



实验 3 MODBUS 主站实验

1.实验目的

串口屏作为 Modbus 主站，读取和设置温湿度模块的参数。

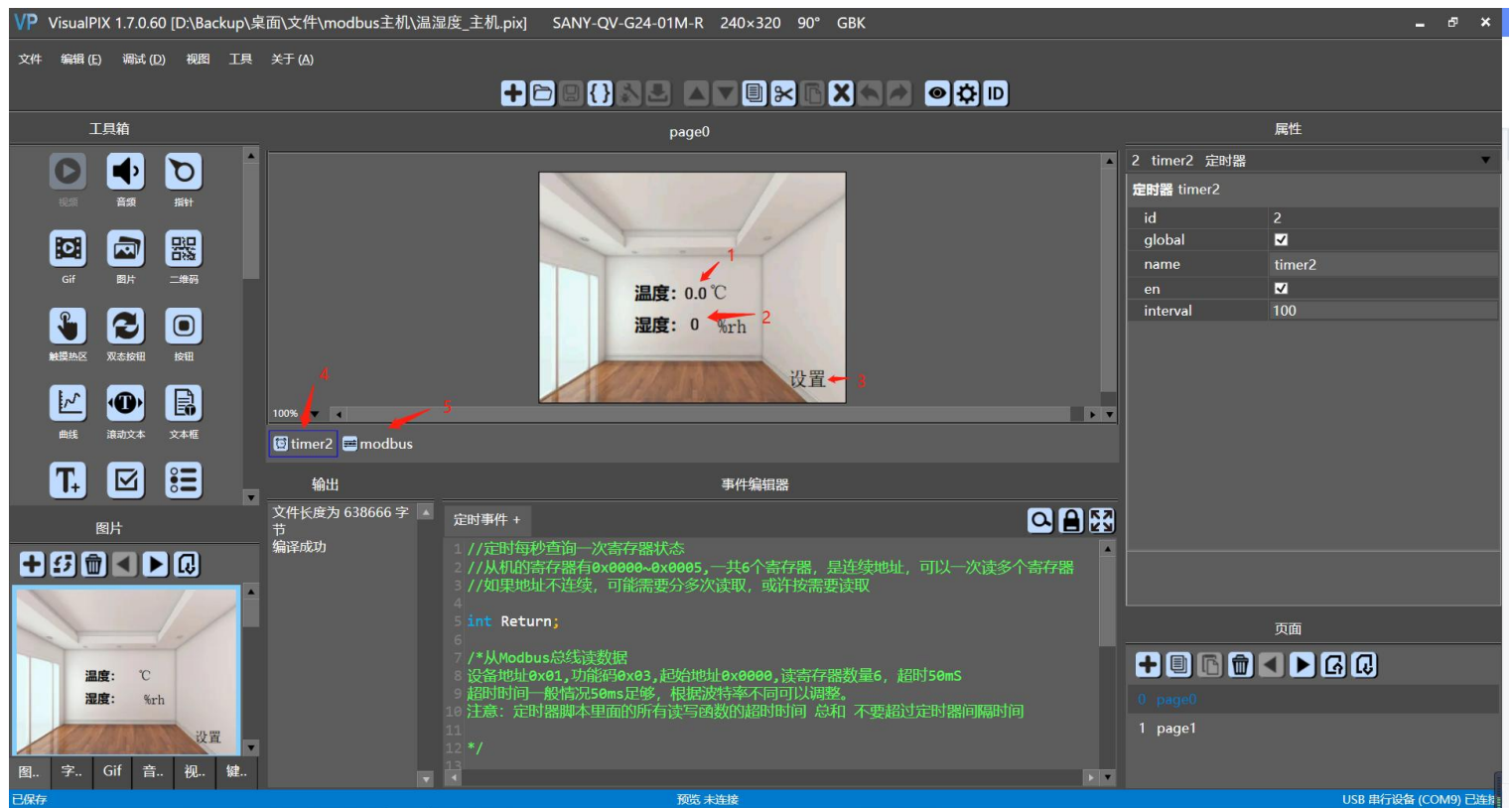
说明:型号名称中带有字母 M 的，表示支持 modbus 协议。

2.页面设计

项目设置里面，勾选 Modbus 主机，表示显示屏作为 modbus 主站。



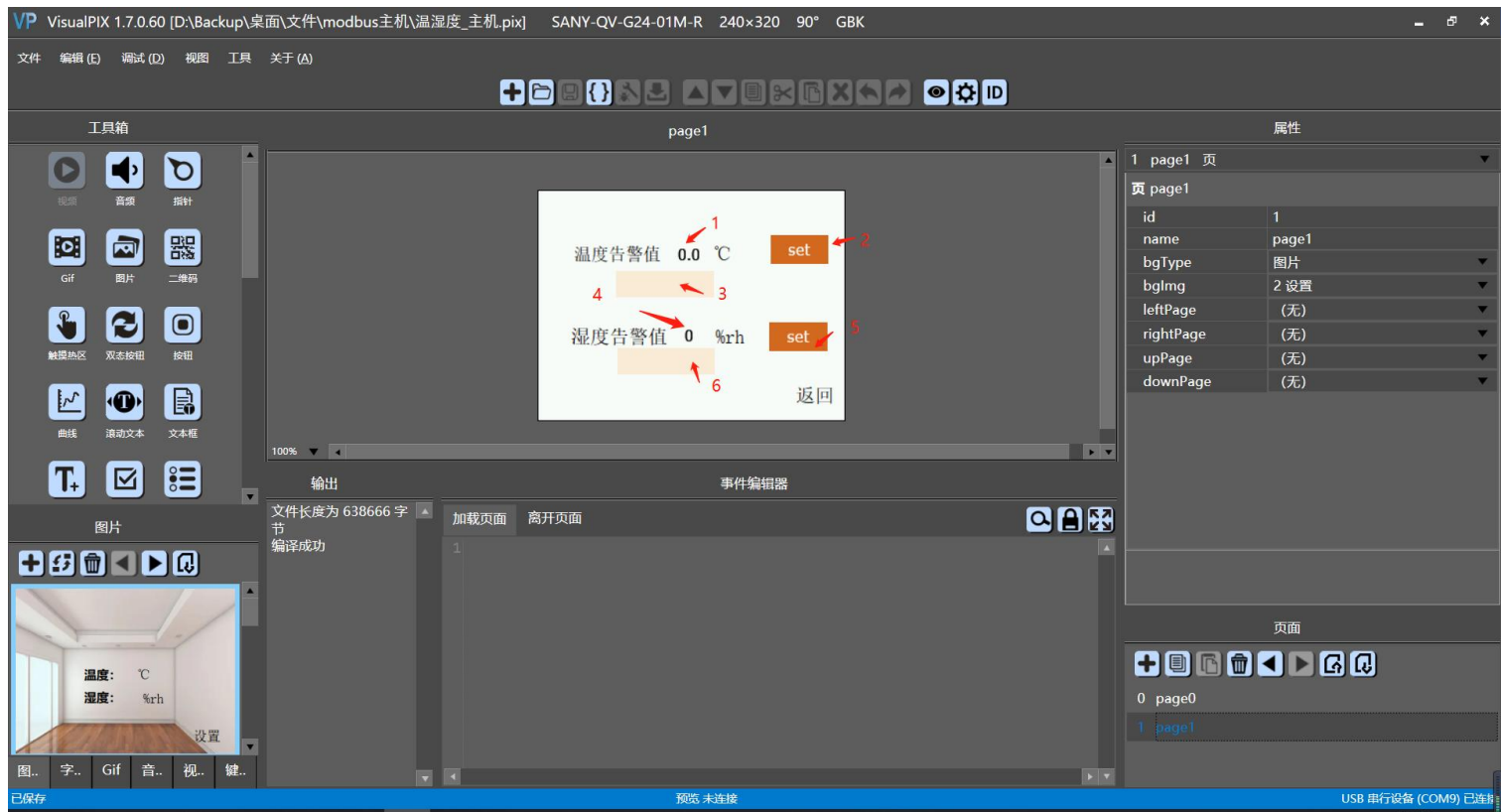
Page0 页面控件





- (1)浮点数控件，显示温度值
- (2)整数控件，显示湿度值
- (3)按钮控件，点击按钮切到第二页
- (4)定时控件，定时每秒发送读取帧，获取从机寄存器值，并显示。选中可以查看事件脚本代码。
- (5)modbus 控件

page1 页面控件



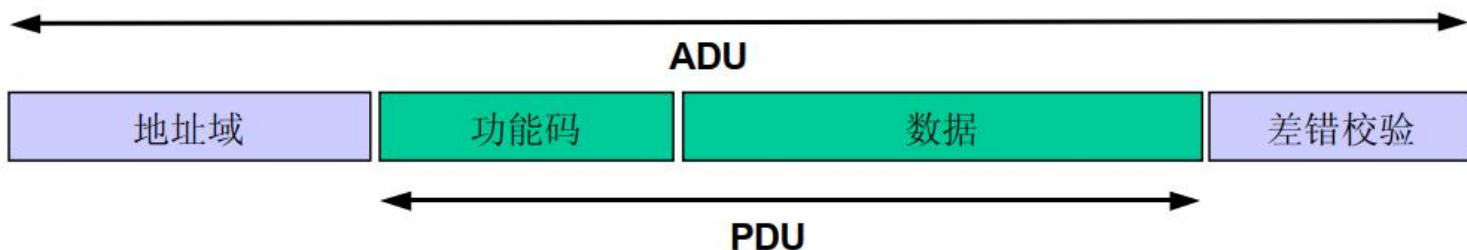
- (1)浮点数控件，显示温度告警值
- (2)按钮控件，调用 modbus 写函数，设置温度告警值
- (3)文本控件，显示温度告警设置是否成功
- (4)整数控件，显示湿度告警值
- (5)按钮控件，调用 modbus 写函数，设置湿度告警值
- (6)文本控件，显示湿度告警设置是否成功

3.串口屏协议处理

modbus 协议简单介绍

注：modbus 介绍取至于《MODBUS 协议中文版》部分，详细介绍请查阅 modbus 相关资料。

MODBUS 协议定义了一个与基础通信层无关的 P 简单协议数据单元（DU）。特定总线或网络上的 MODBUS 协议映射能够在应用数据单元（ADU）上引入一些附加域





modbus 有主站和从站的区分，一个完整的总线上有一个主站和若干从站，每个从站有自己独特的地址，区分是哪个从站设备。

地址域：即从站地址，1 字节，用一个字节编码 MODBUS 数据单元的功能码域。有效的码字范围是十进制 1-255（128-255 为异常响应保留）。

功能码：1 字节，当从客户机向服务器设备发送报文时，功能码通知服务器执行哪种操作。

数据：根据不同功能码，有着不同的数据格式和长度

差错校验：循环冗余校验 (CRC)

有两种串行传输模式被定义:RTU 模式和 ASCII 模式。

它定义了报文域的位内容在线路上串行的传送。它确定了信息如何打包为报文和解码。

Modbus 串行链路上所有设备的传输模式 (和串行口参数) 必须相同。

尽管在特定的领域 ASCII 模式是要求的，但达到 Modbus 设备之间的互操作性只有每个设备都有相同的模式：所有设备必须实现 RTU 模式。ASCII 传输模式是选项。

设备应该由用户设成期望的模式，RTU 或 ASCII。默认设置必须为 RTU 模式。

RTU 传输模式是常用的方式，报文帧格式如下：

子节点地址	功能代码	数据	CRC
1 字节	1 字节	0 到 252 字节	2 字节 CRC 低 CRC 高

Modbus RTU 帧最大为 256 字节。

本实验，显示屏作为主站，通过 modbus 协议标准获取温度模块的参数。

温度模块 modbus 参数如下：

设备地址	寄存器地址			
1	0x0000	温度 H	只读	float 类型，两个寄存器表示温度，单位 0.1℃
	0x0001	温度 L		
	0x0002	温度报警值 H	读写	float 类型，两个寄存器表示温度，单位 0.1℃
	0x0003	温度报警值 L		
	0x0004	湿度	只读	16 位 整 数 类 型 ， 单 位%rh
0x0005	湿度报警值	读写		
读功能码 0x03,写功能码 0x10				

modbus 为控件名称(也可以改为其他名字)，与 modbus 协议帧对应的控件属性如下：

从站地址	功能码	数据	校验
modbus.addr	Modbus.fcode	modbus.rxBuf	

定时器控件用于定时查询寄存器值，脚本代码如下：

```
//定时每秒查询一次寄存器状态
//从机的寄存器有 0x0000~0x0005,一共 6 个寄存器，是连续地址，可以一次读多个寄存器
//如果地址不连续，可能需要分多次读取，或许按需要读取
/*****本实验寄存器列表*****/
0x0000 温度 H 只读 float 类型，两个寄存器表示温度，单位 0.1℃
0x0001 温度 L
0x0002 温度报警值 H 读写 float 类型，两个寄存器表示温度，单位 0.1℃
```



0x0003 温度报警值 L

0x0004 湿度 只读 16 位整数类型，单位%rh

0x0005 湿度报警值 读写

*****/

int Return;

/******modbus 帧与控件属性的对应关系*****

从站地址 功能码 数据 校验

modbus.addr Modbus.fcode modbus.rxBuf

*****/

/*从 Modbus 总线读数据

设备地址 0x01,功能码 0x03,起始地址 0x0000,读寄存器数量 6，超时 50ms

超时时间一般情况 50ms 足够，根据波特率不同可以调整。

注意：定时器脚本里面的所有读写函数的超时时间 总和 不要超过定时器间隔时间

*/

//读寄存器 参数（从机地址 功能码 寄存器起始地址 寄存器数量 超时时间）

Return = modbusRead(0x01,0x03,0x0000,6,50);

if(Return == 0) //一定要检测返回值，返回值为 0，表示读取成功

{

 //获取温度值

 temp.valf = bytesToFloat(modbus.rxBuf,1,1); //将 4 个字节转换成一个浮点数。modbus.rxBuf[1~4]

 //获取温度告警值

 page1.temp_alarm.valf = bytesToFloat(modbus.rxBuf,5,1); //将 4 个字节转换成一个浮点数。modbus.rxBuf[5~8]

 //获取湿度值

 hum.val = (modbus.rxBuf[9]<<8) | modbus.rxBuf[10];

 //获取湿度告警值

 page1.hum_alarm.val = (modbus.rxBuf[11]<<8) | modbus.rxBuf[12];

}

Page1 的设置按钮，弹起事件主要是寄存器写操作，脚本代码如下

int res;

byte a[4];

floatToBytes(temp_alarm.valf,a,0,1);

//写操作函数：参数(从机地址 功能码 寄存器地址 寄存器数量 数组 数组长度 超时时间)

res = modbusWrites (0x01, 0x10, 0x0002, 2, a, 4, 50);

if(res == 0)

{



```
text5.txt = "Set OK";  
}  
else  
{  
    text5.txt = "Set Fail";  
}  
page0.timer2.en = 1;
```

解析的主要过程，是对 `modbus.rxBuf` 字节数组的处理。上面读操作，使用了移位+按位或操作得到一个无符号的整数并赋值为控件，用于显示；还使用字节转浮点数的函数 `bytesToFloat`，将 4 个字节转换成一个浮点数。具体函数相关的介绍，可以参考第五章函数的介绍。

4. 下载验证

编译成功后，点击下载按钮，选择正确的端口号和波特率，下载到串口屏。可以看到显示如下显示：



用 VP 软件将例程下载到串口屏，连接温湿度模块，即可看到温湿度模块的参数。



调试时，将显示屏的串口数据发送到电脑的串口助手上，查看读命令、写命令是否正确：



串口号: COM6

波特率: 115200

数据位: 8

校验位: None

停止位: One

打开串口

接收区设置.

☐ 接收并保存到文件

☒ 十六进制显示

☐ 暂停接收显示

☒ 自动断帧

? 5

☐ 接收脚本



Add Timesta

保存数据

清空数据

发送区设置.

☐ 发送文件

扩展命令

```
01 03 00 00 00 06 C5 C8
01 03 00 00 00 06 C5 C8
01 03 00 00 00 06 C5 C8
01 03 00 00 00 06 C5 C8
01 03 00 00 00 06 C5 C8
01 03 00 00 00 06 C5 C8
01 03 00 00 00 06 C5 C8
01 03 00 00 00 06 C5 C8
01 10 00 02 00 02 04 42 00 00 00 66 0E
01 03 00 00 00 06 C5 C8
01 03 00 00 00 06 C5 C8
01 03 00 00 00 06 C5 C8
01 03 00 00 00 06 C5 C8
01 10 00 05 00 01 02 00 41 66 35
01 03 00 00 00 06 C5 C8
01 03 00 00 00 06 C5 C8
01 03 00 00 00 06 C5 C8
01 03 00 00 00 06 C5 C8
```

← 设置温度告警值32.0

← 设置湿度告警值

← 寄存器查询命令