

	深圳市恒域威电子有限公司
	HENZHEN HENGYUWEI ELECTRONICS CO., LTD
	四阶文件 Four order file
	产品使用说明书 Product manual

<<恒域威智能串口触摸屏使用手册>>

非常感谢您关注和使用我们的液晶显示屏，欢迎您提出宝贵的意见和建议，我们将竭诚为您服务！您可以使用下列方式获取具体的技术咨询与服务。



电话:+86(0755)29100074

传真:+86(0755)29100079

Email:hywdz@126.com

<http://www.hywlcm.com>

公司地址：深圳市宝安区西乡固戍水产井湾第一工业区 B 栋 7 层

开发使用手册改版说明			
版本	日期	说明	改版人（工号）
1.0	15-07-31	第一版	HYW070
2.0	16-05-06	第二版	HYW089

目 录

1 HGUS 概述目录	3
1.1 认识 HGUS 屏	5
1.2 HGUS 屏主要特点	6
1.3 HGUS 屏数据格式	8
1.4 HGUS 软件处理流程图	9
1.5 HGUS 屏开发流程	10
2 HGUS 屏配置	12
2.1 SD 配置接口	12
2.2 SD 下载文件格式说明	13
2.2.1 图片下载	13
2.2.2 字库下载	14
2.2.3 config.txt 配置文件	14
2.2.4 声音文件下载	15
2.2.5 HGUS 应用程序升级	16
2.2.6 用户数据库SD 卡导出	16
2.3 HGUS 屏参数配置	17
2.3.1 屏幕物理分辨率设置 (R0)	17
2.3.2 串口波特率设置 (R1、R5、R9)	17
2.3.3 串口通信帧头设置 (R3、RA)	17
2.3.4 软件工作模式配置寄存器 (R2、RC)	18
2.3.5 触摸屏控制背光 (R2.5 R6 R7 R8)	19
2.3.6 触摸屏校准	19
3 串口操作	20
3.1 数据帧结构	20
3.2 指令集	20
3.3 串口CRC 校验C 程序参考	21
4 寄存器	23
4.1 寄存器一览表	23
4.2 读写RTC	25
4.3 字库读取	25
4.4 128 段音乐播放	26
4.5 数据库读写	26
4.6 按键触发	28
5 触控/键控配置文件 (13.BIN) 说明	28
5.1 触控/键控功能一览表	29
5.2 变量数据录入 (0x00)	30
5.3 弹出菜单选择 (0x01)	32
5.4 增量调节 (0x02)	33
5.5 拖动调节 (0x03)	34
5.6 RTC 设置 (0x04)	35
5.7 按键值返回 (0x05)	36
5.8 文本录入 (0x06)	36
5.8.1 ASCII 文本录入	37
5.8.2 GBK 汉字文本录入	38
5.9 硬件参数配置 (0x07)	39
5.10 触摸屏按压状态同步数据返回 (0x08)	40
5.11 转动调节 (0x09)	41
6 显示变量配置文件 (14.BIN) 说明	42
6.1 显示变量功能一览表	42
6.2 图标变量	44
6.2.1 变量图标显示 (0x00)	44
6.2.2 动画图标显示 (0x01)	45
6.2.3 滑块刻度指示 (0x02)	46

6.2.4 艺术字变量显示 (0x03)47

6.2.5 图片动画显示 (0x04)48

6.2.6 图标旋转指示 (0x05)48

6.2.7 位变量图标显示 (0x06)50

6.3 文本变量51

6.3.1 数据变量显示 (0x10)51

6.3.2 文本显示 (0x11)52

6.3.3 RTC 显示 (0x12)53

6.3.4 HEX 变量显示 (0x13)55

6.3.5 文本滚屏显示 (0x14)55

6.4 图形变量56

6.4.1 实时曲线 (趋势图) 显示 (0x20)56

6.4.2 基本图形显示 (0x21)57

6.4.3 列表显示 (0x22)59

6.4.4 二维QR 码图形显示 (0x25)61

7. HGUS特殊应用软件使用说明.....62

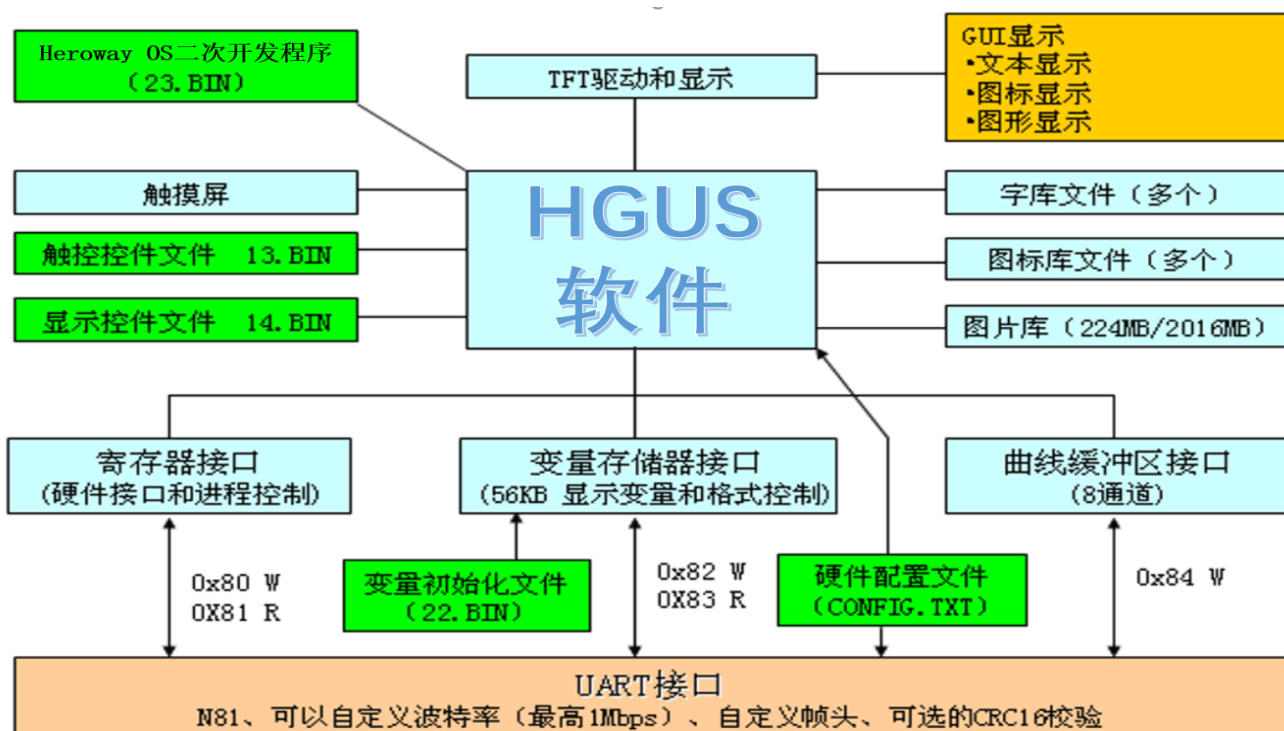
7.1 基于Modbus主机接口的HGUS软件应用说明.....62

7.2 基于Modbus从机接口的HGUS软件应用说明.....63

1 HGUS 概述

1.1 认识恒域威 HGUS 屏

HGUS(Heroway Graphic Utilized Software)是“恒域威电子”有限公司设计的智慧型、图形界面、人机系统软件，软件架构如下图所示：



出厂预装 HGUS 软件的屏称为 HGUS 屏，用最常见的 7 寸屏举例（HYW070S1.1T），硬件组成如下图所示：



1.2 HGUS 屏主要特点

HGUS 屏的主要特点包括：

把图形分解成控件并按页面来配置，控件显示直接由变量控制；

在通过PC上位机软件配置好控件文件（14.BIN，13.BIN）下载到HGUS屏后，用户仅仅需要通过串口改写变量值即可实现控件显示的相应改变。

比如，要在某个页面显示两位小数（2 5.00）的温度值，分两步开发完成：

Step1 配置：借助我司提供的上位机软件，在相应页面位置添加一个数据变量控件，设置好显示格式（字体大小、显示颜色、单位、小数点长度、数据源、数据类别），预览OK后通过SD卡把生成的控件文件（14.BIN）下载到HGUS屏。















Step2 运行：用户软件只需要定时（或者参数变化时）把温度值通过串口刷新到对应的数据源地址，当显示切换到对应页面时，屏幕就自动按照预先设定显示格式显示出来。

触摸屏或键盘录入过程，通过PC软件按照页面定义的触控文件（13.BIN）来控制，用户软件仅需要定时（或者参数改变时的串口中断触发）来读取录入变量值即可。

比如，要在某个页面用触摸屏录入显示两位小数（2 5.00）的设定温度值，分两步开发完成：

Step1 配置：借助我司提供的上位机软件，在相应页面位置添加一个变量数据录入控件，设置好录入格式（字体大小、光标模式、显示颜色、小数点长度、数据源、数据类别），预览OK后通过SD卡把生成的控件文件（13.BIN）下载到HGUS屏。

Step2 运行：当切换到对应页面，并按压触摸屏触发相应控件，HGUS屏会自动完成录入过程。用户软件只需要定时（或者配置为录入完成自动串口下发给用户）查询录入的值即可。

 可靠的硬件平台，经历近 10 年的工业应用，恒域威自主知识产权的软件设计，性能优越，运行稳定可靠。恒域威液晶屏已为新能源、充电桩、医疗、电力等众多领域的国内外用户提供了高性价比的智能串口屏，部分产品在中国移动、中国电信、华为等基站及机房不间断运行超 500 天
 串口工作模式: 显示终端串口通信均采用异步、全双工模式，其工作属性为 8n1 模式。
 恒域威串口触摸屏，具备强大的抗干扰能力，适合室内各种恶劣环境使用。用户 MCU 只需要一个串口就能轻松实现文本、GUI、图片、动画显示和触摸控制等功能。
 为用户节省 99%的程序开发量，大大节省工程师开发时间，缩短开发周期。降低开发成本。是新产品开发或是设备升级的最佳选择。
 产品支持多种常用组态控件：按钮控件、文本控件、仪表控件、键盘、图标控件、动画控件、进度条控件、滑块控件、下拉菜单控件、曲线显示控件、二维码控件和选择控件。
 支持语音、视频播放。
 工业级标准，支持 ESD EMI 测试。
 SD 卡传图导入，通过串口发送数量到 SD 卡的文件中，可以对文件进行添加、删除、修改。可以通过串口发送图片到 SD 卡中，屏显示 SD 中图片。
 把图形分解成控件并按页面来配置，控件显示直接由变量控制；在通过 PC 软件配置好控件文件下载到智能串口屏后，用户仅仅需要通过串口改写变量值即可实现控件显示的相应改变。自由度高，灵活操作，支持图片、字库、配置文件的下载升级。
 支持 RTC 时钟。
 支持屏幕大容量存储功能。图片可存储 1000-2000 张。
 亮度支持 300nit 可进行 64 级亮度调节 300-1000nit 可选。
 支持单点触摸/多点触摸。
 可自定义任意字体显示或 UNICODE 字库显示。

1.3 HGUS 屏数据格式

由于主要面向 MCU 等嵌入式系统应用，为了用户处理的方便，HGUS 屏使用的数据采用整数（字）、无符号整数（字）、长整数（双字）、超长整数（4 个字）表示，相关表示范围如下：

整数：-32768 (0x8000) 到+32767

(0x7FFF)

无符号整数：0 (0x0000) 65535

(0xFFFF)

长整数：-2147483648 (0x80000000) 到+2147483647 (0x7FFFFFFF)

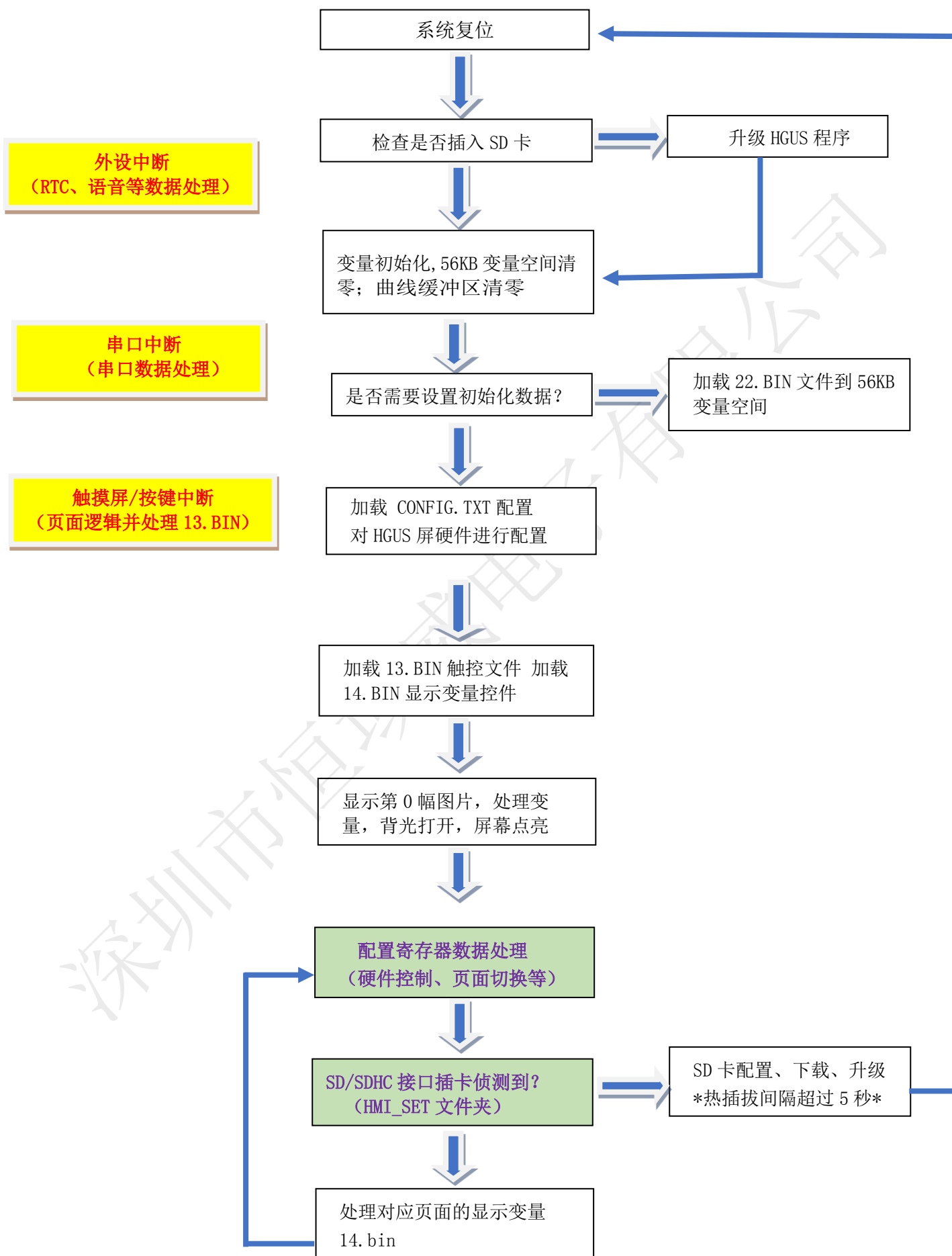
超长整数：-9223372036854775808到9223372036854775807

小数采用定点小数表示，用户自定义小数位数，比如 0x4D2 (1234)，规定小数为 2 位时，表示 12.34。

HGUS 屏使用 65K 色颜色系统，调色板定义如下：

HGUS 屏 使用的65K设调色板位定义																
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Define	R4	R3	R2	R1	R0	G5	G4	G3	G2	G1	G0	B4	B3	B2	B1	B0
	红色 0xF800					绿色 0x07E0						蓝色 0x001F				

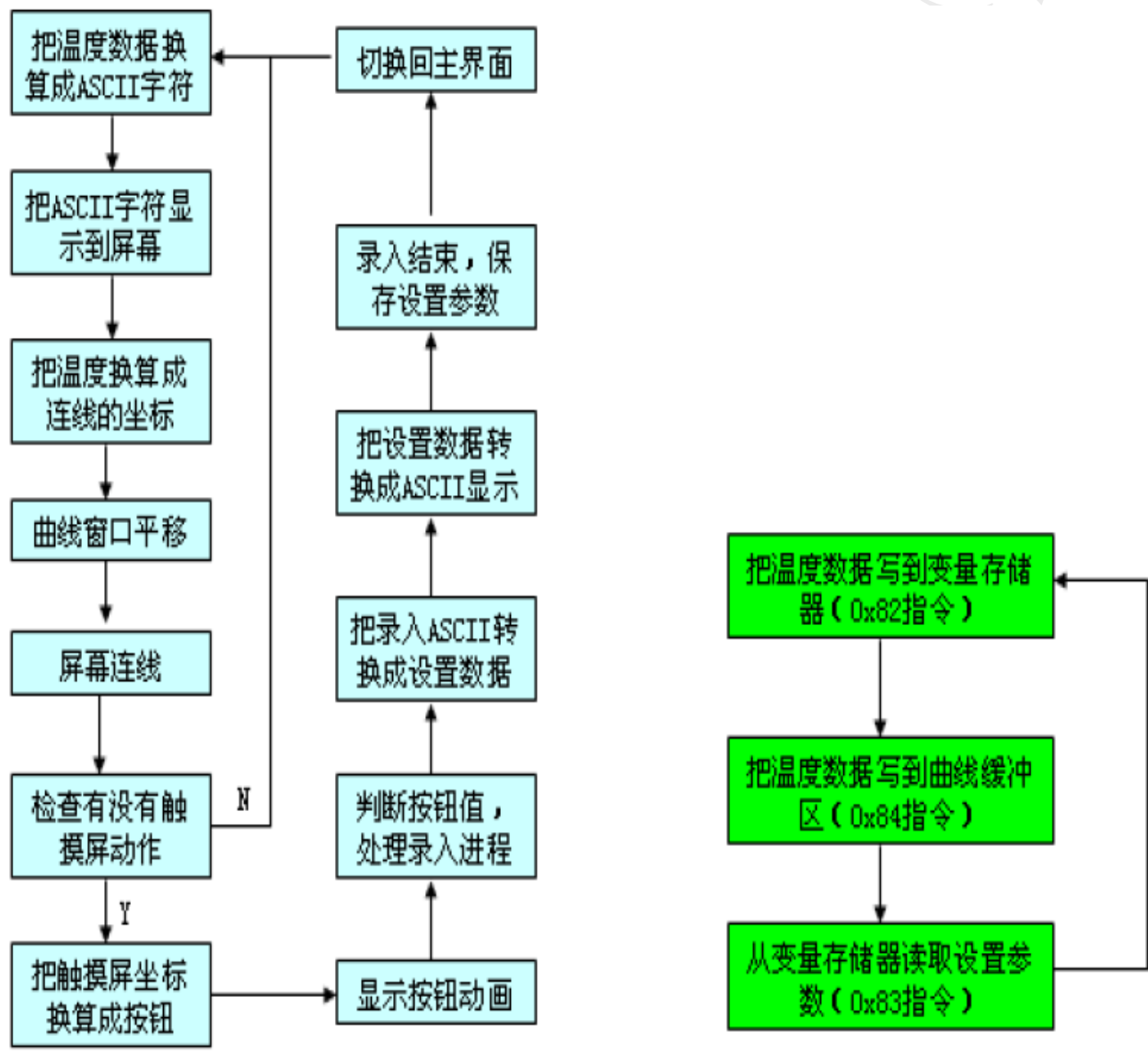
1.4 HGUS软件处理流程图



1.5 HGUS 屏开发流程

与传统的 LCM 通过时序或者指令控制显示不同，HGUS 屏采用直接变量驱动显示方式，所有的显示和操作都是基于预先设置好的变量配置文件来工作的。两种不同的工作方式导致用户应用时的软件架构和二次开发难度完全不同。

举例，假设做一个简单的触摸屏温控仪，要在当前页面显示测量温度，点击触摸屏切换到设置页面进行参数设置。两种不同开发方式下的软件流程图如下：



基于 LCM 开发温控仪流程图

基于 HGUS 屏开发温控仪流程图

上面的例子，只是一个两个参数、两个页面的最简单 GUI，如果对于实际应用中稍微“高级”一点的产品，几十个参数，几十个页面，还要考虑动画、图标等等吸引眼球的 GUI，前一种方式需要一个优秀工程师加班加点干 1 年，而使用 HGUS 开发，1-2 个工程师（可以并行协同做）1-2 天就搞定了。

在有些中小型工业自动化项目应用中，当整个系统由一些相对独立、功能完善的组件（比如支持 Modbus 协议的温控仪或其它二次仪表）构成时，用户可以直接把 HGUS 屏当成主机使用，用 485 网络把设备和 HGUS 屏组成一个网络，

总的来说，HGUS 屏是基于配置文件来工作的，所以整个开发过程也就是通过 PC 软件辅助设计完成变量 配置文件的过程，基本开发流程如下：

第 1 步：变量规划

推荐客户开发过程中用 Excel 表格来记录、整理好变量分配记录，便于将来的修改、升级维护。

第 2 步：界面设计 利用 PS（或者其它绘图软件）进行界面及界面相关元素（图标、字库）设计。设计过程中，请选择调色板系统为 65K 色，确保最终显示效果和设计效果一致。如果想让您的产品至少看起来很有价值，建议委托专业美工或者工业设计公司来设计 UI 和相关界面，一般收费在 150 人民币/1 个页面左右。

第 3 步：界面配置 利用我司提供的工具软件进行界面的配置，生成触控配置文件(13.BIN)和变量配置文件(14.BIN)。

第 4 步：测试修改 把配置文件、图片、字库、图标库等借助 SD 卡下载到 HGUS 屏，进行界面测试和修改（第 2-3 步）。把串口连上用户 MCU 系统，进行数据联调。

第 5 步：定版归档 定版后，把配置文件、图片、字库、图标库等 HGUS 屏涉及的文件保存在一张 SD 卡转生产即可。

2 HGUS屏配置

2.1 SD 配置接口

HUGS 屏的所有硬件参数设置和资料下载，都通过屏上的SD 接口来完成，文件必须使用**FAT32**文件格式。

第 1 步：将 SD 卡连接 PC，或将装有 SD 卡的读卡器连接 PC，待 PC 正确识别后出现磁盘分区,如下图：

第一次使用 SD 卡前，推荐先格式化一次，流程如下：



第 2 步：鼠标点击所属磁盘分区，点击右键，菜单出现，选择格式化并设置，如下图：



第 3 步：点击开始，出现提示，选择确定，并等待完成至出现提示。正确提示如下图：



使用SD卡下载数据的流程

第一步：把上位机生成的 **HMI_SET** 文件夹内的所有文件复制到 **SD卡根目录**。如下图



第二步：将已拷贝好文件包的 SD 卡插入 HGUS 屏的 SD 卡接口上，HGUS 屏检测到 SD 卡后，会显示蓝屏，提示用户检测到 SD 卡，然后开始屏参数配置，或将数据下载到屏上的Flash中；完成后，会有蜂鸣器会发出提示音“滴”，下载结束

注意：SD卡在两次热拔插之间，间隔至少5秒，不然HGUS屏会认为是同一张卡而不会启动SD卡操作。

2.2 SD 下载文件格式说明

为防止误操作，HGUS 屏对SD 卡配置文件有严格的命名和格式要求，不然会导致出错。

2.2.1 图片下载

图片文件必须是和 HGUS 屏分辨率相同的 **24位色BMP格式** 文件，文件的命名必须是表示图片存储位置的阿拉伯数字开头。

比如，要把一副图片用SD卡存储到 HGUS 屏的第20张图片存储位置，图片文件可以命名成“20测试.BMP”、“20.BMP”或者“020测试.BMP”，但不能命名成“测试20.BMP”。不同分辨率 HGUS 屏最大存。

储图片数量如下表所示：

类 别	图片空间 最大容量	数据库间 最大容量	不同分辨率HGUS屏最大存储图片数量						
			320*240	480*272	640*480	800*480	800*600	1024*600	1024*768
标 准	128MB	32MB	650	380	160	130	100	80	60
扩展到1GB	932MB	450MB	3728	3728	1242	1242	932	745	621
扩展到2GB	1896MB	960MB	7584	7584	2528	2528	1896	1516	1264
出厂程序默认用 bmp 格式图片，客户有大量图片需求的再用 jpg 格式程序或是联系我司相关人员对产品升级									

2.2.2 字库下载

HGUS 屏一共有32MB字库空间，分割成 128 个容量固定为256KB的字库空间。和下载图片类似，字库文件的命名也必须是表示字库存储位置（0-127）的阿拉伯数字开头。下载的字库文件包括字库、输入法词库、配置文件、图标文件；

字库文件格式说明如下表所示：

SD 卡文件格式说明			
文件类型	命名规则	举 例	说 明
字库文件	字库存储位置+（可选的）文件名.BIN/HZK/DZK	32_GBK12 汉字库.DZK	可用 TS3 字库提取软件生成
图 标 库	字库存储位置+（可选的）文件名.IC0	41 图标库.IC0	HGUS 组态软件生成
专用 ASCII 字库	0*.HZK，固定位置 0-11	0_HEROWAY_ASC.HZK	出厂预装
输入法词库	12*.BIN，固定位置 12	12_PY_HGUS.BIN	出厂预装
触控配置	13*.BIN，固定位置 13	13 触控文件.BIN	HGUS 组态软件生成
变量配置	14*.BIN，固定位置 14-21	14 变量文件.BIN	HGUS 组态软件生成
变量初始化	22*.BIN，固定位置 22	22 变量初始化.BIN	HGUS 组态软件生成
0-23 号字库（6MB）被 HGUS 系统预留作为将来升级使用，用户字库从 24 位置开始使用			

2.2.3 CONFIG.TXT配置文件

CONFIG.TXT 文件采用 windows 的文本文档格式，类似脚本语言的方式来描述参数寄存器，每一行描述一个参数(格式必须是R?=HH 其中?是寄存器序号，HH是寄存器配置值的16进制（HEX）值，必须大写)，不用 的参数可以不写，如下图所示。

R1=07 ;波特率,07=115200

R2=0C ;Sys_CFG

R3=5A ;

R6=3F ;

R7=10 ;

R8=B4 ;

RA=A5 ;

STOP_OS ;

RC=00 ;

上例中，RA=A5 表示把RA寄存器内容配置成0xA5。

不能写成ra=5a或者RA=5a，并且每个寄存器配置后必须换行写下一个寄存器配置。 CONFIG.TXT文件相关参数说明请参考“2.3 HGUS 屏硬件参数配置”。

2.2.4声音文件下载

有些 HGUS 屏（具体看硬件规格书说明）支持128段语音播放，需要事先下载声音文件存储到屏里面。和下载字库类似，声音文件的命名也必须是表示声音存储位置（0-127）的阿拉伯数字（比如 12 请刷卡.WAV）声音文件的后缀是.WAV，采用32KHz采样，16bit 单声道WAV文件格式。声音文件存储在 FLASH 最后 8MB 空间。

2.2.5 HGUS应用程序升级

HGUS屏支持SD卡升（降）级应用程序，把应用程序（boot.BIN）放到SD卡根目录下HMI_SET目录即可。上电重启HGUS屏，程序自动完成升级。

2.2.6 用户数据库SD卡导出

用户数据库是在HGUS屏的图片存储器中拿出一块空间进行数据存储，空间大小和位置由用户控制，不同内核其Flash大小不同，导致图片存储器空间和可用数据库空间也不相同，如下表所示：

类别	图片空间 最大容量	数据库间 最大容量	不同分辨率HGUS屏最大存储图片数量						
			320*240	480*272	640*480	800*480	800*600	1024*600	1024*768
标准	128MB	32MB	650	380	160	130	100	80	60
扩展到1GB	932MB	450MB	3728	3728	1242	1242	932	745	621
扩展到2GB	1896MB	960MB	7584	7584	2528	2528	1896	1516	1264

出厂程序默认用 bmp 格式图片，客户有大量图片需求的再用 jpg 格式程序或是联系我司相关人员对产品升级

用户数据库导出方法如下：

◆先计算出导出数据库的起始页位置

假设要导出的数据库首地址为ADR，那么地址的最高两个字节+256 即为对应的SD卡导出页ID。

◆用起始页位置命名创建一个和导出数据库等大（对齐到128KB）的DAT文件：

文件：字库起始页位置+（可选的）文件名.DAT

◆把这个DAT文件放到SD卡HMI_SET文件夹下面，插入HGUS屏SD卡接口，HGUS屏会自动把指定的数据库内容读取出来覆盖SD卡上的DAT文件。

举例，

假设要导出数据库空间0x00 10 00 00 到0x00 17 FF FF 共1MB（512KW）数据，那么对应的SD导出页ID是：0x00 10+256=272

在SD卡的HMI_SET目录下放置1个大小是1MB的文件 272数据库记录.DAT（或者其它272***.DAT的文件名都可以），把SD卡插入HGUS屏SD/SDHC接口即可把数据库内容导出。

关于用户数据库的详细说明请见“4.5 数据库读写”。

2.3 HGUS屏参数配置

HGUS屏参数配置通过在CONFIG.TXT文件中写好寄存器参数，然后用SD卡下载到HGUS屏中实现。

2.3.1 屏幕物理分辨率设置 (R0)

显示屏物理分辨率由R0寄存器设置，如下表所示。

R0 设置	分辨率设置 (H*V)	典型 HGUS 屏	备 注
00	320*240	特殊定制屏	
01	480*272	HYW043E01	
02	640*480	HYW056E02	
03	800*480	HYW070EH01	
04	800*600	HYW080E03	
05	1024*600	HYW102E02	
06	1024*768	特殊定制屏	
R0 寄存器出厂已经设置好，用户无须再配置。R0 配置错误将导致显示异常			

2.3.2 串口波特率设置 (R1、R5、R9)

HGUS屏用户接口的串口波特率由R1、R5、R9寄存器设置。

◆当R1取值在00-10时，R5、R9无效，可以选择17档固定波特率之一，如下表(波特率单位为Kbps)：

R1	00	01	02	03	04	05	06	07
波特率	1.2	2.4	4.8	9.6	19.2	38.4	57.6	115.2

HGUS屏出厂波特率预设值 R1=7，波特率为115200bps。

2.3.3 串口通信帧头设置 (R3、RA)

HGUS屏的串口数据帧由5个数据块组成，如下表：

数据块	1	2	3	4	5
定义	帧头	数据长度	指令	数据	指令和数据的 CRC 校验
数据长度	2	1	1	N	2
说明	R3:RA 定义	数据长度包括指令、数据和校验	0x80-0x84		R2.4 决定是否启用

通信帧头的设置主要达到以下两个目的：

- (1) 用于串口数据帧的识别和同步；
- (2) 多台HGUS屏并联工作时，把帧头做为设备地址加以区分；

假设设置R3=AA RA=BB，那么串口指令必须以0xAA 0xBB开头（比如读寄存器指令 AA BB 03 81 00 10），HGUS屏才会接收。HGUS屏出厂通信帧头预设值 R3=5A RA=A5，帧头为0x5A A5。

2.3.4 软件工作模式配置寄存器 (R2、RC)

R2、RC寄存器按位 (bit) 定义，用于配置HGUS屏软件工作模式，如下表所示 (亮光表示出厂设置值)。

◆ R2 寄存器定义

位	权重	定义	说明
. 7	0x80	VDS	0=正常显示 1=偏转 90° 显示
. 6	0x40	未定义	
. 5	0x20	TP_LED	0=背光不受触摸屏状态控制 1=背光受触摸屏状态控制，控制参数由 R6、R7、R8 寄存器设定
. 4	0x10	FCRC	0=不启用串口通信的 CRC16 帧校验 1=启用串口通信的 CRC16 帧校验
. 3	0x08	TPSAUTO	0=触摸屏录入参数后不自动上传 (用户查询) 1=触摸屏录入参数后是否自动上传到串口由相应触控变量的配置决定
. 2	0x04	L22_Init_En	0=56KB 变量存储器上电初始化数据为 0x00 1=56KB 变量存储器上电初始化数据由 22 字库文件加载
. 1	0x02	未定义	
. 0	0x01	未定义	

◆ RC (AUX_CFG 配置字) 说明

位	权重	定义	说明
. 7	0x80	系统保留	
. 6	0x40	系统保留	
. 5	0x20	未定义	
. 4	0x10	PAGE128_EN	0=每页显示变量数目为 64 个，须在 HGUS PC 上位机上对应选择 64 变量模式 1=每页显示变量数目为 128 个，须在 HGUS PC 上位机上对应选择 128 变量模式
. 3	0x08	未定义	
. 2	0x04	未定义	
. 1	0x02	未定义	
. 0	0x01	未定义	

2.3.5 触摸屏控制背光 (R2.5 R6 R7 R8)

当设置R2.5=1 时，背光亮度将受触摸屏状态控制。

R#	取值范围	说 明
R6	0x00-0x40	触摸屏控制背光启动后，点击触摸屏后背光点亮亮度。
R7	0x00-0x40	触摸屏控制背光启动后，一段时间不点击触摸屏，背光关闭的亮度
R8	0x01-0xFF	触摸屏控制背光启动后，触摸屏背光点亮时间，单位为 1.0 秒

举例，设置 R2.5=1 R6=40 R7=10 R8=1E，30 秒（ 0x1E ）不点击触摸屏，背光亮度将自动降低到 0x10（ 25% ）

点击触摸屏后，背光亮度将自动调节到 0x40（ 100% ）。

2.3.6触摸屏校准

HGUS屏有2种方式进入触摸屏校准模式。

➤ 校准方式 1

长按触摸屏不放开，在开机等待 5 秒则进入触摸屏校准模式，；

➤ 校准方式 2

通过串口向0xEA寄存器写入0x5A将启动一次触摸屏校准过程。

3 串口操作

恒域威 HGUS 屏采用异步、全双工串口 (UART) , 串口模式为 8n1 , 即每个数据传送采用 10 个位 : 1 个起始位 , 8 个数据位 , 1 个停止位。 串口波特率通过 SD 卡来配置

串口的所有指令或者数据都是 16 进制 (HEX) 格式 ; 对于字型 (2 字节) 数据 , 总是采用高字节先传送 (MSB) 方式。比如 0x1234 传送时先传送 0x12。

3.1数据帧架构

恒域威HGUS屏的串口数据帧由5个数据块组成，如下表所述：

数据块	1	2	3	4	5
定义	帧头	数据长度	指令	数据	指令和数据的 CRC 校验
数据长度	2	1	1	N	2
说明	CONFIG.TXT 配置文件的 R3: RA 定义	数据长度包括指令、数据和校验	0x80-0x84		CONFIG.TXT 配置文件的 R2.4 决定是否启用
举例	5A A5	05	81	00 10	20 24

CRC校验不包括帧头和数据长度，仅针对指令和数据，采用ANSI CRC-16($X^{16}+X^{15}+X^2+1$) 格式。

3.2指令集

HGUS屏采用变量驱动模式工作，屏的工作模式和GUI状态完全由数据变量来控制。相应的，串口指令也只需要对变量进行读、写即可，指令集非常简单，一共只有5条指令。

功能	指令	数 据	说 明
访问寄存器	0x80	下发：寄存器地址 (0x00-0xFF) + 写入数据	指定地址写寄存器数据
	0x81	下发：寄存器地址 (0x00-0xFF) + 读取字节长度 (0x00-0xFF)	指定地址开始读指定字节长度的寄存器数据
		应答：寄存器地址 (0x00-0xFF) + 字节数据长度 + 读取的寄存器数据	读寄存器的 HGUS 屏应答
	HGUS 屏有 256Byte 的寄存器，主要用于硬件操作的软件接口，按照字节 (Byte) 寻址操作。		
访问变量存储器	0x82	下发：变量存储器地址 (0x0000-0x6FFF) + 写入的变量数据	指定变量地址开始写入数据串 (字数据) 到变量存储区
	0x83	下发：变量存储器地址 (0x0000-0x6FFF) + 读取变量数据字长度 (0x00-0x7F)	从变量存储区指定地址开始读入 RD_LEN 长度字数据
		应答：变量存储器地址 + 变量数据字长度 + 读取的变量数据	读数据存储器的 HGUS 屏应答
	HGUS 屏有 28K word (56K Byte) 的变量存储器，主要用于 GUI 变量数据存储，按照字 (Word) 寻址操作。		
	0x84	CH_Mode (Byte) + DATA0 (Word) + ... + DATA _n	写曲线缓冲区数据。 CH_Mode 定义了后续数据的通道排列顺序： CH_Mode 的每个位对应一个通道： CH_Mode.0 对应 0 通道，.7 对应 7 通道 对应位置 1 表示对应的通道数据存在 对应位置 0 表示对应的通道数据不存在 数据按照低通道数据在前排列。 例 CH_Mode=0x83 (10000011B), 表示后续数据格式为 (通道 0 + 通道 1 + 通道 7) + ... + (通道 0 + 通道 1 + 通道 7)
	HGUS 屏有一个 8K Word，可以存储 8 条曲线趋势图的曲线缓冲区，用于用户简单、快速显示曲线。曲线缓冲区的数据都是 16 位无符号数。		

3.3 串口CRC校验C程序参考

HGUS屏的CRC校验采用ANSI CRC-16($X^{16}+X^{15}+X^2+1$) 格式, 相应C程序参考如下:

unsigned char

```
CRCTABH[256]={0x00,0xC1,0x81,0x40,0x01,0xC0,0x8  
0,0x41,0x01,0xC0  
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  
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};
```

unsigned char

```
CRCTABL[256]={0x00,0xC0,0xC1,0x01,0xC3,0x03,0x02  
,0xC2,0xC6,0x06  
0x07,0xC7,0x05,0xC5,0xC4,0x04,0xCC,0x0C,0x0D,0xC  
D  
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};

```

```

unsigned char index,crch,crcl;

```

```

    crch=0xff;
    crcl=0xff;
    for(i=0;i<j;
        i++)
    { index=crch^txdat[i];    //txdat[i] 是发送数据
      crch=crcl^CRCTABH[
index];
      crcl=CRCTABL[index];}

```

4 HGUS寄存器

4.1 HGUS寄存器一览表

寄存器地址	定 义	R/W	字长	说 明
0x00	Version	R	1	HGUS 版本号, BCD 码表示, 0x10 表示 V1.0
0x01	LED_SET	W	1	LED 亮度控制寄存器, 0x00-0x40
0x02	BZ_TIME	W	1	蜂鸣器鸣叫控制寄存器, 单位为 10mS
0x03	PIC_ID	R/W	2	读: 当前显示页面 ID 写: 切换到指定页面
0x05	TP_Flag	R/W	1	0x5A=触摸屏坐标有更新; 其它=触摸屏坐标未更新。 用户读取数据后未清零本标记, 则触摸屏数据不再更新。
0x06	TP_Status	R	1	0x01=第一次按下 0x03=一直按压中 0x02=抬起 其它=无效
0x07	TP_Position	R	4	触摸屏按压坐标位置: X_H:L Y_H:L
0x0B	TP_Enable	R/W	1	0x00=触控不启用 其它=触控启用(上电默认为 0xFF)。
0x0C-0x0F	RUN_TIME	R	4	上电后运行时间, BCD码时分秒, 其中小时为两个字节, 最大 9999:59:59
0x10-0x1C	R0-RC	R/W	13	SD 卡配置寄存器映射; 当 1D 寄存器配置时, 11-1C 寄存器是改写 R1-RC 数据
0x1D	CONFIG_EN	W		写入 0x5A, R1-RC 重新设置并保存; 写入 0xA5, 只重新设置 R0-RC, 不保存。
0x1E	LED_STA	R	1	背光当前亮度返回值
0x1F	RTC_COM_ADJ	W	1	0x5A 表示用户串口改写了 RTC 数据, HGUS 修改 RTC 后清零。
0x20	RTC_NOW	R/W	16	YY:MM:DD:WW:HH:MM:SS
0x30-0x3F	保留		16	未定义
0x40	En_Lib_OP	R/W	1	0x5A 表示用户申请进行读字库存储器操作, HGUS 操作完后清零。 每个 HGUS 周期执行一次读操作。 0xA0: 把指定字库空间的数据读入变量存储器空间。 指定的字库空间, 0x40-0x7F, 每个字库 128KW, 对应最大 Flash 空间为 8MW 指定字库空间的数据操作首(字)地址, 0x00:00:00-0x01:FF:FF 指定变量存储器空间的数据操作首(字)地址, 0x00:00-0x6F:FF 数据操作的(字)长度, 0x00:01-0x6F:FF。
0x41	Lib_OP_Mode	W	1	
0x42	Lib_ID	W	1	
0x43	Lib_Address	W	3	
0x46	VP	W	2	
0x48	OP_Length	W	2	
0x4A	Timer0	R/W	2	16bit 软件定时器, 单位为 4mS, 自减到零停止。
0x4C	Timer1	R/W	1	8bit 软件定时器, 单位为 4mS, 自减到零停止。
0x4D	Timer2	R/W	1	8bit 软件定时器, 单位为 4mS, 自减到零停止。
0x4E	Timer3	R/W	1	8bit 软件定时器, 单位为 4mS, 自减到零停止。

0x4F	Key_code	W	1	用户键码，用于触发 0x13 触控文件；0x01-0xFF，0x00 表示无效。HGUS 处理键码后会自动清零键码寄存器。
0x50	Play_Music_Set	W	3	0x5A:Play_Strat:Play_Num，音乐播放设定值。 Play_Start 为播放起始段，Play_Num 为连续播放段数（0x00 将停止播放）。
0x53	Volume_Adj	W	2	写入 0x5A:VOL 将调整播放音乐的音量，音量=VOL/64，上电默认值是 0x40。
0x55	保留		1	未定义
0x56	En_DBL_OP	R/W	1	0x5A 表示用户申请进行数据库存储器操作，HGUS 操作完后清零。每个 HGUS 周期执行一次数据库读或写操作。
0x57	OP_Mode	W	1	0x50：把变量存储器空间数据写入数据库空间。 0xA0：把数据库空间的数据读入变量存储器空间。
0x58	DBL_Address	W	4	数据库空间地址，0x00:00:00:00-1D:FF:FF:FF，最大 480MW（960MB，取决于 内核 Flash 情况）数据库空间。数据库从物理存储空间的第 64MB 开始存储，和 图片存储器空间有重合，每 1Byte 数据库存储器占据 2Byte 物理存储器。使用 SD 卡导出数据库时，每个字库大小为 64KW（128KB），编号从 256 开始，960MB 数据库对应字库 ID 范围为 256-7935，每个字库写寿命为 10 万次。读写时，HGUS 会自动处理跨字库的情况。
0x5C	VP	W	2	指定变量存储器空间的数据库操作首（字）地址，0x00:00-0x6F:FF
0x5E	OP_Length	W	2	数据库操作的（字）长度，0x00:01-0x6F:FF。
0x60-0xDF	保留			
0xE0-0xE8	保留			
0xE9	Scan_Status	R		0x01=触摸屏录入状态 0x00=触摸屏未处于录入状
0xEA	TPCal_Triger	W		写入 0x5A 启动一次触摸屏校准，校准完成后会被 HGUS 清零。
0xEB	Trendline_Clear	W		写入特殊定义的数值以清除对应的曲线缓冲区数据。 0x55:清除全部 8 条曲线缓冲区数据； 0x56-0x5D：分别清除 CH0-CH7 通道的曲线缓冲区数据。 曲线缓冲区数据清除后，本寄存器会被 HGUS 清零。
0xEC-0xED	保留			
0xEE-0xEF	Reset_Triger	W		写入 0x5AA5 导致 HGUS 屏软件复位一次。
0xF0-0xFF	保留			

4.2 读写RTC

0x1F	RTC_COM_ADJ	W	1	0x5A表示用户串口改写了RTC数据，HGUS修改RTC后清零。
0x20	RTC_NOW	R/W	16	YY:MM:DD:WW:HH:MM:SS

串口读取RTC

0x20寄存器开始保存了当前RTC值，使用0x81指令读取。

读取日历

(YY:MM:DD:WW:HH:MM:SS) : 5A A5 03 81 20 07

读取时间 (HH:MM:SS) : 5A A5 03 81 24 03

串口修改（写）RTC

用0x80指令改写0x1F寄存器为0x5A，并给0x20开始的寄存器写入需要修订的时间，即改写了RTC。 举例，把RTC设置为2013-11-08 18:56:00，串口下发

5A A5 0A 80 1F 5A 13 11 08 00 18 56 00

改写RTC时，只需要改写公历的年、月、日、时、分、秒即可，星期（00）和农历HGUS会自动换算。

4.3 字库读取

0x40	En_Lib_OP	R/W	1	0x5A表示用户申请进行读字库存储器操作，HGUS操作完后清零。 每个HGUS周期执行一次读字库操作。
0x41	Lib_OP_Mode	W	1	0xA0：把指定字库空间的数据读入变量存储器空间。
0x42	Lib_ID	W	1	指定字库空间 ,0x40-0x7F ,每个字库128KW ,对应最大Flash空间为8MW (16MB)。
0x43	Lib_Address	W	3	指定字库空间的数据操作首（字）地址，0x00:00:00-0x01:FF:FF
0x46	VP	W	2	指定变量存储器空间的数据操作首（字）地址，0x00:00-0x6F:FF
0x48	OP_Length	W	2	数据操作的（字）长度，0x00:01-0x6F:FF。

HGUS的第64-127字库(64个字库 ,16MB) ,可以通过串口指令操作 ,把字库数据读取到变量存储器中(如 果用户系统需要使用，可以使用0x82指令再从变量存储器中读取)。

举例，从第80号字库的0x 00 00 00 地址开始读取 4KW（0x10 00）数据到变量存储器 0x10 00 开始 的位置，串口下发指令：5A A5 0C 80 40 5A A0 50 00 00 00 10 00 10 00

注意，读取数据不能超过字库空间，即Lib_Address+OP_Length<= 0x02 00 00。

4.4 128段音乐播放

0x50	Play_Music_Set	W	3	0x5A:Play_Strat:Play_Num, 音乐播放设定值。 Play_Start为播放起始段, Play_Num为连续播放段数 (0x00将停止播放)。
0x53	Volume_Adj	W	2	写入0x5A:VOL将调整播放音乐的音量, 音量=VOL/64, 上电默认值是0x40。

某些HGUS屏自带128段 (每段1.024秒) 音乐播放功能, 通过SD卡把音乐 (32K采样 16bit单声道 WAV 文件) 下载到屏里面后, 可以用 0x80 指令写相关寄存器控制音乐播放和进行音量调节。

举例, 一段提示音 (比如 “欢迎光临深圳恒域威”) 长度为3.5秒, 保存在第6段语音开始位置, 占据的语音段是第6-9, 一共4段语音, 要以100%音量播放这段提示音, 串口下发:

5A A5 07 80 50 5A 06 04 5A 40 要停止当前语音播放, 只需要把播放指令

中的播放段数设置为 0x00 即可, 比如 5A A5 05 80 50 5A 06 00

要把音量提升到150% ($64 \times 1.5 = 96$ 0x60), 串口下发

指令: 5A A5 04 80 53 5A 60 语音播放过程中, 未保

存语音的空白段将直接被 “略过”

4.5 数据库读写

0x56	En_DBL_OP	R/W	1	0x5A表示用户申请进行数据库存储器操作, HGUS操作完后清零。 每个HGUS周期执行一次数据库读或写操作。
0x57	OP_Mode	W	1	0x50: 把变量存储器空间数据写入数据库空间。 0xA0: 把数据库空间的数据读入变量存储器空间。
0x58	DBL_Address	W	4	数据库空间地址, 0x00:00:00:00-1D:FF:FF:FF, 最大480MW (960MB, 取决于内核Flash情况) 数据库空间。数据库从物理存储空间的第64MB开始存储, 和图片存储器空间有重合, 每1Byte数据库存储器占据2Byte物理存储器。使用SD卡导出数据库时, 每个字库大小为64KW (128KB), 编号从256开始, 960MB 数据库对应字库ID范围为256-7935, 每个字库写寿命为10万次。读写时, HGUS会自动处理跨字库的情况。
0x5C	VP	W	2	指定变量存储器空间的数据库操作首 (字) 地址, 0x00:00-0x6F:FF
0x5E	OP_Length	W	2	数据操作的 (字) 长度, 0x00:01-0x6F:FF。

用户数据库是在 HGUS 屏的图片存储器中拿出一块空间进行数据存储，空间大小和位置由用户控制，不同内核其 Flash 大小不同，导致图片存储器空间和可用数据库空间也不相同，如下表所示：

类 别	图片空间 最大容量	数据库间 最大容量	不同分辨率HGUS屏最大存储图片数量						
			320*240	480*272	640*480	800*480	800*600	1024*600	1024*768
标 准	128MB	32MB	650	380	160	130	100	80	60
扩展到1GB	932MB	450MB	3728	3728	1242	1242	932	745	621
扩展到2GB	1896MB	960MB	7584	7584	2528	2528	1896	1516	1264
出厂程序默认用 bmp 格式图片，客户有大量图片需求的再用 jpg 格式程序或是联系我司相关人员对产品升级									

用户数据库读写过程中，HGUS 对数据进行了加密和纠错操作，确保数据存储的可靠性。用户数据库在物理上是由若干个大小为64KW（128KB）的数据库页构成，每个页面写寿命是10万次（启动1次写操作减少1次写寿命），但读写操作中地址是连续的，不受分页影响，HGUS会自动处理分页问题。

数据库首地址（0x00 00 00 00，对应第64MB物理存储器）对应的图片ID和存储系数K1

分辨率	320*240	480*272	640*480	800*480	800*600	1024*600	1024*768
K1	1	1	3	3	4	5	6
PIC_ID	128	128	42-43	42-43	32	25-26	21-22
"128"表示如果从0x00地址开始使用数据库，则第128幅图片位置开始不能保存图片；“42-43”表示42、43都不能使用。							

图片空间到数据库空间的计算

假设有N（N大于上表中的PIC_ID）幅图片需要存储，那么数据库可以使用的最小首地址

为：数据库最小首地址=（N*K1）-128）*64*1024 已经取整到64KW(128KB)。

举例，480*272分辨率下，需要预留出200幅图片，那么数据库的最小起始地址Adr_Min：

$$\text{Adr_Min} = (200 * 1) - 128) * 64 * 1024 = 0x00\ 48\ 00\ 00$$

数据库空间数据 SD 卡导出

数据库内容可以使用SD卡导出，请参考“2.2.6 用户数据库SD卡导出”。

4.6 按键触发

0x4F	Key_code	W	1	用户键码，用于触发0x13触控文件；0x01-0xFF，0x00表示无效。 HGUS 处理键码后会自动清零键码寄存器。
------	----------	---	---	--

HGUS 屏没有键盘接口，但很多应用需要使用键盘或者按键操作。HGUS 的 0x4F 寄存器提供了用户使用键 盘来控制 HGUS 屏 GUI 触控进程(13 触控文件)的接口，使得用户只需把键码写入 0x4F 寄存器，就可以由HGUS 按照13文件的描述来处理相关GUI。

比如，在13触控文件中，在10号页面定义0xF1键码将进入参数录入界面，那么，当用户在10号页面下发送指令：5A A5 03 80 4F F1 就启动了一次按键触发，HGUS会自动进入参数录入界面。

按键触发和触摸屏触发是并行触发的，可以混在一起使用（即触摸屏界面下也可以同时使用按键）。

5 触控/键控配置文件（13.BIN）说明

触控配置文件由 N 条按照页面配置的触控指令组成，每条触控指令固定占用 16、32、48 或者 64 字节存储空间。一条触控指令由以下 6 部分组成：

序号	定义	数据长度	说 明
1	Pic_ID	2	页面 ID
2	TP_Area	8	触控按钮区域：左上角坐标（Xs,Ys），右下角坐标（Xe,Ye） 当 Xs=0xFFFF（启动按键语音伴音时是 0x5FFF）时，表示触发控制由 0x4F 寄存器的键码值触发，此时 Ys_H 为设定的触发键码值（Ys_L,Xe,Ye 值未定义，可任意写）；由键码值触发时，请把按钮按压效果设置为无效。
3	Pic_Next	2	目标切换页面，0xFF**表示不进行页面切换。
4	Pic_On	2	按钮按压效果图所处的页面，0xFF**表示没有按钮按压效果。
5	TP_Code	2	触控键码： 0xFF**表示无效的键码 0xFE**（或者 0xFD**）表示触控功能按键，比如 0xFE00 表示启动变量数据触摸屏录入。0xFE**的功能按键可以由 R2.3 设置成变量改变后是否自动上传，0xFD**的功能按 键始终禁止变量改变后自动上传。其它表示触控键码，用 ASCII 表示；比如 0x0031 表示按键 “1”。
6	TP_FUN	32	当 TP_Code=0xFE**时，用来对触控功能按键进行描述。

5.1触控/键控功能一览表

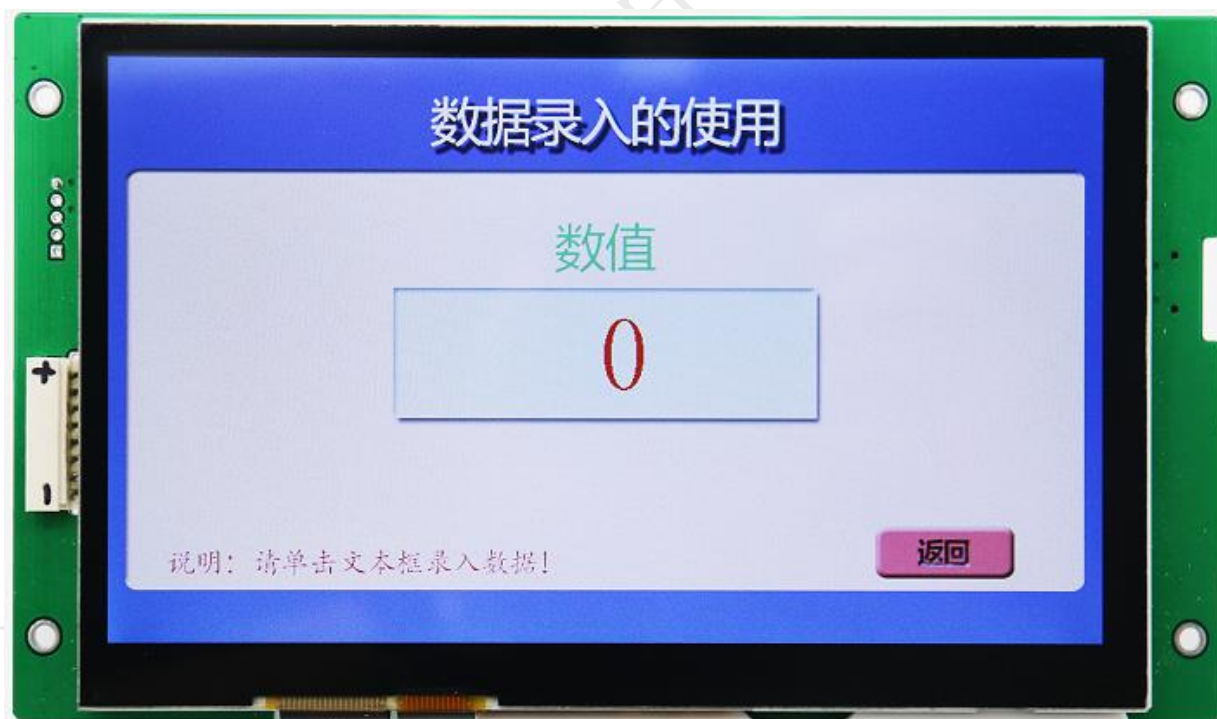
序号	触控键码	功 能	说 明
01	00	变量数据录入	录入整数、定点小数等各种数据到指定变量存储空间。
02	01	弹出菜单选择	点击触发一个弹出菜单，返回菜单项的键码。
03	02	增量调节	点击按钮，对指定变量进行+/-操作，可设置步长和上下限。 设置0-1范围循环调节可以实现栏目复选框功能。
04	03	拖动调节	拖拉滑块实现变量数据录入，可设置刻度范围。
05	04	RTC设置	HGUS屏触摸键盘设置RTC组件，需要完整录入公历 年月日时分秒。
06	05	按键值返回	点击按键，直接返回按键值到变量，支持位变量返回。
07	06	文本录入	文本方式录入各种字符，直接支持ASCII字符、GBK中文输入法录入； 修改字库和0#字库可以支持所有类似ASCII字符的8bit编码文本录入；
08	07_00	寄存器写到变量空间	提供了触摸屏改写寄存器空间的方法，来间接控制硬件。 比如把背光寄存器内容读取到变量，调节变量后再回写来调节背光亮度。
09	07_01	变量空间写到寄存器	
10	07_02	图像转成单色位图（纵向）	把指定区域的彩色位图转换成单色位图并保存在VP指定的变量区域。 主要用于当前屏幕显示内容的打印输出。
11	07_05	图像转成单色位图（横向）	
12	07_03	发送数据块到COM1	点击触摸屏，把指定VP区域的数据发送到用户串口（COM1）。
13	07_04	发送数据块到COM2	点击触摸屏，把指定VP区域的数据发送到扩展串口（COM2）。 COM2用于HGUS屏功能扩展，并没有引出。
14	07_06	发送触摸屏坐标到COM2	点击触摸屏，将把点击位置坐标发送到扩展串口（COM2）。 COM2用于HGUS屏功能扩展，并没有引出。
15	08	触摸屏按压状态数据返回	点击触摸屏，按照规定返回数据到变量或串口。
16	09	转动调节	转动旋钮实现变量数据录入，可设置刻度范围。

5.2 变量数据录入 (0x00)

地址	定义	数据长度	说 明
0x00	Pic_ID	2	页面 ID
0x02	TP_Area	8	触控按钮区域: (Xs, Ys) (Xe, Ye)
0x0A	Pic_Next	2	目标切换页面, 0xFF**表示不进行页面切换。
0x0C	Pic_On	2	按钮按压效果图所处的页面, 0xFF**表示没有按钮按压效果。
0x0E	TP_Code	2	0xFE00
0x10	0xFE	1	0xFE
0x11	*VP	2	录入数据对应的变量地址指针
0x13	V_Type	1	返回变量类型: 0x00=2 字节变量, 整数-32768 到 32767, 无符号整数 0-65535 0x01=4 字节变量, 长整数-2147483648 到 2147483647 无符号长整数 0-4294967295 0x02=*VP 高字节, 无符号数 0 到 255 0x03=*VP 低字节, 无符号数 0 到 255 0x04=超长整数(8 字节) -9223372036854775808 到 9223372036854775807
0x14	N_Int	1	录入的整数位数。比如录入 1234. 56, 则 N_Int=0x04。
0x15	N_Dot	1	录入的小数位数。比如录入 1234. 56, 则 N_Dot=0x02。
0x16	(x, y)	4	输入过程显示位置: 右对齐方式, (x, y) 是字符串输入光标的右上角坐标。
0x1A	Color	2	输入字体显示颜色。
0x1C	Lib_ID	1	显示使用的 ASCII 字库位置, 0x00=默认字库
0x1D	Font_Hor	1	字体大小, X 方向点阵数目
0x1E	Cusor_Color	1	光标颜色, 0x00=黑色 其它=白色
0x1F	Hide_En	1	0x00=输入遮挡, 显示为"*"; 其它, 输入直接显示
0x20	0xFE	1	0xFE
0x21	KB_Source	1	0x00=键盘在当前页面; 其它=键盘不在当前页面。
0x22	PIC_KB	2	键盘所在页面 ID, 仅当 KB_Source 不等于 0x00 时有效。
0x24	AREA_KB	8	键盘区域: (Xs, Ys) 为左上角、, (Xe, Ye) 为右下角坐标。 仅当 KB_Source 不等于 0x00 时有效。
0x2C	AREA_KB_Position	4	键盘在当前页面显示位置, 左上角坐标; 仅当 KB_Source 不等于 0x00 时有效。
0x30	0xFE	1	0xFE
0x31	Limits_En	1	0xFF: 表示启用输入范围限制, 输入越界无效(等同取消); 其它: 输入无范围限制。
0x32	V_min	4	输入下限, 4 字节(长整数或无符号长整数)。
0x36	V_max	4	输入上限, 4 字节(长整数或无符号长整数)。
0x3A-0x3F	保留	6	

输入过程中有效键码:

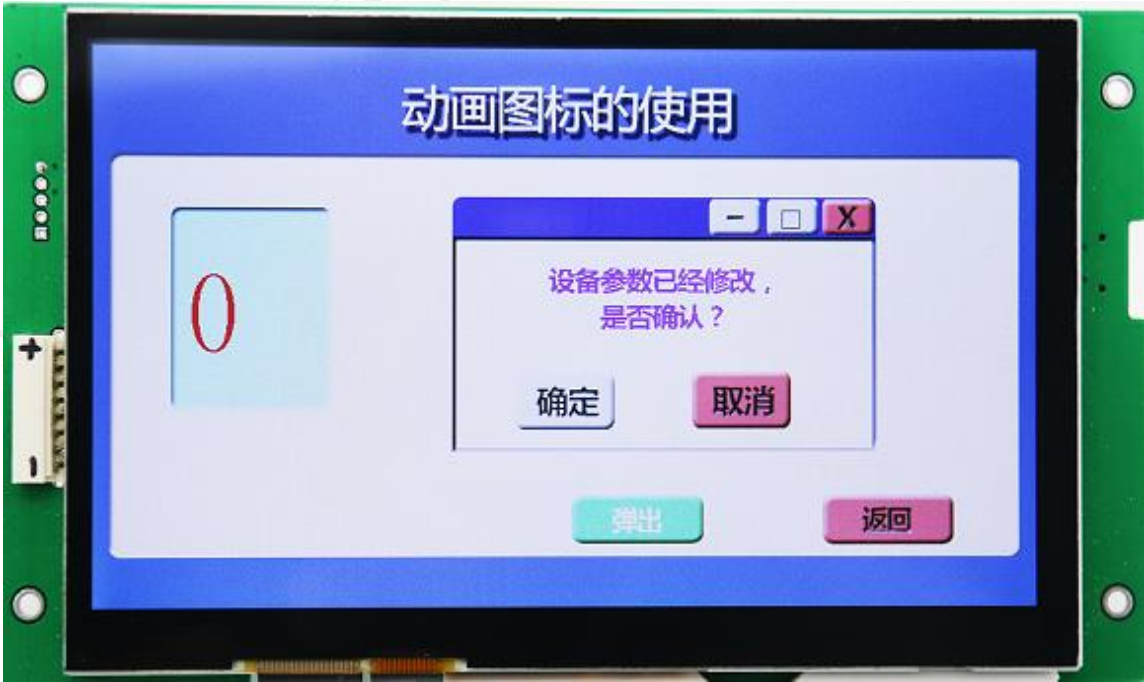
0x0030-0x0039, 0x002E (.), 0x002D(+/-), 0x00F0 (取消), 0x00F1 (确认), 0x00F2 (退格)。



5.3弹出菜单选择 (0x01)

地址	定义	数据长度	说明
0x00	Pic_ID	2	页面 ID
0x02	TP_Area	8	触控按钮区域 : (Xs,Ys) (Xe,Ye)
0x0A	Pic_Next	2	目标切换页面, 0xFF**表示不进行页面切换。
0x0C	Pic_On	2	按钮按压效果图所处的页面, 0xFF**表示没有按钮按压效果。
0x0E	TP_Code	2	0xFE01
0x10	0xFE	1	0xFE
0x11	*VP	2	变量地址指针, 返回数据由 VP_Mode 决定。
0x13	VP_Mode	1	0x00 = 把0x00**键码写入VP字地址 (整型数) ; 0x01 = 把**键码写入VP字地址的高字节地址 (VP_H) ; 0x02 = 把**键码写入VP字地址的低字节地址 (VP_L) ; 0x10-0x1F : 把**键码最低位(1bit)变量并写入VP字地址的指定位 (0x10修改 VP.0 , 0x1F修改VP.F)
0x14	Pic_Menu	2	弹出菜单的图片位置
0x16	AREA_Menu	8	菜单区域 : 左上角坐标 (Xs,Ys) , 右下角坐标 (Xe,Ye)
0x1E	Menu_Position_X	2	菜单在当前页面显示的位置: 左上角X坐标
0x20	0xFE	1	固定
0x21	Menu_Position_Y	2	菜单在当前页面显示的位置: 左上角Y坐标
0x23	NULL	13	写0x00

输入过程中有效键码 : 0x0000-0x00FF , 其中0x00FF为取消 (不选择参数直接返回) 。



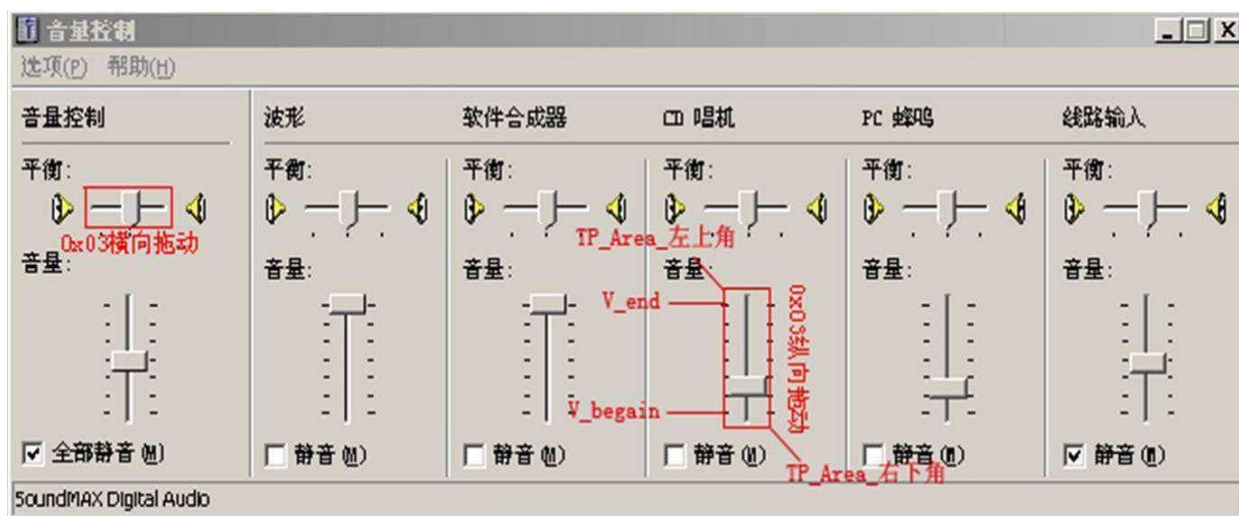
5.4 增量调节 (0x02)

地址	定义	数据长度	说 明
0x00	Pic_ID	2	页面 ID
0x02	TP_Area	8	触控按钮区域 : (Xs,Ys) (Xe,Ye)
0x0A	Pic_Next	2	目标切换页面, 0xFF**表示不进行页面切换。必须为 0xFF**。
0x0C	Pic_On	2	按钮按压效果图所处的页面, 0xFF**表示没有按钮按压效果。
0x0E	TP_Code	2	0xFE02
0x10	0xFE	1	0xFE
0x11	*VP	2	变量地址指针, 返回数据由 VP_Mode 决定。
0x13	VP_Mode	1	0x00 = 调节VP字地址 (整型数) ; 0x01 = 调节VP字地址的高字节地址 (1字节无符号数, VP_H) ; 0x02 = 调节VP字地址的低字节地址 (1字节无符号数, VP_L) ; 0x10-0x1F : 对VP字地址的指定位 (0x10对应VP.0, 0x1F对应VP.F) 进行调节, 调节范围必须设置为0-1。
0x14	Adj_Mode	1	调节方式: 0x00=-- 其它 = ++
0x15	Return_Mode	1	逾限处理方式: 0x00=停止 (等于门限) 其它 = 循环调节
0x16	Adj_Step	2	调节步长, 0x0000-0x7FFF
0x18	V_Min	2	下限: 2字节整数 (VP_Mode=0x01或0x02时, 仅低字节有效)
0x1A	V_Max	2	上限: 2字节整数 (VP_Mode=0x01或0x02时, 仅低字节有效)
0x1C	Key_Mode	1	0x00 : 按住按键时连续调节; 0x01 : 按住按键时只调节1次。
0x1D	NULL	3	写0x00



5.5 拖动调节 (0x03)

地址	定义	数量长度	说明
0x00	Pic_ID	2	页面 ID
0x02	TP_Area	8	触控按钮区域 : (Xs,Ys) (Xe,Ye)
0x0A	Pic_Next	2	目标切换页面, 0xFF**表示不进行页面切换。必须为 0xFF**。
0x0C	Pic_On	2	按钮按压效果图所处的页面, 0xFF**表示没有按钮按压效果。必须为 0xFF**。
0x0E	TP_Code	2	0xFE03
0x10	0xFE	1	0xFE
0x11	*VP	2	变量地址指针。
0x13	Adj_Mode	1	高4比特定义了数据返回格式： 0x0* = 调节VP字地址（整型数）； 0x1* = 调节VP字地址的高字节地址（1字节无符号数，VP_H）； 0x2* = 调节VP字地址的低字节地址（1字节无符号数，VP_L）。 低4bit定义了拖动方式： 0x*0=横向拖动， 0x*1=纵向拖动。
0x14	Area_Adj	8	有效调节区域：Xs,Ys,Xe,Ye；必须和TP_Area（触控区域）一致。
0x1C	V_begain	2	起始位置对应的返回值，整数。
0x1E	V_end	2	终止位置对应的返回值，整数。



5.6RTC 设置 (0x04)

地址	定义	数据长度	说明
0x00	Pic_ID	2	页面 ID
0x02	TP_Area	8	触控按钮区域 : (Xs,Ys) (Xe,Ye)
0x0A	Pic_Next	2	目标切换页面, 0xFF**表示不进行页面切换。
0x0C	Pic_On	2	按钮按压效果图所处的页面, 0xFF**表示没有按钮按压效果。
0x0E	TP_Code	2	0xFE04
0x10	0xFE	1	0xFE
0x11	0x00 00 00	3	0x00 00 00
0x14	(x,y)	4	输入过程显示位置,右对齐方式, (x,y)是字符串右上角坐标。
0x18	Color	2	输入字体显示颜色。
0x1A	Lib_ID	1	显示使用的ASCII字库位置, 0x00 = 默认字库
0x1B	Font_Hor	1	字体大小, X方向点阵数目
0x1C	Cusor_Color	1	光标颜色, 0x00=黑色 其它 = 白色
0x1D	KB_Source	1	0x00=键盘在当前页面; 其它 = 键盘不在当前页面
0x1E	PIC_KB	2	键盘所在页面ID, 仅当KB_Source不等于0x00时有效。
0x20	0xFE	1	0xFE
0x21	AREA_KB	8	键盘区域: 左上角坐标 (Xs,Ys), 右下角坐标 (Xe,Ye); 仅当KB_Source不等于0x00时有效。
0x29	AREA_KB_Position	4	键盘在当前页显示位置, 左上角坐标; 仅当KB_Source不等于0x00时有效
0x2D	NULL	3	写0x00



5.7 按键值返回 (0x05)

地址	定义	数据长度	说 明
0x00	Pic_ID	2	页面 ID
0x02	TP_Area	8	触控按钮区域 : (Xs,Ys) (Xe,Ye)
0x0A	Pic_Next	2	目标切换页面, 0xFF**表示不进行页面切换。
0x0C	Pic_On	2	按钮按压效果图所处的页面, 0xFF**表示没有按钮按压效果。
0x0E	TP_Code	2	0xFE05
0x10	0xFE	1	0xFE
0x11	*VP	2	变量地址指针
0x13	VP_Mode	1	0x00 = 返回键值保存在VP字地址 (整型数) ; 0x01 = 返回键值低字节保存在VP字地址的高字节地址 (VP_H) ; 0x02 = 返回键值低字节保存在VP字地址的低字节地址 (VP_L) ; 0x10-0x1F : 把返回键值的最低位 (1bit) 写入VP字地址的指定位 (0x10修改VP.0 , 0x1F修改VP.F)
0x14	Key_Code	2	返回键值。
0x16	NULL	10	写0x00

5.8 文本录入 (0x06)

输入文本键盘码表

在文本录入的触控文件中, 两字节键码的低字节表示普通键码, 高字节表示大写键码。典型的文本录入键盘定义如下表所示:

键码	普通	大写	键码	普通	大写	键码	普通	大写	键码	普通	大写
0x7E60	`	~	0x5171	q	Q	0x4161	a	A	0x5A7A	z	Z
0x2131	1	!	0x5777	w	W	0x5373	s	S	0x5878	x	X
0x4032	2	@	0x4565	e	E	0x4464	d	D	0x4363	c	C
0x2333	3	#	0x5272	r	R	0x4666	f	F	0x5676	v	V
0x2434	4	\$	0x5474	t	T	0x4767	g	G	0x4262	b	B
0x2535	5	%	0x5979	y	Y	0x4868	h	H	0x4E6E	n	N
0x5E36	6	^	0x5575	u	U	0x4A6A	j	J	0x4D6D	m	M
0x2637	7	&	0x4969	i	I	0x4B6B	k	K	0x3C2C	,	<
0x2A38	8	*	0x4F6F	o	O	0x4C6C	l	L	0x3E2E	.	>
0x2839	9	(0x5070	p	P	0x3A3B	;	:	0x3F2F	/	?
0x2930	0)	0x7B5B	[{	0x2227	'	"	0x2020	SP	SP
0x5F2D	-	_	0x7D5D]	}	0x0D0D	Enter	Enter			
0x2B3D	=	+	0x7C5C	\							

注: 文本键盘键码须小于0x80 (ASCII码)。0x0D键码录入会自动转换成0x0D 0x0A; 0x00和0xFF键码禁用。

键盘功能键码定义

键码	定义	说明
0x00F0	Cancel	取消录入返回, 不影响变量数据。
0x00F1	Return	确认录入返回, 录入文本保存到指定变量位置。
0x00F2	Backspace	向前 (退格) 删除一个字符。
0x00F3	Delete	向后删除 1 个字符。
0x00F4	CapsLock	大写锁定。如果启用, 对应按钮必须定义按钮按下的效果。
0x00F7	Left	光标前移一个字符; GBK 汉字录入中用于翻页。
0x00F8	Right	光标后移一个字符; GBK 汉字录入中用于翻页。

使用键盘（0x4F 寄存器保存的键码）做文本录入时，如果使用CapsLock 键，请把按钮的动画区域定义在需要提示“CapsLock”的区域；这样定义后，发送CapsLock 键时，屏幕的相应位置会自动显示“CapsLock” 的区域图标提示

5.8.1 ASCII 文本录入

地址	定义	数据长度	说 明
0x00	Pic_ID	2	页面 ID
0x02	TP_Area	8	触控按钮区域：(Xs,Ys)(Xe,Ye)
0x0A	Pic_Next	2	目标切换页面，0xFF**表示不进行页面切换。
0x0C	Pic_On	2	按钮按压效果图所处的页面，0xFF**表示没有按钮按压效果。
0x0E	TP_Code	2	0xFE06
0x10	0xFE	1	0xFE
0x11	*VP	2	变量地址指针
0x13	VP_Len_Max	1	文本变量最大长度，字（Word）数目，0x01-0x7B； 文本保存到指定地址时，自动在文本结束处加上0xFFFF作为结束符；录入的文本变量实际可能占用最大变量空间=VP_Len_Max + 1。
0x14	Scan_Mode	1	录入模式控制：0x00=重新录入 0x01=打开原来文本再修改
0x15	Lib_ID	1	显示使用的ASCII字库位置，0x00 = 默认字库
0x16	Font_Hor	1	字体大小，X方向点阵数目
0x17	Font_Ver	1	字体大小，Y方向点阵数目（Lib_ID = 0x00时，Y方向点阵数目必须为2*X）
0x18	Cusor_Color	1	光标颜色，0x00=黑色 其它 = 白色
0x19	Color	2	文本显示颜色。
0x1B	Scan_Area_Start	4	录入文本显示区域左上角坐标（Xs,Ys）
0x1F	Scan_Return_Mode	1	0x55:在*(VP-1)位置保存输入结束标记和有效数据长度： *(VP-1)高字节，输入结束标记：0x5A表示输入结束，输入过程为0x00。 *(VP-1)低字节，有效输入数据长度字节单位。 0x00:不返回输入结束标记和长度
0x20	0xFE	1	
0x21	Scan_Area_End	4	录入文本显示区域右下角坐标（Xe,Ye）
0x25	KB_Source	1	键盘页面位置选择：0x00=键盘在当前页面；其它 = 键盘不在当前页面。
0x26	PIC_KB	2	以下数据，仅当KB_Source不为0x00时有效。键盘所在页面ID
0x28	AREA_KB	8	键盘页面上键盘区域坐标：左上角（Xs,Ys）、右下角（Xe,Ye）
0x30	0xFE	1	
0x31	AREA_KB_Position	4	键盘区域粘贴在当前页面显示的位置，左上角坐标。
0x35	DISPLAY_EN	1	0x00：输入过程正常显示 0x01：输入过程显示"*"，用于密码输入
0x36	NULL	10	写0x00



5.8.2 GBK 汉字文本录入

地址	定义	数据长度	说 明
0x00	Pic_ID	2	页面 ID
0x02	TP_Area	8	触控按钮区域：(Xs,Ys) (Xe,Ye)
0x0A	Pic_Next	2	目标切换页面，0xFF**表示不进行页面切换。
0x0C	Pic_On	2	按钮按压效果图所处的页面，0xFF**表示没有按钮按压效果。
0x0E	TP_Code	2	0xFE06
0x10	0xFE	1	0xFE
0x11	*VP	2	变量地址指针
0x13	VP_Len_Max	1	文本变量最大长度，字（Word）数目，0x01-0x7B； 文本保存到指定地址时，自动在文本结束处加上0xFFFF作为结束符；录入的文本变量实际可能占用最大变量空间 = VP_Len_Max + 1。
0x14	Scan_Mode	1	录入模式控制：0x00=重新录入 0x01=打开原来文本再修改
0x15	Lib_GBK1	1	汉字字符显示使用的GBK字库ID，ASCII字符默认使用0x00字库。
0x16	Lib_GBK2	1	录入过程，汉字字符显示使用的GBK字库ID
0x17	Font_Scale1	1	Lib_GBK1字体大小，点阵数目
0x18	Font_Scale2	1	Lib_GBK2字体大小，点阵数目
0x19	Cusor_Color	1	光标颜色，0x00=黑色 其它=白色
0x1A	Color0	2	录入文本显示颜色。
0x1C	Color1	2	录入过程中文本显示颜色
0x1E	PY_Disb_Mode	1	录入过程中，拼音提示和对应汉字的显示方式： 0x00=拼音提示显示在上边，对应的汉字显示另起一行显示在下面； 拼音提示和汉字显示左对齐，行间距为Scan_Dis。 0x01=拼音提示显示在左边，对应的汉字提示在右边显示；汉字提示起始显示x位置在Scan1_Area_Start + 3×Font_Scale2+Scan_Dis。
0x1F	Scan_Return_Mode	1	0xAA:在*(VP-1)位置保存输入结束标记和有效数据长度： *(VP-1)高字节，输入结束标记：0x5A表示输入结束，输入过程中为0x00。 *(VP-1)低字节，有效输入数据长度，字节单位。 0xFF:不返回输入结束标记和长度。
0x20	0xFE	1	
0x21	Scan0_Area_Start	4	录入文本显示区域左上角坐标（Xs,Ys）
0x25	Scan0_Area_End	4	录入文本显示区域右下角坐标（Xe,Ye）
0x29	Scan1_Area_Start	4	录入过程中拼音提示文本显示区域的左上角坐标
0x2D	Scan_Dis	1	录入过程显示中，每个汉字显示的间距。每行固定显示最多8个汉字。
0x2E	0x00	1	
0x2F	KB_Source	1	键盘页面位置选择：0x00=键盘在当前页面；其它=键盘不在当前页面。
0x30	0xFE	1	
0x31	PIC_KB	2	以下数据，仅当KB_Source不为0x00时有效。键盘所在页面ID
0x33	AREA_KB	8	键盘页面上键盘区域坐标：左上角（Xs,Ys）、右下角（Xe,Ye）
0x3B	AREA_KB_Position	4	键盘区域粘贴在当前页面显示的位置，左上角坐标。
0x3F	SCAN_MODE	1	0x02:拼音输入法



5.9 硬件参数配置 (0x07)

地址	定义	数据长度	说 明
0x00	Pic_ID	2	页面ID
0x02	TP_Area	8	触控按钮区域 : (Xs,Ys) (Xe,Ye)
0x0A	Pic_Next	2	目标切换页面, 0xFF**表示不进行页面切换。
0x0C	Pic_On	2	按钮按压效果图所处的页面, 0xFF**表示没有按钮按压效果。
0x0E	TP_Code	2	0xFE07
0x10	0xFE	1	0xFE
0x11	Mode	1	操作模式选择, 见“操作模式表”说明。
0x12	DATA_PACK	14	操作模式数据包, 见“操作模式表”说明。

操作模式表

Mode	Data_Pack	Data_Pack说明	功 能
0x00	无	无	加载寄存器变量区数据到 0x6F00-0x6FFF 变量存储器空间（占据低字节）；
0x01	无	无	加载0x6F00-0x6FFF变量存储器空间（占据低字节）数据到寄存器变量区；同时改写对应的R1-R3,R5-RA SD/SDHC接口配置变量。
0x02	Tran_Area	将要转换的区域坐标： 左上角、右下角	将指定区域的内容转换成单色位图（纵向取模打印位图格式），并保存到VP指针指向的数据存储器。 1.区域宽度（Xe-Xs+1）必须是偶数； 2.区域高度（Ye-Ys+1）必须是8的倍数； 3.*VP指针保存数据格式如下： *VP：状态位，处理完成后设置成0x5555； *VP+1：横向字长度=（Xe-Xs+1）&0xFFFE/2； *VP+2：数据段个数=（Ye-Ys+1）&0xFFF8/8； *VP+3：位图数据开始，MSB方式。 如果启用了“参数自动上传功能”（R2.3=1），那么转换完成后，会按照*VP内容被修改成0x5555而自动上传一条提示信息。 本指令主要用于屏幕内容的打印输出。
	*VP	保存转换位图数据的缓冲区首地址	
0x03	*VP	数据指针	把*VP指针位置开始的，Tx_Len字节长度的数据发送到用户串口。 Tx_Len是一个字变量，长度从0x0001-0xFFFF。
	Tx_LEN	要发送的数据长度	
0x04	未定义		
0x05	Tran_Area	将要转换的区域坐标： 左上角、右下角	将指定区域的内容转换成单色位图（横向取模打印位图格式），并保存到VP指针指向的数据存储器。1.区域宽度（Xe-Xs+1）必须是16的倍数； 2.*VP指针保存数据格式如下： *VP：状态位，处理完成后设置成0x5555； *VP+1：横向字长度=（Xe-Xs+1）&0xFFF0/16； *VP+2：数据段个数=（Ye-Ys+1）； *VP+3：位图数据开始，MSB方式。 如果启用了“参数自动上传功能”（R2.3=1），那么转换完成后，会按照*VP内容被修改成0x5555而自动上传一条提示信息。 本指令主要用于屏幕内容的打印输出。
	*VP	保存转换位图数据的缓冲区首地址	
0x06	Frame_Head	帧头（2字节）	把当前点击位置的触摸屏坐标发到COM2（系统保留串口），格式如下：Frame_Head+X+Y+Check(X,Y的1字节累加和)+Frame_end。
	Frame_End	帧尾（2字节）	

5.10 触摸屏按压状态同步数据返回 (0x08)

地址	定义	数据长度	说 明
0x00	Pic_ID	2	页面 ID
0x02	TP_Area	8	触控按钮区域: (Xs, Ys) (Xe, Ye)
0x0A	Pic_Next	2	目标切换页面, 0xFF**表示不进行页面切换。
0x0C	Pic_On	2	按钮按压效果图所处的页面, 0xFF**表示没有按钮按压效果。
0x0E	TP_Code	2	0xFE08
0x10	0xFE	1	0xFE
0x11	TP_ON_Mode	1	触摸屏第一次按压下去时, 数据返回模式: 0x00=不返回数据 0x01=读取*VP1S 指向的 LEN1 长度数据到*VP1T 指向的存储空间。 0x02=读取*VP1S 指向的 LEN1 长度数据发送到串口。 0x03=读取*VP1S 指向的 LEN1 长度数据到*VP1T 指向的寄存器空间。
0x12	VP1S	2	触摸屏第一次按压时, 读取数据的地址。
0x14	VP1T	2	触摸屏第一次按压时, 写入数据的地址。
0x16	0x00	1	0x00
0x17	LEN1	1	返回数据长度, 字节数。TP_ON_Mode=0x01 时, LEN1 必须为偶数。
0x18	0xFE	1	0xFE
0x19	TP_ON_Continue_Mode	1	触摸屏第一次按压后, 持续按压时, 数据返回模式: 0x00=不返回数据 0x01=读取*VP2S 指向的 LEN2 长度数据到*VP2T 指向的存储空间。 0x02=读取*VP2S 指向的 LEN2 长度数据发送到串口。 0x03=读取*VP2S 指向的 LEN2 长度数据到*VP2T 指向的寄存器空间。
0x1A	VP2S	2	触摸屏按压时, 读取数据的地址。
0x1C	VP2T	2	触摸屏按压时, 写入数据的地址。
0x1E	0x00	1	0x00
0x1F	LEN2	1	返回数据长度, 字节数。TP_ON_Continue_Mode=0x01 时, LEN2 必须为偶数。
0x20	0xFE	1	0xFE
0x21	TP_OFF_Mode	1	触摸屏松开时, 数据返回模式: 0x00=不返回数据 0x01=读取*VP3S 指向的 LEN3 长度数据到*VP3T 指向的存储空间。 0x02=读取*VP3S 指向的 LEN3 长度数据发送到串口。 0x03=读取*VP3S 指向的 LEN3 长度数据到*VP3T 指向的寄存器空间。
0x22	VP3S	2	触摸屏松开时, 读取数据的地址。
0x24	VP3T	2	触摸屏松开时, 写入数据的地址。
0x26	0x00	1	0x00
0x27	LEN3	1	返回数据长度, 字节数。TP_OFF_Mode=0x01 时, LEN3 必须为偶数。
0x28	0x00	8	保留, 写 0x00

5.11转动调节 (0x09)

地址	定义	数据长度	说 明
0x00	Pic_ID	2	页面 ID
0x02	TP_Area	8	触控按钮区域: (Xs,Ys) (Xe,Ye), 为调节圆域的外框区域。
0x0A	Pic_Next	2	目标切换页面, 0xFF**表示不进行页面切换。必须为 0xFF**。
0x0C	Pic_On	2	按钮按压效果图所处的页面, 0xFF**表示没有按钮按压效果。必须为 0xFF**。
0x0E	TP_Code	2	0xFE09
0x10	0xFE	1	0xFE
0x11	*VP	2	变量地址指针。
0x13	Data_Format	1	调节数据格式: 0x00=调节 VP 字地址 (整型数); 0x01=调节 VP 字地址的高字节地址 (1 字节无符号数, VP_H); 0x02=调节 VP 字地址的低字节地址 (1 字节无符号数, VP_L)。
0x14	(X, Y)	4	调节区域圆心坐标
0x18	R0	2	调节区域内径
0x1A	R1	2	调节区域外径
0x1C	A0	2	调节区域起始角度, 0-719, 单位为 0.5°。
0x1E	V_begain	2	起始角度对应的返回值, 整数。
0x20	0xFE		0xFE
0x21	A1	2	调节区域终止角度, 1-720, 单位为 0.5°。
0x23	V_end	2	终止角度对应的返回值, 整数。

转动调节不支持按键 (0x4F 寄存器保存的键码) 控制。

转动调节始终假定为“顺时针”转动。

6 显示变量配置文件 (14.BIN) 说明

显示变量配置文件由 N 条按照页面配置的变量指令组成，每条变量指令固定占用 32 字节存储空间。每个页面固定分配 2KB 或 4KB (0x0800 或 0x1000) 变量存储空间，每个页面最多可以设置 64 或 128 个变量 (64/128 变量选择由 CONFIG.TXT 配置文件中的 RC.4 选择)。显示变量配置文件最大 2MB，可以配置最多 1024 个页面 (128 变量模式下为 512 页面)。一条显示变量配置指令由以下 6 部分组成：

序号	定义	数据长度	说 明
1	0x5A	1	固定
2	Type	1	变量类别
3	*SP	2	变量描述文件从 Flash 加载后存储到数据存储区的地址指针,0xFFFF 表示不转存到数据存储区。
4	Len_Dsc	2	变量描述内容的字长度
5	*VP	2	变量地址, 0x0000-0x6FFF，有些无需指定地址的变量，写 0x0000 即可。 当变量地址高字节为 0xFF 时，本条指令将被取消
6	Description	N	变量描述内容

6.1显示变量功能一览表

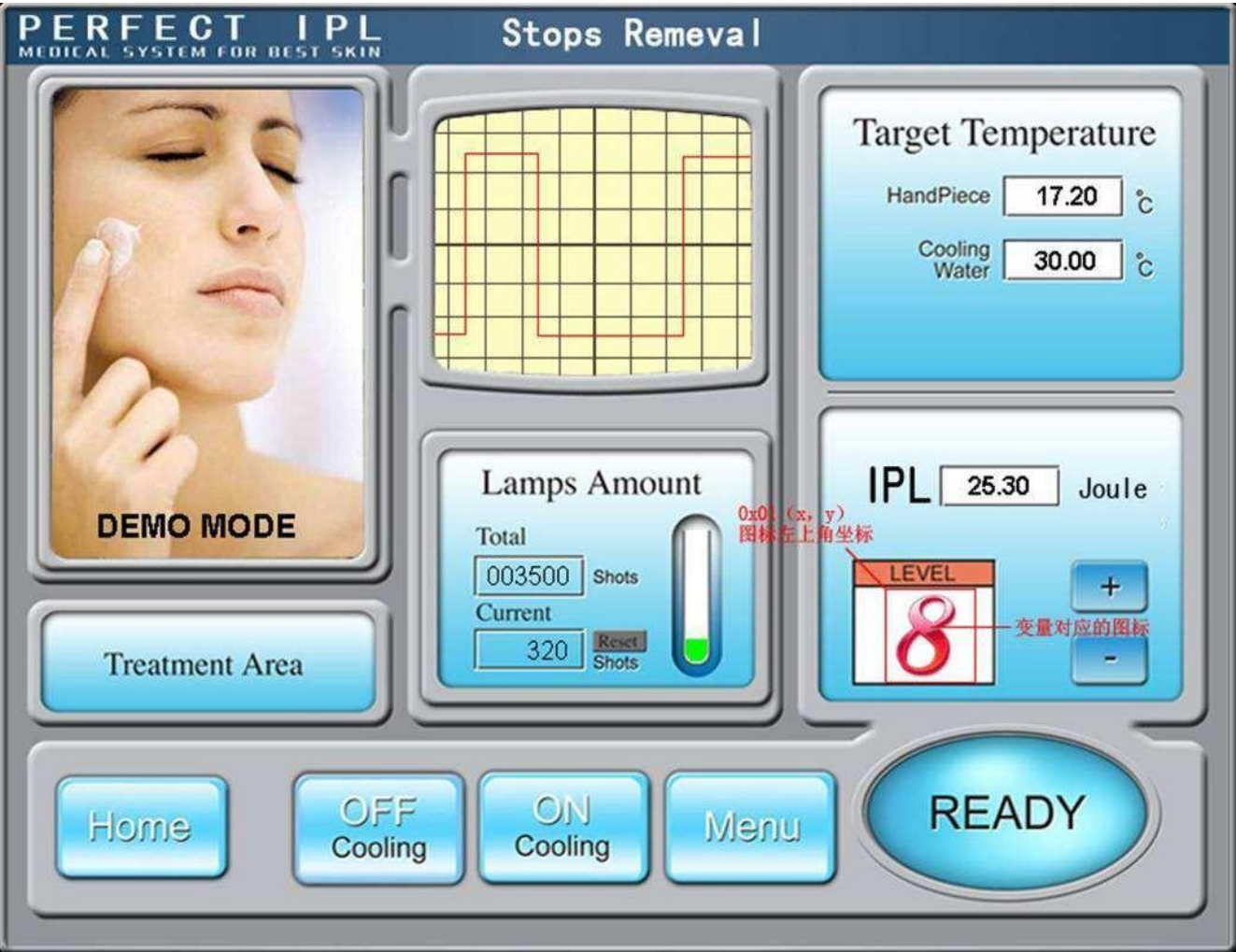
序号	功能代码	功 能	说 明
01	00	变量图标显示	将一个数据变量的变化范围线性对应一组ICON图标显示；当变量变化时，图标也自动相应切换。多用于精细的仪表板、进度条显示。
02	01	动画图标显示	将一个定值数据变量对应了3种不同的图标指示状态：不显示、显示固定图标、显示动画图标。多用于变量的报警提示。
03	02	滑块刻度指示	将一个数据变量的变化范围对应一个图标（滑块）的显示位置变化。多用于液位、刻度盘、进度表的指示。
04	03	艺术字显示	用ICON图标取代字库来显示变量数据。
05	04	图片动画显示	将一组全屏图片按照指定速度播放。多用于开机界面或屏保。
06	05	图标旋转显示	把一个数据变量的变化范围线性对应角度数据，然后把一个ICON图标按照对应的角度数据旋转后显示出来。多用于指针仪表板显示。
07	06	位变量图标显示	把一个数据变量的每个位（bit）的0/1状态对应8种不同显示方案中的两种，用ICON图标（或图标动画）来对应显示。多用于开关状态显示，比如风机的运转（动画）、停止（静止图标）。

08	10	数据变量显示	把一个数据变量按照指定格式（整数、小数、是否带单位）用指定字体和大小的阿拉伯数字显示出来。
09	11	文本显示	把字符串按照指定的格式（选择字库决定），在指定的文本框显示区域显示。
10	12_00	文本格式RTC显示	按照用户编辑的格式把公历RTC用文本显示出来。
11	12_01	表盘格式RTC显示	采用ICON图标旋转，用指针表盘方式把公历RTC显示出来。
12	13	HEX数据显示	把变量数据按照字节HEX方式间隔用户指定的ASCII字符显示出来。 多用于计时显示，比如把1234显示成12:34。
13	20	实时曲线（趋势图）	结合 0x84 串口写曲线缓冲区数据来自动匹配显示实时曲线（趋势图）。可以指定显示区域、中心轴坐标、显示比例（放大/缩小）可控。
14	21_01	绘图_置点	置点（x,y,color）
15	21_02	绘图_端点连线	端点连线（color,(x0,y0),...(xn,yn)）
16	21_03	绘图_矩形	显示矩形，颜色和位置、大小可控。
17	21_04	绘图_矩形填充	填充指定的矩形区域，填充颜色和位置、大小可控。
18	21_05	绘图_画圆	显示整圆弧，颜色和位置、大小可控。
19	21_06	绘图_图片剪切粘贴	从指定图片上剪切一个区域粘贴到当前显示页面上。
20	21_07	绘图_ICON图标显示	ICON图标显示，图标库可以选择。
21	21_08	绘图_封闭区域填充	封闭区域填充，种子点坐标、填充颜色可控。
22	21_09	绘图_频谱显示	根据变量数据显示频谱（垂直线条），线条颜色、位置可控。
23	21_0A	绘图_线段显示	根据变量数据连接线段，端点、颜色可控。
24	21_0B	绘图_圆弧显示	显示圆弧，半径、颜色、起止角度可控。
25	21_0C	绘图_字符显示	根据变量数据进行单个字符显示。
26	21_0D	绘图_矩形域XOR	对指定的矩形域位图数据用指定颜色进行XOR操作，多用于高亮显示。
27	21_0E	绘图_双色位图显示	变量存储器数据看成双色位图数据，0/1对应颜色可指定，多用于自定义光标。
28	21_0F	绘图_位图显示	变量存储器数据位65K色位图数据，多用于实时图标（照片）下载显示。
29	21_10	绘图_区域放大粘贴	把指定区域放大1倍粘贴到指定位置，多用于配合0F指令实现照片实时显示。
30	22	列表显示	把按照二维数组定义的数据用表格分栏显示出来。
31	25	二维码显示	根据指定内容在屏幕显示指定的二维码图形。

6.2图标变量

6.2.1变量图标显示 (0x00)

地址		定义	数据长度	说 明
0x00		0x5A00	2	
0x02		*SP	2	变量描述指针，0xFFFF表示由配置文件加载
0x04		0x0008	2	
0x06	0x00	*VP	2	变量指针，变量为整数格式。
0x08	0x01	(x,y)	4	变量显示位置，图标左上角坐标位置
0x0C	0x03	V_Min	2	变量下限，越界不显示
0x0E	0x04	V_Max	2	变量上限，越界不显示
0x10	0x05	Icon_Min	2	V_Min对应的图标ID
0x12	0x06	Icon_Max	2	V_max对应的图标ID
0x14	0x07:H	Icon_Lib	1	图标库存储位置
0x15	0x07:L	Mode	1	ICON显示模式，0x00=透明（不显示背景） 其它 = 显示图标背景



6.2.2动画图标显示 (0x01)

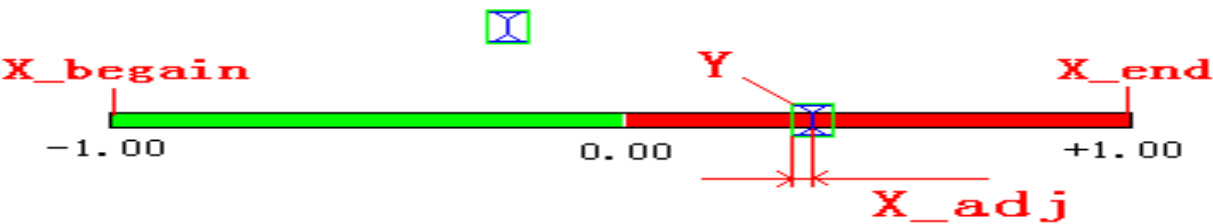
地址		定义	数量长度	说明
0x00		0x5A01	2	
0x02		*SP	2	变量描述指针, 0xFFFF表示由配置文件加载
0x04		0x000A	2	
0x06	0x00	*VP	2	初始图标变量指针, 变量为双字, 低位字保留, 高位字无符号数 (0x0000-0xFFFF) 用户数据控制动画图标显示
0x08	0x01	(x,y)	4	变量显示位置, 图标左上角坐标位置。
0x0C	0x03	Reset_Icon_En	2	0x0000 : 停止时, 不复位动画图标起始值 (动画图标显示从 Icon_Strat 到 Icon_End间1个任意值开始显示)。 0x0001 : 停止时, 复位动画图标起始值 (动画图标显示将固定从 Icon_start 开始显示)。
0x0E	0x04	V_Stop	2	变量为该值时显示固定图标
0x10	0x05	V_Start	2	变量为该值时自动显示动画图标
0x12	0x06	Icon_Stop	2	变量为V_STOP值时固定显示的图标
0x14	0x07	Icon_Start	2	变量为V_Start值时, 自动从Icon_Start到Icon_End 显示图标, 形成动画。
0x16	0x08	Icon_End	2	
0x18	0x09:H	Icon_Lib	1	图标库存储位置
0x19	0x09:L	Mode	1	ICON显示模式, 0x00=透明

当变量不等于V_Stop或者V_Start时, 不显示图标或者动画。



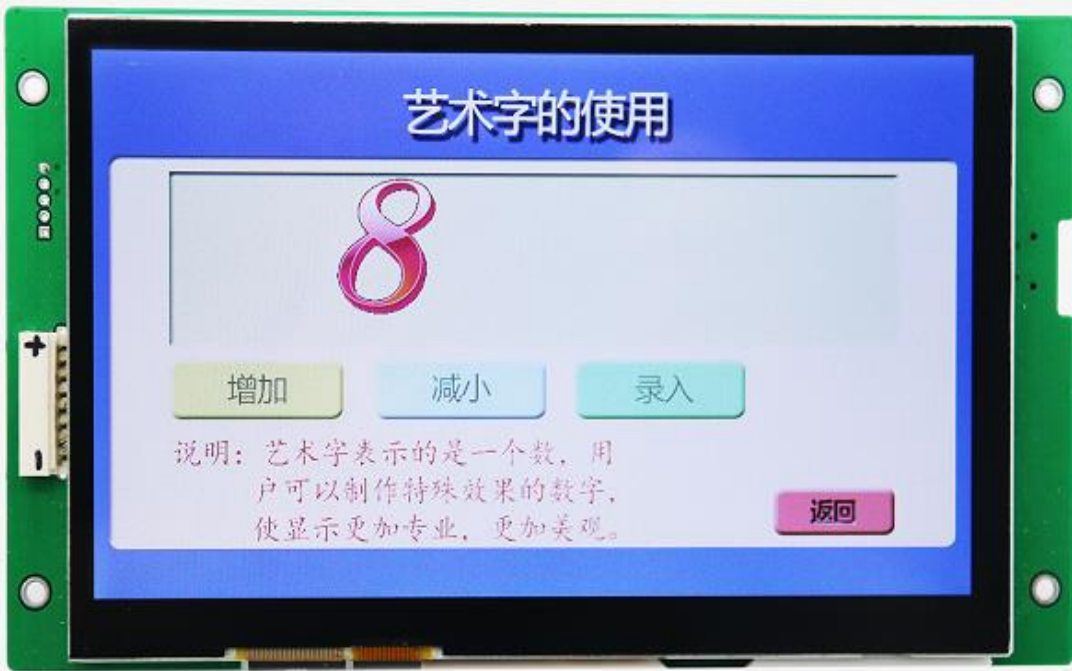
6.2.3滑块刻度显示 (0x02)

地址		定义	数据长度	说 明
0x00		0x5A02	2	
0x02		*SP	2	变量描述指针, 0xFFFF表示由配置文件加载
0x04		0x000A	2	
0x06	0x00	*VP	2	变量指针, 变量格式由VP_DATA_Mode决定。
0x08	0x01	V_begain	2	对应起始刻度的变量值
0x0A	0x02	V_end	2	对应终止刻度的变量值
0x0C	0x03	X_begain	2	起始刻度坐标 (纵向为Y坐标)
0x0E	0x04	X_end	2	终止刻度坐标 (纵向为Y坐标)
0x10	0x05	Icon_ID	2	刻度滑动块的图标ID
0x12	0x06	Y	2	刻度指示图标显示的Y坐标位置 (纵向为X坐标)
0x14	0x07:H	X_adj	1	刻度指示图标显示的X坐标前移偏移量 (纵向为Y), 0x00-0xFF
0x15	0x07:L	Mode	1	刻度模式: 0x00=横向刻度条 01 = 纵向刻度条
0x16	0x08:H	Icon_Lib	1	图标库存储位置
0x17	0x08:L	Icon_mode	1	ICON显示模式, 0x00=透明 (不显示背景) 其它 = 显示图标背景
0x18	0x09:H	VP_DATA_Mode	1	0x00: *VP指向一个整型变量 0x01: *VP指向一个整型变量的高字节数据 0x02: *VP指向一个整型变量的低字节数据



6.2.4 艺术字变量显示 (0x03)

地址		定义	数据长度	说 明
0x00		0x5A03	2	
0x02		*SP	2	变量描述指针，0xFFFF表示由配置文件加载
0x04		0x0007	2	
0x06	0x00	*VP	2	变量指针
0x08	0x01	X,Y	4	起始显示位置： 左对齐模式，坐标为显示字符串左上角坐标；右对齐模式，坐标为显示字符串的右上角坐标。
0x0C	0x03	Icon0	2	0对应的ICON_ID，排列顺序为0123456789-.
0x0E	0x04:H	Icon_Lib	1	Icon库位置
0x0F	0x04:L	Icon_Mode	1	ICON显示模式，0x00=透明（不显示背景）其它=显示图标背景
0x10	0x05:H	整数位数	1	显示的整数位数
0x11	0x05:L	小数位数	1	显示的小数位数
0x12	0x06:H	变量数据类型	1	0x00=整数(2字节)，-32768到32767 0x01=长整数(4字节)-2147483648到2147483647 0x02=*VP高字节,无符号数0到255 0x03=*VP低字节，无符号数0到255 0x04= 超 长 整 数 (8 字 节) -9223372036854775808到9223372036854775807 0x05=无符号整数(2字节)0到65535 0x06=无符号长整数(4字节)0到4294967295
0x13	0x06:L	对齐模式	1	0x00=左对齐 0x01=右对齐



6.2.5 图片动画显示 (0x04)

地址		定义	数据长度	说 明
0x00		0x5A04	2	
0x02		*SP	2	变量描述指针, 0xFFFF表示由配置文件加载
0x04		0x0004	2	
0x06	0x00	0x0000	2	固定
0x08	0x01	Pic_Begain	2	起始图片位置
0x0A	0x02	Pic_End	2	终止图片位置
0x0C	0x03:H	Frame_Time	1	一帧 (一幅图片) 显示的时间, 单位为8mS

起始图片位置必须小于终止图片位置。

如果在Pic_End页面也设置图片动画变量, 将可以实现不断重

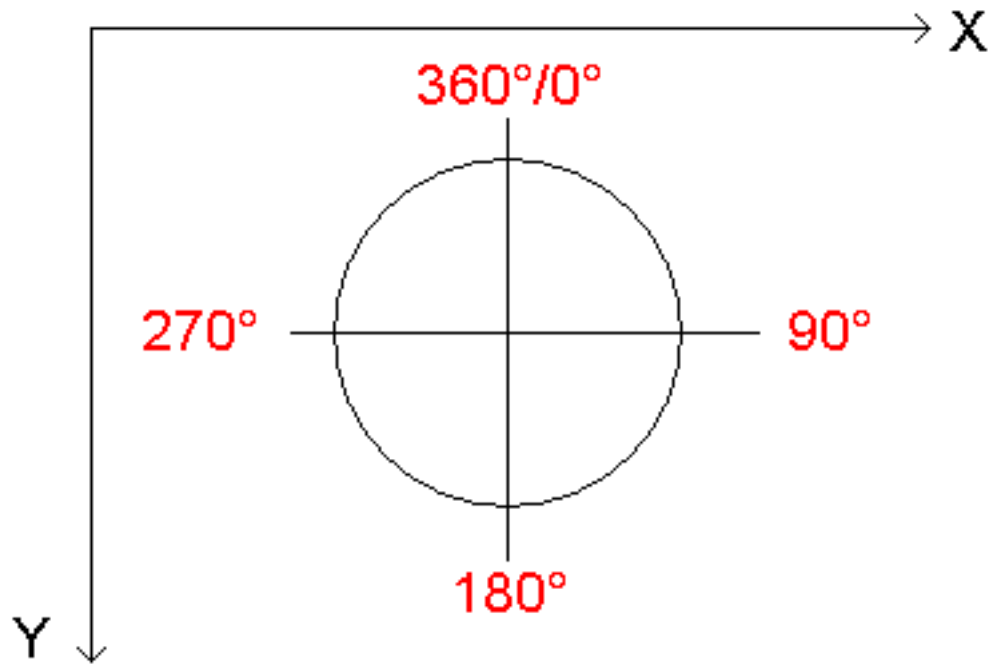
播。 串口指令切换图片或者触控指令切换图片可以结束重播。

6.2.6 图标旋转显示 (0x05)

地址		定义	数据长度	说 明
0x00		0x5A05	2	
0x02		*SP	2	变量描述指针, 0xFFFF表示由配置文件加载
0x04		0x000C	2	
0x06	0x00	*VP	2	变量指针, 变量格式由VP_Mode决定。
0x08	0x01	Icon_ID	2	指定的图标ID。
0x0A	0x02	Icon_Xc	2	ICON图标上的旋转中心位置: X坐标。
0x0C	0x03	Icon_Yc	2	ICON图标上的旋转中心位置: Y坐标。
0x0E	0x04	Xc	2	ICON显示到当前屏幕的旋转中心位置: X坐标。
0x10	0x05	Yc	2	ICON显示到当前屏幕的旋转中心位置: Y坐标。
0x12	0x06	V_Begain	2	对应起始旋转角度的变量值, 整型数, 越界不显示
0x14	0x07	V_End	2	对应终止旋转角度的变量值, 整型数, 越界不显示
0x16	0x08	AL_Begain	2	起始旋转角度, 0-720 (0x000-0x2D0), 单位0.5°。
0x18	0x09	AL_End	2	终止旋转角度, 0-720 (0x000-0x2D0), 单位0.5°。
0x1A	0x0A:H	VP_Mode	1	0x00: *VP指向一个整型变量 0x01: *VP指向一个整型变量的高字节数据 0x02: *VP指向一个整型变量的低字节数据
0x1B	0x0A:L	Lib_ID	1	ICON图标库ID。
0x1C	0x0B	Mode	1	ICON显示模式, 0x00=透明 (不显示背景) 其它 = 显示图标背景

本指令主要用于仪表刻度盘的指针指示。

旋转始终假定为“顺时针”转动，即AL_End必须大于AL_Begain(如果AL_End小于AL_Begain，系统处理时 会自动加上360°)。



6.2.7 位变量图标显示 (0x06)

地址		定义	数据长度	说 明																													
0x00		0x5A06	2																														
0x02		*SP	2	变量描述指针，0xFFFF 表示由配置文件加载																													
0x04		0x000C	2																														
0x06	0x00	*VP	2	位变量指针, 字变量																													
0x08	0x01	*VP_AUX	2	辅助变量指针，双字，用户软件不能访问。																													
0x0A	0x02	Act_Bit_Set	2	为 1 的 bit 位置说明*VP 对应位置需要显示。																													
0x0C	0x03:H	Display_Mode	1	定义了显示模式： <table border="1"><tr><th rowspan="2">Display_Mode</th><th colspan="2">变量位（Bit）值</th></tr><tr><th>0</th><th>1</th></tr><tr><td>0x00</td><td>ICONOS</td><td>ICON1S</td></tr><tr><td>0x01</td><td>ICONOS</td><td>不显示</td></tr><tr><td>0x02</td><td>ICONOS</td><td>ICON1S-ICON1E 动画</td></tr><tr><td>0x03</td><td>不显示</td><td>ICON1S</td></tr><tr><td>0x04</td><td>不显示</td><td>ICON1S-ICON1E 动画</td></tr><tr><td>0x05</td><td>ICONOS-ICONOE 动画</td><td>ICON1S</td></tr><tr><td>0x06</td><td>ICONOS-ICONOE 动画</td><td>不显示</td></tr><tr><td>0x07</td><td>ICONOS-ICONOE 动画</td><td>ICON1S-ICON1E 动画</td></tr></table> 比如设置 Display_Mode=0x02，那么： *VP 对应的变量某个位为“0”时，显示 ICONOS 图标；	Display_Mode	变量位（Bit）值		0	1	0x00	ICONOS	ICON1S	0x01	ICONOS	不显示	0x02	ICONOS	ICON1S-ICON1E 动画	0x03	不显示	ICON1S	0x04	不显示	ICON1S-ICON1E 动画	0x05	ICONOS-ICONOE 动画	ICON1S	0x06	ICONOS-ICONOE 动画	不显示	0x07	ICONOS-ICONOE 动画	ICON1S-ICON1E 动画
				Display_Mode		变量位（Bit）值																											
					0	1																											
				0x00	ICONOS	ICON1S																											
				0x01	ICONOS	不显示																											
				0x02	ICONOS	ICON1S-ICON1E 动画																											
				0x03	不显示	ICON1S																											
				0x04	不显示	ICON1S-ICON1E 动画																											
				0x05	ICONOS-ICONOE 动画	ICON1S																											
0x06	ICONOS-ICONOE 动画	不显示																															
0x07	ICONOS-ICONOE 动画	ICON1S-ICON1E 动画																															
0x0D	0x03:L	Move_mode	1	位图图标排列方式： 0x00=X++，Act_Bit_Set 指定的不处理 bit 不保留位置； 0x01=Y++，Act_Bit_Set 指定的不处理 bit 不保留位置； 0x02=X++，Act_Bit_Set 指定的不处理 bit 保留 DIS_MOV 位置； 0x03=Y++，Act_Bit_Set 指定的不处理 bit 保留 DIS_MOV 位置；																													
0x0E	0x04:H	Icon_Mode	1	ICON 显示模式：0x00=透明 0x01=不透明																													
0x0F	0x04:L	Icon_Lib	1	图标库存储位置																													
0x10	0x05	ICONOS	2	不显示动画模式，bit 0 图标 ID 显示动画模式，bit 0 图标动画起始 ID 位置																													
0x12	0x06	ICONOE	2	显示动画模式，bit 0 图标动画结束 ID 位置																													
0x14	0x07	ICON1S	2	不显示动画模式，bit 1 图标 ID 显示动画模式，bit 1 图标动画起始 ID 位置																													
0x16	0x08	ICON1E	2	显示动画模式，bit 1 图标动画结束 ID 位置																													
0x18	0x09	(x, y)	4	起始位变量显示位置，图标左上角坐标位置。																													
0x1C	0x0B	DIS_MOV	2	下一个图标坐标移动坐标间隔																													
0x1E			2	写 0x00																													

6.3 文本显示

6.3.1数据变量显示 (0x10)

地址		定义	数据长度	说 明	
0x00		0x5A10	2		
0x02		*SP	2	变量描述指针，0xFFFF表示由配置文件加载	
0x04		0x000D	2		
0x06	0x00	*VP	2	变量指针	
0x08	0x01	X,Y	4	起始显示位置，显示字符串左上角坐标。	
0x0C	0x03	COLOR	2	显示颜色	
0x0E	0x04:H	Lib_ID	1	ASCII字库位置	
0x0F	0x04:L	字体大小	1	字符X方向点阵数	
0x10	0x05:H	对齐方式	1	0x00=左对齐 0x01=右对齐 0x02=居中	
0x11	0x05:L	整数位数	1	显示整数位	整数位数和小数位数之和不能超过20。
0x12	0x06:H	小数位数	1	显示小数位	
0x13	0x06:L	变量数据类型	1	0x00=整数(2字节)，-32768到32767 0x01=长整数(4字节) -2147483648到2147483647 0x02=*VP高字节,无符号数 0到255 0x03=*VP低字节，无符号数 0到255 0x04=超长整数(8字节) -9223372036854775808到9223372036854775807 0x05=无符号整数(2字节) 0到65535 0x06=无符号长整数(4字节) 0到4294967295	
0x14	0x07:H	Len_unit	1	变量单位（固定字符串）显示长度，0x00表示没有单位显示	
0x15	0x07:L	String_Unit	Max11	单位字符串，ASCII编码	



6.3.2文本显示 (0x11)

地址		定义	数量长度	说明
0x00		0x5A11	2	
0x02		*SP	2	变量描述指针, 0xFFFF表示由配置文件加载
0x04		0x000D	2	
0x06	0x00	*VP	2	文本指针
0x08	0x01	X,Y	4	起始显示位置, 显示字符串左上角坐标。
0x0C	0x03	Color	2	显示文本颜色
0x0E	0x04	Xs Ys Xe Ye	8	文本框
0x16	0x08	Text_length	2	显示字节数量,遇到 0xFFFF、0x0000 数据或者显示到文本框尾将不再显示。
0x18	0x09:H	Font0_ID	1	编码方式0x01-0x04时ASCII字库位置。
0x19	0x09:L	Font1_ID	1	编码方式0x00、0x05, 以及0x01-0x04的非ASCII字符使用的字库
0x1A	0x0A:H	Font_X_Dots	1	字体X方向点阵数 (0x01-0x04模式, ASCII字符X按照X/2计算)
0x1B	0x0A:L	Font_Y_Dots	1	字体Y方向点阵数目
0x1C	0x0B:H	Encode_Mode	1	.7定义了文本显示的字符间距是否自动调整： .7=0 字符间距自动调整； .7=1 字符间距不自动调整, 字符宽度固定为设定的点阵数。 .6-.0定义了文本编码方式： 0=8bit编码 1=GB2312内码 2=GBK 3=BIG5 4=SJIS 5=UNICODE
0x1D	0x0B:L	HOR_Dis	1	字符水平间隔
0x1E	0x0C:H	VER_Dis	1	字符垂直间隔
0x1F	0x0C:L	未定义	1	写0x00

注意, 文本显示时, 字库中字体的Y方向点阵数目必须为偶数。

HGUS屏预装的0#字库, 包含4*8 - 64*128点阵的所有ASCII字符。

6.3.3 RTC显示 (0x12)

文本RTC显示

地址		定义	数据长度	说 明
0x00		0x5A12	2	
0x02		*SP	2	变量描述指针，0xFFFF表示由配置文件加载
0x04		0x000D	2	
0x06	0x00	0x0000	2	
0x08	0x01	X,Y	4	显示位置，显示字符串左上角坐标。
0x0C	0x03	Color	2	字体颜色
0x0E	0x04:H	Lib_ID	1	字库位置
0x0F	0x04:L	字体大小	1	X方向点阵数目
0x10	0x05	String_Code	MAX16	编码字符串，使用RTC编码表和ASCII字符构成。 假设当前时间是2012-05-02 12:00:00 星期三，那么 Y-M-D H:Q:S0x00将显示为 2012-05-02 12:00:00 M-D W H:Q 0x00 将显示为 05-02 WED 12:00

RTC编码表：

说 明	编码	显示格式
公历_年	Y	2000-2099
公历_月	M	01-12
公历_日	D	01-31
公历_小时	H	00-23
公历_分钟	Q	00-59
公历_秒	S	00-59
公历_星期	W	SUN MON TUE WED THU FRI SAT
编码结束	0x00	

表盘时钟显示

地址		定义	数据长度	说 明
0x00		0x5A12	2	
0x02		*SP	2	变量描述指针，0xFFFF表示由配置文件加载
0x04		0x000D	2	
0x06	0x00	0x0001	2	
0x08	0x01	X,Y	4	时钟表盘的指针中心。
0x0C	0x03	Icon_Hour	2	时针ICON的ID，0xFFFF表示时针不显示。
0x0E	0x04	Icon_Hour_Central	4	时针ICON的旋转中心位置。
0x12	0x06	Icon_Minute	2	分针ICON的ID，0xFFFF表示分针不显示。
0x14	0x07	Icon_Minute_Central	4	分针ICON的旋转中心位置。
0x18	0x09	Icon_Second	2	秒针ICON的ID，0xFFFF表示秒钟指针不显示。
0x1A	0x0A	Icon_Second_Central	4	秒针ICON的旋转中心位置。
0x1E	0x0C:H	ICON_Lib	1	指针图标所在的ICON库文件ID
0x1F		未定义	1	写0x00



6.3.4 HEX变量显示 (0x13)

地址		定义	数据长度	说 明
0x00		0x5A13	2	
0x02		*SP	2	变量描述指针, 0xFFFF 表示由配置文件加载
0x04		0x000D	2	
0x06	0x00	*VP	2	变量指针数据串首地址, 变量为 BCD (HEX) 编码。 比如: 数据0x32 显示为32。 数据0xBF 将显示为BF。
0x08	0x01	X.Y	4	显示起始位置, 显示字符串左上角坐标。
0x0C	0x03	Color	2	字体颜色
0x0E	0x04:H	Byte Num	1	*VP 指针高字节开始显示的字节数目, 0x01-0x0F
0x0F	0x04:L	Lib_ID	1	字库位置; 字库必须是半角方式。 如果Lib_ID 不为0, 字库必须使用8bit 编码。
0x10	0x05:H	Font_x	1	X 方向点阵数目。
0x11	0x05:L	String_Code	MAX15	编码字符串, 用来和时间变量组合出客户需要的显示格式。 每显示一个BCD 时间码后, 会到编码字符串顺序取出一个ASCII 字符来间隔显示。 编码字符串中, 特殊字符定义如下: 0x00: 无效, 本字符不显示, 两个 BCD 时间码将连在一起显示; 0x0D: 换行显示, 即X=Xs, Y=Y+Font_X*2。

6.3.5 文本滚屏显示 (0x14)

地址		定义	数据长度	说 明
0x00		0x5A14	2	
0x02		*SP	2	变量描述指针, 0xFFFF 表示由配置文件加载
0x04		0x000B	2	
0x06	0x00	*VP	2	文本指针。 文本指针前 3 个字必须保留, 用户显示文本内容从 (VP+3) 开始存放。文本必须以0xFF 或0x00 结尾。
0x08	0x01:H	Rolling Mode	1	滚屏模式: 0x00=从右向左滚屏。
0x09	0x01:L	Rolling Dis	1	滚屏间距, 每个 HGUS 周期文本滚动的像素点阵数。
0x0A	0x02:H	Adjust_Mode	1	0x00=左对齐 0x01=右对齐 0x02=居中。 文本显示内容不足文本框时滚屏停止, 此时显示对齐模式方有效。
0x0B	0x02:L	Run_Control	1	运行控制: 0x00=正常滚动 0x01=暂停 0x02=关闭 0x03=初始化 (静止显示)
0x0C	0x03	Color	2	显示文本颜色
0x0E	0x04	Xs Ys Xe Ye	8	文本框
0x16	0x08:H	Font0_ID	1	编码方式为0x01-0x04 时: ASCII 字符显示的字库位置。 编码方式为0x00、0x05 时: 本参数不用设置, 写0x00 即可。
0x17	0x08:L	Font1_ID	1	编码方式为0x01-0x04 时: 非ASCII 字符显示的字库位置。 编码方式为0x00、0x05 时: 显示字符使用的字库位置。
0x18	0x09:H	Font_X_Dots	1	字体 X 方向点阵数 (0x01-0x04 模式, ASCII 字符 X 将自动按照 X/2 计算)。
0x19	0x09:L	Font_Y_Dots	1	字体 Y 方向点阵数目。
0x1A	0x0A:H	Encode_Mode	1	.7 定义了文本显示的字符间距是否自动调整: .7=0 字符间距自动调整; .7=1 字符间距不自动调整, 字符宽度固定为设定的点阵数。 .6-.0 定义了文本编码方式: 0=8bit 编码 1=GB2312 内码 2=GBK 3=BIG5 4=SJIS 5=UNICODE
0x1B	0x0A:L	Text_Dis	1	字符间隔
0x1C	0x0A:H	未定义	4	写0x00

注意, 文本显示时, 字库中字体的 Y 方向点阵数目必须为偶数。

HGUS 屏预装的 0#字库, 包含 4*8—64*128 点阵的所有 ASCII 字符。

6.4图形变量

6.4.1实时曲线（趋势图）显示（0x20）

地址		定义	数据长度	说 明
0x00		0x5A20	2	
0x02		*SP	2	变量描述指针，0xFFFF 表示由配置文件加载
0x04		0x000A	2	
0x06	0x00	0x0000	2	无定义
0x08	0x01	Xs:Ys:Xe:Ye	8	曲线窗口：左上角坐标（Xs,Ys），右下角坐标（Xe,Ye）； 曲线越界将不显示。
0x10	0x05	Y_Central	2	曲线中心轴位置
0x12	0x06	VD_Central	2	中心轴对应的的曲线数据值，一般取最大数据和最小数据和的 50%。
0x14	0x07	Color	2	曲线颜色
x16	0x08	MUL_Y	2	纵轴放大倍数，单位是 1/256，0x0000-0x7FFF。
0x18	0x09:H	CHANEL	1	数据源通道，0x00-0x07
0x19	0x09:L	Dis_HOR	1	横轴间隔，0x01-0xFF。

曲线数据用0x84 指令发送，请参考 指令集 说明。

如果把变量描述内容存储在数据存储空间（*SP指定存储位置），那么：

结合0x02增量触控指令，可以实现不需要用户代码干预的曲线自动缩放；

结合0x03拖动触控指令修改Y_Central值，可以实现无需用户代码干预的曲线上下移动。

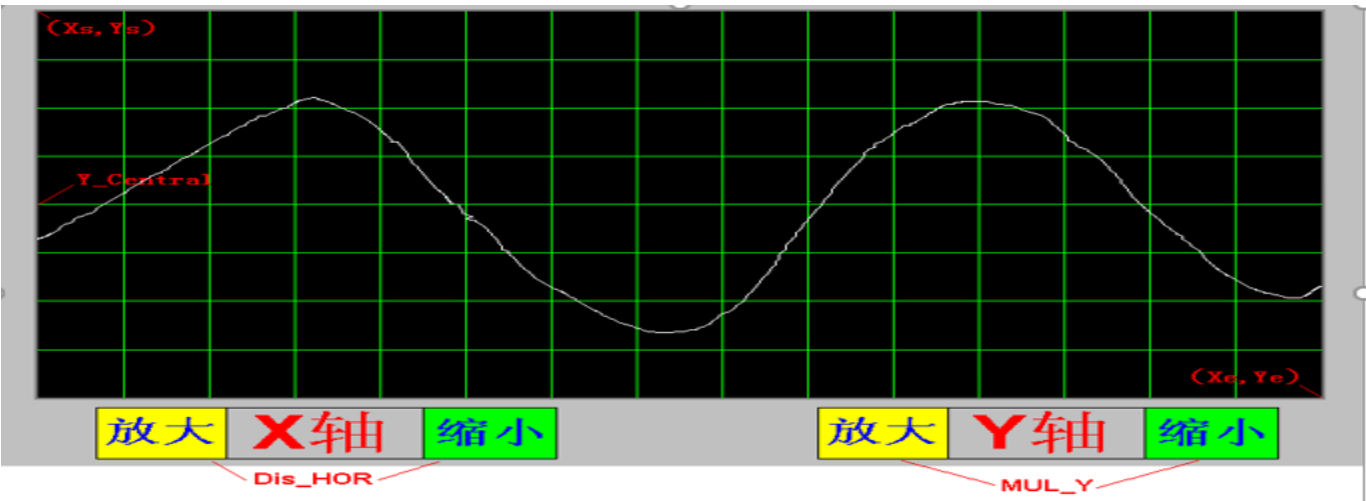
如果需要显示曲线线条比较粗，可以在同一个位置放置多个上下（Y 轴）平移的曲线变量并应引用同一个数据源通道即可实现。

满量程曲线的纵轴放大倍数计算：

$MUL_Y = (Ye - Ys) * 256 / (Vmax - Vmin)$ Ye,Ys为曲线窗口的Y坐标，Vmax,Vmin为曲线数据的最大,最小值。

比如，一个12bit A/D 采集数据（Vmax=4095 Vmin=0）要对应在Ys=50,Ye=430的屏幕区域满量程显示，那么：

$MUL_Y = (430 - 50) * 256 / (4095 - 0) = 23.7$ 向下舍入取23。



6.4.2基本图形显示 (0x21)

地址		定义	数据长度	说 明
0x00		0x5A21	2	
0x02		*SP	2	变量描述指针, 0xFFFF 表示由配置文件加载
0x04		0x0008	2	
0x06	0x00	*VP	2	变量数据指针。
0x08	0x01	Area	8	绘图显示区域定义: 指定区域的左上角、右下角坐标; 绘图越界将不显示。仅对 0x0001-0x0005、0x0009、0x000A、0x000B 指令有效。
0x10	0x05:H	Dashed_Line_En	1	0x5A: 使用线段的绘图指令 (0x02、0x03、0x09、0x0A 指令) 将使用虚线或者点划线显示线段; 其它: 使用线段的绘图指令使用实线显示线段。
0x11	0x05:L	Dash_Set	4	4 个字节依次设置了虚线 (点划线) 格式: 第 1 段实线点阵数、第 1 段虚线点阵数、第 2 段实线点阵数、第 2 段虚线点阵数。 比如, 设置 0x10 0x04 0x10 0x04 将显示虚线; 设置 0x10 0x04 0x02 0x04 将显示点划线。
0x15			13	保留, 写 0x00

基本图形显示先在 14.BIN 中定义一个“绘图板”功能, 而具体的绘图操作则由*VP指向的变量存储器容决定。用户通过改变变量存储器功能来实现不同的绘图功

(变量存储空间的) 变量数据格式说明

地 址	定 义	说明
VP	CMD	绘图指令
VP+1	Data_Pack_Num_Max	最大数据包数目: 连线指令 (0x0002), 定义为连线线条数目 (顶点数-1);
VP+2	DATA_Pack	数据

绘图指令数据包说明

指令 (CMD)	操作	绘图数据包格式说明 (相对地址和长度单位均为字 (word))			
		相对地址	长度	定义	说明
0x0001	置点	0x00	2	(x, y)	置点坐标位置, x 坐标高字节为判断条件。
		0x02	1	Color	置点颜色
0x0002	端点连线	0x00	1	Color	线条颜色
		0x01	2	(x, y) 0	连线顶点 0 坐标, x 坐标高字节为判断条件。
		0x03	2	(x, y) 1	连线顶点 1 坐标, x 坐标高字节为判断条件。
		0x01+2*n	2	(x, y) n	连线顶点 n 坐标, x 坐标高字节为判断条件。
0x0003	矩形	0x00	2	(x, y) s	矩形框左上角坐标, x 坐标高字节为判断条件。
		0x02	2	(x, y) e	矩形框右下角坐标。
		0x04	1	Color	矩形颜色
0x0004	矩形域填充	0x00	2	(x, y) s	矩形域左上角坐标, x 坐标高字节为判断条件。
		0x02	2	(x, y) e	矩形域右下角坐标。
		0x04	1	Color	矩形域填充颜色
0x0005	整圆弧显示	0x00	2	(x, y)	圆心坐标, x 坐标高字节为判断条件。
		0x02	1	Rad	半径。
		0x03	1	Color	圆颜色。
0x0006	图片区域剪切, 粘贴	0x00	1	Pic_ID	剪切图片区域所在页面 ID; 高字节为判断条件。
		0x01	2	(x, y) s	剪切图片区域左上角坐标。
		0x03	2	(x, y) e	剪切图片区域右下角坐标。
		0x05	2	(x, y)	剪切图片区域粘贴到当前页面的坐标位置, 左上角坐标。
0x0007	ICON 图标显示	0x00	2	(x, y)	显示坐标位置, x 坐标高字节为判断条件。
		0x02	1	ICON_ID	图标 ID, 图标库位置由指令高字节指定。 图标固定为不显示背景色。

0x0008	区域填充	0x00	2	(x, y)	种子点坐标, x 坐标高字节为判断条件。
		0x02	1	COLOR	填充颜色。
0x0009	频谱显示	0x00	1	Color0	把 (X0, Y0s) (X0, Y0e) 用 Color0 颜色连线, X0 高字节为判断条件。
		0x01	3	X0, Y0s, Y0e	
0x000A	线段显示	0x00	1	Color	把 (Xs, Ys) (Xe, Ye) 用 Color 颜色连线, Xs 高字节为判断条件
		0x01	2	Xs, Ys	
		0x03	2	Xe, Ye	
0x000B	圆弧显示	0x00	1	Color0	圆弧显示颜色
		0x01	2	(X, Y) 0	圆心 (X, Y) 坐标, X 坐标高字节为判断条件。
		0x03	1	RAD0	半径
		0x04	1	DEG_S0	起始角度, 单位 0.5°, 0-720
		0x05	1	DEG_E0	终止角度, 单位 0.5°, 0-720
0x000C	字符显示	0x00		Color0	字符显示颜色
		0x01		(X, Y) 0	字符显示位置, 字符左上角坐标, X 坐标高字节为判断条
		0x03H		Lib_ID	字库位置
		0x03L		En_Mode	字符编码模式: 0=8bit 1=GB2312 2=GBK 3=BIG5 4=SJIS
		0x04H		X_Dots	字符 X 方向点阵数
		0x04L		Y_Dots	字符 Y 方向点阵数
		0x05		Text0	字符数据, 对 8bit 编码, 仅高字节有效。当编码方式为 01-04 时, 如果字符数据为 ASCII 字符, 将自动使用 0#预装字库显示。
0x000D	矩形区域 XOR	0x00	2	(x, y) s	矩形域左上角坐标, x 坐标高字节为判断条件。
		0x02	2	(x, y) e	矩形域右下角坐标。
		0x04	1	Color	矩形域做 XOR 的颜色, 0xFFFF 将进行反色操作。
0x000E	双色位图显示	0x00	2	(x, y) s	位图显示矩形域左上角坐标, x 坐标高字节为判断条件。
		0x02	1	X_Dots	位图 X 方向点阵数目
		0x03	1	Y_Dots	位图 Y 方向点阵数目
		0x04	1	Color1	“1” bit 对应的显示
		0x05	1	Color0	“0” bit 对应的显示颜色; 如果设置 Color0 和 Color1 相同, 表示 “0” bit 不需要显示, 直接跳过。
		0x06	n	Data_Pack	显示数据, MSB 方式; 为方便用户读写数据, 每行数据必须对齐到一个字, 即下一行数据总是从一个新数据字 (Word) 开始。
0x000F	位图显示	0x00	2	(x, y) s	位图显示矩形域左上角坐标, x 坐标高字节为判断条件。
		0x02	1	X_Dots	位图 X 方向点阵
		0x03	1	Y_Dots	位图 Y 方向点阵
		0x04	n	Data_Pack	显示数据, 每个像素点一个字 (MSB, 5R6G5B 数据格式)。
0x0010	区域放大一倍粘贴显示	0x00	2	(x, y)	放大一倍后图像粘贴在屏幕左上角坐标, x 高字节判断条件
		0x02	2	(x, y) s	待放大矩形域左上角坐标。
		0x04	2	(x, y) e	待放大矩形域右下角坐标。

判断条件: 0xFF 绘图操作结束

0xFE 本次操作跳过

6.4.3列表显示 (0x22)

地址		定义	数据长度	说 明
0x00		0x5A22	2	
0x02		*SP	2	变量描述指针，0xFFFF表示由配置文件加载
0x04		0x000C	2	
0x06	0x00	*VP	2	表格内容指针，即TAB[TAB_X_Num][TAB_Y_Num]数组的首地址。
0x08	0x01:H	TAB_X_Num	1	列数目，0x01-0xFF
0x09	0x01:L	TAB_Y_Num	1	行数目，0x01-0xFF
0x0A	0x02:H	TAB_X_Start	1	表格起始显示列位置，0x00-0xFF。
0x0B	0x02:L	TAB_Y_Start	1	表格起始显示行位置，0x00-0xFF。
0x0C	0x03:H	Unit_Data_Num	1	<p>0x01-0x7F 所有单元格存储数据长度相同 一个单元格所占的数据空间长度（Word，字长度）。</p> <p>0x0 由*VP 指针指向的变量存储空间定义了不同列单元格的数据长度（Word，字长度）。</p> <p>当 Unit_Data_Num=0x00 时，表格数据内容存储位置相应后延（TAB_X_Num/2）向上取整个字地址。</p> <p>比如，*VP=0x1000，TAB_X_Num=0x07，那么：0x1000-0x1003 依次存储了第 0-6 列的表格数据长度，其中 1003 的低字节未使用。0x1004地址开始存储表格内容。</p>
0x0D	0x03:L	Encode_Mode	1	<p>.7定义了文本显示的字符间距是否自动调整： .7=0 字符间距自动调整； .7=1 字符间距不自动调整字符宽度固定为设定的点阵数。</p> <p>.6 定义了表格内容格式 .6=0 表格内容为文本格式； .6=1 表格内容格式由单元格数据的前两个字表示。</p> <p>.5 定义了边框线条是否显示，.5=0 显示边框，0.5=1 不显示边框。</p> <p>.4 未定义，写0。</p> <p>.3-.0定义了文本编码方式： 0=8bit编码 1=GB2312内码 2=GBK 3=BIG5 4=SJIS 5=UNICODE</p>
0x0E	0x04	Xs Ys Xe Ye	8	表格显示区域定义，表格左上角、右下角坐标； 表格总是总是从左上角位置开始显示，越界将结束显示。
0x16	0x08	Color_line	2	表格边框线条颜色
0x18	0x09	Color_text	2	表格文本显示颜色
0x1A	0x0A:H	Font0_ID	1	编码方式0x01-0x04时ASCII字库位置。
0x1B	0x0A:L	Font1_ID	1	编码方式0x00、0x05，以及0x01-0x04的非ASCII字符使用的字库
0x1C	0x0B:H	Font_X_Dots	1	字体X方向点阵数（0x01-0x04模式，ASCII字符X按照X/2计算）
0x1D	0x0B:L	Font_Y_Dots	1	字体Y方向点阵数目
0x1E	0x0C:H	TAB_X_Adj_Mod	1	当设置TAB_X_Start不为零时，进行显示表头控制： 0x00=首列不显示； 0x01=首列显示。
0x1F	0x0C:L	TAB_Y_Adj_Mod	1	当设置TAB_Y_Start不为零时，进行显示表头控制： 0x00=首行不显示； 0x01=首行显示。

当Encode_mode.6=1时，每个单元格数据内容的前两个字定义了表格数据格式，说明如下：

第1个字高字节:Mode 选择数据类型；

0x00=整数(2字节)， -32768到32767

0x01=长整数(4字节) -2147483648到2147483647

0x02=*VP高字节,无符号数 0到255

0x03=*VP低字节，无符号数 0到255

0x04=超长整数(8字节) -9223372036854775808到9223372036854775807

0x05=无符号整数(2字节) 0到65535

0x06=无符号长整数(4字节) 0到4294967295

0x10=时间格式1，12:34:56 BCD码串

0x11=时间格式2，12-34-56 BCD码串

0x12=时间格式3，YYYY-MM-DD HH:MM:SS BCD码串

0xFF=文本格式

第一个字低字节：

Mode=0x00-0x06 定义了变量数据的定点显示格式，高4bit表示整数位数，低

4bit表示小数位数。 Mode=0x10-0x11 时间BCD码串的字长

Mode=其它 无定义

第2个字：定义单元格文本颜色。

如果表格实际内容短于Unit_Data_Num规定的长度时，使用0xFFFF做为单元格文本结束符。对于特别大的表格，通过触摸屏修改TAB_X_Start、TAB_Y_Start值可以很方便的实现表格的定位和拖动。

6.4.4 二维QR码图形显示 (0x25)

地址		定义	数据长度	说 明
0x00		0x5A25	2	
0x02		*SP	2	变量描述指针，0xFFFF表示由配置文件加载
0x04		0x0004	2	
0x06	0x00	*VP	2	二维码显示内容指针。 二维码内容最长458Bytes，0x0000或0xFFFF为结束符。
0x08	0x01	(x , y)	4	二维码显示的坐标位置。 (x , y) 为二维码左上角在屏幕的坐标位置。 二维码图形有45*45单元像素 (数据少于155字节) 和73*73单元像素 (数据少于459Bytes) 两种。
0x0C	0x03	Unit_Pixels	2	每个二维码单元像素所占用的物理像素点阵大小，0x01-0x07。 设置Unit_Pixels=4，那么每个单元像素将显示为4*4点阵大小
0x0B-0x1F		保留	18	未定义，写0x00。



7. HGUS 特殊应用软件使用说明

7.1 基于Modbus主机接口的HGUS软件应用说明

主要功能：用户接口指令集为 Modbus RTU 主机模式，串口不再支持标准的 HGUS 指令集。

通过 22.BIN 字库文件的 20KW-28KW（16KB 空间大小，文件字节地址 0x0A000-0x0DFFF）来定义 Modbus 运行参数，HGUS 上电会自动把这 8KW 配置文件解码加载到变量缓冲区的最后 8KW，定义如下表。

HGUS 变量地址	定义	说明
0x5000	Modbus启用标记	0x5AA5表示启用Modbus通信。
0x5001:H	保存配置文件标记	0x5A：保存HGUS变量缓冲区的Modbus配置文件到22字库。
0x5001:L	加载配置文件标记	0x5A：从22字库加载Modbus配置文件到HGUS变量缓冲区。
0x5002	串口波特率	3.1格式定点小数(115200bps=0x480)，单位为kbps。
0x5003:H	串口模式	0x00=8N1（无校验）0x01=8E1（偶检验，EVEN）0x02=8O1（奇校验，ODD） 0x03=8N2（无校验，2个停止位）
0x5004:H	变量写标记	0x5A 表示保存指定HGUS变量空间（PS:00-PE:00）到22字库。
0x5004:L	变量读标记	0x5A 表示从22字库相应位置读数据到HGUS指定变量空间。
0x5005	变量读写起止地址	PS:PE PS、PE是HGUS变量空间开始、结束地址高字节，低字节固定为0。
0x5006-0x5007	保留	写0x0000
0x5008-0x500F	第1条MODBUS指令 (16字节)	0x00(0x5008H)：0x5A=本条指令有效，其它=本条指令无效 0x01(0x5008L)：读写的Modbus设备地址 0x02(0x5009H)：读/写使用的Modbus指令 0x03(0x5009L)：读写数据长度，0x00表示本条指令无效 0x04(0x500A)：本条指令处理定时时间，包含指令传送时间，4位整数，单位为mS，最大9999mS。对于读指令，定时时间是从机响应的最长时间。 0x06(0x500B)：4个字节规定了Modbus读写指令的发送方式 0x0000:**** 所有页面下均执行指令 0x0001:Page_ID 仅在指定页面下执行指令 0x0002:VP 仅在VP指向的变量缓冲区低字节内容为0x5A才执行指令，所有相关指令执行完后自动清零VP指向的内容。 0x0A(0x500D)：本条指令读写数据在HGUS屏变量存储区的起始地址。如果地址高字节为0xFF，表示读取的数据将写入HGUS曲线缓冲区，此时低字节地址表示为曲线数据格式。 0x0C(0x500E)：本条指令读写的数据在MODBUS设备上的数据起始地址。 0x0E(0x500FH)：总线通信状态反馈，写指令始终为0x00，读指令返回0x00（失败）/0xFF(成功)。 0x0F(0x500FL)：保留，写0x00。
.....		
0x6FF8-0x6FFF	第1023条指令	最多支持1023条Modbus指令

MODBUS 指令操作对应表（注意，Modbus 的位（线圈）变量是按照 LSB 定义的，而 HGUS 是按照 MSB 定义）

MODBUS 指令	功 能	读写数据长度	MODBUS 起始地址
0x01	读取输入线圈状态	线圈个数/8	起始线圈位置
0x02	读取输入位变量状	位变量个数/8	起始输出位置
0x03	读取保存寄存器数据	寄存器个数*2	保存寄存器首地址
0x04	读取输入寄存器数据	寄存器个数*2	输入寄存器首地址
0x05	强置单个线圈	0x02	线圈地址
0x06	预置单个寄存器	0x02	寄存器地址
0x07	读取异常状态	0x01	任意值
0x0F	强置多个线圈	线圈数量	起始线圈位置
0x10	预置多个寄存器	寄存器个数*2	寄存器首地址
0x11	读取从机标示	从机标示字节数	任意值

7.2 基于 Modbus 从机接口的 HGUS 软件应用说明

主要功能：用户接口指令集为Modbus RTU从机模式，串口不再支持标准的HGUS指令集。

CONFIG.TXT配置文件的R3寄存器设置Modbus从机地址。

CONFIG.TXT配置文件的RA寄存器设置Modbus串口格式：

0x00=8N1 (无校验)

0x01=8E1 (偶检验 , EVEN)

0x02=8O1 (奇校验 , ODD)

0x03=8N2 (无校验 , 2个停止位)。

MODBUS 指令操作对应表：

MODBUS 指令	功 能	读写数据长度	地址范围
0x03	读取 HGUS 变量存储器	字长度	0x0000-0x6FFF
	Tx: 5A 03 00 00 00 02 C9 20 Rx: 5A 03 04 00 00 00 00 10 F6 从变量存储区 0x0000 地址开始读取 2 个变量数据		
0x04	读 HGUS 寄存器	寄存器数量/2	0x00-0xFF
	Tx: 5A 04 00 00 00 02 7C E0 Rx: 5A 04 04 71 40 00 00 0B A9 从寄存器区 0x00 地址开始读取 4 个寄存器数据。		
0x0F	写 HGUS 寄存器	字节长度	0x00-0xFF
	Tx: 5A 0F 00 03 00 00 02 00 Rx: 5A 0F 00 03 00 00 A8 E0 01 A8 20 向寄存器区 0x03 地址写入 2 个寄存器数据。		
0x10	写 HGUS 变量存储器	字长度	0x0000-0x6FFF
	Tx: 5A 10 00 00 00 02 04 31 Rx: 5A 10 00 00 00 02 4C E3 32 33 34 6D 9E 向变量存储区 0x0000 地址开始写 2 个变量数据		
0x10	写 HGUS 曲线缓冲区	字长度	地址高字节为 0xFF, 表示数据将写入 HGUS 曲线缓冲区, 此时低字节地址表示为曲线数据格式。
	Tx: 5A 10 FF 01 00 02 04 3F Rx: 5A 10 FF 01 00 02 2D 37 FE 20 00 27 96 向曲线 1 缓冲区写 2 个数据		