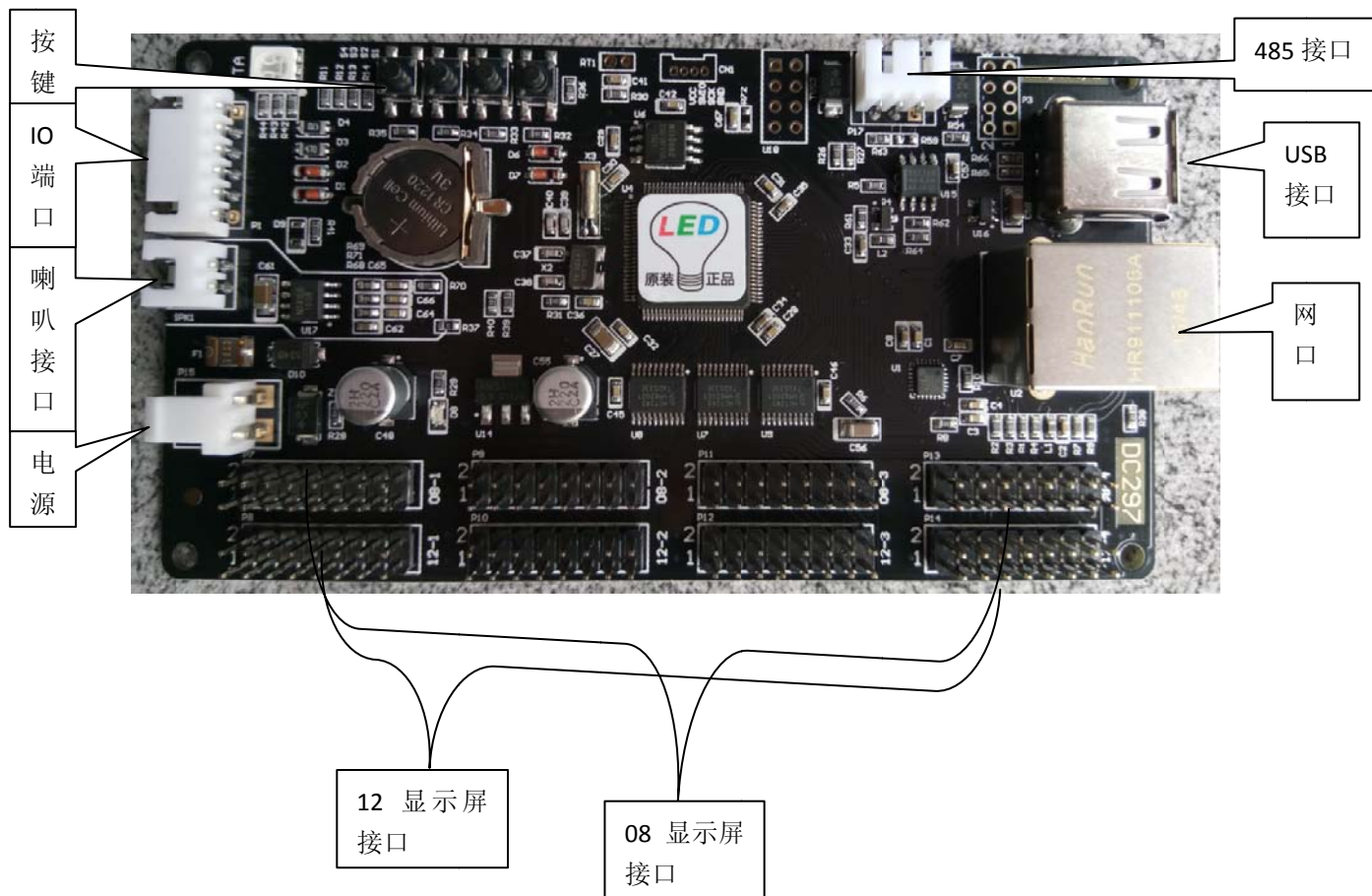


	文档编号	版本	密级
	UG-DC297-317	1.04	开放
	产品名称: LED 显示屏		共 62 页

# 【显示屏使用说明书】





## 文件修改记录

版本	修改日期	修改描述	修改人员
v1.00	2016-3-10	创建此文档	Lyon
v1.01	2016-4-13	增加了动态库	Lyon
v1.02	2016-5-14	1.增加了配置 IO 端口。 2.增加了设置 IO 端口的输出状态。 3.增加了读取 IO 端口的输入状态。 4.增加了同步显示。	Lyon
V1.03	2016-7-15	1.增加 v3.0 指令集 2.优化了 1.5 段落的描述	Lyon
V1.04	2017-4-14	1.增加了恢复出厂设置 2.修改了按键章节	Lyon

## 目录

编写目的 .....	6
概述 .....	7
产品特点: .....	7
基本电气参数: .....	7
1.1 通信接口 .....	7
1.2 屏幕和坐标 .....	8
1.3 GUI 接口 .....	9
1.4 文本存储与文本控件映射 .....	10
1.5 语音播报 .....	10
1.6 如何快速使用显示屏 .....	11
按键功能 .....	13
功能设置向导图: .....	14
常用设置说明: .....	15
功能键说明: .....	15
通信协议 .....	16
2.1 数据包格式 .....	16
2.2 数据校验算法(CRC16) .....	16
2.3 指令集 .....	19
0x01(同步/复位设备端包序列) .....	21
0x05(同步时间) .....	21
0x07(更改通信地址) .....	22
0x08(更改 IP 地址) .....	22
0x09(更改端口) .....	23
0x0A(更改波特率) .....	23
0x0B(更改显示扫描频率) .....	24
0x0C(调整显示亮度) .....	24
0x0D(调整音量) .....	25
0x0E(配置 IO 端口) .....	25
0x0F(设置 IO 输出状态) .....	26
0x10(读取 IO 输入状态) .....	26
0x11(配置 LED 驱动程序) .....	27
0x12(设置 OE 极性) .....	27
0x13(设置数据极性) .....	28
0x14(设置 LE 信号延时) .....	28
0x15(设置网关和掩码) .....	29
0x16(进入安全模式) .....	29
0x18(设置显示尺寸) .....	29
0x19(设置显示方向) .....	30
0x1A(设置 485 工作模式) .....	30
0x1E(恢复出厂设置) .....	31
0x30(播放语音) .....	31
0x31(停止播放语音) .....	32

0x38(画点).....	32
0x39(画线).....	32
0x3A(画矩形).....	33
0x3B(填充矩形).....	33
0x3C(画圆).....	34
0x3D(填充圆).....	34
0x3E(绘制位图).....	35
0x3F(显示文字).....	36
0x40(设置画笔/文字颜色).....	36
0x41(设置背景颜色).....	37
0x42(设置字体).....	37
0x43(绘制系统自带的位图).....	38
0x50(显示窗口).....	38
0x51(隐藏窗口).....	39
0x52(删除窗口).....	39
0x53(自动同步窗口).....	40
0x60(创建文本控件).....	40
0x61(删除文本控件).....	41
0x62(设置文本).....	41
0x63(设置文字字体).....	44
0x64(设置文字颜色).....	44
0x65(设置背景颜色).....	45
0x66(停止显示).....	45
0x67(下载文本文件).....	45
0x68(显示文本文件).....	46
0x69(打开文本文件).....	47
0x6A(关闭文本文件).....	47
0x6B(删除文本文件).....	47
0x6C(同步显示).....	48
0x6D(单包同步显示, 带语音).....	49
0x80(创建图片控件).....	49
0x81(删除图片控件).....	50
0x82(追加图片).....	50
0x83(清除图片列表).....	51
2.4 显示时间.....	52
显示屏控制卡尺寸.....	53
附表 1.....	54
语音列表: .....	54

## 编写目的

本文档循序渐进的介绍了显示屏的整体面貌，详细的阐述了显示屏的功能模块和通信协议，让用户能够快速了解显示屏的基本功能特性和操作方式，并将重点转移到应用层面中。同时介绍了显示屏协议栈的使用，在通信协议的章节中每条指令都给出了操作例子，以致力于读者快速阅读后，能够容易理解。

## 概述

### 产品特点:

- ◆ 支持 4 路 HUB08 接口和 4 路 HUB12 接口，最大支持分辨率为 128\*512 双色和单色。
- ◆ 支持 UDP、TCP 网络通信接口、RS485 通信接口和 WIFI 网络通信接口(可选)。
- ◆ 支持 U 盘下载文本文件、图标文件、语音库文件。
- ◆ 支持简体中文、繁体中文、英语文字显示，支持 6 种字体。
- ◆ 支持语音播报功能，10 级音量调节。可以播报停车场常用多达 300 多种真人语音，语音库可扩展。
- ◆ 扫描频率最高达 960Hz, 移动时稳定无重影，字迹清晰。全户外灯芯，直射仍清晰可见，10 级亮度调节。
- ◆ 板载温度和湿度(可选)传感器。
- ◆ 提供 4 路 IO 线路，可以控制继电器或者作为地感触发输入。
- ◆ 板载实时时钟功能(RTC)，最大误差为每天+/-3 秒。
- ◆ 可脱离 PC 通过设置地址、波特率、时间等参数。
- ◆ 提供通信协议栈动态库和静态库以及 PC 端调试工具。

### 基本电气参数:

显示卡供电电压:5V

模组供电电压:5V

工作温度: -20℃~+80℃

显示卡最大功耗: 7.5W (包括音频放大器)

音频输出功率:4 欧姆 5W

模组最大功耗: (普亮) 15W/64\*16 (Φ 3.75), (高亮) 25W/64\*16 (Φ 3.75),

无故障工作时间: ≥60000 小时

### 1.1 通信接口

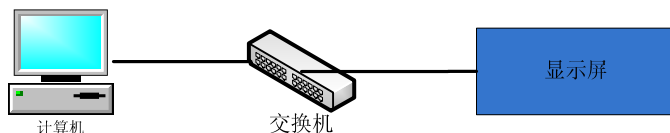


图 1.1A 通过网络连接显示屏的典型应用图

显示屏通信接口支持 UDP、TCP/IP 网络和 RS485 两种接口，通过网络通信时，显示屏连接到交换机或者路由器上，主机端通过局域网或者外网进行远程控制，网络通信是基于 UDP 或者 TCP/IP 协议栈实现面向连接的 C/S 通信架构。出厂的默认 IP 地址为 192.168.0.10 服务端口为 7，用户可以通过软件或者按键更改 IP 和端口地址。

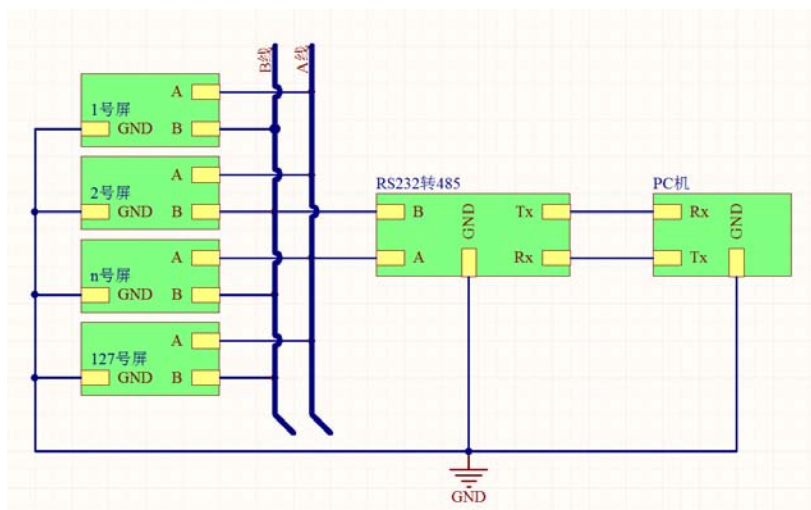


图 1.1B 显示屏与 PC 机通过 RS485 总线连接图

使用 RS485 接口通信时，只支持一主多从的半双工总线通信结构，单节点最多可以挂接 255 个显示卡，每个显示屏通过设备地址区分来寻址通信。波特率支持 2400、9600、19200、115200 四种，出厂默认的设置 19200,E,8,1。用户可以通过软件或者按键更改波特率和通信地址。

## 1.2 屏幕和坐标

屏幕由许多可以单独控制的点组成，这些点称为像素。显示屏控制器提供了绝大部分文本和绘图功能，可以在任何指定的像素上进行文字显示或者绘制图形。

水平尺度称为 X 轴，垂直尺度称为 Y 轴。坐标定义为一对由 X 和 Y 值组成的值 (X, Y)。显示屏（或窗口）左上角的坐标默认为(0,0)。正的 X 值始终向右，正的 Y 值始终向下。下图说明了坐标系以及 X 和 Y 轴的方向。

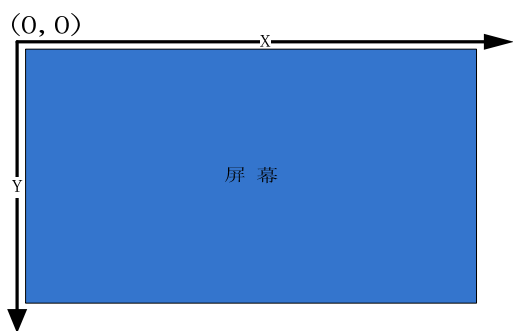


图 1.2 屏幕坐标系



## 1.3 GUI 接口

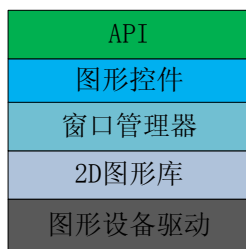


图 1.3A 显示屏的 GUI 软件框架

显示屏软件提供了基本的绘图和文字显示接口，包括 2D 图形库的点、线、矩形、圆、文字等基本的绘图操作。还提供了更高级的图形控件，这些控件都属于窗口类，是针对文字和图形应用进行了参数和方法包装。文本控件用来显示文字信息，并且支持多种动态显示效果。位图控件用来显示一些小图标，这些图标都是预先存储在控制器中的，用户只需要按照图标的编号调用显示即可。

### 窗口 Z 序

窗口实际上就是在屏幕上划分的一个矩形区域，是窗口管理器管理的基本单位，窗口与窗口之间拥有 Z 序列，如图 1.3B 所示。在最上面的窗口，称作顶层窗口，屏幕显示的窗口始终是在 Z 序顶层的窗口。这可以理解为最上面的窗口把下面的窗口都覆盖了，所有下面的窗口不可见。用户在创建窗口时，应该注意 Z 序的存在，最好的做法就是将创建的窗口位置放置于屏幕上没有任何其它窗口的位置。

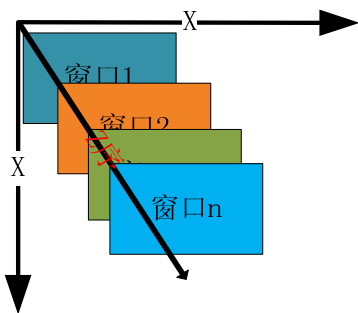
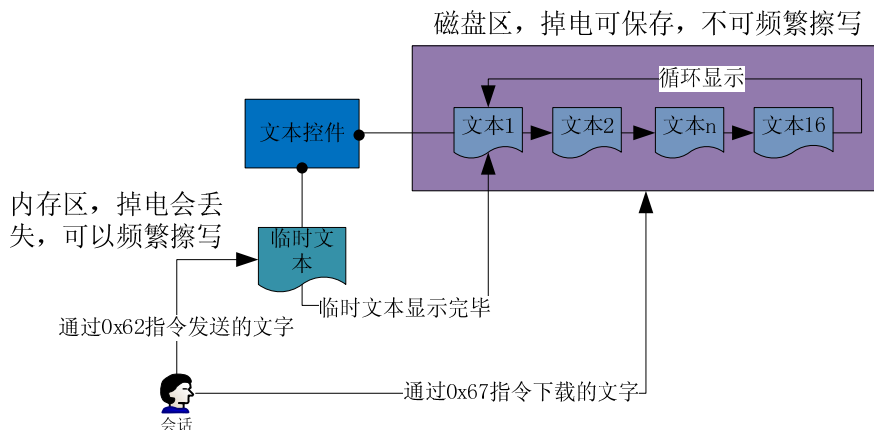


图 1.3B 窗口 Z 序示意图

## 1.4 文本存储与文本控件映射

文本文件是用来存放用户下载的文字信息的一种文档资料。文本文件只能在文本控件上显示。显示屏的文本文件的存储组织结构和文本控件的映射关系如下图所示。



按照文本的存储区域可以分为内存区和磁盘区，内存区的文本是通过 0X62 指令下载到文本控件的缓存中，这部分内容在显示完成之后或者掉电之后就会丢失，因此常用于存放一些临时信息。而磁盘区的文本通过 0x67 指令下载到磁盘中，由文件系统管理，在掉电之后会保存下来，但不建议频繁的擦写磁盘区的文件，这会加快存储器的老化速度，最终导致失效。关于文本文件的大小限制，内存区最大为 2K 字节，相当于 1K 个汉字。磁盘区最大为 60K 字节，相当于可以存储 3W 个汉字。

只有窗口 ID 为 0~7 的文本控件才分配了磁盘存储空间，其它 ID 的文本控件只能存放在内存区，ID 为 0~7 的文本控件被创建之后，会自动调用关联的文本文件，并且依次循环显示，如上图紫色框指示的那样。如果用户不希望磁盘区的文本被显示，可以通过关闭、打开、删除等命令对磁盘区的文本文件进行管理。

## 1.5 语音播报

显示屏系统储存了停车场常用的 300 句短语或词组，用户可以直接调出播放，语音列表参见附表 1。语音文本的匹配方式采用词组或者短语进行匹配，在一些含有变量信息的文字中，如金钱、车牌号、日期等，显示屏会自动处理。在发送语音文本时，需要将词组和短语之间加入逗号或者句号分隔，显示屏才能正确的匹配。显示屏软件提供了语音播放队列(FIFO)，用户可以连续发送最多 50 条语音文本到队列中，显示屏会自动按照队列的顺序逐个播放；这些特性对语音组合使用提供了便利。用户还可以通过 U 盘下载自定义的语音库文件。

## 1.6 如何快速使用显示屏

用户在拿到显示屏之后，首先阅读本文档的前几章节，对产品的整个功能特性有了初步的了解之后，然后在自己的目标平台进行联机调试。在调试之前，用户仅要做的事情就是将随本文档一起发行的通信协议栈的库文件加入到自己项目中，然后编写自己平台的数据收发接口，就可以通过 API 很方便的操作显示屏，实现丰富多彩的应用。

协议栈封装了显示屏的所有功能集合和数据格式，作为应用程序和显示屏设备的中间层，向上层提供了简便的应用接口(API)，向下提供了协议栈的数据流输入输出接口，见图 1.5。协议栈源代码全部采用 C 语言开发完成，内部的数据处理不依赖任何外部的库文件和硬件相关性，与平台相关的部分仅在协议栈的数据输出和输入，这部分需要用户自己实现，协议栈的数据输出和输入可以通过 UDP、TCP Socket 或者串口设备实现。用户只需要简单的实现底层的数据输入输出接口，无需关心内部的数据处理过程，通过 API 就可以操作显示屏，让用户省掉了繁琐的通信接口开发过程，快速的应用于项目中，这大大缩短了开发周期。

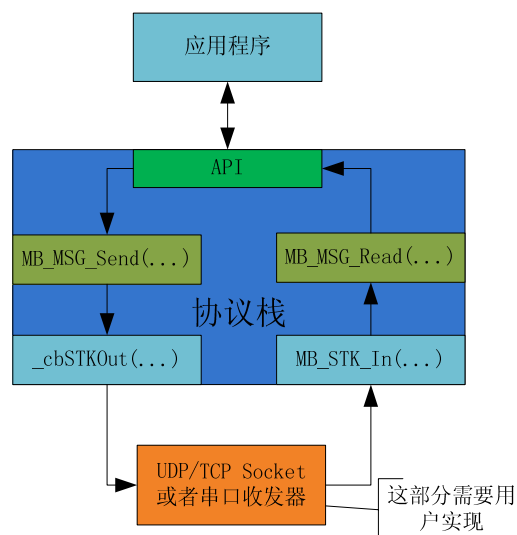


图 1.5 协议栈软件框架

### 协议栈库文件使用重点：

1.实现数据输出的回调函数，函数类型为 `cbSTKOut_t`，包含 2 个参数，第一参数为数据的缓存地址，第二个参数为数据长度。回调函数的内部可以使用 UDP/TCP 或者串口实现。

在每次发送完数据之后，应该等待接收显示屏控制器返回的数据，超时的时间应不小于 50 毫秒，然后将收到的数据通过 `int8_t MB_STK_In(uint8_t * pBuff,uint16_t Size)`传给协议栈。

用 UDP 实现数据输出回调函数的编写流程：

- #1 创建客户端 UDP socket，（只需要连接一次，用全局标记记录）
- #2 定义要连接的服务端口信息(默认的是 192.168.0.10:7)
- #3 连接显示屏控制卡（只需要连接一次，用全局标记记录）
- #4 发送数据到显示屏控制卡
- #5 接收显示屏控制卡的返回数据
- #6 调用 `int8_t MB_STK_In(uint8_t * pBuff,uint16_t Size)`将收到的数据传给协议栈。

用 TCP 实现数据输出回调函数的编写流程：

- #1 创建客户端 TCP socket
- #2 定义要连接的服务端口信息(默认的是 192.168.0.10:7)

#3 连接显示屏控制卡

#4 发送数据到显示屏控制卡

#5 接收显示屏控制卡的返回数据

#6 调用 `int8_t MB_STK_In(uint8_t * pBuff,uint16_t Size)`将收到的数据传给协议栈。

#7 释放 `socket` 并结束一次回话。

用串口设备实现数据输出回调函数的编写流程:

#1 打开串口设备 (可以全局打开, 也可以每次重新打开)

#2 配置号串口波特率和校验位(默认为 19200,N,8,1)

#3 发送数据到显示屏控制卡

#4 接收显示屏控制卡的返回数据

#5 调用 `int8_t MB_STK_In(uint8_t * pBuff,uint16_t Size)`将收到的数据传给协议栈。

#6 关闭串口设备 (可以全局关闭, 也可以每次都关闭)。

2.在使用协议栈之前先调用 `void MB_STK_SetOutCallback(cbSTKOut_t cb)`设置数据输出回调函数。

3.调用协议栈 API 接口实现自己的应用程序。

另外需要注意的是协议栈的 API 函数不支持多个线程操作, 如果用户需要在多个线程下使用协议栈, 必须加入线程同步机制, 以保证协议栈的数据时序要求。

目录文档说明:

DLL/DUCP.h 协议栈公共头文件。

DLL/DUCP.dll 协议栈动态库。

DLL/DUCP.lib 协议静态库。

DLL/DUCP\_Host.h 协议栈主机端头文件。

DLL/MBmsg.h 协议栈消息层头文件。

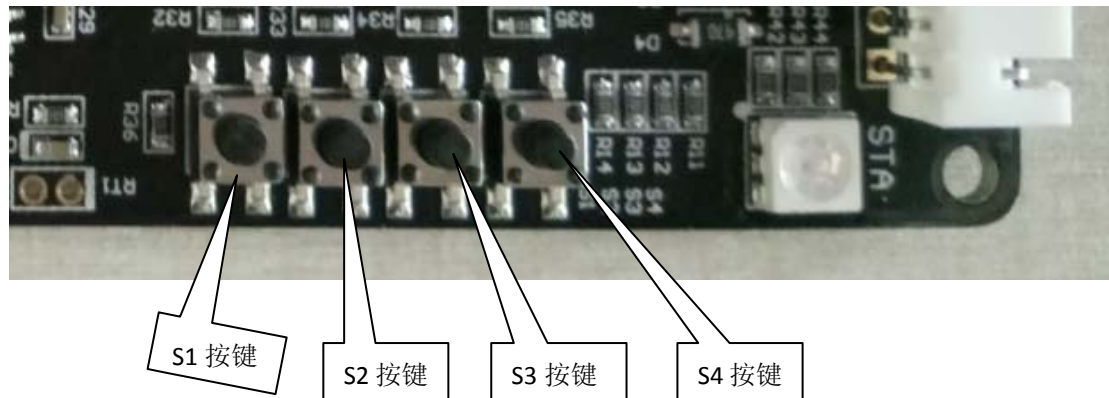
VC/MBPHY.c VC 平台下的协议栈底层数据输入输出接口源文件, 支持 UDP/TCP 和串口设备。

VC/MBPHY.h VC 平台下的协议栈底层数据输入输出接口头文件, 支持 UDP/TCP 和串口设备。

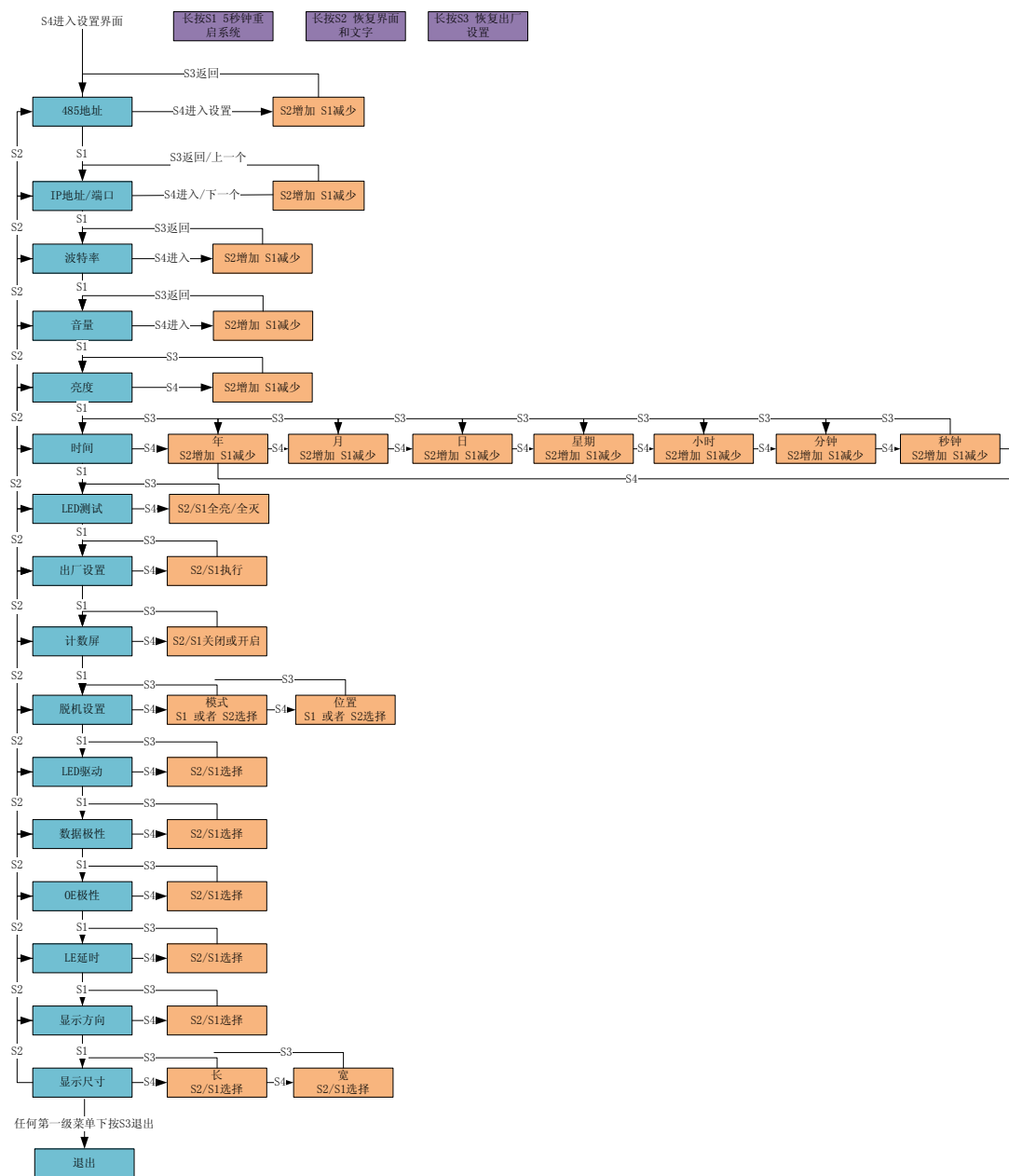
Tool/LEDDebugger.exe PC 平台下的调试工具。

## 按键功能

显示卡板载了 4 个按键，用来设置显示卡的一些参数，包括了 485 通信地址、波特率、IP 地址、音量、亮度、恢复出厂设置、设置显示屏的各种模式等等。



功能设置向导图:



- S4: 主菜单。 (复用功能: 确认修改)
- S1: 下翻菜单。 (复用功能: 减 1)
- S2: 上翻菜单。 (复用功能: 加 1)
- S3: 退出菜单。

使用例子: 亮度: 100%修改为亮度: 50%应用;

- 1、S4: 进入主菜单模式: 地址: 000。
- 2、S1: 下翻菜单, 切换到亮度: 100 模式。
- 3、S4: 确认修改。
- 4、S1: 减 1 操作。
- 5、S3: 退出菜单。“单次退出修改模式”, “二次退出主菜单”。

### 常用设置说明:

1. 更改 485 地址:
  - a. 按 S4 进入主菜单。
  - b. 按 S4 进入设置模式。
  - c. 按 S2 加地址或者按 S1 减地址。
  - d. 按 S3 返回主菜单。
  - e. 再按 S3 退出主菜单。
2. 更改 IP 地址
  - a. 按 S4 进入主菜单。
  - b. 按 S1 下翻页选择 IP 地址界面。
  - c. 按 S4 进入设置模式。
  - d. 按 S2 加地址或者按 S1 减地址。
  - e. 按 S4 选择下一个设置点, 或者 S3 选择上一个设置点。
  - f. 设置完成后按 S3 返回到主菜单。
  - g. 再按 S3 退出主菜单。
3. 恢复出厂
  - a. 按 S4 进入主菜单。
  - b. 按 S1 下翻页选择出厂设置界面。
  - c. 按 S4 进入
  - d. 按 S2 或 S1 确定
  - e. 恢复成功之后, 显示屏会自动重启
4. 更改显示方向
  - a. 按 S4 进入主菜单。
  - b. 按 S1 下翻页选择显示方向界面。
  - c. 按 S4 进入设置
  - d. 按 S1 或者 S2 选择显示方向
  - e. 按 S3 返回至主菜单。
  - f. 再按 S3 退出主菜单。

### 功能键说明:

1. 长按 S1 5 秒钟重启显示卡。
2. 长按 S2 5 秒钟恢复显示界面和文字至出厂值。
3. 长按 S3 5 秒钟恢复出厂参数设置。

## 通信协议

通信协议在这里是指显示屏应用层软件协议，主要功能是用来实现面向终端连接的数据传输，其规范了设备端(显示屏)与主机端的通信所必须遵循的规则和约定。关于传输介质的标准或者接口协议请参见相关的文档。显示屏目前支持 6 类指令，分别为系统管理、文件管理、语音接口、2D 图形接口、窗口管理、窗口控件管理。主机通过这些指令接口，可以命令显示屏实现文字显示、绘图、播报语音等操作。

### 2.1 数据包格式

显示屏通信是以包为最小单元进行数据交互，数据包可以看作是运输数据的载体。在本协议中数据包结构由控制域、数据域、校验域 3 部分组成，见下图。

数据包格式							
字段	控制域			数据域			校验域
	DA (设备地址)	SP (服务端口)	PN (16 位包序列)	CMD (指令 ID)	DL (数据长度)	DATA (数据)	CRC (16 位包校验值)
取值范围	0x00~0xff	0x00~0xff	0x0000~0xffff	0x00~0xff	0x00~0xff	0x00~0xff	0x0000~0xffff
长度	1Byte	1Byte	2Bytes	1Byte	1Byte	Max 255Bytes	2Bytes

**DA:**为显示屏的地址，取值范围为 0x00 - 0xFF。

**SP:**描述了协议层的服务端口，目前作为保留值，固定设置为 0x64。

**PN:**是包的序列，在传输超过 255 个字节的数据时，需要分包传输，PN 表示了包的序列，每次交互完之后自增 1，设置为最大值 0xFFFF 时表示当前包是最后一个包。在传输小于 255 个字节的数据时，该值应该设置为 0xFFFF。

**CMD:**该字段描述了该包的作用，显示屏通过这个值完成不同的功能服务。

**DL:**该字段用来描述 DATA 的数据长度，最大取值为 255。

**DATA:**是参数数据，每条指令携带的参数和长度是不同的，详解参见后指令集章节。

**CRC:**数据包的校验码。参与校验的字段是从 DA 到 DATA 的最后一个字节。校验算法采用 CRC16，见后章节详解。

### 2.2 数据校验算法(CRC16)

CRC16 是数据通信领域中最常用的一种查错校验码，其特征是信息字段和校验字段的长度可以任意选定。循环冗余检查 (CRC) 是一种数据传输检错功能，对数据进行多项式计算，并将得到的结果附在帧的后面，接收设备也执行类似的算法，以保证数据传输的正确性和完整性。

参与计算的字段为 DA 到 DATA 之间的数据。计算得到的 16 位校验码在传送时，采用小端模式，低字节在前，高字节在后。

算法 C 实现:

```
static const uint8_t _CRChi[] = {
    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,
```





```
/*
    CRC16 校验函数
*/
uint16_t MB_CRC16(uint8_t *pFrame, uint16_t count)
{
    uint8_t CRCHi = 0xFF;
    uint8_t CRCLo = 0xFF;
    int32_t index;
    while(count--) {
        index = CRCLo ^ *(pFrame++);
        CRCLo = (uint8_t)(CRCHi ^ _CRCHI[index]);
        CRCHi = _CRCLo[index];
    }
    return (uint16_t)(CRCHi << 8 | CRCLo);
}
```

# 指令集

请求参数格式	回复参数格式	备注	ver
	ACK(成功为 0, 失败返回非 0 的值)	没有使用	v2.1
	ACK(成功为 0, 失败返回非 0 的值)	常用	
	ACK(成功为 0, 失败返回非 0 的值)	没有使用	
	PSWD[6]	没有使用	
	NAME[8] + SVR[8] + XSIZE[4] + YSIZE[4] + UPTIME[4]	没有使用	
S	ACK(成功为 0, 失败返回非 0 的值)	常用	
	ACK(成功为 0, 失败返回非 0 的值)	一般	
	ACK(成功为 0, 失败返回非 0 的值)	常用	
	ACK(成功为 0, 失败返回非 0 的值)	常用	
	ACK(成功为 0, 失败返回非 0 的值)	常用	
	ACK(成功为 0, 失败返回非 0 的值)	常用	
	ACK(成功为 0, 失败返回非 0 的值)	高级应用	
	ACK(成功为 0, 失败返回非 0 的值)	高级应用	
	ACK(成功为 0, 失败返回非 0 的值)	高级应用	
	ACK(成功为 0, 失败返回非 0 的值)	高级应用	
	ACK(成功为 0, 失败返回非 0 的值)	高级应用	
	ACK + SF[4]	高级应用	
	ACK(成功为 0, 失败返回非 0 的值)	高级应用	v3.0
	ACK(成功为 0, 失败返回非 0 的值)	高级应用	
	ACK(成功为 0, 失败返回非 0 的值)	高级应用	
	ACK(成功为 0, 失败返回非 0 的值)	高级应用	
	ACK(成功为 0, 失败返回非 0 的值)	高级应用	
	ACK(成功为 0, 失败返回非 0 的值)	高级应用	
	PAPT[4]	高级应用	
	ACK(成功为 0, 失败返回非 0 的值)	高级应用	
	ACK(成功为 0, 失败返回非 0 的值)	高级应用	
	ACK(成功为 0, 失败返回非 0 的值)	高级应用	5.9
	ACK(成功为 0, 失败返回非 0 的值)	一般	
	ACK(成功为 0, 失败返回非 0 的值)	不公开	
	ACK(成功为 0, 失败返回非 0 的值)	不公开	
	ACK(成功为 0, 失败返回非 0 的值)	不公开	
	ACK(成功为 0, 失败返回非 0 的值)	不公开	
	DATA[MAX 255]	不公开	
	ACK(成功为 0, 失败返回非 0 的值)	不公开	
	ACK(成功为 0, 失败返回非 0 的值)	不公开	
	ACK(成功为 0, 失败返回非 0 的值)	不公开	
	ACK(成功为 0, 失败返回非 0 的值)	不公开	
	ACK(成功为 0, 失败返回非 0 的值)	不公开	

# 显示屏使用说明书

+ DM + DT + EXM + EXS + FINDEX + TC[4] + BC[4] + TL[2] + TEXT[...]	ACK (成功为 0, 失败返回非 0 的值)	常用	
	ACK (成功为 0, 失败返回非 0 的值)	常用	
	ACK (成功为 0, 失败返回非 0 的值)	常用	
	ACK (成功为 0, 失败返回非 0 的值)	常用	
	ACK (成功为 0, 失败返回非 0 的值)	常用	
ID2x + TWIDnx + FIDnx + TWID(n+1)x + FID(n+1)x	ACK (成功为 0, 失败返回非 0 的值)	一般	v2.2
+EXM +EXS + FINDEX + DRS + TC + BC + TL[2] + TEXT[...] + VOICE[...]	ACK (成功为 0, 失败返回非 0 的值)	一般	v3.0
ZE[2] + YSIZE[2]	ACK (成功为 0, 失败返回非 0 的值)	高级应用	
	ACK (成功为 0, 失败返回非 0 的值)	高级应用	
+ SM + ST + EXM + EXS	ACK (成功为 0, 失败返回非 0 的值)	高级应用	
	ACK (成功为 0, 失败返回非 0 的值)	高级应用	

## 0x01(同步/复位设备端包序列)

在分包传输大数据时，比较可靠的做法是在发送数据之前，先通过该指令同步/复位设备端的包序列。这样做的目的是为了纠正在通信过程中，可能由于设备断电或者通信链路故障等导致丢失数据包，而失步；从而在之后的接收过程中出错。设备端在接收到该指令时，自动复位 PN 为 0XFFFF。主机端的 PN 在传输过程中应当从 0 计数开始，直到最后一包将 PN 设置为 0XFFFF。需要注意的是，发送的数据在第一包就能全部发送完毕，那么第一包的 PN 应该是 0XFFFF 而不是 0X0000，PN 等于 0XFFFF 永远只在最后一包设置；在不支持分包传输的指令中，PN 的取值可以不关心。

**请求格式:**DA + SP + PN[2] + 0x00 + DL + CRC[2]

**请求参数描述:**无参数 DL 取值为 0。

**回复格式:**DA + SP + PN[2] + 0x00 + DL + ACK + CRC[2]

**回复参数描述:**ACK 是显示屏返回的结果，长度为 1 个字，取值为 0 表示成功，非 0 表示不成功。

**使用例子:** 同步/复位地址为 0 的显示屏。

**主机发送:** 00 64 FF FF 01 00 70 67

**设备回复:** 00 64 FF FF 01 01 00 67 74

## 0x05(同步时间)

同步时间的功能是从外部同步显示屏的系统时间。显示屏的系统时间每天的误差在+- 5 秒钟之内，因此在需要极高精度的时间应用场合，应该每隔一段时间对显示屏同步时间。

**请求格式:**DA + SP + PN[2] + 0x05 + DL + Y[2] + M + D + W + H + N + S + CRC[2]

**请求参数描述:**包含 8 个字节的参数，DL 取值为 8。

**Y:**为年的 16 位数据，小端模式，取值范围在 1970~2099。

**M:**为月的 8 位数据,取值范围在 1~12。

**D:**为日的 8 位数据，取值范围在 1~31。

**W:**为星期的 8 位数据，取值范围在 1~7。1 对应星期日，2~7 对应星期一至六。

**H:**为小时的 8 位数据,取值范围在 0~23。

**N:**为分钟的 8 位数据，取值范围在 0~59。

**S:**为秒钟的 8 位数据，取值范围在 0~59。

**回复格式:** DA + SP + PN[2] + 0x05 + DL + ACK + CRC[2]

**回复参数描述:**包含 1 个字节的参数，DL 取值为 1，ACK 是显示屏返回的结果，取值为 0 表示成功，非 0 表示不成功。

**使用例子:** 对地址为 0 的显示屏进行同步时间至 2016 年 3 月 10 日星期四 15:05:06。

**主机发送:** 00 64 FF FF 05 08 E0 07 03 0A 07 0F 05 06 65 68

**设备回复:** 00 64 FF FF 05 01 00 26 B5

## 0x07(更改通信地址)

通信地址是用来区分多个显示屏设备的身份标识。主机在进行任何请求时，通过指定通信地址来选择目标设备。主机通过该指令更改了设备端的地址之后，在随后的操作中应当使用新地址来通信。

**请求格式:**DA + SP + PN[2] + 0x07 + DL + NDA + CRC[2]

**请求参数描述:**包含 1 个字节的参数，DL 取值为 1。

**NDA:**为新的通信地址，取值范围在 0~255。

**回复格式:**DA + SP + PN[2] + 0x07 + DL + ACK + CRC[2]

**回复参数描述:**包含 1 个字节的参数，DL 取值为 1，ACK 是显示屏返回的结果，取值为 0 表示成功，非 0 表示不成功。

**使用例子:** 将地址为 0 的显示屏的通信地址更改为 1。

**主机发送:** 00 64 FF FF 07 01 01 46 B5

**设备回复:** 00 64 FF FF 07 01 00 87 75

## 0x08(更改 IP 地址)

IP 地址只有在网络通信中才起作用，同通信地址功能一样作为显示屏的身份标识。主机在通过该指令更改完成之后，需要重新启动显示屏才能生效。同时主机端需要用新的 IP 地址连接显示屏。

**请求格式:**DA + SP + PN[2] + 0x08 + DL + IP[4] + CRC[2]

**请求参数描述:**包含 4 个字节的参数，DL 取值为 4。

**IP:**为新的 IP 地址，长度为 4 个字节，每个字节取值范围在 0~255。

**回复格式:**DA + SP + PN[2] + 0x08 + DL + ACK + CRC[2]

**回复参数描述:**包含 1 个字节的参数，DL 取值为 1，ACK 是显示屏返回的结果，取值为 0 表示成功，非 0 表示不成功。

**使用例子:** 将地址为 0 的显示屏的 IP 地址更改为 192.168.0.188。

**主机发送:** 00 64 FF FF 08 04 C0 A8 00 BC E6 17

**设备回复:** 00 64 FF FF 08 01 00 B7 76

## 0x09(更改端口)

端口是指在网络通信中的 TCP/IP 服务端口。显示屏软件启动之后，显示屏协议栈一直监听该端口，主机通过连接到该服务端口与显示屏完成会话操作。主机在通过该指令更改完成之后，需要重新启动显示屏才能生效。同时主机端需要用新的端口访问显示屏。

**请求格式:**DA + SP + PN[2] + 0x09 + DL + PORT + CRC[2]

**请求参数描述:**包含 1 个字节的参数，DL 取值为 1。

**PORT:**为新的端口，取值范围在 0~255。

**回复格式:**DA + SP + PN[2] + 0x09 + DL + ACK + CRC[2]

**回复参数描述:**包含 1 个字节的参数，DL 取值为 1，ACK 是显示屏返回的结果，取值为 0 表示成功，非 0 表示不成功。

**使用例子:**将地址为 0 的显示屏的端口更改为 7。

**主机发送:**00 64 FF FF 09 01 07 A7 74

**设备回复:**00 64 FF FF 09 01 00 E6 B6

## 0x0A(更改波特率)

波特率是指在串口设备通信中的通信速率，即指每秒钟的比特位 BPS。速率越高每秒钟发送的数据位就越多，速度就越快。显示屏支持的波特率范围在 2400 ~ 115200。在实际应用中建议使用标准的 2400、4800、9600、19200、38400、57600、115200，可以保证最小的频差。

主机通过该指令更改成功之后，需要使用新的波特率与显示屏通信。

**请求格式:**DA + SP + PN[2] + 0x0A + DL + BAUD[4] + CRC[2]

**请求参数描述:**包含 4 个字节的参数，DL 取值为 4。

**BAUD:**为新的波特率，取值范围在 2400~115200。

**回复格式:**DA + SP + PN[2] + 0x0A + DL + ACK + CRC[2]

**回复参数描述:**包含 1 个字节的参数，DL 取值为 1，ACK 是显示屏返回的结果，取值为 0 表示成功，非 0 表示不成功。

**使用例子:**将地址为 0 的显示屏的波特率更改为 19200。

**主机发送:**00 64 FF FF 0A 04 00 4B 00 00 2B B2

**设备回复:**00 64 FF FF 0A 01 00 16 B6

## 0x0B(更改显示扫描频率)

显示屏扫描频率是指显示屏的帧更新速度。在光学动态扫描技术中，根据人眼的视觉惰性，对每秒钟变化高于 50 次的图像将基本看不出闪烁。扫描频率越高，图像越稳定。显示屏的扫描频率支持 30 HZ~ 960HZ。需要注意的是，显示屏的窗口更新时基都来源于显示屏的场同步信号，因此调整扫描频率，同时会改变窗口的更新速度。

**请求格式:**DA + SP + PN[2] + 0x0B+ DL + FRE[2] + CRC[2]

**请求参数描述:**包含 2 个字节的参数，DL 取值为 2。

**FRE:**为新的扫描频率，由 2 个字节组成 16 位数据，小端模式，取值范围在 10~960。

**回复格式:** DA + SP + PN[2] + 0x0B+ DL + ACK + CRC[2]

**回复参数描述:**包含 1 个字节的参数，DL 取值为 1，ACK 是显示屏返回的结果，取值为 0 表示成功，非 0 表示不成功。

**使用例子:** 将地址为 0 的显示屏的扫描频率更改为 480HZ。

**主机发送:** 00 64 FF FF 0B 02 E0 01 4E 32

**设备回复:** 00 64 FF FF 0B 01 00 47 76

## 0x0C (调整显示亮度)

显示屏的明亮程度可以通过该指令从 10%~100% 调节。

**请求格式:**DA + SP + PN[2] + 0x0C+ DL + LIGHT + CRC[2]

**请求参数描述:**包含 1 个字节的参数，DL 取值为 1。

**LIGHT:**为新的亮度等级百分比，取值范围在 10~100。

**回复格式:** DA + SP + PN[2] + 0x0C+ DL + ACK + CRC[2]

**回复参数描述:**包含 1 个字节的参数，DL 取值为 1，ACK 是显示屏返回的结果，取值为 0 表示成功，非 0 表示不成功。

**使用例子:** 将地址为 0 的显示屏的亮度更改为 50%。

**主机发送:** 00 64 FF FF 0C 01 32 77 62

**设备回复:** 00 64 FF FF 0C 01 00 F6 B7



## 0x0D(调整音量)

显示屏的音量等级可以通过该指令从 10%~100% 调节。

**请求格式:**DA + SP + PN[2] + 0x0D+ DL + VOL+ CRC[2]

**请求参数描述:**包含 1 个字节的参数，DL 取值为 1。

VOL:为新的音量等级百分比，取值范围在 10~100。

**回复格式:** DA + SP + PN[2] + 0x0D+ DL + ACK + CRC[2]

**回复参数描述:**包含 1 个字节的参数，DL 取值为 1，ACK 是显示屏返回的结果，取值为 0 表示成功，非 0 表示不成功。

**使用例子:** 将地址为 0 的显示屏的亮度更改为 50%。

**主机发送:**00 64 FF FF 0D 01 32 26 A2

**设备回复:**00 64 FF FF 0D 01 00 A7 77

## 0x0E(配置 IO 端口)

显示屏控制卡提供了 4 路 IO 线路，每路能独立的配置为输入或输出模式。作为触发输入模式时可以连接到外部的信源设备，例如地传感器的输出信号。作为输出模式时，可以驱动外部的继电器。

**请求格式:**DA + SP + PN[2] + 0x0E+ DL + PIN + PF+ CRC[2]

**请求参数描述:**包含 2 个字节的参数，DL 取值为 2。

PIN:为端口线路标号，取值范围见下表

PIN 取值含义	
取值	描述
0x00	线路 1
0x01	线路 2
0x02	线路 3
0x03	线路 4
0xF0	所有线路

PF:为引脚功能，取值范围见下表

FF 取值含义	
取值	描述
0x00	触发输入模式
0x01	输出模式

**回复格式:** DA + SP + PN[2] + 0x0E+ DL + ACK + CRC[2]

**回复参数描述:**包含 1 个字节的参数，DL 取值为 1，ACK 是显示屏返回的结果，取值为 0 表示成功，非 0 表示不成功。

**使用例子:** 将地址为 0 的显示屏的端口 1 配置为输出模式。

**主机发送:**00 64 FF FF 0E 02 01 01 06 AE

**设备回复:** 00 64 FF FF 0E 01 00 57 77

## 0x0F(设置 IO 输出状态)

使用本指令之前，相应的端口必须配置为输出模式，否则操作无效。输出端口的状态为数字量，高电平用 1 表示。低电平用 0 表示。

**请求格式:**DA + SP + PN[2] + 0x0F+ DL + PIN + SF[4]+ CRC[2]

**请求参数描述:**包含 5 个字节的参数，DL 取值为 5。

**PIN:**为端口线路标号，取值范围见下表

PIN 取值含义	
取值	描述
0x00	线路 1
0x01	线路 2
0x02	线路 3
0x03	线路 4
0xF0	所有线路赋值
0xF1	所有线路逻辑或
0xF2	所有线路逻辑与
0xF3	所有线路逻辑异或

**SF:**为端口状态位标志，32 位数据，小端模式。在 PIN 取值为 0x00~0x03 时，非 0 表示输出 1，反之为 0。当 PIN 取值为 0xF0~0xF3 时，表示掩码值，bit[0:3]对应线路 1 至线路 4。

**回复格式:** DA + SP + PN[2] + 0x0F+ DL + ACK + CRC[2]

**回复参数描述:**包含 1 个字节的参数，DL 取值为 1，ACK 是显示屏返回的结果，取值为 0 表示成功，非 0 表示不成功。

**使用例子:** 将地址为 0 的显示屏的端口 1 输出高电平。

**主机发送:** 00 64 FF FF 0F 05 01 01 00 00 00 8D 16

**设备回复:** 00 64 FF FF 0F 01 00 06 B7

## 0x10(读取 IO 输入状态)

使用本指令之前，相应的端口必须配置为输入模式，否则操作无效。输入端口的状态为数字量，高电平用 1 表示。低电平用 0 表示。

**请求格式:**DA + SP + PN[2] + 0x10+ DL + PIN+ CRC[2]

**请求参数描述:**包含 1 个字节的参数，DL 取值为 1。

**PIN:**为端口线路标号，取值范围见下表

PIN 取值含义	
取值	描述
0x00	线路 1
0x01	线路 2
0x02	线路 3
0x03	线路 4
0xF0	所有线路

**回复格式:** DA + SP + PN[2] + 0x10+ DL + ACK+ SF[4] + CRC[2]

**回复参数描述:**包含 5 个字节的参数, DL 取值为 5。

**ACK** 是显示屏返回的结果, 取值为 0 表示成功, 非 0 表示不成功。

**SF:**为端口的状态标志, 32 位数据, 小端模式。在 PIN 取值为 0x00~0x03 时, 非 0 表示 1, 反之为 0。当 PIN 取值为 0xF0 时, bit[0:3]对应线路 1 至线路 4 的输入状态。

**使用例子:** 读取地址为 0 的显示屏的所有端口状态。

主机发送: 00 64 FF FF 10 01 **F0** 37 35

设备回复: 00 64 FF FF 10 05 **00 0F 00 00 00** 5C FF

### 0x11(配置 LED 驱动程序)

显示卡支持 2 种接口的 LED 显示屏, 分别位 HUB08 和 HUB12。HUB08 采用 16 分之一扫描。HUB12 采用 4 分之一扫描方式。

**请求格式:**DA + SP + PN[2] + 0x11+ DL + **MDI** + CRC[2]

**请求参数描述:**包含 1 个字节的参数, DL 取值为 1。

**MDI:**为驱动类型, 取值范围见下表

MDI 取值含义	
取值	描述
0x00	HUB08 扫描接口
0x01	HUB12 扫描接口

**回复格式:** DA + SP + PN[2] + 0x11+ DL + **ACK** + CRC[2]

**回复参数描述:**包含 1 个字节的参数, DL 取值为 1, **ACK** 是显示屏返回的结果, 取值为 0 表示成功, 非 0 表示不成功。

**使用例子:** 将地址为 0 的显示屏的驱动接口配置为 HUB08。

主机发送: 00 64 FF FF 11 01 00 66 B1

设备回复: 00 64 FF FF 11 01 00 66 B1

### 0x12(设置 OE 极性)

OE 是用来使能 LED 模组的信号, 亮度调节一般都是在 OE 上调制 PWM 信号来实现。各 LED 模组生产商的 OE 信号的极性不同, 表现的现象一般是亮度调到最大, 反而亮度最小, 如果是这样, 那么就需要将 OE 的极性设置成负极性。

**请求格式:**DA + SP + PN[2] + 0x12+ DL + **OEPOL** + CRC[2]

**请求参数描述:**包含 1 个字节的参数, DL 取值为 1。

**OEPOL:**为 OE 信号的极性, 取值范围见下表

OEPOL 取值含义	
取值	描述
0x00	正极性
0x01	负极性

**回复格式:** DA + SP + PN[2] + 0x12+ DL + **ACK** + CRC[2]

**回复参数描述:**包含 1 个字节的参数, DL 取值为 1, **ACK** 是显示屏返回的结果, 取值为 0 表示成功, 非 0 表示不成功。

使用例子：将地址为 0 的显示屏的 OE 信号极性配置为正极性。

主机发送:00 64 FF FF 12 01 00 96 B1

设备回复:00 64 FF FF 12 01 00 96 B1

### 0x13(设置数据极性)

各 LED 模组生产商使用的芯片不同，导致点阵的驱动方式也不一致。从驱动极性上分为正极性驱动和负极性驱动。如果显示屏上显示的内容暗亮是反的，那么就需要将数据极性配置为负极性。

请求格式:DA + SP + PN[2] + 0x13+ DL + DPOL+ CRC[2]

请求参数描述:包含 1 个字节的参数，DL 取值为 1。

DPOL:为驱动类型，取值范围见下表

DPOL 取值含义	
取值	描述
0x00	正极性
0x01	负极性

回复格式: DA + SP + PN[2] + 0x13+ DL + ACK + CRC[2]

回复参数描述:包含 1 个字节的参数，DL 取值为 1，ACK 是显示屏返回的结果，取值为 0 表示成功，非 0 表示不成功。

使用例子：将地址为 0 的显示屏的数据极性配置为正极性。

主机发送: 00 64 FF FF 13 01 00 C7 71

设备回复: 00 64 FF FF 13 01 00 C7 71

### 0x14(设置 LE 信号延时)

LE 为数据加载信号。如果显示屏出现第一列或者最后一列没有显示，那么就需要打开 LE 延时。

请求格式:DA + SP + PN[2] + 0x14+ DL + LE+ CRC[2]

请求参数描述:包含 1 个字节的参数，DL 取值为 1。

LE:为加载信号延时开关，取值范围见下表

LE 取值含义	
取值	描述
0x00	关闭
0x01	打开

回复格式: DA + SP + PN[2] + 0x14+ DL + ACK + CRC[2]

回复参数描述:包含 1 个字节的参数，DL 取值为 1，ACK 是显示屏返回的结果，取值为 0 表示成功，非 0 表示不成功。

使用例子：将地址为 0 的显示屏的数据极性配置为正极性。

主机发送: 00 64 FF FF 14 01 00 76 B0

设备回复: 00 64 FF FF 14 01 00 76 B0

## 0x15(设置网关和掩码)

如果是用网络通信，在通信之前需要按照实际的网络环境配置网关和掩码。

**请求格式:**DA + SP + PN[2] + 0x15+ DL + GWIP[4] + MASK[4] + CRC[2]

**请求参数描述:**包含 8 个字节的参数，DL 取值为 8。

**GWIP:**为网关的 IP 地址，包含 4 个字节。

**MASK:**为掩码，包含 4 个字节。

**回复格式:**DA + SP + PN[2] + 0x15+ DL + ACK + CRC[2]

**回复参数描述:**包含 1 个字节的参数，DL 取值为 1，ACK 是显示屏返回的结果，取值为 0 表示成功，非 0 表示不成功。

**使用例子:** 将地址为 0 的显示屏的网关地址为 192.168.0.1 掩码为 255.255.255.0。

**主机发送:** 00 64 FF FF 15 08 C0 A8 00 01 FF FF FF 00 6F EC

**设备回复:** 00 64 FF FF 15 01 00 27 70

## 0x16(进入安全模式)

安全模式为维护使用。进入安全模式后，显示卡只运行了 DUCP 通信服务线程。客户端可以没有限制的使用任何命令。例如对文件的写访问等等。退出安全模式需要重启显示卡。

**请求格式:**DA + SP + PN[2] + 0x16+ DL + PSWD[6] + CRC[2]

**请求参数描述:**包含 6 个字节的参数，DL 取值为 6。

**PSWD:**为 6 位登陆密码。

**回复格式:**DA + SP + PN[2] + 0x16+ DL + ACK + CRC[2]

**回复参数描述:**包含 1 个字节的参数，DL 取值为 1，ACK 是显示屏返回的结果，取值为 0 表示成功，非 0 表示不成功。

**使用例子:** 地址为 0 的显示屏的进入安全模式。

**主机发送:** 00 64 FF FF 16 06 72 6F 6F 74 00 00 9B C7

**设备回复:** 00 64 FF FF 16 01 00 D7 70

## 0x18(设置显示尺寸)

显示卡默认配置为最大的显示尺寸，但如果在使用了显示镜像功能后，由于原点坐标在原来参考方向的右下角，使得原点坐标跟拼接的 LED 模组数量有关，因此需要用户重新配置显示尺寸，否则可能用户看不到显示内容(原点在实际尺寸之外)。

**请求格式:**DA + SP + PN[2] + 0x18+ DL + XSIZE[2] + YSIZE[2] + CRC[2]

**请求参数描述:**包含 4 个字节的参数，DL 取值为 4。

**XSIZE:**为横轴的尺寸，16 位数据类型，小端模式。

**YSIZE:**为横轴的尺寸，16 位数据类型，小端模式。

**回复格式:**DA + SP + PN[2] + 0x18+ DL + ACK + CRC[2]

**回复参数描述:**包含 1 个字节的参数，DL 取值为 1，ACK 是显示屏返回的结果，取值为 0 表示成功，非 0 表示不成功。

使用例子: 地址为 0 的显示屏显示尺寸设置为 256\*128。

主机发送: 00 64 FF FF 18 04 00 01 80 00 68 D6

设备回复: 00 64 FF FF 18 01 00 B6 B3

### 0x19(设置显示方向)

显示卡支持 4 个方位的旋转显示, 支持正常、顺时针 90 度、逆时针 90 度、180 度, 这使得安装工作变得非常便利。需要注意的是, 旋转了方位之后, 需要按照实际的物理显示尺寸设置显示尺寸。

请求格式: DA + SP + PN[2] + 0x19+ DL + DIR+ CRC[2]

请求参数描述: 包含 1 个字节的参数, DL 取值为 1。

DIR: 显示方向, 为 0 时表示正常显示, 为 1 时为旋转 180 度, 为 2 时表示顺时针 90 度, 为 3 时表示逆时针 90 度。

回复格式: DA + SP + PN[2] + 0x19+ DL + ACK + CRC[2]

回复参数描述: 包含 1 个字节的参数, DL 取值为 1, ACK 是显示屏返回的结果, 取值为 0 表示成功, 非 0 表示不成功。

使用例子: 设置地址为 0 的显示屏显示方向为正常显示。

主机发送: 00 64 FF FF 19 01 00 E7 73

设备回复: 00 64 FF FF 19 01 00 E7 73

### 0x1A(设置 485 工作模式)

在有些时候可能不需要显示卡回复, 此时可以通过该指令来设置显示卡的 485 工作模式为半工模式。

请求格式: DA + SP + PN[2] + 0x1A+ DL + MODE+ CRC[2]

请求参数描述: 包含 1 个字节的参数, DL 取值为 1。

MODE: 为 0 时表示半双工模式, 为 1 时表示半工模式。

回复格式: DA + SP + PN[2] + 0x1A+ DL + ACK + CRC[2]

回复参数描述: 包含 1 个字节的参数, DL 取值为 1, ACK 是显示屏返回的结果, 取值为 0 表示成功, 非 0 表示不成功。

使用例子: 设置地址为 0 的显示屏 485 模式为半工模式。

主机发送: 00 64 FF FF 1A 01 01 D6 B3

设备回复: 设备工作在半工模式, 不会返回数据。

### 0x1E(恢复出厂设置)

该指令用来将显示屏设置为出厂设置。

**请求格式:**DA + SP + PN[2] + 0x1E+ DL + CRC[2]

**请求参数描述:**包含 0 个字节的参数，DL 取值为 0。

**回复格式:** DA + SP + PN[2] + 0x1E+ DL + ACK + CRC[2]

**回复参数描述:**包含 1 个字节的参数，DL 取值为 1，ACK 是显示屏返回的结果，取值为 0 表示成功，非 0 表示不成功。

**使用例子:** 将地址为 0 的显示屏恢复出厂设置。

**主机发送:**00 64 FF FF 1E 00 78 57

**设备回复:**00 64 FF FF 1E 01 00 56 B2

### 0x30(播放语音)

需要播报语音的文字通过该指令发送到显示屏，显示屏收到之后会与内置的语音库匹配，如果存在相应的语音文件，就会播报语音。显示屏的语音匹配方式是按照词组或短语匹配的，在发送文字时，需要用逗号或者句号分隔每个词组或短语。例如"欢迎光临,请入场停车"。

**请求格式:**DA + SP + PN[2] + 0x30+ DL + OPT + TEXT[MAX 254]+ CRC[2]

**请求参数描述:**参数长度 DL 为 1 加上文字长度。

**OPT:**为操作选项字。

OPT 取值含义	
取值	描述
0x00	添加到语音队列但是不开始播放
0x01	添加到语音队列并且开始播放
0x02	先清除队列，再添加新语音到队列，然后开始播放

**TEXT:**为播报语音的文字，最大的长度为 254 个字节。文字编码格式只支持 ASCII 和 GBK2312。**注意不支持 UNICODE。**

**回复格式:** DA + SP + PN[2] + 0x30+ DL + ACK + CRC[2]

**回复参数描述:**包含 1 个字节的参数，DL 取值为 1，ACK 是显示屏返回的结果，取值为 0 表示成功，非 0 表示不成功。

**使用例子 1:** 地址为 0 的显示屏立即播放"欢迎光临,请入场停车"。

**主机发送:**00 64 FF FF 30 14 01 BB B6 D3 AD B9 E2 C1 D9 2C C7 EB C8 EB B3 A1 CD A3 B3 B5 A9 36 设

**设备回复:**00 64 FF FF 30 01 00 36 BB

**使用例子 2:** 向地址为 0 的显示屏添加"欢迎光临,请入场停车"不立即播放语音，只添加到语音队列。然后再添加"请交费 10 元" 并且开始播放语音。

**#1 主机发送:**00 64 FF FF 30 14 00 BB B6 D3 AD B9 E2 C1 D9 2C C7 EB C8 EB B3 A1 CD A3 B3 B5 94 E7

**#1 设备回复:**00 64 FF FF 30 01 00 36 BB

**#2 主机发送:**00 64 FF FF 30 0B 01 C7 EB BD BB B7 D1 31 30 D4 AA 43 1E

**#2 设备回复:** 00 64 FF FF 30 01 00 36 BB

### 0x31(停止播放语音)

通过这个指令可以立即停止显示屏播报语音，并且清空语音队列。

**请求格式:**DA + SP + PN[2] + 0x31+ DL+ CRC[2]

**请求参数描述:**无参数，DL 取值为 0。

**回复格式:** DA + SP + PN[2] + 0x31+ DL + ACK + CRC[2]

**回复参数描述:**包含 1 个字节的参数，DL 取值为 1，ACK 是显示屏返回的结果，取值为 0 表示成功，非 0 表示不成功。

**使用例子:** 将地址为 0 的显示屏的亮度更改为 50%。

**主机发送:**00 64 FF FF 31 00 64 67

**设备回复:**00 64 FF FF 31 01 00 67 7B

### 0x38(画点)

画点是最基本的图形绘制接口，可以在屏幕的任何位置画点，如果支持多色的显示屏，可以在画点之前设置画笔颜色。

**请求格式:**DA + SP + PN[2] + 0x38+DL+X0[2] + Y0[2]+ CRC[2]

**请求参数描述:**包含 4 个字节的参数，DL 取值为 4。

**X0:**为屏幕的横轴坐标，16 位数据类型，小端模式。

**Y0:**为屏幕的纵轴坐标，16 位数据类型，小端模式。

**回复格式:** DA + SP + PN[2] + 0x38+ DL + ACK + CRC[2]

**回复参数描述:**包含 1 个字节的参数，DL 取值为 1，ACK 是显示屏返回的结果，取值为 0 表示成功，非 0 表示不成功。

**使用例子:** 在地址为 0 的显示屏上坐标为(0,0)的位置画一个点。

**主机发送:**00 64 FF FF 38 04 00 00 00 00 5F B6

**设备回复:**00 64 FF FF 38 01 00 B7 79

### 0x39(画线)

画线接口可以在屏幕的任何位置画一条任意角度的直线，如果支持多色的显示屏，可以在画线之前设置画笔颜色。

**请求格式:**DA + SP + PN[2] + 0x39+DL+X0[2] + Y0[2] + X1[2] + Y1[2]+ CRC[2]

**请求参数描述:**包含 8 个字节的参数，DL 取值为 8。

**X0:**为直线的起始点横轴坐标，16 位数据类型，小端模式。

**Y0:**为直线的起始点纵轴坐标，16 位数据类型，小端模式。

**X1:**为直线的结束点横轴坐标，16 位数据类型，小端模式。

**Y1:**为直线的结束点纵轴坐标，16 位数据类型，小端模式。

**回复格式:** DA + SP + PN[2] + 0x39+ DL + ACK + CRC[2]

**回复参数描述:**包含 1 个字节的参数，DL 取值为 1，ACK 是显示屏返回的结果，取值为 0 表示成功，非 0 表示不成功。

**使用例子:** 在地址为 0 的显示屏上画一条从(0,0)到(63,31)的直线。

**主机发送:**00 64 FF FF 39 08 00 00 00 00 3F 00 1F 00 F3 13

**设备回复:**00 64 FF FF 39 01 00 E6 B9



## 0x3A(画矩形)

画矩形接口可以在屏幕的任何位置画一个任意大小四边框，如果支持多色的显示屏，可以在绘图之前设置画笔颜色。

**请求格式:**DA + SP + PN[2] + 0x3A+DL+X0[2] + Y0[2] + X1[2] + Y1[2] + CRC[2]

**请求参数描述:**包含 8 个字节的参数，DL 取值为 8。

**X0:**为矩形的起始点横轴坐标，16 位数据类型，小端模式。

**Y0:**为矩形的起始点纵轴坐标，16 位数据类型，小端模式。

**X1:**为矩形的对角点横轴坐标，16 位数据类型，小端模式。

**Y1:**为矩形的对角点纵轴坐标，16 位数据类型，小端模式。

**回复格式:** DA + SP + PN[2] + 0x3A+ DL + ACK + CRC[2]

**回复参数描述:**包含 1 个字节的参数，DL 取值为 1，ACK 是显示屏返回的结果，取值为 0 表示成功，非 0 表示不成功。

**使用例子:** 在地址为 0 的显示屏上画一个起始点(0,0)对角点(63,31)的矩形边框。

**主机发送:**00 64 FF FF 3A 08 00 00 00 00 3F 00 1F 00 03 1C

**设备回复:**00 64 FF FF 3A 01 00 16 B9

## 0x3B(填充矩形)

填充矩形接口可以在屏幕的任何位置填充一个任意大小的矩形，如果支持多色的显示屏，可以在绘图之前设置画笔颜色。

**请求格式:**DA + SP + PN[2] + 0x3B+ DL+X0[2] + Y0[2] + X1[2] + Y1[2]+ CRC[2]

**请求参数描述:**包含 8 个字节的参数，DL 取值为 8。

**X0:**为矩形的起始点横轴坐标，16 位数据类型，小端模式。

**Y0:**为矩形的起始点纵轴坐标，16 位数据类型，小端模式。

**X1:**为矩形的对角点横轴坐标，16 位数据类型，小端模式。

**Y1:**为矩形的对角点纵轴坐标，16 位数据类型，小端模式。

**回复格式:** DA + SP + PN[2] + 0x3B+ DL + ACK + CRC[2]

**回复参数描述:**包含 1 个字节的参数，DL 取值为 1，ACK 是显示屏返回的结果，取值为 0 表示成功，非 0 表示不成功。

**使用例子:** 在地址为 0 的显示屏上填充一个起始点(0,0)对角点(63,31)的矩形。

**主机发送:**00 64 FF FF 3B 08 00 00 00 00 3F 00 1F 00 52 D9

**设备回复:**00 64 FF FF 3B 01 00 47 79

## 0x3C(画圆)

画圆接口可以在屏幕的任何位置画一个任意直径的圆形，如果支持多色的显示屏，可以在绘图之前设置画笔颜色。

**请求格式:**DA + SP + PN[2] + 0x3C+ DL+X0[2] + Y0[2] + R[2]+ CRC[2]

**请求参数描述:**包含 6 个字节的参数，DL 取值为 6。

**X0:**为圆心横轴坐标，16 位数据类型，小端模式。

**Y0:**为圆心纵轴坐标，16 位数据类型，小端模式。

**R:**为圆的直径，16 位数据类型，小端模式。

**回复格式:** DA + SP + PN[2] + 0x3C+ DL + ACK + CRC[2]

**回复参数描述:**包含 1 个字节的参数，DL 取值为 1，ACK 是显示屏返回的结果，取值为 0 表示成功，非 0 表示不成功。

**使用例子:** 在地址为 0 的显示屏上画一个圆点(32,16)直径为 10 的圆形。

**主机发送:** 00 64 FF FF 3C 06 1F 00 0F 00 0A 00 9D AE

**设备回复:** 00 64 FF FF 3C 01 00 F6 B8

## 0x3D(填充圆)

填充圆接口可以在屏幕的任何位置填充一个任意直径的圆形，如果支持多色的显示屏，可以在绘图之前设置画笔颜色。

**请求格式:**DA + SP + PN[2] + 0x3D+ DL + X0[2] + Y0[2] + R[2]+ CRC[2]

**请求参数描述:**包含 6 个字节的参数，DL 取值为 6。

**X0:**为圆心横轴坐标，16 位数据类型，小端模式。

**Y0:**为圆心纵轴坐标，16 位数据类型，小端模式。

**R:**为圆的直径，16 位数据类型，小端模式。

**回复格式:** DA + SP + PN[2] + 0x3D+ DL + ACK + CRC[2]

**回复参数描述:**包含 1 个字节的参数，DL 取值为 1，ACK 是显示屏返回的结果，取值为 0 表示成功，非 0 表示不成功。

**使用例子:** 在地址为 0 的显示屏上画一个圆点(32,16)直径为 10 的圆形。

**主机发送:** 00 64 FF FF 3D 06 1F 00 0F 00 0A 00 5C 62

**设备回复:** 00 64 FF FF 3D 01 00 A7 78

## 0x3E(绘制位图)

绘制位图接口可以在屏幕的任何位置绘制一张自定义的图片。位图的颜色深度需要与显示屏颜色深度一致，否则绘制出来的图片会丢失色彩。

**请求格式:** DA+SP+PN[2]+0x3E+DL+X0[2]+Y0[2]+XSIZE[2]+YSIZE[2]+BPP[2]+BPL[2]+CTS[2]+FSIZE[4]+DATA[...] +CRC[2]

**请求参数描述:**DL 取值为 18 + 数据长度。

**X0:**为绘图的起始点横轴坐标，16 位数据类型，小端模式。

**Y0:**为绘图的起始点纵轴坐标，16 位数据类型，小端模式。

**XSIZE:**为位图的横向长度，16 位数据类型，小端模式。

**YSIZE:**为位图的纵向宽度，16 位数据类型，小端模式。

**BPP:**为位图像素所占用的位数，16 位数据类型，小端模式。

BPP 的取值为 1、2、16、24。

**BPL:**为位图每行所占的字节数，16 位数据类型，小端模式。

**CTS:**为位图的颜色表的大小，16 位数据类型，小端模式。

**FSIZE:**为位图文件的总大小，32 位数据类型，小端模式。


**CT:**为位图的颜色表，如果 CTS 为 0，则不存在颜色表字段。

颜色表的存储格式为 RGBA，每个颜色值占用 4 个字节。颜色的数量必须等于  $2^{BPP}$  次方。当 BPP 等于 24 时，不需要颜色表。BPP 等于 16 位的 565 格式时，也不需要颜色表。

**DATA:**为位图的像素数据，单包最大发送的数据为 255。大数据位图文件需要分包传输。

**回复格式:** DA + SP + PN[2] + 0x3E+ DL + ACK + CRC[2]

**回复参数描述:**包含 1 个字节的参数，DL 取值为 1，ACK 是显示屏返回的结果，取值为 0 表示成功，非 0 表示不成功。

**使用例子:** 在地址为 0 的显示屏上绘制一张单色 32X32 位图 .

**主机发送:** 00 64 FF FF 3E 9A 00 00 00 00 20 00 20 00 01 00 04 00 08 00 88 00 00 00 00 00 00 FF FF  
 FF 00 00 00 00 00 00 00 00 00 3F F0 00 00 3F F8 00 00 3F F8 00 00 38 3C 00 00 38 1C 00 00 38 1C 00  
 00 38 1C 00 00 38 1C 00 00 38 1C 00 00 38 3C 00 00 3F F8 00 00 3F F0 00 00 3F E0 00 00 3F 80 00 00  
 38 00 00 00 38 00 00 00 38 00 00 00 38 00 00 00 38 00 FF F8 38 01 83 1C 38 03 03 1E 38 06 03 1F 38  
 7F FF FF 38 FF FF FF 39 FF FF FF 39 FF FF FF 38 F3 FF CE 38 12 00 48 38 0C 00 30 00 00 00 00 D8 42

**设备回复:** 00 64 FF FF 3E 01 00 57 78

## 0x3F(显示文字)

显示文字接口可以在屏幕的任何位置显示文字，支持 GBK2312 和 ASCII 编码的字符集。在使之前可以先通过设置字体和颜色，请参见设置字体和设置颜色指令。**注意:在有窗口显示的区域，发送的文字将不会被显示；这是因为处于显示的窗口在窗口管理器的最顶层，称为 Z 序。如果一定要在该区域显示，必须先删除或隐藏该窗口。**

**请求格式:**DA + SP + PN[2] + 0x3F+ DL + X0[2] + Y0[2] + TEXT[MAX 251]+ CRC[2]

**请求参数描述:**DL 等于 4 + 文字长度。

**X0:**为显示的起始横轴坐标，16 位数据类型，小端模式。

**Y0:**为显示的起始纵轴坐标，16 位数据类型，小端模式。

**TEXT:**为需要显示的文字，最大长度为 251 个字节。

**回复格式:** DA + SP + PN[2] + 0x3F+ DL + ACK + CRC[2]

**回复参数描述:**包含 1 个字节的参数，DL 取值为 1，ACK 是显示屏返回的结果，取值为 0 表示成功，非 0 表示不成功。

**使用例子:** 在地址为 0 的显示屏的(64,0)位置显示"Hello world!"。

**主机发送:**00 64 FF FF 3F 10 40 00 00 00 48 65 6C 6C 6F 20 77 6F 72 6C 64 21 EA 28

**设备回复:**00 64 FF FF 3F 01 00 06 B8

## 0x40(设置画笔/文字颜色)

设置画笔/文字颜色接口常与绘图和显示文字一起使用，仅在支持彩色的显示屏中才可以在绘图或者显示文字之前改变画笔的颜色。

**请求格式:**DA + SP + PN[2] + 0x40+ DL + TC[4]+ CRC[2]

**请求参数描述:**包含 4 个字节的参数，DL 取值为 4。

**TC:**为 RGB 颜色值，存储结构为 R G B A 三基色，各占 8 位，R 表示红色分量，G 表示绿色分量，B 表示蓝色分量，A 目前没用使用，作为保留字。各取值范围为 0~255。

**回复格式:** DA + SP + PN[2] + 0x40+ DL + ACK + CRC[2]

**回复参数描述:**包含 1 个字节的参数，DL 取值为 1，ACK 是显示屏返回的结果，取值为 0 表示成功，非 0 表示不成功。

**使用例子:** 地址为 0 的显示屏设置绘图前景色为红色。

**主机发送:**00 64 FF FF 40 04 FF 00 00 00 65 DA

**设备回复:**00 64 FF FF 40 01 00 37 60

## 0x41(设置背景颜色)

设置背景颜色接口常与绘图和显示文字一起使用，仅在支持彩色的显示屏中才可以在绘图或者显示文字之前改变背景的颜色。

**请求格式:**DA + SP + PN[2] + 0x41+ DL + **BC[4]** + CRC[2]

**请求参数描述:**包含 4 个字节的参数，DL 取值为 4。

**BC:**为 RGB 颜色值，存储结构为 RGBA 三基色，各占 8 位，R 表示红色分量，G 表示绿色分量,B 表示蓝色分量，A 目前没用使用，作为保留字。各取值范围为 0~255。

**回复格式:** DA + SP + PN[2] + 0x41+ DL + **ACK** + CRC[2]

**回复参数描述:**包含 1 个字节的参数，DL 取值为 1，ACK 是显示屏返回的结果，取值为 0 表示成功，非 0 表示不成功。

**使用例子:** 地址为 0 的显示屏设置绘图背景色为黑色。

**主机发送:** 00 64 FF FF 41 04 **00 00 00 00** 54 1F

**设备回复:** 00 64 FF FF 41 01 **00** 66 A0

## 0x42(设置字体)

设置字体接口常与显示文字一起使用，目前支持的字体有 6 种分别为 ASCII8、ASCII10、ASCII13、宋体 16、宋体 24、宋体 32。

**请求格式:**DA + SP + PN[2] + 0x42+ DL + **FINDEX** + CRC[2]

**请求参数描述:**包含 1 个字节的参数，DL 取值为 1。

**FINDEX:**为字体的索引值，取值含义如下:

FINDEX 取值含义	
取值	描述
0x00	ASCII8
0x01	ASCII10
0x02	ASCII13
0x03	宋体 16
0x04	宋体 24
0x05	宋体 32

**回复格式:** DA + SP + PN[2] + 0x42+ DL + **ACK** + CRC[2]

**回复参数描述:**包含 1 个字节的参数，DL 取值为 1，ACK 是显示屏返回的结果，取值为 0 表示成功，非 0 表示不成功。

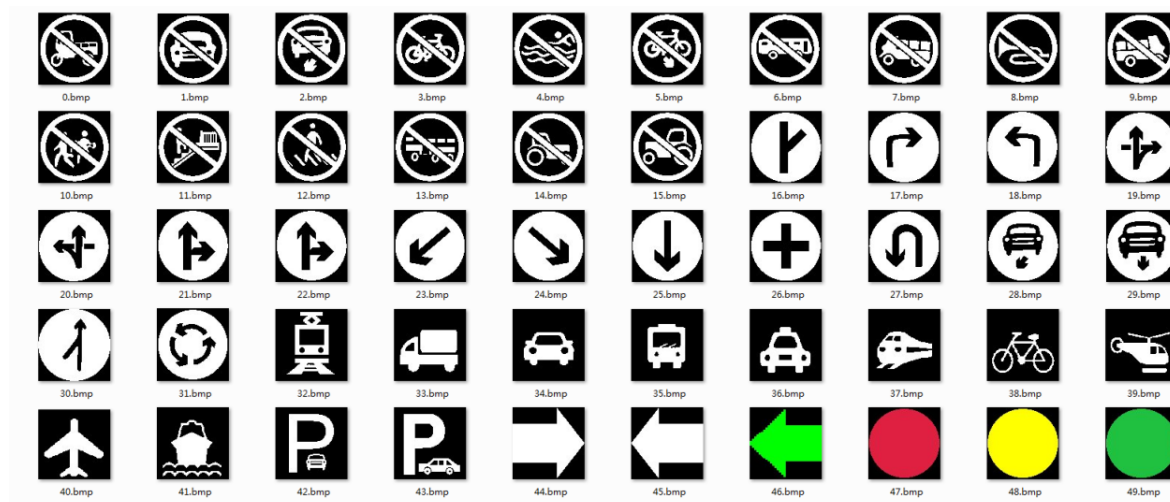
**使用例子:** 地址为 0 的显示屏设置字体为宋体 16。

**主机发送:** 00 64 FF FF 42 01 **03** D6 A1

**设备回复:** 00 64 FF FF 42 01 **00** 96 A0

## 0x43(绘制系统自带的位图)

显示屏系统中自带了一些常用的图标，用户可以通过这个接口直接调出显示。系统自带的位图尺寸为 96\*96 像素，并且支持缩放显示。彩色位图只能在彩色屏上显示。



请求格式:DA+SP+PN[2]+0x3E+DL+X0[2]+Y0[2]+NUM[2]+DENOM[2]+ICON[2]+CRC[2]

请求参数描述:DL 取值为 18 + 数据长度。

**X0:**为绘图的起始点横轴坐标，16 位数据类型，小端模式。

**Y0:**为绘图的起始点纵轴坐标，16 位数据类型，小端模式。


**NUM:**为位图的缩放分子，不能小于等于 0，16 位数据类型，小端模式。

**DENOM:**为位图的缩放分母，不能小于等于 0，16 位数据类型，小端模式。

**ICON:**为位图索引值，可能的取值参见上图的文件名，16 位数据类型，小端模式。

回复格式: DA + SP + PN[2] + 0x3E + DL + ACK + CRC[2]

回复参数描述:包含 1 个字节的参数，DL 取值为 1，ACK 是显示屏返回的结果，取值为 0 表示成功，非 0 表示不成功。

使用例子:在地址为 0 的显示屏上绘制文件名为 42.BMP  位图,显示的比例设置为原图的 1/2。

主机发送:00 64 FF FF 43 0A 00 00 00 00 01 00 02 00 2A 00 4A 30

设备回复:00 64 FF FF 43 01 00 C7 60

## 0x50(显示窗口)

显示窗口可以用来将一个或者所有的窗口设置为可见的状态。如果窗口不存在，显示屏将会忽略本操作。本指令对所有的控件也有效，因为控件都隶属于窗口类。

请求格式:DA + SP + PN[2] + 0x50 + DL + WID[2] + CRC[2]

请求参数描述:包含 2 个字节的参数，DL 取值为 2。

**WID:**为窗口 ID，16 位数据类型，小端模式。当取值为 0XFFFF 时表示对所有窗口进行操作。

回复格式: DA + SP + PN[2] + 0x50 + DL + ACK + CRC[2]

**回复参数描述:**包含 1 个字节的参数, DL 取值为 1, ACK 是显示屏返回的结果, 取值为 0 表示成功, 非 0 表示不成功。

**使用例子:** 显示地址为 0 的显示屏的 0 号窗口。

**主机发送:** 00 64 FF FF 50 02 00 00 D5 16

**设备回复:** 00 64 FF FF 50 01 00 36 A5

## 0x51(隐藏窗口)

隐藏窗口可以用来将一个或者所有的窗口设置为不可见的状态。如果窗口不存在, 显示屏将会忽略本操作。本指令对所有的控件也有效, 因为控件都隶属于窗口类。

**请求格式:** DA + SP + PN[2] + 0x51+ DL + WID[2]+ CRC[2]

**请求参数描述:**包含 2 个字节的参数, DL 取值为 2。

**WID:**为窗口 ID, 16 位数据类型, 小端模式。当取值为 0xFFFF 时表示对所有窗口进行操作。

**回复格式:** DA + SP + PN[2] + 0x51+ DL + ACK + CRC[2]

**回复参数描述:**包含 1 个字节的参数, DL 取值为 1, ACK 是显示屏返回的结果, 取值为 0 表示成功, 非 0 表示不成功。

**使用例子:** 显示地址为 0 的显示屏的 0 号窗口。

**主机发送:** 00 64 FF FF 51 02 00 00 D4 EA

**设备回复:** 00 64 FF FF 51 01 00 67 65

## 0x52(删除窗口)

删除窗口可以用来将一个或者所有的窗口从窗口管理器中移除。如果窗口不存在, 显示屏将会忽略本操作。删除的窗口在之后再不能被操作。本指令对所有的控件也有效, 因为控件都隶属于窗口类。

**请求格式:** DA + SP + PN[2] + 0x52+ DL + WID[2]+ CRC[2]

**请求参数描述:**包含 2 个字节的参数, DL 取值为 2。

**WID:**为窗口 ID, 16 位数据类型, 小端模式。当取值为 0xFFFF 时表示对所有窗口进行操作。

**回复格式:** DA + SP + PN[2] + 0x52+ DL + ACK + CRC[2]

**回复参数描述:**包含 1 个字节的参数, DL 取值为 1, ACK 是显示屏返回的结果, 取值为 0 表示成功, 非 0 表示不成功。

**使用例子:** 显示地址为 0 的显示屏的 0 号窗口。

**主机发送:** 00 64 FF FF 52 02 00 00 D4 AE

**设备回复:** 00 64 FF FF 52 01 00 97 65

## 0x53(自动同步窗口)

同步窗口的功能是所有文本窗口的开始显示的时间是相同的，结束时间根据每个窗口的文字长度会不相同。提前显示完的窗口会等待所有的文本窗口都显示完成然后一起开始。

**请求格式:**DA + SP + PN[2] + 0x53+ DL + EN+ CRC[2]

**请求参数描述:**包含 1 个字节的参数，DL 取值为 1。

**EN:**为 1 时开启同步功能，为 0 时关闭同步功能。

**回复格式:** DA + SP + PN[2] + 0x53+ DL + ACK + CRC[2]

**回复参数描述:**包含 1 个字节的参数，DL 取值为 1，ACK 是显示屏返回的结果，取值为 0 表示成功，非 0 表示不成功。

**使用例子:** 使能显示地址为 0 的显示屏窗口同步功能。

**主机发送:** 00 64 FF FF 53 01 01 07 65

**设备回复:** 00 64 FF FF 53 01 00 C6 A5

## 0x60(创建文本控件)

创建文本控件的功能是在显示屏的任何位置创建一个任意大小的窗口，用于显示文字信息。在请求文本控件的其它操作之前，必须先创建了一个文本控件。

**请求格式:**DA + SP + PN[2] + 0x60+ DL + TWID + STILE + X0[2] + Y0[2] + XSIZE[2] + YSIZE[2] + CRC[2]

**请求参数描述:**包含 10 个字节的参数，DL 取值为 10。

**TWID:**为文本控件的 ID,用于标识创建的窗口身份。

**STILE:**为文本控件的风格类型，目前没有用到，取值固定为 0。

**X0:**为文本控件的起始点横轴坐标，16 位数据类型，小端模式。

**Y0:**为文本控件的起始点纵轴坐标，16 位数据类型，小端模式。

**XSIZE:**为文本控件的横向长度，16 位数据类型，小端模式。

**YSIZE:**为文本控件的纵向宽度，16 位数据类型，小端模式。

**回复格式:** DA + SP + PN[2] + 0x60+ DL + ACK + CRC[2]

**回复参数描述:**包含 1 个字节的参数，DL 取值为 1，ACK 是显示屏返回的结果，取值为 0 表示成功，非 0 表示不成功。

**使用例子:** 在地址为 0 的显示屏的(0,0)位置创建一个大小为 64\*16 的文本控件。

**主机发送:** 00 64 FF FF 60 0A 00 00 00 00 00 00 40 00 10 00 C8 82

**设备回复:** 00 64 FF FF 60 01 00 36 AA



### 0x61(删除文本控件)

删除文本控件的功能是从窗口管理器中移除已经被创建的文本控件。被删除的文本控件不能再被操作。

**请求格式:**DA + SP + PN[2] + 0x61+ DL + TWID + CRC[2]

**请求参数描述:**包含 10 个字节的参数，DL 取值为 10。

**TWID:**为文本控件 ID,用于标识创建的窗口身份。

**回复格式:** DA + SP + PN[2] + 0x61+ DL + ACK + CRC[2]

**回复参数描述:**包含 1 个字节的参数，DL 取值为 1，ACK 是显示屏返回的结果，取值为 0 表示成功，非 0 表示不成功。

**使用例子:** 删除地址为 0 的显示屏的 0 号文本控件。

**主机发送:** 00 64 FF FF 61 01 00 67 6A

**设备回复:** 00 64 FF FF 61 01 00 67 6A

### 0x62(设置文本)

设置文本的功能是配置文本控件的显示参数，包括文字内容，入场方式、停留方式、退场方式等。

**请求格式:**DA + SP + PN[2] + 0x62+ DL + TWID + ETM + ETS + DM + DT + EXM + EXS + FINDEX + DRS + TC[4] + BC[4] + TL[2] + TEXT[...] + CRC[2]

**请求参数描述:**DL 等于 19 个字节再加上文本长度。

**TWID:**为文本控件 ID,用于标识创建的窗口身份。

**ETM:**文字进入窗口的方式。取值范围及含义见下表:

ETM 取值含义	
编码	描述
0x00	立即显示
0x01	从右向左移动
0x02	从左向右移动
0x03	从下向上移动
0x04	从上向下移动
0x05	向下拉窗
0x06	向上拉窗
0x07	向左拉窗
0x08	向右拉窗
0x09	反亮显示
0x0A	无操作
0x0B	中速左移动
0x0C	中速右移动
0x0D	逐字显示
0x0E	向左开栅
0x0F	向右开栅
0x10	雪花显示

0x11	隔行出现
0x12	隔列出现
0x13	慢速左移
0x14	慢速右移
0x15	连续左移
0x16	中速连续左移
0x17	慢速连续左移

**ETS:**为文字进入的速度。取值范围为 1~32。时基取决于当前的扫描周期。

**DM:**为文字停留的方式。取值范围及含义见下表:

DM 取值含义	
编码	描述
0x00	停留 秒单位
0x01	停留 1/10 秒单位
0x02	停留 100*秒单位
0x03	快速闪烁
0x04	中速闪烁
0x05	慢速闪烁
0x06	单次闪烁
0x07	英文日期
0x08	英文时间
0x09	中文日期
0x0A	中文时间
0x0B	控制
0x0C	循环
0x0D	温度
0x0E	声音

**DT:**为文字停留的时间。取值范围为 0~255。

**EXM:**为文字退场的方式。取值范围及含义见下表:

EXM 取值含义	
编码	描述
0x00	立即退出
0x01	从右向左移动
0x02	从左向右移动
0x03	从下向上移动
0x04	从上向下移动
0x05	向下拉窗
0x06	向上拉窗
0x07	向左拉窗
0x08	向右拉窗
0x09	反亮显示
0x0A	无操作
0x0B	中速左移动
0x0C	中速右移动

0x0D	逐字显示
0x0E	向左开栅
0x0F	向右开栅
0x10	雪花显示
0x11	隔行退出
0x12	隔列退出
0x13	慢速左移
0x14	慢速右移
0x15	连续左移
0x16	中速连续左移
0x17	慢速连续左移

**EXS:**为文字退场的速度。取值范围为 1~32。时基取决于当前的扫描周期。

**FINDEX:**为文字的字体索引值。取值范围及含义见下表:

FINDEX 取值含义	
取值	描述
0x00	ASCII8
0x01	ASCII10
0x02	ASCII13
0x03	宋体 16
0x04	宋体 24
0x05	宋体 32

**DRS:**为显示的次数。取值范围为 0~255,当为 0 的时,表示无限循环显示。

**TC:**为文字的颜色值。存储结构为 R G B A 三基色,各占 8 位,R 表示红色分量,G 表示绿色分量,B 表示蓝色分量,A 目前没用使用,作为保留字。各取值范围为 0~255。

**BC:**为背景的颜色值。存储结构为 R G B A 三基色,各占 8 位,R 表示红色分量,G 表示绿色分量,B 表示蓝色分量,A 目前没用使用,作为保留字。各取值范围为 0~255。

**TL:**为文字的长度。16 位数据类型,小端模式。目前最大长度为 2K 字节。可以存储 1000 个汉字信息。

**TEXT:**为显示的文字内容,支持 ASCII 和 GBK2312 编码。单包最大长度为 255,超出单包最大长度时,需要分包传输。

**回复格式:** DA + SP + PN[2] + 0x62+ DL + ACK + CRC[2]

**回复参数描述:**包含 1 个字节的参数,DL 取值为 1,ACK 是显示屏返回的结果,取值为 0 表示成功,非 0 表示不成功。

**使用例子:** 设置地址为 0 的显示屏的 0 号文本控件的文本为"欢迎光临请入场停车",进入模式为连续移动,进入速度为 1,停留模式为正常停留,停留时间为 2 秒,退出模式为连续移动,退出速度为 1,字体为宋体 16,字体颜色为红色,背景颜色为黑色。

**主机发送:** 00 64 FF FF 62 25 00 15 01 00 02 15 01 03 00 FF 00 00 00 00 00 00 12 00 BB B6 D3 AD B9 E2 C1 D9 C7 EB C8 EB B3 A1 CD A3 B3 B5 F4 F5

**设备回复:** 00 64 FF FF 62 01 00 97 6A

## 0x63(设置文字字体)

设置文字字体的功能是设置文本控件的字体大小。

**请求格式:**DA + SP + PN[2] + 0x63+ DL + TWID + FINDEX+ CRC[2]

**请求参数描述:**包含 10 个字节的参数，DL 取值为 10。

**TWID:**为文本控件 ID,用于标识创建的窗口身份。

**FINDEX:**为字体索引值，取值范围及含义见下表:

FINDEX 取值含义	
取值	描述
0x00	ASCII8
0x01	ASCII10
0x02	ASCII13
0x03	宋体 16
0x04	宋体 24
0x05	宋体 32

**回复格式:** DA + SP + PN[2] + 0x63+ DL + ACK + CRC[2]

**回复参数描述:**包含 1 个字节的参数，DL 取值为 1，ACK 是显示屏返回的结果，取值为 0 表示成功，非 0 表示不成功。

**使用例子:** 更改地址为 0 的显示屏的 0 号文本窗口的字体为宋体 16。

**主机发送:** 00 64 FF FF 63 02 00 03 9A 53

**设备回复:** 00 64 FF FF 63 01 00 C6 AA

## 0x64(设置文字颜色)

设置文字颜色的功能是设置文本控件的文字颜色，也称作前景颜色。

**请求格式:**DA + SP + PN[2] + 0x64+ DL + TWID + TC+ CRC[2]

**请求参数描述:**包含 5 个字节的参数，DL 取值为 5。

**TWID:**为文本控件 ID,用于标识创建的窗口身份。

**TC:**为文字的颜色值。存储结构为 R G B A 三基色，各占 8 位，R 表示红色分量，G 表示绿色分量，B 表示蓝色分量，A 目前没用使用，作为保留字。各取值范围为 0~255。

**回复格式:** DA + SP + PN[2] + 0x64+ DL + ACK + CRC[2]

**回复参数描述:**包含 1 个字节的参数，DL 取值为 1，ACK 是显示屏返回的结果，取值为 0 表示成功，非 0 表示不成功。

**使用例子:** 更改地址为 0 的显示屏的 0 号文本控件的字体颜色为红色。

**主机发送:** 00 64 FF FF 64 05 00 FF 00 00 00 5B F8

**设备回复:** 00 64 FF FF 64 01 00 77 6B

## 0x65(设置背景颜色)

设置背景颜色的功能是设置文本控件的背景颜色。

**请求格式:**DA + SP + PN[2] + 0x65+ DL +TWID+BC+ CRC[2]

**请求参数描述:**包含 5 个字节的参数, DL 取值为 5。

**TWID:**为文本控件 ID,用于标识创建的窗口身份。

**BC:**为文字的颜色值。存储结构为 R G B A 三基色, 各占 8 位, R 表示红色分量, G 表示绿色分量, B 表示蓝色分量, A 目前没用使用, 作为保留字。各取值范围为 0~255。

**回复格式:** DA + SP + PN[2] + 0x65+ DL + ACK + CRC[2]

**回复参数描述:**包含 1 个字节的参数, DL 取值为 1, ACK 是显示屏返回的结果, 取值为 0 表示成功, 非 0 表示不成功。

**使用例子:**更改地址为 0 的显示屏的 0 号文本控件的字体颜色为红色。

**主机发送:** 00 64 FF FF 65 05 00 00 00 00 00 7B 2C

**设备回复:** 00 64 FF FF 65 01 00 26 AB

## 0x66(停止显示)

停止显示的功能是立即清除文本控件正在显示的文字信息。如果立即显示的内容来自显示屏存储的文本文件, 那么清除之后, 显示屏会重新开始显示。

**请求格式:**DA + SP + PN[2] + 0x66+ DL +TWID+ CRC[2]

**请求参数描述:**包含 5 个字节的参数, DL 取值为 5。

**TWID:**为文本控件 ID,用于标识创建的窗口身份。

**回复格式:** DA + SP + PN[2] + 0x66+ DL + ACK + CRC[2]

**回复参数描述:**包含 1 个字节的参数, DL 取值为 1, ACK 是显示屏返回的结果, 取值为 0 表示成功, 非 0 表示不成功。

**使用例子:**清除地址为 0 的显示屏的 0 号文本控件的内容。

**主机发送:** 00 64 FF FF 66 01 00 D6 AB

**设备回复:** 00 64 FF FF 66 01 00 D6 AB

## 0x67(下载文本文件)

ID 为 0~7 文本控件都关联了 16 个独立的文本文件, 存储在显示屏的外部存储器中, 掉电后不会丢失。当对应的文本控件被创建之后, 会主动循环的显示这 16 个文本文件(如果存在)。需要注意的是不要频繁的擦写这些文本文件, 否则会降低显示屏的存储器寿命。一般用来存放一些广告语等信息, 频繁擦写的内容可以用 0x62(设置文本)指令显示。

**请求格式:**DA + SP + PN[2] + 0x67+ DL + TWID + FID + AF + ETM + ETS + DM + DT + EXM + EXS + FINDEX + TC[4] + BC[4] + TL[2] + TEXT[...] + CRC[2]

**请求参数描述:**DL 等于 19 个字节再加上文本长度。

**TWID:**为文本控件 ID,用于标识创建的窗口身份。

**FID:**为文件的 ID, 用于指定操作的文本文件。取值范围为 0~15。

**AF:**为文本文件的属性标志位。目前没有用到, 固定取值为 0x0C。

**ETM:**文字进入窗口的方式。取值范围及含义请参考 0x62(设置文本)指令。

**ETS:**为文字进入的速度。取值范围为 1~32。时基取决于当前的扫描周期。

**DM:**为文字停留的方式。取值范围及含义请参考 0X62(设置文本)指令。

**DT:**为文字停留的时间。取值范围为 0~255。

**EXM:**为文字退场的方式。取值范围及含义请参考 0X62(设置文本)指令。

**EXS:**为文字退场的速度。取值范围为 1~32。时基取决于当前的扫描周期。

**FINDEX:**为文字的字体索引值。取值范围及含义请参考 0X62(设置文本)指令。

**TC:**为文字的颜色值。存储结构为 R G B A 三基色，各占 8 位，R 表示红色分量，G 表示绿色分量，B 表示蓝色分量，A 目前没用使用，作为保留字。各取值范围为 0~255。

**BC:**为背景的颜色值。存储结构为 R G B A 三基色，各占 8 位，R 表示红色分量，G 表示绿色分量，B 表示蓝色分量，A 目前没用使用，作为保留字。各取值范围为 0~255。

**TL:**为文字的长度。16 位数据类型，小端模式。目前最大长度为 60K 字节。可以存储 3 万个汉字信息。

**TEXT:**为显示的文字内容，支持 ASCII 和 GBK2312 编码。单包最大长度为 255,超出单包最大长度时，需要分包传输。

**回复格式:** DA + SP + PN[2] + 0x67+ DL + ACK + CRC[2]

**回复参数描述:**包含 1 个字节的参数，DL 取值为 1，ACK 是显示屏返回的结果，取值为 0 表示成功，非 0 表示不成功。

**使用例子:** 下载“一路顺风”到地址为 0 的显示屏的 0 号文本控件的 0 号文本文件，进入模式为连续移动，进入速度为 1，停留模式为正常停留，停留时间为 2 秒，退出模式为连续移动，退出速度为 1，字体为宋体 16，字体颜色为红色，背景颜色为黑色。

**主机发送:** 00 64 FF FF 67 1C 00 00 0C 15 01 00 02 15 01 03 FF 00 00 00 00 00 00 08 00 D2 BB C2 B7 CB B3 B7 E7 DC 4C

**设备回复:** 00 64 FF FF 67 01 00 87 6B

## 0x68(显示文本文件)

显示文本文件的功能是在指定的文本控件上立即显示已经下载的文本文件。

**请求格式:** DA + SP + PN[2] + 0x68+ DL + TWID+FID+ CRC[2]

**请求参数描述:**包含 2 个字节的参数，DL 取值为 2。

**TWID:**为文本控件 ID,用于标识创建的窗口身份。

**FID:**为文件的 ID，用于指定操作的文本文件。取值范围为 0~15。

**回复格式:** DA + SP + PN[2] + 0x68+ DL + ACK + CRC[2]

**回复参数描述:**包含 1 个字节的参数，DL 取值为 1，ACK 是显示屏返回的结果，取值为 0 表示成功，非 0 表示不成功。

**使用例子:** 显示地址为 0 的显示屏的 0 号文本控件的 0 号文本文件。

**主机发送:** 00 64 FF FF 68 02 00 00 D8 76

**设备回复:** 00 64 FF FF 68 01 00 B7 68

## 0x69(打开文本文件)

打开文本文件的功能是标记一个文本控件的文本文件的显示状态，被打开的文件会被相关联的文本窗口自动调用显示。

**请求格式:**DA + SP + PN[2] + 0x69+ DL + TWID+FID+ CRC[2]

**请求参数描述:**包含 2 个字节的参数，DL 取值为 2。

**TWID:**为文本控件 ID,用于标识创建的窗口身份。

**FID:**为文件的 ID，用于指定操作的文本文件。取值范围为 0~15。

**回复格式:** DA + SP + PN[2] + 0x69+ DL + ACK + CRC[2]

**回复参数描述:**包含 1 个字节的参数，DL 取值为 1，ACK 是显示屏返回的结果，取值为 0 表示成功，非 0 表示不成功。

**使用例子:** 打开地址为 0 的显示屏的 0 号文本控件的 0 号文本文件。

**主机发送:** 00 64 FF FF 69 02 00 00 D9 8A

**设备回复:** 00 64 FF FF 69 01 00 E6 A8

## 0x6A(关闭文本文件)

关闭文本文件的功能是标记一个文本控件的文本文件不被该窗口循环显示。

**请求格式:**DA + SP + PN[2] + 0x6A+ DL + TWID+FID+ CRC[2]

**请求参数描述:**包含 2 个字节的参数，DL 取值为 2。

**TWID:**为文本控件 ID,用于标识创建的窗口身份。

**FID:**为文件的 ID，用于指定操作的文本文件。取值范围为 0~15。

**回复格式:** DA + SP + PN[2] + 0x6A+ DL + ACK + CRC[2]

**回复参数描述:**包含 1 个字节的参数，DL 取值为 1，ACK 是显示屏返回的结果，取值为 0 表示成功，非 0 表示不成功。

**使用例子:** 打开地址为 0 的显示屏的 0 号文本控件的 0 号文本文件。

**主机发送:** 00 64 FF FF 6A 02 00 00 D9 CE

**设备回复:** 00 64 FF FF 6A 01 00 16 A8

## 0x6B(删除文本文件)

被删除的文本文件将无法恢复。如果相关联的文本控件正在调用这个文本文件，会立即停止显示。

**请求格式:**DA + SP + PN[2] + 0x6B+ DL + TWID+FID+ CRC[2]

**请求参数描述:**包含 2 个字节的参数，DL 取值为 2。

**TWID:**为文本控件 ID,用于标识创建的窗口身份。

**FID:**为文件的 ID，用于指定操作的文本文件。取值范围为 0~15。

**回复格式:** DA + SP + PN[2] + 0x6B+ DL + ACK + CRC[2]

**回复参数描述:**包含 1 个字节的参数，DL 取值为 1，ACK 是显示屏返回的结果，取值为 0 表示成功，非 0 表示不成功。

**使用例子:** 打开地址为 0 的显示屏的 0 号文本控件的 0 号文本文件。

**主机发送:** 00 64 FF FF 6B 02 00 00 D8 32

**设备回复:** 00 64 FF FF 6B 01 00 47 68

## 0x6C(同步显示)

同步显示指令用来解决让多个文本控件不同时显示的问题。通过该指令可以让任意几组文本控件同步开始显示，并且每个文本控件的显示文件可单独指定。

**请求格式:** DA + SP + PN[2] + 0x6C + DL + TWIDn + FIDn + ..... + CRC[2]

**请求参数描述:** 可变长参数，最大为 255 字节。

**TWIDn:** 为文本控件 ID, 用于指定需要同步的文本控件。

**FIDn:** 为文件的 ID, 用于指定需要显示的文本文件。取值范围为 0~15。

**(注意:** 只有 ID 为 0~7 的文本控件才有效)

**回复格式:** DA + SP + PN[2] + 0x6B + DL + ACK + CRC[2]

**回复参数描述:** 包含 1 个字节的参数，DL 取值为 1，ACK 是显示屏返回的结果，取值为 0 表示成功，非 0 表示不成功。

**使用例子:** 同步显示地址为 0 的显示屏的 0 号文本控件的 0 号文本文件和 1 号文本控件的 0 号文本文件。

**主机发送:** 00 64 FF FF 6C 04 00 00 01 00 53 F2

**设备回复:** 00 64 FF FF 6C 01 00 F6 A9



## 0x6D(单包同步显示, 带语音)

使用单包同步显示, 减少了频繁的单独设置窗口文本的工作, 每行的内容用 ‘\n’ 换行符分隔, 语音用 ‘\a’ 分隔。但是该指令也有一些限制, 其文本头和文字以及语音的长度之和, 不能超出单包的最大长度, 即 255 个字节。

**请求格式:** DA + SP + PN[2] + 0x6D + DL + STWID + ETM + ETS + DM + DT + EXM + EXS + FINDEX + DRS + TC + BC + TL[2] + TEXT[...] + VOICE[...] + CRC[2]

**请求参数描述:** 可变长参数, 最大为 255 字节。

**STWID:** 为第一个设置的文本控件 ID。

**ETM, ETS, DM, DT, EXM, EXS, FINDEX, DRS, TC, BC, TL[2], TEXT[.]:** 请参考 0X62 指令解释。

**VOICE:** 声音文本内容。

**回复格式:** DA + SP + PN[2] + 0x6D + DL + ACK + CRC[2]

**回复参数描述:** 包含 1 个字节的参数, DL 取值为 1, ACK 是显示屏返回的结果, 取值为 0 表示成功, 非 0 表示不成功。

**使用例子:** 从第一个窗口至第 3 个窗口显示 “123”、“456”、“789”, 声音播 “报欢迎光临”。

**主机发送:** 00 64 FF FF 6D 29 00 15 01 00 02 15 01 03 00 FF 00 00 00 00 00 00 00 00 0B 00 31 32 33 0A 34 35 36 0A 37 38 39 0A BB B6 D3 AD B9 E2 C1 D9 07 00 93 4F

**设备回复:** 00 64 FF FF 6D 01 00 A7 69

## 0x80(创建图片控件)

图片控件用来显示图标, 而且可以实现播放动画。

**请求格式:** DA + SP + PN[2] + 0x80 + DL + PICID + X0[2] + Y0[2] + XSIZE[2] + YSIZE[2] + CRC[2]

**请求参数描述:** 包含 9 个字节的参数, DL 取值为 9。

**PICID:** 为图标控件的 ID, 用于标识创建的窗口身份。

**X0:** 为文本控件的起始点横轴坐标, 16 位数据类型, 小端模式。

**Y0:** 为文本控件的起始点纵轴坐标, 16 位数据类型, 小端模式。

**XSIZE:** 为文本控件的横向长度, 16 位数据类型, 小端模式。

**YSIZE:** 为文本控件的纵向宽度, 16 位数据类型, 小端模式。

**回复格式:** DA + SP + PN[2] + 0x80 + DL + ACK + CRC[2]

**回复参数描述:** 包含 1 个字节的参数, DL 取值为 1, ACK 是显示屏返回的结果, 取值为 0 表示成功, 非 0 表示不成功。

**使用例子:** 在地址为 0 的显示屏的(0,0)位置创建一个大小为 64\*32 的图片控件。

**主机发送:** 00 64 FF FF 80 09 00 00 00 00 00 40 00 20 00 53 1D

**设备回复:** 00 64 FF FF 80 01 00 37 5C

### 0x81(删除图片控件)

删除图片控件的功能是从窗口管理器中移除已经被创建的图片控件。被删除的图片控件不能再被操作。

**请求格式:**DA + SP + PN[2] + 0x81+ DL + PICID + CRC[2]

**请求参数描述:**包含 1 个字节的参数，DL 取值为 1。

**PICID:**为图标控件的 ID。

**回复格式:** DA + SP + PN[2] + 0x81+ DL + ACK + CRC[2]

**回复参数描述:**包含 1 个字节的参数，DL 取值为 1，ACK 是显示屏返回的结果，取值为 0 表示成功，非 0 表示不成功。

**使用例子:** 删除地址为 0 的显示屏的 ID 为 0 的图片控件。

**主机发送:** 00 64 FF FF 81 01 00 66 9C

**设备回复:** 00 64 FF FF 81 01 00 66 9C

### 0x82(追加图片)

追加图片功能是向一个图片控件的播放列表里面增加新的图片。

**请求格式:**DA + SP + PN[2] + 0x82+ DL + PICID + ICON[2] + ETM + ETS + SM + ST + EXM + EXS + CRC[2]

**请求参数描述:**包含 9 个字节的参数，DL 取值为 9。

**PICID:**为图标控件的 ID。

**ICON:**为要增加的图片 ID,16 位数据类型，小端模式。

**ETM:**显示图片的方式。

ETM 取值含义	
取值	描述
0x00	立即显示
0x01	从右向左移动
0x02	从左向右移动
0x03	从下向上移动
0x04	从上向下移动

**ETS:**显示图片的速度：显示速度的单位为帧，取值范围为 1~32。

**SM:**图片停留的方式,目前最为保留值取值为 0x00。

**ST:**图片停留的时间,单位为秒钟。

**EXM:**图片退出的方式。

EXM 取值含义	
取值	描述
0x00	立即退出
0x01	从右向左移动
0x02	从左向右移动
0x03	从下向上移动
0x04	从上向下移动
0x05	无动作
0x06	循环从右向左移动
0x07	循环从左向右移动
0x08	循环从上向下移动
0x09	循环从下向上移动

**EXS:**图片退出的速度,单位为帧, 取值范围为 1~32.

**回复格式:** DA + SP + PN[2] + 0x82+ DL + ACK + CRC[2]

**回复参数描述:**包含 1 个字节的参数, DL 取值为 1, ACK 是显示屏返回的结果, 取值为 0 表示成功, 非 0 表示不成功。

**使用例子:** 地址为 0 的显示屏的图片控件 ID 为 0 追加图片 0。

**主机发送:** 00 64 FF FF 82 09 00 00 00 00 01 00 02 06 01 0A C5

**设备回复:** 00 64 FF FF 82 01 00 96 9C

### 0x83(清除图片列表)

清除图片列表的功能是将图片控件的队列清空。

**请求格式:**DA + SP + PN[2] + 0x83+ DL + PICID+ CRC[2]

**请求参数描述:**包含 1 个字节的参数, DL 取值为 1。

**PICID:**为图标控件的 ID。

**回复格式:** DA + SP + PN[2] + 0x83+ DL + ACK + CRC[2]

**回复参数描述:**包含 1 个字节的参数, DL 取值为 1, ACK 是显示屏返回的结果, 取值为 0 表示成功, 非 0 表示不成功。

**使用例子:** 清除地址为 0 的显示屏的 ID 为 0 的图片控件的图片列表。

**主机发送:** 00 64 FF FF 83 01 00 C7 5C

**设备回复:** 00 64 FF FF 83 01 00 C7 5C

## 2.4 显示时间

显示卡显示时间的方式采用转义字符替换的方案。用户在文字中插入相应的转义字符，显示卡收到后会对插入的位置用对应的时间替换成显示的字符。

支持的转义字符有：

`C 替换成常量值 20

`Y 表示年

`M 表示月

`D 表示日

`V 表示星期

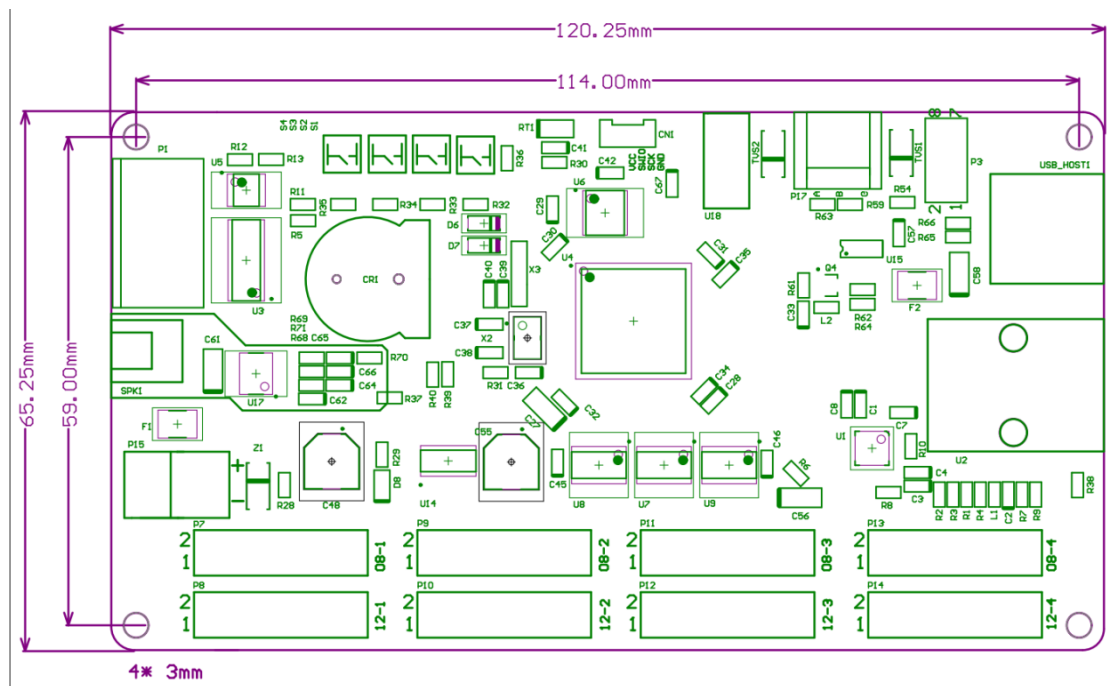
`H 表示小时

`N 表示分钟

`S 表示秒钟

例子："现在时间 20`Y 年`M 月`D 日星期`V `H:`N:`S"

# 显示屏控制卡尺寸



## 附表 1

语音列表：

序号	词、词组、短语
1	祝你早日康复
2	祝您一路顺风
3	祝您一路平安
4	注意安全
5	注册成功
6	重复外出允许
7	重复外出不允许
8	重复外出
9	重复进入允许
10	重复进入不允许
11	重复进入
12	重复读卡
13	纸票机故障
14	闸杆下落
15	暂停使用
16	载客人数
17	月租车到期
18	月租欠费
19	月租快到期
20	月租车未授权
21	月租车剩余
22	月卡车
23	余额不足请及时充值
24	余额不足
25	有效期至
26	已缴费
27	已过期
28	已超过
29	一路顺风
30	一路平安
31	星期
32	信息已确认
33	谢谢惠顾
34	谢谢

35	写卡成功
36	小时
37	现在时间
38	现在
39	系统过期
40	无效卡
41	无权入场
42	无权出场
43	无牌车
44	无卡请加卡
45	未注册卡
46	未授权卡
47	未授权
48	停车时长
49	停车
50	抬闸保持
51	手工落闸
52	收费中
53	收费
54	识别成功
55	时间
56	剩余金额
57	剩余车位
58	剩余
59	设备地址为
60	入口
61	入场时间
62	人工停闸
63	人工入场
64	人工放行
65	人工出场
66	区已满
67	请自检
68	请注意
69	请与管理员联系
70	请勿取卡
71	请往入口处刷卡入场
72	请往其他停车场
73	请往出口处刷卡出场
74	请通行
75	请通过
76	请刷卡或取卡
77	请刷卡

78	请稍后
79	请稍候
80	请塞卡出场
81	请入场停车
82	请入场
83	请取卡入场
84	请取卡
85	请靠边停车检查
86	请尽快延期
87	请交还卡片
88	请交费
89	请将卡片投入卡箱内
90	请见谅
91	请减速慢行
92	请付现金
93	请读卡取卡
94	请读卡交费
95	请读卡
96	请等待确认
97	请充值
98	请插入卡片
99	请泊车
100	欠费外出
101	欠费进入
102	您好
103	内存已满请及时上传
104	内存故障
105	内部车场不允许进入
106	免费停车
107	免费卡
108	密码错误
109	临时车无入场记录
110	临时车无权限
111	临时车辆
112	临时车
113	扣款成功
114	可用日期
115	卡已锁定
116	卡已过期
117	卡箱已空
118	卡片不足
119	军警车免费通行
120	禁止通行



121	禁止入场
122	禁止出场
123	进出延时
124	解挂成功
125	交费
126	加密成功
127	即将过期
128	欢迎再次光临
129	欢迎下次光临
130	欢迎使用车牌识别
131	欢迎使用
132	欢迎入场
133	欢迎光临
134	欢迎泊车
135	过期用户
136	挂失成功
137	固定车
138	该时段不允许进入
139	发卡器有故障
140	对不起
141	读卡有效
142	读卡器故障
143	读卡交费
144	读卡成功
145	电量过低
146	电量正常
147	地址
148	道闸已开启
149	单双号限行
150	此卡余额不足
151	此卡有效期至
152	此卡已入场
153	此卡已解挂
154	此卡已挂失
155	此卡已出场
156	此卡已被使用
157	此卡无效
158	此卡无权出入
159	此卡未入场
160	此卡剩余天数为
161	此卡剩余试用期
162	此卡剩余金额
163	此卡可用时间

164	此卡机缺卡
165	此卡还未使用
166	此卡不符
167	此卡被禁用
168	此卡已超时
169	此卡已被锁定
170	此车位车辆已经入场
171	此车位车辆已进场
172	此车为黑名单车辆
173	此车可用日期
174	此车号无效
175	储值车
176	出入平安
177	出口
178	出口机故障
179	出场
180	车已在场
181	车已过期
182	车位有余
183	车位已满
184	车未入场
185	车未发行
186	车牌号为
187	车牌不符
188	超时交费
189	操作成功
190	本车位已停车
191	正
192	整
193	浙
194	云
195	粤
196	月
197	元
198	豫
199	渝
200	寅
201	乙
202	学
203	戌
204	新
205	辛
206	湘

207	午
208	位
209	未
210	为
211	万
212	皖
213	天
214	台
215	苏
216	使
217	时
218	十
219	沈
220	深
221	申
222	陕
223	日
224	壬
225	人
226	琼
227	青
228	千
229	农
230	宁
231	年
232	南
233	闽
234	秒
235	蒙
236	鲁
237	辽
238	两
239	兰
240	空
241	军
242	警
243	京
244	晋
245	津
246	角
247	甲
248	冀
249	济

250	己
251	吉
252	沪
253	黑
254	号
255	海
256	桂
257	广
258	挂
259	庚
260	个
261	港
262	赣
263	甘
264	负
265	分钟
266	分
267	鄂
268	点
269	川
270	成
271	辰
272	藏
273	丙
274	北
275	半
276	百
277	澳
278	A
279	B
280	C
281	D
282	E
283	F
284	G
285	H
286	L
287	J
288	K
289	L
290	M
291	N
292	O

293	P
294	Q
295	R
296	S
297	T
298	U
299	V
300	W
301	X
302	Y
303	Z

语音列表

out	操作成功.wav	此卡无权出入.wav	海.wav	辽.wav
转换后	超过离场限时.wav	此卡无效.wav	号.wav	临.wav
0.wav	超时交费.wav	此卡已被使用.wav	黑.wav	临时车.wav
1.wav	车场已满.wav	此卡已出场.wav	沪.wav	临时车辆.wav
2.wav	车号已过期.wav	此卡已挂失.wav	欢迎泊车.wav	临时车无权限.wav
3.wav	车辆称重出现异常.wav	此卡已过期.wav	欢迎光临.wav	临时车无入场记录.wav
4.wav	车辆禁止出场.wav	此卡已解挂.wav	欢迎回家.wav	领取告知卡.wav
5.wav	车辆未在规定时间内离场.wav	此卡已入场.wav	欢迎您下次光临.wav	鲁.wav
6.wav	车牌.wav	此卡有效期至.wav	欢迎入场.wav	蒙.wav
7.wav	车牌不符.wav	此卡余额不足.wav	欢迎使用.wav	密码错误.wav
8.wav	车牌号为.wav	单双号限行.wav	欢迎使用车牌识别.wav	免.wav
9.wav	车牌识别失败.wav	道闸已开启.wav	欢迎下次光临.wav	免费车辆.wav
A.wav	车未放行.wav	地址.wav	欢迎再次光临.wav	免费卡.wav
B.wav	车未入场.wav	地址为.wav	吉.wav	免费停车.wav
C.wav	车位已满.wav	点.wav	即将过期.wav	秒.wav
D.wav	车位已使用.wav	电量过低.wav	己.wav	闽.wav
E.wav	车位已用.wav	电量正常.wav	济.wav	内部车场不允许进入.wav
F.wav	车位已用完.wav	东方尊峪.wav	冀.wav	内部车辆.wav
G.wav	车位有余.wav	读卡成功.wav	加密成功.wav	内存故障.wav
H.wav	车已过期.wav	读卡交费.wav	甲.wav	内存已满请及时上传.wav
I.wav	车已在场.wav	读卡器故障.wav	交费.wav	内到达过磅房进行二次称重.wav
IP.wav	辰.wav	读卡有效.wav	角.wav	内离开公司.wav
J.wav	称重完毕.wav	端口.wav	缴费过期.wav	南.wav
K.wav	称重重量.wav	端口为.wav	解挂成功.wav	年.wav
L.wav	成.wav	对不起.wav	金色领航.wav	您好.wav
M.wav	出场.wav	吨.wav	金城东郡.wav	您为.wav
MNull.wav	出卡机故障.wav	鄂.wav	津.wav	您有快速在管理处.wav
N.wav	出口.wav	发卡器有故障.wav	进出延时.wav	宁.wav
O.wav	出入平安.wav	分.wav	进入公司车辆.wav	农.wav
P.wav	储值车.wav	分钟.wav	晋.wav	排队成功.wav
Q.wav	川.wav	负.wav	禁止出场.wav	千.wav
R.wav	春节快乐.wav	该年.wav	禁止出入.wav	欠费进入.wav
S.wav	该车到期时间剩余.wav	该车称重异常未消除.wav	禁止进入.wav	欠费外出.wav
scale01.wav	该车号无效.wav	该车未经装卸点确认.wav	禁止入场.wav	青.wav
scale02.wav	该车可用日期.wav	该车无入场记录.wav	禁止通行.wav	请按公司相关规定办理手续.wav
scale03.wav	该车为黑名单车辆.wav	该车属于.wav	京.wav	请泊车.wav
SNull.wav	该车未登记.wav	该时段不允许进入.wav	警.wav	请插入卡片.wav
SysStart.wav	此车位车辆已进场.wav	甘.wav	拒绝进入.wav	请充值.wav
T.wav	此车位车辆已入场.wav	赣.wav	军.wav	请充值交费.wav
U.wav	此车已被锁定.wav	港.wav	军警车免费通行.wav	请到管理处交费续期.wav
V.wav	此车已超时.wav	个.wav	卡片不足.wav	请到管理处缴费.wav
W.wav	此车已过期.wav	庚.wav	卡箱已空.wav	请到收费处缴费.wav
X.wav	此车已开具出门条.wav	固定车.wav	卡已过期.wav	请登记放行.wav
Y.wav	此卡被禁用.wav	挂.wav	卡已锁定.wav	请等待确认.wav
Z.wav	此卡不符.wav	挂失成功.wav	开闸请注意.wav	请读卡.wav
澳.wav	此卡还未使用.wav	关闭请注意.wav	科发语音屏.wav	请读卡交费.wav
百.wav	此卡机缺卡.wav	管控车辆.wav	可用日期.wav	请读卡取卡.wav
半.wav	此卡即将过期.wav	广.wav	空.wav	请返回过磅房接受处理.wav
北.wav	此卡可用时间.wav	费.wav	扣款成功.wav	请返回装卸点处理.wav
本车位已停车.wav	此卡剩余金额.wav	挂.wav	扣款金额为.wav	请付现金.wav
丙.wav	此卡剩余使用期.wav	国庆节快乐.wav	兰.wav	请给予配合.wav
保持率为.wav	此卡剩余天数.wav	过磅车辆.wav	劳动节快乐.wav	请及时缴纳管理费.wav
赣.wav	此卡未入场.wav	过期用户.wav	丙.wav	请及时缴纳水电费.wav

请减速离秤.wav	设备地址为.wav	新.wav	重复读卡.wav
请减速慢行.wav	申.wav	新年快乐.wav	重复进入.wav
请检查核对后放行.wav	深.wav	信息已确认.wav	重复进入不允许.wav
请见谅.wav	沈.wav	星期.wav	重复进入允许.wav
请将卡片投入卡箱内.wav	圣诞节快乐.wav	戌.wav	重复外出.wav
请交费.wav	剩余.wav	需要办理预约手续.wav	重复外出不允许.wav
请交还卡片.wav	剩余车位.wav	学.wav	重复外出允许.wav
请接受检查.wav	剩余金额.wav	业主车辆.wav	注册成功.wav
请谨慎驾驶.wav	十.wav	液氨车辆卸货后.wav	注意安全.wav
请尽快延期.wav	时.wav	一路平安.wav	祝您健康.wav
请进行登记.wav	时间.wav	一路顺风.wav	祝您一路平安.wav
请靠边停车检查.wav	识别成功.wav	乙.wav	祝您一路顺风.wav
请取卡.wav	使.wav	已超过授权时限.wav	祝您早日康复.wav
请取卡入场.wav	收费.wav	已超时.wav	
请入场.wav	收费中.wav	已过期.wav	
请入场停车.wav	手工开闸.wav	已过有效期.wav	
请塞卡出场.wav	手工落闸.wav	已缴费.wav	
请稍等.wav	授权车辆.wav	已授权.wav	
请稍后.wav	四季花城.wav	已注销.wav	
请稍候.wav	苏.wav	以临时车方式入场.wav	
请刷卡.wav	台.wav	寅.wav	
请刷卡或取卡.wav	抬闸保持.wav	有效期至.wav	
请提前充值.wav	天.wav	渝.wav	
请提前续费.wav	停车.wav	余额不足.wav	
请停车检查.wav	停车时长.wav	余额不足请及时充值.wav	
请通过.wav	停留时间超过.wav	预约车辆.wav	
请通行.wav	外来车辆.wav	豫.wav	
请往出口处刷卡出场.wav	皖.wav	元.wav	
请往其他停车场.wav	万.wav	元旦节快乐.wav	
请往入口处刷卡入场.wav	为.wav	月.wav	
请勿取卡.wav	未.wav	月卡.wav	
请先到中心岗亭缴费.wav	未缴费.wav	月卡车.wav	
请先登记.wav	未进行预报.wav	月租车.wav	
请与管理员联系.wav	未授权.wav	月租车剩余.wav	
请在.wav	未授权卡.wav	月租车未授权.wav	
请重新缴费.wav	未注册卡.wav	月租到期.wav	
请重新进入识别区.wav	位.wav	月租快到期.wav	
请注意.wav	无卡请加卡.wav	月租欠费.wav	
请注意交通安全.wav	无牌车.wav	月租已到期.wav	
请自检.wav	无权出场.wav	粤.wav	
琼.wav	无权入场.wav	云.wav	
区已满.wav	无剩余车位.wav	允许入场.wav	
人.wav	无效卡.wav	载客人数.wav	
人工出场.wav	午.wav	暂停使用.wav	
人工放行.wav	系统过期.wav	闸杆下落.wav	
人工入场.wav	现在.wav	浙.wav	
人工停闸.wav	现在时间.wav	整.wav	
壬.wav	湘.wav	正.wav	
日.wav	小时.wav	正常归队.wav	
入场.wav	写卡成功.wav	正在计费.wav	
入场时间.wav	谢谢.wav	正在人工处理.wav	
入口.wav	谢谢惠顾.wav	纸票机故障.wav	
陕.wav	辛.wav	中秋节快乐.wav	

**播报金钱语音的匹配格式:**

- 1.请交费 XX.YY 元
- 2.请缴费 XX.YY 元
- 3.您的卡上余额为 XX.YY 元
- 4.¥XX.YY 元
- 5.收费 XX.YY 元
- 6.金额 XX.YY 元
- 7.此刻剩余金额 XX.YY 元
- 8.扣款金额为 XX.YY 元

**播报有效期语音的匹配格式:**

- 1.此卡有效期至 YY-MM-DD
- 2.有效期至 YY-MM-DD

**播报剩余天数语音的匹配格式:**

- 1.此卡可用日期 DD 天
- 2.剩余天数为 DD 天
- 3.剩余 DD 天
- 4.剩余 DD 日
- 5.月租车剩余 DD 日
- 6.此卡剩余使用期 DD 天
- 7.此卡剩余天数为 DD 天

**播报停车时长语音的匹配格式:**

- 1.此次停车时间为 DD 天 HH 小时 MM 分钟
- 2.停车时间 DD 天 HH 小时 MM 分钟
- 3.停车时长 DD 天 HH 小时 MM 分钟
- 4.停车 DD 天 HH 小时 MM 分钟

**播报剩余车位语音的匹配格式:**

- 1.剩余车位 NN 位
- 2.剩余 NN 位
- 3.空车位 NN 位

**播报车牌号码语音的匹配格式:**

- 1.车牌号码 P
- 2.车牌号为 P
3. P

**播报入场时间语音的匹配格式:**

- 1.入场时间为 HH-MM

**更多应用请与我司联系!**