深圳市拓普微科技开发有限公司
Shenzhen TOPWAY Technology Co., Ltd.

TOPWAY
智能模块
SGTools 使用手册

快 速 快速实现界面设计

节 省 节省主板硬件资源

可 靠 抗干扰能力更强

快速实现界面显示

公司简介

Company Profile

拓普微成立于1996年，专注于工业类液晶显示模块的设计和生产。

公司座落于深圳市南山区政府规划建设的高科技产业园区内，位于风景秀丽的西丽水库旁，拥有六千平方米全新工业厂房，以及从日本引进的COB，TAB和ACF等先进的生产设备，和完善的ISO9001质量管理体系。公司于2008年被深圳市科技和信息局认证为深圳市高新技术企业。

公司的核心是一群在液晶显示领域工作多年，并曾任职于液晶行业知名企业的工程师。我们的专业团队潜心研究液晶显示新技术新应用，依靠多年的行业积累和技术优势，以国内著名高校雄厚的科研力量与人才资源为持续发展的后盾，与国内外相关企业展开广泛合作，以可靠的产品设计，稳定的产品质量，及时的产品交货赢得市场的认可。

拓普微有着广泛的信息渠道，与世界上许多电子设计公司有着紧密的联系，通过及时深入地了解国内外LCD技术发展以及相关应用产品的市场信息，准确地掌握产品的发展趋势，在设计上不断创新，使我们提供的产品适合不断变化的市场需要。

经过多年的市场拓展，拓普微在工业类液晶显示模块市场已成为知名品牌，是日本、台湾品牌的强有力竞争对手，并已成功进入国际市场多年，产品远销世界各地，主要客户包括艾默生、东芝、通用电气等国际知名企业。

Established in 1996, Topway is a high-tech enterprise specializing in the design and manufacturing of industrial LCD Module.

Topway are located in High-tech. Industrial Park constructed by local government, with over 6000 square meters space, our production plant has stringent quality control to guarantee absolute product quality. Our production equipment are state-of-the-art Japanese technology including COB, TAB, COG, etc.

Our founding members are teams of professional engineers whose careers have long been associated with LCD industry. With such solid experience and technology know-how, and a very close and long standing cooperative relationship with universities and institutions where in-depth technology research, these high-power LCD Module specialists have a strong sense of duty and passion to deliver good products to customers. Topway have been approved by **Reliable Design, Stable Quality** and **In-time Delivery** in the market.

Topway are creating value for our clients with leading and innovating design. Our up-to-date marketing intelligence of LCD technology and related applications, extensive networking with worldwide
product design houses and component makers, all ensure our LCD Module as technology superior and market welcome.

Being a customer-focused company, Topway has gradually gained a strong foothold in the world as a company excels in **Quality**, **Service** and **Innovation**. Well received by the global market, our clients included renowned names like Emerson, Toshiba, GE, etc.

LCD Module 专业厂商
稳定 可靠

SYSTEM CERTIFICATION
ISO 9001:2008 SGS






目录

1 快速开发

1.1 开发流程

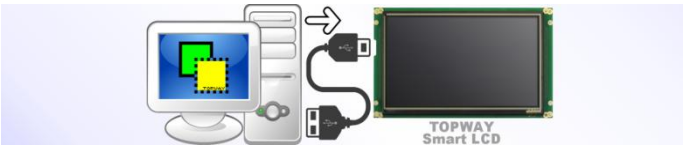
1 安装 SGTools

注: SGTools 支持 Windows XP, Window 7/8/10/11 (需管理员权限)

2 制作界面

注：导入的图片,建议使用 BMP(24bit)格式

3 下载界面到模块



注：使用 USB(A)-USB(mini)连接线．注意线缆质量及长度(建议不要超过 1.5m)保证供电与信号稳定

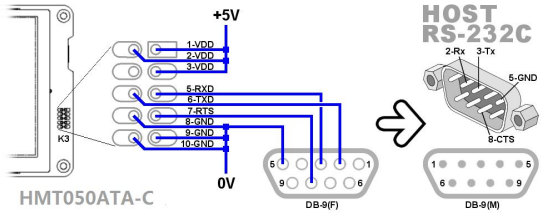
4 供电预览界面

注：供电前请参考用户手册说明注意极性与电压!.

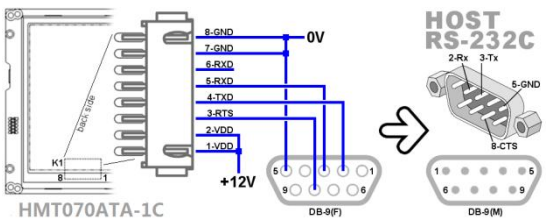
5 连接客户主机
实时显示数据

注：模块的信号接口为 RS-232C(部分模块提供 UART(TTL)电平信号),信号地与电源共地.

1.2 连接向导



HMT050ATA-C 连接示例



HMT070ATA-1C 连接示例

1.3 快速应用

请参考“应用案例”章节。

2 产品概述

TOPWAY 智能模块(Smart LCD)是专为工业显示应用而设计的智能 TFT 液晶模块。智能模块自身带有 MCU/Flash/RAM/RTC 时钟等硬件和实时组态软件系统(显示驱动程序、触摸驱动程序)，客户的图片、字库、配置文件等数据都可存储模块中， 极大的减轻客户主机配置与负荷要求， 从而可降低系统硬件成本、设计成本。
客户可使用我司提供的"界面编辑软件"(名称:TOPWAY SGTools[1],简称:SGTools)通过组态式拖拽式搭建用户界面方式来快速实现用户界面设计，设计界面过程中不需要写任何代码，大大降低开发难度、节省开发时间， 缩短产品上市周期。

2.1 产品框架

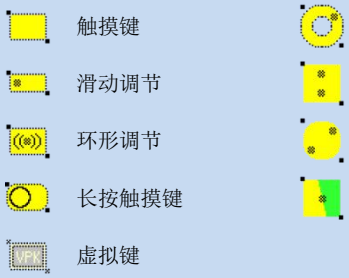
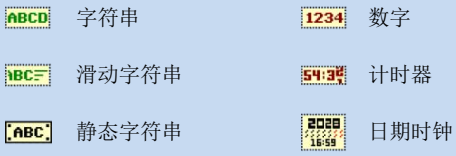
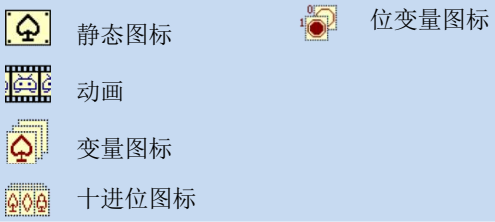
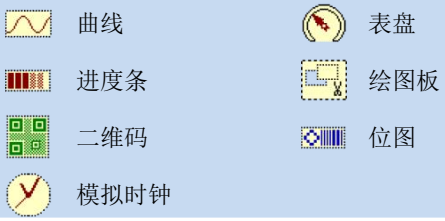
- 内置实时组态式 RTOS 系统，图像显示及触摸操作模块自主处理
- 内置 256MB flash 数据存储空间,可存储 1000 幅界面
- 标准 RS-232C/ UART/RS-485/RJ45 通信接口用于同客户主机通信
- 标准 Mini USB / Type-C 接口用于下载/更新界面
- 简单可靠的通信指令封包
- 组态式的界面发开方式，功能丰富、简单易用
- 所有要显示图像数据提前预存在模块内，发命令可直接调用显示
- 支持 Lua 脚本程序，可实现更灵活的 UI 功能效果

2.2 产品特点

快速	通过 TOPWAY SGTools 开发工具，用户可以组态式搭建 UI 界面，无需任何编程。 可快速完成界面设计。
节省	软件上模块自带显示处理程序和触摸程序，模组自主完成显示和触摸功能 硬件上自带 Flash 存储，用户可以把图片、字库、配置文件等存储在模块中 模块对外通信的接口为简单的串口（RS232/UART/RS485 等） 对用户主板硬件资源要求非常低，用户的主板只需要有串口就可驱动屏幕实现彩色显示。 大大节省主板硬件资源
可靠	一体化结构设计，抗干扰能力更强

2.3 产品功能

SGTools 提供了 4 大类简单实用的控件，通过这些控件的组合能实现丰富的显示效果和功能.

<table>
<tr><td rowspan="2">基础控件</td><td>触控类控件
 
触摸键
滑动调节
环形调节
长按触摸键
虚拟键
环形调节
双指滑动
双指旋转
滑动翻页</td><td>字符/数字显示类控件
 
字符串
滑动字符串
静态字符串
数字
计时器
日期时钟</td></tr>
<tr><td>图像显示类控件
 
静态图标
动画
变量图标
十进位图标
位变量图标</td><td>图形显示类控件
 
曲线
进度条
二维码
模拟时钟
表盘
绘图板
位图</td></tr>
<tr><td>系统功能</td><td colspan="2">Unicode(UTF-8)字符集 (矢量 TTF 字库)
ASCII 字库(提供了 29 个国家字符集编码，如英文、法文、俄文、阿拉伯文等)
DBCS(双字节编码字库: 简体中文 GB2312/GBK、繁体中文 BIG5、日文、韩文)
多国语言一键切换功能
自定义键盘、菜单、弹窗功能
模块加锁功能</td></tr>
<tr><td>通信指令</td><td colspan="2">配置/状态读写指令
显示控制指令
变量 VP 读写指令</td></tr>
<tr><td>扩展功能</td><td colspan="2">支持 lua 脚本程序</td></tr>
</table>

注:
- 详细的控件应用,可参考视频教程
- 详细的通信指令,可参考 user manual
- 详细的脚本 LUA 函数定义，参考 lua manual

3.2 控件列表

3.2.1 响应类型控件

图标	简称	名称	描述
	TPK	触摸键 Touch Key	触摸键是定义于页面中一个矩形区域，在此区域点击触摸屏会触发操作. - 相关区域可在画面中提供显示”反应” (如颜色反转，显示 icon 等) - 可向主机器提供相关触发反馈(如: Page_ID/Key_ID, VP 地址/值 等) - 同时 b 呼叫功能/动作(如:弹出键盘，VP 运算操作，页面转跳等)
 (*1)	SDR	滑动调节 Slider	滑动调节是定义于页面中一个矩形区域，在此区域内水平或垂直单指滑动调节， 根据位置偏移或比例，按照设定转换为相应的数值。
 (*2)	RNG	环形调节 Ring	环形调节是定义于页面中一个环形区域，在此区域内单指环形调节， 根据角度位置偏移，按照设定转换为相应的数值。
 (*3)	TPK RPT	长按触摸键 TPK_Repeat	长按触摸键是定义于页面中一个矩形区域，在此区域内长时间按下， 会周期性执行变量加减的动作。若短按，则每次仅执行一次。
 (*4)	TPK SW	开关触摸键 TPK_Switch	开关触摸键是定义于页面中一个矩形区域，在此区域内按压一次后， 会将变量值的某一位进行取反操作。如: 设置第 0 位控制位，若 value = 0x01,按压一次后,value = 0x00。
 (*5)	SWP PG	滑动翻页 Swap_Page	当单指在屏内水平移动超过 10 个像素，则触发。 当移动方向上超过指定距离，释放后，切换至相应指定页面。
 (*6)	SDR2	双指滑动 Slider_2	将双指在屏内水平或垂直滑动，根据位置偏移，按照设定的比例系数，转换为相应的数值。
 (*7)	RNG2	双指旋转 Ring_2	将双指在屏内环形调节，根据角度位置偏移，按照设定的比例系数，转换为相应的数值。
 (*8)	TPK CH-EN	中英切换 TPK_CHEN	中英切换是定义于页面中一个矩形区域，在此区域内按压一次后， PIP 中文键盘将切换到英文输入法模式，并显示相应的图标。 再次点击则切换为中文模式，并取消图标显示。
	VPK	虚拟键 Virtual Key	虚拟键为非显示控件，满足条件(监控 VP 值与监控值相同时)可被触发 - 同时可呼叫功能/动作(如 弹出键盘，VP 运算操作，页面转跳等) - 相关被监控的 VP 值变为 0 (注: 只有相关页面在显示时，此页面中的 VPK 满足条件时才被触发)

Note.
*1、*2、*3、*4、*5、*6、*7、*8 仅部分型号支持.

3.2.2 字符数字类型控件

图标	简称	名称	描述
	STR	字符串 String Variable	字符串控件可提供字符串变量的显示 - 当字符串变量的内容被更新，相关的内容将会实时更新于画面中
	STR-SRL	滚动字符串(*1) String Stroll	字符串控件可提供字符串变量的滚动显示 - 当字符串变量的内容被更新，相关的内容将会实时更新于画面中
	N16 N32 N64	数字 Number Variable	数字控件可提供数字变量的显示 - 当数字变量的值被更新，相关的内容将会实时更新于画面中
	STS	静态文本 Static String	静态文本可提供静态字符串内容显示于页面中 (建议静态内容固定于背景图中，以达到更灵活多变的显示效果)
	TMR	计时器 Timer	计时器控件按 Timer VP(Timer0~Timer7)的数值照设定的显示格式显示出来 Timer0 ~ Timer7 可通过对应的 Timer 寄存器进行控制开启、关闭、暂停等功能 写 VP_N32 地址指令可置 Timer VP 初值
	RTC	日期时钟 Real Time Clock	日期时钟用来显示模块内部 RTC 的内容. (如:日期，时-分-秒)

Note.
*1 仅部分型号支持.

3.2.3 图像类型控件

图标	简称	名称	描述
	ICO	静态图标 Static Icon	静态图标可提供静图标显示于页面中 (建议静态内容固定于背景图中，以达到更灵活多变的显示效果)
	ANI	动画 Animation	动画控件可把已导入的动画显示于画面中 动画显示速度可于控件属性中定义
	BitIcon	位变量图标 Bit Icon	位变量图标控件是由 VP_N16 或 VP_N32 地址的指定的 bit 位来控制显示的控件 必须选定 bit=0 时和 bit=1 时的图标
	IDX	变量图标 Indexed Icon	变量图标控件是一个可按 VP_N16 或 VP_N32 变量值而显示不同图标的控件 当 VP_N16 或 VP_N32 变量值超过设定的极限值，将会无图标显示.
	CLK	模拟时钟(*1) Round Clock	模拟时钟用来显示模块内部 RTC 的内容。（时-分-秒）
	HND	表盘(*2) Tachometer	指针表盘由刻度盘和指针组成，可设定宽度、颜色、指针类型等。 通过修改 VP 内容可控制表盘指针的显示位置。(显示精度为 0.1 度)
	R32	表盘(*3) Tachometer	以环形填充角度变化来表示数值变化， 通过修改 VP 内容可控制环形填充的显示位置。(显示精度为 0.1 度)
	TCM	表盘 Tachometer	两张图片通过角度旋转合成表盘的各个显示状态. 通过控制 VP 的内容可控制显示哪个状态的图片
	I16 I32	十进位图标 Decimal Icon	十进位图标是一个可按 VP_N16/VP_N32 变量值而显示十进位图标的控件 此控件必须要结合 12 个图标工作 (图标顺序为: 数字 0~9，小数点，负号)

Note.
*1、*2、*3 仅部分型号支持.

3.2.4 图形绘制类型控件

图标	简称	名称	描述
	B16	进度条 Progress Bar	进度条可按 VP_N16/VP_N32 变量值显示不同的填充方块长度 也可以按 VP_N16/VP_N32 的值在显示一个 ICON
	G16	曲线 Graph	曲线控件可提供曲线变量的显示 - 当曲线变量数组中的内容被更新，相关的曲线将会实时更新于画面中
	BP1	位图 Bitmap	位图控件可提供位图变量的显示 - 位图变量的内容为 1bpp - 此控件可定义 1 的显示颜色与 0 的显示颜色 - 当位图变量数组中的内容被更新,相关的位图将会实时更新于画面中
	QRCode	二维码 QR Code	二维码控件可把 VP 变量中的数据以二维码形式显示出来
	DPD	绘图板 Draw Pad	绘图板区域内可通过命令直接绘制点、线、矩形、图标、文本等.

3.3 页面、图像资源和 VP 变量

3.3.1 页面

图标	简称	名称	描述
	PAGE	页面 Page	每个页面就对应用户产品的一个个显示界面； 页面中可放置控件和设置一幅背景图做为底图(或背景色做为底色)

3.3.2 图像资源

以下 4 中都是需要用户导入的图片资源，导入后可以被页面、图像显示类控件调用。

图标	简称	名称	描述
	IMG_BKG	背景图 Background Image	在 SGTools 中导入为背景图后，主要被用作为页面的背景.
	IMG_ICO	图标 Icon	图片一般小于背景图，可导入格式为 24 位 BMP。 在 SGTools 中导入为图标后，在页面中通过图像显示类控件(静态图标、变量图标等)调用后显示。
	IMG_IcoLib	图标库 Icon Library	图片一般小于背景图，可导入格式为 PNG。 在 SGTools 中导入为图标后，在页面中通过图像显示类控件(静态图标、变量图标等)调用后显示。
	IMG_ANI	动画 Animation	动画是由多个图片组成。 可导入一个个 24 位 BMP 图片，或者导入 GIF 格式文件。 在 SGTools 的页面中可被 "动画控件" 调用。

3.3.3 变量 VP

变量 VP 就是指向一个存储数据的地址(或理解为寄存器），不同类型的变量 VP 可存储数据的大小不同；

图标	简称	名称	描述
	VP_STR	字符串变量 String Variable	字符串变量单位长度为 128 字节,用于存储字符串数据(字符串结尾必须以'\0'结束) 可存储 127 个 ASCII 字符或 63 个汉字.
	VP_N16	16 位数字变量 16bit Integer Variable	16 位数字变量单位长度为 2 字节，用于存储整型数值数据 有符号整型: -32768 ~ 32767 无符号整型: 0 ~ 65535
	VP_N32	32 位数字变量 32bit Integer Variable	32 位数字变量单位长度为 4 字节，用于存储整型数值数据、浮点数(float)数据 有符号整型: -2147483648 ~ 2147483647 无符号整型: 0 ~ 4294967295
	VP_N64	64 位数字变量 64bit Integer Variable	64 位数字变量单位长度为 8 字节，用于存储超长整型数值数据 有符号整型: -9223372036854775808~-9223372036854775807 无符号整型: 0 ~ 18446744073709551615
	VP_G16	曲线变量 16bit Graph Variable	16 位曲线变量为一数组，单位长度为 2 字节，用于存储 16 位曲线每个点的数值 有符号整型: -32768 ~ 32767

	VP_BP1	位图变量 Bitmap Variable	位图变量用于存储二值图像数据(每个 bit 位代表一个像素点) 图像数据可通过取模软件生成.
	VP_SYS	系统寄存器变量 System Register	系统寄存器变量单位长度为 1 个字节, 用于控制工程计时器、蜂鸣器、背光等参数 无符号整型: 0 ~ 255

注：变量 VP 中的数据断电不保存.

4 开发工具(SGTools)

4.1 SGTools 工作界面介绍

SGTools 软件由 TOPWAY 开发设计, 用于智能模块产品的界面设计。该软件界面友好、操作简单、运行稳定,且支持中英文语言可切换, 满足国内外不同语言的客户使用。

软件主框架由 6 个部分组成, 见下图：

①菜单栏　　提供常用功能的功能入口(如: 新建工程、保存工程、调用参数功能等)

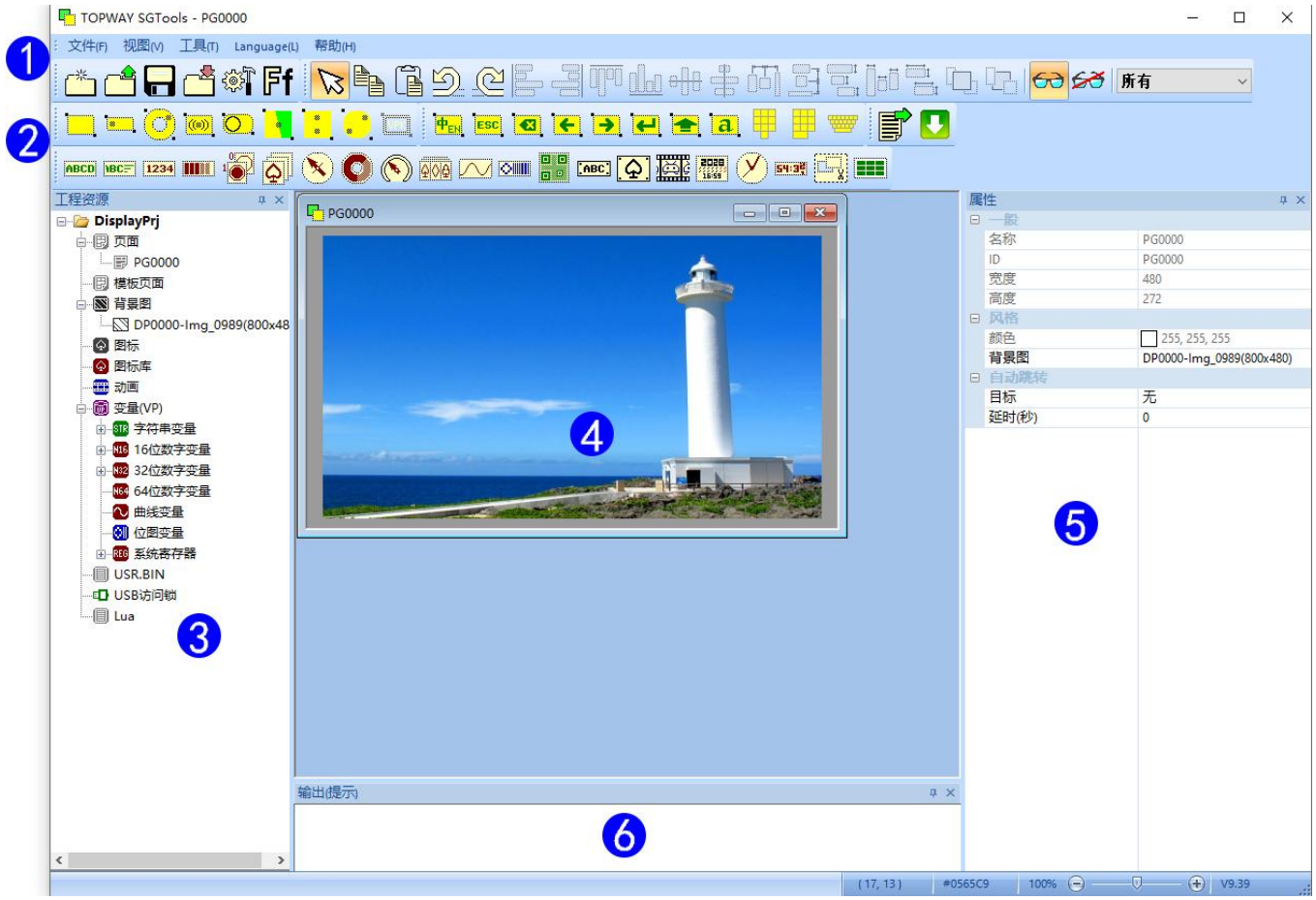
②工具栏　　提供各个功能控件、操作控件、工程编译、工程下载功能键

③资源窗口　　提供管理与显示工程中各种资源(如: 页面、背景图、图标、动画、VP 变量等)

④页面工作区　　页面工作区页面 UI 设计区域, 提供对页面和控件的编辑

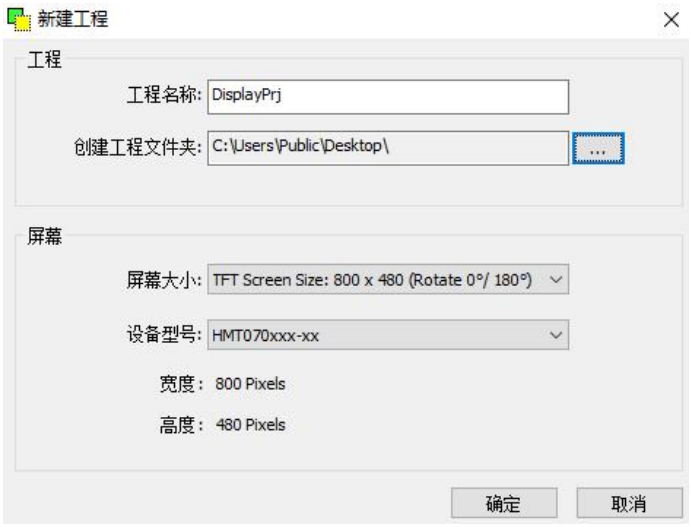
⑤属性窗口　　提供对页面、控件的属性进行参数设定

⑥输出(提示)窗口　　输出窗口主要是显示输出操作和编译工程的过程信息, 如错误/警告信息等.



4.2 SGTools 窗口设置

4.2.1 新建工程



新建工程窗口

工程名称

输入新建工程的名称

创建工程文件夹

新建工程存的储路径

屏幕大小

选择一个工程分辨率(要与屏幕实际分辨率相同)

Rotate 0°/180° 表示水平(横屏)显示

Rotate 90°/270° 表示垂直(竖屏)显示

设备型号

选择一个具体的屏幕型号

比如 7 寸屏幕，当未找到具体型号时，

可选择 HMT070xxx-xx

4.2.2 编译器设置

编辑器设置

用于设定 SGTools 编译时的参数

图像伽马转换

- 调整背景图的图像伽马(Gamma)值

编译选项

- 设置图像的伽马值
- 设置是否生成 U 盘升级的镜像文件
  (U 盘升级方法，见附录 D: 下载工程包方法)
- 设置编译完成后是否弹出编译后工程包文件目录
- 设置下载前是否编译工程
- 设置编译前是否保存工程
- 设置是否编译未用到的背景图像

4.2.3 工程设置

设置工程默认基本参数

- 设置开机启动”首页面”
- 设置背光亮度
- 设置屏保护模式
- 设置 RTC 模式
- 设置蜂鸣器鸣叫时长(单位 ms)
- 设置字库”国家码-Country”与”代码页-Codepage”
- 设置波特率
- 设置使能 ACK
- 设置锁定触摸
- 设置屏幕旋转
- 设置工程设备型号
- 设置命令格式
- 设置命令超时
- 设置分区 2 大小(单位 MB)
- 设置 U 盘更新模式（仅部分型号支持）

工程设置-参数详细说明

1. 屏保模式

屏幕保护

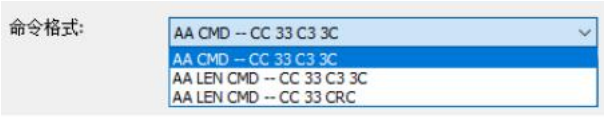
屏幕支持 5 种屏保模式
- 延时 1->设置背光
- 延时 1->设置背光->页面跳转
- 延时 1->设置背光->延时 2->页面跳转
- 延时 1->页面跳转
- 延时 1->页面跳转->延时 2->设置背光

2. RTC 模式

RTC 模式

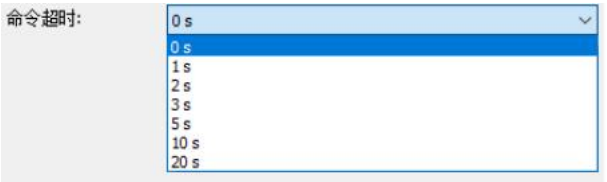
屏幕支持 3 种 RTC 模式
- 使能: 开启 RTC 功能(需安装纽扣电池, 时间误差低) (*1)
- 禁止: 禁用 RTC 功能, 日期时钟控件无效。
- 工程模式: 开启 RTC 功能 (时间误差高)

3. 命令格式、命令超时



命令格式:

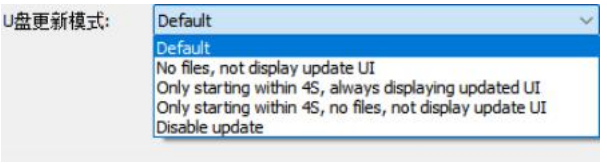
屏幕提供了 3 种通信指令封包格式供选择:
- 普通指令格式(AA CMD -- CC 33 C3 3C)
- 带长度指令格式(AA LEN CMD -- CC 33 C3 3C)
- 带长度+CRC 指令格式(AA LEN CMD -- CC 33 CRC)



命令超时:

屏幕接收 1 帧指令，若超出设定时间，会丢弃该帧指令。
0s : 不做超时处理;
1s : 接收 1 帧指令超过 1 秒，就丢弃该帧指令。
:
20s : 接收 1 帧指令超过 20 秒，就丢弃该帧指令。

4. U 盘更新模式



U 盘更新模式:

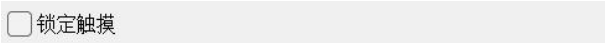
- Default
  任意时刻，只要插入 U 盘，就进入升级界面。
- No files, not display update UI
  任意时刻，只要 U 盘中有工程文件,就会进入升级界面,若无则不进入。
- Only starting within 4S,always displaying updated UI
  仅屏幕上电前 4 秒内，只要插入 U 盘，就进入升级界面。
- Only starting within 4S, no files, not display update UI
  仅屏幕上电前 4 秒内，只要 U 盘中有工程文件,就会进入升级面,若无则不进入。
- Disable update
  禁用 U 盘升级功能。

5. 使能 ACK

使能 ACK

屏幕指令执行结果 是否回复 使能标志。
☑勾选：客户主板给屏幕发送任意指令，
若屏幕执行成功，则屏幕会给主板返回一条固定指令：3A 3E
若屏幕执行失败，则屏幕会给主板返回一条固定指令：21 3E
☐不勾选：屏幕不返回指令执行成功或失败的状态。

6. 锁定触摸



锁定触摸

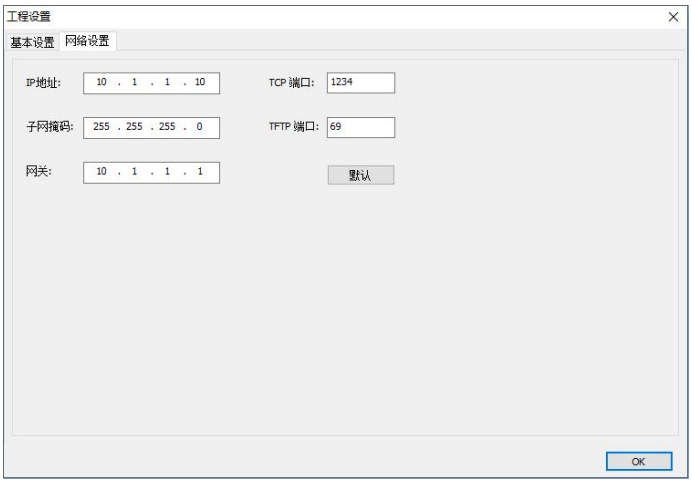
用于处理触摸键松开时不在原按下位置的情况
☑锁定触摸:
若按下了 A 触摸键，无论在任何位置上松开，屏幕执行 A 触摸键功能。
☐锁定触摸:
若按下了 A 触摸键，若滑动到 B 触摸键上松开，屏幕执行 B 触摸键功能。
若移动到非触摸区域松开，屏幕不执行 A 触摸键功能。

注:

*1. RTC 模式"使能"补充说明:
部分型号 RTC 模式选择“使能”后，若不装电池(或电池电压不够),模块每次上电时会进入几秒中的初始化界面。详见下表各种情况:

RTC 模式	上电时间(带电池)	上电时间(不带电池)	描述
使能	第一次上电需约 3-6s; 以后每次上电约~1s	每次上电约耗时 6s	初始化 RTC,并开启运行 RTC
禁止	每次上电都需约 1s	每次上电都约 1s	RTC 停止运行
工程模式	每次上电都需约 1s	每次上电都约 1s	RTC 运行(不做初始化, 时钟有机会运行不准确) 不建议设置此参数

4.2.4 网络设置



网络设置

设置屏幕 TCP/IP 通信参数

IP 地址: 屏幕的 IP 地址

TCP 端口: 设置 TCP 端口号

子网掩码:设置子网掩码

网关: 设置网关

TFTP 端口: 设置 TFTP 协议端口号

注：仅部分支持网口功能的型号，才可以网络设置；

4.2.5 字库设置

有三种字库可供设定使用.

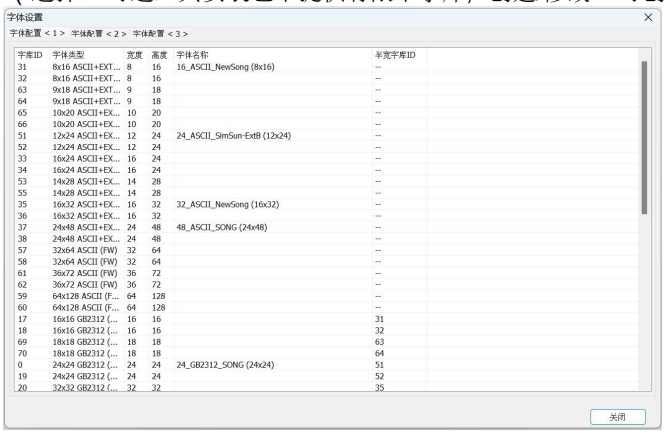
字库配置<1>,字体的宽高是固定的,提供了多种常用的字库. 如: ASCII、GB2312、GBK、特定 NUM 数字字库.

字库配置<2>,字体的宽高是可设定的自由大小字库, 如: ASCII、GB2312、GBK、繁体中文、韩文、日文等.

字库配置<3>,支持 Unicode-UTF8 编码的 TTF 矢量字库。

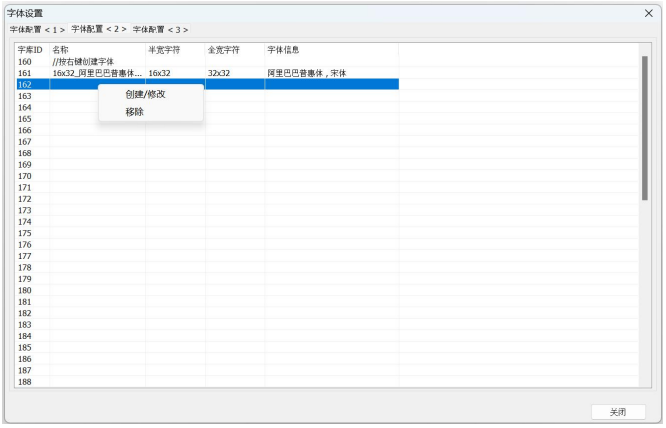
选中字库 ID 的某一行，右击鼠标可 选择、创建、导入新字库；

("选择": 可选工具安装包中提供有限中字库; "创建/修改": 可创建任意字体及大小的字库; "移除": 删除当前选中的字库)

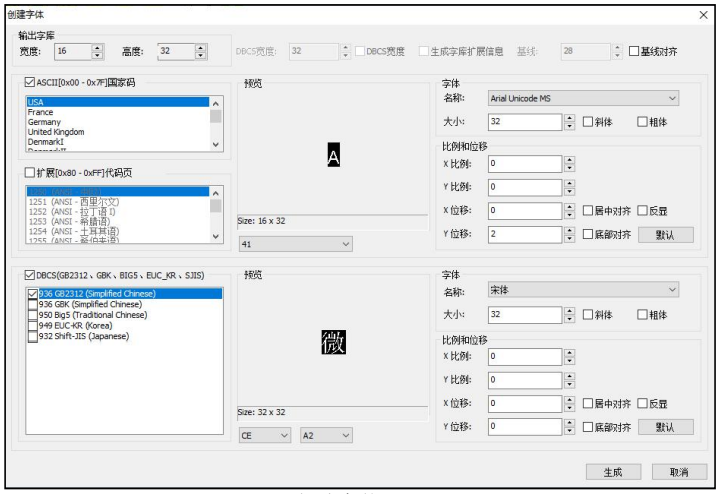


字库配置<1>

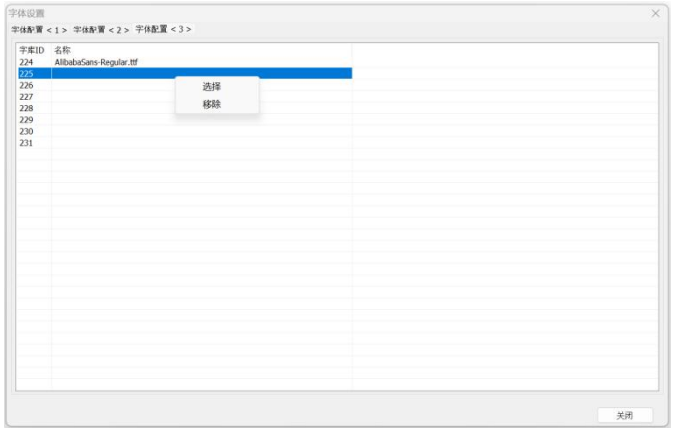
创建字体<1>



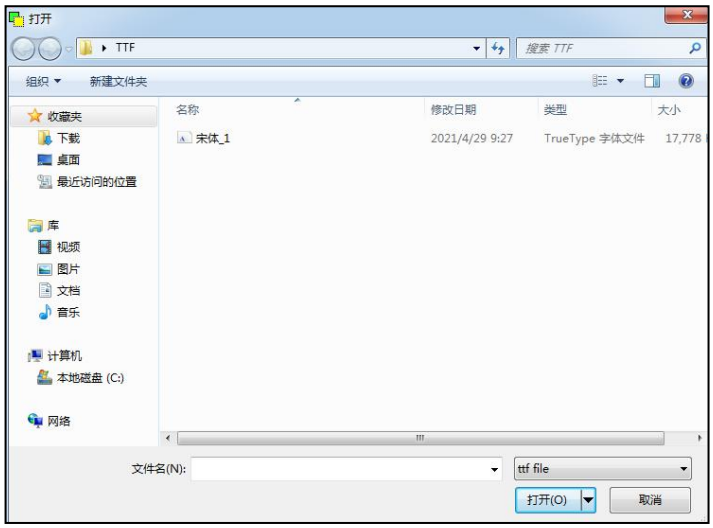
字库配置<2>



创建字体<2>



字库配置<3>



选择 TTF 字体

4.2.6 工程编译

界面设计完成后点击工具栏"编译"项，会输出模块可识别的文件(输出文件包含在"FONT"和"THMT"文件夹内).
可通过"量产工程工具"或直接拷贝 FONT 和 THMT 到模块中完成工程的下载.

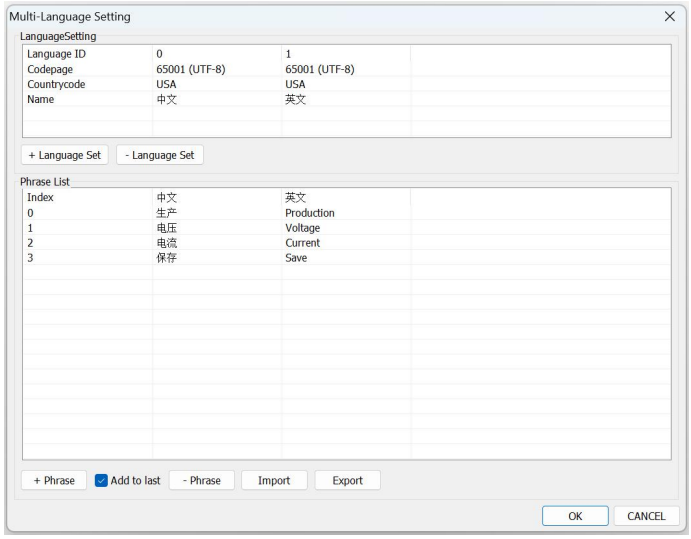
4.2.7 下载工程

下载工程

模块连接电脑后，在工具栏中点击此按钮，
SGTools 会自动查找已连接的智能模块设备。
点"开始下载"后自动把工程包下载到模块存储器中.

注. 下载工程过程中模块中的数据会被全部清除.

4.2.8 多语言设定

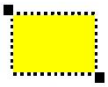


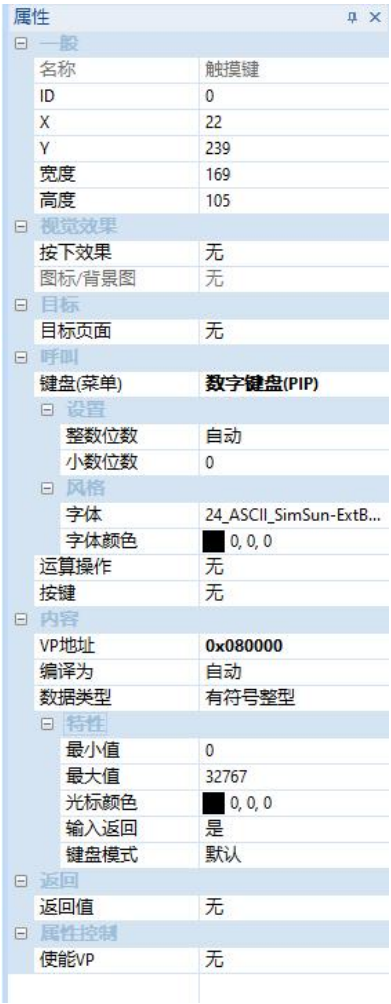
多国语言文本设定
Language ID： 语言 ID 编号
Codepage： 字符集编码（建议选 UTF-8）
Countrycode： 国家码
Name： 语言 ID 编码对应的语言名字
+Language Set： 增加语言
-Language Set： 删除语言

Phrase List (配置多国语言文本对照表)
+Phrase： 添加一行文本
-Phrase： 删除一行文本(需提前选中要删除的文本)
Import： 导入多国语言对照表(.csv 格式)
Export： 导出多国语言对照表(.csv 格式)
Add to last： 勾选后，添加一行文本时，默认添加到最后一行

4.3 基础控件

4.3.1 触摸键

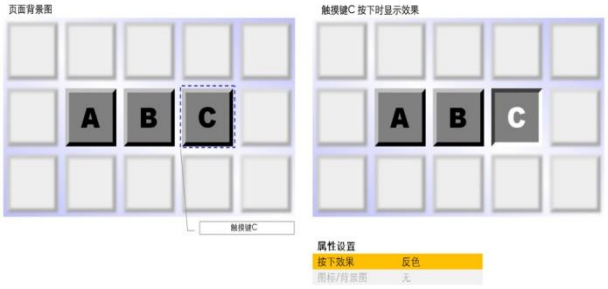




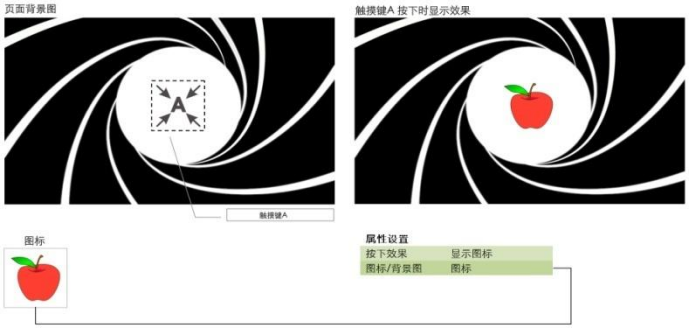
名称	触摸键(Touch key)
ID	编号
X/Y/宽/高	按键有效区域(页面左上角坐标为 0,0)
按下效果	反色：反色按键区域 显示图标：按键区域内显示一个图标 显示剪切图：剪切指定的背景图的一部分区域在按键区域内显示 (剪切的图像大小及位置同触摸键大小和位置相同)
图标/背景图	可选择的图标和背景图
目标页面	触摸后要跳转的页面
键盘(菜单)	触摸后可弹出键盘或菜单，供录入数据或选择
整数位数	自动：可输入整数位个数不限 1~9：可输入整数位个数不超过设定值
小数位数	0~9：可输入小数位个数不超过设定值
字体	选择字库
字体颜色	设置文本显示的颜色
运算操作	触摸后执行运算操作，如：赋值、加减乘除、bit 位赋值操作
标题/值	设置运算操作或按键时的 Value 值
按键	用来制作自定义键盘(菜单)上的具体按键 如：A ~ Z 键、0 ~ 9 键、回车键、删除键...等
VP 地址	存储键盘(菜单)录入结果或运算操作结果
编译为	把 VP 地址强转为其他 VP 地址类型，触摸键对 VP 地址的读写以强转后的地址类型来处理 如：VP 地址为：VP_N16(16 位数字变量地址)，强转为 STR，键盘录入的数据会以字符串方式存入以 VP_N16 为首地址的内容空间中
数据类型	有符号整型：VP_N16 值范围(-32768 ~ +32767) 无符号整型：VP_N16 值范围(0 ~ 65535) 浮点型：32 位浮点数
最小值	VP 数据最小值
最大值	VP 数据最大值
输入长度	可输入字符的最大个数(选择 VP 地址为 STR 变量类型或选择编译为 STR 可弹出此选项)
光标颜色	设置光标显示的颜色
输入返回	是：输入数据并保存后发送指令到串口(0x77 + 发送 VP 地址 + VP 地址值) 否：输入后不发送指令到串口 输入返回仅部分型号支持
键盘模式	默认：等同于重新输入 修改编辑：弹出键盘后显示相应 VP 地址中的数据 重新输入：弹出键盘后显示数据为空 键盘模式仅部分型号支持
返回值	触摸键按下后发送数据指令到串口 按下时，发送指令到串口(0x79 + 发送页面 ID + 触摸键 ID) 抬起时，发送指令到串口(0x78 + 发送页面 ID + 触摸键 ID) 按下和抬起，分别发送指令到串口(0x79/0x78 + 发送页面 ID + 触摸键 ID) **仅当呼叫-运算操作(VP=Value)时，返回值属性可设置以下：** 按下时，发送指令到串口(0x77 + 发送 VP 地址 + Value 值) 抬起时，发送指令到串口(0x77 + 发送 VP 地址 + Value 值) 按下和抬起时，分别发送指令到串口(0x77 + 发送 VP 地址 + Value 值)
使能 VP	可通过 VP 中的数据控制触摸键是否有效 VP 中数据等于 0 时，触摸键无效(不可触摸) VP 中数据等于 1 时，触摸键有效 (上电后使能 VP 中数据默认会设置为 1)

注：按键按下显示效果

1. 触摸键(按下效果) - 反色显示

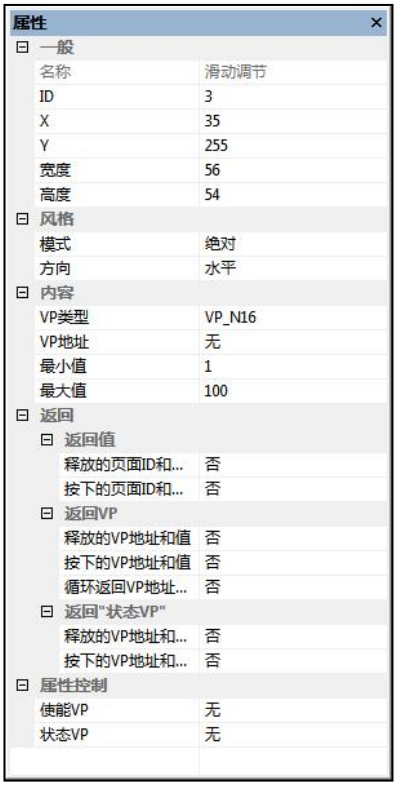


2. 触摸键(按下效果) - 显示图标



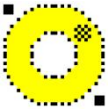
3. 触摸键(按下效果) - 剪切背景图区域显示

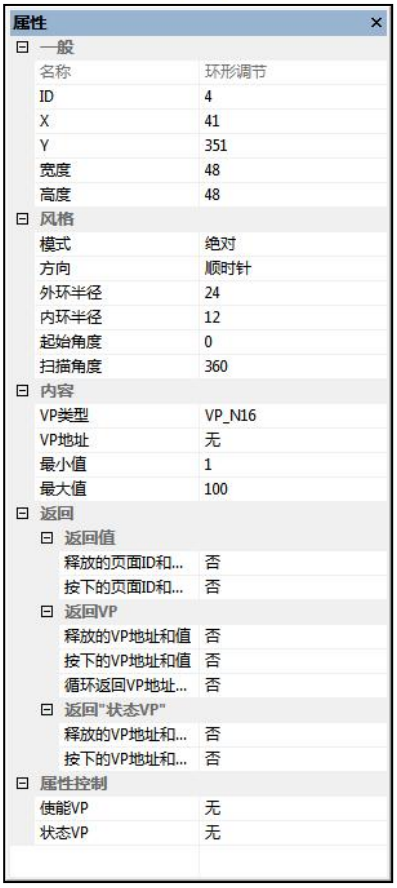
4.3.2 滑动调节

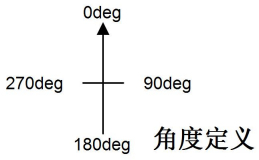


名称	滑动调节 (Slider)
ID	编号
X/Y/宽/高	按键有效区域(页面左上角坐标为 0,0)
模式	绝对：绝对位置触发模式 相对：相对位置触发模式
方向	水平：水平方向滑动 垂直：垂直方向滑动
VP 类型	VP_N16 ： 16 位数字变量 VP_N32 ： 32 位数字变量 VP_REG ： 系统寄存器
VP 地址	数字变量 VP (16/32 位数字变量)和系统寄存器
最小值	VP 数据最小值,值范围: -2147483648~ 2147483647
最大值	VP 数据最大值,值范围: -2147483648~ 2147483647
返回值	抬起时,发送指令到串口(发送页面 ID + 触摸键 ID)，“是”有效
	按下时,发送指令到串口(发送页面 ID + 触摸键 ID)，“是”有效
返回 VP	抬起时,发送指令到串口(发送 VP 地址 + 值)，“是”有效
	按下时,发送指令到串口(发送 VP 地址 + 值)，“是”有效
	每 100ms，发送指令到串口(发送 VP 地址 + 值) ，“是”有效
返回状态 VP	抬起时,发送指令到串口(发送状态 VP 地址 + 触摸键状态)，“是”有效
	按下时,发送指令到串口(发送状态 VP 地址 + 触摸键状态)，“是”有效
使能 VP	通过 VP 中的数据控制触摸键是否有效 VP 中数据等于 0 时，触摸键无效(不可触摸) VP 中数据等于 1 时，触摸键有效 (上电后使能 VP 中数据默认会设置为 1)
状态 VP	通过读取状态 VP 中返回的数据，判断触摸键的状态 VP 中数据等于 0 时，触摸键没有按下 VP 中数据等于 1 时，触摸键按下 VP 中数据等于 2 时，触摸键长按

4.3.3 环形调节

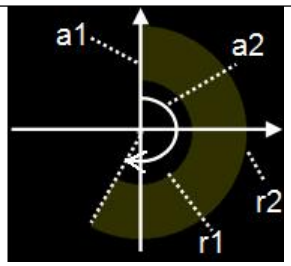




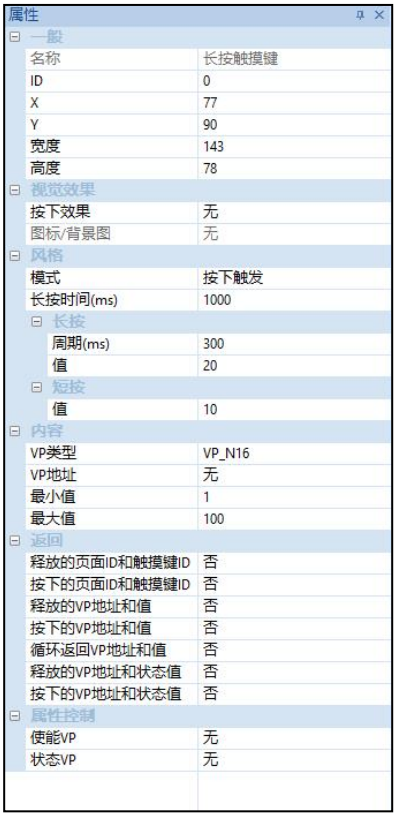


角度定义

名称	环形调节 (Ring)
ID	编号
X/Y/宽/高	按键最大有效区域(页面左上角坐标为 0,0)
模式	绝对：绝对位置触发模式
方向	顺时针：顺时针方向滑动 逆时针：逆时针方向滑动
外环半径	如右图 r2 所示
内环半径	如右图 r1 所示
起始角度	如右图 a1 所示
扫描角度	如右图 a2 所示
VP 类型	VP_N16 ： 16 位数字变量 VP_N32 ： 32 位数字变量 VP_REG ： 系统寄存器
VP 地址	数字变量 VP (16/32 位数字变量和系统寄存器)
最小值	VP 数据最小值,值范围: -2147483648~ 2147483647
最大值	VP 数据最大值,值范围: -2147483648~ 2147483647
返回值	释放时,发送指令到串口(发送页面 ID + 触摸键 ID)，“是”有效
	按下时,发送指令到串口(发送页面 ID + 触摸键 ID)，“是”有效
返回 VP	释放时,发送指令到串口(发送 VP 地址 + 值)，“是”有效
	按下时,发送指令到串口(发送 VP 地址 + 值)，“是”有效
	每 100ms，发送指令到串口(发送 VP 地址 + 值) ，“是”有效
返回状态 VP	释放时,发送指令到串口(发送状态 VP 地址 + 触摸键状态)，“是”有效
	按下时,发送指令到串口(发送状态 VP 地址 + 触摸键状态)，“是”有效
使能 VP	通过 VP 中的数据控制触摸键是否有效 VP 中数据等于 0 时，触摸键无效(不可触摸) VP 中数据等于 1 时，触摸键有效 (上电后使能 VP 中数据默认会设置为 1)
状态 VP	通过读取状态 VP 中返回的数据，判断触摸键的状态 VP 中数据等于 0 时，触摸键没有按下 VP 中数据等于 1 时，触摸键按下 VP 中数据等于 2 时，触摸键长按

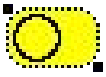


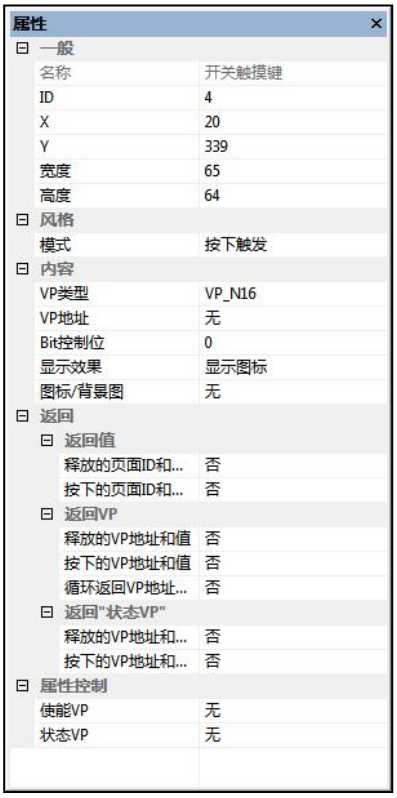
4.3.4 长按触摸键



名称	长按触摸键(TPK_Repeat)
ID	编号
X/Y/宽/高	按键有效区域(页面左上角坐标为 0,0)
按下效果	反色: 反色按键区域 显示图标: 按键区域内显示一个图标 显示剪切图: 剪切图片的一部分区域在按键区域内显示 (剪切的图像大小及位置同触摸键大小和位置相同)
图标/背景图	可选择的图标和背景图
模式	按下触发模式 释放触发模式
长按时间(ms)	判定为长按的时间(ms)
长按周期	长按值变化一次的时间(ms)
长按值	每次长按变化的值
短按值	每次短按变化的值
VP 类型	VP_N16 ： 16 位数字变量 VP_N32 ： 32 位数字变量 VP_REG ： 系统寄存器
VP 地址	数字变量 VP (16/32 位数字变量)和系统寄存器
最小值	VP 数据最小值,值范围: -2147483648~ 2147483647
最大值	VP 数据最大值,值范围: -2147483648~ 2147483647
返回值	抬起时,发送指令到串口(发送页面 ID + 触摸键 ID)，“是”有效
	按下时,发送指令到串口(发送页面 ID + 触摸键 ID)，“是”有效
返回 VP	抬起时,发送指令到串口(发送 VP 地址 + 值)，“是”有效
	按下时,发送指令到串口(发送 VP 地址 + 值)，“是”有效
	每个长按周期(ms)，发送指令到串口(发送 VP 地址 + 值) ，“是”有效
返回状态 VP	抬起时,发送指令到串口(发送状态 VP 地址 + 触摸键状态)，“是”有效
	按下时,发送指令到串口(发送状态 VP 地址 + 触摸键状态)，“是”有效
使能 VP	通过 VP 中的数据控制触摸键是否有效 VP 中数据等于 0 时，触摸键无效(不可触摸) VP 中数据等于 1 时，触摸键有效 (上电后使能 VP 中数据默认会设置为 1)
状态 VP	通过读取状态 VP 中返回的数据，判断触摸键的状态 VP 中数据等于 0 时，触摸键没有按下 VP 中数据等于 1 时，触摸键按下 VP 中数据等于 2 时，触摸键长按

4.3.5 开关触摸键



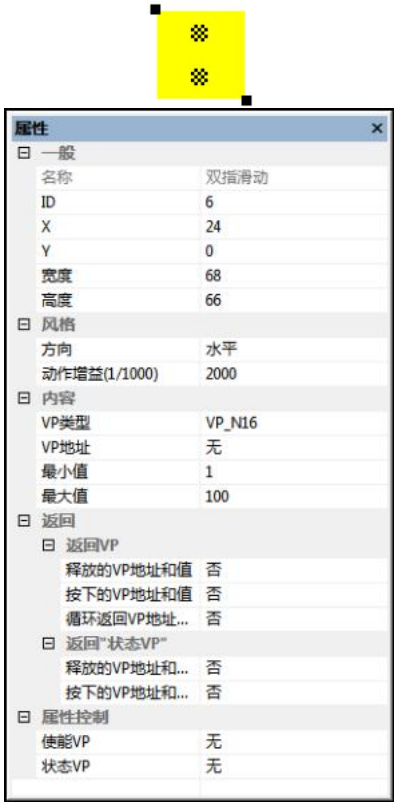


名称	开关触摸键(TPK_Switch)
ID	编号
X/Y/宽/高	按键有效区域(页面左上角坐标为 0,0)
模式	按下后触发模式 释放后触发模式
VP 类型	VP_N16 ： 16 位数字变量 VP_N32 ： 32 位数字变量 VP_REG ： 系统寄存器
VP 地址	数字变量 VP (16/32 位数字变量)和系统寄存器
Bit 控制位	用来控制开关的 bit 位(0~31bit) 按下时，Bit 位=1 再次按下时，Bit 位=0
显示效果	显示图标: 按键区域内显示一个图标 显示剪切图: 剪切图片的一部分区域在按键区域内显示 (剪切的图像大小及位置同触摸键大小和位置相同)
图标/背景图	可选择的图标/背景图
返回值	抬起时,发送指令到串口(发送页面 ID + 触摸键 ID)，“是”有效
	按下时,发送指令到串口(发送页面 ID + 触摸键 ID)，“是”有效
返回 VP	抬起时,发送指令到串口(发送 VP 地址 + 值)，“是”有效
	按下时,发送指令到串口(发送 VP 地址 + 值)，“是”有效
	每 100ms，发送指令到串口(发送 VP 地址 + 值) ，“是”有效
返回状态 VP	抬起时,发送指令到串口(发送状态 VP 地址 + 触摸键状态)，“是”有效
	按下时,发送指令到串口(发送状态 VP 地址 + 触摸键状态)，“是”有效
使能 VP	通过 VP 中的数据控制触摸键是否有效 VP 中数据等于 0 时，触摸键无效(不可触摸) VP 中数据等于 1 时，触摸键有效 (上电后使能 VP 中数据默认会设置为 1)
状态 VP	通过读取状态 VP 中返回的数据，判断触摸键的状态 VP 中数据等于 0 时，触摸键没有按下 VP 中数据等于 1 时，触摸键按下 VP 中数据等于 2 时，触摸键长按

4.3.6 滑动翻页

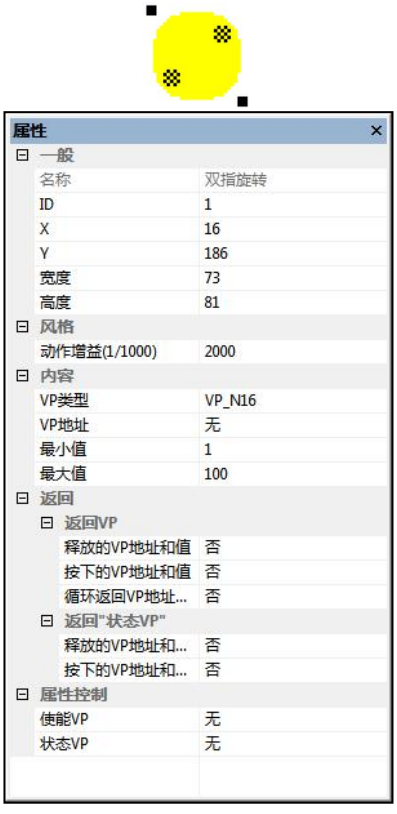
名称	滑动翻页 (Swap_Page)
ID	编号
X/Y/宽/高	控件的位置 (页面左上角坐标为 0,0)
扫描阀值	跳转页面需要拖动的距离(pixel)
左侧页面	向左跳转的页面
右侧页面	向右跳转的页面
返回值	返回跳转的页面 ID
使能 VP	可通过 VP 中的数据控制触摸键是否有效 VP 中数据等于 0 时，触摸键无效(不可触摸) VP 中数据等于 1 时，触摸键有效 (上电后使能 VP 中数据默认会设置为 1)

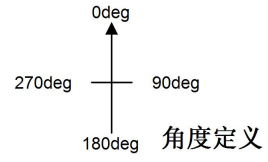
4.3.7 双指滑动



名称	双指滑动 (Slider_2)
ID	编号
X/Y/宽/高	控件的位置 (页面左上角坐标为 0,0)
方向	水平：水平方向滑动 垂直：垂直方向滑动
动作增益(1/1000)	增益值为设定值的 1/1000
VP 类型	VP_N16 ： 16 位数字变量 VP_N32 ： 32 位数字变量 VP_REG ： 系统寄存器
VP 地址	数字变量 VP (16/32 位数字变量)和系统寄存器
最小值	VP 数据最小值,值范围: -2147483648~ 2147483647
最大值	VP 数据最大值,值范围: -2147483648~ 2147483647
返回 VP	抬起时,发送指令到串口(发送 VP 地址 + 值)，“是”有效
	按下时,发送指令到串口(发送 VP 地址 + 值)，“是”有效
	每 100ms，发送指令到串口(发送 VP 地址 + 值) ，“是”有效
返回状态 VP	抬起时,发送指令到串口(发送状态 VP 地址 + 触摸键状态)，“是”有效
	按下时,发送指令到串口(发送状态 VP 地址 + 触摸键状态)，“是”有效
使能 VP	通过 VP 中的数据控制触摸键是否有效 VP 中数据等于 0 时，触摸键无效(不可触摸) VP 中数据等于 1 时，触摸键有效 (上电后使能 VP 中数据默认会设置为 1)
状态 VP	通过读取状态 VP 中返回的数据，判断触摸键的状态 VP 中数据等于 0 时，触摸键没有按下 VP 中数据等于 1 时，触摸键按下 VP 中数据等于 2 时，触摸键长按

4.3.8 双指旋转





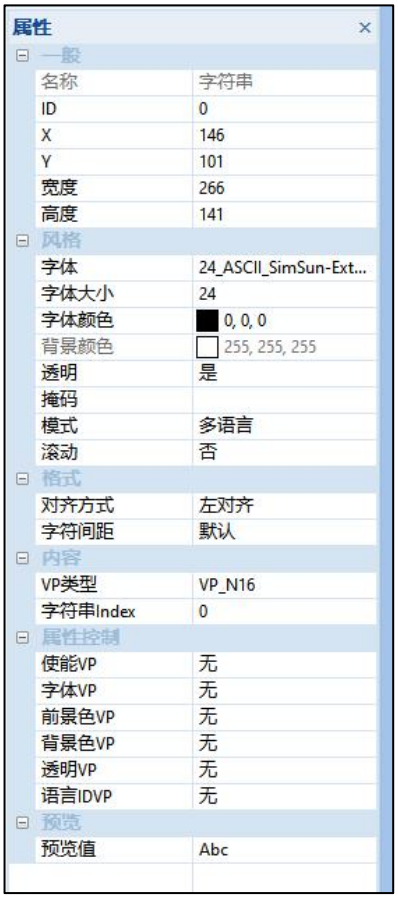
角度定义

名称	双指旋转 (Ring_2)
ID	编号
X/Y/宽/高	控件的位置 (页面左上角坐标为 0,0)
动作增益(1/1000)	增益值为设定值的 1/1000
VP 类型	VP_N16 ： 16 位数字变量 VP_N32 ： 32 位数字变量 VP_REG ： 系统寄存器
VP 地址	数字变量 VP (16/32 位数字变量)和系统寄存器
最小值	VP 数据最小值,值范围: -2147483648~ 2147483647
最大值	VP 数据最大值,值范围: -2147483648~ 2147483647
返回 VP	抬起时,发送指令到串口(发送 VP 地址 + 值)，“是”有效
	按下时,发送指令到串口(发送 VP 地址 + 值)，“是”有效
	每 100ms，发送指令到串口(发送 VP 地址 + 值) ，“是”有效
返回状态 VP	抬起时,发送指令到串口(发送状态 VP 地址 + 触摸键状态)，“是”有效
	按下时,发送指令到串口(发送状态 VP 地址 + 触摸键状态)，“是”有效
使能 VP	通过 VP 中的数据控制触摸键是否有效 VP 中数据等于 0 时，触摸键无效(不可触摸) VP 中数据等于 1 时，触摸键有效 (上电后使能 VP 中数据默认会设置为 1)
状态 VP	通过读取状态 VP 中返回的数据，判断触摸键的状态 VP 中数据等于 0 时，触摸键没有按下 VP 中数据等于 1 时，触摸键按下 VP 中数据等于 2 时，触摸键长按

4.3.9 中英切换

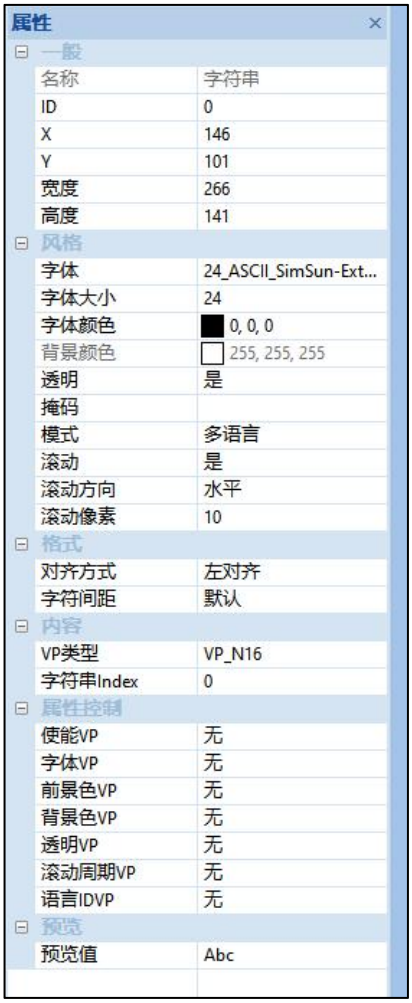
名称	中英切换(TPK-CHEN)
ID	编号
X/Y/宽/高	按键有效区域(页面左上角坐标为 0,0)
英文模式图标	触摸键按下时显示的英文图标

4.3.10 字符串



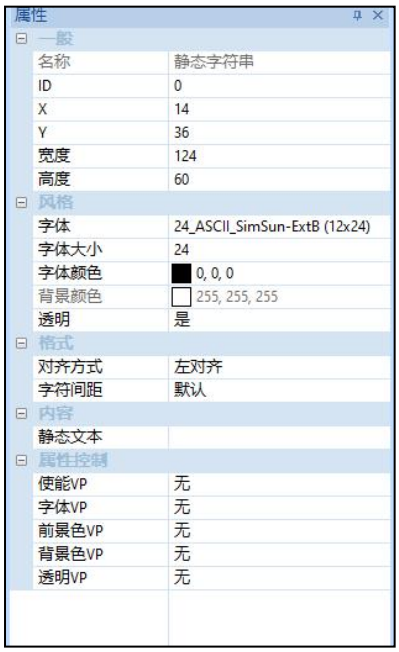
名称	字符串(String)
ID	编号
X/Y/宽/高	文本框大小与位置(页面左上角坐标为 0,0)
字体	选择字库
字体大小	当字体属性选择 TTF 字体，字体大小才能显示相应大小
字体颜色	设置文本显示的颜色
背景颜色	设置文本框背景色
透明	是：不显示文本框背景色 否：显示文本框背景色
掩码	掩码字符，设置后字符串内容全部以掩码字符显示
模式	普通：显示 VP 地址中的数据 多语言：显示 StringTable 表中 Index 值对应的数据。 可变多语言：显示 StringTable 表中 VP 值与 Index 相同的数据。 映射可变多语言：显示映射表中 VP 值与 Index 相同的数据。
滚动	是：字符串滚动显示 否：字符串不滚动显示 滚动属性仅部分型号支持
对齐方式	左对齐、右对齐、居中对齐 多行左对齐、多行右对齐、多行居中对齐
字符间距	默认：依实际字库显示 收缩：自动去除字符与字符间空白间隔(留 1 个像素) 自动：在字体配置<2>创建字体，勾选 DBCS 宽度、字库补充信息选项才能生效，仅部分型号支持
VP 类型	VP_N16(16 位数字变量) VP_STR(字符串变量)
VP 地址	变量地址(16 位数字变量地址/字符串变量地址)
长度	实际显示字符串长度(范围: 0 ~ 127) 最大可显示 127 个 ASCII 字符或 63 个中文字符
使能 VP	可通过 VP 中的数据控制字符串控件是隐藏/显示 显示：0x0001，隐藏：0x0000
字体 VP	往 VP 变量中写入数据可改变字体 (数据应为字库 ID，详见字库设置)
前景色 VP	往 VP 变量中写入数据可改变字体颜色 数据格式：RGB565
背景色 VP	往 VP 变量中写入数据可改变背景颜色 数据格式：RGB565
透明 VP	往 VP 变量中写入数据可改变透明属性 透明：0x0001 不透明：0x0000
语言 ID VP	往 VP 变量中写入数据可改变语言 (未设置该 VP 时，默认寄存器 0xFFFF2D 的值为当前语言)
预览值	预览字符(仅提供预览显示，不是赋初值)

4.3.11 滚动字符串



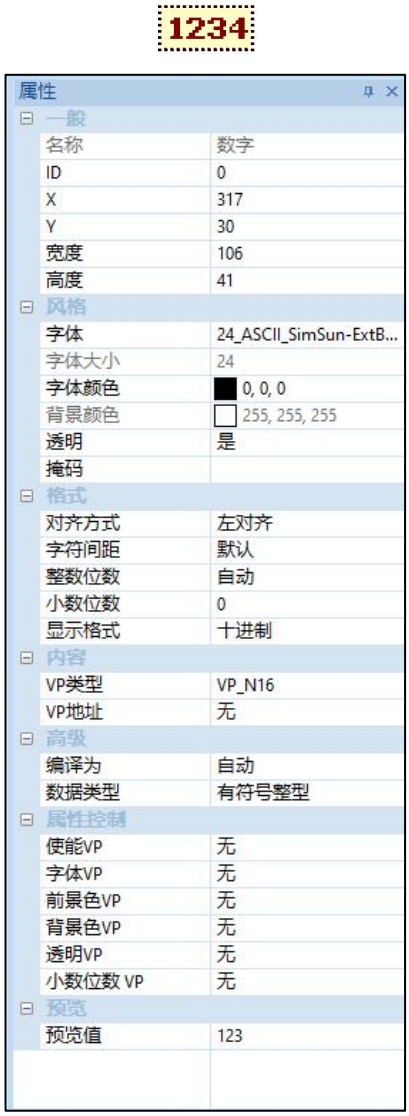
名称	字符串(String)
ID	编号
X/Y/宽/高	文本框大小与位置(页面左上角坐标为 0,0)
字体	选择字库
字体大小	当字体属性选择 TTF 字体，字体大小才能显示相应大小
字体颜色	设置文本显示的颜色
背景颜色	设置文本框背景色
透明	是：不显示文本框背景色 否：显示文本框背景色
掩码	掩码字符，设置后字符串内容全部以掩码字符显示
模式	普通：显示 VP 地址中的数据 多语言：显示 StringTable 表中 Index 值对应的数据，显示语言可切换 可变多语言：显示 StringTable 表中 VP 值对应的数据，显示语言可切换 映射可变多语言：创建映射表，显示映射表中 VP 值对应的数据，显示语言可切换 多语言功能仅部分型号支持
滚动	是：字符串滚动显示 否：字符串不滚动显示 滚动属性仅部分型号支持
滚动方向	水平、垂直
滚动像素	可设置字符滚动的速度，设置的值越大，滚动越快
对齐方式	左对齐、右对齐、居中对齐 多行左对齐、多行右对齐、多行居中对齐
字符间距	默认：依实际字库显示 收缩：自动去除字符与字符间空白间隔(留 1 个像素) 自动：在字体配置<2>创建字体，勾选 DBCS 宽度、字库补充信息选项才能生效，仅部分型号支持
VP 类型	VP_N16(16 位数字变量) VP_STR(字符串变量)
VP 地址	变量地址(16 位数字变量地址/字符串变量地址)
长度	实际显示字符串长度(范围: 0 ~ 127) 最大可显示 127 个 ASCII 字符或 63 个中文字符
使能 VP	可通过 VP 中的数据控制字符串控件是隐藏/显示 显示：0x0001，隐藏：0x0000
字体 VP	往 VP 变量中写入数据可改变字体 (数据应为字库 ID，详见字库设置)
前景色 VP	往 VP 变量中写入数据可改变字体颜色 数据格式：RGB565
背景色 VP	往 VP 变量中写入数据可改变背景颜色 数据格式：RGB565
透明 VP	往 VP 变量中写入数据可改变透明属性 透明：0x0001 不透明：0x0000
滚动周期 VP	往 VP 变量中写入数据可改变字符滚动速度
语言 ID VP	往 VP 变量中写入数据可改变语言 (未设置该 VP 时，默认寄存器 0xFFFF2D 的值为当前语言)
预览值	预览字符(仅提供预览显示，不是赋初值)

4.3.12 静态字符串



名称	静态字符串(Static String)
ID	编号
X/Y/宽/高	文本框大小与位置(页面左上角坐标为 0,0)
字体	选择字库
字体大小	当字体属性选择 TTF 字体，字体大小才能显示相应大小
字体颜色	设置文本显示的颜色
背景颜色	文本框背景色
透明	是：不显示文本框背景色 否：显示文本框背景色
对齐方式	左对齐、右对齐、居中对齐
字符间距	默认:依实际字库显示 收缩:自动去除字符与字符间空白间隔(留 1 个像素) 自动：在字体配置<2>创建字体，勾选 DBCS 宽度、字库补充信息选项才能生效，仅部分型号支持
静态文本	显示的内容 最大可显示 127 个 ASCII 字符或 63 个中文字符
使能 VP	可通过 VP 中的数据控制显示内容是否隐藏/显示 显示：0x0001， 隐藏：0x0000
字体 VP	往 VP 变量中写入数据可改变字体 (数据应为字库 ID,详见字库说明)
前景色 VP	往 VP 变量中写入数据可改变字体颜色 数据格式: RGB565
背景色 VP	往 VP 变量中写入数据可改变背景颜色 数据格式: RGB565
透明 VP	往 VP 变量中写入数据可改变透明属性 透明：0x0001，不透明：0x0000

4.3.13 数字



名称	数字(Number)
ID	编号
X/Y/宽/高	文本框大小与位置(页面左上角坐标为 0,0)
字体	选择字库
字体大小	当字体属性选择 TTF 字体，字体大小才能显示相应大小
字体颜色	设置文本显示的颜色
背景颜色	文本框背景色
透明	是：不显示文本框背景色 否：显示文本框背景色
掩码	掩码字符，设置后字符串内容全部以掩码字符显示
对齐方式	左对齐、右对齐、居中对齐
字符间距	默认:依实际字库显示 收缩:自动去除字符与字符间空白间隔(留 1 个像素) 自动：在字体配置<2>创建字体，勾选 DBCS 宽度、字库补充信息选项才能生效，仅部分型号支持
整数位数	整数位个数(自动:依实际数字位数显示) 不够位数时自动补零
小数位数	小数位个数（不够位数时自动补零）
显示格式	十进制、十六进制
VP 类型	VP_N16、VP_N32、VP_N64、VP_REG
VP 地址	数字变量 VP (16/32/64 位数字变量和系统寄存器变量)
编译为	自动(依实际 VP 地址类型) N16(强制转为 16 位数字变量类型) N32(强制转为 32 位数字变量类型)
数据类型	有符号整型 无符号整型 浮点型(float), VP 地址中写入浮点数据时,必须设置为此类型
使能 VP	可通过 VP 中的数据控制数字是否隐藏/显示 显示：0x0001 隐藏：0x0000
字体 VP	往 VP 变量中写入数据可改变字体 (数据应为字库 ID,详见字库说明)
前景色 VP	往 VP 变量中写入数据可改变字体颜色 数据格式: RGB565
背景色 VP	往 VP 变量中写入数据可改变背景颜色 数据格式: RGB565
透明 VP	往 VP 变量中写入数据可改变透明属性 透明=0x0001 不透明=0x0000
小数位数 VP	往 VP 变量中写入数据可改变小数位数 值范围：0~9
预览值	预览字符 (仅提供预览显示,非赋初值)

注:

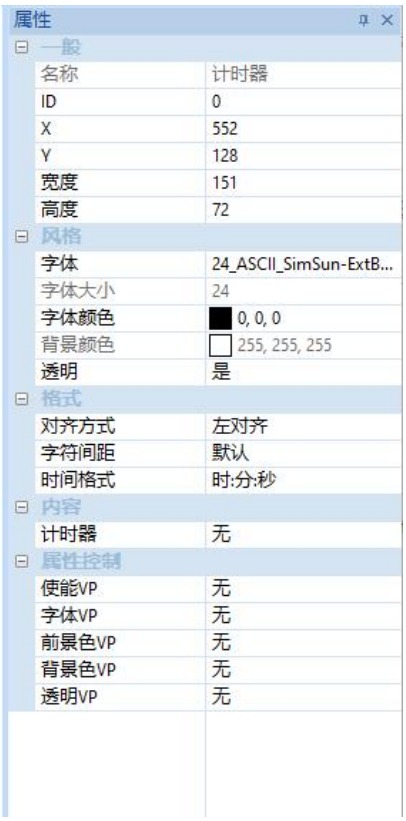
*1. 整数位、小数位设定显示示例

数据 (十进制)	整数位	小数位	显示结果	描述
15	自动	3	0.015	小数位= 3，整数位=自动
23762	自动	3	23.762	小数位= 3，整数位=自动
5629	5	1	00562.9	小数位= 1，整数位= 5
-87913	3	2	-879.13	小数位= 2，整数位= 3
-13277	2	2	-99.99	VP 地址中数据不在有效值范围内(-99.99 ~ +99.99)
1758	3	0	999	VP 地址中数据不在有效值范围内(-999 ~ +999)
24.81	自动	3	24.810	编译数据类型为“浮点型”时，输入小数位不够，末尾自动补 0
125.568	自动	2	125.57	编译数据类型为“浮点型”时，根据小数位设置，四舍五入显示正确结果

*2. 由于显示设定，浮点类型值可能会舍入或截断.

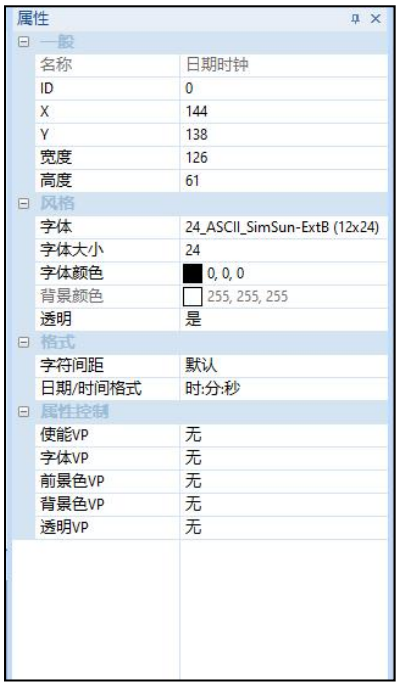
*3. 显示浮点数据时，建议设置"整数位=自动，小数位=0".

4.3.14 计时器



名称	计时器(Timer)
ID	编号
X/Y/宽/高	文本框大小与位置(页面左上角坐标为 0,0)
字体	选择字库
字体大小	设置字体大小，仅选择 TTF 字体(字库配置 3)时有效
字体颜色	设置文本显示颜色
背景颜色	文本框背景色
透明	是：不显示文本框背景色 否：显示文本框背景色
对齐方式	左对齐、右对齐、居中对齐
字符间距	默认:依实际字库显示 收缩:自动去除字符与字符间空白间隔(留 1 个像素) 自动：在字体配置<2>创建字体，勾选 DBCS 宽度、字库补充信息选项才能生效，仅部分型号支持
时间格式	时:分:秒 分:秒 秒
计时器	32 位数字 VP 变量 计时器显示的内容来自此数字变量，可通过指令写入初值
使能 VP	可通过 VP 中的数据控制计时器是否隐藏/显示 显示：0x0001 隐藏：0x0000
字体 VP	往 VP 变量中写入数据可改变字体 (数据应为字库 ID,详见字库说明)
前景色 VP	往 VP 变量中写入数据可改变字体颜色 数据格式: RGB565
背景色 VP	往 VP 变量中写入数据可改变背景颜色 数据格式: RGB565
透明 VP	往 VP 变量中写入数据可改变透明属性 透明=0x0001 不透明=0x0000

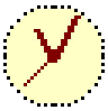
4.3.15 日期时钟



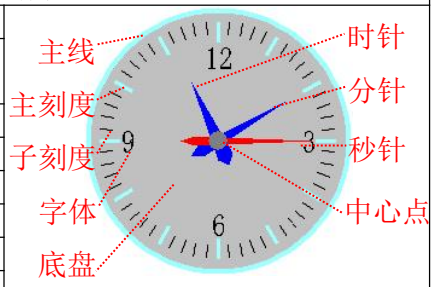
名称	日期时钟(Real Time Clock)
ID	编号
X/Y/宽/高	文本框大小与位置(页面左上角坐标为 0, 0)
字体	选择字库
字体大小	当字体属性选择 TTF 字体，字体大小才能显示相应大小
字体颜色	设置文本显示颜色
背景颜色	文本框背景色
透明	是：不显示文本框背景色 否：显示文本框背景色
字符间距	默认:依实际字库显示 收缩:自动去除字符与字符间空白间隔(留 1 个像素) 自动：在字体配置<2>创建字体，勾选 DBCS 宽度、字库补充信息选项才能生效，仅部分型号支持
日期/时间格式	序 格式 序 格式 1 星期 年-月-日 时:分:秒 7 月 2 星期 年-月-日 8 日 3 年-月-日 时:分:秒 9 时 4 年-月-日 10 分 5 时:分:秒 11 秒 6 年
使能 VP	往 VP 变量中写入数据可控制显示内容隐藏/显示 显示：0x0001，隐藏：0x0000
字体 VP	往 VP 变量中写入数据可改变字体 (数据应为字库 ID,详见字库说明)
前景色 VP	往 VP 变量中写入数据可改变字体颜色 数据格式: RGB565
背景色 VP	往 VP 变量中写入数据可改变背景颜色 数据格式: RGB565
透明 VP	往 VP 变量中写入数据可改变透明属性 透明：0x0001，不透明：0x0000

4.3.16 模拟时钟

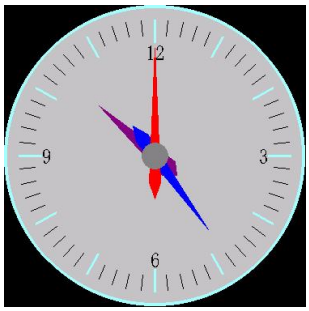
4.3.16.1 指针表盘模式



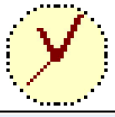
名称	模拟时钟(Round_Clock)
ID	编号
X/Y/宽/高	文本框大小与位置(页面左上角坐标为 0, 0)
模式	指针表盘：固件自绘指针和表盘 图标旋转：指针使用图标来显示，无表盘
透明	是：表盘无背景色填充 否：表盘有背景色填充，底盘使能有效
背景颜色	如右图的底盘颜色所示
盘面	是：显示表盘盘面 否：不显示表盘盘面
主线颜色	如右图的主线颜色所示
主线宽度	如右图的主线宽度所示(pixel)
主刻度颜色	如右图的主刻度颜色所示
主刻度宽度	如右图的主刻度宽度所示(pixel)
子刻度颜色	如右图的子刻度颜色所示
子刻度宽度	如右图的子刻度宽度所示(pixel)
字体	如右图的字体所示
字体颜色	如右图的字体颜色所示
指针类型	时分秒三个指针的几何形状(直线/四边形/三角形)
时针颜色	如右图的时针颜色所示
分针颜色	如右图的分针颜色所示
秒针颜色	如右图的秒针颜色所示
中心点颜色	如右图的中心点颜色所示
使能 VP	往 VP 变量中写入数据可控制显示内容隐藏/显示 显示：0x0001，隐藏：0x0000
透明 VP	往 VP 变量中写入数据可改变透明属性 透明=0x0001，不透明=0x0000
背景色 VP	往 VP 变量中写入数据可改变背景颜色 数据格式: RGB565
指针类型 VP	往 VP 变量中写入数据可改变指针类型 直线 = 0 ;四边形 = 1;三角形 = 2;
时针颜色 VP	往 VP 变量中写入数据可改变时针颜色 数据格式: RGB565
分针颜色 VP	往 VP 变量中写入数据可改变分针颜色 数据格式: RGB565
秒针颜色 VP	往 VP 变量中写入数据可改变秒针颜色 数据格式: RGB565
中心点颜色 VP	往 VP 变量中写入数据可改变中心点颜色 数据格式: RGB565
主线颜色 VP	往 VP 变量中写入数据可改变主线颜色 数据格式: RGB565
主刻度颜色 VP	往 VP 变量中写入数据可改变刻度颜色 数据格式: RGB565
子刻度颜色 VP	往 VP 变量中写入数据可改变子刻度颜色 数据格式: RGB565
字体颜色 VP	往 VP 变量中写入数据可改变字体颜色 数据格式: RGB565



注：示例效果 10:24:00

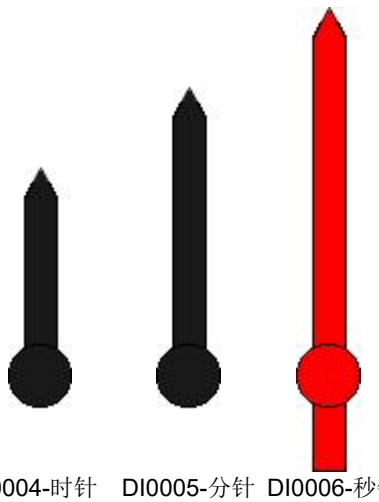


4.3.16.2 图标旋转模式



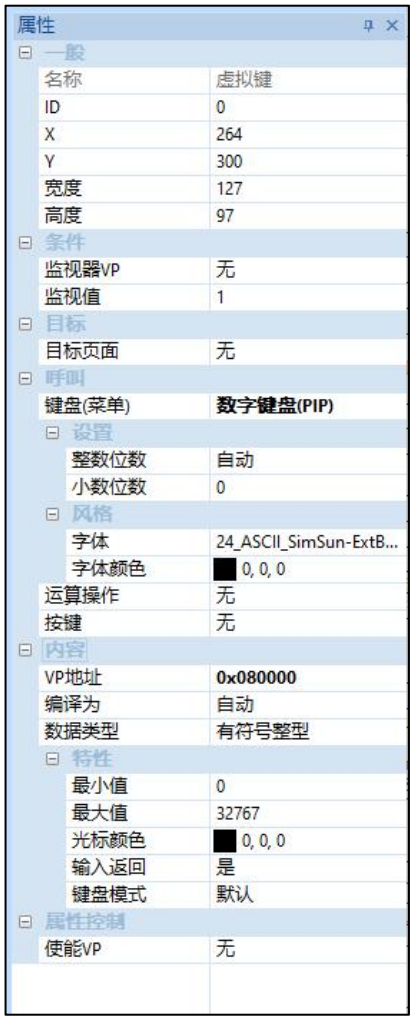
名称	模拟时钟(Round_Clock)
ID	编号
X/Y/宽/高	文本框大小与位置(页面左上角坐标为 0, 0)
模式	指针表盘：固件自绘指针和表盘 图标旋转：指针使用图标来显示，无表盘
透明	是：指针图片中与透明色相同的颜色不显示 否：指针图片全部显示
透明色	指针图片中透明掉的颜色 数据格式：RGB565
时针图标	选择时针图标
时针图标中心点 X	设置时针图标中心点 X 坐标
时针图标中心点 Y	设置时针图标中心点 Y 坐标
分针图标	选择分针图标
分针图标中心点 X	设置分针图标中心点 X 坐标
分针图标中心点 Y	设置分针图标中心点 Y 坐标
秒针图标	选择秒针图标
秒针图标中心点 X	设置秒针图标中心点 X 坐标
秒针图标中心点 Y	设置秒针图标中心点 Y 坐标
使能 VP	往 VP 变量中写入数据可控制显示内容隐藏/显示 显示= 0x0001，隐藏= 0x0000
透明 VP	往 VP 变量中写入数据可改变透明属性 透明=0x0001，不透明=0x0000
透明色 VP	往 VP 变量中写入数据可改变指针图标透明颜色 数据格式: RGB565
时针图标 VP	往 VP 变量中写入数据可改变时针图标
分针图标 VP	往 VP 变量中写入数据可改变分针图标
秒针图标 VP	往 VP 变量中写入数据可改变没针图标

注：示例效果



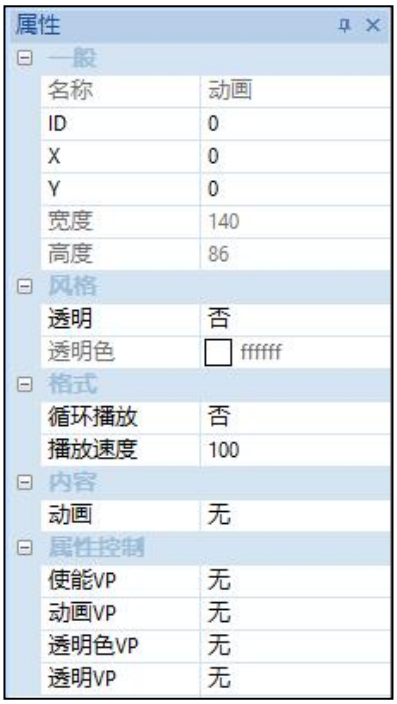
DI0004-时针 DI0005-分针 DI0006-秒针

4.3.17 虚拟键



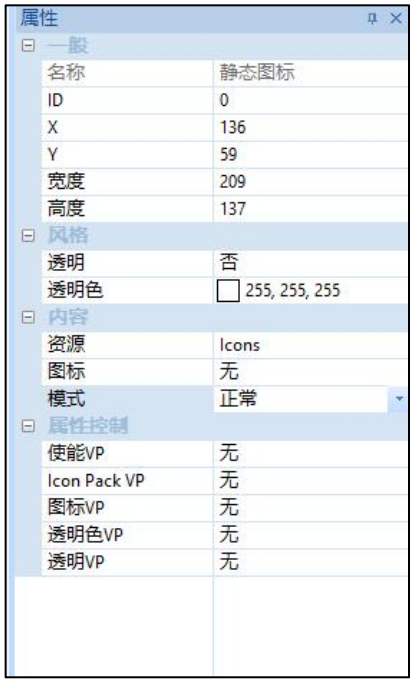
名称	虚拟键(Virtual Key)
ID	编号
X/Y/宽/高	--
监视器 VP	运行中监控的 VP 地址
监视值	运行中监控的数值(范围 1~32767) (当监控的 VP 地址中数据等于此值时，虚拟键被触发。同时设置监控 VP 地址中的数据为 0x00)
目标页面	跳转的页面
键盘(菜单)	可弹出键盘或菜单，供录入数据或选择
整数位数	自动：可输入整数位个数不限 1~9：可输入整数位个数不超过设定值
小数位数	0~9：可输入小数位个数不超过设定值
字体	选择字库
字体颜色	设置文本显示的颜色
运算操作	触摸后执行运算操作，如：赋值、加减乘除、bit 位赋值操作
标题/值	设置运算操作或按键时的 Value 值
按键	用来制作自定义键盘(菜单)上的具体按键 如：A ~ Z 键、0 ~ 9 键、回车键、删除键...等 (呼叫功能详细说明，请参考"触摸键/虚拟键呼叫功能")
VP 地址	存储键盘(菜单)录入结果或运算操作结果
编译为	把 VP 地址强转为其他 VP 地址类型，触摸键对 VP 地址的读写以强转后的地址类型来处理 如：VP 地址为：VP_N16(16 位数字变量地址)，强转为 STR，键盘录入的数据会以字符串方式存入以 VP_N16 为首地址的内容空间中
数据类型	有符号整型：值范围(-32768 ~ +32767) 无符号整型：值范围(0 ~ 65535) 浮点型：32 位浮点数
最小值	VP 数据最小值
最大值	VP 数据最大值
输入长度	可输入字符的最大个数(选择 VP 地址为 STR 变量类型或选择编译为 STR 可弹出此选项)
光标颜色	设置光标显示的颜色
输入返回	是：输入数据并保存后发送指令到串口(0x77 + 发送 VP 地址 + VP 地址值) 否：输入后不发送指令到串口 输入返回仅部分型号支持
键盘模式	默认：等同于重新输入 修改编辑：弹出键盘后显示相应 VP 地址中的数据 重新输入：弹出键盘后显示数据为空 键盘模式仅部分型号支持
使能 VP	可通过 VP 中的数据控制虚拟键是否有效 VP 中数据等于 0 时，虚拟键无效(不可触发) VP 中数据等于 1 时，虚拟键有效 (上电后使能 VP 中数据默认会设置为 1)

4.3.18 动画



名称	动画(Animation)
ID	编号
X/Y/宽/高	显示位置和大小(页面左上角坐标为 0,0)
透明	是：图片中与透明色相同的颜色不显示 否：图片全部显示
透明色	图片中透明掉的颜色
循环播放	是：在当前画面循环播放 否：在当前画面播放一次
播放速度	每帧间隔时间(100ms~2000ms)
动画	选中一个资源栏中创建的动画文件
使能 VP	往 VP 变量中写入数据可控制动画隐藏/显示 显示：0x0001 隐藏：0x0000
动画 VP	往 VP 变量中写入数据更改动画 值范围：0 ~ 999
透明色 VP	往 VP 变量中写入数据可改变透明颜色 数据格式：RGB565
透明 VP	往 VP 变量中写入数据可改变透明属性 透明：0x0001 不透明：0x0000

4.3.19 静态图标

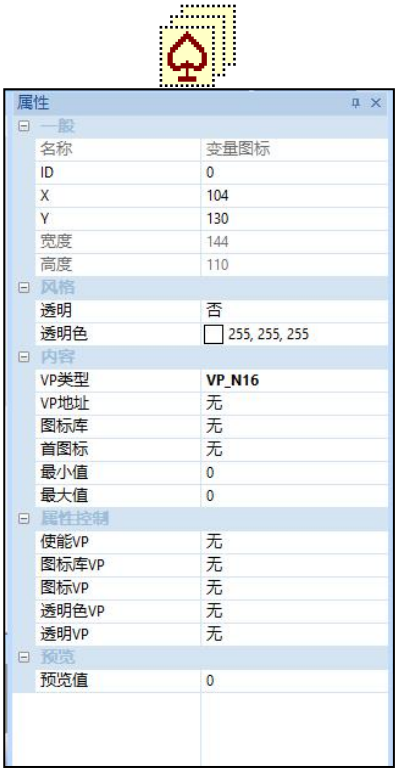


名称	静态图标(Static Icon)
ID	编号
X/Y/宽/高	图标显示位置和大小(页面左上角坐标为 0,0)
透明	是：图片中与透明色相同的颜色不显示 否：图片全部显示
透明色	图片中透明掉的颜色
图标库	工程资源栏创建图标库，选择 ICO001，仅限部分型号支持，比如新建工程的分辨率是 800x480，在工程设置里，选择设备型号为：HMT070ETD-C
图标	图标 ID
模式	正常：显示图标大小固定 缩放：可更改显示图标大小
使能 VP	往 VP 变量中写入数据可控制静态图标隐藏/显示 显示：0x0001 隐藏：0x0000
图标库 VP	往 VP 变量中写入数据更改图标库 ID 值范围：0 ~ 9999
图标 VP	往 VP 变量中写入数据更改图标 ID 值范围：0 ~ 9999
透明色 VP	往 VP 变量中写入数据可改变透明颜色 数据格式：RGB565
透明 VP	往 VP 变量中写入数据可改变透明属性 透明：0x0001 不透明：0x0000

4.3.20 位变量图标

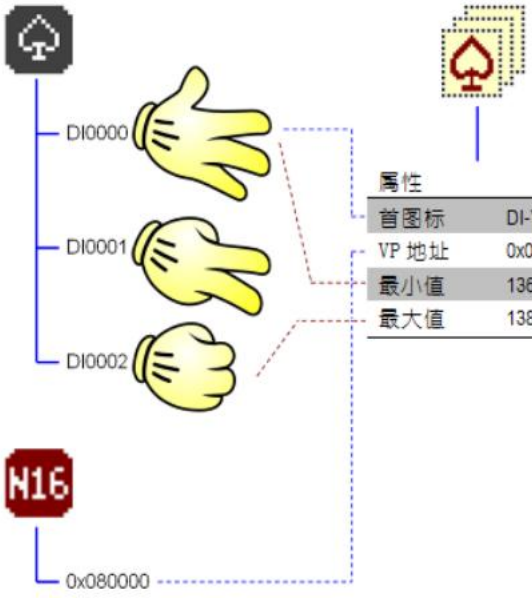
名称	位变量图标(BitIcon)
ID	编号
X/Y/宽/高	显示位置和大小(页面左上角坐标为 0,0)
透明	是：图片中与透明色相同的颜色不显示 否：图片全部显示
透明色	图片中透明掉的颜色
图标库	工程资源栏创建图标库，选择 ICO001，仅限部分型号支持，比如新建工程的分辨率是 800x480，在工程设置里，选择设备型号为：HMT070ETD-C
Bit=1,显示	显示图标 显示背景图剪切区域
Bit=1,图标/背景图	DI0000/DP0000
Bit=0,显示	显示图标 显示背景图剪切区域
Bit=0,图标/背景图	DI0000/DP0000
VP 类型	VP_N16(16 位数字变量) VP_N32(32 位数字变量) VP_REG(系统寄存器变量)
VP 地址	变量地址(16/32 位数字变量和系统寄存器变量)
Bit 控制位	控制位图标的 bit 位(0~31bit) Bit 位=1，显示图标_1 Bit 位=0，显示图标_0
使能 VP	往 VP 变量中写入数据可控制位变量图标隐藏/显示 显示：0x0001 隐藏：0x0000
图标库 VP	往 VP 变量中写入数据更改图标库 ID 值范围：0 ~ 9999
图标_1_VP	往 VP 变量中写入数据可更改属性"图标_1"的图标 ID 值范围：0~9999
图标_0_VP	往 VP 变量中写入数据可更改属性"图标_0"的图标 ID 值范围：0~9999
透明色 VP	往 VP 变量中写入数据可改变透明颜色 数据格式：RGB565
透明 VP	往 VP 变量中写入数据可改变透明属性 透明：0x0001 不透明：0x0000
预览值	预览(仅提供预览显示，非赋初值)

4.3.21 变量图标



名称	变量图标(Index Icon)
ID	编号
X/Y/宽/高	显示位置和大小(页面左上角坐标为 0,0)
透明	是：图片中与透明色相同的颜色不显示 否：图片全部显示
透明色	图片中透明掉的颜色
VP 类型	VP_N16(16 位数字变量) VP_N32(32 位数字变量) VP_REG(系统寄存器变量)
VP 地址	变量地址(16/32 位数字变量和系统寄存器变量)
图标库	工程资源栏创建图标库，选择 ICO001，仅限部分型号支持，比如新建工程的分辨率是 800x480，在工程设置里，选择设备型号为：HMT070ETD-C
首图标	选择首图标 ID
最小值	数据最小值，与首图标 ID 对应
最大值	数据最大值，与首图标 ID +(max-min)图标 ID 对应
使能 VP	往 VP 变量中写入数据可控制变量图标隐藏/显示 显示：0x0001 隐藏：0x0000
图标库 VP	往 VP 变量中写入数据更改图标库 ID 值范围：0 ~ 9999
图标 VP	往 VP 变量中写入数据可更改属性"首图标"的图标 ID 值范围：0~9999
透明色 VP	往 VP 变量中写入数据可改变透明颜色 数据格式：RGB565
透明 VP	往 VP 变量中写入数据可改变透明属性 透明：0x0001 不透明：0x0000
预览值	预览(仅提供预览显示，非赋初值)

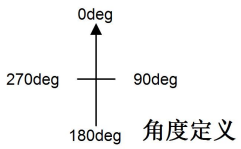
注：数值与图表对应关系



VP value	IDX display result
0x080000=0 (outside range, less than MIN)	Blank
0x080000=136 (within range, same as MIN)	(show 1st ICON)
0x080000=137 (within range, MIN+1)	(show 2nd ICON)
0x080000=138 (within range, MIN+2)	(show 3rd)
0x080000=9997 (outside range, more than max)	blank

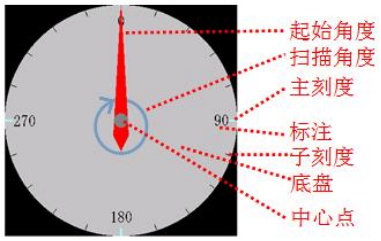
4.3.22 表盘

4.3.22.1 指针表盘模式



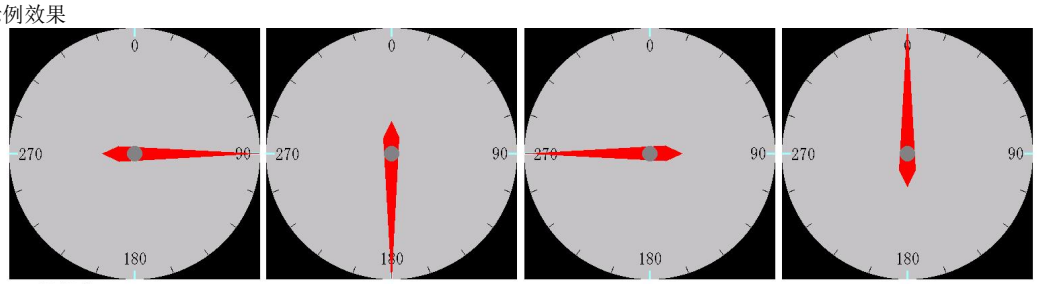
角度定义

名称	表盘(Tachometer)
ID	编号
X/Y/宽/高	显示位置和大小(页面左上角坐标为 0,0)
模式	指针表盘：固件根据设定值自绘指针和表盘 环形条混色：固件根据设定值自绘渐变环形条 环形条变色：固件根据设定值自绘相应色阶的环形条 图标旋转：根据设定角度旋转图片 图标开口：根据角度百分比显示相应的图片内容 预合图标旋转：软件生成图片群，根据设定显示相应的图片 预合图标开口：软件生成图片群，根据设定显示相应的图片
方向	顺时针：顺时针旋转 逆时针：逆时针旋转
起始角度	如右图起始角度所示(单位 1 度)
扫描角度	如右图扫描角度所示(单位 1 度)
刻度&标注使能	是：显示刻度和标注 否：不显示刻度和标注
底色显示	是：显示底盘 否：不显示底盘
底色	底盘（如右图所示）的背景颜色
主刻度半径	设置主刻度（如右图所示）的半径
主刻度数量	设置主刻度（如右图所示）的个数
主刻度宽度	设置主刻度（如右图所示）的宽度
主刻度颜色	设置主刻度（如右图所示）的颜色
标注显示	是：显示标注 否：不显示标注
标注字体	设置标注（如右图所示）显示的字体
标注颜色	设置标注（如右图所示）的颜色
标注最小值	刻度上标记的最小数值
标注最大值	刻度上标记的最大数值
子刻度显示	是：显示子刻度 否：不显示子刻度
子刻度数量	设置子刻度（如右图所示）的个数
子刻度宽度	设置每个子刻度（如右图所示）的宽度
子刻度颜色	设置子刻度（如右图所示）的颜色
指针类型	指针的几何形状(直线/四边形/三角形)
指针颜色	设置指针的颜色
中心点颜色	设置中心点（如下图所示）的颜色
旋转中心	自动：自动设置旋转中心，默认为圆心，不可修改 高级：自定义旋转中心，可手动修改 如上图中心点所示
旋转中心点 X	设置旋转中心点的位置坐标，控件左上角坐标为 0,0
旋转中心点 Y	设置旋转中心点的位置坐标，控件左上角坐标为 0,0
VP 类型	VP_N16: 16 位数字变量 VP_N32: 32 位数字变量 VP_REG:系统寄存器变量
VP 地址	变量 VP (16/32 位数字变量和系统寄存器变量)
最小值	旋转范围最小值,值范围: -2147483648~ 2147483647
最大值	旋转范围最大值,值范围: -2147483648~ 2147483647
使能 VP	往 VP 变量中写入数据可控制隐藏/显示 显示= 0x0001 ;隐藏= 0x0000
透明 VP	往 VP 变量中写入数据可控制是否透明 透明= 0x0001 ;不透明= 0x0000
底色 VP	往 VP 变量中写入数据可控制背景颜色 数据格式: RGB565
指针类型 VP	往 VP 变量中写入数据可控制指针类型 0 = 直线 ;1 = 四边形 ;2 = 三角形
指针颜色 VP	往 VP 变量中写入数据可控制指针颜色



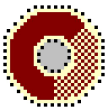
	数据格式: RGB565
中心点颜色 VP	往 VP 变量中写入数据可控制中心点颜色 数据格式: RGB565
主刻度颜色 VP	往 VP 变量中写入数据可控制刻度颜色 数据格式: RGB565
子刻度颜色 VP	往 VP 变量中写入数据可控制子刻度颜色 数据格式: RGB565
标注颜色 VP	往 VP 变量中写入数据可控制字体颜色数 据格式: RGB565
预览值	预览(仅提供预览显示,非赋初值)

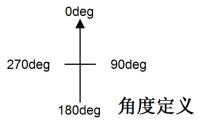
注：示例效果



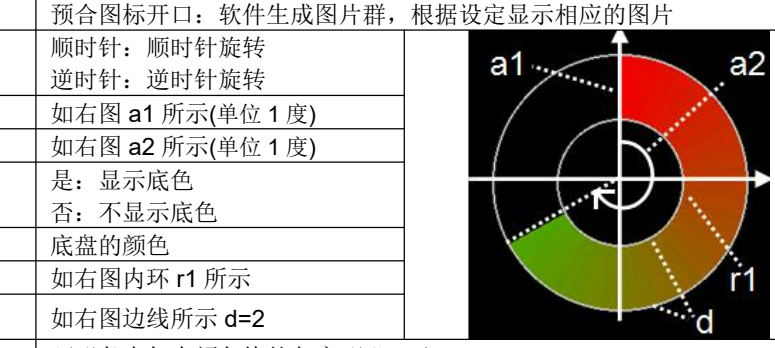
VP数据值 899 1799 2699 3599

4.3.22.2 环形条混色模式

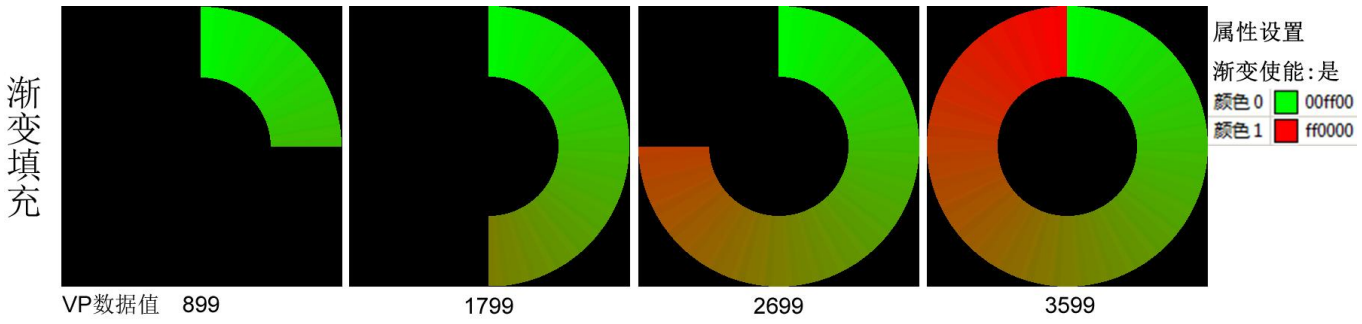




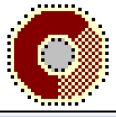
名称	表盘(Tachometer)
ID	编号
X/Y/宽/高	显示位置和大小(页面左上角坐标为 0,0) ，外环直径与宽高相等且联动
模式	指针表盘：固件根据设定值自绘指针和表盘 环形条混色：固件根据设定值自绘渐变环形条 环形条变色：固件根据设定值自绘相应色阶的环形条 图标旋转：根据设定角度旋转图片 图标开口：根据角度百分比显示相应的图片内容 预合图标旋转：软件生成图片群，根据设定显示相应的图片 预合图标开口：软件生成图片群，根据设定显示相应的图片
方向	顺时针：顺时针旋转 逆时针：逆时针旋转
起始角度	如右图 a1 所示(单位 1 度)
扫描角度	如右图 a2 所示(单位 1 度)
底色显示	是：显示底色 否：不显示底色
底色	底盘的颜色
内环半径	如右图内环 r1 所示
边线宽度	如右图边线所示 d=2
步进值	环形条中每个颜色块的角度(单位 1 度)
间隔值	每两个颜色块之间的间隔角度 (单位 1 度)
色阶数量	设置颜色的总数
色阶	设置每种颜色的颜色值
旋转中心	自动：自动设置旋转中心，不能修改，默认为圆心 高级：自定义旋转中心，可手动修改
旋转中心点 X	设置旋转中心点的位置坐标，控件左上角坐标为 0,0
旋转中心点 Y	设置旋转中心点的位置坐标，控件左上角坐标为 0,0
VP 类型	VP_N16: 16 位数字变量 VP_N32: 32 位数字变量 VP_REG: 系统寄存器变量
VP 地址	变量 VP (16/32 位数字变量和系统寄存器变量)
最小值	VP 变量数据最小值,值范围: -2147483648~ 2147483647
最大值	VP 变量数据最大值,值范围: -2147483648~ 2147483647
使能 VP	往 VP 变量中写入数据可控制隐藏/显示 显示：0x0001 ； 隐藏：0x0000
颜色数组 VP	往 VP 变量中写入数据可改变颜色数组里的值 数据格式：RGB565
预览值	预览(仅提供预览显示,非赋初值)

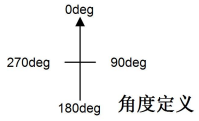


注：示例效果

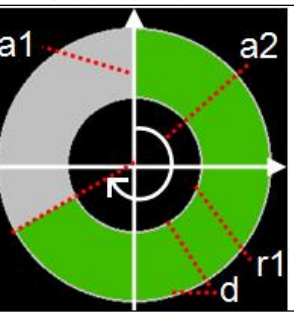


4.3.22.3 环形条变色模式

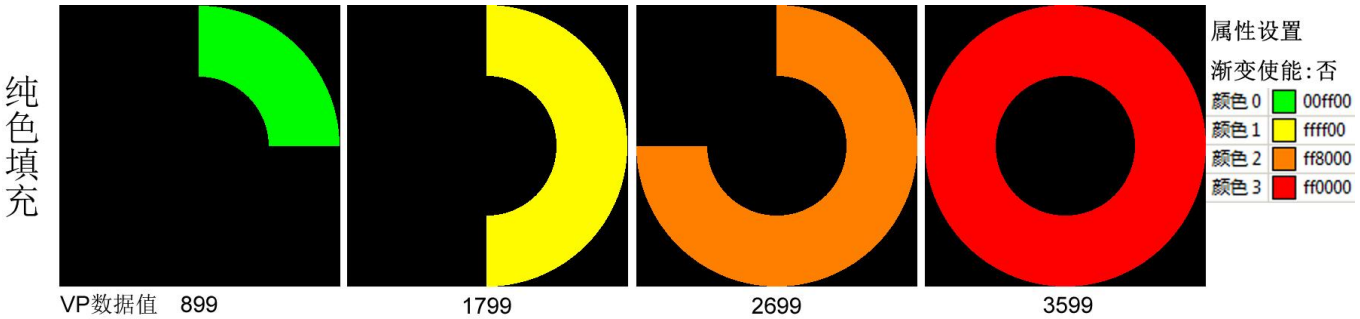




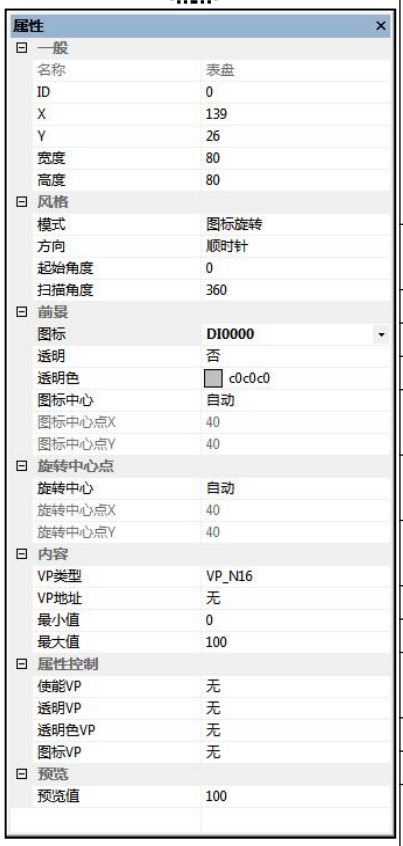
名称	表盘(Tachometer)
ID	编号
X/Y/宽/高	显示位置和大小(页面左上角坐标为 0,0)，外环直径与宽高相等且联动
模式	指针表盘：固件根据设定值自绘指针和表盘 环形条混色：固件根据设定值自绘渐变环形条 环形条变色：固件根据设定值自绘相应色阶的环形条 图标旋转：根据设定角度旋转图片 图标开口：根据角度百分比显示相应的图片内容 预合图标旋转：软件生成图片群，根据设定显示相应的图片 预合图标开口：软件生成图片群，根据设定显示相应的图片
方向	顺时针：顺时针旋转 逆时针：逆时针旋转
起始角度	如右图 a1 所示(单位 1 度)
扫描角度	如右图 a2 所示(单位 1 度)
底色显示	是：显示底色 否：不显示底色
底色	底盘的颜色
内环半径	如右图内环 r1 所示
边线宽度	如右图边线所示 d=2
步进值	环形条中每个颜色块的角度(单位 1 度)
间隔值	每两个颜色块之间的间隔角度(单位 1 度)
色阶数量	设置颜色的总数
色阶	设置每种颜色的颜色值
旋转中心	自动：自动设置旋转中心，不能修改，默认为圆心 高级：自定义旋转中心，可手动修改
旋转中心点 X	设置旋转中心点的位置坐标，控件左上角坐标为 0,0
旋转中心点 Y	设置旋转中心点的位置坐标，控件左上角坐标为 0,0
VP 类型	VP_N16: 16 位数字变量 VP_N32: 32 位数字变量 VP_REG:系统寄存器变量
VP 地址	变量 VP (16/32 位数字变量和系统寄存器变量)
最小值	VP 变量数据最小值,值范围: -2147483648~ 2147483647
最大值	VP 变量数据最大值,值范围: -2147483648~ 2147483647
使能 VP	往 VP 变量中写入数据可控制隐藏/显示 显示：0x0001；隐藏：0x0000
颜色数组 VP	往 VP 变量中写入数据可改变颜色数组里的值 数据格式: RGB565
预览值	预览(仅提供预览显示，非赋初值)

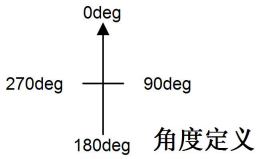


注：示例效果



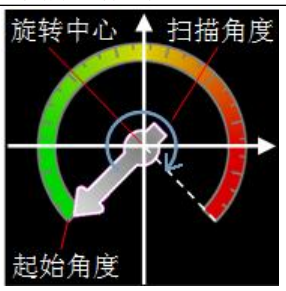
4.3.22.4 图标旋转模式





角度定义

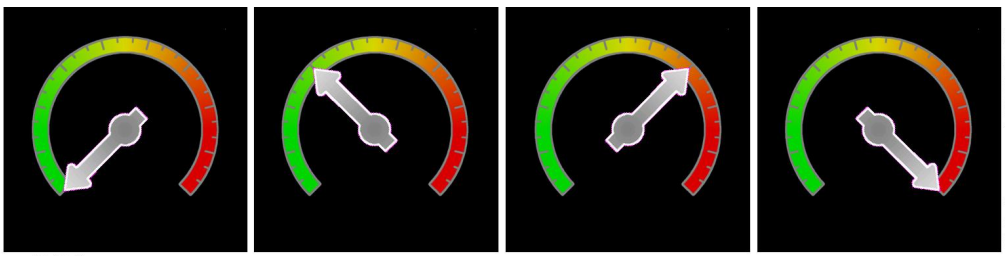
名称	表盘(Tachometer)
ID	编号
X/Y/宽/高	显示位置和大小(页面左上角坐标为 0,0)
模式	指针表盘：固件根据设定值自绘指针和表盘 环形条混色：固件根据设定值自绘渐变环形条 环形条变色：固件根据设定值自绘相应色阶的环形条 图标旋转：根据设定角度旋转图片 图标开口：根据角度百分比显示相应的图片内容 预合图标旋转：软件生成图片群，根据设定显示相应的图片 预合图标开口：软件生成图片群，根据设定显示相应的图片
方向	顺时针：顺时针旋转 逆时针：逆时针旋转
起始角度	如右图起始角度所示(单位 1 度)
扫描角度	如右图扫描角度所示(单位 1 度)
图标	选择图标
透明	是：图片中与透明色相同的颜色不显示 否：图片全部显示
透明色	指针图片中透明掉的颜色 数据格式：RGB565
图标中心	自动：自动设置图片中心；不能修改，与旋转中心重合，如上图所示 高级：自定义图片中心；可手动修改
图标中心点 X	设置图标中心点位置坐标
图标中心点 Y	设置图标中心点位置坐标
旋转中心	自动：自动设置旋转中心；不能修改，默认为图片中心，如上图所示 高级：自定义旋转中心；可手动修改
旋转中心点 X	设置旋转中心点的位置坐标，控件左上角坐标为 0,0
旋转中心点 Y	设置旋转中心点的位置坐标，控件左上角坐标为 0,0
VP 类型	VP_N16: 16 位数字变量 VP_N32: 32 位数字变量 VP_REG:系统寄存器变量
VP 地址	变量 VP (16/32 位数字变量和系统寄存器变量)
最小值	旋转范围最小值,值范围: -2147483648~ 2147483647
最大值	旋转范围最大值,值范围: -2147483648~ 2147483647
使能 VP	往 VP 变量中写入数据可控制隐藏/显示 显示：0x0001 ;隐藏：0x0000
透明 VP	往 VP 变量中写入数据可改变"透明"属性 透明：0x0001，不透明：0x0000
透明色 VP	往 VP 变量中写入数据可改变属性"透明色"的颜色 数据格式: RGB565，例: 红色：0xF800
图标 VP	往 VP 变量中写入数据可改变所选的属性"首图标"ID 号 有效值: 0 ~ 9999
预览值	预览(仅提供预览显示,非赋初值)



注：示例效果

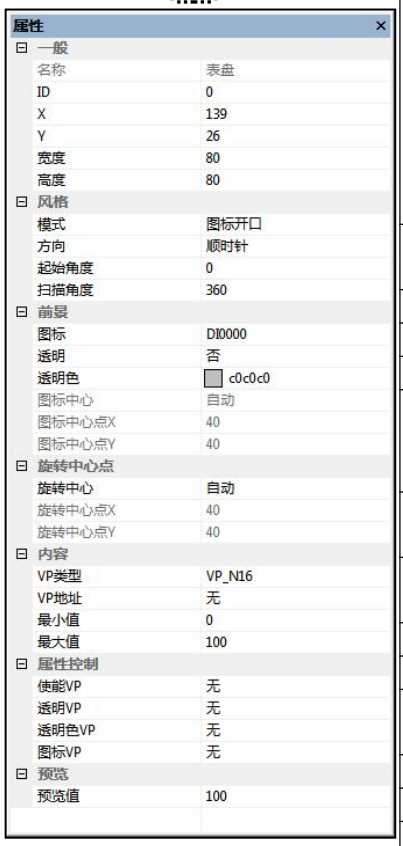

DI0000

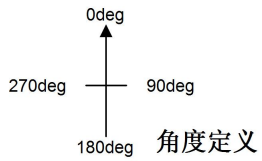
旋转模式



VP数据值 0 899 1799 2699

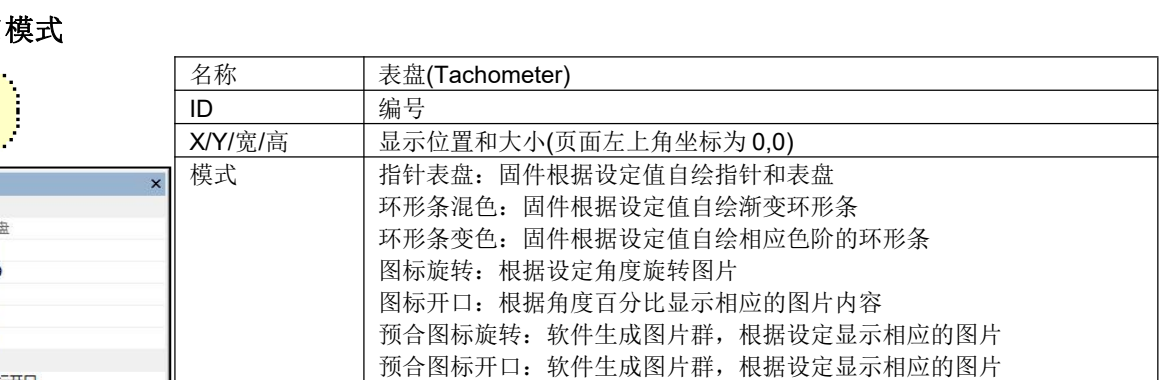
4.3.22.5 图标开口模式



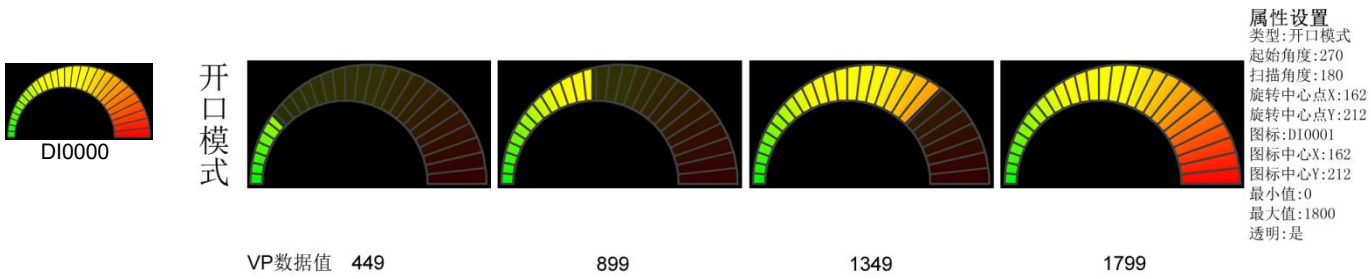


角度定义

名称	表盘(Tachometer)
ID	编号
X/Y/宽/高	显示位置和大小(页面左上角坐标为 0,0)
模式	指针表盘：固件根据设定值自绘指针和表盘 环形条混色：固件根据设定值自绘渐变环形条 环形条变色：固件根据设定值自绘相应色阶的环形条 图标旋转：根据设定角度旋转图片 图标开口：根据角度百分比显示相应的图片内容 预合图标旋转：软件生成图片群，根据设定显示相应的图片 预合图标开口：软件生成图片群，根据设定显示相应的图片
方向	顺时针：顺时针旋转 逆时针：逆时针旋转
起始角度	如右图起始角度所示(单位 1 度)
扫描角度	如右图扫描角度所示(单位 1 度)
图标	选择图标
透明	是：图片中与透明色相同的颜色不显示 否：图片全部显示
透明色	指针图片中透明掉的颜色 数据格式：RGB565
图标中心	自动：自动设置图片中心，不能修改 高级：自定义图片中心，可手动修改
图标中心点 X	图标中心点 X 坐标，控件左上角坐标为 0,0
图标中心点 Y	图标中心点 Y 坐标，控件左上角坐标为 0,0
旋转中心	自动：自动设置旋转中心，不能修改，默认为图片中心点 高级：自定义旋转中心，可手动修改，如上图旋转中心所示
旋转中心点 X	设置旋转中心点的位置坐标，控件左上角坐标为 0,0
旋转中心点 Y	设置旋转中心点的位置坐标，控件左上角坐标为 0,0
VP 类型	VP_N16: 16 位数字变量 VP_N32: 32 位数字变量 VP_REG: 系统寄存器变量
VP 地址	变量 VP (16/32 位数字变量和系统寄存器变量)
最小值	旋转范围最小值,值范围: -2147483648~ 2147483647
最大值	旋转范围最大值,值范围: -2147483648~ 2147483647
使能 VP	往 VP 变量中写入数据可控制隐藏/显示 显示：0x0001 ； 隐藏：0x0000
透明 VP	往 VP 变量中写入数据可改变"透明"属性 透明：0x0001，不透明：0x0000
透明色 VP	往 VP 变量中写入数据可改变属性"透明色"的颜色 数据格式：RGB565，例：红色= 0xF800
图标 VP	往 VP 变量中写入数据可改变所选的属性"首图标"ID 号 有效值：0 ~ 9999
预览值	预览(仅提供预览显示，非赋初值)

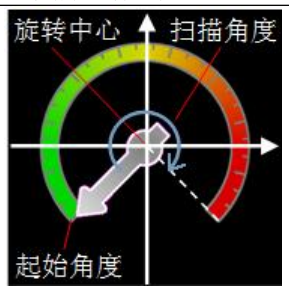


注：示例效果



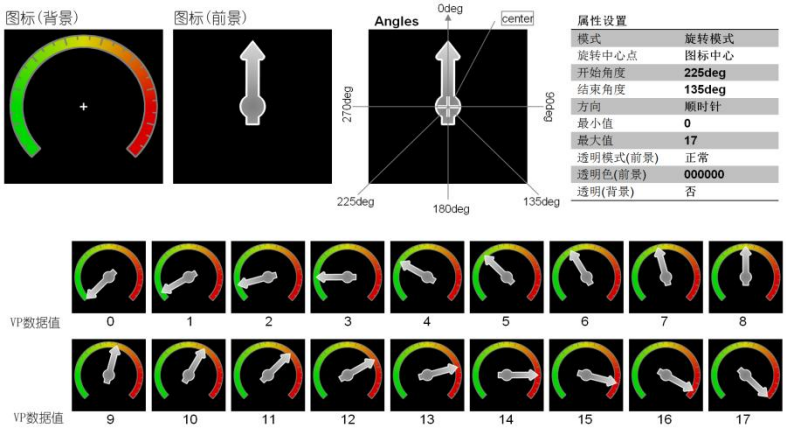
4.3.22.6 预合图标旋转模式

名称	表盘(Tachometer)
ID	编号
X/Y/宽/高	显示位置和大小(页面左上角坐标为 0,0)
模式	指针表盘：固件根据设定值自绘指针和表盘 环形条混色：固件根据设定值自绘渐变环形条 环形条变色：固件根据设定值自绘相应色阶的环形条 图标旋转：根据设定角度旋转图片 图标开口：根据角度百分比显示相应的图片内容 预合图标旋转：软件生成图片群，根据设定显示相应的图片 预合图标开口：软件生成图片群，根据设定显示相应的图片
方向	顺时针：顺时针旋转 逆时针：逆时针旋转
起始角度	如右图的起始角度所示
扫描角度	如右图的扫描角度所示
图标(背景)	底图
透明	是：图片中与透明色相同的颜色不显示 否：图片全部显示
透明色	图片中透明掉的颜色 数据格式: RGB565
图标(前景)	前景图标会根据旋转模式设定的相关参数与图标(背景)合成多个图标
透明模式	正常、变暗、变亮
透明色	图片中透明掉的颜色 数据格式: RGB565
旋转中心	自动：自动设置旋转中心，不能修改，默认为图片中心点 高级：自定义旋转中心，可手动修改 如上图旋转中心所示
旋转中心点 X	设置旋转中心点的位置坐标，控件左上角坐标为 0,0
旋转中心点 Y	设置旋转中心点的位置坐标，控件左上角坐标为 0,0
VP 类型	VP_N16: 16 位数字变量 VP_N32: 32 位数字变量 VP_REG: 系统寄存器变量
VP 地址	变量 VP (16/32 位数字变量和系统寄存器变量)
最小值	最小值对应"起始角度"状态下的图标
最大值	最大值对应"扫描角度"状态下的图标
使能 VP	往 VP 变量中写入数据可控制隐藏/显示 显示：0x0001 隐藏：0x0000
预览值	预览(仅提供预览显示,非赋初值)

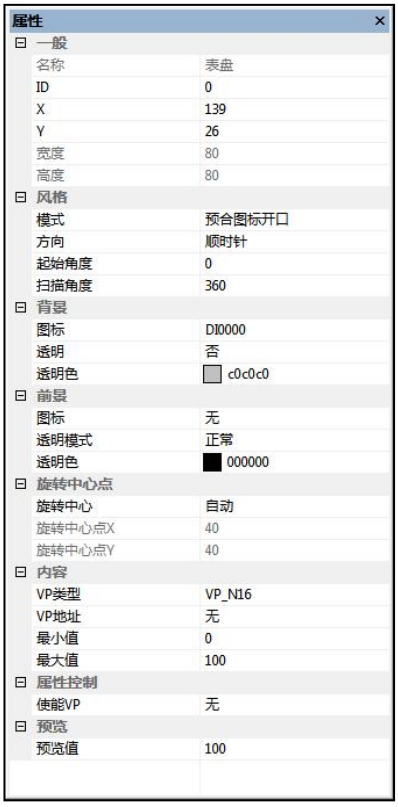


注：示例效果

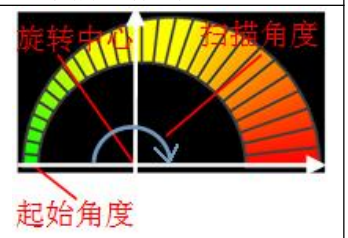
旋转模式



4.3.22.7 预合图标开口模式

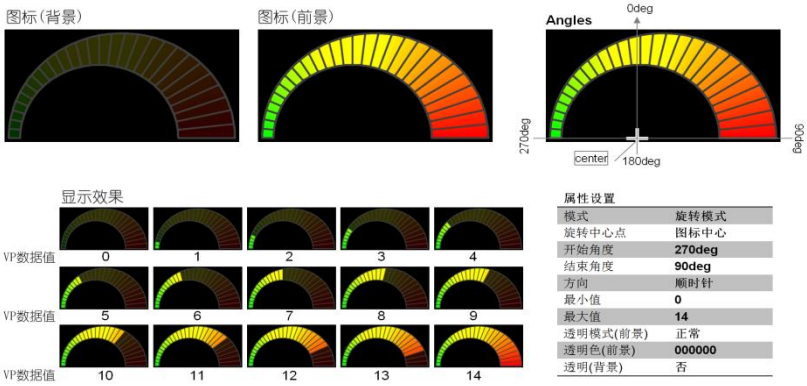


名称	表盘(Tachometer)
ID	编号
X/Y/宽/高	显示位置和大小(页面左上角坐标为 0,0)
模式	指针表盘：固件根据设定值自绘指针和表盘 环形条混色：固件根据设定值自绘渐变环形条 环形条变色：固件根据设定值自绘相应色阶的环形条 图标旋转：根据设定角度旋转图片 图标开口：根据角度百分比显示相应的图片内容 预合图标旋转：软件生成图片群，根据设定显示相应的图片 预合图标开口：软件生成图片群，根据设定显示相应的图片
方向	顺时针、逆时针
起始角度	如右图所示的起始角度所示
扫描角度	如右图所示的扫描角度所示
图标(背景)	底图
透明	是：图片中与透明色相同的颜色不显示 否：图片全部显示
透明色	图片中透明掉的颜色 数据格式：RGB565
图标(前景)	前景图标会根据旋转模式设定的相关参数与图标(背景)合成多个图标
透明模式	正常、变暗、变亮
透明色	图片中透明掉的颜色 数据格式: RGB565
旋转中心	自动：自动设置旋转中心，不能修改，默认为图片中心点 高级：自定义旋转中心，可手动修改，如上图旋转中心所示
旋转中心点 X	设置旋转中心点的位置坐标，控件左上角坐标为 0,0
旋转中心点 Y	设置旋转中心点的位置坐标，控件左上角坐标为 0,0
VP 类型	VP_N16: 16 位数字变量 VP_N32: 32 位数字变量 VP_REG: 系统寄存器变量
VP 地址	变量 VP (16/32 位数字变量和系统寄存器变量)
最小值	最小值对应"起始角度"状态下的图标
最大值	最大值对应"扫描角度"状态下的图标
使能 VP	往 VP 变量中写入数据可控制隐藏/显示 显示：0x0001 隐藏：0x0000
预览值	预览(仅提供预览显示，非赋初值)

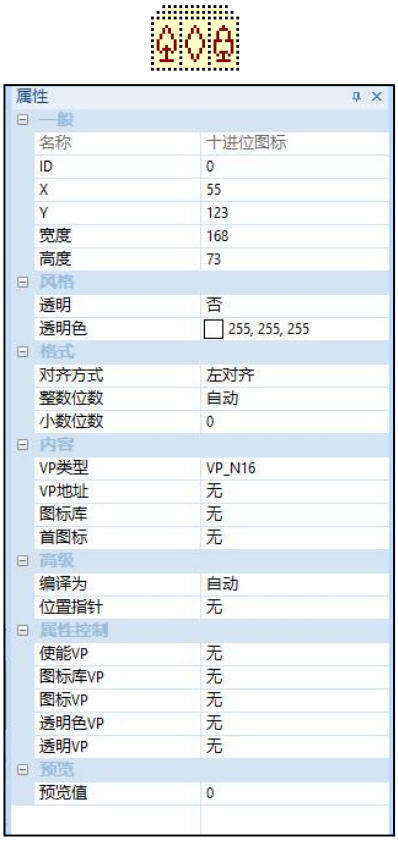


注：示例效果

开口模式

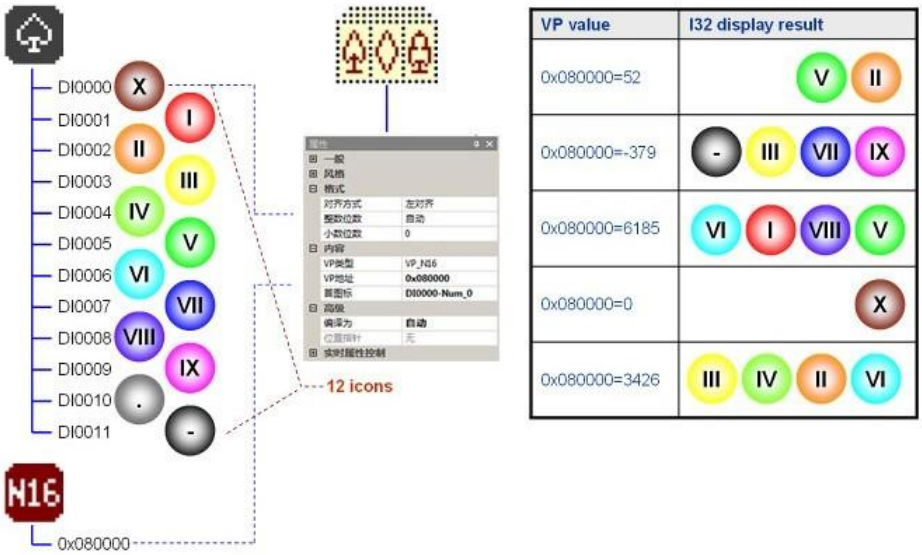


4.3.23 十进位图标

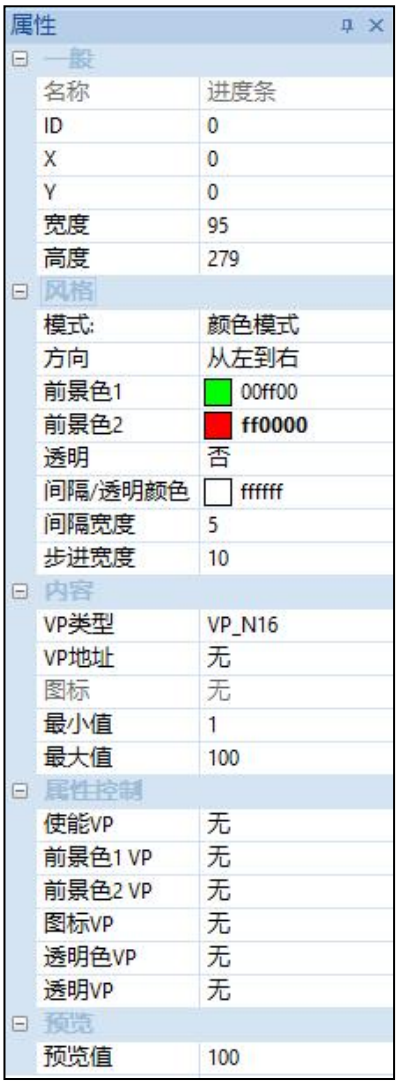


名称	十进位图标(Decimal Icon)
ID	编号
X/Y/宽/高	显示位置和大小(页面左上角坐标为 0,0)
透明	是：图片中与透明色相同的颜色不显示 否：图片全部显示
透明色	图片中透明掉的颜色
对齐方式	左对齐、右对齐、居中对齐
整数位数	整数位个数(自动：依实际数字位数显示) 不够位数时自动补零
小数位数	小数位个数 不够位数时自动补零
VP 类型	VP_N16(16 位数字变量) VP_N32(32 位数字变量) VP_REG(系统寄存器变量)
VP 地址	变量地址(16/32 位数字变量和系统寄存器变量)
图标库	工程资源栏创建图标库，选择 ICO001，仅限部分型号支持，比如新建工程的分辨率是 800x480，在工程设置里，选择设备型号为：HMT070ETD-C
首图标	ICON 首图标，从首图标 ID 号开始连续 12 图标 ID 分别对应" **0123456789.-** "
编译为	自动(依实际 VP 地址类型) I16(强制转为 16 位数字变量类型) I32(强制转为 32 位数字变量类型)
位置指针	保留
使能 VP	往 VP 变量中写入数据可控制十进位图标隐藏/显示 显示：0x0001，隐藏：0x0000
图标库 VP	往 VP 变量中写入数据更改图标库 ID 值范围：0 ~ 9999
图标 VP	往 VP 变量中写入数据可改变所选的属性"首图标"ID 号 有效值：0 ~ 9999
透明色 VP	往 VP 变量中写入数据可改变属性"透明色"的颜色 数据格式：RGB565，例：红色=0xF800
透明 VP	往 VP 变量中写入数据可改变"透明"属性 透明：0x0001 不透明：0x0000
预览值	预览(仅提供预览显示，非赋初值)

注：示例效果

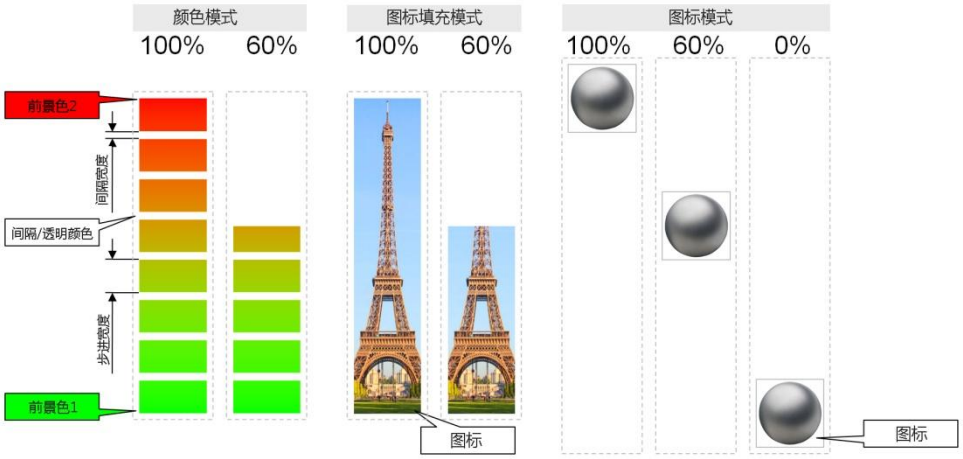


4.3.24 进度条

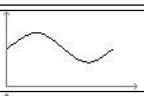
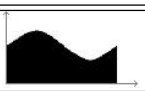
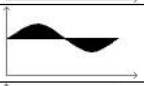
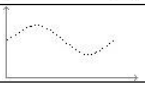


名称	进度条(Progress Bar)
ID	编号
X/Y/宽/高	显示位置和大小(页面左上角坐标为 0,0)
模式	颜色模式、图标模式、图标填充模式
方向	从左到右、从右到左、从上到下、从下到上
前景色 1	前景色 1 为进度条最小值开始时颜色 (颜色模式下有效)
前景色 2	前景色 2 为进度条 100%结束时颜色 (颜色模式下有效)
透明	图标是否透明 (图标模式和图标填充模式下有效)
间隔/透明色	两个颜色块之间间隔的颜色 或 图标模式下图标的透明色
间隔宽度	两个颜色块之间的间隔宽度
步进宽度	每个颜色块的宽度
VP 类型	VP_N16(16 位数字变量) VP_N32(32 位数字变量) VP_REG(系统寄存器变量)
VP 地址	变量地址(16/32 位数字变量和系统寄存器变量)
图标	进度图标(图标模式和图标填充模式下有效)
最小值	进度值范围最小值，值范围：0 ~ 2147483647
最大值	进度值范围最大值，值范围：0 ~ 2147483647
使能 VP	往 VP 变量中写入数据可控制进度条隐藏/显示 显示：0x0001，隐藏：0x0000
前景色 1VP	往 VP 变量中写入数据可改变属性"前景色 1"的颜色 数据格式：RGB565
前景色 2VP	往 VP 变量中写入数据可改变属性"前景色 2"的颜色 数据格式：RGB565
图标 VP	往 VP 变量中写入数据可改变所选的属性"图标"ID 号 有效值：0 ~ 9999 注意：更改后的 ID 一定要存在，且和当前设置的图标要大小相同
透明色 VP	往 VP 变量中写入数据可改变属性"间隔/透明色"的颜色 数据格式：RGB565
透明 VP	往 VP 变量中写入数据可改变"透明"属性 透明：0x0001 不透明：0x0000
预览值	预览(仅提供预览显示，非赋初值)

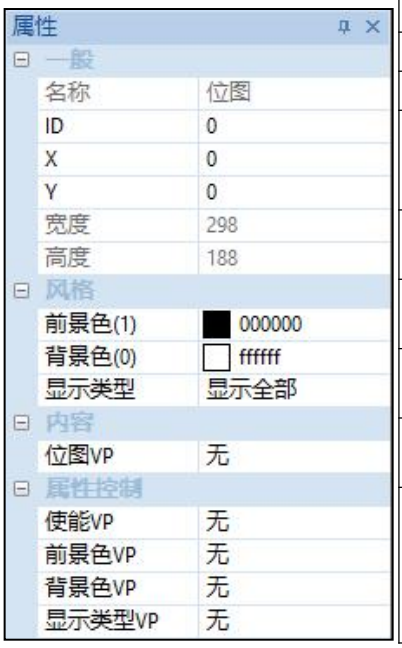
注：模式(示例效果)



4.4.25 曲线

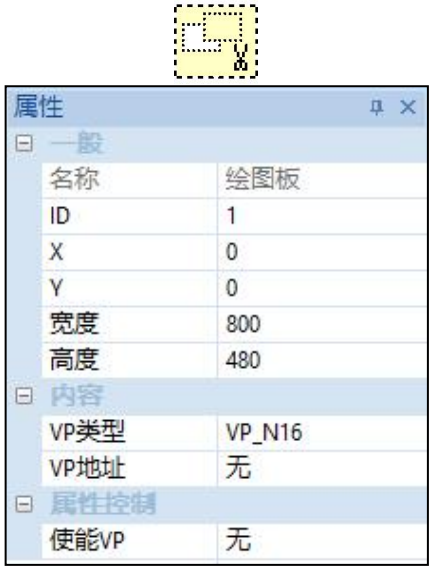
名称	曲线(Graph)
ID	编号
X/Y/宽/高	显示位置和大小(页面左上角坐标为 0,0)
前景色	曲线的颜色
点宽	曲线线条宽(X 方向)
点高	曲线线条高(Y 方向)，点宽/点高属性用来设置曲线的粗细
曲线类型	连线  底部填充  区域填充  点  顶部填充 
曲线 VP	曲线数据 VP 地址
最小值	曲线数据(点)最小值，最小值显示在控件框底部
最大值	曲线数据(点)最大值，最大值显示在控件框顶部
使能 VP	可通过 VP 中的数据控制曲线隐藏/显示 显示：0x0001，隐藏：0x0000
前景色 VP	往 VP 变量中写入数据可改变曲线的颜色 数据格式：RGB565
最小值 VP	往 VP 变量中写入数据可改变曲线的最小值 值范围：-32768 ~ +32767
最大值 VP	往 VP 变量中写入数据可改变曲线的最大值 值范围：-32768 ~ +32767
标尺 1 VP	往 VP 变量中写入数据可画一条标线在曲线上 - 标线长度与控件框宽度相同 - 标线的 Y 轴位置取决于设置的数值(最小值≤数值≤最大值)
标尺 2 VP	同标尺 1 功能

4.3.26 位图



名称	位图(Bitmap)
ID	编号
X/Y/宽/高	显示位置和大小(页面左上角坐标为 0,0)
前景色(1)	"Bits=1"的颜色
背景色(0)	"Bits=0"的颜色
显示类型	显示全部- 前景色(1)和背景色(0)都显示 前景色(1)- 仅显示"Bits=1"数据对应的颜色 背景色(0)- 仅显示"Bits=0"数据对应的颜色
位图 VP	位图 VP 变量 通过指令往 VP 变量中写入数据可实时更新位图显示内容
使能 VP	可通过 VP 中的数据控制位图是隐藏/显示 显示：0x0001，隐藏：0x0000
前景色 VP	往 VP 变量中写入数据可改变前景色(1)的颜色 数据格式：RGB565
背景色 VP	往 VP 变量中写入数据可改变背景色(0)的颜色 数据格式：RGB565
显示类型 VP	往 VP 变量中写入数据可改变位图的显示类型 显示全部：0x0000 前景色(1) ：0x0001 背景色(0) ：0x0002

4.3.27 绘图板



名称	绘图板(DrawPad)
ID	编号
X/Y/宽/高	显示位置和大小(页面左上角坐标为 0,0)
VP 类型	VP_N16(十六位数字变量)
VP 地址	变量地址 往此 VP 地址变量中写的数据必须按照一定的格式即可绘图 详细请参考"绘图板命令数据格式"
使能 VP	可通过 VP 中的数据控制绘图板隐藏/显示示 显示：0x0001 隐藏：0x0000

注: 绘图板命令数据格式

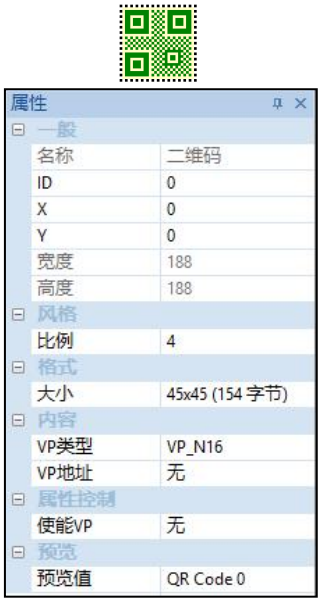
绘图板数据包格式

VP地址	内容	描述
VP	CMD	绘图指令码，不同的指令码表示绘制的图形不同
VP+2	Data_Pack_Num_Max	最大数据包数目：连线指令(0x0002)，定义为连线线条数目(顶点数－1)
VP+4	DATA_Pack	数据

绘图板命令说明

指令码	名称	参数设置			描述
		相对地址	字节大小	定义	
0x0001	画点	0x00	4	xh,xl,yh,yl	指定坐标画点
		0x04	2	Color	颜色(RGB565格式，例：红色为0xF800)
0x0002	端点连线	0x00	2	Color	颜色(RGB565格式)
		0x02	4	x0h,x0l,y0h,y0l	第0个端点坐标
		0x06	4	X1h,x1l,y1h,y1l	第2个端点坐标
		:	4	:	:
		0x02+4*n	4	xnh,xnl,ynh,ynl	第n个端点坐标
0x0003	画线	0x00	4	xsh,xsl,ysh,ysl	起始端点坐标
		0x04	4	xeh,xel,yeh,yel	终点端点坐标
		0x08	2	Color	颜色(RGB565格式)
0x0004	画矩形	0x00	4	xsh,xsl,ysh,ysl	左上角XY坐标
		0x04	4	xeh,xel,yeh,yel	右下角XY坐标
		0x08	2	Color	颜色(RGB565格式)
0x0005	填充矩形	0x00	4	xsh,xsl,ysh,ysl	左上角XY坐标
		0x04	4	xeh,xel,yeh,yel	右下角XY坐标
		0x08	2	Color	颜色(RGB565格式)
0x0006	显示剪切的背景图区域图像	0x00	2	Page_IDh, Page_IDl	页面ID
		0x02	4	xsh,xsl,ysh,ysl	左上角XY坐标
		0x06	4	xeh,xel,yeh,yel	右下角XY坐标
		0x0a	4	xh,xl,yh,yl	剪切后的数据显示位置XY坐标
0x0007	显示图标	0x00	4	xh,xl,yh,yl	显示位置XY坐标
		0x04	2	IMG_ICO_ID	图标ID
0x0008	显示文本	0x00	4	xh,xl,yh,yl	显示位置XY坐标
		0x04	2	Color	文本颜色(RGB565格式)
		0x06(0x06H)	1	FONT_ID	字库ID
		0x07(0x06L)	1	Text_len	字符长度(按字节计算)
		0x08	N	Text_STRING	文本数据

4.3.28 二维码



名称	二维码(QRCode)
ID	编号
X/Y/宽/高	显示位置和大小(页面左上角坐标为 0,0)
比例	显示放大比例(1 至 8 倍)
大小	二维码存储数据大小(二维码版本)

大小(二维码版本)	说明
29x29(53 字节)	最大可显示 53 字节信息
37x37(106 字节)	最大可显示 106 字节信息
45x45(154 字节)	最大可显示 154 字节信息
73x73(458 字节)	最大可显示 458 字节信息
109x109(1091 字节)	最大可显示 1091 字节信息

VP 类型	VP_STR(字符串变量)：每个字符串变量可存储 127Byte 有效数据. VP_N16(16 位数字变量)：数据的最后一个字节必须为'\0' (0x00)
VP 地址	变量地址(通过指令往 VP 地址中写入数据可实时更新二维码显示)
使能 VP	可通过 VP 中的数据控制二维码控件隐藏/显示 显示：0x0001，隐藏：0x0000
预览值	预览(仅提供预览显示，非赋初值)

4.3.29 页面属性

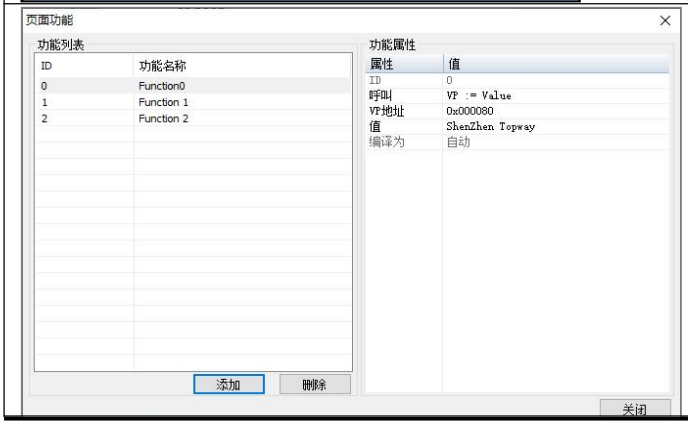
名称	显示页面的名称、ID、页面宽高
ID	编号
宽、高	页面分辨率
颜　色	页面颜色(底色)，无背景图时页面显示此颜色
背景图	背景图(IMG_BKG，图片编号：0000-1999)
目标	(延时后)跳转的页面
延时(秒)	若页面设置的自动跳转属性,则页面显示后自动延时计时 计时时间到后跳转到目标属性设定的页面.

4.3.30 页面功能

页面功能类似于页面的一个初始化的功能函数；
当页面每一次被显示时，都会执行一次页面功能；
每个页面都有一个页面功能。

在页面功能中可以对变量 VP 做赋值运算操作。

注：在页面中的空白处右击鼠标，弹出的菜单中
可以调出"页面功能"设置窗口。



页面功能列表

ID	编号
呼叫	可执行的操作(*1)
VP 地址	VP 地址(VP_STR, VP_N16, VP_N32, VP_N64)
值	写入 VP 地址中的数据(十进制数值)
编译为	把 VP 地址强转为其他 VP 地址类型

例：由 A 页面跳转切换到 B 页面,B 页面的"页面功能"会被
立刻执行一次,B 画面显示过程中不再执行"页面功能"

注：页面功能可执行操作列表

名称	描述	名称	描述
VP:= Value	把 Value 数据写入到 VP 地址中	Byte0(VP):= Byte0(Value)	复制 Value 的最低字节数据到 VP 的 Byte0 位置
VP:= VP + Value	VP 中的数据与 Value 相加，结果再写入 VP	Byte1(VP):= Byte0(Value)	复制 Value 的最低字节数据到 VP 的 Byte1 位置
VP:= VP - Value	VP 中的数据与 Value 相减，结果再写入 VP	Byte2(VP):= Byte0(Value)	复制 Value 的最低字节数据到 VP 的 Byte2 位置
VP:= VP * Value	VP 中的数据与 Value 相乘，结果再写入 VP	Byte3(VP):= Byte0(Value)	复制 Value 的最低字节数据到 VP 的 Byte3 位置
VP:= VP/Value	VP 中的数据与 Value 相除，结果再写入 VP	Bit0(VP):= LSB(Value)	复制 Value 的 bit0 数据到 VP 地址的 Bit0 位
BUFF:= VP	VP 地址中的数据复制到 0x00000000_BUFF 中.	Bit1(VP):= LSB(Value)	复制 Value 的 bit0 数据到 VP 地址的 Bit1 位
VP:= BUFF	0x00000000_BUFF 中的数据写入到 VP 地址中	Bit2(VP):= LSB(Value)	复制 Value 的 bit0 数据到 VP 地址的 Bit2 位
VP:= DelLastChar(VP)	删除键,功能等于键盘的 Backspace 按键	Bit3(VP):= LSB(Value)	复制 Value 的 bit0 数据到 VP 地址的 Bit3 位
VP:= Concatenate(VP,Value)	VP 数据中追加一个字符(Value 表示一个字符)	Bit4(VP):= LSB(Value)	复制 Value 的 bit0 数据到 VP 地址的 Bit4 位
VP:= VP XOR Value	VP 中的数据与 Value 进行"异或"，结果再写入到	Bit5(VP):= LSB(Value)	复制 Value 的 bit0 数据到 VP 地址的 Bit5 位
Bit7(VP):= LSB(Value)	复制 Value 的 bit0 数据到 VP 地址的 Bit7 位	Bit6(VP):= LSB(Value)	复制 Value 的 bit0 数据到 VP 地址的 Bit6 位

4.4 呼叫功能

″呼叫″是触摸键、虚拟键、页面可调用的一个子功能， 通过触摸键或虚拟键控件可调用键盘、菜单、VP 变量赋值运算操作、 键盘按键等功能。

4.4.1 呼叫 - 键盘/菜单

名称	描述
数字键盘(PIP)	弹出一个数字键盘，按下"OK"键后录入的值写入 VP 并同时写入到串口
英文键盘(PIP)	弹出一个英文键盘，按下"OK"键后录入的字符写入 VP 并同时写入到串口
密码键盘(PIP)	弹出一个密码键盘，按下"OK"键后录入的值写入 VP 同时以字符串方式写入到串口
RTC 键盘(PIP)	弹出一个日期时间设定键盘，按下"OK"键后录入的日期时间写入到串口
PIP 菜单	弹出一个菜单窗口(可做菜单窗口、提示、警告或其他窗口等) 按下" VP:=Value,Enter "键后录入的数据写入 VP 并同时写入到串口
PIP 键盘	弹出一个可录入 ASCII 字符的键盘，按下"Enter"键后录入的数据写入 VP 并同时写入到串口
PIP 数字键盘	弹出一个可录入浮点数的数字键盘，按下"Enter"键后录入的数据写入 VP 并同时写入到串口
PIP 时钟设定	弹出一个可录入日期与时间的键盘，按下"Enter"键后录入的日期时间写入到串口并关闭键盘
PIP 中文键盘	弹出一个可录入中文的键盘，按下"Enter"键后录入的中文写入到串口并关闭键盘
数字键盘	弹出一个数字键盘，按下"OK"键后录入的值写入 VP 并同时写入到串口
密码键盘	弹出一个数字键盘(录入时显示*号，最大 9 个字符) 按下"OK"键后录入的值写入 VP 同时以字符串方式写入到串口
英文键盘	弹出一个英文键盘，按下"OK"键后录入的字符写入 VP 并同时写入到串口
中文键盘	弹出一个中文键盘(可录入中文和 ASCII 字符) 按下"OK"键后录入的字符写入 VP 并同时写入到串口
背光设定	弹出背光设定窗口，设定当前背光亮度(断电后再上电以工程配置亮度参数为准)
日期时钟设定	弹出一个日期时间设定键盘，按下"OK"键后录入的日期时间写入到串口
单杆数值输入	弹出一个单水平滑杆的窗口，通过滑杆可更改数字 VP 的数值 按下"OK"键后数字 VP 和数值写入到串口
双杆数值录入	弹出一个双水平滑杆的窗口，通过滑杆可更改数字 VP 的数值 按下"OK"键后数字 VP 和数值写入到串口
列表显示	弹出一个列表(1 列)窗口，显示字符串 VP 中的数据
可滑动列表显示	弹出一个可上下滑动的列表(1 列)窗口，显示字符串 VP 中的数据

4.4.2 呼叫 - 按键

名称	描述
Enter	确认键 (按下后 PIP 键盘/菜单窗口关闭，并返回录入或选择的数据到串口)
Esc	取消键 (按下后 PIP 键盘/菜单窗口关闭，无返回数据)
CapsLock	大小写切换键 (同键盘的 Caps Lock 键)
VP:=Value,Enter	赋值确认键 (按下后 PIP 键盘/菜单窗口关闭,并返回 VP 和 Value 值到串口)
光标左移	光标向左移动一个字符
光标右移	光标向右移动一个字符
VP:=删除尾字符(VP)	删除键,功能等于键盘的 Backspace 按键
VP:=连接(VP,Value)	VP 的数据中追加一个字符(Value 表示一个字符)
Buf:=Con(Buff,Cap/Nom(Byte0/Byte1))	按键 (表示具体字符的按键) - "标题/值" 属性中的数据必须为十六进制,比如: 0x4161) - CapsLock 未按下时，低字节数据有效. 例:0x41**61** 时，表示字符'**a**' - CapsLock 按下时，高字节数据有效. 例:0x**41**61 时，表示字符'**A**'

4.4.3 呼叫 - 运算操作

名称	描述
VP:= Value	把 Value 数据写入到 VP 地址中，Value 值不超过 59 字节
VP:= VP + Value	VP 地址中的数据与 Value 相加，相加结果再写入到 VP 地址中
VP:= VP + Value, loop	VP 地址中的数据与 Value 相加，相加结果再写入到 VP 地址中， MIN≤VP 数值≤MAX 相加结果超过 MAX 设定时,把 MIN 值写入到 VP 地址中.
VP:= VP - Value	VP 地址中的数据与 Value 相减，相减结果再写入到 VP 地址中
VP:= VP - Value, loop	VP 地址中的数据与 Value 相减，相减结果再写入到 VP 地址中， MIN≤VP 数值≤MAX 相减结果小于 MIN 设定时,把 MAX 值写入到 VP 地址中.
VP:= VP * Value	VP 地址中的数据与 Value 相乘，相乘结果再写入到 VP 地址中
VP:= VP/Value	VP 地址中的数据与 Value 相除，相除结果再写入到 VP 地址中
VP:= VP XOR Value	VP 地址中的数据与 Value 进行"异或"操作，操作结果再写入到 VP 地址中
BUFF:= VP	VP 地址中的数据复制到 0x00000000_BUFF 中. 若 VP 地址类型为 16/32/64 位数字变量地址,则把数字变量字符串再复制到 BUFF 中
VP:= BUFF	0x00000000_BUFF 中的数据写入到 VP 地址中 若 VP 地址类型为 16/32/64 位数字变量地址,则把 BUFF 中的数据变为数字再写入到 VP 中
Byte0(VP):= Byte0(Value)	复制 Value 的最低字节数据到 VP 地址 Byte0 位置 例：Value=0xAABB, VP 地址中数据=0x11223344，运算后 VP 地址中的数据=0x112233BB
Byte1(VP):= Byte0(Value)	复制 Value 的最低字节数据到 VP 地址 Byte1 位置 例：Value=0xAABB, VP 地址中数据=0x11223344，运算后 VP 地址中的数据=0x1122BB44
Byte2(VP):= Byte0(Value)	复制 Value 的最低字节数据到 VP 地址 Byte2 位置 例：Value=0xAABB, VP 地址中数据=0x11223344，运算后 VP 地址中的数据=0x11BB3344
Byte3(VP):= Byte0(Value)	复制 Value 的最低字节数据到 VP 地址 Byte3 位置 例：Value=0xAABB, VP 地址中数据=0x11223344，运算后 VP 地址中的数据=0xBB223344
Bit0(VP):= LSB(Value)	复制 Value 的 bit0 数据到 VP 地址的 Bit0 位 例：Value=0x0001, VP 地址中数据=0x00000000，运算后 VP 地址中的数据=0x00000001
Bit1(VP):= LSB(Value)	复制 Value 的 bit0 数据到 VP 地址的 Bit1 位 例：Value=0x0001, VP 地址中数据=0x00000000，运算后 VP 地址中的数据=0x00000002
Bit2(VP):= LSB(Value)	复制 Value 的 bit0 数据到 VP 地址的 Bit2 位 例：Value=0x0001, VP 地址中数据=0x00000000，运算后 VP 地址中的数据=0x00000004
Bit3(VP):= LSB(Value)	复制 Value 的 bit0 数据到 VP 地址的 Bit3 位 例：Value=0x0001, VP 地址中数据=0x00000000，运算后 VP 地址中的数据=0x00000008

Bit4(VP):= LSB(Value)	复制 Value 的 bit0 数据到 VP 地址的 Bit4 位 例：Value=0x0001, VP 地址中数据=0x00000000，运算后 VP 地址中的数据=0x00000010
Bit5(VP):= LSB(Value)	复制 Value 的 bit0 数据到 VP 地址的 Bit5 位 例：Value=0x0001, VP 地址中数据=0x00000000，运算后 VP 地址中的数据=0x00000020
Bit6(VP):= LSB(Value)	复制 Value 的 bit0 数据到 VP 地址的 Bit6 位 例：Value=0x0001, VP 地址中数据=0x00000000，运算后 VP 地址中的数据=0x00000040
Bit7(VP):= LSB(Value)	复制 Value 的 bit0 数据到 VP 地址的 Bit7 位 例：Value=0x0001, VP 地址中数据=0x00000000，运算后 VP 地址中的数据=0x00000080
Bitn(VP):= LSB(Value)	复制 Value 的 bit0 数据到 VP 地址的 Bitn 位 例：Value=0x0001, n=8,VP 地址中数据=0x00000000，运算后 VP 地址中的数据=0x00000100

4.5 Modbus 协议配置

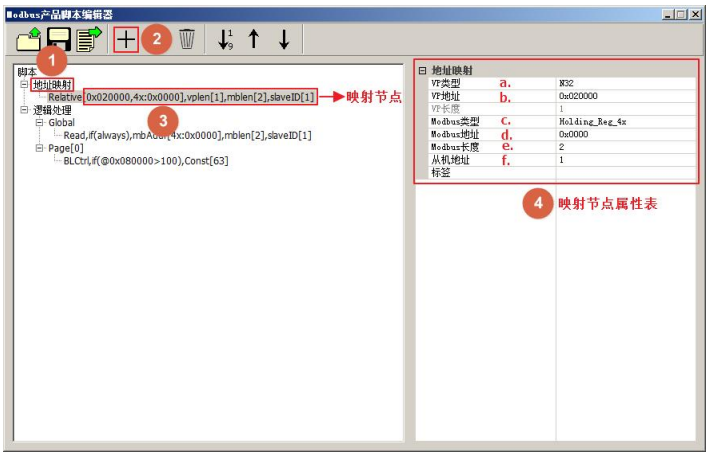
智能模块 RS485 硬件接口的产品一般是使用 Modbus-RTU 通讯协议。
屏幕可以作为主机模式(Master)或从机模式(Slave)。
作为主机模式(Master)的智能模块，制作工程时需要通过 SGTools 提供的”Modbus 产品脚本编辑器” 配置 VP 变量地址和 Modbus 地址的映射关系、读写寄存器/线圈的逻辑功能。

4.5.1 配置步骤

第一步 打开脚本编辑器

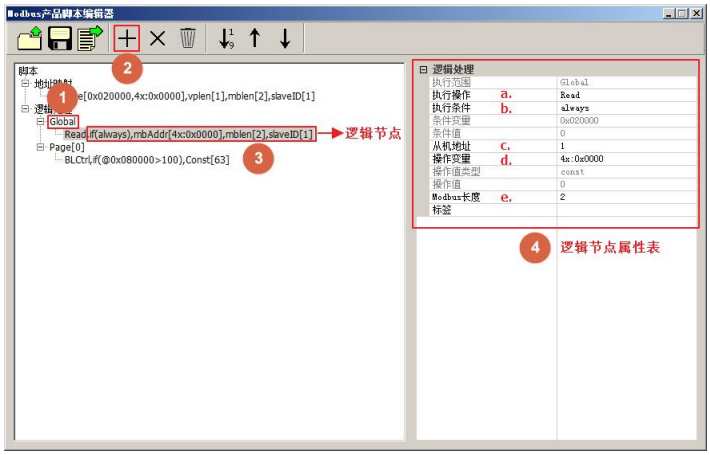
① 单击“工具”
② 单击“Modbus 产品脚本编辑器”

第二步 添加一个“映射节点”



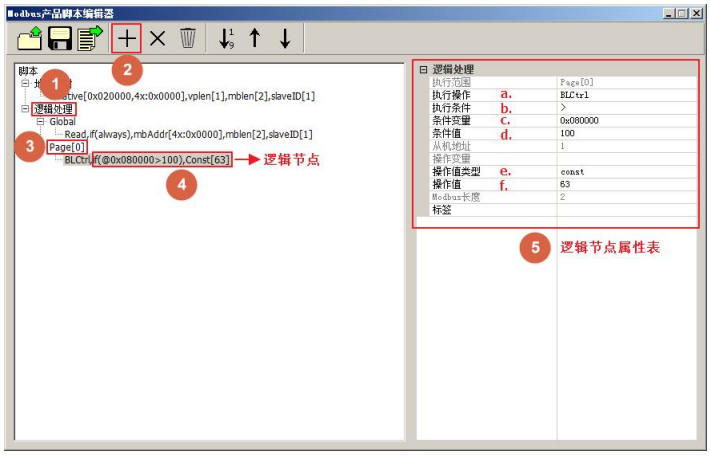
① 选中“地址映射”
② 单击按钮 ＋
③ 选中添加的“映射节点”
④ 设置映射节点属性
   a. 设置 VP 类型为 N32
   b. 选择合适的 VP 地址，这里设置为 0x020000
   c. 设置 Modbus 类型为 Holding_Reg_4x
   d. 设置 Modbus 地址为 0x0000
   e. 设置 Modbus 长度为 2
   f. 设置从机地址为 1

第三步 添加一个 Global 的“逻辑节点”



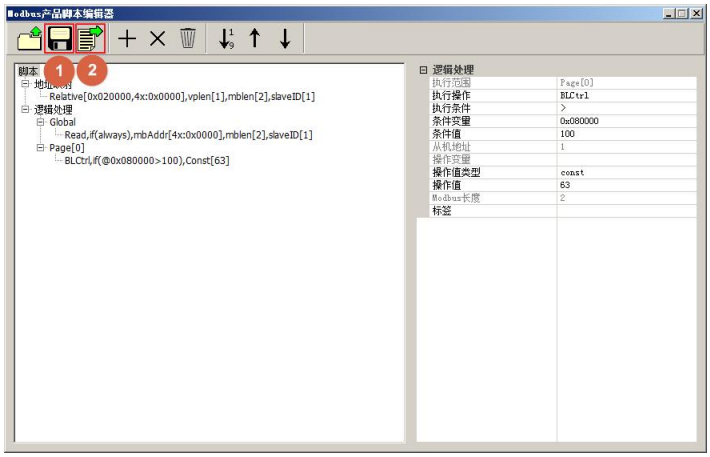
① 选中“Global”
② 单击按钮＋
③ 选中添加的“逻辑节点”
④ 设置逻辑节点属性
a. 设置执行操作为 Read
b. 设置执行条件为 always
c. 设置从机地址为 1
d. 选择操作变量为 4x：0x0000
e. 设置 Modbus 长度为 2

第四步 添加一个 Page[n]的“逻辑节点”



① 选中“逻辑处理”
② 单击按钮 ＋
③ 选中添加的“Page[0]”节点，
再次执行 ② 中操作
④ 选中添加的“逻辑节点”
⑤ 设置逻辑节点属性：
a. 设置执行操作为 BLCtrl
b. 设置执行条件为 >
c. 选择条件变量为 0x080000
d. 设置条件值为 100
e. 设置操作值类型为 const
f. 设置操作值为 63

第五步 保存和编译



① 单击“保存”按钮 
② 单击“编译”按钮 

4.5.2 功能介绍

4.5.2.1 工具栏

图标	名称	描述
	打开	选择.xml 文件进行导入，并将内容显示在脚本信息中
	保存	将当前脚本信息，保存在当前工程目录/Resource/Script.xml 中
	编译	将当前脚本信息，编译输出在当前工程目录/Resource/Modbus.xml 中
	添加	a. 选中“地址映射”，则添加子节点在尾部；选中“映射节点”，则添加节点在其后 b. 选中“逻辑处理”，则添加 Page[n]子节点，Page[n]为当前工程已存在的页面 c. 选中“Global/ Page[n]”，则添加子节点在尾部；选中“逻辑节点”，则添加节点在其后

	删除	删除被选中的节点(映射节点/逻辑节点/Page[n]节点)
	清空	清空所有脚本信息
	排序	将所有映射节点，根据 VP 地址从小到大排序
	上移	将被选中的节点(映射节点/逻辑节点)向上移动一格
	下移	将被选中的节点(映射节点/逻辑节点)向下移动一格

4.5.2.2 脚本信息

① 地址映射

将 VP 地址和 Modbus 地址的对应关系显示在节点内，格式如下：

[标签]Relative[VP 地址，mb 类型：mb 地址]，vplen[vp 地址长度]，mblen[mb 地址长度]，slaveID[从机地址]

② 逻辑处理

a. 全局操作： Global

b. 单页面操作：Page[n]，n 为页面 ID

c. 操作类型：

- Read(读数据)：主机模块读取从机寄存器的数据。
- BLCtrl(背光控制)：通过选择操作值类型 const 或者 variable，进行写常量或者写变量，来控制模块的背光亮度。
- BeepCtrl(蜂鸣器控制)：通过选择操作值类型 const 或者 variable，进行写常量或者写变量，来控制模块的蜂鸣器开关 (0 即关闭，非 0 即打开)
- Write(写数据) 通过选择操作值类型 const 或者 variable，进行常量值的赋值或者变量值的赋值，赋值给操作变量地址 (vpAddr[VP 地址])

操作类型	格式
Read	[标签] Read，if(@条件变量 执行条件 条件值)，mbAddr[mb 类型：mb 地址]，mblen[mb 地址长度]，slaveID[从机地址]
BLCtrl	[标签] BLCtrl，if(@条件变量 执行条件 条件值)，Const[常量值]/ Variable[VP 地址]
BeepCtrl	[标签] BeepCtrl，if(@条件变量 执行条件 条件值)，Const[常量值]/ Variable[VP 地址]
Write	[标签] Write，if(@条件变量 执行条件 条件值)，vpAddr[VP 地址]= Const[常量值]/ Variable[VP 地址]，vplen[vp 地址长度]

注： a. 执行条件包含："always",">","==","<","!=",">=","<="

b. 条件变量和条件值，仅执行条件">","==","<","!=",">=","<="有效，且条件变量为 VP 地址

4.5.2.3 属性设置

地址映射	
VP 类型	N16 ：16 位数字变量 N32 ：32 位数字变量 N64 ：64 位数字变量 G16 ：16 位曲线变量
VP 地址	选择数字变量，仅显示当前工程中存在的数字变量
VP 长度	VP 地址的长度，1~255
Modbus 类型	Coil_0x/Input_1x/Input_Reg_3x/Holding_Reg_4x
Modbus 地址	Modbus 地址，0x0000~0xFFFF 有效
Modbus 长度	Modbus 地址长度
从机地址	从机地址，1~247
标签	字符串标识符

类型	描述
Coil_0x	线圈
Input_1x	离散输入
Input_Reg_3x	输入寄存器
Holding_Reg_4x	保持寄存器

逻辑处理	
执行范围	显示执行范围，Global/Page[n]
执行操作	选择操作方式，Read/BLCtrl/BeepCtrl/Write
执行条件	"always",">","==","<","!=",">=","<="
条件变量	选择 VP 地址
条件值	输入十进制常量
从机地址	从机地址，1~247
操作变量	选择 VP 地址/Modbus 地址，Modbus 地址可手动输入；

Modbus 输入格式	描述
0x:0x0001	线圈地址 0x0001
1x:0x0001	离散输入地址 0x0001
3x:0x0001	输入寄存器地址 0x0001

VP 长度	VP 地址的长度，1~255
操作值	输入十进制常量/选择 VP 地址
Modbus 长度	Modbus 地址长度，1~10000
标签	字符串标识符

4x:0x0001	保持寄存器地址 0x0001

Page[n] 属性设置	
操作页面	更改操作页面 ID，仅工程中存在且未被使用的页面 ID 有效

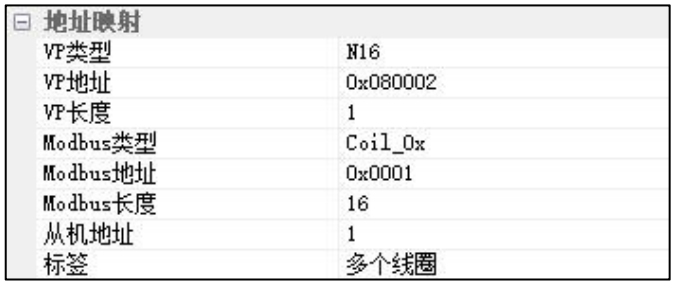
4.5.3 Modbus 功能码实现

主要实现以下 8 种常用功能码：
①“0x01”读线圈状态
②“0x02”读离散输入状态
③“0x04”读输入寄存器
④“0x03”读保持寄存器
⑤“0x05”写单个线圈状态
⑥“0x0F”写多个线圈状态
⑦“0x06”写单个保持寄存器
⑧“0x10”写多个保持寄存器
模块通信的前提条件是“地址映射”关系，对从机中的寄存器进行读写操作，首先需要映射从机的寄存器地址，将模块的 VP 地址映射到需要操作的从机 MB 地址，然后才能读写已映射的从机寄存器。模块 VP 地址与 Modbus 地址的数量

对应关系如下：
VP_N16(0x80000)：最多映射 16 个 0x 地址、16 个 1x 地址、1 个 3x 地址或 1 个 4x 地址
VP_N32(0x20000)：最多映射 32 个 0x 地址、32 个 1x 地址、2 个 3x 地址或 2 个 4x 地址
VP_N64(0x30000)：最多映射 64 个 0x 地址、64 个 1x 地址、4 个 3x 地址或 4 个 4x 地址

4.5.3.1 功能码“0x01”读线圈状态实现

地址映射：

映射节点编辑	映射关系说明
	将 1 个单位的 16 位数字变量“0x080002”对应 16 个从机 ID 为 1 的线圈状态，起始地址“0x0001”，相关信息： 1、从机地址 ：0x01 2、寄存器类型：Coil_0x 3、起始寄存器地址：0x0001 4、寄存器数量：16 (0x0010)

实现方式：编辑 Modbus 逻辑节点

逻辑节点编辑	报文数据解析
	根据图中所标记的参数“执行操作”、“从机地址”、“操作变量”以及 “Modbus 长度”可以自动获取以下信息： 1、从机地址 ：0x01 2、功能码 ：0x01 3、寄存器地址：0x0001 4、寄存器数量 ：0x0010 当满足“执行条件”时，模块发送报文： 01 01 00 01 00 10 6C 0C

4.5.3.2 功能码“0x02”读离散输入状态实现

地址映射：

映射节点编辑	映射关系说明
地址映射 VP类型 N16 VP地址 0x080004 VP长度 1 Modbus类型 Input_1x Modbus地址 0x0000 Modbus长度 16 从机地址 1 标签 离散输入	将 1 个单位的 16 位数字变量“0x080004”对应 16 个从机 ID 为 1 的离散输入状态，起始地址“0x0000”，相关信息： 1、从机地址 ：0x01 2、寄存器类型：Input_1x 3、起始寄存器地址：0x0000 4、寄存器数量：16 (0x0010)

实现方式：编辑 Modbus 逻辑节点

逻辑节点编辑	报文数据解析
逻辑处理 执行范围 Global 执行操作 Read 执行条件 always 条件变量 0x020000 条件值 0 从机地址 1 操作变量 1x:0x0000 VP长度 1 操作值类型 const 操作值 0 Modbus长度 16 标签 读离散输入	根据图中所标记的参数“执行操作”、“从机地址”、“操作变量”以及 “Modbus 长度”可以自动获取以下信息： 1、从机地址 ：0x01 2、功能码 ：0x02 3、寄存器地址：0x0000 4、寄存器数量 ：0x0010 当满足“执行条件”时，模块发送报文： 01 02 00 00 00 10 79 C6

4.5.3.3 功能码“0x04”读输入寄存器实现

地址映射：

映射节点编辑	映射关系说明
地址映射 VP类型 N16 VP地址 0x080006 VP长度 1 Modbus类型 Input_Reg_3x Modbus地址 0x0000 Modbus长度 1 从机地址 1 标签 输入寄存器	将 1 个单位的 16 位数字变量“0x080006”对应 1 个从机 ID 为 1 的输入寄存器“0x0000”，相关信息： 1、从机地址 ：0x01 2、寄存器类型：Input_Reg_4x 3、起始寄存器地址：0x0000 4、寄存器数量：1 (0x0001)

实现方式：编辑 Modbus 逻辑节点

逻辑节点编辑	报文数据解析
逻辑处理 执行范围 Global 执行操作 Read 执行条件 always 条件变量 0x020000 条件值 0 从机地址 1 操作变量 3x:0x0000 VP长度 1 操作值类型 const 操作值 0 Modbus长度 1 标签 读输入寄存器	根据图中所标记的参数“执行操作”、“从机地址”、“操作变量”以及 “Modbus 长度”可以自动获取以下信息： 1、从机地址 ：0x01 2、功能码 ：0x04 3、寄存器地址：0x0000 4、寄存器数量 ：0x0001 当满足“执行条件”时，模块发送报文： 01 04 00 00 00 01 31 CA

4.5.3.4 功能码“0x03”读保持寄存器实现

地址映射：

映射节点编辑	映射关系说明
地址映射 VP类型 N16 VP地址 0x080008 VP长度 2 Modbus类型 Holding_Reg_4x Modbus地址 0x0000 Modbus长度 2 从机地址 1 标签 保持寄存器	将 2 个连续的 16 位数字变量“0x080008/0x08000A”对应 2 个从机 ID 为 1 连续的保持寄存器“0x0000/0x0001”，相关信息： 1、从机地址 ：0x01 2、寄存器类型：Holding_Reg_4x 3、起始寄存器地址：0x0000 4、寄存器数量：2 (0x0002)

实现方式：编辑 Modbus 逻辑节点

逻辑节点编辑	报文数据解析
逻辑处理 执行范围 Global 执行操作 Read 执行条件 always 条件变量 0x020000 条件值 0 从机地址 1 操作变量 4x:0x0000 VP长度 1 操作值类型 const 操作值 0 Modbus长度 2 标签 读保持寄存器	根据图中所标记的参数“执行操作”、“从机地址”、“操作变量”以及 “Modbus 长度”可以自动获取以下信息： 1、从机地址 ：0x01 2、功能码 ：0x03 3、起始寄存器地址：0x0000 4、寄存器数量 ：0x0002 当满足“执行条件”时，模块发送报文： 01 03 00 00 00 02 C4 0B

4.5.3.5 功能码“0x05”写单个线圈状态实现

地址映射：

映射节点编辑	映射关系说明
地址映射 VP类型 N16 VP地址 0x080000 VP长度 1 Modbus类型 Coil_0x Modbus地址 0x0000 Modbus长度 1 从机地址 1 标签 单个线圈	将 1 个单位的 16 位数字变量“0x080000”对应 1 个从机 ID 为 1 的线圈类型寄存器“0x0000”。 当屏的变量“0x080000”被修改时，模块会发送报文修改相应地址的线圈类型寄存器，相关信息： 1、从机地址 ：0x01 2、寄存器类型：coil_0x 3、寄存器地址：0x0000 4、寄存器数量：1 (0x0001)

方式一、编辑 Modbus 逻辑节点实现

逻辑节点编辑	报文数据解析
逻辑处理 执行范围 Global 执行操作 Write 执行条件 always 条件变量 0x020000 条件值 0 从机地址 1 操作变量 0x080000 VP长度 1 操作值类型 const 操作值 1 Modbus长度 1 标签 写单个线圈	根据图中所标记的参数“执行操作”、“操作变量”、“操作值”以及变量“0x080000”的映射关系可以自动获取以下信息： 1、从机地址 ：0x01 2、功能码 ：0x05 3、寄存器地址：0x0000 4、写入数据 ：0xFF00(操作值为 1) 当满足“执行条件”时，模块发送报文： 01 05 00 00 FF 00 8C 3A

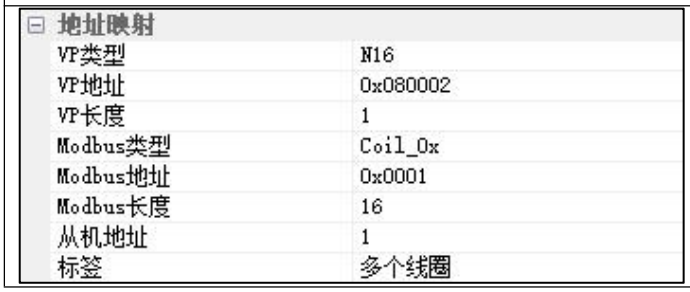
方式二、设置触摸键、虚拟键或页面功能属性实现

触摸键属性设置	报文数据解析
	根据图中所标记的参数“运算操作”、“VP 地址”、“标题/值”以及变量“0x080000”的映射关系可以自动获取以下信息： 1、从机地址 ：0x01 2、功能码 ：0x05 3、寄存器地址：0x0000 4、写入数据 ：0x0000(标题/值为 0) 当屏检测该触摸键被按下时，模块发送报文： 01 05 00 00 00 00 CD CA

注：触摸键运算操作“VP:=value”作用将“标题/值”的数据写入“VP 地址”中。

4.5.3.6 功能码“0x0F”写多个线圈状态实现

地址映射：

映射节点编辑	映射关系说明
	将 1 个单位的 16 位数字变量“0x080002”对应 16 个从机 ID 为 1 的线圈类型寄存器，起始地址“0x0001”。 当屏的变量“0x080002”被修改时，模块会发送报文修改相应地址的线圈类型寄存器，相关信息： 1、从机地址 ：0x01 2、寄存器类型 ：coil_0x 3、起始寄存器地址：0x0001 4、寄存器数量 ：16 (0x0010)

方式一、编辑 Modbus 逻辑节点实现

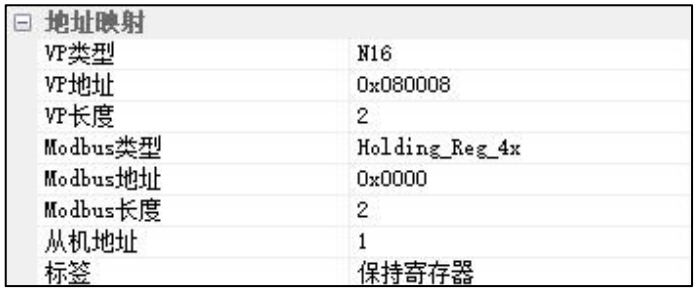
逻辑节点编辑	报文数据解析
	根据图中所标记的参数“执行操作”、“操作变量”、“操作值”以及变量“0x080002”的映射关系可以自动获取以下信息： 1、从机地址 ：0x01 2、功能码 ：0x0F 3、起始寄存器地址：0x0001 4、寄存器数量 ：0x0010 5、数据个数 ：0x02 6、写入的数据 ：0xD204(低字节在前，十进制 1234) 当满足“执行条件”时，模块发送报文： 01 0F 00 01 00 10 02 D2 04 BE 92

方式二、设置触摸键、虚拟键或页面功能属性实现

触摸键属性设置	报文数据解析
	根据图中所标记的参数“运算操作”、“VP 地址”、“标题/值”以及变量“0x080002”的映射关系可以自动获取以下信息： 1、从机地址 ：0x01 2、功能码 ：0x0F 3、起始寄存器地址：0x0001 4、寄存器数量 ：0x0010 5、数据个数 ：0x02 6、写入数据 ：0x6809(低字节在前，十进制 2408) 当屏检测该触摸键被按下时，模块发送报文： 01 0F 00 01 00 10 02 68 09 0C 37

4.5.3.7 功能码“0x06”写单个保持寄存器实现

地址映射：

映射节点编辑	映射关系说明
	将 2 个连续的 16 位数字变量“0x080008/0x08000A”对应 2 个从机 ID 为 1 连续的保持寄存器“0x0000/0x0001”。 当屏的变量“0x080008/0x08000A”被修改时，模块会发送报文修改相应地址的保持寄存器，相关信息： 1、从机地址 ：0x01 2、寄存器类型：Holding_Reg_4x 3、起始寄存器地址：0x0000 4、寄存器数量：2 (0x0002)

方式一、编辑 Modbus 逻辑节点实现

逻辑节点编辑	报文数据解析
	根据图中所标记的参数“执行操作”、“操作变量”、“操作值”以及变量“0x080008”的映射关系可以自动获取以下信息： 1、从机地址 ：0x01 2、功能码 ：0x06 3、寄存器地址：0x0000 4、写入数据 ：0x09B1(操作值为 2481) 当满足“执行条件”时，模块发送报文： 01 06 00 00 09 B1 4F EE

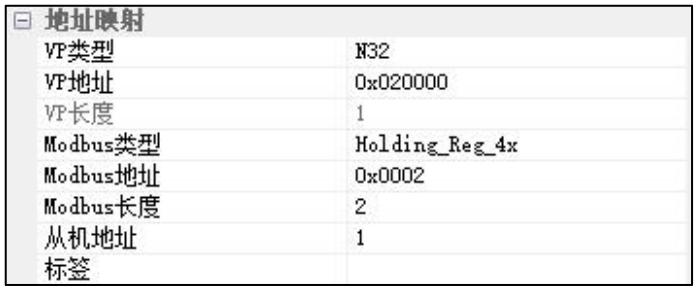
方式二、设置触摸键、虚拟键或页面功能属性实现

触摸键属性设置	报文数据解析
	根据图中所标记的参数“运算操作”、“VP 地址”、“标题/值”以及变量“0x08000A”的映射关系可以自动获取以下信息： 1、从机地址 ：0x01 2、功能码 ：0x06 3、寄存器地址：0x0001 4、写入数据 ：0x11D8(标题/值为 4568) 当屏检测该触摸键被按下时，模块发送报文： 01 06 00 01 11 D8 D4 00

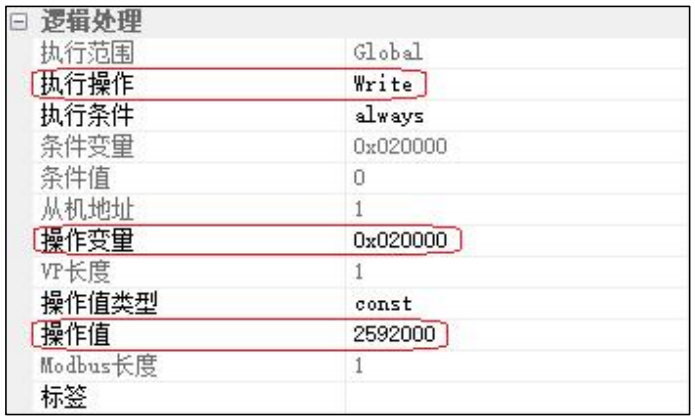
注：触摸键运算操作“VP:=value”作用将“标题/值”的数据写入“VP 地址”中。

4.5.3.8 功能码“0x10”写多个保持寄存器实现

地址映射：

映射节点编辑	映射关系说明
	将 1 个单位的 32 位数字变量“0x020000”对应 2 个从机 ID 为 1 连续的保持寄存器“0x0002/0x0003”。 当屏的变量“0x020000”被修改时，模块会发送报文修改相应地址的保持寄存器，相关信息： 1、从机地址 ：0x01 2、寄存器类型：Holding_Reg_4x 3、起始寄存器地址：0x0002 4、寄存器数量：2 (0x0002)

方式一、编辑 Modbus 逻辑节点实现

逻辑节点编辑	报文数据解析
	根据图中所标记的参数“执行操作”、“操作变量”、“操作值”以及变量“0x020000”的映射关系可以自动获取以下信息： 1、从机地址 ：0x01 2、功能码 ：0x10 3、起始寄存器地址：0x0002 4、寄存器数量 ：0x0002 5、数据个数 ：0x04 6、写入数据：0x00278D00(操作值为 2592000) 当满足“执行条件”时，模块发送报文： 01 10 00 02 00 02 04 00 27 8D 00 A7 2D

方式二、设置触摸键、虚拟键或页面功能属性实现

触摸键属性设置	报文数据解析
	根据图中所标记的参数“运算操作”、“VP 地址”、“标题/值”以及变量“0x020000”的映射关系可以自动获取以下信息： 1、从机地址 ：0x01 2、功能码 ：0x10 3、起始寄存器地址：0x0002 4、寄存器数量 ：0x0002 5、数据个数 ：0x04 6、写入数据 ：0x00C8EEBC(标题/值为 13168316) 当屏检测该触摸键被按下时，模块发送报文： 01 10 00 02 00 02 04 00 C8 EE BC BF 99

5 串口通信

智能模块通信指令用于实时传输数据和访问及控制。
主机通过模块提供的串口可实时获取模块键盘录入的数据或实时发送数据让模块显示。
智能模块的串行 UART 接口基于 RS232-C 标准，默认配置为 8N1 模式 115200bps。

5.1 指令帧格式

目前支持 3 种指令帧格式，在 SGTool 中"工程设置"窗口中可选中使用其中一种。

1. 指令帧格式<一>（普通协议指令）

序	包格式	字节说明	字节大小
1	0xAA	帧头	1byte
2	Cmd-code	命令码	1byte
3	Par-data	参数/数据	(*1)
:	:	-	-
:	:	-	-
:	:	-	-
N-3 th	0xCC	帧尾	4byte
N-2 th	0x33		
N-1 th	0xC3		
N th	0x3C		

注：所有数据都是高字节在前、低字节在后.
例：数据 0x1234,发送顺序应为 0x12 在前, 0x34 在后.

2. 指令帧格式<二>（带长度协议指令）

序	包格式	字节说明	字节大小
1	0xAA	帧头	1byte
2	Len	命令长度	2byte
3	Cmd-code	命令码	1byte
4	Par-data	参数/数据	(*1)
:	:	-	-
:	:	-	-
:	:	-	-
N-3 th	0xCC	帧尾	4byte
N-2 th	0x33		
N-1 th	0xC3		
N th	0x3C		

注：所有数据都是高字节在前、低字节在后.
例：数据 0x1234,发送顺序应为 0x12 在前, 0x34 在后.

3. 指令帧格式<三>（带长度和 CRC 协议指令）

序	包格式	字节说明	字节大小
1	0xAA	帧头	1byte
2	Len	命令长度	2byte
3	Cmd-code	命令码	1byte
4	Par-data	参数/数据	(*1)
:	:	-	-
:	:	-	-
:	:	-	-
N-3 th	0xCC	帧尾	2byte
N-2 th	0x33		
N-1 th	CRC16	CRC 校验	2byte
N th			

注：所有数据都是高字节在前、低字节在后.
例：数据 0x1234,发送顺序应为 0x12 在前, 0x34 在后.

确认指令格式

数据(Hex)	字符 ASCII	描述
3A 3E	":>"	指令执行成功时,屏返回 3A 3E 命令
21 3E	"!>"	指令执行失败/指令码/数据有误时,屏返回 21 3E 命令

注:

1. 数据确认包由模块发出
2. 在 SGTools"工程设置"中可通过设置 "使能 ACK" 选择是否开启确认响应. (勾选:启用指令确认)。

指令帧中数据格式定义:

16bit 颜色值定义

16bit 颜色															
R4	R3	R2	R1	R0	G5	G4	G3	G2	G1	G0	B4	B3	B2	B1	B0
高字节 (MSB)								低字节 (LSB)							
D7	D6	D5	D4	D3	D2	D1	D0	D7	D6	D5	D4	D3	D2	D1	D0

VP 变量数据定义

64 位数字变量							
D63…D56	D55…D48	D47…D40	D39..D32	D31…D24	D23…D16	D15…D8	D7…D0
Byte7(MSB)							Byte0(LSB)
D7…D0	D7…D0	D7…D0	D7…D0	D7…D0	D7…D0	D7…D0	D7…D0

32 位数字变量			
D31…D24	D23…D16	D15…D8	D7…D0
Byte3 (MSB)			Byte0 (LSB)
D7…D0	D7…D0	D7…D0	D7…D0

16 位数字变量	
D15…D8	D7…D0
高字节 (MSB)	低字节(LSB)
D7…D0	D7…D0

5.2 指令集

功能	指令码	指令名称	说明
参数设定（25 条）	0x30	握手	用于确认屏是否连接正常
	0x31	读版本号	读版本号
	0x32	读页面号	读取当前正在显示的页面 ID
	0x72/0x73	触摸坐标返回	自动返回触摸位置坐标
	0x77	触摸数据返回	自动返回数据(键盘录入数据或触摸键运算操作数据)
	0x78/0x79	触摸 ID 返回	自动返回页面 ID 和触摸键 ID
	0xE0	设置工作模式	设定模块的通信波特率、触摸返回码、文本显示模式
	0xE1	切换工程	切换到指定工程并显示
	0xE4	设置触摸屏校准	设定触摸屏进入校准模式
	0x5E	设置屏保	设置进入屏保时间和屏保状态下的背光亮度
	0x5F	设置背光	设置背光亮度
	0x79	设置蜂鸣器时长	设置蜂鸣器 1 次的鸣叫时间
	0x7A	蜂鸣器控制	控制蜂鸣器打开关闭、声音频率、时间长短
	0x9C	设置时钟	设置模块日期时间
	0x9B	读取时钟	读取模块日期时间
	0x90	写 Flash	写入数据到 flash (Flash 中数据断电可保存)
	0x91	读 Flash	读取 flash 数据
	0x93	读 USR.BIN 数据	读取 USR.BIN 中用户文件数据
	0xE2	格式化内部存储器	格式化内部存储器(工程包,Flash 数据,加密设定都会被清空)
	0xE3	解密内部存储器	用于内部存储器盘加密的情况下，对内部存储器进行一次解锁
	0xEE	复位	设置模块复位（重新启动）
	0x96	YModem 传输模式	设置模块进入 YModem 传输模式
	0x97	CRC 校验	计算指定文件的 CRC 结果
显示控制（5 条）	0x70	页面显示	显示指定的页面(画面)
	0x7E	设置控件字体颜色	更改页面中字符串变量或数字变量的字体颜色
	0x7F	设置控件背景颜色	改变页面中字符串变量或数字变量的背景色
	0xE7	设置字库	设置字库国家码和外扩 Codepage
	0xE8	刷新控制	设定当前画面暂停刷新或开始刷新,暂停刷新后触摸键无响应、变量显示不更新数据。
变量读写（22 条）	0x82	写数字变量	可往 16/32/64 位数字变量地址中连续写入多个数据
	0x83	读数字变量	读取 16/32/64 位数字变量地址中的连续多个数据
	0x4D	曲线数据写	对曲线图数据写入，控制曲线图的显示,以修改的方式进行刷新
	0x4E	左推进写曲线数据	对曲线图数据写入，以推进的方式控制曲线图的显示有利于对数据变化随时间变化的曲线显示减小传输的数据量
	0x3B	8 位系统寄存器设定	用于控制计数寄存器的工作方式
	0x3C	读取系统寄存器数据	读取寄存器中数据
	0x4B	位图数据写	单色位图数据写入，控制位图显示
	0x4C	位图压缩数据写	单色位图压缩数据写入，可减小数据量,控制位图显示
	0x42	写字符串变量	对字符串变量进行读写，可用于控制字符串控件显示
	0x43	读字符串变量	
	0x46	填充字符串变量	
	0x3D	写 16 位数字变量	对 16 位数据变量进行读写，可用于控制:数值控件显示，图标索引显示，十进位图标显示进度条显示, VPK 操作等
	0x3E	读 16 位数字变量	
	0x3F	填充 16 位数字变量	
	0x44	写 32 位数字变量	对 32 位数据变量进行读写。可用于控制数值控件显示 控制十进制图标显示
	0x45	读 32 位数字变量	
	0x47	填充 32 位数字变量	
	0x48	写 64 位数字变量	对 64 位数据变量进行读写。可用于控制数值控件显示
	0x49	读 64 位数字变量	
	0x4A	填充 64 位数字变量	
	0x94(*1)	变量与 Flash 数据交换	对变量与 Flash 之间进行数据交换
	0x95(*1)	写 usr.bin 文件到 VP	从 usr.bin 文件中读取数据到 VP 变量中

Note.

*1. 0x94，0x95 指令仅支持部分型号，详见产品用户手册.

5.3 指令描述

5.3.1 参数设定

	帧头	命令码	数据:字节数	帧尾
握手 (0x30)	AA	30	--	CC 33 C3 3C

例. [主机]: AA 30 CC 33 C3 3C
例. [模块]: AA 30 54 6F 70 77 61 79 20 48 4D 54 20 52 65 61 64 79 00 CC 33 C3 3C
(返回 ASCII 字符: "Topway HMT Ready")

读版本号 (0x31)	AA	31	--	CC 33 C3 3C

例. [主机]: AA 31 CC 33 C3 3C
例. [模块]: AA 31 31 2E 31 32 CC 33 C3 3C (返回 ASCII 字符:"1.12")

读页面号 (0x32)	AA	32	无或页面 ID:2	CC 33 C3 3C

例. [主机]: AA 32 CC 33 C3 3C
例. [模块]: AA 32 00 01 CC 33 C3 3C

触摸坐标返回 (0x72/0x73)	AA	72/73	X 坐标点:2 Y 坐标点:2	CC 33 C3 3C

例. [模块]: AA 72 00 32 00 64 CC 33 C3 3C (触摸键抬起后返回坐标(50,100))
例. [模块]: AA 73 00 32 00 64 CC 33 C3 3C (触摸键按下后返回坐标(50,100))

触摸数据返回 (0x77)	AA	77	VP 地址:4 数据:2/4/8/n	CC 33 C3 3C

例. [模块]: AA 77 00 02 00 00 00 00 00 32 CC 33 C3 3C

触摸 ID 返回 (0x78/79)	AA	78/79	Page_ID:2 TPK_ID:1	CC 33 C3 3C

例. [模块]: AA 78 00 02 01 CC 33 C3 3C (触摸键抬起后返回页面 ID 和触摸键 ID)
例. [模块]: AA 79 00 02 01 CC 33 C3 3C (触摸键按下后返回页面 ID 和触摸键 ID)

设置工作模式 (0xE0)	AA	E0 55 AA 5A A5	波特率:1 参数 1:1 参数 2:1	CC 33 C3 3C

例. [主机]:AA E0 55 AA 5A A5 07 83 00 CC 33 C3 3C
(设置波特率为 115200, 设置触摸键按下时响应) (下次上电时波特率恢复到工程参数设定值)

切换工程 (0xE1)	AA	E1	工程 ID:1	CC 33 C3 3C

例. [主机]: AA E1 01 CC 33 C3 3C
(切换 0x01 号工程, (THMT01 文件夹))

设置触摸屏校准 (0xE4)	AA	E4 55 AA 5A A5	--	CC 33 C3 3C

例. [主机]: AA E4 55 AA 5A A5 CC 33 C3 3C
(开启触摸屏校准)

设置屏保 (0x5E)	AA	5E	时间:2 亮度等级:1	CC 33 C3 3C

例. [主机]: AA 5E 00 0A 00 CC 33 C3 3C
(10 秒后无操作,设置背光亮度为 0)

设置背光 (0x5F)	AA	5F	Level:1	CC 33 C3 3C

例. [主机]: AA 5F 3F CC 33 C3 3C
(设置背光亮度位 63, 背光亮度分为 64 个等级, 0x00 最暗 ~ 0x3F 最亮)

设置蜂鸣器时长 (0x79)	AA	79	鸣叫时长(单位 10ms):1	CC 33 C3 3C

例. [主机] : AA 79 01 CC 33 C3 3C
(设置蜂鸣器鸣叫一次的时长为 10ms)

蜂鸣器控制 (0x7A)	AA	7A	Loops T1 T2 Freq1 Freq2	CC 33 C3 3C

例. [主机] : AA 7A 10 0A 08 05 32 CC 33 C3 3C
(设置蜂鸣器鸣叫声音频率和鸣叫时长)

写 Flash (0x90)	AA	90	VP 地址:4 数据长度:2 数据:n	CC 33 C3 3C

例. [主机]: AA 90 00 00 00 00 00 02 30 31 CC 33 C3 3C
(从 0x00000000 地址开始写入 2 个字节数据 0x30,0x31)

	帧头	命令码	数据:字节数	帧尾
读 Flash (0x91)	AA	91	ADDR:4 Len:2 or Data:n	CC 33 C3 3C

例. [主机]: AA 91 00 00 00 00 00 02 CC 33 C3 3C
(从 0x0000000 地址开始读 2 个字节数据)
[模块]: AA 91 30 31 CC 33 C3 3C
(模块返回 2 个字节数据 0x30, 0x31)

设置时钟 (0x9C)	AA	9C	日期时间格式:6(年月日时分秒)	CC 33 C3 3C

例. [主机]: AA 9C 0E 07 0A 17 3B 30 CC 33 C3 3C
(设置日期时间为:2014 - 07- 10 23:59:48)

读取时钟 (0x9B)	AA	9B	日期时间格式:6(年月日时分秒)	CC 33 C3 3C

例. [主机]: AA 9B CC 33 C3 3C (主机读取日期时间)
例. [模块]: AA 9B 0E 07 0B 00 01 12 CC 33 C3 3C
(返回日期时间: 2014-07-11 00:01:18)

读 USR.BIN 数据 (0x93)	AA	93	文件地址:4 读取长度:2 或 数据:n	CC 33 C3 3C

例. [主机]: AA 93 00 00 00 00 00 02 CC 33 C3 3C
(从 USR.BIN 文件地址 0x0000 0000 开始读取 2 个字节数据)
例. [模块]: AA 93 30 31 CC 33 C3 3C
(模块返回 2 个字节数据 0x30 0x31)

格式化内部存储器 (0xE2)	AA	E2	55 AA 5A A5	CC 33 C3 3C

例. [主机]: AA E2 55 AA 5A A5 CC 33 C3 3C
(格式化整个内部存储器空间)

解密内部存储器 (0xE3)	AA	E3	密码: n	CC 33 C3 3C

例. [主机]: AA E3 30 31 32 33 34 35 36 00 CC 33 C3 3C
(发送一个字符串密码解密内部存储器可访问, 密码在 SGTools"资源窗口-USB 访问锁中设定")

复位 (0xEE)	AA	EE	AA 55 A5 5A	CC 33 C3 3C

例. [主机]: AA EE AA 55 A5 5A CC 33 C3 3C
(程序恢复到初始状态，重新开始执行)

YModem 传输模式 (0x96)	AA	96	55 AA 5A A5	CC 33 C3 3C

例.[主机]: AA 96 55 AA 5A A5 CC 33 C3 3C
(主机发送指令后，模块会进入 YModem 文件传输模式画面, 用于通过串口升级工程界面)

CRC 校验 (0x97)	AA	97	文件路径+文件名：32 字节	CC 33 C3 3C

例.计 算模块中得 config.tml 文件 CRC 校验和 （config.tml 在模块的路劲是：/thmt/config.tml)
[主机->屏]: 帧头+指令码+文件名(thmt/config.tml)+帧尾
AA 97 2F 74 68 6D 74 2F 63 6F 6E 66 69 67 2E 74 6D 6C 0000 0000 0000 0000 0000 0000 0000 0000 CC 33 C3 3C
[屏->主机]: 帧头+指令码+文件名(thmt/config.tml)+checksum(0x34333231)+帧尾
AA 97 2F 74 68 6D 74 2F 63 6F 6E 66 69 67 2E 74 6D 6C 0000 0000 0000 0000 0000 0000 0000 0000 34 33 32 31 CC 33 C3 3C

5.3.2 显示控制

	帧头	命令码	数据:字节数	帧尾
显示页面 (0x70)	AA	70	页面 ID:2	CC 33 C3 3C

例. [主机]: AA 70 00 01 CC 33 C3 3C
(显示 0x0001 页面)

设置控件字体颜色 (0x7E)	AA	7E	控件类型:1 页面 ID:2; 控件 ID:1 颜色值:2	CC 33 C3 3C

例. [主机]: AA 7E 00 00 03 05 00 FF FF CC 33 C3 3C
(设置 0x0003 页面中 ID 号为 0x05 的字符串变量字体颜色为 0xFFFF 白色)

设置控件背景色 (0x7F)	AA	7F	控件类型:1 页面 ID:2 控件 ID:1 颜色值:2 透明模式:1	CC 33 C3 3C

例. [主机]: AA 7F 00 00 03 05 00 00 00 CC 33 C3 3C
(设置 0x0003 页面中 ID 号为 0x05 的字符串变量背景色为 0x0000 黑色);

设置字库 (0xE7)

AA	E7	国家码:1 代码页:1	CC 33 C3 3C

例. [主机]: AA E7 03 07 CC 33 C3 3C
(设置字库的国家码为 0x03:德语, 外扩 ASCII 代码页为 0x07:OEM-俄语)

刷新控制 (0xE8)

AA	E8	55 AA 5A A5 模式	CC 33 C3 3C

例. [主机]: AA E8 55 AA 5A A5 01 CC 33 C3 3C
(暂停刷新并停用触摸键; 模式 = 00 释放暂停)

5.3.3 变量读写

	帧头	命令码	数据:字节数	帧尾
写数字变量 (0x82)	AA	82	VP 地址:4 长度:1 数据:n	CC 33 C3 3C

例. [主机]: AA 82 00 08 00 00 03 00 32 00 33 00 34 CC 33 C3 3C
(从 0x00080000 地址开始写入 3 个 16 位的数据)
[主机]: AA 82 00 02 00 00 03 00 00 00 32 00 00 00 33 00 00 00 34 CC 33 C3 3C
(从 0x00020000 地址开始写入 3 个 32 位的数据)
[主机]: AA 82 00 08 00 00 02 0000000000000032 0000000000000034 CC 33 C3 3C
(从 0x00030000 地址开始写入 2 个 64 位的数据)

读数字变量 (0x83)	AA	83	VP 地址:4 或长度:1 数据:n	CC 33 C3 3C

例. [主机]: AA 83 00 08 00 00 03 CC 33 C3 3C
[模块]: AA 83 00 08 00 00 03 00 32 00 33 00 34 CC 33 C3 3C
[主机]: AA 83 00 02 00 00 03 CC 33 C3 3C
[模块]: AA 83 00 02 00 00 03 00 00 00 32 00 00 00 33 00 00 00 34 CC 33 C3 3C
[主机]: AA 83 00 03 00 00 02 CC 33 C3 3C
[模块]: AA 83 00 03 00 00 02 0000000000000032 0000000000000034 CC 33 C3 3C

写字符串变量 (0x42)	AA	42	VP 地址:4 数据:n	CC 33 C3 3C

例. [主机]: AA 42 00 00 00 80 54 4F 50 57 41 59 00 CC 33 C3 3C
(写字符串"TOPWAY" 到 0x0000 0080 地址中, 字符串必须要有 0x00 结尾)

读字符串变量 (0x43)	AA	43	VP 地址:4 或 数据:n	CC 33 C3 3C

例. [主机]: AA 43 00 00 00 80 CC 33 C3 3C
(读取 0x00000080 地址中的字符串数据)
[模块]: AA 43 54 4F 50 57 41 59 00 CC 33 C3 3C
(读回的数据"TOPWAY", 含有 0x00 结尾符)

填充字符串变量 (0x46)	AA	46	VP 地址:4 长度:2 数据:n	CC 33 C3 3C

例. [主机]: AA 46 00 00 00 80 00 03 54 4F 50 57 41 59 00 CC 33 C3 3C
(从 0x00000080 地址开始的连续 0x0003 个字符串 VP 地址中写入"TOPWAY"字符串)

写 16 位数字变量 (0x3D)	AA	3D	VP 地址:4 数据:2	CC 33 C3 3C

例. [主机]: AA 3D 00 08 00 00 00 32 CC 33 C3 3C
(写 16 位数据 0x0032 到 0x00080000 地址中)

读 16 位数字变量 (0x3E)	AA	3E	VP 地址:4 或 数据:2	CC 33 C3 3C

例. [主机]: AA 3E 00 08 00 00 CC 33 C3 3C
(读取 0x0008000 地址中的数据)
[模块]: AA 3E 00 32 CC 33 C3 3C
(读取到的数据为 0x0032)

填充 16 位数字变量 (0x3F)	AA	3F	VP 地址:4 长度:2 数据:2	CC 33 C3 3C

例. [主机]: AA 3F 00 08 00 00 00 03 00 32 CC 33 C3 3C
(从 0x00080000 地址开始的连续 0x0003 个地址中写入数据 0x0032)

写 32 位数字变量 (0x44)	AA	44	VP 地址:4 数据:4	CC 33 C3 3C

例. [主机]: AA 44 00 02 00 00 00 00 00 32 CC 33 C3 3C
(写数据 0x00000032 到地址 0x00020000 中)

读 32 位数字变量	AA	45	VP 地址:4 或 数据:4	CC 33 C3 3C

(0x45)

例. [主机]: AA 45 00 02 00 00 CC 33 C3 3C
(读取 0x00020000 地址中的数据)
[模块]: AA 45 00 00 00 32 CC 33 C3 3C
(读取到的数据为 0x00000032)

填充 32 位数字变量

AA	47	VP 地址:4 长度:2 数据:4	CC 33 C3 3C

(0x47)

例. [主机]: AA 47 00 02 00 00 00 03 00 00 00 32 CC 33 C3 3C
(从 0x00020000 地址开始的连续 0x0003 个地址中写入数据 0x00000032)

	帧头	命令码	数据:字节数	帧尾
写 64 位数字变量	AA	48	VP 地址:4 数据:8	CC 33 C3 3C

(0x48)

例. [主机]: AA 48 00 03 00 00 00 00 00 00 00 00 32 CC 33 C3 3C
(写数据 0x0000000000000032 到地址 0x00030000 中)

读 64 位数字变量	AA	49	VP 地址:4 数据:8	CC 33 C3 3C

(0x49)

例. [主机]: AA 49 00 03 00 00 CC 33 C3 3C
(读取 0x00030000 地址中的数据)
[模块]: AA 49 00 00 00 00 00 00 00 32 CC 33 C3 3C
(读取到的数据为 0x0000 0000 0000 0032)

填充 64 位数字变量	AA	4A	VP 地址:4 长度:2 数据:4	CC 33 C3 3C

(0x4A)

例. [主机]: AA 4A 00 03 00 00 00 03 00 00 00 00 00 00 00 32 CC 33 C3 3C
(从 0x00030000 地址开始的连续 0x0003 个地址中写入数据 0x0000000000000032)

写位图数据	AA	4B	VP 地址:4 长度:4	CC 33 C3 3C

(0x4B)

例. [主机]: AA 4B 00 04 00 00 00 00 02 00 CC 33 C3 3C + 512byte data...
(从 0x00040000 地址处写入 512 字节的位图数据)

写位图压缩数据	AA	4C	VP 地址:4 长度:4	CC 33 C3 3C

(0x4C)

例. [主机]: AA 4C 00 04 00 00 00 00 02 00 CC 33 C3 3C +512byte data...
(从 0x00040000 地址处写入 512 字节的压缩位图数据，模块自动解压)

写曲线数据	AA	4D	VP 地址:4 长度:2 数据:n	CC 33 C3 3C

(0x4D)

例. [主机]: AA 4D 00 06 00 00 00 02 00 32 00 33 CC 33 C3 3C
(从 0x00060000 地址写 2 个点数据 0x0032, 0x0033)

左推进写曲线数据	AA	4E	VP 地址:4 长度:2 数值:2	CC 33 C3 3C

(0x4E)

例. [主机]: AA 4E 00 06 00 00 00 32 00 33 CC 33 C3 3C
(从 0x00060000 地址开始的 0x0032 个数据向左移动 1 个数据点并把 0x0033 插入到 0x0032 位置)
显示效果类似电脑"任务管理器-性能"栏中显示的曲线效果

写系统寄存器	AA	3B	寄存器地址:4 数据:1	CC 33 C3 3C

(0x3B)

例. [主机]: AA 3B 00 FF FF 00 01 CC 33 C3 3C
(设置 0x00FFFFF00-Timer_Ctrl0 开始倒计时)

读系统寄存器	AA	3C	寄存器地址:4	CC 33 C3 3C

(0x3C)

例. [主机]: AA 3C 00 FF FF 00 CC 33 C3 3C
(读取系统寄存器 0x00FFFF00 地址中的数据)

VP 与 Flash 交换	AA	94	交换方向:1	Flash 偏移地址:4	VP 地址:4	变量长度:2	CC 33 C3 3C

(0x94)

例. [主机]: AA 94 00 00 00 00 00 00 08 00 00 02 00 CC 33 C3 3C （交换方向 0：vp to flash 1:flash to vp）
(将 VP 变量中起始地址为 0x00080000 的 1KB 数据写到 Flash 中，Flash 起始地址为 0x00000000)

写 usr.bin 文件到 VP	AA	95	01	usr.bin 偏移地址:4	VP 地址:4	变量长度:2	CC 33 C3 3C

(0x95)

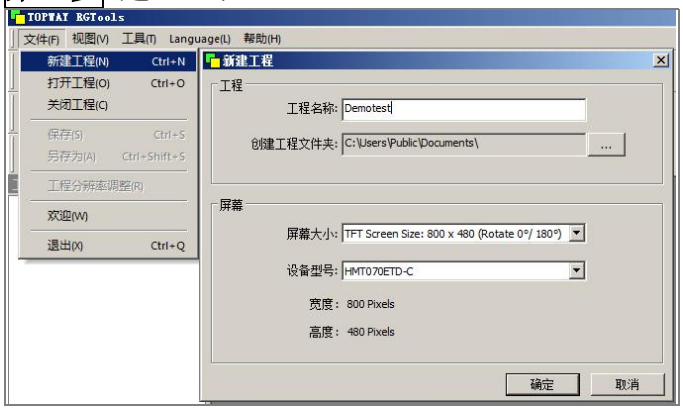
例. [主机]: AA 95 01 00 00 00 00 00 08 00 00 02 00 CC 33 C3 3C
(将 usr.bin 中起始地址为 0x00000000 的 1KB 数据写到 VP 变量中，VP 变量起始地址为 0x00080000)

6 应用案例

6.1 创建工程

本应用例子介绍了如何创建工程、创建页面、下载工程。

第一步 建立工程



①打开 SGTools，点击菜单栏建立新工程
"文件"->"新建工程"
②输入工程名字工程名
工程名称: Demotest
③输入工程保存位置
创建工程文件夹: C:\Users\Public\Document
④选择智能模块分辨率
屏幕大小: TFT Screen Size：800x480(Rotate 0°/180°)
注意: 选择的分辨率要和模块屏幕分辨率一致,否则显示乱.
Rotate Rotate 0°/180° ：表示横屏显示
Rotate Rotate 90°/270°：表示竖屏显示
⑤选择合适设备型号
参考所购模块标签。

第二步 建立页面和导入背景图

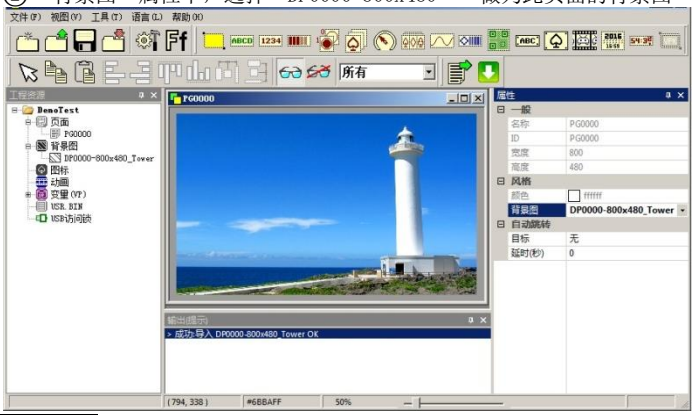
①新建页面
右击"页面"->"新建页面"

②导入背景图
右击"背景图"->"导入背景图"

③选择电脑中合适大小的图片
图片要求: 建议使用 24 位 BMP，分辨率大小与工程相同

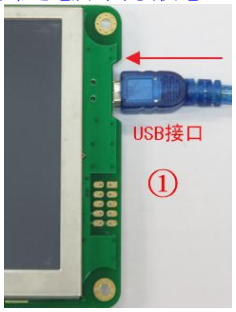
第三步 关联页面与背景图

①点击 PG0000 页面工作区域，右面属性窗口中会显示页面属性.
②"背景图"属性中，选择"DP0000-800x480…"做为此页面的背景图



第四步 编译与下载

①用 USB 线连接屏和电脑
屏幕连电脑时不要供电！



②工具栏->点击"下载"（或 F9 键）
③对话框中点击"开始下载"

第五步 上电显示

①屏 USB 断开与电脑连接
②给屏供电，可以看到显示效果
(接口定义及供电电压请参考用户手册)

显示效果

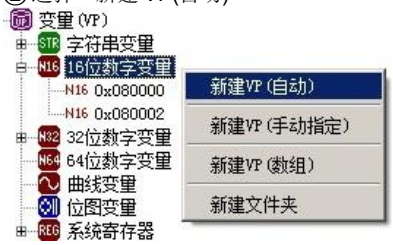
- Done -

6.2 数字控件应用

本应用例子主要介绍了如何新建 VP 变量、数字控件如何显示整数、浮点小数(Float)。

第一步 创建 VP 变量

①工程资源栏中右击“16 位数字变量”
②选择“新建 VP(自动)”



③右击新建的 VP，选择“重命名”
重命名是给 VP 变量起一个易识别的名称.

新建后:

重命名后：

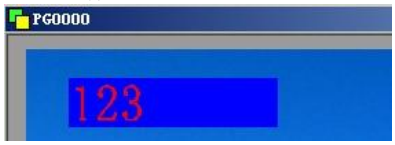
第二步 创建数字控件

①工具栏点击“数字控件”或按快捷键 Ctrl +I
②在 PG0000 页面中创建出“数字控件”

第三步 设置属性(整数显示效果)

①设置字体"48_ASCII_SONG(24x48)"
②设置字体颜色(红色=0xFF0000)
③设置背景色(蓝色=0x0000FF)
④设置 VP 地址 0x080004-数字显示
其他属性按默认值.

设置后控件显示效果：



第四步 编译与下载(略)

第五步 上电通过串口发送数据

①供电并连接串口
②通过串口往 VP 地址“0x080004”中写入数据“0x0064”显示数字 100，发送如下指令：
AA 3D 00080004 0064 CC 33 C3 3C

注：
1. 浮点数类型数据占 4 个字节，单片机给屏发送 float 类型数据时，数字控件属性必须是"浮点型"才可正确显示.
2. 若数据类型为整型类型(有符号/无符号整型)且设置了小数位≥1 时,数据会被格式化成小数格式显示.
   例: 整数位=自动,小数位=2.
   数字 1234 显示为 12.34
   数字 50 显示为 0.50

显示整数

①依据显示的数值的大小,创建合适的数字 VP
②设置控件属性
重点设置 VP 地址/数据类型/整数位/小数位
例: VP 地址 := 0x080004
数据类型 = 无符号整型
整数位 = 自动(根据实际位数显示)
小数位 = 0 (不显示小数位)
③串口发数据
AA 3D 00080004 EA60 CC 33 C3 3C

显示小数

①依据显示的数值的大小,创建合适的数字 VP
②设置控件属性
重点设置 VP 地址/数据类型/整数位/小数位
例: VP 地址 := 0x080004
数据类型 = 无符号整型
整数位 = 自动(根据实际位数显示)
小数位 = 2 (显示 2 位小数)
③串口发数据
AA 3D 00080004 EA60 CC 33 C3 3C

显示小数(浮点数)

①创建 32 位数字变量(float 类型数据占 4 字节)
②设置控件属性
重点设置 VP 地址/数据类型/整数位/小数位
例: VP 地址 := 0x020000
数据类型 = 无符号整型
整数位 = 自动(根据实际位数显示)
小数位 = 2 (显示 2 位小数,四舍五入)
③串口发数据
AA 44 00020000 4048F5C3 CC 33 C3 3C

- Done -

6.3 字符串控件应用

本应用例子介绍如何创建字符串 VP 变量、字符串如何显示英文字符。

第一步 创建 VP 变量

①工程资源栏中右击“字符串变量”
②选择“新建 VP(自动)”
③重命名 VP 为“文本显示”，如下图：

第二步 创建字符串控件

①工具栏中点击“字符串”或按快捷键 Ctrl +T
②在 PG0000 页面中创建字符串控件

第三步 设置字符串控件属性

①设置字体"32_ASCII_NewSong (16x32)"
②设置字体颜色(红色=0xFF0000)
③设置背景色(蓝色=0xFFFF00)
④设置 VP 地址 0x000080-文本显示
其他属性按默认值.

设置后控件显示效果:

第四步 编译与下载(略)

第五步 上电并通过串口发送数据

①供电并连接串口
②通过串口往 VP 地址：0x000080 中写入数据“0x54 0x4F 0x50 0x57 0x41 0x59 0x00”显示“TOPWAY”内容，发送如下指令：

AA 42 00000080 54 4F 50 57 41 59 00 CC 33 C3 3C

注:
- 最后一个字符必须为'\0'结尾符(即 0x00)
- "TOPWAY"对应的 ASCII 码为:54 4F 50 57 41 59

其他 显示中文字符

①“字体”需要设置为中文字体，如：24_GB2312_SONG(24x24)
②重新下载工程
③通过串口往 VP 地址：0x000080 中写入数据“0xCD 0xD8 0xC6 0xD5 0xCE 0xA2 0x00”显示“拓普微”内容， 发送如下指令：

AA 42 00000080 CD D8 C6 D5 CE A2 00 CC 33 C3 3C

注:
- 显示中文,字体属性必须要设置中文字库. 显示其他字体类同.
- 最后一个字符必须为'\0'结尾符(即 0x00)
- 中文字符要根据选择的字库是 GB2312 还是 GBK 等,发送汉字对应的编码.
- 本例中, "拓普微"对应的 GB2312 编码为: CDD8 C6D5 CEA2

- Done -

6.4 变量图标控件应用

本应用例子介绍了变量图标控件使用。

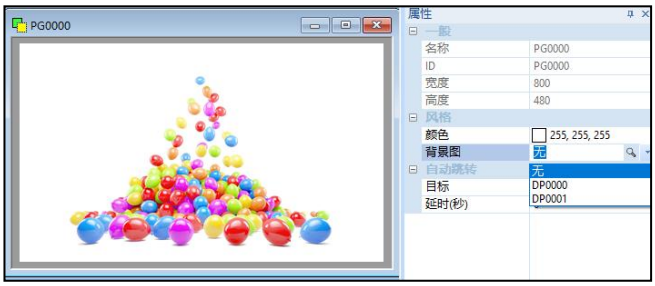
第一步 建立工程(略)

第二步 建立页面 ，导入背景图(略)

第三步 关联页面与背景图

① 点击工作区域，右面显示页面属性

② 属性中“背景图”选择导入的页面背景

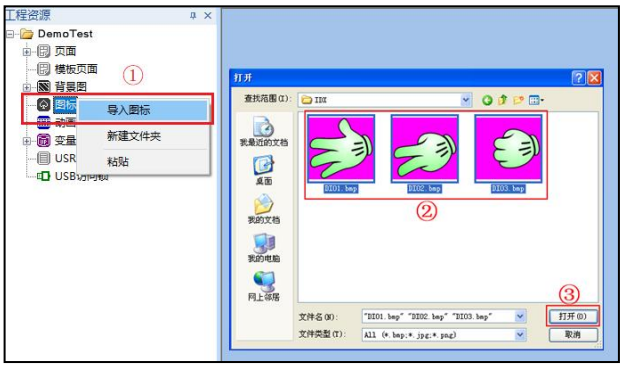


第四步 导入变量图标

① 在资源窗口中，右击“图标” 选择“导入图标”

② 选中需要导入图片

③ 点击“打开”按钮，完成



第五步 建立 VP_N16 地址与变量图标控件

① 在资源窗口中，右击“16 位数字变量”选择“新建 VP”

② 菜单工具栏点击变量图标控件

③ 在页面工作区拖出矩形区，新建一个变量图标控件

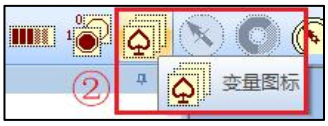
④ 属性中“透明”选择：是

⑤ 属性中“透明颜色”设置：0xFF00FF (255，0，255)

⑥ 属性中“图标”选择刚刚导入的图标：DI0000

⑦ 属性中“VP 地址”： 0x080000

⑧ 属性中“最大值”设置：132；“最小值”设置：130



第六步 编译与下载(略)

第七步 上电显示

① 断开 USB 与模块连接

② 连接串口并上电

③ 发送第 1 条命令

指令 1:AA 3D 00 08 00 00 00 82 CC 33 C3 3C

注：对 0x00080000 地址写入 0x0082 数据

(N16 值为最小值，显示图标第一个图标)

④ 发送第 2 条命令

指令 2: AA 3D 00 08 00 00 00 84 CC 33 C3 3C

注：对 0x00080000 地址写入 0x0084 数据

(N16 值为最大值，显示第三个图标)

⑤ 发送第 3 条命令

指令 3: AA 3D 00 08 00 00 00 85 CC 33 C3 3C

注：对 0x00080000 地址写入 0x0085 数据

(N16 值为超最大/最小值，不显示图标)

- Done -

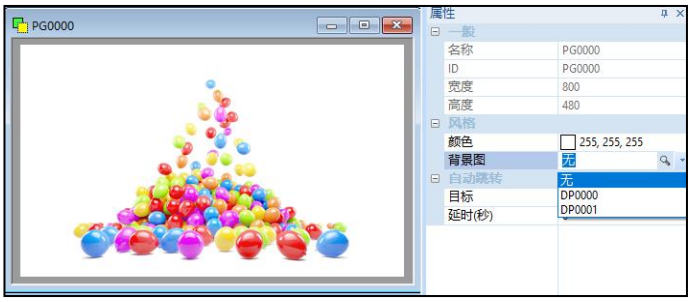
6.5 十进位变量图标控件应用

第一步 建立工程(略)

第二步 建立页面 ，导入背景图(略)

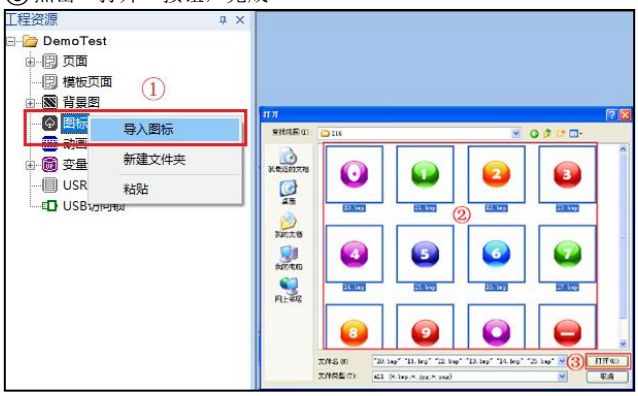
第三步 关联页面与背景图

①点击工作区域，右面显示页面属性
②属性中“背景图”选择页面背景图



第四步 导入图标

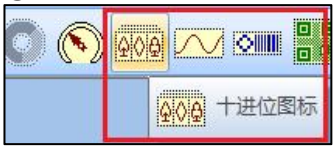
①在资源窗口中，右击“图标”选择导入图标
②选中需要导入图片
③点击“打开”按钮，完成

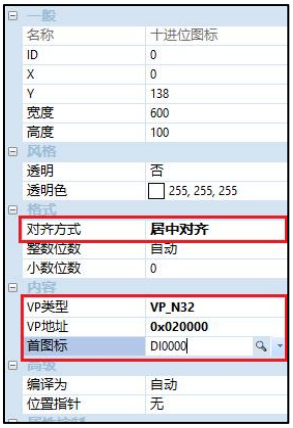


注： ICON

第五步 建立十进位控件与关联 VP

①菜单工具栏点击 十进位图标控件
②在页面工作区拖出矩形区，新建一个十进位图标控件
③属性中“对齐方式”选择：居中对齐
④属性中“首图标”选择：DI0000
⑤属性中“VP 类型”选择：VP_N32
⑥属性中“VP 地址”选择：0x020000

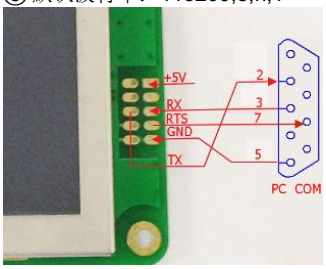




第六步 编译与下载(略)

第七步 上电显示

①断开 USB 与模块连接
②向模块提供 5V 电源
③上电显示
④连接串口线
⑤默认波特率：115200,8,n,1



⑥发送指令：
AA 44 00 02 00 00 00 1C B6 59 CC 33 C3 3C
VP 地址：0x00020000
VP 数据：0x 001CB659
注：对 0x00020000 地址写入 0x001CB659 数据

⑦发送指令：
AA 44 00 02 00 00 00 66 14 49 CC 33 C3 3C
VP 地址：0x00020000
VP 数据：0x 00661449
注：对 0x00020000 地址写入 0x00661449 数据

- Done -

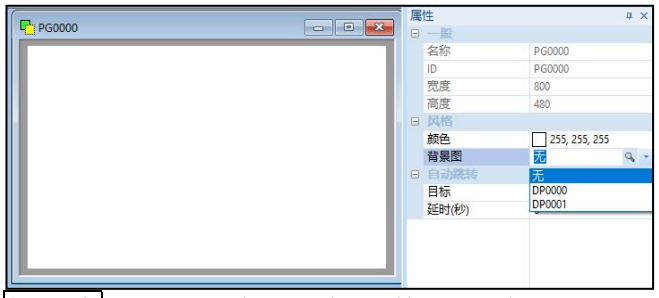
6.6 位变量图标控件应用

第一步 建立工程(略)

第二步 建立页面 ，导入背景图(略)

第三步 关联页面背景图

① 点击工作区域，右边显示页面属性
② 属性中“背景图” 选择页面背景



第四步 导入 2 张图标

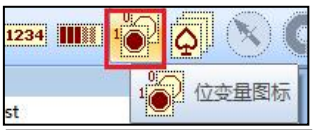
① 于资源窗口中，右击“图标” 选择“导入图标”
② 选中需要导入图标
③ 点击“打开”按钮，(导入的图标如下说明)

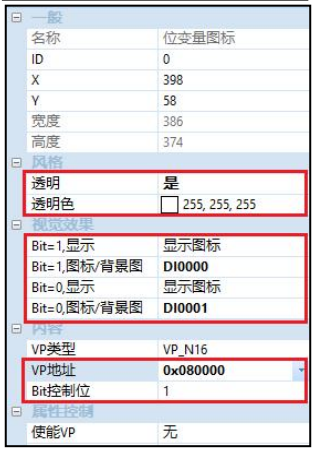
Bit=0，关闭状态：

Bit=1，开启状态：

第五步 建立 1 个位图标控件与关联 VP

① 点击菜单工具栏位变量控件图标
② 在页面工作区新建 1 个位变量图标控件
③ 属性中“透明”选择：是
④ 属性中“透明颜色”选择：0xFFFFFF(255，255，255)
⑤ 属性中“Bit=1，图标”和“Bit=0，图标”分别选择：DI0000 和 DI0001
⑥ 属性中“VP 地址”选择：0x080000
⑦ 属性中“Bit 控制位”设置：1 (使用 D1 作为控制位)





第六步 建立 2 个触摸键控件与设置键属性

① 点击菜单工具栏触摸键控件图标
② 分别在页面 OFF 和 ON 区域处新建 2 个触摸键控件
③ 属性中“运算操作”选择：bit1(VP):=LSB(Value)
④ 属性中“VP 地址”选择：0x080000
⑤ 属性中“标题/值”分别设置：0 和 1

第八步 编译与下载（略）

第九步 上电显示

① 断开 USB 与模块连接
② 向模块提供 5V 电源
③ 上电显示

④ 点击 ON 处触摸键，图标变化

⑤ 点击 OFF 处触摸键，图标变化

- Done -

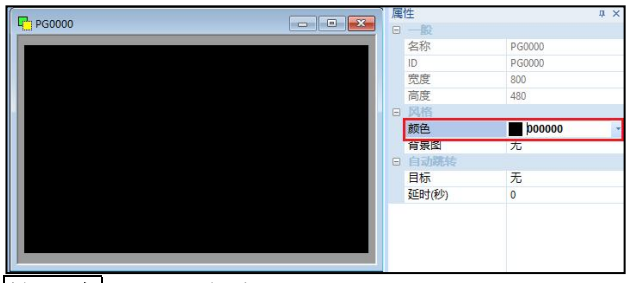
6.7 计时器控件应用

第一步 建立工程(略)

第二步 建立 1 个页面(略)

第三步 设置页面背景图

① 点击页面工作区域，右面显示页面属性
② 属性中“颜色”设置：000000



第四步 导入 2 张变量图标

① 于资源窗口中，右击“图标”选择导入图标
② 选中需要导入图标
③ 点击“打开”按钮，(导入的图标如下说明)

VP=0，关闭状态：

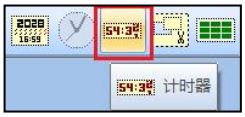
VP=1，开启状态：

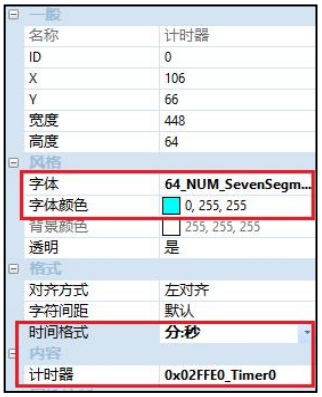
第五步 导入字库

① 点选菜单栏“字体设置”
② 点选 44 号右击选“选择”
③ 选择“64_NUM_SevenSegment(40x64)”字体.
④ 点击“选择”选择
⑤ 点击“关闭”完成

第六步 建立计时器控件

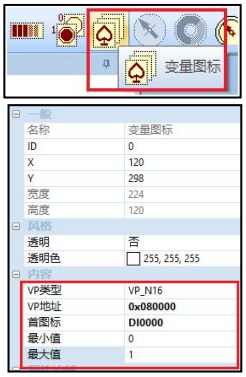
① 点选菜单工具栏“计时器”控件
② 在 PG0000 页面工作区创建 1 个计时器控件
③ 属性中“字体颜色”选择：0x00FFFF(0，255，255)
④ 属性中“字体”选择：64_NUM_SevenSegment(40x64)
⑤ 属性中“时间格式”选择：分:秒
⑥ 属性中“计计时器”选择：0x02FFE0-Timer0





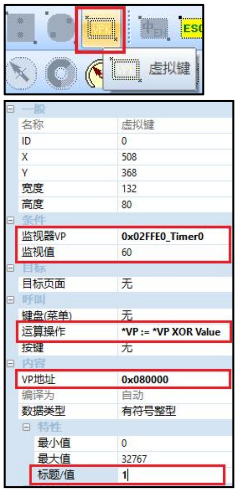
第七步 建立变量图标与属性设置

① 点选菜单工具栏“变量图标”
② 在 PG0000 页面工作区创建 1 个变量图标控件
③ 属性中“首图标”关联刚导入的图标：DI0000
④ 属性中“VP 地址” 选择：0x80000
⑤ 属性中“最大值”设置：1



第八步 建立虚拟键与属性设置

① 点选菜单工具栏“虚拟键”
② 在 PG0000 页面工作区新建 1 个虚拟键控件
③ 属性中“监视器 VP”选择：0x02FFE0-Timer0
④ 属性中“监视值”设置：60
⑤ 属性中“运算操作”选择：VP:=*VP XOR Value
⑥ 属性中“VP 地址”选择：0x080000
⑦ 属性中“标题/值”设置：1



第九步 编译与下载（略）

第十步 上电显示

① 断开 USB 与模块连接
② 向模块提供 5V 电源，上电显示
③ 连接串口线

④ 默认波特率：115200,8,n,1
⑤ 指令：AA 3B 00 FF FF 00 03 CC 33 C3 3C
注：对 0xFFFF00 地址写入 0x03 数据，向上计数

⑥ 计时器计数到 60，虚拟机按被触发，把被监控的计时器 VP 清 0；同时执行“运算操作”功能，对应变量图标改变

- Done -

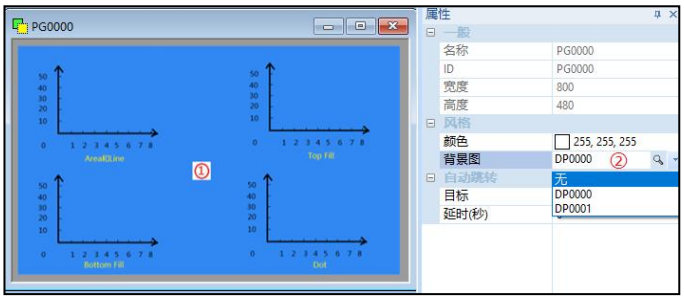
6.8 曲线控件应用

第一步 建立工程(略)

第二步 建立页面和导入背景图(略)

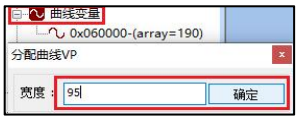
第三步 关联页面与背景图

① 点击工作区域，右面显示页面属性
② 属性中选择“背景图”页面背景图 DP0000



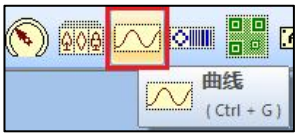
第四步 建立曲线 VP 变量

① 在资源窗口中，右击“曲线变量”选择“新建曲线 VP”
② 建立两个曲线 VP 变量，“宽度”分别为 190 和 95
③ 点击“确定”按钮完成

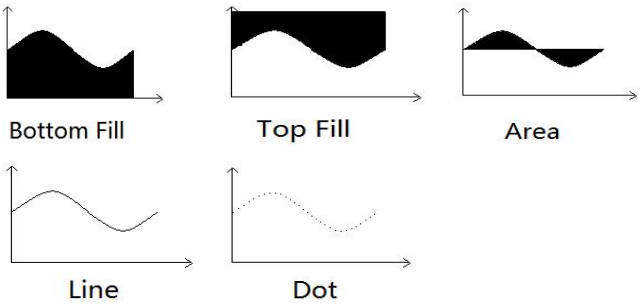


第五步 建立 4 个曲线控件与关联 VP

① 在工具栏中点击“曲线”图标，在页面上新建 4 个曲线控件。



注： 曲线的 5 种类型如右图说明：
页面上 4 个曲线控件的属性说明如下信息。



① 设置 ID0 控件属性
曲线类型 = 区域填充
最小值 = -400
最大值 = 400
点宽 = 1
点高 = 3
前景色 = 0xFF00FF
曲线 VP = 0x060000

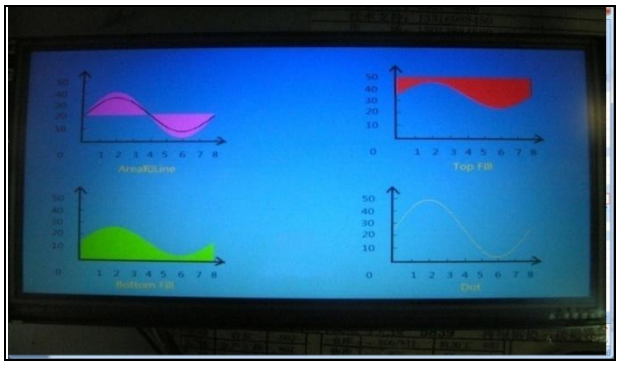
② 设置 ID1 控件属性
曲线类型 = 顶部填充
最小值 = -400
最大值 = 400
点宽 = 1
点高 = 3
前景色 = 0xFF0000
曲线 VP = 0x060000

③ 设置 ID2 控件属性
曲线类型 = 底部填充
最小值 = -400
最大值 = 400
点宽 = 1
点高 = 3
前景色 = 0x00FF00
曲线 VP = 0x060000

④ 设置 ID3 控件属性
曲线类型 = 点
最小值 = -50
最大值 = 50
点宽 = 2
点高 = 1
前景色 = 0xFFFF00
曲线 VP = 0x0600BE

第六步 编译与下载(略)

第七步 上电显示&发送数据



① 连接串口线并供电
② 发送第一条曲线数据，共 190 个点
AA 4E 00 06 00 00 00 BE 00 01 CC 33 C3 3C
AA 4E 00 06 00 00 00 BE 00 0A CC 33 C3 3C
AA 4E 00 06 00 00 00 BE 00 14 CC 33 C3 3C
AA 4E 00 06 00 00 00 BE 00 1E CC 33 C3 3C
：
③ 发送第二条曲线数据，共 95 个点
AA 4E 00 06 00 BE 00 5F 00 01 CC 33 C3 3C
AA 4E 00 06 00 BE 00 5F 00 04 CC 33 C3 3C
AA 4E 00 06 00 BE 00 5F 00 07 CC 33 C3 3C
AA 4E 00 06 00 BE 00 5F 00 0A CC 33 C3 3C
：

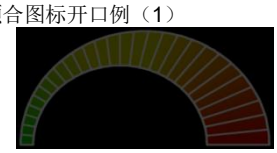
- Done -

6.9 表盘控件应用

第一步 建立工程(略)

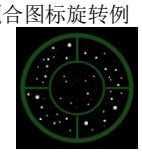
第二步 导入背景图、关联页面背景、导入图标文件（略）

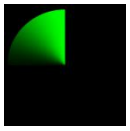
图标说明：预合图标开口例（1）

背景 DI0000： 

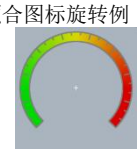
前景 DI0001： 

图标说明：预合图标旋转例（1）

背景 DI0004： 

前景 DI0005： 

图标说明：预合图标旋转例（2）

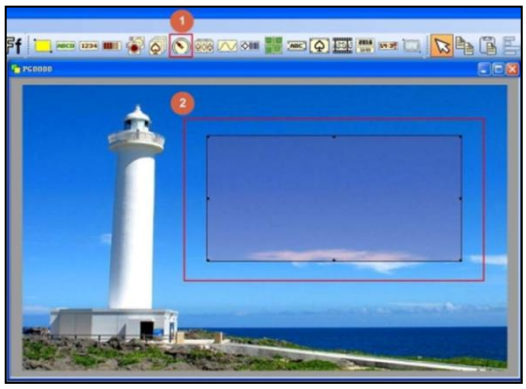
背景 DI0002： 

前景 DI0003： 

第三步 放置表盘控件

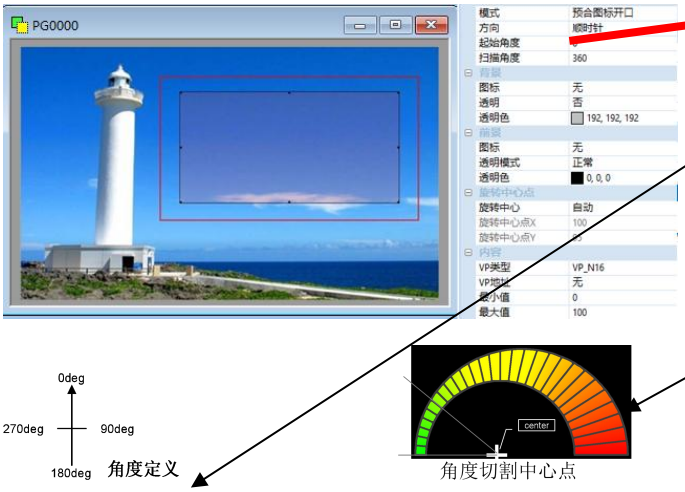
① 菜单工具栏点击“表盘”控件，然后在工作区域新建一个表盘
② 鼠标左键拖放控件，并调整合适大小

注：表盘属性设置及应用效果参考第四、五、六步的三个案例说明。



第四步 预合图标开口例（1）

① 设置表盘属性(详细如右表)



No.	属性	属性值描述
1	模式	选择“预合图标开口”
2	方向	旋转方向，选择“顺时针”方向
3	起始角度	设置 270°
4	扫描角度	设置 180°
5	图标	设置背景图标，这里选择 DI0000
6	透明	设置图标透明，这里选择“自动”模式
7	透明色	“自动”模式下会自动选择图标的背景色，不可手动设置，此例为黑色
8	图标	设置前景图标，这里选择 DI0001
9	透明模式	这里选择“正常”
10	透明色	选择图标的背景色，此例为黑色
11	旋转中心	设置中心点模式，选择“高级”
12	旋转中心点 X	设置合适中心点 X 坐标，此例为 156
13	旋转中心点 Y	设置合适中心点 Y 坐标，此例为 194
14	VP 类型	选择 VP_N16(16 位数字变量)
15	VP 地址	关联 VP，此例为 0x080000
16	最小值	设定(第一张图片)，此例为 0
17	最大值	设定(最后一张图片)，此例为 28

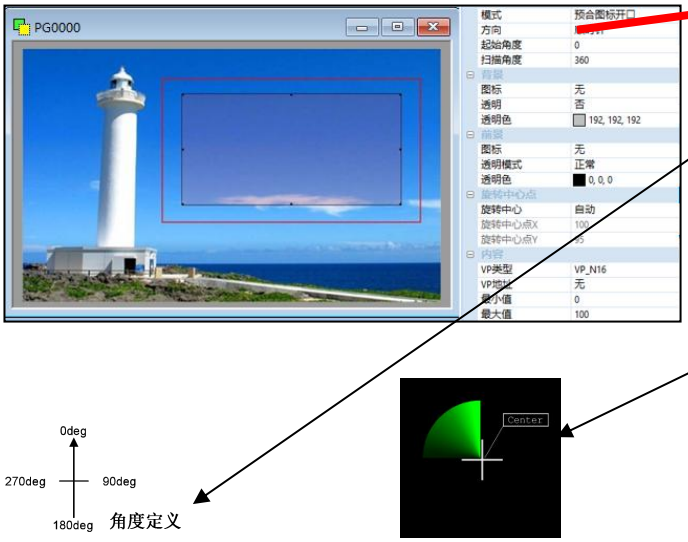
②编译下载到智能模块，上电并发送指令

AA 3D 00 08 00 00 0000 CC 33 C3 3C 显示第 1 张图片
AA 3D 00 08 00 00 0001 CC 33 C3 3C 显示第 2 张图片
AA 3D 00 08 00 00 0002 CC 33 C3 3C 显示第 3 张图片
：
AA 3D 00 08 00 00 001A CC 33 C3 3C 显示第 27 张图片
AA 3D 00 08 00 00 001B CC 33 C3 3C 显示第 28 张图片
AA 3D 00 08 00 00 001C CC 33 C3 3C 显示第 29 张图片

模块实际显示图像(以第 1 和 23 张为例)

第五步 预合图标旋转例（1）

①配置表盘属性(详细如右表)



No	属性	属性值描述
1	模式	选择“预合图标旋转”
2	方向	旋转方向，选择“逆时针”方向
3	起始角度	设置 0°
4	扫描角度	设置 360°
5	图标	设置背景图标，这里选择 DI0004
6	透明	设置图标透明，这里选择“自动”模式
7	透明色	“自动”模式下会自动选择图标的背景色，不可手动设置，此例为黑色
8	图标	设置前景图标，这里选择 DI0005
9	透明模式	这里选择“变亮”
10	透明色	此例为黑色(“变亮”模式下不可设置)
11	旋转中心	设置中心点模式，选择“自动”
12	旋转中心点 X	设置合适中心点 X 坐标，此例为 193
13	旋转中心点 Y	设置合适中心点 Y 坐标，此例为 193
14	VP 类型	选择 VP_N16(16 位数字变量)
15	VP 地址	关联 VP，此例为 0x080000
16	最小值	设定(第一张图片)，此例为 0
17	最大值	设定(最后一张图片)，此例为 11

②编译下载到智能模块，上电并发送指令

发送指令到 0x080000

AA 3D 00 08 00 00 0000 CC 33 C3 3C 显示第 1 张图片

AA 3D 00 08 00 00 0001 CC 33 C3 3C 显示第 2 张图片

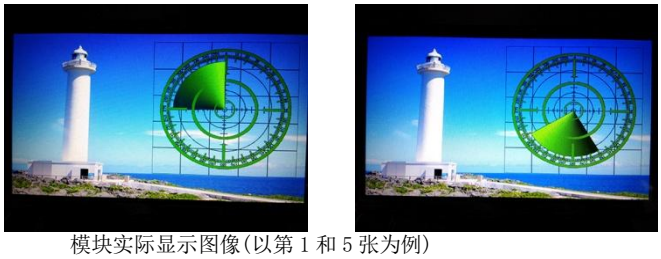
AA 3D 00 08 00 00 0002 CC 33 C3 3C 显示第 3 张图片

：

AA 3D 00 08 00 00 000B CC 33 C3 3C 显示第 11 张图片

AA 3D 00 08 00 00 000C CC 33 C3 3C 显示第 12 张图片

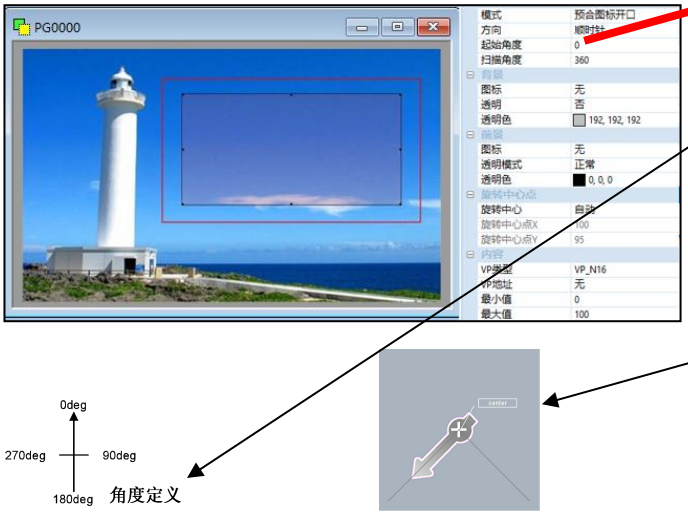
AA 3D 00 08 00 00 000D CC 33 C3 3C 显示第 13 张图片



模块实际显示图像(以第 1 和 5 张为例)

第六步预合图标旋转例（2）

①配置表盘属性(详细如右表)



No	属性	属性值描述
1	模式	选择“预合图标旋转”
2	方向	旋转方向，选择“顺时针”方向
3	起始角度	设置 0°
4	扫描角度	设置 270°
5	图标	设置背景图标，这里选择 DI0002
6	透明	设置图标透明，这里选择“自动”模式
7	透明色	“自动”模式下会自动选择图标的背景色，不可手动设置，此例为(170,180,190)
8	图标	设置前景图标，这里选择 DI0003
9	透明模式	这里选择“正常”
10	透明色	此例为(170,180,190)
11	旋转中心	设置中心点模式，选择“自动”
12	旋转中心点 X	设置合适中心点 X 坐标，此例为 154
13	旋转中心点 Y	设置合适中心点 Y 坐标，此例为 154
14	VP 类型	选择 VP_N16(16 位数字变量)
15	VP 地址	关联 VP，此例为 0x080000
16	最小值	设定(第一张图片)，此例为 0
17	最大值	设定(最后一张图片)，此例为 6

②编译下载到智能模块，上电并发送指令

发送指令到 0x080000

AA 3D 00 08 00 00 0000 CC 33 C3 3C 显示第 1 张图片

…

AA 3D 00 08 00 00 0006 CC 33 C3 3C 显示第 7 张图片

模块实际显示图像(以第 1 和 4 张为例)

- Done -

6.10 绘图板控件应用

第一步 建立工程

第二步 导入背景图、导入图标文件（略）

第三步 关联页面与背景图

① 点击工作区域,，右边显示页面属性

②属性中“背景图”选择页面背景图 DP0000

第四步 建立 DrawPad 控件，并关联 VP

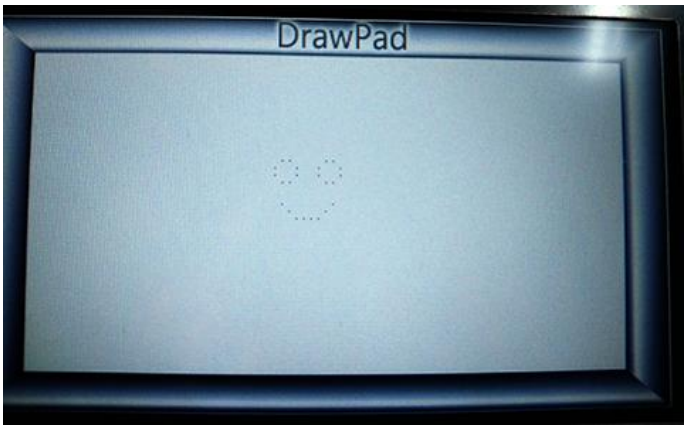
① 菜单工具栏点击“绘图板”控件，在工作区域新建控件，调整大小

② 属性中“VP 地址”选择 VP：0x080000

第五步 下载(略)

第六步 上电&发送绘图指令

①置点

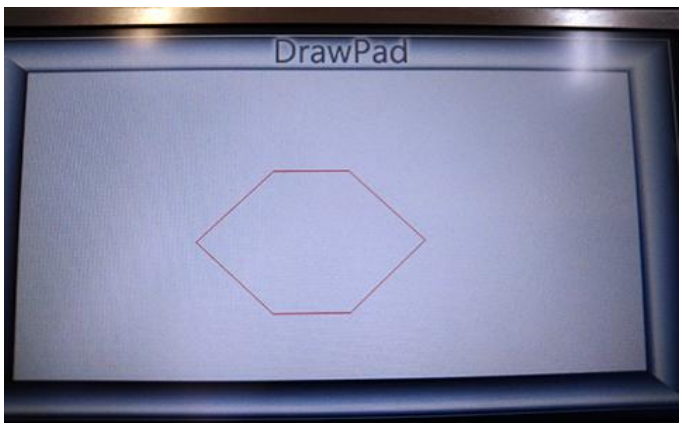


① 描述：在“绘图板”控件刷新范围内，指定位置绘制点
② CMD ： 连续写 16 位数据指令码，0x82
③ VP 地址：“绘图板”控件关联的 VP 地址，0x080000
④ Len：16 位数据的个数(阴影部分)
⑤ Type：置点指令，0x0001
⑥ Num ： 点的个数
⑦ X/Y ： 点的坐标
⑧ 颜色：点的颜色，16 位 RGB565 色值
⑨ 指令格式(使用连续写 16 位数据指令 0x82 发送)：

帧头	CMD	VP 地址			Len	Type	Num	X1	Y1	颜色	X2	Y2	颜色
AA	82	00080000			4A	0001	0018	00C8	0069	0000	00C8	006E	0000
X3	Y3	颜色	X4	Y4	颜色	X5	Y5	颜色	X6	Y6	颜色	X7	Y7
00CD	0064	0000	00CD	0073	0000	00CD	0082	0000	00D2	0064	0000	00D2	0073
颜色	X8	Y8	颜色	X9	Y9	颜色	X10	Y10	颜色	X11	Y11	颜色	X12
0000	00D2	0087	0000	00D7	0069	0000	00D7	006E	0000	00D7	008C	0000	00DC
Y12	颜色	X13	Y13	颜色	X14	Y14	颜色	X15	Y15	颜色	X16	Y16	颜色
008C	0000	00E1	008C	0000	00E6	008C	0000	00E6	0069	0000	00E6	006E	0000
X17	Y17	颜色	X18	Y18	颜色	X19	Y19	颜色	X20	Y20	颜色	X21	Y21
00EB	0064	0000	00EB	0073	0000	00EB	0087	0000	00F0	0064	0000	00F0	0073
颜色	X22	Y22	颜色	X23	Y23	颜色	X24	Y24	颜色	帧尾			
0000	00F0	0082	0000	00F5	0069	0000	00F5	006E	0000	CC33 C33C			

注：以上均为 16 进制数据

②端点连线

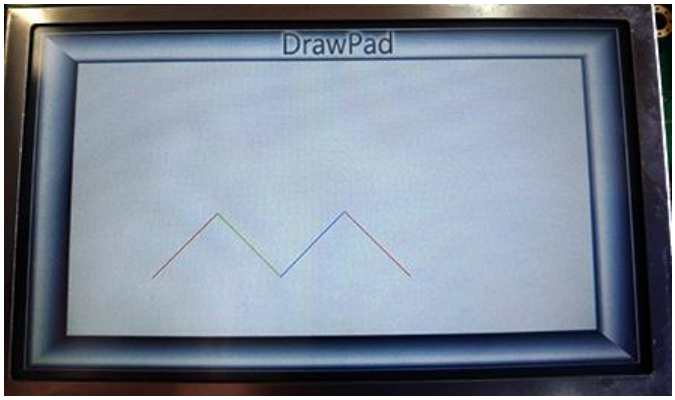


① 描述：在“绘图板”控件刷新范围内，指定位置端点连线
② CMD ： 连续写 16 位数据指令，0x82
③ VP 地址：“绘图板”控件关联的 VP 地址，0x080000
④ Len：16 位数据的个数(阴影部分)
⑤ Type：端点连线指令，0x0002
⑥ Num ： 端点的个数
⑦ X/Y ： 端点坐标
⑧ 颜色：连线颜色，16 位 RGB565 色值
⑨ 指令格式(使用连续写 16 位数据指令 0x82 发送)：

帧头	CMD	VP 地址			Len	Type	Num	颜色	X1	Y1	X2	Y2	X3
AA	82	00080000			11	0002	0007	F800	0096	0096	00C8	00C8	00FA
Y3	X4	Y4	X5	Y5	X6	Y6	X7	Y7	帧尾				
00C8	012C	0096	00FA	0064	00C8	0064	0096	0096	CC33C33C				

注：以上均为 16 进制数据

③线段

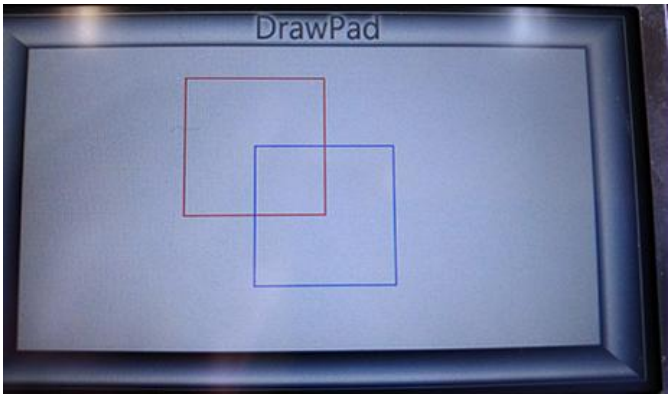


① 描述：在“绘图板”控件刷新范围内，指定位置绘制线段
② CMD ：连续写 16 位数据指令，0x82
③ VP 地址：“绘图板”控件关联的 VP 地址，0x080000
④ Len：16 位数据的个数(阴影部分)
⑤ Type：线段指令，0x0003
⑥ Num ：线段的条数
⑦ X1a/Y1a ：线段起点坐标
X1b/Y1b ：线段终点坐标
⑧ 颜色：线段的颜色，16 位 RGB565 色值
⑨ 指令格式(使用连续写 16 位数据指令 0x82 发送)：

帧头	CMD	VP 地址		Len	Type	Num	X1a	Y1a	X1b	Y1b	颜色	X2a
AA	82	00080000		16	0003	0004	0064	00C8	0096	0096	F800	0096
Y2a	X2b	Y2b	颜色	X3a	Y3a	X3b	Y3b	颜色	X4a	Y4a	X4b	Y4b	颜色
0096	00C8	00C8	07E0	00C8	00C8	00FA	0096	001F	00FA	0096	012C	00C8	F800
												帧尾	
												CC33C33C	

注：以上均为 16 进制数据

④矩形

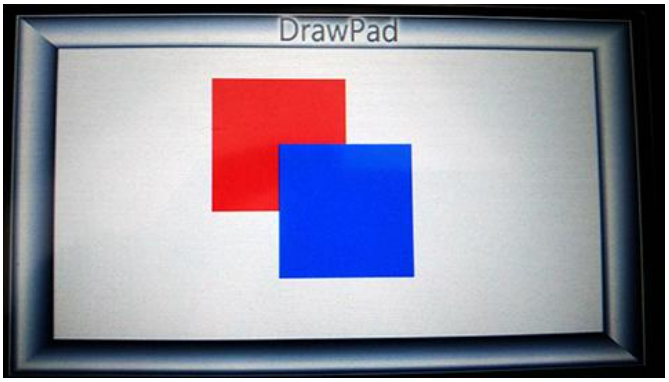


① 描述：在“绘图板”控件刷新范围内，指定位置绘制矩形
② CMD ：连续写 16 位数据指令，0x82
③ VP 地址：“绘图板”控件关联的 VP 地址，0x080000
④ Len：16 位数据的个数(阴影部分)
⑤ Type：矩形指令，0x0004
⑥ Num ：矩形的个数
⑦ X1a/Y1a ：矩形左上角坐标
X1b/Y1b ：矩形右下角坐标
⑧ 颜色：矩形的颜色，16 位 RGB565 色值
⑨ 指令格式(使用连续写 16 位数据指令 0x82 发送)：

帧头	CMD	VP 地址	Len	Type	Num	X1a	Y1a	X1b	Y1b	颜色
AA	82	00080000	0C	0004	0002	0096	0032	00FA	0096	F800
				X2a	Y2a	X2b	Y2b	颜色	帧尾	
				00C8	0064	012C	00C8	001F	CC33C33C	

注：以上均为 16 进制数据

⑤矩形填充



① 描述：在“绘图板”控件刷新范围内，指定位置绘制矩形填充
② CMD ：连续写 16 位数据指令，0x82
③ VP 地址：“绘图板”控件关联的 VP 地址，0x080000
④ Len：16 位数据的个数(阴影部分)
⑤ Type：矩形填充指令，0x0005
⑥ Num ：矩形填充的个数
⑦ X1a/Y1a ：矩形填充左上角坐标
X1b/Y1b ：矩形填充右下角坐标
⑧ 颜色：矩形填充的颜色，16 位 RGB565 色值
⑨ 指令格式(使用连续写 16 位数据指令 0x82 发送)：

帧头	CMD	VP 地址	Len	Type	Num	X1a	Y1a	X1b	Y1b	颜色
AA	82	00080000	0C	0005	0002	0096	0032	00FA	0096	F800
				X2a	Y2a	X2b	Y2b	颜色	帧尾	
				00C8	0064	012C	00C8	001F	CC33C33C	

注：以上均为 16 进制数据

⑥ 图片复制/粘贴

① 工程资源窗口中增加 PG0001 页面，并关联导入的背景图(请参考前面例子)
② 编译下载新工程到模块(请参考前面例子)

① 描述 ：在“绘图板”控件刷新范围内，指定位置复制粘贴页面背景
② CMD ：连续写 16 位数据指令，0x82
③ VP 地址：“绘图板”控件关联的 VP 地址，0x080000
④ Len ：16 位数据的个数(阴影部分)
⑤ Type ：图片复制/粘贴指令，0x0006
⑥ Num ：图片复制/粘贴的个数
⑦ X1/Y1 ：复制区域左上角坐标
X2/Y2 ：复制区域右下角坐标
X3/Y3 ：粘贴区域左上角坐标
⑧ ID ：复制的背景图片所在页面 ID，0x0001
⑨ 指令格式(使用连续写 16 位数据指令 0x82 发送)：

帧头	CMD	VP 地址	Len	Type	Num	ID	X1	Y1	X2	Y2
AA	82	00080000	09	0006	0001	0001	0064	0000	017C	00C8
							X3	Y3	帧尾	
							0064	0023	CC33C33C	

注：以上均为 16 进制数据

⑦ 图标显示

① 资源窗口中右击“图标”，选择“导入图标”
② 根据路径选择图标，并打开。

DI0000： 

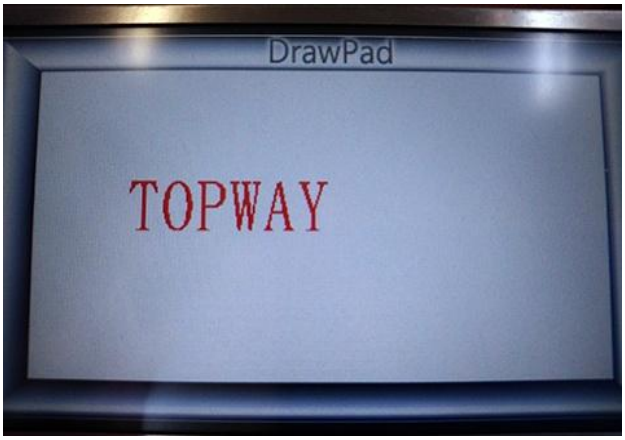
DI0001： 

① 描述 ：在“绘图板”控件刷新范围内，指定位置显示图标
② CMD ：连续写 16 位数据指令，0x82
③ VP 地址：“绘图板”控件关联的 VP 地址，0x080000
④ Len ：16 位数据的个数(阴影部分)
⑤ Type ：图标显示指令，0x0007
⑥ Num ：图标显示的个数
⑦ X/Y ：图标显示坐标
⑧ ID ：图标 ID
⑨ 指令格式(使用连续写 16 位数据指令 0x82 发送)：

帧头	CMD	VP 地址	Len	Type	Num	X1	Y1	ID	X2	Y2	ID
AA	82	00080000	08	0007	0002	0064	0064	0000	00C8	0064	0001
										帧尾	
										CC33C33C	

注：以上均为 16 进制数据

⑧ 字符串显示



① 描述 ：在“绘图板”控件刷新范围内，指定位置显示字符串
② CMD ：连续写 16 位数据指令，0x82
③ VP 地址：“绘图板”控件关联的 VP 地址，0x080000
④ Len ：16 位数据的个数(阴影部分)
⑤ Type ：字符串显示指令，0x0008
⑥ Num ：字符串显示的个数
⑦ X/Y ：字符串显示坐标
⑧ 颜色 ：字体颜色
⑨ FtID ：字库 ID Len1 ：字符串长度
⑩ 指令格式(使用连续写 16 位数据指令 0x82 发送)：

帧头	CMD	VP 地址	Len	Type	Num	X	Y	颜色	FtID	Len1
AA	82	00080000	09	0008	0001	0064	0064	F800	01	06
				字符串(TOPWAY)					帧尾	
				54 4F 50 57 41 59					CC33C33C	

注：以上均为 16 进制数据

- Done –

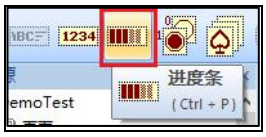
6.11 进度条控件应用

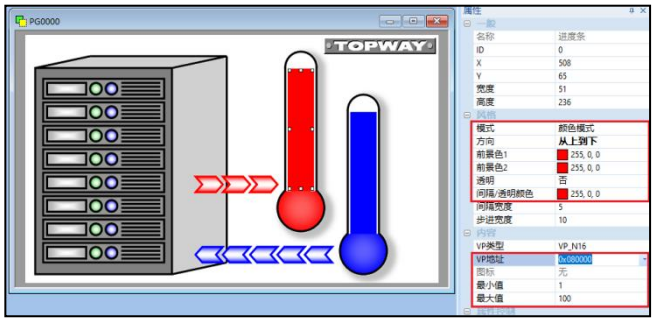
第一步 建立工程(略)

第二步 导入背景图并关联页面(略)

第三步 建立进度条控件与关联 VP

①属性中“方向”选择：“从下到上”
②属性中“前景色 1”、“前景色 2”、“间隔/透明颜色“选择：0xFF0000(255，0，0)
③属性中“最大值”设置：100
④属性中“VP 地址”选择：0x080000
⑤重复以上步骤，建立第二个进度条控件

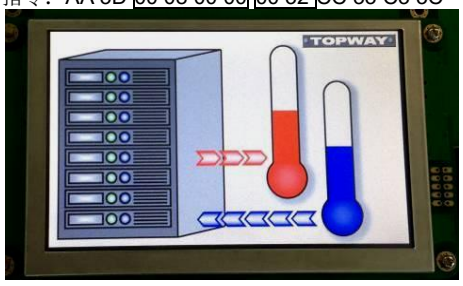




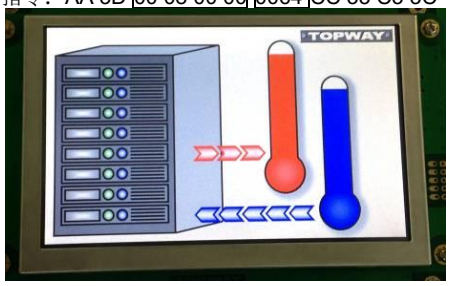
第四步 编译与下载(略)

第五步 连接串口并供电

①连接串口并供电
②发送命令(显示 50%进度)
指令：AA 3D 00 08 00 00 00 32 CC 33 C3 3C



③发送命令(显示 100%进度)
指令：AA 3D 00 08 00 00 0064 CC 33 C3 3C



- Done -

6.12 位图控件应用

第一步 建立工程(略)

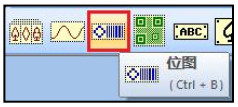
第二步 导入背景图并关联页面(略)

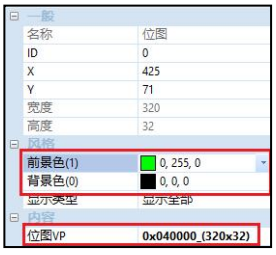
第三步 建立位图 VP 地址

①在资源窗口中，右击“位图变量”选择“新建位图变量”
②输入位图宽度(320)和高度(32)

第四步 创建控件并设置属性

①在页面中创建位图控件
②设置控件属性“位图 VP”选择：0x040000
③设置控件属性“前景色(1)”选择：0x00FF00(0，255，0)
④设置控件属性“背景色(0)”选择：0x000000(0，0，0)

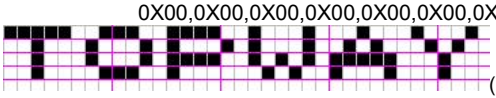




第五步 编译与下载(略)

第六步 连接串口并供电

① 发送命令：首先发送: AA 4B 00 04 00 00 00 00 05 00 CC 33 C3 3C
②最后发送：0XF9,0XCF,0X22,0X22,0X20,0X00,0X00,0X00,0X00,
0X00,0X00,0X00,0X00,0X00,0X00,0X00,0X00,0X00,.....



(可用取模软件取数据)

注：黑色小方块为：1;白色小方块为：0
图中左上方 8 个小方块：11111001(0xF9)第一个字节
第二组 8 个小方块：11001111(0xCF)第二个字节
第三组 8 个小方块：00100010(0x22)第三个字节

- Done -

6.13 二维码控件应用

第一步 建立工程(略)

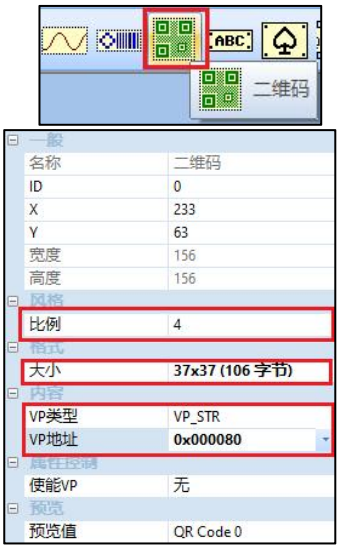
第二步 导入背景图并关联页面(略)

第三步 建立触摸键与关联键盘

① 菜单工具栏点击触摸键
② 在页面工作区拖出矩形区，新建一个触摸键
③ 属性中“呼叫/键盘”选择：“英文键盘”
④ 属性中“标题/值”设置：“Input”
⑤ 属性中“VP 地址”选择：0x000080

第四步 建立二维码控件与关联 VP

①点击菜单工具栏“二维码”控件图标
② 在页面工作区拖出矩形区，新建 2 个“二维码”控件
③ 属性中 X/Y Pop Position 分别设为：465，110
④ 属性中“VP 类型”选择：VP_STR
⑤ 属性中“VP 地址”选择：0x000080
⑥ 属性中“大小”选择：37*37(106Byte)
⑦ 属性中“比例”分别：4 和 7



第五步 编译与下载(略)

第六步 上电显示

①断开 USB 与模块连接
②连接串口并供电
(供电后显示显示效果,见下图)

③点击屏幕中触摸键，弹出的键盘中输入数据
当输入数据,并按下 OK 键后,数据写入 VP 地址中,并同时把数据发送到串口(主机可接收到)
或往 VP“0x000080”地址写入数据
发送以下指令：
AA 42 00 00 00 80 53 48 45 4E 5A 48 45 4E 20 54 4F 50 57 41 59 00 CC 33 C3 3C
④ 可用手机扫二维码，扫描二维码内容

注：1. 内嵌 Keyboard English 最多可支持 35 个 ASCII 字符输入。如需更长，请使用 PIP Keyboard (可支持 127 字符)。
2. VP_STR 最多可支持 127 个字节输入。如需更长，请使用 VP_N16 空间，并使用 0x82，0x83 指令读写。

6.14 自定义 PIP 数字键盘

第一步 建立工程(略)

第二步 建立两个页面，导入图片(略)

① PG0000 是数字输入页面，关联背景图 DP0000
② PG0001 是数字键盘页面，关联背景图 DP0001，详细设置查看第七步。
③ 背景图 DP0002 是键盘按下效果图片

第三步 关联页面与背景图(略)

第四步 建立字符串控件与设置属性值

① 在资源窗口中，右击“字符串”选择“新建 VP”
② 工具栏点击“字符串”，在页面上新建字符串控件
③ 属性中“字体”选择：32_ASCII_SysBold
④ 属性中“字体颜色”选择：0xFF0000(255，0，0)
⑤ 属性中“透明”选择：是
⑥ 属性中“VP 地址”选择 VP：0x000080

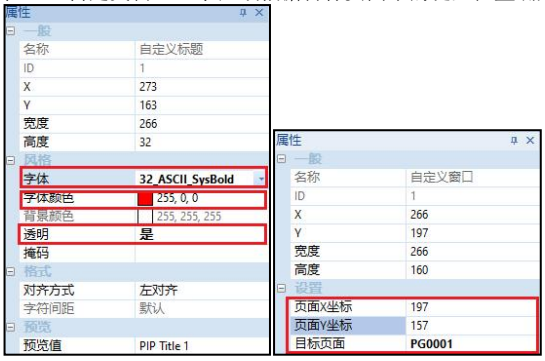
第五步 建立触摸键与设置属性值

① 工具栏点击“触摸键”，在页面上新建触摸键控件
② 属性中“呼叫”选择：PIP 键盘
③ 属性中“VP 地址”选择：0x000080
④ 设置完 PIP 键盘后弹出的“自定义标题”和“自定义窗口”如第六步设置。

第六步 设置 PIP 键盘窗口属性值

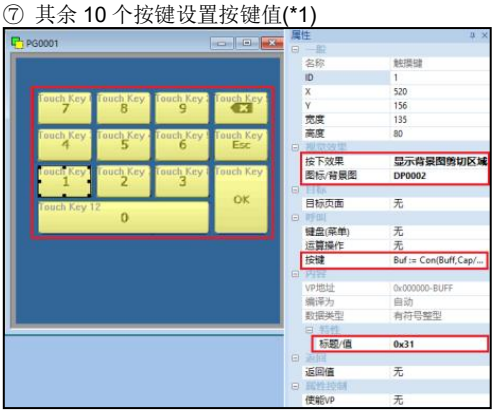
① 自定义标题 属性中“字体”选择：32_ASCII_SysBold
② 自定义标题 属性中“字体颜色”选择：0xFF0000 (255，0，0)
③ 自定义标题 属性中“透明”选择：是
④ 自定义窗口 属性中“页面 X/Y 坐标”分别：197,157
⑤ 自定义窗口 属性中“目标页面” 选择：PG0001

注：（自定义窗口显示区域根据目标页面中的键盘位置确定）



第七步 建立自定义键盘与设置属性值

① 菜单工具栏点击“触摸键”
② 在数字键盘页面，新建 13 个触摸键控件（可复制粘贴）
③ 13 个触摸键属性中“按下效果”选择：显示背景图剪切区域
④ 13 个触摸键属性中“图标/背景图”选择：DP0002
⑤ 其中 10 个按键“呼叫/按键”选择：Buf:=Con(Buff,Cap/Nom(Byte0/Byte1))
⑥ 其他 3 个按键设置：“X”：VP:=删除尾字符(vp)；“ESC”：Esc；“OK”：Enter.
⑦ 其余 10 个按键设置按键值(*1)



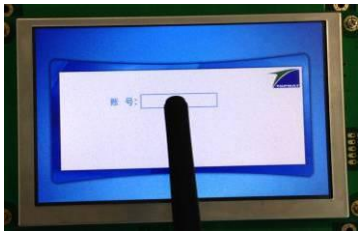
*1：键码值对应表

键码	0x31	0x32	0x33	0x34	0x35	0x36	0x37	0x38	0x39	0x30
按键	1	2	3	4	5	6	7	8	9	0

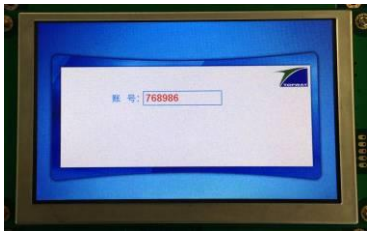
第八步 下载与编译(略)

第九步 上电显示

① 断开 USB 与模块连接
② 向模块提供 5V 电源
③ 上电显示
④ 点击界面账号输入框



⑤ 在 PIP 键盘输入：“768986”
⑥ 点击“确认”，完成



- Done -

6.15 自定义 PIP 英文键盘

第一步 建立工程(略)

第三步 关联页面与背景图(略)

第四步 建立字符串控件与设置属性值

① 在资源窗口中，右击“字符串变量”选择 新建 vp（自动）
② 工具栏点击“字符串”，在页面中新建字符串控件
③ 属性中“字体”选择：32_ASCII_SysBold
④ 属性中“字体颜色”选择： 0xFF0000(255，0，0)
⑤ 属性中“透明”选择：是
⑥ 属性中“VP 地址”选择： 0x000080

第二步 建立两个页面，导入图片(略)

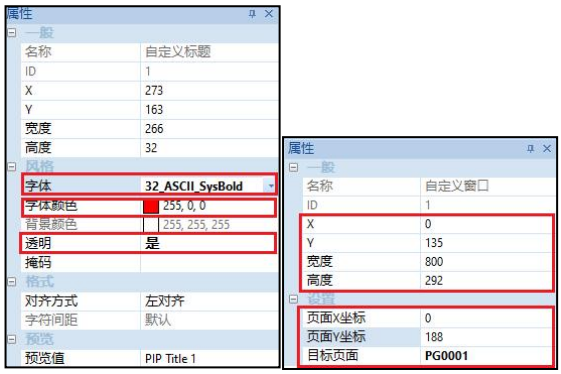
① PG0000 是英文字符输入页面，关联背景图 DP0000；
② PG0001 是字符键盘页面，关联背景图 DP0001，详细设置查看第七步

第五步 建立触摸键与设置属性值

① 菜单工具栏点击“触摸键”
② 在页面工作区拖出矩形区，新建一个触摸键控件
③ 属性中“呼叫”选择：PIP 键盘
④ 属性中“VP 地址”选择：0x000080
⑤ 设置完 PIP 键盘后弹出的“自定义标题”和“自定义窗口”如第六步设置

第六步 设置 PIP 键盘窗口属性值

① 自定义标题 属性中“字体颜色”选择：0xFF0000(255，0，0)
② 自定义标题 属性中“透明”选择：是
③ 自定义窗口 属性中 X=0;Y=135;宽度:800;高度:292
④ 自定义窗口 属性中“目标页面”选择：PG0001
注：（自定义窗口显示区域根据目标页面中的键盘位置确定）



第七步 建立自定义键盘与设置属性值

① 工具栏点击“触摸键”
② 在页面工作区拖出矩形区，创建触摸键控件，一共需要新建 45 个触摸键。为保持大小一致可以复制粘贴。
③ 创建的 45 个触摸键属性中“按下效果”选择：反色
④ 40 个属性中“呼叫”选择：Buf:=Con(Buff,Cap/Nom(Byte0/Byte1))
注：呼叫“按键”设置(*1)，“标题/值”设置(*2)

*1：其中 5 个呼叫按键属性设置如下如下：
“Caps Lock“呼叫按键属性设置为: "CapLock"；
“更正”呼叫按键属性设置为: "VP:=删除尾字符(vp)"；
“确定”呼叫按键属性设置为: "Enter"；“→”呼叫按键属性设置为: "光标右移"
“←”呼叫按键属性设置为: "光标左移"。

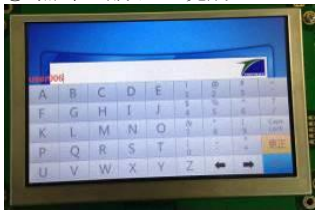
*2：键码值对应表

键码	普通	大写	键码	普通	大写	键码	普通	大写	键码	普通	大写
0x4161	a	A	0x4D6D	m	M	0x5979	y	Y	0x2930	0	)
0x4262	b	B	0x4E6E	n	N	0x5A7A	z	Z	0x5F2D	-	_
0x4363	c	C	0x4F6F	o	O	0x7E60	‘	~	0x2B3D	=	+
0x4464	d	D	0x5070	p	P	0x2131	1	!	0x3F2F	/	?
0x4565	e	E	0x5171	q	Q	0x4032	2	@			
0x4666	f	F	0x5272	r	R	0x2333	3	$			
0x4767	g	G	0x5373	s	S	0x2434	4	#			
0x4868	h	H	0x5474	t	T	0x2535	5	%			
0x4969	i	I	0x5575	u	U	0x5E36	6	^			
0x4A6A	j	J	0x5676	v	V	0x2637	7	&			
0x4B6B	k	K	0x5777	w	W	0x2A38	8	*			
0x4C6C	l	L	0x5878	x	X	0x2839	9	(			

第八步 下载与编译(略)

第九步 上电显示

① 断开 USB 与模块连接
② 向模块提供 5V 电源
③ 上电显示
④ 点击界面用户输入框
⑤ 在 PIP 键盘输入：“user006”，
⑥ 点击“确认”，完成



- Done -

6.16 自定义 PIP 菜单

第一步 建立工程（略）

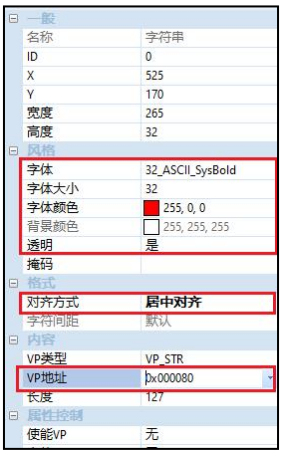
第二步 建立两个页面,导入图片(略)

① PG0000 是菜单输入页面，关联背景图 DP0000
② PG0001 是 PIP 菜单页面，关联背景图 DP0001

第三步 关联页面与背景图

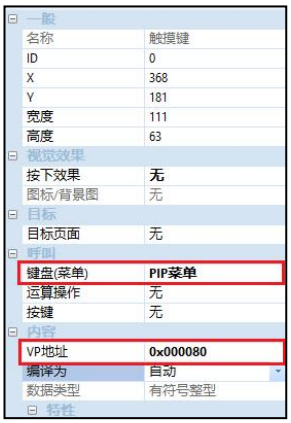
第四步 建立字符串控件与设置属性值

① 于资源窗口中，右击“字符串变量”选择“新建 VP”
② 菜单工具栏点击字符串控件
③ 在页面工作区拖出矩形区，新建一个字符串控件
④ 属性中“字体颜色”选择：0xFF0000 (255，0，0)
⑤ 属性中“透明”选择： 是
⑥ 属性中“字体选择”：32_ASCII_SysBold
⑦属性中“对齐方式”选择：居中对齐
⑧属性中“VP 地址”选择：0x000080



第五步 建立触摸键与设置属性值

① 菜单工具栏点击“触摸键”
② 在页面工作区拖出矩形区，新建一个触摸键控件
③ 属性中“呼叫”选择：PIP 菜单
④ 属性中“VP 地址”选择：0x000080



第六步 设置 PIP 菜单窗口属性值

① 点击 PIP 菜单窗口
② 自定义窗口 属性中“目标页面”选择：PG0001
③ 自定义窗口 属性中“页面 X/Y 坐标”分别：516,169(*1)

(*1：也可按住“Ctrl+鼠标左键”移动 PIP 窗口内容)
(注：PIP 菜单窗口中的 5 个触摸键设置查看第七步)

第七步 建立五个自定义键盘与设置属性值

① 进入 PG0001，然后点击工具栏“触摸键”图标
② 在页面工作区拖出矩形区，新建 5 个触摸键控件
③ 5 个属性中“按下效果”选择：反色
④ 5 个属性中“呼叫”选择：*VP:=Value,Enter
⑤ 第一个触摸键“标题/值”设置：RMB
第二个触摸键“标题/值”设置：HKD
第三个触摸键“标题/值”设置：USD
第四个触摸键“标题/值”设置：YAN
第五个触摸键“标题/值”设置：GBP

第八步 下载与编译(略)

第九步 上电显示

① 断开 USB 与模块连接
② 向模块提供 5V 电源
③ 上电显示
④ 点击界面货币输入框，选择不同货币类型显示

- Done -

6.17 自定义 PIP 日期时钟键盘

第一步 建立工程(略)

第三步 关联页面与背景图

① 分别点击 PG0000 和 PG0001 工作区域,，右面显示页面属性
② 设置页面“背景图”属性如下
PG0000 页面设置为：DP0000-PIP_RTC 显示背景
PG0001 页面设置为：DP0001-PIP_KEY 背景

第四步 导入字库

① 菜单栏 “工具”-->“字体设置”
② 点选 44 号字库右击 “选择”
③ 找到“64_NUM_SevenSegment(40x64) ”字体
④ 点击“选择”选择，点击“OK”完成

第五步 建立自定义键盘与键属性

① 选择 PG0001，点击工具栏的触摸键图标
② 在页面上新建 13 个触摸键控件（可以复制粘贴）
③ 13 个按键属性中“按下效果”选择：反色
④ 数字 0~9 按键属性中“按键”选择：(VP:=连接(VP,Value) 其他 3 个按键设置：“Del”(VP:=删除尾字符(vp))；“Del”(Esc)；“OK” (Enter)
⑤ 数字0~9按键属性中“标题/值”设置对应数值(如下表)

Title/Value	1	2	…	8	9
键值	1	2	…	8	9

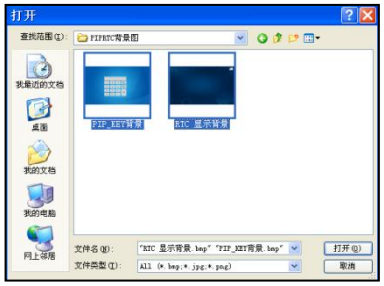
第七步 建立触摸键与关联 PIP 时钟设定键盘

① 选择 PG0000，点击菜单工具栏触摸键图标
② 在页面工作区拖出矩形区，创建触摸键控件于页面
③ 属性中“呼叫”选择：PIP 时钟设定
④ 属性中“按下效果”选择：反色

注：设置PIP键盘后弹出“自定义标题”和“自定义窗口”如第八步设置

第二步 建立 2 个页面 ，导入 2 张背景图

① 在资源窗口中，右击“页面”->新建页面，PG0000和PG0001
② 在资源窗口中，右击“背景图”->导入背景图，DP0000和DP0001



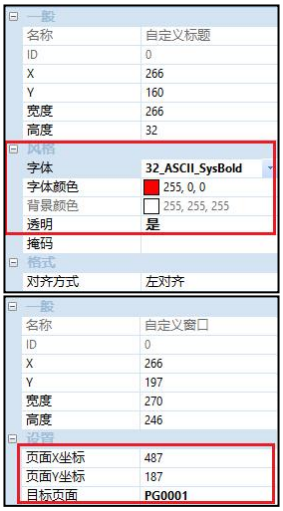
第六步 建立日期时钟控件与属性设置

① 选择 PG0000，点击工具栏日期时钟控件图标
② 在页面工作区拖出矩形区，新建 3 个日期时钟控件
③ 属性中“字体颜色”设置：0x00FFFF (0，255，255)
④ 属性中“透明”选择：是
⑤ 属性中“字体”选择：64_NUM_SevenSegment(40x64)
⑥ 属性中“日期/时间格式”分别选择：“年-月-日 时：分：秒”、“年-月-日”和“时：分：秒”

第八步 设置 PIP键盘窗口属性值

① 自定义标题 属性中“字体颜色”选择：0xFF0000 (255，0，0)
② 自定义标题 属性中“透明”选择：是
③ 自定义标题 属性中“字体”选择：32_ASCII_SysBold
④ 自定义窗口 属性中“页面 X/Y 坐标”分别：474,187
⑤ 自定义窗口 属性中“目标页面”选择：PG0001

注：TPK 选中状态，PIP 窗口显示。“Ctrl+鼠标左键”移动 PIP 窗口内容



第九步 编译与下载(略)

第十步 上电显示

① 断开 USB 与模块连接
② 向模块提供 5V 电源
③ 按下屏幕触摸键位置，弹出键盘，输入 VP 值，点击 “OK”
④ 点击“确认”，完成.

- Done -

6.18 自定义 PIP 中文键盘

第一步 建立工程 （请参考前面例子）

第二步 建立页面并导入背景图和图标

（请参考前面 PIP 键盘例子新建 PG0000 和 PG0001 中文键盘页）

第三步 关联页面与背景图（请参考前面例子）

第四步 建立字符串和设置属性值

① 菜单工具栏点击“字符串”图标
② 在页面工作区拖出矩形区，创建字符串元素于页面中
③ 属性中“字体”选择：32x64_Times New Roman_宋体_160（参考新建 Font2 字库）
④ 属性中“字体颜色”选择：0xFF0000(255，0，0)
⑤ 属性中“透明”选择：是
⑥ 属性中“VP 地址”选择 VP：0x000080

第五步 建立触摸键与设置属性值

① 菜单工具栏点击触摸键
② 在页面工作区拖出矩形区，新建一个触摸键控件
③ 属性中“呼叫”选择：PIP 中文键盘
④ 属性中“VP 地址”选择：0x000080

第六步 设置 PIP 中文键盘窗口属性值

① 自定义标题 属性中，“字体”选择：32x64_Times New Roman_宋体 _160
② 自定义标题 属性中，“字体颜色”选择：0xFF0000(255，0，0)
③ 自定义窗口 属性中 X=0；Y=216；宽度：800；高度：264
④ 自定义窗口 属性中“目标页面” 选择：PG0001

第七步 设置自定义中文键盘页属性

① 菜单工具栏点击触摸键
② 放置“中英文切换按键”，属性中“英文模式图标”选择：DI0000
③ 放置另外 53 个按键(按键可以复制粘贴)，属性中“按下效果”选择：“显示背景图剪切区域”
④ 53 个按键属性中“图标/背景图”选择：DP0000
⑤ 47 个常用键属性中“呼叫”选择：Buf:=Con(Buff,Cap/Nom(Byte0/Byte1))
⑥ Title/Value 设置按键值(*1)

*1：常用按键及键码值表

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

第八步 下载与编译（请参考前面例子）

第九步 上电显示

① 断开 USB 与模块连接
② 向模块提供 12V 电源
③ 上电显示
④ 点击界面用户输入框
⑤ 在弹出的键盘输入：“拓普微 TOPWAYtopway”
⑥ 点击“enter”键，完成

- Done -

7 附录

7.1 附录 A: 快捷键

工程设置及页面布局工具栏

图标	名称	快捷键
	新建工程 New Project	Ctrl + N
	打开工程 Open Project	Ctrl + O
	保存工程 Save Project	Ctrl + S
	关闭工程 Close Project	--
	工程设置 Project Setting	--
	字体设置 Font Setting	--
	箭头 Arrow	--
	复制 Copy	Ctrl + C
	粘贴 Paste	Ctrl + V
	撤销 Undo	Ctrl + Z
	重做 Redo	Ctrl + Y
	左对齐 Align Left	--
	右对齐 Align Right	--
	顶部对齐 Align Top	--
	底部对齐 Align Bottom	--
	水平中线对齐 Align Centers Horizontally	
	垂直中线对齐 Align Centers Vertically	
	水平分布 Distribute Horizontal	-
	垂直分布 Distribute Vertical	-
	等高 Same Height	
	等宽 Same Width	

触摸键及虚拟键工具栏

图标	简称	名称	快捷键
	TPK	触摸键 Touch Key	Ctrl + K
	SDR	滑动调节 Slider	--
	RNG	环形调节 Ring	--
	TPK-RPT	长按触摸键 TPK_Repeat	--
	TPK-SW	开关触摸键 TPK_Switch	--
	SWP-PG	滑动翻页 Swap_Page	--
	SDR2	双指滑动 Slider_2	--
	RNG2	双指旋转 Ring_2	--
	VPK	虚拟键 Virtual Key	--

快捷触摸键及键盘工具栏

图标	简称	名称	快捷键
	TPK-CHEN	中英切换 TPK_CHEN	--
	TPK-ESC	退出键 TPK_ESC	--
	TPK-BKS	退格键 TPK_Backspace	--
	TPK-ARL	光标左移 TPK_Left	--
	TPK-ARR	光标右移 TPK_Right	--
	TPK-ENT	确认键 TPK_Enter	--
	TPK-CAP	大写键 TPK_Caps	--
	TPK-CHR	字符输入按键 TPK_Char	--
	Num1_KB	数字键盘 1 Number1 Keybaord	--
	Num2_KB	数字键盘 2 Number2 Keybaord	--
	ABC_KB	英文键盘 English Keyboard	--

	等大小 Same Size	
	置顶 Move To Top	--
	置底 Move To Bottom	--
	显示所选控件 Show Selected Element	--
	隐藏所选控件 Hide Selected Element	--

控件工具栏

图标	简称	名称	快捷键
	STR	字符串 String	Ctrl + T
	STR-STL	滚动字符串 String Stroll	--
	N16 N32 N64	数字 Number	Ctrl + I
	TMR	计时器 Timer	--
	RTC	日期时钟 Real Time Clock	Ctrl + R
	IDX_BIT	位变量图标 Bit Icon	--
	IDX	变量图标 Indexed Icon	--
	ICO	静态图标 Static Icon	Ctrl + Shift + I
	ANI	动画 Animation	Ctrl + Shift + A
	HND	表盘 Tachometer	--
	R32	表盘 Tachometer	--
	TCM	表盘 Tachometer	--
	CLK-R	模拟时钟 Round_Clock	--

	List	表格 Strings Table	--

编译下载工具栏

图标	名称	快捷键
	编译工程 Generate Files	F7
	下载 Download to Module	F9

控件工具栏

图标	简称	名称	快捷键
	STS	静态文本 Static String	Ctrl + Shift + T
	I16 I32	十进位图标 Decimal Icon	--
	B16	进度条 Progress Bar	Ctrl + P
	G16	曲线 Graph	Ctrl + G
	BP1	位图 Bitmap	Ctrl + B
	QRCode	二维码 QRCode	--
	DPD	绘图板 Draw Pad	--

7.2 附录 B: 工程限定

页面，图片资源和 VP 变量限定关系

图标	简称	名称	关联 VP	关联图标	数量限定	内存限定	编号/地址范围
	PAGE	页面 Page	--	背景图 IMG_BKG	≤1000/工程	256M byte	PG0000 ~ PG0999
	IMG_BKG	背景图 Background Image	--	--	≤10000/工程 (*1)		DP0000 ~ DP9999
	IMG_ICO	图标 Icon	--	--	≤10000/工程 (*2)		DI0000 ~ DI9999
	IMG_IcoLib	图图标库(*4) Icon Library	--	--	≤999/工程		ICO001~ICO999
	IMG_ANI	动画 Animation	--	--	≤1000/工程 (*2)(*3)		ANI000 ~ ANI999
	VP_STR	字符串变量 String Variable	--	--	≤1024 /工程	1024(MAX) x (127+1)byte	0x000000 ~ 0x01FF80
	VP_N16	16 位数字变量 16bit Integer Variable	--	--	≤32768/工程	32768(MAX) x (2)byte	0x080000 ~ 0x08FFFE
	VP_N32	32 位数字变量 32bit Integer Variable	--	--	≤16368/工程	16368(MAX) x (4)byte	0x020000 ~ 0x02FFBC
	VP_N64	64 位数字变量 64bit Integer Variable	--	--	≤7168/工程	7168(MAX) x (8)byte	0x030000 ~ 0x03DFF8
	VP_G16	曲线变量 16bit Graph Variable	--	--	≤16384 /工程	16384(MAX) x (8)byte (dynamic array allocation)	0x060000 ~ 0x07FFF8
	VP_BP1	位图变量 Bitmap Variable	--	--	≤2048 /工程	2048(MAX) x (64)byte (dynamic array allocation)	0x060000 ~ 0x05FFBF
	VP_SYS	系统寄存器变量 System Register	--	--	≤255 /工程	255(MAX) x (1)byte	0xFFFF00 ~ 0xFFFFFF

注:

*1. 导入的背景图必须大于工程分辨率的 1/2.
- SGTools 会自动拉伸或缩放至与分辨率大小相同
- 图像拉伸缩放不排除有颜色失真现象
- 导入的背景图分辨率大小最好是与工程分辨率大小相同.

*2. 图标(IMG_ICO)和动画(IMG_ANI)导入要求/限制

工程分辨率	大小(分辨率)限制
320x240	320x240 max. (Full-Screen)
480x272	480x272 max. (Full-Screen)
640x480	131072pixels (42% of Full Screen)
800x480	131072pixels (34% of Full Screen)
800x600	131072pixels (27% of Full Screen)
1366X480	131072pixels (20% of Full Screen)

*3. 帧间隔最小 100ms, 每个动画最大 128 帧. 可循环播放

*4. 图标库支持 PNG 图片显示，仅部分型号支持.

页面/控件/资源限定关系

图标	简称	名称	关联 VP	关联图标	数量限定/页面	内存限定	编号范围（每个页面）
	**TPK**	触摸键 Touch Key	--	图标(IMG_ICO) 背景图(IMG_BKG)	≤256	--	0 ~ 255
	**SDR**	滑动调节 Slider	VP_N16 VP_N32 VP_REG	--	≤256	--	0 ~ 255
	**RNG**	环形调节 Ring	VP_N16 VP_N32 VP_REG	--	≤256	--	0 ~ 255
	**TPK-RPT**	长按触摸键 TPK_Repeat	VP_N16 VP_N32 VP_REG	图标(IMG_ICO) 背景图(IMG_BKG)	≤256	--	0 ~ 255
	**TPK-SW**	开关触摸键 TPK_Switch	VP_N16 VP_N32 VP_REG	图标(IMG_ICO) 背景图(IMG_BKG)	≤256	--	0 ~ 255
	**SWP-PG**	滑动翻页 Swap_Page	--	--	≤1	--	0 ~ 255
	**SDR2**	双指滑动 Slider_2	VP_N16 VP_N32 VP_REG	--	≤2	--	0 ~ 255
	**RNG2**	双指旋转 Ring_2	VP_N16 VP_N32 VP_REG	--	≤1	--	0 ~ 255
	**VPK**	虚拟键 Virtual Key	VP_N16 VP_N32	--	≤64	--	0 ~ 63
	**STS**	静态文本 Static String	--	--	≤128	--	0 ~ 127
	**STR**	字符串 String Element	VP_STR VP_N16	--	≤128	--	0 ~ 127
	**STR-STL**	字符串 String Element	VP_STR VP_N16	--	≤128	--	0 ~ 127
	**N16** **N32** **N64**	数字 Number Element	VP_N16 VP_N32 VP_N64 VP_REG	--	≤120	--	0 ~ 119
	**TMR**	计时器 Timer	VP_N32 (timer only)	--	≤8	--	0 ~ 7
	**RTC**	日期时钟 Real Time Clock	--	--	≤12	--	0 ~ 11
	**ICO**	静态图标 Static Icon	--	图标 IMG_ICO	≤128	--	0 ~ 127
	**ANI**	动画 Animation	--	动画资源 IMG_ANI	≤8	--	0 ~ 7
	**IDX_BIT**	位变量图标 Bit Icon	VP_N16 VP_N32 VP_REG	图标 IMG_ICO	≤64	--	0 ~ 63
	**IDX**	变量图标 Indexed Icon	VP_N16 VP_REG	图标 IMG_ICO	≤64	--	0 ~ 63
	**TCM**	表盘 Tachometer	VP_N16 VP_N32 VP_REG	图标 IMG_ICO	≤8	--	0 ~ 7
	**HND**	表盘 Tachometer	VP_N16 VP_N32 VP_REG	--	≤8	--	0 ~ 7
	**R32**	表盘 Tachometer	VP_N16 VP_N32 VP_REG	--	≤8	--	0 ~ 7
	**I16** **I32**	十进位图标 Decimal Icon	VP_N16 VP_N32 VP_REG	图标 IMG_ICO	≤32	--	0 ~ 31
	**B16**	进度条 Progress Bar	VP_N16 VP_REG	图标 IMG_ICO	≤32	--	0 ~ 31
	**G16**	曲线 Graph	VP_G16	--	≤12	--	0 ~ 11
	**BP1**	位图 Bitmap	VP_BP1	--	≤32	--	0 ~ 31
	**QRCode**	二维码 QRCode	VP_STR VP_N16 (in seq.)	--	≤10	--	0 ~ 9
	**DPD**	绘图板 Draw Pad	VP_N16 (in seq.)	--	≤8	--	0 ~ 7
	**CLK-R**	模拟时钟 Round Clock	--	图标 IMG_ICO	≤12	--	0 ~ 11

注：更改配置文件,可设定页面中每种控件的最大支持数量.

系统变量资源限定关系/功能描述

地址	系统变量	功能描述	注意事项
0xFFFF00 : 0xFFFF07	Timer_Ctrl0 : Timer_Ctrl7	0=停止计时器 1=倒计时 3=开始计时器 倒计时至 0x00000000 正计时至 0x7FFFFFFF	(*1)
0xFFFF10	RTC_Year	系统 RTC -年，基础值：2000，范围：0~99	应该同时更改变量值 0xFFFF10~0xFFFF16
0xFFFF11	RTC_Month	系统 RTC -月，范围：1~12	
0xFFFF12	RTC_Day	系统 RTC -日，范围：1~31	
0xFFFF13	RTC_Hour	系统 RTC -时，范围：0~23	
0xFFFF14	RTC_Minute	系统 RTC -分，范围：0~59	
0xFFFF15	RTC_Second	系统 RTC -秒，范围：0~59	
0xFFFF16	RTC_Set	1=同步 RTC，将系统变量 0xFFFF10~0xFFFF15 的值写入 RTC	
0xFFFF20	Buzzer	蜂鸣器鸣叫持续时长，范围：0~63	详见工程设置 (*1)
0xFFFF21	Backlight	实时背光亮度设置，范围：0~63	
0xFFFF22	ScreenSaverBacklight	屏保背光亮度，范围：0~63	有关详细信息，请参阅屏保部分 (*1) (*2)
0xFFFF23	ScreenSaverPage_H	屏保显示页面，范围：0~999	
0xFFFF24	ScreenSaverPage_L		
0xFFFF25	ScreenSaver Timer1_H	屏保延时时间 1，范围：0~65535	
0xFFFF26	ScreenSaver Timer1_L		
0xFFFF27	ScreenSaver Timer2_H	屏保延时时间 2，范围：0~65535	
0xFFFF28	ScreenSaver Timer2_L		
0xFFFF2E	OTG_Mode	用于设定 U 盘连接屏幕时，屏幕是否进入 OTG 模式界面 0=插入 U 盘就进入 OTG 模式 1=仅 U 盘中有工程包"x.img"文件时才进入 OTG 模式 2=仅上电 4 秒内,检测到 U 盘，就进入 OTG 模式 3=仅上电 4 秒内,检测到 U 盘且有"x.img"文件,进入 OTG 模式 4=禁用 OTG 模式	
0xFFFF2D	LangIDSetting	StringTable 表语言数量索引值	(*1)
0xFFFF30	Font_CodePage	字体代码页索引值	(*1) (*3)
0xFFFF31	Font_Country	字体国家码索引值	(*1) (*4)

注:

*1.可以通过指令 0x3B 更改寄存器的值

*2. 更改屏保延时时间可以通过命令 0x82 同时更改多个变量的值

*3. 代码页索引值

索引	代码页
1	437(OEM 美国)
2	737(OEM 希腊语 437G)
3	852(OEM 拉丁语 II)
4	860(OEM 葡萄牙语)
5	863(OEM 加拿大法语)
6	865(OEM 挪威语)
7	866(OEM 俄语)
8	874(ANSI/OEM 泰语)
9	932(ANSI/OEM 日语 Shift JIS)
10	1250(ANSI 中欧)
11	1251(ANSI 西里尔文)

索引	代码页
12	1252(ANSI 拉丁语 I)
13	1253(ANSI 希腊语)
14	1254(ANSI 土耳其语)
15	1255(ANSI 希伯来语)
16	1256(ANSI 阿拉伯语)
17	1257(ANSI 波罗的海)
18	1258(ANSI/OEM 越南)
19	GB2312 (Simplified Chinese)
20	GBK (Simplified Chinese)
21	ECU-KR (Korea)
22	BIG5 (Traditional Chinese)

*4. 国家码索引值

索引	国家码
1	USA
2	France
3	Germany
4	UK
5	Denmark Ⅰ
6	Denmark Ⅱ

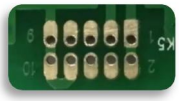
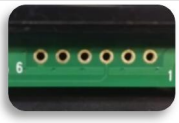
索引	国家码
7	Denmark Ⅱ
8	Sweden
9	Italy
10	Spain
11	Japan

7.3 附录 C: 智能模块接口功能说明

智能模块的接口按功能划分可以分为以下两类，
1、工作接口：正常使用模式下需要用到的接口，模块供电和串口通信。
2、烧录接口：用于显示工程的下载、更新。

说明：

1、工作接口(按常用顺序排序)

序号	类型	说明(*1)(*2)
1	 10 Pin 排线口	Pin 1,2,3 是 VDD Pin 5 是 RX Pin 6 是 TX Pin 8,9,10 是 GND
2	 8 Pin 插线口(1)	Pin 1,2 是 VDD Pin 4 是 TX Pin 5,6 是 RX Pin 7,8 是 GND
3	 6 Pin 端子口	该端口据型号而定，RS232、RS485、Uart 都有可能,具体参考用户手册
4	 10 Pin 插针孔	Pin 1,2,3 是 VDD Pin 5 是 RX Pin 6 是 TX Pin 8,9,10 是 GND
5	 6 Pin 插针孔	该端口据型号而定，RS232、RS485、Uart 都有可能,具体参考用户手册
6	 8 Pin 插线口(2)	同序号 2，但 Pin 脚间距不同 Pin 1,2 是 VDD Pin 4 是 TX Pin 5,6 是 RX Pin 7,8 是 GND
7	 RJ45 通信网口	目前不支持 POE 供电。需要单独供电该接口主要用于协议通信，与显示工程更新升级使用。

2、烧录接口

序号	USB 类型	适用方式(*3)(*4)
1	 Mini	PC 下载 OTG(U 盘升级)
2	 Type A	仅支持 OTG(U 盘升级)
3	 Type C	PC 下载 OTG(U 盘升级)

注:

(*1) 供电电压及串口类型，具体参考用户手册，备注中是常规型号。

(*2) 定制品的接口类型不在该文档说明范畴，属于客户自己提供。

(*3) PC 下载，使用 USB 线缆连接 PC 与模块 USB 接口。然后通过开发工具点击“下载”（F9）按钮，将工程下载到模块中完成更新、下载。

(*4) OTG(U 盘升级)，使用 OTG 线缆连接 U 盘和模块 USB 接口。然后模块上电，自动读取 U 盘中的镜像文件，进行工程更新。

7.4 附录 D: 工程下载方法

用户在 SGTools 中制作的工程界面，可以通过以下几种方式下载到屏幕中：

序	下载方法	说明
1	SGTools 一键下载	在 PC(电脑)上通过 SGTools 的一键下载按钮，快速下载工程。
2	量产工具下载	在 PC(电脑)上通过”量产工具”可自动、快速下载工程。
3	U 盘(OTG)下载	屏幕上电或运行中自动检测并读取 U 盘中的工程文件并更新工程。

1. SGTools 一键下载

操作步骤:

① 使用 USB 线缆连接屏幕的 USB 和 PC 的 USB 口，连接成功后, PC 提示发现 USB 设备.

② SGTools 打开工程，点击工具栏"  " 下载按钮

③ SGTools 识别到名字为”TOPWAY” 的移动磁盘，会弹出对话框,点击对话框中的 "下载" 按钮开始下载.

④ 下载成功后,拔掉屏幕的 USB 线. 重新上电. 完成.

注意:

1. 更新开始后,屏幕中原先的工程会被覆盖.
2. 屏幕先断电后，才可以通过 USB 线缆与 PC 连接

2. 量产工具下载

操作步骤:

①打开量产工具 Smart LCD Production Tools
② 选择 IMG 格式工程包。
③ 点击开始下载按钮(START)。
④ 屏幕通过 USB 线连接电脑，量产工具自动识别并下载。下载完成后屏幕断开 USB 连接即可。

注意:

1. 在 SGTools: 菜单->工具->选项，弹出的对话框中勾选：“输出工程镜像文件(256MB 镜像)”
   编译工程后会生成 IMG 格式工程包文件。

3. U 盘(OTG)下载

操作步骤:

① SGTools 编译工程并生成 IMG 格式工程包文档.

文件名称,如下表:

NO.	文件名称	说明
1	Project_Image.img	工程包(该文档包中包含变量、控件、图像、配置信息)等

② 把工程包文件和校验文件拷贝到 U 盘根目录

③ 使用支持 OTG 的 USB 线缆连接 U 盘和屏幕，重新给屏幕上电。

④ 屏幕成功检测到 U 盘和工程包文件和检验文件后开始更新界面，并做校验比对

⑤ 更新和校验都完成后，屏幕会提示更新完成。拔掉 U 盘或者拔掉 OTG 线即可。

注意:

1. 更新开始后,会覆盖掉模块中原先的工程.
2. U 盘要求：①容量最好在 16G 以下；②U 盘文件格式 FAT 或 FAT32；③U 盘不能分区或作为启动盘。

7.5 附录 E：CRC Calculate

```
uint16_t const CRC16[256]={
/* 16: 8005 reflected */
  0x0000,0xc0c1,0xc181,0x0140,0xc301,0x03c0,0x0280,0xc241,
  0xc601,0x06c0,0x0780,0xc741,0x0500,0xc5c1,0xc481,0x0440,
  0xcc01,0x0cc0,0x0d80,0xcd41,0x0f00,0xcfc1,0xce81,0x0e40,
  0x0a00,0xcac1,0xcb81,0x0b40,0xc901,0x09c0,0x0880,0xc841,
  0xd801,0x18c0,0x1980,0xd941,0x1b00,0xdbc1,0xda81,0x1a40,
  0x1e00,0xdec1,0xdf81,0x1f40,0xdd01,0x1dc0,0x1c80,0xdc41,
  0x1400,0xd4c1,0xd581,0x1540,0xd701,0x17c0,0x1680,0xd641,
  0xd201,0x12c0,0x1380,0xd341,0x1100,0xd1c1,0xd081,0x1040,
  0xf001,0x30c0,0x3180,0xf141,0x3300,0xf3c1,0xf281,0x3240,
  0x3600,0xf6c1,0xf781,0x3740,0xf501,0x35c0,0x3480,0xf441,
  0x3c00,0xfcc1,0xfd81,0x3d40,0xff01,0x3fc0,0x3e80,0xfe41,
  0xfa01,0x3ac0,0x3b80,0xfb41,0x3900,0xf9c1,0xf881,0x3840,
  0x2800,0xe8c1,0xe981,0x2940,0xeb01,0x2bc0,0x2a80,0xea41,
  0xee01,0x2ec0,0x2f80,0xef41,0x2d00,0xedc1,0xec81,0x2c40,
  0xe401,0x24c0,0x2580,0xe541,0x2700,0xe7c1,0xe681,0x2640,
  0x2200,0xe2c1,0xe381,0x2340,0xe101,0x21c0,0x2080,0xe041,
  0xa001,0x60c0,0x6180,0xa141,0x6300,0xa3c1,0xa281,0x6240,
  0x6600,0xa6c1,0xa781,0x6740,0xa501,0x65c0,0x6480,0xa441,
  0x6c00,0xacc1,0xad81,0x6d40,0xaf01,0x6fc0,0x6e80,0xae41,
  0xaa01,0x6ac0,0x6b80,0xab41,0x6900,0xa9c1,0xa881,0x6840,
  0x7800,0xb8c1,0xb981,0x7940,0xbb01,0x7bc0,0x7a80,0xba41,
  0xbe01,0x7ec0,0x7f80,0xbf41,0x7d00,0xbdc1,0xbc81,0x7c40,
  0xb401,0x74c0,0x7580,0xb541,0x7700,0xb7c1,0xb681,0x7640,
  0x7200,0xb2c1,0xb381,0x7340,0xb101,0x71c0,0x7080,0xb041,
  0x5000,0x90c1,0x9181,0x5140,0x9301,0x53c0,0x5280,0x9241,
  0x9601,0x56c0,0x5780,0x9741,0x5500,0x95c1,0x9481,0x5440,
  0x9c01,0x5cc0,0x5d80,0x9d41,0x5f00,0x9fc1,0x9e81,0x5e40,
  0x5a00,0x9ac1,0x9b81,0x5b40,0x9901,0x59c0,0x5880,0x9841,
  0x8801,0x48c0,0x4980,0x8941,0x4b00,0x8bc1,0x8a81,0x4a40,
  0x4e00,0x8ec1,0x8f81,0x4f40,0x8d01,0x4dc0,0x4c80,0x8c41,
  0x4400,0x84c1,0x8581,0x4540,0x8701,0x47c0,0x4680,0x8641,
  0x8201,0x42c0,0x4380,0x8341,0x4100,0x81c1,0x8081,0x4040,
};

static __inline uint16_t rshiftu16(uint16_t value, int nb)
{
   return (uint16_t)((value >> nb) & ~((( uint16_t) 0x8000) >> (nb-1)));
}
uint16_t crc16_calc(unsigned char *q, int len)
{
   uint16_t crc = 0xffff;
   while (len-- > 0)
      crc=(rshiftu16(crc,8) ^ CRC16[(crc ^ *q++) & 0xff]);
   return crc;
}
```

7.6 附录 F: 常见问题

序	问答
1	问：如何把工程界面下载到屏中？ 答：三种下载方式 通过 PC 下载: 1. 屏断开与电源连接(注意:不要给屏供电) 2. 用 USB 线连接屏和电脑 3. 通过 SGTools 或量产工具下载界面工程到屏存储器中 通过 U 盘下载: 1. SGTools 菜单栏选择 Tool->Option->"Output Project Image File(256MByte Image)" [勾选] 2. 编译工程(F7),拷贝"Output"文件夹下的"Project_Image.img"和"project.chk"到 U 盘根目录 3. 使用 OTG 线连接屏(断电)和 U 盘,再次上电,屏自动读取 U 盘数据并更新 拷贝/粘贴： 1 用 USB 线连接屏和电脑 2. 编译工程(F7),拷贝"Output"文件夹下的"THMT"和"FONT" 到屏存储器中
2	问：如何处理 PC 无法识别屏存储设备？ 答：首先确保屏没有供电. 若有供电,断开电源和 USB 连接. 再次用 USB 线连接屏和电脑(正常情况下,电脑会发现 USB 存储设备). 注意： 请确保 PC 的 USB 端口供电稳定充足(500mA/5V). 可尝试连接 PC 台式机后面板 USB 端口. 可尝试使用双 USBA 线连接
3	问：如何设置通信波特率？ 答：更改波特率有两种方法. 方法 A. SGTools 工程设定窗口中可设置波特率参数,设定完成后 重新下载工程到屏中.屏每次上电都会以工程设定的波特率为准. 方法 B.通过指令修改波特率,修改后约 1S 后生效. 注意:断电后再次上电,波特率以工程设定的波特率为准
4	问：发送指令无响应怎么处理？ 答：请尝试从以下方面检查 1. 串口连接是否正确(交叉连线 Rx 接 Tx,Tx 接 Rx) 2. 波特率是否一致 3. 发数据是否以十六进制(Hex)方式发送 4. 接口电平是否一致(主板和屏需同时为 RS232 或 TTL 电平) 5. 电源是否共地
5	问：如何让上电时不显示"Starting RTC. Please wait a few seconds..."画面信息? 答："Starting RTC..." 为时钟初始化提示信息: 若不使用 RTC 时钟,可通过工程设置(Project Setting)中的 RTC Mode 参数设置为”Disable” 若使用 RTC 时钟时,首先需 RTC Mode 参数设置为”Enable”, 在 RTC Mode 为 Enable 时, 屏安装电池后每次上电就不再显示”Starting RTC……”初始化信息, 否则上电后每次都显示. 注意: RTC Mode 为”Disable”时,控件 RTC 时钟显示停止走.
6	问：如何设置屏幕保护亮度？ 答：在 SGTools 的"工程设置"中可设置 屏幕保护时间、屏保亮度, 当屏在设定的屏保时间无任何操作时, 会自动进度屏保(屏亮度值为屏保亮度)
7	问：如何退出屏幕保护？ 答：点击触摸屏任意位置或发送指令跳页、刷新数据等即可退出屏幕保护.
8	问：如何进行触摸屏校准? 答：两种方法可进行触摸屏校准 方法 A.发送触摸屏校准指令,屏会进入校准界面. 方法 B.按住屏右上区域再上电,屏会进入校准界面.

9	问：如何控制控件内容隐藏和显示? 答：每个控件都有使能 VP(Enable VP)属性,通过往 VP 中写数据 0x00 或 0x01 可控制隐藏或显示. 首先设定使能 VP 属性,设置一个 VP 地址. 其次通过串口发送写地址指令,往 VP 中写 0x00 或 0x01.
10	问：如何显示中文? 答：屏支持 3 种中文字库码,分别是 GB2312、GBK、BIG5. 其中 GB2312 最为常用 6763 常用汉字,基本满足常用的汉字显示需求. 使用方法: - 控件字体属性设置中文字库 - 主板通过指令把中文字码写入控件关联的 VP 地址中,屏即可显示出中文汉字 注意：SGTools 安装包中默认附带了几种常见大小的 GB2312 字库. 如：24x24、48x48 等,若需要其他点阵大小的字库可通过 SGTools 生成.(详见:字库生成说明书)
11	问：如何显示小数? 答：有两种方式可以显示小数数字 **整型数据显示为小数方式** 1.设定控件参数(整数位显示个数、小数位显示个数) 2. VP 地址的数据为整型(signed/unsigned 类型) 例：VP 地址中数据为数字 1234,控件参数属性设置为：2 位整数位、2 位小数位. 实际显示结果为:12.34 **浮点型数据小数为小数方式** 1.设定控件参数(整数位显示个数、小数位显示个数) 2. VP 地址的数据为浮点型(float 类型) 例：VP 地址中数据为浮点型数字 0x414570A4,控件参数属性设置为：2 位整数位、2 位小数位. 实际显示结果为:12.34
12	问：RTS 引脚需要连接吗? 答：RTS 引脚(忙信号)功能是串口通信过程中,主板在给屏长时间发送大量数据时,屏的串口缓冲区已满,屏会置 RTS 引脚为高电平,缓冲区可以接受数据时会置 RTS 引脚为低电平. 因屏内置有 32KByte 的指令接收缓冲区,绝大部分情况下不需要连接 RTS 引脚去判断忙信号.
13	问：屏可以存储多少张全屏显示的图片? 答：存储多少张图与图的分辨率有关系. 800x480 分辨率(约 340 张)、480x272 分辨率(约 1000 张)、800x600 分辨率(约 273 张) 每张图片可被不同的页面重复使用,存储页面可达 1000 个画面. 完全不用担心显示画面不够的问题.
14	问：对客户导入到 SGTools 中的图片有什么要求? 答：客户导入 SGTools 中的图片有三种(背景图、ICON 图标、动画图片) 背景图 - 大小：尺寸大小同工程页面大小相同. - 格式：建议 24 位深度的 BMP 格式图像 ICON 图标 - 大小：图标大小不能超过背景图大小(1/4 背景图大小) - 格式：建议 24 位深度的 BMP 格式图像 动画图标 - 大小：图标大小不能超过背景图大小(1/4 背景图大小) - 格式：建议 24 位深度的 BMP 格式图像 注意:不符合要求可能产生的问题：无法导入(提示导入失败)、显示变形、显示乱

8 版本信息

版本号	修改备注	制订/修改	发布日期
1.03	-更新说明书封面 -增加“2.1 名称定义”SYS_VP 系统寄存器变量 -更新“2.2 控件列表”部分控件功能描述 -更新所有控件截图 -更新章节“2.3.2~2.3.18”属性描述 -增加“2.4.1 呼叫-键盘/菜单”四种键盘(PIP) -更新“2.4.2 呼叫-按键”部分描述 -更新“2.5 窗口设置”举例的图片修改描述，同时增加部分窗口的功能描述 -更新“4.1 制作第一个显示界面”“4.3 显示字符串”用图用词 -增加“5.1 附录 A:快捷键”部分快捷键，更改布局 -更新“5.2 附录 B:工程限定”部分描述，增加 VP_SYS	Huchubin	2020-02-11
1.04	-更新“4.3RGTools 控件属性及功能”之前的章节顺序排布 -更新“4.2.5 字库设置”的字库窗口图片 -增加“7.1 附录 A”新建快捷“表格”功能	Huchubin	2020-05-15
1.05	修改“5.3.2 显示控制”的设置控件字体颜色(0x7E)和设置控件背景色(0x7F)的指令格式	Liaoliliang	2020-08-26
1.06	-更新“4.1 界面组成”的图片 -更新“4.2.3 工程设置（菜单栏→工具→工程设置）”的图片，增加命令格式，命令超时 -更新“4.2.5 字库设置”的字库配置 2 的图片 -增加“7.1 附录 A”的工程设置及页面布局工具栏的 5 种对齐方式：水平中线对齐、垂直中线对齐、等高、等宽、等大小 -更新“7.2 附录 B: 工程限定”的 16 位数字变量、32 位数字变量、64 位数字变量的数量限定、地址范围 -修改“5.3.2 显示控制”的设置字库(0xE7)，例.[主机]: AA E5 03 07 CC 33 C3 3C，修改为 AA E7 03 07 CC 33 C3 3C	Liaoliliang	2021-4-08
1.07	SGTools 产品说明书合并到 RGTools -3.2.1、3.2.2、3.2.3 增加 SGTools 部分控件类型 -3.3.1 页面和图像资源，增加图标库描述 -4.1 更新界面图片 -4.2.1 更新新建工程页面，工程名称修改为 Displayprj； 修改设备型号说明； -4.2.2 增加 U 盘升级方法跳转超链接“见 7.3 附录 C: 下载工程包方法” -4.2.5 字库设置，增加字体配置 3 说明 -4.3 增加 SGTools 部分控件属性及功能 -4.3.10 更新字符串控件属性，“字体大小、字符间距-自动” -4.3.11 更新滚动字符串控件属性，“字体大小、字符间距-自动” -4.3.12 更新静态字符串控件属性，“字体大小、字符间距-自动” -4.3.13 更新数字控件属性，“字体大小、字符间距-自动” -4.3.14 更新计时器控件属性，“字符间距-自动” -4.3.15 更新日期时钟控件属性，“字体大小、字符间距-自动” -4.3.19 更新静态图标属性，“图标库、图标库 VP” -4.3.20 更新位变量图标属性，“图标库、图标库 VP” -4.3.21 更新变量图标属性，“图标库、图标库 VP” -4.3.23 更新十进位图标属性，“图标库、图标库 VP” -4.3.30 增加打开页面功能方式 -4.4.1 增加 PIP 中文键盘描述 -5.1 增加“带长度协议指令”和“带 CRC 协议指令” -5.2 变量读写增加 0x94、0x95 指令 -5.3.3 变量读写增加 0x94、0x95 指令例子 -6.14 表盘应用更新属性列表 -7.1 更新触摸键及虚拟键工具栏图标、快捷触摸键及键盘工具栏、控件工具栏 -7.2“页面, 图片资源和 VP 变量限定关系”增加图标库 更新页面/控件/资源限定关系 -7.3U 盘更新，增加 IMG 文件生成，跳转超链接“见 4.2.2 编译器设置”	Liaoliliang	2021-09-23
1.08	-增加第 1 章节“1.4 智能模块接口功能说明” -修改第 6 章节“应用案例”中英文案例用图为中文版本 -修改第 6 章节“应用案例”中参数描述部分，英文说明修改为中文 -修改部分说明	Huchubin	2022-03-11
1.09	-4.2.3 更新工程设置（菜单栏→工具→工程设置）的图片，修改工程默认基本参数：增加设置分区 2 大小（单位 MB）、U 盘更新模式、使能 ACK	Liwenming	2024-02-01

	-4.2.8 增加多语言设定（菜单→工具→StringTable） -4.3.1 增加触摸键属性，“整数位数、小数位数、字体、字体颜色、标题/值、最大值、最小值、输入长度、光标颜色、输入返回、键盘模式” -4.3.10 增加字符串控件属性，“模式、语言 ID VP” -4.3.11 增加滚动字符串控件属性，“模式、语言 ID VP” -4.3.13 增加数字控件属性，“小数位数 VP” -4.3.14 增加计时器控件属性，“字体大小” -4.3.17 增加虚拟键属性，“整数位数、小数位数、字体、字体颜色、标题/值、最大值、最小值、输入长度、光标颜色、输入返回、键盘模式” -4.3.19 增加静态图标属性，“模式” -4.4.3 增加呼叫-运算操作，“Bitn(VP)=LSB(Value)” -5.2 参数设定增加 0xEE 指令 -5.3.1 参数设定增加 0xEE 指令例子 -7.2 附录 B：工程设定增加系统变量资源限定关系/功能描述		
1.09a	-5.2 指令集增加了 0x96、0x97 指令 -5.3.1 参数设定增加了 0x96、0x97 指令 -6.2 附录 B 系统变量资源限定关系/功能描述：增加了 0xFFFF2E 地址(更改 U 盘模式) -增加了 modbus 脚本编辑器使用说明（4.5） -重新编排了章节顺序	Liuhuanqing	2024-08-13
1.10	-更新第 2 章节、第 3 章节、第 4 章、第 4 章节内容	Liuhuanqing	2025-04-09