

VGUS 串口屏用户开发指南

VTc/SDWb/SDWn 系列适用



电话: 027-87617912 / 87617960 / 87596062

网址: www.viewtech.cn

中显

修改记录

日期	修改内容
2015-6-1	初稿。
2015-8-15	增加音视频播放功能、增加二维码显示功能。
2016-4-7	增加文本滚动显示功能。
2017-06-20	增加第7章“VGUS 开发工具使用说明”及虚拟串口屏。
2018-11-30	增加附2 系统自检寄存器 0xf3-0xf6 描述
2019-08-21	修改第7章内容 VGUS 开发工具使用说明
2020-04-17	在屏参配置7.2节中增加选择触摸屏类型 删除屏参配置读写的描述 删除在线下载的描述 增加SDWn系列串口屏的相关描述
2020-7-2	增加上电串口外发版本号指令的选项功能
2020-12-1	增加SDWb系列串口屏的相关描述
2021-9-13	增加进度条控件、翻页切图动画属性描述
2022-6-29	增加圆形进度条相关描述
2022-11-8	增加屏参配置功能
2023-05-31	SDWb系列增加串口在线下载功能
2023-10-18	增加VTc系列相关说明
2023-11-27	增加“5.5 弹出窗口”，介绍了弹窗中支持的显示控件和触摸控件种类，以及典型应用。
2024-4-17	增加自适应二维码控件相关描述。
2024-10-10	VTc系列增加高级控件：滑动窗口、嵌入窗口、滚轮、下拉列表、多语言文本标签、开关控件。
2024-12-16	SDWb串口屏升级为VTb固件后支持六款高级控件。
2025-05-09	VTc/VTb系列增加数据记录采样、数据记录表格、数据曲线控件相关描述。
2025-08-18	VTc/VTb增加数值显示、多功能按钮、键盘控件相关描述。
2025-11-8	VTc/VTb增加掉电保存/密码登录、动态更新jpg/png图片、矢量字库、数据表格控件等功能。
2025-12-19	VTc/VTb文字标签及中英文键盘控件增加密码(*)显示功能； VTc/VTb多功能按钮控件的密码切换页面模式，增加密码变量地址和初始值。

目 录

1 VGUS 组态串口屏概述	6
1.1 产品分类	6
1.2 认识组态	7
1.3 存储空间	9
2 串口指令集	10
2.1 格式约定	10
2.2 寄存器读写指令 0x80、0x81	11
2.3 变量存储器读写指令 0x82、0x83	12
2.4 扩展指令 0x85	13
2.5 CRC 校验	14
3 寄存器	16
3.1 寄存器一览表	16
3.2 常用寄存器	19
3.2.1 读取版本号	19
3.2.2 修改背光亮度	19
3.2.3 控制蜂鸣器	19
3.2.4 切换图片	19
3.2.5 按键触发	19
3.2.6 时钟 RTC	20
3.2.7 触摸屏校准	20
3.2.8 屏参配置读写	21
3.2.9 语音播放寄存器	22
3.2.10 视频播放寄存器	23
3.2.11 用户数据文件	24
3.2.12 数据库读写	25
4 显示控件	26
4.0 显示控件介绍	26
4.1 数据与文本变量	29
4.1.1 数据变量	29
4.1.2 文本变量	30
4.1.3 滚动文本	31
4.2 图片与图标	32
4.2.1 变量图标	32
4.2.2 动画图标	33
4.2.3 动画图片	34
4.2.4 滑动刻度	35
4.2.5 艺术字变量	36



4.2.6 旋转图标	37
4.2.7 位变量图标	38
4.2.8 进度条显示	40
4.2.9 圆形进度条显示	42
4.3 图形变量	44
4.3.1 实时曲线（趋势图）	44
4.3.2 基本图形（绘图）	46
4.3.3 列表显示（表格）	51
4.3.4 二维码	52
4.3.5 自适应二维码	53
4.4 实时时钟 RTC 的显示	54
4.5 视频控件	57
4.6 摄像头控件	58
5 触摸控件	59
5.0 触摸控件介绍	59
5.1 基本按钮	61
5.1.1 按钮	61
5.1.2 按钮键值返回	62
5.1.3 按钮状态返回	62
5.2 弹出菜单	64
5.3 键盘录入	66
5.3.1 制作键盘	66
5.3.2 数据录入	67
5.3.3 英文 ASCII 录入	69
5.3.4 中文 GBK 录入	70
5.3.5 时钟设置	72
5.4 调节控件	73
5.4.1 增量调节	73
5.4.2 拖动调节	74
5.4.3 转动调节	75
5.4.4 硬件参数配置	76
5.4.5 滑动翻页	77
5.5 弹出窗口	78
6 高级控件	79
6.1 高级控件汇总	79
6.2 数值显示	80
6.3 文字标签	82

6.4 键盘控件	85
6.5 多功能按钮	87
6.6 下拉列表	89
6.7 滚轮	91
6.8 开关	93
6.9 滑动窗口	94
6.10 嵌入窗口	96
6.11 数据记录采样	98
6.12 数据记录表格	100
6.13 数据曲线	103
6.14 数据表格	107
6.15 变量掉电存储	112
6.16 矢量字库	113
6.17 变量存储器 JPG/PNG 图标显示	115
7 描述指针	117
7.1 描述指针介绍	117
7.2 描述指针应用举例	117
7.3 描述指针汇总	120
8 VGUS 开发工具使用说明	159
8.1 操作界面	159
8.2 屏参配置	161
8.3 设计流程	163
8.4 工程下载	165
8.4.1 脱机下载	165
8.4.2 在线下载	166
8.4.3 追加下载	167
8.5 文件格式	168
附 1 系统自检寄存器	172
附 2 固件更新	173

1 VGUS 组态串口屏概述

1.1 产品分类

项目	VTc ^[1]	SDWb	SDWn ^[3]
尺寸范围	10.4 寸以下	21 寸~2.8 寸	4.3 寸以下
开发工具	统一的开发工具 VGUS2025		
脱机下载	TF 卡	U 盘/TF 卡	TF 卡
串口下载 New!	支持	支持	不支持
追加下载 New!	支持	支持	不支持
FLASH 内存	8M/16M 字节	8M/16M/32M/128M 字节	8M 字节
系统占用内存	1.2M 字节	1M/1M/1M/4.5M 字节	0.2M 字节
数据库	64K 字节	64K 字节	不支持
变量存储器	128K 字节	128K 字节	2K 字节
每页变量个数	256	256	32
描述指针	支持	支持	不支持
Jpg 图片/图标	支持	支持	不支持
png 图标	支持	支持	不支持
ADZK 字库抗锯齿	支持	支持	不支持
弹窗叠加变量 New!	支持	支持	不支持
UTF-8 编码 New!	支持	支持	不支持
Lua 脚本编程	不支持	支持	不支持
WAV/MP3 音频	不支持	支持	不支持
AVI 视频控件	不支持	支持	不支持
摄像头控件	不支持	支持	不支持
动画效果切换页面	不支持	支持	不支持
数值显示 New!	支持	升级 ^[2]	不支持
多语言文字标签 New!	支持	升级 ^[2]	不支持
键盘控件 New!	支持	升级 ^[2]	不支持
多功能按钮 New!	支持	升级 ^[2]	不支持
下拉列表 New!	支持	升级 ^[2]	不支持
滚轮 New!	支持	升级 ^[2]	不支持
开关 New!	支持	升级 ^[2]	不支持
嵌入窗口 New!	支持	升级 ^[2]	不支持
滑动窗口 New!	支持	升级 ^[2]	不支持
数据记录采样 New!	支持	升级 ^[2]	不支持
数据记录表格 New!	支持	升级 ^[2]	不支持
数据曲线 New!	支持	升级 ^[2]	不支持
密码登录/掉电保存 New!	支持	升级 ^[2]	不支持
矢量字库 New!	支持	升级 ^[2]	不支持
动态更新 jpg/png 图片 New!	不支持	升级 ^[2]	不支持

[1] VTc串口屏变量存储器空间为128K字节, 仅不支持音视频、滑动翻页、Lua脚本等功能;

[2] SDWb串口屏固件升级为VTb后支持高级控件, VTb固件不支持滑动翻页等功能;

[3] SDWn串口屏变量存储器空间为2K字节, 仅支持部分寄存器、控件功能, 详细信息参考表3-1-1、表4-0-1、表5-0-1.



1.2 认识组态

VGUS 组态串口屏采用变量驱动，所有显示的数字、字符、图标等都定义为一个变量，并通过上位机软件分配变量存储地址、定义显示格式（如字体、字号大小、颜色、显示位置等），然后生成配置文件并下载保存到串口屏里。需要刷新显示某个变量时，用户单片机仅需将变量内容和变量存储地址通过串口对应发送给串口屏，串口屏会自动按照定义好的显示格式显示。

和过去指令集串口屏相比，VGUS 组态串口屏将显示格式与变量内容分开，在刷新变量时用户单片机只需要发送变量内容即可，而传统的指令集串口屏是要把变量内容和显示格式一起发送的。

VGUS 组态串口屏大大简化了指令数量和格式，仅仅需要 4 条指令 0x80-0x83 就可以完成所有控制和变量显示功能了。

举例：在1号界面上显示浮点型变量 “25.3”， VGUS组态串口屏实现步骤：

- **Step1 显示配置：**首先通过VGUS组态软件在1号界面上添加一个“数据变量”，设置“数据变量显示格式”的基本属性(包括变量存储地址、显示颜色、大小，显示方式等)，如右所示。然后生成配置文件，并下载到串口屏里。
- **Step2 运行并刷新显示：**需要刷新显示时，只需要通过82指令将变量内容连同变量存储地址一起发送给串口屏即可。

82指令数据格式: A5 5A 05 82 00 00 00 FD

A5 5A: 串口数据帧帧头（帧头内容用户可以自定义）
05: 数据长度
00 00: 用户定义的25.3这个变量的变量存储地址
00 FD: 25.3这个变量（小数点是定点显示，即数据“253”的16进制）

区域范围设置	
X坐标	115
Y坐标	179
宽度	144
高度	55

变量属性	
名称定义	数据变量显示
描述指针(0x)	FFFF
变量存储地址(0x)	0000
初始值	0
变量类型	int (2Byte)
整数位数	2
小数位数	1
文本颜色	255; 0; 0
字库位置	0
X方向点阵数	12
对齐方式	居中

说明：如果切换到2号界面后，再回到1号界面，串口屏会自动从变量存储器里读取变量内容并显示。

图1-2-1 组态屏

VGUS 开发工具是款组态型、用户图形界面设计软件，VGUS 屏系统架构如图 1-2-2 所示。

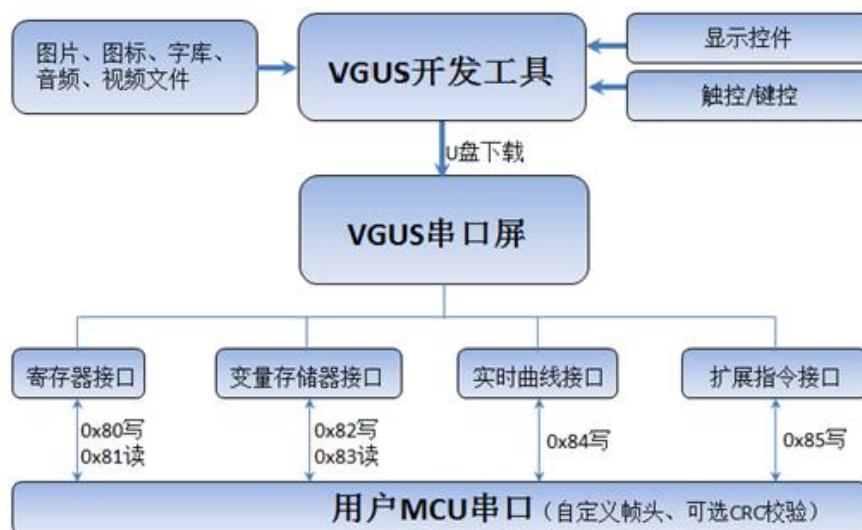


图 1-2-2 VGUS 屏系统架构

所有用到的图片、图标、字库、音频文件需要导入到 VGUS 开发工具中，然后在开发工具中设计显示控件、触摸按钮等，设计完毕后生成下载文件夹并通过 TF 卡下载到串口屏里。

VGUS 串口屏与用户 MCU 之前通过串口连接，通过 0x80 和 0x81 指令实现控制、设置等相关功能，通过 0x82、0x83 指令实现显示和触摸等相关功能。指令详细介绍见第 2 章内容。

VGUS 屏开发只需要三个步骤，便可完成复杂的人机交互设计，如图 1-2-3。详细信息见第 7.3 节。

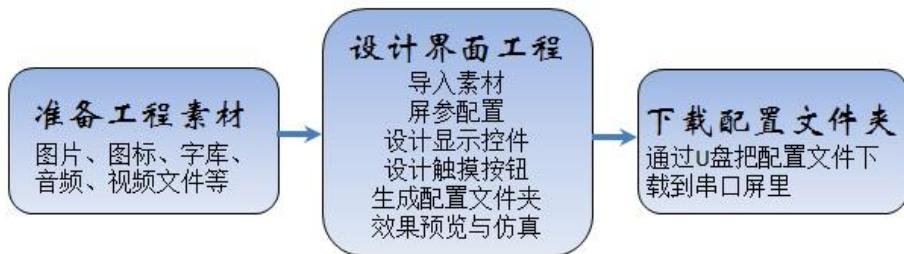


图 1-2-3 VGUS 屏开发步骤

1.3 存储空间

为了顺利学习 VGUS 组态串口屏的使用方法, 本节内容必须充分理解。

VGUS 屏内部存储空间分为三个部分: 寄存器区、变量存储器区和 Flash 存储器。

寄存器区共 256 个单元 (地址 0x00~0xFF), 每个单元对应一个字节, 共 256 字节。用于硬件配置和控制操作, 典型功能如背光亮度调节、版本信息读取、发串口指令控制图片显示、音频播放控制、视频播放控制等。用户单片机可以分别通过 0x80、0x81 指令写入和读取寄存器单元内容。**寄存器详细介绍见第三章内容。**

变量存储器区共 64K 单元 (地址 0x0000~0xFFFF), **每个单元对应两个字节, 共 128K 字节 (SDWn 系列共 1K 单元, 地址为 0x0000~0x03FF, 共 2K 字节)**。顾名思义, 变量存储器就是用来存储变量值的。用户单片机分别通过 0x82、0x83 指令写入和读取变量存储器单元内容。**变量存储器详细介绍见第四章、第五章内容。**

Flash 存储器 8M-512M 字节, SDWb 系列可提供 8M/16M/32M/128M/512M 字节等内存容量, VTc 系列 16M 字节, SDWn 系列 8M 字节。Flash 存储器主要用于存储配置文件 (如变量显示格式)、图片文件、图标文件、字库文件、音频文件、视频文件等。在界面工程设计完毕后, 以上内容通过 TF 卡下载到串口屏里并永久保存。VGUS 屏采用了 JPG 数据压缩算法和连续存储模式, 极大地提高了存储效率和存储图片的数量, 对于一般图片应用场合已经足够, 例如分辨率为 1024*768 的满幅面图片, 128M 字节大约可以存储近千张 (实际存储张数与图片色彩有关)。

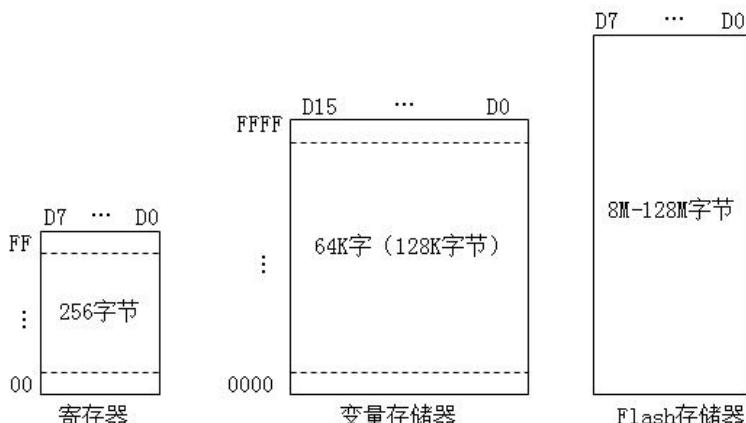


图 1-4-1 存储空间

表 1-4-1 寄存器与变量存储器的区别

	寄存器	变量存储器
容量	256 字节	64K 字 (128K 字节)
地址范围	00-FF	0000-FFFF
单元大小	一个单元对应一个字节	一个单元对应两个字节
读写指令	写 0x80/读 0x81	写 0x82/读 0x83
功能	单元功能已经定义好, 用于硬件配置和控制操作, 典型功能如背光亮度调节、版本信息读取、发串口指令控制图片切换、音频播放控制、视频播放控制等。	单元功能由用户定义, 用于存储变量值。在使用 VGUS 开发工具设计变量时, 用户要指定每个变量对应的存储地址 (定义变量存储器单元功能)。不同的变量分配的存储地址不能冲突。

2 串口指令集

VGUS 屏采用变量驱动模式工作，屏的工作模式和GUI状态完全由变量和寄存器来控制。相应的，串口指令也只需要对变量和寄存器进行读、写即可，一共6条指令。

VGUS屏采用异步、全双工串口（UART），每字节数据传送采用10个位：1位起始位、8位数据位、1位停止位，无校验位。串口波特率可通过VGUS 开发工具中“屏参配置”设置。

2.1 格式约定

➤ 数据格式

VGUS屏支持整数（双字节）、无符号整数（双字节）、长整数（4字节）、超长整数（8字节）等数据格式，其数值范围如表2-1所示。

表2-1 数值范围

数据格式	最小值	最大值
整数（双字节）	-32768（0x8000）	+32767（0x7FFF）
无符号整数（双字节）	0（0x0000）	65535（0xFFFF）
长整数（4字节）	-2147483648（0x80000000）	+2147483647（0x7FFFFFFF）
超长整数（8字节）	-9223372036854775808	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。

➤ 色彩定义

VGUS 屏所有颜色数据均为 16 位，两个字节，如表 1-1 所示，其格式为 Red5-Green6-Blue5，即红色占高 5 位，绿色占中间 6 位，蓝色占低 5 位。可以显示的颜色为 2^{16} 色，即 65536 色。

表 2-2 色彩定义

D15	D14	D13	D12	D11	D10	D9	D8	D7	D6	D5	D4	D3	D2	D1	D0
R4	R3	R2	R1	R0	G5	G4	G3	G2	G1	G0	B4	B3	B2	B1	B0
红色					绿色					蓝色					

例：红色：0xF800，绿色：0x07E0，蓝色：0x001F，白色：0xFFFF，黑色：0x0000

➤ 指令帧结构

所有的串口指令或者数据采用16进制（HEX）格式。对于双字节数据采用高字节先传送（MSB）的方式。例如0x1234发送时先发送0x12，再发送0x34。

表2-3 指令帧结构

指令帧头	指令长度	指令	起始地址	[数据长度]	[数据内容]	[CRC 校验码]
------	------	----	------	--------	--------	-----------

每条指令都包括有指令帧头、指令长度、指令、起始地址四个部分。[数据长度]、[数据内容]、[CRC校验码]这三个参数不是每条指令都有。

指令帧头为双字节，其内容用户可以自定义（寄存器R3、RA），通过VGUS开发工具中“屏参配置”设置。用户可以通过设定不同的帧头，实现多台VGUS屏的并联应用。



指令长度为单字节，包括指令及后面所有的内容。

指令为单字节，包括两条寄存器读写指令(0x80、0x81)、两条变量存储器读写指令 (0x82、0x83) 、一条写曲线缓冲区指令0x84、一条扩展功能指令0x85。

2.2 寄存器读写指令 0x80、0x81

VGUS屏有256Byte的寄存器，主要用于相关硬件控制操作，按照字节（Byte）寻址。

起始地址对于寄存器读写指令(0x80、0x81)是单字节，地址范围为0x00~0xff。

数据长度(N)以字节为单位，表示连续读寄存器单元个数。

支持顺序读写多个寄存器。

表2-4 串口发送0x80指令帧结构（写寄存器）

定义	指令帧头	指令长度	指令	起始地址	数据内容
长度（字节）	2	1	1(0x80)	1	N

举例：寄存器0x03和0x04单元连续写入0x00 、0x01

发送：0xA5 0x5A 0x04 0x80 0x03 0x00 0x01

表2-5 串口发送0x81指令帧结构（读寄存器发送）

定义	指令帧头	指令长度	指令	起始地址	数据长度
长度（字节）	2	1	1(0x81)	1	1(N)

表2-6 串口接收0x81指令帧结构（读寄存器接收，串口屏发送的）

定义	指令帧头	指令长度	指令	起始地址	数据长度	数据内容
长度（字节）	2	1	1(0x81)	1	1(N)	N

举例：连续读取寄存器寄存器0x03和0x04单元

发送：0xA5 0x5A 0x03 0x81 0x03 0x02

返回：0xA5 0x5A 0x05 0x81 0x03 0x02 0x00 0x01

表2-7 寄存器读写指令详解

功能	指令	数据	说明
访问 寄存器 接口	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VGUS屏应答
	VGUS屏有256Byte的寄存器，主要用于相关硬件控制操作，按照字节（Byte）寻址。		



2.3 变量存储器读写指令 0x82、0x83

VGUS屏有64K word (128K Byte) 的变量存储器, 按照字 (word) 寻址操作, 地址为0x0000-0xFFFF。用户在规划变量时, 要根据变量长度手工分配变量存储器地址。

起始地址对于变量存储器读写指令 (0x82、0x83) 是双字节, 地址范围为0x0000~0xffff。

数据长度 (N)以字 (双字节) 为单位, 表示连续读变量存储器单元个数。

数据内容是2N字节, 因为每个变量存储单元包含2个字节。

支持顺序读写多个变量存储器单元。

表2-8 串口发送0x82指令帧结构 (写变量存储器单元)

定义	指令帧头	指令长度	指令	起始地址	数据内容
长度 (字节)	2	1	1(0x82)	2	2N

举例1: 变量存储器0x0003单元写入0x000 、 0x01

发送: 0xA5 0x5A 0x05 0x82 0x00 0x03 0x00 0x01

举例2: 连续写入多个变量地址, 变量存储地址0x0003单元写入0x00、0x01, 0x0004单元写入0x00 、 0x02

发送: 0xA5 0x5A 0x07 0x82 0x00 0x03 0x00 0x01 0x00 0x02

表2-9 串口发送0x83指令帧结构 (读变量存储器发送)

定义	指令帧头	指令长度	指令	起始地址	数据长度
长度 (字节)	2	1	1(0x83)	2	1(N)

表2-10 串口接收0x83指令帧结构 (读变量存储器接收, 串口屏发送的)

定义	指令帧头	指令长度	指令	起始地址	数据长度	数据内容
长度 (字节)	2	1	1(0x83)	2	1(N)	2N

举例: 连续读取变量存储器0x0003和0x0004单元

发送: 0xA5 0x5A 0x04 0x83 0x00 0x03 0x02

返回: 0xA5 0x5A 0x08 0x83 0x00 0x03 0x02 0x00 0x01 0xff 0xff

表2-11 变量存储器读写指令详解

功能	指令	数据	说明
访问 变量 存储器 接口	0x82	下发: 变量地址 (0x0000-0xFFFF) +写入的变量数据	指定变量地址开始写入数据 (字数据) 到变量存储区
	0x83	下发: 变量地址 (0x0000-0xFFFF) +读取变量数据字长度 (0x00-0x7D)	从变量存储区指定地址开始读入指定字长度的数据
		应答: 变量存储器地址+变量数据长度+读取的变量数据	读数据存储器的 VGUS 应答
	VGUS 屏采用变量驱动方式, 将变量数值和变量显示格式分开。变量显示格式是预先通过配置文件形式下载在显示终端里。而变量数值是通过串口实时传送给显示终端的, 变量存储器就是用来存储接收到的变量数值。		
VGUS 屏有 64K word (128K Byte) 的变量存储器, 按照字 (word) 寻址操作, 地址为 0x0000-0xFFFF。用户在规划变量时, 要根据变量长度手工分配变量存储器地址。			



2.4 扩展指令 0x85

表 2-12 直接写显存 0x85 00

定义	指令帧头	指令长度	指令	数据	说明
长度 (字节)	2	1	2 (85 00)	X(2byte)+Y(2byte) +DATA0(2byte)+... +DATA _n (2byte)	X 为起始位置的 X 坐标, Y 为起始位置的 Y 坐标; DATA 为像素点颜色值, 双字节。写入数据超过一行时自动换行。

注意: VTc/VTb 系列不支持该功能。

表 2-14 连续播放音频文件 0x85 03

定义	指令帧头	指令长度	指令	数据	说明
长度 (字节)	2	1	2 (85 03)	Mode(1byte)+ NUM0(2byte)+... +NUM _n (2byte)	Mode 定义播放模式: 0 循环播放, 1 顺序播放, 其他停止播放; NUM 指定音频文件名编号 (双字节, 取值范围 0x0000~0xFFFF), 如果设为不存在的音频文件名将直接跳过。

为提高播放效果, 指令中的音频文件格式必须一致, 如同为 WAV 或者 MP3 格式。**WAV 格式必须用 22KHz、单声道数据, 指令中音频文件总大小不得大于 2M。 MP3 格式音频文件总大小没有限制。**

0x85 03 指令连续播放指定的多个语音文件。该指令**可用于实现窗口/柜台的数字播报 (音频文件必须用 WAV 格式), 也可以用于娱乐应用 (音频文件必须用 MP3 格式)。**



2.5 CRC 校验

用户可以自定义是否启用CRC检验，通过VGUS开发工具中“屏参配置”设置。启用CRC校验后，所有指令尾部将增加一个双字节的CRC校验码。

CRC校验不包括指令帧头和指令长度，采用ANSI-16（X16+X15+X2+1）格式。

举例：带CRC校验方式将寄存器0x03和0x04单元连续写入0x00、0x01

发送: 0xA5 0x5A 0x06 0x80 0x03 0x00 0x01 0x18 0x24

定义	帧头	指令长度	指令	起始地址	数据内容	CRC 校验码
举例	0xA5,0x5A	0x06	0x80	0x03	0x00, 0x01	0x18,0x24

当同时启用CRC校验和CRC校验结果应答后（通过VGUS开发工具中“屏参配置”设置），VGUS屏会在CRC校验后自动通过串口外发校验结果。CRC检验结果应答帧结构如表2-16所示。

表2-16 CRC校验结果应答帧结构

定义	帧头	数据长度	指令	数据	CRC 校验码
长度 (字节)	2	1	1	1	2
说明	用户自定义	固定为 0x02	接收到的指令	0xff 表示 CRC 校验正确 0x00 表示 CRC 校验错误	
举例	0xA5,0x5A	0x02	0x80	0xff	0x20,0x30

CRC 校验程序

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} ;

```
uint16_t CRC16(uint8_t * pMsg, uint16_t Len)
{
    uint8_t i;
    uchCRCHi = 0xFF;
    uchCRCLo = 0xFF;
    while (Len--)
    {
        i = uchCRCHi ^ (*pMsg++);
        uchCRCHi = uchCRCLo ^ auchCRCHi[i];
        uchCRCLo = auchCRCLo[i];
    }
    return (uchCRCHi << 8 | uchCRCLo);
}
```



3 寄存器

VGUS 屏内部都设计有 256 字节的寄存器，用于实现硬件相关操作，如背光调整、时钟读写、语音播放等。

3.1 寄存器一览表

表 3-1-1 寄存器汇总表

寄存器地址	定义	R/W	字节长度	说明	VTc	SDWb	SDWn
0x00	Version	R	1	VGUS 版本号, BCD 码表示, 0x40 表示 V4.0	✓	✓	✓
0x01	Brightness	R/W	1	背光亮度控制寄存器, 0x00-0x40	✓	✓	✓
0x02	BZ_TIME	W	1	蜂鸣器鸣叫控制寄存器, 单位 10ms	✓	✓	✓
0x03-0x04	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 表示触摸屏坐标有更新; 其他表示触摸屏坐标未更新	✓	✓	✓
0x06	TP_Status	R	1	0x01 表示第一次按下; 0x03 表示一直按下中; 0x02 表示抬起; 其他无效	✓	✓	✓
0x07-0x0A	TP_Position	R	4	触摸屏按压坐标位置 X_H:L Y_H:L	✓	✓	✓
0x0B	TPC_Enable	R/W	1	0x00 表示触控不启用; 其他表示触控启用 (上电默认 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	✓	✓	x
0x10-0x1D	保留		14	保留			
0x1F-0x26	RTC_COM_ADJ	W	1	0x5A 表示用户串口申请改写 RTC 数据, VGUS 修改 RTC 后清零	x	✓	x
	RTC_NOW	R/W	7	YY:MM:DD:WW:HH:MM:SS, 年:月:日:星期:时:分:秒, 格式为 BCD 码	x	✓	x
0x27-0x3F	保留		16	保留			
0x40-0x49	En_Lib_OP	R/W	1	0x5A 表示申请用户数据文件加载到变量空间, VGUS 操作完后清零。每个 VGUS 周期执行一次读操作	✓	✓	x
	Lib_OP_Mode	W	1	固定为 0xA0	✓	✓	x
	Lib_ID	W	1	指定用户数据文件名 0x00-0xFF	✓	✓	x
	Lib_Address	W	3	指定数据操作首 (字) 地址, 0x00:00:00-0x01:FF:FF	✓	✓	x
	VP	W	2	指定变量空间的数据操作 (首) 地址 0x0000-0xFFFF	✓	✓	x
	OP_Length	W	2	操作的 (字) 数据长度, 0x0001-0xFFFF	✓	✓	x
0x4A-0x4E	Timer0	R/W	2	16bit 软件定时器, 单位 4ms, 自减到零停止	设置值和实际运行值之间有 +/-4ms 误差, 比如设置位 2, 实际运行值在 4-12ms 之间	✓	x
	Timer1	R/W	1	8bit 软件定时器, 单位 4ms, 自减到零停止		✓	x
	Timer2	R/W	1	8bit 软件定时器, 单位 4ms, 自减到零停止		✓	x
	Timer3	R/W	1	8bit 软件定时器, 单位 4ms, 自减到零停止		✓	x
0x4F	Key_code	W	1	用户键码, 用于触发配置文件, 0x01-0xFF, 0x00 表示无效 VGUS 处理键码后会自动清零键码寄存器		✓	✓
0x50-0x55	Music_Set	W	1	0x5b:播放语音 0x5c:停止播放		x	x



	Music_Num	W	2	音频播放设定值 (0~4095)	x	✓	x
	Vol_Adj_En	W	1	0x5A 表示申请调整播放音乐的音量	x	✓	x
	Vol	W	1	音量值, 范围 0x00~0x40, 上电默认值是 0x40	x	✓	x
	Vol_Status	R	1	音频播放状态 0x00: 播放停止; 0x01: 播放中	x	✓	x
0x56~0x5F	En_DBL_OP	R/W	1	0x5A 表示用户申请进行数据库操作, VGUS 操作完后清零 每个 VGUS 周期执行一次数据库读或写操作	✓	✓	x
	OP_Mode	W	1	0x50 表示把变量存储器空间数据写入数据库空间 0xA0 表示把数据库空间的数据读入变量存储器空间	✓	✓	x
	DBL_Address	W	4	数据库首 (字) 地址	✓	✓	x
	VP	W	2	变量空间的首 (字) 地址, 0x0000~0xFFFF	✓	✓	x
	OP_Length	W	2	数据库操作的 (字) 长度, 0x0001~0xFFFF	✓	✓	x
0x60~0x6e	Play_Avi_Set	R/W	1	0x5a: 申请播放 avi 视频, VGUS 播放完后清零	x	✓	x
	Avi_Type	W	1	0x00: 单曲播放 VGUS 屏内视频 (默认模式) 0x01: 单曲循环播放 VGUS 屏内视频 0x02: 顺序循环播放 VGUS 屏内视频 0x03: 单曲播放 TF 卡内视频 0x04: 单曲循环播放 TF 卡内视频 0x05: 顺序循环播放 TF 卡内视频 注: 视频文件扩展名必须是*.avi; 单曲播放时文件名必须为阿拉伯数字, 如“123.avi”; 顺序播放时文件名可以为字母+数字, 如“wuhan123.avi”。	x	✓	x
	Play_Position	W	4	视频窗口左上角坐标位置 (X_H, X_L, Y_H, Y_L) 注: (0,0) 表示居中显示。	x	✓	x
	Play_Avi_Num	W	2	通过视频文件名选择播放视频曲目, 最多允许 65536 个视频; 仅用于单曲播放, 顺序播放时无效。	x	✓	x
	Vol_Adj_En	W	1	0x5A 表示申请调整播放视频音量	x	✓	x
	Vol	W	1	播放视频音量值, 范围 0x00~0x3F, 上电默认值是 0x3F。	x	✓	x
	Play_Control	W	1	0x5A: 播放/暂停 对于单曲播放方式, 当播放完当前视频后, 系统自动跳回到当前图片界面。	x	✓	x
	Play_Stop	W	1	0x5A: 停止 执行停止播放视频后, 系统自动跳回到当前图片界面, 也可以按照按钮跳转。	x	✓	x
	Play_Next	W	1	0x5a: 播放下一首(VGUS 处理后清零, 单曲播放时结束)	x	✓	x
	Play_Prev	W	1	0x5A: 播放前一首	x	✓	x
	Play_Status	R	1	0x00=空闲; 0x01=播放中; 0x02=暂停。	x	✓	x
0x6f~0xE8	保留		122	未定义			
0xE9	Scan_Status	R/W	1	读: 0x00 表示触摸屏未处于录入状态 0x01 表示触摸屏录入状态; 写: 如果触摸屏在录入状态下写 0x00, 强制退出录入状态。	✓	✓	✓
0xEA	TPCal_Triger	W	1	写入 0x5A 启动一次触摸屏校准, 校准完成后会被 VGUS 清零。	✓	✓	✓

0xEB	Trendline _Clear	W	1	0x55 表示清除全部 8 条曲线缓冲区数据； 0x56-0x5D 表示分别清除 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 VGUS 清零。	√	√	x
0xEC-0xED	保留		2	保留	√	√	√
0xEE-0xEF	Reset_Triger	W	2	写入 0x5AA5 导致 VGUS 屏软件复位一次	√	√	√
0xF0-0xF2	Release Time	R	3	固件发布日期 YY:MM:DD, 年: 月: 日, 格式为 BCD 码 如 18: 06: 27, 则代表发布日期为 2018 年 6 月 27 日	√	√	√
0xF3-0xF6			4	系统自检寄存器	√	√	√
0xF7-0xF9			3	系统备份寄存器 (sysinitial/config/spi), 内部使用	√	√	√
0xFA-0xFF			6	保留			

3.2 常用寄存器

3.2.1 读取版本号

寄存器地址	定义	R/W	字节长度	说明
0x00	Version	R	1	VGUS 版本号, BCD 码表示, 0x43 表示 V4.3

读取版本信息, 串口下发指令 A5 5A 03 81 00 01

返回 A5 5A 04 81 00 01 43

说明: A5 5A: 帧头

04: 指令的字节长度, 81 00 01 43 共4字节 (不含帧头)

81: 读寄存器指令

00: 寄存器地址

01: 返回数据的字节长度, 43为1字节

43: 返回的数据信息

3.2.2 修改背光亮度

寄存器地址	定义	R/W	字节长度	说明
0x01	Brightness	R/W	1	背光亮度控制寄存器, 0x00-0x40

例如, 关闭背光 A5 5A 03 80 01 00

建议: 1、设备对功耗要求较高时, 可以通过修改屏的背光亮度达到降低功耗的目的;

2、开机亮度以及屏保功能可以通过屏参配置设置。

3.2.3 控制蜂鸣器

寄存器地址	定义	R/W	字节长度	说明
0x02	BZ_TIME	W	1	蜂鸣器鸣叫控制寄存器, 单位 10ms

例如, 控制蜂鸣器鸣叫2s, 发送指令A5 5A 03 80 02 C8

3.2.4 切换图片

寄存器地址	定义	R/W	字节长度	说明
0x03-0x04	PIC_ID	R/W	2	读: 当前显示页面 ID; 写: 切换到指定页面

例如, 切换到2号界面的指令A5 5A 04 80 03 00 02

3.2.5 按键触发

寄存器地址	定义	R/W	字节长度	说明
0x4F	Key_code	W	1	用户键码, 使用用户主板上的键盘来触发显示界面上的按钮 (详见第 5 章), 有效键码范围 0x01-0xFF, 0x00 表示无效。4F 寄存器可以与串口屏外接键盘、触摸屏同时使用。VGUS 处理键码后会自动清零键码寄存器。



例如，画面上有个按钮的键码属性设置为01，当用单片机口发送指令“A5 5A 03 80 4F 01”后，串口屏会响应该按钮的操作，执行该按钮被按下的所有操作。

3.2.6 时钟RTC

寄存器地址	定义	R/W	字节长度	说明
0x1F-0x26	RTC_COM_ADJ	W	1	0x5A 表示用户串口申请改写 RTC 数据，VGUS 修改 RTC 后清零
	RTC_NOW	R/W	7	YY: MM:DD:WW:HH:MM:SS, 年:月:日:星期:时:分:秒，格式为 BCD 码

➤ 串口读取时钟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) : A5 5A 03 81 20 07

读取时间 (HH:MM:SS) : A5 5A 03 81 24 03

时间显示需要使用文本时钟或者表盘时钟控件，详见本文4.4节。

➤ 串口修改 (写/设置) 时钟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。

例如把 RTC 设置为 2015-06-01 星期一 18:56:00，串口发送以下指令：

A5 5A 0A 80 1F 5A 15 06 01 00 18 56 00 VGUS 屏会自动换算星期，改写时间时星期可以写任意值。

3.2.7 触摸屏校准

所有 VGUS 屏出厂前均已经校准，用户使用前无需逐个再次校准。VGUS 屏提供了两种启动触摸屏校准方法，分别适用于不同应用场合。

➤ 方法1——产品集成

用户 MCU 通过串口向 VGUS 屏 0xEA 寄存器写入 0x5A 将启动一次触摸屏校准。通过该方法，可以将触摸屏校准功能作为最终设备的一个选项功能，提供给设备操作人员。该方法适用于用户 MCU 发串口指令使用。

寄存器地址	定义	R/W	字节长度	说明
0xEA	TPCal_Triger	W	1	写入 0x5A 启动一次触摸屏校准，校准完成后会被 VGUS 清零

触摸屏校准发送指令 A5 5A 03 80 EA 5A

➤ 方法2——批量生产

在 VGUS 开发工具“屏参配置”中，勾选“TF 卡下载配置文件后启动触摸校准”，则在每次下载配置文件完毕后，VGUS 屏将立即自动启动一次触摸屏校准。如果客户希望在生产环节，每块屏都再次校准一次，可以使用该方法，该方法适用于批量生产阶段使用。所有 VGUS 屏出厂都已经进行校准，不建议客户再次校准。

无论使用上述哪种方法，一旦触摸屏校准启动后，需要按照屏幕提示操作，依次点击屏幕“左上角”、“左下角”、“右下角”、“右上角”和“中心点”十字交叉点提示的触摸位置；当校准完成时，显示终



端会自动进入触摸测试状态，点击触摸屏可观察触摸位置准确度。

3.2.8 屏参配置读写

对于简单应用场合屏参配置通过 VGUS 开发工具设置实现，参见第 7 章，无需了解本节知识。

屏参配置除了通过 VGUS 开发工具设置外，用户可以通过寄存器动态设置屏保和蜂鸣器伴音功能。

仅 SDWb 系列 2022.11.8 及以后固件版本、VTc 系列支持动态设置屏保和蜂鸣器伴音功能。

寄存器地址	定义	R/W	字节长度	说明
0x12	MODE_CONFIG1	R/W	1	bit7-bit6:未定义 bit5: 0=禁止屏保功能，背光不受触摸屏状态控制 1=启用屏保功能，背光受触摸屏状态控制， 控制参数由0x16、0x 0x17、0x18寄存器设定。 bit4-bit0:未定义
0x16	Rel_Brightness	R/W	1	点亮亮度 (0x00-0x40)。
0x17	Saver_Brightness	R/W	1	屏保亮度 (0x00-0x40)。
0x18	Saver_Delay	R/W	1	屏保延时时间 (单位/s)。
0x1C	MODE_CONFIG2	R/W	1	bit7-bit6:未定义 bit5: 0=点击触摸屏有效区域时有蜂鸣器提示音。 1=点击触摸屏有效区域时无蜂鸣器提示音，但可以通过向0x02寄存器写入数据控制蜂鸣器鸣叫。 bit4-bit0:未定义
0x1D	CONFIG_EN	W	1	写入 0xA5，使上述设置生效。 上述未定义的位设置无效。 注意：上述设置掉电不保存。



3.2.9 语音播放寄存器

VGUS 屏支持 WAV 和 MP3 两种音频文件格式，扩展名分别为*.wav 或*.mp3。WAV 格式占用空间大，语音立即输出无延迟，适合按键伴音、数字播报等较短时间播放场合。WAV 格式文件必须选用 22KHz、单声道数据。MP3 格式占用空间小，语音输出稍有延迟（0.5 秒），适合开机音乐、背景音乐、语音提示等较长时间播放场合。

用户通过VGUS组态软件将语音文件添加到工程中，再通过0x80指令写相关寄存器控制语音播放和进行音量调节。

寄存器地址	定义	R/W	字节长度	说明
0x50-0x52	Music_Set	W	1	0x5b:播放语音 0x5c:停止播放
	Music_Num	W	2	音乐播放设定值（0-4095）
0x53-0x54	Vol_Adj_En	W	1	0x5A 表示申请调整播放音乐的音量
	Vol	W	1	音量值，范围 0x00-0x40，上电默认值是 0x40
0x55	Vol_Status	R	1	音频播放状态 0x00: 播放停止；0x01: 播放中

举例，一段语音（比如“欢迎光临武汉中显”）保存为6.wav，要以100%音量播放，串口下发：

A5 5A 07 80 50 5b 00 06 5A 40

要停止语音播放，串口下发：A5 5A 05 80 50 5c 00 06

当正在播放一首语音过程中，如果接受到新的播放语音指令，将立即结束当前语音并开始播放新的语音内容。

此外，扩展指令0x85 03提供连续播放音频功能，详细参见2.4节内容。

3.2.10 视频播放寄存器

关于视频播放更详细信息请参考文档《VGUS串口屏视频播放功能详解》。

寄存器地址	定义	R/W	字节长度	说明
0x60-0x67	Play_Avi_Set	R/W	1	0x5A:申请设置播放器参数
	Avi_Type	W	1	0x00: 单曲播放 VGUS 屏内视频 (默认模式)
				0x01: 单曲循环播放 VGUS 屏内视频
				0x02: 顺序循环播放 VGUS 屏内视频
				0x03: 单曲播放 TF 卡内视频
0x68-0x69	Play_Position	W	4	0x04: 单曲循环播放 TF 卡内视频
				0x05: 顺序循环播放 TF 卡内视频
	Play_Avi_Num	W	2	注: 视频文件扩展名必须是*.avi;
				单曲播放时文件名必须为阿拉伯数字, 如“123.avi”;
0x6a	Play_Control	W	1	顺序播放时文件名可以为字母+数字, 如“wuhan123.avi”。
				视频窗口左上角坐标位置 (XH, XL, YH, YL)
	Vol_Adj_En	W	1	注: (0, 0, 0, 0) 表示居中显示。
				通过视频文件名选择播放视频曲目, 最多允许 65536 个视频;
0x6b	Play_Stop	W	1	仅用于单曲播放, 顺序播放时无效。
				0x5A: 播放/暂停
	Play_Next	W	1	对于单曲播放方式, 当播放完当前视频后, 系统自动跳回到当前图片界面。
				0x5A: 停止
0x6c	Play_Prev	W	1	执行停止播放视频后, 系统自动跳回到当前图片界面, 也可以按照按钮跳转。
				0x5A: 播放下一首
	Play_Status	R	1	0x5A: 播放前一首
				0x00=空闲; 0x01=播放中; 0x02=暂停。

使用视频寄存器, 可以通过 0x80 写寄存器指令控制和触控两种方式, 来控制视频播放。

方式一: 用户指令控制方式

用户通过 0x80 指令写寄存器 0x60-0x6d, 实现 avi 视频的播放、暂停、继续、停止等功能。

视频寄存器应用举例, 在显示屏 (10, 10) 位置单曲播放 TF 卡内 01 号视频:

首先发送播放参数 A5 5A 0A 80 60 5A 03 00 0A 00 0A 00 01,

然后发送播放视频指令 A5 5A 03 80 6A 5A

方式二: 触控方式

寄存器 0x61-0x6e 映射于变量存储器 0xff01-0xff0e, 即操作用户变量存储器 0xff01-0xff0e 可实现相同的功能 (主要是用来触控实现播放 avi 视频, 无须用户 MCU 指令干预)。在播放器界面上设计制作播



放/暂停、停止、下一曲、音量调节等按钮，通过“按钮键值返回”控件修改变量存储器 0xff01-0xff0e（寄存器 0x61-0x6e），实现无需用户 MCU 干预下播放 avi 视频；

小技巧：使用视频寄存器实现开机自动播放视频“0.AVI”

通过 VGUS 开发工具，在 0xff0a 单元位置设计一个数据变量控件，并将初值设置为 0x5a，将开机动画视频文件名设置为 0.avi，串口屏上电后将自动播放开机动画 0.avi。

注意：播放视频功能时显示屏上变量不再刷新、上一节中语音播放不可用。

3.2.11 用户数据文件

用户数据文件必须命名为“*.bin”，文件名编号为0-255（文件名不可以使用13、14、22和23，系统占用）。

1 通过 TF 卡导入用户数据文件

用户数据文件“*.bin”通过VGUS软件添加到工程里面。通过TF卡下载工程文件时，用户数据文件会一并导入到串口屏里。

2 通过串口读取用户数据文件

通过寄存器0x40-0x49，可以将用户数据文件里面指定位置数据读出加载到变量存储器里面，如表3-5所示。如果用户单片机需要使用，可以使用0x83指令再从变量存储器中读取。

寄存器地址	定义	R/W	字节长度	说明
0x40-0x49	En_Lib_OP	R/W	1	0x5A 表示申请用户数据文件加载到变量空间，VGUS 操作完后清零。每个 VGUS 周期执行一次读操作
	Lib_OP_Mode	W	1	固定为 0xA0
	Lib_ID	W	1	指定用户数据文件名编号 0x00-0xff
	Lib_Address	W	3	指定数据操作首（字）地址，0x00:00:00-0x01:FF:FF
	VP	W	2	指定变量空间的数据操作（首）地址 0x0000-0xFFFF
	OP_Length	W	2	操作的（字）数据长度，0x0001-0xFFFF

例如从“80.bin”用户数据文件的 0x 00 00 00 地址开始读取4KW (0x10 00) 数据到变量存储器 0x1000 单元开始的位置，串口下发指令：

A5 5A 0C 80 40 5A A0 50 00 00 00 10 00 10 00

读取用户数据文件大约需要花费十毫秒级时间，用户单片机发送连续指令时需要注意插入延时。



3.2.12 数据库读写

用户数据库的文件名为“UserDb.bin”，具有掉电存储用户数据的功能。**SDWb/VTc系列串口屏数据库大小固定为64K字节。**

1 通过串口读写数据库

用户单片机使用串口访问数据库，需要通过**变量存储器**实现。通过寄存器0x56-0x5F，可以将变量存储器指定位置的数据写入保存到FLASH里面，也可以将FLASH里面的指定位置数据读出加载到变量存储器里，如下表所示。

寄存器地址	定义	R/W	字节长度	说明
0x56-0x5F	En_DBL_OP	R/W	1	0x5A 表示用户申请进行数据库操作，VGUS 操作完后清零 每个 VGUS 周期执行一次数据库读或写操作
	OP_Mode	W	1	0x50 表示把变量存储器空间数据写入数据库空间 0xA0 表示把数据库空间的数据读出加载到变量存储器空间
	DBL_Address	W	4	数据库首（字）地址
	VP	W	2	变量空间的首（字）地址，0x0000-0xFFFF
	OP_Length	W	2	数据库操作的（字）长度，0x0001-0xFFFF

例如把变量存储器地址 0000-0100 的数据写入数据库地址 0000 0000-0000 0100，数据长度为 256W

(0x01 00)，串口下发指令：A5 5A 0C 80 56 5A 50 00 00 00 00 00 00 00 01 00

把数据库地址 0000 0000-0000 0100 的数据读出保存到变量存储器地址 0000-0100，数据长度为 256W

(0x01 00)，串口下发指令：A5 5A 0C 80 56 5A A0 00 00 00 00 00 00 00 01 00

如果用户单片机需要使用数据库里数据，可以先把数据库读出加载到变量存储器，然后再使用0x83指令从变量存储器中读取，反之亦然。

2 通过 TF 卡导出数据库

将TF卡（Fat32格式）插入VGUS串口屏中，上电后VGUS屏会自动将数据库文件“UserDb.bin”拷贝到TF卡根目录下。出厂默认状态下，串口屏里面没有数据库文件“UserDb.bin”，只有通过寄存器0x56-0x5F执行了数据库写入操作后，串口屏里面才会生成数据库文件“UserDb.bin”。

数据库保存在 FLASH 存储器里面，因此其具有掉电保存功能。不推荐将数据库作为重要数据的存储介质使用，也不推荐频繁修改写入数据库！该类应用建议用户单片机外扩一个 SPI Nor Flash 存储器。

读写数据库大约需要花费十毫秒级时间，用户单片机发送连续指令时需要注意插入延时。



4 显示控件

4.0 显示控件介绍

4.0.1 显示控件汇总

显示控件通过“VGUS开发工具->变量配置工具栏”添加，该工具栏各图标定义如图4-0-1所示。

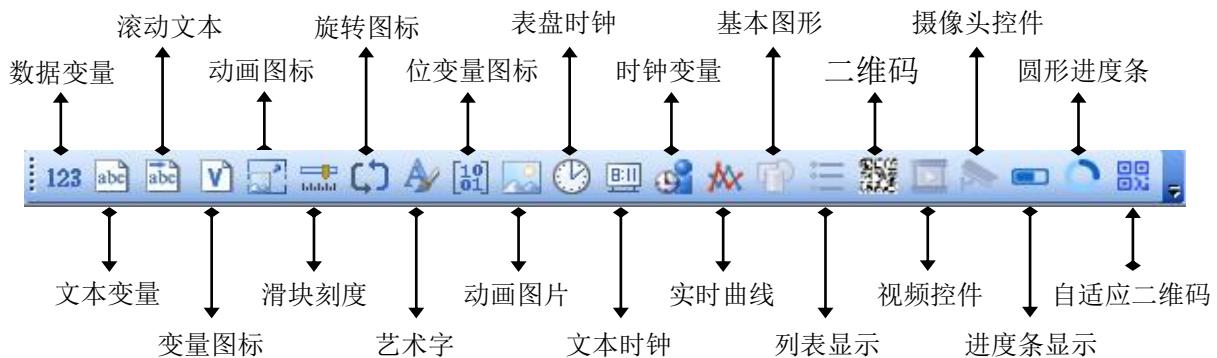


图 4-0-1 变量配置工具栏

表 4-0-1 显示控件汇总表

	功能	说明	VTc	SDWb	SDWn
1	数据变量	把一个数据变量按照指定格式（整数、小数、是否带单位）用指定字体和大小显示出来。	√	√	√
2	文本变量	把字符串按照指定的格式，在指定的文本框显示区域显示。	√	√	√
3	滚动文本	从右向左滚动文本显示。	√	√	x
4	文本时钟	按照用户编辑的格式、用文本方式显示公历 RTC。时钟源来自串口屏 RTC 时钟外设或用户单片机。	√	√	x
5	表盘时钟	用表盘方式显示公历 RTC。时钟源来自串口屏 RTC 时钟外设或用户单片机。	√	√	x
6	时钟变量	hex 格式显示变量存储器数据。支持设置分隔符。	√	√	x
7	变量图标	将数据变量的变化范围对应一组 ICON 图标；当变量变化时图标也自动相应切换。用于仪表板、进度条。	√	√	√
8	动画图标	将一个定值数据变量对应 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。	√	√	√
9	滑块刻度	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。用于液位、刻度盘、进度表的指示。	√	√	√
10	艺术字变量	用 ICON 图标取代字库来显示变量数据。	√	√	√
11	动画图片	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。	√	√	√
12	旋转图标	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，用于指针、仪表板。	√	√	x
13	位变量图标	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。用于开关状态显示，比如风机的运转（动画）、停止（静止图标）。	√	√	√
14	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。	√	√	x
15	基本图形	画点、线、矩形、圆形等，详细见 4.3.2 节。	[1]	[2]	x
16	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。	√	√	x
17	二维码显示	根据指定内容在屏上显示指定的二维码图形信息。	√	√	√
18	自适应二维码	二维码单元大小根据数据量自动缩放。	√	√	x

19	视频控件	播放指定视频文件。	x	√	x
20	摄像头控件	在显示屏指定区域显示摄像头监控画面。	x	√	x
21	进度条	将一个数据变量的变化范围线性对应 ICON 图标或框选范围内单色/切图的部分区域显示; 当变量变化时, 显示区域自动相应调整。	√	√	x
22	圆形进度条	将一个数据变量的变化范围线性对应圆环的角度区域的变化; 当变量变化时, 圆环的角度区域自动相应调整。	√	√	x

[1] VTC 系列不支持基本图形指令 0x0008\0x000D\0x0010。

[2] SDWb 系列固件升级为 VTb 后不支持基本图形指令 0x0008\0x000D\0x0010、不支持列表显示。



4.0.2 显示控件的添加方法

控件设计在VGUS开发工具中完成，所有控件设计都分为2个步骤：第一步：在需要添加相应显示功能的页面下，点击相应控件，把该控件添加到页面上，并拖动控件框调整显示位置和大小。第二步：在该页面上点选已经添加的控件，在右侧窗口中设置控件属性。

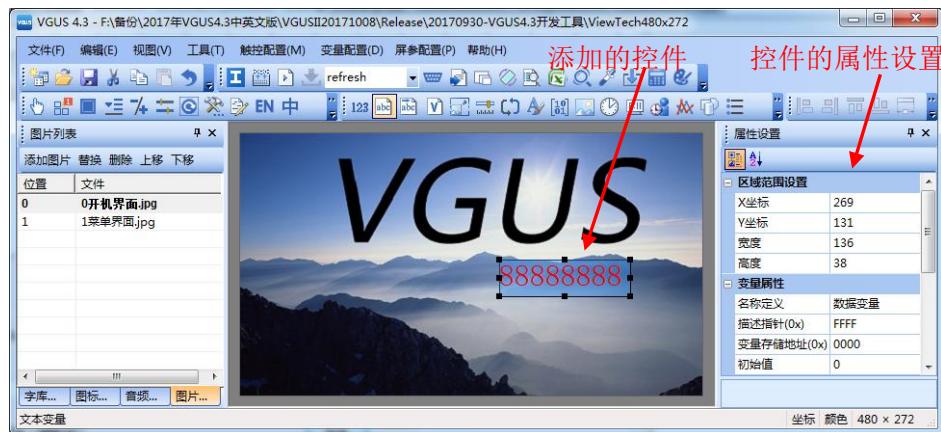


图 4-0-3 点选控件及属性设置举例

4.0.2 显示控件的典型属性

显示控件的典型属性如图4-0-3所示，控件属性分为区域范围设置、变量属性和显示格式三大类。区域范围设置与变量属性对所有控件都是类似，而显示格式对不同显示控件可能完全不同。学习显示控件的用法，务必了解清楚变量存储地址的作用和用法、以及显示格式属性栏的功能。**鼠标点击指定属性，在属性设置窗口下方会显示关于该属性的详细介绍。**

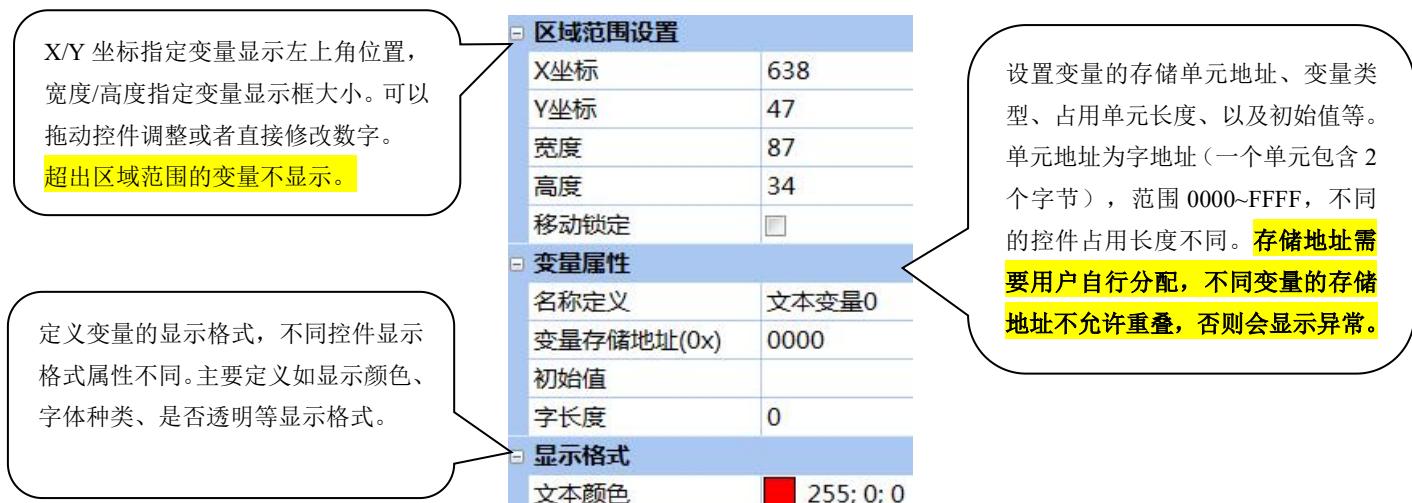


图 4-0-4 显示控件属性说明

4.1 数据与文本变量

串口屏预装了0号字库，包含 4*8到64*128 点阵的所有ASCII码字符。当需要用到其它类型字库时，可以在VGUS开发工具中添加常用字库（宋体、微软雅黑和数码管字体）、或者找技术支持索取、或者自行通过TS3软件生成相应字库文件。新字库需要从1开始命名，不要与0号字库重名。

4.1.1 数据变量

用于整数、小数的显示。小数显示采用定点方式，先化整并设置好整数位数和小数位数（整数位数和小数位数之和不能超过20），例如显示12.345，需要发送数据12345并设置整数位数为2、小数位数为3。

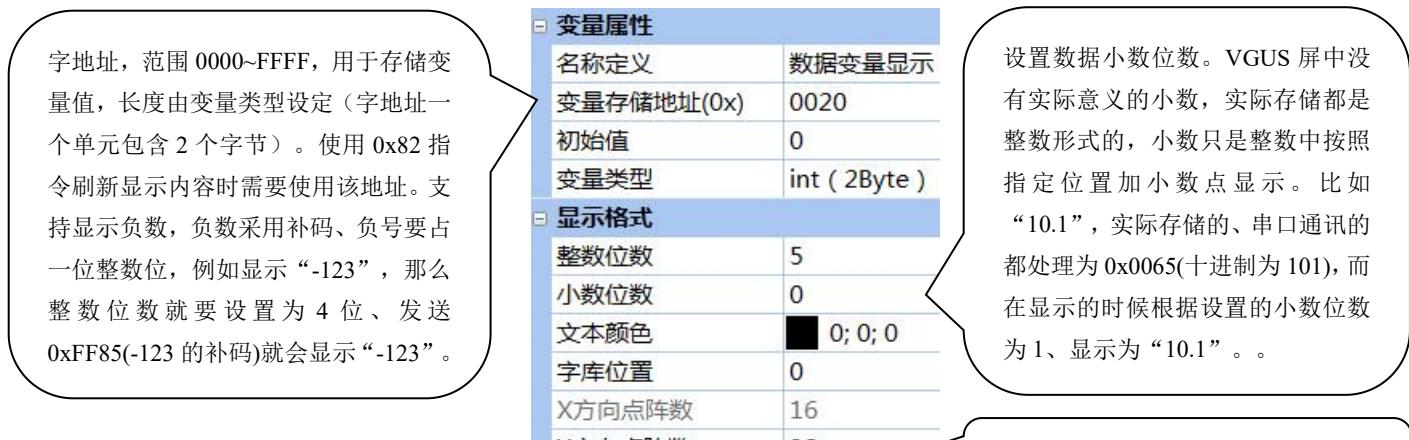


图 4-1-1 数据变量属性设置



图 4-1-2 数据变量举例

例如，按照图 4-1-1 属性设置，在图 4-1-2 中显示整数“1234”，串口发送指令：A5 5A 05 82 00 20 04 D2

说明：A5 5A：帧头

05：指令字节长度，82 00 20 04 D2 共 5 字节（不含帧头）

82：写变量存储器指令

00 20：变量地址，在控件属性中设置

04 D2：数据内容，1234 的十六进制数据

支持负数显示，负数采用补码、且负号要占一位整数位，例如显示“-123”，那么整数位数就要设置为4位、发送0xFF85(-123的补码)就会显示“-123”。



4.1.2 文本变量

用于中英文字符串的显示。对于页面上的固定文字提示信息，建议直接做到背景图片上，一方面在图片上可以设计出更好的字体显示效果，另一方面可以减少页面上的变量，有利于提高串口屏的响应速度。

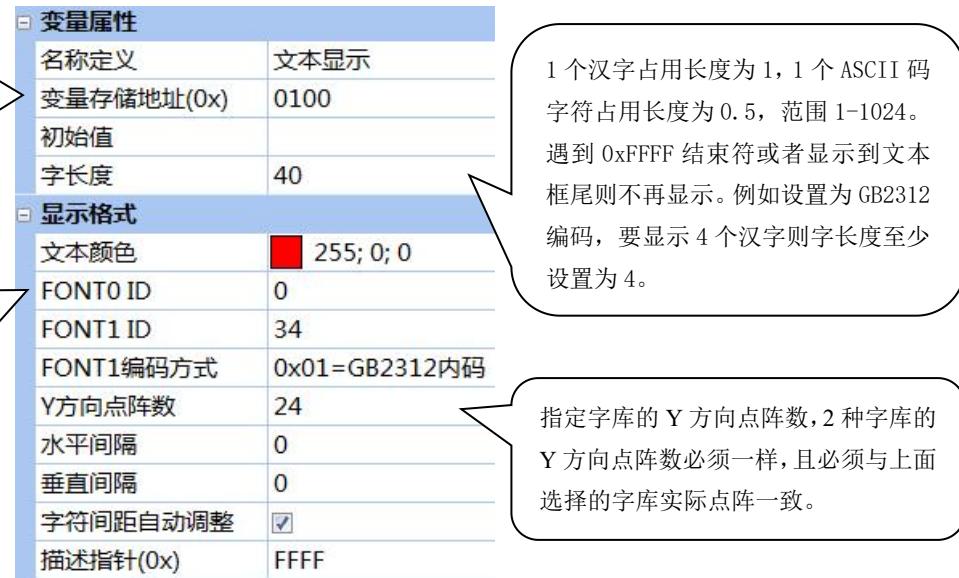


图4-1-2 文本变量属性设置

文本变量允许最长文本长度为1024字(2K字节), 实际文本显示长度取决于以下三个条件: 文本长度限制(属性设置中的字长度)、文本结束符0xFFFF、或者文本框尾, 当遇到上述任意一个条件后, 显示就会结束。因此, 当把字长度设置的大于实际文本显示长度时, 在用0x82指令写入文本显示内容, 必须要在文本结尾的地方写入文本结束符0xFFFF, 否则会在文本尾部显示多余的信息。

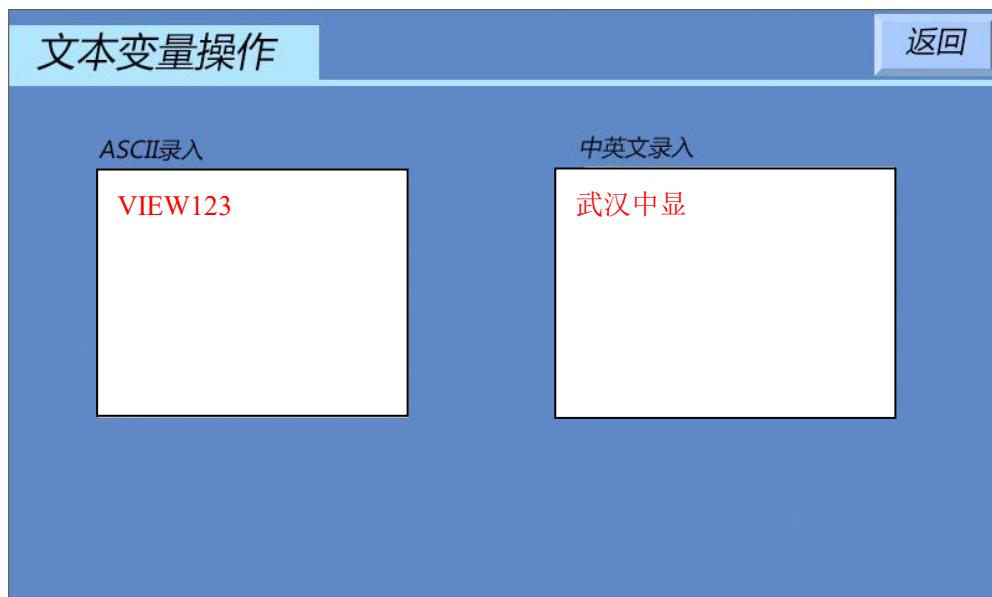


图4-1-3 文本变量举例

例如, 按照图4-1-2属性设置, 在图4-1-3中显示英文加数字“VIEW123”, 串口发送指令:

A5 5A 0C 82 01 00 56 49 45 57 31 32 33 FF FF

说明: A5 5A: 帧头

0C: 指令字节长度, 82 01 40 56 49 45 57 31 32 33 FF FF (不含帧头)

82: 写变量存储器指令

01 00: 变量地址, 在控件属性中设置

56 49 45 57 31 32 33: 数据内容, “VIEW123”的 ASCII 码

FF FF: 文本结束符

例如: 显示中文“武汉中显”, 串口发送指令:

A5 5A 0D 82 01 40 CE E4 BA BA D6 D0 CF D4 FF FF

说明: A5 5A: 帧头

0D: 指令字节长度, 82 01 40 CE E4 BA BA D6 D0 CF D4 FF FF 共 13 字节 (不含帧头)

82: 写变量存储器指令

01 40: 变量地址, 在控件属性中设置

CE E4 BA BA D6 D0 CF D4: 数据内容, “武汉中显”的汉字内码

FF FF: 文本结束符

如果需要显示日文、韩文、俄文等文字, 通常可以使用unicode编码字库。日文也可以使用SJIS编码字库。关于特殊语种显示欢迎联系技术支持。

4.1.3 滚动文本

滚动文本的属性设置与文本变量基本类似, 只是多了滚屏模式选择和滚屏间距设置。需要注意的是:

1. 变量存储地址开始的三个单元没有使用, 文本从“变量存储地址+3”单元开始存储; 例如变量存储单元地址为0x0100, 我们在用0x82指令写入文本时, 要从0x0103单元开始写入。
2. 属性设置中的字长度仅用于VGUS开发工具自动分配变量存储地址和检查单元地址冲突, 实际长度以文本结束标志为准, 遇到文本结束标志0xFFFF后不再显示。
3. 文本实际长度没有限制, 最大允许长度为1024字, 使用0x82指令写入文本时必须以0xFFFF结尾, 作为文本结束标记。

变量属性	
名称定义	滚动文本0
变量存储地址(0x)	0000
初始值	
字长度	0
显示格式	
文本颜色	255; 0; 0
对齐方式	0x00=左对齐
FONT0 ID	0
FONT1 ID	0
FONT1编码方式	0x00=8bit 编码
Y方向点阵数	16
字符间隔	0
字符间距自动调整	<input type="checkbox"/>
滚屏模式	0x00=从右向左滚
滚屏间距	0
描述指针(0x)	FFFF

1 个汉字占用长度为 1, 1 个 ASCII 码字符占用长度为 0.5, 范围为 1-1024 字。该长度仅用于 VGUS 开发工具自动分配变量存储地址和检查单元地址冲突。实际长度以文本结束标志为准, 遇到文本结束标志 0xFFFF 后不再显示, 最大允许文本长度为 1024 字。

字地址, 范围 0000~FFFF, 用于存储变量值, 长度由文本结束标志 0xFFFF 指定 (字地址一个单元包含 2 个字节), 文本实际长度没有限制。使用 0x82 指令刷新显示内容时需要使用该地址。需要注意上文中的三点。

选择文本滚动方向, 仅支持自右向左滚动。

设置每个 VGUS 刷新周期文本滚动的像素点阵数。

图 4-1-4 滚动文本属性设置

4.2 图片与图标

VGUS 屏中，约定将分辨率与串口屏分辨率一致的图片称为“图片”，而将所有分辨率小于串口屏分辨率的图片称为“图标”。

在使用图片和图标相关控件以前，必须先设计要图片和图标素材，正确添加到 VGUS 开发工具的文件列表中。关于图片、图标的文件格式要求和管理等信息，请参见本文 8.5 节《文件格式》。

4.2.1 变量图标

该控件是把图标编号映射到变量存储单元，通过向变量单元里写入不同的变量，然后串口屏根据变量值检索图标编号，再调用对应图标显示。通常用于串口发 0x82 指令方式，来调用不同的图标在相同的位置显示，**例如实现进度条的显示功能、电池电量显示。**

需要注意的是：图标编号不是变量值，图标编号与变量值等比均分对应。变量上下限设定的变量范围与图标数量可以不一一对应，串口屏会自动将变量或者图标数进行均分，例如变量为 0 到 100 而图标为 5 到 14，这样变量 0 到 10 对应的图标都是 5。



图4-2-3 变量图标属性设置



图 4-2-4 变量图标应用举例

例如，按照图 4-2-1 图标素材、图 4-2-3 属性设置，为让上图中左边进度条显示 10 种不同状态，串口可以分别发送指令：

A5 5A 05 82 01 20 00 00

A5 5A 05 82 01 20 00 01

.....

A5 5A 05 82 01 20 00 09

说明: A5 5A: 帧头

05: 指令字节长度, 82 01 20 00 01 共 5 字节

82: 写变量存储器指令

01 20: 设置的变量地址, 如上图变量属性所示

00 00: 设置的变量值, 变量的上下限 0-9 与编号为 5-14 的图标一一对应, 即发送 00 00 显示 5 号图标,

00 01: 显示 6 号图标, 以此类推。

4.2.2 动画图标

该控件是在显示屏某一区域连续循环显示一串图标, 实现动画效果, 动画播放时间间隔为一个 VGUS 刷新周期 (约 100 毫秒)。变量只有 2 个取值: 一个为开始值, 一个为停止值。当通过 0x82 指令向变量单元写入开始值时, 动画开始播放; 当写入停止值时, 动画停止播放。



图 4-2-5 动画图标属性设置

按照图 4-2-5 设置:

当向 0121 单元写入 1 后, 将把 24.ICO 文件中的 15 到 28 号图标循环播放。

当向 0121 单元写入 0 后, 动画停止播放, 画面停留到 15 号图标上。

当向 0121 单元写入其它值时, 不显示图标和动画。



图 4-2-6 动画图标举例

为在上图中间位置显示旋转的“VGUS”动画, 需要准备右侧的图标素材。按照图 4-2-5 所示设置, 为让上图中间动画图标显示动画效果, 可以发送指令:



电话: 027-87617912 / 87617960 / 87596062

网址: www.viewtech.cn

中显

开始动画指令 **A5 5A 05 82 01 21 00 01**

停止动画指令 **A5 5A 05 82 01 21 00 00**

说明: **A5 5A:** 帧头

05: 指令字节长度, 82 01 21 00 01 共 5 字节

82: 写变量存储器指令

01 21: 设置的变量地址, 如上图变量属性所示

00 00: 设置的变量开始值, 发送开始值后循环播放开始图标到结束图标, 发送停止值后显示停止图标。

小技巧: 如果把图 4-2-5 中的初始值设置为 1, 就可以实现开机动画, 上电自动播放。

4.2.3 动画图片

与动画图标不同, 动画图片播放整张图片、播放的起始/终止图片和播放时间间隔可以在属性中设置、播放到终止位置图片后自动结束播放。需要特别注意的是, 动画图片没有分配变量存储地址单元, **显示动画图片控件所在的页面, 将自动启动播放。**



图 4-2-7 动画图片属性设置

上图设置从第 35 号图片开始, 播放到第 49 号图片结束, 播放时间间隔为 $5*10\text{ms}=0.05$ 秒。

启动播放方法: 显示动画图片控件所在的页面, 将自动启动播放。

停止播放方法: 播放到终止图片位置自动停止播放。

循环播放方法: 在终止图片页面上也添加一个属性一样的动画图片控件。

串口发送指令切换图片或者触控按钮切换图片可以结束循环播放。

开机自动播放: 在 0 号图片上 (开机页面上) 添加动画图片控件。

注意: 动画图片控件所在页面无法驻留显示。

4.2.4 滑动刻度

该控件是指定一个图标（滑块），在水平方向或者垂直方向上、在指定范围内变化位置显示（滑动），显示位置通过变量设定。通常可以配合拖动调节控件使用（详见 5.4.2 节）。需要注意的是：变量值不是坐标，滑块显示位置与变量值等比均分对应。起始/终止刻度值设定的变量范围与滑动区域坐标可以不一一对应，串口屏会自动将变量或者滑动区域坐标进行均分后对应。



图 4-2-8 滑动刻度属性设置

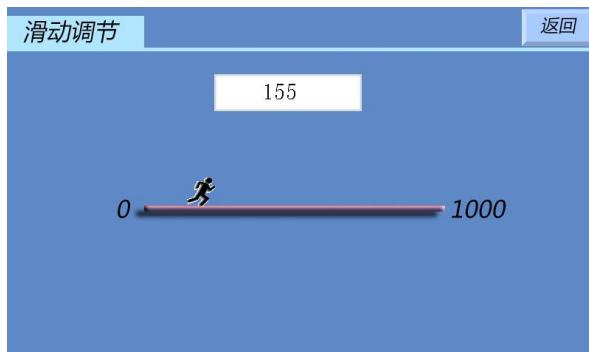


图 4-2-9 滑动刻度举例

为在上图中实现人形滑块显示，可以在横线上方添加一个滑块刻度控件，属性设置如图 4-2-8。让滑动刻度显示在指定位置，串口可以发送指令：

A5 5A 05 82 01 80 00 00

A5 5A 05 82 01 80 00 01

.....

A5 5A 05 82 01 80 03 E7

A5 5A 05 82 01 80 03 E8

说明：A5 5A：帧头

05：指令字节长度，82 01 80 00 01 共 5 字节

82：写变量存储器指令

01 80：设置的变量地址，如上图变量属性所示

00 00：设置的变量刻度值，即从 00 00 变化到 03 E8

4.2.5 艺术字变量

该控件与数据变量应用一样，都是可以实现一个数据显示。与数据变量不同的是，艺术字变量使用图标（“0123456789-.” 共 12 个图标）代替字库，可以实现各种特殊字体效果、特殊颜色效果的数据显示。常见的数码管字体建议使用数据变量实现，显示速度更快。



图 4-2-10 艺术字变量属性设置

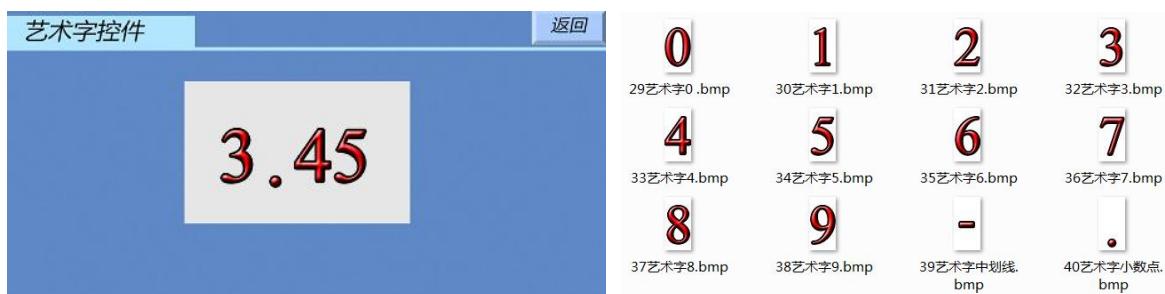


图 4-2-11 艺术字变量举例

为实现上图中的数据显示效果，只能使用艺术字变量。首先要制作图标素材如上图右侧。在文本框位置添加一个艺术字变量，属性设置参照 4-2-10，当向 0160 单元写入“345”后，就可以显示出上图艺术字出来。

4.2.6 旋转图标

该控件控件通常用于仪表盘的显示，需要预先设计一张表盘背景图片和一个指针图标，指针图标根据变量值确定显示角度，指针图标旋转中心坐标可以设置、指针图标旋转中心在表盘背景图片上的显示位置也可以设置。

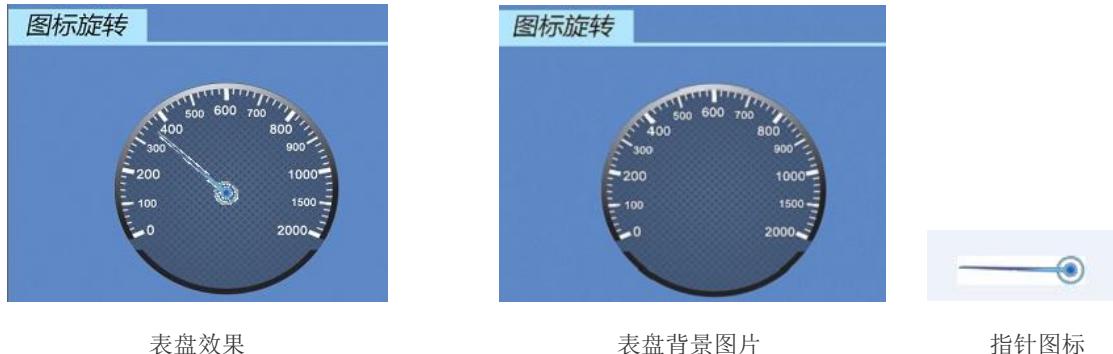


图 4-2-12 旋转图标举例

字地址，范围 0000~FFFF，用于存储变量，长度为一个单元（字地址一个单元包含 2 个字节）。使用 0x82 指令刷新指针（图标）显示角度时需要使用该地址。

变量属性	
名称定义	图标旋转
变量存储地址(0x)	0040
变量模式	指向一个整型变量
初始值	0

显示格式	
图标文件	24.ICO
图标ID	1
图标显示模式	透明
图标旋转中心	(122,15)
屏幕旋转中心	(272,264)
起始值	0
终止值	2000
起始角度	0
终止角度	400
描述指针(0x)	FFFF

变量最小、最大值。

选择指针图标。

变量最大值对应的旋转角度。
1-720，单位 0.5 度。

设置指针旋转中心。

变量最小值对应的旋转角度。
0-719，单位 0.5 度。

图 4-2-13 旋转图标属性设置

需要注意的是：变量值不是角度值，指针显示角度与变量值等比均分对应。起始/终止值设定的变量范围、与起始/终止角度角度设定的旋转角度可以不一一对应，串口屏会自动将变量或者转动角度均分后对应。

按照上图属性设置，通过向 0040 单元写入不同的值，就可以控制指针的旋转显示位置。旋转始终约定为“顺时针”转动，即 终止旋转角度 必须大于 起始旋转角度(如果终止旋转角度小于起始旋转角度，系统处理时会自动加上 360°)。

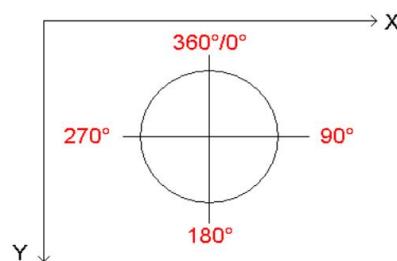


图 4-2-14 旋转方向

4.2.7 位变量图标

该控件是把一个数据变量（双字节/共 16 个位）的每个位（bit）的 0/1 状态对应 8 种不同显示方案（显示模式）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、报警指示灯的显示、风机状态显示（运转为动画、停止为静止图标）。**图 4-2-15 中，有 5 个指示灯状的图标，就是使用位变量图标实现的。该例中使用了变量图标中 16 个位的低 5 位，位值为 0 显示的是蓝色方块图标，位值为 1 显示的是带有红色指示灯的图标，5 个图标自上向下依次排列。

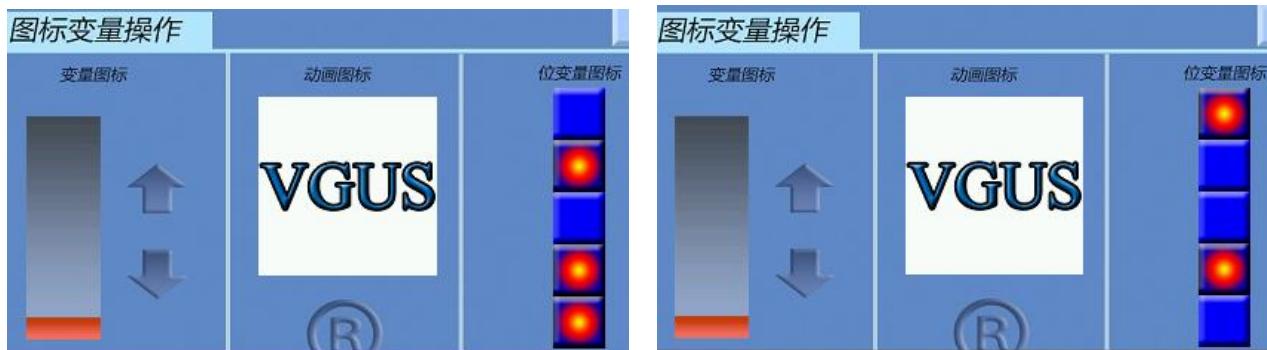


图 4-2-15 位变量图标典型应用举例

表 4-2-1 八种显示方案（显示模式）

显示模式	位值	
	0	1
0x00	ICON0S	ICON1S
0x01	ICON0S	不显示
0x02	ICON0S	ICON1S-ICON1E 动画
0x03	不显示	ICON1S
0x04	不显示	ICON1S-ICON1E 动画
0x05	ICON0S-ICON0E 动画	ICON1S
0x06	ICON0S-ICON0E 动画	不显示
0x07	ICON0S-ICON0E 动画	ICON1S-ICON1E 动画

表 4-2-2 位变量图标排列模式（16 个图标的排列方向）

排列模式	说 明
0x00	水平方向排列，自左向右、高位在左，不保留未启用位的图标位置；
0x01	垂直方向排列，自上向下、高位在上，不保留未启用位的图标位置；
0x02	水平方向排列，自左向右、高位在左，保留未启用位的图标位置；
0x03	垂直方向排列，自上向下、高位在上，保留未启用位的图标位置。

字地址, 范围 0000~FFFF, 用于存储变量值, 长度为一个单元 (字地址一个单元包含 2 个字节)。使用 0x82 指令刷新显示内容时需要使用该地址。

图标排列模式, 自上向下, 不保留未启用的图标位置。排列间隔定义相邻图标的心距, 单位像素。

图 4-2-16 位变量图标属性设置

图 4-2-15 中, 在每个图标位置还定义了一个增量调节按钮, 点击后, 控制位变量图标切换显示。控件设计如图 4-2-17, 在位变量图标的每个图标上还设计了一个增减调节按钮, 实现复选功能。增量调节按钮 0 的属性设置如图 4-2-18, 变量存储地址与位变量图标的一样, 变量模式为位调节。



图 4-2-17 位变量图标控件设计举例

变量存储地址(0x) 0122
变量模式 位调节
写入位 4
调节方式 ++
调节步长 1
按键模式 按键单次调节
下限 0
上限 1
逾限处理方式 循环

图 4-2-18 增量调节按钮属性设置

4.2.8 进度条显示

该控件是指用单色、切图或图标，在框选（单色或切图）或图标宽高（图标）范围内变化前景的显示区域，显示区域通过变量设定。以前景的显示区域的变化表示进度的变化。

需要注意的是：变量值不是前景区域在所选方向上的显示长度，前景区域在所选方向上的实际显示长度与变量值等比均分对应。



图 4-2-19 进度条属性设置

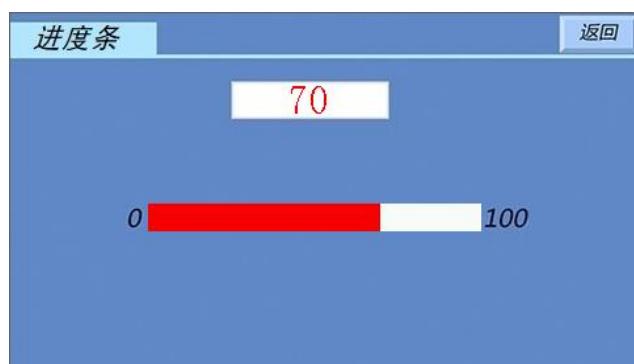


图 4-2-20 进度条举例

如上图所示，白色为背景，红色为前景，以红色显示的不同的宽度，表示当前的进度。属性设置如图 4-2-19。让进度条显示当前的进度情况，串口可以发送指令：

```
A5 5A 05 82 01 00 00 00
A5 5A 05 82 01 00 00 01
.....
A5 5A 05 82 01 00 00 63
A5 5A 05 82 01 00 00 64
```

说明: A5 5A: 帧头

05: 指令字节长度, 82 01 00 00 01 共 5 字节

82: 写变量存储器指令

01 00: 设置的变量地址, 如上图变量属性所示

00 00: 设置的进度值, 即从 00 00 变化到 00 64



电话: 027-87617912 / 87617960 / 87596062

网址: www.viewtech.cn

中显

4.2.9 圆形进度条显示

该控件使用圆环的方式显示进度条。

背景用于显示进度条的最大变化范围，前景用于显示当前的进度值。可单独设置是否显示背景/前景。

进度条的大小，厚度，显示位置，进度方向，边缘圆角，填充方式均可设置。

注意：SDWb 系列的角度范围为 0-720。VTC 系列的角度范围为 0-360。

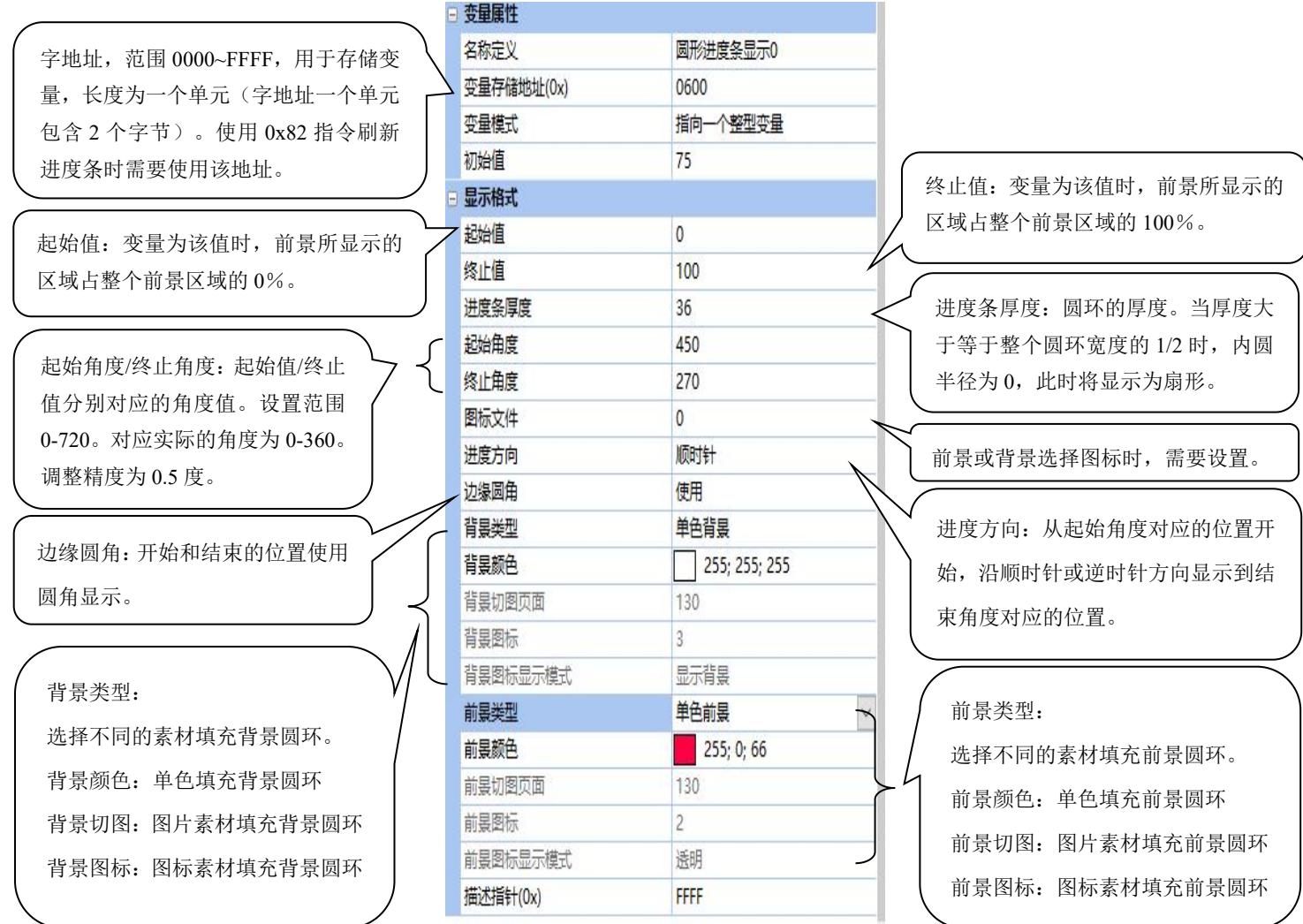


图 4-2-21 圆形进度条属性设置

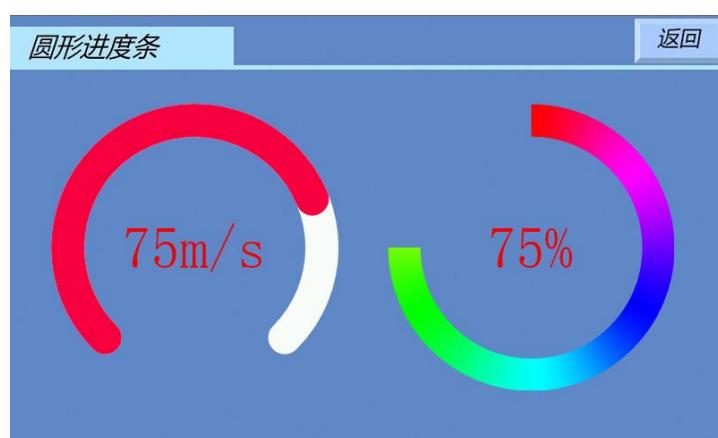


图 4-2-22 圆形进度条举例

如上图所示，两种圆形进度条的用法示例。

左侧的圆形进度条的设置如图 4-2-21 所示。起始角度为 450，结束角度为 270。（这里的角度为实际角度乘以 2，对应到实际角度，可设置的精度为 0.5）。沿顺时针方向显示。使用边缘圆角，使用白色填充背景圆弧，使用红色填充前景圆弧。

右侧圆形进度条，起始角度为 0，结束角度为 720，进度值为 100%时，将显示整个完整的圆弧。使用图标填充圆弧，进度条圆环上的颜色可以显示的更加丰富。

注：圆形进度条中间的字符使用的是抗锯齿字库以及数据变量控件显示。

更改圆形进度条的进度值，串口可以发送指令：

A5 5A 05 82 06 00 00 32

圆形进度条的进度值将更改为 50。

4.3 图形变量

4.3.1 实时曲线（趋势图）

VGUS 屏有一个 8K Word、可以存储 8 条曲线的缓冲区，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。8K word的曲线缓冲区不占用64K word的变量存储器空间，也没有分配变量存储地址。

◆ 属性设置

变量属性	
名称定义	实时曲线0
显示格式	
Y_Central	310
VD_Central	100
曲线颜色	255; 0; 0
纵轴放大倍数	161
数据源通道	0
横轴间隔	10
描述指针(0x)	6000
线条显示模式	抗锯齿模式
曲线越界处理方式	调整到边界值显示
曲线移动方向	从右到左

曲线中心位置对应的曲线数据值。

曲线中心位置，Y 坐标值。

MUL_Y，单位是 1/256，0x0000-0xFFFF。放大倍数计算方法见下文。

最多可以显示 8 条曲线，一条曲线对应有一个数据源通道，编号 0 到 7。

X 方向的像素间隔，实现水平方向的放大。

支持抗拒线条和普通线条。

支持从右到左或从左到右显示。

支持调整到边界值显示或超出部分不显示。

图4-3-1 实时曲线属性设置

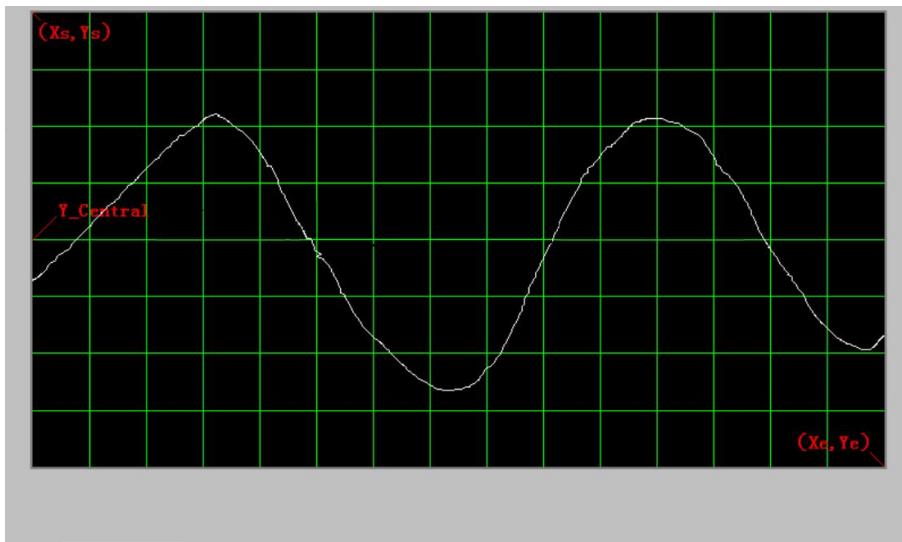


图4-3-2 实时曲线显示举例

首先在属性设置中定义曲线显示区域，如上图网格区域，左上角坐标 (Xs, Ys)、右下角坐标 (Xe, Ye)。然后设置波形中心线对应在显示屏上的Y坐标、以及中心线对应的曲线数据值，这样就确定了波形显示位置。曲线水平方向放大倍数直接通过调整横轴间隔实现。

满量程曲线纵轴放大倍数计算方法：

$$MUL_Y = (Ye - Ys) * 256 / (Vmax - Vmin)$$

Ye, Ys 为曲线窗口的 Y 坐标，

Vmax, Vmin 为曲线数据的最大，最小值。

例如一个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。

如果需要显示粗线条的曲线, 可以把做好的曲线控件复制一个出来, 把属性设置中的X坐标加1、Y_Central加1、其它属性不变, 这样就会把曲线复制并平移一个像素, 看起来就是曲线变粗了。

利用第6章介绍的描述指针的方法:

- 结合增量调节按钮, 把按钮的变量存储地址设置与纵轴放大倍数或者水平间隔存储地址一样, 可以实现通过增量调节按钮放大缩小曲线显示。无需用户程序干预。
- 结合拖动调节按钮, 把按钮的变量存储地址设置与Y_Central的存储地址一样, 可以实现通过拖动调节按钮上下移动曲线。无需用户程序干预。

◆ 曲线缓冲区写指令0x84

表4-3-1 曲线缓冲区写指令0x84帧结构

定义	指令帧头	指令长度	指令	曲线通道	曲线数据
长度 (字节)	2	1	1(0x84)	1 (CH_Mode)	2n (DATA0 (Word) +...+DATA _n)

表4-3-2 曲线通道CH_Mode定义

位	CH_Mode. 0	CH_Mode. 1	CH_Mode. 2	CH_Mode. 3	CH_Mode. 4	CH_Mode. 5	CH_Mode. 6	CH_Mode. 7
通道	通道 0	通道 1	通道 2	通道 3	通道 4	通道 5	通道 6	通道 7

- ◆ CH_Mode的每个位 (bit) 对应 1 个通道;
- ◆ CH_Mode的相应位值为1表示该条曲线存在, 为0表示不存在, 允许同时存在多条曲线;
- ◆ 曲线数据按照低通道数据在前排列, 例如CH_Mode=0x83 (10000011B), 表示曲线数据格式为: (通道0+通道1+通道7) +...+ (通道 0+通道1+通道 7)。

用户单片机通过0x84指令、按照通道号把曲线数据发送给串口屏。当串口屏收到0x84指令后, 接收到的曲线数据总是靠曲线窗口右侧显示、之前的曲线会向左移动、超出窗口长度部分的曲线会移出。

4.3.2 基本图形（绘图）

表 4-3-3 基本图形指令汇总

图形指令	功能	说明
0x0001	置点	置点 (x, y, color)
0x0002	端点连线	端点连线 (color, (x0, y0), ... (xn, yn))
0x0003	矩形	显示矩形，颜色和位置、大小可控
0x0004	矩形填充	填充指定的矩形区域，填充颜色和位置、大小可控
0x0005	画圆	显示整圆弧，颜色和位置、大小可控
0x0006	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
0x0007	ICON 图标显示	ICON 图标显示，图标库可以选择
0x0008 ^[1]	封闭区域填充	封闭区域填充，种子点坐标、填充颜色可控
0x0009	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
0x000A	线段显示	根据变量数据连接线段，端点、颜色可控
0x000B	圆弧显示	显示圆弧，半径、颜色、起止角度可控
0x000C	字符显示	根据变量数据进行单个字符显示
0x000D ^[1]	矩形区域 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
0x000E	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
0x000F	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。
0x0010 ^[1]	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 OF 指令实现照片实时显示
0x0021	JPG/PNG 图片显示	显示数据源位于变量存储器的 JPG 或 PNG 格式图片。用于实时图片下载显示，相较于“位图显示”可有效减少数据量，支持显示更高分辨率的图片。 该指令仅 VTb 系列支持。

[1] VTc/VTb 系列不支持图形指令 0x0008\0x000D\0x0010。



图 4-3-3 基本图形属性设置

基本图形显示的属性设置非常简单，如图 4-3-3 所示。SDWb 系列支持全屏显示，VTc/VTb 系列区域范围设置必须足够大，超出范围的图形不显示。变量属性中主要设置的也就是变量存储地址，需要显示的图形数据内容需要写入该地址单元中，实际占用地址单元数量取决于图形数据内容的多少，不同基本图形指令（表 4-3-3）的图形数据内容格式不同，参见表 4-3-4、表 4-3-5。

表4-3-4 图形数据内容格式定义

地址（字）	定义	说明
-------	----	----



VP	CMD	双字节, 图形指令
VP+1	Data_Pack_Num_Max	双字节, 最大数据包数目, 连线指令 (0x0002) 定义为连线线条数目, 即顶点数-1
VP+2	DATA_Pack	双字节, 数据包, 参见表4-3-5

基本图形指令举例: a5 5a 15 82 20 00 00 02 00 02 f8 00 02 3B 00 A4 02 EE 01 18 02 C2 01 B9

a5 5a : 指令帧头

15 : 指令长度, 21 个字节, 单位字节

82 : 写变量存储器指令

20 00 : 变量地址

00 02 : 端点连线的图形指令,

00 02 : 连线线条数目,

f8 00 : 连线的颜色 红色, 双字节

02 3B 00 A4 : 连线第一个点的坐标

02 EE 01 18 : 连线第二个点的坐标

02 C2 01 B9 : 连线第三个点的坐标



表 4-3-5 图形指令数据包格式定义

图形指令	绘图功能	绘图数据包格式说明 (偏移量和长度单位均为字 (word))			
		偏移量	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	2	Pic_ID	剪切图片区域所在页面ID, 高字节为判断条件
		0x02	2	(x, y) s	剪切图片区域左上角
		0x04	2	(x, y) e	剪切图片区域右下角
		0x06	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
0x008	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	0x00	1	Color0	把(X0, Y0s) (X0, Y0e) 用 Color0 颜色连线, X0 高字节为判断条件

	显示 (垂直 线条)	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
		0x01	2	(X, Y) 0	圆心 (X, Y) 坐标, X坐标高字节为判断条件
		0x03	1	RADO	半径
		0x04	1	DEG_SO	起始角度, 单位 0.5° , 0-720
		0x05	1	DEG_EO	终止角度, 单位 0.5° , 0-720
0x000C	字符 显示	0x00	1	Color0	字符显示颜色
		0x01	2	(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件
		0x03H	0.5	Lib_ID	字库位置
		0x03L	0.5	En_Mode	字符编码模式: 0=8bit 1=GB2312 2=GBK 3=BIG5 4=SJIS 5=UNICODE
		0x04H	0.5	X_Dots	字符 X 方向点阵数
		0x04L	0.5	Y_Dots	字符 Y 方向点阵数
		0x05	1	Text0	字符数据, 8bit 编码, 对仅高字节有效 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	矩形区域左上角坐标, x坐标高字节为判断条件
		0x02	2	(x, y) e	矩形区域左下角坐标
		0x04	1	Color	矩形区域XOR的颜色, 0xFFFF将进行反色操作
0x000E	双色位 图显示	0x00	2	(x, y) s	位图显示矩形区域左上角坐标, x坐标高字节为判断条件
		0x02	1	X_Dots	位图X方向点阵数
		0x03	1	Y_Dots	位图Y方向点阵数
		0x04	1	Color1	“1” bit对应的显示颜色
		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, 即下一行的数据总是从一个新数据字 (word) 开始
0x000F	位图显 示	0x00	2	(x, y) s	位图显示矩形区域左上角坐标, 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	N	Data_Pack	显示数据, 每个像素点一个字 (MSB, 565RGB 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 高字节为判断条件
		0x02	2	(x, y) s	待放大矩形域左上角坐标
		0x04	2	(x, y) e	待放大矩形域右上角坐标
0x0021	JPG/PNG	0x00	2	(x, y)	JPG/PNG 图片显示左上角坐标, X 高字节为判断条件

图片显示	0x02	1	mark	功能标记。写入以下特定值配置地址模式并开启显示。 0x5AA1 表示图片数据字地址寻址。 0x5AA2 表示图片数据双字地址寻址。 图片数据校验或解析异常时，该地址写入异常值。
	0x03	1	VP	图片数据变量存储地址。
	0x04	1	len	图片数据长度。字地址寻址时表示字长度。双字地址寻址时表示双字长度。
	0x05	1	checksum	图片数据校验和低16位。 JPG/PNG图片显示更多功能说明请查看6.14章节。

判断条件: 0xFF 绘图操作结束, 0xFE 本次操作跳过(忽略)。

基本图形显示举例 (以 0x0006 图片区域剪切、粘贴指令为例)

Step1 在界面上定义一个基本图形显示控件, 变量存储地址设置为 0x1000;



图 4-3-4 基本图形显示举例

Step2 串口向 0x1000 地址单元写入 0x0006 指令相关内容 (把第三幅页面的(100,100)、(512,256)区域剪切粘贴到当前界面的(0,0)位置)。



图 4-3-5 基本图形指令举例

当显示切换到基本图形显示控件所在的页面时, VGUS 屏将执行剪切、粘贴。

坐标 (340, 237) 与坐标 (585, 237) 端点连线:

A5 5A 11 82 10 00 00 02 00 01 00 00 01 54 00 ED 02 49 00 ED

坐标 (340, 237) 置点:

A5 5A 0D 82 10 00 00 01 00 01 01 54 00 ED 00 00

左上角坐标 (340, 237) 右下角坐标 (585, 340) 矩形:

A5 5A 11 82 10 00 00 03 00 01 01 54 00 ED 02 49 01 54 00 00

圆心坐标 (470, 325) , 半径为 50 整圆:

A5 5A 0F 82 10 00 00 05 00 01 01 D6 01 45 00 32 00 00

1号界面左上角坐标 (265, 64) 右下角坐标 (785, 256) 剪切粘贴到当前界面坐标 (265, 64) :

A5 5A 15 82 10 00 00 06 00 01 00 01 01 09 00 40 03 11 01 00 01 09 00 40

端点坐标 (80, 80) (128, 128) 线段显示:

A5 5A 11 82 10 00 00 0A 00 01 00 00 00 50 00 50 00 80 00 80

说明: 上面所有指令中 **00 01** 指最大数据包数目, 即执行画图指令的次数, 如画一条线段、一个矩形框、一个圆设置为

00 01, 画两条线段、两个矩形框、两个圆设置为 00 02。

4.3.3 列表显示（表格）

把按照二维数组定义的数据用表格分栏显示出来。

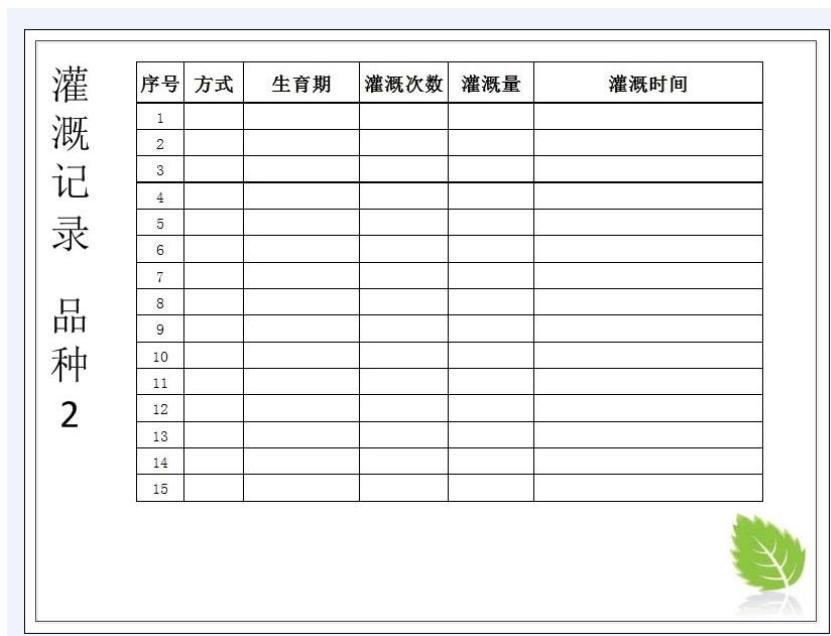
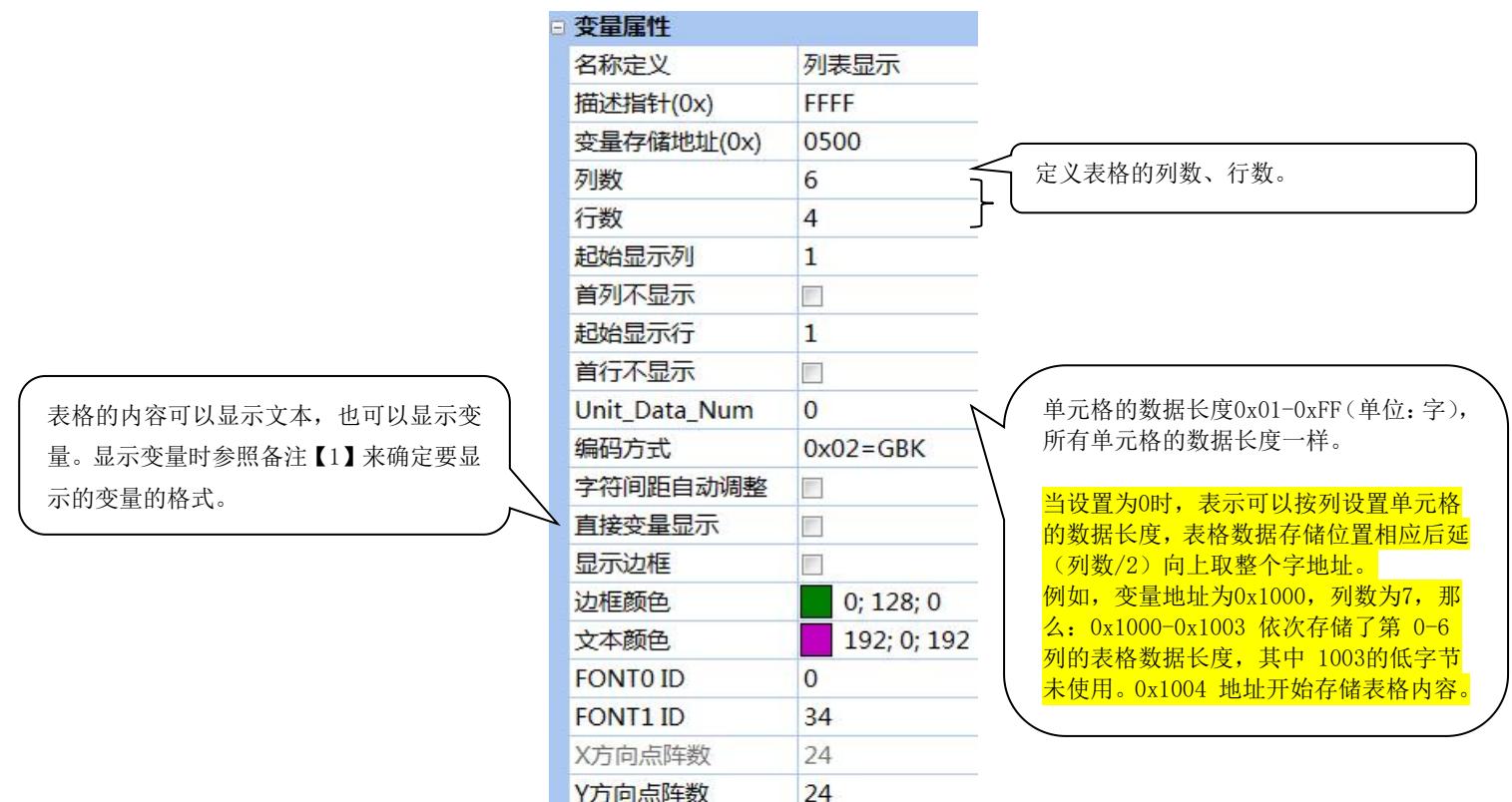


图 4-3-6 列表显示举例



表格的内容可以显示文本，也可以显示变量。显示变量时参照备注【1】来确定要显示的变量的格式。

变量属性

名称定义	列表显示
描述指针(0x)	FFFF
变量存储地址(0x)	0500
列数	6
行数	4
起始显示列	1
首列不显示	<input type="checkbox"/>
起始显示行	1
首行不显示	<input type="checkbox"/>
Unit_Data_Num	0
编码方式	0x02=GBK
字符间距自动调整	<input type="checkbox"/>
直接变量显示	<input checked="" type="checkbox"/>
显示边框	<input type="checkbox"/>
边框颜色	0; 128; 0
文本颜色	192; 0; 192
FONT0 ID	0
FONT1 ID	34
X方向点阵数	24
Y方向点阵数	24

单元格的数据长度0x01~0xFF(单位:字),所有单元格的数据长度一样。

当设置为0时,表示可以按列设置单元格的数据长度,表格数据存储位置相应后延(列数/2)向上取整个字地址。例如,变量地址为0x1000,列数为7,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。

图 4-3-7 列表显示属性设置

备注【1】

当不勾选“直接变量显示”，每个单元格数据内容的前两个字节定义了表格数据格式，说明如下：

第一个字高字节选择数据类型Mode

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, 12:34:56BCD 码串
 0x11=时间格式 2, 12-34-56BCD 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 码串的字节长度

Mode=其它无定义

第二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, 通过描述指针修改起始显示列、起始显示行可以很方便的实现表格的定位和拖动。

4.3.4 二维码

该控件实现将一段文字转换为二维码显示在串口屏上。把需要显示的二维码对应的文字、通过 0x82 指令写入变量存储地址单元中后, 串口屏会根据写入的文字内容生成二维码、并按照指定位置显示出来。

当文字内容小于等于 154 字节时, 二维码大小为 45x45 单元; 当文字内容大于 154 字节时, 二维码大小为 73x73 单元。每个单元所占实际物理像素点数量由图 4-3-8 中的 Unit_Pixels 定义。

首先在显示页面上添加一个二维码控件, 设置属性中的变量存储地址, 如图 4-3-8。然后把需要显示的二维码对应的文字、通过 0x82 指令写入变量存储地址单元中就可以了。VGUS 屏会根据写入的文字内容生成二维码、并按照指定位置显示出来。

需要注意的是: 使用 0x82 指令发送文本时, 需要在文本结束位置加上 0xFFFF 作为结束符。

字地址, 范围 0000~FFFF, 用于存储变量值, 长度由文本结束标志 0xff 或者 0x00 指定 (字地址一个单元包含 2 个字节), 允许文本最长为 458 字节。使用 0x82 指令刷新显示内容时需要使用该地址。

每个二维码单元像素所占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。

变量属性	
名称定义	二维码
变量存储地址(0x)	00E7
字长度	229
显示格式	
Unit_Pixels	4
边框显示	不显示
描述指针(0x)	FFFF

1 个汉字占用长度为 1, 1 个 ASCII 码字符占用长度为 0.5, 范围为 1-229 字。该长度仅用于 VGUS 开发工具自动分配变量存储地址和检查单元地址冲突。实际长度以文本结束标志为准, 遇到文本结束标志 0xFFFF 后不再显示。

设置是否在四周各显示一个单元的边框。

图 4-3-8 二维码属性设置

例如: 显示网站 www.viewtech.cn 生成的二维码, 串口发送指令:

A5 5A 14 82 00 00 77 77 77 2E 76 69 65 77 74 65 63 68 2E 63 6E FF FF

A5 5A: 帧头

14: 指令字节长度, 共 20 字节 (不含帧头)

82: 写变量存储器指令

00 00: 变量地址, 在控件属性中设置

77 77 77 2E 76 69 65 77 74 65 63 68 2E 63 6E: 数据内容 “www.viewtech.cn” 的 ASCII 码。

FF FF: 文本结束符



需要清除上述二维码显示，串口需要发送指令：

A5 5A 05 82 00 00 FF FF

4.3.5 自适应二维码

该控件实现将一段文字转换为二维码显示在串口屏上。把需要显示的二维码对应的文字、通过 0x82 指令写入变量存储地址单元中后，串口屏会根据写入的文字内容生成二维码、并在“框选区域”显示出来。

相较于二维码控件有以下特点：

1. 自适应二维码单元大小根据数据量自动缩放。少量数据时，单元显示更大，利于扫码识别。
2. 能够支持的文本数据量更大。
3. 支持居中对齐缩放，在“框选区域”内缩放不影响整体控件布局。
4. 支持纠错等级和颜色设置。

首先在显示页面上添加一个自适应二维码控件，设置属性中的变量存储地址，如图 4-3-8。然后把需要显示的二维码对应的文字、通过 0x82 指令写入变量存储地址单元中就可以了。VGUS 屏会根据写入的文字内容生成二维码、并按照指定位置显示出来。

需要注意的是：使用 0x82 指令发送文本时，需要在文本结束位置加上 0xFFFF 作为结束符。



图 4-3-9 自适应二维码属性设置

例如：显示网站 www.viewtech.cn 生成的二维码，串口发送指令：

A5 5A 14 82 00 00 77 77 77 2E 76 69 65 77 74 65 63 68 2E 63 6E FF FF

A5 5A：帧头

14：指令字节长度，共 20 字节（不含帧头）

82：写变量存储器指令

00 00：变量地址，在控件属性中设置

77 77 77 2E 76 69 65 77 74 65 63 68 2E 63 6E：数据内容“www.viewtech.cn”的 ASCII 码。

FF FF：文本结束符

需要清除上述二维码显示，串口需要发送指令：

A5 5A 05 82 00 00 FF FF

4.4 实时时钟 RTC 的显示



图 4-4-1 时钟显示举例

4.4.1 文本时钟

该控件用于实现图 4-4-1 中的文本时钟显示。时钟源可以选择串口屏自带 RTC 电路，也可以选择用户单片机提供的时间信息。对于没有 RTC 时钟功能的 VGUS 串口屏，必须把时钟源选择“用户单片机”模式（2018 年 11 月及以后固件版本支持）。



图 4-4-2 文本时钟属性设置

时间显示格式支持“年、月、日、时、分、秒、星期”七个时间单元，分别用表 5-4-1 的编码代替。可以在时间格式中任意组合这七个时间单元顺序、以及时间单元间的分隔符。

表 5-4-1 时间格式编码定义

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

如果需要使用中文分隔符，例如显示“2015年06月01日”，需要设计三个文本时钟控件，分别用于显示年、月、日，并且把中文的年、月、日分隔符设计到背景图片上。

4.4.2 表盘时钟

该控件用于实现图 4-4-1 中的表盘时钟显示。时钟源可以选择串口屏自带 RTC 电路，也可以选择用户单片机提供的时间信息。对于没有 RTC 时钟功能的 VGUS 串口屏，必须把时钟源选择“用户单片机”模式（2018年11月及以后固件版本支持）。

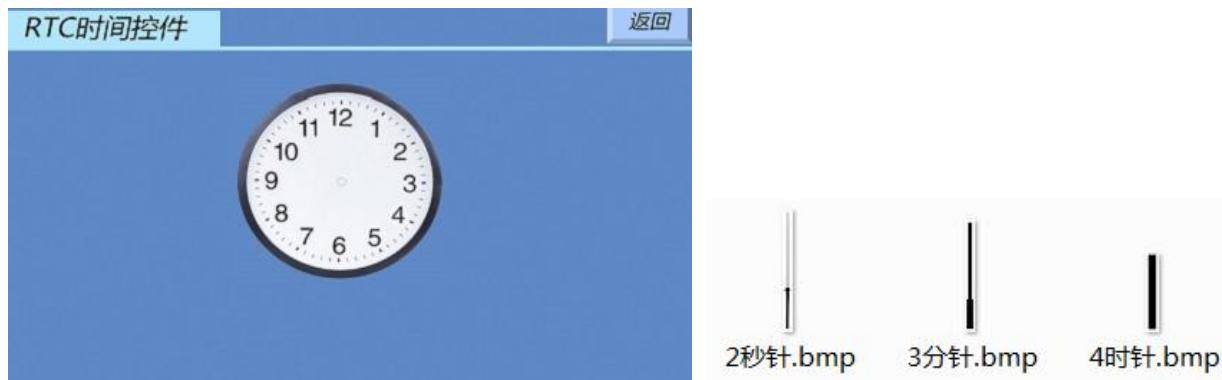


图 4-4-3 表盘时钟素材

在使用表盘时钟控件前，需要先准备一张表盘背景图片和三张图标（时针、分针、秒针），如上图所示。



图 4-4-4 表盘时钟属性设置

4.4.3 变量时钟

变量时钟只能使用用户主板提供的时钟信息，通过串口把时间信息实时发送给串口屏，然后使用变量时钟控件来显示。

变量时钟只能显示文本形式的时钟。变量时钟分配了一段变量存储单元，单元长度通过 BCD 码字节数设定，单元功能用于存储单片机发送过来的“年、月、日、时、分、秒、星期”等时间信息，格式是 BCD 码格式。用户单片机将“年、月、日、时、分、秒、星期”等时间信息以 BCD 码格式、通过 0x82

指令写入变量存储单元。串口屏依次取出变量单元里面的 BCD 时间并按照设置好的显示位置、颜色、字库、点阵大小显示出来，字节与字节之间的分隔符客户可以设置（例如习惯年月日用中划线、时分秒用冒号）。显示一个字节的 BCD 码时间、取一个分隔符显示。

字地址，范围 0000~FFFF，用于存储 BCD 码格式的时钟，长度由 BCD 码字节数设定（字地址一个单元包含 2 个字节）。使用 0x82 指令刷新显示内容时需要使用该地址。用户单片机将“年、月、日、时、分、秒、星期”等时间信息以 BCD 码格式、通过 0x82 指令写入变量存储单元。

变量属性	
名称定义	时钟变量0
变量存储地址(0x)	0000
BCD 码字节数	
BCD 码字节数	1
显示格式	
分隔符	
16 进制分隔符	
文本颜色	0; 0; 0
字库位置	0
X 方向点阵数	16
Y 方向点阵数	32
描述指针(0x)	FFFF

设置分隔符，每显示一个字节的 BCD 码数据，依次取一个分隔符。

以十六进制形式录入分隔符。

图 4-4-5 变量时钟属性设置

图 4-4-5 中的分隔符和 16 进制分隔符都是用来录入时间单元间的分隔符号，时间显示中的常用分隔符号只有中划线、冒号和空格，中划线用于分隔年月日、冒号用来分隔时分秒、空格用于分隔年月日和时分秒。ASCII 码中十六进制数 2D 是中划线、20 是空格、3A 是冒号。图 4-4-5 中时钟变量设置有 6 字节的 BCD 码时间数据、5 个分隔符号（2 个中划线+1 个空格+2 个冒号），假设变量存储单元里的 6 字节 BCD 数据依次是 15、09、01、09、20、25，那么显示出来的时间样式就是“15-09-01 09:20:25”。

分隔符还支持 2 个特殊编码 00 和 0D，只能通过 16 进制分隔符位置录入。00 不插入分隔符，实现 2 个 BCD 连到一起显示，常用与 4 位数的年份显示。0D 为换行显示，实现纵向排列时间显示，即 X=Xs, Y=Y+Font_X*2。例如设置变量时钟属性如图 4-4-6 所示，BCD 字节长度为 7 字节、6 个分隔符（00+2 个中划线+1 个空格+2 个冒号），假设变量存储单元里的 7 字节 BCD 是 20、15、09、01、09、20、25，那么显示出来的时间样式就是“2015-09-01 09:20:25”。

Byte_Num	7
编码字符串	--::
16 进制录入	002D2D203A3A

图 4-4-6 四位数的年份显示设置

例如：显示时间变量“08: 30”，串口发送指令：

A5 5A 05 82 6F 00 08 30

说明：A5 5A：帧头

05：指令字节长度，82 6F 00 08 30 共 5 字节（不含帧头）

82：写变量存储器指令

6F 00：变量存储地址，在控件属性中设置

08 30：数据内容，08 30 的 BCD 码



4.5 视频控件

该控件是在显示屏某一位置播放某一指定 AVI 视频。变量只有 2 个取值：一个为开始值，一个为停止值。当通过 0x82 指令向变量单元写入开始值时，视频开始播放；当写入停止值时，视频停止播放。



图 4-5-1 视频控件属性设置

按照图 4-5-1 设置：

当向 4000 单元写入 1 后，将播放“0.AVI”文件，播放完后切换到 0 号页面。

发送指令：A5 5A 05 82 40 00 00 01

A5 5A：帧头

05：指令字节长度，82 40 00 00 01 共 5 字节

82：写变量存储器指令

40 00：设置的变量地址，如上图变量属性所示

00 01：设置的开始值。

当向 4000 单元写入 0 后，视频停止播放。

当向 4000 单元写入其它值时无效，不执行任何操作。

小技巧：使用视频控件实现开机自动播放视频“0.AVI”

如果把图 4-5-1 中的变量初始值设置为 1，就可以实现开机自动播放视频“0.AVI”。

关于视频播放的更详细信息请参考文档《VGUS 串口屏视频播放功能详解》。

4.6 摄像头控件

摄像头控件是在显示屏指定区域显示摄像头监控画面。目前仅 SDWb 系列串口屏支持该控件功能。

变量只有 3 个取值：停止值、摄像头通道 1 的开始值、摄像头通道 2 的开始值。当通过 0x82 指令向变量单元写入开始值时，指定区域将播放摄像头监控画面；当写入停止值时，视频停止播放。



图 4-6-1 摄像头控件属性设置

按照图 4-6-1 设置：

当向 09BB 单元写入 1 后，将在屏幕指定区域显示摄像头 1 的监控画面。

发送指令：A5 5A 05 82 09 BB 00 01

A5 5A：帧头

05：指令字节长度，82 09 BB 00 01 共 5 字节

82：写变量存储器指令

09 BB：设置的变量地址，如上图变量属性所示

00 01：设置的摄像头通道 1 开始值。

当向 09BB 单元写入 0 后，监控画面停止显示。

当向 09BB 单元写入其它值时无效，不执行任何操作。

小技巧：开机后自动在指定区域播放摄像头通道 1 的监控画面

如果把上图中的变量初始值设置为 1，就可以实现开机自动播放摄像头 1 的监控画面。

5 触摸控件

5.0 触摸控件介绍

5.0.1 触摸控件汇总

触摸控件需要通过“VGUS开发工具->触控配置工具栏”添加，触控配置工具栏各图标定义如图5-0-1所示。本章将逐一介绍各触控/键控按钮的使用方法、属性设置等。

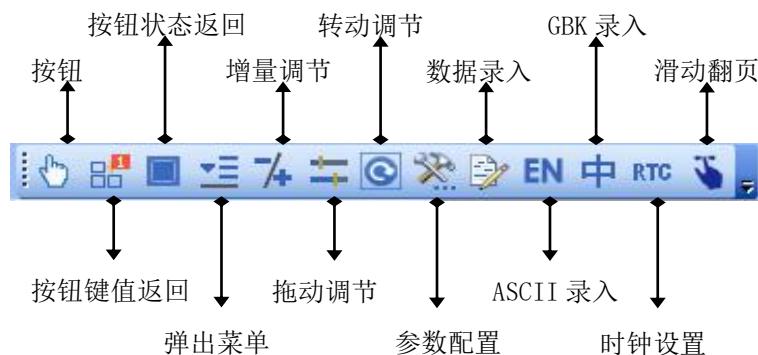


图 5-0-1 触控配置工具栏

表 5-0-1 触控/键控汇总表

	功能	说明	VTc	SDWb	SDWn
1	按钮	“按钮”不能返回键值，常用于下文提到的“弹出菜单”、“数据、ASCII 与 GBK 录入”。	√	√	√
2	按钮键值返回	“按钮键值返回”按键按下后可以通过串口以 0x83 指令形式把键值发送给用户单片机。	√	√	√
3	按钮状态返回	“按钮状态返回”可以返回按钮的三种按压状态“按下、长按、抬起”，可实现长按键。	√	√	√
4	弹出菜单	利用“按钮”控件，实现典型的弹出菜单、下拉菜单功能。	√	√	√
5	数据录入	数据录入支持录入整数、定点小数等各种数据到指定变量存储空间。	√	√	√
6	ASCII 录入	文本方式录入各种字符，录入过程支持光标移动、编辑；直接支持 ASCII 字符、GBK 中文、繁体注音输入法录入；修改字库和 0 号字库可以支持所有类似 ASCII 字符的 8bit 编码文本录入。	√	√	x
	GBK 录入		√	√	x
7	RTC 设置	RTC 设置需要完整录入公历年月日时分秒。	x	√	x
8	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复选框。	√	√	√
9	拖动调节	拖拉滑块实现变量数据录入，可设置刻度范围。	√	√	√
10	转动调节	转动旋钮实现变量数据录入，可设置刻度范围。	√	√	x
11	硬件参数配置	提供了触摸屏改写寄存器空间的方法，来间接控制硬件。	√	√	x
12	滑动翻页	通过滑动方式，实现页面切换。	x	[1]	x

[1]. SDWb 串口屏固件升级为 VTb 后不支持滑动翻页。

5.0.2 触摸控件的典型属性

触摸控件的典型属性如图5-0-2所示，控件属性分为区域范围设置、按钮属性、变量属性和控件属性四大类。区域范围设置、按钮属性和变量属性对所有控件都是类似，而控件属性对不同触摸控件可能完全不同。

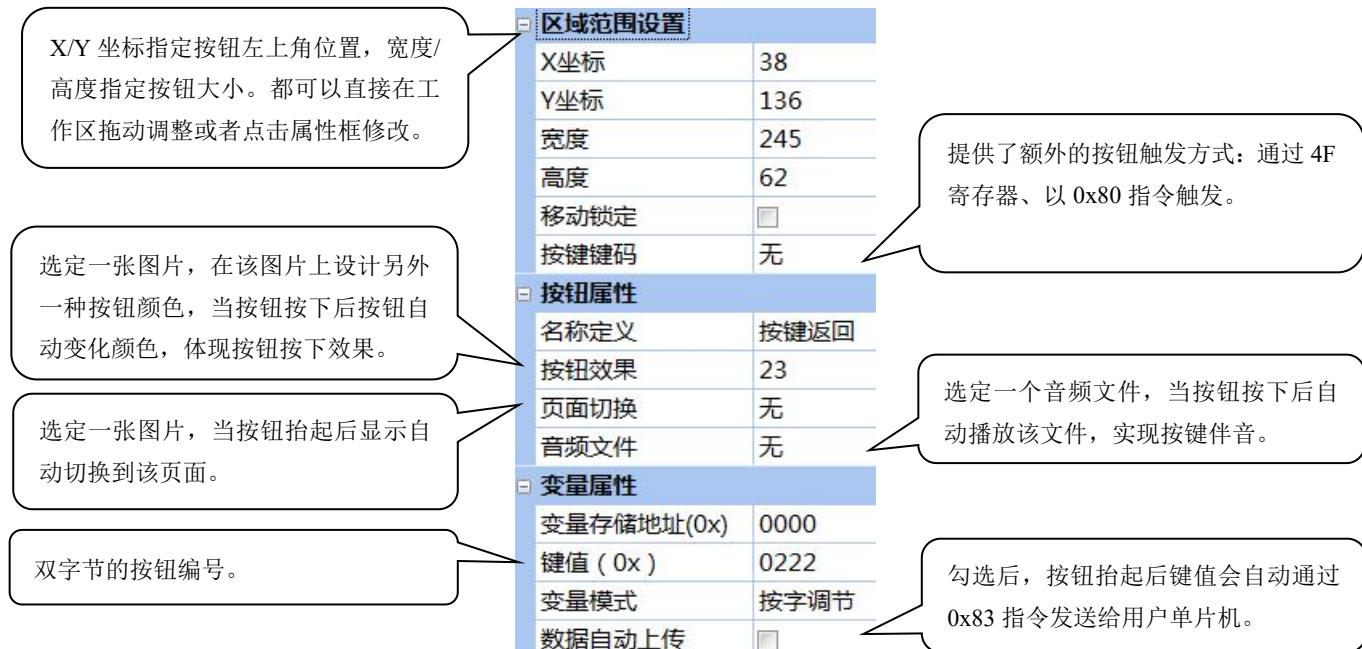


图 5-0-2 按钮区域范围设置属性

区域范围设置用于设置按钮的大小和位置。

按钮属性用于设置按钮按下反色显示效果、按钮按下后页面切换到那张图片显示、以及按钮按下播放哪个音频文件。

变量属性用于设置变量存储地址、键值、以及按钮按下后是否发送键值等。学习触摸控件的用法，务必了解清楚变量存储地址的作用和用法、以及控件属性的功能。变量存储单元用于存储键值，键值是双字节的按钮编号。在把键值保存到变量存储单元之前，先要把键值按照变量模式调整，0x83指令发送的键值也是调整之后的。勾选数据自动上传，按钮抬起后键值会自动通过0x83指令发送给用户单片机。

触摸控件的区域范围设置、按钮属性、变量属性用法基本一样，后续不再赘述。鼠标点击指定属性，在属性设置窗口下方会显示关于该属性的详细介绍。

5.0.3 按钮的基本触发方式

按钮的基本触发方式是通过点击触摸屏按钮的有效区域（由 X/Y 坐标和宽度/高度设置）。此外通过按键键码可以提供额外的触发方式，通过 4F 寄存器、以 0x80 指令方式触发（通常以弹出菜单方式用于设备状态报警显示、或者通过用户单片机主板键盘触发）。详细信息参考文档：《按钮触发方式》。

注意区分按键键码与键值。按键键码是用来触发按钮的；键值是指按钮按下后，通过串口发送给用户单片机的按钮编号。

5.1 基本按钮

VGUS 屏设置有三种基本按钮控件：按钮、按钮键值返回、按钮状态返回，三者之间各有异同。三种按钮都可以实现页面切换功能，但是“按钮”不能返回键值，“按钮键值返回”可以返回键值，“按钮状态返回”可以返回按钮的三种按压状态：按下、持续按下和抬起。

5.1.1 按钮

按钮可以实现基本的页面切换，但不能通过串口返回键值，因为按钮没有分配变量存储地址。按钮一般用于弹出菜单和键盘录入、以及不需要返回按钮键值的场合。

第一步：准备 2 张按钮的图片素材，坐标位置重叠。一张是正常显示效果，一张是按下显示效果。

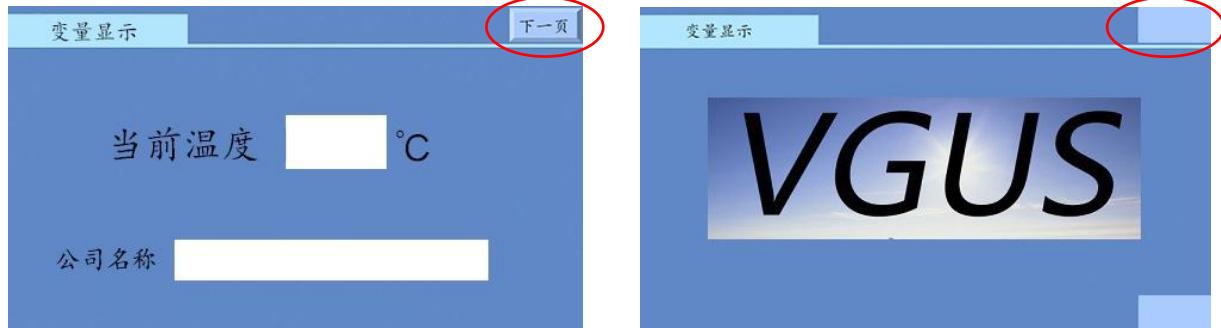


图 5-1-1 按钮效果图片素材

第二步：添加按钮并设置属性

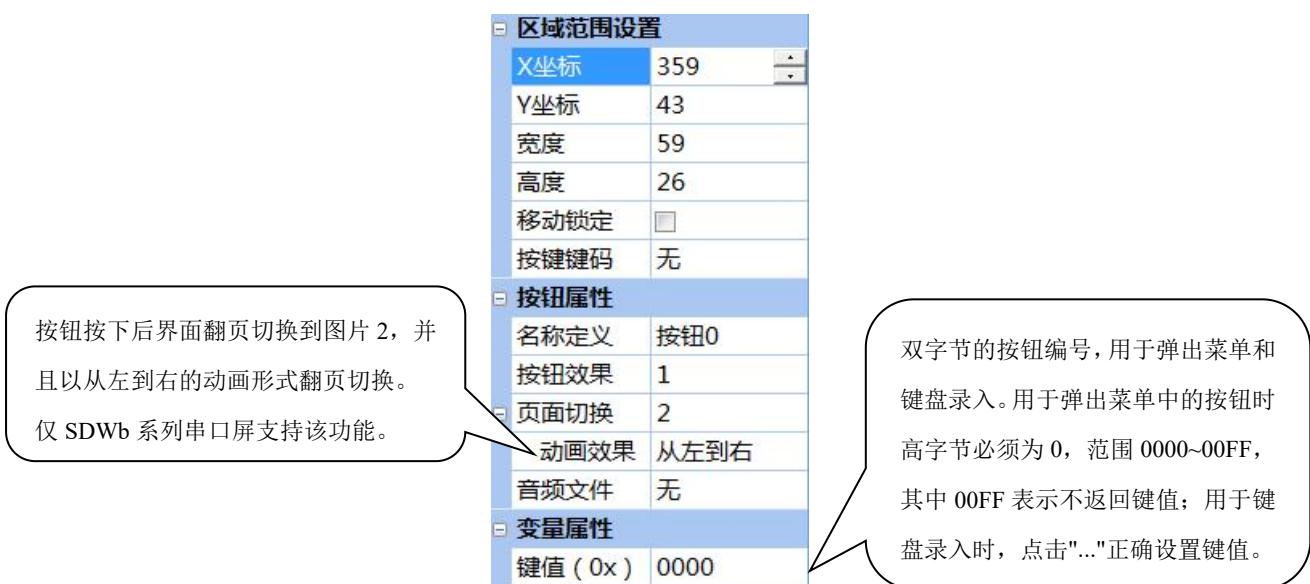


图 5-1-2 按钮属性设置

区域范围设置、按钮属性的用法参见图 5-0-2。

变量属性只有一栏键值，没有分配变量存储地址，因此不能通过 0x83 指令返回键值；在“弹出菜单”控件中，弹出窗口中设计的“按钮”控件，会利用弹出菜单控件的变量存储地址来存储“按钮”控件键值，同时也能实现通过 0x83 指令返回键值。

5.1.2 按钮键值返回

相对“按钮”控件而言，“按钮键值返回”控件多出了键值自动上传功能。当按键抬起后，可以通过串口以0x83指令把键值发送给用户单片机。按钮键值返回属性设置如图5-1-3所示。

字地址，范围0000~FFFF，用于存储键值，长度为一个单元（字地址一个单元包含2个字节）。按钮抬起后相应键值会保存到该单元中。通常把所有按钮都分配相同的变量存储地址，通过键值来区分按钮。使用0x83指令发送键值时会使用该地址。

区域范围设置	
按钮属性	
变量属性	
变量存储地址(0x)	0000
键值(0x)	0222
变量模式	按字调节
数据自动上传	<input type="checkbox"/>

图5-1-3 按钮键值返回属性设置

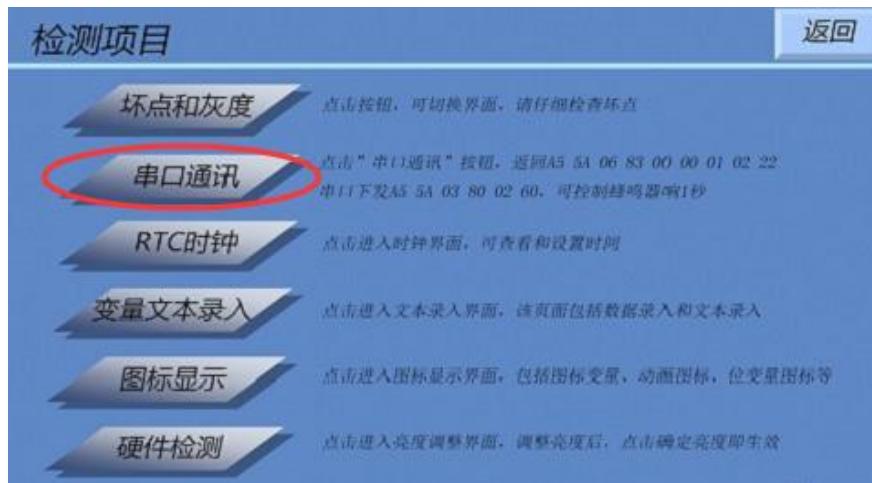


图5-1-4 按钮键值返回举例

例如，在图5-1-4“串口通讯”位置添加有按钮键值返回按钮，属性设置如图5-1-3，当点击“串口通讯”按钮后，串口返回指令：**A5 5A 06 83 00 00 01 02 22**

说明：**A5 5A**：帧头

06：指令字节长度，**83** 00 00 01 02 22 共6字节（不含帧头）

83：读变量存储器指令

00 00：变量存储器地址

01：数据字长度，**02 22** 共1字长

02 22：数据内容，设置的键值

5.1.3 按钮状态返回

持续按下触摸按钮后，如图5-1-5所示，分为三个状态：第一次按压状态、持续按压状态和抬起状态。在有些应用场合用户单片机可能需要识别这些状态，可以通过按钮状态返回控件实现。

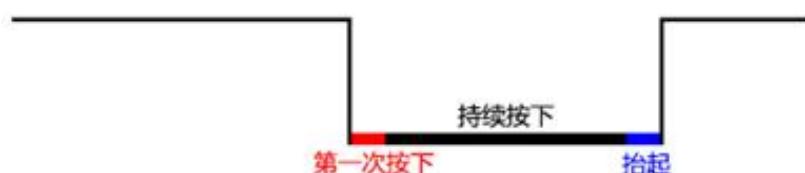


图5-1-5 按钮持续按压的三种状态

相对“按钮键值返回”控件而言，“按钮状态返回”控件可以返回按钮按下的三种状态，并且返回值用户可以灵活自定义。按钮状态返回按钮属性如图 5-1-6 所示，分别定义了三种状态下的按压模式、S 变量存储地址、S 变量数据长度、S 变量初始值、T 变量存储地址、T 变量初始值。



图 5-1-6 按钮状态返回属性设置

表 5-1-1 按压模式

按压模式	说明
0x00	不做任何处理
0x01	读取 S 地址处的指定长度数据写到 T 地址里 (数据长度必须为偶数字节)
0x02	把 S 变量数据通过串口发送给用户单片机
0x03	读取 S 地址处的指定长度数据写到 T 地址对应的寄存器里 (T 地址高字节必须为 0x00)

当把按压模式都设置为 0x02 时，三种状态下串口屏都会将各自的 S 变量数据通过串口发送出去；其中持续按压状态下，每一个 VGUS 刷新周期（约 100 毫秒）都会重复发送一次。向 S 变量中写入特定的初始值，用户单片机根据接收到的数据，就可以识别出持续按钮的各个状态。

需要注意的是，在把 S 变量数据通过串口发送的时候，是不加帧头的，为了统一指令格式，可以把帧头信息放到 S 变量里面。例如 S 变量地址是 0x0110、长度为 2 个字节、初始值是 0x6666，那可以把 S 变量长度设置为 9 个字节，初始值设置为“a5 5a 06 83 01 10 01 66 66”，这样用户单片机接收到的数据就符合统一指令格式。

0x01、0x03 两种按压模式使用较少。

5.2 弹出菜单

利用“按钮”控件，实现典型的弹出菜单、下拉菜单设计。点击“弹出菜单”按钮后会弹出一个新窗口，新窗口上可以设计“按钮”控件，按钮键值可以通过弹出菜单的变量存储地址返回。点击弹出菜单窗口上面任意一个按钮，弹出窗口会自动消失。下拉菜单也可以使用本控件设计。

第一步：在弹出菜单窗口上添加按钮并设置键值

首先要把弹出菜单窗口保存成一张图片。然后在该弹出菜单窗口有效区域里，添加需要的按钮（见 5.1.1 节内容）并设置好按钮键值（取值范围 0000 到 00fe，00ff 专用于取消按钮）。弹出菜单上的按钮必须设置键值。



图 5-2-1 在弹出菜单窗口上添加按钮

弹出菜单窗口上设计了 2 个按钮，一个 OK 按钮和一个取消按钮。OK 按钮的键值设置为 0001，取消按钮的键值设置为 00ff。当弹出菜单窗口弹出后，点击上面任意一个按钮，弹出窗口会自动消失。

当点击 OK 按钮时，按钮键值 0001 会写入弹出菜单变量单元里，变量单元地址见 5.2.2 弹出菜单按钮属性设置，键值也可以通过串口指令 0x83 发送给用户单片机。取消按钮键值设置为 0x00ff，点击取消按钮后不会把键值写入变量单元里，也不会通过 0x83 指令发送键值给用户单片机。

第二步：添加弹出菜单按钮并设置属性



图 5-2-2 添加弹出菜单按钮

上图为添加一个弹出菜单按钮。一般来讲，弹出菜单是用于系统报警提示用，不会通过按钮来触发的。

所以实际使用中，通常会把该图中的弹出菜单按钮隐藏起来（触摸区域做的很小、放到角落位置、并且不要在背景图片上体现按钮），设置按键键码、通过 0x80 指令写 4F 寄存器来触发。

字地址，范围 0000~FFFF，用于存储键值，占用一个单元（字地址一个单元包含 2 个字节）。弹出菜单页面上的按钮抬起后，相应按钮键值会保存到该单元中。使用 0x83 指令上传键值时会使用该地址。



图 5-2-3 弹出菜单按钮属性设置



图 5-2-4 弹出菜单执行效果

例如：设置“弹出菜单”变量存储地址为 0200，在弹出菜单页面上添加“OK”按钮的键值为 0001。

点击“弹出菜单”按钮会弹出菜单窗口，点击菜单上的“OK”按钮，会将按钮的键值 0001 返回到地址 0200 单元里，弹出菜单窗口消失，串口返回指令：

A5 5A 06 83 02 00 01 00 01

说明：A5 5A：帧头

06：指令字节长度，83 02 00 01 00 01 共 6 字节（不含帧头）

83：读变量存储器指令

02 00：变量地址，在弹出菜单控件属性中设置

01：数据字长度，00 01 共 1 字长

00 01：数据内容，设置的按钮键值 00 01

5.3 键盘录入

VGUS 屏支持数据录入、英文录入、中文全拼录入、时钟设置等。在使用录入控件之前，必须先制作满足功能需要的录入键盘。

5.3.1 制作键盘

第一步：准备素材。

键盘有效区域分为两个部分：录入过程中的文本显示窗口、和按钮区域。把所有需要用到的按钮做成图片，按照实际使用的效果排列好。一张正常显示的图片，一张按钮按下时显示的图片。

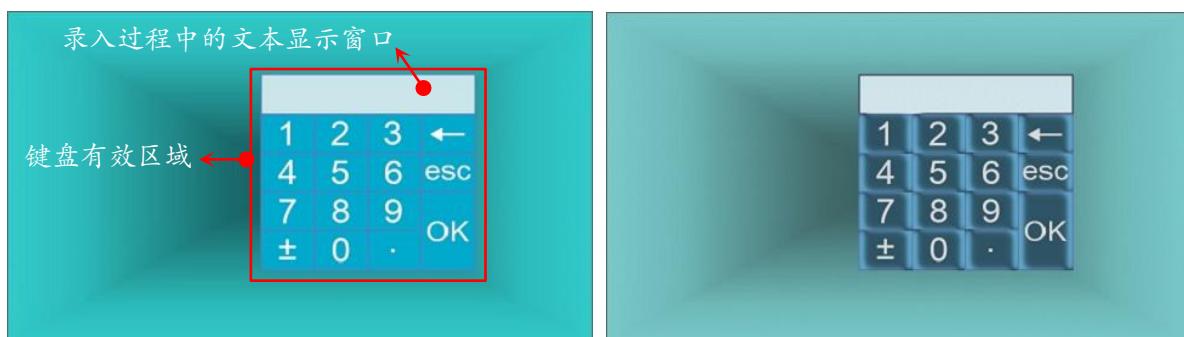


图 5-3-1 数字小键盘素材

第二步：添加按钮并设置键值。

参照 5.1.1 节内容，添加上图中的 15 个按钮并设置好属性。此处按钮键值不能任意写入数字，必须首先将鼠标指向键值属性并点击，键值属性右侧会出现图标“...”，点击该图标展开键值选择窗口，根据每个按键的实际功能做相应选择，选择后系统会自动正确提供该按钮的键值。



图 5-3-2 数字键盘按钮 1 的键值设置过程

5.3.2 数据录入

点击“数据录入”按钮后，会弹出录入键盘实现数据录入的功能。在学习数据录入之前，需要先掌握数据变量显示一节的内容(4.1.1 节)。



图 5-3-3 数据录入应用举例

上面三张图片介绍了我们将要实现的数据录入功能。为了实现上述功能，我们需要在文本框位置添加两个控件：一个是数据变量显示控件，一个是数据录入控件。

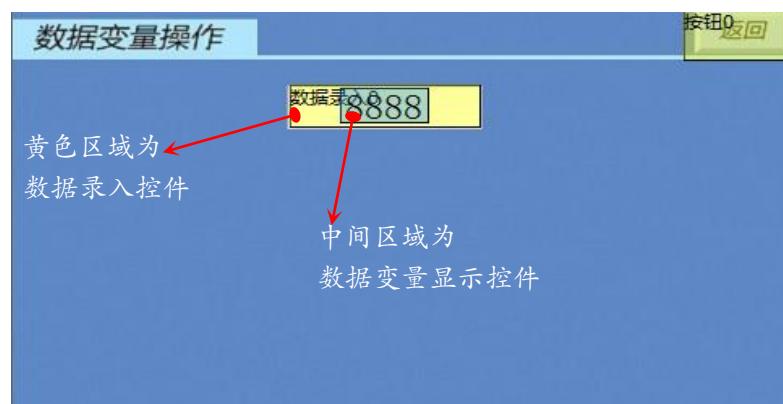


图 5-3-4 数据录入控件设计

数据变量显示控件属性设置参见 4.1.1 节，数据变量录入控件如图 5-3-5 所示。

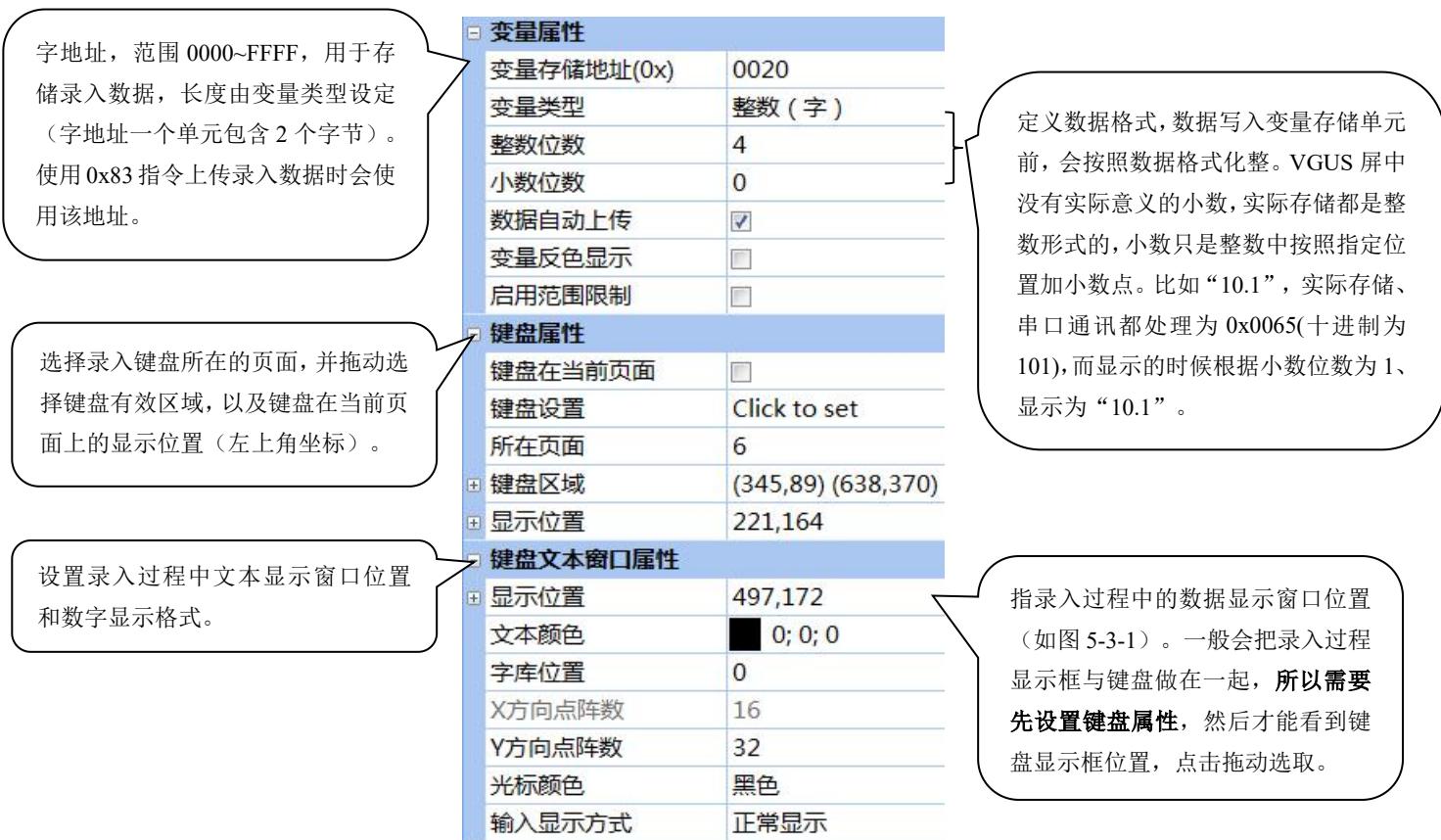


图 5-3-5 数据录入按钮属性设置

按钮属性栏里面有个变量“反色显示属性”，勾选后，将对相应的存储单元内变量反色显示。一般用于页面上有多个录入变量时，用此功能将当前录入的变量反色显示，以示区分。

如图 5-3-3 所示，如希望录入完成后，把录入的数据直接显示在文本框中，需要图 5-4-4 中的数据变量控件和数据录入控件设置相同的变量存储地址。

例如，按照图 5-3-5 设置属性，设置录入数据整数位数为 4 位，点击“数据录入”按钮输入“1234”，点击“OK”，串口返回指令：

A5 5A 06 83 00 20 01 04 D2

说明：**A5 5A：帧头**

06：指令字节长度，83 00 20 01 04 D2 共 6 字节（不含帧头）

83：读变量存储器指令

00 20：变量地址，在控件属性中设置

01：数据字长度，04 D2 共 1 字长

04 D2：数据内容，1234 的十六进制数据

5.3.3 英文ASCII录入

点击“ASCII录入”按钮后，会弹出录入键盘实现ASCII码字符录入。在学习ASCII录入之前，需要先掌握文本变量显示一节的内容(4.1.2节)。



图 5-3-6 ASCII 录入举例

字地址，范围 0000~FFFF，用于存储录入的 ASCII 码字符，长度由字节长度设定（字地址一个单元包含 2 个字节），实际占用地址单元个数=字长度+2（一个单元的输入状态和一个单元的结束符）。使用 0x83 指令上传录入文本会使用该

勾选表示要返回状态，状态信息参见备注1，不勾选表示不返回。

指录入过程中的文本显示窗口位置。一般会把录入过程显示框与键盘做在一起，所以需要先设置下面的键盘属性，然后才能看到键盘显示框位置，点击拖动选取。

变量属性	
变量存储地址(0x)	0100
字长度	40
数据自动上传	<input checked="" type="checkbox"/>
输入状态返回	<input checked="" type="checkbox"/>
录入模式	重新录入
键盘属性	
键盘在当前页面	<input type="checkbox"/>
键盘设置	Click to set
所在页面	8
键盘区域	(108,102) (723,429)
显示位置	68,141
键盘文本窗口属性	
文本显示区域	(78,153) (676,201)
文本颜色	0; 0; 0
字库位置	0
X方向点阵数	12
Y方向点阵数	24
光标颜色	黑色
输入过程显示方式	正常显示

设定录入 ASCII 码字符串的长度，范围 1-123,2 个 ASCII 码字符占用一个字。字符保存到指定地址时，自动在字符串结束位置加上 0xFFFF 作为结束符，输入结束上传到串口的数据也会加上 0xFFFF。结束符会占用一个变量存储单元，分配变量存储地址时需要注意，其它变量不能占用该单元地址。

设置录入过程中的文本显示位置和显示格式。

图 5-3-7 ASCII 录入按钮属性设置

备注1：勾选输入状态返回后，串口屏将把当前录入状态（0x00为录入中、0x5a为录入结束）和录入的有效数据长度（单位为字节）写在变量存储地址的前一个单元里。用户可以用0x83指令去读取录入状态和有效数据长度。例如上例中，变量存储地址是0x0100，因此会占用0x00ff单元，其中高字节用于存储录入状态、低字节用于存储录入的有效数据长度。在该例中因为占用了0x00ff单元，所以其它变量不能再使用。

ASCII录入实际占用的变量存储单元=字长度+2。额外占用的2个单元一个在变量存储位置前（状态返回）、一个在变量存储位置后（结束符）。在分配变量单元地址时要注意，其它变量地址不能重叠。

5.3.4 中文GBK录入

点击“GBK录入”按钮后，会弹出录入键盘实现中英文字符录入。在学习之前，需要先掌握文本变量显示（4.1.2 节）。首先我们看一款经典中文全拼录入键盘的界面设计。拼音窗口上部显示录入的拼音、下部显示检索到的汉字，文本窗口显示已经录入的汉字。拼音窗口与文本窗口中的汉字都必须使用 GBK 字库，需要在属性窗口分别设置字库、点阵大小等显示格式。

在 GBK 录入过程中，可以用 enter 健直接把拼音转为字母录入，实现 ASCII 码录入。拼音窗口和文本窗口中用到的 ASCII 码字符一律默认使用 0 号字库，点阵大小与中文 GBK 字库一致。



图 5-3-8 GBK 录入键盘举例

字地址，范围 0000~FFFF，用于存储录入中英文字符，长度由字符串长度设定（字地址一个单元包含 2 个字节），实际占用地址单元个数 = 字长度+2（一个单元的输入状态和一个单元的结束符）。使用 0x83 指令上传录入文本时会使用该地

勾选表示要返回状态，状态信息参见备注【1】，不勾选这表示不返回。

设置文本窗口的显示位置、显示格式。必须先设置键盘属性。

设置拼音窗口显示位置、显示格式。必须先设置键盘属性。

设置拼音窗口中拼音字母与检索到的汉字排列关系。

变量属性	
变量存储地址(0x)	0140
字长度	40
数据自动上传	<input checked="" type="checkbox"/>
输入状态返回	<input checked="" type="checkbox"/>
录入模式	重新录入
键盘属性	
键盘在当前页面	<input type="checkbox"/>
键盘设置	Click to set
所在页面	8
键盘区域	(109,103) (723,429)
显示位置	63,148
键盘文本窗口属性	
文本显示区域	(342,154) (669,212)
文本字库位置	66
文本颜色	0; 0; 0
Y方向点阵大小	24
键盘拼音窗口属性	
拼音显示位置	71,154
拼音字库位置	66
拼音文本颜色	255; 0; 0
Y方向点阵大小	24
拼音显示方式	上边
光标颜色	黑色
显示间距	1

设定录入字符串的长度，范围 1-123，1 个汉字占用 1 个字、2 个 ASCII 码占用 1 个字。文本保存到指定地址时，自动在字符串结束位置加上 0xFFFF 作为结束符，输入结束上传到串口的数据也会加上 0xFFFF。结束符会占用一个变量存储单元，分配变量存储地址时需要注意，其它变量不能占用该单元地址。

设置文本窗口汉字显示的字库、点阵大小和颜色，字库必须是GBK字库。ASCII字母默认用0号字库。

设置拼音窗口汉字显示的字库、点阵大小和颜色，字库必须是GBK字库。ASCII字母默认用0号字库。

图 5-3-9 GBK 录入按钮属性设置

备注1：与ASCII录入一样，勾选输入状态返回后，串口屏将把当前录入状态（0x00为录入中、0x5a为录入结束）和录入的有效数据长度（单位为字节）写在变量存储地址的前一个单元里。用户可以用0x83指令去读取录入状态和有效数据长度。例如上例中，变量存储地址是0x0180，因此会占用0x017f单元，其中高字节用于存储录入状态、低字节用于存储录入的有效数据长度。在该例中因为占用了0x017f单元，所以其它变量不能再使用。

与ASCII录入一样，GBK录入实际占用的变量存储单元=字长度+2。额外占用的2个单元一个在变量存储位置前（状态返回）、一个在变量存储位置后（结束符）。在分配变量单元地址时要注意，其它变量不能再占用。



图 5-3-10 GBK 录入举例

例如：点击“GBK 录入”按钮输入 “武汉中显”，点击 “Enter” 按钮，串口返回指令：

A5 5A 0E 83 01 40 05 CE E4 BA BA D6 D0 CF D4 FF FF

说明： **A5 5A：** 帧头

0E： 指令字节长度， **83 01 40 06 CE E4 BA BA D6 D0 CF D4 FF FF** 共 14 字节（不含帧头）

83： 读变量存储器指令

01 40： 变量地址，在控件属性中设置

05： 数据字长度， **CE E4 BA BA D6 D0 CF D4 FF FF** 共 5 字长

CE E4 BA BA D6 D0 CF D4 FF FF： 数据内容，“武汉中显”的汉字内码 **CE E4 BA BA D6 D0 CF D4**，**最后 2 字节 FF FF 是录入结束的标志位，文本录入及 ASCII 录入时有结束标志位。**

注：中显预装的 0#字库包含 4*8-64*128 点阵的所有 ASCII 字符。

不使用触摸屏，使用键盘（0x4F 寄存器保存的键码）来做 GBK 录入时，必须用 0x01-0x08 键码来选择对应的汉字。

5.3.5 时钟设置

点击时钟设置按钮，会弹出录入键盘完成时间设置，设置的时间会自动保存到 0x20-0x26 寄存器中。时钟设置按钮仅用于有 RTC 功能的串口屏，也就是针对文本时钟和表盘时钟控件，用于校准年月日、时分秒。

由于“时钟设置”控件没有分配变量存储地址，因此不能通过 0x83 指令返回设置时间；设置时间是直接写入了 0x20-0x26 寄存器中，因此是通过 0x81 指令返回设置时间。

时钟设置控件属性与数据录入基本一样，如图 5-3-11。



图 5-3-11 时钟设置按钮属性



图 5-3-12 时钟设置举例

例如：启动时钟设置按钮后，输入时间 16-10-26 16: 09: 00，串口返回指令：

A5 5A 0A 81 20 07 16 10 26 03 16 09 00

说明：A5 5A：帧头

0A：指令字节长度，81 20 07 16 10 26 03 16 09 00 共 10 字节（不含帧头）

81：读寄存器指令

20：控制 RTC 的寄存器地址

07：数据字节长度，16 10 26 03 16 09 00 共 7 字节长

16 10 26 03 16 09 00：时间变量的 BCD 码

5.4 调节控件

5.4.1 增量调节

点击增量调节按钮，可以将指定变量、按照指定步长增加或者减小。



图5-4-1 增量调节按钮属性设置

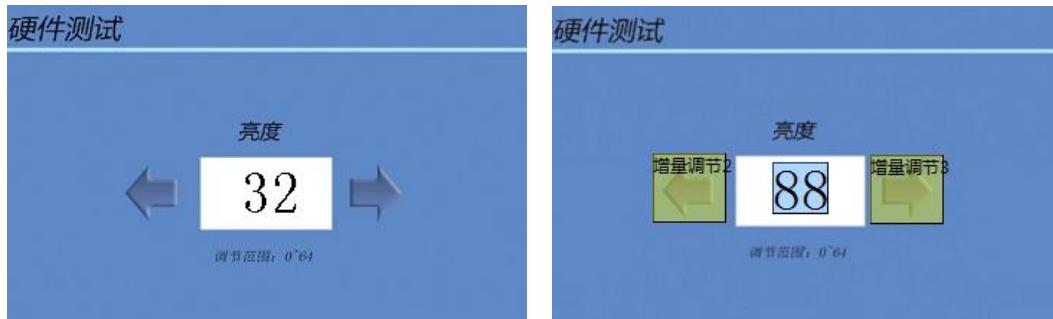


图5-4-2 增量调节举例

如图 5-4-2 左图，点击左箭头文本框数据减小，点击右箭头文本框数据增大。为实现上述增量调节功能，如上图右侧，需要设计两个增量调节按钮、一个数据变量显示。左箭头设置为--、右箭头设置为++，为了能让增量调节后的值能够显示在中间的数据变量位置，图中的 2 个调节按钮和数据变量必须定义相同的变量存储地址。

例如图 5-4-2，右箭头增量调节按钮属性设置如图 5-4-1 所示，连续点击右箭头按钮，串口返回指令：

A5 5A 06 83 6f 01 01 00 01 A5 5A 06 83 6f 01 01 00 02 A5 5A 06 83 6f 01 01 00 09

说明：A5 5A：帧头

06：指令字节长度，83 6f 01 01 00 01 共 6 字节（不含帧头）

83：读变量存储器指令

6f 01：变量地址，在控件属性中设置

01：数据字长度，00 01 共 1 字长

00 01：数据内容，从设置的下限变化到上限，即从 00 01 连续变化到 00 09

当把属性中下限和上限分别设置为 0 和 1 时，遇限处理方式设置为循环调节，配合变量图标显示，可以方便的设计出栏目复选框功能（点击一下选中，再点击取消）。

当把属性设置中的下限和上限都设置为 0，遇限处理方式设置为停止，可以用于实现连续按键识别。

按键按下 0.5 秒后，每个 VGUS 刷新周期（约 100 毫秒）都会发出串口数据。

5.4.2 拖动调节

按下并移动“拖动调节”按钮，可以将指定变量、按照设定的范围自动变化。



图5-4-3 拖动调节属性设置

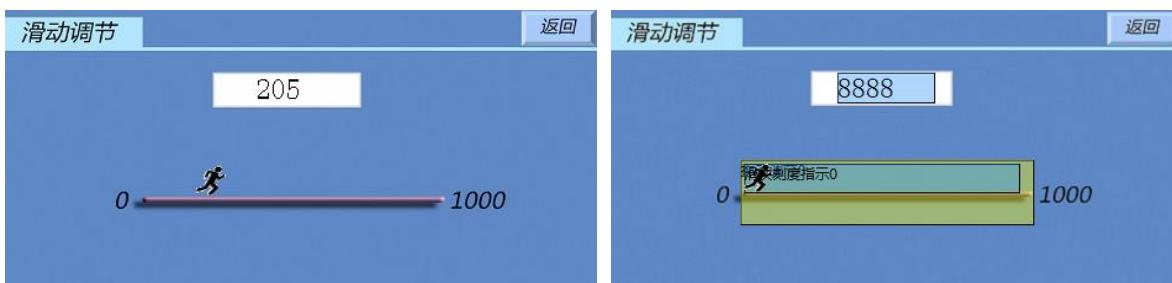


图5-4-4 拖动调节举例

如图 5-4-4 左图，点击并拖动人形滑块，人形滑块会跟随移动，同时上方文本框数值会跟随变化。为实现上述典型的拖动功能，如上图右侧，需要设计一个数据变量、一个拖动调节按钮、一个滑块刻度显示控件，并且三者必须定义相同的变量存储地址。

例如，如图 5-4-4 设计，拖动调节属性设置如图 5-4-3，拖动“拖动调节”按钮，串口返回指令：

A5 5A 06 83 01 80 01 00 00

.....

A5 5A 06 83 01 80 01 03 E8

说明：A5 5A：帧头

06：指令字节长度，83 01 80 01 00 00 共 6 字节（不含帧头）

83：读变量存储器指令

01 80：变量地址，在控件属性中设置

01：数据字长度，00 00 共 1 字长

00 01：数据内容，从设置的起始值变化到终止值，即从 00 00 连续变化到 03 E8

拖动录入的优点是直接、快捷，而且参数不会越界。拖动调节不支持按键控制（不支持外接键盘和 4F 寄存器）。为防止误操作，拖动调节须按压有效拖动区域超过 0.1 秒后拖动才启动。

5.4.3 转动调节

和“拖动调节”按钮类似，按下并移动“转动调节”按钮，可以将指定变量、按照设定的范围自动变化。
转动调节不支持按键控制（不支持外接键盘和4F寄存器）。转动调节始终假定为顺时针转动。



图5-4-5 转动调节按钮属性设置

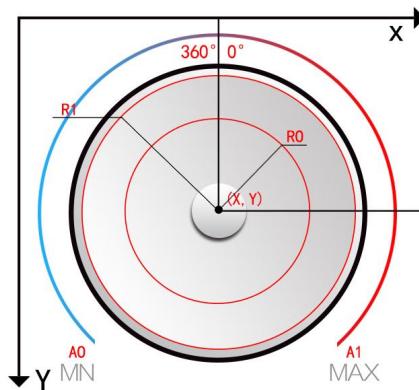


图 5-4-6 转动调节区域示意图

5.4.4 硬件参数配置

常用于把变量存储器内的数据写入寄存器里，或者把寄存器内的数据写入变量存储器里。在变量存储器区开辟了一块特殊区域 0x6F00-0x6FFF，共 256 个单元（仅仅使用低字节），这 256 个单元与寄存器区单元（0x00-0xff）一一对应。



图 5-4-7 硬件参数配置按钮属性设置

表 5-4-1 硬件参数配置按钮操作模式

模式	Data_Pack	Data_Pack说明	功能
0	无	无	把 256 字节寄存器区数据复制到到变量存储器单元 0x6F00-0x6FFF（占据低字节）。VGUS 屏上电自动把寄存器区数据复制到 0x6F00-0x6FFF 单元。
1	无	无	把变量存储器单元 0x6F00-0x6FFF（占据低字节）数据复制到寄存器区。
3	*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 是双字节变量，长度从 0x0001-0xFFFF。 举例：0110 0001 说明：通过 83 指令，将 0x0110 地址，2 字节长度的数据发送到用户串口。
	Tx_LEN	要发送的数据长度	
5	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) & 0xFFFF0/16； *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。 举例：015E 0064 019E 0084 0200 说明：指定区域左上角坐标（350, 100）右下角坐标（414, 132），保存到 0x0200 指针指向的数据存储器。
	*VP	保存转换位图数据的 缓冲区首地址	



图 5-4-10 硬件参数配置举例

在增量调节图 5-4-2 的例子中，已经有用到上图例子，通过左右键增量调节按钮我们可以修改文本框中的亮度值。如图 5-4-8 中，再在右下角添加一个硬件参数配置按钮“OK”，就可以直接用文本框中的数值去控制液晶屏的亮度，硬件参数配置按钮属性如图 5-4-7。

为实现上述功能，必须将增量调节按钮、数据变量的变量存储器地址都设置为 0x6F01，正好对应亮度设置寄存器 0x01，硬件参数配置按钮设置为模式 1。当硬件参数配置按钮按下后，变量存储器 0x6F00-0x6FFF 单元里的数据（仅低字节）会复制到寄存器区，其中 0x6F01 单元里的亮度值正好对应写入亮度寄存器 0x01 单元，屏幕亮度跟随变化。

如果工程中用到硬件参数配置按钮，则所有变量地址必须避开 0x6F00-0x6FFF 区域。

5.4.5 滑动翻页

在“滑动翻页”按钮范围内点击并滑动，可实现页面切换。

目前仅 SDWb 系列串口屏支持该控件功能。



图5-4-11 滑动翻页属性设置

如图 5-4-11 所示，相较与按钮直接切换页面的方式，滑动翻页提供了一种更加灵活的页面切换途径。

可以对滑动翻页的目标界面进行设置。

支持两种滑动方向的选择。

同时可以对触发翻页的灵敏度进行设置。

滑动一定距离并且按下一定时间后，可判定为翻页。抬起触摸后，可用通过动画方式“滑动”到目标页面。也可以选择直接切换到目标页面。

5.5 弹出窗口

为丰富弹窗中的功能, SDWb、VTC 系列串口屏在弹窗中支持部分显示控件和触摸控件, 使得弹窗中可显示更加丰富的内容, 满足更多的应用场景。

支持弹出窗口的控件有: 弹出菜单、数据录入、ASCII 录入、GBK 录入、RTC 录入控件、文字标签、数值显示、多功能按钮。弹窗中支持的显示控件和触摸控件汇总如表 5-5-1 所示。

表 5-5-1 弹窗中支持的显示控件和触摸控件汇总

	显示控件	触摸控件
SDWb 串口屏	数据变量、文本变量、滚动文本变量、变量图标、动画图标、艺术字图标、二维码控件、自适应二维码控件。	按钮控件
VTC/VB 串口屏	数据变量、文本变量、滚动文本变量、变量图标、动画图标、滑块刻度、旋转图标、艺术字图标、位变量图标、表盘时钟、文本时钟、时钟变量、实时曲线、二维码控件、进度条、圆形进度条控件、自适应二维码控件、下拉列表、滚轮、开关、文字标签、数据记录表格、数据曲线、数值显示、多功能按钮、键盘。不支持动画图片控件。	按钮控件、按钮键值返回控件、按钮状态返回控件、增量调节控件、硬件参数配置控件、拖动调节控件、转动调节控件。

在弹窗中使用上述显示控件或触摸控件时, 只需要在弹窗所在页面创建相应控件。当弹窗弹出时, 这些控件会一并显示出来, 控件位置会跟随“弹窗”位置。弹窗中支持最多 64 个显示控件、128 个触摸控件。

在弹出窗口中支持显示控件和触摸控件的三种典型应用如下图所示。详细介绍参见应用笔记《VGUS 串口屏在弹窗中显示控件》。



图 5-5-1A 在弹窗中设置数值



图 5-5-1B 在录入窗口中显示提醒范围



图 5-5-1C 在弹出菜单中显示可变报警信息

6 高级控件

VTc 串口屏新增多种控件功能（[固件日期 20250811 及以后版本支持](#)），具体用法本节逐一介绍。

SDWb 串口屏固件升级为 VTb 后支持高级控件功能（[升级方法](#)）。

6.1 高级控件汇总

高级控件通过“VGUS 开发工具->变量配置工具栏”添加，该工具栏各图标定义如图 6-1-1 所示。

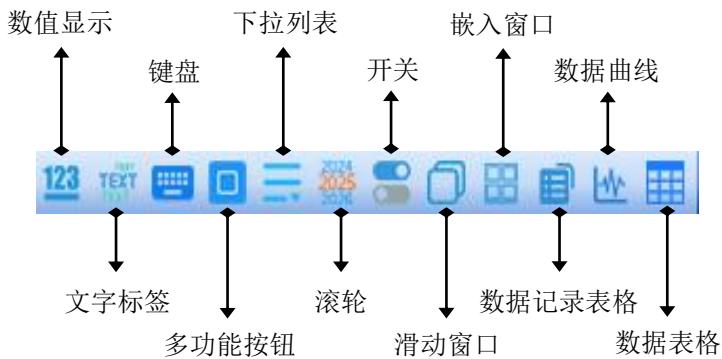


图 6-1-1 高级控件变量配置工具栏

表 6-1-1 高级控件汇总表

	功能	说明	VTc	SDWb	SDWn
1	数值显示	以指定格式显示数据变量。支持多种背景模式，支持 UTF8 编码单位及多语言切换。支持直接触发录入。	√	[1]	x
2	文字标签	显示文字标签中的多国语言字符串或变量存储器的字符串。	√	[1]	x
3	键盘控件	快速创建的数字键盘和中英文键盘。在弹窗页面显示键盘并录入数值或字符串数据。	√	[1]	x
4	多功能按钮	通过组合文字和背景显示按钮。支持多种按钮功能。	√	[1]	x
5	下拉列表	动态 UI，从弹出列表的待选项中选择一项。	√	[1]	x
6	滚轮	动态 UI，通过上下滑动的方式从待选项中选择一项。	√	[1]	x
7	开关	动态 UI，显示开或关两种状态，支持通过触摸或写变量地址的切换。通过触摸切换时支持过渡动画。	√	[1]	x
8	滑动窗口	动态 UI，支持通过滑动或写变量地址的方式切换窗口内的图片页面。	√	[1]	x
9	嵌入窗口	动态 UI，支持在窗口内滑动平移图片，实现在有限窗口中显示更多内容，实现超大幅面图片显示。	√	[1]	x
10	数据记录采样	以一定的时间间隔或特定条件触发时将一段数据保存下来，可以保存多组这样的数据。	√	[1]	x
11	数据记录表格	以表格方式显示数据记录采样的数据。支持数字及字符格式的数据显示。支持触摸滑动。	√	[1]	x
12	数据曲线	以曲线图的方式显示数据记录采样、变量存储器或曲线缓冲区数据。支持触摸滑动。	√	[1]	x
13	数据表格	在表格中以文本、数字、图标或格式化字符显示变量存储器数据。支持触摸滑动。	√	[1]	x

[1]. SDWb 串口屏固件升级为 VTb 后支持高级控件。

6.2 数值显示

数值显示控件用于显示数据变量。将背景和前景整合到一个控件，不需要提前在页面背景图上绘制数值显示的背景，使用更加方便。功能上可完全覆盖“数据变量控件”。

相较于数据变量控件，数值显示控件新增特性如下：

1. 支持范围设置，超出范围时支持显示字符或背景警告色；
2. 支持密码模式、数值转换、控件隐藏。
3. 支持闪烁显示字符。增强特殊场景下的提示作用。
4. 支持录入功能。录入时所需的变量类型、整数位数、小数位数、范围限制等属性可自动继承，无需重复设置。
5. 支持动画方式加载控件。加载类型、加载时间、加载效果均可设置。
6. 垂直方向自动居中对齐，水平方向对齐方式可设置。
7. 支持多种背景模式，单色背景、图标背景以及透明背景。位置布局设置更加灵活。
8. 支持字符间距设置及多种宽度调节模式，字符及小数点宽度调整更加灵活。
9. 支持背景圆角和背景边框设置。
10. 支持UTF8编码的单位字符显示，如°C。支持单位字符多语言一键切换，多语言应用更方便。

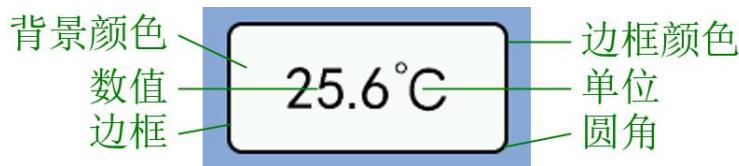


图 6-2-1 数值显示控件显示效果及部分元素说明

变量属性	
名称定义	数值显示2
变量存储地址(0x)	018C
初始值	0
功能设置属性	
变量类型	整数 (字)
整数位数	2
小数位数	1
无效位补零	<input type="checkbox"/>
启用范围限制	<input checked="" type="checkbox"/>
下限值	-1000
上限值	1000
警告色模式	不启用警告色
下限警告色	255; 255; 0
上限警告色	255; 0; 0
启用密码模式	<input type="checkbox"/>
启用数值转换	<input checked="" type="checkbox"/>
数值转换放值	1
数值转换偏移值	0
进入隐藏状态	<input type="checkbox"/>
闪烁间隔	不闪烁
启用录入	<input checked="" type="checkbox"/>
键盘控件所在页面	25
键盘显示位置	屏幕居中
非弹出区域的不透明度(0x)	FF
加载类型	从上飞入
加载时间	30
加载效果	线性
外观设置属性	
对齐方式	居中对齐
文本颜色	0; 0; 0
背景模式	单色背景
背景图标库ID	无
图标编号	0
背景颜色	255; 255; 255
Y方向点阵数	32
FONT0 ID	55
FONT1 ID	82
圆角	8
字符间距	2
边框宽度	2
边框颜色	0; 0; 0
宽度调整模式	所有字符均自动调整宽度
“单位”字符所用文字标签编号	单位
语言索引的变量存储地址	0000
描述指针(0x)	FFFF

图6-2-2 数值显示控件属性1

图6-2-3 数值显示控件属性2

数值控件主要从两个方面描述其属性，分别为功能属性和外观属性。

功能属性：

核心功能属性：变量存储地址、变量类型、整数位数、小数位数。

范围限制相关：警告色用于变量值超出范围时，修改背景颜色或文本颜色提示当前值超出范围。当启用录入，录入的数据超过范围时，点击确认按无效。

录入相关：设置键盘所在页面，并且该页面的键盘控件类型为数字键盘，否则无法正常录入数据。键盘显示位置可根据需求选择，默认相对屏幕居中弹出。只有弹出的键盘所在页面小于屏幕时，非弹窗区域的不透明度修改才能看到效果。

加载动画：支持控件从设定的方向飞入。加载动画使页面效果更加灵动。

外观属性：

文字：对齐方式、颜色、字体文件及字体大小、字符间距、宽度调整。

背景：背景模式、背景颜色、背景图标、边框、圆角。

单位：支持引用“文字标签”作为单位显示在数字之后。因为来自“文字标签”的字符固定为UTF8编码，FONT1 ID 需要选择 UNICODE 编码的字库。通过修改“语言索引的变量存储地址”的数据，可实现多语言一键切换。



6.3 文字标签

文字标签控件是一个文本显示类型的控件，主要用于多语言切换场景，同时具有文本变量和滚动文本控件的功能。

相较于文本变量和滚动文本控件，能够提供更多的显示方式和更好的显示效果，具体如下：

1. 支持多语言一键切换；
2. 支持文本居中显示等更多水平/垂直方向对齐方式设置；
3. 支持文本字符的闪烁显示、支持使用‘*’密码显示；
4. 支持文本框背景色、边框宽度、边框颜色、圆角、填充设置；
5. 支持文本触摸滑动（显示模式为多行显示并且文本显示范围大于文本框时）；
6. 支持水平方向循环滚动模式和来回滚动模式（显示宽度小于文本框且显示高度大于文本框时为垂直方向滚动）。
7. 支持动画方式加载控件。加载类型、加载时间、加载效果均可设置。
8. 支持点击控件触发录入，结合“键盘控件”实现中英文录入功能。注意：该功能仅“数据来源”为“变量存储器”时支持。

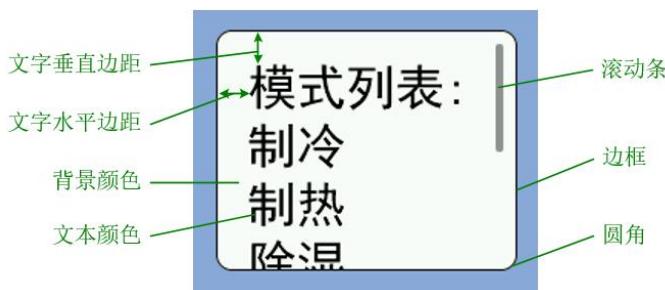


图 6-3-1 文字标签控件显示效果及部分元素说明

文字标签控件数据来源可以是文字标签工具中定义的字符串或变量存储器。

使用文字标签工具中定义的字符串作为文字标签控件的数据来源时，可以方便的实现多语言切换显示。该类应用中，需要借助“文字标签工具”，其是一个管理多语言字符串的工具。通过该工具预设多个字符串，每个字符串支持预设不同语言对应的字符，更改语言索引对应变量地址数据就可以切换到对应语言显示。

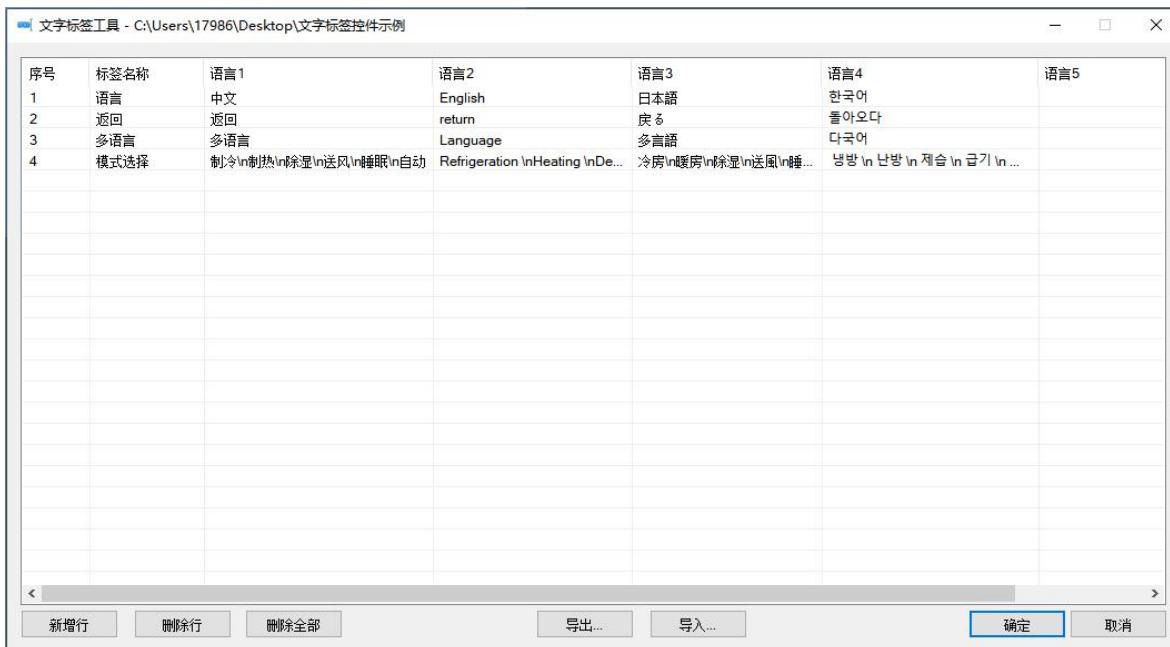


图 6-3-2 文字标签工具界面

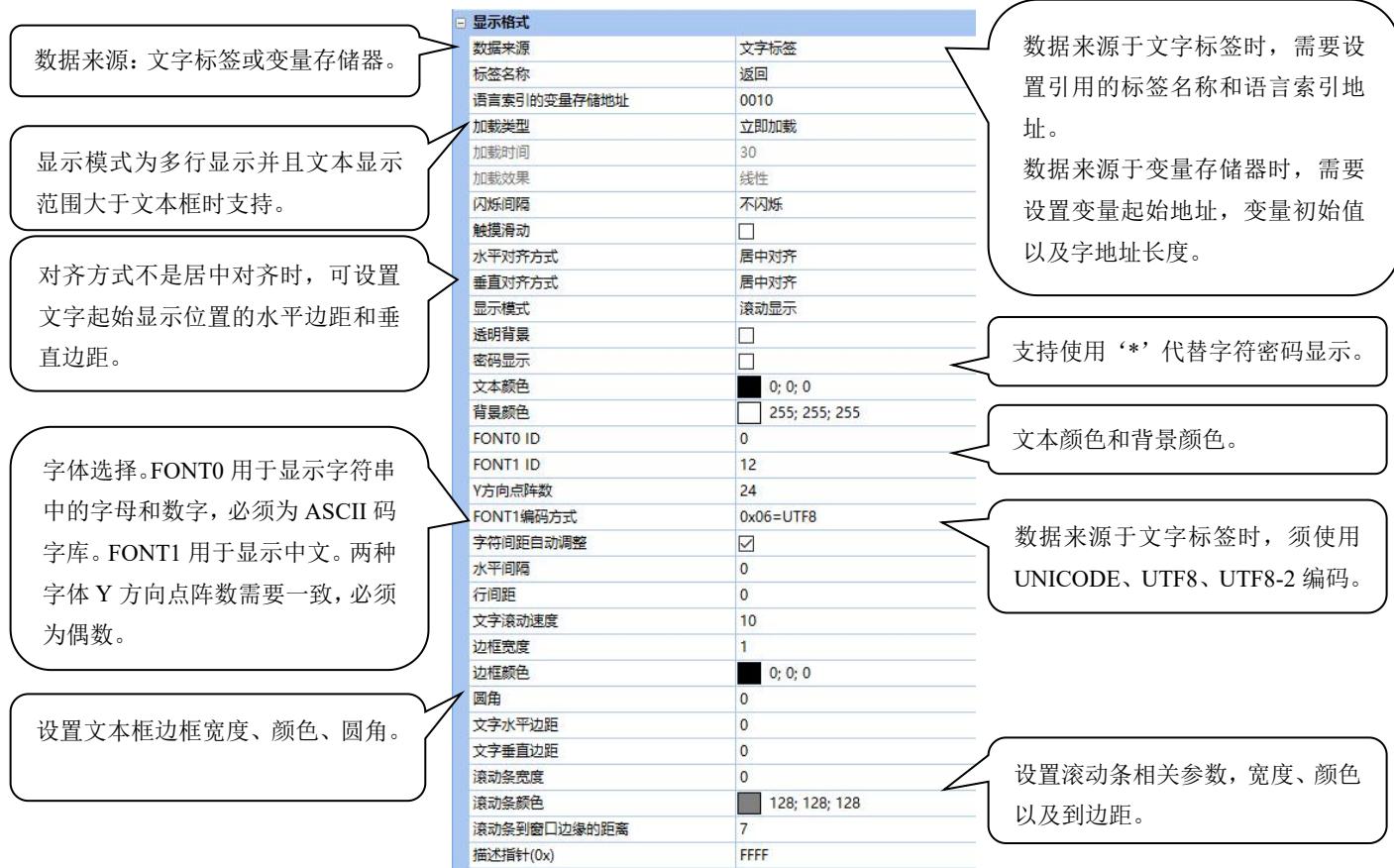


图 6-3-3 文字标签控件属性 (数据来自文字标签工具)



图 6-3-4 文字标签控件多语言显示示例 (数据来自文字标签工具)

如图6-3-3所示, 数据来源为文字标签, 并且选择的标签名称为“返回”。结合图6-3-2, 引用了文字

标签工具中的第二个标签。默认显示语言1对应的字符串。结合图6-3-3所示，修改语言索引地址（0x0010）的数据，数据为0显示“返回”，数据为1显示“return”，数据为2显示“戻る”，数据为3显示“돌아오다”。可以方便快速的实现切换其他语言所对应的字符串显示。

数据来源于文字标签时，必须使用UTF8编码。字符编码小于0x80使用FONT0字库，否则使用FONT1字库。FONT1字库需要为UNICODE字符集。新版字库提取软件支持UNICODE字库裁剪，支持分区裁剪、以及提取指定字符点阵信息两种方式，能够有效解决UNICODE字库文件过大、内存不足的问题。

数据来源	变量存储器
变量存储地址(0x)	0240
初始值	abcd
文本的字地址长度	16

图 6-3-5 文字标签控件属性（数据来自变量存储器）

使用变量存储器作为文字标签控件的数据来源时，可以覆盖文本变量和滚动文本控件的功能。如图6-3-5所示，此时要设置变量存储器的起始地址、初始值、以及最大字地址长度。

加载类型	从上飞入
加载时间	30
加载效果	线性

图 6-3-6 文字标签控件加载动画相关属性

文字标签控件支持动画方式加载控件。加载类型为“立即加载”时表示不使用加载动画。支持的加载类：从上飞入、从下飞入、从左飞入、从右飞入、左上飞入、右上飞入、左下飞入、右下飞入。加载时间可设置单位10ms，最小100ms，最大2550ms。支持线性、缓入、缓出、缓入缓出、过冲、反弹6种加载效果。

启用录入	<input checked="" type="checkbox"/>
键盘控件所在页面	24
键盘显示位置	屏幕居中
非弹出区域的不透明度(0x)	C0

图 6-3-7 文字标签控件录入相关属性

文字标签控件支持点击控件后弹出键盘页面以录入数据，通过勾选“启用录入”使能该功能。

仅数据来源为变量存储器时支持启用录入功能。键盘所在页面需要有键盘类型为“中英文键盘”的键盘控件才能进行字符录入。键盘显示位置可根据需要选择，可选择基于屏幕的相对位置和基于控件的相对位置。当键盘页面的大小小于屏幕大小时，四周的“空白”区域命名为“非弹出区域”，通过设置不透明度可以将该区域变暗，以实现突出显示键盘区域的效果。

6.4 键盘控件

键盘控件用于在页面上显示数字键盘或中英文键盘，配合数值显示控件（数字键盘）和文字标签控件（中英文键盘）等控件实现录入功能。用户无需关注键盘内部按钮的布局和繁杂的处理逻辑，将整个键盘作为一个整体，只需要设置各元素的颜色以及字体相关信息即可。

数字键盘和中英文键盘示例效果如图：

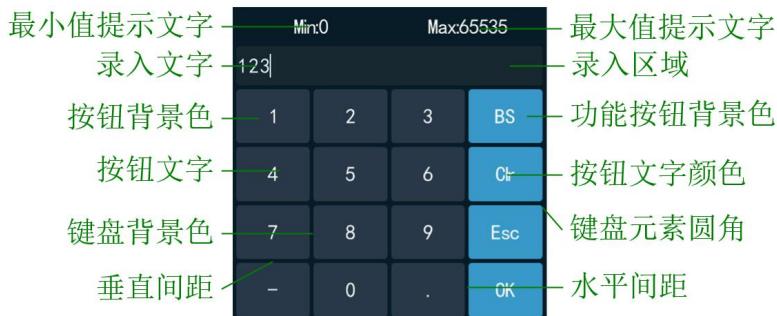


图6-4-1 键盘控件数字键盘效果示例及元素说明



图6-4-2 键盘控件中英文键盘效果示例及元素说明

键盘控件特性如下：

1. 拖拽即可创建键盘，方便快捷。键盘大小可按需调整。键盘所在页面建议与键盘大小一致。
2. 可在不同的页面创建多个键盘。典型应用分别创建一个数字键盘和中英文键盘。
3. 引用方式：键盘控件一般显示在“弹出页面”中作为键盘使用，根据是哪个控件触发的弹窗，获取对应控件的录入相关信息，如录入地址，数据类型时变量类型，文本类型时的编码方式等等。录入的数据也会根据相关控件的属性返回到对应的变量地址。实现多个、多种控件共用键盘的功能。
4. 以下控件支持使用“弹出页面”中的“数字键盘”控件进行录入：弹出菜单、数据录入、RTC录入、数值显示、多功能按钮。
5. 以下控件支持使用“弹出页面”中的“中英文键盘”控件进行录入：ASCII录入、GBK录入、文字标签（数据源为变量存储器）。
6. 支持数据录入、ASCII录入、GBK录入、RTC录入、文字标签、数值显示控件的初始值显示、数据自动上传、变量类型、整数位数、小数位数、ASCII录入固定为UTF8编码、GBK录入固定为GBK编码、录入状态返回等功能。
7. 可设置是否支持中文录入。特别是只需要英文录入的场合，将隐藏切换为中文录入的按钮、拼音及待选汉字，完全当做英文键盘使用，避免“中文”录入的干扰。
8. 支持录入GB2312、GBK、UTF8（UTF8-2）、UNICODE编码的中文，以适应当前广泛使用的UTF8编码。
9. 内置多种键盘配色风格，一键切换，快速调整键盘风格。

10. 支持录入数据密码显示。

键盘控件属性如下图所示：

显示格式	
键盘类型	数字键盘
数字键盘按键伴音起始编号	无
显示初始值	<input checked="" type="checkbox"/>
录入数据自动上传	<input checked="" type="checkbox"/>
录入数据密码显示	<input type="checkbox"/>
自动隐藏未使用按钮	<input type="checkbox"/>
启用范围限制并显示限制值	<input checked="" type="checkbox"/>
启用中文录入	<input type="checkbox"/>
启用录入状态返回	<input type="checkbox"/>
键盘元素水平间距	0
键盘元素垂直间距	0
键盘配色风格	深海探秘 (科技感)
键盘背景颜色	 15; 29; 42
键盘元素圆角	5
键盘背景图标库	无
键盘背景图标编号	0
键盘按钮背景颜色	 42; 59; 76
键盘功能按钮背景颜色	 58; 155; 203
键盘按钮文字颜色	 255; 255; 255
键盘按钮效果颜色	 0; 0; 0
键盘按钮背景不透明度(0x)	FF
键盘按钮文字不透明度(0x)	FF
键盘按钮效果不透明度(0x)	4C

图6-4-3 键盘控件数字键盘属性1

键盘按钮文字字库	55
键盘按钮文字字库Y方向点阵数	32
键盘按钮文字字符间距	0
录入文字字库FONT0	55
录入文字字库FONT1	无
录入文字字库Y方向点阵数	32
录入文字字符间距	0
录入文字颜色	 220; 240; 255
录入区域背景颜色	 25; 40; 55
录入区域背景不透明度(0x)	FF
录入字符间距自动调整	<input type="checkbox"/>
录入文字对齐方式	左对齐
数字键盘录入范围提示文字字库	55
数字键盘录入范围提示文字字库Y方向点阵	32
数字键盘录入范围提示文字颜色	<input type="checkbox"/> 255; 255; 255
拼音及待选汉字字库FONT0	0
拼音及待选汉字字库FONT1	无
拼音及待选汉字字库Y方向点阵数	16
拼音及待选汉字区域不透明度(0x)	FF
拼音及待选汉字颜色	 220; 240; 255
拼音及待选汉字区域背景颜色	 25; 40; 55

图6-4-4 键盘控件数字键盘属性2

说明：

1. 数字键盘支持按键音频伴音。按钮顺序为：1、2、3、4、5、6、7、8、9、0、-、.、BS、Clr、Esc、OK。如起始编号设置为21，则按钮“5”的音频伴音文件为25。
2. “按钮文字”“按钮背景”“录入区域背景”属性的不透明度设置为0x00时，这些元素将隐藏，背景图将完整显示。背景图上可以根据键盘布局设计自己的按钮样式。
3. 如果弹出页面有键盘控件，则认为使用键盘控件进行录入。隐藏数据录入、ASCII录入，GBK录入，RTC录入的“键盘文本窗口”相关属性和“键盘拼音窗口”相关属性。避免这两种窗口与键盘控件同时显示时造成的干扰。
4. 触发弹窗控件所需键盘类型和键盘控件不匹配时，如数据录入控件或数值显示控件，触发弹窗页面的键盘是中英文键盘。键盘按钮可点击，录入数据可以显示，但录入数据无效。
5. 有键盘控件的页面直接显示而不是作为弹窗显示时，键盘输入的“目标”未知，这时键盘可以显示和点击，但录入数据无效。
6. 数字键盘。

待录入数据有范围限制，键盘启用范围限制时：限制值正常显示，范围限制功能正常。

待录入数据有范围限制，键盘未启用范围限制时：不显示限制值，无范围限制功能。

待录入数据没有范围限制，键盘启用范围限制时：限制值显示为NULL，无范围限制功能。

7. 中英文键盘。支持中文录入的条件：

控件启用中文录入功能。

待录入字符编码方式为以下编码其中一种。GB2312、GBK、UNICODE、UTF8、UTF8-2。



6.5 多功能按钮

多功能按钮控件用于显示按钮并且可指定按钮的功能。通过背景或图标与文字的结合，简单快捷生成所需按钮，不再依赖背景图上的效果来呈现按钮，位置布局更加灵活。按钮的外观和文字支持更改以适应更复杂的应用场景。



图 6-5-1 多功能按钮控件显示效果及部分元素说明

多功能按钮控件特性如下：

1. 支持多种变量模式，以支持通过变量关联各类控件。支持数据自动上传。按钮状态更改或按钮功能触发自动上传对应变量数据。
2. 支持多种按钮工作模式。常量模式、点动模式、按钮状态模式、递增模式、递减模式、页面切换、弹出页面、关闭页面、密码切换页面。
3. 支持多种按钮状态。抬起状态、按下状态、长按状态、无效状态、隐藏状态。
4. VTb 系列按钮触摸后支持触发Lua脚本。
5. 支持多种背景模式。透明背景、单色背景、图标背景、图片切图背景。背景圆角和背景边框设置。
6. 支持多种按钮文字数据源模式。不显示文字、数据来源于文字标签、数据来源于变量存储器。支持不同的状态下显示不同的文字。支持多语言一键切换。
7. 支持动画方式加载控件。加载类型、加载时间、加载效果均可设置。
8. 支持按钮文字的显示模式、对齐方式、闪烁显示、颜色、字体文件、字体大小、编码方式、字符间距、字符宽度调整以及字符边距调整。

变量属性		外观设置属性	
名称定义	多功能按钮1	圆角	10
变量存储地址(0x)	018D	边框宽度	2
功能设置属性		边框颜色	0; 0; 0
音频文件	无	按钮背景模式	纯色
键码	无	抬起时的背景颜色数据	0; 162; 232
变量模式	按钮状态调节	按下时的背景颜色数据	173; 221; 11
数据自动上传	<input checked="" type="checkbox"/>	长按时的背景颜色数据	222; 105; 10
进入隐藏状态	<input type="checkbox"/>	无效时的背景颜色数据	128; 128; 128
进入无效状态	<input type="checkbox"/>	文字数据来源	文字标签
工作模式	按钮状态模式	标签名称	设置
按下时写入VP的常量值(0x)	0001	语言索引的变量存储地址	0010
抬起时写入VP的常量值(0x)	0002	文字显示模式	多行显示
长按时写入VP的常量值(0x)	0003	水平对齐方式	居中对齐
长按迟滞时间	5	垂直对齐方式	居中对齐
长按间隔时间	10	闪烁间隔	不闪烁
加载类型	立即加载	文本颜色	0; 0; 0
加载时间	30	FONT0 ID	55
加载效果	线性	FONT1 ID	82
		V方向点阵数	32
		FONT1 编码方式	0x06=UTF8
		字符间隔自动调整	<input type="checkbox"/>
		水平间隔	0
		行间距	0
		文字水平边距	0
		文字垂直边距	0
		描述指针(0x)	FFFF

图 6-5-2 多功能按钮控件属性1

图 6-5-3 多功能按钮控件属性2

9. 支持选择按钮功能的触发时机。支持功能触发时，写入常量值到指定地址。

10. 工作模式为“弹出页面”或“密码切换页面”，支持变量控制打开弹窗。

按钮状态说明：

抬起状态：按钮抬起或无触摸时的状态。

按下状态：按钮按下时状态。

长按状态：持续按下一段时间后触发长按状态。

长按迟滞时间：从按下到触发长按的时间间隔。

长按间隔时间：持续触发长按的时间间隔。

应用场景：用于需要按钮持续按下一段时间才触发或者按下持续触发的场合。

无效状态：按钮有显示但不支持触摸。

应用场景：用于提示当前按钮暂不能点击或按钮不具备权限的场合。

隐藏状态：按钮不显示且不支持触摸。

应用场景：用于需要临时隐藏按钮的场合。通过变量地址控制弹窗时不显示和触发按钮。

工作模式说明：

常量模式：在所选执行时机将常量值写入变量存储器。

点动模式：分别在按下和抬起时写入不同的常量值。

按钮状态模式：分别在按下、长按和抬起时写入不同的常量值。

递增模式：变量值依次增加。

递减模式：变量值依次减小。

页面切换：在所选执行时机切换到指定页面，并更新对应变量。

弹出页面：在所选执行时机弹出指定页面，弹出位置可设置。

关闭页面：在所选执行时机关闭弹窗页面并更新对应变量。

密码切换页面：在所选执行时机弹出页面，页面内数字键盘录入数据与变量数据一致才切换页面。

按钮背景说明：

透明背景：背景区域透明。每种状态均为透明背景。

单色背景：使用纯色作为背景。每种状态支持设置不同的颜色。

图标背景：使用图标作为背景。每种状态支持设置不同的图标。

图片切图背景：剪切图片的对应位置作为背景。每种状态支持设置不同的图片。

特别说明：长按状态时的背景数据未设置时，将使用按下状态的背景数据。

按钮文字说明：按钮文字显示在按钮背景之上。支持多种数据源模式。

不显示文字：适用于不需要显示文字的场景。

数据来源于文字标签：方便的实现多语言一键切换功能。

数据来源于变量存储器：文字可灵活更改。

特别说明：多种状态的文字位于同一个字符串中，使用','(0x2C)进行分隔。如按钮文字为“抬起,按下,长按,无效”依次表示抬起、按下、长按、无效时显示的文字。如按钮文字为“设置”表示所有状态均显示“设置”。如按钮文字为“,,无效”表示仅显示无效时的文字。需要多行显示的文字，支持使用‘\n’(0x0D)主动换行。如果对应状态无文字将使用抬起状态时对应的文字。



6.6 下拉列表

下拉列表控件支持从弹出的列表中选择一个选项并自动更新显示所选内容。选中项的“序号”会更新到变量存储器的指定地址，以告知哪一项被选中。列表中的选项内容超过列表窗口高度时，支持上下滑动显示选项内容。



图 6-6-1 下拉列表控件关闭状态示例

如图6-6-1所示，显示了下拉列表控件关闭状态的效果。在关闭状态时，指示图标为向左的箭头。



图 6-6-2 下拉列表控件打开状态示例

如图6-6-2所示，显示了下拉列表控件打开状态的效果。在打开状态时，指示图标顺时针旋转270度显示。可根据需要设计指示图标的内容。根据列表的初始值，列表有6个选项，打开状态时，只显示了3个选项，弹窗最大高度、字体Y方向点阵数、行间距这三个属性决定了当前显示选项的个数，当选项没有完全显示时，支持通过触摸上下滑动的方式显示和选择更多选项。

选项内容支持以下3种来源：纯数字、文本变量、文字标签。

纯数字：

选项内容为数字，控件自动将所设置的数字范围以字符方式展现出来并提供选择。

典型应用场景：年、月、日、时、分、秒等具有明显范围特征的数字类型的变量设置。

需要设置数字的开始值以及选项的个数。

文本变量：

选项内容为变量存储器中的文本字符串。字符串以' \n'（十六进制数据0x0A）作为每个选项分割符。

需要设置文本字符串在变量存储器的起始地址及字符串长度。根据分割符自动计算选项个数。

文字标签：

选项内容为文字标签中的字符串，字符串以' \n'（十六进制数据0x0A）作为每个选项分割符。

需要设置文字标签编号及语言索引控制地址。根据分割符自动计算选项个数。适用于多语言切换的应用场合。

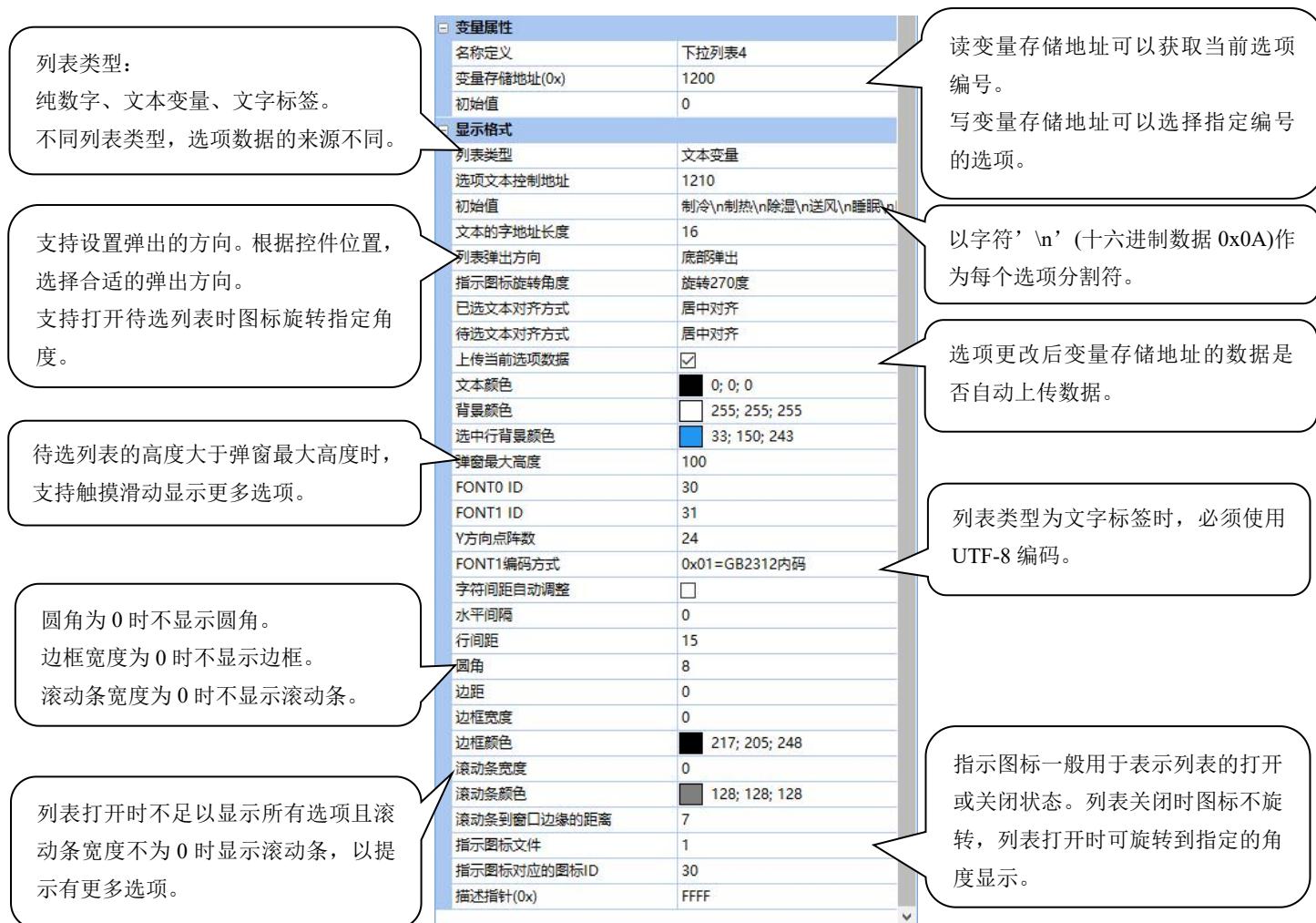


图 6-6-3 下拉列表控件属性

选项文本控制地址, 用于菜单选项更新标记及文本内容在变量存储器的起始地址。

ADDR: 更新标记。更新该值, 触发菜单选项文本内容的更新。用于避免内容过长, 不能一条指令发送完整文本内容时, 导致选项异常的情况, 确保内容完整后再更新标记, 刷新选项。

ADDR+1: 文本内容开始地址。

6.7 滚轮

滚轮控件支持从选项中以滚动的方式选择一个选项，并将选中项突出显示。选中项的“序号”会更新到变量存储器的指定地址，以告知哪一项被选中。

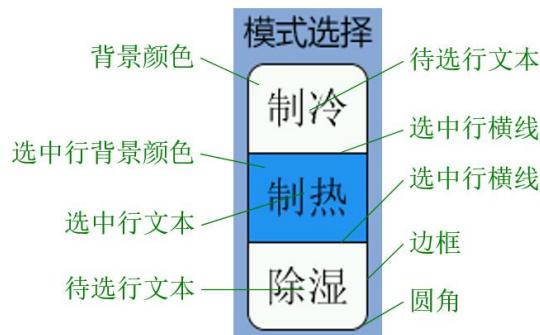


图 6-7-1 滚轮控件显示效果及部分元素说明

如图6-7-1所示，滚轮控件显示了3个选项，当前选中的选项为“制热”，背景色与上下未选中行有明显的区别。选中行和未选中行的字体颜色、背景颜色均支持单独设置。支持通过触摸上下滚动的方式切换到其他选项。

选项内容支持以下3种来源：纯数字、文本变量、文字标签。

纯数字：

选项内容为数字，控件自动将所设置的数字范围以字符方式展现出来并提供选择。

典型应用场景：年、月、日、时、分、秒等具有明显范围特征的数字类型的变量设置。

需要设置数字的开始值以及选项的个数。

文本变量：

选项内容为变量存储器中的文本字符串。字符串以’\n’（十六进制数据0x0A）作为每个选项分割符。

需要设置文本字符串在变量存储器的起始地址及字符串长度。根据分割符自动计算选项个数。

文字标签：

选项内容为文字标签中的字符串，字符串以’\n’（十六进制数据0x0A）作为每个选项分割符。

需要设置文字标签编号及语言索引控制地址。根据分割符自动计算选项个数。适用于多语言切换的应用场合。



图 6-7-2 滚轮控件属性

选项文本控制地址, 用于菜单选项更新标记及文本内容在变量存储器的起始地址。

ADDR: 更新标记。更新该值, 触发菜单选项文本内容的更新。用于避免内容过长, 不能一条指令发送完整文本内容时, 导致选项异常的情况, 确保内容完整后再更新标记, 刷新选项。

ADDR+1: 文本内容开始地址。

6.8 开关

开关控件有开、关两种状态，状态变化时支持动画过渡，根据变量模式将对应状态的变量值写入变量存储地址。



图 6-8-1 开关控件打开和关闭状态显示效果及部分元素说明

支持通过触摸点击或修改变量地址数据的方式更改状态。



图 6-8-2 开关控件属性

支持4种变量模式：按字调节、调节高字节、调节低字节、调节位，以适应不同的应用需求。

按字调节：将关闭/打开状态变量值完整写入变量地址。如变量值为0xAA55时打开某项功能。

调节高字节：将关闭/打开状态变量值低字节写入变量地址高字节。适用于修改变量高字节の場合。

调节低字节：将关闭/打开状态变量值低字节写入变量地址低字节。适用于修改变量低字节の場合。

调节位：关闭时对应位写0，打开时对应位写1。适用于按位修改变量值の場合。

开关控件支持禁止触摸点击功能，显示效果为当前状态“灰化”后的效果，以指示点击无效。用于临时关闭设置权限的场景。该功能动态控制需要使用描述指针。

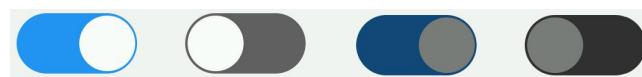


图 6-8-3 开关控件显示效果

典型效果如图6-8-3所示，自左到右分别对应打开状态、关闭状态、禁止触摸点击时的打开状态、禁止触摸点击时的关闭状态。

6.9 滑动窗口

在当前背景图片上定义一个特定矩形区域，称之为窗口。在该窗口中实现上、下、左、右四个方向滑动切换多张图片显示。滑动切换显示的图片上支持设计各种显示变量和触摸控件。滑动切换显示的图片大小通常与矩形区域窗口大小保持一致。当窗口设置为整个串口屏画面大小时，滑动窗口可以实现滑动切换翻页的效果。当设置为透明背景时，可以实现类似安卓系统的桌面图标滑动翻页的效果，而背景不变。

如图 6-9-1 所示的网格布局，实现 1 号图片与 2 号图片之间左右滑动切换，并且支持循环切换；1 号图片与 3 号图片之间上下滑动切换。

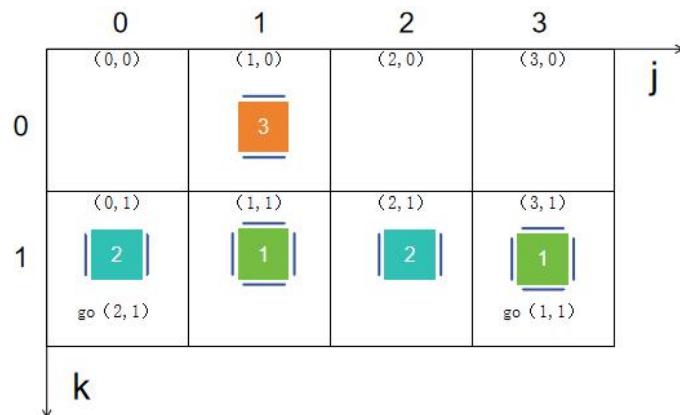


图 6-9-1 滑动窗口网格布局示例

网格布局说明：

1. 每个网格都有唯一的位置编号 (j,k)；
2. 有颜色图块的网格表示已使用，为有效网格，图中有效网格个数为 5；
3. 颜色图块中的编号即为该网格所引用的图片编号；
4. 图块四周的线条表示该网格所支持的滑动方向，有线条表示支持该方向的滑动；
5. 图块下方的“go (j,k)”表示滑动切换到该网格的动画播放完后，立即跳转到指定网格位置，以实现页面循环切换的功能。



图 6-9-2 滑动窗口控件属性

窗口背景显示模式可以设置为图片背景、透明背景、以及单色背景三种模式。设置为透明背景时，可以实现类似安卓系统的桌面图片滑动翻页的效果，而背景不变。设置为图片背景时，实现带背景图片滑动翻页的效果。设置为单色背景时，可以提升滑动流畅度。

注意：滑动窗口控件只是属于某个图片页面内，窗口内滑动切换显示图片时，不会更改 VGUS 寄存器中的图片页面 ID 号（PIC_ID）。通过 VP 地址指示当前正在显示的网格位置。

滑动窗口控件所引用图片页面中不再支持滑动窗口控件。滑动窗口尺寸不宜过大，否则动态效果变差。

结合图 6-9-1 所示的网格布局，5 个有效网格的网格配置数据的配置页面如图 6-9-3 所示：

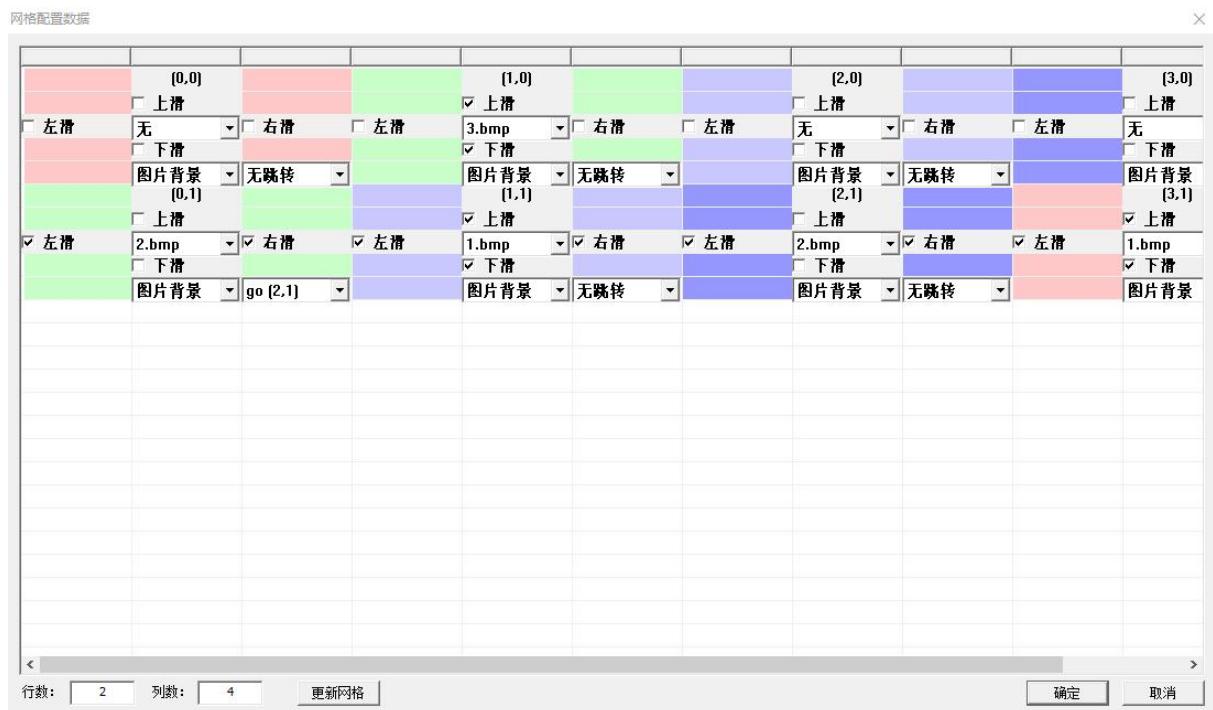


图 6-9-3 滑动窗口控件网格配置数据

如图 6-9-3 所示，网格配置数据使用步骤如下：

1. 根据设计需求，画出图6-9-1所示的简图。在左下角设置“行数”和“列数”然后点击“更新网格”；
2. 设置有效网格位置对应的图片ID；如“3.bmp”、“2.bmp”、“1.bmp”。“无”表示该网格未启用，为无效网格；
3. 设置滑动方向。支持“上滑”、“下滑”、“左滑”、“右滑”；
4. 设置背景类型，支持图片背景、透明背景、以及单色背景三种模式；
5. 设置是否需要跳转以及跳转的网格位置（用于循环切换）。

6.10 嵌入窗口

在当前背景图片上定义一个特定矩形区域，称之为窗口。在该窗口中显示另一张图片，称之为窗口图片。当窗口图片分辨率大于窗口尺寸时，窗口中支持上、下、左、右四个方向滑动显示窗口图片。窗口图片上支持设计各种显示变量和触摸控件。该控件适用于在窗口内显示大于窗口尺寸内容的应用场景，如长图显示。



图 6-10-1 嵌入窗口控件显示效果及部分元素说明

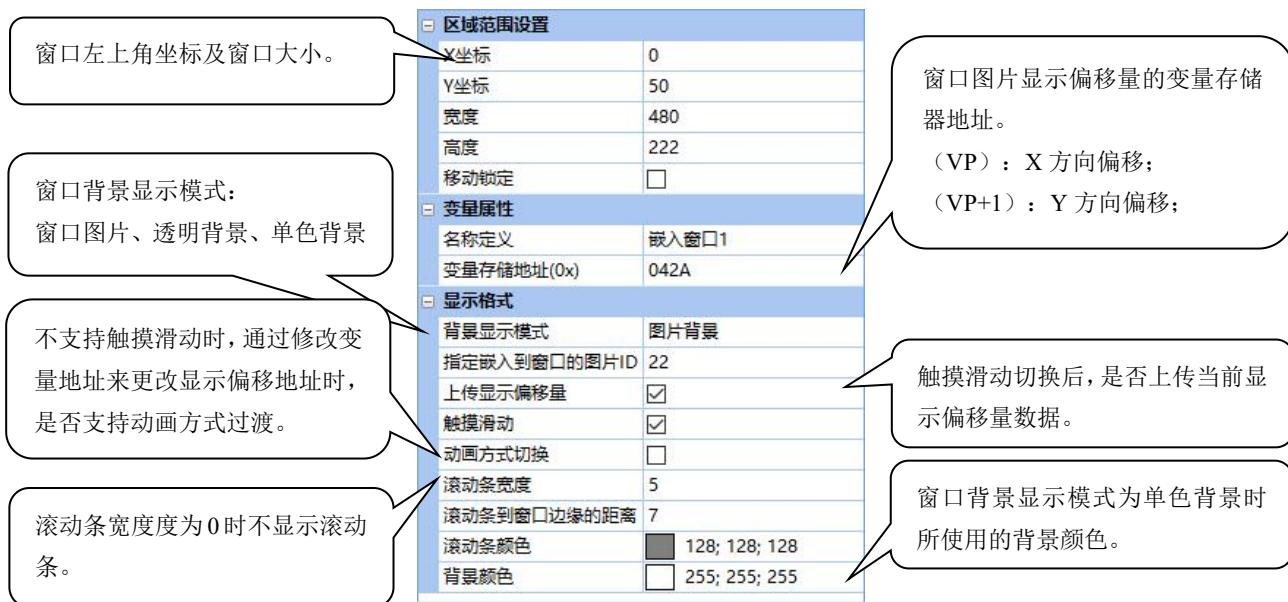


图 6-10-2 嵌入窗口控件属性



图 6-10-3 嵌入窗口控件显示位置示例（定义窗口大小 480x222）



图 6-10-4 嵌入窗口控件的嵌入图片示例（长图大小 480x1200）

如图 6-10-3 和图 6-10-4 所示，将大小为 480x1200 的图片嵌入到 480x222 大小的窗口显示。通过上滑动的方式，可以将更多的图片内容展示出来。显示的偏移地址存储在变量地址 VP 中，VP 表示 X 方向偏移量，VP+1 表示 Y 方向偏移量。当从其他页面切换回当前页面时，根据 VP 变量地址数据偏移到上次显示的位置，相当与具有位置记忆的功能。当然也可以通过修改 VP 变量地址数据的方式更改控件的显示偏移量，以到达切换到该页面时就显示在指定的位置。

图 6-10-4 所示的图片中可任意组合各类显示及触摸控件，以满足个性化的需求。

嵌入窗口控件所引用图片页面中不再支持嵌入窗口控件。嵌入窗口尺寸不宜过大，否则动态效果变差。

6.11 数据记录采样

数据记录采样是指以一定的时间间隔或特定条件触发时将一段数据保存下来，可以保存多组这样的数据。支持使用“数据记录表格”控件或“数据曲线”控件显示已采样数据，以观察数据的变化、数据还原等功能。

数据记录采样配置工具通过“VGUS 开发工具->工具->数据记录采样配置”打开。



图 6-11-1 数据记录采样配置工具

配置属性说明：

采样模式： 定时采样或触发采样。

定时采样： 以一定的时间间隔周期采样。

采样周期： 间隔多长时间采样一次。采样周期小于1秒不支持掉电存储。

周期单位： 0表示秒，1表示毫秒。

触发采样： 触发地址数据控制采样。

触发地址： 触发采样的变量存储地址。

触发模式： 上升沿触发采样（数据从0变为非0）、下降沿触发采样（数据从非0变为0）、数据变化触发采样（不支持复位）。

复位： 采样后自动复位触发地址数据。上升沿触发时复位为0。下降沿触发时复位为1。

注意： 触发采样时，为确保数据被完整采样，请启用复位功能，且确认触发地址的数据已复位后再更新采样数据。

记录条数： 记录多少次采样的数据。一般存储记录条数达到上限后，使用循环存储的方式以记录最新的数据。范围10-1000000。实际可记录条数与单条记录大小以及FLASH容量有关。

数据地址： 采样起始变量存储地址。

数据类型： UINT16、INT16、UINT32、INT32、UINT64、INT64、STRING。

数据个数： 采集数量（连续地址），范围 1~1000。比如要记录 100 个 INT16 类型的数据，那么单次



采样的数据量为 100 个字地址 (200 字节) 的长度。类型为 STRING 时表示字符串最大字长度, 遇到结束符 (0xFF 或 0x00) 提前结束。(“数据记录表格”控件支持使用逗号 ‘,’ 分割显示字符串类型的数据。用于单次采样中多种类型混合的场景。)

小数位数: 数值类型时有效, 数据的小数位数。仅用于导出数据时格式化。

高低互换: 采样字符数据时交换高低字节。仅数据类型为 STRING 时有效。

导出标题: 设置是否支持导出标题。

标题文字标签编号: 导出数据时标题所用文字标签编号。

标题语言索引地址: 在文字标签中设置多语言标题, 可通过改地址选择导出时所用的语言。

采样控制地址: 控制采样是否开启。不启用该功能时, 采样默认开启。

控制模式: 上升沿开启 (数据从 0 变为非 0)、下降沿开启 (数据从非 0 变为 0)。

控制地址: 数据清除, 数据导出。

ADDR: 控制。执行完成后清零。

写 0x33CC 清除数据。

写 0x55AA 导出数据到 SD 卡/U 盘, 使用默认文件名;

写 0x55BB 导出数据到 SD 卡/U 盘, 使用自定义的文件名 (自定义文件名长度最长支持 60 字节);

ADDR+1: 状态。注意: 功能不正常时, 可查看状态地址的数据排查原因。

0: 默认状态, 正常采样;

1: 未采样, flash 空间不足;

2: 未采样, 内存空间不足;

3: 正在清除数据;

4: 清除数据完成;

5: 正在导出数据;

6: 导出完成;

7: 导出失败, 读 flash 异常;

8: 导出失败, 写 SD 卡异常;

ADDR+2: 自定义的文件名开始地址。

掉电存储: 采样数据是否支持掉电存储。掉电存储只支持添加采样数据, 不支持修改, 插入, 删除已采样的数据。不启用掉电存储时, 采样数据存储的变量存储器, 需设置开始存储地址。如果变量存储器不足以存储所有采样次数的数据时将不会启动采样。

掉电存储	
占用块数	12
存储起始地址(0x)	FFFF
占用变量数(0x)	00000000

掉电存储	
占用块数	0
存储起始地址(0x)	8000
占用变量数(0x)	00004E20

图 6-11-2 数据记录采样存储位置及大小

注意: 掉电存储时从 flash 尾部向前存储数据, 请充分考虑 flash 大小, 工程文件大小, 空间不足将不会启动采样。虽然虚拟屏能正常采样, 但是实际屏受上述因素的影响, 可能不能正常采样。

6.12 数据记录表格

数据记录表格控件以表格形式显示已采样数据。

如图 6-12-1 示例 1 所示, 首行作为“标题行”, 依次显示了各个标题字符。其它行依次显示了采样数据的序号、采样时间以及具体的采样数据。

例 1 中采样的数据有 16 个通道, 对应通道 0-通道 15。水平滚动条提示了右侧还有更多数据。左右滑动可显示更多通道的数据。该滚动条支持点击并拖动以便快速显示指定位置的数据。

支持显示多次采样的数据。支持设置是否显示该次采样的相对序号和采样时间。垂直滚动条提示了底部还有更多采样数据。上下滑动可显示更多采样数据。该滚动条支持点击并拖动以便快速显示指定位置的数据。

采样数据类型为 UNIT16, 并且小数位数均为 1。

序号	时间	通道0	通道1	通道2
1	2025/04/29 18:45:20	0.0	0.1	0.2
2	2025/04/29 18:45:18	0.0	0.1	0.2
3	2025/04/29 18:45:16	0.0	0.1	0.2
4	2025/04/29 18:45:14	0.0	0.1	0.2
.....				

图 6-12-1 “数据记录表格”控件效果示例 1

如图 6-12-2 示例 2 所示, 与示例 1 相比, 没有显示“序号”列; 标题对应的字符有所变化; 采样的数据类型为 STRING, 单次采样的数据只在一列显示。表格宽度大于等于所设置的列宽之和, 所以没有水平滚动条。采样的数据以字符串方式解析并显示; 启用了奇偶行异色功能, 使表格整体更有层次感。

时间	详情
2025/04/29 19:22:12	按钮1触发记录
2025/04/29 19:22:10	按钮2触发记录
2025/04/29 19:22:09	按钮3触发记录
2025/04/29 19:21:58	按钮3触发记录

图 6-12-2 “数据记录表格”控件效果示例 2

如图 6-12-3 示例 3 所示, 与示例 2 相比, 采样类型同样为 STRING, 采样数据分为多列显示。首先需要在控件属性中启用“字符分割显示”, 采样数据使用逗号 ‘,’ (十六进制 0x2C) 分割。如序号为 1 的采样数据为 “22.6,40,210.23,10.75,正常”。从表格上看起来采样记录的是数字, 实际记录的是字符串。采样数据的小数位数不一致、或多种采样类型混合的应用场景下, 可用这种统一转换为字符串再分割显示的方式。

序号	时间	温度/°C	湿度/%RH	电压/V	电流/A	备注
1	2025/04/29 19:36:13	22.6	40	210.23	10.75	正常
2	2025/04/29 19:36:12	42.6	50	217.50	13.11	温度异常
3	2025/04/29 19:36:11	22.6	40	210.23	10.75	正常
4	2025/04/29 19:36:09	42.6	60	217.50	17.03	温度、电流异常
5	2025/04/29 19:36:08	42.6	50	217.50	13.11	温度异常
6	2025/04/29 19:36:07	42.6	60	217.50	17.03	温度、电流异常
7	2025/04/29 19:36:06	42.6	50	217.50	13.11	温度异常
8	2025/04/29 19:36:05	22.6	40	210.23	10.75	正常
	2025/04/29					

图 6-12-3 “数据记录表格”控件效果示例 3

上述3种示例的动态效果，可打开对应的案例工程，使用虚拟串口屏查看。

标题行数据支持4种模式，分别对应来自不同的数据来源：不显示标题、文字标签、变量存储器、默认标题。

不显示标题：

适用于不需要显示标题行的场景。

文字标签：

标题行内容为变量存储器中的文本字符串。字符串以'，'（十六进制数据0x2C）作为分割符。

需要设置文本字符串在变量存储器的起始地址及字符串长度。适用于多语言切换的应用场合。

变量存储器：

标题行内容为文字标签中的字符串。字符串以'，'（十六进制数据0x2C）作为分割符。

需要设置文字标签编号及语言索引控制地址。适应于更改标题行数据的应用场合。

默认标题：

标题行内容为内部默认值。

序号列使用英文字符"Number"；时间列使用英文字符"Time"；数据列从1开始编号的数字。

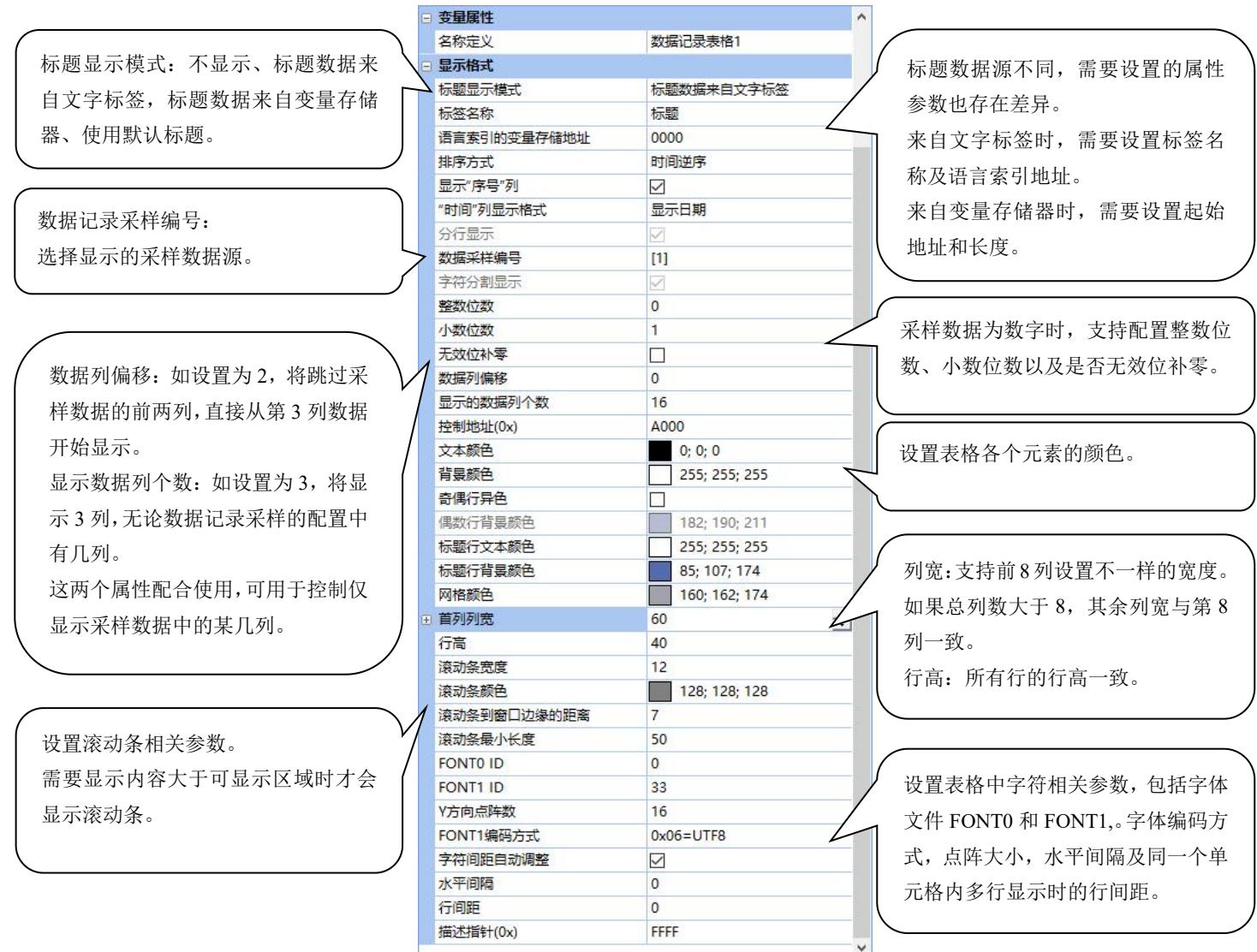


图 6-12-4 数据记录表格控件属性

控制地址: 通过变量存储地址间接控制表格翻页。

ADDR: 翻页控制。1: 下一页; 2: 上一页; 3: 右一页; 4: 左一页。

ADDR+1: 时间戳定位。UINT16类型。以天为单位的时间戳数据。一般配合下拉列表控件使用。

6.13 数据曲线

数据曲线控件以曲线形式显示数据记录采样、变量存储器、曲线缓冲区这三种数据源中的数据。

背景网格、刻度数据等均由控件绘制，不需要提前制作背景。

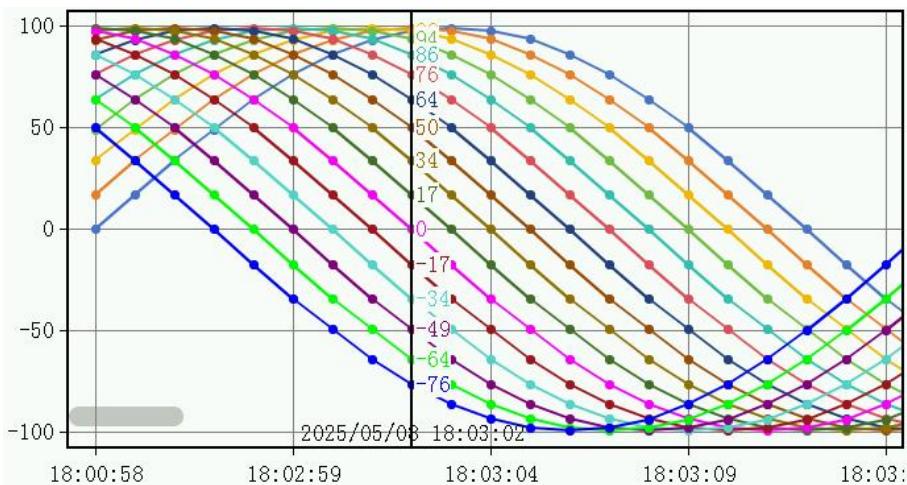


图 6-13-1 数据曲线控件效果示例 1

如图 6-13-1 所示，曲线数据来源为“数据记录采样”，X 轴表示采样时间，Y 轴表示数据大小范围。

图表各元素说明：

X 轴：采样时间、X 轴刻度线；

Y 轴：数据大小范围，Y 轴刻度线；

背景线条：边框线，水平网格线，垂直网格线；

曲线：线宽及数据点大小均可设置。每个曲线支持设置单独的颜色；

光标：点击曲线，可显示该位置的光标、数据值以及采样日期和时间。再次点击可隐藏；

滚动条：窗口不足以显示所有数据时滚动条出现。默认为半透明状态。支持拖动滚动条快速翻页；

触摸滑动：当出现滚动条（滚动条宽度不为 0）时，支持通过触摸左右滑动预览数据。

不同数据来源说明：

数据记录采样：以曲线图的方式将采样数据显示出来。单个控件支持同时显示 16 条曲线。支持超大数据量（上万条采样数据）的快速浏览。示例效果参考图 6-13-1。

变量存储器：以曲线图的方式将变量存储器数据显示出来。相较与曲线缓冲区数据只能添加新数据，变量存储器支持修改任一点的数据，甚至修改整个曲线数据。

如图 6-13-2 所示，曲线数据来源为变量存储器。每次更新时，只需要更新 2 组数据，其中 1 组为当前最新的数据，另一组为“越界”值。将控件属性中的“越界显示方式”设置为“不显示数据点及相邻线段”，这样就会“空”出一段来，以模拟“当前刷新位置”的效果。依次更改不同位置的数据，达到“循环刷新”曲线的整体效果。

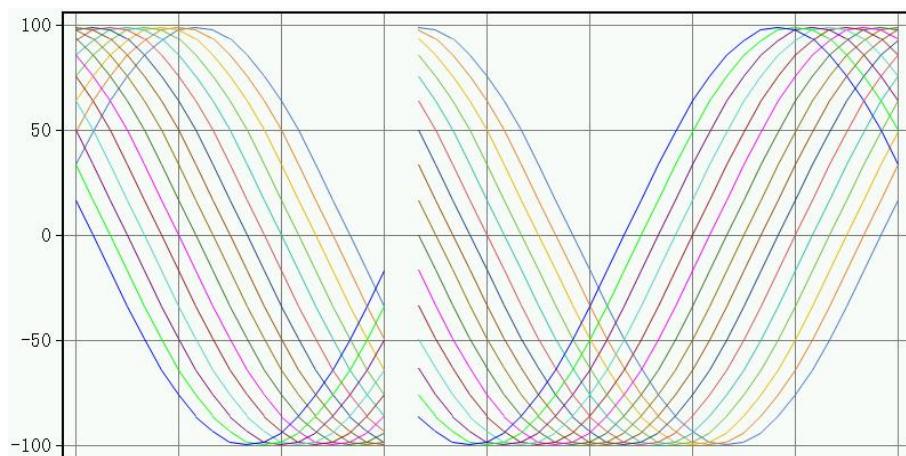


图 6-13-2 数据曲线控件效果示例 2

如图 6-13-3 所示, 曲线数据来源为变量存储器。支持每次更新所有曲线数据, 以达到快速刷新整个曲线的显示效果。实现类似示波器的显示功能。

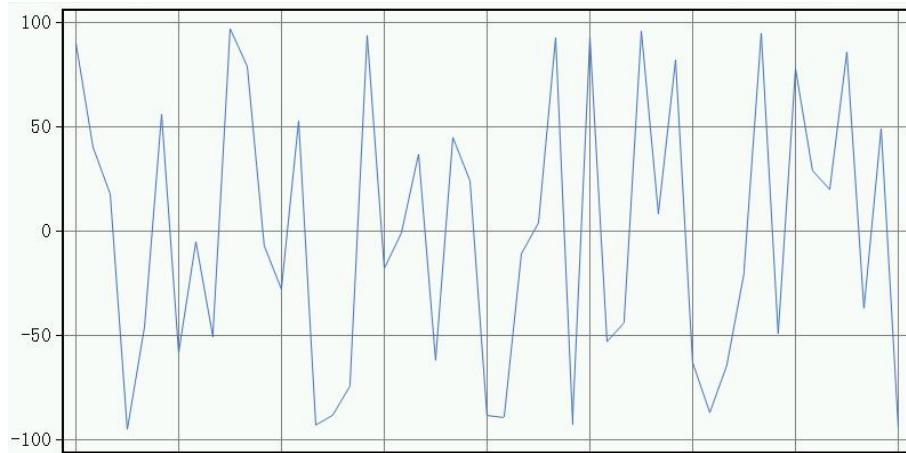


图 6-13-3 数据曲线控件效果示例 3

曲线缓存区: 以曲线图的方式将曲线缓存区数据显示出来。曲线缓存区只能添加新的数据, 控件自动显示最新的数据。如图 6-13-4 所示, 显示曲线缓冲区通道 0 数据。相较于“实时曲线”控件, 数据曲线控件元素得到了极大的丰富, 可以满足更多功能需求。

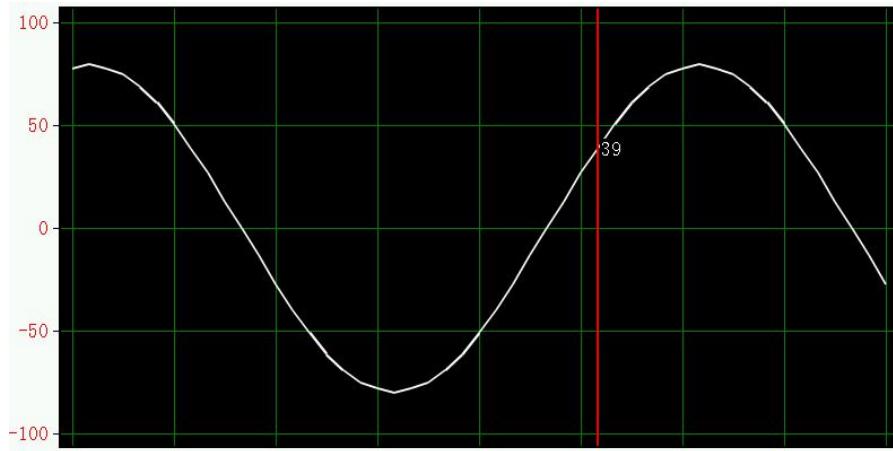


图 6-13-4 数据曲线控件效果示例 4

控件属性说明：

变量属性	
名称定义	数据曲线1
显示格式	
数据来源	数据采样
数据采样编号	[1]
通道偏移	0
通道数量 (曲线个数)	16
数据类型	INT16
控制地址(0x)	7000
曲线区域背景透明显示	<input type="checkbox"/>
刻度区域背景透明显示	<input type="checkbox"/>
显示水平网格线	<input checked="" type="checkbox"/>
显示垂直网格线	<input checked="" type="checkbox"/>
显示X轴刻度线	<input checked="" type="checkbox"/>
显示Y轴刻度线	<input checked="" type="checkbox"/>
显示X轴文字	<input checked="" type="checkbox"/>
显示Y轴文字	<input checked="" type="checkbox"/>
显示光标及当前值	<input checked="" type="checkbox"/>
X轴时间显示方式	仅显示时间
网格线类型	实线
无效位补零	<input type="checkbox"/>
字符宽度自动调整	<input type="checkbox"/>
越界显示方式	正常显示
曲线宽度	2
数据点半径	3
数据X轴像素间隔	30
X轴刻度 (网格线) 间隔数据个数	5
网格线为虚线时线段长度	4
网格线为虚线时空白长度	4
Y轴刻度 (网格线) 个数	5
X轴和Y轴刻度长度	5
刻度与文本间距	5
Y轴数据开始值	-100
Y轴数据范围	200
水平方向填充	20
垂直方向填充	10
刻度区域大小	100

图 6-13-5 数据曲线控件属性 1

通道颜色	
1号通道颜色	[72; 116; 203]
2号通道颜色	[238; 130; 47]
3号通道颜色	[242; 186; 0]
4号通道颜色	[117; 189; 66]
5号通道颜色	[48; 192; 180]
6号通道颜色	[229; 76; 94]
7号通道颜色	[37; 67; 128]
8号通道颜色	[158; 76; 13]
9号通道颜色	[145; 112; 0]
10号通道颜色	[70; 113; 40]
11号通道颜色	[255; 0; 255]
12号通道颜色	[160; 23; 39]
13号通道颜色	[82; 212; 201]
14号通道颜色	[128; 0; 128]
15号通道颜色	[0; 255; 0]
16号通道颜色	[0; 0; 255]
背景颜色	[255; 255; 255]
刻度区域背景颜色	[255; 255; 255]
网格线颜色	[128; 128; 128]
光标颜色	[0; 0; 0]
文本颜色	[0; 0; 0]
滚动条宽度	15
滚动条颜色	[128; 128; 128]
滚动条到窗口边缘的距离	15
滚动条最小长度	50
字符间距	0
边框宽度	2
边框颜色	[0; 0; 0]
整数位数	0
小数位数	0
FONT0 ID	0
Y方向点阵数	18
描述指针(0x)	FFFF

图 6-13-6 数据曲线控件属性 2

数据来源：

数据记录采样：按照采样时间顺序显示采样数据的历史变化趋势。

数据记录采样编号：所引用的数据记录采样。

通道偏移：从采样数据的第几个通道开始显示。

变量存储器：显示多个变量地址数据的变化趋势。

变量存储地址：数据开始地址。

数据个数：每个通道显示数据个数。

注意：实际所需地址长度与数据类型、数据个数及通道数量有关。如地址设置为0x0100，数据类型为UINT16（每个数据占用1个字地址），数据个数为10，通道数据量为3。那么 (addr+0) 表示通道0的第一个数据、(addr+1) 表示通道1的第一个数据、(addr+2) 表示通道2的第一个数据、(addr+3) 表示通道0的第二个数据、(addr+4) 表示通道1的第二个数据、(addr+5) 表示通道2的第二个数据依次类推。所需字地址总长度为 (1*3) *10。

曲线缓冲区：显示曲线缓冲区数据的变化趋势。

开始通道编号：曲线缓冲区通道编号。



最多显示数据个数: 仅显示最新的 N 个数据。

通道数量(曲线个数): 显示的曲线数量, 最多 16 个。

数据类型: UINT16、INT16、INT32、UINT8、INT8。数据源为数据记录采样时仅支持 UINT16、INT16、INT32 这 3 种类型, 并且数据类型跟随所选数据记录采样设置, 当前参数设置无效。数据源为实时曲线时仅支持 UINT16、INT16 这两种类型。

控制地址: addr:通道隐藏控制。bit 为 1 时表示隐藏该通道。

addr+1:翻页控制。1:下一页;2:上一页;3:首页;4:尾页。

addr+2:时间戳控制。配合下拉列表显示所选日期的数据(时间戳控制功能仅数据源为数据记录采样时支持)。

曲线区域背景透明显示、刻度区域背景透明显示: 勾选时启用对应功能。

显示水平网格线、显示垂直网格线、显示 X 轴刻度线、显示 Y 轴刻度线、显示 X 轴文字、显示 Y 轴文字、显示光标及当前值: 勾选时启用对应功能。

X 轴时间显示方式: 仅显示时间、仅显示日期、分行显示日期和时间、分行显示日期和时间。(仅数据源为数据记录采样时支持该功能)

网格线类型: 实线、虚线。为虚线时支持设置线段长度和空白长度。

无效位补零: 适用于数字的显示, 包括 Y 轴刻度值, 光标线当前值。

字符宽度自动调整: 适用于所有字符显示。

越界显示方式: 数据大于最大值或小于最小值时的处理方式。正常显示、调整到边界值显示、不显示数据点及相邻的线段。

曲线宽度: 最小宽度为 1。

数据点半径: 与曲线宽度不同, 可用于突出显示数据点。设置为 0 不显示数据点。

数据 X 轴像素间隔: 设置为 20, 表示每间隔 20 个像素显示一个数据。

X 轴刻度(网格线)间隔数据个数: 设置为 3, 表示每间隔 3 个数据, 显示 1 个垂直网格线及 X 轴刻度。

Y 轴刻度(网格线)个数: Y 轴刻度及水平网格线的个数。

X 轴和 Y 轴刻度长度: 刻度显示在边框外侧。

刻度与文本间距: X 轴文本显示在刻度下方, Y 轴文本显示在刻度左侧。

Y 轴数据开始值、Y 轴数据范围: Y 轴刻度对应的最大值和最小值。

水平方向填充、垂直方向填充: 边框内侧四周的“空白”区域大小。

刻度区域大小: 在框选曲线区域的左侧和底部显示 Y 轴和 X 轴刻度及字符的区域大小。

通道颜色、背景颜色、刻度区域颜色、网格线颜色、光标颜色、文本颜色: 颜色设置。

滚动条宽度、滚动条颜色、滚动条到窗口边缘的距离、滚动条最小长度: 滚动条相关参数。

字符间距: 字符间的额外间隔。

边框宽度、边框颜色: 边框相关参数。

整数位数、小数位数: 适用于数字的显示, 包括 Y 轴刻度值, 光标线当前值。

FONT0 ID、Y 方向点阵数: 字库相关参数。

描述指针: 描述指针地址。

注意: 数据曲线控件属性繁多, 建议在示例工程中, 修改相关属性后使用虚拟屏查看对应效果, 以便快速准确的理解相关参数的作用。



6.14 数据表格

在表格中以文本、数字、图标或格式化字符方式，显示变量存储器中的数据。

数据表格可用于展示列表信息、设备信息或批量数据，支持通过数字、文字、图标等形式呈现内容。

数据表格简单示例如图 6-14-1 所示。

序号	长度	宽度	高度
1	0.00	0.00	0.00
2	0.00	0.00	0.00

图 6-14-1 数据表格控件显示效果及部分元素说明

数据表格控件相关特性说明如下：

1. 表格的行数目和列数目决定了显示的单元格数量，每个单元格显示的内容对应到具体的变量地址。
2. 数据区域支持显示数值、文本、图标以及格式化数据。通过“数据模式”属性配置。
3. 支持修改表格内容，通过“支持编辑表格内容”属性启用。“键盘控件”录入数值或文本数据，单元格类型为“变量图标”时点击对应单元格修改图标。
4. 启用“标题行”时，最终行数目在设置行数目的基础上增加一行。标题行数据来自文字标签，支持多语言一键切换，使用英文逗号 ‘,’ (0x2C)分隔每列所显示的字符。
5. 启用“序号列”时，最终列数目在设置列数目的基础上增加一列。序号列显示数字，从 1 开始编号。
6. 支持显示大量数据，当表格所需的显示范围大于显示窗口时显示滚动条，并且支持通过触摸滑动以显示更多内容。滚动条可用于提示当前显示位置和大小相对于整个表格的对应关系。
7. 对于表格行数目动态变动的场景，如实时扫描的 wifi 列表，可以结合描述指针动态的设置行数目。
8. 表格数据模式：配置从变量存储器获取数据的类型。

行数目	20
列数目	10
表格数据起始地址(0x)	1000
数据模式	单元格数值数据
数据类型	整数 (字)
整数位数	2
小数位数	2

行数目	4
列数目	7
表格数据起始地址(0x)	3000
数据模式	单元格文本数据
单元格数据“字长度”	16

行数目	16
列数目	2
表格数据起始地址(0x)	5000
数据模式	单元格自定义数据
单元格数据“字长度”	8

图 6-14-2 单元格数值数据

图 6-14-3 单元格文本数据

图 6-14-4 单元格自定义数据

如图 6-14-2 所示，表格数据区域内容均为数值，数据类型为整数，占用 1 个字地址，行数目 20，列数目 10，总字长度为 $20*10*1=200$ 。起始地址为 0x1000。表格显示变量地址 0x1000–0x10C7 范围内的数据。

如图 6-14-3 所示，表格数据区域内容均为文本，每个单元格的文本字长度最大 16，行数目 4，列数目 7，总字长度为 $4*7*16=448$ 。起始地址为 0x3000。那么表格显示变量地址 0x3000–0x31BF 范围内的数据。

如图 6-14-4 所示，表格数据区域显示内容由单元格数据定义，可以是数字，文本，图标或日期数据等，具体显示内容请查看表 6-15-1。每个单元格字长度 8，行数目 16，列数目 2，总字长度为 $16*2*8=256$ 。

起始地址为 0x5000。那么表格显示变量地址 0x5000–0x50FF 范围内的数据。

特别说明：当“单元格数据“字长度”设置为 0 时，表示按列设置单元格的数据长度，表格数据存储位置相应后延（列数/2）向上取整个字地址。例如，表格数据起始地址为 0x1000，列数为 7，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。

9. 控制地址 ADDR：翻页控制以及返回当前选中单元格行列坐标。

ADDR+0：翻页控制。1:下一页;2:上一页;3:右一页;4:左一页。

ADDR+1：当前选中单元格行地址。0xFFFF 表示无效。

ADDR+2：当前选中单元格列地址。0xFFFF 表示无效。

仅数据区域有效，点击“标题”行或“序号”列时不会触发选中单元格。

显示“标题”行	<input checked="" type="checkbox"/>
标签名称	无
语言索引的变量存储地址	0000
显示“序号”列	<input checked="" type="checkbox"/>
固定显示数据首列	<input checked="" type="checkbox"/>
固定显示首行	<input checked="" type="checkbox"/>
显示水平网格线	<input checked="" type="checkbox"/>
显示垂直网格线	<input checked="" type="checkbox"/>
显示边框线	<input checked="" type="checkbox"/>
奇偶行异色	<input checked="" type="checkbox"/>
偶数行背景颜色	<input type="color" value="182; 190; 211"/>
支持编辑表格内容	<input checked="" type="checkbox"/>
显示选中行背景色	<input checked="" type="checkbox"/>
显示选中单元格背景色	<input checked="" type="checkbox"/>
支持数据自动上传	<input checked="" type="checkbox"/>
控制地址(0x)	0000
数字键盘所在页面ID	无
中英文键盘所在页面ID	无
首列列宽	100
第二列列宽	100
第三列列宽	100
第四列列宽	100
第五列列宽	100
第六列列宽	100
第七列列宽	100
第八列列宽	100
行高	60

图6-14-5 数据表格控件属性1

文本颜色	<input type="color" value="0; 0; 0"/>
背景颜色	<input type="color" value="255; 255; 255"/>
网格颜色	<input type="color" value="160; 162; 174"/>
固定首行文本颜色	<input type="color" value="255; 255; 255"/>
固定首行背景颜色	<input type="color" value="85; 107; 170"/>
选中行背景色	<input type="color" value="0; 0; 255"/>
选中单元格背景色	<input type="color" value="192; 0; 0"/>
滚动条宽度	15
滚动条颜色	<input type="color" value="128; 128; 128"/>
滚动条到窗口边缘的距离	15
滚动条最小长度	50
FONT0 ID	0
FONT1 ID	1
Y方向点阵数	32
FONT1编码方式	0x06=UTF8
字符间距自动调整	<input type="checkbox"/>
水平间隔	0
行间距	0
描述指针(0x)	FFFF

图6-14-6 数据表格控件属性2

10. 数据自动上传：点击表格后支持上传当前选中单元格行列坐标。启用“支持编辑表格内容”时，录入数值、文本后上传对应地址录入后的数据。“变量图标”点击后上传循环增量调节后的数据。

11. 列宽及行高。最多支持定义前 8 列的宽度。如果大于 8 列，后续列宽与第 8 列宽度一致。所有行的高度均保持一致。

12. 数据模式为单元格自定义数据时，每个单元显示内容由单元格数据决定。

表6-15-1 单元格自定义数据格式说明。



第一个字高字节 (数据类型 Mode)	第一个字低字节	第二个字	第三-N 个字	备注
0x00/0x30 整数(2 字节) -32768 到 32767				0x00: 支持录入 0x30: 不支持录入
0x01/0x31 长整数(4 字节) -2147483648 到 2147483647				0x01: 支持录入 0x31: 不支持录入
0x02/0x32 变量高字节 0 到 255	显示格式。 高 4bit 表示整数 位数， 低 4bit 表示小数 位数。			0x02: 支持录入 0x32: 不支持录入
0x03/0x33 变量低字节 0 到 255				0x03: 支持录入 0x33: 不支持录入
0x04/0x34 超长整数(8 字节) -9223372036854775808 到 9223372036854775807				0x04: 支持录入 0x34: 不支持录入
0x05/0x35 无符号整数(2 字节) 0 到 65535		单元格文字 颜色。	具体数据。	0x05: 支持录入 0x35: 不支持录入
0x06/0x36 无符号长整数(4 字节) 0 到 4294967295				0x06: 支持录入 0x36: 不支持录入
0x10 时间格式 1 12:34:56 BCD 码串				不支持录入
0x11 时间格式 2 12-34-56 BCD 码串	BCD 码串的字节 长度。			不支持录入
0x12 时间格式 3 YYYY-MM-DD HH:MM:SS BCD 码串				不支持录入
0xFE/FF 文本	-			0xFF: 支持录入; 0xFE: 不支持录入; 文本最大长度=单元

				格数据长度-2。
0x20 图标	图标库 ID。	图标 ID。	-	不支持调节
0x21 变量图标		当前变量值。	VP+2: 变量下限值; VP+3: 变量上限值; VP+4: 变量下限对应的图标 ID。	点击单次加1增量循环调节, 调节数据更新到当前变量值。
0x22 变量图标+弹出页面		当前变量值。	VP+2: 变量下限值; VP+3: 变量上限值; VP+4: 变量下限对应的图标 ID; VP+5: 弹出页面 ID。	实现弹出菜单功能。 弹出页面内的基本按钮, 返回值更新到当前变量值。

注意:

表格所述录入及调节功能需要勾选“支持编辑表格内容”时有效;

录入功能需要配合键盘控件使用,“数字键盘”录入数字,“中英文键盘”录入文本。

更多个性化的自定义单元格格式支持定制,包括但不限于以下属性:字体,背景色,对齐方式,间距,边距,范围限制等。

数据表格控件示例及说明

如图 6-14-7 所示,表格数据区域均显示数字,小数位数为 2。支持显示标题行和序号列。显示水平和垂直滚动条。页面滑动时,标题行和序号列固定显示。标题行背景色,奇偶行异色,背景色三种颜色使得表格层次分明。

序号	长度	宽度	高度	数据4	数据5
1	1.23	54.01	90.02	95.00	75.07
2	71.15	44.35	98.43	4.81	98.83
3	71.73	38.12	17.06	4.59	8.76
	62.44	54.70	42.42	46.44	56.07

图 6-14-7 数据表格控件批量数值显示示例

如图 6-14-8 所示,表格数据区域均显示文本,支持显示汉字及 ASCII 字符。支持显示标题行和序号列。显示水平滚动条,支持左右滑动以浏览更多数据,滑动时,标题行和序号列固定显示。本示例适合设备信息显示等应用场景。

序号	名称	型号	编号
1	笔记本电脑	ThinkPad X1	XZ-20231202
2	投影仪	EPSON CB-2105	SC-20230305
3	打印机	TX-5410D	SC-20230521
4	路由器	AX3000	XZ-20230607

图 6-14-8 数据表格控件批量文本显示示例

如图 6-14-9 所示,本示例显示 wifi 列表及对应信号强度,表格数据模式为“单元格自定义数据”,数



据区域显示图标和文本。支持显示标题行。显示垂直滚动条，支持上下滑动以浏览更多数据。滑动时，标题行固定显示。隐藏垂直网格线，两列数据显示更加整体，没有割裂感。本示例中的表格内容不需要编辑，建议取消勾选“支持编辑表格内容”。本示例需要显示当前选中行，勾选“选中行背景色”以突出选中行。选中行地址自动更新到控制地址+1，获取该地址数据以执行进一步操作。

信号强度	名称
	CMCC
	ChinaUnicom
	TP-LINK_XXXX
	TP-LINK_XXXX_5G
	MECLOUD_VVVV

图 6-14-9 数据表格控件 wifi 列表示例

如图 6-14-10 所示，本示例显示多个功能选项，支持多选，表格数据模式为“单元格自定义数据”，数据区域显示文本和变量图标。显示垂直滚动条，支持上下滑动以浏览更多数据。隐藏垂直网格线，两列数据显示更加整体，没有割裂感。本示例中的图标支持修改以显示对应选项是否选中，勾选“支持编辑表格内容”属性，点击图标自动切换选中与未选中状态。

功能1	
功能2	
功能3	
功能4	
功能5	

图 6-14-10 数据表格控件功能设置示例

如图 6-14-11 所示，本示例显示配方数据管理，可修改配方名称和参数，表格数据模式为“单元格自定义数据”，数据区域显示文本和数值。本示例中，配方名称列显示文本数据，所需字长度 16。参数列显示数值，每列字长度 4。“单元格数据字长度”设置为 0，表示按列设置单元格数据字长度，各列数据字长度通过表格数据起始地址的前 N 个字节设置。这样做有利于提高数据密度，减少变量地址浪费。

如图 6-14-12 所示，支持通过弹出页面的键盘控件录入参数数据。支持弹出位置跟随点击位置。突出显示当前选中行和选中单元格，对当前录入位置有明显的提示作用，减少录入时的误操作。

序号	配方名称	参数1	参数2	参数3	参数4	参数5
1	配方1	41	467	334	500	169
2	配方2	962	464	705	145	281
3	配方3	995	942	827	436	391
4	配方4	292	382	421	716	718
5	配方5	771	538	869	912	667
6	配方6	703	811	322	333	673

6-14-11 数据表格控件配方管理示例

序号	配方名称	参数1	参数2	参数3	参数4	参数5
1	配方1	41	467	334	500	169
2	配方2	962	464	705	145	281
3	配方3	995	942	827	436	391
4	配方4	292	382	421	716	718
5	配方5	771	538	869	912	667
6	配方6	703	811	322	333	673

6-14-12 数据表格控件配方管理参数录入示例

6.15 变量掉电存储

1.工作原理

串口屏自动监测变量存储器的某个指定区域，当该区域中任何数据有变化时，串口屏就会自动将变量数据写入数据库中保存下来。当重新上电后，保存在数据库中的数据，自动恢复到变量存储器相应区域中，从而实现了变量的掉电存储功能。该功能可用于掉电存储配置参数、设备信息、登录信息、密码等参数。配合“多功能按钮”控件工作模式“密码切换页面”功能，可实现密码登录、密码修改与存储功能。

该功能仅在 NorFlash(8M/16M/32M 字节)上支持，NandFlash(128M 字节)不支持该功能。

2.设置方法

在 VGUS2025 组态软件中，点击菜单栏上“工具->屏参配置”，弹出屏参配置窗口，设置掉电存储功能如图所示。通过变量地址和变量字长度，指定一个需要掉电保存的变量存储器区域。通过数据库块编号和数据库块个数，指定连续的两个以上的数据库块用于掉电保存。



图 6-15-1 掉电存储设置

3.功能说明

当指定变量地址范围内的数据修改后自动更新到用户数据库(掉电存储)，最大更新频率不超过 1s。再次上电后，保存在用户数据库中的数据自动更新到变量存储器对应单元里。

(1)最多可以保存 2016 字(4032 字节)的变量数据，通过变量起始地址和字长度指定需要掉电保存的变量数据位置。

(2)串口屏共有 64K 字节用户数据库，分为 16 个块(编号 0-15)，每个块 2K 字(4K 字节)。通过数据库起始块编号和块个数指定数据保存位置。可设置多个块，至少设置 2 个块，最多允许设置 8 个块。对于存储较频繁的场景，可适当增加块数量，如设置为 5 块，有利于提高存储器寿命和掉电存储可靠性。

设置好以上内容后，串口屏就自动对指定变量存储器区域做掉电存储。在设计 UI 界面时，把需要掉电存储的变量地址指定到该区域中就可以了。

6.16 矢量字库

6.16.1 概述

VUGS 串口屏全面支持点阵字库应用，用户在界面设计中需要哪种字体、字号的文字，只需通过字库生成工具精准生成对应字库文件，即可便捷添加至界面中。点阵字库凭借结构简单、资源占用低等优势，在串口屏行业历经技术迭代仍保持广泛应用，堪称“历久弥新”的实用技术方案。

作为配套支持，中显科技提供的点阵字库生成工具功能完善：不仅可灵活提取不同字体（如宋体、黑体等）、不同字号的点阵数据，还支持抗锯齿处理以优化文字边缘显示效果，同时具备字库裁剪功能（按需保留所需字符，剔除冗余数据），有效减少字库文件大小，进一步适配串口屏的存储限制。

然而，随着显示应用场景的不断升级，点阵字库的局限性日益凸显，逐渐难以满足复杂需求：

1.多字号管理困境

在精细化界面设计中，标题、内容、标注等元素往往需要多种字号搭配。但点阵字库的本质是“一字号一文件”——每个字号对应独立的点阵数据，无法通过缩放复用。这直接导致字库文件数量随字号增加而成倍增长，不仅占用大量存储资源，更增加了开发阶段的文件管理和后期维护的复杂度。

2.多语言适配压力

伴随设备出海需求的爆发，多语言支持已成为刚需。从英语、西班牙语到阿拉伯语、印地语，不同语言的字符集差异显著，而每种语言又需适配不同字号以保证界面整体显示效果。若依赖点阵字库，需为“语言 × 字号”的每一种组合单独生成文件，导致点阵字库文件数据剧增，存储空间的限制也日益凸显。

3.复杂文字显示瓶颈

对于泰语、老挝语等具有特殊构形的文字，点阵字库更是力不从心。这类文字以“字符簇”为基本单位，辅音、元音、声调符号需在上下左右多维度组合，结构层次复杂。传统点阵字库的固定像素网格难以精准分配各符号的空间，极易出现笔画重叠、符号错位或缺失，导致文字无法识别，严重影响多语言场景的实用性。

针对这些痛点，VTc/VTb 系列串口屏全面引入矢量字库支持，从根源上解决文字显示难题：矢量字库基于数学曲线描述字符形态，可无损缩放至任意字号，一个字库即可覆盖全场景字号需求，大幅减少文件数量；同时，其对多语言字符集的兼容性更强，无需为每种语言单独适配，显著降低存储占用；更重要的是，矢量渲染能精准解析泰语等复杂文字的构形逻辑，确保字符簇各元素的位置和形态完整呈现。这一升级不仅简化了开发流程，更让设备在复杂界面设计、多语言出海等场景中具备了更灵活、更可靠的文字显示能力。

6.16.2 TTF 矢量字库适配方法

TTF 矢量字库适配方法与注意事项：

1. 仅 VTc/VTb 系列 20251020 及以后固件版本支持。
2. 文件命名。必须以阿拉伯数字开头，编号范围 1-255。如 20.ttf、21-fangsong.ttf。
3. 字库导入。在文件录入树中，右键“字库文件”点击“添加”对应的文件即可导入到工程。
4. 字库引用。通过各类控件的“FONT0 ID”或“FONT1 ID”属性，选择对应的字库文件即可完成该字库的引用。
5. 字体大小。通过各类控件的“Y 方向点阵数”属性设置所需显示的字体大小。



6. 特别注意：

- 6.1 对于数字显示类控件，通过“FONT0 ID”引用字库，需保证字库中有 ASCII 编码字符。
- 6.2 对于文本类控件，使用矢量字库时“FONT0 ID”或“FONT1 ID”仅需设置其中一个即可，如果一个设置为点阵字库，一个设置为矢量字库，那么使用矢量字库。如果两个均设置为矢量字库，那么仅使用“FONT1 ID”对应的矢量字库。
- 6.3 对于文本类控件，使用矢量字库时“FONT1 编码方式”属性仅支持 UNICODE、UTF8 以及 UTF8-2 这三种编码方式，否则显示异常。
- 6.4 矢量字库已包含每个字符独有的宽度信息。使用矢量字库时各类控件的“字符宽度自动调整”属性设置无效。

6.16.3 TTF 矢量字库文件裁剪方法

由于串口屏的内存十分有限，TTF 矢量字库文件的裁剪（即“字库瘦身”）十分重要，核心目标是保留必要字符、剔除冗余字形数据，在保证显示需求的前提下大幅缩减文件大小。例如，一个包含全量汉字的 TTF 文件约 5-10MB，仅保留 3500 个常用字可压缩至 1-2MB，若再叠加英文和数字，文件大小可控制在 1MB 以内，显著适配串口屏的存储限制。

常用的矢量字库裁剪工具有 FontCreator、TTFSUBSET 等，中显科技针对串口屏应用，也即将发布定制化字库裁剪工具操作更简化。

通过合理的裁剪方法，TTF 矢量字库既能保留“任意缩放、多语言兼容”的优势，又能适配资源受限场景，成为解决点阵字库局限性的理想方案。



6.17 变量存储器 JPG/PNG 图标显示

通过“基本图形”控件的 0x0021 指令显示数据源位于变量存储器的 JPG/PNG 格式图片。用于实时图片的显示。**注意：该功能仅 VTb 系列 20251020 及以后固件版本支持。**

采用压缩的 JPG/PNG 格式图片，大幅减小文件大小，减少图片数据传输时间，能够在有限的变量存储器中暂存更多图片或暂存更大分辨率的图片。图片文件最大支持 256KB。

实际应用中，图片数据量小于 128KB 时，可以通过 0x82 指令写入到变量存储器的前 128K 字节。如果图片数据大于 128K 或需要暂存多个图片但是图片总量大于 128K 时，需要使用 0x86 指令。

变量存储器双字读写指令 0x86、0x87 说明：

VTb 系列变量存储器容量大小支持扩展到 64K 双字（256K Byte），按照双字寻址操作，地址为 0x0000-0xFFFF，每个地址 4 字节，可寻址 256K 字节。（仅 VTb 系列 20251020 及以后固件版本支持）

注意：目前双字地址仅支持配合“基本图形”控件 0x0021 图片显示指令使用。

起始地址 对于变量存储器读写指令（0x86、0x87）是双字地址，地址范围为 0x0000~0xffff。

数据长度 (N) 以双字（4 字节）为单位，表示连续读写变量存储器双字单元个数。

数据内容 是 4N 字节，因为每个变量存储器双字单元包含 4 个字节。

支持顺序读写多个变量存储器双字单元。

表 6-17-1 串口发送 0x86 指令帧结构（写变量存储器双字单元）

定义	指令帧头	指令长度	指令	起始地址		数据内容
长度（字节）	2	1	1(0x86)	2		4N

举例 1： 变量存储器双字地址 0x0003 写入 0x00、0x01、0x02、0x03

发送：0xA5 0x5A 0x07 0x86 0x00 0x03 0x00 0x01 0x02 0x03

举例 2： 连续写入多个变量地址双字单元，变量存储双字地址 0x0003 单元写入 0x00、0x01、0x02、0x03，0x0004 单元写入 0x04、0x05、0x06、0x07

发送：0xA5 0x5A 0x0B 0x86 0x00 0x03 0x00 0x01 0x02 0x03 0x04 0x05 0x06 0x07

表 6-17-2 串口发送 0x87 指令帧结构（读变量存储器双字单元发送）

定义	指令帧头	指令长度	指令	起始地址	数据长度
长度（字节）	2	1	1(0x87)	2	1(N)

表 6-17-3 串口接收 0x87 指令帧结构（读变量存储器接收，串口屏发送的）

定义	指令帧头	指令长度	指令	起始地址	数据长度	数据内容
长度（字节）	2	1	1(0x87)	2	1(N)	4N

举例： 连续读取变量存储器双字地址 0x0003 和 0x0004 单元

发送：0xA5 0x5A 0x04 0x87 0x00 0x03 0x02

返回：0xA5 0x5A 0x0C 0x87 0x00 0x03 0x02 0x00 0x01 0x02 0x03 0x04 0x05 0x06 0x07

表 6-17-4 变量存储器双字读写指令详解

功能	指令	数据	说明
访问变量	0x86	下发：变量双字地址 (0x0000-0xFFFF) + 写入的变量数据	指定双字地址开始写入数据（双字数据）到变量存储区



存储器 接口	0x87	下发: 变量双字地址 (0x0000-0xFFFF) + 读取变 量数据双字长度 (0x00-0x3E)	从变量存储区指定双字地址开始读入指定双字长度的 数据
		应答: 变量存储器双字地址+变量数据双字长度+ 读取的变量数据	双字读数据存储器的 VGUS 应答

JPG/PNG 图片显示指令数据包格式定义:

表 6-17-5 JPG/PNG 图片显示指令数据包格式定义

图形 指令	绘图 功能	偏移量	长度	定义	说明
0x0021	JPG/PNG 图 片 显 示	0x00	2	(x, y)	JPG/PNG 图片显示左上角坐标。
		0x02	1	mark	功能标记。写入以下特定值配置地址模式并开启显示。 0x5AA1 表示图片数据字地址寻址。 0x5AA2 表示图片数据双字地址寻址。 图片数据校验或解析异常时, 该地址写入异常值。
		0x03	1	VP	图片数据变量存储地址。
		0x04	1	len	图片数据长度。字地址寻址时表示字长度。双字地址寻址时表示双 字长度。
		0x05	1	checksum	图片数据校验和低16位。

了解更多“基本图形”控件相关用法请查看 4.3.2 基本图形（绘图）章节。

使用说明及注意事项:

1. 图片类型为 JPG 还是 PNG 内部自动判断。
2. 图片数据长度说明。字地址寻址时允许最大 128K 字节。双字地址寻址时允许最大 256K 字节。(双字地址相关) 设置时需“向上取整”并转换为字长度或双字长度。并且使用 0 填充多余数据, 否则影响校验和的计算值。
3. 图片分辨率限制说明。JPG 格式允许分辨率宽高乘积最大 393216。PNG 格式允许分辨率宽高乘积最大 262144。
4. 图片更新流程: 清除标记 -> 更新数据 -> 写入配置信息 (显示位置, 寻址方式, 寻址地址, 数据长度以及校验和)。(如果当前显示数据区域与更新数据区域不重叠, 可不清除标记, 数据更新完成后写入配置数据完成“图片刷新”)
5. 串口屏解析及结果反馈。“mark”写入 0x5AA1 或 0x5AA2 后, 按照配置信息校验数据后解析图片并显示。如果失败, 该地址写入异常值: 0xAA01 表示数据校验失败、0xAA02 表示图片解码异常、0xAA03 表示图片分辨率超过允许范围。
6. 如果图片数据字地址在 0xff01-0xff0e 范围内, 建议关闭视频播放寄存器映射功能, 否则会破坏数据完整性, 影响数据校验和导致图片显示异常。通过设置 miscellaneous 选项位功能.B 位可关闭视频播放寄存器映射功能。



7 描述指针

对于简单应用场合，用户无需了解本章知识。

SDWn 系列不支持描述指针功能

7.1 描述指针介绍

VGUS 屏采用变量驱动，所有显示的字符、图标等都定义为一个变量，并分配变量存储地址、定义显示格式，然后生成配置文件并下载到显示终端上。在需要刷新显示时，用户仅需将变量内容和变量存储地址通过串口对应发送到显示终端，显示终端会自动按照定义好的显示格式显示。

对于每个变量来讲，其显示格式是固定的，是由下载到 VGUS 屏中的配置文件定义的。当用户需要临时修改变量显示格式时，可以通过本章介绍的描述指针实现。

用户在定义每个变量的时候，需要设置是否启用描述指针，如图 6-1 所示。



图 7-1 设置数据变量显示属性

如图描述指针一栏填写 FFFF 表示当前变量禁止描述指针功能。如果填写的是其它数据（0000–FFFE 中任意一个）表示当前变量启用描述指针，并且填写的数据用来指定变量存储器地址，以该地址为起始单元，在变量存储器内连续开辟了一块空间（不同的变量类型，块长度以及数据格式也都不同，每种变量本章都有详细表格定义了对应数据格式），用于存储变量的显示格式。用户可以通过变量存储器地址，利用 0x82 指令去动态修改变量显示格式。

不同控件属性长度不一致，请查看各自控件的详细说明，为了避免出现不同控件的描述指针重叠，建议在当前控件属性基础上，预留至少 16word 地址。确保不同控件的描述指针地址不会出现重叠。

仅部分控件可以通过描述指针来读写，触摸控件和部分高级控件不支持描述指针。描述指针地址范围 0x0000–0xFFFF，与用户变量共用变量存储器区，因此用户定义的变量存储地址不能与描述指针的地址重叠。

7.2 描述指针应用举例

本节以改变一个数据变量的显示颜色为例，介绍描述指针的简单应用。

1、新建工程添加图片后，在界面上添加一个数据变量显示控件，设置描述指针为4000，变量存储地址为0000。如图 6-2-1 所示；

名称定义	数据变量显示
描述指针(0x)	4000
变量存储地址(0x)	0000
文本颜色	119; 119; 119
字库位置	0
字体大小	16
对齐方式	左对齐
变量类型	int (2Byte)
整数位数	4
小数位数	0
显示单位	
初始值	1234

图 7-2-1 添加数据变量

2、查看关于数据变量描述指针的说明；

表 7-2-1 数据变量描述指针数据格式

地址偏移量 (双字节)	定义	数据长度 (字节)	说明
0x00	*VP	2	变量指针
0x01	X, Y	4	起始显示位置，显示字符串左上角坐标
0x03	Color	2	显示颜色
0x04:H	Lib_ID	1	ASCII字库位置
0x04:L	字体大小	1	字符X方向的点阵数
0x05:H	对齐方式	1	0x00=左对齐，0x01=右对齐，0x02=居中
0x05:L	整数位数	1	显示的整数位数（整数位数和小数位数之和不能超过20）
0x06:H	小数位数	1	显示的小数位数（整数位数和小数位数之和不能超过20）
0x06:L	变量数据 类型	1	0x00=整数(两字节): -32768到32767 0x01=长整数(4字节): -2147483648到2147483647 0x02=VP*高字节: 0到255 0x03=VP*低字节: 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65536 0x06=无符号长整数(4字节): 0到4294967295
0x07:H	Len_unit	1	变量单位显示长度，0x00表示不显示单位
0x07:L	String_unit	10	单位字符串，ASCII编码

从图中可以看到，颜色对应的偏移量为03，那么颜色属性对应的变量存储器地址为0x4000+03=0x4003。通过0x82指令修改0x4003单元内的数据即改变了数据变量的显示颜色。

例如串口发送: A5 5A 05 82 40 03 F8 00

那么就设置数据变量显示为红色 (0xF800 为红色)。

本例中使用按钮键值返回控件, 按下按钮时, 直接将键值作为对应颜色值写入变量存储器 0x4003 单元。按键返回控件属性设置如图 7-2-2 所示。



图 7-2-2 按键返回控件属性设置

7.3 描述指针汇总

表 7-3 描述指针汇总表

编号	名称
表 7-3-1	变量图标
表 7-3-2	动画图标
表 7-3-3	滑动刻度
表 7-3-4	艺术字变量
表 7-3-5	动画图片
表 7-3-6	旋转图标
表 7-3-7	位变量图标
表 7-3-8	数据变量
表 7-3-9	文本变量
表 7-3-10	文本时钟
表 7-3-11	表盘时钟
表 7-3-12	时钟变量
表 7-3-13	滚动文本
表 7-3-14	实时曲线
表 7-3-15	基本图形
表 7-3-16	列表显示
表 7-3-17	二维码
表 7-3-18	视频控件
表 7-3-19	摄像头控件
表 7-3-20	进度条显示
表 7-3-21	圆形进度条显示
表 7-3-22	自适应二维码
表 7-3-23	文字标签
表 7-3-24	下拉列表
表 7-3-25	滚轮
表 7-3-26	开关
表 7-3-27	数据记录表格
表 7-3-28	数据曲线
表 7-3-29	数值显示
表 7-3-30	多功能按钮
表 7-3-31	数据表格

表 7-3-1 变量图标描述指针定义

地址偏移量 (双字节)	定义	数据长度 (字节)	说明
0x00	*VP	2	变量指针, 变量为整数格式。
0x01	X, Y	4	变量显示位置, 图标左上角坐标位置。
0x03	V_Min	2	变量下限, 越界不显示。
0x04	V_Max	2	变量上限, 越界不显示。
0x05	Icon_Min	2	V_Min对应的图标ID。
0x06	Icon_Max	1	V_Max对应的图标ID。
0x07:H	Icon_Lib	1	图标库存储位置。
0x07:L	Mode	1	ICON显示模式。0x00=透明(不显示背景), 其他=显示图标背景。

表 7-3-2 动画图标描述指针定义

地址偏移量 (双字节)	定义	数据长度 (字节)	说明
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
0x01	x, y	4	变量显示位置, 图标左上角坐标位置。
0x03	0x0000	2	固定
0x04	V_Stop	2	变量为该值时固定。
0x05	V_Start	2	变量为该值时自动显示动画图标。
0x06	Icon_Stop	2	变量为V_Stop时固定显示的图标。
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。
0x08	Icon_End	2	
0x09:H	Icon_Lib	1	图标库存储位置。
0x09:L	Mode	1	ICON显示模式。0x00=透明

表 7-3-3 滑动刻度描述指针定义

地址偏移量 (双字节)	定义	数据长度 (字节)	说明
0x00	*VP	2	变量指针, 变量格式由VP_DATA_Mode来决定。
0x01	V_begain	2	对应起始变量刻度值。
0x02	V_end	2	对应终止变量刻度值。
0x03	X_begain	2	起始刻度坐标(纵向为Y坐标)。
0x04	X_end	2	终止刻度坐标(纵向为Y坐标)。
0x05	Icon_ID	2	滑动刻度块的图标ID。
0x06	Y	2	刻度指示图标显示的Y坐标值。
0x07:H	X_adj	1	刻度指示图标显示的X坐标前移偏移量。
0x07:L	Mode	1	刻度模式。0x00=横向刻度条, 其他=纵向刻度条。
0x08:H	Icon_Lib	1	图标库存储位置。
0x08:L	Icon_mode	1	ICON显示模式。0x00=透明(不显示背景), 其他=显示图标背景。
0x09:H	VP_DATA_Mode	1	0x00=*VP指向一个整型变量; 0x01=*VP指向一个整型变量高字节数据; 0x02=*VP指向一个整型变量低字节数据。

表 7-3-4 艺术字变量



地址偏移量 (双字节)	定义	数据长度 (字节)	说明
0x00	*VP	2	变量指针
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
0x03	Icon0	2	0对应的ICON_ID, 排列顺序为0123456789-.
0x04:H	Icon_Lib	1	图标库存储位置。
0x04:L	Icon_Mode	1	ICON显示模式: 0x00=透明 (不显示背景) 其他=显示图标背景
0x05:H	整数位数	1	显示的整数位数
0x05:L	小数位数	1	显示的小数位数
0x06:H	数据变量类型	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
0x06:L	对齐模式	1	0x00=左对齐, 0x01=右对齐

表 7-3-5 动画图片描述指针定义

地址偏移量 (双字节)	定义	数据长度 (字节)	说明
0x00	0x0000	2	固定
0x01	Pic_begain	2	起始图标位置
0x02	Pic_end	2	终止图标位置
0x03:H	Frame_Time	1	一幅图片显示的时间, x10ms

表 7-3-6 旋转图标描述指针定义

地址偏移量 (双字节)	定义	数据长度 (字节)	说明
0x00	*VP	2	变量指针, 变量格式由Mode决定。
0x01	Icon_ID	2	指定图标ID。
0x02	Icon_Xc	2	Icon图标上的指定旋转中心位置X坐标。
0x03	Icon_Yc	2	Icon图标上的指定旋转中心位置Y坐标。
0x04	Xc	2	Icon图标显示到当前屏幕的旋转中心位置X坐标。
0x05	Yc	2	Icon图标显示到当前屏幕的旋转中心位置Y坐标。
0x06	V_Begain	2	对应起始旋转角度的变量值, 整型, 越界不显示。
0x07	V_End	2	对应终止旋转角度的变量值, 整型, 越界不显示。
0x08	AL_Begain	2	起始旋转角度0-720 (0x0000-0x02D0), 单位0.5度。



0x09	AL_End	2	转动角度= (终止角度-起始角度) / (终止值-起始值) *输入值; 终止旋转角度0-720 (0x0000-0x02D0), 单位0.5度。
0x0A: H	VP_Mode	1	0x00=*VP指向一个整型变量; 0x01=*VP指向一个整型变量高字节数据; 0x02=*VP指向一个整型变量低字节数据。
0x0A: L	Lib_ID	1	图标库存储位置。
0x0B	Mode	1	ICON显示模式。0x00=透明 (不显示背景), 其他=显示图标背景。

表 7-3-7 位变量图标

地址偏移量 (双字节)	定义	数据长度 (字节)	说明																													
0x00	*VP	2	位变量指针, 字变量																													
0x01	*VP_AUX		辅助变量指针, 双字, 用户禁用																													
0x02	Act_Bit_Set	2	为1的bit值说明*VP对应位置需要显示																													
0x03: H	Display_Mode	1	<p>显示模式定义:</p> <table border="1"> <thead> <tr> <th rowspan="2">Display_Mode</th> <th colspan="2">Bit值</th> </tr> <tr> <th>0</th> <th>1</th> </tr> </thead> <tbody> <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> </tbody> </table>	Display_Mode	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	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																														
0x03: L	Move_mode	1	<p>位图图标排列方式:</p> <p>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</p>																													
0x04:H	Icon_Mode	1	ICON显示模式: 0x00=透明 (不显示背景) 其他=显示图标背景																													
0x04:L	Icon_Lib	1	图标库存储位置																													
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																													
0x06	ICONOE	2	显示动画模式, bit 0 图标动画结束 ID 位置																													
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																													



0x08	ICON1E	2	显示动画模式, bit 1 图标动画结束 ID 位置
0x09	(x, y)	4	起始变量显示位置, 图标左上角坐标
0x0B	Dis_MOV	2	下一个图标坐标移动的间隔

表 7-3-8 数据变量描述指针定义

地址偏移量 (双字节)	定义	数据长度 (字节)	说明
0x00	*VP	2	变量指针
0x01	X, Y	4	起始显示位置, 显示字符串左上角坐标
0x03	Color	2	显示颜色
0x04:H	Lib_ID	1	ASCII字库位置
0x04:L	字体大小	1	字符X方向的点阵数
0x05:H	对齐方式	1	0x00=左对齐, 0x01=右对齐, 0x02=居中
0x05:L	整数位数	1	显示的整数位数 (整数位数和小数位数之和不能超过20)
0x06:H	小数位数	1	显示的小数位数 (整数位数和小数位数之和不能超过20)
0x06:L	变量数据 类型	1	0x00=整数(两字节): -32768到32767 0x01=长整数(4字节): -2147483648到2147483647 0x02=VP*高字节: 0到255 0x03=VP*低字节: 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65536 0x06=无符号长整数(4字节): 0到4294967295
0x07:H	Len_unit	1	变量单位显示长度, 0x00表示不显示单位
0x07:L	String_unit	10	单位字符串, ASCII编码

表 7-3-9 文本变量描述指针定义

地址偏移量 (双字节)	定义	数据长度 (字节)	说明
0x00	*VP	2	文本指针
0x01	X, Y	4	起始显示位置, 显示字符串左上角坐标
0x03	Color	2	显示文本颜色
0x04	Xs, Ys, Xe, Ye	8	文本框
0x08	Text_Length	2	显示字数量, 遇到0xFFFF数据或者显示到文本框尾则不再显示
0x09:H	Font0_ID	1	编码方式为0x01-0x04时ASCII字库位置
0x09:L	Font1_ID	1	编码方式为0x00、0x05, 以及0x01-0x04的非ASCII字符使用的字库位置
0x0A:H	Font_X_Dots	1	字体X方向点阵数, 0x01-0x04模式时ASCII字符X按照Y/2计算



0x0A:L	Font_Y_Dots	1	字体Y方向点阵数, 字库中Y方向的点阵数必须为偶数
0x0B:H	Encode_Mode	1	.7定义了文本显示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, 字符宽度为固定设定的点阵数 .6-.0定义了文本的编码方式: 0x00=8bit编码, 0x01=GB2312内码, 0x02=GBK, 0x03=BIG5, 0x04=SJIS, 0x05=UNICODE
0x0B:L	HOR_Dis	1	字符水平间距
0x0C:H	VER_Dis	1	字符垂直间距
0x0C:L	未定义	1	写0x00

表 7-3-10 文本时钟描述指针定义

地址偏移量 (双字节)	定义	数据长度 (字节)	说明
0x00:H	未定义	1	
0x00:L	type	1	0x00:文本RTC, 时钟源为串口屏 0x02:文本RTC, 时钟源为用户单片机
0x01	X, Y	4	起始显示位置, 显示字符串左上角坐标
0x03	Color	2	字体颜色
0x04:H	Lib_ID	1	字库位置
0x04:L	Font_Size	1	X方向的点阵数
0x05	String_Code	Max14	编码字符串, 使用RTC编码表和ASCII字符组成 假设当前时间2014-05-01 12:00:00 星期三 Y_M_D H:Q:S 0x00将显示为 2014-05-01 12:00:00 M_D W H:Q 0x00将显示为 05-01 WEN 12:00
0x0c	*VP	2	时钟源为用户单片机时, 时间信息的变量指针。 时间信息在变量地址的存储方式: 字地址, 范围0000-FFFF。字长度为4。 YY:MM:DD:WW:HH:MM:SS, 年: 月: 日: 星期: 时: 分: 秒 格式为BCD码格式。

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 7-3-11 表盘时钟描述指针定义

地址偏移量 (双字节)	定义	数据长度 (字节)	说明
0x00:H	*VP 高字节	1	
0x00:L	type	1	0x01:表盘时钟, 时钟源为串口屏 0x03:表盘时钟, 时钟源为用户单片机
0x01	X, Y	4	时钟表盘的指针中心
0x03	Icon_Hour	2	时针 ICON 的 ID, 0xFFFF 表示时针不显示。
0x04	Icon_Hour_center	4	时针 ICON 的旋转中心位置。
0x06	Icon_Minute	2	分针 ICON 的 ID, 0xFFFF 表示分针不显示
0x07	Icon_Minute_center	4	分针 ICON 的旋转中心位置。
0x09	Icon_Second	2	秒针 ICON 的 ID, 0xFFFF 表示秒针指针不显示
0x0A	Icon_Second_center	4	秒针 ICON 的旋转中心位置。
0x0C:H	ICON_Lib	1	指针图标所在的 ICON 库文件 ID。
0x0C:L	*VP 低字节	1	时钟源为用户单片机时, 时间信息的变量指针。 时间信息在变量地址的存储方式: 字地址, 范围0000-FFFF。字长度为4。 YY:MM:DD:WW:HH:MM:SS, 年: 月: 日: 星: 期: 时: 分: 秒 格式为BCD码格式。

表 7-3-12 变量时钟描述指针定义

地址偏移量 (双字节)	定义	数据长度 (字节)	说明
0x00	*VP	2	变量指针数据串首地址, 变量为BCD编码
0x01	X, Y	4	起始显示位置, 显示字符串左上角坐标
0x03	Color	2	字体颜色
0x04:H	Byte_Num	1	*VP指针高字节开始显示的数目, 0x01-0x0F
0x04:L	Lib_ID	1	字库位置, 必须是半角字库, 如果Lib_ID不为0, 则必须用8bit编码



0x05:H	Font_X	1	X方向点阵数目
0x05:L	String_Code	Max13	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编码后, 会从编码字符串中顺序的取出一个ASCII字符来间隔显示 编码字符串中特殊编码定义: 0x00: 无效, 不显示字符, 两个BCD编码连在一起 0x0D: 换行显示。即 X=Xs, Y=Y+Font_X*2

表 7-3-13 滚动文本描述指针定义

地址偏移量 (双字节)	定义	数据长度 (字节)	说明
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
0x01:H	Rolling_Mode	1	滚屏模式: 0x00表示从右向左滚屏
0x01:L	Rolling_Dis	1	滚屏间距, 每个VGUS周期文本滚动的像素点阵数
0x02:H	Adjust_Mode	1	对齐方式: 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
0x02:L	未定义	1	写 0x00
0x03	Color	2	显示文本颜色
0x04	Xs Ys Xe Ye	8	文本框
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
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
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 Y/2计算)
0x09:L	Font_Y_Dots	1	字体Y方向点阵数目
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
0x0A:L	Text_Dis	1	字符间隔
0x0B	Text_Length	2	显示字数量, 遇到0xFF或0x00提前结束。 (仅VTC系列支持)

表 7-3-14 实时曲线描述指针定义

地址偏移量 (双字节)	定义	数据长度 (字节)	说明
----------------	----	--------------	----



0x00	0x0000	2	无定义
0x01	Xs:Ys:Xe:Ye	8	曲线窗口: 左上角坐标 (Xs, Ys), 右下角坐标 (Xe, Ye)
0x05	Y_Central	2	曲线中心位置
0x06	VD_Central	2	中心曲线对应的曲线数据值
0x07	Color	2	曲线颜色
0x08	NUL_Y	2	曲线放大倍数, 单位是1/256, 0x0000-0xFFFF
0x09:H	CHANNEL	1	数据源通道: 0x01-0x07
0x09:L	Dis_HOR	1	横轴间隔: 0x01-0xFF
0x0A:H	MODE_CONFIG	1	.7-.3:保留, 写0 .2 定义了线条显示模式: 0: 抗锯齿模式 1: 普通模式 .1 定义了曲线越界处理方式: 0: 调整到边界值显示 1: 越界部分不显示 .0 定义了曲线移动方向: 0: 从右向左 1: 从左向右

注意:

MODE_CONFIG功能仅20230528以后版本固件支持。

VTc系列不支持线条显示模式设置。

表 7-3-15 基本图形描述指针定义

地址偏移量 (双字节)	定义	数据长度 (字节)	说明
0x00	*VP	2	变量数据指针
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
0x05:H	Dashed_Line_ Ent	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
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 例如, 设置 0x10 0x04 0x10 0x04 将显示虚线, 设置 0x10 0x04 0x02 0x04 将显示点划线
		13	保留, 写0x00

表 7-3-16 列表显示描述指针定义

地址偏移量 (双字节)	定义	数据长度 (字节)	说明
0x00	*VP	2	表格内容指针, 即TAB_X_Num与TAB_Y_Num数组的首地址



0x01:H	TAB_X_Num	1	列数目, 0x01-0xFF
0x01:L	TAB_Y_Num	1	行数目, 0x01-0xFF
0x02:H	TAB_X_Start	1	表格起始显示的列位置, 0x00-0xFF
0x02:L	TAB_Y_Start	1	表格起始显示的行位置, 0x00-0xFF
0x03:H	Unit_Data_Num	1	<ul style="list-style-type: none"> ➤ 0x01-0xFF所有单元格存储的数据长度相同, ➤ 0x00由*VP变量地址指针指向变量存储空间定义不同列单元格的数据长度。 (word, 字长度) <p style="color: red;">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>例如, *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>
0x03:L	Encode_Mode	1	<ul style="list-style-type: none"> .7定义了文本显示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, 字符宽度为固定设定的点阵数 .6定义了表格内容格式: .6=0 表格内容为文本格式 .6=1. 未定义 .5 定义了边框线条是否显示: .5=0 显示边框, .5=1 不显示边框 .4 未定义, 写 0。 .3-.0定义了文本的编码方式: 0x00=8bit 编码, 0x01=GB2312内码, 0x02=GBK, 0x03=BIG5, 0x04=SJIS, 0x05=UNICODE
0x04	Xs, Ys, Xe, Ye	8	表格显示区域定义, 左上角坐标, 右下角坐标, 越界不显示
0x08	Color_line	2	表格边框颜色
0x09	Color_text	2	表格文本颜色
0x0A:H	Font0_ID	1	编码方式为0x01-0x04时ASCII字库位置
0x0A:L	Font1_ID	1	编码方式为0x00、0x05, 以及0x01-0x04的非ASCII字符使用的字库位置
0x0B:H	Font_X_Dots	1	字体X方向点阵数, 0x01-0x04模式时ASCII字符X按照X/2计算
0x0B:L	Font_Y_Dots	1	字体Y方向点阵数, 字库中Y方向的点阵数必须为偶数
0x0C:H	TAB_X_Adj_Mod	1	设置TAB_X_Start不为0时, 进行表头显示控制, 0x00=首列不显示, 0x01=首列显示
0x0C:L	TAB_Y_Adj_Mod	1	设置TAB_Y_Start不为0时, 进行表头显示控制, 0x00=首行不显示, 0x01=首行显示

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

第1个字节高字节: 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, 12:34:56BCD 码串
 0x11=时间格式 2, 12-34-56BCD 码串
 0x12=时间格式 3, YYYY-MM-DD HH:MM:SS BCD 码串
 0xFF=文本格式

第1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 码串的字节长度

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

表 7-3-17 二维码描述指针定义

地址偏移量 (双字节)	定义	数据长度 (字节)	说明
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。
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 459 字节) 两种。
0x03	Unit_Pixels	2	每个二维码单元像素所占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。
0x04:H	border	1	bit7-bit1:保留, 写0。 bit0是否显示边框。 0: 显示; 1: 不显示。
0x04:L	Text_Length	1	文本字数量 (1-229)。 (仅VTc系列支持)

表 7-3-18 视频控件描述指针定义

地址偏移量 (双字节)	定义	数据长度 (字节)	说明
0x00	*VP	2	变量指针, 变量为整数格式。
0x01	X,Y	4	视频窗口左上角坐标位置。
0x03	V_Stop	2	停止值。
0x04	V_Start	2	开始值。
0x05	Video_Num	2	视频文件编号
0x06:H	Action	1	播放结束后, 是否需要页面切换, Action=1表示需要切换。
0x06:L-0x07:H	Pic_Next	2	播放结束后, 切换的页面编号。
0x07:L	Repeat	1	播放重复次数。 (最大255次循环或无限次循环)



			0表示不断循环
0x08:H	Vol	1	播放音量。

表 7-3-19 摄像头控件描述指针定义

地址偏移量 (双字节)	定义	数据长度 (字节)	说明
0x00	*VP	2	变量指针, 变量为整数格式。
0x01	Xs,Ys,Xe,Ye	8	摄像头视频窗口位置及大小。
0x05	AV_Stop	2	停止值。 向变量存储单元写入停止值时, 停止播放摄像头视频。
0x06	Channel1_Start	2	通道1开始值。 向变量存储单元写入通道1开始值时, 播放通道1的摄像头视频。
0x07	Channel2_Start	2	通道2开始值。 向变量存储单元写入通道2开始值时, 播放通道2的摄像头视频。
0x08:H	Channel1_config	1	通道1配置: bit7-bit3:保留, 写0。 bit2模式选择。 0: 模式1; 1: 模式2。 bit1-bit0信号制式。 0: 自动选择; 1: NTSC ; 2: PAL 。
0x08:L	Channel2_config	1	通道2配置: bit7-bit3:保留, 写0。 bit2模式选择。 0: 模式1; 1: 模式2。 bit1-bit0信号制式。 0: 自动选择; 1: NTSC ; 2: PAL 。
	NULL	10	0X00。

表 7-3-20 进度条显示描述指针定义

地址偏移量 (双字节)	定义	数据长度 (字节)	说明
0x00	*VP	2	变量指针, 变量格式由VP_DATA_Mode来决定。
0x01	V_begin	2	对应起始变量值。
0x02	V_end	2	对应终止变量值。
0x03	Xs, Ys	4	左上角坐标。
0x05	Xe, Ye	4	右下角坐标。
0x07:H	VP_DATA_Mode	1	0x00=*VP指向一个整型变量; 0x01=*VP指向一个整型变量高字节数据; 0x02=*VP指向一个整型变量低字节数据。
0x07:L	Mode	1	进度条类型:



			0x00=横向进度条，左侧对应开始值，右侧对应结束值； 0x01=横向进度条，右侧对应开始值，左侧对应结束值； 0x02=纵向进度条，下侧对应开始值，上侧对应结束值； 0x03=纵向进度条，上侧对应开始值，下侧对应结束值。
0x08:H	Icon_Lib	1	图标库存储位置。
0x08:L	Icon_Mode	1	背景图标显示模式： .0位：0=透明（不显示背景），1=显示图标背景。 前景图标显示模式： .1位：0=透明（不显示背景），1=显示图标背景。
0x09:H	Background_Mode	1	背景类型： 0x00=不显示； 0x01=单色背景； 0x02=切图背景； 0x03=图标背景；
0x09:L	Prospect_Mode	1	前景类型： 0x00=不显示； 0x01=单色前景； 0x02=切图前景； 0x03=图标前景；
0x0A	Background_Data	2	背景类型=0x01时，表示背景颜色数据； 背景类型=0x02时，表示背景切图所在界面； 背景类型=0x03时，表示背景图标ID；
0x0B	Prospect_Data	2	前景类型=0x01时，表示前景颜色数据； 前景类型=0x02时，表示前景切图所在界面； 前景类型=0x03时，表示前景图标ID；

表 7-3-21 圆形进度条显示描述指针定义

地址偏移量 (双字节)	定义	数据长度	说明
0x00	*VP	2	变量指针，变量格式由VP_DATA_Mode来决定。
0x01	V_begain	2	对应起始变量值。默认值0。
0x02	V_end	2	对应终止变量值。默认值100。
0x03	Xs, Ys	4	左上角坐标。
0x05	outside_radius	2	外圆半径。范围：4-400。
0x06	inner_radius	2	内圆半径。范围：0-399。
0x07:H	VP_DATA_Mode	1	0x00=*VP指向一个整型变量； 0x01=*VP指向一个整型变量高字节数据； 0x02=*VP指向一个整型变量低字节数据。
0x07:L	Mode	1	进度条类型： 进度方向： .0位：0=顺时针，1=逆时针。 是否使用边缘圆角： .1位：0=不使用，1=使用。 背景类型：.2-.3位 0=不显示； 1=单色背景； 2=切图背景； 3=图标背景；

			前景类型: .4-.5位 0=不显示; 1=单色前景; 2=切图前景; 3=图标前景; 注意: VTc系列不支持切图背景和切图前景。
0x08:H	Icon_Lib	1	图标库存储位置。
0x08:L	Icon_Mode	1	背景图标显示模式。 .0位: 0=透明(不显示背景), 1=显示图标背景。 前景图标显示模式。 .1位: 0=透明(不显示背景), 1=显示图标背景。
0x09	Background_Data	2	背景类型=1时, 表示背景颜色数据; 背景类型=2时, 表示背景切图所在界面; 背景类型=3时, 表示背景图标ID;
0x0A	Prospect_Data	2	前景类型=1时, 表示前景颜色数据; 前景类型=2时, 表示前景切图所在界面; 前景类型=3时, 表示前景图标ID;
0x0B	AL_Begain	2	起始角度。 SDWb系列: 角度范围0-720, 单位0.5°。 VTc系列: 角度范围0-360, 单位1°。
0x0C	AL_End	2	终止角度。 SDWb系列: 角度范围0-720, 单位0.5°。 VTc系列: 角度范围0-360, 单位1°。

表 7-3-22 自适应二维码描述指针定义

地址偏移量 (双字节)	定义	数据长度	说明
0x00	*VP	2	二维码数据变量存储地址。
0x01	Length	2	二维码数据字长度。范围: 1-545。 遇到0x00或0xFF提前结束。 不同纠错等级支持的最大字长度如下: L级: 字长度最大545。 M级: 字长度最大428。 Q级: 字长度最大305。 H级: 字长度最大230。
0x02	Xs, Ys	4	左上角坐标。
0x04	Width	2	二维码宽度(高度(正方形宽高一致))。
0x05	LEVEL	2	.15-.2位: 保留, 写0。 .1-.0位: 二维码纠错等级。范围: 0-3。 0: L级, 可以纠正大约7%的错误。 1: M级, 可以纠正大约15%的错误。 2: Q级, 可以纠正大约25%的错误。 3: H级, 可以纠正大约30%的错误。
0x06	BackColor	2	背景颜色。
0x07	Color	2	二维码颜色。

表 7-3-23 文字标签控件描述指针定义

地址偏移量 (双字节)	定义	数据长度	说明
0x00	Mode	2	<p>工作模式配置。</p> <p>. F-. C保留。</p> <p>. B定义了是否使用密码显示:</p> <p>0: 不使用; 1: 使用;</p> <p>. A定义了是否使用透明背景:</p> <p>0: 不使用; 1: 使用;</p> <p>. 9-. 8定义了显示模式:</p> <p>0: 多行显示; 1: 滚动显示;</p> <p>2: 循环滚动显示;</p> <p>. 7-. 6定义了垂直对齐方式:</p> <p>0: 上对齐; 1: 居中对齐; 2: 下对齐;</p> <p>. 5-. 4定义了水平对齐方式:</p> <p>0: 左对齐; 1: 居中对齐; 2: 右对齐;</p> <p>. 3定义了是否支持触摸滑动:</p> <p>0: 显示文本大于文本框时不支持滑动;</p> <p>1: 显示文本大于文本框时支持滑动;</p> <p>. 2-. 1定义了是否闪烁显示:</p> <p>0: 不闪烁; 1: 闪烁间隔0.5s;</p> <p>2: 闪烁间隔1s; 3: 闪烁间隔2s;</p> <p>. 0定义了数据来源:</p> <p>0: 文字标签; 1: 变量存储器</p>
0x01	VP	2	<p>1. 数据来源为文字标签时, 表示所用文字标签编号;</p> <p>从1开始使用。</p> <p>2. 数据来源为变量存储器时, 表示起始地址;</p>
0x02	LEN	2	<p>1. 数据来源为文字标签时, 表示语言索引的变量存储地址。可通过修改该地址数据实现多语言切换。</p> <p>2. 数据来源为变量存储器时, 表示文本的字地址长度, 最大1024;</p>
0x03	Xs:Ys:Xe:Ye	8	文本框: 左上角坐标 (Xs, Ys), 右下角坐标 (Xe, Ye)。
0x07	Txtcolor	2	文本颜色。
0x08	Bgcolor	2	背景颜色。
0x09:H	Font0_ID	1	编码方式为0x01-0x04、0x06时ASCII字库位置
0x09:L	Font1_ID	1	编码方式为0x00、0x05、0x06, 以及0x01-0x04的非ASCII字符使用的字库位置

0x0A:H	Font_Y_Dots	1	字体Y方向点阵数, 字库中Y方向的点阵数必须为偶数。
0x0A:L	Encode_Mode	1	<p>.7定义了文本显示字符间距是否自动调整:</p> <p>.7=0字符间距自动调整</p> <p>.7=1字符间距不自动调整, 字符宽度为固定设定的点阵数</p> <p>.6-.4保留:</p> <p>.3-.0定义了文本的编码方式:</p> <p>0x00=8bit 编码, 0x01=GB2312 内码, 0x02=GBK, 0x03=BIG5, 0x04=SJIS, 0x05=UNICODE, 0x06=UTF8, 0x07=UTF8-2</p> <p>注意: 数据来源为文字标签时, 固定使用UTF8编码;</p>
0x0B:H	H_Dis	1	字符间距。最大255。
0x0B:L	V_Dis	1	行间距。最大255。
0x0C:H	Rolling_Dis	1	滚动速度: 每秒移动的像素数量。最大255。
0x0C:L	Borderwidth	1	边框宽度。最大255。宽度为0时不显示。
0x0D	0x5DFE	2	固定为0x5DFE。
0x0E	Bordercolor	2	边框颜色。
0x0F:H	Radius	1	圆角。最大255。
0x0F:L	TextHorMargin	1	文字水平边距。使文字与左右边界保持一定距离。最大255。
0x10:H	TextVerMargin	1	文字垂直边距。使文字与上下边界保持一定距离。最大255。
0x10:L	Scrollbar	1	<p>滚动条参数。</p> <p>.7-.4定义了滚动条宽度, 0表示不显示滚动条。最大值15;</p> <p>.3-.0定义了滚动条到窗口边缘的距离: 最大值15;</p> <p>仅多行显示时支持滚动条</p>
0x11	ScrollbarColor	2	滚动条颜色。
0x12	Load_Data	2	<p>加载动画。</p> <p>.F-.C定义了加载类型: 0-8依次表示立即加载、从上飞入、从下飞入、从左飞入、从右飞入、左上飞入、右上飞入、左下飞入、右下飞入。</p> <p>.B-.8定义了加载效果: 0-5线性、缓入、缓出、缓入缓出、过冲、反弹。</p> <p>.7-.0定义了加载时间。单位10ms。最小值100ms, 最大2550ms。</p>

			注意：仅在切换页面时显示一次加载动画。
0x13	KBMode	2	<p>.F定义了是否启用录入功能。1：启用。</p> <p>.E-.C保留。</p> <p>.B-.8定义了键盘显示位置。</p> <p>0-8依次表示屏幕居中、屏幕上边、屏幕下边、屏幕左边、屏幕右边、控件上侧、控件下侧、控件左侧、控件右侧。9-15保留。</p> <p>.7-.0定义了非弹出区域的不透明度。</p> <p>用于淡化背景。该值越小，非弹窗区域越暗，间接突出弹窗区域。最小值0x20。</p>
0x14	Pic_KB	2	键盘控件所在页面。

表 7-3-24 下拉列表控件描述指针定义

地址偏移量 (双字节)	定义	数据长度	说明
0x00	*VP	2	<p>当前选项编号的变量存储器地址。</p> <p>读该变量可以获取当前选项编号。</p> <p>写该变量可以选择指定编号的选项。</p>
0x01	Mode	2	<p>工作模式配置。</p> <p>.F-.B保留。</p> <p>.A-.9定义了窗口弹出时指示图标旋转角度。</p> <p>0：不旋转； 1：旋转90； 2：旋转180； 3：旋转270；</p> <p>.8定义了选项更新后，VP地址数据是否自动上传：</p> <p>0：不上传；1：上传；</p> <p>.7-.6定义了待选数字/文本/标签的对齐方式：</p> <p>0：左对齐； 1：居中对齐； 2：右对齐；</p> <p>.5-.4定义了已选数字/文本/标签的对齐方式：</p> <p>0：左对齐； 1：居中对齐； 2：右对齐；</p> <p>.3-.2定义了菜单弹出方向：</p> <p>0：左侧弹出；</p>

			1: 右侧弹出; 2: 顶部弹出; 3: 底部弹出; .1-.0定义了列表类型: 0: 纯数字; 1: 文本变量; 2: 文字标签
0x02	ADDR	2	1. 列表类型为纯数字时, 表示数字起始值。整型数据, 范围: -32768到32767; 2. 列表类型为文本变量时, 表示列表选项更新标记及文本内容在变量存储器的起始地址; ADDR: 更新标记。更新该值, 触发列表选项文本内容的更新。用于避免内容过长, 不能一条指令发送完整文本内容时, 导致选项异常的情况, 确保内容完整后再更新标记, 刷新选项。 ADDR+1: 文本内容开始地址。 3. 列表类型为文字标签时, 表示所用文字标签编号;
0x03	LEN	2	列表类型为纯数字时, 表示选项个数, 最多支持255个; 列表类型为文本变量时, 表示文本的字地址长度, 最大1024; 列表类型为文字标签时, 表示语言索引的变量存储地址;
0x04	Xs:Ys:Xe:Ye	8	窗口位置: 左上角坐标 (Xs, Ys), 右下角坐标 (Xe, Ye)。
0x08	Txtcolor	2	文本颜色。
0x09	Bgcolor	2	背景颜色。
0x0A	Selcolor	2	选中行背景颜色。
0x0B	Height	2	弹窗最大高度。
0x0C:H	Font0_ID	1	编码方式为0x01-0x04、0x06时ASCII字库位置
0x0C:L	Font1_ID	1	编码方式为0x00、0x05、0x06, 以及0x01-0x04的非ASCII字符使用的字库位置
0x0D	0x5DFE	2	固定为0x5DFE。
0x0E:H	Font_Y_Dots	1	字体Y方向点阵数, 字库中Y方向的点阵数必须为偶数。
0x0E:L	Encode_Mode	1	.7定义了文本显示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, 字符宽度为固定设定的点阵数 .6-.0定义了文本的编码方式: 0x00=8bit 编码, 0x01=GB2312 内码, 0x02=GBK,

			0x03=BIG5, 0x04=SJIS, 0x05=UNICODE, 0x06=UTF8 注意: 菜单类型为文字标签时, 固定使用UTF8编码;
0x0F:H	H_Dis	1	字符间距。
0x0F:L	Radius	1	圆角。最大255。
0x10:H	Pad	1	填充。使文字和边界保持一定距离。最大255。
0x10:L	Borderwidth	1	边框宽度。最大255。
0x11	Bordercolor	2	边框颜色。
0x12:H	Scrollbar	1	滚动条参数。 .7-.4定义了滚动条宽度, 最大值15; .3-.0定义了滚动条到窗口边缘的距离: 最大值15;
0x12:L	Icon_Lib	1	指示图标的图标库存储位置。
0x13	Icon_ID	2	指示图标对应的图标ID。
0x14	ScrollbarColor	2	滚动条颜色。
0x15:H	V_Dis	1	行间距。用于调整待选项文本Y方向的间距。
0x15:L	-	1	未定义, 写0。

表 7-3-25 滚轮控件描述指针定义

地址偏移量 (双字节)	定义	数据长度	说明
0x00	*VP	2	当前选项编号的变量存储器地址。读该变量可以获取当前选项编号。 写该变量可以选择指定编号的选项。
0x01	Mode	2	工作模式配置。 .F-.9保留。 .8定义了是否支持触摸滑动: 0: 不支持; 1: 支持; .7定义了不支持触摸滑动时, 通过修改VP地址切换选项时是否支持动画: 0: 不支持; 1: 支持; .6定义了选项更新后, VP地址数据是否自动上传: 0: 不上传; 1: 上传; .5-.4定义了数字/文本/标签的对齐方式: 0: 左对齐; 1: 居中对齐; 2: 右对齐; .3定义了是否使用透明背景: 0: 不使用; 1: 使用

			使用透明背景时，背景色和选中行背景色不显示。 .2定义了是否使用循环滚动模式： 0: 不使用；1: 使用 .1-.0定义了滚轮类型： 0: 纯数字；1: 文本变量；2: 文字标签
0x02	ADDR	2	1. 滚轮类型为纯数字时，表示数字起始值。整型数据，范围：-32768到32767； 2. 滚轮类型为文本变量时，表示滚轮选项更新标记及文本内容在变量存储器的起始地址； ADDR: 更新标记。更新该值，触发菜单选项文本内容的更新。用于避免内容过长，不能一条指令发送完整文本内容时，导致选项异常的情况，确保内容完整后再更新标记，刷新选项。 ADDR+1: 文本内容开始地址。 3. 滚轮类型为文字标签时，表示所用文字标签编号；
0x03	LEN	2	滚轮类型为纯数字时，表示选项个数，最多支持255个； 滚轮类型为文本变量时，表示文本的字地址长度，最大1024； 滚轮类型为文字标签时，表示语言索引的变量存储地址； 无论哪种滚轮类型，选项个数需满足如下条件： (选项个数*(字体Y方向点阵数+行间距))<=8191。 当使用循环滚动模式时，允许的选项个数为正常时的1/5。
0x04	Xs:Ys:Xe:Ye	8	窗口位置：左上角坐标(Xs, Ys)，右下角坐标(Xe, Ye)。
0x08	Txtcolor	2	文本颜色。
0x09	Bgcolor	2	背景颜色。
0x0A	Selbgcolor	2	选中行背景颜色。
0x0B	Seltxtcolor	2	选中行文本颜色。
0x0C:H	Font0_ID	1	编码方式为0x01-0x04、0x06时ASCII字库位置
0x0C:L	Font1_ID	1	编码方式为0x00、0x05、0x06，以及0x01-0x04的非ASCII字符使用的字库位置
0x0D	0x5DFE	2	固定为0x5DFE。
0x0E:H	Y_Dots	1	字体Y方向点阵数，必须为偶数。
0x0E:L	Encode_Mode	1	.7定义了文本显示字符间距是否自动调整： .7=0字符间距自动调整

			<p>.7=1字符间距不自动调整，字符宽度为固定设定的点阵数</p> <p>.6-.0定义了文本的编码方式：</p> <p>0x00=8bit 编码， 0x01=GB2312 内码， 0x02=GBK， 0x03=BIG5， 0x04=SJIS， 0x05=UNICODE， 0x06=UTF8</p> <p>注意：菜单类型为文字标签时，固定使用UTF8编码；</p>
0x0F:H	Sel_Font0_ID	1	选中行编码方式为0x01-0x04、0x06时ASCII字库位置
0x0F:L	Sel_Font1_ID	1	选中行编码方式为0x00、0x05、0x06, 以及0x01-0x04的非ASCII字符使用的字库位置
0x10:H	Sel_Y_Dots	1	选中行字体Y方向点阵数，必须为偶数。
0x10:L	Linewidth	1	选中行横线宽度；为0不显示。
0x11:H	H_Dis	1	字符间距。
0x11:L	V_Dis	1	行间距。
0x12:H	Radius	1	圆角。最大255。
0x12:L	Pad	1	填充。使文字和边界保持一定距离。最大255。
0x13:H	Borderwidth	1	边框宽度。最大255。
0x13:L	-	1	未定义。写0。
0x14	Bordercolor	2	边框颜色。
0x15	Linecolor	2	选中行横线颜色。

表 7-3-26 开关控件描述指针定义

地址偏移量 (双字节)	定义	数据长度	说明
0x00	*VP	2	变量存储器地址。（需要支持设置初始值）
0x01	Mode	2	<p>工作模式配置。</p> <p>.F-.9保留。</p> <p>.8定义了是否支持触摸点击：</p> <p>0: 不支持；1: 支持；</p> <p>.7-.4定义了调节位：</p> <p>0-15分别对应变量地址从低到高16个二进制位。</p> <p>.3定义了是否使用动画过渡：</p> <p>0: 不使用；1: 使用。</p> <p>.2定义了开关状态更改后VP地址数据是否自动上传：</p> <p>0: 不上传；1: 上传。</p> <p>.1-.0定义了变量模式：</p> <p>0: 按字调节；</p> <p>1: 调节高字节；</p>

			<p>2: 调节低字节； 3: 调节位。</p> <p>变量模式说明：</p> <p>按字调节：将关闭/打开状态变量值完整写入变量地址。</p> <p>调节高字节：将关闭/打开状态变量值低字节写入变量地址高字节。</p> <p>调节低字节：将关闭/打开状态变量值低字节写入变量地址低字节。</p> <p>调节位：关闭时对应位写0，打开时对应位写1。</p>
0x02	offvalue	2	关闭状态变量值。
0x03	onvalue	2	<p>打开状态变量值。</p> <p>打开状态变量值说明：由于控件只有两种状态，打开和关闭，而变量值有很多种，所以只有VP地址对应位的数据等于“打开状态变量值”时显示打开状态，其他值均显示关闭状态。</p>
0x04	Xs:Ys:Xe:Ye	8	窗口位置：左上角坐标(Xs, Ys)，右下角坐标(Xe, Ye)。
0x08	offcolor	2	关闭状态背景颜色。
0x09	oncolor	2	打开状态背景颜色。
0x0A	knobcolor	2	旋钮颜色。
0x0B:H	pad	1	填充。用于调整旋钮到边界的距离。可间接调整旋钮的大小。最大255。
0x0B:L	-	1	未定义，写0。

表 7-3-27 数据记录表格控件描述指针定义

地址偏移量 (双字节)	定义	数据长度	说明
0x00	Mode	2	<p>工作模式配置。</p> <p>.F-.A保留。</p> <p>.9定义了是否启用奇偶行异色：</p> <p>0: 不启用； 1: 启用；</p> <p>.8定义了是否启用“无效位补0”：</p> <p>0: 不启用； 1: 启用；</p> <p>注意：采样类型为数字时该参数有效。</p> <p>.7定义了是否“字符分割显示”：</p>



			<p>0: 整体显示; 1: 分隔显示; 注意: 采样类型为“STRING”时该参数有效。以逗号‘,’(十六进制0x2C)分割字符串在单元格中依次显示。未启用时,所有字符串在一个单元格显示。</p> <p>.6定义了日期和时间是否分行显示:</p> <p>0: 单行; 1: 分行显示; 注意: “时间”列同时显示日期和时间时该参数有效。</p> <p>.5-.4定义了“时间”列显示格式:</p> <p>0: 不显示; 1: 显示时间; 2: 显示日期; 3: 显示日期和时间;</p> <p>.3定义了是否显示“序号”列:</p> <p>0: 不显示; 1: 显示;</p> <p>.2定义了排序方式:</p> <p>0: 时间顺序; 1: 时间逆序;</p> <p>.1-.0定义了标题显示模式:</p> <p>0: 不显示标题; 1: 标题数据来自文字标签; 2: 标题数据来自变量存储器; 3: 使用默认标题; (默认标题: 序号列使用英文字符“Number”; 时间列使用英文字符“Time”; 数据列从1开始编号的数字。)</p>
0x01	ADDR	2	<p>1. 标题文本来自于文字标签时,表示所用文字标签编号;从1开始使用。 2. 标题文本来自于变量存储器时,表示变量存储地址;</p>
0x02	LEN	2	<p>1. 标题文本来自于文字标签时,表示语言索引的变量存储地址。可通过修改该地址数据实现多语言切换。 2. 标题文本来自于变量存储器时,表示文本的字地址长度。最大1024。</p>
0x03	Xs:Ys:Xe:Ye	8	表格位置及大小: 左上角坐标 (Xs, Ys), 右下角坐标 (Xe, Ye)。
0x07	Srcnum	2	引用的数据记录采样编号。从1开始编号。0表示未引用

			数据。
0x08	dataoffset	2	数据列偏移。设置从第几列数据开始显示。
0x09	datacolnum	2	显示的数据列个数。最小值限制为1。
0x0A	controladdr	2	控制地址。 addr:翻页控制。1:下一页;2:上一页;3:右一页;4:左一页。 addr+1:时间戳控制。配合下拉列表显示所选日期的数据(时间戳控制功能仅数据源为数据记录采样时支持)。
0x0B	Txtcolor	2	文本颜色。
0x0C	Bgcolor	2	背景颜色。
0x0D	0x5DFE	2	固定为0x5DFE。
0x0E	Bg2color	2	偶数行背景颜色。用于奇偶行异色。
0x0F	Txttitlecolor	2	标题行文本颜色。
0x10	Bgttitlecolor	2	标题行背景颜色。
0x11	gridcolor	2	网格颜色。
0x12	cellwid	2*8	列宽。最多支持定义前8列的宽度。后续列宽与最后一个列宽数据一致。如果仅设置一个列宽,所有列的宽度均为该值。最小值限制为10。
0x1A	cellheight	2	行高。最小值限制为10。
0x1B:H	barwidth_pad	1	滚动条参数。 .7-.4定义了滚动条宽度,0表示不显示滚动条。最大值15; .3-.0定义了滚动条到窗口边缘的距离:最大值15;
0x1B:L	barminlen	1	滚动条最小长度。
0x1C	barcolor	2	滚动条颜色。
0x1D	0x5DFE	2	固定为0x5DFE。
0x1E:H	int_bit	1	整数位数。0表示自动整数位。使用自动整数位时,不支持无效位补零。
0x1E:L	float_bit	1	小数位数。
0x1F:H	Font0_ID	1	编码方式为0x01-0x04、0x06时ASCII字库位置。
0x1F:L	Font1_ID	1	编码方式为0x00、0x05、0x06,以及0x01-0x04的非ASCII字符使用的字库位置
0x20:H	Font_Y_Dots	1	字体Y方向点阵数,字库中Y方向的点阵数必须为偶数。
0x20:L	Encode_Mode	1	.7定义了文本显示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,字符宽度为固定设定的点阵

			数 .6-.0定义了文本的编码方式： 0x00=8bit 编码， 0x01=GB2312 内码， 0x02=GBK， 0x03=BIG5， 0x04=SJIS， 0x05=UNICODE， 0x06=UTF8， 0x07=UTF8-2
0x21:H	H_Dis	1	字符间距。最大255。
0x21:L	V_Dis	1	行间距。最大255。

表 7-3-28 数据曲线控件描述指针定义

地址偏移量 (双字节)	定义	数据长度	说明
0x00	Mode	2	<p>.F定义了是否启用字符宽度自动调整： 0: 不启用； 1: 启用；</p> <p>.E定义了是否启用“无效位补0”： 0: 不启用； 1: 启用；</p> <p>.D定义了网格线类型： 0: 实线； 1: 虚线；</p> <p>.C-.B定义X轴时间显示方式： 0: 仅显示时间； 1: 仅显示日期； 2: 分行显示日期和时间； 3: 单行显示日期和时间； 注：该选项仅在数据源为“数据记录采样”时有效。</p> <p>.A定义了是否显示光标及当前值： 0: 不显示； 1: 显示；</p> <p>.9定义了是否显示Y轴文字： 0: 不显示； 1: 显示；</p> <p>.8定义了是否显示X轴文字： 0: 不显示； 1: 显示；</p> <p>.7定义了是否显示Y轴刻度线： 0: 不显示； 1: 显示；</p> <p>.6定义了是否显示X轴刻度线： 0: 不显示； 1: 显示；</p> <p>.5定义了是否显示垂直网格线： 0: 不显示； 1: 显示；</p>

			<p>.4定义了是否显示水平网格线: 0: 不显示; 1: 显示;</p> <p>.3定义了刻度区域背景是否透明显示: 0: 不透明; 1: 透明;</p> <p>.2定义了曲线区域背景是否透明显示: 0: 不透明; 1: 透明;</p> <p>.1-.0定义了数据源: 0: 曲线数据来自数据记录采样; 1: 曲线数据来自变量存储器; 2: 曲线数据来自实时曲线; 3: 保留;</p>
0x01	Xs:Ys:Xe:Ye	8	表格位置及大小: 左上角坐标 (Xs, Ys), 右下角坐标 (Xe, Ye)。
0x05	srcnum	2	<p>1. 数据源为数据记录采样时表示“数据记录采样”编号。1-16。</p> <p>2. 数据源为变量存储器时表示变量存储地址。实际所需地址长度与数据类型、数据个数及通道数量有关。如地址设置为0x0100, 数据类型为UINT16 (每个数据占用1个字地址), 数据个数为10, 通道数据量为3。那么 (addr+0) 表示通道0的第一个数据、(addr+1) 表示通道1的第一个数据、(addr+2) 表示通道2的第一个数据、(addr+3) 表示通道0的第二个数据、(addr+4) 表示通道1的第二个数据、(addr+5) 表示通道2的第二个数据依次类推。所需字地址总长度为 (1*3) *10。</p> <p>3. 数据源为曲线缓冲区时表示开始通道编号。0-7。(如开始通道编号设置为3, 通道数量设置为2, 则显示通道3、通道4两条曲线)</p>
0x06	dataoffset	2	<p>1. 数据源为数据记录采样时表示通道偏移。最小值0, 表示不偏移。(如数据记录采样有8个通道, 通道偏移设置为2, 表示从第3个通道的数据开始显示)</p> <p>2. 数据源为变量存储器时表示数据个数。最小值2。</p> <p>3. 数据源为曲线缓冲区时表示最多显示数据个数。2-1024。(曲线缓冲区每通道支持缓冲1024个数据。如设置100, 表示只显示最新的100个数据)</p>
0x07	channelcnt	2	通道数量 (曲线个数)。1-16。

0x08:H	overstepmode	1	越界显示方式: 0: 正常显示、 1: 调整到边界值显示、 2: 不显示数据点及相邻的线段。
0x08:L	datatype	1	数据类型: 0: UINT16; 1: INT16; 2: INT32; 3: UINT8; 4: INT8。 数据源为数据记录采样时仅支持UINT16、 INT16、 INT32 这3种类型，并且数据类型跟随所选数据记录采样设置， 当前参数设置无效。数据源为实时曲线时仅支持 UINT16、 INT16这两种类型。
0x09	controladdr	2	控制地址。 addr:通道隐藏控制。bit为1时表示隐藏该通道。 addr+1:翻页控制。1:下一页;2:上一页;3:首页;4:尾 页。 addr+2:时间戳控制。配合下拉列表显示所选日期的数 据(时间戳控制功能仅数据源为数据记录采样时支持)。
0x0A:H	linewidth	1	曲线宽度。最小值为1。
0x0A:L	point_radius	1	数据点半径。0表示不显示数据点。
0x0B:H	pointgap	1	数据X轴像素间隔。最小值为1。 设置为20，表示每间隔20个像素显示一个数据。
0x0B:L	vdivgap	1	X轴刻度(网格线)间隔数据个数。最小值为1。 设置为3，表示每间隔3个数据，显示1个垂直网格线及X 轴刻度。
0x0C:H	dashwidth	1	网格线为虚线时线段长度。最小值为2。
0x0C:L	dashgap	1	网格线为虚线时空白长度。最小值为2。
0x0D	0x5DFE	2	固定为0x5DFE。
0x0E	hdiv_cnt	2	Y轴刻度(网格线)个数。最小值为2。
0x0F:H	div_len	1	X轴和Y轴刻度长度。最小值为3。
0x0F:L	div_text_space	1	刻度与文本间距。
0x10	start_value	4	Y轴数据开始值。INT32类型。 范围: -2147483648-2147483647。
0x12	rang_value	4	Y轴数据范围。范围: 2-2147483647。
0x14:H	horpad	1	水平方向填充。曲线区域左下角到第一个垂直网格线的 距离。
0x14:L	verpad	1	垂直方向填充。曲线区域左下角到第一个水平网格线的 距离。
0x15	ext_size	2	刻度区域大小。在框选曲线区域的左侧和底部显示Y轴 和X轴刻度及字符的区域大小。

0x16	color	14	前7个通道颜色。
0x1D	0x5DFE	2	固定为0x5DFE。
0x1E	color2	18	后9个通道颜色。
0x27	bgcolor	2	背景颜色。
0x28	divbgcolor	2	刻度区域背景颜色。
0x29	gridcolor	2	网格线颜色。
0x2A	cursorcolor	2	光标颜色。
0x2B	Txtcolor	2	文本颜色。
0x2C	barcolor	2	滚动条颜色。
0x2D	0x5DFE	2	固定为0x5DFE。
0x2E:H	barwidth_pad	1	滚动条参数。 .7-.4定义了滚动条宽度, 0表示不显示滚动条。最大值15; .3-.0定义了滚动条到窗口边缘的距离: 最大值15;
0x2E:L	barminlen	1	滚动条最小长度。最小值为10。
0x2F:H	H_Dis	1	字符间距。最大255。
0x2F:L	Borderwidth	1	边框宽度。最大255。
0x30	Bordercolor	2	边框颜色。
0x31:H	int_bit	1	整数位数。0表示自动整数位。使用自动整数位时, 不支持无效位补零。
0x31:L	float_bit	1	小数位数。
0x32:H	Font0_ID	1	ASCII字库编号。
0x32:L	Font_Y_Dots	1	字体Y方向点阵数, 字库中Y方向的点阵数必须为偶数。

表 7-3-29 数值显示控件描述指针定义

地址偏移量 (双字节)	定义	数据长度	说明
0x00	Xs:Ys	4	左上角坐标。
0x02	Width:Height	4	宽高。
0x04	VP	2	变量地址。
0x05	Work_Mode	2	.F-.C保留。 .B定义了是否进入隐藏状态。1: 设置为隐藏状态。 .A定义了是否启用数值转换。1: 启用。 (启用后, 显示值=变量值*缩放值+偏移值) .9定义了是否支持无效位补0。1: 支持。 .8定义了是否支持密码模式。1: 支持。 .7定义了是否支持录入。1: 支持。.6-.5定义了是否支



			<p>持警告色。 1: 支持。 (启用范围限制后支持该功能) 0x00: 不启用警告色。 0x01: 启用警告色, 并用于文字。 0x02: 启用警告色, 并用于背景。</p> <p>.4定义了是否支持范围限制。 1: 支持。</p> <p>.3-.0定义了变量类型:</p> <p>0x00=整数(2字节): -32768到32767 0x01=长整数(4字节): -2147483648到2147483647 0x02=VP*高字节: 0到255 0x03=VP*低字节: 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65536 0x06=无符号长整数(4字节): 0到4294967295</p>
0x06	V_min	4	下限值。长整数(4字节)-2147483648到2147483647。
0x08	V_max	4	上限值。长整数(4字节)-2147483648到2147483647。
0x0A	Digit	2	<p>.F-.8定义了小数位数。 0-19。</p> <p>.7-.0定义了整数位数。 0-20。 0表示自动整数位数, 此时不支持无效位补零。</p>
0x0B	Zoom	2	数值转换缩放值。-32768到32767。
0x0C	Offset	2	数值转换偏移值。-32768到32767。
0x0D	0x5DFE	2	固定为0x5DFE。
0x0E	Color_min	2	下限警告色。
0x0F	Color_max	2	上限警告色。
0x10	Label_Number	2	“单位”字符所用文字标签编号; 字符使用UTF8编码存储, 需添加Unicode编码的字库。
0x11	VP_Uint_Index	2	“单位”标签语言索引变量存储地址。
0x12	Disp_Mode	2	<p>.F-.E定义了宽度调整模式。</p> <p>0: 不调整; 1: 所有字符均自动调整宽度; 2: 仅小数点自动调整宽度;</p> <p>.D-.C定义了是否支持闪烁显示。</p> <p>0: 不闪烁; 1: 闪烁间隔0.5s; 2: 闪烁间隔1s; 3: 闪烁间隔2s;</p>

			<p>.B-.A定义了对齐方式。</p> <p>0: 左对齐； 1: 居中对齐； 2: 右对齐；</p> <p>.9-.8定义了背景模式。</p> <p>0: 透明背景； 1: 单色背景； 2: 图标背景；</p> <p>.7-.0定义了背景图标库ID。</p>
0x13	Bg_Data	2	<p>背景模式为单色背景时表示背景颜色。 背景模式为图标背景时表示图标编号。</p>
0x14	Color	2	文字颜色。
0x15:H	Font_Y_Dots	1	字体Y方向点阵数，字库中Y方向的点阵数必须为偶数。
0x15:L	H_Dis	1	字符间隔。
0x16:H	Font0_ID	1	FONT0字库（用于显示数字或ASCII单位）。
0x16:L	Font1_ID	1	FONT1字库（用于显示非ASCII编码的“单位”字符。固定使用UTF8编码，需要添加UNICODE字库）。
0x17:H	Radius	1	圆角。最大255。
0x17:L	Borderwidth	1	边框宽度。最大255。宽度为0时不显示。
0x18	Bordercolor	2	边框颜色。
0x19	Load_Data	2	<p>加载动画。</p> <p>.F-.C定义了加载类型：0-8依次表示立即加载、从上飞入、从下飞入、从左飞入、从右飞入、左上飞入、右上飞入、左下飞入、右下飞入。</p> <p>.B-.8定义了加载效果：0-5线性、缓入、缓出、缓入缓出、过冲、反弹。</p> <p>.7-.0定义了加载时间。单位10ms。最小值100ms，最大2550ms。</p> <p>注意：仅在切换页面时显示一次加载动画。</p>
0x1A	KBMode	2	<p>.F-.C保留。</p> <p>.B-.8定义了键盘显示位置。</p> <p>0-8依次表示屏幕居中、屏幕上边、屏幕下边、屏幕左边、屏幕右边、控件上侧、控件下侧、控件左侧、控件右侧。9-15保留。</p> <p>.7-.0定义了非弹出区域的不透明度。</p> <p>用于淡化背景。该值越小，非弹窗区域越暗，间接突出</p>

			弹窗区域。最小值0x20。
0x1B	Pic_KB	2	键盘控件所在页面。

表 7-3-30 多功能按钮控件描述指针定义

地址偏移量 (双字节)	定义	数据长度	说明
0x00	Xs:Ys	4	左上角坐标。
0x02	Width:Height	4	按钮宽高。
0x04	AudioNum	2	按钮按下时播放音频文件编号。0xFFFF表示不播放音频。
0x05:H	KeyCode	1	键码。0表示不触发。（仅用于触发Lua脚本的触摸回调函数）
0x05:L	VPMode	1	0x00=调节VP地址（整型数）； 0x01= 调节VP地址高字节地址（一个字节的无符号数，VP_H）； 0x02=调节VP地址低字节地址（一个字节的无符号数，VP_L）； 0x10-0x1F对VP地址的指定位（0x10对应VP.0, 0x1F对应VP.F）
0x06	VP	2	变量地址指针，返回数据由VPMode决定。
0x7	Work_Mode	2	.8定义了是否进入隐藏状态。 1: 设置为隐藏状态。 .7定义了是否进入无效状态。 1: 设置为无效状态。 (注意：无效状态和隐藏状态同时设置时，隐藏状态有效) .6定义了数据是否自动上传。0: 不上传、1自动上传。 .5定义了递增或递减模式时是否支持循环调节。1: 支持。 .4定义了递增或递减模式时是否支持长按调节。1: 支持。 .3-.0定义了工作模式： 0=常量模式。在所选执行时机将常量值写入变量存储器。 1=点动模式。分别在按下和抬起时写入不同的常量值。 2=按钮状态模式。分别在按下、长按和抬起时写入不同的常量值。



			<p>3=递增模式。变量值依次增加。</p> <p>4=递减模式。变量值依次减小。</p> <p>5=页面切换。在所选执行时机切换到指定页面，并更新对应变量。</p> <p>6=弹出页面。</p> <p>7=关闭页面。在所选执行时机关闭弹窗页面并更新对应变量。</p> <p>8=密码切换页面。在所选执行时机弹出键盘，键盘录入数据与密码地址变量数据一致才切换页面。此时固定使用VP地址无符号长整型数据。</p>
0x08	DATA1	2	<p>不同工作模式下表示的含义如下：</p> <p>模式0:</p> <ul style="list-style-type: none"> . 0-. 1表示执行时机： 0=抬起时、 1=按下时、 2=长按时； . 2-. F保留。 <p>模式1或2: 按下时写入VP的常量值；</p> <p>模式3或4: 最小值； INT16；</p> <p>模式5:</p> <ul style="list-style-type: none"> . 0-. 1表示执行时机， 0=抬起时、 1=按下时、 2=初次长按时； . 2-. 5表示页面切换动画效果。（0表示无动画。1-13依次表示13种动画切图方式。） . 6-. F保留。 <p>模式6:</p> <ul style="list-style-type: none"> . 0-. 1表示执行时机， 0=抬起时、 1=按下时、 2=初次长按时； . 2-. 5表示弹出位置。0-8依次表示屏幕居中、屏幕上边、屏幕下边、屏幕左边、屏幕右边、控件上侧、控件下侧、控件左侧、控件右侧。9-15保留。 . 6-. 7保留。 . 8-. F表示非弹窗区域的不透明度，最小值0x20。（背景灰化效果） <p>模式7:</p> <ul style="list-style-type: none"> . 0-. 1表示执行时机， 0=抬起时、 1=按下时、 2=初次长按时； . 2-. F保留。 <p>模式8:</p> <ul style="list-style-type: none"> . 0-. 1表示执行时机， 0=抬起时、 1=按下时、 2=初次长按时；

			<p>按时；</p> <p>.2-.5表示弹出位置。0-8依次表示屏幕居中、屏幕上边、屏幕下边、屏幕左边、屏幕右边、控件上侧、控件下侧、控件左侧、控件右侧。9-15保留。</p> <p>.6-.7表示密码数据位数。</p> <p>0=6位、1=7位、2=8位、3=9位。</p> <p>.8-.F表示非弹窗区域的不透明度，最小值0x20。（背景灰化效果）</p>
0x09	DATA2	2	<p>不同工作模式下表示的含义如下：</p> <p>模式0：写入VP的常量值；</p> <p>模式1或2：抬起时写入VP的常量值；</p> <p>模式3或4：最大值； INT16；</p> <p>模式5：切换页面编号。</p> <p>模式6：弹出页面编号。</p> <p>模式7：关闭弹窗时写入VP的常量值；</p> <p>模式8：键盘所在页面编号；</p>
0x0A	DATA3	2	<p>模式0或1：未定义。</p> <p>模式2：长按时写入VP的常量值；</p> <p>模式3或4：调节步长； UINT16；</p> <p>模式5：页面切换时写入VP的常量值；</p> <p>模式6：控制弹窗打开的变量地址。（0xFFFF表示不支持通过变量打开弹窗）；</p> <p>模式7：控制弹窗关闭的变量地址。（0xFFFF表示不支持通过变量关闭弹窗）；</p> <p>模式8：控制弹出键盘页面的变量地址。（0xFFFF表示不支持通过变量弹出键盘页面）。变量值为0x5AA5时弹出键盘页面；</p>
0x0B	DATA4	2	<p>模式0-5：未定义。</p> <p>模式6：控制弹窗打开的变量值，非0值。当“DATA3”地址的值等于变量值时，将弹出指定页面；</p> <p>模式7：控制弹窗关闭的变量值，非0值。当“DATA3”地址的值等于变量值时，将关闭弹窗；</p> <p>模式8：目标切换页面编号；</p>
0x0C:H	Long_press_delay	1	长按迟滞时间（第一次按下多长时间后触发长按）， 单位100ms 。最大25500ms。
0x0C:L	Long_press_space	1	长按间隔时间（长按循环触发的间隔）， 单位10ms 。最小30ms，最大2550ms。



0x0D	0x5DFE	2	固定为0x5DFE。
0x0E	Load_Data	2	<p>加载动画。</p> <p>.F-.C定义了加载类型：0-8依次表示立即加载、从上飞入、从下飞入、从左飞入、从右飞入、左上飞入、右上飞入、左下飞入、右下飞入。</p> <p>.B-.8定义了加载效果：0-5线性、缓入、缓出、缓入缓出、过冲、反弹。</p> <p>.7-.0定义了加载时间。单位10ms。最小值100ms，最大2550ms。</p> <p>注意：仅在切换页面时显示一次加载动画。</p>
0x0F:H	Radius	1	圆角。最大255。
0x0F:L	Borderwidth	1	边框宽度。最大255。宽度为0时不显示。
0x10	Bordercolor	2	边框颜色。
0x11:H	Icon_Lib	1	图标库存储位置。
0x11:L	BG_Mode	1	<p>.7-.6保留。</p> <p>.5背景模式为图标时的图标对齐方式。 0=居中对齐，1=左上角对齐。</p> <p>.4背景模式为图标时的图标显示模式。 0=透明（不显示背景），1=显示图标背景。</p> <p>.3-.0定义按钮背景模式。</p> <p>支持的按钮背景模式：0=透明、1=纯色、2=图标、3=切图。注意：背景模式为图片或图标时，长按时的背景数据未设置或无效时，将使用按下时的背景数据。</p>
0x12	BG_Data1	2	<p>抬起时的按钮背景数据。</p> <p>背景模式为1时表示颜色数据。</p> <p>背景模式为2时表示图标编号。0xFFFF表示未选择图标。</p> <p>背景模式为3时表示用于切图的图片文件编号。0xFFFF表示未选择图片。</p>
0x13	BG_Data2	2	按下时的按钮背景数据。定义如上所示。
0x14	BG_Data3	2	长按时的按钮背景数据。定义如上所示。
0x15	BG_Data4	2	无效时的按钮背景数据。定义如上所示。
0x16	Text_Mode	2	<p>.F-.A保留。</p> <p>.9-.8定义了字符显示模式：</p> <p>0: 多行显示；1: 滚动显示；2: 循环滚动显示；</p> <p>.7-.6定义了垂直对齐方式：</p> <p>0: 上对齐；1: 居中对齐；2: 下对齐；</p> <p>.5-.4定义了水平对齐方式：</p> <p>0: 左对齐；1: 居中对齐；2: 右对齐；</p>

			<p>.3-.2定义了是否闪烁显示:</p> <p>0: 不闪烁; 1: 闪烁间隔0.5s;</p> <p>2: 闪烁间隔1s; 3: 闪烁间隔2s;</p> <p>.1-.0定义了文字数据来源。</p> <p>0: 不显示文字; 1: 文字标签; 2: 变量存储器;</p>
0x17	NUM	2	<p>1. 文字数据来源为文字标签时, 表示所用文字标签编号; 从1开始使用。</p> <p>2. 文字数据来源为变量存储器时, 表示起始地址; 各个状态下显示的文字位于同一字符串中, 使用' , ' (0x2C)进行分隔, 依次表示抬起、按下、长按、无效时显示的文字。如“抬起, 按下, 长按, 无效”。</p>
0x18	LEN	2	<p>1. 文字数据来源为文字标签时, 表示语言索引的变量存储地址。可通过修改该地址数据实现多语言切换。</p> <p>2. 文字数据来源为变量存储器时, 表示数据的字地址长度, 最大128;</p>
0x19	Color	2	字体颜色。
0x1A:H	Font0_ID	1	编码方式为0x01-0x04、0x06时ASCII字库位置
0x1A:L	Font1_ID	1	编码方式为0x00、0x05、0x06、0x07, 以及0x01-0x04的非ASCII字符使用的字库位置。
0x1B:H	Font_Y_Dots	1	字体Y方向点阵数, 字库中Y方向的点阵数必须为偶数。
0x1B:L	Encode_Mode	1	<p>.7定义了文本显示字符间距是否自动调整:</p> <p>.7=0字符间距自动调整</p> <p>.7=1字符间距不自动调整, 字符宽度为固定设定的点阵数</p> <p>.6-.0定义了文本的编码方式:</p> <p>0x00=8bit 编码, 0x01=GB2312 内码, 0x02=GBK, 0x03=BIG5, 0x04=SJIS, 0x05=UNICODE, 0x06=UTF8, 0x07=UTF8-2。</p> <p>注意: 数据来源为文字标签时, 固定使用UTF8编码;</p>
0x1C:H	H_Dis	1	字符间距。
0x1C:L	V_Dis	1	行间距。
0x1D	0x5DFE	2	固定为0x5DFE。
0x1E:H	TextHorMargin	1	文字水平边距。使文字与左右边界保持一定距离。
0x1E:L	TextVerMargin	1	文字垂直边距。使文字与上下边界保持一定距离。
0x1F:H	Rolling_Dis	1	滚动速度: 每秒移动的像素数量。1-255。
0x1F:L	-	1	未定义。

0x20	DATA5	2	不同工作模式下表示的含义如下： 模式0-5： 未定义。 模式6： 弹窗打开时写入VP的常量值。 模式7： 未定义。 模式8： 弹出键盘页面时写入VP的常量值。
0x21	DATA6	2	不同工作模式下表示的含义如下： 模式0-7： 未定义。 模式8： 表示密码变量地址。录入数据与该地址的“无符号长整型”数据一致时切换页面。
0x22	DATA7	2	不同工作模式下表示的含义如下： 模式0-7： 未定义。 模式8： 控制弹出键盘页面的变量值。

表 7-3-31 数据表格控件描述指针定义

地址偏移量 (双字节)	定义	数据长度	说明
0x00	Xs:Ys:Xe:Ye	8	表格位置及大小：左上角坐标 (Xs, Ys)，右下角坐标 (Xe, Ye)。
0x04	Mode	2	工作模式配置。 .F保留。 .E定义了是否支持数据自动上传： 0: 不支持；1: 支持； 选中单元格、图标变量更改后上传数据。 .D定义了是否显示选中单元格背景色： 0: 不显示；1: 显示； .C定义了是否显示选中行背景色： 0: 不显示；1: 显示； .B定义了是否支持编辑表格内容： 0: 不支持；1: 支持；启用后支持录入数字、字符串或更改图标变量。 .A定义了是否使用奇偶行异色： 0: 不使用；1: 使用； .9定义了是否显示边框线： 0: 不显示；1: 显示； .8定义了是否显示垂直网格线： 0: 不显示；1: 显示； .7定义了是否显示水平网格线：

			<p>0: 不显示; 1: 显示;</p> <p>.6定义了是否固定显示首行:</p> <p>0: 不固定; 1: 固定;</p> <p>.5定义了是否固定显示数据首列:</p> <p>0: 不固定; 1: 固定;</p> <p>.4定义了是否显示“序号”列:</p> <p>0: 不显示; 1: 显示; 从1开始编号。启用时将固定显示序号列。</p> <p>.3定义了是否显示“标题”行:</p> <p>0: 不显示; 1: 显示; “标题”数据来自“文字标签”。</p> <p>.1-.2定义了数据模式:</p> <p>0: 单元格数值数据; 表格数据区域均显示数字。</p> <p>1: 单元格文本数据; 表格数据区域均显示文本字符串。</p> <p>2: 单元格自定义数据; 表格数据区域可自由组合显示数字, 文本或图标。</p> <p>3-7: 保留。</p>
0x05	RowNum	2	行数目。1-65535。
0x06	ColNum	2	列数目。1-255。
0x07	dataaddr	2	表格数据起始地址。
0x08	datalen	2	<p>数据模式为0时表示数据类型: 不同数据类型, 每个单元格所需变量地址长度不同。</p> <p>0x00=整数(2字节): -32768到32767</p> <p>0x01=长整数(4字节): -2147483648到2147483647</p> <p>0x02/0x03=单字节: 0到255</p> <p>0x04=超长整数(8字节): -9223372036854775808到 9223372036854775807</p> <p>0x05=无符号整数(2字节): 0到65536</p> <p>0x06=无符号长整数(4字节): 0到4294967295</p> <p>数据模式为1或数据模式为2时表示单元格数据“字长度”:</p> <p>为0x10表示每个单元格的变量地址字长度为16。</p> <p>当设置为0时, 表示可以按列设置单元格的数据长度, 表格数据存储位置相应后延(列数/2)向上取整个字地址。例如, 表格数据起始地址为0x1000, 列数为7, 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>

0x09	Digit	2	仅数据模式为0时有效。 .F-.8定义了小数位数。0-19。 .7-.0定义了整数位数。0-20。 0表示自动整数位数。
0x0A	TitleAddr	2	启用文字标签标题时，文字标签编号。
0x0B	TitleLangueAddr	2	文字标签语言索引地址。
0x0C	controladdr	2	控制地址。 addr+0: 翻页控制。1:下一页;2:上一页;3:右一页;4:左一页。 addr+1: 当前选中单元格行地址。0xFFFF表示无效。 addr+2: 当前选中单元格列地址。0xFFFF表示无效。 仅数据区域有效，点击“标题”行或“序号”列时不会触发选中单元格。
0x0D	0x5DFE	2	固定为0x5DFE。
0x0E	Pic_KB1	2	数字键盘所在页面ID。
0x0F	Pic_KB2	2	中英文键盘所在页面ID。
0x10	cellwid[8]	2*8	列宽。最多支持定义前8列的宽度。后续列宽与最后一个列宽数据一致。 如果仅设置一个列宽，所有列的宽度均为该值。 最小值限制为10。
0x18	cellheight	2	行高。 最小值限制为10。
0x19	Txtcolor	2	文本颜色。
0x1A	Bgcolor	2	背景颜色。
0x1B	Bg2color	2	偶数行背景颜色。使用奇偶行异色时有效。
0x1C	gridcolor	2	网格颜色。
0x1D	0x5DFE	2	固定为0x5DFE。
0x1E	Txttopcolor	2	固定首行文本颜色。
0x1F	Bgtopcolor	2	固定首行背景颜色。
0x20	selectrowcolor	2	选中行背景色。
0x21	selectcellcolor	2	选中单元格背景色。
0x22:H	barwidth_pad	1	滚动条参数。 .7-.4定义了滚动条宽度，0表示不显示滚动条。最大值15； .3-.0定义了滚动条到窗口边缘的距离：最大值15；
0x22:L	barminlen	1	滚动条最小长度。
0x23	barcolor	2	滚动条颜色。
0x24:H	Font0_ID	1	编码方式为0x01-0x04、0x06时ASCII字库位置。

0x24:L	Font1_ID	1	编码方式为0x00、0x05、0x06, 以及0x01-0x04的非ASCII字符使用的字库位置。
0x25:H	Font_Y_Dots	1	字体Y方向点阵数, 字库中Y方向的点阵数必须为偶数。
0x25:L	Encode_Mode	1	.7定义了文本显示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, 字符宽度为固定设定的点阵数 .6-.0定义了文本的编码方式： 0x00=8bit 编码, 0x01=GB2312 内码, 0x02=GBK, 0x03=BIG5, 0x04=SJIS, 0x05=UNICODE, 0x06=UTF8, 0x07=UTF8-2
0x26:H	H_Dis	1	字符间距。最大255。
0x26:L	V_Dis	1	行间距。最大255。

8 VGUS 开发工具使用说明

本章内容提供有配套的视频教程“**VGUS 开发工具及设计流程**”。

VGUS 开发工具是一款组态型、图形用户界面设计软件。软件功能强大且易学，一站式管理，用户无需直接操作繁琐的配置文件。

VGUS 开发工具免安装，可直接运行。支持 Windows XP、Win7、Win8 和 Win10 操作系统。Win8 或 Win10 系统下需要设置兼容性才能运行。首次运行 VGUS 开发工具，软件弹出一个窗口，提示设置兼容性的方法，Win8 和 Win10 系统务必按照提示正确设置兼容性。

8.1 操作界面

VGUS 开发工具界面如图 8-1-1 所示。

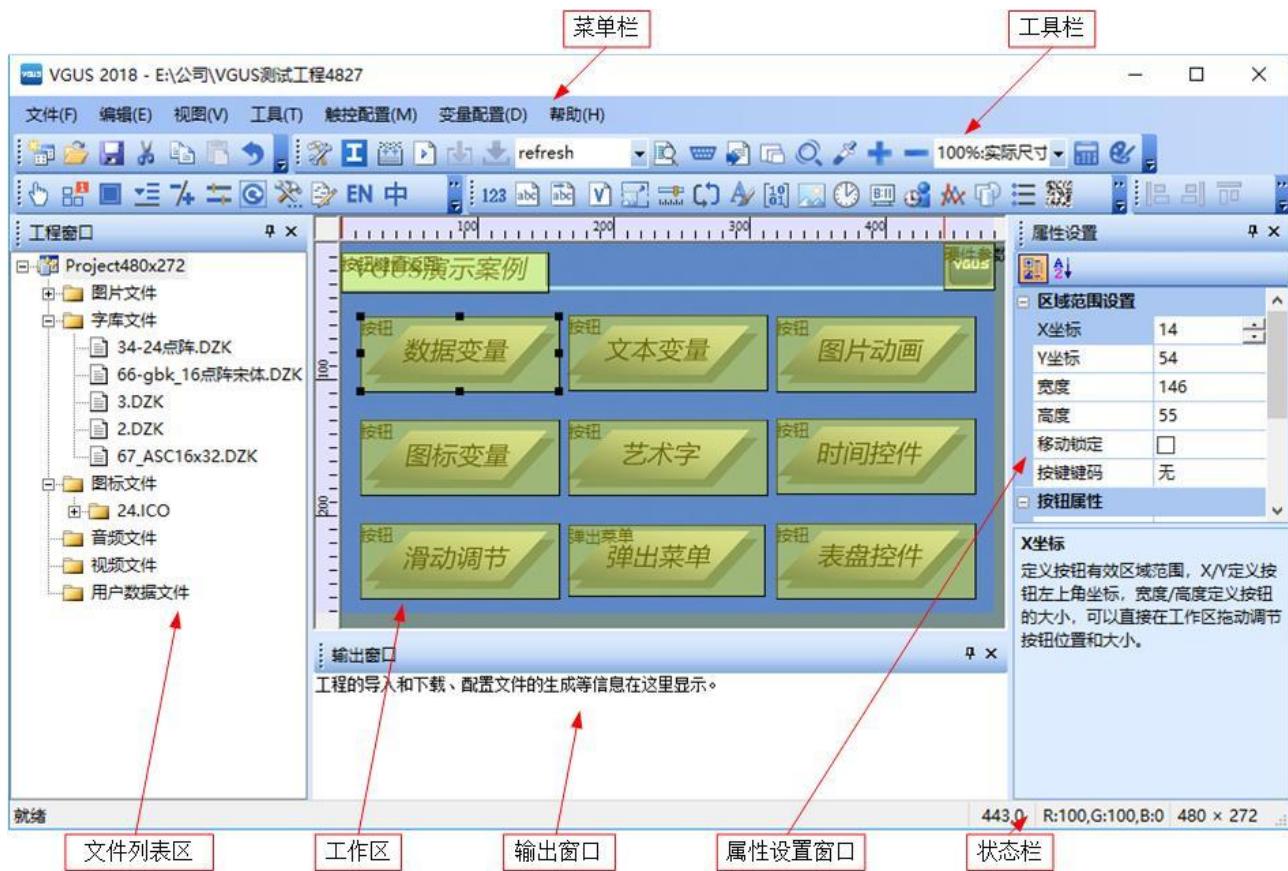


图 8-1-1 VGUS 开发工具软件界面

➤ 菜单栏

- 文件里面的新建工程经常用到，里面的屏幕属性设置非常重要。
- 编辑菜单功能提供了复制粘贴等常规功能和针对控件的位置大小等编辑功能。
- 视图菜单用于选择工具栏、各窗口是否显示，显示的项目前面会有标记。
- 工具菜单包含了 VGUS 中提供的各类小工具软件，功能如表 8-1-1 所示。屏参配置、生成配置文件、虚拟串口屏、下载到 U 盘、串口调试工具、项目属性设置等都是经常用到的功能。

表 8-1-1 工具汇总



工具名称	功能说明
屏参配置	用于设置串口波特率、指令帧头、CRC 校验、以及外设和屏保等特性, 详见 8.2 节
生成配置文件	界面设计完毕后, 生成下载文件夹
虚拟串口屏	预览和虚拟调试设计效果, 参见文档“ 虚拟串口屏使用说明 ”
下载到 U 盘	把下载文件夹直接复制到 TF 卡里, 用于脱机下载, 适合批量生产使用
变量查看	查看所有控件的地址分配情况
串口调试工具	类似串口调试助手的功能, 可以直接生成指令, 调试串口屏更方便
导入工程	可以将 VGUS3.2 的工程、其它厂商 xGUS 类的下载文件夹导入生成新工程
项目属性设置	变换产品系列、修改工程分辨率、工程旋转 180 度等
查找	查找变量
16 位颜色数据	将指定颜色转换为 5-6-5 格式数据
放大	调整工作区大小
缩小	调整工作区大小
计算器	打开计算器
图像编辑工具	打开画图
点阵字库生成工具	字库生成软件, 支持生成 DZK 字库和 ADZK 字库。
图片压缩工具	图片压缩存储为 JPG 格式, 支持批量处理、图片质量设置。
图片渐变色优化工具	渐变色图片优化处理。

■ 触控配置用于设计各种触摸按钮控件, 详细内容见第 5 章。

■ 变量配置用于设计各种显示控件和高级控件, 详细内容见第 4 章和第 6 章。

■ 帮助用于可查看当前软件版本。当打开一个工程时, 可查看生成该工程所使用的软件版本。

➤ 工具栏

■ 提供了常用菜单的图标快捷方式, 鼠标短暂停留可显示文字提示。

➤ 文件列表区

■ 非常重要, 用于管理界面中用到的各种素材文件。通过鼠标右键添加、删除文件。

➤ 工作区

■ 用户在此区域进行图形化界面设计, 可以拖放移动控件等。

■ 点击工作区非控件的地方, 属性设置窗口会出现**全局设置属性**, 可以设置**自动保存时间间隔**。在设计过程中, 为防止因意外因素造成的数据丢失, 请设置自动保存, 或者及时手动保存文件。

➤ 属性设置窗口

■ 用于显示选中控件的属性, 不同的控件对应不同的属性, 可以对其查看修改;

■ 点击某项属性, 属性栏下方将显示相应的帮助内容。

➤ 输出窗口: 用于显示工程导入、配置文件生成、下载等操作过程信息。

➤ 状态栏: 用于显示鼠标所指工作区像素点的坐标和颜色信息, 以及项目分辨率。

8.2 屏参配置

点击菜单栏上“工具->屏参配置”，弹出屏参配置窗口，如图 8-2-1 所示。



图 8-2-1 屏参配置

➤ 串口参数配置：

- 设置串口波特率，**VGUS 屏出厂波特率为 115200bps。**
- 设置指令帧头，指令帧头的设置主要达到以下两个目的：(1) 用于串口指令帧的识别和同步；(2) 多台VGUS屏并联工作时，把帧头作为设备地址加以区分。**VGUS屏出厂帧头为A5 5A。**
- 设置是否启用 CRC 校验。
- 设置串口电平。2021.6.1 以后的固件版本支持，硬件是否支持需要查看规格书确认。

➤ 参数配置：

- 设置变量初值，可以全部设置为 0、或者启用变量属性设置的初始值。
- 背光设置（启用屏保）。

设置是否启用屏保，以及屏保延时动作时间、屏保亮度、和正常显示亮度，如表 8-2-1；当屏幕进入屏保后，第一次点击触摸屏不会触发动作，而只是恢复进入正常显示状态；利用点亮亮度寄存器，也可以实现设置串口屏的默认开机亮度。

表8-2-1 屏保功能寄存器

名称	取值范围	说明
延时时间 R8	0x01-0xFF	触摸屏间隔多久没有触摸后进入屏保，单位为 1.0 秒。
屏保亮度 R7	0x00-0x40	触摸屏控制背光启动后，一段时间不点击触摸屏，进入屏保后的背光亮度。
点亮亮度 R6	0x00-0x40	触摸屏控制背光启动后，点击触摸屏后背光正常点亮的亮度。

如设置启用屏保、R6=0x40、R7=0x10、R8=0x1E，当 30 秒 (0x1E) 不点击触摸屏，背光亮度将自动降低到 0x10 (25% 亮度)；点击触摸屏后，背光亮度将自动调节到 0x40 (100% 亮度)。

注：R7（屏保保护亮度）设置为 0，屏保时，将自动进入休眠模式。具体使用方法参考 [《屏保与休](#)

睡眠使用指南》。

- 设置 VGUS 运行周期。页面上的变量采用定时刷新，建议将固定的背景提示语、单位提示符、表格等直接做到图片上，减少串口屏的刷新任务，有利于提高串口屏的刷新速度。
- 设置是否启用电容屏灵敏度设置。设置范围 0x00-0x1F。
- 设置 TF 卡脱机下载后启动触摸校准，通过该功能可以很方便的实现设备出厂前校准触摸屏。
- 设置触屏录入参数后自动上传到串口。
- 设置关闭蜂鸣器。
- 设置开机页面的图片序号。
- 设置读指令的返回帧延时时间，单位为毫秒。
- 设置 miscellaneous 选项，共 16 个，其中各个位定义如表 8-2-2。

表8-2-2 miscellaneous选项位功能定义 (0代表对应位不勾选, 1代表勾选)

位	功能说明
.F-C	未定义
.B	表示是否关闭视频播放寄存器映射（仅 VTb 系列 202501020 以后固件版本支持）。 0: 不关闭； 1: 关闭。
.A	表示 ASCII 字库编码小于 0x20 时字符宽度（20250627 以后固件版本支持）。 0: 宽度为 0（默认状态）； 1: 宽度为 Y 方向点阵数的一半。
.9	表示控制文本自动换行方式（VTc 和 VTb 系列有效）。 0: 宽度不足以显示“单词”时自动换行； 1: 宽度不足以显示单个“字符”时自动换行。
.8	寄存器 0x01（背光亮度控制寄存器）设置为 0 是否进入休眠模式。 SDWb 系列 2022.3.16 及以后固件版本支持。VTc 系列支持。 0: 进入休眠（默认状态）； 1: 不进入休眠。
.7	上电串口屏是否自动外发读取版本号指令（读取寄存器 0x00 单元），仅 2020.7.1 以后固件版本支持。 0: 上电不外发； 1: 上电自动外发。
.6-5	修改触摸屏类型（工程与系统参数一起下载时，工程高优先级），2020.6.1 以后固件版本支持。 00: 不修改触摸屏类型； 01: 修改为电阻触摸屏（建议同时开启“TF 卡下载配置文件后启动触摸校准”功能）； 10: 修改为 I 类电容触摸屏（GT911）； 11: 未定义。
.4	系统自检寄存器错误代码是否在屏幕上显示 0: 关闭屏幕显示错误代码信息； 1: 启用屏幕显示错误代码信息。
.3	数据变量录入范围限制控制位 0: 不考虑小数位限制（默认状态）；



	1: 考虑小数位限制。 例如: 变量设置为 2 位整数 1 位小数, 范围设置为 1-100, 不勾选则实际允许录入范围为 1~100; 勾选后因为有一位小数位, 则实际允许录入范围是 0.1~10.0。
.2	是否启用触摸屏自动校准控制位。 自动校准是指 4 秒内快速点击触摸屏的非按钮区域超过 20 次, 则进入触摸屏校准模式。 0: 关闭自动校准 (默认状态); 1: 打开自动校准。
.1	数据录入长度限制控制位 0: 非循环录入, 录入长度达到设置值后禁止录入 (默认状态); 1: 循环录入, 录入长度超出设置值后高位挤出。
.0	触控和弹出键盘等像素控制位 0: 弹出图块在右侧和下侧各少一个像素 (默认状态); 1: 弹出图块与控件指定大小位置一致。

8.3 设计流程

详细设计流程如图 8-3-1 所示。

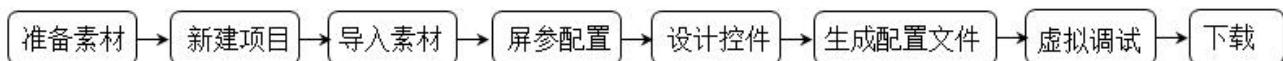


图 8-3-1 详细设计流程

1 准备素材

开始使用 VGUS 开发工具设计界面前, 首先要准备好用到的背景图片、图标, 以及可能用到的字库、音频和视频文件等。文件格式要求参见 8.5 节。

画面中固定显示的文字提醒语、符号、表格和图标等, 建议全部做到背景图片上, 切不可用变量方式来实现。因为页面上的变量越多, 会占用 CPU 的时间越多, 影响显示效果。

2 新建项目

打开软件, 选择“文件→新建”选项, 弹出如图 8-3-2 对话框。正确设置产品系列、存储空间、分辨率、旋转角度、项目名称和项目路径。产品系列、存储空间大小、分辨率在产品规格书上都可以查到对应信息。

中显所有系列串口屏, 统一使用 VGUS 开发工具设计用户界面, 新建工程时要根据实际使用的产品型号正确设置产品系列, 否则编译生成下载文件夹后无法下载到串口屏里。

产品系列、存储空间、分辨率三项参数, 要查看所用串口屏对应型号规格书, 正确选择设置。

旋转角度根据需要设置, 如果遇到横屏竖用、或者竖屏横用就需要设置 90 度旋转。注意: 设置 90 度旋转时, 分辨率不需要交换, 还是要参照规格书上给出的分辨率正确设置。例如, SDWb070T74T 这款默认为横屏, 规格书给出的分辨率为 800x480, 如果需要用作竖屏显示, 就要设置 90 度旋转, 但是分辨率依然要按照规格书设置为 800x480, 而不能设置为 480x800。



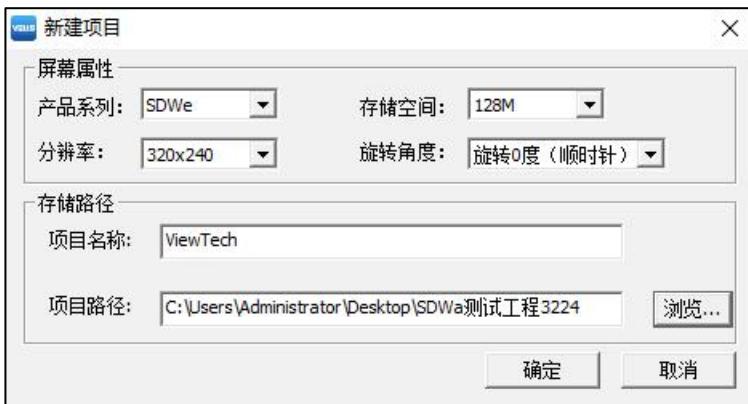


图 8-3-2 新建项目对话框

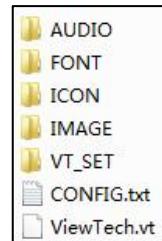


图 8-3-3 项目文件夹

点击图 8-3-2 上确定按钮后,将在项目路径下生成名称与项目名称相同的项目文件夹,如图 8-3-3 所示,用以存放整个项目文件。“*.vt”是 VGUS 项目工程文件,用户可以使用 VGUS 开发工具直接打开。

3 导入素材

把准备好的图片、图标、字库、音频文件导入到 VGUS 开发工具左侧的文件列表窗口中。

4 屏参配置

根据项目需要, 正确设置串口波特率、指令帧头、CRC 校验、蜂鸣器、触摸屏和屏保等外设特性。

5 设计控件

包括显示控件 (变量配置/第4章) 和触摸控件 (触控配置/第5章), VGUS 开发工具提供有功能完善的多种显示控件和触摸控件和高级控件, 分别见第4章、第5章、第6章。在 VGUS 开发工具菜单栏中或者工具栏上找到相应控件按钮, 点击按钮并拖放到工作区, 调整控件大小及位置、并在右侧属性窗口中完成控件属性设置。

6 生成配置文件

设计完所有显示控件和触摸控件后, 点击菜单栏或者工具栏上的“生成配置文件”按钮, 会在项目文件夹下生成下载文件夹。输出窗口会提示下载文件夹是否生成成功。产品系列不同, 生成的下载文件夹名称也不同, 详细信息参考8.4节。

7 虚拟调试

VGUS 开发工具提供有虚拟串口屏和串口调试工具功能。在生成配置文件夹后, 可以预览及虚拟调试设计效果 (实际效果以下载到VGUS屏内为准)。详细方法参见文档[《虚拟串口屏使用方法》](#)。

8.4 工程下载

8.4.1 脱机下载

SDWn/ VTc系列支持TF卡脱机下载，SDWb系列支持TF卡、U盘两种脱机下载方式。TF卡/U盘容量要求不大于32G，TF卡/U盘必须格式化为FAT32或者FAT文件格式。

下载需要经过 2 个步骤：

- ✓ 首先是把下载文件夹复制到 TF 卡； 把 TF 卡读卡器插入电脑，刷新盘符并选择插入的 TF 卡，点击“下载到 U 盘”，就可以直接把下载文件夹拷贝到 TF 卡里，跳过复制、粘贴的繁琐过程，如下图。



图8-4-1 下载到U盘按钮

根据屏幕属性设置中产品系列设置的不同，会自动识别拷贝下载文件夹。

- ✓ 然后把装有下载文件夹的 TF 卡插入到串口屏上(TF 卡不支持热插拔，需要先插卡后上电)，串口屏将自动启动数据下载。下载成功后，显示屏会提示 3 秒文件下载成功并伴有蜂鸣器提示。下载失败时，会提示相关错误信息。下载工程文件或 Lua 脚本文件时，下载完成后将自动运行新的工程或 Lua 脚本。其他文件下载完成后，显示屏左上角提示"Please turn off power, plug out the SD disk and restart."，需要先断电，拔出 TF 卡再重新上电。

说明：

- ✓ 如果出现下载异常：
 1. 首先排查 TF 卡是否格式化为 FAT32 或者 FAT 文件格式；
 2. 其次排查 TF 卡里面是否混入不同产品类型的工程文件；
 3. 由于 TF 卡不支持热插拔，需要先插卡后上电；
 4. 然后排查屏幕属性设置是否正确，检查产品系列和分辨率设置是否与所用串口屏一致，可以在“工具->项目属性设置”中重新设置产品系列、分辨率等屏幕属性；产品系列选择不同，编译生成的下载文件夹不同。必须正确选择产品系列，产生对应下载文件夹，并将其复制到 TF 卡里，否则无法下载。

SDWb 系列下载文件夹为“VT_SET_SDWb”，

SDWn 系列下载文件夹为“VT_SET_SDWn”，

VTb 系列下载文件夹为“VT_SET_VTb”，

VTc 系列下载文件夹为“VT_SET_VTc”。

如果以上都没有问题，可以尝试更换一只 TF 卡。

- ✓ 用户不能对下载文件夹内文件做任何修改。
- ✓ 下载文件夹名称后面用户可以追加字符如“VT_SET_SDWbxxxx”，便于管理版本和机型信息。
- ✓ SDWb 系列下载工程文件或 Lua 脚本文件时，下载成功后经过短暂的提示，将自动运行新的工程或 Lua 脚本，一定程度上避免频繁的插拔 TF 卡以及断电/上电，提高开发阶段的效率。

8.4.2 在线下载

SDWb(VTb)/VTc 串口屏支持串口在线下载工程文件、Lua 脚本文件、固件等。在开发调试阶段，使用 VGUS 开发工具自带的“串口调试工具 2”，通过串口在线下载工程文件和 Lua 脚本文件，可以避免频繁插拔 TF 卡，提高开发的效率。上位机软件和串口屏固件需使用 2023.05.28 及以后版本。

串口调试工具 2 打开方式：VGUS 开发工具→工具→串口调试工具→串口调试工具 2（新手推荐）。

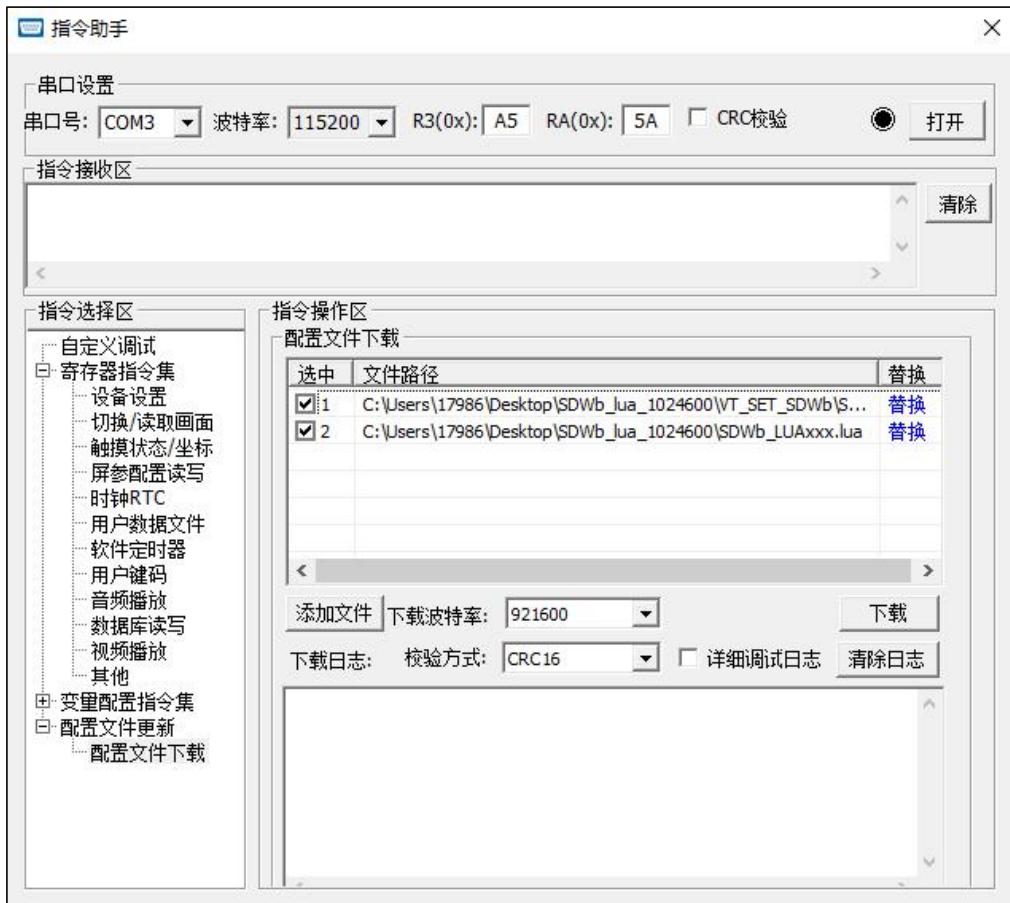


图 8-4-2 串口调试工具 2 界面

首先正确设置串口相关参数。正确设置串口号，上图中识别出来的串口号为 COM3。正确设置波特率、R3、RA、CRC 校验，这些参数要与下载到串口屏里界面工程的屏参配置一致。如果串口屏里下载了 Lua 脚本、并使用 Lua 脚本修改了波特率，需要使用 Lua 脚本中所设置的波特率一致。正确设置以上参数后，打开串口，确保与串口屏通讯正常。

然后在左侧指令选择区中选择“配置文件更新→配置文件下载”，添加所需下载的文件，点击下载，即可按照提示信息完成文件下载。

串口在线下载的文件传输协议开放，可结合用户主板，实现自主的文件自动更新功能。结合网络可实现远程文件更新功能。关于在线下载更加详细信息、以及下载协议，请参考文档《[串口在线下载文件功能说明](#)》。

8.4.3 追加下载

典型应用：如设备终端用户替换开机页面，而不用下载整个工程。追加下载一方面可以简化替换开机页面的过程，同时可以防止设备界面工程文件的泄露。为此，SDWb(20230530 及以后固件版本)和 VTc 串口屏专门设计了追加下载功能，在不修改工程文件的前提下，替换工程中的某个素材文件或添加新的素材文件。

追加下载文件存放在 Flash 内存中的界面工程文件之后。可用的追加下载文件存储空间大小与 Flash 内存大小、界面工程文件大小有关。

同一用途同一编号的追加文件比界面工程内的文件拥有更高的使用优先级，如追加文件 0.JPG 文件将替代工程中原有 0 号图片显示。

追加下载的文件在以下三种情况下会被清空：一是更新下载界面工程文件，二是通过 TF 卡重新下载追加文件，三是发送串口指令 **850504** 清空。

SDWb 系列支持追加下载 jpg 格式图片、ZIMG 格式图片（压缩 bmp 图片、需要通过位图转换工具转换）、图标（ICX 格式）、字库、音频、视频、用户数据等各类素材文件。在 TF 卡根目录创建名称为“SDWb_APPENDxxxx”的文件夹，将需要追加下载的文件复制到上述文件夹中，支持多个文件。“xxxx”为自定义的字符，名称长度不可超过 60 字节。

VTc 系列支持追加下载 RIMG 格式图片（压缩 bmp 图片、需要通过位图转换工具转换）、图标（ICX 格式）、字库、用户数据等素材文件。在 TF 卡根目录创建名称为“VTc_APPENDxxxx”的文件夹，将需要追加下载的文件复制到上述文件夹中，支持多个文件。“xxxx”为自定义的字符，名称长度不可超过 60 字节。

SDWb 系列和 VTc 系列均支持通过串口在线下载的方式下载“追加文件”。详见《[串口在线下载文件功能说明](#)》。



8.5 文件格式

表8-5-1 VGUS 屏支持的文件格式

文件格式	含义	编号限制
.Jpg、.bmp	图片文件	0-8191
*.ICO	图标库文件	0-255
.Jpg、.bmp、*.png	图标文件	0-1023
.DZK/.ADZK	用户字库文件	0-255
*.mp3	音频文件, 推荐	0-1023
*.wav	音频文件, 不推荐	0-1023
*.avi	视频文件	0-1023
*.bin	用户数据文件	0-255
UserDb.bin	用户数据库文件	固定
*.ttf	用户矢量字库文件	1-255

1 文件命名规则

上表中所有类型文件的文件名必须以阿拉伯数字开头。例如, 要把一副图片序号编为20, 图片文件可以命名为“20_测试.jpg”或者“20.jpg”, 但不能命名为“测试 20.jpg”。

2 图片文件格式

图片的大小要和屏的分辨率一致, 液晶屏分辨率在产品规格书中都可以查到, 如果下载与屏分辨率不匹配的图片会导致显示异常。所有图片要以数字命名(编号), 上电默认显示0号图片, 用户也可以在屏参配置中自定义设置。通过寄存器0x03-0x04以0x80指令直接调用、或者通过触摸按钮切换图片显示。

VGUS 开发工具中, 图片文件以树状形式管理, 如图 8-5-1 所示。通过鼠标右键可以实现图片的添加、删除、排序等操作。

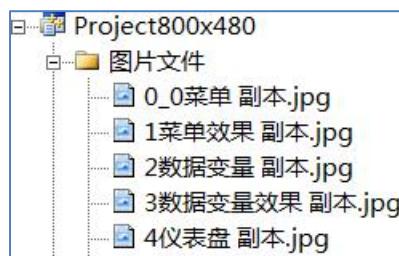


图 8-5-1 图片的管理

VGUS 屏支持 JPG、BMP 等图片格式。VTc、SDWn 系列建议使用 BMP 格式图片, 压缩效率更高。

表8-5-2 图片源文件格式汇总

产品系列	SDWn	SDWb	VTc
添加图片格式	bmp/jpg	bmp/jpg	bmp/jpg
系统存储格式	压缩 bmp	ZIMG 压缩位图/jpg	RIMG 压缩位图
说明	建议添加 bmp 图片, 系统存储时压缩效率高、不会有颜色损失。	建议添加 bmp 图片, 无损压缩存储, 颜色数据还原更准确。	建议添加 bmp 图片, 系统存储时压缩效率高、不会有颜色损失。

注: 1.添加图片格式是指在 VGUS 组态软件中添加的图片的源文件格式。

2.系统存储格式是指下载到串口屏中存储的图片数据的文件格式。

3.SDWb 系列, 使用 VGUS2022 开发工具编译带有 bmp 格式素材的工程, 串口屏需要 20220620 以后固件版本支持。

JPG 格式图片说明:

JPG格式种类繁多, VGUS屏仅支持Baseline 基线模式。当使用其它模式的JPG图片后, 可能会导致显示异常! 遇到图片不能显示的情况, 可能是图片格式不符合要求, 可以使用如下提到的两种方式对图片进行格式转换, 再添加到工程中。不要用画图、QQ截图等工具直接另存JPG格式文件, 这样显示效果较差, 因为JPG格式文件为有损压缩文件。

方法一: 建议使用VGUS2022开发工具的图片压缩工具处理。打开路径为:工具->图片压缩工具。该工具支持调整图片质量等级, 批量处理。简单高效完成图片处理。

方法二: 也可以使用专业工具, 将图片另存。使用PhotoShop另存为JPG格式, 具体步骤如下:

- 1、用PhotoShop打开图片;
- 2、另存为jpg格式, 如图8-5-2。



图8-5-2 另存图片品质选项



图8-5-3 BMP另存选项

- 3、选择图像效果“最佳”选项。

上述两种方法既可以输出VGUS串口屏可以识别的jpg图片格式, 又可以通过设置不同的图像质量来调整图片文件大小。

BMP 格式图片说明:

BMP格式为原始位图文件, 包含信息最全, 如果使用JPG格式文件显示效果不能满足要求, 用户可选择BMP格式文件。

建议美工直接输出16位深度RGB565格式的BMP图片。可提前避免因色彩深度不一致导致的图片在电脑和串口屏上显示效果差异。使用PhotoShop保存RGB565格式的BMP图片如图8-5-3所示。

如果图片上的渐变色较多, 可以尝试使用 VGUS2022 开发工具中的图片渐变色优化工具进行处理。打开路径为:工具->图片渐变色优化工具。

3 图标文件格式

图标其实就是小图片，所有图标都要以数字命名（编号）。图标支持 bmp、jpg、png 等格式。图标素材尺寸建议不要超过 255*255，过大尺寸的图标会影响显示速度。

表 8-5-3 图标源文件格式汇总

产品系列	SDWn	SDWb	VTc
添加图标格式	bmp	bmp/jpg/png	bmp/jpg/png
系统存储格式	压缩 bmp	压缩 bmp/jpg/png	压缩位图
透明显示	不支持	支持 ^[2]	支持 ^[6]
说明		bmp 图标显示效果好； jpg 图标占用存储空间少； png 图标透明显示效果好。	bmp 图标压缩效率较高； jpg 不建议使用； png 图标透明显示效果好。

注：1.选择图标左上角像素颜色为背景色，凡是颜色与背景色相同的像素都被滤除不显示。

2.SDWb 系列： bmp 格式的图标透明显示时自动滤除与左上角像素颜色相同的像素。

jpg 格式固定不支持透明。

png 格式根据文件本身透明属性显示，透明图标建议用 png 格式。

3.添加图片/图标格式是指在 VGUS 组态软件中添加的图片/图标的源文件格式。

4.系统存储格式是指下载到串口屏中存储的图片/图标数据的文件格式。

5.bmp/jpg 图标文件格式要求及处理方法参考上一小节“图片文件格式”。

6. VTc 系列： bmp/jpg 格式的图标透明显示时自动滤除与左上角像素颜色相同的像素。

png 格式根据文件本身透明属性显示，透明图标建议用 png 格式。

VGUS 开发工具中，图标以 ICO 文件形式管理，一个工程允许存在多个 ICO 文件，命名编号范围 0-255。一个 ICO 文件可以包含最多 1024 个图标，支持编号范围 0-1023。不同 ICO 文件中的图标编号可以相同。如图 8-5-4 所示。ICO 文件以及图标文件的新建、添加、删除、浏览等操作都直接通过鼠标右键完成。

图标的引用通过 ICO 文件+图标编号的形式，如图 8-5-5 所示。

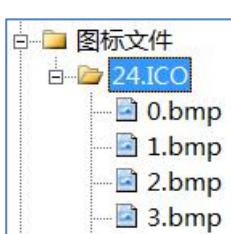


图 8-5-4 图标的管理



图 8-5-5 图标的引用

4 字库文件格式

VGUS 屏出厂预装有 0 号字库文件（0.dzk），包含 4*8 到 64*128 点阵的所有 ASCII 码字库。

SDWn/ SDWb/ VTc 系列对 0 号字库进行了剪裁，只有在控件中使用到的点阵才会下载到串口屏。

当需要用到其它类型字库时，可以找技术支持索取或者自行通过点阵字库生成工具生成相应字库文件。新字库需要从 1 开始命名，不要与 0 号字库重名。



点阵字库生成工具详细用法参考文档《[ADZK 点阵字库文件生成工具使用说明](#)》。

字库编码支持 ASCII 码、GB2312 内码、GBK、BIG5、SJIS、UNICODE、UTF8。相关文档参见《[VGUS 串口屏字库编码与点阵字库生成工具](#)》。

SDWb/ VTc 系列支持 ADZK 格式字库。支持字体抗锯齿，并且采用压缩方式存储，存储效率高。

TTF 矢量字库支持：

1. 仅 VTc/VTb 系列 20251020 及以后固件版本支持。
2. 文件命名。必须以阿拉伯数字开头，编号范围 1-255。如 20.ttf、21-fangsong.ttf。
3. 字库导入。在文件录入树中，右键“字库文件”点击“添加”对应的文件即可导入到工程。
4. 字库引用。通过各类控件的“FONT0 ID”或“FONT1 ID”属性，选择对应的字库文件即可完成该字库的引用。
5. 字体大小。通过各类控件的“Y 方向点阵数”属性设置所需显示的字体大小。
6. 特别注意：
 - 6.1 对于数字显示类控件，通过“FONT0 ID”引用字库，需保证字库中有 ASCII 编码字符。
 - 6.2 对于文本类控件，使用矢量字库时“FONT0 ID”或“FONT1 ID”仅需设置其中一个即可，如果一个设置为点阵字库，一个设置为矢量字库，那么使用矢量字库。如果两个均设置为矢量字库，那么仅使用“FONT1 ID”对应的矢量字库。
 - 6.3 对于文本类控件，使用矢量字库时“FONT1 编码方式”属性仅支持 UNICODE、UTF8 以及 UTF8-2 这三种编码方式，否则显示异常。
 - 6.4 矢量字库已包含每个字符独有的宽度信息。使用矢量字库时各类控件的“字符宽度自动调整”属性设置无效。

5 音频文件格式

SDWb 串口屏支持 WAV、MP3 两种音频文件格式。支持两种播放方式：扩展指令 0x8503 和寄存器 0x50-0x55，详细信息分别参见本文 2.4 节、3.3 节以及文档《[音频播放使用说明及案例](#)》。

WAV 格式文件必须选用 22KHz、单声道数据。寄存器 0x50-0x55 播放时单个 WAV 文件不得超过 2M 字节，扩展指令 8503 播放时播放列表的 WAV 文件总和不得大于 2M 字节。

6 视频文件格式

SDWb 串口屏支持 AVI 格式视频文件播放。详细信息参见本文 3.4 节、4.5 节和文档《[VGUS 串口屏视频播放功能详解](#)》。



附 1 系统自检寄存器

系统自检的启动方式：

方式1：读取寄存器0xf3-0xf6，系统自动启动一次所有文件的校验；

方式2：系统执行程序时会实时检查使用到的文件。

系统自检的结构输出方式：

当校验到文件错误信息后，会将错误代码写入寄存器0xf3-0xf6中，并通过以下方式输出：

在屏幕的左上角显示错误代码。

可通过“屏参配置”中的“miscellaneous.4”设置是否显示错误代码，详见图8-2-1和表8-2-2。

勾选“miscellaneous.4”则启用屏幕显示错误代码信息。

如果不希望在最终产品上显示错误代码信息，可以不勾选“miscellaneous.4位”选项。

表附2-1 系统自检错误代码

(0xf3 0xf4)	错误代码 (0xf5 0xf6)	描述
0000	0x0001	SPI 初始化异常
0000	0x0002	系统参数 sysinitial 异常
0000	0x0003	读取 Config 异常
BIN_ID	0x0004	用户数据文件异常
WAV_ID	0x0005	Wav 文件异常
Mp3_ID	0x0006	Mp3 文件异常
AVI_ID	0x0007	AVI 文件异常
FT0_ID	0x0008	0 号字库异常
V22_ID	0X0009	变量初始化文件异常
UNICODE_ID	0x000a	UNICODE 字库异常
FT_ASC_ID	0x000b	用户 ASCII 码字库异常
FT_GB_ID	0x000c	用户全角字库异常
ICON_ID[7:0], ICON_NUM[7:0]	ICON_ID[11:8], 0x0d	图标文件异常
PIC_ID	0x000e	图片文件异常
0000	0x0010	工程校验错误，读 F3 寄存器启动并返回工程自检结果
0000	0x0011	串口接收到异常指令
0000	0x0012	控件内存使用率大于 90%
0000	0x0013	页面异常
0000	0x0014	ttf 矢量字库异常

附 2 固件更新

在固件更新过程中，请保证供电稳定！

在固件更新过程中，显示屏会自动重新启动，期间切不可断电！

使用TF卡更新固件：

1. 将固件复制到 TF 卡根目录下；
2. TF 卡不支持热插拔，需要先给屏断电，插 TF 卡，然后上电。上电后自动开始更新；

下图表示正在更新固件。



下图表示固件更新成功，将自动重新启动运行新固件，请保持供电稳定。



重启后，由于固件刚更新过，所以提示不需要更新。



出现以下提示，表示已更新完成，同时会有蜂鸣器提示音。此时应断电，拔卡然后重新上电。



使用串口更新固件（仅SDWb/ VTc系列支持）可参考本文8.4.2节内容。

固件与工程一起更新下载的方法：

对于批量用户需要将固件更新到指定日期的情况，用户可以将指定固件文件放到TF根目录下，随同用户界面工程一起下载到串口屏里，避免重复下载操作。