

<sup>操作指南 • 9/2017</sup> 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 Unrestricted

目录

1	Modbu	ɪs TCP 通讯概述	3
	1.1	通讯所使用的以太网参考模型	3
	1.2	Modbus TCP 数据帧	3
	1.3	Modbus TCP 使用的端口号	4
2	S7-300	//400 集成 PN □ Modbus TCP 通讯概述	5
3	配置 PI	N CPU 作为 Modbus TCP Server 与通信伙伴建立通讯	6
	3.1	组态硬件	6
	3.2	编程	8
	3.3	通信测试	17
4	配置 PI	N CPU 作为 Modbus TCP Client 与通信伙伴建立通讯	19
	4.1	组态硬件	19
	4.2	编程	20
	4.3	通信测试	31

## 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。

Copyright © Siemens AG Copyright year All rights reserved





Modbus TCP/IP ADU

(This information is embedded into the data portion of the TCP frame)

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。

Copyright © Siemens AG Copyright year All rights reserved 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:

指令		
选项		
	4 Mit 😳 🖏	
> 收藏夹		
> 基本指令		
> 扩展指令		
> 工艺		
✔ 通信		
名称	描述	版本
🕨 🛅 S7 通信		V1.3
一 开放式用户通信		<u>V5.1</u>
▶ 🚞 WEB 服务器		V1.1
▼ 🛄 其它		
MODBUS TCP		<u>V2.0</u>
MODBUSPN	Communicate via PROFINET as Modbus TCP server or client	V2.0
▶ 🗀 通信处理器		

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

# 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:

创建新项目	×
项目名称:	PN_MB_TCP
路径:	C:\Users\ xxx \Desktop
版本:	V14 SP1
作者:	III
注释:	~
	创建取消

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

- PN\_MB\_TCP → server [CPU 315-2 PN/DP] 書事 server [CPU 315-2 PN/DP] 💌 🖽 📰 🕰 🖽 🛄 🍳 ± 2 7 8 9 10 11 -4 5 6 导乳 0 11 11 11 Ing . < III > 100% -常规 10 变量 系统常数 文本 常规 以太网地址 以太网地址 时间同步 接口连接到 操作模式 ▶ 高级选项 子网: 未联网 诊断地址 添加新子网 IP协议 ● 在项目中设置 IP 地址 IP 地址: 172 . 23 . 108 . 201 子网摘码: 255.255.255.0
- 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:

添加新块 <sub>名称:</sub> DB_param				×
<b>OB</b> 组织块	类型 : 语言 : 编号 :	[] MB_PN_PARAM DB 2 ○ 手动		
FB 函数块	描述: 数据块 (DB) 保存 更多信息	<ul> <li>自动</li> <li>程序数据。</li> </ul>	必须选择此类型	
FC 函数				
し サウトウ				
<ul> <li>▶   其它信息</li> <li>▼ 新增并打开(0)</li> </ul>			确定	取消

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:

PN	_M	B_T	СР	server [CPU 315-	2 PN/DP] > 程序	<b>块 ▶</b> DE	3_param [DB2]
ill.	1	2	5	🛃 🚬 😤 保持实际	值 🔒 快照 🎮	鸣 将	快照值复制到起始
	DB	pa	rar	n		1	
		名利	尔		数据类型	偏移童	起始值
1	-	-	Sta	tic			
2	-		•	Connection settings	Struct	0.0	
3	-00	1 2		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	-00			local_tsap_id_len	Byte	7.0	16#2
9	-0			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	-00		•	next_staddr_len	Byte	11.0	16#0
13	-		•	<ul> <li>local_tsap_id</li> </ul>	Array[116] 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]	用十六进制格式表示的端口号 low byte
local_tsap_id[2]	用十六进制格式表示的端口号 high byte
local_tsap_id[3-16]	B#16#00
对于 connection_type B#16#11:	
对于 connection_type B#16#11: local_tsap_id[1]	用十六进制格式表示的端口号 high byte
对于 connection_type B#16#11: local_tsap_id[1] local_tsap_id[2]	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

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:

PN	_MB_	TCP 🕨 s	erver [CPU 315	-2 PN/DP] > 程序	块 ▶ DI	B_param [DB2]
	D					
1	ن ا ناچ نا		😑 ° 保持实际	病值 🔒 快照 🧠	風将	快照值复制到起始
	DB n	aram				
	名	称		数据类型	偏移軍	記始值
1		Static				
2	-	► Conr	nection settings	Struct	0.0	
3	-	▼ Mod	bus settings	Struct	64.0	
4	-00	= s	erver_client	Bool	64.0	true
5	-0	• s	ingle_write	Bool	64.1	false
6	-	• c	onnect_at_startup	Bool	64.2	false
7	-	= re	eserved	Byte	65.0	16#0
8	-00	• • d	ata_areas	Struct	66.0	
9	-		data_area_1	Struct	66.0	
10	-00		data_type	Byte	66.0	16#3
11	-		db	Word	68.0	16#B
12	-		start	Word	70.0	16#0
13	-0	-	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	-0		data_area_6	Struct	106.0	
19	-0		data_area_7	Struct	114.0	
20	-0		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:

opyright © Siemens	AG Copyright year	All rights reserved
Copy	AG	P

加新块			_		×
名称: REG_KEY_DB					
	类型:	🧧 全局 D8	•		
OB	语言:	DB	Ŧ		
组织块	编号:	3	÷		
		○手动			
		<ul> <li>自动</li> </ul>			
FB	描述:	and an an an			
函数块	数据块 (DB) 保 更多信息	存柱序数据。			
-FC					
DB					
数据块					
其它信息					
新增并打开(0)				确定	取消
MB TCP ▸	server [CPU 3	15-2 PN/DP] ▶ ≸	呈序块 → RE(	G KEY DB [DI	B31
			ita ita unu		
" 🖭 🛰 🕏	► 〒 保持	头际值 🔐 快照	う ち 将り	快照值复制到起始	加且月
REG_KEY_D	В	dit. In als wi	Interior pro-	43.44.7#	必须创建为
475		额据类型	偏移重	起始值	
名称					/ 字符的字符

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:

PROG_ERR	]		
●のB 組织块 回数块 回数块	<ul> <li>Time interrupts</li> <li>Hardware interrupts</li> <li>Alarming</li> <li>Fault interrupts</li> <li>CYCL_FLT[OB 80]</li> <li>I/O_FLT1 [OB 82]</li> <li>I/O_FLT2 [OB 83]</li> <li>OBNL_FLT [OB 85]</li> <li>RACK_FLT [OB 86]</li> <li>COMM FLT [OB 87]</li> <li>FROG_ERR [OB 121]</li> <li>MOD_ERR [OB 122]</li> </ul>	语言: 选择 OB: 描述: 组织诀(OB)控 在程序执行期间 动的事件。	LAD 121 制程序执行。使用 OB. 您可 响应循环、基于时间或中断
して 数据块		更多信息	
其它信息		E>ina-	

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。

添加新块			×
名称: COMPLETE RESTART			
<b>また。</b> 通叙块 函数块 の数块 正教	<ul> <li>Time interrupts</li> <li>Hardware interrupts</li> <li>Startup</li> <li>COMPLETE RESTART [OB 100]</li> <li>Alarming</li> <li>Fault interrupts</li> </ul>	语言: 51L 选择 OB: 100 描述: 组织块(OB)控制程序执行。使用 在程序执行期间响应循环、基于时 动的事件。	● 第一以 前或中断驱
<b>DB</b> 数据块		更多信息	
> 其它信息			
☑ 新增并打开(0)		确定	取消
PN_MB_TCP → ser	ver [CPU 315-2 PN/DP] ▶ 程序块 ± 🔍 📰 🔚 🚍 💬 觷± 塁±	▶ COMPLETE RESTART [OB100] 5 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1	■ 初始化署
▶ 状标题: *Comp	olete Restart"		/ 位操作
▼ 在序段1: 5	tarts the init		
1 SI 2 S	"MODBUSPN_DB".Init	%DB1.DBX60.1 starts the in	it

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:

数据块\_1 [DB11] PN\_MB\_TCP > server [CPU 315-2 PN/DP] > 程序块 D 00 🥩 💣 🐛 🋃 🗮 保持实际值 調 调 将快照值复制到起始 快照 数据块\_1 数据类型 起始值 名称 偏移量 Static 1 -0 -DATA 2 Array[0..500] of Int 0.0 1 

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:

	Remote TCP/IP Server	<b>_</b>
	IP Address:	172.23.108.201
	Service Port:	502
inguration -		- Hardware Flow Control
laud Rate:	9600 👻	
ord Length:	8 👻	Delay 5 ms after RTS before
Parity:	NONE -	transmitting first character Wait for CTS from slave
Stop Bits:	1 💌	Delay 8 ms after last character before releasing RTS

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:

PN_MB_TCP	▶ clie	ent [CPl	J 315-2	PN/DP]							
											₽ 招
de client [C	PU 315-	2 PN/DP]	-				) 🔍 ±				
50 .	1	2	<b>▼</b> 4	5	6	7	8	9	10	11	
47 <b>64_</b> 5											
< III		<u>E8</u>			-		> 100%	6		•	
PROFINET 报 登报	:LL_1 0 恋音			4	¢★	1					
常规	以	太网地:	th.	x .	~~						
以太网地址 时间同步 操作模式 ▶ 高级选项 诊断地址	ŧ	<b>亲口连</b> 挂	<b>姜</b> 到		子网:	未联网	添加薪	子网			
		P协议				·	*****				
						• 在项	目中设置	IP 地址			
							子网摘	)) [26]: [	255 . 25	5 . 255	. 201

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:

> client [CPU 315-2 PN/DP] > 程序块 > Main [OB1] _ 『 = 』 = ×	指令
	选项
(3) (3) 글 글 👘 🔚 🚍 🖆 🗐 웹 ± 웹 ± 웹 ± ☱ 🎲 💪 🕐 🕫 , 📑	SF 🔂 101 101
块接口	> 收藏夹
▶ 块标题: "Main Program Sweep (Cycle)" ^	> 基本指令
▼ 程序段1:	> 扩展指令
8/00.1	> 工艺
"MODBUSPN_DB"	✓ 通信
MODBUSPN	名称 描述 版本
EN ENO	▶ 🛄 S7 通信 V1.3
1 — id LICENSED —	▶ → 开放式用户通信 <u>V5.1</u>
%DB2 CONN_	▶ 🛄 WEB 服务器 V1.1
%DB2 — db_param	▼ 🛄 其它
%DB3 DONE NOT	MODBUS TCP V2.0
%DB3 - REG_KEY_DB	MODBUSPN Communicate via V2.0
T#20ms RECV_TIMEOUT	▶ 🛄 通信处理器
T#100ms - CONN_TIMEOUT	▶ 🔄 300C功能 V1.0
false - DISCONNECT	▶ 🛄 与智能从站 智 V1.0
false - ENQ_ENR	PROFINET CBA V1.0
	▶ 🛄 MPI通信 V1.0
DATA_TYPE	▶ 🛄 远程服务 V1.9
START_ADDRESS	
LENGTH	
··· - WRITE_READ	
<b>—</b> Init	

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:

添加新块				×
名称:				
Db_param				
	类型:	🖪 MB_PN_PARAM		
OB	语言:	DB	$\wedge$	
组织块	编号:	2		
		○手动	心态建地来和	
		● 自动	见则远律此突空	
FB	描述:			
函数块	数据块 (DB) 保存程	序数据。		
	更多信息			
12000				
<b>■</b> DB				
数据块				
> 其它信息				
✓ 新增开打开(O)			确定	

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	l_ME	3_TCP	••	client [CPU 315-2	2 PN/DP] > 程序均	R, ▶ DB_	param [DB2]
	Ð						
ill'	* =*		₽,	🚬 🙄 保持实际	值 🔒 快照 🛰	鸟 将	央照值复制到起始
	DB	para	m	- 1 - 1			
		名称			数据类型	偏移重	起始值
1	-	• St	atic				
2	-		Con	nection settings	Struct	0.0	
З	-0		1	block_length	Word	0.0	W#16#0040
4			i	id	Word	2.0	16#1
5	-		(	connection_type	Byte	4.0	16#11
6	-			active_est	Bool	5.0	true
7			1	local_device_id	Byte	6.0	16#2
8			1	local_tsap_id_len	Byte	7.0	16#0
9	-		1	rem_subnet_id_len	Byte	8.0	16#0
10			1	rem_staddr_len	Byte	9.0	16#4
11			1	rem_tsap_id_len	Byte	10.0	16#2
12	-		1	next_staddr_len	Byte	11.0	16#0
13			• 1	local_tsap_id	Array[116] of Byte	12.0	
14			•	rem_subnet_id	Array[16] of Byte	28.0	
15	-		• 1	rem_staddr	Array[16] 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[116] 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)

0 (192)
0(保留)
CO (192)
.8 (168)
0 (0)
1 (1)
0(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]	用十六进制格式表示的端口号 low byte
rem_tsap_id[2]	用十六进制格式表示的端口号 high byte
rem_tsap_id[3-16]	B#16#00
对于 connection_type B#16#11:	
对于 connection_type B#16#11: rem_tsap_id[1]	用十六进制格式表示的端口号 high byte
对于 connection_type B#16#11: rem_tsap_id[1] rem_tsap_id[2]	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
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:

PN	_M	B_T	СР	▶ c	lient [CPU 315-	2 PN/DP] > 程序	块 ▶ DB_	param [DB2]		
	Ð									
HIL.	i di	¥ 🚺	5	2	🗄 🙄 保持实际	标值 🔒 快照 🏁	ト 門 将!	央照值复制到起如		
	DB	pa	ra	m				• [		
		名利	尔			数据类型	偏移重	起始值		
1	-	•	Sta	atic						
2	-00		•	Conn	ection settings	Struct	0.0			
3	-		•	Modb	us settings	Struct	64.0			
4	-			se	erver_client	Bool	64.0	false		
5	-00			si	ngle_write	Bool	64.1	false		
6	-			co	onnect_at_startup	Bool	64.2	false		
7	-			re	served	Byte	65.0	16#0		
8	-00		•	▼ da	ata_areas	Struct	66.0			
9	-			• •	data_area_1	Struct	66.0			
10	-				data_type	Byte	66.0	16#3		
11	-0				db	Word	68.0	16#B		
12	-				start	Word	70.0	16#0		
13	-0				end	Word	72.0	16#1F3		
14	-				data_area_2	Struct	74.0			
15	-				data_area_3	Struct	82.0			
16	-00				data_area_4	Struct	90.0			
17	-				data_area_5	Struct	98.0			
18	-00				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:

	类型:	📔 全局 DB			
	语言:	DB	*		
组织种	编号:	3	<b>(</b>		
44000		○ 手动			
		() 自动			
FB	描述:				
函数块	数据块 (DB) {	<b>保存程序数据</b> 。			
	更多信息				
-					
FC					
函数					
DB					
数据块					
1它信自					
新增并打开(0)				确定	取消
				· · · · · · · · · · · · · · · · · · ·	
MB_TCP ▶	client [CPU 3	15-2 PN/DP] > 程	『序块 ・ REG	_KEY_DB [DB3	3]
	1		ka ka usu		
			0.5, 0.5, 12,0	光照信复制到纪始	1自
🥐 🔩 🛃	E % 保	时头际值 📲 伏照	T + 191	STOLEN PERFORMANCE	

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:

一下に 函数				
	- MOD_ERR [OB 122]			
函数块	COMM_FLT[OB 87]			
FB	SACK FLT[OB 85]			
	I/O_FLT1 [OB 82] I/O_FLT2 [OB 83]	在程序执行期间动的事件。	响应循环、基于时间	國中断到
组织块	<ul> <li>Fault interrupts</li> <li>CYCL_FLT [OB 80]</li> </ul>	描述: 组织块 (OB) 控	制程席执行。 <b>使用</b> O	8. 余司い
OB	Alarming	选择 OB :	121	
1.000	Time interrupts	语言:	LAD	

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。

添加新块				×
名称: COMPLETE RESTART				
<b>この</b> 通知快 通数快 函数块	Time interrupts     Hardware interrupts     Startup     COMPLETE RESTART [O     Alarming     Fault interrupts	语言: 选择 OB: 描述: 组织块 (O 在程序执行 动的事件。	5元 100 8) 控制程序执行。使用( 7期间响应循环、基于时 。	▼ 08. 您可以 间或中睡醒
一 正 正 正 正 正 正 正 正 正 正 正 正 正		更多信息。		
> 其它信息				
☑ 新增并打开(0)			确定	取消
PN_MB_TCP ▶ clie ⊮রী⊮র ≇ 🗐 📑	nt [CPU 315-2 PN/DP] > 程序 生 🔩 🗄 🚍 🚍 💬 冶 ± .	家块 → COMPLETE RES 當生 醫士 ☰ 診 ( <sup>©</sup> )	TART [OB100] Go 信 强 V G= 块接口	1
▶ 块标题: *Comp	lete Restart"		1 - 1 - 1 - 1	初始化置
▼ 程序段 1: s	tarts the init		/	1以採作
1 SI 2 S	ET "MODBUSPN_DB".Init	%DB1.DBX60.1	starts the init	

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:

PN	N_ME	8_TCP → client [CPU	315-2 PN/DP]	<ul> <li>程序块</li> </ul>	い 数据	块_1 [DB11]
101	کے ڈ		保持实际值 🔒	快照 🔤	鸣,将快	照值复制到起始
	£U1	<b>告吠_1</b> 名称	数据类型	<u>1</u>	偏移量	起始值
1	<b>0</b>	<ul> <li>Static</li> <li>DATA</li> </ul>	Array[0	500] of Int	0.0	

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:

File Connection View Help Connect Disconnect Status Port 3 Port 4 Port 5 Port 6 Port 7 Port 8 Port 9 Modbus/TCP Svr	TEI ModSim32	
Connect Port 1   Disconnect Port 2   Port 3 Port 4   Port 5 Port 6   Port 7 Port 8   Port 9 Modbus/TCP Svr	File Connection View Help	
Disconnect Port 2 Port 3 Port 4 Port 5 Port 6 Port 7 Port 8 Port 9 Modbus/TCP Svr Select Service Port	Connect >	Port 1
Select Service Port	Disconnect +	Port 2
Port 4 Port 5 Port 6 Port 7 Port 8 Port 9 Modbus/TCP Svr	Status	Port 3
Port 5 Port 6 Port 7 Port 8 Port 9 Modbus/TCP Svr		Port 4
Port 6 Port 7 Port 8 Port 9 Modbus/TCP Svr		Port 5
Port 7 Port 8 Port 9 Modbus/TCP Svr		Port 6
Port 8 Port 9 Modbus/TCP Svr		Port 7
Port 9 Modbus/TCP Svr		Port 8
Modbus/TCP Svr Select Service Port		Port 9
Select Service Port	and the second se	Modbus/TCP Svr
	Select Service Port	×

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:

PN_I	MB_1	TCP + client [CPU 315-2 PN/DP] +	监控与强制表	・ 监控表_1		
				上升	十沿操 与:	测试软件中的
	-	12 10 91 92 19 19		11=1=		
1	i	名称	地址	显示格式	监视值	修改值
		*MODBUSPN_DB*.ENQ_ENR	%DB1.DBX14.1	布尔型		TRUE
		"MODBUSPN_DB".UNIT	%DB1.DBB54	十六进制	16#01	16#01
		*MODBUSPN_DB*.DATA_TYPE	%DB1.DBB55	十六进制	16#03	16#03
		*MODBUSPN_DB*.START_ADDRESS	%DB1.DBW56	十六进制	16#0000	
		"MODBUSPN_DB".LENGTH	%DB1.DBW58	十六进制	16#0005	16#0005
		*MODBUSPN_DB*.WRITE_READ	%DB1.DBX60.0	布尔型	FALSE	
		*MODBUSPN_DB*.CONN_ESTABLISHED	%DB1.DBX16.1	布尔型	TRUE	
					3	虚写模式选择
		*数据块_1*.DATA[0]	%DB11.DBW0	带符号十进制	1	FALSE为读,
0		*数据块_1*.DATA[1]	%DB11.DBW2	带符号十进制	2	TRUE为写。
1		*数据块_1*.DATA[2]	%DB11.DBW4	带符号十进制	3	
2		*数据块_1*.DATA[3]	%DB11.DBW6	带符号十进制	4	
3		*数据块_1*.DATA[4]	%DB11.DBW8	带符号十进制	5	
4			<添加>			
		ModSim32 - [ModSim1]	dow Help			
		Address: 0001 MC	vice Id: 1 DBUS Point Typ	be		
		Length: 10		<b>n</b>		
		400001:         40006:         0           40002:         40007:         0           40003:         40003:         40008:         0           40004:         40004:         40009:         0           40005:         40005:         40019:         0	0000> 0000> 0000> 0000> 0000> 0000>			

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。