ICS 25.040
N 10

中华人民共和国国家标准

GB/T 19582.1—2008
代替 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

2008-02-27 发布　　2008-09-01 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

目　次

前　言

GB/T 19582—2008《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 部分提供了有助于开发者在串行链路上实现 Modbus 应用层的参考信息；第 3 部分提供了有助于开发者在 TCP/IP 上实现 Modbus 应用层的参考信息。

GB/T 19582—2008 包括两个通信规程中使用的 Modbus 应用层协议和服务规范：

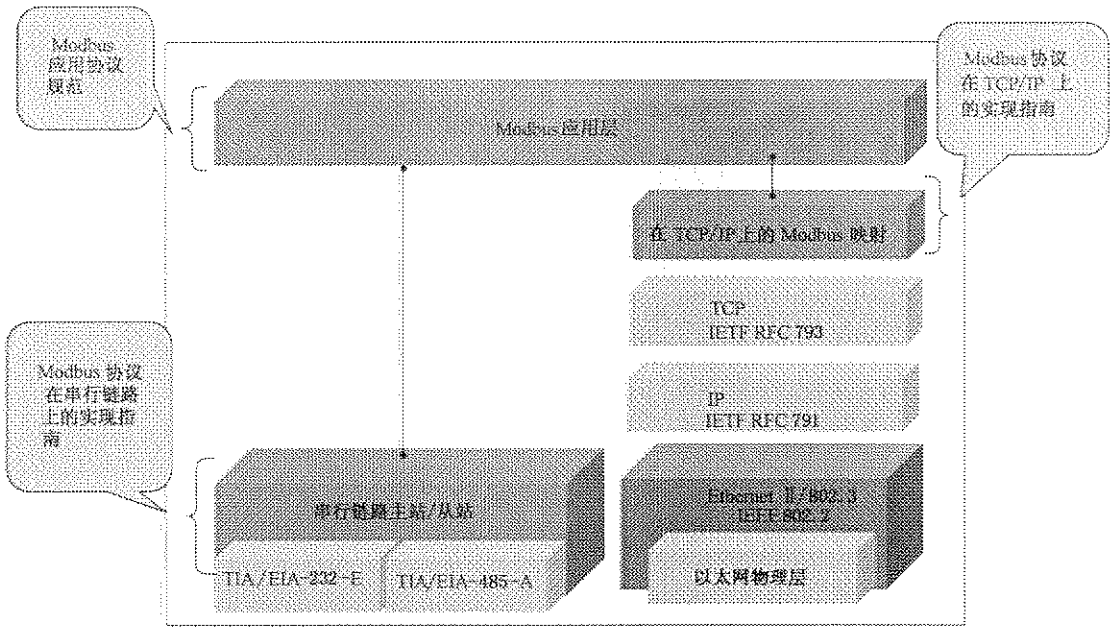
——串行链路上的 Modbus

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

——TCP/IP 上的 Modbus

Modbus TCP/IP 基于 IETF 标准：RFC793 和 RFC791。

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 GB/T 19582—2008 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)。



本部分从实施之日起代替 GB/Z 19582.1—2004；GB/Z 19582.1—2004 并于该日起予以废止。

本部分的附录 A、附录 B 为资料性附录。

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第四分技术委员会归口。

本部分起草单位：机械工业仪器仪表综合技术经济研究所、西南大学、上海自动化仪表股份有限公司、北京交通大学现代通信研究所、北京机械工业自动化研究所、国家继电器质量监督检验中心、中国四联仪器仪表集团有限公司、中海石油研究中心、西北工业大学、施耐德电气(中国)投资有限公司。

本部分主要起草人：王玉敏、柳晓菁、刘枫、包伟华、孙昕、刘云男、唐济扬、贺春、刘渝新、徐伟华、欧阳劲松、何军红、华锦、王勇。

GB/Z 19582.1 首次发布时间为 2004 年 9 月 21 日，本部分第一次修订。

引　言

GB/T 19582—2008 是对 GB/Z 19582—2004《基于 Modbus 协议的工业自动化网络规范》的修订。修订的依据是 IEC 61158 CPF15(FDIS):2006 实时以太网 Modbus-RTPS。本部分的结构与GB/Z 19582.1—2004 基本一致,但在技术内容上对 GB/Z 19582.1—2004 进行了补充和完善。

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 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，见 GB/T 19582.3。

——各种介质（有线：EIA/TIA-232-E、EIA-422、EIA/TIA-485-A；光纤、无线等等）上的异步串行传输。

——Modbus+，一种高速令牌传递网络。

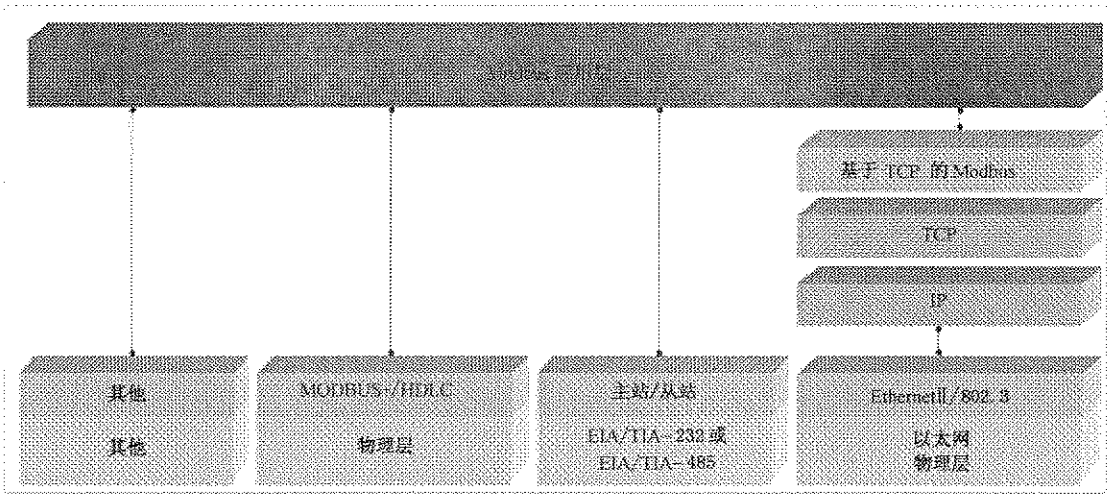


图 1 Modbus 通信栈

2 规范性引用文件

下列文件中的条款通过 GB/T 19582 的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

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)	因特网协议
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 Encapsulated Interface)	Modbus 封装接口
MSB(Most Significant Bit)	最高有效位
NAK(Negative Acknowlegment)	否定确认
PDU(Protocol Data Unit)	协议数据单元
PLC(Programmable Logic Controller)	可编程序逻辑控制器
RFC(Request For Comment)	互联网"请求注解"文档
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。

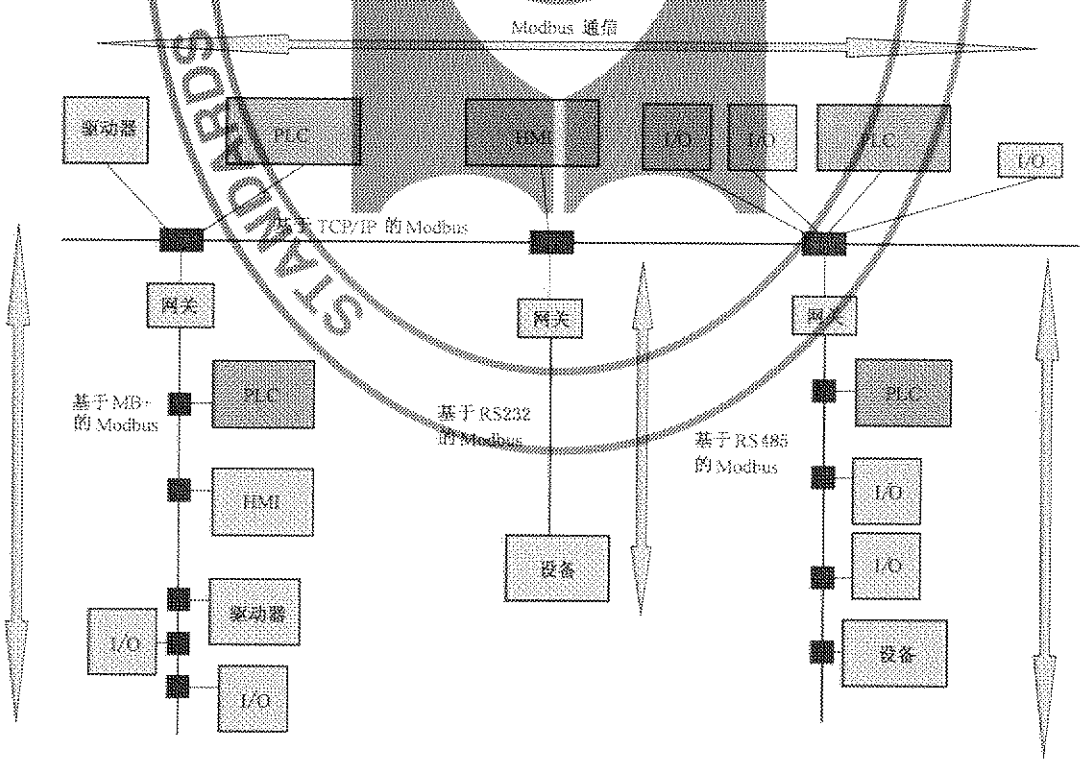


图 2 Modbus 网络体系结构的示例

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

在基于串行链路和 TCP/IP 以太网上能够进行同样的通信。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)上引入一些附加字段,见图 3。

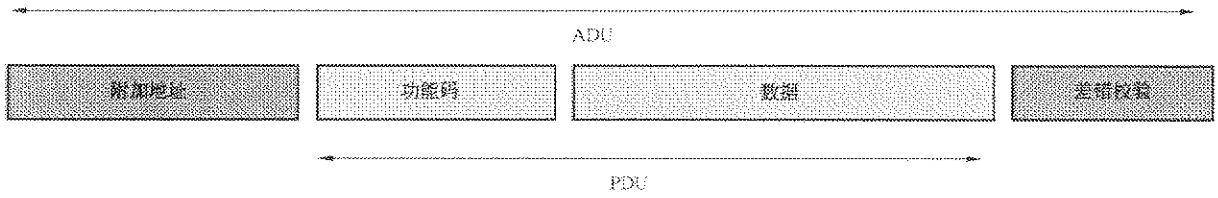


图 3 通用 Modbus 帧

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

用一个字节编码 Modbus 数据单元的功能码字段。有效的码范围是十进制 1～255(128～255 保留用于异常响应)。当从客户机向服务器设备发送报文时,功能码字段通知服务器执行哪种操作。功能码"0"无效。

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字段包括附加信息,服务器使用这个信息执行功能码定义的操作。这个字段还包括离散量和寄存器地址、处理的项目数量以及字段中的实际数据字节数。

在某种请求中,数据字段可以是不存在的(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字段包括所请求的数据。如果出现与所请求的 Modbus 功能有关的差错,那么该字段包括一个异常码,服务器应用能够使用这个字段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,它使用功能码字段来指示正常(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。对于正常响应,服务器仅复制原始功能码。

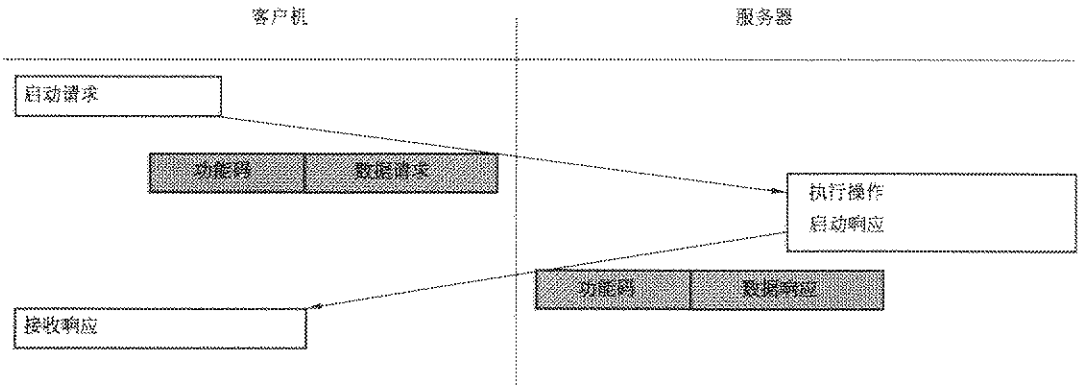


图 4 Modbus 事务处理(无差错)

对于异常响应，服务器将请求 PDU 中的原始功能码的最高有效位设置逻辑 1 后返回。

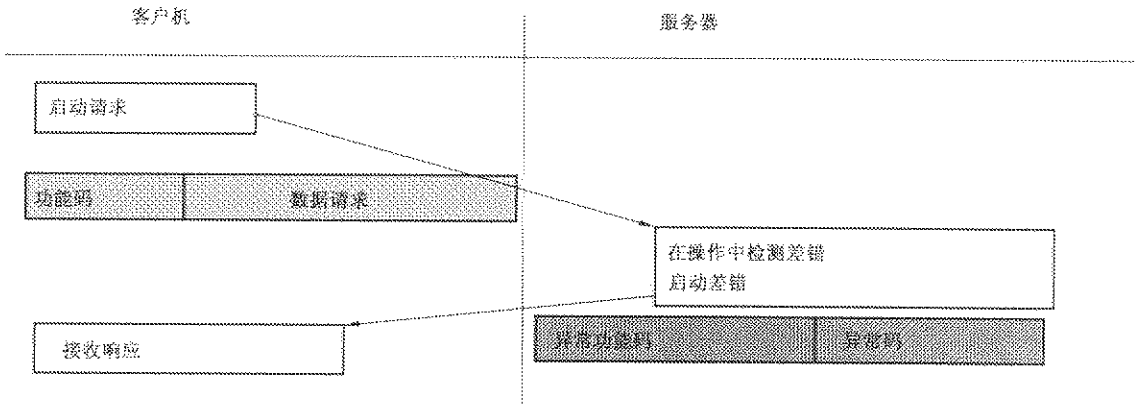


图 5　Modbus 事务处理(异常响应)

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 RS485 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：

RS232/RS485 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_excep_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]，这个字段与功能码有关，通常包括诸如变量引用、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]，这个字段与功能码有关，通常包括诸如变量引用、变量计数、数据偏移量、子功能码等信息。

定义 mb_excep_rsp_pdu 为：

mb_excep_rsp_pdu＝{exception-function_code，exception_code}，其中：

exception-function_code＝[1 字节] Modbus 功能码＋0x80。

exception_code＝[1 字节] Modbus 异常码，在表“Modbus 异常码”中定义(见第 8 章)。

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：更详细的信息参见参考文献[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 Modbus 数据模型基本表

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的数据
线圈	单个位	读写	通过应用程序可改变这种类型的数据
输入寄存器	16 位字	只读	I/O 系统可提供这种类型的数据
保持寄存器	16 位字	读写	通过应用程序可改变这种类型的数据

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 65 536 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，该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数据引用混淆。仅要求将数据引用与物理地址链接。

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

——Modbus 模型实现的示例

下列示例表示了在设备中组织数据的两种方法。组织数据的方法有多种，在本部分中没有对所有方法进行描述。每个设备根据其应用都有它自己的组织数据的方法。

示例 1：含有 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，因此每个块是相互独立的。这样可通过不同 Modbus 功能码访问每个块。

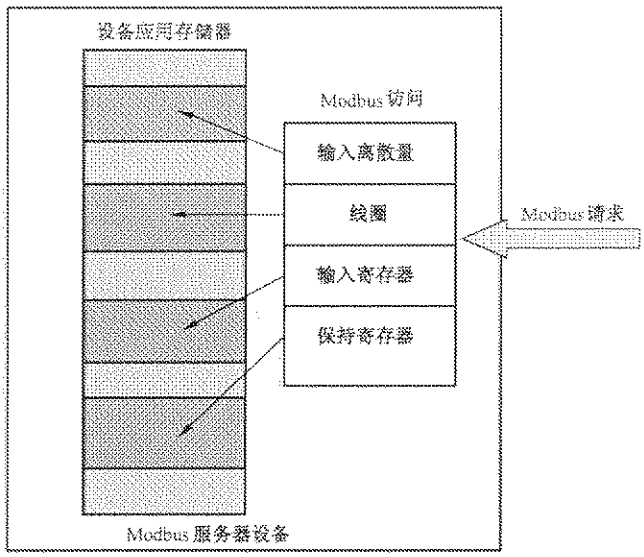


图 6 含有独立块的 Modbus 数据模型

示例 2：仅含有 1 个块的设备

在图 7 中，设备仅含有 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。

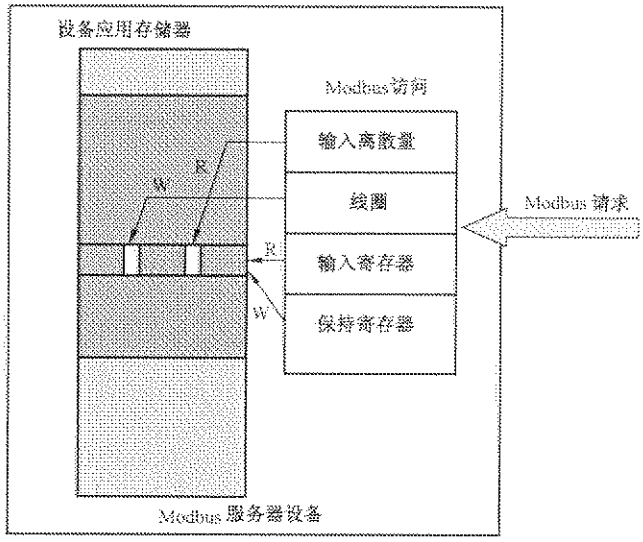


图 7　仅含有 1 个块的 Modbus 数据模型

5.4　Modbus 寻址模型

Modbus 应用协议精确地定义了 PDU 寻址规则。

在 Modbus 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 数据模型，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 来编号数据块中的每个元素。

然后，必须将 Modbus 数据模型与设备应用结合(GB/T 15969 对象，其他应用模型)。

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 表示了用 Modbus PDU 的 X－1 来寻址编号为 X 的 Modbus 数据。

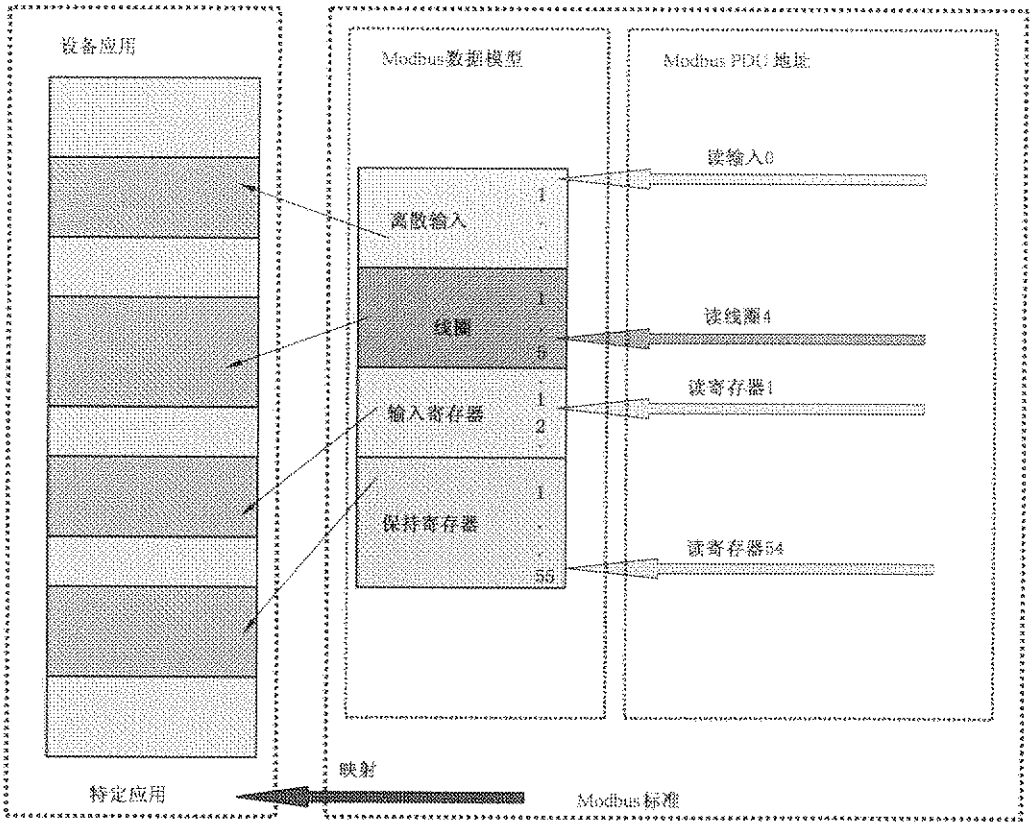


图 8　Modbus 寻址模型

5.5 Modbus事务处理的定义

图9是Modbus事务处理状态图，描述了在服务器侧Modbus事务处理的一般处理过程。

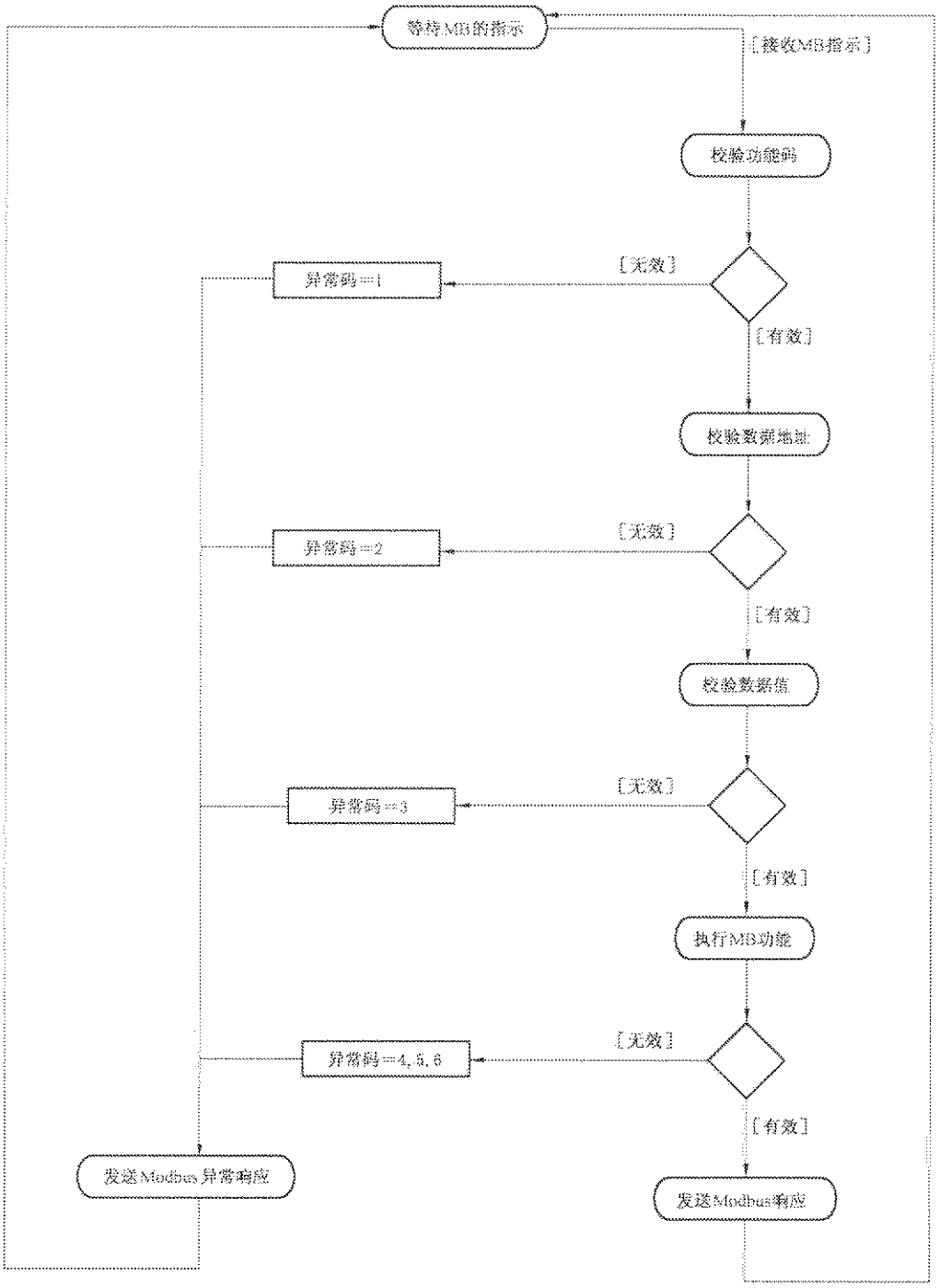


图9 Modbus事务处理的状态图

一旦服务器处理请求，就使用相应的Modbus服务器事务处理生成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：

a） 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-IDA.org 确认的；

4） 公开的文档；

5） 可进行一致性测试；

6） 包含已被定义的公共功能码和保留给未来使用的功能码。

b） 用户定义功能码

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） 不能保证为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，以便将这种变更引入公共分类中，并且指配一个新的公共功能码。

c） 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。

注：参见附录 A。

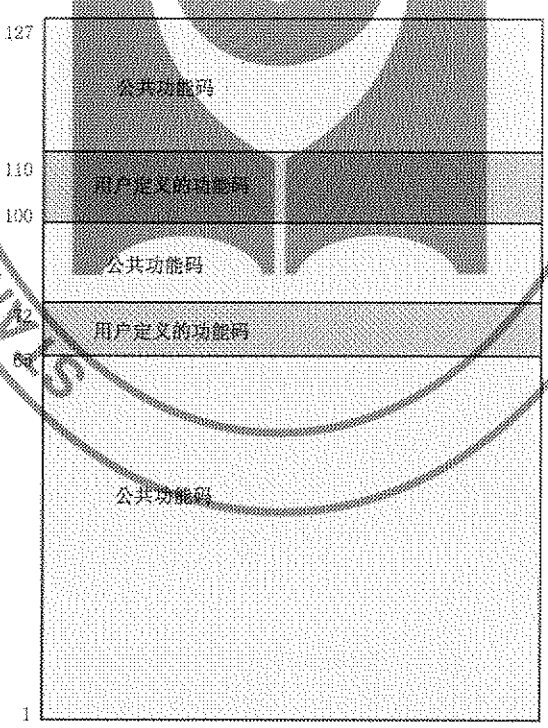


图 10 Modbus 功能码分类

6.1 公共功能码定义

见表 2。

表 2 公共功能码

				功能码			
				码	子码	十六进制	章节
数据访问	比特访问	物理离散量输入	读离散量输入	02		02	7.2
		内部比特或物理线圈	读线圈	01		01	7.1
			写单个线圈	05		05	7.5
			写多个线圈	15		0F	7.11
	16 比特访问	物理输入寄存器	读输入寄存器	04		04	7.4
		内部寄存器或物理输出器寄存器	读保持寄存器	03		03	7.3
			写单个寄存器	06		06	7.6
			写多个寄存器	16		10	7.12
			读/写多个寄存器	23		17	7.17
			屏蔽写寄存器	22		16	7.16
			读 FIFO 队列	24		18	7.18
	文件记录访问		读文件记录	20	6	14	7.14
			写文件记录	21	6	15	7.15
诊断			读异常状态	07		07	7.7
			诊断	08	00-18,20	08	7.8
			获得事件计数器	11		0B	7.9
			获得事件记录	12		0C	7.10
			报告从站 ID	17		11	7.13
			读设备标识码	43	14	2B	7.21
其他			封装接口传输	43	13,14	2B	7.19
			CANopen 通用引用	43	13	2B	7.20

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字段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的顺序排列。

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)。字节计数字段指定了数据的全部字节数。

表 3　读线圈请求

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(0x7D0)

表 4　读线圈响应

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　读线圈错误响应

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

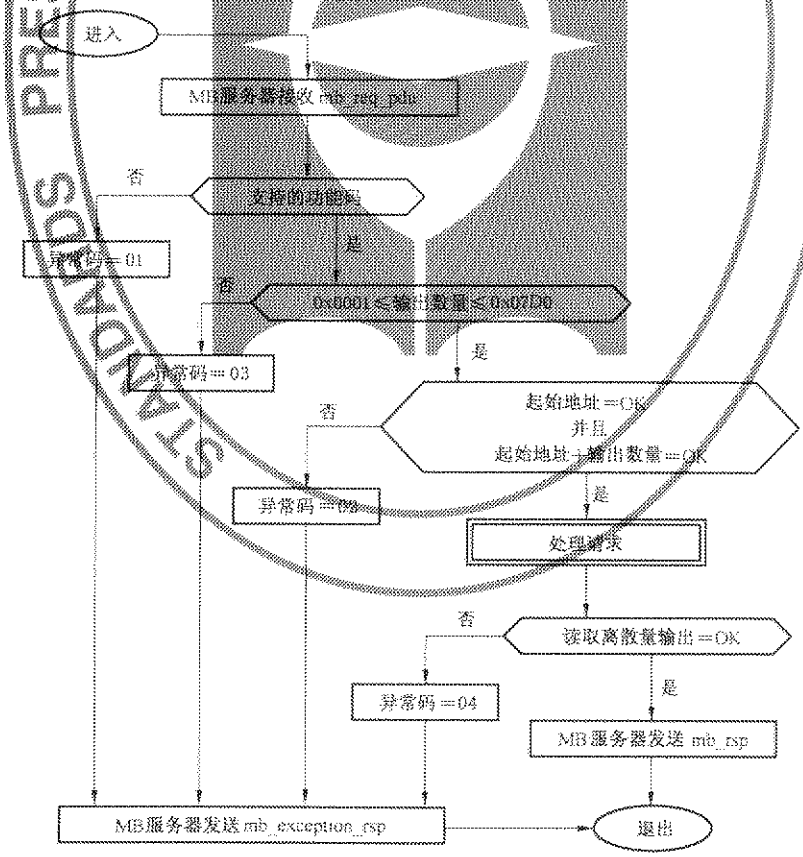


图 11　读线圈状态图

表 63 是一个请求将一个引用组写入远程设备的示例。

组包括文件 4 中的 3 个寄存器，以寄存器 7 开始(地址 0007)。

表 63　一个引用组写入远程设备

请　求		响　应	
字段名	十六进制	字段名	十六进制
功能	15	功能	15
请求数据长度	0D	请求数据长度	0D
子请求 1,引用类型	06	子请求 1,引用类型	06
子请求 1,文件号 Hi	00	子请求 1,文件号 Hi	00
子请求 1,文件号 Lo	04	子请求 1,文件号 Lo	04
子请求 1,记录号 Hi	00	子请求 1,记录号 Hi	00
子请求 1,记录号 Lo	07	子请求 1,记录号 Lo	07
子请求 1,寄存器长度 Hi	00	子请求 1,寄存器长度 Hi	00
子请求 1,寄存器长度 Lo	03	子请求 1,寄存器长度 Lo	03
子请求 1,寄存器数据 Hi	06	子请求 1,寄存器数据 Hi	06
子请求 1,寄存器数据 Lo	AF	子请求 1,寄存器数据 Lo	AF
子请求 1,寄存器数据 Hi	04	子请求 1,寄存器数据 Hi	04
子请求 1,寄存器数据 Lo	BE	子请求 1,寄存器数据 Lo	BE
子请求 1,寄存器数据 Hi	10	子请求 1,寄存器数据 Hi	10
子请求 1,寄存器数据 Lo	0D	子请求 1,寄存器数据 Lo	0D

7.16　22(0x16)屏蔽写寄存器

见图 26 和表 64～表 66。

该功能码用于典型利用"AND_Mask"、"OR_Mask"以及当前寄存器内容的组合来修改指定的保持寄存器的内容。这个功能可用来设置或清除寄存器中的不同位。

请求指定了被写入的保持寄存器，被用于"AND_Mask"的数据以及被用于"OR_Mask"的数据。从 0 开始寻址寄存器。因此，寄存器 1～16 被寻址为 0～15。

功能的算法为：

结果＝(当前内容 AND And_Mask) OR(Or_Mask AND(NOT And_Mask))

例如：

	十六进制	二进制
当前内容＝	12	0001 0010
And_Mask＝	F2	1111 0010
Or_Mask＝	25	0010 0101
NOT And_Mask＝	0D	0000 1101
结果＝	17	0001 0111

注 1：如果 Or_Mask 值为零，那么结果是当前内容和 And_Mask 的简单逻辑 AND(与)。如果 And_Mask 值为零，则结果等于 Or_Mask 值。

注 2：使用读保持寄存器功能(功能码 03)可读出寄存器的内容。但是，当控制器扫描它的用户逻辑程序时，随之可以改变寄存器的内容。

正常的响应是请求的复制。在寄存器被写入之后，返回响应。

表 64　屏蔽写寄存器请求

功能码	1 字节	0x16
引用地址	2 字节	0x0000～0xFFFF
And_Mask	2 字节	0x0000～0xFFFF
Or_Mask	2 字节	0x0000～0xFFFF

表 65　屏蔽写寄存器响应

功能码	1 字节	0x16
引用地址	2 字节	0x0000～0xFFFF
And_Mask	2 字节	0x0000～0xFFFF
Or_Mask	2 字节	0x0000～0xFFFF

表 66　屏蔽写寄存器错误响应

异常功能码	1 字节	0x96
异常码	1 字节	01 或 02 或 03 或 04

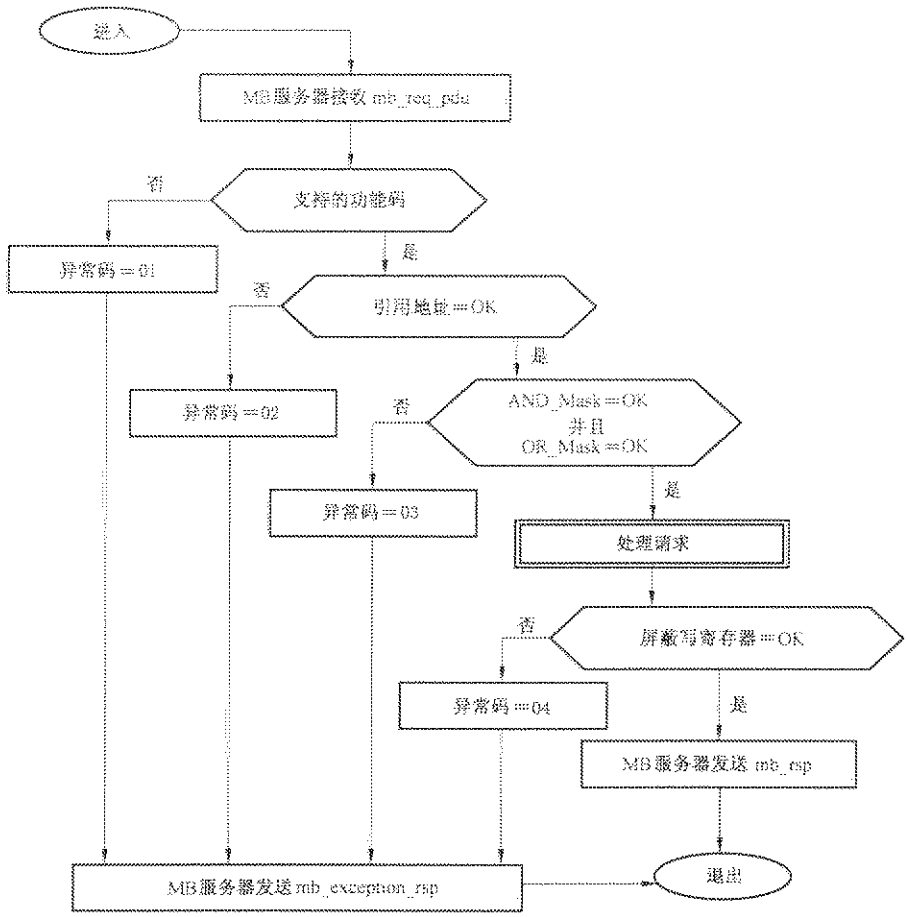


图 26　屏蔽写保持寄存器状态图

表 67 是一个利用上述屏蔽值在远程设备中对第 5 个寄存器的屏蔽写入的示例。

表 67 远程设备中对第 5 个寄存器的屏蔽写入

请求		响应	
字段名	十六进制	字段名	十六进制
功能	16	功能	16
引用地址 Hi	00	引用地址 Hi	00
引用地址 Lo	04	引用地址 Lo	04
And_Mask Hi	00	And_Mask Hi	00
And_Mask Lo	F2	And_Mask Lo	F2
Or_Mask Hi	00	Or_Mask Hi	00
Or_Mask Lo	25	Or_Mask Lo	25

7.17 23(0x17) 读/写多个寄存器

见图 27 和表 68～表 70。

这个功能码实现了在一个单独 Modbus 事务处理中一个读操作和一个写操作的组合。在读之前进行写操作。

从零开始寻址保持寄存器。因此，在 PDU 中保持寄存器 1～16 被寻址为 0～15。

请求指定了被读取的保持寄存器的起始地址和数目、被写入的保持寄存器的起始地址和数目以及数据。在写数据字段中，字节计数指定随后的字节数目。

正常的响应包含所读寄存器组中的数据。在读数据字段中，字节计数字段指定随后的字节数目。

表 68 读/写多个寄存器请求

功能码	1 字节	0x17
读起始地址	2 字节	0x0000～0xFFFF
读的数量	2 字节	0x0001～0x007D
写起始地址	2 字节	0x0000～0xFFFF
写的数量	2 字节	0x0001～0x0079
写字节计数	1 字节	2×N[a]
写寄存器值	N[a]×2 字节	

[a] N＝写的数量。

表 69 读/写多个寄存器响应

功能码	1 字节	0x17
字节计数	1 字节	2×N[a]
读寄存器值	N[a]×2 字节	…

[a] N＝读的数量。

表 70 读/写多个寄存器错误响应

异常功能码	1 字节	0x97
异常码	1 字节	01 或 02 或 03 或 04

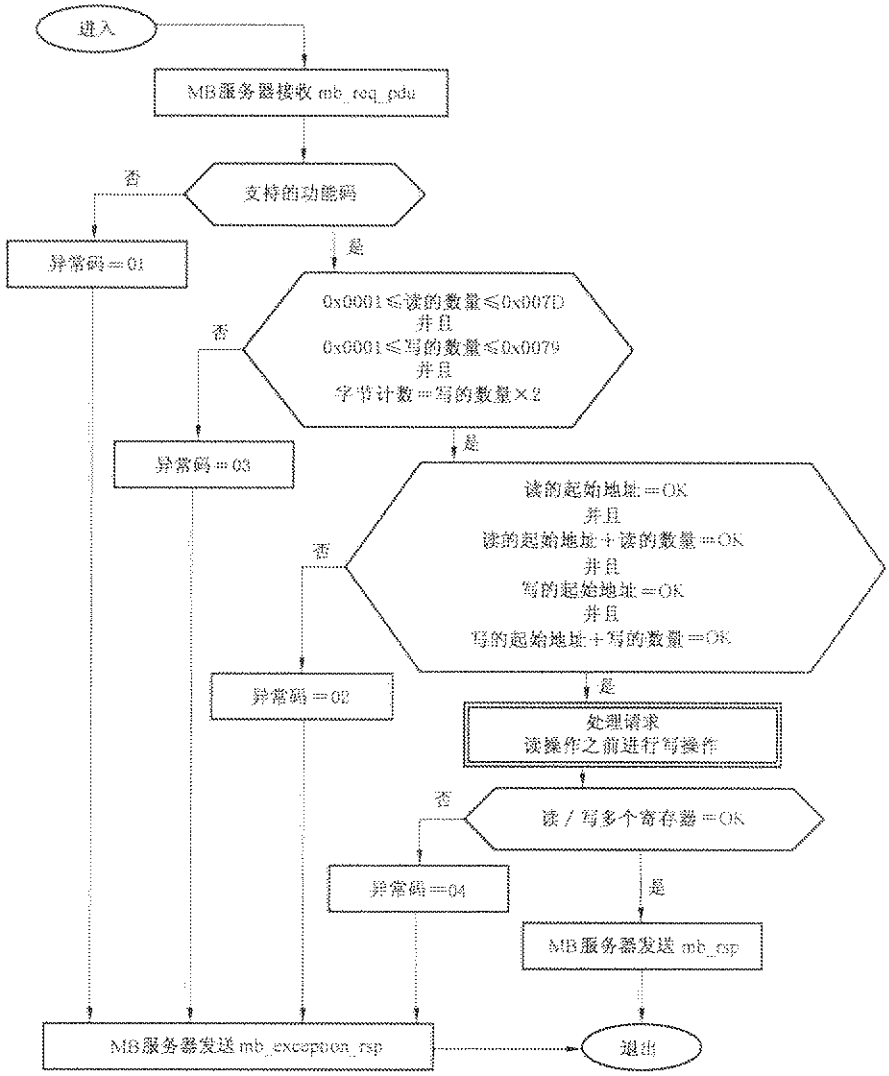


图 27　读/写多个寄存器状态图

表 71 是请求从第 4 个寄存器开始读 6 个寄存器并且从第 15 个寄存器开始写 3 个寄存器的示例。

表 71　从寄存器读和从寄存器写

请　求		响　应	
字段名	十六进制	字段名	十六进制
功能	17	功能	17
读起始地址 Hi	00	字节计数	0C
读起始地址 Lo	03	读寄存器值 Hi	00
读的数量 Hi	00	读寄存器值 Lo	FE
读的数量 Lo	06	读寄存器值 Hi	0A
写起始地址 Hi	00	读寄存器值 Lo	CD

表 71(续)

请求		响应	
字段名	十六进制	字段名	十六进制
写起始地址 Lo	0E	读寄存器值 Hi	00
写的数量 Hi	00	读寄存器值 Lo	01
写的数量 Lo	03	读寄存器值 Hi	00
写字节计数	06	读寄存器值 Lo	03
写寄存器值 Hi	00	读寄存器值 Hi	00
写寄存器值 Lo	FF	读寄存器值 Lo	0D
写寄存器值 Hi	00	读寄存器值 Hi	00
写寄存器值 Lo	FF	读寄存器值 Lo	FF
写寄存器值 Hi	00		
写寄存器值 Lo	FF		

7.18 24(0x18)读 FIFO 队列

见图 28 和表 72～表 74。

这个功能码允许读远程设备中的先入先出(FIFO)寄存器队列内容。这个功能码返回队列中寄存器的计数,随后为队列的数据。最多可以读 32 个寄存器:计数加上最多 31 个队列数据寄存器。首先返回队列计数寄存器,随后为队列数据寄存器。

这个功能码读队列内容,但不能清除队列内容。

在正常的响应中,字节计数表示随后的字节数量,包括队列计数字节和数值寄存器字节(不包括差错校验字段)。

队列计数是队列中数据寄存器的数量(不包括计数寄存器)。

如果队列计数大于 31,则返回一个含有异常码 03(非法数据值)的异常响应。

表 72 读 FIFO 队列请求

功能码	1 字节	0x18
FIFO 指针地址	2 字节	0x0000～0xFFFF

表 73 读 FIFO 队列响应

功能码	1 字节	0x18
字节计数	2 字节	…
FIFO 计数	2 字节	≤31
FIFO 数值计数	N[a]×2 字节	…
a N=FIFO 计数。		

表 74 读 FIFO 队列错误响应

异常功能码	1 字节	0x98
异常码	1 字节	01 或 02 或 03 或 04

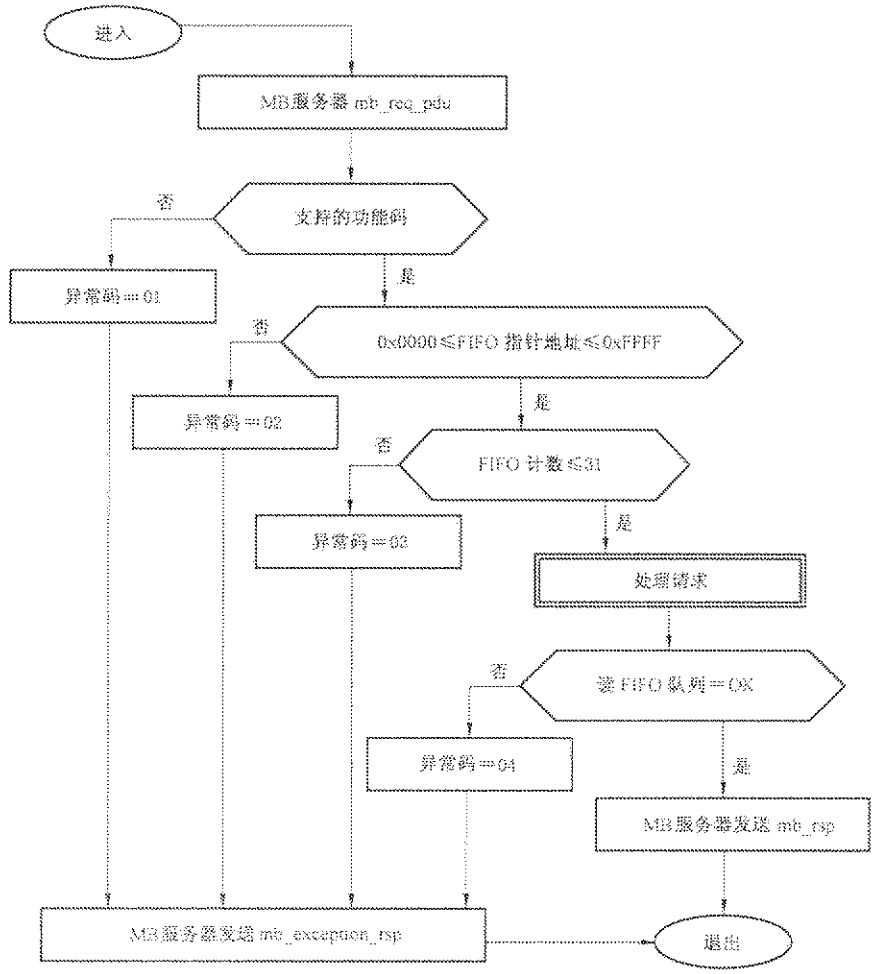


图 28　读 FIFO 队列状态图

表 75 是对远程设备读 FIFO 队列请求的示例。请求是读取以指针寄存器 1246(0x04DE)开始的队列。

表 75　对远程设备读 FIFO 队列请求

请　求		响　应	
字段名	十六进制	字段名	十六进制
功能	18	功能	18
FIFO 指针地址 Hi	04	字节计数 Hi	00
FIFO 指针地址 Lo	DE	字节计数 Lo	06
		FIFO 计数 Hi	00
		FIFO 计数 Lo	02
		FIFO 数值寄存器 Hi	01
		FIFO 数值寄存器 Lo	B8
		FIFO 数值寄存器 Hi	12
		FIFO 数值寄存器 Lo	84

在这个示例中，FIFO 指针寄存器(请求中的 1246)返回的内容为队列计数值 2。队列计数后面是两个数据寄存器。它们是 1247(十进制内容 440—0x01B8)和 1248(十进制内容 4740—0x1284)。

7.19　43(0x2B) 封装接口传输

见图 29 和表 76～表 78。

注：见附录 A。

功能码 43 及其用于设备标识的 MEI 类型 14 是本部分中当前可提供的两个封装接口传输方法之一。下面的功能码和 MEI 类型不属本部分的内容，这些功能码和 MEI 类型是专门保留的：43/0-12 和 43/15-255。

Modbus 封装接口(MEI)传输是一个在 Modbus PDU 内部将服务请求、方法调用和相关返回隧道化的机制。

MEI 传输的主要特征是封装属于已定义的接口组成部分的方法调用或服务请求，以及方法调用返回或服务响应。

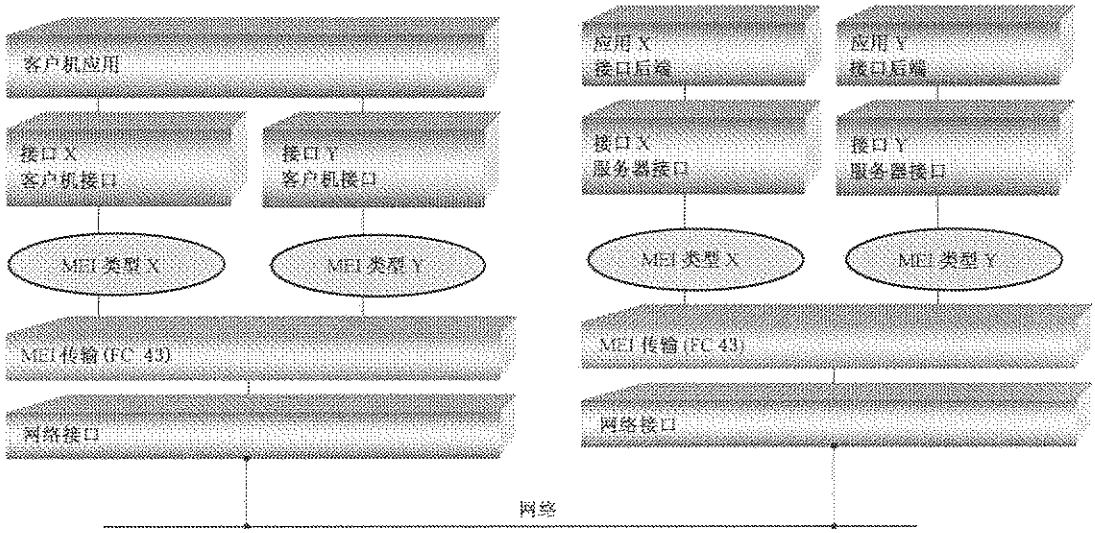


图 29 Modbus 封装接口传输

网络接口可以是用于发送 Modbus PDU 的任何通信栈，如 TCP/IP 或串行链路。

MEI 类型是 Modbus 指配的编号，因此，它是唯一的。根据附录 A，值 0～255 保留，MEI 类型 13 和 MEI 类型 14 除外。

MEI 传输实现使用 MEI 类型来分配对所指定接口的方法调用。

由于 MEI 传输服务是接口未知的，因此接口所需的任何特性或策略必须由该接口提供，例如：MEI 事务处理和 MEI 接口差错处理等。

表 76 封装接口传输请求

功能码	1 字节	0x2B
MEI 类型	1 字节	0x0D 或 0x0E
MEI 类型规定的数据	N 字节	…

表 77 封装接口传输响应

功能码	1 字节	0x2B
MEI 类型	1 字节	请求中的 MEI 类型的复制
MEI 类型规定的数据	N 字节	…

表 78 封装接口传输错误响应

异常功能码	1 字节	0xAB Fc 0x2B+0x80
异常码	1 字节	01 或 02 或 03 或 04

示例见读设备标识请求。

7.20 43/13(0x2B/0x0D) CANopen 通用引用请求和响应 PDU

CANopen 通用引用命令是服务的一种封装，这些服务被用来访问(读或写)CAN-Open 设备对象字典(CAN-Open Device Object Dictionary)的条目以及控制和监视 CANopen 系统和设备。

MEI 类型 13(0x0D)是 Modbus 分配给 CiA(CAN in Automation)，许可 CiA 用于 CANopen 通用引用的指定号码。

系统将工作于现有 Modbus 网络的限制内。因此，询问或者修改系统内对象字典的所需信息被映射到 Modbus 报文的格式中。在请求报文和响应报文中，PDU 被限制为 253 个字节。

注：关于 MEI 类型 13 的信息见附录 B。

7.21 43/14(0x2B/0x0E)读设备标识

见图 30 和表 79～表 82。

这个功能码允许读取与远程设备的物理和功能描述相关的标识和附加信息。

读设备标识接口由包含一组可寻址数据元素组成的地址空间构成。数据元素被称作对象，由对象 ID 识别它们。

接口由 3 类对象组成：

——基本设备标识。该类别的所有对象都是强制的：厂商名称、产品代码和修订版本号。

——常规设备标识。除基本数据对象以外，设备提供了附加的和可选择的标识以及数据对象描述。本部分定义了该类别的所有对象，但是它们的实现是可选的。

——扩展设备标识。除常规数据对象以外，设备提供了附加的和可选的标识以及关于物理设备本身的专用数据描述。所有这些数据都与设备有关。

表 79 读设备标识

对象 ID	对象名称/描述	类 型	强制的/可选的	类 别
0x00	厂商名称	ASCII 字符串	强制的	基本的
0x01	产品代码	ASCII 字符串	强制的	
0x02	主次版本号	ASCII 字符串	强制的	
0x03	厂商网址	ASCII 字符串	可选的	常规的
0x04	产品名称	ASCII 字符串	可选的	
0x05	型号名称	ASCII 字符串	可选的	
0x06	用户应用名称	ASCII 字符串	可选的	
0x07 … 0x7F	保留		可选的	
0x80 … 0xFF	可选择地定义专用对象 范围(0x80～0xFF)与产品有关	与设备相关	可选的	扩展的

表 80 读设备标识请求

功能码	1 字节	0x2B
MEI 类型	1 字节	0x0E
读设备 ID 码	1 字节	01/02/03/04
对象 ID	1 字节	0x00～0xFF

表 81 读设备标识响应

功能码	1 字节	0x2B
MEI 类型	1 字节	0x0E
读设备 ID 码	1 字节	01/02/03/04
一致性等级	1 字节	0x01 或 0x02 或 0x03 或 0x81 或 0x82 或 0x83
接续标识	1 字节	00/FF
下一个对象 ID	1 字节	对象 ID 号
对象数量	1 字节	…
列表		
对象 ID	1 字节	…
对象长度	1 字节	…
对象值	对象长度	与对象 ID 有关

表 82 读设备标识错误响应

功能码	1 字节	0xAB Fc0x2B+0x80
异常码	1 字节	01 或 02 或 03 或 04

请求参数描述：

Modbus 封装接口指配的编号 14 表示读标识请求。

参数“读设备 ID 码”允许定义四种访问类型：

01：请求获得基本设备标识(流访问)

02：请求获得常规设备标识(流访问)

03：请求获得扩展设备标识(流访问)

04：请求获得特定标识对象(单个访问)

如果读设备 ID 码是无效的，则在响应中返回异常码 03。

在设备标识不适用单独响应的情况下，需要进行多个事务处理(请求/响应)。对象 ID 字节给出了所要获得的第一个对象的标识。对于第一个事务处理来说，客户机必须设置对象 ID 为 0，以便获得设备标识数据的起始信息。对于随后事务处理来说，客户机必须将对象 ID 设置为服务器在此前响应中返回的值。

注：对象是不可分割的，所以任何对象的大小与事务处理响应大小一致。

如果对象 ID 不匹配任何已知对象，那么服务器的响应如同指向对象 0(从头开始)。

在单个访问的情况下：读设备 ID 码 04，请求中的对象 ID 给出了所要获得的对象标识。

如果在单个访问中对象 ID 不匹配任何已知对象，那么服务器返回一个异常码＝02(非法数据地址)的异常响应。

如果要求服务器设备高于其一致性等级的描述，则服务器设备必须根据其对应的实际等级响应。

响应参数描述：

功能码： 功能码 43(十进制)0x2B(十六进制)

MEI 类型： 为设备标识接口指配的编号 MEI 类型 14(0x0E)

读设备 ID 码： 与请求读设备 ID 码相同；01、02、03 或 04

一致性等级： 设备的标识一致性等级和支持的访问类型

0X01：基本标识(仅流访问)

0X02：常规标识(仅流访问)

0X03：扩展标识(仅流访问)

0X81：基本标识(流访问和单个访问)

0X82：常规标识(流访问和单个访问)
0X83：扩展标识(流访问和单个访问)

接续标识：　在读设备 ID 码 01、02 或 03(流访问)的情况下，
在标识数据不适用单独响应的情况下，需要进行多个事务处理(请求/响应)。
00：没有后续对象
FF：有后续标识对象，并且需要进一步的 Modbus 事务处理
在读设备 ID 码 04(单个访问)的情况下，
必须设置这个字段为 00。

下一个对象 ID：　如果“接续标识＝FF”，那么请求下一个对象 ID
如果“接续标识＝00”，那么必须设置为 00(无用的)

对象数目　在响应中返回的标识对象的数目(对于单个访问，对象数目＝1)

对象 0. ID　PDU 中返回的第一个对象的标识(流访问)或请求的对象的标识(单个访问)

对象 0. 长度　第一个对象的字节长度

对象 0. 值　第一个对象的值(对象 0. 长度字节)

…

对象 N. ID　最后对象的标识(在响应中)

对象 N. 长度　最后对象的字节长度

对象 N. 值　最后对象的值(对象 N. 长度字节)

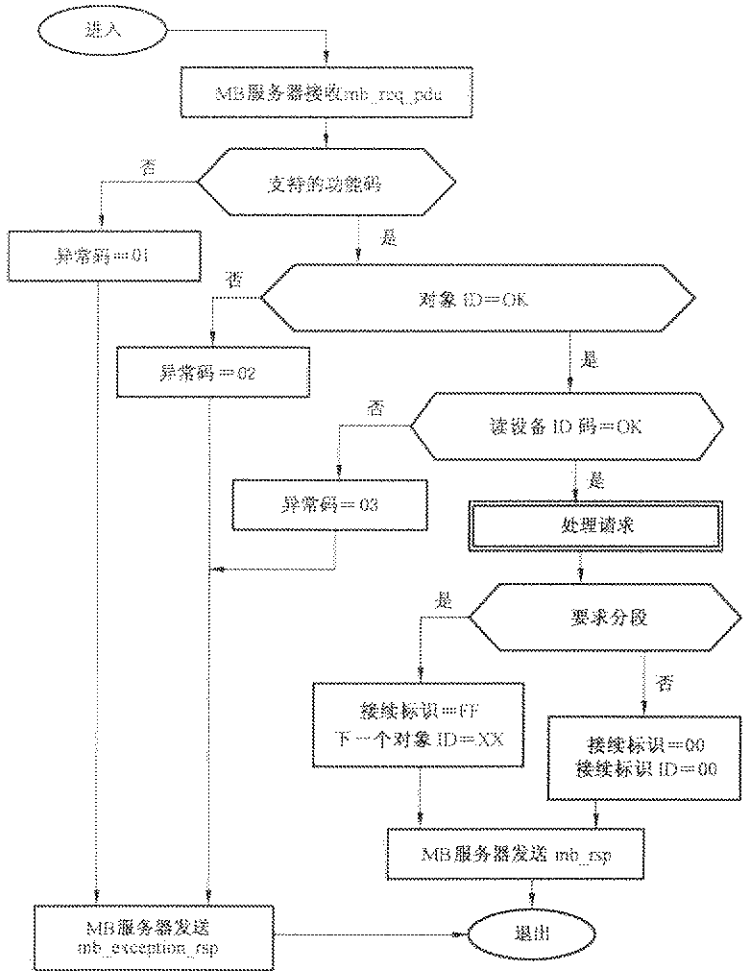


图 30　读设备标识状态图

“基本设备标识”的读设备标识请求的示例见表 83。在这个示例中，所有信息在一个响应 PDU 中发送。

表 83 读“基本设备标识”请求

请求		响应	
字段名	值	字段名	值
功能	2B	功能	2B
MEI 类型	0E	MEI 类型	0E
读设备 ID 码	01	读设备 ID 码	01
对象 ID	00	一致性等级	01
		接续标识	00
		下一个对象 ID	00
		对象数量	03
		对象 ID	00
		对象长度	16
		对象值	“Company identification”
		对象 ID	01
		对象长度	0D
		对象值	“product code XX”
		对象 ID	02
		对象长度	05
		对象值	“V2.11”

如果一个设备需要多个事务处理发送响应，那么启动随后事务处理。

第一个事务处理见表 84。

表 84 事务处理 1

请求		响应	
字段名	值	字段名	值
功能	2B	功能	2B
MEI 类型	0E	MEI 类型	0E
读设备 ID 码	01	读设备 ID 码	01
对象 ID	00	一致性等级	01
		接续标识	FF
		下一个对象 ID	02
		对象数量	03
		对象 ID	00
		对象长度	16
		对象值	“Company identification”
		对象 ID	01
		对象长度	1C
		对象值	“Product code ××××××××” (0x1C 个字符)

第二个事务处理见表 85。

表 85　事务处理 2

请　求		响　应	
字段名	值	字段名	值
功能	2B	功能	2B
MEI 类型	0E	MEI 类型	0E
读设备 ID 码	01	读设备 ID 码	01
对象 ID	02	一致性等级	01
		接续标识	00
		下一个对象 ID	00
		对象数量	03
		对象 ID	02
		对象长度	05
		对象值	"V2.11"

8　Modbus 异常响应

当客户机设备向服务器设备发送请求时，客户机希望得到一个正常的响应。主站的询问可能导致下列四种事件之一：

——如果服务器设备接收到无通信错误的请求，并且可以正常地处理询问，那么服务器设备将返回一个正常响应。

——如果由于通信错误服务器没有接收到请求，那么不能返回响应。客户机程序将按超时处理。

——如果服务器接收到请求，但是检测到一个通信错误（奇偶校验、LRC、CRC、...），那么不能返回响应。客户机程序将按超时处理。

——如果服务器接收到无通信错误的请求，但不能处理这个请求（例如，如果请求读一个不存在的输出或寄存器），那么服务器将返回一个异常响应，通知客户机出错误的原因。

异常响应报文有两个与正常响应不同的字段：

功能码字段：在正常响应中，服务器在响应的功能码字段复制原始请求的功能码。所有功能码的 MSB 都为 0（它们的值都低于十六进制 80）。在异常响应中，服务器设置功能码的 MSB 为 1。这使得异常响应中的功能码值比正常响应中的功能码值高十六进制 80。

通过设置功能码的 MSB，客户机的应用程序能够识别异常响应，并且能够检测异常码的数据字段。

数据字段：在正常的响应中，服务器可以在数据字段中返回数据或统计值（请求中要求的任何信息）。在异常响应中，服务器在数据字段中返回异常码。这说明了服务器产生异常的原因。

表 86 为客户机请求和服务器异常响应的示例。

表 86　客户机请求和服务器异常响应

请　求		响　应	
字段名	十六进制	字段名	十六进制
功能	01	功能	81
起始地址 Hi	04	异常码	02
起始地址 Lo	A1		
输出数量 Hi	00		
输出数量 Lo	01		

在这个示例中，客户机向服务器设备发出一个请求。功能码（01）用于读输出状态操作。它请求地

址为 1185(十六进制 04A1)的输出状态。根据输出字段(0001) 所指定的数量,只读出一个输出。

如果在服务器设备中不存在该输出地址,那么服务器将返回带有异常码(02)的异常响应。这就说明客户机指定的是非法的从站数据地址。

表 87 列出了异常码。

表 87 异常码

Modbus 异常码		
代码	名 称	含 义
01	非法功能	对于服务器(或从站)来说,询问中接收到的功能码是不允许的操作。这也许是因为功能码仅适用于新设备而在被选单元中没有实现;还可能表示服务器(或从站)在错误状态中处理这种请求,例如:因为它是未配置的,并且正被要求返回寄存器值
02	非法数据地址	对于服务器(或从站)来说,询问中接收到的数据地址是不允许的地址。特别是,寄存器编号和传输长度的组合是无效的。对于带有 100 个寄存器的控制器来说,PDU 赋值第一个寄存器为 0,最后一个为 99。如果起始寄存器编号 96 和寄存器数量为 4 的请求被处理,那么这个请求会成功操作于寄存器 96、97、98 和 99;而如果起始寄存器编号 96 和寄存器数量为 5 的请求被处理,那么将产生异常码 02"非法数据地址",因为它试图作用于寄存器 96、97、98、99 和 100,而寄存器 100 是不存在的
03	非法数据值	对于服务器(或从站)来说,询问数据字段中包含的是不允许的值。它表示组合请求中剩余部分结构方面的错误,例如,隐含长度不正确。它决不表示寄存器中被提交存储的数据项有一个应用程序期望之外的值,因为 Modbus 协议并不知道任何特殊寄存器的任何特殊值的具体含义
04	从站设备故障	当服务器(或从站)正在试图执行所请求的操作时,产生不可恢复的差错
05	确认	与编程命令一起使用。 服务器(或从站)已经接受请求,并且正在进行处理,但是需要较长的处理时间。返回这个响应以防止在客户机(或主站)中发生超时错误。客户机(或主站)可以继续发送轮询程序完成报文来确定是否完成处理
06	从站设备忙	与编程命令一起使用。 服务器(或从站)正在处理较长时间的程序命令。当服务器(或从站)空闲时,客户机(或主站)应该稍后重新传送报文
08	存储奇偶性差错	与功能码 20、21 和引用类型 6 一起使用,以指示扩展文件区不能通过一致性校验; 服务器(或从站)试图读记录文件,但是在存储器中发现一个奇偶校验错误。客户机(或主站)可以重新发送请求,但可能需要在服务器(或从站)设备上提供这种服务
0A	网关路径不可用	与网关一起使用。它表示网关不能为处理请求分配输入端口至输出端口的内部通信路径。通常表示网关配置错误或过载
0B	网关目标设备响应失败	与网关一起使用。它表示没有从目标设备中获得响应。通常表示设备未在网络中

附 录 A
（资料性附录）
Modbus 保留的功能码、子码及 MEI 类型

下列功能码和子码不属本部分的内容，这些功能码和子码是特殊保留的。格式是功能码/子码，或仅有功能码，其所有子码(0～255)均被保留：8/19、8/21～65 535、9、10、13、14、41、42、90、91、125、126和 127。

功能码 43 中用于设备标识的 MEI 类型 14，以及用于 CANopen 通用引用请求和响应 PDU 的 MEI 类型 13 都是本部分中目前可用的封装接口传输。

下列功能码和 MEI 类型不属本部分的内容，这些功能码和 MEI 类型是特殊保留的：43/0～12 和 43/15～255。在本部分中，不支持具有与封装接口传输相同或类似结果的用户定义的功能码。

附 录 B
（资料性附录）
CANopen 通用引用命令

关于功能码 43 和 MEI 类型 13 的用法，请访问 MODBUS-IDA 网站或 CiA(CAN in Automation)网站。

参 考 文 献

[1] ANSI/EIA/TIA-232-E,1997 interface between data terminal equipment and data circuit-terminating equipment employing serial binary data interchang.

[2] ANSI/EIA/TIA-485-A,1998 electrical characteristics of generators and recievers for use in balanced digital multipoint systems.

[3] MODBUS-IDA.org Modbus application protocol specification.

中华人民共和国
国家标准
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/T 19582.1—2008
*
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.spc.net.cn
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 97 千字
2008年5月第一版 2008年5月第一次印刷
*
书号:155066·1-31328 定价 36.00 元

GB/T 19582.1-2008