

Modbus 协议在串行链路上的 的实现指南

1 引言

1.1 范围

Modbus 标准定义了 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。它还将串行链路上的协议标准化，以便在一个主站和一个或多个从站之间交换 Modbus 请求。

本文件的目的是表述串行链路上的 Modbus 协议。使用对象为在他们的产品实现串行链路 Modbus 协议的系统设计者。

本文件将增进使用 Modbus 协议的设备之间的互通性。

本文件还是对“Modbus 协议规范”的补充。

第 5 章定义了“Modbus 串行链路”的实现等级。级别的规范是对一个设备能够属于某个级别而必须遵守的要求的总和。

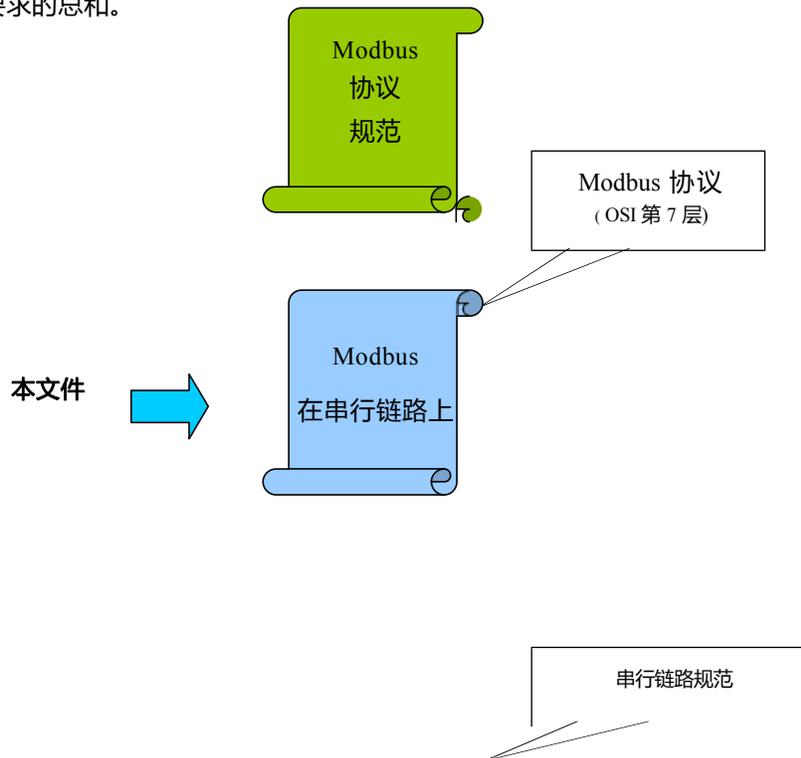


图 1：Modbus 文件概要

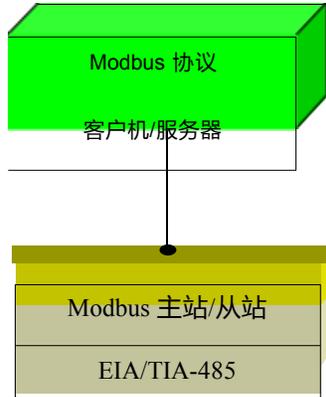
1.2 协议概述

本文件描述 Modbus 串行链路协议。**Modbus 串行链路协议是一个主/从协议。**该协议位于 OSI 模型的第二层。

一个主从类型的系统有一个向某个“子”节点发出显式命令并处理响应的节点(主节点)。典型的子节点在没有收到主节点的请求时并不主动发送数据，也不与其它子节点通信。

在物理层，Modbus 串行链路系统可以使用不同的物理接口(RS485、RS232)。最常用的是 TIA/EIA-485 (RS485) 两线制接口。作为附加的选项，也可以实现 RS485 四线制接口。当只需要短距离的点到点通信时，TIA/EIA-232-E (RS232) 串行接口也可以使用。(参见有关“物理层”的章节)

下图给出了 Modbus 串行通信栈对应于 7 层 OSI 模型的一般关系。



层	ISO/OSI 模型	
7	应用层	Modbus 协议
6	表示层	空
5	会话层	空
4	传输层	空
3	网络层	空
2	数据链路层	Modbus 串行链路协议
1	物理层	EIA/TIA-485 (或 EIA/TIA-232)

图 1 Modbus 协议和 ISO/OSI 模型

位于 OSI 模型第 7 层的 Modbus 应用层报文传输协议，提供了连接于总线或网络的设备之间的客户机/服务器通信。在 Modbus 串行链路上客户机的功能由主节点提供而服务器功能由子节点实现。

1.3 约定

在本文件中，下列词汇用于定义每一种要求的重要程度。

👉 "必须" / "要求的"

含有词语 "必须" 的要求是强制的。词语**必须**，或形容词**要求的**，表示该项为实现的绝对要

求。这些词语带有下划线。

"应该" / "建议的"

所有包含 "应该", 或形容词 "建议的" 建议, 为期望的功能。这些建议应该作为选择不同的实现选项时的指南。在有合理的理由的特定条件下, 可以忽略这些项。但是, 对其全部含义应该理解并且基于情况做出选择时应仔细斟酌。 这些词语带有下划线。

"可以" / "可选的"

词语 "可以", 或形容词 "可选的", 表示该项为真正意义的可选的。某个设计者可以选择包含该项(基于特定的市场需求或产品功能增强); 而另一个可以选择忽略该项。

1.4 一致性

如果某个实现不满足实现级别中一个或多个**必须**的要求，则是**不符合的**。

如果某个实现满足实现级别中所有的**必须**要求和所有的**应该**的建议，则称为**"无条件符合的"**。

如果某个实现满足实现级别中所有的**必须**要求和不是所有的**应该**的建议，则称为**"有条件符合的"**。

1.5 缩略语

定义本文件中用到的特定词汇、符号和缩略语。

2W	在“电气接口”一章中定义的两线制配置，或其中的一个接口。
4W	在“电气接口”一章中定义的四线制配置，或其中的一个接口。
2W+2W	在两线制系统中使用四线制接口的特殊配置。 (需要时可参见 Schneider Electric Momentum 文件 870 USE 101 10)。
AUI	连接单元接口 (Attachment Unit Interface)
公共端	EIA/TIA 标准中的信号公共端 (Common)。在两线制或四线制 RS485 Modbus 网络中，信号和可选的电源的公共端。 Power Supply 公共端
DCE	一个 Modbus 设备。例如，实现了 RS232 数据电路设备 (Data-Circuit Equipment) 的可编程序控制器适配器。也称作数据通信设备(Data Communication Equipment)。
设备	或“Modbus 设备”：参见其定义。
驱动器	发生器，或发送器
DTE	一个 Modbus 设备。例如，实现了 RS232 数据终端设备 (Data Terminal Equipment) 编程终端或 PC
ITr	干缆侧的物理总线接口 (Interface on Trunk Side)。
IDv	设备侧的物理总线接口 (Interface on Derivation Side)。
LT	线路终端(Line Termination)。
Modbus 设备	实现了 Modbus 串行链路并遵循技术规范的设备。
RS232	EIA/ TIA -232 标准。

RS485 EIA/ TIA -485 标准。

RS485 Modbus 与该技术标准一致的两线制或四线制网络

收发器 (Transceiver) 发送器和接收器。(a **Transmitter** and a **Receiver** 或驱动器和接收器)。

2 Modbus 数据链路层

2.1 Modbus 主站/从站协议原理

Modbus 串行链路协议是一个主-从协议。在同一时刻，只有一个主节点连接于总线，一个或多个子节点（最大编号为 247）连接于同一个串行总线。Modbus 通信总是由主节点发起。子节点在没有收到来自主节点的请求时，从不会发送数据。子节点之间从不会互相通信。主节点在同一时刻只会发起一个 Modbus 事务处理。

主节点以两种模式对子节点发出 Modbus 请求：

→ 在**单播模式**，主节点以特定地址访问某个子节点，子节点接到并处理完请求后，子节点向主节点返回一个报文(一个 '应答')。

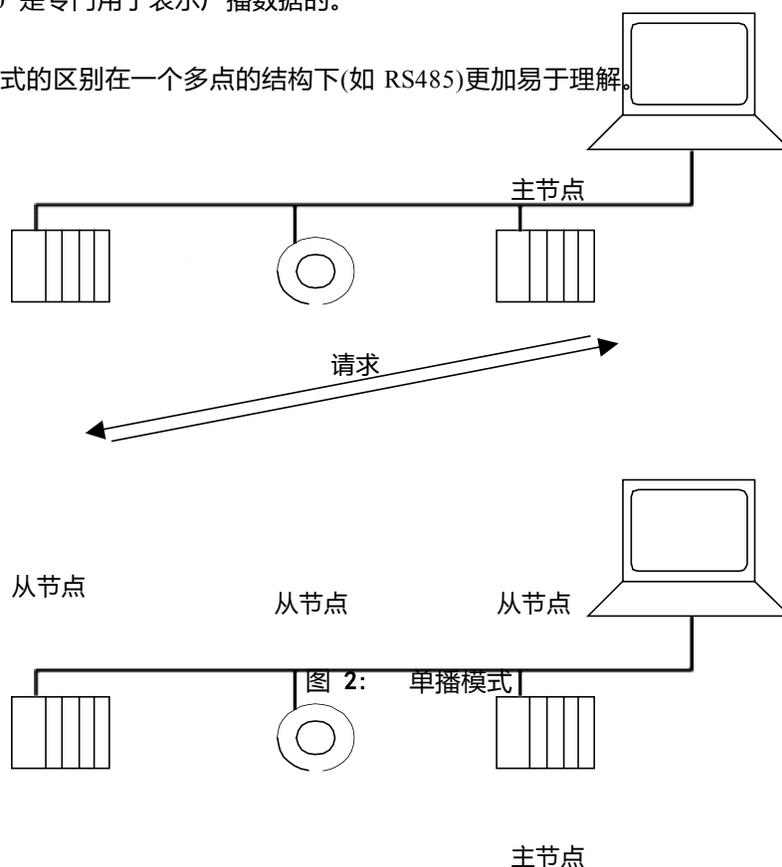
在这种模式，一个 Modbus 事务处理包含 2 个报文：一个来自主节点的请求，一个来自子节点的应答。

每个子节点必须有唯一的地址 (1 到 247)，这样才能区别于其它节点被独立的寻址。

→ 在**广播模式**，主节点向所有的子节点发送请求。

对于主节点广播的请求没有应答返回。广播请求一般用于写命令。**所有设备必须接受广播模式的写功能**。地址 0 是专门用于表示广播数据的。

单播和广播模式的区别在一个多点的结构下(如 RS485)更加易于理解。



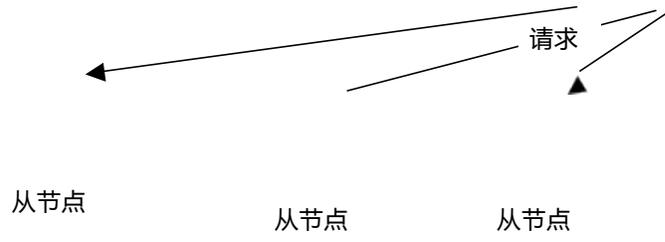


图 3: 广播模式

2.2 Modbus 地址规则

Modbus 寻址空间有 256 个不同地址。

0	1 ~ 47	248 ~ 55
广播地址	子节点单独地址	保留

地址 0 保留为广播地址。 所有的子节点必须识别广播地址。

Modbus 主节点没有地址， 只有子节点必须有一个地址。 该地址必须在 Modbus 串行总线上唯一。

2.3 Modbus 帧描述

Modbus 应用协议 [1] 定义了简单的独立于其下面通信层的**协议数据单元(PDU - Protocol Data Unit)**：

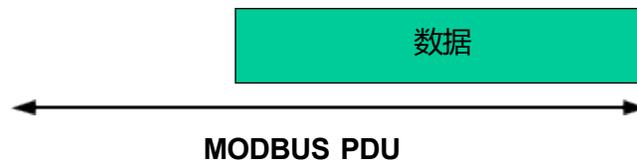
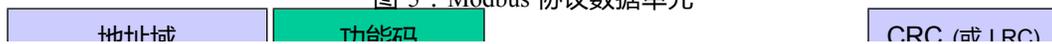


图 5：Modbus 协议数据单元



在不同总线或网络的 Modbus 协议映射在**协议数据单元**之外引入了一些附加的域。发起 Modbus 事务处理的客户端构造 Modbus PDU，然后添加附加的域以构造适当的通信 PDU。

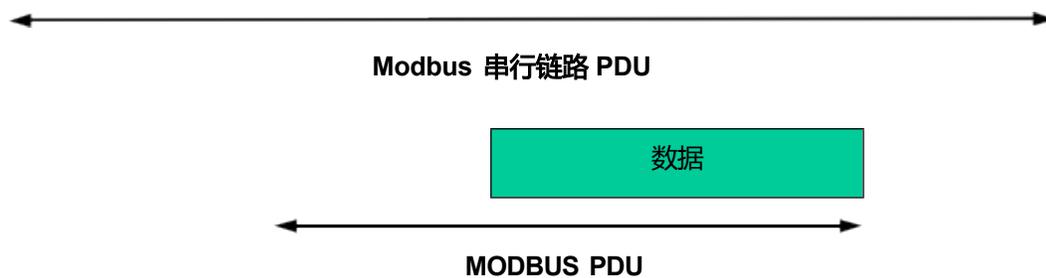


图 6：串行链路上的 Modbus 帧

 在 Modbus 串行链路，地址域只含有子节点地址。

如前文所述，合法的子节点地址为十进制 0 - 247。每个子设备被赋予 1 - 247 范围中的地址。主节点通过将子节点的地址放到报文的地址域对子节点寻址。当子节点返回应答时，它将自己的地址放到应答报文的地址域以让主节点知道哪个子节点在回答。

- 👁️ 功能码指明服务器要执行的动作。功能码后面可跟有表示含有请求和响应参数的数据域。
- 👁️ 错误检验域是对报文内容执行 "冗余校验" 的计算结果。根据不同的传输模式 (RTU or ASCII) 使用两种不同的计算方法。（参见 2.5 节，"*两种串行传输模式*"）

2.4 主站/从站状态图

Modbus 由两个不同的子层组成：

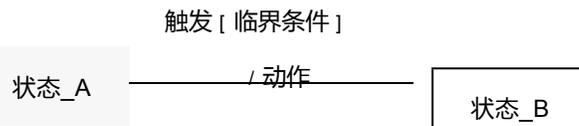
- 主/从协议
- 传输模式 (RTU 和 ASCII 模式)

下面的章节描述了主节点和子节点与传输模式无关的状态图。

RTU 和 ASCII 传输模式在下一章用两个状态图具体说明。描述了一个帧的接收和发送。

状态图词法：

下面的状态图使用与 UML 标准标记法绘制。标记法要点如下：



当一个系统处于"状态_A"时发生"触发"事件,只有当"临界条件"为真时系统会转换到"状态_B",然后,一个"动作"被执行。

2.4.1 主站状态图

下图描述了主节点的状态特征：

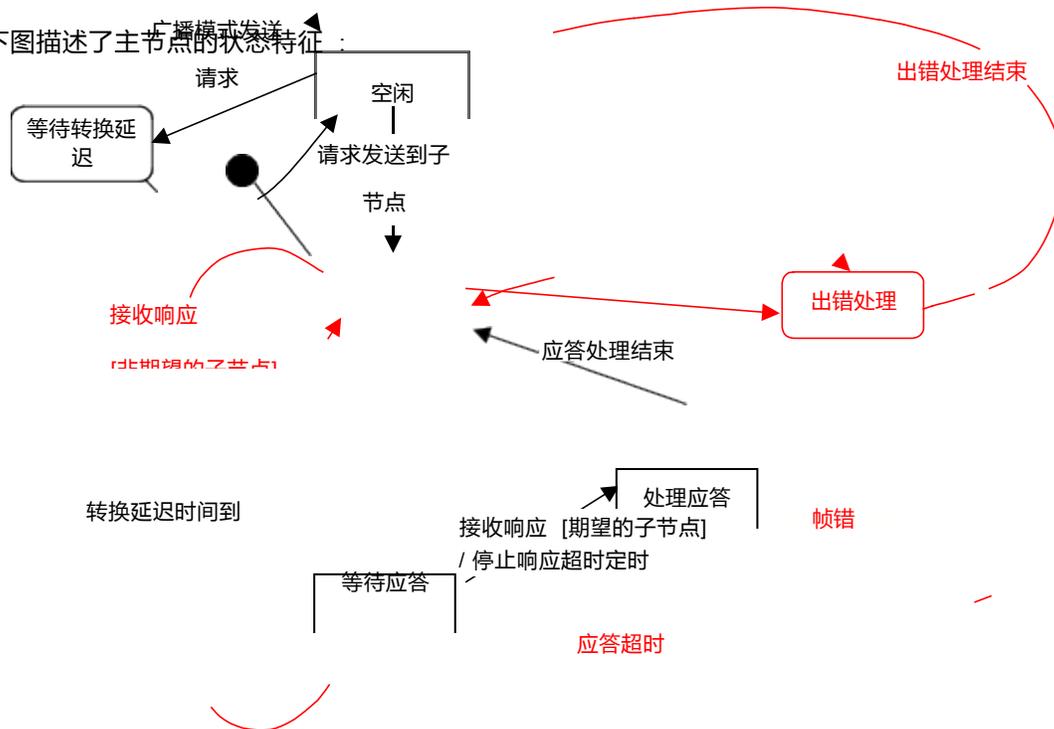


图 4: 主节点状态图

对上面的状态图的一些解释：

- ☞ 状态 "空闲" = 无等待的请求。这是电源上电后的初始状态。只有在"空闲"状态请求才能被发送。发送一个请求后，主节点离开"空闲"状态，而且不能同时发送第二个请求。
- ☞ 当单播请求发送到一个子节点，主节点将进入"等待应答" 状态，同时一个临界超时定时启动。这个超时称为 "响应超时"。它避免主节点永远处于"等待应答" 状态。响应超时的时间依赖于具体应用。
- ☞ 当收到一个应答时，主节点在处理数据之前检验应答。在某些情况下，检验的结果可能为错误。如收到来自非期望的子节点的应答，或接收的的帧错误。在收到来自非期望子节点的应答时，响应超时继续计时；当检测到帧错时，可以执行一个重试。

- ☞ 响应超时但没有收到应答时，则产生一个错误。那么主节点进入“空闲”状态，并发出一个重试请求。重试的最大次数取决于主节点 的设置。
- ☞ 当广播请求发送到串行总线上，没有响应从子节点返回。然而主节点需要进行延迟以便使子节点在发送新的请求处理完当前请求。该延迟被称作“转换延迟”。因此，主节点会在返回能够发送另一个请求的“空闲”状态之前，到“等待转换延迟”状态。
- ☞ 在单播方式，响应超时必须设置到足够的长度以使任何子节点都能处理完请求并返回响应。而广播转换延迟必须有足够的长度以使任何子节点都能只处理完请求而可以接收新的请求。因此，转换延迟应该比响应超时要短。典型的响应超时在 9600bps 时从 1 秒到几秒，而转换延迟从 100 ms 到 200ms。
- ☞ 帧错误包括：1) 对每个字符的奇偶校验; 2) 对整个帧的冗余校验。详细解释参见 §2.6 “差错校验方法”。

状态图以简洁的方式绘出。它没有包含对线路的访问、报文帧及传输错误重试等等。有关帧传输的细节，请参见 2.5 中的图，“两种串行式”。

2.4.2 从站状态图

下图描述了子节点的状态特征:

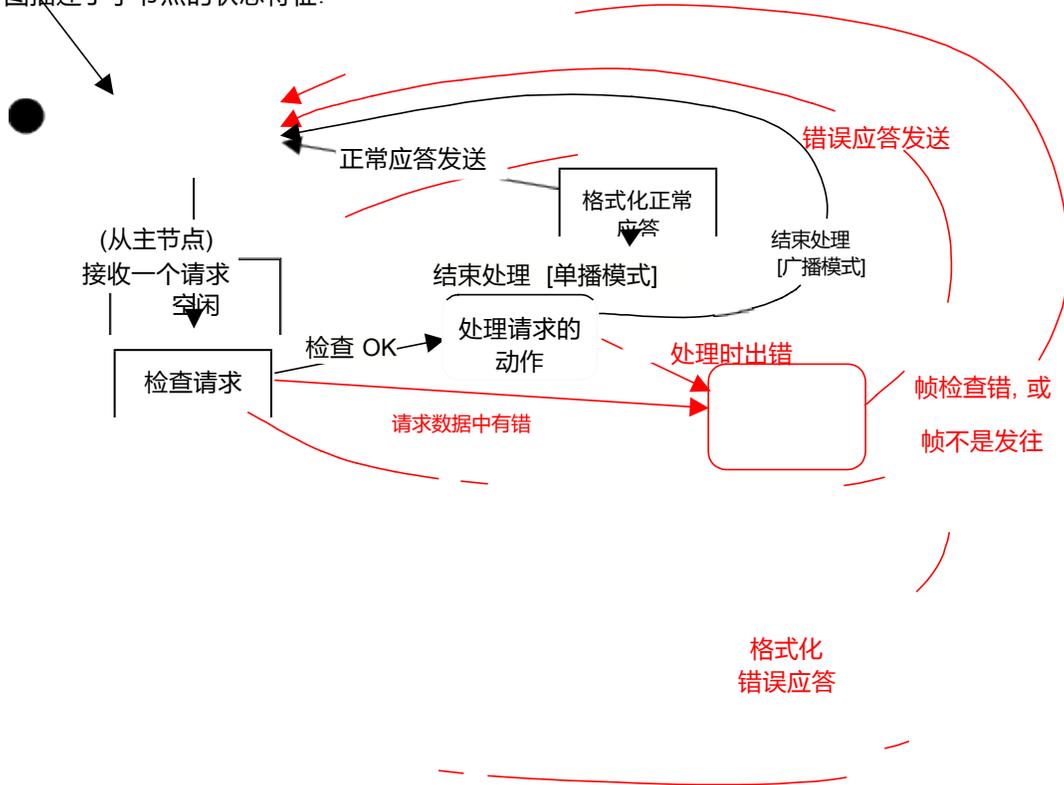


图 5： 子节点状态图

对上面的状态图的一些解释：

- ☞ 状态“空闲”= 没有等待的请求。这是电源上电后的初始状态。

- 👉 当收到一个请求时，子节点在处理请求中要求的动作前检验报文包。不同的错误可以发生于：请求的格式错，非法动作，…… 当检测到错误时，必须 向主节点发送应答。
- 👉 当要求的动作完成后，单播报文要求必须格式化一个应答并发往主节点。
- 👉 如果子节点在接收到的帧中检测到错误，则没有响应返回到主节点。
- 👉 任何子节点均应该定义并管理 Modbus 诊断计数器以提供诊断信息。通过使用 Modbus 诊断功能码，可以得到这些计数值。（参见 附录 A，和 Modbus 应用协议规范 [1]）。

2.4.3 主站/从站通信时序图

下面的图显示了主/从通信的 3 种典型情况。

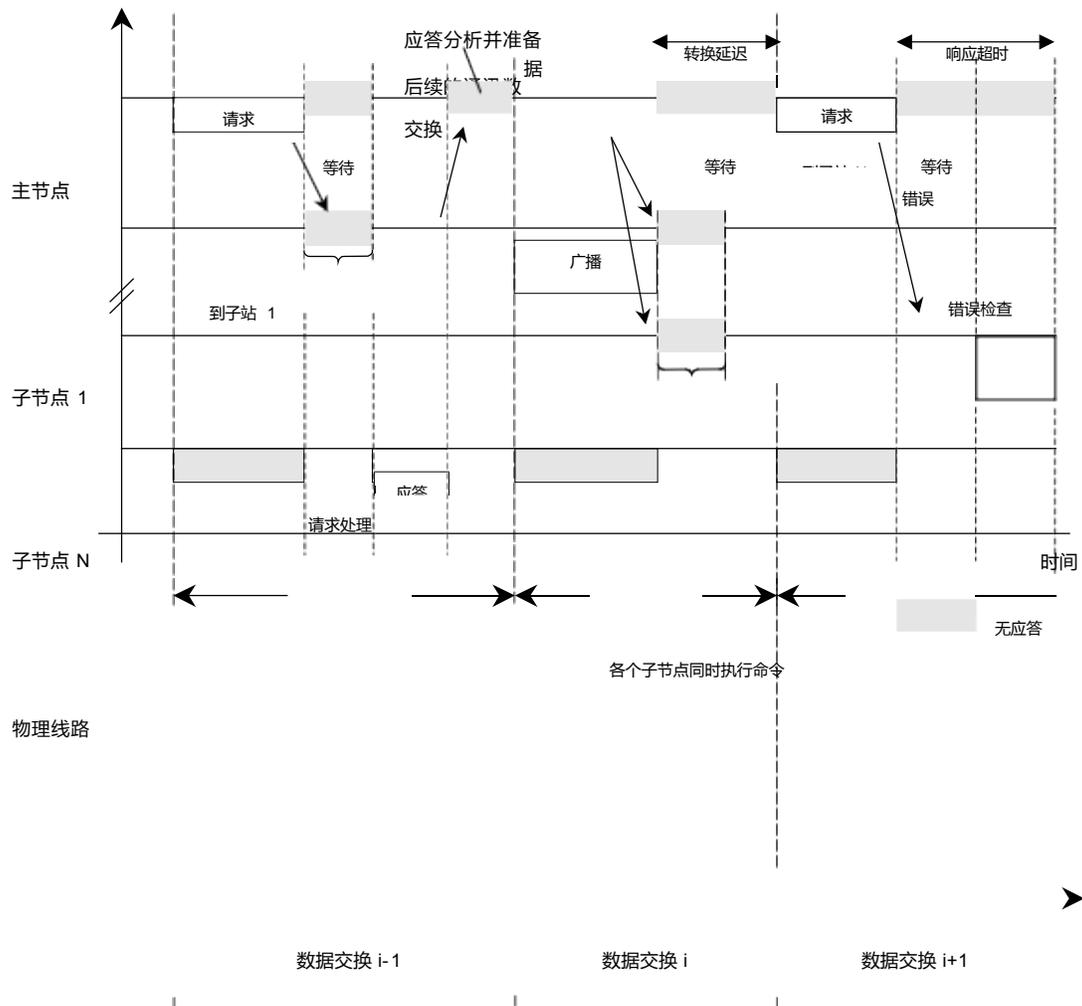


图 6: 各种情形的主/从通信时序图

注：

- 👉 请求， 应答， 广播阶段的持续时间依赖于通信特征（帧长度和吞吐量）。
- 👉 等待和处理阶段的持续时间取决于子节点应用的请求处理时间。

2.5 两种串行传输模式

有两种串行传输模式被定义: RTU 模式 和 ASCII 模式。

它定义了报文域的位内容在线路上串行的传送。它确定了信息如何打包为报文和解码。

Modbus 串行链路上所有设备的传输模式 (和串行口参数) 必须相同。

尽管在特定的领域 ASCII 模式是要求的，但达到 Modbus 设备之间的互操作性只有每个设备都有

相同的模式：**所有设备必须实现 RTU 模式。**

设备应该由用户设成期望的模式，RTU 或 ASCII。ASCII 传输模式是选项。

默认设置必须为 RTU 模式。

2.5.1 RTU 传输模式

当设备使用 RTU (Remote Terminal Unit) 模式在 Modbus 串行链路通信，报文中每个 8 位字节含有两个 4 位十六进制字符。这种模式的主要优点是较高的数据密度，在相同的波特率下比 ASCII 模式有更高的吞吐率。每个报文必须以连续的字符流传送。

RTU 模式每个字节 (11 位) 的格式为：

编码系统: 8-位二进制

报文中每个 8 位字节含有两个 4 位十六进制字符(0-9, A-F)

Bits per Byte: 1 起始位

8 数据位，首先发送最低有效位

1 位作为奇偶校验

1 停止位

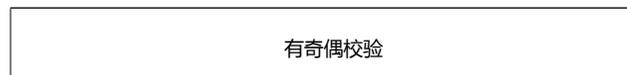
偶校验是要求的，其它模式 (奇校验，无校验) 也可以使用。为了保证与其它产品的最大兼容性，同时支持无校验模式是建议的。默认校验模式必须为偶校验。

注：使用无校验要求 2 个停止位。

字符是如何串行传送的:

每个字符或字节均由此顺序发送(从左到右):

最低有效位 (LSB) . . . 最高有效位 (MSB)



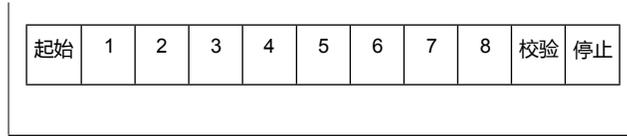


图 7: RTU 模式位序列

设备配置为奇校验、偶校验或无校验都可以接受。如果无奇偶校验，将传送一个附加的停止位以填充字符帧:



图 8: RTU 模式位序列 (无校验的特殊情况)

帧检验域: 循环冗余校验 (CRC)

帧描述 :

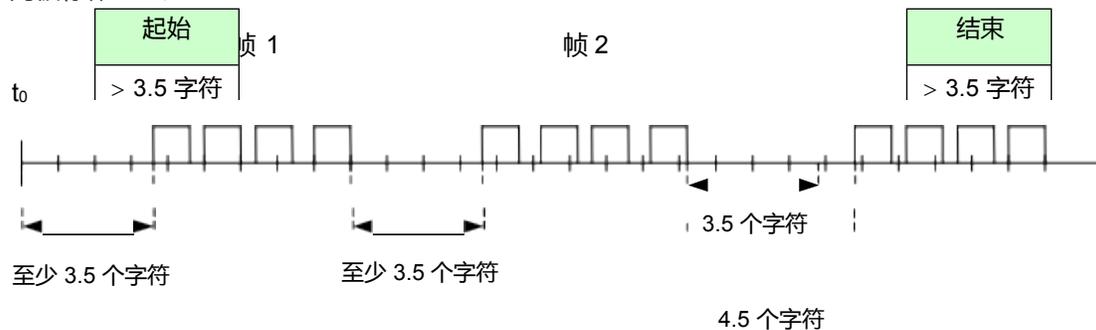
子节点地址	功能代码	数据	CRC
1 字节	1 字节	0 到 252 字节	2 字节 CRC 低 CRC 高

图 9: RTU 报文帧

➔ Modbus RTU 帧最大为 256 字节。

2.5.1.1 Modbus 报文 RTU 帧

由发送设备将 Modbus 报文构造为带有已知起始和结束标记的帧。这使设备可以在报文的开始接收新帧，并且知道何时报文结束。不完整的报文必须能够被检测到而错误标志必须作为结果被设置。在 RTU 模式，报文帧由时长至少为 3.5 个字符时间的空闲间隔区分。在后续的部分，这个时间区间被称作 t_{3.5}。

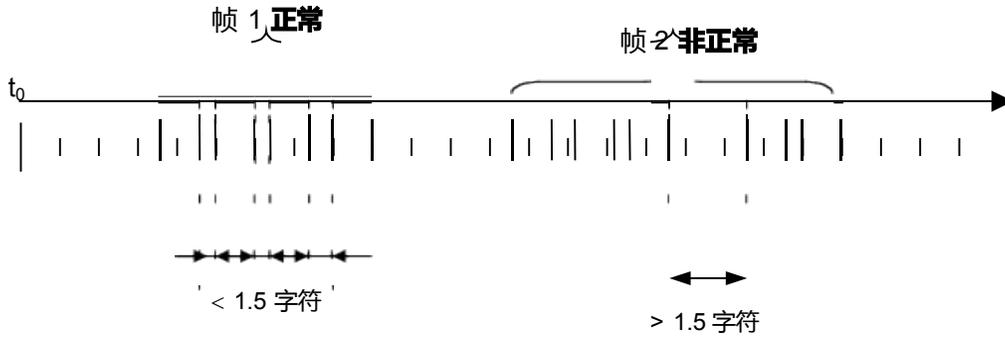


地址	功能代码	数据	CRC 校验
8 位	8 位	N x 8 位	16 位

图 10: RTU 报文帧

整个报文帧必须以连续的字符流发送。

如果两个字符之间的空闲间隔大于 1.5 个字符时间，则报文帧被认为不完整应该被接收节点丢弃。



注：

RTU 接收驱动程序的实现，由于 $t_{1.5}$ 和 $t_{3.5}$ 的定时，隐含着大量的对中断的管理。在高通信速率下，这导致 CPU 负担加重。因此，在通信速率等于或低于 19200 Bps 时，这两个定时必须严格遵守；对于波特率大于 19200 Bps 的情形，应该使用 2 个定时的固定值：建议的字符间超时时间($t_{1.5}$)为 750 μ s，帧间的超时时间 ($t_{1.5}$) 为 1.750ms。

下图表示了对 RTU 传输模式状态“主节点”和“子节点”的不同角度均在相同的图中表示：

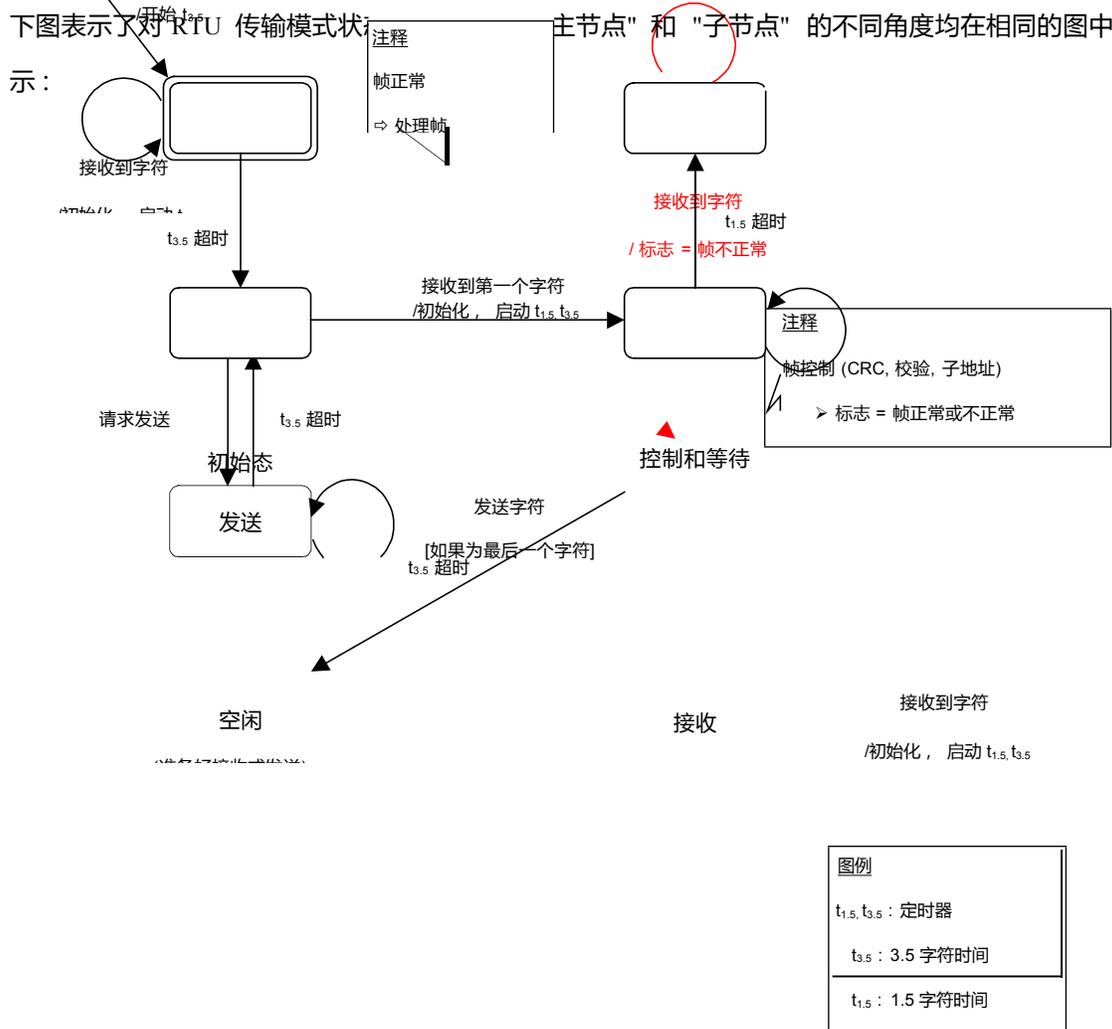


图 11: RTU 传输模式状态图

上面状态图的一些解释:

- 👉 从 "初始" 态到 "空闲" 态转换需要 $t_{3.5}$ 定时超时: 这保证帧间延迟
- 👉 "空闲" 态是没有发送和接收报文要处理的正常状态。
- 👉 在 RTU 模式, 当没有活动的传输的时间间隔达 3.5 个字符长时, 通信链路被认为在 "空闲" 态。
- 👉 当链路空闲时, 在链路上检测到的任何传输的字符被识别为**帧起始**。链路变为 "活动" 状态。然后, 当链路上没有字符传输的时间间隔达到 $t_{3.5}$ 后, 被识别为**帧结束**。
- 👉 检测到帧结束后, 完成 CRC 计算和检验。然后, 分析地址域以确定帧是否发往此设备, 如果不是, 则丢弃此帧。为了减少接收处理时间, 地址域可以在一接到就分析, 而不需要等到整个帧结束。这样, CRC 计算只需要在帧寻址到该节点 (包括广播帧) 时进行。

2.5.1.2 CRC 校验

在 RTU 模式包含一个对全部报文内容执行的，基于循环冗余校验 (CRC - Cyclical Redundancy Checking) 算法的错误检验域。CRC 域检验整个报文的内容。不管报文有无奇偶校验，均执行此检验。

CRC 包含由两个 8 位字节组成的一个 16 位值。

CRC 域作为报文的最后的域附加在报文之后。计算后，首先附加低字节，然后是高字节。CRC 高字节为报文发送的最后一个子节。

附加在报文后面的 CRC 的值由发送设备计算。接收设备在接收报文时重新计算 CRC 的值，并将计算结果于实际接收到的 CRC 值相比较。如果两个值不相等，则为错误。

CRC 的计算，开始对一个 16 位寄存器预装全 1。然后将报文中的连续的 8 位子节对其进行后续的计算。只有字符中的 8 个数据位参与生成 CRC 的运算，起始位，停止位和校验位不参与 CRC 计算。

CRC 的生成过程中，每个 8-位字符与寄存器中的值异或。然后结果向最低有效位(LSB)方向移动(Shift) 1 位，而最高有效位(MSB)位置充零。然后提取并检查 LSB：如果 LSB 为 1，则寄存器中的值与一个固定的预置值异或；如果 LSB 为 0，则不进行异或操作。

这个过程将重复直到执行完 8 次移位。完成最后一次（第 8 次）移位及相关操作后，下一个 8 位字节与寄存器的当前值异或，然后又同上面描述过的一样重复 8 次。当所有报文中子节都运算之后得到的寄存器中的最终值，就是 CRC。

当 CRC 附加在报文之后时，首先附加低字节，然后是高字节。在附录 B 含有 CRC 生成的详细示例。

2.5.2 ASCII 传输模式

当 Modbus 串行链路的设备被配置为使用 ASCII (American Standard Code for Information Interchange) 模式通信时，报文中的每个 8 位子节以两个 ASCII 字符发送。当通信链路或者设备无法符合 RTU 模式的定时管理时使用该模式。

注：由于一个子节需要两个字符，此模式比 RTU 效率低。

→ 例：子节 0X5B 会被编码为两个字符：0x35 和 0x42 (ASCII 编码 0x35 ="5"，0x42 ="B")。

ASCII 模式每个字节 (10 位) 的格式为：

编码系统: 十六进制，ASCII 字符 0-9，A-F。

报文中每个 ASCII 字符含有 1 个十六进制字符

Bits per Byte: 1 起始位

7 数据位，首先发送最低有效位

1 位作为奇偶校验

1 停止位

偶校验是要求的，其它模式 (奇校验，无校验) 也可以使用。为了保证与其它产品的最大兼容性，同时支持无校验模式是建议的。默认校验模式模式 必须为偶校验。

注：使用无校验要求 2 个停止位。

字符是如何串行传送的:

每个字符或字节均由此顺序发送(从左到右):

最低有效位 (LSB) ... 最高有效位 (MSB)



图 12: ASCII 模式位序列

设备配置为奇校验、偶校验或无校验都可以接受。如果无奇偶校验，将传送一个附加的停止位以填充字符帧:



图 13: ASCII 模式模式位序列(无校验的特殊情况)

帧检验域:纵向冗余校验 (LRC - Longitudinal Redundancy Checking)

2.5.2.1 Modbus ASCII 报文帧

由发送设备将 Modbus 报文构造为带有已知起始和结束标记的帧。这使设备可以在报文的开始接收新帧，并且知道何时报文结束。不完整的报文必须能够被检测到而错误标志必须作为结果被设置。

报文帧的地址域含有两个字符。

在 ASCII 模式，报文用特殊的字符区分帧起始和帧结束。一个报文必须以一个‘冒号’（:）(ASCII 十六进制 3A)起始，以‘回车-换行’(CR LF)对 (ASCII 十六进制 0D 和 0A) 结束。

注：LF 字符可以通过特定的 Modbus 应用命令 (参见 Modbus 应用协议规范) 改变。

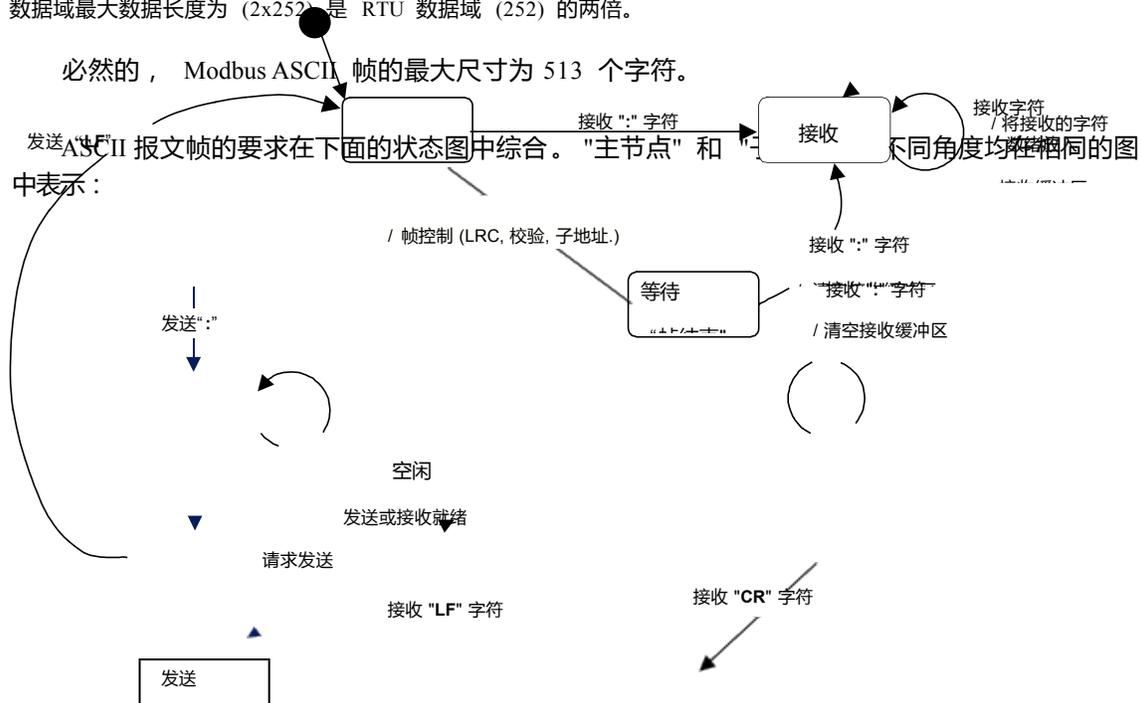
对于所有的域，允许传送的字符为十六进制 0-9，A-F (ASCII 编码)。设备连续的监视总线上的‘冒号’字符。当收到这个字符后，每个设备解码后续的字符一直到帧结束。

报文中字符间的时间间隔可以达一秒。如果有更大的间隔，则接受设备认为发生了错误。下图显示了一个典型的报文帧。

起始	地址	功能	数据	LRC	结束
1 字符 :	2 字符	2 字符	0 到 2x252 字符	2 字符	2 字符 CR,LF

图 14: ASCII 报文帧

注：每个字符字节需要用两个字符编码。因此，为了确保 ASCII 模式和 RTU 模式在 Modbus 应用级兼容，ASCII 数据域最大数据长度为 (2x252) 是 RTU 数据域 (252) 的两倍。



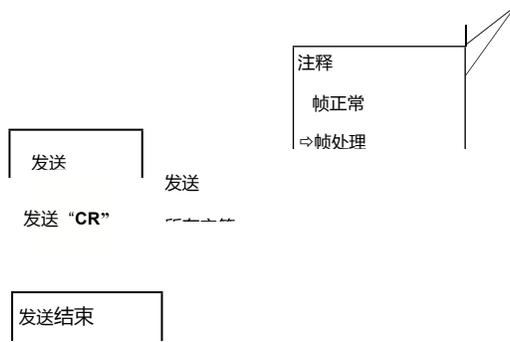


图 15: ASCII 传输模式状态图

上面状态图的一些解释:

- ☞ “空闲” 态是没有发送和接收报文要处理的正常状态。
- ☞ 每次接收到 ":" 字符表示新的报文的开始。如果在一个报文的接收过程中收到该字符，则当前地报文被认为不完整并被丢弃。而一个新的接收缓冲区被重新分配。
- ☞ 检测到帧结束后，完成 LRC 计算和检验。然后，分析地址域以确定帧是否发往此设备，如果不是，则丢弃此帧。为了减少接收处理时间，地址域可以在一接到就分析，而不需要等到整个帧结束。

2.5.2.2 LRC 校验

在 ASCII 模式，包含一个对全部报文内容执行的，基于纵向冗余校验 (LRC - Longitudinal Redundancy Checking) 算法的错误检验域。LRC 域检验不包括起始“冒号”和结尾 CRLF 对的整个报文的内容。不管报文有无奇偶校验，均执行此检验。

LRC 域为一个子节，包含一个 8 位二进制值。LRC 值由发送设备计算，然后将 LRC 附在报文后面。接收设备在接收报文时重新计算 LRC 的值，并将计算结果于实际接收到的 LRC 值相比较。如果两个值不相等，则为错误。

LRC 的计算，对报文中的所有连续 8 位字节相加，忽略任何进位，然后求出其二进制补码。执行检验针对不包括起始“冒号”和结尾 CRLF 对的整个 ASCII 报文域的内容。在 ASCII 模式，LRC 的结果被 ASCII 编码为两个字节并放置于 ASCII 模式报文帧的结尾，CRLF 之前。

附录 B 含有 LRC 生成的详细示例。

2.6 差错检验方法

标准 Modbus 串行链路的可靠性基于两种错误检验:

☞ 奇偶校验 (偶或奇) 应该被每个字符采用。

☞ 帧检验 (LRC or CRC) 必须运用于整个报文。

由设备 (主节点或子节点) 生成的字符检验和帧检验发送前附加于报文体。设备 (子节点或主节点) 在接收时检验每个字符和整个报文。

主节点被用户配置为在放弃事务处理前等待一个预定的超时间隔 (响应超时)。这个间隔被设置成任何子节点有足够的时间正常响应 (单播请求)。如果子节点检测到错误, 则报文不起作用。子节点将不会构造对主节点的响应。因此, 将达到超时时间能使主节点的程序处理错误。注意, 当寻址到不存在的子设备的报文也会导致超时错误。

2.6.1 奇偶检验

用户可以配置设备使用偶 (要求的) 或奇校验, 或无校验 (建议的)。这将确定每个字符的奇偶位如何设置。

无论指定了偶还是奇校验, 则数据部分的为 1 的位的总数被计数 (ASCII 模式 7 数据位, RTU 8 数据位)。而奇偶位会被设置为 0 或 1 以使为 1 的位的总数为偶数或奇数。

例如, RTU 字符帧的数据为:

1100 0101

为 1 的位的总数为 4。如果使用偶校验, 帧的奇偶位为 0, 使为 1 的位的总数仍然为偶数(4); 如果使用奇校验, 帧的奇偶位为 1, 使为 1 的位的总数为奇数(5)。

当报文发送时, 奇偶位被计算并作用于每个字符帧。接收的设备计算为 1 的位的总数, 如果与设备配置不符, 则设置错误标记。(Modbus 串行链路的所有设备必须被配置成使用相同的奇偶检验方法)。注意, 奇偶检验只能检测到一个字符帧在传输过程中奇数个的增加或丢失的位。例如, 假如使用奇校验, 字符帧中含有的 3 个为 1 的位丢失了两个, 而为 1 的位的计数的结果仍然为奇数。

如果没有指定奇偶检验, 奇偶位不会被传送, 也不可以进行奇偶检验: 一个附加的位被传送以填充字符帧。

2.6.2 帧检验

依赖于传输模式, 两种检验方法被使用: RTU 或 ASCII。

☞ 在 RTU 模式, 包含一个对全部报文内容执行的, 基于循环冗余校验 (CRC - Cyclical Redundancy Checking) 算法的错误检验域。CRC 域检验整个报文的内容。不管报文有无奇偶校验, 均执行此检验。

☞ 在 ASCII 模式, 包含一个对全部报文内容执行的, 基于纵向冗余校验 (LRC - Longitudinal Redundancy Checking) 算法的错误检验域。LRC 域检验不包括起始“冒号”和结尾 CRLF 对的整个报文的内容。不管报文有无奇偶校验, 均执行此检验。

有关差错检验方法的详细内容, 参见前面的章节。

3 物理层

3.1 引言

新的串行链路上的 MODBUS 解决方案应该按照 EIA/TIA-485(即已知的 RS485 标准)实现电气接口。该标准允许“两线结构”的点对点和多点系统。此外,某些设备可能实现“四线”RS485 接口。

设备也可能实现 RS232 接口。

在这种 MODBUS 系统中,一个主站和一个或几个从站在一个无源串行链路上通信。

在标准的 MODBUS 系统中,所有设备(并行)连结在一条由 3 条导线组成的干线电缆上。其中两条导线(“两线”结构)形成一对平衡双绞线,双向数据在其上传送,典型比特率为每秒 9600 比特。

每台设备可能连结(见图 19):

- 或是双向连到主干电缆上,形成菊花链,
- 或是经分支电缆连到一个无源接头上,
- 或是经特种电缆连到一个有源接头上。

在设备上可用螺钉端子, RJ45, 或 9 芯 D-型连接器与电缆相接(见“机械接口”章节)。

3.2 数据信号发送速率

要求 9600bps 波特率。

推荐 19200bps 波特率。

该值(19200)必须被作为约定值来实现。

其它波特率可选择来实现: 1200, 2400, 4800, ... 38400bps, 56Kbps, 115Kbps, ...

每种波特率,对发送方,要求其精度必须高于 1%,而对接收方,必须允许 2%误差。

3.3 电气接口

3.3.1 多点串行总线结构

图 19 展现的是 MODBUS 多点串行链路系统中串行总线结构的总貌。

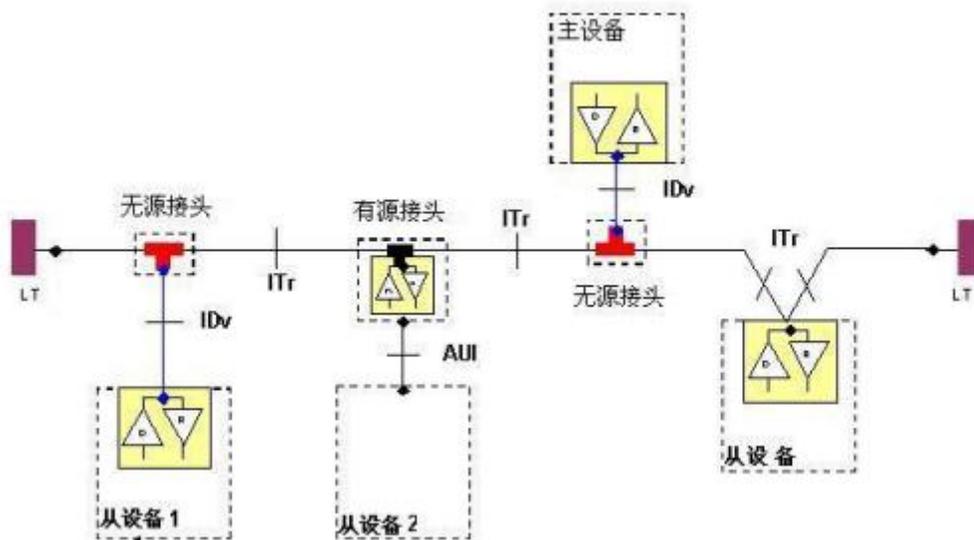


图 19: 串行总线基本结构

一个 MODBUS 多点串行链路系统是由主电缆(主干), 和一些可能的分支电缆组成。

在主干电缆的两端需要有线路终端以使阻抗匹配 (详见§ “2 线-MODBUS 定义” 和 “可选 4 线-MODBUS 定义”)。

如图 19 所示, 不同的设备可以在同一个 MODBUS 串行链路系统中运行:

- ☞ 集成有通信收发器的设备通过**无源接头**和分支电缆连接到主干上(例如从站 1 和主站);
- ☞ 没有集成通信收发器的设备通过**有源接头**和分支电缆连接到主干上(有源接头集成有收发器)(例如从站 2);
- ☞ 设备以**菊花链形式**直接连接到主干电缆上(例如从站 n)

我们采用下列规定:

- ☞ 主干间的接口称为 **ITr**(主干接口)
- ☞ 设备和**无源接头**间的接口称为 **IDv** (分支接口)
- ☞ 设备和有源接头间的接口称为 **AUI** (附加单元接口)

注:

1. 某些情况下, 接头可能直接连接到设备的 IDv-插槽或 AUI-插槽上, 而不使用分支电缆。
2. 一个接头可能有几个 IDv 插槽以连接几台设备。当它是无源接头时, 称为分配器。

3. 当使用有源接头时，可以通过接头的 AUI 或 ITr 接口向其提供电源。

在接下来的章节中，我们将介绍 ITr 和 IDv 接口(见§ “2 线-MODBUS 定义” 和 “4 线-MODBUS 定义”)

3.3.2 2 线-MODBUS 定义

串行链路上的 MODBUS 解决方案应当依照 EIA/TIA-485 标准实现 “2-线” 电气接口。

在这个 2 线-总线上，在任何时候只有一个驱动器有权发送信号。

实际上，还有第三条导线把总线上所有设备相互连接：公共地。

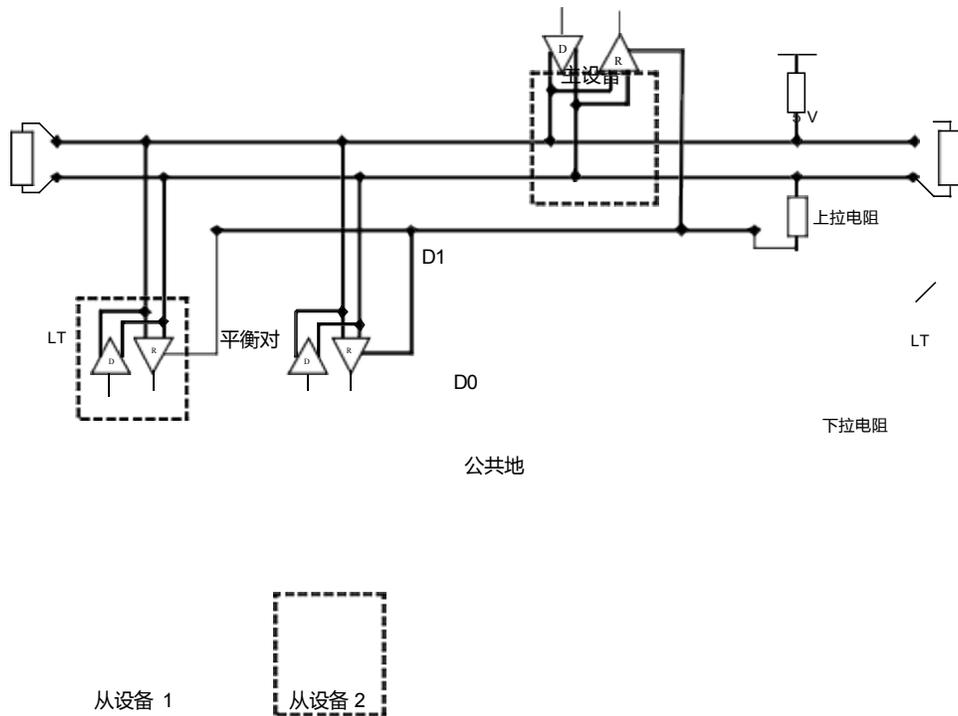


图 20: 2-线制的一般拓扑结构

2 线-MODBUS 电路定义

所需电路		设备	设备需求	EIA/TIA-485 的命名	说明
在 ITr 上	在 IDv 上				
D1	D1	I/O	X	B/B'	收发器端子 1, V1 电压 (V1 > V0 表示二进制 1[OFF]状态)

D0	D0	I/O	X	A/A'	收发器端子 0, V0 电压 (V0 > V1 表示二进制 0[ON]状态)
公共地	公共地	--	X	C/C'	信号和可选的电源公共地

注:

- . 对于线路终端 (TL), 即上拉和下拉电阻, 请参考 “多点系统要求” 部分。
- . 在与设备和接头有关的文件 (用户指南, 连线指南, ...) 中, 必须使用 D0, D1 和公共地的电路名字, 以提高互操作能力。
- . 可以增加可选的电气接口, 例如:
 - ☞ **电源:** 5.24 V 直流电。
 - ☞ **端口模式控制:** PMC 电路 (TTL 兼容)。当需要的时候, 可由这个外电路和/或另一种方式 (如该

设备上的一个开关)来

控制端口模式。在第一种情况下，尽管一个开路 PMC 将要求 2-线MODBUS 模式，但实际上，在 PMC 上的低电平将把端口置于 4-线 MODBUS 模式或RS232-MODBUS 模式。

3.3.3 可选的 4 线-MODBUS 定义

这种 MODBUS 设备同样允许实现 2 对总线(4 线)单向数据传输。在**主对总线**(RXD1-RXD2)上的数据只能由从站接收，而在**从对总线**(TXD0-TXD1)上的数据只能由主站接收。

实际上，公共地作为第五条导线必须把 4-线总线上的所有设备相互连接。

和 2 线-MODBUS 一样，在任何时刻只有一个驱动器有权利发送数据。

这种设备必须**依照 EIA/TIA-485** 对每一对平衡线实现一个驱动器和一个收发器。(有时候这种方式被称为“RS422”，这是错误的：RS422 标准不支持几台设备在一对平衡线上。)

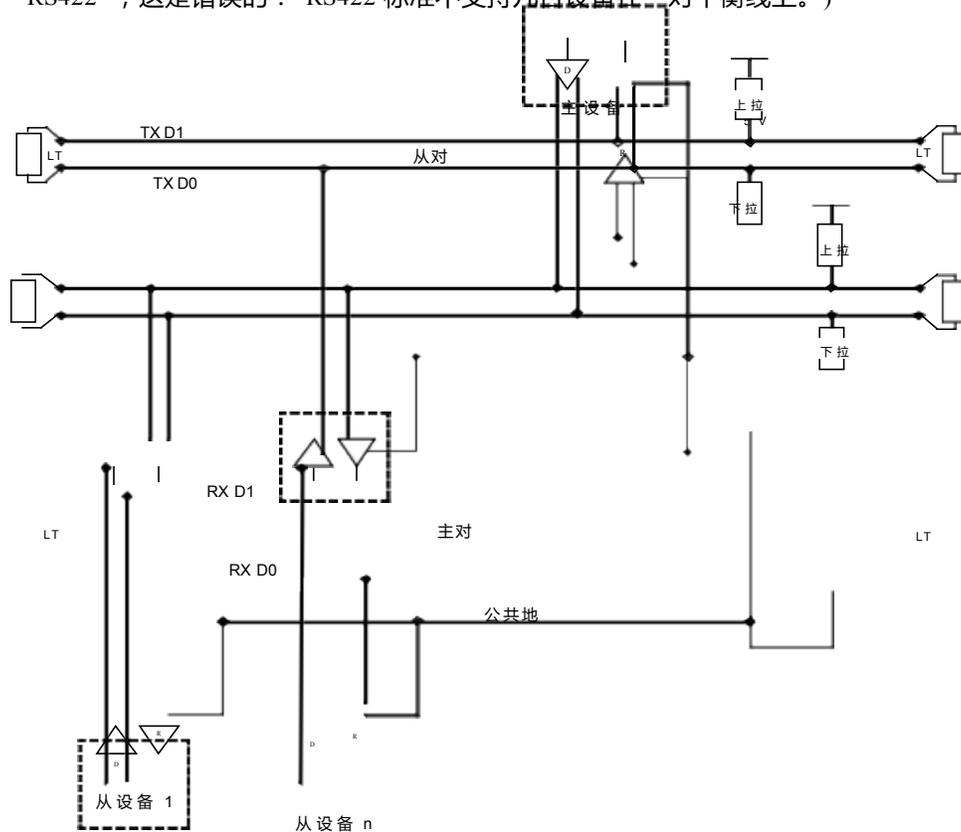


图 21: 4-线制的一般拓扑结构

以上内容仅为本文档的试下载部分，为可阅读页数的一半内容。

如要下载或阅读全文，请访问：

<https://d.book118.com/707055163110010004>