



目 录

1. DGUS 配置软件简介	2
1.1 DGUS 配置软件 V3.4 功能特点	2
1.2 DGUS 配置软件操作指南	2
1.3 DGUS 配置软件主界面	4
2. 使用 DGUS 配置软件实现功能.....	7
2.1 配置触控按钮.....	8
2.1.1 弹出菜单.....	8
2.1.2 数据录入.....	9
2.1.3 增量调节.....	11
2.1.4 拖动调节.....	12
2.1.5 RTC 设置	13
2.1.6 基本触控.....	14
2.1.7 按键值返回.....	15
2.1.8 ASCII 录入	17
2.1.9 GBK 录入	18
2.1.10 参数配置.....	20
2.2 配置变量按钮.....	21
2.2.1 变量图标.....	21
2.2.2 动画图标.....	22
2.2.3 滑动刻度.....	23
2.2.4 艺术字变量.....	24
2.2.5 图片动画.....	25
2.2.6 图标旋转.....	26
2.2.7 数据变量.....	27
2.2.8 文本显示.....	28
2.2.9 RTC 显示	29
2.2.10 时钟显示.....	30
2.2.11 曲线显示.....	31
2.2.12 列表显示.....	33
2.2.13 基本图形.....	35
2.2.14 行业应用.....	37
2.2.15 位变量图标显示.....	38
2.2.16 时间变量显示.....	40
3. 指令的用法.....	41
3.1 数据帧架构.....	41
3.2 指令集.....	41
3.3 寄存器空间说明.....	42




1. DGUS 配置软件简介

1.1 DGUS 配置软件 V3.4 功能特点

- 1.另存为功能，能自动转换图片格式和大小，使另存后可以直接使用；
- 2.图标的预览功能，能方便用户进行精确定位。在预览中可直接看出设置的位置对不对，从而直接修改位置；
- 3.变量查看及按钮命名功能，可使用户进行应用设计时一目了然，点击地址时可进行排序，方便用户检测地址的连续性；
- 4.能把以前的 13 触控、14 变量配置文件直接导入，提高用户应用的延续性；
- 5.分辨率变化时，设置的按钮可自动变化，这样可使一套应用方案可直接应用于不同大小的屏上。

1.2 DGUS 配置软件操作指南

1. 在添加按钮配置时，根据不同操作习惯，在 Config 文件夹 Terminal.ini 中[init] 下方增加 Rkey=1 或者 Rkey=0:

Rkey=1 时，可实现在按钮配置时，在没有按右键或  按钮时，可进行连续添加；
Rkey=0 时，添加一个按钮配置后，自动取消添加功能。

2. 90 度图片设置方式：

如用户在应用时需要竖显应用，用户在 Terminal.ini 文件中：把 Rx 后的分辨率（如红色部分）替换一个不常用的分辨率，如 800X480 进行竖显配置。

[Resolution]

R2=480X800

R3=800X480

在设计时选择480X800分辨率进行设计即可。配置完成后，需要把工程进行另存处理，另存目录下的DWIN_SET文件夹下的图片可直接使用。

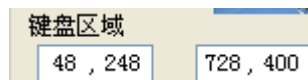
3. 在需要处理图标设置时，**请先把图标库拷贝到工程路径下的DWIN_SET目录下**，在设计时用户就可直接选择图标文件进行设置了。

4. 设置显示位置时操作方法：

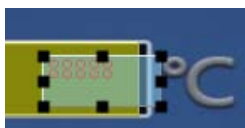
A、可以通过输入坐标进行定位，



或者



B、可以通过鼠标移动或光标键进行位置调整。



5. SP地址：

因为SP地址是一个比较特殊的存在，它记录了变量配置的设置，所以尽量避免地址不要交叉重复；软件自带地址重合判断，如果设置的地址与已有的地址区域重合，点击“生成配置文件”时，会有错误提示信息，而不能生成配置文件。



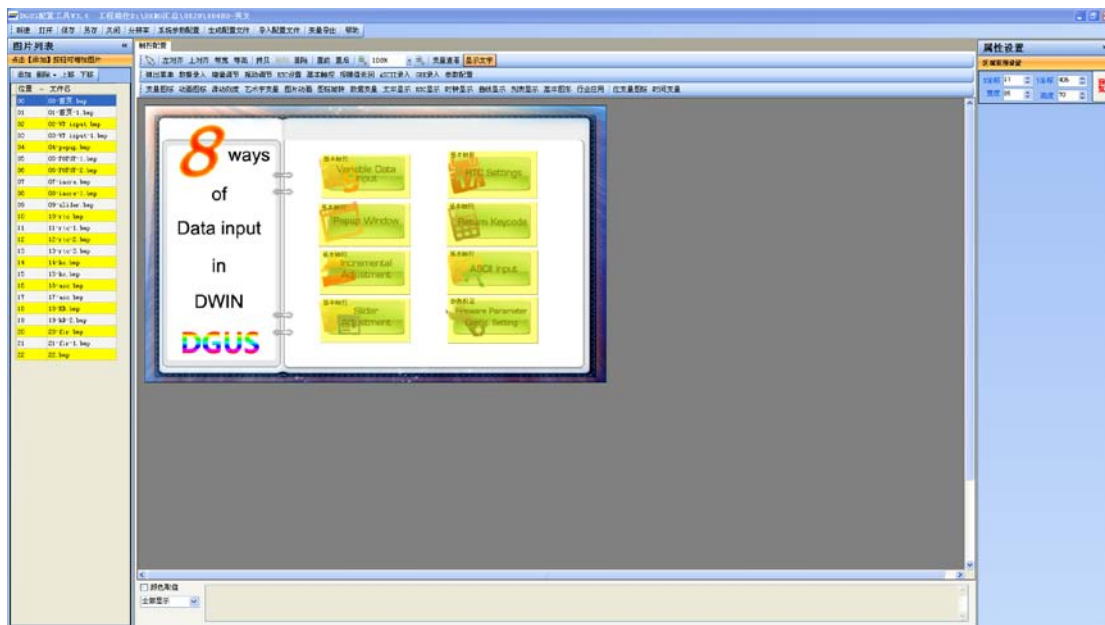
6. 怎么把以前的 DGUS 配置导入到新系统中:

- A. 新建一个工程;
- B. 增加老工程图片;
- C. 导入配置文件;
- D. 如有图标文件, 将图标 x. ICO 拷贝到新工程的 DWIN_SET 目录下。

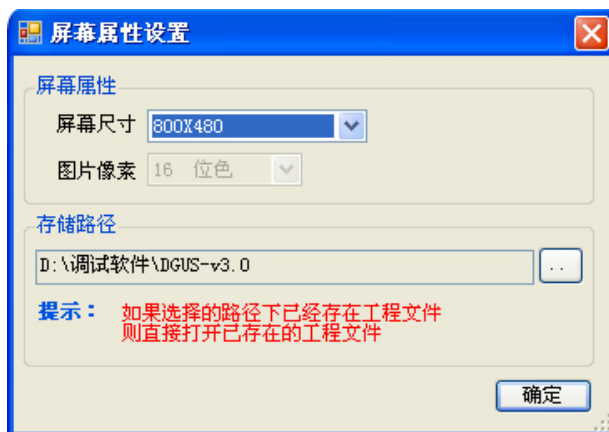
注: 指令帧头用户可自定义, 本文档中示例指令均使用 **A55A** 作为帧头。

北京迪文科技有限公司

1.3 DGUS 配置软件主界面



点击“新建”按钮弹出如下图：



选择存储路径、设置位色和分辨率：

属性设置：

如：DMT80480T070_06WT，

位色 T：16 位，

分辨率 80480：800x480。

按钮说明:

新建: 新建一个工程文件;

打开: 选择已存在的工程文件 (工程后缀名是.hmi);

保存: 保存当前工程 (包括配置文件) 到软件上方显示的: “工程路径……” 中;

另存: 将目前工程改存到其它位置 (包括配置文件);

关闭: 关闭当前工程, 软件不关闭;

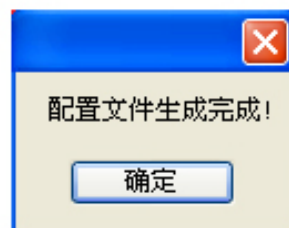
分辨率: 操作过程中可以随时修改分辨率;

系统参数配置: 如下图, 配置 CONFIG.txt 文件:



CONFIG.txt 配置文件中, **TPSAUTO** 为全局设置, 当勾选 “触摸屏录入参数自动上传到串口” 时, 在触控按钮中勾选 “数据自动上传”, 录入的数据才会自动上传至串口, 否则参数不会自动上传。

生成配置文件: 生成 “13 触控配置文件.bin” 和 “14 变量配置文件.bin” (工程目录下);



导入配置文件: 导入已有的配置文件 (如: DWIN_SET 文件夹 (包括 13 触控配置文件.bin、14 变量配置文件.bin, 13 触控配置文件.txt, 14 变量配置文件.txt));

变量导出: 生成 “触控配置.xls” “显示配置.xls” 便于按钮配置;

左对齐: 选中区域后点击 “左对齐”, 软件左上角显示 “选择对齐方式的基准按钮”, 选择被选区域内基准元素, 区域内左对齐;

上对齐: 同上;

等宽：同上；

等高：同上；

拷贝：选中一个或多个元素复制；

粘贴：将复制元素黏贴到相应位置；

删除：选中一个或多个进行删除；

置前：将选中元素放于最上端；

置后：将选中元素放于最下端；

变量查看：查看已制作完成的触控与变量配置（含变量地址 VP 和描述指针地址 SP）；

显示文字：选择是否显示按钮的文本名称。

页面参数设置

当前页面：【0】 00基本图形演示副本.bmp

变量地址查看

页面	名称	变量地址	变量类型	SP地址
0	选型表	0x1000	列表显示	0x5000
0	GBK录入	0x0000	GBK录入	
0	RTC显示	无变量设置	RTC显示	0xFFFF
0	RTC显示	无变量设置	RTC显示	0xFFFF
0	时间变量	0x1111	时钟显示	0xFFFF
0	位变量图标	0x3322	时钟显示	0xFFFF
0	行业应用	0x1212	行业应用	0xFFFF
0	基本图形	0x2000	基本图形	0xFFFF
0	曲线显示	无变量定义	曲线显示	0xFFFF
0	时钟显示	无变量定义	时钟显示	0xFFFF
0	文本显示	0x2211	文本显示	0x2000
0	数据变量	0x0999	数据变量显示	0xFFFF
0	图标旋转	0x00F0	图标旋转指示	0xFFFF
0	图片动画	无变量设置	动画图片	0xFFFF
0	艺术字变量	0x2233	艺术字显示	0x2020
0	滑动刻度	0x5544	滑块刻度指示	0xFFFF
0	动画图标	0x0321	动画图标显示	0xFFFF
0	变量图标	0x3333	变量图标	0xFFFF
0	参数配置	0x0000	硬件参数配置	



2. 使用 DGUS 配置软件实现功能

触控配置：弹出菜单、数据录入、增量调节、拖动调节、RTC 设置、基本触控、按键值返回、ASCII 录入、GBK 录入、参数配置（制作触控配置按钮时，为黄色区域显示）。

显示配置：变量图标、动画图标、滑动刻度、艺术字变量、图片动画、图标旋转、数据变量、文本显示、RTC 显示、时钟显示、曲线显示、列表显示、基本图形、行业应用、位变量图标、时间变量（制作显示配置区域时，为浅蓝色区域显示）。

参数配置：在 DGUS 配置软件中属性框前带有“0x”标识，必须用 16 进制填写，未带“0x”标识的用十进制填写。

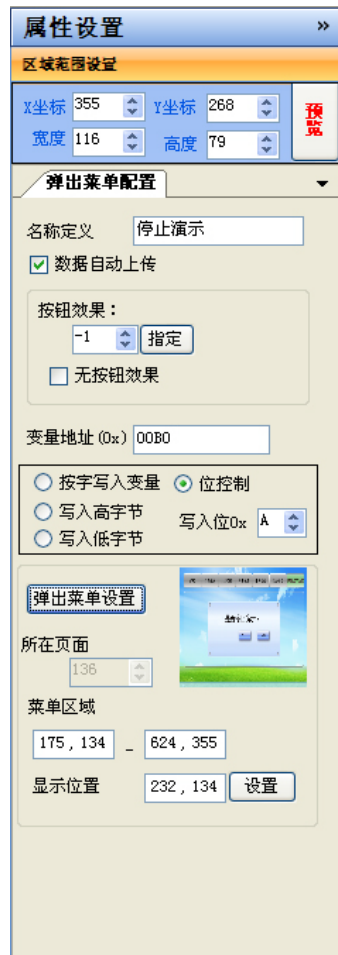
动画效果：直观显示按压效果。

注：

变量地址 (VP)：定义数据存储地址，即输入或显示内容的存储地址；范围 0x0000-0x6FFF；

描述指针 (SP)：描述文件的存储地址（描述文件，即：各个功能的配置说明），变量描述文件从 Flash 加载后存储到数据存储区的地址，0xFFFF 表示由配置文件加载，不转到数据存储区；范围 0x0000-0x6FFF。

2.1 配置触控按钮



2.1.1 弹出菜单

区域范围设置: 设置触控按钮区域;

预览: 查看触控按钮效果;

名称定义: 为按钮设置一个名称, 在**变量查看**中方便查询;

数据自动上传: 勾选时, 数据自动上传至串口;

按钮效果: 按钮按压效果图所在页面 (-1 默认无动画);

变量地址: 定义数据存储地址;

变量类型: 0x00=键码写入 VP 字地址;

0x01=键码写入 VP 地址的高字节 (VP_H);

0x02=键码写入 VP 地址的低字节 (VP_L);

0x10-0x1F=键码最低位 (1bit) 变量并写入 VP 字地址的指定位 (0x10 修改 VP.0, 0x1F 修改 VP.F);

弹出菜单设置: 设置弹出菜单所在页面及菜单区域;

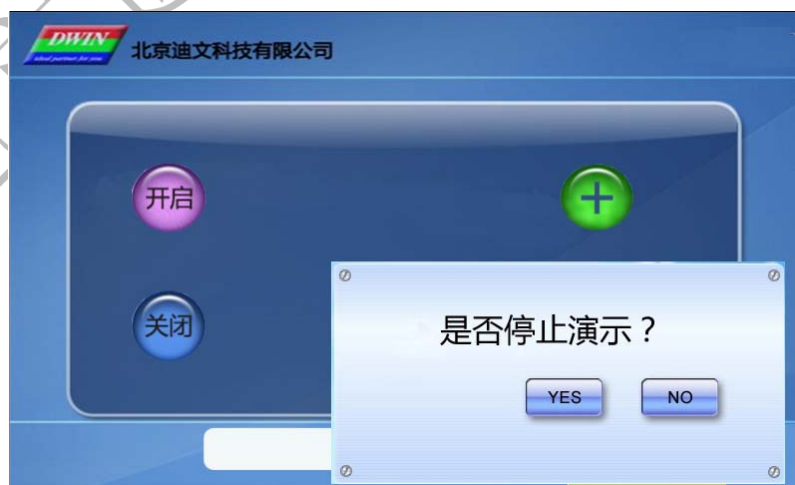
所在页面: 所要弹出菜单所在的页面;

菜单区域: 所要弹出菜单在页面的区域;

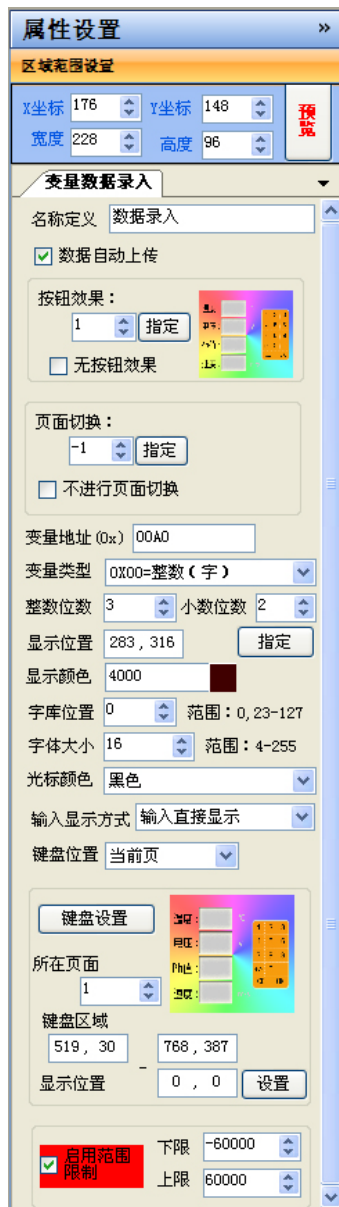
显示位置: 弹出的菜单在当前页面显示的位置。

注: 弹出菜单上只能做基本触控和按键值返回。

将左图配置完成后, 即可实现如下图功能。



点击关闭按钮→弹出菜单→YSE/NO



2.1.2 数据录入

区域范围设置： 设置触控按钮区域；

预览： 查看触控按钮的效果；

名称定义： 为按钮设置一个名称，在**变量查看**中方便查询；

数据自动上传： 勾选时，数据自动上传至串口；

页面切换： 指定到切换目标图片；

按钮效果： 按钮按压效果图所在页面（-1 默认无动画）；

变量地址： 定义数据存储地址；

变量类型： 0x00=整数（字）；

0x01=长整数（双字）；

0x02=无符号字节参数（变量地址高字节）；

0x03=无符号字节参数（变量地址低字节）；

0x04=超长整数（8 字节）；

整数位数： 录入数据整数位数；

小数位数： 录入数据小数位数；

显示位置： 输入过程中数据显示的位置；

显示颜色： 输入过程中字体显示的颜色，可以手动填写；

字库位置： 显示的 ASCII 字库位置，“0”为 0 号字库；

字体大小： X 方向的点阵数目；

光标颜色： 黑色/白色；

输入显示方式： 直接显示/显示*号；

键盘位置： 其他页/当前页（**切换的目标页面**）；

键盘设置： 设置键盘所在页面及键盘区域；

所在页面： 选择键盘所在页面；

键盘区域： 键盘所在页键盘区域；

显示位置： 键盘在当前页的位置（可以拾取）；

启用范围限制： 勾选后规定输入数字的上下限（越界不能录入）。

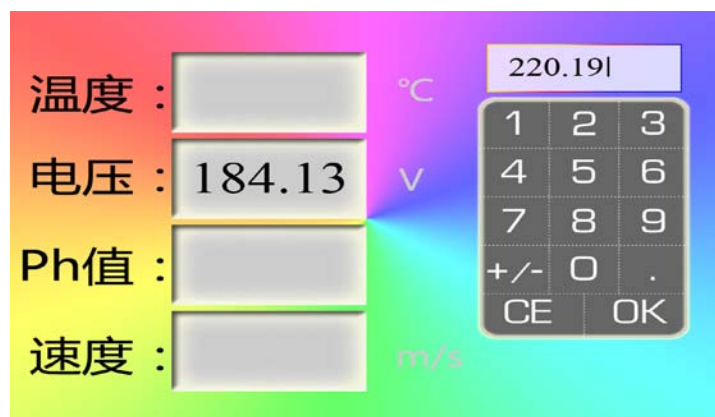
注： 变量录入调用的键盘要做基本触控，如：

0-9 对应返回值为 0x0030-0x0039、0x00F1（确定）、0x00F0（取消）、0x00F2（退格）、0x002D（+/-）、0x002E（.）

范围限制的限值是整数位加上小数位的取值范围，例如设了 3 个整数位，2 个小数位，那么取值的限值上限就为 10000，而不是 100。

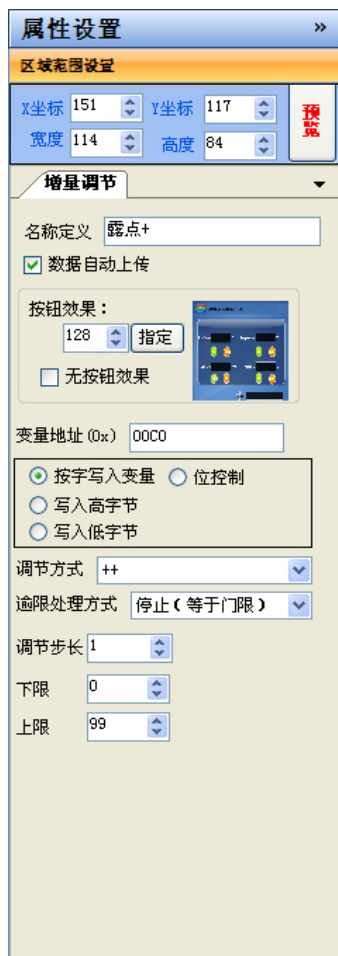
录入数据后通过“数据变量、艺术字变量”等将数据显示出来。

将上图配置完成后，即可实现如下图功能。



↑

点击文本框（凹下效果）→弹出键盘→录入数据(显示录入过程，光标跟随)→确认；
录入完成，如果在录入过程中点击清除/取消终止录入过程，键盘消失。



2.1.3 增量调节

区域范围设置: 设置触控按钮区域;

预览: 查看变量按钮的效果;

名称定义: 为按钮设置一个名称, 在**变量查看**中方便查询;

数据自动上传: 勾选时, 数据自动上传至串口;

按钮效果: 按钮按压效果图所在页面 (-1 默认无动画);

变量地址: 定义数据存储地址;

变量类型: 0x00=按字写入 VP 字地址 (整数型);

0x01=写入 VP 字地址的高字节 (VP_H);

0x02=写入 VP 字地址的低字节 (VP_L);

0x10-0x1F=对 VP 字地址的指定位 (0x10 对应 VP.0, 0x1F 对应 VP.F) 进行调节, 调节范围必须设置为 0-1;

调节方式: ++/-- (加或者减);

超限处理方式: 循环调节/停止 (等于门限);

调节步长: 增加或者减少的步长 (范围 0-32767);

下限: 增量调节的最小值 (范围 0-65535);

上限: 增量调节的最大值 (范围 0-65535)。

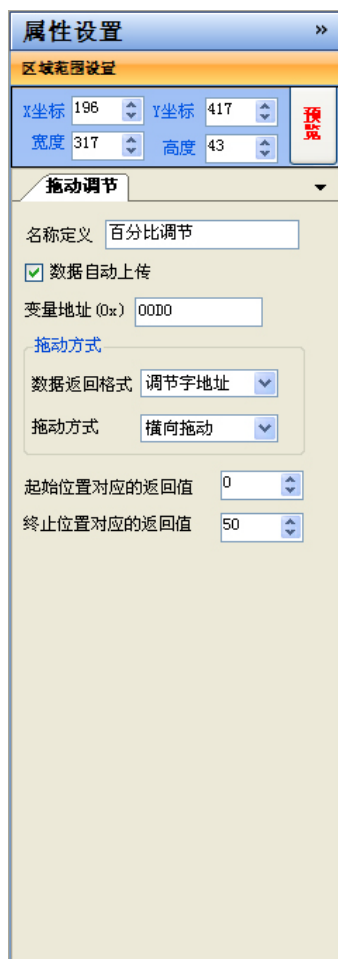
调节变量后可通过“数据变量、图标、艺术字”等将数据显示出来。

将左图配置完成后, 即可实现如下图功能。



↑
点击+/-→数据框内数据+/-;

持续按下 1 秒数据将按照设定步长持续+/-。



2.1.4 拖动调节

区域范围设置： 设置拖动调节区域；

预览： 查看触控按钮的效果；

名称定义： 为按钮设置一个名称，在**变量查看**中方便查询；

数据自动上传： 勾选时，数据自动上传至串口；

变量地址： 定义数据存储地址；

数据返回格式： 调节字地址；

调节高字节；

调节低字节；

拖动方式： 横向/纵向拖动；

起始位置对应的返回值： 拖动调节对应数值的最小值；

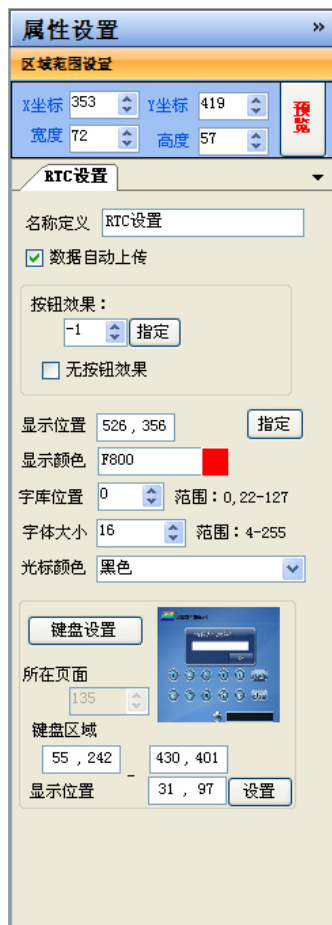
终止位置对应的返回值： 拖动调节对应数值的最大值。

滑块使用“滑块刻度指示”来显示；
调节滑块数据可通过“数据变量、变量图标”等
将数据显示出来。

将左图配置完成后，即可实现如下图功能；



↑
 点击滑块（0.5S 以上）拖动，右侧数据框数值跟随改变。



2.1.5 RTC 设置

区域范围设置： 设置触控按钮区域；

预览： 查看触控按钮的效果；

名称定义： 为按钮设置一个名称，在**变量查看**中方便查询；

数据自动上传： 勾选时，数据自动上传至串口；

按钮效果： 按钮按压效果图所在页面（-1 默认无动画）；

显示位置： 录入过程中字符串显示位置；

显示颜色： 录入过程中字符串显示颜色；

字库位置： 显示使用的 ASCII 字库位置，默认为 0 号字库；

字体大小： 字体 X 方向显示的点阵数目（4-128）；

光标颜色： 黑色/白色；

键盘设置： 设置键盘所在页面及键盘区域；

所在页面： 选择键盘所在页面；

键盘区域： 键盘所在页键盘区域；

显示位置： 键盘在当前页的位置（可以拾取）。

注： 键盘制作同变量录入的键盘；

时间显示由变量“RTC 显示、时钟显示”来完成。

将左图配置完成后，即可实现如下图功能。

下图所示

点击“RTC 设置”触控区域→弹出键盘→修改时钟





2.1.6 基本触控

区域范围设置： 设置触控按钮区域；

预览： 查看触控按钮的效果；

名称定义： 为按钮设置一个名称，在**变量查看**中方便查询；

数据自动上传： 勾选时，数据自动上传至串口；

按钮效果： 按钮按压效果图所在页面（-1 默认无动画）；

页面切换： 指定到切换目标图片；

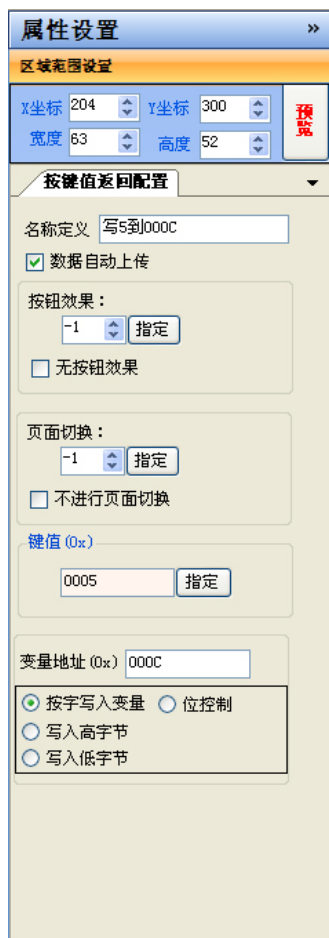
键值：

所有键盘按键都做基本触控，

（1）、ASCII、GBK 文本录入时做键盘时可以选择文本录入键盘选择键盘键值；

（2）、变量录入、RTC 设置调用键盘有效键值码：0x0030-0x0039（0-9），0x002E（.），0x002D（+/-），0x00F0（取消），0x00F1（确认），0x00F2（退格）。





2.1.7 按键值返回

区域范围设置： 设置触控按钮区域；

预览： 查看触控按钮的效果；

名称定义： 为按钮设置一个名称，在**变量查看**中方便查询；

数据自动上传： 勾选时，数据自动上传至串口；

按钮效果： 按钮按压效果图所在页面（-1 默认无动画）；

页面切换： 指定到切换目标图片；

键值： 根据需要自定义键值；

变量地址： 定义数据存储地址，将键值保存到该地址；

返回值保存方式： 0x00=按字写入 VP 地址；

0x01=写入 VP 地址的高字节（VP_H）；

0x02=写入 VP 地址的低字节（VP_L）；

0x10-0x1F=把返回键值的最低位（1bit）

写入 VP 字地址的指定位（0x10 修改 VP.0, 0x1F 修改 VP.F）。

将左图配置完成后，即可实现如下图功能。

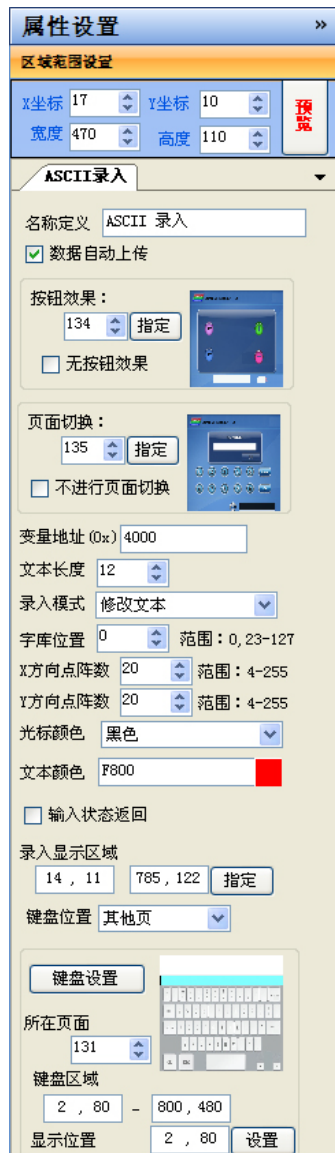


如上图所示，在开始按钮上做“按键返回值”，

定义：地址 000C、返回值 0005

返回格式为：A5 5A 06 83 00 0C 01 00 05 如下图





2.1.8 ASCII 录入

区域范围设置: 设置触控按钮区域;

预览: 查看触控按钮的效果;

名称定义: 为按钮设置一个名称, 在**变量查看**中方便查询;

数据自动上传: 勾选时, 数据自动上传至串口;

按钮效果: 按钮按压效果图所在页面 (-1 默认无动画);

页面切换: 指定到切换目标图片;

变量地址: 定义数据存储地址;

文本长度: 显示文本的长度 (以字为单位, 范围 1-123);

录入模式: 修改文本/重新录入;

字库位置: 显示的 ASCII 字库位置, 默认为 0 号字库;

X 方向点阵数: 字体 X 方向显示的点阵数目;

Y 方向点阵数: 字体 Y 方向显示的点阵数目 (0 号字库时, Y 方向点阵数目必须为 2X);

光标颜色: 黑色/白色;

文本颜色: 文本录入过程中字体颜色;

输入状态返回: 勾选时, 在 (VP-1) 位置保存输入结束标记和有效数据长度;

录入显示区域: 文本录入过程中显示区域;

键盘位置: 其他页/当前页 (**切换的目标页面**);

键盘设置: 设置键盘所在页面及区域;

所在页面: 选择键盘所在页面;

键盘区域: 键盘所在页键盘区域;

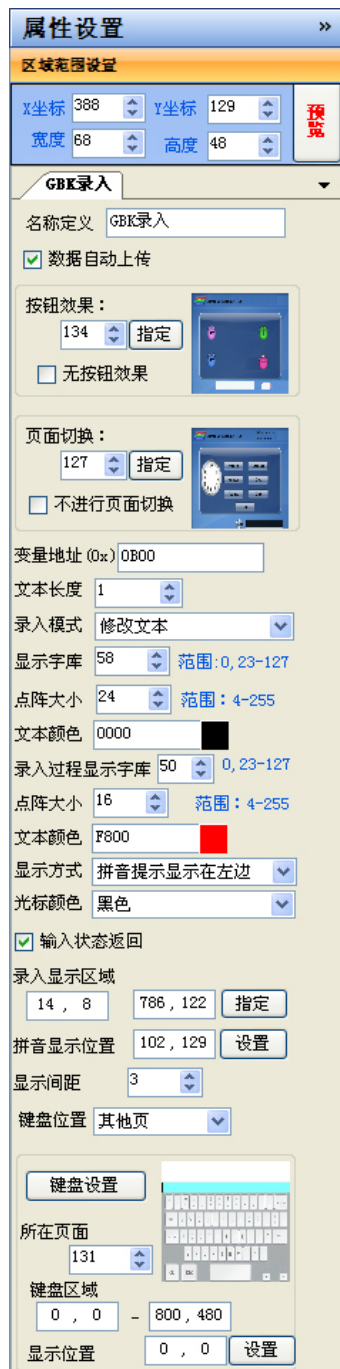
显示位置: 键盘在当前页的位置 (可以拾取)。

注: 键盘的有效键值码可以通过基本触控“文本录入键盘”选择键值 (如 A 键值为 0x4161)。

录入数据可由“文本显示”功能显示到屏上。

将上图配置完成后, 即可实现如下图功能。





2.1.9 GBK 录入

区域范围设置： 设置触控按钮区域；

预览： 查看触控按钮的效果；

名称定义： 为按钮设置一个名称，在**变量查看**中方便查询；

数据自动上传： 勾选时，数据自动上传至串口；

按钮效果： 按钮按压效果图所在页面（-1 默认无动画）；

页面切换： 指定到切换目标图片；

变量地址： 定义数据存储地址；

文本长度： 显示文本的长度（以字为单位，范围 1-123）；

录入模式： 修改文本/重新录入；

显示字库： 汉字的 GBK 字库存储位置（需下载字库到 DGUS 屏内，字库编号与字库位置一致，默认为 0 号 ASCII 字库）；

点阵大小： 与所选字库字体大小一致；

文本颜色： 确定文本后文本的颜色；

录入过程显示：

显示字库：调用的 GBK 字库位置；

点阵大小：与所选字库大小一致；

文本颜色：文本显示的颜色；

显示方式：拼音提示显示在汉字左边/上边；

光标颜色：黑色/白色；

输入状态返回： 勾选时，在（VP-1）位置保存输入结束标记和有效数据长度；

录入显示区域： 文本录入选取文本显示区域；

拼音显示位置： 拼音录入过程中的左上角坐标；

显示间距： 录入过程中每个汉字之间的距离（范围 1-255）；

键盘位置： 其他页/当前页（**切换的目标页面**）；

键盘设置： 设置键盘所在页面及区域；

所在页面： 选择键盘所在页面；

键盘区域： 键盘所在页键盘区域；

显示位置： 键盘在当前页的位置（可以拾取）。

GBK 文本录入：支持汉字录入（汉字录入需要下载 DWIN 拼音输入法）和 ASCII 录入；

注：键盘的有效键值码可以通过基本触控“文本录入键盘”选择键值（如 A 键值为 0x4161）。

录入数据由“文本显示”功能显示到屏上。

将上图配置完成后，即可实现如下图功能。

北京迪文科技



注:

拼音“bd”对应所有GBK 编码的全角标点符号录入;

迪文预装的 0#字库包含4*8-64*128 点阵的所有ASCII 字符;

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

属性设置

区域范围设置

X坐标 353 Y坐标 339
宽度 174 高度 68

参数配置

名称定义 位图打印

☐ 数据自动上传

按钮效果：
-1 指定

☐ 无按钮效果

页面切换：
-1 指定

☐ 不进行页面切换

Mode 5 0x00-0x07

DataPack 00190019025800C81000

2.1.10 参数配置

区域范围设置： 设置按钮区域；

预览： 查看按钮的效果；

名称定义： 为按钮设置一个名称，在**变量查看**中方便查询；

数据自动上传： 勾选时，数据自动上传至串口；

按钮效果： 按钮按压效果图所在页面（-1 默认无动画）；

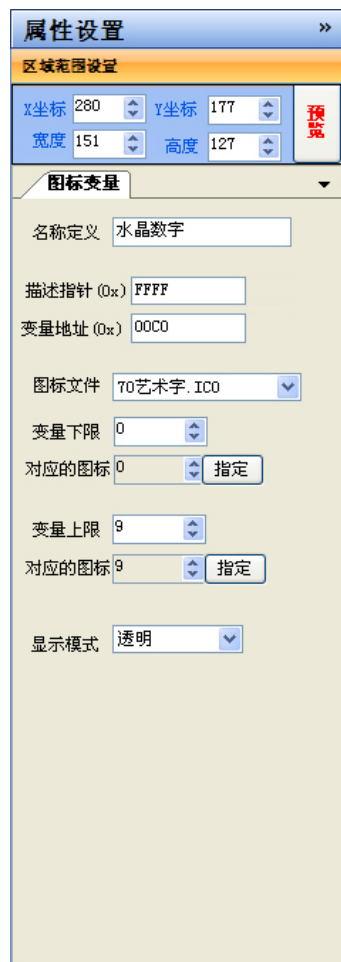
页面切换： 指定切换目标图片；

Mode： 操作模式设置，见下表。

操作模式表：

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

2.2 配置变量按钮



2.2.1 变量图标

区域范围设置: (X, Y)为 ICON 图标左上角显示在当前页位置坐标;

名称定义: 为按钮设置一个名称, 在**变量查看**中方便查询;

描述指针: 定义描述文件数据存储器的地址, 默认为 0xFFFF, 由配置文件加载 (一般不用修改);

变量地址: 变量数据存储的地址;

图标文件: 要调用的图标库 x.ICO 文件;

变量上限、变量下限: 规定变量显示的范围, 越界不显示;

对应的图标: 变量上下限对应的图标;

显示模式: 透明显示(滤除背景色显示)/显示背景。

将左图配置完成后, 即可实现如下图功能。

可以通过修改变量指针内的数据, 实现不同图标的切换, 如右图显示 0、9 对应的艺术字图标的 0 和 9。



属性设置

区域范围设置

X坐标: 257 Y坐标: 192
宽度: 42 高度: 33

动画显示设置

名称定义: 火焰文字动画
描述指针 (0x): FFFF
变量地址 (0x): 0B00
停止值: 0
开始值: 9
图标文件: 23图标艺术字. ICO
停止图标ID: 0 指定
开始图标ID: 1 指定
结束图标ID: 9 指定
显示模式: 透明

2.2.2 动画图标

区域范围设置: (X, Y)为 ICON 图标左上角显示在当前页位置坐标;

名称定义: 为按钮设置一个名称, 在**变量查看**中方便查询;

描述指针: 定义描述文件数据存储器的地址, 默认为 0xFFFF, 由配置文件加载 (一般不用修改);

变量地址: 变量数据存储的地址;

停止值: 变量为该值时显示固定图标;

开始值: 变量为该值时自动显示动画图标;

图标文件: 要调用的图标库 x.ICO 文件;

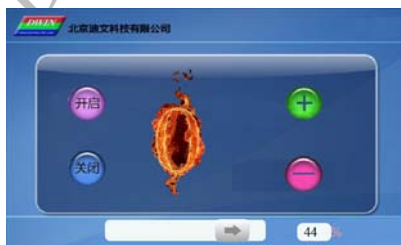
停止图标 ID: 变量地址数据为“停止值”时, 对应的图标;

开始、结束图标 ID: 变量地址数据为“开始值”时, 图标动画循环的范围;

显示模式: 透明显示(滤除背景色显示)/显示背景。

将左图配置完成后, 即可实现如下图功能。

当 0B00 地址变量值为 0 时: 显示固定图标 0;



当 0B00 地址变量值为 9 时: 1-9 对应的图标循环动画。

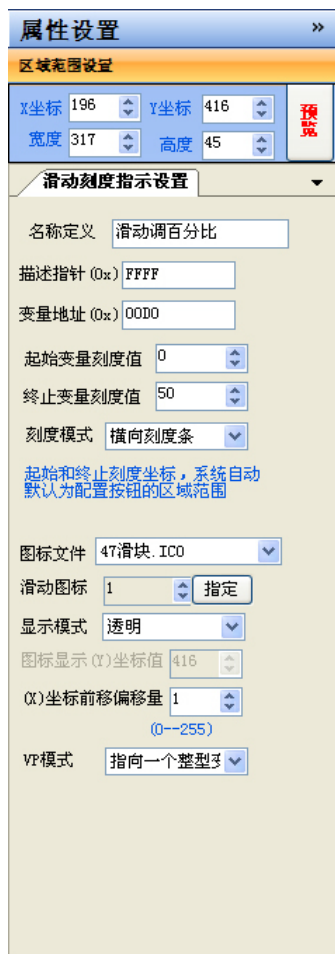


.....



.....





2.2.3 滑动刻度

区域范围设置: (X, Y)为 ICON 图标左上角显示在当前页位置坐标; **X 坐标**、**宽度**为横向滑动起始与终止坐标, **Y 坐标**、**高度**为纵向滑动起始与终止坐标;

名称定义: 为按钮设置一个名称, 在**变量查看**中方便查询;

描述指针: 定义描述文件数据存储器的地址, 默认为 0xFFFF, 由配置文件加载 (一般不用修改);

变量地址: 变量数据的存储地址;

起始、终止变量刻度值: 起始与终止刻度的变量值;

刻度模式: 横向刻度条/纵向刻度条;

图标文件: 要调用的图标库 x.ICO 文件;

滑动图标: 要调用的滑块图标 ID;

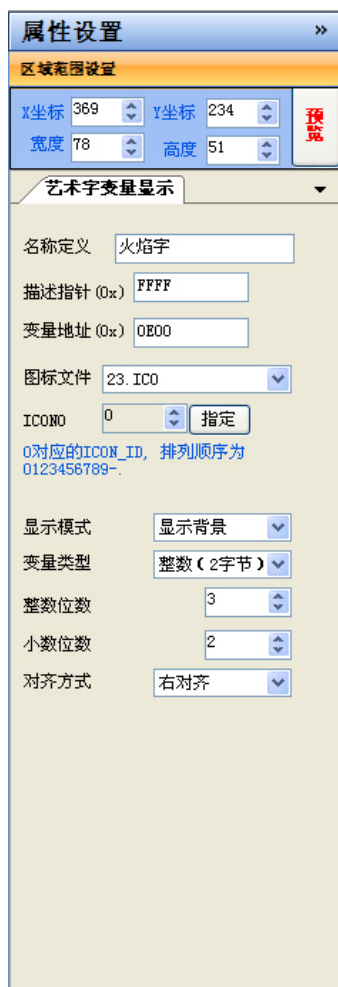
显示模式: 透明显示/显示背景;

X 坐标前移偏移量: 刻度图标显示的 X 坐标前移偏移量 (纵向为 Y, 范围 0-255);

VP 模式: 指向一个整型变量;
指向一个整型变量的高字节;
指向一个整型变量的低字节。

注: 滑动刻度和拖动调节配合使用 (地址要一致), 定义刻度范围应和拖动调节的取值范围对应。

另: 滑动刻度指示可用作进度条显示。



2.2.4 艺术字变量

区域范围设置: (X, Y)为显示字符串左上角显示在当前页位置坐标;

名称定义: 为按钮设置一个名称, 在**变量查看**中方便查询;

描述指针: 定义描述文件数据存储器的地址, 默认为 0xFFFF, 由配置文件加载 (一般不用修改);

变量地址: 变量数据存储的地址;

图标文件: 要调用的图标库 x.ICO 文件;

ICON0: 0 对应的 ICON_ID, 排列顺序为 0123456789-..;

显示模式: 透明显示/显示背景;

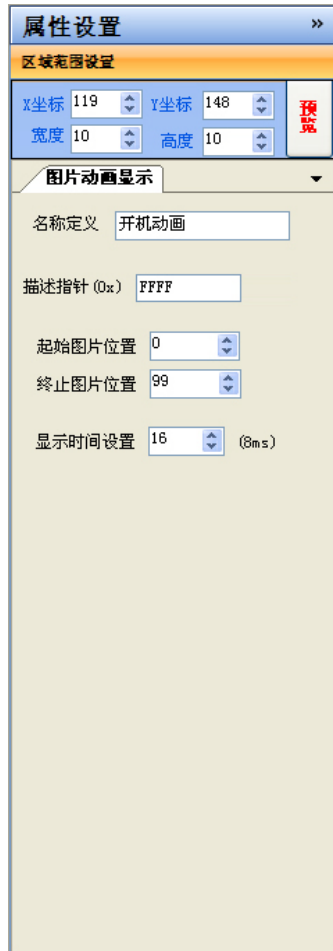
变量模式: 整数 (2 字节) /长整数 (4 字节);

整数位数、小数位数: 显示整数和小数的位数;

对齐方式: 左对齐/右对齐。

将左图配置完成后, 即可实现如下图功能。





2.2.5 图片动画

名称定义: 为按钮设置一个名称, 在**变量查看**中方便查询;

描述指针: 定义描述文件数据存储器的地址, 默认为 0xFFFF, 由配置文件加载 (一般不用修改);

起始图片位置: 动画开始的图片;

终止图片位置: 动画结束的图片;

显示时间设置: 一帧 (一幅图片) 显示的时间, 单位是 8ms。

起始图片位置必须小于终止图片位置。

在终止图片上也设置图片动画变量, 可以实现图片不间断重播, 实现动画效果。

发送串口指令或触控切换图片可以结束动画。

属性设置

区域范围设置

X坐标: 384 Y坐标: 239
宽度: 52 高度: 83

图标旋转指示

名称定义: 图标旋转

描述指针 (0x): FFFF

变量地址 (0x): 00F0

图标文件: 43.ICO

图标ID: 1 指定

图标旋转中心 X: 71 Y: 39

屏幕旋转中心, 系统自动默认为配置按钮的左上坐标

起始旋转角度变量值: 0

终止旋转角度变量值: 260

起始旋转角度: 0

终止旋转角度: 260

旋转角度, 0-720 (0x000-0x2D0), 单位0.5°

显示模式: 透明

VP_Mode: 指向整型变量

2.2.6 图标旋转

区域范围设置: (X, Y)为当前页面的旋转中心坐标;

名称定义: 为按钮设置一个名称, 在**变量查看**中方便查询;

描述指针: 定义描述文件数据存储器的地址, 默认为

0xFFFF, 由配置文件加载 (一般不用修改);

变量地址: 变量数据存储的地址;

图标文件: 要调用的图标库 x.ICO 文件;

图标 ID: 要调用的图标在 ICON 图标库中的存储位置;

图标旋转中心: (X, Y) ICON 图标上的旋转中心坐标;

起始、终止旋转角度变量值: 对应起始、终止旋转角度的变量值, 整数型, 越界不显示;

起始、终止旋转角度: 对起始、终止旋转角度, 变化单位 0.5°, 范围 (0-720);

显示模式: 透明显示/显示背景。

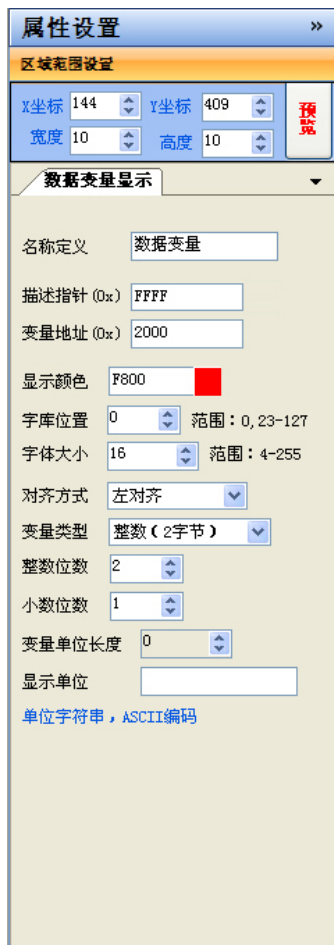
VP_Mode: 指向整型变量/
指向整型变量的高字节/
指向整型变量的低字节。

将左图配置完成后, 即可实现如下图功能。

通过串口发送指令或通过关联修改变量数据,
表盘指针转动相应角度。

调用的指针:





2.2.7 数据变量

区域范围设置: (X, Y)为显示字符串左上角显示在当前页面的位置坐标;

名称定义: 为按钮设置一个名称, 在**变量查看**中方便查询;

描述指针: 定义描述文件数据存储器的地址, 默认为 0xFFFF, 由配置文件加载 (一般不用修改);

变量地址: 变量数据存储的地址;

显示颜色: 字体显示颜色;

字库位置: 显示的 ASCII 字库位置, 默认为 0 号字库;

字体大小: 字体 X 方向显示的点阵数目 (范围 4-128);

对齐方式: 左对齐/右对齐/居中;

变量类型: 整数/长整数/VP 高字节/VP 低字节/超长整数/无符号整数/无符号长整数;

整数位数: 显示的整数位数;

小数位数: 显示的小数位数;

变量单位长度: 默认值 (跟随单位内容自动更改);

显示单位: 默认无单位, 设置单位只能为 ASCII 编码。

通过串口发送指令或通过关联修改变量数据, 修改显示数据和文本。

将左图配置完成后, 即可实现如下图功能。



属性设置

区域范围设置

X坐标: 63 Y坐标: 113
宽度: 312 高度: 73

文本显示

名称定义: 文本显示

描述指针 (0x): FFFF

变量地址 (0x): 0B00

显示颜色: F800

编码方式: 0x01=GB2312内码
☐ 字符间距不自动调整

文本长度: 20

Font0_ID: 0

编码方式为0x01-0x04时ASCII字库位置

Font1_ID: 50

编码方式为0x00, 0x05, 以及0x01-0x04的非ASCII字符使用的字库

X方向点阵数: 16 范围: 4-255

Y方向点阵数: 16 范围: 4-255

水平间隔: 1

垂直间隔: 2

2.2.8 文本显示

区域范围设置: (X, Y)为显示字符串左上角显示在当前页面的位置坐标; 整个区域为文本框;

名称定义: 为按钮设置一个名称, 在**变量查看**中方便查询;

描述指针: 定义描述文件数据存储器的地址, 默认为 0xFFFF, 由配置文件加载 (一般不用修改);

变量地址: 文本变量数据存储的地址;

显示颜色: 字体显示颜色;

编码方式: 0x00=8bit 编码, 0x01=GB2312 内码, 0x02=GBK, 0x03=BIG5, 0x05=UNICODE;

字符间距不自动调整: 字符间距自动调整 开/关;

文本长度: 本文显示数据的长度 (范围 0-65535);

Font0_ID: 编码方式为 0x01-0x04 时 ASCII 字库位置;

Font1_ID: 编码方式为 0x00, 0x05, 以及 0x01-0x04 的非 ASCII 字符使用的字库;

X、Y 方向点阵数: 显示字体大小 (与调用字库点阵一致);

水平、垂直间隔: 字符间水平与垂直距离。

Font1_ID 所对应的就是字库在屏中的位置。(字库存储在 23-127)

将左图配置完成后, 即可实现如下图功能 (左下角)。



属性设置

区域范围设置

X坐标 445 Y坐标 439
宽度 306 高度 37

RTC显示

名称定义 时间显示

描述指针 (0x) FFFF

字体颜色 F800

字库位置 0

X方向点阵数 8 范围: 4-255

日期格式: Y-M-D H:Q:S W

编码字符串: 使用RTC编码表和ASCII字符构成

RTC编码表:

说明	编码
公历_年	Y
公历_月	M
公历_日	D
公历_小时	H
公历_分钟	Q
公历_秒	S
公历_星期	W

假设当前时间是:
2012-05-02 12:00:00 星期三
Y-M-D H:Q:S 将显示为
2012-05-02 12:00:00
M-D W H:Q 将显示为
05-02 WED 12:00

2.2.9 RTC 显示

区域范围设置: (X,Y)为时间字符串左上角显示在当前页面位置左上角坐标;

名称定义: 为按钮设置一个名称, 在**变量查看**中方便查询;

描述指针: 定义描述文件数据存储器的地址, 默认为 0xFFFF, 由配置文件加载 (一般不用修改);

字体颜色: 字体显示的颜色;

字库位置: 显示的 ASCII 字库位置, 默认为 0 号字库;

X 方向点阵数: 字体 X 方向显示的点阵数目 (4-128);

日期格式: 见左图红色字体区域 (最长 16 个字符)。

上电时钟自动运行, 通过“**RTC 设置**”可以修改当前时钟。

将左图配置完成后, 即可实现如下图功能 (右下角)。



属性设置

区域范围设置

X坐标: 159 Y坐标: 235
宽度: 10 高度: 10

表盘时钟设置

名称定义: 表盘时钟显示
描述指针 (0x): FFFF
时钟表盘的指针中心位置为坐标的左上角

图标文件: 41指针.ICO

☐ 不显示时钟
时针图标: 2 指定
时针中心坐标: 14, 58

☐ 不显示分钟
分针图标: 1 指定
分针中心坐标: 12, 71

☐ 不显示秒钟
秒针图标: 3 指定
秒针中心坐标: 17, 79

2.2.10 时钟显示

区域范围设置: (X,Y)为时钟表盘显示在当前页面位置的指针中心;

名称定义: 为按钮设置一个名称, 在**变量查看**中方便查询;

描述指针: 定义描述文件数据存储器的地址, 默认为 0xFFFF, 由配置文件加载 (一般不用修改);

图标文件: 要调用的图标库 x.ICO 文件;

时钟:

时针图标: 要调用的时针图标在 ICON 图标库中的存储位置;

时针中心坐标: 时针图标旋转中心坐标 (可拾取);

分钟:

分针图标: 要调用的分针图标在 ICON 图标库中的存储位置;

分针中心坐标: 分针图标旋转中心坐标 (可拾取);

秒钟:

秒针图标: 要调用的秒针图标在 ICON 图标库中的存储位置;

秒针中心坐标: 秒针图标旋转中心坐标 (可拾取)。

上电时钟自动运行, 通过“**RTC 设置**”可以修改当前时钟。

将上图配置完成后, 即可实现如下图功能 (左面表盘时钟)。





2.2.11 曲线显示

区域范围设置: 曲线显示窗口，曲线越界将不显示；

名称定义: 为按钮设置一个名称，在**变量查看**中方便查询；

描述指针: 定义描述文件数据存储器的地址，默认为 0xFFFF，由配置文件加载（一般不用修改）；

Y_Central: 曲线中心轴位置；

VD_Central: 中心轴对应的曲线数据值，一般取最大数据和最小数据和的 50%；

曲线颜色: 曲线显示的颜色；

纵轴放大倍数: 单位是 1/256，(范围 0-32767)；

数据源通道: 0x00-0x07；

横轴间隔: x 轴间隔（范围 1-255）。

注：第 0—7 通道，CH 值分别是 0x01, 0x02, 0x04, 0x08, 0x10, 0x20, 0x40, 0x80；

另：多条曲线显示比如：0、1 通道，CH 值为 03。

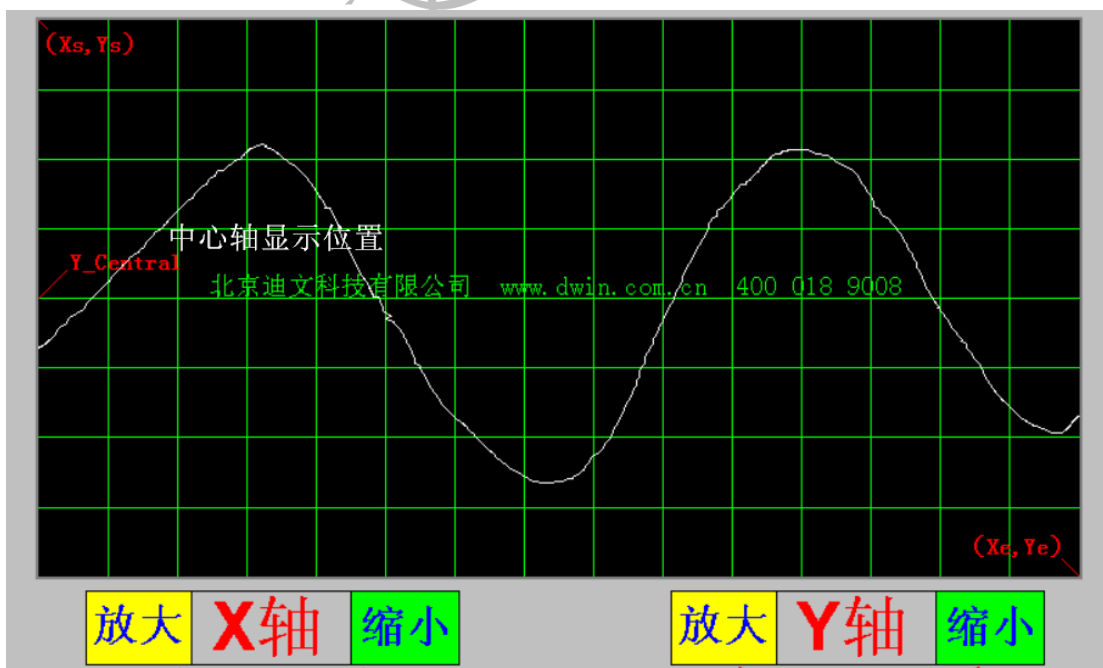
将左图配置完成后，即可实现如下图功能。

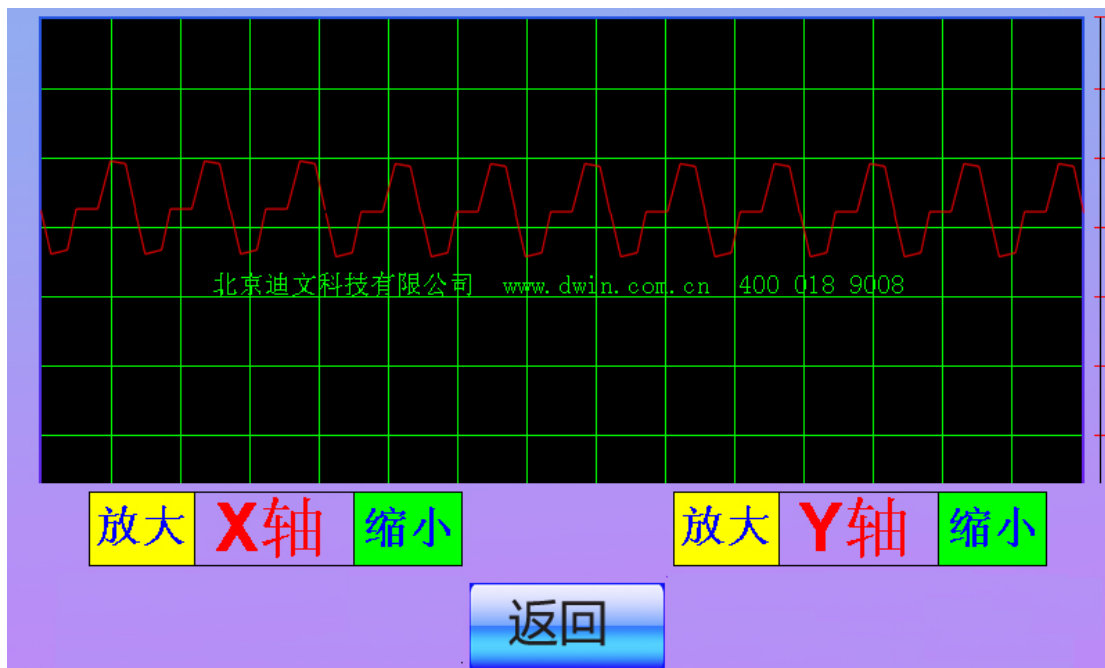
满量程曲线的纵轴放大倍数计算：

$MUL_Y = (Y_e - Y_s) * 256 / (V_{max} - V_{min})$ ，H 为曲线窗口的高度， V_{max} 、 V_{min} 为曲线数据的最大、最小值。

比如：一个 12bit A/D 采集数据 ($V_{max}=4095$, $V_{min}=0$) 要对应应在 $Y_s=50$, $Y_e=430$ 的屏幕区域满量程显示，那么：

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



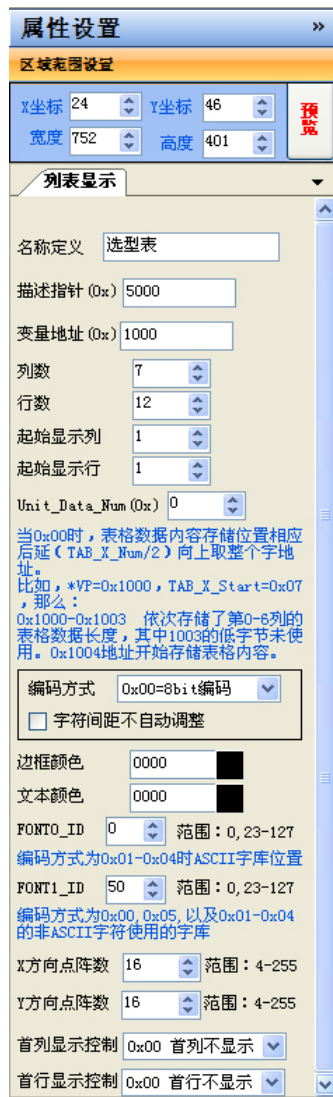


如上图所示，发送写曲线缓存区指令：

帧头	长度	指令	数据源通道	发送数据
A5	5A	18	84	01
				0800 0700 0710 0733 0800 0800 0800 0800 0900 0908 0906

就可以显示出入上图所示的曲线。

如果把变量描述内容存储在数据存储空间（描述指针指定存储位置），那么：
 结合增量触控指令，可以实现不需要用户代码干预的曲线的自动缩放；
 结合拖动触控指令修改曲线中心轴位置，可以实现不需要用户代码干预的曲线上下移动。



2.2.12 列表显示

区域范围设置: 表格显示区域, 越界将结束显示;

名称定义: 为按钮设置一个名称, 在**变量查看**中方便查询;

描述指针: 定义描述文件数据存储器的地址, 默认为

0xFFFF, 由配置文件加载 (一般不用修改);

变量地址: 表格内容地址 (**列数、行数数组开始的首地址**);

列数、行数: 表格显示的列数与行数;

起始显示列: 表格起始显示列位置;

起始显示行: 表格起始显示行位置;

Unit_Data_Num: 0x01-0x7F (所有存储数据长度相同);

0x00 由变量地址指向的变量存储空间定义了不同的不同列单元格的数据长度 (字长度);

当 Unit_Data_Num=0x00 时, 表格数据内容存储位置相应后延(列/2)向上取整个字地址。

例如变量指针为 0x1000, 起始显示列为 0x07, 那么: 0x1000-0x1003 依次存储了第 0-6 列的表格数据长度, 其中 0x1003 低字节未使用。0x1004 地址开始存储表格内容。

编码方式: 00=8bit 编码, 01=GB2312 内码, 02=GBK,

03=BIG5, 04=SJIS, 05=UNICODE;

字符间距不自动调整: 字符间距自动调整 开/关;

边框颜色: 表格边框颜色;

文本颜色: 表格文本显示颜色;

Font0_ID: 编码方式 0x01-0x04 时 ACSII 字库位置;

Font1_ID: 编码方式 0x00, 0x05, 以及 0x01-0x04 的非 ASCII 字符使用的字库;

X 方向点阵数: 字体 X 方向点阵数目 (**与调用字库点阵一致**);

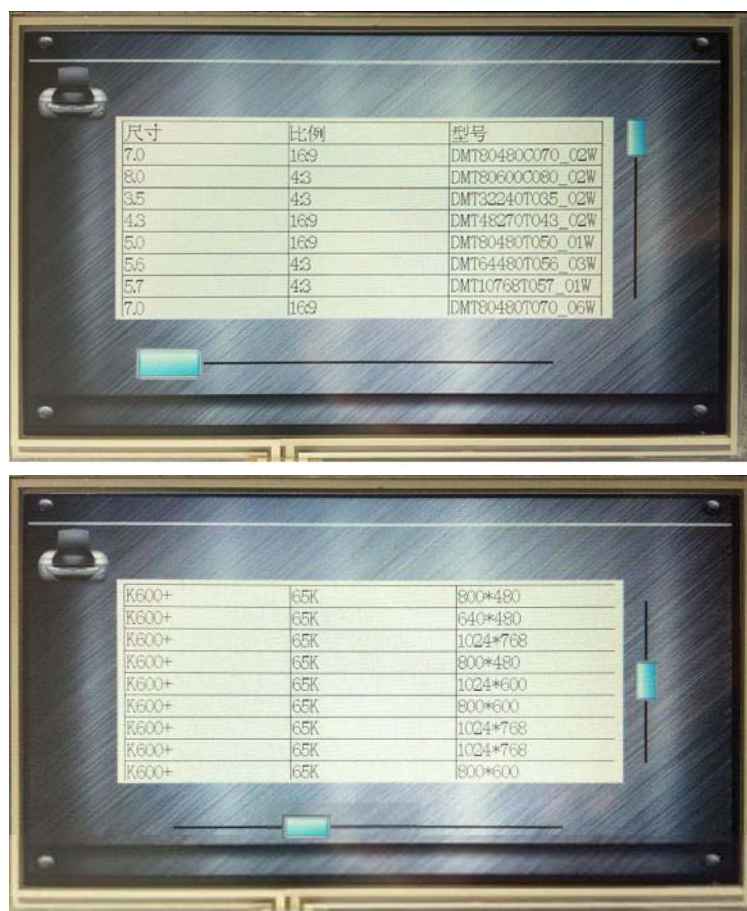
Y 方向点阵数: 字体 Y 方向点阵数目 (**与调用字库点阵一致**);

首列显示控制: 0x00 首列不显示/0x01 首列显示;

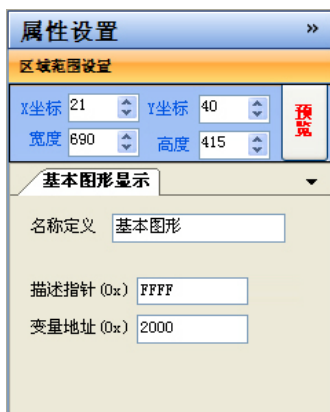
首行显示控制: 0x00 首行不显示/0x01 首行显示。

表格显示, 显示内容可由 22 变量初始化配置文件加载到屏上。

使用上面图片中的配置参数, 即可实现如下功能。



如果表格内容短于 Unit_Data_Num 规定的长度时,使用 0xFFFF 作为单元格文本结束符。
对于特别大的表格,通过触摸屏修改起始显示列、行值可以很方便地实现表格的定位和拖动。



2.2.13 基本图形

区域范围设置： 绘图显示区域，越界将不显示；（仅对 0x0001-0x0005 绘图指令有效）

名称定义： 为按钮设置一个名称，在**变量查看**中方便查询；

描述指针： 定义描述文件数据存储器的地址，默认为 0xFFFF，由配置文件加载（一般不用修改）；

变量地址： 定义数据存储的地址。

变量数据格式说明：

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

绘图指令数据包说明：

指 令 (CMD)	操 作	绘图数据包格式说明（相对地址和长度单位均为字（word））			
		相 对 地 址	长 度	定 义	说 明
0x0001	置点	0x00	2	(x, y)	置点坐标位置，x 坐标高字节为判断条件。
		0x02	1	Color	置点颜色
0x0002	连线	0x00	1	Color	线条颜色
		0x01	2	(x, y) 0	连线顶点 0 坐标，x 坐标高字节为判断条件。
		0x03	2	(x, y) 1	连线顶点 1 坐标，x 坐标高字节为判断条件。
		0x01+2*n	2	(x, y) n	连线顶点 n 坐标，x 坐标高字节为判断条件。
0x0003	矩形	0x00	2	(x, y) s	矩形框左上角坐标，x 坐标高字节为判断条件。
		0x02	2	(x, y) e	矩形框右下角坐标。
		0x04	1	Color	矩形颜色
0x0004	矩形域填充	0x00	2	(x, y) s	矩形域左上角坐标，x 坐标高字节为判断条件。
		0x02	2	(x, y) e	矩形域右下角坐标。
		0x04	1	Color	矩形域填充颜色
0x0005	圆	0x00	2	(x, y)	圆心坐标，x 坐标高字节为判断条件。
		0x02	1	Rad	半径。
		0x03	1	Color	圆颜色。
0x0006	图片区域剪切、粘贴	0x00	1	Pic_ID	剪切图片区域所在页面 ID；高字节为判断条件。
		0x01	2	(x, y) s	剪切图片区域左上角坐标。
		0x03	2	(x, y) e	剪切图片区域右下角坐标。
		0x05	2	(x, y)	剪切图片区域粘贴到当前页面的坐标位置，左上角坐标。
0x**07	ICON 图标显示	0x00	2	(x, y)	显示坐标位置，x 坐标高字节为判断条件。
		0x02	1	ICON_ID	图标 ID, 图标库位置和显示模式由指令高字节指定。图标固定为不显示背景色。
0x0008	区域填充	0x00	2	(x, y)	种子点坐标，x 坐标高字节为判断条件。
		0x02	1	COLOR	填充颜色。
0x0009	垂直线条	0x00	1	Color0	把 (X0, Y0s) (X0, Y0e) 用 Color0 颜色连线，X0 高字节为判断条件。
		0x01	1	X0	
		0x02	1	Y0s	
		0x03	1	Y0e	

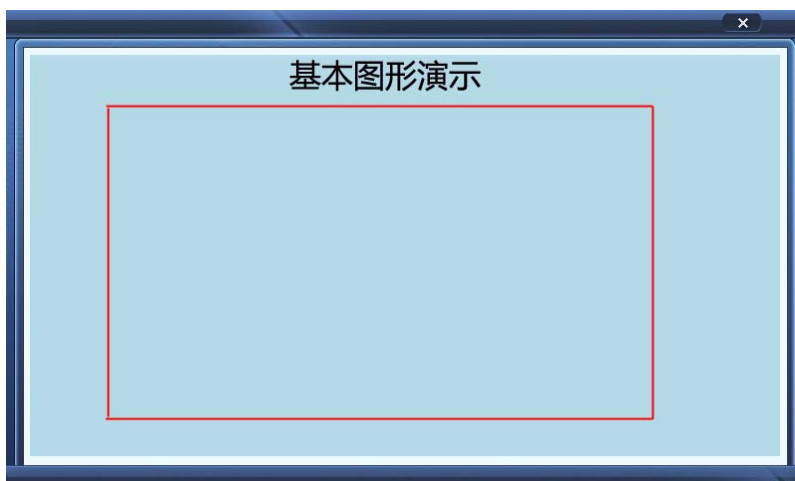


判断条件: 0xFF 绘图操作结束 0xFE 本次操作跳过 (忽略)。

举例:

如上表所示, 发送画矩形指令:

帧头	长度	指令	地址	控制字	数据包数目	左上角/右下角坐标	颜色
A5 5A	11	82	20 00	00 03	00 01	00 64 00 64 02 8C 01 90	F8 00



显示出上图所示矩形框, 也可以直接用 22 初始化文件显示。

属性设置

区域范围设置

X坐标: 201 Y坐标: 143
宽度: 71 高度: 49

特殊行业应用

名称定义: 塔吊应用

描述指针 (0x): FFFF

变量地址 (0x): 3000

2.2.14 行业应用

名称定义: 为按钮设置一个名称, 在**变量查看**中方便查询;

描述指针: 定义描述文件数据存储器的地址, 默认为

0xFFFF, 由配置文件加载 (一般不用修改);

变量地址: 定义数据存储地址。

变量数据格式说明

地 址	定 义	说 明
VP	CMD	指令
VP+1	Data_Pack_Num_Max	最大数据包数目; 遇到结束条件会自动终止。
VP+2	DATA_Pack	

指令数据包说明

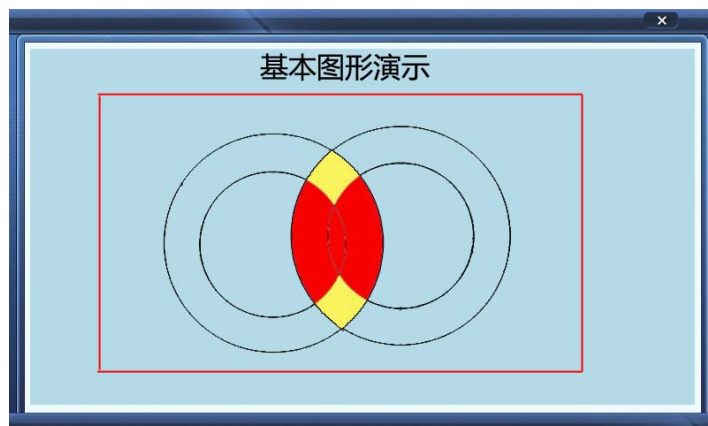
指 令 (CMD)	操 作	绘图数据包格式说明 (相对地址和长度单位均为字 (word))			
		相对地址	长度	定 义	说 明
0x0001	多个圆域交集填充	0x00	1	Color0	安全区域填充颜色。
		0x01	1	Color1	一般干涉区域 (一次相交) 填充颜色。
		0x02	1	Color2	严重干涉区域 (两次及以上相交) 填充颜色。
		0x03	1	Color3	圆域外廓线显示颜色
		0x04	1	Color4	回避颜色 (比如不填充的背景文本颜色或网格线)。
		0x05	4	Disp_Area	显示区域, 越界将不显示
		0x09+4*n	2	(x, y) n	第 n 个同心圆域圆心坐标, x 坐标高字节为 0xFF 表示无效。
		0x0B+4*n	1	RADn_1	第 n 个同心圆域的大圆半径。
		0x0C+4*n	1	RAD2n_2	第 n 个同心圆域的小圆半径。


举例:

如上表所示, 发送指令:

帧头 长度 指令 地址 控制字 数据包数 数据

A5 5A 29 82 3000 0001 0002 FFFF FFE0 F800 0000 FFFF 0064 0064 028C
0190 012C 00C8 0078 005A 01C2 00C0 0078 005A





属性设置

区域范围设置

X坐标: 40 Y坐标: 384
宽度: 70 高度: 30

位变量图标显示

名称定义: 位变量图标
描述指针 (0x): FFFF
变量地址 (0x): 0008
辅助地址 (0x): 4000
Act_Bit_Set: 1111111111111111

设置

显示模式: 0x00
移动模式: 0x02
移动间隔: 45
图标文件: 40.ICO

ICON0S: 0 指定
ICON0E: 0 指定
ICON1S: 1 指定
ICON1E: 1 指定
ICON Mode: 0x00=透明

2.2.15 位变量图标显示

区域范围设置: (X,Y)为起始位变量显示位置, 图标左上角坐标位置;

名称定义: 为按钮设置一个名称, 在**变量查看**中方便查询;

描述指针: 定义描述文件数据存储器的地址, 默认为 0xFFFF, 由配置文件加载 (一般不用修改);

变量地址: 位变量指针, 字变量;

辅助地址: 辅助变量指针, 双字; 不要与其他地址冲突;

Act_Bit_Set: 点击设置弹出“位控制设置”菜单, 如下图; 为 1 的 bit 位置说明变量地址对应位置需要显示;

显示模式: 0x01-0x07, 定义见下表;

移动模式: 位图图标排列方式,

0x00: X++, Act_Bit_Set 指定的不处理 bit 不保留位置;

0x01: Y++, Act_Bit_Set 指定的不处理 bit 不保留位置;

0x02: X++, Act_Bit_Set 指定的不处理 bit 保留移动间隔位置;

0x03: Y++, Act_Bit_Set 指定的不处理 bit 保留移动间隔位置;

移动间隔: 下一个图标坐标移动间隔;

图标文件: 要调用的图标库 x.ICO 文件;

ICON0S: 不显示动画模式, bit 0 图标 ID;

显示动画模式, bit 0 图标动画起始 ID 位置;

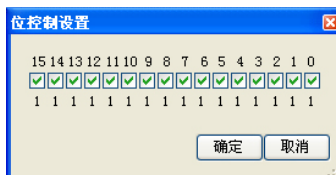
ICON0E: 显示动画模式, bit 0 图标动画结束 ID 位置;

ICON1S: 不显示动画模式, bit 1 图标 ID;

显示动画模式, bit 1 图标动画起始 ID 位置;

ICON1E: 显示动画模式, bit 1 图标动画结束 ID 位置;

ICON Mode: 透明显示/显示背景。



位控制设置

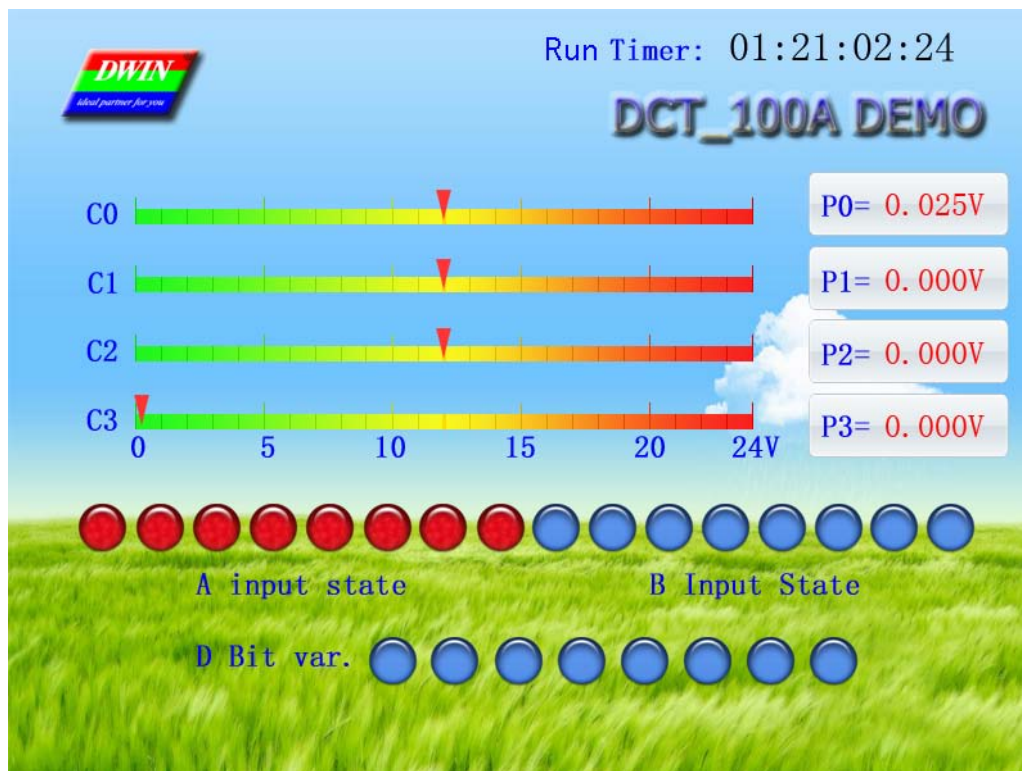
15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
☒ ☒ ☒ ☒ ☒ ☒ ☒ ☒ ☒ ☒ ☒ ☒ ☒ ☒ ☒
 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1

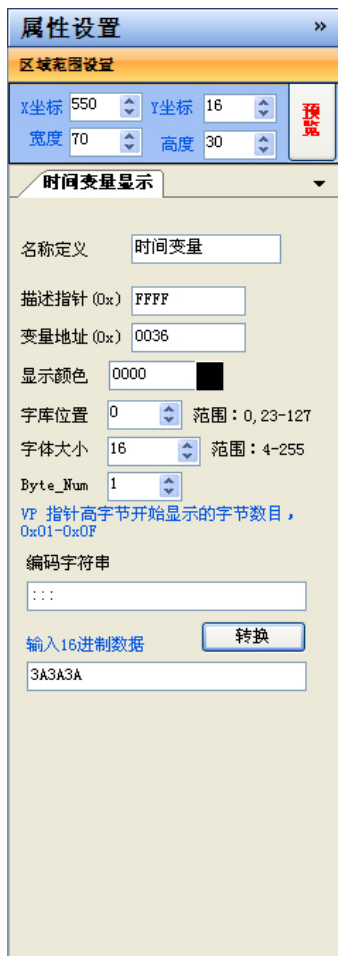
确定 取消

显示模式表

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

将上图配置完成后, 即可实现如下图功能 (页底图标)。





2.2.16 时间变量显示

区域范围设置: (X, Y)为字符串左上角显示在当前页面位置坐标;

名称定义: 为按钮设置一个名称, 在**变量查看**中方便查询;

描述指针: 定义描述文件数据存储器的地址, 默认为 0xFFFF, 由配置文件加载 (一般不用修改);

变量地址: 变量指针数据串首地址, 变量为 BCD 编码;

Byte_Num: VP 指针高字节开始显示的字节数目, 0x01-0x0F;

显示颜色: 字体显示的颜色;

字库位置: 调用的字库位置; 字库必须是半角方式; 如果字库位置不为 0, 字库必须使用 8bit 编码;

字体大小: 字体 X 方向点阵数目;

编码字符串: 直接输入 ASCII 字符, **字符转换工具**;

输入 16 进制数据: **点击转换**, 自动将编码字符串转换成 16 进制数据; 此项也可直接输入。

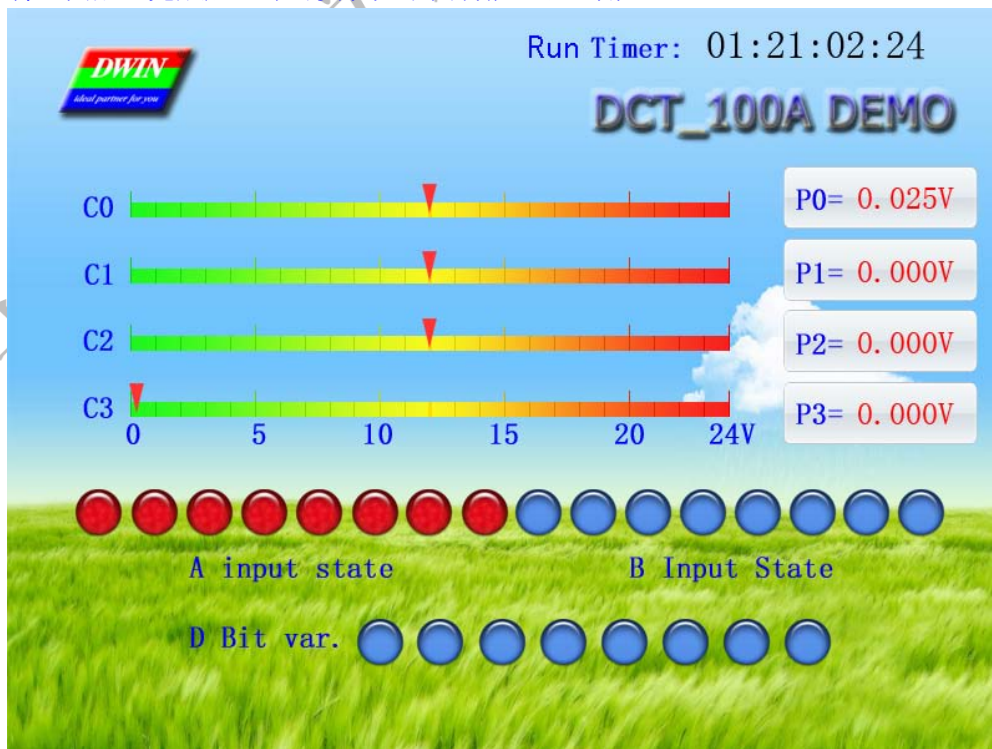
编码字符串, 用来和时间变量组合出客户需要的显示格式。每显示一个 BCD 码后, 会到编码字符串顺序取出一个 ASCII 字符来间隔显示;

特殊字符定义如下:

0x00: 无效, 本字符不显示, 两个 BCD 时间码将连在一起;

0x0D: 换行显示, 即 $X=X_s$, $Y=Y+Font_X*2$ 。

将左图配置完成后, 即可实现如下图功能 (右上角)。





3.指令的用法

3.1 数据帧架构

迪文 DGUS 的串口数据帧由 4 个数据块组成，如下表所述：

数据块	1	2	3	4	5
定 义	帧 头	数据长度	指令	数据	指令和数据的 CRC 校验
数据长度	2	1	1	N	2
说 明	CONFIG.TXT 配置文件的 R3:RA 定义。	数据长度包括指令、数据和校验	0x80-0x84		CONFIG.TXT 配置文件的 R2 决定是否启用

3.2 指令集

功能	指令	数据	说 明
访问控制寄存器	0x80	ADR(0x00-0xFF)+Data_Pack	指定地址写寄存器数据
	0x81	ADR(0x00-0xFF)+RD_LEN(0x00-0xFF)	指定地址读 RD_LEN 字节 寄存器数据
		ADR(0x00-0xFF)+RD_LEN+Data_Pack	读寄存器的迪文屏应答
访问数据存储器	0x82	ADR_H:L(0x0000-0x6FFF)+DATA0...DATA _n	指定地址开始写入数据串(字 数据)到变量存储区
		ADR_H:L(0x0000-0x6FFF)+RD_LEN(0x00-0x7F)	从变量存储区指定地址开始读入 RD_LEN 长度 字 数据
	0x83	ADR_H:L+RD_LEN+DATA0.....DATA _n	读数据存储器的迪文屏应答
写曲线缓冲区	0x84	CH_Mode(Byte)+DATA0(Word)+...+DATA _n	写曲线缓冲区数据。 CH_Mode 定义了后续数据的通道排列顺序： > CH_Mode 的每个位(bit)对应 1 个通道； > CH_Mode.0 对应 0 通道，.7 对应 7 通道； > 位置 1 表示对应的通道数据存在； > 数据按照低通道数据在前排列。 比如 CH_Mode=0x83 (10000011B)，表示后续数据格式为： (通道 0+通道 1+通道 7) +...+(通道 0+通道 1+通道 7)。

寄存器的访问以字节(Byte)为数据单位，而数据存储器、曲线缓冲区的访问以字(Word)为数据单位。关于存储器的定义和说明，请参考 3.3 存储器空间说明



3.3 寄存器空间说明

迪文 DGUS 提供 1 个 256B 寄存器接口，用于硬件操作或者图片显示等进程控制。

寄存器地址	定 义	字节长度	说 明	
0x00	Version	1	DGUS 版本号，BCD 码表示，0x10 表示 V1.0	
0x01	LED_NOW	1	LED 亮度控制寄存器，0x00-0x40	
0x02	BZ_TIME	1	蜂鸣器鸣叫控制寄存器，单位为 10mS	
0x03	PIC_ID	2	读：当前显示页面 ID； 写：切换到指定页面（进程控制）	
0x05	TP_Flag	1	0x5A=触摸屏坐标有更新； 其它=触摸屏坐标未更新 用户读取数据后未清零本标记，则触摸屏数据不再更新。	
0x06	TP_Status	1	0x01=第一次按下；0x03=一直按压中；0x02=抬起；其它=无效	
0x07	TP_Position	4	触摸屏按压坐标位置：X_H:L Y_H:L	
0x0B	TPC_Enable	1	0x00=触控不启用 其它=触控启用（上电默认为 0xFF）。	
0x0C-0x0F	保留	4	未定义	
0x10-0x1A	R0-RA	11	SD 卡配置寄存器的映射，串口只读，串口写无效。	
0x20	RTC_NOW	16	YY:MM:DD:WW:HH:MM:SS YY:MM:DD+天干地支+生肖	
串口修改 RTC 举例：A5 5A 0A 80 1F 5A 12 08 08 04 12 00 01 04 指星期四，实际可以写任何值				
0x30-0xFF	保留	16	未定义	
0x40	En_Lib_OP	1	0x5A 表示用户申请进行字库存储器操作，DGUS 操作完后清零	
0x41	Lib_OP_Mode	1	0x50：把指定的变量存储器空间数据写入指定字库空间 0xA0：把指定字库空间的数据读入变量存储器空间	
0x42	Lib_ID	1	指定的字库空间，0x40-0x7F，每个字库 128KW，对应最大 Flash 空间为 8MW (16MB)	
0x43	Lib_Address	3	指定字库空间的数据操作首(字)地址， 0x00:00:00-0x01:FF:FF	
0x46	VP	2	指定变量存储器空间的数据操作首(字)地址， 0x00:00-0x6F:FF	
0x48	OP_Length	2	数据操作的(字)长度，0x00:01-0x6F:FF	
把 0x1000 开始的 1KW 变量数据保存到 64#字库 0x0000 开始的位置，串口下发指令举例： A5 5A 0C 80 40 5A 50 40 00 00 00 10 00 02 00				
0x4A	Timer0	2	16bit 软件定时器，单位为 4ms，自减到 0 停止。	设置值和实际运行值之间有 +/-4ms 误差
0x4C	Timer1	1	8bit 软件定时器，单位为 4ms，自减到 0 停止。	
0x4D	Timer2	1	8bit 软件定时器，单位为 4ms，自减到 0 停止。	
0x4E	Timer3	1	8bit 软件定时器，单位为 4ms，自减到 0 停止。	
0x4F	Key_Code	1	用户键码，用于触发 0x13 触控文件；0x01-0xFF，0x00 表示无效。DGUS 处理键码后会自动清零键码寄存器。	
0x4A-0xFF	保留	182	未定义	

寄存器空间使用 0x80/0x81 串口指令访问。

➤ 变量存储器空间

迪文 DGUS 提供了 28KW（56KB）的 SRAM 做为用户变量存储器，用于 GUI 变量数据的存储。在读写实时性（mS 级别）要求不高时，可以作为用户的外扩串口访问存储器（RMA）。变量存储器空间只能按照字（Word）读写，对应地址空间为 0x0000-0x6FFF。

数据存储器空间使用 0x82/0x83 串口指令访问。

➤ 曲线数据缓冲区

为了简化实时曲线的显示，迪文 DGUS 设计有一个曲线数据缓冲区用来缓存用户的曲线数据，曲线缓冲区不占用数据存储器空间，最多可以同时缓冲 8 条曲线。

曲线数据缓冲区只能按照字（Word）写，每个曲线点的数据均用 2 字节无符号整数表示。

曲线数据缓冲区使用 0x84 串口指令访问（只写）。