SIEMENS

S7-300 CP341 作主 S7-200 作从的 Modbus 通信

Modbus Communication -- S7-300 CP341 as Master and S7-200 as Slave

Getting Started **Edition (2009 年 11 月)**

摘要 自动化各个厂家在工业控制通信方面都有各自的通信协议及方式。西门子控制产品中通信的主要方式有 MPI，Profibus，Ethernet。在现场应用中，往往需要两个厂家的控制器进行通信交换数据。Modbus 通信是常用的一种。本文就以 CP341 都作为 Modbus 主站，S7-200 作为 Modbus 从站的通信实验作介绍。

关键词 CP341 S7-200 Modbus

Key Words CP341 S7-200 Modbus

目　录

1. 系统简介及软硬件需求

Modbus 是公开通信协议，其具有两种串行传输模式，ASCII 和 RTU。它们定义了数据如何打包、解码的不同方式。通信双方必须同时支持上述模式中的一种，通常支持 Modbus 通信的设备大都支持 RTU 格式。Modbus 通信标准协议可以通过各种传输方式传播，如 RS232C、RS485、光纤、无线电等。 在 S7-200 CPU 通信口上实现的是 RS485 半双工通信，使用的是 S7-200 的自由口功能。

Modbus 是一种单主站的主/从通信模式。Modbus 网络上只能有一个主站存在，主站在 Modbus 网络上没有地址，从站的地址范围为 0 - 247，其中 0 为广播地址，从站的实际地址范围为 1 - 247。

在实现 Modbus 通信方面，西门子 AS 产品中往往会用到 CP341 和 S7-200。其二者之间的不同是 CP341 的接口类型多，包含 RS 232C (V.24) 20 mA (TTY), RS 422/RS 485 (X.27)。由于其实现 Modbus 通信需要购买 Modbus Dongle，在实现功能成本方面比较高。但是由于 CP341 可安装在 ET200M 站上通过 Profibus 的方式与主站相通，此种方案很适合比较大型系统进行的 Modbus 通信设计和改造。S7-200 系列产品是西门子 AS 产品中低端的产品。但是其小而精湛集成了很多通信功能。虽然 S7-200 在实现 Modbus 通信时使用的是 S7-200 的自由口功能，接口采用 RS485，或是通过适配器转换成 RS 232 接口类型不如 CP341 的型号丰富，但是其在实现 Modbus 功能上无需组态和额外购买组件，所以实现起来既简单而且成本低廉，在比较低端的场合是个不错的选择。

在现场应用中 CP341 往往都作为 Modbus 主站来读取第三方设备的数据，而 S7-200 常作为 Modbus 从站与其他设备进行 Modbus 通信。本文就以 CP341 都作为 Modbus 主站，S7-200 作为 Modbus 从站，来实现其二者的 Modbus 通信，阐述其二者在实现通信方面的设置和注意事项。需要说明的是 S7-300 与 S7-200 的通讯方式有很多种包括 MPI、Profibus、Ethernet、Modbus 等。本文旨在说明其二者在 Modbus 通信方面的具体安装和编程步骤。

1.1 软件环境

1.1.1 STEP7 V5.4 SP3

用于编写 S7-300/400 等 PLC 程序，此软件需要购买，本文档中所有的程序代码均使用 Step7 V5.4 SP3 编写。

1.1.2 CP PtP Param V5.1 SP8

串行通信模板的驱动程序，安装此驱动后才能配置 PtP 模板，并在 Step7 中集成通信编程需要使用的功能块。此驱动随购买模板一起提供，也可以从以下的链接下载。

http://support.automation.siemens.com/CN/view/zh/27013524

1.1.3 CP PTP Modbus Master V3.1.4

CP341 或CP441-2 用于Modbus 主站时，需要安装此驱动协议，但安装此驱动之前必须先安装PtP driver，此驱动可以在购买Modbus Dongle时选择购买，或者可以从以下链接下载。 http://support.automation.siemens.com/CN/view/zh/8713784

1.1.4 STEP 7 MicroWIN V4.0 SP6

此软件是用于 S7-200 编程和组态的软件。此软件可以免费下载到。

1.1.5Toolbox_V32-STEP 7-Micro WIN 32 Instruction Library

S7-200 实现 Modbus 功能，需要使用 Modbus 的指令库，其实质是自由口通信。STEP 7-Micro/WIN V4.0 以上版内部已经带有新的指令库，支持 Modbus 通过 Port0、Port1 进行通信，但在未安装西门子 Instruction Library 软件包的情况下，不能显示出来使用。

要使用西门子的标准指令库，必须先安装西门子的指令库软件包 Instruction Library。安装了 Instruction Library 之后，只要安装的 STEP 7-Micro/WIN 版本是最新的，就能获得相应版本的新指令库。安装 Micro/WIN 的升级包（Service Pack）也会更新指令库的版本。

1.2 硬件列表

- S7-300 站：

CPU315-2DP	6ES7 315-2AG10-0AB0
CP343-1	6ES7 341-1EX30-0XE0
CP341 RS422/485	6ES7 341-1CH01-0AE0
Dongle	6ES7 870-1AA01-0YA0 (MODBUS master)

- S7-200 站：

224XP	6ES7 214-2BD23-0XB8 (选用双口CPU便于调试)

1.3 硬件安装

物理接口方面S7-200的通信口为RS485物理接口， CP341选用的也是RS 422/485接口类型的模块。二者之间可采用6ES7 902-3AB00-0AA0 RS 422/485 5m连接电缆。在本系统中

采用的电缆为DP 红B 绿A 两线电缆。

在接线之前首先要仔细阅读CP341及S7-200通信接口的手册，查看通信接口针脚的定义。如图1所示。

S7-200 CPU通信口引脚定义：

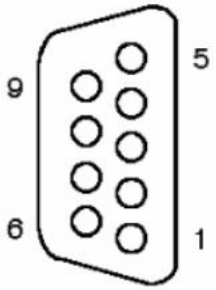
CPU插座(9针母头)	引脚号	PROFIBUS名称	Port0/Port1（端口0/端口1）引脚定义
	1	屏蔽	机壳接地（与端子PE相同）/屏蔽
	2	24V返回	逻辑地（24V公共端）
	3	RS-485信号 B	RS-485信号 B 或 TxD/RxD +
	4	发送请求	RTS（TTL）
	5	5V返回	逻辑地（5V公共端）
	6	+5V	+5V，通过100 Ohm电阻
	7	+24V	+24V
	8	RS-485信号 A	RS-485信号 A 或 TxD/RxD -
	9	不用	10位协议选择（输入）
	金属壳	屏蔽	机壳接地（与端子PE相同）/与电缆屏蔽层连通

图 1 S7-200 CPU 通信口引脚定义

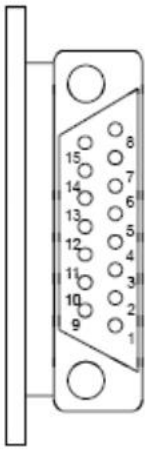
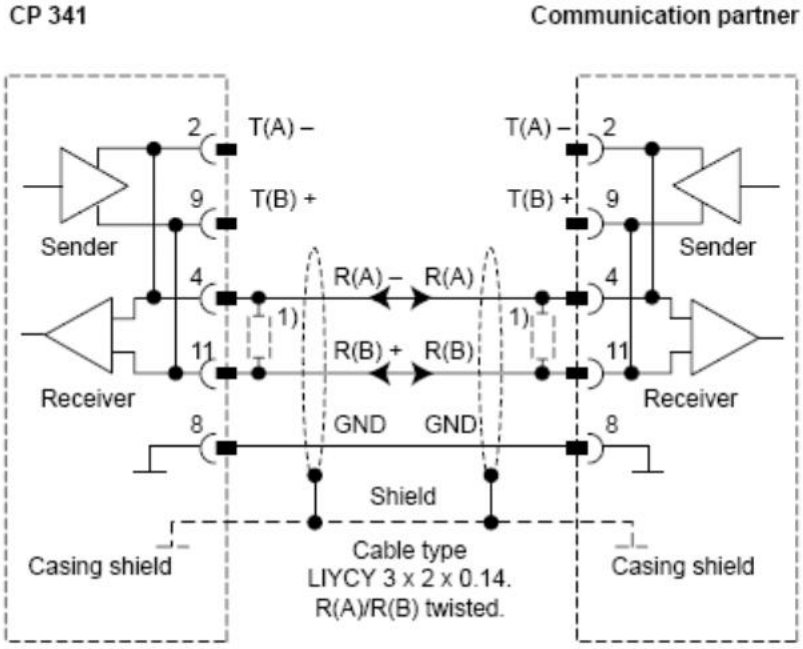
Female Connector on CP341-RS422/485*	Pin	Designation	Input/Output	Meaning
	1	–	–	–
	2	T (A) –	Output	Transmitted data (four-wire operation)
	3	–	–	–
	4	R (A)/ T (A) –	Input Input/Output	Received data (four-wire operation) Received/transmitted data (two-wire operation)
	5	–	–	–
	6	–	–	–
	7	–	–	–
	8	GND	–	Functional ground (floating)
	9	T (B) +	Output	Transmitted data (four-wire operation)
	10	–	–	–
	11	R (B)/ T (B) +	Input Input/Output	Received data (four-wire operation) Received/transmitted data (two-wire operation)
	12	–	–	–
	13	–	–	–
	14	–	–	–
	15	–	–	–

图 2 CP341 RS 422/485 通信口引脚定义



1) In the case of cables longer than 50 m you must solder in a terminating resistor of approx. 330 Ω on the receiver for trouble-free data traffic.

图 3 CP341 通过 RS485 与通信对象的连接方式

在接线时，S7-200端由于是9针 RS485口，故用标准的DP连接器。而CP341上的接口为15针口，其4、11对应RS485接线方式的连根线。从S7-200端口3引出的是红色B线，其连接CP341 15针口的11端口。从S7-200端口8引出的是绿色A线，其连接CP341 15针口的4端口。如图4所示。

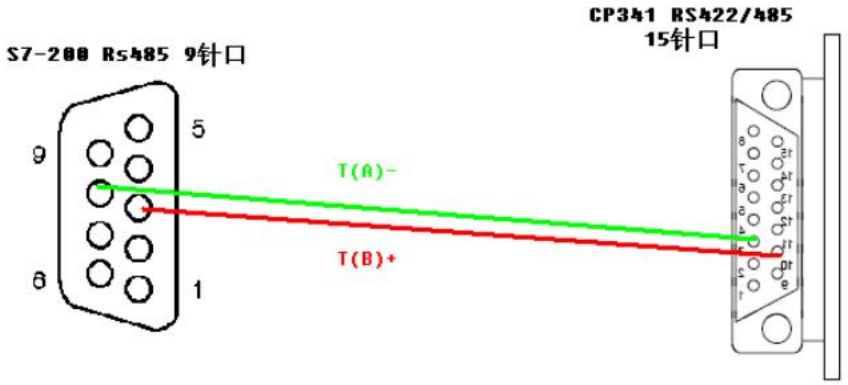
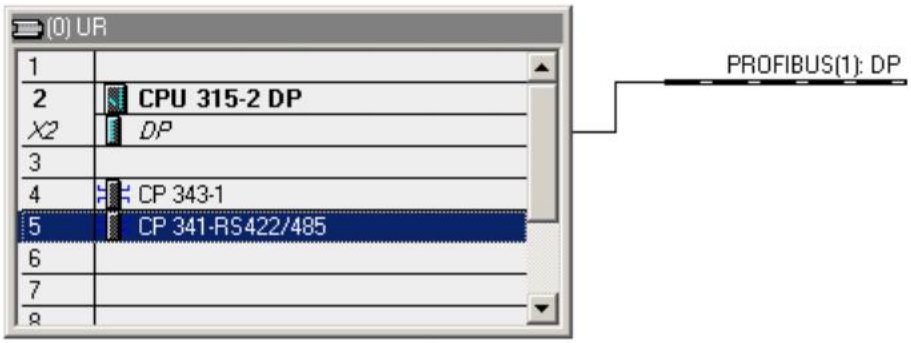


图 4 S7-200 与 CP341 RS 422/485 接口通过 DP 电缆的接线图

2. CP341 作 Modbus 主站的设置与编程

2.1 CP341 作 Modbus 主站的硬件组态

2.1.1 硬件组态 CPU 及 CP341



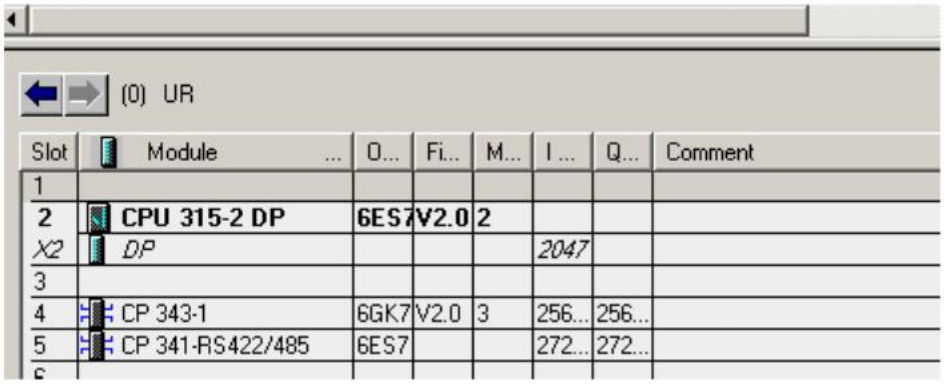


图 5 将 CP341 逻辑地址为 272

2.1.2 设置 Modbus 总线传输速率和帧字符结构

双击CP341->Parameters—>Protocol中选择Modbus Master；

双击信封Protocol，选择Modbus Master设置总线传输速率和帧字符结构。

如图 6，**本例传输速率为9.6Kbit/s，帧字符选用8位数据位，1位停止位，无校验。此设置可根据实际情况调整，原则是通信双方选择一致**。

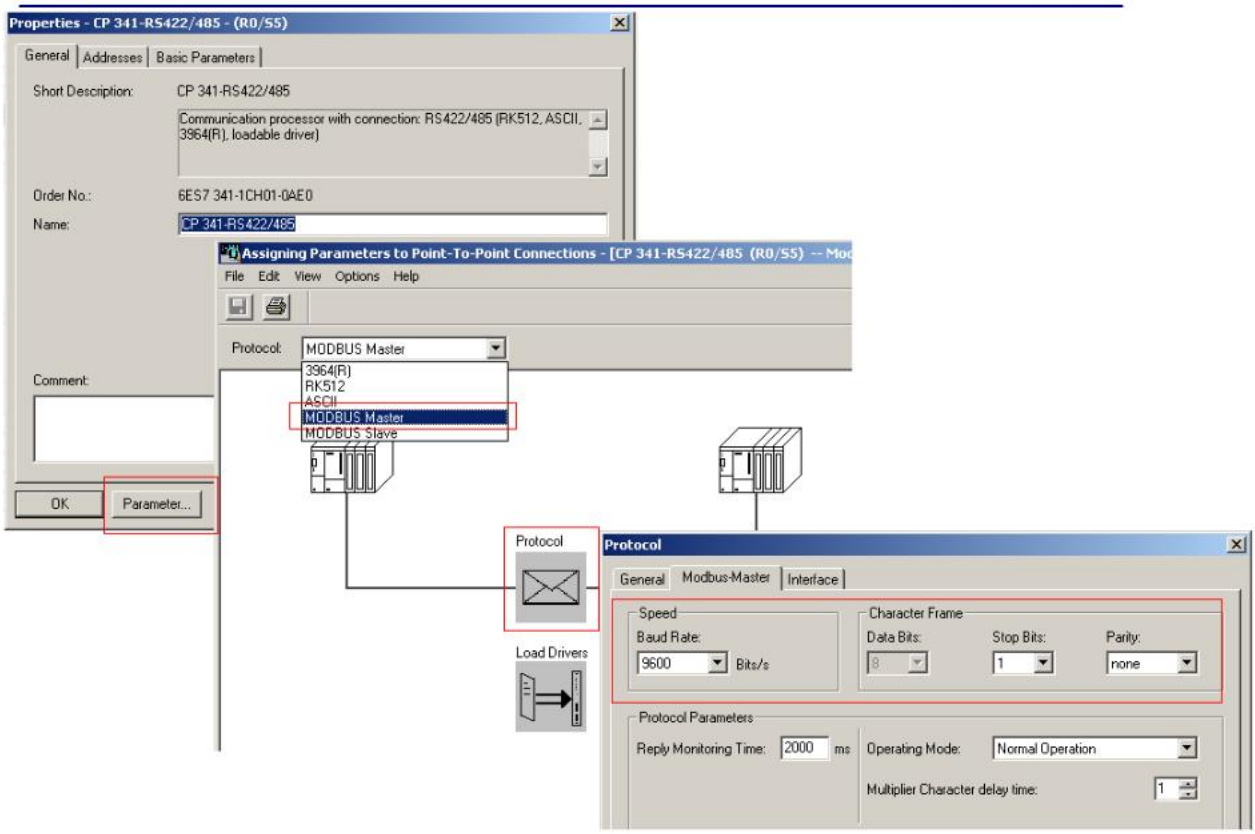


图 6 通信帧字符结构

2.1.3 设置电气接口

选择半双工 RS 485，默认设置是 R（A）为-，R（B）为+。此设置主要是与 RS485 A、B 两线正负定义有关，在 CP341 接线与通信对象 A、B 两线正负定义相反时，可无需修改硬件接线，可直接修改此处颠倒接线正负。如图 7 所示。

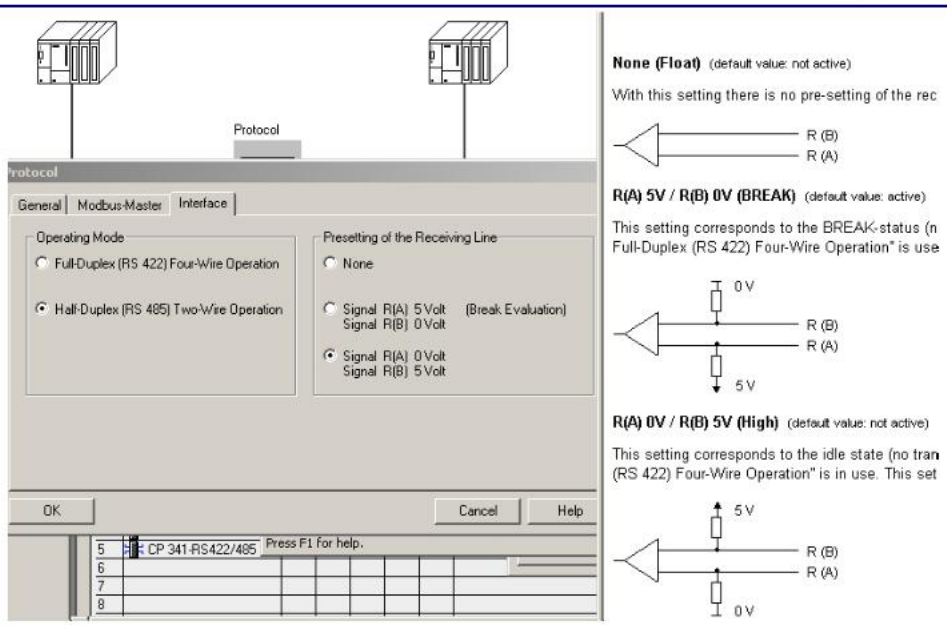


图 7 CP341 接线与通信对象 A、B 两线定义

2.1.4 Modbus 配置的下载

当配置好 Modbus 通信的参数后，在向 CPU 下载硬件组态前，要向 CP341 的 Dongle 中下载 Modbus Master 的驱动，一旦下载完成后此后无需再次下载。对于通信参数的调整只需要进行 HW 对 CPU 的硬件下载即可。

要注意的是，在下载 Dongle 时，一定要 **CPU 停机下载**。如图 8 所示

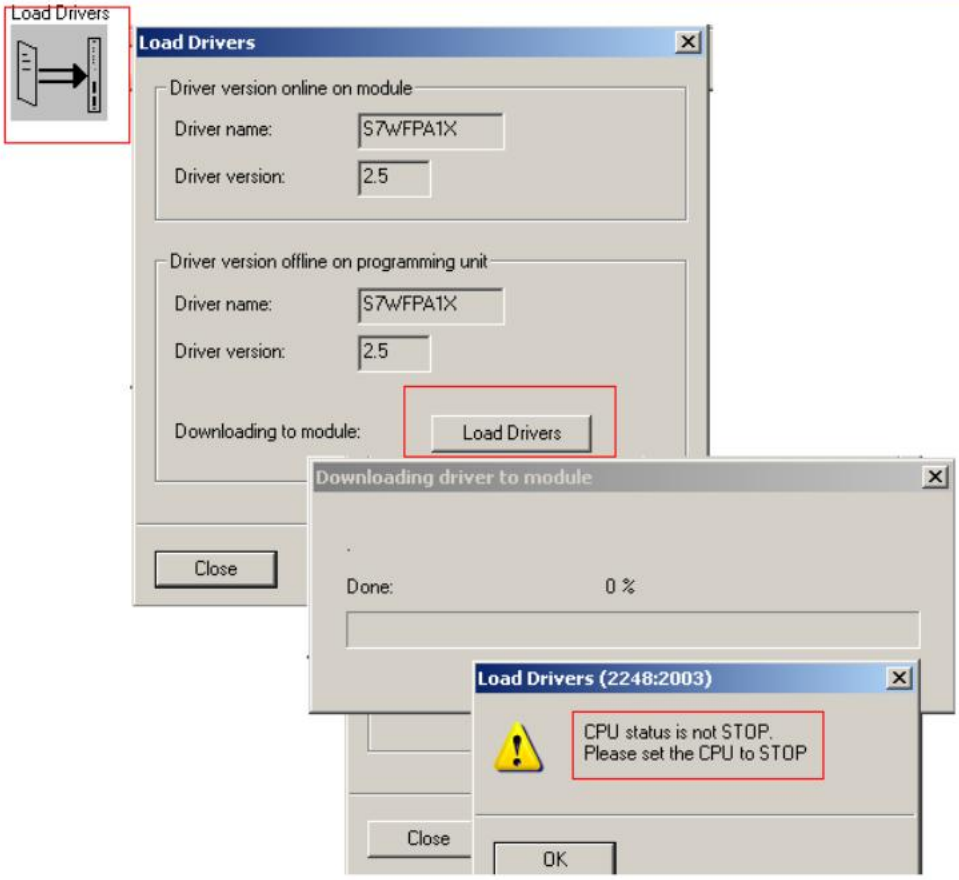


图 8 下载 Dongle 时，一定要 CPU 停机下载

2.2 CP341 作 Modbus 主站的编程

实现 CP341 作 Modbus 主站的通信程序是“P_SND_RK”FB8 负责发送控制字。编程如下：

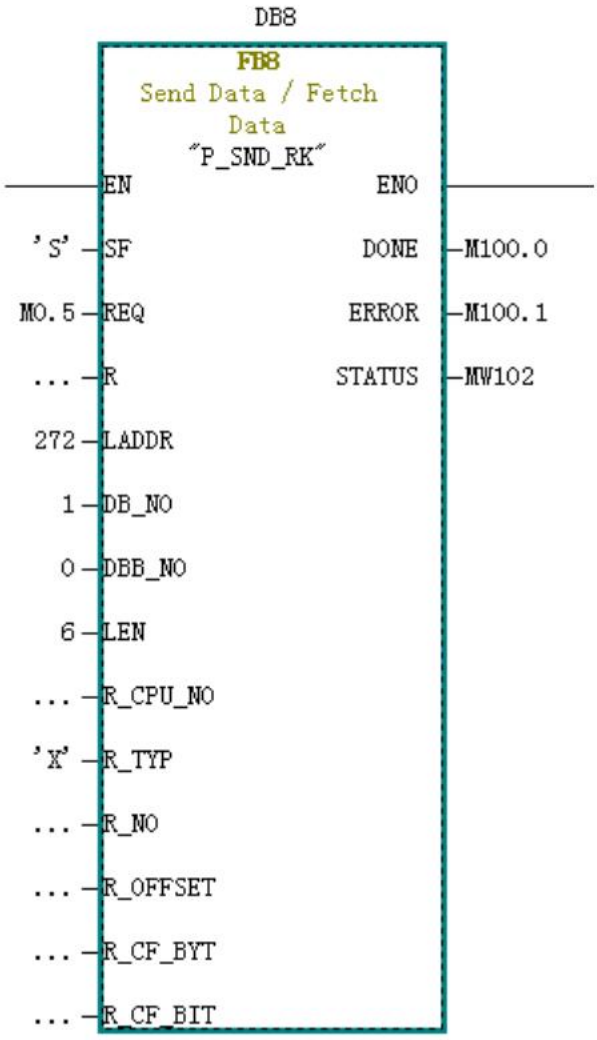


图 9 “P_SND_RK” FB8

FB8 参数说明表格 1。

SF	‘S’为发送，此处必须为大写的‘S’
LADDR	硬件组态中的起始逻辑地址，本例中为 272
REQ	发送数据触发位，上升沿触发，本例中为 M0.5，以 1s 为周期的脉冲信号
DB_NO	发送数据块号，本例中为 1
DBB_NO	发送数据的起始地址，本例中为 0
LEN	发送数据的长度，本例中暂时为 6
R_TYP	‘X’为扩展的数据块，此处必须为大写的‘X’
R	取消通信，本例始终为初始值 FALSE
DONE	发送完成位，无故障发送完成后为 true，M100.0
ERROR	错误位，为 true 说明有错误，M100.1
STATUS	状态字，标识错误代码，查看 Modbus Master 和 CP341 手册
其它参数	查看在线帮助

表 1

其中创建发送数据块 DB1 结构如图 10 所示：

Address	Name	Type	Initial value
0.0		STRUCT	
+0.0	address	BYTE	B#16#1
+1.0	code	BYTE	B#16#3
+2.0	reg_startadr	INT	0
+4.0	reg_count	INT	4
=6.0		END_STRUCT	

图 10 发送 DB 块源区域结构

创建的发送数据块 DB1，至少要有 6 个字节的长度，后面根据功能码的不同，所需长度不同，建议发送数据块创建的长度长一些。以上图为例 6 个字节 Address 为所通讯对象的 Modbus 地址为 1，code 为功能码 FC03,所读对象寄存器的起始地址为 0，所读寄存器的数量为 4，其中一个寄存器为两个字节。

此处有几个注意事项：

1） R_TYP 必须为大写的'X'，

为小写"x"时，CP341 作为 Modbus master 时,调用 FB8 的状态字显示为 "0E4F" 错误。注意 R_TYP 必须为大写。且对于 R_TYP 可以写的值如下：

'D' DB 区

'X' DX 区，extended data block

'E' I 区

'A' Q 区

'M' M 区

'T' T 区

'C' C 区

2) LEN 的长度要根据通信所需功能码来针对填写，

如下表所示

The length LEN depends on the function code used.

Function Code	Length LEN in Bytes
01	6
02	6
03	6
04	6
05	6
06	6
07	2
08	6
11	2
12	2
15	>6
16	>6

表 2

LEN 的长度与功能码的对应。

“P_RCV_RK” FB7 用于接收通信数据。编程如图 11 所示：

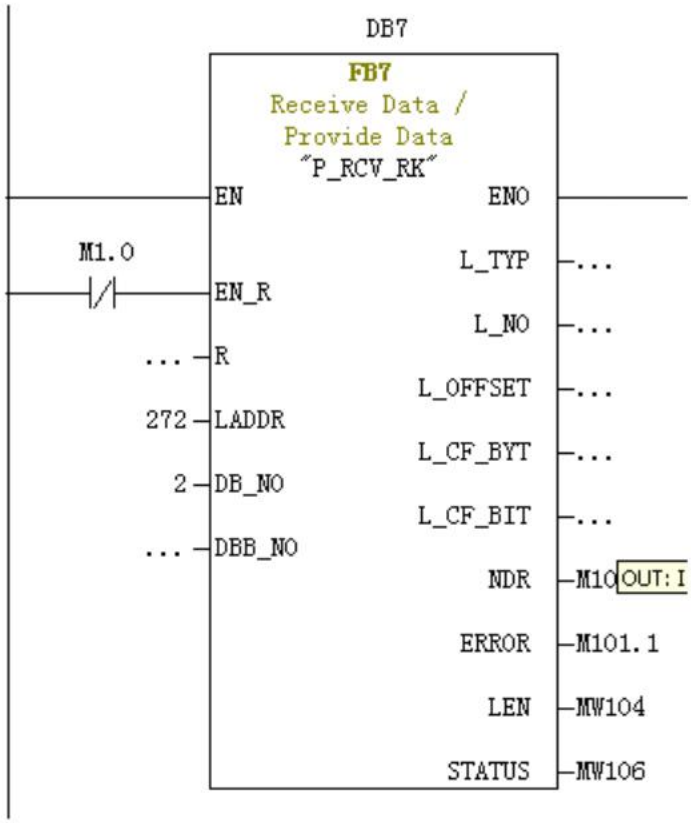


图 11 “P_RCV_RK” FB7

FB7 参数说明表格 3

LADDR	硬件组态中的起始逻辑地址，本例中为 272
DB_NO	接收数据块号，本例中为 DB2
DBB_NO	接收数据的起始地址，本例中默认为 0

LEN	接收数据的长度反馈
EN_R	使能接收位，本例中始终为 TURE
R	取消通信，本例始终为初始值 FALSE
NDR	接收完成位，无故障接收完成后为 true
ERROR	错误位，为 true 说明有错误
STATUS	状态字，标识错误代码，查看 Modbus Master 和 CP341 手册
其它参数	查看在线帮助

表 1

3．S7-200 作 Modbus 从站的设置

S7-200 作 Modbus 通信要用到自由口通信下的 Modbus Slave 库，对于此库的应用要注意的是

- Modbus Slave 库仅支持 Modbus RTU 通信模式，不支持 ASCII 通信模式。
- 目前的 Modbus Slave 库仅支持通信口 Port0。
- 使用 Modbus Slave 库时一定要注意对库分配内存区空间。否则编译后出现很多错误。

如图 12 所示：

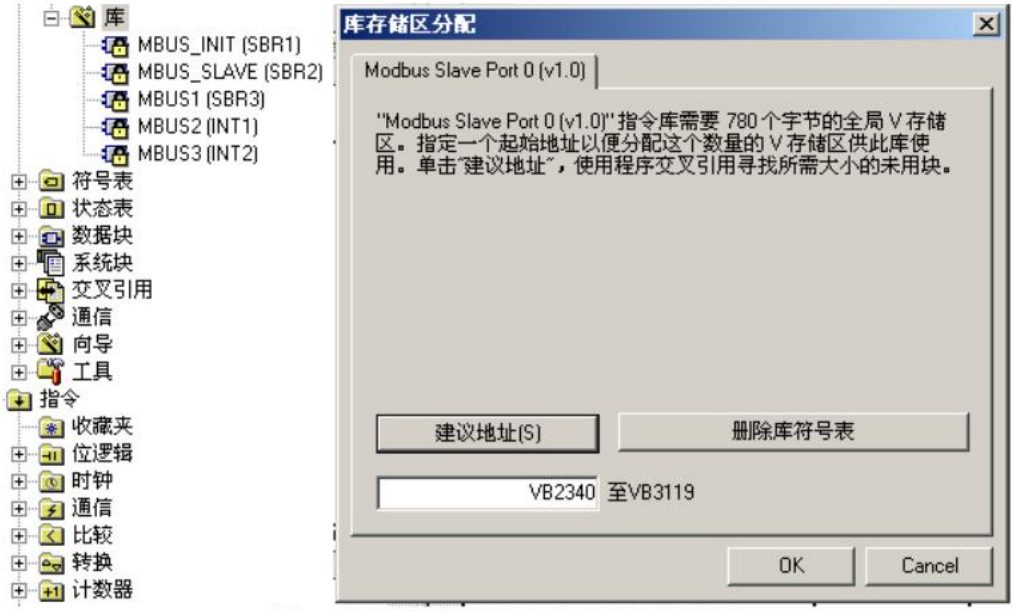


图 12 调用的库要分配系统内存地址区

编程时使用 SM0.1 调用子程序 MBUS_INIT 进行初始化，使用 SM0.0 调用 MBUS_SLAVE，并指定相应参数。关于参数的详细说明，可在子程序的局部变量表中找到。

图 13 为 S7-200 实现 Modbus 从站的程序。

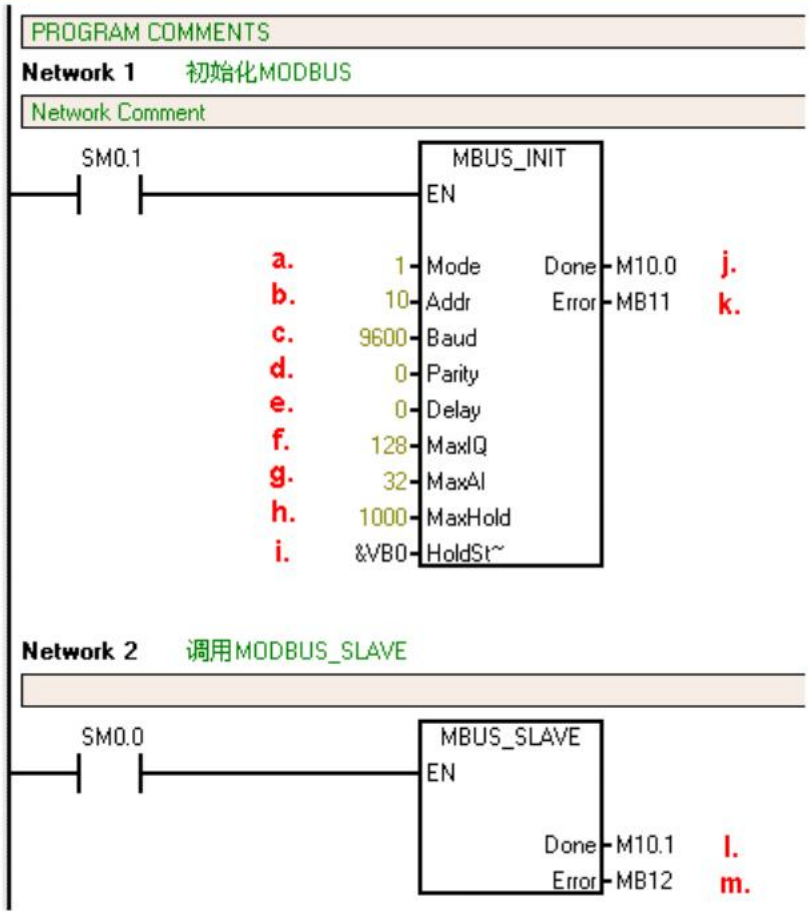


图 13 调用 Modbus RTU 通信指令库

图中参数意义如下：

a. 模式选择：启动/停止 Modbus，1=启动；0=停止

b. 从站地址：Modbus 从站地址，取值 1~247

c. 波特率：可选 1200，2400，4800，9600，19200，38400，57600，115200

d. 奇偶校验：0=无校验；1=奇校验；2=偶校验

e. 延时：附加字符间延时，缺省值为 0

f. 最大 I/Q 位：参与通信的最大 I/O 点数，S7-200 的 I/O 映像区为 128/128，缺省值为 128

g. 最大 AI 字数：参与通信的最大 AI 通道数，可为 16 或 32

h. 最大保持寄存器区：参与通信的 V 存储区字（VW）

i. 保持寄存器区起始地址：以&VBx 指定（间接寻址方式）

j. 初始化完成标志：成功初始化后置 1

k. 初始化错误代码

l. Modbus 执行：通信中时置 1，无 Modbus 通信活动时为 0

m. 错误代码：0=无错误

从程序截图中可见，S7-200 作为 Modbus 从站，从站地址为 10，接收存储区为 VB0 开始。

4. 通信测试

至此，CP341 和 S7-200 双方的程序及物理连线已经做好。在调试 S7-300 时可通过 CP343-1 以太网模块，以便可以用以太网通信调试速度快且方便。S7-200 站选用 224XP，其中 Port0 作为 Modbus 通信口，Port1 用于与笔记本的 CP5512 通信。

在进行通信测试前还要明确 Modbus 通信的功能码。

下表为 Micro 'n Power 中 S7-200 作为 Modbus RTU 从站通信功能码。

Modbus RTU 从站功能码

功能码	主站使用相应功能码作用于此从站的效用
1	读取单个/多个线圈（离散量输出点）状态。 功能 1 返回任意个数输出点（Q）的 ON/OFF 状态。
2	读取单个/多个触点（离散量输入点）状态。 功能 2 返回任意个数输入点（I）的 ON/OFF 状态。
3	读取单个/多个保持寄存器。功能 3 返回 V 存储区的内容。在 Modbus 协议下保持寄存器都是“字”值，在一次请求中可以读取最多 120 个字的数据。
4	读取单个/多个输入寄存器。功能 4 返回 S7-200 的模拟量数据值。
5	写单个线圈（离散量输出点）。功能 5 用于将离散量输出点设置为指定的值。这个点不是被强制的，用户程序可以覆盖 Modbus 通信请求写入的值。
6	写单个保持寄存器。功能 6 写一个值到 S7-200 的 V 存储区的保持寄存器中。
15	写多个线圈（离散量输出点）。功能 15 把多个离散量输出点的值写到 S7-200 的输出映像寄存器（Q 区）。输出点的地址必须以字节边界起始（如 Q0.0 或 Q2.0），并且输出点的数目必须是 8 的整数倍。这是此 Modbus RTU 从站指令库的限制。些点不是被强制的，用户程序可以覆盖 Modbus 通信请求写入的值。
16	些多个保持寄存器。功能 16 写多个值到 S7-200 的 V 存储区的保持寄存器中。在一次请求中可以写最多 120 个字的数据。

表 4 S7-200 作为 Modbus RTU 从站通信功能码

以下测试为现场应用中经常用到的功能码 FC01、02、03、06、16

4.1 FC01 主站读取从站多个 DO 点状态

FC01 功能下，“P_SND_RK”FB8 的 LEN 为 6，DB1 的头两个字节分别是所要读取从站的地址 10 号站（16#A），和功能码 01，如表 5 所示。图 14 为 S7-200 的 QB0 的 4 个位被传送到 DB2 的接收区。其中 Reg_num 位数为 1-2040。

地址	名称	类型	值	注释
0.0	Slave_address	BYTE	B#16#0A	从站地址
1.0	Function_code	BYTE	B#16#01	功能代码
2.0	Reg_startAdr	WORD	W#16#0	位起始地址
4.0	Reg_num	WORD	W#16#4	位数

表 5 FC01 的 SEND 源区域结构

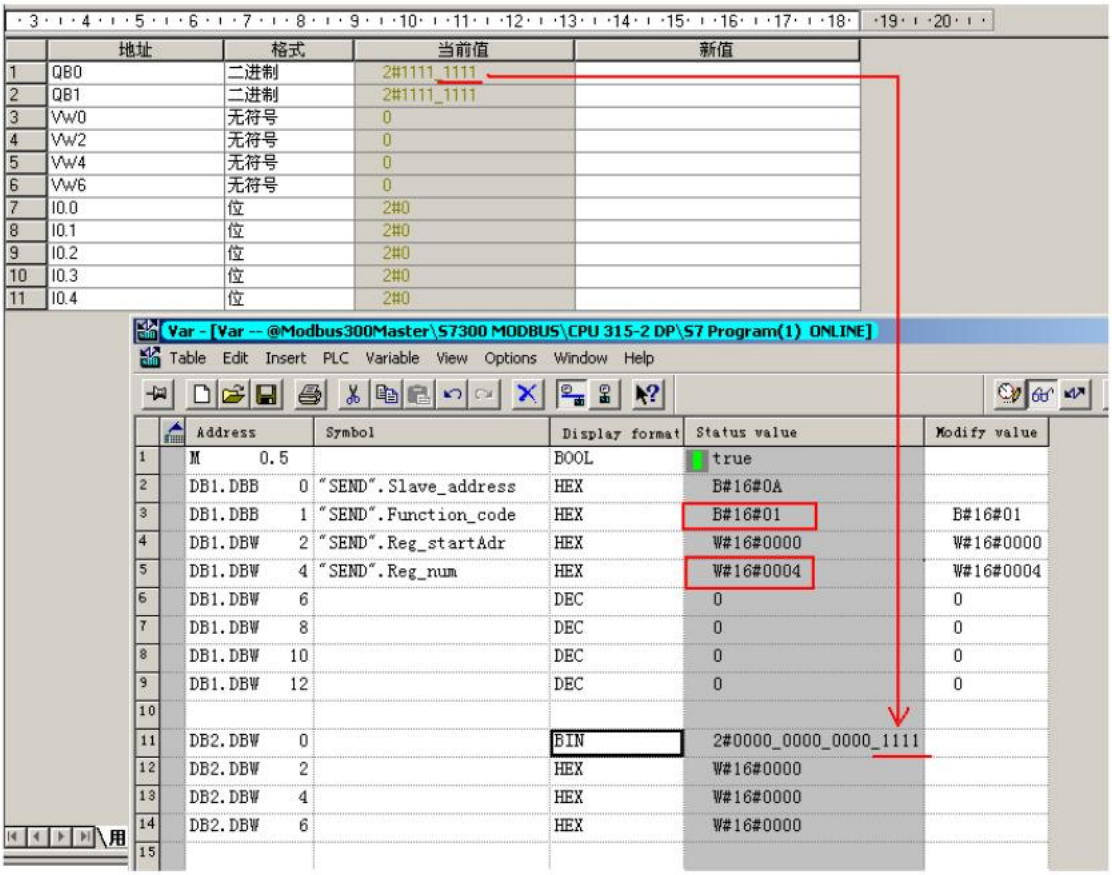


图 14 FC01 的数据交换

4.2 FC02 主站读取从站多个 DI 点状态

FC02 功能下，“P_SND_RK”FB8 的 LEN 为 6，DB1 的头两个字节分别是所要读取从站的地址 10 号站（16#A），和功能码 02，如表 6 所示。图 15 为 S7-200 的 I0.0-I0.3 的四个位被传送到 DB2 的接收区。其中 Reg_num 位数为 1-2040。

地址	名称	类型	值	注释
0.0	Slave_address	BYTE	B#16#0A	从站地址
1.0	Function_code	BYTE	B#16#02	功能代码
2.0	Reg_startAdr	WORD	W#16#0	位起始地址
4.0	Reg_num	WORD	W#16#4	位数

表 6 FC02 的 SEND 源区域结构

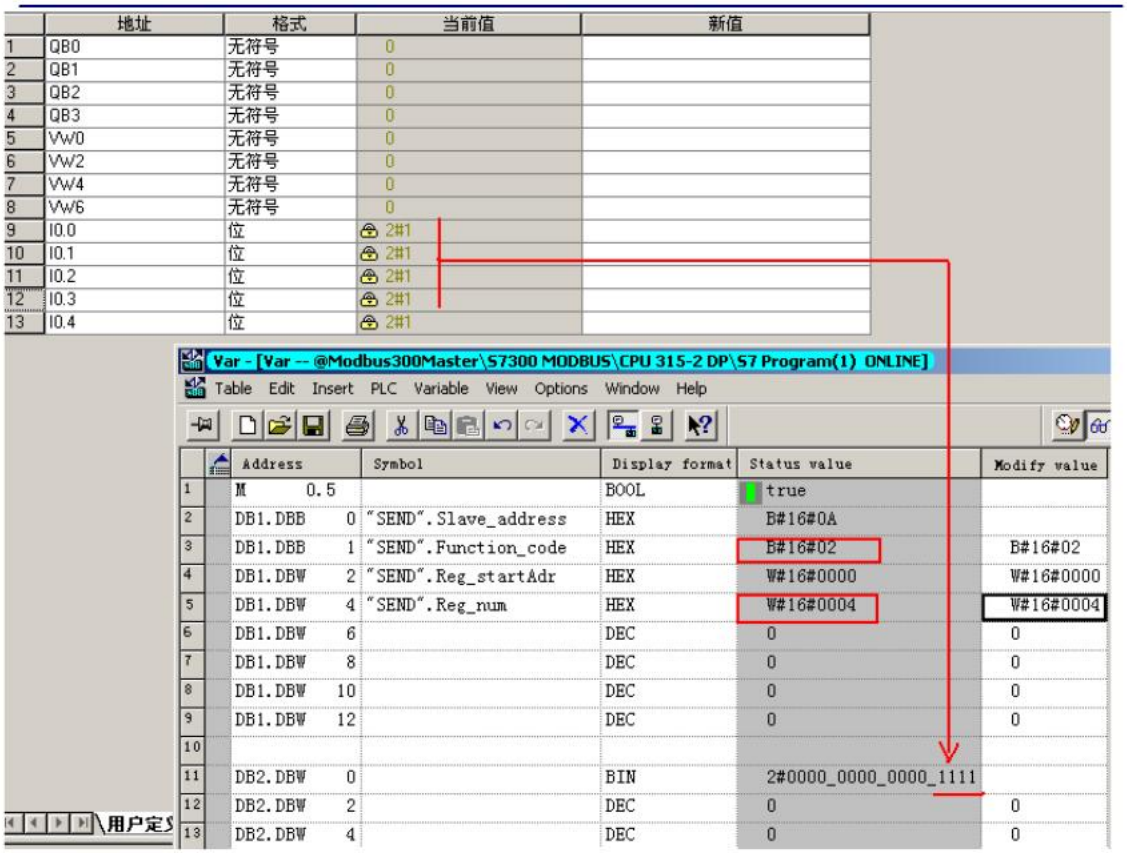


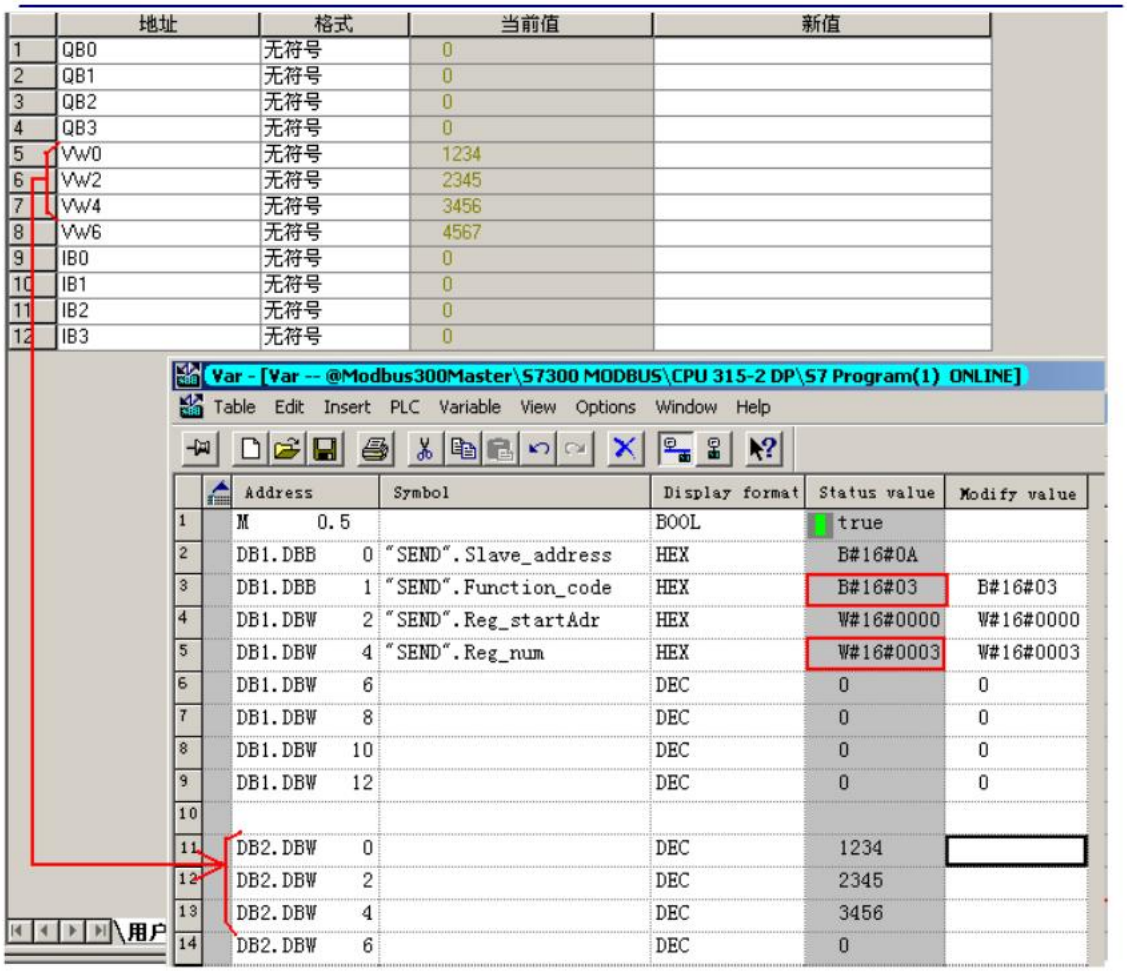
图 15 FC02 的数据交换

4.3 FC03 主站读取从站多个内部寄存器状态

FC03 功能下，“P_SND_RK”FB8 的 LEN 为 6，DB1 的头两个字节分别是所要读取从站的地址 10 号站（16#A），和功能码 03，如表 7 所示。图 16 为 S7-200 的 V 区 3 个寄存器传送到 DB2 的接收区。注意在一次请求中可以读取最多 127 个寄存器（每个寄存器 2 个字节）的数据。

地址	名称	类型	值	注释
0.0	Slave_address	BYTE	B#16#0A	从站地址
1.0	Function_code	BYTE	B#16#03	功能代码
2.0	Reg_startAdr	WORD	W#16#0	寄存器起始地址
4.0	Reg_num	WORD	W#16#3	寄存器数

表 7 FC03 的 SEND 源区域结构



	地址	格式	当前值	新值
1	QB0	无符号	0	
2	QB1	无符号	0	
3	QB2	无符号	0	
4	QB3	无符号	0	
5	VW0	无符号	1234	
6	VW2	无符号	2345	
7	VW4	无符号	3456	
8	VW6	无符号	4567	
9	IB0	无符号	0	
10	IB1	无符号	0	
11	IB2	无符号	0	
12	IB3	无符号	0	

	Address	Symbol	Display format	Status value	Modify value
1	M 0.5		BOOL	true	
2	DB1.DBB 0	"SEND".Slave_address	HEX	B#16#0A	
3	DB1.DBB 1	"SEND".Function_code	HEX	B#16#03	B#16#03
4	DB1.DBW 2	"SEND".Reg_startAdr	HEX	W#16#0000	W#16#0000
5	DB1.DBW 4	"SEND".Reg_num	HEX	W#16#0003	W#16#0003
6	DB1.DBW 6		DEC	0	0
7	DB1.DBW 8		DEC	0	0
8	DB1.DBW 10		DEC	0	0
9	DB1.DBW 12		DEC	0	0
10					
11	DB2.DBW 0		DEC	1234	
12	DB2.DBW 2		DEC	2345	
13	DB2.DBW 4		DEC	3456	
14	DB2.DBW 6		DEC	0	

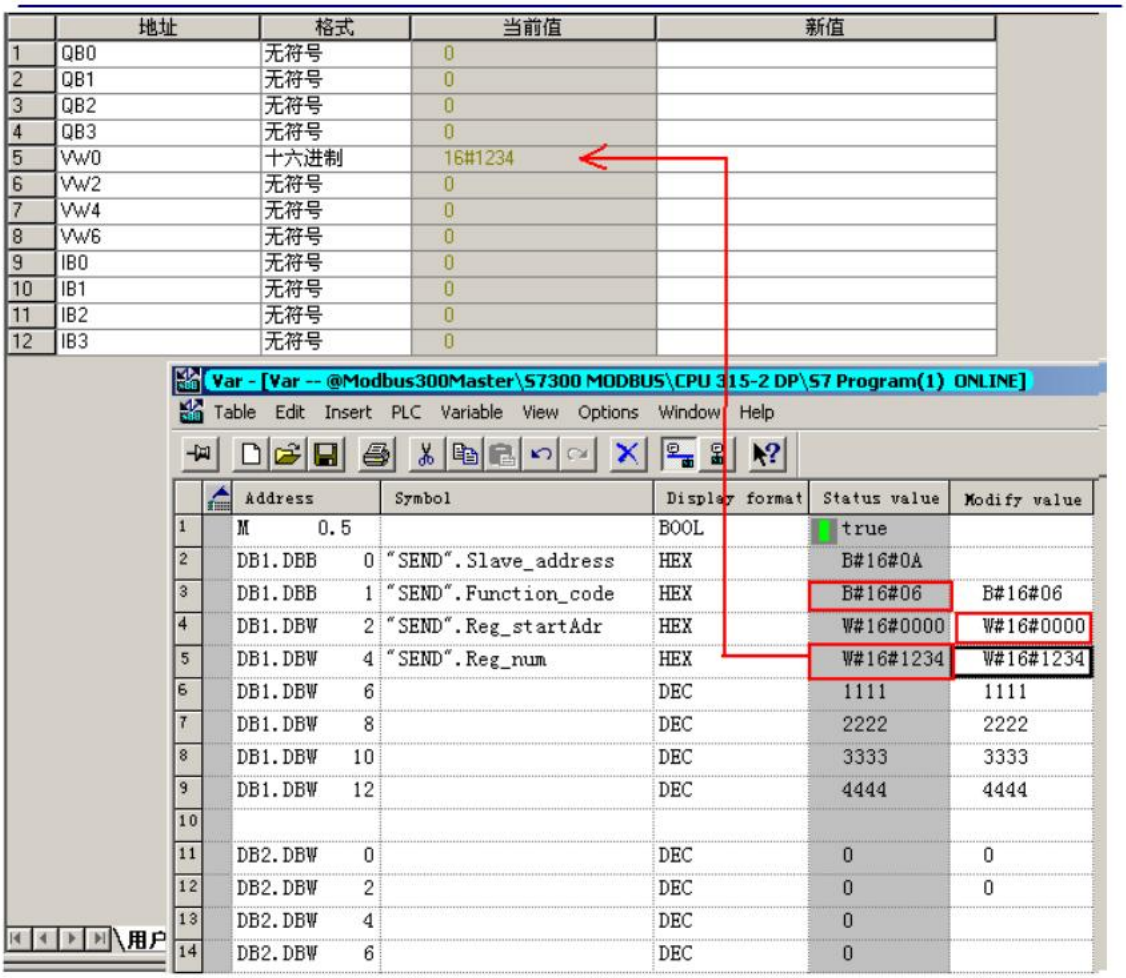
图 16 FC03 的数据交换

4.4 FC06 主站写单字到从站内部寄存器

FC06 功能下，“P_SND_RK”FB8 的 LEN 为 6，DB1 的头两个字节分别是所要读取从站的地址 10 号站（16#A），和功能码 06，如表 8 所示。图 17 为 DB1.DBW4 传送到从站 VW0 的接收区。

地址	名称	类型	值	注释
0.0	Slave_address	BYTE	B#16#0A	从站地址
1.0	Function_code	BYTE	B#16#06	功能代码
2.0	Reg_startAdr	WORD	W#16#0	寄存器地址
4.0	Reg_num	WORD	W#16#1234	寄存器值

表 8 FC06 的 SEND 源区域结构



	地址	格式	当前值	新值
1	QB0	无符号	0	
2	QB1	无符号	0	
3	QB2	无符号	0	
4	QB3	无符号	0	
5	VW0	十六进制	16#1234	
6	VW2	无符号	0	
7	VW4	无符号	0	
8	VW6	无符号	0	
9	IB0	无符号	0	
10	IB1	无符号	0	
11	IB2	无符号	0	
12	IB3	无符号	0	

	Address	Symbol	Display format	Status value	Modify value
1	M 0.5		BOOL	true	
2	DB1.DBB 0	"SEND".Slave_address	HEX	B#16#0A	
3	DB1.DBB 1	"SEND".Function_code	HEX	B#16#06	B#16#06
4	DB1.DBW 2	"SEND".Reg_startAdr	HEX	W#16#0000	W#16#0000
5	DB1.DBW 4	"SEND".Reg_num	HEX	W#16#1234	W#16#1234
6	DB1.DBW 6		DEC	1111	1111
7	DB1.DBW 8		DEC	2222	2222
8	DB1.DBW 10		DEC	3333	3333
9	DB1.DBW 12		DEC	4444	4444
10					
11	DB2.DBW 0		DEC	0	0
12	DB2.DBW 2		DEC	0	0
13	DB2.DBW 4		DEC	0	
14	DB2.DBW 6		DEC	0	

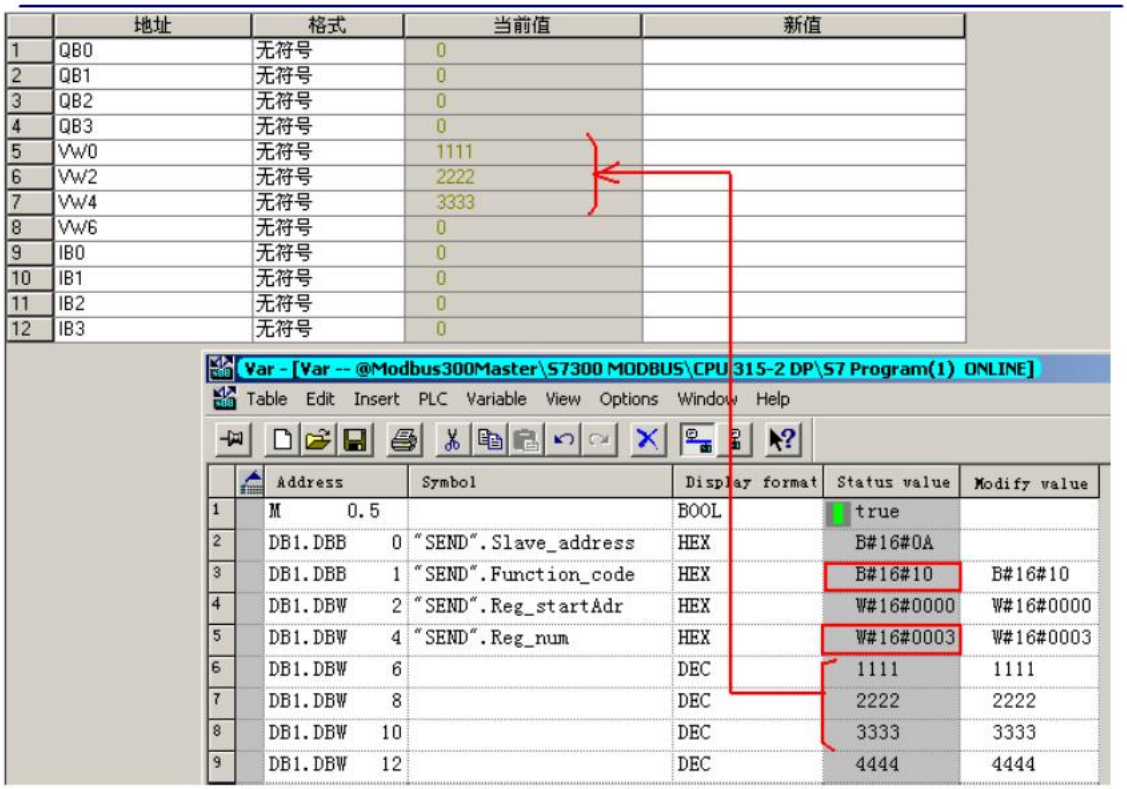
图 17 FC06 的数据交换

4.5 FC16 主站写多字到从站内部寄存器

FC16 功能下，“P_SND_RK”**FB8 的 LEN 不为 6，而是发送命令及数据的总长度本例中设为 20**。DB1 的头两个字节分别是所要读取从站的地址 10 号站（16#A），和功能码 16（16#10）。功能码 FC16 时，所要发送的数据从 DB1.DBW6 开始（从第 7 个字节开始）如表 9 所示。图 18 为 DB1.DBW6 开始的多字发送到 VW0 开始的接收区。Reg_num 为写寄存器的总数目。注意在一次请求中可以写最多 127 个寄存器（每个寄存器 2 个字节）的数据。

地址	名称	类型	值	注释
0.0	Slave_address	BYTE	B#16#0A	从站地址
1.0	Function_code	BYTE	B#16#03	功能代码
2.0	Reg_startAdr	WORD	W#16#0	寄存器起始地址
4.0	Reg_num	WORD	W#16#3	寄存器数目
6.0	Data1	WORD	W#16#0	寄存器值
8.0	Data2	WORD	W#16#0	寄存器值
10.0	Data3	WORD	W#16#0	寄存器值

表 9 FC16 的 SEND 源区域结构



	地址	格式	当前值	新值
1	QB0	无符号	0	
2	QB1	无符号	0	
3	QB2	无符号	0	
4	QB3	无符号	0	
5	VW0	无符号	1111	
6	VW2	无符号	2222	
7	VW4	无符号	3333	
8	VW6	无符号	0	
9	IB0	无符号	0	
10	IB1	无符号	0	
11	IB2	无符号	0	
12	IB3	无符号	0	

	Address	Symbol	Display format	Status value	Modify value
1	M 0.5		BOOL	true	
2	DB1.DBB 0	"SEND".Slave_address	HEX	B#16#0A	
3	DB1.DBB 1	"SEND".Function_code	HEX	B#16#10	B#16#10
4	DB1.DBW 2	"SEND".Reg_startAdr	HEX	W#16#0000	W#16#0000
5	DB1.DBW 4	"SEND".Reg_num	HEX	W#16#0003	W#16#0003
6	DB1.DBW 6		DEC	1111	1111
7	DB1.DBW 8		DEC	2222	2222
8	DB1.DBW 10		DEC	3333	3333
9	DB1.DBW 12		DEC	4444	4444

图 18 FC 16 的数据交换

注 1：如果有多个 RTU 从站需要 Modbus 主站轮询读取通信，那么可以参考**《CP341 Modbus RTU 多站点轮询》**一文，非常受益。

注 2：CP341 做从站，S7-200 做主站的通讯介绍也会随后完成。

附录一推荐网址

自动化系统
西门子（中国）有限公司
工业自动化与驱动技术集团 客户服务与支持中心
网站首页：www.4008104288.com.cn
自动化系统 **下载中心**：
http://www.ad.siemens.com.cn/download/DocList.aspx?TypeId=0&CatFirst=1
自动化系统 **全球技术资源**：
http://support.automation.siemens.com/CN/view/zh/10805045/130000
"找答案"自动化系统版区：
http://www.ad.siemens.com.cn/service/answer/category.asp?cid=1027

驱动技术
西门子（中国）有限公司
工业自动化与驱动技术集团 客户服务与支持中心
网站首页：www.4008104288.com.cn
驱动技术 **下载中心**：
http://www.ad.siemens.com.cn/download/DocList.aspx?TypeId=0&CatFirst=85
驱动技术 **全球技术资源**：
http://support.automation.siemens.com/CN/view/zh/10803928/130000
"找答案"驱动技术版区：
http://www.ad.siemens.com.cn/service/answer/category.asp?cid=1038

注意事项

应用示例与所示电路、设备及任何可能结果没有必然联系，并不完全相关。应用示例不表示客户的具体解决方案。它们仅对典型应用提供支持。用户负责确保所述产品的正确使用。这些应用示例不能免除用户在确保安全、专业使用、安装、操作和维护设备方面的责任。当使用这些应用示例时，应意识到西门子不对在所述责任条款范围之外的任何损坏/索赔承担责任。我们保留随时修改这些应用示例的权利，恕不另行通知。如果这些应用示例与其它西门子出版物(例如，目录)给出的建议不同，则以其它文档的内容为准。

声明

我们已核对过本手册的内容与所描述的硬件和软件相符。由于差错难以完全避免，我们不能保证完全一致。我们会经常对手册中的数据进行检查，并在后续的版本中进行必要的更正。欢迎您提出宝贵意见。