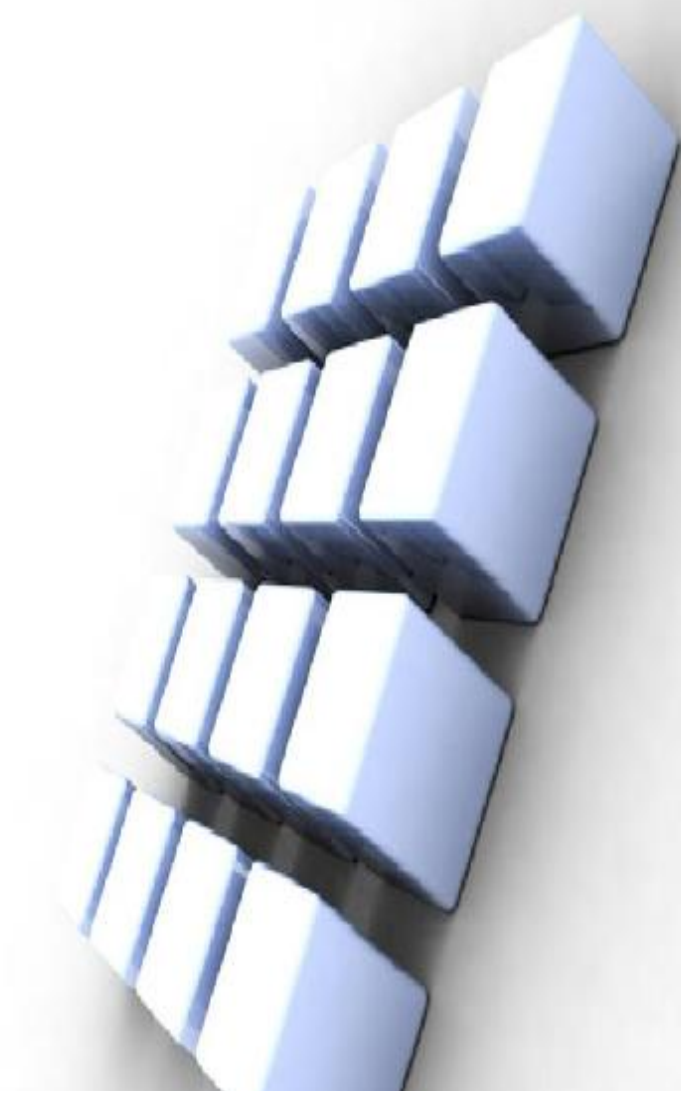
S7-200 Modbus 通信

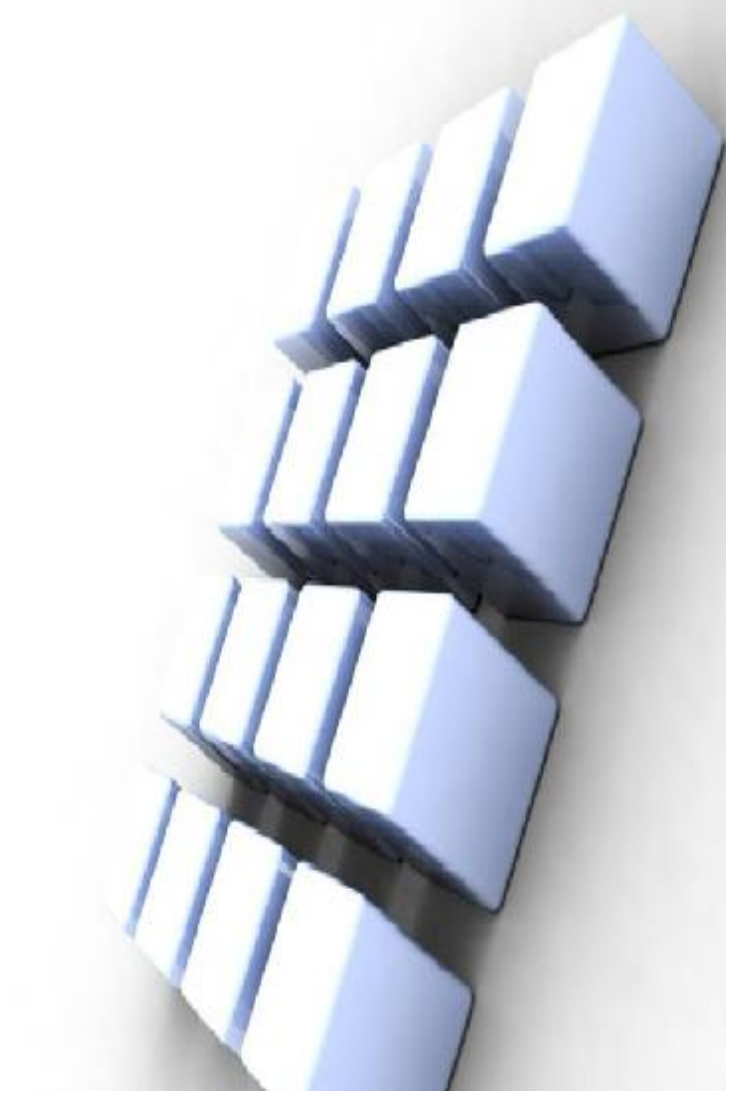
S7-200 Modbus 通信

- 概述
  - 定义
  - **Modbus** 报文传输格式
  - **Modbus** 主站协议指令
  - **Modbus** 从站协议指令
- **Micro/Win** 指令库的管理
  - 指令库的安装
  - 指令库的卸载
  - 指令库的添加
  - 指令库的删除
- 编程示例
  - 功能要求
  - 实现步骤



概述

- 定义
- **Modbus** 报文传输格式
- **Modbus** 主站协议指令
- **Modbus** 从站协议指令



定义

- 使用 **Modbus** 通信协议是 **Modicon** 公司提出的一种报文传输协议，它广泛应用于工业控制领域，并已经成为一种通用的行业标准。不同厂商提供的控制设置可通过 **Modbus** 协议连成通信网络，从而实现集中控制。

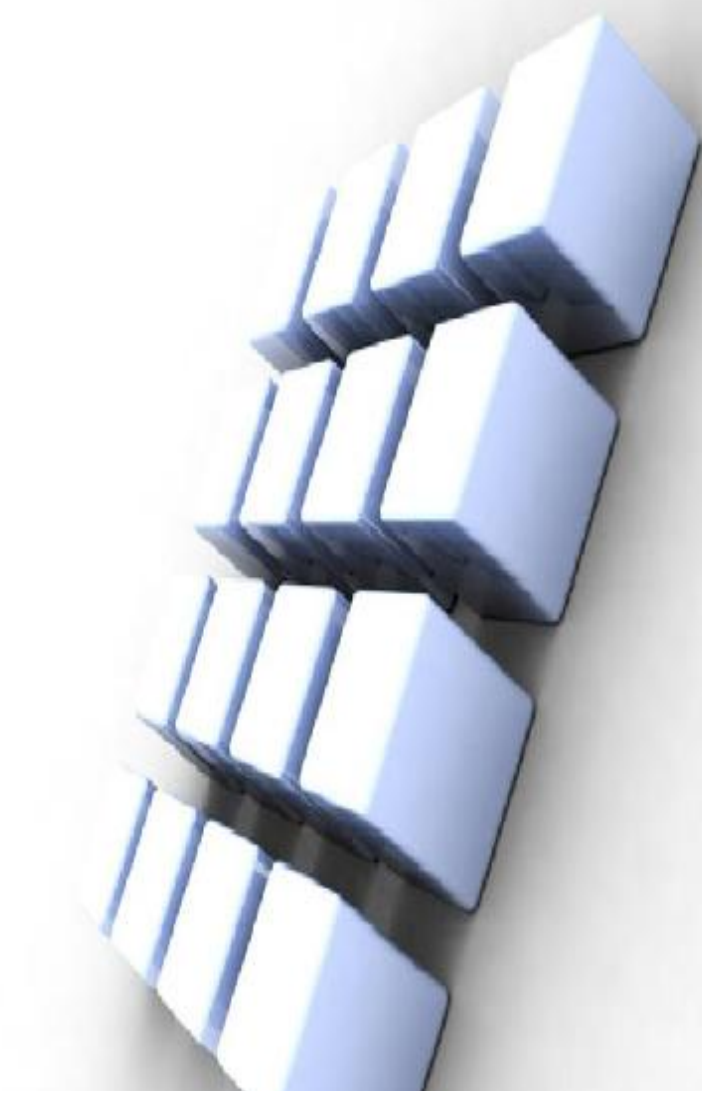
- 根据传输网络类型的区别，**Modbus** 通信协议又分为串行链路上的 **Modbus** 和基于 **TCP/IP** 协议的 **Modbus**。

- **Modbus** 串行链路协议只有一个主站，可以有**1~247**个从站。**Modbus** 通信只能从主站发起，从站在未收到主站的请求时，不能发送数据或互相通信。

- **Modbus** 串行链路协议的通信接口可采用 **RS-485** 接口，也可使用 **RS-232C** 接口。其中**RS-485** 接口可用于远距离通信，**RS-232C** 接口只能用于短距离通信。

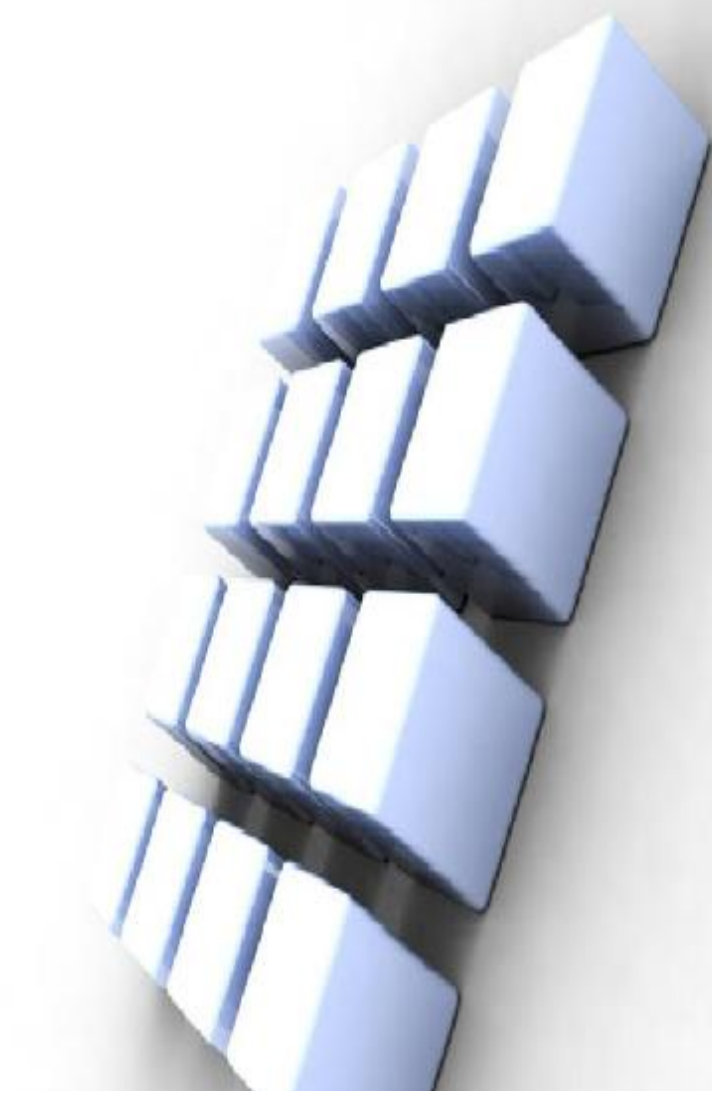
概述

- 定义
- **Modbus** 报文传输格式
  - **Modbus** 寻址
  - **ASCII** 模式
  - **RTU** 模式
- **Modbus** 主站协议指令
- **Modbus** 从站协议指令



Modbus 报文传输格式

- **Modbus 寻址**
- **ASCII 模式**
- **RTU 模式**



Modbus 寻址

- **Modbus** 地址通常是包含数据类型和偏移量的**5**个或**6**个字符值。第一个或前两个字符决定数据类型，最后的四个字符是符合数据类型的一个适当的值。**Modbus** 主设备指令能将地址映射至正确的功能，以便发送到从站。
- **1 Modbus** 主站寻址
- **Modbus**主设备指令支持下列**Modbus**地址：
- （**1**）**00001**至**09999**是离散输出（线圈）。
- （**2**）**10001**至**19999**是离散输入（触点）。
- （**3**）**30001**至**39999**是输入寄存器（通常是模拟量输入）。
- （**4**）**40001**至**49999**是保持寄存器。
- 所有**Modbus**地址均以**1**为基位，表示第一个数据值从地址**1**开始。有效地址范围将取决于从站。不同的从站将支持不同的数据类型和地址

Modbus 寻址

- **2 Modbus 从站寻址**
- **Modbus从站指令支持以下地址：**
- **（1）000001至000128是实际输出，对应于Q0.0——Q15.7。**
- **（2）010001至010128是实际输入，对应于I0.0——I15.7。**
- **（3）030001至030032是模拟输入寄存器，对应于AIW0至AIW2。**
- **（4）040001至04XXXX是保持寄存器，对应于V区。**
- **Modbus 从站协议允许您对Modbus主站可访问的输入、输出、模拟输入和保持寄存器（V区）的数量进行限定。MBUS_INIT 指令的参数 MaxIQ 指定 Modbus 主站允许访问的实际输入或输出（I或Q）的最大数量。MBUS_INIT 指令的 MaxAI 参数指定 Modbus 主站允许访问的输入寄存器（AIW）的最大数量。MBUS_INIT 指令的MaxHold 参数指定 Modbus 主站允许访问的保持寄存器（V存储区字）的最大数量。**

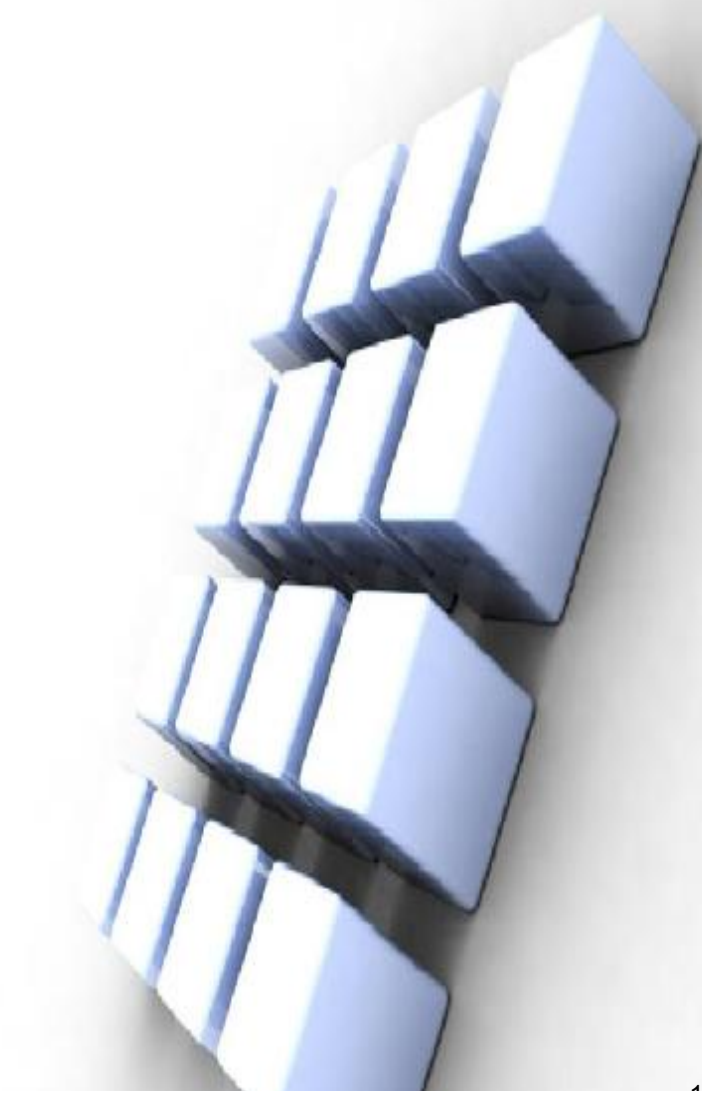
Modbus 寻址

- **Modbus 地址与 S7-200 地址对应关系如下表所示。**

000001	Q0.0
000002	Q0.1
000003	Q0.2
…	…
000127	Q15.6
000128	Q15.7
010001	I0.0
010002	I0.1
010003	I0.2
…	…
010127	I15.6
010128	I15.7
030001	AIW0
030002	AIW2
030003	AIW4
…	…
030032	AIW62
040001	HoldStart
040002	HoldStart+2
040003	HoldStart+4
…	…
04xxxx	HoldStart+2 x（xxxx-1）

Modbus 报文传输格式

- **Modbus 寻址**
- **ASCII 模式**
- **RTU 模式**



Modbus 报文传输格式

- Modbus 通信协议有 ASCII 和 RTU （远程传输单元）两种报文传输模式。Modbus 网络中所有的站必须采用相同的传输模式和串口参数。

- ASCII模式采用LRC（纵向冗余校验）方式进行校验，其报文格式如下表所示：

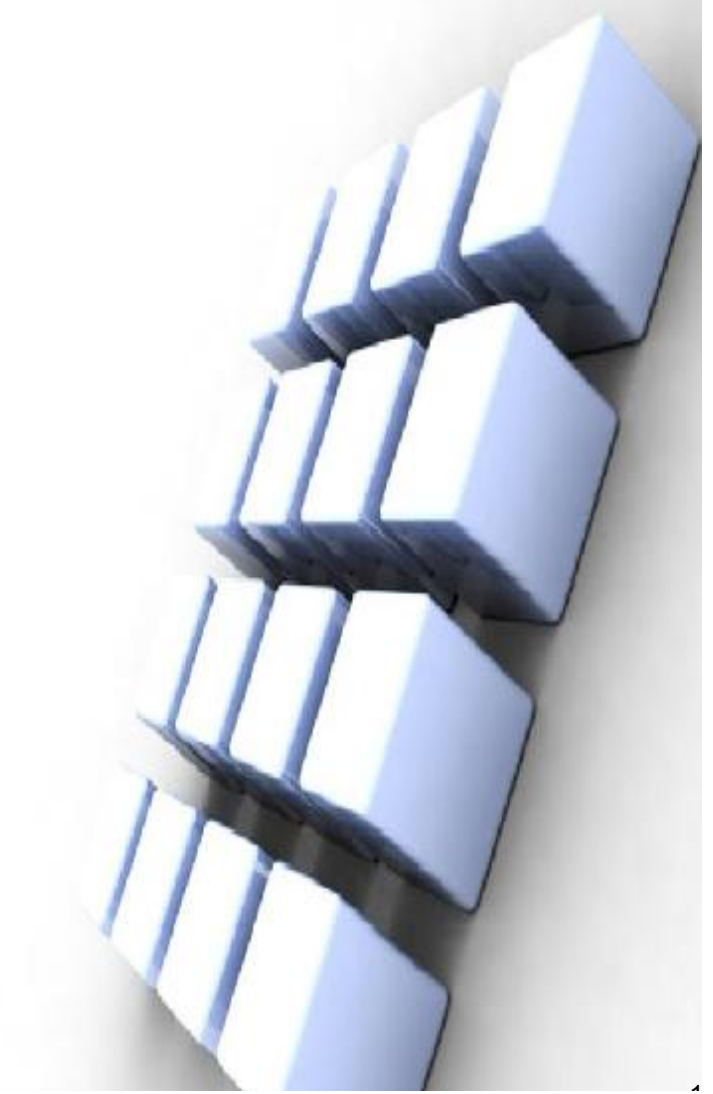
:	地址	功能码	数据字节数	数据1	…	数据n	LRC高字节	LRC低字节	回车	换行

- ASCII 模式中，报文帧中的每个8位字节都转换为两个 ASCII 码发送。报文中的每个ASCII码都由16进制字符组成，传输的每个字符都包括1个起始位、7个数据位、1个奇偶校验位、1个停止位；如果没有校验位，则有两个停止位。

- LRC计算时不包括开始的冒号符、LRC本身和回车换行符。

Modbus 报文传输格式

- **Modbus 寻址**
- **ASCII 模式**
- **RTU 模式**



Modbus 报文传输格式

- **RTU模式的报文格式如下表所示：**

地址	功能码	数据 1	…	数据 n	CRC 高字节	CRC 低字节

- **地址：Modbus 地址，1个字节。**
- **功能码：Modbus功能代码， 1个字节。Modbus协议支持的功能码共16条（1~16），其中西门子Modbus RTU协议库支持最常用的8条。**
- **信息数据：N个字节，格式与功能码有关。**
- **CRC ：循环冗余校验，两个字节。**

Modbus 报文传输格式

- 西门子**Modbus RTU**协议库支持最常用的**8**条功能码如下表所示。

功能码	描述
1	读取单个/多个线圈的实际输出状态。功能 1 返回任意数量输出点的接通/断开状态（Q）。
2	读取单个/多个线圈的实际输入状态。功能 2 返回任意数量的输入点的接通/断开状态（I）。
3	多个保持寄存器。功能 3 返回 V 存储器的内容。保持寄存器在 Modbus 下是字类型，在一个请求中最多可读 120 个字。
4	读单个/多个输入寄存器，返回模拟输入值。
5	写单个线圈（实际输出）。功能 5 将实际输出点设置为指定值。该输出点不是被强制，用户程序可以重写由 Modbus 的请求而写入的值。
6	写单个保持寄存器。功能 6 写一个单个保持寄存器的值到 S7--200 的 V 存储区。
15	写多个线圈（实际输出）。功能 15 写多个实际输出值到 S7--200 的 Q 映象区。起始输出点必须是一个字节的开始（如，Q0.0 或 Q2.0），并且要写的输出的数量是 8 的倍数。这是 Modbus 从站协议指令的限定。这些点不是被强制，用户程序可以重写由 Modbus 的请求而写入的值。
16	写多个保持寄存器。功能 16 写多个保持寄存器到 S7--200 的 V 区。在一个请求中最多可写 120 字。

Modbus 报文传输格式

- **RTU**模式下，报文中的每个**8**位字节被转化为两个**16**进制字符，然后以字节为单位进行传输，并采用**CRC**（循环冗余校验）方式进行校验。**RTU**模式的优点在于同波特率下有着比**ASCII**模式更高的传输效率。

- 目前支持 **Modbus** 通信的 **DCS** 系统和过程仪表大都采用基于串行接口的 **Modbus RTU** 模式，西门子提供了针对西门子 **PLC Modbus RTU** 通信的协议库。

- 如果要在西门子 **PLC** 上实现 **Modbus ASCII** 模式通信，用户可根据相关协议规定利用自由口模式自主编程实现。

Modbus 报文传输格式

- 西门子专门为 **Modbus RTU** 通信开发了指令库，极大地简化了 **Modbus RTU** 通信的开发，以便于快速实现相关应用。通过 **Modbus RTU** 从站指令库，使得 **S7-200** 可作为 **Modbus RTU** 中的从站设备集成到 **Modbus** 网络中，以实现与 **Modbus** 主站设备的通信。

- 而在最近推出的 **STEP7-Micro/Win SP5** 升级包中，西门子又增加了 **Mobus RTU Master** 指令库，使得 **S7-200 CPU** 可作为 **Modbus** 主站，实现与 **Modbus RTU** 从站的通信任务。

- 要使用 **Modbus** 指令库必须注意以下四点：

- （1）使用 **Modbus** 指令库前，需要将其安装到 **Step7-Micro/Win** 中，**STEP 7-Micro/WIN** 必须为 **V3.2** 或以上版本。

- （2）**S7-200 CPU** 必须是固化程序修订版**2.00**或最好支持 **Modbus** 主设备协议库**(CPU MLFB 21x-2xx23-0XB0)**。

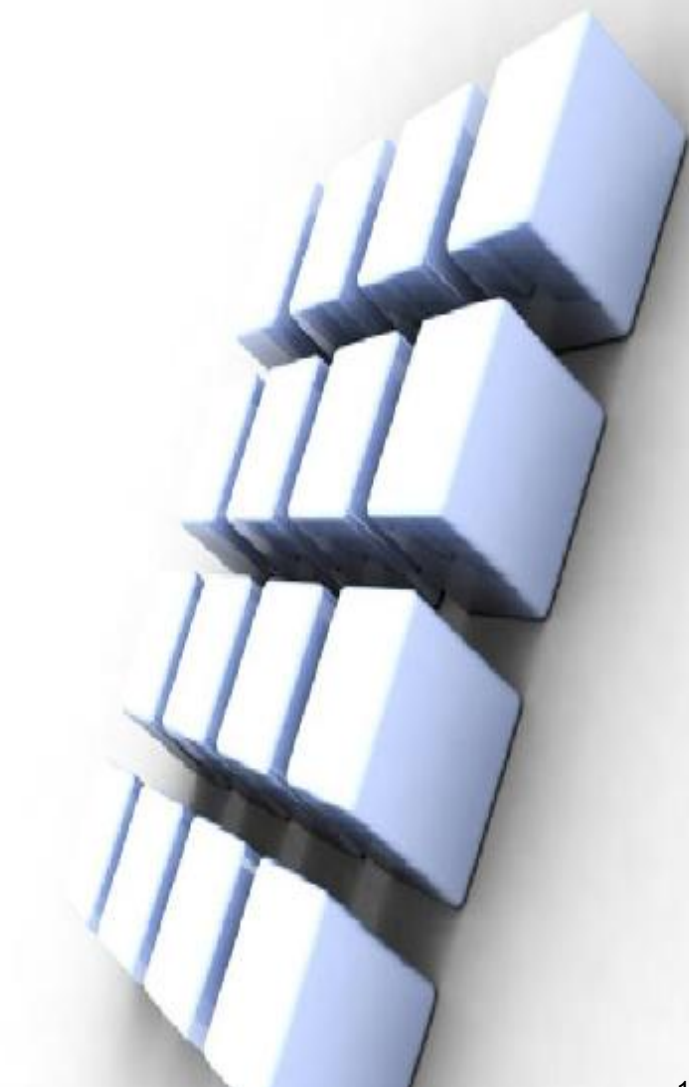
Modbus 报文传输格式

- （3）由于目前已经推出了针对端口0和端口1的 Modbus RTU 主站指令库 Modbus Master Port0 和 Modbus Master Port1、以及针对端口0的 Modbus RTU 从站指令库，故可利用指令库实现端口0的 Modbus RTU 主/从站通信。

- （4）一旦 CPU 端口被用于 Modbus RTU 主/从站协议通信时，该端口就无法用于任何其它用途，包括与 STEP 7-Micro/WIN 通讯。

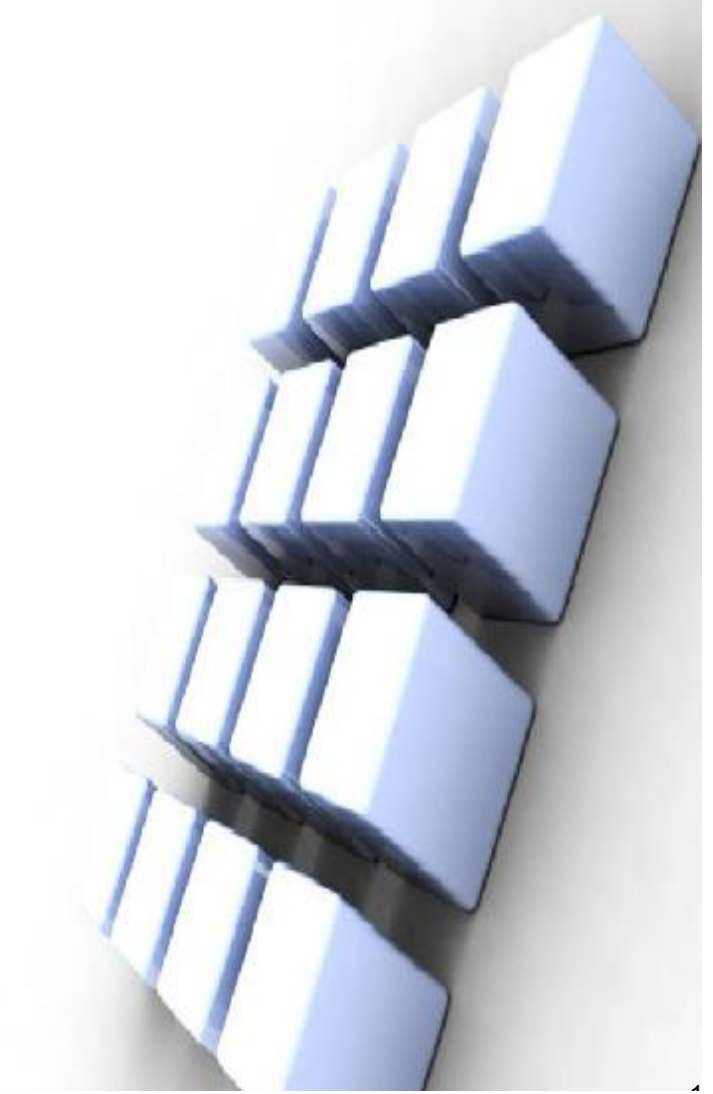
概述

- 定义
- **Modbus** 报文传输格式
- **Modbus** 主站协议指令
  - **MBUS_CTRL** 指令
  - **MBUS_MSG** 指令
- **Modbus** 从站协议指令
  - **MBUS_INIT** 指令
  - **MBUS_SLAVE** 指令



Modbus 主站协议指令

- **MBUS_CTRL 指令**
- **MBUS_MSG 指令**

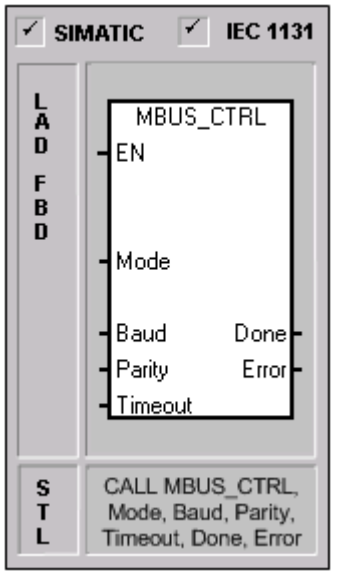


MBUS_CTRL 指令

- 西门子 **Modbus** 主站协议库包括两条指令：**MBUS_CTRL** 指令和 **MBUS_MSG** 指令。
- **MBUS_CTRL** 指令用于初始化主站通信，**MBUS_MSG** 指令(或用于端口1的**MBUS_MSG_P1**)用于启动对**Modbus**从站的请求并处理应答。
- **MBUS_CTRL** 指令用于 **S7-200** 端口**0**的 **MBUS_CTRL** 指令(或用于端口1的 **MBUS_CTRL_P1** 指令)可初始化、监视或禁用 **Modbus** 通讯。在使用 **MBUS_MSG** 指令之前，必须正确执行 **MBUS_CTRL** 指令。指令完成后立即设定“完成”位，才能继续执行下一条指令。
- **MBUS_CTRL** 指令在每次扫描且**EN**输入打开时执行。**MBUS_CTRL** 指令必须在每次扫描时(包括首次扫描)被调用，以允许监视随 **MBUS_MSG** 指令启动的任何突出消息的进程。除非每次调用 **MBUS_CTRL**，否则**Modbus** 主设备协议将不能正确运行。

MBUS_CTRL 指令

- 西门子**EN**：指令使能位。



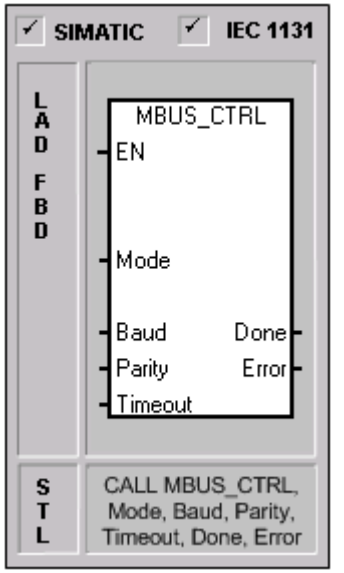
- **Mode**：“模式”参数。“模式”输入数值选择通讯协议。
  输入值**1**将**CPU**端口分配给**Modbus**协议并启用该协议。
  输入值**0**将**CPU**端口分配给**PPI**系统协议，
  并禁用**Modbus**协议。

- **Baud**：“波特率”参数。**MBUS_CTRL**指令支持的
  波特率为**1200**、**2400**、**4800**、**9600**、**19200**、**38400**、**57600**或**115200bit/s**。

- **Parity**：“奇偶校验”参数。“奇偶校验”参数被设为与**Modbus**从站奇偶校验相匹配。所有设置使用一个起始位和一个停止位。可接受的数值为：
- **0**　无奇偶校验
- **1**　奇校验
- **2**　偶校验

MBUS_CTRL 指令

- **Timeout**：“超时”参数。“超时”参数设为等待来自从站应答的毫秒时间数。“超时”数值可以设置的范围为1毫秒到32767毫秒。典型值是1000毫秒(1秒)。“超时”参数应该设置的足够大，以便从站有时间对所选的波特率作出应答。



- **Done**：**MBUS_CTRL**指令成功完成时，“完成”输出为**1**，否则为**0**。

- **Error**：“错误”输出代码。“错误”输出代码由反应执行该指令的结果的特定数字构成。“错误”输出代码的含义如下：

- **0** 无错误
- **1** 奇偶校验选择无效
- **2** 波特率选择无效
- **3** 超时选择无效
- **4** 模式选择无效

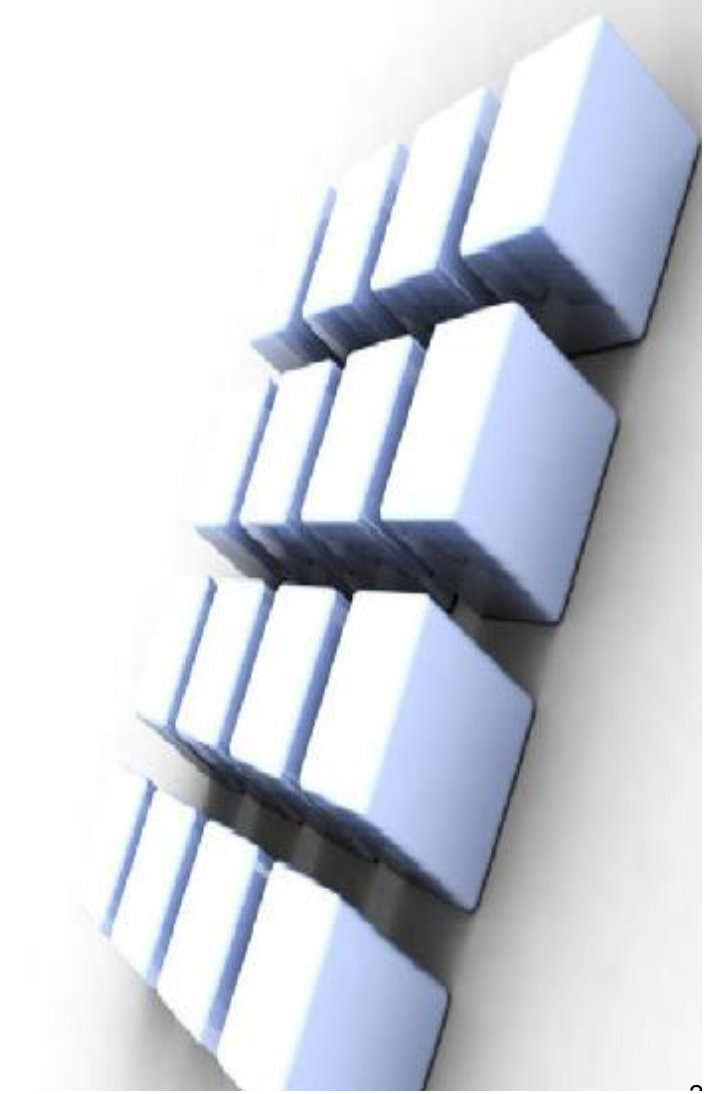
MBUS_CTRL 指令

- 上述参数支持的操作数和数据类型如下表所示。

输入/输出	操作数	数据类型
Mode	I，Q，M，S，SM，T，C，V，L	布尔
Baud	VD，ID，QD，MD，SD，SMD，LD，AC，Constant，*VD，*AC，*LD	双字
Parity	VB，IB，QB，MB，SB，SMB，LB，AC，Constant，*VD，*AC，*LD	字节
Timeout	VW，IW，QW，MW，SW，SMW，LW，AC，Constant，*VD，*AC，*LD	字
Done	I，Q，M，S，SM，T，C，V，L	布尔
Error	VB，IB，QB，MB，SB，SMB，LB，AC，*VD，*AC，*LD	字节

Modbus 主站协议指令

- **MBUS_CTRL 指令**
- **MBUS_MSG 指令**



MBUS_MSG 指令

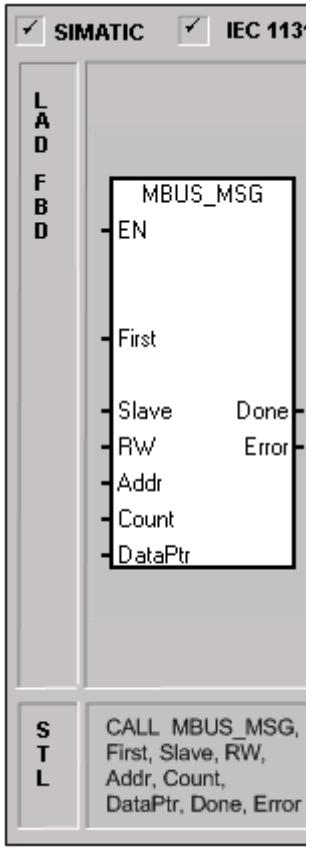
- **MBUS_MSG** 指令(或用于端口1的 **MBUS_MSG_P1**)用于启动对 **Modbus**从站的请求并处理应答。

- 当 **EN** 输入和“首次”输入都为**1**时，**BUS_MSG** 指令启动对 **Modbus** 从站的请求。发送请求、等待应答、并处理应答通常需要多次扫描。**EN**输入必须打开以启用请求的发送，并应该保持打开直到“完成”位被置位。

- 必须注意的是，一次只能激活一条 **MBUS_MSG** 指令。如果启用了多条 **MBUS_MSG** 指令，则将处理所执行的第一条 **MBUS_MSG** 指令，之后的所有 **MBUS_MSG** 指令将中止并产生错误代码**6**。

MBUS_MSG 指令

- **EN**：指令使能位。
- **First**：“首次”参数。“首次”参数应该在有新请求要发送时才打开以进行一次扫描。“首次”输入应当通过一个边沿检测元素(例如上升沿)打开，这将导致请求被传送一次。
- **Slave**：“从站”参数。“从站”参数是Modbus从站的地址。允许的范围是0到247。地址0是广播地址，只能用于写请求。不存在对地址0的广播请求的应答。
  并非所有的从站会支持广播地址，S7-200 Modbus 从站协议库不支持广播地址。
- **RW**：“读写”参数。“读写”参数指定是否要读取或写入该消息。“读写”参数允许使用下列两个值：0——读，1——写。



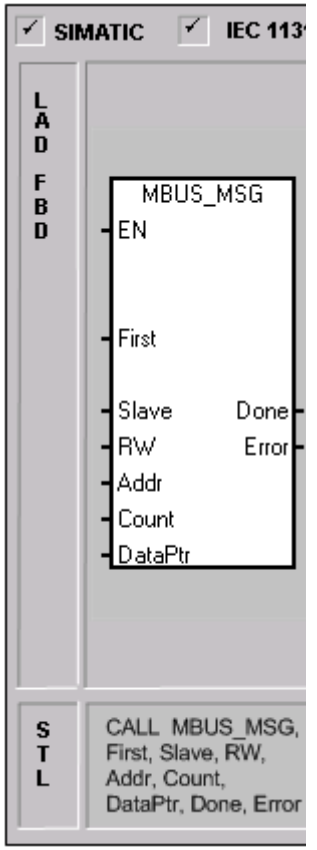
MBUS_MSG 指令

- **Addr**：“地址”参数。“地址”参数是起始的**Modbus** 地址。允许使用下列取值范围：

- **00001**至**09999**是离散输出(线圈)
- **10001**至**19999**是离散输入(触点)
- **30001**至**39999**是输入寄存器
- **40001**至**49999**是保持寄存器

- 其中离散输出(线圈)和保持寄存器支持读取和写入请求，而离散输入(触点)和输入寄存器仅支持读取请求。
- “地址”的具体值应与**Modbus**从站支持的地址一致。

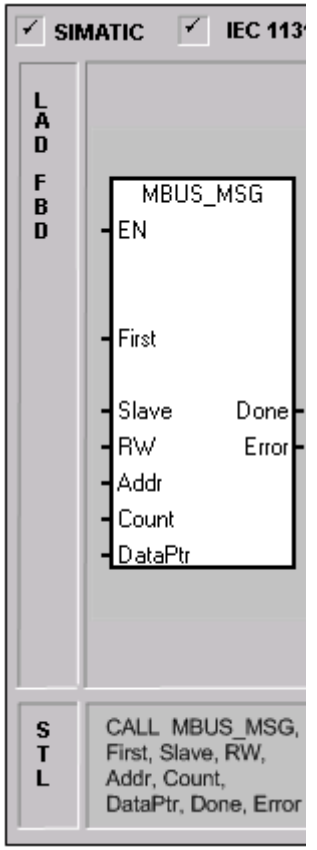
- **Count**：“计数”参数。“计数”参数指定在该请求中读取或写入的数据元素的数目。“计数”数值是位数(对于位数据类型)和字数(对于字数据类型)。



MBUS_MSG 指令

- 根据Modbus协议，“计数”参数与 Modbus 地址存在以下对应关系：

地址	计数
0xxxx	“计数”是要读取或写入的位数。
1xxxx	“计数”是要读取的位数。
3xxxx	“计数”是要读取的输入寄存器的字数。
4xxxx	“计数”是要读取或写入的保持寄存器的字数。



- **MBUS_MSG** 指令将读取或写入最大120个字或1920个位(240字节的数据)。“计数”的实际限值还取决于 **Modbus** 从站中的限制。

- **DataPtr**：“DataPtr” 参数。“DataPtr” 参数是指向 **S7-200 CPU** 的 **V** 存储器中与读取或写入请求相关的数据的间接地址指针。对于读取请求，**DataPtr** 应指向用于存储从 **Modbus** 从站读取的数据的第一个CPU存储器位置。对于写入请求，**DataPtr** 应指向要发送到Modbus从站的数据的第一个CPU存储器位置。

MBUS_MSG 指令

- **Done**：完成输出。完成输出在发送请求和接收应答时关闭。“完成”输出在应答完成或**MBUS_MSG**指令因错误而中止时打开。

- **Error**：“错误”输出仅当“完成”输出打开时有效。低位编号的错误代码(1到8)是由**MBUS_MSG**指令检测到的错误。这些错误代码通常指示与**MBUS_MSG**指令的输入参数有关的问题，或接收来自从站的应答时出现的问题。奇偶校验和**CRC**错误指示存在应答但是数据未正确接收。这通常是由电气故障(例如连接有问题或者电噪声)引起的。

- 高位编号的错误代码(从**101**开始)是由**Modbus**从站返回的错误。这些错误指示从站不支持所请求的功能，或者所请求的地址(或数据类型或地址范围)不被**Modbus**从站支持。

- **MBUS_MSG**指令错误代码含义如下所示：

MBUS_MSG 指令

- **MBUS_MSG 错误代码含义表**

MBUS_MSG 错误代码	说明
0	无错误。
1	应答时奇偶校验错误：仅当使用偶校验或奇校验时才会发生。传输被干扰，可能会收到不正确的数据。该错误通常是由电气故障(例如错误接线或者影响通讯的电噪声)引起的。
2	保留位，暂未启用。
3	接收超时：在"超时"时间内，没有来自从站的应答。可能有以下一些原因：与从站的电气连接有问题、主设备和从站设置为不同的波特率/奇偶校验设置，以及错误的从站地址。
4	请求参数出错：一个或多个输入参数(从站、读写、地址或计数)被设置为非法值。检查文档中输入参数的允许值。
5	Modbus 主设备未启用：在调用 MBUS_MSG 前，每次扫描时都调用 MBUS_CTRL。
6	Modbus 忙于处理另一个请求：一次只能激活一条 MBUS_MSG 指令。
7	应答时出错：收到的应答与请求不相关。这表示从站中出现了某些错误，或者错误的从站应答了请求。
8	应答时 CRC 错误：传输被干扰，可能会收到不正确的数据。该错误通常是由电气故障(例如错误接线或者影响通讯的电噪声)引起的。

MBUS_MSG 指令

- **MBUS_MSG 错误代码含义表（续）**

101	从站不支持在该地址处所请求的功能，请参阅表 1-4。
102	从站不支持数据地址："地址"加上"计数"所要求的地址范围超出了从站所允许的地址范围。
103	从站不支持数据类型：该"地址"类型不被从站支持。
104	从站故障。
105	从站已接受消息但应答延迟：这是 MBUS_MSG 的错误，用户程序应在稍后重新发送请求。
106	从站忙，因此拒绝消息：可以再次尝试相同的请求，以获得应答。
107	从站因未知原因而拒绝消息。
108	从站存储器奇偶校验错误：从站中有错误。

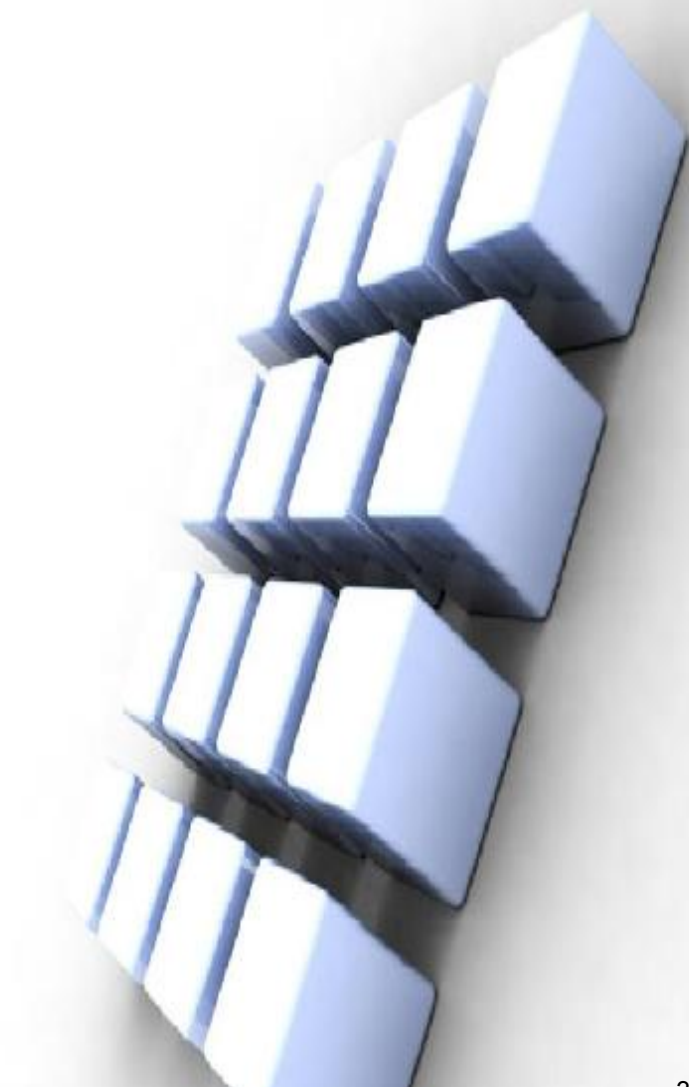
MBUS_MSG 指令

- 上述参数支持的操作数和数据类型如下表所示。

输入/输出	操作数	数据类型
First	布尔	I，Q，M，S，SM，T，C，V，L(以上升沿检测元素为条件的功率流)
Slave	字节	VB，IB，QB，MB，SB，SMB，LB，AC，Constant，*VD，*AC，*LD
RW	字节	VB，IB，QB，MB，SB，SMB，LB，AC，Constant，*VD，*AC，*LD
Addr	双字	VD，ID，QD，MD，SD，SMD，LD，AC，Constant，*VD，*AC，*LD
Count	整型	VW，IW，QW，MW，SW，SMW，LW，AC，Constant，*VD，*AC，*LD
DataPtr	双字	&VB
Done	布尔	I，Q，M，S，SM，T，C，V，L
Error	字节	VB，IB，QB，MB，SB，SMB，LB，AC，*VD，*AC，*LD

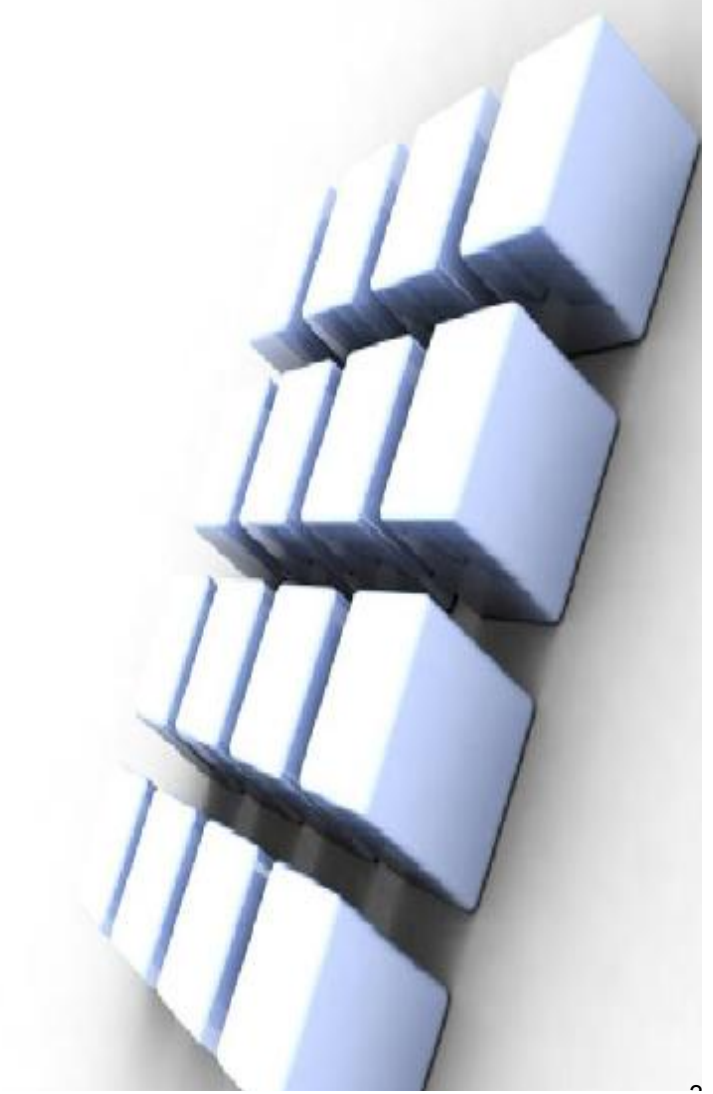
概述

- 定义
- **Modbus** 报文传输格式
- **Modbus** 主站协议指令
  - **MBUS_CTRL** 指令
  - **MBUS_MSG** 指令
- **Modbus** 从站协议指令
  - **MBUS_INIT** 指令
  - **MBUS_SLAVE** 指令



Modbus 从站协议指令

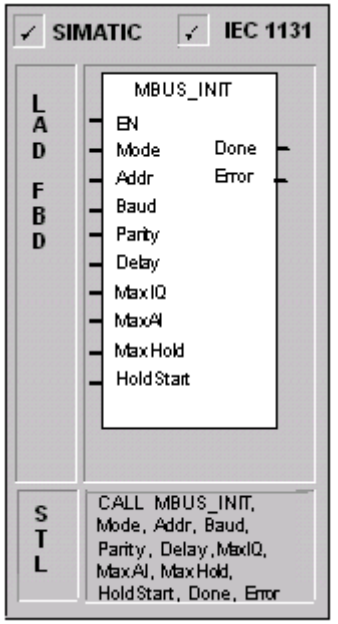
- **MBUS_INIT 指令**
- **MBUS_SLAVE 指令**



MBUS_INIT 指令

- 西门子 **Modbus** 从站协议库包括两条指令：**MBUS_INIT** 指令和 **MBUS_SLAVE** 指令。
- **MBUS_INIT** 指令用于启用和初始化或禁止**Modbus** 通讯。
- **MBUS_SLAVE** 指令用于为 **Modbus** 主设备发出的请求服务。
- **MBUS_INIT** 指令用于启用和初始化或禁止 **Modbus** 从站通讯。在使用**MBUS_SLAVE** 指令之前，必须正确执行 **MBUS_INIT** 指令。指令完成后立即设定 “完成” 位，才能继续执行下一条指令。

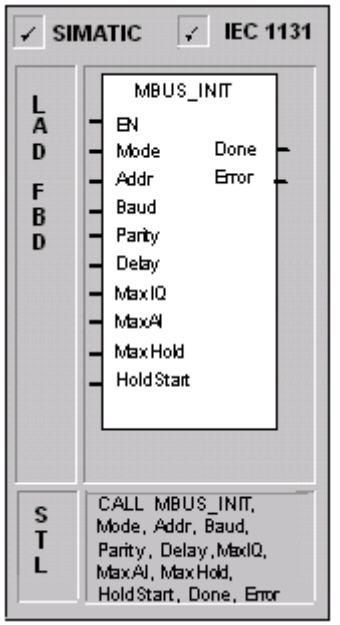
MBUS_INIT 指令



- **EN**：指令使能位。
- **Mode**：模式选择，启动/停止 **Modbus** 从站通信。**Mode** 参数允许使用以下两个数值：**1**——启动，**0**——停止。
- **Address**：从站地址，**MODBUS**从站地址，取值**1~247**。
- **Baud**：波特率，可选**1200**，**2400**，**4800**，**9600**，**19200**，**38400**，**57600**，**115200**。
- **Parity**：奇偶校验，**0**=无校验；**1**=奇校验；**2**=偶校验。
- **Delay**：延时，附加字符间延时，缺省值为**0**。
- **MaxIQ**：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**。

MBUS_INIT 指令

- **MaxAI**：最大AI字数，参与通信的最大AI通道数，可为16或32。
- **MaxHold**：设定供Modbus地址4xxxx使用的V存储器中的字保持寄存器数目。
- **HoldStart**：保持寄存器区起始地址，以&VBx指定（间接寻址方式）。
- **Done**：初始化完成标志，成功初始化后置1。
- **Error**：初始化错误代码。



MBUS_INIT 指令

- **MBUS_INIT 指令错误代码的含义如下表所示：**

错误代码	说明
0	无错误
1	内存范围错误
2	非法波特率或奇偶校验
3	非法从属地址
4	非法 Modbus 参数值
5	保持寄存器与 Modbus 从属符号重叠
6	收到奇偶校验错误
7	收到 CRC 错误
8	非法功能请求/功能不受支持
9	请求中的非法内存地址
10	从属功能未启用

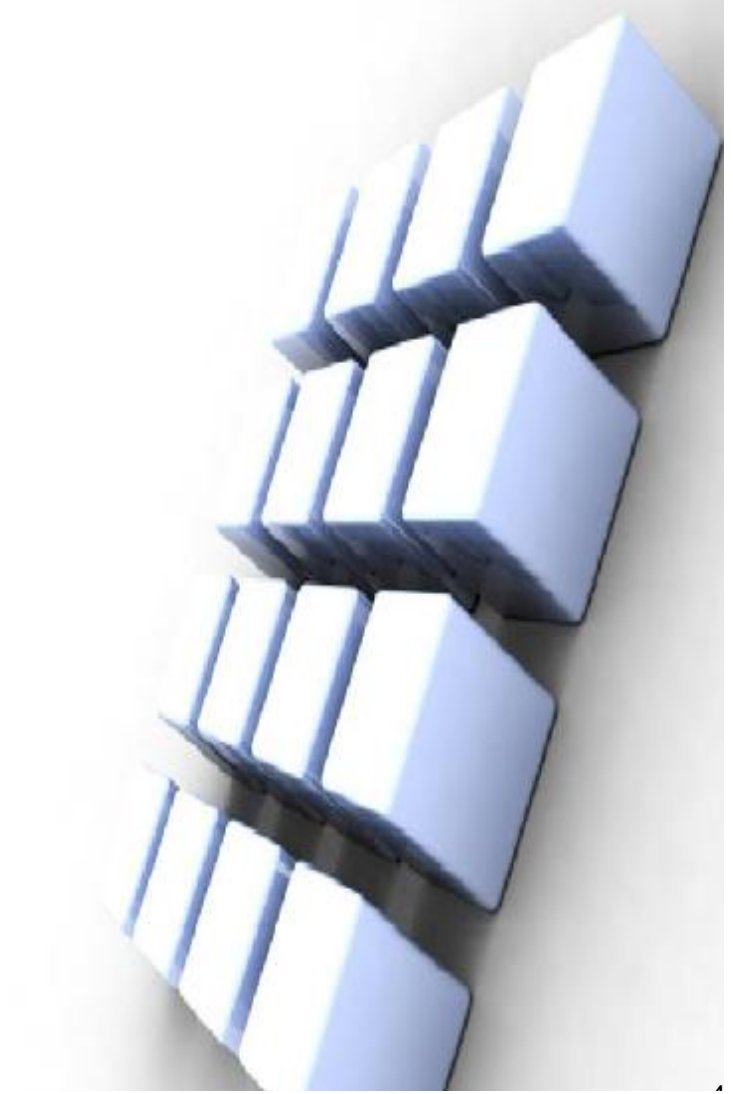
MBUS_INIT 指令

- 上述参数支持的操作数和数据类型如下表所示。

输入/输出	操作数	数据类型
模式、地址、奇偶校验	VB，IB，QB，MB，SB，SMB，LB，AC，Constant，*VD，*AC，*LD	字节
波特、HoldStart	VD，ID，QD，MD，SD，SMD，LD，AC，Constant，*VD，*AC，*LD	双字
延时、MaxIQ、MaxAI、MaxHold	VW，IW，QW，MW，SW，SMW，LW，AC，Constant，*VD，*AC，*LD	字
完成	I，Q，M，S，SM，T，C，V，L	布尔
错误	VB，IB，QB，MB，SB，SMB，LB，AC，*VD，*AC，*LD	字节

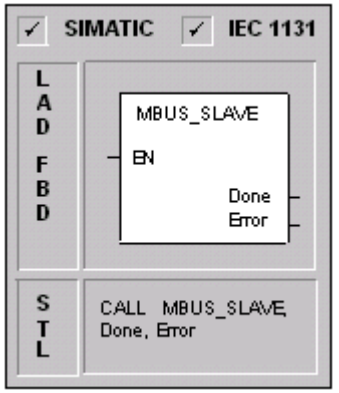
Modbus 从站协议指令

- **MBUS_INIT 指令**
- **MBUS_SLAVE 指令**



MBUS_SLAVE 指令

- **MBUS_SLAVE** 指令被用于为 **Modbus** 主设备发出的请求服务，并且必须在每次扫描时执行，以便允许该指令检查和回答 **Modbus** 请求。**MBUS_SLAVE** 指令无输入参数，在每次扫描且 **EN** 输入开启时执行。



- **EN**：指令使能位。
- **Done**：**Modbus**执行通信中时置**1**，无 **MODBUS** 通信活动时为 **0**。
- **Error**：错误代码。

MBUS_SLAVE 指令

- **MBUS_SLAVE** 指令错误代码的含义如下表所示。

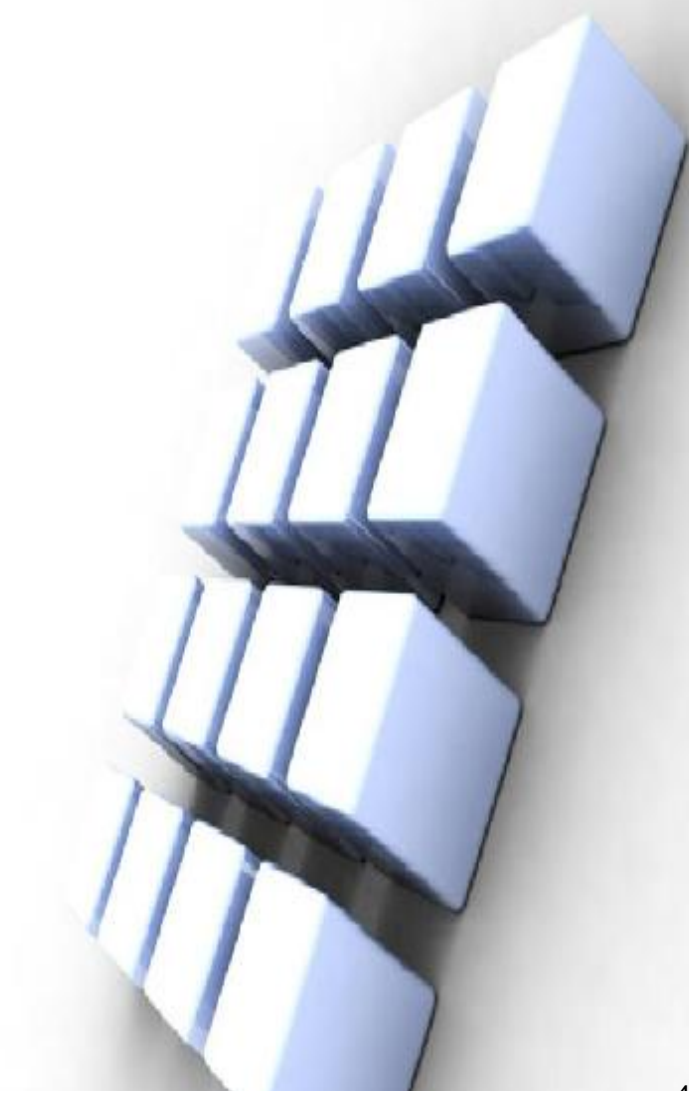
错误代码	说明
0	无错误
1	内存范围错误
2	非法波特率或奇偶校验
3	非法从属地址
4	非法 Modbus 参数值
5	保持寄存器与 Modbus 从属符号重叠
6	收到奇偶校验错误
7	收到 CRC 错误
8	非法功能请求/功能不受支持
9	请求中的非法内存地址
10	从属功能未启用

- 上述参数支持的操作数和数据类型如下表所示。

输入/输出	操作数	数据类型
完成	I，Q，M，S，SM，T，C，V，L	布尔
错误	VB，IB，QB，MB，SB，SMB，LB，AC，*VD，*AC，*LD	字节

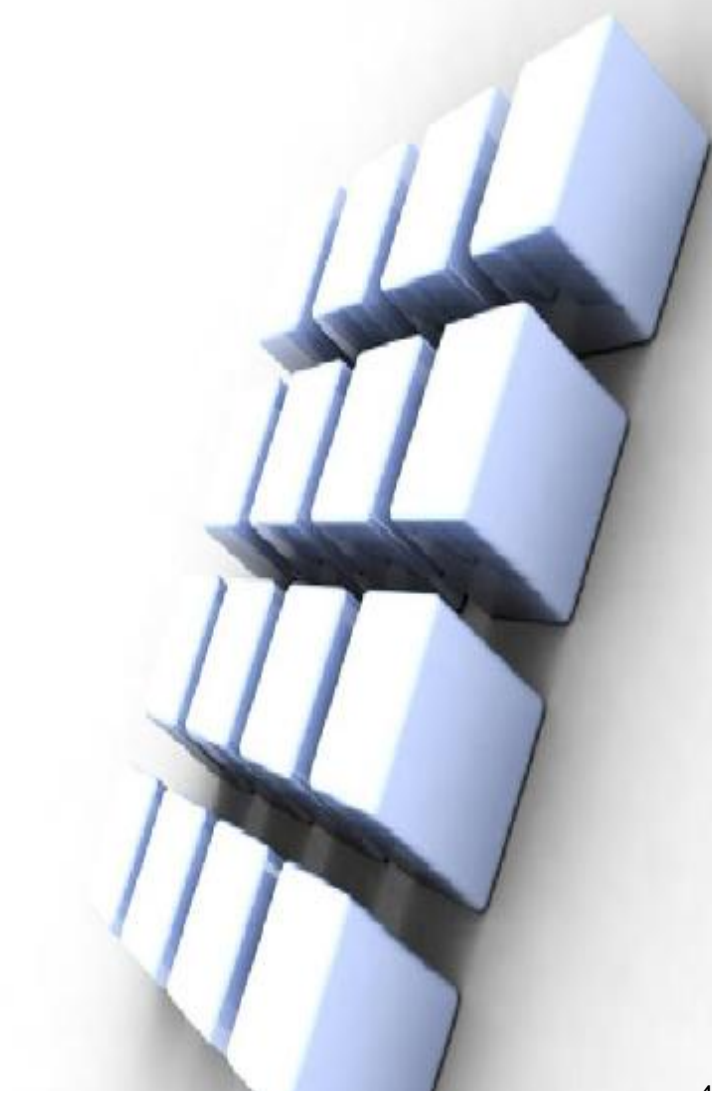
S7-200 Modbus 通信

- 概述
  - 定义
  - **Modbus** 报文传输格式
  - **Modbus** 主站协议指令
  - **Modbus** 从站协议指令
- **Micro/Win** 指令库的管理
  - 指令库的安装
  - 指令库的卸载
  - 指令库的添加
  - 指令库的删除
- 编程示例
  - 功能要求
  - 实现步骤



Micro/Win 指令库的管理

- 指令库的安装
- 指令库的卸载
- 指令库的添加
- 指令库的删除



指令库的安装

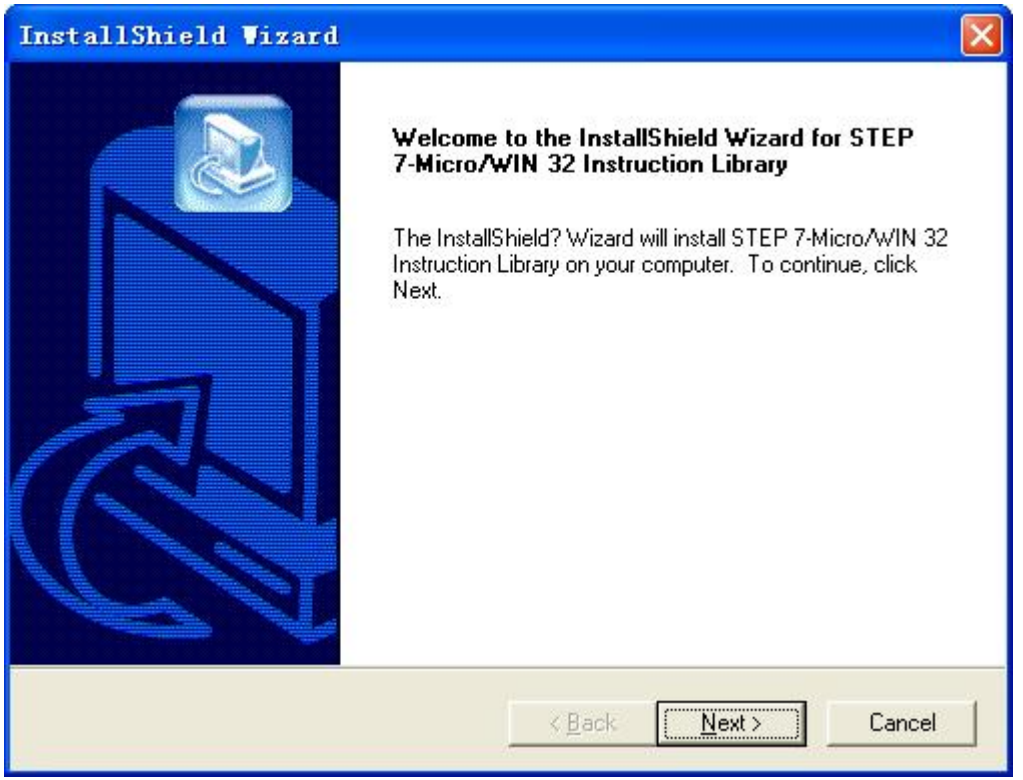
- **Step7-Micro/Win** 指令库光盘可直接从西门子订购，名称为 **STEP 7-Micro/WIN Add-On: Instruction Library (STEP 7-Micro/WIN** 附件：指令库)，订购编号为 **6ES7 830 2BC00 0YX0**。**Step7-Micro/Win** 指令库光盘内包含了 **USS** 协议指令库和 **Modbus** 指令库，安装后可在 **Step7-Micro/Win** 中调用。

- 使用西门子指令库光盘安装指令库（本文以 **Step7-Micro/Win V3.2**版的库安装文件为例）的步骤如下：

- （**1**）单击光盘的 **Inst_Library_V11** 下 **“Setup.exe”** 文件，在弹出的安装语言选择框中选择安装语言，单击 **“**确定**”** 按钮。

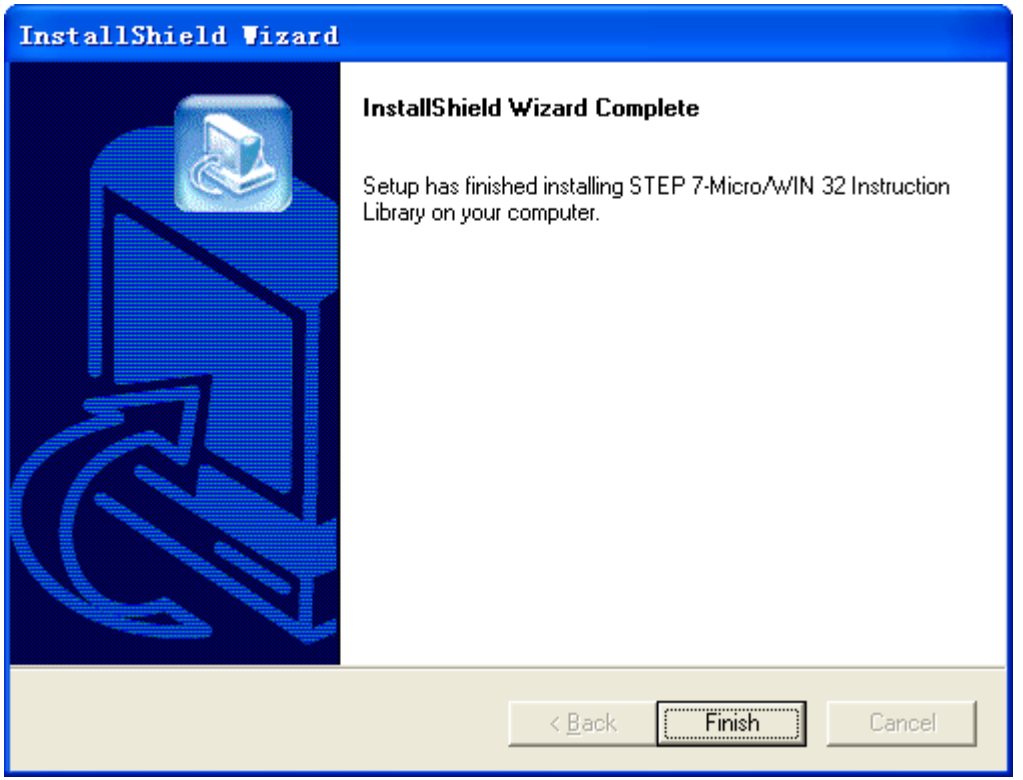
指令库的安装

- （2）在欢迎页面中，单击 **“Next”** 按钮开始安装。



指令库的安装

- （3）安装完成后，单击 “Finish” 按钮结束安装，关闭安装程序。

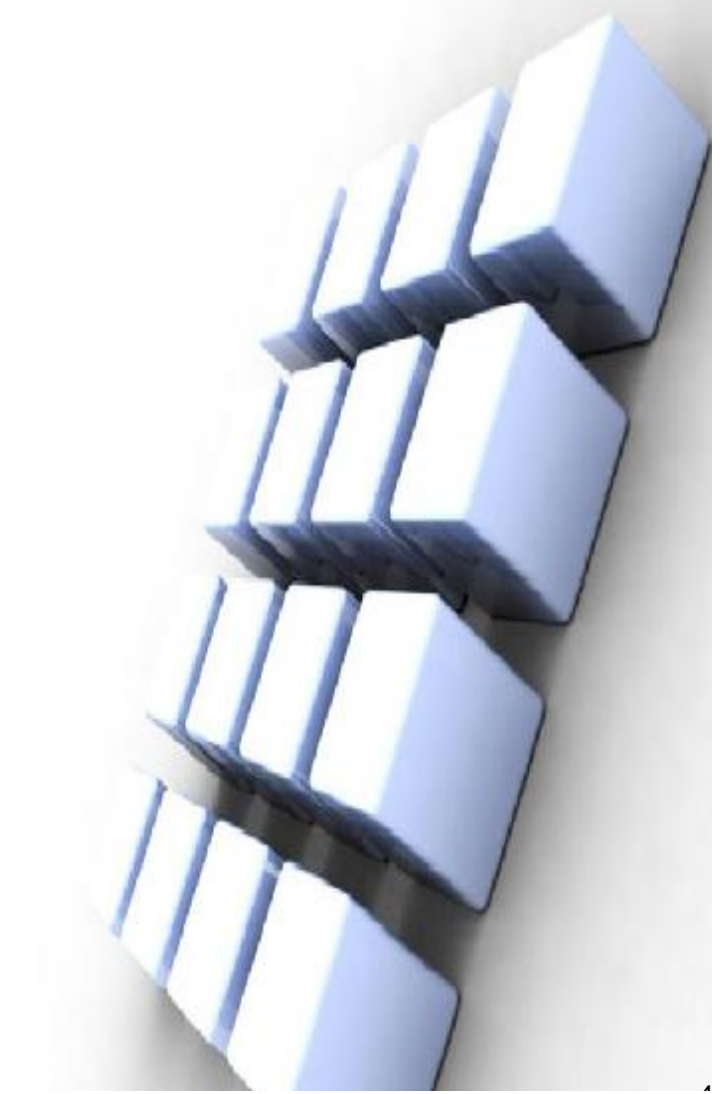


指令库的安装

- （4）安装完成后，启动 **Step7-Micro/Win**，在“指令树”>“库”项下可以发现多出了 **USS** 协议库和 **Modbus** 协议库。

Micro/Win 指令库的管理

- 指令库的安装
- 指令库的卸载
- 指令库的添加
- 指令库的删除

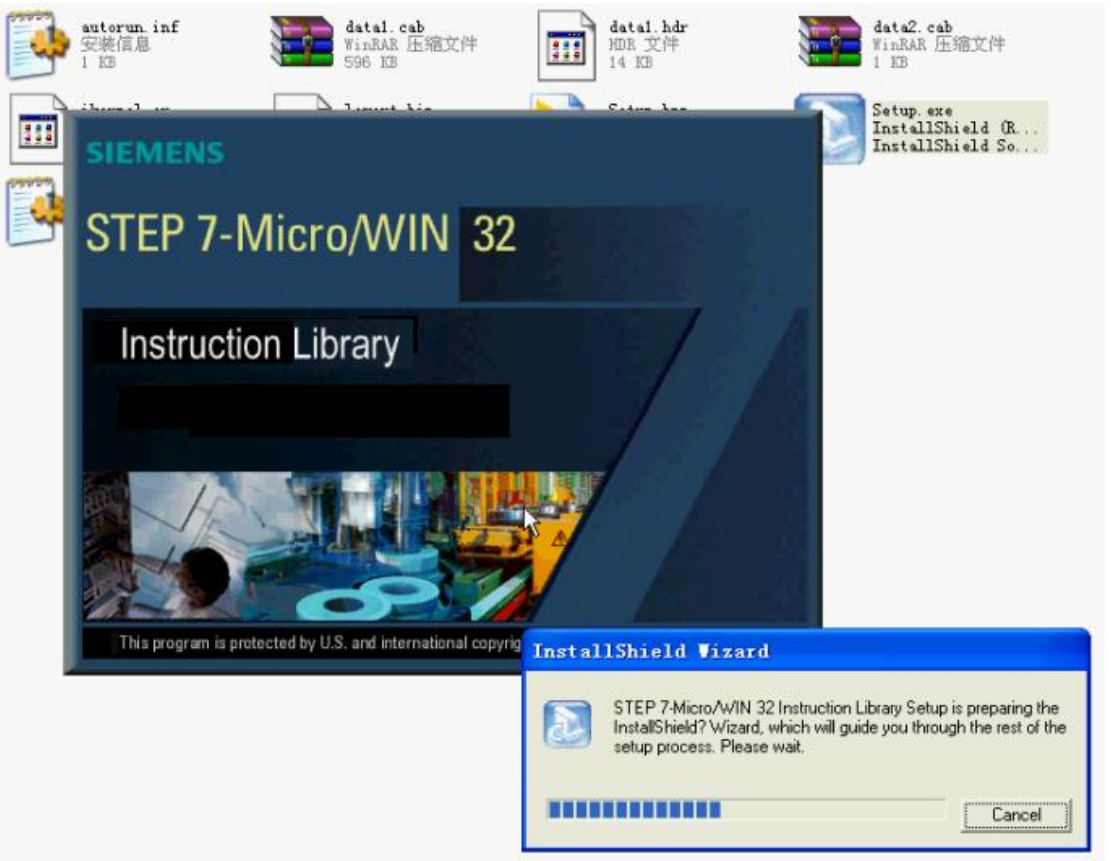


指令库的卸载

- 卸载西门子指令库光盘安装的指令库，按以下步骤即可：
- （1）单击光盘的 **Inst_Library_V11** 下 **“Setup.exe”** 文件，在弹出的语言选择框中选择语言，单击“确定”按钮进入下一步。

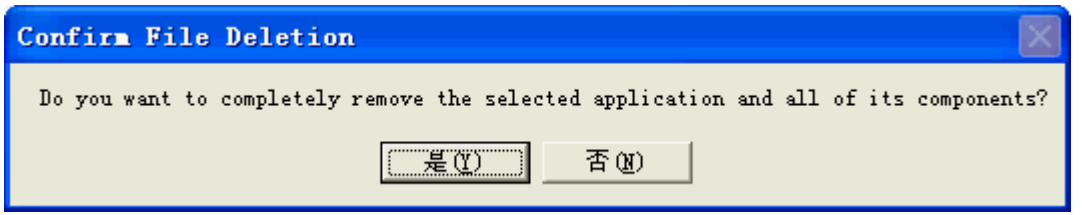
指令库的卸载

- （2）系统将自动检测指令库的安装信息。



指令库的卸载

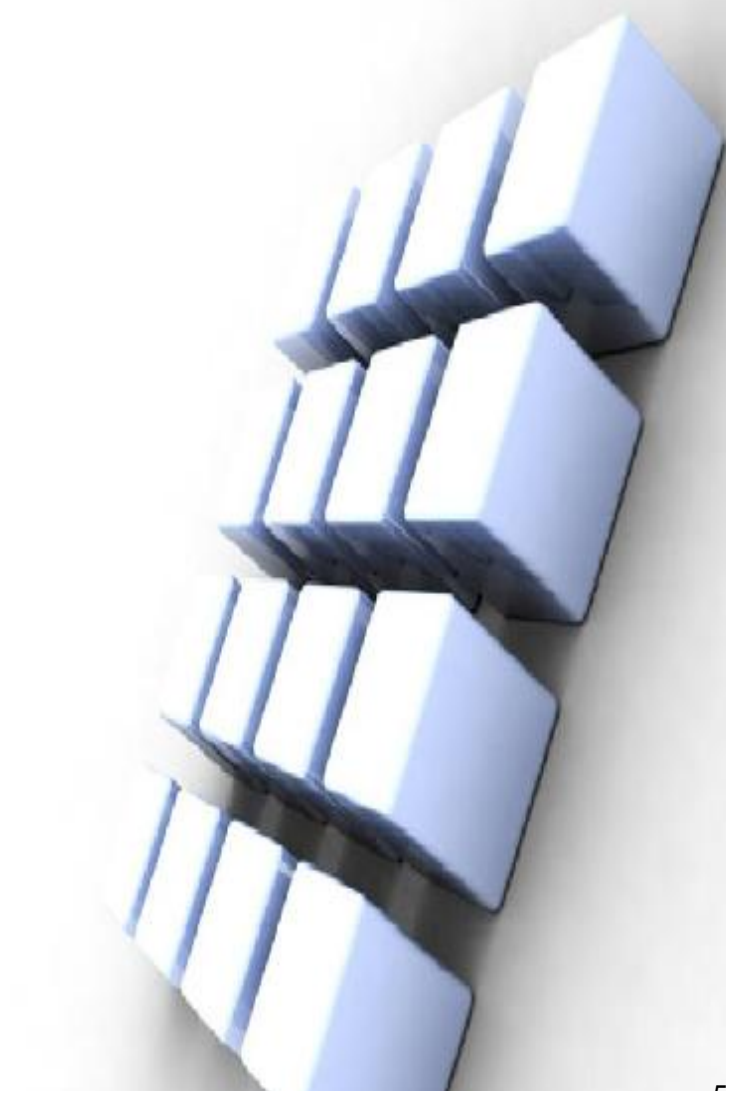
- （3）在“确认卸载”对话框中，若想放弃卸载，可单击“否”按钮退出卸载程序。单击“是”按钮启动卸载程序。



- （4）卸载完成后，单击“Finish”按钮结束卸载，关闭卸载程序。

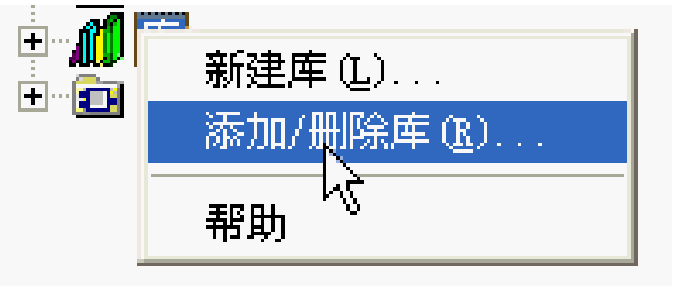
Micro/Win 指令库的管理

- 指令库的安装
- 指令库的卸载
- 指令库的添加
- 指令库的删除



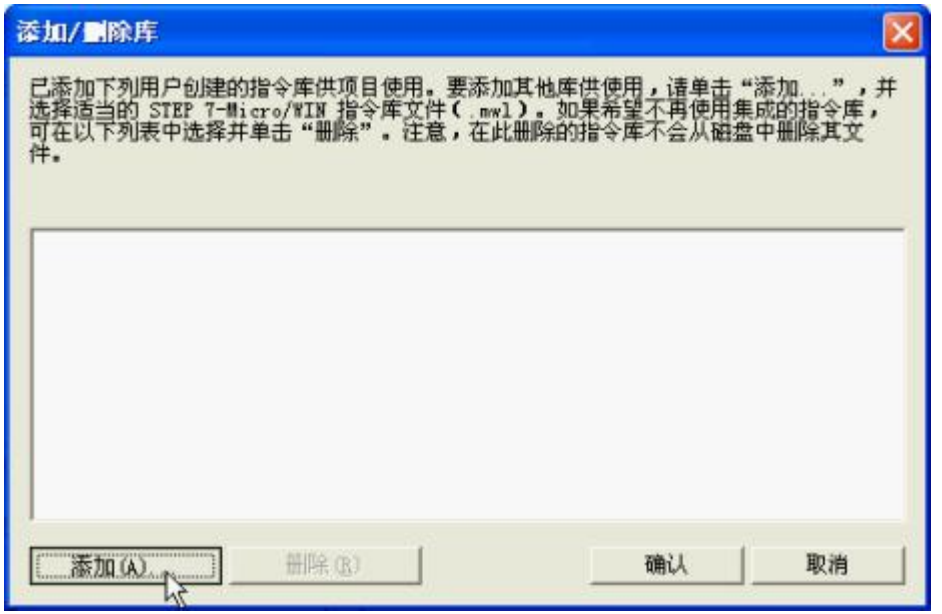
指令库的添加

- 若有*.mwl格式的指令库文件，也可手动添加指令库。手动添加指令库的步骤如下：
- （1）将指令库文件拷贝到“Step7-Micro/Win V4.0\lib”目录下。
- （2）在“指令树”>“库”项处单击右键菜单，执行菜单命令“添加/删除库”。



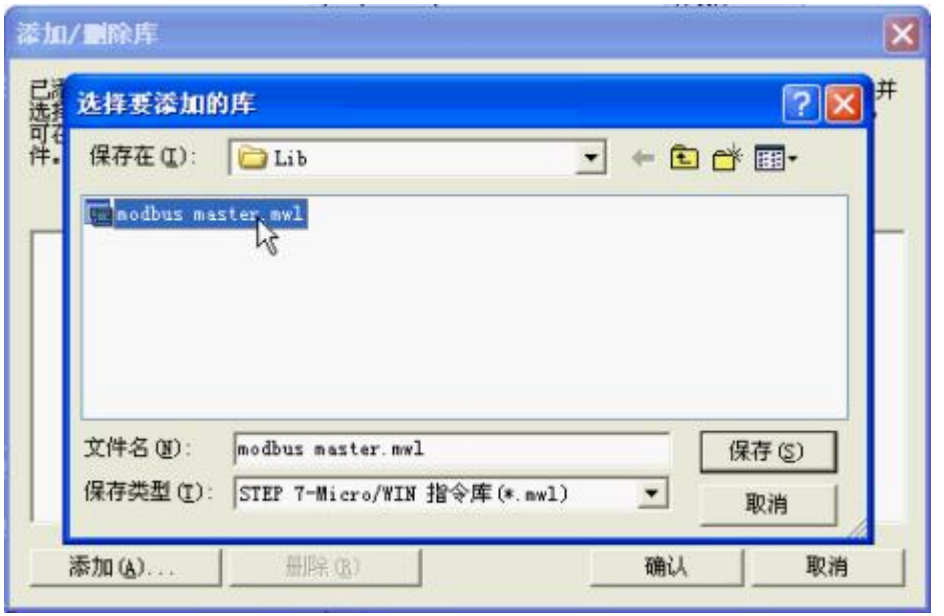
指令库的添加

- （3）在弹出的“添加/删除库”对话框中，单击“添加”按钮。



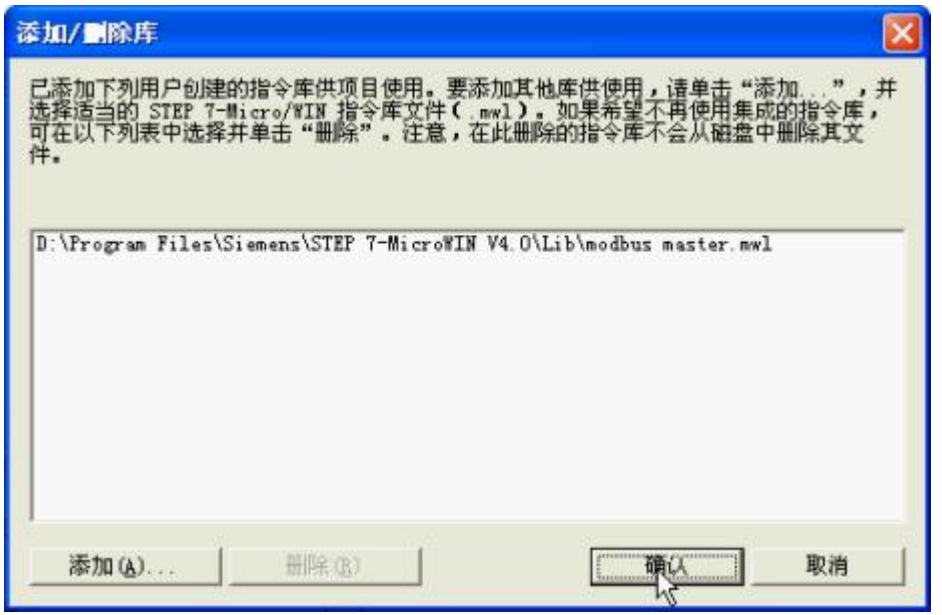
指令库的添加

- （4）双击选择要添加的库文件。



指令库的添加

- （5）单击“确认”按钮，确认添加刚才的选择库文件。



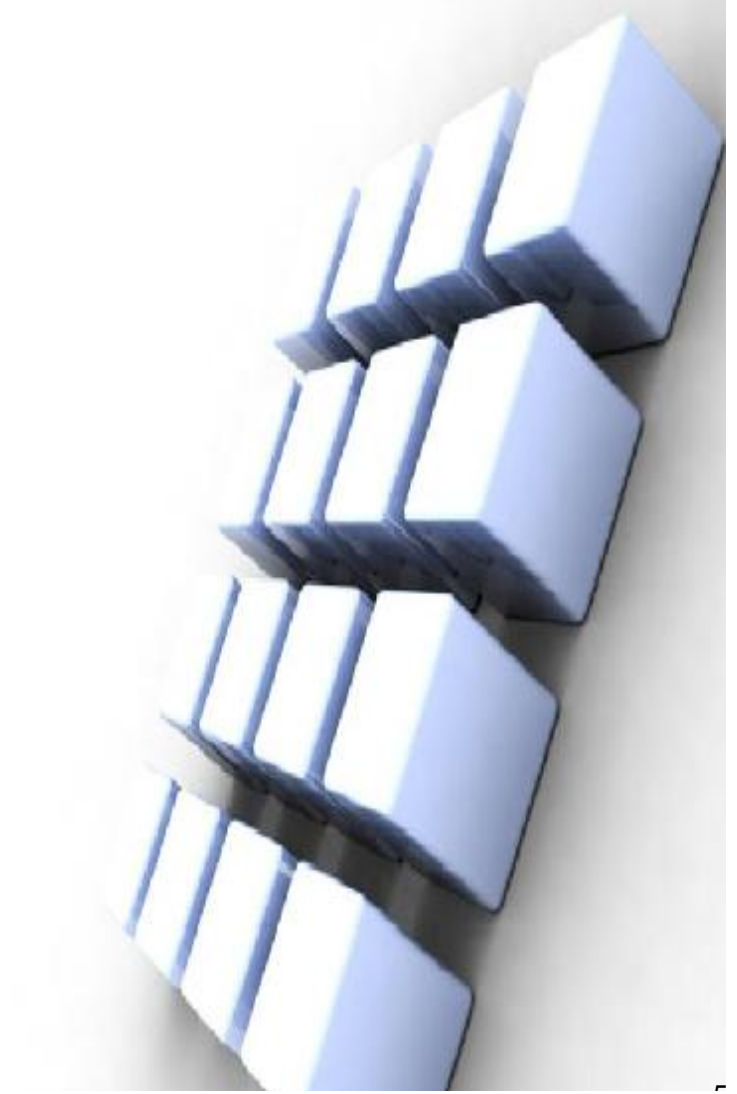
- （6）添加完毕之后，重新启动 **Micro/Win**，会发现“库”中出现了刚才添加的库。

指令库的添加

- 补充说明：
- （1）指令库文件也可拷贝到其他路径，然而考虑到便于管理，统一拷贝到 **“Step7-Micro/Win V4.0\lib”** 路径下更为合理。
- （2）添加成功后，请勿直接删除或移动指令库文件的位置，否则启动 **Step7-Micro/Win V4.0** 时，将出现找不到库文件的错误提示（如下图所示）。此错误解决方法为恢复该库文件到原始添加位置、或者在 **Step7-Micro/Win V4.0** 删除该库文件。

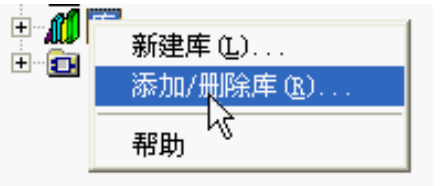
Micro/Win 指令库的管理

- 指令库的安装
- 指令库的卸载
- 指令库的添加
- 指令库的删除



指令库的删除

- 手动删除指令库的步骤如下：
- （1）在“指令树”>“库”项处单击右键菜单，执行菜单命令“添加/删除库”。



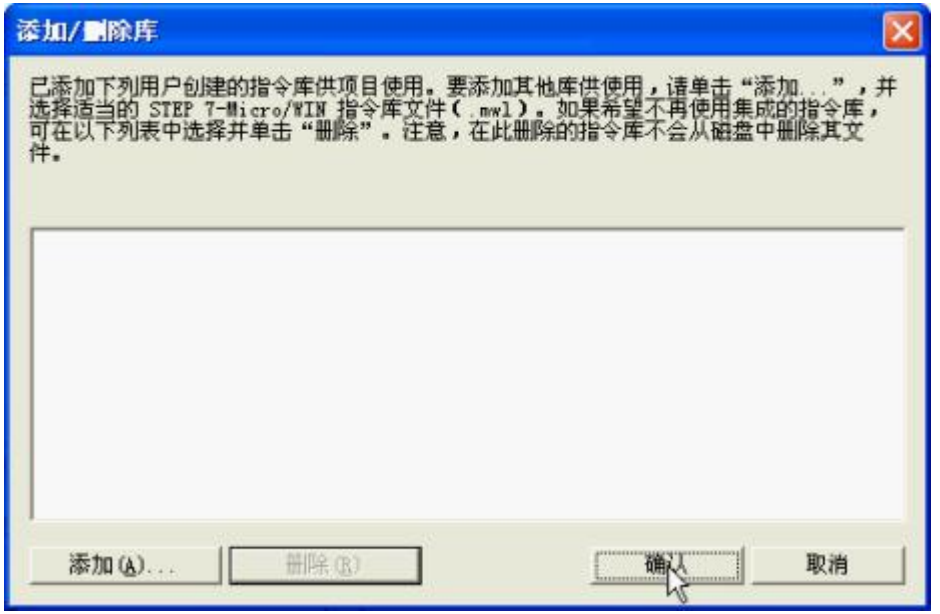
- （2）在弹出的“添加/删除库”对话框中，选中欲删除的库程序所对应的库文件，单击“删除”按钮。

指令库的删除

- （3）在提示框中，单击“删除”按钮确认删除。

指令库的删除

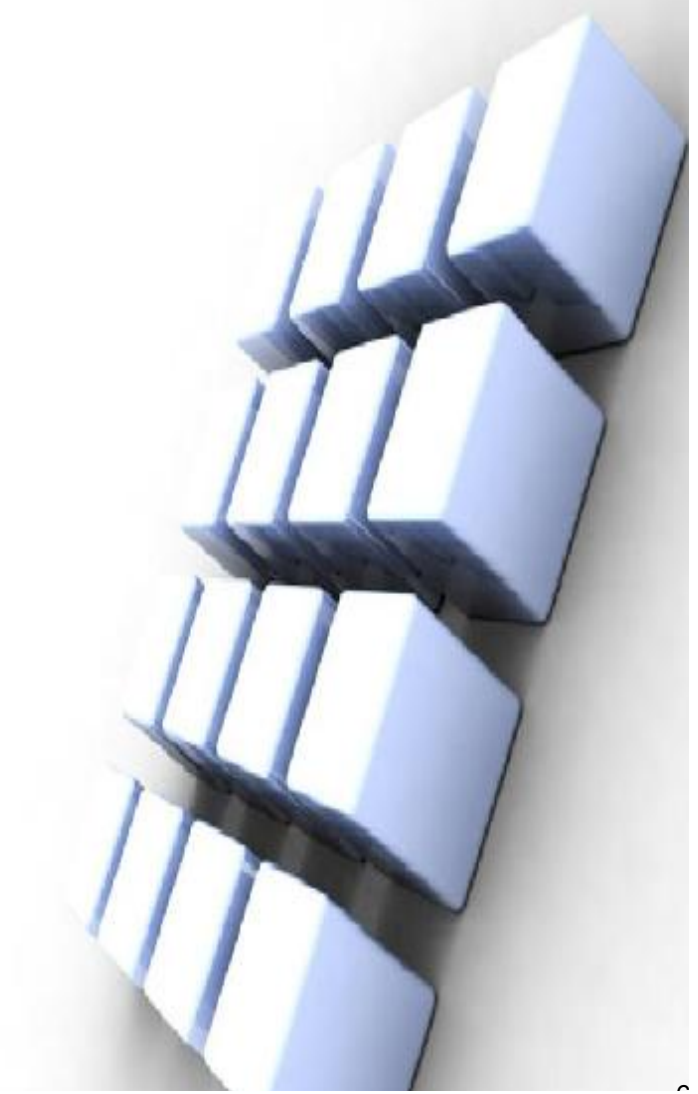
- （4）单击“确认”按钮关闭“添加/删除库”对话框。



- 注意：按此方法“手动删除指令库”，并不会删除计算机上删除该文件，因此以后还可根据需要重新添加。

S7-200 Modbus 通信

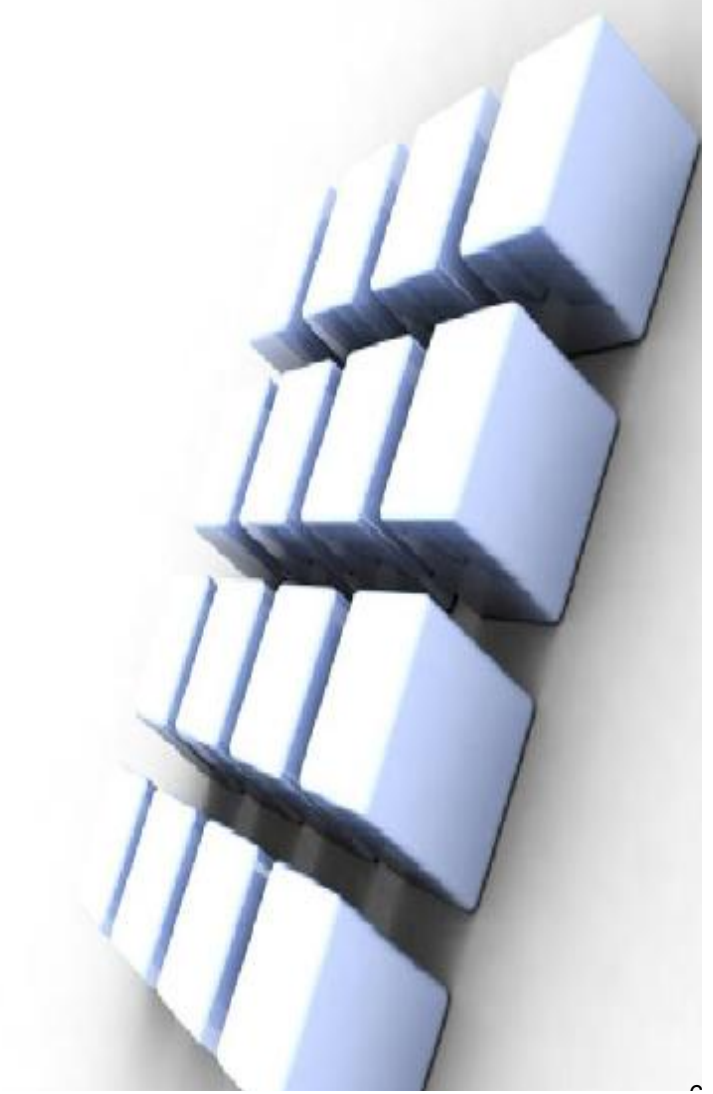
- 概述
  - 定义
  - **Modbus** 报文传输格式
  - **Modbus** 主站协议指令
  - **Modbus** 从站协议指令
- **Micro/Win** 指令库的管理
  - 指令库的安装
  - 指令库的卸载
  - 指令库的添加
  - 指令库的删除
- 编程示例
  - 功能要求
  - 实现步骤



编程示例

- 功能要求

- 实现步骤
  - **Modbus** 从站组态说明
  - **Modbus** 主站组态说明
  - 通信测试



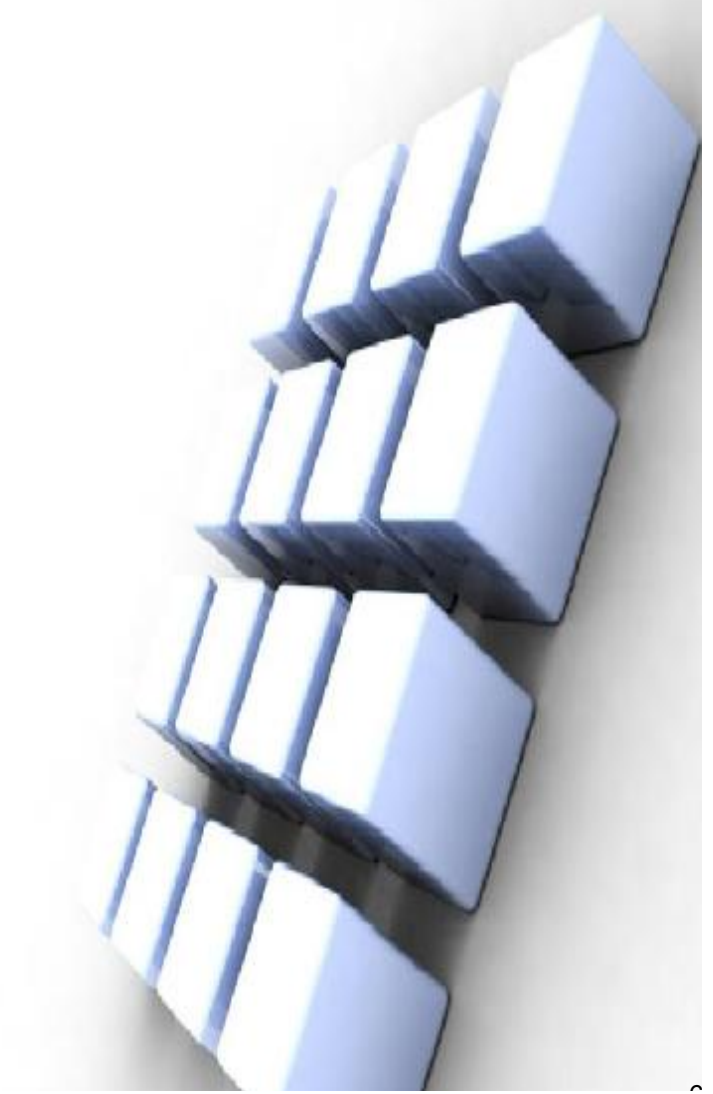
功能要求

- 将一台 S7-200 CPU224XP 组态为 Modbus 主站，当主站 I0.3 为 ON 时，读取另一台作为 Modbus 从站的 S7-200 CPU224XP 的 I0.0~I0.7 的数值。
- 硬件需求：
- PC 机、2台 S7-200 CPU 224XP、RS 232 电缆（推荐采用西门子 S7-200 串口编程电缆）
- 示例的简要实现步骤如下：
- （1）编写作为 Modbus 从站的 S7-200 CPU 的 PLC 程序，将程序下载到从站 PLC 中。
- （2）编写作为 Modbus 主站的 S7-200 CPU 的 PLC 程序，将程序下载到主站PLC中。
- （3）用串口电缆连接 Modbus 主从站，在Step-7 Micro/Win 的状态表中观察 Modbus 主站保持寄存器中的数值，并与实际数值对比。

编程示例

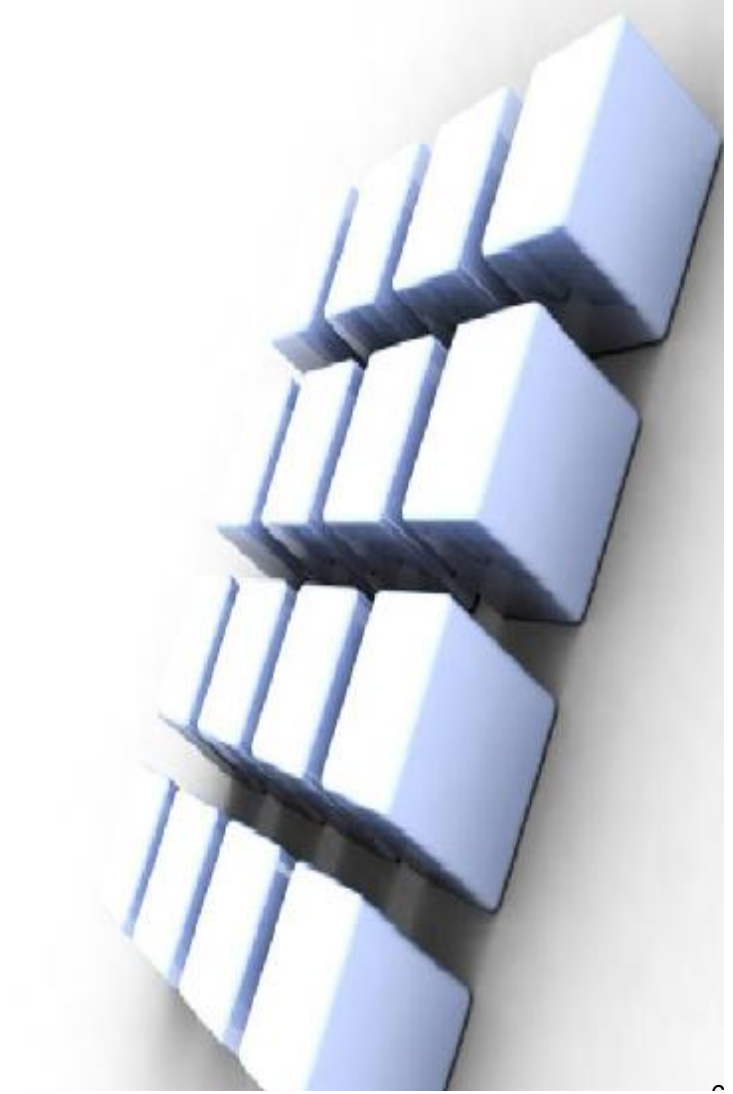
- 功能要求

- 实现步骤
  - **Modbus** 从站组态说明
  - **Modbus** 主站组态说明
  - 通信测试



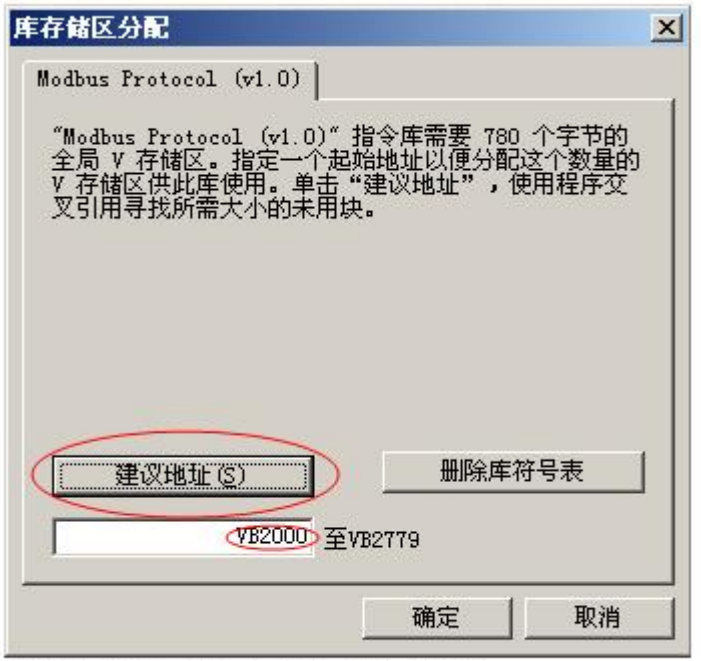
实现步骤

- **Modbus** 从站组态说明
- **Modbus** 主站组态说明
- 通信测试



Modbus 从站组态说明

- **1** 分配库存储区
- 利用指令库编程前首先应为其分配存储区，否则 **Step7-Micro/Win** 编译时会报错。具体方法如下：
- （**1**）执行 **Step7-Micro/Win** 菜单命令“文件”>“库存储区”，打开“库存储区分配”对话框。



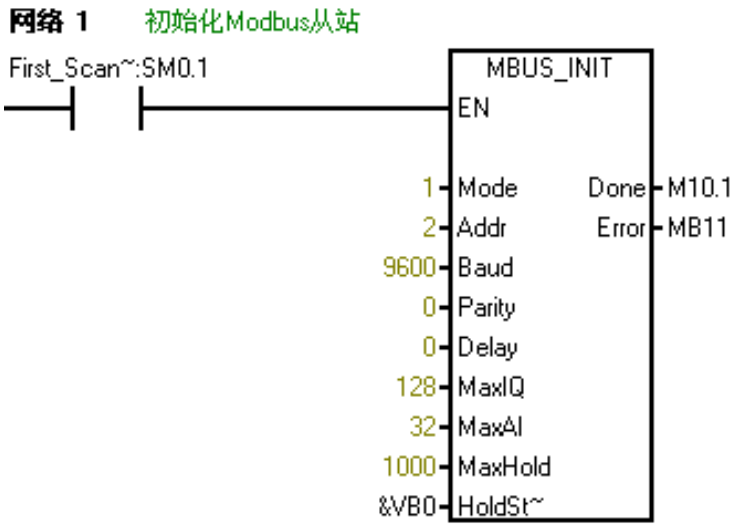
地址文本框：输入库存储区的起始地址。

Modbus 从站组态说明

- （2）在“库存储区分配”对话框中输入库存储区的起始地址，注意避免该地址和程序中已经采用或准备采用的其它地址重合。
- （3）点击“建议地址”按钮，系统将自动计算存储区的截止地址。
- （4）点击“确定”按钮确认分配，关闭对话框。

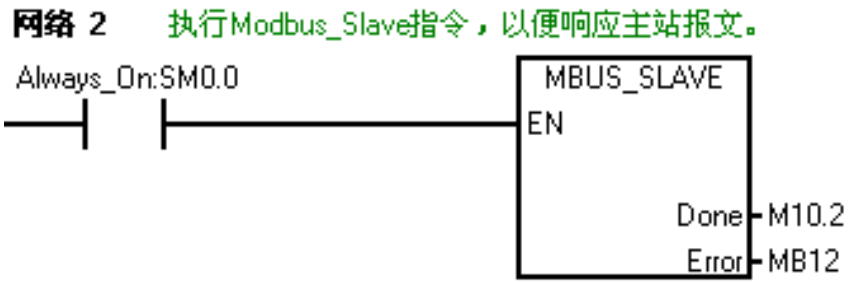
Modbus 从站组态说明

- **2** 从站组态说明
- 根据示例要求，本从站要响应主站报文，故只需编写主程序。主程序由以下两个网络构成，程序说明如下。
- （1）网络1用于初始化 **Modbus** 从站，即是将从站地址设为1，将端口0的波特率设为9600、无校验、无延迟，允许存取所有的I、Q 和 AI 数值，保存寄存器的存储空间为从 **VB0** 开始的1000个字(2000个字节)。



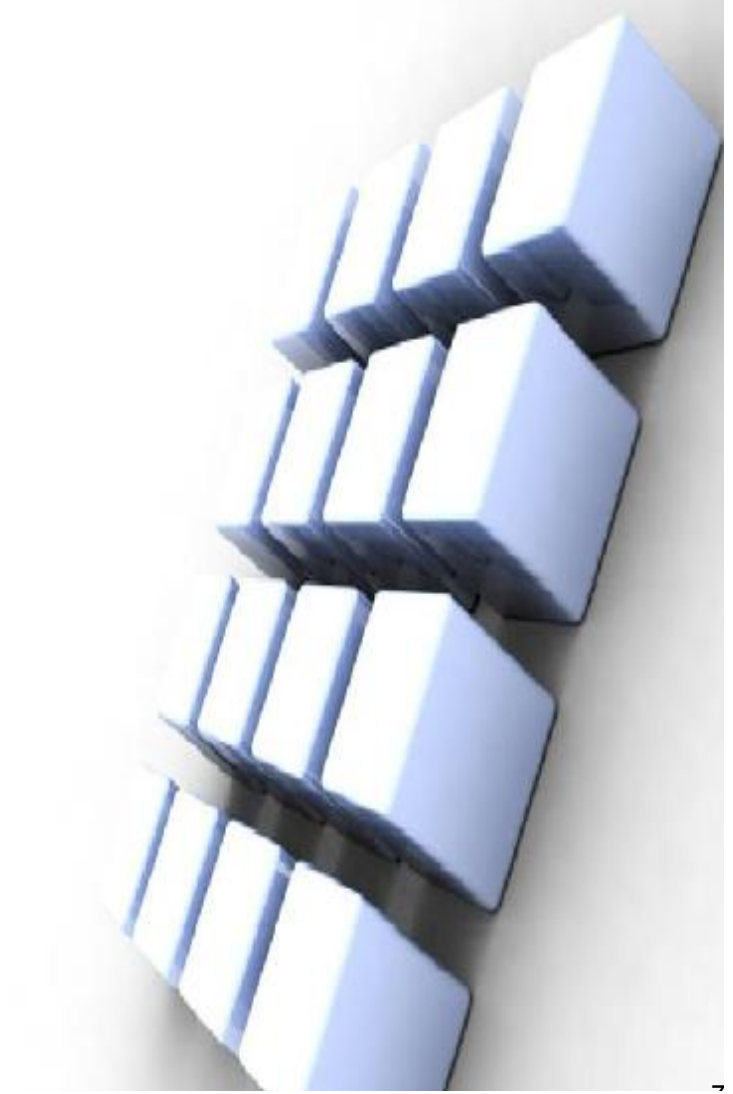
Modbus 从站组态说明

- （2）网络2用于在每次扫描时执行Modbus_Slave指令。



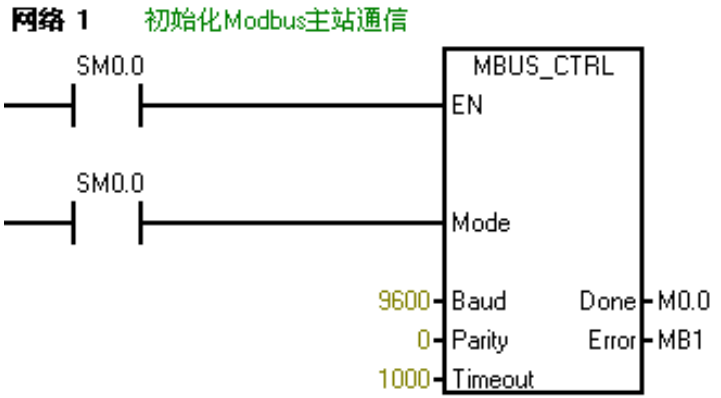
实现步骤

- **Modbus** 从站组态说明
- **Modbus** 主站组态说明
- 通信测试



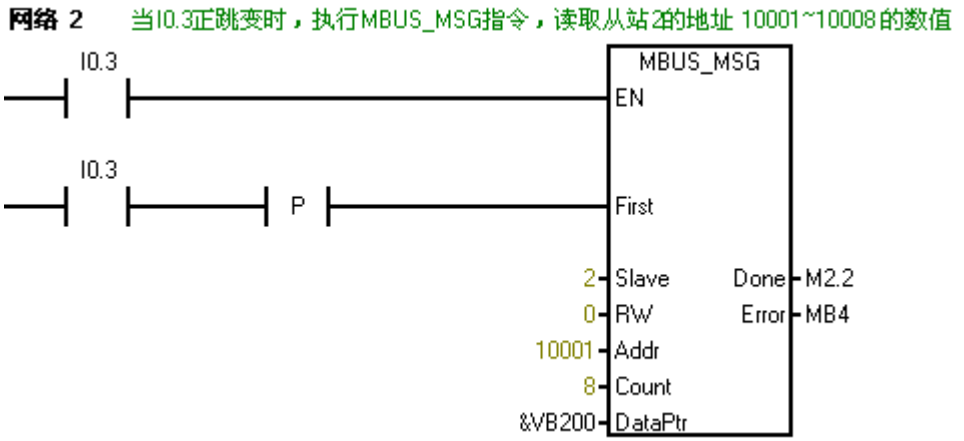
Modbus 主站组态说明

- 调用 **Modbus** 主站指令编程前也应分配库存储区，与从站编程类似，不再赘述。
- **Modbus**主站指令也只需编写主程序，主程序也由两个网络构成。
- （1） 网络1用于每次扫描时调用 **MBUS_CTRL** 指令来初始化和监视 **Modbus** 主站设备。**Modbus** 主设备设置为**9600**波特，无奇偶校验，允许从站**1000**毫秒**(1**秒**)**的应答时间。



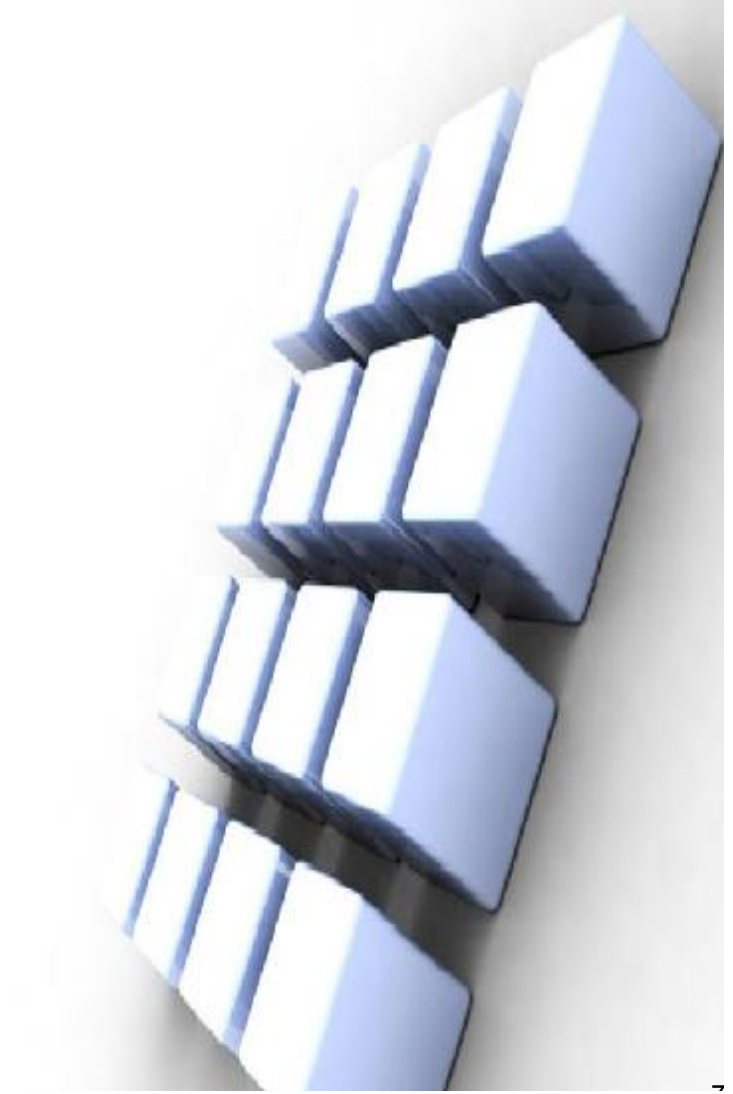
Modbus 主站组态说明

- （2）网络2实现在 **I0.3** 正跳变时执行 **MBUS_MSG** 指令读取从站2的地址 **10001~10008** 的数值。保持寄存器存储区从 **VB200** 开始，长8个字。根据 **Modbus** 从站寻址规约，**10001~10008** 即 **S7-200 PLC** 中**I0.1~I0.7**的 **Modbus** 地址值。



实现步骤

- **Modbus** 从站组态说明
- **Modbus** 主站组态说明
- 通信测试



通信测试

- 通信测试的步骤如下：
- （1）用串口电缆连接主从站 PLC 的端口0。
- （2）将主从站 PLC 设置为 Run 状态。
- （3）设置从站 I0.0~I0.7。
- （4）将主站的 I0.3 设为ON，利用 Step7-Micro/Win 状态表监测主站保持寄存器的数值。

	地址	格式	当前值	新值
1	VB200	十六进制	16#FF	
2	VB201	无符号	0	
3	VB202	无符号	0	
4	VB203	无符号	0	
5	VB204	无符号	0	

- 从图中可以看出VB200存储的即是 I0.0~I0.7 的数值，此时均为 ON 状态，与这些输入点的实际状态一致。

通信测试

- 补充说明：
- （1）如果 **Modbus RTU** 通信失败，可分别从主从站的报文入手。
- （2）测试 **Modbus** 从站通信是否正常，可利用计算机上的串口通信调试软件向从站发送请求帧，查看是否能接受到正确的响应帧。
- （3）测试 **Modbus** 主站通信是否正常，可由主站向计算机串口发送请求帧，在计算机上用串口通信调试软件查看请求帧是否正常。

Thank you for your attention!

End of presentation!