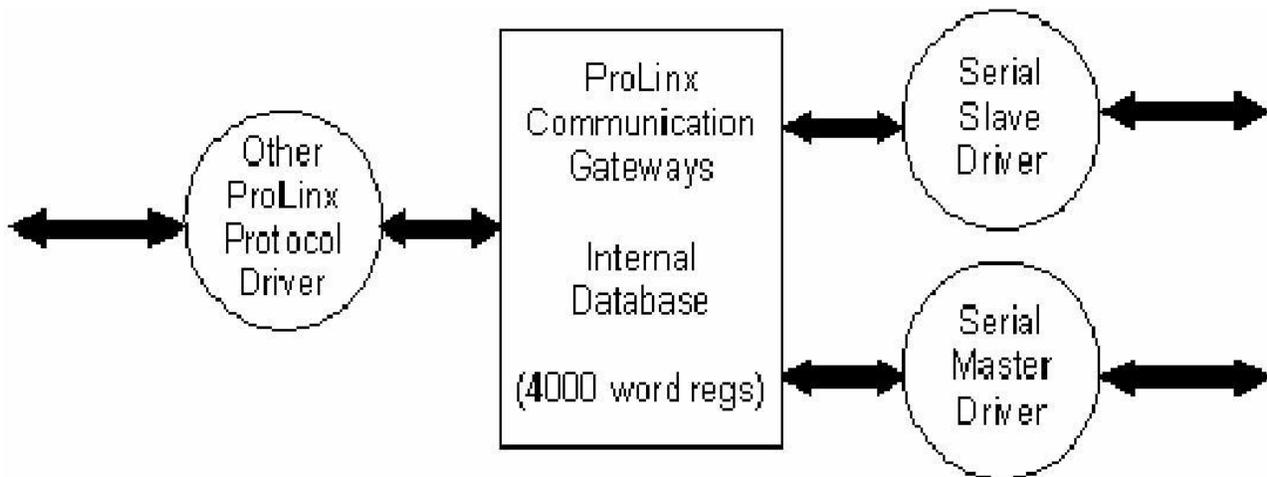


PLX31 EtherNet/IP 转 Modbus 快速向导

1. PLX31 工作原理:

PLX31 EtherNet/IP 转 Modbus 网关有两个型号, 分别为 PLX31-EIP-MBS (单独 Modbus 接口) 和 PLX31-EIP-MBS4 (4 个 Modbus 接口), 两个型号内部均为 4000 个字的寄存器区域。下图为该模块的工作示意图。



模块通过串口和 Modbus 设备交换数据到 PLX 的内部寄存器; 再通过 Ethernet/IP 和各种以太网设备进行数据交换。需要注意的是如下几点:

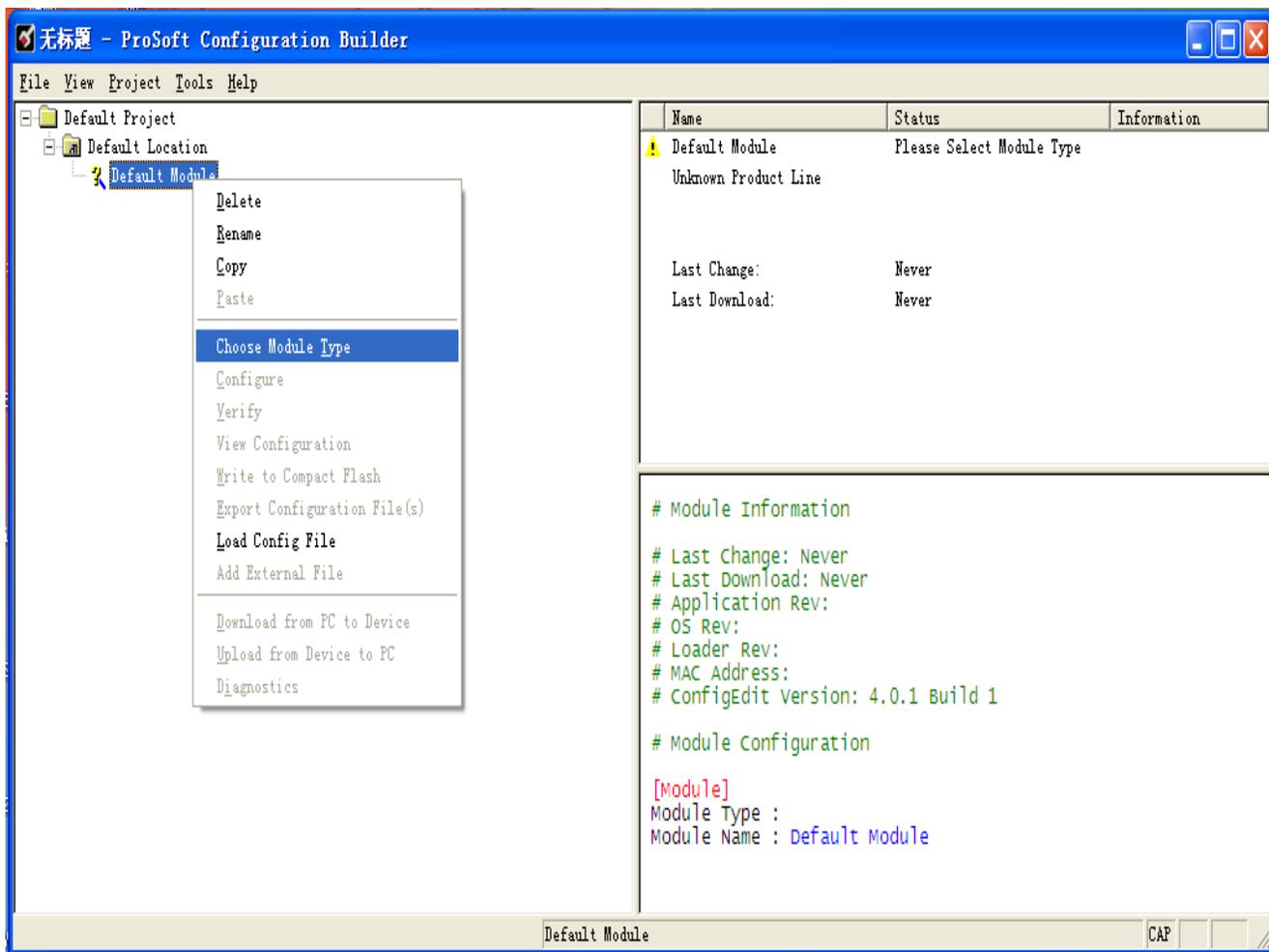
- 1) Modbus 串口支持 ASCII 码协议和 ModbusRTU 协议
- 2) Modbus 串口可组态成 RS232, RS422, RS485
- 3) 不论是 1 个还是 4 个 Modbus 串口, 都可以根据用户需求组态成主站或者从站, 每个主站端口支持 100 条 Modbus 指令。
- 4) 以太网端口可以支持两种 EtherNet/IP 设备 Class1 和 Class 3
- 5) Class 3 主要适合于连接 PLC5, SLC500, HMI 等设备 (也可以连接 Logix 系列), 当选择 Class3 的时候需要采用 MSG 指令, 该模块 EtherNet/IP 端口可以同时作为 5 个从站和 3 个主站来使用。
- 6) Class 1 主要适合于连接 Logix 系列 PLC, 是采用 I/O 扫描的方式来和 PLC 进行数据交换。
- 7) 选择 Class 1 来和 Logix 系列 PLC 通讯时, 当 Logix5000 版本在 v20 或者更高的时候, 可以直接添加 AOP 来组态以太网端口。
- 8) 选择 Class 1 来和 Logix 系列 PLC 通讯时, 当 Logix5000 版本在 v20 以下时, 需要在 I/O tree 之下添加 EtherNet/IP bridge 设备, 并且定义每个 I/O 连接的大小以及 RPI 时间, CIP 路径等等参数. (本说明采用此方式作为案例, 组态该模块以太网端口)

9) 所有 4000 个字的数据刷新时间在选择 Class 1 的时候, 最短 RPI 时间可达 5ms。

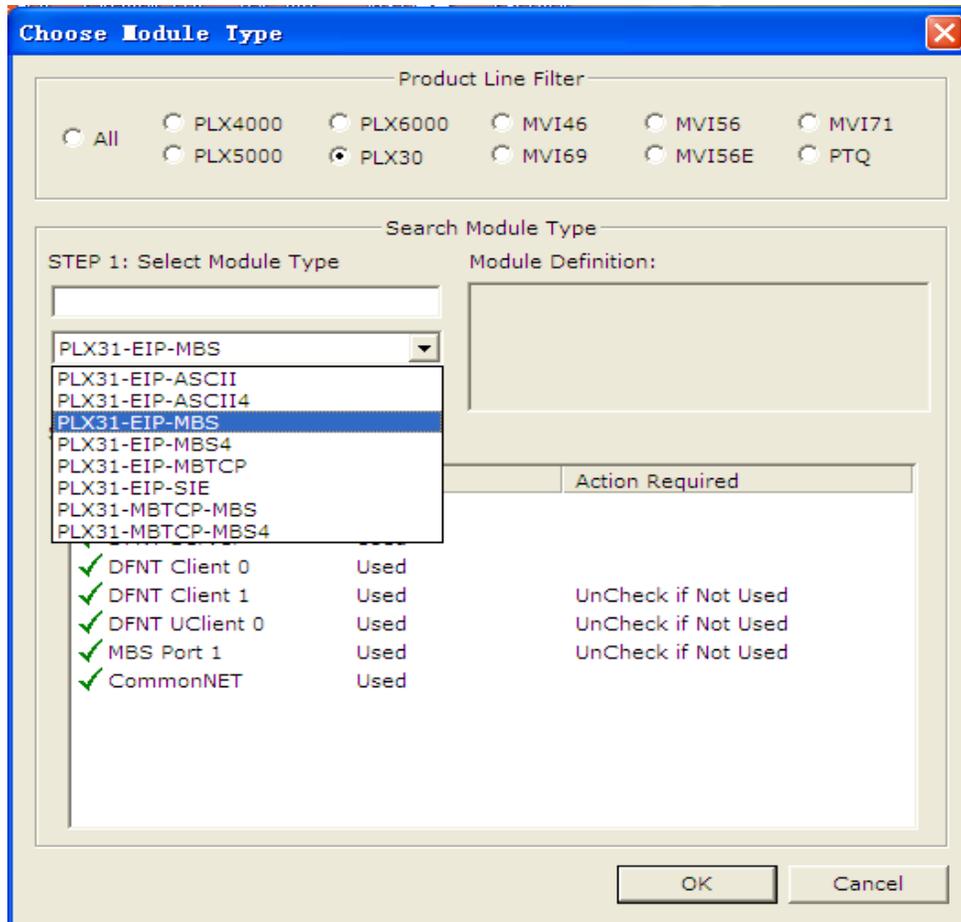
2、PLX31 配置

2.1 安装 ProSoft Configuration Builder (要求最低版本为 4.0.1.1.185)

2.2 打开 PCB , File-new 新建一个项目, 然后右键点击 **default module** 选择 **choose module type** 如下图:

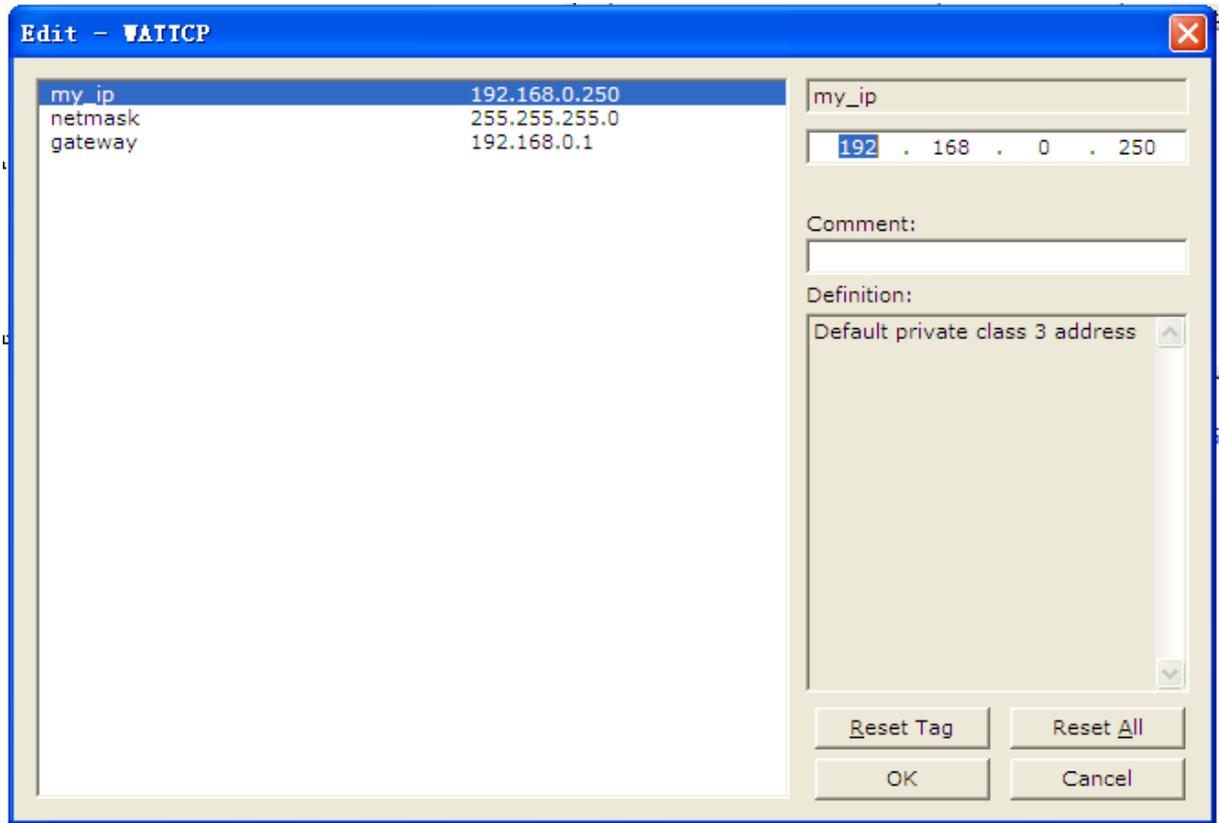


2.3 选择 PLX31 并在下拉框选择 PLX31-EIP-MBS, 并点击 OK 确认。

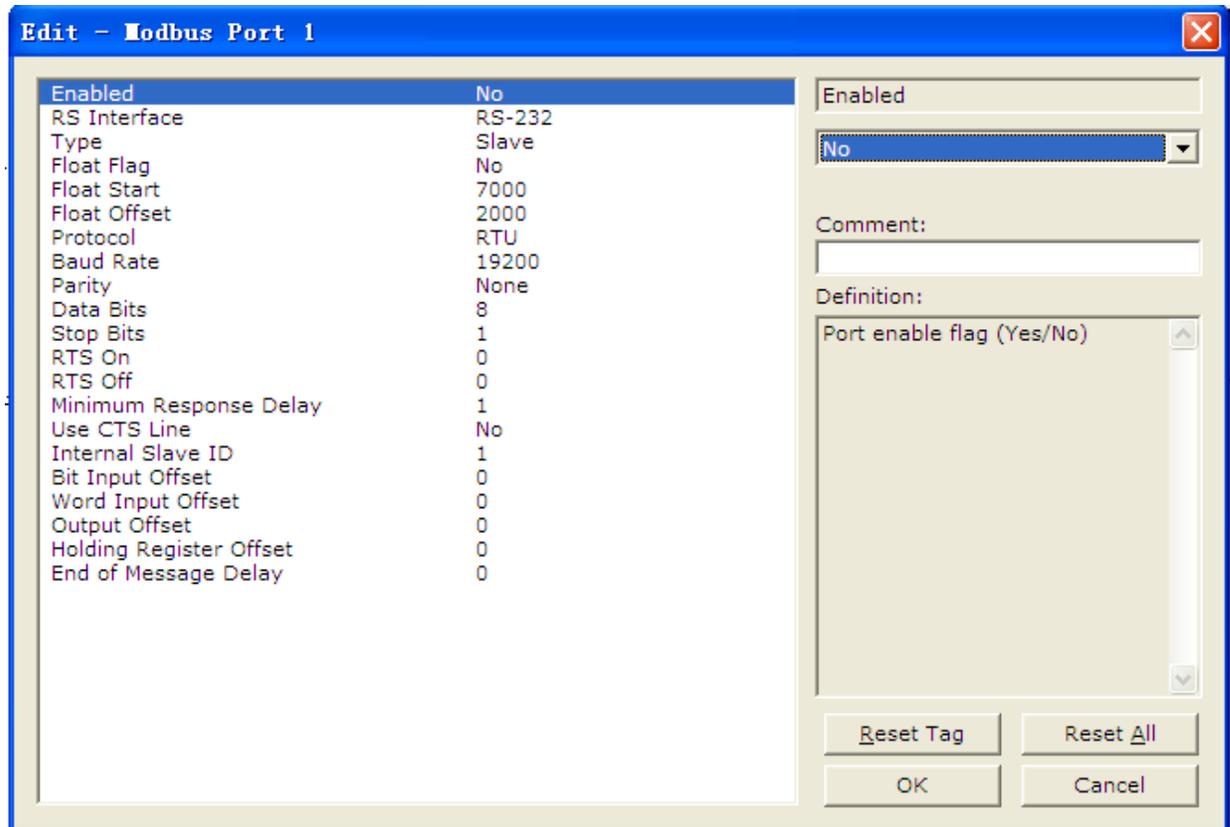


2.4 展开模块 PLX31-EIP-MBS 前面的“+”号会出现模块的配置息；

2.4.1 配置模块的 IP 地址，双击 Ethernet Configuration 配置模块 IP 地址，默认 IP 为 192.168.0.250；该 IP 即为模块的 IP 地址，也是 Ethernet/IP 协议 IP 地址。



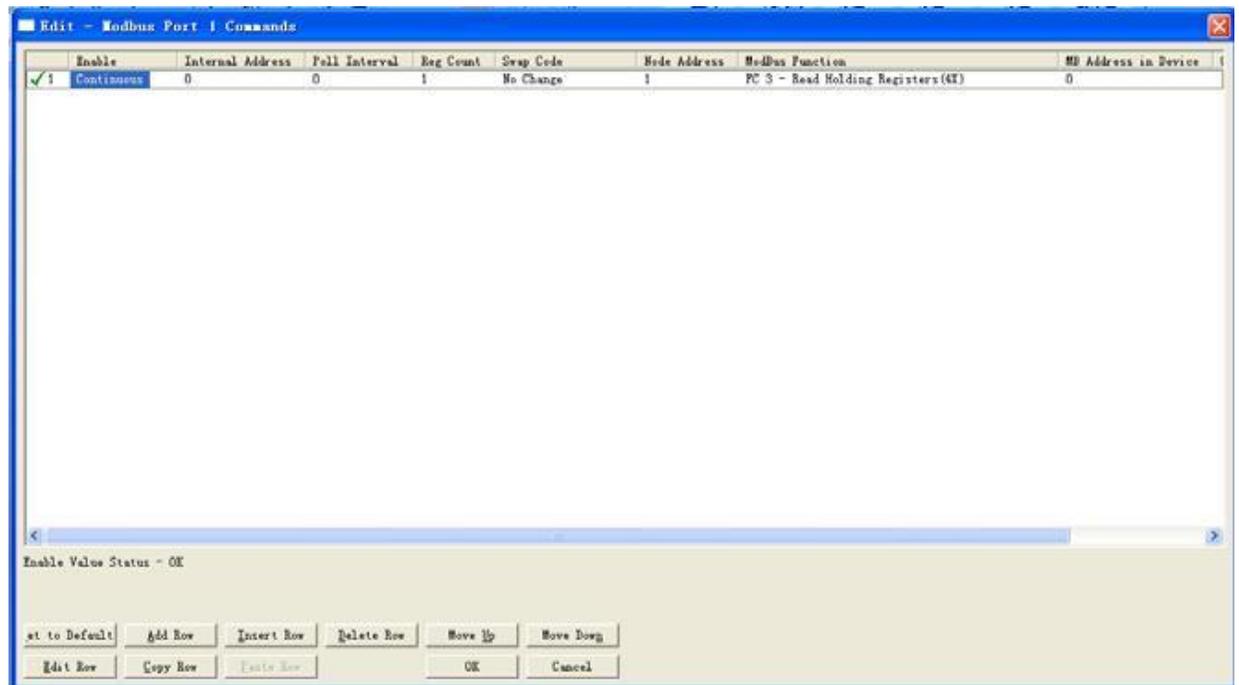
2. 4. 2 配置模块 Modbus 协议，展开 MBS Port1 前面的“+”号，并双击 Modbus Port1 出现如下界面，该界面用于配置 Modbus 端口，详细参数说明如下：



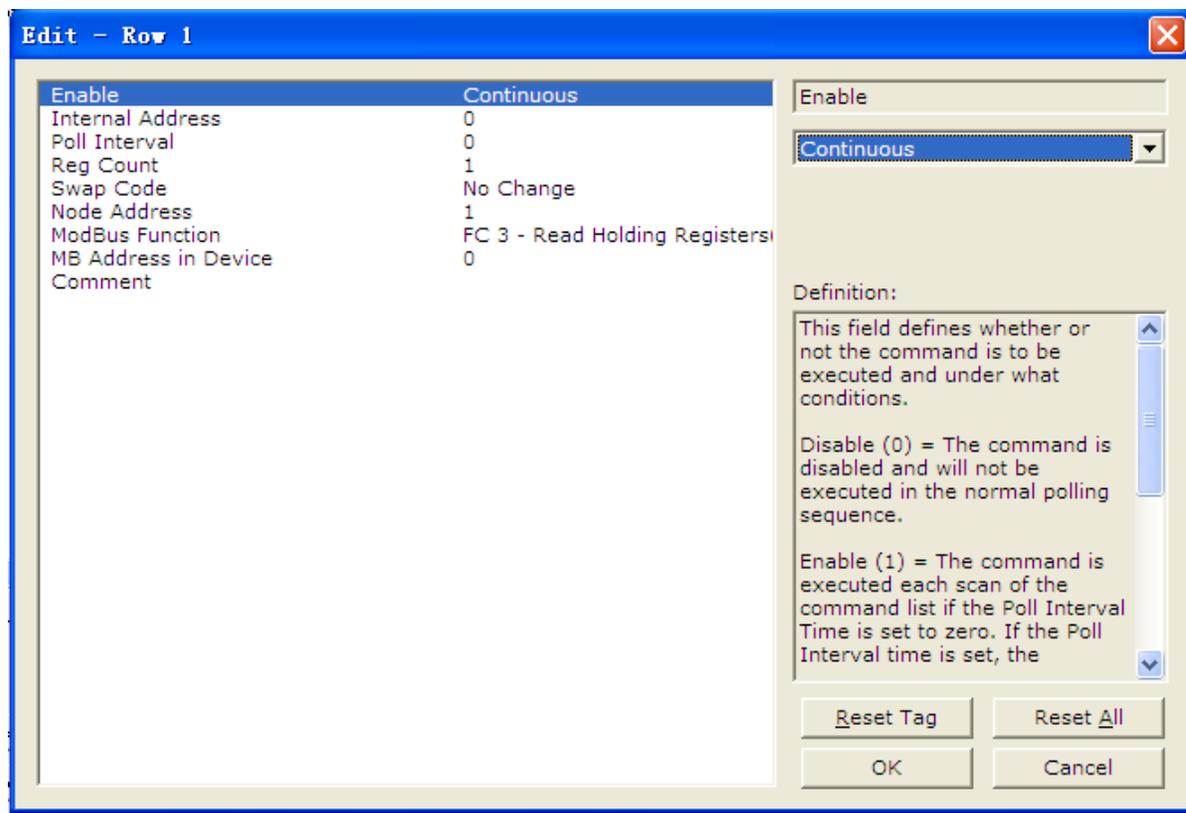
需要配置的参数见下表，其他参数选择默认即可。

Enabled	端口使能，配置为 yes
Rs interface	可配置为 Rs232/Rs485/Rs422
Type	如果做 Modbus 主站配置为 Master，如果作为从站配置为 Slave。
Protocol	可选择 Modbus RTU 协议或者 Modbus ASCII 协议，一般配置为 RTU 协议
Baud Rate、Parity、Data Bits、 Stop Bits	波特率、奇偶校验、数据位、停止位该四个串口参数需与从站设备一致。

若模块作为主站需配置 Modbus 轮询指令，双击 Modbus Port1 Command 会现
轮询指令配置界面，点击“Add row”按钮出现如下界面：



双击新增加的行进入到指令配置参数界面，每个端口最大支持 100 条指令。详细参数说明如下：



Enable: 指令使能, 可配置为 Disabled、Continuous, Event command、conditional;

一般如果读数据指令配置为 Continuous; 写数据指令配置为 Conditional。

Internal address: 模块内部数据库地址, 用于指定读过来的数据所存储的寄存器位置, 或者从哪个寄存器往外写数据。

Poll interval: 轮询间隔, 如果指令小于 5 条可配置为 0, 超过 5 条可配置为 1。

Reg Count: 所读写的寄存器数量, 如果读写的是 16 位的字, 则该参数为字的数量, 如果为 bit, 则为位的数量。

Swap code: 字节交换, 如果读取的数据高位发生变化, 可通过该参数调整。

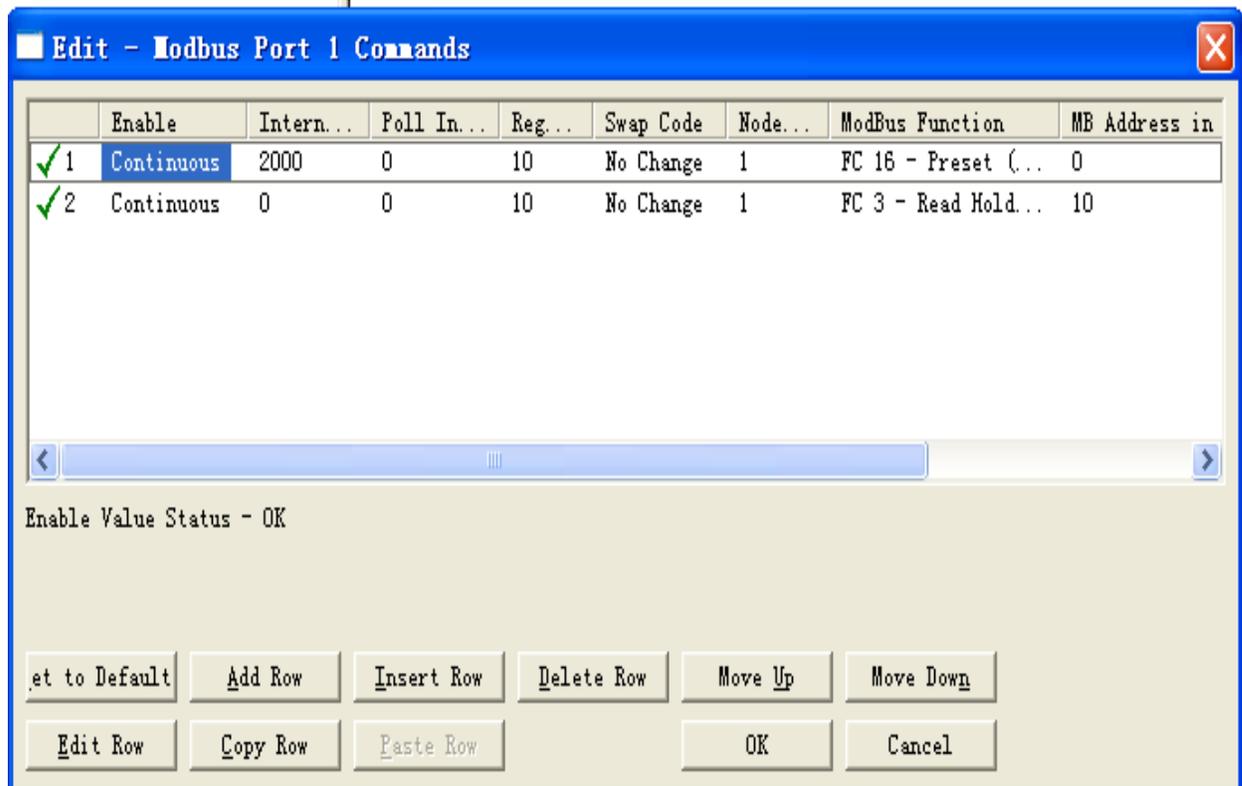
Node Address: 从站设备节点地址

Modbus Function: 功能码, 常用功能码如下:

- | | |
|----------------|----------------|
| 1、FC1: 读 0x 的值 | FC5: 对 0x 单个写 |
| 2、FC2: 读 1x 的值 | FC15: 对 0x 连续写 |
| 3、FC3: 读 4x 的值 | FC6: 对 4x 单个写 |
| 4、FC4: 读 3x 的值 | FC16: 对 4x 连续写 |

MB address in device: 所读取设备的数据地址。

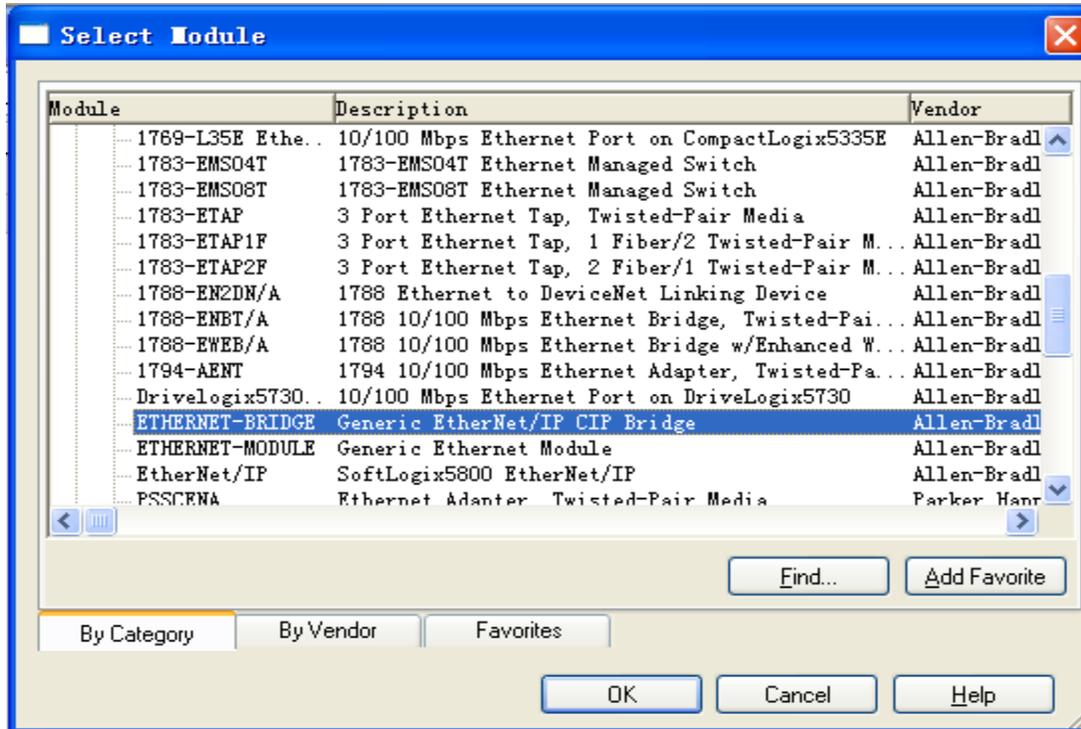
这里我们配置两条命令，第一条是写命令，第二条是读指令，如下图：



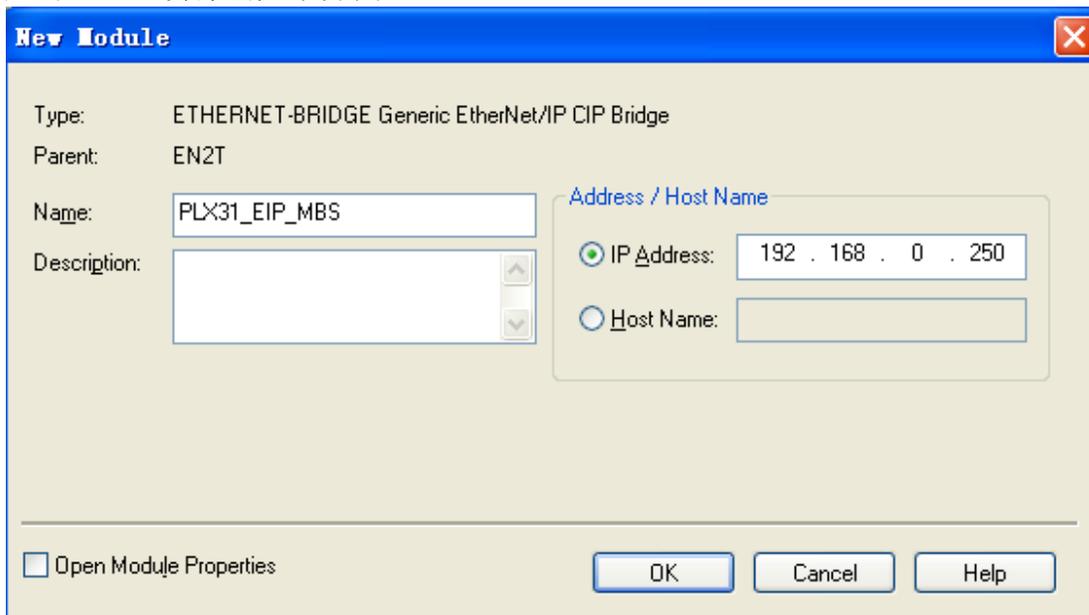
通过 2.4.2 的操作可将 Modbus 的数据读写模块的内部寄存器，下面介绍如何通过 EtherNet/IP 协议操作数据库。

2.4.3 配置 Ethernet/IP 协议：

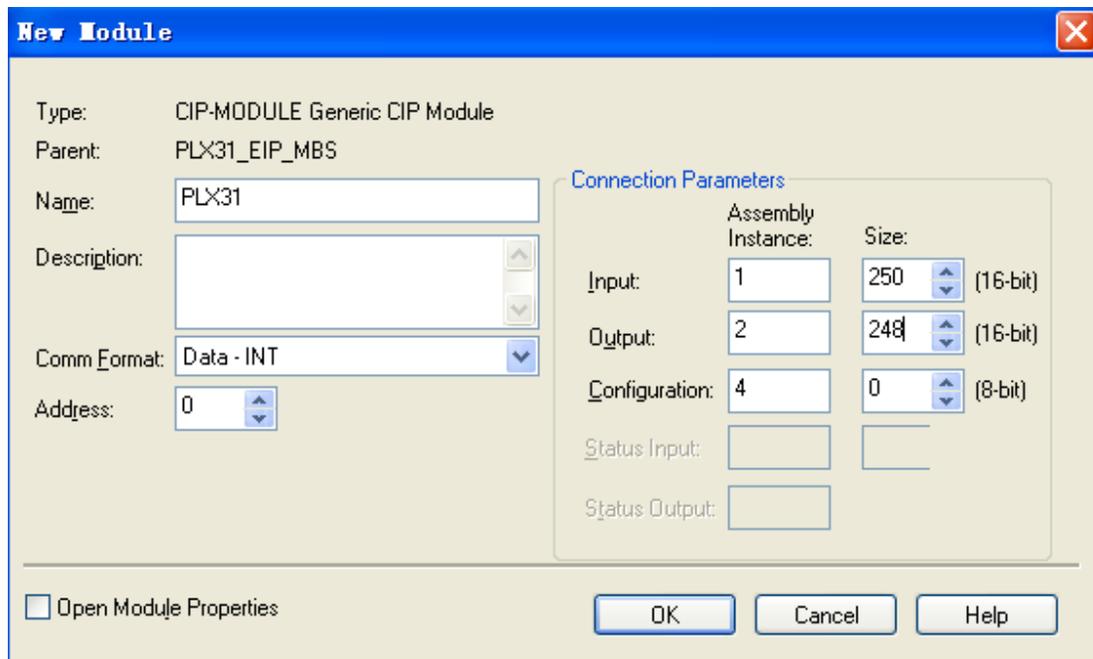
在 Logix5000 里建立 IO 模块；在 Ethernet IP 下增加 Ethernet-BRIDGE 模块如下：



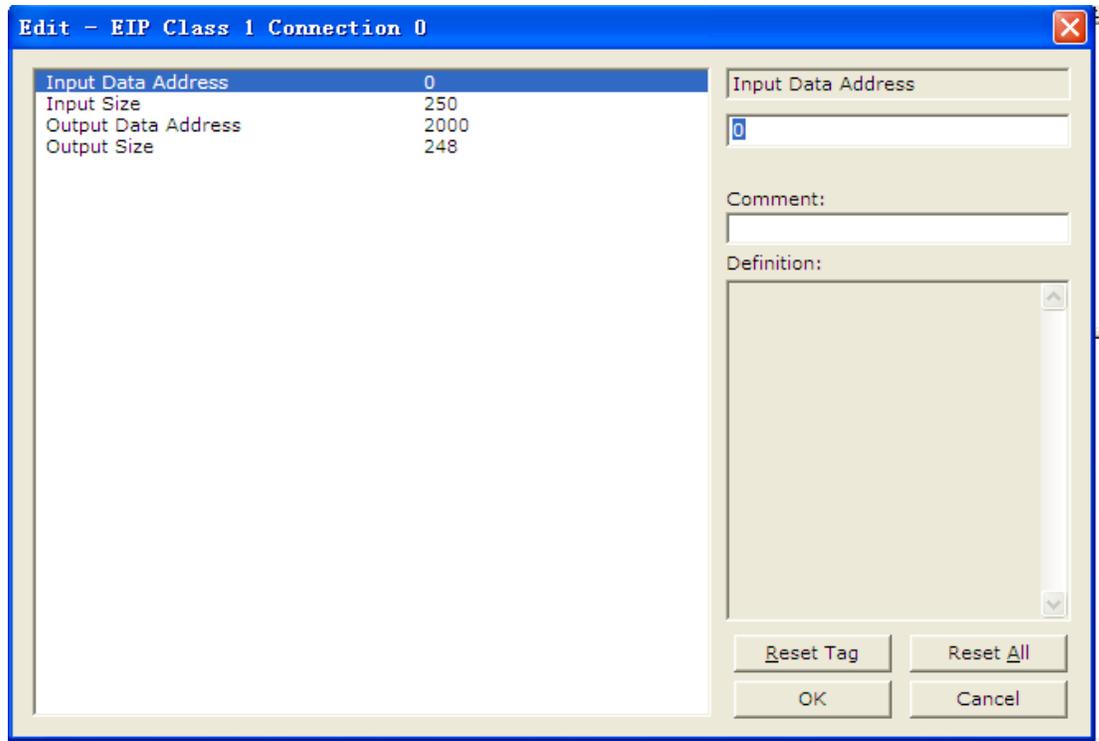
点击“OK”会弹出如下界面：



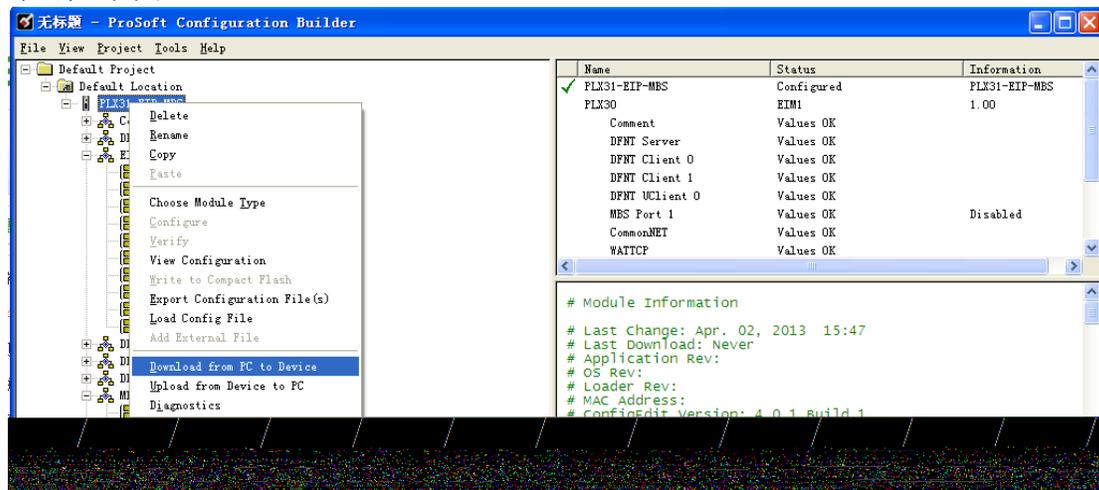
在这个界面里面对该模块进行命名何配置 IP 地址，IP 地址为模块在 PCB 配置的 IP，点击“OK”保存。右击“CIP BUS”选择“New Module”弹出如下界面：



数据类型设置为 Data-INT，输入/输出数据配置按上图配置。这样可以直接将模块的数据映射到标签EIP_MBS:I.Data 和 EIP_MBS:O.Data 里。至于把模块那些寄存器的数据映射到输入区，哪些映射到输出区，通过 PCB配置，展开 EIP Class 1 Connection 前面的“+” 双击 EIP Class 1 Connection 0，配置输入输出寄存器分配，默认输入寄存器为 0-1999，输出寄存器为 2000-3999。界面如下：



Input size 设置于 output size 设置于 Logix5000 里设置一致。
至此所有配置结束。最后将配置文件下载到模块，下载方法如下：
右击 PLX31-EIP-MBS 选择 Download from PC to Device 选择模块 IP 地址下载
即可如下图：

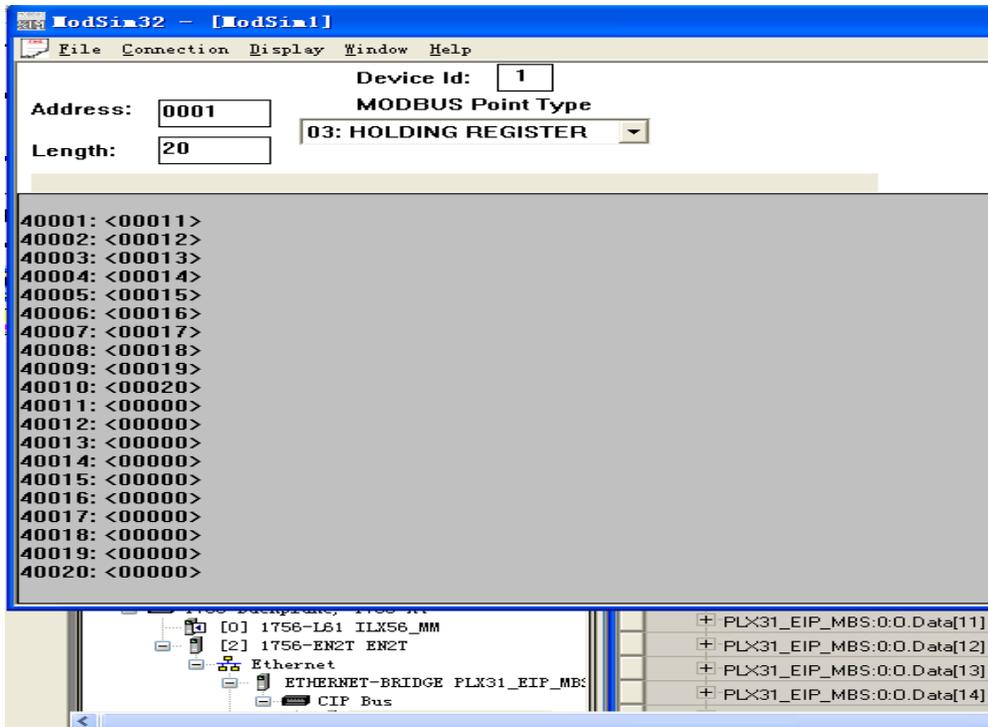


以上就是这个模块的配置步骤。下面我们用 Modsim32 作为从站 进行简单测试，我们在上面 MODBUS 命令中配置两条命令，分别想 Modsim32 中从 40001 开始的 10 个字的存储区中写数据和读取 40011 开始的十个字的数据。我们首先在 Logix5000 中

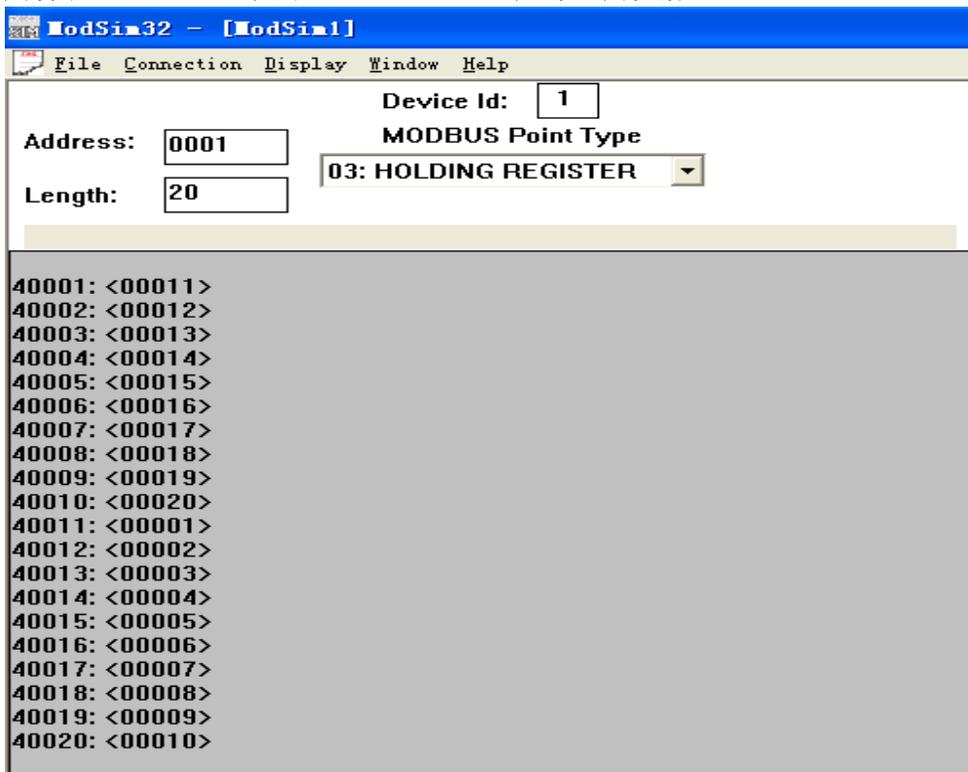
的 PLX31_EIP_MBS:0:0:DATA 中的前十个字中写入数据如下图:

Name	Value	Force Mask	Style	Data Type
PLX31_EIP_MBS:0:0:Data	{...}	{...}	Decimal	INT[248]
PLX31_EIP_MBS:0:0:Data[0]	11		Decimal	INT
PLX31_EIP_MBS:0:0:Data[1]	12		Decimal	INT
PLX31_EIP_MBS:0:0:Data[2]	13		Decimal	INT
PLX31_EIP_MBS:0:0:Data[3]	14		Decimal	INT
PLX31_EIP_MBS:0:0:Data[4]	15		Decimal	INT
PLX31_EIP_MBS:0:0:Data[5]	16		Decimal	INT
PLX31_EIP_MBS:0:0:Data[6]	17		Decimal	INT
PLX31_EIP_MBS:0:0:Data[7]	18		Decimal	INT
PLX31_EIP_MBS:0:0:Data[8]	19		Decimal	INT
PLX31_EIP_MBS:0:0:Data[9]	20		Decimal	INT
PLX31_EIP_MBS:0:0:Data[10]	0		Decimal	INT
PLX31_EIP_MBS:0:0:Data[11]	0		Decimal	INT
PLX31_EIP_MBS:0:0:Data[12]	0		Decimal	INT
PLX31_EIP_MBS:0:0:Data[13]	0		Decimal	INT
PLX31_EIP_MBS:0:0:Data[14]	0		Decimal	INT
PLX31_EIP_MBS:0:0:Data[15]	0		Decimal	INT
PLX31_EIP_MBS:0:0:Data[16]	0		Decimal	INT

这时 Modsim32 中的数据如下:



同样在 Modsim32 中的 40011-40020 写入如下数据:



这时查看 Logix5000 中的 PLX31_EIP_MB:0: I:DATA 中的前十个字如下:

Name	Value	Force Mask	Style	Data Type
[-] PLX31_EIP_MBS:0:I.Data	{...}	{...}	Decimal	INT[250]
[+] PLX31_EIP_MBS:0:I.Data[0]	1		Decimal	INT
[+] PLX31_EIP_MBS:0:I.Data[1]	2		Decimal	INT
[+] PLX31_EIP_MBS:0:I.Data[2]	3		Decimal	INT
[+] PLX31_EIP_MBS:0:I.Data[3]	4		Decimal	INT
[+] PLX31_EIP_MBS:0:I.Data[4]	5		Decimal	INT
[+] PLX31_EIP_MBS:0:I.Data[5]	6		Decimal	INT
[+] PLX31_EIP_MBS:0:I.Data[6]	7		Decimal	INT
[+] PLX31_EIP_MBS:0:I.Data[7]	8		Decimal	INT
[+] PLX31_EIP_MBS:0:I.Data[8]	9		Decimal	INT
[+] PLX31_EIP_MBS:0:I.Data[9]	10		Decimal	INT
[+] PLX31_EIP_MBS:0:I.Data[10]	0		Decimal	INT
[+] PLX31_EIP_MBS:0:I.Data[11]	0		Decimal	INT

通讯完成