

# CPU S06 可编程逻辑控制器

## 用户手册

版本：V2.01

发布日期：10/2023

大连德嘉工控设备有限公司

# 目录

1	产品概述.....	3
2	参数设置.....	4
3	STEP 7-MicroWIN SMART 连接设置.....	8
4	WinCC 连接设置.....	10
5	组态王连接设置.....	19
6	力控连接设置.....	23
7	连接 SMART LINE 参数设置.....	25
8	ModbusRTU 通讯（填表方式）.....	26
9	ModbusTCP 通讯（填表方式）.....	30
10	PLC 之间通讯设置（填表方式）.....	33
11	PLC 之间通讯实例.....	36
12	C# Modbus TCP 通讯实例.....	40

## 1 产品概述

S06 型 PLC 与西门子 S7-200SMART 完全兼容，带有一个以太网口，本体不带 IO 点，具有 1 路 RS485 通讯功能，使用西门子 STEP 7-MicroWIN SMART 编程，内嵌 ModbusRTU、ModbusTCP、S7 PUT/GET（非编程），可与 Modbus 变频器、仪器仪表，PLC 等通讯，通过软件填表方式，实现上述几种方式的通讯。

该款 PLC 可支持用户定制，定制类型支持本体自带 IO 点，最多支持 16DI/16DO，8AI/8AO。

S06 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。
- S06 型 PLC 支持一路 485 通讯的功能，主要用于模拟量的采集与控制，485 采用填表方式，具有通讯断线诊断功能，简单快速方便。
- 可以外接 6 个 SMART IO 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兼容型 IO 模块。
- 可以使用西门子 STEP 7-MicroWIN SMART 编程软件，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 通讯（S7-200 SMART/S7-300/1200/1500，使用 S7 PUT/GET 命令）
- 可用于 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）。

暂不支持：

- (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-VB3966 具有断电保持功能，而大于 VB3966 部分则没有断电保持功能（此存储区总数为 3967，还可用 VB1000-VB4966 或 VB1000-VB2000+VB3000-VB5966 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

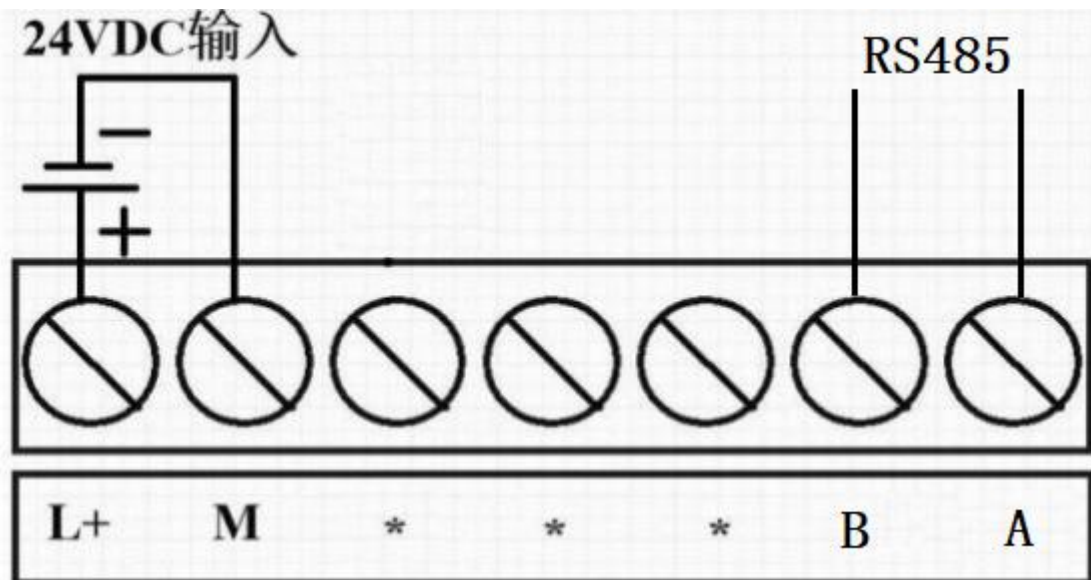
防护等级：IP20

网口通讯速率：100Mbps



## 2 参数设置

端子接线图





### LED 指示灯说明

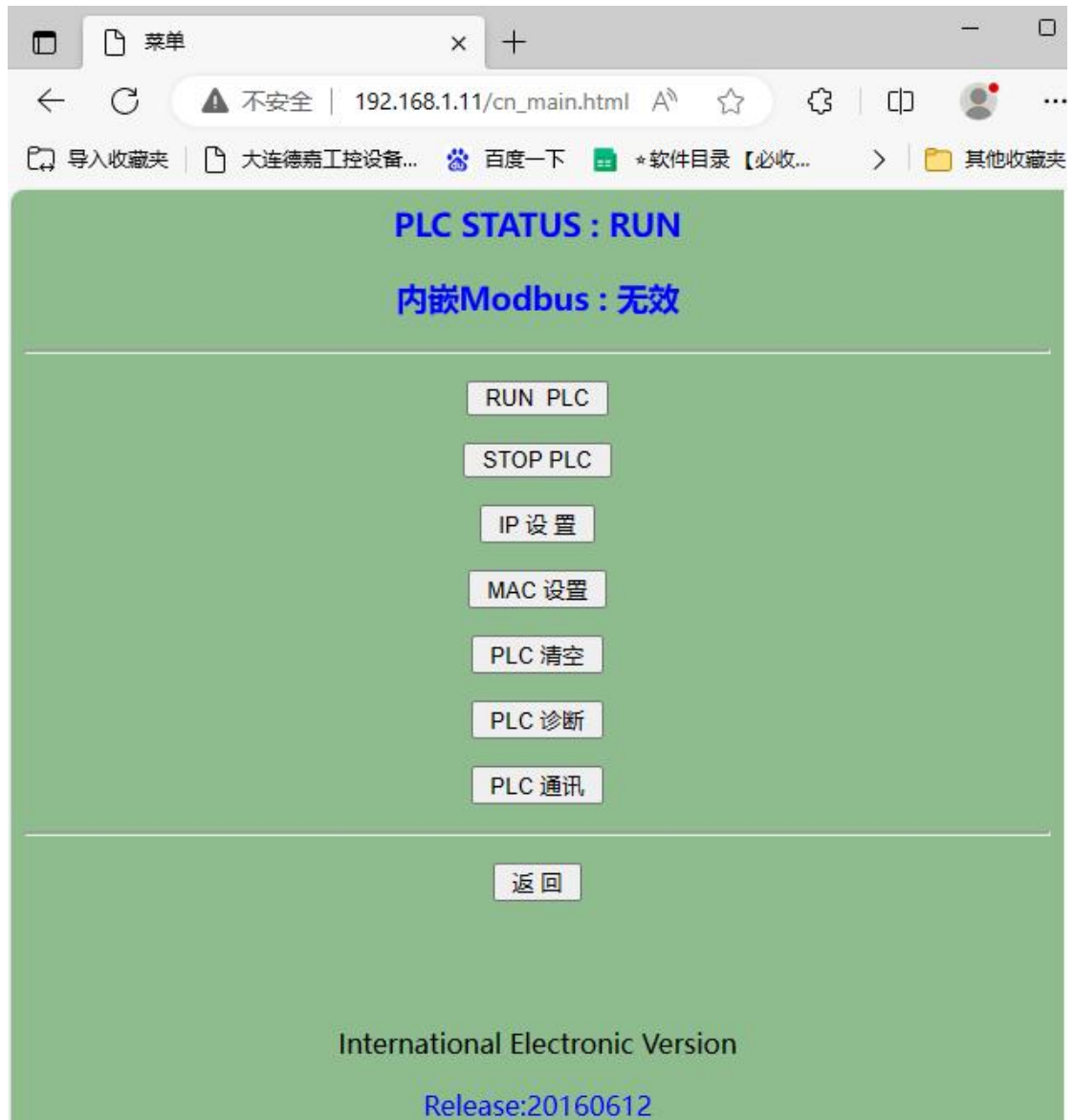
LED 指示灯			含义
DIAG (诊断)	RUN (运行)	STOP (停止)	
灭	灭	灭	PLC 电源电压缺失或不足
红(常亮)	绿(常亮)	绿(常亮)	PLC 未接 IO 模块
绿(闪烁)	绿(闪烁)	-	编程软件里闪烁指示灯(后接 IO 模块状态时)
-	绿(闪烁)	-	PLC 处于运行状态(后接 IO 模块状态时)
-	-	绿(闪烁)	PLC 处于停止状态(后接 IO 模块状态时)
绿(闪烁)	-	-	PLC 处于强制状态(后接 IO 模块状态时)

## 参数配置及查看

S06 PLC 可以登录网页进行查看型号以及参数配置，具体操作方法：电脑 IP 地址设置成 192.168.1.xxx（如：192.168.1.100），浏览器地址栏里输入 192.168.1.222（回车），即可配置以及查看参数。

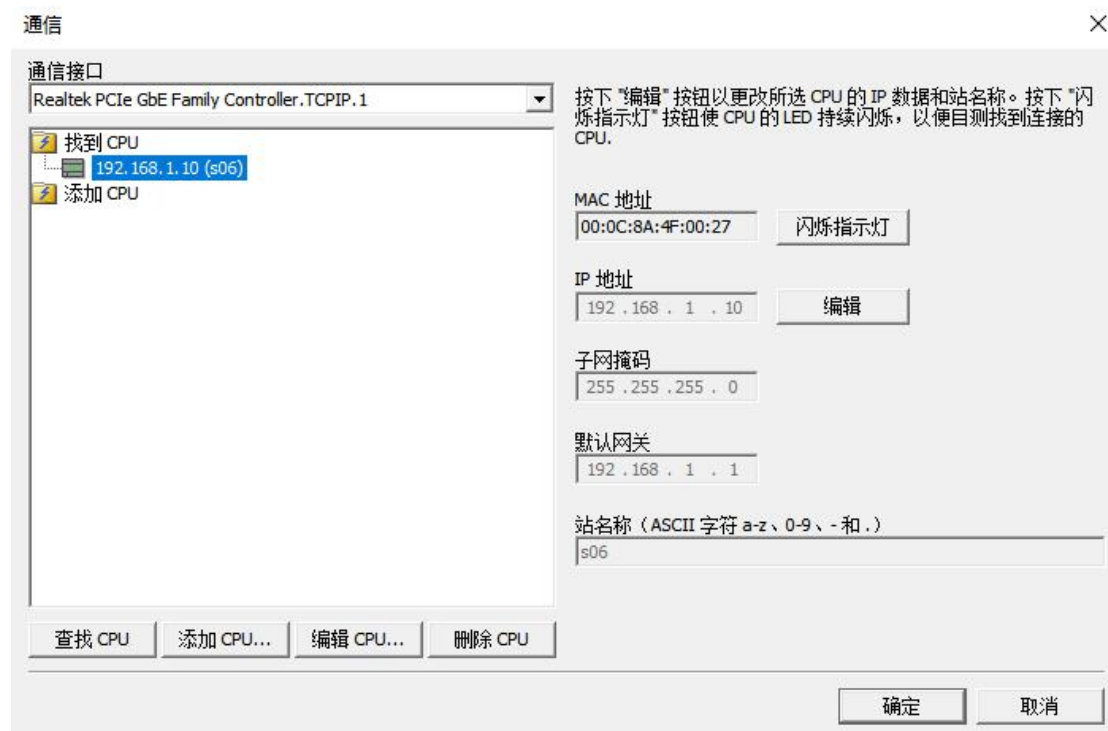


点击“Chinese”进入子菜单，可设置 PLC 的 IP 地址，运行/停止，清空参数等



### 3 STEP 7-MicroWIN SMART 连接设置

打开 STEP 7-MicroWIN SMART，点击“通信”，查找 CPU，就能找到 S06 CPU，其默认 IP 地址为 192.168.1.10，可通过后门网页【IP 设置】或者直接点击当前界面“编辑”修改 IP 地址。



系统块中配置的 CPU 类型为 CPU SR20 (AC/DC/Relay)，版本号：V02.04.00





注意：以下功能无法使用

系统块	模块	版本	输入	输出	订货号
CPU	CPU SR20 (AC/DC/Relay)	V02.04.00_00.00.00.00	I0.0	Q0.0	6ES7 288-1SR20-0AA0
SB					
EM 0					
EM 1					
EM 2					
EM 3					
EM 4					
EM 5					

- 通信
- 数字量输入
  - I0.0 - I0.7
  - I1.0 - I1.7
- 数字量输出
- 保持范围
- 安全
- 启动

### CPU 模式

选择 CPU 启动后的模式

STOP

### 硬件

- 允许缺少硬件
- 允许硬件配置错误

该功能无法使用，默认CPU上电状态为RUN模式

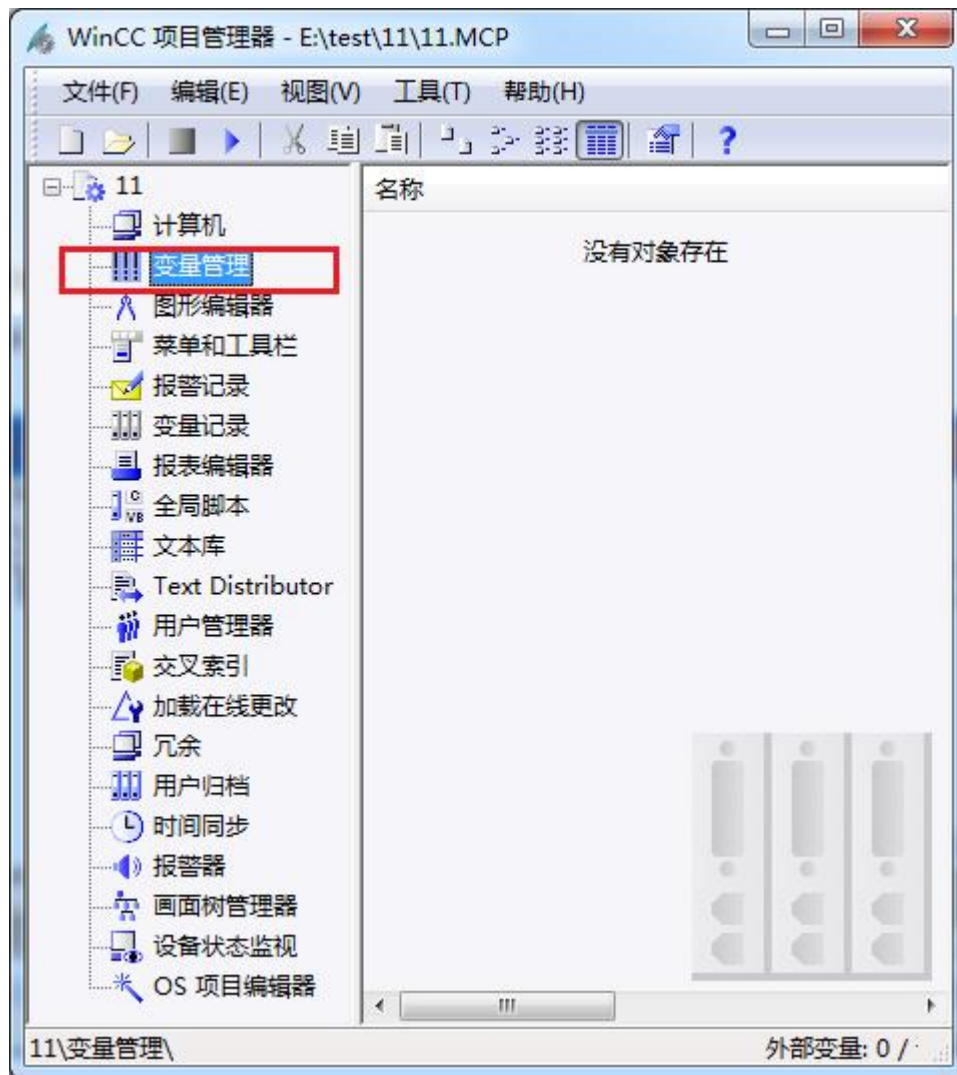
The screenshot shows the SIMATIC Manager software interface. The 'Tools' menu is highlighted with a red box, containing options like '运动' (Motion), 'PID', 'PWM', '文本显示' (Text Display), 'Get/Put', '数据日志' (Data Log), 'PROFINET', '运动控制面板' (Motion Control Panel), 'PID 控制面板' (PID Control Panel), 'SMART 驱动器组态' (SMART Drive Configuration), '查找' (Find), and '选项' (Options). Below the menu, a ladder logic diagram is visible, showing a normally open contact labeled '程序段注释' (Program Segment Comment) leading to a coil labeled '程序段注释'.

The screenshot shows the library tree in SIMATIC Manager. A red box highlights the '库' (Library) folder, which contains the following communication protocols:

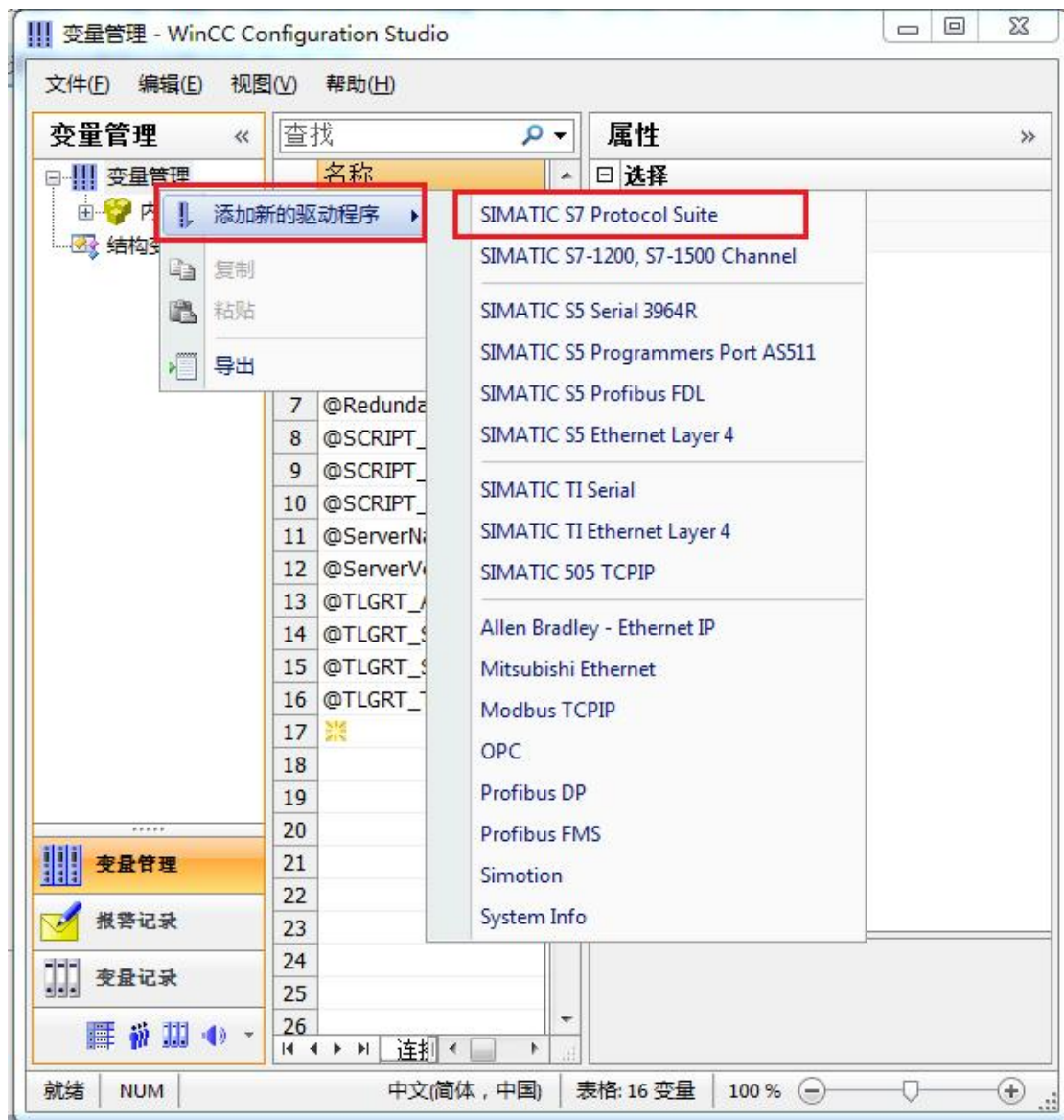
- Modbus RTU Master (v2.0)
- Modbus RTU Master2 (v2.0)
- Modbus RTU Slave (v3.1)
- Modbus TCP Client (v1.4)
- Modbus TCP Server (v1.0)
- Open User Communication (v1.0)
- PN Read/Write Record (v1.0)
- SINAMICS Control (v1.1)
- SINAMICS Parameter (v1.0)
- USS Protocol (v2.1)

## 4 WinCC 连接设置

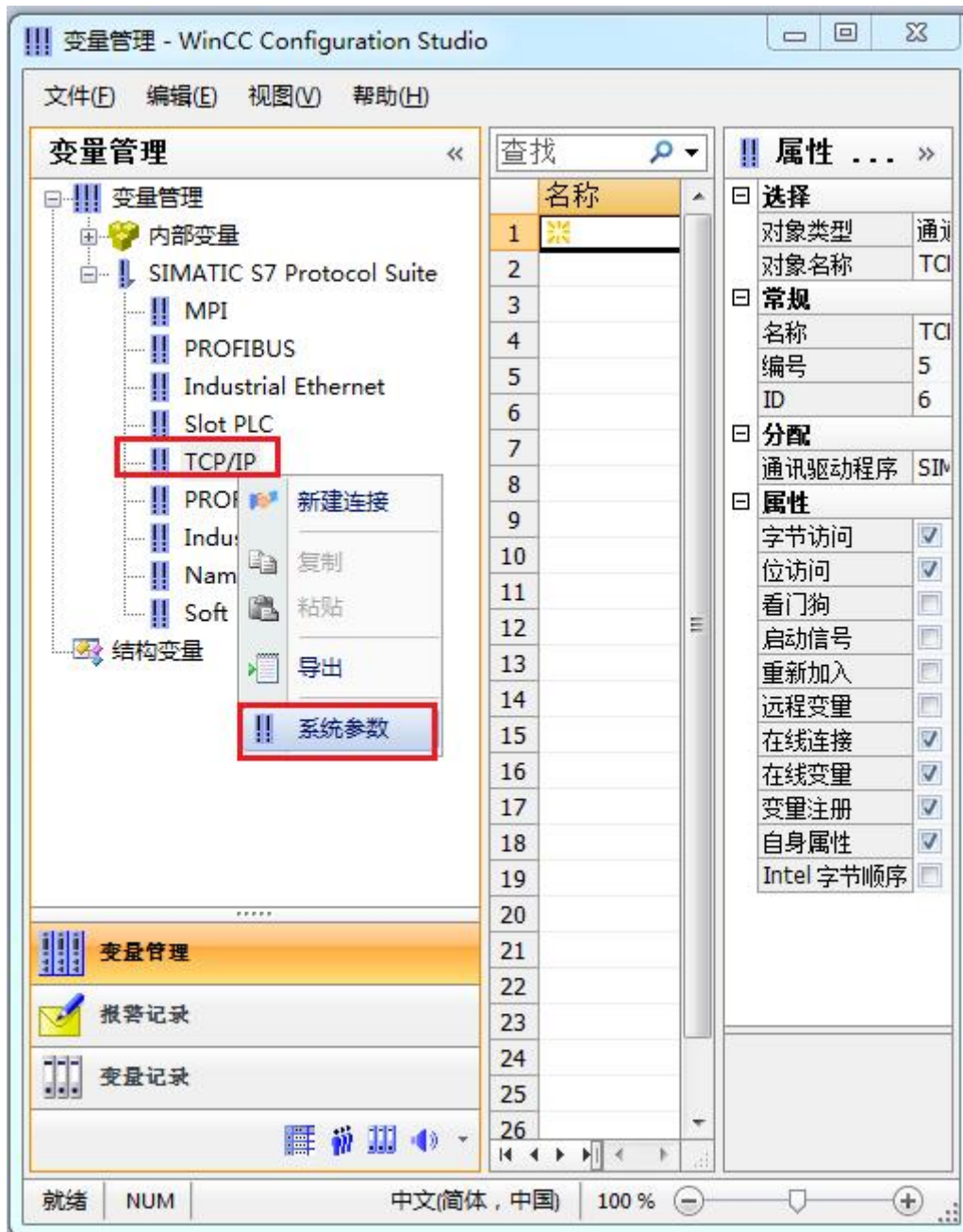
1. 打开 WinCC（以 WinCC7.3 为例），双击变量管理，打开变量管理器，添加驱动：



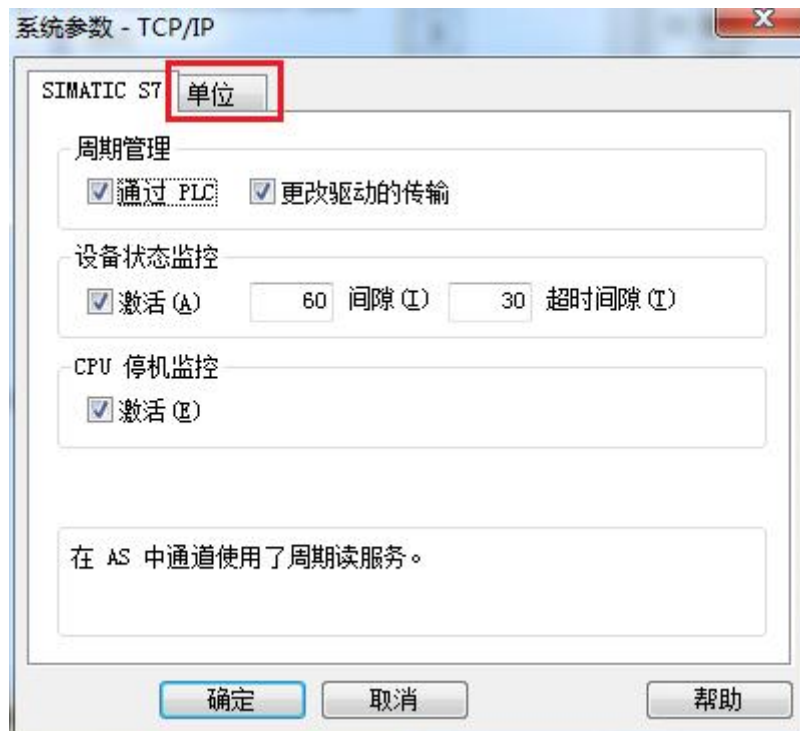
2. 右键单击变量管理，在弹出的菜单中选择添加驱动，SIMATIC S7 Protocol Suite，如下图所示



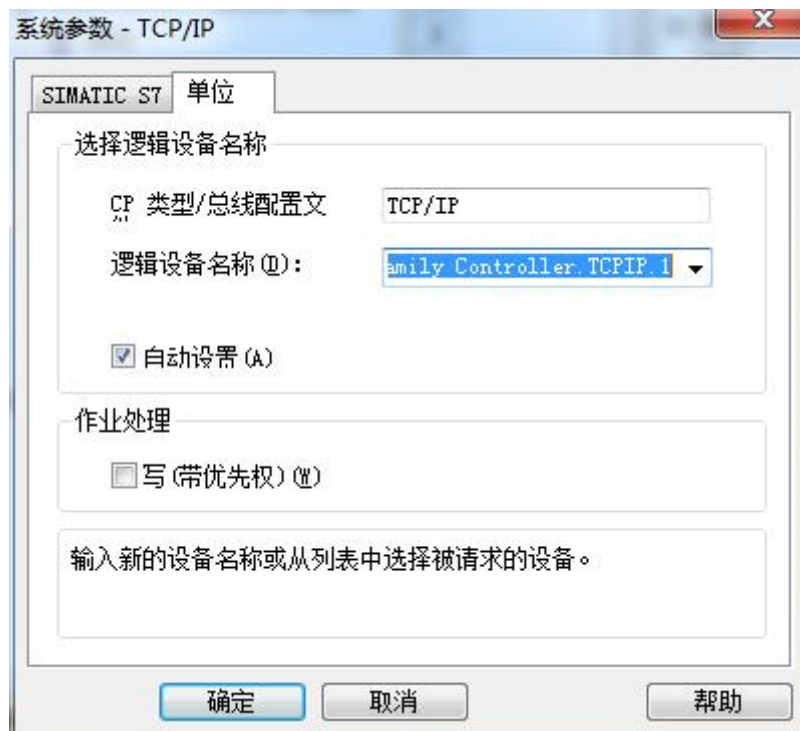
3. 添加好驱动之后，右键单击 SIMATIC S7 Protocol Suite 下的 TCP/IP，在弹出的菜单中选择系统参数



- 在弹出的对话框中点击单位选项卡



- 在逻辑设备名称选框中选择驱动为：网卡名.TCPIP.1



6. 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



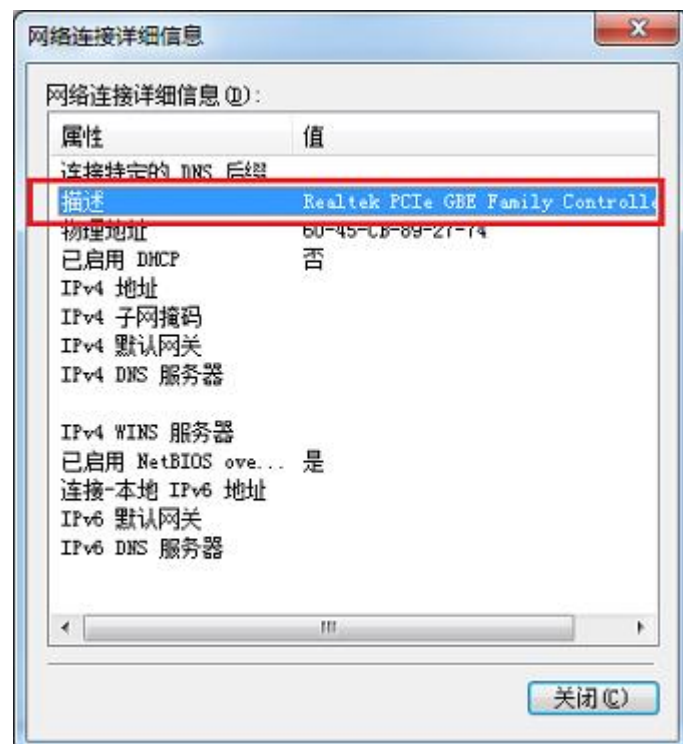
在网络共享中心中点击本地连接



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

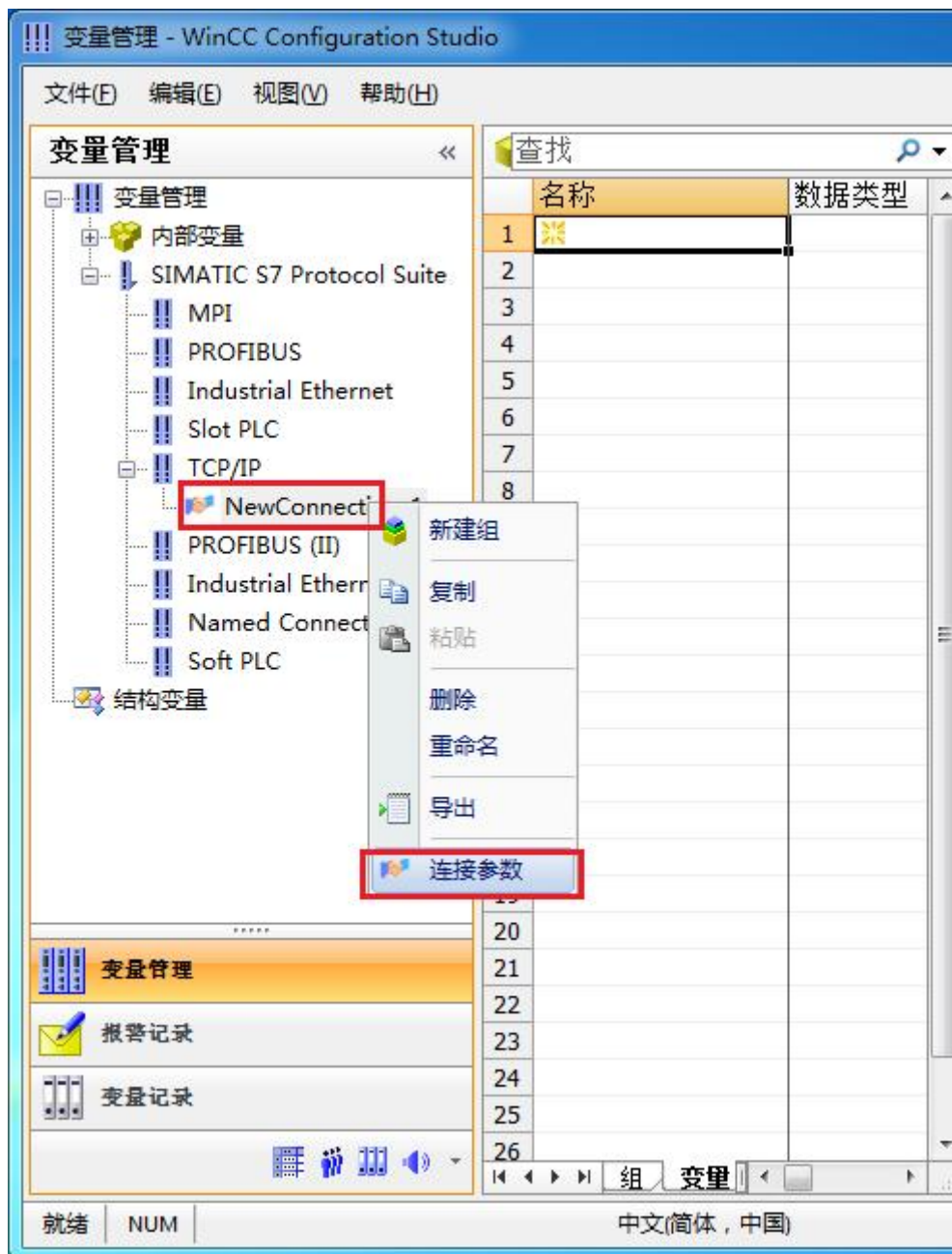


7. 再回到变量管理器中，右键点击 TCP/IP，选择新建连接，在 TCP/IP 选项下会生成一个名为 NewConnection\_1 的新连接选项。





8. 右键单击 NewConnection\_1，在弹出的菜单中选择



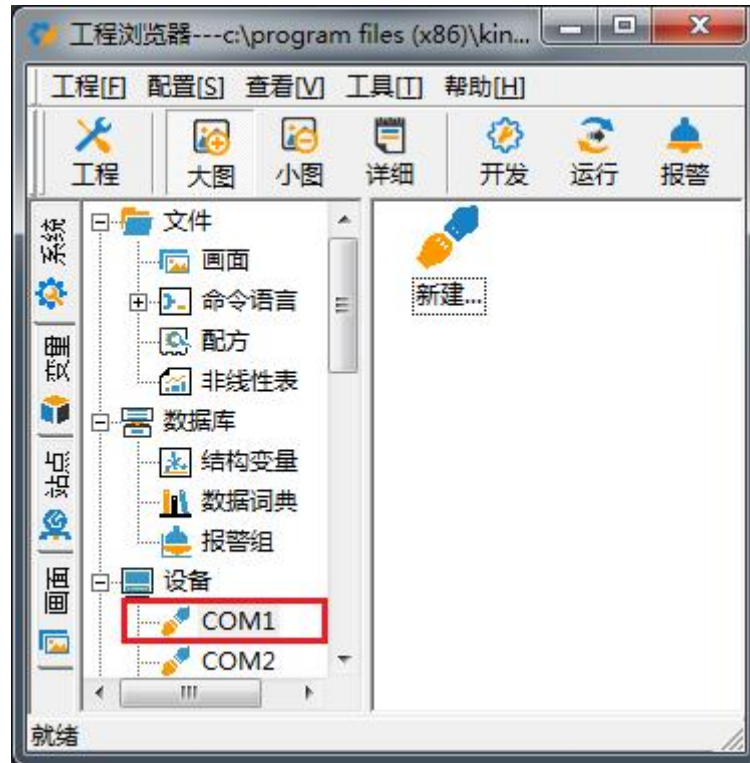
9. 在弹出的对话框中填写 M02 的 IP 地址，192.168.1.10



现在连接已经建立成功，已经可以建立变量和画面了。

## 5 组态王连接设置

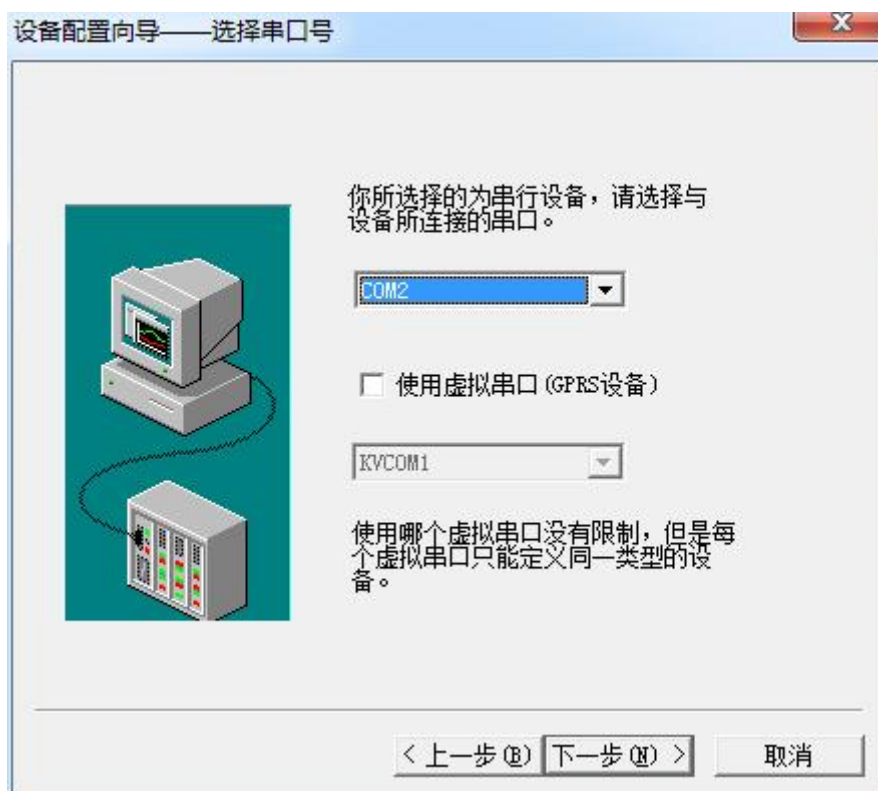
1. 打开组态王开发软件，选择设备→COM1



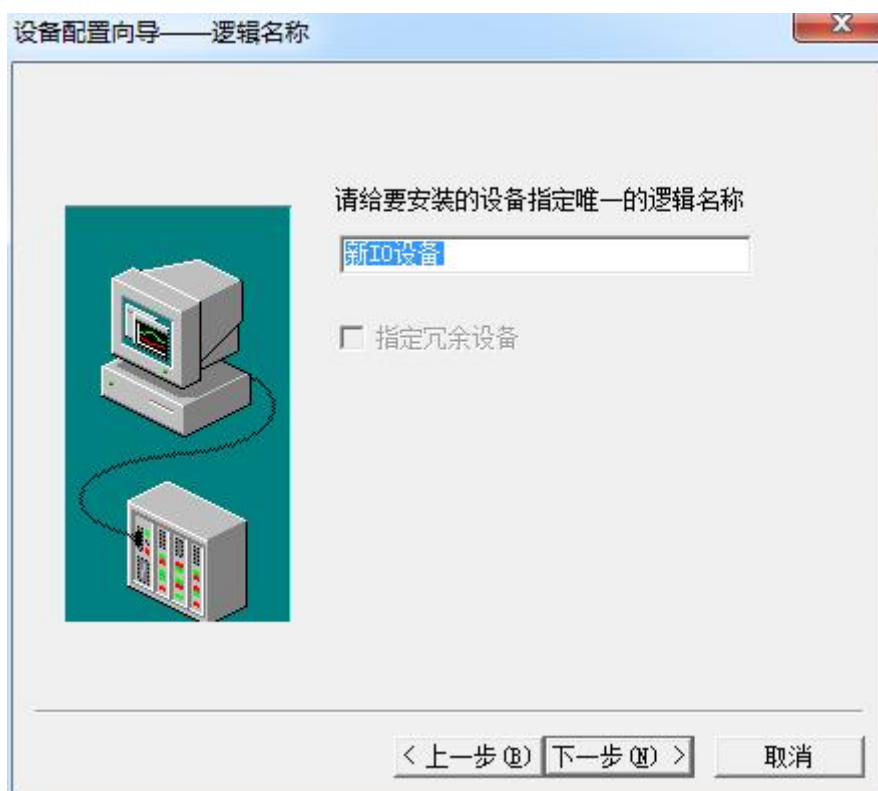
2. 双击“新建”，选择 S7-200 系列（TCP）→TCP



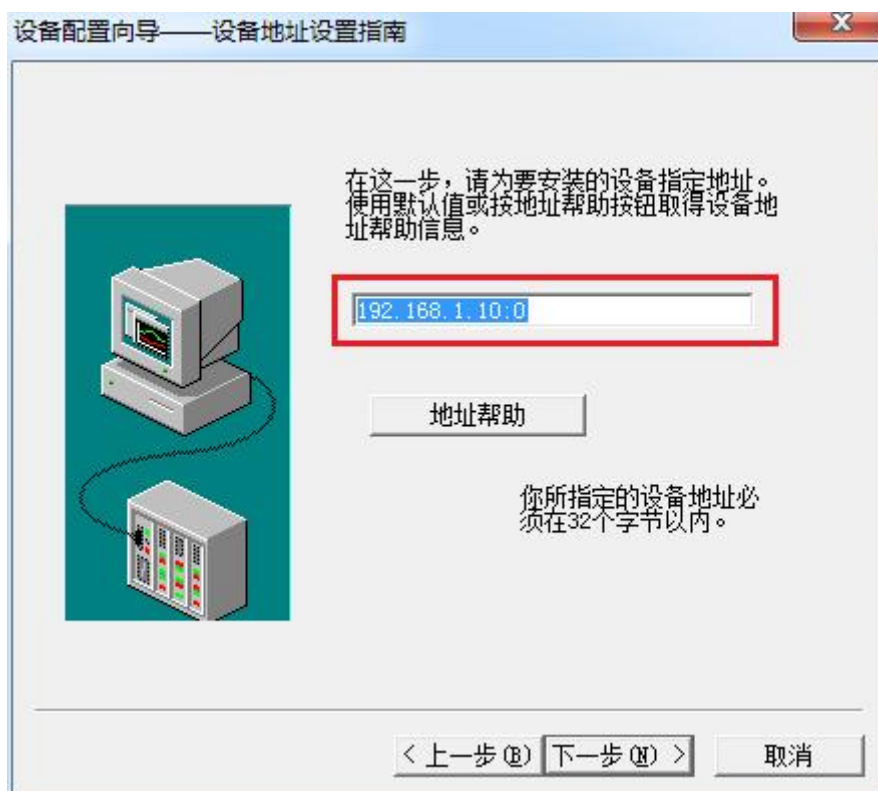
3. 选择 com 口号，此处选择默认值 com2



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”



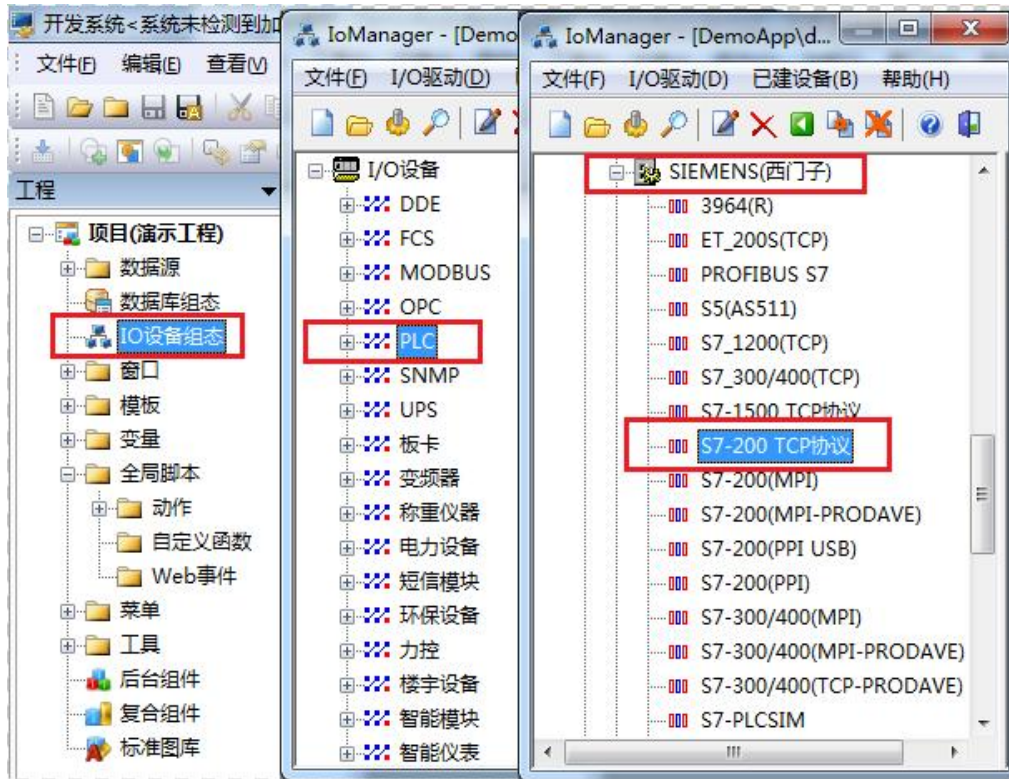
7. 单击“完成”，就配置了一个“TCP”设备。



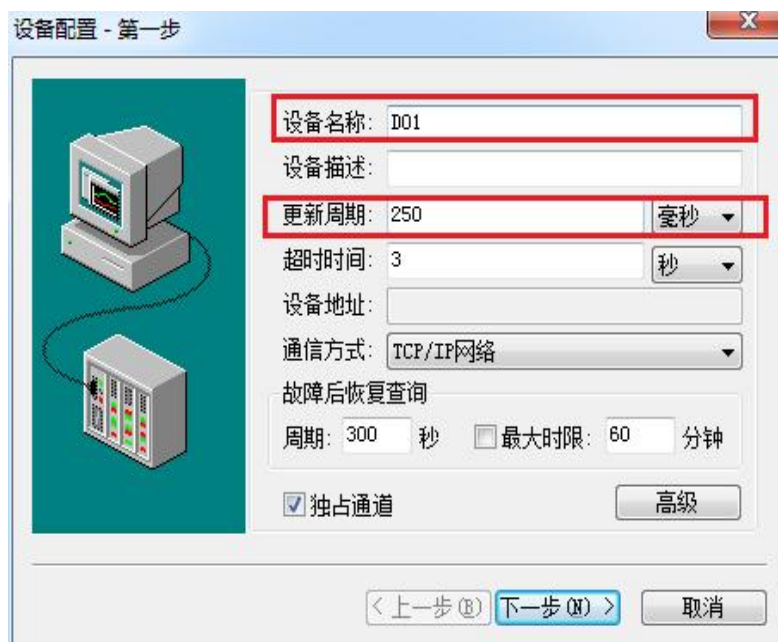
至此，就完成了 PLC 与组态王的连接。

## 6 力控连接设置

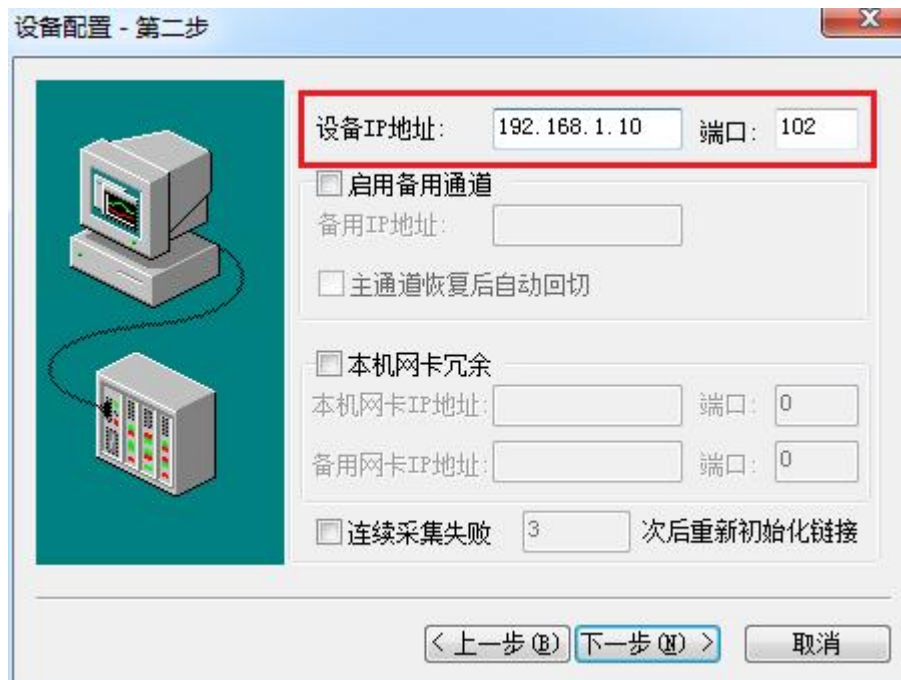
1. 打开组态软件，进入开发系统，打开“IO 设备组态”->“PLC”->“SIEMENS”->“S7-200 TCP 协议”，画面如下：



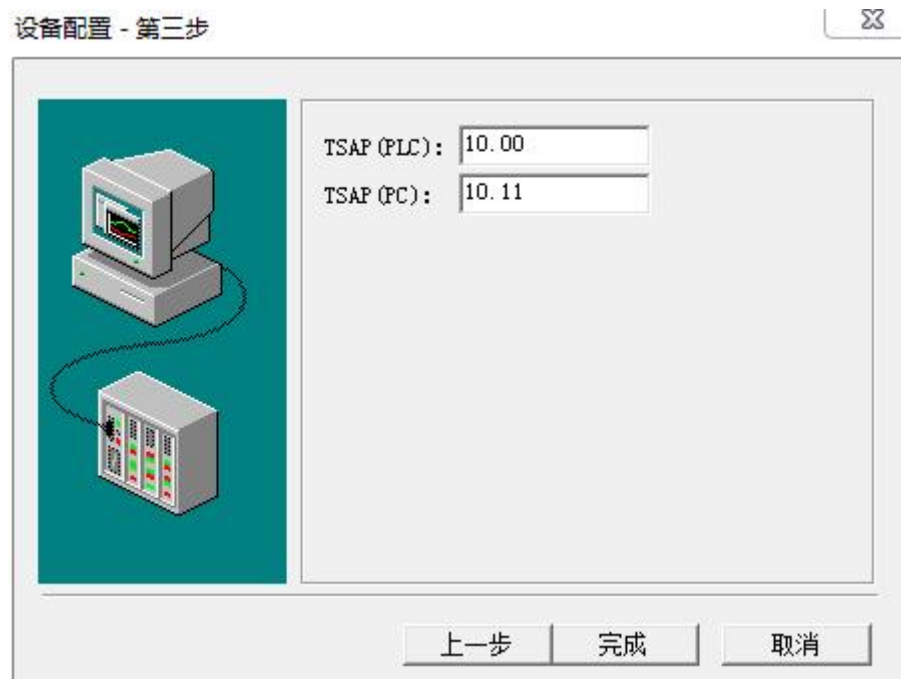
2. 第一步：基本参数配置，定义设备名称，修改更新周期。（更新周期一定要修改为 250 毫秒以上！）



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。

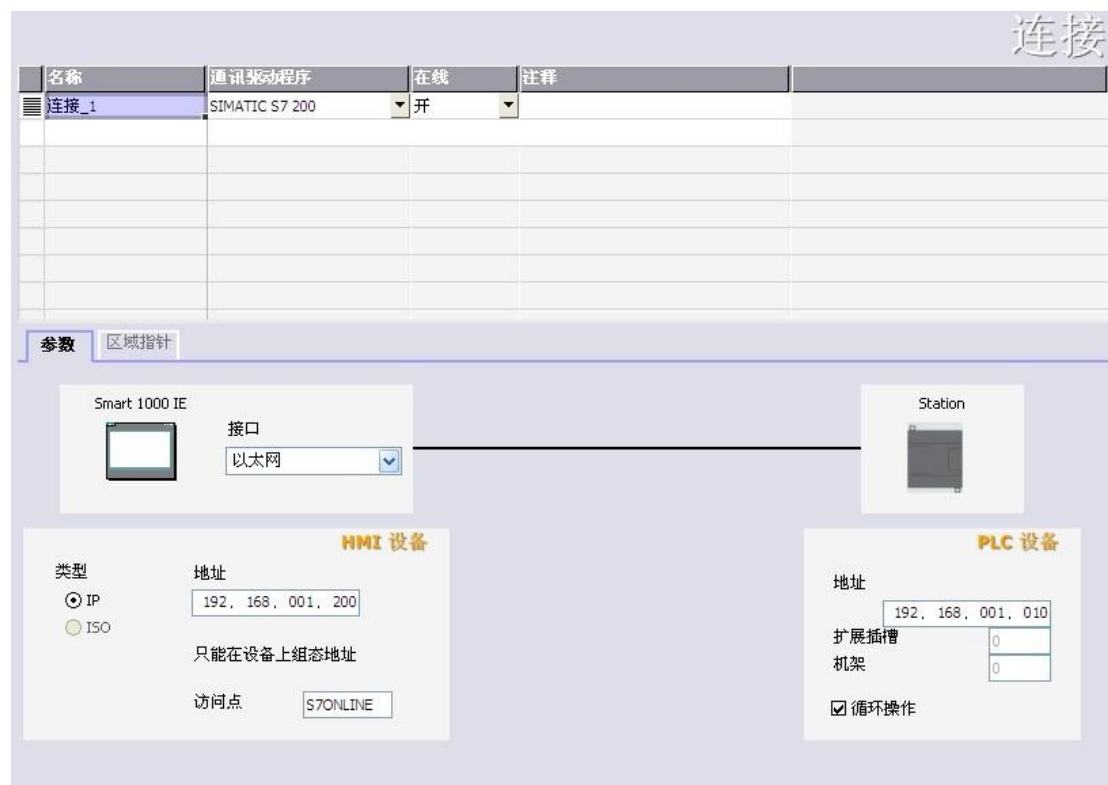




## 7 连接 SMART LINE 参数设置

1、在触摸屏上设置好触摸屏的 IP 地址，如 192.168.1.200

2、在 WinCC flexible SMART，给触摸屏编程，PLC 设备 IP 地址为 S06 IP 地址即可，如下图所示

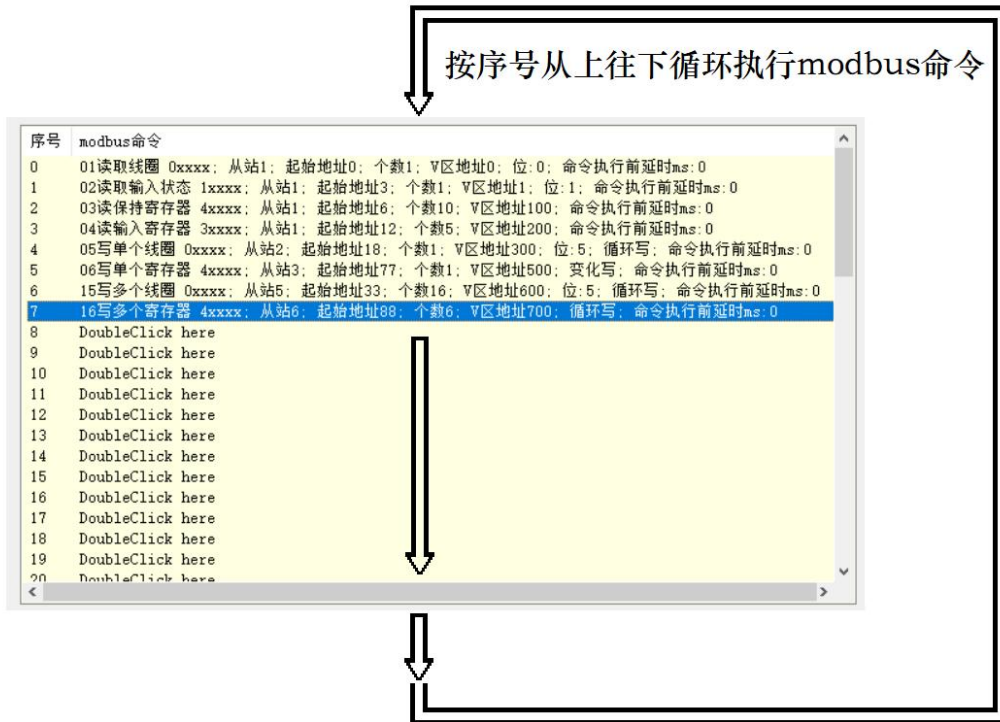


## 8 ModbusRTU 通讯（填表方式）

下载 PLC 通讯组态插件：[点击下载](#)

[http://www.dl-winbest.com/download/PLC\\_Config.rar](http://www.dl-winbest.com/download/PLC_Config.rar)

下面为 Modbus 命令从上往下循环执行的方式示意图：



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：

0001-00128                    Q0.0、 Q0.1 、 Q0.2 ..... Q15.7

1001-10128                    I0.0 、 I0.1 、 I0.2 ..... I15.7

3001-30032                    AIW0、 AIW2、 AIW4..... AIW62

400n-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：

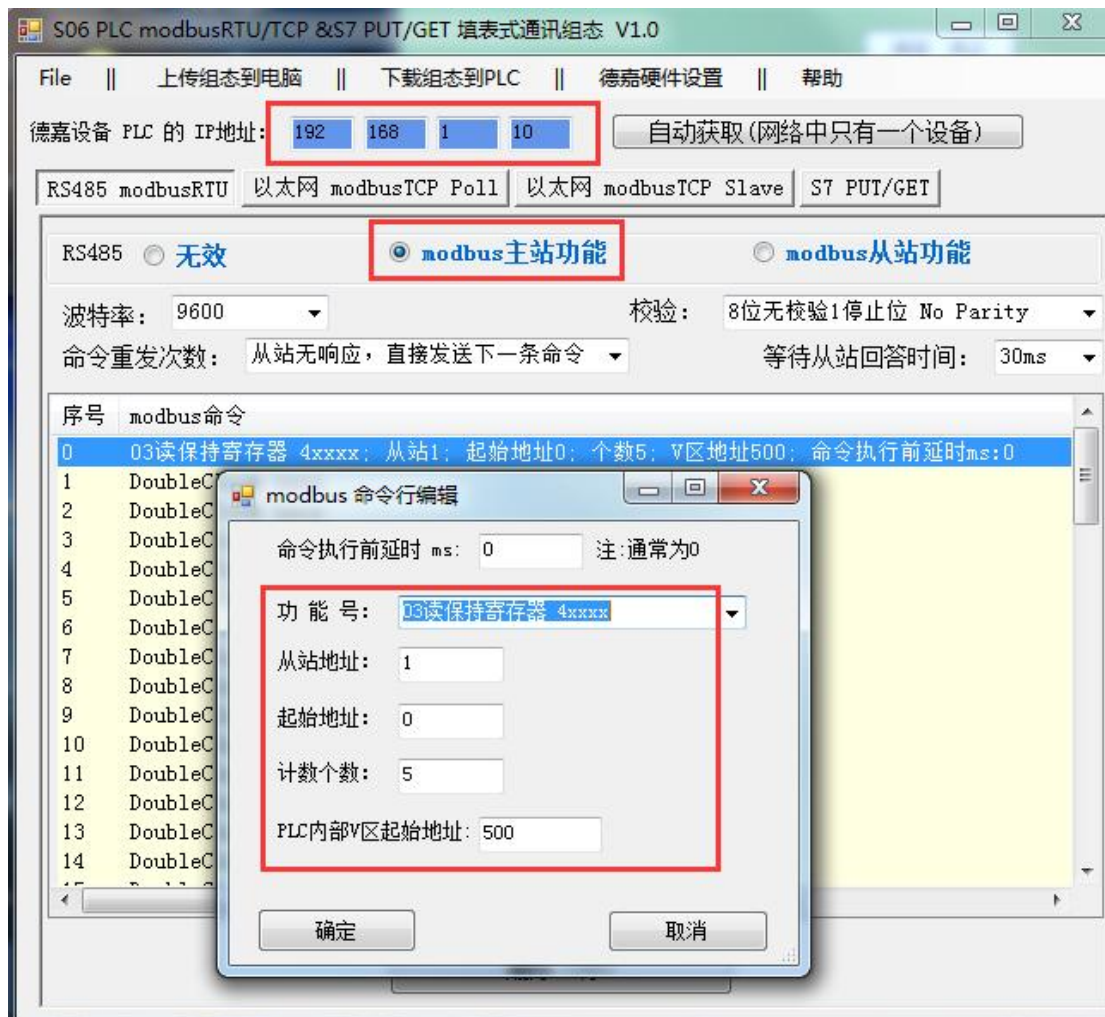
首先填写波特率、校验方式、等待从站应答时间、命令重发次数（是指 modbus 命令发送后，如果没有收到从站的正确应答，是发送下一条 modbus 命令，还是再次或多次发送本条命令）

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。

以两个 S06 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 5、V 区起始地址 500，RS485 接线方式为 A--A，B--B，方法如下：

(1) 其中一个 S06PLC，编辑 Modbus 主站命令，下载到 PLC 中

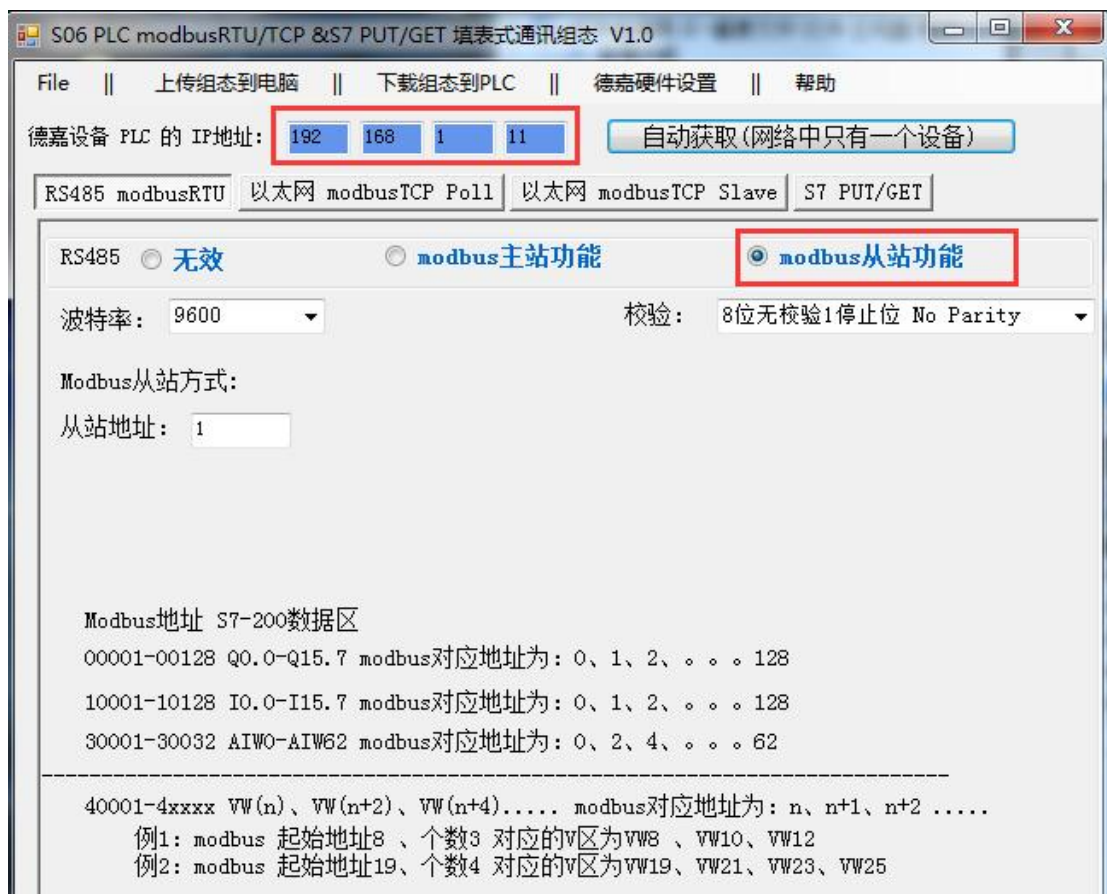


如需上传组态命令到电脑功能，先把 PLC 切换到 STOP 状态，再上传即可。

下载成功后，可以网页通过后门地址查看 Modbus 主从站方式



(2) 另一个 S06PLC，编辑 Modbus 从站命令，下载到 PLC 中



下载成功后，也可网页通过后门地址查看 Modbus 主从站方式



最后同时监控两个 PLC 的状态数据，如下：



提供 485 通讯断线诊断功能，通过 CPU 状态位监控可以判断出通讯异常情况，断线响应时长为 16s，下面为 Modbus64 条命令具体对应 CPU 状态位关系：

命令（序号）	状态位地址
0	SM200.0
1	SM200.1
.....	.....
63	SM207.6
64	SM207.7

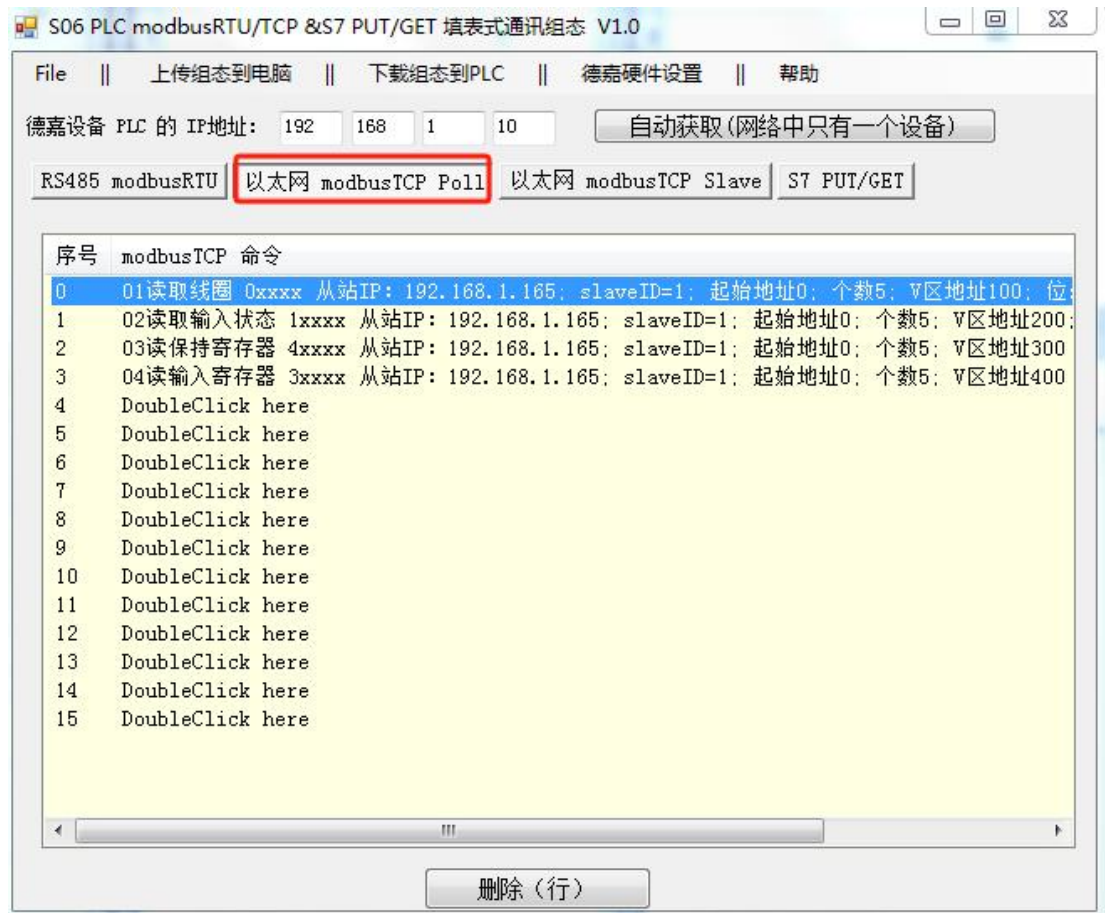
485 通讯正常时状态位地址值为 1，异常（断线或者通讯参数错误）状态位地址值为 0。

## 9 ModbusTCP 通讯（填表方式）

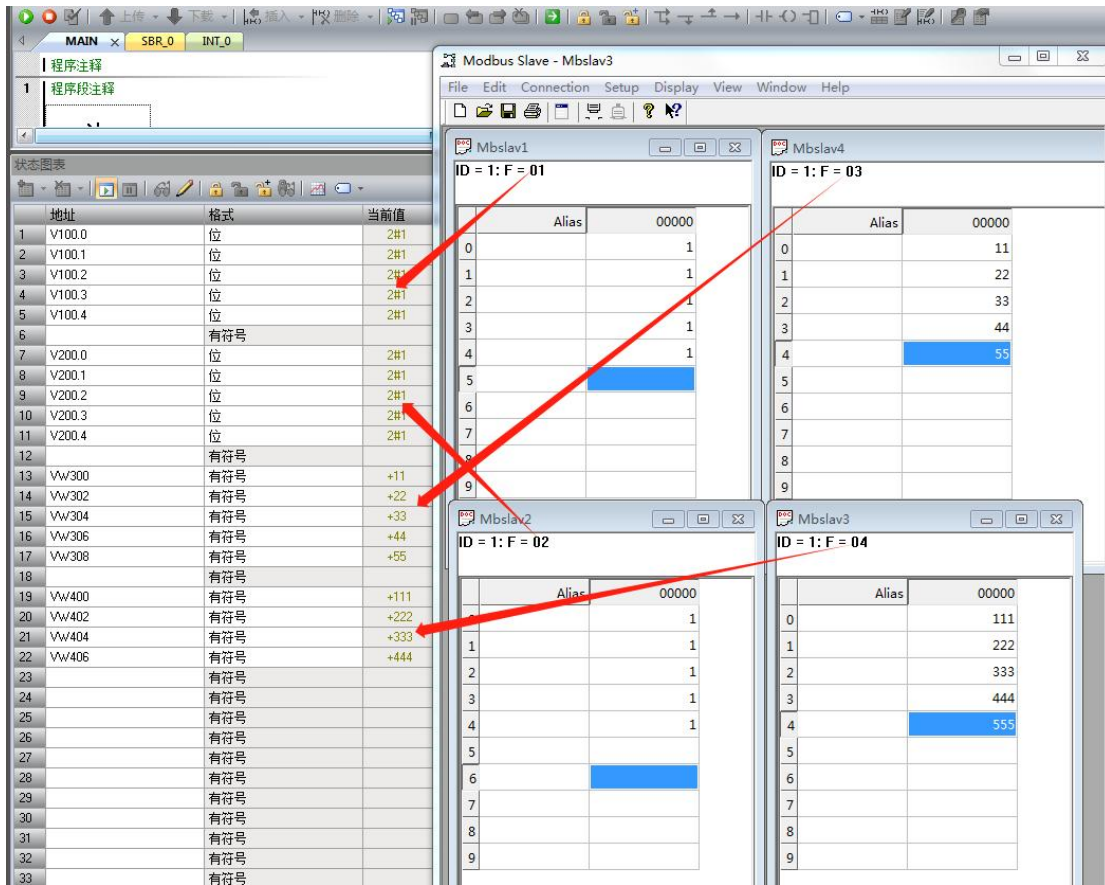
下载 PLC 通讯组态插件：[点击下载](#)

[http://www.dl-winbest.com/download/PLC\\_Config.rar](http://www.dl-winbest.com/download/PLC_Config.rar)

(1) 以 ModbusTCP 作为主站（客户端）为例，编辑如下几种命令，下载到 PLC 中



通过电脑端使用 Modbus Slave 模拟从站（服务器），该电脑 IP 地址为：192.168.1.165，建立对应功能码和连接，最终监控如下：



(2) 再以 ModbusTCP 作为从站（服务器）为例，下载到 PLC 中



通过电脑端使用 Modbus Poll 模拟主站（客户端）来读取或写入，建立对应功能码和连接，最终监控如下：

The image shows a Modbus Poll software interface with four client windows (Mbpoll1, Mbpoll2, Mbpoll3, Mbpoll4) and a PLC status table. Red arrows indicate the mapping between the PLC status table and the Modbus Poll data tables.

**PLC Status Table (状态图表):**

地址	格式	当前值
1	CPU_输出0:Q0.0	2#1
2	CPU_输出1:Q0.1	2#0
3		
4	CPU_输入0:I0.0	2#0
5	CPU_输入1:I0.1	2#1
6		
7	VW0	无符号 255
8		
9	QI.0	位 2#1
10		
11		
12		
13		
14		

**Mbpoll1 Data Table:**

Alias	00000
0	1
1	0
2	
3	
4	
5	
6	
7	
8	
9	

**Mbpoll2 Data Table:**

Alias	00000
0	0
1	1
2	
3	
4	
5	
6	
7	
8	
9	

**Mbpoll3 Data Table:**

Alias	00000
0	255
1	
2	
3	
4	
5	
6	
7	
8	
9	

**Mbpoll4 Data Table:**

Alias	00000
0	
1	
2	
3	
4	
5	
6	
7	
8	1
9	

Red arrows show the following connections:

- Mbpoll1 Alias 0 (value 1) connects to PLC address 1 (value 2#1).
- Mbpoll1 Alias 1 (value 0) connects to PLC address 2 (value 2#0).
- Mbpoll2 Alias 1 (value 1) connects to PLC address 5 (value 2#1).
- Mbpoll3 Alias 0 (value 255) connects to PLC address 7 (value 无符号 255).
- Mbpoll4 Alias 8 (value 1) connects to PLC address 9 (value 位 2#1).



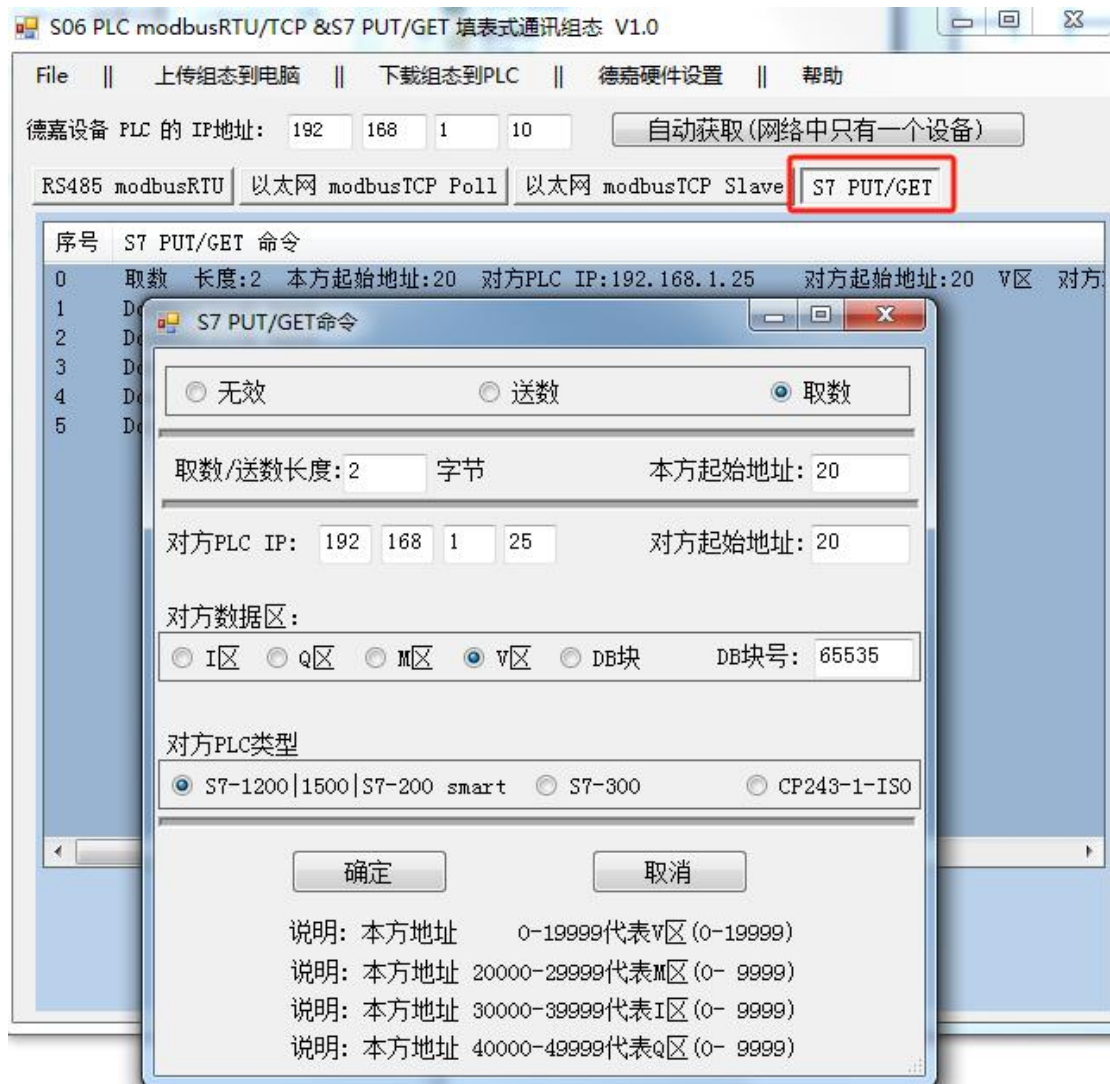
## 10 PLC 之间通讯设置（填表方式）

下载 PLC 通讯组态插件：[点击下载](#)

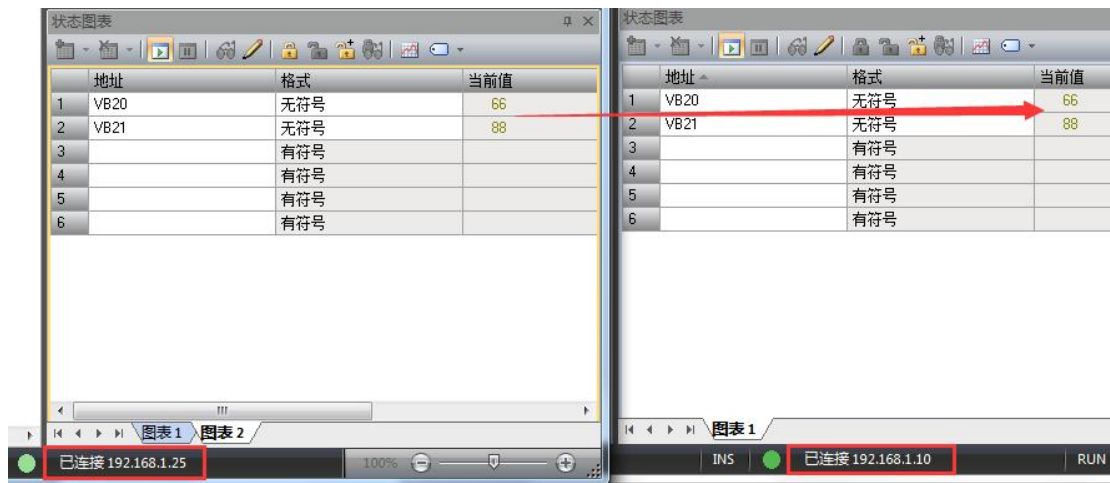
[http://www.dl-winbest.com/download/PLC\\_Config.rar](http://www.dl-winbest.com/download/PLC_Config.rar)

该方式也可以在网页中【PLC 通讯】功能中进行设置，数据是同步的，通过通讯组态插件或者是在网页中设置，两种方式选一种即可。

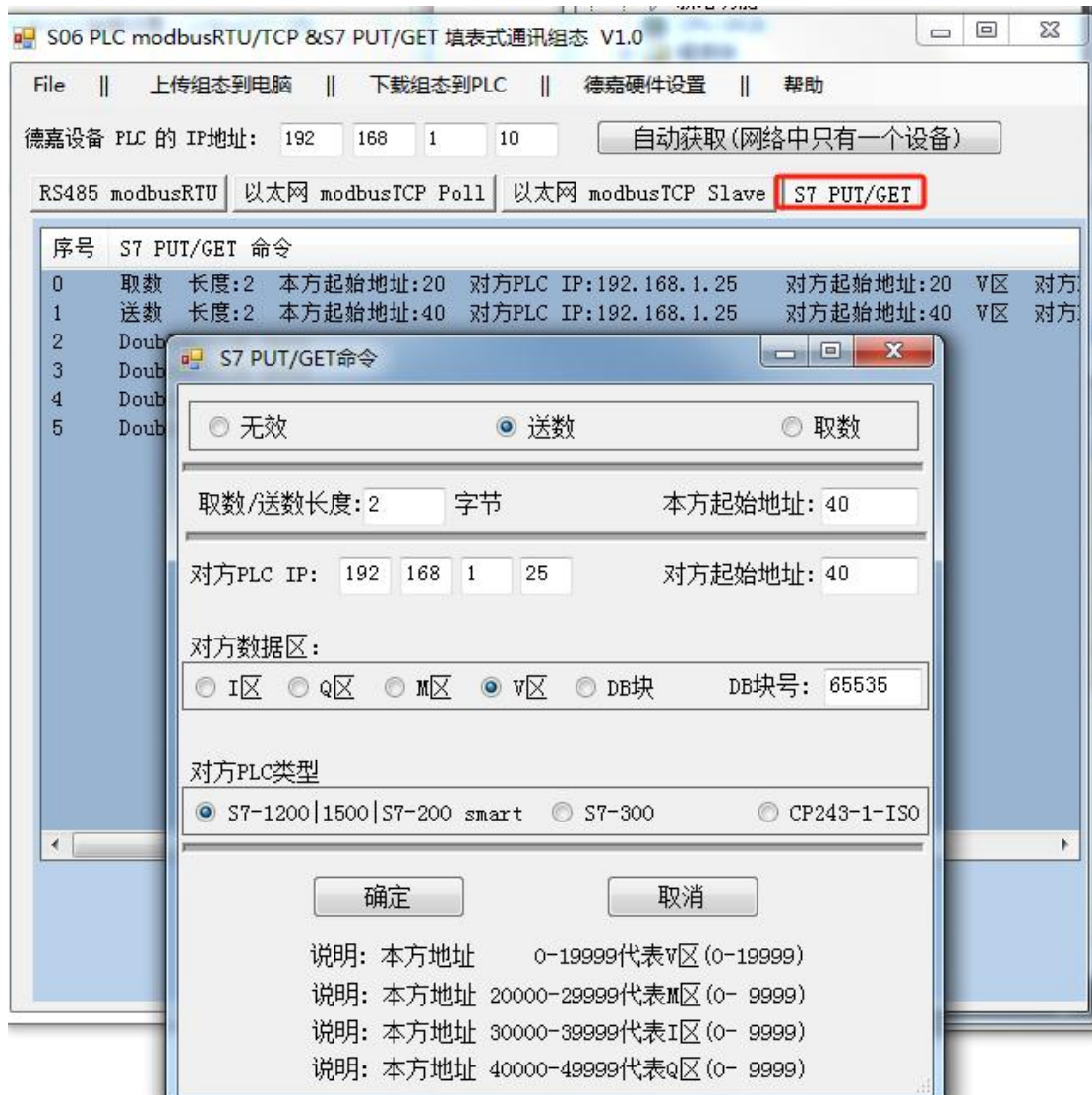
(1) 以“取数”方式为例，编辑该命令如下，其中对方 PLC 类型为 S7-200SMART，下载到 PLC



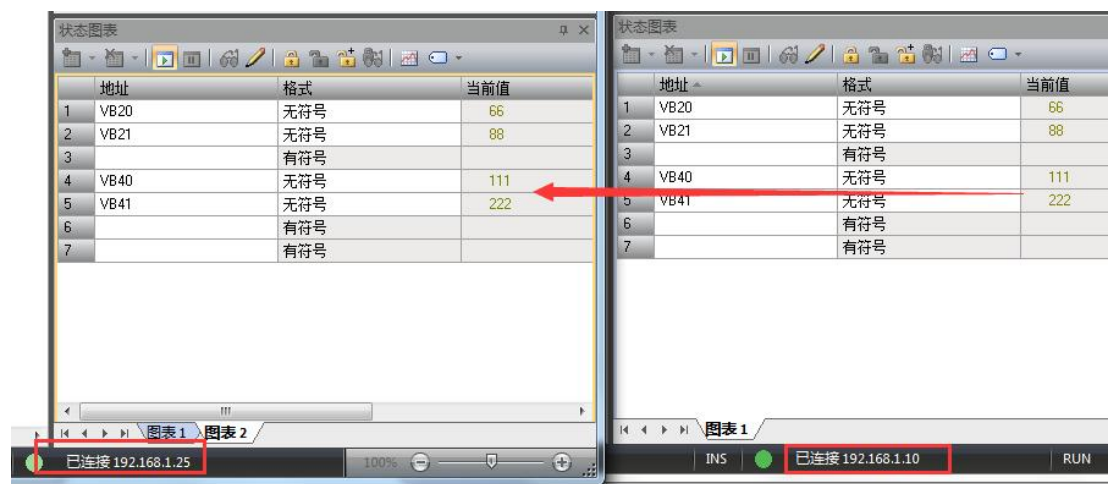
监控数据如下：



(2) 再以“送数”方式为例，编辑该命令如下，其中对方 PLC 类型为 S7-200SMART，下载到 PLC 中



监控数据如下：



可见，通讯成功，这种方式要比指令编程方式更快捷，更方便。

## 11 PLC 之间通讯实例

此例为网页中设置方式，也可通过通讯组态插件完成设置，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 S06 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. S06 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

通道:1 取数或送数

无效     送数     取数

取数/送数长度: 002 字节    本方起始地址 00100

对方 PLC IP: 192 168 001 022 [000-255]    起始地址 00000

对方数据区:     I区     Q区     M区     V区     DB块    DB块号 00000

对方PLC类型:     S7-1200|S7-200 smart|CP243(remote)     S7-300     SIEMENS CP243-1-ISO

提交    取消

返回

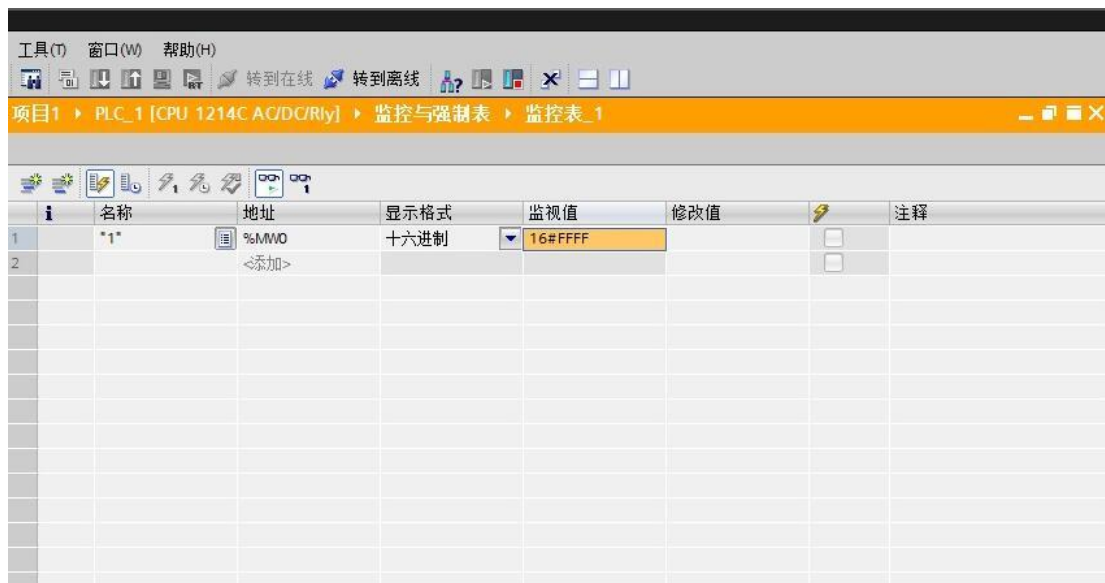
说明: 本方地址 0-19999代表V区(0-19999)  
说明: 本方地址20000-29999代表M区(0-9999 )  
说明: 本方地址30000-39999代表I区(0-9999 )  
说明: 本方地址40000-49999代表Q区(0-9999 )

Release:20160612

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

地址	名称	类型	初始值	实际值
0.0	STAT0	WORD	W#16#FFFF	W#16#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 <a href="http://blog.csdn.net/">http://blog.csdn.net/</a>	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