SIEMENS

应用与工具 •3 月/2014 年

基于 S7-1500 CPU 集成 PN 口的 ModbusTCP 通信快速入门

S7-1500,MODBUS/TCP,Library

http://support.automation.siemens.com/CN/view/zh/90974593

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’s，I/O 模块，以及连接其它简单域总线或 I/O 模块的网关服务的.

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 址 IP 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(MBAP、Modbus Application Protocol、Modbus 应用协议)分 4 个域，共 7 个字节.

1.3 Modbus TCP 使用的通讯资源端口号

在 Moodbus 服务器中按缺省协议使用 Port 502 通信端口,在 Modus 客户器程序中设置任意通信端口，为避免与其他通讯协议的冲突一般建议 2000 开始可以使用.

1.4 Modbus TCP 使用的功能代码

按照使用的通途区分,共有 3 种类型分别为：

1) 公共功能代码：已定义好功能码，保证其唯一性，由 Modbus.org 认可；

2) 用户自定义功能代码有两组，分别为 65～72 和 100～110，无需认可，但不保证代码使用唯一性,如变为公共代码，需交 RFC 认可；

3) 保留功能代码，由某些公司使用某些传统设备代码，不可作为公共用途。

按照应用深浅，可分为 3 个类别

1) 类别 0,客户机/服务器最小可用子集：读多个保持寄存器(fc.3)；写多个保持寄存器(fc.16)。

2) 类别 1，可实现基本互易操作常用代码：读线圈(fc.1)；读开关量输入(fc.2)；读输入寄存器(fc.4)；写线圈(fc.5)；写单一寄存器(fc.6)。

3) 类别 2，用于人机界面、监控系统例行操作和数据传送功能：强制多个线圈(fc.15)；读通用寄存器(fc.20)；写通用寄存器(fc.21)；屏蔽写寄存器(fc.22)；读写寄存器(fc.23)

1.5 Modbus TCP 通讯应用举例

在读寄存器的过程中,以 Modbus TCP 请求报文为例,具体的数据传输过程如下:

1) Modbus TCP 客户端实况，用 Connect()命令建立目标设备 TCP 502 端口连接数据通信过程

2) 准备 Modbus 报文，包括 7 个字节 MBAP 内请求；

3) 使用 send()命令发送；

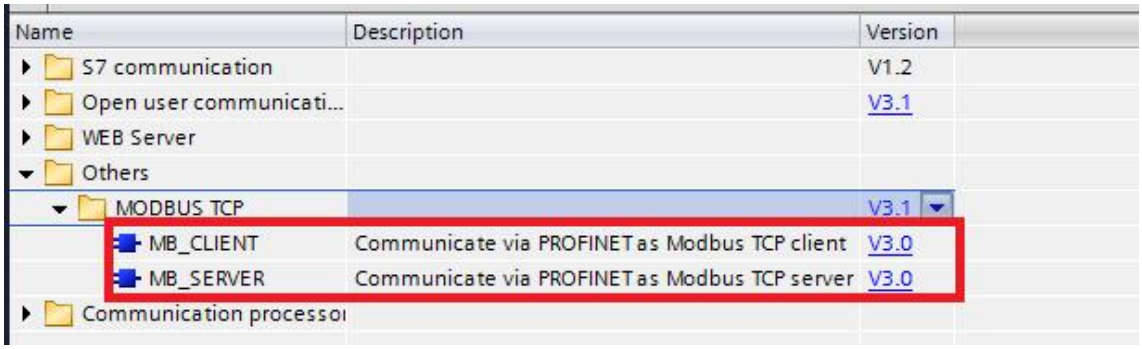
4) 同一连接等待应答；

5) 同 recv()读报文，完成一次数据交换过程

6) 当通信任务结束时，关闭 TCP 连接，使服务器可以为其他服务

2 SIMATIC S7-1500 Modbus TCP 通讯概述

S7-1500 PLC 需要通过 TIA Portal 博途软件进行组态配置，从 TIA Portal V12 SP1 开始软件中增加了 S7-1500 的 Modbus/TCP 块库，用于 S7-1500 与支持 Modbus/TCP 的通信伙伴进行通信，如下图 1 所示：



Picture1:TIA Portal 中包含的 ModbusTCP 块库

在使用 该块库时需要注意以下几点：

1）该块库只针对于 S7-1500 CPU 的集成 PROFINET 接口，对于集成的普通以太网口不适用。

2) 该块库包含为客户端/服务器，可分别将 S7-1500 创建为 Modbus/TCP Sever 及 Client 用于与通信伙伴通讯。

下面将分别介绍如何配置 S7-1500 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下表 1、2 所示：

名称	数量	订货号
SIMATIC CPU1516-3PN/DP(固件 V1.5)	1	6ES7151-3AN00-0AB0
网线	若干	
编程器兼软件测试机	1	

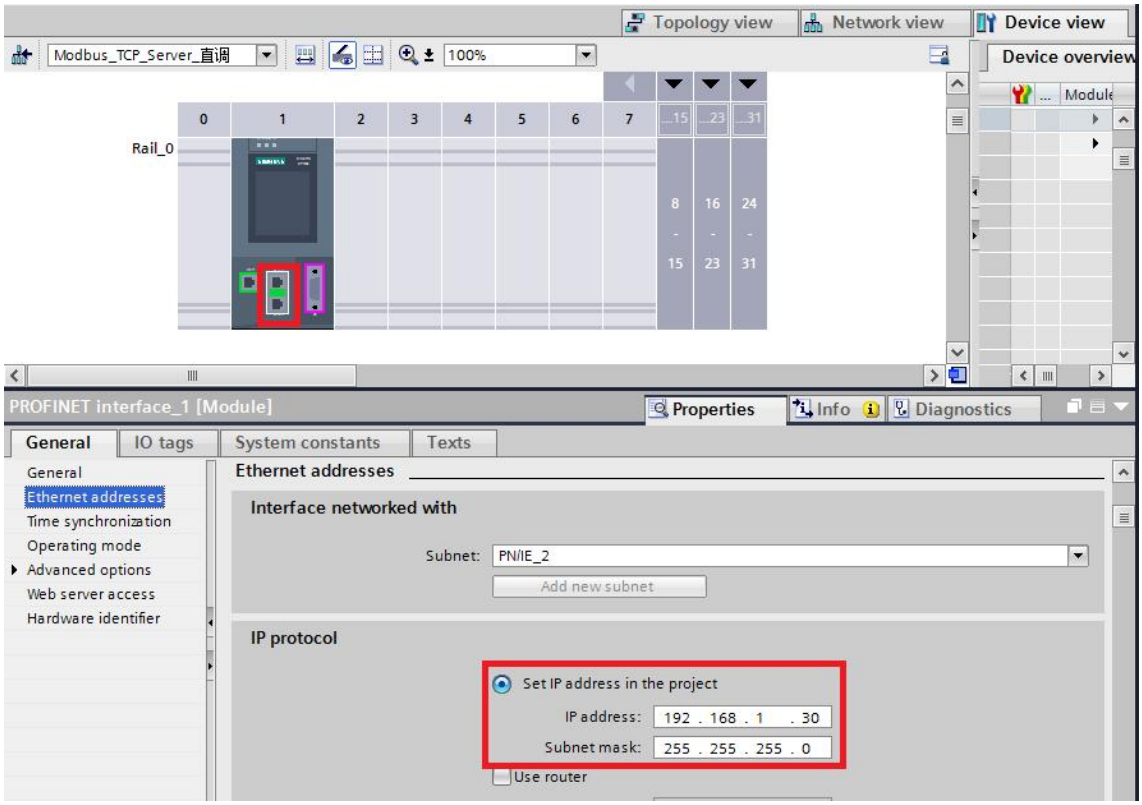
Table1：例程中用到的硬件列表

名称	订货号
SIMATIC STEP7 Prossional V13	
Modscan32 用于在 PC 中模拟 Modbus Client	
Modbus Slave V4.3 用于在 PC 中模拟 Modbus Server	V4.3.0(免授权版)

Table2：例程中用到的软件列表

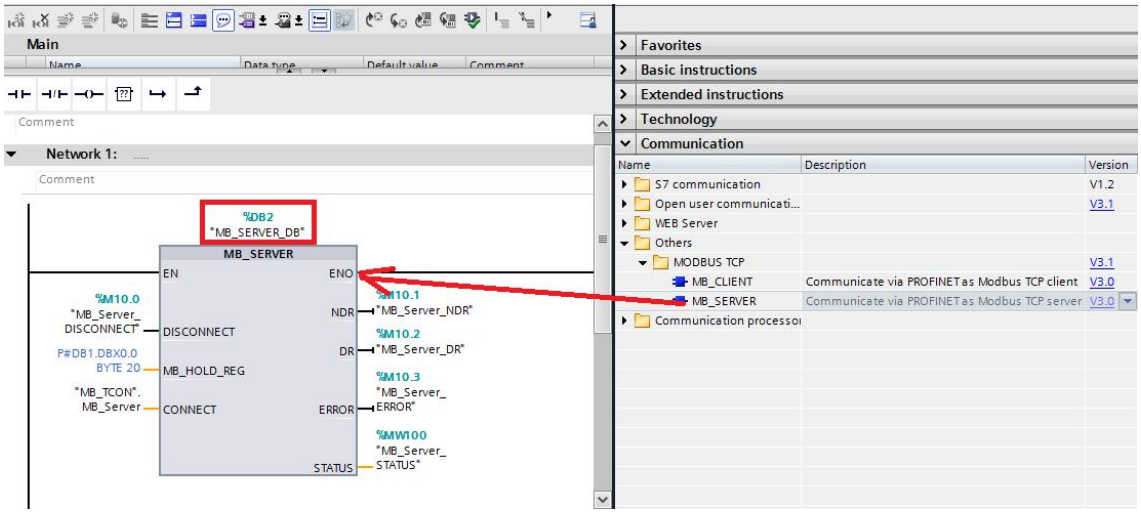
3 配置 S7-1500 CPU 作为 Modbus/TCP Server 与通信伙伴建立通讯

打开 TIA Portal V13 软件，新建一个项目，本例中命名为“S7_1500ModbusTCP_Final_V13”，在项目中添加 CPU1516-3PN/DP，为集成的 PROFINET 接口新建一个子网并设置 IP 地址，本例中为“192.168.1.30”，如下图 2 所示：



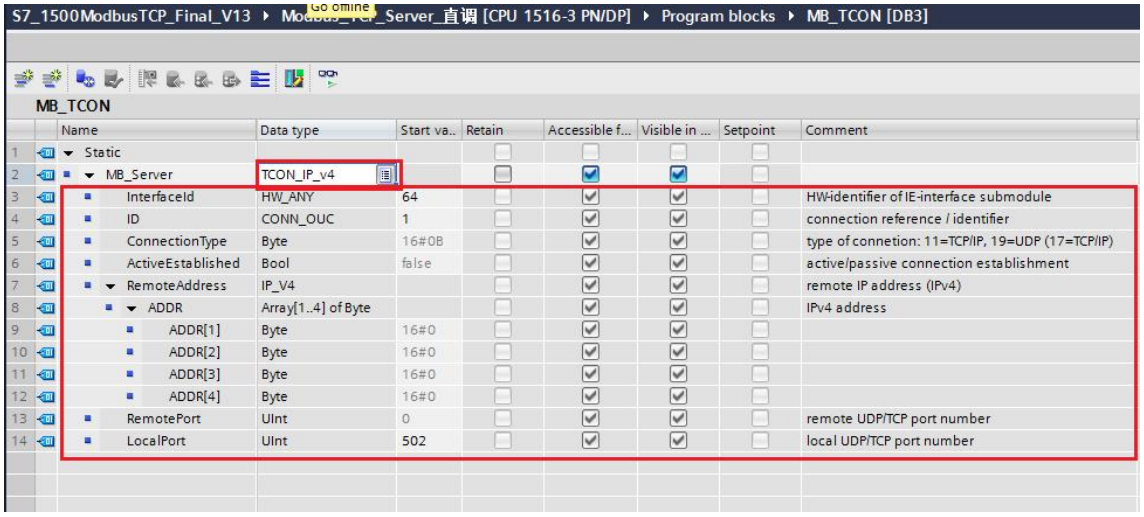
Picture2：新建一个 S7-1500 项目并配置 IP 地址

在 CPU1516-3PN/DP 的 OB1 组织块中添加 Modbus/TCP Server 功能块“ MB_SERVER”， 软件将提示会为该 FB 块增加一个背景数据块,本例中为 DB2“ MB_SERVER_DB”， 如下图 3 所示：



Picture3：添加“ MB_SERVER”功能块

之后在 CPU1516-3PN/DP 中添加一个全局数据块用于匹配功能块“ MB_SERVER”的管脚参数“ CONNECT ”,本例中为数据块“ MB_TCON”， 打开该数据块，手动输入一个“ TCON_IP_v4”的数据类型，如下图 4 所示：



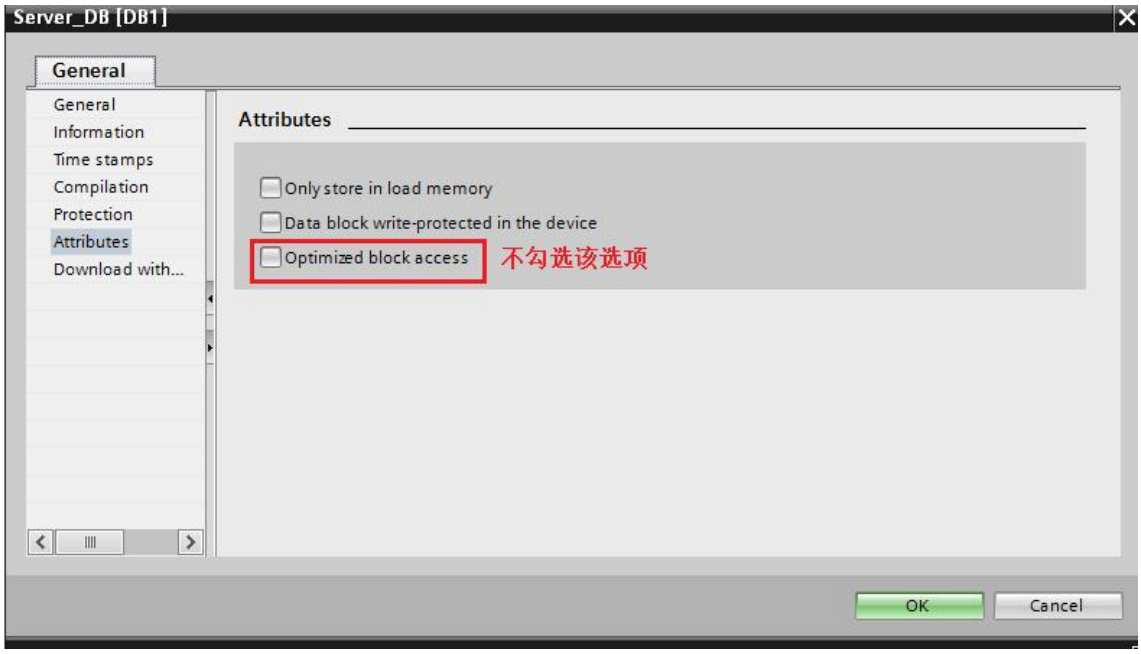
Picture4：添加一个与管脚“ CONNECT”匹配的全局数据块

展开 DB 块后其“ TCON_IP_v4”的数据类型的各参数设置如下：

TCON_IP_V4 数据类型管脚定义	含义	本例中的情况
Interfaced	接口，固定为 W#16#64	W#16#64
ID	连接 ID,每个连接必须独立	W#16#01
ConnectionType	连接类型，16#0B=TCP/IP;16#13=UDP	16#0B=TCP/IP
ActiveEstablished	是否主动建立连接，True=主动	False
RemoteAddress	通信伙伴 IP 地址，设置为 0 允许远程任意的 IP 建立连接	16#0
RemotePort	通信伙伴端口号，设置为 0 允许远程任意的端口建立连接	16#0
LocalPort	本地端口号，缺省的 Modbus/TCP Server 为 502	502

Table3：“ TCON_IP_v4”的数据类型的各参数设置

创建一个全局数据块用于匹配功能块“ MB_SERVER”的管脚参数“ MB_HOLD_REG”，本例中为“ SERVER_DB”，用于存储保持寄存器的通信数据，需要注意的是该数据块必须为非优化数据块(支持绝对寻址)，在该数据块的属性中不勾选“ Optimized block access”选项，如下图 5 所示：



Picture5：创建保持寄存器存储数据块

功能块“ MB_SERVER”的其它管脚参数如下表 4 所示：

“ MB_SERVER ”的管脚参数	管脚声明	数据类型	含义
DISCONNECT	输入	BOOL	0: 0：在无通信连接时建立被动连接。 1: 终止连接初始化。
NDR	输出	BOOL	0: 无新数据 1: 从 Modbus 客户端写入的新数据
DR	输出	BOOL	0: 未读取数据 1: 从 Modbus 客户端读取的数据
ERROR	输出	BOOL	通信故障
STATUS	输出	WORD	通信状态信息，用于诊断

Table4：功能块“ MB_SERVER”的其它管脚参数

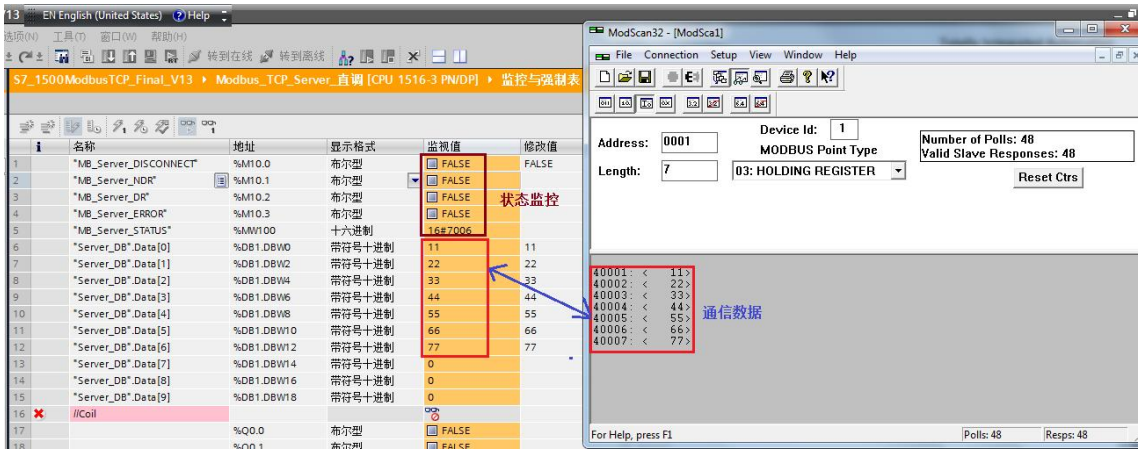
上面提到保持寄存器是由功能块“ MB_SERVER”的管脚参数“ MB_HOLD_REG”关联，对于其它数据类型，如线圈、离散输入、输入寄存器等通过功能块均已经与 S7-1500 的过程映像区进行了映射，其映射地址对应如下图 6 所示：

Modbus 功能						S7-1500、S7-1200 V4.0	
功能代码	功能	数据区	地址空间			数据区	CPU 地址
01	读取：位	Output	1	至	9999	过程映像输出	Q0.0 至 Q1249.6
02	读取：位	Input	1	至	19999	过程映像输入	I0.0 至 I1249.6
04	读取：WORD	Input	1	至	39999	过程映像输入	IW0 至 IW19996
05	写入：位	Output	1	至	9999	过程映像输出	Q0.0 至 Q1249.6
15	写入：位	Output	1	至	9999	过程映像输出	Q0.0 至 Q1249.6

Picture7:S7-1500 的 Modbus 地址映射表

设置完上述各管脚参数后，下载项目到 CPU1516-3PN/DP 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。

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 Modbus 偏移量及长度，建立与 CPU1315-3PN/DP 集成 PN 口的通信连接，可以看到双方可以建立通信连接并进行数据读写，如下图 8 所示：



Picture 8：通信测试

对于其它数据类型，由于与 S7-1500CPU 的过程映像区进行了映射，其过程类似。

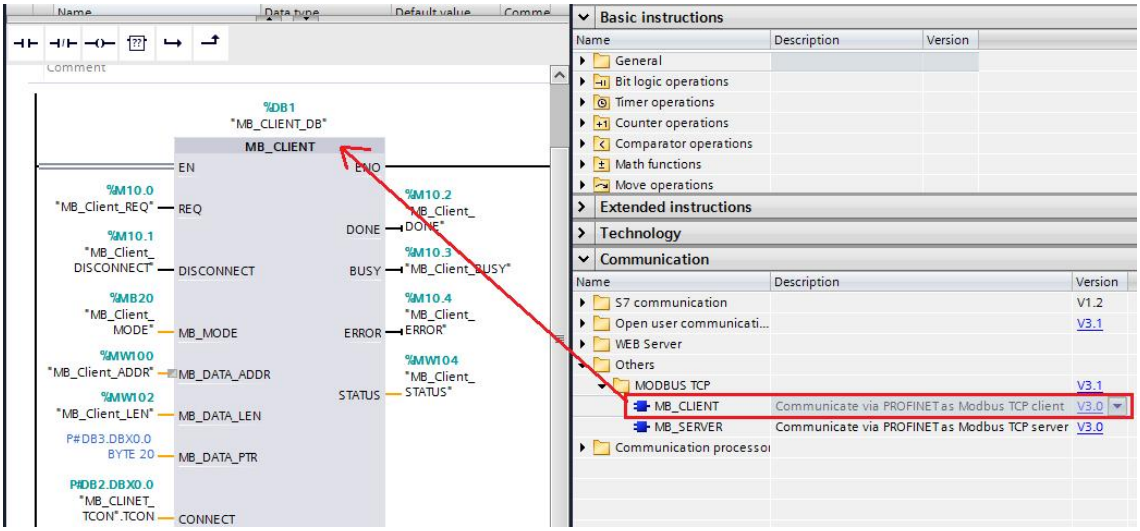
使用功能块“ MB_SERVER”的一些注意事项：

1）S7-1500 CPU 的集成 PN 口通过功能块“ MB_SERVER”支持与多个 Modbus 客户端的通信，支持的个数取决于 CPU 集成 PN 口所支持的 TCP 连接数，必须为每一个客户端连接分别调用一次功能块“ MB_SERVER”，其背景数据块、ID、端口号等参数必须唯一。

2）S7-1500 CPU 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

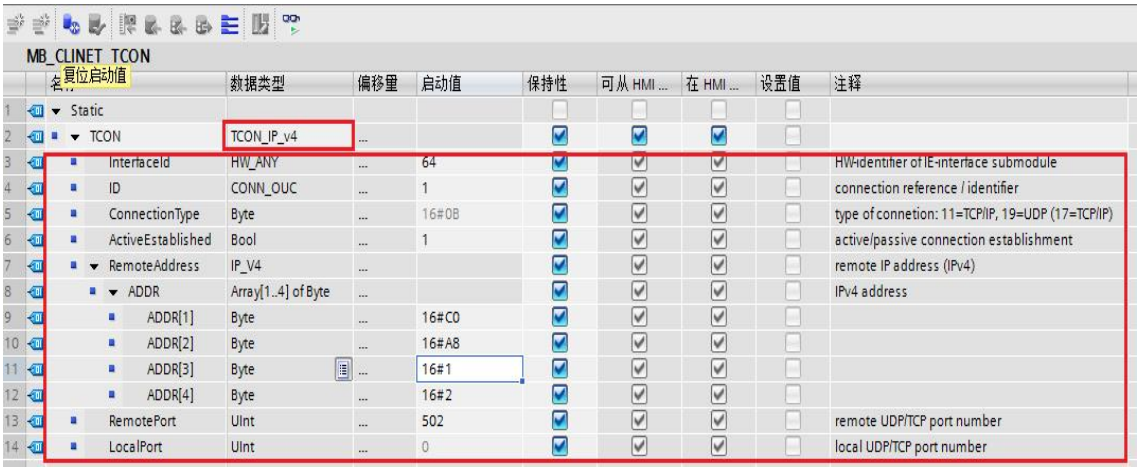
4 配置 S7-1500 CPU 作为 Modbus/TCP Client 与通信伙伴建立通讯

在上述新建的项目中增加一个 CPU1516-3PN/DP 的站点，设置 PROFINET 的 IP 地址“192.168.1.30”，之后在 CPU1516-3PN/DP 的 OB1 组织块中添加 Modbus/TCP Client 功能块“MB_CLIENT”，软件将提示会为该 FB 块增加一个背景数据块,本例中为 DB1“MB_CLIENT_DB”，如下图 9 所示：



Picture9：插入一个 MB_CLIENT 功能块

同样在 CPU1516-3PN/DP 中添加一个全局数据块用于匹配功能块“MB_CLIENT”的管脚参数“CONNECT”,本例中为数据块“MB_CLIENT_TCON”，打开该数据块，手动输入“TCON_IP_v4”的数据类型，如下图 10 所示：



Picture10：添加一个与管脚“CONNECT”匹配的全局数据块

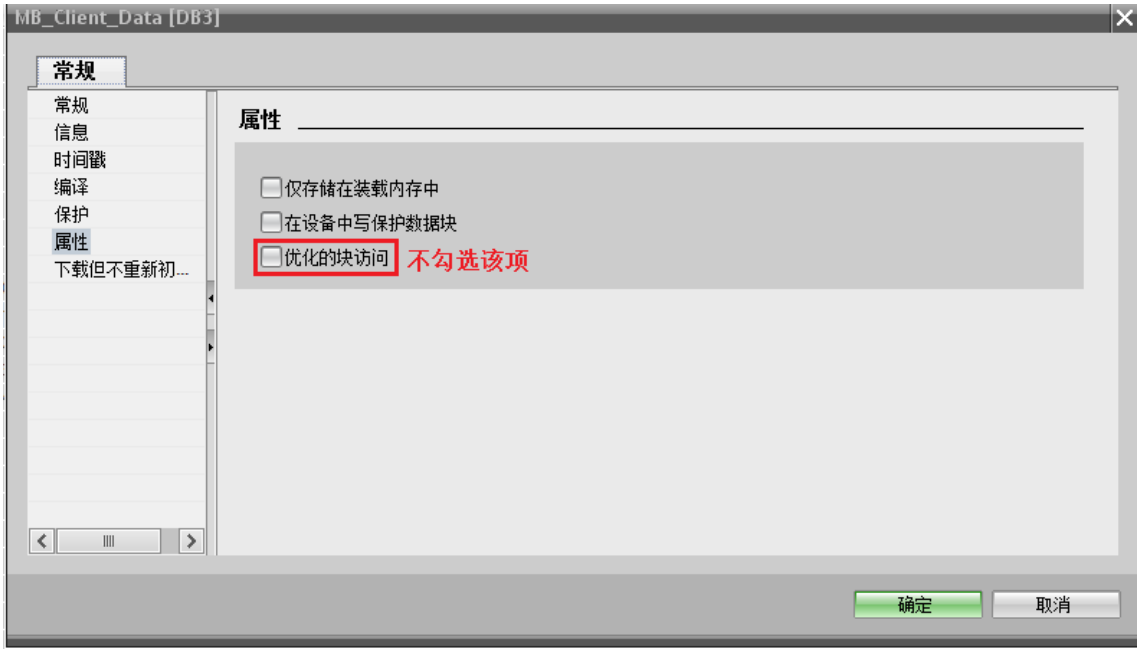
展开 DB 块后其“TCON_IP_v4”的数据类型的各参数设置如下：

TCON_IP_V4 数据类型管脚定义	含义	本例为 CLIENT 时 的情况

Interfaced	接口，固定为 W#16#64	W#16#64
ID	连接 ID,每个连接必须独立	W#16#01
ConnectionType	连接类型，16#0B=TCP/IP;16#13=UDP	16#0B=TCP/IP
ActiveEstablished	是否主动建立连接，True=主动	True
RemoteAddress	通信伙伴 IP 地址	192.168.1.2
RemotePort	通信伙伴端口号	16#502
LocalPort	本地端口号，设置为 0 将由软件自己创建	0

Table5：“ TCON_IP_v4”的数据类型的各参数设置

创建一个全局数据块用于匹配功能块“ MB_CLIENT”的管脚参数“ MB_DATA_PTR”，本例中为 DB3“ MB_Client_Data”，用于存储 Modbus 通信的各数据类型，需要注意的是该数据块必须为非优化数据块(支持绝对寻址)，在该数据块的属性中不勾选“ Optimized block access”选项，如下图 11 所示：



Picture11：创建 Modbus 存储 Modbbus 通信数据的数据块

对于功能块“ MB_CLIENT”的其它参数管脚含义如下表 6 所示：

“ MB_CLIENT”的管脚参数	管脚类型声明	数据类型	含义
REQ	输入	BOOL	与 Modbus TCP 服务器之间的通信请求,上升沿有效
DISCONNECT	输入	BOOL	0：与通过 CONNECT 参数组态的连接伙伴建立通信连接。 1： 断开通信连接

MB_MODE	输入	USINT	选择 Modbus 请求模式（0=读取、1=写入或诊断）
MB_DATA_ADDR	输入	UDINT	由“ MB_CLIENT”指令所访问数据的起始地址
MB_DATA_LEN	输入	UINT	数据长度： 数据访问的位数或字数
DONE	输出	BOOL	只要最后一个作业成功完成，立即将输出参数 DONE 的位置位为“ 1 ”
BUSY	输出	BOOL	0：无 Modbus 请求在进行中 1：正在处理 Modbus 请求
ERROR	输出	BOOL	0: 无错误 1: 出错。 出错原因由参数 STATUS 指示
STATUS	输出	WORD	指令的详细状态信息

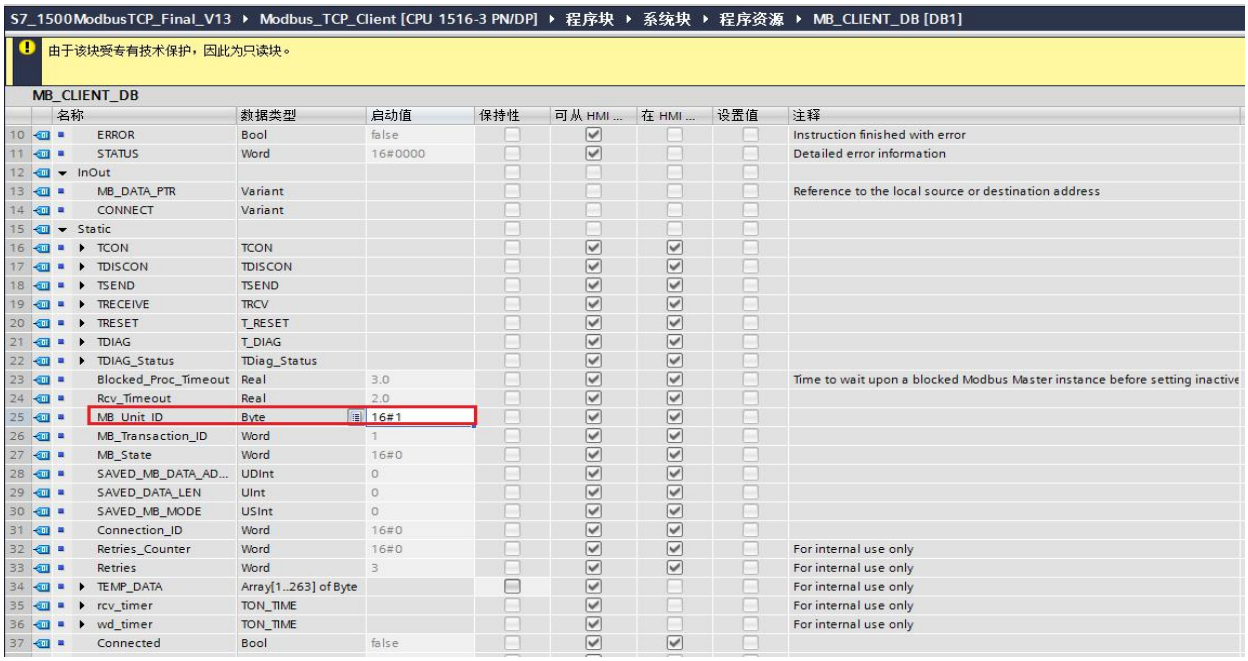
Table 6：功能块“ MB_CLIENT”的其它管脚参数

对于“ MB_MODE ” “ MB_DATA_ADDR”和“ MB_DATA_LEN”参数，其对应关系如下图 12 所示：

MB_MODE	MB_DATA_ADDR	MB_DATA_LEN	Modbus 功能	功能和数据类型
0	起始地址： • 1 到 9999	每个调用的数据长度（位）： • 1 到 2000	01	读取输入位： • 1 到 2000
0	起始地址： • 10001 到 19999	每个调用的数据长度（位）： • 1 到 2000	02	读取输入位： • 1 到 2000
0	起始地址： • 40001 到 49999 • 400001 到 465535	每个调用的数据长度(WORD)： • 1 到 125 • 1 到 125	03	读取保持寄存器： • 0 到 9998 • 0 到 65534
0	起始地址： • 30001 到 39999	每个调用的数据长度(WORD)： • 1 到 125	04	读取输入字： • 0 到 9998
1	起始地址： • 1 到 9999	每个调用的数据长度（位）： • 1	05	写入输出位： • 0 到 9998
1	起始地址： • 40001 到 49999 • 400001 到 465535	每个调用的数据长度(WORD)： • 1 • 1	06	写入保持寄存器： • 0 到 9998 • 0 到 65524
1	起始地址： • 1 到 9999	每个调用的数据长度（位）： • 2 到 1968	15	写入多个输出位： • 0 到 9998
1	起始地址： • 40001 到 49999 • 400001 到 465534	每个调用的数据长度(WORD)： • 2 到 123 • 2 到 123	16	写入多个保持寄存器： • 0 到 9998 • 0 到 65534
2	起始地址： • 1 到 9999	每个调用的数据长度（位）： • 1 到 1968	15	写入一个或多个输出位： • 0 到 9998
2	起始地址： • 40001 到 49999 • 400001 到 465535	每个调用的数据长度(WORD)： • 1 到 123 • 1 到 123	16	写入一个或多个保持寄存器： • 0 到 9998 • 0 到 65534

Picture12：“ MB_MODE ” “ MB_DATA_ADDR”和“ MB_DATA_LEN”参数对应关系

之后打开上述功能块“ MB_CLIENT”的背景数据块，在“ MB_UNIT_ID”参数中表示通信服务器伙伴的从站地址，该地址必须与通信伙伴一致，如 下图 13 所示：

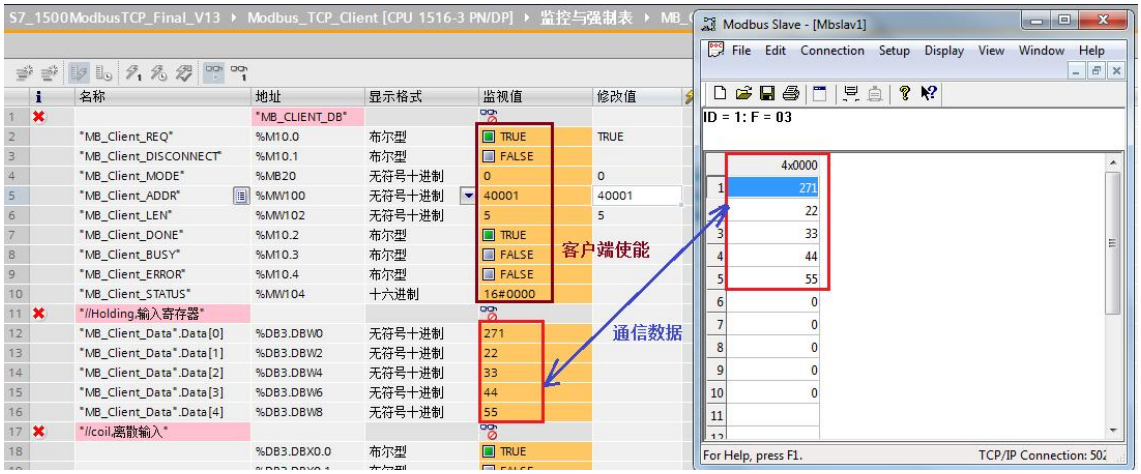


	名称	数据类型	启动值	保持性	可从 HMI ...	在 HMI ...	设置值	注释
10	ERROR	Bool	false		☑			Instruction finished with error
11	STATUS	Word	16#0000		☑			Detailed error information
12	InOut							
13	MB_DATA_PTR	Variant						Reference to the local source or destination address
14	CONNECT	Variant						
15	Static							
16	TCON	TCON			☑	☑		
17	TDISCON	TDISCON			☑	☑		
18	TSEND	TSEND			☑	☑		
19	TRECEIVE	TRCV			☑	☑		
20	TRESET	T_RESET			☑	☑		
21	TDIAG	T_DIAG			☑	☑		
22	TDIAG_Status	TDiag_Status			☑	☑		
23	Blocked_Proc_Timeout	Real	3.0		☑	☑		Time to wait upon a blocked Modbus Master instance before setting inactive
24	Rcv_Timeout	Real	2.0		☑	☑		
25	MB_Unit_ID	Byte	16#1		☑	☑		
26	MB_Transaction_ID	Word	1		☑	☑		
27	MB_State	Word	16#0		☑	☑		
28	SAVED_MB_DATA_AD...	UDInt	0		☑	☑		
29	SAVED_DATA_LEN	UInt	0		☑	☑		
30	SAVED_MB_MODE	USInt	0		☑	☑		
31	Connection_ID	Word	16#0		☑	☑		
32	Retries_Counter	Word	16#0		☑	☑		For internal use only
33	Retries	Word	3		☑	☑		For internal use only
34	TEMP_DATA	Array[1..263] of Byte			☑			For internal use only
35	rcv_timer	TON_TIME			☑			For internal use only
36	wd_timer	TON_TIME			☑			For internal use only
37	Connected	Bool	false		☑	☑		

Picture13：在功能块“ MB_CLIENT”的背景数据块设置从站地址

设置完上述各管脚参数后，下载项目到 CPU1516-3PN/DP 中，打开 Modslave 应用程序，下面以保持寄存器为例介绍通信测试过程。

在 Modslave 的数据定义界面中设置数据类型为保持寄存器，在在功能块“ MB_CLIENT”的“ MB_DATA_ADDR”和“ MB_DATA_LEN”设置起始地址和长度，可以看到双方可以建立通信连接并进行数据读写，如下图 14 所示：



	名称	地址	显示格式	监视值	修改值
1		"MB_CLIENT_DB"			
2	"MB_Client_REQ"	%M10.0	布尔型	TRUE	TRUE
3	"MB_Client_DISCONNECT"	%M10.1	布尔型	FALSE	
4	"MB_Client_MODE"	%MB20	无符号十进制	0	0
5	"MB_Client_ADDR"	%MW100	无符号十进制	40001	40001
6	"MB_Client_LEN"	%MW102	无符号十进制	5	5
7	"MB_Client_DONE"	%M10.2	布尔型	TRUE	
8	"MB_Client_BUSY"	%M10.3	布尔型	FALSE	
9	"MB_Client_ERROR"	%M10.4	布尔型	FALSE	
10	"MB_Client_STATUS"	%MW104	十六进制	16#0000	
11	"//Holding,输入寄存器"				
12	"MB_Client_Data".Data[0]	%DB3.DBW0	无符号十进制	271	
13	"MB_Client_Data".Data[1]	%DB3.DBW2	无符号十进制	22	
14	"MB_Client_Data".Data[2]	%DB3.DBW4	无符号十进制	33	
15	"MB_Client_Data".Data[3]	%DB3.DBW6	无符号十进制	44	
16	"MB_Client_Data".Data[4]	%DB3.DBW8	无符号十进制	55	
17	"//coil,离散输入"				
18		%DB3.DBX0.0	布尔型	TRUE	

	4x0000
1	271
2	22
3	33
4	44
5	55
6	0
7	0
8	0
9	0
10	0
11	

Picture14：通信测试

对于其它数据类型，测试过程类似。

使用功能块“ MB_CLIENT”的一些注意事项：

1）S7-1500 CPU 的集成 PN 口通过功能块“ MB_CLIENT”支持与多个 Modbus 服务器的

通信，支持的个数取决于 CPU 集成 PN 口所支持的 TCP 连接数，必须为每一个服务器连接需要分别调用一次功能块“ MB_CLIENT”，其背景数据块、ID 等参数必须唯一。

2）S7-1500 CPU 的集成 PN 口可以同时作为 Modbus/TCP 的 Server 及 Client