



中华人民共和国国家标准化指导性技术文件

GB/Z 19582.1—2004

基于 Modbus 协议的工业自动化网络规范 第 1 部分:Modbus 应用协议

Modbus industrial automation network specification—
Part 1:Modbus application protocol

2004-09-21 发布

2005-03-01 实施

中华人民共和国国家质量监督检验检疫总局
中国国家标准化管理委员会

发布

中 华 人 民 共 和 国
国家标准化指导性技术文件
基于 Modbus 协议的工业自动化网络规范
第 1 部分:Modbus 应用协议
GB/Z 19582.1--2004

*

中国标准出版社出版发行
北京复兴门外三里河北街 16 号
邮政编码:100045

网址 www.bzeps.com

电话:68523946 68517548

中国标准出版社秦皇岛印刷厂印刷
各地新华书店经销

*

开本 880×1230 1/16 印张 3.5 字数 102 千字
2004 年 12 月第一版 2004 年 12 月第一次印刷
印数 1—5 000

*

书号: 155066·1-21772 定价 22.00 元

如有印装差错 由本社发行中心调换
版权专有 侵权必究
举报电话:(010)68533533

前 言

本指导性技术文件包括两个通信规程中使用的 Modbus 应用层协议和服务规范：

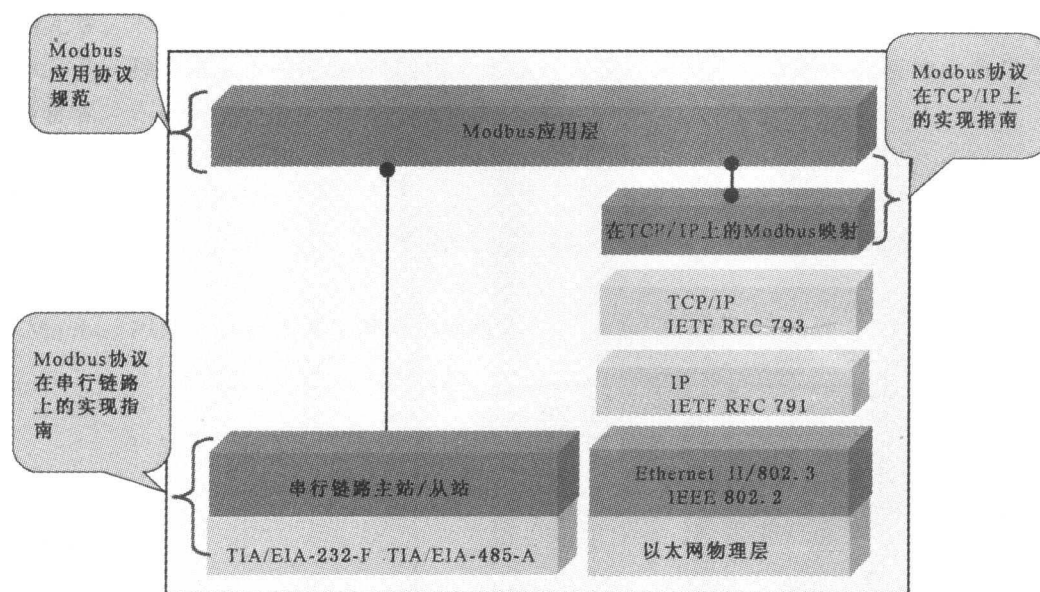
——串行链路上的 Modbus

Modbus 串行链路基于 TIA/EIA 标准：232-F 和 485-A。

——TCP/IP 上的 Modbus

Modbus TCP/IP 基于 IETF 文件：RFC 793 和 RFC 791。

串行链路和 TCP/IP 上的 Modbus 是根据相应 ISO 分层模型说明的两个通信规程。下图强调指出了本指导性技术文件的主要部分。深色方框表示规范，浅色方框表示已有的国际标准（TIA/EIA 和 IETF 标准）。



基于 Modbus 协议的工业自动化网络规范分为三部分。

——第 1 部分：Modbus 应用协议

——第 2 部分：Modbus 协议在串行链路上的实现指南

——第 3 部分：Modbus 协议在 TCP/IP 上的实现指南

第 1 部分描述了 Modbus 事务处理；第 2 部分提供了一个有助于开发者实现串行链路上的 Modbus 应用层的参考信息；第 3 部分提供了一个有助于开发者实现 TCP/IP 上的 Modbus 应用层的参考信息。

本部分的附录 A 是资料性附录。

本部分由中国机械工业联合会提出。

本部分由全国工业过程测量和控制标准化技术委员会归口。

本部分起草单位：机械工业仪器仪表综合技术经济研究所、北京交通大学现代通信研究所、上海自动化仪表股份有限公司、施耐德电气（中国）投资有限公司、冶金工业钢铁研究总院、宝钢集团上海宝钢软件股份有限公司。

本部分主要起草人：欧阳劲松、孙昕、刘铁椎、冯晓升、王勇、张荣生、丛力群、段永康。

目 次

前言	I
1 范围	1
2 规范性引用文件	1
3 缩略语	2
4 背景概要	2
5 总体描述	3
5.1 协议描述	3
5.2 数据编码	4
5.3 Modbus 数据模型	4
5.4 Modbus 寻址模型	6
5.5 Modbus 事务处理的定义	6
6 功能码分类	7
6.1 公共功能码定义	8
7 功能码描述	9
7.1 01(0x01)读线圈	9
7.2 02(0x02)读离散量输入	10
7.3 03(0x03)读保持寄存器	12
7.4 04(0x04)读输入寄存器	14
7.5 05(0x05)写单个线圈	15
7.6 06(0x06)写单个寄存器	16
7.7 07(0x07)读异常状态(仅用于串行链路)	18
7.8 08(0x08)诊断(仅用于串行链路)	19
7.9 11(0x0B)获得通信事件计数器(仅用于串行链路)	23
7.10 12(0x0C)获得通信事件记录(仅用于串行链路)	24
7.11 15(0x0F)写多个线圈	27
7.12 16(0x10)写多个寄存器	29
7.13 17(0x11)报告从站 ID(仅用于串行链路)	30
7.14 20/6(0x14/0x06)读文件记录	31
7.15 21/6(0x15/0x06)写文件记录	33
7.16 22(0x16)屏蔽写寄存器	36
7.17 23(0x17)读/写多个寄存器	37
7.18 24(0x18)读 FIFO 队列	40
7.19 43(0x2B)封装接口传输	41
7.20 43/14(0x2B/0x0E)读设备标识	43
8 Modbus 异常响应	47
附录 A(资料性附录)参考文献	50

基于 Modbus 协议的工业自动化网络规范

第 1 部分: Modbus 应用协议

1 范围

Modbus 是 OSI 模型第 7 层上的应用层报文传输协议,它在连接至不同类型总线或网络的设备之间提供客户机/服务器通信,见图 1。

从 1979 年开始,Modbus 作为工业串行链路的事实标准,Modbus 使成千上万的自动化设备能够通信。目前,对简单而精致的 Modbus 结构的支持仍在增长。互联网用户能够使用 TCP/IP 栈上的保留系统端口 502 访问 Modbus。

Modbus 是一个请求/应答协议,并且提供功能码规定的服务。Modbus 功能码是 Modbus 请求/应答 PDU 的元素。本部分描述了 Modbus 事务处理框架内使用的功能码。

Modbus 是一种应用层报文传输协议,用于在通过不同类型的总线或网络连接的设备之间的客户机/服务器通信。

目前,通过下列方式实现 Modbus 通信:

- 以太网上的 TCP/IP。
- 各种介质(有线:EIA/TIA-232-F、EIA-422、EIA/TIA-485-A;光纤、无线等等)上的异步串行传输。
- Modbus PLUS,一种高速令牌传递网络。

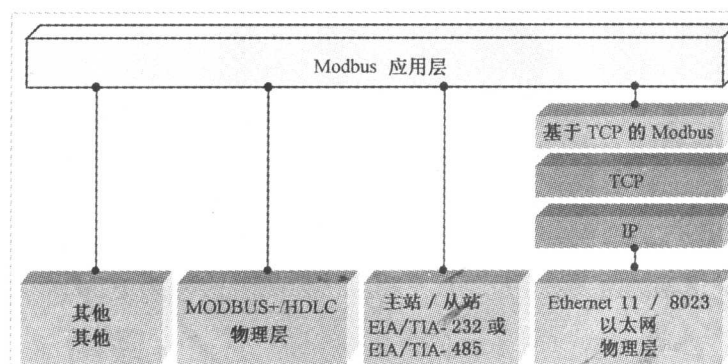


图 1 Modbus 通信栈

2 规范性引用文件

下列文件中的条款通过 GB/Z 19582 本部分的引用而成为本部分的条款。凡是注日期的引用文件,其随后所有的修改单(不包括勘误的内容)或修订版均不适用于本部分,然而,鼓励根据本部分达成协议的各方研究是否可使用这些文件的最新版本。凡是不注日期的引用文件,其最新版本适用于本部分。

GB/T 15969 可编程序控制器

RFC 791, Internet Protocol, Sep81 DARPA*

* 参考文献[1]。

3 缩略语

ADU(Application Data Unit)	应用数据单元
HDLC(High Level Data link control)	高级数据链路控制
HMI(Human Machine Interface)	人机界面
IETF(Internet Engineering Task Force)	互联网工程工作组
I/O(Input/Output)	输入/输出
IP(Internet Protocol)	互联网协议
LSB(Least Significant Bit)	最低有效位
MAC(Medium Access Control)	介质访问控制
MB(Modbus Protocol)	Modbus 协议
MBAP(MODBUS Application Protocol)	Modbus 应用协议
MEI(MODBUS Encapslated Interface)	Modbus 封装接口
MSB(Most Significant Bit)	最高有效位
PDU(Protocol Data Unit)	协议数据单元
PLC(Programmable Logic Controller)	可编程逻辑控制器
RFC(Request For Comment)	请求评注
TCP(Transport Control Protocol)	传输控制协议

4 背景概要

Modbus 协议可以方便地在各种网络体系结构内进行通信,见图 2。

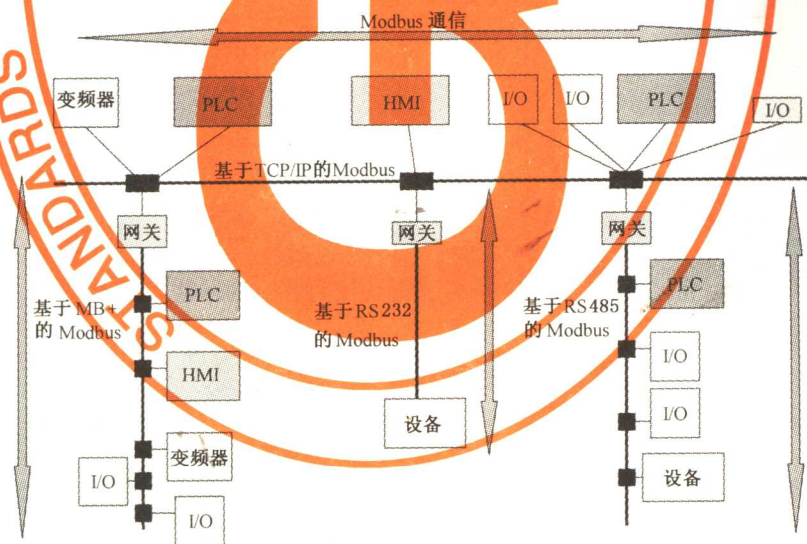


图 2 Modbus 网络体系结构的实例

每种设备(PLC、HMI、控制面板、变频器、运动控制、I/O 设备……)都能使用 Modbus 协议来启动远程操作。

同样的通信能够在基于串行链路和以太网 TCP/IP 网络上进行。网关能够实现在各种使用 Modbus 协议的总线或网络之间的通信。

5 总体描述

5.1 协议描述

Modbus 协议定义了一个与基础通信层无关的简单协议数据单元(PDU)。特定总线或网络上的 Modbus 协议映射能够在应用数据单元(ADU)上引入一些附加域,见图 3。

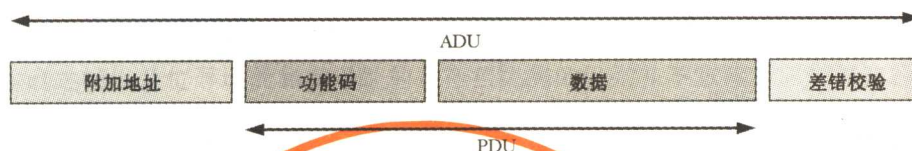


图 3 通用 Modbus 帧

启动 Modbus 事务处理的客户机创建 Modbus 应用数据单元。功能码向服务器指示将执行哪种操作。Modbus 协议建立了客户机启动的请求格式。

用一个字节编码 Modbus 数据单元的功能码域。有效的码字范围是十进制 1~255(128~255 为异常响应保留)。当从客户机向服务器设备发送报文时,功能码域通知服务器执行哪种操作。

向一些功能码加入子功能码来定义多项操作。

从客户机向服务器设备发送的报文数据域包括附加信息,服务器使用这个信息执行功能码定义的操作。这个域还包括离散量和寄存器地址、处理的项目数量以及域中的实际数据字节数。

在某种请求中,数据域可以是不存在的(0 长度),在此情况下服务器不需要任何附加信息。功能码仅说明操作。

如果在一个正确接收的 Modbus ADU 中,不出现与所请求的 Modbus 功能有关的差错,那么服务器至客户机的响应数据域包括所请求的数据。如果出现与所请求的 Modbus 功能有关的差错,那么该域包括一个异常码,服务器应用能够使用这个域确定下一个执行的操作。

例如,客户机能够读一组离散量输出或输入的开/关状态,或者客户机能够读/写一组寄存器的数据内容。

当服务器对客户机响应时,它使用功能码域来指示正常(无差错)响应(见图 4)或者出现某种差错(称为异常响应),见图 5。对于一个正常响应来说,服务器仅复制原始功能码。

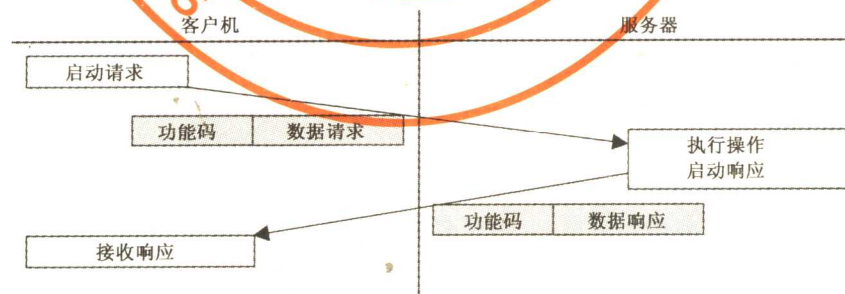


图 4 Modbus 事务处理(无差错)

对于异常响应,服务器将原始功能码的最高有效位设置逻辑 1 后返回。

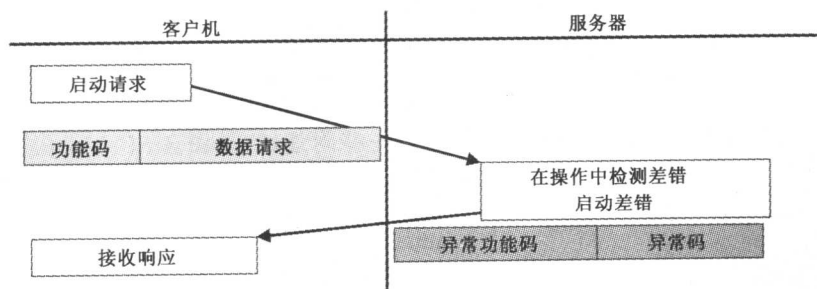


图 5 Modbus 事务处理(异常响应)

注:需要超时管理,以避免无限期地等待可能不会出现的应答。

Modbus 最初在串行链路上的实现(最大 RS 485 ADU=256 字节)限制了 Modbus PDU 的长度。

因此,对串行链路通信来说,Modbus PDU=256-服务器地址(1 字节)-CRC(2 字节)=253 字节。

从而:

RS 232/RS 485 ADU=253 字节+服务器地址(1 字节)+CRC(2 字节)=256 字节。

TCP Modbus ADU=253 字节+MBAP(7 字节)=260 字节。

Modbus 协议定义了三种 PDU。它们是:

- Modbus 请求 PDU, mb_req_pdu;
- Modbus 响应 PDU, mb_rsp_pdu;
- Modbus 异常响应 PDU, mb_except_rsp_pdu。

定义 mb_req_pdu 为:

mb_req_pdu={function_code, request_data}, 其中

function_code: [1 字节]Modbus 功能码。

request_data: [n 字节], 这个域与功能码有关, 并且通常包括诸如参考变量、变量计数、数据偏移量、子功能码等信息。

定义 mb_rsp_pdu 为:

mb_rsp_pdu={function_code, response_data}, 其中

function_code: [1 字节]Modbus 功能码。

response_data: [n 字节], 这个域与功能码有关, 并且通常包括诸如参考变量、变量计数、数据偏移量、子功能码等信息。

定义 mb_except_rsp_pdu 为:

mb_except_rsp_pdu={function_code, request_data}, 其中

function_code: [1 字节]Modbus 功能码+0x80。

exception_code: [1 字节], 在第 7 章中定义了 Modbus 异常码。

5.2 数据编码

Modbus 使用最高有效字节在低地址存储的方式表示地址和数据项。这意味着当发送多个字节时, 首先发送最高有效字节。例如:

寄存器大小 值

16 位 0x1234 发送的第一字节为 0x12 然后 0x34

注: 更详细的信息参见[1]。

5.3 Modbus 数据模型

Modbus 的数据模型是以一组具有不同特征的表为基础建立的。4 个基本表见表 1:

表 1

基本表	对象类型	访问类型	注释
离散量输入	单个位	只读	I/O 系统可提供这种类型数据
线圈	单个位	读写	通过应用程序可改变这种类型数据
输入寄存器	16 位字	只读	I/O 系统可提供这种类型数据
保持寄存器	16 位字	读写	通过应用程序可改变这种类型数据

输入与输出之间以及位寻址的和字寻址的数据项之间的区别并不意味着应用特性的差别。如果所有 4 个表相互覆盖是对该目标机器最自然的解释,也是完全可接受的,而且很普遍。

对于每个基本表,协议都允许单个地选择 65536 个数据项,而且其读写操作被设计为可以越过多个连续数据项直到数据大小规格限制,这个数据大小规格限制与事务处理功能码有关。

很显然,必须将 Modbus 处理的所有数据(位,寄存器)放置在设备应用存储器中。但是,存储器的物理地址不应该与寄存器编号混淆。仅要求将寄存器编号与物理地址链接。

Modbus 功能码中使用的 Modbus 寄存器逻辑编号是以 0 开始的无符号整数索引。

——Modbus 模型实现的实例

下列实例表示了两种在设备中组织数据的方法。有多种组织数据的方法,在本部分中没有被全部描述。每个设备根据其应用都有它自己的组织数据的方法。

实例 1:有 4 个独立块的设备

图 6 表示了含有数字量和模拟量、输入量和输出量的设备中的数据组织。由于不同块中的数据不相关,每个块是相互独立的。可通过不同 Modbus 功能码访问每个块。

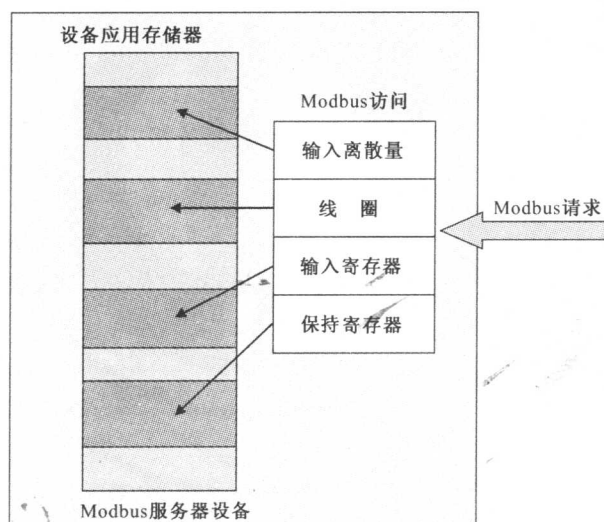


图 6 带有独立块的 Modbus 数据模型

实例 2:仅有 1 个块的设备

图 7 实例中,设备仅有 1 个数据块。通过几个 Modbus 功能码能够得到相同数据,既可通过 16 位访问也可通过 1 位访问。

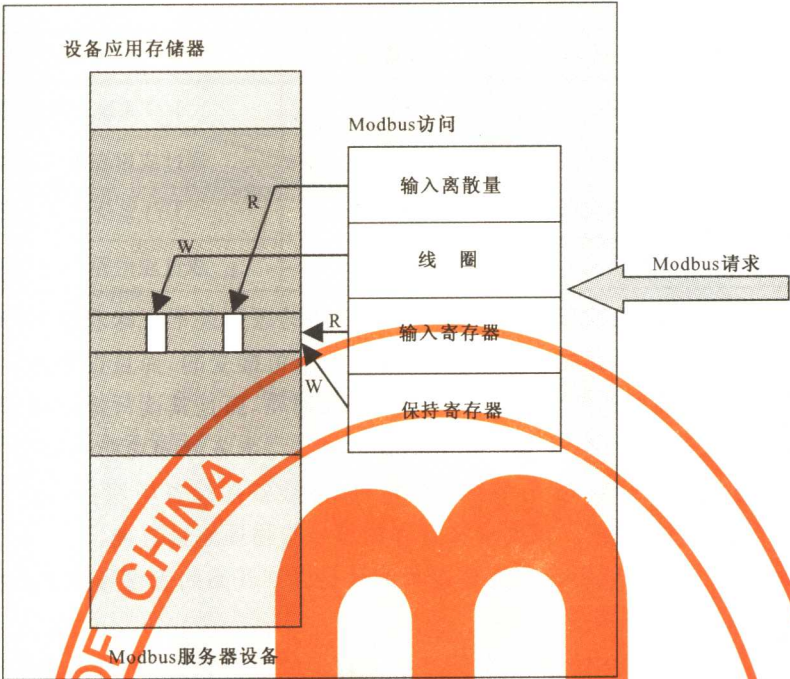


图 7 仅带有 1 个块的 Modbus 数据模型

5.4 Modbus 寻址模型

Modbus 应用协议精确地定义了 PDU 寻址规则。
在 MUDBUS PDU 中,从 0~65535 寻址每个数据。
Modbus 应用协议还明确地定义了由 4 个块构成的 Modbus 数据协议,每个块由几个编号为 1~n 的元素构成。
在 Modbus 数据模型中,从 1~n 编号数据块中的每个元素。
然后,必须将 Modbus 模型与设备应用结合(GB/T 15969)。
Modbus 数据模型和设备应用之间的映射完全与特定设备相关。

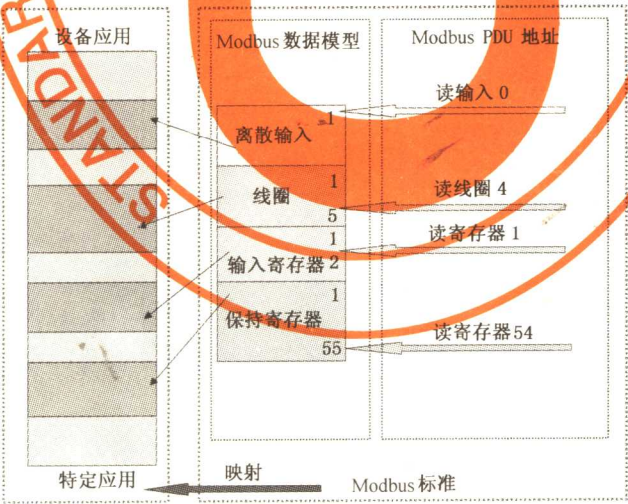


图 8 Modbus 寻址映射

图 8 表示了用 Modbus PDU 的 X-1 寻址编号为 X 的 Modbus 数据。
5.5 Modbus 事务处理的定义
图 9 是状态图,描述了在服务器侧 Modbus 事务处理的一般处理过程。

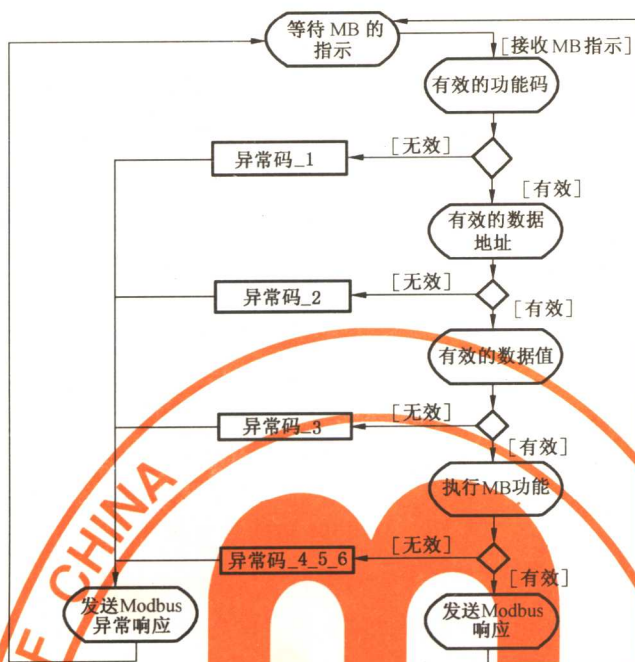


图 9 Modbus 事务处理的状态图

一旦服务器处理请求,使用合适的 Modbus 服务器事务处理建立 Modbus 响应。

根据处理结果,可以建立两种类型响应:

——一个正常的 Modbus 响应:响应功能码=请求功能码。

——一个异常的 Modbus 响应(见第 8 章):

- 1) 用来为客户机提供处理过程中与所发现的差错相关的信息;
- 2) 异常功能码=请求功能码+0x80;
- 3) 提供一个异常码来指示差错原因。

6 功能码分类

有三类 Modbus 功能码,见图 10。它们是:

——公共功能码

- 1) 被确切定义的功能码;
- 2) 保证是唯一的;
- 3) 由 Modbus.org 确认的;
- 4) 公开的文档;
- 5) 可进行一致性测试;
- 6) 在 MB IETF RFC 中归档;
- 7) 包含已被定义的公共功能码和保留给未来使用的功能码。

——用户定义的功能码

- 1) 有两个用户定义功能码的区域,即十进制的 65~72 和 100~110。
- 2) 用户无需 Modbus 组织的任何批准就可以选择和实现一个功能码。
- 3) 不能保证被选功能码的使用是唯一的。
- 4) 如果用户希望将某种功能设置为一个公共功能码,那么用户必须启动 RFC,以便将改变引入公共分类中,并且指派一个新的公共功能码。

——保留功能码

某些公司在传统产品上现行使用的功能码,不作为公共使用。

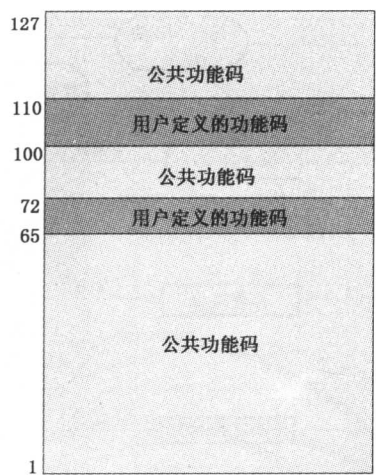


图 10 Modbus 功能码分类

注：在功能码的公共区域中，可以保留一些功能码。

6.1 公共功能码定义

见表 2。

表 2

				功能码		
				码	子码	(十六进制)
数 据 访 问	比特访问	物理离散量输入	读离散量输入	02	—	02
		内部比特或物理线圈	读线圈	01	—	01
			写单个线圈	05	—	05
			写多个线圈	15	—	0F
	16 比特访问	输入寄存器	读输入寄存器	04	—	04
		内部存储器或物理输出存储器	读多个寄存器	03	—	03
			写单个寄存器	06	—	06
			写多个寄存器	16	—	10
			读/写多个寄存器	23	—	17
			屏蔽写寄存器	22	—	16
			读 FIFO 队列	24	—	18
		文件记录访问	读文件记录	20	6	14
			写文件记录	21	6	15
	诊断		读异常状态	07	—	—
			诊断	08	00-18	—
			获得公用事件计数器	11	—	0B
			获得公用事件记录	12	—	0C
			报告从站 ID	17	—	11
			读设备识别码	43	14	2B
	其他		封装接口传输	43	—	2B

7 功能码描述

7.1 01(0x01)读线圈

见图 11 和表 3~表 5。

使用该功能码从一个远程设备中读 1~2 000 个连续的线圈状态。请求 PDU 指定了起始地址,即指定了第一个线圈地址和线圈数目。在 PDU 中,从 17(十六进制 11)返回从站忙计数

响应数据域返回上一次重启动、清除计数器操作或加电之后对返回从站设备忙异常响应的远程设备寻址的报文数量。

子功能	数据域(请求)	数据域(响应)
00 11	00 00	从站设备忙,从零开始寻址线圈。因此编号 1~16 的线圈寻址为 0~15。

响应报文中的线圈按数据域的每位一个线圈进行打包。状态被表示成 1=ON 和 0=OFF。第一个数据字节的 LSB(最低有效位)包含询问中所寻址的输出。其他线圈依次类推,一直到这个字节的高位端为止,并在后续字节中按照从低位到高位顺序排列。

如果返回的输出数量不是 8 的倍数,将用零填充最后数据字节中的剩余位(一直到字节的高位端)。字节数量域说明了数据的全部字节数。

表 3

请求

功能码	1 字节	0x01
起始地址	2 字节	0x0000~0xFFFF
线圈数量	2 字节	1~2 000(0x07D0)

表 4

响应

功能码	1 字节	0x01
字节计数	1 字节	N*
线圈状态	n 字节	n=N 或 N+1
* N=输出数量/8,如果余数不等于 0,那么 N=N+1。		

表 5

错误

功能码	1 字节	功能码+0x80
异常码	1 字节	01 或 02 或 03 或 04

表 6 是一个请求读离散量输出 20~38 的实例:

表 6

请 求		响 应	
域名	(十六进制)	域名	(十六进制)
功能	01	功能	01
起始地址 Hi	00	字节计数	03
起始地址 Lo	13	输出状态 27~20	CD
输出数量 Hi	00	输出状态 35~28	6B
输出数量 Lo	13	输出状态 38~36	05

将输出 27~20 的状态表示为十六进制字节值 CD,或二进制 1100 1101。输出 27 是这个字节的 MSB,输出 20 是 LSB。

通常,一个字节的 MSB 位于左侧,LSB 位于右侧。第一个字节的输出从左至右为 27~20。下一个字节的输出从左到右为 35~28。当串行发送位时,从 LSB 向 MSB 传输: 20...27、28...35 等等。

在最后的的数据字节中,将输出 38~36 的状态表示为十六进制字节值 05,或二进制 0000 0101。输出 38 是左侧第 6 个位位置,输出 36 是这个字节的 LSB。用零填充 5 个剩余高位位。

注:用零填充 5 个剩余位(一直到高位端)。



图 11 读线圈状态图

7.2 02(0x02)读离散量输入

见图 12 和表 7~表 9。

使用该功能码从一个远程设备中读 1~2 000 个连续的离散量输入状态。请求 PDU 指定了起始地址,即指定了第一个离散量输入地址和离散量输入数目。在 PDU 中,从零开始寻址离散量输入。因此编号 1~16 的离散量输入寻址为 0~15。

响应报文中的离散量输入按数据域的每一位一个离散量输入进行打包。状态被表示成 1=ON 和 0=OFF。第一个数据字节的 LSB(最低有效位)包含询问中所寻址的输入。其他离散量输入依次类推,一直到这个字节的高位端为止,并在后续字节中按照从低位到高位顺序排列。

如果返回的输入数量不是 8 的倍数,将用零填充最后数据字节中的剩余位(一直到字节的高位端)。字节数量域说明了数据的全部字节数。

表 7

请求

功能码	1 字节	0x02
起始地址	2 字节	0x0000~0xFFFF
输入数量	2 字节	1~2 000(0x07D0)

表 8

响应

功能码	1 字节	0x02
字节计数	1 字节	N*
输入状态	N*×1 个字节	...
* N=输入数量/8,如果余数不等于 0,那么 N=N+1。		

表 9

错误

差错码	1 字节	0x82
异常码	1 字节	01 或 02 或 03 或 04

表 10 是一个请求读离散量输入 197~218 的实例:

表 10

请 求		响 应	
域名	(十六进制)	域名	(十六进制)
功能	02	功能	02
起始地址 Hi	00	字节计数	03
起始地址 Lo	C4	输入状态 204~197	AC
输出数量 Hi	00	输入状态 212~205	DB
输出数量 Lo	16	输入状态 218~213	35

将离散量输入 204~197 的状态表示为十六进制字节值 AC,或二进制 1010 1100。输入 204 是这个字节的 MSB,输入 197 是这个字节的 LSB。

将离散量输入 218~213 的状态表示为十六进制字节值 35,或二进制 0011 0101。输入 218 位于左侧第 3 位,输入 213 是 LSB。

注:用零填充 2 个剩余位(一直到高位端)。

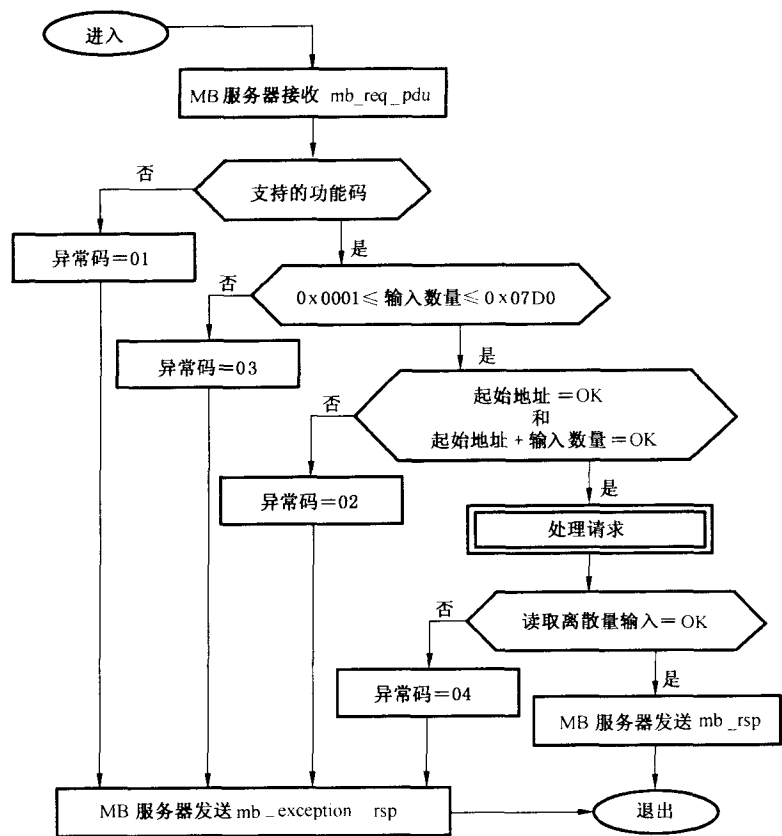


图 12 读离散量输入的状态图

7.3 03(0x03)读保持寄存器

见图 13 和表 11～表 13。

使用该功能码从远程设备中读保持寄存器连续块的内容。请求 PDU 说明了起始寄存器地址和寄存器数量。在 PDU 中,从零开始寻址寄存器。因此编号 1～16 的寄存器寻址为 0～15。

将响应报文中的寄存器数据打包成每个寄存器有两字节。

对于每个寄存器,第一个字节为高位字节,第二个字节为低位字节。

表 11

请求

功能码	1 字节	0x03
起始地址	2 字节	0x0000～0xFFFF
寄存器数量	2 字节	1～125(0x007D)

表 12

响应

功能码	1 字节	0x03
字节数	1 字节	2×N*
寄存器值	N*×2 字节	...

* N=寄存器的数量。

表 13

错误

差错码	1 字节	0x83
异常码	1 字节	01 或 02 或 03 或 04

表 14 是一个请求读寄存器 108~110 的实例：

表 14

请 求		响 应	
域名	(十六进制)	域名	(十六进制)
功能	03	功能	03
起始地址 Hi	00	字节计数	06
起始地址 Lo	6B	寄存器值 Hi(108)	02
寄存器数量 Hi	00	寄存器值 Lo(108)	2B
寄存器数量 Lo	03	寄存器值 Hi(109)	00
		寄存器值 Lo(109)	00
		寄存器值 Hi(110)	00
		寄存器值 Lo(110)	64

将寄存器 108 的内容表示为两个十六进制字节值 02 2B,或十进制 555。将寄存器 109~110 的内容分别表示为十六进制 00 00 和 00 64,或十进制 0 和 100。

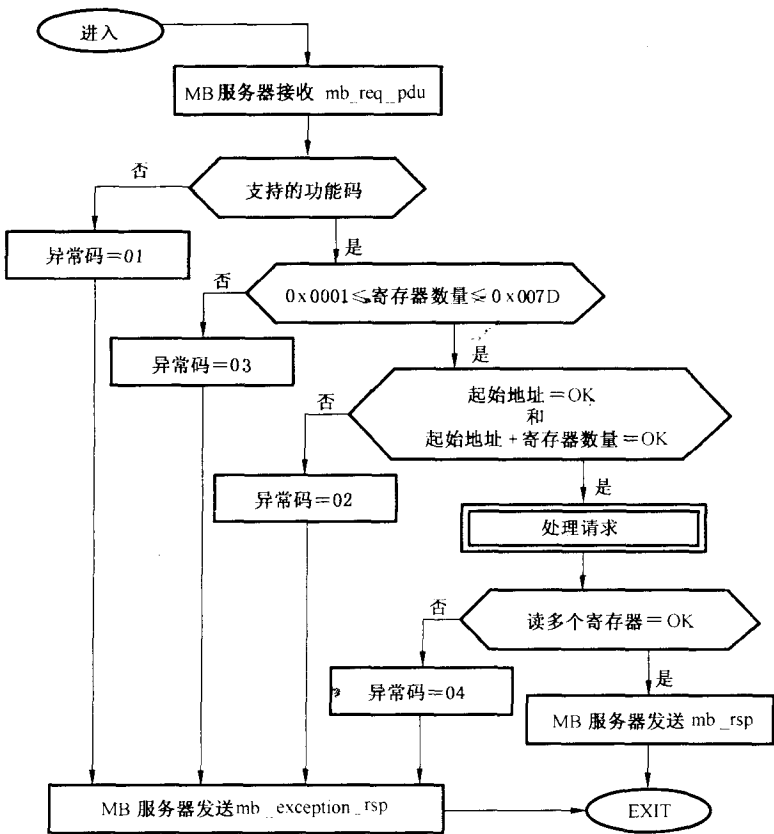


图 13 读保持寄存器的状态图