CPU A07 可编程逻辑控制器

用户手册

版本：V2.01
发布日期：08/2023
大连德嘉工控设备有限公司

目录

1 产品概述

A07 型 PLC 与西门子 S7-200 完全兼容，本体不带 IO 点，可以用西门子 STEP7-MicroWin 编程，内嵌 Modbus RTU 模式（非编程），连接 Modbus 变频器、仪器仪表等，通过软件内填表方式，实现 Modbus 的通讯。

A07 型 PLC 该产品具有以下特点：

- 体积小节省空间，价格低性能稳定。
- A07PLC 支持一路 485 通讯的功能，主要用于模拟量的采集与控制，内置两种 485 通讯方式，主要推荐填表方式，简单快速方便。
- 可以外接 7 个 SMART 扩展模块,扩展模块可以使用西门子原装的 IO 模块，也可以使用大连德嘉的 IO 模块。
- 可以使用西门子 S7-200 STEP7-MicroWin 编程软件，与西门子 S7-200 完全兼容；具有 Modbus TCP，S7-200 TCP，S7-300 TCP 协议，可以与 WINCC 直连（既无需使用 PC ACCESS 作为 OPC 连接 ），组态王，力控等主流的上位机相连接。
- 可以实现 PLC 之间的通讯（包括 S7-200 SMART、 CP243-1、CP243i、CP243-ibus、S7-300、S7-1200、S7-1500，使用的是 S7 PUT/GET 命令）
- 具有自由口通讯功能，如 Modbus RTU 主站、从站，USS 变频器通讯等
- 适用于 C++、delphi、C#、VB 等高级语言编程通讯（使用 Modbus TCP 协议）
- 可以连接西门子精彩系列 SMART LINE 触摸屏（Smart 1000IE 和 Smart 700IE ）
- 具有 PID 功能（但暂不支持参数自整定）。

目前它取消了 2 个命令：

（1）PLS：脉冲输出和脉冲计数输出

（2）HSC：高速脉冲计数指令

断电保持寄存器的有效范围对 V 区做了缩减，只可以对 VB0-VB2499 具有断电保持功能，而大于 VB2499 部分则没有断电保持功能(此存储区总数为 2500 还可用 VB1000-VB3499 或 VB1000-VB2000+VB3000-VB4499 这类使用）。

需要注意的是 强制输出在断电以后没有保存功能，重新上电以后取消强制

技术参数：

供电电源：标准工业 24VDC

安装方式：DIN35mm 标准导轨安装

尺寸 W x H x D (mm)：45x100x81

防护等级：IP20

网口通讯速率：100Mbps

A07 型 PLC——最新升级

升级内容: 增加 modbus 主从站非编程，以填表方式实现（简单实用）

使用填表方式时，有 modbus 主站和 modbus 从站两种选项

1.modbus 从站方式：

只需填写波特率，校验方式，从站地址即可完成

modbus 地址与 S7-200PLC 的数据对应关系如下：

00001-00128　　Q0.0、 Q0.1 、Q0.2 ..... Q15.7

10001-10128　　I0.0 、 I0.1 、 I0.2 ..... I15.7

30001-30032　　AIW0、AIW2、AIW4..... AIW62

4000n-4xxxx　　VW(n)、VW(n+2)、VW(n+4) .....

例 1：modbus 起始地址 8 、个数 3 对应 PLC 的 V 区为 VW8 、VW10、VW12

例 2： modbus 起始地址 19、个数 4 对应 PLC 的 V 区为 VW19、VW21、VW23、VW25

2.modbus 主站方式：

首先填写波特率、校验方式、等待从站应答时间、命令重发次数（是指 modbus 命令发送后，如果没有收到从站的正确应答，是发送下一条 modbus 命令，还是再次或多次发送本条命令）

主站方式可以有最多 64 条 modbus 命令，它通过在表中双击鼠标来添加或修改 modbus 命令行来轻松实现编程，这些命令从上致下按顺序不断循环发送执行。

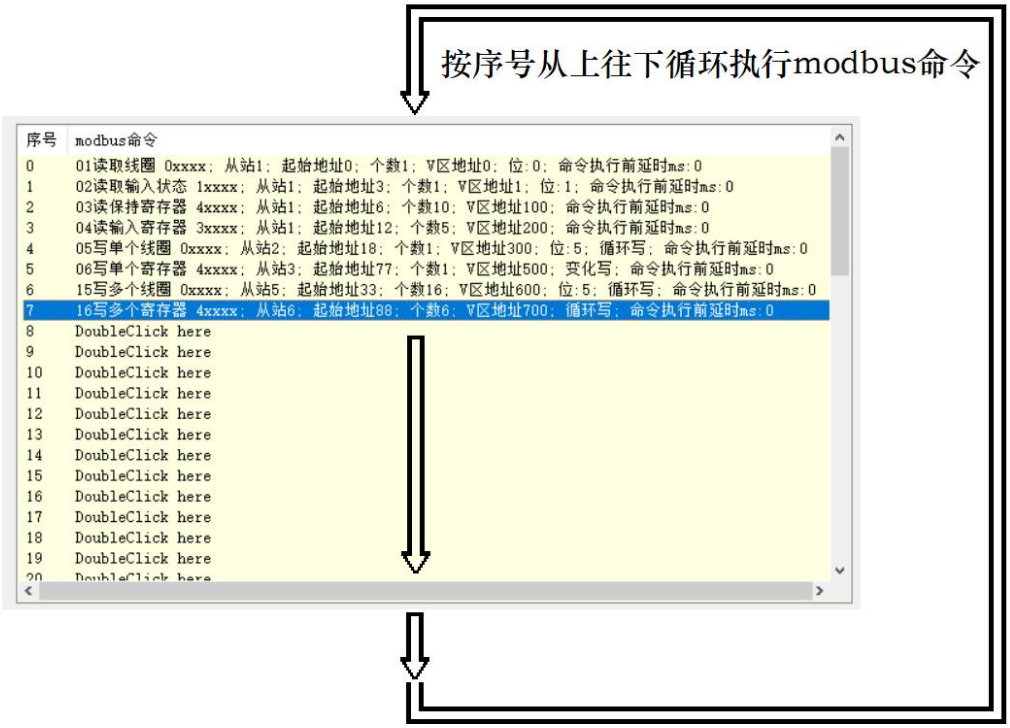
每条 modbus 命令中唯一要说明的是“命令执行前延时 ms”,它是指该命令执行前要延时一段时间，主要用于给从站一个缓冲时间，一般情况下是无需延时的，填写“0”即可。

2 Modbus 通讯（填表方式：简单方便快捷）

Modbus RTU 通讯设置软件下载：modbus_edit（右键点击下载）

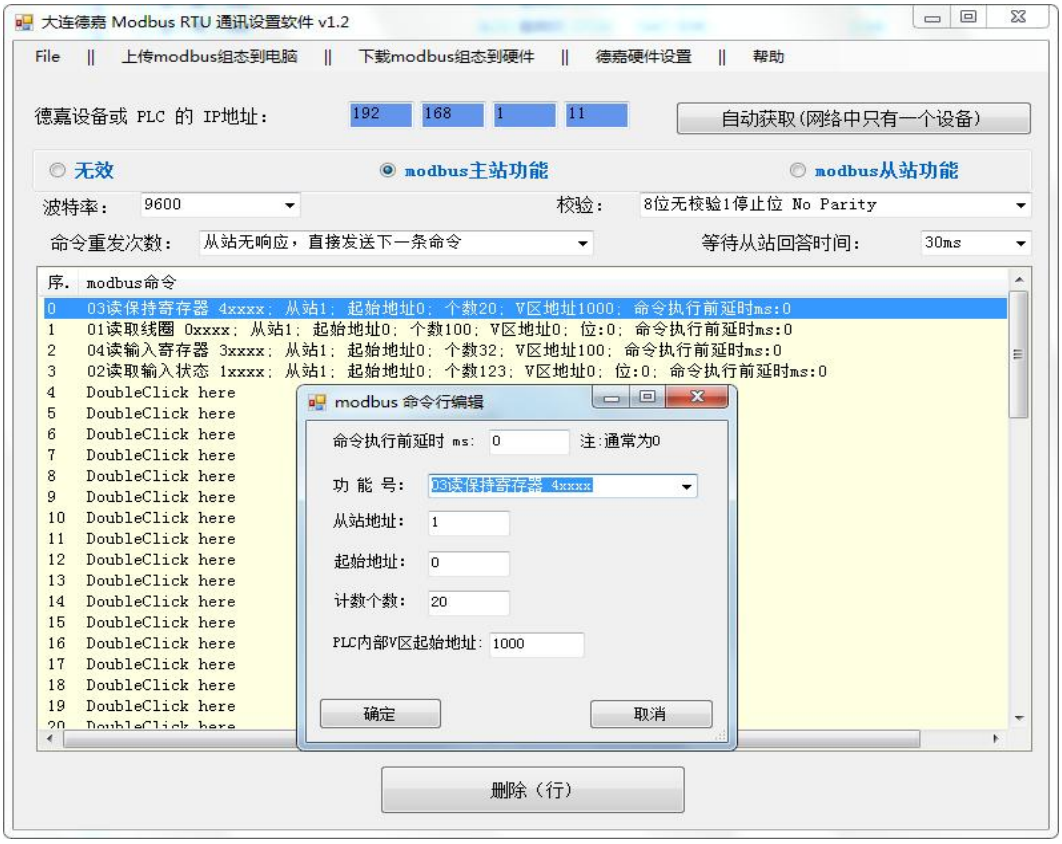
http://www.dl-winbest.com/download/modbus_edit.rar

下面为 Modbus 命令从上往下循环执行的方式示意图：

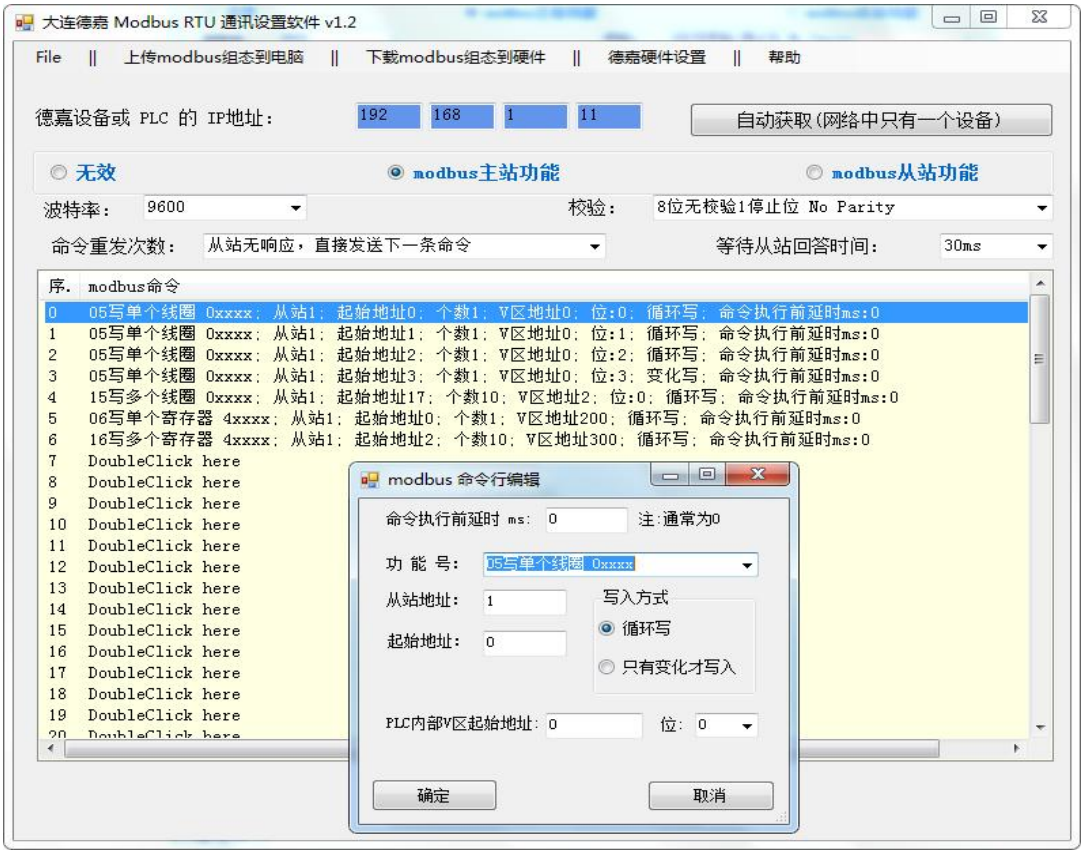


下图是 Modbus 作为从站功能的相关设置参数：

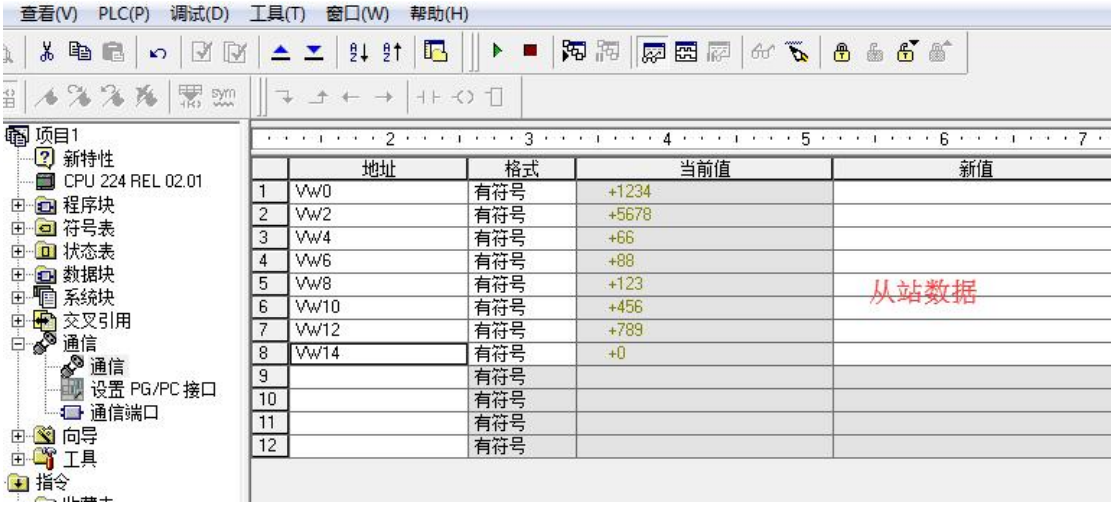
下图是 Modbus 作为主站（读）功能的相关设置参数：



下图是 Modbus 作为主站（写）功能的相关设置参数：



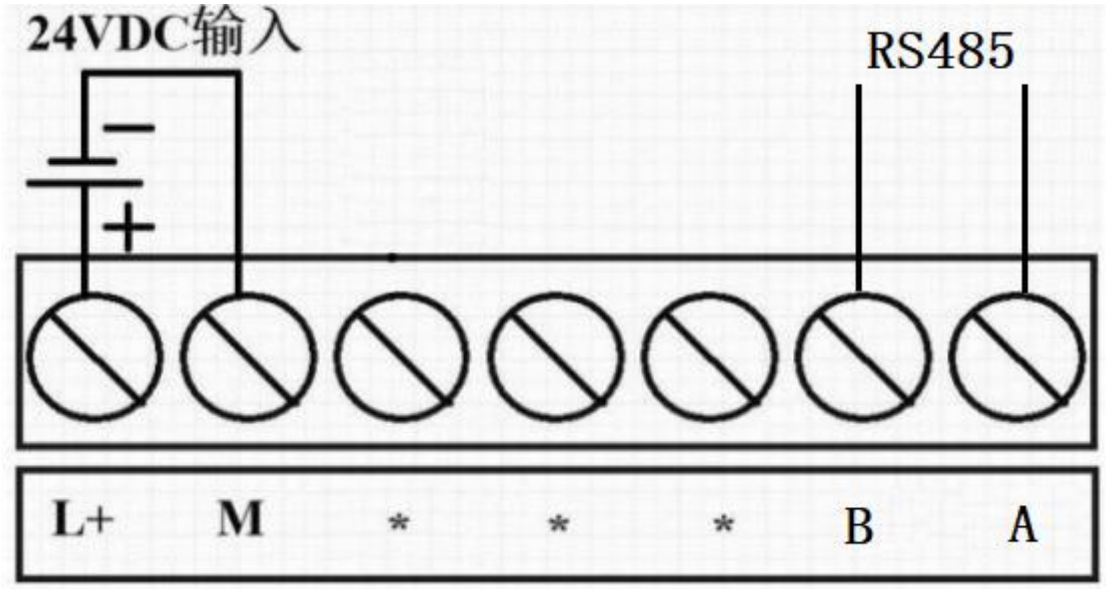
以两个 PLC 之间的 Modbus 通讯为例，一个 PLC 做从站，保持寄存器 4xxxx、从站地址 1、Modbus 起始地址 0；一个 PLC 做主站（读）的方式，功能码为 03 读保持寄存器 4xxxx、从站地址 1、计数个数 20、V 区起始地址 1000，监控数据如下：



可见已经完成了 Modbus 的通讯。

3 参数设置

端子接线图


24VDC输入
RS485
L+
M
*
*
*
B
A

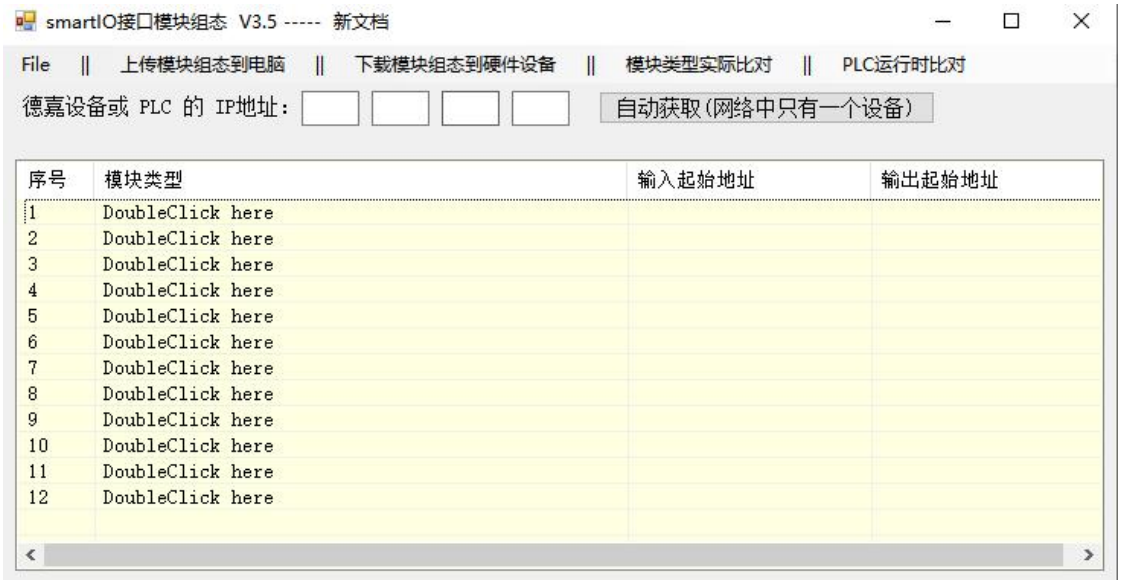

X10
L+ M
24 VDC
B A
RS485
DIAG
RUN
STOP
CPU A07

LED 指示灯说明

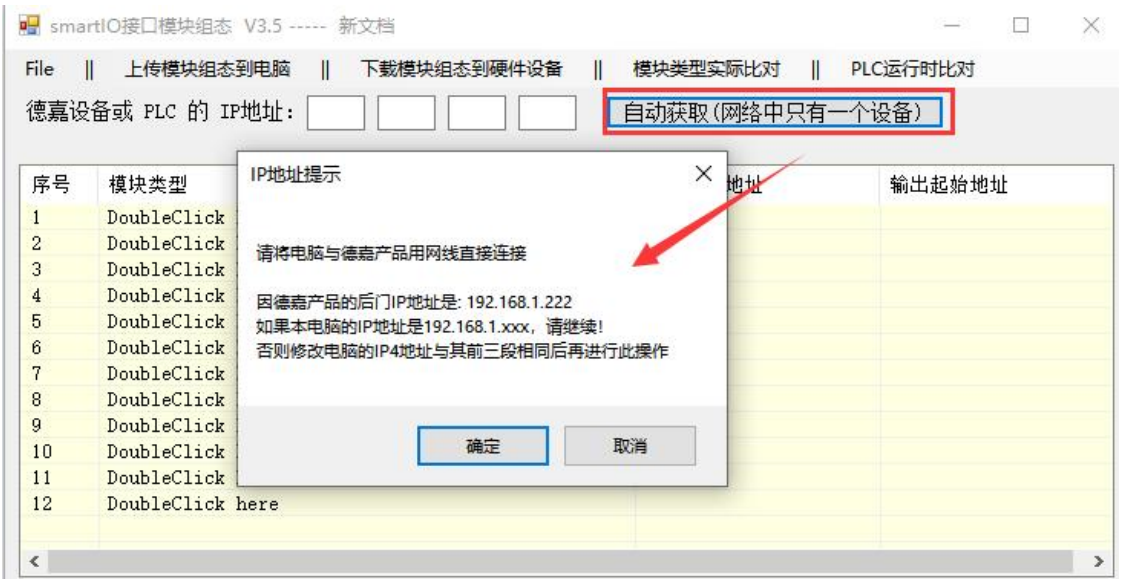
LED 指示灯			含义
DIAG（诊断）	RUN（运行）	STOP（停止）	
灭	灭	灭	PLC 电源电压缺失或不足
红（常亮）	绿（常亮）	绿（常亮）	PLC 未接 IO 模块
红（闪烁）	-	-	PLC 在组态插件里组态错误/无任何组态
-	绿（闪烁）	-	PLC 处于运行状态（后接 IO 模块状态时）
-	-	绿（闪烁）	PLC 处于停止状态（后接 IO 模块状态时）
绿（闪烁）	-	-	PLC 处于强制状态（后接 IO 模块状态时）

IO 模块组态插件参数配置

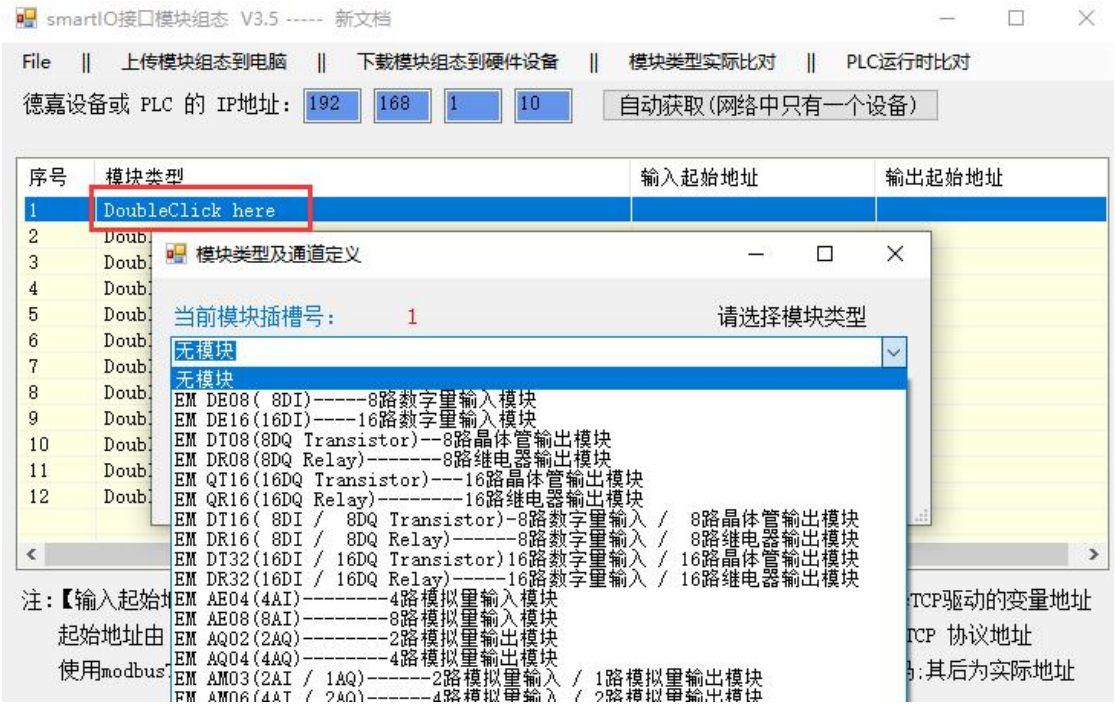
（1）下载 IO 模块组态插件：**点击下载**



（2） 点击自动获取 IP 地址，如：192.168.1.10

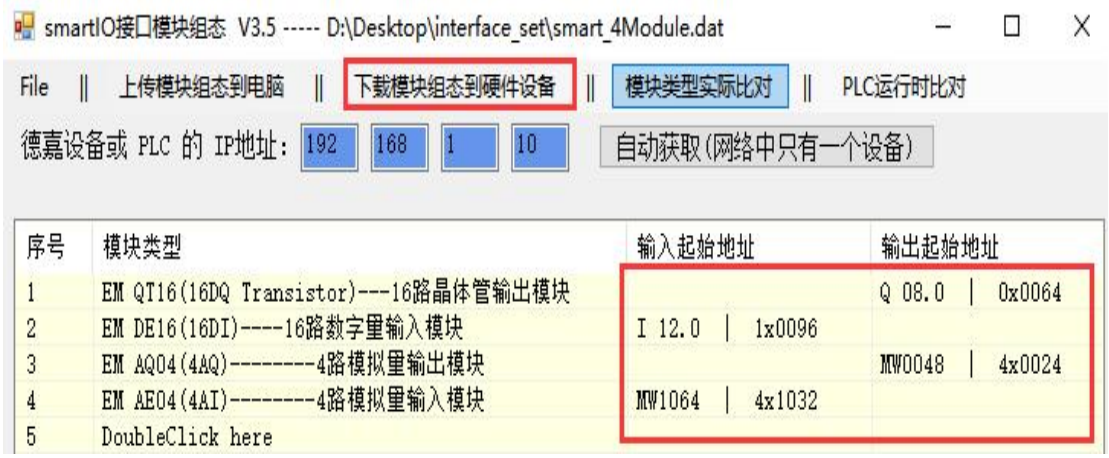


（3） 双击序号 1 槽位，添加组态 PLC 实际连接的 200Smart IO 模块类型



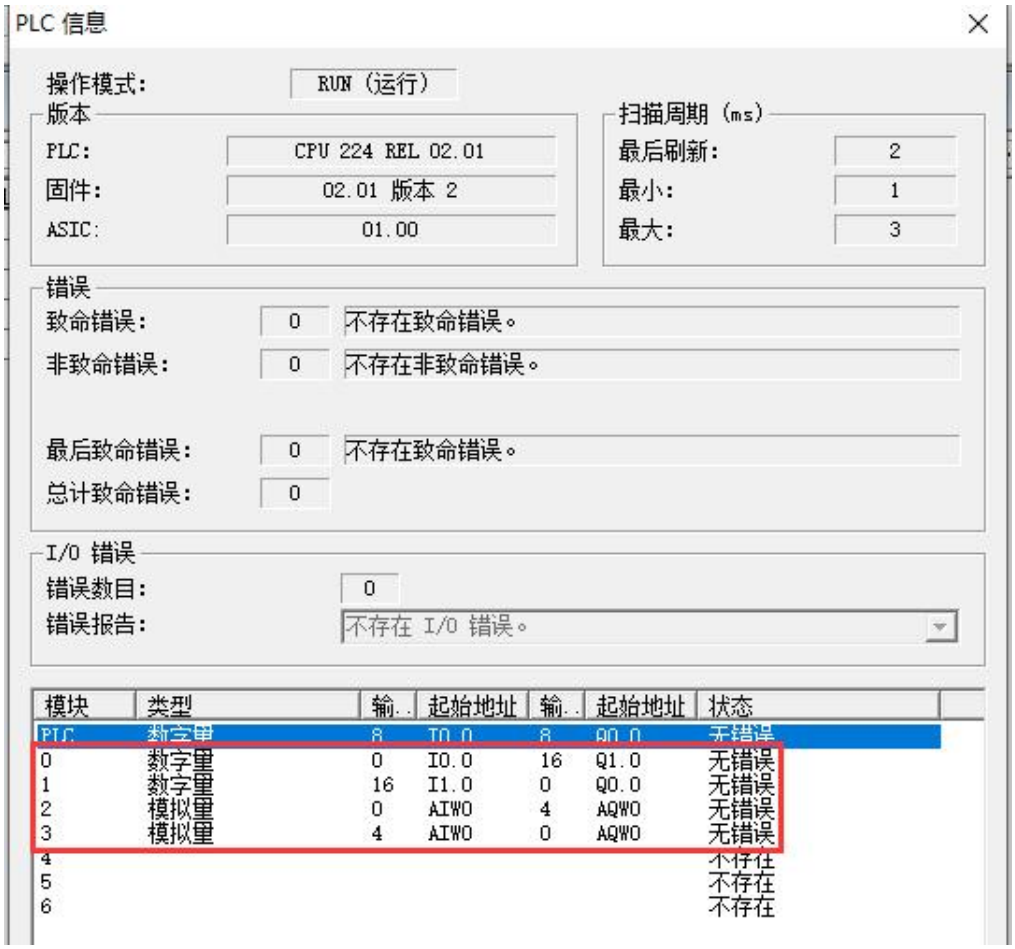
（4）选择好模块类型，可以通过插件设置具体参数，如：EM AQ04 具体参数配置

（5）将接口模块实际连接的 Smart IO 模块组态好后，点击“下载模块组态到硬件设备”

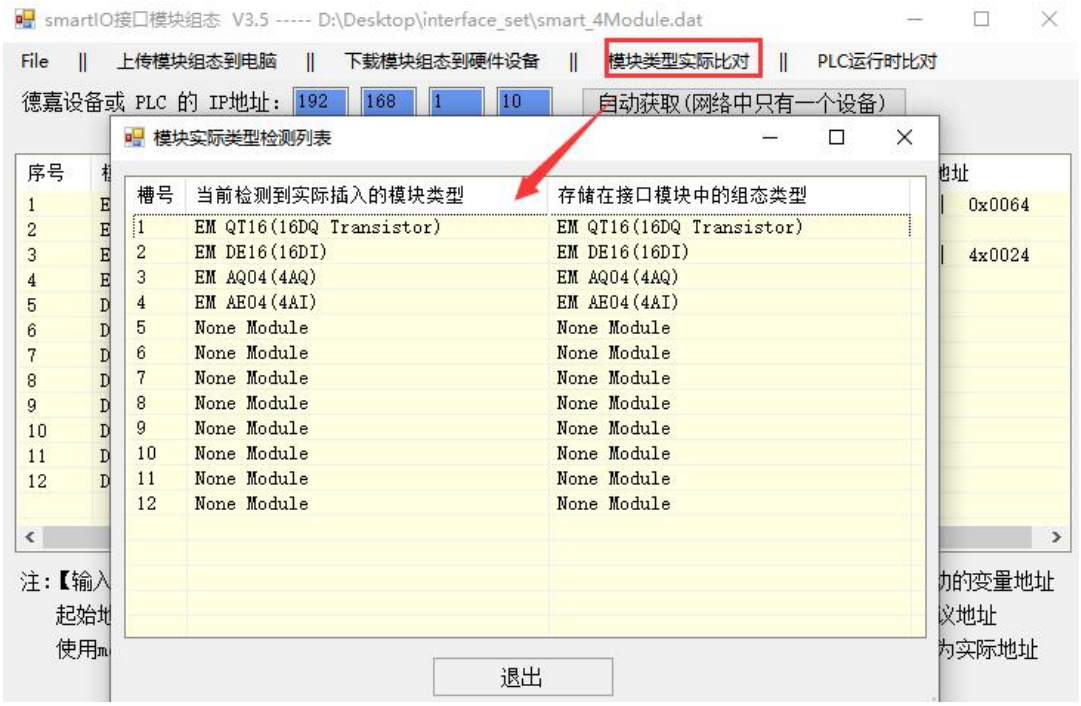


此红色框格内的 IO 地址不用做编程地址使用，以编程软件 IO 地址为准

实际各个模块 IO 地址通过菜单栏“PLC”-->“信息”查找 PLC 信息为准

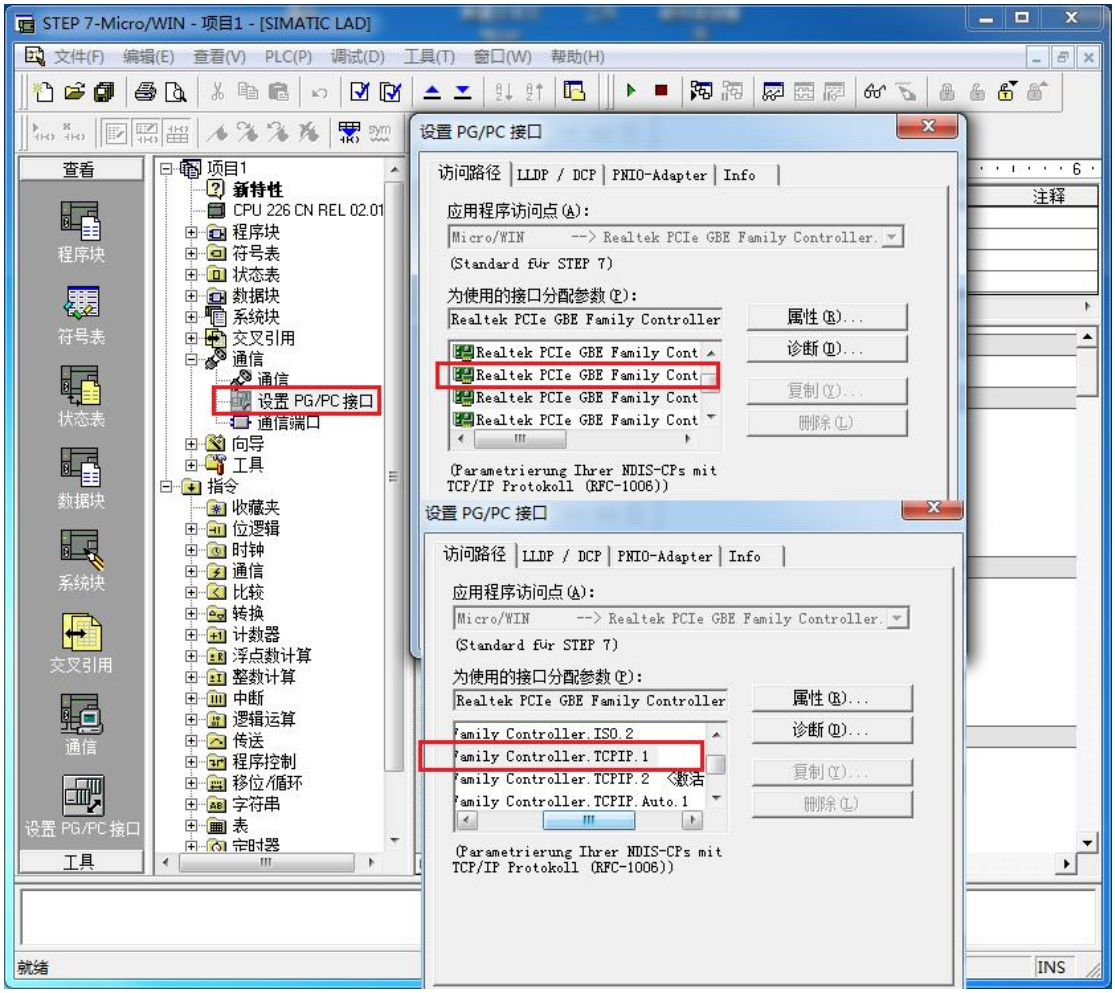


(6) 点击“模块类型实际比对”，可以比对当前实际插入的模块类型和接口模块组态类型 是否一致



4 MicroWin 连接设置

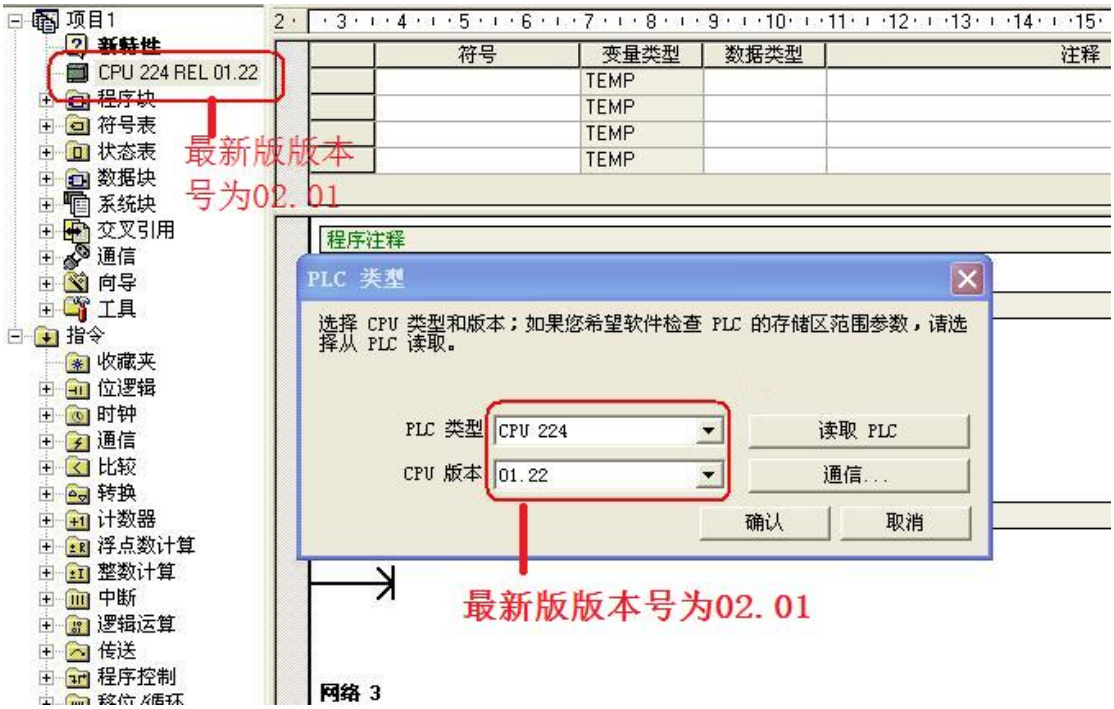
1. 打开 MicroWin，双击[设置 PG/PC 接口]，选择如下图驱动（网卡名.TCPIP.1），选好后点击确定.



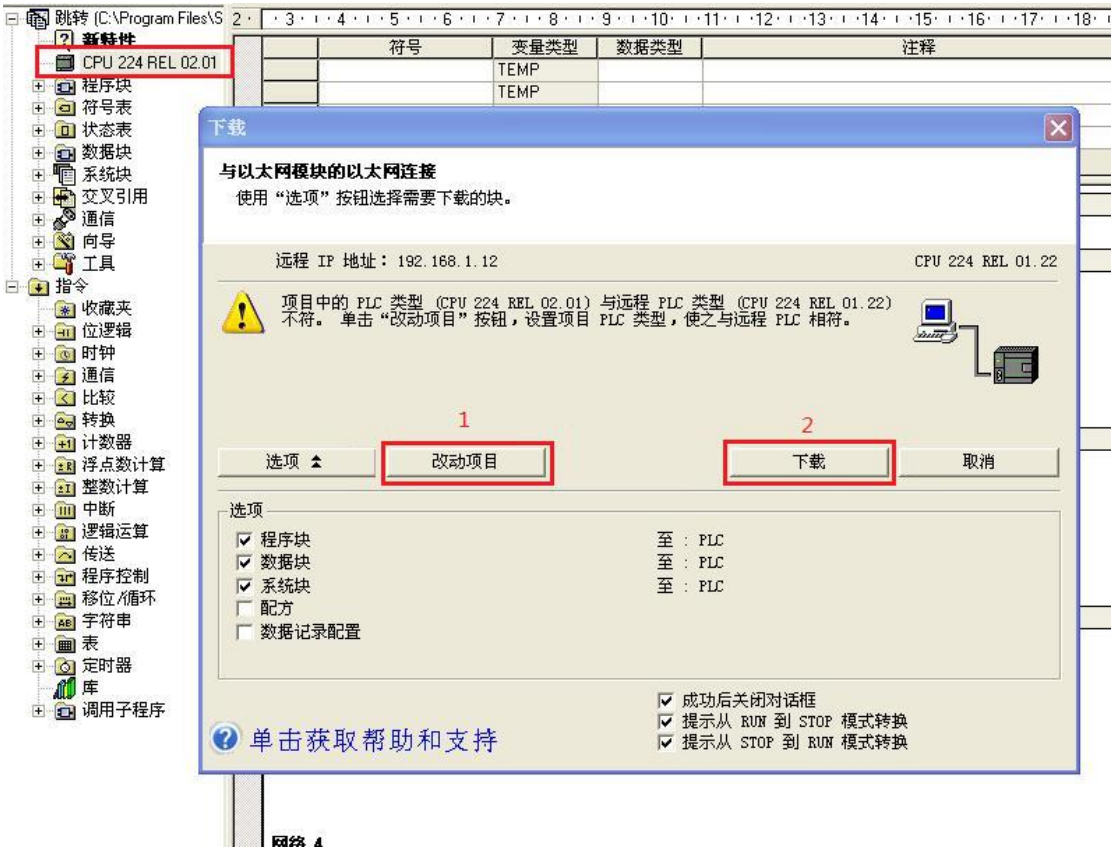
2. 在下图中双击[通信]，在“远程：”框中填入该 CPU 的 IP 地址，如 192.168.1.10，然后点击确认

MicroWin 设置完毕，现在就可以用 MicroWin 对大连德嘉该 PLC 进行编程了！

注意事项：下载项目时，一定要注意 PLC 的型号，如图：



1. 这个型号必须是 CPU 224REL 01.22，如果不是请右键 PLC 型号选择类型改成如图型号。如果型号不符时点击下载会出现如图提示：



这里请选择改动项目，然后再点击下载即可。如果没有点击改动项目直接下载则可能出现如图情况：

此时我们需要清除 PLC 如图：

先选择 PLC 中的清除选项，出现如图界面点击清除即可。

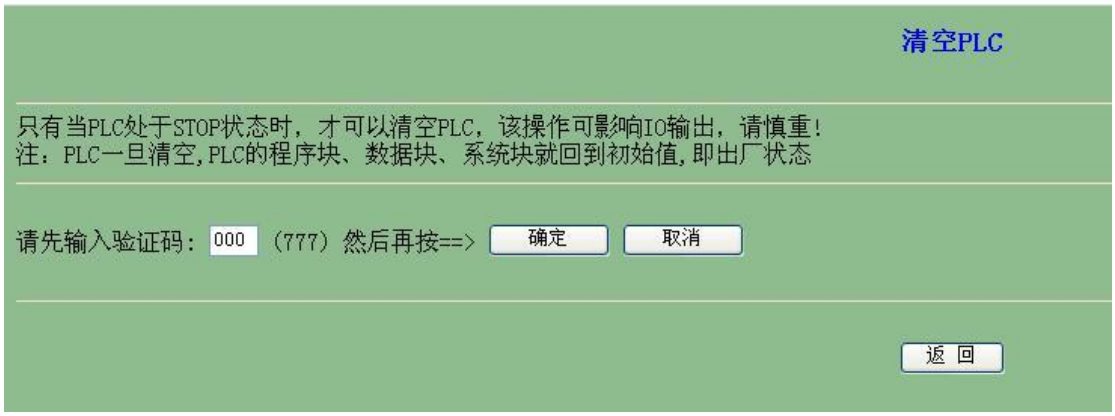
当然，我们也可以使用 ie 浏览器中的网页来清除 PLC

首先我们在 IE 浏览器地址栏中输入 192.168.1.222（这个是后门地址，并不是实际地址，仅用于设置参数）进入设置界面：

这里是语言选项，我们选择 Chinese，进入下一界面：

请按照这三步操作：首先，如果 PLC 没有停止，则先选择 STOP PLC：

输入验证码 888 后点击确定即可。之后在上一界面选择清空 PLC：



输入 777 后点击确定即可。清空后可能会使 PLC 的 IP 清零，我们需要重新进入设置页面设置 IP 地址：

这里它的段址一定要与相连的计算机相同即前三项相同第四相不同。
例一：计算机 IP(192.168.1.100)，掩码(255.255.255.0)，网关(192.168.1.1)，PLC

的 IP(192.168.1.10) 。

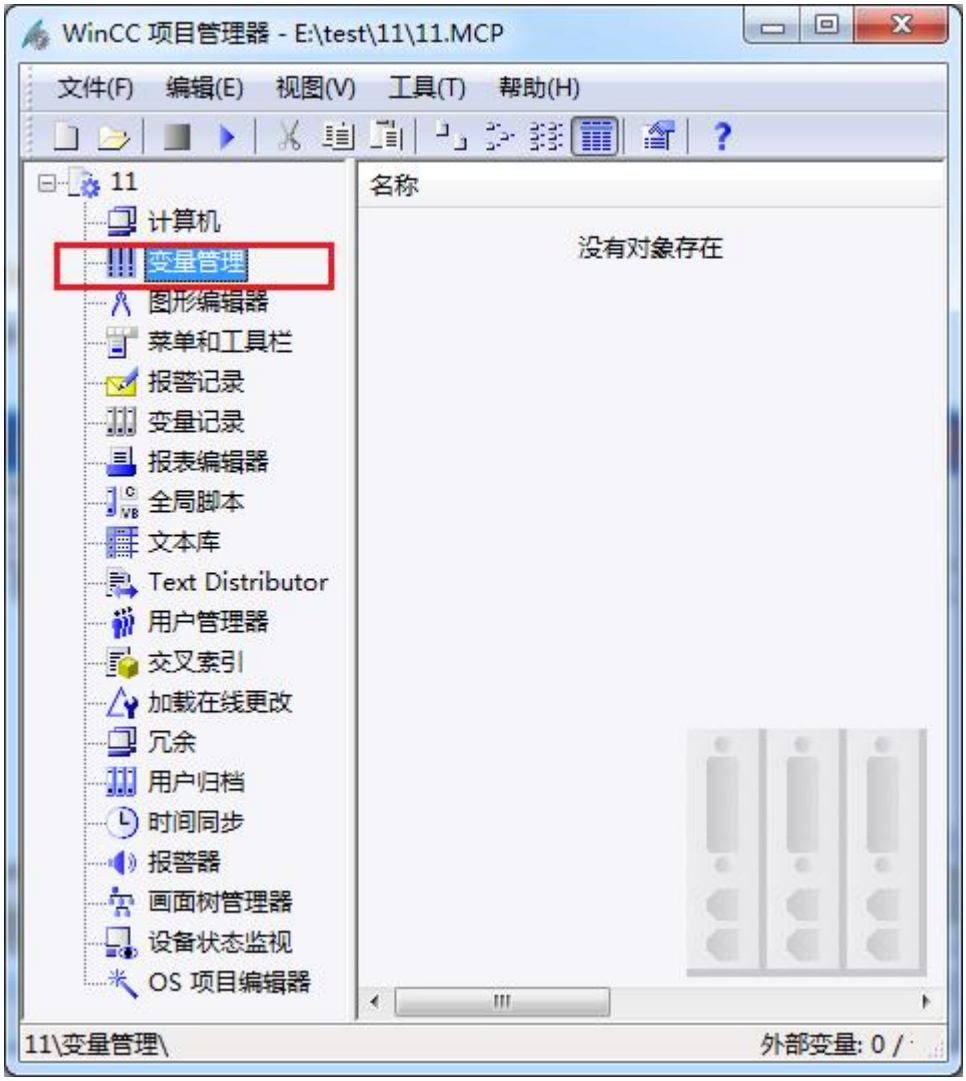
注：参数设置提交后，最好在 cmd 窗口键入 arp -d（删除计算机中已保留的 IP/MAC 表），以便

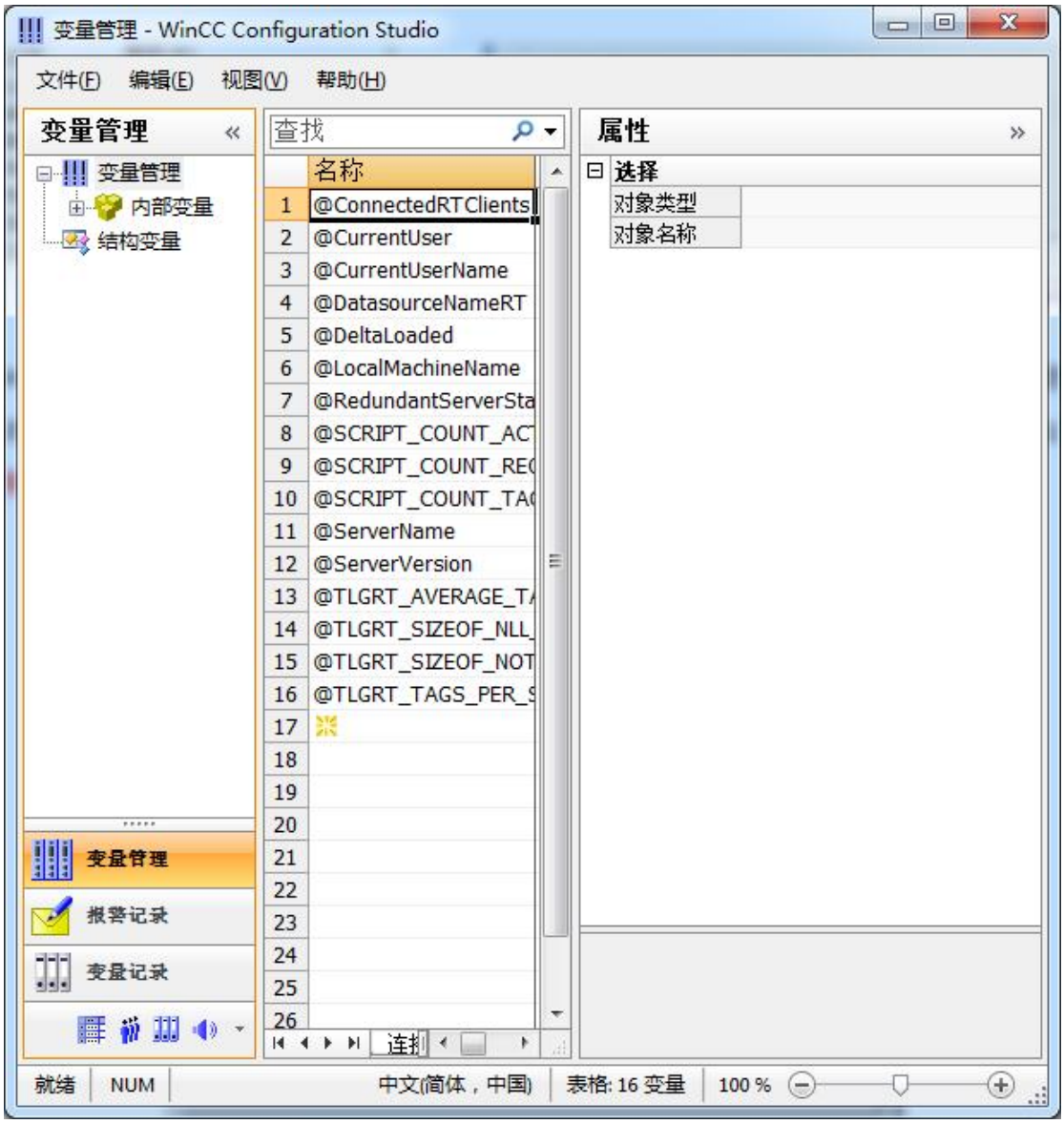
PLC 新改动的 IP/MAC 与老地址无冲突 。

系统块中红框里面的功能有所保留，蓝框里面的功能暂不支持

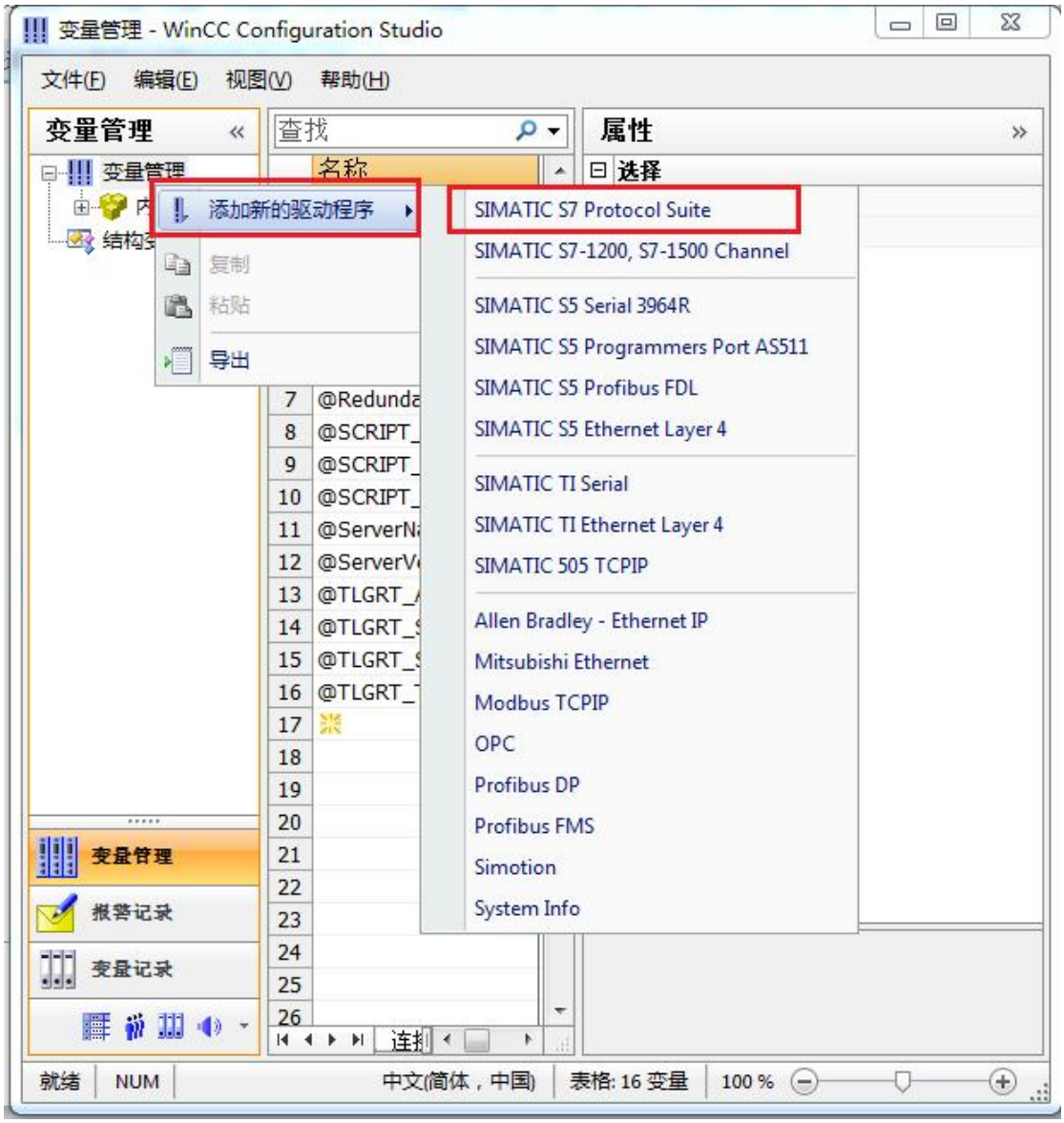
5 WinCC 连接设置（以 WinCC7.3 为例）

1. 打开 Wincc，双击变量管理，打开变量管理器，添加驱动：

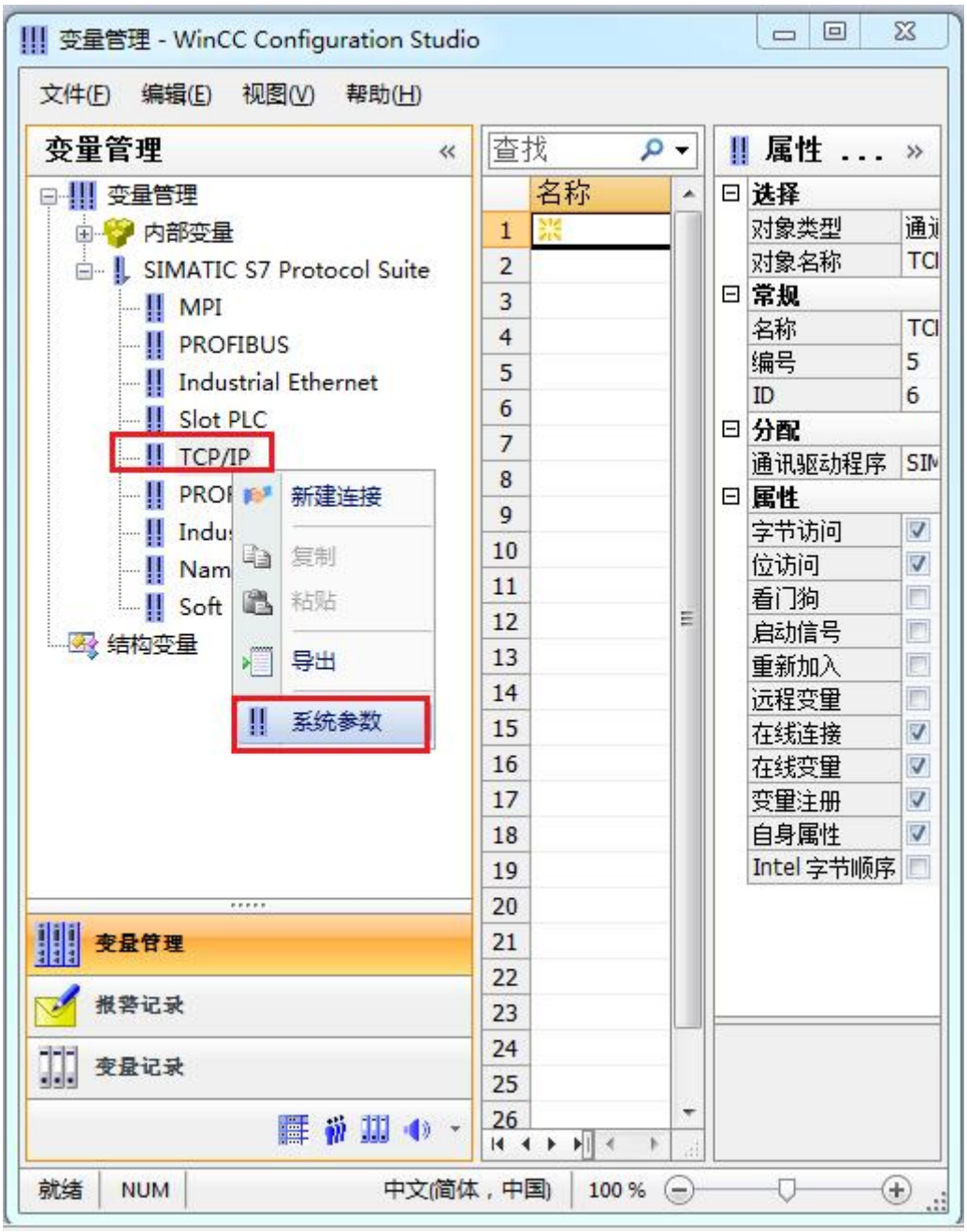



变量管理 - WinCC Configuration Studio
文件(F) 编辑(E) 视图(V) 帮助(H)
变量管理
变量管理
内部变量
结构变量
查找
名称
1 @ConnectedRTClients
2 @CurrentUser
3 @CurrentUserName
4 @DatasourceNameRT
5 @DeltaLoaded
6 @LocalMachineName
7 @RedundantServerSta
8 @SCRIPT_COUNT_AC
9 @SCRIPT_COUNT_RE
10 @SCRIPT_COUNT_TA
11 @ServerName
12 @ServerVersion
13 @TLGRT_AVERAGE_T
14 @TLGRT_SIZEOF_NLL
15 @TLGRT_SIZEOF_NOT
16 @TLGRT_TAGS_PER_S
17
18
19
20
21
22
23
24
25
26
连接
属性
选择
对象类型
对象名称
变量管理
报警记录
变量记录
就绪
NUM
中文(简体，中国)
表格: 16 变量
100 %

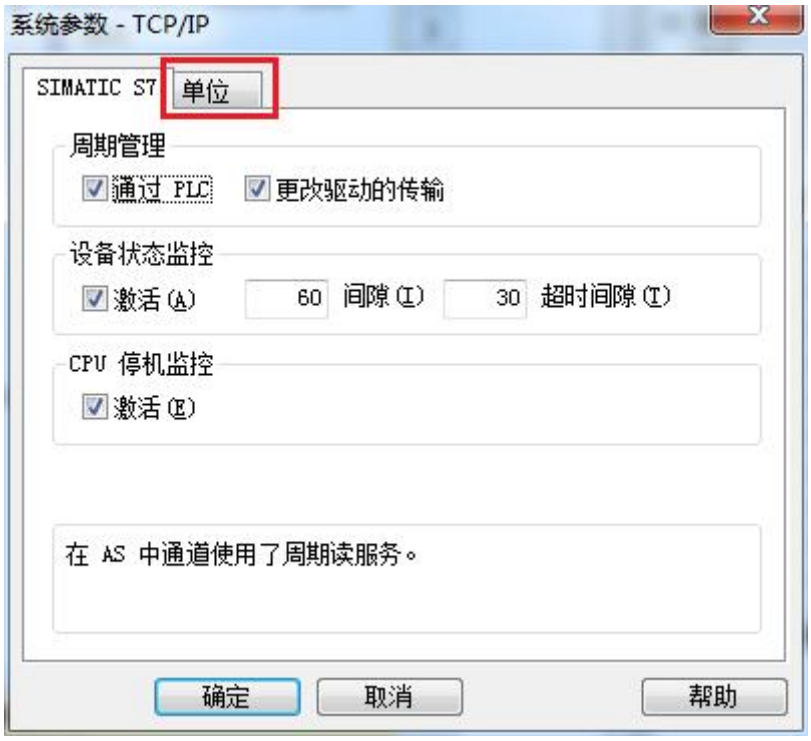
2. 填右键单击变量管理，在弹出的菜单中选择添加驱动，SIMATIC S7 Protocol Suite，如下图所示



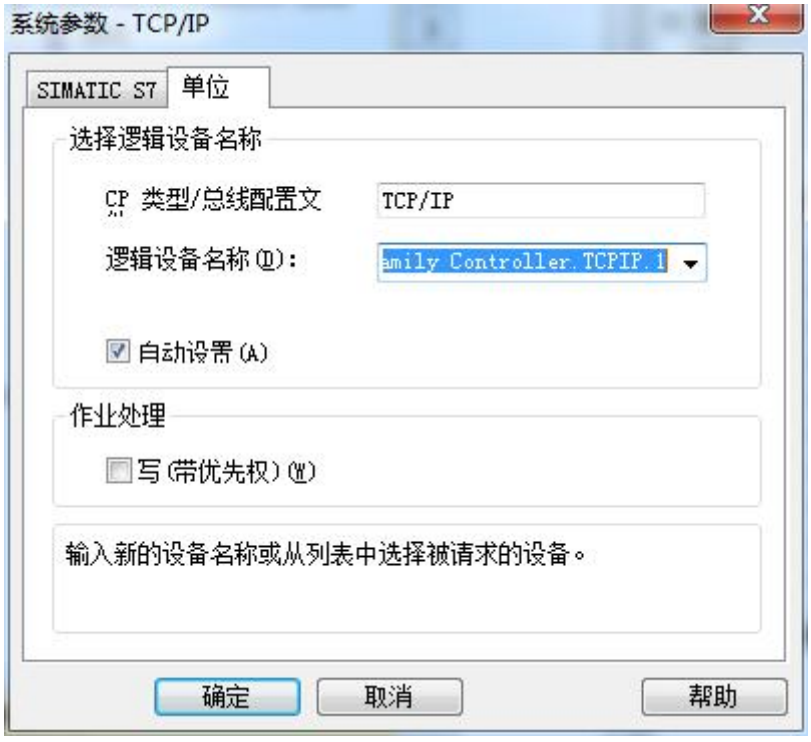
3. 添加好驱动之后，右键单击 SIMATIC S7 Protocol Suite 下的 TCP/IP，在弹出的菜单中选择系统参数



4. 在弹出的对话框中点击单位选项卡



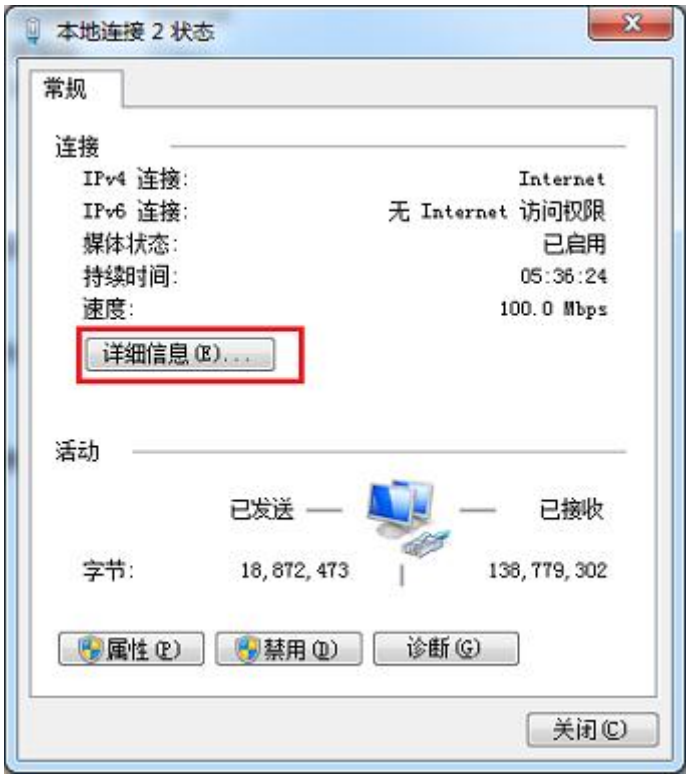
5. 在逻辑设备名称选框中选择驱动为：网卡名. TCPIP. 1



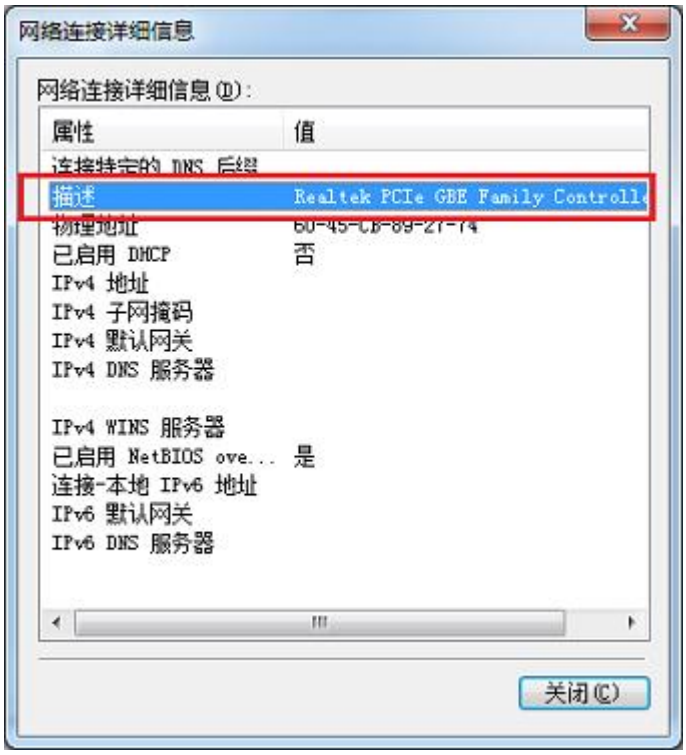
如何查看网卡名：点击屏幕右下角的电脑图标，选择打开网络和共享中心

在网络共享中心中点击本地连接

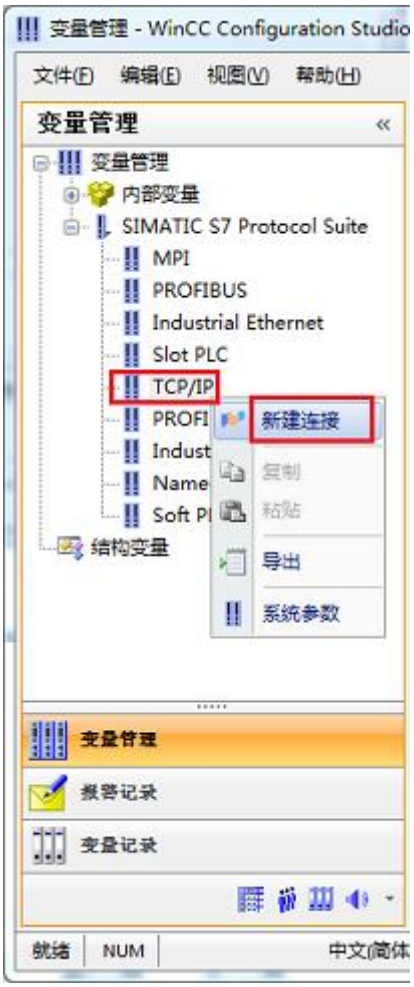
在弹出的对话框中点击详细信息

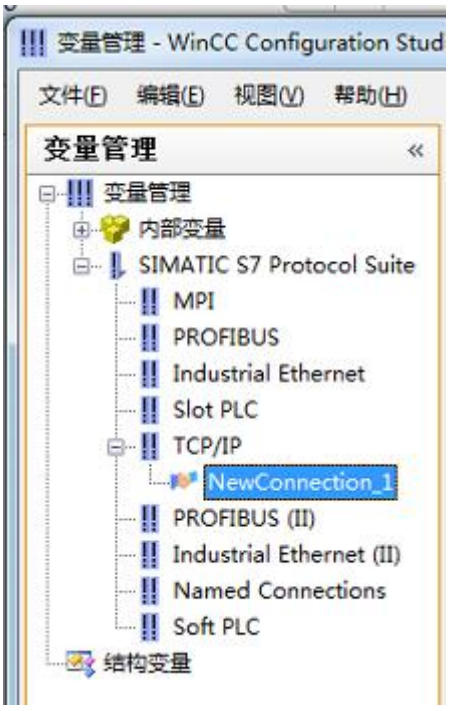


下图中的描述内容就是你的网卡名

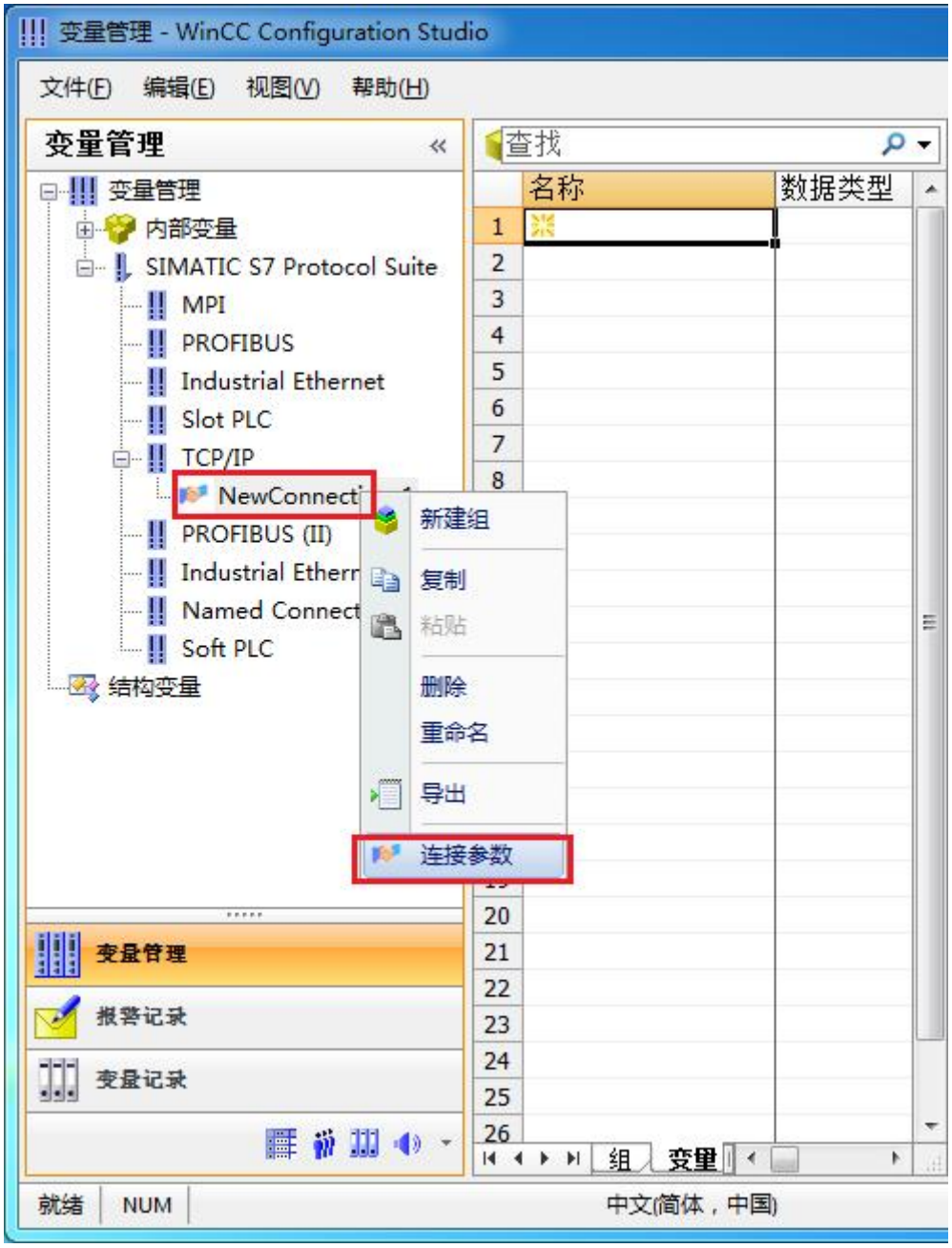


6. 再回到变量管理器中，右键点击 TCP/IP，选择新建连接，在 TCP/IP 选项下会生成一个名为 NewConnection_1 的新连接选项。





7. 右键单击 NewConnection_1，在弹出的菜单中选择



8. 在弹出的对话框中填写 M02 的 IP 地址，192.168.1.10

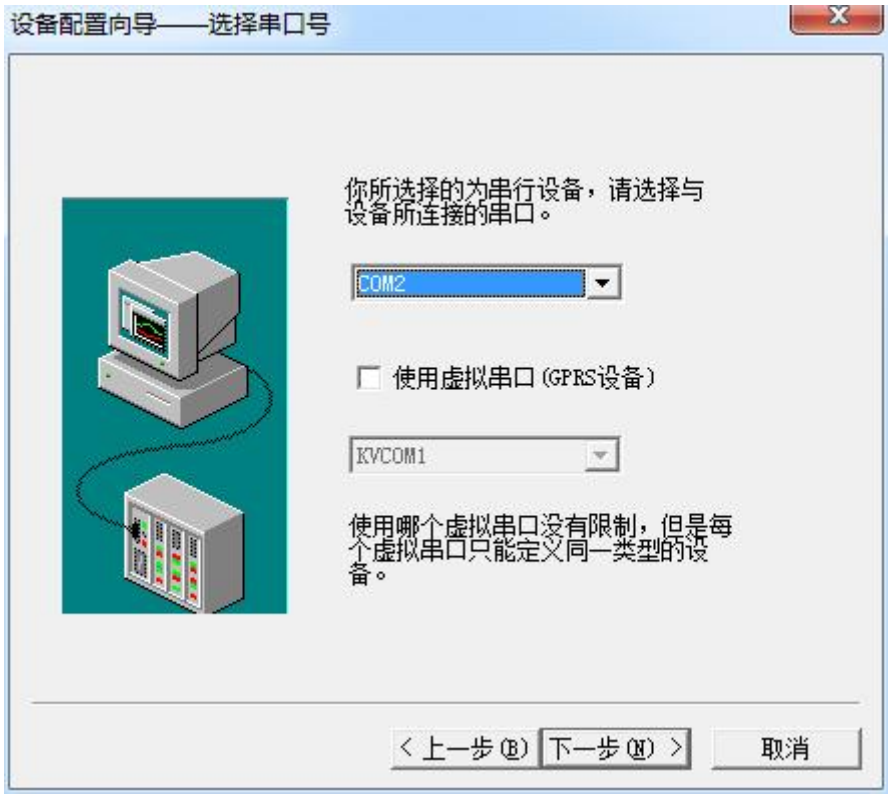
现在连接已经建立成功，已经可以建立变量和画面了。

6 组态王连接设置

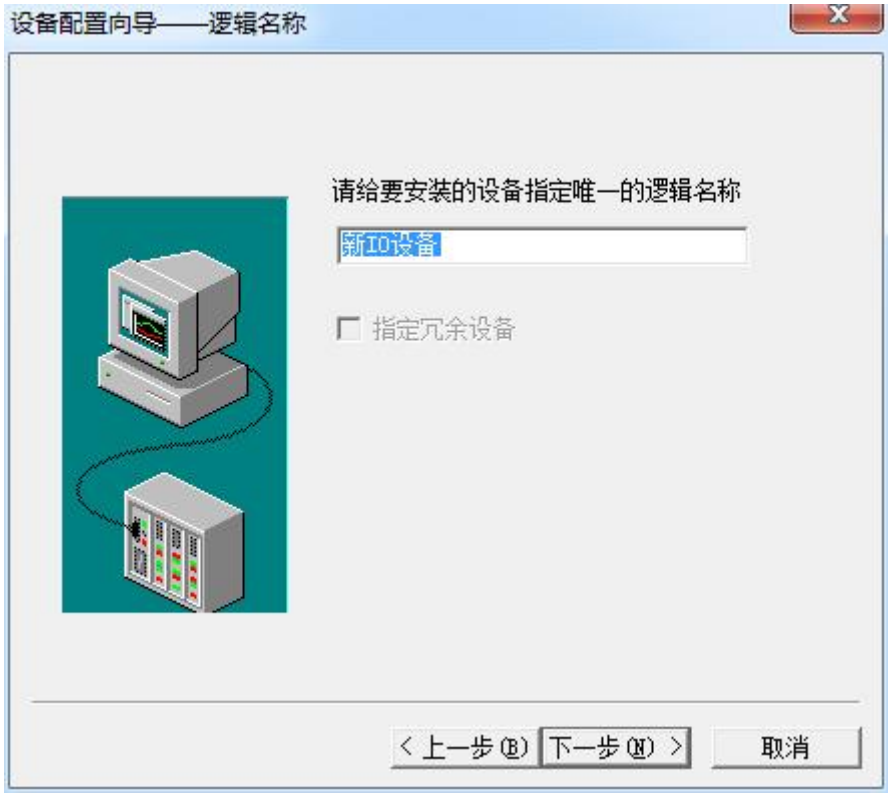
1. 打开组态王开发软件，选择设备→COM1

2. 双击“新建”，选择 S7-200 系列（TCP）→TCP

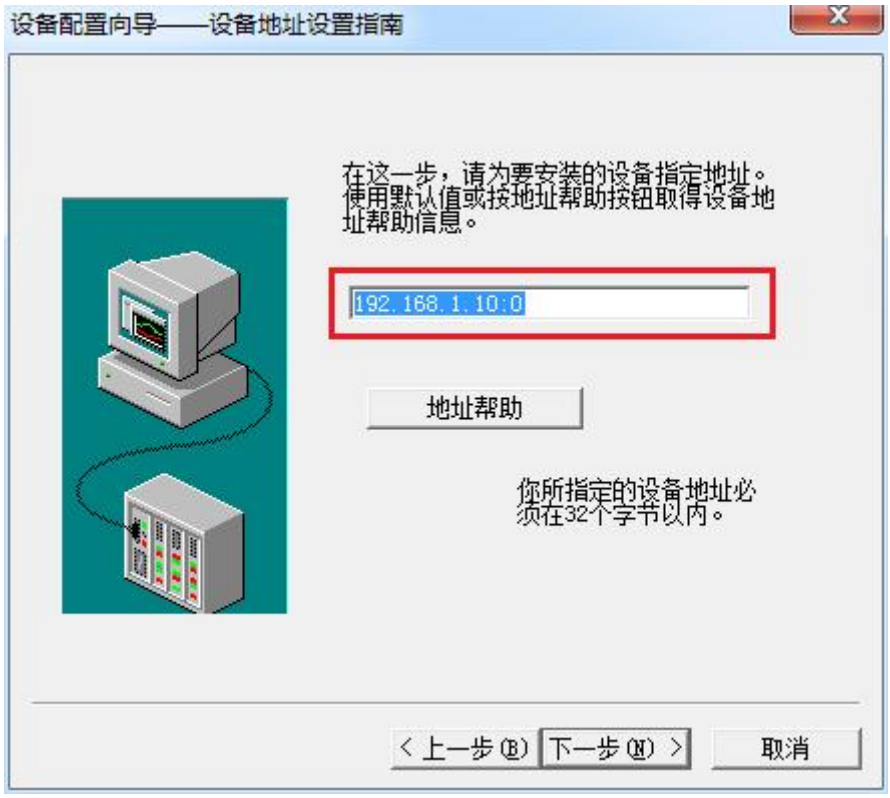
3. 选择 com 口号，此处选择默认值 com2



4. 单击“下一步”，输入要安装的设备的逻辑名称



5. 再单击“下一步”，输入设备的 IP 地址及相对于 PLC 的位置



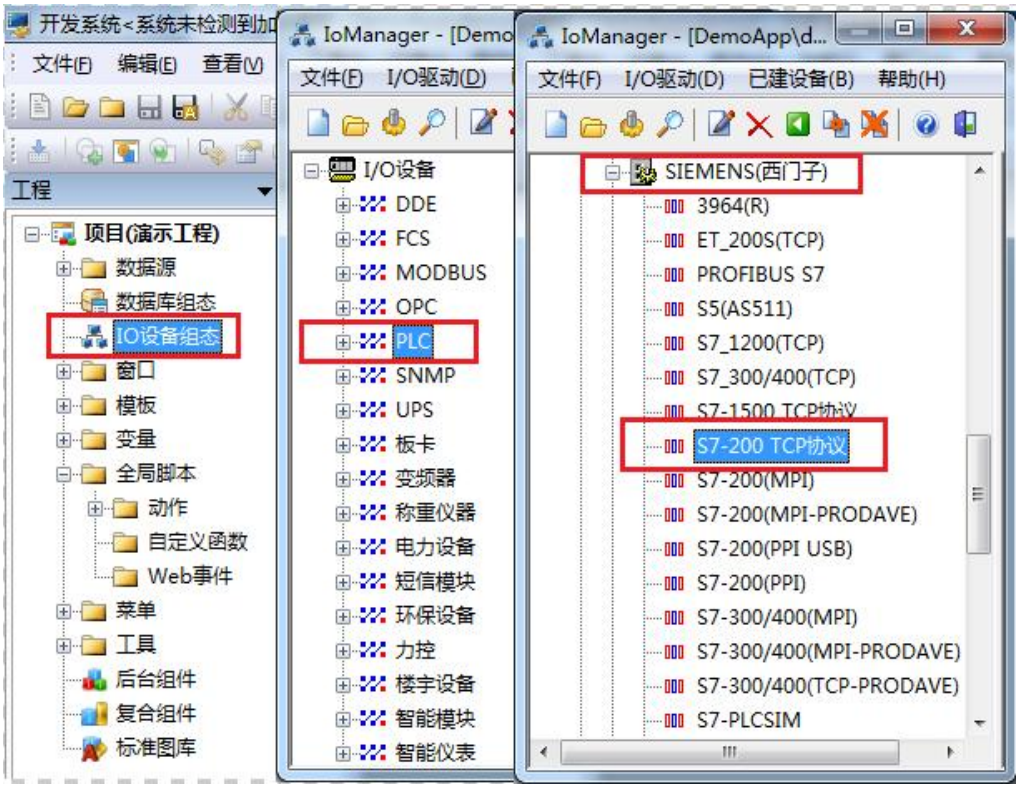
6. 再单击“下一步”，保持默认值，直接单击“下一步”

7. 单击“完成”，就配置了一个“TCP”设备。

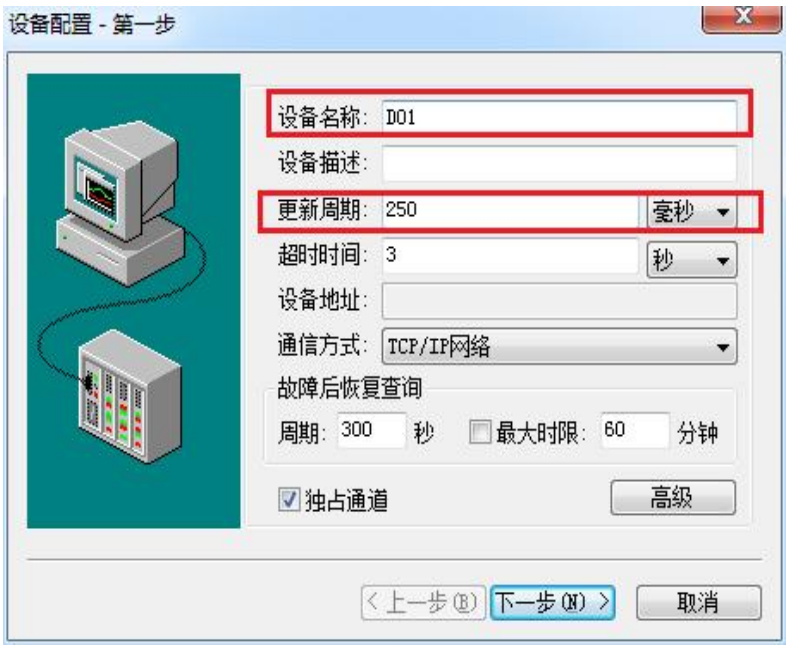
至此，就完成了 PLC 与组态王的连接。

7 力控连接设置

1. 打开组态软件，进入开发系统，打开“IO 设备组态”->“PLC”->“SIEMENS”->“S7-200 TCP 协议”，画面如下：

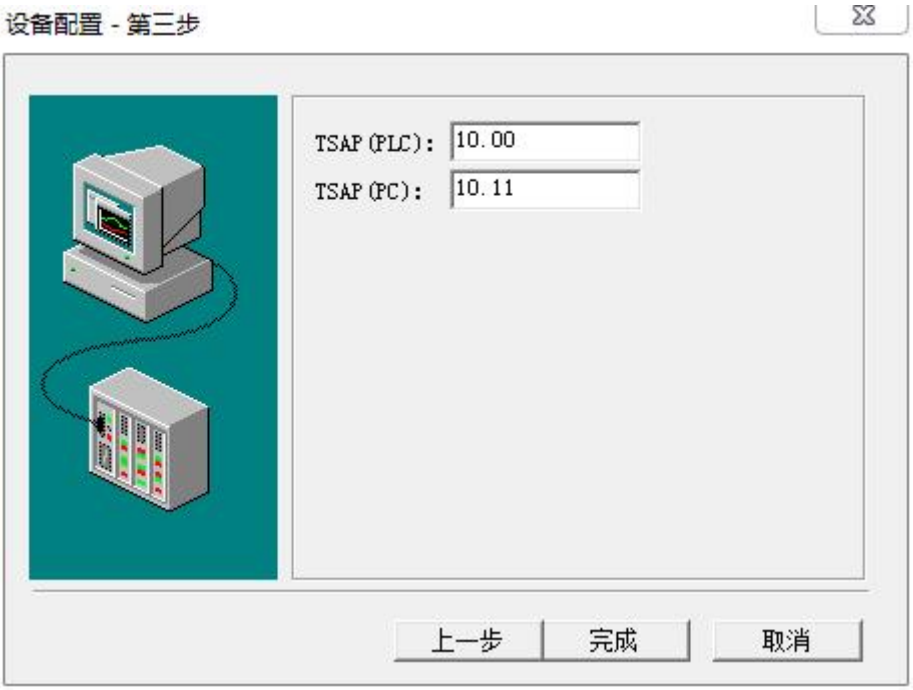


2. 第一步：基本参数配置，定义设备名称，修改更新周期。（更新周期一定要修改为 250 毫秒以上！）



3. 第二步：通讯参数。设备 IP 地址：192.168.1.10，端口号：102

4. 点击完成，现在你的 PLC 可以与力控软件连接了。



8 连接 SMART LINE 参数设置

1、在触摸屏上设置好触摸屏的 IP 地址，如 192.168.1.200

2、在 SIMATIC WinCC flexible 2008，给触摸屏编程，如下图所示

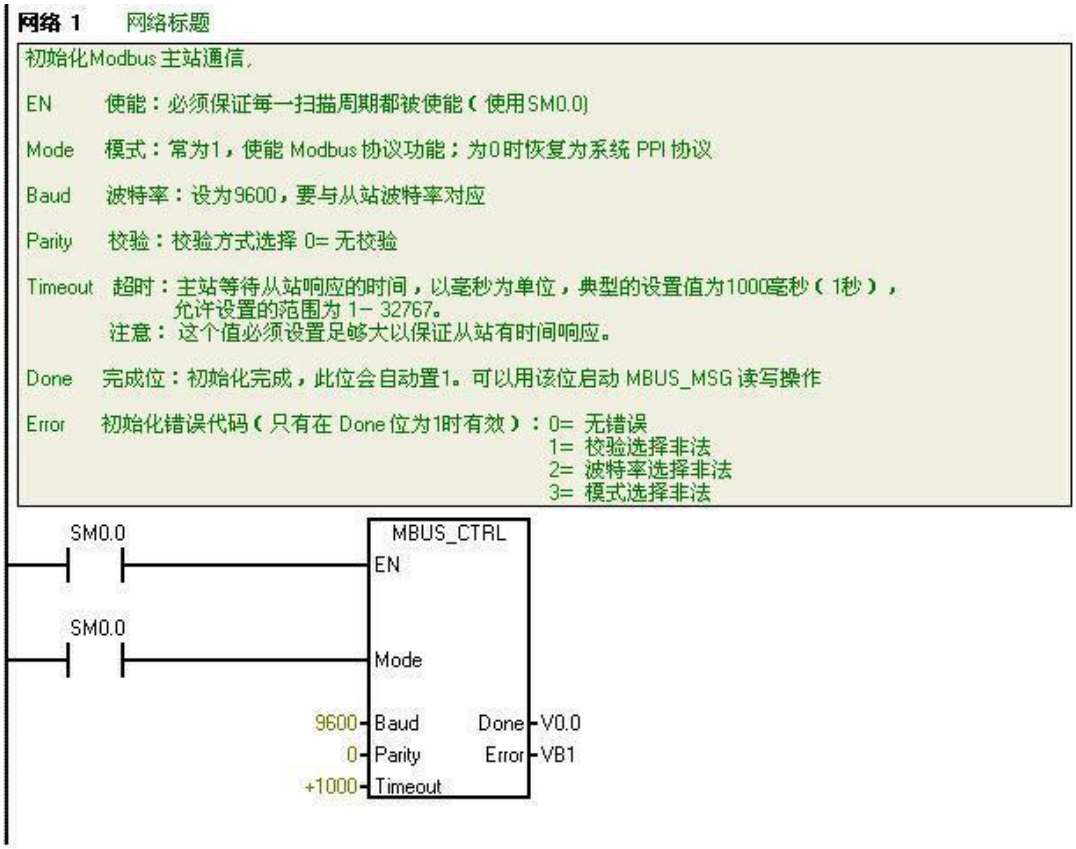
9 Modbus 通讯（梯形图方式）

想要进行 modbus 通讯必须安装 modbus 指令库，指令库大家可以网上下载一个，这里就不提供了。

接线：本例是在两个 S7-200 CPU 的接线端子 AB 接线柱进行的 modbus 通讯，两个 CPU 通过 AB 线进行连接（仪器仪表为 AB 线分别接 AB 接线柱）。主站编程时，调用主站 port0 端口程序。

这个例子能实现的功能是读取从站 40001 地址开始的 10 个字，存到主站 VB400 开始的 10 个字。

主站程序如下：



网络 2

读取从站保持寄存器的数据

EN　　使能：同一时刻只能有一个读写功能（即 MBUS_MSG）使能
注意：建议每一个读写功能（即 MBUS_MSG）都用上一个 MBUS_MSG 指令的 Done 完成位来激活，以保证所有读写指令循环进行（见程序）。

First　　读写请求位：每一个新的读写请求必须使用脉冲触发

Slave　　从站地址：可选择的范围 1– 247

RW　　读写操作：0= 读， 1= 写
注意：1. 开关量输出和保持寄存器支持读和写功能
2. 开关量输入和模拟量输入只支持读功能

Addr　　读写从站的数据地址：选择读写的数据类型 00001至0xxxx– 开关量输出
10001至1xxxx– 开关量输入
30001至3xxxx– 模拟量输入
40001至4xxxx– 保持寄存器

Count　　通讯的数据个数（位或字的个数）
注意： Modbus主站可读/写的最大数据量为120个字（是指每一个 MBUS_MSG 指令）

DataPtr　　数据指针：1. 如果是读指令，读回的数据放到这个数据区中
2. 如果是写指令，要写出的数据放到这个数据区中

Done　　读写功能完成位

Error　　错误代码 只有在 Done 位为1时，错误代码才有效
错误代码： 0= 无错误
1= 响应校验错误
2= 未用
3= 接收超时（从站无响应）
4= 请求参数错误（ slave address,Modbus address,count,RW）
5= Modbus/自由口未使能
6= Modbus正在忙于其它请求
7= 响应错误（响应不是请求的操作）
8= 响应CRC校验和错误

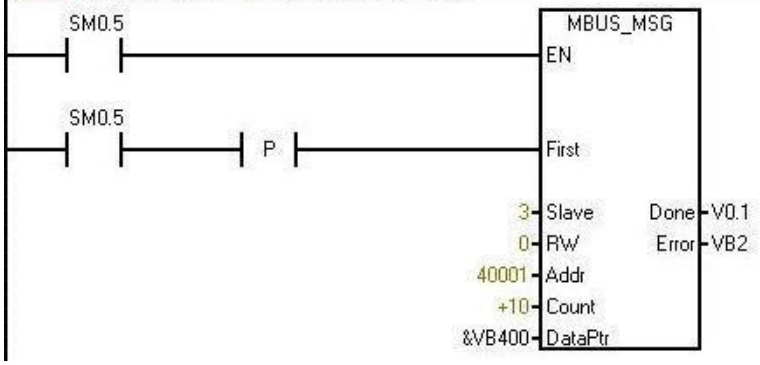
101= 从站不支持请求的功能
102= 从站不支持数据地址
103= 从站不支持此种数据类型
104= 从站设备故障
105= 从站接受了信息，但是响应被延迟
106= 从站忙，拒绝了该信息
107= 从站拒绝了信息
108= 从站存储器奇偶错误

常见的错误及其错误代码：

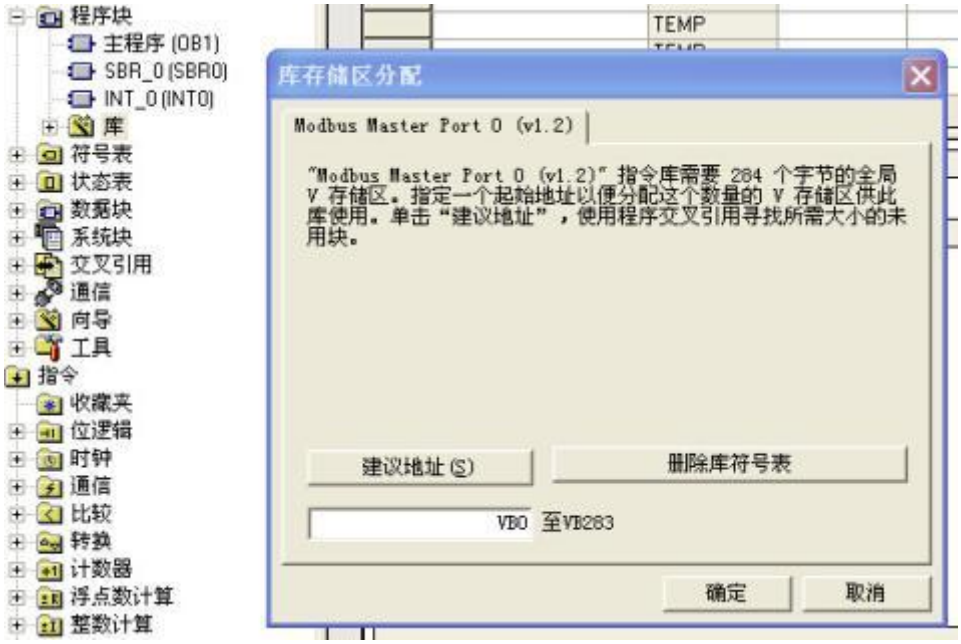
1. 如果多个 MBUS_MSG 指令同时使能会造成6号错误

2. 从站 delay 参数设的时间过长会造成3号错误

3. 从站掉电或不运行，网络故障都会造成3号错误



我们要注意的是需要分配库存储区地址，如下图：



这段寄存器地址不能再被程序中的任何指令使用，包括 MBUS_INIT 和 MBUS_SLAVE 指令在内。

从站程序如下：

网络 1 网络标题

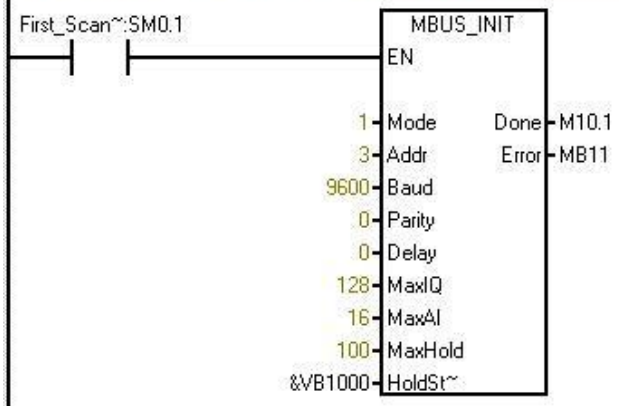
在第一个循环周期内初始化Modbus从站协议

Mode: 模式选择，启动/停止MODBUS，1=启动；0=停止
Address: 从站地址，MODBUS从站地址，取值1~247
Baud: 波特率，可选1200，2400，4800，9600，19200，38400，57600，115200
Parity: 奇偶校验，0=无校验；1=奇校验；2=偶校验
Delay: 延时，附加字符间延时，缺省值为0
MaxIQ: 最大I/Q位，参与通信的最大I/O点数，S7-200的I/O映像区为128/128，缺省值为128
MaxAI: 最大AI字数，参与通信的最大AI通道数，可为16或32
MaxHold: 最大保持寄存器区，参与通信的V存储区字（VW）
HoldStart: 保持寄存器区起始地址，以&VBx指定（间接寻址方式）

Done: 初始化完成标志，成功初始化后置1
Error: 初始化错误代码

在本例子中，设置从站地址为3
Port0通讯波特率为9600
无校验
可以使用的S7-200最大数字量输入输出点数为128
可以使用的S7-200最大模拟量输入寄存器字数为32
可以使用的V区寄存器地址字数为100，起始地址为VB1000

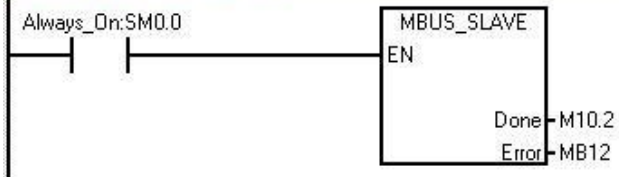
注意：本例子中，Modbus RTU 从站指令使用的库存储为VB0---VB779，这段寄存器地址不能再被程序中的任何指令使用，包括MBUS_INIT 和 MBUS_SLAVE 指令在内



网络 2

在每个循环周期内执行Modbus 从站协议

Done: MODBUS执行，通信中时置1，无 MODBUS 通信活动时为 0
Error: 错误代码：0=无错误



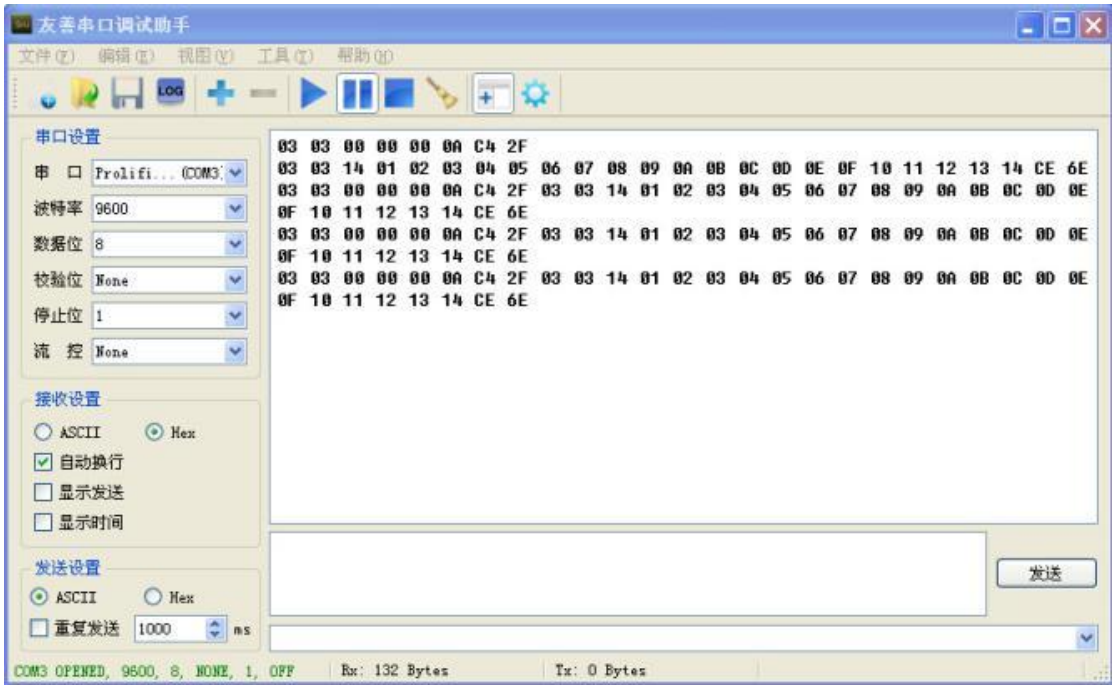
同样，从站也需要分配库存储区。之后把程序编译下载运行，这里我们给从站 VB1000 开始的 20 个字节赋值如下：

	地址	格式	当前值
1	VB1000	无符号	1
2	VB1001	无符号	2
3	VB1002	无符号	3
4	VB1003	无符号	4
5	VB1004	无符号	5
6	VB1005	无符号	6
7	VB1006	无符号	7
8	VB1007	无符号	8
9	VB1008	无符号	9
10	VB1009	无符号	10
11	VB1010	无符号	11
12	VB1011	无符号	12
13	VB1012	无符号	13
14	VB1013	无符号	14
15	VB1014	无符号	15
16	VB1015	无符号	16
17	VB1016	无符号	17
18	VB1017	无符号	18
19	VB1018	无符号	19
20	VB1019	无符号	20
21		有符号	

然后可以观察到主站 VB400 开始的 20 个字节如下：

	地址	格式	当前值
1	VB400	无符号	1
2	VB401	无符号	2
3	VB402	无符号	3
4	VB403	无符号	4
5	VB404	无符号	5
6	VB405	无符号	6
7	VB406	无符号	7
8	VB407	无符号	8
9	VB408	无符号	9
10	VB409	无符号	10
11	VB410	无符号	11
12	VB411	无符号	12
13	VB412	无符号	13
14	VB413	无符号	14
15	VB414	无符号	15
16	VB415	无符号	16
17	VB416	无符号	17
18	VB417	无符号	18
19	VB418	无符号	19
20	VB419	无符号	20
21		有符号	
22		有符号	

这就是一个简单的 200 之间的 modbus 通讯。下面我们在两台 PLC 中间加一个 USB 转 485 串口监视器，然后我们用串口调试助手可以看到例子执行时的请求码和响应码。



图中第一行为请求码，下面讲一下这个码是怎么来的。
我们例子中使用的是 03 功能码（读保持寄存器）：

请求

功能码	1 个字节	**0x03**
起始地址	2 个字节	0x0000 至 0xFFFF
寄存器数量	2 个字节	1 至 125（0x7D）

响应

功能码	1 个字节	**0x03**
字节数	1 个字节	2×N*
寄存器值	N*×2 个字节	

*N＝寄存器的数量

首先是请求：03 03 00 00 00 0A C4 2F

03 为串口号，接的是 COM 3

03 为功能码

00 00 为起始地址即 40001

00 0A 为寄存器数量，读 10 个字

C4 2F 为 CRC 校验码

响应：03 03 14 01 02 03 04 05 06 07 08 09 0A 0B 0C 0D 0F 10 11 12 13 14 CE 6E

03 为串口号

03 为功能码

14 为字节数，一共 20 个字节

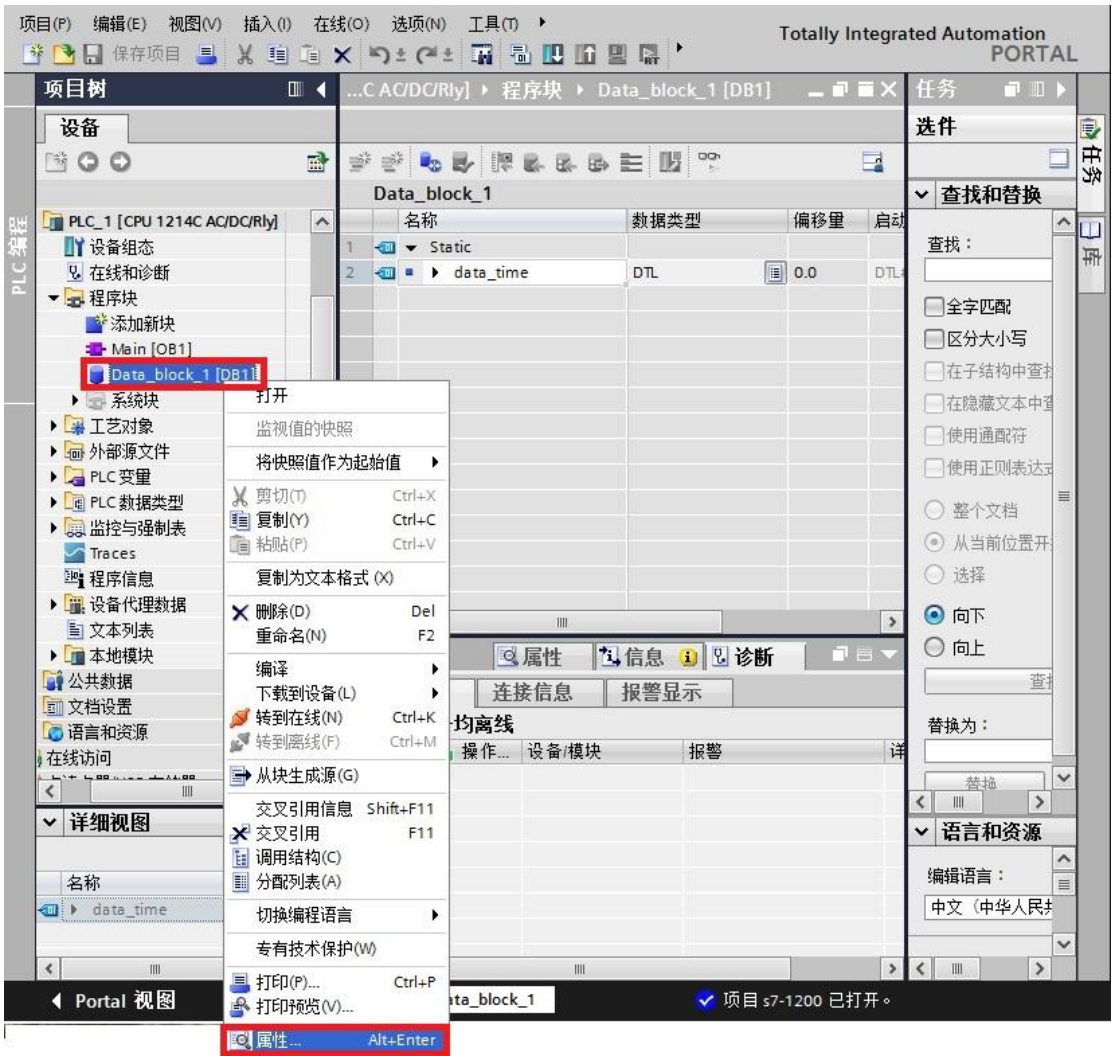
01～14 为寄存器中的值

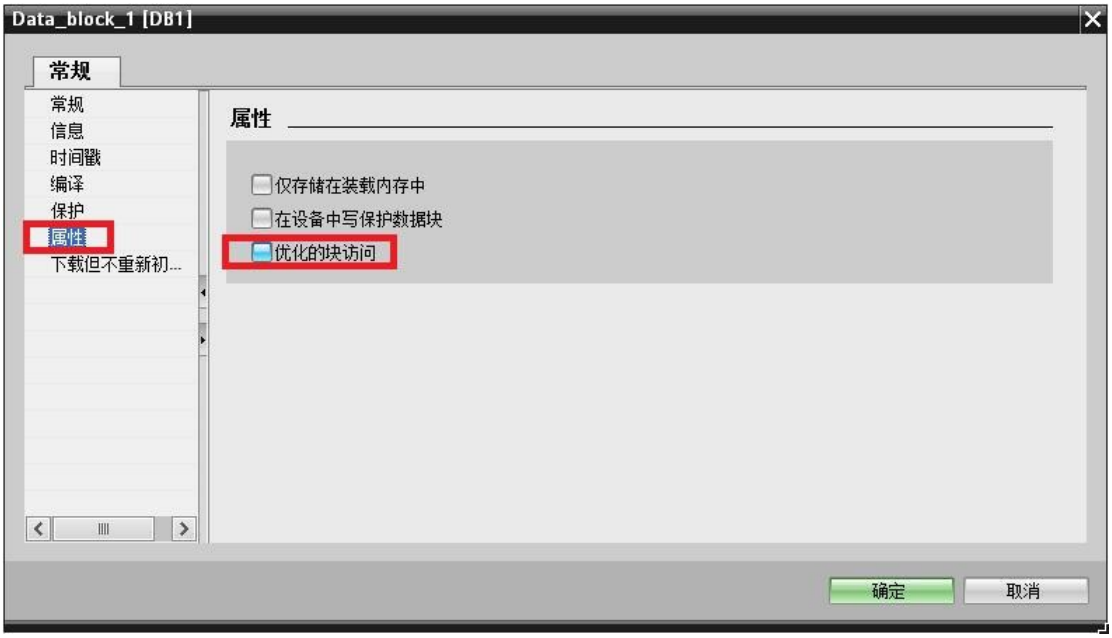
CE 6E 为 CRC 校验码

10 PLC 之间通讯设置

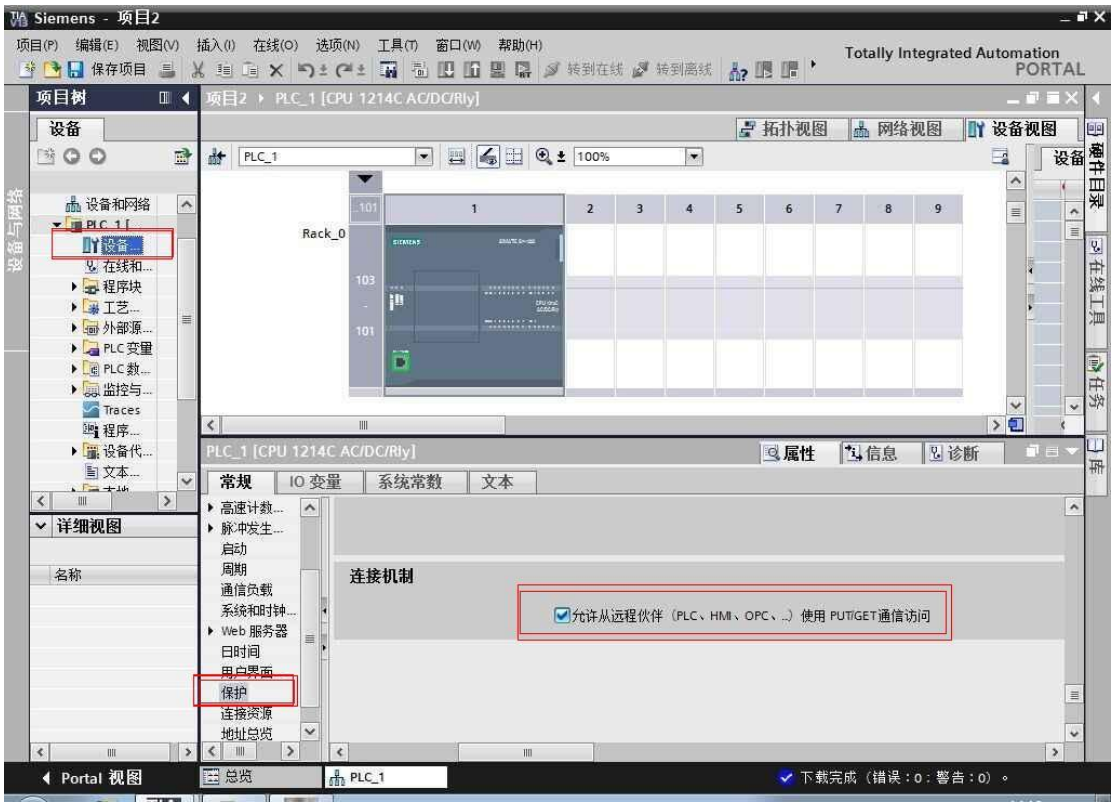
此产品可以实现西门子 CP243-1，大连德嘉的 CP243i，CP243-ibus，ETH-PPI，ETH-ibus，还有西门子 S7-200 SMART、S7-300、S7-1200、S7-1500 之间的通讯。

注：在 S7-1200/S7-1500 的编程软件 Portal 中，初始定义 DB 块时， 【仅符号访问】的选项不要打对号“v”





注：在博图 V13 中的设备组态--->属性--->连接机制---->允许从远程伙伴（PLC、HMI、OPC、...）使用 PUT/GET 通讯访问打上勾。如下图所示：



1. 首先在 IE 浏览器中输入后门地址 192.168.1.222 进入 A07 PLC 的设置界面。

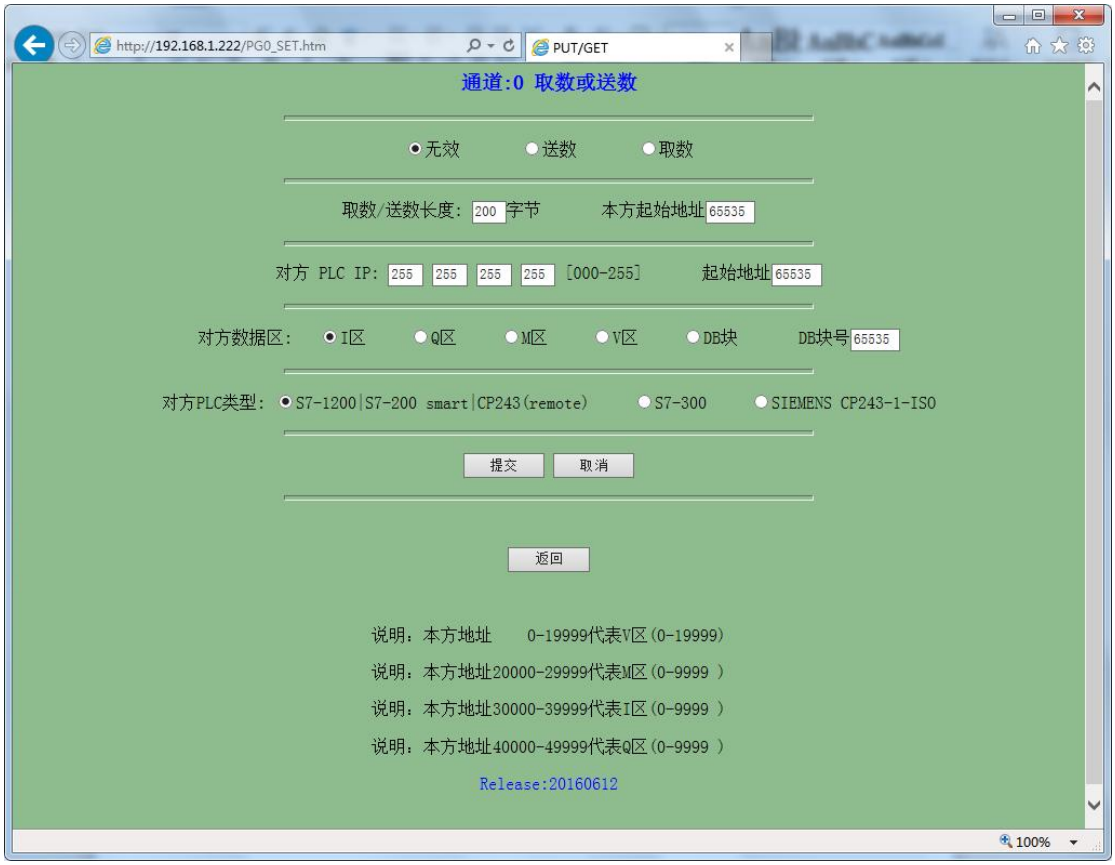
这里可以选择中英文，我们点击中文进入

2. 选择 PLC 通讯，进入下一界面

A07 型 PLC 提供了 6 个通道。

3. 数据通讯设置界面，这里可以选择取数/送数，不进行通讯时选择无效即可。我们只需要填入取/送数的长度，本方地址，对方 PLC 的 IP 以及起始地址，设置起来十分简单。

注意下方说明的地址对应关系。



注意：传送数据时 A07 型 PLC 需要保持运行状态。

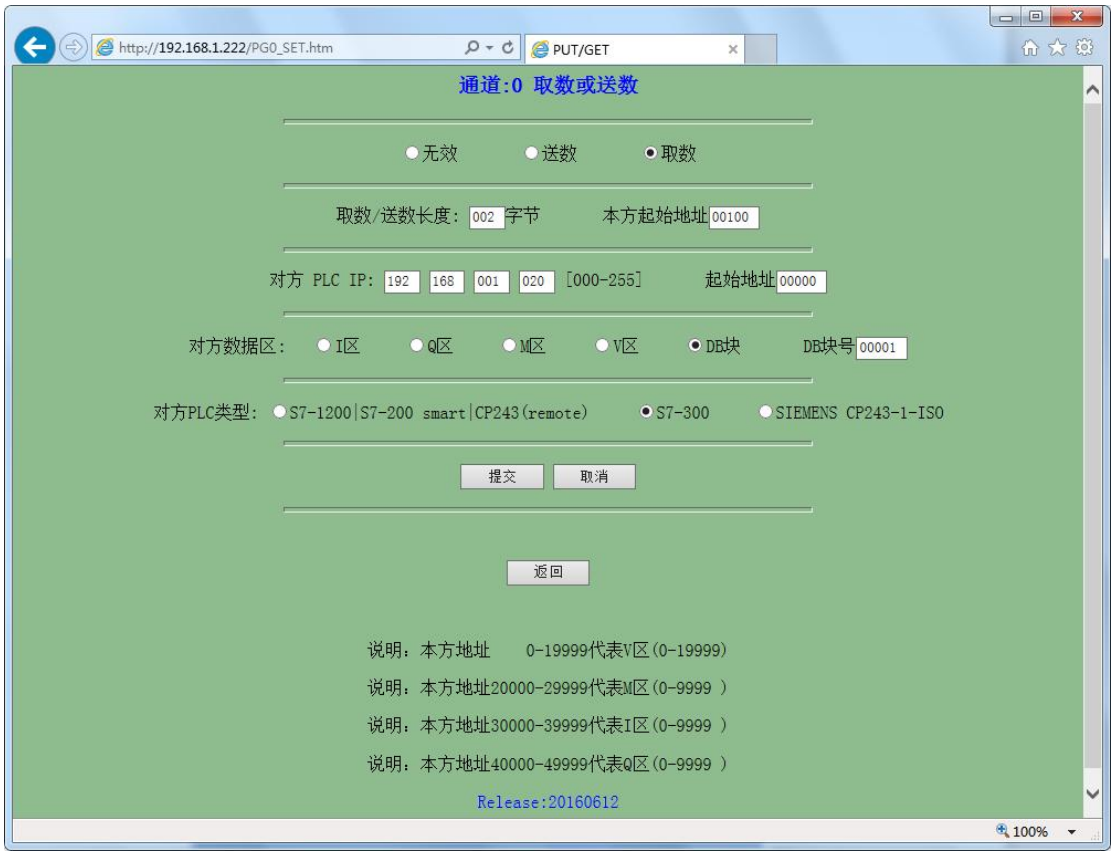
11 PLC 之间通讯实例

这是一个 3 个 PLC 之间的通讯，我们从 S7-300 中 DB1.DBW0 数据取出来，存在我们的 A07 的 VW100 中，并将数据送到 S7-1200 的 MW0 中，送到 S7-200 SMART 的 MW0 中。

S7-300 的 IP 地址设置为 192.168.1.20
S7-1200 的 IP 地址设置为 192.168.1.21
S7-200 SMART 的 IP 地址设置为 192.168.1.22

1. A07 通过网页设置 PLC 之间通讯参数

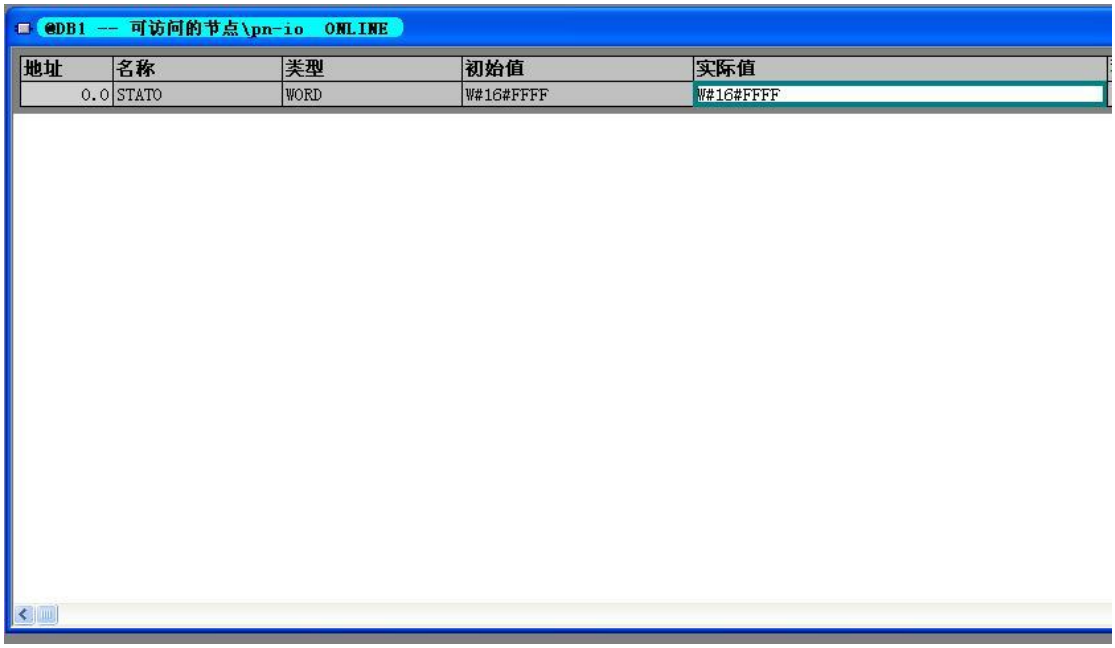
从 S7-300 中取数设置：



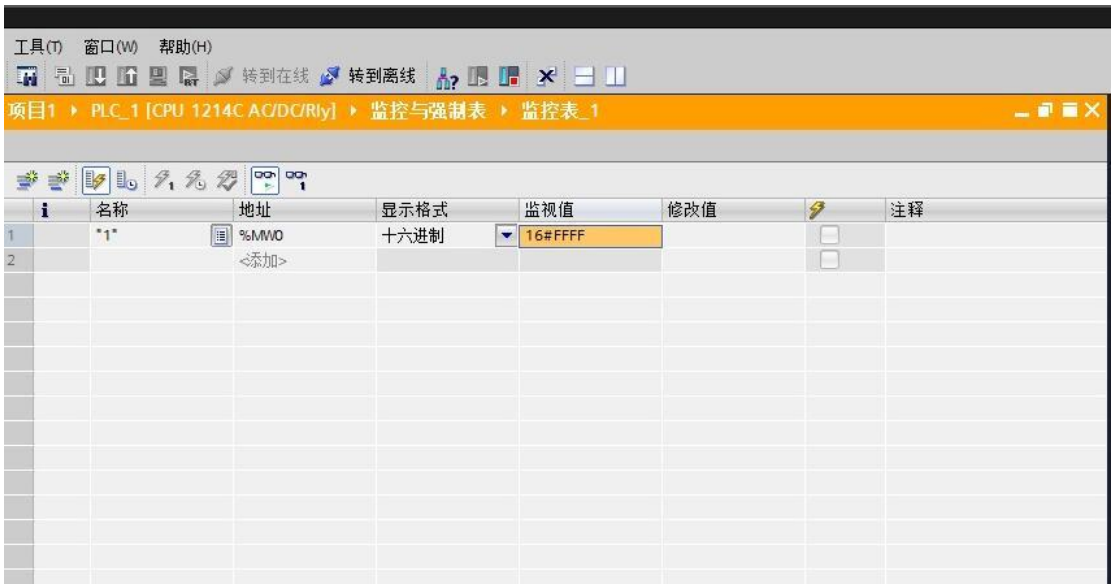
将数据送到 S7-1200 的 MW0

将数据送到 S7-200SMART 的 MW0 中，与上图 1200 设置（除更改 IP 地址）其它一样

2. 我们首先观察一下 S7-300 中的数据，我们将数据值定义为 FFFF



3. 再观察一下 S7-1200 的 MW0 的数据值



4. 最后看一下 S7-200 SMART 的 MW0 数据值

实现数据的传送就这么简单。

12 C# Modbus TCP 通讯实例

这里我只是简单的理解一下 Modbus TCP/IP 协议的内容，就是去掉了 modbus 协议本身的 CRC 校验，增加了 MBAP 报文头。

这里只是简单的理解，深入之后可能会有更多的东西需要学习，但为了可以快速入门，我们先按照这个思路往下走。

我们首先来看一下，MBAP 报文头都包括了哪些信息和内容

MBAP 报文头包括下列域：

域	长度	描述	客户机	服务器
事务元标识符	2 个字节	MODBUS 请求/响应事务处理的识别码	客户机启动	服务器从接收的请求中重新复制
协议标识符	2 个字节	0=MODBUS 协议	客户机启动	服务器从接收的请求中重新复制
长度	2 个字节	以下字节的数量	客户机启动（请求）	服务器（响应）启动
单元标识符	1 个字节	串行链路或其它总线上连接的远程从站的识别码	客户机启动	服务器从接收的请求中重新复制

下面我们再来介绍一下针对我们 PLC 的功能码

1、0x01 功能码： 按位读取 Q 区（线圈 ）

例：我们来读取从 Q0.0 到 Q0.5 这 6 个线圈

发送码分析：

请求 PDU

功能码	1 个字节	**0x01**
起始地址	2 个字节	0x0000 至 0xFFFF
线圈数量	2 个字节	1 至 2000（0x7D0）

根据上面的分析，我们需要发送 0x00, 0x01, 0x00, 0x00, 0x00, 0x06, 0x01, 0x01, 0x00, 0x00, 0x00, 0x06

接收码分析：

响应 PDU

功能码	1 个字节	0x01
字节数	1 个字节	N*
线圈状态	N 个字节	n=N 或 N+1

我们收到的数据为 0x00, 0x01, 0x00, 0x00, 0x00, 0x04, 0x01, 0x01, 0x01, 0x2A

modbus 数据中从左数，0x01 表示功能码，0x01 表示 1 个字节数据，0x2A 表示数据值

把 0x2A 转换为 2 进制为 0010 1010 ， 从左数起，前 2 位是补充数据 00，剩下的 101010 表示我们读取的 Q0.5 到 Q0.0 的状态。

Q0.5------ ON,

Q0.4 ------ OFF,

Q0.3-------ON,

Q0.2--------OFF,

Q0.1-------ON,

Q0.0----------OFF。

注意数据的顺序，左侧是高位，右侧是低位。

注意：上述发送及接收数据中，红色数码是 MBAP 报文头，黑色码是 modbus 数据， 下同

2、0x02 功能码：按位读取 I 区 （离散输入）

例：我们来读取从 I0.0 到 I0.5 这 6 个离散输入点

发送码分析：

请求 PDU

功能码	1 个字节	**0x02**
起始地址	2 个字节	0x0000 至 0xFFFF
输入数量	2 个字节	1 至 2000（0x7D0）

根据上面的分析，我们需要发送 0x00，0x01，0x00，0x00，0x00，0x06，0x01， 0x02，0x00，0x00，0x00， 0x06

接收码分析：

响应 PDU

功能码	1 个字节	0x82
字节数	1 个字节	N*
输入状态	N*×1 个字节	

*N＝输出数量/8，如果余数不等于 0，那么N＝N+1

我们收到的数据为 0x00， 0x01， 0x00， 0x00， 0x00， 0x04， 0x01，0x02，0x01，0x00

modbus 数据中从左数，0x02 表示功能码，0x01 表示 1 个字节数据，0x00 表示数据值

把 0x0 转换为 2 进制为 0000 0000 ， 从左数起，前 2 位是补充数据 00，剩下的 000000 表示我们读取的 I0.5 到 I0.0 的状态。

3、0x03 功能码：按双字节（VW）读取 V 区或者读 MW

Modbus 寄存器 0-------19999 是读取 VW

Modbus 寄存器 20000------20031 是读取 MW

例：我们来读取从 VW0 到 VW2 这个数据

发送码分析：

请求

功能码	1 个字节	**0x03**
起始地址	2 个字节	0x0000 至 0xFFFF
寄存器数量	2 个字节	1 至 125（0x7D）

根据上面的分析，我们需要发送 0x00, 0x01, 0x00, 0x00, 0x00, 0x06, 0x01, 0x03, 0x00, 0x00, 0x00, 0x03

接收码分析：

响应

功能码	1 个字节	**0x03**
字节数	1 个字节	2×**N***
寄存器值	**N***×2 个字节	

***N**=寄存器的数量

我们收到的数据为 0x00, 0x01, 0x00, 0x00, 0x00, 0x09, 0x01, 0x03，0x06，0x04，0x00，0x03，0x01，0x02，0x05

modbus 数据中从左数，0x03 表示功能码，0x06 表示 6 个字节数据，0x04，0x00，0x03，0x01，0x02，0x05 表示数据值

VW0 为 0x0400，VW2 为 0x0301，VW4 为 0x0205

4、0x05 功能码：按位写 Q 区

例：我们来把 Q0.0 置 1，请注意，置位数据为 0xFF00，清零数据为 0x0000

发送码分析：

请求

功能码	1 个字节	**0x05**
输出地址	2 个字节	0x0000 至 0xFFFF
输出值	2 个字节	0x0000 至 0x00

根据上面的分析，我们需要发送 0x00, 0x01, 0x00, 0x00, 0x00, 0x06, 0x01, 0x05, 0x00, 0x00, 0xFF, 0x00

接收码分析：

响应

功能码	1 个字节	**0x05**
输出地址	2 个字节	0x0000 至 0xFFFF
输出值	2 个字节	0x0000 至 0xFF00

我们收到的数据为 0x00，0x01，0x00，0x00，0x00，0x06，0x01，0x05，0x00，0x00，0xFF，0x00，

5、0x06 功能码： 按双字节（VW）写 V 区或者写 MW

Modbus 寄存器 0-------19999 是写 VW

Modbus 寄存器 20000------20031 是写 MW

例：我们将数据 0x2636 写入 VW0

发送码分析：

请求

功能码	1 个字节	**0x06**
寄存器地址	2 个字节	0x0000 至 0xFFFF
寄存器值	2 个字节	0x0000 至 0xFFFF

根据上面的分析，我们需要发送 0x00，0x01，0x00，0x00，0x00，0x06，0x01， 0x06，0x00，0x00，0x26， 0x36

接收码分析：

响应

功能码	1 个字节	**0x06**
寄存器地址	2 个字节	0x0000 至 0xFFFF
寄存器值	2 个字节	0x0000 至 0xFFFF

我们收到的数据为 0x00，0x01，0x00，0x00，0x00，0x06，0x01，0x06，0x00，0x00，0x26，0x36

6、0x0F 功能码：按多个位写 Q 区

例：我们将 Q0.0 到 Q0.5 共 6 个线圈全部置位 1

发送码分析：

请求 PDU

功能码	1 个字节	**0x0F**
起始地址	2 个字节	0x0000 至 0xFFFF
输出数量	2 个字节	0x0001 至 0x07B0
字节数	1 个字节	**N***
输出值	**N***×1 个字节	

***N**=输出数量/8，如果余数不等于 0，那么N＝N+1

我们要将 Q0.0 到 Q0.5 输出 1，要发送的值应该为二进制 0011 1111，转换为 16 进制为 0x3F

根据上面的分析，我们需要发送 0x00，0x01，0x00，0x00，0x00，0x08，0x01， 0x0F，0x00，0x00，0x00， 0x06，0x01，0x3F

接收码分析：

响应 PDU

功能码	1 个字节	**0x0F**
起始地址	2 个字节	0x0000 至 0xFFFF
输出数量	2 个字节	0x0001 至 0x07B0

我们收到的数据为 0x00，0x01，0x00，0x00，0x00，0x06，0x01，0x0F，0x00，0x00，0x00，0x06

7、0x10 功能码： 写 2N 个 VW 或者 MW

Modbus 寄存器 0-------19999 是写 VW

Modbus 寄存器 20000------20031 是写 MW

例：我们将数据 0x01，0x05，0x0A，0x09 写入 VW0 和 VW2

发送码分析：

请求 PDU

功能码	1 个字节	**0x10**
起始地址	2 个字节	0x0000 至 0xFFFF
寄存器数量	2 个字节	0x0001 至 0x0078
字节数	1 个字节	2×N*
寄存器值	N*×2 个字节	值

*N=寄存器数量

根据上面的分析，我们需要发送 0x00, 0x01, 0x00, 0x00, 0x00, 0x0B, 0x01, 0x10, 0x00, 0x00, 0x00, 0x02, 0x04, 0x01, 0x05, 0x0A, 0x09

接收码分析：

响应 PDU

功能码	1 个字节	**0x10**
起始地址	2 个字节	0x0000 至 0xFFFF
寄存器数量	2 个字节	1 至 123（0x7B）

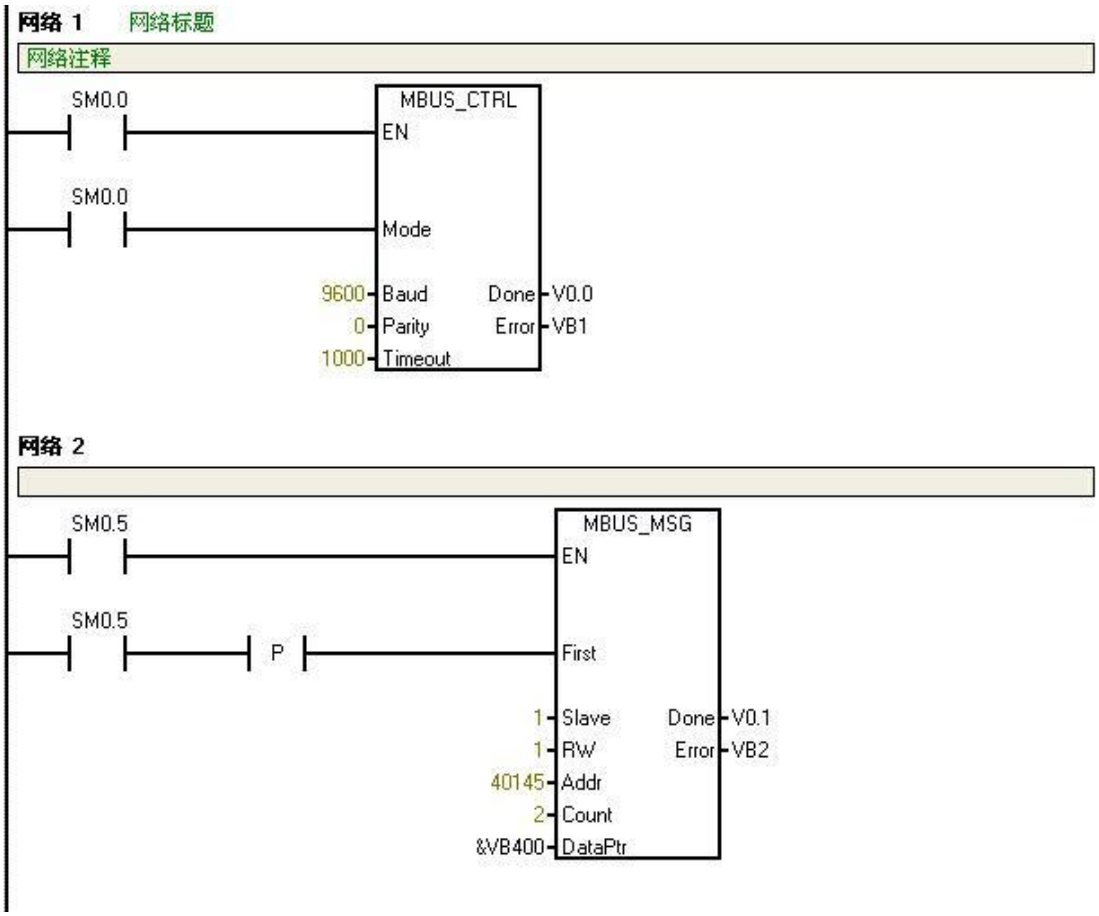
我们收到的数据为 0x00, 0x01, 0x00, 0x00, 0x00, 0x06, 0x01, 0x10，0x00，0x00，0x00，0x02

好的，至此，我们关于 Modbus TCP 命令连接我们 PLC 的分析就结束了，后面我上传了我做好的 C#程序供大家参考，

这里要注意一个问题，此程序中缺少断线重连机制，请大家自己添加一下吧

13 与数码管 Modbus 通讯实例

1. 首先，将 PLC 与数码管显示器接好，然后在 PLC 中建立 modbus 主站，如图：



其中的参数要根据数码管的说明来填写，数码管手册如下图：

功能	指令
	10H 功能码
显示 10 进制数（带正负号和小数点）	PLC 发送 ：01 10 00 90 00 02 04 00 02 01 EA DB 1C ● 01：数码管屏的站号（RS485 地址） ● 10：功能码，表示写多个寄存器 ● 00 90：数码管屏的显示寄存器(带小数点和正负号的整数) ● 00 02：寄存器个数 ● 04：数据个数（字节数） ● 00 02： 00 表示正负号（00=正数；01=负数，数字前显示-） 02 表示小数点位数，0 表示无小数点。2 表示小数点后有 2 位数字 ● 01 EA：2 位整数，高字节在前。01 EA 表示十进制 490 ● DB 1C：二个字节 CRC 码 此命令将显示“4.90” 数码管屏返回 ：01 10 00 90 00 02 41 E5 例子： （1） 01 10 00 90 00 02 04 01 01 00 0A 2A F8 将显示“-1.0” （2） 01 10 00 90 00 02 04 00 01 00 02 2A C2 将显示“0.2”

可以看到这个例子为 PLC 发送：01 10 00 90 00 02 04 00 02 01 EA DB 1C

其中 01 为数码管地址，即 Slave 填入 1 ，因为需要向数码管写入，所以 RW 填 1

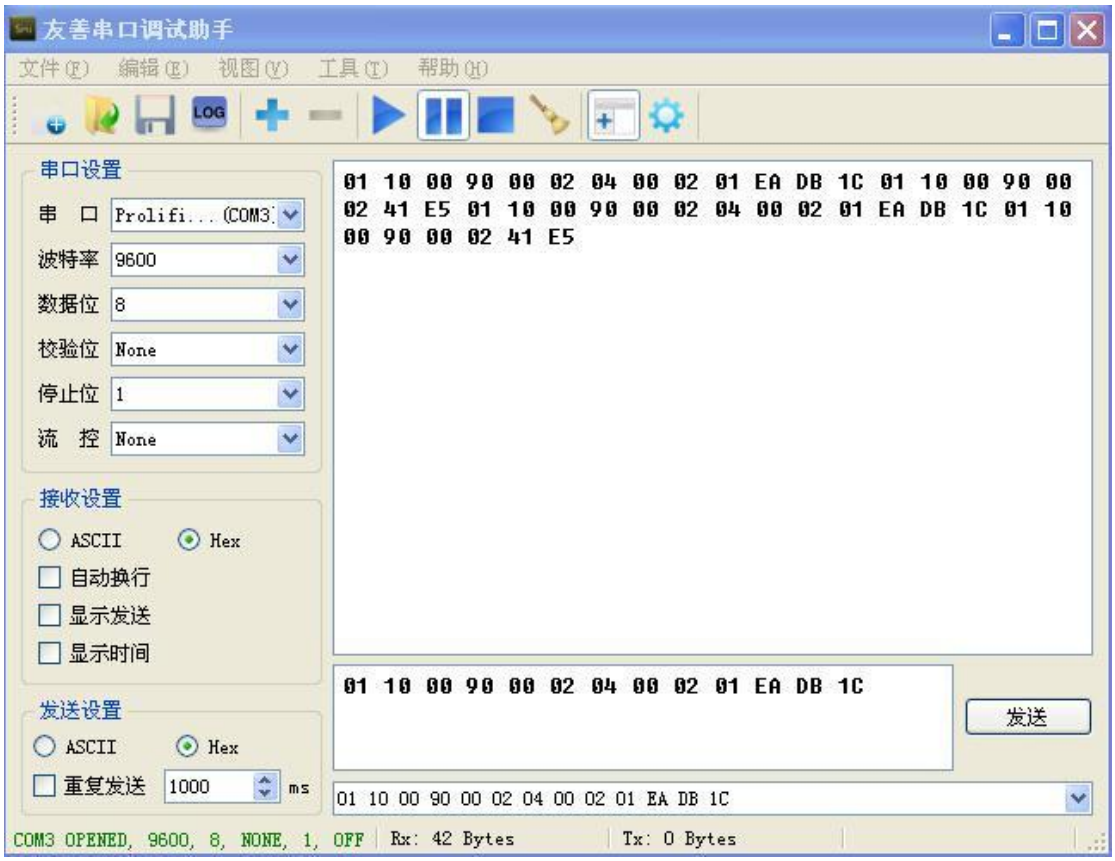
00 09 为数码管屏的显示寄存器，转换成 10 进制为 144，因为起始地址为 40001，所以我们这里要填入 40001+144 = 40145

04 为数据字节数，即 2 个字，所以 Count 填入 2
最后数据指针自己选择即可，注意不要与分配的库存储区冲突，这里我们用的是 VB400

	地址	格式	当前值
1	VB400	十六进制	16#00
2	VB401	十六进制	16#02
3	VB402	十六进制	16#01
4	VB403	十六进制	16#EA

如图，我们要向 VB400～VB403 写入 00 02 01 EA 。

这里接一个 USB 转 485 串口监视器，然后我们用串口调试助手可以对通信进行监视，如下图：



我们可以看到 01～1C 为 PLC 发送的码，01～E5 为返回的码，和手册完全一样，同时你的数码管也会显示 4.90 如下图：