

## MODBUS 示例工程

| 类别  | 内容         |
|-----|------------|
| 关键词 | MODBUS 串口屏 |
| 摘要  |            |

## 修订历史

| 版本   | 日期         | 原因   | 编制  | 审查 |
|------|------------|------|-----|----|
| V1.0 | 2018/03/16 | 创建文档 | 刘启鑫 |    |
|      |            |      |     |    |

## 销售与服务

### 广州大彩光电科技有限公司

电 话: 020-82186683-601

传 真: 020-22059566

Email: hmi@gz-dc.com (公共服务)

网 站: www.gz-dc.com

地 址: 广州高新技术产业开发区玉树工业园 C 栋

官网零售淘宝店: <https://gz-dc.taobao.com>

## 目 录

|                             |    |
|-----------------------------|----|
| 1. 适用范围.....                | 5  |
| 2. 开发环境版本.....              | 6  |
| 3. MODBUS 示例工程概述 .....      | 7  |
| 3.1    串口屏协议说明.....         | 7  |
| 3.2    开发工程流程.....          | 7  |
| 3.3    MODBUS 协议运行说明.....   | 7  |
| 3.4    工程画面概览.....          | 7  |
| 3.5    协议与变量配置概览.....       | 8  |
| 4. 工程详解.....                | 10 |
| 4.1    创建工程.....            | 10 |
| 4.1.1    新建工程.....          | 10 |
| 4.1.2    新建画面.....          | 10 |
| 4.1.3    配置 modbus 协议 ..... | 10 |
| 4.2    画面与逻辑配置.....         | 11 |
| 4.2.1    首页.....            | 12 |
| 4.2.2    工程说明.....          | 13 |
| 4.2.3    线圈.....            | 13 |
| 4.2.4    离散输入.....          | 15 |
| 4.2.5    保持寄存器.....         | 17 |
| 4.2.6    输入寄存器.....         | 19 |
| 4.2.7    按钮控件.....          | 21 |
| 4.2.8    文本控件.....          | 22 |
| 4.2.9    进度条、仪表、滑块.....     | 24 |
| 4.2.10    RTC 控件 .....      | 25 |
| 4.2.11    动画控件.....         | 28 |
| 4.2.12    图标控件.....         | 30 |
| 4.2.13    曲线控件.....         | 32 |
| 4.2.14    菜单控件.....         | 34 |
| 4.2.15    选择控件.....         | 36 |
| 4.2.16    二维码.....          | 38 |
| 4.2.17    数据记录控件.....       | 40 |
| 4.2.18    历史曲线控件.....       | 44 |
| 4.2.19    隐藏显示控件.....       | 46 |
| 4.2.20    多语言.....          | 48 |

## 1. 适用范围

文档适合经济型、基本型、物联网、86 盒、485 系列等所有 MODBUS 协议的串口屏产品。

## 2. 开发环境版本

1、VisualTFT 软件版本：V3.0.0.827 及以上；

版本查看：

- (1) 打开软件,右下角显示的软件版本号。
- (2) 打开 VisualTFT，点击帮助->关于 VisualTFT 可以查看当前软件版本号。  
最新版本可登陆 [www.gz-dc.com](http://www.gz-dc.com) 进行下载



2、串口屏硬件版本：V2.22.1025.XXX 及以上。

版本查看：

- (1) 查看屏幕背面版本号贴纸。
- (2) 画面中创建一个文本控件，属性“输入方式”选择“系统变量”，属性“系统变量”选择“固件版本号”，编译工程下载到屏中查看。

### 3. MODBUS 示例工程概述

#### 3.1 串口屏协议说明

大彩所有系列串口屏除 NANO 系列外，均有两种协议版本：一种是大彩指令协议版本、一种是 MODBUS 协议版本，本示例工程主要用于演示如何开发 MODBUS 协议的工程。

#### 3.2 开发工程流程

串口屏工程的开发流程遵循以下流程：

- 1、新建工程
- 2、创建并配置画面和控件
- 3、配置 MODBUS 协议，主要是建立变量、逻辑处理、MiniC 脚本、Lua 脚本等；
- 4、控件属性中绑定在【协议与变量设置】中定义的变量。

MODBUS 协议配置主要涉及 VisualTFT 三个部分：

- (1) 菜单【工具】->【协议与变量设置】；
- (2) 菜单【工具】->【MiniC 脚本编程】；
- (3) 菜单【工具】->【Lua 脚本编程】，此功能只适用于物联网型系列串口屏，其它系列不支持。

#### 3.3 MODBUS 协议运行说明

开发的 MODBUS 工程，经过 VisualTFT 编译后会生成可以下载到串口屏中的工程包，编译的过程会将配置的 MODBUS 信息转换为串口屏工程文件；串口屏会根据工程文件运行，自动生成符合 MODBUS 协议的格式的指令。如果屏做主机，会周期性的通过串口发送这些指令到从机，接收从机返回的指令并解析，更新绑定的控件或执行预先配置的动作等。

#### 3.4 工程画面概览

本工程中画面功能介绍主要分三组，如图 3-1 所示：

- 1、区域 1：演示 MODBUS 协议的中的线圈、离散输入、保持寄存器、输入寄存器的数据显示，直接显示出数值；
- 2、区域 2：演示如何使用控件显示关联的变量，即 MODBUS 数据；
- 3、区域 3：演示如何实现显示隐藏控件、多语言控制等分组 1、分组 2 中未使用的功能；



图 3-1 工程画面

### 3.5 协议与变量配置概览

本工程协议与变量配置如图 3-2 所示。

- 1、区域 1：用于配置工程的协议类型，支持 MODBUS 主机、从机。
- 2、区域 2：用于建立变量，变量类型为 MODBUS 协议中线圈、离散输入、保持寄存器、输入寄存器，另外还支持内存变量、系统变量；
- 3、区域 3：用于配置逻辑设置；建立画面、控件与 MODBUS 变量之间的关系。

关于协议与变量配置的详细介绍可以参考大彩文档《大彩串口屏 MODBUS 通信》。

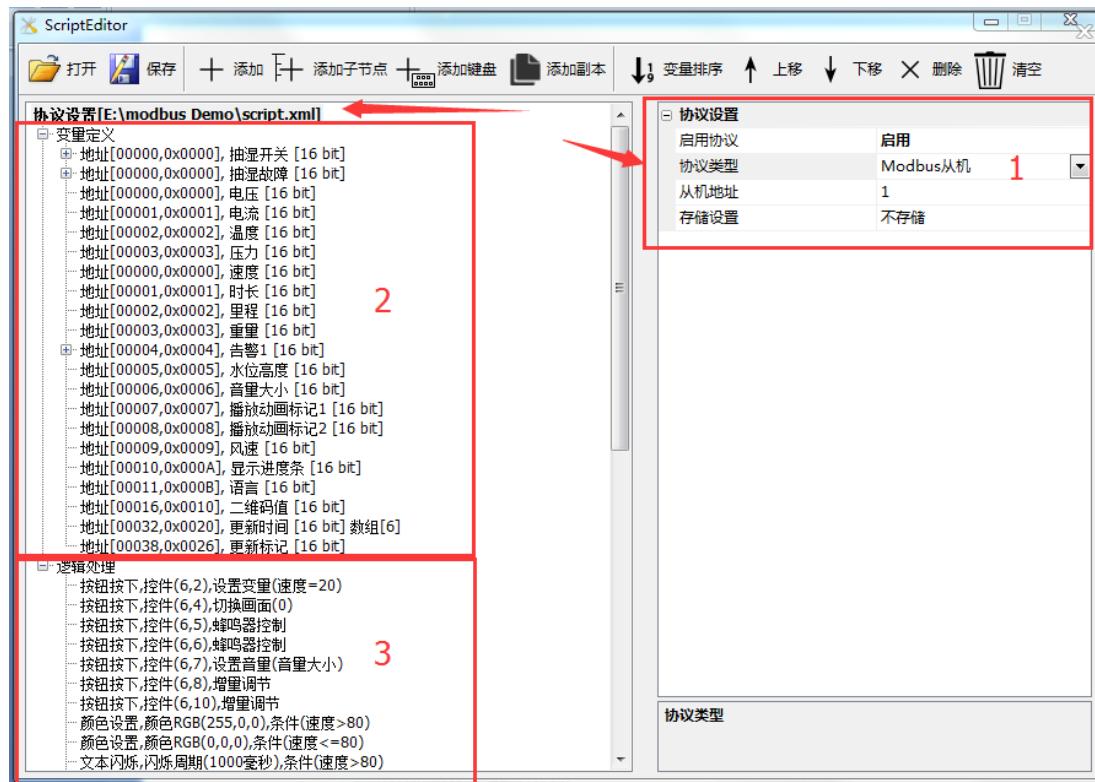


图 3-2 脚本与协议配置

## 4. 工程详解

### 4.1 创建工程

首先做好工程的准备工作，主要包括创建工程、建立页面、配置 MODBUS 协议类型等。

#### 4.1.1 新建工程

打开菜单【文件】->【新建工程】，弹出新建工程窗口，设置好工程名称、路径、设备类型，点击【确定】即可。



图 4-1 新建工程

#### 4.1.2 新建画面

新建的工程默认会创建一个画面，也可以自己添加画面。

#### 4.1.3 配置 modbus 协议

打开菜单【工具】->【协议与变量设置】，双击左侧文件路径，在右侧的协议设置处，设置协议类型，本处设置 MODBUS 从机。

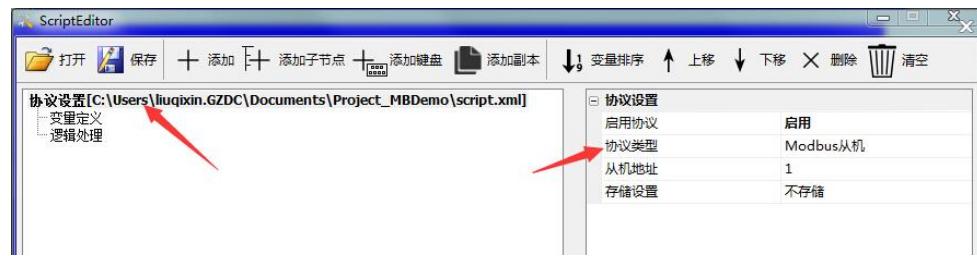


图 4-2 配置协议类型

至此，MODBUS 协议的工程雏形已经建立，后面即可根据实际需求配置工程画面和逻辑。

## 4.2 画面与逻辑配置

工程配置方式采用画面和 MODBUS 逻辑同时进行。

工程逻辑结构采用首页建立导航按钮，不同按钮切换到不同的画面，每个画面有返回首页按钮，以返回首页。

以下为对工程各个画面配置的介绍。

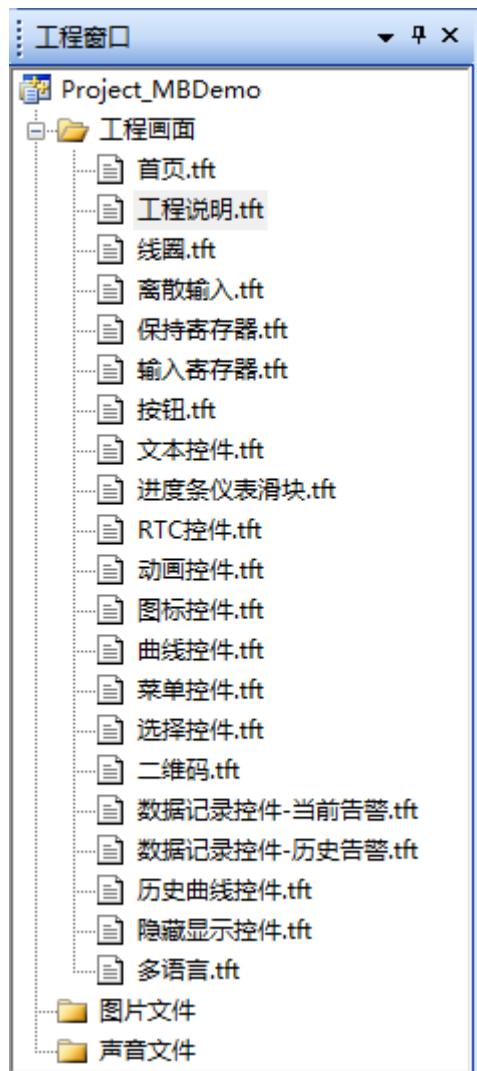


图 4-3 工程画面

#### 4.2.1 首页

【首页】画面中建立各个按钮，如图 4-4 所示，按钮可以配置切换到不同画面。



图 4-4 首页

例如：按钮 2 “工程说明”，可以按钮的属性用于切换到画面【工程说明】，按钮配置如图 4-5 所示，其它按钮同理。



图 4-5 按钮配置

#### 4.2.2 工程说明

【工程说明】画面为对本工程简单的说明，无其它功能。

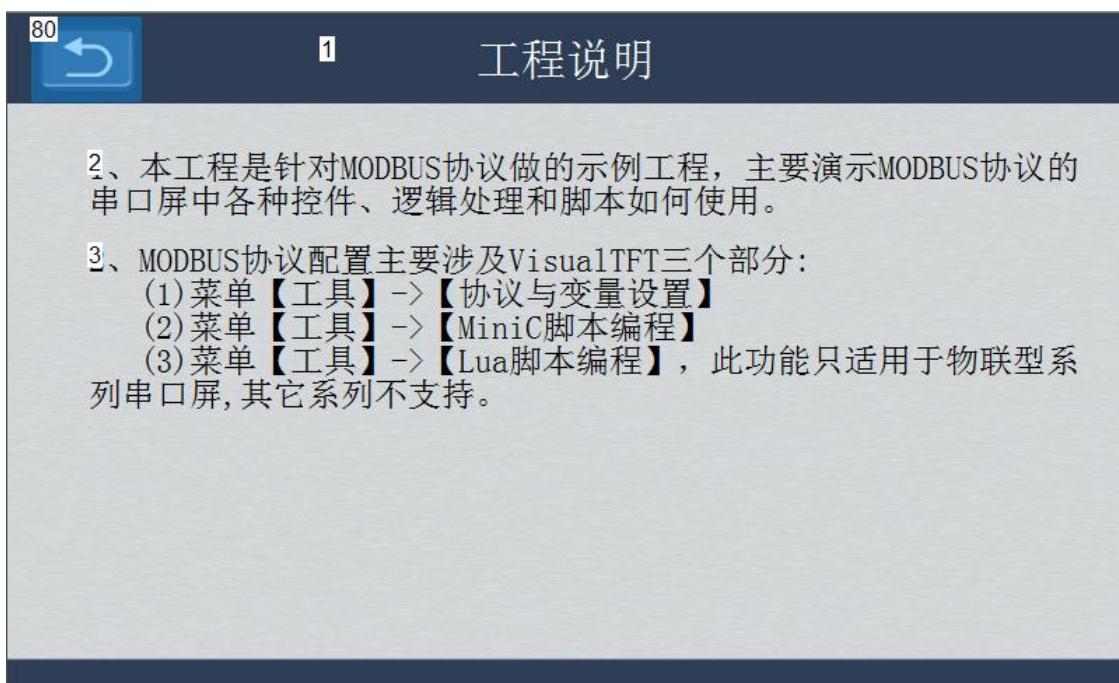


图 4-6 工程说明

#### 4.2.3 线圈

【线圈】画面是对 MODBUS 协议中“线圈”变量数值的直接显示，文本控件直接绑定“线圈”类型的变量，运行后，文本控件会显示变量的值，当该地址的“线圈”值发生变化后，文本控件同样也会发生变化。

配置方法：

1、添加变量：创建“抽湿开关”变量，添加子节点“开关机”，变量类型设置为“线圈”，配置变量地址为 0x0000，如图 4-8 所示；

2、画面中添加一个文本控件（ID：2），如图 4-7 所示，设置文本控件的属性，设置关联变量“抽湿开关/开关机”，如图 4-9 所示；

通过以上两步，ID 为 2 的文本控件与 0x0000 地址的“抽湿开关/开关变”便建立了关联。此时，编译工程并运行串口屏，文本控件便显示 0x0000 地址的线圈的数值。

同理，ID 为 3、4、5 的文本控件通过以上方法分别与不同的变量关联后，便可显示相应地址的变量值。

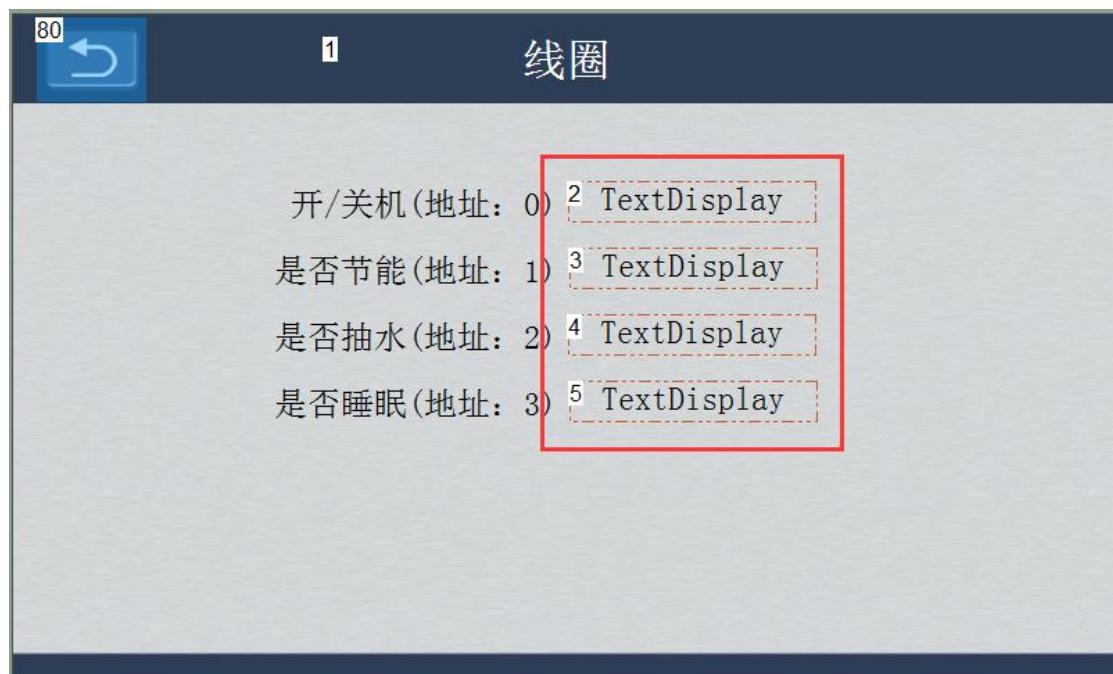


图 4-7 线圈

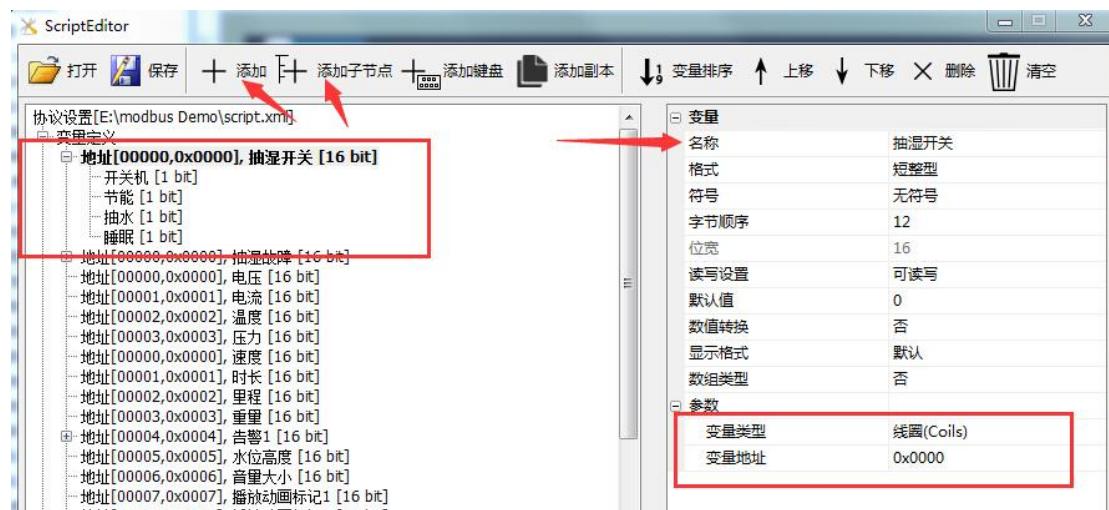


图 4-8 新建线圈变量

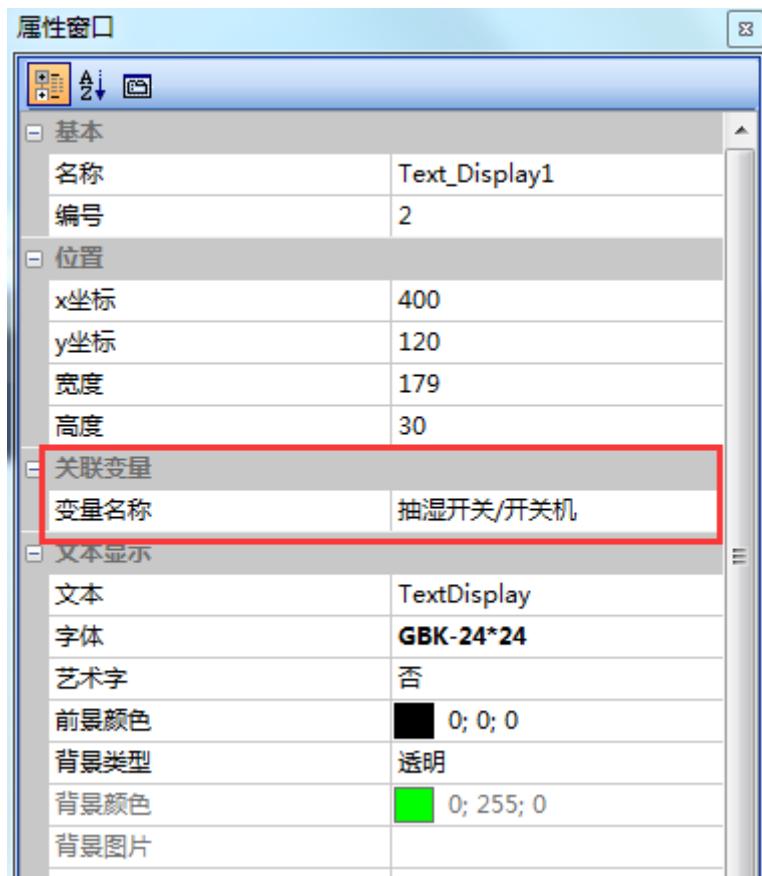


图 4-9 关联变量

#### 4.2.4 离散输入

【离散输入】画面是对 MODBUS 协议中“离散输入”变量数值的直接显示，文本控件直接绑定“离散输入”类型的变量，运行后，文本控件会显示变量的值，当该地址的“离散输入”值发生变化后，文本控件同样也会发生变化。

方法：

- 1、添加变量：创建“抽湿故障”变量，添加子节点“故障 0”，变量类型设置为“离散

输入”，配置变量地址为 0x0000，如图 4-11 所示；

2、画面中添加一个文本控件（ID：2），如图 4-10 所示，设置文本控件的属性，设置关联变量“抽湿故障/故障 0”，如图 4-12 所示；

通过以上两步，ID 为 2 的文本控件与 0x0000 地址的“抽湿故障/故障 0”便建立起了关联。此时，编译工程并运行虚拟串口屏，文本控件便显示 0x0000 地址的离散输入的数值。

同理，ID 为 3、4、5 的文本控件通过以上方法分别与不同的变量关联后，便可显示相应地址的变量值。



图 4-10 离散输入

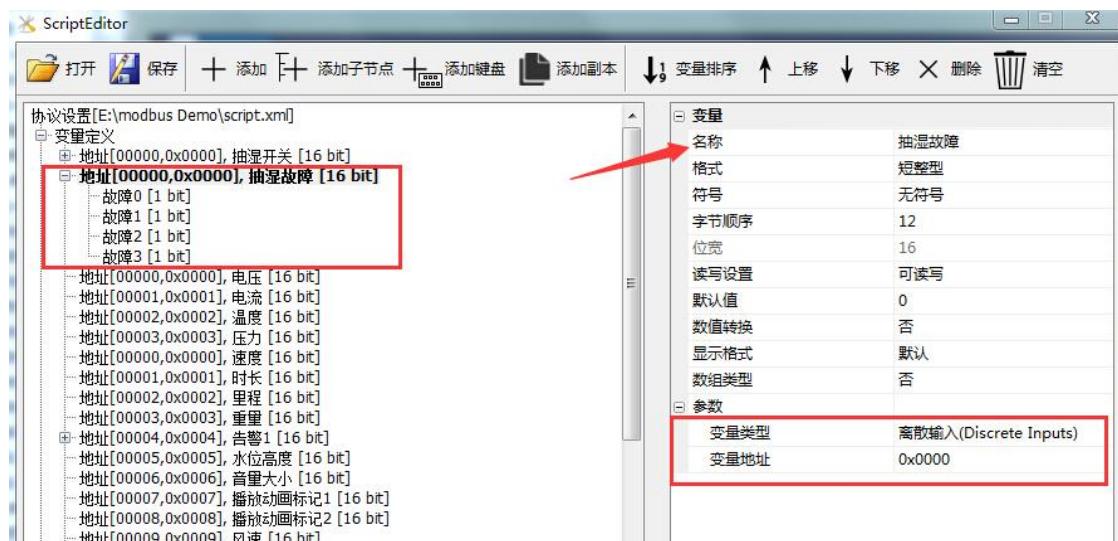


图 4-11 新建离散输入



图 4-12 关联变量

#### 4.2.5 保持寄存器

【保持寄存器】画面是对 MODBUS 协议中“保持寄存器”变量数值的直接显示，文本控件直接绑定“保持寄存器”类型的变量，运行后，文本控件会显示变量的值，当该地址的“保持寄存器”值发生变化后，文本控件同样也会发生变化。

方法：

1、添加变量：创建“速度”变量，变量类型设置为“保持寄存器”，配置变量地址为 0x0000，如图 4-14 所示；

2、画面中添加一个文本控件（ID：2），如图 4-13 所示，设置文本控件的属性，设置关联变量“速度”，如图 4-15 所示；

通过以上两步，ID 为 2 的文本控件与 0x0000 地址的“速度”便建立了关联。此时，编译工程并运行虚拟串口屏，文本控件便显示 0x0000 地址的保持寄存器的数值。

同理，ID 为 3、4、5 的文本控件通过以上方法分别与不同的变量关联后，便可显示相应地址的变量值。

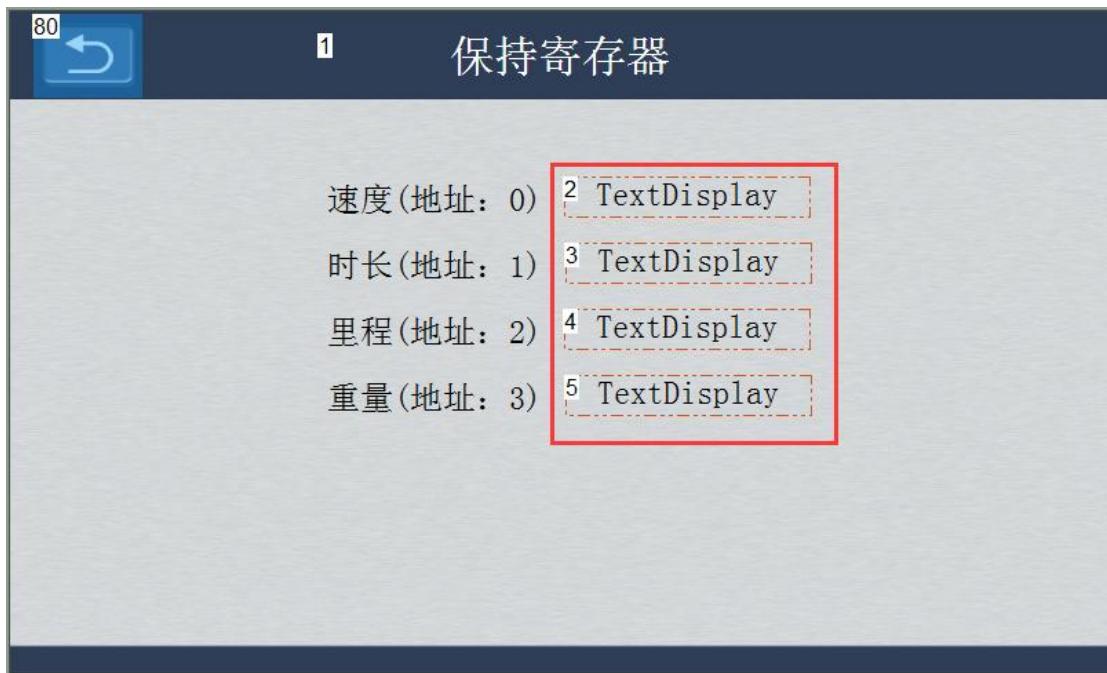


图 4-13 保持寄存器

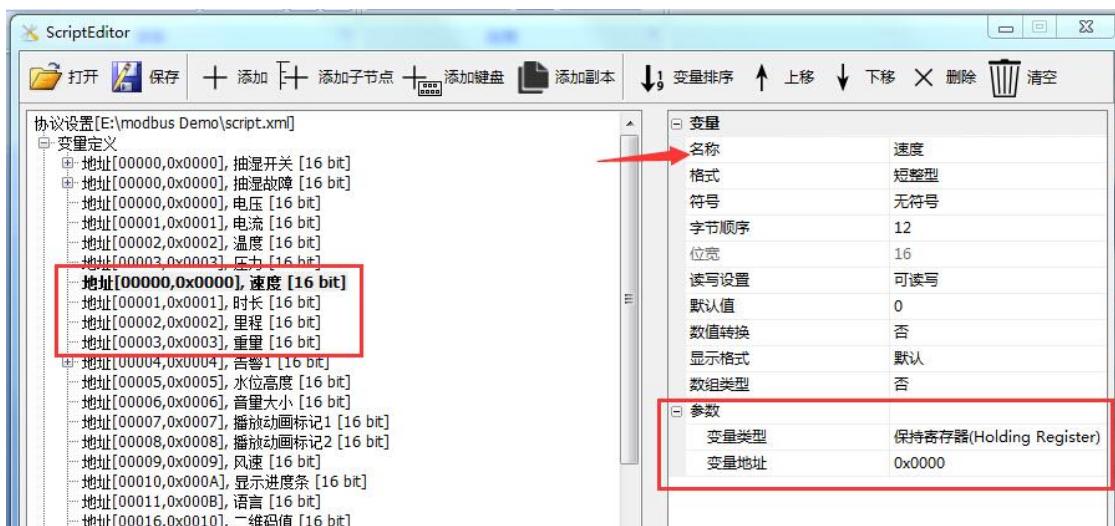


图 4-14 新建保持寄存器



图 4-15 关联变量

#### 4.2.6 输入寄存器

【输入寄存器】画面是对 MODBUS 协议中“输入寄存器”变量数值的直接显示，文本控件直接绑定“输入寄存器”类型的变量，运行后，文本控件会显示变量的值，当该地址的“输入寄存器”值发生变化后，文本控件同样也会发生变化。

方法：

1、添加变量：创建“电压”变量，变量类型设置为“输入寄存器”，配置变量地址为 0x0000，如图 4-17 所示；

2、画面中添加一个文本控件（ID: 2），如图 4-16 所示，设置文本控件的属性，设置关联变量“电压”，如图 4-18 所示；

通过以上两步，ID 为 2 的文本控件与 0x0000 地址的“电压”便建立了关联。此时，编译工程并运行虚拟串口屏，文本控件便显示 0x0000 地址的保持寄存器的数值。

同理，ID 为 3、4、5 的文本控件通过以上方法分别与不同的变量关联后，便可显示相应地址的变量值。



图 4-16 输入寄存器

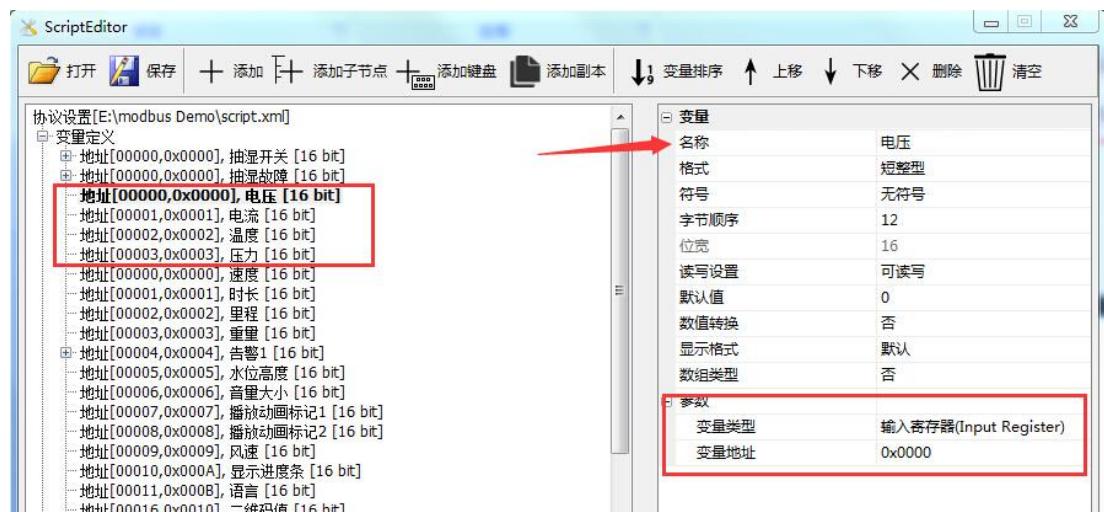


图 4-17 新建输入寄存器



图 4-18 关联变量

#### 4.2.7 按钮控件

【按钮控件】画面主要对按钮控件的使用做示例说明。MODBUS 协议版本中，按钮按下或弹起时可以执行一些预设的逻辑操作。

VisualTFT 的【资源窗口】可以看到【按钮控件】的画面 ID 为 6，如图 4-19 所示，按钮“设置速度为 20”的控件 ID 为 2；在【协议与变量设置】中添加一条逻辑处理，该逻辑处理的具体设置如图 4-20 所示。此时，编译工程并运行虚拟串口屏，当按下“设置速度为 20”按钮，则变量“速度”的值被设置为 20。ID 为 3 的文本控件关联了变量“速度”，用于观察变量“速度”的值。

同样，按钮按下时还可以用于切换画面、设置蜂鸣器、增量调节等功能。



图 4-19 按钮控件

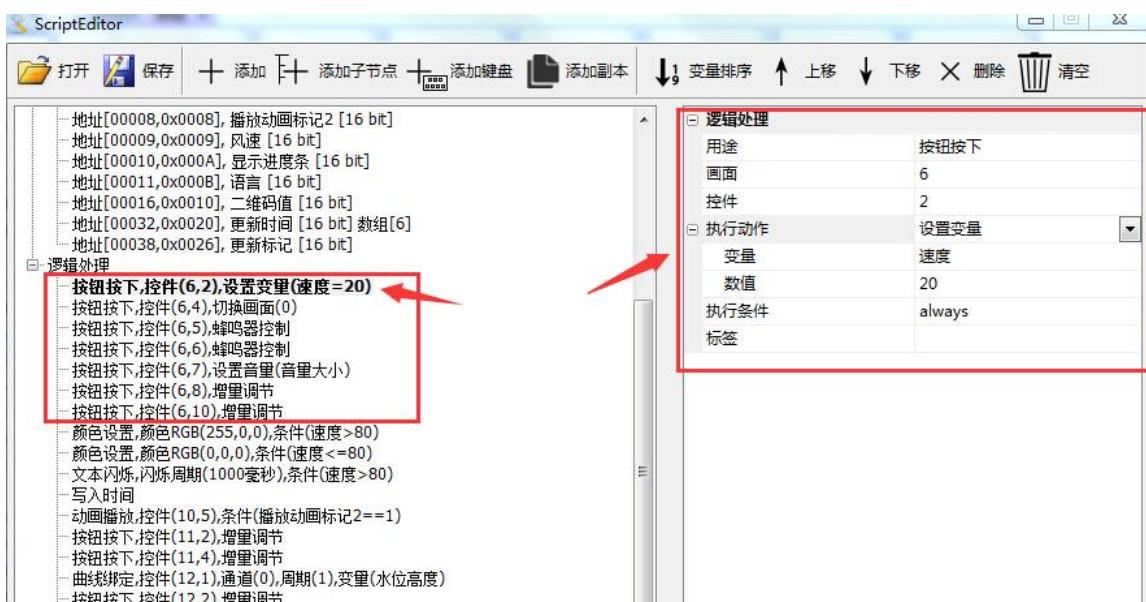


图 4-20 按钮逻辑处理

#### 4.2.8 文本控件

【文本控件】画面主要对文本控件的使用做示例说明。MODBUS 协议版本中，文本控件主要是用于显示绑定的变量值，对于线圈、保持寄存器等类型的可修改变量，可以设置“输入方式”为弹出系统键盘输入，通过弹出的键盘对绑定的变量值进行修改。另外可以绑定修改显示颜色、闪烁的逻辑处理。

【文本控件】画面 ID 为 2 的文本控件，如图 4-21 所示，在其属性中绑定变量“速度”，如图 4-22 所示；在【协议与变量设置】中添加 3 条逻辑处理：第一条，用途为“设置颜色”，

颜色设置为红色 (RGB (255,0,0)), 执行条件为“速度”“大于”“80”; 第二条, 用途为“设置颜色”, 颜色设置为黑色 (RGB (0,0,0)), 执行条件为“速度”“小于等于”“80”; 第三条, 用途为“文本闪烁”, 执行条件为“速度”“大于”“80”; 如图 4-23 所示。

此时, 编译工程并运行虚拟串口屏, 当“速度”的值大于 80 时, 文本显示为红色并闪烁, 如果“速度”的值小于等于 80 时, 文本显示为黑色, 并且不闪烁。

同样, ID 为 3、4、5 的文本控件均可做此操作。



图 4-21 文本控件



图 4-22 关联变量

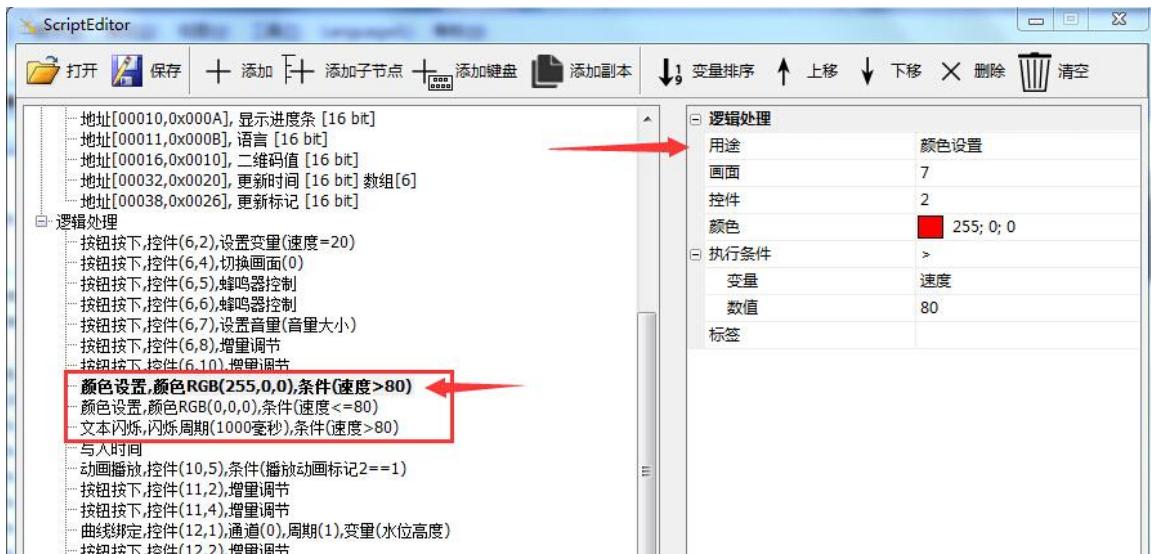


图 4-23 文本控件逻辑处理

#### 4.2.9 进度条、仪表、滑块

【进度条仪表滑块】画面主要对进度条、仪表、滑块控件的使用做示例说明。MODBUS 协议版本中，进度条、仪表、滑块控件主要是用于显示绑定的变量值；另外，滑块可以通过拖动修改绑定的变量值。

【进度条仪表滑块】画面中创建的文本控件、进度条、滑块、仪表控件，如图 4-24 所

示；这几个控件均绑定变量“速度”，如图 4-25 所示，为进度条关联变量，其它几个控件类似；其中文本控件主要用于观察值的大小，也可以通过其弹出的键盘修改对应值。

此时，编译工程并运行虚拟串口屏，当“速度”的值发生变化时，文本控件、进度条、滑块、仪表均会发生变化，更新显示的速度值；另外，当拖动滑块时，其它控件的也会更新显示速度值。



图 4-24 进度条、仪表、滑块

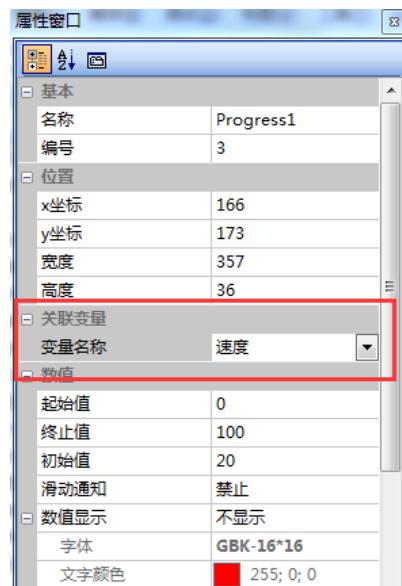


图 4-25 关联变量

#### 4.2.10 RTC 控件

【RTC 控件】画面主要对 RTC 控件的使用及更新 RTC 时间做示例说明，如图 4-26 所示。RTC 控件支持不同的属性格式，如所示图 4-27；也可以用户自定义显示格式，如图 4-28

所示。

MODBUS 协议版本串口屏中，更新 RTC 时间常见有两种应用：

#### 1、设置本机 RTC 时间

通常有两种方法：1、直接设置 RTC 控件属性“允许编辑”为“编辑日期和时间”、“编辑日期”或者“编辑时间”；如此，当长按 RTC 控件或着点击 RTC 控件时即可弹出键盘以修改 RTC 时间。2、使用 set\_date() 和 set\_time() 设置 RTC 时间，具体可以参照图 4-26 中，设置本机 RTC 时间的说明和图 4-30 的 MiniC 代码。

#### 2、屏做主机时，修改从机 RTC 时间；

通常做法是：从机创建固定地址的保持寄存器，用于存放时间（连续 7 个保持寄存器，分别存放年、月、星期、日、时、分、秒），主机周期性的把最新的时间发送给从机。逻辑处理中有“写入时间”的功能，可以实现此操作。如图 4-29 中红色框所示，建立“写入时间”的逻辑处理，具体设置如图中所示，其中“地址”为从机中保存时间的地址，“周期”为主机每隔多长时间写入一次；可以根据自己需求建立条件执行。

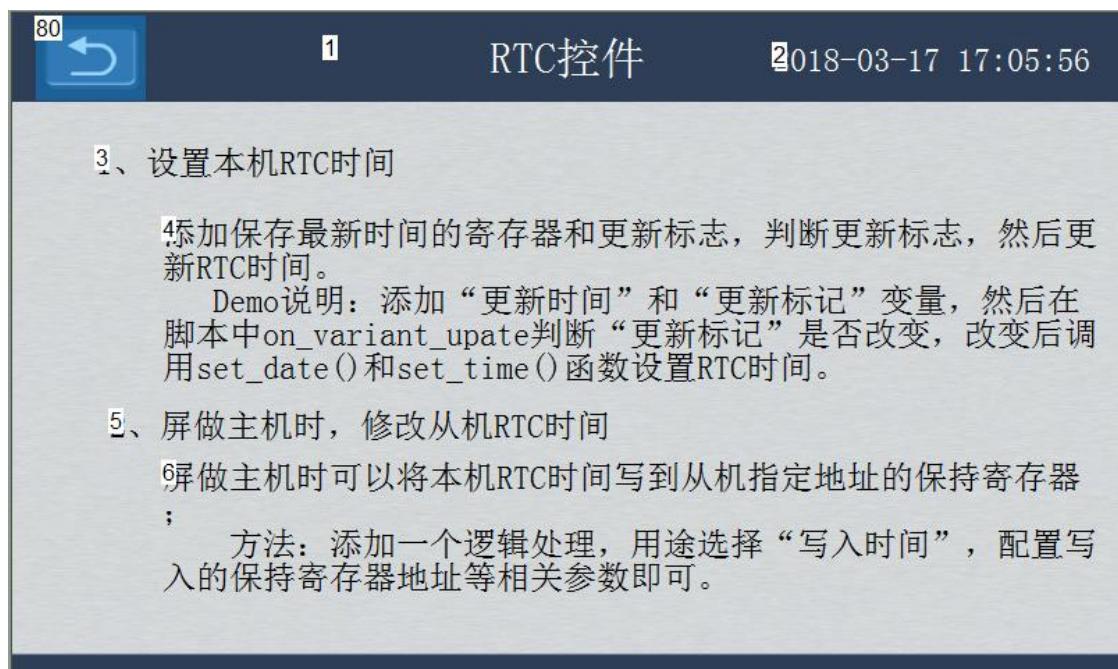


图 4-26 RTC 控件

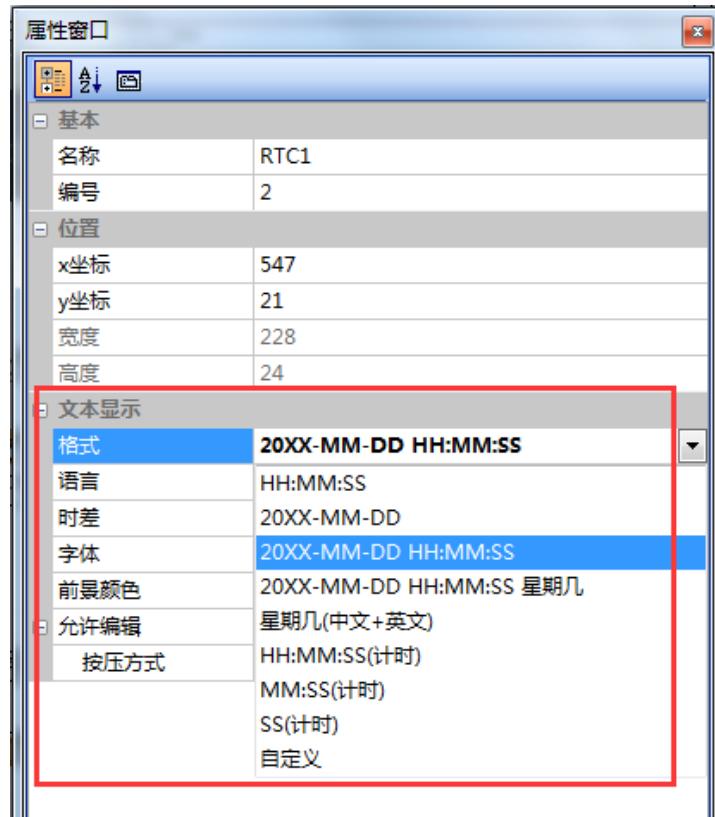


图 4-27 RTC 控件属性

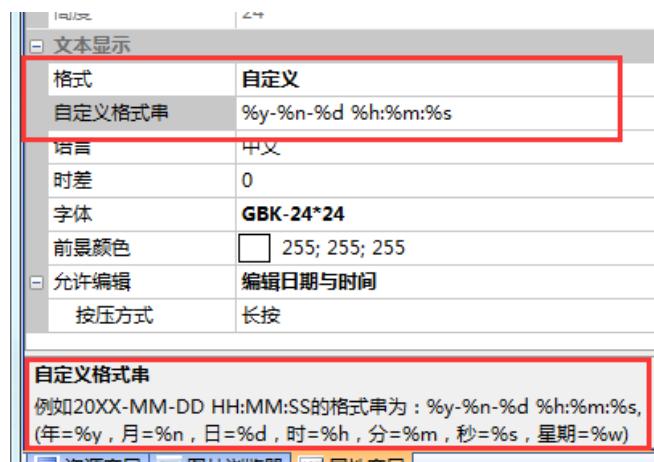


图 4-28 RTC 自定义

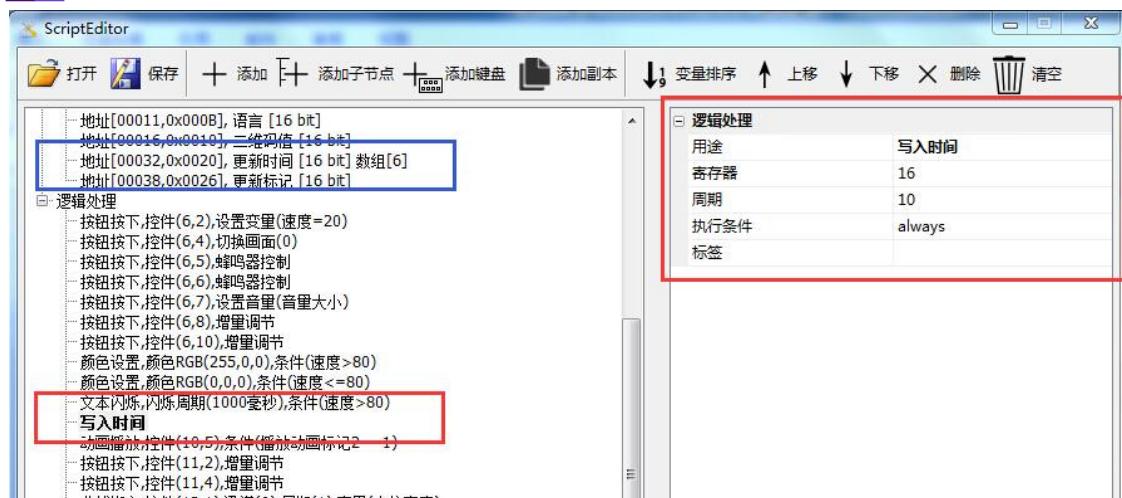


图 4-29 更新 RTC 的寄存器、逻辑处理

```

47  /*
48  函数: on_variant_update
49  功能: 串口通信导致变量更新时, 执行此函数
50 */
51 void on_variant_update()
52 {
53     //操作符'@'用于判定某个寄存器是否发生改变
54     //更新RTC时间
55     if(@"更新标记")
56     {
57         set_date("更新时间"[0], "更新时间"[1], "更新时间"[2]);
58         set_time("更新时间"[3], "更新时间"[4], "更新时间"[5]);
59     }
60
61
62     //“语言”发生变化时, 修改设置系统语言
63     if(@"语言")
64     {
65         sys.lang = "语言";
66     }
67 }
68

```

图 4-30 更改本机 RTC 时间

#### 4.2.11 动画控件

【动画控件】画面主要对动画控件的使用做示例说明, 如图 4-31 所示。动画控件用于播放设置的 GIF 或 ICON 格式的动画文件, 可以控制暂停或者播放。

MODBUS 协议中有两种控制动画控件的方式:

##### 1、绑定变量方式

动画控件直接关联变量, 当变量的值为 0 时, 动画停止; 当变量的值为 1 时, 动画播放。

【动画控件】画面中的 ID 为 2 的动画控件, 设置关联变量“播放动画标记 1”, 如图 4-32 所示。此时, 编译工程并运行虚拟串口屏, 当变量“播放动画标记 1”为 0 时动画停止, 当变量值为 1 时, 动画开始播放。

##### 2、逻辑处理方式

逻辑处理中有“动画播放”功能, 当预设的条件满足时, 可以控制动画播放或停止。

【动画控件】画面中的 ID 为 5 的动画控件，创建了“动画播放”的逻辑处理，如图 4-33 所示，设置的执行条件为变量“播放动画标记 2”“==”“1”时，ID 为 5 的动画会自动播放。

此时，编译工程并运行虚拟串口屏，当变量“播放动画标记 2”为 0 时动画停止，当变量值为 1 时，动画开始播放。

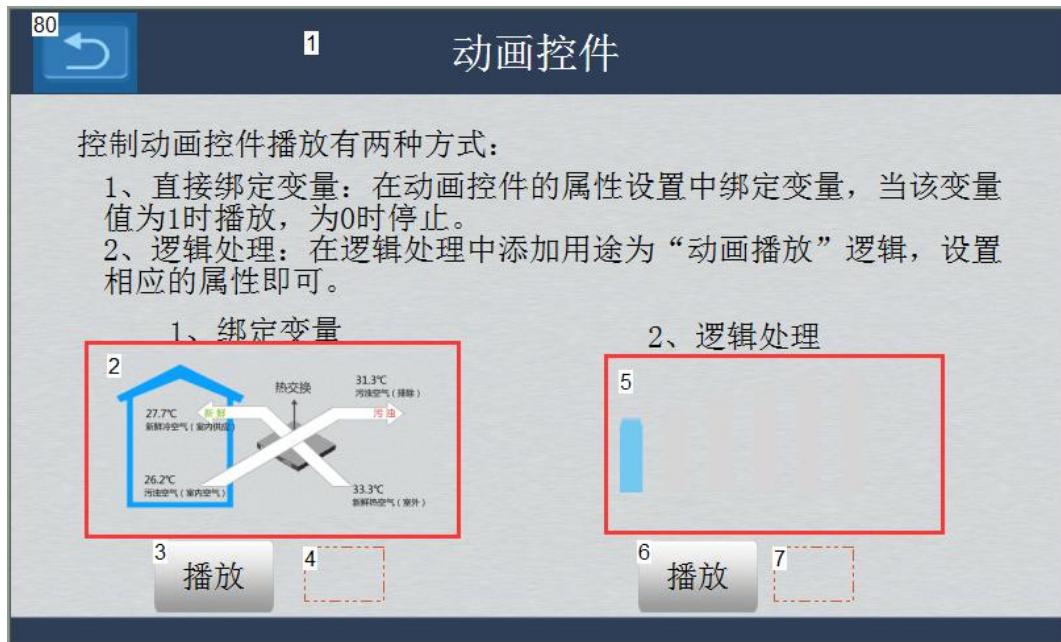


图 4-31 动画控件

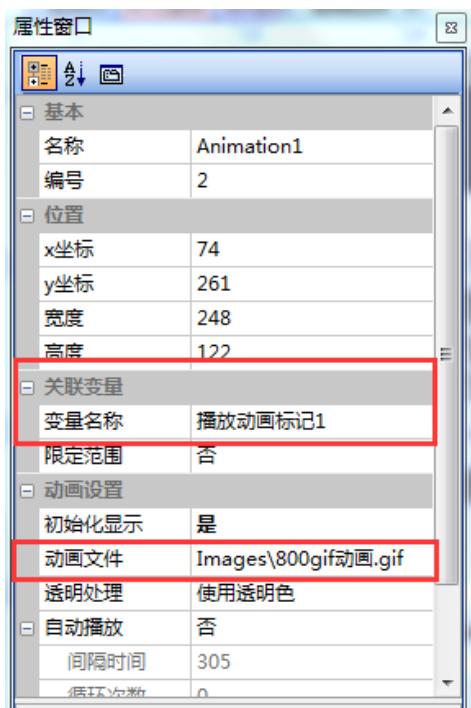


图 4-32 关联变量

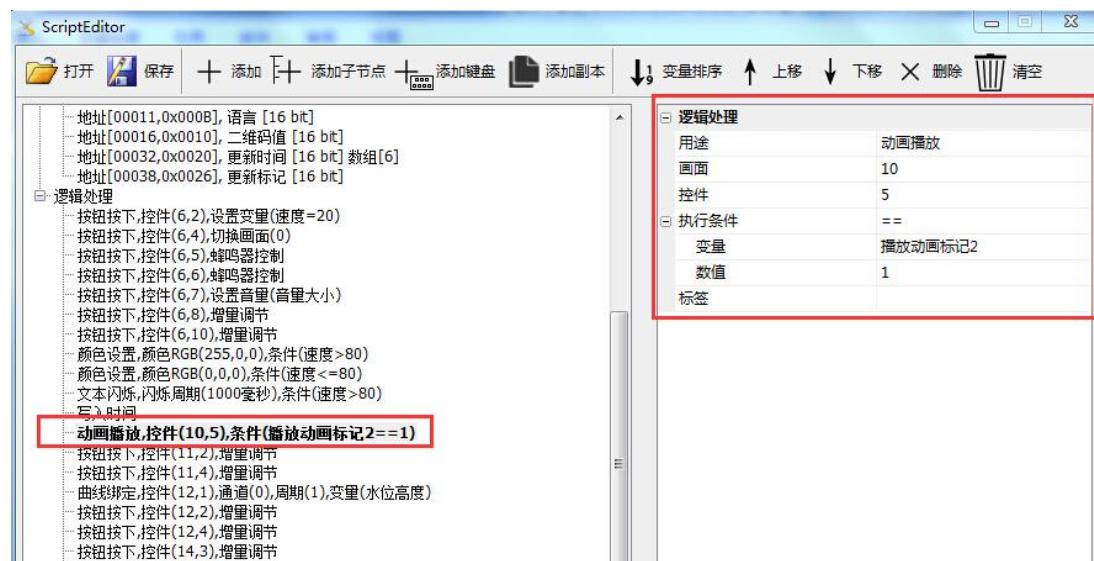


图 4-33 动画逻辑处理

#### 4.2.12 图标控件

【图标控件】画面主要对图标控件的使用做示例说明，如图 4-34 所示。图标控件用于显示设置的 GIF 或 ICON 格式的动画文件，通过显示不同的图片表示不同的状态，可以控制显示第几帧图片。

MODBUS 协议中可以通过关联变量的方式控制显示第几帧图片，关联变量的变量值默认对应图标的帧，索引从 0 开始，值为 0 时显示第 0 帧，值为 1 时显示第 1 帧，以此类推；如果设置了限定范围，则变量下限对应设置的图标下限，变量上限对应图标上限。

举例：【图标控件】画面中 ID 为 1 的图标控件关联了变量“风速”，如图 4-35 所示。此时，编译工程并运行虚拟串口屏，当变量“风速”为 0 时，图标控件中显示的风速为 1 格；当变量“风速”为 1 时，图标控件中显示的风速为 2 格；当变量“风速”为 2 时，图标控件中显示的风速为 3 格；以此类推。

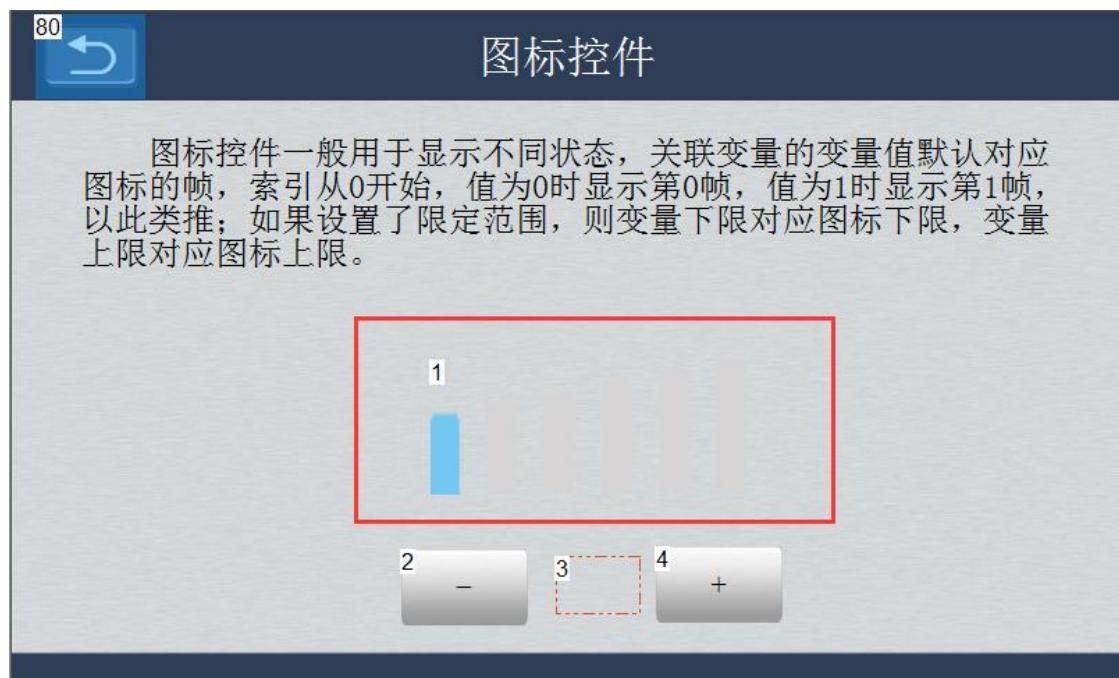


图 4-34 图标控件



图 4-35 关联变量

#### 4.2.13 曲线控件

【曲线控件】画面主要对曲线控件的使用做示例说明，如图 4-36 所示。MODBUS 协议版本中，曲线控件用于显示关联的变量值，曲线控件周期性的采样变量值，然后将所有采样值绘制到曲线控件内，连接形成曲线；曲线控件的每个通道都可以关联一个变量，最多可以设置 8 个通道。

MODBUS 协议版本串口屏中，曲线通道显示变量值，通过设置“曲线绑定”用途的逻辑处理使用。

【曲线控件】画面中 ID 为 1 的曲线控件，预设一个通道，为该控件的通道 0，添加关联的逻辑处理，参数如图 4-37 所示，每“1”秒采样变量一次“水位高度”的值。画面 ID 为 2、4 的按钮用于调整“水位高度”值，ID 为 3 的文本控件实时显示“水位高度”值；经过动态的调整“水位高度”，可以形成如图 4-38 所示的曲线控件图。

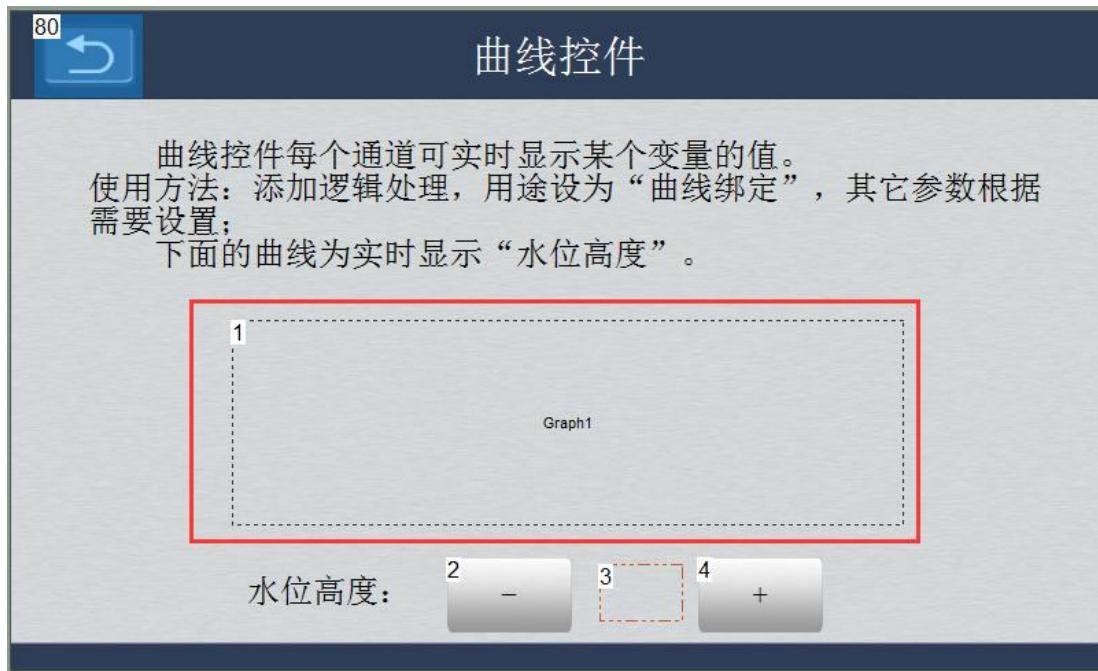


图 4-36 曲线控件

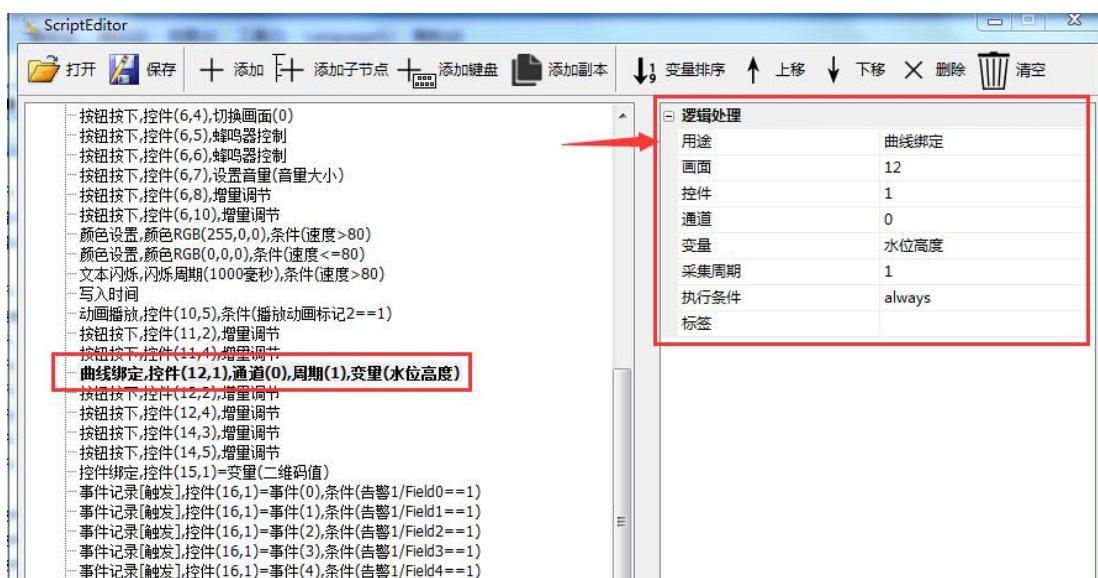


图 4-37 曲线绑定

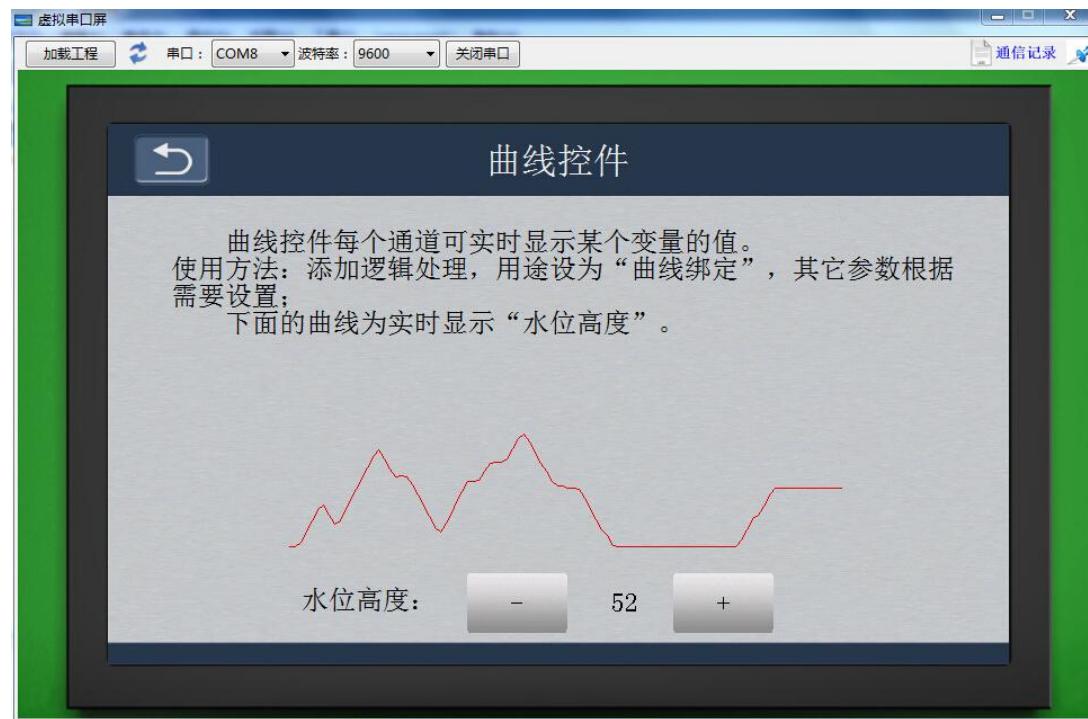


图 4-38 曲线控件运行示例

#### 4.2.14 菜单控件

【菜单控件】画面主要对菜单控件的使用做示例说明，菜单的“菜单风格”有两种：“弹出菜单”和“固定菜单”，此处主要做弹出菜单使用说明。

“弹出菜单”需要按钮控件将菜单和文本控件关联使用，点击按钮弹出菜单，选中菜单项后直接显示在文本控件中。

【菜单控件】画面如图 4-39 所示，创建 ID 为 1 的文本控件，ID 为 2 的按钮控件和 ID 为 3 的菜单控件；菜单控件的属性设置如图 4-40 所示，关联变量“风速”，菜单风格为“弹出菜单”，菜单选项为“一级；二级；三级；四级；五级；六级”，索引分别为 0、1、2、3、4、5；按钮设置如图 4-41 所示，触控用途设置为“弹出菜单”，“菜单控件 ID”设置为“3”，“输入控件 ID”设置为“1”；此时，编译工程并运行虚拟串口屏，当点击文本控件时，会弹出菜单，当点击菜单中选项后，文本控件的内容变为选中的菜单选项，同时，变量“风速”也被设置为选项对应的索引值；如果不设置变量，则只有文本控件的值发生变化。

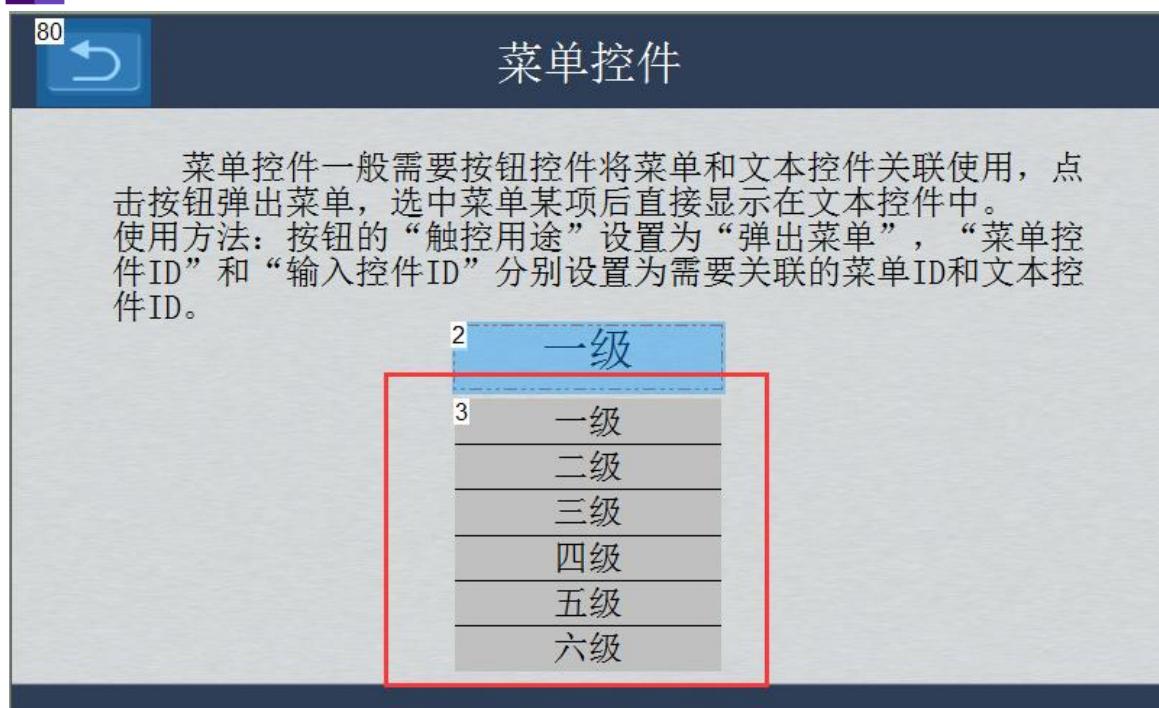


图 4-39 菜单控件



图 4-40 菜单设置



图 4-41 按钮配置

#### 4.2.15 选择控件

【选择控件】画面主要对选择控件的使用做示例说明。选择控件主要有两种用途：滑动选择和状态显示。“滑动选择”主要用于通过触摸屏滑动选择预设的选项；“状态显示”主要用于显示不同的预设选项，类似图标控件。

【选择控件】画面中 ID 为 1 的选择控件为“滑动选择”方式，ID 为 2 的选择控件为“状态显示”方式。

##### 1、滑动选择

ID 为 1 的选择控件，属性设置如图 4-43 所示，关联变量“风速”，用途设置为“滑动选择”，设置两端缩放和候选项，设置预设的数据选项“一级；二级；三级；四级；五级；六级；”；

此时，编译工程并运行虚拟串口屏，可以通过鼠标点中选择控件并拖动（模拟滑动选择），当拖动停止后，选中某个选项，同时变量“风速”的值也会变化更新。

##### 2、状态选择

ID 为 2 的选择控件，属性设置如图 4-44 所示，关联变量“风速”，用途设置为“状态显示”，设置预设的数据选项“一级；二级；三级；四级；五级；六级；”；

此时，编译工程并运行虚拟串口屏，当变量“风速”的值发生变化时，控件 2 显示的状态值也会相应的更新。

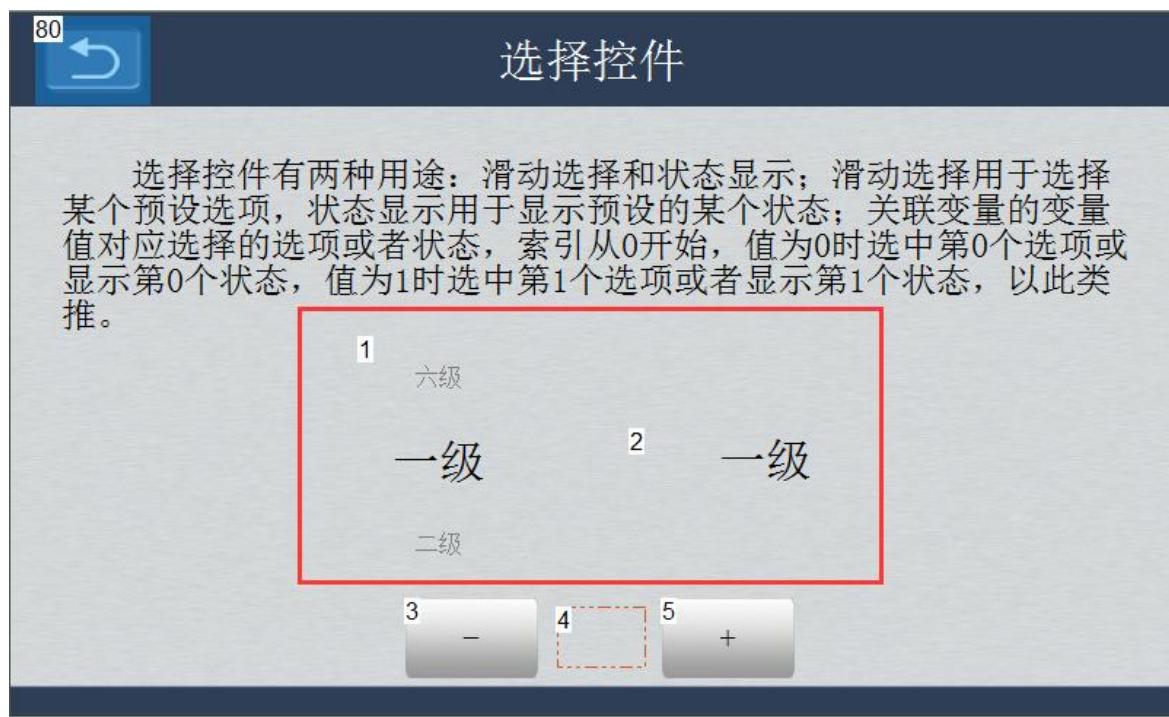


图 4-42 选择控件



图 4-43 滑动选择



图 4-44 状态显示

#### 4.2.16 二维码

【二维码】画面主要对二维码控件的使用做示例说明。二维码控件可以将设定的字符串以二维码的形式显示出来。MODBUS 协议版本的串口屏，二维码控件显示关联的字符串类型变量。注意，二维码控件需要特殊的固件支持。

【二维码】画面中 ID 为 1 的二维码控件属性具体设置如图 4-46 所示；“文本”即为默认显示的字符串，“放大系数”即为单个点的大小，“错觉纠正”该级别越高，可以显示的字符串长度越小，“编码方式”即二维码的编码方式。

创建“二维码值”变量，变量类型设置为“字符串类型”，具体设置如图 4-47 所示，创建“控件绑定”的逻辑处理，将控件与变量“二维码值”相关联。

此时，编译工程并运行虚拟串口屏，二维码控件将显示绑定变量“二维码值”的二维码，当变量“二维码值”发生变化时，二维码控件会自动更新，生成新的二维码。

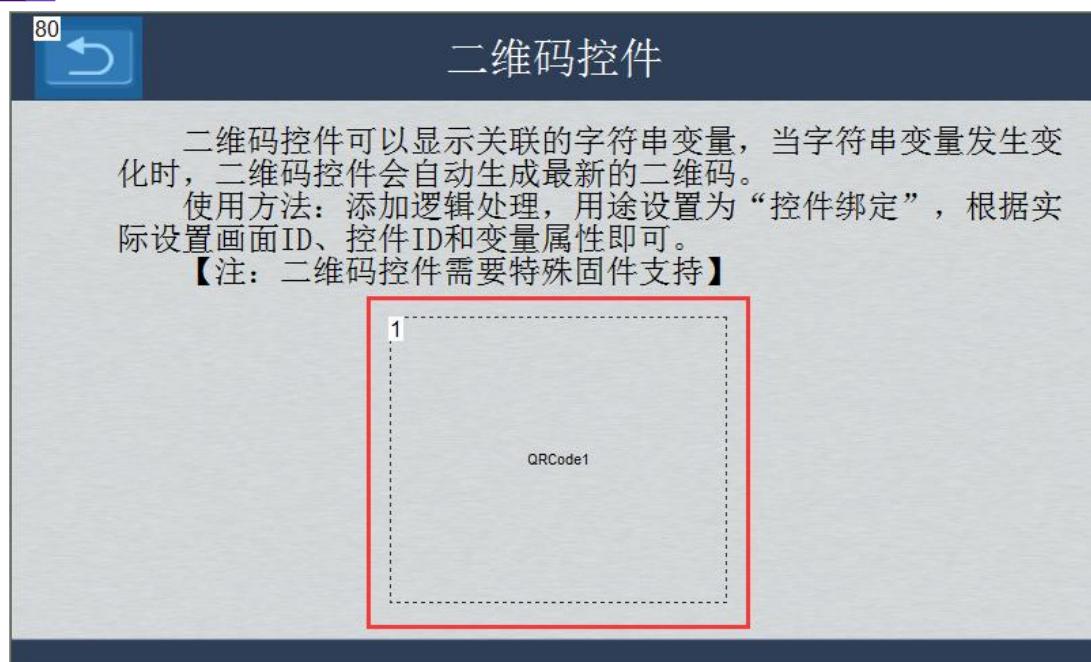


图 4-45 二维码控件

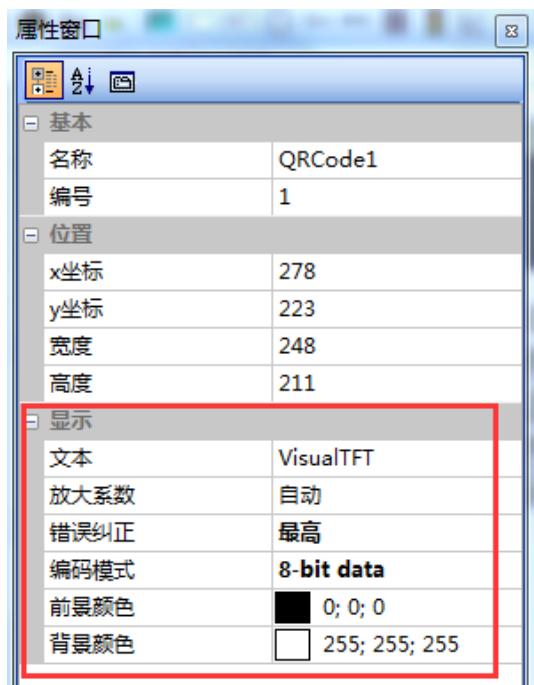


图 4-46 二维码属性

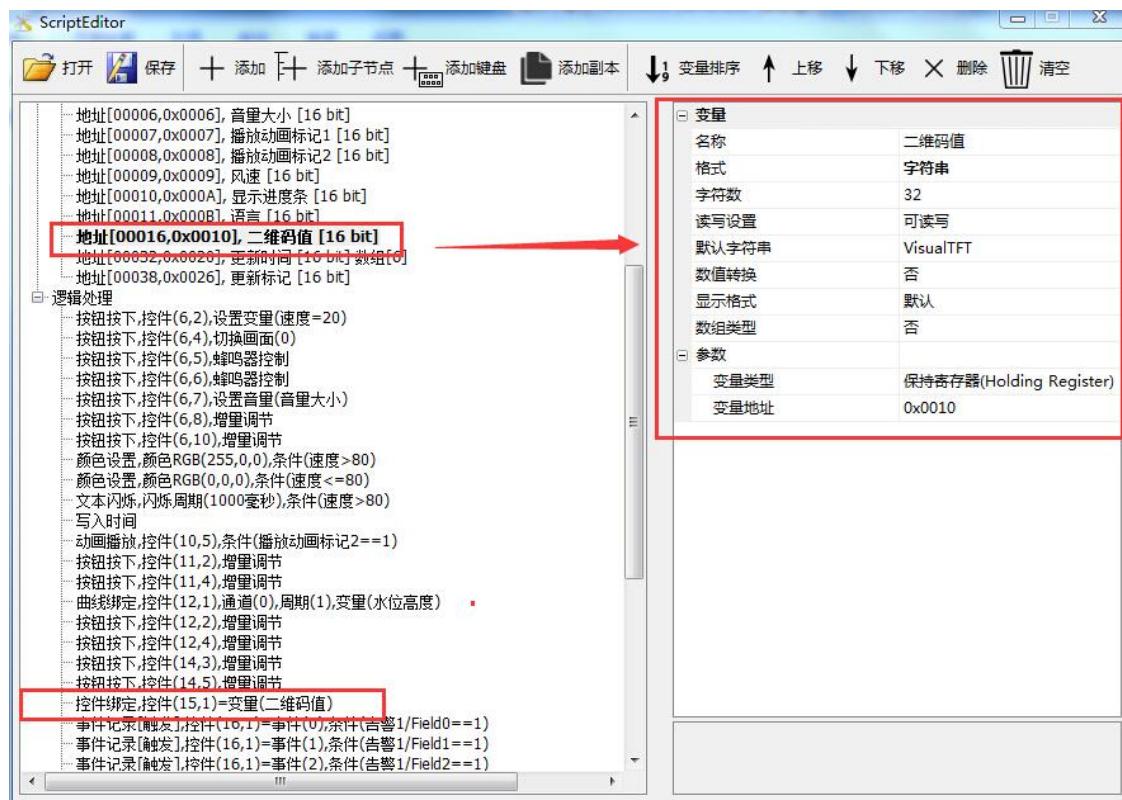


图 4-47 二位码变量及控件绑定

#### 4.2.17 数据记录控件

【数据记录控件-当前告警】和【数据记录控件-历史告警】画面主要对数据记录控件的使用做示例说明。

数据记录控件在 MODBUS 协议版本的串口屏中常用于三种“记录类型”：“当前告警”、“历史告警”和“历史告警与解除”，其中“历史告警”与“历史告警与解除”的基本相同，“历史告警与解除”相比“历史告警”，每条告警记录增加了告警的解除时间。本例只对“当前告警”和“历史告警”做示例说明。

数据记录控件的记录类型作为“当前告警”和“历史告警”的使用设置过程相同，步骤如下：

- 1、设置数据记录控件的“记录类型”为“当前告警”或“历史告警”，如图 4-50 所示；
- 2、在“告警解析”中设置可能的事件 ID 数值、颜色及描述信息，如图 4-51 所示；
- 3、针对每个事件 ID 添加用途为“事件记录[触发]”的逻辑处理，如图 4-52 所示；每个逻辑处理中根据需要设置“执行条件”；本例中创建了一个保持寄存器类型的变量“告警 1”，再对“告警 1”的每个位建立“Field0”、“Field1”、“Field2”子节点，判断“Field”，是否为 1，作为是否执行的条件。

按照如上步骤配置完成“当前告警”和“历史告警”，此时，编译工程并运行虚拟串口屏。当先后设置“告警 1”的值为 3（Field0 = 1, Field1 = 1, Field2 = 0）、4（Field0 = 0, Field1 = 0, Field2 = 1）后，当前告警显示如图 4-53 所示，历史告警如图 4-54 所示。

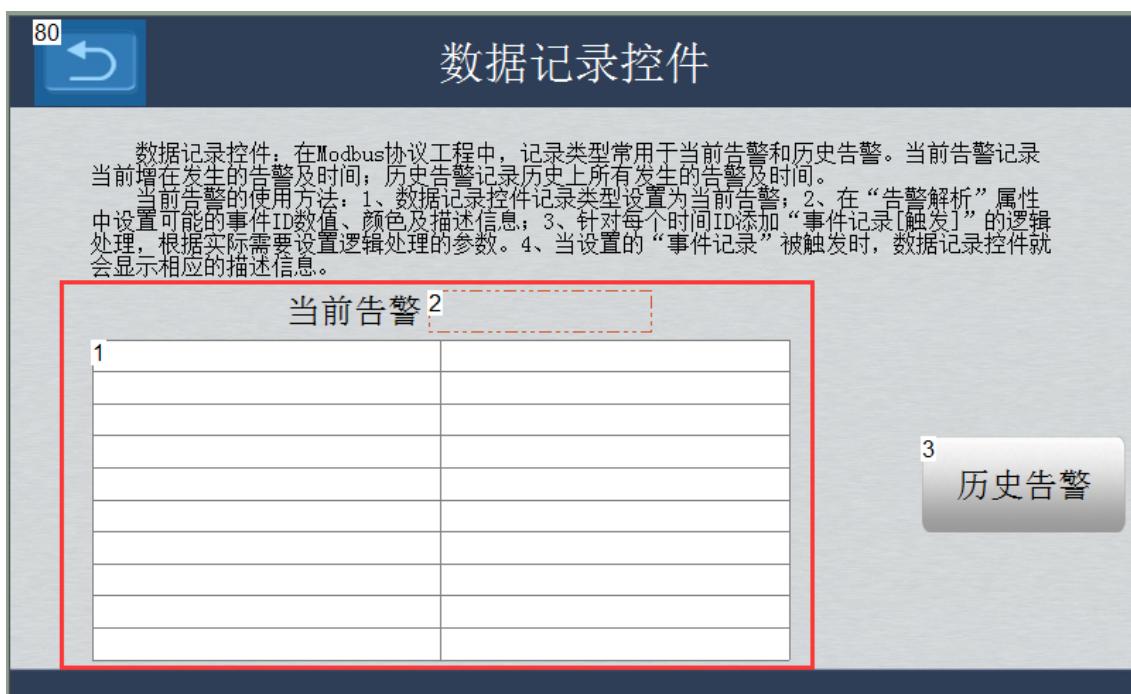


图 4-48 数据记录控件-当前告警

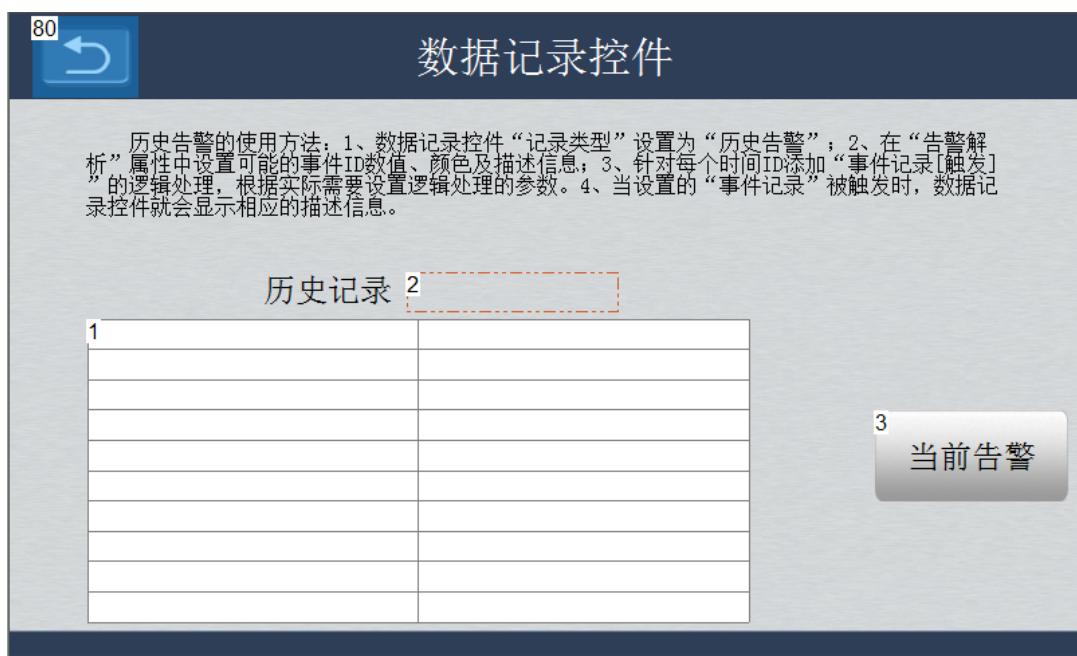


图 4-49 数据记录控件-历史告警、

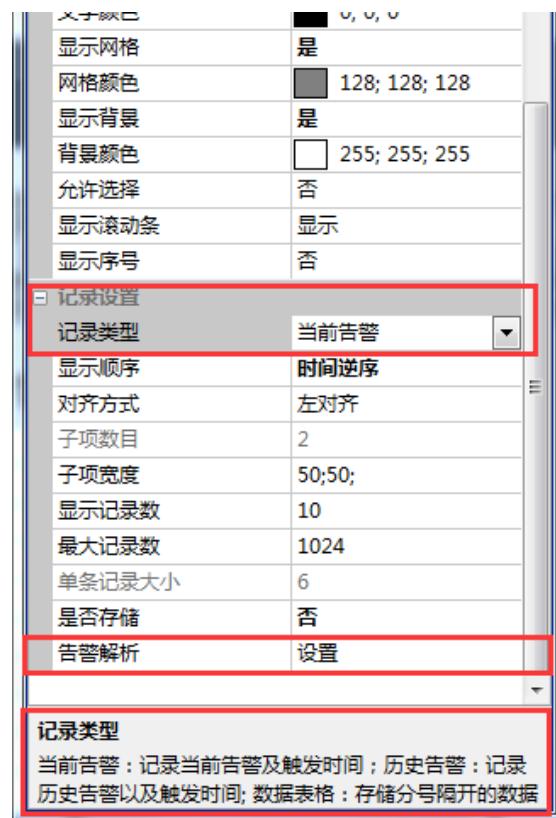


图 4-50 数据记录控件属性设置

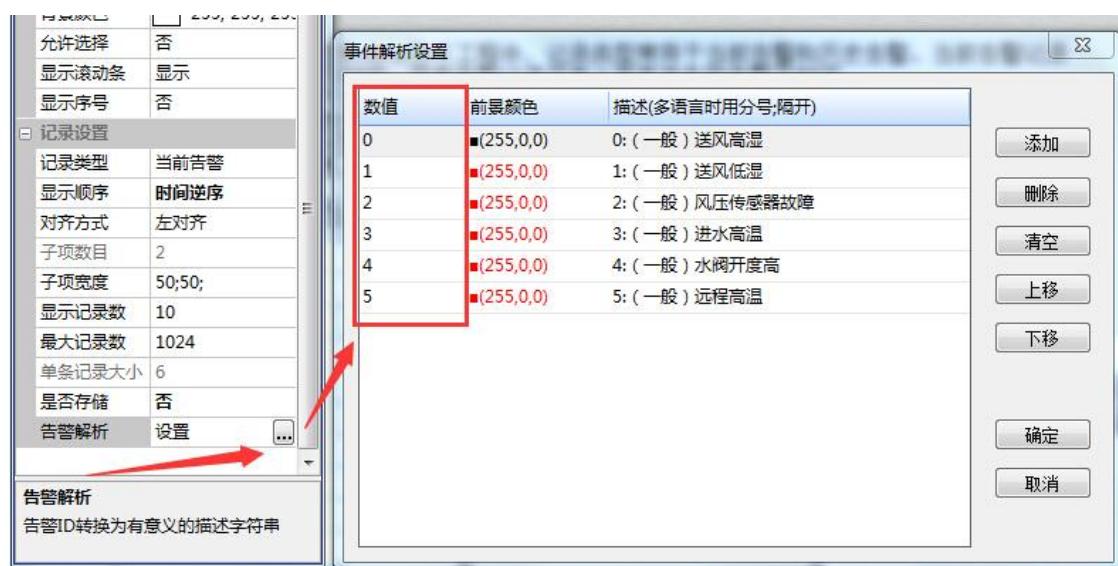


图 4-51 告警解析设置

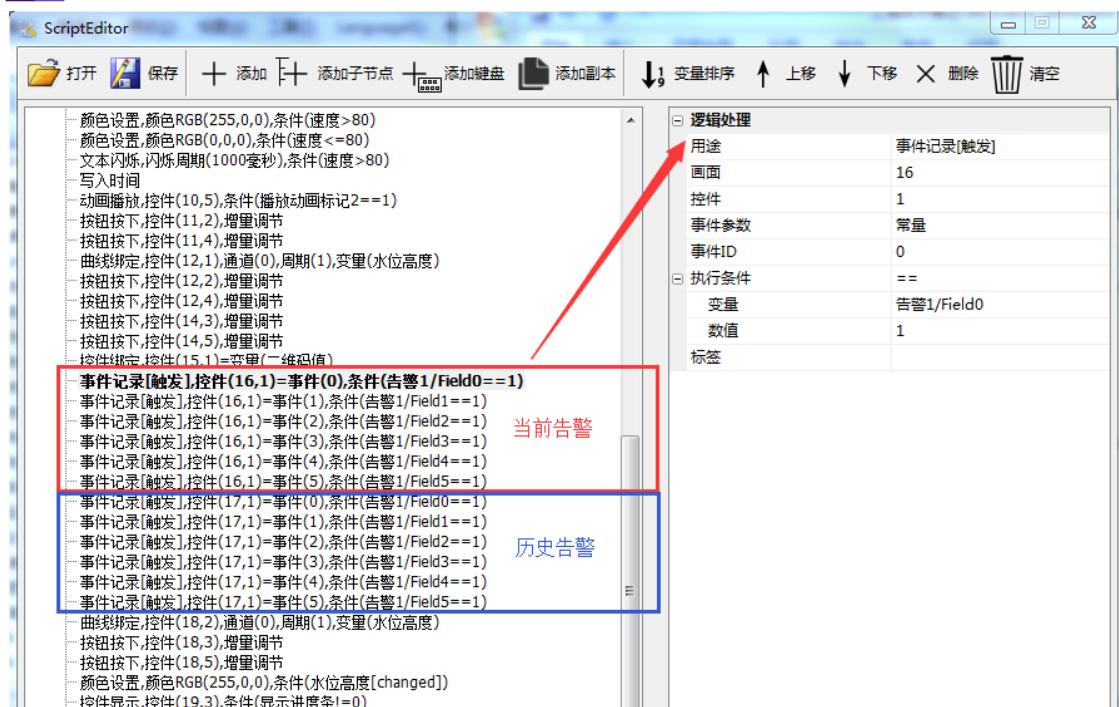


图 4-52 当前告警的逻辑处理

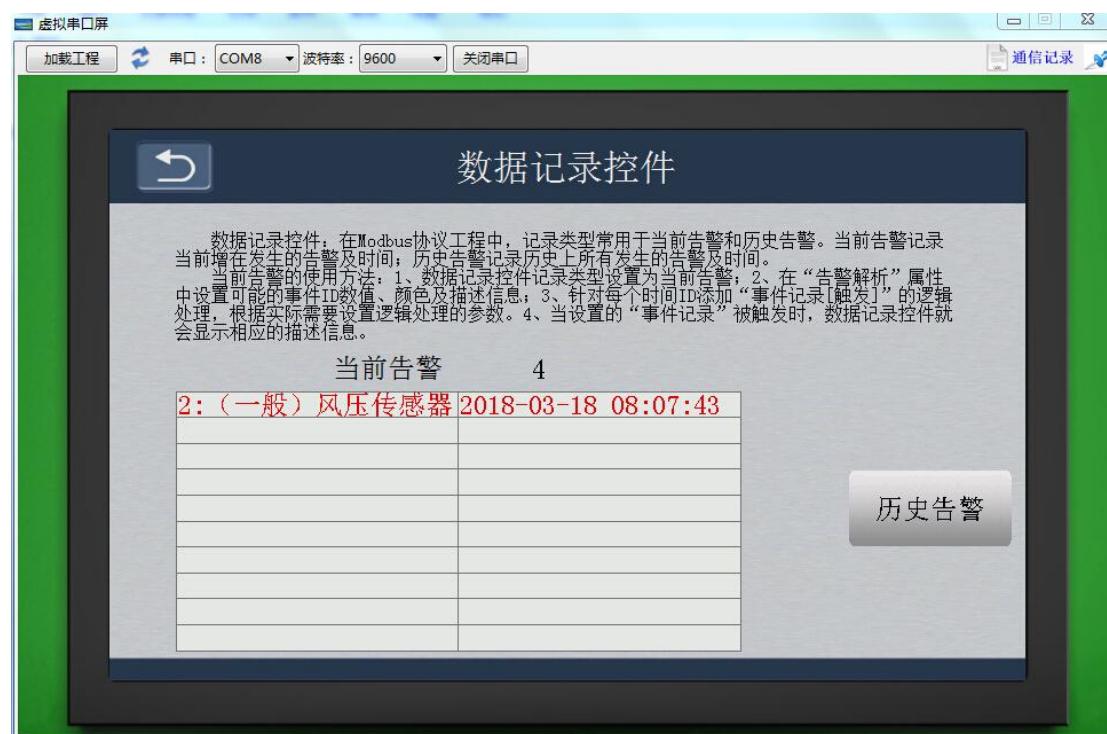


图 4-53 当前告警

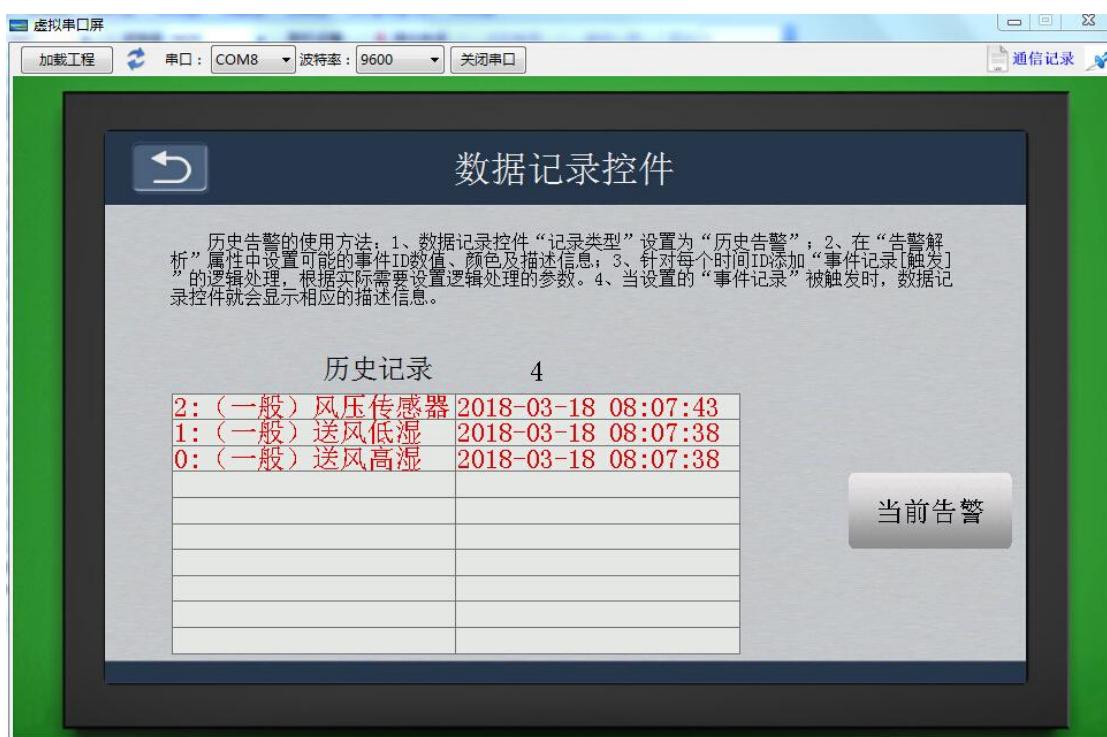


图 4-54 历史告警

#### 4.2.18 历史曲线控件

【历史曲线控件】画面主要对历史曲线控件的使用做示例说明。历史曲线控件和曲线控件都可以实时采样、显示绑定的变量值，不同处是历史曲线控件保存前期的数据，可以查看历史数据。MODBUS 协议版本中，历史曲线控件用于显示关联的变量值，历史曲线控件周期性的采样变量值，然后将所有采样值绘制到历史曲线控件内，连接形成曲线；历史曲线控件的每个通道都可以关联一个变量，最多可以设置 8 个通道。通过设置“曲线绑定”用途的逻辑处理绑定显示。

【历史曲线控件】画面中 ID 为 2 的历史曲线控件，预设一个通道，为该控件的通道 0，添加关联的逻辑处理，参数如图 4-56 所示，每“1”秒采样变量一次“水位高度”的值。画面 ID 为 3、5 的按钮用于调整“水位高度”值，ID 为 4 的文本控件实时显示“水位高度”值；经过动态的调整“水位高度”，可以形成如图 4-57 所示的曲线控件图，同时可以通过滚动条拖动查看历史的数据。

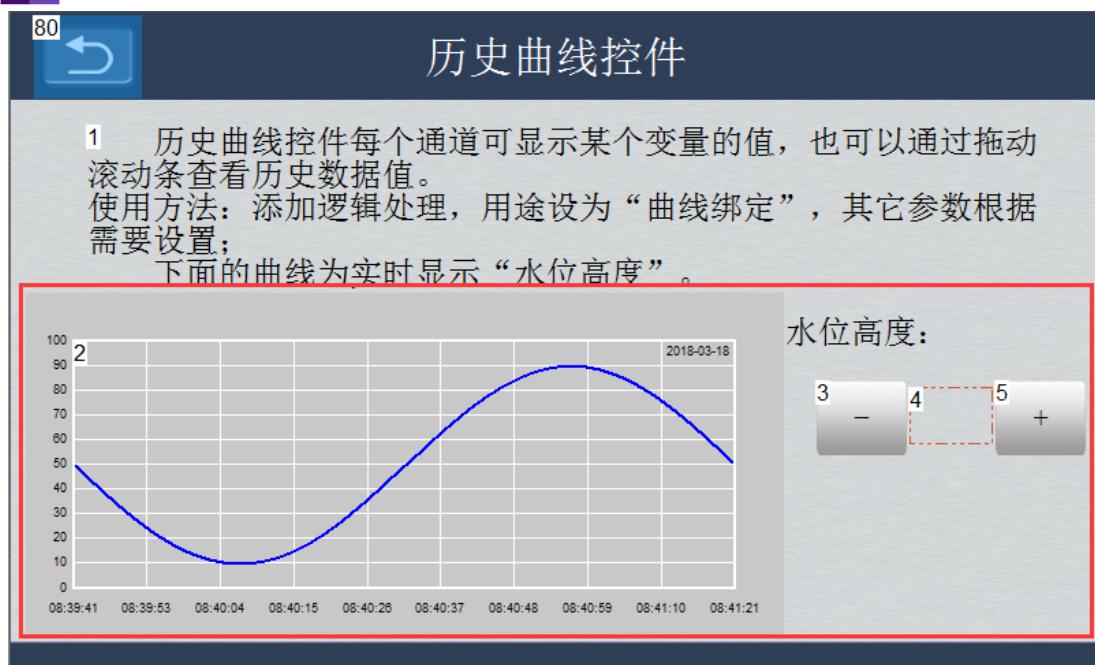


图 4-55 历史曲线控件

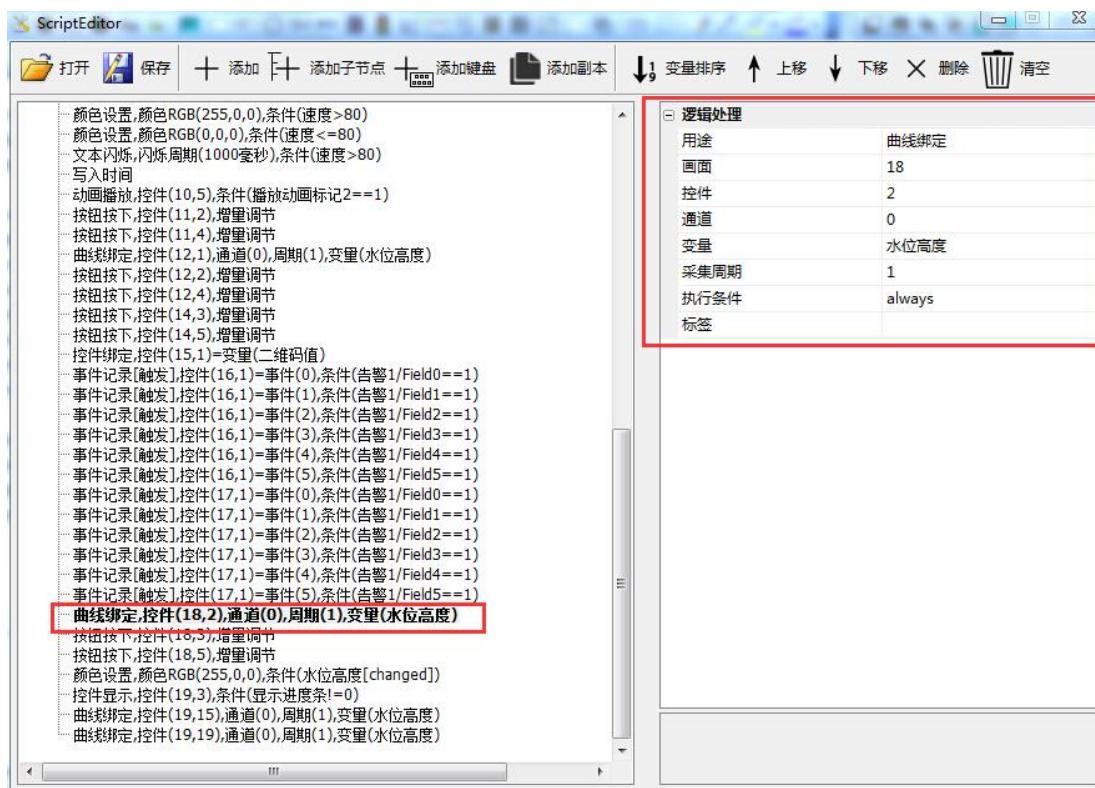


图 4-56 历史曲线控件的逻辑处理

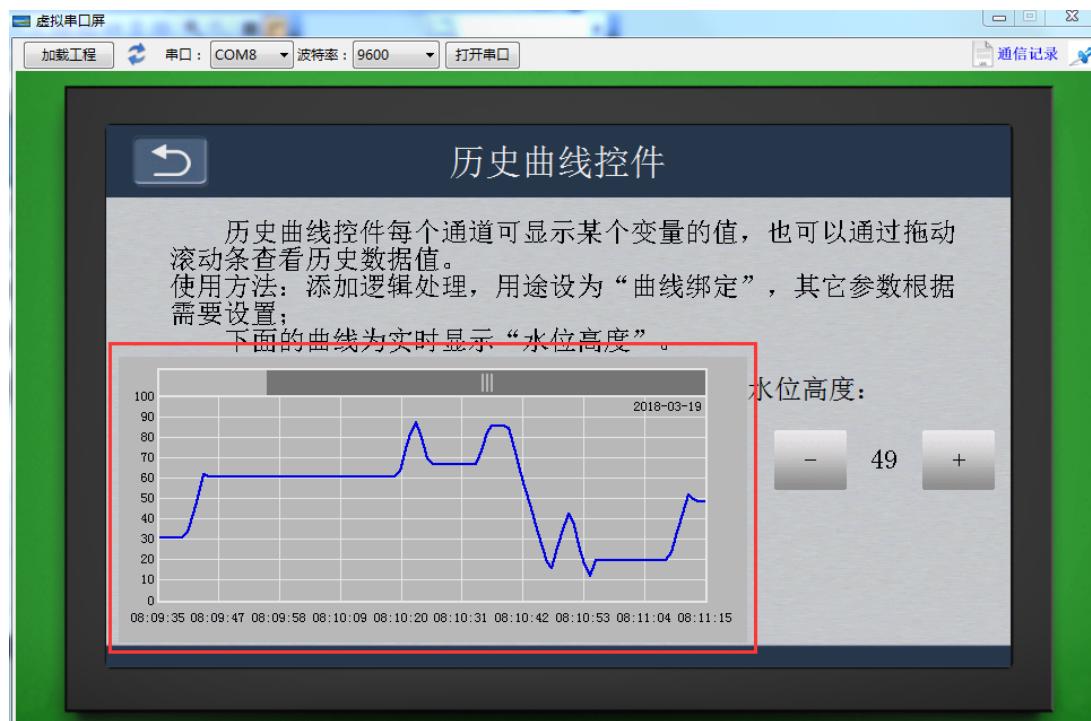


图 4-57 历史曲线控件运行示例

#### 4.2.19 隐藏显示控件

【隐藏显示控件】画面主要对隐藏显示控件功能的使用做示例说明。很多情况下工程需要在某个条件下隐藏或显示一些控件。隐藏和显示控件通过两种方式实现：1、逻辑处理的控件显示，2、MINIC 中的 show() hide() 函数。

##### 1、逻辑处理

【隐藏显示控件】画面增加如图 4-58 所示，ID 为 2 的文本控件、ID 为 3 的进度条控件，文本控件绑定变量“显示进度条”，增加逻辑处理，如图 4-59 所示。

此时，编译工程并运行虚拟串口屏，设置变量“显示进度条”的值为 0 时，进度条隐藏，设置变量为 1，进度条显示。

##### 2、MINIC 的 show() hide() 函数

【隐藏显示控件】画面增加如图 4-58 所示，ID 为 5 的“隐藏”按钮控件、ID 为 6 的“显示”按钮控件，增加 ID 从 7-19 的各种控件。在 MINIC 代码中添加如图 4-60 所示的按钮按下的动作处理。

此时，编译工程并运行虚拟串口屏，按下“隐藏”按钮控件时，ID 为 7-19 的控件隐藏，按下“显示”按钮控件时，ID 为 7-19 的控件显示。

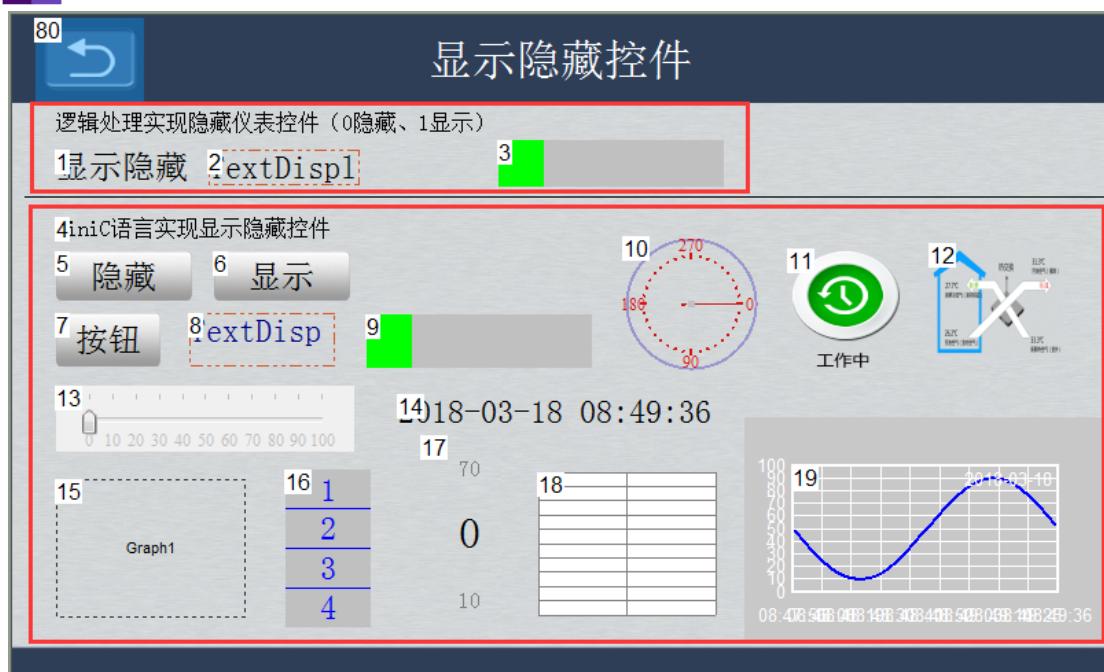


图 4-58 隐藏显示控件

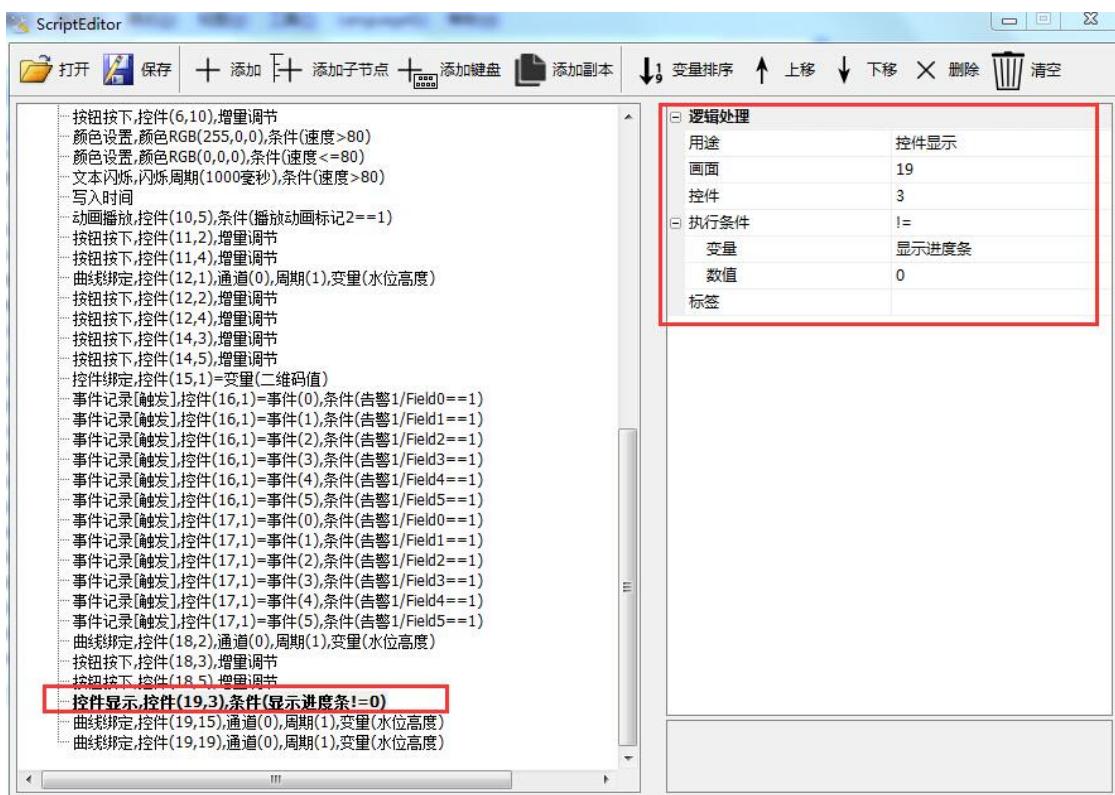


图 4-59 隐藏显示控件的逻辑处理

```
69 /*  
70 函数: on_control_notify  
71 功能: 控件值更新通知  
72 参数: screen_id, 画面ID  
73 参数: control_id, 控件ID  
74 参数: value, 新值  
75 */  
76 void on_control_notify(int screen_id,int control_id,int value)  
77 {  
78     int i;  
79     if(screen_id == 19)  
80     {  
81         //隐藏显示画面  
82         if(control_id == 5 && value == 1)  
83         {  
84             //隐藏  
85             for(i = 7; i <=19; ++i)  
86             {  
87                 hide(19, i);  
88             }  
89         }  
90         else if(control_id == 6 && value == 1)  
91         {  
92             //显示  
93             for(i = 7; i <=19; ++i)  
94             {  
95                 show(19, i);  
96             }  
97         }  
98     }  
99 }  
100
```

图 4-60 隐藏显示控件 MINIC 代码

#### 4.2.20 多语言

【多语言】画面主要对多语言功能的使用做示例说明。很多情况下客户工程需要多语言选择，文字、菜单控件、选择控件支持多语言设置。

【多语言】画面中增加文字、菜单控件、选择控件，如图 4-61 所示，多语言属性设置三种语言，增减每种语言的选项。如图 4-62、图 4-63、图 4-64 所示。另外增加一个选择控件选择语言类型，绑定变量“语言”，在 MINIC 种增加对语言选择的处理，设置语言的系统变量 sys.lang。

此时，编译工程并运行虚拟串口屏，通过选择控件选择语言种类，文字、菜单控件、选择控件的内容，可以显示不同的语言。

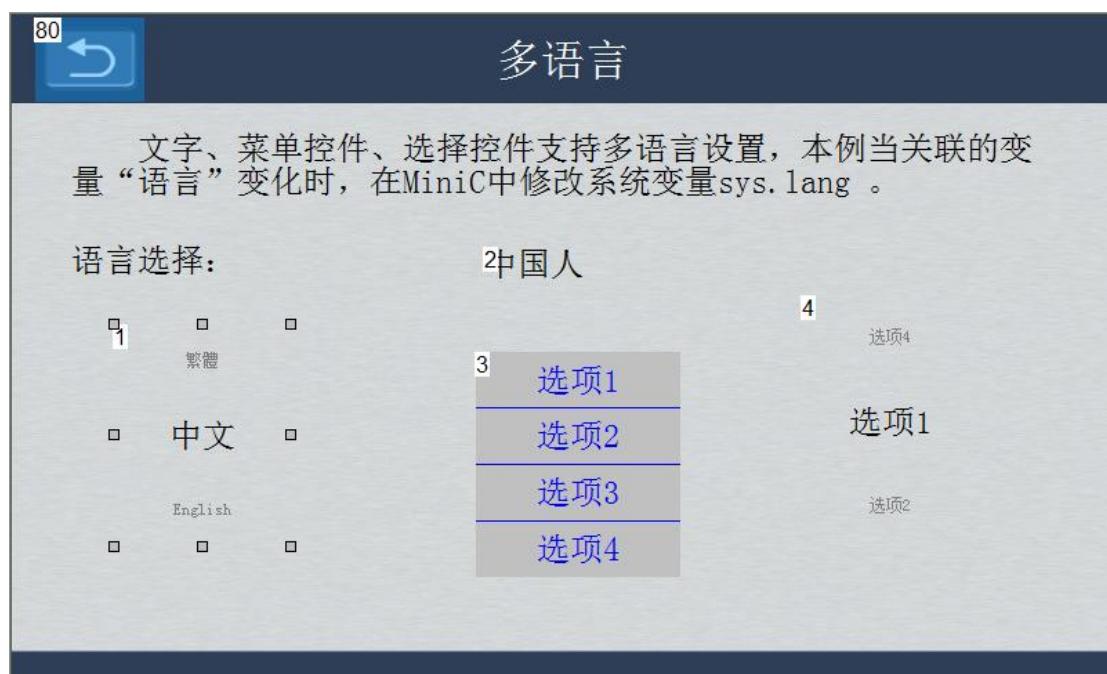


图 4-61 多语言

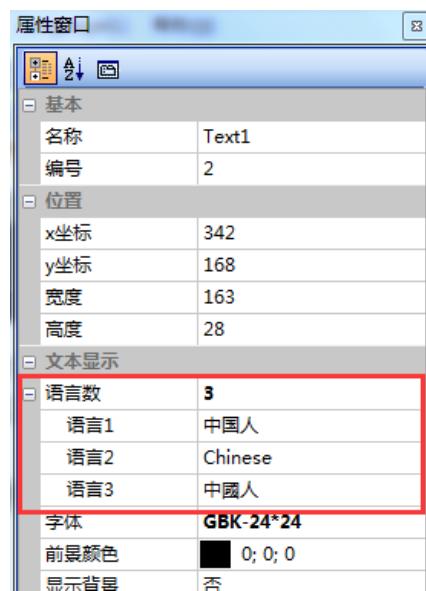


图 4-62 文字多语言设置



图 4-63 菜单多语言设置

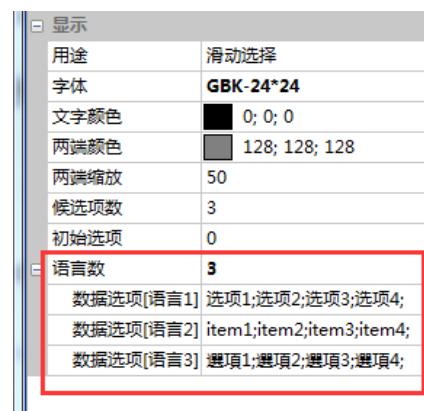


图 4-64 选择控件多语言设置

```

47 */
48 * 函数: on_variant_update
49 * 功能: 串口通信导致变量更新时, 执行此函数
50 */
51 void on_variant_update()
52 {
53     //操作符'@'用于判定某个寄存器是否发生改变
54     //更新RTC时间
55     if(@"更新标记")
56     {
57         set_date("更新时间"[0],"更新时间"[1],"更新时间"[2]);
58         set_time("更新时间"[3],"更新时间"[4],"更新时间"[5]);
59     }
60
61     //“语言”发生变化时, 修改设置系统语言
62     if(@"语言")
63     {
64         sys.lang = "语言";
65     }
66 }
67
68

```

图 4-65 多语言