如写地址为 0x13ffff，长度为 2 字节，那么下一 Block 会被擦除。如果写地址正好是 Block 边界，比如 0x20000，那么当前 Block 会被擦除。

Table 3-43

参数	宽度（字节）	说明
Address	4	用户顺序存储数据区的地址，取值范围是 0x00000000-0x01ddffff，一共 30592KB，用户随机存储数据区的地址，取值范围是 0x01de0000~0x01deffff 共 64KB
Data	可变	需要写到用户数据区的数据

从闪存里的用户数据区读数据(0x91)

发送: 0xaa 0x91 Address Length 0xcc 0x33 0xc3 0x3c

接收: 0xaa 0x91 Address Length Data 0xcc 0x33 0xc3 0x3c

说明: 如果地址加上长度大于用户数据区的大小(30592KB 或 64KB), 那么返回的

长度会小于指定的长度,且当开启 CRC16 校验码时不返回校验码。

Table 3-44

参数	宽度 (字节)	说明
Address	4	用户顺序存储数据区的地址, 取值范围是 0x00000000-0x01ddffff, 一共 30592KB, 用户随机存储数据区的地址, 取之范围是 0x01de0000~0x01deffff 共 64KB
Length	2	需要从用户数据区读的数据的长度, 一次最多是 64KB 个数据。
Data	可变	从用户数据区读出来的数据

当前终端 128MB 存储区域空间划分如下表:

Table 3-45 128MB 闪存空间划分说明

地址范围	大小	说明
0x00000000-0x01ffffff	32MB	字库存储区
0x02000000-0x05ffffff	64MB	图片存储区
0x06000000-0x07dfffff	30592KB	顺序存储区 (前 26M 也可以用来存储图片)
0x07de0000-0x07deffff	64KB	随机存储区
0x07df0000-0x07ffffff	2112KB	保留

用户可使用 30656KB 的闪存区域作为用户数据存储区。该区域分为两部分，顺序存储区域和随机存储区域，地址范围分别为 0x000000-0x01ddffff 和 0x01de0000-0x01deffff（此地址为去掉基地址后的偏移地址，使用时只取偏移地址即可）。闪存每 Block 的大小为 128KB。访问顺序存储区时，如果遇到 Block 边界，那么当前 Block 为被擦除，如从 0x1fff0 地址写 100 个字节的数据，由于该操作涵盖了 2 个 Block，首地址为 0 和首地址为 0x20000 的 Block，因此终端在执行此指令的时候，会从 0x1fff0 地址开始写 16 个数据，然后擦除首地址为 0x20000 的 Block，接着从 0x20000 地址开始写剩下的 84 个字节的数据（提示：我们不能对顺序区非 Block 开头的区域进行反复的写操作，因为非 Block 开头的顺序区在写入时并不擦除，而闪存同一位置只能在擦除后才能写入，这样重复写入就会造成写入错误，建议使用顺序区时直接按次序向后写入，这样既可以均匀的使用闪存各 Block 增加闪存使用寿命，又可以保障写入数据的正确性）。访问随机存储区域时，用户可以随意地改写该区域的内容，而不会影响随机区其他未改变区域的内容。

键盘操作

键码上传(0x71)

发送：无

接收：0xaa 0x71 KeyCode 0xcc 0x33 0xc3 0x3c

Table 3-46

参数	宽度（字节）	说明
KeyCode	1	用户预先设置好的键码。用户每按一次按键，上传一次键码。如果按键一直被按下，那么会每隔 200ms 上传一次键码。

配置键码(0xe5)

发送: 0xaa 0xe5 0x55 0xaa 0x5a 0xa5 KeyCode 0xcc 0x33 0xc3 0x3c

接收: 无

Table 3- 47

参数	宽度（字节）	说明
KeyCode	64	需要设置的键码值。固定为 64 字节。对于 4x4 的键盘，只有 16 个键码有效。

触摸屏操作

松开触摸屏时最后一次上传坐标(0x72)

发送: 无

接收: 0xaa 0x72 TouchX TouchY 0xcc 0x33 0xc3 0x3c

Table 3- 48

参数	宽度（字节）	说明
TouchX	2	触屏非自动切换模式下，当触摸屏由按下状态变成抬起状态时，触摸点的 X 轴坐标
TouchY	2	触屏非自动切换模式下，当触摸屏由按下状态变成抬起状态时，触摸点的 Y 轴坐标

按下触摸屏时上传坐标(0x73)

发送: 无

接收: 0xaa 0x73 TouchX TouchY 0xcc 0x33 0xc3 0x3c

Table 3- 49

参数	宽度（字节）	说明
TouchX	2	触屏非自动切换模式下，当按下及一直按着触摸屏时，触摸点的 X 轴坐标
TouchY	2	触屏非自动切换模式下，当按下及一直按着触摸屏时，触摸点的 Y 轴坐标

说明：当触屏被一直按下时，只上传第一次按下时的坐标，即使在一直按下过程中触摸点发生了移动，上传的坐标也不会改变。

触摸屏校准(0xe4)

发送：0xaa 0xe4 0x55 0xaa 0x5a 0xa5 0xcc 0x33 0xc3 0x3c

接收：0xaa 0xe4 0x4f 0x4b 0xcc 0x33 0xc3 0x3c

说明：发送完校准指令后，屏幕会清屏并在屏幕的左上、右上及中下位置显示三个十字校准点，为提高校准精度，请连续点击每一个十字校准点中心（一个十字连续点击间隔不超过 1s 将累加并取平均值（连续点击十来次即可），间隔 1s 后显示下一个十字校准点），完成十字校准点点击后，会出现绿框进行校准确认，点击几下绿框内的位置，如果校准正将退出，否则将重新出现十字校准点进行下一遍校准，一旦进入校准程序，不完成校准终端将不会退出此程序，因此不支持触摸屏的 LCD 请不要发此指令，否则将只能采用重上电的方法退出。对于从未校准成功过的触屏，第一次点击触屏将直接进入校准程序。

在触屏自动切换模式下自动上传预设键码(0x78)

发送：无

接收：0xaa 0x78 TouchCode 0xcc 0x33 0xc3 0x3c

Table 3- 50

参数	宽度（字节）	说明
TouchCode	2	触屏自动切换模式下，在触控界面自动切换模式下自动上传预设键码，详见 0x1E 配置文件，相当于非自动切换模式下 0x72 指令

在触控界面自动切换模式下自动上传预设键码(0x79)

发送：无

接收：0xaa 0x79 TouchCode 0xcc 0x33 0xc3 0x3c

Table 3- 51

参数	宽度（字节）	说明
TouchCode	2	触屏自动切换模式下，在触控界面自动切换模式下自动上传预设键码，详见 0x1E 配置文件，相当于非自动切换模式下 0x73 指令，需要在 0xe0 参数配置中配置 Para1.0 为 1 时上传。

时间操作

开启/关闭时间显示或读取时间(0x9b)

关闭时间显示

发送：0xaa 0x9b 0x00 0xcc 0x33 0xc3 0x3c

接收：无

开启时间显示

发送：0xaa 0x9b 0xff Mode Font Color X Y 0xcc 0x33 0xc3 0x3c

接收：无

Table 3- 52

参数	宽度（字节）	说明
Mode	1	时间的显示样式, 详见表 Table 3- 53
Font	1	时间字体的大小, 详见表 Table 3- 54
Color	2	时间字体的颜色(RGB565)
X	2	时间在屏幕上显示位置左上角的 X 坐标
Y	2	时间在屏幕上显示位置左上角的 Y 坐标

Table 3- 53 时间的显示样式

Mode	说明
0x00	时间显示样式：HH:MM:SS
0x01	时间显示样式：20YY-MM-DD HH:MM:SS
0x02	时间显示样式：20YY-MM-DD WEEK HH:MM:SS
0x03	时间显示样式：20YY-MM-DD 星期 HH:MM:SS

Table 3- 54 时间的显示样式

Font	说明
0x00	时间字体为 8x8
0x01	时间字体为 6x12
0x02	时间字体为 8x16

0x03 时间字体为 12x24

0x04 时间字体为 16x32

读取公历时间

发送: 0xaa 0x9b 0x5a 0xcc 0x33 0xc3 0x3c

接收: 0xaa 0x9b 0x5a YY MM DD WW HH MM SS 0xcc 0x33 0xc3 0x3c

说明: 时间的表示采用 BCD 方式。

设置时间(0xe7)

发送: 0xaa 0xe7 0x55 0xaa 0x5a 0xa5 YY MM DD HH MM SS 0xcc 0x33 0xc3 0x3c

接收: 无

说明: 时间的表示采用 BCD 方式。

例子: 把时间设置成 2010-5-12 11:06:35

0xaa 0xe7 0x55 0xaa 0x5a 0xa5 0x10 0x05 0x12 0x11 0x06 0x35 0xcc 0x33 0xc3 0x3c

闹钟设定 (0xe8)

发送: 0xaa 0xe8 Num Week Hour Minute Mode 0xcc 0x33 0xc3 0x3c

接收: 0xaa 0xe8 Num Week Hour Minute Mode En 0xcc 0x33 0xc3 0x3c

Table 3-55

参数	宽度 (字节)	说明
Num	1	0x00~0x07 : 总共 8 个闹钟任选 1
Week	1	0x00~0x06 : 周日~周六其中 1 天有效

		0xff : 每天有效
Hour	1	0x00~0x23 : BCD 码，要设定的小时值
Minute	1	0x00~0x59 : BCD 码，要设定的分钟值
Mode	1	0x00 : 串口回传数据（数据格式尚未定义，暂返回和设定返回一样的数据） 0x01 : Buzzer 鸣响 30 次（周期 100ms）

说明：错误的设定将没有任何返回。

闹钟使能及状态查询：

发送：0xaa 0xe8 Num En 0xcc 0x33 0xc3 0x3c

接收：0xaa 0xe8 Num Week Hour Minute Mode En 0xcc 0x33 0xc3 0x3c

Table 3-56

参数	宽度（字节）	说明
Num	1	0x00~0x07 : 总共 8 个闹钟任选 1
En	1	0x00 : 关闭 Num 所指闹钟 0x01 : 打开 Num 所指闹钟 0x02~0xff : 查询 Num 所指闹钟状态

说明：错误的参数将没有任何返回。

注：如需串口返回 RTC 数据请参考下节“参数配置(0xe0)衍生指令”。

参数配置

参数配置(0xe0)

发送：0xaa 0xe0 0x55 0xaa 0x5a 0xa5 LCD_ID BaudrateID Para1 (Para2) 0xcc 0x33 0xc3 0x3c

接收：0xaa 0xe0 LCD_ID BaudrateID Para1 (Para2) 0xcc 0x33 0xc3 0x3c

说明：本配置保存在闪存中，掉电后配置依然存在，如 Para2 已被配置过则返回。

Table 3- 57

参数	宽度（字节）	说明
LCD_ID	1	显示屏的类型，详见 Table 3- 58。
BaudrateID	1	波特率编号，详见 Table 3- 59（注意：波特率配置后将存储，故配置前请检查您的软硬件是否支持新的波特率，如果不支持，将造成后继配置失败而无法控制）。
Para1	1	键盘和触屏控制方式设定 1，详见 Table 3- 60。
(Para2)	1	键盘和触屏控制方式设定 2，详见 Table 3- 61。

Table 3- 58 显示屏类型

LCD_ID	分辨率	示例 LCD 屏型号
0x00	(100*200)	--
0x01	QVGA (320*240)	--
0x02	(400*400)	--
0x03	WQVGA (480*272)	--
0x04	(512*300)	--
0x05	(512*480)	--

0x06	VGA (640*480)	--
0x07	(640*512)	--
0x08	WVGA (800*480)	--
0x09	SVGA (800*600)	--
0x0A	XGA (1024*768)	--

说明：对应相同分辨率不同型号的 LCD 屏，模块硬件可能需要相应修改程序。

Table 3- 59 波特率编号

BaudrateID	说明
0x00	1200 bps (不支持此速率)
0x01	2400 bps (不支持此速率)
0x02	4800 bps
0x03	9600 bps
0x04	19200 bps
0x05	38400 bps
0x06	57600 bps
0x07	115200 bps
0x08	28800 bps (注 1)
0x09	76800 bps (注 1)
0x0A	62500 bps (注 1)
0x0B	125000 bps (注 1, 注 2)
0x0C	250000 bps (注 1, 注 2)

0x0D	230400 bps（注 1，注 2）
0x0E	460800 bps（注 1，注 2，注 3）
0x0F	921600 bps（注 1，注 2，注 3）

注 1：特殊串口速率可能不为 PC 或虚拟串口所支持，谨慎使用。

注 2：115200bps 以上速率可能不为低速 RS232 芯片所支持，谨慎使用

注 3：高串口速率下载图片时会引发串口忙信号（同时蜂鸣器会有反应），此速率下下载图片尽量关闭 RTC 显示以获得更高的串口带宽或者按忙信号控制下载速度。

Table 3- 60 键盘和触屏控制方式 1

Para1	位说明
.7	0= 触屏非自动切换模式下，点击触摸屏后，松开触摸屏时，自动上传 0x72 指令；触屏自动切换模式下必须为 0。 1= 触屏非自动切换模式下，点击触摸屏后，松开触摸屏时，不上传 0x72 指令。
.6	0= 触屏非自动切换模式下，按下触摸屏后，以 100mS 的间隔自动上传 0x73 指令；触屏自动切换模式下，按下触摸屏后，以 100mS 的间隔自动上传 0x79 指令（Para1.0 需同时使能）； 1= 触屏非自动切换模式下，按下触摸屏后，只会在按下时自动上传 1 次 0x73 指令；触屏自动切换模式下，按下触摸屏后，只会在按下时自动上传 1 次 0x79 指令（Para1.0 需同时使能）。
.5	0= 设定为触屏非自动切换模式，点击触摸屏后，不进行触控界面的切换； 1= 设定为触屏自动切换模式，点击触摸屏或键盘后，按照 0x1E 配置文件的要求进行上传 0x1A 中记录指令或执行 0x1A 中记录指令(上传还是执行通过 Para2.6 配置)。
.4	0= 背光不受触摸屏或键盘控制； 1= 背光由触摸屏或键盘控制，同时用户也可以通过 0x6E/0x6F 指令强制开关。
.3	0= 触摸屏或按键有蜂鸣器伴音； 1= 触摸屏或按键无蜂鸣器伴音。
.2	0= 预留指令模式；

	1= 当前指令模式（按本指令集所述指令运行）。
.1	0= 触屏自动切换模式下，蜂鸣器伴音一直开启； 1= 触屏自动切换模式下，蜂鸣器伴音只在触摸有效区域时鸣响 1 次。
.0	0= 触屏自动切换模式下，不上传 0x79 指令； 1= 触屏自动切换模式下，上传 0x79 指令。

Table 3-61 键盘和触屏控制方式 2

Para2	位说明
.7	保留写 0（不关注）。
.6	0= 关闭触屏指令自动执行； 1= 打开触屏指令自动执行（在触屏自动切换模式下，按下触屏按钮或触屏按钮关联的键盘按键后，如果此按钮的 TouchCode（见 0x1B 及 0x1E 配置文件说明）索引到了 0x1A 中记录的指令，则终端将自动执行此指令而不再上传）。
.0~5	保留写 0（不关注）。

参数配置(0xe0)衍生指令：

发送：0xaa 0xe0 KTMMode 0xcc 0x33 0xc3 0x3c

接收：无

说明：此指令为衍生指令，供一些有特殊需求的用户使用，用来关闭或打开键盘和触摸屏，防止一些误操作如 Demo 演示时的不当触摸等。

Table 3- 62

KTMode	位说明
.7~3	保留写 0（不关注）。
.2	0= 时钟每分钟回传关闭； 1= 时钟每分钟回传打开（回传的 RTC 数据和“读取公历时间”指令返回格式相同）。
.1	0= 触摸屏打开； 1= 触摸屏关闭。
.0	0= 键盘打开； 1= 键盘关闭。

蜂鸣器控制

蜂鸣器控制(0x49)

发送：0xaa 0x49 OnTime 0xcc 0x33 0xc3 0x3c

接收：无

说明：控制蜂鸣器鸣响一声，时间为 OnTime*10 ms。

背光控制

背光控制说明

模组硬件上同时提供 LCD_POWER_EN（PWM2）和 LCD_PWM（PWM1）两个控制信号，LCD_POWER_EN 输出高低电平，LCD_PWM 输出可调 PWM 波形。

背光使能 POWER_EN 控制(0x6e)*POWER_EN 恒输出低电平*

发送：0xaa 0x6e 0x00 0xcc 0x33 0xc3 0x3c

接收：无

POWER_EN 恒输出高电平

发送：0xaa 0x6e 0x01 0xcc 0x33 0xc3 0x3c

接收：无

触控背光亮度 PWM 控制(0x6e)*背光 PWM 恒输出低电平*

发送：0xaa 0x6e 0xcc 0x33 0xc3 0x3c

接收：无

触控背光亮度 PWM 设定

发送：0xaa 0x6e 0x55 0xaa 0x5a 0xa5 V_ON V_OFF ON_TIME 0xcc 0x33 0xc3 0x3c

接收：无

Table 3-63

参数	宽度（字节）	说明
V_ON	1	当背光的开关受触摸或键盘控制时，点击触摸屏（或键盘）后背光自动点亮亮度，取值为 0x00-0x3f。
V_OFF	1	当背光的开关受触摸或键盘控制时，在一段时间（ON_TIME）没有点击触摸屏（或键盘）后背光自动关闭时维持的亮度，取值为 0x00-0x3f。
ON_TIME	1	当背光的开关受触摸或键盘控制时，点击触摸屏（或键盘）后背光自动点亮的时间，单位为 0.5 秒，0x00-0xff（最大 127.5 秒）。背光熄灭时，第一次点击将只会点亮背光而不会处理（注：此值应根据实际情况适当设置，不能设置太小，设置太小会唤醒后马上变黑，加上第一次点击不会做处理，造成误认为屏未工作的“黑屏”现象）。背光亮度触控（键控）功能，需通过 0xe0 指令来启用。

背光亮度 PWM 调节(0x6f)

背光亮度调节指令分为断电保存与不保存的两种选择，不保存配置是在保存配置基础上的参数末尾添加了 0xff 作为区分。

背光 PWM 恒输出高电平

发送：0xaa 0x6f 0xcc 0x33 0xc3 0x3c (保存配置)

接收：无

发送：0xaa 0x6f 0xff 0xcc 0x33 0xc3 0x3c (不保存配置)

接收：无

背光亮度 PWM 调节

发送：0xaa 0x6f PWM_Duty 0xcc 0x33 0xc3 0x3c (保存配置)

接收：无

发送：0xaa 0x6f PWM_Duty 0xff 0xcc 0x33 0xc3 0x3c (不保存配置)

接收：无

Table 3- 64

参数	宽度（字节）	说明
PWM_Duty	1	背光亮度 PWM 控制设定值，取值 0x00-0x3f，设置 0x00 时 PWM 引脚将恒输出低电平，设置 0x3f 时 PWM 引脚将恒输出高电平。对于使用逆变器的 LCD 屏，其背光亮度不能使用 PWM 调整。

背光 PWM 频率设定

发送：0xaa 0x6f PWM_DivH PWM_DivL 0xcc 0x33 0xc3 0x3c (保存配置)

接收：无

发送：0xaa 0x6f PWM_DivH PWM_DivL 0xff 0xcc 0x33 0xc3 0x3c (不保存配置)

接收：无

Table 3-65

参数	宽度（字节）	说明
PWM_DivH	1	背光 PWM 频率系数高。
PWM_DivL	1	背光 PWM 频率系数低。 说明：不同的屏需要的 PWM 输出频率，此输出频率假定为 PWM_Rate，则 PWM 频率系数的计算采用如下公式： $PWM_Div = 515625 / PWM_Rate$

二维码生成与显示

3.18 二维码生成与显示(0x68)

发送：0xaa 0x68 X Y Scale String 0xcc 0x33 0xc3 0x3c

接收：无

Table 3-66

参数	宽度（字节）	说明
X	2	显示位置左上角的横坐标
Y	2	显示位置左上角的纵坐标
Scale	1	二维码生成后显示的放大倍数，1 为以单个像素点生成后的二维码大小，1~5
String	不定	要生成二维码的 ASCII 码字符，目前最大支持 85 个字节

配置文件介绍

系统配置文件(0x00，0x19)

系统的工作参数保存在此文件，禁止用户修改此文件。

触控自动执行指令配置文件(0x1A)

触控自动执行指令配置文件用来进行按下触控按键后指令的自动执行，是由最多 1024 条指令组成的二进制文件，每条指令最多 127 个字节，固定占 128 字节存储器空间，单条指令定义如 Table 4- 1：

Table 4- 1 触控自动执行指令定义

首地址	宽度 (字节)	定义	说明
0x00	1	Length_Command	本条指令长度。
		指令	如果 Length_Command=0x00，表示本条指令是一条组合指令（批处理指令）。
0x01	不定	指令指针	如果 Length_Command 不是 0x00，0x01 开始是要发送的指令。
		(Cmd_EN,Cmd_ID,Tx_Delay)	如果 Length_Command=0x00，后面是组合指令的指针，每个指针由 4 个字节组成，最多 31 个指令指针（组合指令）。 指针定义如下： Cmd_EN：0x00 表示指令组发送结束，其它表示指令发送； Cmd_ID：要发送的指令 ID，0x000-0x3FF，对应 0x1A 文件的非组合指令编号，不支持组合指令的嵌套； Tx_Delay：每条指令的发送间隔，单位为 0.1 秒。 发送组合指令期间，TG01 不响应任何用户指令和外设操作。

按键自动切换配置文件(0x1B)

按键配置文件用来进行按键触发功能的配置，是由最多 5957 条键控指令组成的二进制文件，

每条键控指令 22 个字节，定义如 Table 4-2：

Table 4-2 键控界面配置指令定义

首地址	宽度 (字节)	定义	说明
0x00	2	Pic_Now	当前显示屏幕的图片编号； 如果 Pic_Now 的高字节为 0xFF 表示键控指令结束。
0x02	2	0x00:Key_Code	按键键码
0x04	2	Pic_Next	按键切换到下一个界面的图片编号； 如果 Pic_Next 的高字节为 0xFF 表示不进行界面切换。 如果 Pic_Next 的高字节为 0xFE，表示进行区域图片的切换（此时 Pic_Next 仅低 8 位表示要跳转到的图片编号）。
0x06	14	Pic_Cut, Xs,Ys, Xe,Ye, X,Y	Pic_next 高字节=0xFE 区域图片切换区域定义 Pic_next 高字节=其它 无定义
0x14	2	Touch_Code	按键后，上传的键码（作为触发用户软件的消息）；如果 Touch_Code 的高字节为 0xFF 表示不上传键码。 如果 Touch_Code 的高字节为 0x00-0xF3，表示上传的数据串索引到 0x1A 配置文件，此时（Touch_Code-0xF000）为索引 ID。0x1A 配置文件中，每条索引长度固定为 128 字节，第一个字节为本条索引的有效长度。

自动循环执行指令组配置文件(0x1C)

自动循环执行指令配置文件用来执行需要批量自动运行的指令，使用 0x9A 指令来打

开或关闭，此配置文件由 16（PackID=0~15）个 8KB 的指令组组成，每个指令组固

定 8KB，包含最多 64 条指令，每条指令固定占据 128 字节存储器空间。

每条指令格式定义如 Table 4-3 :

Table 4-3 自动循环执行指令配置文件定义

首地址	宽度 (字节)	定义	说明
0x00	1	Command_Delay	指令执行后延时时间，单位为 8ms，0x00 表示不延时
0x01	1	Command_Length	本条指令长度 0x00：本条指令无效 其它：本条指令长度（0x02 开始计算长度）
0x02- 0x7F	不定	指令	去掉 0xaa 帧头和 0xcc 0x33 0xc3 0xc3c 帧结束符后的标准指令

图标配置文件(0x1D)

图标配置文件最多包含 13107 个图标定义，每个图标定义如下表所示。0x99 指令的 IconID 对应于图标定义在本文件中的索引。比如，本文件包含 20 个图标定义，那么 IconID 的取值范围是 0-19，如 Table 4-4 :

Table 4-4 图标定义

首地址	宽度 (字节)	定义	说明
0x00	2	Pic_ID	图标所在的图片编号
0x02	4	Xs,Ys	图标的左上角坐标
0x06	4	Xe,Ye	图标的右下角坐标

触控自动切换配置文件(0x1E)

触控配置文件用来进行触控界面的自动切换；是由最多 8192 条触控指令组成的二进制文件，

每条触控指令 16 个字节，定义如 Table 4- 5：

Table 4-5 触控界面配置指令定义

首地址	宽度 (字节)	定义	说明
0x00	2	Pic_Now	当前屏幕显示图片的编号； 如果 Pic_Now 的高字节为 0xff 表示触控指令结束。
0x02	4	Xs,Ys	有效触控区域的左上角坐标。
0x06	4	Xe,Ye	有效触控区域的右下角坐标。
0x0A	2	Pic_Next	点击有效触控区域后切换到下一个界面的图片编号； 如果 Pic_Next 的高字节为 0xff 表示不进行界面切换。
0x0C	2	Pic_Cut	触控动画图片编号； 如果 Pic_Cut 的高字节为 0xff 表示没有触控动画图片。
0x0E	2	Touch_Code	点击有效触控区域后，上传的触控键码（作为触发用户软件的消息）； 如果 Touch_Code 的高字节为 0xff 表示不上传触控键码。 如果 Touch_Code 的高字节为 0xf0-0xf3，表示上传的数据串索引到 0x1A 配置文件，此时（Touch_Code-0xf000）为索引 ID。0x1A 配置文件中，每条索引长度固定为 128 字节，第一字节为本条索引的有效长度。