

问题：如何实现 400H 冗余控制器与 S7-200 通讯？

解答：S7-400 H 系统和 S7-200 之间的通讯可以通过两种方法实现. 第一种方案是通过 CP243 实现以太网的通讯,第二种方案是通过 Y-LINK 和 EM277，实现基于 DP 总线通讯。

一、S7-400H 与 S7-200 通过以太网通讯

通过在双方建立“未指定”的 S7 连接，在连接中指定对方的 IP 地址和 TSAP 号来实现数据通讯。关于这方面的内容，可以参考编号为 17369594 的 FAQ。

<http://support.automation.siemens.com/CN/view/zh/4000024+17369594>

由于 S7-400H 冗余控制器由 RACK0 和 RACK1 两个子系统组成，每个子系统包括单独的控制器和 CP443-1 以太网通讯模板。如何实现包含两个独立子系统的冗余系统和 S7-200 系统之间的冗余通讯即是本文主要介绍的内容。其实现的基本原理如下：分别建立 S7-200 与 RACK0 和 RACK1 控制器的通讯，双方在交换的数据里发送心跳检测信号。通过心跳信号判断两个连接的通讯状态，决定程序里采用哪个连接的数据，实现了通讯的“冗余”。

下面以一个 414H 控制器与 S7-200(CPU224)为例，介绍一下具体的组态过程。S7-200 即可做 SERVER，又可做 CLIENT，一般建议 S7-200 做 Client。本例中 S7-200 做客户端，400H 做 SERVER，只需在 CLIENT 侧编程即可。

□ 项目的具体配置如下：

- ✓ S7-400H 侧：软件为 STEP7 v5.4 SP3，硬件上需要配备 CP443-1 模板。
- ✓ S7-200 侧：软件为 Micro WIN V4.0, 硬件配 CP243-1 以太网模块。

□ 实现的目的：

- ✓ 400H 的主从控制器分别与 200 建立连接。
- ✓ 每个连接的双方分别交换 8 个字节的数据。
- ✓ S7-400 侧判断两个通讯连接的好坏，程序里使用正常连接的数据。

1. S7-400H 侧的组态

1.1 建立 400H 项目

首先打开 STEP7，建立一个 400H 的项目，插入 CP443-1 模板，要激活 CP443-1 的 IP 协议，因为与 200 侧只支持 IP 地址通讯，不支持 MAC 地址通讯方式。RACK0 和 RACK1 的地址要在同一个网段，本例中 RACK0 侧 IP 地址设定为 192.168.1.10, RACK1 的 IP 设为 192.168.1.11，子网掩码都是：255.255.255.0，

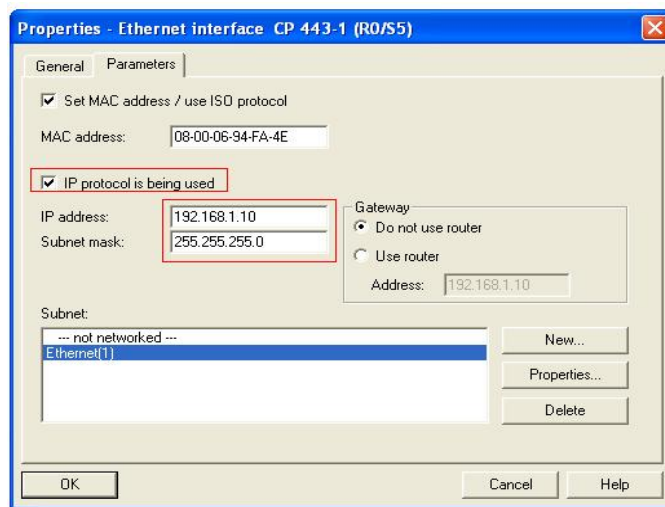



图 1.1 400H 的 RACK0 侧 CP443-1 的属性设置

1.2 组态网络连接

选择菜单中的 ，打开“NetPro”进行网络组态，先选择 RACK0 侧 CPU，鼠标右键单击，在弹出的菜单里选择添加一个新连接，会弹出“插入新连接”的对话框，在这里，我们要选择连接方为“Unspecified”，连接类型选“S7 connection”，如图 1.2 所示：

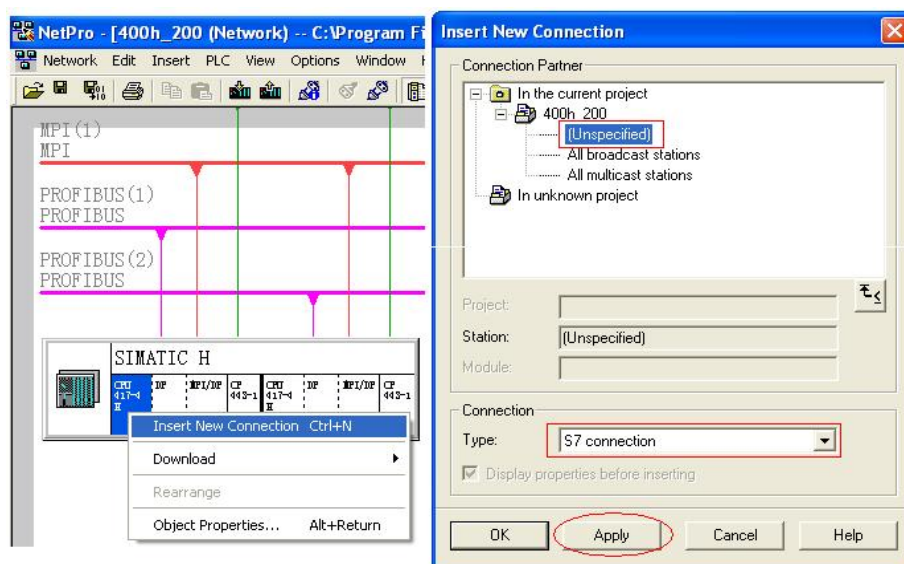


图 1.2 网络组态里，插入一个新连接

1.3 配置连接参数

选择连接类型后，在 1.2 里点击“Apply”按钮，会出现连接的属性对话框，在这里，我们要去掉“Establish an active connection”这一项，因为这个连接是由 S7-200 客户端主动建立的。

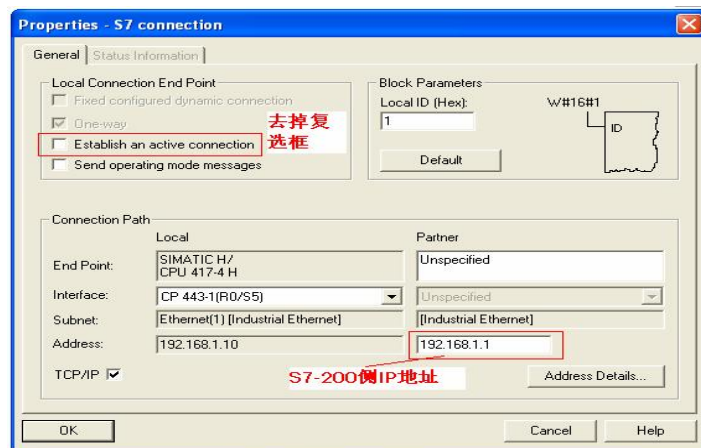


图 1.3 设定 S7 连接的参数

1.4 配置 RACK0 连接 TSAP 号

在图 1.3 中点击“Address Details”按钮，弹出设置连接方的 TSAP 号对话框，本地即 400H 的 RACK0 侧，设定本站的 TSAP 为 10.03，远端 Partner 站的设定要与 S7-200 侧的设定值相对应，该值在 S7-200 侧建立连接时确定，这里我们先设定为“10.00”，需要在组态远端时再次进行确认。

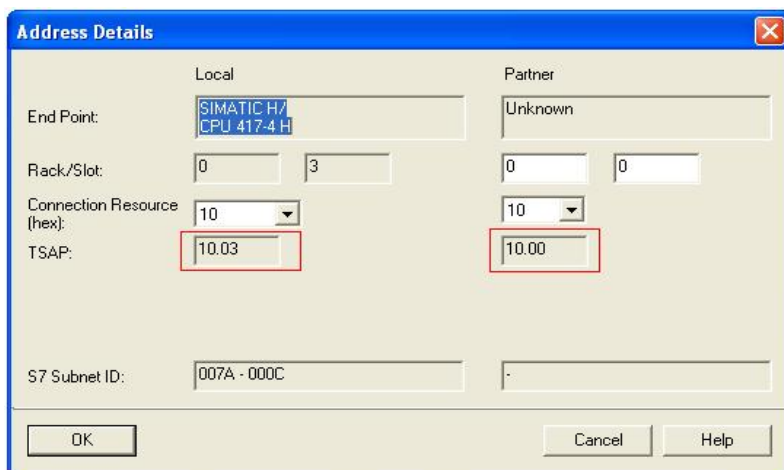


图 1.4 设置 RACK0 CPU 的 TSAP 号

注意：TSAP 号由两个字节组成，第一个字节定义连接号，第二个字节定义了机架号和 CP 槽号，通过它可以指定通讯的数据包由哪个 CPU 来打开，采用什么协议来处理。

1.5 配置 RACK1 连接 TSAP 号

对 RACK1 CPU 进行同样的设置，除 TSAP 号外其它与 RACK0 同，远端的 TSAP 号我们设定 11.00，从下图可以看到，本地侧的 TSAP 号是：11.23。以后要在远端的 S7-200 配置时用到。

The image shows a 'Address Details' dialog box with two main sections: 'Local' and 'Partner'. The 'Local' section has the following fields: 'End Point' (SIMATIC H/CPU 417-4 H(1)), 'Rack/Slot' (1 / 3), 'Connection Resource (hex):' (11), and 'TSAP:' (11.23). The 'Partner' section has: 'End Point' (Unknown), 'Rack/Slot' (0 / 0), 'Connection Resource (hex):' (11), and 'TSAP:' (11.00). Below the 'TSAP:' fields, there are red labels: '本地:400H RACK1' for the local side and '远端:S7-200' for the partner side. At the bottom, there is an 'S7 Subnet ID:' field with the value '007A - 000C'. The dialog box has 'OK', 'Cancel', and 'Help' buttons at the bottom.

Local	Partner
End Point: SIMATIC H/CPU 417-4 H(1)	Unknown
Rack/Slot: 1 / 3	0 / 0
Connection Resource (hex): 11	11
TSAP: 11.23	11.00
本地:400H RACK1	远端:S7-200
S7 Subnet ID: 007A - 000C	-

图 1.5 RACK 1 的连接设定

1.6 编译下载连接

在网络组态窗口里进行全盘编译，然后选择 RACK0 CPU，从连接列表里，选中连接，从鼠标右键弹出的菜单里，选择“Download selected connections”，将连接下装到 RACK0 控制器，如图 1.6 所示。

2. S7-200 侧的通讯组态

2.1 打开 STEP7 Micro/WIN32，新建一个项目，选择好 CPU 的类型为 CPU224，接着选择菜单“TOOLS Ethernet Wizard”，通过向导进行 CP243-1 的组态，如图：2.1 所示。

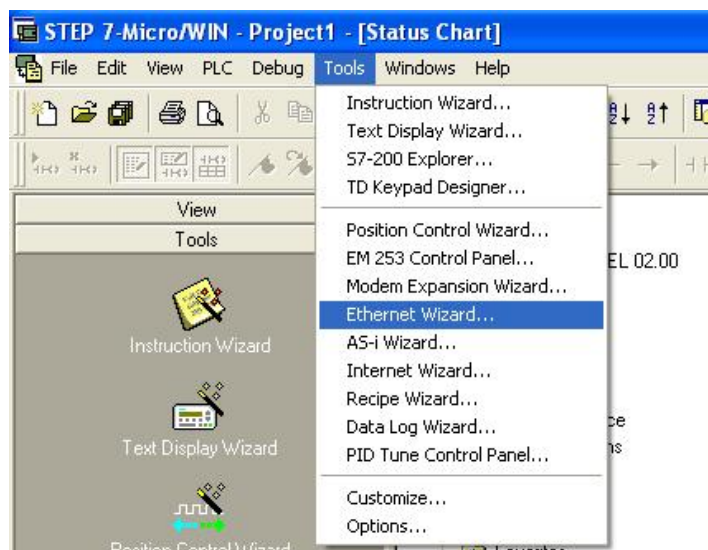


图 2.1 以太网组态向导

2.2 选择 CP243-1 的模板的位置，由于在本例子里，CP243-1 在 CPU 右边的第一个扩展模板的位置，它的槽号是“0”。

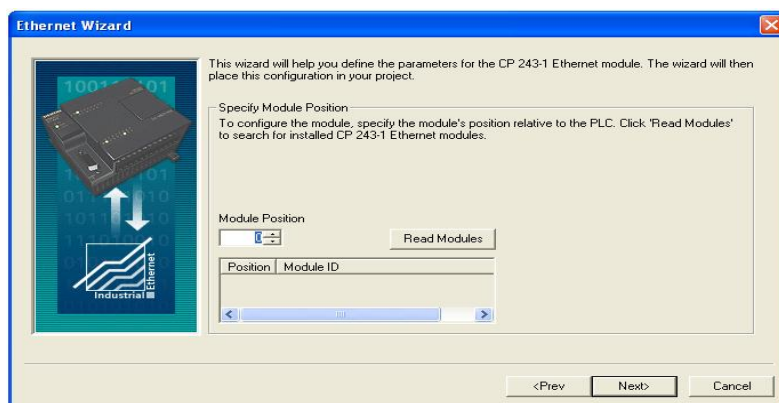


图 2.2 设置槽号

2.3 接下来，设置 CP243-1 模板的 IP 地址：192.168.1.1 子网掩码：255.255.255.0，连接的网络速度选择自适应：“Auto Detect Communications”。

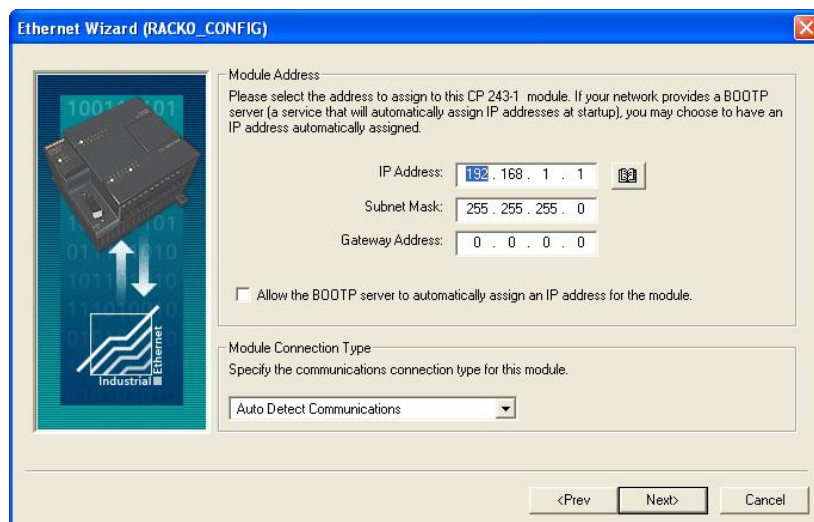


图 2.3 设置地址

2.4 设置 CP243-1 的连接数，CP243-1 最多可以建立 8 个以太网连接，这里需要分别与 400H 的两个 CPU 各建立一个连接，所以连接数应选择“2”。

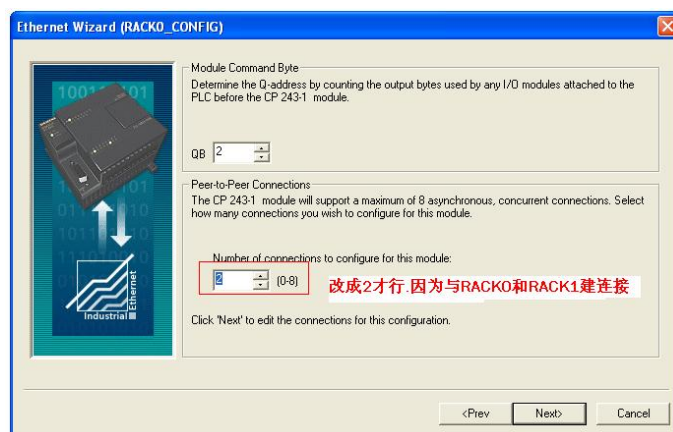


图 2.4 设置连接数

2.5 点击下一步，进入到第一个连接的设置，这里我设置与 RACK0 的连接，如下图所示：将连接名改成“TO-RACK0”，以便于在程序里区别，防止输入错误。这里设定为客户端连接“This is a

client connection”，设定远端伙伴的 TSAP 号，要与前边图 1.4 里本地 RACK0 本地 TSAP 号相同，设定远端 PLC(RACK0 CP443-1)的 IP 地址:192.168.1.10，为了便于识别，将连接符号名改成“TO-RACK0”，同时对照一下在前边配置 400H 侧的 RACK0 连接属性里远端的 TSAP 号是否与这里的一致，如果不一致，从 STEP7 网络组态里改正。

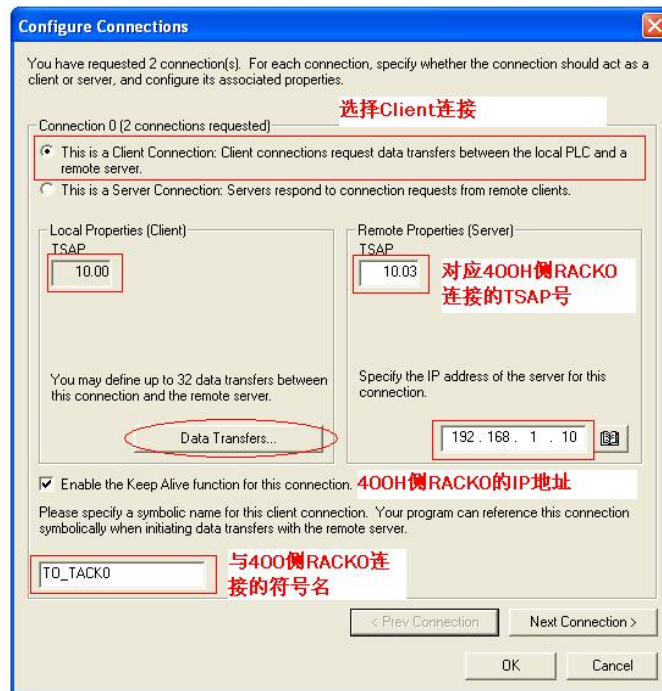


图 2.5 地址设置

2.6 选择“Data Transfers”按钮，会弹出读写的数据连接对话框，如下图所示，将数据传输改成“读”，字节数设成 8，S7-200 侧发送的数据区为 VB0-VB7，S7-400 侧接收数据为 DB1。DBB0-DBB7，本数据传输的符号名改为“read”，符号名要在以后的程序里用到，用于在通讯块里识别传输操作。

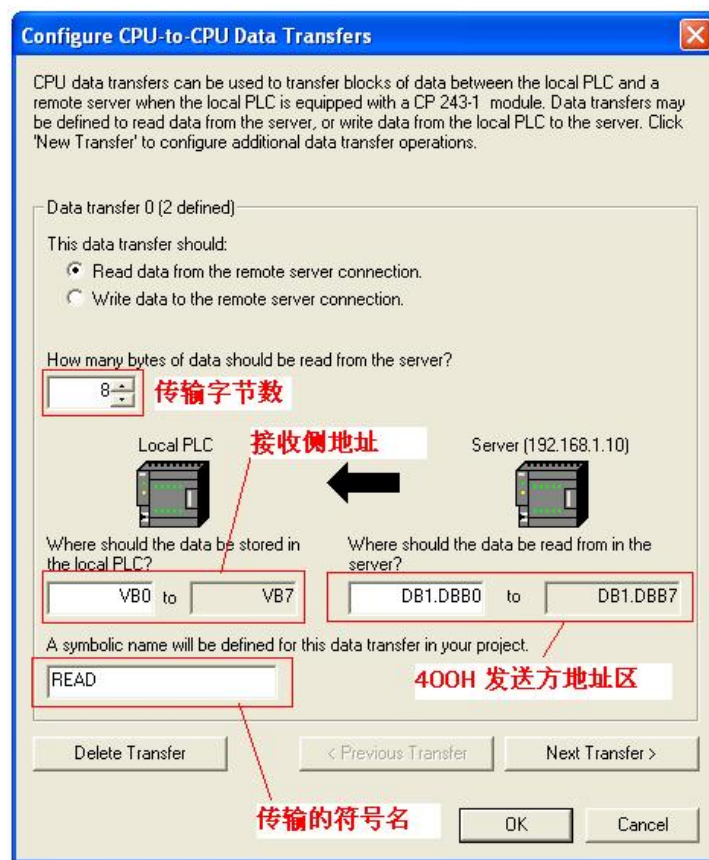


图 2.6 通讯字节设置

2.7 点击 NEXT TRANSFER 按钮，进行 S7-200 向 400 侧 RACK0 CPU 发送数据的配置，如下图所示，将数据传输改成“写”，字节数设成 8，S7-200 侧发送的数据区为 VB10-B17，S7-400 侧接收数据为 DB2。DBB0-DBB7,本数据传输的符号名改为“WRITE”。

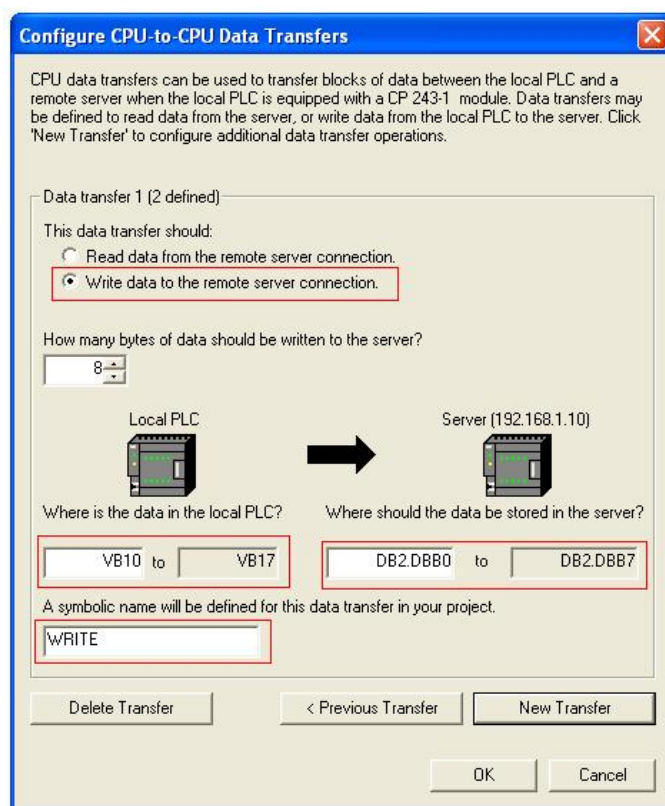


图 2.7 通讯字节设置

2.8 选择 OK 按钮，结束第一个连接，即与 400H 侧 RACK0 的连接配置。

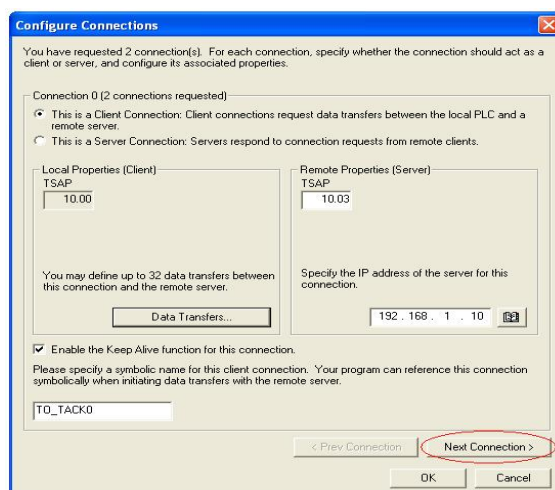


图 2.8 创建另外一个连接

2.9 参考上图，继续进行第二个连接的配置，在这里，远端伙伴的 TSAP 号参照图 1.5 里 RACK1 建立连接时的数值“11.23”，记下本地 PLC 的 TSAP 号，检查 400H 侧的 RACK1 连接属性里远端的 TSAP 号是否与这里的一致，如果不一致，从 STEP7 网络组态里改正。



图 2.9 组态连接

2.10 点击 Data Transfers 按钮，开始进行数据传输的配置，先进行 S7-200 接收数据的配置，由于 400H 侧两个连接传送的数据是同一组数据，所以 SERVER 端的数据区与上次 RACK0 连接相同，但是本地 PLC 接收侧的数据区要分别设置，设置为 VB1000-VB1007。如下图所示：

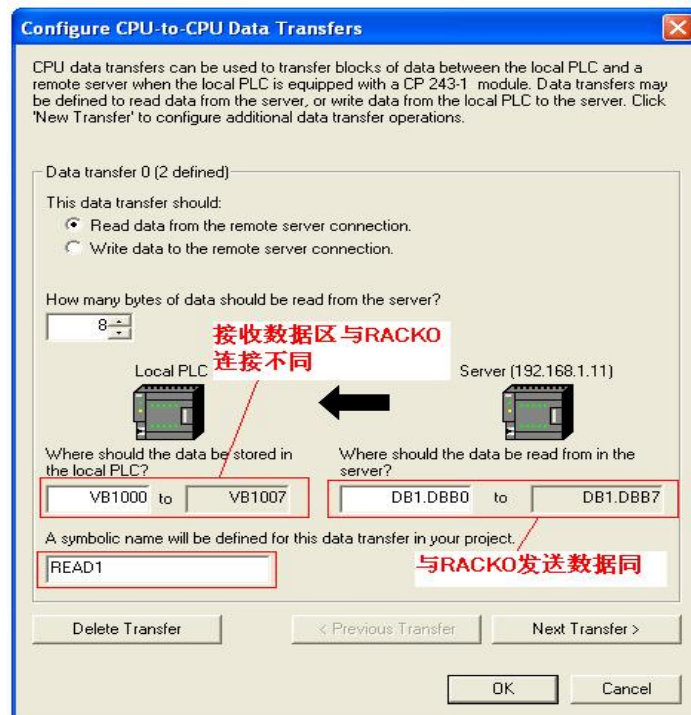


图 2.10 设置地址区间

2.11 进行 S7-200 侧向 S7-400 侧 RACK1 CPU 写数据的配置，在这里，S7-200 发送相同的数据分别给 RACK0 和 RACK1 CPU，故本地 PLC 侧发送的数据区也设置为 VB10-VB17，RACK1 侧的数据接收区设定为 DB12。

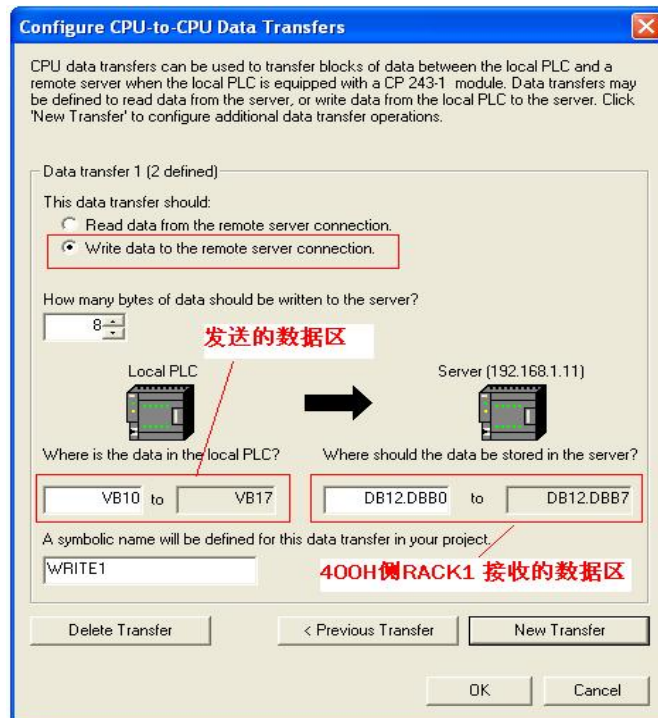


图 2.11 设置地址区间

2.12 点击“OK”，完成与 RACK1 CPU 的通讯配置，点击下一步，填写模块所占用的 V 存储区的起始地址，选择默认值即可。

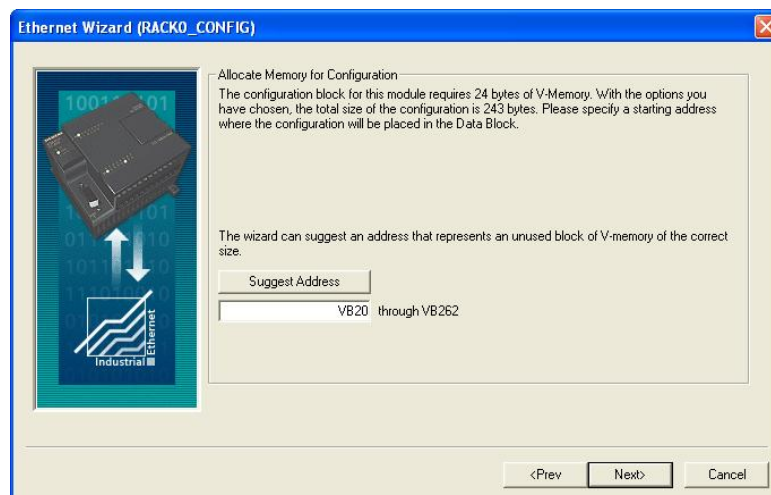


图 2.12 设置地址区间

2.13 上图里点击“下一步”，结束向导，系统会自动生成关于通讯的两个子程序，如下图所示。

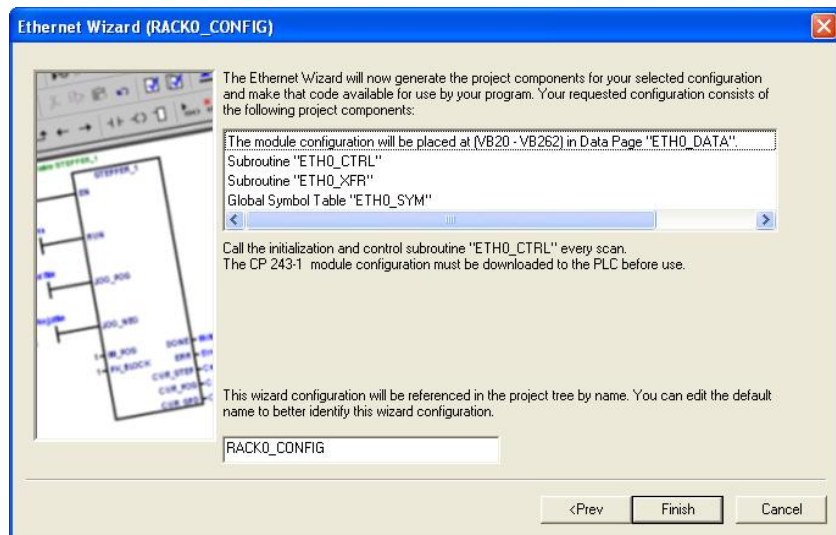


图 2.13 完成组态

3. 通讯双方编程

3.1 在 200 侧，调用前边“连接向导”生成的子程序“ETH0_CTRL”和“ETH0_XFR”，首先添加与 400H 侧 RACK0 通讯的程序，如下图所示，Chan_ID 参数指两 CP 间建立的连接资源，选择前边向导里与 RACK0 建立的连接资源符号名，这里选择 T0_RACK0: Data 指的是两 CP 间定义的传输资源符号名,选择“READ”。在发送与接收的“START”信号的控制上，要使用脉冲信号，而且发送与接收不能同时进行，在下边的例子里，将“读取”数据的结束信号做为“写数据”的启动信号，如下图所示。

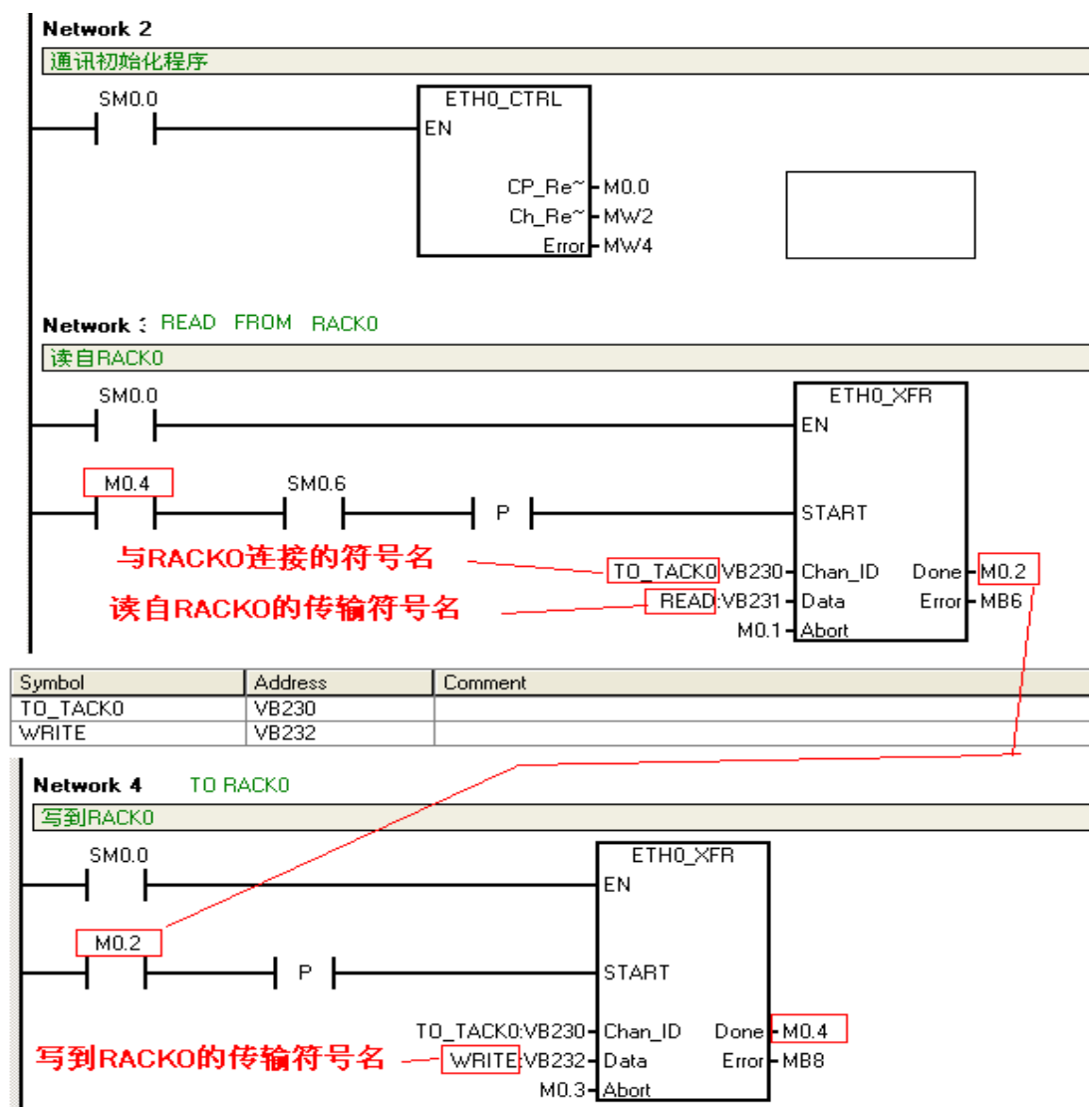


图 3.1 与 400H 侧 RACK0 CPU 通讯程序

3.2 编写与 RACK 1 CPU 通讯的程序

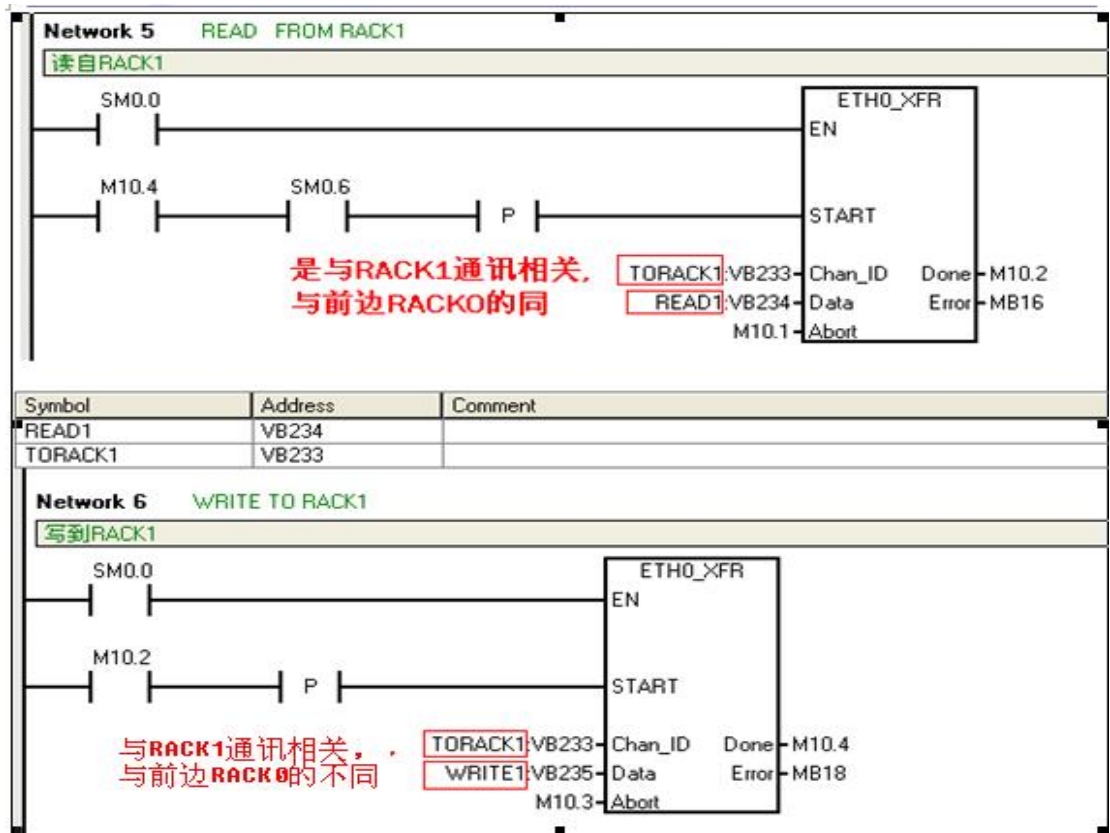


图 3.2 与 400H 侧 RACK1 CPU 通讯程序

3.3 如下图所示，编写 S7-200 发送心跳脉冲信号，并将常数 89 写到 VB11，通过前边的通讯功能块，S7-200 会通过建立的两个连接，把这一数值分别发送到 400H 侧里接收的两个 DB 块里，即 DB2 和 DB12 的前两个字节里。

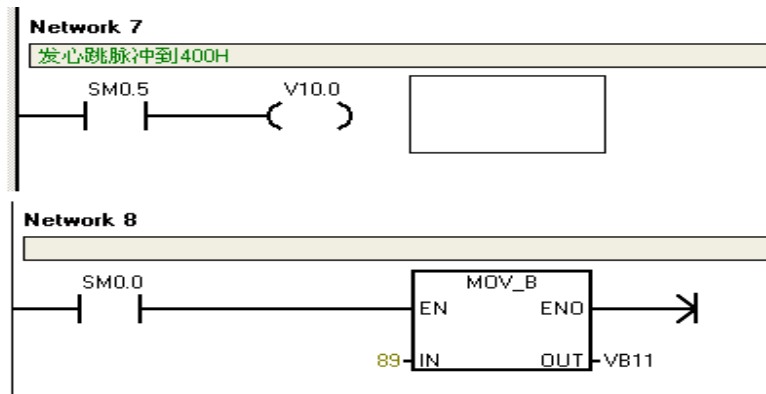


图 3.3 发送心跳脉冲的程序

3.4 400H 侧的编程

首先建立与 S7-200 通讯的 DB 块，包括 DB1，DB2，DB10，DB11，DB12，每个 DB 里至少 8 个字节，接下来，编写心跳监测和选择程序，在本例中，心跳监测时间是 3 秒，即如果心跳信号超过 3 秒不变化，就认为这个连接已经断掉了，实际应用中根据工艺要求来更改合适的时间。

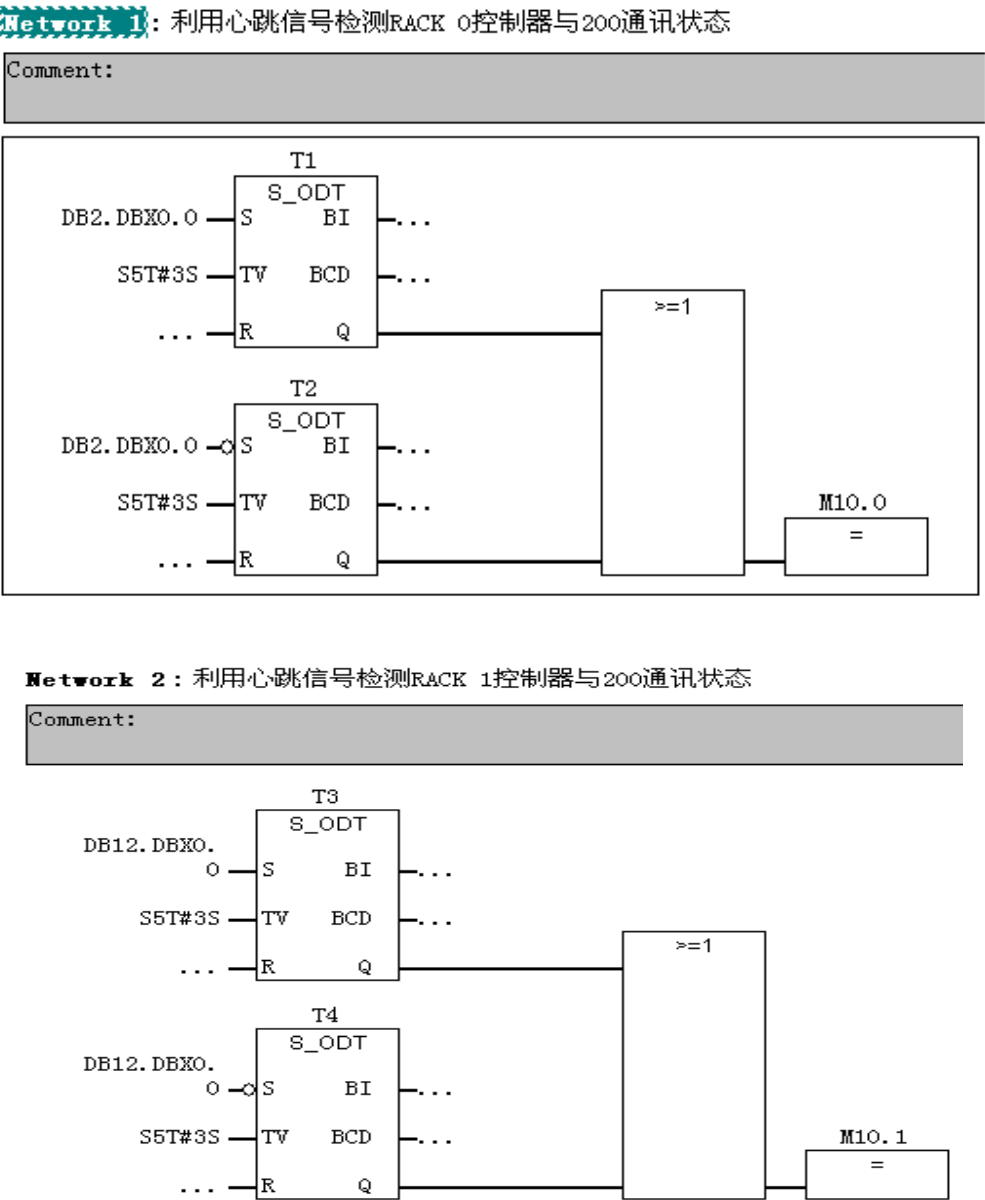


图 3.4 S7-400H 侧根据两个连接的状态，进行活动连接的判断

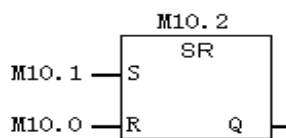
Network 3: 两个连接都断了,跳转出去,连接保持

Comment:



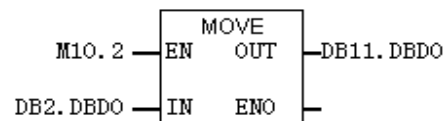
Network 4: 根据两个连接的状态计算活动连接

Comment:



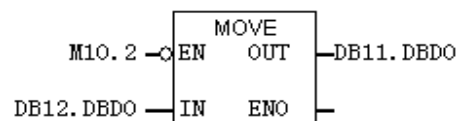
Network 5: M10.2=1使用RACK0连接数据

Comment:



Network 6: M10.2=0使用RACK1连接数据

Comment:



Network 7: Title:

Comment:

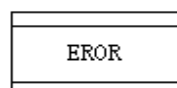
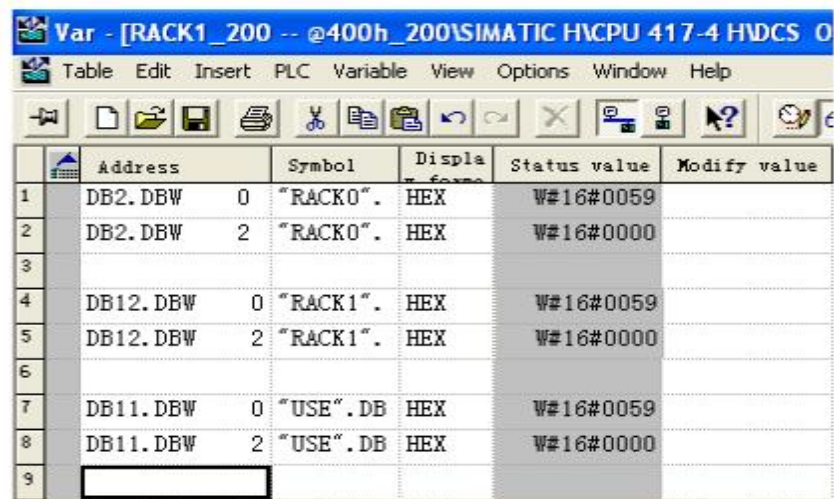


图 3.5 利用前边的活动连接, 判断在程序里使用哪个通讯连接的数据。

4. 以太网通讯测试

4.1 测试 S7-200 与 400H 通讯的效果，当 RACK0 与 RACK1 的网线都连接时，如下图所示，RACK0 和 RACK1 从 S7-200 取来的数值相等。



	Address	Symbol	Display format	Status value	Modify value
1	DB2.DBW 0	"RACK0".	HEX	W#16#0059	
2	DB2.DBW 2	"RACK0".	HEX	W#16#0000	
3					
4	DB12.DBW 0	"RACK1".	HEX	W#16#0059	
5	DB12.DBW 2	"RACK1".	HEX	W#16#0000	
6					
7	DB11.DBW 0	"USE".DB	HEX	W#16#0059	
8	DB11.DBW 2	"USE".DB	HEX	W#16#0000	
9					

图 4.1 S7-400H 侧正常时的数据

4.2 断开 RACK1 的 CP443-1 的网线，结果如下图所示，

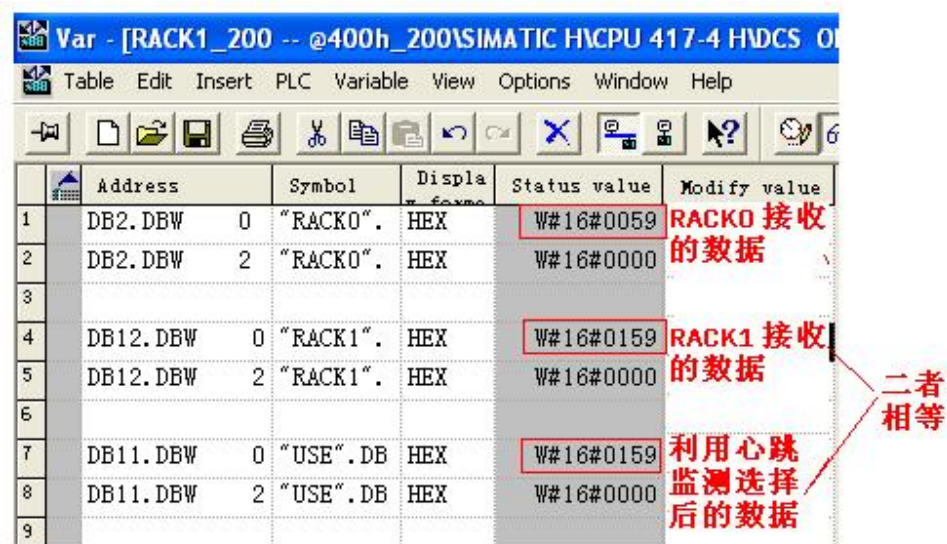


	Address	Symbol	Display format	Status value	Modify value
1	DB2.DBW 0	"RACK0".	HEX	W#16#0059	
2	DB2.DBW 2	"RACK0".	HEX	W#16#0000	
3					
4	DB12.DBW 0	"RACK1".	HEX	W#16#0159	
5	DB12.DBW 2	"RACK1".	HEX	W#16#0000	
6					
7	DB11.DBW 0	"USE".DB	HEX	W#16#0059	
8	DB11.DBW 2	"USE".DB	HEX	W#16#0000	
9					

注:通讯正常时,由于DBX0.0是脉冲,故DBW0是在W#16#0059和W#0159之间不断变化

图 4.2 断开 400H 的 RACK1 的 CP443-1 的网线后的结果

4.3 恢复 RACK1 CP443-1 的网线，拔掉 RACK0 CP443-1 的网线，结果如下图所示：



	Address	Symbol	Display format	Status value	Modify value
1	DB2.DBW 0	"RACK0".	HEX	W#16#0059	RACK0 接收的数据
2	DB2.DBW 2	"RACK0".	HEX	W#16#0000	
3					
4	DB12.DBW 0	"RACK1".	HEX	W#16#0159	RACK1 接收的数据
5	DB12.DBW 2	"RACK1".	HEX	W#16#0000	
6					
7	DB11.DBW 0	"USE".DB	HEX	W#16#0159	利用心跳监测选择后的数据
8	DB11.DBW 2	"USE".DB	HEX	W#16#0000	
9					

二者相等

图 4.3 恢复 400H 的 RACK1 的 CP443-1 的网线后的结果

小结：通过合理的编程，通过以太网，可以实现 S7-400H 与 S7-200 的“冗余”通讯，通讯成功与否的关键在如何判断两个连接的状态。判断连接状态好坏的方法并不唯一，本文中提供的方法仅供参考。

二、400H 与 S7-200 通过 DP 进行通讯

400H 与 S7-200 通讯的另一种解决方案是用 DP 通讯，S7-200 侧需配置 EM277 DP 模块，由于 S7-400H 冗余系统拥有两条 DP 总线，而 EM277 只有一条 DP 总线，因此需要在 400H 侧借助于 Y-LINK 接口模块。Y-LINK 可以在两条 DP 总线中实现自动的切换，S7-200 侧的 EM277 可以挂在 Y-LINK 转换出来的单 DP 总线上，这种解决方案双方都无需编程，非常方便。

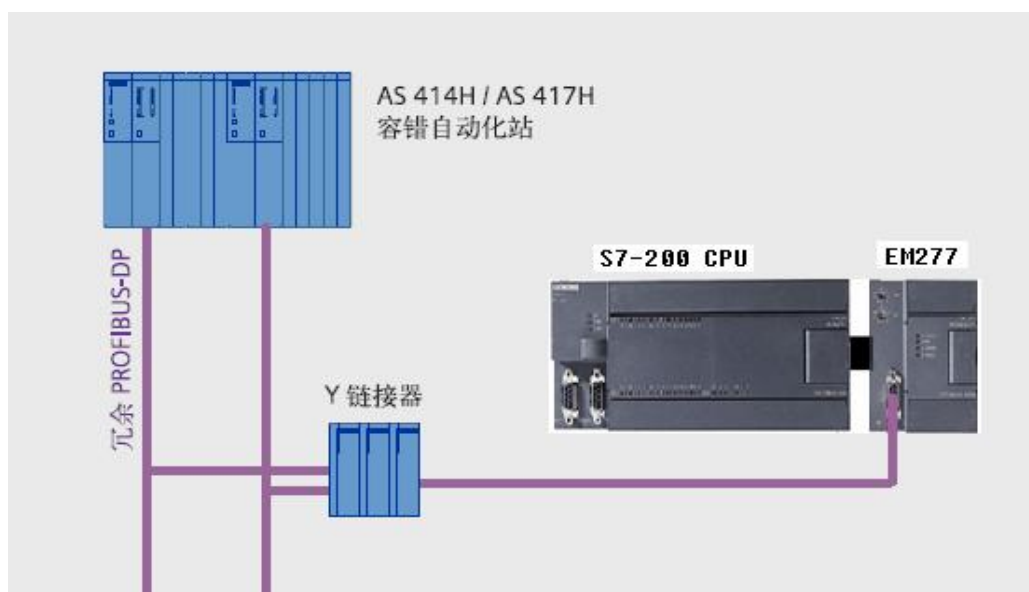


图 4.4 S7-400H 与 S7-200 通过 Y-LINK 连接示意图

具体的配置步骤如下：

1. S7-200 侧的配置

只需将 EM277 硬件扩展到 S7-200 CPU 后，并设置正确的 DP 地址，如下图所示：

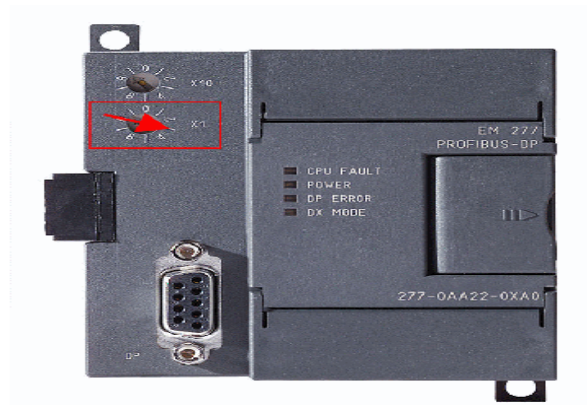


图 4.5 EM277 的地址设置方法

EM277 的地址设定方法是，转动左上角的地址开关，设置正确的 DP 地址，该地址必须与主站中组态的地址相匹配。注意：EM277 的地址设置后，要再次上电才可以生效。

2. 400H 侧的硬件配置：

2.1 在下装并安装 EM277 的 GSD 文件，

必须通过 GSD 文件将 EM277 集成到 STEP 7 的硬件目录中，因为缺省情况下硬件目录中不包含该硬件，它位于 SIMATIC 客户支持网站的互联网中，可以在条目号 [183371](#) 中找到相关信息。关于如何在 STEP 7 中导入 GSD 文件，请参考下图：

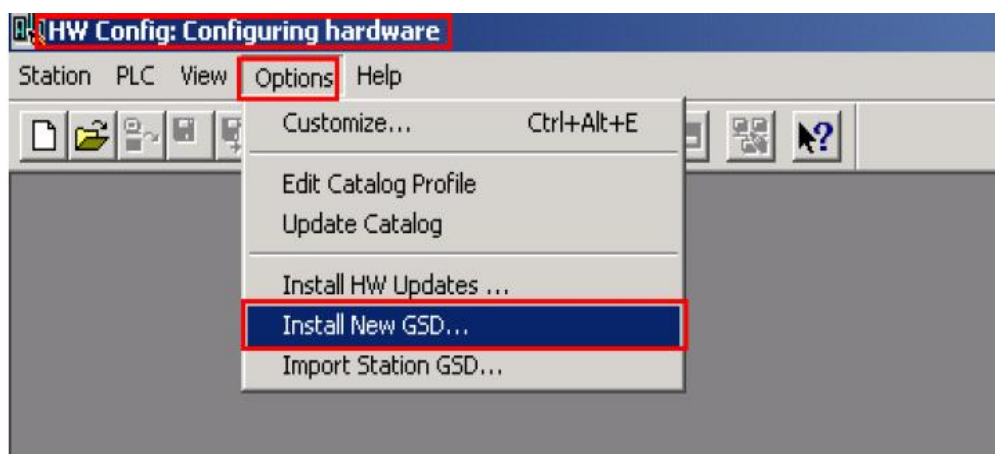


图 4.6 导入 GSD 文件

2.2 插入并配置 Y-LINK

在 SIMATIC 管理器中创建一个新项目,插入一个 S7-400H 站，配置好硬件，在 DP/PA Link 组里，找到对应的 Y-LINK，将它拖到 400H 的 DP 总线上，在弹出的对话框里选择 “Interface module for PROFIBUS-DP”,注意插入的 Y-LINK 要与实际的硬件订货号对应好，如下图所示：

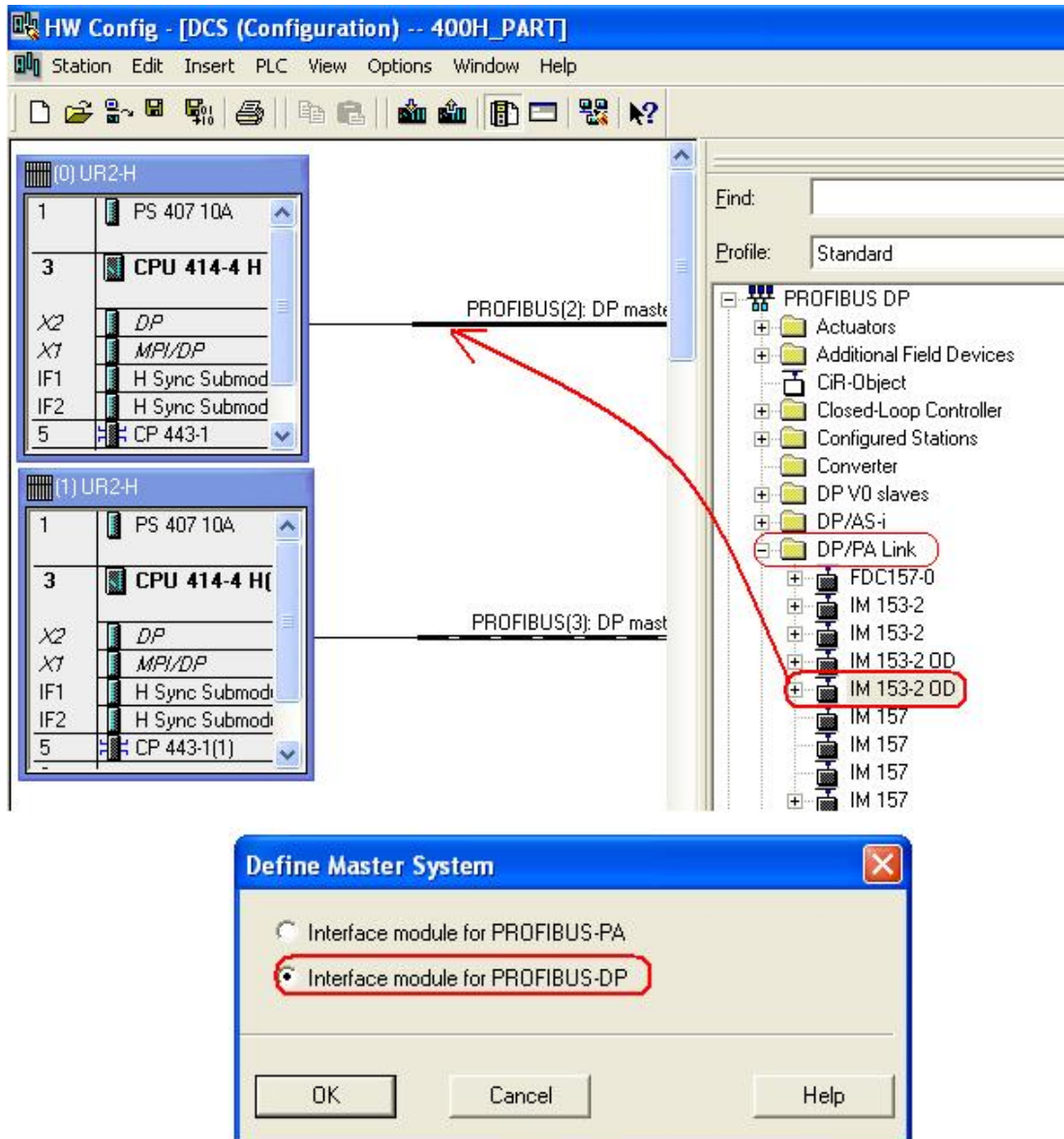


图 4.7 400H 里插入 Y-LINK

在弹出的对话框里，设定 Y-LINK 的地址,本例中地址为“3”。通过 Y-LINK,将 H 系统的双 DP 总线转换成了单 DP 总线,接下来可以配置 EM277 了。

2.3 Y-LINK 后插入 EM277

将 EM277 模块移动到 PROFIBUS DP 网段，然后设置从站的地址。在此示例程序中，地址设置为“3”(注意 Y-LINK 后的 DP 总线的站址和 400H 的双 DP 总线站址是相对独立的)。

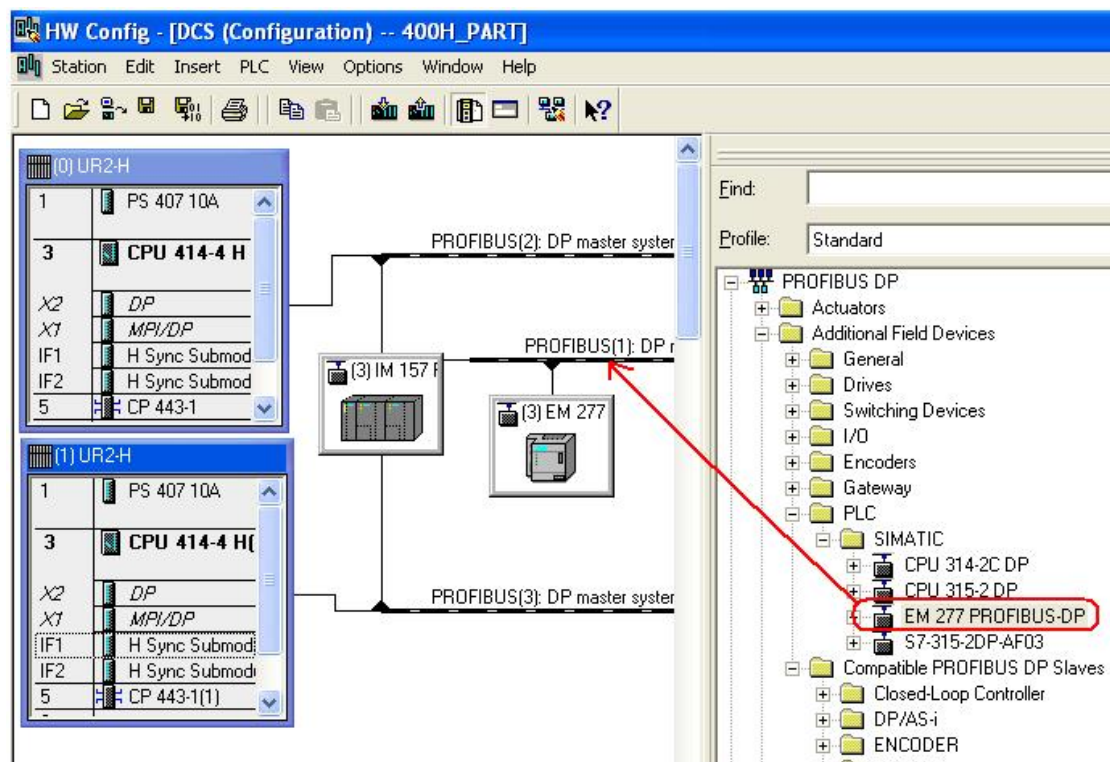


图 4.8 Y-LINK 后插入 EM277

2.4 配置双方通讯数据区

对于主站和从站之间的数据通信，在插入硬件后，必须在通讯两端为接收和发送数据定义地址区，400H 侧对应的是 I/O 地址，S7-200 侧，这些区域位于变量存储区 V 区中，配置过程如下：

2.4.1 配置 S7-400H 侧的通讯地址区

根据所选择 400H 侧的接收和发送地址区，从硬件目录中添加相关的输入/输出模块，本实例中要求是 4 字节输出/ 26 字节输入，列表中并没有，可以通过添加“Universal module”（通用模块）的方法来解决，添加方法如下图所示：

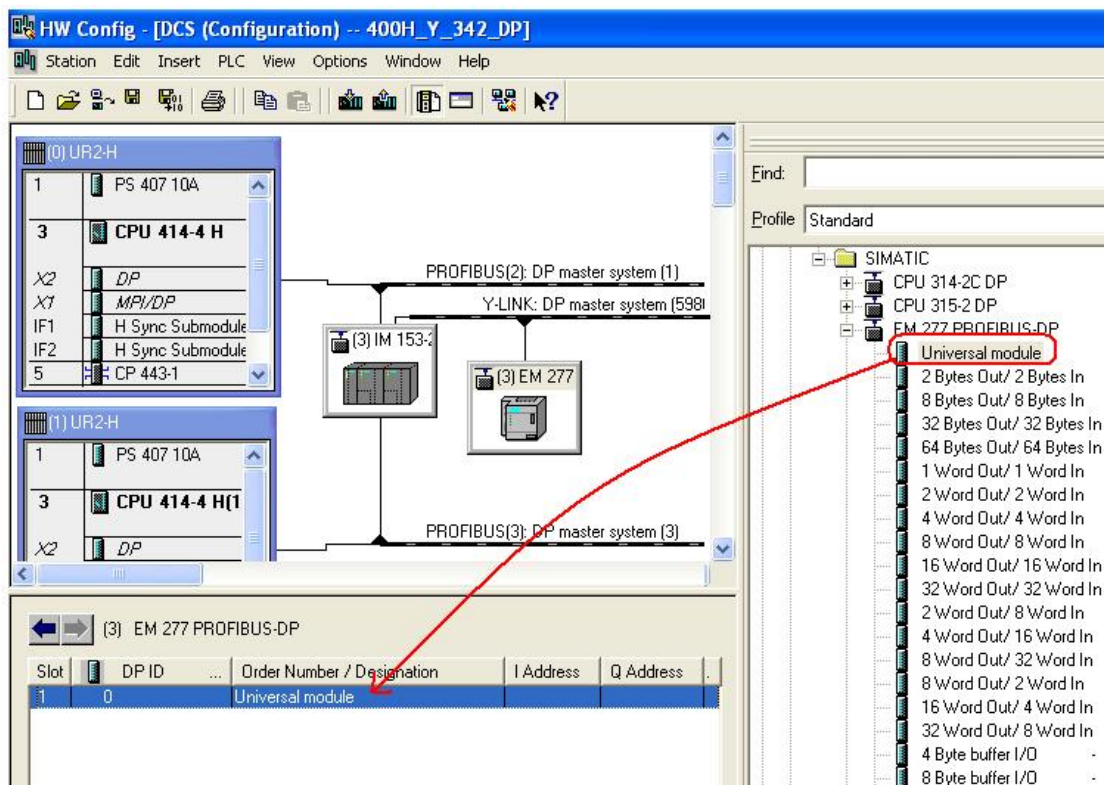


图 4.9 配置 400H 插入通讯数据区

2.4.2 配置 400H 侧 I/O 地址

对添加的“Universal module”模块进行配置，可通过双击打开它的属性，在 I/O Type 里默认是 Empty slot（空槽），将它改成 Output-Input(输出-输入)类型，并添加相应的地址区，本例中是 4 个字节的输出和 26 个字节的输入，400H 侧的配置完成。

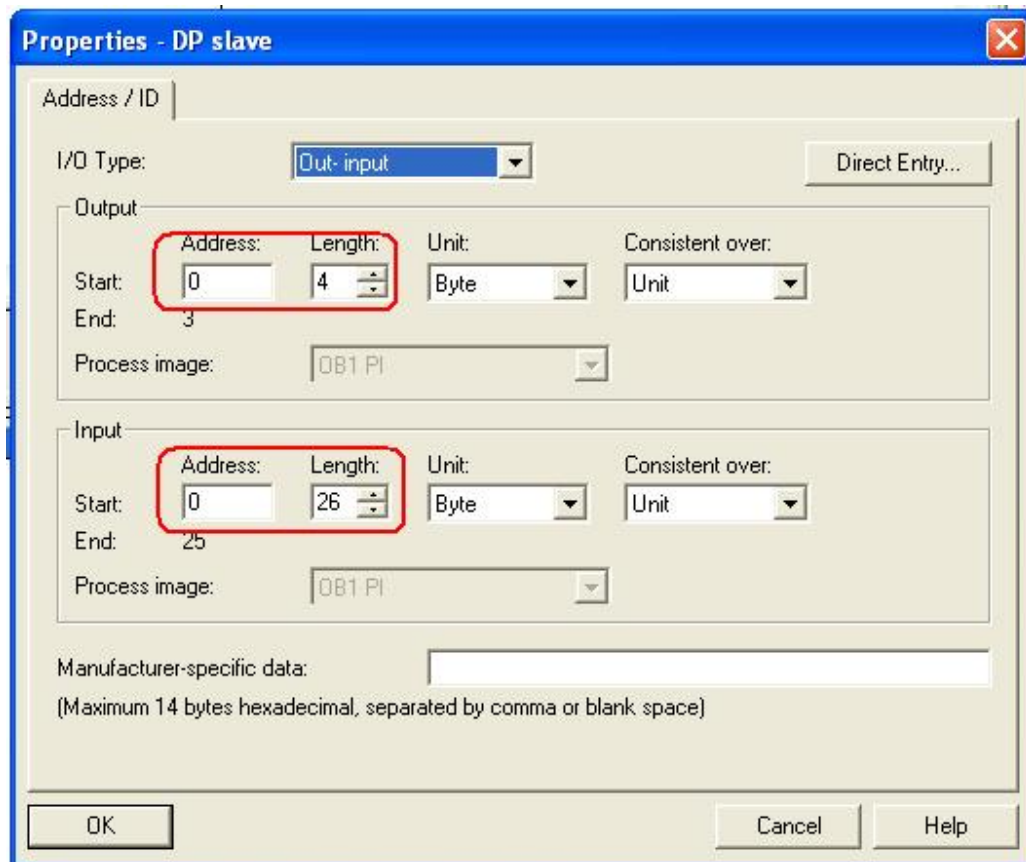


图 4.10 配置 400H 侧的通讯数据区

2.5 配置 S7-200 侧的通讯地址区

在硬件组态里，打开 EM277 模块的属性窗口，然后通过参数 Station parameters→Device-specific parameters→I/O offset in the V-memory(V 存储器中的 I/O 偏移地址)来指定接收区的起始地址。在示例组态中，选择 VB290 作为起始地址，即 400H 发到 S7-200 的数据从 V290 开始接收。

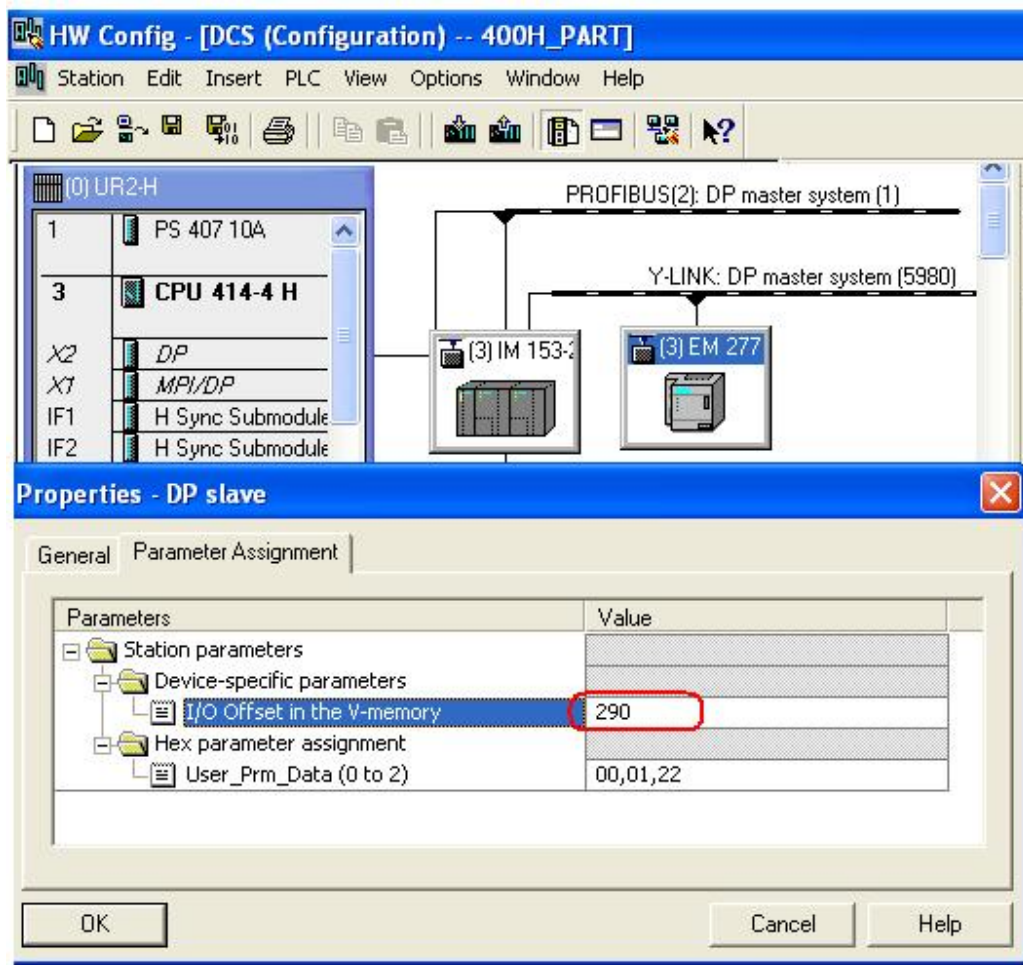


图 4.11 指定 S7-200 侧的通讯地址区

在前边的示例中的组态里，400H 侧的 4 个字节的输出对应 S7-200 侧的输入，26 个字节的输入对应 S7-200 侧的输出，它们的地址区间如下所示：

- 接收区 S7-400H: IB0 至 IB26
- 发送区 S7-400H: QB0 和 QB4
- 接收区 S7-200: VB290 和 VB293
- 发送区 S7-200: VB294 和 VB319

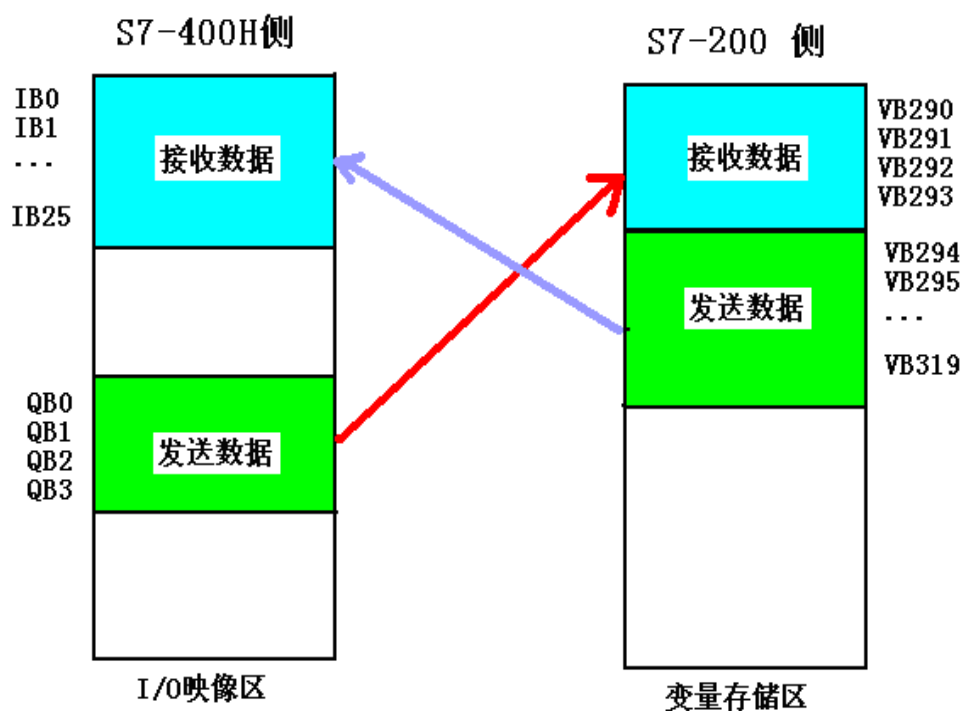


图 4.12 S7-400H 与 S7-200 的通讯地址示意图

2.6 下装硬件组态到 400H 控制器.

3. 组态完成

由于 S7-200 侧的 EM277 无需组态,S7-400H 侧硬件组态完成并下装正常后,即可实现控制器级的通讯。

注意事项:

在数据通信中，主站发送的数据存储在从站的接收区 (变量存储区) 中,S7-200 CPU 的用户程序必须将此数据“保存”到其它数据区，否则这些数据将在下一次数据发送时被覆盖。当用户程序正在处理变量存储区中的数据时，不能覆盖这些数据。