



实验 8 485 组网实验_主机

1.实验目的

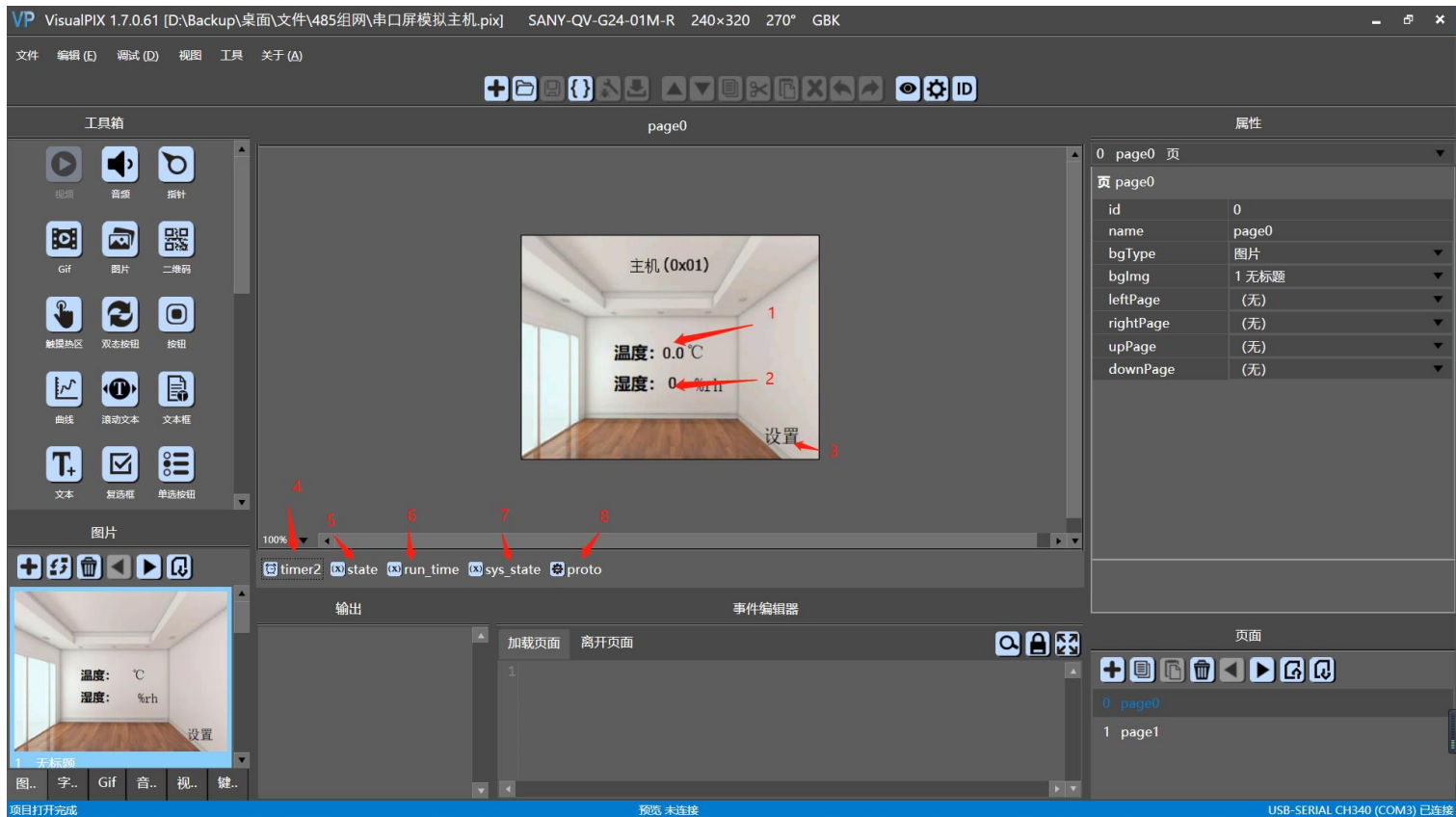
本实验为非 modbus 方式，实现 485 组网。modbus 组网参考实验三和实验四。

带 485 接口的串口屏，也可以进行 485 组网。要进行 485 组网，需要定义一套比较完整的协议。主机是谁，从机是哪些，主机和从机的地址需要定义好，不要冲突。

比较常见的，是 485 总线上一主机多从机的模式，主机可以主动发起查询和设置，从机只能被动响应主机的命令。本实验的协议仅仅简单的体现 485 怎么实现组网，仅供参考。

本实验模拟实验八中的主机。

2.页面设计



(1)浮点控件，显示温度值

(2)整控件，显示湿度

(3)按钮，翻页

(4)定时器，定时器间隔 100ms，定时时间到会执行定时器脚本。在定时器里面发送查询和设置命令

(5)变量控件，存放命令发送状态

(6)变量控件，存放系统运行时间

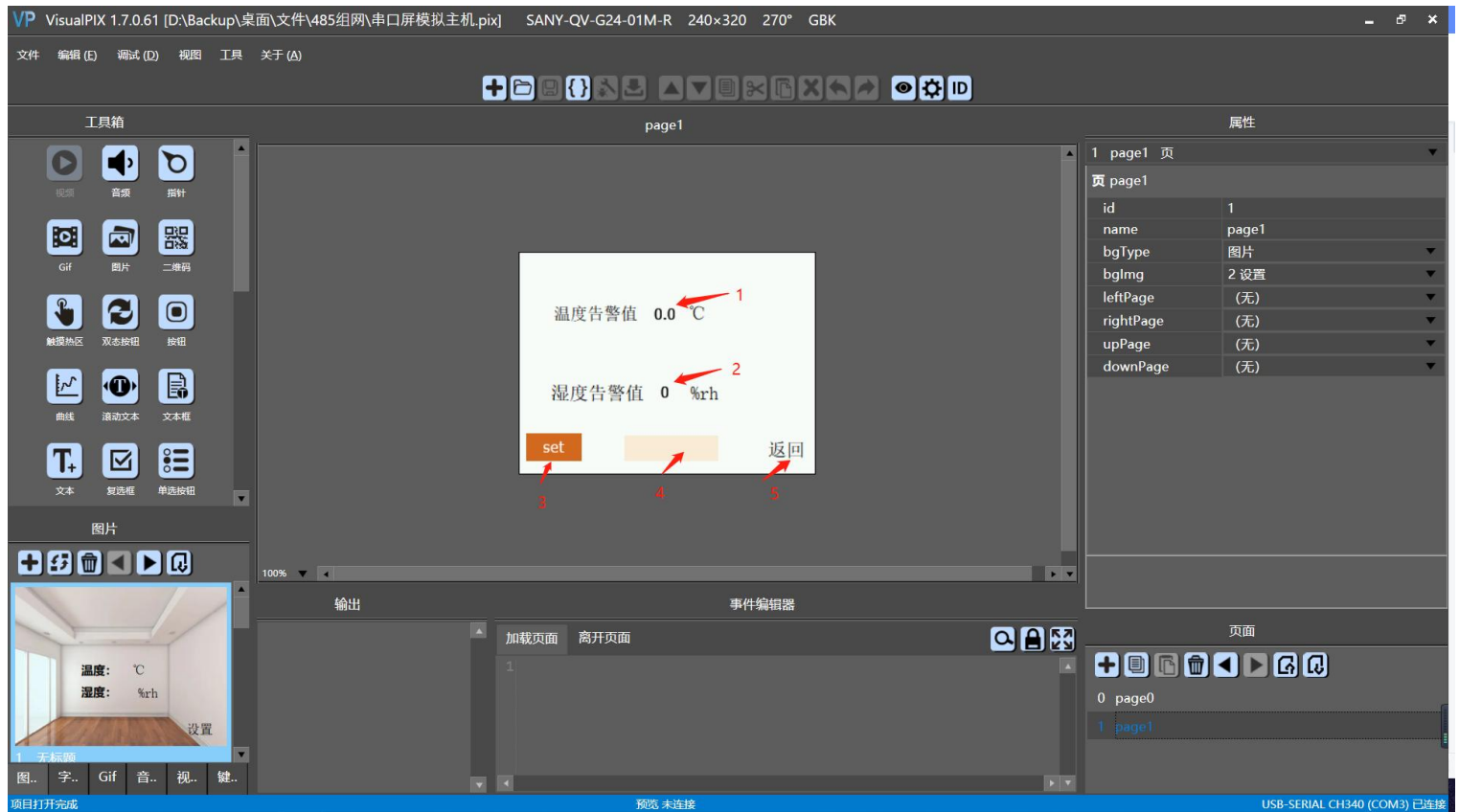
(7)变量控件，存放系统故障状态



(8)协议解析器控件，处理接收到的串口命令

选中控件，可以在右侧属性栏查看控件的各个属性和事件代码。详细属性说明可参考 第四章 控件的介绍和用法。

Page1



- 1、浮点数控件，显示温度告警值
- 2、整数控件，显示湿度告警值
- 3、按钮，点击按钮时会发送设置命令
- 4、文本控件，显示设置成功或失败
- 5、按钮，返回 page0

选中控件，可以在右侧属性栏查看控件的各个属性和事件代码。详细属性说明可参考 第四章 控件的介绍和用法。

3.串口屏协议处理

本实验通信协议定义如下：

报文帧格式

帧头	原地址	目的地址	命令码	参数	校验
0xA5	1Byte	1Byte	1Byte	N	2Byte

主机：STM32F103 实验板 主机地址 0x01

从机地址	命令码	参数长度	参数	校验
0xF1 温湿度模块	0x01 主机发送 查询温度值	0 字节		
	0xF1 从机响应	6 字节	[0~3]温度值 [4~5]湿度值	



	0x02 主机发送 查询告警温度值	0 字节		
	0xF2 从机响应	6 字节	[0~3]温度告警值 [4~5]湿度告警值	
	0x03 主机发送 设置告警温度值	6 字节	[0~3]温度告警值 [4~5]湿度告警值	
	0xF3 从机响应	1 字节	[0]是否成功	0 成功, 1 失败
0xF3 从机显示屏	0x01 主机发送 系统运行时间	2 字节	[0~1]运行时间	单位, 分钟
	0xF1 从机响应	1 字节		
	0x02 系统运行状态	1 字节	[0]运行状态	0 系统正常 1 系统故障
	0xF2 从机响应	0 字节		

本实验是串口屏模拟 STM32F103 实验板, 地址为 0x01.

定时器用于间隔发送查询和设置命令, 脚本代码如下

```
byte buf[10];
int crc_val;

if(state.val == 0) //查询温度值
{
    buf[0] = 0xA5;
    buf[1] = 0x01;
    buf[2] = 0xF1;
    buf[3] = 0x01;
    crc_val = crc16(buf,0,4);
    buf[4] = (byte)crc_val;
    buf[5] = (byte)(crc_val>>8);
    uartSendBytes(buf, 0, 6);//发送数组的数据
}
else if(state.val == 1) //查询温度告警
{
    buf[0] = 0xA5;
    buf[1] = 0x01;
    buf[2] = 0xF1;
    buf[3] = 0x02;
    crc_val = crc16(buf,0,4);
    buf[4] = (byte)crc_val;
    buf[5] = (byte)(crc_val>>8);
    uartSendBytes(buf, 0, 6);//发送数组的数据
}

else if(state.val == 2) //发送系统运行时间
```



```
{
    run_time.val++;

    buf[0] = 0xA5;
    buf[1] = 0x01;
    buf[2] = 0xF3;
    buf[3] = 0x01;
    buf[4] = (byte)run_time.val;
    buf[5] = (byte)(run_time.val>>8);
    crc_val = crc16(buf,0,6);
    buf[6] = (byte)crc_val;
    buf[7] = (byte)(crc_val>>8);
    uartSendBytes(buf, 0, 8);//发送数组的数据
}

else if(state.val == 3) //发送系统状态
{
    buf[0] = 0xA5;
    buf[1] = 0x01;
    buf[2] = 0xF3;
    buf[3] = 0x02;
    buf[4] = (byte)sys_state.val;
    crc_val = crc16(buf,0,5);
    buf[5] = (byte)crc_val;
    buf[6] = (byte)(crc_val>>8);
    uartSendBytes(buf, 0, 7);//发送数组的数据
}

state.val++;
if(state.val > 3)
    state.val = 0;

delay(20); //延时 20ms,等待从机响应
```

协议解析器处理串口接收的命令，脚本代码如下

```
int src_addr;
int dst_addr;
int len;
int crc_data;
int crc_val;

if(proto.rxBuf[0] == 0xA5)
{
    src_addr = proto.rxBuf[1];
    dst_addr = proto.rxBuf[2];

    if(dst_addr == 0x01) //检测该命令，是否是本机的命令
    {
        len = proto.rxBuf[3];
    }
}
```



```
crc_data = (proto.rxBuf[len-1]<<8) | proto.rxBuf[len-2];
crc_val = crc16(proto.rxBuf,0,len-2);
if(crc_data == crc_val) //CRC 判断
{
    if(src_addr == 0xF1) //温度模块的数据
    {
        if(proto.rxBuf[3] == 0xF1)
        {
            //获取温度值
            temp.valf = bytesToFloat(proto.rxBuf,4,0);

            //获取湿度值
            hum.val = (proto.rxBuf[9]<<8) | proto.rxBuf[8];
        }
        else if(proto.rxBuf[3] == 0xF2)
        {
            //获取温度告警值
            page1.temp_alarm.valf = bytesToFloat(proto.rxBuf,4,0);

            //获取湿度告警值
            page1.hum_alarm.val = (proto.rxBuf[9]<<8) | proto.rxBuf[8];
        }
        else if(proto.rxBuf[3] == 0xF3)
        {
            if(proto.rxBuf[4] == 0)
                page1.text5.txt = "设置成功";
            else
                page1.text5.txt = "设置失败";
        }
    }
    else if(src_addr == 0xF3) //从机显示屏的数据
    {
        //不用处理
    }
}
}
```

Page1 的设置按钮用于发送设置命令，脚本代码如下

```
byte buf[16];
int crc_val;

buf[0] = 0xA5;
buf[1] = 0x01;
buf[2] = 0xF1;
buf[3] = 0x03;
```



```
floatToBytes(temp_alarm.valf,buf,4,0); //温度告警值
buf[8] = (byte)(hum_alarm.val>>0); //湿度告警值 L
buf[9] = (byte)(hum_alarm.val>>8); //湿度告警值 H
crc_val = crc16(buf,0,10);
buf[10] = (byte)crc_val;
buf[11] = (byte)(crc_val>>8);
uartSendBytes(buf, 0, 12); //发送数组的数据

page0.timer2.en = 1;
```

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。
调试时，用电脑的串口助手监控串口屏的数据。



可以看到串口屏不停的发出命令

- (1)查询温度湿度值
- (2)查询温度湿度告警值
- (3)发送系统运行时间
- (4)发送系统状态

发送测试命令

//模拟从机 0xF1 的响应

```
A5 F1 01 F1 33 33 BB 41 3A 00 29 47 //温度 23.4，湿度 58
A5 F1 01 F2 00 00 20 42 46 00 22 10 //温度告警 40，湿度告警 70
A5 F1 01 F3 00 AB 4C //设置成功
```