SIEMENS

操作指南 • 11 月 2014 年

# TIA 中使用 ET200MP/SP PTP 模块的 Modbus RTU 通信入门

http://support.automation.siemens.com/CN/view/zh/106313008

# 目录

# 1 Modbus RTU通讯概述

Modbus 通信协议是 OSI 模型第 7 层上的应用层报文传输协议，是一种广泛应用的公开协议，它已经成为一种通用的工业标准。不同厂商生产的控制设备可以通过 Modbus 通信协议连接到工业网络，进行集中控制。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 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,分布式 IO ET200SP/ET200MP 系列都推出了 PTP 模块，包含 RS 232，RS 422 和 RS 485 接口，并且都可以安装在分布式 IO 上，通过 Profibus 或 Profinet 的方式与主站相连，此种方案很适合比较大型系统进行的 Modbus 通信设计和改造（特别需要注意的是，ET200MP 所带的 PTP 模块也可以直接和 S7-1500 CPU 安装在一个机架上使用）。

本文将通过简单的 Modbus 主、从通信例程，并配合软件 modscan32 和 modsim32 做通信测试，描述在 TIA 博途软件中，如何实现 S7-1500/300/400 通过分布式 IO ET200SP/ET200MP 的 PTP 模块做 modbus rtu 通信。

注意：由于 ET200MP PTP 模块和 ET200SP PTP 模块的使用完全一致，故本文中的实验都使用 ET200SP PTP 模块进行测试。

# 2 S7-1500 的分布式IO中使用ET200SP的ptp模块

## 2.1 硬件和软件需求

| 名称 | 数量 | 订货号 |
|---|---|---|
| 电源模块 PM190W 120~230VAC | 1 | 6EP1333-4BA00 |
| CPU 1511 | 1 | 6ES7 511-1AK00-0AB0 |
| ET200SP IM155-6 PN ST | 1 | 6ES7 155-6AU00-0BN0 |
| ET200SP CM PTP | 1 | 6ES7 137-6AA00-0BA0 |
| PC，带232串口 | 1 | |
| RS232转RS485转换器 | 1 | |
| 网线 | 若干 | |

表 2-1 硬件订货信息

| 名称 | 订货号 |
|---|---|
| TIA PORTAL V13 professional | 6ES7 822-1AA03-0YA5 |
| Modscan32 用于在 PC 中模拟主站 | |
| Modsim32 用于在 PC 中模拟从站 | |

表 2-2 软件订货信息

## 2.2 硬件接线

对于 ET200SP/ET200MP 的 PTP 模块接线，请参考模块手册。

SIMATIC ET200SP CM PtP RS232/422/485 手册：

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

SIMATIC S7-1500 CM PtP RS422/485 HF 手册：

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

SIMATIC S7-1500 CM PtP RS232 HF 手册：

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

本例中使用的 ET200SP CM PTP 模块，测试时使用 RS485 接口，根据手册中的信息，端子 14 为信号正极，端子 12 为信号负极；接线方式，如图 2-1 和 2-2 所示：

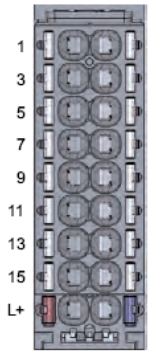
| 通信模块 BaseUnit 的端子分配 | 针脚 | 标识 | 输入/输出 | 含义 |
|---|---|---|---|---|
|  | 11 | T (A) - | 输出 | 发送数据（四线制模式） |
| | 12 | R (A) -<br>T(A)/R(A) | 输入<br>输入/输出 | 接收数据（四线制模式）<br>接收/发送数据（两线制模式） |
| | 13 | T (B) + | 输出 | 发送数据（四线制模式） |
| | 14 | R (B) +<br>T(B)/R(B) | 输入<br>输入/输出 | 接收数据（四线制模式）<br>接收/发送数据（两线制模式） |
| | 15+16 | PE 接地 | - | GND 功能性接地（隔离） |

图 2-1 RS422/485 连接端子图

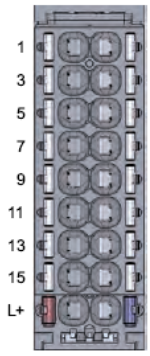
| 通信模块 BaseUnit 的端子分配 | 针脚 | 标识 | 输入/输出 | 含义 |
|---|---|---|---|---|
|  | 1 | TXD 传输数据 | 输出 | 传输数据 |
| | 2 | RXD 接收数据 | 输入 | 接收数据 |
| | 3 | RTS 请求发送 | 输出 | 请求发送 |
| | 4 | CTS 清除以发送 | 输入 | 清除以发送 |
| | 5 | DTR 数据终端准备就绪 | 输出 | 数据终端准备就绪 |
| | 6 | DSR 数据集准备就绪 | 输入 | 数据集准备就绪 |
| | 7 | DCD 数据载体检测 | 输入 | 接收的信号电平 |
| | 8 | RI 环形指示灯 | 输入 | 呼入 |
| | 9+10 | PE 接地 | - | GND 功能性接地（隔离） |

图 2-2 RS232 连接端子图

本例测试系统的硬件连接。如图 2-3 所示：

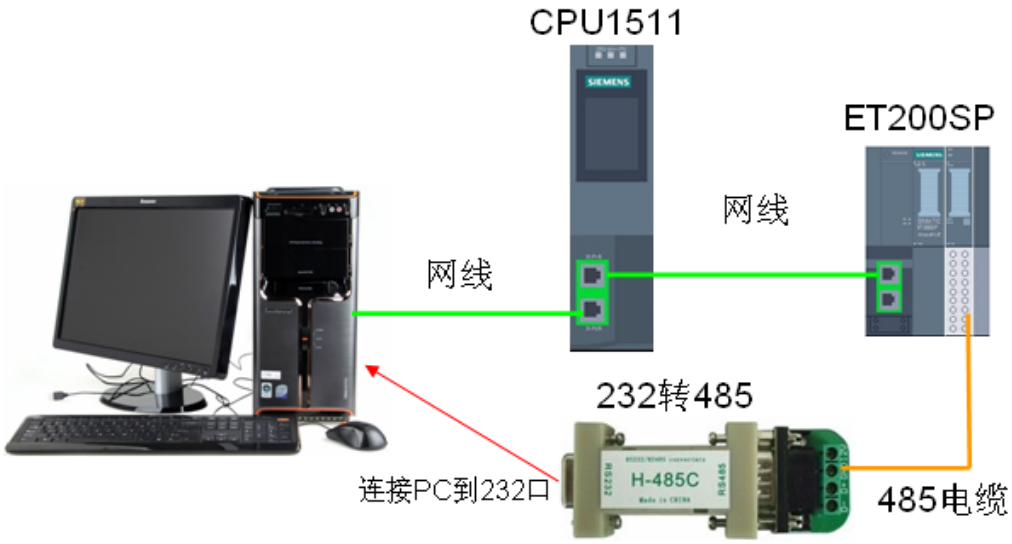



图 2-3 系统的硬件结构

## 2.3 Modbus master 协议通信

（1）硬件配置

按照图 2-3 硬件配置图进行连接，配置一套 S7-1500 PLC 连接 ET200SP 系统作为 Modbus 主站，ET200SP CM PTP 和 PC 端的 RS232/RS485 接口相连，以便使用模拟软件进行通信测试。PC 的以太网接口和 S7-1500 的 PN 接口相连。

（2）系统组态及参数设置

在 TIA PORTAL 新建一个项目，插入一个 S7-1500 站点，命名为 PLC1500_master，然后在设备视图和网络视图中插入 CPU 和 ET200SP，并配置 profinet 网络：CPU1500 PN 接口 IP：192.168.70.11；ET200SP 的接口模块的 IP：192.168.70.12。如图 2-4 所示：


图 2-4 设备和网络视图

然后在 ET200SP CM PTP 模块的端口组态中，选择 modbus 通信协议。如图 2-5 所示：

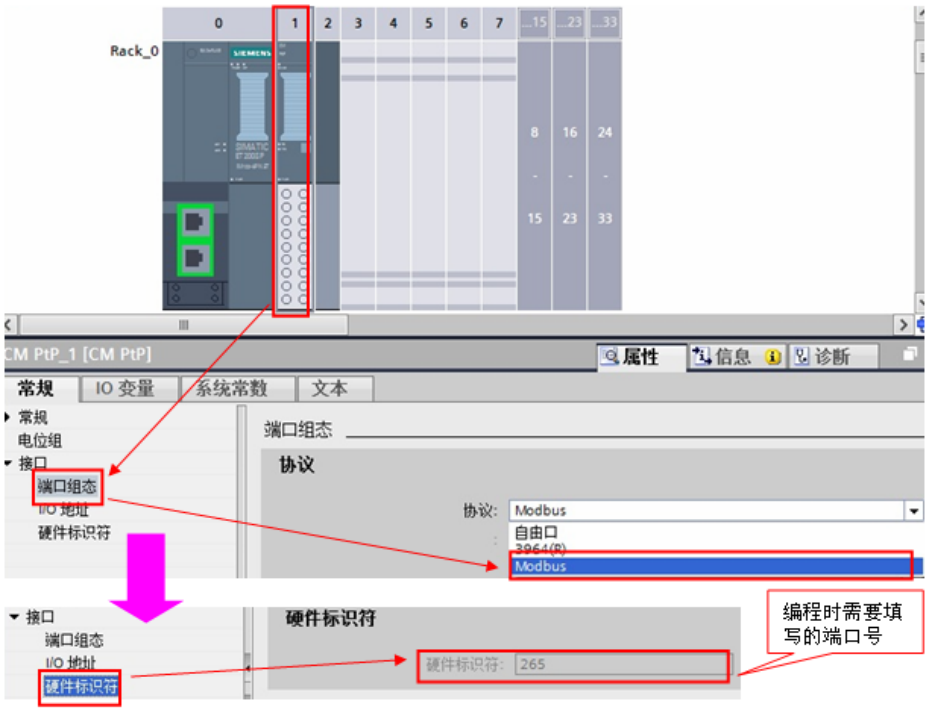



图 2-5 设置 ET200SP CM PTP 模块协议

### 2.3.1 编写通信程序

（1）OB1 编程

在项目的 OB1 组织块中添加 Modbus RTU 初始化功能块“Modbus_Comm_Load”，并为该 FB 块增加一个背景数据块，本例中为 DB1“Modbus_Comm_Load _DB”；然后在下一个网络中添加主站操作指令“Modbus_Master”，为该 FB 块增加一个背景数据块，本例中为 DB2“Modbus_ Master _DB”；如下图 2-6 所示：

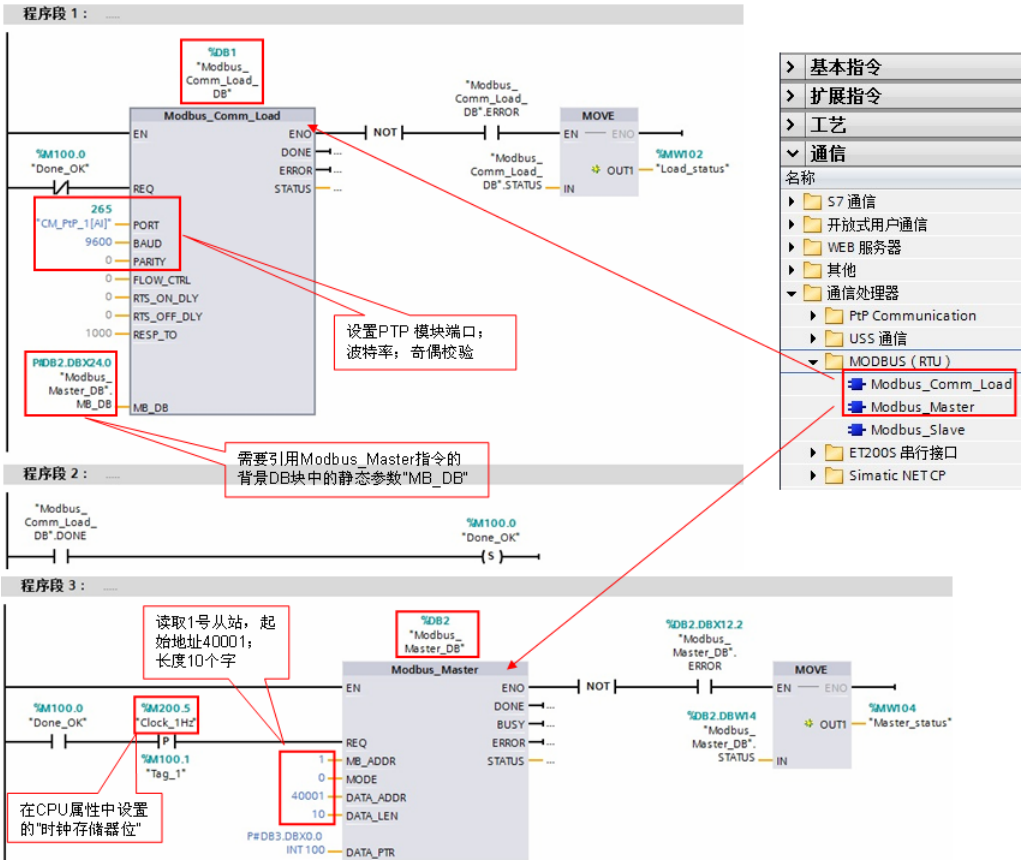



图 2-6 添加“Modbus_Comm_Load”和“Modbus_Master”功能块

功能块“Modbus_Master”的主要管脚参数如下表 2-3 所示：

| “Modbus_Master”的管脚参数 | 管脚声明 | 数据类型 | 含义 |
|---|---|---|---|
| REQ | 输入 | Bool | 0：无请求；1：请求向 Modbus 从站发送数据 |
| MB_ADDR | 输入 | UInt | Modbus RTU 站地址：<br>标准地址范围（1 到 247 以及 0，用于 Broadcast）<br>扩展地址范围（1 到 65535 以及 0 ，用于 Broadcast）<br>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 | 输入 | USInt | 模式选择： 指定请求类型（读取、 |

| | | | |
|---|---|---|---|
| | | | 写入或诊断）。 |
| DATA_ADDR | 输入 | UDInt | 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 | 输入 | UInt | 数据长度： 指定此指令将访问的位或字的个数。 |
| DATA_PTR | 输入 | 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。 |
| DONE | 输出 | Bool | 如果上一个请求完成并且没有错误，DONE 位将变为 TRUE 并保持一个周期。 |
| BUSY | 输出 | Bool | FALSE - Modbus_Master 无激活命令<br>TRUE - Modbus_Master 命令执行中 |
| ERROR | 输出 | Bool | 如果上一个请求完成出错，则 ERROR 位将变为 TRUE 并保持一个周期。STATUS 参数中的错误代码仅在 ERROR = TRUE 的周期内有效。 |
| STATUS | 输出 | Word | 通信状态信息，用于诊断。 |

表 2-3 功能块“Modbus_Master”的管脚参数

（2）modbus 地址对应关系

Modbus_Master 指令使用 MODE 输入，不使用功能代码输入。 MODE 和 DATA_ADDR 结合使用可指定在实际 Modbus 帧中使用的功能代码。 下表显示了 MODE 参数、Modbus 功能代码和 DATA_ADDR 中 Modbus 地址范围之间的关系。

| 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 |

表 2-4 modbus 地址、功能码对应关系

（3）选择接口类型和创建数据块

ET200SP CM PTP 模块支持 RS 232，RS 422 和 RS 485 接口，根据通信对象的不同，需要将模块设置为不同的工作模式，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（多点主站，CM PtP (ET 200SP)）

3 = 全双工 (RS 422) 四线制模式（多点从站，CM PtP (ET 200SP)）

4 = 半双工 (RS485) 二线制模式

本例中以 485 为例，则需要在功能块“Modbus_Comm_Load”的背景块 DB1 中找到“MODE”参数，并将其启动值改为 4。如图 2-7 所示：

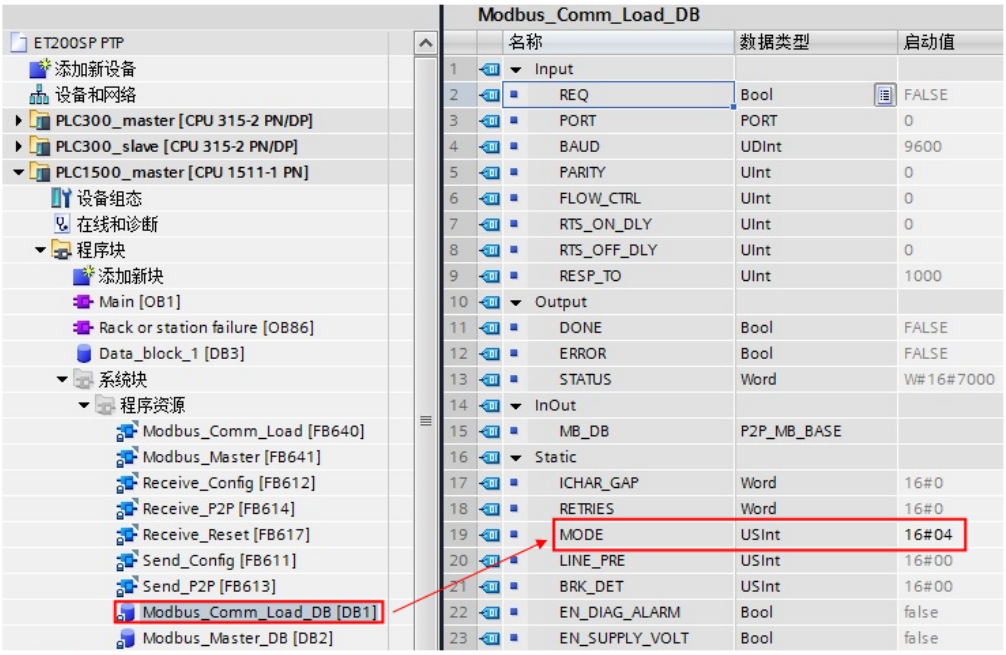



图 2-7 修改 ET200SP CM PTP 模块工作模式

然后，创建一个全局数据块用于匹配功能块“MB_Master”的管脚参数“DATA_PTR”，本例中创建 100 个字的数组数据块 DB3 “Data_block_1”，用于存储保持寄存器的通信数据，本例中读取的 modbus 地址 40001~40010 中的数据将存放到 DB3 的前 10 个字中。并且注意，需要在 DB 块属性中将其设置为标准 DB 块。如下图 2-8 所示：

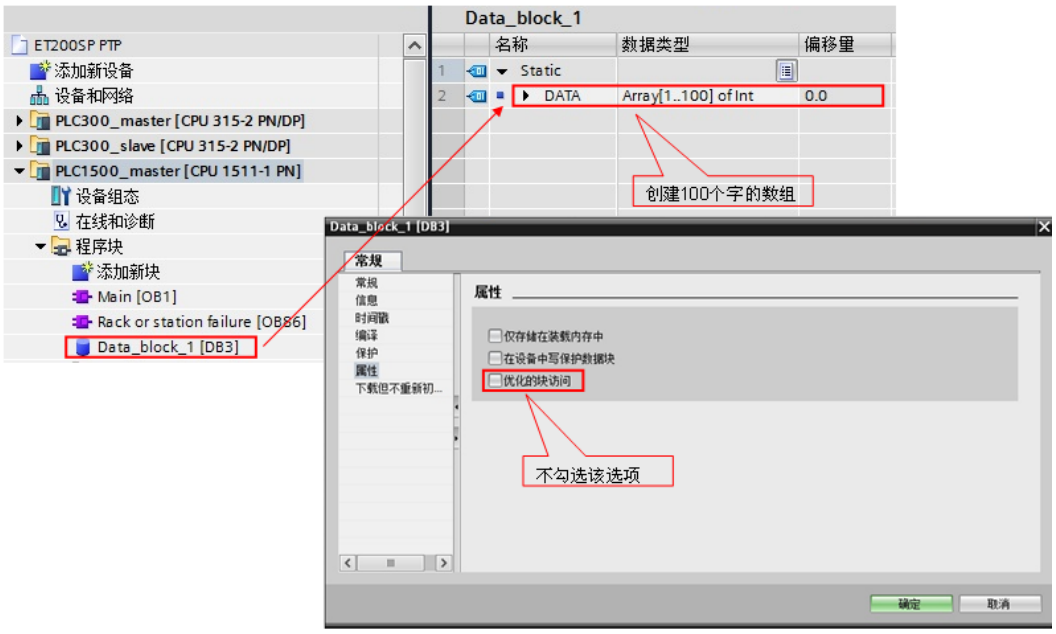



图 2-8 创建 DB 块并将其设置为标准块

（4）调用 OB86 防止掉站停机

本例中使用的 ET200SP 为 PROFINET IO 设备，为避免由于 IO 设备的掉站导致 CPU 停止工作，则需要添加组织块 OB86；并且需要在分布式 IO 设备恢复连接时，重新初始化 ET200SP CM PTP 模块，利用 OB86 的临时变量“Event_Class”的状态值执行一次“Modbus_Comm_Load”指令。如下图 2-9 所示：

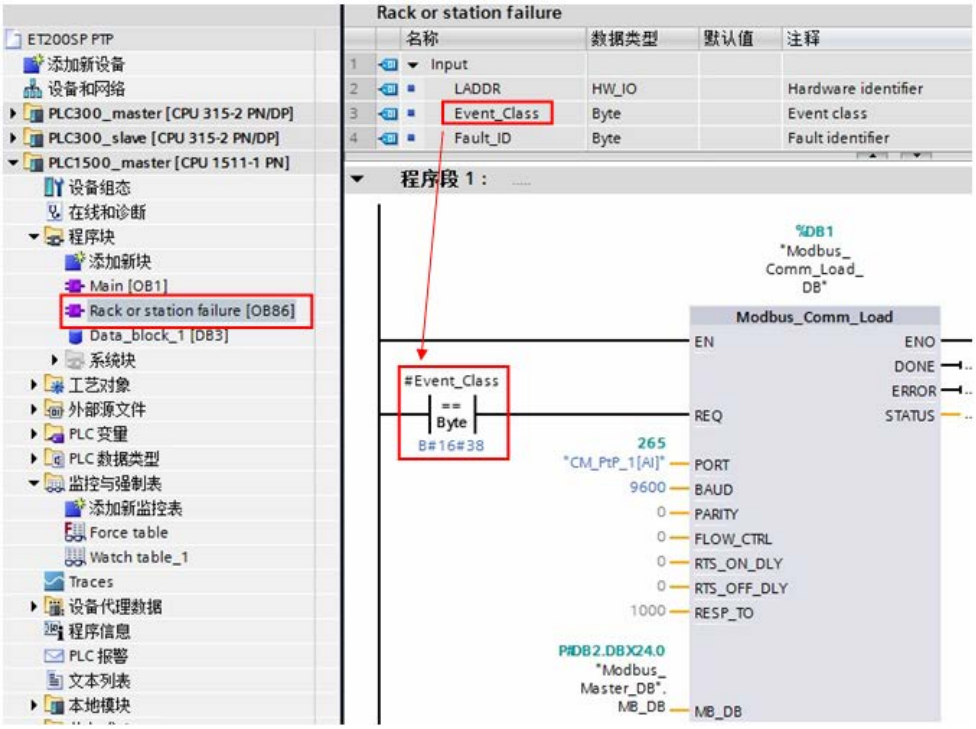



图 2-9 OB86 编程

说明：当有分布式 IO 掉站时，OB86 临时变量“Event_Class”=B#16#39；当有分布式 IO 恢复连接时，OB86 临时变量“Event_Class”=B#16#38；更多信息请查看 OB86 组织块的帮助说明。

### 2.3.2 下载程序

分配设备名称（注：如果使用的分布式 IO 是 Profibus DP，则跳过该步骤）：将软件切换到“网络视图”，找到 PN/IE 总线，查看设备名称是否正确。如图 2-10、2-11 所示：


图 2-10 网络视图

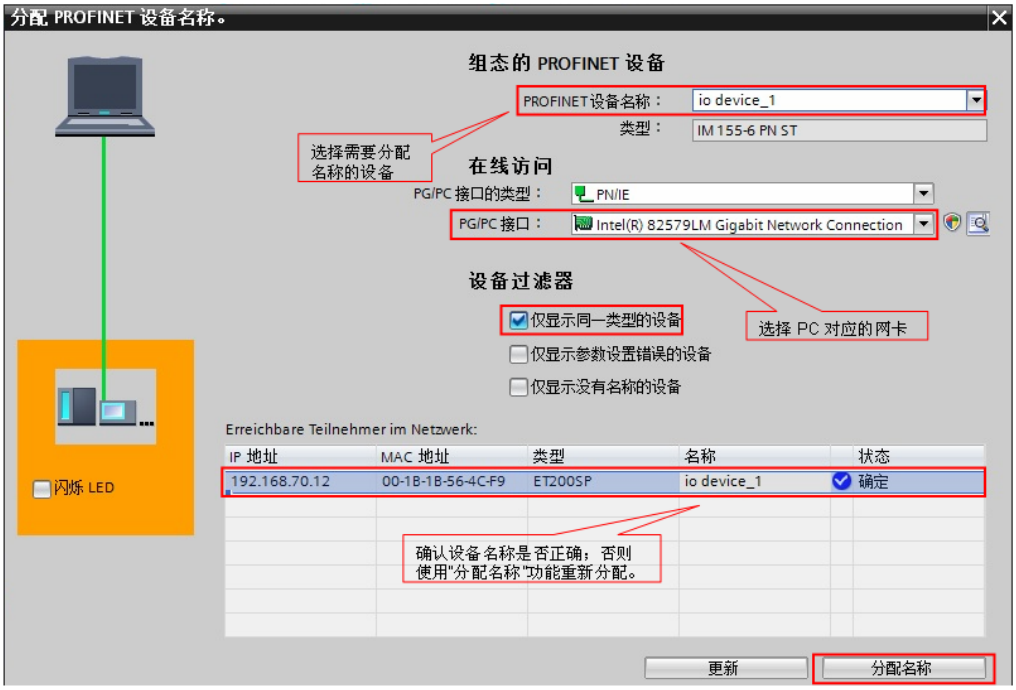



图 2-11 确认设备名称和 IP 地址

编译并下载程序到 PLC 中。

### 2.3.3 通信测试

由于 Modbus Master 指令支持功能码 FC1，2，3，4，5，6，15，16，不同的功能码测试过程类似，因此本例中的测试以 FC3（读保持寄存器）为例来说明通讯测试的过程，本例中读取的 modbus 地址 40001~40010 中的数据将存放到 DB3 的前 10 个字中。对于其他功能码的测试将不再重复描述。

打开 ModSim32 软件，在“Connection——>Connect”中打开连接属性对话框，连接接口选择“Port1”，设置相应的波特率和奇偶校验等参数。如图 2-12 所示：

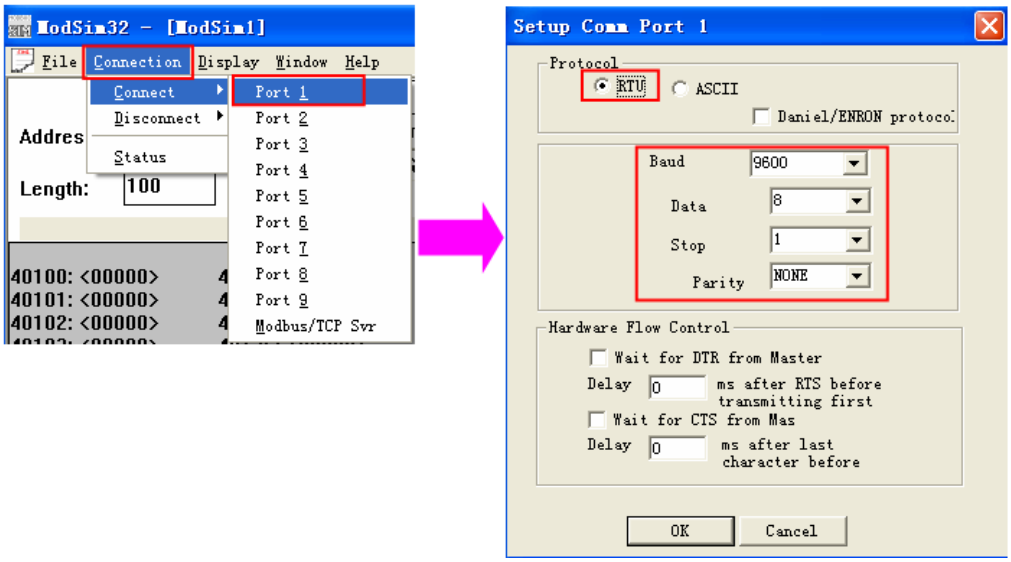



图 2-12 设置测试软件

然后，在 TIA Portal 中新建监控表，添加通信数据区，在线监控。如图 2-13 所示：

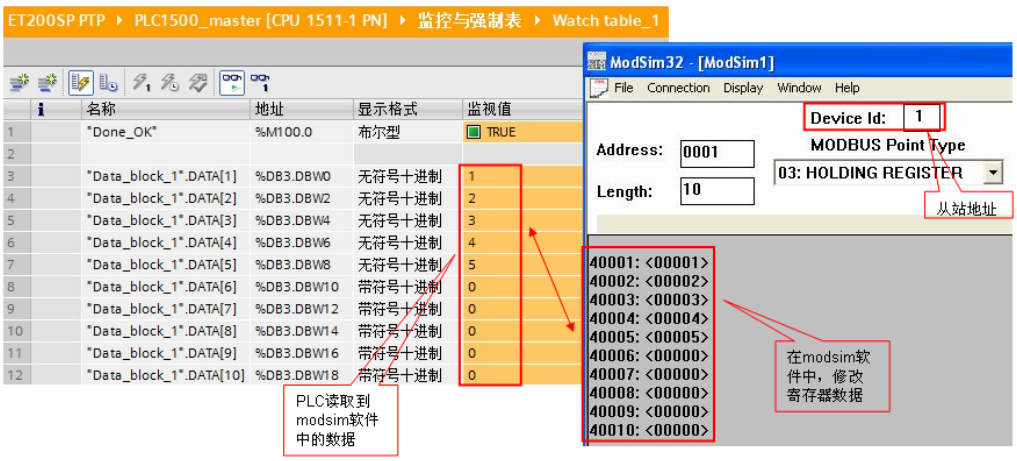



图 2-13 通讯测试

## 2.4 Modbus slave 协议通信

（1）硬件配置

按照图 2-3 硬件配置图进行连接，配置一套 S7-1500 PLC 连接 ET200SP 系统作为 modbus 从站，ET200SP CM PTP 和 PC 端的 RS232/RS485 接口相连，以便使用模拟软件进行通信测试。PC 的以太网接口和 S7-1500 的 PN 接口相连。

（2）系统组态及参数设置

在 TIA PORTAL 新建一个项目，插入一个 S7-1500 站点，命名为 PLC1500_slave，然后在设备视图和网络视图中插入 CPU 和 ET200SP，并配置 profinet 网络：CPU1500 PN 接口 IP：192.168.70.11；ET200SP 的接口模块的 IP：192.168.70.12。如图 2-14 所示：


图 2-14 设备和网络视图

然后在 ET200SP CM PTP 模块的端口组态中，选择 modbus 通信协议。如图 2-15 所示：


图 2-15 设置 ET200SP CM PTP 模块协议

### 2.4.1 编写通信程序

（1）OB1 编程

在项目的 OB1 组织块中添加 Modbus RTU 初始化功能块“Modbus_Comm_Load”，并为该 FB 块增加一个背景数据块，本例中为 DB1“Modbus_Comm_Load _DB”；然后在下一个网络中添加主站操作指令“Modbus_Slave”， 为该 FB 块增加一个背景数据块，本例中为 DB2“Modbus_ Slave _DB”；如下图 2-16 所示：

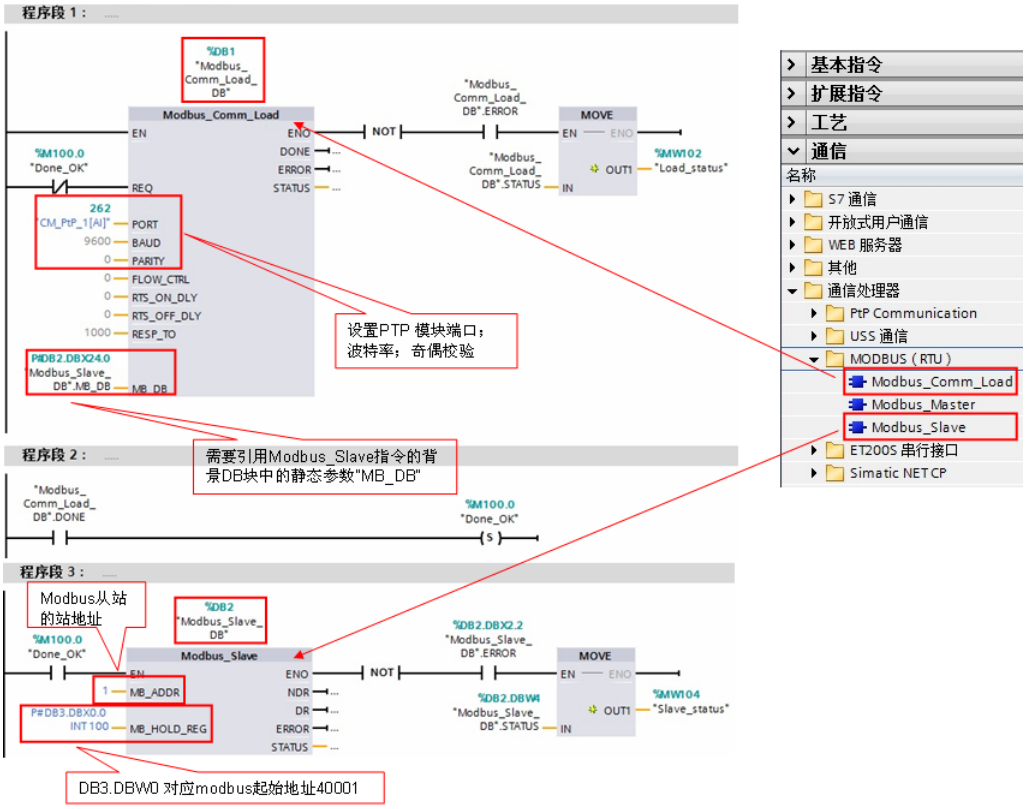



图 2-16 添加“Modbus_Comm_Load”和“Modbus_Slave”功能块

功能块“Modbus_Slave”的主要管脚参数如下表 2-5 所示：

| “Modbus_Slave”的管脚参数 | 管脚声明 | 数据类型 | 含义 |
|---|---|---|---|
| MB_ADDR | 输入 | UInt | Modbus 从站的标准寻址：<br>标准寻址范围（1 到 247）<br>扩展寻址范围（0 到 65535）<br>注意： 0 是广播地址 |
| MB_HOLD_REG | 输入 | USInt | Modbus 保持寄存器 DB 的指针： Modbus 保持寄存器可能为标志或数据块的存储区。 |
| NDR | 输出 | Bool | 可用的新数据：<br>FALSE - 无新数据<br>TRUE - 表示新数据已由 Modbus 主站写入<br>如果上一个请求完成并且没有错误，NDR 位将变为 TRUE 并保持一个周期。 |
| DR | 输出 | Bool | 读取数据： |

| | | | |
|---|---|---|---|
| | | | FALSE – 未读取数据<br>TRUE – 表示该指令已将 Modbus 主站接收到的数据存储在目标区域中。<br>如果上一个请求完成并且没有错误，DR 位将变为 TRUE 并保持一个周期。 |
| ERROR | 输出 | Bool | 如果上一个请求完成出错，则 ERROR 位将变为 TRUE 并保持一个周期。 STATUS 参数中的错误代码仅在 ERROR = TRUE 的周期内有效。 |
| STATUS | 输出 | Word | 通信状态信息，用于诊断。 |

表 2-5 功能块“Modbus_Slave”的管脚参数

（2）选择接口类型和创建数据块

ET200SP CM PTP 模块支持 RS 232，RS 422 和 RS 485 接口，根据通信对象的不同，需要将模块设置为不同的工作模式，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（多点主站，CM PtP (ET 200SP)）

3 = 全双工 (RS 422) 四线制模式（多点从站，CM PtP (ET 200SP)）

4 = 半双工 (RS485) 二线制模式

本例中以 485 为例，则需要在功能块“Modbus_Comm_Load”的背景块 DB1 中找到“MODE”参数，并将其启动值改为 4。如图 2-17 所示：

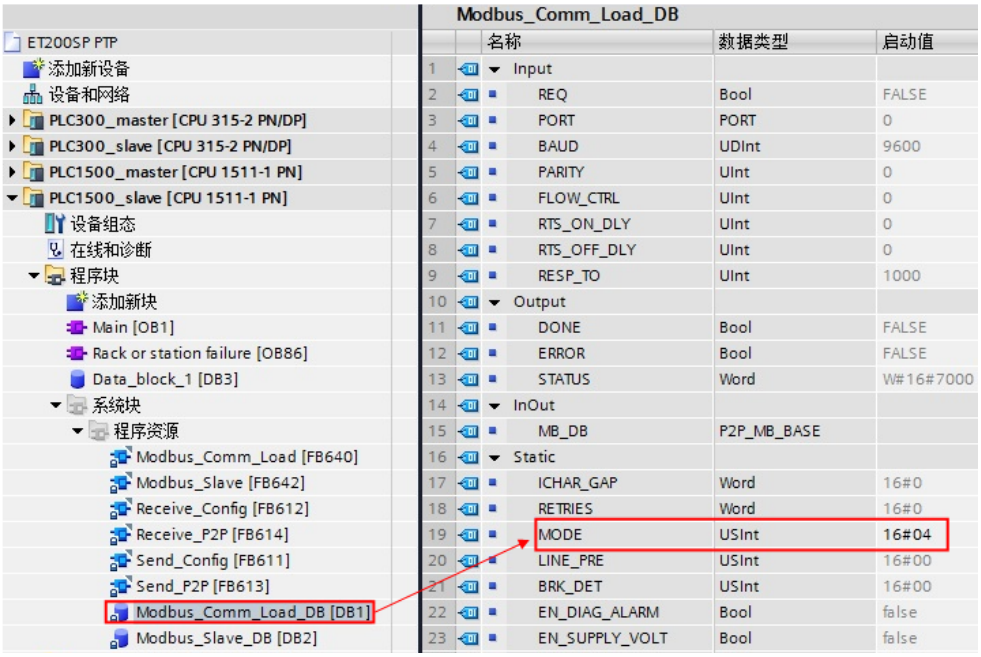



图 2-17 修改 ET200SP CM PTP 模块工作模式

然后，创建一个全局数据块用于匹配功能块“MB_Slave”的管脚参数“MB_HOLD_REG”，本例中创建数据块 DB3 “Data_block_1”，用于对应 modbus 保持寄存器，本例中 DB3 定义了 100 个字的数组对应于 modbus 地址 40001~40100。并且注意，需要在 DB 块属性中将其设置为标准 DB 块。如下图 2-18 所示：

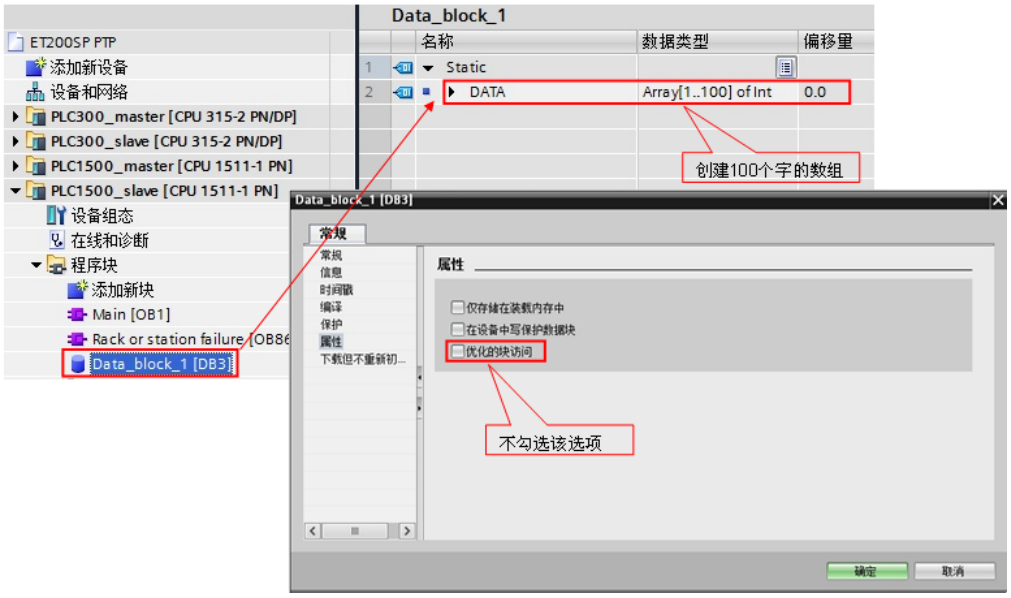



图 2-18 创建 DB 块并将其设置为标准块

（3）modbus 地址对应关系

上面提到保持寄存器是由功能块“Modbus_Slave”的管脚参数“MB_HOLD_REG”定义的 DB 块关联，其对应如下表 2-6 所示：

| MB_SLAVE Modbus功能 | | | | S7-1500/300/400 | |
|---|---|---|---|---|---|
| 代码 | 功能 | 数据区 | 地址范围 | CPU DB数据区 | CPU地址 |
| 3 | 读字 | 保持寄存器 | 40001到49999 | MB_HOLD_REG | 字1到字9999 |
| | | | 400001到465535 | | 字1到字65535 |
| 6 | 写单个字 | 保持寄存器 | 40001到49999 | MB_HOLD_REG | 字1到字9999 |
| | | | 400001到465535 | | 字1到字65535 |
| 16 | 写字 | 保持寄存器 | 40001到49999 | MB_HOLD_REG | 字1到字9999 |
| | | | 400001到465535 | | 字1到字65535 |

表 2-6 Modbus 的寄存器地址映射表

对于其它数据类型，如线圈、离散输入、模拟量输入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表 2-7 所示：

| Modbus 功能 | | | | | | S7-1500 / S7-300 / S7-400 | | | |
|---|---|---|---|---|---|---|---|---|---|
| 代码 | 功能 | 数据区 | 地址区 | | | 数据区 | CPU 地址 | | |
| 01 | 读取位 | 输出 | 0 | 到 | 9998 | 输出的过程映像 | O0.0 | 到 | O1248.6 |
| 02 | 读取位 | 输入 | 0 | 到 | 9998 | 输入的过程映像 | I0.0 | 到 | I1248.6 |
| 04 | 读取字 | 输入 | 0 | 到 | 9998 | 输入的过程映像 | IW0 | 到 | IW19996 |
| 05 | 写入位 | 输出 | 0 | 到 | 9998 | 输出的过程映像 | O0.0 | 到 | O1248.6 |
| 15 | 写入位 | 输出 | 0 | 到 | 9998 | 输出的过程映像 | O0.0 | 到 | O1248.6 |

表 2-7 Modbus 地址映射表

（4）调用 OB86 防止掉站停机

本例中使用的 ET200SP 为 PROFINET IO 设备，为避免由于 IO 设备的掉站导致 CPU 停止工作，则需要添加组织块 OB86；并且需要在分布式 IO 设备恢复连接时，重新初始化 ET200SP CM PTP 模块，利用 OB86 的临时变量“Event_Class”的状态值执行一次“Modbus_Comm_Load”指令。如下图 2-19 所示：

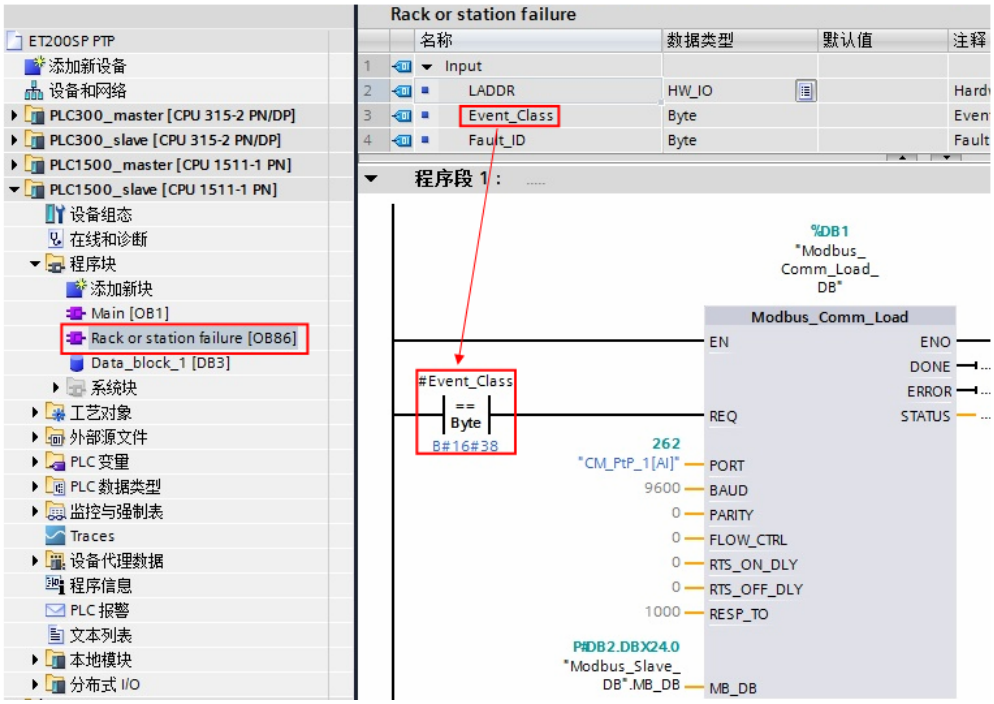



图 2-19 OB86 编程

说明：当有分布式 IO 掉站时，OB86 临时变量“Event_Class”=B#16#39；当有分布式 IO 恢复连接时，OB86 临时变量“Event_Class”=B#16#38；更多信息请查看 OB86 组织块的帮助说明。

### 2.4.2 下载程序

分配设备名称（注：如果使用的分布式 IO 是 Profibus DP，则跳过以下步骤）：将软件切换到“网络视图”，找到 PN/IE 总线，查看设备名称是否正确。如图 2-20、2-21 所示：

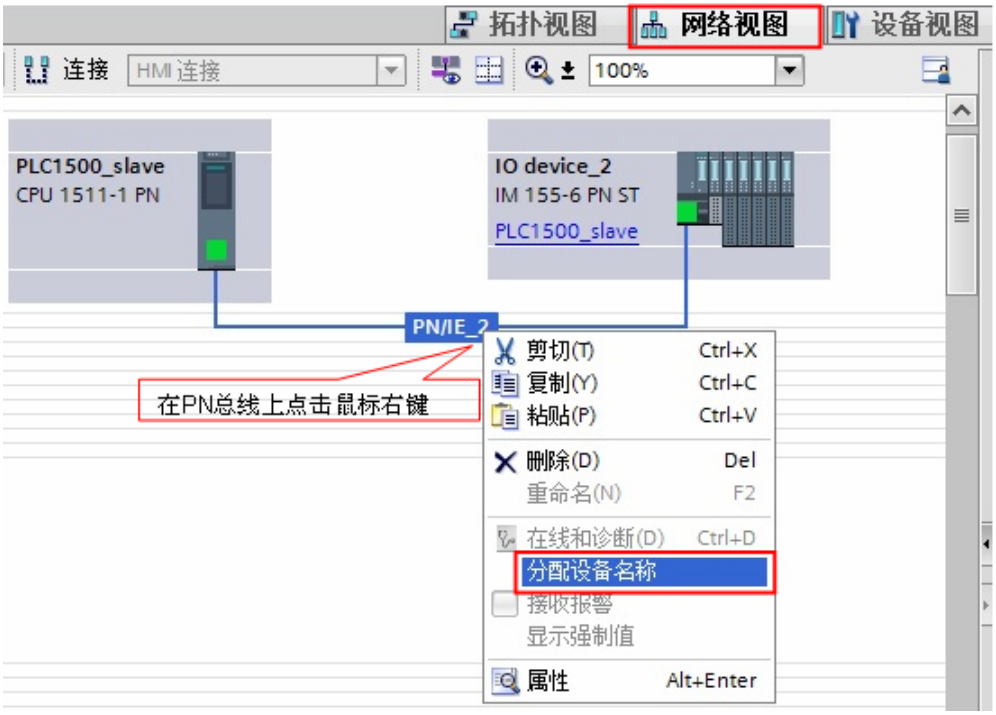



图 2-20 网络视图

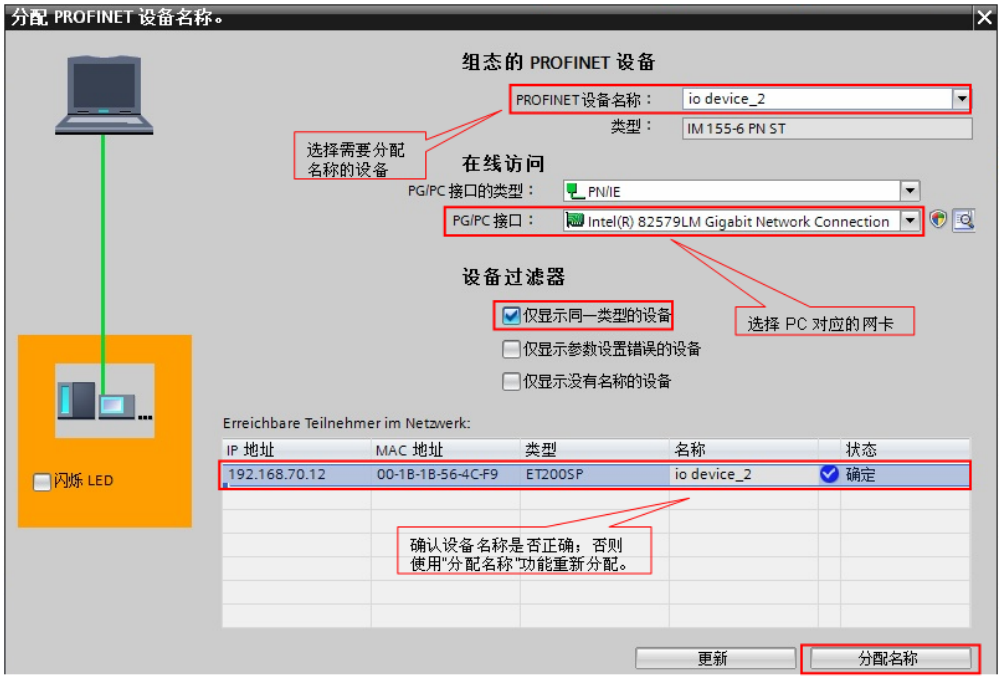



图 2-21 确认设备名称和 IP 地址

编译并下载程序到 PLC 中。

### 2.4.3 通信测试

由于 Modbus Slave 指令支持功能码 FC1，2，3，4，5，6，15，16，不同的功能码测试过程类似，因此本例中的测试以 FC3（读保持寄存器）为例来说明通讯测试的过程，本例使用的 modscan32 软件将读取来自 PLC 的 DB3 前 10 个字，对应的 modbus 地址为 40001~40010。对于其他功能码的测试将不再重复描述。

打开 ModScan32 软件，在“Connection——>Connect”中打开连接属性对话框，连接接口选择“Port1”，设置相应的波特率和奇偶校验等参数。如图 2-22 所示：

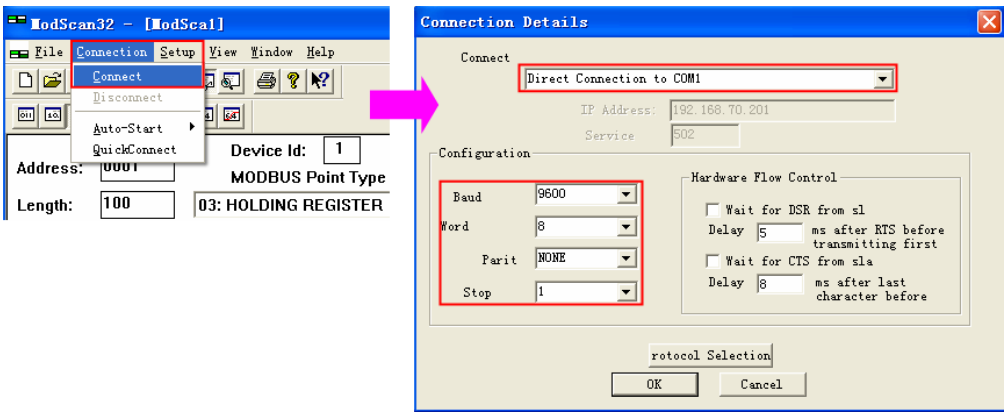



图 2-22 设置测试软件

然后，在 TIA Portal 中新建监控表，添加通信数据区，在线修改数据、监控。如图 2-23 所示：

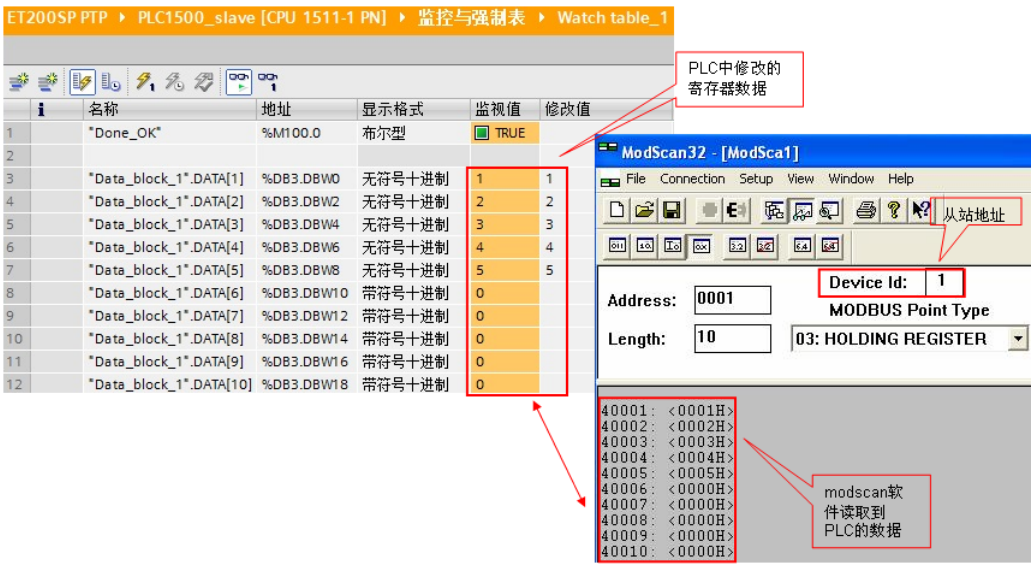



图 2-23 通讯测试

# 3 S7-300 的分布式IO中使用ET200SP的ptp模块

## 3.1 硬件和软件需求

| 名称 | 数量 | 订货号 |
|---|---|---|
| 电源模块 PS307 | 1 | 6ES7 307-1EA00-0AA0 |
| CPU 315-2PN/DP | 1 | 6ES7 315-2EH14-0AB0 |
| ET200SP IM155-6 PN ST | 1 | 6ES7 155-6AU00-0BN0 |
| ET200SP CM PTP | 1 | 6ES7 137-6AA00-0BA0 |
| PC，带232串口 | 1 | |
| RS232转RS485转换器 | 1 | |
| 网线 | 若干 | |

表 3-1 硬件订货信息

| 名称 | 订货号 |
|---|---|
| TIA PORTAL V13 professional | 6ES7 822-1AA03-0YA5 |
| Modscan32 用于在 PC 中模拟主站 | |
| Modsim32 用于在 PC 中模拟从站 | |

表 3-2 软件订货信息

## 3.2 硬件接线

对于 ET200SP/ET200MP 的 PTP 模块接线，请参考模块手册。

SIMATIC ET200SP CM PtP RS232/422/485 手册：

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

SIMATIC S7-1500 CM PtP RS422/485 HF 手册：

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

SIMATIC S7-1500 CM PtP RS232 HF 手册：

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

本例中使用的 ET200SP CM PTP 模块，测试时使用 RS485 接口，根据手册中的信息，端子 14 为信号正极，端子 12 为信号负极；接线方式，如图 3-1 和 3-2 所示：

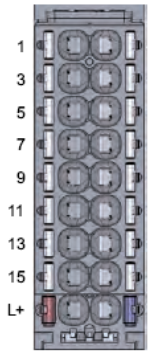
| 通信模块 BaseUnit 的端子分配 | 针脚 | 标识 | 输入/输出 | 含义 |
|---|---|---|---|---|
|  | 11 | T (A) - | 输出 | 发送数据（四线制模式） |
| | 12 | R (A) -<br>T(A)/R(A) | 输入<br>输入/输出 | 接收数据（四线制模式）<br>接收/发送数据（两线制模式） |
| | 13 | T (B) + | 输出 | 发送数据（四线制模式） |
| | 14 | R (B) +<br>T(B)/R(B) | 输入<br>输入/输出 | 接收数据（四线制模式）<br>接收/发送数据（两线制模式） |
| | 15+16 | PE 接地 | - | GND 功能性接地（隔离） |

图 3-1 RS422/485 连接端子图

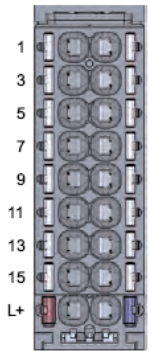
| 通信模块 BaseUnit 的端子分配 | 针脚 | 标识 | 输入/输出 | 含义 |
|---|---|---|---|---|
|  | 1 | TXD 传输数据 | 输出 | 传输数据 |
| | 2 | RXD 接收数据 | 输入 | 接收数据 |
| | 3 | RTS 请求发送 | 输出 | 请求发送 |
| | 4 | CTS 清除以发送 | 输入 | 清除以发送 |
| | 5 | DTR 数据终端准备就绪 | 输出 | 数据终端准备就绪 |
| | 6 | DSR 数据集准备就绪 | 输入 | 数据集准备就绪 |
| | 7 | DCD 数据载体检测 | 输入 | 接收的信号电平 |
| | 8 | RI 环形指示灯 | 输入 | 呼入 |
| | 9+10 | PE 接地 | - | GND 功能性接地（隔离） |

图 3-2 RS232 连接端子图

本例测试系统的硬件连接。如图 3-3 所示：

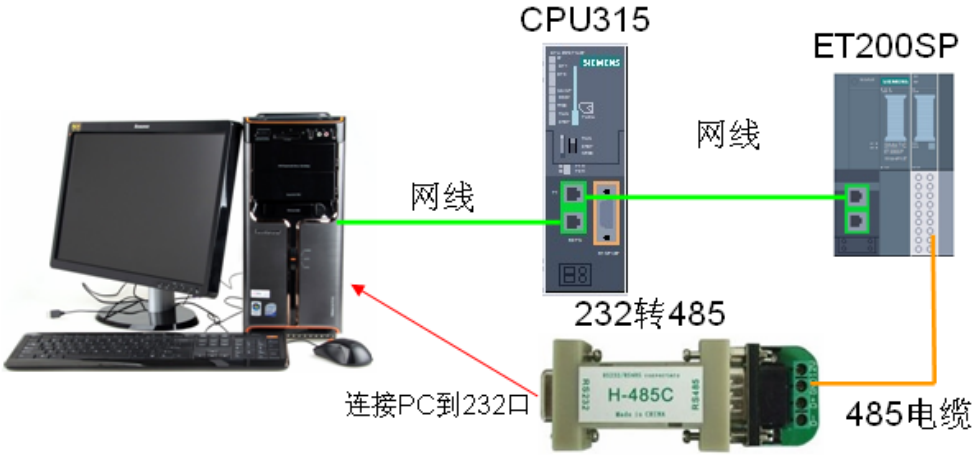



图 3-3 系统的硬件结构

## 3.3 Modbus master 协议通信

（1）硬件配置

按照图 3-3 硬件配置图进行连接，配置一套 S7-300 PLC 连接 ET200SP 系统作为 Modbus 主站，ET200SP CM PTP 和 PC 端的 RS232/RS485 接口相连，以便使用模拟软件进行通信测试。PC 的以太网接口和 S7-300 的 PN 接口相连。

（2）系统组态及参数设置

在 TIA PORTAL 新建一个项目，插入一个 S7-300 站点，命名为 PLC300_master，然后在设备视图和网络视图中插入 CPU 和 ET200SP，并配置 profinet 网络：CPU300 PN 接口 IP：192.168.70.201；ET200SP 的接口模块的 IP：192.168.70.202。如图 3-4 所示：


图 3-4 设备和网络视图

然后在 ET200SP CM PTP 模块的端口组态中，选择 modbus 通信协议。如图 3-5 所示：

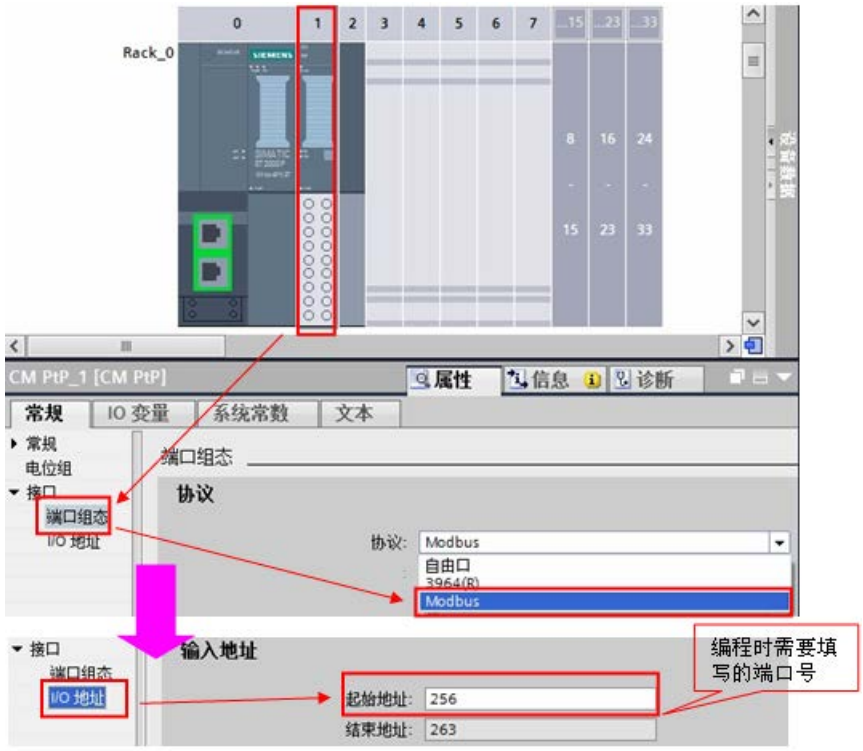



图 3-5 设置 ET200SP CM PTP 模块协议

### 3.3.1 编写通信程序

（1）OB1 编程

在项目的 OB1 组织块中添加 Modbus RTU 初始化功能块“Modbus_Comm_Load”，并为该 FB 块增加一个背景数据块，本例中为 DB1“Modbus_Comm_Load _DB”；然后在下一个网络中添加主站操作指令“Modbus_Master”， 为该 FB 块增加一个背景数据块，本例中为 DB2“Modbus_ Master _DB”；如下图 3-6 所示：

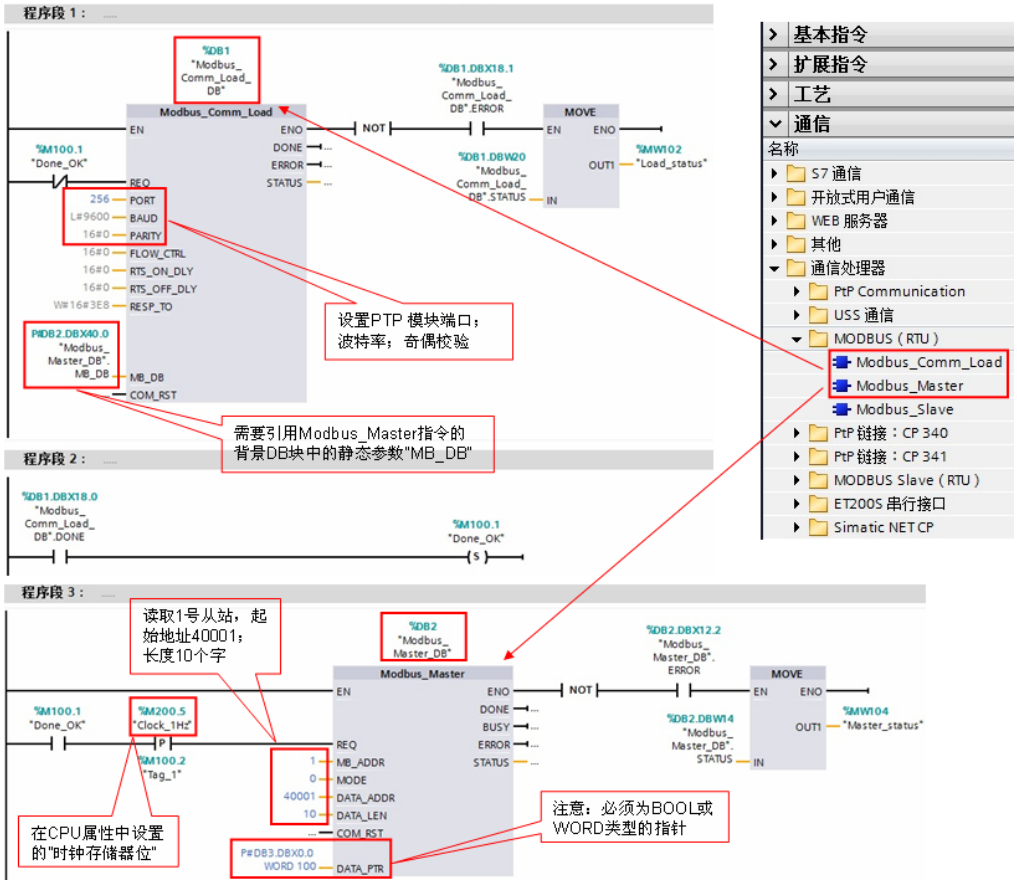



图 3-6 添加“Modbus_Comm_Load”和“Modbus_Master”功能块

功能块“Modbus_Master”的主要管脚参数如下表 3-3 所示：

| “Modbus_Master”的管脚参数 | 管脚声明 | 数据类型 | 含义 |
|---|---|---|---|
| REQ | 输入 | Bool | 0：无请求；1：请求向 Modbus 从站发送数据 |
| MB_ADDR | 输入 | UInt | Modbus RTU 站地址：<br>标准地址范围（1 到 247 以及 0，用于 Broadcast）<br>扩展地址范围（1 到 65535 以及 0，用于 Broadcast）<br>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 | 输入 | USInt | 模式选择： 指定请求类型（读取、写入或诊断）。 |
| DATA_ADDR | 输入 | UDInt | 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 | 输入 | UInt | 数据长度： 指定此指令将访问的位或字的个数。 |
| COM_RST | 输入/输出 | Bool | Modbus_Master 指令的初始化：指令在 TRUE 时执行。 随后会将 COM_RST 复位为 FALSE。 |
| DATA_PTR | 输入 | Any | 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。 |
| DONE | 输出 | Bool | 如果上一个请求完成并且没有错误，DONE 位将变为 TRUE 并保持一个周期。 |
| BUSY | 输出 | Bool | FALSE - Modbus_Master 无激活命令<br>TRUE - Modbus_Master 命令执行中 |
| ERROR | 输出 | Bool | 如果上一个请求完成出错，则 ERROR 位将变为 TRUE 并保持一个周期。STATUS 参数中的错误代码仅在 ERROR = TRUE 的周期内有效。 |
| STATUS | 输出 | Word | 通信状态信息，用于诊断。 |

表 3-3 功能块“Modbus_Master”的管脚参数

（2）modbus 地址对应关系

Modbus_Master 指令使用 MODE 输入，不使用功能代码输入。 MODE 和 DATA_ADDR 结合使用可指定在实际 Modbus 帧中使用的功能代码。 下表显示了 MODE 参数、Modbus 功能代码和 DATA_ADDR 中 Modbus 地址范围之间的关系。

| 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 |

表 3-4 modbus 地址、功能码对应关系

（3）选择接口类型和创建数据块

ET200SP CM PTP 模块支持 RS 232，RS 422 和 RS 485 接口，根据通信对象的不同，需要将模块设置为不同的工作模式，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（多点主站，CM PtP (ET 200SP)）

3 = 全双工 (RS 422) 四线制模式（多点从站，CM PtP (ET 200SP)）

4 = 半双工 (RS485) 二线制模式

本例中以 485 为例，则需要在功能块“Modbus_Comm_Load”的背景块 DB1 中找到“MODE”参数，并将其启动值改为 4。如图 3-7 所示：

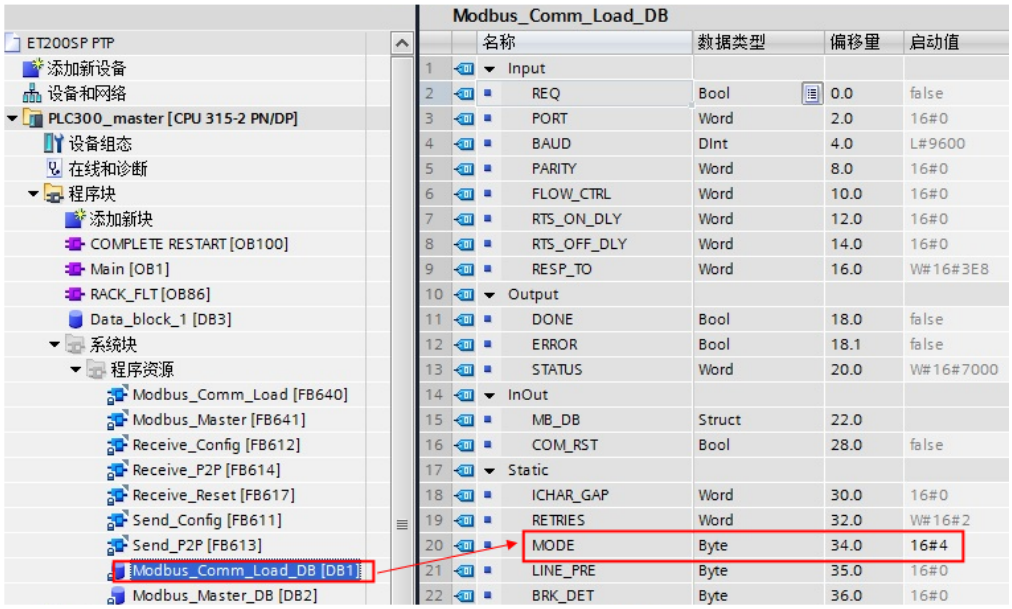



图 3-7 修改 ET200SP CM PTP 模块工作模式

然后，创建一个全局数据块用于匹配功能块“MB_Master”的管脚参数“DATA_PTR”，本例中创建 100 个字的数组数据块 DB3 “Data_block_1”，用于存储保持寄存器的通信数据，本例中读取的 modbus 地址 40001~40010 中的数据将存放到 DB3 的前 10 个字中。如下图 3-8 所示：

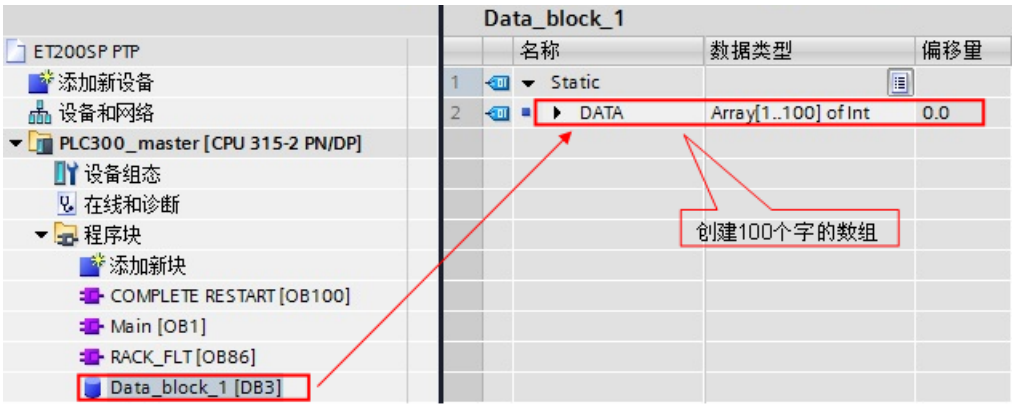



图 3-8 创建 DB 块

（4）调用 OB100 进行初始化

在 OB100 中，分别对初始化指令“Modbus_Comm_Load”和主站指令“Modbus_Master”的引脚“COM_RST”进行置位操作。如下图 3-9 所示：

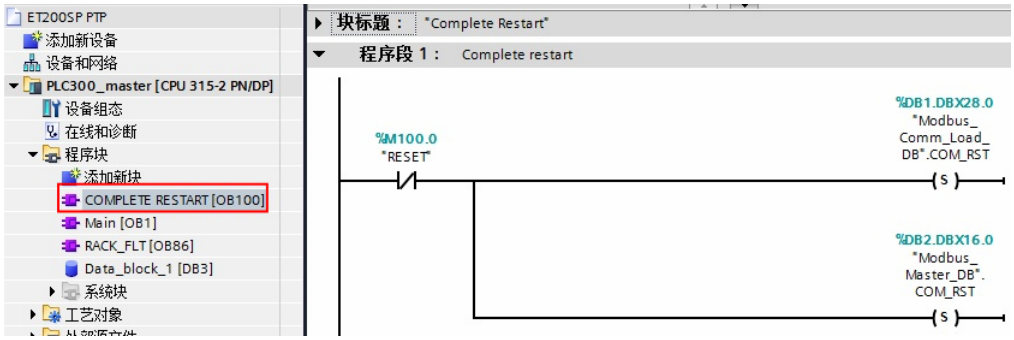



图 3-9 OB100 编程

（5）调用 OB86 防止掉站停机

本例中使用的 ET200SP 为 PROFINET IO 设备，为避免由于 IO 设备的掉站导致 CPU 停止，则需要添加组织块 OB86；并且需要在分布式 IO 设备恢复连接时，重新初始化 ET200SP CM PTP 模块，利用 OB86 的临时变量“Event_Class”的状态值对“Modbus_Comm_Load”和“Modbus_Master”指令的“COM_RST”引脚进行置位；同时，需要复位“Modbus_Comm_Load”的“REQ”引脚到达重新进行初始化的目的。如下图 3-10 所示：

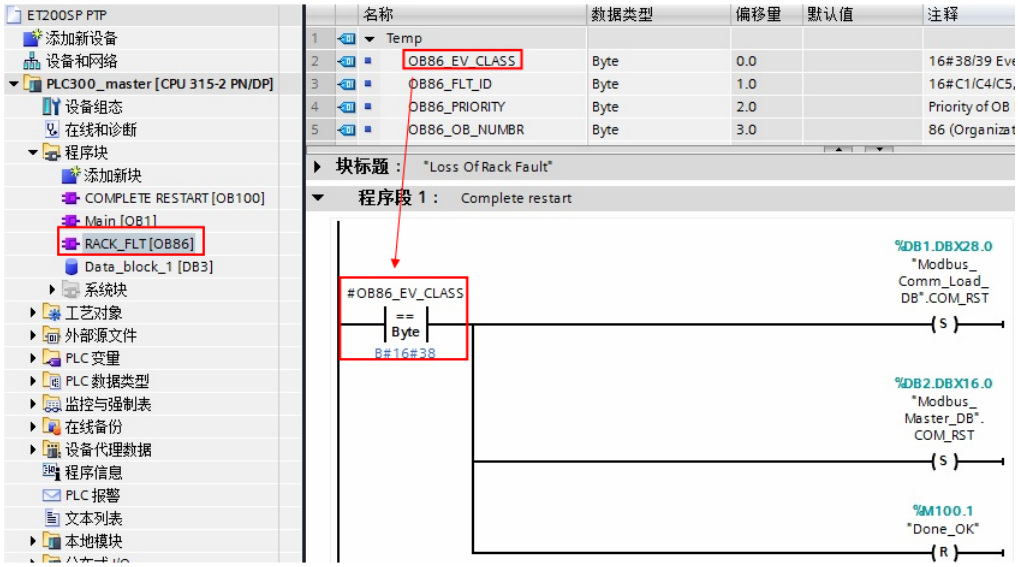



图 3-10 OB86 编程

说明：当有分布式 IO 掉站时，OB86 临时变量“Event_Class”=B#16#39；当有分布式 IO 恢复连接时，OB86 临时变量“Event_Class”=B#16#38；更多信息请查看 OB86 组织块的帮助说明。

### 3.3.2 下载程序

分配设备名称（注：如果使用的分布式 IO 是 Profibus DP，则跳过该步骤）：将软件切换到“网络视图”，找到 PN/IE 总线，查看设备名称是否正确。如图 3-11、3-12 所示：


图 3-11 网络视图

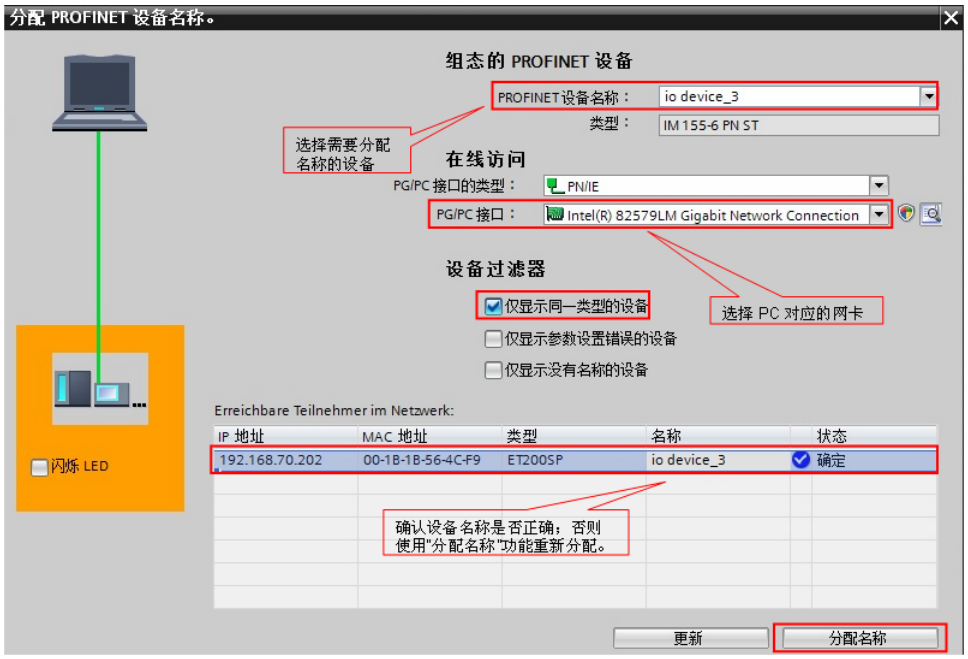



图 3-12 确认设备名称和 IP 地址

编译并下载程序到 PLC 中。

### 3.3.3 通信测试

由于 Modbus Master 指令支持功能码 FC1，2，3，4，5，6，15，16，不同的功能码测试过程类似，因此本例中的测试以 FC3（读保持寄存器）为例来说明通讯测试的过程，本例中读取的 modbus 地址 40001~40010 中的数据将存放到 DB3 的前 10 个字中。对于其他功能码的测试将不再重复描述。

打开 ModSim32 软件，在“Connection——>Connect”中打开连接属性对话框，连接接口选择“Port1”，设置相应的波特率和奇偶校验等参数。如图 3-13 所示：

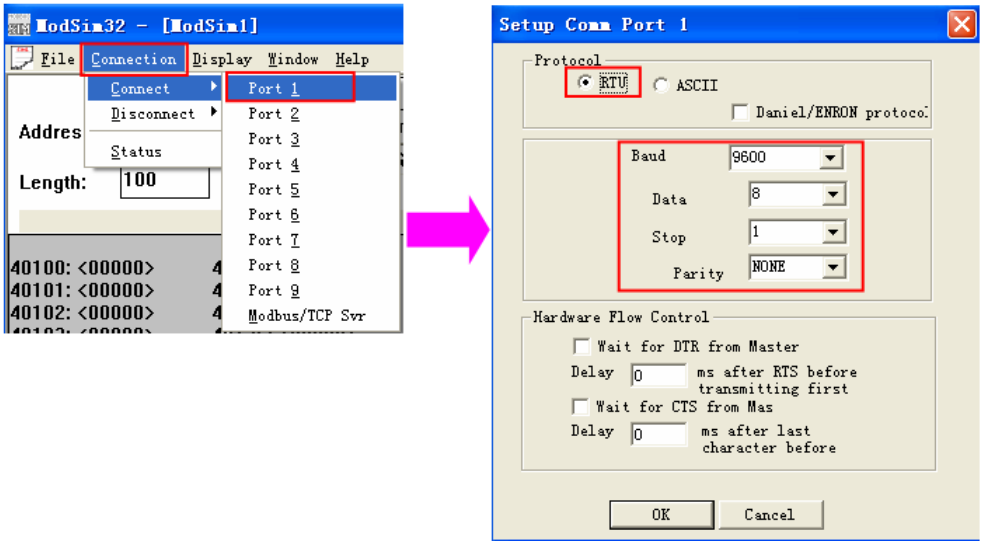



图 3-13 设置测试软件

然后，在 TIA Portal 中新建监控表，添加通信数据区，在线监控。如图 3-14 所示：

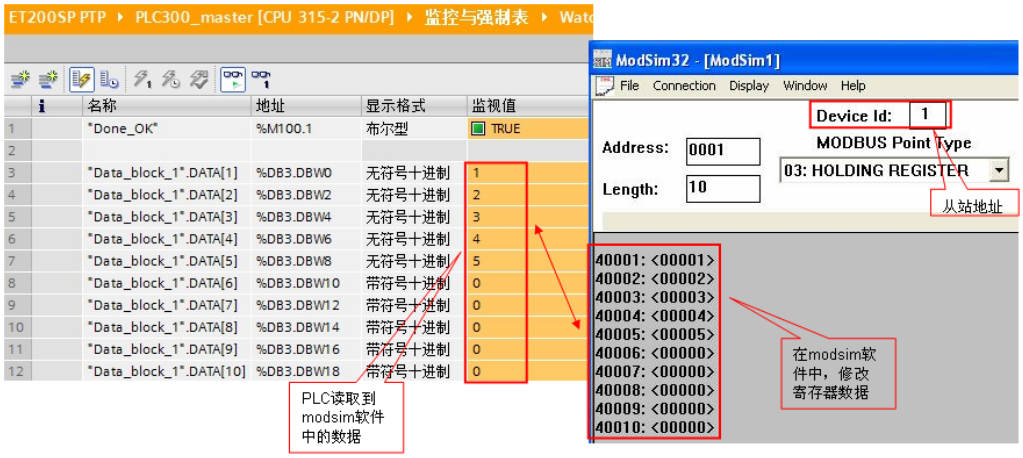



图 3-14 通讯测试

## 3.4 Modbus slave 协议通信

（1）硬件配置

按照图 3-3 硬件配置图进行连接，配置一套 S7-300 PLC 连接 ET200SP 系统作为 modbus 从站，ET200SP CM PTP 和 PC 端的 RS232/RS485 接口相连，以便使用模拟软件进行通信测试。PC 的以太网接口和 S7-300 的 PN 接口相连。

（2）系统组态及参数设置

在 TIA PORTAL 新建一个项目，插入一个 S7-300 站点，命名为 PLC300_slave，然后在设备视图和网络视图中插入 CPU 和 ET200SP，并配置 profinet 网络：CPU300 PN 接口 IP：192.168.70.201；ET200SP 的接口模块的 IP：192.168.70.202。如图 3-15 所示：


图 3-15 设备和网络视图

然后在 ET200SP CM PTP 模块的端口组态中，选择 modbus 通信协议。如图 3-16 所示：

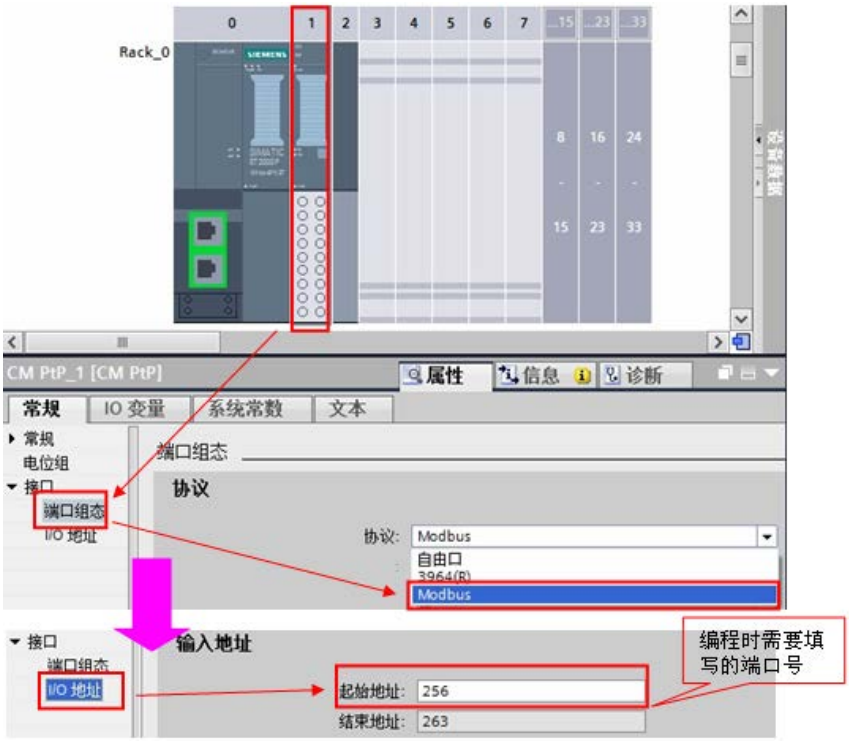



图 3-16 设置 ET200SP CM PTP 模块协议

### 3.4.1 编写通信程序

（1）OB1 编程

在项目的 OB1 组织块中添加 Modbus RTU 初始化功能块

“Modbus_Comm_Load”，并为该 FB 块增加一个背景数据块，本例中为 DB1

“Modbus_Comm_Load _DB”；然后在下一个网络中添加主站操作指令

“Modbus_Slave”， 为该 FB 块增加一个背景数据块，本例中为 DB2

“Modbus_ Slave _DB”；如下图 3-17 所示：

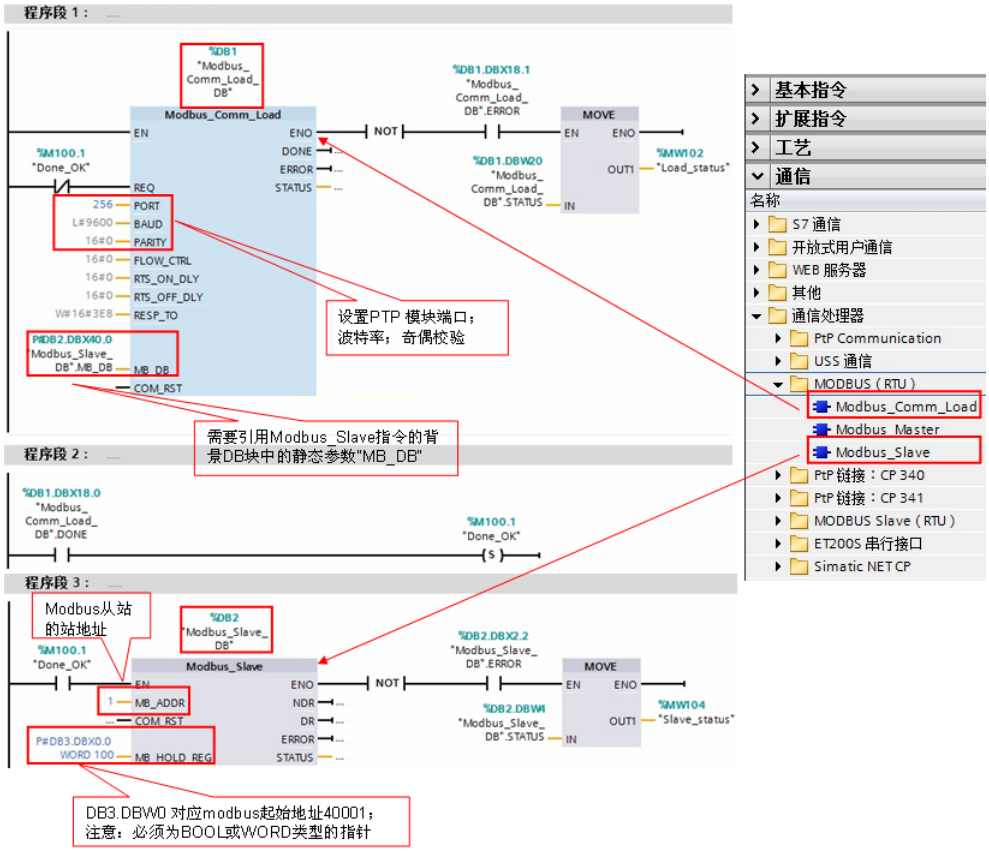



图 3-17 添加“Modbus_Comm_Load”和“Modbus_Slave”功能块

功能块“Modbus_Slave”的管脚参数如下表 3-5 所示：

| “Modbus_Slave”的管脚参数 | 管脚声明 | 数据类型 | 含义 |
|---|---|---|---|
| MB_ADDR | 输入 | UInt | Modbus 从站的标准寻址：<br>标准寻址范围（1 到 247）<br>扩展寻址范围（0 到 65535）<br>注意： 0 是广播地址 |
| COM_RST | 输入/输出 | Bool | Modbus_Slave 指令的初始化：指令在 TRUE 时执行。 随后会将 COM_RST 复位为 FALSE。 |
| MB_HOLD_REG | 输入 | Any | Modbus 保持寄存器 DB 的指针：<br>Modbus 保持寄存器可能为标志或数据块的存储区。 |
| NDR | 输出 | Bool | 可用的新数据： |

| | | | |
|---|---|---|---|
| | | | FALSE - 无新数据<br>TRUE - 表示新数据已由 Modbus 主站写入<br>如果上一个请求完成并且没有错误，NDR 位将变为 TRUE 并保持一个周期。 |
| DR | 输出 | Bool | 读取数据：<br>FALSE - 未读取数据<br>TRUE - 表示该指令已将 Modbus 主站接收到的数据存储在目标区域中。<br>如果上一个请求完成并且没有错误，DR 位将变为 TRUE 并保持一个周期。 |
| ERROR | 输出 | Bool | 如果上一个请求完成出错，则 ERROR 位将变为 TRUE 并保持一个周期。 STATUS 参数中的错误代码仅在 ERROR = TRUE 的周期内有效。 |
| STATUS | 输出 | Word | 通信状态信息，用于诊断。 |

表 3-5 功能块“Modbus_Slave”的管脚参数

（2）选择接口类型和创建数据块

ET200SP CM PTP 模块支持 RS 232，RS 422 和 RS 485 接口，根据通信对象的不同，需要将模块设置为不同的工作模式，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（多点主站，CM PtP (ET 200SP)）

3 = 全双工 (RS 422) 四线制模式（多点从站，CM PtP (ET 200SP)）

4 = 半双工 (RS485) 二线制模式

本例中以 485 为例，则需要在功能块“Modbus_Comm_Load”的背景块 DB1 中找到“MODE”参数，并将其启动值改为 4。如图 3-18 所示：

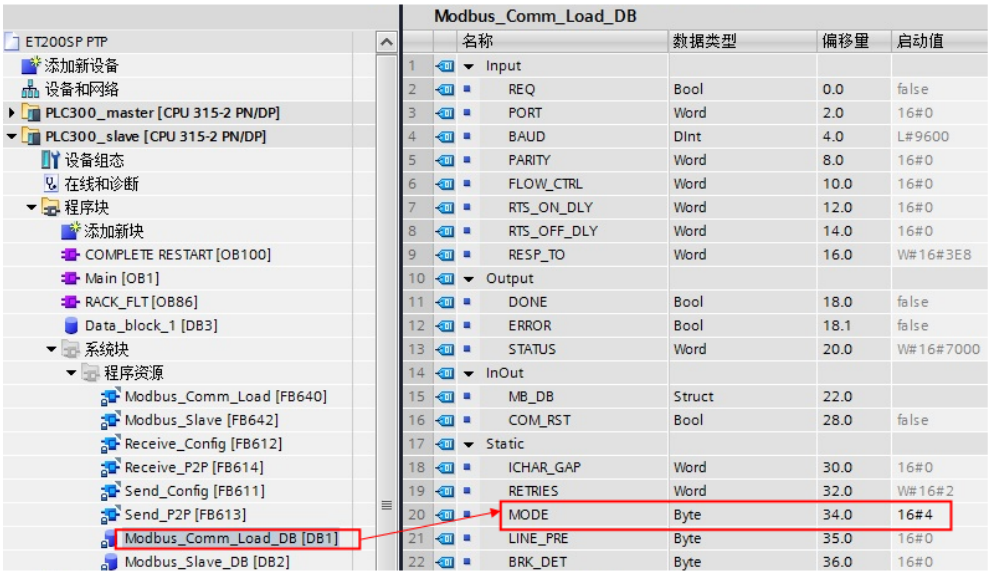



图 3-18 修改 ET200SP CM PTP 模块工作模式

然后，创建一个全局数据块用于匹配功能块“MB_Slave”的管脚参数“MB_HOLD_REG”，本例中创建数据块 DB3 “Data_block_1”，用于对应 modbus 保持寄存器，本例中 DB3 定义了 100 个字的数组对应于 modbus 地址 40001~40100。如下图 3-19 所示：

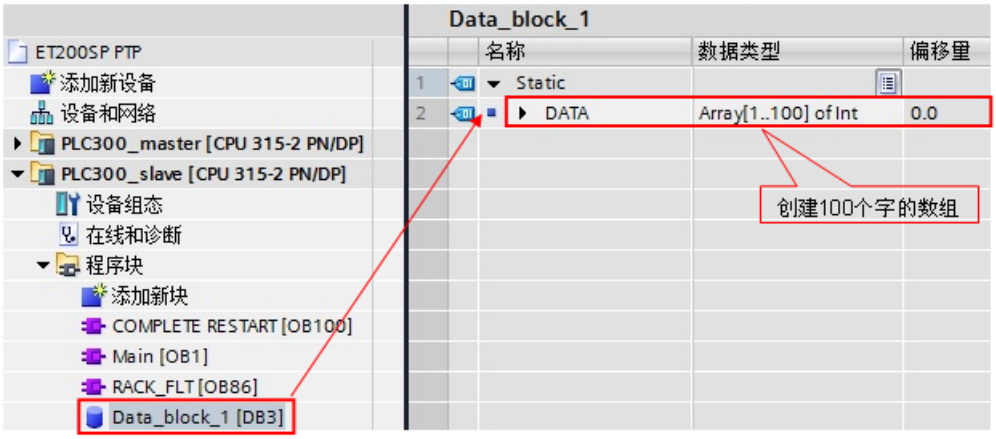



图 3-19 创建 DB 块

（3）modbus 地址对应关系

上面提到保持寄存器是由功能块“Modbus_Slave”的管脚参数“MB_HOLD_REG”定义的 DB 块关联，其对应如下表 3-6 所示：

| MB_SLAVE Modbus功能 | | | | S7-1500/300/400 | |
|---|---|---|---|---|---|
| 代码 | 功能 | 数据区 | 地址范围 | CPU DB数据区 | CPU地址 |
| 3 | 读字 | 保持寄存器 | 40001到49999 | MB_HOLD_REG | 字1到字9999 |
| | | | 400001到465535 | | 字1到字65535 |
| 6 | 写单个字 | 保持寄存器 | 40001到49999 | MB_HOLD_REG | 字1到字9999 |
| | | | 400001到465535 | | 字1到字65535 |
| 16 | 写字 | 保持寄存器 | 40001到49999 | MB_HOLD_REG | 字1到字9999 |
| | | | 400001到465535 | | 字1到字65535 |

表 3-6 Modbus 的寄存器地址映射表

对于其它数据类型，如线圈、离散输入、模拟量输入等通过功能块均已经与 S7-300 的过程映像区进行了映射，其映射地址对应如下表 3-7 所示：

| Modbus 功能 | | | | | | S7-1500 / S7-300 / S7-400 | | | |
|---|---|---|---|---|---|---|---|---|---|
| 代码 | 功能 | 数据区 | 地址区 | | | 数据区 | CPU 地址 | | |
| 01 | 读取位 | 输出 | 0 | 到 | 9998 | 输出的过程映像 | O0.0 | 到 | O1248.6 |
| 02 | 读取位 | 输入 | 0 | 到 | 9998 | 输入的过程映像 | I0.0 | 到 | I1248.6 |
| 04 | 读取字 | 输入 | 0 | 到 | 9998 | 输入的过程映像 | IW0 | 到 | IW19996 |
| 05 | 写入位 | 输出 | 0 | 到 | 9998 | 输出的过程映像 | O0.0 | 到 | O1248.6 |
| 15 | 写入位 | 输出 | 0 | 到 | 9998 | 输出的过程映像 | O0.0 | 到 | O1248.6 |

表 3-7 Modbus 地址映射表

（4）调用 OB100 进行初始化

在 OB100 中，分别对初始化指令“Modbus_Comm_Load”和主站指令“Modbus_Slave”的引脚“COM_RST”进行置位操作。如下图 3-20 所示：

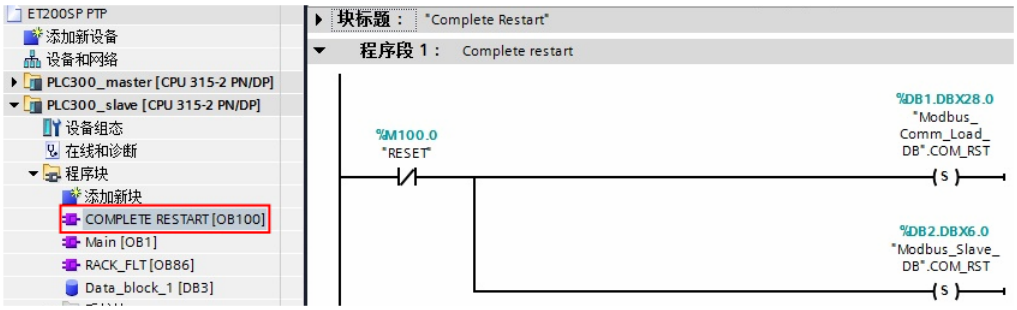



图 3-20 OB100 编程

（5）调用 OB86 防止掉站停机

本例中使用的 ET200SP 为 PROFINET IO 设备，为避免由于 IO 设备的掉站导致 CPU 停止，则需要添加组织块 OB86；并且需要在分布式 IO 设备恢复连接时，重新初始化 ET200SP CM PTP 模块，利用 OB86 的临时变量“Event_Class”的状态值对“Modbus_Comm_Load”和“Modbus_Slave”指令的

“COM_RST”引脚进行置位；同时，需要复位“Modbus_Comm_Load”的“REQ”引脚到达重新进行初始化的目的。如下图 3-21 所示：

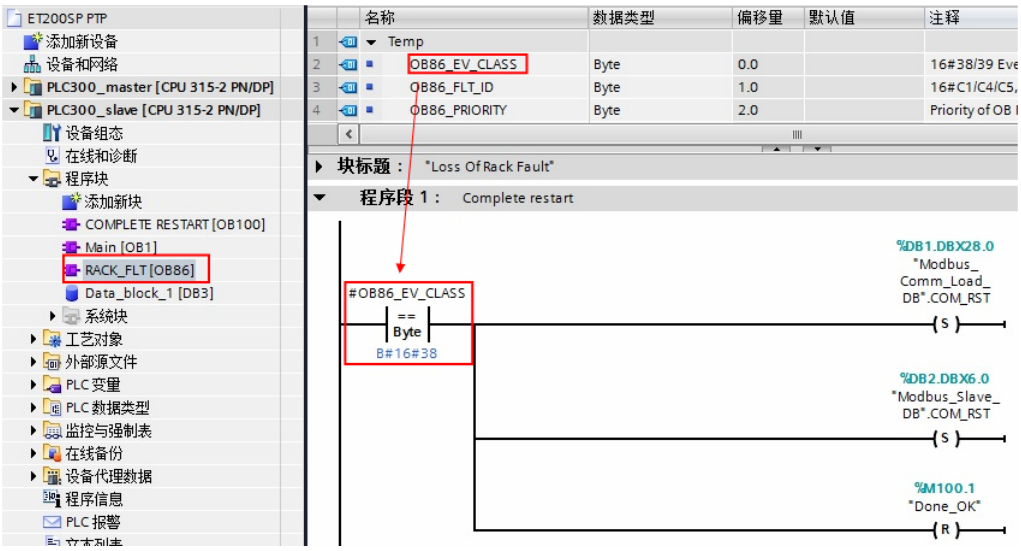



图 3-21 OB86 编程

说明：当有分布式 IO 掉站时，OB86 临时变量“Event_Class”=B#16#39；当有分布式 IO 恢复连接时，OB86 临时变量“Event_Class”=B#16#38；更多信息请查看 OB86 组织块的帮助说明。

### 3.4.2 下载程序

分配设备名称（注：如果使用的分布式 IO 是 Profibus DP，则跳过该步骤）：将软件切换到“网络视图”，找到 PN/IE 总线，查看设备名称是否正确。如图 3-22、3-23 所示：

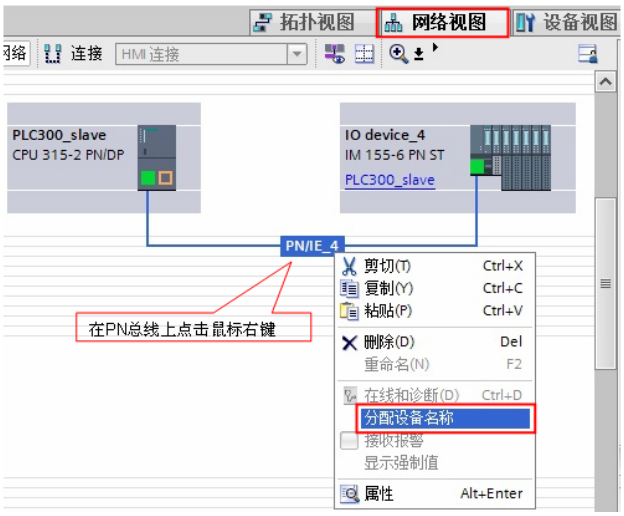



图 3-22 网络视图

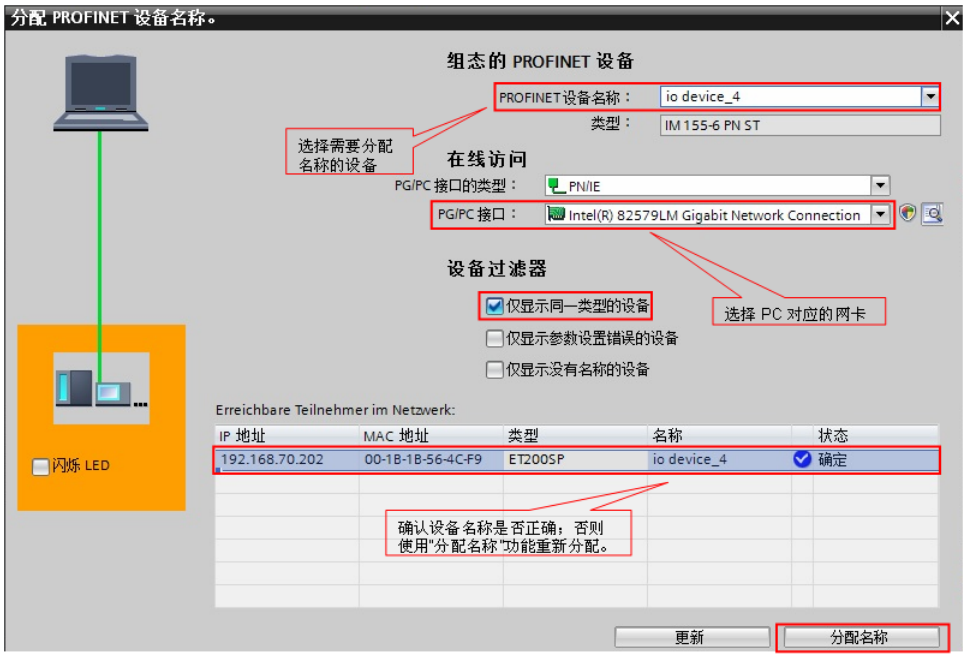



图 3-23 确认设备名称和 IP 地址

编译并下载程序到 PLC 中。

### 3.4.3 通信测试

由于 Modbus Slave 指令支持功能码 FC1，2，3，4，5，6，15，16，不同的功能码测试过程类似，因此本例中的测试以 FC3（读保持寄存器）为例来说明通讯测试的过程，本例中读取的 modbus 地址 40001~40010 中的数据将存放到 DB3 的前 10 个字中。对于其他功能码的测试将不再重复描述。

打开 ModScan32 软件，在“Connection——>Connect”中打开连接属性对话框，连接接口选择“Port1”，设置相应的波特率和奇偶校验等参数。如图 3-24 所示：

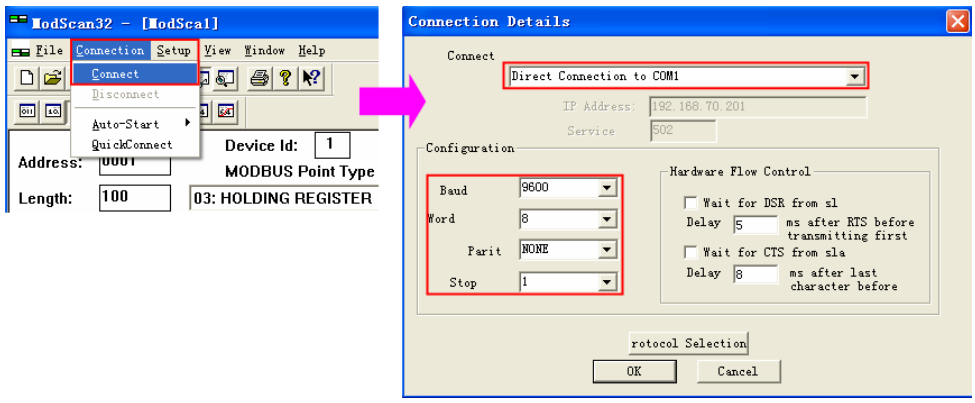
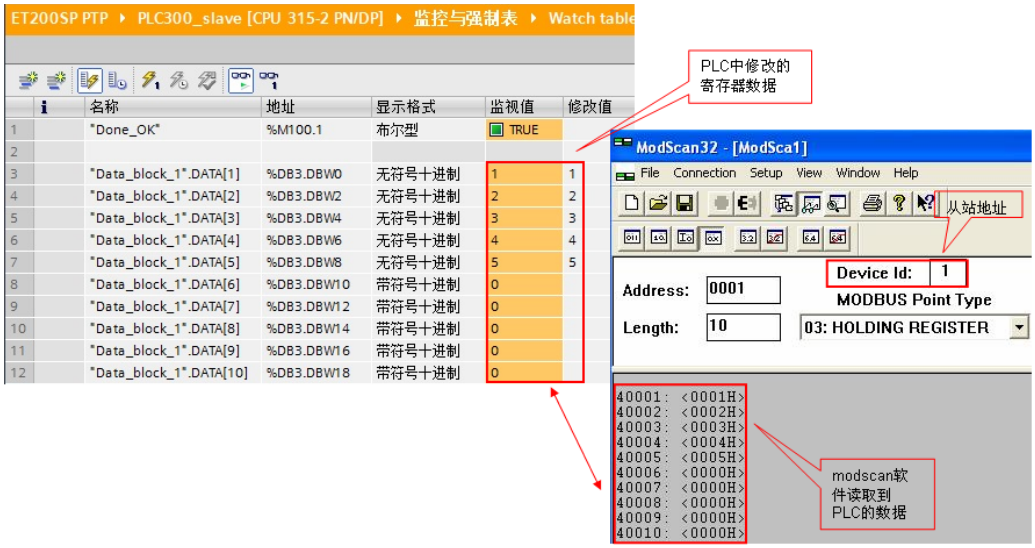



图 3-24 设置测试软件

然后，在 TIA Portal 中新建监控表，添加通信数据区，在线监控。如图 3-25 所示：




| | i | 名称 | 地址 | 显示格式 | 监视值 | 修改值 |
|---|---|---|---|---|---|---|
| 1 | | "Done_OK" | %M100.1 | 布尔型 | TRUE | |
| 2 | | | | | | |
| 3 | | "Data_block_1".DATA[1] | %DB3.DBW0 | 无符号十进制 | 1 | 1 |
| 4 | | "Data_block_1".DATA[2] | %DB3.DBW2 | 无符号十进制 | 2 | 2 |
| 5 | | "Data_block_1".DATA[3] | %DB3.DBW4 | 无符号十进制 | 3 | 3 |
| 6 | | "Data_block_1".DATA[4] | %DB3.DBW6 | 无符号十进制 | 4 | 4 |
| 7 | | "Data_block_1".DATA[5] | %DB3.DBW8 | 无符号十进制 | 5 | 5 |
| 8 | | "Data_block_1".DATA[6] | %DB3.DBW10 | 带符号十进制 | 0 | |
| 9 | | "Data_block_1".DATA[7] | %DB3.DBW12 | 带符号十进制 | 0 | |
| 10 | | "Data_block_1".DATA[8] | %DB3.DBW14 | 带符号十进制 | 0 | |
| 11 | | "Data_block_1".DATA[9] | %DB3.DBW16 | 带符号十进制 | 0 | |
| 12 | | "Data_block_1".DATA[10] | %DB3.DBW18 | 带符号十进制 | 0 | |

图 3-25 通讯测试