

A man in a light blue shirt is seen from the side, holding a tablet. The background is a blurred industrial factory floor. Overlaid on the scene are various futuristic digital graphics: a '24/7' icon with a circular arrow, a 'NEWS' section with a profile icon, a 'Home' button, and a network diagram with nodes and binary code. The overall color scheme is dominated by blues and teals.

**SIEMENS**

*Ingenuity for life*

# SMART MODBUS 轮询的应用库

STEP 7-Micro/WIN SMART V2.7

## 法律信息

### 应用实例的使用

应用实例说明了通过文本、图形和/或软件模块形式的几个组件的交互来实现自动化任务的解决方案。本应用程序示例是由西门子公司和/或西门子公司(以下简称“西门子”)的子公司提供的免费服务。它们是非约束性的,并且不声明关于配置和设备的完整性或功能性。应用程序示例仅提供典型任务的帮助;它们并不构成客户特定的解决方案。您有责任按照适用的法规,对产品的正确和安全操作负责,并必须检查相应的应用示例的功能,并为您的系统定制它。

西门子授予您非排他性、不可再授权和不可转让的权利,让经过技术培训的人员使用应用示例。对应用程序示例的任何更改都由您负责。与第三方共享应用示例,或复制应用示例或摘录,仅允许与您自己的产品结合使用。该应用实例无须接受收费产品的惯常测试和品质检验;它们可能有功能和性能缺陷以及错误。您有责任使用它们,使任何可能发生的故障不会导致财产损失或人身伤害。

### 免责声明

由于任何法律原因, Siemens 不承担任何责任,包括但不限于对应用示例的可用性、可用性、完整性和不存在缺陷以及相关信息、配置和性能数据以及由此造成的任何损害承担责任。这个不适用强制责任的情况下,例如在德国的产品责任法,或意图的情况下,重大过失,或有罪的生命损失,人身伤害或损坏健康,不符合担保,欺骗性的非披露缺陷或有罪的违反合同义务。但因违反重大合同义务而提出的损害赔偿要求应限于协议类型的典型可预见损害,但因故意或重大过失或基于生命损失、身体伤害或健康损害而产生的责任除外。上述规定并不意味着对您不利的举证责任的任何改变。对于第三方在此方面的现有或未来索赔,您应向西门子作出赔偿,除非西门子负有强制责任。

通过使用应用示例,您承认西门子对上述责任条款之外的任何损害不承担责任。

### 其他信息

西门子保留随时更改应用示例的权利,无需另行通知。如果应用实例中的建议与其他西门子出版物(如目录)之间存在差异,则应优先考虑其他文件的内容。

### 安全信息

西门子提供具有工业安全功能的产品和解决方案,支持工厂、系统、机器和网络的安全运行。

为了保护工厂、系统、机器和网络免受网络威胁,有必要实施——并持续维护——一个整体的、最先进的工业安全概念。西门子的产品和解决方案构成了这一概念的一个元素。

客户有责任防止对其工厂、系统、机器和网络的未经授权的访问。

这些系统、机器和组件只应在必要的情况下连接到企业网络或 Internet,并且只有在适当的安全措施(例如防火墙和/或网络分割)到位的情况下才应连接到这种连接。有关可能实施的工业保安措施的其他资料,请浏览 <https://www.siemens.com/industrialsecurity>。

西门子的产品和解决方案经过不断的发展,使其更加安全。西门子强烈建议,一旦产品更新可用,就立即应用产品更新,并使用最新的产品版本。使用不再受支持的产品版本以及未能应用最新更新可能会增加客户遭受网络威胁的风险。

了解产品更新,请订阅西门子工业安全 RSS Feed: <https://www.siemens.com/industrialsecurity>。

# 目录

<b>1</b>	<b>应用概述</b> .....	<b>4</b>
1.1	通用描述 .....	4
1.2	硬件及软件需求 .....	4
<b>2</b>	<b>程序库</b> .....	<b>5</b>
<b>3</b>	<b>例程说明</b> .....	<b>9</b>
3.1	Modbus 初始化 .....	9
3.2	Modbus 轮询序列 .....	9
3.3	Modbus 读控制 .....	10
3.4	Modbus 写控制 .....	11
3.5	Modbus 设定值改变功能块 .....	11
<b>4</b>	<b>更新日志</b> .....	<b>13</b>

# 1 应用概述

## 1.1 通用描述

Modbus-RTU 协议是一种开放的串行通信协议，在不同的行业中都有非常广泛的应用，因为其是串行通信，所以其报文的传送需要按照串行队列来发送，而具体到 S7-200 SMART 的 Modbus-RTU 主站通信的编程中就需要考虑针对不同地址和不同站的轮询的问题，为了提高轮询的效率和轮询程序的通用性，本文推出了 Modbus 轮询的应用库，可以简化用户轮询编程，也可以实现根据需要写入功能，更可以实现仅当设定值改变时的参数值写入功能。

## 1.2 硬件及软件需求

### 本应用软硬件的需求

为了使得本应用案例成功运行，必须满足以下硬件和软件需求。

#### 硬件

S7-200 SMART CPU 控制器:

- SIMATIC S7-200 SMART 产品家族

#### 软件

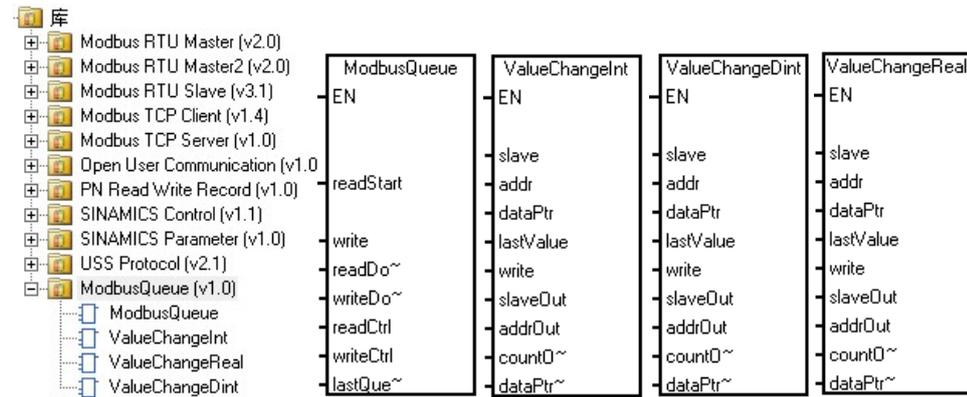
- STEP 7-Micro/WIN SMART V2.7

## 2 程序库

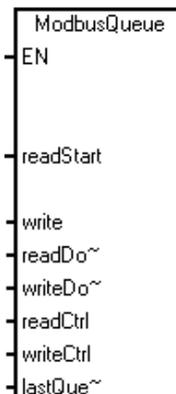
### 简要说明

该程序库实现了 Modbus 轮询队列的产生，可以产生两种独立的队列，读队列和写队列，并可以根据需要实时操作写队列，也可以实现当设定值改变时实时写入的功能。

### 程序库



### 轮询队列控制块



### 程序块引脚

参数 & 类型	数据类型	描述
EN	BOOL	程序块使能
readStart	IN	启动读轮询队列，上升沿触发
write	IN/OUT	启动写轮询队列，使用置位信号
readDone	IN/OUT	读指令 MSG 完成位
writeDone	IN/OUT	写指令 MSG 完成位
readCtrl	IN/OUT	读指令 MSG 控制字，每一位控制一个站点，最多可以控制 32 个站
WriteCtrl	IN/OUT	写指令 MSG 控制字，每一位控制一个站点，最多可以控制 16 个站写

## 2 程序库

lastQueue	IN/OUT	DWORD	储存写入时的读取队列，写入完成后返回队列
-----------	--------	-------	----------------------

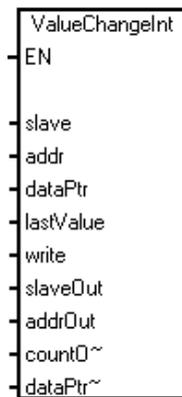
### 使用说明

该子程序主要用于产生 Modbus 轮询队列，会产生读队列 readCtrl 控制字和写队列 writeCtrl 控制字，正常执行时，仅执行读队列 readCtrl 控制字运算，当有写操作时，中断读操作，优先执行写操作队列 writeCtrl 控制字运算，仅当写操作完成后，会返回到之前读队列的中断点开始继续执行读队列。

### 注意事项

该子程序需要配合 Modbus 通信库使用。

### 整数设定值改变时输出功能块



### 程序块引脚

参数 & 类型	数据类型	描述
EN	BOOL	程序块使能
slave	IN BYTE	从站地址（参考 MBUS_MSG 指令介绍）
addr	IN DWORD	参数地址（参考 MBUS_MSG 指令介绍）
dataPtr	IN DWORD	数据指针（参考 MBUS_MSG 指令介绍）
lastValue	IN/OUT WORD	存储上一个周期的参数值
write	IN/OUT BOOL	当参数改变时置位该位输出
slaveOut	IN/OUT BYTE	参数改变时输出从站地址（对应 MSG 输入）
addrOut	IN/OUT DWORD	参数改变时输出参数地址（对应 MSG 输入）
countOut	IN/OUT WORD	参数改变时输出参数数量（对应 MSG 输入）
dataPtrOut	IN/OUT DWORD	参数改变时输出数据指针（对应 MSG 输入）

### 使用说明

该子程序可以实现当输入整数值发生改变时，把对应地址传送到输出地址，并输出标志位 write。

### 注意事项

该子程序需要配合 Modbus 通信库和 ModbusQueue 库使用。

双整数设定值改变时输出功能块



程序块引脚

参数 & 类型		数据类型	描述
EN		BOOL	程序块使能
slave	IN	BYTE	从站地址 (参考 MBUS_MSG 指令介绍)
addr	IN	DWORD	参数地址 (参考 MBUS_MSG 指令介绍)
dataPtr	IN	DWORD	数据指针 (参考 MBUS_MSG 指令介绍)
lastValue	IN/OUT	DWORD	存储上一个周期的参数值
write	IN/OUT	BOOL	当参数改变时置位该位输出
slaveOut	IN/OUT	BYTE	参数改变时输出从站地址 (对应 MSG 输入)
addrOut	IN/OUT	DWORD	参数改变时输出参数地址 (对应 MSG 输入)
countOut	IN/OUT	WORD	参数改变时输出参数数量 (对应 MSG 输入)
dataPtrOut	IN/OUT	DWORD	参数改变时输出数据指针 (对应 MSG 输入)

使用说明

该子程序可以实现当输入双整数值发生改变时，把对应地址传送到输出地址，并输出标志位 write。

注意事项

该子程序需要配合 Modbus 通信库和 ModbusQueue 库使用。

浮点数设定值改变时输出功能块



程序块引脚

参数 & 类型		数据类型	描述
EN		BOOL	程序块使能
slave	IN	BYTE	从站地址 (参考 MBUS_MSG 指令介绍)
addr	IN	DWORD	参数地址 (参考 MBUS_MSG 指令介绍)
dataPtr	IN	DWORD	数据指针 (参考 MBUS_MSG 指令介绍)
lastValue	IN/OUT	REAL	存储上一个周期的参数值
write	IN/OUT	BOOL	当参数改变时置位该位输出
slaveOut	IN/OUT	BYTE	参数改变时输出从站地址 (对应 MSG 输入)
addrOut	IN/OUT	DWORD	参数改变时输出参数地址 (对应 MSG 输入)
countOut	IN/OUT	WORD	参数改变时输出参数数量 (对应 MSG 输入)
dataPtrOut	IN/OUT	DWORD	参数改变时输出数据指针 (对应 MSG 输入)

使用说明

该子程序可以实现当输入浮点数值发生改变时，把对应地址传送到输出地址，并输出标志位 write。

注意事项

该子程序需要配合 Modbus 通信库和 ModbusQueue 库使用。

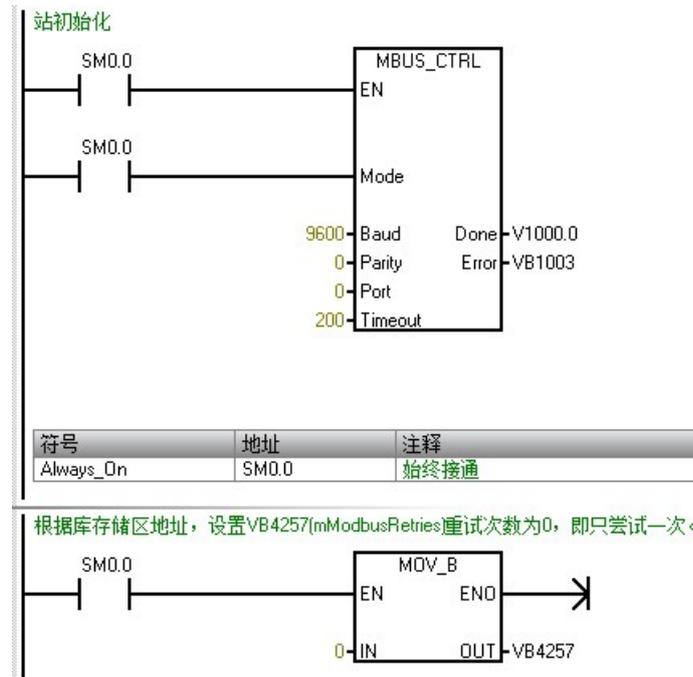
## 3 例程说明

### 3.1 Modbus 初始化

#### 简单说明

初始化 Modbus 程序，本例程中并根据库存储区地址，设置 VB4257 (mModbusRetries) 重试次数为 0，即只尝试一次，加快轮询效率。

#### 程序



### 3.2 Modbus 轮询序列

#### 简单说明

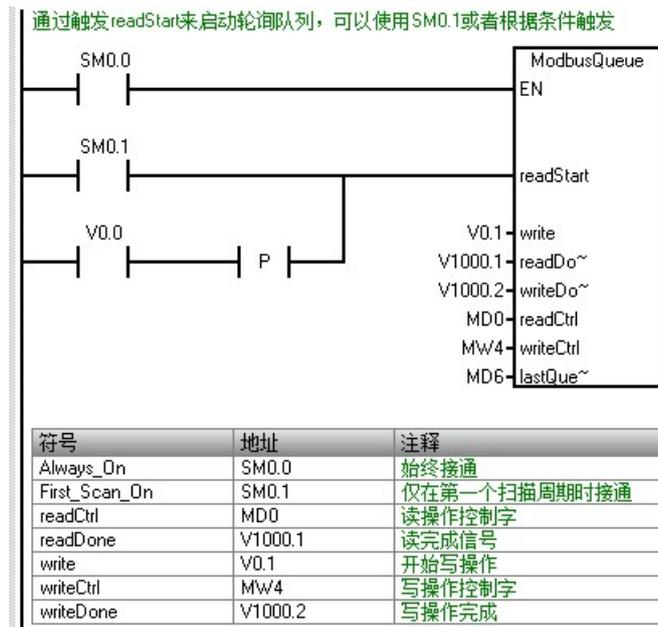
该程序实现了 Modbus 轮询队列的产生，可以产生两种独立的队列，读队列和写队列，并可以根据需要实时操作写队列，也可以实现当设定值改变时实时写入的功能。

本例程中，读队列的控制字为 MD0，写队列的控制字为 MW4。

MD0 每一个位可以控制一个站点读指令 MSG，最多可以控制 32 个 MSG 指令。

MW0 每一个位可以控制一个站点的写指令 MSG，最多可以控制 32 个 MSG 写指令

程序



### 3.3 Modbus 读控制

简单说明

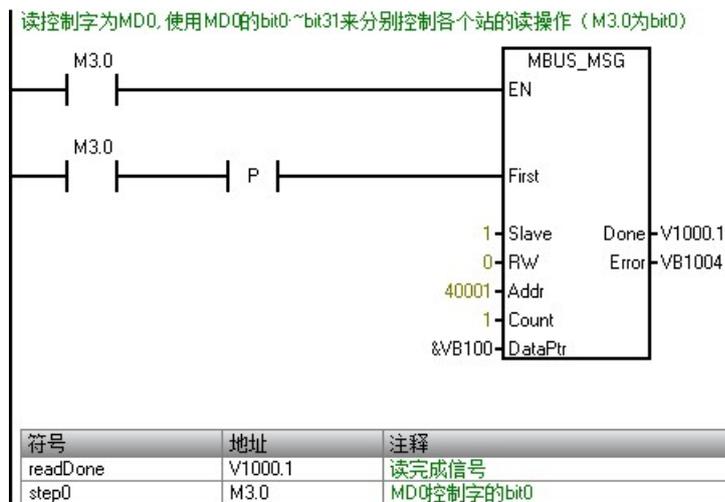
本例程中读控制由读控制字 MD0 控制，其中

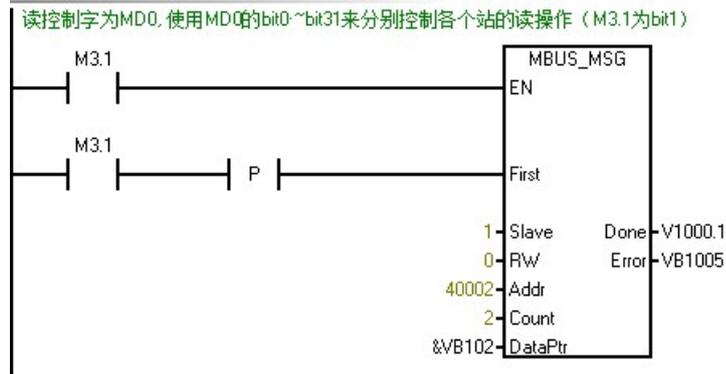
M3.0~3.7 为 MD0 的 bit0~bit7；M2.0~2.7 为 MD0 的 bit8~bit15

M1.0~1.7 为 MD0 的 bit16~bit23；M0.0~0.7 为 MD0 的 bit24~bit31

用户可以根据现场实际需要来灵活增加或者减少读指令个数，每个读指令的控制由 MD0 对应的位来控制即可。

程序



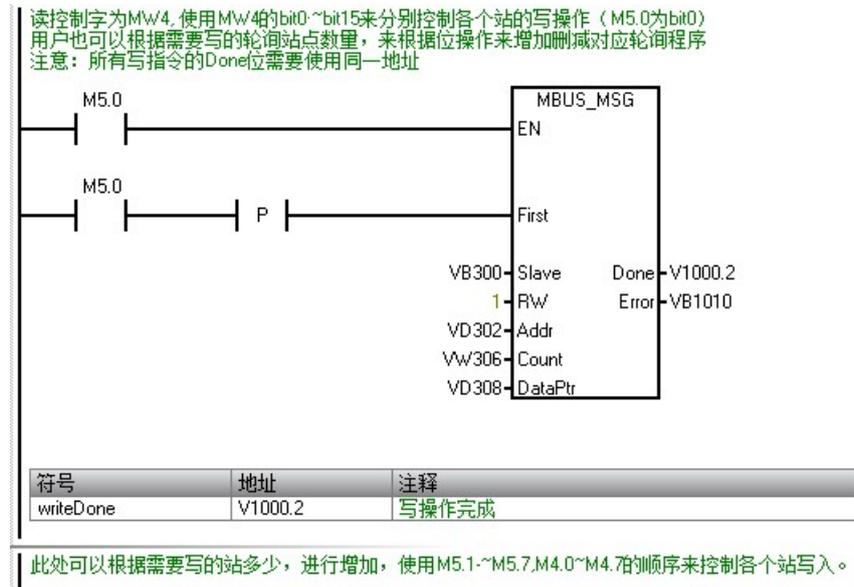


### 3.4 Modbus 写控制

#### 简单说明

本例程中写控制由写控制字 MW4 来控制，其中 M5.0~5.7 为 MW4 的 bit0~bit7；M4.0~4.7 为 MW4 的 bit8~bit15；用户可以根据现场实际需要来灵活增加或者减少写指令个数，每个写指令的控制由 MW4 对应的位来控制即可。

#### 程序



### 3.5 Modbus 设定值改变功能块

#### 简单说明

用户可以根据设定值的类型 (INT/DINT/REAL) 来选择合适的指令库来判断设定值是否变化，若发生变化，则传送对应参数到 Modbus 写指令，并启动写标志位。用户可以根据需要修改的参数个数来灵活调用对应指令库。

程序

当设定值改变时（设定值为INT），把对应数据地址发送到写操作指令的地址，同时发出写操作write

符号	地址	注释
Always_On	SM0.0	始终接通
write	V0.1	开始写操作

当设定值改变时（设定值为REAL），把对应数据地址发送到写操作指令的地址，同时发出写操作write

符号	地址	注释
Always_On	SM0.0	始终接通
write	V0.1	开始写操作

## 4 更新日志

版本& 日期	更新描述
V1.0.0 5/2023	