SIEMENS

使用入门 •11/2014

S7-1500 Modbus-RTU 使用快速入门

S7-1500、ET 200MP、ET 200SP、Modbus-RTU

http://support.automation.siemens.com/CN/view/zh/105784261

目录

1 S7-1500 Modbus-RTU 功能简介

S7-1500 可以在主机架或分布式 IO 站使用点对点通信模块来实现 Modbus-RTU 通信，如需在 S7-1500 的分布式 IO 站上实现 Modbus-RTU 通信，推荐通过 Profinet 或者 Profibus 的方式扩展 ET 200MP 或者 ET 200SP 站，通过在 ET 200MP 或者 ET 200SP 上配置 HF（高性能）的点对点通信模块来实现。

根据实际使用通信物理接口的不同，S7-1500 主机架和 ET 200MP 可使用的模板分 CM PtP RS232 HF（6ES7 541-1AD00-0AB0）和 CM PtP RS422/485 HF（6ES7 541-1AB00-0AB0） 两种，普通的点对点通信模块（型号 BA 结尾）是无法通过本文档实现 Modbus-RTU 通信 ；而 ET 200SP 的点对点通信模块只有一个，即 CM PtP（6ES7 137-6AA00-0BA0），该模块物理接口支持 RS-232/422/485。

不同的接口类型其端子接线并不相同，详见该模块接线图，链接如下：

S7-1500 CM PtP RS232 HF:

http://support.automation.siemens.com/CN/view/en/59057160/0/zh

S7-1500 CM PtP RS422/485 HF:

http://support.automation.siemens.com/CN/view/en/59061372/0/zh

ET 200SP CM PtP:

http://support.automation.siemens.com/CN/view/en/59061378/0/zh

以上介绍的模板，无论是安装在 S7-1500 主机架的模板还是分布式 IO 站的模板，都可做 Modbus-RTU 的主站或从站，通过直接调用 Modbus-RTU 相关程序块即可实现 Modbus-RTU 通信，而无需硬件狗。

S7-1500 Modbus-RTU 的特点还在于：无论点对点通信模块安装在 S7-1500 主机架还是分布式 IO 站；也无论分布式 IO 站是通过 ET 200MP 还是 ET 200SP 来实现 Modbus-RTU 通信，其组态步骤、方法、调用的程序块及注意事项等均完全相同。

S7-1500 支持的 Modbus 功能代码如下表 1 和表 2。

表 1. 用于读取数据的功能： 读取分布式 I/O

Modbus 功能代码	用于读取从站（服务器）数据的功能 - 标准寻址
01	读取输出位： 每个请求 1 至 2000/1992[1] 位
02	读取输入位： 每个请求 1 至 2000/1992[1] 位
03	读取保持寄存器： 每个请求 1 至 125/124[1] 字
04	读取输入字： 每个请求 1 至 125/124[1] 字

1) 用于扩展寻址

表 2. 用于写入数据的功能： 更改分布式 I/O 和程序数据

Modbus 功能代码	用于向从站（服务器）写入数据的功能 - 标准寻址
05	写入一个输出位： 每个请求 1 位
06	写入一个保持寄存器： 每个请求 1 个字
15	写入一个或多个输出位：每个请求 1 至 1960 位
16	写入一个或多个保持寄存器：每个请求 1 至 122 个字

2 S7-1500 Modbus-RTU 系统实例

本文以示例项目为例，阐述 S7-1500 实现 Modbus-RTU 通信的配置步骤、程序编写及注意事项，系统配置如下图 1。

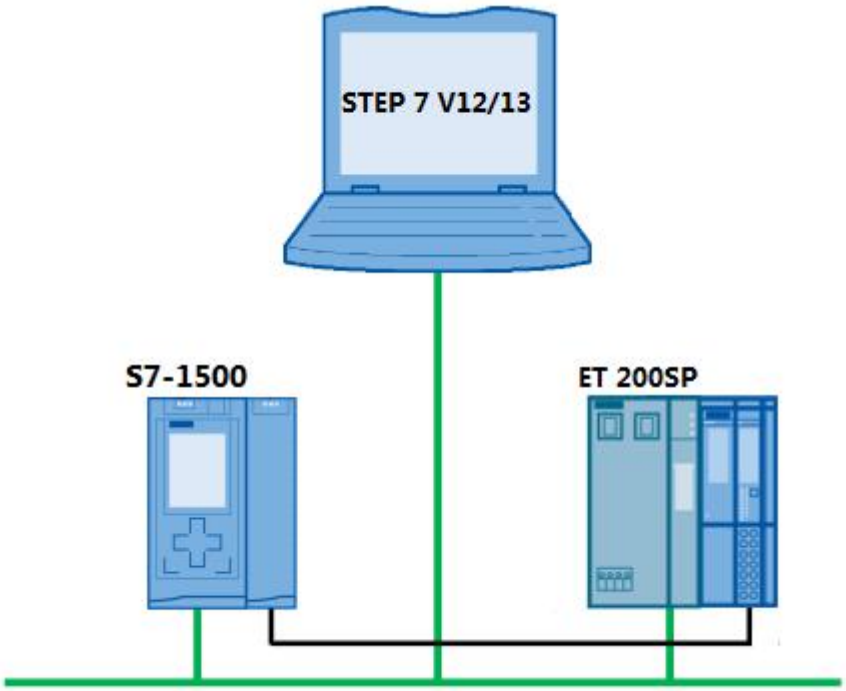

图 1 实例系统构成

该实例中，Modbus-RTU 主站为安装在 S7-1500 主机架上的 CM PtP RS422/485 HF，Modbus-RTU 从站模块（站地址为 2）为安装在 ET 200SP 分布式 IO 上的 CM PtP，接口类型为 RS485，通信波特率为 9600bit/s，无奇偶校验。

3 该实例所用的软件及硬件

SIMATIC 部件

Component	Qty.	MLFB / Order number	Note
CPU 1516-3 PN/DP	1	6ES7516-3AN00-0AB0	FW1.6
存储卡，24 MB	1	6ES7954-8LF01-0AA0	以实际容量为准
通信模块，CM PtP，RS422/485，高性能型	1	6ES7541-1AB00-0AB0	
IM 155-6 PN ST，带服务器模块和总线适配器 2xRJ45 (6ES7193-6AR00-0AA0)	1	6ES7155-6AA00-0BN0	
CM PtP（自由口，3964R，USS，Modbus RTU）	1	6ES7137-6AA00-0BA0	
BU A0 型，16 个直插式端子，2 个单独馈电端子（数字量/模拟量，最高 24VDC/10A）	1	6ES7193-6BP00-0DA0	
PC	1		

软件

Component	Qty.	MLFB / Order number	Note
SIMATIC STEP 7 Professional V13	1	6ES7822-1AA03-0YA5	Step7 V13 Update 6

4 硬件组态步骤

按照系统配置图，首先组态 S7-1516 CPU 主机架，在主机架上配置点对点通信模块，并将该模块的协议设置为“ Modbus”，如下图 2 。

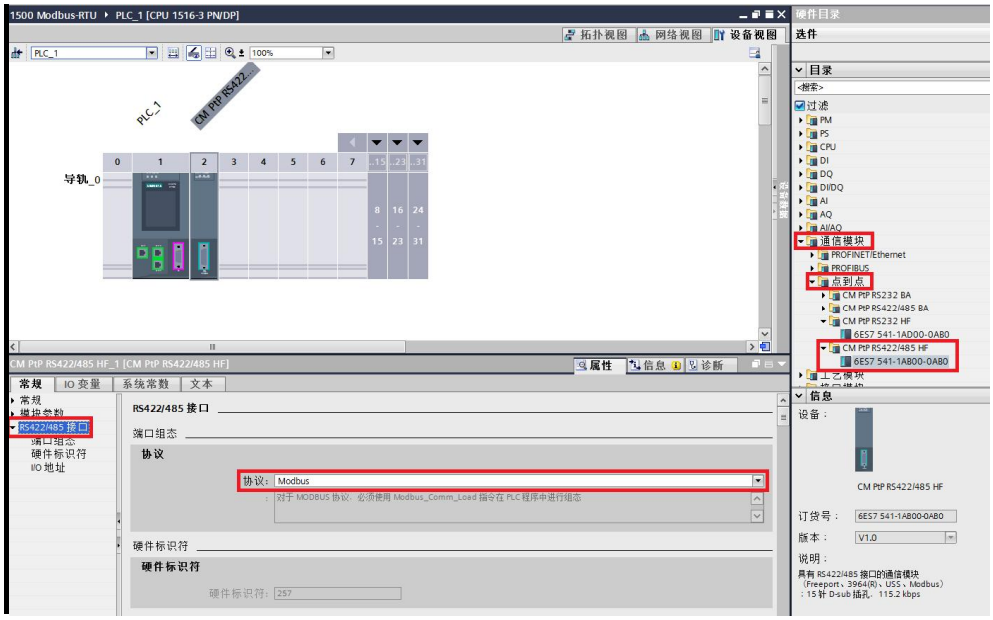

图 2 S7-1500 CPU 主机架点对点通信模块组态

然后激活 CPU 的“系统和时钟存贮器功能”，如下图 3 。（本例程中使用了 CPU 首次扫描位来实现 Modbus 的初始化，使用 CPU 的时钟信号来控制发送频率，该方法供参考。）

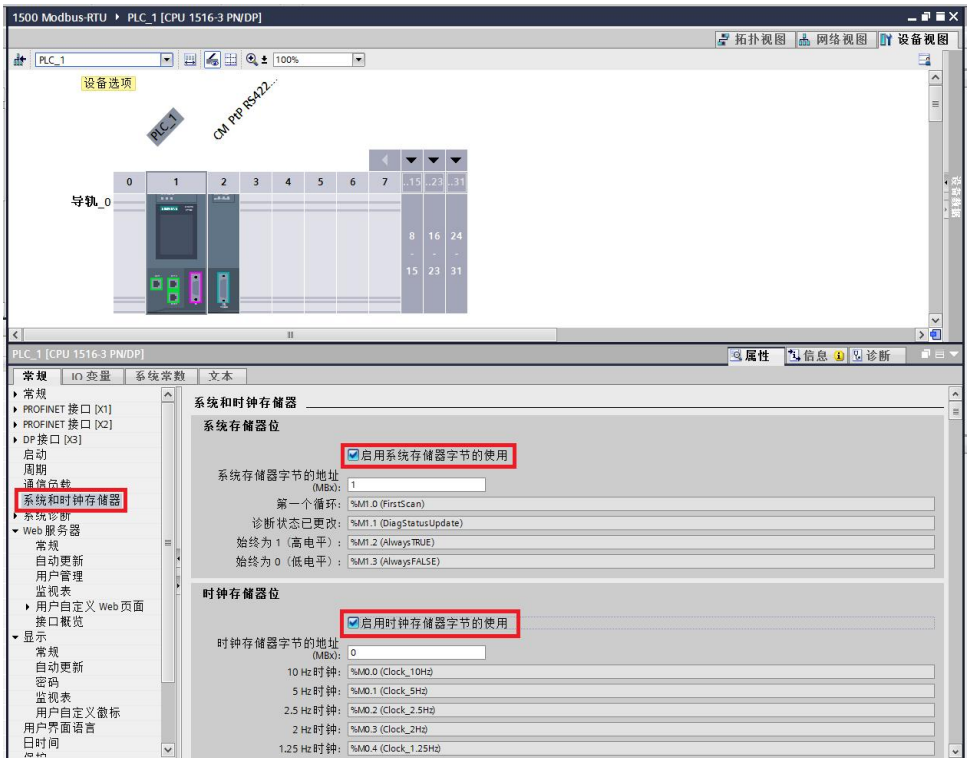

图 3 激活 CPU 的系统和时钟存储器

然后组态 ET 200SP 分布式 IO 站，并在该分布式 IO 上组态点对点通信模块，并将该模块的通信协议设置为“ Modbus”，如下图 4。

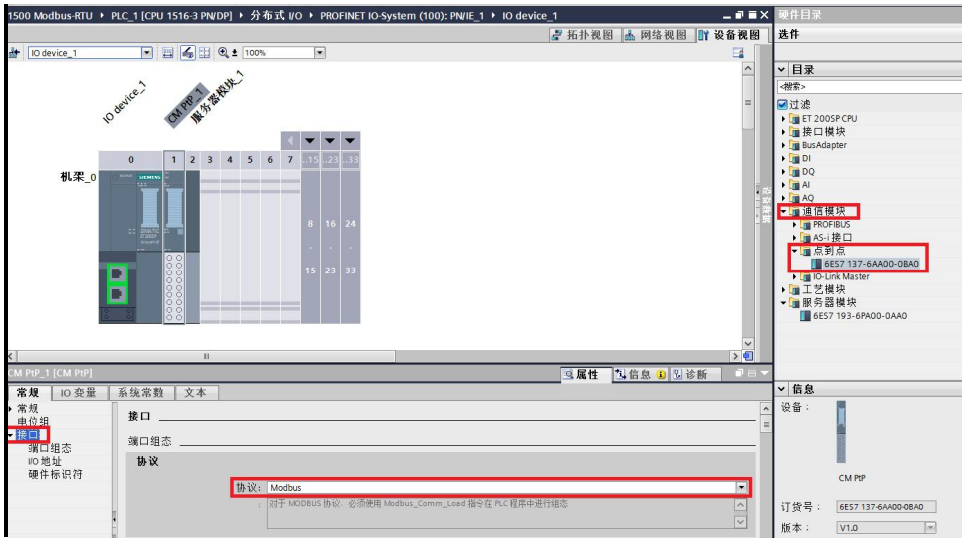

图 4 ET200 SP 分布式 IO 站点对点模块组态

至此硬件配置已完成。有关 ET 200SP 分布式 IO 组态详细步骤及如何为 Profinet 设备分配 Device Name，本使用入门不再阐述，如需帮助，请参阅下载中心文档：《ET 200SP 使用快速入门》，链接如下：

http://www.ad.siemens.com.cn/download/docMessage.aspx?ID=7363&loginID=&srno=&sendtime=

5 软件编写

S7-1500 实现 Modbus-RTU 功能，需要调用以下指令，见下表 3，其中“ Modbus_Comm_Load”指令用于通信模块的组态，“ Modbus_Master ” 指令和“ Modbus_Slave ” 指令分别实现 Modbus 主站通信和 Modbus 从站通信。“ Modbus_Comm_Load”指令和“ Modbus_Master ”/ “ Modbus_Slave ”指令是通过“ Modbus_Comm_Load”指令的“ MB_DB”参数来实现关联的。

表 3. Modbus-RTU 相关指令

指令	含义
Modbus_Comm_Load	指令 Modbus_Comm_Load 允许组态 Modbus RTU 的通信模块端口。
Modbus_Master	Modbus_Master 指令允许通过 PtP 端口作为 Modbus 主站进行通信。
Modbus_Slave	Modbus_Slave 指令允许通过 PtP 端口作为 Modbus 从站进行通信。

在此，首先编写 Modbus 主站程序，添加一个新 FB，将其命名为“ Modbus-Master”，如下图 5。

图 5 添加 Modbus-Master 功能块

在该 FB 中以多重背景方式调用“ Modbus_Comm_Load”指令，该指令在指令目录下“通信—〉通信处理器—〉Modbus（RTU）”下，如下图 6。

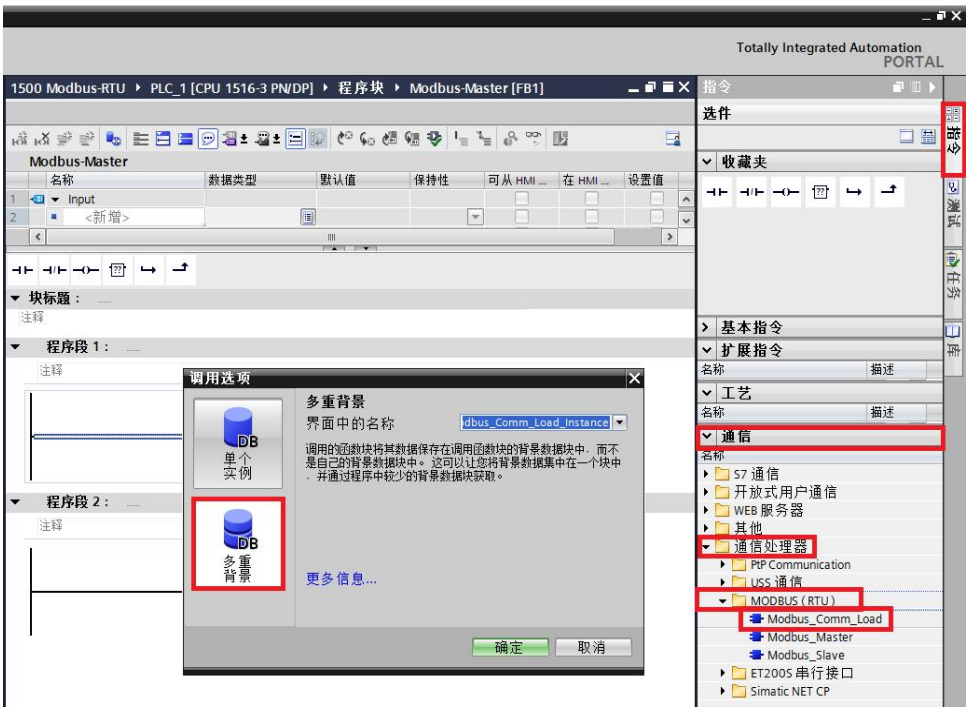

图 6 调用“ Modbus_Comm_Load ”

在该 FB 中以多重背景方式调用“ Modbus_Master”指令，该指令在指令目录下“通信—〉通信处理器—〉Modbus（RTU）”下，如下图 7。

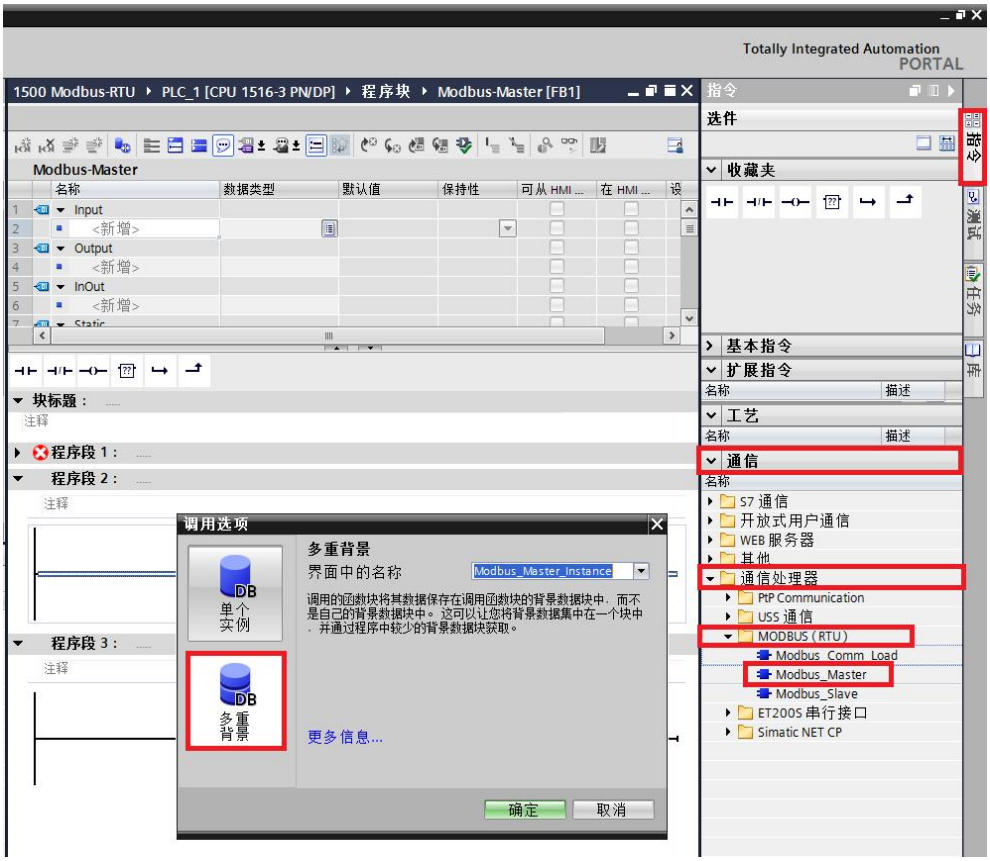

图 7 调用“ Modbus_Master”指令

然后对“ Modbus_Comm_Load”指令进行参数化，由于该指令参数较多，在此只列出必须要关注的参数，如下表 4 所示，其它参数解释见手册或在线帮助。

表 4. Modbus_Comm_Load 主要参数列表

参数	声明	数据类型	缺省值	说明
REQ	IN	Bool	FALSE	当此输入出现上升沿时，启动该指令。
PORT	IN	Port	0	CM 端口值，即“硬件 ID”(Hardware ID) 。符号端口名称在 PLC 变量表的“系统常数”(System constants) 选项卡中指定。
BAUD	IN	UDInt	9600	选择数据传输速率 有效值为： 300, 600, 1200, 2400, 4800, 9600, 19200, 38400, 57600, 76800, 115200 bit/s.
PARITY	IN	UInt	0	选择奇偶校验： ● 0 – 无 ● 1 – 奇校验 ● 2 – 偶校验

MB_DB	IN/OUT	MB_BASE		对 Modbus_Master 或 Modbus_Slave 指令的背景数据块的引用。 MB_DB 参数必须与 Modbus_Master 或 Modbus_Slave 指令的静态参数 MB_DB 关联。
MODE	Static	USInt	0	工作模式 有效的工作模式包括： ● 0 = 全双工 (RS232) ● 1 = 全双工 (RS422) 四线制操作（点对点） ● 2 = 全双工 (RS 422) 四线制模式（多点主，CM PtP (ET 200SP)） ● 3 = 全双工 (RS 422) 四线制模式（多点从，CM PtP (ET 200SP)） ● 4 = 半双工 (RS485) 二线制模式

首先要为“ Modbus_Comm_Load”指令指定端口，即该指令是针对哪个点对点模块进行参数化的。在硬件配置中，每个硬件均有一个硬件标识符，该硬件标识符在硬件属性中可以查看到，如下图 8。

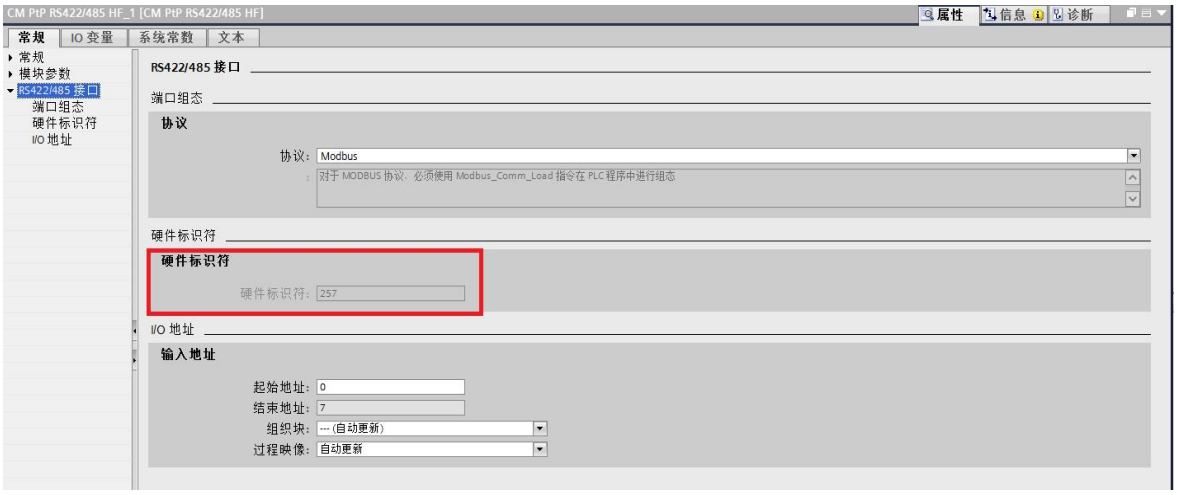

图 8 在硬件属性中查看模块硬件标识符

同样，该硬件标识符也可以在“ PLC 变量—〉显示所有变量—〉系统变量”下可以查看到，如下图 9。

图 9 PLC 变量表中查看系统常量

所以可以通过如图 10 所示方法，通过拖拽的方式，将 Modbus 主站接口的硬件标识符拖至“ Modbus_Comm_Load”指令的“ Port”接口参数处，如下图 10。

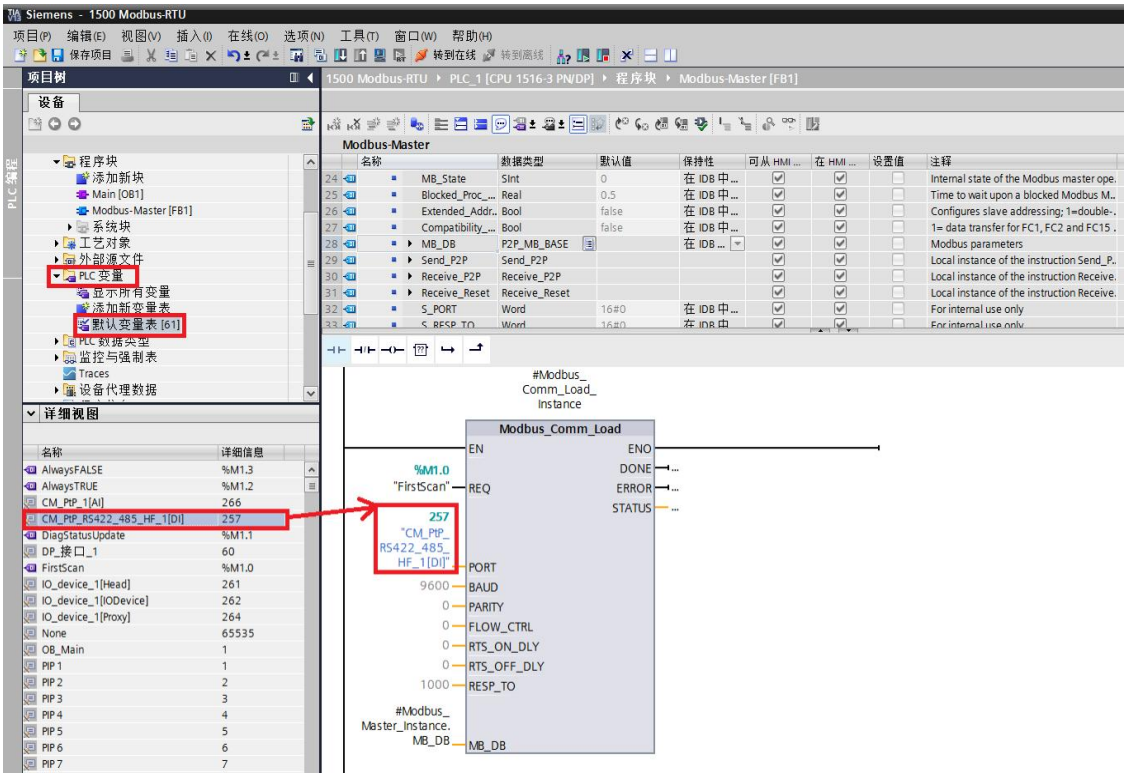

图 10 为“ Modbus_Comm_Load”指定端口

接下来，定义端口的工作模式，本示例中，点对点模块的工作模式为 RS485，所以需要将“ Modbus_Comm_Load”背景数据中静态变量的“ MODE”参数赋值为 4，赋值既可以通过“ Move”指令来完成，也可以通过直接修改该静态变量的默认值来实现，本实例使用后一种方法，参见下图 11。

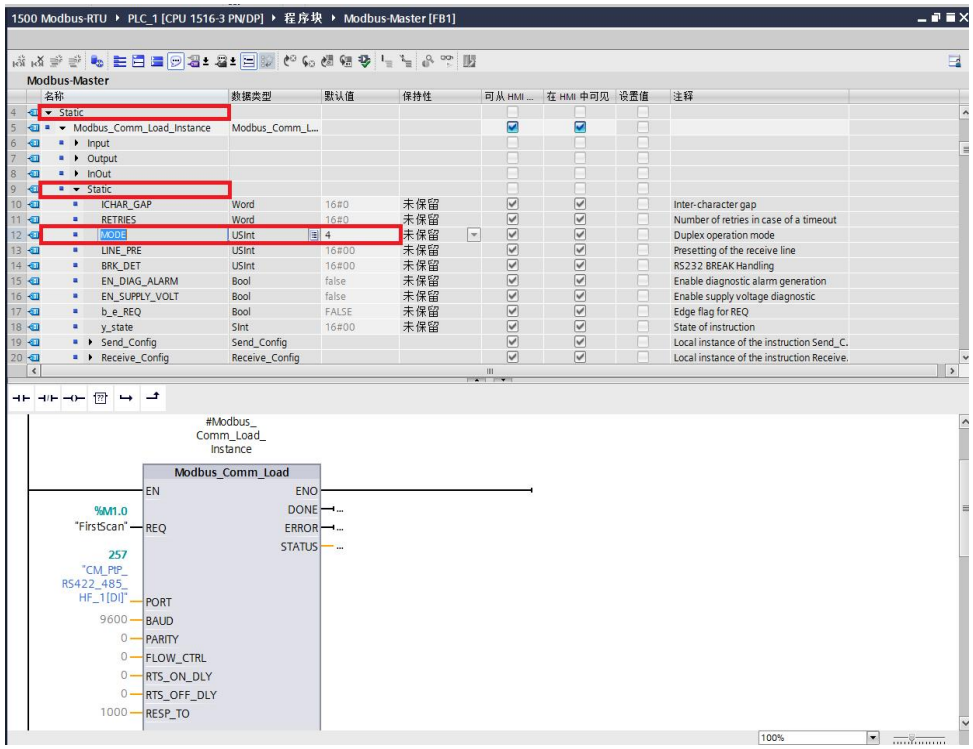

图 11 定义 Modbus-RTU 主站端口工作模式为 RS485

接下来，通过对“Modbus_Comm_Load”指令的“MD_DB”参数赋值，将“Modbus_Comm_Load”指令与“Modbus_Master”指令进行关联，即将“Modbus_Master”指令的背景 DB 块中静态变量“MB_DB”赋值给“Modbus_Comm_Load”指令的“MD_DB”，可以通过拖拽的方式来实现，拖拽路径如下图 12。

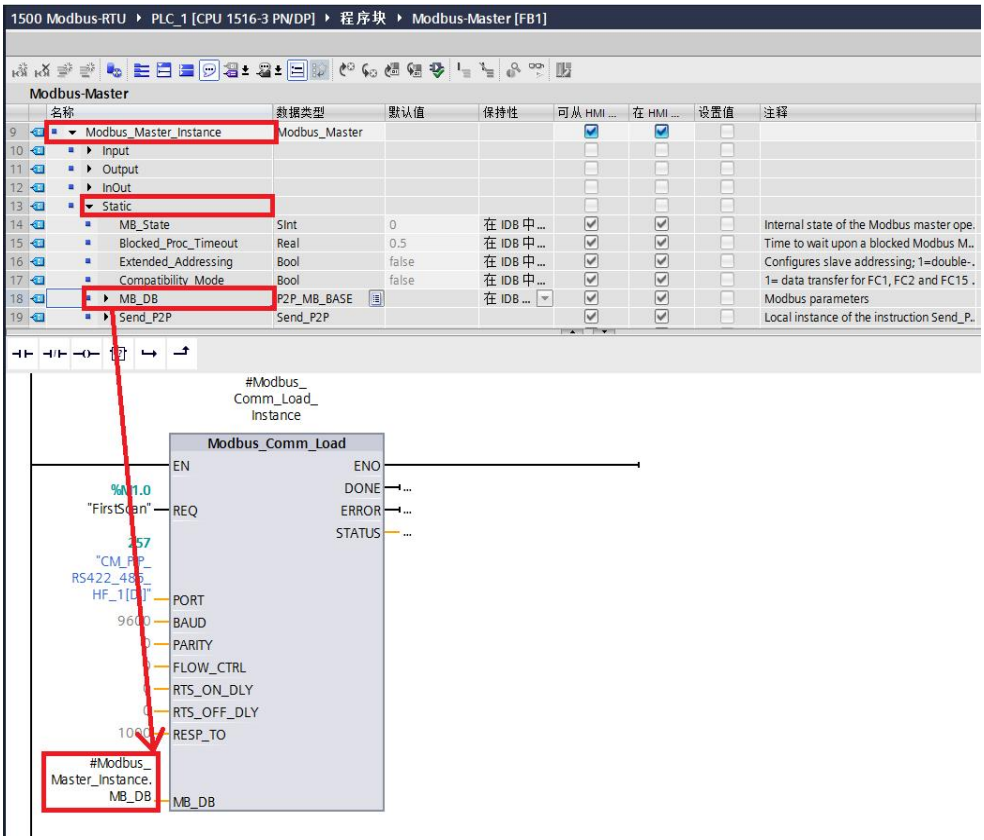

图 12 对“ Modbus_Comm_Load”指令的“ MD_DB”参数赋值

除以上操作外，对于“ Modbus_Comm_Load”指令的“ REQ”参数，本实例使用 PLC 的首个扫描位来完成。其它参数如波特率，奇偶校验等，请根据实际使用情况对这些参数进行赋值，因本实例波特率为 9600bit/s，无奇偶校验，所以以上参数使用缺省设置即可。

接下来，对指令“ Modbus_Master”进行参数设置，该指令主要参数如下表 5 所示。

表 5. Modbus_Master 主要参数列表

参数	声明	数据类型	缺省值	说明
REQ	IN	Bool	FALSE	FALSE = 无请求 TRUE = 请求向 Modbus 从站发送数据
MB_ADDR	IN	UInt	0	Modbus RTU 站地址： 标准地址范围（1 到 247 以及 0，用于 Broadcast） 扩展地址范围（1 到 65535 以及 0，用于 Broadcast） 值 0 为将帧广播到所有 Modbus 从站预留。 广播仅支持 Modbus 功能代码 05、06、15 和 16。
MODE	IN	USInt	0	模式选择： 指定请求类型（读取、写入或诊断）。

DATA_ADDR	IN	UDInt	0	从站中的起始地址： 指定在 Modbus 从站中访问的数据的起始地址。
DATA_LEN	IN	UInt	0	数据长度： 指定此指令将访问的位或字的个数。
DATA_PTR	IN/OUT	Variant		数据指针： 指向要进行数据写入或数据读取的标记或数据块地址。

由于 Modbus 指令读取或写入的数据区必须为指针寻址，所以必须是有绝对地址的区域方可访问，而 S7-1500 创建的 DB 块缺省为优化的 DB 块，变量没有绝对地址，无法直接使用。本例中我们创建 DB 块，并在该 DB 块内创建一个名为“M_Data”的数组，类型为 WORD，长度为 100，即创建了 100 个字的存储空间。然后在该 DB 块点右键，在属性中将“优化的块访问”前的勾去掉，重新编译该 DB 块，该 DB 块就会生成，可以看到每个变量都有偏移地址了，如下图 13。

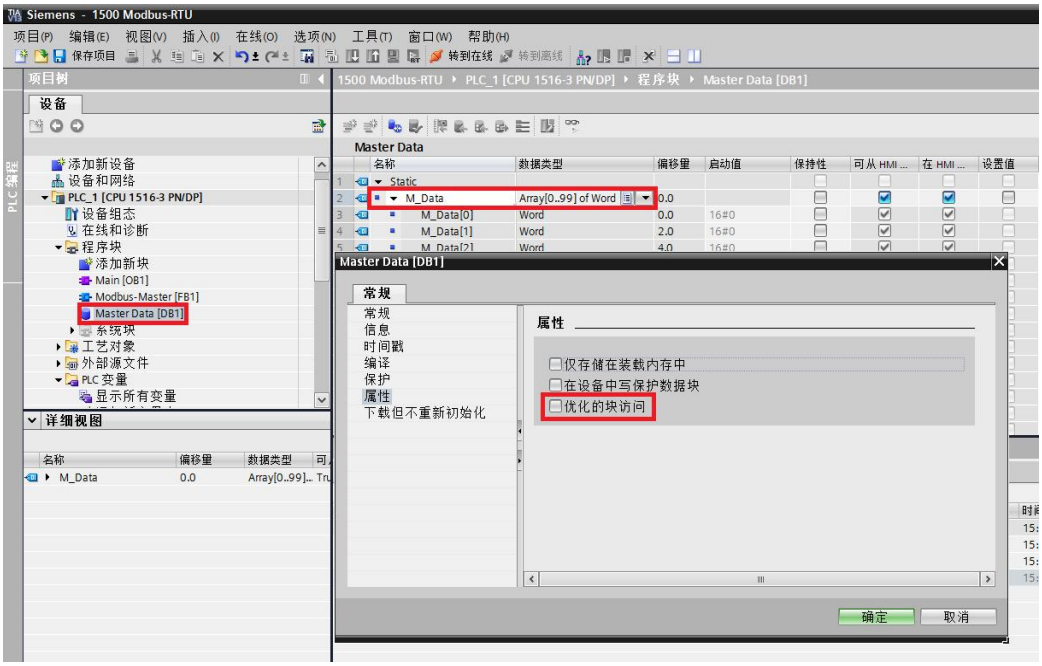

图 13 创建一个标准 DB 块

根据下表 6 所示的 Modbus 功能码，本实例需要 Modbus 主站读取 Modbus 从站保持寄存器从起始地址开始的 10 个字的内容到创建的“ Master_Data”中，即 Modbus 功能码 03 的功能：

表 6. Modbus 功能码的选择

MODE	DATA_ADDR（Modbus 地址）			DATA_LEN（数据长度）			Modbus 功能代码	运行和数据		
0				每个请求的位数			01	读取输出位：		
	1	到	9999	1	到	2000/1992 [1]		0	到	9998
0				每个请求的位数			02	读取输入位：		
	10001	到	19999	1	到	2000/1992 [1]		0	到	9998
0				每个请求的字数			03	读取保持寄存器：		
	40001	到	49999	1	到	125/124 [1]		0	到	9998
	400001	到	465535	1	到	125/124 [1]		0	到	65534
0				每个请求的字数			04	读取输入字：		
	30001	到	39999	1	到	125/124 [1]		0	到	9998
1				每个请求的位数			05	写入一个输出位：		
	1	到	9999	1				0	到	9998
1				每个请求 1 个字			06	写入一个保持寄存器：		
	40001	到	49999	1				0	到	9998
	400001	到	465535	1				0	到	65524
1				每个请求的位数			15	写入多个输出位：		
	1	到	9999	2	到	1968/1960 [1]		0	到	9998
1				每个请求的字数			16	写入多个保持寄存器：		
	40001	到	49999	2	到	123/122		0	到	9998
	400001	到	465534	2	到	123/122 [1]		0	到	65534
2 [2]				每个请求的位数			15	写入一个或多个输出位：		
	1	到	9999	2	到	1968/1960 [1]		0	到	9998
2 [2]				每个请求的字数			16	写入一个或多个保持寄存器：		
	40001	到	49999	1	到	123		0	到	9998
	400001	到	465535	1	到	122 [1]		0	到	65534
11	此功能将忽略 Modbus_Master 的 DATA_ADDR 和 DATA_LEN 操作数。						11	读取从站通信的状态字和事件计数器。 状态字表示“忙”（0 - 不忙，0xFFFF - 忙）。 事件计数器随着帧的每次成功处理而递增。		
80				每个请求 1 个字			08	使用数据诊断代码 0x0000 检查从站状态（回送测试 - 从站返回请求的回应）		
	-			1				-		
81				每个请求 1 个字			08	利用数据诊断代码 0x000A 重新设置从站事件计数器		
	-			1				-		
3 到 10，12 到 79，82 到 255	-			-				保留		

根据以上要求，则“ Modbus_Master”指令应按如下赋值：

“ REQ”：本实例使用 PLC 时钟信号来完成，即下图中的 M0.5；

“ MB_ADDR”：2 //访问的从站地址；

“ MODE”：0；//与“ DATA_ADDR”参数一起决定 Modbus 功能码为 03

“ DATA_ADDR ”:40001 //Modbus 地址

“ DATA_LEN”：10 //数据长度为 10 个字

“ DATA_PTR”：该参数可以通过拖拽的方式，将创建的标准 DB 内的变量“ M_Data”拖拽到“ DATA_PTR”处，拖拽路径如下图 14。

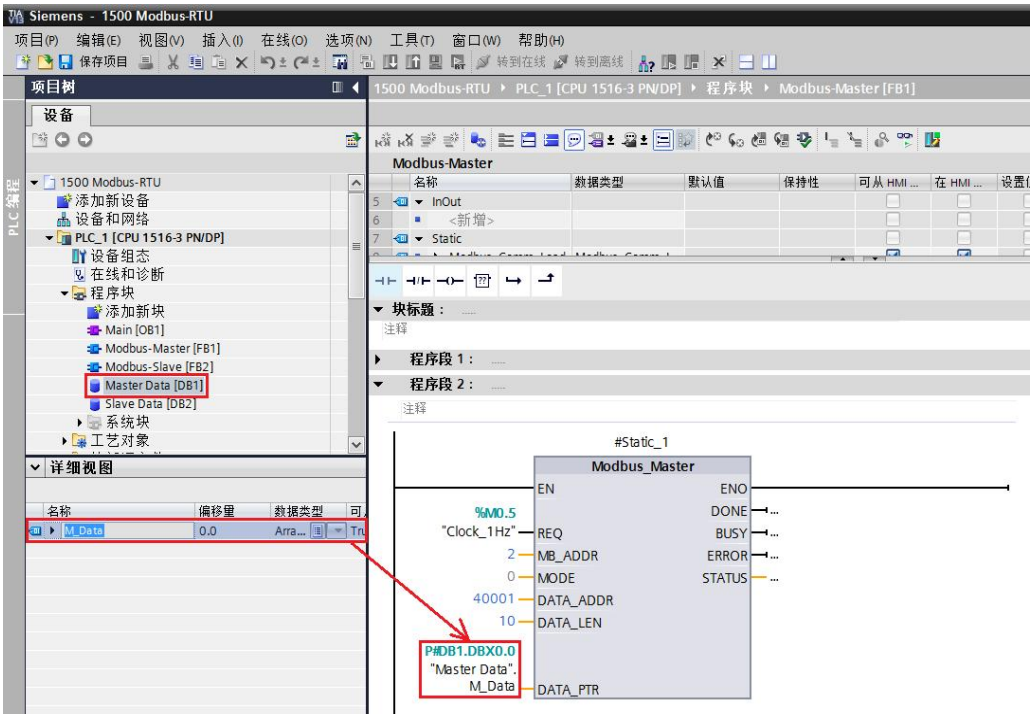

图 14 为“ Modbus_Master”指令进行参数赋值

接下来，编写 Modbus_RTU 从站程序，参照主站的程序，添加一个新 FB 2“Modbus-Slave”，在该 FB 中以多重背景方式调用“ Modbus_Comm_Load ”指令，同样为该指令选择 ET 200SP 的硬件地址，其它通信参数如波特率、奇偶校验等与主站的“ Modbus_Comm_Load”指令相同，如下图 15 。

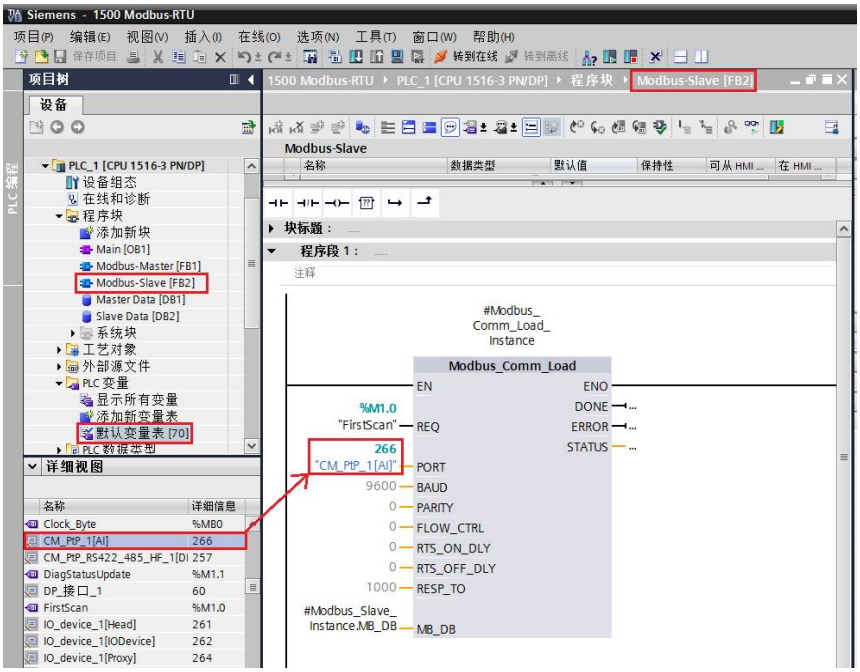

图 15 在从站 FB 中调用“Modbus_Comm_Load”指令

使用定义主站端口的工作模式相同的方法，将从站点对点模块的工作模式定义为 RS485，所以需要将“Modbus_Comm_Load”背景数据中静态变量的“MODE”参数赋值为 4，如下图 16。

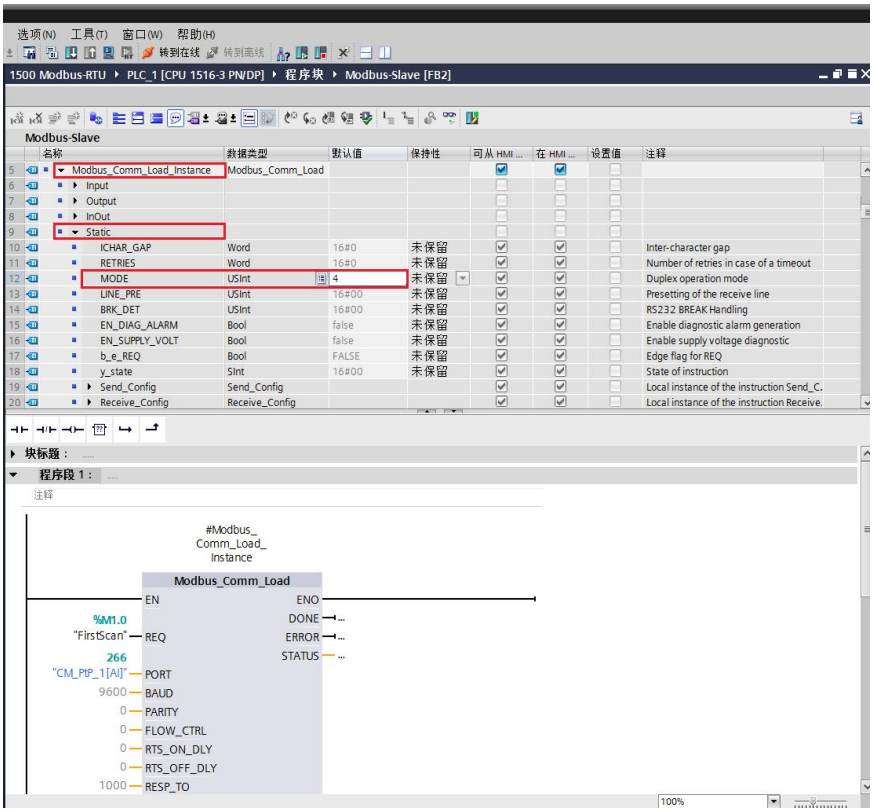

图 16 定义 Modbus-RTU 从站端口工作模式为 RS485

然后，以多重背景的方式调用从站指令“ Modbus_Slave”指令，并设置 Modbus 从站地址为 2，并为从站创建一个标准的 DB 块“ Slave Data ”，长度根据实际情况定，本例中在“ Slave Data ”中创建了一个长度为 200 字的数组，并将该变量以拖拽的形式（也可以通过指针 P#的方式），将该变量填在“Modbus_Slave”的“ MB_HOLD_REG”参数处，如下图 17。

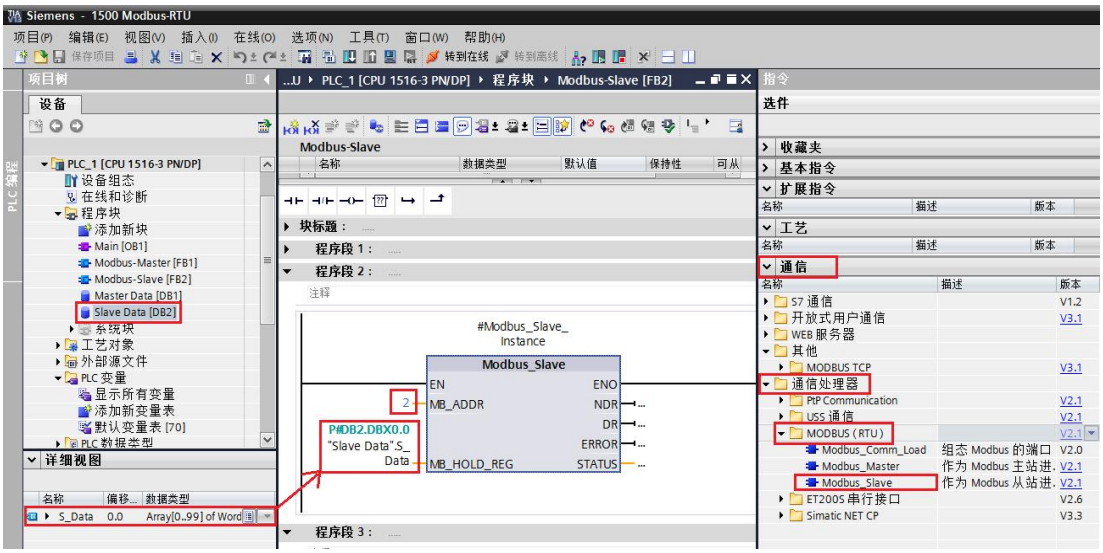

图 17 调用“ Modbus_Slave”指令

然后对“ Modbus_Comm_Load”指令的“ MD_DB”参数赋值，将“ Modbus_Comm_Load”指令与“ Modbus_ Slave”指令进行关联，即将“ Modbus_Slave”指令的背景 DB 块中静态变量“ MB_DB”赋值给“ Modbus_Comm_Load”指令的“ MD_DB”，可以通过拖拽的方式来实现，拖拽路径如下图 18。

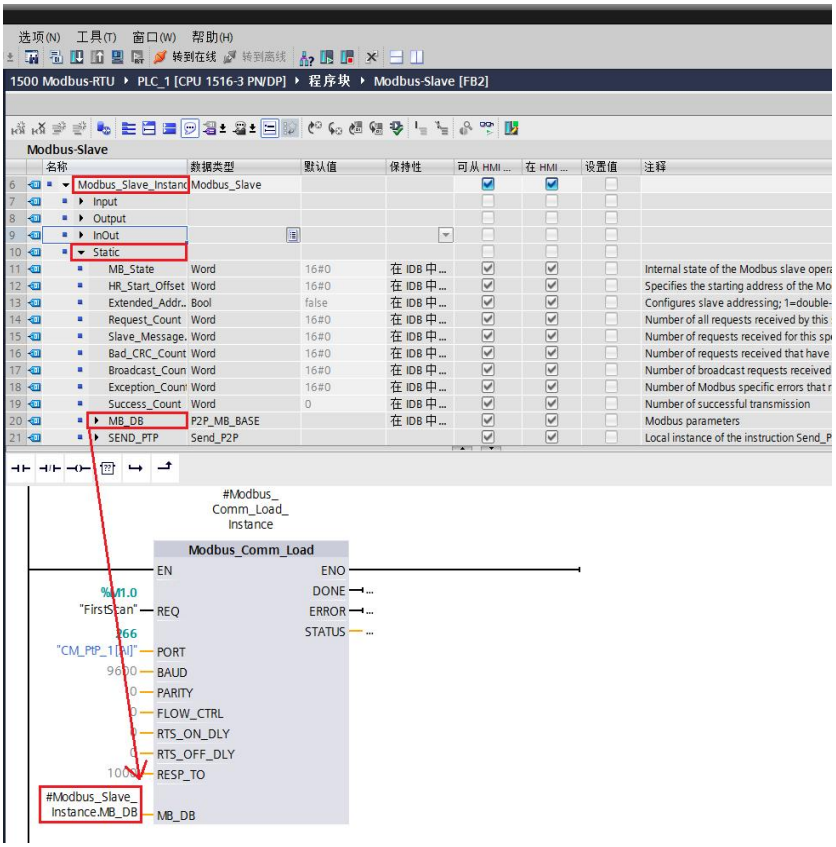

图 18 将“ Modbus_Slave ” 指令和“ Modbus_Comm_Load”指令关联

在 OB1 中分别调用 Modbus 主站程序块和 Modbus 从站程序块，并为其分配全局 DB 块，如下图 19。

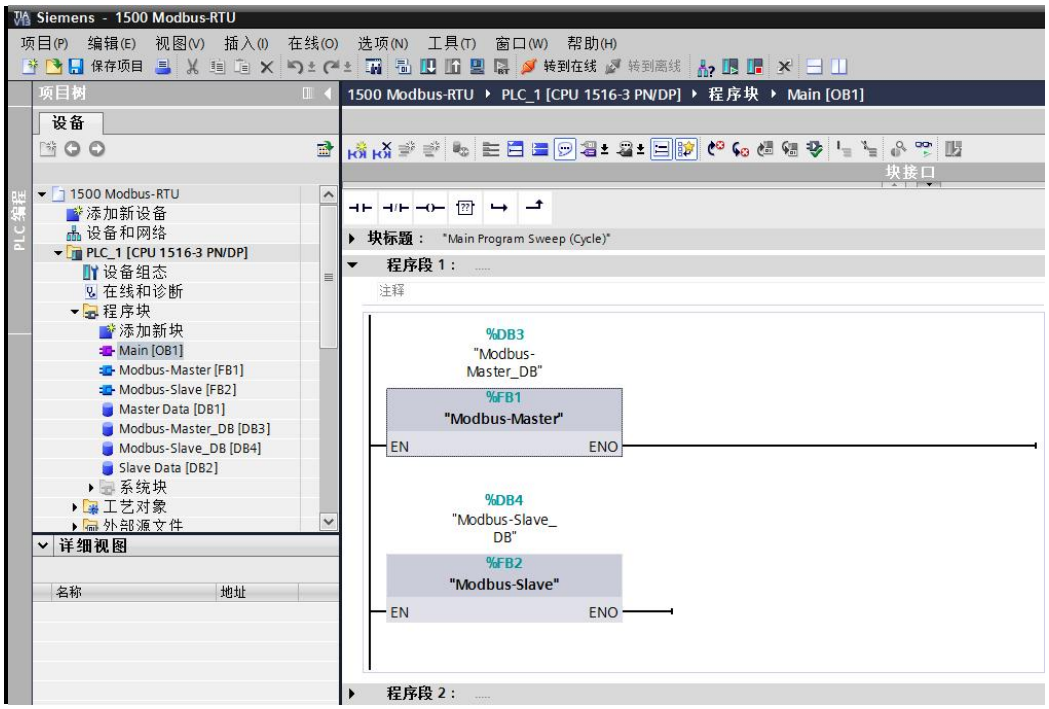

图 19 在 OB1 中分别调用主站程序和从站程序

至此，程序编写基本结束，建立 2 个变量监视表，分别用来监视 Modbus 主站数据和 Modbus 从站数据，将该例程下载到 PLC 后，可以看到通信模块对应的收发 LED 指示灯在闪烁，表示端口正在发送/接收数据。将 DB 块“ Slave Data ”中的变量赋值，监视主站“ Master Data ”中的变量，可以看到主站已经读取到从站的数据，如下图 20。

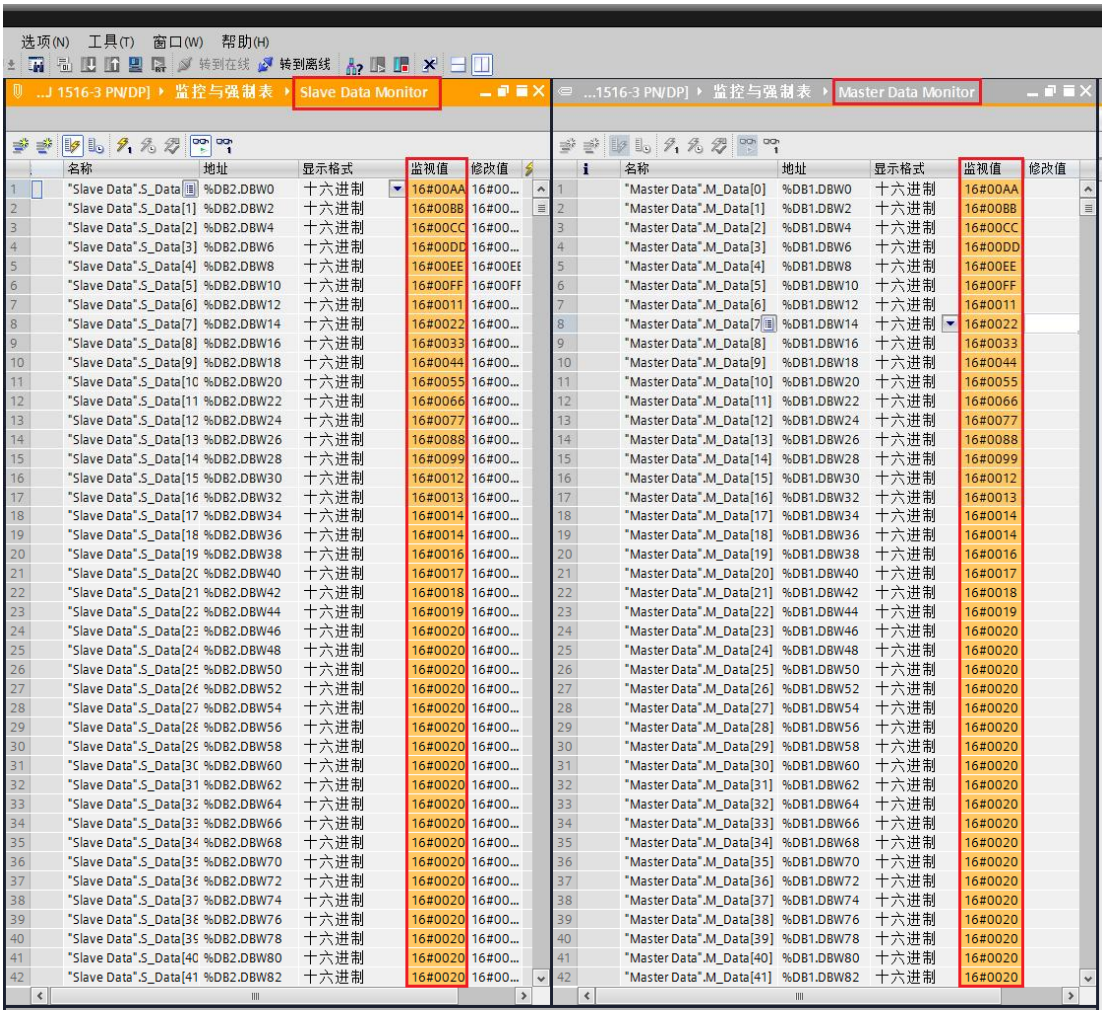

图 20 使用变量监视表测试，主站侧已成功读取到从站数据

Modbus 其它功能码使用方法类似，请参照表 6 修改相应的变量即可，此处不再一一举例。

也可以根据此例程，举一反三，编写 Modbus-RTU 轮询程序。

6 注意事项

Modbus 通信，不论是主站侧还是从站侧，其收发数据区必须是可通过指针寻址的，即必须是有确切地址的数据区；而优化的 DB 块由于其内部变量没有地址，所以收发数据区均不能使用优化的 DB 块，必须使用标准 DB，否则 PLC 运行会报错。

除 Modbus 收发数据区必须使用标准的 DB 块，其它部分如程序等均可使用优化的程序块实现。

“ Modbus_Comm_Load”指令的初始化信号端“ REQ”和“ Modbus_Master ”指令的发送/接收信号端“ REQ”必须使用边沿信号触发，否则初始化、发送/接收都无法完成。

7 常见错误

- 没有定义端口的工作模式。

  如未定义，缺省模式为 RS232。
- 收发数据区使用了优化的 DB。

  将优化的 DB 修改为绝对 DB。
- “ Modbus_Comm_Load”初始化未执行。

  重新执行初始化指令。
- “ Modbus_Master”指令输入接口参数“ DATA_LEN”和“ DATA_PTR”不匹配，无法实现收发。

  “ DATA_LEN”必须小于等于“ DATA_PTR”指向的数据存储区。
- 点对点通信模块的“ TX ”/“ RX”或“ TXD ”/“ RXD”灯无闪烁。

  检查“ Modbus_Comm_Load”初始化参数，确保其被正确初始化；检查“ Modbus_Master”参数和“ Modbus_ Slave”指令参数，确保参数正确。

8 本例程下载地址

http://support.automation.siemens.com/CN/view/zh/105784261