SIEMENS

应用与工具 •10 月/2014 年

S7-400 PN-H 冗余系统集成 PN 口使用 ModbusTCP PN RED 软件包的 Modbus TCP 快速入门

PN-H,MODBUS/TC P,MODBUS/TCP PN RED 软件包

http://support.automation.siemens.com/CN/view/zh/81715142

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；

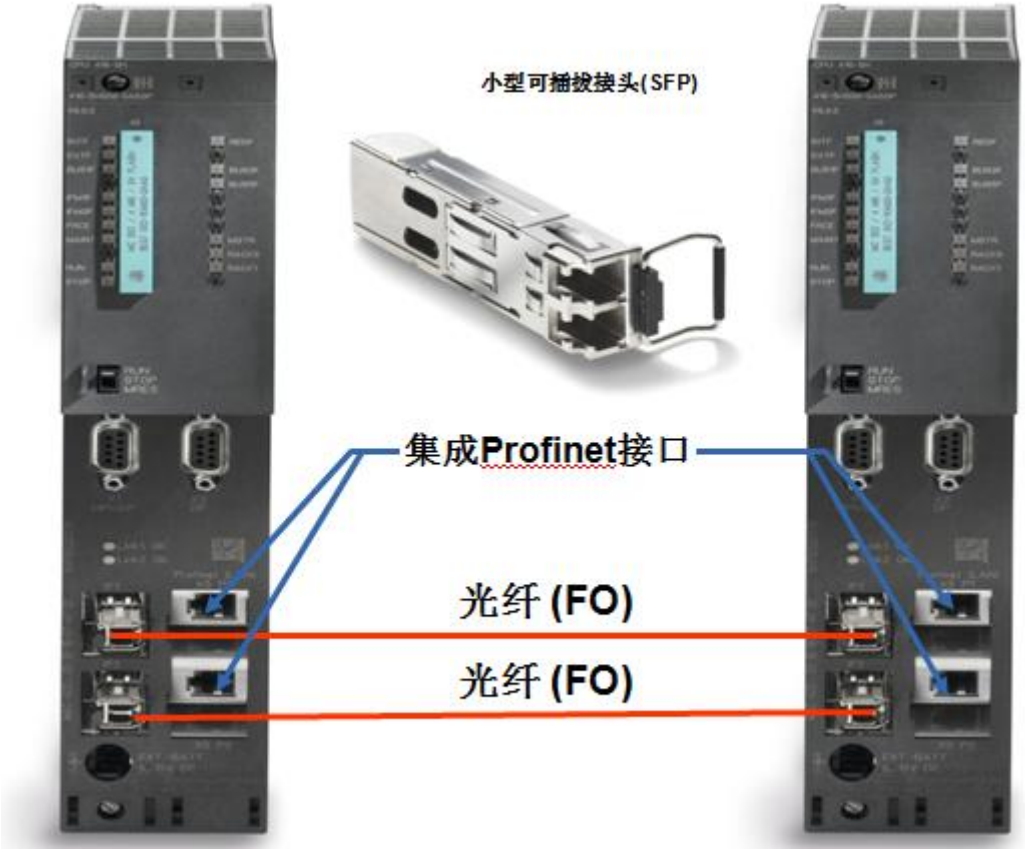
5) 同 recv()读报文，完成一次数据交换过程；

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 S7-400H 冗余系统 Modbus/TCP 通讯概述

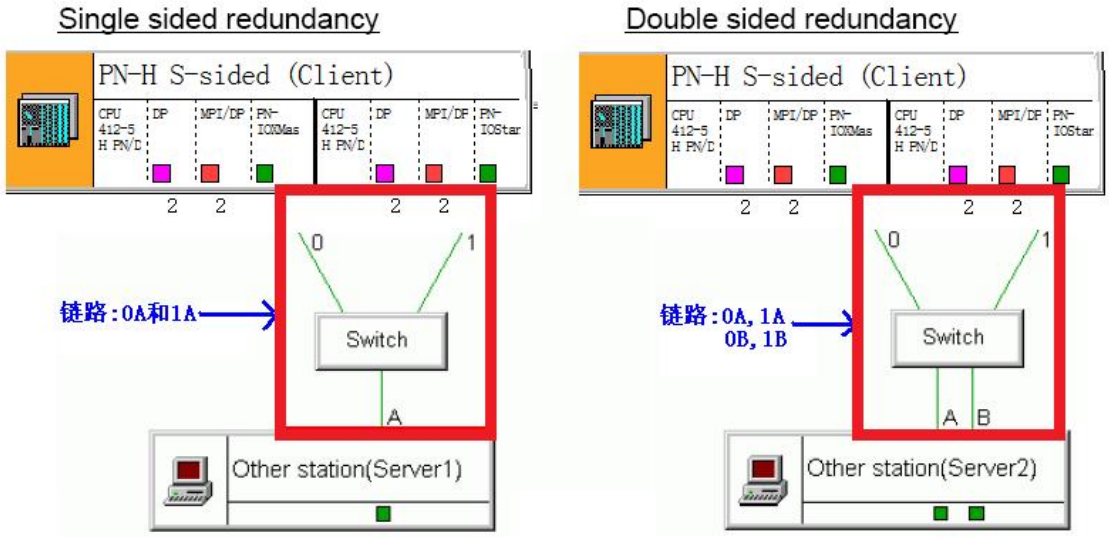
2.1 S7-400 PN-H 冗余系统 CPU 集成 PN 口 Modbus/TCP 通讯机理概述

S7-400 PN-H 冗余系统控制器是西门子推出的基于 Profinet 系统冗余的解决方案，其中每个 CPU 上集成了两个 Profinet 接口，如下图 1 所示：



Picture 1：集成 PN 接口的 S7-400 PN-H 冗余系统

2013 年 11 月西门子推出了基于 S7-400PN-H 冗余系统 CPU 集成 PROFINET 接口的 Modbus/TCP 冗余通信软件包“ Modbus/TCP PN CPU Redundant V1.0”，其通信网络架构和程序块版本如下图 2、3 所示：



Picture 2：S7-400 PN-H 冗余系统 CPU 集成 PN 口的 Modbus/TCP 网络架构

Product	Identification number	From version
Modbus/TCP PN CPU redundant	6AV6676-6MB10-0AX0	1.0
FB 913 "TCP_COMM"		3.2
FB 914 "MOD_CLI"		1.6
FB 915 "MB_PNHCL"		1.0
FB 916 "MOD_SERV"		1.5
FB 917 "MB_PNHSV"		1.0

Picture 3：软件包“ Modbus/TCP PN CPU Redundant V1.0”程序版本

软件包“ Modbus/TCP PN CPU Redundant V1.0”的通信机理及特点如下：

1）如果 S7-400 PN-H CPU 做 Modbus/TCP Server，该解决方案能够允许通信伙伴通过任意一个机架 CPU 的 PN 口通信链路(图 2 中单边的链路为 0A、1A；双边链路为 0A、1A/0B、1B)建立通信及数据的一致性，链路的选择完全由客户端自行根据链路的通信状况决定，当任一链路中断、PN-H CPU 冗余模式的切换均不会对通信造成任何影响

2）如果 S7-400 PN-H CPU 做 Modbus/TCP client，解决方案能够保证通过任意一个机架 CPU 的 PN 口通信链路(图 2 中单边的链路为 0A、1A；双边链路为 0A、1A/0B、1B)与通信伙伴建立通信及数据的一致性，当任一链路中断、PN-H CPU 冗余模式的切换均不会对通信造成任何影响

3）解决方案中提供了完整的 TCP 层及 Modbus 应用层的诊断

2.2 软件包“ Modbus/TCP PN CPU Redundant V1.0”使用说明

2.2.1 软件包“ Modbus/TCP PN CPU Redundant V1.0”软硬件需求

该软件包需要的软件需求如下图 4、5 所示：

400 PN-H-CPU (only Modbus/TCP PN and Modbus/TCP PN Red)	
6ES7 412-5HK06-0AB0	V6.0.1
6ES7 414-5HM06-0AB0	V6.0.1
6ES7 416-5HS06-0AB0	V6.0.1
6ES7 417-5HT06-0AB0	V6.0.1
6ES7 410-5HX08-0AB0	V8.0

Picture 4：软件包“ Modbus/TCP PN CPU Redundant V1.0”所需要的硬件需求

Software requirements

· SIMATIC STEP 7 version 5.5 SP2 HF1 or higher for the PN Red PLC version of the MODBUS blocks

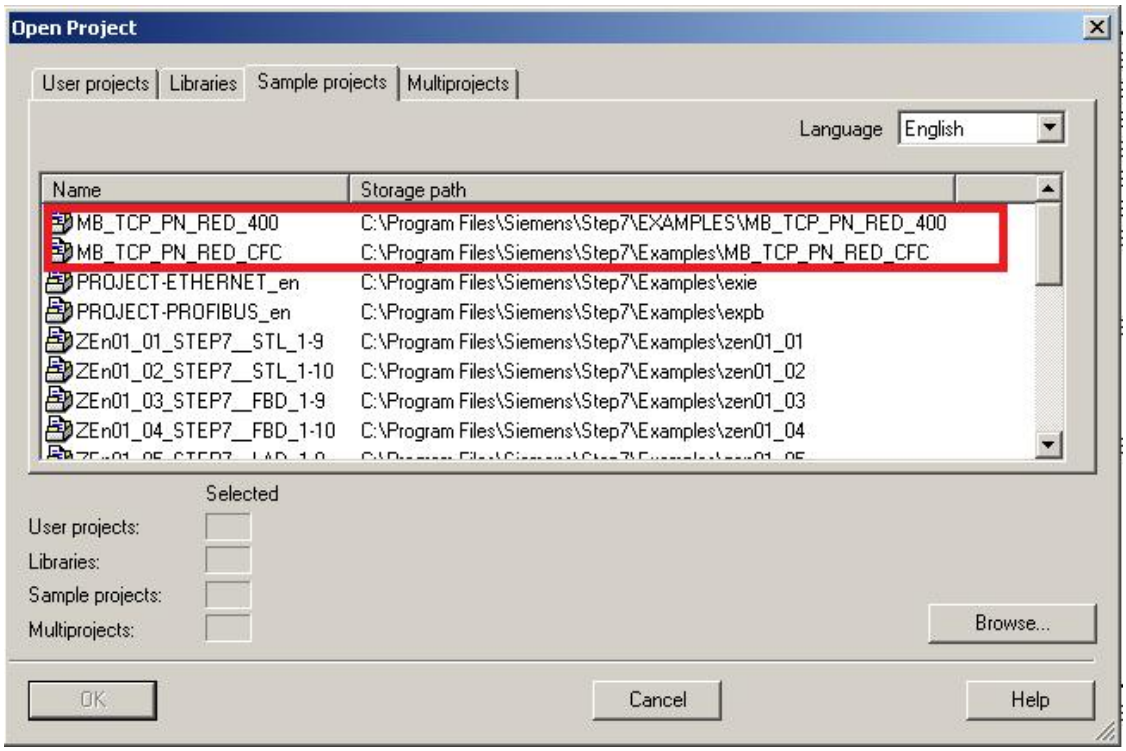
Picture 5：软件包“ Modbus/TCP PN CPU Redundant V1.0”所需要的软件需求

2.2.2 软件包“ Modbus/TCP PN CPU Redundant V1.0”软硬件需求

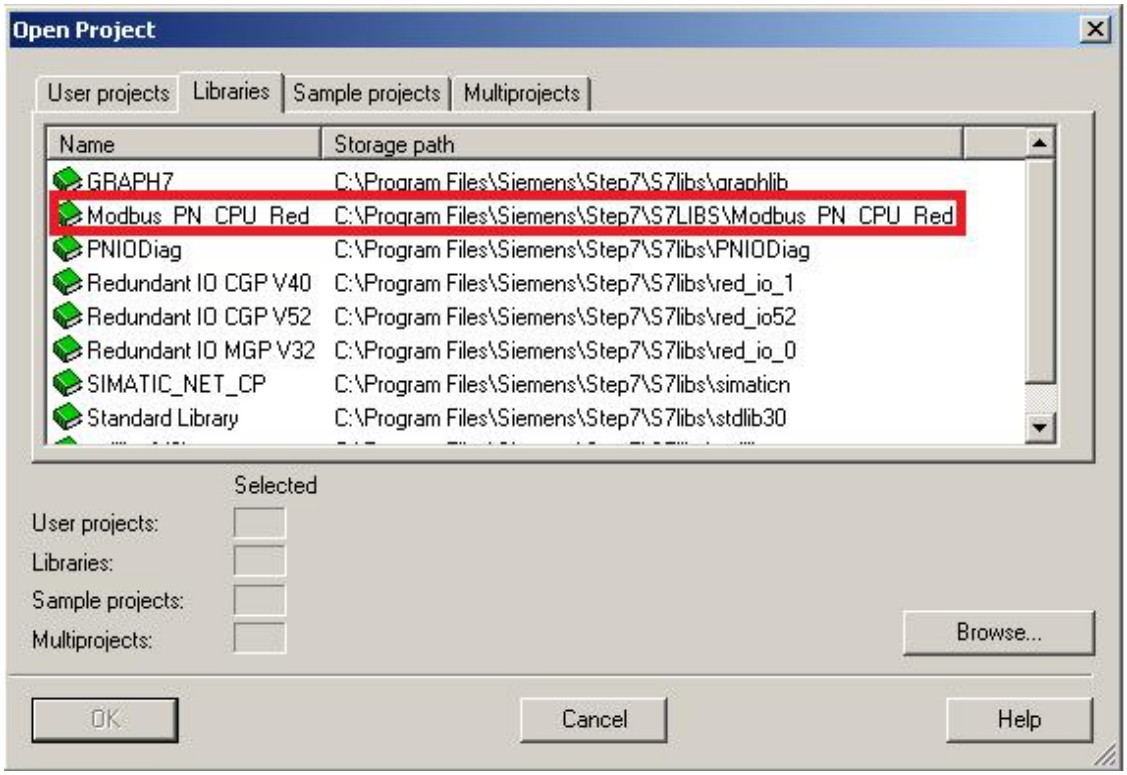
当将软件选项包安装完集成到 Step7 时可以在 Step7 安装文件的相应目录中找到块库、例程、英文手册，如下图 6-8 所示，在实际的项目调试过程中由于例子程序的各项功能比较完善，因此可以直接使用例子程序根据项目的实际情况修改相应的参数即可，可以节省大量的参数设置时间。

- The libraries under \Program Files\Siemens\Step7\S7libs,
- The example projects under \Program Files\Siemens\Step7\Examples,
- The manual under \Program Files\Siemens\Step7\S7manual\S7Comm.
- The software registration form under \Program Files\Siemens\Step7\S7libs\ Modbus_PN_CPU_Red.

Picture 6： 块库、例程、英文手册和软件注册的文件夹位置



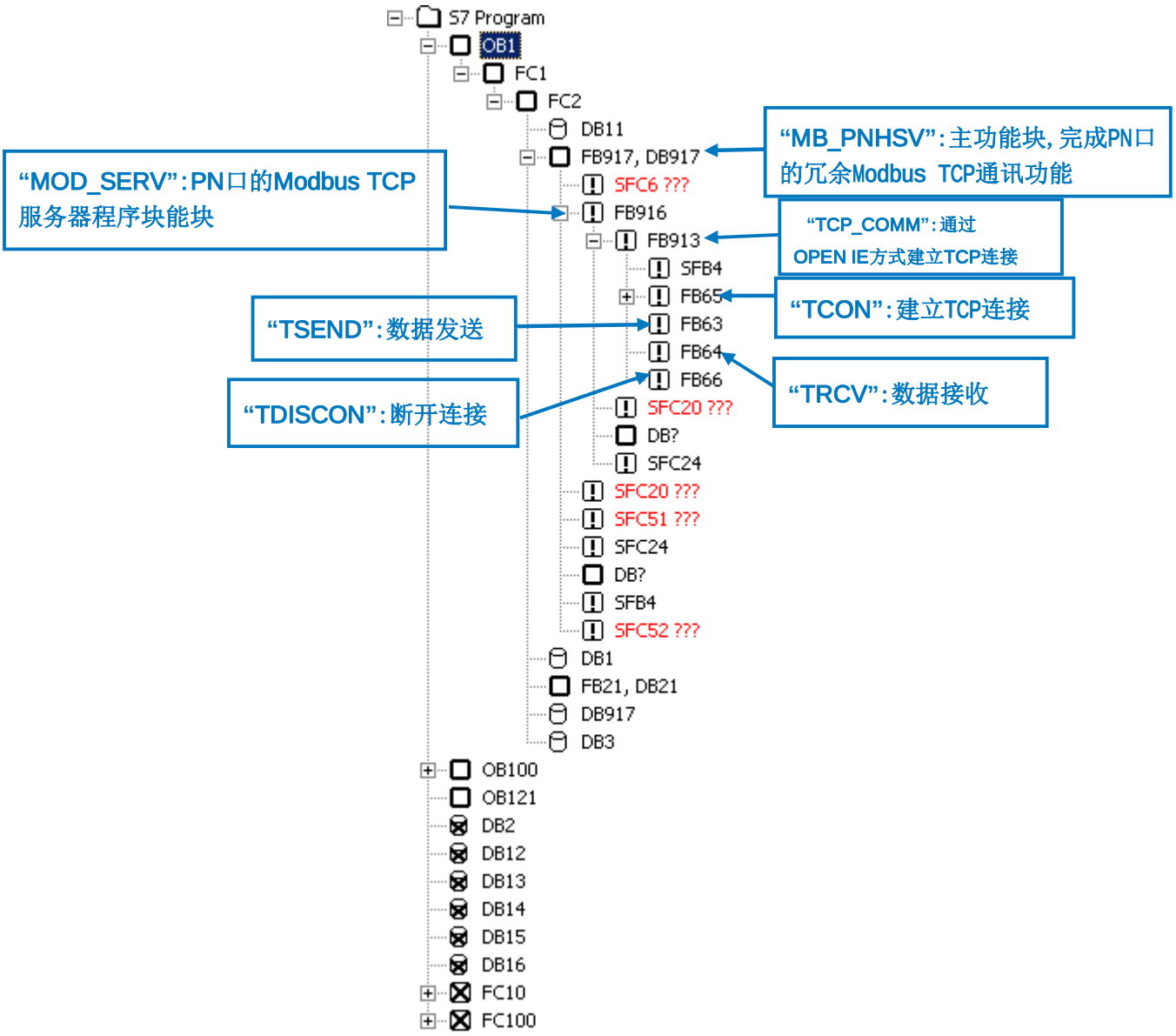
Picture 7：例程(注：当找不到例程时可以通过”Browse..”按钮来进行查找)



Picture 8：功能块库(注：当找不到块库时可以通过"Browse.."按钮来进行查找)

3 配置 S7-400 PN-H 冗余系统通过 CPU 集成 PN 口作为 Server 进行 Modbus TCP 通讯

下面以 S7-400 PN-H 冗余系统及 Modscan32 软件为例，详细介绍如何将 S7-400 PN-H 冗余系统通过 CPU 集成 PN 口配置为 Server，Modscan32 为 Client 进行 Modbus TCP 通讯，下图 9 为服务器功能块库的程序结构及各功能块完成的功能：



Picture 9：软件包“ Modbus/TCP PN CPU Redundant V1.0”服务器程序架构

注：Modscan32 软件可以从网上免费下载得到，本例中使用的版本为 V7.0 版，由于各版本的功能不尽相同，因此需要注意版本问题。

3.1 例子中使用的硬件设备及软件

本例中所用的硬件设备如下表：

名称	数量	订货号

S7-400 电源模块 PS 407 10A	2	6ES7407-0KA02-0AA0
S7-400 CPU412-5H PN/DP	2	6ES7412-5HK06-0AB0(V6.0)
S7-400 机架	1	6ES7400-2JA00-0AA0
网线	若干	
笔记本电脑	1	

Table 1:服务器硬件清单

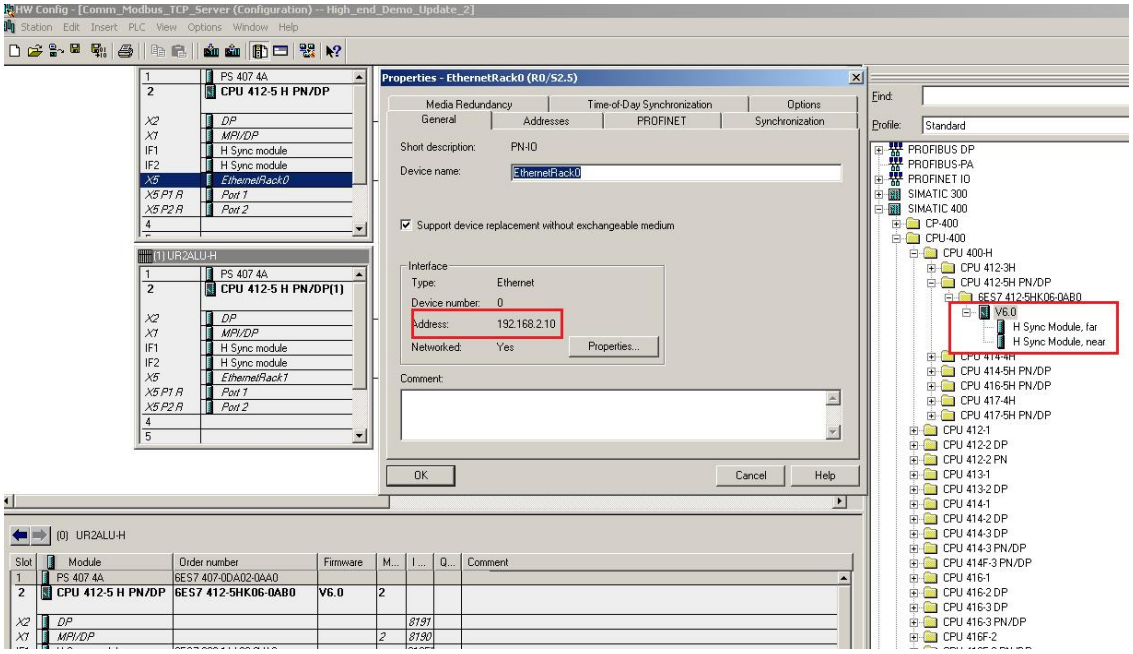
所用到软件如下表：

名称	订货号
STEP7 V5.5 SP3 组态编程软件 英文版	
“ Modbus/TCP PN CPU Redundant V1.0 ” 软件选项包	6AV6676-6MB10-0AX0
Modscan32 V7.0	

Table 2:服务器软件清单

3.2 S7-400 PN-H 冗余系统及 Modscan32 软件组态

打开 STEP7 软件， 新建一个项目文件，插入一个“ SIMATIC H 站”，在硬件组态中分别插入 PS407 电源，CPU412-5H PN/DP 等并设置 Rack 0、1 CPU 的集成 PN 接口的 IP 地址，如下图 10 所示 ：



Picture 10：硬件组态

硬件组态完成后，编译保存，并将例程站点“ H Single-sided (Server)”中的程序（System data 不需要拷贝）拷贝到该项目中。

由于需要在 SIMATIC 站与其他通讯伙伴之间建立 TCP 连接用于 Modbus 通讯，而对于 CPU 的集成 PN 口来说须通过 Open IE(开放式以太网通讯)的方式来建立 TCP 连接， 通过 S7-CPU 的 PROFINET 接口 进行 Modbus TCP 通信时，需要使用通信块 FB65 "TCON"、FB66 "TDISCON"、FB63 "TSEND" 和 FB64 "TRCV"，要进行 Modbus TCP 通信，必须在数据块中为 Rack0 及 Rack1 冗余 CPU 指定相应的参数，相应得参数在程序中主要由 DB2 “ MODBUS_HPARAM_PN_2”来完成初始化(注意：参数设置必须在“ Data View”视图下的“ Actual value”列中设置)，其中各参数的含义如下图 11、12 所示：

Address	Name	Type	Initial value	Comment
0.0		STRUCT		
+0.0	double_sided_red	BOOL	FALSE	
+2.0	connection_0A	STRUCT		
+0.0	block_length	WORD	W#16#40	#!Verbindung_0001!#
+2.0	id	WORD	W#16#1	
+4.0	connection_type	BYTE	B#16#11	
+5.0	active_est	BOOL	FALSE	
+6.0	local_device_id	BYTE	B#16#5	
+7.0	local_tsap_id_len	BYTE	B#16#2	
+8.0	rem_subnet_id_len	BYTE	B#16#0	
+9.0	rem_staddr_len	BYTE	B#16#0	
+10.0	rem_tsap_id_len	BYTE	B#16#0	
+11.0	next_staddr_len	BYTE	B#16#0	
+12.0	local_tsap_id	ARRAY[1..16]	B#16#1, B#16#F6	
*1.0		BYTE		
+28.0	rem_subnet_id	ARRAY[1..6]	B#16#0, B#16#0,	
*1.0		BYTE		
+34.0	rem_staddr	ARRAY[1..6]	B#16#0, B#16#0,	
*1.0		BYTE		
+40.0	rem_tsap_id	ARRAY[1..16]	B#16#0, B#16#0,	
*1.0		BYTE		
+56.0	next_staddr	ARRAY[1..6]	B#16#0, B#16#0,	
*1.0		BYTE		
+62.0	spare	WORD	W#16#0	#!Verbindung_0001!#
=64.0		END_STRUCT		

Rack0 CPU集成PN的OPEN IE TCP连接参数

+66.0	connection_1A	STRUCT		
+0.0	block_length	WORD	W#16#40	#!Verbindung_0002!#
+2.0	id	WORD	W#16#2	
+4.0	connection_type	BYTE	B#16#11	
+5.0	active_est	BOOL	FALSE	
+6.0	local_device_id	BYTE	B#16#15	
+7.0	local_tsap_id_len	BYTE	B#16#2	
+8.0	rem_subnet_id_len	BYTE	B#16#0	
+9.0	rem_staddr_len	BYTE	B#16#0	
+10.0	rem_tsap_id_len	BYTE	B#16#0	
+11.0	next_staddr_len	BYTE	B#16#0	
+12.0	local_tsap_id	ARRAY[1..16]	B#16#1, B#16#F6	
*1.0		BYTE		
+28.0	rem_subnet_id	ARRAY[1..6]	B#16#0, B#16#0,	
*1.0		BYTE		
+34.0	rem_staddr	ARRAY[1..6]	B#16#0, B#16#0,	
*1.0		BYTE		
+40.0	rem_tsap_id	ARRAY[1..16]	B#16#0, B#16#0,	
*1.0		BYTE		
+56.0	next_staddr	ARRAY[1..6]	B#16#0, B#16#0,	
*1.0		BYTE		
+62.0	spare	WORD	W#16#0	#!Verbindung_0002!#
=64.0		END_STRUCT		

Rack1 CPU集成PN的OPEN IE TCP连接参数

Picture11：DB2 “ MODBUS_HPARAM_PN_2”的 TCP 连接参数设置部分

关于 DB2 “ MODBUS_HPARAM_PN_2”的 Rack0 及 Rack1 冗余 CPU TCP 连接参数含义如下表 3 所示：

类型	参数	含义
OPEN IE 通讯参数	double_sided_red	通信伙伴是否为冗余系统，1=冗余，0=单站
	block_length	固定值W#16#40
	Id	连接ID,用于FB63/64/65/66 ,Rack0和Rack1CPU必须唯一
	connection_type	取决于CPU类型，用于FB65(TCON)

		TCP(兼容模式): CPU315、317<= FWV2.3 W#16#01； TCP:CPU315,317>= FW V2.4、IM151-8PN/DP CPU、 CPU314C、CPU319、CPU412、CPU414与CPU416 W#16#11
	active_est	主动或被动连接: S7作Client时为主动 TRUE S7作Server时为被动 FALSE
	local_device_id	取决于CPU类型： IM151-8PN/DP B#16#1 CPU314C、315、317 B#16#2 CPU319 B#16#3 CPU412(H)、414(H)、416(H) B#16#5 Rack1中的CPU B#16#15
	local_tsap_id_len	local_device_id的长度： 主动连接时 W#16#0 被动连接时 W#16#2
	rem_subnet_id_len	未使用
	rem_staddr_len	参数rem_staddr的长度： 未具体定义连接 B#16#0 有具体连接 B#16#4
	rem_tsap_id_len	rem_tsap_id的长度： 主动连接时 W#16#2 被动连接时 W#16#0
	next_staddr_len	通讯接口类型选择: 通过外部CP模块： 非0的其它值 通过CPU的集成PN 口： W#16#0
	local_tsap_id	本地连接TSAP号,与参数connection_type有关： 1)connection_type= B#16#01时 local_tsap_id[1] 本地连接端口号的低字节[16进制] local_tsap_id[2] 本地连接端口号的高字节[16进制] local_tsap_id[3-16] B#16#00 2)connection_type= B#16#11时

		local_tsap_id[1] 本地连接端口号的高字节[16进制] local_tsap_id[2] 本地连接端口号的低字节[16进制] local_tsap_id[3-16] B#16#00
	rem_subnet_id	未使用
	rem_staddr	通信伙伴的IP地址，与参数connection_type有关，以192.168.0.1为例： 1)connection_type= B#16#01时 rem_staddr[1]= B#16#01(1), rem_staddr[2]= B#16#00(0) rem_staddr[3]= B#16#A8(168) rem_staddr[4]= B#16#C0(192) rem_staddr[5-6]=B#16#00(为IPV6预留) 2)connection_type= B#16#11时 rem_staddr[1]= B#16#C0(192) rem_staddr[2]= B#16#A8(168) rem_staddr[3]= B#16#00(0) rem_staddr[4]= B#16#01(1) rem_staddr[5-6]=B#16#00(为IPV6预留)
	rem_tsap_id	远程连接TSAP号,与参数connection_type有关： 1)connection_type= B#16#01时 local_tsap_id[1] 本地连接端口号的低字节[16进制] local_tsap_id[2] 本地连接端口号的高字节[16进制] local_tsap_id[3-16] B#16#00 2)connection_type= B#16#11时 local_tsap_id[1] 本地连接端口号的高字节[16进制] local_tsap_id[2] 本地连接端口号的低字节[16进制] local_tsap_id[3-16] B#16#00
	next_staddr	CP的机架号和槽号，当使用CPU的PN口时为 B#16#00

Table 3： DB2 “ MODBUS_HPARAM_PN_2”的 TCP 连接参数含义

+130.0	server_client	BOOL	TRUE
+130.1	single_write	BOOL	FALSE
+130.2	connect_at_startup	BOOL	FALSE
+131.0	reserved	BYTE	B#16#0
+132.0	data_type_1	BYTE	B#16#3
+134.0	db_1	WORD	W#16#B
+136.0	start_1	WORD	W#16#0
+138.0	end_1	WORD	W#16#1F3
+140.0	data_type_2	BYTE	B#16#3
+142.0	db_2	WORD	W#16#C
+144.0	start_2	WORD	W#16#2D0
+146.0	end_2	WORD	W#16#384
+148.0	data_type_3	BYTE	B#16#4
+150.0	db_3	WORD	W#16#D
+152.0	start_3	WORD	W#16#2D0
+154.0	end_3	WORD	W#16#3E8
+156.0	data_type_4	BYTE	B#16#0
+158.0	db_4	WORD	W#16#0
+160.0	start_4	WORD	W#16#0
+162.0	end_4	WORD	W#16#0
+164.0	data_type_5	BYTE	B#16#1
+166.0	db_5	WORD	W#16#E
+168.0	start_5	WORD	W#16#280
+170.0	end_5	WORD	W#16#4E2
+172.0	data_type_6	BYTE	B#16#2
+174.0	db_6	WORD	W#16#F
+176.0	start_6	WORD	W#16#6A4
+178.0	end_6	WORD	W#16#8FC
+180.0	data_type_7	BYTE	B#16#1
+182.0	db_7	WORD	W#16#10
+184.0	start_7	WORD	W#16#6A4
+186.0	end_7	WORD	W#16#8FC
+188.0	data_type_8	BYTE	B#16#0
+190.0	db_8	WORD	W#16#0
+192.0	start_8	WORD	W#16#0
+194.0	end_8	WORD	W#16#0
+196.0	conn_0A_send_buffer	ARRAY[1..4	260 (B#16#0)
*1.0		BYTE	
+456.0	conn_0A_recv_buffer	ARRAY[1..4	260 (B#16#0)

Identifier	Data type	Size
0	Area not used	
1	Coils	Bit
2	Inputs	Bit
3	Holding Register	Word
4	Input Register	Word

Picture 12：DB2 “ MODBUS_HPARAM_PN_2”的 Modbus 参数设置部分

3.3 通讯测试

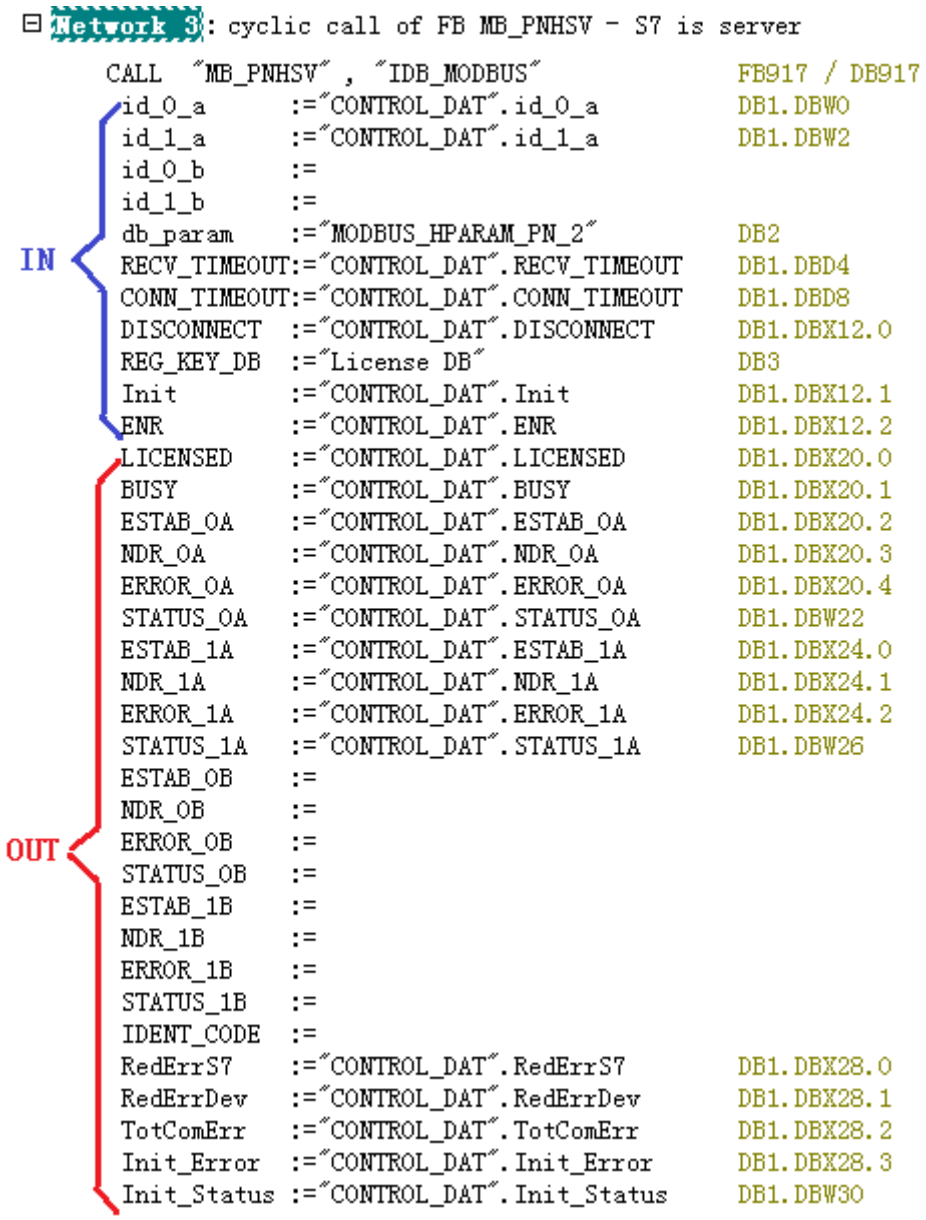
由于“ Modbus/TCP PN CPU Redundant V1.0”选项包支持功能码FC1，2，3，4，5，6，15，16，不同的功能码测试过程中类似，因此下面以FC03(读写保持寄存器)为例来说明通讯测试的整个过程，对于其他功能码的测试将不再重复描述，对于Modbus的数据类型可参考下表4：

基本表	对象类型	访问类型	注释
离散量输入	单个位	只读	I/O系统可提供这种类型数据
线圈	单个位	读写	通过应用程序可改变这种类型数据
输入寄存器	16位字	只读	I/O系统可提供这种类型数据
保持寄存器	16位字	读写	通过应用程序可改变这种类型数据

Table 4：Modbus 数据类型

在测试过程中我们将重点关注通讯连接的建立和当一个链路中断时自动切换到另一个链路的过程。

由于服务器主功能块 FB917“ MB_PNHSV”的参数需要初始化，因此分别在 OB100 及 OB1 中调用 FB917，在 OB100 中调用 FB917 完成相关参数的初始化，FB917 的管脚分布如下图 13 所示：



Picture 13： 功能块 FB917 “ MB_PNHSV”管脚分布

FB917 “ MB_PNHSV”的各参数含义如下表 5：

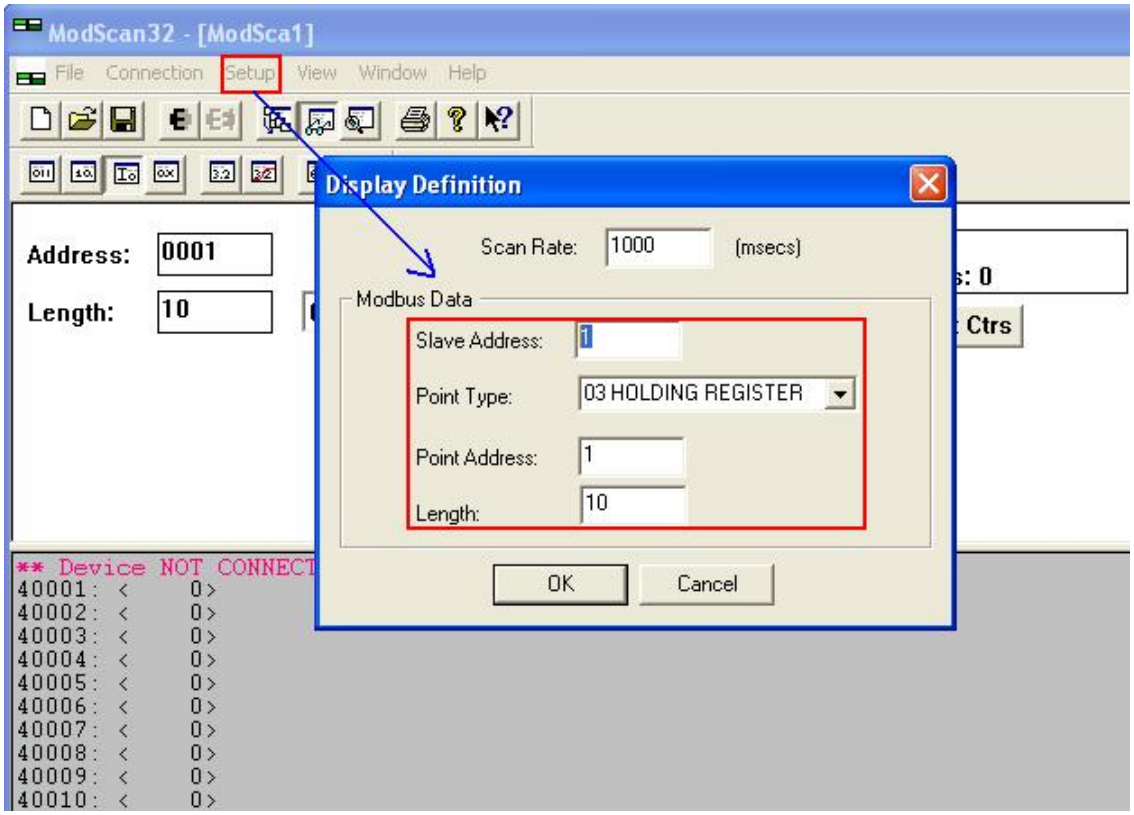
类型	参数	格式	含义		初始化
IN	id_0_a	WORD	假定两个 Rack CPU 简称为 CPU0、CPU1(下同)，如通讯伙伴也	CPU0 与 UP0 的连接 ID	是
	id_1_a	WORD		CPU1 与 UP0 的连接 ID	是
	id_0_b	WORD		CPU0 与 UP1 的连接 ID	是

	id_1_b	WORD	为冗余的话简称为 UP0、UP1	CPU1 与 UP1 的连接 ID	是
	db_param	BLOCK_DB	参数化 DB 块		是
	RECV_TIMEOUT	TIME	监视应用层接收数据的超时时间，最少 20ms		否
	CONN_TIMEOUT	TIME	TCP 连接建立超时监控时间，最短 100ms		否
	DISCONNECT	BOOL	在 ENR=FALSE 情况下为 TRUE 时断开远程伙伴的连接		否
	REG_KEY_DB	BLOCK_DB	授权 DB 块		否
	Init	BOOL	手动初始化		否
	ENR	BOOL	接收使能		否
OUT	LICENSED	BOOL	功能块是否授权		否
	BUSY	BOOL	作业正在处理		否
	NDR_0A	BOOL	CPU0 与 UP0 的连接请求成功响应和处理		否
	ERROR_0A	BOOL	CPU0 与 UP0 的连接错误		否
	STATUS_0A	WORD	CPU0 与 UP0 的连接状态		否
	NDR_1A	BOOL	CPU1 与 UP0 的连接请求成功响应和处理		否
	ERROR_1A	BOOL	CPU1 与 UP0 的连接错误		否
	STATUS_1A	WORD	CPU1 与 UP0 的连接状态		否
	NDR_0B	BOOL	CPU0 与 UP1 的连接请求成功响应和处理		否
	ERROR_0B	BOOL	CPU0 与 UP1 的连接错误		否
	STATUS_0B	WORD	CPU0 与 UP1 的连接状态		否
	NDR_1B	BOOL	CPU1 与 UP1 的连接请求成功响应和处理		否
	ERROR_1B	BOOL	CPU1 与 UP1 的连接错误		否
	STATUS_1B	WORD	CPU1 与 UP1 的连接状态		否

	IDENT_CODE	STRING [18]	预授权解码输出，将此码连同软件序列号发给西门子 IT 部门后可得到授权	否
	RedErrS7	BOOL	S7 冗余丢失	否
	RedErrDev	BOOL	通信伙伴冗余丢失	否
	TotComErr	BOOL	通信完全丢失	否
	Init_Error	BOOL	初始化错误	否
	Init_Status	WORD	初始化状态	否

Table 5：FB917“ MB_PNHSV”管脚参数定义

下载网络组态及程序到 CPU 中，使能参数 ENR=1，在新建打开的两个 Modscan32 窗口的的“ Set up->Data Definition “中设置数据扫描周期、寄存器连接类型、起始地址、长度等，如下图 14 所示：



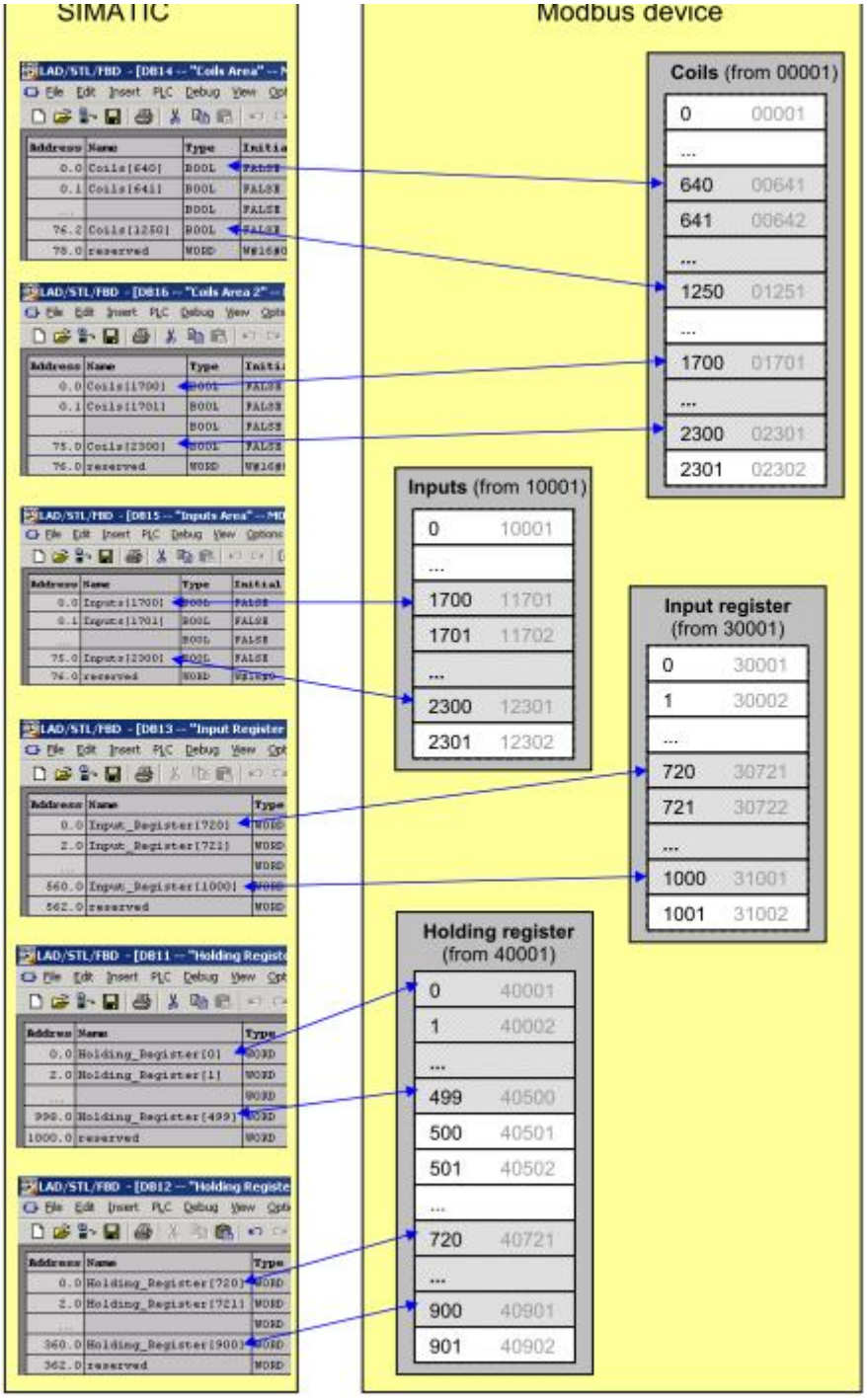
Picture14：Modscan32 中 Modbus 数据参数定义

由于 Modbus 的内部地址编排时基于数据链路层和应用层有一定的映射关系，因此 Modbus 的地址与 SIMATIC 中的 DB 块的地址时按照一定的地址映射关系来相对应，这样造

成了 DB 块中有一定的地址偏移量,在本例中假设数据区的定义如下图 15 所示，其 DB 偏移量、Modbus 物理编址、应用层编址如下图 16 所示：

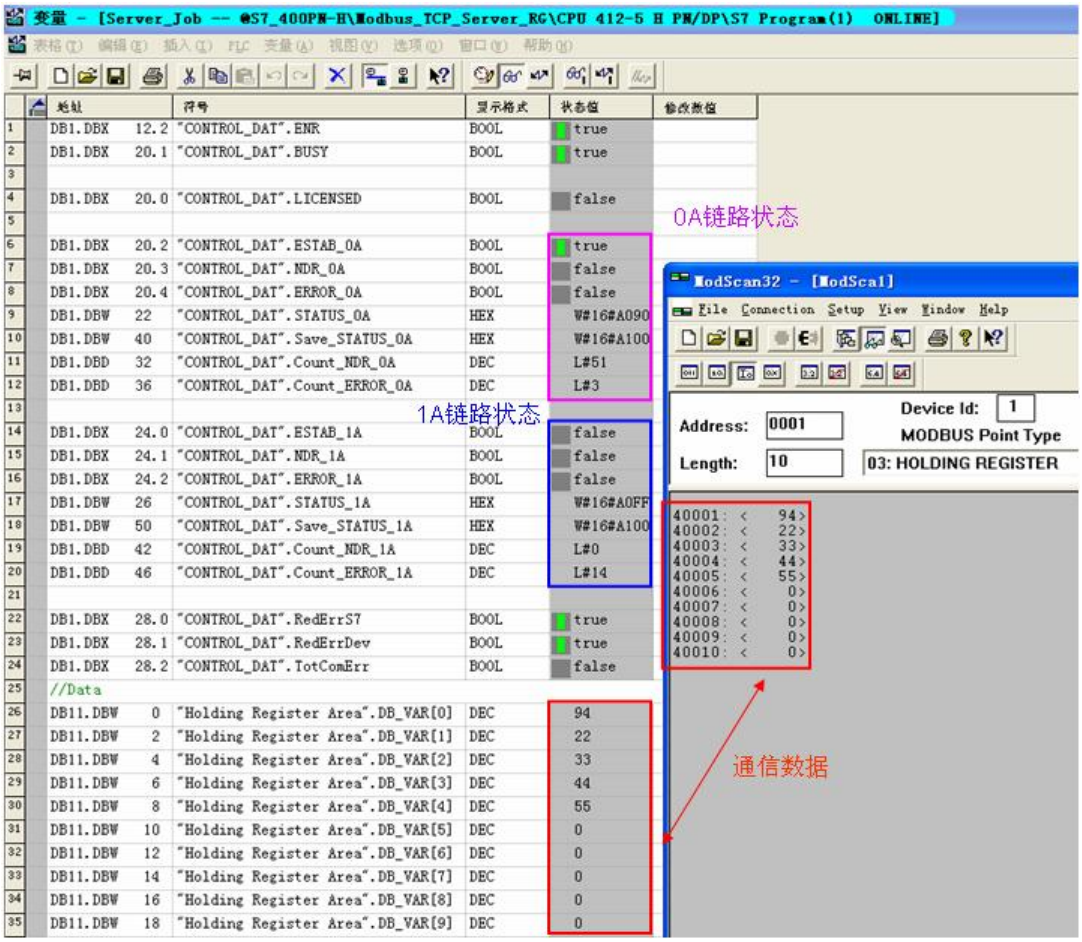
data_type_1	B#16#3	Holding register
db_1	W#16#B	DB 11
start_1	W#16#0	Start address: 0
end_1	W#16#1F3	End address: 499
data_type_2	B#16#3	Holding register
db _2	W#16#C	DB 12
start _2	W#16#2D0	Start address: 720
end _2	W#16#384	End address: 900
data_type_3	B#16#4	Input register
db _3	W#16#D	DB 13
start _3	W#16#2D0	Start address: 720
end _3	W#16#3E8	End address: 1000
data_type_4	B#16#0	Not used
db _4	0	0
start _4	0	0
end _4	0	0
data_type_5	B#16#1	Coils
db _5	W#16#E	DB 14
start _5	W#16#280	Start address: 640
end _5	W#16#4E2	End address: 1250
data_type_6	B#16#2	Inputs
db _6	W#16#F	DB 15
start _6	W#16#6A4	Start address:1700
end _6	W#16#8FC	End address: 2300
data_type_7	B#16#1	Coils
db _7	W#16#10	DB 16
start _7	W#16#6A4	Start address: 1700
end _7	W#16#8FC	End address: 2300
data_type_8	B#16#0	Not used
db _8	0	0
start _8	0	0
end _8	0	0

Picture15：本例中的数据区定义



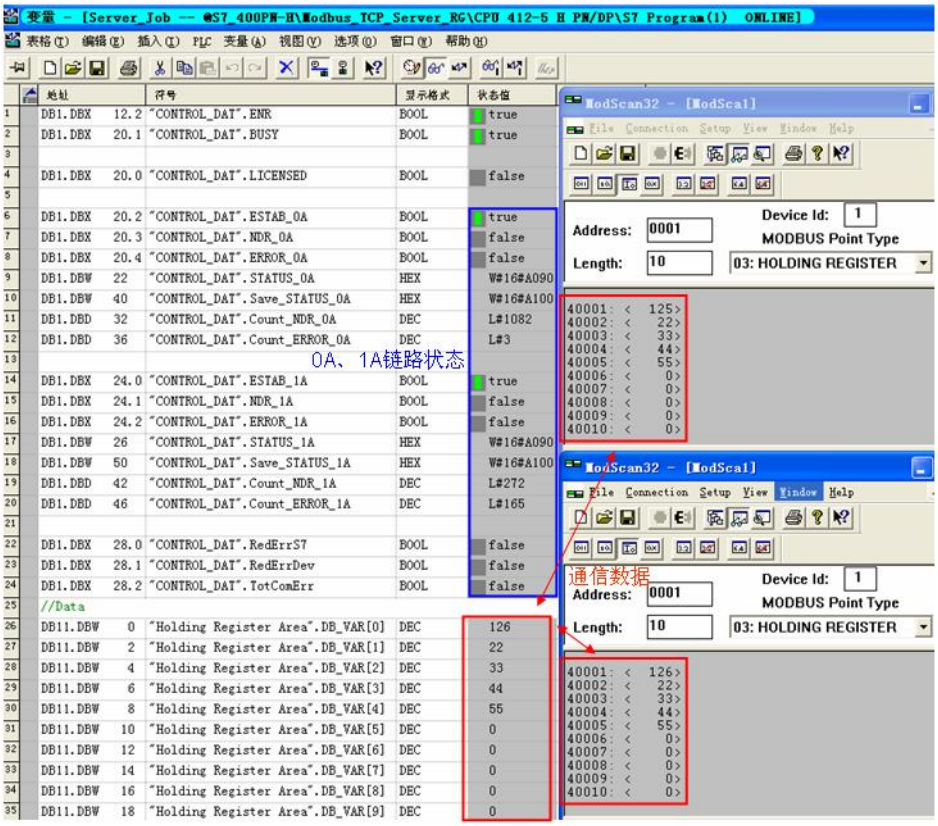
Picture16：DB 偏移量、Modbus 物理编址、应用层编址对应关系

在 Step7 的项目程序中新建一个变量监控表，插入需要监控的参数和数据区变量，可以看到 Modscan32 软件与 S7-400 的数据通讯已经建立起来了，双方可以进行正常的保持寄存器数据读写操作，如下图 17 所示：

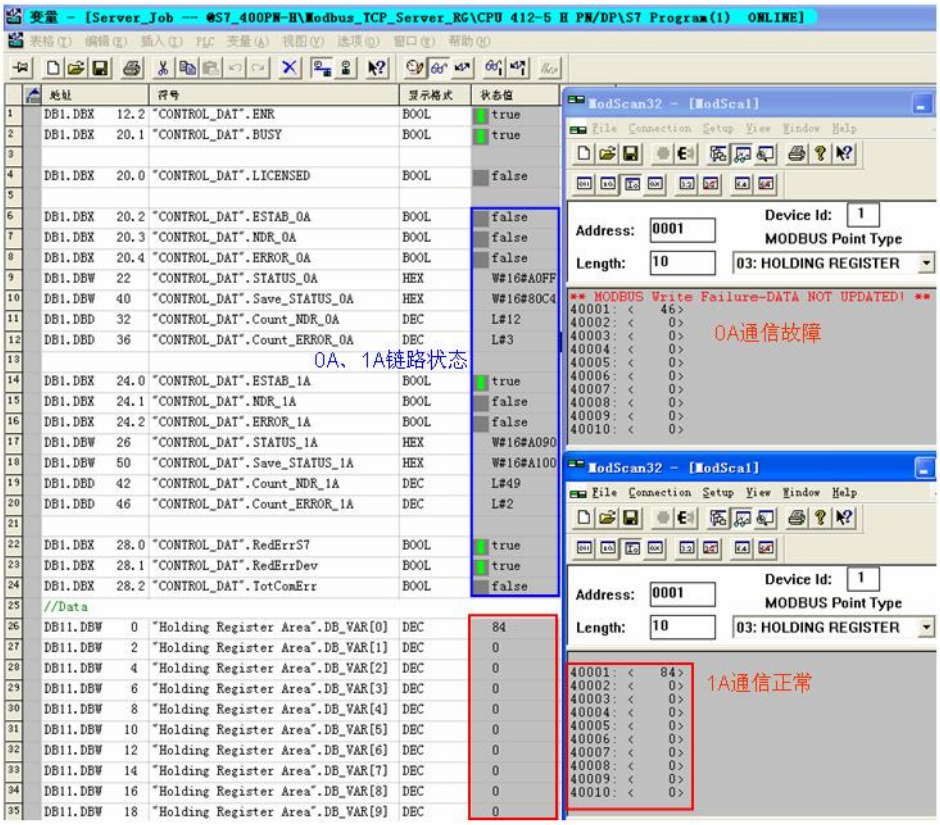


Picture17： 通讯连接建立

下面来看一下链路冗余使用的过程，正常情况下通过任何的一个 Modscan32 窗口(对应 IP192.168.2.10 和 192.168.2.11)均可以与 S7-400H 系统建立通讯，如下图 18 所示，当断开其的一个链路(比如可以拔掉网线或将 CPU 转到 Stop 状态，本例将 IP 为 192.168.2.10 断开)，可以看到 IP 为 192.168.2.11 链路仍保持正常通讯，从而不影响 S7-400H 系统与对方的通讯，另外通过观察各链路连接参数也可监控其连接状态，如下图 18、19 所示：



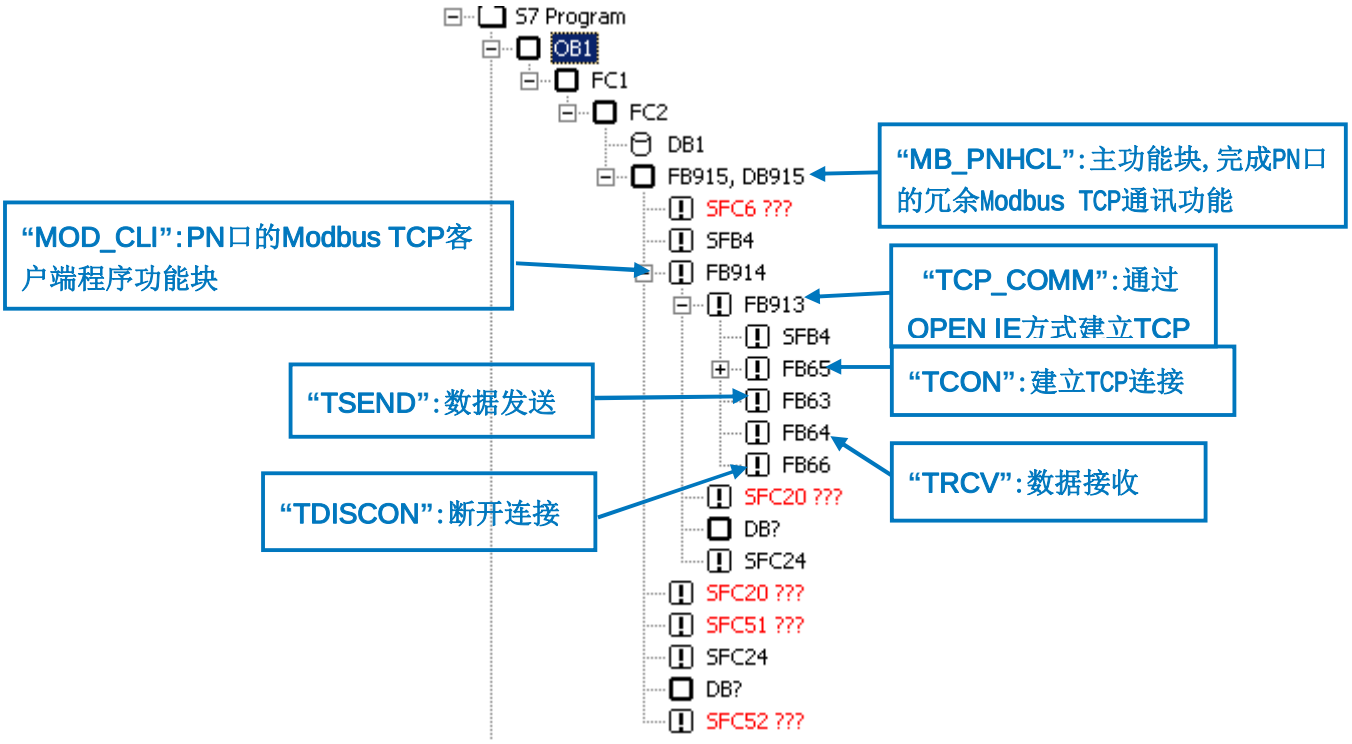
Picture 18： IP 192.168.2.10 及 192.168.2.11 链路连接监控表



Picture 19： 将 IP 192.168.2.10 链路中断后监控表

4 配置 S7-400 PN-H 冗余系统通过 CPU 集成 PN 口作为 Client 进行 Modbus TCP 通讯

下面以 S7-400 PN-H 冗余系统及 Modbus slave 软件为例，详细介绍如何将 S7-400 PN-H 冗余系统通过 CPU 集成 PN 口配置为 Client，Modbus slave 为 Server 进行 Modbus TCP 通讯，下图 20 为服务器功能块库的程序结构及各功能块完成的功能：



Picture 20：软件包“ Modbus/TCP PN CPU Redundant V1.0”客户端程序架构

注：Modslave 软件可以从网上免费下载得到，本例中使用的版本为 V4.3 版，由于各版本的功能不尽相同，因此需要注意版本问题。

4.1 例子中使用的硬件设备及软件

本例中所用的硬件设备如下表：

名称	数量	订货号
S7-400 电源模块 PS 407 10A	2	6ES7407-0KA02-0AA0
S7-400 CPU412-5H PN/DP	2	6ES7412-5HK06-0AB0(V6.0)
S7-400 机架	1	6ES7400-2JA00-0AA0
网线	若干	
笔记本电脑	1	

Table 6：客户端硬件清单

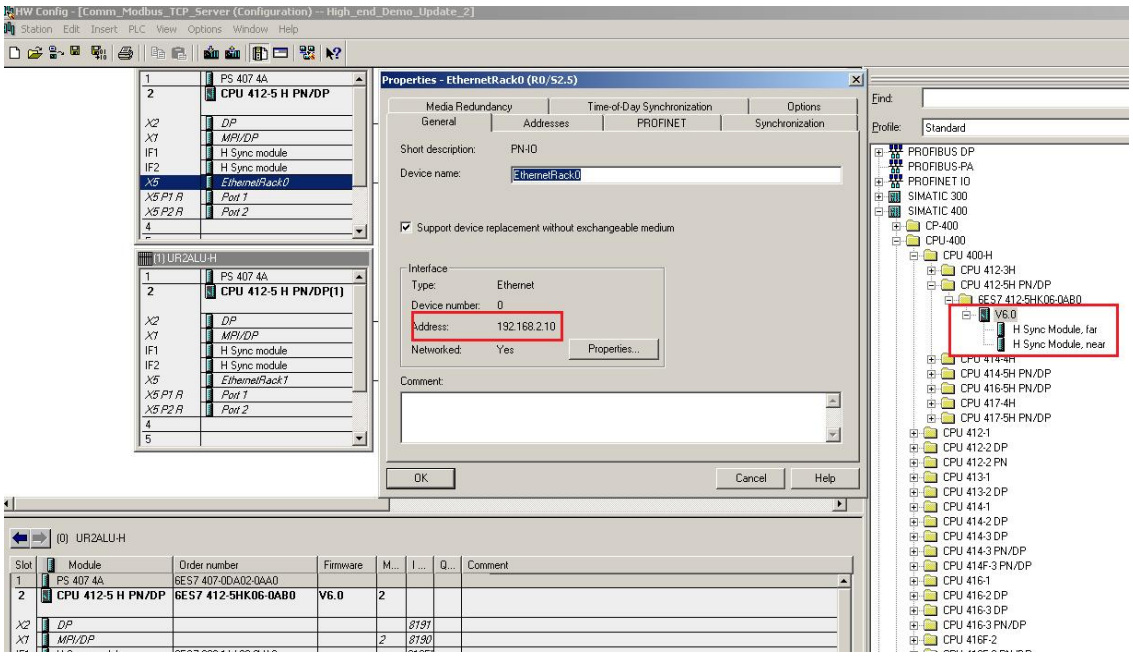
所用到软件如下表：

名称	订货号
STEP7 V5.5 SP3 组态编程软件 英文版	
“ Modbus/TCP PN CPU Redundant V1.0 ” 软件选项包	6AV6676-6MB10-0AX0
Modbus slave V4.3	

Table7 ：客户端软件清单

4.2 S7-400 PN-H 冗余系统及 Modbus slave 软件组态

打开 STEP7 软件， 新建一个项目文件 ，插入一个“ SIMATIC H 站”，在硬件组态中分别插入 PS407 电源，CPU412-5H PN/DP 等并设置 Rack 0、1 CPU 的集成 PN 接口的 IP 地址，如下 21 所示 ：



Picture 21：硬件组态

硬件组态完成后，编译保存，并将例程站点“ H Single-sided (Client) ”中的程序（System data 不需要拷贝）拷贝到该项目中。

由于需要在 SIMATIC 站与其他通讯伙伴之间建立 TCP 连接用于 Modbus 通讯，而对于 CPU 的集成 PN 口来说须通过 Open IE(开放式以太网通讯)的方式来建立 TCP 连接， 通过 S7-CPU 的 PROFINET 接口 进行 Modbus TCP 通信时，需要使用通信块 FB65 "TCON"、FB66 "TDISCON"、FB63 "TSEND" 和 FB64 "TRCV"，要进行 Modbus TCP 通信，必须在数据块中为 Rack0 及 Rack1 冗余 CPU 指定相应的参数，相应得参数在程序中主要由 DB2

“ MODBUS_HPARAM_PN_2”来完成初始化(注意：参数设置必须在“ Data View”视图下的“ Actual value”列中设置)，其中各参数的含义如下图 22、23 所示：

Address	Name	Type	Initial value	Comment
0.0		STRUCT		
+0.0	double_sided_red	BOOL	FALSE	
+2.0	connection_0A	STRUCT		
+0.0	block_length	WORD	W#16#40	#!Verbindung_0001!#
+2.0	id	WORD	W#16#1	
+4.0	connection_type	BYTE	B#16#11	
+5.0	active_est	BOOL	TRUE	
+6.0	local_device_id	BYTE	B#16#5	
+7.0	local_tsap_id_len	BYTE	B#16#0	
+8.0	rem_subnet_id_len	BYTE	B#16#0	
+9.0	rem_staddr_len	BYTE	B#16#4	
+10.0	rem_tsap_id_len	BYTE	B#16#2	
+11.0	next_staddr_len	BYTE	B#16#0	
+12.0	local_tsap_id	ARRAY[1..16]	16 (B#16#0)	
*1.0		BYTE		
+28.0	rem_subnet_id	ARRAY[1..6]	B#16#0, B#16#0,	
*1.0		BYTE		
+34.0	rem_staddr	ARRAY[1..6]	B#16#A, B#16#0,	
*1.0		BYTE		
+40.0	rem_tsap_id	ARRAY[1..16]	B#16#1, B#16#F6	
*1.0		BYTE		
+56.0	next_staddr	ARRAY[1..6]	B#16#0, B#16#0,	
*1.0		BYTE		
+62.0	spare	WORD	W#16#0	#!Verbindung_0001!#
=64.0		END_STRUCT		

Rack0 CPU集成PN口Open IE TCP连接参数

+66.0	connection_1A	STRUCT		
+0.0	block_length	WORD	W#16#40	#!Verbindung_0002!#
+2.0	id	WORD	W#16#2	
+4.0	connection_type	BYTE	B#16#11	
+5.0	active_est	BOOL	TRUE	
+6.0	local_device_id	BYTE	B#16#15	
+7.0	local_tsap_id_len	BYTE	B#16#0	
+8.0	rem_subnet_id_len	BYTE	B#16#0	
+9.0	rem_staddr_len	BYTE	B#16#4	
+10.0	rem_tsap_id_len	BYTE	B#16#2	
+11.0	next_staddr_len	BYTE	B#16#0	
+12.0	local_tsap_id	ARRAY[1..16]	16 (B#16#0)	
*1.0		BYTE		
+28.0	rem_subnet_id	ARRAY[1..6]	B#16#0, B#16#0,	
*1.0		BYTE		
+34.0	rem_staddr	ARRAY[1..6]	B#16#A, B#16#0,	
*1.0		BYTE		
+40.0	rem_tsap_id	ARRAY[1..16]	B#16#1, B#16#F6	
*1.0		BYTE		
+56.0	next_staddr	ARRAY[1..6]	B#16#0, B#16#0,	
*1.0		BYTE		
+62.0	spare	WORD	W#16#0	#!Verbindung_0002!#
=64.0		END_STRUCT		

Rack1 CPU集成PN口Open IE TCP连接参数

Picture22：DB2 “ MODBUS_HPARAM_PN_2”的 TCP 连接参数设置部分

关于 DB2 “ MODBUS_HPARAM_PN_2”的 Rack0 及 Rack1 冗余 CPU TCP 连接参数含义如下表 8 所示：

类型	参数	含义
OPEN IE 通讯参数	double_sided_red	通信伙伴是否为冗余系统，1=冗余，0=单站
	block_length	固定值W#16#40
	Id	连接ID,用于FB63/64/65/66 ,Rack0和Rack1CPU必须唯一
	connection_type	取决于CPU类型，用于FB65(TCON) TCP(兼容模式): CPU315、317<= FWV2.3 W#16#01； TCP:CPU315,317>= FW V2.4、IM151-8PN/DP CPU、 CPU314C、CPU319、CPU412、CPU414与CPU416 W#16#11
	active_est	主动或被动连接: S7作Client时为主动 TRUE S7作Server时为被动 FALSE
	local_device_id	取决于CPU类型： IM151-8PN/DP B#16#1 CPU314C、315、317 B#16#2 CPU319 B#16#3

		CPU412(H)、414(H)、416(H) B#16#5 Rack1中的CPU B#16#15
	local_tsap_id_len	local_device_id的长度： 主动连接时 W#16#0 被动连接时 W#16#2
	rem_subnet_id_len	未使用
	rem_staddr_len	参数rem_staddr的长度： 未具体定义连接 B#16#0 有具体连接 B#16#4
	rem_tsap_id_len	rem_tsap_id的长度： 主动连接时 W#16#2 被动连接时 W#16#0
	next_staddr_len	通讯接口类型选择: 通过外部CP模块： 非0的其它值 通过CPU的集成PN 口： W#16#0
	local_tsap_id	本地连接TSAP号,与参数connection_type有关： 1)connection_type= B#16#01时 local_tsap_id[1] 本地连接端口号的低字节[16进制] local_tsap_id[2] 本地连接端口号的高字节[16进制] local_tsap_id[3-16] B#16#00 2)connection_type= B#16#11时 local_tsap_id[1] 本地连接端口号的高字节[16进制] local_tsap_id[2] 本地连接端口号的低字节[16进制] local_tsap_id[3-16] B#16#00
	rem_subnet_id	未使用
	rem_staddr	通信伙伴的IP地址，与参数connection_type有关，以192.168.0.1为例： 1)connection_type= B#16#01时 rem_staddr[1]= B#16#01(1), rem_staddr[2]= B#16#00(0) rem_staddr[3]= B#16#A8(168) rem_staddr[4]= B#16#C0(192) rem_staddr[5-6]=B#16#00(为IPV6预留)

		2)connection_type= B#16#11时 rem_staddr[1]= B#16#C0(192) rem_staddr[2]= B#16#A8(168) rem_staddr[3]= B#16#00(0) rem_staddr[4]= B#16#01(1) rem_staddr[5-6]=B#16#00(为IPV6预留)
	rem_tsap_id	远程连接TSAP号,与参数connection_type有关： 1)connection_type= B#16#01时 local_tsap_id[1] 本地连接端口号的低字节[16进制] local_tsap_id[2] 本地连接端口号的高字节[16进制] local_tsap_id[3-16] B#16#00 2)connection_type= B#16#11时 local_tsap_id[1] 本地连接端口号的高字节[16进制] local_tsap_id[2] 本地连接端口号的低字节[16进制] local_tsap_id[3-16] B#16#00
	next_staddr	CP的机架号和槽号，当使用CPU的PN口时为 B#16#00

Table 8： DB2 “ MODBUS_HPARAM_PN_2”的 TCP 连接参数含义

+130.0	server_client	BOOL	TRUE
+130.1	single_write	BOOL	FALSE
+130.2	connect_at_startup	BOOL	FALSE
+131.0	reserved	BYTE	B#16#0
+132.0	data_type_1	BYTE	B#16#3
+134.0	db_1	WORD	W#16#B
+136.0	start_1	WORD	W#16#0
+138.0	end_1	WORD	W#16#1F3
+140.0	data_type_2	BYTE	B#16#3
+142.0	db_2	WORD	W#16#C
+144.0	start_2	WORD	W#16#2D0
+146.0	end_2	WORD	W#16#384
+148.0	data_type_3	BYTE	B#16#4
+150.0	db_3	WORD	W#16#D
+152.0	start_3	WORD	W#16#2D0
+154.0	end_3	WORD	W#16#3E8
+156.0	data_type_4	BYTE	B#16#0
+158.0	db_4	WORD	W#16#0
+160.0	start_4	WORD	W#16#0
+162.0	end_4	WORD	W#16#0
+164.0	data_type_5	BYTE	B#16#1
+166.0	db_5	WORD	W#16#E
+168.0	start_5	WORD	W#16#280
+170.0	end_5	WORD	W#16#4E2
+172.0	data_type_6	BYTE	B#16#2
+174.0	db_6	WORD	W#16#F
+176.0	start_6	WORD	W#16#6A4
+178.0	end_6	WORD	W#16#8FC
+180.0	data_type_7	BYTE	B#16#1
+182.0	db_7	WORD	W#16#10
+184.0	start_7	WORD	W#16#6A4
+186.0	end_7	WORD	W#16#8FC
+188.0	data_type_8	BYTE	B#16#0
+190.0	db_8	WORD	W#16#0
+192.0	start_8	WORD	W#16#0
+194.0	end_8	WORD	W#16#0
+196.0	conn_0A_send_buffer	ARRAY[1..4	260 (B#16#0)
*1.0		BYTE	
+456.0	conn_0A_recv_buffer	ARRAY[1..4	260 (B#16#0)

客户端/服务器选择

与功能码相关, 单写模式

建立连接模式(ENQ_ENR/PLC启动后)选择

可定义8个数据区，支持功能码1、2、3、4、5、6、15、16

IN：含义如下

Data_type_x:预定义的 Modbus数据类型

Identifier	Data type	Size
0	Area not used	
1	Coils	Bit
2	Inputs	Bit
3	Holding Register	Word
4	Input Register	Word

db_x: 数据块号

start_x: modbus寄存器或比特值起始地址, 对应DB从0字节开始

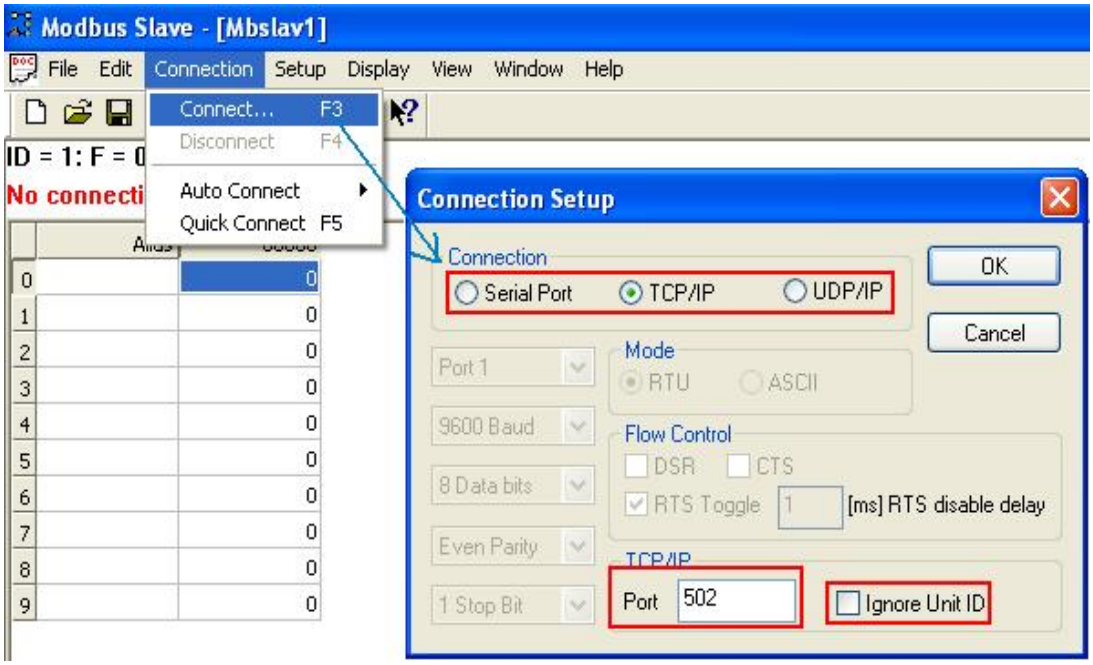
End_x: modbus寄存器或比特值结束地址

消息内部存储区

接收数据存储区

Picture 23：DB2 “ MODBUS_HPARAM_PN_2”的 Modbus 参数设置部分

由于有 S7-400H 系统有 2 条链路(对应两个 CPU)与 Modbus Slave 软件服务器端模拟通讯，打开 Modbus Slave 软件，在 Connection-connect 中打开连接属性对话框，连接接口选择“TCP/IP”，TCP/IP Server Port 为分别为本地服务器的端口 502，并可以勾选“Ignore Unit ID”选项，如下图 24 所示：



Picture 24： Modbus Slave 连接窗口

（说明-“**Ignore Unit ID**”选项的含义如下：

Ignore Unit ID-在一些厂商的 PLC 的程序或网关中可能会用到 Unit ID 以指定处理类型）

4.3 通讯测试

由于“ Modbus/TCP PN CPU Redundant V1.0”选项包支持功能码FC1，2，3，4，5，6，15，16，不同的功能码测试过程中类似，因此下面以FC03(读写保持寄存器)为例来说明通讯测试的整个过程，对于其他功能码的测试将不再重复描述。

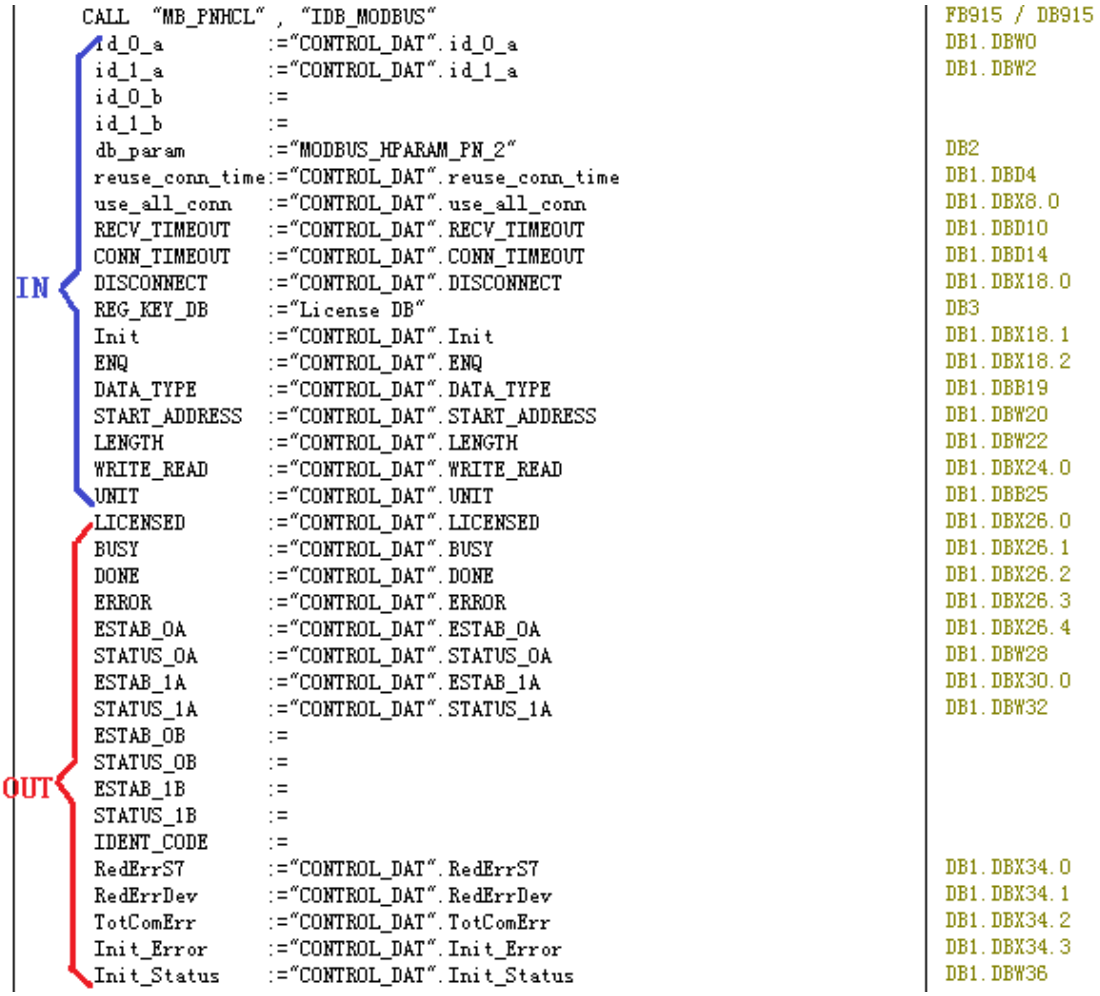
需要说明的是由于客户端功能块需要定义具体的功能码，而主功能块FB915“MB_PNHCL”并没有直接的管脚来定义功能码，而是由其中的两个参数“DATA_TYPE”和“single-write”共同决定(参见后面的管脚参数说明)，详细情况如下图 25 所示：

Data type	DATA_ TYPE	Function	Length	single_ write	Function code
Coils	1	Read	Any	Irrelevant	1
Coils	1	Write	1	TRUE	5
Coils	1	Write	1	FALSE	15
Coils	1	Write	>1	Irrelevant	15
Inputs	2	Read	Any	Irrelevant	2
Holding register	3	Read	Any	Irrelevant	3
Holding register	3	Write	1	TRUE	6
Holding register	3	Write	1	FALSE	16
Holding register	3	Write	>1	Irrelevant	16
Input register	4	Read	Any	Irrelevant	4

Picture 25：S7-400 PN-H 做客户端时不同的功能码的参数定义

在测试过程中我们同样将重点关注通讯连接的建立和当一个链路中断时自动切换到另一个链路的过程。

由于客户端主功能块 FB915“ MB_PNHCL”的参数需要初始化，因此分别在 OB100 及 OB1 中调用 FB915，在 OB100 中调用 FB915 完成相关参数的初始化，FB915 的管脚分布如下图 26 所示：



Picture 26： 功能块 FB915 “ MB_PNHCL”管脚分布

FB915 “ MB_PNHCL”的各参数含义如下表 9：

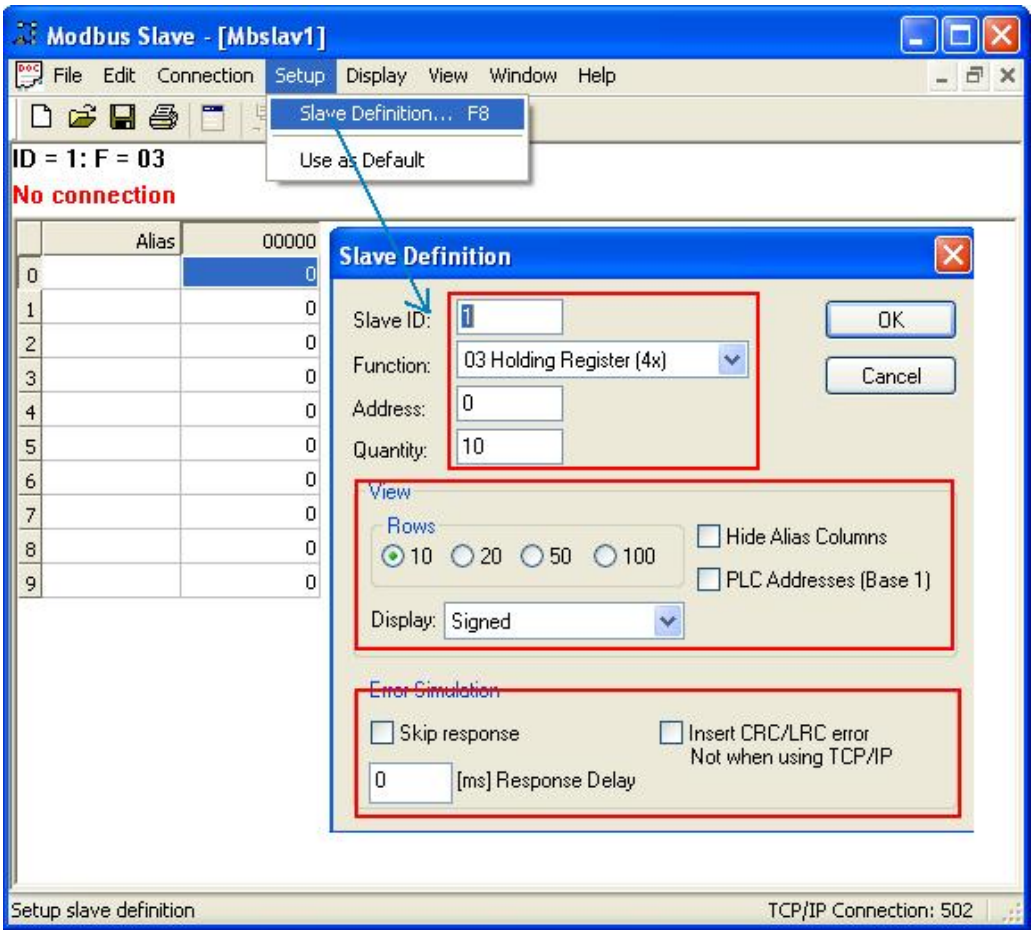
类型	参数	格式	含义		初始化
IN	id_0_a	WORD	假定两个 Rack CPU 简称为 CPU0、CPU1(下同)，如通讯伙伴也为冗余的话简称为 UP0、UP1	CPU0 与 UP0 的连接 ID	是
	id_1_a	WORD		CPU1 与 UP0 的连接 ID	是
	id_0_b	WORD		CPU0 与 UP1 的连接 ID	是
	id_1_b	WORD		CPU1 与 UP1 的连接 ID	是
	db_param	BLOCK_DB	参数化 DB 块		是

	reuse_conn_time	TIME	一个链路故障后尝试重新建立连接的间隔	是
	Use_all_conn	Bool	0=客户端通过一个链路发送报文 1=客户端通过所有组态链路发送报文	是
	RECV_TIMEOUT	TIME	监视应用层接收数据的超时时间，最少 20ms	否
	CONN_TIMEOUT	TIME	TCP 连接建立超时监控时间，最短 100ms	否
	DISCONNECT	BOOL	TRUE 时断开远程伙伴的连接	否
	REG_KEY_DB	BLOCK_DB	授权 DB 块	否
	Init	BOOL	手动初始化	否
	ENQ	BOOL	发送报文使能	否
	DATA_TYPE	BYTE	请求的数据类型	否
	START_ADDRESS	WORD	请求的 Modbus 偏移量	否
	LENGTH	WORD	请求的 modbus 长度	否
	WRITE_READ	BOOL	0=读数据请求 1=写数据请求	否
	UNIT	BYTE	modbus 单元从站地址	
OUT	LICENSED	BOOL	功能块是否授权	否
	BUSY	BOOL	作业正在处理	否
	DONE	BOOL	至少一个连接的报文正常处理	否
	ERROR	BOOL	所有组态的连接通信故障	否
	ESTAB_0A	BOOL	连接 0A 建立	否
	STATUS_0A	WORD	连接 0A 状态	否
	ESTAB_0A	BOOL	连接 1A 建立	否
	STATUS_0A	WORD	连接 1A 状态	否
	ESTAB_0B	BOOL	连接 0B 建立	否
	STATUS_0B	WORD	连接 0B 状态	否
	ESTAB_0B	BOOL	连接 1B 建立	否
	STATUS_0B	WORD	连接 1B 状态	否

	IDENT_CODE	STRING [18]	预授权解码输出，将此码连同软件序列号发给西门子 IT 部门后可得到授权	否
	RedErrS7	BOOL	S7 冗余丢失	否
	RedErrDev	BOOL	通信伙伴冗余丢失	否
	TotComErr	BOOL	通信完全丢失	否
	Init_Error	BOOL	初始化错误	否
	Init_Status	WORD	初始化状态	否

Table 9：FB915 “ MB_PNHCL”管脚参数定义

下载网络组态及程序到 CPU 中，给参数 ENQ 发送脉冲信号，在打开的 Modbus Slave 软件窗口的“Set up->Slave Definition“中设置、寄存器连接类型、起始地址、长度、显示的列数、数据显示格式及响应时间等，并可勾选“Hide Alias Columns”、“PLC Adresses(Base1)”、“Insert CRC/LRC error”、“Skip response”，如下图 27 所示：



Picture 27： Modbus Slave 中 Modbus 数据参数定义

(说明-各勾选选项的含义如下：

Hide Alias Columns –隐藏注释选项

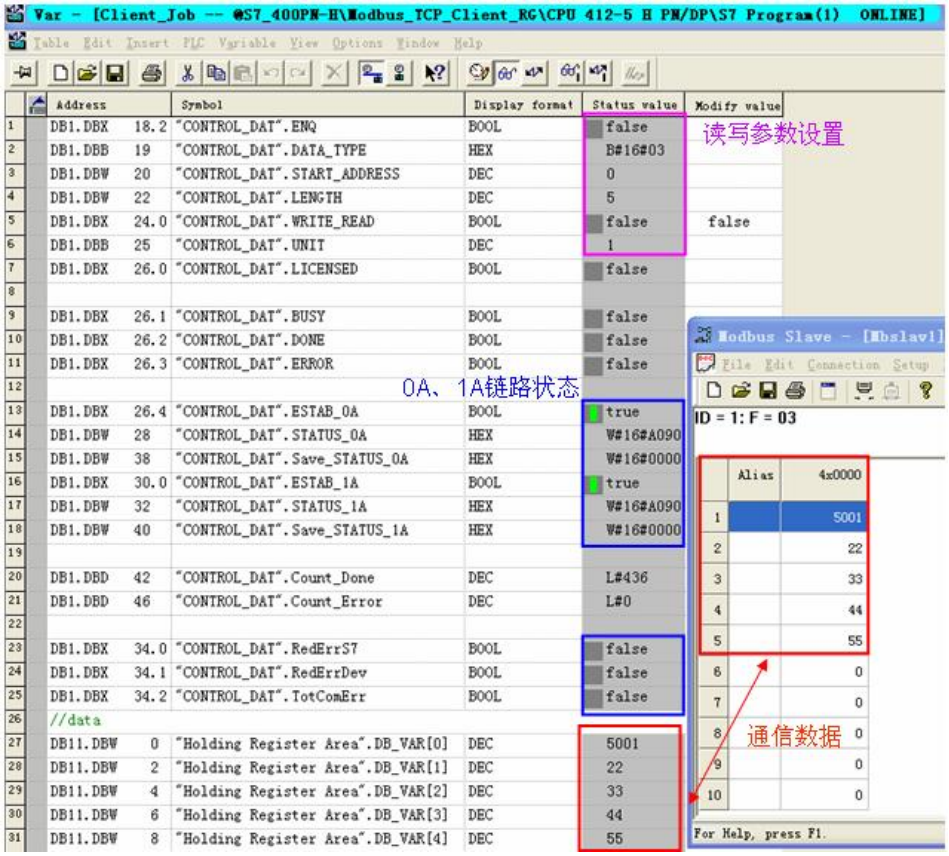
PLC Addresses(Base1) **-** 选择寄存器地址是基于 PLC 地址编排(1..65535)还是基于协议编排(0-65535)

Insert CRC/LRC error - 选择是否进行 CRC/LRC 错误校验

Skip response – 选择是否忽略报文丢失响应)

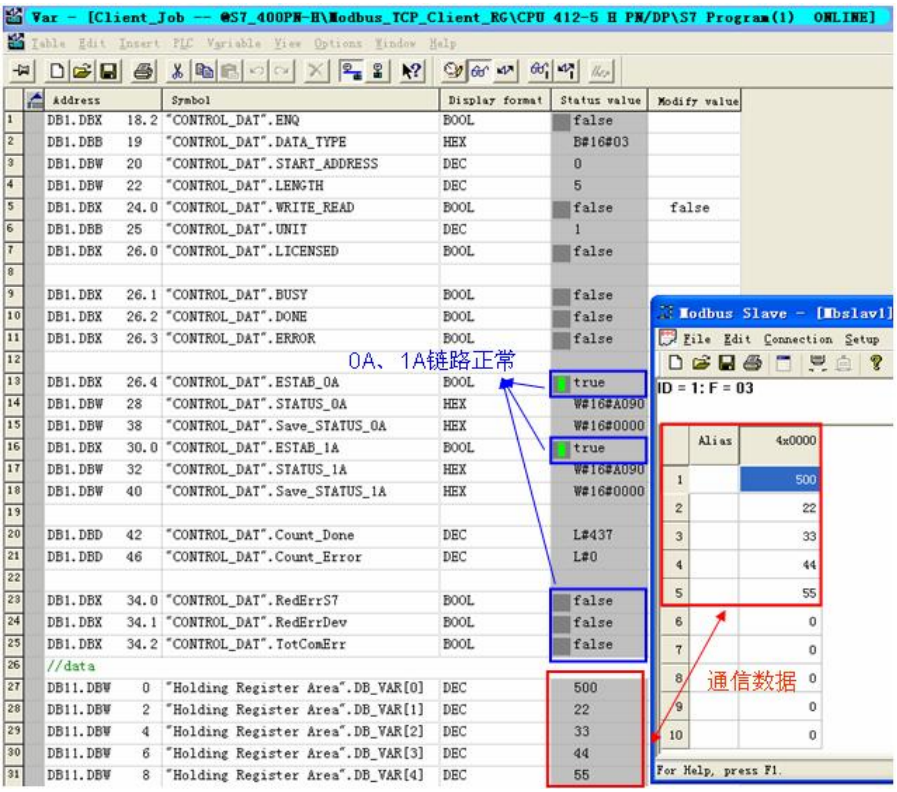
关于 SIMATIC 中 DB 偏移量、Modbus 物理编址、应用层编址对应关系请参考本文中 V3.3 章节说明

在 Step7 的项目程序中新建一个变量监控表，插入需要监控的参数和数据区变量，可以看到 Modsbus Slave 软件与 S7-400H 的数据通讯已经建立起来了，双方可以进行正常的保持寄存器数据读写操作，如下图 28 所示：

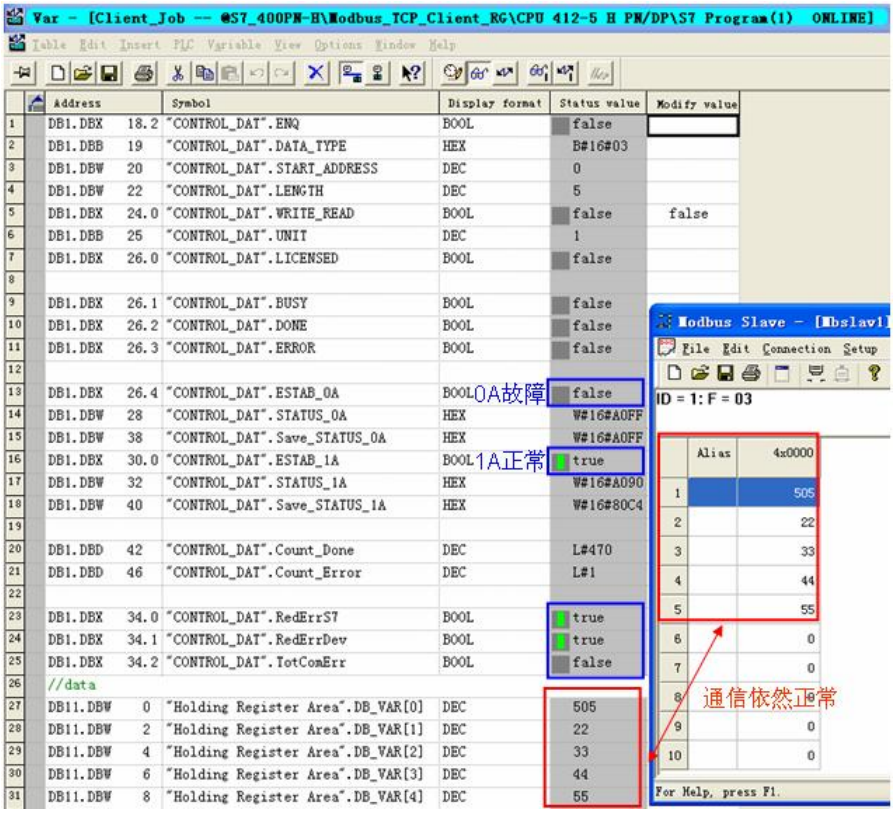


Picture 28：S7-400PN-H 作为客户端与 Modbus Slave 软件通讯

下面看一下链路冗余使用及中断后自动切换的过程，在 Mosbus Slave 软件窗口中，假设其中的一个 Mosbus Slave 通过端口号为 502 与作为 IP 地址为 192.168.2.10 的 CPU 集成 PN 接口通讯，当断开该链路(比如可以拔掉网线、将正在通讯的 CPU 转到 Stop)可以看到功能块将自动切换到另一个链路(Modbus Slave 和 IP 地址为 192.168.2.11 的 CPU 集成 PN 接口)进行通讯，通过观察各链路连接状态参数也可观察得到，如下图 29、30 所示：



Picture29：两个链路均正常情况下按照缺省链路取值



Picture30：:链路中断后切换到另一个链路通讯

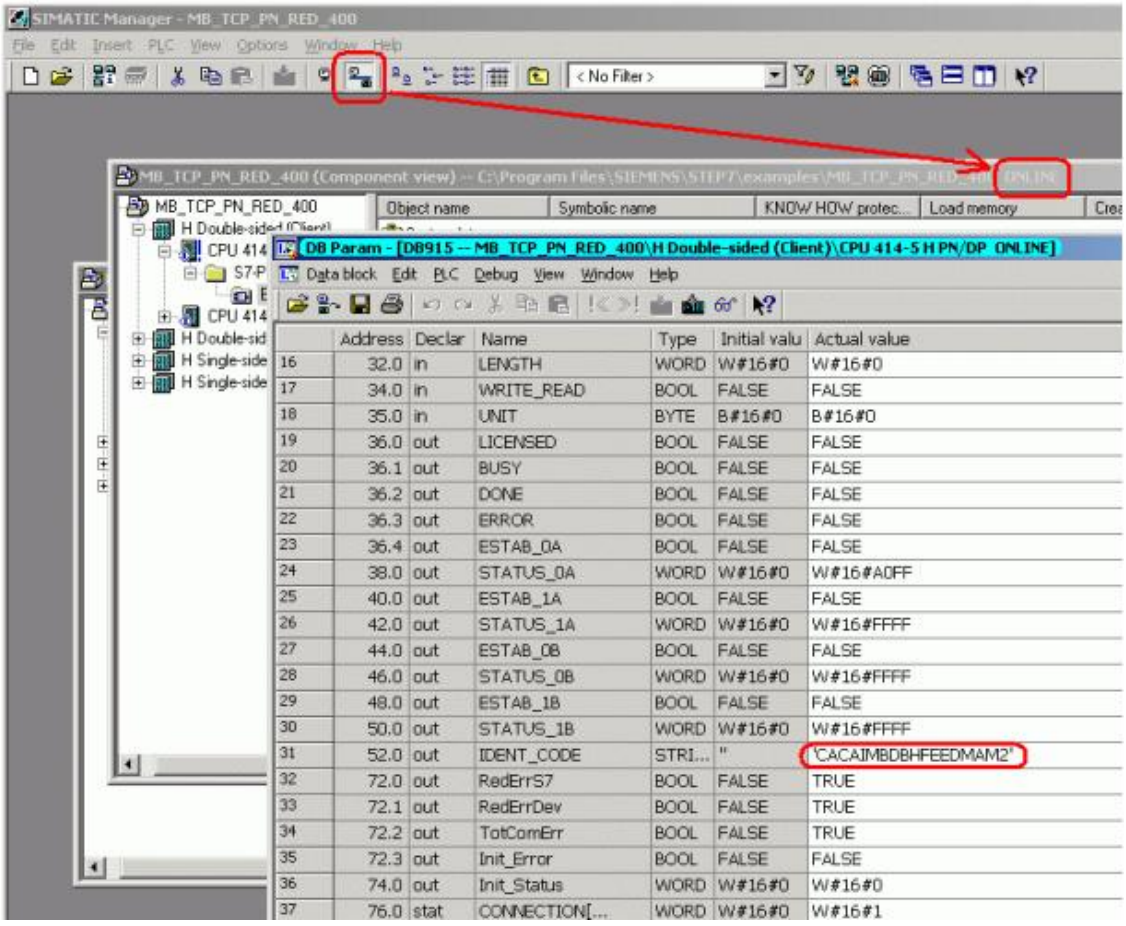
5 软件包“ Modbus/TCP PN CPU Redundant V1.0”授权

每个 CPU 都需要对功能块进行授权，**对于 PN-H 冗余系统 CPU 来说，将只对 Rack0 的 CPU 进行授权验证**，授权有两个步骤：读取 IDENT_CODE 和申请注册码 REG_KEY，且在 CPU 中必须调用 OB121，下面以客户端程序块为例来说明授权步骤。

5.1 读取 IDENT_CODE

1、下载程序并将 CPU 切换到 RUN 模式；

2、打开功能块 FB915“ MB_PNHCL”背景块 DB915，确认 IDENT_CODE 的偏移地址为 52；如图 31 所示：



Picture 31： 确认 IDENT_CODE 的偏移地址

3、监视 DB915.DBB52 开始的 20 个字节，偏移地址 52 开始的 18 个字符即为 IDENT_CODE，监控如图 32 所示：

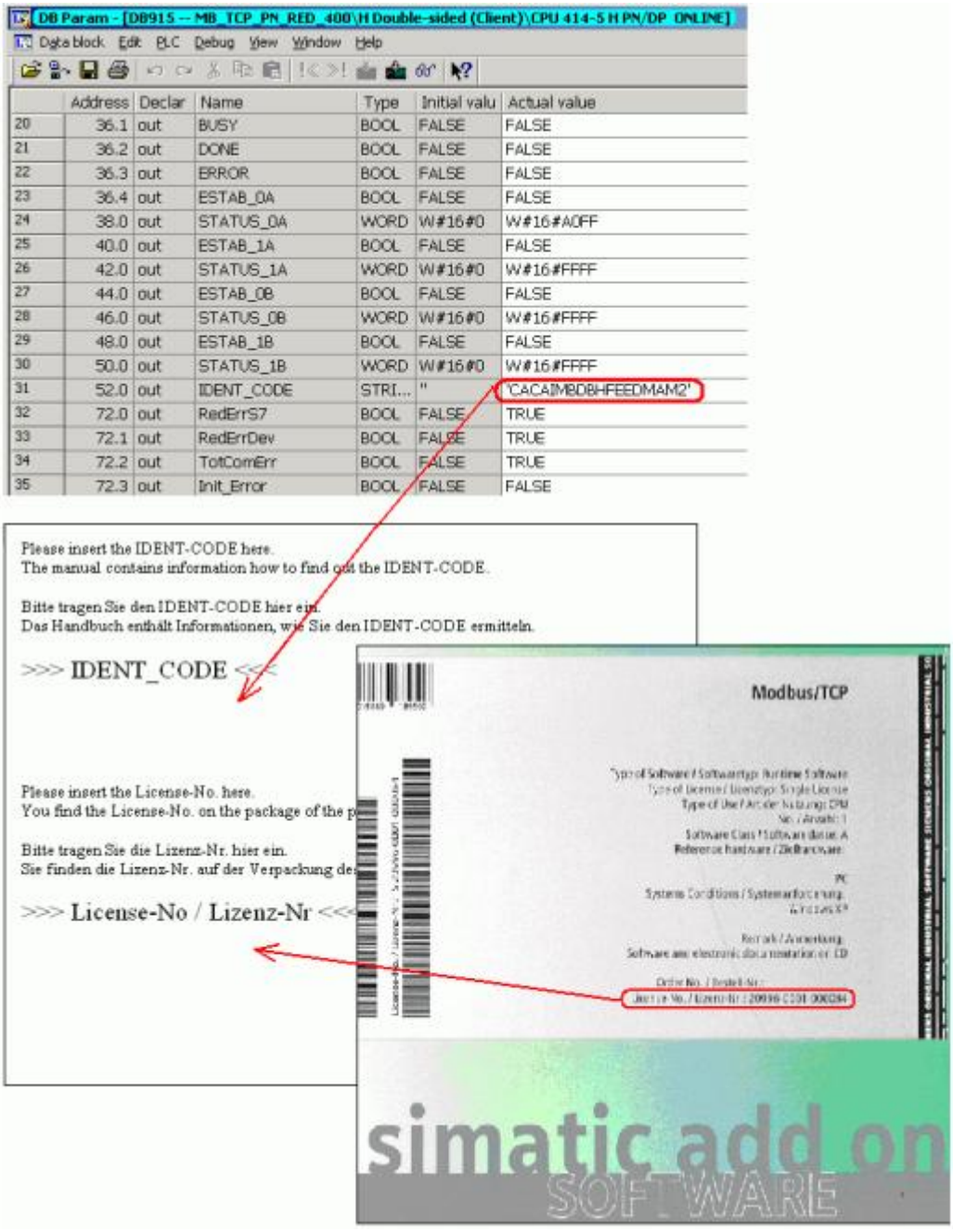


图 34: 确认 IDENT_CODE

4、按上图方式，获取 IDENT_CODE 和软件包装上的 License-No，并按照章节 5.2 和 5.3 的描述步骤申请注册码。

5.2 通过拨打西门子授权服务中心申请注册码 REG_KEY

授权中心联系方式：010-64757575

通过西门子授权服务中心申请注册码时，需要您提供所购买的软件订货号、IDENT_CODE 和软件包装上的 License-No ，如图 34 所示。

5.3 通过网站申请注册码 REG_KEY

1、通过西门子技术支持网站申请，打开如下网址，点击“技术问题提交”：

http://support.automation.siemens.com/CN/llisapi.dll?func=cslib.csinfo2&aktprim=99&lang=zh

图 35: 技术支持网站

2、请按如下示例的步骤进行操作（注意：由于步骤 3 搜索出来的参考信息无法解决授权问题，请直接点击“继续”进入步骤 4），如图 36~40 所示。

图 36: 步骤 1

图 37: 步骤 2

图 38: 步骤 4

图 39: 步骤 5

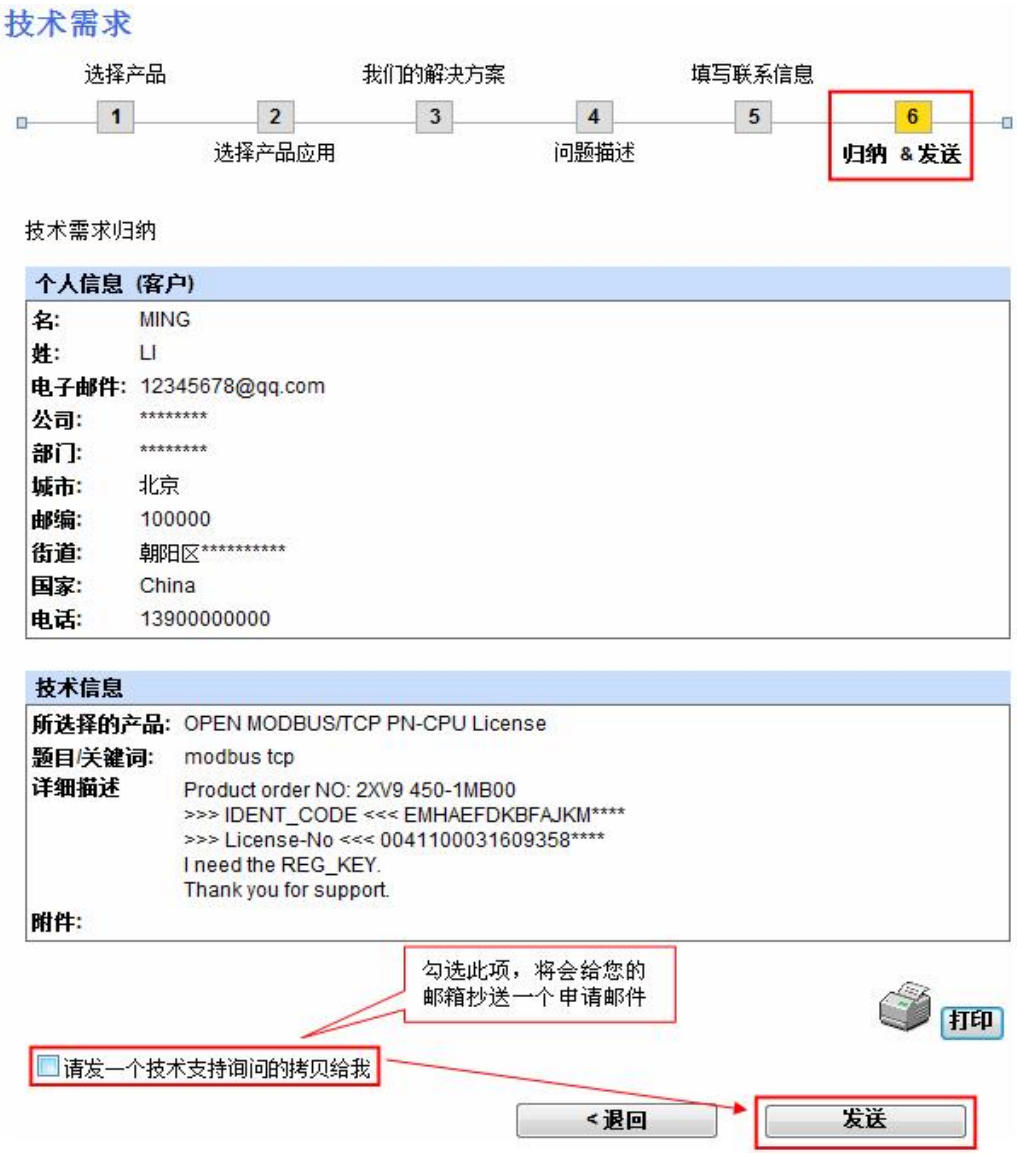


图 40: 步骤 6

5.4 使用注册码 REG_KEY

1、西门子授权中心收到技术支持申请后，将会尽快给您回复邮件；

2、当获取到注册码后，在项目中打开 LICENSE_DB（DB3）；

3、通过菜单“ View--->Data View”将 DB 块切换到数据视图模式，将获取的 17 位注册码填写到“ Actual value”中，如图 41 所示。

Address	Name	Type	Initial value	Actual value	Comment
0.0	REG_KEY	STRING [17]	'insert REG_KEY'	'insert REG_KEY'	Registration Key

Address	Name	Type	Initial value	Actual value	Comment
0.0	REG_KEY	STRING[17]	'insert REG_KEY'	'QODGHNBVWSFJZXNHN'	Registration Key

图 41: 输入注册码

4、将 LICENSE_DB（DB3）下载到 CPU 中，并可通过查看 FB915 “ MB_PNHCL”的输出引脚 LICENSED 为 true 则注册码激活成功。