

# 矿用本安型触摸屏 使用手册



# 目 次

声明 .....	XI
1 使用目录 .....	1
1.1 软件首页工具使用 .....	1
2 基本概念 .....	1
2.1 软件的坐标体系 .....	2
2.2 HMI 的运行原理 .....	2
2.3 自绘图形 .....	3
2.4 缓冲区 .....	3
2.5 向量图 .....	3
2.6 工程 .....	4
2.7 部件 .....	5
2.8 画面 .....	5
2.9 编译 .....	6
3 开始一个新工程 .....	6
3.1 建立新工程 .....	6
3.2 设计画面 .....	9
3.3 建立新画面 .....	10
3.4 添加文本说明 .....	11
3.5 添加一个位开关部件 .....	12
3.6 添加位状态指示灯 .....	12
3.7 位报警的使用 .....	13
3.8 改变部件的外观 .....	13
3.9 使用位图外观—选择位图 .....	15
3.10 设置部件类型和操作地址 .....	16
3.11 编译和调试 .....	17
4 工程与画面 .....	18
4.1 新建工程 .....	18
4.2 新建画面 .....	19
4.3 画面属性 .....	21
4.4 删除画面 .....	22
4.5 复制画面 .....	22
4.6 画面与子画面 .....	23
5 系统参数 .....	25
5.1 一般属性 .....	25
5.2 通讯参数 .....	26
5.3 分期付款模式 .....	27
5.4 HMI 以太网设置 .....	29
5.5 安全等级及部件等级密码 .....	30

6	位图与位图库 .....	33
6.1	位图库 .....	33
6.2	位图 .....	34
6.3	HMI 位图 .....	35
6.4	位图编辑 .....	35
6.5	添加位图 .....	36
6.6	使用 HMI 位图 .....	37
6.7	导入图库 .....	39
6.8	新建多状态 HMI 位图 .....	41
7	地址与地址库 .....	42
7.1	地址库 .....	42
7.2	多连线与多站号 .....	42
7.3	地址的使用 .....	43
7.4	地址编辑器 .....	43
7.5	地址标识库 .....	45
7.6	地址预览 .....	47
8	部件的复制 .....	48
8.1	部件的复制 .....	48
8.2	部件重叠 .....	48
9	文本库 .....	50
9.1	使用文本库 .....	50
9.2	当前语言 .....	51
9.3	文本库 .....	52
10	报警 .....	55
10.1	报警机制 .....	55
10.2	位报警区 .....	59
10.3	字报警区 .....	60
11	按钮开关 .....	62
11.1	按键 .....	62
11.2	字开关 .....	62
11.3	位开关 .....	64
11.4	滑动开关 .....	66
11.5	功能开关 .....	67
11.6	配方传输 .....	69
11.7	超级组合按钮 .....	70
12	数字输入/显示 .....	72
12.1	数字输入/显示 .....	72
12.2	文本输入/显示 .....	78
13	报警显示 .....	81
13.1	事件记录显示 .....	81
13.2	报警记录显示 .....	83
13.3	报警走马灯 .....	85

13.4	当前报警表 .....	87
13.5	报警历史表 .....	88
14	图标曲线 .....	89
14.1	历史圆盘记录图 .....	89
14.2	百分比趋势图 .....	89
14.3	数据记录显示 .....	91
14.4	圆盘记录图 .....	91
14.5	历史趋势图 .....	92
14.6	历史 XY 图 .....	93
14.7	XY 趋势图 .....	95
14.8	趋势图 .....	97
15	动画位图 .....	99
15.1	旋转位图 .....	99
15.2	轨迹动画 .....	99
15.3	流动部件 .....	101
15.4	移动图形 .....	102
15.5	移动图形缩放功能 .....	103
15.6	移动多边形 .....	106
15.7	视频输入显示部件 .....	106
16	仪表 .....	109
16.1	扇形图 .....	109
16.2	棒图 .....	109
16.3	饼图 .....	112
16.4	罐图 .....	113
16.5	仪表 .....	114
16.6	时钟 .....	117
17	指示灯 .....	119
17.1	位状态指示灯 .....	119
17.2	字状态显示 .....	119
17.3	四状态指示灯 .....	120
18	图形 .....	122
18.1	间接画面显示 .....	122
18.2	直接画面显示 .....	122
18.3	自定义部件 .....	123
18.4	下拉式清单 .....	125
18.5	广告走马灯 .....	126
18.6	打印部件 .....	126
18.7	日期显示 .....	127
18.8	时间显示 .....	127
18.9	文件列表 .....	127
18.10	配方显示 .....	131
18.11	列表框 .....	132
19	自会图形 .....	134



19.1	自绘图形 .....	134
19.2	静态向量图 .....	134
19.3	弧形刻度 .....	135
19.4	线性刻度 .....	137
19.5	圆/椭圆 .....	137
19.6	多边形 .....	138
19.7	位图 .....	139
19.8	矩形 .....	139
19.9	表格 .....	139
19.10	直线 .....	140
19.11	文本 .....	140
19.12	折线 .....	141
19.13	点 .....	141
19.14	弧 .....	141
20	配方 .....	144
20.1	概述 .....	144
20.2	举例说明 .....	144
20.3	配方信息编辑 .....	144
20.4	配方显示的添加 .....	146
20.5	配方传输的添加 .....	147
20.6	配方传输的使用 .....	149
20.7	配方数据的索引 .....	150
21	记录区信息 .....	151
21.1	WebSever 数据记录区 .....	151
21.2	实时圆盘记录图信息 .....	151
21.3	多机互联 (Multi-Link) 使用说明 .....	152
21.4	历史 XY 趋势图 .....	154
21.5	趋势图信息 .....	154
22	安全保护 .....	156
22.1	概念及用途 .....	156
22.2	画面安全等级模式 .....	156
22.3	按键密码保护模式 .....	158
23	脚本 .....	159
23.1	概述 .....	159
23.2	使用脚本应该注意的问题 .....	159
23.3	脚本对设备的访问 .....	160
23.4	第一次使用脚本 .....	161
23.5	初识脚本编辑器 .....	163
23.6	脚本的分类 .....	165
23.7	初始化脚本 .....	167
23.8	位触发脚本 .....	168
23.9	关闭脚本 .....	169
23.10	定时脚本 .....	169
23.11	全局函数 .....	170

23.12	语法检查 .....	172
23.13	语法错误 .....	172
23.14	系统函数 .....	176
23.14.1	A2H 函数.....	176
23.14.2	Abs 函数.....	177
23.14.3	ACos 函数.....	177
23.14.4	AddrStringCompare 函数.....	178
23.14.5	Asc 函数.....	179
23.14.6	AsFloating 函数.....	179
23.14.7	ASin 函数.....	180
23.14.8	AsInteger 函数.....	180
23.14.9	AsString 函数.....	181
23.14.10	ATan 函数.....	182
23.14.11	ATan2 函数.....	183
23.14.12	B2W 子程序.....	183
23.14.13	BCD 函数.....	185
23.14.14	Beep 功能.....	186
23.14.15	BIN 函数.....	187
23.14.16	BMOV 子程序.....	188
23.14.17	Chr 函数.....	189
23.14.18	ClrB 子程序.....	191
23.14.19	常量 .....	192
23.14.20	Cos 函数.....	192
23.14.21	D2F 子程序.....	193
23.14.22	D2Float 函数.....	194
23.14.23	<b>DegToRad 函数</b> .....	195
23.14.24	Dim .....	196
23.14.25	Do .....	197
23.14.26	End .....	198
23.14.27	W2B 子程序.....	198
23.14.28	F2D 子程序.....	199
23.14.29	F2S 函数.....	199
23.14.30	FILL 子程序.....	201
23.14.31	Float2D 函数.....	201
23.14.32	For .....	202
23.14.33	Function .....	203
23.14.34	Goto .....	204
23.14.35	H2A 函数.....	205
23.14.36	Hypot 函数.....	206
23.14.37	IF .....	207
23.14.38	Script 1.0 参考.....	210
23.14.39	InStr 函数.....	212
23.14.40	InvB 子程序.....	214
23.14.41	IsFloating 函数.....	215
23.14.42	IsInteger 函数.....	215
23.14.43	IsString 函数.....	216

23.14.44	LCase 函数.....	218
23.14.45	Left 函数.....	218
23.14.46	Len 函数.....	220
23.14.47	Log 函数.....	222
23.14.48	Log10 函数.....	223
23.14.49	LTrim 函数.....	224
23.14.50	MAX 函数.....	226
23.14.51	Mid 函数.....	227
23.14.52	MIN 函数.....	229
23.14.53	MSeconds 函数.....	230
23.14.54	NewNoAddr 函数.....	232
23.14.55	NewStatAddr 函数.....	233
23.14.56	NStringCompare 函数.....	234
23.14.57	操作符.....	235
23.14.58	power 函数.....	238
23.14.59	程序结构.....	238
23.14.60	RadToDeg 函数.....	238
23.14.61	RAND 函数.....	238
23.14.62	ReadAddr 函数.....	239
23.14.63	Right 函数.....	239
23.14.64	RTrim 函数.....	239
23.14.65	SetB sub 子程序.....	240
23.14.66	SignedInt16 函数.....	240
23.14.67	SignedInt32 函数.....	241
23.14.68	Sin 函数.....	241
23.14.69	SleepA 函数.....	241
23.14.70	Sqr 函数.....	242
23.14.71	Sub.....	242
23.14.72	SWAP 子程序.....	243
23.14.73	Tan 函数.....	244
23.14.74	Trim 函数.....	246
23.14.75	UCase 函数.....	246
23.14.76	变量.....	247
23.14.77	W2D 子程序.....	247
23.14.78	W2F 函数.....	248
23.14.79	W2S 函数.....	249
23.14.80	WHILE ... WEND.....	250
23.14.81	WriteAddr 函数.....	251
24	编译过程.....	253
24.1	编译过程.....	253
24.2	编译错误(警告).....	253
25	下载.....	255
25.1	工程下载/上传.....	255
25.2	配方上传/下载.....	256

26	模拟 .....	257
26.1	离线模拟 .....	257
26.2	在线模拟 .....	257
27	HMI 内部保留寄存器 .....	258
27.1	概述 .....	258
27.2	系统数据区 (HSW) .....	258
27.3	数据存储区 (HDW) .....	258
27.4	特殊存储区 (HPW) .....	259
27.5	配方索引区 (RPW) .....	259
27.6	保留寄存器 .....	259
28	串行通讯与 PLC 驱动程序 .....	272
28.1	RS-422 与 RS-485 串行接口标准 .....	272
28.2	RS-232 串行接口标准 .....	273
28.3	RS-422 电气规定 .....	274
28.4	RS-485 电气规定 .....	275
29	字体库 .....	277
29.1	自定义字体的使用 .....	278
29.2	添加字体 .....	279
30	数据记录 .....	280
30.1	通过读卡器导出数据 .....	280
30.2	数据记录功能 .....	281
30.3	数据记录显示 .....	281
30.4	记录数据打印 .....	282
30.5	数据记录区 .....	283
31	穿透 .....	287
31.1	应用方案 1 .....	287
31.2	应用方案 2 .....	287
32	远程穿透 .....	288
32.1	远程穿透系统架构 .....	288
32.2	远程穿透系统架构 .....	288
33	人机界面的常见使用问题 .....	290
34	高级高级用户自定义协议——用户界面 .....	294
34.1	高级用户自定义协议——使用步骤 .....	297
35	OpenCAN 使用手册 .....	299
36	1939 寄存器编辑说明 .....	305
37	AB PLC 地址编辑说明 .....	308
38	用户自定义协议使用说明 .....	309
39	开放内部地址设置以太网 IP .....	312
40	modbus All Fuction 驱动说明文档 .....	314
41	关于 Modbus RTU master 映射内部 HDX、HDW 地址 .....	315

42 初级教程 .....	316
42.1 软件帮助在哪里 .....	316
42.2 软件版本 .....	316
42.3 Usb 线的类型 (D 口、一口、迷你口) .....	316
42.4 232 的下载线接线: .....	318
42.5 串口引脚的定义: .....	318
42.6 各个型号的 com 口 (com1 和 com2、com3) .....	319
42.7 下载时的工具选择 (PC 端口) .....	321
42.8 测试界面的进入及内容 .....	322
42.9 u 盘根目录和 u 盘升级文件 .....	331
42.10 画面的复制: .....	332
42.11 画面属性的设置 .....	333
42.12 子画面的制作 .....	334
42.13 画面 (部件) 安全等级: .....	335
42.14 脚本的使用和分类: .....	337
42.15 部件叠放顺序的使用: .....	339
42.16 工程所有使用过的部件列表在哪里: .....	340
42.17 字位开关的使用: .....	341
42.18 流动部件使用: .....	344
42.19 多重复制 .....	344
42.20 键盘制作使用的部件 .....	346
42.21 向量图的修改 .....	351
42.22 外观位图的修改 .....	352
42.23 透明操作 .....	354
42.24 字体的编辑 .....	357
42.25 部件对齐排版 .....	359
42.26 批量修改相关属性: .....	359

# 1 使用目录

## 1.1 软件首页工具使用

打开软件，主画面会显示“欢迎使用”界面。在工程画面中，右侧有六个选项：工程反编译、工程上传/下载、工程模拟、帮助主题、软件设置、U盘升级。

**工程反编译**：从HMI下上传上来的hmt文件，需要进行反编译后才能用软件打开。

**HMT工程位置**：导入HMT文件，将它进行反编译，最后通过打开hmp文件来打开工程。

**保存路径**：反编译后的工程文件保存的路径。

**工程密码**：在原工程中的工程设置里的安全等级选项里设置的设计者密码。

**工程上传/下载**：不打开工程，直接通过这个工具来上传下载工程文件、配方或者镜像文件。

“PC端口”：通过下载工具(串口线或USB线)把工程文件、镜像文件上传或下载到HMI。

“波特率”：默认为115200，不需要修改。

“文件类型”：选择要下载的文件，类型包括工程文件(.hmt)、配方(.rcp)、镜像文件(.osf)。

“密码”：仅上传时需要。密码是工程的工程参数里的“安全等级及密码”选项中的“设计者密码”。

“取消检测更新”：勾选上，则在下载工程的同时，会去检测安装目录下的update文件夹里是否有新版本的镜像文件，有的话它会先更新镜像文件，然后再下载工程。

“HMI→PC”：即上传工程文件到电脑。

“PC→HMI”：即下载工程文件、配方、镜像文件到HMI屏上。

“更新HMI时间”：将屏的系统时间更新为电脑的系统时间。

**工程模拟**：浏览hmt文件，进行离线模拟或者在线模拟。

**软件设置**：**设置软件的部件属性的显示方式**：属性栏或者对话框。可以两者都勾选，则在工程编辑中，点击画面中部件，在左边会以属性栏的方式显示部件的所有属性，或者双击该部件，则跳出属性对话框。

**U盘升级**：将工程文件或者镜像文件复制到U盘中，通过工厂测试画面的UPDATE 。OSF和COPY PROJECT从U盘灌工程或者镜像文件到HMI。

进入工厂测试画面的方式是：按住屏的右上角三到四秒钟，触摸屏会进入黑白画面的工厂测试界面。

**U盘升级工程文件/U盘升级镜像文件**：选择要复制到U盘的文件，HMT工程文件或者osf镜像文件。

## 2 基本概念

在了解和开始使用之前，只需要知道简单的基本概念，其中很多概念都是在HMI应用中是很常见的。

### 2.1 软件的坐标体系

软件的坐标原点是触摸屏的左上角，其坐标值是以像素为单位的，在软件开发系统中，其坐标系与的坐标系是一致的。如下图所示：



图 2-1 软件的坐标系统

### 2.2 HMI 的运行原理

HMI 是 Human Machine Interface(人机界面)的简写，它为 PLC 控制系统、I/O 模块或者其他控制系统提供一个人性化的操作界面，而人机界面则通过 RS232、485、MODBUS、CAN 总线等协议与这些控制系统交互数据与控制信号。由于这种交互是实时的，这就使得 HMI 实现了下面的关键功能：

实时监控 PLC、I/O 模块的运行，并在 HMI 上形象地显示出来；

操作人员可以通过 HMI 向自动化设备发出控制信号，使得自动化设备可以按照操纵者的意图运行。

图 2-2 形象地说明了 HMI 这种典型应用。

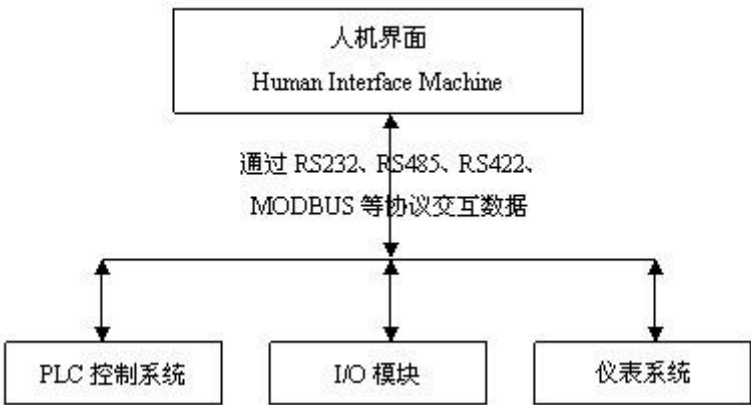


图 2-2 HMI 典型应用示意图

HMI 的应用领域决定了它必须具备下列特性：

1. HMI 必须具备和各种自控设备交互数据的能力，这需要 HMI 集成各种型号的 PLC、I/O 模块、各种总线协议等自控设备的通信协议；
2. HMI 应该提供一个工具软件，供客户开发基于特定控制系统定制相应的应用；
3. HMI 必须是工业级的产品，能够适应工业应用环境，具有可靠的质量和优秀的性能。

### 2.3 自绘图形

自绘图形是用户可以根据定制的图形，自绘图形不涉及任何 PLC 的数据交互，是静态显示的。它的存在是为了能让客户设计出更为美观和形象的画面。自绘图形主要包括矩形、弧、饼图、圆/椭圆、多义线、直线、静态文本等等。

### 2.4 缓冲区

缓冲区是由用户指定的，不依赖具体画面的监控区域，现在有下列缓冲区：

**位报警信息缓冲区：**录入与保存位报警的相关信息

**字报警信息缓冲区：**录入与保存字报警的相关信息

**数据记录区：**录入与保存数据记录功能的相关信息

**实时曲线缓冲区：**录入与保存趋势图的相关信息

**历史 XY 图缓冲区：**录入与保存历史 XY 图的相关信息

**WebServer 数据记录区：**录入与保存 Webserver 数据的相关信息

**多机互联数据区：**录入与保存多机互联数据的相关信息

缓冲区的运行是不依赖画面运行的，而是当工程在模拟器上或者在 HMI 启动后，系统就根据缓冲区的设置开始监视数据，并把监控到的数据保存在 (模拟器/HMI) 上一段预先开辟的缓存里。当缓存超出用户设定时，系统将按照先进先出的原则进行溢出处理。

监控所得到的数据保存在模拟器/HMI 的缓存里，所得数据必须依赖显示类部件才能查看，如：

- 1、显示报警信息缓冲区的内容，则要在画面上放置报警显示类部件 (报警条、当前报警表、报警历史表等)；
- 2、显示历史 XY 图缓冲区的内容，则要在画面上放置 XY 图显示部件；
- 3、显示趋势图缓冲区的内容，则要在画面上放置趋势图或者百分比趋势图显示部件；
- 4、显示数据记录区的内容，可以通过数据记录显示、历史曲线图部件来显示；也可以上传到上位机来通过特定的读取工具读取信息。

WebServer 数据记录区的详细操作，请参考 WebServer 一章。

多机互联数据区的具体操作，请参考多机互联使用说明一章。

### 2.5 向量图

向量图是部件的主要表现方式之一，向量图其实是表现简洁生动的几何图形，概括地表现了工业现场的各种情况，尤其是它的动态显示方面，有着比位图更丰富的表现方法。

每一种部件，都有多个与它对应的向量图。在每个部件的属性设置里，可以在向量图列表里，选取合适的向量图，如图 2-3 所示：



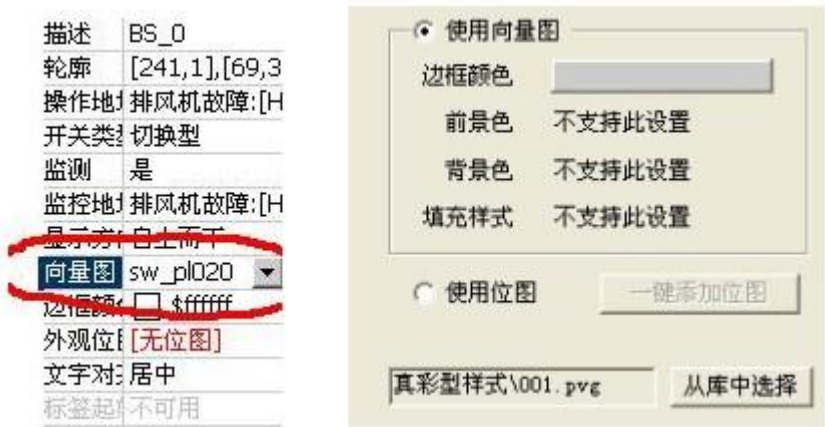


图 2-3 向量图属性

点击向下键，会弹出与部件对应的向量图列表，如图 2-4 所示。



图 2-4 矢量外观选择

一个向量图分为两部分：一部分是静态的，在向量图的显示过程中不会变化的，不能设置的；另一部分是可变化的，可以根据 PLC 寄存器值的变化而改变其显示效果的。

2.6 工程

在软件中，客户搭建的一个 HMI 应用被称之为“工程”，每一个工程都包含了必备的基本元素：

●工程配置

工程所应用的控制系统(PLC 或者仪表)型号, HMI 型号, 以及软件启动工程时所必须初始化的系统参数、工程启动时初始画面号等;

### ●画面

客户根据实际需要定制的图形页面, 画面里面包含了实现监控功能的部件等, 画面能比较形象地显示被监控的控制系统行为, 并给受控系统发送相应的控制指令。

### ●资源库

客户在制作画面时用到的资源:

**地址标识库**用来整合当前工程的地址资源;

**工程位图库**用来整合当前工程的位图资源;

**文本库**用来整合当前工程的文本资源;

**字体库**用来在当前工程中支持所有的 Windows 矢量字体。

### ●信息录入

包括位报警区、字报警区、数据记录区、实时曲线图、历史 XY 图、WebServer 数据记录区、多机互联数据区等信息的录入。

工程编辑好以后, 需将工程编译成 hmt 文件, 方可做离线模拟或者下载到 HMI 上运行。

## 2.7 部件

部件是组成画面的基本元素, 对 PLC 的监控和控制都必须依赖于部件, 部件分为三大类:

### ●控制类部件

这类部件既可以显示 PLC 的状态, 也可以向 PLC 写入数据, 这类部件有位开关、字开关、超级组合按钮、数值输入/显示、字符输入/显示、配方传输部件等。

### ●显示类部件

这类部件用于显示 PLC 的数据或者状态, 包括各种指示灯、各种仪表部件、动画、报警显示、配方显示、旋转位图、事件显示、趋势图、历史 XY 图等。

### ●特殊部件

这类部件是为了实现特殊的功能而实现的; 包括打印部件, 花型预览图、圆盘记录图、历史圆盘记录图等。

## 2.8 画面

画面是工程的基本组成元素, 也是 HMI 产品的一个基本概念, 画面概念与其它产品是一样的, HMI 任何显示或者操作都必须放置在画面上来完成。画面分为基本画面和子画面, 基本画面与子画面的区别主要是:

- 如果一个画面独立显示出来则是基本画面(可以通过功能开关切换), 反之, 如果一个画面的显示必须依赖画面显示部件则为子画面; 调用到子画面的部件: 直接画面显示、报警画面。

- 基本画面的大小就是显示屏的实际大小, 而子画面的大小是可以指定的, 这是最重要的区别。

## 2.9 编译

如果要在 PLC 上模拟工程, 则要编译成 HMT 文件, 编译是一个对工程进行全局优化的过程, 编译后

的工程可以在模拟器上高效的运行，编译过程还会报告工程发现的警告和错误，以使用户修改这些错误或者警告。

在编译完成后，自动生成一个该工程的 HMT 文件。

## 3 开始一个新工程

### 3.1 建立新工程

启动后，我们可以通过菜单或者工具条的快捷方式来新建一个工程。具体的做法如下：


在工程管理器中选择菜单“工程/新建工程”，或者点击工具栏的 “新建工程”按钮，出现“新建工程”对话框，如图 3-1。



图 3-1 新建工程窗口

根据图中的提示，默认工程名为 NewProject，选择一个存放的路径，在工程属性选择 PLC 驱动和 HMI 型号，点击下一步按钮，进入模版选择对话框，如图 3-2 所示：



图 3-2 工程初始化对话框

在这里您可以选择工程初始化的方式，如果选择**使用模版**，那么将按照选择的模版来创建工程。为了能够更加全面而清晰地展示如何创建一个工程，在这里，我们选择了**自定义**，即创建工程时我们只需要一个初始画面和提供的一些风格。

点击**完成**之后，对话框将关闭，在工程配置对话框里呈现出工程的基本情况，如图 3-3 所示。



图 3-3 工程配置窗口

在工程配置窗口中我们可以看到在树形控件分为基本画面、工程配置和资源库三个分支。我们可以看到工程所创建一个主画面；在工程配置的节点下我们可以设置工程中使用到全局的配置信息如系统参数、配方、报警信息、事件信息、趋势图信息和 XY 图信息；在资源库中我们则可以设置地址标识库、字符串库和工程位图库的信息。这些具体的内容会在其他的章节进行阐述，请参考其他的相应内容。

至此，新的工程已经建立。在继续制作工程前，需要设置下部件属性框的格式。在菜单栏下的“工具”——“软件设置”，选择“使用对话框模式”或“使用属性栏模式”，两者可以都选上。

新建工程时候支持返回选择型号：



图 3-4 工程属性

3.2 设计画面

工程编辑器是个集工程管理和画面设计于一体的集成开发环境。在工程建立之后我们就可以开始设计新的画面。

在这个例子中我们将使用位开关，数值输入显示部件。

3.3 建立新画面

为建立一个新的画面，在工程管理器中选择菜单“画面/新建画面”，或者点击工具栏的“新建画面”按钮，出现“新建画面”对话框，如图 3-5。

在新建画面对话框中需要输入画面的基本信息画面号和画面名称，画面号不能重复，在新建画面的时候工程编辑器会自动根据当前最后的画面号进行增加，在这里我们设置画面号为 1，画面名称不填写。

新建画面

基本信息

画面号(I):

画面名称(N):

安全等级

当前等级(C): 零级

画面背景

画刷填充(F)

背景色(G):

位图填充(B)

无位图填充

背景...

子画面(S)

宽度(W):

高度(H):

确定(O)

取消(C)

图 3-5 新建画面

我们可以在安全等级中设置当前的等级为一级，画面背景的属性框中设置画面的背景色、前景色和填充图样。因为这个画面是基本画面，所以子画面的属性在这里不设置。在设置完这些画面的信息之后，点击“确认”按钮，整个工程编辑器的格局变成如图 3-6 所示的画面。

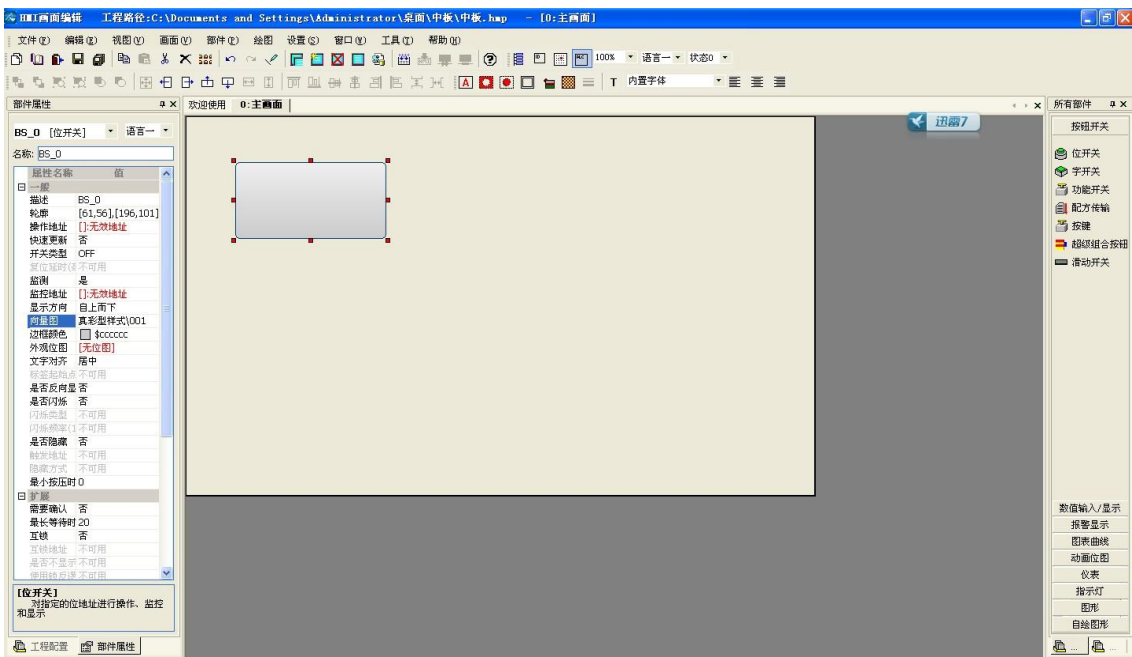


图 3-6 窗口布局

如图 3-6 所示，在屏幕中间的编辑区域是画面的大小的一个编辑区域，在画面区域的上面是画面切换的 Tab 框，在工程管理器的基本画面节点中自动添加了 1: 这个节点。左侧的部件和图形箱这个时候变得可用。至此，我们已经成功新建了一个基本画面。

3.4 添加文本说明

在完成上面的设计后，为了与后面我们要做的其他的类型的位开关进行区分，在它们的下方放置一个“ON 开关测试”文本内容加以说明。

文本的部件是在自绘图形栏里，点中“文本”，在画面的适当位置点击，即显示“无文本”，在文本属性框中输入“ON 开关测试”。默认颜色为黑色，设置字体。字体修改后，下一次再使用文本部件，默认使用上次文本设置的字体。参考画面如图 3-7 所示。

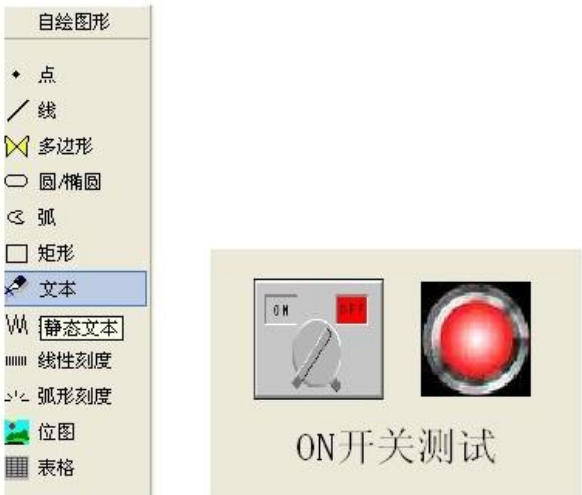


图 3-7 文本



### 3.5 添加一个位开关部件

新建完一个画面之后，我们就可以在画面中添加我们需要的部件，设计出满足不同需要的功能。下面我们设计一个简单对于位开关和位开关指示灯的示例。

点击右侧部件箱中选择“位开关”，移动到画面的区域内，左键点击位开关想要放置的左上角的位置，此时会出现一个黑色可以缩放的矩形框，拖放到我们想要的大小，再次 点击左键，这样就在画面我们需要的位置放置了一个位开关的部件。我们可以按照此方法添加其他的部件。在放置完一个部件之后，我们所看到的画面如图 3-8 所示。

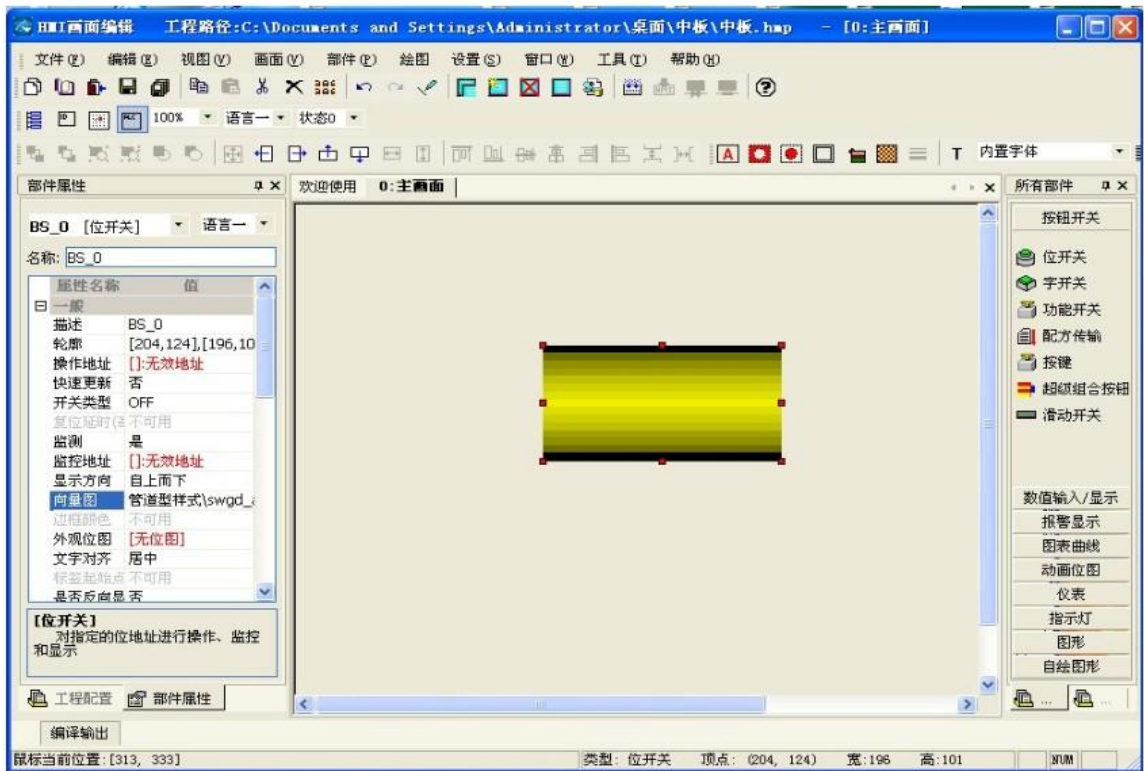


图 3-8 在画面中添加一个部件

在图 3-8 中，我们可以看到一个矩形向量图的位开关部件。屏幕左侧自动切换到该部件的属性窗口，我们可以在部件属性窗口中设置该部件的相关属性。

### 3.6 添加位状态指示灯

按照上述的操作方法，我们可以在画面中添加一个“位状态指示灯”，将“位状态指示灯”放置在位开关的右侧，设置“位状态指示灯”的监控地址和前面的“ON 类型”的开关一致为 HDX0.0，设置向量图为空，选择外观位图 Lamp 为其外观。在添加完成后可以看到画面的状态如图 3-9 所示。

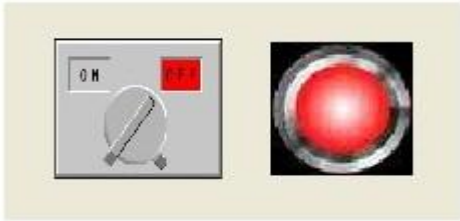


图 3-9 添加状态指示灯

### 3.7 位报警的使用

我们已经完成了一组位开关的设计。接下来对这个位做报警处理。

首先我们需要先在左侧“属性编辑框”——“工程配置”——“工程设置”里添加“位报警区”做记录，或者点击菜单栏“设置”里的位报警区，点击“添加”，如图 3-10 所示。在弹出的设置框中输入位地址 HDX0.0，设置触发条件为“ON 时报警”，填写报警文本。“是否记录”选项勾选上时，报警记录会记录到 CF 卡里（前提是在工程参数里设置存储介质为“外接 CF 卡”）。报警画面选择“无”。如果发生报警时要求跳出报警画面，则在这里选择报警画面即可（报警画面必须为子画面）。

然后在画面上放置报警条，属性框中的“是否透明”选择“是”，这样报警条就不会显示背景色。最后画面如图 3-11 所示。



图 3-10

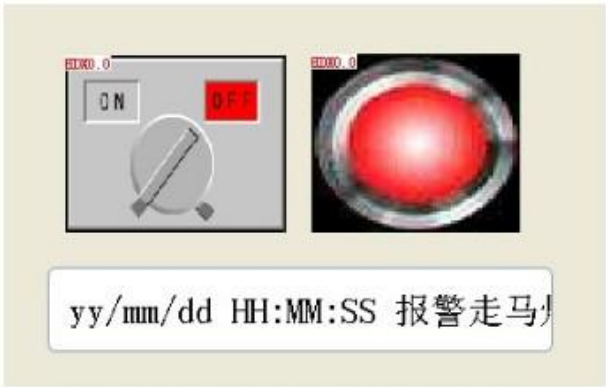


图 3-11

### 3.8 改变部件的外观

我们已经设置完成了一个位开关的类型和地址，但是大家在实际的应用中可能会需要不同形状的外观。在属性窗口中我们可以通过选择不同的显示方向、向量图、边框颜色、外观位图和文字对齐属性来设置。

显示方向设置了外观的不同显示方向，包括向量图和文本；文字对齐属性设置文本的显示位置。部分向量图可以选择各状态下的颜色，部分向量图不允许修改状态颜色，对应的颜色设置属性会以灰度显

示。

左键点击向量图属性的下拉按钮，出现如图 3-12 的外观选择界面，我们浏览这些不同的外观图案，选择一个合适的外观。



图 3-12 外观选择窗口

我们选择其中的一个含有 ON/OFF 指示的外观来作为位开关比较合适。选择完成后我们看到的 1 画面的显示如图 3-13 所示。



图 3-13 改变位开关的外观

### 3.9 使用位图外观—选择位图

左键单击“外观位图”属性的下拉按钮，弹出图 3-14 所示的位图选择窗口。浏览并选择 Switch 位图作为位开关状态的外观。点击“查看位图状态 (S):”，可以查看各状态下的位图。单击“使用此位图”按钮确认选择。

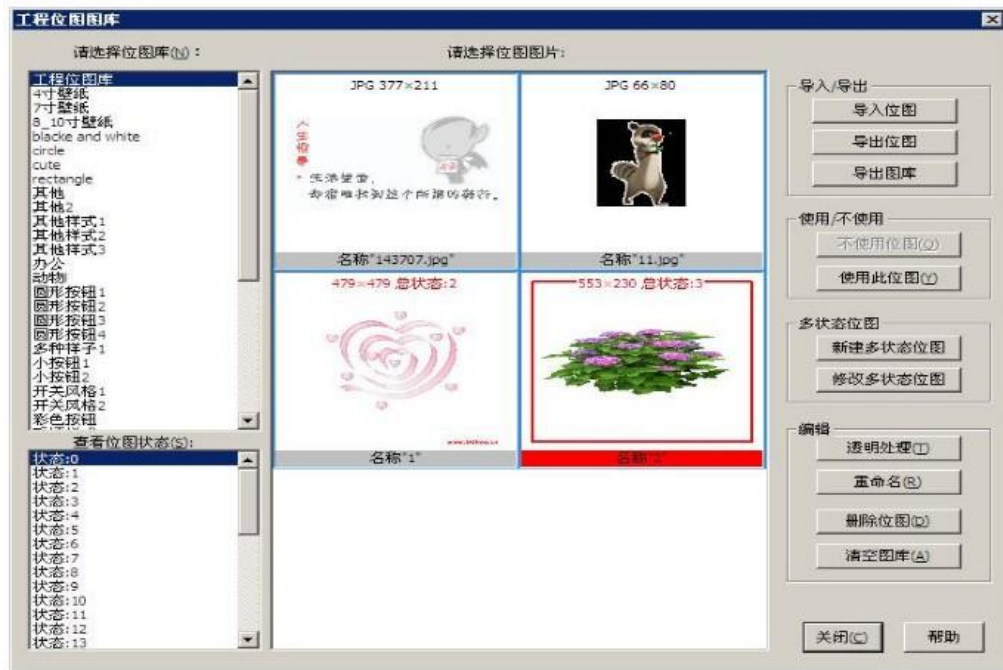


图 3-14 位图外观选择窗口

选择完成之后，画面 1 的画面状态如图 3-15 所示。

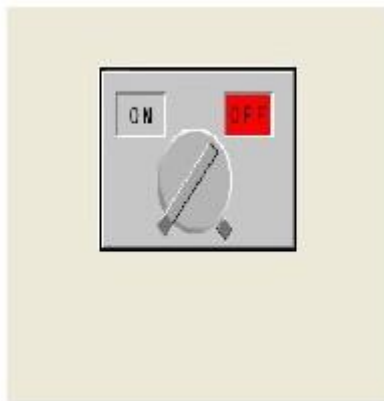


图 3-15 位开关使用位图外观后的状态

### 3.10 设置部件类型和操作地址

我们可以通过属性框修改部件的属性，属性的设置都在左侧的部件属性设置窗口，如果软件设置您选择对话框模式的话，双击该部件，就可以弹出部件属性对话框。不同的部件的属性会有些不同。在部件属性中最为重要的是和操作地址相关的属性，这个设置关系到我们这个部件的具体功能的完成，其他

的属性我们可以根据需要进行调整。在这里我们介绍简单的操作地址的设置。

首先我们在开关类型选择下拉选择窗口中选择“ON 型”开关，表示这个开关的操作是对位地址写值 1。另外，开关类型还有“置 off”型(置 0)、“切换”型、“复位”型。

操作地址可以直接输入，也可以点击右边的按钮，通过地址编辑器来写地址。弹出的地址编辑窗口如图 3-16 所示。



图 3-16 地址输入窗口

在地址编辑器中选择 HDX 地址类型，地址值为 0.0。确认地址输入，这个地址的含义是第 0 个字的第 0 个位。

在属性窗口中设置“监测”位是，同时设置监测的地址为 HDX0.0，则位地址 HDX0.0 的状态可以实时的通过开关的不同的状态表示出来。

### 3.11 编译和调试

在依据第二节所示的方法完整画面设计之后，我们就可以对工程进行编译和打包，如果编译无误，我们可以在离线的模拟环境中进行模拟测试。通过选择菜单“工具/工程编译”，或者在快捷键中选择编译工程，或者键盘的功能键 F9，对于工程进行编译。编译的窗口如图 3-17 所示。

在工程编译窗口中记录了各个画面的编译情况和打包处理的情况，双击编译信息，画面会指定到该报警或错误信息的部件，我们可以根据信息提示进行修改。

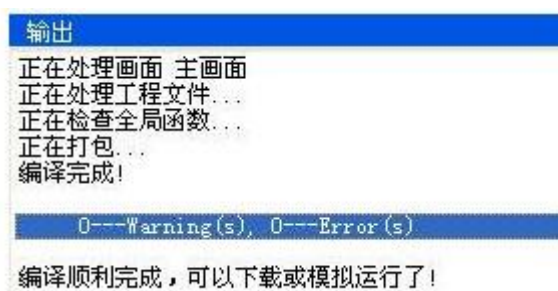




图 3-17 输出窗口

在成功编译完成后，我们可以通过我们提供的离线模拟工具进行测试。模拟运行的效果如图 3-18。



图 3-18 工程离线模拟运行

## 4 工程与画面

### 4.1 新建工程

从工程菜单里选择“新建工程”，如图 4-1。

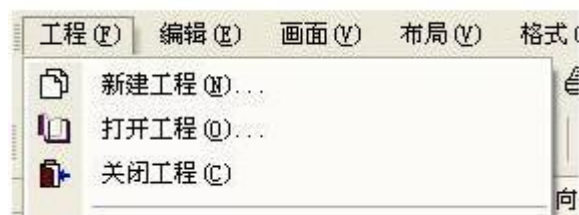


图 4-1 从菜单新建工程

或者从工具栏里选择，如图 4-2



图 4-2 从工具栏新建工程

点“新建工程”项，就会弹出下面的对话框，如图 4-3 所示，这里涵盖了创建工程所必需的初始化信息。

同时设置：



图 4-3 工程属性

**工程名：**是工程的名称，工程的名字必需由可以创建目录的合法字符组成也可以是中文。

**路径：**是工程存放到硬盘的路径

**PLC 型号：**当前 HMI 用于所连接的自控设备的型号。

**HMI 类型：**软件系列的具体产品型号。

点击“下一步”，进入工程模版对话框，用户可以通过该步骤来对工程进行初始化。软件内置了一些工程模版，基于这些模板框架搭建工程可以极大地提高制作工程的效率，并可以让工程以用户比较熟悉的风格与形式呈现在用户面前。

如图 4-4 所示，各项的含义如下：

**使用模版：**使用软件内置的工程模版对工程进行初始化

**模版列表：**软件内置的所有工程模版的名称。

**预览窗口：**所选中模版的预览效果图。

**静态文本：**所选中模版的文字说明。

**自定义：**不采用内置模版初始化工程。

**起始画面号：**指定要创建的起始画面号。



图 4-4 工程初始化

点击“完成”，工程新建完毕，在工程视图里可以看到新工程的画面集合。

4.2 新建画面

软件提供了多个方式新建画面，比如：从菜单栏“画面”---“新建画面”，如图 4-5 所示：



图 4-5 从菜单新建画面

或者从工具栏中选择，如图 4-6 所示。



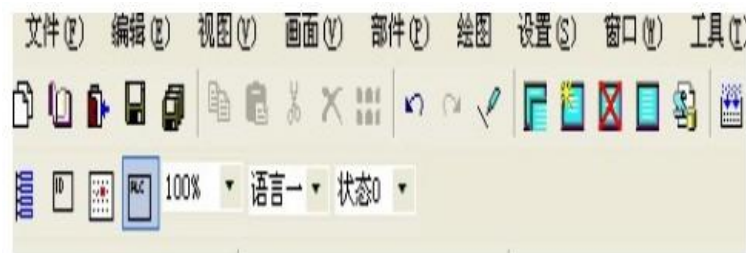


图 4-6 从工具条新建画面

从工程视图的快捷工具栏也可以新建画面，如图 4-7。



图 4-7 工程配置中新建画面

新建画面的对话框如图 4-8 所示。



图 4-8 新建画面窗口

窗口中具体的元素的含义如下：

**画面号：**画面的唯一 ID 号，是一个无符号整数，这个 ID 号是唯一的，不同的画面具有不同的 ID 号。

**画面名称：**画面的说明性描述，用于说明画面的用途。

**安全等级：**限定画面的安全等级，只有用户输入同级或者更好安全等级的密码才可以访问该画面。

**画面背景：**填充画面的前景色、背景色、填充图样，或者使用位图填充。

**子画面：**子画面的宽度和高度是可以指定的。

4.3 画面属性

打开工作区活动画面的属性对话框，调整其属性，如图 4-9 所示。



图 4-9 画面属性工具栏

画面号、名称、填充样式、以及是否是子画面，均可以调整，但是画面号不能与现有的画面号重复。

画面属性的对话框与新建画面是一致的。

4.4 删除画面

将工作区的活动画面删除，删除的画面将不可恢复，如图 4-10（红色方框处）。



图 4-10 删除画面工具栏

4.5 复制画面

复制画面是指从系统、其他工程或者当前工程复制一个新的画面到当前工程，画面复制的功能可以避免重复劳动，提高工程的制作效率，只要画面类似，便可以从以前的工程把画面复制过来。从下面工具条可以打开复制画面的对话框，如图 4-11 所示。



图 4-11 复制画面工具栏

复制画面的对话框如图 4-12。



图 4-12 画面复制窗口

其中窗口界面中元素的内容说明如表 4-1 所示。

表 4-1 画面复制窗口界面元素说明

界面元素	说明
本地	源画面来自本地工程
系统	源画面来自系统，软件内置了一些基本的画面，如各种键盘画面、向导画面等等，用户可以从系统引用这些画面，节省大量制作时间。
其他	画面来自其他工程，可以通过浏览找到其他工程。
画面列表	将源工程可以复制的画面都列举出来。其内容显示格式是[画面号]: [画面名称]
画面预览	显示当前选中画面的简略图。
目的画面 编号	画面复制到工程后的画面编号，新的编号不能与已存在的画面编号相同，否则复制不成功。
目的画面 名称	复制后的画面名称
确定	进行画面复制，并关闭对话框
取消	取消当前复制。

4.6 画面与子画面

画面可以独立地在 HMI 上显示，而子画面则要依赖直接/间接画面显示部件、功能开关中的弹出窗口来显示，在软件上显示的基本画面可以由功能开关来切换，功能开关切换的画面只能是基本画面。

一个基本画面可以同时显示多个子画面，这取决于基本画面放置了多少个画面显示部件。由于子画面概念的引入，给软件的工程增加了极大的灵活性。

子画面与子画面之间还可以嵌套显示。

画面之间的用户信息是不通透的，也就是说在多个画面叠加使用的时候，用户发出的信息只会被最上层的画面所接收处理，系统不会将消息下传到下面的画面。

关于画面与子画面的区别，表 4-2 可以描述清楚。

表 4-2 画面和子画面的比较

比较类别	基本画面	子画面
显示	可以独立显示，通过功能开关切换，也可以指定为起始画面	依赖直接画面显示部件、间接画面显示部件来显示
尺寸	触摸屏的实际大小	可以指定大小，具体尺寸还与画面显示部件的实际尺寸有关
消息	本画面处理用户消息	最顶层的子画面处理消息

传递		
显示 顺序	基本画面可以直接显示在软件上	子画面的显示顺序取决于画面显示部件的顺序，在软件中，画面显示部件的顺序决定了画面的显示顺序
部件	可以使用所有部件	可以使用所有部件

## 5 系统参数

### 5.1 一般属性

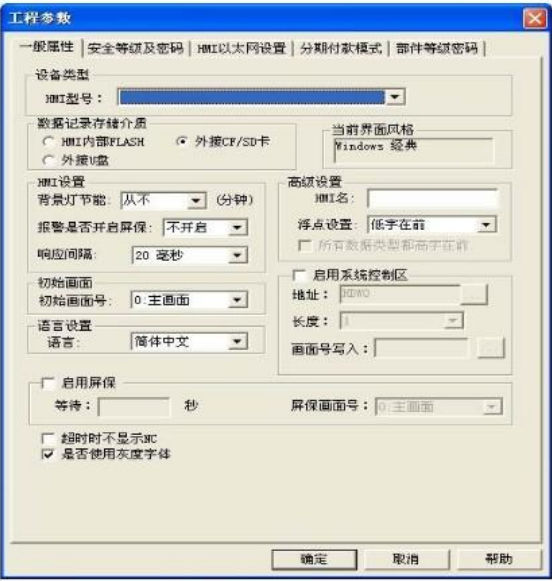


图 5-1 工程参数

表 5-1 一般属性说明

界面元素	说明
HMI 型号	HMI 的具体型号
数据存储介质	如果录入了数据记录区，记录数据的存放位置。
背景灯节能	在指定的时间内如果没有触摸屏消息，系统将自动关闭液晶背光灯，用户点击触摸屏将自动点亮屏幕。
报警是否 开启屏保	当选择背景灯节能后,如果选择不开启,一旦发生报警,屏幕也不开启,若选择开启,一旦报警,背景灯自动打开.
响应间隔	连续点击触摸屏的最小时间间隔
编译语言	工程的编译打包语言
浮点设置	设置处理浮点时候的一些设置，低位在前或高位在前。
初始画面号	工程运行时，所显示的第一画面，该画面应该为基本画面
屏保	设置相应的等待时间和屏保执行时的画面
超时时不显示 NC	超时时不再出现 NC 提示
是否使用灰色字体	是/否
高级设置	HMI 名：设置需要的 HMI 名 浮点设置：设置整个工程里面的高低字的关系
系统控制区	系统控制区提供了 PLC 程序控制 HMI 程序的一个途径。 地址：外部地址或内部地址，用来控制画面切换。

	长度：设置为 1。 画面号写入：外部地址或内部地址，用来显示当前画面编号。
--	--

5.2 通讯参数

从设置-通讯口配置进入通讯口设置界面。

通讯参数是 HMI 与 PLC 进行通信所必须配置的参数，它包括通信口的硬件参数配置与超时设置，设置和内容如图 5-2 所示，通讯设置主要是用来设置工程中所有有用到的通讯接口的通讯参数。



图 5-2 通讯参数设置

各个输入值的含义如表 5-2 所示。

表 5-2 系统参数通讯设置说明

界面元素	说明
HMI 站号	和设备通信时，HMI 的站号。默认为 0
PLC 站号	设置为 PLC 的地址编号。如果部件地址没有填写站号信息，那么部件就缺省访问这个站号。
串口参数设置	此设置和 PLC 的通讯参数设置一致！ 连接方式：RS232、RS485、RS422（com2 不支持 RS422） 校验位：EVEN（偶校验）ODD（奇校验）SPACE（空格校验） NONE（无校验）

以太网参数设置	设置以太网通讯参数 网络类型: TCP_Client_2N:TCP 协议, 可以接多个 PLC。 UDP_Client_2N:UDP 协议, 可以接多个 PLC。 TCP_Server:TCP 协议, HMI 做服务器。
超时设置	等待超时(毫秒): HMI 等待 PLC 响应的时间。 接收超时(毫秒): HMI 接收两个字符之间, 最长的等待时间 重试次数: HMI 与 PLC 通讯无响应时, 重试的次数 重试超时: HMI 与 PLC 通讯无响应时, 重试超时时间内不会访问 PLC。 通讯延迟时间: HMI 与设备通讯时的速度 连读间隔(0 则使用默认值 10): 读 PLC 地址的间隔, 如两个同寄存器的地址, 地址间隔小于设置值, 则连读、写这两个地址, 否则分两次读写。
com2 扩展	接打印机或者键盘 com2 串口参数的设置和打印机的通讯参数一致。
PLC 调试支持	设置穿透通讯参数, 具体设置参看二十三章穿透应用。
修改配置 不 改变参数	当选择此项时, 修改配置的时候, 参数将保持不变。
用户自定义 协议	当工程用了用户自定义协议后, 先通过这个选项进入进行编辑相关的信息。

### 5.3 分期付款模式

HMI 系统时间超过分期设定时间时, HMI 会弹出分期密码输入画面。只有输入密码后, HMI 才能继续正常使用, 否则, HMI 一直显示输入密码的画面。

分期付款信息的设置, 可以在工程制作中设置, 如下图 5-3; 也可以在 HMI 上设置。功能开关的“功能选择”中, 选择“分期付款”选项。在 HMI 中, 点击它, 就可以进入分期付款设置画面。

时间判断的依据是 HMI 的系统时间。

分期付款设置画面和密码输入画面是系统内置的, 不能供软件使用者修改。

图 5-3 为分期付款模式的设置对话框。



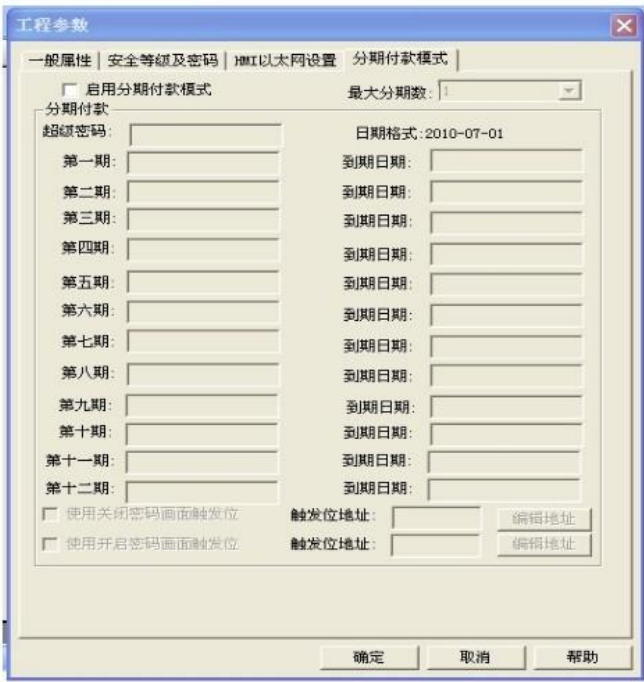


图 5-3 分期付款模式

表 5-3 分期付款模式说明

界面元素	说明
最大分期数	款项分几次分清。
超级密码	超级密码适用于任何分期。
使用关闭密码画面触发位	关闭分期密码画面后，会置触发位为 1，脚本里可以写指令来实现具体功能。
使用开启密码画面触发位	开启分期密码画面后，会置触发位为 1，脚本里可以写指令来实现具体功能。

图 5-4 为密码输入画面。



图 5-5 分期付款密码输入画面

图 5-6 为在 HMI 上设置分期付款信息的画面。

总结算期数：

1

起始期数设置：

0

超级密码：

目前设置处于：第

1

期设置

分期付款：

1

分期付款：

2010

年

1

月

1

日

上一期

下一期

保存

取消

退出

图 5-6 HMI 上分期付款设置画面

5.4 HMI 以太网设置

点击“工程参数”→“HMI 以太网设置”可进行以太网参数设置。

图 5-7 为 HMI 以太网设置对话框。

工程参数

一般属性 | 安全等级及密码 | HMI以太网设置 | 分期付款模式 | 部件等级密码

HMI以太网参数设置

HMI IP地址：

192

168

1

2

HMI 子网掩码：

255

255

255

0

HMI 默认网关：

192

168

1

1

HMI以太网高级应用选项

☐ 使用WEBSEVER服务器

☐ 使用WebServer密码保护

WEBSEVER服务器默认端口

80

页面存放路径

☒ 默认ASP页面

☐ ASP页面存放工程内

☐ ASP页面存放CF卡内

确定

取消

帮助

图 5-7 HMI 以太网设置

表 5-4 HMI 以太网设置说明

界面元素	说明
使用 Webserver 服务器	将此打开，即可使用 webserver 功能。
使用 Webserver 密码保 护器	若将此选项打开，即可将 webserver 功能加入密码。若有客户通过浏览器访问此主机时，将会提示输入密码。（在软件安装文件夹里 UserASP---USER. ASP 网页文件进行设置密码）
页面存放路 径	可选择不同页面存放地址。

## 5.5 安全等级及部件等级密码

### 安全等级

如果工程使用了安全等级，用户必须输入密码才能访问受保护的画面或者按键，安全等级的使用可以有效地保护设备，防止客户和没有权限的人误操作。

软件提供两种安全保护方式：画面密码保护和按键密码保护。

### 画面密码保护

每个画面都有自己的安全等级，只有输入相应的密码才能访问同等级的画面，比如，如果当前输入的密码是等级一的密码，则高等级的画面将不会显示，除非输入高等级的密码。

### 按键密码保护

对 HMI 进行安全保护，当用户在设定的时间内没有触碰屏，则屏就自动锁起来，通过点击画面的功能开关的（密码保护），在弹出的键盘中输入密码，即可激活触摸屏。

如果不使用安全等级，则当工程在 HMI 上运行时，则用户可以随意访问 HMI 的任何画面，而不会受到限制。

### 部件等级密码

使用部件等级密码后，当使用对应的部件时提示输入密码才可以使用该部件。

（部件密码中等级**自动降低**的意思是说：

如果选择“是”：比如当前部件的等级是等级 2，输入密码后。再次操作其他部件如果其他部件的等级是 $\geq 1$ 的那么都是需要输入密码的。

如果选择“否”：比如当前部件的等级是等级 2，输入密码后。再次操作其他部件如果其他部件的等级是 $\geq 3$ 的才需要输入密码。 $\leq 2$ 的都是不需要输入密码的。再次点击这个部件还是需要等级 2 的密码。）



图 5-8 安全等级设置

表 5-5 安全等级说明表

界面元素	说明
使用安全等级	在工程中使用安全等级，默认为不使用。
画面密码保护	使用画面安全等级模式保护工程。
等级一	等级一的安全密码，系统地址位 HSW404~HSW407，共 8 个字符。
等级二	等级二的安全密码，系统地址位 HSW408~HSW411，共 8 个字符。
等级三	等级三的安全密码，系统地址位 HSW412~HSW415，共 8 个字符。 (后面的等级密码地址类推)
启动等级	工程运行时，默认的安全等级。
各等级密码独立	勾选：每次进入高等级画面都需要输入密码。 不勾选：开机后第一次进入高等级画面需要输入密码，再次进入该等级画面就不需要输入密码。
按键密码保护	设置 HMI 的保护密码。
密码	按键保护的密码。
按键超时	超出设置时间，则 HMI 被锁起来了。
设计者密码	这个密码用于从 HMI 上传应用时，需要上传者输入密码。

## 6 位图与位图库

### 6.1 位图库

位图库位图库可以从“设置”→“位图库”打开。

位图库分为“本地图库”和“系统图库”，新建一个工程时，“本地图库”缺省是空的，用户添加的位图都将显示在本地图库中，而“系统图库”则软件自带的精美的图库，这些图库都可以导入到本地图库中来，或者直接使用，系统图库是分类的，一个图库文件就是一类形状和特性比较类似的图库。

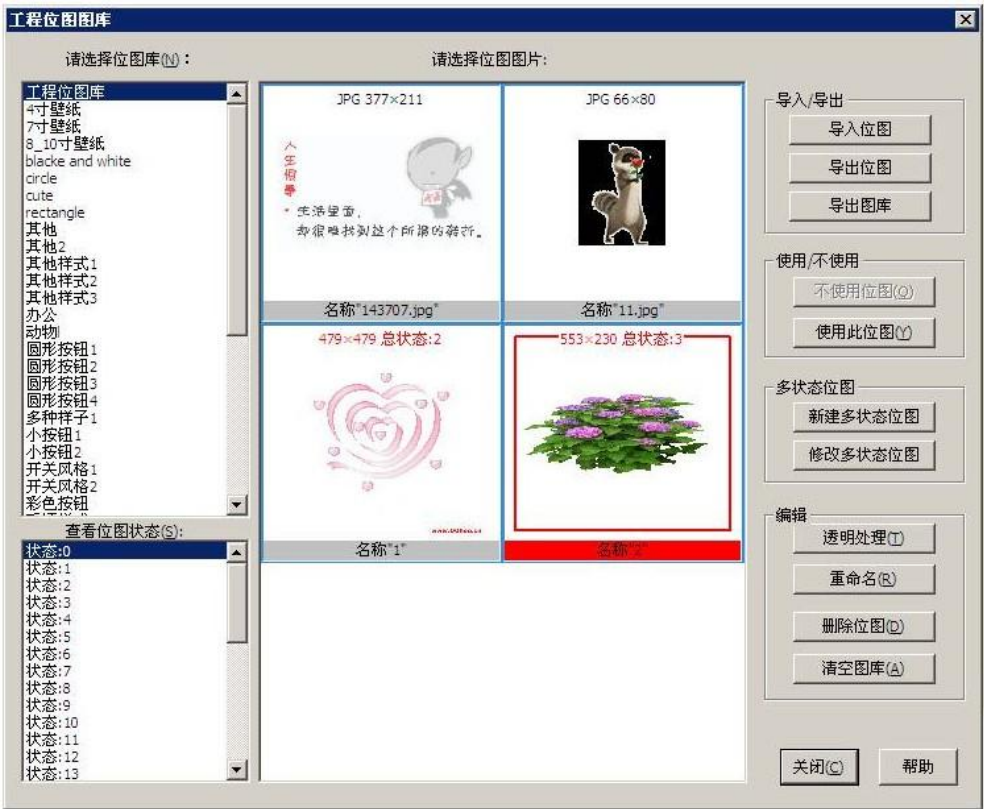


图 6-1 位图图库编辑窗口

如图 6-1 所示，通过这个窗口可以进行位图库的操作，表 6-1 对于这个窗口中的各个元素进行详细的说明。

表 6-1 位图库窗口说明

界面元素	说明
导入位图	导入一个或多个 HMI 位图
导出位图	将当前工程的位图另存到一个图库文件
导出图库	将位图库图片以 .BLB 格式存入电脑
多状态位图	新建编辑多状态位图：建立所需状态数的位图

	修改多状态位图：对已存在多状态位图进行编辑修改，可进行位图增加, 替换状态图片，删除状态图片等操作，如图 6-2。
使用/不使用	对某一位图进行使用和不使用的操作
透明处理	编辑图片，可以消除指定颜色
重命名	对选中的 HMI 位图进行重命名，HMI 位图的名字是唯一的，是不可重复的。
删除位图	删除当前选中的位图。
清空图库	一次性删除本地图库的所有图片。
关闭	保存位图所做的操作，关闭图库对话框。

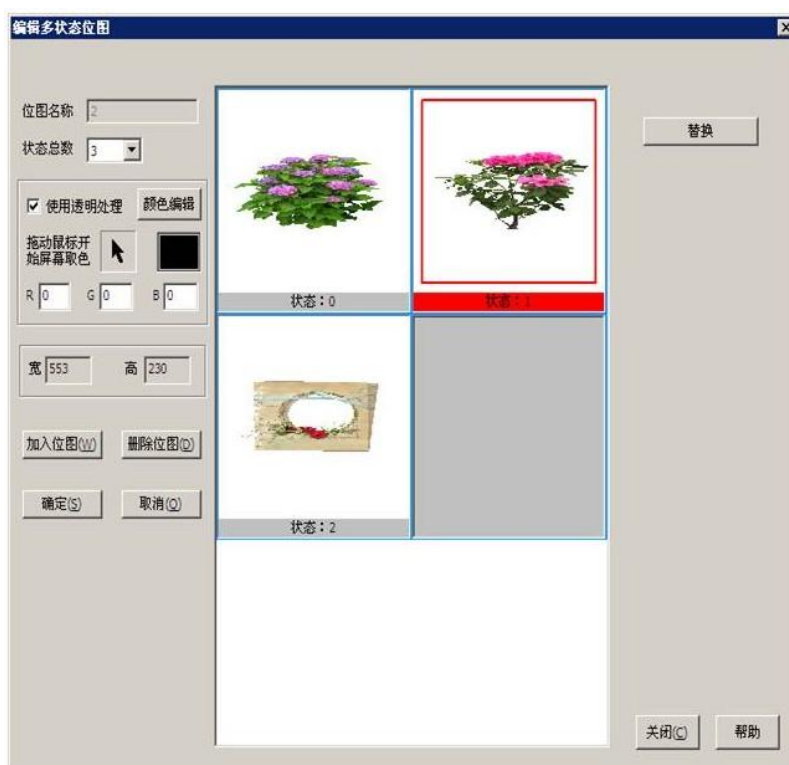


图 6-2 多状态位图编辑窗口

## 6.2 位图

使用位图可以设计出更加精美的画面。软件既可以支持静态位图，也可以支持有状态的位图，而且还可以使用 GIF 图片，有状态的位图可以用于多状态的部件，比如位开关、字开关、位状态指示灯、字状态指示灯等。

在软件中，所有的位图都是通过工程位图库来使用的，也就是说，部件所使用的位图是从工程位图库引用而来的。

## 6.3 HMI 位图

软件所说的位图与普通的位图概念是不一样的，与普通位图区别在于 HMI 位图是由多张位图组成，一个 HMI 位图是有状态的(最多有 32 个状态)，每个状态对应着一张普通位图，这种位图成为 HMI 位图。

6.4 位图编辑

可以通过位图编辑，更换指定状态的位图，或者改变位图的透明显示参数，点击“设置”-“位图库”-“位图编辑”。出现如图 6-3 所示。

**注：**GIF 图片不可以指定透明色，只有 BMP 图片可以指定不失真像素的透明色。

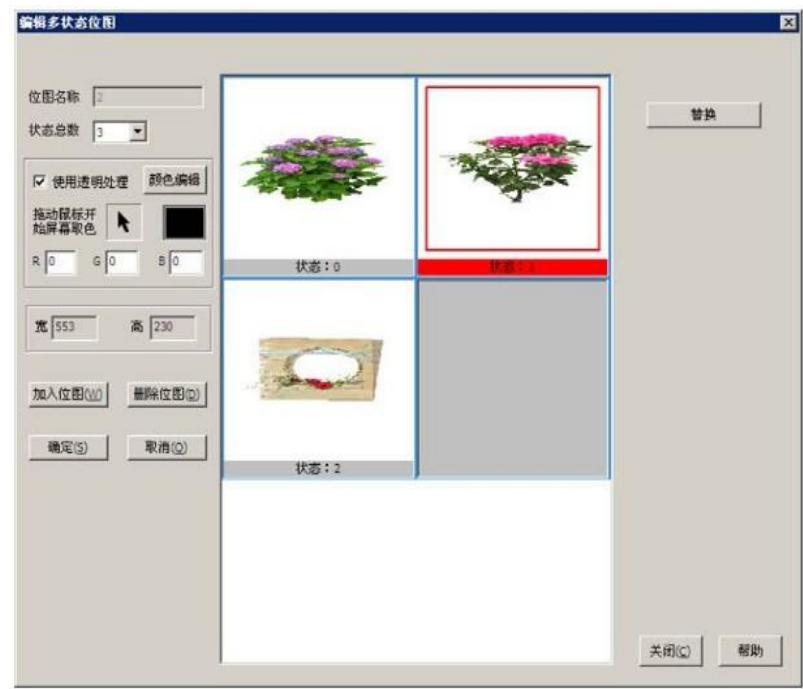


图 6-3 位图编辑窗口

表 6-2 的内容对图 6-3 的位图编辑窗口进行了详细的说明。

表 6-2 新建位图窗口说明

界面元素	说明
使用透明	是否使用透明色
替换	替换某一状态的位图
拖动鼠标开始屏幕取色	按住鼠标拖动可以在屏幕上选取颜色
状态总数	部件所具有的状态数
加入位图	加入当前状态的位图文件
删除位图	删除当前状态的位图文件
确定	关闭对话框，保存当前所作的修改
取消	取消当前所做的改动，关闭对话框



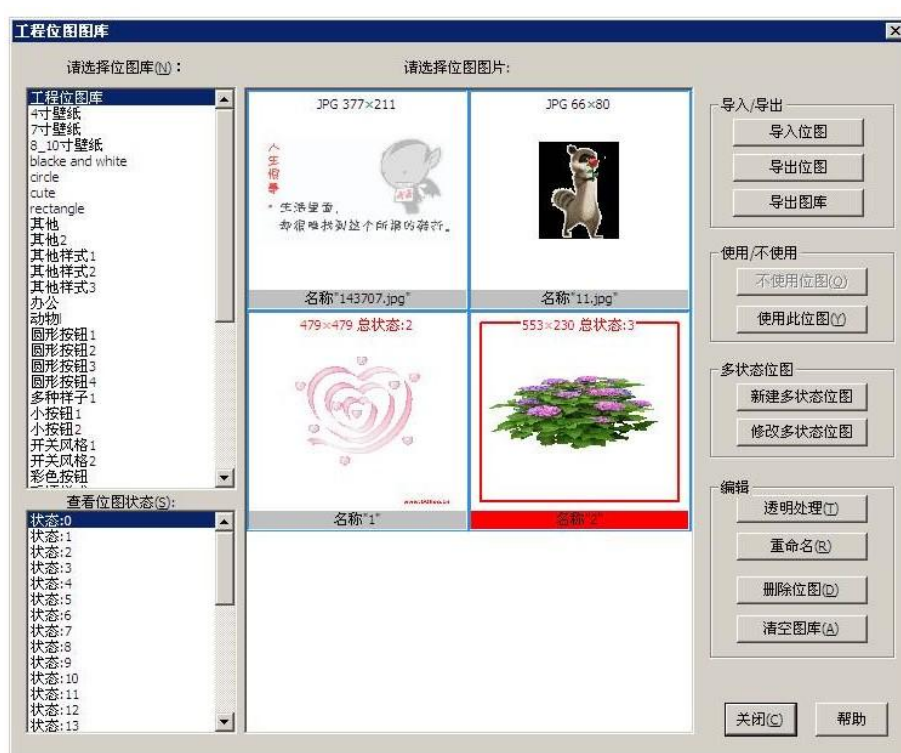
## 6.5 添加位图

在软件中添加位图

打开工程配置中的工程位图库



选择新建位图

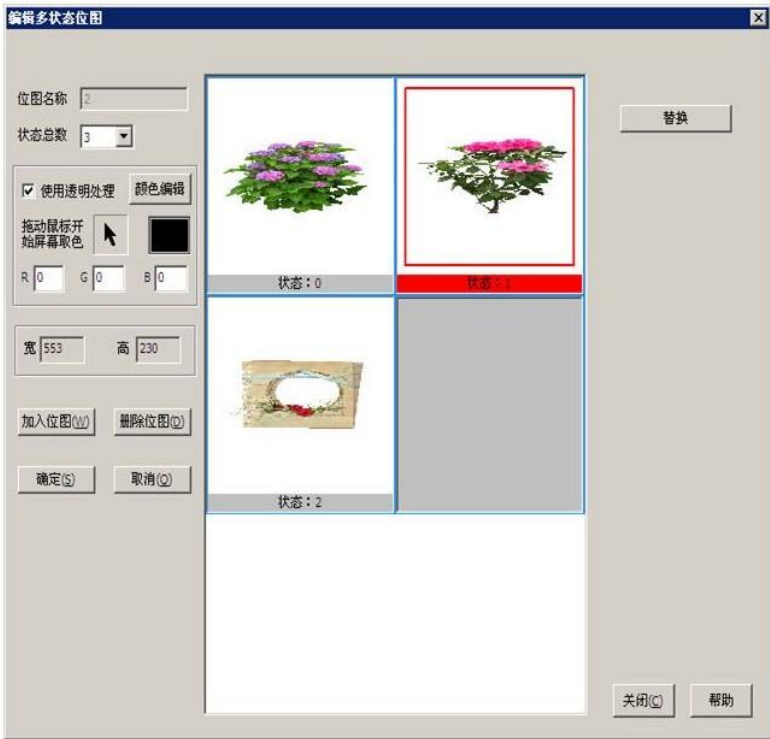


为您要添加的位图命名

如果是多状态的请在选择所要的状态

然后在所在位置添加入位图即可。





6.6 使用 HMI 位图

只要部件具有“外观位图”的属性，就可以使用 HMI 位图，“静态位图”对位图的引用也是通过图库来实现的。

在软件中，向量图和位图是可以共存的，当一个部件既采用了位图，又采用了向量图，那么系统将先描绘该部件的向量图，然后再显示位图。

在软件中，向量图可以随意拉伸而不失真。



图 6-4 外观位图选择属性

以“位开关”为例，属性“外观位图”的值是当前应用的 HMI 位图名称，外观位图获得输入焦点时，一个下拉按钮就会出现最右边，点击下拉按钮，出来图 6-5 所示对话框。

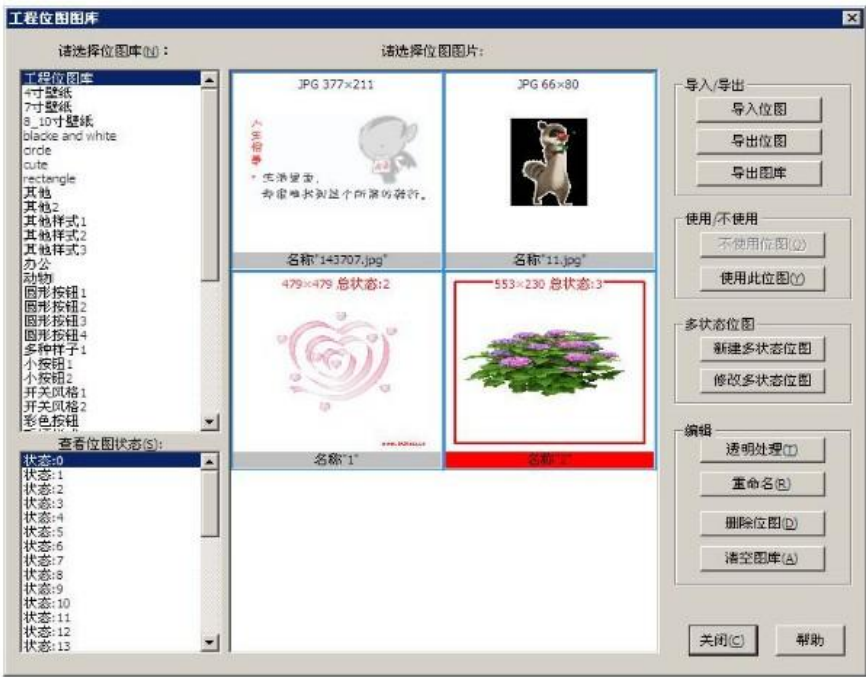


图 6-5 位图选择窗口

表 6-3 的内容对图 6-5 的位图选择窗口进行了详细的说明。

表 6-3 导入位图窗口说明

界面元素	说明
导入/导出	导入位图：将某一位图导入位图库 导出位图：将某一所需的位图导出指定的位置 导出图库：图库所有的图片都导出，导出成 b1b 格式文件。
使用/不使用	使用位图：对某一部件使用此位图 不使用位图：对某一部件不使用此位图
编辑	透明处理：对所用的位图进行透明处理，处理对应图片的背景色。 重命名：重命名图片名字 删除位图：对不需要的位图进行删除 清空图库：对位图库进行清空操作

当一个 HMI 位图选择到部件之后，系统会将此 HMI 位图所有状态的位图进行色深转换，转换成软件能够支持的位图类型。

如果选中的是非本地图库的位图，那么软件会把该位图导入到本地图库中来，然后，再转换成与色深相匹配的位图。

如果要在视图中查看部件不同状态的位图外观，只需要在视图操作工具条中切换一下状态即可。

## 6.7 导入图库

将一个全新的位图库导入到当前工程,可以将其他图库的某一个 HMI 位图导入到工程中来。点击“设置”-“位图库”-“导入图库/位图”,进入导入图库对话框,如图 6-6。

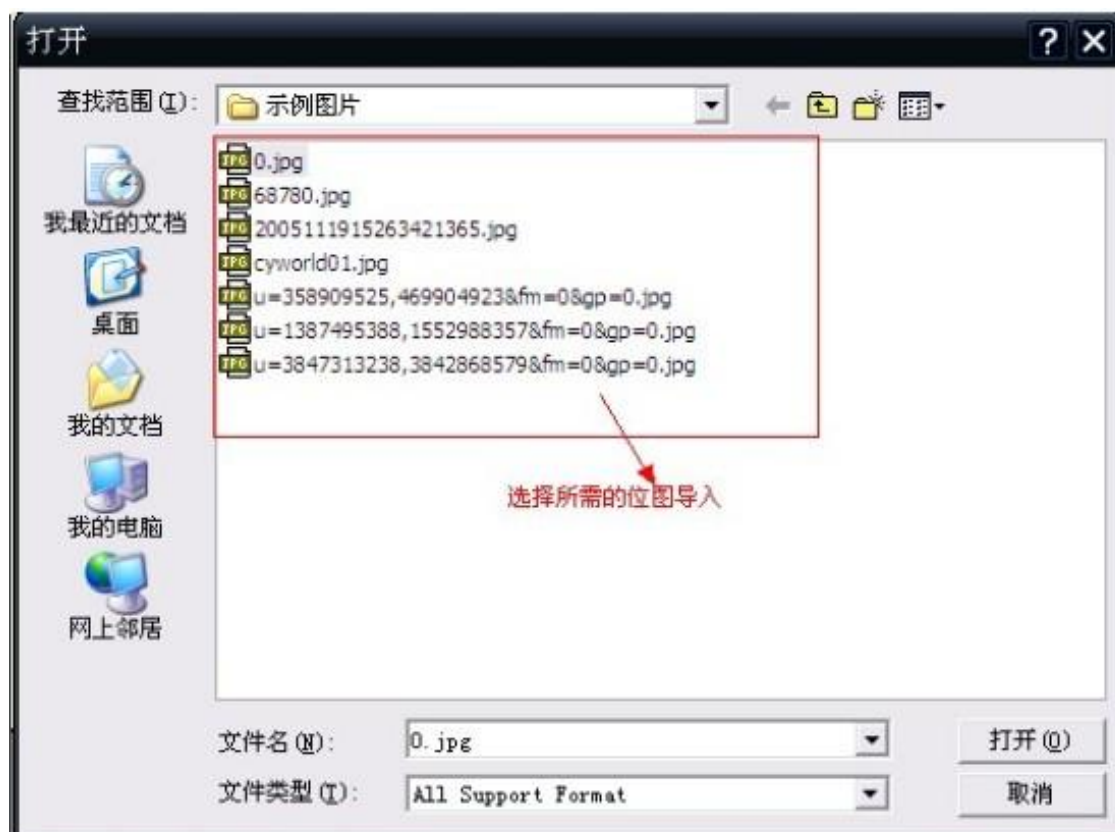
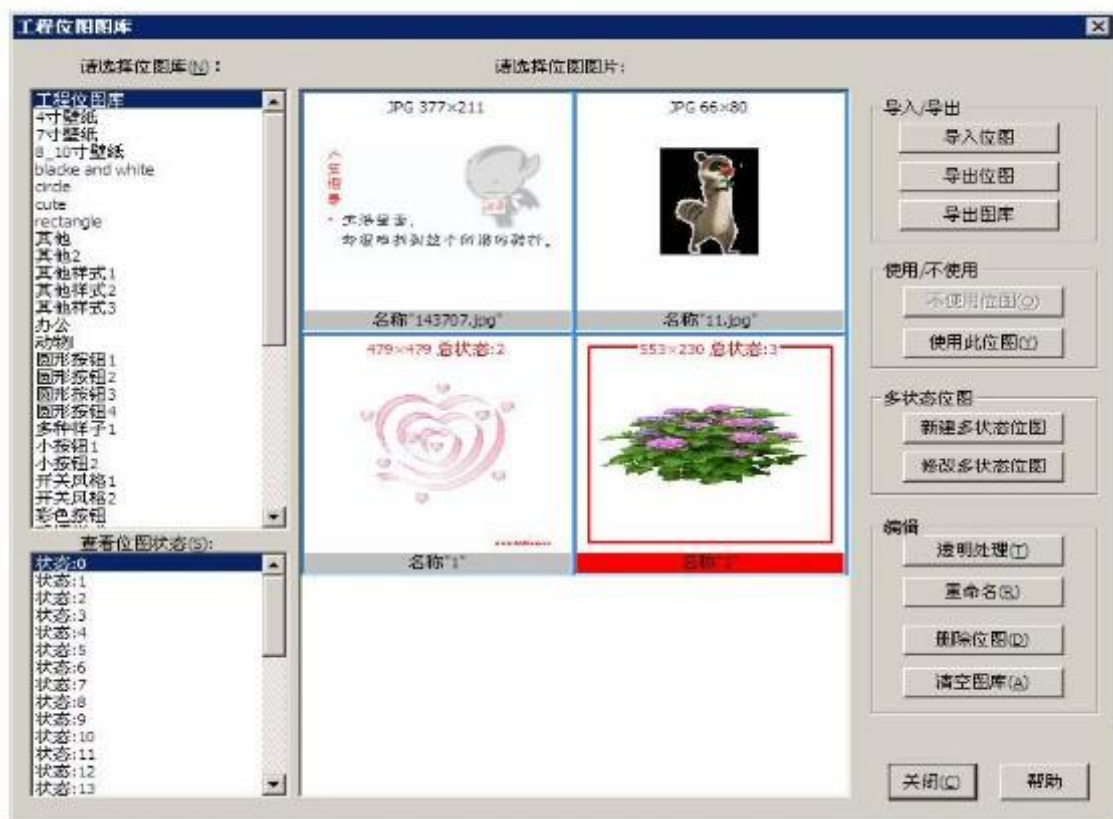


图 6-6 导入图库

软件可以导入各种色深的位图或者灰度图像。

6.8 新建多状态 HMI 位图

HMI 位图是多个普通位图叠加而成的，每张位图对应一个状态，软件提供了一个向导来引导新位图的创建，点击“设置”-“位图库”-“新建多状态位图”。出现如图 6-2 所示对话框。

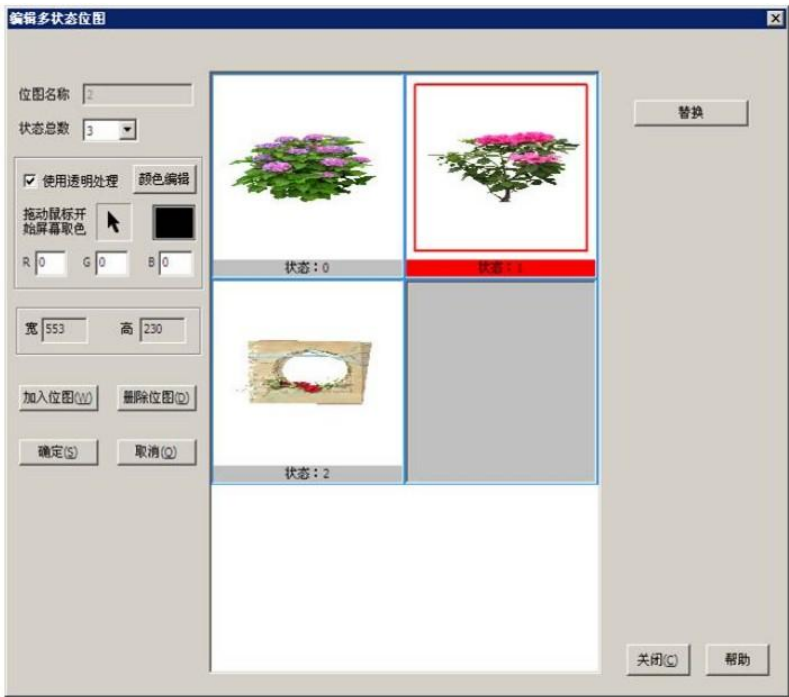


图 6-7 新建位图

表 6-4 对图 6-7 中的窗口中的各个元素进行详细的说明。

表 6-4 新建位图窗口说明

界面元素	说明
名称	新位图的名称
状态数	新位图的状态总数，支持多达 32 种状态
位图类型	BMP, JPG, PNG 图片，这些图片是静态图片 GIF 图片是动态图片
使用透明	指定颜色透明显示
RGB	透明颜色的值
宽/高	设置所需的高和宽
替换	将某一不需要的位图替换成所需的位图
删除位图	将不需要的位图删除
加入位图	将所需的位图加入到位图库

## 7 地址与地址库

### 7.1 地址库

软件支持的所有部件都是通过地址来和受控设备进行通信,受控设备的地址和受控类型因型号不同而不同。地址提供了部件访问(写入)PLC 数据的基本信息。

在 PLC 应用中,PLC 本身梯形图程序和与其连接的 HMI 都必须通过地址进行相关编程。绝大多数 PLC 的地址都有如下通用结构:

设备类型(如 I、IW)	设备地址(如 0, 1)	如果是位地址,则有可能还有位编号。
--------------	--------------	-------------------

提供了两种访问 PLC 地址数据的方法:“位地址”与“字地址”,支持的所有部件无非就是字地址或者位地址与 PLC 交互数据的。

此外,为了编程方便,也提供了一套保留的内部寄存器,其编址方式与 PLC 的通用结构是一样,其设备类型有三大类 HS、HP、HD、RP。其中 HSX、HPX、HDX 表示位地址,关于其具体用途请参考软件保留寄存器;其中 HSW、HPW、HDW、RPW 表示字地址,于其具体用途请参考软件保留寄存器。

这些软件内部保留的寄存器是重叠的,比如 HSW000001 是一个字,而它的十六个位可顺序表示为 HSX0.0, HSX0.2, …, HSX0.15。其他的保留寄存器也是一样的。

位地址的编码方式一般两种:间接编号和直接编号;间接编号是指通过字的方式来索引位地址,比如,HSX0.0 表示第 0 个字的第 0 个位,此时,中间必须以小数点隔开;直接编号就是对位进行直接索引,如果 I18,表示 I 标识的第 18 个位。

位地址的编码方式需要参考相关的 PLC 产品说明书;软件支持的编码方式与厂家所支持的方式是一致的。

地址作为一个字符串,除去 HSX 之类的标识,最多可达 32 个字符,但一般的 PLC 地址是 6 个字符,具体数值与受控设备的类型与型号有关。

不同厂家生产的 PLC,其编码方式与地址规则皆有不同,软件集成了所有能够支持的 PLC 的地址规则,并根据这些规则,对于用户输入的地址进行判断。

### 7.2 多连线与多站号

软件的新版本支持多个通讯口同时挂接多个 PLC 设备,并且在一个口可以挂接多个同种类型的设备。这种情况下,软件设计的地址字符串规则如下:

连线#站号:地址。

**连线：**就是要选择哪个通信口进行通信，范围 1~2。（如果没有连线，既没有标记符号“#”，则表示选择编号为 1 的连线进行通信）。

**站号：**设备站号，范围 0~255。（没有连线，既没有标记符号“:”，则表示按照“系统参数”的“通讯与连线”的预设站号设备进行通信）。

**地址：**地址字符串

连线和站号都可以省略不写，比如“A#B”表示按照编号为“A”的连线，访问缺省站号设备的“B”地址数据。而“A:B”则表示按照编号为“1”的连线，访问站号为 A 的设备的“B”地址数据

7.3 地址的使用

在软件中，使用到地址的地方大多以下列字段出现：**字地址、位地址、监控地址、操作地址**等。假设我们已经在画面中设计了一个字开关 WS3，则在部件属性窗口我们可以看到如图 7-1 所示的内容。



图 7-1 字开关的部件属性

如果属性“**字地址**”字段获得焦点并处于编辑状态，则会出现一个编辑框和一个下拉按钮，编辑框可以直接输入，一是可以输入该部件所使用的地址，如 IW000000。二可以输入所引用的地址库条目名称，如间接画面字。

如果用户不直接通过编辑框输入地址，也可以点击下拉按钮，将弹出地址编辑器，可以在地址编辑器的提示下输入地址。也可以引用地址库的地址条目。

如果用户的输入不正确，地址字段的属性值一栏将会显示无效地址。

其他地址与上面介绍的情况类似。

7.4 地址编辑器

地址编辑器是用来辅助生成地址字符串的工具，根据前面所述，部件的地址主要有下述部分组成：连线，设备站号，地址的设备类型，地址值。





图 7-2 地址编辑器

表 7-1 中详细说明了图 7-2 中地址编辑器的窗口中元素的详细内容。

表 7-1 地址编辑器窗口说明

界面元素	说明
直接输入	表示该地址不是来自地址库，是一个地址字符串
来自地址库	表示该地址从地址库选取，如果地址库没有数据，该项将不可用
名称	显示地址库的所有条目，当前选中的条目是被该部件引用的地址
连线	目前工程已经存在的 PLC 设备
PLC 地址 / 站号	设置要访问的设备站号，如果选择“预设值”，站号就是“系统参数”设置“通讯与连线”的预设站号。
设备类型	工程所选的 PLC 所包含的设备类型；
地址编辑框	用于输入地址值
A-F，0-9	是一个软键盘，可以鼠标来输入地址值；
清除	清除地址编辑框的内容；
删除	删除地址编辑框的最后一个字符；
关闭	关闭地址编辑器，对地址不做改动；
确定	关闭地址编辑器，并保存编辑结果； 如果选择“来自地址库”，则修改结果将保存到地址库相应得条目 如果选择“自定义”，则修改结果将保存部件的地址属性中。

静态框的内容是系统自动提示的帮助信息。选择不同的设备类型，提示的帮助是不同的。关于帮助的内容说明在表 7-2 中。

表 7-2 地址编辑器帮助说明

界面元素	说明
地址模式	表示当前地址是字地址还是位地址；
标识	当前选中的设备类型；
主编号	有些 PLC 的地址采用间接编号，则分为主编号和子编号，这里提示了主编号的编码范围，以及所采用的进制。
子编号位数	如果子编号位数为零，则表示地址是直接编号；
子编号进制	子编号所采用的进制(即小数点后的数，仅间接编号有)；
子编号范围	子编号所采用的范围(即小数点后的数，仅间接编号有)。

7.5 地址标识库

地址库是预先设置好的地址集合，从“设置”--“地址标识库”，可以打开地址库。图 7-3 所示。



图 7-3 工程编辑器中选择地址标识库

双击鼠标，打开地址标识库的编辑对话框，如图 7-3 所示。





图 7-4 地址标识库

表 7-3 对图 7-4 中的地址标识库窗口的所有元素进行详细的说明。

表 7-3 地址库界面说明

界面元素	说明
字地址	如果选中字地址，则列表里列举出来的是地址库里的所有字地址；
位地址	如果选中位地址，则列表里列举出来的是地址库里的所有位地址；
列表框	列表框有三个内容，ID 号是系统生成的 ID 索引号，地址名称是用户输入地址代号，地址库里的地址名称必须是互不相同的，地址值是当前条目实际代表的地址。
添加	如果选中位地址，则添加的是位地址，反之，则添加的是字地址。
插入	在当前行之前插入一个地址标识。
修改	对选中的条目进行编辑。
删除	将选中的条目删除，（如果有部件使用该地址，则这些部件的地址将变得无效）。
清空	清空所有标识。
上移	选中条目上移。
下移	选中条目下移。
关闭	关闭对话框。

7.6 地址预览

软件提供了一个方便的工具，可以查看当前工程的地址，或者是一个地址同时关联了那些部件。从视图工具条例选择下面的图标。如图 7-5。



图 7-5 地址预览工具栏

就会在工作区弹出部件一览表视图，如图 7-6 所示。

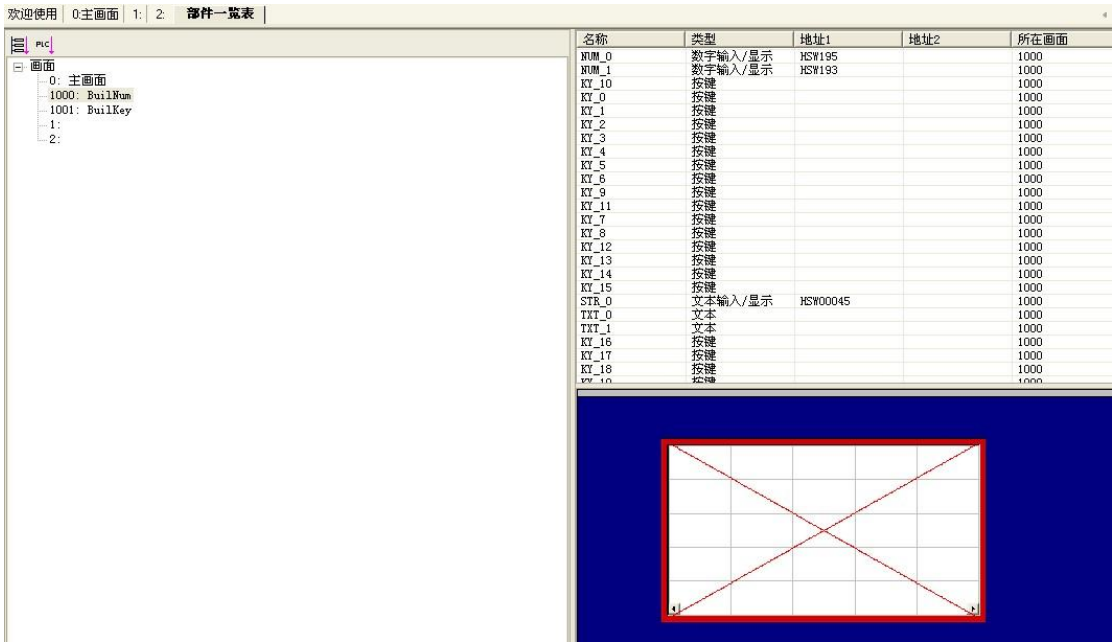


图 7-6 部件一览表窗口

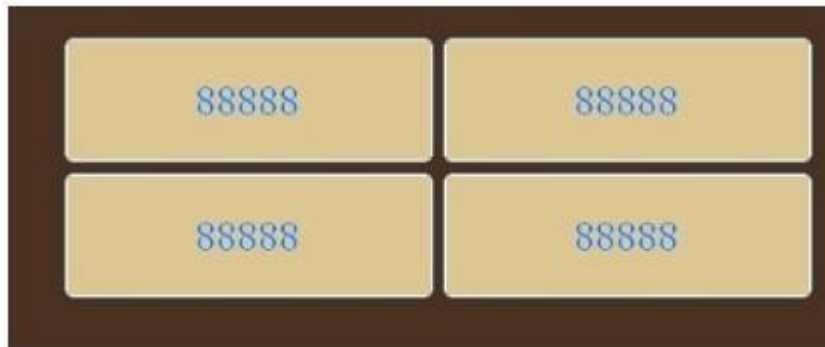
这个工具可以清楚地了解当前工程所使用的地址资源。也可以知道一个地址已经分配给那些部件。

## 8 部件的复制

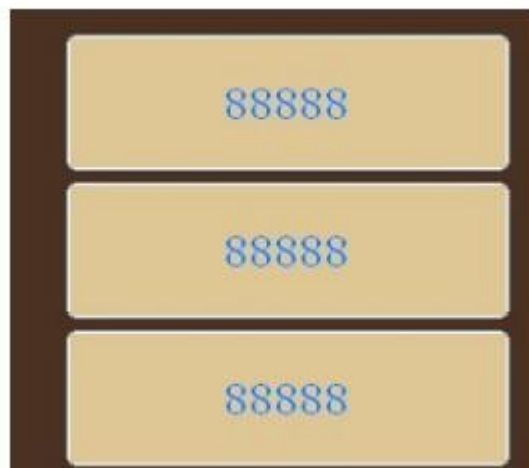
### 8.1 部件的复制

多重复制时可以选择横轴方向或者纵轴方向复制

横向：



纵向：



### 8.2 部件重叠

打开部件对话框时，如果部件重叠，出现部件列表对话框



## 9 文本库

### 9.1 使用文本库

在软件中，画面的静态文本、部件的状态文本、报警内容均可以引用文本库的内容。在部件中，能够使用文本的属性名一般是文本内容。

以按键部件为例，如图 9-1。



图 9-1 按键部件属性窗口

当文本内容获得输入焦点时,右边会弹出一个多行的编辑框和一个下拉按钮。多行编辑框可以输入,直接输入文本的内容,支持回车。

如果该内容已经引用了文本库，则多行编辑框编辑的是文本库该条目当前语言的内容，如果别的部件也引用了该文本，将会受到影响。

如果点击了下拉按钮，将从文本库中选择文本，如图 9-2。



图 9-2 文本选择窗口

表 9-1 为图 9-2 中的修改文本窗口中所有窗口中元素的说明。

表 9-1 文本选择窗口元素说明

界面元素	说明
ID	该文本条目的 ID 号
当前语言	工程当前浏览的语言
选择	为文本内容选择当前选中的条目
清空	如果文本内容是从文本库里选择的话，这个功能将清空从文本库的选择，用户又可以直接输入文本了。
取消	取消所做的修改，并关闭对话框
文本库	打开文本库，对文本库进行编辑

9.2 当前语言

当前语言是画面视图显示的语言，也就是说，如果当前语言改变了，则部件的标签文本也切换到显示相应的语言(如果这些文本引用了文本库的话)，这项设置对于没有引用文本库的文本是没有效的。

要改变当前语言，只需要如图 9-3 所示操作。



图 9-3 当前语言选择

9.3 文本库

文本库是用户预先输入的字符串的集合，部件的状态文本、静态文本、报警文本皆可以引用文本库的内容。字符串提供了三种语言，客户在编辑文本库时可以输入这三种语言。

文本库可以从“设置”-“文本库”的操作路径进入当前工程的文本库，如图 9-4 所示。



图 9-4 工程管理器中文本库的位置

单击“文本库”节点，打开工程的文本库编辑器，如图 9-5。



图 9-5 文件库窗口

表 9-2 详细说明了文本库窗体中元素的内容。

表 9-2 文本库窗口元素说明

界面元素	说明
ID	系统为文本库条目自动生成的索引号；
语言一	文本条目第一种语言的内容
语言二	文本条目第二种语言的内容

语言三	文本条目第三种语言的内容
添加	向文本库添加新的文本条目
删除	将选中的条目从文本库删除
修改	编辑选中的条目的内容
关闭	关闭对话框

点击添加或者修改，将进入文本编辑对话框，如图 9-6 所示。

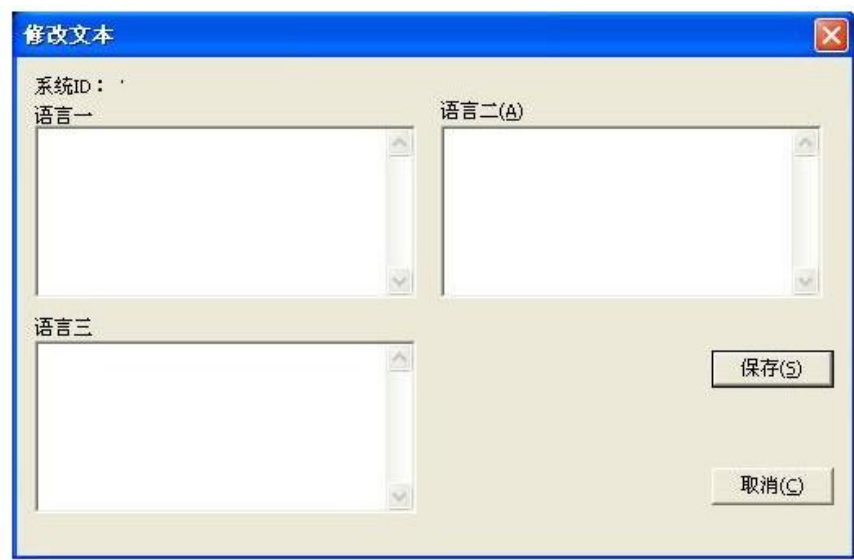


图 9-6 修改文本窗口

表 9-3 为图 9-6 的修改文本窗口中所有窗口中元素的说明。

表 9-3 修改文本窗口元素说明

界面元素	说明
系统 ID	系统为该条目自动生成的 ID 号
语言一	输入第一种语言的内容，支持回车，最多支持 256 个字符、128 个双字节字符
语言二	输入第二种语言的内容，支持回车，最多支持 256 个字符、128 个双字节字符
语言三	输入第三种语言的内容，支持回车，最多支持 256 个字符、128 个双字节字符
保存	保存当前修改
取消	取消所做的修改，回到文本库



# 10 报警

## 10.1 报警机制

### 软件报警机制

当人机界面加载工程时，就会加载工程中的报警信息到人机的报警缓冲区，然后不停地循环监控报警数据，如果符合报警条件，就产生报警信息，并将该信息暂存到报警的系统缓冲区，如果缓冲区已满，系统就会按照先进先出的原则进行溢出处理。

软件的新版本支持字报警和位报警，也支持将报警的历史记录保存到 CF 卡、SD 卡或者 U 盘。进行永久保存。

modbus tcp 支持将报警信息通过以太网上传到 PC 机。一旦发生报警，HMI 就将报警信息通过以太网发送到 PC。

### 一、工程设置

- 1. 选择 modBus TCP Slave 协议作为上传数据的协议。



## 2. 设置 HMI 名

【设置】/【工程参数】/【一般属性】/【高级设置】/【HMI 名】

注意：这个名字在所有需要上传报警记录的人机中必须唯一。长度不得超过 30 个字符（15 个汉字）。

当报警时候有个报警数据上传，当数据上传时“HMI 名”会在指定的上传地址显示出来。

HMI 名：HMI 名字 + 时间 + 报警信息 用来发送到指定地址（中间不需要“+”来连接）



## 3. 设置需要报警项

【工程设置】/【位（字）报警区】中的新建和修改对话框中，把【是否上传】勾上，并填写上传地址。注意，这个地址必须是 modBus TCP 的字地址。



HMI 名使用的设置案例效果



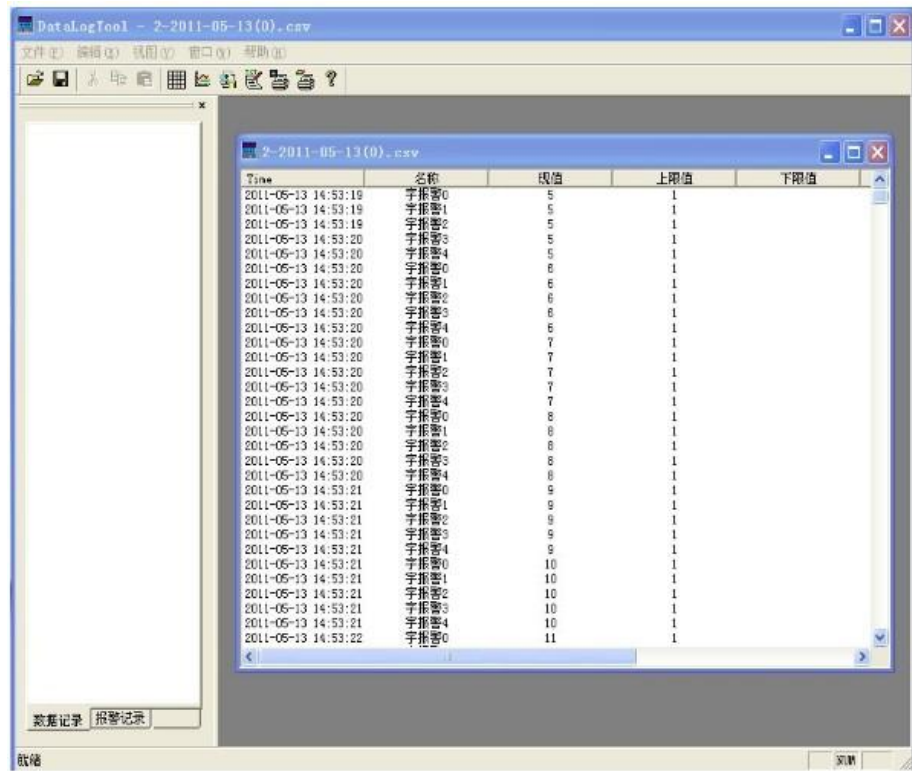
二、PC 端报警查看说明

(SD 卡和 CF 卡的报警数据)



当您想查看的 SD 卡和 CF 卡的报警数据时候，双击这个软件。注意当软件关闭时，报警数据是不会上传到 PC 端的。

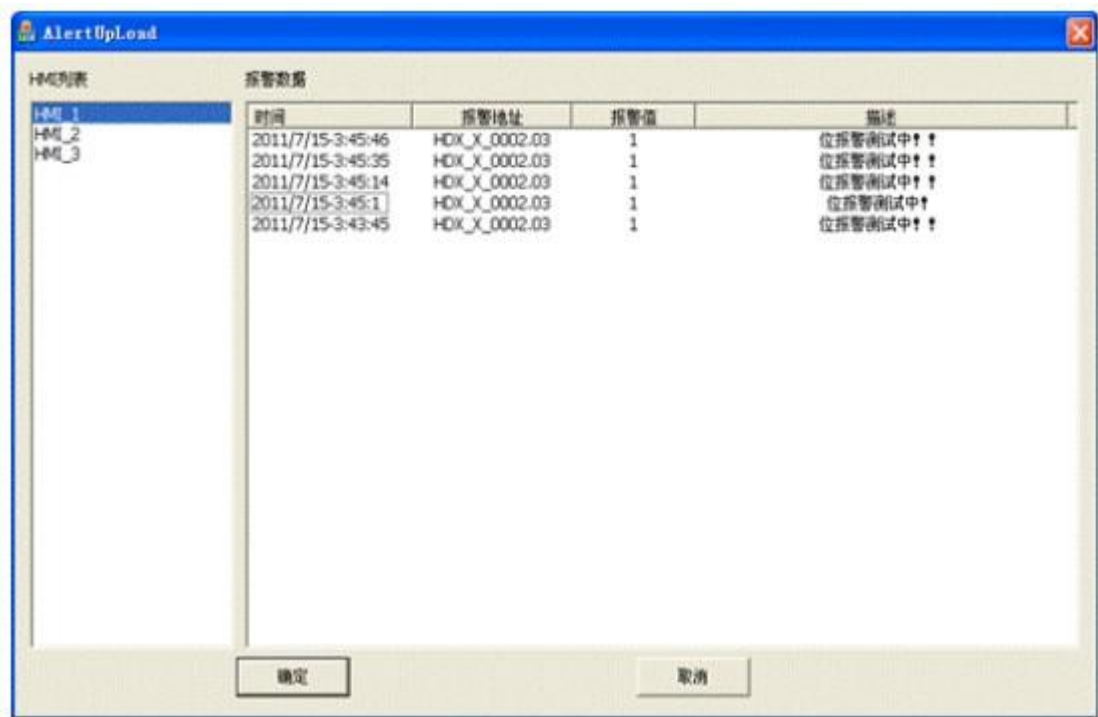
报警数据在软件的目录下的 log 文件夹中。建议您每隔一段时间备份一下数据。然后删除这个文件夹。Log 目录下的.dat 文件可以用 Excel 打开查看。



HMI 名上传的报警数据查看：



使用此软件<sup>AlertUpload.exe</sup>进行查看报警数据时,通讯参数等设置只能使用 Ethernet 和 ModBus Tcp 协议;使用 ModBus Tcp Slave 进行通讯时,在屏幕上显示 NC 属正常情况。



单机左边的 HMI 名即可查看相应的 HMI 的报警信息。

[illegible]

位报警信息编辑

位地址(A):  ...

☐ 是否记录

☐ 是否快速更新

☐ 不保存解除报警记录

内容

☐ 使用文本库(U)

文本(T):

☐ 报警时启用蜂鸣器

蜂鸣器频率:  持续时间:

☐ 是否上传

上传地址:  ...

上传高级配置:

触发条件

☒ On 时报警(N) ☐ Off 时报警(E)

通知

☐ 是否启用

触发地址:  ...

报警画面:  ☐ 弹出一

打开坐标: X  Y

标题栏:  ☐ 报警解除时关闭报警窗口

位地址:	报警监控的位地址。
是否记录:	如果选择是，则该条数据的报警历史将保存到 CF 卡、SD 卡或者 U 盘里。
是否上传:	上传到指定地址。
触发条件:	报警发生的条件。
内容:	报警发生后，显示的文本内容。
通知:	用来标记这个报警是否发生 了； 发生报警就会把通知的地址写 1 ， 消除报警了会把这个通知地址写 0。

报警画面：	点击右边下拉菜单可选择报警时是否跳出指定画面。如下图所示（注意：选择做的报警画面必须为子画面。）
-------	--

制作位报警的步骤：

- 1. 调出“位报警信息”框
- 2. 设置对应的位地址、触发条件、内容、报警显示的画面等相关信息。
- 3. 工程里放置相关的报警部件，如：跑马灯、报警表、报警记录等；同时放置相关的触发按钮，用来触发报警。

10.3 字报警区

字报警信息录入

设备名称：

☐ 是否记录

☐ 不保存解除报警记录

☐ 是否快速更新

一般

设备地址：

数据格式：

小数点位置：

报警描述：

☐ 使用文本库

文本库...

☒ 上限报警

☐ 阈值是变量

上限值：

上限寄存器：

☐ 下限报警

☐ 阈值是变量

下限值：

下限寄存器：

☐ 范围报警

☐ 范围之内报警

下限值：

下限寄存器：

阈值是变量

上限值：

上限寄存器：

☐ 等值报警

☐ 阈值是变量

等值：

等值寄存器：

☐ 报警时启用蜂鸣器

蜂鸣器频率：

持续时间：

报警画面：

☐ 弹出一次

通知

☐ 是否启用

触发地址：

☐ 是否上传

上传地址：

上传高级配置：

打开坐标：X  Y

标题栏：

☐ 报警解除时关闭报警窗口

软件的新版本支持字报警，字报警就是对一个设备字地址进行监控，如果数据满足了一定条件就会产生报警，软件支持的字报警类型有：

- 上限报警：**如果数据超过了上限值(可以是常数也可以是寄存器变量)，就会产生报警。
- 下限报警：**如果数据低于下限值(可以常数也可以是变量)，就会产生报警
- 范围报警：**如果数据超出了给定的数据范围(可以是常数，也可以是变量)，就会产生报警。

相关属性设置：

- 设备名称：**报警的条目名称
- 是否记录：**是否记录到 CF 卡，如果选“是”，则 HMI 必须配备 CF 卡才有效。
- 设备地址：**报警所监视的设备字地址，可以使用地址库。
- 数据格式：**可以是无符号、有符号、BCD 码，如果阈值是变量的话，那么也是按照一致的数据格式来比较的。

**小数点位置：**数据显示的小数点位置。

**报警描述：**报警的文本描述

**上限报警：**需要输入上限阈值或者阈值所在的寄存器。

**下限报警：**需要输入下限阈值或者阈值所在的寄存器。

**范围报警：**报警条件是范围报警，并要求输入相关的报警条件。

**报警画面：**功能可参照‘10.2 位报警区的录入与编辑’。

**通知：**用来标记这个报警是否发生 了； 发生报警就会把通知的地址写 1， 消除报警了会把这个通知地址写 0。

# 11 按钮开关

## 11.1 按键

- 1. 按键在部件栏的按钮开关中。
- 2. 按键的功能：按键是一种特殊的部件，输入部件的输入必须依赖键盘画面，而键盘画面必须有按键部件才能输入数据。
- 3. 按键的属性：

属性名称	属性分类	说明
控制键	是 否	是:该键输入的是控制字符否:该键输入的是 ASCII 字符
键值	控制键为是时有效, 四种类 型 ENTER/DELETE/ESC/CLEAR	ENTER:确认回车, 关闭键盘画面 DELETE:相当于 DEL 键, 删除游标前的字符; CLEAR:清空 HSW45 字里面的内容, 这是输入缓冲区; ESC:取消当前输入, 关闭键盘画面
	控制键为否时有效,	键值可以是任意一个的 ASCII 字符
部件等级密码	是:需输入密码才可使用 否:正常使用	是:设置部件等级密码否:不设置部件等级密码
等级自动降低	是 否	是:密码输入成功后, 密码等级降为一; 再次点击, 则输入等级一或更高等级的密码即可。 否:不改变部件密码等级。
部件操作等级		设置部件的密码等级。部件的密码设置在“工程参数”的部件等级密码选项下设置。

## 11.2 字开关



1. 字开关在部件栏的“按钮开关”中。

2. 字开关的功能：

当用户操作字开关时，该部件监控的字地址的值会根据字开关类型而发生变化。

3. 字开关的分类：

**字设置：**为字地址所指向的字设置一个常数；

**递加：**每次按下字开关时，字地址所指向的字的值递增一次，递增幅度是“操作数”的值；

**递减：**每次按下字开关时，字地址所指向的字的值递减一次，递减的值是“操作数”的值；

### 字开关属性表

属性名称	属性分类	说明
(读取/写入) 字地址	无	字开关读取或写入的 PLC 或 HMI 内部寄存器地址
操作字数	单字 双字	单字：字开关操作一个字； 双字：字开关操作两个字；
数据格式	BCD 无符号十进制 有符号十进制	BCD：以 BCD 码的方式显示或者写入数据； 无符号十进制：以无符号 BIN 码的方式显示或者写入数据； 有符号十进制：以有符号 BIN 码的方式显示或者写入数据；
字开关类型	递减 递加 字设置	有以下几种类型： 字设置：为字地址所指向的字设置一个常数； 递加：每次按下字开关时，字地址所指向的字的值递增一次，递增幅度是“操作数”的值； 递减：每次按下字开关时，字地址所指向的字的值递减一次，递减的值是“操作数”的值；
操作数	一个常数值,对于不同的字开关类型,其含义也不一样	递减：递减幅度； 递加：递增幅度； 字设置：往字地址写的常数值
上/下限	当字开关类型设为递加或递减时,此项才可用.用于设置最大值和最小值.	仅当字开关类型为递加、递减时有效； 当开关类型为“递加”时该常数就是递加上限； 当开关类型为“递减”时该常数就是递减下限。
显示方向	向量图和标签的现实方向	自上而下、自左而右、自下而上，自右而左
扩展	无	扩展功能的设置
需要确认	否 是	否：不需要确认，直接将用户操作写入 PLC。 是：当用户操作字开关，HMI 将弹出确认窗口，需要用户再次确认后，HMI 才会将值得改变写入到 PLC。



最长等待时间	当需要确认为“是”时有效	当“需要确认”为“是”时，有效；单位：秒，表示在此时间内如果等不到用户的确认，确认窗口将消失。
互锁	否 是	否：不使用互锁 是：使用互锁地址
互锁地址	当互锁为“是”时有效	当“互锁”为是时有效；若使用“互锁地址”，仅当互锁位地址所指向位的值为 1 时，位开关才可用。
触摸屏可用	是 否	否：不允许使用 是：允许使用位开关设置功能

### 11.3 位开关

1. 位开关在部件栏的按钮开关中。

2. 位开关的功能：

对指定的 PLC 或者 HMI 位地址进行操作、监控和显示；当位开关被按下时，操作地址的位值将按照开关类型所描述的那样变化。

当“监测”可用时，位开关所显示的状态是监控地址所指向的位的状态。

例如：如果监控地址所指的位的值为 1，则位开关将显示开状态的标签、位图和向量图；反之，则显示关状态的标签、位图和向量图。

3. 位开关的分类：

**OFF**：将操作地址所指的位置 0；

**ON**：将操作地址所指的位置 1；**复位型**：当位开关按下时，操作地址所指的位被置为 1，弹起时又被置为 0；

**切换型**：每次按下位开关时，操作地址所指的位的值将被切换一次（在 0 和 1 之间来回切换）。

4. 位开关的相关属性如下：

属性名称	属性分类	说明
操作地址	PLC 或 HMI 内部寄存器地址	位开关操作的 PLC 或 HMI 内部寄存器地址
开关类型	OFF ON 复位型 切换型	OFF：将操作地址所指的位置 0； ON：将操作地址所指的位置 1； 复位型：当位开关按下时，操作地址所指的位被置为 1，弹起时又被置为 0； 切换型：每次按下位开关时，操作地址所指的位的值将被切换一次（在 0 和 1 之间来回切换）。
复位延时（毫秒）	当开关类型设为复位型时，此项才可用	当设有延时，即按下次开关时，等待所设置的时间过后才执行复位。

		(例如:复位延时中设为 5, 即当按下此按键, 要等 5 毫秒后才执行复位功能)
监测	是否	是: 位开关具有监视功能(位开关将监测监视地址所指的位); 位开关的显示将反映监控地址位的状态。 否: 位开关没有监视功能;
监控地址	当监测设为是时, 此项才可用	位开关的状态显示将依赖监控地址所指的位 (该地址可以和操作数地址一样, 也可以不一样)
文字对齐	当后面的“文本内容”有内容时, 此项才可用	标签文本的对齐方式 自定义: 随意拖动标签的位置 居中: 标签在部件内居中显示 居左: 标签在部件内居左显示 居右: 标签在部件内居右显示 该设置对所有状态的标签均有效
标签起始点	无	仅当文字对齐为自定义时有效, 可以通过拖动确定标签显示位置。
是否反向显示	否 是	否: “否”为默认的, 正常显示。 是: 反向显示该部件的状态图和文本
是否隐藏	否 是	是: 隐藏该部件 否: “否”为默认的
触发地址	当前面的“是否隐藏”为“是”时 此项才可用;	填写触发隐藏的位地址
隐藏方式	当前面的“是否隐藏”为“是”时, 此项才可用	否: 隐藏触发位为 0 时隐藏 是: 隐藏触发位为 1 时隐藏
最小按压时间	单位: 毫秒	按压规定的时间后, 才对该地址进行对应的操作。
需要确认	否 是	否: 不需要确认, 直接将用户操作写入地址位。 是: 当用户操作位开关, HMI 将弹出确认窗口, 需要用户再次确认后, HMI 才会将值的改变写入到地址位。
最长等待时间	当需要确认为“是”时有效	当需要确认为“是”时, 有效。 单位: 秒, 表示在此时间内如果等不到用户的确认, 确认窗口将消失。
互锁	否/是	是: 使用互锁地址; 否: 不使用互锁;

互锁地址	当互锁为是时有效	若使用互锁地址，仅当互锁位地址所指向的位为 1 时，位开关才可用。
触摸屏可用	是/否	是：触摸该部件有效 否：触摸该部件无效
状态		开 关
使用蜂鸣器	是/否	是：触摸该部件蜂鸣 否：触摸该部件无蜂鸣
文本	部件显示的文本内容	输入的是当前语言、当前状态下的文本。对于多语言，多状态显示，需要更改语言和状态，然后再进行文本设置。

## 11.4 滑动开关

1. 滑动开关在部件栏的按钮开关里；

2. 滑动开关的功能：

通过推动滑块可改变滑动开关的读写地址的值，同时可以反应地址的值的变化情况；

3. 滑动开关格式说明：（参考数字输入/显示）

共 4 种：1. 无符号十进制（单双字），2. 有符号十进制（单双字），3. BCD 码（单双字），4. 32 位浮点数（默认双字）；

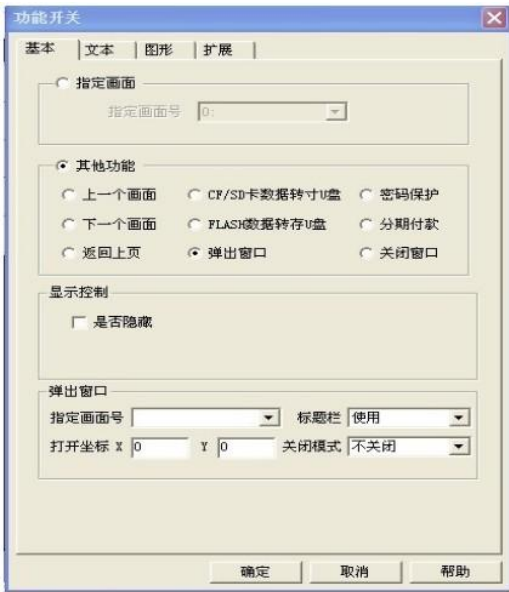
4. 滑动开关主要属性如下表：

属性名称	属性分类	说明
读写地址	PLC 或 HMI 内部寄存器地址	所监控的地址
操作字数	单字/双字	单字：显示或者输入只涉及一个字 双字：涉及以属性“字地址”为首的两个字
数据格式	BCD 浮点数 无符号十进制 有符号十进制	无符号十进制：以无符号 BIN 码的方式读出数据； BCD：以 BCD 码的方式读出数据； 有符号十进制：以有符号 BIN 码的方式读出数据； 32 位浮点：以标准浮点格式读出数据
小数位数	常量	
最小刻度	常量	设置最小刻度，当数据为浮点型是无效
显示方向	从上到下 从左到右 从下到上 从右到左	设置递增方向/显示方向

数据范围是变量	是 否	如为否，则数据范围是常数；如为是，则数据范围是变量
上限	常量	“数据范围是变量”为否时有效
下限	常量	同上
存储上限值地址	PLC 或 HMI 内部寄存器地址	“数据范围是变量”为是时有效
存储下限值地址	PLC 或 HMI 内部寄存器地址	同上
卷动模式	是 否	设置是否每次移动固定的值
卷动值	常量	“卷动模式”为是时有效
边框是否透明	是 否	是否设置表框为透明
边框颜色	色彩	“边框是否透明”为否时有效
背景是否透明	是 否	是否设置背景为透明
背景颜色	色彩	“背景是否透明”为否时有效
滑轨是否透明	是 否	是否设置滑轨为透明
滑轨颜色	色彩	“滑轨是否透明”为否时有效
滑块宽度	常量	设置滑块的宽度
滑块颜色	色彩	设置滑块的颜色
X 坐标	常量	滑动开关起始点横坐标
Y 坐标	常量	滑动开关起始点纵坐标
位置-宽	常量	滑动开关的宽
位置-高	常量	滑动快关的高

## 11.5 功能开关

1. 功能开关在部件栏的图形中。
2. 功能开关的功能：功能开关用来切换 HMI 上的基本画面。
3. 功能开关的属性表



属性名称	属性分类	说明
指定画面号	是否	是：切换到指定的画面 否：选择特定的功能选项
功能选择	当“指定画面号”为“否”时有效： 上一画面 下一画面 弹出窗口 密码保护 数据转存 分期付款 返回上页 FLASH 转存 U 盘 关闭窗口	上一画面：在基本画面列表中相邻的上一画面 下一画面：在基本画面列表中相邻的下一画面 (特别说明，切换画面仅对基本画面有效) 弹出窗口：点击弹出指定子画面，子画面的关闭需要用该部件的【关闭窗口】来关闭。 密码保护：当工程保护方式选择为 按键密码保护 时，用于输入密码，以解开屏幕保护。 数据转存：将保存在 CF 卡或是 CD 卡的数据转存到 U 盘 分期付款：按下功能键跳转到输入分期付款的密码画面 返回上页：返回到原来的画面 FLASH 转存 U 盘：将保存在 flash 的数据转存到 U 盘 关闭窗口：关闭指定子画面
画面编号	当“指定画面号”为“是”时有效	功能开关要切换到基本画面号 (注意，该编号所代表的画面不能是本部件所在画面，否则将无法切换)
子画面编号	功能选择为：弹出窗口 时有效	设置弹出的子画面
弹出窗	功能选择为：弹出窗口 时	设置弹出的子画面的左上角坐标

口坐标	有效	
不使用标题栏	功能选择为：弹出窗口 时有效	是：不使用标题栏 否：使用标题栏
随父窗口关闭	功能选择为：弹出窗口 时有效	是：父窗口关闭时，弹出的子窗口也自动关闭。 否：父窗口关闭时，弹出的子窗口不关闭。
是否隐藏	是 否	是：隐藏该部件 否：默认状态
触发地址	当“是否隐藏”为“是”时有效	填写触发隐藏的位地址
隐藏方式	当“是否隐藏”为“是”时有效	否：隐藏触发位为 0 时隐藏 是：隐藏触发位为 1 时隐藏
互锁	是 否	是：使用互锁地址； 否：不使用互锁
互锁地址	当“互锁地址”为“是”时有效	若使用互锁地址，仅当互锁位地址的值为 1 时，位开关才可用
是否不显示“锁”图	是 否	是：不显示锁的图标 否：显示锁的图标
部件等级密码	是：输入密码后，才能使用该部件 否：正常使用	是：设置部件等级密码 否：不设置部件等级密码
等级自动降低	是 否	是：密码输入成功后，密码等级降为一；再次点击，则输入等级一或更高等级的密码即可。 否：不改变部件密码等级。
弹出窗口	指定画面/标题/坐标/关闭模式	弹出指定的画面，标题使用与否、窗体关闭设置：不关闭、随父窗口关闭
部件操作等级		设置部件的密码等级。部件的密码设置在“工程参数”的部件等级密码选项下设置。

## 11.6 配方传输

1. 配方传输在在部件栏的图形中。

2. 配方传输的功能：

配方传输部件可以将 HMI 上已有的配方下载到 PLC，也可以从 PLC 上载配方数据到 HMI。并保存在 HMI 上。

配方传输部件无论是上传还是下载，都必须依赖软件保留寄存器 HPW0，HPW0 指明了 HMI 配方表的当前组号，上传时从 PLC 上传配方数据到该组，并覆盖掉 HMI 配方列表中该组的数据，下载时从 HMI 下载该组的配方数据到 PLC。

关于工程配方数据的编辑，请参考配方一章。

### 3. 配方传输的属性：

属性名称	属性分类	说明
传输方式	上传 下载	传输方式的选择： <b>上传：</b> 从 PLC 上传配方数据到 HMI，并保存在 HMI 上，HPW0 记录了 HMI 上的当前配方组，上传后的配方数据将保存在该组上，并覆盖原来的配方数据。 <b>下载：</b> 从 HMI 配方表下载配方数据到 PLC，HPW0 的值，就是要下载至 PLC 的配方组号。
部件等级密码	是：输入密码后，才能使用该部件 否：正常使用	是：设置部件等级密码 否：不设置部件等级密码
等级自动降低	是 否	是：密码输入成功后，密码等级降为一；下次再点击这个部件，则输入等级一或更高等级的密码即可。 否：不改变部件密码等级。
部件操作等级		设置部件的密码等级。部件的密码设置在“工程参数”的部件等级密码选项下设置。

## 11.7 超级组合按钮

1. 超级组合按钮在部件栏的按钮开关中。

2. 超级组合按钮的功能：超级组合按钮将多种操作整合到一个开关来触发。具体的功能包括置位、复位、位切换、线圈复制、画面跳转、设置数据、寄存器复制、上传下载配方以及做简单的四则运算。

### 3. 超级组合按钮的功能设置

功能名称	说明
置位线圈	指定位地址写为 1
复位线圈	指定位地址写为 0
线圈取反	指定位地址取反
线圈复制	将源位地址的值写入目标位地址
画面跳转	画面跳转到指定画面

设置数据	给目标字地址写数值
寄存器复制	将指定长度的寄存器源地址的各个值复制到目标地址里
下载配方	配方下载
上传配方	配方上传
四则运算	<p>加：操作地址的值（常量）加上被操作地址的值（常量）的结果赋给目标地址</p> <p>减：操作地址的值（常量）减去被操作地址的值（常量）的结果赋给目标地址</p> <p>乘：操作地址的值（常量）乘以被操作地址的值（常量）的结果赋给目标地址</p> <p>除：操作地址的值（常量）除以被操作地址的值（常量）的结果赋给目标地址</p> <p>求余：操作地址的值（常量）除以被操作地址的值（常量）的余数赋给目标地址</p> <p>次方：操作地址的值（常量）为底数，被操作地址的值（常量）为次数，求得的次方结果赋给目标地址</p>
部件等级密码	<p>是：设置部件等级密码</p> <p>否：不设置部件等级密码</p>
等级自动降低	<p>是：密码输入成功后，密码等级降为一；下次再点击这个部件，则输入等级一或更高等级的密码即可。</p> <p>否：不改变部件密码等级。</p>
部件操作等级	设置部件的密码等级。部件的密码设置在“工程参数”的部件等级密码选项下设置。
定时置位	在指定的时间里对指定的位地址写 1
定时复位	在指定的时间里对指定的位地址写 0
定时触发字	在指定的时间里对指定的字地址操作，如对指定的字地址写值。



## 12 数字输入/显示

### 12.1 数字输入/显示

1. 数字输入/显示在部件栏的数字输入/显示中。

2. 数字输入/显示的功能：

数值输入/显示以指定的格式输入/显示设备数据，或者以指定的格式显示设备的数据或者往设备写数据，可以对读取写入的数据进行缩放。

3. 数值显示格式说明：

二进制：数据显示为二进制(0~1)格式，小数点选项被忽略，但可以设定要显示的位数。

例如：二进制(110011)，它的显示位数为三位，那么最终将显示为(011)，即显示最后的几位。显示位数对数据的输入没有影响。

八进制：数据显示为8进制(0~7)格式，小数点选项被忽略，但可以设定要显示的位数。例如：八进制(1234567)，它的显示位数为五位，那么最终将显示为(34567)，即显示最后的几位。显示位数对数据的输入没有影响。

十进制：数据显示为十进制(0~9)格式；显示位数对数据的输入没有影响。

小数点选项将根据下面两种不同的情况起不同的作用：

如果“是否缩放”为“否”，那么显示的数据将是原始数据，比如从设备读取的数据是123456，而小数点位置2.1，那么数据将最终显示45.6。

如果“是否缩放”为“是”，那么显示的数据为经过增益和偏移运算后的数据，运算公式为：

显示值=原始数据\*比例增益+偏移，仅当数据为十进制或者浮点时有效。

然后根据小数点位置转换为最终显示的数据：

例如：经过比例增益计算后的数据为1234.56，而小数点位置为4.1，那么最后显示为1234.5

对于输入数据，属性“小数点位置”将被忽略，其计算公式如下：

写入值=(输入值-偏移)/增益

十六进制：数据显示为16进制(0~F)格式，小数点选项被忽略，但可以设定要显示的位数。

例如：十六进制（ABCDEF），它的显示位数为五位，那么最终将显示为（BCDEF），即显示最后的几位。**显示位数对数据的输入没有影响。**

**BCD 码：**除了按照 BCD 的格式输入或读取外，其余操作与十进制是一致的。

**32 位浮点数：**(单精度浮点数, 默认双字):控制器中的数据从 32 位 IEEE 浮点格式转换为十进制数据并显示出来，32 位浮点数（单精度浮点数格式）如下图

单精度浮点数格式( 32 位)		
S	e[30:23]	f[22:0]
3 30		22
1 23		0
单精度位模式		值
0< e < 255		$(-1)^S \times 2^{e-127} \times 1.f$
e=0 ; f!= 0		$(-1)^S \times 2^{-126} \times 0.f$
e=0; f=0		$(-1)^S \times 0.0$
s=0, e=255, f=0		正无穷大
s=1; e=255; f=0		负无穷大
s=0 or 1; e=255; fl=0		无效数

**间接读寻址：**是指该部件最终读的地址由间接读寻址地址来决定。间接读寻址地址最多可以有三个。先以一个例子来说明。

数值输入显示部件的读取地址为：F11. 设置间接读寻址地址三个，分别为 D0，E1，HDW2。系统先读 HDW2 地址的值，如它的值为 22，则第二个间接读寻址地址 E1 更新为 E23，然后系统再读 E23 地址的值，如它的值为 33，则第一个间接读寻址地址 D0 更新为 D33. 再读 D33 地址的值为 44，则部件的读取地址由 F11 更新为 F55，最终读取的地址为 F55。

间接读寻址地址为两个，则类似，第二个寻址地址的值和第一个寻址地址的寄存器组合成一个新的地址，系统读这个新地址的值再和读取地址的寄存器组合生成新地址，读取该地址的值显示出来。

间接读寻址地址为一个，则直接读寻址地址的值和读取地址的寄存器组合生成新地址，读取该地址的值显示出来。

**间接写寻址**的机制和间接读寻址的机制一样。这样不再做介绍。

4. 数字输入/显示的属性：

属性名称	属性分类	说明
读取地址	PLC 或 HMI 内部寄存器地址	部件显示的数据将从这里读取
操作字数	单字 双字	<b>单字：</b> 显示或者输入只涉及一个字  <b>双字：</b> 涉及以属性字地址为首的连续的两个字
快速更新	否	以较高的频率更新显示设备的数据(大约比普通部件的更新

	是	频率高出一倍)
允许输入	否 是	<b>是:</b> 当部件获得输入焦点时, 弹出键盘画面, 完成数据输入, 前提是键盘画面存在。 <b>否:</b> 该部件只用于显示指定地址的数据
写入地址	当允许输入为是时有效 PLC 或 HMI 内部寄存器地址	当 <b>允许输入</b> 为是时有效。 输入的数据将往该地址写入。 如果 <b>操作地址</b> 为空, 且部件允许输入, 则操作地址将和读取地址一致。
键盘画面号	当允许输入为是时有效	当 <b>允许输入</b> 为是时, 有效; 键盘画面的几个条件: 1、画面存在; 2、该画面是子画面; 3、含有按键部件 <b>内置数字键盘:</b> 内置数字键盘是 HMI 的标准数字键盘, 该键盘只能输入十进制和浮点数据。
键盘弹出位置	居中 中上 智能 中下 左上 右上 左中 右中 左下 右下	当有用到键盘时, 此设置才有效, 这个是设定键盘位置
不显示此部件	否 是	否: 显示部件 是: 不显示部件
高位补零	否 是	否: 高位不补零 是: 高位补零
是否显示*号	否 是	否: 输入的数值直接显示 是: 输入的数值以* 号显示
是否闪烁	否 是	否: 数值不闪烁 是: 数值在满足闪烁类型的条件下闪烁
闪烁触发地址		触发闪烁的位地址
闪烁类型	当“是否闪烁”为“是”时有效, 不闪烁为 ON 时闪烁	不闪烁: 数值不闪烁 为 ON 时闪烁: 当触发位地址为“ON”时数值闪烁 为 OFF 时闪烁: 当触发位地址为“OFF”时数值闪烁

	为 OFF 时闪烁 为 ON 时交替闪烁 为 OFF 时交替闪烁	为 ON 时交替闪烁: 当触发位地址为“ON”时数值默认颜色和闪烁颜色交替闪烁 为 OFF 时交替闪烁: 当触发位地址为“OFF”时数值默认颜色和闪烁颜色交替闪烁
闪烁频率 (100ms)	无	设置闪烁的速率, 例如里面填 5, 表示的是闪烁的速率为 500ms
闪烁颜色	色彩	无
零值不显示	是 否	否: 显示 是: 不显示
是否间接读寻址	是 否	是: 读取地址做寻址处理 否: 直接读取
间接读寻址地址		最多可以输入三个寻址地址。从最后一个寻址地址开始读取, 依次向前。
是否间接写寻址	是 否	是: 写入地址做寻址处理 否: 直接写入
间接写寻址地址		最多可以输入三个寻址地址。从最后一个寻址地址开始读取, 依次向前。
数据格式	二进制 八进制 无符号十进制 十六进制 BCD 码 32 位浮点数 有符号十进制	集成了六种显示格式  <b>二进制:</b> 以二进制的格式读取(写入)  <b>八进制:</b> 以八进制的格式读取(写入)  <b>无符号十进制:</b> 以无符号十进制的格式读取(写入)  <b>十六进制:</b> 以十六进制的格式读取(写入)  <b>BCD 码:</b> 以 BCD 码的格式读取(写入)  <b>32 位浮点数:</b> 以 32 位浮点的标准格式读取(写入), 操作字数为双字。  <b>有符号十进制:</b> 以有符号十进制的格式读取(写入)
小数点位置	无	当且仅当 <b>数据格式</b> 是 <b>浮点</b> 和 <b>十进制</b> 时, 显示数值的小数点数位有效
数据范围	无	数据显示的范围, 超出该范围的数据将无法输入。
是否缩放	否 是	<b>是:</b> 读出来的数据可以用比例增益和偏移来运算

		<b>否</b> ：读出来的数据以原始数据显示，不经过缩放。
比例增益	当是否缩放为是时有效	当 <b>是否缩放</b> 为是时有效 显示值=PLC 值*比例增益 + 偏移，仅当数据为十进制或者浮点时有效
偏移	当是否缩放为是时有效	当 <b>是否缩放</b> 为是时有效 显示值=PLC 值*比例增益 + 偏移，仅当数据为十进制或者浮点时有效
互锁	当允许输入为是时有效	当 <b>允许输入</b> 为是时有效。  <b>是</b> ：只有当互锁地址的值为 ON，才可以输入  <b>否</b> ：不对输入操作进行互锁
互锁地址	当允许输入为是时有效	仅当 <b>允许输入</b> 、 <b>互锁</b> 皆为 <b>是</b> 时，有效。 如果该地址值为 ON，则部件可以接受用户输入，反之，则禁止用户输入
触发方式	无触发 写入后触发 写入前触发 写入后触发并复位 写入前触发并复位 写入前触发写入后复位	触发方式有六种：  <b>无触发</b> ：没有任何触发方式  <b>写入后触发</b> ：在输入数据完成后[点击键盘的 ENT 键]，触发位置为 1  <b>写入前触发</b> ：在输入第一个数值时，触发位置为 1；  <b>写入后触发并复位</b> ：在输入数据之后[点击键盘的 ENT 键]，触发位置为 1 后再写为 0  <b>写入前触发并复位</b> ：在输入第一个数值时，触发位置为 1 后再写为 0  <b>写入前触发写入后复位</b> ：在输入数据时，触发位置为 1，点击键盘的 ent 键，触发位置为 0.
触发地址	PLC 或 HMI 内部寄存器地址	触发的位地址
键盘触发位	键盘无触发 键盘打开前触发	触发方式有六种：

	键盘关闭前触发 键盘打开前触发并复位 键盘关闭前触发发并复位 键盘弹出触发关闭前发复位	<b>键盘无触发：</b> 没有触发方式  <b>键盘打开前触发：</b> 键盘打开前将触发位置为 1  <b>键盘关闭前触发：</b> 键盘关闭前将触发位置为 1  <b>键盘打开前触发并复位：</b> 键盘打开前将触发位置为 1，然后置为 0  <b>键盘关闭前触发发并复位：</b> 键盘关闭前将触发位置为 1，然后置为 0  <b>键盘弹出触发关闭前发复位：</b> 键盘弹出时将触发位置为 1，关闭键盘前将触发位置为 0.
触发地址	PLC 或 HMI 内部寄存器地址	PLC 或 HMI 内部寄存器地址
部件等级密码	是：输入密码后，才能使用该部件 否：正常使用	是：设置部件等级密码 否：不设置部件等级密码
等级自动降低	是否	是：密码输入成功后，密码等级降为一；下次再点击这个部件，则输入等级一或更高等级的密码即可。 否：不改变部件密码等级。
部件操作等级		设置部件的密码等级。部件的密码设置在“工程参数”的部件等级密码选项下设置。

## 5. “数值输入/显示”的缩放功能

这个图 12-1 的动画设置（也就是移动缩放功能的设置）就是十五章移动图形的缩放和移动的两者结合。

之所以“移动图形”部件的两个设置是分开的，是为了和原有的“移动图形”兼容，而又不重复移动功能。（[详见十五章移动图形缩放功能篇](#)）



如图 12-1 “数值输入/显示” 部件的移动缩放设置

“数值输入/显示” 部件的设置和使用和“移动图形”唯一不同的地方在于移动类型的地址不再控制位图状态，只控制移动。

例如，若移动类型的起始地址为 HDW10，移动类型为“XY 轴移动”，数据类型是“无符号十进制”，属于单字，那么 HDW10 控制 X 轴移动，HDW11 控制 Y 轴移动。

12.2 文本输入/显示

- 1. 数字输入/显示在部件栏的数字输入/显示中。
- 2. 文本输入/显示是以 ASCII 字符格式来显示(写入)PLC 的数据，它可以显示(写入)一个字的数据，也可以显示(写入)多个连续字的数据。

对于一个字而言，存在低字节的字符码显示在左边，存在高字节的则显示在右边，对文本输入也是一样的；

如果允许部件输入，则必须为部件指定**键盘画面(如下图)**；

**间接读寻址：**是指该部件最终读的地址由间接读寻址地址来决定。间接读寻址地址最多可以有三个。先以一个例子来说明。

文本输入显示部件的读取地址为：F11。设置间接读寻址地址三个，分别为 D0，E1，HDW2。系统先读 HDW2 地址的值，如它的值为 22，则第二个间接读寻址地址 E1 更新为 E23，然后系统再读 E23 地址的值，如它的值为 33，则第一个间接读寻址地址 D0 更新为 D33. 再读 D33 地址的值为 44，则部件的读取地址由 F11 更新为 F55，最终读取的地址为 F55。

间接读寻址地址为两个，则类似，第二个寻址地址的值和第一个寻址地址的寄存器组合成一个新的地址，系统读这个新地址的值再和读取地址的寄存器组合生成新地址，读取该地址的值显示出来。

间接读寻址地址为一个，则直接读寻址地址的值和读取地址的寄存器组合生成新地址，读取该地址的值显示出来。

间接写寻址的机制和间接读寻址的机制一样。这样不再做介绍



3. 文本输入/显示的属性表

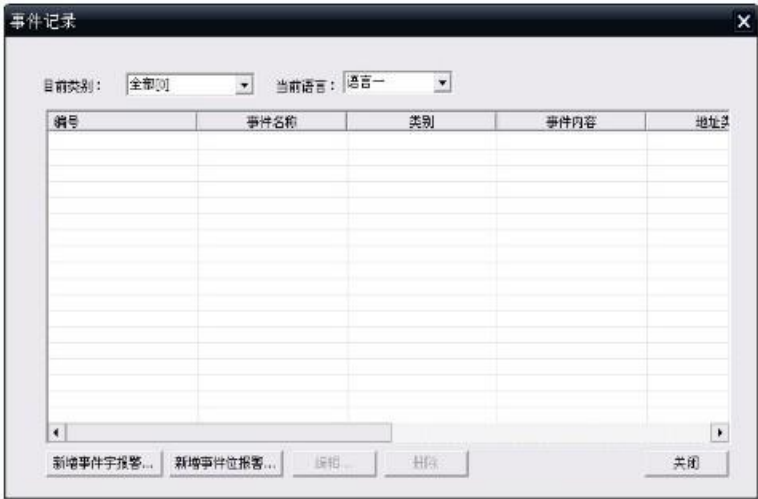
属性名称	说明
一般	部件的一般属性
描述	部件的注释
轮廓	部件最小外包矩形的位置、大小
读取地址	部件读取文本数据的首地址
操作字数	文本输入/显示的连续显示(写入)的字的个数， 最多支持 32 个字
快速更新	以较高的频率更新显示设备的数据(大约比普通部件的更新频率高出一倍)
允许输入	是：允许写入数据 否：不允许写入数据
写入地址	仅当“允许输入”为“是”时有效。 部件写入文本数据时的设备地址，如果“写入地址”为空，则表示，写入地址与读取地址一致。
键盘画面号	仅当“允许输入”为“是”时，才有效； 指定键盘画面编号，键盘画面必须是子画面，且含有按键部件 “内置字符键盘”：使用 HMI 内置的字符键盘，内置字符键盘可以输入从 1-10，a-z，A-Z 的所有 ASCII 字符。
向量图	部件的向量图外观，从向量图库选择。
边框颜色	向量图的边框颜色选择，如果无向量图或者向量图不支持，该属性将不可用。



前景色	数据显示区域的文本颜色
背景色	数据显示区域的背景色
字体	状态文本所采用的字体内容
文本对齐	<p>标签文本的对齐方式</p> <p><b>居中：</b> 标签在部件内居中显示</p> <p><b>居左：</b> 标签在部件内居左显示</p> <p><b>居右：</b> 标签在部件内居右显示</p>
扩展	扩展属性
互锁	<p>当“允许输入”为“是”时，有效；</p> <p><b>否：</b> 不对输入操作进行互锁</p> <p><b>是：</b> 只有当互锁地址的值为 ON，才可以输入</p>
互锁地址	<p>仅当“允许输入”和“互锁”皆为“是”，有效。</p> <p>当互锁地址内容为 ON 时，部件才可以接受输入，否则，将不接受输入。</p>
触发方式	<p>触发方式有三种：</p> <p><b>无触发：</b> 没有任何触发方式</p> <p><b>写入前触发：</b> 在系统弹出输入键盘时，将触发位置为 ON；</p> <p><b>写入后触发：</b> 在完成 PLC 寄存器写数据之后[点击键盘的 ENT 键]，将触发位置为 ON</p>
是否显示*号	是/否
触发地址	仅当触发方式为“写入前触发”与“写入后触发”有效。被触发的位地址

13.1 事件记录显示

- 1. 事件记录显示部件在报警显示中。
- 2. 事件记录显示的功能：  
    将所有事件按照所设定的显示相应的事件。
- 3. 事件记录显示的属性：



事件记录主界面

类别：

名称：

是否快速更新

一般

设备地址：

...

数据格式：

无符号十进

小数点位置：

4.0

...

报警描述：

使用文本库

文本库...

上限报警

阈值是变量

上限值：

上限寄存器：

...

下限报警

阈值是变量

下限值：

下限寄存器：

...

范围报警

阈值是变量

范围之内报警

下限值：

下限寄存器：

...

上限值：

上限寄存器：

...

事件确认时写入

是否启用

写入值：

等值报警

阈值是变量

等值：

等值寄存器：

...

报警时启用蜂鸣器

蜂鸣器频率：

低

持续时间：

短

通知

是否启用

触发地址：

...

是否上传

上传地址：

...

上传高级配置：

编辑

报警画面：

无

弹出一次

打开坐标：

X

0

Y

0

标题栏：

使用

报警解除时关闭报警窗口

确定

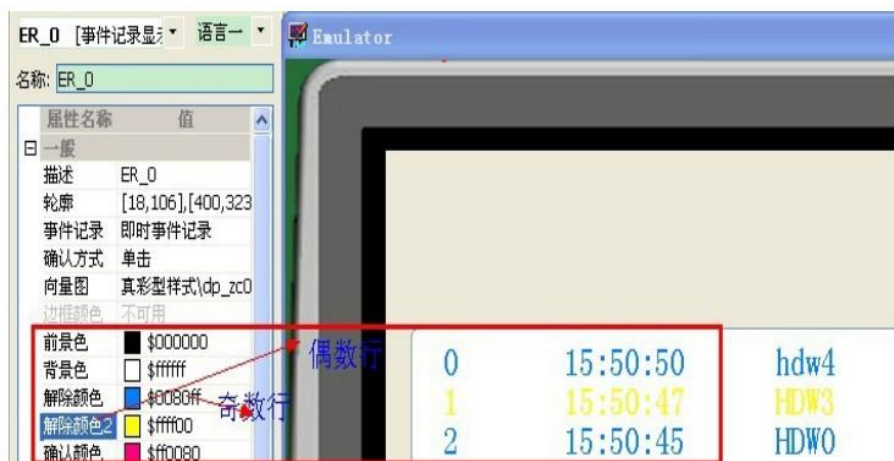
取消

事件记录里面的“字事件记录”界面



事件记录里面的“位事件记录”界面

4. 解除的颜色：可分为奇数行和偶数行。解除 1 的颜色为奇数行解除报警时的颜色。解除 2 的颜色为偶数行解除报警时的颜色。（如下图）



### 13.2 报警记录显示

1. 报警记录显示在部件栏的报警显示中。
2. 报警记录显示的功能:

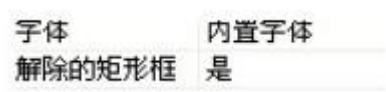
可以根据客户的需要，查询某个时间段的所有报警记录，然后按照时间顺序以表格方式显示出来。  
在 HMI 上可以动态设定查询时间段，并且可以实现掉电保存，如下图所示：



属性

属性名称	属性分类	说明
报警类型	位报警 字报警	<b>位报警</b> :显示的报警记录为位报警记录 <b>字报警</b> :显示的报警记录为字报警记录
起始时间	无	显示的是从这个时间点开始采集的数据记录，这个时间格式是：年/月/日/时/分/秒。 这个时间在部件第一次运行时有效，之后，HMI 可以动态保存用户设置的查询时间段。
跨度	无	表示部件将显示多少数据。跨度最长 31 天，也就是部件垂直滚动条的拖动范围，格式是：日/时/分/秒。 这个时间在部件第一次运行时有效，之后，HMI 可以动态保存用户设置的跨度查询时间段。
查询时间	无	上次设置的时间 当前时间 当前时间往前 1 小时 当前时间往前 1 天

3. 实时报警，报警时候不显示确认按钮



设置为是与否的效果如下：



上图是设置为“是” 下图设置为“否” 的效果

13.3 报警走马灯

- 1. 报警走马灯在部件栏的报警显示中。
- 2. 报警走马灯的功能：  
以走马灯的方式显示 HMI 报警缓冲区所有已发生的，但还没有解除的报警。
- 3. 报警走马灯的属性：

属性名称	属性分类	说明
显示日期	否 是	是：报警历史表中是否显示报警发生和报警解除的日期 否：不显示日期
日期格式	当显示日期为是时有效	当显示日期为是时有效，日期的显示格式有下面几种选择 yy/mm/dd：年/月/日 mm/dd/yy：月/日/年 dd/mm/yy：日/月/年
显示时间	否 是	是：在报警历史表中是否显示报警发生或解除的时间 否：不显示时间
字体	无	报警内容的字体。
走马灯速度	低 中 高	低：走马灯的流水速度低速度流 中：走马灯的流水速度中速度流 高：走马灯的流水速度高速度流
是否透明	否 是	否：背景色不透明 是：背景透明

#### 4. 报警部件支持矢量字体



效果：使用新宋体\_12



新建字体



13.4 当前报警表

- 1. 当前报警表在部件栏的报警显示中。
- 2. 当前报警表的功能：  
以列表的方式显示 HMI 报警缓冲区所有已发生的，但没有解除的报警信息。
- 3 当前报警表的属性：

属性名称	属性分类	说明
显示日期	否 是	是：报警历史表中是否显示报警发生和报警解除的日期 否：不显示日期
日期格式	当显示日期为是时有效	当显示日期为是时有效，日期的显示格式有下面几种选择 yy/mm/dd：年/月/日 mm/dd/yy：月/日/年 dd/mm/yy：日/月/年
显示时间	否 是	是：在报警历史表中是否显示报警发生或解除的时间 否：不显示时间
字体	无	报警内容的字体。
报警表不显示时解除的矩	是/否	是：发生报警了，报警表信息显示前有矩形框。



形框		否：发生报警了，报警表信息显示前没有矩形框。
----	--	------------------------

13.5 报警历史表

- 1. 报警历史表在部件栏的报警显示中。
- 2. 报警历史表的功能：

将报警缓冲区的所有记录以列表的方式列举出来，报警缓冲区记录了所有报警的发生时间和解除时间。

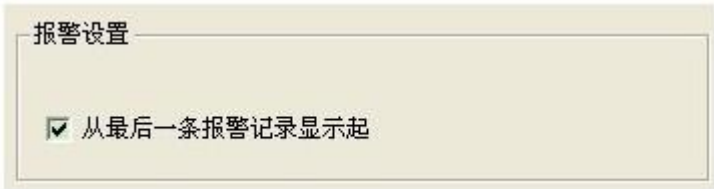
有关报警的详细信息，以及报警在 HMI 上的行为。

- 3. 报警历史表主要的属性：

属性名称	属性分类	说明
显示日期	否 是	是：报警历史表中是否显示报警发生和报警解除的日期 否：不显示日期
日期格式	当显示日期为是时有效	当显示日期为是时有效，日期的显示格式有下面几种选择 yy/mm/dd：年/月/日 mm/dd/yy：月/日/年 dd/mm/yy：日/月/年
显示时间	否 是	是：在报警历史表中是否显示报警发生或解除的时间 否：不显示时间
字体	无	报警内容的字体。

- 4. 可选功能：

新发生的报警可以在表顶（如下图）



## 14 图标曲线

### 14.1 历史圆盘记录图

- 1. 历史圆盘记录图在部件栏的图表曲线里。
- 2. 历史圆盘记录图的功能：

历史圆盘记录图是通过圆盘来读取缓冲区里的地址的值，曲线绕着圆心画图。历史圆盘记录图的地址是从数据记录区里记录的组。

- 3. 历史圆盘记录图的属性：

属性名称	属性分类	说明
记录组号		要显示的历史曲线图的记录组编号。
开始位置	上 右 下 左	上：从时钟零点位置开始画点； 右：从时钟三点位置开始画点； 下：从时钟六点位置开始画点； 左：从时钟九点位置开始画点。
起始时间		从对应缓冲区组记录中读取起始时间开始的数据到圆盘
轮转时间		设置圆盘画一周所需的时间，即一周画多长时间段的数据
曲线		可以选择显示哪几条曲线，曲线取用后可设置各曲线的标度范围和标度颜色。（标度范围：圆心到最外圈的数值范围） 最多能显示 4 条曲线。
通道号		对组里的通道进行选择显示

### 14.2 百分比趋势图

- 1. 百分比趋势图在部件栏中的图标曲线里。
- 2. 百分比趋势图的功能

百分比趋势图提供放大、缩小、上移、下移以及还原五个功能，

(1) 对应于设置的连续五个寄存器进行每个动作的步长操作，即设置的字地址的值对应每次操作的百分比数。

如组态中设置读取地址为“HDW200”，则在（包含自身）其后的连续 5 个地址，对应相应的操作：

HDW200->“放大”，HDW201->“缩小”，HDW202->“上移”，HDW203->“下移”，HDW204->“还原”，

例如设置 HDW200 设置为 5，表示操作一次每次放大的比例为 5%；

(2) 设置“还原”的字地址值为 1，则还原趋势图的显示，“还原”故名思意为还原为 0~100%的显示。

设置操作方式时，一般可以用字开关来设置相应的动作类型，当然也可用其他的部件来实现此功能。

3. 百分比趋势图的主要属性：

属性名称	属性分类	说明
显示刻度值	是 否	是否显示刻度
缓冲区编号	无	指定百分比趋势图所要读取的缓冲区
X 轴刻度	无	显示 X 轴需要多少等分
Y 轴刻度	无	显示 Y 轴需要多少等分
满屏点数	无	趋势图每屏显示的采样点数
读取地址	PLC 或 HMI 内部寄存器地址	即指定百分比趋势图提供放大，缩小，上移，下移以及还原的五个功能的起始地址。
是否显示滑杆	否 是	是:显示滑杆 否:不显示滑杆
滑杆颜色	颜色	滑杆颜色的设置
是否在曲线图 部件上显示数 据	否 是	否:不显示在曲线线图部件上 是:显示在曲线线图部件上
读取数据地址	PLC 或 HMI 内部寄存器地址	PLC 或 HMI 内部寄存器地址
读取时间地址	PLC 或 HMI 内部寄存器地址	PLC 或 HMI 内部寄存器地址
曲线	曲线 0-11	本部件 <b>最多可以显示 12 条曲线</b>
启用	否 是	是：显示曲线 否：不显示曲线

读取地址详细说明表

地址	动作类型
起始地址 X	放大
起始地址 X+1	缩小
起始地址 X+2	上移
起始地址 X+3	下移

起始地址	还原
X+4	

14.3 数据记录显示

- 1. 数据记录显示在部件栏中的图表显示中。
- 2. 数据记录显示的功能：

可以根据客户需要，查询某个时间段的所有记录数据，并以表格的方式显示出来，如果工程没有编辑数据记录区或者历史数据不存在，数据记录显示将没有任何内容。

在 HMI 上可以动态设定查询时间段，并且可以实现掉电保存，如下图所示：（参见 13.3 章节）



- 3.数据记录显示的属性：

属性名称	属性分类	说明
起始时间	无	表明部件显示从这个时间点以后的数据，这个时间格式是：年/月/日/时/分/秒。 这个时间在部件第一次运行时有效，之后，HMI 可以动态保存用户设置的查询时间段。
跨度	无	表示部件将显示多少数据。跨度最长 31 天，也就是部件垂直滚动条的拖动范围，格式是：日/时/分/秒。 这个时间在部件第一次运行时有效，之后，HMI 可以动态保存用户设置的跨度查询时间段。
查询时间	无	当前时间 上次设置的时间 当前时间往前 1 天 当前时间往前 1 小时

14.4 圆盘记录图

- 1. 圆盘记录图在部件栏的“图表曲线”里。
- 2. 圆盘记录图的功能：

圆盘记录图是通过圆盘来“实时读取”缓冲区里的地址的值，曲线绕着圆心画图。圆盘记录的地址必须要先在“工程设置”中的“实时圆盘记录图信息”里进行记录后，才能读取。
- 3. 圆盘记录图的属性：

属性名称	属性分类	说明
缓冲区编号		选择实时圆盘记录信息区里的缓冲区组。
开始位置	上 右 下 左	上：从时钟零点位置开始画点； 右：从时钟三点位置开始画点； 下：从时钟六点位置开始画点； 左：从时钟九点位置开始画点。
曲线		可以选择显示哪几条曲线，曲线取用后可设置各曲线的标度范围和标度颜色。 (标度范围：圆心到最外圈的数值范围)最多能显示 4 条曲线。

14.5 历史趋势图

- 1. 历史趋势图在部件栏的图表曲线里。
- 2. 历史趋势图的功能：

历史趋势图以时间为横轴，数据为纵轴，以图表的方式显示 HMI 历史缓冲区的数据。
- 3.历史趋势图的属性：

属性名称	属性分类	说明
数据范围是变量	是 否	趋势图显示的数据范围是否是变量 否：数据范围是常数 是：数据范围是变量，从指定的设备寄存器中读取
上限寄存器	当“数据范围是变量”为是时有效	当数据范围是变量为是时有效，趋势图显示范围的上限寄存器。
下限寄存器	当“数据范围是变量”为是时有效，	当数据范围是变量为是时有效，趋势图显示范围的下限寄存器。
X 轴刻度	无	X 轴——时间轴(横轴)的刻度数目
Y 轴刻度	无	Y 轴——数据轴(纵轴)的刻度数目
起始时间	无	设置数据查看的起始时间

跨度	无	要查看的数据范围
平移度	无	点击“左”或者“右”箭头的时候，要移动的范围。
记录组号	无	选择要查看的记录组号
曲线号变量	无	可以动态的显示不同的曲线号
通道号起始地址	当“数据范围是变量”为是时有效	修改此地址，可以显示不同的曲线
查询时间	无	查询的起始时间设置
曲线	曲线 0-7	一个趋势图部件最多可以显示 8 条曲线
启用	是 否	是：启用这条曲线 否：关闭这条曲线的显示
通道号	无	不同组别的默认通道
线颜色	无	描绘曲线时所用的颜色
线类型	无	描绘曲线时所用的线型

历史趋势图部件只负责以静态方式显示 HMI 数据缓冲区的数据，监控那些 PLC 地址的数据，则在设置---数据记录区里录入要监控的地址信息。

曲线号变量的设置：如果曲线号设置成变量，则可以根据寄存器的数据，动态的显示不同的曲线。  
例如：曲线 0：启动，曲线 1：启动，曲线 2：启动，曲线号是变量，通道号起始地址：HDW0，则 HDW0、HDW1、HDW2 的 3 个寄存器分别可以动态的修改 3 条曲线。

读取地址数	4 个字
读取地址	曲线 0 的变量寄存器
读取地址+1	曲线 1 的变量寄存器
读取地址+2	曲线 2 的变量寄存器
读取地址+3	曲线 3 的变量寄存器

14.6 历史 XY 图

- 1. 历史 XY 图在部件栏的图表曲线里。
- 2. 历史 XY 图的功能：

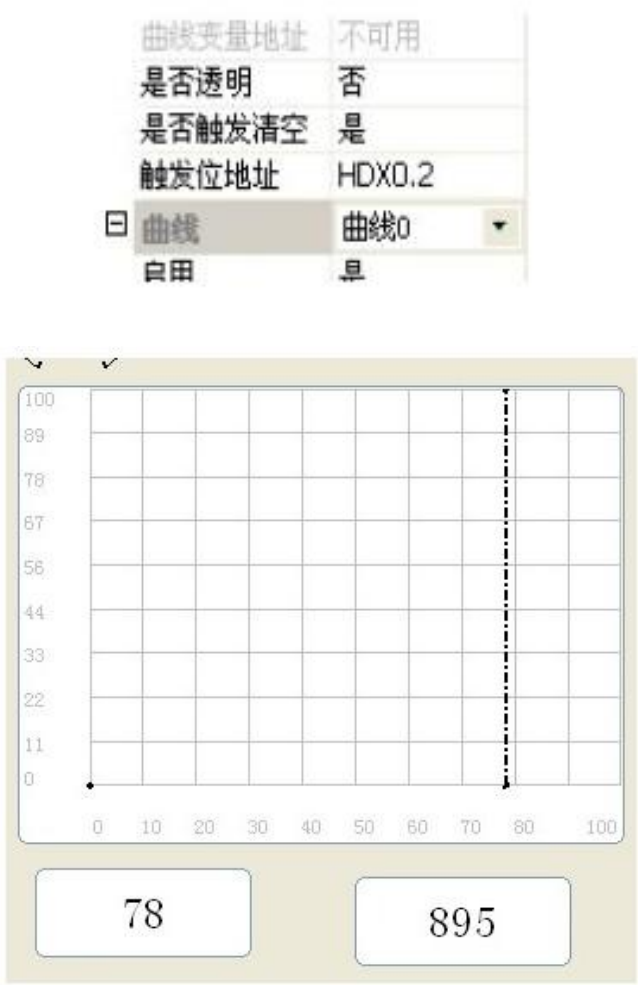
历史 XY 图是以连续曲线或者散列点的方式显示 HMI 中 XY 图缓冲区的内容。历史 XY 图缓冲区与趋势图缓冲区所不同的地方是，对于每一个地址，XY 图连续读两个字上来，第一个字作为 X 轴(横轴)，第二个字作为 Y 轴(纵轴)，来确定一个点的坐标。

历史 XY 图部件只负责以动态曲线的方式显示 HMI XY 图缓冲区的数据，置于监控那些 PLC 地址的数据，则在设置---历史 XY 图里录入要监控的地址信息。

- 3. 历史 XY 图的属性：

属性名称	属性分类	说明
XY 图样式	散列点 连结点	散列点：以散列点的方式描绘 XY 图的数据 连结点：用线段将先后描绘的两点连接起来
数据范围是变量	是 否	否：数据范围是常数 是：数据范围不是常数，是变量
X 值范围	当“数据范围是变量”为是时有效	XY 图能显示的 X 轴的最大值和最小值，超出这个范围的点数将无法正常显示。
X 上限寄存器	当“数据范围是变量”为是时有效，	当“数据范围是变量”为是时有效，上限寄存器是指棒图测量范围是 PLC 或者 HMI 内部寄存器的一个内部变量
X 下限寄存器	当“数据范围是变量”为是时有效，	当“数据范围是变量”为是时有效，下限寄存器是指棒图测量范围是 PLC 或者 HMI 内部寄存器的一个内部变量
Y 值范围	当“数据范围是变量”为是时有效，	XY 图能显示的 Y 轴的最大值和最小值，超出这个范围的点数将无法正常显示。
Y 上限寄存器	当“数据范围是变量”为是时有效，	当“数据范围是变量”为是时有效，上限寄存器是指棒图测量范围是 PLC 或者 HMI 内部寄存器的一个内部变量
Y 下限寄存器	当“数据范围是变量”为是时有效，	当“数据范围是变量”为是时有效，下限寄存器是指棒图测量范围是 PLC 或者 HMI 内部寄存器的一个内部变量
点类型	1 点 2 点 3 点 5 点 没有点	描绘 XY 图坐标点的点类型：1 点：一个坐标点由一个像素点组成，2 点：一个坐标点由二个像素点组成，3 点：一个坐标点由三个像素点组成，5 点：一个坐标点由五个像素点组成
X 轴刻度	无	垂直格线的条数
Y 轴刻度	无	水平格线的条数
曲线变量	否 是	否：曲线不为变量 是：曲线是为变量
曲线变量地址	PLC 或 HMI 内部寄存器地址	所监控的字地址的
曲线	曲线 0-7	一个 XY 图部件最多可以显示设 8 条曲线
启用	否 是	是：显示曲线 否：不显示曲线
是否触发清空	是/否	是：一个位地址触发，对数据进行清空操作。否：不操作
XY 曲线编号	无	曲线在趋势图缓冲区的编号

4. 历史 XY 图支持触发清空曲线



14.7 XY 趋势图

- 1. XY 曲线图在部件栏的图表曲线里。
- 2. XY 曲线图的功能：

读取 PLC 寄存器一段连续的数值，再按照指定数据格式转换所读的数值，然后在屏幕上以 XY 图的方式示出来。

假设读取 PLC 寄存器的字地址为 D，而点数为 m  
则

- 字地址 D 为第一个点的 X 坐标
- 字地址 D+1 为第一个点的 Y 坐标
- 字地址 D+2 为第二个点的 X 坐标
- 字地址 D+3 为第二个点的 Y 坐标
- .....
- 字地址 D+2m-1 为第 m 个点的 X 坐标
- 字地址 D+2m 为第 m 个点的 Y 坐标



因为如果 XY 点数是 m，则要从 PLC 的地址 D 开始，连续读取 2m 个字上来，才能绘制完整的 XY 图。

### 3. XY 曲线图的属性：

属性名称	属性分类	说明
XY 图样式	散列点 连结点	散列点：以散列点的方式描绘 XY 图的数据 连结点：用线段将先后描绘的两点连接起来
X 值范围	无	XY 图能显示的 X 轴的最大值和最小值，坐标超出该范围的点将无法显示。
Y 值范围	无	XY 图能显示的 Y 轴的最大值和最小值，坐标超出该范围的点将无法显示。
点类型	没有点 1 点 2 点 3 点 5 点	描绘 XY 图坐标点的点类型 1 点：一个坐标点由一个像素点组成 2 点：一个坐标点由二个像素点组成 3 点：一个坐标点由三个像素点组成 5 点：一个坐标点由五个像素点组成
X 轴刻度	无	垂直格线的条数
Y 轴刻度	无	水平格线的条数
是否启用 自动采集	否 是	否：不采集 是：是采集
采集时间	数值	采集数据的时间段要多少
X 轴坐标是否 固定	否 是	否：不固定 X 坐标 是：固定 X 坐标
X 轴坐标放大 倍数	当“X 轴坐标是否 固定”为是时有效，	对 X 轴的数值进行放大
是否触发采样	否 是	否：不触发采样 是：触发采样
触发位地址	当“是否触发采样” 为是时有效，	当“是否触发采样”为是时有效，PLC 或 HMI 内部寄存器地址。
曲线变量	否 是	否：曲线不为变量 是：曲线是为变量
曲线变量 地址	PLC 或 HMI 内 部寄存器地址	部件显示的曲线编号为该地址里的值
是否透明	否 是	否：不透明 是：透明
曲线	曲线 0-7	一个 XY 图部件最多可以显示 8 条曲线

启用		是：启用这条曲线 否：关闭这条曲线的显示
读取地址	PLC 或 HMI 内部寄存器地址	

## 14.8 趋势图

1. 趋势图在部件栏的图表曲线里。

2. 趋势图的功能：

趋势图以时间为横轴，数据为纵轴，以图表的方式显示 HMI 趋势图缓冲区的数据。

趋势图部件只负责以动态曲线的方式显示 HMI 趋势图缓冲区的数据，监控那些 PLC 地址的数据，则在“设置”——“实时趋势图”里录入要监控的地址信息。

有关趋势图的详细实现，以及趋势图在终端的行为，请参考 13.1 章节。

3. 趋势图的属性：

属性名称	属性分类	说明
显示刻度值	是 否	是:显示刻度 否:不显示刻度
X 轴刻度	当“显示刻度值”为是时有效	X 轴——时间轴(横轴)的刻度数目
Y 轴刻度	当“显示刻度值”为是时有效	Y 轴——数据轴(纵轴)的刻度数目
满屏点数	无	趋势图每屏显示的采样点数
曲线号是变量	否 是	否:曲线号不是变量 是:如果曲线号是变量，那么将由曲线号寄存器来决定显示趋势图缓冲区的那一组曲线
缓冲区编号	当“曲线号是变量”为否时有效	仅当“曲线号是变量”为否时有效，趋势图缓冲区的编号
曲线号寄存器	当“曲线号是变量”为是时有效	仅当“曲线号是变量”为是时有效，趋势图部件显示的曲线将由该寄存器的值来决定，数据格式与趋势图是一致的，如果曲线号在趋势图缓冲区不存在，将无法显示。
显示滚动条	否 是	是：当数据满屏时出现滚动条 否：数据满屏也不会出现滚动条，直接将缓冲区最前面的数据显示出来。

绘制报警区颜色	否 是	否：不在趋势图上绘制报警区域 是：在趋势图上绘制上限报警和下限报警区域
上限值地址	PLC 或 HMI 内部寄存器地址	报警区域的上限值地址从哪里读出来，以此绘制上限报警线。
上限线型	线形	绘制上限报警线所采用的线型
上限线颜色	色彩	绘制上限线时所采用的颜色
下限值地址	PLC 或 HMI 内部寄存器地址	报警区域的下限值地址从哪里读出来，以此绘制下限报警线。
下限线型	线形	绘制下限报警线所采用的线型
下限线颜色	色彩	绘制下限线所采用的颜色
目标值地址	PLC 或 HMI 内部寄存器地址	目标曲线的值从哪里读出来，以此绘制目标线。
目标线型	线形	绘制目标线所采用的线型
目标线颜色	色彩	绘制目标线所采用的颜色
曲线	曲线 0 曲线 1 曲线 2 曲线 3 曲线 4 曲线 5 曲线 6 曲线 7	一个趋势图部件最多可以显示 8 条曲线
启用	否 是	是：显示曲线 否：不显示曲线
是否不显示 Y 轴	是/否	否：显示 Y 轴刻度值 是：不显示 Y 轴刻度值

## 15 动画位图

### 15.1 旋转位图

1. 旋转位图显示在部件栏的动画位图中。（旋转位图支持 bmp 图片, 不支持 jpg 图片）

**功能：** 图片以一个指定的圆心旋转，旋转的方式由属性设置决定。

旋转位图的属性：

属性名称	说明
数据格式	有符号十进制，无符号十进制，BCD，32 位浮点数
旋转角度地址	PLC 或 HMI 内部寄存器地址
数据范围是变量	是 否
旋转中心点位置	数据范围是变量为“否”时有效， 显示坐标用鼠标拖动来改变
旋转方式	是 否
旋转周期	旋转方式为“是”时有效，单位为：100 毫秒； 例如：输入 10，就是 10*100 毫秒=1 秒
旋转角度	填写旋转角度的大小，为正整数
是否终点返回	是/否
是否透明	是/否
逆时针还是顺时针	是 否

### 15.2 轨迹动画

1. 轨迹动画在部件栏的动画位图中。

2. 轨迹动画的功能：

轨迹动画是 HMI 位图按照屏幕上的预先设置的轨迹来运动，由 PLC 来决定 HMI 位图在哪个轨迹点上显示，以及显示哪个状态的位图。

3. 添加轨迹动画的过程：

1. 在画面上选取适当的位置和大小放置此部件；

2. 在画面上点击鼠标左键，并且移动鼠标。在画面上会留下一系列的点 0(1, 2, 3...)，这些点是预先设好的轨迹点。可以选择指定的轨迹点，点 0 为第一个轨迹点，点 1 为第二个轨迹点，依次类推；支持最多 50 个轨迹点；

- 3. 单击鼠标右键，完成对轨迹点的输入；
- 4. 轨迹点输入完成后，如果想改变某个某个轨迹，可以通过下面的办法：
  - 1、直接从属性框的定点集中修改坐标。
  - 2、在轨迹动画处于选中的状态下，再一次鼠标左键要拖动的点，此时图形进入内部选中状态，即可以拖动轨迹点到预想的位置。

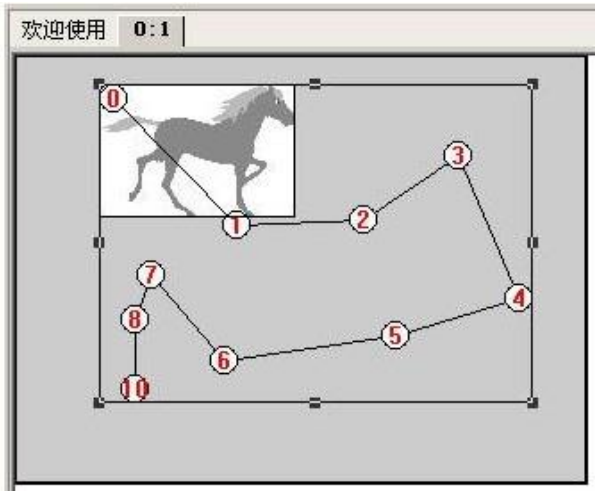


图 15-1 轨迹动画示例

根据字地址所设置的值，轨迹动画需要读连续的两个字上来，这两个字的功能和作用如表 15-1。

表 15-1 读取地址说明

读取地址数	2 个字
读取地址（第一个字）	设置向量图或位图状态
读取地址+1（第二个字）	设置预设轨迹上的位置编号

5. 轨迹动画的属性：

属性名称	属性分类	说明
读取地址	PLC 或 HMI 内部 寄存器地址	移动图形监视的首地址
顶点集	无	轨迹图形预先设置的轨迹点个数
按时间移动	否 是	否:不按指定时间移动 是:按指定时间移动
移动频率 (100ms)	当“按移动时 间”设为是时有效	移动速度快慢
自动变换	当“按移动时间” 设为是时有效 否 是	否:不变换 是:变换

变换频率 (100ms)	当“按移动时间”设为是时有效	变换快慢设定
终点返回	当“按移动时间”设为是时有效 否 是	否:不返回 是:返回
是否触发	当“按移动时间”设为是时有效 否 是	否:不触发 是:触发
状态	状态 0-31	轨迹动画最多支持 32 个状态

### 15.3 流动部件

1. 该部件在动画位图栏里
2. 可以实现流水动画式的效果

线类型	对线条的外光选取
边框颜色	修改边框的色体
填充图案	对填充样式的修改
外观位图	修改外观图形
流动块前景色	前景色的设置
流动块背景色	背景色的设置
管道外观位图	自定义外观图
管道前景色	管道前景色的设置
管道背景色	管道前背色的设置
流动块个数	流动块地个数
触发位地址	触发流动的位地址
边框透明	是/否
显示方向	流动部件流动运行的方向
流动速度	速度控制

### 15.4 移动图形

- 1. 移动图形在部件栏中的动画位图里。
- 2. 移动图形的功能：

移动图形可用来把一个 HMI 位图(向量图)放在屏幕上指定的位置,图形的状态和绝对位置由 PLC 或 HMI 寄存器上 3 个连续的字地址寄存器提供。

一般地说,第一个寄存器控制部件的状态,第 2 个控制水平坐标(X),第 3 个控制垂直坐标(Y)。

- 3. 移动图形的分类：

移动图形部件最多有 32 种不同的状态。

移动图形的位置是相对于原来的初始位置而言的。(X 轴表示横坐标,Y 轴表示纵坐标。)

表 15-2 移动方式说明

移动方式	X 轴移动	Y 轴移动	XY 轴移动
读取地址数	2 个字	2 个字	3 个字
读取地址	位图的状态	位图的状态	位图的状态
( 读 取 地 址 +1)* 比 例 增 益	X 轴位移	Y 轴位移	X 轴位移
(读取地址+2)*比例增益	——	——	Y 轴位移

表 15-3 动作类型说明表

动作类型	说明
沿 X 轴水平移动	移动图形部件只做 X 轴水平移动,第一个寄存器存放部件的状态,第二个字的值乘以比例增益就是 X 轴位移。
沿 Y 轴垂直移动	移动图形部件只做 Y 轴垂直移动,第一个寄存器存放部件状态,第二个字的值乘以比例增益就是 Y 轴位移。
沿 X 和 Y 轴移动	移动部件做 X 和 Y 轴移动,第一个寄存器存放部件状态,第二个字的值乘以比例增益就是 X 轴位移,第三个字的值乘以比例增益就是 Y 轴位移。。

如果从设备元件读取得值超出了“数据范围”,假如数据范围是[0,100](X 的范围是[0,100],Y 的范围是[0,100])：当读的值小于 0 是,则图形将显示初始位置；

如果大于 100,则图形放置在 “100\*比例增益” 的位置上。

这个数据对 X、Y 轴位移都是等效的。

- 5. 移动图形的属性：

属性名称	属性分类	说明
读取地址	PLC 或 HMI 内	移动图形所监控两(或三)个连续字地址的首地址

	部寄存器地址	
外观位图	无	部件的 HMI 位图，从位图库选择
动作类型	X 轴移动、 Y 轴移动、 XY 轴移动	移动图形的三种动作类型： X 轴移动、Y 轴移动、XY 轴移动
数据范围	无	移动图形所能显示的数据范围
比例增益	无	移动图形的计算比例
状态	状态 0-31	移动图形最多有 32 个状态

15.5 移动图形缩放功能

1. 移动、缩放功能简介

为了让我们的产品功能更加强大，满足更多的客户需求，特开发了部件的移动、缩放功能。

由于短期内对所有部件均开放这一功能是不现实的，所以决定先让“移动图形”部件和“数值输入/显示”部件实现移动、缩放功能，后续将会逐渐增加其它部件实现这一功能。

因为“移动图形”本身具备移动功能，所以现在只是对其增加了缩放功能，且对旧“移动图形”部件兼容，大家可以放心使用。移动、缩放是两个在使用上是两个独立的功能，可以同时使用，也可以单独使用。

2. “移动图形”部件的移动：

如图 15-2，是移动图形和移动图形的基本设置界面。



图 15-2 移动图形移动设置



“基本”设置就是移动图形原来的移动功能设置。地址、数据格式和动作类型之间比较密切的联系。设置的读取地址控制的是移动位图中位图的状态，读取地址的下一个地址控制什么取决于动作类型。

如果动作类型是“X 轴移动”，那么这个地址就是控制 X 轴移动；如果是“Y 轴移动”，那么这个地址就是控制 Y 轴移动，如果是“XY 轴移动”，那么这个地址是控制 X 轴移动，再下一个地址控制 Y 轴移动，而这两个地址的间隔和数据格式有关，如果数据类型是单字，则是联系的地址，如果是双字，则间隔 1。

举个例子，如图 15-2 中，读取地址为 HDW0，这个地址也是控制这个部件的位图状态号。

移动类型为 XY 轴移动，那么 HDW0 的下一个地址 HDW1 控制 X 轴移动，由于数据类型是无符号十进制，属于单字，所以再接下去的一个地址 HDW2 控制 Y 轴移动。如果选的不是 XY 轴移动，而是 Y 轴移动，那么 HDW1 控制 Y 轴移动，HDW2 则未定义。

3. “移动图形”部件的缩放：如图 15-3，是移动图形的缩放功能设置界面。

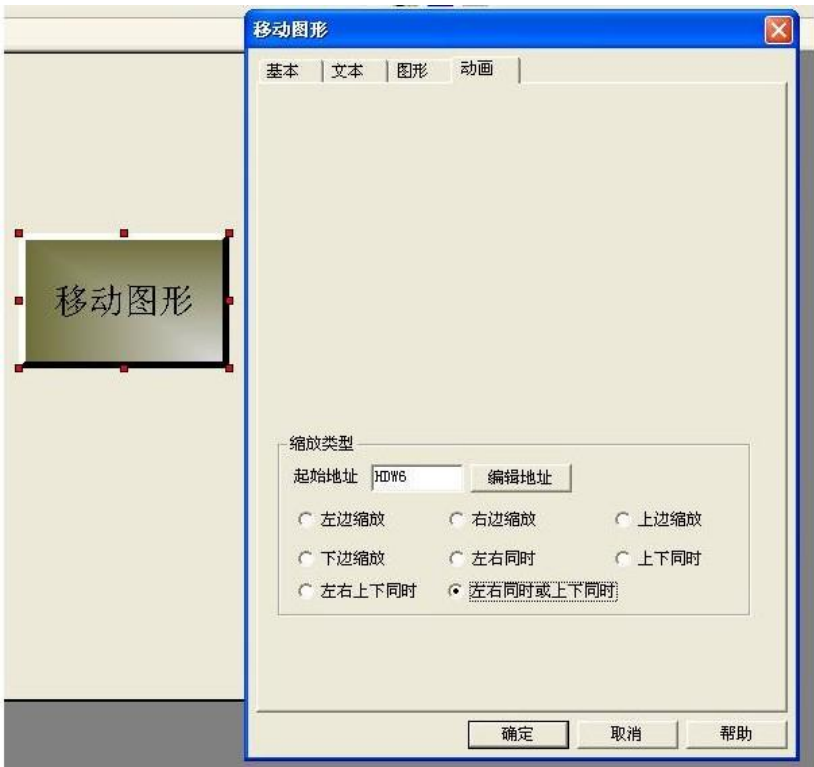


图 15-3 移动图形缩放设置

首先设置起始地址（注意不要和其它地址重复了），然后选择缩放类型。如果选择除“左右同时或上下同时”以外的其它缩放类型，那么，起始地址就是控制该类型的地址。如果选择“左右同时或上下同时”，那么起始地址控制左右同时缩放，下一个地址和移动设置中的数据类型相关，如果是单字，则起始地址的下一个地址控制上下同时，如果是双字类型，则间隔一个地址控制上下同时。

如图 15-3，起始地址是 HDW6，那么 HDW6 控制左右同时，又由于数据类型是单字的，所以 HDW7 控制上下同时。

4. 比例增益：

如图 15-2，比例增益设置是对移动、缩放值的比例变化。比如比例增益值为 x，移动或缩放的值为 y，那么实际移动或缩放的值为（x\*y）。

5. 拓展运用

利用移动、缩放功能，我们可以实现以下运用：

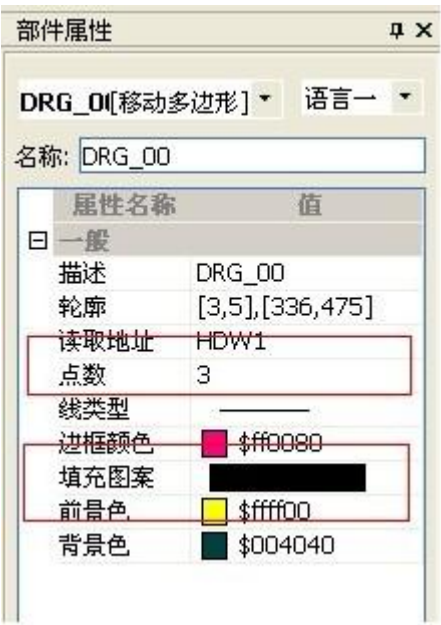
- 1. 移动图形、数值输入/显示部件的可控制移动
- 2. 移动图形、数值输入/显示部件的可控制缩放
- 3. 移动位图部件边移动边缩放并可同时改变位图状态。
- 4. 可以由脚本控制产生一个按指定坐标变化轨迹的动画
- 5. 另外，数值输入/显示部件新增标签功能，可结合使用，实现移动的带标签的字符。

以上的只是几个例子，您完全可以运用它们实现您自己想要的功能。当然，如果您有更好的创意，可以将这功能变得更通用，请您联系我们。

15.6 移动多边形

根据寄存器的值变换图形的位置和状态：

属性编辑如下图：





根据起始地址后的 N 个地址来改变画出想要绘制的多边形图形。

15.7 视频输入显示部件

一. 简介:

为了满足客户的需求，推出新功能：视频输入显示。实现对视频的实时监控，满足客户对当前情况的最新最快的了解掌握。

示例使用说明

假设操作地址 OperateAddr 为 HDW100，那么（之后连续的 13 个地址）工程设置如下：

特殊寄存器地址	值	含义
HDW100	0	暂停
	1	开始
	2	停止
HDW101, HDW102, HDW103, HDW104		设置视频输入显示的原始画面坐标参数
HDW105		视频输入当前播放的视频路数设置
HDW106		视频输入亮度设置
HDW107		视频输入对比度设置
HDW108		视频输入饱和度设置
HDW109		视频输入色调控制设置
HDW110		复位并初始化视频输入芯片

HDW111		设置视频的旋转角度 (0、90、180、270)（位地址）
HDW112		复位视频输入显示的颜色设置（位地址）

工程设置如下：（读取地址是 HDW100，那么地址就是之后的 13 个地址）



二：关于属性的设置

	明 度	对比 度	饱和 度	色调的设置
明 度	范围 0~255，0 最暗，255 最亮			
对 比 度	范围-128~0~127，-128 亮度反转，0 暗，127 强			
饱 和 度	范围-128~0~127，-128 反色，0 暗，127 强			
色 调	范围 0~255，0 正常，255 冷色			

关于路数的设置

最大支持 4 路视频输入（同一时间只能显示其中一路），设置范围 0~3，0 显示第 0 路，1 显示第 1 路，以此类推。

关于复位的设置

当寄存器位输入时，会复位视频输入的设置。所有的设置会恢复为系统默认值。

### 关于旋转角度的设置

1. 旋转角度范围 0~3，分别对应旋转 0 度，旋转 90 度，旋转 180 度，旋转 270 度。
2. 旋转 90 度和 270 度会影响图片的处理速度，尽量少用。
3. 在竖屏的 HMI 中，必须设置旋转角度，否则按照默认横屏处理。

三：注意事项：

- 此视频输入信号显示在最高层，任何弹出键盘，画面，等窗口。都会被视频显示区域所覆盖。
- 视频信号必须设置好其显示区域，明度，对比度，饱和度，色调等设置，否则会照成显示异常，如果不会设置，请按照默认设置，如果设置错误，请使用复位设置功能。
- 在 PAL 制式下，视频输入视频源输入最大输入 312 行。
- 在 NTSC 制式下，最大只有 256 行，请自行设置好视频的最大行数，如果在 NTSC 制式下，使用 312 行画图，有可能照成最 56（312-256）行数据显示错误。

## 16 仪表

### 16.1 扇形图

- 1、扇形图在部件栏的仪表里。
- 2、扇形图的功能：

扇形图是通过地址画出相关扇形样式。

- 3、扇形图的属性：



属性名称	属性分类	说明
坐标	X 轴/Y 轴	根据坐标轴设置扇形图的位置
大小	宽/高	根据需求设置扇形图的大小
变量编辑		成员个数：用来设置组成扇形图的成分个数（小于 32 的正整数） 数据格式:BCD 码/有符号十进制/无符号十进制/32 位浮点/32 位有符号十进制/32 位无符号十进制
地址编辑		起始地址：填上控制扇形图的首地址 地址范围：根据首地址和成员个数，自动锁定关联地址。
统计数值		数值类型：不显示/数值型/百分比型 字体颜色：对字体设置相关的颜色

16.2 棒图

- 1. 棒图在部件栏的仪表里。
- 2. 棒图的功能：

棒图是通过柱状图的方式来反应字地址的值的变化情况。

- 4. 棒图的主要属性如下表所示

属性名	属性分类	说明
-----	------	----

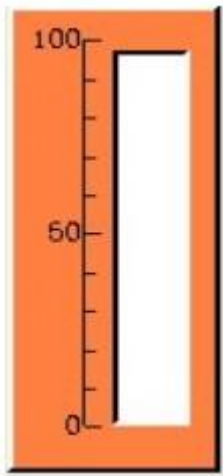
称		
读取地址	PLC 或 HMI 内部 寄存器地址	所监控的字地址的
数据格式	BCD 32 位浮点 无符号十进制 有符号十进制	BCD: 以 BCD 码的方式读出数据; <b>32 位浮点</b> : 以标准浮点格式读出数据 <b>无符号十进制</b> : 以无符号 BIN 码的方式读出数据; <b>有符号十进制</b> : 以有符号 BIN 码的方式读出数据;
数据 范围是变 量	是  否	否: 数据范围是变量 是: 数据范围不是常数, 是变量
数据范围	当“数据范围是 变量”为否时有效,	当“数据范围是变量”为否时有效, 可以在仪表类部件上显示的 数据范围。 这个值相当于物理仪表的量程, 值低于量程, 仪表将最低位显示; 值高于量程, 仪表将满量程显示。
向量 图	色彩	仪表类部件的外观
边框颜色	色彩	向量图的边框颜色选择, 如果无向量图或者向量图不支持, 该属性 将不可用
是否双色	是与否 否: 不启用 是: 双色显示监控;	是: 双色控制棒图双地址。 一个读地址, 一个控制棒图双色区的下限地址。
表盘颜色	色彩	棒图表盘的背景色
绘制报警	否  是	是: 报警时绘制 否: 报警时不绘制
上线颜色	当“绘制报警” 为是时有效,	当“绘制报警”为是时, 而当读取的值比设定值大时就用该栏设定 的颜色显示
下线颜色	当“绘制报警” 为是时有效,	当“绘制报警”为是时, 而当读取的值比设定值小时就用该栏设定 的颜色显示
数据 范围是变 量	是  否	否: 数据范围是变量 是: 数据范围不是常数, 是变量
数据范围	当“数据范围是 变量”为否时有效,	当“数据范围是变量”为否时有效, 可以在仪表类部件上显示的 数据范围。 这个值相当于物理仪表的量程, 值低于量程, 仪表将最低位显示; 值高于量程, 仪表将满量程显示。

Y 上限值寄存器	当“数据范围是变量”为是时有效，	Y 上限的量程
Y 下限值寄存器	当“数据范围是变量”为是时有效，	Y 下限的量程
上限值地址	有报警时的数据范围是变量的时候	报警时的数据范围是变量的上限值
下限值地址	有报警时的数据范围是变量的时候	报警时的数据范围是变量的下限值
图表	无	以各自图表的方式显示当前的状态
填充图案	色彩	棒图活动区域的填充图案 当向量图为空或者所选择的向量图不支持前景色，此项将不可用
前景色	色彩	棒图活动区域填充所使用的前景色， 当向量图为空或者所选择的向量图不支持前景色，此项将不可用
背景色	色彩	棒图活动区域填充所使用的背景色， 当向量图为空或者所选择的向量图不支持前景色，此项将不可用
显示刻度值	是 否	否：不显示刻度标记 是：显示刻度文本标度
标度颜色	色彩	刻度标记的文本颜色，仅当显示刻度值为是有效。
标度范围	无	标度的范围，仅当显示刻度值为是有效。
小数点位数	显示刻度值为是有效， 数值	标度的小数点位置，仅当显示刻度值为是有效, 用来设定刻度线上的数值小数位的位数
显示刻度线	是 否	否：不显示刻度线 是：在棒图中显示刻度线
主刻度数	当显示刻度线为是时有效， 数值	主刻度的数目，仅当显示刻度线为是时有效, 用来设定每个大格子的跨度
次刻度数	当显示刻度线为是时有效， 数值	相邻主刻度之间，次级刻度的数目，其刻度线长度为主刻度的一半， 用来设定每个小格子的跨度，仅当显示刻度线为是时有效。
刻度	色彩	刻度线的颜色，仅当显示刻度线为是时有效



颜色		
----	--	--

下图是一个典型的棒图，各部分属性如图中标注：



活动区的大小反映了字地址所指的值的变化

16.3 饼图

- 1. 饼图在部件栏的仪表里。
- 2. 饼图的功能：

饼图是通过饼状图形的方式来反应字地址值的变化情况。

- 5. 饼图的属性：

饼图的主要属性如下表所示：

属性名称	属性分类	说明
读取地址	PLC 或 HMI 内部寄存器地址	所监视的字地址
数据格式	BCD 32 位浮点 有符号十进制 无符号十进制	BCD：以 BCD 码的方式读出数据； 32 位浮点：以标准浮点格式读出数据 有符号十进制：以有符号 BIN 码的方式读出数据； 无符号十进制：以无符号 BIN 码的方式读出数据；
数据范围	数值	可以在仪表类部件上显示的数据范围，这个值相当于物理仪表的量程，值低于量程，仪表将无法显示；值高于量程，仪表将满量程显示。
向量图	无	仪表类部件的外观

表盘颜色	色彩	饼图表盘的背景色
主刻度数		总的刻度的个数
次刻度数		每段里的刻度个数

下图是一个典型的饼图

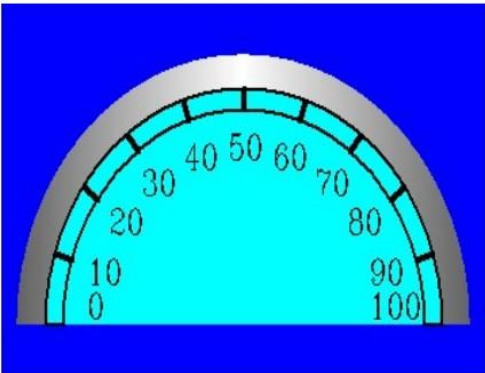


图 16-1 饼图的属性含义

扇形活动区的大小显示了字地址所指的值的变化。

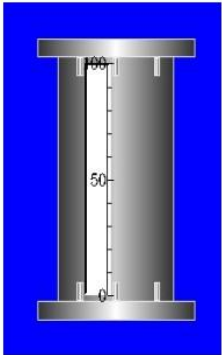
16.4 罐图

- 1. 罐图在部件栏的仪表里。
- 2. 罐图的功能：  
罐图是通过罐状图形的方式来反应字地址的值的变化情况
- 3. 罐图的属性：  
罐图的属性如下表所示：

属性名称	属性分类	说明
读取地址	PLC 或 HMI 内部寄存器地址	所监视的字地址
数据格式	BCD 32 位浮点 有符号十进制 无符号十进制	BCD：以 BCD 码的方式读出数据； 32 位浮点：以标准浮点格式读出数据 有符号十进制：以有符号 BIN 码的方式读出数据； 无符号十进制：以无符号 BIN 码的方式读出数据；
数据范	数值	可以在仪表类部件上显示的数据范围，这个值相当于物理仪表的量程，值

围		低于量程，仪表将无法显示；值高于量程，仪表将满量程显示
向量图	无	仪表类部件的外观
边框颜色	色彩	罐图边框颜色
标盘背景色	色彩	标盘的背景色
主刻度数		总的刻度的个数
次刻度数		每段里的刻度个数

下图是一个典型的罐图。各部分属性如图中标注：



16.5 仪表

- 1、仪表在部件栏的仪表里。
- 2、仪表的功能：

仪表是通过表盘和指针的方式来反应所指字地址的值的变化情况，仪表在外观上与实际的仪表很相像。

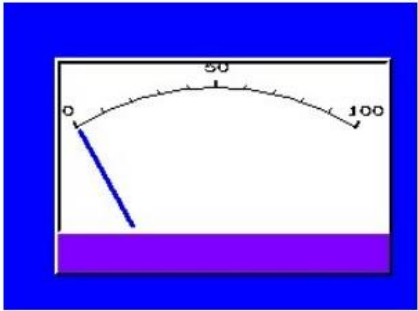
- 3、仪表的属性：

属性名称	属性分类	说明
字地址	PLC 或 HMI 内部寄存器地址	所监控的字地址的
数据格式	BCD 32 位浮点 有符号十进制 无符号十进制	BCD：以 BCD 码的方式读出数据； 32 位浮点：以标准浮点格式读出数据； 有符号十进制：以有符号 BIN 码的方式读出数据； 无符号十进制：以无符号 BIN 码的方式读出数据；
数据范围	数据范围是变量为：“否”时有效	可以在仪表类部件上显示的数据范围，这个值相当于物理仪表的量程，值低于量程，仪表将无法显示；值高于量程，仪表将满量程显示

Y 上限寄存器	数据范围是变量 为：“是”时有效	Y 上限寄存器地址
Y 下限寄存器	数据范围是变量 为：“是”时有效	Y 下限寄存器地址
仪表反向显示	否 是	是:就是反向显示 否:就是不反向显示
仪表方向		指针的转动方向：顺时针和逆时针方向
显示指针	是 否	活动区域填充所使用的前景色， 当向量图为空或者所选择的向量图不支持前景色，此 项将不可用。
指针颜色	当“显示指针”为 否时有效，	仪表指针的颜色
显示刻度值	是 否	是：显示刻度标度 否：不显示刻度标记
主刻度数	当“显示刻度值” 为是时有效 数值	主刻度的数目，仅当“显示刻度值”为是时有 效,用来设定每个大格子的跨度。
次刻度数	当“显示刻度值” 为是时有效 数值	相邻主刻度之间，次级刻度的数目，其刻度线长 度为主刻度的一半，用来设定每个小格子的跨度。仅 当“显示刻度值”为是时有效。
刻度颜色	当“显示刻度值” 为是有效	刻度标记的文本颜色，仅当“显示刻度值”为 是有效。
显示刻度值	是 否	否：不显示刻度标记 是：显示刻度文本标度
标度范围是 否是变量	是 否	是:启用标度范围的范围 否:不启用标度范围的范围
标度范围	当“标度范围”为 否时有效	标度的范围，当“标度范围”为否时有效
标度范围上 线地址	当“标度范围”为 否时有效， PLC 或 HMI 内部寄存器 地址	当“标度范围”为否时有效，PLC 或 HMI 内部寄 存器地址
标度范围下 线地址	当“标度范围”为 否时有效， PLC 或 HMI 内部寄存器	当“标度范围”为否时有效，PLC 或 HMI 内部寄 存器地址

	地址	
小数点位数	当“显示刻度值”为是有效	标度的小数点位置，仅当显示刻度值为是有效。
显示刻度线	是 否	否：不显示刻度线 是：在棒图中显示刻度线
异常显示	否 是	是：在仪表盘上标出异常区域 否：不在仪表盘上标出异常区域
正常值范围	当“异常显示”时有效	仪表正常现实的数据范围，仅当异常显示为是时有效
低区颜色	当“异常显示”时有效	低于正常范围的表盘颜色，仅当异常显示为是时有效 (低区：在标度范围内，低于正常值外的数值范围)
高区颜色	当“异常显示”时有效	高于正常范围的表盘颜色，仅当异常显示为是时有效 (高区：在标度范围内，高区正常值外的数值范围)

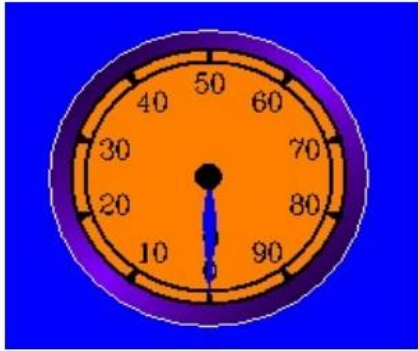
下图是一个典型的仪表图。各部分属性如图中标



4、仪表部件支持 360 角度的仪表外观

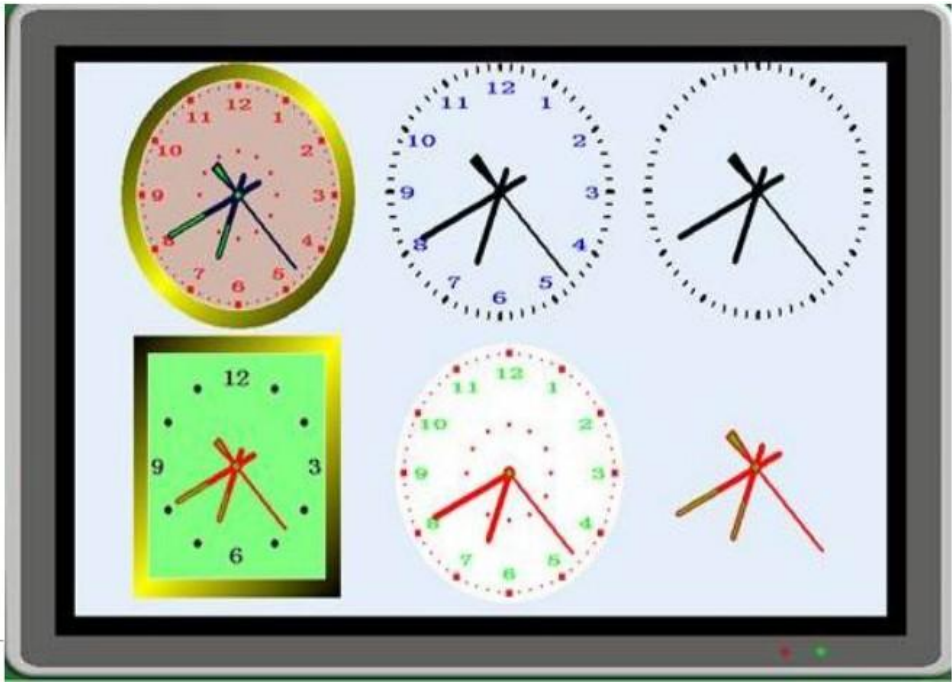
如下图所示

曲线变量地址	不可用
是否透明	否
是否触发清空	是
触发位地址	HDX0.2
曲线	曲线0
白田	是



16.6 时钟

- 1、时钟在部件栏的仪表里。
- 2、时钟的功能：  
时钟是通过不同的外观样式显示当前时间。
- 3、时钟的属性：



属性名称	属性分类	说明
边框颜色		对边框颜色自定义设置
表盘背景色		对表盘颜色自定义设置
样式		根据各自喜好，选择时钟的外观样式。
坐标	X 轴/Y 轴	根据坐标轴设置时钟的位置
大小	宽/高	根据需求设置时钟的大小
刻度值和刻度线类型		共 5 个类型，根据需求选择设置。
显示刻度值	是/否	

标度原色		当显示刻度值为是时，对标度色度进行设置
显示指针	是 否	是：进行设置相关的原色和指针类型。（共 5 个类型） 否：不进行任何设置
显示刻度线	是/否	是：设置刻度颜色 否：不进行设置

## 17 指示灯

### 17.1 位状态指示灯

1. 位状态指示灯在部件栏的指示灯中。
2. 位状态指示灯的功能：

位状态指示灯监视指定的位的状态，此位为 0，指示灯将显示关状态的位图、向量图和状态文本；为 1，则显示开状态的位图、向量图和状态文本。

3. 位状态指示灯的属性：

属性名称	属性分类	说明
监控地址	内部地址或外部 PLC 地址	位指示灯所监视的位地址
显示方向	向量图和标签文本的显示方向	显示方向（安旋转角度显示）0 度，90 度，180 度，270 度
是否反向显示	否 是	(机子内部默认地址为 0) 是: 表示该地址的位为 1 否: 正常显示部件的状态，0 显示关状态信息，1 显示开状态信息。
是否闪烁	否 是	是: 表示闪烁 否: 表示不闪烁
闪烁类型	不闪烁 为 ON 时闪烁 为 OFF 时闪烁 为 ON 时交替闪烁 为 OFF 时交替闪烁 当 “是否闪烁” 为 “是” 时有效,	不闪烁: 指示灯不闪烁 为 ON 时闪烁: 当为地址为 “ON” 时闪烁 ( “是否反向” 为 “是” 时有效, 因为内部默认为 “0” ) 为 OFF 时 闪烁: 当为地址为 “OFF” 时闪烁 为 ON 时交替闪烁: 当为地址为 “ON” 时交替闪烁 为 OFF 时交替闪烁: 当为地址为 “OFF” 时交替闪烁 ( “是否反向” 为 “是” 时有效, 因为内部默认为 “0” )
闪烁频率 (100ms)	无	设置闪烁的速率, 例如里面填 5, 表示的是闪烁的速率为 500ms。
状态	关状态 开状态	位指示灯有两种状态：关状态，开状态

17.2 字状态显示

- 1. 字状态显示开关在部件栏的指示灯中。
- 2. 字状态显示的功能：

字状态显示会根据字的数值不同而切换到不同的状态。

如果值为 0， 则字状态显示的向量图、位图、状态文本都将切换到状态 0；  
如果值为 1， 则字状态显示的向量图、位图、状态文本都将切换到状态 1；  
如果值为 2， 则字状态显示的向量图、位图、状态文本都将切换到状态 2；  
.....

- 3. 字状态显示的属性：

字状态的属性框如软件, 字状态的属性下表



属性名称	属性分类	说明
字地址	PLC 或 HMI 内部寄存器地址	监控字地址的设置
数据格式	无符号十进制 BCD 有符号十进制	无符号十进制: 以无符号 BIN 码的方式读出数据; BCD: 以 BCD 码的方式读出数据; 有符号十进制: 以有符号 BIN 码的方式读出数据;
自动变换	否 是	是: 要自动变换
变换频率 (100ms)	当“自动变换”设为是时, 此项才可用。	变换频率: 即上面各个不同字状态自动变换的频率设置, 例如 5, 表示的就是 500ms。
终点返回	类别: 是/否, 当“自动变换”设为是时, 此项才可用。	否: 循环返回显示 是: 不循环返回显示
是否位触发	否 是	是: 触发 否: 不触发
触发地址	PLC 或 HMI 内部寄存器地址	当触发地址为“1”时有效
状态	64 种状态	字状态显示最多支持 64 种状态

### 17.3 四状态指示灯

1. 四状态指示灯在部件栏的按钮开关中。

2. 四状态指示灯的功能:

四状态指示灯用于监测两个位地址的状态。

将检测到的两个位地址 ON 和 OFF 状态组合起来, 来切换四状态指示灯向量图、HMI 位图、标签的不同状态。

3. 四状态指示灯的属性:

属性名称	属性分类	说明
监测地址 1	PLC 或 HMI 内部寄存器地址	监控位地址的设置
监测地址 2	PLC 或 HMI 内部寄存器地址	监控位地址的设置
状态	四种状态	四状态指示灯有四种状态: 1 开 2 开, 1 开 2 关, 1 关 2 开,

		1 关 2 关，在四种状态下, 可以显示各状态下的文本内容。 (关为 0, 开为 1, 默认初始状态都为 0)
--	--	--

## 18 图形

### 18.1 间接画面显示

1. 间接画面显示在部件栏的图形中。
2. 间接画面显示的功能：

间接画面显示用于显示子画面，其字地址的值就是要显示的画面编号，如果编号为该值得画面不存在，则间接画面显示什么也不显示。比如，如果字的值为 10，但工程中不存在编号为 10 的画面，则间接画面显示什么也不显示，反之，则显示画面号 10 的子画面。

间接画面具有如下特点：

1. 所显示的画面应该是子画面，如果不是子画面，则可能不会正常显示；
2. 如果画面号不存在，则什么也不显示，（相当于把画面显示关闭了）；
3. 最终显示的画面大小与位置，与“间接画面显示”部件的外包矩形是一致的。
4. 间接画面显示的属性：

属性名称	属性分类	说明
触发位地址	PLC 或 HMI 内部寄存器地址	PLC 或 HMI 内部寄存器地址
轮廓	无	部件最小外包矩形的位置、大小
显示为窗口	否 是	是否以 POPUP 的方式显示窗口 是：用户可以拖动窗口的位置，也自行关闭显示的窗口；而不论位值变为 OFF，这种显示看起来很像 Windows 的对话框 否：只是显示的功能，要关闭窗口须由触发位地址控制；

## 18.2 直接画面显示

1. 直接画面显示在部件栏的图形里。
2. 直接画面显示的功能：

直接画面显示用来弹出一个指定的子画面，如果触发位地址所指向的位为 ON，则弹出该子画面，否则，则关闭子画面的显示。

直接画面显示的几个特点：

1. 直接画面显示部件只能用来显示子画面，如果用于基本画面，则有可能无法正常显示。
2. 子画面的显示与否，完全受触发位地址所指向的位值的控制，ON 时则显示子画面，OFF 时关闭子画面。
3. 画面显示的大小与位置，是与部件的外包矩形一致的。
3. 直接画面显示的属性：

属性名称	属性分类	说明
触发位地址	PLC 或 HMI 内部寄存器地址	PLC 或 HMI 内部寄存器地址
轮廓	无	部件最小外包矩形的位置、大小
画面编号	无	直接画面要显示的画面编号

		编号所代表的画面应该是子画面，若是基本画面，可能无法正常显示。
显示为窗口	否  是	是否以 POPUP 的方式显示窗口 是：用户可以拖动窗口的位置，也自行关闭显示的窗口；而等位值变为 OFF，这种显示看起来很像 Windows 的对话框。 否：只是显示的功能，要关闭窗口须由触发位地址控制；

18.3 自定义部件

自定义部件应用说明

—— 以 XXXX 为工程名举例说明

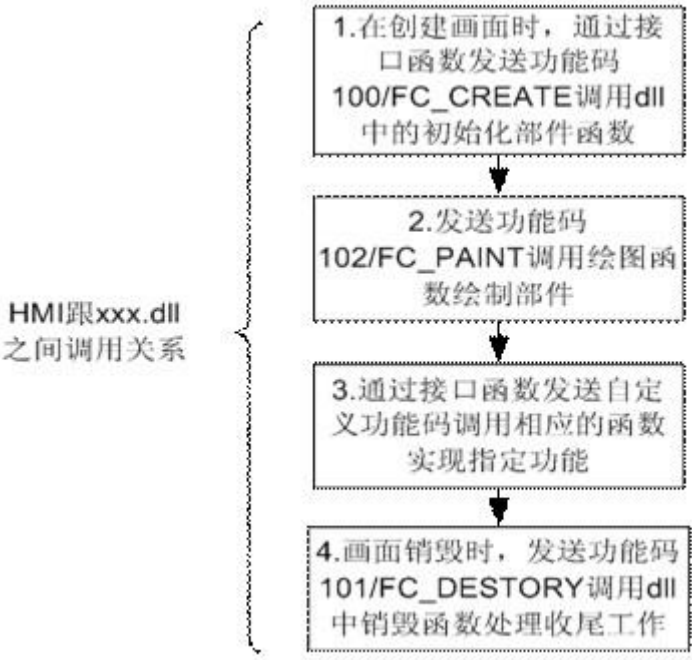
一. 简介

自定义部件是为客户提供的、可自由设计部件的通用接口，满足客户自定义需求。

二. 组成部分

- 1. 在软件上自定义部件的属性为：
  - 1. 轮廓 - 部件的大小；
  - 2. 功能地址 - 部件以此地址中的值作为功能码，处理相应的操作；
  - 3. D11 文件名称 - 选择要使用的 d11（就是客户提供的 d11）；
  - 4. 是否使用定时器 - 默认为否；
  - 5. 定时器时间 - 当上一个属性为是时有效，范围为 100ms 以上。
- 2. 客户最终需要提供的文件 - 两个名为 xxxx.d11 文件和一个说明文档。
  - a. 一个 d11 为 PC 上调用，放到软件安装目录中 d11\_pc 文件夹下。
  - b. 另一个 d11 为 HMI 上调用，放到软件安装目录中 d11\_hmi 文下。
  - c. 说明文档说明此 d11。

三.HMI 和 d11 的调用关系图



四. 自定义部件的一个实例——吹瓶机

1. 吹瓶机介绍：吹瓶机（blow moulding）也称中空吹瓶机，是一种发展迅速的塑料加工方法。吹瓶机的最终产品包括多层复合薄膜和各类聚烯烃中空容器，可广泛用于食物、医药和化妆品等行业。国内的吹瓶机产业主要集中在广东东莞、浙江台州、江苏张家港和上海等地，随着科技的不断进步和生产的规模化，国内吹瓶机产业也沿着从完全仿制转向自我研发，从低端向中高端的趋势发展。

2. 下图为自定义部件属性列表 - 吹瓶机

CST\_0 [自定义部件]

语言一

名称: CST\_0

属性名称	值
一般	
描述	CST_0
轮廓	[120,40],[300,300]
功能地址	HDW10
Dll文件名	Blowing.dll
是否使用定时	是
定时器时间	1000

1. 下图为离线模拟时的图片



可以自由改变大小，设计出符合需求的形状。

五总结:

自定义部件的推出，逐步靠近客户需求，最大限度的实现客户的需要，打造出新的亮点，给客户新的使用感观。竭诚欢迎提供宝贵建

18.4 下拉式清单

- 1. 下拉式清单在部件栏的图形中。
- 2. 下拉式清单的功能：

下拉式清单可以对字地址进行写操作，数值范围为 0~31，部件里用文本来显示对应的值，与列表框不同的是，点击部件右边的下拉框才能罗列出所有的状态文本，设置完后下拉框会自动收回。

属性名称	属性分类	说明
背景色	无	选择下拉式清单的背景颜色
读取地址	无	下拉式清单读取的字地址
写入地址	无	下拉式清单写入的字地址

数据格式	无符号十进制	无符号十进制:写入的值为无符号十进制
	BCD	BCD: 写入的值为 BCD 码
	有符号十进制	有符号十进制: 写入的值为有符号十进制
字体	无	设置各状态文本的字体
粗体	是 否	设置字体是否为粗体
状态	无	共支持 32 种状态
行间距		文本行与行之间的距离
文本内容	无	设置对应状态下的文本内容

18.5 广告走马灯

- 1. 广告走马灯在部件栏的图形中。
- 2. 广告走马灯的功能：

广告走马灯是以走马灯的方式显示文本信息，广告走马灯以特定的速度从右到左滚屏显示文字，周而复始，不受其他条件约束。

- 3. 广告走马灯的属性：

属性名称	属性分类	说明
走马灯速度	低	低：走马灯的流水速度低速度流
	中	中：走马灯的流水速度中速度流
	高	高：走马灯的流水速度高速度流
走向		左到右 右到左
文本		三种语言的文本显示

18.6 打印部件

- 1. 打印部件显示在部件栏的图形中。

功能：打印出打印部件选中范围内的内容。打印部件只能打印屏幕能够显示的内容，看不见的内容不能打印。

建议：打印内容都放在基本画面

- 2. 打印部件的属性：

属性名称	说明
触发位地址	PLC 或 HMI 内部寄存器地址，触发打印的位地址
导出文件触发	PLC 或 HMI 内部寄存器地址，导出打印区域图片的位地址

地址	
导出文件目录	将导出的图片保存在指定目录下, 格式要求: 在线模拟, 将图片保存在 D 盘 aaa 文件夹, 格式是 D: \aaa\ 下载到屏中, 将图片保存在 U 盘, 格式是: /Udisk/ 将图片保存在 SD 卡, 格式是: /CFDC/
轮廓大小	如仅保存为图片, 则大小最大为工程画面大小; 如连接微型嵌入式打印机, 则高度受纸张高度的限制。

18.7 日期显示

- 1. 日期显示在部件栏的图形中。
- 2. 日期显示的功能:  
用于显示 HMI 的系统日期, 星期。
- 3. 日期显示的属性:

属性名称	属性分类	说明
显示格式	显示日期 显示星期	显示日期:显示日期 显示星期:显示星期
日期格式	当“显示格式” 设为显示日期 时有效	日期的显示格式 yy/mm/dd: 年/月/日 mm/dd/yy: 月/日/年 dd/mm/yy: 日/月/年
星期格式	当“显示格式” 设为显示星期时有 效	中文样式一 英文样式一 中文样式二 英文样式二

18.8 时间显示

- 1. 时间显示在部件栏的图形中。
- 2. 时间显示的功能:  
用于显示 HMI 当前的系统时间

18.9 文件列表

文件列表功能简介:

用来实现配方的导入导出, 文件格式是 CSV 格式(具体的格式可以参照 LEVI 工具栏中-配方编辑器)。  
可以实现 SD 卡, CF 卡, U 盘, 和 FLASH 下配方文件的导入导出。



可以生成新的配方文件

可以导出新一组数据到指定文件的文件夹。

可以更改指定文件的某一组数据

通过 HSW505 来控制删除选中的文件或者文件夹。

文件列表的导出文件目录：

U 盘中文件夹 rcp 中的 CSV 文件 /Udisk/rcp/

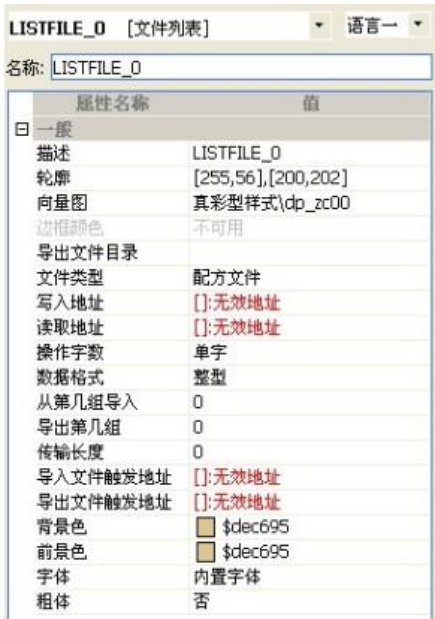
如果直接是 U 盘根目录下的所有 CSV 文件/Udisk/

如果是 SD 卡或者 CF 卡中文件夹 rcp 中的文件/CFDC/rcp/

如果直接是 SD 卡或者 CF 卡根目录下的所有 CSV 文件/CFDC/

1. 导入文件触发位地址：例如 HSX1.4，当此地址的值是 1 时，将从选定文件的指定组导出到 PLC 地址。
2. 导出文件触发位地址：例如 HSX2.4，当此地址的值是 1 时，将读取指定 PLC 地址 并且将指定成份数写入 CSV 文件。
3. 写入地址：比如 HDW0，是将配方中成份导入到 PLC 地址时候，写入到 PLC 的首地址。
4. 读取地址：比如 HDW2000，是读取 PLC 的首地址，将指定的长度写入到 CSV 文件。
5. 导入文件名称（名称是阿拉伯数字）HDW25000（文件名最长是 0-65535 个字）
6. 导出文件名称（名称是阿拉伯数字）HDW25001（文件名最长是 0-65535 个字）
7. 操作字数：单字或者双字
8. 数据格式：整型或者浮点（如果选择浮点时，会被默认为双字）
9. 从第几组导入：是指导入配方到 PLC 时候，从 CSV 的第几组读取数据，默认输入 0 则从第一组导入，内部保留地址 HSW503。
10. 导出第几组：指写入 CSV 文件时候，写入到第几组，如果默认为 0 则写入到 CSV 的最后一组，内部保留地址 HSW504。
11. 传输长度：指成分个数（字个数，如果）

例：数据长度是 10，如果是双字：则没组有 5 个数据值；单字则 10 个数据值。



CSV 文件的格式

1. CSV 文件的第一行用来存储配方的设置参数

E7				
	A	B	C	D
1	1(浮点1, 整型0)	1 (双字1, 单字0)	3 (组数)	300 (成份数)

2. 一个文件最大是 200000 个点，每组分成分最大是 2000.

例如：一个文件列表 200 组，1000 个成分。（不能超过 200000）

工程实例：

一. 简介

文件列表实现了配方导入导出的效果。同时文件列表还包括的文件类型有：配方文件 、 日志文件、记录文件。

二. 工程设置

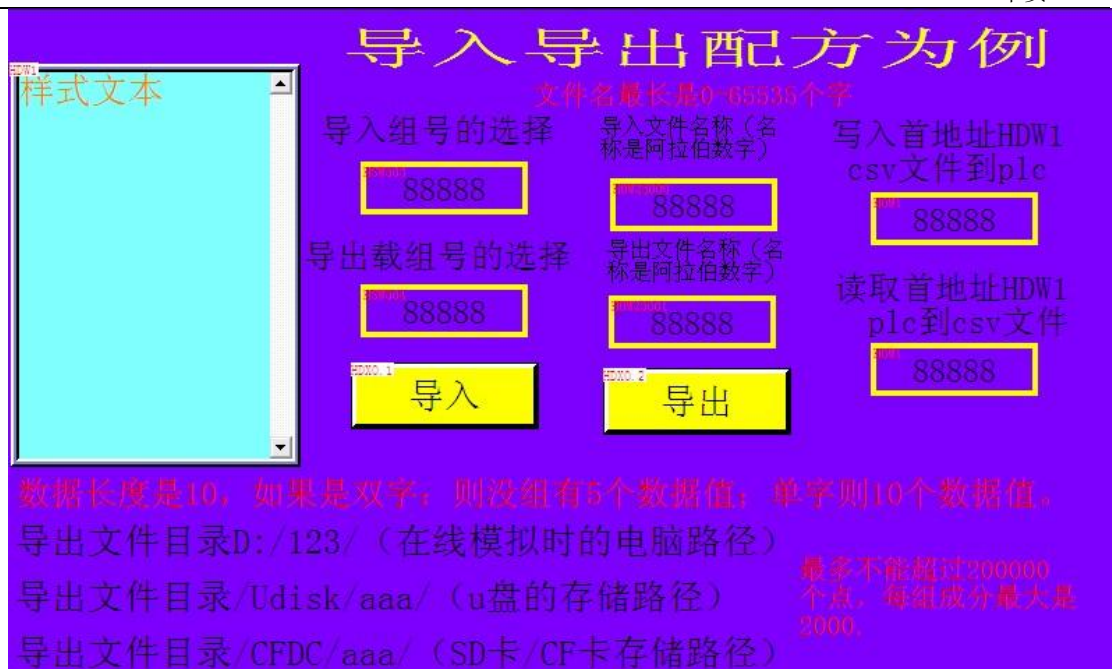
配方编辑 3 个方式：1. 离散配方编辑 2. 手动自己编辑配方文件 CSV 文件（格式要完全相同才能打开，如下图打开成功） 3. 不在以上地方设置, 直接根据工程设置默认生成配方文件。

工具-----配方编辑器，对编辑离散配方的编辑。

新建工程（文件列表）在工程里设置 “文件列表” 相关信息。如下图：



在工程画面中添加“文件列表”部件



工程模拟效果如下图



### 三. 总结:

文件列表配方的文件名最多是 0-65535 个字符, 且文件名地址、组号地址都是固定的特殊寄存器地址。文件存储的格式是 CSV 文件格式, 查看的话直接打开就好了, 不要用数据记录查看工具。存储的目录也必须是以上规定的写法。配方不能超过 20 万个点, 每组分成分最大是 2000。

#### 18.10 配方显示

1. 配方显示在部件栏的图形中。
2. 配方显示的功能:

在 HMI 上可以以表格形式显示当前工程的配方数据，如果工程没有设置配方，则部件的表格将是空的，什么也不显示。

### 3. 配方显示的属性：

属性名称	属性分类	说明
允许输入	否 是	配方的数据是否可以修改
键盘画面号	当允许输入 为是时，有效	当“允许输入”为是时，有效指定用于输入数据的内部键盘类型

## 18.11 列表框

1. 列表框在部件栏里的图形里。

2. 列表框的功能：

列表框可以对字地址进行写操作，数值范围为 0~31，部件里用文本来显示对应的值。

3. 列表框的属性：

部件属性的详细说明如下：

属性名称	属性分类	说明
背景色	无	选择列表框的背景颜色
读取地址	无	列表框读取的字地址
写入地址	无	列表框写入的字地址
数据格式	无符号十进制 BCD 有符号十进制	无符号十进制：写入的值为无符号十进制 BCD：写入的值为 BCD 码 有符号十进制：写入的值为有符号十进制
字体	无	设置各状态文本的字体
部件等级密码	否：正常使用 是：输入密码后，才能使用该部件	是：设置部件等级密码 否：不设置部件等级密码
等级自动降低	是 否	否：不改变部件密码等级。 是：密码输入成功后，密码等级降为一；下次再点击这个部件，则输入等级一或更高等级的密码即可。
部件操作等级		设置部件的密码等级。部件的密码设置在“工程参数”的部件等级密码选项下设置。
粗体	是	设置字体是否为粗体

	否	
状态	无	共支持 32 种状态
文本内容	无	设置对应状态下的文本内容

## 19.1 自绘图形

自绘图形可以极大地增强画面的表现力，也是 HMI 产品所必需支持的基本功能，现在可以支持点、直线、弧、饼图、矩形、折线、多边形、圆、静态文本、弧形刻度、直线刻度、位图等等。可以方便的支持这些自绘图形的编辑、修改。

### 自绘图形

在软件的界面布局上,用户可以通过点击“部件”-“绘图部件”，在画面上制作丰富的自绘图形，如图 19-1 所示。

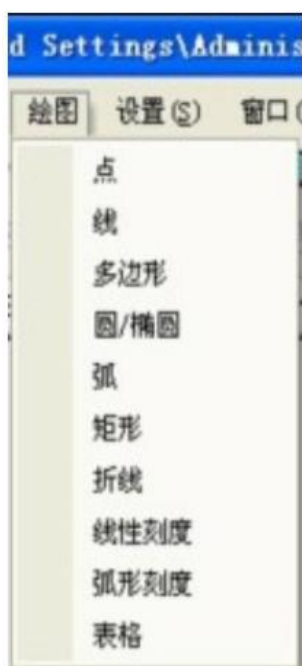


图 19-1 自绘图形部件箱

## 19.2 静态向量图

1. 静态向量图在部件栏的图形中。

2. 静态向量图的功能：

静态向量图只是纯粹的显示向量图，它不和地址有关系。

3. 静态向量图的属性：

要在组态上显示一个静态位图，首先把位图倒入工程的位图库里，当然，系统自带的位图直接引用即可。

位图导入后，软件会自动将位图的色深转换为和 HMI 相匹配的位图，比如对于 705G，位图将转换为 256 级灰度位图；对于 705C，位图将转换为 256 色位图，对于 908T，位图将转换为 16 位真彩位图。

组态支持任意尺寸的位图缩放。





属性名称	属性分类	说明
轮廓		设置向量图的大小和位置。
显示方向	自上而下	自上而下: 正常显示向量图库里的向量图。
	自左而右	自左而右: 向量图向左旋转 90 度显示。
	自下而上	自下而上: 向量图向左旋转 180 度显示。
	自右而左	自右而左: 向量图向右旋转 90 度显示。
向量图		选择向量图库里的向量图。

19.3 弧形刻度

在指定的圆弧区域内显示刻度；弧形刻度需要指定一个圆心坐标；内圆和外圆两个同心圆指定刻度长度和显示区域；起始角度和终止角度说明刻度均匀分布的区域，是按照逆时针走向来分布的。

表 19- 1 弧形刻度的属性表

属性名称	含义	可否编辑
线类型	刻度线的线型	可以在属性框中编辑
边框颜色	刻度线的颜色	可以在属性框中编辑
圆心	弧形刻度所在圆心坐标	
外径	外径指定了刻度区域的外圆	
内径	内径指定了刻度区域的外圆	
起始角度	线性刻度分布区域的起	可以在属性框中编辑



	始角度(相对画面的 X 轴)	
终止角度	线性刻度分布区域的终止角度(相对画面的 X 轴)	
主刻度数	刻度个数, 默认的刻度数是六个	可以在属性框中编辑
次刻度数	在相邻的主刻度之间, 次刻度的数目, 次刻度的刻度线长是主刻度的二分之一。	可以在属性框中编辑

如果要视图上改变弧形刻度的内径、外径、起始角度、终止角度，则只要在在刻度线处于选中的状态下，如图 19-2 所示。

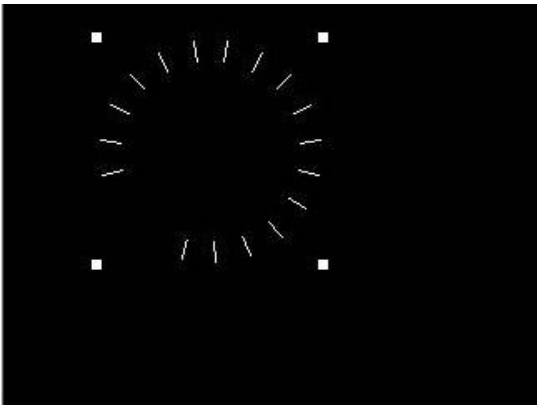


图 19-2 选中弧刻度部件

然后再次在刻度线点击鼠标左键，进入刻度线的编辑选中状态，如图 19-3。

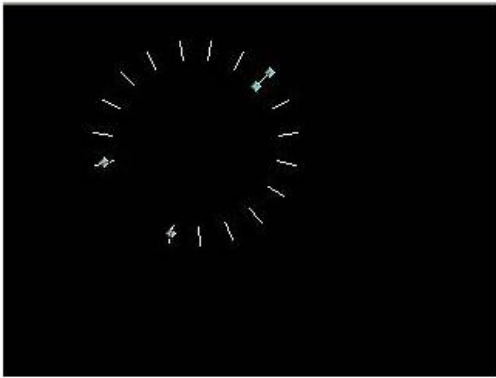


图 19-3 选中弧刻度线

此时共有四个选中点出现，分别用于改变刻度线的起始角度、终止角度、外径、内径。

19.4 线性刻度

表 19-2 线性刻度的属性表

属性名称	含义	可否编辑
显示方向	刻度的开口方向，有四个选择	

	自上而下 自左而右 自下而上 自右而左	
线类型	刻度线的线型	可以在属性框中编辑
边框颜色	刻度线的颜色	可以在属性框中编辑
主刻度数	主刻度数目，主刻度是均匀描绘在刻度区域的	可以在属性框中编辑
次刻度数	在相邻的主刻度之间，次刻度的数目，次刻度的刻度线长是主刻度的二分之一。	可以在属性框中编辑
坐标	X 轴/Y 轴	根据坐标调整位置
大小	宽/高	根据需求调整大小

线性刻度是直线形刻度，将指定的区域用等分线绘制而成，刻度的具体标度可以用静态文本来标注。

### 19.5 圆/椭圆

表 19-3 圆/椭圆的属性表

属性名称	含义	可否编辑
线类型	多边形的线类型，软件提供了十二种线型	可以在属性框中编辑
边框颜色	多边形的颜色	可以在属性框中编辑
填充图案	多边形的填充样式，如果选择“无填充”，则多边形是透明的。	可以在属性框中编辑
前景色	填充图案所使用的前景色	可以在属性框中编辑
背景色	填充图案所使用的背景色	可以在属性框中编辑
X 半轴	椭圆的 X 半轴长度 如果 X 半轴和 Y 半轴的长度相等，则成了圆	可以在属性框中编辑
Y 半轴	椭圆的 Y 半轴长度 如果 X 半轴和 Y 半轴的长度相等，则成了圆	可以在属性框中编辑
圆心	椭圆的中心坐标	可以在属性框中编辑

椭圆的圆心坐标可以在画面视图通过鼠标拖拽来修改。

### 19.6 多边形

表 19-4 多边形的属性表

属性名称	含义	可否编辑
边数	多边形的边数 软件所支持的最大边数为 50	可以在属性框中编辑每个顶点的坐标
线类型	多边形的线类型，软件提供了十二种线型	可以在属性框中编辑
边框颜色	多边形的颜色	可以在属性框中编辑
填充图案	多边形的填充样式，如果选择“无填充”，则多边形是透明的。	可以在属性框中编辑
前景色	填充图案所使用的前景色	可以在属性框中编辑
背景色	填充图案所使用的背景色	可以在属性框中编辑

在创建多边形的过程中，通过点击鼠标右键来封闭多边形，完成一个多边形的创建。多边形创建成功后，如果需要改变多边形的顶点坐标，则只要在在多边形处于选中状态的时候，如图 19-4。

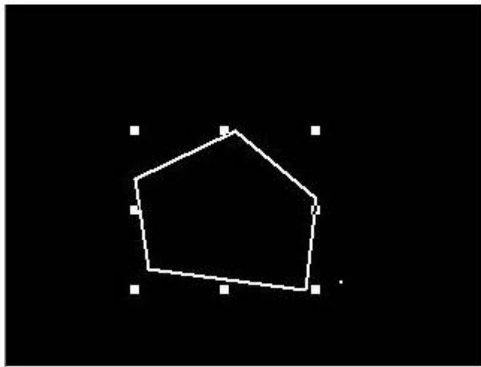


图 19-4 选中多边形部件

然后在多边形的边上再次点击鼠标右键，就变成如图 19-5 中的选中状态。

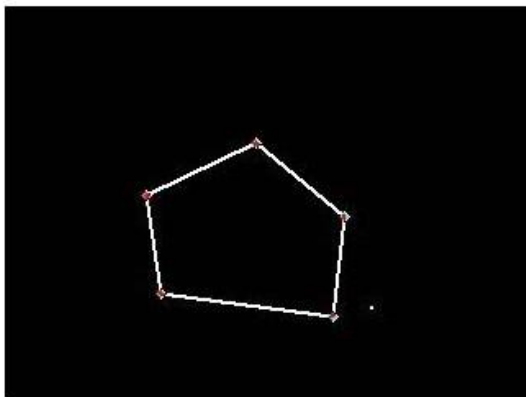


图 19-5 选中多边形

此时每个多边形顶点都是可以用鼠标拖动的。

## 19.7 位图

表 19-5 位图的属性表

属性名称	含义	可否编辑
描述	位图的文字描述	可以在属性框中编辑
起点	位图的起始坐标	可以在属性框中编辑
外观位图	位图名称	

要在软件上显示一个位图，首先把位图倒入工程的位图库里，当然，系统自带的位图直接引用即可。  
位图导入后，软件会自动将位图的色深转换为和 HMI 相匹配的位图。  
软件支持任意尺寸的位图缩放。（位图支持静态位图也支持 gif 图片）

19.8 矩形

表 19-6 矩形的属性表

属性名称	含义	可否编辑
轮廓	矩形的外型尺寸	可以在属性框中编辑
线类型	矩形的边所采用的线型	可以在属性框中编辑
边框颜色	矩形的边的颜色	可以在属性框中编辑
填充图案	矩形的内部填充样式,如果选择”无填充”,则矩形是透明的	可以在属性框中编辑
前景色	填充图案所使用的前景色	可以在属性框中编辑
背景色	填充图案所采用的背景色	可以在属性框中编辑

19.9 表格

表 19-7 表格的重要属性说明

属性名称	含义	可否编辑
是否透明	设置表的背景色是否透明	可以在属性框中编辑
表头交织	表头是否隔列显示不同的背景色。	可以在属性框中编辑
表格交织	表格是否隔列显示不同的背景色。	
交织颜色	设置间隔列的背景色	

表格的行数和列数可以通过属性设置来确定。通过属性设置，可以设置表格的背景色，表格线样式等。表格只能画等间距的行和列。

19.10 直线

表 19-8 直线的属性表

属性名称	含义	可否编辑
边框颜色	直线的颜色	可以在属性框修改值
线类型	直线的线型	可以在属性框修改值
起点	直线的起点	可以在属性框修改值
终点	直线的终点	可以在属性框修改值
自动垂直吸附	单位：像素	设置直线在设定像素内吻合

直线的起点与终点也可以视图上通过拖拽来实现。

### 19.11 文本

表 19-9 静态文本的属性表

属性名称	含义	可否编辑
文本	文本内容 软件的静态文本现支持最多显示 128 个字符、64 个汉字	可以在属性框中编辑
文本颜色	文本的显示颜色	可以在属性框中编辑
显示方向	文字显示方向,有上下左右四个方向	可以在属性框中编辑
字体	文本所采用的字体	可以在属性框中编辑
起点	文本的起始点 对于自上而下的方向,起始点在文本的左上角 对于自左而右的方向,起始点在文本的右下角 对于自下而上的方向,起始点在文本的左下角 对于自右而左的方向,文本起始点在文本的右上角	可以通过拖动编辑

### 19.12 折线

表 19-10 折线的属性表

属性名称	含义	可否编辑
------	----	------

边数	多边形的边数 软件所支持的最大边数为 50	可以在属性框中编辑每个顶点的坐标
线类型	多边形的线类型, 软件提供了十二种线型	可以在属性框中编辑
边框颜色	边的颜色	可以在属性框中编辑

折线是首尾相相接的线段连接而成，其操作和多边形类似，也一样可以通过拖动来改各个顶点的坐标。

19.13 点

表 19-11 点的属性表

属性名称	含义	可否编辑
前景色	点的颜色	可以在属性框中编辑
点类型	表示点的大小，由多少个像素组成，软件支持 1 点、2 点、3 点、5 点	可以在属性框中编辑
点坐标	表示点的位置	可以在属性框中编辑

点的坐标也可以在画面视图上通过拖拽来实现。

19.14 弧

表 19-12 弧的属性表

属性名称	含义	可否编辑
轮廓	指弧所在椭圆的外包矩形	可以在属性框中编辑
边框颜色	弧的边框颜色	可以在属性框中编辑
饼形	弧是否封闭，如果封闭了，就成为饼形	
填充图案	只有弧封闭为饼形时，此项才可用 指定封闭区域的填充样式	可以在属性框中编辑
前景色	只有弧封闭为饼形时，此项才可用 指定填充样式的前景色	可以在属性框中编辑
背景色	只有弧封闭为饼形时，此项才可用 指定填充样式的背景色	可以在属性框中编辑

起点	弧的起点坐标	可以通过拖动编辑
终点	弧的终点坐标	可以通过拖动编辑

软件采用弧的画法是给定起点与终点、采用逆时针的画法。弧的轮廓、起点与终点均可以在画面视图通过拖拽修改。如果要在画面视图拖拽改变弧的起点与终点，首先应该让弧处于选中状态，如图 19-6 所示。

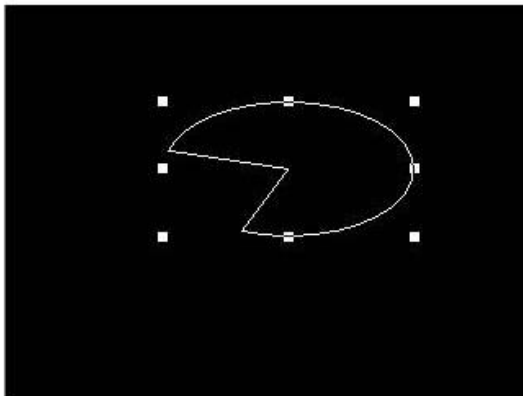


图 19-6 选中弧

然后，在弧线再次点击鼠标左键，弧就变成内部选中状态，如图 19-7 所示。

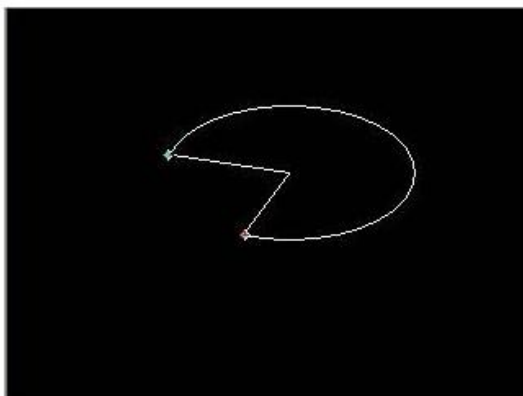


图 19-7 拖动弧的起点与终点

此时，弧的起点与终点是可以拖动的。如果要回到八点选中状态，只需要再次点击鼠标左键即可。

## 20 配方

### 20.1 概述

配方是由多组具备相同结构、不同数据的信息资料而组成的，由于这些资料的相同性，使用者可以将它们编辑为一组配方，以方便在 HMI 和 PLC 之间相互传输数据。因此具备类似的信息资料，可利用此功能以达到这些信息资料有效且正确传送。（另一配方，详见十八章节——文件列表）

下面举一个具体的例子说明配方是如何使用的。

### 20.2 举例说明

下面举一个汽车油漆喷涂的例子，假如一部新制造的汽车需要进行喷漆这道工序时，需要对其车顶部、车底部，车外部三个地方喷上不同（红、绿、蓝）颜色的油漆，而所提供的原始颜色只有红、绿、蓝三色，这时要喷不同的颜色需要这三种颜色适当的配比，而且不同的部位喷的时间不同。这时需要采用配方的功能实现。详见表 20-1。

表 20-1 配方示例

部位	红(公斤)	绿(公斤)	兰(公斤)	喷涂时间 (秒)
车顶部	2	2	1	30
车底部	3	1	2	40
车外部	2			20



		3	3	
--	--	---	---	--

从上表看，喷涂不同的部位时都需要一组配方，这里可以建立三组配方；而每一组配方都有四 4 个成份：红、绿、蓝、喷涂时间；在不同组的配方中，每个成份的值是不同的。

下面将根据这个例子，进行配方信息的具体设置。

20.3 配方信息编辑

为了配方显示部件能够显示配方信息，必须进行必要的设置。具体的操作如下：

- 1. 在“工程管理器”窗口中“工程设置”选项选择“配方”，显示如图 20-1 所示：



图 20-1 工程管理器中的配方

- 2. 双击“配方”图标，出如图 20-2 窗口。  
(最大支持 50 组，1000 个成分，且总的点数不能超过 25\*1000 个点)



图 20-2 配方数据编辑

在上例中车顶部、车底部，车外部 3 组配方，所以选择组数为 3，而每一组配方都有 4 个成员：红、绿、蓝、喷涂时间，所以选连续地址个数为 4，组名依次为如下图：组 1—车顶部、组 2—车底部，组 3—车外部，假设起始地址为 HDW000000，则红→对应地址 HDW000000、绿→对应地址 HDW000001、蓝→对应地址 HDW000002、喷涂时间→对应地址 HDW000003，修改后的结果如图 14-2 所示。

3. 如果要使用非手动的传输配方数据，还需设置触发条件，从 HMI 往 PLC 传输配方数据时，需要设置写触发位地址；若从 PLC 读出配方数据时，需要设置读触发位地址，ON 或 OFF 表示何时有效，是 ON 有效还是 OFF 有效。

4. 设置完成后，按“确定”即可。

5. 在一个 HMI 工程中，依次只能设置一个配方文件。可以对配方文件进行清除、复制、黏贴等编辑操作，还可对配方文件进行导入、导出操作。

6、“数据格式”是软件将数据写入设备或者从设备读出配方的数据格式。支持 BCD 码。

7.、“工具”-----“配方编辑器”-----“离散配方的编辑”



## 20.4 配方显示的添加

当设置好配方信息后，就可利用配方显示部件和配方传输部件进行显示和传输。具体的设置重新详细说明一下。

1、点击“所有部件”中的图形箱中的“配方显示”部件，如图 20-3。



图 20-3 部件图形箱中的配方显示

- 2、在画面上选取适当的位置和大小放置此部件；
- 3、单击此部件，在左边的属性框中填写属性内容，如图 20-4。



图 20-4 配方显示部件属性框

- 4、填写各自所需的属性值，则添加部件完毕，最终的部件的内容如图 20-5 所示。



图 20-5 完成的配方示例

20.5 配方传输的添加

- 1、点击“所有部件”箱中的配方传输部件图标，如图 20-6。



图 20-6 配方传输

- 2、在画面上选取适当的位置和大小放置此部件；
- 3、选中此部件，在左边的属性框中填写属性内容，如图 20-7。



图 20-7 配方传输部件属性框

- 4、填写各自所需的属性值，则添加部件完毕。完成后的结果如图 20-8 所示。

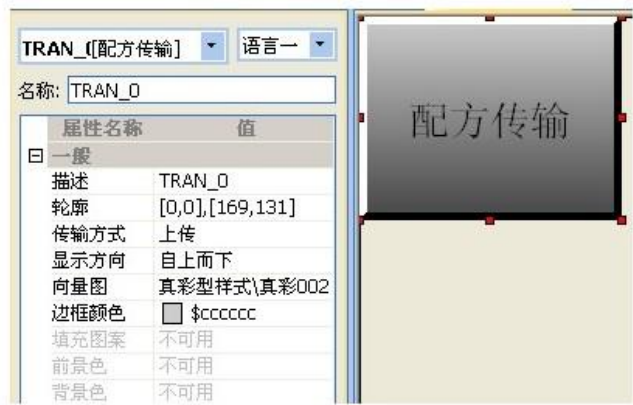


图 20-8 完成后的配方传输的画面

20.6 配方传输的使用

现在以模拟仿真 Emulator 为例，给大家演示一下。配方的显示如图 20-9。



图 20-9 配方显示的效果

如图 20-9，可以在配方显示部件上点击触摸屏来选择当前组号，这个组号将保存在配方组号寄存器 HPW0 上。

如果是手动传输配方部件，则

- 1、首先需要选择配方的组号，我们首先在最左边的组名一栏中选择“车顶部”这组，在上图中，画有矩形框的一组。
- 2、 选择完毕后，在配方传输部件中点击【上传】或【下载】的部件，用于配方数据传输方向的选择：

表 20-2 配方的上传与下载

行为	说明
上传	从 PLC 寄存器传输到 HMI 上，并修改配方文件，如果字 HPW0 的值为 A，那么编号为 A 组的配方数据将被现有需要上传的数据覆盖，并保存配方文件。

下载	将字 HPW0 的值作为组号，从配方文件中找到该组配方，并下载到 PLC 寄存器中去。
----	---

如果在配方文件里设置触发位，则可以实现配方的自动传输

- 1、为了使配方数据能够自动的传输，首先应当满足在【配方信息】设置中读写的触发条件；
- 2、其次，还需要选择配方的组号，在我们 HMI 寄存器规划中存放这个配方组号的寄存器地址是 HPW000000，当满足配方传输的条件时，我们会从 HPW0 这个寄存器中读取值，这个值就是我们要进行配方数据传输操作的组号。
- 3、这样就可进行配方数据的自动传输操作。

20.7 配方数据的索引

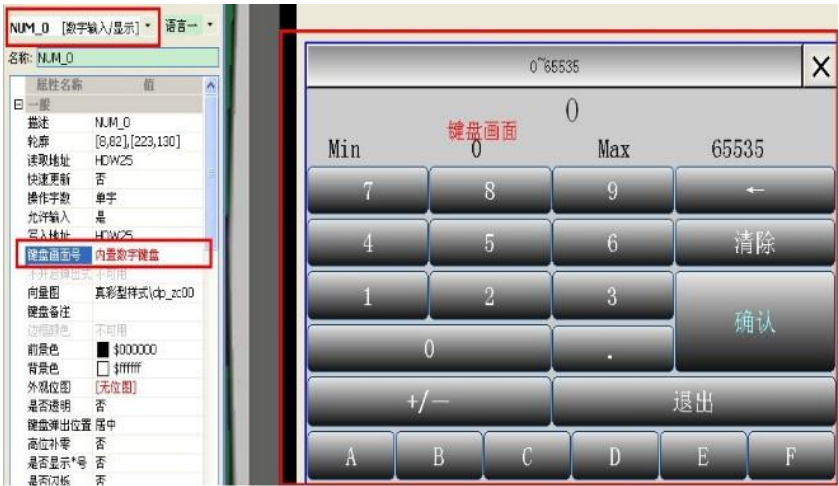
配方的任何数据可以通过软件的特殊寄存器 RPW 来索引，只要在使用“数值输入/显示”或者“字开关”部件中使用这个地址，就可以操作配方文件。

21 记录区信息

21.1 WebSever 数据记录区

软件提供的数据输入部件有两种：字符输入/显示部件、数字输入/显示部件(此外，配方显示部件也可以输入数据)，数据的输入必须依赖键盘画面。

在软件中，画面只有基本画面和子画面的区别，键盘画面也是一般的子画面，但与一般子画面不同的是键盘画面必须包含按键部件，如果被指定用于键盘画面的子画面不包含按键部件，则该键盘画面将不能输入任何东西。（键盘画面 如下图所示）



21.2 实时圆盘记录图信息

实时圆盘记录图信息用于录入与保存圆盘记录部件的相关信息，软件中通过部件圆盘记录图、历史圆盘记录图来读取。

添加实时圆盘记录图信息的对话框如下图：

- 1、一组记录最多支持采集 4 条曲线。
- 2、可以通过“是否位触发”来触发该组记录的采集。
- 3、“采样时间”的单位为 0.1 秒，“轮转时间”指实时圆盘记录图画一圈的时间。

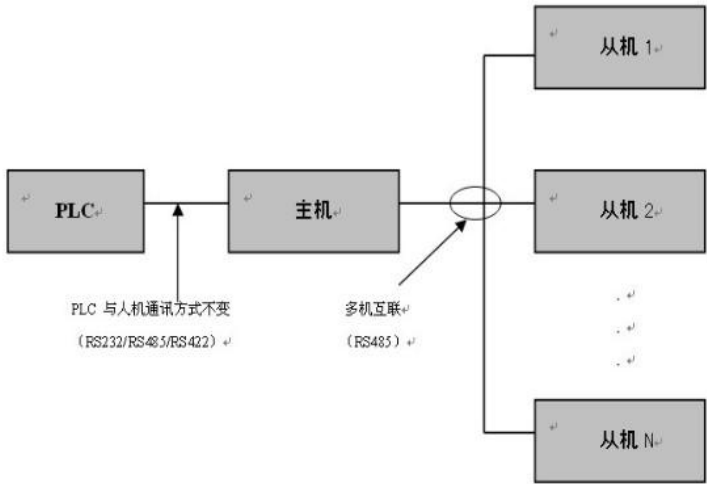


21.3 多机互联（Multi-Link）使用说明

多机互联（Multi-link）提供了一种既经济又方便的解决方案，使多台人机得以通过 RS485 连线来通讯。在多机互联操作中，一台人机为主机（MASTER），其它为从机（SLAVE）。作为主机的人机是唯一与 PLC 通讯的，其接收来自 PLC 的数据都通过主机（MASTER）传输到各个从机中，因此每一从机都要设置一个通讯站号，使得主机能与各个从机进行交互，从而与 PLC 交互。

1. 连接方式

下图以一台 PLC 与四台人机通讯的范例。注意：主从机通常通过 RS485 连接，且每个从机需要设置不同的通讯站号。当从机只有一台时，也可以用 RS232 相连。





2. 多机互联通信参数设定

以下是设定多机互联通信参数的操作步骤：

- 1. 选择“设置”--“通讯口设置”，在页面中选择添加新连接，在弹出的对话框中选择多机互联的通讯接口，在“设备类型”中选择“Multi-Link Protocol”这个协议。
- 2. 设定主从机，以及相应的从机数量和从机站号，如图 21-1。

注意：设置从机数量越多，通讯速度可能越慢。

从机站号必须从 1 开始的连续数字，不可重复。

主从机的接口通讯参数必须一样。

主从机在选择协议时，必须保证协议的编号是一样的，如图 21-2。

串口设备参数设置:

连接方式: RS485

波特率: 115200

停止位: 1

数据位: 8

校验位: NONE

以太网设备参数设置:

PLC IP地址:

PLC 端口号: 0

网络类型: TCP\_Client

☐ 使用广播地址

广播地址站号: 0

预设地址/站号

HMI站号: 0 PLC站号: 0

☒ 启用多机互联

☐ 主机 从机数量: 2

☒ 从机 HMI站号: 1

超时设置

等待超时(ms): 300

接收超时(ms): 50

重试次数: 2

重试超时(s): 3

通讯延迟时间(ms): 30

连读长度(0则使用默认): 0

图 21-1

请选择设备类型:

编号	通信口	设备类型
1	COM1	MIT FX2N NOPROTOCOL
2	COM2	Multi-Link Protocol

图 21-2

3. 注意事项

使用多机互联时，请注意以下几点注意事项：

- 请保证主、从机的工程内容是一样的。**除了主从机参数的设置。**
- 为了保证工程正确的运行，建议不使用脚本。若要使用脚本，脚本中使用的外部地址必须添加到“设置”--“多级互联数据区”。



- 主从机的接口通讯参数必须一样。
- 主从机在选择协议时，必须保证协议的顺序编号是一样的。
- 从机站号必须从 1 开始的连续数字，不可重复。
- 设置从机数量越多，通讯速度可能越慢。
- 使用多机互联时，建议使用 RS485 通信方式，当只有一台从机时，也可以使用 RS232。

21.4 历史 XY 趋势图

历史 XY 趋势图用于录入与保存历史 XY 图的相关信息。

添加历史 XY 趋势图信息的对话框如下图：

- 1、一组记录最多支持采集 12 个字地址。
- 2、可以通过“是否位触发”来触发该组记录的采集。
- 3、采样时间的单位为 0.1 秒，“记录总数”指趋势图或百分比趋势图能显示的最多数据点数。

The dialog box is titled "新增记录条目" (Add New Record Item). It contains the following sections:

- 记录编号:** 新增 (Record Number: New)
- 地址信息:** (Address Information)
  - ☒ 连续地址 (Continuous Address): Includes a text field, an ellipsis button, and a "曲线数目: 1" (Number of Curves: 1) dropdown.
  - ☐ 不连续地址 (Discontinuous Address): Includes a grid of 12 address fields (0-11), each with an ellipsis button.
- ☐ 是否位触发 (Whether Bit Triggering): Includes a "位触发地址" (Bit Triggering Address) field with an ellipsis button.
- 显示:** (Display)
  - 显示格式: 无符号十进 (Display Format: Unsigned Decimal) with a dropdown.
  - 小数点位置: 40 (Decimal Point Position: 40) with an ellipsis button.
  - 注: 百分比趋势图支持 12 条曲线, 实时曲线图只支持前 4 条曲线 (Note: Percentage trend graph supports 12 curves, real-time curve graph only supports the first 4 curves).
- 采样:** (Sampling)
  - 采样时间(t): 1 (Sampling Time(t): 1) with a dropdown and a multiplier of (1-9999) X 100ms.
  - 记录总数(s): 100 (Record Total(s): 100) with a dropdown and a multiplier of (1-1000).
- Buttons: 确定(O) (OK) and 取消(C) (Cancel).

21.5 趋势图信息

趋势图信息用于录入与保存趋势图的相关信息，软件中通过部件趋势图、百分比趋势图来读取。

添加趋势图信息的对话框如下图：

- 1、一组记录最多支持采集 12 个字地址。
- 2、可以通过“是否位触发”来触发该组记录的采集。
- 3、采样时间的单位为 0.1 秒，“记录总数”指趋势图或百分比趋势图能显示的最多数据点数。

新增记录项目

记录编号： 新增

地址信息

☒ 连续地址

... 曲线数目： 1

☐ 是否位触发

位触发地址  ...

☐ 不连续地址

0  ... 1  ... 2  ... 3  ...

4  ... 5  ... 6  ... 7  ...

8  ... 9  ... 10  ... 11  ...

显示

显示格式： 无符号十进制 ... 小数点位置： 40 ...

注：百分比趋势图支持12条曲线  
实时曲线图只支持前4条曲线

采样

采样时间(μs)： 1 ... (1-9999)X100ms 记录总数(s)： 100 ... (1-1000)

确定(O) 取消(C)

## 22 安全保护

### 22.1 概念及用途

在默认情况下，用户可以访问所有 HMI 的画面，操作所有部件而没有任何限制，但有的工业场合需要对包含敏感数据或敏感动作的画面加以限制，对画面当中的某些部件的操作加以屏蔽，使得没有权限的人不能查看这些数据，不能进行某些敏感的操作。

软件提供了两种现场安全保护的方案：一种是基于画面安全等级的模式；一种是基于按键密码的模式。

画面安全等级是以画面作为安全保护的基本单位，每个画面（无论是基本画面还是子画面）都有自己的安全等级。系统以默认的安全等级启动工程，如果要访问高安全等级的画面必须输入高等级的密码。

按键密码模式是当 HMI 在设定时间内没有触摸动作，点击 HMI 的部件则不起作用。只有点击设置为“密码保护”的功能开关，弹出密码输入键盘，输入正确的密码后，点击 HMI 的部件才有效。

同一个工程只能选择一种安全模式，关于安全模式的选择，从“设置”--“工程参数”--“安全等级及密码”来选择所需要的安全保护模式。



图 22-1 工程的安全等级配置

### 22.2 画面安全等级模式

软件提供了十二个安全等级，每个等级都有自己的密码，每个画面都有自己的安全等级，如下图所示：



图 22-2 画面的安全等级

软件的安全等级分为十二级，安全程度依次增加。如果软件当前访问权限等级是二级，则比这个安全等级低(包括同级)的画面都是可以访问的，而比这个等级高的则是禁止访问的。

安全等级对画面与子画面都是有效的，软件在加载任何画面时都会将 HMI 当前的安全等级与该画面的安全等级相比较，如果画面的安全等级高于当前等级，画面(子画面)将不会显示并且 HMI 将弹出密码键盘，而低于当前等级的画面都能够正常显示。

从“设置”--“工程参数”--“安全等级及密码”，可以设置工程的密码及安全等级，如下图所示：



图 22-3 工程的安全保护

使用安全保护：启用工程的安全保护机制，在默认情况下不启用安全保护等级，这种情况所有的画面都可以随便访问；

画面密码保护：使用画面的安全等级模式来保护画面

启动等级：系统启动时缺省的密码等级。

各等级独立密码：如果设置此项，那么每次从低等级画面进入高等级画面，都需要输入密码；否则只需要在第一次进入高等级画面时输入。

按键密码保护：使用按键密码保护的 mode 来保护工程

密码：激活画面触摸有效的密码

按键超时：如果在此时间内没有用户响应，则 HMI 将进入按键保护状态。

设计者密码：用户从机器上上传应用时，应该输入的密码。

需要注意的是这个十二个等级的密码最多不超过八个字符；并且最好互不相同，如果有两个等级的密码相同，用户输入这个密码后，当前的安全等级将被置为较低的那个等级。

22.3 按键密码保护模式

如果安全模式是选择按键密码保护，那么在设置超时时间内无用户操作的话，触摸屏会进入屏幕保护状态，这时点击画面的部件无效，只有点击设置为“密码保护”的功能开关，在弹出的密码键盘中输入正确的密码，触摸屏的部件才能恢复有效状态。

功能开关有一个特定的功能就是用来输入按键的密码。如图 22-4



图 22-4 用功能开关输入密码

23 脚本

## 23.1 概述

在很多工程应用中，为了实现复杂的控制功能光靠部件是没有办法满足这些需求的，而脚本就是用编程的手段利用人机界面的系统资源来实现这些复杂控制。

开发软件提供了功能强大、使用简单、效能可靠的脚本系统，开发软件 I 的脚本具备如下特点：

1、采用与 BASIC 相类似的语法结构；

BASIC 做为第一种面向普通大众的计算机高级语言具有语法简单、易学易用、实时高效等特点，BASIC 在 IT 各界有着广泛的应用，BASIC 不需要初学者投入太多的精力就可以轻松掌握。

2、支持所有程序逻辑控制结构；

软件脚本支持顺序、条件、循环等三大逻辑控制结构，因为可以实现任何复杂度的程序。

3、强大的函数功能；

脚本的函数功能分为两大类：系统函数和自定义函数。

系统函数指将某些常用的系统函数做成函数的形式，可以在脚本中随意引用。这些函数有 BCD 码与 BIN 码的转换等等。

自定义函数是指用户将频繁调用的一段程序代码封装成函数的形式，可以在所有的脚本中调用。

4、支持 IEC61131 架构；

IEC61131 是 PLC 编程语言的一个国际标准，如果支持这个标准的 PLC 那么他的程序可移植性和可操作性是可以脱离平台的。软件支持 IEC61131 架构，这使得脚本与支持 IEC61131 架构的 PLC 之间移植程序成为可能。

5、支持多种数据类型；

脚本支持整形、浮点、BCD 码、字符、字符串等多种数据类型。

6、简单易学、功能强大、性能可靠

## 23.2 使用脚本应该注意的问题

开发软件平台使用方便、简单灵活、性能可靠，是现实复杂 HMI 应用的不可或缺的帮手。但应该正确使用，如果使用不当，脚本会消耗大量的 CPU 时间，并最终影响整个工程的效率。

总得来说，有以下几个问题需要谨慎处理。

1、在脚本中尽量不要使用过大的循环，如果脚本执行一次的循环次数太多，势必影响 HMI 在其他方面的效率(尤其是画面响应和数据采集的效率)；

2、在脚本中尽量不要频繁访问外部寄存器(如 PLC 的寄存器等)，由于串口通信比较慢，如果频繁访问外部寄存器，则会导致脚本执行效率的严重降低。甚至也影响画面响应效率，当然，由于本地寄存器(HSX, HDX, HPX, HSW, HDW, HPW, RPW)就是内存本身，所以不存在效率问题；

3、虽然每个画面的定时器(位触发脚本)分别最多允许有 32 个，但由于画面可以嵌套子画面，所以，同时运行的脚本个数可能远远不止这个数。但为了保证性能和效率，在任何时刻，同时运行在软件上的脚本以不超过 16 个为宜；

4、每个脚本的最大长度现在 512 行，但在实际使用中，脚本过长也会导致效率下降。

## 23.3 脚本对设备的访问

软件脚本可以提供了一种直接有效的办法来设备地址，这种办法是通过@符号来对地址进行直接访问。

表 23-1 脚本对设备的访问

写法	含义	举例
@B_ ;@b_ ;	对某个位进行访问	@B_I0.0: 访问位地址 I0.0 @b_HDX0.0: 访问位地址 HDX0.0
@W_ ;@w_ ;	对某个字进行访问	@W_IW0: 访问字地址 IW0 @b_HDW0: 访问字地址 HDW0
@B_连线#站号:地址 @W_连线#站号:地址	当 HMI 挂接了多个自控设备时，“#”前面表示连线即设备，“:”前面表示要访问的站号。 如果没有“#”即表示访问主连线，如果没有“:”即表示访问缺省站号	@B_2#2:I0.0: 访问连线号为2的站号为2的位地址 I0.0; @W_2#2:IW0: 访问连线号为2的站号为2的字地址 IW0; @B_I0.0: 访问缺省连线上的缺省站号的位地址 I0.0;

脚本对设备的访问有两种方式：读和写。具体采取何种方式来访问控制设备，这取决于软件脚本的分析结果。

比如下面的代码：

```
if @B_HDX0.0 = 1 then    ‘从寄存器读取 HDX0.0 的值
@B_HDX0.0 = 0          ‘将 0 写往寄存器 HDX0.0
else
@B_HDX0.0 = 1    ‘将 1 写往寄存器 HDX0.0
Endif
```

而下面的代码则是既有读也有写：

```
@W_QW0 = @W_QW0 + 1 ‘先从寄存器 QW0 读出数据，将该数加 1，再写往 QW0
```

注意，这种访问方式对于位来说，只能访问一个位；对于字来说，只能访问一个字，如果访问长度大于 1，则必须用到软件提供的访问控制设备的系统函数。

对于 HMI 内部寄存器的访问，不需要连线号和站号。

## 23.4 第一次使用脚本

下面我们通过一个很简单的例子来说明软件如何使用脚本的。例子：让一个位状态指示灯不停地闪烁。

首先新建一个工程，PLC 型号选择“SIMUTOCOL”，程序模板选择“自定义”，然后在画面上放置一个“位状态指示灯”，并将位状态指示灯的监控地址设置为 hdx0.0，这是 HMI 的内部寄存器地址。位状态指示灯的其他属性如下图所示：



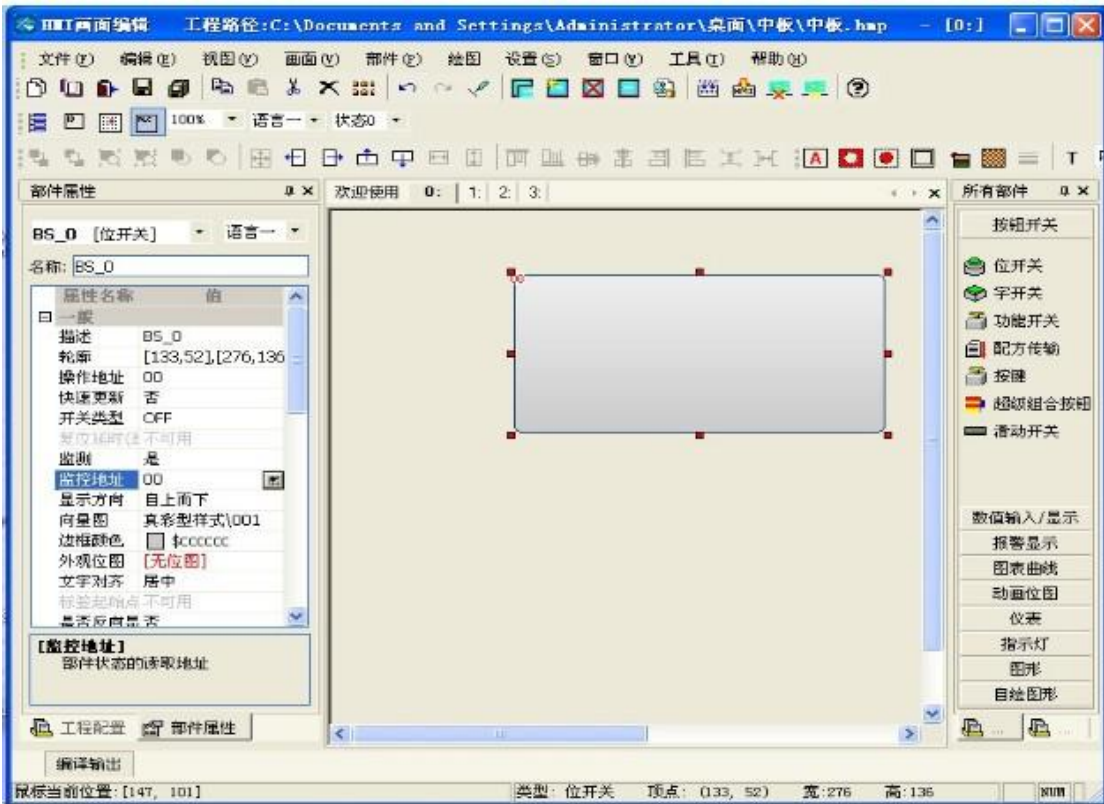


图 23-1 第一次使用脚本的工程

然后，编译，选择离线模拟，发现位状态指示灯一直处于灯灭的状态，如果加入了脚本，则可以让等闪烁起来。

那怎么加入脚本呢？下面几个办法：

办法一：在画面视图上点击右键，从弹出菜单上选择“画面脚本”，如下图；



图 23-2 右键弹出菜单打开脚本编辑器

办法二：从工具条中选择点击工具 进入当前画面的编辑器。如下图所示：





图 23-3 从工具条打开脚本编辑器



办法三：从工程管理器中所要编辑的画画，然后点击右键，选择“画面脚本”或工程管理器的 MiniTool 工具条选择，就可以打开画面的脚本编辑器，如下图所示：



图 23-4 从工程管理器打开脚本编辑器

进入画面编辑器后，从脚本工具条中选择，添加一个靠定时器触发运行的脚本，触发周期是 0.5 秒，如下图所示，

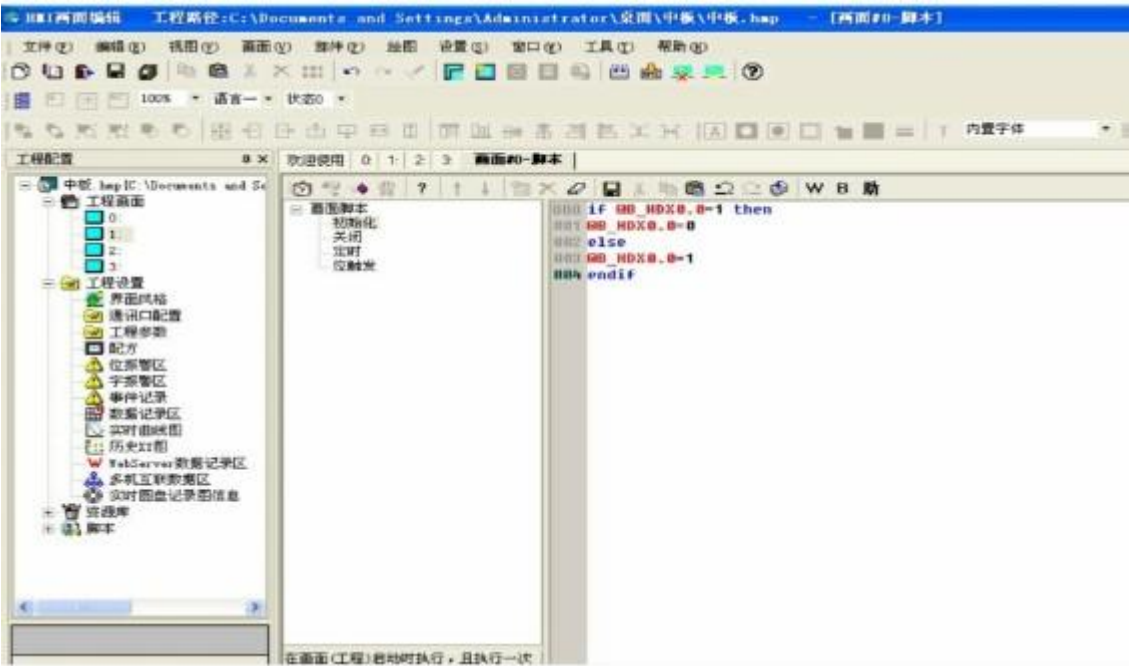



图 23-5 脚本编辑器

然后在脚本视图上写上如下代码：@符号表示访问 HMI 或设备的寄存器，B\_表示是以位的方式来访问。

```
if @B_HDX0.0 = 1 then
  @B_HDX0.0 = 0
else
  @B_HDX0.0 = 1
endif
```

图 23-6 定时触发脚本

这段脚本的意思就是将位 HDX0.0 的值在 0 和 1 直接来回切换，切换的频率多少，取决于脚本触发器的频率，在这里例子中，采用的是每隔 0.5s 触发一次脚本。

然后从脚本编辑工具条中选择  对脚本进行语法检查，只有经过语法检查的脚本才能确保在上正确运行。结果语法检查没有发现任何问题。

然后，“保存”→“编译”，选择“离线模拟”，位开关就开始闪烁了。

从这个简单的例子知道，脚本是在画面背景脚本下运行的。

## 23.5 初识脚本编辑器

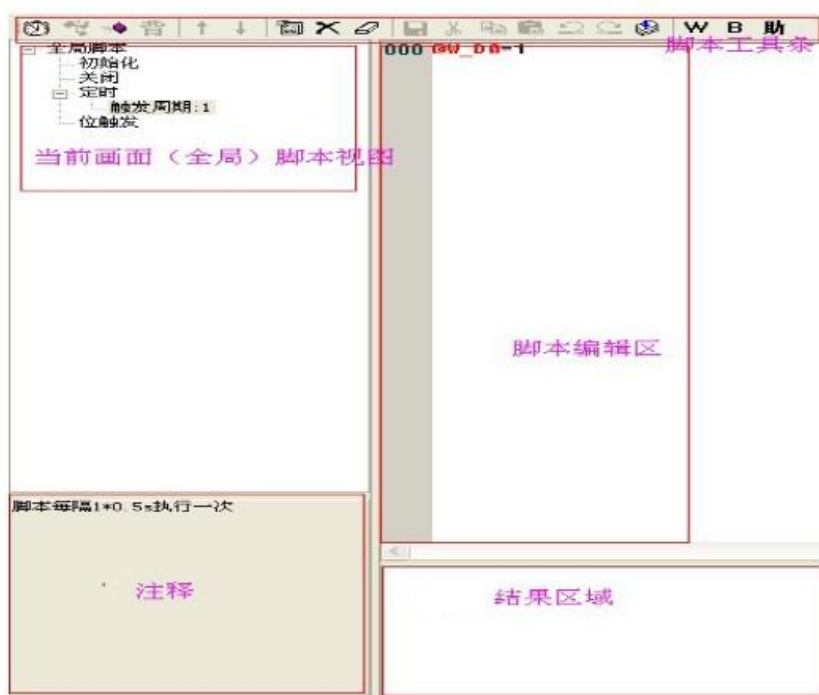


图 23-7 脚本编辑器的概貌

**脚本工具条：**用于新建脚本、对脚本进行编辑。

**当前画面(全局)脚本视图：**当前画面(全局)已经有的脚本，以树形的方式列举出来。

**脚本编辑区：**编写脚本的视图。

**注释：**当脚本视图选择一个脚本时，在注释区域就会出现这个脚本的运行条件等等。

**结果区域：**对脚本进行语法检查的结果将列举在这里。

脚本工具条：



图 23-8 脚本编辑工具条

- ：新建一个定时触发的脚本。
- ：添加一个自定义函数，仅仅在打开全局函数时有效。
- ：添加一个位触发脚本。
- ：查看一个脚本的属性。
- ：删除脚本视图中处于选中状态的脚本。
- ：清空当前的脚本内容，清空后将不可恢复。
- ：保存当前正处于编辑的脚本。
- ：将选中的文本内容剪贴到系统剪贴板上。
- ：将选中的文本内容复制到系统剪贴板。
- ：从系统剪贴板复制文本内容到脚本编辑器。
- ：取消当前操作。
- ：重做已经取消的操作。
- ：对当前脚本进行语法检查，检查的结果将在“**结果区域**”列举出来。
- ：用地址编辑器在脚本编辑区里输入字地址。
- ：用地址编辑器在脚本编辑区里输入位地址。
- ：通过脚本生成帮助对话框来添加函数。

脚本视图：

- 脚本视图列举了当前画面(全局)的所有脚本。
- 初始化**：工程(画面)的初始化时运行的脚本；
- 关闭**：工程(画面)关闭时运行的脚本；
- 定时**：所有的定时触发运行的脚本节点；
- 位触发**：所有的靠位触发运行的脚本节点；
- 鼠标“**双击脚本节点**”就可以打开相应的脚本进行编辑

23.6 脚本的分类

脚本分三类：一类是背景脚本，只有定时触发脚本，运行时间间隔最小为 0，即开机后这段脚本一直在运行；一类是在工程存在期间运行的脚本，称为全局脚本，当加载工程时，脚本就开始运行，直到工程结束，定时触发脚本的运行时间间隔最小为 0.5 秒；另一类就是在画面存在期间运行的脚本，称为画面脚本。当画面被加载时，画面脚本就开始运行，直至画面被关闭（或切换到其他画面）。

背景脚本只有周期脚本运行时间间隔的单位为 1 毫秒，可以设置多个定时触发运行的背景脚本，在需要访问、操作外部地址时，脚本写在背景脚本里。

要打开背景脚本进行编辑，只要从“工程管理器”双击“背景脚本”即可。点击“背”图标，添加背景脚本，设置运行间隔，即可进入脚本编辑画面进行编辑。

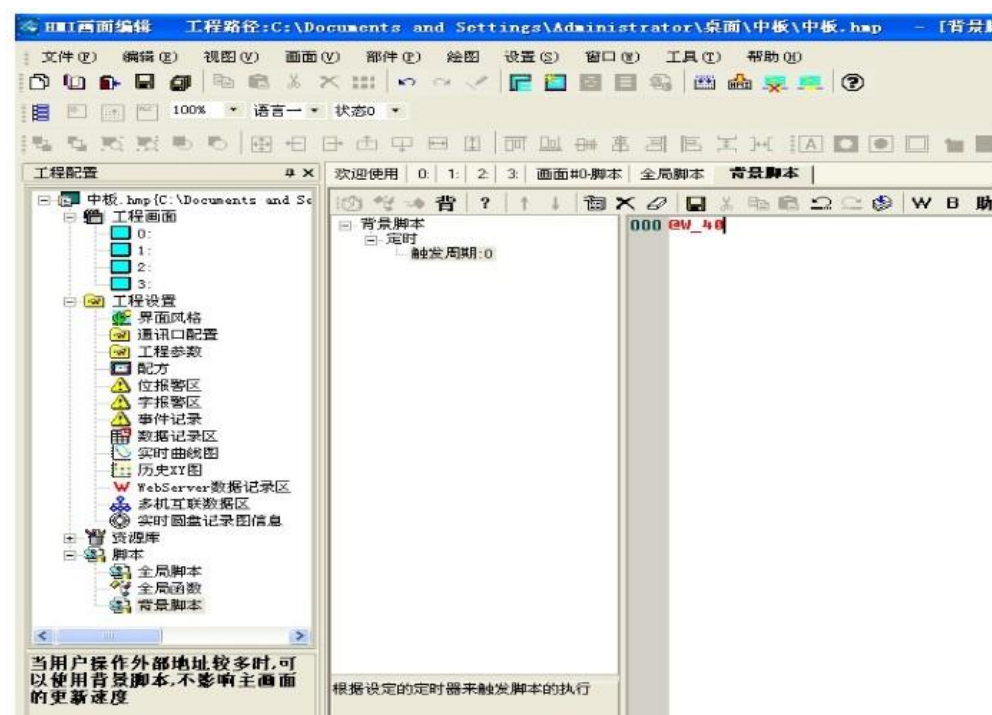


图 23-9 打开背景脚本

**全局脚本**从运行条件分，可以分为四大类：

**初始化：**在工程加载时运行，这个脚本只在工程加载的时候运行，且只运行一次；

**关闭：**在工程关闭时运行，这个脚本只在工程关闭的时候运行，且只运行一次；

**定时：**在工程运行期间，每隔一段时间(如 0.5s)就触发运行，直至工程结束。每个工程可以设置多个定时触发运行的全局脚本。

**位触发：**在工程运行期间，当指定的位满足位触发条件时而执行的脚本，只要满足位触发的条件，脚本将被重复运行。位触发有四个条件：

**TRUE：**不停检测触发位的值，只要该值为 TRUE，脚本将被执行；

**FALSE：**不停检测触发位的值，只要该值为 FALSE，脚本将被执行；

**上升沿：**不停检测触发位的值，当该值从 FALSE 变 TRUE，脚本将被执行；

**下降沿：**不停检测触发位的值，当该值从 TRUE 变 FALSE，脚本将被执行；

在一个工程中，位触发的全局脚本也可以有多个。

要打开全局脚本进行编辑，只要从“工程管理器”双击“全局脚本”即可。



图 23-10 打开全局脚本

画面脚本从运行条件分，也可以分为四大类：

**初始化：**在画面加载时运行，在画面存续期间，这个脚本只执行一次；

**关闭：**在画面销毁或关闭时运行，在画面存续期间，这个脚本只执行一次；

**定时：**在画面存续期间，每隔一段时间(如 0.5s)就触发运行，直至画面销毁或者关闭。

每个画面可以设置多个定时触发运行的全局脚本。

**位触发：**在画面存续期间，当指定的位满足位触发条件时而执行的脚本，只要满足位触发的条件，脚本将被重复运行。位触发有四个条件：

**TRUE：**不停检测触发位的值，只要该值为 TRUE，脚本将被执行；

**FALSE：**不停检测触发位的值，只要该值为 FALSE，脚本将被执行；

**上升沿：**不停检测触发位的值，只要该值从 FALSE 变 TRUE，脚本将被执行；

**下降沿：**不停检测触发位的值，只要该值从 TRUE 变 FALSE，脚本将被执行；


在一个画面中，位触发运行的脚本也可以有多个。

子画面和基本画面的脚本分类以及运行机制是一样的。

从上面的分类可以看出，全局脚本和画面脚本的分类是一样的，区别仅仅在于：画面脚本仅在画面存续期间运行；而全局脚本则在工程存续期间运行。

### 23.7 初始化脚本

初始化脚本是在工程(画面)初始化时运行的脚本，在工程(画面)存续期间只运行一次。从“脚本树形视图”双击“初始化”节点，就可以打开脚本进行编辑。

如果工程(画面)初始化时不想运行“初始化”脚本，只要从点击“”清空脚本内容，然后保存就可以了。

### 23.8 位触发脚本

位触发脚本是指在工程(画面)存续期间，软件每隔约 20ms 就会去检查指定的位是否满足触发条件，

如果满足触发条件，就执行脚本一次，所以，位触发脚本只要满足触发条件，就会被执行，直到工程(画面)关闭。

每个画面的位触发脚本最多允许有 32 个，每个工程的全局位触发脚本最多也是 32 个。

新建位触发脚本


打开全局(画面)的脚本编辑器，然后从工具条中点击，接着弹出如下图所示的对话框：



图 23-11 新建位触发脚本对话框


**位地址：**触发器要监控的位。

触发条件的含义如下表所示：


表 23-1 位触发条件的含义

触发条件	含义
TRUE	每个约 20ms 检测监控位，只要该位值为 TRUE，脚本就执行一次；
FALSE	每个约 20ms 检测监控位，只要该位值为 FALSE，脚本就执行一次；
位改变	当某个位发生变化时候执行一次
上升沿	每个约 20ms 检测监控位，只要该位值从 FALSE 变为 TRUE，脚本就执行一次；
下降沿	I 每个约 20ms 检测监控位，只要该位值从 TRUE 变为 FALSE，脚本就执行一次；

删除位触发脚本

从脚本视图中选择一个位触发脚本，然后从工具条中选择点击删除，删除后的脚本将不可恢复。

编辑位触发脚本属性

编辑定时器脚本属性指的是修改触发条件和监控位，从脚本视图中选中脚本，然后从工具条中点击，修改完毕后，点击“确定”保存。

23.9 关闭脚本

关闭脚本是在工程(画面)关闭时运行的脚本，在工程(画面) **存续期间只运行一次**。从“脚本树形视图”双击“关闭”节点，就可以打开脚本进行编辑。



如果工程(画面)关闭时不想运行关闭脚本, 只要从点击“”清空脚本内容, 然后保存, 就可以了。

## 23.10 定时脚本

定时脚本是在工程(画面)存续期间定时运行的脚本, 定时器的触发脚本每个画面最多允许有 32 个, 全局的定时脚本最多也是 32 个。

### 新建定时器脚本

从点击脚本编辑工具条中的图标, 会弹出下面的对话框:

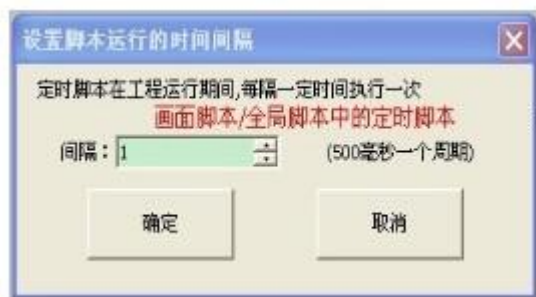


图 23-12 新建定时器脚本

**间隔:** 每个多长时间脚本运行一次, 单位是: 500ms (0.5s)【背景脚本中的“定时”脚本的定时间隔为: 1ms】。

**确定:** 新建一个定时器脚本。

**取消:** 取消新建。

新建一个定时器脚本后, 在“脚本视图”中的定时节点会列举出来。

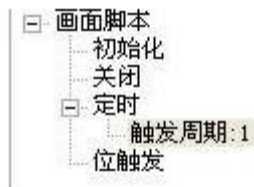




图 23-13 新建定时器脚本后

### 删除定时器脚本

从脚本视图中选择一个定时器脚本, 然后从工具条中选择点击删除, 删除后的脚本将不可恢复。

### 编辑定时器脚本属性

编辑定时器脚本属性指的是修改定时器的时间间隔, 从脚本视图中选择定时器脚本, 然后从工具条中点击, 就弹出下面的对话框:

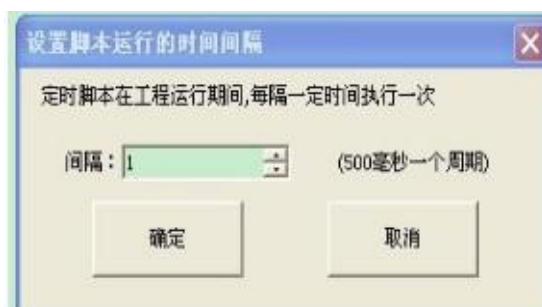


图 23-14 修改定时器间隔

修改完定时间隔后，点击“确定”，就修改成功了。

## 23.11 全局函数

软件允许用户将频繁使用的代码做成一个全局函数的形式，这些全局函数可以在任何脚本中调用，犹如软件自身提供的系统函数一样。

打开全局函数

从**工程配置**中双击**全局函数**，打开全局函数编辑器。如下图所示：



图 23-15 打开全局函数

全局函数编辑器如下图所示：



图 23-16 全局函数视图

在全局函数视图里列举出来了所有已定义的函数。

### 新建全局函数


从脚本编辑工具条中点击 这个图标用于添加一个全局函数，点击鼠标后，出现下面对话框。






图 23-17 新建函数对话框


- 函数名称：**全局函数的名称，不能与已有的函数名称重名。
  - 返回值类型：**函数的返回值类型，有四个选择：无类型、string、integer、float。
  - 参数一：**参数一的名称。
  - 数据类型：**参数的数据类型。有四个选择：无类型、string、integer、float。
- 其余的依次类推。

需要注意的是全局函数**最多只能有四个参数**，如果没有填写参数，则表示函数没有形参，函数有多少个形参取决于实际填写了多少个参数。函数参数是不能重名的。

**删除全局函数**

从函数视图(图 23-16)中，选中要删除的函数，从脚本编辑工具条中选择，就可以删除该脚本。需要注意的是，如果在其他脚本中引用了该函数，则会编译出错。

**编辑全局函数属性**

属性就是要对函数名称或者函数参数进行修改。从函数视图(图 23-16)中，选中要编辑的函数，然后从工具条中选择，就可以打开如图 23-17 的对话框调整函数的参数。

注意，如果函数已经被引用，调整函数后可能带来语法错误。

**全局函数的调用**


与调用系统函数是一样的。

**23.12 语法检查**

在每个脚本保存之前，都应该进行语法检查，只有通过正确语法检查的脚本才可以在 HMI 上正确运行；否则，HMI 是不会执行有错误的脚本的。

当工程被编译时，软件会检查每个脚本的语法是否正确，检查到的语法错误将被列举出来。

当然，用户也可以在编写脚本时，进行语法检查。

具体做法：从脚本编辑工具条中选择，如果语法正确，系统就会提示没有语法错误，而发现的语法都将被枚举出来，以便用户修改。

### 23.13 语法错误

本节列举了软件脚本 t 常见的语法错误。如下：

1、identifier \*\*\* contains invalid characters.

标识\*\*\*含有非法字符

2、attempt to redeclare sub \*\*\*

重复声明的子过程名称

3、attempt to redeclare function \*\*\*

重复声明的函数名称

4、attempt to use reserved word \*\*\* as identifier.

将关键词\*\*\*当成标识

5、attempt to use type \*\*\* as identifier.

将数据类型当成名字标识

6、unexpected ')' while parsing arguments for function \*\*\*

在分析函数\*\*\*时，遇到异常的')'

7、cannot parse expression (one of the arguments of function \*\*\*)

无法分析表达式(该表达式是函数\*\*\*的一个参数)

8、cannot parse arguments of \*\*\*

\*\*\*是无法解析的函数参数

9、too many arguments for function \*\*\*

函数\*\*\*的参数太多

10、not enough arguments for function \*\*\*

函数\*\*\*的参数太少

11、'(' expected after sub name \*\*\*

子过程\*\*\*后需要有' ('

12、unexpected ')' while parsing arguments for sub \*\*\*

分析子过程\*\*\*的参数时遇到意外的')'

13、cannot parse expression (one of the arguments of sub \*\*\*)

无法分析表达式(该表达式是子过程\*\*\*的参数之一)

14、cannot parse arguments of \*\*\*

无法分析子过程\*\*\*的参数

15、too many arguments for sub/function \*\*\*

调用子过程/函数\*\*\*的参数太多

16、not enough arguments for sub/function \*\*\*

调用子过程/函数\*\*\*的参数太少

17、cannot parse expression

无法分析的表达式

18、'(' expected after function name \*\*\*

函数名\*\*\*后应该有' ('

19、unexpected use of sub \*\*\* as a part of expression

子过程\*\*\*被意外地当成表达式的一部分

20、illegal statements preceding subs/functions declaration

在处理子过程/函数的声明中遇到非法的描述

21、unexpected end of file while looking for 'endsub'

在寻找'endsub' 文件意外结束

22、end of line expected after 'else'.

'else' 后应该换行

23、end of line expected after 'endif'.

'endif' 后应该换行

24、end of line expected after 'next'.

'next' 后应该换行

25、end of line expected after 'wend'.

'wend' 后应该换行

26、'while', 'until' or end of line expected after 'do'.

'do' 之后应该是'while', 'until' 或者换行

27、cannot parse expression after 'while'.

无法分析'while' 之后的表达式

28、cannot parse expression after 'until'.

无法分析'until' 之后的表达式

29、'do' without 'loop'.

'do' 之后应该有'loop' 匹配

30、sub \*\*\* contains invalid character '@'.

子过程\*\*\*含有非法字符'@'

31、sub \*\*\* already declared.

子过程\*\*\*已经被声明

32、function \*\*\* already declared.

函数\*\*\*已经被声明

33、sub name expected after 'sub'.

'sub' 关键词后应该是子过程的名称

34、function name expected after 'function'.

'function' 关键词后应该是函数名称

35、variable name expected

应该有变量名称

36、argument \*\*\* contains invalid character '@'.

参数\*\*\*包含有保留字符 '@'

37、'integer', 'floating' or 'string' expected

后面应该有'integer', 'floating' 或者 'string'

38、' ', ' or ')' expected

后面应该是' ', ' , 或者 ')'

39、'endsub' without 'sub'.

'endsub' 没有和相应的关键词 'sub' 匹配.

40、'endfunction' without 'function'.

'endfunction' 没有和相应的关键词 'function' 匹配

41、end of line expected after 'beep'.

'beep' 之后应该换行

42、'dim' unexpected here.

非法的'dim'

43、variable name expected after 'dim'

'dim' 之后应该是变量名

44、'as' expected after variable name.

变量名应该有'as'

45、'integer' 'floating' or 'string' expected after 'as'.

'as' 之后应该是'integer' 'floating' 或 'string'

46、', ' or end of line expected after type in dim statement.

Dim 语句之后应该是', ' 或者换行

47、cannot parse expression after 'while'.

无法分析'while' 之后的表达式

48、end of line expected after 'while' condition.

'while' 条件之后应该换行

49、'while' without 'wend'

'while' 没有相应的'wend' 匹配

50、end of line expected after 'wend'.

'wend' 之后应该换行

51、'wend' without 'while'

'wend' 没有相应的关键词'while' 相匹配

52、variable name expected after 'for'.

'for' 之后应该是变量名

53、'=' expected after variable name

变量名之后应该有 '='

54、cannot parse expression after 'for'

无法分析 'for' 之后的表达式

55、'to' expected

应该有 'to'

56、cannot parse expression after 'to'

无法分析 'to' 之后的表达式

57、cannot parse expression after 'step'

无法分析 'step' 之后的表达式

58、end of line expected

应该换行

59、'for' without 'next' "

'for' 没有相应的 'next' 匹配

60、end of line expected after 'next'.

'next' 之后应该换行

61、'next' without 'for'

'next' 没有相应的 'for' 匹配

62、cannot parse expression after 'if'.

无法分析 'if' 之后的表达式

63、'then' expected.

后面应该有 'then'

64、unexpected end of file while looking for 'endif'.

在寻找 'endif' 时文件意外结束

65、unexpected end of file while looking for 'else' or 'endif'.

在寻找 'else' 或 'endif' 时文件意外结束

66、'else' without 'if'.

'else' 没有相应的 'if' 与之相匹配

67、'endif' without 'if'.

'endif' 没有相应的 'if' 与之相匹配.

68、label name expected after 'goto'.

'goto' 之后应该是标识名

69、unexpected end of line while looking for ')' in function call.

在函数调用时寻找 ')' 遇到文件意外结束。

70、', ' expected.

应该有 ', '

71、missing ')'.

没有找到 ')'.

72、unexpected end of line in expression.

在表达式中遇到意外的换行

73、unexpected end of file in expression.

## 23.14 系统函数

### 23.14.1 A2H 函数

#### 一、函数功能

将以 A2 为起始地址的二进制数的字符串转换为 4 位 2 进制数结果存入 A1 中.

表达式:  $A1 = A2H(A2)$

#### 二、示例

```
@W_HDW0=@W_SendNum1+@W_SendNum0<<8?
```

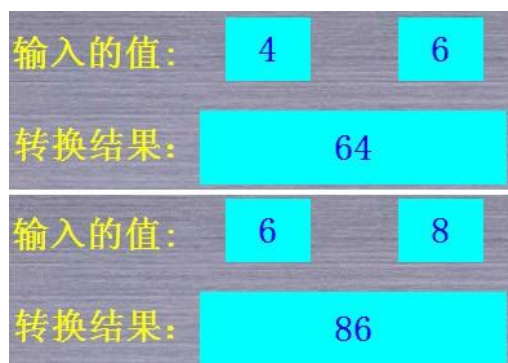
```
@W_HDW7890=A2H(@W_HDW1000)
```

注: 1、SendNum1 一个字表示一个字节, 将 SendNum0 左移 8 位加上 SendNum1 组成一个字。

2、A2H 里的参数必须是地址。

3、协议选择的是用户自定义协议。

#### 三、模拟效果



### 23.14.2 Abs 函数

#### 1. 函数功能

返回一个数的绝对值。

表达式:  $val = Abs(N)$

#### 2. 示例

```
dim a,b,c as integer
a = SignedInt16("@W_HDW0")
b=a
c = Abs(b)
@W_HDW1=c
```

注: 函数 SignedInt16 功能是将无符号的单字转成有符号

#### 3.模拟效果

写入值	转换后
-6	6
写入值	转换后
6	6

23. 14. 3 ACos 函数

一、函数功能

返回反余弦的值。

表达式: var = ACos(expr)

二、示例

```
Dim a ,b? as floating
a = D2Float("@W_HDW0",a)
b=ACos(a)
b= RadToDeg(b)
float2d("@W_HDW2", b)
```

注：RadToDeg 函数左右将弧度值转换成角度值

三、模拟效果

写入值	转换后
0.0	90.000
写入值	转换后
0.5	60.000

23. 14. 4 AddrStringCompare 函数

一、简介

比较 A1 和 A2 的 length 长度的值，如果相等则 1，否则 0。

二、语法

```
n =AddrStringCompare(A1, A2, length)
```

### 三、说明

A1A2: 必须为以 '@' 字符开始的系统内存或 PLC 地址（如@W\_00002）。

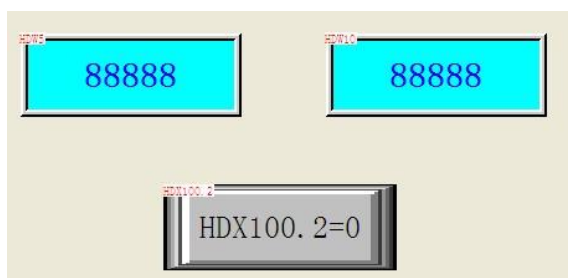
Length: 字符串的长度

### 四、示例

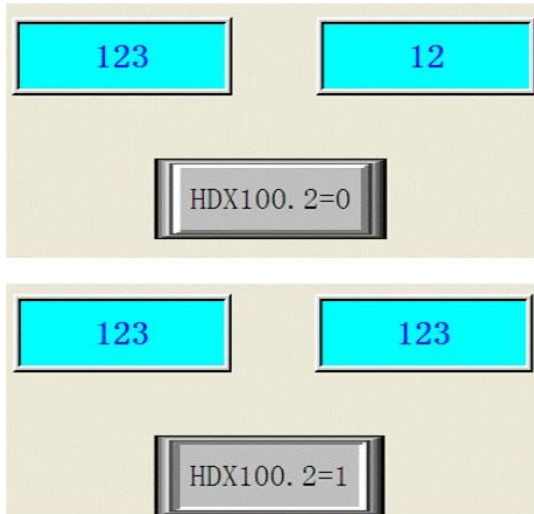
```
if AddrStringCompare("@W_HDW10","@W_HDW0",5)=1 then
  @B_HDX100.2=1
else
  @B_HDX100.2=0
endif
```

### 五、演示效果

界面设计如图



演示效果如图



#### 23.14.5 Asc 函数

##### 一、函数功能

返回字符串的第一个字符的 ASCII 值。

表达式: val = Asc(expr)

##### 1. 示例



a = Asc("A")' 返回 97

a = Asc("a")' 返回 65

a = Asc("Apple")' 返回 49

a = Asc(123)' 返回 49

注：expr 任何表达式，如果不是字符串，就把它转换成字符串。

模拟效果



### 23.14.6 AsFloating 函数

一、函数功能 将参数转换成浮点。

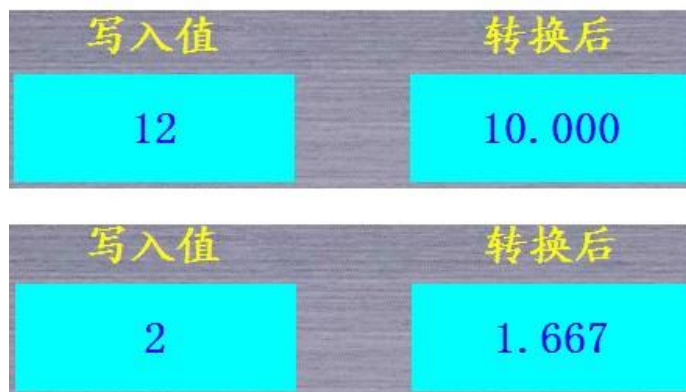
表达式：val = AsFloating(expr)

1. 示例

```
dim a as integer
a=@W_HDW0
b= AsFloating(a)
b/1.2
Float2D("@W_HDW2",b)
```

注：expr 使用定义好的变量。函数 Float2D 把 b 的浮点型数据传给浮点型地址 hdw2

2. 模拟效果



### 23.14.7 ASin 函数

一、函数功能

返回反正弦的值。

表达式：var = ASin(expr)

二、示例

```
Dim a,b as floating
a= D2Float("@W_HDW0",a)
```

```
b=ASin(a)
b= RadToDeg(b)
float2d("@W_HDW2", b)
```

注：RadToDeg 函数左右将弧度值转换成角度值

三、模拟效果

写入值	转换后
0.5	30.000
写入值	转换后
1.0	90.000

23. 14. 8 AsInteger 函数

一、 函数功能

将参数转换成整型。

表达式：val = AsInteger(expr)

1. 示例

```
dim    a        as    floating
a = D2Float("@W_HDW0",a)
b= AsInteger(a)
@W_HDW112=b
```

注：函数 Float2D 把 b 的浮点型数据传给浮点型地址 hdw2

2.模拟效果

写入值	转换后
2.300	2
写入值	转换后
63.300	63

23. 14. 9 AsString 函数

一、 函数功能

将参数转换成字符串。

表达式：val = AsString(expr)

## 二、示例

a=123

b=234

c= AsString(a)+ AsString(b)

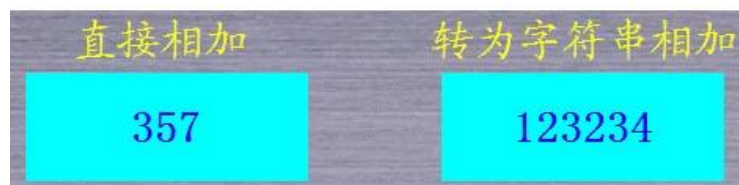
@W\_HDW0=c

d=a+b

@W\_HDW100=d

输出： HDW100=357, HDW0=123234

## 三、模拟效果



### 23. 14. 10 ATan 函数

返回一个数的反正切值。

var = ATan(expr)

var 为计算完后值存储的地址或变量

expr 可为 HIM 地址 (PLC 地址) 或变量

返回弧度值。结果的范围为 $-\pi/2$  到  $\pi/2$  弧度。要从弧度值转换成角度值请调用 RedToDeg 函数。使用 Atan2 函数来获取 $-\pi$  到  $\pi$  弧度范围的结果。

## ATan函数

返回一个数的反正切值

语法  
`var = ATan(expr)`

说明  
 返回的值为弧度值。结果的范围为 $-\pi/2$ 到 $\pi/2$ 弧度

例：  
`@W_HDW20=ATan(@W_HDW10)`

输入数值

弧度

## ATan函数

返回一个数的反正切值

语法  
`var = ATan(expr)`

说明  
 返回的值为弧度值。结果的范围为 $-\pi/2$ 到 $\pi/2$ 弧度

例：  
`@W_HDW20=ATan(@W_HDW10)`

输入数值

弧度

### 23.14.11 ATan2 函数

返回 x/y 的反正切值。

`var = ATan2(x, y)`

var 为计算完后值存储的地址或变量

x 可为地址或变量

y 可为地址或变量

ATan2函数

返回x/y的反正切值。

语法

var = ATan2(x, y)

说明

返回弧度值。结果的范围是-pi到pi之间。  
ATan2使用两个参数的符号来确定返回值的象限。要从弧度值转换成角度值请调用RedToDeg函数。

例:

@W\_HDW20=ATan2(@W\_HDW10,@W\_HDW12)

88888

88888

0.8888

ATan2函数

返回x/y的反正切值。

语法

var = ATan2(x, y)

说明

返回弧度值。结果的范围是-pi到pi之间。  
ATan2使用两个参数的符号来确定返回值的象限。要从弧度值转换成角度值请调用RedToDeg函数。

例:

@W\_HDW20=ATan2(@W\_HDW10,@W\_HDW12)

6

32

0.1853

返回弧度值。结果的范围是-pi 到 pi 之间。ATan2 使用两个参数的符号来确定返回值的象限。要从弧度值转换成角度值请调用 RedToDeg 函数。

23. 14. 12 B2W 子程序

将一串从 A2 开始的字节数组（其大小由 A3 指定）转换为一串字符串。结果保存在 A1 开始的系统内存或 PLC 地址里。字符串的高位为 0。

语法

B2W(A1, A2, A3)

A1 为转换后存储的地址

A2 为被转换数的地址

A3 为转换的长度

说明

这是一个子程序，所以没有返回值。A1、A2 必须为以 '@' 字符开始的系统内存或 PLC 地址（如 @W\_00002）。A3 可以为系统地址或其它类型的变量。

示例

B2W子程序

将一串从A2开始的字节数组（其大小由A3指定）转换为一串字数组。结果保存在A1开始的系统内存或PLC地址里。字数组的高位为0

**语法**  
B2W(A1, A2, A3)

**说明**  
 这是一个子程序，所以没有返回值。  
 A1、A2必须为以 '@' 字符开始的系统内存或PLC地址（如@W\_00002）。  
 A3可以为系统地址或其它类型的变量。

**例：**  
B2W(@W\_HDW20, @W\_HDW10, 2)

A2

FFFF

A1

FFFF

FFFF



### B2W子程序

将一串从A2开始的字节数组（其大小由A3指定）转换为一串字数组。结果保存在A1开始的系统内存或PLC地址里。字数组的高位为0

**语法**  
B2W(A1, A2, A3)

**说明**  
这是一个子程序，所以没有返回值。  
A1、A2必须为以'@'字符开始的系统内存或PLC地址（如@W\_00002）。  
A3可以为系统地址或其它类型的变量。

**例：**  
B2W(@W\_HDW20, @W\_HDW10, 2)

A2	A2E6
	E6
A1	A2

### B2W子程序

将一串从A2开始的字节数组（其大小由A3指定）转换为一串字数组。结果保存在A1开始的系统内存或PLC地址里。字数组的高位为0

**语法**  
B2W(A1, A2, A3)

**说明**  
这是一个子程序，所以没有返回值。  
A1、A2必须为以'@'字符开始的系统内存或PLC地址（如@W\_00002）。  
A3可以为系统地址或其它类型的变量。

**例：**  
B2W(@W\_HDW20, @W\_HDW10, 2)

A2	CD3F
	3F
A1	CD

#### 23. 14. 13 BCD 函数

将 A2 从一个二进制数转换为 BCD 数，并将结果保存在 A1 里。

##### 语法

将一个二进制数转换为 BCD 数

A1 = BCD(A2)

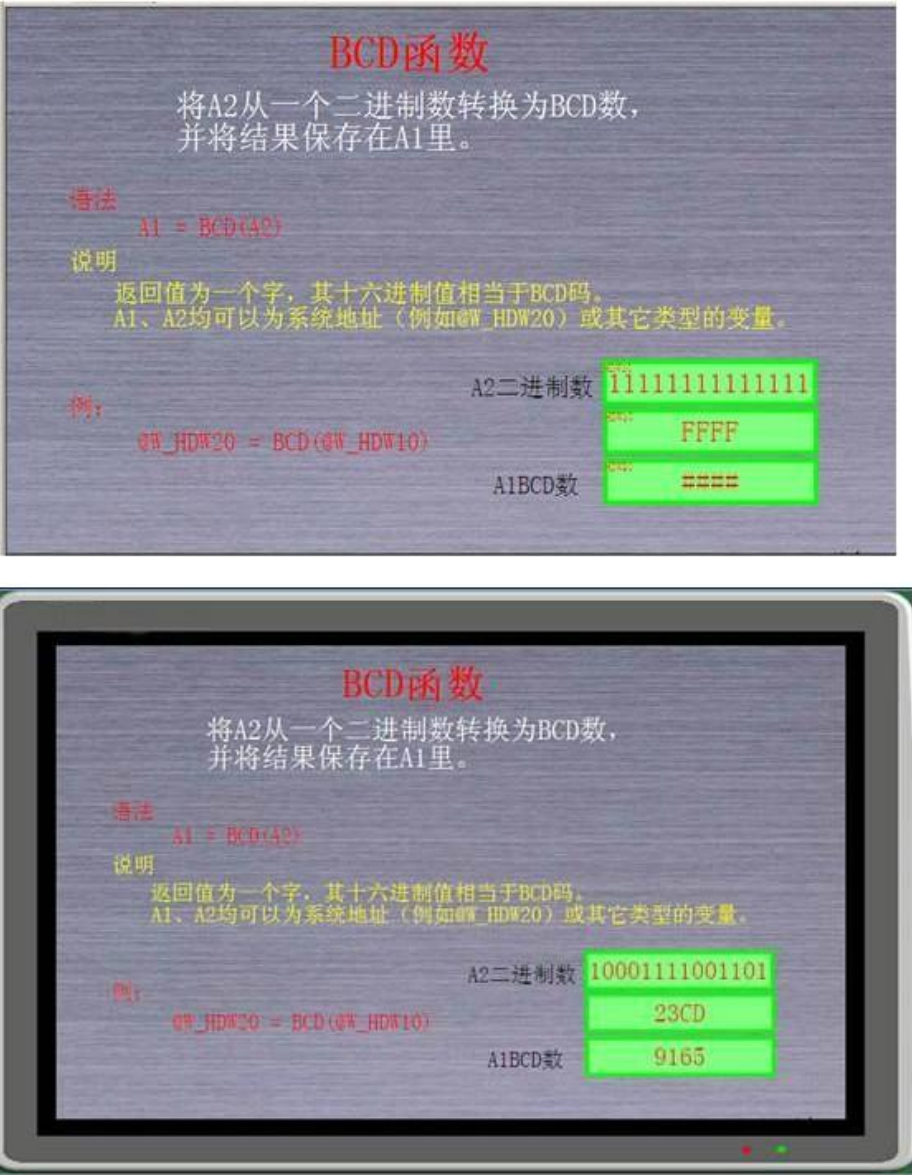
A1 为转换后的值存放的地址或变量

A2 为被转换的值

##### 说明

返回值为一个字，其十六进制值相当于 BCD 码。A1、A2 均可以为系统地址（例如@W\_0002）或其它类型的变量。

##### 示例



23. 14. 14 Beep 功能

通过蜂鸣器发出一个蜂鸣声。

语法

Beep

说明

没有参数

示例



## Beep功能

通过蜂鸣器发出一个蜂鸣声。

语法  
Beep

说明  
没有参数

例:

蜂鸣

```
if QB_HDX0.1=1 then Beep
```

## Beep功能

通过蜂鸣器发出一个蜂鸣声。

语法  
Beep

说明  
没有参数

例:

蜂鸣

```
if QB_HDX0.1=1 then Beep
```

## Beep功能

通过蜂鸣器发出一个蜂鸣声。

语法  
Beep

说明  
没有参数

例:

解除

```
if QB_HDX0.1=1 then Beep
```

### 23.14.15 BIN 函数

将 A2 从一个 BCD 数转换成二进制数，并将结果保存在 A1 里。

语法

将一个 BCD 数转换成二进制数。

A1 = BIN(A2)

A1 为转换后的值存放的地址或变量

A2 为被转换的值，可以是地址或变量

说明

BIN函数

将A2从一个BCD数转换成二进制数，并将结果保存在A1里。

语法

A1 = BIN(A2)

说明

返回值为一个字。A1、A2均可以为系统地址（例如@W\_HDW20）或其它类型的变量。

例：

A2BCD数

=====

A1二进制数

1111111111111111

FFFF

QW\_HDW20 = BCD(QW\_HDW10)

BIN函数

将A2从一个BCD数转换成二进制数，并将结果保存在A1里。

语法

A1 = BIN(A2)

说明

返回值为一个字。A1、A2均可以为系统地址（例如@W\_HDW20）或其它类型的变量。

例：

A2BCD数

3692

A1二进制数

111001101100

E6C

QW\_HDW20 = BCD(QW\_HDW10)

23. 14. 16 BMOV 子程序

将从 A2 开始的数据块拷贝到 A1 开始的内存块里。A3 表示要拷贝的字的个数。

语法

数据块拷贝。

BMOV(A1, A2, A3)

A1 为拷贝数据的目标地址

A2 为拷贝数据的源地址  
A3 为拷贝数据的长度  
说明

BMOV子程序

将从A2开始的数据块拷贝到A1开始的内存块里。A3表示要拷贝的字的个数。

语法

`BMOV(A1, A2, A3)`

说明

这是一个子程序，所以没有返回值。

A1、A2必须为以'@'字符开始的系统内存或PLC地址（如@W\_00002）。

A3可以为系统地址或其它类型的变量。

A3个数

例:

A2

88888

88888

88888

88888

88888

BMOV(@W\_HDW20, @W\_HDW10, 5)

A1

88888

88888

88888

88888

88888

BMOV子程序

将从A2开始的数据块拷贝到A1开始的内存块里。A3表示要拷贝的字的个数。

语法

`BMOV(A1, A2, A3)`

说明

这是一个子程序，所以没有返回值。

A1、A2必须为以'@'字符开始的系统内存或PLC地址（如@W\_00002）。

A3可以为系统地址或其它类型的变量。

A3个数

例:

A2

25

2589

36

58

375

BMOV(@W\_HDW20, @W\_HDW10, 5)

A1

25

2589

36

58

375

23. 14. 17 Chr 函数

将其参数从整型转换成相对应的 ASCII 字符，并返回这些字符组成的字符串。

val = Chr(expr1, expr2, ...)

val 为转换完的值存放的地址或变量

expr 为被转换的值存放的地址或变量

示例工程



## Chr函数

将其参数从整型转换成相对应的ASCII字符，并返回这些字符组成的字符串。

**语法** val = Chr(expr1, expr2, ...)  
exprX任何表达式，如果不是整型，则转换成整型。

**说明**  
该函数可以接收任意个数的参数；  
下面的例子解释了Chr函数的用法：  
a = Chr(72) ' 返回"H"。  
a = Chr(72, 69, 76, 76, 79) ' 返回"HELLO"。  
a = Chr() ' 返回空字符串。

**例：**

@W\_HDW30 = Chr(@W\_HDW20, @W\_HDW21, @W\_HDW22, @W\_HDW23, @W\_HDW24)  
往HDW20 21 22 23 24 中依次输入72, 69, 76, 76, 79 返回字符串HELLO

88888 88888 88888 88888 88888 01234

## Chr函数

将其参数从整型转换成相对应的ASCII字符，并返回这些字符组成的字符串。

**语法** val = Chr(expr1, expr2, ...)  
exprX任何表达式，如果不是整型，则转换成整型。

**说明**  
该函数可以接收任意个数的参数；  
下面的例子解释了Chr函数的用法：  
a = Chr(72) ' 返回"H"。  
a = Chr(72, 69, 76, 76, 79) ' 返回"HELLO"。  
a = Chr() ' 返回空字符串。

**例：**

@W\_HDW30 = Chr(@W\_HDW20, @W\_HDW21, @W\_HDW22, @W\_HDW23, @W\_HDW24)  
往HDW20 21 22 23 24 中依次输入72, 69, 76, 76, 79 返回字符串HELLO

72 69 76 76 79 HELLO

## Chr函数

将其参数从整型转换成相对应的ASCII字符，并返回这些字符组成的字符串。

**语法** val = Chr(expr1, expr2, ...)  
exprX任何表达式，如果不是整型，则转换成整型。

**说明**  
该函数可以接收任意个数的参数；  
下面的例子解释了Chr函数的用法：  
a = Chr(72) ' 返回"H"。  
a = Chr(72, 69, 76, 76, 79) ' 返回"HELLO"。  
a = Chr() ' 返回空字符串。

**例：**

@W\_HDW30 = Chr(@W\_HDW20, @W\_HDW21, @W\_HDW22, @W\_HDW23, @W\_HDW24)  
往HDW20 21 22 23 24 中依次输入85, 36, 47, 20, 65 返回字符串US/PA

85 36 47 20 65 US/PA

### 23. 14. 18 C1rB 子程序

将位 A1 设为 FALSE(0)。

ClrB(A1)

A1 可以为系统地址(位)或其它类型的变量。





### 23.14.19 常量

Script 提供了几个固定的常量，你可以在脚本中使用：

pi = 3.14159265358979321

TRUE = 1

FALSE = 0

### 23.14.20 Cos 函数

返回一个角度的余弦值。

var = Cos(*expr*)

**语法**

*expr*

任意表达式，其值为一个角度的弧度值。

输入输出值都为浮点

---

**示例**

```
dim a as floating
dim b as floating
b=asfloating(@W_HDW11)
a = Cos(b)
@W_HDW13=Float2D("@W_HDW13", a)
```

**Remarks**

要从弧度值转换成角度值请调用 **RedToDeg** 函数。

**23. 14. 21 D2F 子程序**

将一个 32 位的整型数转换成 32 位浮点数。被转换的整数在 A2 及下一个字里。转换结果保存在 A1 及下一个字里。

**语法**

将一个 32 位的整型数转换成 32 位浮点数。

D2F(A1, A2)

A1 转换之后的值存放的地址或变量

A2 被转换数的值存放的地址

说明



D2F子程序

将一个32位的整型数转换成32位浮点数。  
被转换的整数在A2及下一个字里。  
转换结果保存在A1及下一个字里。

语法

D2F(A1, A2)

说明

这是一个子程序，所以没有返回值。  
A2必须为以'@'字符开始的系统内存或PLC地址（如@W\_00002）。  
A1可以为系统地址(浮点)或其它类型的变量。

例:

D2F(@W\_HDW13, @W\_HDW11)  
@W\_HDW13为浮点型, @W\_HDW11为32位整型数。

88888

=====

D2F子程序

将一个32位的整型数转换成32位浮点数。  
被转换的整数在A2及下一个字里。  
转换结果保存在A1及下一个字里。

语法

D2F(A1, A2)

说明

这是一个子程序，所以没有返回值。  
A2必须为以'@'字符开始的系统内存或PLC地址（如@W\_00002）。  
A1可以为系统地址(浮点)或其它类型的变量。

例:

D2F(@W\_HDW13, @W\_HDW11)  
@W\_HDW13为浮点型, @W\_HDW11为32位整型数。

2360

2360.0

23. 14. 22 D2Float 函数

把浮点型数值 A1 存储 f 的浮点型数据里

语法

把浮点型数值 A1 存储到 f 的浮点型数据里。

f = D2Float(A1, f)

A1 可以为系统地址(位)或其它类型的变量。

f 为用户自己定义的浮点型的变量

说明



D2Float函数

把浮点型数值A1存储到f的浮点型数据里

语法

$$f = D2Float(A1, f)$$

说明

A1:A1必须为以'@'字符开始的系统内存或PLC地址（如@W\_HDW2）。  
f是浮点型的变量

例:

```
dim f as floating
dim a as floating
a=1.2
float2d("@w_hdw100",a)
f = D2Float("@w_hdw100", f)
float2d("@w_hdw102", f)
```

=====

=====

D2Float函数

把浮点型数值A1存储到f的浮点型数据里

语法

$$f = D2Float(A1, f)$$

说明

A1:A1必须为以'@'字符开始的系统内存或PLC地址（如@W\_HDW2）。  
f是浮点型的变量

例:

```
dim f as floating
dim a as floating
a=1.2
float2d("@w_hdw100",a)
f = D2Float("@w_hdw100", f)
float2d("@w_hdw102", f)
```

1.2

1.2

23. 14. 23    DegToRad 函数

从角度转换成弧度值。

var = DegToRad(*expr*)

*expr* 为输入地址

*var* 为输出地址

输入输出值都为浮点

示例工程



23.14.24 Dim

DIM ... AS ...

声明一个变量，固定其类型。

Dim *varname1* [AS *type*], *varname2* [AS *type*] ...

语法

*varname*

变量名称。参看 [variable](#) 命名规则。

*type*

变量类型。参看 [variable](#) 的类型。

#### Remarks

用 Dim 声明的变量在运行期间并不改变其类型，考虑如下的例子：

```

' a was not declared with dim
a = 10      ' a is integer and contains 10
a = 3.15    ' a becomes floating and contains 3.15
a          = ' a becomes string and contains
"Hello"    "Hello"
```

```

Dim a as Integer      ' a was declared and can be only integer
```

```

a = 10      ' a is integer and contains 10
a = 3.15    ' a is integer and contains 3
a = "Hello" ' a is integer and contains 0
```

如果没有定义变量的类型，Dim 会被略过。

一次只能声明一个变量。

下面的例子解释了 Dim 的用法：

```

Dim a                                ' No type specified. Ignored.
Dim a as Integer                     ' OK
Dim a, b, c as Integer               ' Three variables in single Dim statement.
Dim str1, str2 as String, int1,      ' Multiple variables of different types in
int2 as Integer                      single Dim statement.
```

### 23.14.25 Do

DO ... LOOP

当 condition 为 True 或直到 condition 为 True 为止，重复执行一个块的指令。

```

Do [While | Until condition]
    [statements]
```

Loop

或者，你可以使用如下的语法：

```

Do
    [statements]
Loop [While | Until condition]
```

参数

*condition*

求得 True 或 False 的表达式。

*statements*

当 condition 为 True 或直到 condition 为 True 为止，重复执行的一条或多条指令。

说明

如果 condition 为 True，statements 里的所有指令都会被执行，直到碰到 Wend 指令为止，然后控制权回到 While，并且 condition 再一次被检查。如果 condition 仍然为 True，上面的过程就被重复。如果它不是 True，指令会接着 Wend 之后继续执行。

与 For 不同的是，While 在每一个循环通过时检查 *condition*。

下面的例子解释了 While...Wend 的用法：

```
dim i as integer
do while i<100
  i=i+1
  @W_HDW0=i
loop
```

当 i=100 时 DO 循环结束

#### 23. 14. 26 End

终止脚本的执行。例如：

If a = 10 Then End

#### 23. 14. 27 W2B 子程序

将 A2 去高字补上下一个地址的地址放到 A1。

W2B(A1, A2, A3)

A1 为转换后存储的地址

A2 为被转换数的地址

A3 为转换的长度

示例工程

W2B子程序

将A2去高字补上下一个地址的地址放到A1

语法

W2B(A1, A2, A3)

说明

这是一个子程序，所以没有返回值。  
A1、A2必须为以'@'字符开始的系统内存或PLC地址（如@W\_00002）。  
A3可以为系统地址或其它类型的变量。

例：

W2B(@W\_HDW20, @W\_HDW10, 2)

A1

A2

FFFF

FFFF

FFFF

W2B子程序

将A2去高字补上下一个地址的地址放到A1

语法

W2B(A1, A2, A3)

说明

这是一个子程序，所以没有返回值。  
A1、A2必须为以'@'字符开始的系统内存或PLC地址（如@W\_00002）。  
A3可以为系统地址或其它类型的变量。

例：

W2B(@W\_HDW20, @W\_HDW10, 2)

A1

A2

12AB

CD56

56AB

23. 14. 28 F2D 子程序

将一个 32 位浮点数转换成为 32 位的整型变量。浮点数在 A2 及其下一个字里，转换结果保存在 A1 及其下一个字里。

语法

F2D(A1, A2)

说明

这是一个子程序，所以没有返回值。A1 必须为以'@'字符开始的系统内存或 PLC 地址(如@W\_00002)。A2 可以为系统地址(浮点)或其它类型的变量。

例子

F2D(@W\_00001, @W\_0002)

23. 14. 29 F2S 函数

把 A1 的浮点型转成字符型，按 s1 格式存储存 A2

一：语法

F2S(A1, A2, s1)



二：说明

A1、A2 必须为以 '@' 字符开始的系统内存或 PLC 地址（如@W\_HDW200）。

S1: 存储的格式。比如：03.03f--（3 位小数）

三：示例

F2S，由浮点转换成字符串，可以指定格式，比如

@W\_HDW200 是浮点型数值输入；

@W\_HDW100 文本数值输入；

F2S("@W\_HDW200", "@W\_HDW100", "03.03f")

四：工程效果图



以值 A2 填充从 A1 开始的系统地址，A3 表示填充的字数。

### 一：语法

FILL(A1, A2, A3)

### 二：说明

这是一个子程序，所以没有返回值。

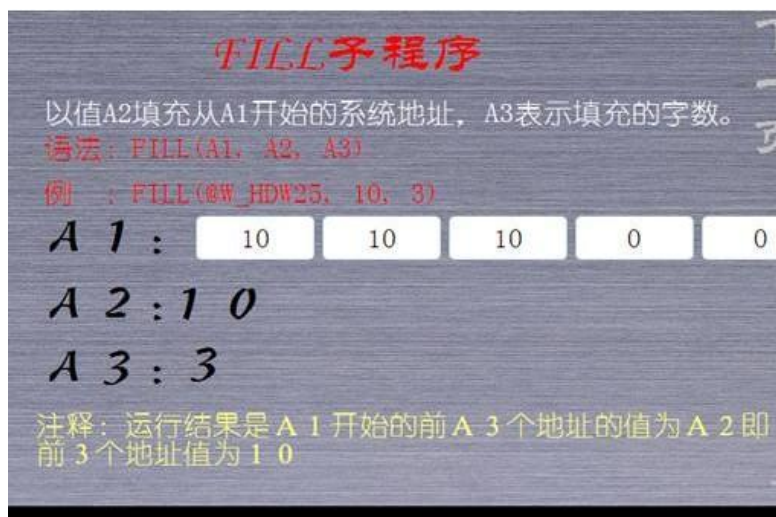
A1 必须为以 '@' 字符开始的系统内存或 PLC 地址（如 @W\_00002）。A2、A3 可以为系统地址（浮点）或其它类型的变量。

### 三：示例

FILL(@W\_HDW25, 10, 3)

注释：运行结果是 @W\_HDW25 开始的三个地址的值为 10，即 @W\_HDW25=10, @W\_HDW26=10, @W\_HDW27=10.

### 四：工程效果图



## 23.14.31 Float2D 函数

把 A2 的浮点型传给 A1 浮点型

### 一：语法

Float2D(A1, A2);

### 二：说明

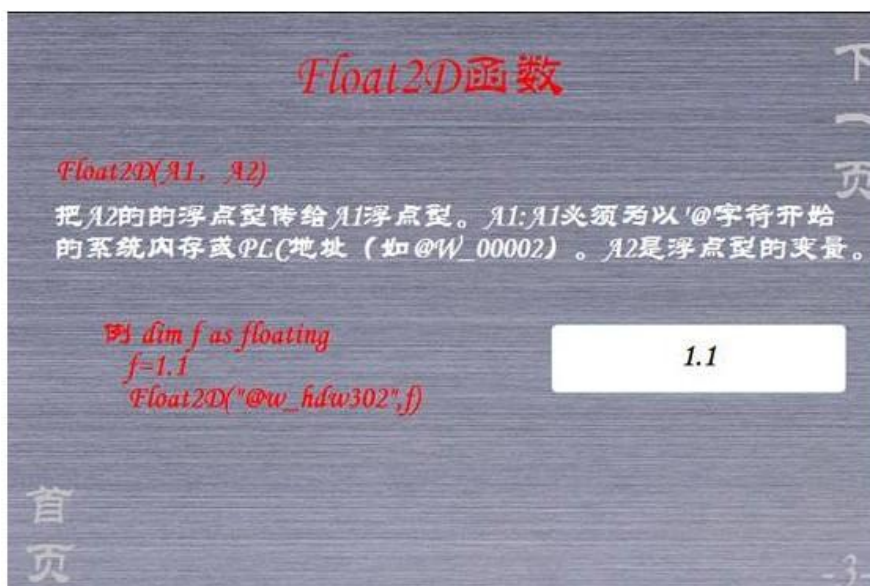
A1:A1 必须为以 '@' 字符开始的系统内存或 PLC 地址（如 @W\_00002）。A2 是浮点型的变量。

### 三：示例

dim f as floating

f=1.1

Float2D("@w\_hdw102",f)-- @w\_hdw102=1.1



### 23.14.32 For

23.14.32 FOR ... TO ... STEP ... NEXT

指定次数的重复执行一块指令。

For *counter* = *start* To *end* [Step *step*]

[*statements*]

Next

#### 一：语法

*counter*

用来作为循环计数器的一个变量。

*start*

*counter* 的初始值。

*end*

*counter* 的最后一个值。

*step*

每一次循环，*counter* 改变的值。如果没有指定，*step* 默认为 1。For 依赖于 *start* 和 *end* 的值，自动定义要加或减 *counter*。

*statements*

在 **For** 和 **Next** 之间的，执行指定次数的指令集合。

#### 二：说明

当循环开始，并执行了循环里的所有指令，*step* 被加到 *counter* 上。这就是说，无论循环中的指令是会被再执行（取决于循环的条件是否满足），还是循环退出并且继续执行 Next 之后的指令，加 *counter* 的动作都会执行到。

*start*、*end* 以及 *step* 可以为任何表达式或任何类型的变量：

通过放置一个 For...Next 循环在另一个循环中，可以嵌套调用 For...Next 循环。



和 While 不同, For 只对 *end*取值一次。

空的 For 循环会被忽略, 并不能起到延时的作用。

三：示例工程

说明：为体现效果下图示例工程中对例 1 做了地址偏移

脚本如下

```
dim tem as string
dim a as integer
for i = 333 To 325 Step -2
tem=newnoaddr("@W_HDW104", 333-i)
WriteAddr(tem, i)
```

@W\_HDW11=i

Next



23. 14. 33 Function

声明一个函数的名称, 参数, 以及函数体的代码。

Function *name* (*arglist*)

*statements*

*name* = *expression*

*statements*

End Function

语法

*name*

函数名称. 参看标准 [variable](#) 命名规则.

*arglist*

代表参数的变量列表, 当函数被调用时, 这些参数会被传递进入. 多个参数用逗号分隔。

*statements*

函数体中运行的代码集合。

## 说明

可以对 *varlist* 里的变量定义其类型或不定义。

不能在任意其它程序体中定义一个 **Function** 程序。

调用 **Function** 时，先写函数名，后跟参数列表。

**Function** 声明必须先于 **Function** 调用。

要从一个函数返回值，对函数名的值进行赋值。在函数体内部，这样的赋值可以在任何地方进行。如果没有对函数名赋值，函数体返回整型值 0。

函数可以递归调用，即，可以在一个任务里调用自身。但是，递归调用有可能会产生堆栈溢出。

下面的例子解释了 **Function** 的用法：

```
Function sincos (angle as floating)
    sincos = sin(angle) + cos(angle)
End Function<
.....
@W_0002 = sincos(pi/2)
.....
```

### 23.14.34 Goto

在一个函数体内部无条件的跳到指定的一行指令。

#### 语法

**Goto label**

行标号可以为以字母开始，冒号（:）结束的任意的字符串。行标号对大小写不敏感。行标号必须在行的开始。

#### 例



## 说明

**Goto** 只能跳到函数体内部，它可见的标号行去。

### 23.14.35 H2A 函数

将一个十六位二进制数转换成用 ASCII 字符表示的 4 位 16 进制数。要转换的数在 A2 里。

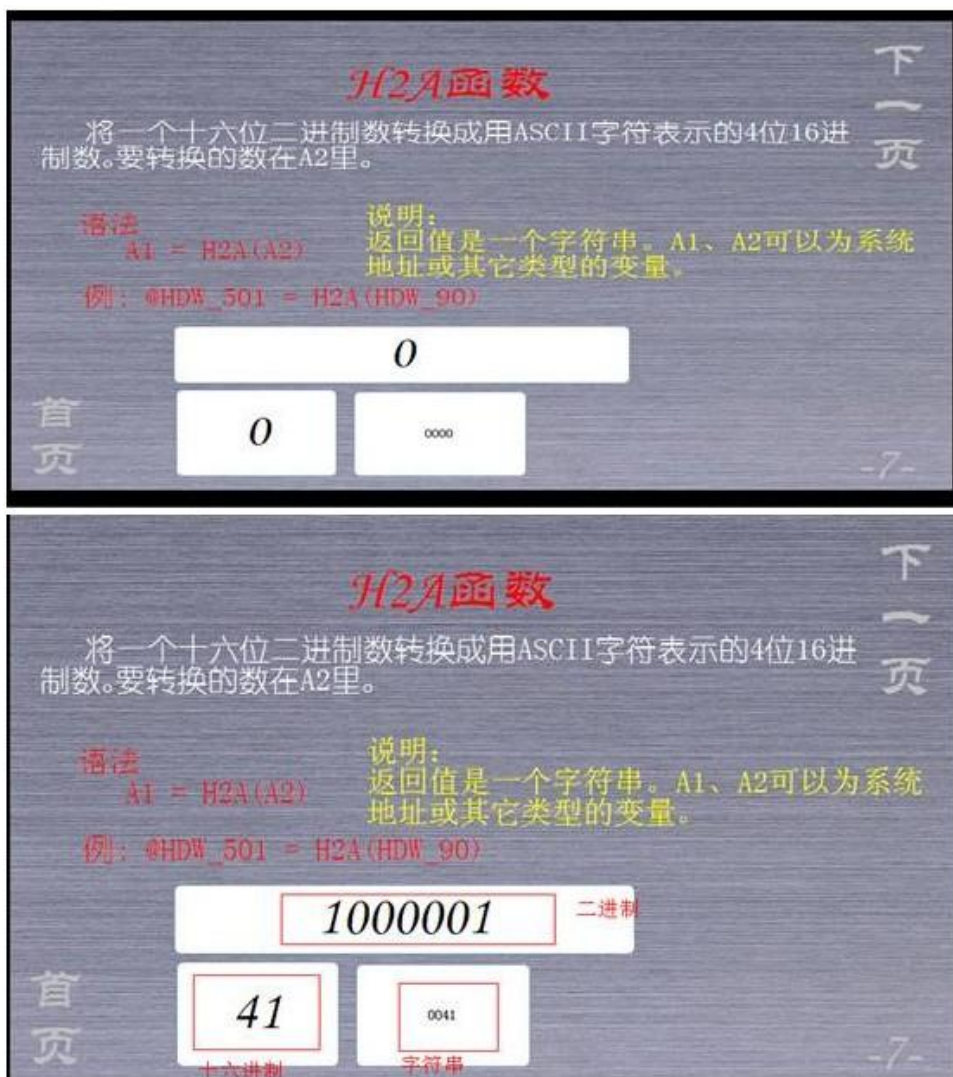
#### 一：语法

$A1 = H2A(A2)$

## 二：说明

返回值是一个字符串。A1、A2 可以为系统地址或其它类型的变量。

## 三：示例工程



### 23. 14. 36 Hypot 函数

计算出直角三角形斜边的值。

$var = \text{Hypot}(expr1, expr2)$

## 一：说明

Hypot 计算出一个直角三角形斜边的值，其他两边的值为  $expr1$  及  $expr2$ 。  
 $expr1$ ,  $expr2$  可为地址变量或常量  
 $var$  为输出地址。

## 示例

$A = \text{Hypot}(@W\_HDW105, @W\_HDW108)$

## 二：示例工程

Hypot函数

计算出直角三角形斜边的值。

语法  
var = Hypot(expr1, expr2)

说明:  
Hypot计算出一个直角三角形斜边的值,  
其他两边的值为expr1及 expr2.

例: A=Hypot(@W\_HDW105,@W\_HDW108)  
@W\_HDW111=A  
如@W\_HDW105输入3, @W\_HDW108输入4,  
@W\_HDW111显示结果为5

下一页

0.00

0.00

0.00

首页

-8-

如输入 3,4 自动得出结果 5，输入 6,8 结果为 10

Hypot函数

计算出直角三角形斜边的值。

语法  
var = Hypot(expr1, expr2)

说明:  
Hypot计算出一个直角三角形斜边的值,  
其他两边的值为expr1及 expr2.

例: A=Hypot(@W\_HDW105,@W\_HDW108)  
@W\_HDW111=A  
如@W\_HDW105输入3, @W\_HDW108输入4,  
@W\_HDW111显示结果为5

下一页

3.00

4.00

5.00

首页

-8-

Hypot函数

计算出直角三角形斜边的值。

语法  
var = Hypot(expr1, expr2)

说明:  
Hypot计算出一个直角三角形斜边的值,  
其他两边的值为expr1及 expr2.

例: A=Hypot(@W\_HDW105,@W\_HDW108)  
@W\_HDW111=A  
如@W\_HDW105输入3, @W\_HDW108输入4,  
@W\_HDW111显示结果为5

下一页

6.00

8.00

10.00

首页

-8-

23. 14. 37 IF

IF ... THEN ... ELSE ... END IF  
依赖于表达式的值，有条件的运行一块指令。  
If condition Then statement [Else elsestatement]

---

或者，你可以用如下的语法：

```
If condition Then
    statements
[Else
    elstatements]
End If
```

语法

*condition*

任意表达式，其值可以为 **True** 或 **False**

*statement, statements*

当 *condition* 为 **True** 时运行的指令块。

*elstatement, elstatements*

当 *condition* 为 **False** 时运行的指令块。

说明

当执行 If 时, *condition* 会被测试。如果 *condition* 为 **True**, Then 之后的指令块会被执行。如果 *condition* 为 **False**, Else 之后的指令块会被执行。当执行完 Then 或 Else 之后的指令块后，会接着执行 End If 之后的指令块。



**IF ... THEN ... ELSE ... END IF**

依赖于表达式的值，有条件的运行一块指令。

**语法**      **例:** If @B\_HDX0.3 = 1 Then

If condition Then      If @B\_HDX0.4 = 1 Then

    statements      @B\_HDX0.5 = 1

[Else      Else

    elsestatements]      @B\_HDX0.5 = 0

EndIf      EndIf

**说明:**      EndIf

当执行If时，condition会被测试。如果condition为True，Then之后的指令块会被执行。如果condition为False，Else之后的指令块会被执行。当执行完Then或Else之后的指令块后，会接着执行EndIf之后的指令块。

下一页

-9-

**IF ... THEN ... ELSE ... END IF**

依赖于表达式的值，有条件的运行一块指令。

**语法**      **例:** If @B\_HDX0.3 = 1 Then

If condition Then      If @B\_HDX0.4 = 1 Then

    statements      @B\_HDX0.5 = 1

[Else      Else

    elsestatements]      @B\_HDX0.5 = 0

EndIf      EndIf

**说明:**      EndIf

当执行If时，condition会被测试。如果condition为True，Then之后的指令块会被执行。如果condition为False，Else之后的指令块会被执行。当执行完Then或Else之后的指令块后，会接着执行EndIf之后的指令块。

下一页

-9-

**IF ... THEN ... ELSE ... END IF**

依赖于表达式的值，有条件的运行一块指令。

**语法**      **例:** If @B\_HDX0.3 = 1 Then

If condition Then      If @B\_HDX0.4 = 1 Then

    statements      @B\_HDX0.5 = 1

[Else      Else

    elsestatements]      @B\_HDX0.5 = 0

EndIf      EndIf

**说明:**      EndIf

当执行If时，condition会被测试。如果condition为True，Then之后的指令块会被执行。如果condition为False，Else之后的指令块会被执行。当执行完Then或Else之后的指令块后，会接着执行EndIf之后的指令块。

下一页

-9-

如工程画面中脚本说写：当 1，2 灯同时亮时，3 灯亮

## 23.14.38 Script 1.0 参考

本参考描述了 Script 以及其内置函数、子程序的大部分特征。

---

## 特性

- [Variables](#)
- [Operators](#)
- [Program structure](#)

## 语法关键字

- [DIM ... AS ...](#)
- [IF ... THEN ... ELSE ... END IF \(single-line & multy-line\)](#)
- [FOR ... TO ... STEP ... NEXT](#)
- [DO ... LOOP](#)
- [WHILE ... WEND](#)
- [GOTO](#)
- [SUB ... END SUB](#)
- [FUNCTION ... END FUNCTION](#)
- [END](#)
- [BEEP](#)

## 函数

- [sqr](#)
- [hypot](#)
- [abs](#)
- [sin](#)
- [cos](#)
- [tan](#)
- [atan](#)
- [asin](#)
- [acos](#)
- [atan2](#)
- [degtorad](#)
- [radtodeg](#)
- [exp](#)
- [log](#)
- [log10](#)
- [rand](#)
- [power](#)
- [Len](#)
- [InStr](#)
- [Left](#)
- [Right](#)

- [Mid](#)
- [LTrim](#)
- [RTrim](#)
- [Trim](#)
- [UCase](#)
- [LCase](#)
- [Chr](#)
- [Asc](#)
- [IsInteger](#)
- [IsFloating](#)
- [IsString](#)
- [AsInteger](#)
- [AsFloating](#)
- [AsString](#)
- [A2H](#)
- [BCD](#)
- [BIN](#)
- [H2A](#)
- [MAX](#)
- [MIN](#)
- [W2F](#)
- [MSeconds](#)

#### 子程序

- [SetB](#)
- [InvB](#)
- [ClrB](#)
- [BMOV](#)
- [FILL](#)
- [B2W](#)
- [SWAP](#)
- [W2B](#)
- [W2D](#)
- [D2F](#)
- [F2D](#)
- [AddrStringCompare](#)
- [D2Float](#)
- [Float2D](#)



- [NewNoAddr](#)
- [NewStatAddr](#)
- [NStringCompare](#)
- [ReadAddr](#)
- [WriteAddr](#)
- [F2S](#)
- [W2S](#)
- [SleepA](#)

[costants](#)

## 23. 14. 39 InStr 函数

### 一、简介

返回 str2 字符串里出现 str1 字符串的位置（从 0 开始），没找到为-1。

### 二、函数功能描述

str1

被搜索的字符串，如果不是字符串，则将之转换成字符串。

str2

要搜索的字符串，如果不是字符串，则将之转换成字符串。

### 三、案例

```
dim a as string
```

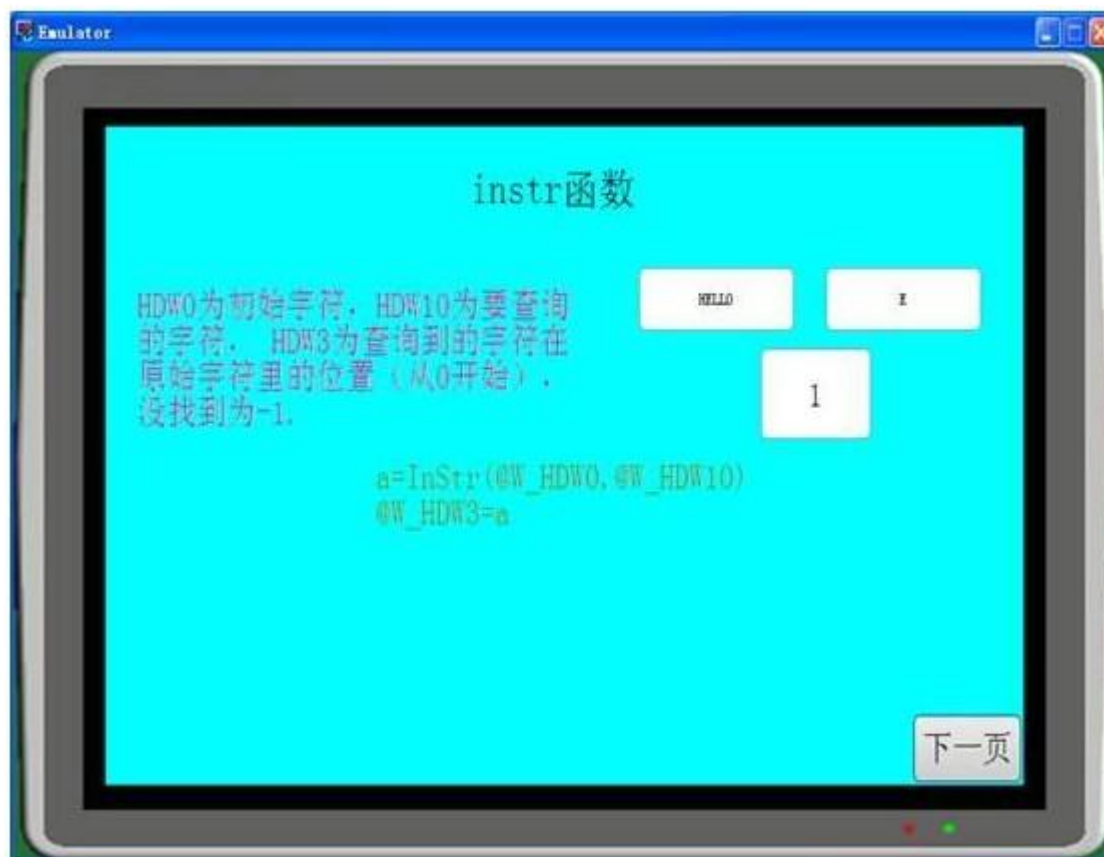
```
a = InStr("Hello", "o")      得到      Returns 4.
```

```
例：a = InStr("Hello", "He") 得到      Returns 0.
```

```
例：a = InStr("Hello", "abc") 得到      Not found. Returns -1.
```

```
float2d("@W_HDWO", a)
```

得到值如下图





dim a as string

例: @W\_HDW0= InStr("Hello", "o") 得到 Returns 4.

例: a = InStr("Hello", "He") 得到 Returns 0.

例: a = InStr("Hello", "abc") 得到 Not found. Returns -1.

例: float2d("@W\_HDW0",a)

注: 直接用地址去等值没有效果, 但函数后可以为地址。

## 23. 14. 40 InvB 子程序

### 一、简介

反转 A1 位的状态, 这是一个子程序, 所以没有返回值。A1 可以为系统地址(位)或其它类型的变量。

### 二、函数功能描述

运用这个函数会使为开关不停的自动切换状态。

### 三、案例

```
InvB(@B_HDX0.1)
```

dim a as integer

InvB(a)

@B\_HDX0.1=a

如果不直接用地址改用变量, 只会让这个为开关复位

## 23. 14. 41 IsFloating 函数

### 一、简介

如果参数为浮点数，是则返回 **TRUE**，否则返回 **FALSE**。

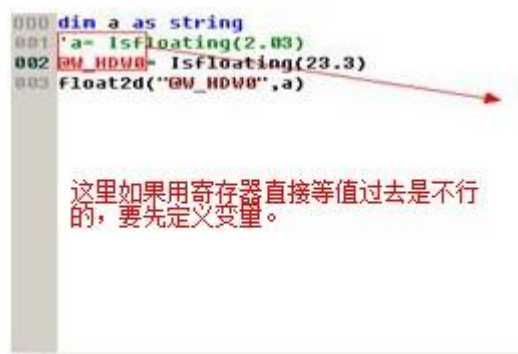
### 二、函数功能描述

判断一个参数是否为浮点

### 三、案例

```
dim      a      as      integer
a = IsFloating(20)--a=0
a = IsFloating(18.1)--a=1
```

模拟效果



## 23.14.42 IsInteger 函数

### 一、简介

如果参数为浮点数，则返回 **TRUE**，否则返回 **FALSE**。

### 二、函数功能描述

判断一个参数是否为整型

三、案例            `dim a as integer`

`a = IsFloating(20)--a=0`

`a = Isinteger(18.1)--a=1`

`@W_HDW0=a`

模拟效果



`dim a as integer`

`@W_HDW0 = IsFloating(20)--a=0`

`\a = Isinteger(18.1)\--a=1`

`\@W_HDW0=a`

直接调用寄存器是无效果的，必须先定义变量。

#### 23.14.43 IsString 函数

如果参数为字符串，则返回 TRUE，否则返回 FALSE。

`val = IsString(expr)` 函数功能描述

判断一个参数是否为字符串

案例

```

000 dim a as integer
001
002 'a = Isstring("hello")'--a=0
003
004 a = Isstring("23")'--a=1
005
006 @W_HDVB=a|

```

这边跟的有双引号为字符，可以跟变量，寄存器不行

模拟效果



```

000 dim a as string
001 a= Isstring("Hello")
002 @W_HDVB=Isstring(23)
003 float2d("@W_HDVB",a) |

```

这边如果直接用寄存器等值过去的话是没有效果的，只能是变量

## 一、简介

返回将参数全部转化为小写的字符串。

## 二、函数功能描述

任何表达式。如果不是字符串，则转换为字符串。

## 三、案例

```
@W_HDW33 = LCase(@W_HDW25)
```

模拟效果



```
dim a as string
\a = LCase(123)
\@W_HDW0= LCase("HAPPY")
\@W_HDW0= LCase("HELLO")
\@W_HDW0=a
```

直接用地址等值也可以

### 23.14.45 Left 函数

## 一、简介

从一个字符串的左边开始数起返回指定个数的字符串。

## 二、函数功能描述

string

返回其最左边开始的字符的字符串。如果不是字符串，则转换为字符串。

length

要返回的字符个数。如果不是整数，则转换为整数。

如果 length 小于 1，则返回空字符串。

如果 length 大于或等于字符串中字符的个数，则返回整个字符串。

### 三、案例

```
dim a as string
```

```
@W_HDW30=Left(@W_HDW36, @W_HDW40) \Returns "Hel".
```

模拟效果





## 23.14.46 Len 函数

### 一、简介

返回字符串中字符的个数。

### 二、函数功能描述

任何表达式。如果不是字符串，则转换为字符串。

### 三、案例

```
dim a as integer
a =Len(123)      \ Returns 3.
a= Len("Hello")  \Returns    5.
@W_HDW0=a
```

模拟效果



@W\_HDW50=Len(@W\_HDW52)

上面直接用地址去等值过去也是可以的



## 23. 14. 47 Log 函数

### 一、简介

组态软件提供运算函数供用户使用---Log 函数。

### 二、函数功能描述

Log 函数：返回一个数的自然对数。用 x 的自然对数除以 n 的自然对数，可以为任意数 x 计算基于 n 的对数。表达式：变量  $a = \text{Logn}(x) = \text{Log}(x) / \text{Log}(n)$ 。

### 三、演示案例

1、本案例使用 1 个位触发脚本、一个“数值输入/显示”部件(地址 HDW101)、位开关部件(地址 HDX0.1)来演示 Log 函数。添加位触发脚本如下图所示：



2、在打开的脚本编辑区输入如下图代码：



案例说明：此脚本为求  $\log_{327}$  的值，正确的输出结果为 3；脚本第 1 行定义一个整型变量 a 用以存放运行结果，HDW101 使用内部寄存器保存变量；当屏上“位开关”HDX0.1 置 1 时触发脚本运行（注：参数只能是 plc 地址或 hmi 内部地址，地址内值的传递必需通过定义变量来完成）。

3、在工程编辑区拖放 1 个“数值输入/显示”部件(读取地址为：HDW101 用以显示结果值)和 1 个“位开关”部件（读取地址为 HDX0.1 用以触发脚本），位开关选择切换型，编辑工程成功后，模拟点击位开关，出现如下图所示界面：



## 23.14.48 Log10 函数

### 一、简介

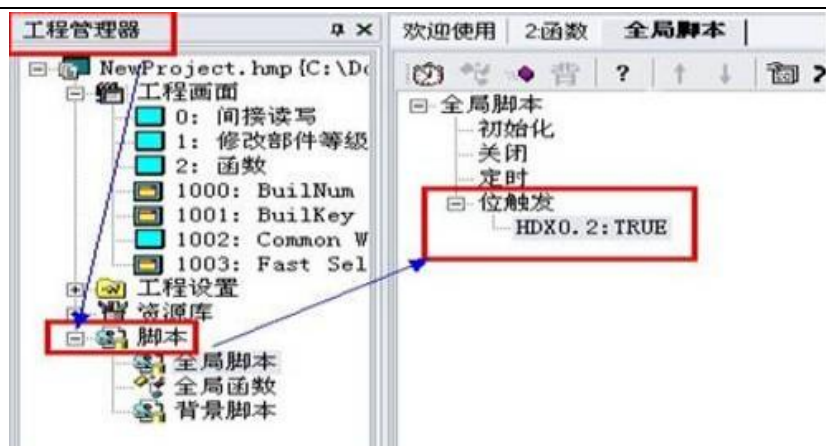
组态软件提供一系列运算函数供用户使用——Log10 函数。

### 二、函数功能描述

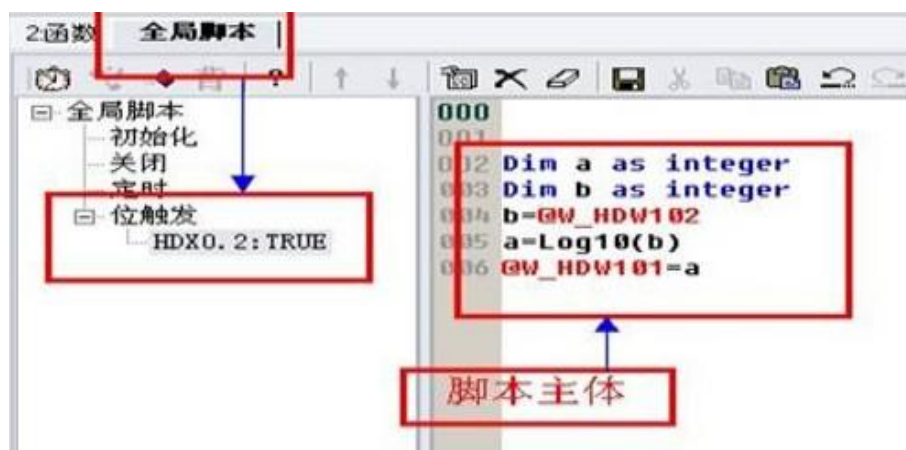
Log10 函数：返回基于 10 的对数值。用 x 的自然对数除以 10 的自然对数，可以为任意数 x 计算基于 10 的对数。表达式： $\text{Log10}(x) = \text{Log}(x) / \text{Log}(10)$  [x 为 10 的整数倍数]。

### 三、演示案例

1、本案例使用 1 个位触发脚本、两个“数值输入/显示”部件（地址 HDW101、HDW102）、位开关部件(地址 HDX0.2)来演示 Log10 函数。添加位触发脚本步骤如下图所示：



2、在打开的脚本区编写如下图代码：



案例说明：求 Log10100 正确输出结果值 2。脚本第 1、2 行分别定义整型变量 a、b，a 用以存放运行结果，b 作为内部地址 HDW102 值传递的中间变量；HDW101 用以保存显示变量，HDW102 用来输入上述表达式中 X 的值；HDX0.2 作为脚本触发位，当屏上“位开关”HDX0.2 置 1 时触发脚本运行（注：参数只能是 plc 地址或 hmi 内部地址，地址内值的传递必需通过定义变量来完成）。

3、在工程编辑区拖放 2 个“数值输入/显示”部件和 1 个“位开关”部件，分别给地址 HDW101、HDW102 和 HDX0.1，位开关选择切换型，编辑工程成功后，显示计算结果“2”模拟将出现如下图所示界面：



## 23.14.49 LTrim 函数

### 一、简介

组态软件提供字符串操作函数接口供用户使用—LTrim 函数。

## 二、函数功能描述

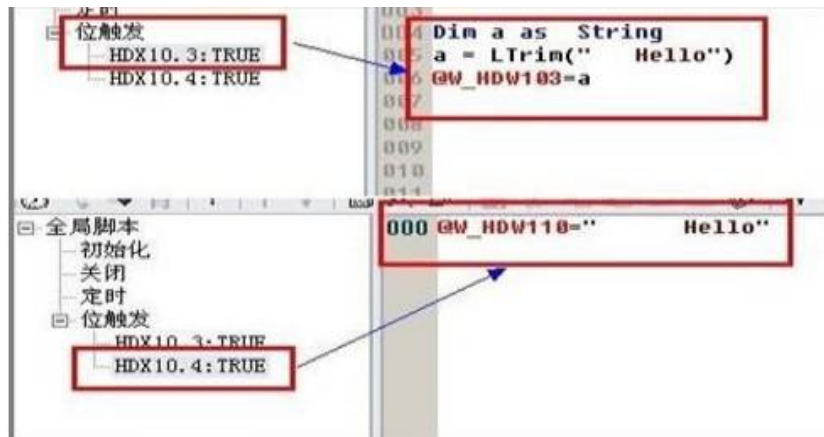
**LTrim 函数：**去掉一个字符串左边空格并返回。表达式：变量=LTrim("字符串")。

## 三、演示案例

1、案例采用对比法说明：使用两个位触发脚本、两个“文本输入/显示”部件(地址 HDW104、HDW110)、两个位开关部件(地址 HDX10.3、HDX10.4)来对比演示 LTrim 函数。添加位触发脚本如下图所示：



2、在打开的脚本编辑区输入如下图代码：



案例说明：采用两个位触发脚本进行对比说明，两个脚本触发同一串字符串。

HDX10.3 的触发脚本含义：当 HDX10.3 置 1 脚本被触发，执行 LTrim 函数，文本输入/显示部件将显示经函数处理后的字符串；HDX10.4 的触发脚本含义：当 HDX10.3 置 1 脚本被触发，地址为 HDW110 的文本输入/显示部件将显示赋值后不经处理的字符串。

3、在工程编辑区拖放两个“文本输入/显示”部件和 2 个“位开关”部件，分别给地址 HDW103、HDW110 和 HDX10.3、HDX10.4，位开关选择切换型，编辑工程成功后，离线模拟点击位开关将出现如下图所示界面：





## 23. 14. 50 MAX 函数

### 一、简介

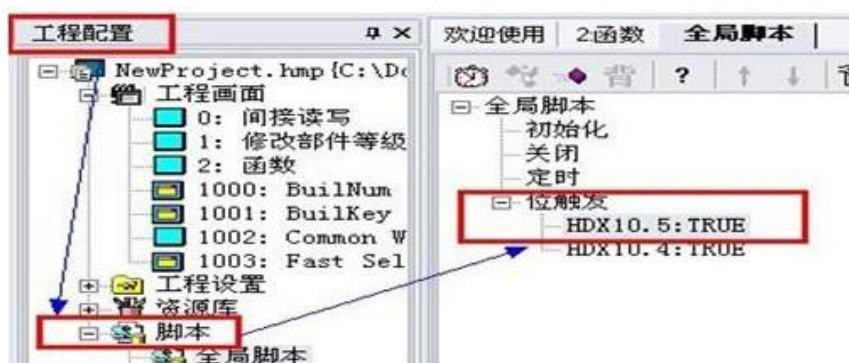
组态软件提供数值大小比较函数供用户使用—MAX 函数。

### 二、函数功能描述

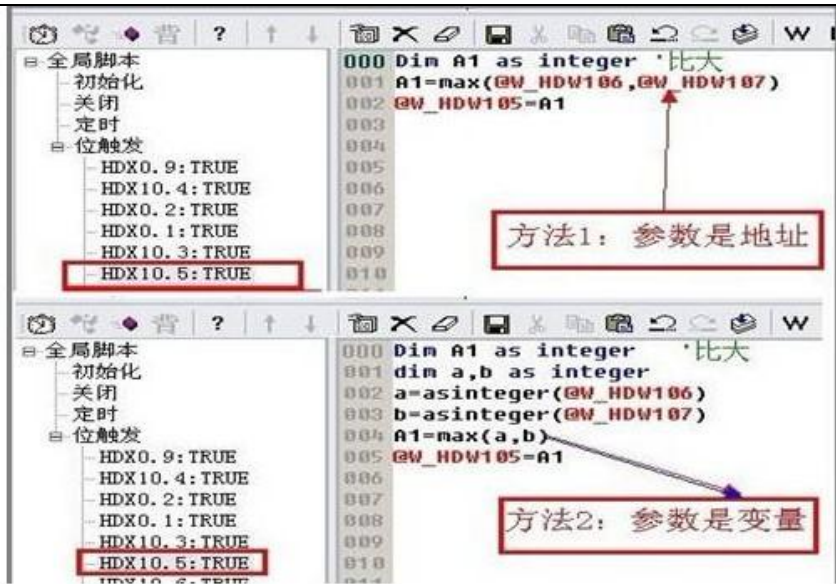
Max 函数表达式:  $A1=MAX(\text{字地址 } 1, \text{字地址 } 2)$  或  $A1=MAX(\text{变量 } A2, \text{变量 } A3)$ , 将两个字地址寄存器中数值作比较, 把大的赋值给变量 A1 并输出, 使用到的数值类型可以是任何数值类型的, 但必须是同类型。

### 三、演示案例

1、本案例使用 1 位触发脚本、3 个“数值输入/显示”部件 (地址 HDW105-HDW107)、1 个位开关部件(地址 HDX10.5)来演示 Max 函数。添加的位触发脚本如下图所示:



2、在打开的脚本编辑区输入的代码如下图所示:



案例说明：本例用到的 3 个字地址为无符号整型数。定义 1 整型变量 A1 用来存放比较结果，字地址 HDW105 用来显示输出比较大的数，HDW106 和 HDW107 用来存放显示比较数。

3、在工程编辑区拖放 3 个“数值输入/显示”部件和 1 个“位开关”部件，分别给地址 HDW105、HDW106、HDW107 和 HDX10.5，位开关选择切换型，编辑工程成功后，离线模拟点击位开关，将出现如下图所示界面：



23. 14. 51 Mid 函数

一、简介

软件提供一字符串截取函数接口供用户使用—Mid 函数。

二、函数功能描述

Mid 函数：返回一个字符串中指定长度的字符串，不是字符串的转为字符串。

表达式:变量=mid(“string”/字地址,start, length); 第 1 个参数为要从截取的字符串或要操作的字地址；start 为开始截取的字符位置在 string+1; length 为要截取的字符个数。（参数: “string”/字地址、start、

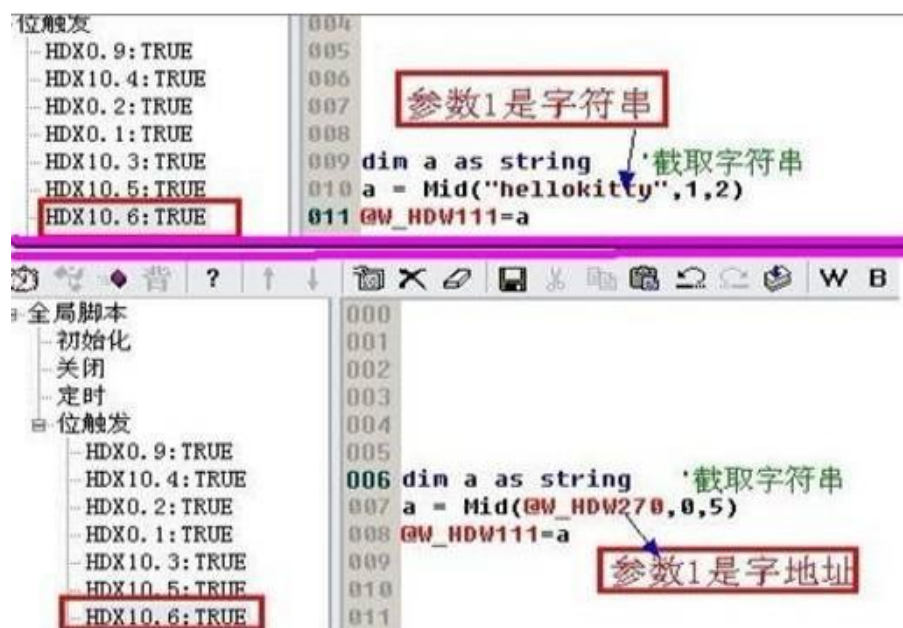
length 不是整数的则转换为整数)。

### 三、演示案例

1、本案例使用 1 位触发脚本、1 个“文本输入/显示”部件(地址 HDW111)、位开关部件(地址 HDX10.6)来演示 Mid 函数。添加位触发脚本如下图所示：



2、在打开的脚本编辑区输入如下图代码：



脚本说明：采用位触发脚本，触发位 HDX10.6，在脚本中定义 1 个字符串型变量 a 作为中间变量暂存截取结果，当位置 1 时触发脚本运行；将从字符串的第二个字符(“e”)开始截取 2 个字符，再用一字地址显示截取结果(“e1”)。

3、在工程编辑区拖放 1 个“文本输入/显示”部件和 1 个“位开关”部件，分别给地址 HDW111 和 HDX10.6，位开关选择切换型，编辑工程成功后，离线模拟点击位开关，将输出结果如下图：





### 23.14.52 MIN 函数

#### 一、简介

组态软件提供数值比较函数供用户使用---Min 函数。

#### 二、函数功能描述

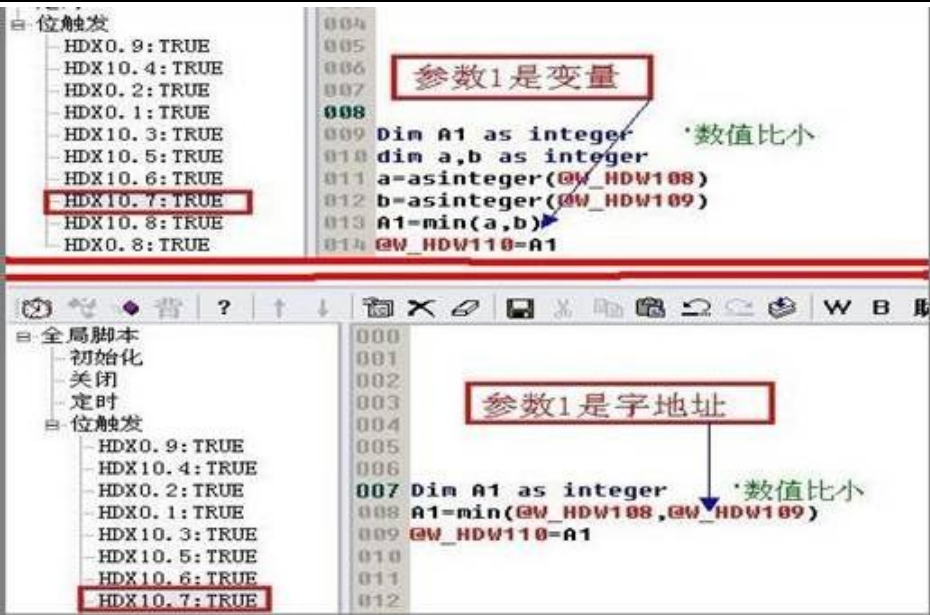
Min 函数表达式： $A1 = \text{Min}(\text{字地址 } 1, \text{字地址 } 2)$  或  $A1 = \text{Min}(\text{变量 } A2, \text{变量 } A3)$ ，将两个字地址寄存器中数值作比较，将数值小的赋值给变量 A1 并输出，使用到的数值类型必须是同种类型。

#### 三、演示案例

1、本案例使用 1 位触发脚本、3 个“数值输入/显示”部件（地址 HDW108-HDW110）、1 个位开关部件(地址 HDX10.7)来演示 Min 函数。添加的位触发脚本如下图所示：



2、在打开的脚本编辑区输入代码如下图所示：



案例说明：本例用到的 3 个字地址为无符号整型数。定义 1 整型变量 A1 用来存放比较结果，字地址 HDW110 用来显示输出比较小的那个数，HDW108 和 HDW109 用来存放显示比较数。

3、在工程编辑区拖放 3 个“数值输入/显示”部件和 1 个“位开关”部件，分别给地址 HDW108、HDW109、HDW110 和 HDX10.7，位开关选择切换型，编辑工程成功后，离线模拟点击位开关，将输出比较结果如下图所示界面：



23. 14. 53 MSeconds 函数

一、简介

组态软件提供函数供用户使用查看系统参数---MSeconds 函数。

二、函数功能描述

MSeconds 函数：返回当前系统的微秒值。表达式:变量=MSeconds(), 返回值由系统定，本函数无参数。

### 三、演示案例

1、本案例使用 1 位触发脚本、1 个“数值输入/显示”部件(地址 HDW120)、位开关部件(地址 HDX10.8)来演示 MSeconds 函数。添加位触发脚本如下图所示：

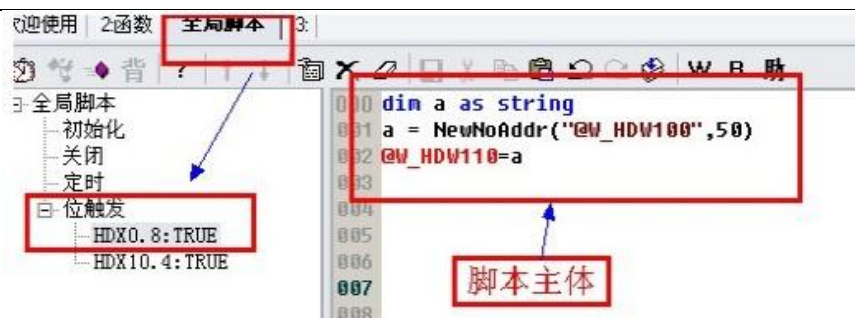


2、在打开的脚本编辑区输入如下图代码：



脚本说明：脚本功能显示当前系统的微秒值。定义 1 浮点型变量 a 作为存储中介；用 1 字地址来显示值。

3、在工程编辑区拖动 1 个“数值输入/显示”部件和 1 个“位开关”部件，分别给地址 HDW120 和 HDX10.8，位开关选择切换型，编辑工程成功后，离线模拟点击位开关，出现结果如下图所示：



## 23.14.54 NewNoAddr 函数

### 一、简介

组态软件提供变址函数供用户使用——NewNoAddr 函数。

### 二、函数功能描述

NewNoAddr 函数：指定地址偏移 length 位数。表达式：字符串变量= NewNoAddr(指定地址,length)，hmi 内部地址和 plc 地址皆可，length 为整数。

### 三、演示案例

1、案例使用 1 个位触发脚本、1 个“文本输入/显示”部件(地址 HDW110)、1 个位开关部件(地址 HDX0.8)来演示 NewNoAddr 函数。添加位触发脚本如下图所示：



2、在打开的脚本编辑区输入如下图代码：

脚本说明：用位 HDX0.8 置 1 来触发脚本运行，定义 1 个字符串型变量 a 以存放偏移后的地址；用 1 个字地址 HDW110 显示偏移后的地址；HDW100 为要偏移的原地址；脚本运行后实际操作地址变为 HDW150。

3、在工程编辑区拖放 1 个“文本输入/显示”部件（用以显示偏移后的地址）和 1 个“位开关”部件，分别给地址 HDW110 和 HDX0.8，位开关选择切换型，编辑工程成功后，离线模拟点击位开关，出现如

下图所示界面：



### 23. 14. 55 NewStatAddr 函数

#### 一、简介

组态软件提供一变站(外接设备号)函数供用户使用——NewStatAddr 函数。

#### 二、函数功能描述

NewStatAddr 函数：指定站偏移 length 位数。表达式：字符串变量 = NewStatAddr(地址, length)，hmi 内部地址和 plc 地址皆可，length 为整数。该函数用于多设备的环境。

#### 三、案例说明

1、在打开的脚本编辑区输入如下图代码：



脚本说明：用位 HDX0.9 置 1 来触发脚本运行(亦可用其他类脚本触发运行)。定义 1 个字符串型变量 a 用来存放偏移后的地址；再用 1 个字地址显示偏移后的地址。此脚本运行后便能使原先 1 号站地址偏移到 3 号站地址（注：要接上设备方可实现变站功能）。

### 23. 14. 56 NSStringCompare 函数

#### 一、简介

组态软件提供字符串长度比较函数供用户使用——NSStringCompare 函数。

#### 二、函数功能描述

NSStringCompare 函数表达式：变量= NSStringCompare(“地址 1”,字符串 1,length)，比较“地址 1”的字符串和字符串 1 的 length 长度范围内的字符是否完全相等；相等输出 1，不等输出 0。



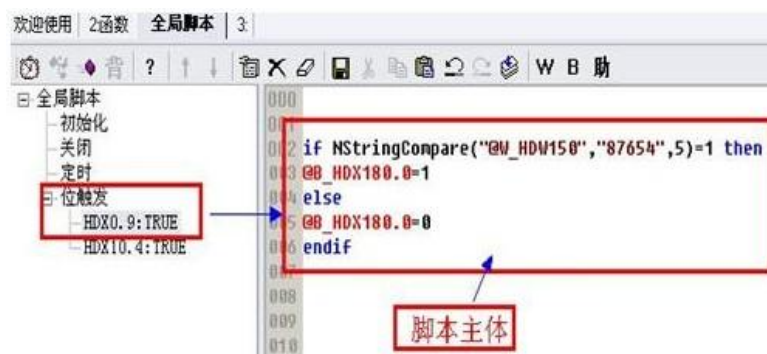
参数定义：参数 1 为地址所存内容；字符串 1 为参照比较的字符串；length 为字符串实际长度。

### 三、演示案例

1、本案例使用 1 位触发脚本、1 个“文本输入/显示”部件（地址 HDW150）、两个位开关部件(地址 HDX0.9、HDX180.0)来演示该函数。添加位触发脚本步骤如下图所示：



2、在打开的脚本编辑区输入如下图代码：



脚本说明：若 HDW150 输入的字符串和比较字符串“87654”相等，则位 HDX180.0 置 1；不相等则位 HDX180.0 置 0。

3、在工程编辑区拖放 1 个“文本输入/显示”部件(用以输入要比较的字符串)和 2 个“位开关”部件，分别给地址 HDW150 和 HDX0.9(用以触发脚本)、HDX180.0(用以显示比较结果)，位开关皆为切换型，编辑工程成功后，模拟点击位开关 HDX0.9，出现如下图所示：



23.14.57 操作符

+

result = expression1 + expression2

计算两个数的和。

expression1	expression2	result
integer	integer	integer
floating	floating	floating
string	string	string
integer	floating	floating
integer	string	string
floating	integer	floating
floating	string	string
string	integer	string
string	floating	string

-

result = expression1 - expression2

计算两个数的差。

expression1	expression2	result
integer	integer	integer

floating	floating	floating
string	string	floating
integer	floating	floating
integer	string	floating
floating	integer	floating
floating	string	floating
string	integer	floating
string	floating	floating

\*

result = expression1 \* expression2

计算两个数的乘积。

expression1	expression2	result
integer	integer	integer
floating	floating	floating
string	string	floating
integer	floating	floating
integer	string	floating
floating	integer	floating
floating	string	floating
string	integer	floating
string	floating	floating

/

result = expression1 / expression2

计算两个数相除的结果。

expression1	expression2	result
integer	integer	floating
floating	floating	floating
string	string	floating
integer	floating	floating
integer	string	floating



floating	integer	floating
floating	string	floating
string	integer	floating
string	floating	floating

<<  
result = expression1 << expression2  
将 expression1 的数值左移 expression2 个位，并将位移结果返回。

>>  
result = expression1 >> expression2  
将 expression1 的数值右移 expression2 个位，并将位移结果返回。

~  
result = ~expression  
对一个表达式所代表的数值执行位取反操作。返回值的类型是整型。

mod (%)  
result = expression1 mod expression2  
result = expression1 % expression2  
返回两个数相除的余数。返回值的类型是整型。

or (|)  
result = expression1 or expression2  
返回两个表达式逻辑或或位或的结果。返回值的类型是整型。

and (&)  
result = expression1 and expression2  
返回两个表达式逻辑与或位与的结果。返回值的类型是整型。

not (!)  
result = not expression  
对一个表达式所代表的数值执行逻辑取反操作。返回值的类型是整型。

xor (^)  
result = expression1 xor expression2  
result = expression1 ^ expression2  
对两个表达式执行位异或操作，返回值的类型是整型。

23. 14. 58 power 函数

返回对 expr1 求 expr2 幂的结果。

var = power(expr1, expr2)

语法

*expr1, expr2*

任何表达式。

下例说明了 power 函数的用法：

```
dim a as integer
a=power(2, 3)
@W_HDW107=a
a=8
```

### 23. 14. 59 程序结构

Script 的程序结构

Script 的程序可能有任意个[子程序](#)或[函数](#)。程序从最后一个 End Sub 或 End Function 标记处之后的第一条指令开始执行。

### 23. 14. 60 RadToDeg 函数

将弧度值转换成角度值。

`var = RadToDeg(expr)`

示例：dim a as integer

```
a = RadToDeg(π )
```

```
@W_HDW11=A
```

注释：返回的值是 180

### 23. 14. 61 RAND 函数

返回一个随机数。

`var = rand([nSeedValue])`

语法

*nSeedValue*

指定随即种子的值。这是一个可选的参数。

说明

```
@W_HDW0=rand(@W_HDW0)
```

### 23. 14. 62 ReadAddr 函数

读 A1 的值

语法

`word = ReadAddr(A1)`

说明

A1:A1 必须为以 '@' 字符开始的系统内存或 PLC 地址（如 @W\_00002）。

示例

```
Dim word as integer
```

@W\_HDW100=10

word = ReadAddr("@W\_HDW100") --word=10

### 23. 14. 63 Right 函数

从一个字符串的右边返回指定个数的字符串。

val = Right(*string*, *length*)

语法

*string*

返回其最右边开始的字符串。如果不是字符串，则转换为字符串。

*length*

要返回的字符个数。如果不是整数，则转换为整数。

说明

如果 *length* 小于 1，则返回空字符串。

如果 *length* 大于或等于字符串中字符的个数，则返回整个字符串。

下例说明了 Right 函数的用法：

```

a = Right("Hello", 3, ' ', Returns
3)          "llo"
a = Right("Hello", 5, ' ', Returns
50)         "Hello"
a = Right("Hello", -50, ' ', Returns ""
-50)

```

### 23. 14. 64 RTrim 函数

返回一个字符串的拷贝，将右边的空白字符去掉。

val = RTrim(*str*)

语法

*str*

被截短的字符串。如果不是字符串，则转换为字符串。

说明

下例说明了 RTrim 函数的用法：

```

a = RTrim("   -Hello-   ", ' ', Returns "
")          -Hello-".

```

### 23. 14. 65 SetB sub 子程序

将位 A1 设为 TRUE (1).

语法

SetB(A1)

说明

这是一个子程序，所以没有返回值。A1 可以为系统地址(位)或其它类型的变量。

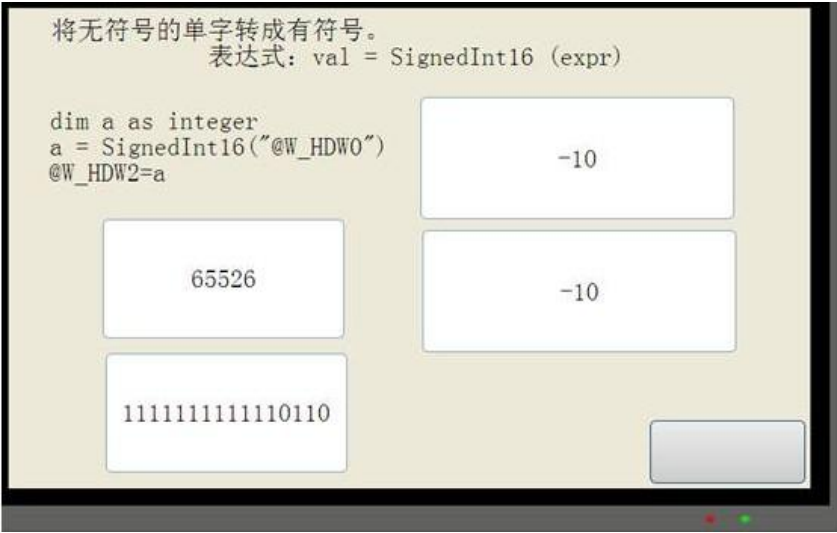
示例

```
SetB(@B_00001)
```

注释：返回 @B\_00001=1

23. 14. 66 SignedInt16 函数

函数功能	
将无符号的单字转成有符号	
表达式	val = SignedInt16 (expr)
示例	<pre>dim a as integer a = SignedInt16("@W_HDW0") @W_HDW2=a</pre>
输出	当 a 为负数时 HDW2 的值也为负数



23. 14. 67 SignedInt32 函数

函数功能	
将无符号的双字转成有符号	
表达式	val = SignedInt32 (expr)
示例	<pre>dim a as integer a = SignedInt32("@W_HDW0") @W_HDW2=a @W_HDW3=a&gt;&gt;16</pre>

输出	当 a 为负数时 HDW2 的值也为负数
----	----------------------

23. 14. 68 Sin 函数

返回一个角度的正弦值。

var = Sin(*expr*)

Syntax

*expr*

任意表达式，其值为一个角度的弧度值。

Remarks

要从弧度值转换成角度值请调用 RedToDeg 函数。

23. 14. 69 SleepA 函数

等待时间 A1ms

语法

SleepA(A1

说明

A1:A1 可以为系统地址(浮点)或其它类型的变量。

示例

等待多少毫秒

SleepA(10) 等待 10ms

23. 14. 70 Sqr 函数

返回一个数的平方根。

var = Sqr(*expression*)

Remarks

如果 *expression* 为负数，Sqr 其绝对值的平方根。

下例说明了 Sqr 函数的用法：

a = '4'

Sqr(4) Returns 2.

a = '0'

Sqr(0) Returns 0.

23. 14. 71 Sub

声明 Sub（子程序）的名称，参数，以及子程序体的代码。

Sub *name* (*arglist*)

*statements*

End Sub

语法

---

*name*

子程序名称. 参看标准 [variable](#) 命名规则。

*arglist*

代表参数的变量列表, 当子程序被调用时, 这些参数会被传递进入。多个参数用逗号分隔。

*statements*

子函数体中运行的代码集合。

### 说明

可以对 *varlist* 里的变量定义其类型或不定义, 或者仅写 **Sub** 子程序名称。

不能在任意其它程序体中定义一个 Sub 子程序。

调用 Sub 时, 先写函数名, 后跟参数列表。

Sub 声明必须先于 Sub 调用。

下例说明了 Sub 的用法:

```
sub samesub(a,b as integer)
```

```
    c=a+b
```

```
    @W_HDW0=c
```

```
endsub
```

```
samesub(1,12)
```

## 五、工程模拟效果图

### 部件相关信息

@W\_HDW0为无符号  
十进制输入/输出

#####

```
工程脚本: sub samesub(a,b as integer)
            c=a+b
            @W_HDW0=c
        endsub
        samesub(1, 12)
```

@W\_HDW0为无符号  
十进制输入/输出

13

```
工程脚本: sub samesub(a,b as integer)
            c=a+b
            @W_HDW0=c
        endsub
        samesub(1, 12)
```

调用函数名

## 23. 14. 72 SWAP 子程序

### 一、简介

对从 A1 开始的一段内存交换其高字节和低字节。A2 表示多少个字节的内存被做这个操作。

### 二、语法

SWAP(A1,A2)

### 三、说明

这是一个子程序，所以没有返回值。A1 必须为以 '@' 字符开始的系统内存或 PLC 地址（如 @W\_00002）。A2 可以为系统地址(整型)或其它类型的变量。

### 四、示例

@W\_HDW103=0X1234

@W\_HDW104=0X2345

@W\_HDW105=0X2565

@W\_HDW106=0X2675

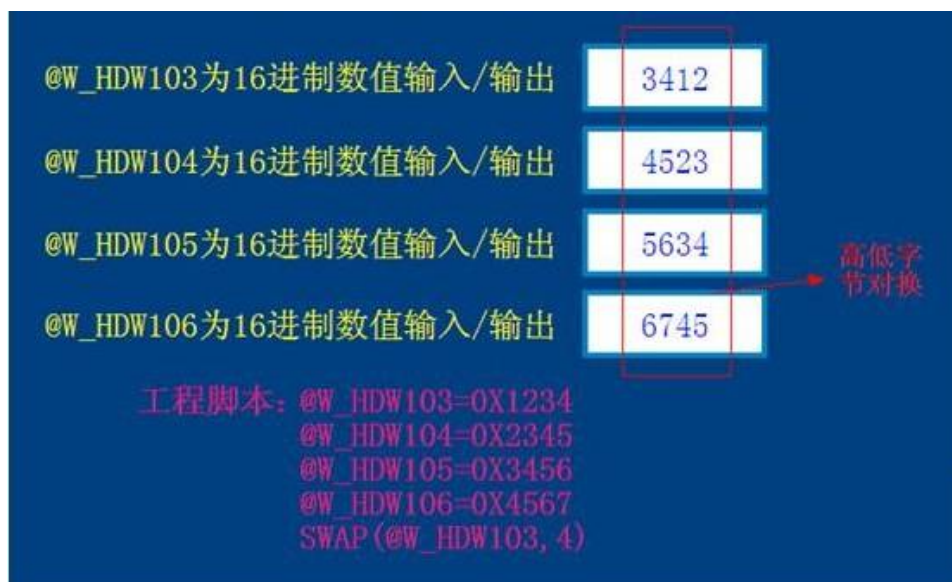
SWAP(@W\_HDW103,4)

### 五、工程效果图

部件相关信息



模拟效果图



### 23. 14. 73 Tan 函数

#### 一、简介

返回一个角度的正切值。

#### 二、语法

var = Tan(expr)

#### 三、说明

expr

任意表达式，其值为一个角度的弧度值。

要从弧度值转换成角度值请调用 RedToDeg 函数。



#### 四、示例

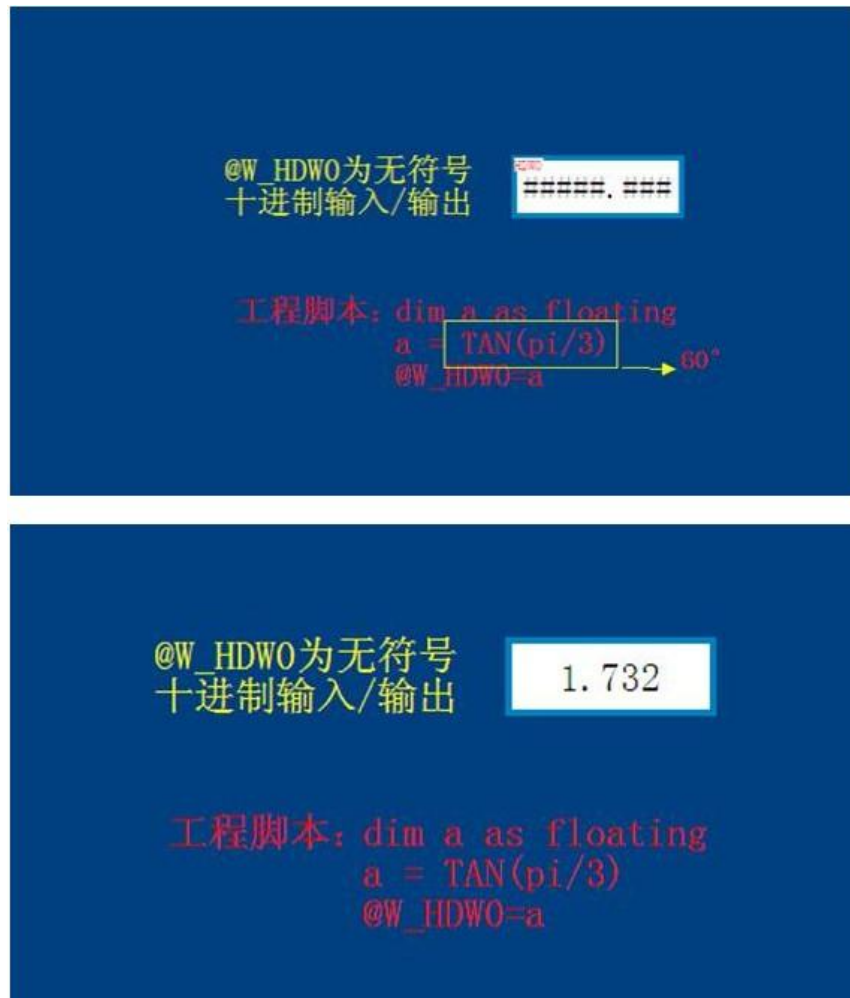
Dim a as floating

a=TAN(pi/3)

@W\_HDW0=a

#### 五、工程模拟效果图

部件相关信息



#### 23.14.74 Trim 函数

##### 一、简介

返回一个前后都没有空白字符的字符串

##### 二、语法

**val** = Trim(*str*)

##### 三、说明

*str*

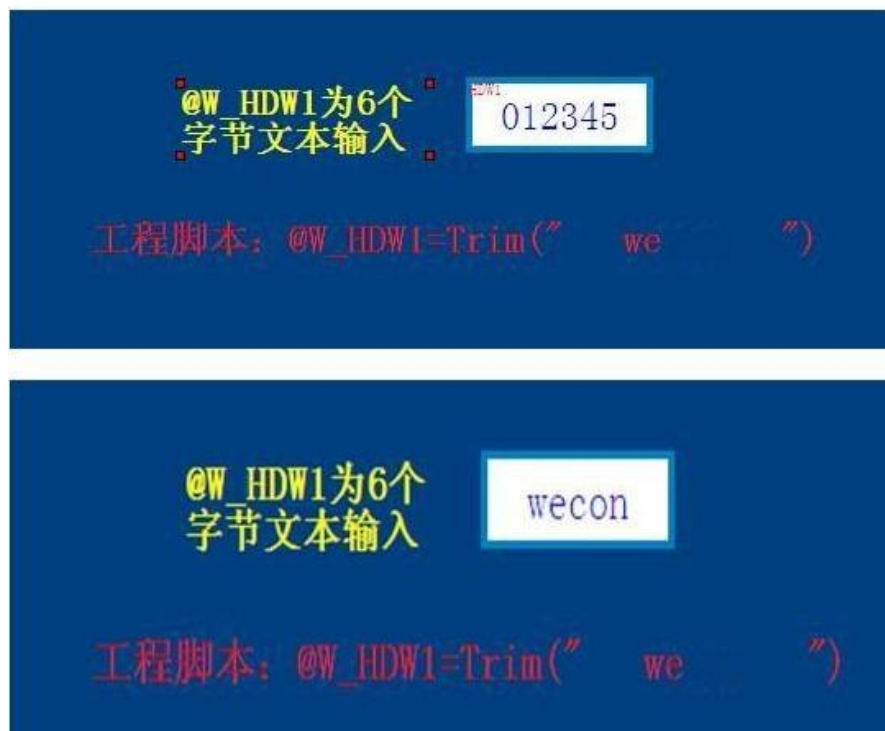
被截短的字符串。如果不是字符串，则转换为字符串。

##### 四、示例

@W\_HDW1=Trim( we )

## 五、工程模拟效果

部件相关信息



### 23. 14. 75 UCase 函数

返回将参数全部转化为大写的字符串。

val = UCase(*expr*)

语法

*expr*

任何表达式。如果不是字符串，则转换为字符串。

说明



### 23. 14. 76 变量

变量是有名字的存储位置，其有在程序运行的过程中可以被修改的数据。每一个变量都有唯一标识它的名字以及它的作用范围。

变量名：

- 必须以字母开头（如果是系统内存地址或 PLC 地址，则必须以'@'打头），其他字符可以是字母、数字、下划线（'\_'）。
- 不得多于 MAX\_NAME\_LENGTH 个字符。

变量可以保存如下类型的数据：

string

floating

integer

在运行中，以 [Dim](#) 声明的变量不能修改其数据类型，但是没有声明的变量可以修改其数据类型。

### 23. 14. 77 W2D 子程序

将 A2 从一个单字的数转换成双字的数，并将结果保存在 A1。

语法

**W2D**(A1, A2)

说明

这是一个子程序，所以没有返回值。A1 必须为以 '@' 字符开始的系统内存或 PLC 地址（如 @W\_00002）。  
A2 可以为系统地址 (WORD) 或其它类型的变量。

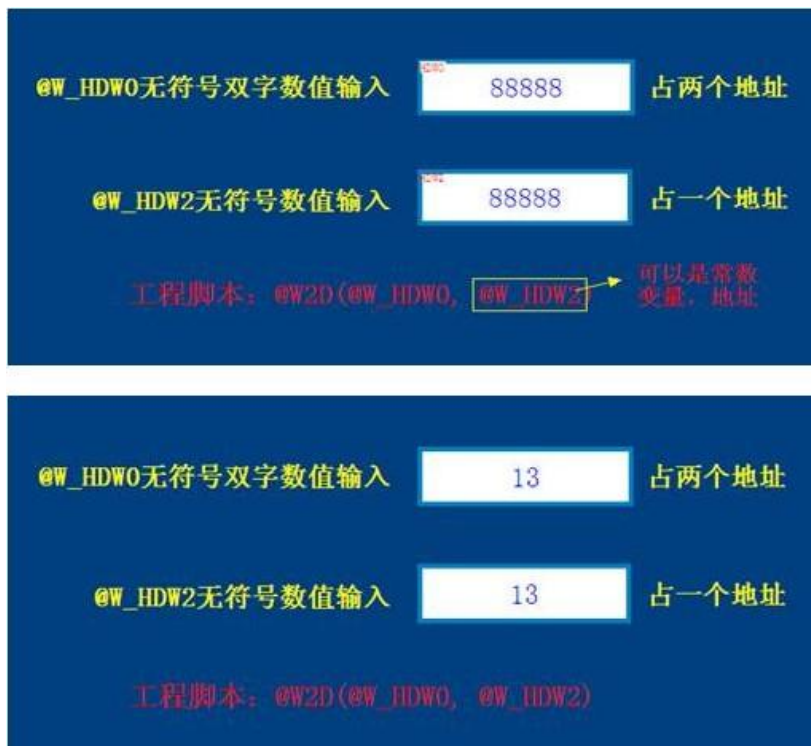
示例

**W2D**, 将 A2 从一个单字的数转换成双字的数

@W\_HDW0 无符号双字数值输入；

@W\_HDW2 无符号单字数值输入；

**W2D(@W\_HDW0, @W\_HDW2)**



## 23. 14. 78 W2F 函数

将一个 16 位的整数转换成 32 位的浮点数。要转换的整数在 A2 地址，转换成的浮点数保存到 A1 及其下一个字。

语法

**A1 = W2F(A2)**

说明

返回值是一个浮点数。 A2 可以为以 '@' 字符开始的系统内存或 PLC 地址（如 @W\_HDW51）或其它变量，A1 可以为系统地址或其它已声明的变量。

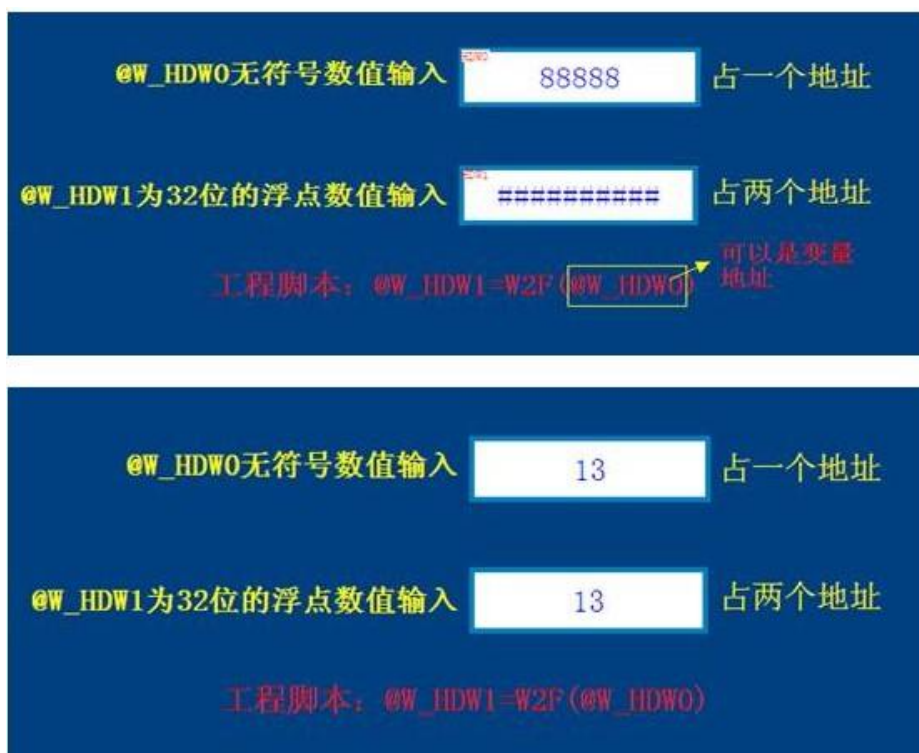
示例

**W2F**, 由 16 位的整数转换成 32 位的浮点数。

@W\_HDW0 无符号数值输入；

@W\_HDW1 为 32 位浮点数值输入；

**@W\_HDW1=W2F(@W\_HDW0)**



#### 23. 14. 79 W2S 函数

把 A1 的整形转成字符型，按 s1 格式存储存 A2

##### 语法

W2S(A1, A2, S1)

##### 说明

A1 A2:A1A2 必须为以'@' 字符开始的系统内存或 PLC 地址（如@W\_00002）。

S1:存储的格式。

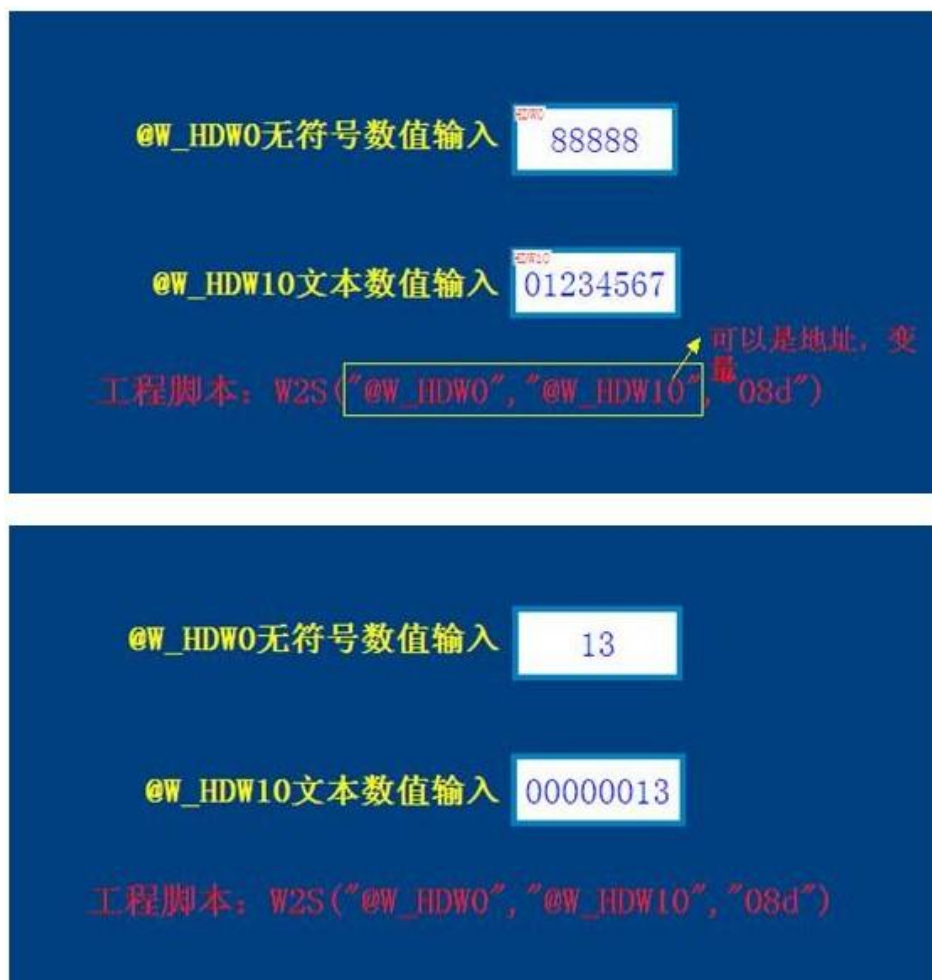
##### 示例

W2S，由字转换成字符串，可以指定格式，如

@W\_HDW0 无符号数值输入；

@W\_HDW10 文本数值输入

W2S("@W\_HDW0", "@W\_HDW10", "08d")



## 23. 14. 80 WHILE ... WEND

### 一、简介

当 condition 为 True 时，执行一系列指令。

While condition

statements

Wend condition

表达式，其值为 True 或 False.

statements

一条或多条在 condition 为 True 时执行的指令。

### 二、语法

While condition

statements

Wend condition

### 三、说明

如果 condition 为 True，statements 里的所有指令都会被执行，直到碰到 Wend 指令。然后控制权回到 While 指令并且再一次检查 condition 的值。如果 condition 仍然为 True，上面的过程会

被重复。如果它不是 True，则执行 **Wend** 指令之后的程序。与 **For** 不同的是，**While** 每一次循环都会检查 *condition* 的值。

#### 四、示例

初始化: HDW0=100

周期脚本: while @W\_HDW0>50

@W\_HDW0=@W\_HDW0-1

wend

#### 五、工程模拟效果图

部件相关信息



模拟效果图



#### 23.14.81 WriteAddr 函数

把 A2 值写入 A1 里

##### 语法

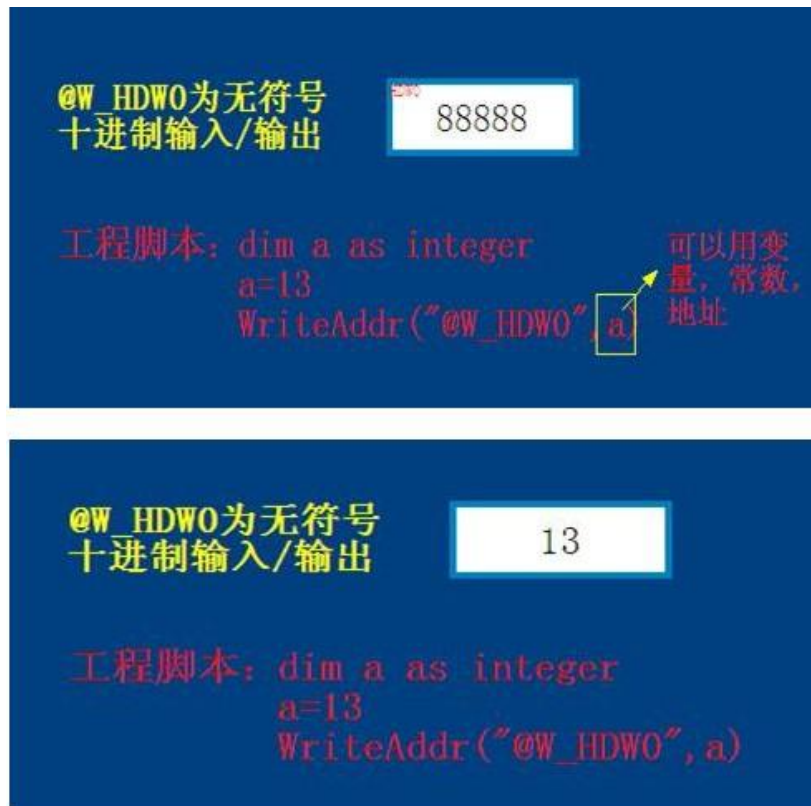
WriteAddr(A1,A2);

##### 说明

A1,A2:A1 必须为以'@'字符开始的系统内存或 PLC 地址(如@W\_00002)。A2 可以为系统地址(浮点)或其它类型的变量。

## 示例

```
dim f as integer          定义 f 为整数
f=13
WriteAddr("@W_HDWO",f) --- @W_HDWO=13
```






24.1 编译过程

工程编辑好后，需要编译才能模拟或者下载到工程中运行。软件的编译过程主要完成如下工作：

- 检验画面或者工程中可能存在的用户错误或者警告；
- 优化处理资源库，使得画面能够以更快速度在 HMI 上运行；
- 将根据工程参数的“编译语言”设置，链接文本库的文本资源（关于编译语言，请参考文本库一章）
- 处理工程中用到的 Windows 字体；
- 将工程与画面打包成 HMI 文件，以便下载到 HMI 上运行。

从菜单“工具”--“工程编译”可以完成编译动作，也可以从工具条的图标来编译工程。在编译之前会遍历所有已打开的画面，保存所做的修改。

24.2 编译错误(警告)

编译过程中，软件将会提示画面中存在的错误或者警告。表 16-1 列举了最常见的错误警告及其处理办法。

表 24-1 编译错误列表

序号	内容	类型	原因及后果
	操作地址(或者触发地址、监控地址)格式不对	警告	部件的地址书写格式与工程对应的 PLC 地址格式不一致，或者当前 PLC 类型不能识别该地址； 或者部件所引用的地址库的条目已经被删除 可能导致部件无法与 PLC 正确通信
	功能开关指定的画面号不存在	警告	所指向的画面号不存在，或者已经被删除； 可能导致功能开关无法切换到指定画面号
	所指向的键盘画面不存在	警告	数字输入、字符串输入所指向的键盘画面不存在，或者已经被删除； 可能导致无法输入部件无法正确弹出窗口
	直接画面显示所指向的画面号不存在	警告	直接画面显示部件所指向的画面号不存在，或者画面已经被删除； 可能导致直接画面显示部件无法显示指定的画面。
	数字输入部件(或者字符输入部件)所指定的键盘画面不是子画面	警告	键盘画面号所指向的画面不是子画面； 可能导致键盘画面无法正常显示。
	所指向的画面不是子画面	警告	画面显示类部件或者键盘的画面不是子画面，可能导致画面无法正常显示。
	起始画面不存在	错误	起始画面已经删除或者变成了子画面，这样的工程在 HMI 上将无法正常启动。

	配方的触发地址、起始地址不对	警告	同 1
	报警的触发地址格式不对	警告	同 1

## 25 下载

### 25.1 工程下载/上传

下载是将工程编译生成的 hmt 文件下载到 HMI 的过程，可以通过串口或者 USB 下载。

一、串口下载过程如下：

- 1、 下载程序之前，首先需要用串口连接线将 HMI 的调试口 (COM2) 和 PC 的串口连接起来；
- 2、 串口连接好后，HMI 将自动进入下载状态；
- 3、 从菜单 “工具” → “下载工程”，如图 25-1 所示



图 25-1 下载工程

- 4、 也可以从软件安装目录进入下载界面，如图 25-2 所示



图 25-2 上传/下载界面

表 25-1 上传/下载界面说明

属性名称	说明
串口	PC 的下载串口选择
波特率	PC 串口与 HMI 进行通信的波特率，波特率默认是 115200bps
文件类型	要传输的文件是配方还是编译好的 HMT 文件
HMI->P	从 HMI 上传所需要的文件到 PC 保存

C	
PC→HMI	从 PC 下载工程或者配方到 HMI 上
I	
关闭	关闭对话框

- 5、工程下载完毕后，需要重新启动软件，才能让新的 HMT 文件起运行；
- 6、如果下载过程出错，程序将会给出相应提示，HMI 上原来的 HMT 文件将继续保留。

**9 针下载线的接线方式为：**

9 针母头	9 针母头
2	3
3	2
5	5

二、用 **USB 下载线** 下载，需要先安装 USB 驱动。

当您第一次用 USB 数据线连接设备与您的计算机时，

## 25.2 配方上传/下载

配方的上传/下载与工程是一样的，配方文件的下载成功将会覆盖原有的配方文件，也可以将 HMI 的配方上传到 PC 上来保存。

## 26 模拟

模拟是利用软件所提供的模拟功能，在 PC 机上模拟实现 HMI 的功能，用户可以模拟功能方便地发

现工程所存在的逻辑错误，简化工程调试环节，提高工作效率。

在软件中，提供了两种类型的模拟：离线模拟与在线模拟。

从菜单“工具”→“**离线模拟**”可以进入离线模拟，

从菜单“工具”—“**在线模拟**”可以进入在线模拟。

在模拟之前，如果工程或者画面已经做了修改，软件会提示是否需要重新编译，然后进入模拟运行。

## 26.1 离线模拟

离线模拟是在 PC 的 RS232 口不接 PLC 设备的情况进行模拟，这种模拟可能会导致一些需要和 PLC 进行通信的部件无法通过离线模拟来反映其运行情况。

## 26.2 在线模拟

在线模拟则是在 PC 的 RS232 接口已经连接了相应的 PLC 设备，并且 PLC 已经预设好了梯形图程序，这种模拟可以很好地反映工程在 HMI 上以及在实际工程环境中的运行情况。

运用在线模拟可以测试工程和 PLC 通讯是否正常。在线模拟若通讯不上，可能的原因是通讯参数设置错误，请检查“通讯口设置”的各个参数是否已经和 PLC 的通讯参数一致。

# 27 HMI 内部保留寄存器

## 27.1 概述

为了编写工程的方便，软件提供了几大类的内部存储器，用户可以将它们当作程序的临时变量使用，

使用方法与访问 PLC 的寄存器方式是一致的，也是通过地址的方式来引用的，软件总共提供了三类这样的存储器。

系统数据区(HSW)： 被系统定义的特殊寄存器；

数据存储区(HDW)： 用户存放用户数据；

特殊存储区(HPW)： 软件保留寄存器；

配方索引区(RPW)： 配方文件索引寄存器。

注意：HPW、HSW 的有些寄存器已经被软件所保留，每个字都可能自己的特殊用途，这种用途已为软件所保留。

HDW 用户一般可以随便当作临时变量使用。

## 27.2 系统数据区(HSW)

软件的系统数据区(HSW)用于保存系统数据，可以以两种方式来访问：

1. 以字的方式访问，前缀为 HSW，编号范围为：HSW0-HSW2047，这是以十进制的方式来访问的。比如：HSW0 表示系统数据区的第零个字，HSW1 表示系统数据区的第一个字。

2. 以位的方式来访问，前缀是 HSX，编号范围为：HSX0.0-HSX2047.15，”.”前面的编号是字编号(范围 0-2047)，而后面的是位编号(范围 0-15)。比如 HSX1020.12，其含义就是以位方式访问系统数据区，具体位置是第 1020 个字的第 12 位。

需要注意的是，HSW 和 HSX 的访问空间是重叠的；比如，HSX1020.12 是访问第 1020 个字的第 12 位，这个位的值与通过 HSW001020 访问得到一个字，这个字的第 12 位实际上与 HSX1020.12 是同一个位。

如果某个 HSW 已经被系统保留，则 HSX 访问该字某个位的值是没有任何意义的。

## 27.3 数据存储区(HDW)

软件的系统存储区(HDW)用于保存工程的暂时数据，可以以两种方式来访问：

1. 以字的方式访问，前缀为 HDW，编号范围为：HDW0-HDW8191，这是以十进制的方式来访问的。比如：HDW0 表示系统数据区的第 0 个字，HDW1 表示系统数据区的第一个字。

2. 以位的方式来访问，前缀是 HDX，编号范围为：HDX0.0-HDX8191.15，”.”前面的数表示字的编号，而后面的数字是该字的位编号，这两个编号都是十进制。比如 HDX1020.12，其含义就是以位方式访问系统数据区，具体位置是第 1020 个字的第 12 位。

需要注意的是，HDW 和 HDX 的访问空间是重叠的，也就是说 HDX1020.12 是访问第 1020 个字的第 12 位，通过 HDW1020 访问得到一个字，这个字的第 12 位与 HDX1020.12 是一致的。

该区是专门开放编程用的，软件一般不会在此区预设保留寄存器，用户可以放心使用。HDW8000 以上的地址为断电保持区。

## 27.4 特殊存储区(HPW)

软件的特殊存储区(HPW)用于保存系统的特殊数据，可以以两种方式来访问：

1. 以字的方式访问，前缀为 HPW，编号范围为：HPW0-HPW8191，这是以十进制的方式来访问的。比如：HPW0 表示系统数据区的第零个字，HPW1 表示系统数据区的第一个字。

2. 以位的方式来访问，前缀是 HPX，编号范围为：HPX0-HPX8191.15，”.”前面的数字表示字的

编号(范围:0-8191)，而后面的两位数是该字的位编号(范围：0-15)，这两个编号都是十进制。比如 HPX1020.12，其含义就是以位方式访问系统数据区，具体位置是第 1020 个字的第 12 位。

用户需要特别注意的是，HPW 和 HPX 的访问空间是重叠的，也就是说 HPX1020.12 是访问第 1020 个字的第 12 位，这个位的值与通过 HPW1020 访问得到一个字，这个字的第 12 位是一致的。

27.5 配方索引区 (RPW)

软件提供了一种灵活的方式来索引配方文件的配方数据，这就使得可以利用数值输入/显示部件、字开关来修改/显示配方文件的配方数据。

RPW 的用法如下：

RPW\*\*#####, 共有六个数字，前两个\*\*表示第几组号，####表示是第几个成分，比如：RPW010002 表示第一组配方的第 2 个成分。而 RPW110002 表示是第 11 组配方的第 2 个成分；如果被索引的组号或者成分号不存在，则 RPW 的值默认为零，比如 RPW110011 表示的是第 11 组配方的第 11 个成分，如果不存在，则访问 RPW110011 返回的值为零，而往 RPW110011 写值则不会写到文件里。

需要注意的是，只要 RPW\*\*####存在，则任何部件对 HPW\*\*#### “写”都会被保存到配方文件里。

RPW 没有提供位的访问方式，因为这没有任何意义。

从 RPW\*\*####的编码方法可知，RPW 的范围从 RPW000000-RPW501000，其中组号的范围是 0-49，而成份的范围是 0-1000。

27.6 保留寄存器

保留字地址	含义	用途
HPW0	HMI 配方组号寄存器	对于下载,当配方传输部件被用户触控时，软件将从 HPW000000 获得配方组号，然后将该组配方下载到终端。  对于上传,当配方传输部件被用户触控时，软件将从 HPW000000 获得配方组号,然后将 PLC 上传的配方放到该组所在的位置保存,该组原有的配方数据将被覆盖。

系统存储区 (HS) 保留的寄存器

保留字地址	含义	用途
HSW00000	保存参数设置	当 HSW000000 值为 1 时，系统将保存工程参数所做的修改
HSW000001	取消参数设置	取消所做的保存

		软件对 HSW0000000 和 HSW0000001 是互斥的
HSW0000002	HMI 站号	系统保留
HSW0000003	PLC 站号	系统保留
HSW0000004	COM1 的协议链接类型	0: RS232 1: RS422 2: RS485
HSW0000005	与 PLC 的通信端口号	0: COM1 1: COM2
HSW0000006	COM1 的波特率 (bps)	0: 2400 1: 4800 2: 9600 3: 19200 4: 38400 5: 57600 6: 115200
HSW0000007	COM1 的数据位	0: 7 1: 8
HSW0000008	COM1 的校验位	0: 无 1: 奇校验 2: 偶校验 3:SPACE
HSW0000009	COM1 的停止位	0: 1 Bit 1: 2 Bit
HSW0000010	COM1 的流控方式	0: 无流控 1:软流控 2:硬流控
HSW0000011	COM1 的连接失败重试次数	
HSW0000012	COM1 的串口等待超时	以 ms 为单位
HSW0000013	COM1 的串口接收数据超时	以 ms 为单位
HSW0000014	启动画面号	工程的初始画面号
HSW0000015	语言类型	在工程中设置不同的语言，可以



		显示不同的字体，HMI 支持 3 种语言， 0: 语言一 1: 语言二 2: 语言三
HSW000016	字体类型	保留
HSW000017	字体大小	保留
HSW000018	字体质量	保留
HSW000019	触摸蜂鸣	0: 蜂鸣 1: 不蜂鸣
HSW000021	LCD 是否反色显示	0: 正常显示 1: 反色显示
HSW000024	是否重起 HMI	0: 不启用 1: 启用
HSW000025	脚本是否要读透	0: 不用（不读透：读取缓存里的数据） 1: 启用（读透：直接读取 PLC 上传来的数据）
HSW000026	打印机类型	保留
HSW000027	打印端口	保留
HSW000028	本地时间年	有效值范围 0-9999（系统保留）
HSW000029	本地时间月	有效值范围 01-12（系统保留）
HSW000030	本地时间日	有效值范围 01-31（系统保留）
HSW000031	本地时间时	有效值范围 0-23（系统保留）
HSW000032	本地时间分	有效值范围 0-59（系统保留）
HSW000033	本地时间秒	有效值范围 0-59（系统保留）
HSW000034	上传程序	1: 上传（系统保留）
HSW000035	下载程序	1: 下载（系统保留）
HSW000036	COM2 的链接类型	0: RS232 1: RS422 2: RS485
HSW000037	COM2 的波特率	0: 2400 1: 4800 2: 9600 3: 9600

		4: 19200 5: 38400 6: 57600 7: 115200
HSW000038	COM2 的数据位	0: 7 1: 8
HSW000039	COM2 的校验位	0: 无 1: 奇校验 2: 偶校验 3: SPACE
HSW000040	COM2 的停止位	0: 1 Bit 1: 2 Bit
HSW000041	COM2 的流控方式	0: 无流控 1: 软流控 2: 硬流控
HSW000042	COM2 的失败连接重试次数	
HSW000043	COM2 的等待超时	以 ms 为单位
HSW000044	COM2 接收数据超时	以 ms 为单位
HSW000045	键盘字符串输入显示缓存区域	键盘画面输入缓存区域
HSW0000112	是否启用安全保护	当 HSW000112 值为 1 时，系统将启用安全等级的设置
HSW000113	默认安全等级	默认安全等级。当 HSW000113 值为以下值时含义： 1: 1 级 2: 2 级 3: 3 级
HSW000114	当前安全等级	当前安全等级。当 HSW000114 值为以下值时含义： 1: 1 级 2: 2 级 3: 3 级 4: 4 级 5: 5 级 6: 6 级

		7: 7 级 8: 8 级 9: 9 级 10: 10 级 11: 11 级 12: 12 级
HSW125	设置液晶显示对比度	
HSW126	星期	0: 星期日 1: 星期一 2: 星期二 3: 星期三 4: 星期四 5: 星期五 6: 星期六
HSW127	控制 BEEP 响铃	0: 正常响铃 1: 一直响铃 (断电不保存, 默认为 0)
HSW128	运行脚本单位周期值	
HSW129	运行脚本单位周期值是否为变量	
HSW130	监控地址 Timer 间隔单位周期值	
HSW131	监控地址 Timer 间隔单位周期值是否为变量	
HSW132	PLC 读写 cache 更新时间间隔	
HSW133	PLC 读写 cache 更新时间间隔是否为变量	
HSW136	HMI 本机的 IP 地址 1 (从左往右)	
HSW137	HMI 本机的 IP 地址 2 (从左往右)	
HSW138	HMI 本机的 IP 地址 3 (从左往右)	

HSW139	HMI 本机的 IP 地址 4(从左往右)	
HSW140	HMI 本机的子网掩 码地址 1(从左往右)	
HSW141	HMI 本机的子网掩 码地址 2(从左往右)	
HSW142	HMI 本机的子网掩 码地址 3(从左往右)	
HSW143	HMI 本机的子网掩 码地址 4(从左往右)	
HSW144	HMI 本机的本地网 关地址 1(从左往右)	
HSW145	HMI 本机的本地网 关地址 2(从左往右)	
HSW146	HMI 本机的本地网 关地址 3(从左往右)	
HSW147	HMI 本机的本地网 关地址 4(从左往右)	
HSW148	HMI 本地端口地址	
HSW000157	安全等级相互独立	0-不独立 1-独立
HSW000160	当前用户输入的密 码	预留 HSW000160~HSW000163, 共 4 个字, 最多输入 8 个字符密码
HSW000168	当前安全等级密码 往低切换的寄存器号	当前安全等级。当HSW000168值为 以下值时含义: 1: 1 级 2: 2 级
HSW000169	串口 1 重试超时	0~65535
HSW000170	串口 2 重试超时	0~65535
HSW000173	串口 1 等待 IO 延时 读写	0~65535
HSW000174	串口 2 等待 IO 延时 读写	0~65535
HSW000175	串口 3 等待 IO 延时 读写	0~65535

HSW000176	重画清空历史 XY 图	0: 不重画清空 1: 重画清空
HSW000189	是否显示错误消息框的寄存器	0: 不显示 1: 显示
HSW000191	清空记录数据文件,	0: 否 1: 是
HSW000192	清空系统设置文件,	0: 否 1: 是
HSW000193~194	键盘输入的最小值范围, 双字	0~0xFFFFFFFF
HSW000195~196	键盘输入的最大值范围, 双字	0~0xFFFFFFFF
HSW000200	当前位报警个数	
HSW000201	当前字报警个数	
HSW000202	清空位报警历史数据	1=是 0=否
HSW000203	清空字报警历史数据	1=是 0=否
HSW000205	清空\Flash\Data 分区下的所有文件, 断电重启后人机即可查询数据。	0: 否 1: 是
HSW214	关闭弹出式窗口 (如报警窗口)	1 时关闭
HSW000216	键盘大小写切换	0: 大写 1: 小写
HSW000217	删除多国语言的系统设置, 恢复为默认的语言	0: 否 1: 是
HSW000218	设置屏保的开关	0: 否 1: 是
HSW000219	设置屏保的倒计时起始时间	以秒为单位
HSW000220	设置屏保的倒计时时间	以秒为单位

HSW000221	设置关背景灯时间	以秒为单位
HSW000223	设置是否报警时响铃	0-响铃 1-不响铃
HSW000224	PROT2口 PLC 站号	0~255
HSW000225	端口号2	1~3
HSW000226	PROT3口 PLC 站号	0~255
HSW000227	端口号3	1~3
HSW000230	CF/SD 卡数据转存到 U 盘	0: 否 1: 是
HSW000233	不通过 Cache 访问 PLC 等设备	0: 通过 1: 不通过
HSW000235	Flash 下数据记录文件拷到 U 盘	0: 否 1: 是
HSW000238	位报警标志位	值为 1 表示发生了位报警
HSW000239	字报警标志位	值为 1 表示发生了字报警
HSW000243	当前离散编辑的配方组号	1~3
HSW000244	配方上传下载	1=上传 2=下载
HSW000245	清除报警记录	清除报警记录
HSW000247	COM3 的链接类型	0: RS232 1: RS422 2: RS485
HSW000248	COM3 的波特率	0: 2400 1: 4800 2: 9600 3: 9600 4: 19200 5: 38400 6: 57600 7: 115200
HSW000249	COM3 的数据位	0: 7 1: 8
HSW000250	COM3 的校验位	0: 无 1: 奇校验

		2: 偶校验 3: SPACE
HSW000251	COM3 的停止位	0: 1 Bit 1: 2 Bit
HSW000252	COM3 的流控方式	0: 无流控 1: 软流控 2: 硬流控
HSW000253	COM3 的失败连接重试次数	
HSW000254	COM3 的等待超时	以 ms 为单位
HSW000255	COM3 接收数据超时	以 ms 为单位
HSW000256	第零组数据记录参数设置 是否不保存记录 该设置可以掉电保存	0: 保存 1: 不保存
HSW000257	第零组数据记录参数设置 数据采样的间隔 该设置可以掉电保存	秒为单位 0~65535
HSW000258	第零组数据记录参数设置 保存间隔 该设置可以掉电保存	分钟为单位 0~65535
HSW000259	第零组数据记录参数设置 立刻存储数据记录 该设置可以掉电保存	0: 根据工程的设置时间保存 1: 立刻存储至 CF/SD/SD 卡内, 系统将自动清零
HSW000260	第一组数据记录参数设置 是否不保存记录 该设置可以掉电保存	0: 保存 1: 不保存

HSW000261	第一组数据记录参数设置 数据采样的间隔 该设置可以掉电保存	秒为单位 0~65535
HSW000262	第一组数据记录参数设置 保存间隔 该设置可以掉电保存	分钟为单位 0~65535
HSW000263	第一组数据记录参数设置 立刻存储数据记录 该设置可以掉电保存	0: 根据工程的设置时间保存 1: 立刻存储至 CF/SD/SD 卡内, 系统将自动清零
...	...	...
HSW000296	第十组数据记录参数设置 是否不保存记录 该设置可以掉电保存	0: 保存 1: 不保存
HSW000297	第十组数据记录参数设置 数据采样的间隔 该设置可以掉电保存	秒为单位 0~65535
HSW000298	第十组数据记录参数设置 保存间隔 该设置可以掉电保存	分钟为单位 0~65535
HSW501	默认部件密码等级	
HSW502	当前部件密码等级	
HSW503	文件列表配方导入组号	
HSW504	文件列表配方导出	



	组号	
HDW25000	文件列表配方导入 文件名称	
HDW25001	文件列表配方导出 文件名称	
HSW506	清空当前事件显示 的链表	
HSW507	清空历史事件显示 的链表	
HSW508	删除 EVReg. dat 文 件	
HSW509	部件等级密码相互 独立	
HSW500	是否启用部件密码	
HSW543 --- HSW544	用来标记触摸屏被 点击的时候 被点击的 X 坐标的值	
HSW544	用来标记触摸屏被 点击的时候 被点击的 Y 坐标的值	
HSW546	打印部件保存图片 时候用来标志 U 盘是否 加载成功	HSW546=1
HSW547	打印部件保存图片 时候用来标志 U 图片是 否保存成功	HSW547=1 表示正在保存 HSW547=2 表示保存成功
HSW545	屏保画面号	
HSW000299	第十组数据记录参 数设置 立刻存储数据记录 该设置可以掉电保 存	0: 根据工程的设置时间保存 1: 立刻存储至 CF/SD/SD 卡内, 系统将自动清零
HSW00744~HSW001255	掉电保存区	开放给用户使用的掉电保存区, 共大小为 512 个字, 用户可以将需要 掉电保存的数据放置在这里。 由于 HMI 是采用 FLASH 来保存数

		据，强烈建议不要过于频繁地写这个区段,过于频繁地写操作会影响 FLASH 的寿命
HSW542	等于 1 时	内存已满
HDW8000~HDW30000	掉电保存区	<p>新增掉电保护区开放给用户使用的掉电保存区，共大小为 16K+2000 个字，用户可以将需要掉电保存的数据放置在这里。</p> <p>由于 HMI 是采用 FLASH 来保存数据，强烈建议不要过于频繁地写这个区段,过于频繁地写操作会影响 FLASH 的寿命</p>

画面等级密码

HSW000404	安全等级 1 密码	共 8 个字符
HSW000408	安全等级 2 密码	共 8 个字符
HSW000412	安全等级 3 密码	共 8 个字符
HSW000416	安全等级 4 密码	共 8 个字符
HSW000420	安全等级 5 密码	共 8 个字符
HSW000424	安全等级 6 密码	共 8 个字符
HSW000428	安全等级 7 密码	共 8 个字符
HSW000432	安全等级 8 密码	共 8 个字符
HSW000436	安全等级 9 密码	共 8 个字符
HSW000440	安全等级 10 密码	共 8 个字符
HSW000444	安全等级 11 密码	共 8 个字符
HSW000448	安全等级 12 密码	共 8 个字符

部件等级密码

部件密码使用内部地址字地址类型 HSW		
等级一	共 8 个字符	HSW452
等级二	共 8 个字符	HSW456
等级三	共 8 个字符	HSW460
等级四	共 8 个字符	HSW464

等级五	共 8 个字符	HSW468
等级六	共 8 个字符	HSW472
等级七	共 8 个字符	HSW476
等级八	共 8 个字符	HSW480
等级九	共 8 个字符	HSW484
等级十	共 8 个字符	HSW488
等级十一	共 8 个字符	HSW492
等级十二	共 8 个字符	HSW496
是否启用部件等级密码	HSW500	
默认部件密码等级	HSW501	
当前部件密码等级	HSW502	
本地 IP 地址 高字 1	HSW136	
本地 IP 地址 高字 2	HSW137	
本地 IP 地址 高字 3	HSW138	
本地 IP 地址 高字 4	HSW139	
本地掩码 IP 地址 高字 1	HSW140	
本地掩码 IP 地址 高字 2	HSW141	
本地掩码 IP 地址 高字 3	HSW142	
本地掩码 IP 地址 高字 4	HSW143	
本地网关 IP 地址 高字 1	HSW144	
本地网关 IP 地址 高字 2	HSW145	
本地网关 IP 地址 高字 3	HSW146	
本地网关 IP 地址 高字 4	HSW147	
本地端口地址	HSW148	

## 28 串行通讯与 PLC 驱动程序

RS-232、RS-422 与 RS-485 都是串行数据接口标准，最初都是由电子工业协会（EIA）制订并发布的，RS-232 在 1962 年发布，命名为 EIA-232-E，作为工业标准，以保证不同厂家产品之间的兼容。RS-422 由 RS-232 发展而来，它是为弥补 RS-232 之不足而提出的。为改进 RS-232 通信距离短、速率低的缺点，RS-422 定义了一种平衡通信接口，将传输速率提高到 10Mb/s，传输距离延长到 4000 英尺（速率低于 100kb/s 时），并允许在一条平衡总线上连接最多 10 个接收器。RS-422 是一种单机发送、多机接收的单向、平衡传输规范，被命名为 TIA/EIA-422-A 标准。为扩展应用范围，EIA 又于 1983 年在 RS-422 基础上制定了 RS-485 标准，增加了多点、双向通信能力，即允许多个发送器连接到同一条总线上，同时增加了发送器的驱动能力和冲突保护特性，扩展了总线共模范围，后命名为 TIA/EIA-485-A 标准。由于 EIA 提出的建议标准都是以“RS”作为前缀，所以在通讯工业领域，仍然习惯将上述标准以 RS 作前缀称谓。

RS-232、RS-422 与 RS-485 标准只对接口的电气特性做出规定，而不涉及接插件、电缆或协议，在此基础上用户可以建立自己的高层通信协议。因此在视频界的应用，许多厂家都建立了一套高层通信协议，或公开或厂家独家使用。如录像机厂家中的 Sony 与松下对录像机的 RS-422 控制协议是有差异的，

视频服务器上的控制协议则更多了，如 Louth、Odetis 协议是公开的，而 ProLINK 则是基于 Profile 上的。

## 28.1 RS-422 与 RS-485 串行接口标准

RS-422、RS-485 与 RS-232 不一样，数据信号采用差分传输方式，也称作平衡传输，它使用一对双绞线，将其中一线定义为 A，另一线定义为 B，如图 28-1。

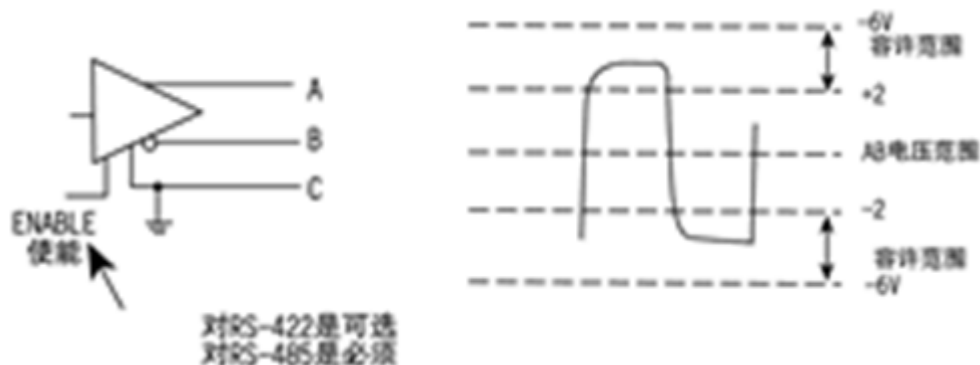


图 28-1 RS-422/RS-485 线定义

通常情况下，发送驱动器 A、B 之间的正电平在 +2~+6V，是一个逻辑状态，负电平在 -2~6V，是另一个逻辑状态。另有一个信号地 C，在 RS-485 中还有一“使能”端，而在 RS-422 中这是可用可不用的。“使能”端是用于控制发送驱动器与传输线的切断与连接。当“使能”端起作用时，发送驱动器处于高阻状态，称作“第三态”，即它是有别于逻辑“1”与“0”的第三态。

接收器也作与发送端相对的规定，收、发端通过平衡双绞线将 AA 与 BB 对应相连，当在收端 AB 之间有大于 +200mV 的电平时，输出正逻辑电平，小于 -200mV 时，输出负逻辑电平。接收器接收平衡线上的电平范围通常在 200mV 至 6V 之间。参见图 28-2。

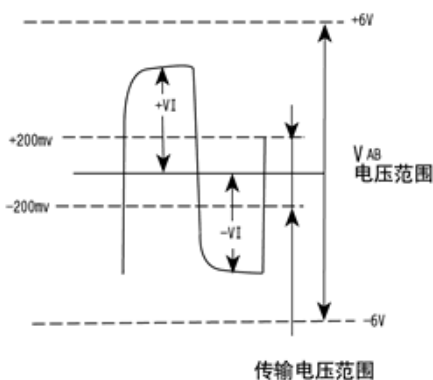


图 28-2 传输电压范围

端 AB 之间有大于 +200mV 的电平时，输出正逻辑电平，小于 -200mV 时，输出负逻辑电平。接收器接收平衡线上的电平范围通常在 200mV 至 6V 之间。

## 28.2 RS-232 串行接口标准

目前 RS-232 是 PC 机与通信工业中应用最广泛的一种串行接口。RS-232 被定义为一种在低速率串

行通讯中增加通讯距离的单端标准。RS-232 采取不平衡传输方式，即所谓单端通讯。

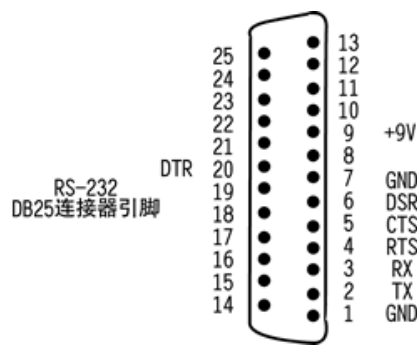


图 28-3 RS-232 DB25 连接器引脚

收、发端的数据信号是相对于信号地，如从 DTE 设备发出的数据在使用 DB25 连接器时是 2 脚相对 7 脚（信号地）的电平，DB25 各引脚定义参见图 22-1。典型的 RS-232 信号在正负电平之间摆动，在发送数据时，发送端驱动器输出正电平在+5~+15V，负电平在-5~-15V 电平。当无数据传输时，线上为 TTL，从开始传送数据到结束，线上电平从 TTL 电平到 RS-232 电平再返回 TTL 电平。接收器典型的工作电平在+3~+12V 与-3~-12V。由于发送电平与接收电平的差仅为 2V 至 3V 左右，所以其共模抑制能力差，再加上双绞线上的分布电容，其传送距离最大为约 15 米，最高速率为 20kb/s。RS-232 是为点对点（即只用一对收、发设备）通讯而设计的，其驱动器负载为 3~7kΩ。所以 RS-232 适合本地设备之间的通信。其有关电气参数参见表 28-1。

表 28-1 RS232、RS422、RS485 电气特性表

规定		RS232	RS422	R485
工作方式		单端	差分	差分
节点数		1 收、1 发	1 发 10 收	1 发 32 收
最大传输电缆长度		50 英尺	4000 英尺	4000 英尺
最大传输速率		20Kb/S	10Mb/s	10Mb/s
最大驱动输出电压		+/-25V	-0.25V ~ +6V	-7V~+12V
驱动器输出信号电平 (负载最小值)	负载	+/-5V~+/-15V	+/-2.0V	+/-1.5V
驱动器输出信号电平 (空载最大值)	空载	+/-25V	+/-6V	+/-6V
驱动器负载阻抗(Ω)		3K~7K	100	54
摆率(最大值)		30V/μ s	N/A	N/A
接收器输入电压范围		+/-15V	-10V ~ +10V	-7V~+12V

接收器输入门限	+/-3V	+/-200mV	+/-200mV
接收器输入电阻( $\Omega$ )	3K~7K	4K(最小)	$\geq 12K$
驱动器共模电压		-3V~+3V	-1V~+3V
接收器共模电压		-7V~+7V	-7V~+12V

28.3 RS-422 电气规定

RS-422 标准全称是“平衡电压数字接口电路的电气特性”，它定义了接口电路的特性。图 20-4 中的图 B 是典型的 RS-422 四线接口。实际上还有一根信号地线，共 5 根线。图 20-4 中的图 A 是其 DB9 连接器引脚定义。由于接收器采用高输入阻抗和发送驱动器比 RS232 更强的驱动能力，故允许在相同传输线上连接多个接收节点，最多可接 10 个节点。即一个主设备（Master），其余为从设备（Salve），从设备之间不能通信，所以 RS-422 支持点对多的双向通信。接收器输入阻抗为 4k，故发端最大负载能力是  $10 \times 4k + 100\Omega$ （终接电阻）。RS-422 四线接口由于采用单独的发送和接收通道，因此不必控制数据方向，各装置之间任何必须的信号交换均可以按软件方式（XON/XOFF 握手）或硬件方式（一对单独的双绞线）。

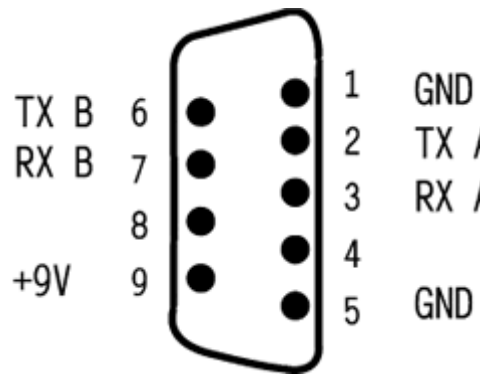


图 A

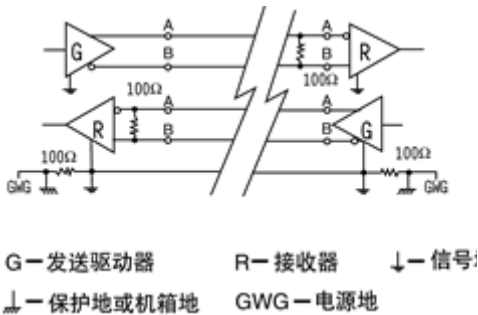


图 B

图 28-4 RS-422 电气规定

RS-422 的最大传输距离为 4000 英尺（约 1219 米），最大传输速率为 10Mb/s。其平衡双绞线的长度与传输速率成反比，在 100kb/s 速率以下，才可能达到最大传输距离。只有在很短的距离下才能获得最高速率传输。一般 100 米长的双绞线上所能获得的最大传输速率仅为 1Mb/s。

RS-422 需要一终接电阻，要求其阻值约等于传输电缆的特性阻抗。在短距离传输时可不需终接电阻，即一般在 300 米以下不需终接电阻。终接电阻接在传输电缆的最远端。

RS-422 有关电气参数见 RS232、RS422、RS485 电气特性表

28.4 RS-485 电气规定

由于 RS-485 是从 RS-422 基础上发展而来的，所以 RS-485 许多电气规定与 RS-422 相仿。如都采用平衡传输方式、都需要在传输线上接终接电阻等。RS-485 可以采用二线与四线方式，二线制可实现真正的多点双向通信，参见图 20-5 中的图 C。

而采用四线连接时，与 RS-422 一样只能实现点对多的通信，即只能有一个主（Master）设备，

其余为从设备，但它比 RS-422 有改进，无论四线还是二线连接方式总线上可多接到 32 个设备。参见图 28-5 中的图 D。

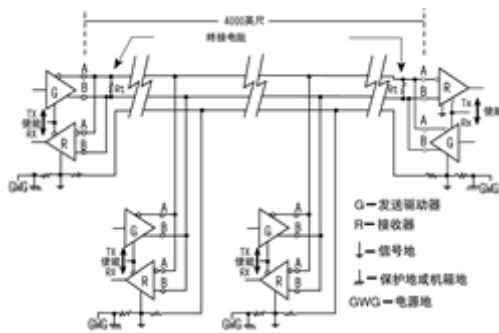


图 C

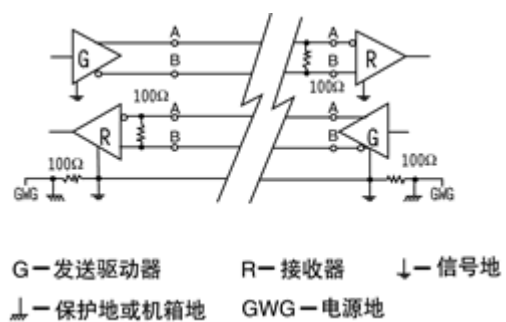


图 D

图 28-5 RS-485 电气规定

RS-485 与 RS-422 的不同还在于其共模输出电压是不同的，RS-485 是 -7V 至 +12V 之间，而 RS-422 在 -7V 至 +7V 之间，RS-485 接收器最小输入阻抗为 12k 而 RS-422 是 4k；因为 RS-485 满足所有 RS-422 的规范，所以 RS-485 的驱动器可以用在 RS-422 网络中应用。

RS-485 有关电气规定参见 RS232、RS422、RS485 电气特性表。

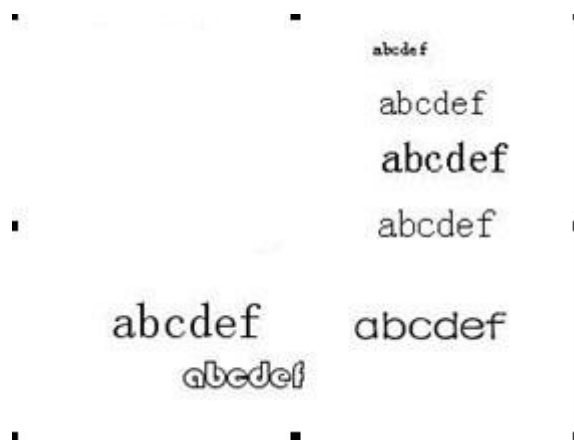
RS-485 与 RS-422 一样，其最大传输距离约为 1219 米，最大传输速率为 10Mb/s。平衡双绞线的长度与传输速率成反比，在 100kb/s 速率以下，才可能使用规定最长的电缆长度。只有在很短的距离下才能获得最高速率传输。一般 100 米长双绞线最大传输速率仅为 1Mb/s。

RS-485 需要 2 个终接电阻，其阻值要求等于传输电缆的特性阻抗。在短距离传输时可不需终接电阻，即一般在 300 米以下不需终接电阻。终接电阻接在传输总线的两端。



## 29 字体库

软件支持所有的 Windows 矢量字体，矢量字体可以增加画面的表现力度。用户根据自己的需要可以自行定义各种矢量字体，软件最多可以支持多达 32 种自定义字体。



从主菜单栏**设置**中选择**字体库**，出现如下图所示得对话框：



图 29-1 自定义字体库

**列表框：**列举当前已经建立的自定义字体库，软件内置得字体不在这里列举

**字体名称：**当前选中的矢量字体名称

**大小：**当前选中的字体大小

**添加字体：**添加新的自定义字体，软件最多支持 32 种自定义字体

**删除字体：**删除所选中的字体

**编辑属性：**编辑所选中的字体属性

**关闭：**关闭对话框

**范例：**选中字体的实际效果示范。

29.1 自定义字体的使用

下列部件可以使用自定义字体：字开关、位开关、功能开关、字状态显示、位状态显示、移动图形、轨迹图形、静态文本、四指示灯、按键。

有下面两种方式使用自定义字体：

从部件的属性框选择字体属性，如图 29-2 所示

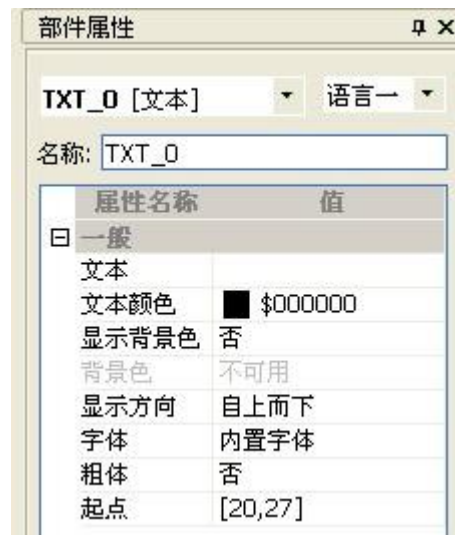


图 29-2 属性框里改变部件标签字体

弹出下面的对话框：



图 29-3 选择字体

在图 29-3 中，排在前面的字体是内置字体，后面的是自定义字体，选中您所要得字体，然后点击**选择**。

还有一种办法是从属性编辑工具条，为当前部件选择所需要得字体。



图 29-4 从工具条选择字体

## 29.2 添加字体

点击图 29-5 的添加字体，出现下面的对话框：



图 29-5 添加字体

**字体名称:** 字体得自定义名称，不可重复。

**字体:** Windows 系统所支持的矢量字体。

**样式:** Regular:格则, Bold: 粗体, Italic: 斜体, Bold Italic: 粗斜体

**大小:** 自定义字体的大小

**删除线:** 文字中间添加删除线

**下划线:** 文件下面天加下划线

**范例:** 字体的预览效果

## 30 数据记录

只要配备了大容量存储，就进行数据记录，把记录的数据保存在 CF 卡中，记录时间的长度取决于 CF 卡容量、记录内容和采样间隔。以 1G 容量的 CF 卡为例：可存储的天数为：

- (1) 4 路数据：约 1400 天
- (2) 6 路数据：约 1100 天
- (3) 8 路数据：约 900 天
- (4) 12 路数据：约 650 天
- (5) 16 路数据：约 500 天
- (6) 32 路数据：约 250 天

记录到 CF 卡后的数据，用户可以通过网络或者串口上传到 PC 机，也可以直接拔下 CF 卡，通过 CF 卡读写器导出数据。

软件记录文件的格式是 CSV 格式，是很通用的数据存储格式，用户可以将这种格式的数据导入到数

数据库或者 EXCEL 电子表格进行数据加工或保存。

30.1 通过读卡器导出数据

将 CF 卡从 HMI 中拔出(强烈建议在拔插 CF 卡时关闭 HMI 电源)，HMI 在 CF 卡中保存的数据是通过目录组织的。所以，只有熟悉了里面的目录结构才能找到相应的文件。

报警数据的组织。

AL(报警数据所在目录) 0 是位报警数据所在目录，1 是字报警数据所在目录 200701，200702 这些目录分别表示 2007 年 1 月的数据、2007 年 2 月的数据 01，02 目录那一天的数据 里面的文件是某个时间段所记录的数据文件。

比如，要找 2006 年 10 月 23 日的字报警数据。您首先要从 CF 卡找到 AL 目录，然后在 AL 中找到目录名为“1”的子目录，打开这个目录，找到目录名为“200610”这个目录。然后再找到“23”这个目录。这个目录保存的就是 2006 年 10 月 23 日的所有字报警记录数据。

数据文件是按照时间段的来命名的，比如 12-0.csv 表示是 12 点的数据文件。

记录数据的组织

AL(记录数据所在目录) 子目录名为组的编号 200701，200702 这些目录分别表示 2007 年 1 月的数据、2007 年 2 月的数据 01，02 目录那一天的数据 里面的文件是某个时间段所记录的数据文件。

比如，要找 2006 年 10 月 23 日的字报警数据。您首先要从 CF 卡找到 DL 目录，然后在 DL 中找到目录名为记录组序号的子目录，打开这个目录，找到目录名为“200610”这个目录。然后再找到“23”这个目录。这个目录保存的就是 2006 年 10 月 23 日的所有记录数据。

数据文件是按照时间段的来命名的，比如 12-0.csv 表示是 12 点的数据文件。]

30.2 数据记录功能

数据记录功能是按照下面得结构组织的。在一个工程中，可以有多个记录组，每个记录组包含有多个通道，不同的组，其采样时间是不一样的，其保存时间也不一样。

拓扑图如下图所示：

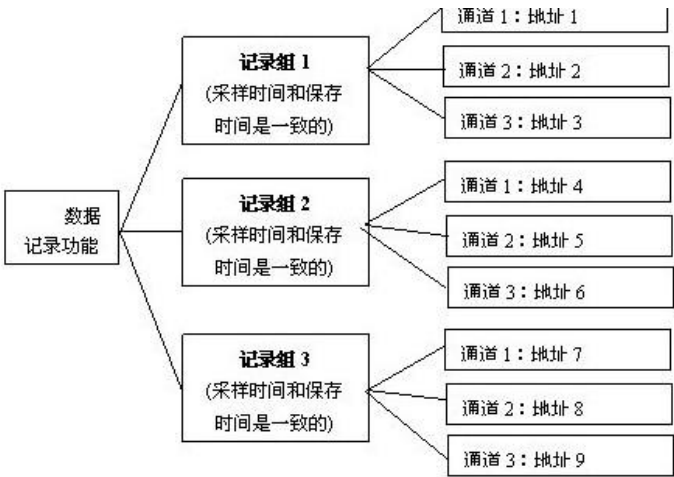


图 30-1 数据记录的拓扑结构

记录数据在 CF 卡上的保存也是按照上述结构来组织的。

同一个记录组，它的采样时间与保存时间是一致的，组内会包含有多个通道，采样时间一到，每个通道都会按照自己的地址到 PLC 寄存器采集数据。

采样间隔：记录组连续采样两次的時間间隔，这个间隔是个概数，记录内容越少，将会越准确。

保存间隔：采样得到数据会先保存在内存里，每过一段时间(就是保存时间)，系统会把内存中的采样数据保存到 CF 卡中，如果此时系统断电，那么人机将不会保存尚在内存中的数据。

只要工程使用了数据记录功能，并且是支持大容量存储模块，那么系统将按照设定的采样间隔采样数据，按照设定的保存间隔保存数据。

30.3 数据记录显示

记录在 CF 卡中的数据也可以在 HMI 上查看，只要放置一个数据记录显示部件就可以了。

数据记录显示离线模拟的情况如下图所示，点击图标 S，弹出查询设置对话框。



图 30-2 数据记录显示

数据记录显示以表格的方式描绘出复合查询条件的数据。每个数据记录显示部件只能显示一个记录组的数据，



图 30-3 查询条件设置

30.4 记录数据打印

上传后的数据可以通过下面两种方式打印：

1、用 EXCEL 电子表格打开 CSV 格式的数据文件，有时时间那一列会以“####”的方式显示，这不是没有时间，您只要在 EXCEL 里把时间所在列的宽度放大就可以显示时间，然后，从菜单中选择“文件”“打印”，即可打印数据文件。

2、用 Windows 系统自带的记事本程序打开 CSV 格式的数据文件，然后从菜单中选择“文件”“打印”，完成数据文件的打印。

两者打印出来的数据是一样，只是前者有表格线，更美观；后者没有表格线，更紧凑。

30.5 数据记录区

工程需要配置好数据记录区才能使用数据记录功能。

从设置->数据记录区，打开图 30-4 所示的对话框。

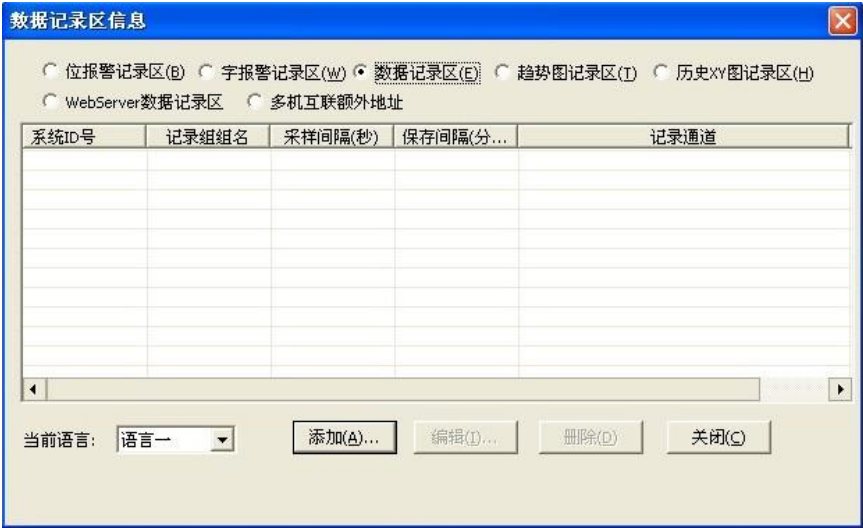


图 30-4 数据记录区

图 30-4 所示的对话框枚举了当前工程的记录组情况。

**添加：**添加一个记录组

**编辑：**编辑选中的记录组

**删除：**删除当前的记录组

**关闭：**关闭对话框

点击**添加**，然后进入如图 30-5 所示的对话框。

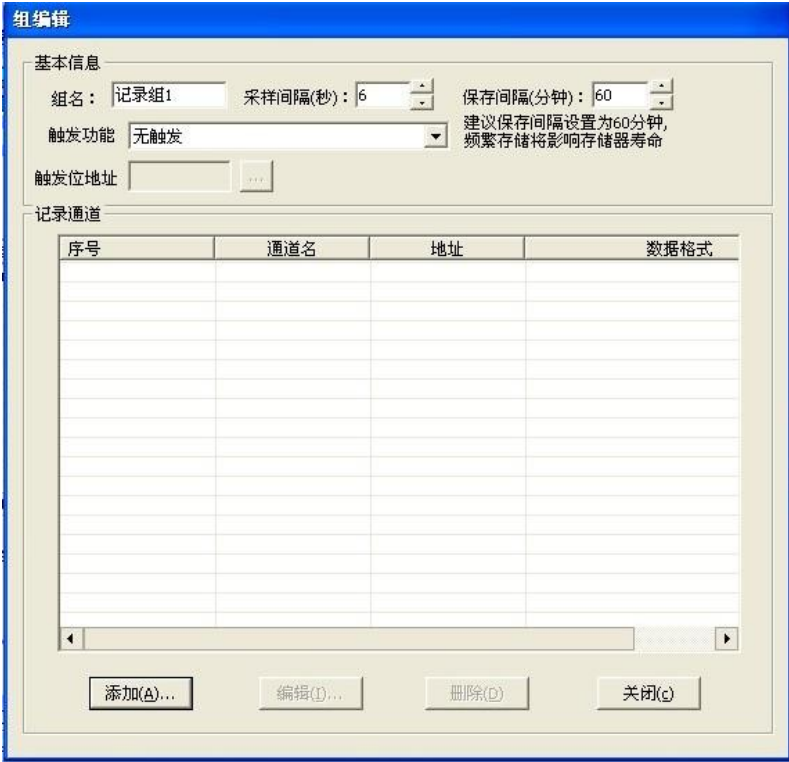


图 30-5 组编辑对话框

**组名：**记录组的名称，该名称是唯一的，不能重名。

**采样间隔：**该记录组的采样间隔，以秒为单位。

**保存间隔：**该记录组连续两次做保存操作(将数据从内存写到 CF 卡)的时间间隔，以分钟为单位

**触发功能：**无触发、触发记录（按时间采样），触发记录并自动清零触发位（记录一条），触发记录并不清零（记录一条）

**触发位地址：**触发的操作

**确定：**确定上述更改。

**记录通道：**列举该记录组有多少记录通道。

**添加：**添加一个记录通道。

**编辑：**编辑选中的记录通道。

**删除：**删除选中的记录通道

**关闭：**关闭对话框

每个记录组的采样间隔、保存间隔、数据记录文件的处理，在 HMI 上都可以动态修改，并且修改后可以掉电保存，只要通过部件对软件保留的寄存器进行操作就可以实现，这些寄存器在**保留寄存器**一节中有详细描述

在图 30-6 所示的组编辑对话框，点击添加，进入通道编辑：





图 30-6 通道编辑对话框

**通道名称:** 通道的名称, 该名称在记录组内是唯一的, 不能重复。

**地址:** 记录组要监视的字地址, 可以是 PLC 寄存器, 也可以是 HMI 寄存器

**显示格式:** 按照什么格式来转换数据, 目前支持无符号十进制、有符号十进制、二进制、八进制、16 位浮点 BCD。

**小数点位置:** 数据保存在 CF 卡中时, 是否要显示小数点, 以及小数点的格式。

### 30.6 数据上传

软件记录数据可以上传到 PC 上来进行处理, 利用数据记录上传 (DataUpload) 便可以将数据从 HMI 传至 PC, 文件格式是 CSV。软件运行后, 出现如下图所示的对话框。

将 HMI 的 COM2 口与指定的 PC 通信口连接, 然后运行 DataUpload, 出现下面对话框。



图 30-7 通讯口设置

**串口:** 选择 PC 上的通信口

**波特率:** 通信波特率

**报警数据:** 上传 HMI 上的报警历史数据

**记录数据:** 上传 HMI 上的历史记录数据

点击图 30-8 所示的确定, 然后进入如图 22-7 所示的对话框:

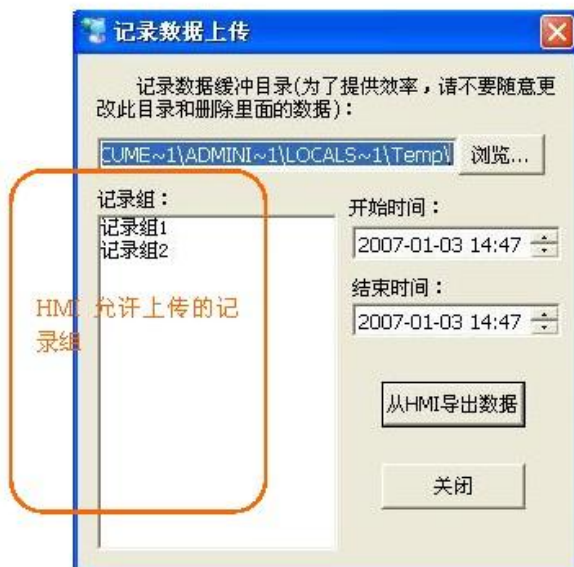


图 30-8. 数据上传对话框

**缓冲目录:** 上传数据的过程中, 需要指定一个缓冲目录来缓冲数据

**记录组:** HMI 现有的记录组, 上传时必须指定一个记录组, 一次只能上传一个记录组

**开始时间、结束时间:** 选择要数据上传得数据范围

**从 HMI 导出数据:** 根据当前选择从 HMI 上传数据

**关闭:** 取消, 不做数据上传。

## 31 穿透

### 31.1 应用方案 1

需要用的“通讯助手”软件进行拨号。拨号完成后，在 PC 机上运行 PLC 的编辑软件（如三菱的 GX Develop），可如同本地操作一样操作远程 PLC。

如果远程机器出现故障，可将 PLC 里的程序及数据读出来，第一时间拿到现场信息，为您的快速反应建立良好的基础，同时也节省了您的维护费用。

使用该方案，可以随时远程下载 PLC 的程序，也可以调试新的程序，方便您的后期维护。

### 31.2 应用方案 2

在本地 PC 机上运行的“通讯助手”软件，拨号连接到远程的 HMI，或用以太网连接到远程 HMI。

之后，你可以使用“通讯助手”软件，随时监控或修改远程变频器或仪表，更好的获取、分析、修改现场信息，方便您的后期维护。

如果 HMI 的画面或功能需要升级，您也可以使用“通讯助手”软件进行升级。

## 32 远程穿透

32.1 远程穿透系统架构

系列 HMI 提供的远程穿透配置简单，功能实用，其应用架构如下：

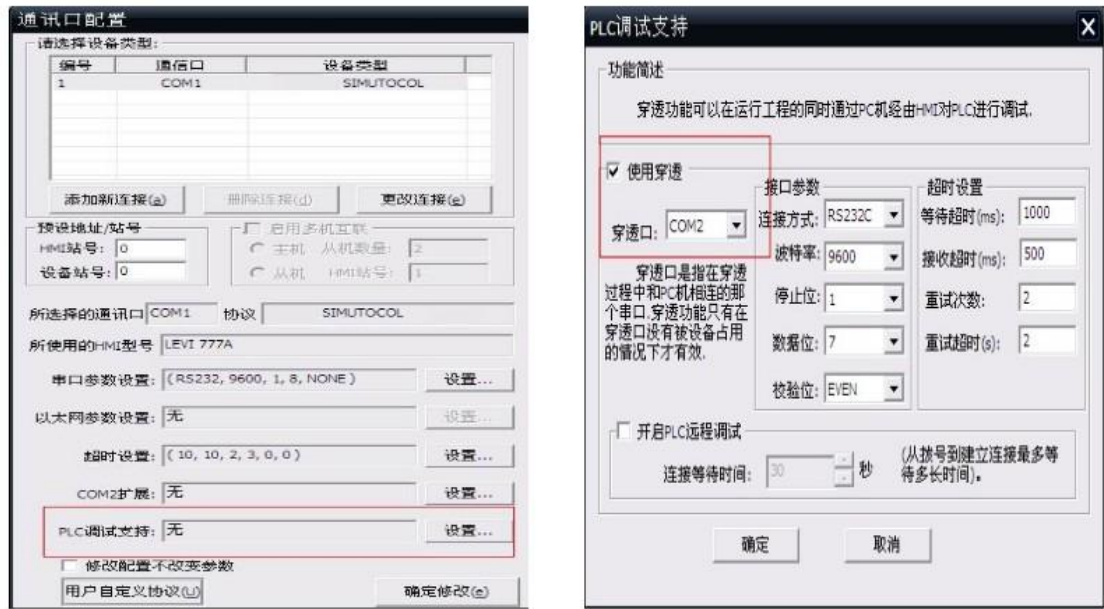


在该应用中，系列 HMI 处于整个系统的中心位置，上位机通过拨号网络、无线网络、以太网等网络形式与 HMI 连接，并通过系列 HMI 与 PLC、变频器、仪表等设备相连接。如此，上位机可以随时与远程的 PLC、变频器、仪表等设备进行通信、调试，效果和在本本地调试一样。这样有效的减少了维护成本与时间。

32.2 远程穿透系统架构

参数设置

如要使用软件 HMI 的远程穿透功能，需要在软件左侧的工程管理器中“工程配置”的“通讯口配置”进行配置（出现如下左图对话框）。点击“plc 调试支持”的“设置”，出现如下右图对话框。



选中“plc 调试支持”后，需要对“穿透口”及口的设置、超时设置进行合适的配置。如果要使用拨号网络进行远程连接，请选中“开启 PLC 远程调试”。

注意：接口参数不正确，会导致不能通信；超时设置过长或过短，都有可能导致通信不正常。

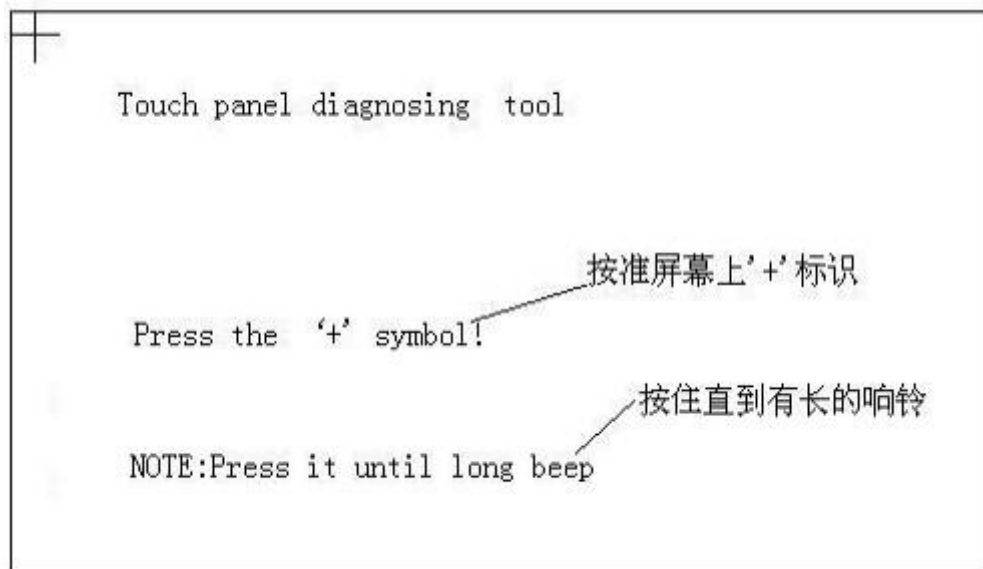
如果您不知道如何配置，请联系本公司的技术部门，我们将根据您的具体情况，为您配置合适的设置。

在初次使用的界面时，客户经常会遇到以下几个方面的问题：

- (1) 触摸屏按键位置不准确，触摸屏的测试；
- (2) 升级更新工程和内核程序；
- (3) 液晶屏的对比度和亮度调节；
- (4) CF 卡纪录数据的上传；
- (5) HMI 系统参数的修改；
- (6) HMI 的工厂测试；

(1) 触摸屏按键位置不准确，该怎么办？

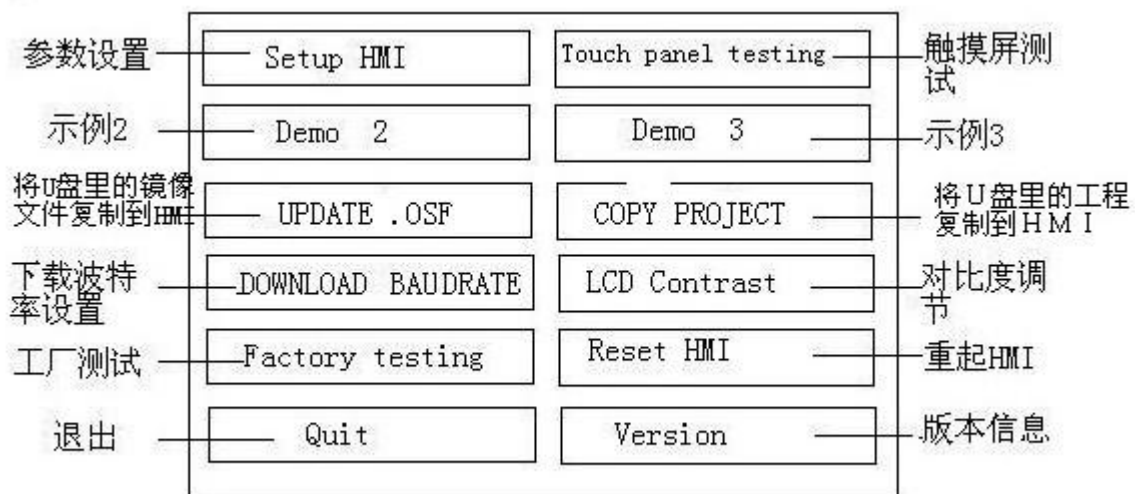
解决：按住屏幕的左上角大约 3~5 秒钟，出现如下界面



依次点中左上角，左下角，右下角，右上角，中心点的 '+'，重新启动即可。

(2) 怎样进入触摸屏的测试画面？

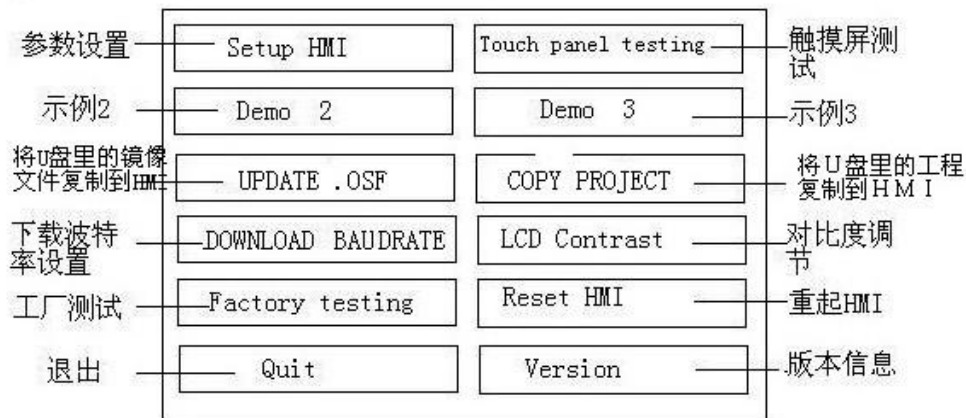
解决：按住屏幕的右上角大约 3~5 秒钟，出现如下界面



选择“Touch panel testing”，进入触摸屏的测试界面；

### (3) 需要升级更新工程和内核程序，要怎么做？

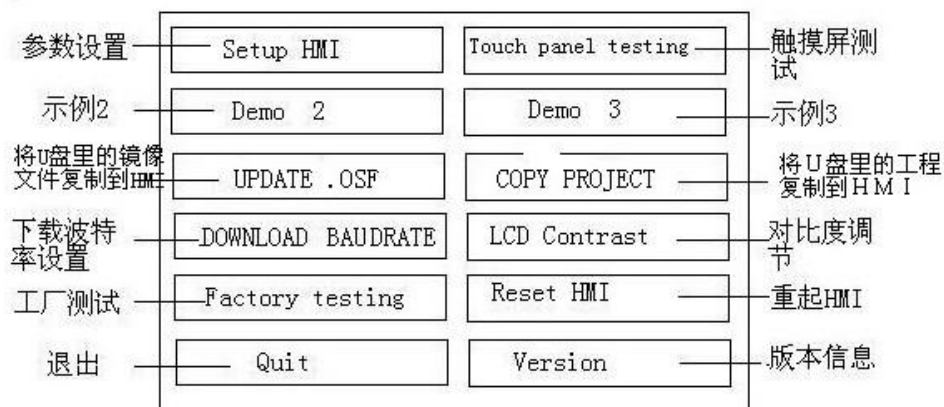
解决：按住屏幕的右上角或右下角大约 3~5 秒钟，出现如下界面：



然后，再选择相应的工程或内核文件下载升级；

### (4) 如何调节液晶屏的对比度？

解决：按住屏幕的右上角大约 3~5 秒钟，出现如下界面



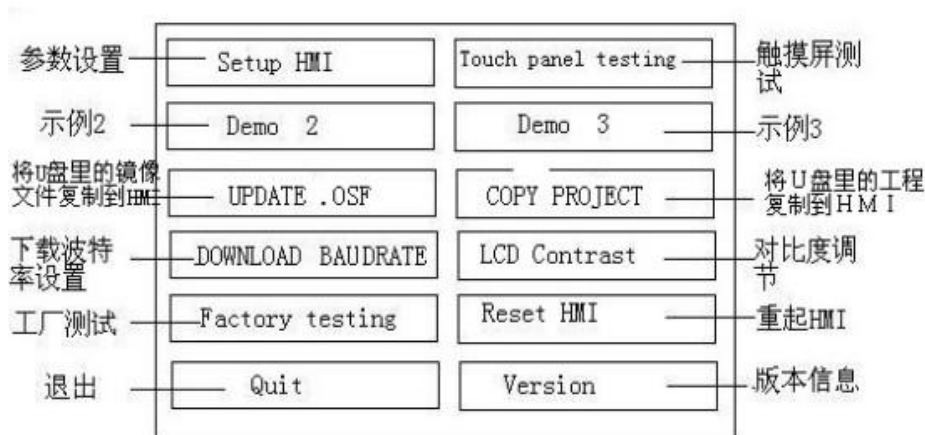
再点中“LCD Contrast”对比度调节，进入设置画面调节；

### (5) 如何调节液晶屏的亮度？

解决：在屏的背面，电源接线柱上方大约 5cm 的位置，有个小孔，找一把十字的小螺丝刀，插进去进行微调，即可；

### (6) 需要修改 HMI 系统参数，该怎么做？

解决：按住屏幕的右上角大约 3~5 秒钟，出现如下界面：

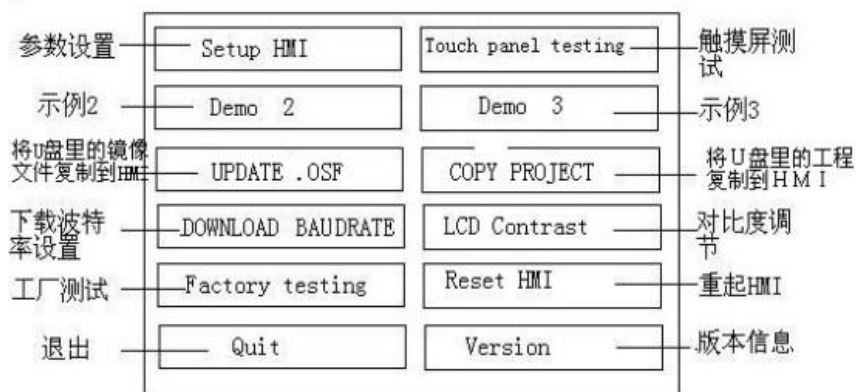


选择“Setup HMI”，进入触摸屏的参数设置界面；

**注意：如果不熟悉操作的客户，尽量不要随便修改，否则后果自负。**

### (7) 如何进入 HMI 的工厂测试？

解决：按住屏幕的右上角大约 3~5 秒钟，出现如下界面：



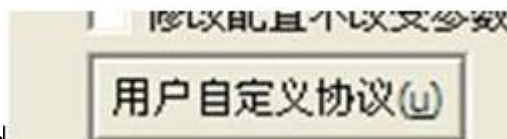
选择“Factory testing”，进入触摸屏的工厂测试界面。



## 34 高级高级用户自定义协议——用户界面

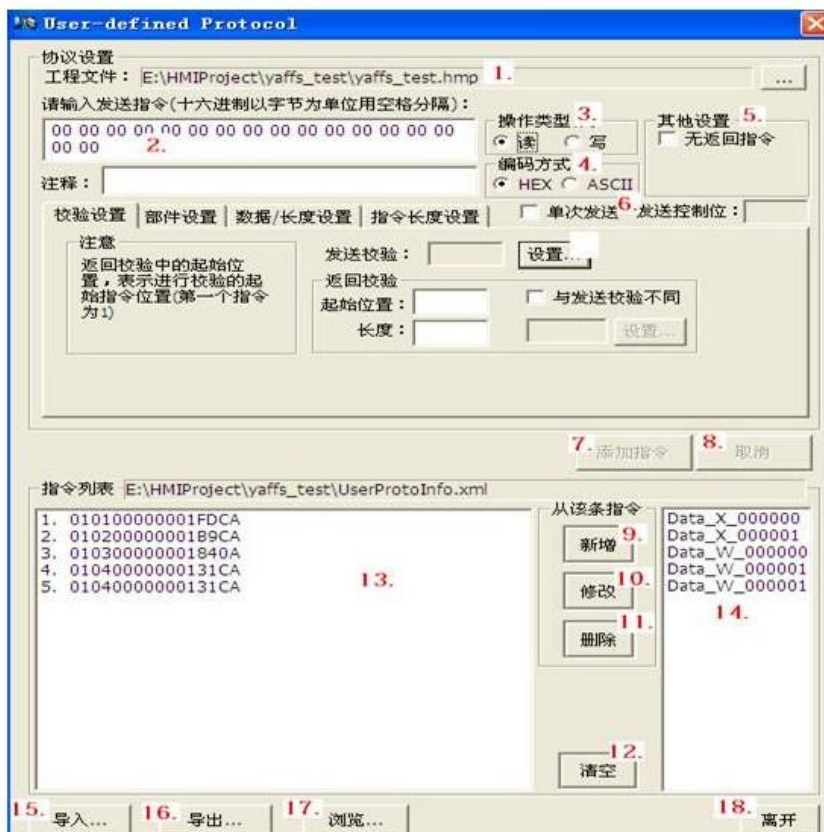
### 启动高级用户自定义协议

1. 选择 User-defined Protocol 连接
2. 打开高级用户自定义协议的配置对话框



1. 从组态软件通讯配置中启动
2. 单独启动高级用户自定义协议程序

程序界面一、主界面



1. 工程文件：选择要使用自定义协议的工程文件(如从组态软件启动忽略该步骤)。
2. 指令输入：输入所要使用的指令
3. 操作类型：选择指令操作类型，读寄存器或写寄存器。
4. 编码方式：指出指令的编码方式。HEX 和 ASCII
5. 其他设置：
  1. 无返回指令：指定某条指令发送后没有返回指令
  2. 先接收后发送：指定先进行指令的接收在发送指令
6. 单次发送：该条指令为由指定控制位来控制单次发送
 

发送控制位：设置该条指令的发送控制位
1. 添加指令：设置完参数后，添加该指令。
2. 取消：取消当前操作
3. 新增：新增一条指令
4. 修改：修改当前选中指令
5. 删除：删除当前选中指令
6. 清空：清空所有指令，并删除配置文件。
7. 指令列表：浏览添加的所有发送指令。
8. 部件地址列表：浏览添加的所有部件地址，与发送指令列表一一对应。
9. 导入：将用户自己制作的配置用户配置文件，导入到工程目录中使用。
10. 导出：将用户配置好的文件导出到任意位置

- 11. 浏览：浏览用户配置文件
- 12. 离开：退出程序

二、子界面校验设置

注意

返回校验中的起始位置，表示进行校验的起始指令位置(第一个指令为1)

发送校验：

CRC校验

设置...

返回校验

起始位置：

1

长度：

6

☐ 与发送校验不同

CRC校验

设置...

为指令添加控制字符：

ACK

添加

- 1. 发送校验：为指令添加发送指令的校验部分
- 2. 返回校验：
  - 2-1. 起始位置：指出返回指令中进行校验计算的第一个指令，位于整串指令中的起始位置。
  - 2-2. 长度：进行校验计算的指令长度
  - 2-3. 与发送校验不同：当返回校验与发送校验不同时，设置返回校验。
- 3. 为指令添加控制符：在 ASCII 模式下，为指令添加控制符部件设置

简介

设置该条指令属于哪个部件。

部件设置

部件类型：

位部件

部件地址

Data0

注释：

部件类型：选择位部件或字部件

部件地址：设置所要处理指令的部件地址

数据字数：部件显示某数据时该数据的字数

注释：对部件做简单注释数据/长度设置

简介

数据部分的设置因操作读和写的不同而不同，读操作针对返回指令中的数据，而写操作则针对发送时写入的数据。

返回数据

所在位置：数据所占字节的第一个字节在指令中的位置(以1开始)

所在位置：

4

长度：

1

写入数据

驱动添加指令：该项选中，则无需在发送指令中设置数据和校验部分

☒ 驱动完善指令

所在位置：

长度：

注：返回数据对读操作有效，写入数据对写操作有效。返回数据：指出返回指令中数据所占部分

- 1. 所在位置：数据部分的第一个字节位于整串指令中的位置（从 1 开始索引）
- 2. 长度：数据所占字节数

写入数据：

- 1. 驱动完善指令：该项选中，则无需在发送指令中设置数据部分和教育部分，由驱动进行添加（当前仅支持校验紧挨数据后添加的情况）。
- 2. 所在位置：与返回数据的所在位置同义
- 3. 长度：与返回数据的长度同义指令长度设置

注意

发送长度系统会根据输入的发送指令进行计算，无需设置。

指令长度

发送长度： /字节

返回长度： /字节

发送长度：程序将根据用户设置的发送指令进行计算

返回长度：用户设置某条发送指令对应返回指令的长度校验界面

校验

已输入指令：  
010400 0300 01

参加校验指令：  
010400 0300 01

校验类型

☐ 无校验

☒ CRC

☐ 求和

☐ 异或

☐ LRC

计算

=

C1 CA

☐ 从结果中取  位

取位

合成指令

结果：

010400 0300 01 C1 CA

确定

取消

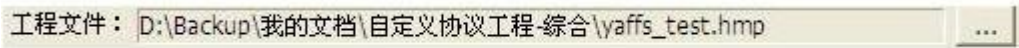
- 1. 已输入指令：用户设置的发送指令
- 2. 参加校验指令：用于计算校验的指令
- 3. 校验类型：选择校验类型
- 4. 计算：根据选择的校验类型，计算出校验
- 5. 从结果中取位：从计算出的校验中进行取位计算
- 6. 合成指令：将原有指令与校验合成
- 7. 结果：添加计算校验后的发送指令

34.1 高级用户自定义协议——使用步骤

使用步骤

1. 工程文件的选择：

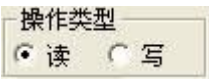
选择要使用自定义协议的工程文件(如从组态 i 中打开高级用户自定义协议窗口省略该步)



点击“新增”，开始第一条指令的添加：

2. 指令设置：

选择操作类型，如果为读取设备数据，则选“读”，如果为将数据写入到设备中，则选“写”。

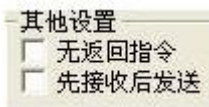


选择编码方式，如该条指令以十六进制形式输入，则选“HEX”；如以 ASCII 字符的形式进行输入，则选“ASCII”。

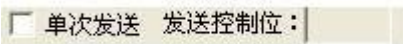


其他设置：指令的特殊情况。选中“无返回指令”，则该条指令，屏只进行发送，不进行对返回指令的处理；

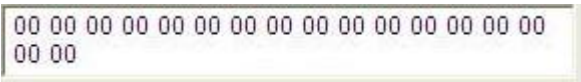
选中“先接收后发送”，则对于该条指令，屏先进行指令的接收，在发送用户设置的指令。



选中“单次发送”，设置发送控制位（1~10，工程上需设置对应的控制部件，寄存器类型为 OneCtrlBit）；设置完后，当将控制位置 1 时，屏将发送一次该条指令。



输入发送指令，当编码方式选 HEX 时，两个字符代表一个十六进制数，两个十六进制数之间以空格间隔；ASCII 则以字符串形式输入。



1. 检验设置面板：单击发送校验按钮，打开校验对话框，校验对话框的第一个文本框显示用户输入的指令，在第二对话框输入需进行校验的指令；选择校验类型，单击计算（如需取位，选中从结果中取位进行取位操作），最后单击合成指令，将发送指令与校验合成。确定返回高级用户自定义协议窗口。（界面请参阅高级高级用户自定义协议-界面介绍）

1. 部件设置面板：选择部件类型，单击部件地址弹出“地址编辑”对话框，进行部件地址的设置。（界面请参阅高级高级用户自定义协议-界面介绍）

2. 指令长度设置面板：发送长度程序会自行计算；填写返回指令长度。（界面请参阅高级高级用户自定义协议-界面介绍）

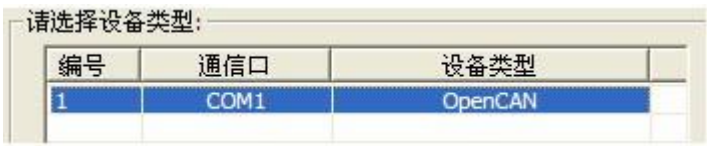
3. 数据/长度设置面板：

分别填写数据位于指令中的起始位置和长度。

1. 以上参数都设置完毕后，单击“添加指令”，进行新指令的添加。

35 OpenCAN 使用手册

1. 启动前需先选择 OpenCAN 设备类型



2. 从组态软件上启动，通过菜单“设置”——》“通讯口设置”——》“OpenCAN 配置”。
- 主界面



1. 数据访问：增加一条与寄存器地址有关的帧。
2. 帧列表：该列表显示用户所添加的每一条帧的部分主要参数。
3. 修改：对“数据访问”中新建的帧进行修改操作
4. 删除：删除列表中所选的帧；如列表中未选中任何帧，则删除第一条帧。
5. 清空：将清空列表中的所有帧。



6. 浏览：在 IE 中显示 xml 格式的配置文件。
7. 完成：完成帧的配置并推出程序。

### 数据访问界面

1. ID：设置一条 CAN 帧的 ID，该 ID 格式为十六进制
2. ID 分配：将 ID 按 PF、PS 和 SA 拆分设置。

1. 帧类型：选择该条帧为标准帧或者扩展帧。
2. 帧格式：选择该条帧为数据帧或者远程帧
3. 数据：设置 CAN 帧的数据部分，以两个数字代表一个十六进制数，并以空格间隔开来；根据 CAN 报文定义最大支持 8 字节数据。
4. 使用地址：设置与该条 CAN 帧相关寄存器地址，该寄存器地址与主态上设置的寄存器地址一一对应。地址所获得的数据按连续分配。

编辑：按位或字的地址格式设置位地址或字地址

选项：设置地址与帧相关联的选项，进入“寄存器地址选项”界面，具体浏览以下的“寄存器地址选项”界面。

交互：触摸屏的交互方式有两种，一种是触摸屏主动发送帧，设备接收并进行处理和反馈；另一种则相反触摸屏被动接收设备发来的帧，进行处理和反馈。

先接收后发送：选中该项则触摸屏的此次交互将作为被动方，触摸屏将先接收 CAN 帧后作出相应反馈。未选中该项的交互方式则相反。

反馈方式：反馈方式包括无应答、确认应答和数据应答。

无应答：设备或触摸屏将不会接收到反馈

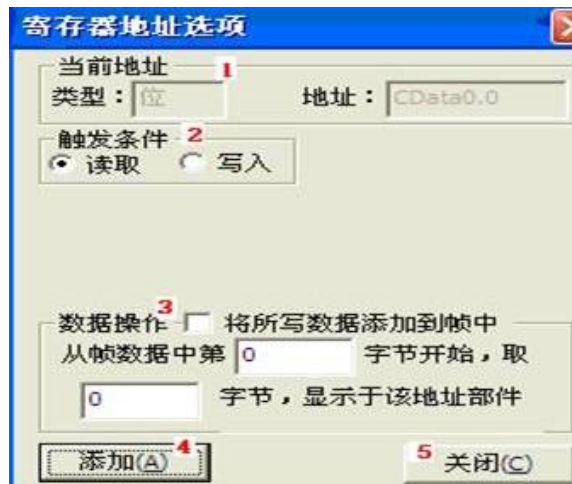
确认应答：设备或触摸屏将接收到带确认的反馈，可进行数据部分的比较

数据应答：设备或触摸屏将接收到带数据的反馈，需设置反馈帧中要剥离出来的数据存放于寄存器地址。

应答 ID：如果该地址想要接收数据位于不同 ID 的帧上，则需设置该项，勾选“与发送方不同”， 在后面的输入框输入不同的 ID。

未进行该项设置，则屏将接收与发送方相同的 ID 的帧，并处理该帧的数据。

### 寄存器地址选项界面



1. 当前地址：显示用户在“数据访问”界面设置的寄存器类型和寄存器地址

2. 触发条件：基于屏上寄存器的两种操作“读”与“写”。

如果选中“读”，则该寄存器地址在进行设备数据的读取，读取方式为循环发送用户设置的帧。

如果选中“写”，则该寄存器地址进行屏数据写入到设备，写入方式为用户对屏进行一次写操作，将触发发送一次用户设置的帧。

### 3. 数据操作：对触发条件中的读写操作进行相应的数据设置。

如果触发条件为读操作，该部分需要设置当前地址所要获取到的数据在帧中的位置和长度。

如果触发条件为写操作，则有两种情况：

A. 未选中“将所写数据添加到帧中”选项，则当用户在屏上进行写操作时，直接发送用户设置的帧。

B. 选中“将所写数据添加到帧中”选项，则当用户在屏上进行写操作时，程序将根据用户设置的数据插入位置和长度，将写入的数据置于用户设置的帧进行发送。

### 位置和长度输入格式：

1. 如果寄存器地址类型为位时，则需用小数点表示字节中的位，如 1.1 表示数据帧中 8 个字节的第一个字节的第一个位，并且长度的单位为位，以此类推。

2. 如寄存器地址类型为字时，则只需用整数表示字节，如 1，表示数据帧中 8 字节数据的第一个字节，并且长度的单位为字节，以此类推。

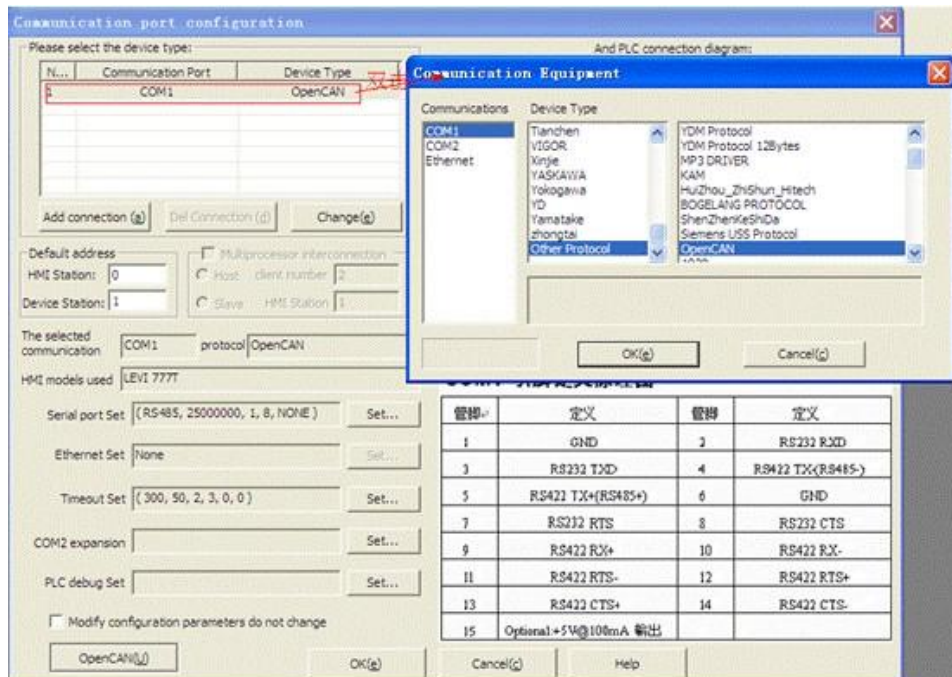
4. 添加：添加当前的设置。

5. 关闭：关闭该窗口。



使用示例第一步，选择协议。

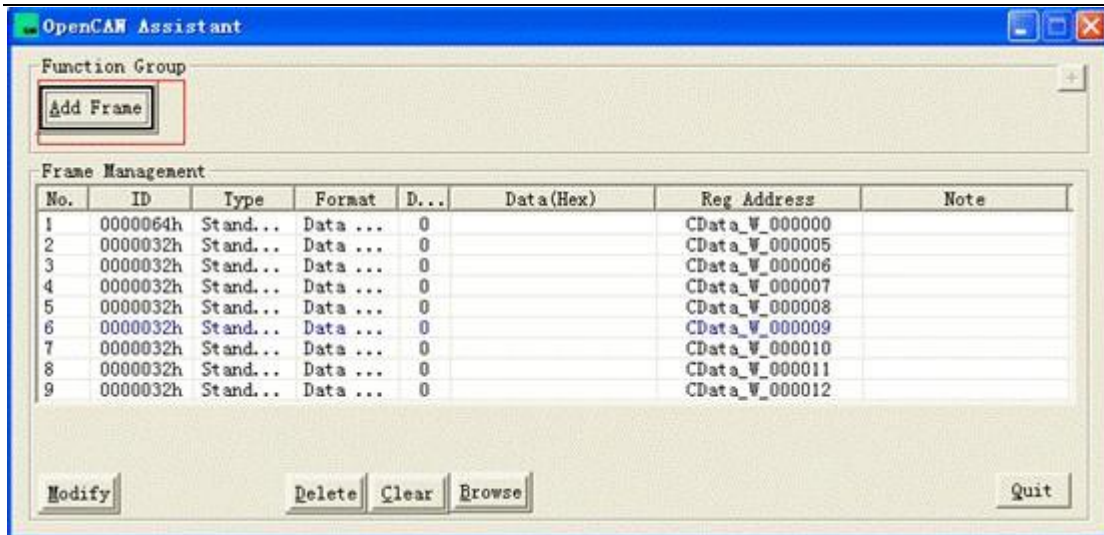
在我们创建工程的时候就需要你设置通讯串口，HMI 型号跟设备类型。这个在设备类型那边选择我们的 opencan 协议。就是打开我们通信口设置，双击设备类型然后找到 opencan 点确定。



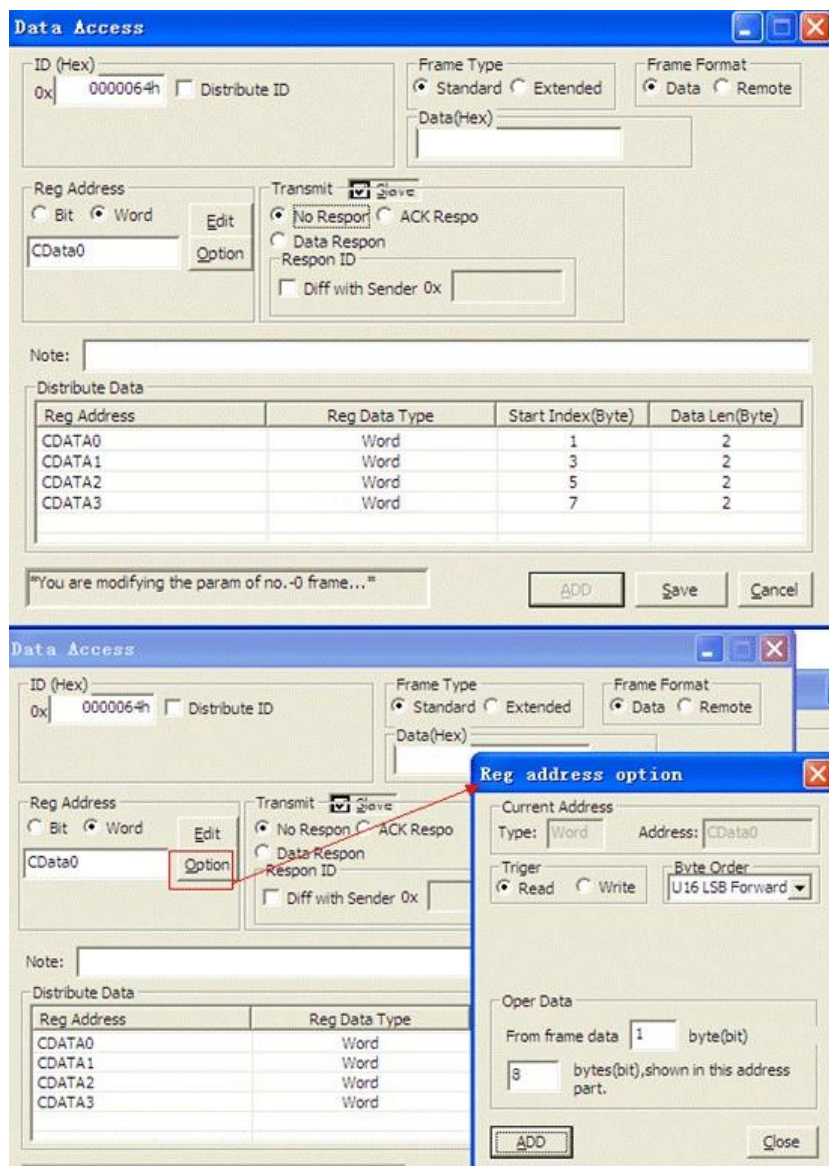
## 第二步添加帧指令



在我们设置、然后通信口配置。接着在左下角有个 opencan (u) 点击就能进入以下界面



然后我们点击添加指令。根据协议，id 输入 0000064h 然后地址是字地址，点编辑。  
我们都是从设备那边读数据，所以数据不用填，然后不需要反馈所以先接受后发送跟无应答都勾起来。



一开始我们是读的，所以在选项那边点读，然后选择 U8 SingleByte 这个是以字节为单位的。  
然后点 ADD 就可以了，最后就是在工程画面添加你编辑的地址。

## 36 1939 寄存器编辑说明

CAN 协议 1939 地址编辑方式说明

地址编辑器中采用 2 个十六进制的数据代表一个信息

例如:此图片中 十六进制 04 代表 PF 值

十六进制 30 代表起始地址

十六进制 08 代表长度

十进制 167 代表 SOURCE ADDRESS(SA)



数据:0101 0101 0101 0101 0101 01.....

例如地址 010008 PF=0x01: 起始地址 0x00 、长度 0x08 、数据为 0101 0101

数据:1110 0010 1110 1110 1010 1001 0.....

例如 地址 320703 PF=0x32: 起始地址 0x07、 长度 0x03、 数据为 011 起始地址的下标是从 0 开始

长度是根据起始地址的那位开始计算, 若长度大于 0x10 必须设置成双字, 否则设置成单字即可, 长度不允许超过 0x20。

数据

数据格式(E): 无符号十进制 字数(E): 双字

小数点位置

允许的最大总位数: 10

整数位数(Z): 3 小数位数(D): 0

数据范围

允许的数据范围: 0-999

最小值(O): 0 最大值(M): 1

说明: 对于二进制、八进制、十六进制的数据范围, 要求以十六进制的格式输入。对于其他格式范围一律使用十进制

确定 取消

增益、偏移的使用

是否缩放	否
比例增益	不可用
偏移	不可用
四舍五入	否

如果要使用增益或者偏移, 是否缩放选择“是”, 然后直接在上面输入数值即可  
若不明确增益, 偏移的确定数据, 可以点击

数据范围

是否缩放 是

比例增益 1.00000

偏移 0.00000

比例增益计算

工程数据下限: 0 工程数据上限: 32767

原始数据下限: 0 原始数据上限: 5

计算

计算结果

右边的三角箭头, 就会出现此计算工具.

下面是计算公式的说明:



**说明**

计算公式为： $Y=kX+b$ 。其中X为人机界面输入(显示)的数据，Y是写入到PLC的数据(或者从PLC读取的数据)。

工程数据：是人机界面输入(显示)的数据。

原始数据：是写入到PLC的数据(或者从PLC读取的数据)。

$k=(\text{原始数据上限}-\text{原始数据下限})/(\text{工程数据上限}-\text{工程数据下限})$ 。

$b=\text{原始数据下限}-k*\text{工程数据下限}$ 。

## 37 AB PLC 地址编辑说明

可操作地址是以一个 Allen-Bradley PLC 与触摸屏相连接的情况而定，若有扩展模块或其他特殊情况，请参照 Allen-Bradley PLC 使用说明手册。

以下是以 Allen-Bradley MicroLogix1200 为例：

**位地址 I：** 地址范围：0.0 ~ 255.15

小数点前面的数据为 0~255 为字地址, 格式为十进制；小数点后的 0~15 为子地址格式为十进制。

**位地址 B：** 地址范围：000.0 ~ fff.15

粉红色范围为 0~f 格式为十六进制, 代表块号；

红色范围为 00~ff 格式为十六进制, 代表字地址；

小数点后面为 0~15 代表子地址, 格式为十进制

**字地址 S：** 地址范围:0~255 代表字地址, 格式为十进制；

**字地址 TS：** 地址范围 0000:ffff 红色 00~ff 代表块号, 格式为十六进制；

黄色 00~ff 代表字地址, 格式为十六进制。

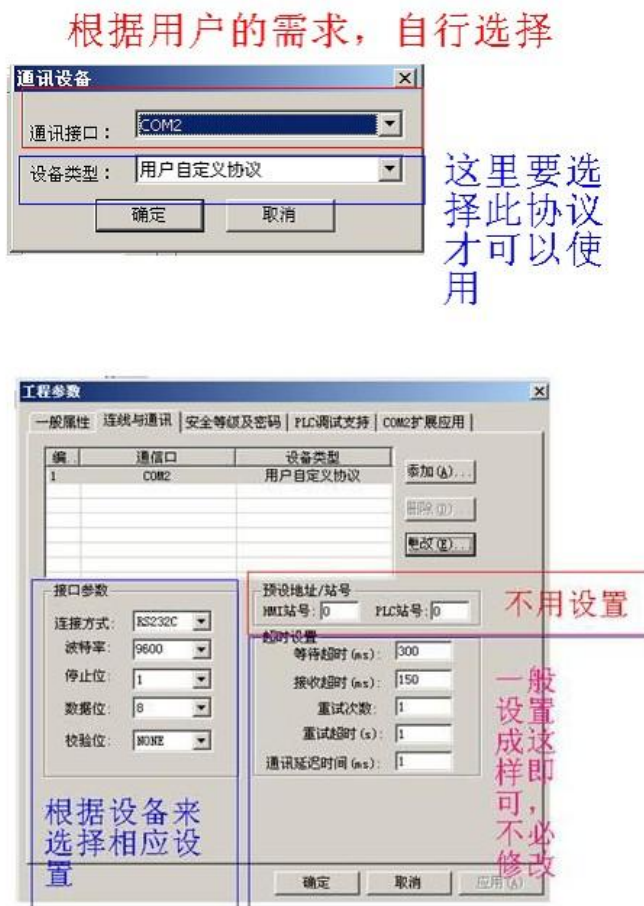
寄存器地址 TP、CS、CP、N、F 和 TS 的地址编辑一样的。

PLC 位地址 类型	地址格式	地址范围
I	dd. dd	I 0.0 ~ 255.15
O	dd. dd	O 0.0 ~ 255.15
B	<b>fff. dd</b>	B 000.0 ~ fff.15
S	dd. dd	S 0.0 ~ 255.15
N	ffff. dd	N 000.0 ~ fff.15
PLC 字地址类型	地址格式	地址范围
S	ddd	S0 ~ 255
TS	<b>ffff</b>	TS0 ~ FFFF
TP	<b>ffff</b>	TP0 ~ FFFF
CS	<b>ffff</b>	CS0 ~ FFFF
CP	<b>ffff</b>	CP0 ~ FFFF
N	<b>ffff</b>	N0 ~ FFFF
F	<b>ffff</b>	F0 ~ FFFF

d 表示 10 进制，f 表示 16 进制 不同的 PLC 型号支持的寄存器有可能不同，以实际 PLC 支持的寄存器为准。

## 38 用户自定义协议使用说明

### 一、 配置



#### 协议设置：

这里的设置最主要是设备类型，必须选择“用户自定义协议”

通信口，可以根据用户的需要来自行修改

#### 接口参数设置：

主要根据设备来修改

#### 预设地址、站号：

这里一般不用设置

#### 超时设置：

设置成和图片一样即可。

### 二、 串口发送数据的操作

- 1、 填写 SendNum 寄存器 这个寄存器是要发送串口的数据，将数据从串口发出去，单位为 BYTE 。
- 2、 设置 SendLen 寄存器 这个寄存器是要发送的数据长度，单位为 BYTE
- 3、 将 SendFlag 寄存器置位 将数据全部写完后，然后将此寄存器设置为 1，即可发出数据 。



**示例：** 此例子将往串口发送长度为 10，数据为

11 22 33 44 55 66 77 88 99 AA 的数据

### 三、 串口读取数据的操作

1、 可以先将 ReadNum 位寄存器的数据清空 ReadNum 寄存器是从串口读取到的数据，放入这里，单位为 BYTE

2、 设置 ReadMaxLen 位寄存器 这个寄存器是要从串口接收数据的长度，最好将此寄存器接收的长度设置一定范围，否则将影响串口通信的速度，即必须等待超时

3、 将 ReadFlag 位寄存器置位 ReadFlag 置位后，就开始读取串口，将读取到的数据放在 ReadNum 寄存器中，将实际读取到的字节放在 ActualReadLen 寄存器中。

4、 ActualReadLen 位寄存器为实际收到的数据 可以比较此寄存器，来确定是否读取成功，如果为 0，则代表没有读取到任何数据 **示例：**

此例子将从串口读取 10 个数据，放入 ReadNum 寄存器中

### 四、串口清空操作

COMIO 寄存器用来清空串口的缓冲区，将此位置位后，串口的所有数据将被清空

### 五、CRC 校验发送数据的 CRC 校验码

Check0 到 check4 是计算发送数据的 crc 校验，也就是 sendnum 寄存器地址里的值的校验。

Check0: 开始地址。

Check1: 长度

Check2: 值为 1，开始计算

Check3: crc 校验高位，

Check4: crc 校验低位。

**举例：** 计算 sendnum3 到 sendnum10 这些数据的 cRC 校验码

Check0=3

Check1=7

Check2=1

最后的 crc 校验在地址 check3 和 check4 里读取。

接收数据的 crc 校验码 Check5 到 check9 是计算接收数据的 crc 校验,也就是 readnum 寄存器地址里的值的校验。

Check5: 开始地址。

Check6: 长度

Check7: 值为 1, 开始计算

Check8: crc 校验高位,

Check9: crc 校验低位。

**举例:** 计算 readnum3 到 readnum10 这些数据的 cRC 校验码

Check5=3

Check6=7

Check7=1

最后的 crc 校验在地址 check8 和 check9 里读取。

## 39 开放内部地址设置以太网 IP

内部地址设置以太网 IP

HMITERM 中开放内部地址来设置以太网通讯的 IP 地址，设置 IP 后要让 HSW0=1 才会保存新 IP 地址，且重启触摸屏后新 IP 地址生效，用于通讯。

这里的 IP 地址指目标地址：

目前开放的地址是 HSW561-----HSW660. 5 个一组。每个连接分配 5 个地址。

1. HSW561----565 是第一个以太网的。其中 HSW561, HSW562, HSW563, HSW564 是 IP 地址

192.168.1.46 HSW565 是端口号 502

2. HSW566---HSW570 是第二个以太网的地址 前四个是 IP，192.168.1.201 第 5 个是端口号 500

以此类推，通过通讯口中使用的以太网个数，实行地址偏移，用来存储它对应的 IP 号。

**通讯口配置**

请选择设备类型:

编号	通信口	设备类型
1	Ethernet	ModBus TCP Slave
2	Ethernet	FATEK TCP ASCII
3	COM2	ABB

添加新连接(a)    删除连接(d)    更改连接(e)

预设地址/站号  
HMI站号: 0  
设备站号: 1

☐ 启用多机互联  
☐ 主机    从机数量: 2  
☐ 从机    HMI站号: 1

所选择的通讯口: Ethernet    协议: ModBus TCP Slave

所使用的HMI型号:

串口参数设置: 无    设置...

以太网参数设置: 192.168.1.46:502    设置...

超时设置: ( 300, 50, 2, 3, 0, 0 )    设置...

COM2扩展: 无    设置...

在通讯过程中，如果 HSW561---660 的地址被改变了，同时 HSW0=1 那么就会将改变的值保存起来，重启触摸屏将会以新的 IP 地址来通讯（可以硬重启或者通过地址 HSW24=1 来软重启）

第一个以太网通讯IP	<small>以太网地址1</small> 88888	<small>以太网地址2</small> 88888	<small>以太网地址3</small> 88888	<small>以太网地址4</small> 88888	<small>以太网地址5</small> 88888
第二个以太网通讯IP	<small>以太网地址6</small> 88888	<small>以太网地址7</small> 88888	<small>以太网地址8</small> 88888	<small>以太网地址9</small> 88888	<small>以太网地址10</small> 88888
第三个以太网通讯IP	<small>以太网地址11</small> 88888	<small>以太网地址12</small> 88888	<small>以太网地址13</small> 88888	<small>以太网地址14</small> 88888	<small>以太网地址15</small> 88888
⋮					
⋮	<small>以太网地址16</small> 88888	<small>以太网地址17</small> 88888	<small>以太网地址18</small> 88888	<small>以太网地址19</small> 88888	<small>以太网地址20</small> 88888
⋮					
⋮	<small>以太网地址21</small> 88888	<small>以太网地址22</small> 88888	<small>以太网地址23</small> 88888	<small>以太网地址24</small> 88888	<small>以太网地址25</small> 88888
⋮					

保存IP
重启HMI

注：modbus 协议还增加了字中取位的功能，可以访问像地址如 100.1 等格式的地址，不过发出去的功能码跟字读写的是一样的！

寄存器类型	功能码（十六进制）	访问方式	可读可写？
3	04（读）	16 比特访问	√
	06（写单个寄存器）		
	10（写多个即连写）		
4	03（读）	16 比特访问	√
	06（写单个寄存器）		
	10（写多个即连写）		
W6	03（读）	16 比特访问	√
	06（写单个寄存器）		
	10（写多个即连写）		
W16	03（读）	16 比特访问	√
	0F（写多个寄存器）		

寄存器类型	功能码（十六进制）	访问方式	可读可写？
0	01（读）	比特访问	√
	05（写单个位）		
	0F（写多个位即连续写）		
1	02（读）	比特访问	√
	05（写单个位）		
	0F（写多个位即连续写）		
W5	01（读）	比特访问	√
	05（写单个位）		
	0F（写多个位即连续写）		
W15	01（读）	比特访问	√
	0F（写多个位即连续写）		

## 41 关于 Modbus RTU master 映射内部 HDX、HDW 地址

位地址 HDX3000.0~HDX3499.15 对应 0 0 ~0 7999 可读可写

字地址 HDW3500~HDW7999 对应 4 0 ~4 4499 也是可读可写

## 42 初级教程

{注：该教程相关图片是旧软件截图，但是操作过程不变，原理一样！}

### 42.1 软件帮助在哪里

1. 软件打开，按 F1 键
2. 软件工具栏上“帮助”。



### 42.2 软件版本

打开软件-----关于（查看时间和版本号）



### 42.3 Usb 线的类型（D 口、一口、迷你口）

----->:430T, 777T

1. .usb 接口：用于下载工程文件、配方或者镜像文件，u 盘数据下载。



----->:777A, 102A

1.D 型口：用于下载工程文件、配方或者镜像文件；

2.usb 接口：用于 u 盘数据下载。



----->:908T, 104T

背面实物图：





1. usb 迷你口：用于下载工程文件、配方或者镜像文件；
- 2..usb 接口：用于 u 盘数据下载



#### 42.4 232 的下载线接线：

----->:交叉接线

2-----	3
3-----	2
5-----	5

#### 42.5 串口引脚的定义：

----->：串口的引脚分别有 9 针、15 针和 25 针的三种类型。

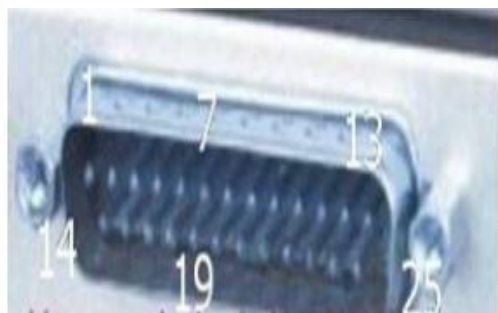
9 针的定义如下图：



15 针的定义如下图:



25 针的定义如下图:



#### 42.6 各个型号的 com 口 (com1 和 com2、com3)

----->:430T, 777T



----->:777A



----->:102A



----->:908T, 910T



#### 42.7 下载时的工具选择（PC 端口）

----->:1. 打开工程上传和下载软件

2. 根据使用的下载工具在 PC 端口上选择相应的工具。（串口、usb、usb 转串口）



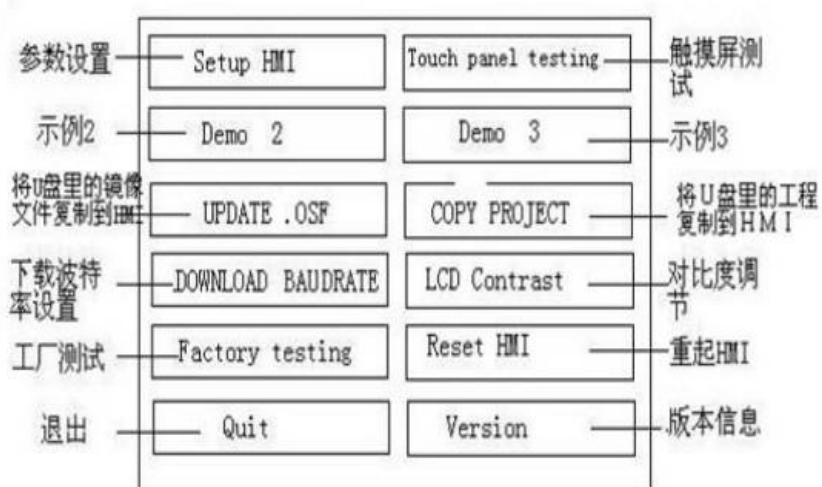
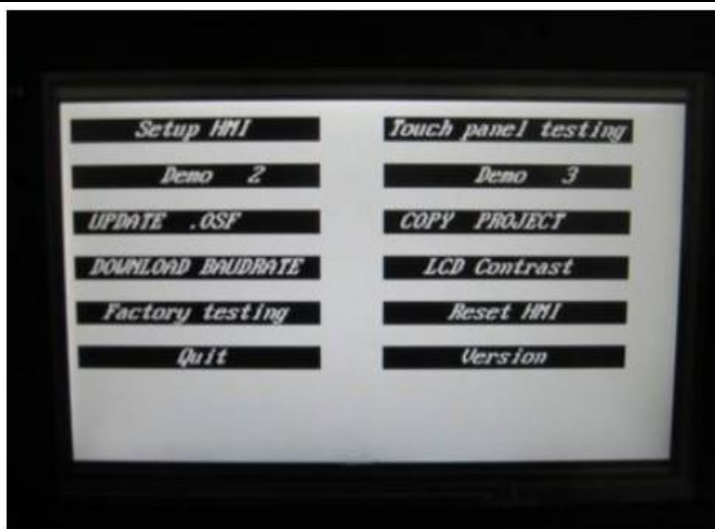
## 42.8 测试界面的进入及内容

### (1) 怎样进入触摸屏的测试画面？

----->:解决：按住屏幕的右上角大约 3~5 秒钟，如图



出现如下界面：



选择“Touch panel testing”，进入触摸屏的测试界面；

## (2) COM 口参数设置

----->:解决：按住屏幕的右上角大约 3~5 秒钟，进入测试画面，点击 *Setup HMI*





出现如下界面



选择需要设置的 com 口



如选 COM1 进行设置，出现以下画面：





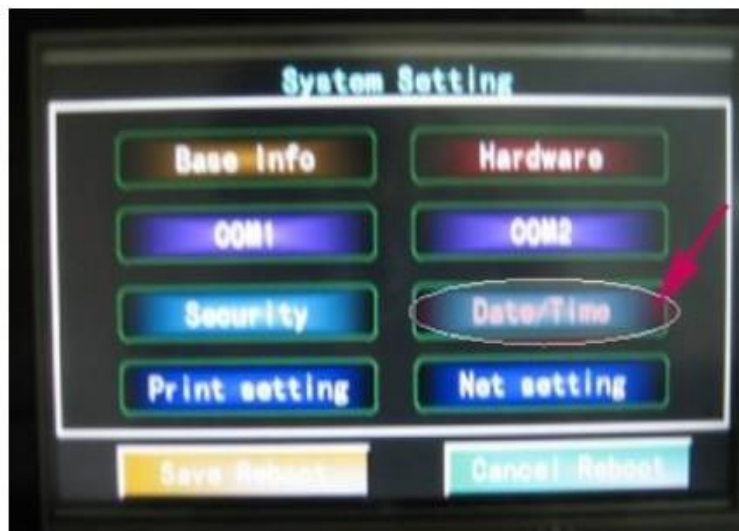
按工程需要设置参数，设置完成点击 Save 保存，保存成功点击 Home 返回



### (3) 时间设置

解决：按住屏幕的右上角大约 3~5 秒钟，进入测试画面，点击 Setup HMI

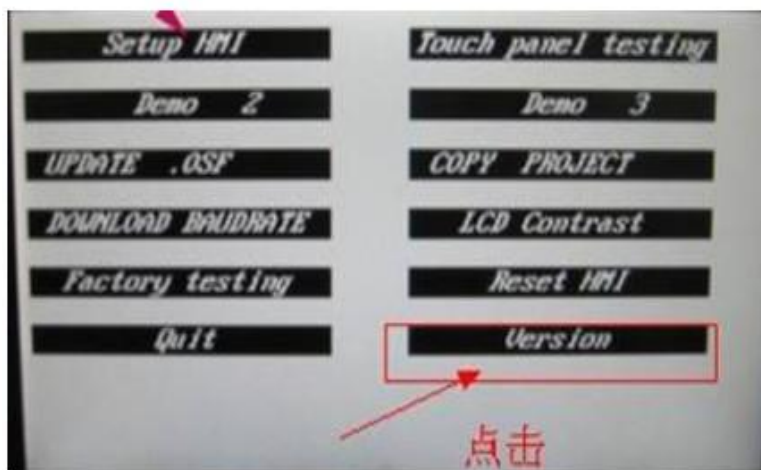




设置完成点击 save 保存，Home 返回

#### (4) 触摸屏版本信息的查看

1. 按住屏的右上角
2. 点击 “version”
3. 查看 “system”、“HMIApplication”时间很版本（用于更新相关镜像）

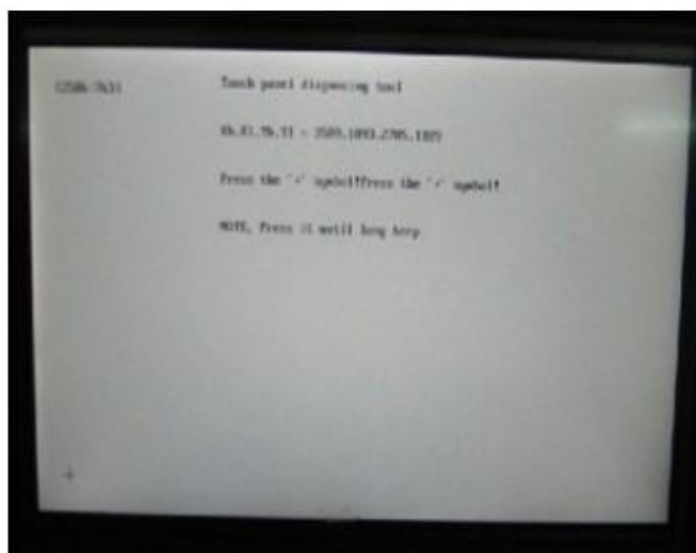


(5) 触摸屏按键位置不准确，该怎么办？

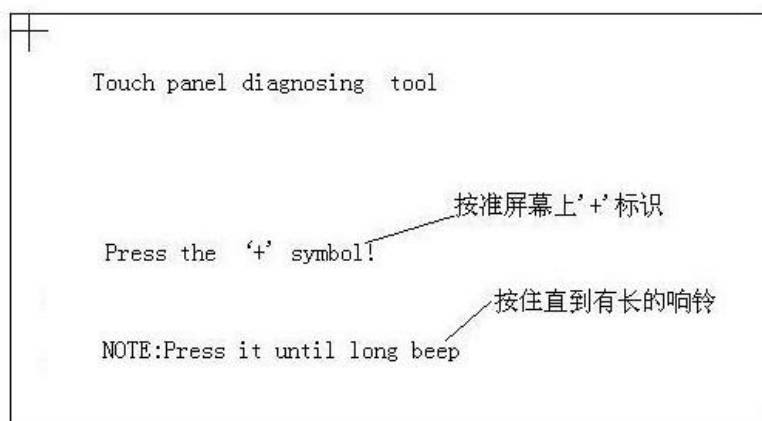
----->解决：按住屏幕的左上角大约 3~5 秒钟，



出现如下界面



依次点中左上角，左下角，右下角，右上角，中心点的 '+'，重新启动即可。



## 42.9 u 盘根目录和 u 盘升级文件

----->:比如说我的 u 盘是 G 盘：文件是 project.hmt

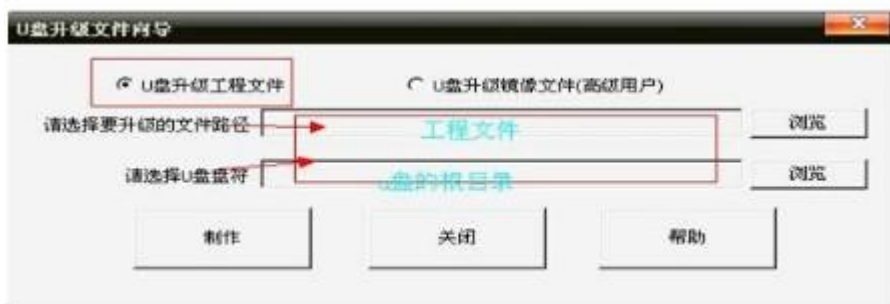
那么根目录文件---->G:\project.hmt

----->:u 盘升级文件：

1. 软件“欢迎界面”-----u 盘下载【工程-----制作 u 盘升级文件】



2. 工程文件的选择和 u 盘存放工程文件



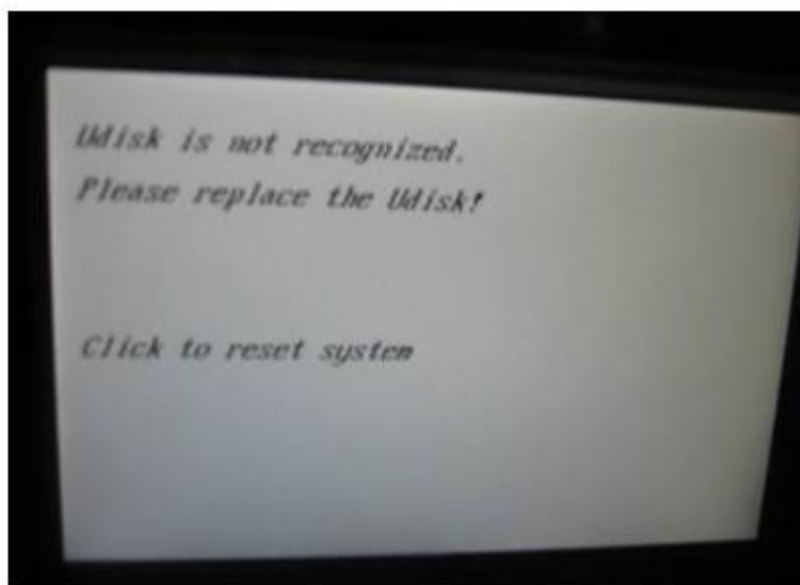
3. 按住屏的右上角 3-5 秒进入黑白的设置界面





4. 成功提示 “copy project success” 下载成功，自动重启人机进入工程。

失败提示 “Udisk is not recognized ..... ”（检测下 u 盘工程是否有且存在 u 盘根目录）



#### 42.10 画面的复制:

----->:

1. 软件工具栏----->画面复制
2. 在当前画面 “右击” -----画面复制



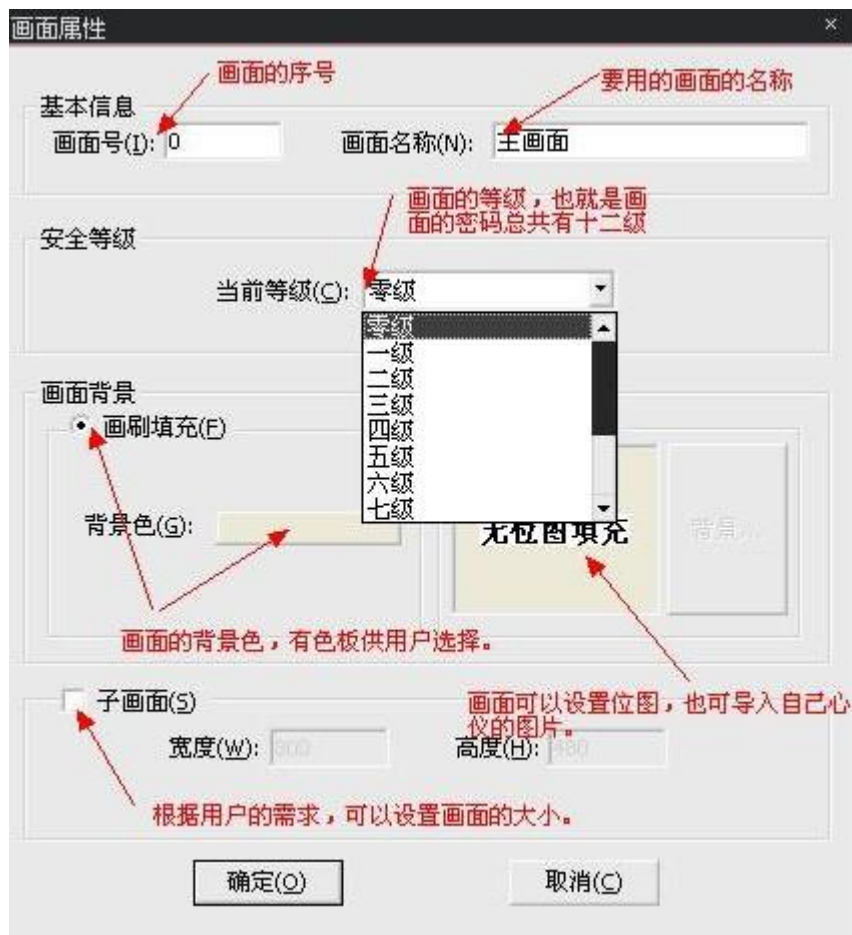
## 42.11 画面属性的设置

----->:

1. 在当前画面“右击”----画面属性
2. 属性编辑框，选择在当前画面“右击”----画面属性

画面属性编辑框如下图：基本信息、安全等级、画面背景、子画面编辑





## 42.12 子画面的制作

----->: 新建画面



勾起：“子画面”设置大小



42.13 画面（部件）安全等级：

----->

- 1. 设置-----工程参数-----画面等级/部件等级

工程参数

一般属性 | 安全等级及密码 | HMI以太网设置 | 分期付款模式 | 部件等级密码

☒ 启用安全保护

安全保护类型

☒ 画面密码保护

等级一:  等级七:

等级二:  等级八:

等级三:  等级九:

等级四:  等级十:

等级五:  等级十一:

等级六:  等级十二:

启动等级:  ☐ 各等级密码独立

最大等级数:

☐ 按键密码保护

密码:  按键超时:

设计者密码:

确定 取消 帮助

2.相应的画面/部件属性设置画面的等级:



42.14 脚本的使用和分类：

----->内部地址可用于画面脚本、背景脚本、全局脚本中, 外部地址尽量用于背景脚本中，避免影响速率。

脚本有 3 类：

画面脚本、全局脚本、背景脚本

画面脚本（实现当前画面效果的脚本）3 个地方可以进入：

1. 工具栏上，如下图。



2. 画面右击-----画面脚本



### 3. 属性编辑框-----画面脚本



#### 全局脚本:

1. 设置-----全局脚本
2. 属性编辑框-----全局脚本



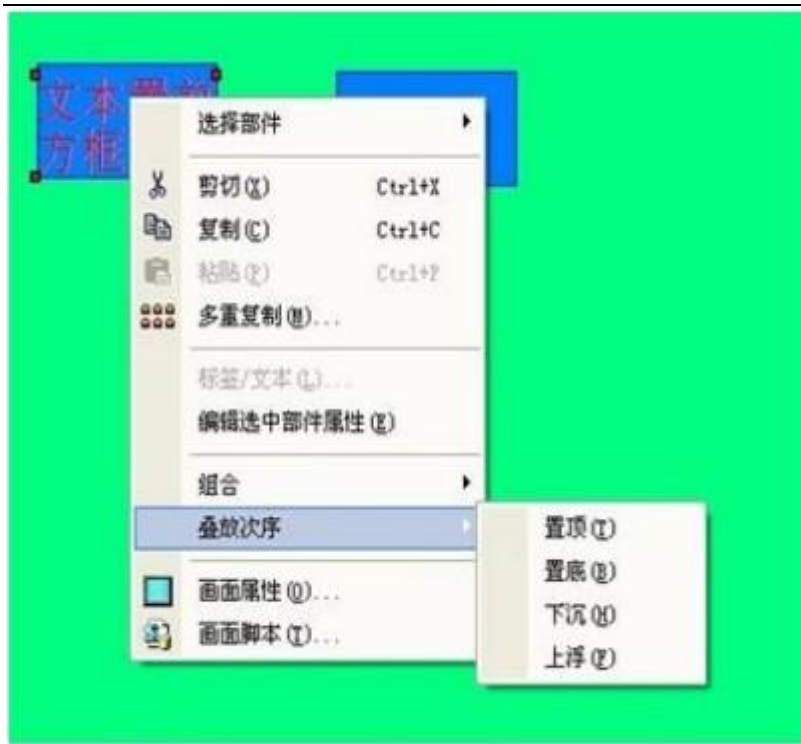
### 背景脚本:

- 1 设置-----背景脚本
- 2 属性编辑框-----背景脚本



### 42.15 部件叠放顺序的使用:

----->: 叠放顺序是指同一个位置的两个或两个以上部件的叠放顺序。  
选中叠放部件中的其中一个目标部件-----“右击”-----“叠放次序”，如下图



42.16 工程所有使用过的部件列表在哪里：

----->工具栏-----部件列表



打开部件列表如下图

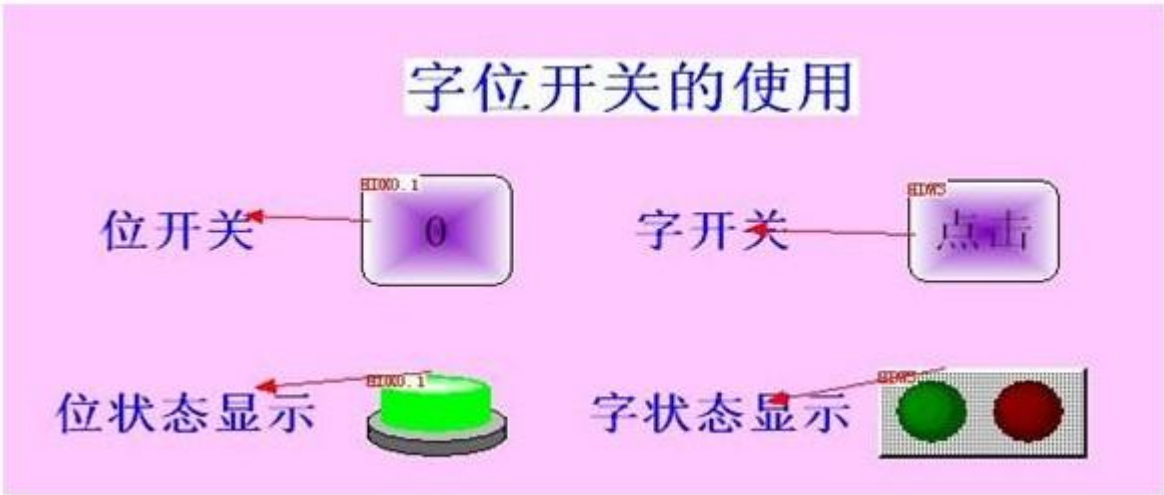


42. 17 字位开关的使用：

----->

- 1. 工程使用部件“位开关”、“字开关”、（“位状态指示灯”、“字状态显示”）
- 2. 选择相应的部件，编辑左边的属性编辑框，对相应的属性按照实际的需要设置相应的属性。


工程设置如下图：



属性设置如下图：

位开关属性设置：



属性名称	值
一般	
描述	BS_0
轮廓	[246,126],[77,64]
操作地址	HDX0.1
快速更新	否
开关类型	切换型
复位实时值不可用	
监测	是
监控地址	HDX0.1
显示方向	0度
向量图	真彩型样式\真彩00
边框颜色	 \$cccccc
外观位图	[无位图]
文字对齐	居中
标签起始点	不可用
是否反向显	否
是否闪烁	否
闪烁类型	不可用
闪烁频率(1	不可用
是否隐藏	否

字开关属性设置：

字开关

基本 | 文本 | 图形 | 扩展

地址编辑

读取地址

H0V3

编辑地址

☒ 读取地址与写入地址相同

写入地址

H0V3

编辑地址

数据格式

格式

无符号十进制

字数

单字

功能

☐ 字设置

☒ 递增

☐ 递减

数值

操作数

1

上限

3

确定

取消

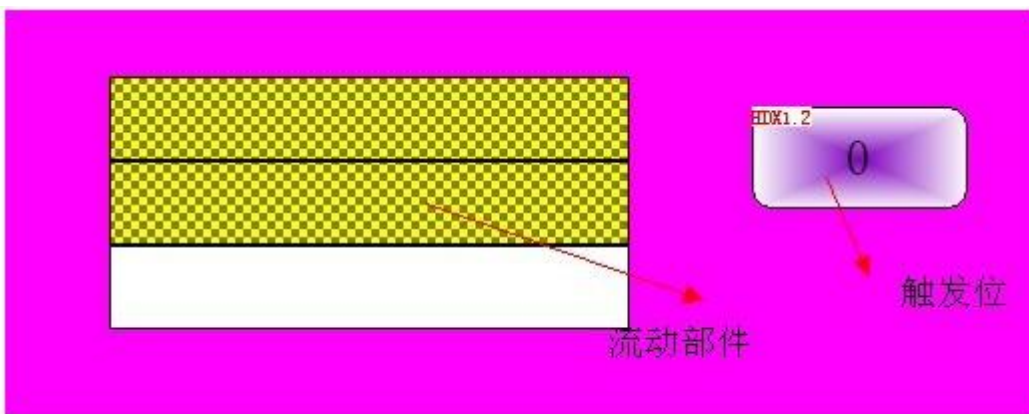
帮助

模拟效果如下图：



#### 42.18 流动部件使用：

----->:工程里拖用“流动部件”，设置相关属性。



流动部件设置如下图：

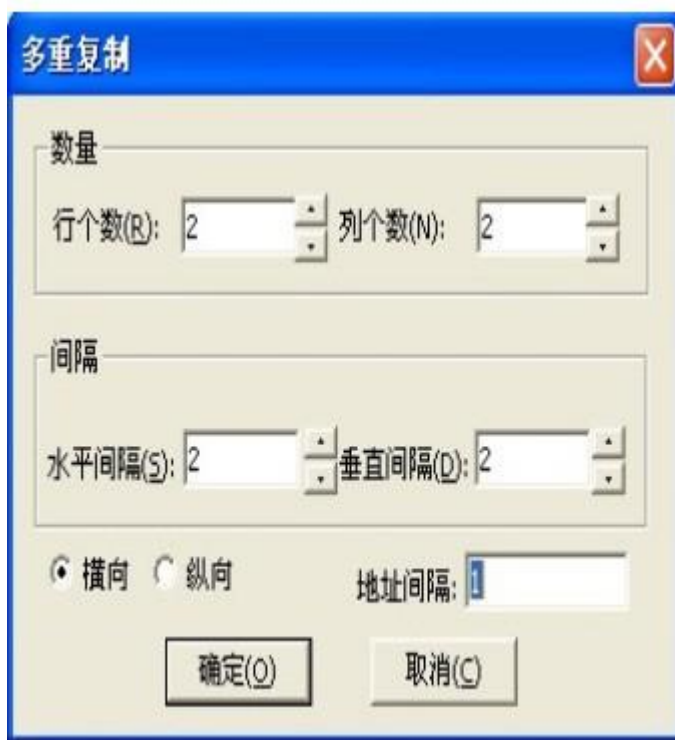


## 42.19 多重复制

-----> 选择目标部件-----右击-----多重复制

(可以设置行数和列数，间隔的设置、横向纵向的设置、地址间隔的大小。)

设置如下：



效果如下：

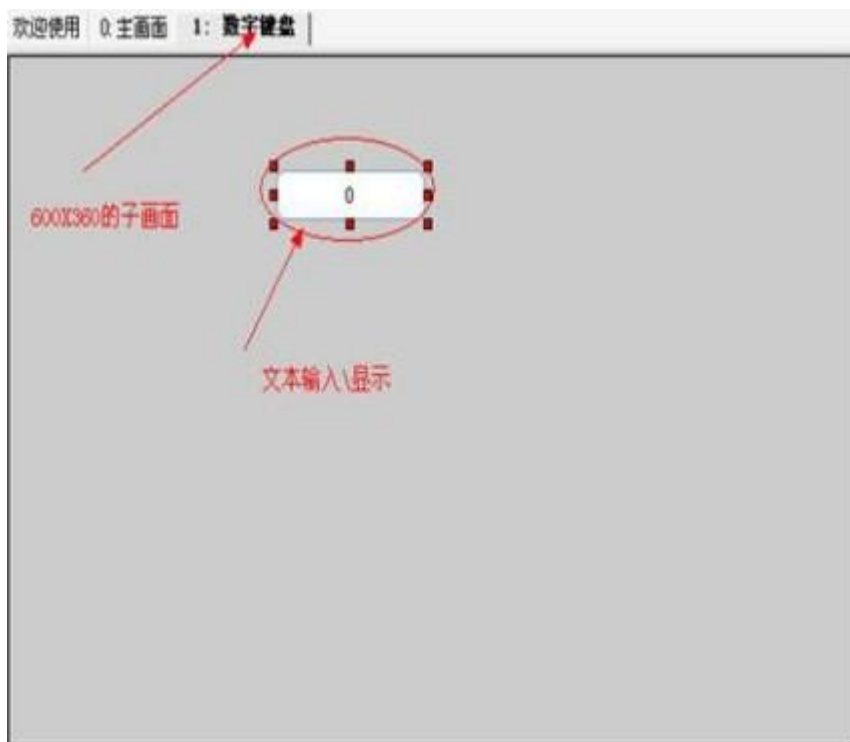


## 42.20 键盘制作使用的部件

----->:如图, 1. 新建一个 600×360 的子画面, 点击确定。



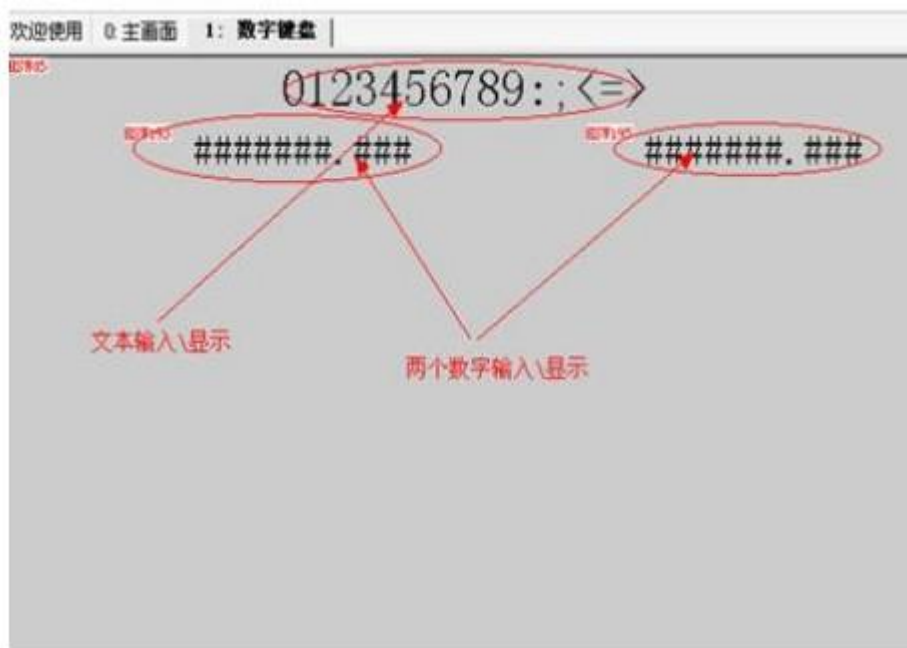
2. 在画面编辑区添加一个文本输入/显示部件, 双击文本属性。



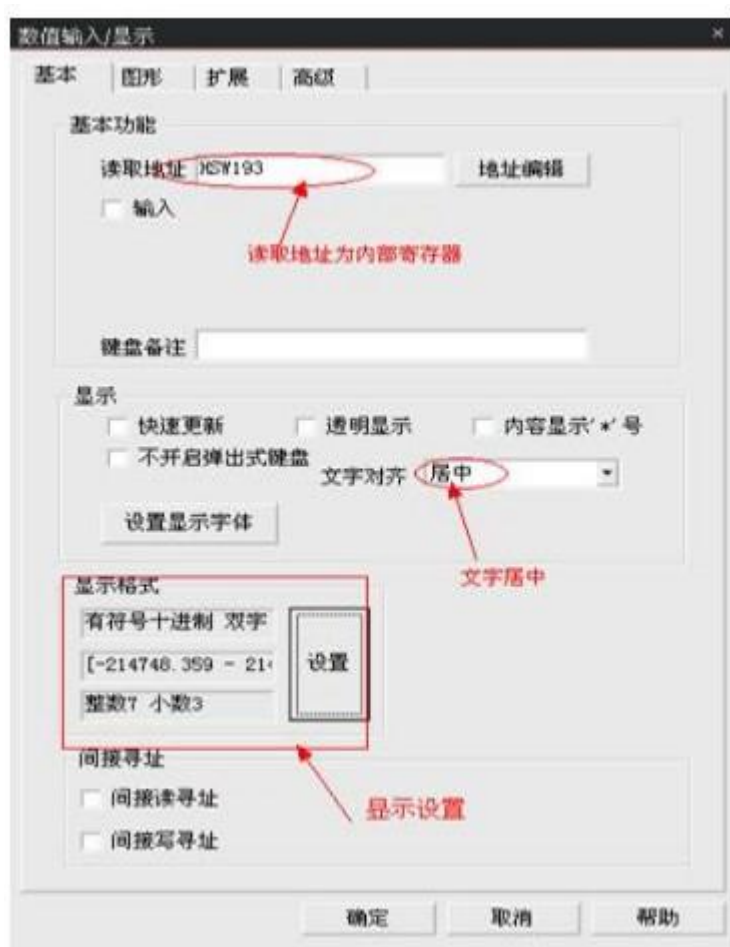
3. 读取地址为 HSW00045, 操作字符设置为十五, 点击确定。



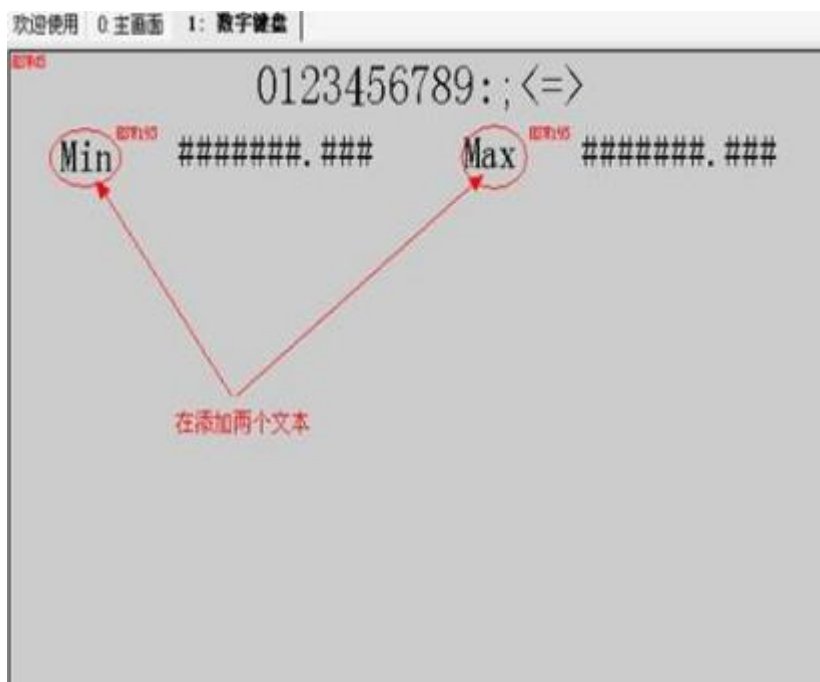
4. 在添加两个数值输入/显示部件，双击文本属性编辑。



如图所示，读取地址为 HSW193，文字居中。



5. 在添加两个文本，双击文本的属性对话框。



如下图所示，文本的语言为 Min 和 Max, 字体显示方向可设置为从上到下。

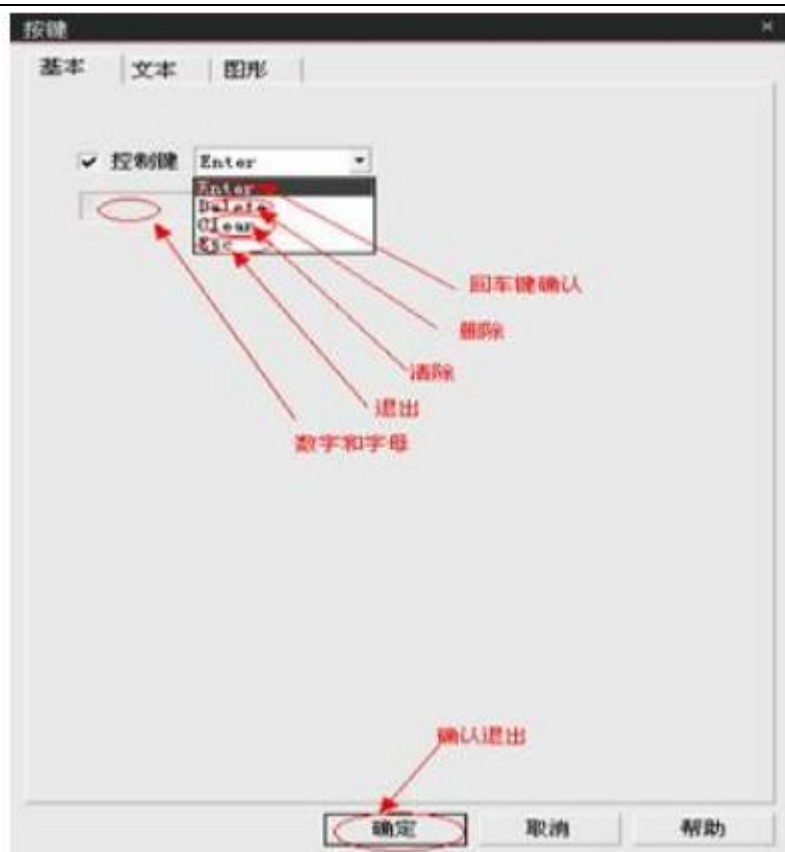


如下图所示，再添加一个按键，双击按键属性



如下图所示，单击控制键，有确认、删除、清除、退出等。





如图所示，在以同样的方式添加所有的按键



效果图如下



42.21 向量图的修改

----->:如下图所示，以位开关为例，有三种操作方式，

- 1 点击工具栏向量图图标
- 2 点击部件属性编辑框的向量图
- 3 在画面上选中部件-----双击



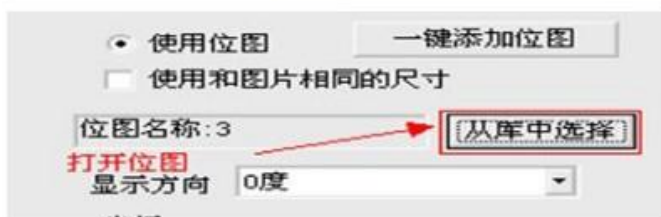
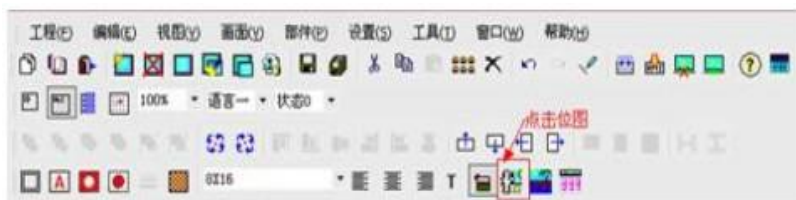
如下图所示，弹出外观选择对话框，选择自己喜欢的图片，可以修改边框的颜色，前景色，背景色，添加位图等，选好向量图单击确定。



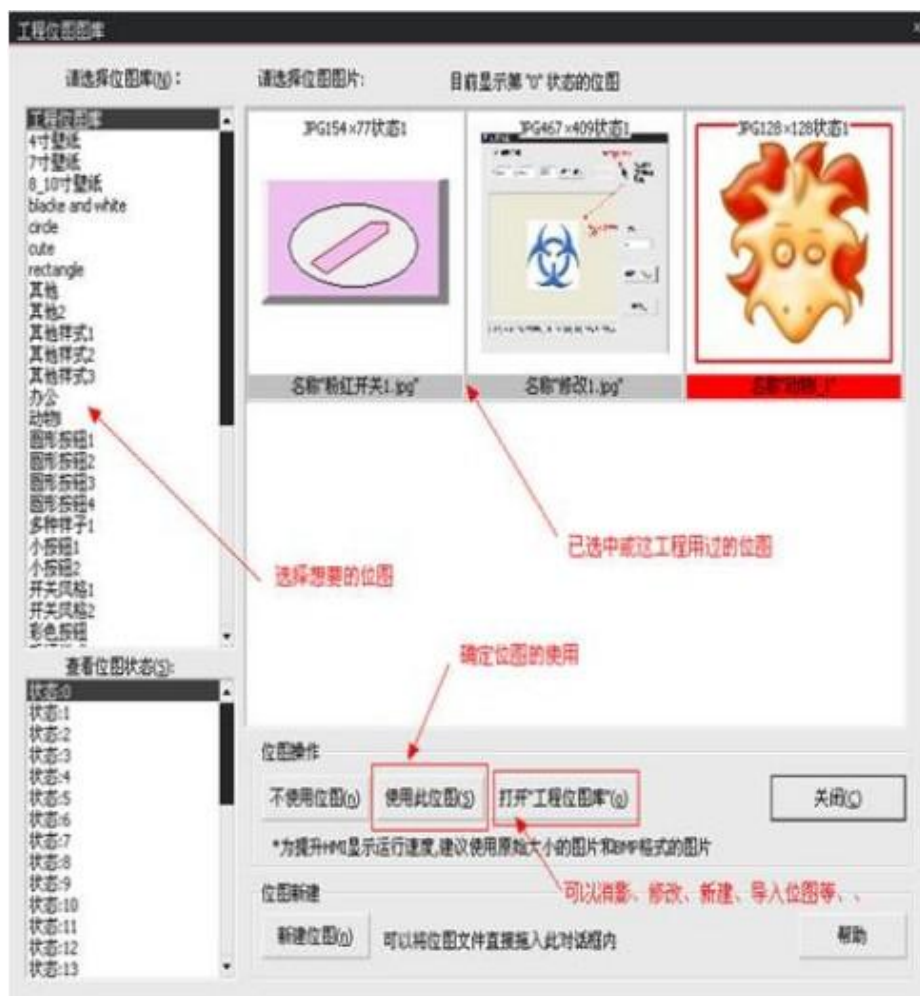
## 42.22 外观位图的修改

----->:如下图，有三种方式打开位图

- 1 点击工具栏位图
- 2 在部件属性编辑框选择外观位图
- 3 双击编辑区的位开关，点击图型-从库中选择



在工程位图图库中，可以选用图片，也可以导入、修改、消影（透明）图片等，选完点击-使用位图，



## 42.23 透明操作

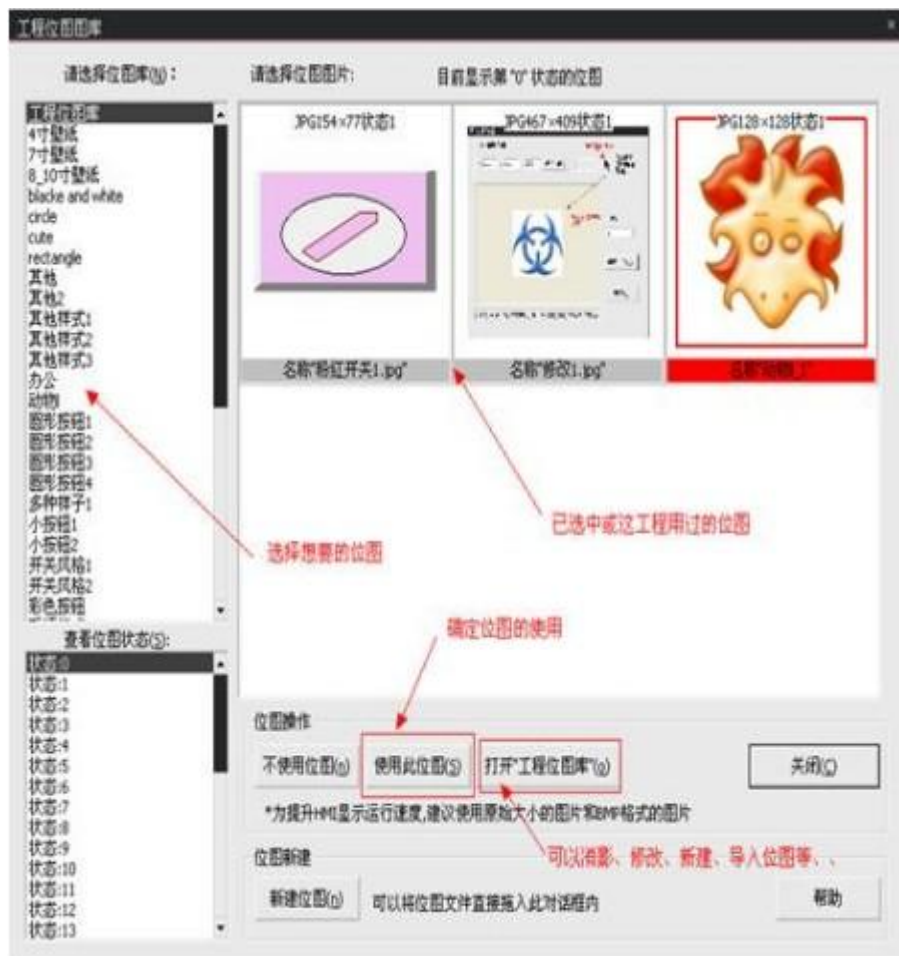
----->:1. 选中相应的部件-----双击



2. 在图形编辑区: 从库中选择



3. 在工程位图库：选择相应的位图-----“打开工程位图库”





## 4. 选择“位图编辑”



## 5. 位图透明操作，使用鼠标拖动操作。



## 42.24 字体的编辑

----->: 如下图，以位开关为例，有三种方式设置字体

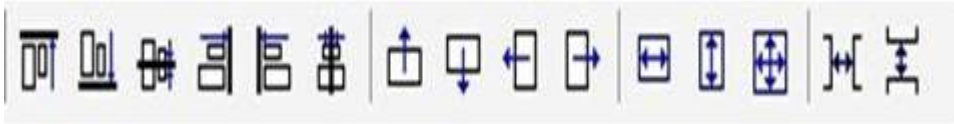
1. 工具栏上默认字体大小
2. 在部件属性编辑框-----字体
3. 在画面上双击需要添加文本的部件-----文本-----字体设置





## 42.25 部件对齐排版

- >:1. 选中相应的若干个部件
2. 在工具栏找到下图工具，更具需要排版相关部件。（下对齐、左对齐、等间距）



## 42.26 批量修改相关属性：

- >:1. 选中相应的若干个部件（按住 Shift，连续点击要选部件）
2. 在工具栏找到下图工具，根据需要排版相关部件。（外观图、字体、背景色）

