MODBUS 应用协议规范

MODBUS Application Protocol Specification V1.1b3

版本：V1.1b3

日期：2012年4月26日

中文翻译版 | www.modbus.cn

本文件为 Modbus.org
官方规范文档的中文翻译版，仅供学习参考之用。官方英文原版请访问 www.modbus.org 获取。如翻译内容与英文原版存在歧义，以英文原版为准。

1 引言

1.1 文档范围

MODBUS 是位于 OSI 模型第 7 层的应用层消息协议，为连接在不同类型总线或网络上的设备之间提供客户端/服务器端通信。

自 1979 年以来，MODBUS 一直是工业串行通信的事实标准，持续为数百万自动化设备提供通信支持。如今，对 MODBUS 简洁而优雅的结构的支持仍在不断增长。互联网社区可以通过 TCP/IP 协议栈上预留的系统端口 502 访问 MODBUS。

MODBUS 是一种请求/应答协议，并提供由功能码指定的服务。MODBUS 功能码是 MODBUS 请求/应答 PDU 的元素。本文档的目标是描述在 MODBUS 事务框架内使用的功能码。

MODBUS 是用于在连接在不同类型总线或网络上的设备之间进行客户端/服务器端通信的应用层消息协议。目前通过以下方式实现：

- TCP/IP over Ethernet，参见 MODBUS 消息实现指南 V1.0a
- 通过各种介质的异步串行传输（线缆：EIA/TIA-232-E、EIA-422、EIA/TIA-485-A；光纤、无线电等）
- MODBUS PLUS，一种高速令牌传递网络

图 1：MODBUS 通信协议栈

参考文献：

1. RFC 791, Internet Protocol, Sep81 DARPA

2 缩略语

缩略语	全称	中文
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	互联网工程任务组
I/O	Input/Output	输入/输出
IP	Internet Protocol	互联网协议
MAC	Media Access Control	媒体访问控制
MB	MODBUS Protocol	MODBUS 协议
MBAP	MODBUS Application Protocol	MODBUS 应用协议
PDU	Protocol Data Unit	协议数据单元
PLC	Programmable Logic Controller	可编程逻辑控制器
TCP	Transmission Control Protocol	传输控制协议
CRC	Cyclic Redundancy Check	循环冗余校验
LSB	Least Significant Bit	最低有效位
MSB	Most Significant Bit	最高有效位
MEI	MODBUS Encapsulated Interface	MODBUS 封装接口

3 背景

MODBUS 协议允许在所有类型的网络架构中轻松通信。各种类型的设备（PLC、HMI、控制面板、驱动器、运动控制、I/O 设备等）都可以使用 MODBUS 协议发起远程操作。相同的通信既可以在串行线路上完成，也可以在以太网 TCP/IP 网络上完成。网关允许使用 MODBUS 协议在多种类型的总线或网络之间进行通信。

图 2：MODBUS 网络架构示例

4 概述

4.1 协议描述

MODBUS 协议定义了一个简单的协议数据单元（PDU），与底层通信层无关。将 MODBUS 协议映射到特定总线或网络时，可能会在应用数据单元（ADU）上引入一些附加字段。

图 3：通用 MODBUS 帧

```
□□□□ + □□□ + □□ + □□□□ = ADU □□□ + □□ = PDU
```

MODBUS 应用数据单元由发起 MODBUS 事务的客户端构建。功能码向服务器端指示要执行的操作类型。MODBUS 应用协议规定了客户端发起的请求的格式。

MODBUS 数据单元的功能码字段以一个字节编码。有效码值范围为 1 ... 255 十进制（128 - 255 范围保留用于异常响应）。当消息从客户端设备发送到服务器端设备时，功能码字段告诉服务器端要执行什么操作。功能码 '0' 无效。

某些功能码添加了子功能码以定义多个操作。从客户端发送到服务器端设备的消息数据字段包含服务器端用于执行功能码所定义操作的附加信息。这可以包括离散量和寄存器地址、要处理的项目数量以及字段中实际数据字节的计数。

在某些类型的请求中，数据字段可以不存在（长度为零），在这种情况下，服务器端不需要任何附加信息，仅功能码就指定了操作。

如果在正确接收的 MODBUS ADU 中，请求的 MODBUS 功能没有发生错误，则从服务器端到客户端的响应的数据字段包含所请求的数据。如果请求的 MODBUS 功能发生错误，则该字段包含一个异常码，服务器端应用程序可以使用该异常码来确定接下来要采取的操作。

例如，客户端可以读取一组离散输出或输入的 ON/OFF 状态，或者读取/写入一组寄存器的数据内容。

当服务器端响应客户端时，它使用功能码字段来指示正常（无错误）响应或发生了某种错误（称为异常响应）。对于正常响应，服务器端只需将原始功能码回显给请求。

图 4：MODBUS 事务（无错误）

对于异常响应，服务器端返回一个代码，该代码等于请求 PDU 中的原始功能码，并将其最高有效位设置为逻辑 1。

图 5：MODBUS 事务（异常响应）

注意：建议管理超时机制，以避免无限期地等待可能永远不会到达的应答。

MODBUS PDU 的大小受到第一个 MODBUS 串行线路网络实现的大小限制（最大 RS485 ADU = 256 字节）。因此：

```
MODBUS PDU□□□□□□□□□= 256 − □□□□□□□1 □□□− CRC□2 □□□= 253 □□□ □□□ RS232 / RS485 ADU = 253 □□ + □□□□□□□1 □□□+ CRC□2 □□□= 256 □□□ TCP MODBUS ADU = 253 □□ + MBAP□7 □□□= 260 □□□
```

MODBUS 协议定义了三种 PDU：

- MODBUS 请求 PDU，mb_req_pdu
- MODBUS 响应 PDU，mb_rsp_pdu
- MODBUS 异常响应 PDU，mb_excep_rsp_pdu

```
mb_req_pdu □□□□ mb_req_pdu = {function_code, request_data} □□□ function_code = [1 □□] MODBUS □□□ request_data = [n □□] □□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□
```

```
mb_rsp_pdu □□□□ mb_rsp_pdu = {function_code, response_data} □□□ function_code = [1 □□] MODBUS □□□ response_data = [n □□] □□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□
```

```
mb_excep_rsp_pdu □□□□ mb_excep_rsp_pdu = {exception−function_code, exception_code} □□□ exception−function_code = [1 □□] MODBUS □□□ + 0x80
exception_code = [1 □□] MODBUS □□□□□□□ 7 □□□□
```

4.2 数据编码

MODBUS 使用大端序（Big-Endian）表示地址和数据项。这意味着当传输大于单个字节的数值时，最高有效字节首先发送。例如：

寄存器大小	值	传输顺序
16 位	0x1234	先发送 0x12，然后 0x34

注意：更多细节请参见 [1]。

4.3 MODBUS 数据模型

MODBUS 基于一系列具有不同特征的表来构建其数据模型。四个主要表为：

主要表	对象类型	访问类型	说明
离散量输入	单个位	只读	此类数据可由 I/O 系统提供
线圈	单个位	读写	此类数据可由应用程序修改
输入寄存器	16 位字	只读	此类数据可由 I/O 系统提供
保持寄存器	16 位字	读写	此类数据可由应用程序修改

输入与输出之间，以及位寻址与字寻址数据项之间的区别并不暗示任何应用行为。将所有四个表视为彼此叠加是完全可接受的，并且非常常见——如果这是目标机器上最自然的解释。

对于每个主要表，协议允许单独选择 65536 个数据项，并且对这些项的读写操作设计为跨越多个连续数据项，最大数据大小取决于事务功能码。

显然，通过 MODBUS 处理的所有数据（位、寄存器）必须位于设备应用程序内存中。但内存中的物理地址不应与数据引用混淆。唯一的要求是将数据引用与物理地址关联起来。

MODBUS 逻辑引用号（在 MODBUS 功能中使用）是从零开始的无符号整数索引。

图 6：MODBUS 数据模型（独立块）

图 7：MODBUS 数据模型（单一数据块）

4.4 MODBUS 寻址模型

MODBUS 应用协议精确定义了 PDU 寻址规则。在 MODBUS PDU 中，每个数据寻址范围为 0 到 65535。它明确定义了一个由 4 个块组成的 MODBUS 数据模型，每个块包含编号从 1 到 n 的元素。

在 MODBUS 数据模型中，数据块中的每个元素编号为从 1 到 n。随后，MODBUS 数据模型必须绑定到设备应用程序（IEC-61131 对象或其他应用模型）。MODBUS 数据模型与设备应用程序之间的预映射完全由供应商设备特定。

图 8：MODBUS 寻址模型

4.5 定义 MODBUS 事务

以下状态图描述了 MODBUS 事务在服务器端的通用处理流程。一旦服务器端处理了请求，将使用适当的 MODBUS 服务器端事务构建 MODBUS 响应。根据处理结果，构建两种类型的响应：

- 正 MODBUS 响应：响应功能码 = 请求功能码
- MODBUS 异常响应（见第 7 章）：
  - 目的是向客户端提供在处理过程中检测到的错误的相关信息；
  - 异常功能码 = 请求功能码 + 0x80；
  - 提供异常码以指示错误的原因。

图 9：MODBUS 事务状态图

5 功能码分类

MODBUS 功能码分为三类：

公共功能码（Public Function Codes）：

- 是明确定义的功能码
- 保证唯一
- 由 MODBUS.org 社区验证
- 公开记录
- 有可用的一致性测试
- 包括已定义的公共分配功能码以及保留供将来使用的未分配功能码

用户定义功能码（User-Defined Function Codes）：

- 有两个用户定义功能码范围，即 65 到 72 十进制和 100 到 110 十进制
- 用户可以选择并实现规范不支持的功能码
- 不保证所选功能码的使用是唯一的
- 如果用户想将功能重新定位为公共功能码，必须发起 RFC 以将更改引入公共类别并分配新的公共功能码
- MODBUS Organization, Inc 明确保留制定所提议 RFC 的权利

保留功能码（Reserved Function Codes）：

- 某些公司当前用于遗留产品的功能码，不供公共使用
- 说明性注释：读者请参阅附录 A（资料性）MODBUS 保留功能码、子码和 MEI 类型

图 10：MODBUS 功能码分类

5.1 公共功能码定义

功能码	子码	十六进制	章节	数据访问	说明
位访问					
读离散输入	02	02	6.2	物理离散输入	读离散输入
读线圈	01	01	6.1	内部位或物理线圈	读线圈
写单个线圈	05	05	6.5		写单个线圈
写多个线圈	15	0F	6.11		写多个线圈
16 位访问					
读输入寄存器	04	04	6.4	物理输入寄存器	读输入寄存器
读保持寄存器	03	03	6.3	内部寄存器或物理输出寄存器	读保持寄存器
写单个寄存器	06	06	6.6		写单个寄存器
写多个寄存器	16	10	6.12		写多个寄存器
读/写多个寄存器	23	17	6.17		读写多个寄存器
掩码写寄存器	22	16	6.16		掩码写寄存器
读 FIFO 队列	24	18	6.18		读 FIFO 队列
文件记录访问					
读文件记录	20	14	6.14		读文件记录
写文件记录	21	15	6.15		写文件记录
诊断					
读异常状态	07	07	6.7		读异常状态
诊断	08	00-18,20	08	6.8	诊断
获取通信事件计数器	11	0B	6.9		获取通信事件计数器

功能码	子码	十六进制	章节	数据访问	说明
获取通信事件日志	12	0C	6.10		获取通信事件日志
报告服务器 ID	17	11	6.13		报告服务器 ID
读设备标识	43	14	2B	6.21	读设备标识
其他					
封装接口传输	43	13,14	2B	6.19	封装接口传输
CANopen 通用参考	43	13	2B	6.20	CANopen 通用参考

6 功能码描述

6.1 01 (0x01) 读线圈（Read Coils）

此功能码用于读取远程设备中 1 到 2000 个连续线圈的状态。请求 PDU 指定起始地址（即第一个线圈的地址）和线圈数量。在 PDU 中，线圈寻址从零开始。因此编号为 1-16 的线圈寻址为 0-15。

响应消息中的线圈以数据字段的每比特一个线圈的方式打包。状态指示：1 = ON，0 = OFF。第一个数据字节的 LSB 包含查询中寻址的输出。其他线圈按该字节的高位方向排列，并在后续字节中从低位到高位排列。

如果返回的输出数量不是 8 的倍数，最后一个数据字节中的剩余位将用零填充（朝字节高位方向）。字节计数字段指定完整数据字节的数量。

请求格式：

字段	长度	范围
功能码	1 字节	0x01
起始地址	2 字节	0x0000 至 0xFFFF
线圈数量	2 字节	1 至 2000 (0x7D0)

响应格式：

字段	长度	范围
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，如果余数不为 0 则 N = N+1

错误格式：

字段	长度	范围
错误码	1 字节	功能码 + 0x80 (0x81)
异常码	1 字节	01 或 02 或 03 或 04

示例：读取离散输出 20-38

请求		响应	
字段名	(Hex)	字段名	(Hex)
功能码	01	功能码	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

输出 27-20 的状态显示为字节值 CD hex，即二进制 1100 1101。输出 27 是该字节的 MSB，输出 20 是 LSB。依惯例，字节内的位以 MSB 在左、LSB 在右的方式显示。因此第一个字节中的输出按从左到右的顺序为 '27 到 20'。下一个字节中的输出从左到右为 '35 到 28'。位串行传输时从 LSB 到 MSB 流动：20 ... 27, 28 ... 35，依此类推。

在最后一个数据字节中，输出 38-36 的状态显示为字节值 05 hex，即二进制 0000 0101。输出 38 位于从左边数第六位，而输出 36 是该字节的 LSB。其余五个高位用零填充。

图 11：读线圈状态图

6.2 02 (0x02) 读离散输入（Read Discrete Inputs）

此功能码用于读取远程设备中 1 到 2000 个连续离散输入的状态。请求 PDU 指定起始地址（即第一个输入的地址）和输入数量。在 PDU 中，离散输入寻址从零开始。因此编号为 1-16 的离散输入寻址为 0-15。

响应消息中的离散输入以数据字段的每比特一个输入的方式打包。状态指示：1 = ON，0 = OFF。第一个数据字节的 LSB 包含查询中寻址的输入。其他输入按该字节的高位方向排列，并在后续字节中从低位到高位排列。如果返回的输入数量不是 8 的倍数，最后一个数据字节中的剩余位将用零填充（朝字节高位方向）。字节计数字段指定完整数据字节的数量。

请求格式：

字段	长度	范围
功能码	1 字节	0x02
起始地址	2 字节	0x0000 至 0xFFFF
输入数量	2 字节	1 至 2000 (0x7D0)

响应格式：

字段	长度	范围
功能码	1 字节	0x02
字节计数	1 字节	N*
输入状态	N* x 1 字节	

* N = 输入数量 / 8，如果余数不为 0 则 N = N+1

错误格式：

字段	长度	范围
错误码	1 字节	0x82
异常码	1 字节	01 或 02 或 03 或 04

示例：读取离散输入 197-218

请求		响应	
字段名	(Hex)	字段名	(Hex)
功能码	02	功能码	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
输入数量 Hi	00	输入状态 212-205	DB
输入数量 Lo	16	输入状态 218-213	35

离散输入 204-197 的状态显示为字节值 AC hex，即二进制 1010 1100。输入 204 是该字节的 MSB，输入 197 是 LSB。离散输入 218-213 的状态显示为字节值 35 hex，即二进制 0011 0101。输入 218 位于从左边数第三位，输入 213 是 LSB。注意：其余两位（朝高位方向）用零填充。

图 12：读离散输入状态图

6.3 03 (0x03) 读保持寄存器（Read Holding Registers）

此功能码用于读取远程设备中连续保持寄存器块的内容。请求 PDU 指定起始寄存器地址和寄存器数量。在 PDU 中，寄存器寻址从零开始。因此编号为 1-16 的寄存器寻址为 0-15。

响应消息中的寄存器数据以每个寄存器两个字节的方式打包，二进制内容在每个字节内右对齐。对于每个寄存器，第一个字节包含高位，第二个包含低位。

请求格式：

字段	长度	范围
功能码	1 字节	0x03
起始地址	2 字节	0x0000 至 0xFFFF
寄存器数量	2 字节	1 至 125 (0x7D)

响应格式：

字段	长度	范围
功能码	1 字节	0x03
字节计数	1 字节	2 x N*
寄存器值	N* x 2 字节	

* N = 寄存器数量

错误格式：

字段	长度	范围
错误码	1 字节	0x83
异常码	1 字节	01 或 02 或 03 或 04

示例：读取寄存器 108-110

请求		响应	
字段名	(Hex)	字段名	(Hex)
功能码	03	功能码	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

寄存器 108 的内容显示为两个字节值 02 2B hex，即 555 十进制。寄存器 109-110 的内容分别为 00 00 和 00 64 hex，即 0 和 100 十进制。

图 13：读保持寄存器状态图

6.4 04 (0x04) 读输入寄存器（Read Input Registers）

此功能码用于读取远程设备中 1 到 125 个连续输入寄存器。请求 PDU 指定起始寄存器地址和寄存器数量。在 PDU 中，寄存器寻址从零开始。因此编号为 1-16 的输入寄存器寻址为 0-15。

响应消息中的寄存器数据以每个寄存器两个字节的方式打包，二进制内容在每个字节内右对齐。对于每个寄存器，第一个字节包含高位，第二个包含低位。

请求格式：

字段	长度	范围
功能码	1 字节	0x04
起始地址	2 字节	0x0000 至 0xFFFF
输入寄存器数量	2 字节	0x0001 至 0x007D

响应格式：

字段	长度	范围
功能码	1 字节	0x04
字节计数	1 字节	2 x N*
输入寄存器	N* x 2 字节	

* N = 输入寄存器数量

错误格式：

字段	长度	范围
错误码	1 字节	0x84
异常码	1 字节	01 或 02 或 03 或 04

示例：读取输入寄存器 9

请求		响应	
字段名	(Hex)	字段名	(Hex)
功能码	04	功能码	04
起始地址 Hi	00	字节计数	02
起始地址 Lo	08	输入寄存器 9 Hi	00
输入寄存器数量 Hi	00	输入寄存器 9 Lo	0A
输入寄存器数量 Lo	01		

输入寄存器 9 的内容显示为两个字节值 00 0A hex，即 10 十进制。

图 14：读输入寄存器状态图

6.5 05 (0x05) 写单个线圈（Write Single Coil）

此功能码用于将远程设备中的单个输出写入 ON 或 OFF 状态。请求的 ON/OFF 状态由请求数据字段中的常量指定。值 0xFF00 请求输出为 ON，值 0x0000 请求输出为 OFF。所有其他值均为非法值，不会影响输出。

请求 PDU 指定要强制的线圈地址。线圈寻址从零开始。因此编号为 1 的线圈寻址为 0。

正常响应是请求的回显，在线圈状态写入后返回。

请求格式：

字段	长度	范围
功能码	1 字节	0x05
输出地址	2 字节	0x0000 至 0xFFFF
输出值	2 字节	0x0000 或 0xFF00

响应格式：

字段	长度	范围
功能码	1 字节	0x05
输出地址	2 字节	0x0000 至 0xFFFF
输出值	2 字节	0x0000 或 0xFF00

错误格式：

字段	长度	范围
错误码	1 字节	0x85
异常码	1 字节	01 或 02 或 03 或 04

示例：将线圈 173 写入 ON

请求		响应	
字段名	(Hex)	字段名	(Hex)
功能码	05	功能码	05
输出地址 Hi	00	输出地址 Hi	00
输出地址 Lo	AC	输出地址 Lo	AC
输出值 Hi	FF	输出值 Hi	FF
输出值 Lo	00	输出值 Lo	00

图 15：写单个输出状态图

6.6 06 (0x06) 写单个寄存器（Write Single Register）

此功能码用于写入远程设备中的单个保持寄存器。请求 PDU
指定要写入的寄存器地址。寄存器寻址从零开始。因此编号为 1 的寄存器寻址为 0。

正常响应是请求的回显，在寄存器内容写入后返回。

请求格式：

字段	长度	范围
功能码	1 字节	0x06
寄存器地址	2 字节	0x0000 至 0xFFFF
寄存器值	2 字节	0x0000 至 0xFFFF

响应格式：

字段	长度	范围
功能码	1 字节	0x06
寄存器地址	2 字节	0x0000 至 0xFFFF
寄存器值	2 字节	0x0000 至 0xFFFF

错误格式：

字段	长度	范围
错误码	1 字节	0x86
异常码	1 字节	01 或 02 或 03 或 04

示例：将寄存器 2 写入 00 03 hex

请求		响应	
字段名	(Hex)	字段名	(Hex)
功能码	06	功能码	06
寄存器地址 Hi	00	寄存器地址 Hi	00
寄存器地址 Lo	01	寄存器地址 Lo	01
寄存器值 Hi	00	寄存器值 Hi	00
寄存器值 Lo	03	寄存器值 Lo	03

图 16：写单个寄存器状态图

6.7 07 (0x07) 读异常状态（Read Exception Status，仅串行线路）

此功能码用于读取远程设备中八个异常状态输出的内容。该功能提供了一种简单的方法来访问此信息，因为异常输出引用是已知的（功能中不需要输出引用）。

正常响应包含八个异常状态输出的状态。输出被打包到一个数据字节中，每个输出占一个位。最低输出引用的状态包含在字节的最低有效位中。八个异常状态输出的内容是设备特定的。

请求格式：

字段	长度	范围
功能码	1 字节	0x07

响应格式：

字段	长度	范围
功能码	1 字节	0x07
输出数据	1 字节	0x00 至 0xFF

错误格式：

字段	长度	范围
错误码	1 字节	0x87
异常码	1 字节	01 或 04

示例：读取异常状态

请求		响应	
字段名	(Hex)	字段名	(Hex)
功能码	07	功能码	07
		输出数据	6D

在此示例中，输出数据为 6D hex（0110 1101 二进制）。从左到右，输出状态为 OFF-ON-ON-OFF-ON-ON-OFF-ON。状态从最高寻址输出到最低寻址输出显示。

图 17：读异常状态状态图

6.8 08 (0x08) 诊断（Diagnostics，仅串行线路）

MODBUS 功能码 08 提供一系列测试，用于检查客户端设备和服务器端之间的通信系统，或检查服务器端内部的各种错误条件。该功能在查询中使用两个字节的子功能码字段来定义要执行的测试类型。服务器端在正常响应中回显功能码和子功能码。某些诊断会导致数据在正常响应的数据字段中从远程设备返回。

通常，向远程设备发出诊断功能不会影响远程设备中用户程序的运行。诊断功能不访问用户逻辑，如离散量和寄存器。某些功能可以选择性地重置远程设备中的错误计数器。

但是，服务器端设备可以强制进入'仅监听模式'（Listen Only Mode），在此模式下，它将监控通信系统上的消息但不对其作出响应。如果您的应用程序依赖于与远程设备的任何进一步数据交换，这可能会影响其运行结果。通常，该模式用于从通信系统中移除故障远程设备。

请求格式：

字段	长度	范围
功能码	1 字节	0x08
子功能码	2 字节	
数据	N x 2 字节	

响应格式：

字段	长度	范围
功能码	1 字节	0x08
子功能码	2 字节	
数据	N x 2 字节	

错误格式：

字段	长度	范围
错误码	1 字节	0x88
异常码	1 字节	01 或 03 或 04

6.8.1 串行线路设备支持的子功能码

子功能码 (Hex)	子功能码 (Dec)	名称	中文
00	00	Return Query Data	返回查询数据
01	01	Restart Communications Option	重启通信选项
02	02	Return Diagnostic Register	返回诊断寄存器
03	03	Change ASCII Input Delimiter	更改 ASCII 输入分隔符
04	04	Force Listen Only Mode	强制仅监听模式
05..09		RESERVED	保留
0A	10	Clear Counters and Diagnostic Register	清除计数器和诊断寄存器
0B	11	Return Bus Message Count	返回总线消息计数
0C	12	Return Bus Communication Error Count	返回总线通信错误计数
0D	13	Return Bus Exception Error Count	返回总线异常错误计数
0E	14	Return Server Message Count	返回服务器消息计数
0F	15	Return Server No Response Count	返回服务器无响应计数
10	16	Return Server NAK Count	返回服务器 NAK 计数
11	17	Return Server Busy Count	返回服务器忙计数
12	18	Return Bus Character Overrun Count	返回总线字符溢出计数
13	19	RESERVED	保留
14	20	Clear Overrun Counter and Flag	清除溢出计数器和标志
N.A.	21..65535	RESERVED	保留

00 返回查询数据（Return Query Data）

请求数据字段中传递的数据将在响应中返回（环回）。整个响应消息应与请求相同。

子功能码	数据字段（请求）	数据字段（响应）
00 00	任意	回显请求数据

01 重启通信选项（Restart Communications Option）

远程设备串行线路端口必须初始化和重启，所有通信事件计数器都被清除。如果端口当前处于仅监听模式，则不返回响应。此功能是唯一能将端口从仅监听模式中恢复的功能。如果端口当前不处于仅监听模式，则返回正常响应。这在重启执行之前发生。当远程设备收到请求时，它尝试重启并执行其上电信心测试。成功完成测试将使端口上线。请求数据字段内容 FF 00 hex 也会导致端口的通信事件日志被清除。内容 00 00 则使日志保持重启前的状态。

子功能码	数据字段（请求）	数据字段（响应）
00 01	00 00	回显请求数据
00 01	FF 00	回显请求数据

02 返回诊断寄存器（Return Diagnostic Register）

远程设备 16 位诊断寄存器的内容在响应中返回。

子功能码	数据字段（请求）	数据字段（响应）
00 02	00 00	诊断寄存器内容

03 更改 ASCII 输入分隔符（Change ASCII Input Delimiter）

请求数据字段中传递的字符 'CHAR' 成为未来消息的消息结束分隔符（替换默认的 LF 字符）。此功能在 ASCII 消息末尾不需要换行符的情况下很有用。

子功能码	数据字段（请求）	数据字段（响应）
00 03	CHAR 00	回显请求数据

04 强制仅监听模式（Force Listen Only Mode）

强制寻址的远程设备进入 MODBUS 通信的仅监听模式。这将其与网络上的其他设备隔离，允许它们继续通信而不受寻址远程设备的干扰。不返回响应。当远程设备进入仅监听模式时，所有活动通信控制都被关闭。就绪看门狗定时器被允许超时，锁定控制关闭。当设备处于此模式时，任何寻址到它或广播的 MODBUS 消息都会被监控，但不会采取任何操作，也不会发送任何响应。进入模式后唯一会被处理的功能是重启通信选项功能（功能码 8，子功能 1）。

子功能码	数据字段（请求）	数据字段（响应）
00 04	00 00	不返回响应

10 (0A Hex) 清除计数器和诊断寄存器

目标是清除所有计数器和诊断寄存器。计数器在上电时也会被清除。

11 (0B Hex) 返回总线消息计数

响应数据字段返回远程设备自上次重启、清除计数器操作或上电以来在通信系统上检测到的消息数量。

12 (0C Hex) 返回总线通信错误计数

响应数据字段返回远程设备自上次重启、清除计数器操作或上电以来遇到的 CRC 错误数量。

13 (0D Hex) 返回总线异常错误计数

响应数据字段返回远程设备自上次重启、清除计数器操作或上电以来返回的 MODBUS 异常响应数量。异常响应在第 7 章中描述和列出。

14 (0E Hex) 返回服务器消息计数

响应数据字段返回寻址到远程设备或广播的、远程设备自上次重启、清除计数器操作或上电以来已处理的消息数量。

15 (0F Hex) 返回服务器无响应计数

响应数据字段返回寻址到远程设备且远程设备自上次重启、清除计数器操作或上电以来未返回响应（既未返回正常响应也未返回异常响应）的消息数量。

16 (10 Hex) 返回服务器 NAK 计数

响应数据字段返回寻址到远程设备且远程设备自上次重启、清除计数器操作或上电以来返回了否定确认（NAK）异常响应的消息数量。

17 (11 Hex) 返回服务器忙计数

响应数据字段返回寻址到远程设备且远程设备自上次重启、清除计数器操作或上电以来返回了服务器设备忙异常响应的消息数量。

18 (12 Hex) 返回总线字符溢出计数

响应数据字段返回寻址到远程设备但由于字符溢出条件而无法处理的消息数量——自上次重启、清除计数器操作或上电以来。字符溢出是由数据字符到达端口的速度快于其存储速度，或由于硬件故障导致字符丢失引起的。

20 (14 Hex) 清除溢出计数器和标志

清除溢出错误计数器并重置错误标志。

6.8.2 示例和状态图

以下是向远程设备请求返回查询数据的示例。这使用子功能码零（两字节字段中 00 00 hex）。要返回的数据在两字节数据字段（A5 37 hex）中发送。

请求		响应	
字段名	(Hex)	字段名	(Hex)
功能码	08	功能码	08
子功能码 Hi	00	子功能码 Hi	00
子功能码 Lo	00	子功能码 Lo	00
数据 Hi	A5	数据 Hi	A5
数据 Lo	37	数据 Lo	37

图 18：诊断状态图

6.9 11 (0x0B) 获取通信事件计数器（Get Comm Event Counter，仅串行线路）

此功能码用于从远程设备的通信事件计数器中获取状态字和事件计数。通过在发送一系列消息之前和之后获取当前计数，客户端可以确定这些消息是否被远程设备正常处理。

设备的事件计数器在每次成功完成消息后递增一次。异常响应、轮询命令或获取事件计数器命令不会使其递增。事件计数器可以通过诊断功能（代码 08）使用重启通信选项子功能（代码 00 01）或清除计数器和诊断寄存器子功能（代码 00 0A）来重置。

正常响应包含一个两字节状态字和一个两字节事件计数。如果远程设备仍在处理先前发出的程序命令（存在忙碌条件），则状态字将为全 1（FF FF hex）。否则，状态字将为全零。

请求格式：

字段	长度	范围
功能码	1 字节	0x0B

响应格式：

字段	长度	范围
功能码	1 字节	0x0B
状态	2 字节	0x0000 至 0xFFFF
事件计数	2 字节	0x0000 至 0xFFFF

错误格式：

字段	长度	范围
错误码	1 字节	0x8B
异常码	1 字节	01 或 04

示例：

请求		响应	
字段名	(Hex)	字段名	(Hex)
功能码	0B	功能码	0B
		状态 Hi	FF
		状态 Lo	FF
		事件计数 Hi	01
		事件计数 Lo	08

在此示例中，状态字为 FF FF hex，表示程序功能仍在远程设备中执行。事件计数显示设备已计数 264 (01 08 hex) 个事件。

图 19：获取通信事件计数器状态图

6.10 12 (0x0C) 获取通信事件日志（Get Comm Event Log，仅串行线路）

此功能码用于从远程设备获取状态字、事件计数、消息计数和事件字节字段。

状态字和事件计数与获取通信事件计数器功能（11，0B hex）返回的相同。消息计数器包含远程设备自上次重启、清除计数器操作或上电以来处理的消息数量。此计数与诊断功能（代码

08）子功能返回总线消息计数（代码 11，0B hex）返回的相同。

事件字节字段包含 0-64 个字节，每个字节对应远程设备一次 MODBUS 发送或接收操作的状态。远程设备按时间顺序将事件写入字段。字节 0 是最近的事件。每个新字节会将最旧的字节从字段中清除。

请求格式：

字段	长度	范围
功能码	1 字节	0x0C

响应格式：

字段	长度	范围
功能码	1 字节	0x0C
字节计数	1 字节	N*
状态	2 字节	0x0000 至 0xFFFF
事件计数	2 字节	0x0000 至 0xFFFF
消息计数	2 字节	0x0000 至 0xFFFF
事件	(N-6) x 1 字节	

* N = 事件数量 + 3 x 2 字节（状态、事件计数、消息计数的长度）

错误格式：

字段	长度	范围
错误码	1 字节	0x8C
异常码	1 字节	01 或 04

示例：

请求		响应	
字段名	(Hex)	字段名	(Hex)
功能码	0C	功能码	0C
		字节计数	08
		状态 Hi	00
		状态 Lo	00
		事件计数 Hi	01
		事件计数 Lo	08
		消息计数 Hi	01
		消息计数 Lo	21
		事件 0	20
		事件 1	00

在此示例中，状态字为 00 00 hex，表示远程设备未处理程序功能。事件计数显示远程设备已计数 264 (01 08 hex) 个事件。消息计数显示已处理 289 (01 21 hex) 条消息。

最近的通信事件显示在事件 0 字节中。其内容（20 hex）表示远程设备最近进入了仅监听模式。前一个事件显示在事件 1 字节中。其内容（00 hex）表示远程设备收到了通信重启。

事件字节包含的内容：

获取通信事件日志功能返回的事件字节可以是四种类型之一。类型由每个字节中的位 7（高位）定义，也可以由位 6 进一步定义。

- 远程设备 MODBUS 接收事件：位 7 设为逻辑 1。其他位在相应条件为 TRUE 时设为逻辑 1。位定义：0=未用，1=通信错误，4=字符溢出，5=当前处于仅监听模式，6=收到广播。
- 远程设备 MODBUS 发送事件：位 7 设为逻辑 0，位 6 设为 1。位定义：0=发送读异常(异常码1-3)，1=发送服务器中止异常(异常码4)，2=发送服务器忙异常(异常码5-6)，3=发送服务器程序NAK异常(异常码7)，4=写超时错误发生，5=当前处于仅监听模式。
- 远程设备进入仅监听模式：内容为 04 hex。

- 远程设备启动通信重启：内容为零。

图 20：获取通信事件日志状态图

6.11 15 (0x0F) 写多个线圈（Write Multiple Coils）

此功能码用于强制远程设备中的一系列线圈为 ON 或 OFF。请求 PDU 指定要强制的线圈引用。线圈寻址从零开始。因此编号为 1 的线圈寻址为 0。

请求的 ON/OFF 状态由请求数据字段的内容指定。字段中位位置的逻辑 '1' 请求相应输出为 ON，逻辑 '0' 请求其为 OFF。正常响应返回功能码、起始地址和强制的线圈数量。

请求 PDU：

字段	长度	范围
功能码	1 字节	0x0F
起始地址	2 字节	0x0000 至 0xFFFF
输出数量	2 字节	0x0001 至 0x07B0
字节计数	1 字节	N*
输出值	N* x 1 字节	

* N = 输出数量 / 8，如果余数不为 0 则 N = N+1

响应 PDU：

字段	长度	范围
功能码	1 字节	0x0F
起始地址	2 字节	0x0000 至 0xFFFF
输出数量	2 字节	0x0001 至 0x07B0

错误格式：

字段	长度	范围
错误码	1 字节	0x8F
异常码	1 字节	01 或 02 或 03 或 04

示例：从线圈 20 开始写入 10 个线圈

请求数据内容为两个字节：CD 01 hex（1100 1101 0000 0001 二进制）。

第一个传输的字节（CD hex）寻址输出 27-20，最低有效位寻址该组中最低的输出（20）。下一个传输的字节（01 hex）寻址输出 29-28，最低有效位寻址该组中最低的输出（28）。最后一个数据字节中未使用的位应零填充。

请求		响应	
字段名	(Hex)	字段名	(Hex)
功能码	0F	功能码	0F
起始地址 Hi	00	起始地址 Hi	00
起始地址 Lo	13	起始地址 Lo	13
输出数量 Hi	00	输出数量 Hi	00
输出数量 Lo	0A	输出数量 Lo	0A
字节计数	02		
输出值 Hi	CD		
输出值 Lo	01		

图 21：写多个输出状态图

6.12 16 (0x10) 写多个寄存器（Write Multiple Registers）

此功能码用于在远程设备中写入连续寄存器块（1 到 123 个寄存器）。请求的写入值在请求数据字段中指定。数据以每个寄存器两个字节的方式打包。正常响应返回功能码、起始地址和写入的寄存器数量。

请求格式：

字段	长度	范围
功能码	1 字节	0x10
起始地址	2 字节	0x0000 至 0xFFFF
寄存器数量	2 字节	0x0001 至 0x007B
字节计数	1 字节	2 x N*
寄存器值	N* x 2 字节	

* N = 寄存器数量

响应格式：

字段	长度	范围
功能码	1 字节	0x10
起始地址	2 字节	0x0000 至 0xFFFF
寄存器数量	2 字节	1 至 123 (0x7B)

错误格式：

字段	长度	范围
错误码	1 字节	0x90
异常码	1 字节	01 或 02 或 03 或 04

示例：从地址 2 开始写入两个寄存器，值分别为 00 0A 和 01 02 hex

请求		响应	
字段名	(Hex)	字段名	(Hex)
功能码	10	功能码	10
起始地址 Hi	00	起始地址 Hi	00
起始地址 Lo	01	起始地址 Lo	01
寄存器数量 Hi	00	寄存器数量 Hi	00
寄存器数量 Lo	02	寄存器数量 Lo	02
字节计数	04		
寄存器值 Hi	00		
寄存器值 Lo	0A		
寄存器值 Hi	01		
寄存器值 Lo	02		

图 22：写多个寄存器状态图

6.13 17 (0x11) 报告服务器 ID（Report Server ID，仅串行线路）

此功能码用于读取远程设备的类型描述、当前状态和其他特定信息。正常响应的格式如以下示例所示。数据内容特定于每种设备类型。

请求格式：

字段	长度	范围
功能码	1 字节	0x11

响应格式：

字段	长度	范围
功能码	1 字节	0x11
字节计数	1 字节	
服务器 ID	设备特定	
运行指示灯状态	1 字节	0x00 = OFF, 0xFF = ON
附加数据		

错误格式：

字段	长度	范围
错误码	1 字节	0x91
异常码	1 字节	01 或 04

示例：

请求		响应	
字段名	(Hex)	字段名	(Hex)
功能码	11	功能码	11
		字节计数	设备特定
		服务器 ID	设备特定
		运行指示灯状态	0x00 或 0xFF
		附加数据	设备特定

图 23：报告服务器 ID 状态图

6.14 20 (0x14) 读文件记录（Read File Record）

此功能码用于执行文件记录读取。所有请求数据长度以字节数提供，所有记录长度以寄存器数提供。文件是记录的组织。每个文件包含 10000 条记录，寻址范围 0000 至 9999 十进制或 0x0000 至 0x270F。例如，记录 12 寻址为 12。

该功能可以读取多个引用组。组可以是分离的（非连续），但每组内的引用必须是连续的。每个组在单独的'子请求'字段中定义，包含 7 个字节：

- 引用类型：1 字节（必须指定为 6）
- 文件号：2 字节
- 文件内起始记录号：2 字节
- 要读取的记录长度：2 字节

要读取的寄存器数量加上预期响应中的所有其他字段，不得超过 MODBUS PDU 的允许长度：253 字节。

请求格式：

字段	长度	范围
功能码	1 字节	0x14
字节计数	1 字节	0x07 至 0xF5 字节
子请求 x, 引用类型	1 字节	06
子请求 x, 文件号	2 字节	0x0001 至 0xFFFF
子请求 x, 记录号	2 字节	0x0000 至 0x270F
子请求 x, 记录长度	2 字节	N
子请求 x+1, ...		

响应格式：

字段	长度	范围
功能码	1 字节	0x14
响应数据长度	1 字节	0x07 至 0xF5
子请求 x, 文件响应长度	1 字节	0x07 至 0xF5
子请求 x, 引用类型	1 字节	6
子请求 x, 记录数据	N x 2 字节	
子请求 x+1, ...		

错误格式：

字段	长度	范围
错误码	1 字节	0x94
异常码	1 字节	01 或 02 或 03 或 04 或 08

虽然文件号允许在 1 到 0xFFFF 范围内，但应注意如果文件号大于 10 (0x0A)，可能与遗留设备的互操作性会受到影响。

示例：从远程设备读取两组引用：组 1 包含文件 4 中从寄存器 1（地址 0001）开始的两个寄存器；组 2 包含文件 3 中从寄存器 9（地址 0009）开始的两个寄存器。

请求		响应	
字段名	(Hex)	字段名	(Hex)
功能码	14	功能码	14
字节计数	0E	响应数据长度	0C
子请求 1，引用类型	06	子请求 1，文件响应长度	05
子请求 1，文件号 Hi	00	子请求 1，引用类型	06
子请求 1，文件号 Lo	04	子请求 1，寄存器数据 Hi	0D
子请求 1，记录号 Hi	00	子请求 1，寄存器数据 Lo	FE
子请求 1，记录号 Lo	01	子请求 1，寄存器数据 Hi	00
子请求 1，记录长度 Hi	00	子请求 1，寄存器数据 Lo	20
子请求 1，记录长度 Lo	02	子请求 2，文件响应长度	05
子请求 2，引用类型	06	子请求 2，引用类型	06
子请求 2，文件号 Hi	00	子请求 2，寄存器数据 Hi	33
子请求 2，文件号 Lo	03	子请求 2，寄存器数据 Lo	CD
子请求 2，记录号 Hi	00	子请求 2，寄存器数据 Hi	00
子请求 2，记录号 Lo	09	子请求 2，寄存器数据 Lo	40
子请求 2，记录长度 Hi	00		
子请求 2，记录长度 Lo	02		

图 24：读文件记录状态图

6.15 21 (0x15) 写文件记录（Write File Record）

此功能码用于执行文件记录写入。所有请求数据长度以字节数提供，所有记录长度以 16 位字数提供。文件是记录的组织。每个文件包含 10000 条记录，寻址范围 0000 至 9999 十进制或 0x0000 至 0x270F。

该功能可以写入多个引用组。组可以是分离的（非连续），但每组内的引用必须是连续的。每个组在单独的'子请求'字段中定义，包含 7 个字节加数据：

- 引用类型：1 字节（必须指定为 6）
- 文件号：2 字节
- 文件内起始记录号：2 字节
- 要写入的记录长度：2 字节
- 要写入的数据：每个寄存器 2 字节

要写入的寄存器数量加上请求中的所有其他字段，不得超过 MODBUS PDU 的允许长度：253 字节。正常响应是请求的回显。

请求格式：

字段	长度	范围
功能码	1 字节	0x15
请求数据长度	1 字节	0x09 至 0xFB
子请求 x, 引用类型	1 字节	06
子请求 x, 文件号	2 字节	0x0001 至 0xFFFF
子请求 x, 记录号	2 字节	0x0000 至 0x270F
子请求 x, 记录长度	2 字节	N
子请求 x, 记录数据	N x 2 字节	

响应格式：

字段	长度	范围
功能码	1 字节	0x15
响应数据长度	1 字节	0x09 至 0xFB
子请求 x, 引用类型	1 字节	06
子请求 x, 文件号	2 字节	0x0001 至 0xFFFF
子请求 x, 记录号	2 字节	0x0000 至 0x270F
子请求 x, 记录长度	2 字节	N
子请求 x, 记录数据	N x 2 字节	

错误格式：

字段	长度	范围
错误码	1 字节	0x95
异常码	1 字节	01 或 02 或 03 或 04 或 08

示例：写入一组引用到远程设备——该组包含文件 4 中从寄存器 7（地址 0007）开始的三个寄存器。

请求		响应	
字段名	(Hex)	字段名	(Hex)
功能码	15	功能码	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, 记录长度 Hi	00	子请求 1, 记录长度 Hi	00

请求		响应	
子请求 1，记录长度 Lo	03	子请求 1，记录长度 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

图 25：写文件记录状态图

6.16 22 (0x16) 掩码写寄存器（Mask Write Register）

此功能码用于通过 AND 掩码、OR 掩码和寄存器当前内容的组合来修改指定保持寄存器的内容。该功能可用于设置或清除寄存器中的单个位。

请求指定要写入的保持寄存器、用作 AND 掩码的数据和用作 OR 掩码的数据。寄存器寻址从零开始。因此寄存器 1-16 寻址为 0-15。

该功能的算法为：

```
Result = (Current Contents AND And_Mask) OR (Or_Mask AND (NOT And_Mask))
```

例如：

	Hex	Binary
Current Contents =	12	0001 0010
And_Mask =	F2	1111 0010
Or_Mask =	25	0010 0101
(NOT And_Mask) =	0D	0000 1101
Result =	17	0001 0111

注意：如果 Or_Mask 值为零，结果就是当前内容与 And_Mask 的逻辑 AND。如果 And_Mask 值为零，结果等于 Or_Mask 值。寄存器内容可以用读保持寄存器功能（功能码 03）读取，但可能在控制器扫描其用户逻辑程序时随后被更改。

请求格式：

字段	长度	范围
功能码	1 字节	0x16
引用地址	2 字节	0x0000 至 0xFFFF
And_Mask	2 字节	0x0000 至 0xFFFF
Or_Mask	2 字节	0x0000 至 0xFFFF

响应格式：

字段	长度	范围
功能码	1 字节	0x16
引用地址	2 字节	0x0000 至 0xFFFF
And_Mask	2 字节	0x0000 至 0xFFFF
Or_Mask	2 字节	0x0000 至 0xFFFF

错误格式：

字段	长度	范围
错误码	1 字节	0x96
异常码	1 字节	01 或 02 或 03 或 04

示例：对远程设备中的寄存器 5 进行掩码写入，使用上述掩码值

请求		响应	
字段名	(Hex)	字段名	(Hex)
功能码	16	功能码	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

图 26：掩码写保持寄存器状态图

6.17 23 (0x17) 读/写多个寄存器（Read/Write Multiple Registers）

此功能码在单个 MODBUS 事务中执行一次读操作和一次写操作的组合。写操作在读操作之前执行。保持寄存器寻址从零开始。因此保持寄存器 1-16 在 PDU 中寻址为 0-15。

请求指定要读取的保持寄存器的起始地址和数量，以及要写入的起始地址、保持寄存器数量和数据。字节计数字段指定写入数据字段中要跟随的字节数。正常响应包含从读取的寄存器组中获取的数据。字节计数字段指定读取数据字段中要跟随的字节数。

请求格式：

字段	长度	范围
功能码	1 字节	0x17
读起始地址	2 字节	0x0000 至 0xFFFF
读数量	2 字节	0x0001 至 0x007D
写起始地址	2 字节	0x0000 至 0xFFFF
写数量	2 字节	0x0001 至 0x0079
写字节计数	1 字节	2 x N*
写寄存器值	N* x 2 字节	

* N = 写数量

响应格式：

字段	长度	范围
功能码	1 字节	0x17
字节计数	1 字节	2 x N'*
读寄存器值	N'* x 2 字节	

* N' = 读数量

错误格式：

字段	长度	范围
错误码	1 字节	0x97
异常码	1 字节	01 或 02 或 03 或 04

示例：读取从寄存器 4 开始的六个寄存器，并写入从寄存器 15 开始的三个寄存器

请求		响应	
字段名	(Hex)	字段名	(Hex)
功能码	17	功能码	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
读数量 Hi	00	读寄存器值 Lo	FE
读数量 Lo	06	读寄存器值 Hi	0A
写起始地址 Hi	00	读寄存器值 Lo	CD
写起始地址 Lo	0E	读寄存器值 Hi	00
写数量 Hi	00	读寄存器值 Lo	01
写数量 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		

图 27：读/写多个寄存器状态图

6.18 24 (0x18) 读 FIFO 队列（Read FIFO Queue）

此功能码允许读取远程设备中先进先出（FIFO）寄存器队列的内容。该功能返回队列中寄存器的计数，后跟队列数据。最多可读取 32 个寄存器：计数，加上最多 31 个队列数据寄存器。队列计数寄存器首先返回，后跟队列数据寄存器。

该功能读取队列内容，但不清除它们。在正常响应中，字节计数显示要跟随的字节数量，包括队列计数字节和值寄存器字节（但不包括错误检查字段）。队列计数是队列中数据寄存器的数量（不包括计数寄存器）。

如果队列计数超过 31，则返回异常响应，错误码为 03（非法数据值）。

请求格式：

字段	长度	范围
功能码	1 字节	0x18
FIFO 指针地址	2 字节	0x0000 至 0xFFFF

响应格式：

字段	长度	范围
功能码	1 字节	0x18
字节计数	2 字节	
FIFO 计数	2 字节	⩽ 31
FIFO 值寄存器	N* x 2 字节	

* N = FIFO 计数

错误格式：

字段	长度	范围
错误码	1 字节	0x98
异常码	1 字节	01 或 02 或 03 或 04

示例：向远程设备请求读 FIFO 队列。请求读取从指针寄存器 1246 (0x04DE) 开始的队列

请求		响应	
字段名	(Hex)	字段名	(Hex)
功能码	18	功能码	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 值寄存器 Hi	01
		FIFO 值寄存器 Lo	B8
		FIFO 值寄存器 Hi	12
		FIFO 值寄存器 Lo	84

在此示例中，FIFO 指针寄存器（请求中的 1246）返回队列计数 2。两个数据寄存器跟在队列计数之后。它们是：1247（内容 440 十进制 -- 0x01B8）和 1248（内容 4740 -- 0x1284）。

图 28：读 FIFO 队列状态图

6.19 43 (0x2B) 封装接口传输（Encapsulated Interface Transport）

> 说明性注释：请参阅附录 A（资料性）MODBUS 保留功能码、子码和 MEI 类型。

功能码 43 及其用于设备标识的 MEI 类型 14 是本规范中当前可用的两种封装接口传输之一。以下功能码和 MEI 类型不应成为本已发布规范的一部分，这些功能码和 MEI 类型被特别保留：43/0-12 和 43/15-255。

MODBUS 封装接口（MEI）传输是一种在 MODBUS PDU 内部传递服务请求和方法调用及其返回的隧道机制。MEI 传输的主要特性是封装属于已定义接口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 的任何通信协议栈，如 TCP/IP 或串行线路。MEI 类型是MODBUS 分配号，因此是唯一的。除 MEI 类型 13 和 MEI 类型 14 外，0 到 255 之间的值根据附录 A（资料性）保留。MEI 类型由 MEI 传输实现用于将方法调用分发到指定的接口。

请求格式：

字段	长度	范围
功能码	1 字节	0x2B
MEI 类型*	1 字节	0x0D 或 0x0E
MEI 类型特定数据	n 字节	

* MEI = MODBUS 封装接口

响应格式：

字段	长度	范围
功能码	1 字节	0x2B
MEI 类型	1 字节	回显请求中的 MEI 类型
MEI 类型特定数据	n 字节	

错误格式：

字段	长度	范围
错误码	1 字节	0xAB (Fc 0x2B + 0x80)
异常码	1 字节	01 或 02 或 03 或 04

6.20 43/13 (0x2B/0x0D) CANopen 通用参考请求和响应 PDU

CANopen 通用参考命令是对服务的封装，这些服务将用于访问（读取或写入）CANopen 设备对象字典的条目，以及控制和监控 CANopen 系统和设备。MEI 类型 13 (0x0D) 是授权给 CiA 用于 CANopen 通用参考的 MODBUS 分配号。

该系统旨在在现有 MODBUS 网络的限制内工作。因此，查询或修改系统中对象字典所需的信息被映射为 MODBUS 消息的格式。PDU 在请求和响应消息中都有 253 字节的限制。

资料性：有关 MEI 类型 13 的信息，请参阅附录 B 中的规范参考。

6.21 43/14 (0x2B/0x0E) 读设备标识（Read Device Identification）

此功能码仅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。

读设备标识接口被建模为由一组可寻址数据元素组成的地址空间。数据元素称为对象，由对象 ID 标识。该接口包含 3 类对象：

- 基本设备标识（Basic Device Identification）。此类别中的所有对象都是必需的：VendorName、ProductCode 和 revision number。
- 常规设备标识（Regular Device Identification）。除基本数据对象外，设备还提供额外和可选的标识和描述数据对象。此类别中的所有对象都在标准中定义，但其实现是可选的。
- 扩展设备标识（Extended Device Identification）。除常规数据对象外，设备还提供关于物理设备本身的额外和可选标识及描述私有数据。所有这些数据都是设备相关的。

对象 ID 和名称表：

对象 ID	对象名称/描述	类型	M/O	类别
0x00	VendorName（供应商名称）	ASCII 字符串	必备	基本
0x01	ProductCode（产品代码）	ASCII 字符串	必备	基本
0x02	MajorMinorRevision（主次版本号）	ASCII 字符串	必备	基本
0x03	VendorUrl（供应商 URL）	ASCII 字符串	可选	常规
0x04	ProductName（产品名称）	ASCII 字符串	可选	常规
0x05	ModelName（型号名称）	ASCII 字符串	可选	常规
0x06	UserApplicationName（用户应用名称）	ASCII 字符串	可选	常规
0x07..0x7F	保留		可选	
0x80..0xFF	私有对象（可选择性定义）	设备相关	可选	扩展

请求格式：

字段	长度	范围
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 至 0xFF

* MEI = MODBUS 封装接口

响应格式：

字段	长度	范围
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
一致性级别	1 字节	0x01/02/03/81/82/83
更多后续	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	
对象 ID 列表	1 字节	
对象长度	1 字节	
对象值	取决于对象 ID	

错误格式：

字段	长度	范围
错误码	1 字节	0xAB (Fc 0x2B + 0x80)
异常码	1 字节	01 或 02 或 03 或 04

请求参数描述：

MODBUS 封装接口分配号 14 标识读设备标识请求。参数'读设备 ID 码'允许定义四种访问类型：

- 01：请求获取基本设备标识（流访问）
- 02：请求获取常规设备标识（流访问）
- 03：请求获取扩展设备标识（流访问）
- 04：请求获取一个特定标识对象（个别访问）

如果读设备 ID 码非法，响应中将发送异常码 03。在响应无法放入单个响应的情况下，必须进行多次事务（请求/响应）。对象 ID 字节给出要获取的第一个对象的标识。对于第一次事务，客户端必须将对象 ID 设置为 0 以获取设备标识数据的开头。对于后续事务，客户端必须将对象 ID 设置为服务器端在其先前响应中返回的值。

备注：对象不可分割，因此任何对象的大小必须与事务响应的大小一致。如果对象 ID 不匹配任何已知对象，服务器端响应如同指向对象 0（从头开始）。在个别访问情况下（ReadDevId code 04），请求中的对象 ID 给出要获取的对象的标识，如果对象 ID 不匹配任何已知对象，服务器端返回异常响应，异常码 = 02（非法数据地址）。

如果要求服务器端设备提供高于其一致性级别的描述级别（readDevice Code），它必须根据其实际一致性级别进行响应。

响应参数描述：

一致性级别（Conformity Level）——设备标识一致性级别和支持的访问类型：

一致性级别	含义
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（流访问和个别访问）
0x82	常规标识（流访问和个别访问）
0x83	扩展标识（流访问和个别访问）

更多后续（More Follows）——在 ReadDevId 码 01、02 或 03（流访问）情况下：如果标识数据无法放入单个响应，可能需要多次请求/响应事务。0x00：没有更多对象可用；0xFF：有其他标识对象可用，需要进一步的 MODBUS 事务。在 ReadDevId 码 04（个别访问）情况下：此字段必须设置为 00。

示例 1：基本设备标识——所有信息在一个响应 PDU 中发送

请求		响应	
字段名	值	字段名	值
功能码	2B	功能码	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	一致性级别	01
		更多后续	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"

示例 2：需要多次事务的设备——第一次事务

请求		响应	
字段名	值	字段名	值
功能码	2B	功能码	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	一致性级别	01
		更多后续	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 XXXXXX XXXXXXXXXX"

第二次事务：

请求		响应	
字段名	值	字段名	值
功能码	2B	功能码	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	一致性级别	01
		更多后续	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"

图 30：读设备标识状态图

7 MODBUS 异常响应

当客户端设备向服务器端设备发送请求时，它期望正常响应。客户端的查询可能发生以下四种事件之一：

- 如果服务器端设备收到请求而没有通信错误，并且可以正常处理查询，则返回正常响应。
- 如果服务器端由于通信错误未收到请求，则不返回响应。客户端程序最终将处理请求的超时条件。
- 如果服务器端收到请求，但检测到通信错误（奇偶校验、LRC、CRC...），则不返回响应。客户端程序最终将处理请求的超时条件。
- 如果服务器端收到请求而没有通信错误，但无法处理（例如，如果请求是读取不存在的输出或寄存器），服务器端将返回异常响应，告知客户端错误的性质。

异常响应消息有两个字段将其与正常响应区分开来：

功能码字段：在正常响应中，服务器端在响应的功能码字段中回显原始请求的功能码。所有功能码的最高有效位（MSB）为 0（它们的值都低于 80 十六进制）。在异常响应中，服务器端将功能码的 MSB 设置为 1。这使得异常响应中的功能码值比正常响应的值高 80 十六进制。通过设置功能码的 MSB，客户端的应用程序可以识别异常响应并检查数据字段中的异常码。

数据字段：在正常响应中，服务器端可以在数据字段中返回数据或统计信息（请求中要求的任何信息）。在异常响应中，服务器端在数据字段中返回异常码。这定义了导致异常的服务器端条件。

客户端请求与服务器端异常响应示例：

请求		响应	
字段名	(Hex)	字段名	(Hex)
功能码	01	功能码	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		

在此示例中，客户端向服务器端设备寻址请求。功能码（01）用于读线圈操作。它请求地址 1185 (04A1 hex) 处的输出状态。请注意，仅读取一个输出，如输出数量字段 (0001) 所指定。如果输出地址在服务器端设备中不存在，服务器端将返回带有所示异常码 (02) 的异常响应。这指定了对服务器端而言非法的数据地址。

MODBUS 异常码表：

码	名称	中文	含义
01	ILLEGAL FUNCTION	非法功能	查询中接收到的功能码不是服务器端允许的操作。这可能是因为该功能码仅适用于较新的设备，而在所选单元中未实现。也可能表示服务器端处于错误状态，无法处理此类请求，例如因为其未配置而被要求返回寄存器值。
02	ILLEGAL DATA ADDRESS	非法数据地址	查询中接收到的数据地址不是服务器端允许的地址。更具体地说，引用号和传输长度的组合无效。对于具有 100 个寄存器的控制器，PDU 将第一个寄存器寻址为 0，最后一个寻址为 99。如果提交起始寄存器地址为 96、寄存器数量为 4 的请求，则此请求将成功操作（至少在地址方面）寄存器 96、97、98、99。如果提交起始寄存器地址为 96、寄存器数量为 5 的请求，则此请求将失败，异常码为 0x02' 非法数据地址'，因为它试图操作寄存器 96、97、98、99 和 100，而地址 100 处没有寄存器。
03	ILLEGAL DATA VALUE	非法数据值	查询数据字段中包含的值不是服务器端允许的值。这表明复杂请求剩余部分的结构存在错误，例如隐含长度不正确。它明确不表示提交存储在寄存器中的数据项值超出应用程序预期的范围，因为 MODBUS 协议不知道任何特定寄存器值的含义。
04	SERVER DEVICE FAILURE	从站设备故障	当服务器端尝试执行请求的操作时发生不可恢复的错误。
05	ACKNOWLEDGE	确认	与编程命令结合使用的专门用途。服务器端已接受请求并正在处理，但需要较长时间来完成。返回此响应以防止客户端发生超时错误。客户端接下来可以发出轮询程序完成消息以确定处理是否完成。
06	SERVER DEVICE BUSY	从站设备忙	与编程命令结合使用的专门用途。服务器端正在处理一个长时间的程序命令。客户端应稍后在服务器端空闲时重新发送消息。

码	名称	中文	含义
08	MEMORY PARITY ERROR	存储器奇偶校验错误	与功能码 20 和 21 以及引用类型 6 结合使用的专门用途，指示扩展文件区域未能通过一致性检查。服务器端尝试读取记录文件，但在内存中检测到奇偶校验错误。客户端可以重试请求，但服务器端设备可能需要服务。
0A	GATEWAY PATH UNAVAILABLE	网关路径不可用	与网关结合使用的专门用途，指示网关无法分配从输入端口到输出端口的内部通信路径以处理请求。通常表示网关配置错误或过载。
0B	GATEWAY TARGET DEVICE FAILED TO RESPOND	网关目标设备响应失败	与网关结合使用的专门用途，指示未从目标设备获得响应。通常表示设备不在网络上。

附录 A（资料性）：MODBUS 保留功能码、子码和 MEI 类型

以下功能码和子码不得成为本已发布规范的一部分，这些功能码和子码被特别保留。格式为功能码/子码，或者仅为功能码（当所有子码 0-255 都被保留时）：8/19; 8/21-65535, 9, 10, 13, 14, 41, 42, 90, 91, 125, 126 和 127。

功能码 43 及其用于设备标识的 MEI 类型 14 和用于 CANopen 通用参考请求和响应 PDU 的 MEI 类型 13 是本规范中当前可用的封装接口传输。以下功能码和 MEI 类型不得成为本已发布规范的一部分，这些功能码和 MEI 类型被特别保留：43/0-12 和 43/15-255。

在本规范中，不支持具有与封装接口传输相同或相似结果的用户定义功能码。

MODBUS 是 Schneider Automation Inc. 的注册商标。

附录 B（资料性）：CANopen 通用参考命令

请参阅 MODBUS 网站或 CiA（CAN in Automation）网站，获取涵盖功能码 43 MEI 类型 13 的副本和使用条款。

— 文档结束 —