

实验 7 485 组网实验_从机

1.实验目的

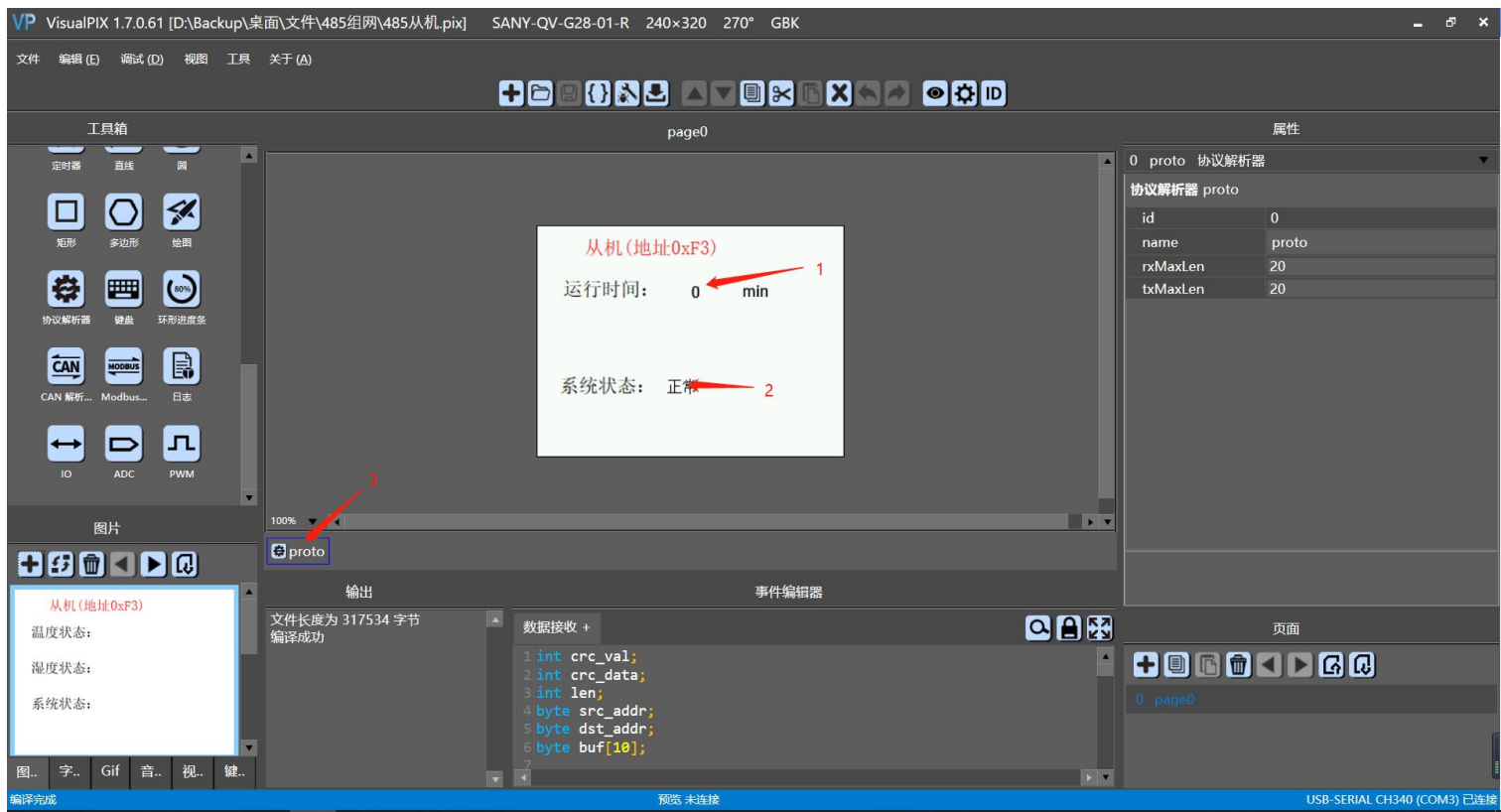
本实验为非 modbus 方式，实现 485 组网。modbus 组网参考实验三和实验四。

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。

本实验和实验二类似，区别是解析的协议不同。本实验的协议仅仅简单的体现 485 怎么实现组网，仅供参考。

2.页面设计



- (1)整数码件，显示系统运行时间
- (2)文本控件，中文显示系统运行状态
- (3)协议解析器控件，解析串口协议

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 查询温湿度值	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 主机发送 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		

本实验是从机串口屏，地址为 0xF3，显示主机的系统参数。

串口屏处理串口数据，在协议解析器控件 `proto` 的接收事件里面。包括帧头，地址，命令码和校验的判断，参数获取然后赋值给控件，显示出来。

协议解析器接收事件的脚本代码

```
int crc_val;
int crc_data;
int len;
byte src_addr;
byte dst_addr;
byte buf[10];

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

    if(dst_addr == 0xF3) //检测该命令，是否是本机的命令
    {
        len = proto.rxLen;

        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(proto.rxBuf[3] == 0x01) //主机设置温湿度故障信息
{
    time.val = (proto.rxBuf[5]<<8) | proto.rxBuf[4];

    //命令响应
    buf[0] = 0xA5;
    buf[1] = dst_addr;
    buf[2] = src_addr;
    buf[3] = 0xF1;
    crc_val= crc16(buf,0,4); //计算 CRC 值
    buf[4] = (byte)crc_val;
    buf[5] = (byte)(crc_val>>8);
    uartSendBytes(buf, 0, 6); //发送数组的数据
}
else if(proto.rxBuf[3] == 0x02) //主机设置系统运行状态
{
    if(proto.rxBuf[4] == 0)
        sta.txt = "正常";
    else
        sta.txt = "故障";

    //命令响应
    buf[0] = 0xA5;
    buf[1] = dst_addr;
    buf[2] = src_addr;
    buf[3] = 0xF1;
    crc_val= crc16(buf,0,4); //计算 CRC 值
    buf[4] = (byte)crc_val;
    buf[5] = (byte)(crc_val>>8);
    uartSendBytes(buf, 0, 6); //发送数组的数据

}
}
}
```

每收到一次正确命令。解析完成后，需要组帧，然后调用 `uartSendBytes`，响应主机的命令。

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，用电脑的串口助手给串口屏发命令



测试命令

» A5 01 F3 01 64 02 E2 3C //运行时间 612
« A5 F3 01 F1 E9 4C //响应

» A5 01 F3 02 01 61 A6 //系统状态 故障
« A5 F3 01 F1 E9 4C //响应

显示屏如下图显示，则正确。

