SIEMENS

操作指南 • 9/2017

基于 S7-300/400 CPU 集成 PN 接口的 Modbus TCP 在 TIA Portal 的使用入门

PN,CPU,MB,TCP,TIA

https://support.industry.siemens.com/cs/cn/zh/view/ 109750622

目录

1 Modbus TCP 通讯概述

MODBUS TCP 是简单的、中立厂商的用于管理和控制自动化设备的 MODBUS 系列通讯协议的派生产品，显而易见，它覆盖了使用 TCP/IP 协议的“Intranet”和“Internet”环境中 MODBUS 报文的用途。协议的最通用用途是为诸如 PLC 以及连接其它简单域总线或 I/O 模块的网关服务的。

MODBUS TCP 使 MODBUS_RTU 协议运行于以太网，MODBUS TCP 使用 TCP/IP 和以太网在站点间传送 MODBUS 报文，MODBUS TCP 结合了以太网物理网络和网络标准 TCP/IP 以及以 MODBUS 作为应用协议标准的数据表示方法。MODBUS TCP 通信报文被封装于以太网 TCP/IP 数据包中。与传统的串口方式，MODBUS TCP 插入一个标准的 MODBUS 报文到 TCP 报文中，不再带有数据校验和地址。

1.1 通讯所使用的以太网参考模型

Modbus TCP 传输过程中使用了 TCP/IP 以太网参考模型的 5 层：

第一层：物理层，提供设备物理接口，与市售介质/网络适配器相兼容。

第二层：数据链路层，格式化信号到源/目硬件址数据帧。

第三层：网络层，实现带有 32 位 IP 地址报文包。

第四层：传输层，实现可靠性连接、传输、查错、重发、端口服务、传输调度。

第五层：应用层，Modbus 协议报文。

1.2 Modbus TCP 数据帧

Modbus 数据在 TCP/IP 以太网上传输，支持 Ethernet II 和 802.3 两种帧格式，Modbus TCP 数据帧包含报文头、功能代码和数据 3 部分，MBAP 报文头 (Modbus Application Protocol)分 4 个域，共 7 个字节。

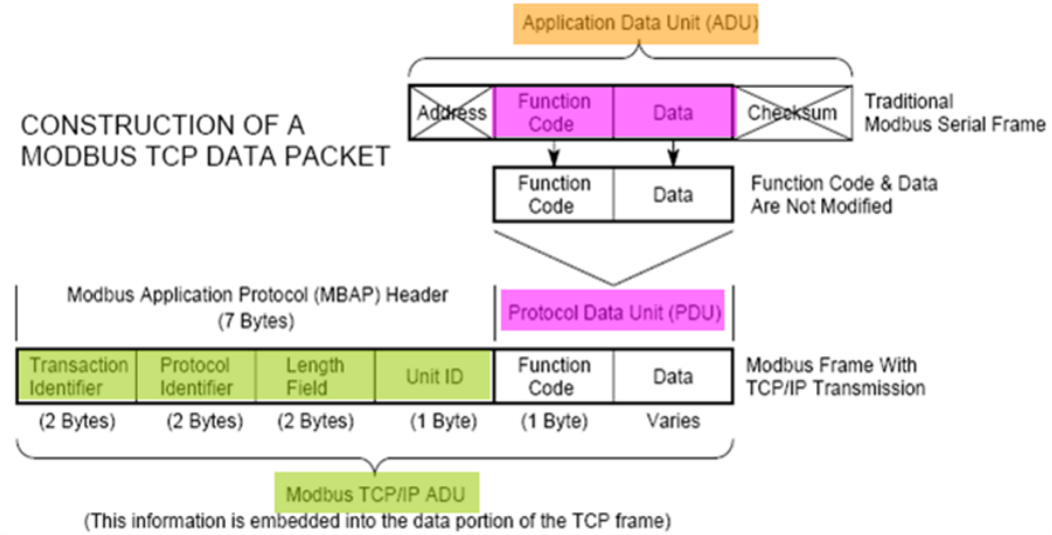


图 1-1 MBAP 报文头

域	长度	描述	客户机	服务器
Transaction ID	2字节	Modbus请求/响应事务处理的识别	客户机启动	服务器从接收的请求中重新复制
Protocol ID	2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
Length	2字节	随后字节的数量	客户机启动（请求）	服务器（响应）启动
Unit ID	1字节	远程从站的识别ID	客户机启动	服务器从接收的请求中重新复制

图 1-2 MBAP 报文头说明

1.3 Modbus TCP 使用的端口号

（1） PLC 作为 Modbus 服务器时，按缺省协议使用 Port 502 通信端口，在 Modus 客户端程序中设置任意通信端口，

（2） PLC 作为 Modbus 客户端时，无须设置本机端口号；如要指定客户端端口号，为避免与其他通讯协议的冲突一般建议 2000 开始可以使用。

2 S7-300/400 集成 PN 口 Modbus TCP 通讯概述

本文适用于带有集成 PN 接口的 SIMATIC S7-300、S7-400 CPU 和 IM 151-8 PN/DP CPU 的软件产品。相关指令允许在带有集成 PN 接口的 SIMATIC CPU 和支持 Modbus TCP 协议的设备之间进行通信。

根据客户端——服务器原理进行数据传输。传输过程中，可以将 SIMATIC S7 用作客户端，也可以用作服务器。

从 TIA Portal V14 SP1 开始软件中增加了 Modbus TCP V2.0 版本的指令，可用于 SIMATIC S7-300、S7-400 CPU 和 IM 151-8 PN/DP CPU 与支持 Modbus TCP 的通信伙伴进行通信，如下图 2-1 所示：

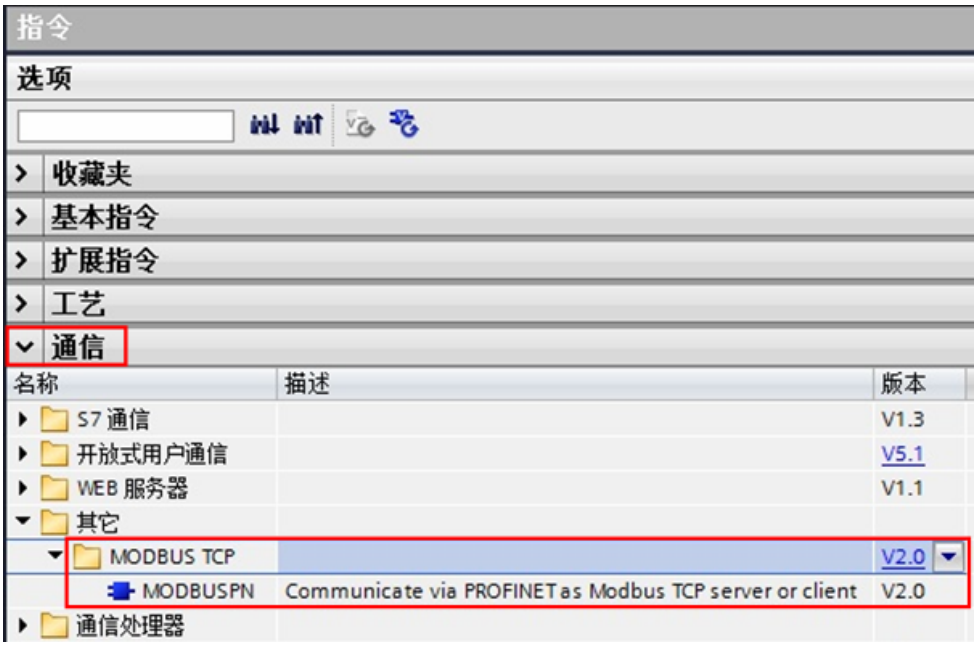


图 2-1 指令版本

下面例子将分别介绍如何配置 315-2PN/DP 为 Modbus/TCP 的 Server，Client 与通信伙伴建立通信，测试例程中用到的软硬件如图 2-2 所示：

名称	数量	订货号
SIMATIC CPU315-2PN/DP(FW V3.2)	1	6ES7 315-2EH14-0AB0
SIMATIC STEP7 Professional V14 SP1	1	6ES7 822-1AA04-0YA5
MODBUS/TCP PN-CPU V2 Single license	1	6AV6 676-6MB20-3AX0
Modscan32 用于在 PC 中模拟 Modbus Client	1	网上免费下载
Modsim32 用于在 PC 中模拟 Modbus Server	1	网上免费下载

图 2-2 例程中用到的软硬件列表

3 配置 PN CPU 作为 Modbus TCP Server 与通信伙伴建立通讯

下面以 S7-300 单站系统及 Modscan32 软件为例，详细介绍如何将 S7-300 单站系统通过 CPU 集成 PN 口配置为 Modbus TCP Server，Modscan32 为 Client 进行 Modbus TCP 通讯。

3.1 组态硬件

在 TIA V14 SP1 中创建一个新项目（项目名称：PN_MB_TCP），选择项目版本 V14 SP1，如图 3-1 所示：

图 3-1 创建新项目

然后选择“添加新设备”——>“控制器”，选择正确的 CPU 型号，设备名称“server”，如图 3-2 所示：

图 3-2 添加新设备

接着，在“设备视图”中，选择 CPU 以太网口，设置 IP 地址，如图 3-3 所示：

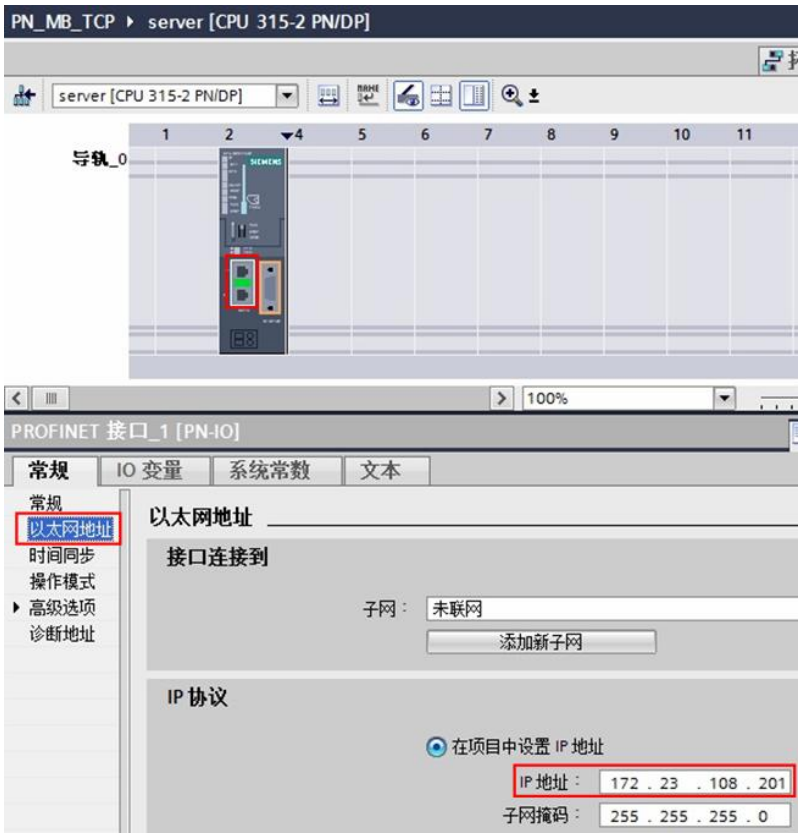


图 3-3 设置 IP 地址

3.2 编程

（1）OB1 调用 Modbus TCP 指令

在项目的 OB1 组织块中调用 Modbus TCP 指令，如图 3-4 所示：

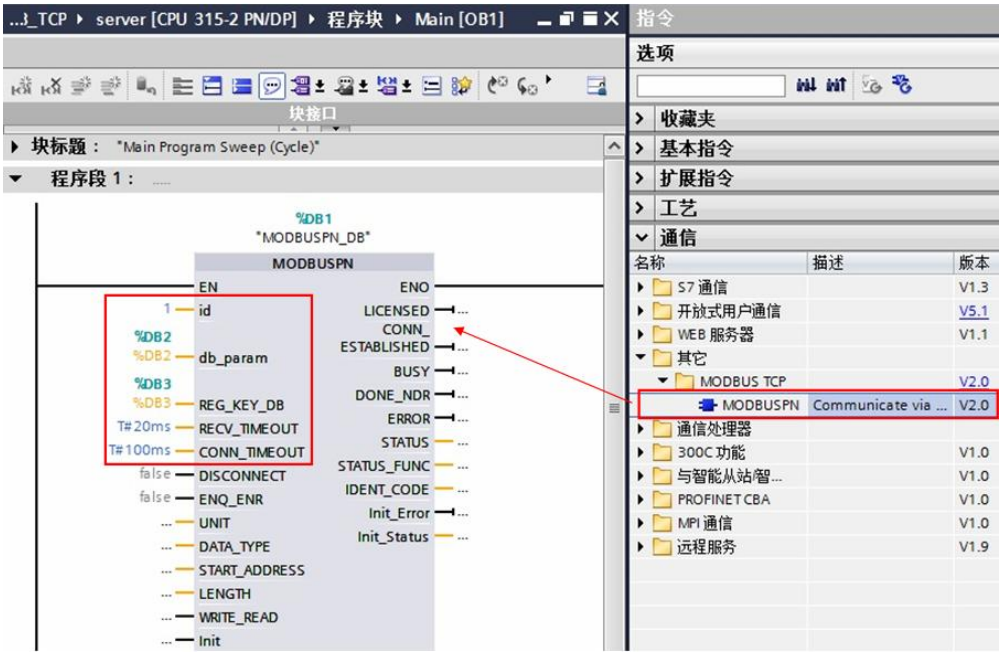


图 3-4 调用 Modbus TCP 指令

以下为部分管脚说明（其它管脚信息请查看在线帮助）：

id：连接 ID 必须与参数 DB 中相关的 id 参数相同。

db_param：参数 DB 的编号，包含此 modbus 块实例的连接参数和 modbus 数据参数。CPU 决定该参数的取值范围。DB 编号 0 为系统保留，不允许使用。以纯文本格式输入 DB 编号“DBxy”。

REG_KEY_DB：具有可用于授权的注册表项的数据块。

RECV_TIMEOUT：对从耦合伙伴接收数据进行监视。超出监视时间后，将发出错误信号并终止连接。 最小值为 20 ms。

在“S7 为服务器”模式下将 RECV_TIMEOUT 设置为 < 20 ms，则使用默认值 1.2 s。RECV_TIMEOUT 监视 TCP 流的运行系统。不考虑各个客户端请求之间的中断。

CONN_TIMEOUT：监视调用建立或终止所用的时间。如果在组态的监视时间内无法成功建立或终止连接，则会在输出 STATUS 中显示相应的错误消息。最小值为 100 ms。

在“S7 为服务器”操作模式下，如果将 CONN_TIMEOUT 设置为 < 100 ms，则会使用默认值 5 s。

（2）创建参数数据块

创建数据块 DB2（名称 DB_param），选择类型为“MB_PN_PARAM”，如图 3-5 所示：

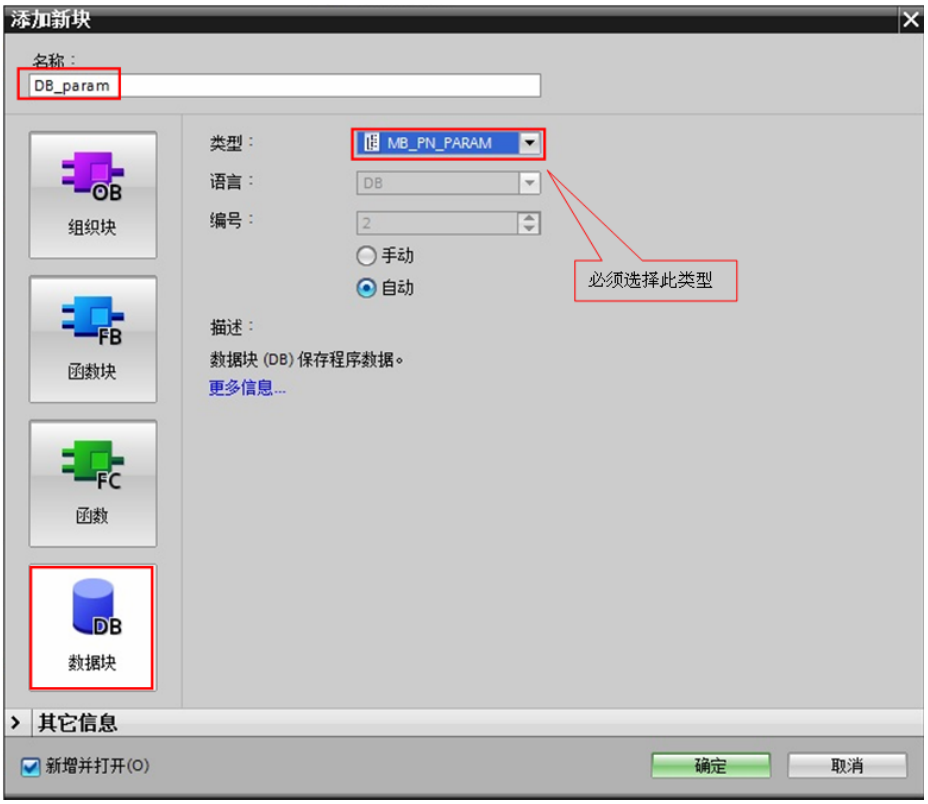
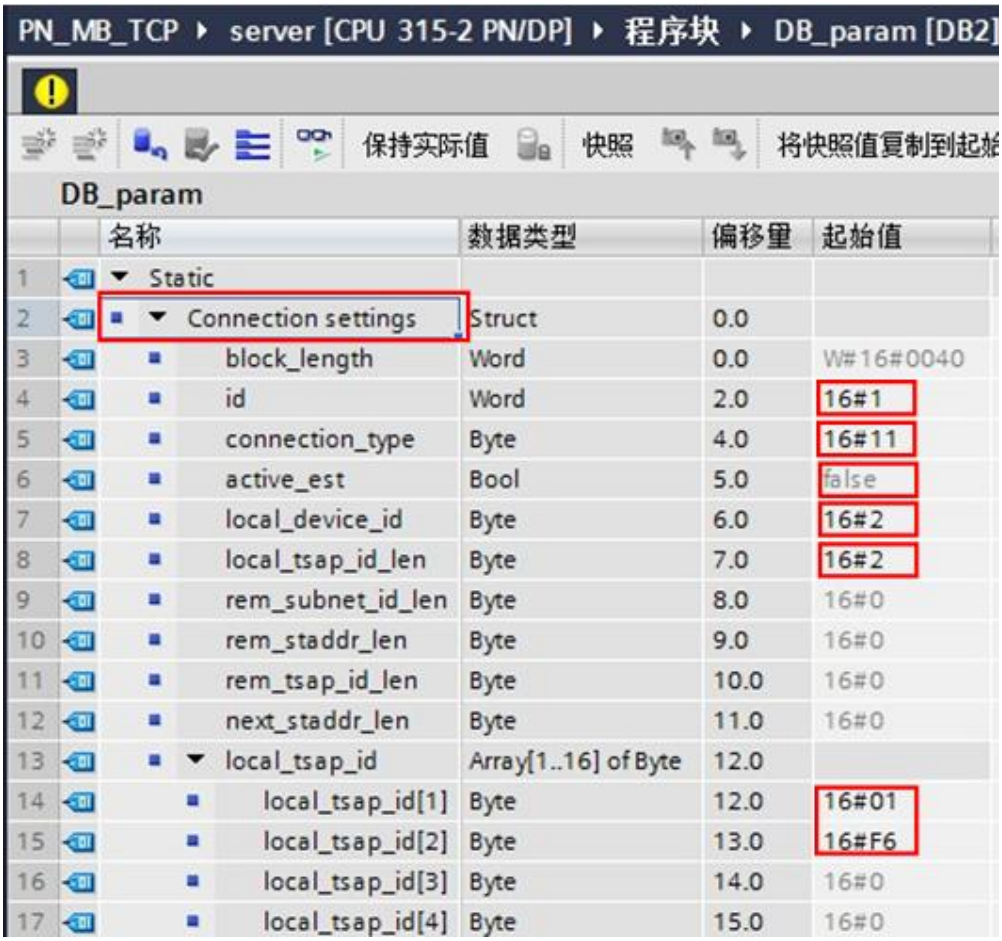


图 3-5 创建参数数据块

打开参数数据块，展开结构变量“Connection_settings”，并按下图参数设置，如图 3-6 所示：



	名称	数据类型	偏移量	起始值
1	Static			
2	Connection settings	Struct	0.0	
3	block_length	Word	0.0	W#16#0040
4	id	Word	2.0	16#1
5	connection_type	Byte	4.0	16#11
6	active_est	Bool	5.0	false
7	local_device_id	Byte	6.0	16#2
8	local_tsap_id_len	Byte	7.0	16#2
9	rem_subnet_id_len	Byte	8.0	16#0
10	rem_staddr_len	Byte	9.0	16#0
11	rem_tsap_id_len	Byte	10.0	16#0
12	next_staddr_len	Byte	11.0	16#0
13	local_tsap_id	Array[1..16] of Byte	12.0	
14	local_tsap_id[1]	Byte	12.0	16#01
15	local_tsap_id[2]	Byte	13.0	16#F6
16	local_tsap_id[3]	Byte	14.0	16#0
17	local_tsap_id[4]	Byte	15.0	16#0

图 3-6 Connection_settings 参数

以下为部分参数说明（其它参数信息请查看在线帮助）：

id：每个 PN CPU 与通信伙伴之间的连接都需要一个连接 ID。如果有多个通信伙伴，则每个逻辑连接会使用不同的连接 ID。该连接 ID 在参数数据块中包含的“连接参数块”中组态。连接 ID 唯一地描述 CPU 与链接伙伴之间的连接，取值范围为 1 到 4095。必须在此处输入参数块中的连接 ID；该 ID 在整个 CPU 中必须唯一。

connection_type：建立连接的连接类型通过 TCON 指令定义。CPU 决定必须要设置的值。

TCP（兼容模式）：B#16#01，针对 CPU 315 或 317 <= FW V2.3 。

TCP：B#16#11，针对 CPU 315 或 317 >= FW V2.4、IM 151-8 PN/DP CPU、CPU314C、CPU319、CPU412、CPU414 和 CPU416。

该信息可能因固件不同而有所不同。

active_est：该参数表示连接建立类型，主动或被动。Modbus 客户端负责建立主动连接而 Modbus 服务器负责建立被动连接。

主动连接的建立：TRUE

被动连接的建立：FALSE

local_device_id：定义所用 PN CPU 的 IE 接口。根据不同的 PN CPU 类型，需要不同的设置。

CPU 类型	local_device_id
IM 151-8 PN/DP CPU	B#16#1
CPU 314C、315 或 317	B#16#2
CPU 319	B#16#3
CPU 412、414 或 CPU 416	B#16#5

local_tsap_id_len：参数 local_tsap_id（= 本地端口号）的长度是特定的。

主动连接的建立： 0

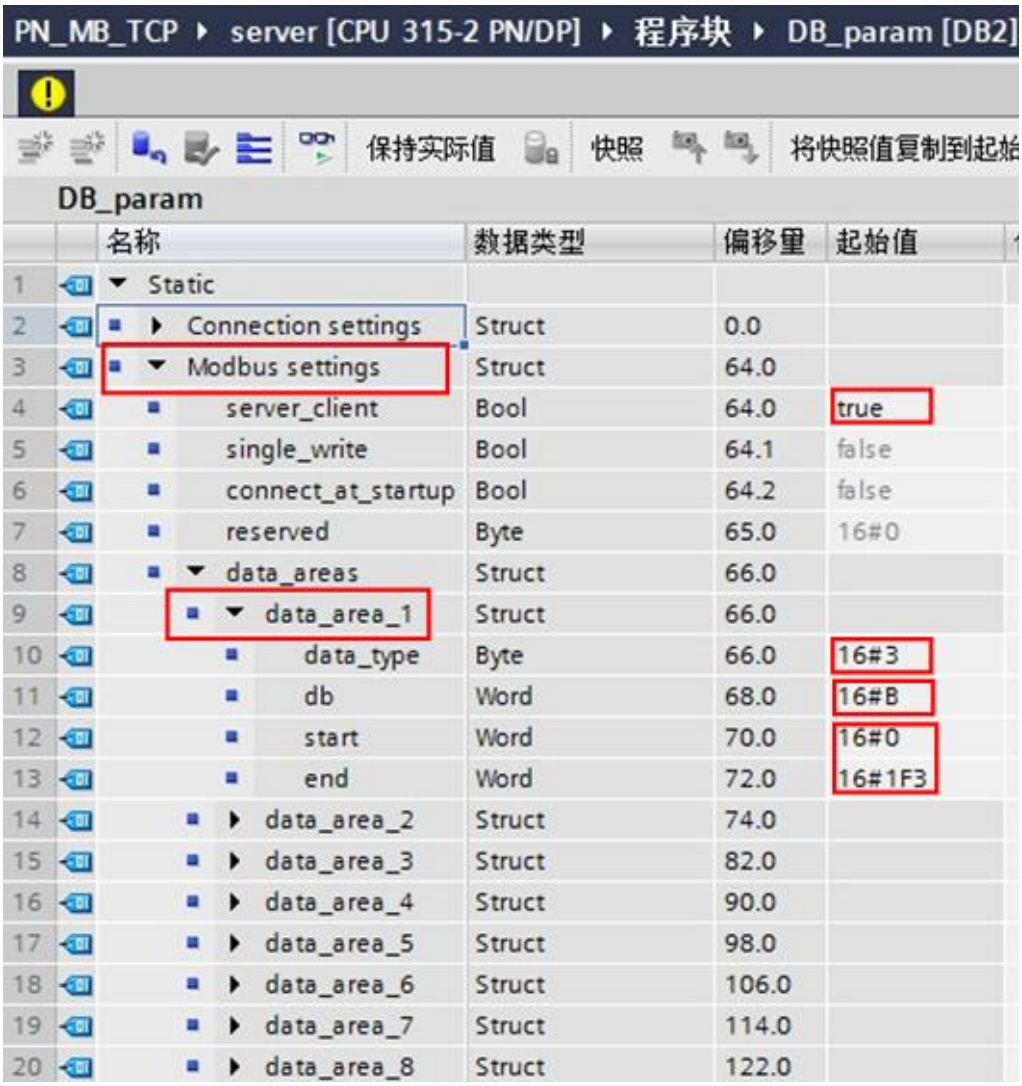
被动连接建立：2

local_tsap_id：使用该参数设置本地端口号。表示类型会因 connection_type 参数不同而有所不同。CPU 决定值范围。端口号在 CPU 中必须唯一。

对于 connection_type B#16#01： local_tsap_id[1] local_tsap_id[2] local_tsap_id[3-16]	 用十六进制格式表示的端口号 low byte 用十六进制格式表示的端口号 high byte B#16#00
对于 connection_type B#16#11： local_tsap_id[1] local_tsap_id[2] local_tsap_id[3-16]	 用十六进制格式表示的端口号 high byte 用十六进制格式表示的端口号 low byte B#16#00

本例中，CPU 为 315-2PN，connection_type B#16#11，端口号设置为 502（16#01F6），则对应于 local_tsap_id[1]= 16#01，local_tsap_id[2]= 16#F6。

在参数数据块中，继续展开结构变量“Modbus_settings”，并按下图参数设置，如图 3-7 所示：



	名称	数据类型	偏移量	起始值
1	Static			
2	Connection settings	Struct	0.0	
3	Modbus settings	Struct	64.0	
4	server_client	Bool	64.0	true
5	single_write	Bool	64.1	false
6	connect_at_startup	Bool	64.2	false
7	reserved	Byte	65.0	16#0
8	data_areas	Struct	66.0	
9	data_area_1	Struct	66.0	
10	data_type	Byte	66.0	16#3
11	db	Word	68.0	16#B
12	start	Word	70.0	16#0
13	end	Word	72.0	16#1F3
14	data_area_2	Struct	74.0	
15	data_area_3	Struct	82.0	
16	data_area_4	Struct	90.0	
17	data_area_5	Struct	98.0	
18	data_area_6	Struct	106.0	
19	data_area_7	Struct	114.0	
20	data_area_8	Struct	122.0	

图 3-7 Modbus_settings 参数

server_client：S7 是服务器=TRUE；S7 是客户端=FALSE。

data_areas： S7 存储器中有八个可以用于映射 MODBUS 地址的数据区。必须至少定义第一个数据区，其余七个数据区可选择性定义。根据作业类型，将从数据区读取数据或向其中写入数据。

任何作业都只能从一个 DB 读取数据或向一个 DB 写入数据。访问寄存器或位于多个 DB 中的位值时，即使编号连续无间隔，也将分为两个作业。组态时请务必注意。

一个数据块中可以映射的 Modbus 区（寄存器或位值）数目比一个消息帧可以处理的数目多。

data_type：指定该数据块中映射的 MODBUS 数据类型。如果在 data_type 中输入值 16#0，则不使用相应的区域。

标识符	数据类型	数据宽度
16#0	未使用区域	
16#1	线圈	Bit
16#2	输入	Bit
16#3	保持寄存器	Word
16#4	输入寄存器	Word

db：指定映射 MODBUS 寄存器或下面定义的位值的数据块。DB 编号 0 为系统保留，不允许使用。

DB 编号：1 到 65535（W#16#0001 到 W#16#FFFF）。

start / end：start 指定 DB 的数据字 0 中映射的第一个 Modbus 地址。end 参数定义最后一个 MODBUS 地址。

对于寄存器访问，带有最后一个 Modbus 地址输入的 S7 DB 中的数据字编号如下计算：DBW 编号 = (end – start) * 2

对于位访问，带有最后一个 Modbus 地址输入的 S7 DB 中的数据字节编号如下计算：DBB 编号 = (end – start + 7) / 8

定义的数据区不得重叠。end 参数不得小于 start。如果发生错误，指令启动将中止并提示错误。如果两个值相同，则将分配一个 Modbus 地址（1 个寄存器或 1 个位值）。

注意：数据块必须比已组态数据所需的长度多两个字节。最后的两个字节供内部使用。

（3）创建授权密钥数据块和编程错误组织块

创建授权密钥数据块 DB3（名称 REG_KEY_DB），选择类型为“全局 DB”，打开该 DB 块，创建变量“REG_KEY”，数据类型为“String[17]”，如图 3-8 所示：

图 3-8 创建授权密钥数据块

授权密钥的获取方法，请查看“MODBUSPN”指令的在线帮助，主题为“**使用参数 IDENT_CODE 和 REG_KEY_DB 进行授权**”的部分。

由于在获取授权密钥前，“MODBUSPN”指令是无授权状态，会使 CPU 报错而停机。而为了读取 CPU 的“IDENT_CODE”码，需要 CPU 运行起来，则必须添加编程错误组织块 OB121，如图 3-9 所示：

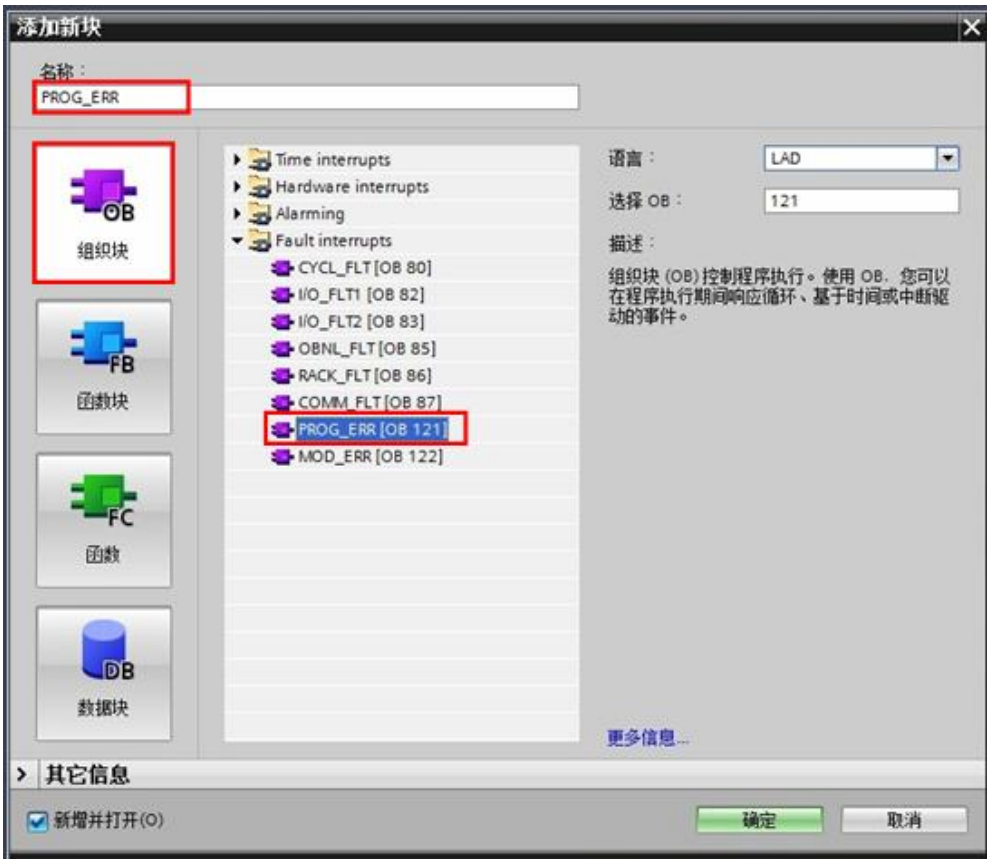


图 3-9 添加编程错误组织块 OB121

（4）创建启动组织块

创建启动组织块 OB100（名称 COMPLETE RESTART），语言为“STL”，打开该 OB 块，置位初始化位“MODBUSPN_DB”.init，如图 3-10 所示：

Init：在参数中有上升沿时，初始化 Modbus 块。只有当前没有作业正在运行时，才能执行初始化。必须通过 ENQ_ENR = FALSE 和 BUSY = FALSE 在程序中确保此条件。

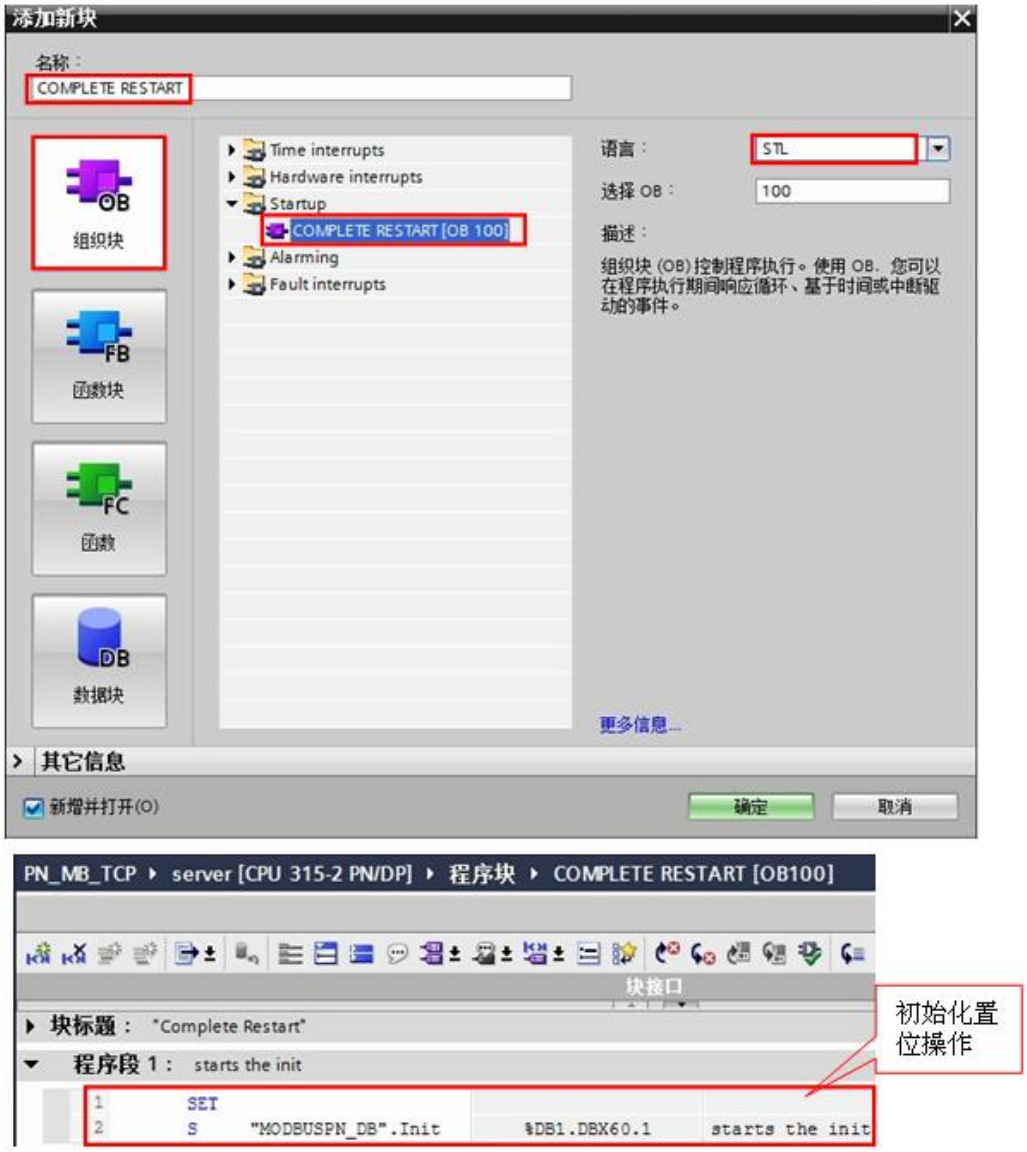


图 3-10 创建启动组织块

（5）创建全局数据块

创建全局数据块 DB11，用于关联 modbus 寄存器地址 40001~40500，如图 3-11 所示：

图 3-11 创建全局数据块

3.3 通信测试

完成上述操作后，下载项目到 CPU 中，打开 Modsan32 应用程序，下面以保持寄存器为例介绍通信测试过程。

首先，需要置位“ENQ_ENR”，使 modbus TCP 的 server 端处于等待连接的状态。

然后，在 ModScan32 的 Connection 菜单下选择 connect，并设置 ModScan32 访问作为 server 端的 PLC 的 IP 地址和端口号，如图 3-12 所示：

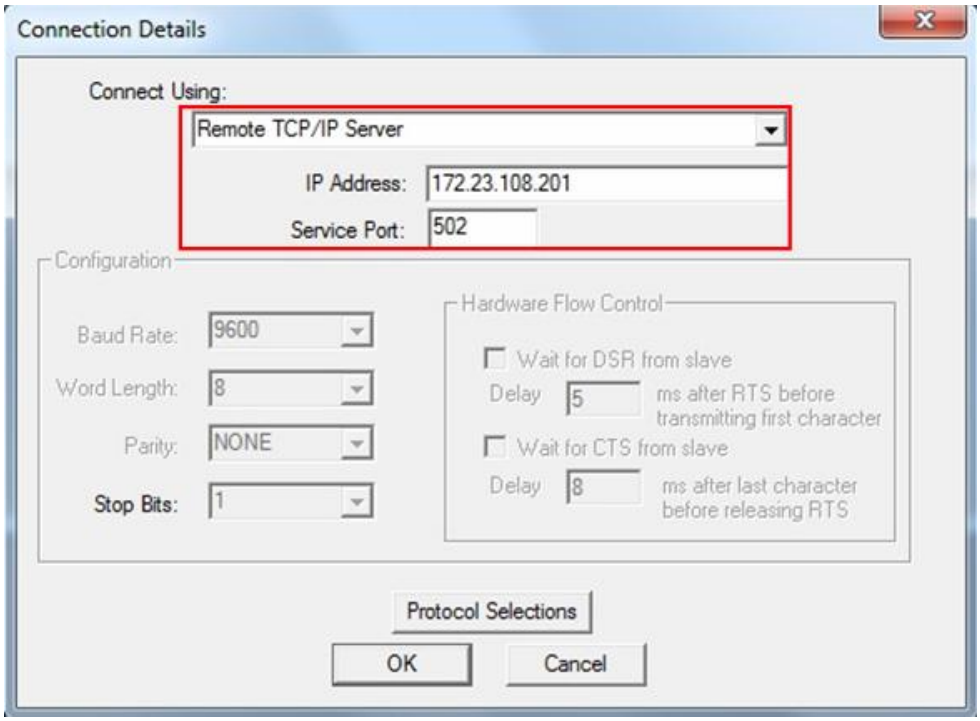


图 3-12 Connection 菜单下选择 connect

最后，在 Modscan32 的数据定义界面中设置数据类型为保持寄存器，并设置需要访问的 Modbus 起始地址及长度，建立与 CPU 集成 PN 口的通信连接，可以看到双方可以建立通信连接（变量 CONN_ESTABLISHED=TRUE 为已建立连接的状态），并进行数据读写，如图 3-13 所示：

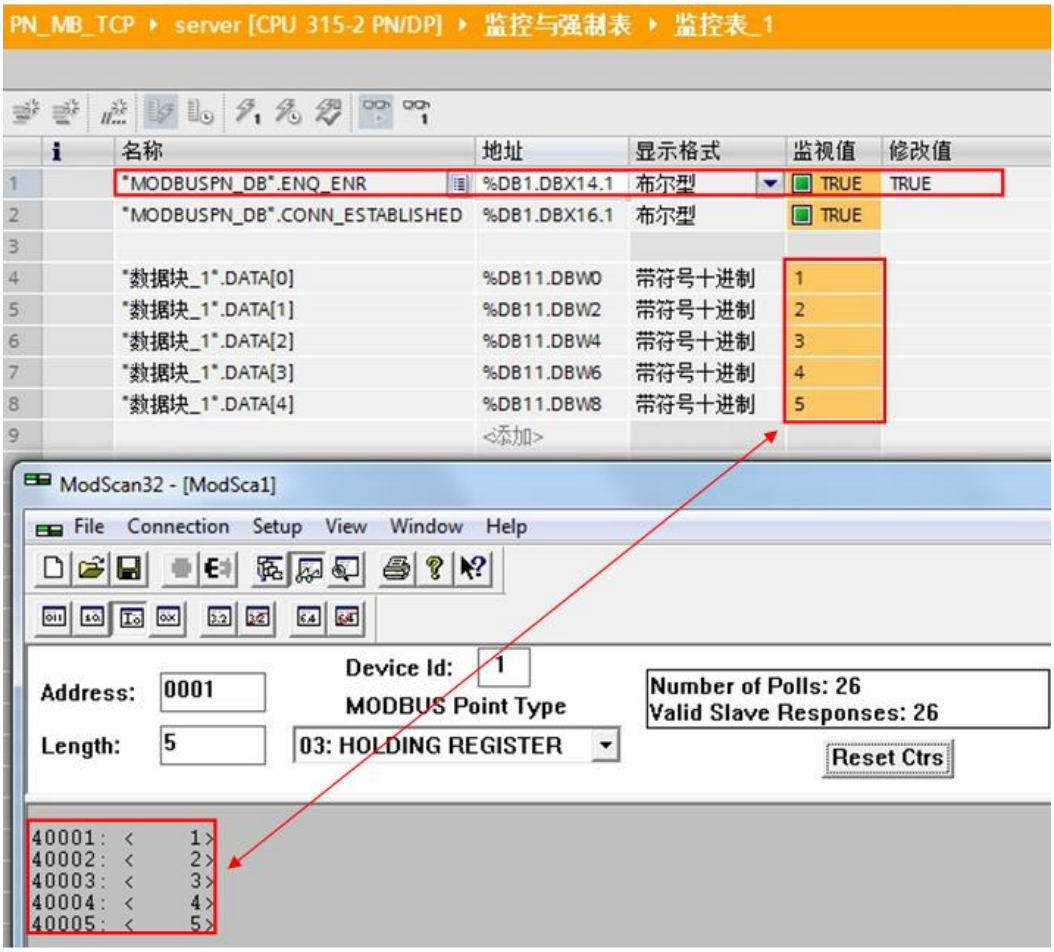


图 3-13 通信测试

使用功能块“MODBUSPN”的一些注意事项：

1）S7-300CPU 的集成 PN 口通过功能块“MODBUSPN”支持与多个 Modbus 客户端的通信，支持的个数取决于 CPU 所支持的 TCP 连接数，必须为每一个客户端连接分别调用一次功能块“MODBUSPN”，其背景数据块、ID、端口号等参数必须唯一。

2）S7-300CPU 的集成 PN 口可以同时作为 Modbus TCP 的 Server 及 Client。

3）S7-300CPU 的集成 PN 口支持多协议，除了运行 Modbus TCP 协议外，同时可以运行 PROFINET、TCP/IP、S7 等协议。

4 配置 PN CPU 作为 Modbus TCP Client 与通信伙伴建立通讯

下面以 S7-300 单站系统及 ModSim32 软件为例，详细介绍如何将 S7-300 单站系统 CPU 的集成 PN 口配置为 Client，ModSim32 为 Server 进行 Modbus TCP 通讯。

4.1 组态硬件

在章节 3.1 中创建的项目中，添加一个新的控制器，选择“添加新设备”——>“控制器”，选择正确的 CPU 型号，设备名称“client”，如图 4-1 所示：

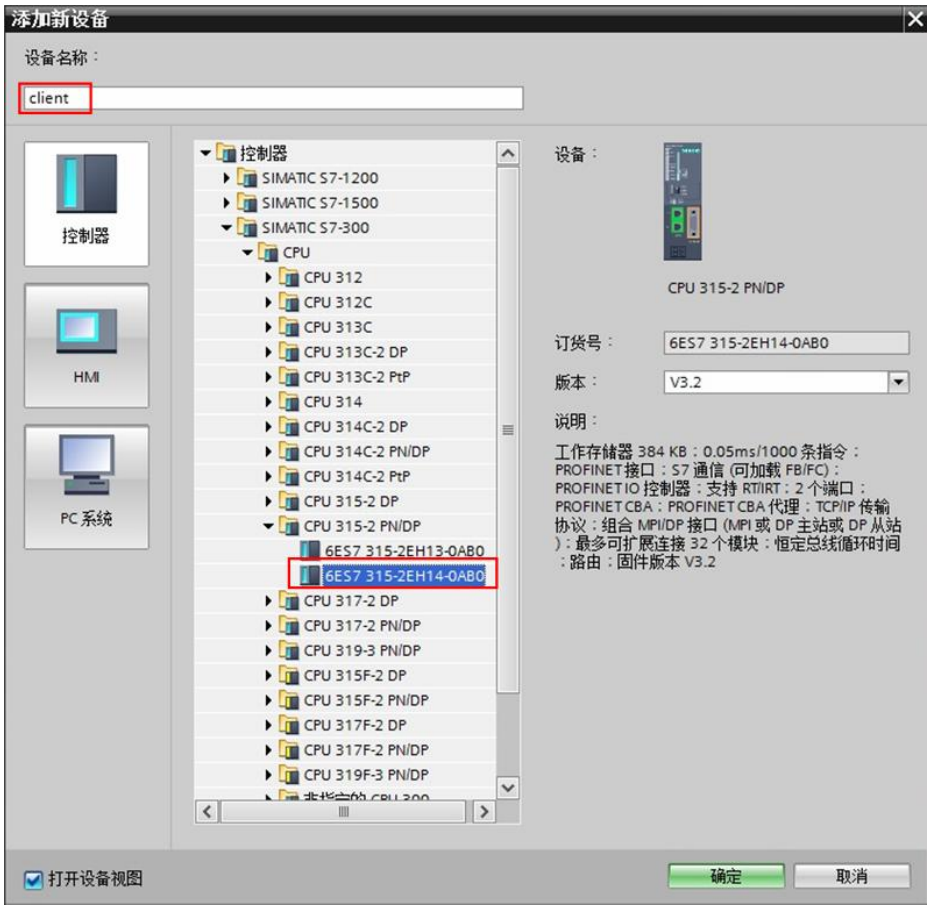


图 4-1 添加新设备

接着，在“设备视图”中，选择 CPU 以太网口，设置 IP 地址，如图 4-2 所示：

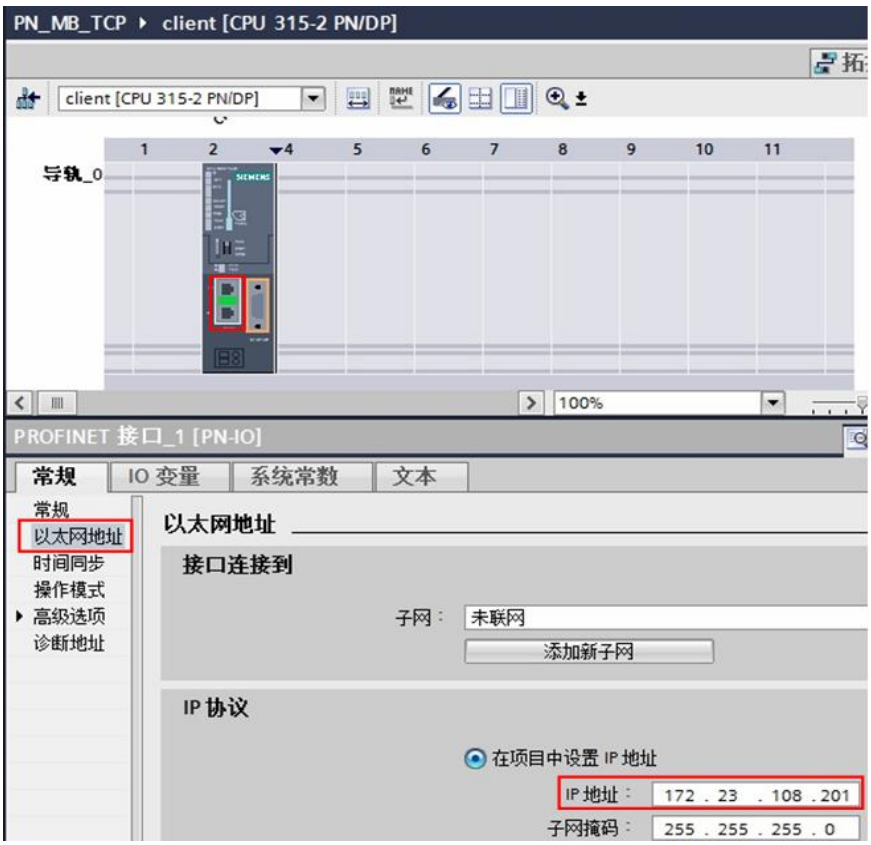


图 4-2 设置 IP 地址

4.2 编程

（1）OB1 调用 Modbus TCP 指令

在项目的 OB1 组织块中调用 Modbus TCP 指令，如图 4-3 所示：

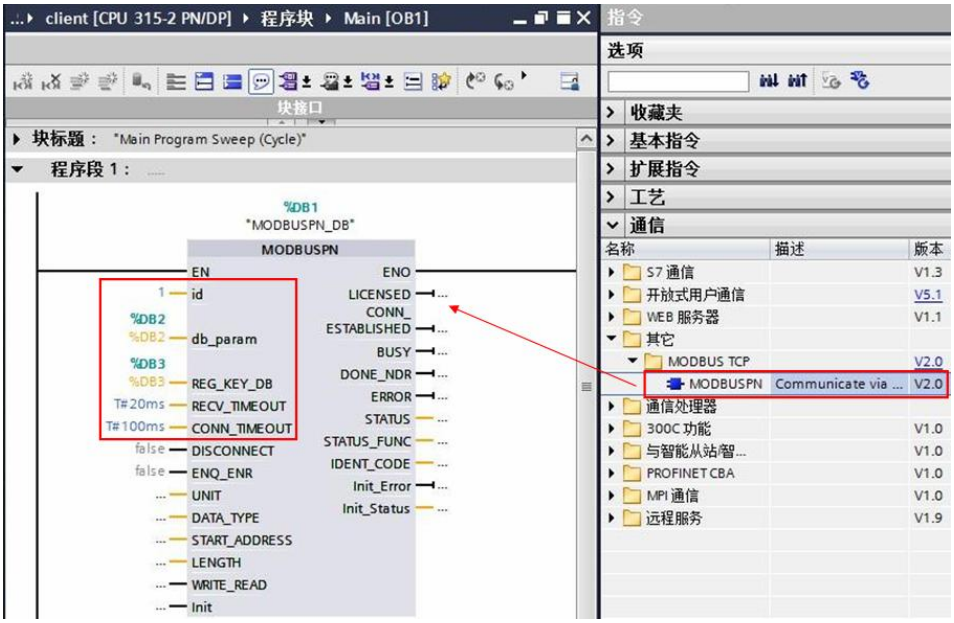


图 4-3 调用 Modbus TCP 指令

部分管脚说明（其它管脚信息请查看在线帮助）：

id：连接 ID 必须与参数 DB 中相关的 id 参数相同。

db_param：参数 DB 的编号，包含此 modbus 块实例的连接参数和 modbus 数据参数。CPU 决定该参数的取值范围。DB 编号 0 为系统保留，不允许使用。以纯文本格式输入 DB 编号“DBxy”。

REG_KEY_DB：具有可用于授权的注册表项的数据块。

RECV_TIMEOUT：对从耦合伙伴接收数据进行监视。超出监视时间后，将发出错误信号并终止连接。 最小值为 20 ms。

在“S7 为服务器”模式下将 RECV_TIMEOUT 设置为 < 20 ms，则使用默认值 1.2 s。RECV_TIMEOUT 监视 TCP 流的运行系统。不考虑各个客户端请求之间的中断。

CONN_TIMEOUT：监视调用建立或终止所用的时间。如果在组态的监视时间内无法成功建立或终止连接，则会在输出 STATUS 中显示相应的错误消息。最小值为 100 ms。

在“S7 为服务器”操作模式下，如果将 CONN_TIMEOUT 设置为 < 100 ms，则会使用默认值 5 s。

（2）创建参数数据块

创建数据块 DB2（名称 DB_param），选择类型为“MB_PN_PARAM”，如图 4-4 所示：

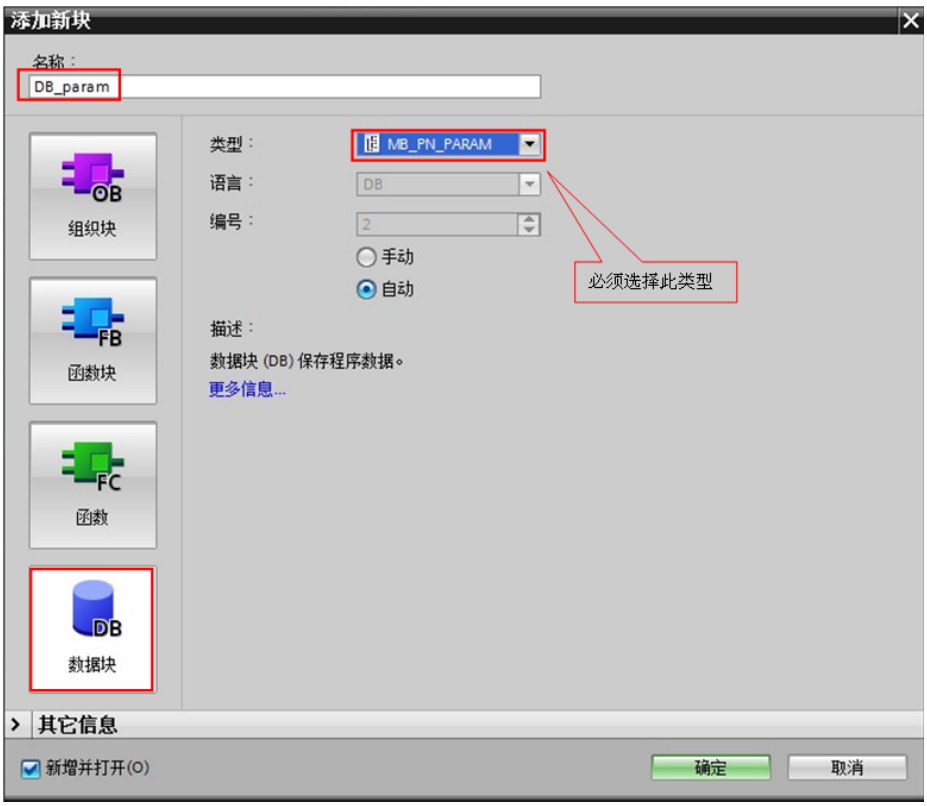


图 4-4 创建参数数据块

打开参数数据块，展开结构变量“Connection_settings”，并按下图参数设置，如图 4-5 所示：

PN_MB_TCP ▸ client [CPU 315-2 PN/DP] ▸ 程序块 ▸ DB_param [DB2]

保持实际值 快照 将快照值复制到起始

DB_param

		名称	数据类型	偏移量	起始值
1		Static			
2		Connection settings	Struct	0.0	
3		block_length	Word	0.0	W#16#0040
4		id	Word	2.0	16#1
5		connection_type	Byte	4.0	16#11
6		active_est	Bool	5.0	true
7		local_device_id	Byte	6.0	16#2
8		local_tsap_id_len	Byte	7.0	16#0
9		rem_subnet_id_len	Byte	8.0	16#0
10		rem_staddr_len	Byte	9.0	16#4
11		rem_tsap_id_len	Byte	10.0	16#2
12		next_staddr_len	Byte	11.0	16#0
13		local_tsap_id	Array[1..16] of Byte	12.0	
14		rem_subnet_id	Array[1..6] of Byte	28.0	
15		rem_staddr	Array[1..6] of Byte	34.0	
16		rem_staddr[1]	Byte	34.0	16#AC
17		rem_staddr[2]	Byte	35.0	16#17
18		rem_staddr[3]	Byte	36.0	16#6C
19		rem_staddr[4]	Byte	37.0	16#F5
20		rem_staddr[5]	Byte	38.0	16#0
21		rem_staddr[6]	Byte	39.0	16#0
22		rem_tsap_id	Array[1..16] of Byte	40.0	
23		rem_tsap_id[1]	Byte	40.0	16#01
24		rem_tsap_id[2]	Byte	41.0	16#F6
25		rem_tsap_id[3]	Byte	42.0	16#0

图 4-5 Connection_settings 参数

以下为部分参数说明（其它参数信息请查看在线帮助）：

id：每个 PN CPU 与通信伙伴之间的连接都需要一个连接 ID。如果有多个通信伙伴，则每个逻辑连接会使用不同的连接 ID。该连接 ID 在参数数据块中包含的“连接参数块”中组态。连接 ID 唯一地描述 CPU 与链接伙伴之间的连接，取值范围为 1 到 4095。必须在此处输入参数块中的连接 ID；该 ID 在整个 CPU 中必须唯一。

connection_type：建立连接的连接类型通过 TCON 指令定义。CPU 决定必须要设置的值。

TCP（兼容模式）：B#16#01，针对 CPU 315 或 317 <= FW V2.3 。

TCP：B#16#11，针对 CPU 315 或 317 >= FW V2.4、IM 151-8 PN/DP CPU、CPU314C、CPU319、CPU412、CPU414 和 CPU416。

该信息可能因固件不同而有所不同。

active_est：该参数表示连接建立类型，主动或被动。Modbus 客户端负责建立主动连接而 Modbus 服务器负责建立被动连接。

主动连接的建立：TRUE

被动连接的建立：FALSE

local_device_id：定义所用 PN CPU 的 IE 接口。根据不同的 PN CPU 类型，需要不同的设置。

CPU 类型	local_device_id
IM 151-8 PN/DP CPU	B#16#1
CPU 314C、315 或 317	B#16#2
CPU 319	B#16#3
CPU 412、414 或 CPU 416	B#16#5

rem_ staddr_len：指定 rem_staddr 参数的长度，该参数为通信伙伴的 IP 地址。如果要通过未指定的连接进行通信，则不为伙伴指定 IP 地址。

未指定的连接：B#16#0

指定的连接：B#16#4

rem_tsap_id_len：参数 rem_tsap_id 的长度和远程通信伙伴的端口号。

主动连接的建立：2

被动连接建立：0

rem_ staddr：在此字节数组中输入远程通信伙伴的 IP 地址。使用未指定的连接时，不输入 IP 地址。表示类型取决于 connection_type 参数。示例：IP 地址 192.168.0.1：

对于 connection_type B#16#01：	
rem_staddr[1] =	B#16#01 (1)
rem_staddr[2] =	B#16#00 (0)
rem_staddr[3] =	B#16#A8 (168)

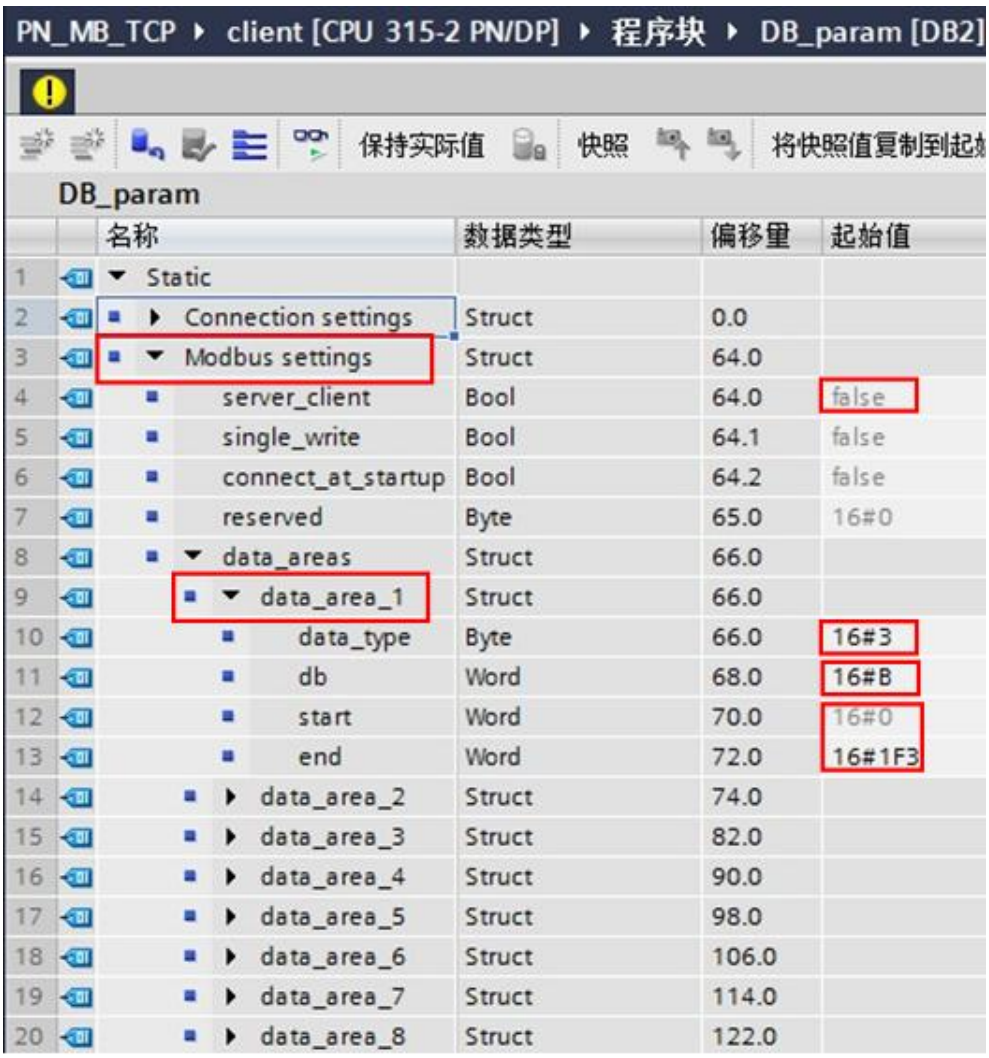
rem_staddr[4] = rem_staddr[5-6]=	B#16#C0 (192) B#16#00（保留）
对于 connection_type B#16#11： rem_staddr[1] = rem_staddr[2] = rem_staddr[3] = rem_staddr[4] = rem_staddr[5-6]=	 B#16#C0 (192) B#16#A8 (168) B#16#00 (0) B#16#01 (1) B#16#00（保留）

rem_tsap_id：使用该参数设置 remote 端口号。表示类型会因 connection_type 参数不同而有所不同。CPU 决定值范围。

对于 connection_type B#16#01： rem_tsap_id[1] rem_tsap_id[2] rem_tsap_id[3-16]	 用十六进制格式表示的端口号 low byte 用十六进制格式表示的端口号 high byte B#16#00
对于 connection_type B#16#11： rem_tsap_id[1] rem_tsap_id[2] rem_tsap_id[3-16]	 用十六进制格式表示的端口号 high byte 用十六进制格式表示的端口号 low byte B#16#00

本例中，CPU 为 315-2PN，connection_type B#16#11，远程伙伴的 IP 地址为：172.23.108.245（16#AC，16#17，16#6C，16#F5），端口号设置为 502（16#01F6），则对应于 rem_tsap_id[1]= 16#01，rem_tsap_id[2]= 16#F6。

在参数数据块中，继续展开结构变量“Modbus_settings”，并按下图参数设置，如图 4-6 所示：



	名称	数据类型	偏移量	起始值
1	Static			
2	Connection settings	Struct	0.0	
3	Modbus settings	Struct	64.0	
4	server_client	Bool	64.0	false
5	single_write	Bool	64.1	false
6	connect_at_startup	Bool	64.2	false
7	reserved	Byte	65.0	16#0
8	data_areas	Struct	66.0	
9	data_area_1	Struct	66.0	
10	data_type	Byte	66.0	16#3
11	db	Word	68.0	16#B
12	start	Word	70.0	16#0
13	end	Word	72.0	16#1F3
14	data_area_2	Struct	74.0	
15	data_area_3	Struct	82.0	
16	data_area_4	Struct	90.0	
17	data_area_5	Struct	98.0	
18	data_area_6	Struct	106.0	
19	data_area_7	Struct	114.0	
20	data_area_8	Struct	122.0	

图 4-6 Modbus_settings 参数

server_client：S7 是服务器=TRUE；S7 是客户端=FALSE。

data_areas： S7 存储器中有八个可以用于映射 MODBUS 地址的数据区。必须至少定义第一个数据区，其余七个数据区可选择性定义。根据作业类型，将从数据区读取数据或向其中写入数据。

任何作业都只能从一个 DB 读取数据或向一个 DB 写入数据。访问寄存器或位于多个 DB 中的位值时，即使编号连续无间隔，也将分为两个作业。组态时请务必注意。

一个数据块中可以映射的 Modbus 区（寄存器或位值）数目比一个消息帧可以处理的数目多。

data_type：指定该数据块中映射的 MODBUS 数据类型。如果在 data_type 中输入值 16#0，则不使用相应的区域。

标识符	数据类型	数据宽度
16#0	未使用区域	
16#1	线圈	Bit
16#2	输入	Bit
16#3	保持寄存器	Word
16#4	输入寄存器	Word

db：指定映射 MODBUS 寄存器或下面定义的位值的数据块。DB 编号 0 为系统保留，不允许使用。

DB 编号：1 到 65535（W#16#0001 到 W#16#FFFF）。

start / end：start 指定 DB 的数据字 0 中映射的第一个 Modbus 地址。end 参数定义最后一个 MODBUS 地址。

对于寄存器访问，带有最后一个 Modbus 地址输入的 S7 DB 中的数据字编号如下计算：DBW 编号 = (end – start) * 2

对于位访问，带有最后一个 Modbus 地址输入的 S7 DB 中的数据字节编号如下计算：DBB 编号 = (end – start + 7) / 8

定义的数据区不得重叠。end 参数不得小于 start。如果发生错误，指令启动将中止并提示错误。如果两个值相同，则将分配一个 Modbus 地址（1 个寄存器或 1 个位值）。

注意：数据块必须比已组态数据所需的长度多两个字节。最后的两个字节供内部使用。

（3）创建授权密钥数据块和编程错误组织块

创建授权密钥数据块 DB3（名称 REG_KEY_DB），选择类型为“全局 DB”，打开该 DB 块，创建变量“REG_KEY”，数据类型为“String[17]”，如图 4-7 所示：

图 4-7 创建授权密钥数据块

授权密钥的获取方法，请查看“MODBUSPN”指令的在线帮助，主题为“**使用参数 IDENT_CODE 和 REG_KEY_DB 进行授权**”的部分。

由于在获取授权密钥前，“MODBUSPN”指令是无授权状态，会使 CPU 报错而停机。而为了读取 CPU 的“IDENT_CODE”码，需要 CPU 运行起来，则必须添加编程错误组织块 OB121，如图 4-8 所示：

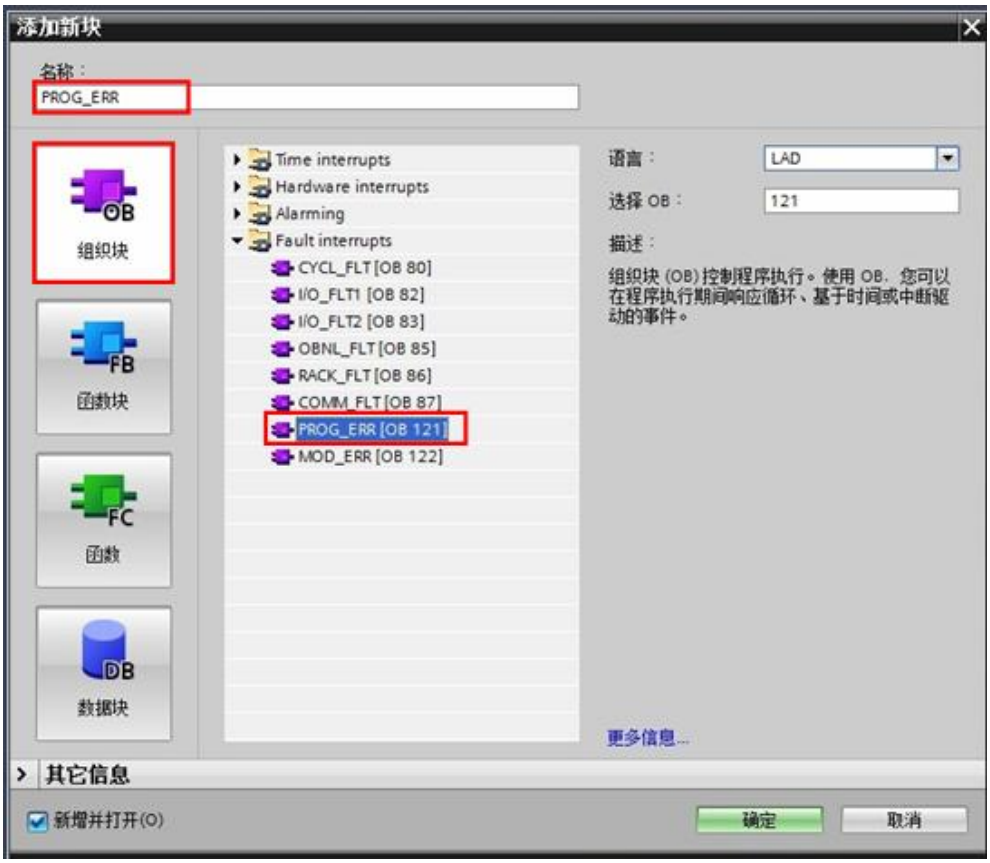


图 4-8 添加编程错误组织块 OB121

（4）创建启动组织块

创建启动组织块 OB100（名称 COMPLETE RESTART），语言为“STL”，打开该 OB 块，置位初始化位“MODBUSPN_DB”.init。如图 4-9 所示：

Init：在参数中有上升沿时，初始化 Modbus 块。只有当前没有作业正在运行时，才能执行初始化。必须通过 ENQ_ENR = FALSE 和 BUSY = FALSE 在程序中确保此条件。

图 4-9 创建启动组织块

（5）创建全局数据块

创建全局数据块 DB11，用于关联 modbus 寄存器地址 40001~40500，如图 4-10 所示：

图 4-10 创建全局数据块

4.3 通信测试

完成上述操作后，下载项目到 CPU 中，打开 ModSim32 应用程序，下面以保持寄存器为例介绍通信测试过程。

首先，在 ModSim32 的 Connection 菜单下选择 Connect——>Modbus/TCP Svr，并设置 ModSim32 作为 server 端的端口号，如图 4-11 所示：

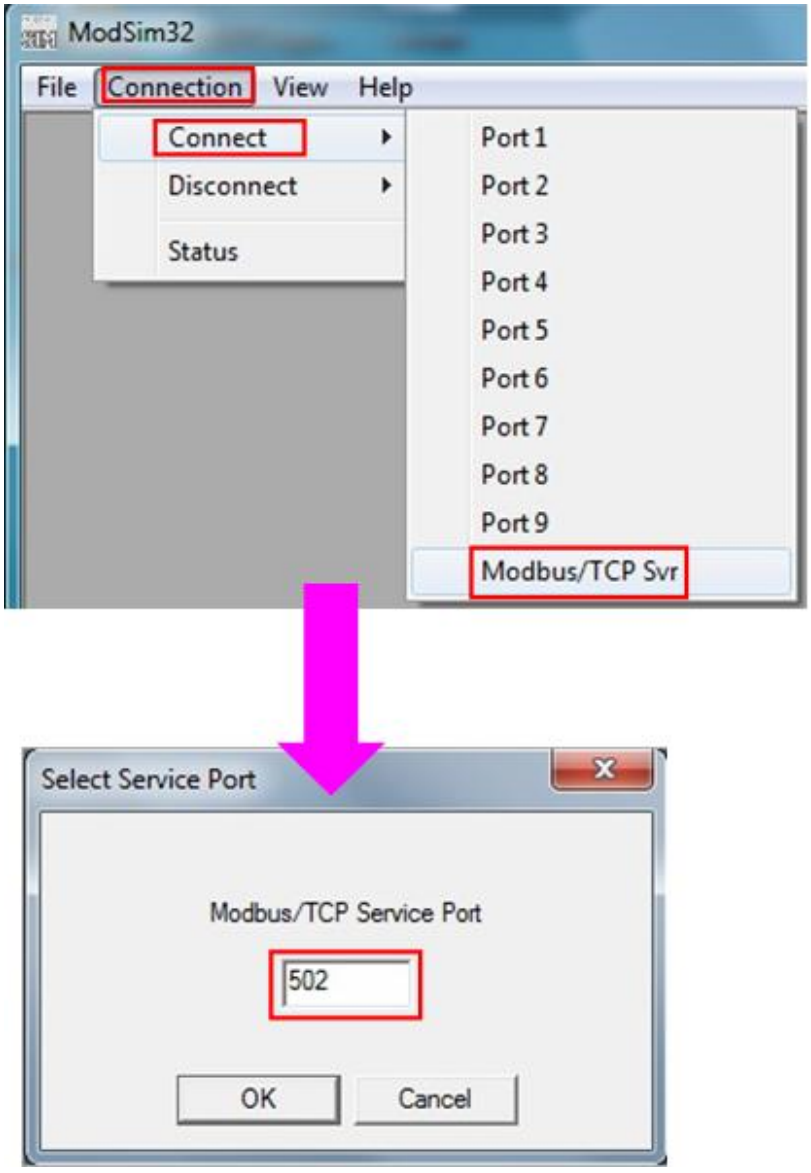


图 4-11 设置 ModSim32 作为 server 端的端口号

然后，在 PLC 监控表中设置 UNIT，DATA_TYPE，START_ADDRESS，LENGTH 等参数。

最后，需要置位“ENQ_ENR”，作为 modbus TCP 的 client 端的 PLC 将创建连接，并发送读取寄存器的请求，可以看到双方可以建立通信连接（变量

CONN_ESTABLISHED= TRUE 为已建立连接的状态），并进行数据读写，如图 4-12 所示：

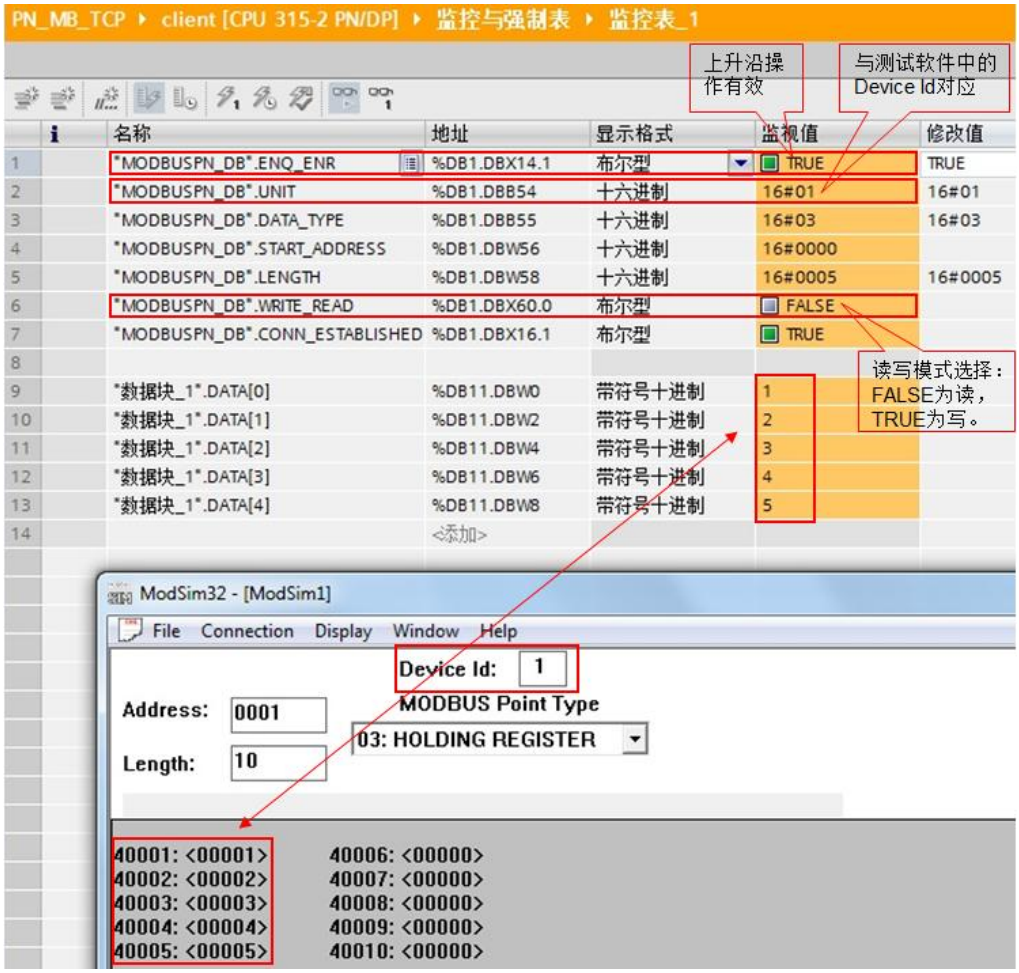


图 4-12 通信测试

使用功能块“MODBUSPN”的一些注意事项：

1）S7-300CPU 的集成 PN 口通过功能块“MODBUSPN”支持与多个 Modbus 服务器的通信，支持的个数取决于 CPU 所支持的 TCP 连接数，必须为每一个服务器连接分别调用一次功能块“MODBUSPN”，其背景数据块、ID 必须唯一，必须指定唯一的服务器 IP 地址。

2）S7-300CPU 的集成 PN 口可以同时作为 Modbus TCP 的 Server 及 Client。

3）S7-300CPU 的集成 PN 口支持多协议，除了运行 Modbus TCP 协议外，同时可以运行 PROFINET、TCP/IP、S7 等协议。