



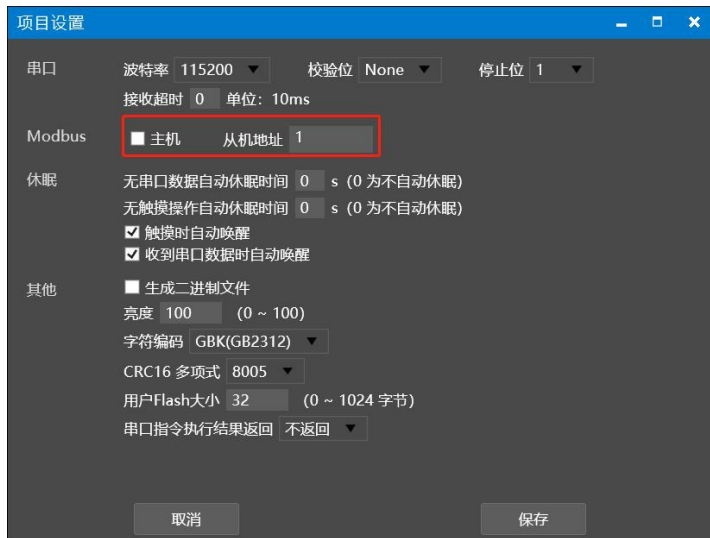
## 实验 4 MODBUS 从站实验

### 1.实验目的

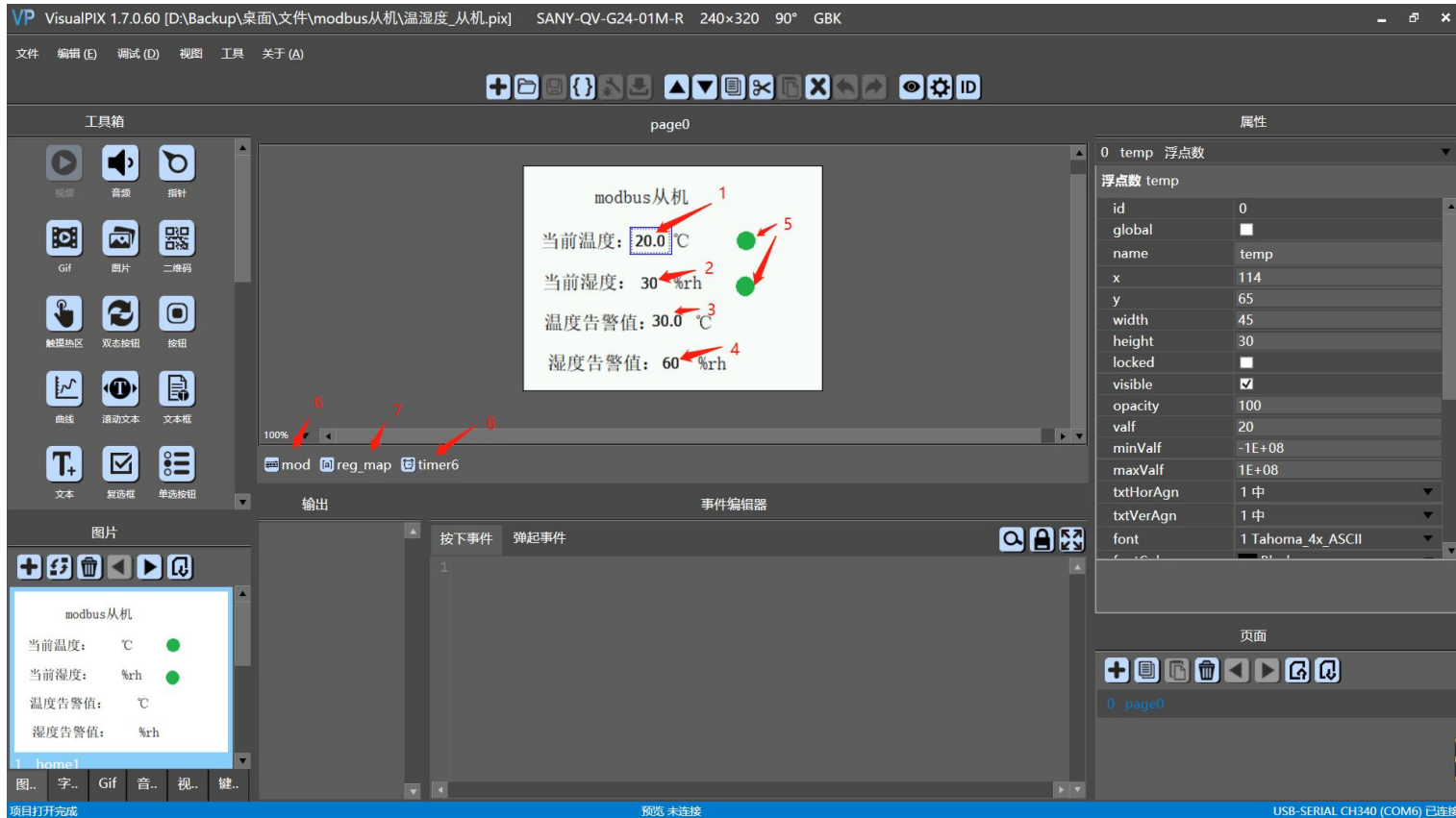
本实验模拟实验四 用到的温湿度模块，显示屏作为 modbus 从站。

### 2.页面设计

项目设置里面，modbus 不勾选主机，填写从机(从站)地址，默认为 1.



### page0 界面





- (1)浮点数控件，用于显示温度值，一位小数点
- (2)整数控件，显示湿度
- (3)浮点数控件，显示温度告警值
- (4)整数控件，显示湿度告警值
- (5)图片控件，用于显示告警状态，绿色正常，红色异常
- (6)Modbus 控件，协议处理主要在控件的事件编辑器里
- (7)数组控件，用于存放寄存器值的表
- (8)定时器，定时 1 秒模拟温湿度的变化，并判断是否告警

## 3.串口屏协议处理

寄存器列表

| 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<br>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 |    |

串口屏作为从站时，当收到一个主站的读/写命令后，显示屏自己判断地址和校验是否匹配，若无误则会触发一个 modbus 控件的事件，客户需要做的是完善这个 modbus 控件的事件。modbus 的相关协议处理，在 modbus 控件数据接收里面，脚本代码如下：

```
/*
modbus 从机接收处理 脚本
响应主机的读命令 和 写命令，并发送响应命令帧
从机地址 可在 项目设置里配置
*/

/*****modbus 协议帧对应的控件属性*****/
从站地址    功能码    数据    校验
modbus.addr  Modbus.fcode  modbus.rxBuf
*****/

int reg=0; //寄存器地址
int index;
int num; //寄存器个数
int Byte_num;
byte buf[16];

int j;
```



```
int i;

if(mod.fcode == 0x03) //功能码为 0x03,表示主站来读寄存器
{
    reg = (mod.rxBuf[0]<<8) | mod.rxBuf[1]; //取出要读的寄存器地址
    num = (mod.rxBuf[2]<<8) | mod.rxBuf[3]; //取出要读的寄存器个数
    if((reg <= 0x0005)&&(num <= 6)) //检测参数范围是否有效

    index = reg*2;
    {
        buf[0] = (byte)(num*2);          //字节数
        j = 1;
        for(i=0;i<num;i++)              //根据要读的寄存器个数，取出寄存器值，放到 buf 里面
        {
            buf[j++] = reg_map.valbs[index++];
            buf[j++] = reg_map.valbs[index++];
        }
        modbusSendBytes(mod.addr, mod.fcode, buf, 1+num*2, 0); //发送一帧 modbus 响应数据，从机发送无需无超时
    }
}

else if(mod.fcode == 0x10) //功能码为 0x10,表示主站写寄存器
{
    reg = (mod.rxBuf[0]<<8) | mod.rxBuf[1]; //取出要写的寄存器地址(起始地址)
    num = (mod.rxBuf[2]<<8) | mod.rxBuf[3]; //取出要写的寄存器数量
    Byte_num = mod.rxBuf[4]; //取出需要写的字节数

    if(num*2 < Byte_num) //处理寄存器数量和字节数不相等的情况
        num = Byte_num / 2;

    if((reg <= 0x0005)&&(num <= 6)) //检测参数范围是否有效
    {
        index = reg*2;
        for(i=0;i<num;i++)
        {
            reg_map.valbs[index] = mod.rxBuf[j+5];
            reg_map.valbs[index+1] = mod.rxBuf[j+6];
            index+=2;
            j+=2;
        }

        for(i=0;i<num;i++)
        {
            if(reg == 0) //温度 H
```



```
{
    //只读
}
else if(reg == 1) //温度 L
{
    //只读
}
else if(reg == 2) //温度告警值 H
{
    //放到 reg == 3 的时候，解析数据
}
else if(reg == 3) //温度告警值 L
{
    temp_alarm.valf = bytesToFloat(reg_map.valbs,(reg-1)*2,1); //将 4 字节转换成 float 类型，并赋值给控件
}
else if(reg == 4) //湿度
{
    //只读
}
else if(reg == 5) //湿度告警值
{
    hum_alarm.val = (reg_map.valbs[reg*2]<<8) | reg_map.valbs[reg*2+1];
}
reg++;
}
modbusSendBytes(mod.addr, mod.fcode, mod.rxBuf, 4, 0); //发送一帧 modbus 响应数据，从机发送无需无超时
}
```

串口屏每次收到主机的读操作、写操作都会执行 modbus 数据接收的脚本，因此在脚本里面，需要检测功能码，根据功能码(读操作或写操作)做处理，并发送响应帧。作为从机时，建议用一个数组控件 reg\_map 来缓存寄存器的值，便于操作。

定时器控件用于模拟温度和湿度值的变化，并检测是否需要报警，脚本代码如下：

```
/**模拟数据变化***/
temp.valf += 0.2;
if(temp.valf > 40)temp.valf = 20;
hum.val += 1;
if(hum.val > 70)hum.val = 30;
/***/
```

```
//将数据放到寄存器列表中
floatToBytes(temp.valf,reg_map.valbs,0,1);
reg_map.valbs[8] = (byte)(hum.val>>8);
reg_map.valbs[9] = (byte)(hum.val>>0);
```

```
//检测是否要告警
if(temp.valf > temp_alarm.valf)
```



```
{
    sta1.img = user.image.home2;
}
else
{
    sta1.img = user.image.home1;
}
if(hum.val > hum_alarm.val)
{
    sta2.img = user.image.home2;
}
else
{
    sta2.img = user.image.home1;
}
```

## 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：



用串口助手按照 modbus 协议帧，给串口屏发送查询和设置命令。测试命令如下：

获取温度：

助手发送 01 03 00 00 00 02 C4 0B

串口屏回复 01 03 04 41 E4 41 E4 9E 23 //28.5 度

设置温度告警值

助手发送 01 10 00 02 00 02 04 41 F1 99 9A DD 82 //30.2 度

串口屏回复 01 10 00 02 00 02 E0 08

获取湿度

助手发送 01 03 00 04 00 01 C5 CB

串口屏回复 01 03 02 00 2B F8 5B



设置湿度告警值

助手发送      01 10 00 05 00 01 02 00 3C A6 14

串口屏回复    01 10 00 05 00 01 11 C8

注意：这里温度是小数表示，4 个字节表示一个小数，大端模式，如 41 E4 41 E4，表示 28.5。

将本实验的从机 接到实验三 的主机串口上，主机可以看到从机的温湿度参数，并且可以设置温湿度告警值。