ICS 25.040
N 10

中华人民共和国国家标准

GB/T 19582.2—2008
代替 GB/Z 19582.2—2004

基于 Modbus 协议的工业自动化网络规范 第 2 部分：Modbus 协议在串行链路上的实现指南

Modbus industrial automation network specification—Part 2: Modbus protocol implementation guide over serial link

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

《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：

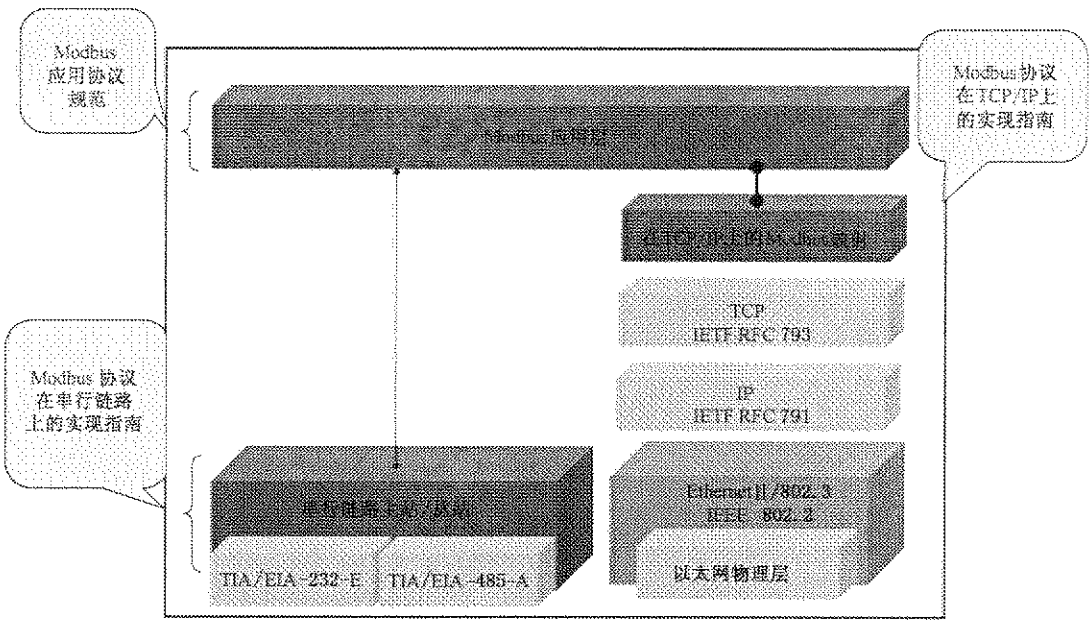
——串行链路上的 Modbus

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 19582.2—2004；GB/Z 19582.2—2004 并于该日起予以废止。

本部分的附录 A、附录 B 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、华锫、王勇。

GB/Z 19582.2 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 19582.2—2004基本一致，但在技术内容上对GB/Z 19582.2—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 2 部分：Modbus 协议在串行链路上的实现指南

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。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。

本部分的目标是提出串行链路上的 Modbus 协议，以便系统设计者在实现基于串行链路的 Modbus 协议时使用。

本部分将促进使用 Modbus 协议的设备之间的互操作性。

本部分还是对 GB/T 19582.1—2008 的补充，具体见图 1。

在第 9 章中定义了“Modbus 串行链路”的不同实现等级。等级的规定是设备能够属于某个等级而必须遵守的全部要求。

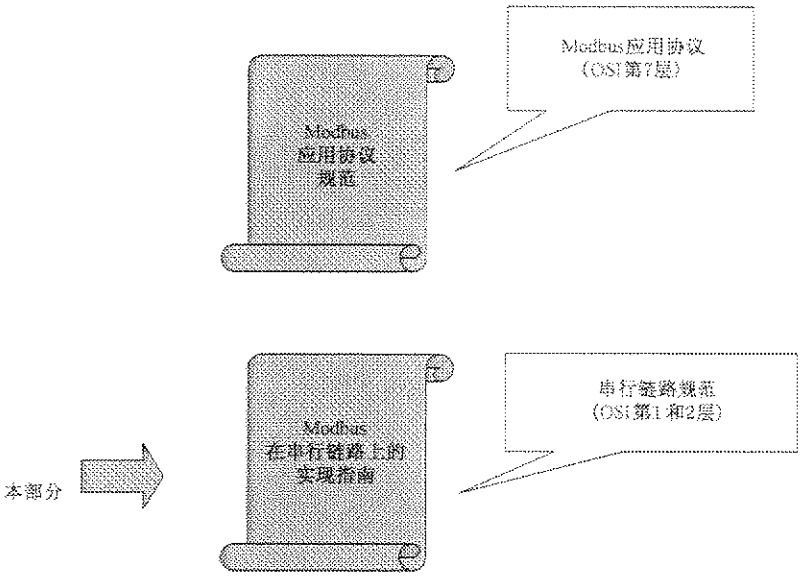


图 1 Modbus 协议文本的概述

2 协议概述

本部分描述串行链路上的 Modbus 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 2 层。

主—从类型的系统有一个主节点(主站)，它向某个从节点(从站)发出显式命令并处理响应。从站在没有收到主站的请求时并不主动地传输数据，也不与其他从站通信。

在物理层，串行链路上的 Modbus 系统可以使用不同的物理接口(RS485、RS232)。最常用的物理接口是 TIA/EIA-485(RS485)二线制接口。作为附加选项，该物理接口也可以使用 RS485 四线制接口。当只需要近距离的点对点通信时，也可以使用 TIA/EIA-232-E(RS232)串行接口作为 Modbus 系统的物理接口(见第 7 章)。

图 2 给出了与 7 层 OSI 模型对应的 Modbus 串行通信栈的一般表示。

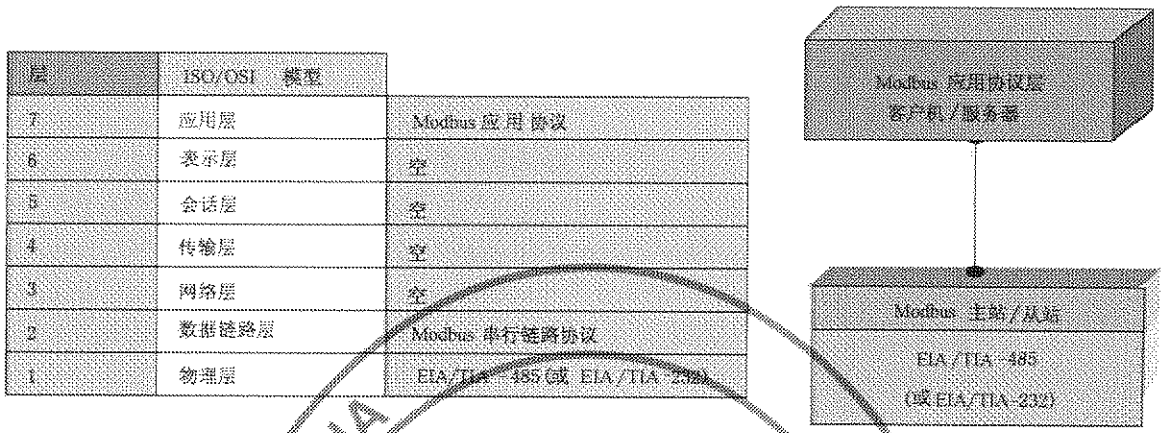


图 2 Modbus 协议和 ISO/OSI 模型

位于 OSI 模型第 7 层的 Modbus 应用层报文传输协议提供了总线或网络上连接的设备之间的客户机/服务器通信。在 Modbus 串行链路上，串行总线的主站作为客户机，从站作为服务器。

3 约定

在本部分中，使用下列词语定义每个特定要求的重要程度。

——“必须”/“要求的”

含有词语“必须”的所有要求是强制的。词语必须或形容词“要求的”表示该项为执行的绝对要求。这些词语带有下划线。

——“应该”/“建议的”

包含“应该”或形容词“建议的”的所有建议是期望的功能。应该使用这些建议作为选择不同的实现功能选项时的指南。在特定条件下，可以有合理的理由忽略这些项目，但是，应该理解其全部含义，并且在选择不同过程之前仔细考虑各种情况。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”表示该项目为真正意义上可选的。例如：一个设计者由于特定的市场需求或增强产品功能，可以选择包含该项目，而另一个设计者可以选择忽略该项目。

4 一致性

如果某个实现不满足实现等级中的一个或多个必须的要求，那么这个实现不符合一致性。

如果某个实现满足实现等级中的所有必须的要求和所有应该的建议，那么称这个实现为“无条件符合一致性”。

如果某个实现满足实现等级中的所有必须的要求但不满足所有应该的建议，那么称这个实现为“有条件符合一致性”。

5 术语和缩略语

本部分中使用下列特定词语、符号和缩略语的定义。

2W(2-Wire)	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(4-Wire)	在“电气接口”一章中定义的四线制配置，或其中的一个接口
AUI(Attachment Unit Interface)	附属单元接口
AWG (American Wire Gauge)	美国线规，表示线径的标准方法(参见参考文献[3])

公共端(Common)	EIA/TIA 标准中的信号公共端。在 2 线制或 4 线制 RS485 Modbus 网络中,信号和可选电源的公共端
DCE(Data Communication Equipment)	数据通信设备。一个实现了 RS232 数据电路终端设备的 Modbus 设备,例如:可编程序控制器适配器
设备(Device)	同 Modbus 设备定义
驱动器(Driver)	发生器或发送器
DTE(Data Terminal Equipment)	数据终端设备。一个实现了 RS232 数据终端设备的 Modbus 设备。例如:可编程终端或 PC
ITr(Interface on trunk)	干线侧的物理总线接口
IDv(Interface on Derivation)	设备侧(或分支器或设备分支)的物理总线接口
LT(Line Termination)	线路终端
Modbus 设备(Modbus Device)	在串行链路上实现 Modbus 并遵循其技术规范的设备
RS232	EIA/TIA-232 标准
RS485	EIA/TIA-485 标准
RS485-Modbus	遵循其技术规范的 2 线制或 4 线制网络
收发器(Transceiver)	发送器和接收器(或驱动器和接收器)

6 Modbus 数据链路层

6.1 Modbus 主/从协议原理

Modbus 串行链路协议是一个主-从协议。在同一时间,总线上只能有一个主站,和一个或多个(最多 247 个)从站。Modbus 通信总是由主站发起。当从站没有收到来自主站的请求时,不会发送数据。从站之间不能相互通信。主站同时只能启动一个 Modbus 事务处理。

主站用两种模式向从站发出 Modbus 请求:

——单播模式(见图 3),主站寻址单个从站。从站接收并处理完请求之后,向主站返回一个报文(一个"应答")。

在这种模式下,一个 Modbus 事务处理包含 2 个报文:一个是主站的请求,另一个是从站的应答。

每个从站必须有唯一的地址(1～247),这样才能区别于其他站独立地被寻址。

——广播模式(见图 4),主站可以向所有的从站发送请求。

对于主站发送的广播请求没有应答返回。广播请求必须是写命令。所有设备必须接受广播方式的写命令。地址 0 被保留用来识别广播通信。

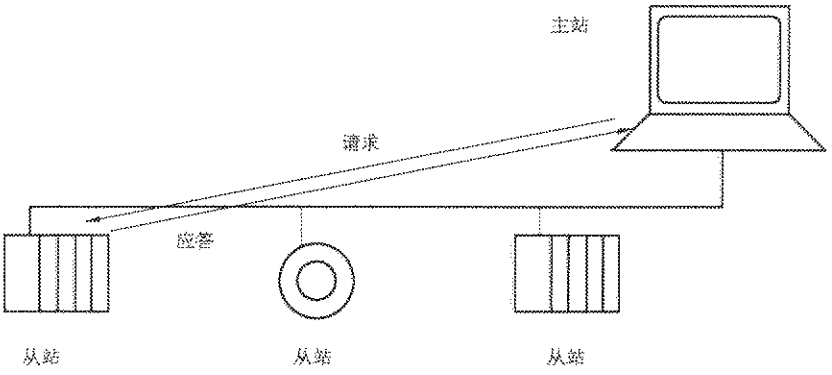


图 3 单播模式

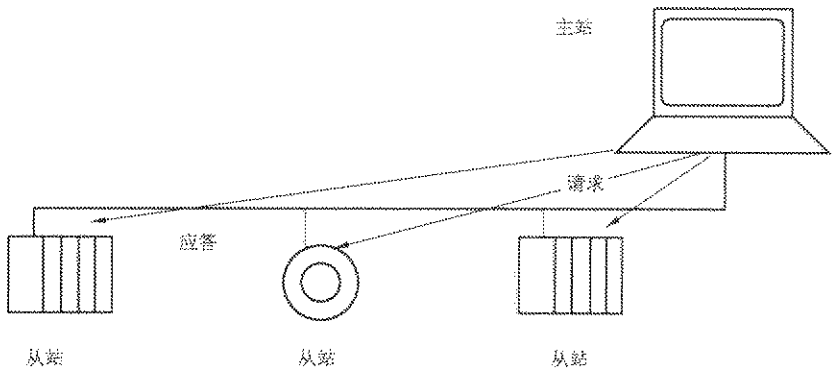


图 4　广播模式

6.2　Modbus 寻址规则

见表 1。

Modbus 寻址空间由 256 个不同地址组成。

地址 0 为广播地址。所有从站必须识别广播地址。

表 1　Modbus 寻址范围

0	1～247	248～255
广播地址	从站地址	保留

Modbus 主站没有特定地址，只有从站有一个地址。在 Modbus 串行总线上，这个地址必须是唯一的。

6.3　Modbus 帧描述

见图 5 和图 6。

Modbus 应用协议定义了一个与下层通信无关的简单协议数据单元(PDU)。

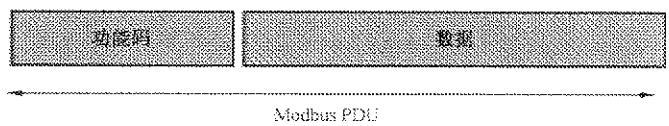


图 5　Modbus 协议数据单元

通过在协议数据单元(PDU)上增加一些附加字段完成 Modbus 协议到具体总线或网络的映射，启动 Modbus 事务处理的客户机构造 Modbus PDU，然后添加附加字段，以便构造相应的通信 PDU。

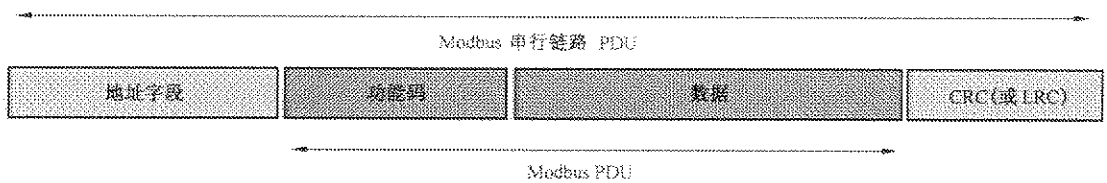


图 6　串行链路上的 Modbus 帧

——在 Modbus 串行链路上，地址字段只含有从站地址。

如前面各节所述，有效的从站地址范围为十进制 0～247。在 1～247 范围中为每个从站指配单独的地址。主站通过将从站地址放置在报文地址字段中来寻址从站。当从站返回响应时，它将自己的地址放到响应地址字段中，以便使主站知道哪个从站正在响应。

——功能码指示服务器要执行何种操作。功能码的后面是含有请求或响应参数的数据字段。

——差错检验字段是根据报文内容执行“冗余校验”计算的结果。根据使用的传输模式(RTU 或 ASCII)，有两种校验计算方法(见 6.5，两种串行传输模式)。

6.4 主站/从站状态图

Modbus 数据链路层由两个独立子层组成：

——主/从协议；

——传输模式(RTU 和 ASCII 模式)。

下列各节描述了与所使用传输模式无关的主站和从站状态图。

在 6.5 中使用两个状态图详细说明了 RTU 和 ASCII 传输模式。描述了帧的接收和发送。

状态图语法 ：

使用 UML(统一建模语言)标准表示法绘制下列状态图(见图 7)。表示法要点如下：

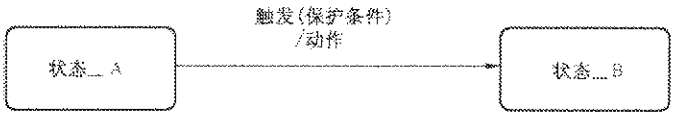


图 7 UML 表示

当处于“状态_A”的系统发生“触发”事件时，只有当“保护条件”为真时，系统才会进入“状态_B”。然后，执行一个“动作”。

6.4.1 主站状态图

图 8 说明了主站的行为。

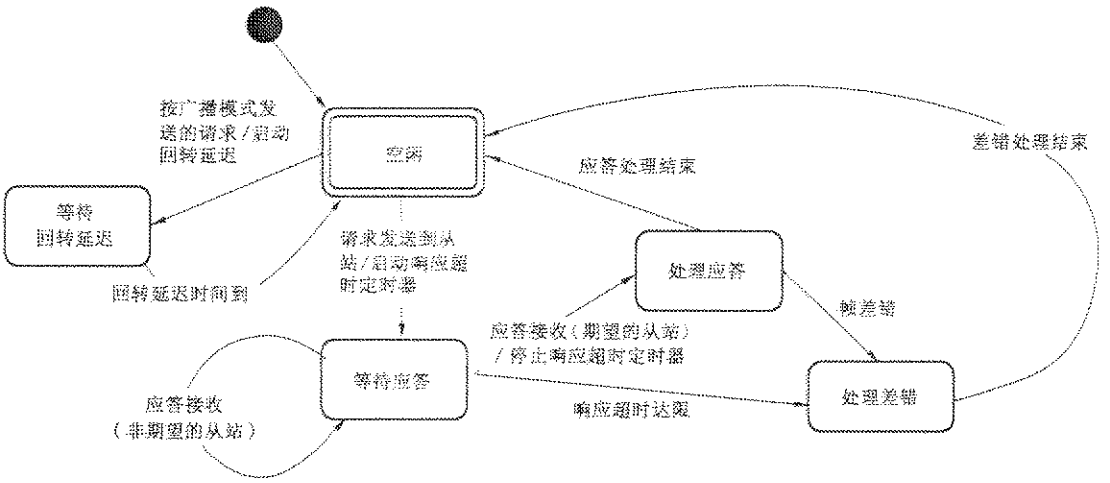


图 8 主站状态图

上述状态图的一些解释：

——状态“空闲”＝无挂起请求。这是电源加电后的初始状态。只有在“空闲”状态下才能发送请求。发送一个请求之后，主站离开“空闲”状态，并且不能同时发送第二个请求。

——当向从站发送单播请求时，主站将进入“等待应答”状态，并且启动一个“响应超时”定时器。它防止主站无限期地停留在“等待应答”状态下。响应超时的时间与具体应用有关。

——当收到一个应答时，主站在处理数据之前检验应答。在某些情况下，检验的结果发现错误，例如：收到来自非期望从站的应答或在接收到的帧中有错误。当收到来自非期望从站的应答时，响应超时继续计时。如果在帧上检测到差错，可以进行重试。

——如果没有接收到应答，响应超时时间到，产生一个错误。然后，主站进入“空闲”状态，并发出一个重试请求。重试的最大次数与主站设置有关。

——当在串行总线上发送广播请求时，从站不返回响应。然而，主站需要考虑延迟，以便在发送新的请求之前允许从站处理当前请求。这个延迟被称作“回转延迟”。因此，在返回“空闲”状态并且能够发送另一个请求之前，主站进入“等待回转延迟”状态。

——在单播模式下，必须设置足够长的响应超时时间，以便从站处理请求并返回响应；在广播模式下，必须有足够长的回转延迟，以便从站处理请求并能够接收新请求。因此，回转延迟应该比响应超时短。通信速率在 9600 bit/s 时，典型的响应超时值为 1 s 到几秒，而回转延迟为 100 ms～200 ms。

——帧错误校验包括：1)每个字符的奇偶校验；2)整个帧的冗余校验。详细解释见 6.6。

状态图本身设计的非常简单。它没有考虑对链路的访问、报文组帧及在传输错误之后的重试等。有关帧传输的细节见 6.5。

6.4.2 从站状态图

图 9 说明了从站的行为。

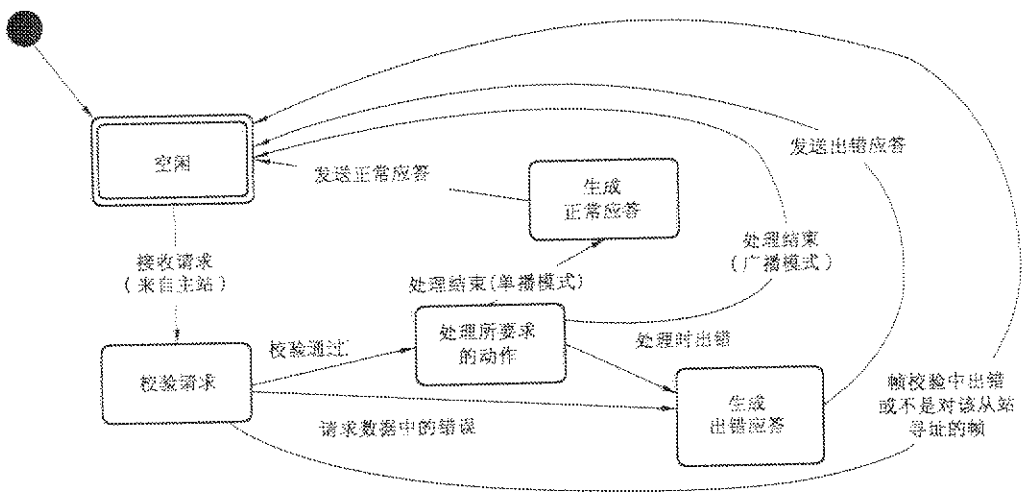


图 9 从站状态图

上述状态图的解释：

——状态“空闲”=无挂起请求。这是设备上电后的初始状态。

——当收到一个请求时，在处理报文包中所请求的动作之前，从站校验报文包。可能出现不同错误：如请求的格式错误、无效动作等。当检测到错误时，必须向主站发送应答。

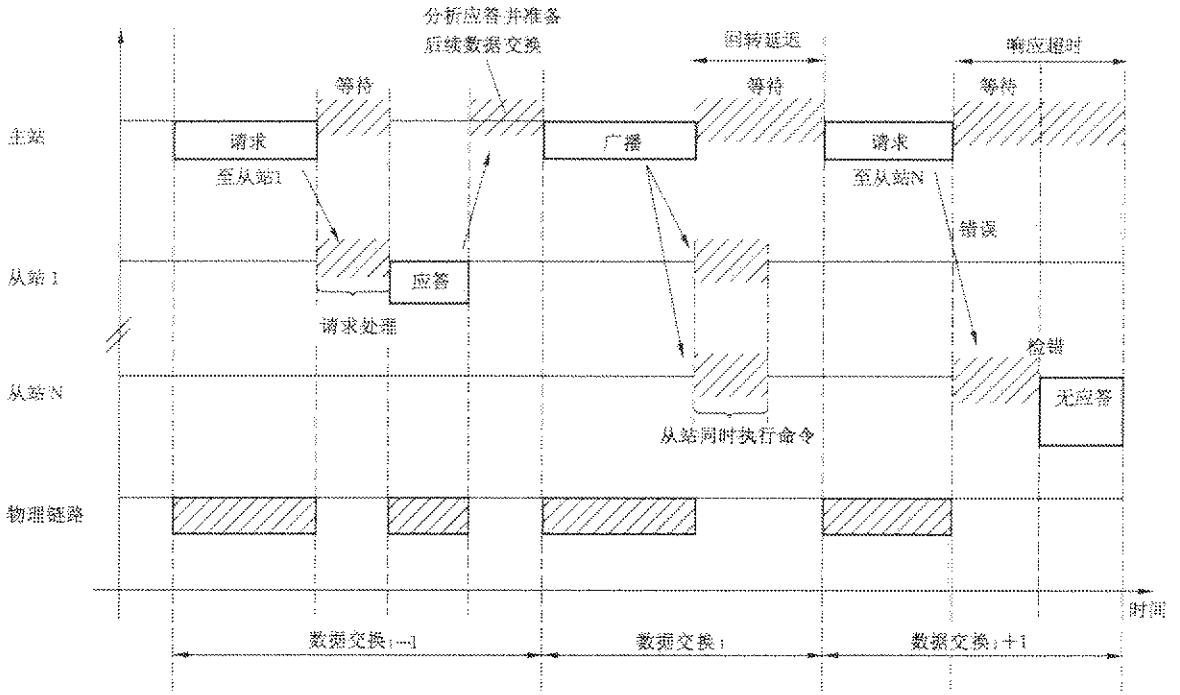
——一旦完成请求的动作，单播报文要求必须生成应答并将其发送给主站。

——如果从站检测到接收帧中的错误，那么不向主站返回响应。

任何从站应该定义并管理 Modbus 诊断计数器，以便提供诊断信息。通过使用 Modbus 诊断功能码可以得到这些计数值(见附录 A 和 GB/T 19582.1—2008)。

6.4.3 主站/从站通信时序图

图 10 表示了 3 种典型的主站/从站通信的时序图。



注1：请求、应答、广播阶段的持续时间与通信特征(帧长度和吞吐量)有关。

注2：等待和处理阶段的持续时间与从站应用所需的请求处理时间有关。

图 10　主站/从站通信时序图

6.5　两种串行传输模式

定义了两种串行传输模式：RTU 模式和ASCII 模式。

定义了链路上串行传送报文的位内容。它确定了信息如何打包为报文和如何解码。

在 Modbus 串行链路上，所有设备的传输模式(及串行口参数)必须相同。

尽管在某些特定应用中要求 ASCII 模式，但只有每个设备都有相同的模式才能进行 Modbus 设备之间的互操作：所有设备必须实现 RTU 模式。ASCII 传输模式是一个可选项。

用户应该将设备设置成所期望的模式：RTU 或 ASCII 模式。默认设置必须为 RTU 模式。

6.5.1　RTU 传输模式

当设备在 Modbus 串行链路上使用 RTU(远程终端单元)模式通信时，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有较高的字符密度，在相同的波特率下，比 ASCII 模式有更高的数据吞吐量。每个报文必须以连续的字符流传输。

RTU 模式中每个字节(11 位)的格式为：

编码系统：　　8 位二进制

每个字节的位：　1 个起始位

8 个数据位，首先发送最低有效位

1 个奇偶校验位

1 个停止位

要求使用偶校验。也可以使用其他模式(奇校验、无校验)。为了保证与其他产品的最大兼容性，建议还支持无校验模式。默认校验模式必须是偶校验。

注：使用无校验时要求 2 个停止位。

串行地传送字符的方法为：

发送每个字符或字节的顺序是从左到右(见图 11)：

最低有效位(LSB)…最高有效位(MSB)

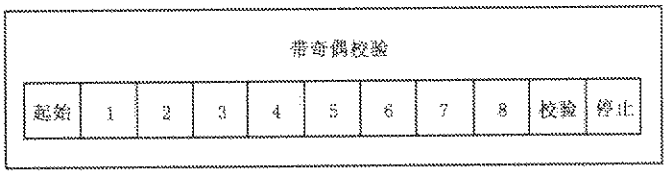


图 11 RTU 模式中的位序列

通过配置,设备可以接受奇校验、偶校验或无校验。如果无校验,那么传送一个附加的停止位来填充字符帧使其成为完整的 11 位异步字符(见图 12、图 13)。

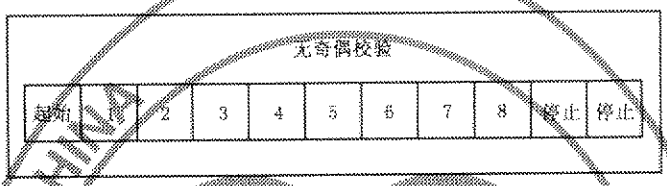


图 12 RTU 模式中的位序列(无校验的特殊情况)

帧校验字段:循环冗余校验(CRC)

帧描述:

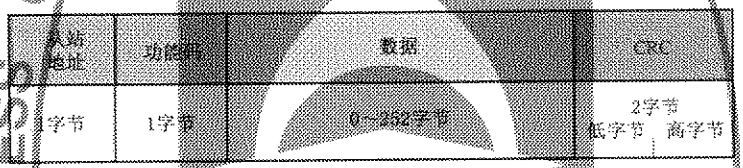


图 13 RTU 报文帧

Modbus RTU 帧最大长度是 256 个字节。

6.5.1.1 **Modbus报文 RTU 帧**

传送设备将 Modbus 报文放置在带有已知起始和结束点的帧中,这就允许接收新帧的设备在报文的起始处开始接收,并且知道报文传输何时结束。必须能够检测到不完整的报文,并且必须设置错误标志。

在 RTU 模式中,时长为 3.5 个字符时间的空闲间隔将报文帧区分开。在后续部分中,这个时间间隔称为 $t_{3.5}$,见图 14。

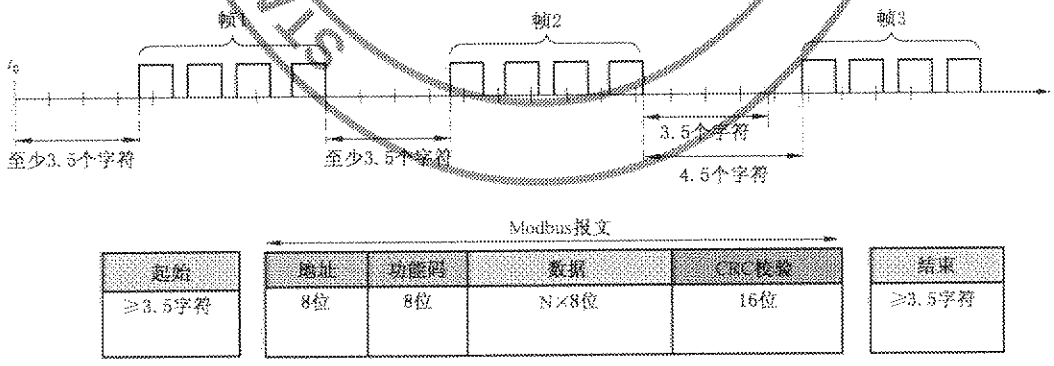


图 14 RTU 报文帧

必须以连续的字符流发送整个报文帧。

如果两个字符之间的空闲间隔大于 1.5 个字符时间,那么认为报文帧不完整,并且接收站应该丢弃这个报文帧,见图 15。

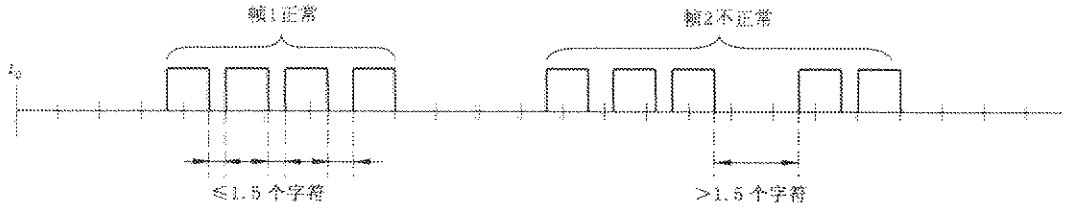
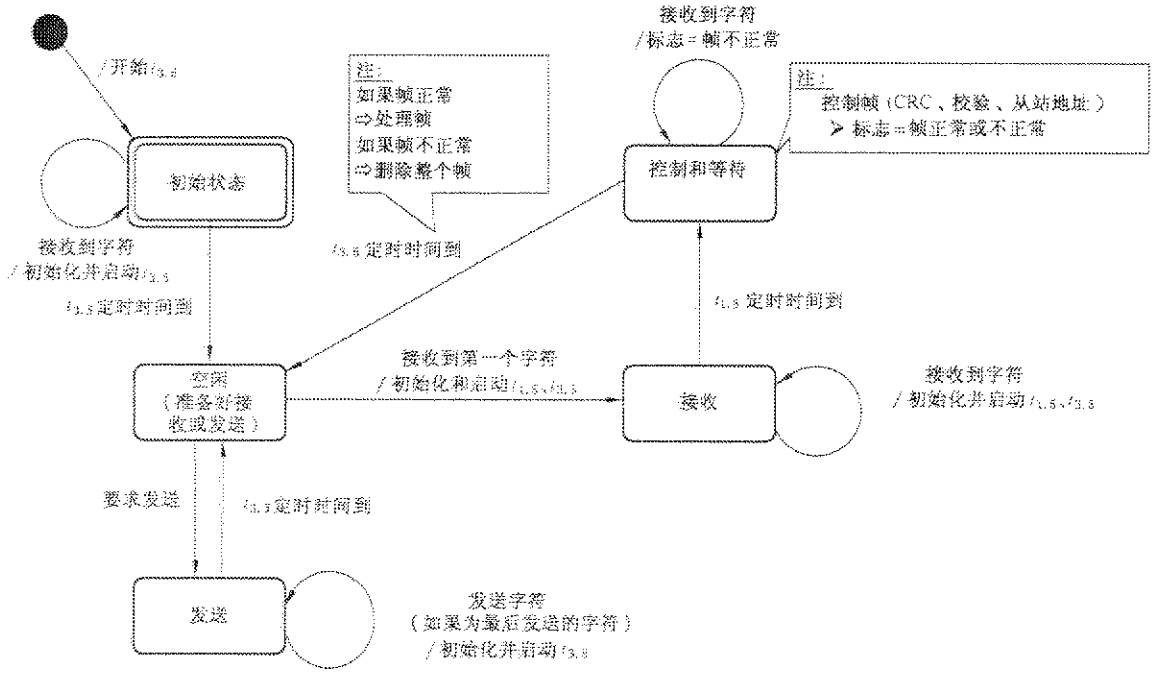


图 15　Modbus 帧内间隔

注：实现了 RTU 接收的驱动程序会隐含着对由 $t_{1.5}$ 和 $t_{3.5}$ 定时器引起的大量中断的管理。在较高的通信波特率下，这将导致 CPU 负担加重。因此，当波特率等于或低于 19 200 bit/s 时，必须严格地遵守这两个定时；波特率大于 19 200 bit/s 的情况下，两个定时器宜使用固定值：建议字符间超时时间（$t_{1.5}$）为 750 μs，帧间的延迟时间（$t_{3.5}$）为 1.750 ms。

图 16 描述了 RTU 传输模式的状态图。“主站”和“从站”均在这个图中表示：



注：$t_{1.5}$、$t_{3.5}$：定时器；

$t_{3.5}$：3.5 个字符时间；

$t_{1.5}$：1.5 个字符时间。

图 16　RTU 传输模式的状态图

上述状态图的解释：

——从“初始”状态到“空闲”状态转换需要 $t_{3.5}$ 定时器超时：这保证帧间延迟。

——当没有发送和接收活动时，“空闲状态”是一个正常状态。

——在 RTU 模式中，当至少 3.5 个字符的时间间隔之后没有传输活动时，称通信链路为“空闲”状态。

——当链路在空闲状态时，在链路上检测到的任何传输的字符被视为帧起始。链路进入“激活”状态。然后，当在时间间隔 $t_{3.5}$ 之后链路上还没有传输字符时，视为帧结束。

——检测到帧结束之后，执行 CRC 计算和校验。然后分析地址字段来确定帧是否发往这个设备。如果不是发往这个设备，那么丢弃这个帧。为了减少接收处理时间，在接收到地址字段时，就可以分析地址字段，而不需要等到整个帧结束。这样，CRC 计算和校验只需要在帧寻址到该从站（包括广播帧）时进行。

6.5.1.2 CRC 校验

RTU 模式包含一个差错校验字段，该字段是基于循环冗余校验(CRC)方法对全部报文内容执行的。

CRC 字段校验整个报文的内容。无论单个字符报文使用何种奇偶校验方式，均应用这种 CRC 校验。

CRC 包含两个 8 位字节组成的一个 16 位值。

CRC 字段作为报文的最后字段附加到报文上。当进行这种附加时，首先附加字段的低位字节，然后附加字段的高位字节。CRC 高位字节是报文中发送的最后字节。

将 CRC 附加到报文上的发送设备计算 CRC 值。在接收报文过程中，接收设备重新计算 CRC 值，并将计算值与 CRC 字段中接收到的实际 CRC 值相比较。如果两个值不相等，则产生错误结果。

通过对一个 16 位寄存器预装载全 1 来启动 CRC 计算。然后，开始将后续报文中的 8 位字节与当前寄存器中的内容进行计算。只有每个字符中的 8 个数据位参与生成 CRC 的计算。起始位、停止位和校验位不参与 CRC 计算。

在生成 CRC 过程中，每个 8 位字符与寄存器中的值异或。然后，向最低有效位(LSB)方向移动这个结果，而用零填充最高有效位(MSB)。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)移位之后，下一个 8 位字节与寄存器的当前值异或，然后像上面描述的那样重复 8 次这个过程。在已经计算报文中所有字节之后，寄存器的最终值就是 CRC。

当将 CRC 附加到报文上时，首先附加低位字节，然后附加高位字节。附录 B 中包含产生 CRC 的详细实例。

6.5.2 ASCII 传输模式

当使用 ASCII(美国信息交换标准代码)模式设置设备在 Modbus 串行链路上通信时，用两个 ASCII 字符发送报文中的一个 8 位字节。当物理通信链路或者设备能力不能满足 RTU 模式的定时管理要求时，使用该模式。

注：由于每个字节需要两个字符发送，所以这种模式比 RTU 模式效率低。

示例：将字节 0x5B 编码为两个字符，0x35 和 0x42(用 ASCII 表示的 0x35＝"5"，0x42＝"B")。

ASCII 模式中每个字节的格式为(10 位)：

编码系统：　十六进制，ASCII 字符 0～9，A～F

在报文中每个 ASCII 字符中，1 个十六进制字符包含 4 个数据位

每个字节的位：

1 个起始位

7 个数据位，首先发送最低有效位

1 个奇偶校验位

1 个停止位

要求使用偶校验。也可以使用其他模式(奇校验、无校验)。为了保证与其他产品的最大兼容性，建议还支持无校验模式。默认校验模式必须是偶校验。

注：使用无校验时要求 2 个停止位。

串行地传送字符的方法为：

发送每个字符或字节的顺序是从左到右(见图 17)。

最低有效位(LSB)…最高有效位(MSB)

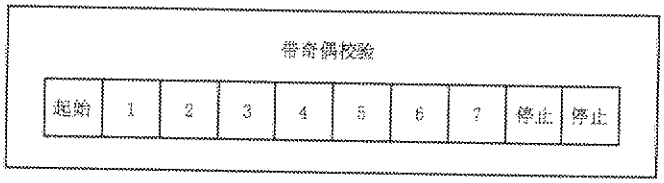


图 17　ASCII 模式中的位序列

通过配置，设备可以接受奇校验、偶校验或无校验。如果无校验，那么传送一个附加的停止位来填充字符帧(见图 18)。

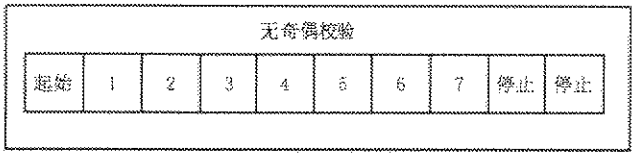


图 18 ASCII 模式中的位序列(无校验的特殊情况)

帧校验字段：纵向冗余校验(LRC)

6.5.2.1 **Modbus 报文 ASCII 帧**

传送设备将 Modbus 报文放置在带有已知起始和结束点的帧中。这就允许接收新帧的设备在报文的起始处开始接收，并且知道报文传输何时结束。必须能够检测到不完整的报文，并且必须作为结果设置错误标志。

报文帧的地址字段包含两个字符。

在 ASCII 模式中，用特定的帧起始和帧结束字符区分一个报文。一个报文必须以一个“冒号”(:)字符(十六进制 ASCII 码为 3A)起始，以“回车-换行”(CRLF)(十六进制 ASCII 码为 0D 和 0A)结束。

注：可以通过特定的 Modbus 应用命令改变 LF 字符(见 GB/T 19582.1—2008)。

对于其他的字段来说，允许传输的字符为十六进制 0～9，A～F(ASCII 编码)。设备不断地监视通信总线上的“:”字符。当收到这个字符之后，每个设备译码后续字符直到检测出帧结束为止。

报文中字符间的时间间隔可以达 1 s。大于 1 s 的时间间隔表示已经出现错误，除非用户已配置了较长时间的超时时间值。某些广域网应用可以要求 4 s～5 s 的超时时间。

图 19 表示了一个典型的报文帧。

起始符	地址	功能码	数据	LRC	结束符
1个字符 :	2个字符	2个字符	0～2x252 个字符	2个字符	2个字符 CR-LF

图 19 ASCII 报文帧

注：每个数据字节需要用两个字符编码。因此，为了在 Modbus 应用级上确保 ASCII 模式和 RTU 模式兼容，ASCII 数据字段最大数据长度(2x252)为 RTU 数据字段最大数据长度(252)的两倍。因此，Modbus ASCII 帧的最大长度为 513 个字符。

在图 20(状态图)中综述了 ASCII 报文组帧的要求。“主站”和“从站”均在同一个图中表示。

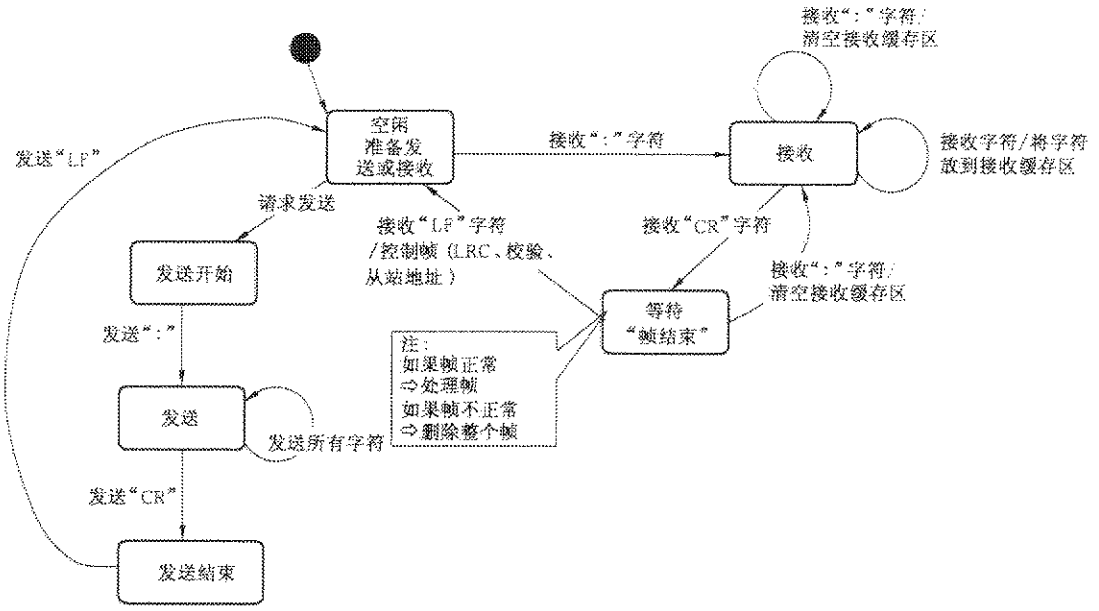


图 20 ASCII 传输模式的状态图

上述状态图的解释：

——“空闲”状态是在没有发送和接收报文活动情况下的正常状态。

——每次接收到“:”字符表示新报文的开始。如果在一个报文的接收过程中收到这个字符，则当前报文被认为不完整并被丢弃。然后，分配一个新接收缓存区。

——检测到帧结束之后，执行 LRC 计算和校验。然后，分析地址字段来确定帧是否发往这个设备。如果不是发往这个设备，那么丢弃这个帧。为了减少接收处理时间，在接收到地址字段，就可以分析地址字段，而不需要等到整个帧结束。

6.5.2.2 LRC 校验

在 ASCII 模式中，报文包含一个差错校验字段，该字段是基于对全部报文内容执行的纵向冗余校验(LRC)计算结果，不包括起始“冒号”和结束 CRLF 对。无论报文中的单个字符使用何种奇偶校验，均应用这种 LRC 校验。

LRC 字段为一个字节，包含一个 8 位二进制值。发送报文的设备计算 LRC 值，将 LRC 值附加到报文中。在接收报文过程中，接收报文的设备重新计算 LRC 值，并将计算值与 LRC 字段中接收到的实际值相比较。如果两个值不相等，则产生错误。

对报文中的所有连续 8 位字节相加，丢弃任何进位，然后求出其二进制补码作为计算得到的 LRC 码。对报文中的每个字节进行这种操作。在对原报文中每个字节进行 ASCII 码编码之前，对每个字节进行这种操作。这种计算不包括报文起始“冒号”和报文结束 CRLF 对字段。

LRC 的结果被编码为两个字节的 ASCII 码，并将其放置在 ASCII 模式报文帧的 CRLF 之前。

附录 B 含有 LRC 生成的详细示例。

6.6 差错校验方法

标准 Modbus 串行链路的安全性是基于两种差错校验：

——应该将奇偶校验(偶或奇)应用于每个字符。

——必须将帧校验(LRC 或 CRC)应用于整个报文。

在发送设备(主站或从站)中生成字符校验和帧校验，并且在发送前将其附加于报文内容中。在接收时，设备(从站或主站)校验每个字符和整个报文帧。

主站由用户配置成在放弃事务处理前等待一个预定的超时间隔(响应超时)。这个间隔被设置成足够的时间长度，以便任何从站正常地响应(单播请求)。如果从站检测到传输错误，则报文将不起作用。从站不会对主站响应。因此，超时时间到，允许主站的程序来处理错误。访问不存在的从站也会产生超时错误。

6.6.1 奇偶校验

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

无论规定了偶校验还是奇校验，都是计算每个字符数据部分中为 1 的位的总数(ASCII 模式有 7 个数据位，RTU 模式有 8 个数据位)。然后，将奇偶位设置为 0 或 1，以便使 1 的个数为偶数或奇数。

例如：RTU 字符帧中包含的 8 个数据位是：

1100 0101

这个帧中为 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)。

当发送报文时，计算奇偶位，并将其附加到每个字符帧。接收的设备计算为 1 的位的数量，并且如果与设备配置不同，则设置错误标记(必须将 Modbus 串行链路上的所有设备配置成使用相同的奇偶校验)。

奇偶校验只能检测到传输过程中一个字符帧中增加或丢失奇数个“1”的错误。例如：如果使用奇校验，含有 3 个为 1 的位的字符帧中丢失了 2 个为 1 的位，而结果仍然为奇数个为 1 的位。

如果没有规定奇偶校验，不会传送奇偶位，也不用进行奇偶校验。传送一个附加的停止位来填充字符帧。

6.6.2 帧校验

根据传输模式(RTU 模式或 ASCII 模式)使用两种帧校验方法。

——在 RTU 模式中,报文包含一个基于循环冗余校验(CRC)方法的差错校验字段。CRC 字段校验整个报文的内容。无论报文中单个字符使用何种奇偶校验,均应用这种 CRC 校验。

——在 ASCII 模式中,报文包含一个基于纵向冗余校验(LRC)方法的差错校验字段。LRC 字段校验报文的内容,不包括报文中的起始"冒号"和结束 CRLF 对。无论报文中单个字符使用何种奇偶校验,均应用这种 LRC 校验。

有关差错检验方法的详细内容见前面的章节。

7 物理层

7.1 引言

新的串行链路上的 Modbus 解决方案应该按照 EIA/TIA-485 标准(也称 RS485 标准)实现电气接口。该标准允许"两线配置"的点对点和多点系统。此外,一些设备可以实现"四线"RS485 接口。

设备还可以实现 RS232 接口。

在这种 Modbus 系统中,一个主站设备和一个或几个从站设备在一个无源串行链路上进行通信。

在标准 Modbus 系统中,在一条由 3 根导线组成的干线电缆上连接所有设备(并联)。其中两条导线("两线"配置)形成一对平衡双绞线,在这个双绞线上双向传送数据,典型的速率为 9 600 bit/s。

每台设备可按如下方法连接(见图 21):

——或直接连接到干线电缆上,形成菊花链;

——或经分支电缆连接到一个无源分支器;

——或经专用电缆连接到一个有源分支器。

在设备上,可以使用接线端子、RJ45 或 9 芯 D-型连接器与电缆连接(见 7.5)。

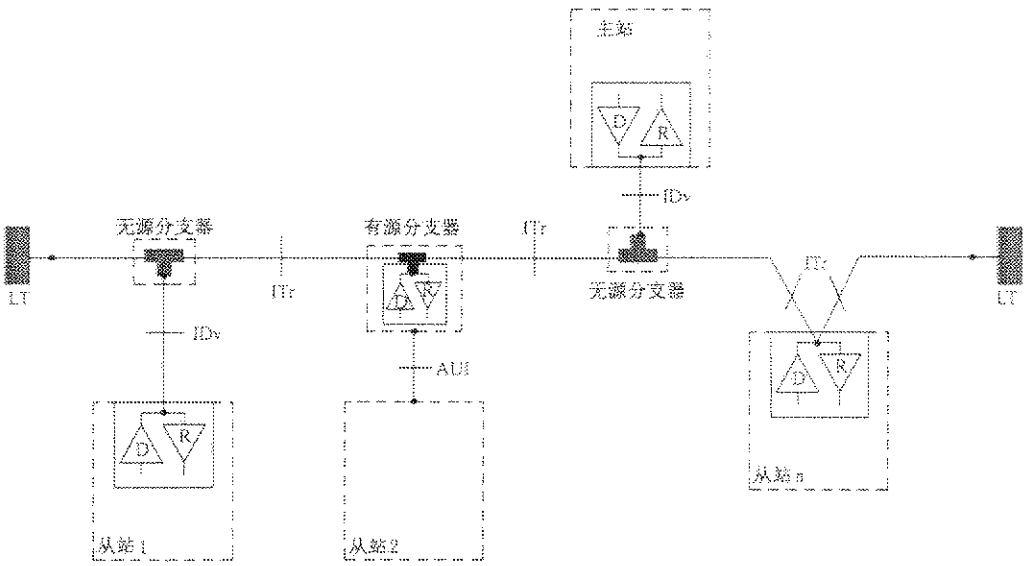


图 21 串行总线结构

7.2 数据信号传输速率

要求实现 9 600 bit/s 和 19.2 kbit/s 传输速率。默认值为 19.2 kbit/s。

也可使用其他波特率:1 200 bit/s、2 400 bit/s、4 800 bit/s……38 400 bit/s、56 kbit/s、115 kbit/s 等。

在发送的情况下,每种实现的波特率精度必须高于 1%;在接收的情况下,必须允许 2%误差。

7.3 电气接口

7.3.1 多点串行总线结构

图 21 表示了 Modbus 多点串行链路系统中串行总线的总体结构。

Modbus 多点串行链路总线是由主电缆(干线)和一些分支电缆组成。

在干线电缆的两端需要使用线路终端电阻以使阻抗匹配(详细情况分别见 7.3.2 和 7.3.3)。

如图 21 所示,在同一个 Modbus 串行链路总线中可以有不同的实现方式:

——将集成通信收发器的设备通过无源分支器和分支电缆连接到干线上(例如:从站 1 和主站);

——将没有集成通信收发器的设备通过有源分支器和分支电缆连接到干线上(有源分支器集成了收发器)(例如:从站 2);

——将设备以菊花链形式直接连接到干线电缆上(例如:从站 n)。

采用下列约定:

——干线间的接口称为 ITr(干线接口);

——设备和无源分支器间的接口称为 IDv(分支接口);

——设备和有源分支器间的接口称为 AUI(附属单元接口)。

注 1:在某些情况下,可以直接将分支器连接到设备的 IDv-插槽或 AUI-插槽上,而不使用分支电缆。

注 2:一个分支器可以有多个 IDv 插槽来连接多台设备。当它是无源分支器时,称这个分支器为分配器。

注 3:当使用有源分支器时,可以通过 AUI 或 ITr 接口提供分支器电源。

在后续章节中,将介绍 ITr 和 IDv 接口(详细情况分别见 7.3.2 和 7.3.3)。

7.3.2 2 线 Modbus 定义

见图 22 和表 2。

Modbus 在串行链路上的解决方案应该依照 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,还必须使用第三条导线将总线上所有设备相互连接:公共端。

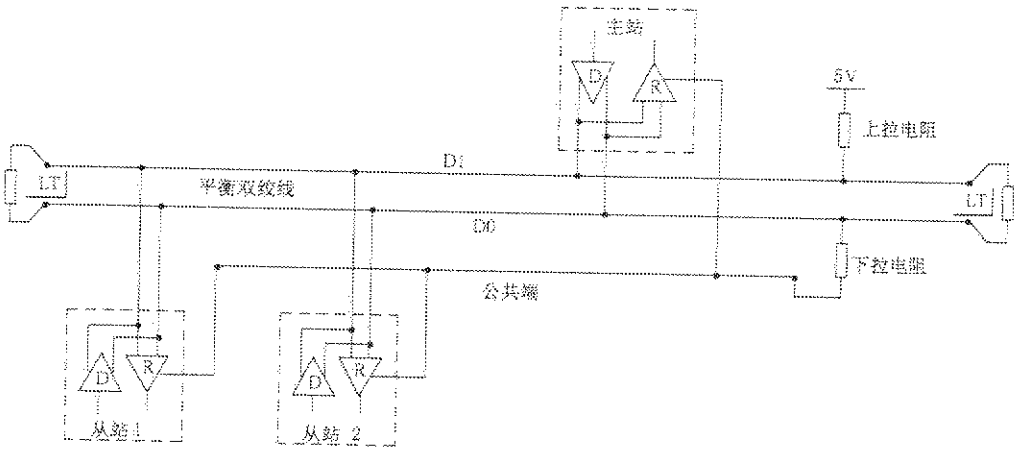


图 22 通用的 2 线拓扑结构

表 2 2 线 Modbus 电路定义

在电路上要求的		设备	在设备上要求的	EIA/TIA-485 的名称	描 述
在 ITr 上	在 IDv 上				
D1	D1	I/O	X	B/B'	收发器端子 1,U_1 电压 ($U_1 > U_0$,对于二进制 1[OFF]状态)
D0	D0	I/O	X	A/A'	收发器端子 0,U_0 电压 ($U_0 > U_1$,对于二进制 0[ON]状态)
公共端	公共端	—	X	C/C'	信号和可选电源的公共端

注 1:对于线路终端(LT)电阻、上拉和下拉电阻,见 7.4。

注 2:在编写与设备和分支器有关的文件(用户指南,接线指南,……)时,必须使用 D0、D1 和公共端的电路名称,以

便易于互操作。

注 3：可以增加可选的电气接口，例如：

a) 电源：5 V~24 V D.C.。

b) 端口模式控制：PMC 电路(TTL 兼容)。需要时，可由这个外电路和/或其他方式(例如：设备上的开关)来控制端口模式。在第一种情况下，一个开路 PMC 将要求 2 线 Modbus 模式，但实现过程中，根据实现的不同，PMC 也可以将接口设置成 4 线 Modbus 或 RS232 Modbus 模式。

7.3.3 可选的 4 线 Modbus 定义

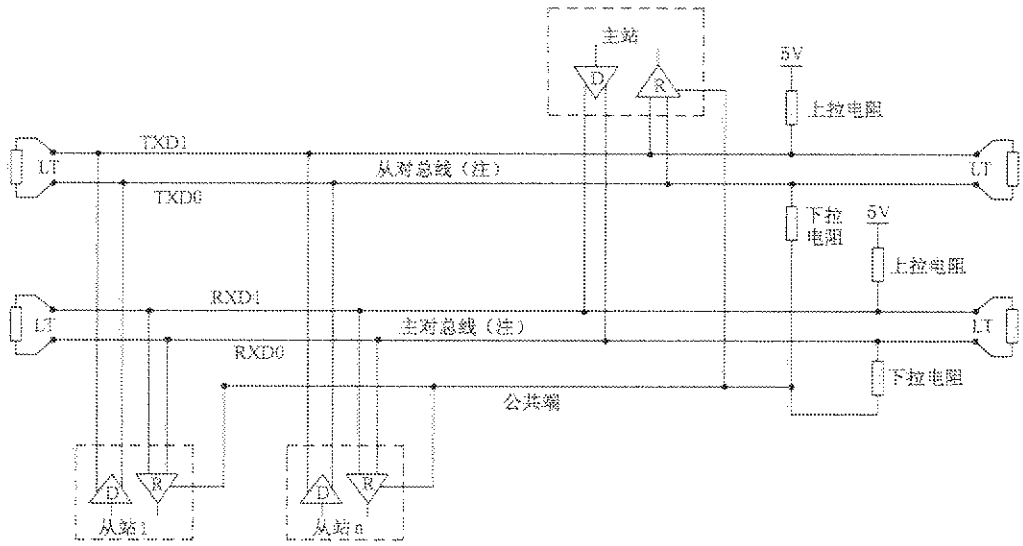
见图 23 和表 3。

一个 Modbus 设备同样可选择实现 2 对总线(4 线)单向数据传输。从站只能接收主对总线(RXD1-RXD0)上的数据，而主站只能接收从对总线(TXD1-TXD0)上的数据。

实际上，必须使用第 5 条导线作为公共端将 4 线总线上的所有设备相互连接。

像 2 线 Modbus 那样，在任何时刻只有一个驱动器有权发送数据。

这种设备必须依照 EIA/TIA-485 在每个平衡线对上实现驱动器和接收器。(有时候这种解决方式被称为“RS422”，这是错误的：RS422 标准不支持一个平衡线对上的多个驱动器。)



注：从对总线是用于从站发送信号的总线；主对总线是用于主站发送信号的总线。

图 23 通用的 4 线拓扑结构

表 3 可选的 4 线 Modbus 电路定义

在电路上要求的		设备	在设备上要求的	EIA/TIA-485 的名称	IDv 的描述
在 ITr 上	在 IDv 上				
TXD1	TXD1	输出	X	B	发生器端子 1，U_b 电压 ($U_b > U_a$，对于二进制 1[OFF]状态)
TXD0	TXD0	输出	X	A	发生器端子 0，U_a 电压 ($U_a > U_b$，对于二进制 0[ON]状态)
RXD1	RXD1	输入	(1)	B′	接收器端子 1，U'_b 电压 ($U'_b > U'_a$，对于二进制 1[OFF]状态)
RXD0	RXD0	输入	(1)	A′	接收器端子 0，U'_a 电压 ($U'_a > U'_b$，对于二进制 0[ON]状态)
公共端	公共端	—	X	C/C′	信号和可选电源的公共端

注 1：对于线路终端(LT)电阻、上拉和下拉电阻，见 7.4。

注 2：只有在实现 4 线 Modbus 选项时，表 3 中(1)所示的那些电路才是要求的。

注 3：当使用 5 根线时，在编写与设备和分支器有关的文件(用户指南，接线指南，……)时，必须使用上述电路的名称，以便易于互操作。

注 4：可以增加可选的电气接口，例如：

a) 电源：5 V～24 V D.C.。

b) PMC 电路：见 7.3.2 中关于此可选电路的注(在 2 线 Modbus 电路定义中)。

7.3.3.1 4 线接线系统中的要点

见表 4。

在这种 4 线 Modbus 中，主站设备和从站设备均有上述 5 个 IDv 接口。

作为主站必须：

——接收在从对总线(TXD1-TXD0)上从站发送的数据；

——在主对总线(从站接收的 RXD1-RXD0)上发送数据。

4 线接线系统必须交叉主站的 ITr 和 IDv 之间的两对总线。

表 4 主站的 ITr 和 IDv 之间的两对总线

	主站 IDv 上的信号		EIA/TIA-485 名称	ITr 上的电路
	名称	类型		
从对总线	RXD1	输入	B′	TXD1
	RXD0	输入	A′	TXD0
主对总线	TXD1	输出	B	RXD1
	TXD0	输出	A	RXD0
	公共端	——	C/C′	公共端

通过交叉电缆实现这种交叉，但是在 2 线系统中这种交叉电缆的连接可能会造成损坏。为了连接 4 线主站设备(带有 Modbus 连接器)，较好的解决方法是使用含有交叉功能的分支器。

7.3.3.2 4 线与 2 线接线的兼容性

为了将带有 2 线物理接口的设备接入一个现有 4 线系统中，可以对 4 线接线系统进行下列改动：

——应将 TXD0 信号与 RXD0 信号连接，使之成为 D0 信号；

——应将 TXD1 信号与 TXD0 信号连接，使之成为 D1 信号；

——应重新设计上拉、下拉电阻和线路终端电阻以正确地适应 D0、D1 信号的要求。

图 24 给出一个使用 2 线接口的从站 2 和 3 能与使用 4 线接口的主站和从站 1 一起工作的示例。

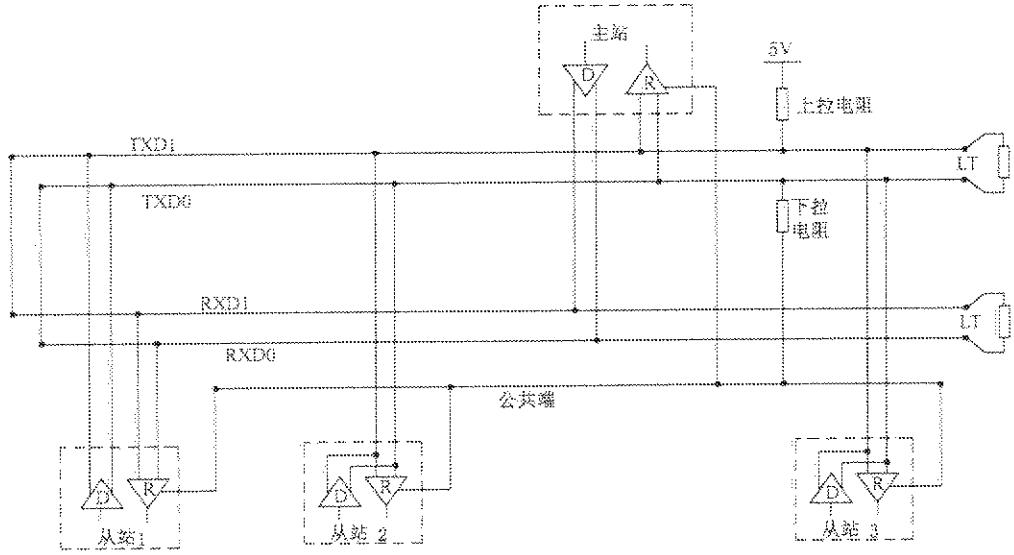


图 24 将 4 线接线系统转换为 2 线接线系统

为了将带有 4 线物理接口的设备接入一个现有 2 线系统中，可以按下述要求安排该新接入设备的 4 线接口，在每个 4 线设备接口上：

——应将 TXD0 信号与 RXD0 信号连接，然后将其连接到干线的 D0 信号上；

——应将 TXD1 信号与 RXD1 信号连接，然后将其连接到干线的 D1 信号上。

图 25 给出一个使用 4 线接口的从站 2 和 3 与使用 2 线接口的主站和从站 1 一起工作的示例。

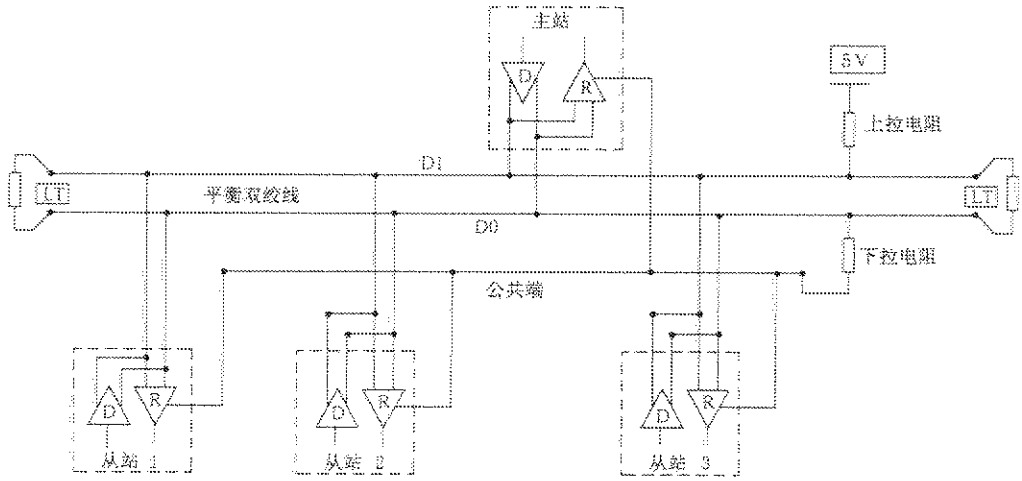


图 25　将带 4 线接口的设备连接到 2 线接线系统

7.3.4　RS232-Modbus 定义

某些设备可以在 DCE 和 DTE 之间实现 RS232 接口(见表 5)。

表 5　可选的 RS232-Modbus 电路定义

信号	DCE	在 DCE 端要求的	在 DTE 端要求的	描　述
公共端	—	X	X	信号公共端
CTS	输入			清除发送
DCD	—			检测到数据载波(从 DCE 到 DTE)
DSR	输入			数据设置就绪
DTR	输出			数据终端就绪
RTS	输出			请求发送
RXD	输入	X	X	接收的数据
TXD	输出	X	X	发送的数据

注 1：只在实现 RS232-Modbus 选项时必须使用标有“X”的信号。

注 2：信号符合 EIA/TIA-232 标准。

注 3：每个 TXD 必须与其他设备的 RXD 连接。

注 4：RTS 可以与其他设备的 CTS 连接。

注 5：DTR 可以与其他设备的 DSR 连接。

注 6：可以增加可选的电气接口，例如：

a)　电源：5～24 V DC。

b)　PMC 电路：见 7.3.2 中关于此可选电路的注(在 2 线 Modbus 电路定义中)。

7.3.5　RS232-Modbus 要求

这种可选串行链路上的 Modbus 应该只应用于短距离(一般小于 20 m)的点对点的互连。同时，必

须符合 EIA/TIA-232 标准：

——电路定义；

——最大线路对地电容(2 500 pF，对于 100 pF/m 的电缆，长度为 25 m)。

关于屏蔽"电缆"和使用 5 类电缆的可能性，见 7.6。

设备提供的文件必须指出：

——该设备是否必须作为 DCE 或 DTE；

——若是上述情况，必须说明可选的电路如何工作。

7.4 多点系统要求

对于任何 EIA/TIA-485 多点系统，无论是 2 线配置还是 4 线配置，均适用下列要求。

7.4.1 无中继器情况下，最大设备数量

在没有中继器的 RS485-Modbus 系统中，最多允许有 32 个设备。

与下列项目有关：

——所有可能的地址；

——设备使用的 RS485 单元负载总量；

——以及需要的线路极性偏置。

一个 RS485 系统可以容纳许多设备。有些设备在没有中继器情况下允许在设备数大于 32 个的 RS485-Modbus 串行链路上运行。

在这种情况下，必须在这个 Modbus 设备文件中说明没有中继器时能允许接多少个这类设备。

也可以在两个重负载的 RS485-Modbus 之间使用中继器。

7.4.2 拓扑结构

没有配置中继器的 RS-485-Modbus 有一个与所有设备直接连接(菊花链)或通过短分支电缆连接的干线电缆。

干线电缆，又称总线，可能很长。它的两端必须接线路终端。

也可以在多个 RS-485 Modbus 之间使用中继器。

7.4.3 长度

必须限制干线电缆的端到端长度。最大长度与波特率、电缆(规格、电容或特性阻抗)、菊花链上的负载数量以及网络配置(2 线或 4 线制)有关。

对于最高波特率为 9 600 bit/s、AWG26(或更粗)规格的电缆来说，其最大长度为 1 000 m。在图 24(4 线制接线用作 2 线制接线的系统中)中所示的情况下，必须将最大长度除以 2。

分支必须短，不能超过 20 m。如果使用 n 个分支的多端口分支器，每个分支最大长度必须限制为 40 m 除以 n。

7.4.4 接地形式

必须将"公共端"电路(信号与可选电源的公共端)直接连接到保护地上，最好是整条总线单点接地。通常，该点可选在主站上或其分支器上。

7.4.5 线路终端

沿线路传播的信号遇到阻抗不连续，会在传输线路中产生反射。为了使从 RS-485 电缆端的反射最小，要求在总线接近两端处放置线路终端。

由于传播是双向的，故在线路两端配置终端是非常重要的，但是，在一个无源 D0-D1 平衡线对上放置的线路终端不允许超过 2 个。也不允许在分支电缆上放置任何线路终端。

每个线路终端必须连接在平衡线 D0 和 D1 的两条导线之间。

线路终端可以是 150 Ω(0.5 W)的电阻。

当双绞线必须进行极性偏置时，较好的选择是使用电容(1 nF，最低 10 V)与 120 Ω(0.25 W)电阻串联。

在 4 线系统中，在总线的两端，每对线都必须有终端。

在 RS232 系统中，不应该连接线路终端。

7.4.6 线路极性偏置

当在 RS485 平衡线对上没有数据传输时，这个线路没有任何驱动，因此容易受到外部噪声或干扰的影响。为确保其接收器处于一个稳定状态，在没有数据信号出现时，一些设备需要使总线偏置。

每个 Modbus 设备都必须用文件说明：

——该设备是否需要线路极性偏置；

——该设备是否已经实现或可以实现这样的线路极性偏置。

如果一个或多个设备需要线路极性偏置，则必须在该 RS485 平衡线对上连接一对电阻：

——D1 线上的上拉电阻连接至 5V 电压；

——D0 线上的下拉电阻连接至公共端。

这些电阻的阻值必须在 450～650 Ω 之间。在串行链路总线上，650 Ω 的电阻值可以允许接入较多设备。

在这种情况下，必须在整个串行总线的一个地方实现双绞线的极性偏置。通常，将该点选在主站或其分支器上。其他设备不能实现任何极性偏置。

在这类 Modbus 串行链路上允许的最多设备数比无极性偏置的 Modbus 系统少 4 个。

7.5 机械接口

在 IDv 与 IDz 两种连接中可以使用接线端子。必须向用户提供有关每个信号确切接线位置及相关信息，这些信号名称要与 7.3“电气接口”所述一致。

如果一台设备此使用一个 RJ45(或小型 DIN 或 D-型)连接器作为 Modbus 机械接口，则必须选用屏蔽的孔连接器。因而，电缆终端必须带有屏蔽的针连接器。

7.5.1 2 线 Modbus 连接器的引脚

见图 26 和图 27。

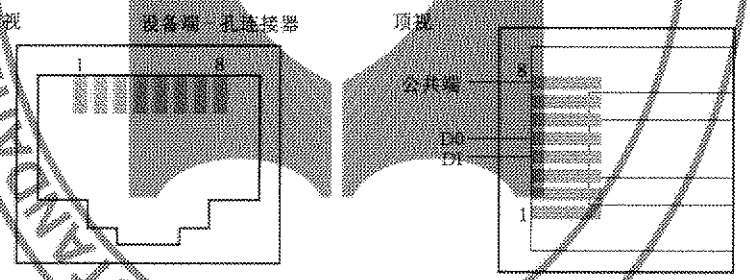


图 26 2 线 Modbus 中使用的 RJ45 连接器(要求的插脚引线)

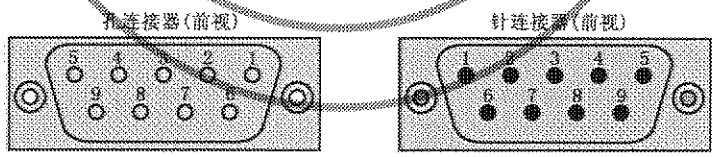


图 27 9 针 D-型连接器

还可以使用螺钉型的连接器。

如果一台标准 Modbus 设备使用 RJ45 或 9 针 D-型连接器，对每种实现电路必须按照表 6 分配引脚。

7.5.2 可选的 4 线 Modbus 连接器引脚

见图 28 和图 29。

还可以使用螺钉型的连接器。

如果一台 4 线 Modbus 设备使用 RJ45 或 9 针 D-型连接器，对每种实现电路必须按照表 7 分配引脚。

表 6　2 线 Modbus RJ45 和 9 针 D-型连接器引脚分配

RJ45 引脚	D9-型连接器引脚	要求的等级	IDv 电路	ITr 电路	EIA/TIA-485 名称	IDv 的描述
3	3	可选的	PMC	—	—	端口模式控制
4	5	要求的	D1	D1	B/B′	收发器端子 1，U_1 电压(U_1)>U_0，对于二进制的 1 [OFF]状态)
5	9	要求的	D0	D0	A/A′	收发器端子 0，U_0 电压(U_0 > U_1，对于二进制的 0 [ON]状态)
7	2	建议的	VP	—	—	正的 5…24 V DC 电源
8	1	要求的	公共端	公共端	C/C′	信号和电源的公共端

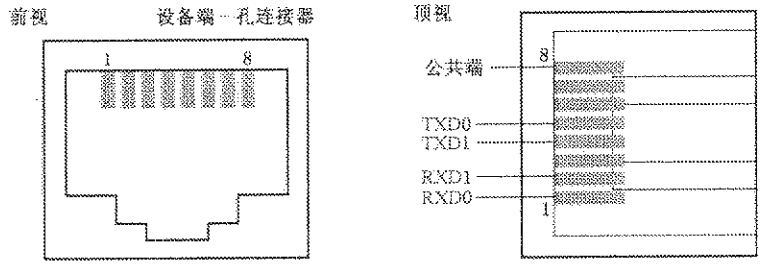


图 28　4 线 Modbus 上的 RJ45 连接器(要求的插脚引线)

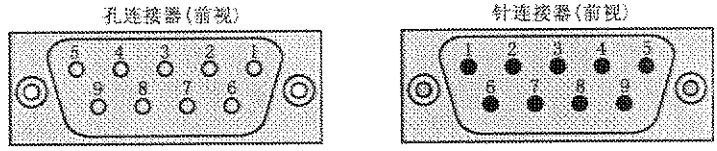


图 29　9 针 D-型连接器

表 7　可选的 4 线 Modbus RJ45 和 9 针 D-型连接器引脚分配

RJ45 引脚	D9-型连接器引脚	要求的等级	IDv 信号	ITr 信号	EIA/TIA-485 名称	IDv 的描述
1	8	要求的	RXD0	RXD0	A′	接收器端子 0，U'_0 电压(U'_0>U'_1，对于二进制的 0 [ON]状态)
2	4	要求的	RXD1	RXD1	B′	接收器端子 1，U'_1 电压(U'_1>U'_0，对于二进制的 1 [OFF]状态)
3	3	可选的	PMC	—	—	端口模式控制
4	5	要求的	TXD1	TXD1	B	发送器端子 1，U_1 电压(U_1>U_0，对于二进制的 1 [OFF]状态)
5	9	要求的	TXD0	TXD0	A	发送器端子 0，U_0 电压(U_0>U_1，对于二进制的 0 [ON]状态)
7	2	建议的	VP	—	—	正的 5…24 V DC 电源
8	1	要求的	公共端	公共端	C/C′	信号和电源的公共端

注：当在同一个接口上既有 2 线又有 4 线的配置时，必须使用 4 线接线标记。

7.5.3 可选 RS232-Modbus 的 RJ45 和 9 针 D-型连接器引脚

如果一个 RS232-Modbus 设备使用 RJ45 或 9 针 D-型连接器，对每种实现电路必须按照表 8 分配引脚。

表 8 可选 RS232-Modbus 的 RJ45 和 9 针 D-型连接器引脚分配

DCE 带下划线的引脚为输出			电路			DTE 带下划线的引脚为输出		
RJ45 引脚	D9-型连接器引脚	要求的等级	名称	描述	RS232 源	要求的等级	RJ45 引脚	D9-型连接器引脚
1	2	要求的	TXD	发送的数据	DTE	要求的	2	3
2	3	要求的	RXD	接收的数据	DCE	要求的	1	2
3	7	可选的	CTS	清除发送	DCE	可选的	6	8
6	8	可选的	RTS	请求发送	DTE	可选的	3	7
8	5	要求的	公共端	信号公共端	—	要求的	8	5

注：某些 DCE 已经与 DTE 的同名端接线作了交叉处理，此时的连接电缆就不允许使用交叉电缆，而必须使用直通电缆(有些情况下同名端是直连的，不用交叉，具体连接应见产品说明书)。

7.6 电缆

串行链路上的 Modbus 电缆必须是屏蔽的。在每条电缆一端，其屏蔽必须连接到保护地上。若在这端使用了连接器，则将连接器外壳连接到电缆屏蔽层上。

RS485-Modbus 必须使用一对平衡线对(用于 D0-D1)和第三根线(用于公共端)。此外，在 4 线 Modbus 系统中必须使用第二对平衡线对(用于 RXD0 RXD1)。

若使用 4 对线的 5 类电缆来连接，应在编写用户指南时提醒用户：

"在 2 线 Modbus 系统中，交叉电缆的连接可能造成破坏"。

为减少电缆连接中的错误，在 RS485-Modbus 电缆中，建议接线采用色彩标记，见图 30

	信号名称	建议的颜色
	D1-TXD1	黄
	D0-TXD0	橙
	公共端	灰
4 线(可选的)	RXD0	白
4 线(可选的)	RXD1	蓝

图 30 RS485-Modbus 连线的色彩标记

注：5 类电缆使用其他颜色。

对 RS485-Modbus 来说，必须选择足够宽的线缆直径以便允许使用最大长度(1 000 m)。AWG24 能够满足 Modbus 数据传输的需要。

RS485-Modbus 使用 5 类电缆，最大长度可达 600 m。

对 RS485-系统中使用的平衡线对，特别是对 19 200 bit/s 和更高波特率，可以首选高于 100 Ω 的特性阻抗。

7.7 可视诊断

对于可视诊断来说，必须用 LED(发光二极管)指示通信状态和设备状态，见表 9。

表 9 LED 通信状态和设备状态

LED	要求的等级	描 述	建议的颜色
通信	要求的	在帧接收或发送期间置于 ON (两个 LED 分别表示帧接收和帧发送，或全部用一个 LED 表示)	黄
故障	建议的	置 ON：内部故障 闪烁：其他故障(通信故障或配置故障)	红
设备状态	可选的	置 ON：设备通电	绿

8 安装和文档

8.1 安装

产品供应商应该注意向用户提供 Modbus 系统或 Modbus 设备的全部有用信息，以便他们避免电缆连接错误或错误使用电缆连接附件：

——一些其他的现场总线(例如：CANOpen)使用相同的连接器类型(D-型，RJ45…)。

——正在研究的带有电源的以太网平衡线对。

——其他产品使用相同连接器类型(D-型，RJ45…)用于 I/O 电路。

对于绝大部分的连接器，没有简单的验证方法(偏置程度或其他实现)。

8.2 用户指南

任何 Modbus 设备或电缆连接系统组件的用户指南必须含有但不限于下列一种或两种类型信息：

8.2.1 所有 Modbus 产品

在文档中应该具有下列信息：

——所有的实现要求。

——操作模式。

——可视诊断。

——可访问的寄存器和支持的功能码。

——安装规则。

——在文档中应该具有下列章节中要求的信息：

1)“2 线 Modbus 定义”(涉及要求的电路)；

2)“可选的 4 线 Modbus 定义”(涉及要求的电路)；

3)“线路极性偏置”(涉及可能的需求或实现)；

4)“电缆”(特别注意交叉电缆)。

——用重要警告的方式书写有关设备地址的说明：

“在设定设备地址的过程中，保证两个设备不用相同地址是非常重要的。在两个设备地址相同的情况下，整个串行总线工作将不正常，主站将不能与当前总线上所有从站正常通信。”

——特别建议编写“简易入门”一章，作为简易入门，同时给出一个典型的应用示例。

8.2.2 带有可实现选项的 Modbus 产品

必须清晰详尽地描述不同的可选参数：

——可选的串行传输模式；

——可选的奇偶校验；

——可选的波特率；

——可选的电路：电源，端口配置；

——可选的接口；

——如果支持大于 32 个节点，要说明最大允许的设备数量(无中继器)。

9 实现等级

见表 10。

Modbus 串行链路上的每个设备必须遵守相同实现等级的所有强制要求。

使用下列参数对 Modbus 串行链路设备进行分类：

——寻址；

——广播；

——传输模式；

——波特率；

——字符格式；

——电气接口参数。

推荐的两种实现等级：基本和常规等级。

常规等级必须提供可配置的功能。

表 10 实现等级

	基本等级		常规等级	默认值
寻址	从站： 1～247 的可配置地址	主站： 能够从地址 1～247 寻址从站	与基本等级相同	—
广播	是		是	—
波特率	9 600 bit/s(也推荐 19 200 bit/s)		9 600 bit/s,19 200 bit/s +附加的可配置波特率	19 200 bit/s(如果是可实现的,否则 9 600 bit/s)
奇偶校验	偶校验		偶校验+可配置为无校验和奇校验	偶校验
模式	RTU		RTU+ASCII	RTU
电气接口	RS485 2W-电缆接线或 RS232		RS485 2W-电缆接线(4W-电缆接线作为附加选项)或 RS232	RS485 2W-电缆接线
连接器类型	RJ 45(推荐)			—

附 录 A
（资料性附录）
串行链路诊断计数器的管理

A.1 一般描述

Modbus 串行链路定义了一个诊断计数器列表，进行性能和出错管理。

这些计数值可以通过 Modbus 应用协议及其诊断功能访问（功能码 08）。

可以通过一个带有计数器编号的子功能码得到每个计数值。可以利用子功能码 0x0A 清除所有计数器。

在 Modbus 应用协议规范中描述诊断功能的格式。

表 A.1 是串行链路设备支持的诊断和相应子功能码的列表。

表 A.1 串行链路设备支持的诊断和相应子功能码

子功能码	计数器编号	计数器名称	注释（配合后续图表）
十六进制	十进制		
0x0B	1	返回总线报文计数	在上一次重启动、清除计数器操作或加电之后，远程设备在通信系统中检测到的报文数量，不考虑带有出错 CRC 的报文
0x0C	2	返回总线通信出错计数	在上一次重启动、清除计数器操作或加电之后，远程设备遇到的 CRC 出错数量。在检测到字符出错（超限差错、奇偶校验差错）的情况下，或在报文长度<3 个字节的情况下，接收设备不能计算 CRC。在这种情形下，依然增加计数值
0x0D	3	返回从站异常出错计数	在上一次重启动、清除计数器操作或加电之后，远程设备检测到的 Modbus 异常出错数量。它也包含广播报文中检测到的出错，即使这种情况下不返回异常报文 在 GB/T 19582.1—2008 中描述并列出异常差错
0x0E	4	返回从站报文计数	在上一次重启动、清除计数器操作或加电之后，对远程设备寻址的报文数量，包括远程设备处理的广播报文
0x0F	5	返回从站无响应计数	在上一次重启动、清除计数器操作或加电之后，没有返回响应（既没有正常响应也没有异常响应）的远程设备接收的报文数量。也就是说，这个计数器计算已接收到的广播报文数量
0x10	6	返回从站 NAK 计数	在上一次重启动、清除计数器操作或加电之后，远程设备对接收到的报文返回否定确认（NAK）异常响应报文的数量 在 GB/T 19582.1—2008 中描述并列出异常响应
0x11	7	返回从站忙计数	在上一次重启动、清除计数器操作或加电之后，远程设备对接收到的报文返回从站忙异常响应报文的数量 在 GB/T 19582.1—2008 中描述并列出异常响应
0x12	8	返回总线字符超限计数	在上一次重启动、清除计数器操作或加电之后，由于字符超限状况而无法处理的寻址远程设备的报文数量。由于字符抵达端口的速度高于存储字符的速度，或者由于硬件故障而丢失字符，均产生字符超限

A.2 计数器管理流程图

图 A.1～图 A.3 描述了必须将前面每个计数器计数值增加的条件。

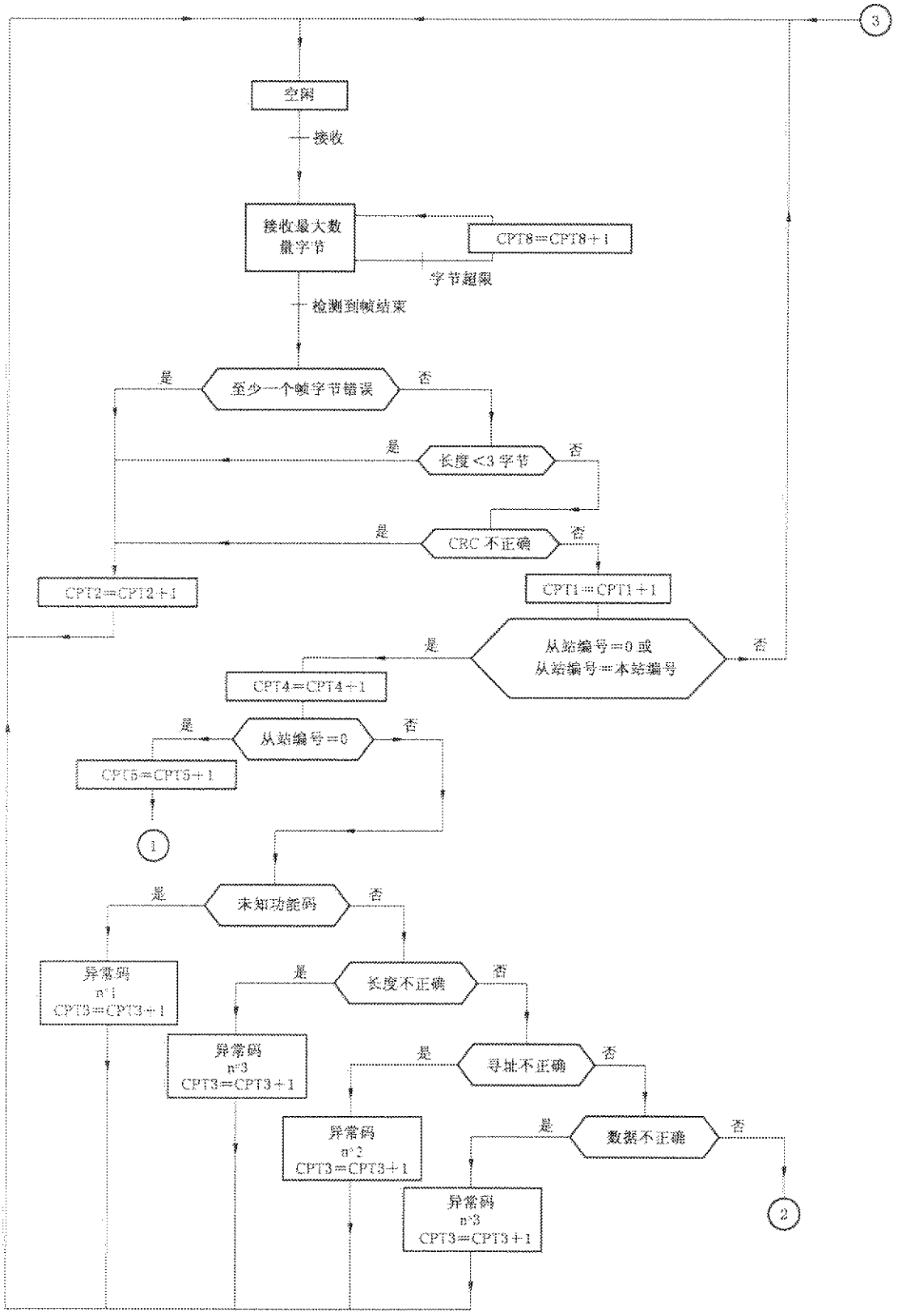


图 A.1 计数器管理流程图-1

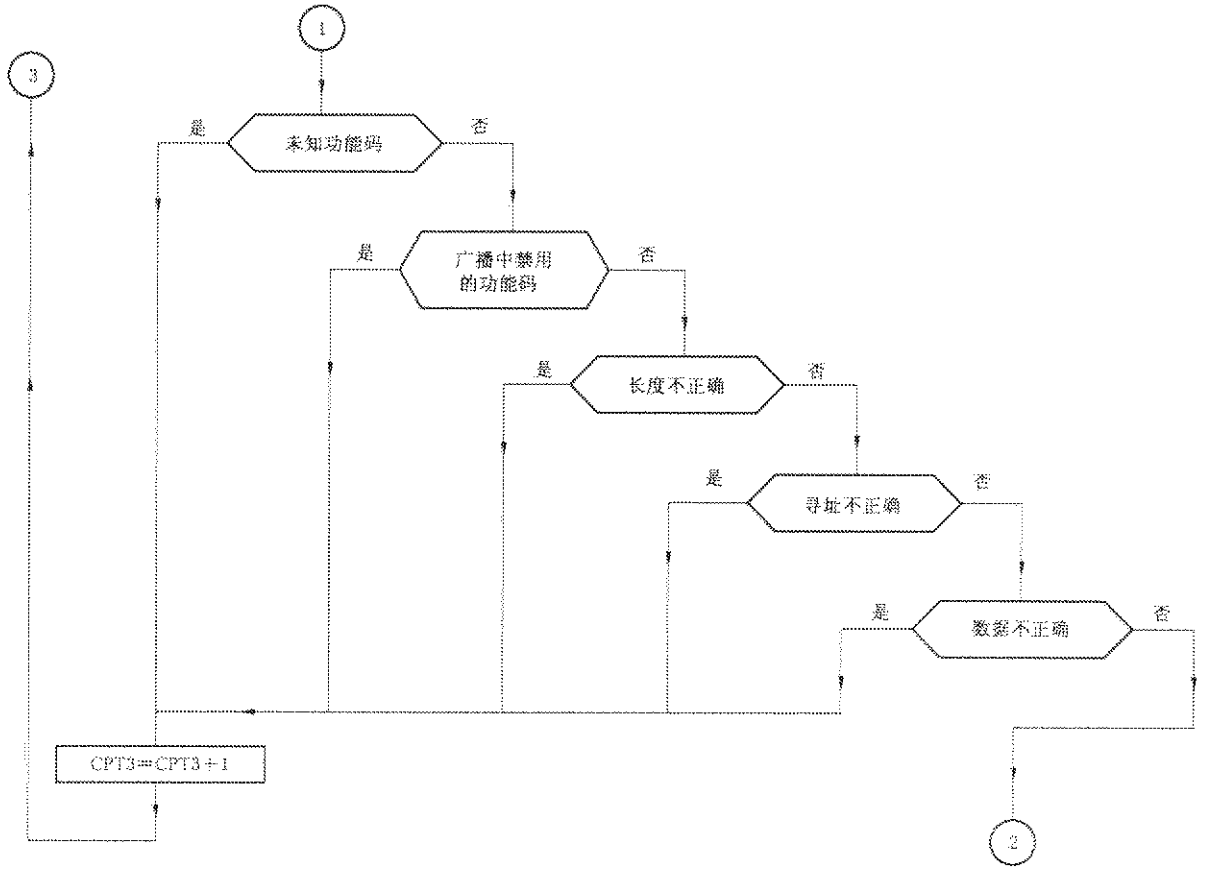


图 A.2 计数器管理流程图-2

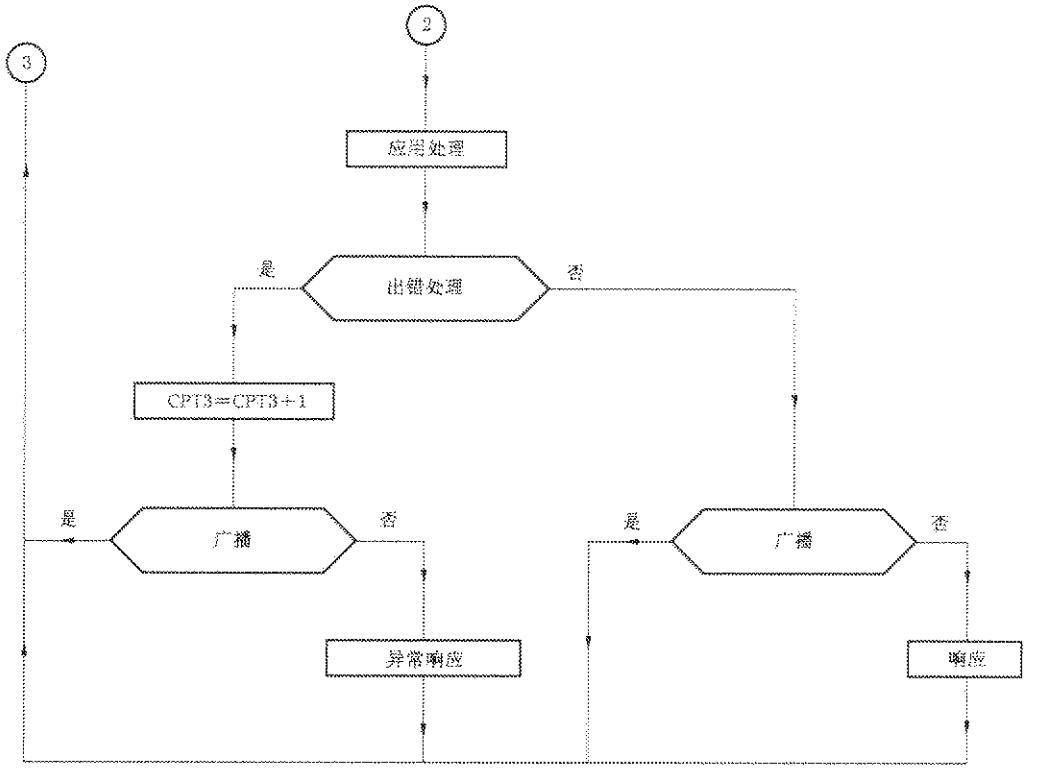


图 A.3 计数器管理流程图-3

附 录 B
（资料性附录）
LRC/CRC 生成

B.1 LRC 生成

纵向冗余校验(LRC)字段是一个字节，包含一个 8 位二进制值。发送设备计算 LRC 值，将 LRC 值附加到报文中。在接收报文过程中，接收设备重新计算 LRC 值，并将计算值与接收到的 LRC 字段中实际值相比较。如果两个值不相等，则说明报文有错误。

计算 LRC，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。LRC 是一个 8 位字段，因此导致结果大于 255 的每个新的相加运算，只是简单地将字段值回零“循环”。因为没有第 9 位，自动放弃进位。

生成一个 LRC 的过程是：

a) 将报文中的所有字节相加，不包括起始“:”和结束 CRLF。将相加结果放到 8 位字段中，以便丢弃进位。

b) 从 FF(全 1)十六进制中减去最终的字段值，产生 1 的补码(二进制反码)。

c) 加 1 产生二进制补码。

B.1.1 将 LRC 放置报文中

当在报文中发送 8 位 LRC(2 个 ASCII 字符)时，首先发送高位字符，然后发送低位字符。例如：如果 LRC 值为十六进制 61(0110 0001)，见图 B.1。

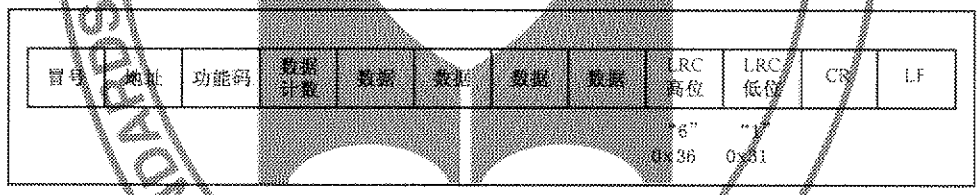


图 B.1 LRC 字符序列

例：下面表示了 C 语言函数进行 LRC 生成的示例。

函数带有两个参数：

unsigned char * auchMsg；含有生成 LRC 所使用的二进制数据的报文缓存区指针，

unsigned short usDataLen；报文缓存区中的字节数。

B.1.2 LRC 生成函数

```
static unsigned char LRC(auchMsg,usDataLen)      /* 函数返回 unsigned char 类型的 LRC */
unsigned char *auchMsg;                          /* 用于计算 LRC 的报文 */
unsigned short usDataLen;                        /* 报文的字节数量 */
{
    unsigned char uchLRC=0;                      /* LRC 字节初始化 */
    while (usDataLen--)                          /* 遍历报文缓存区 */
    uchLRC += *auchMsg++;                        /* 缓存区字节相加，无进位 */
```

```
    return ((unsigned char)(-((char)uchLRC)));  /* 返回二进制补码 */
}
```

B.2 CRC 生成

循环冗余校验(CRC)字段为两个字节,包含一个二进制 16 位值。发送设备计算 CRC 值,将 CRC 值附加到报文中。在接收报文过程中,接收设备重新计算 CRC 值,并将计算值与接收到的 CRC 字段中实际值相比较,如果两个值不相等,则说明报文有错误。

通过对一个 16 位寄存器预装载全"1"来启动 CRC 计算,然后开始将报文中的后续 8 位字节与当前寄存器中的内容进行计算,只有每个字符中的 8 个数据位参与生成 CRC 的计算,起始位、停止位和校验位不参与 CRC 计算。

在生成 CRC 过程中,每个 8 位字符与寄存器中的值异或,然后,向最低有效位(LSB)方向移动这个结果,而用零填充最高有效位(MSB),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)移位之后,下一个 8 位字节与寄存器的当前值异或,然后像上述描述的那样重复 8 次这个过程。在已经计算报文中所有字节之后,寄存器的最终值就是 CRC。

生成一个 CRC 的过程是:

a) 将十六进制 FFFF(全 1)装入一个 16 位寄存器。将这个寄存器称作 CRC 寄存器。

b) 将报文的第一个 8 位字节与 16 位 CRC 寄存器的低字节异或,将结果放置在 CRC 寄存器中。

c) 将 CRC 寄存器右移 1 位(向 LSB 方向),MSB 填充零。提取并检测 LSB。

d) (如果 LSB 为 0):重复步骤 c)(进行另一次移位)。
(如果 LSB 为 1):将 CRC 寄存器与多项式值 0xA001(1010 0000 0000 0001)异或。

e) 重复步骤 c)和 d),直到完成 8 次移位。在完成这个操作之后,即完成了对一个完整的 8 位字节的处理。

f) 对报文的下一个 8 位字节重复步骤 b)~e)。继续进行这种操作,直到处理报文中所有字节为止。

g) CRC 寄存器中的最终内容为 CRC 值。

h) 当将 CRC 值放置到报文中时,必须按如下所述交换高位和低位字节。

B.2.1 将 CRC 放置报文中

当在报文中发送 16 位 CRC(2 个 8 位字节)时,首先发送低位字节,然后发送高位字节。

例如:如果 CRC 值为十六进制 1241(0001 0010 0100 0001),见图 B.2。

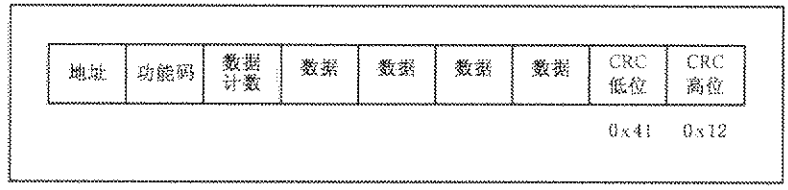


图 B.2 CRC 字节序列

计算 CRC16 的算法见图 B.3。

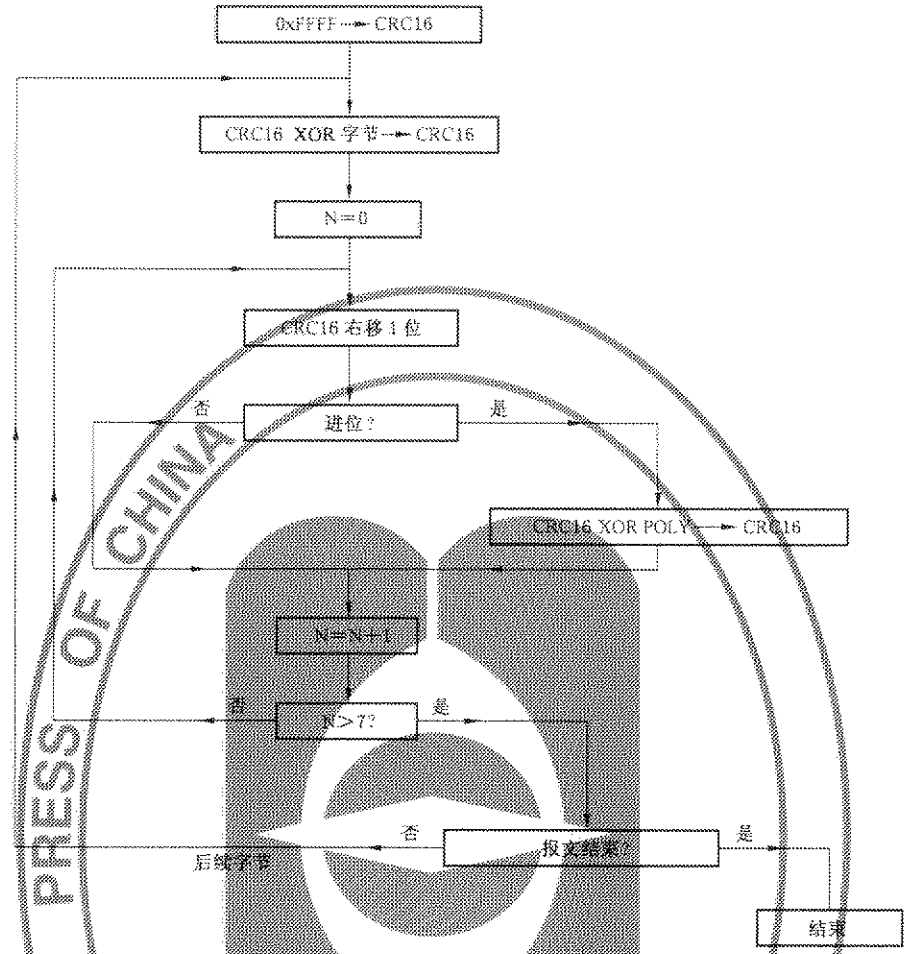


图 B.3 计算 CRC16 的方法

XOR＝异或

N＝信息位的数量

POLY＝计算 CRC16 的多项式＝1010 0000 0000 0001

(生成多项式＝$1+x^{2}+x^{15}+x^{16}$)

在 CRC16 中，发送的第一个字节为最低有效字节。

CRC 计算示例(帧 02 07)

CRC 寄存器初始化		1111	1111	1111	1111
XOR 第一个字符		0000	0000	0000	0010
		1111	1111	1111	1101
	移位 1	0111	1111	1111	1110|1
标志 1，XOR 多项式		1010	0000	0000	0001
		1101	1111	1111	1111
	移位 2	0110	1111	1111	1111|1
标志 1，XOR 多项式		1010	0000	0000	0001
		1100	1111	1111	1110
	移位 3	0110	0111	1111	1111|0
	移位 4	0011	0011	1111	1111|1

	1010	0000	0000	0001
	1001	0011	1111	1110
移位 5	0100	1001	1111	1111|0
移位 6	0010	0100	1111	1111|1
	1010	0000	0000	0001
	1000	0100	1111	1110
移位 7	0100	0010	0111	1111|0
移位 8	0010	0001	0011	1111|1
	1010	0000	0000	0001
XOR 第二个字符	1000	0001	0011	1110
	0000	0000	0000	0111
	1000	0001	0011	1001
移位 1	0100	0000	1001	1100|1
	1010	0000	0000	0001
	1110	0000	1001	1101
移位 2	0111	0000	0100	1110|1
	1010	0000	0000	0001
	1101	0000	0100	1111
移位 3	0110	1000	0010	0111|1
	1010	0000	0000	0001
	1100	1000	0010	0110
移位 4	0110	0100	0001	0011|0
移位 5	0011	0010	0000	1001|1
	1010	0000	0000	0001
	1001	0010	0000	1000
移位 6	0100	1001	0000	0100|0
移位 7	0010	0100	1000	0010|0
移位 8	0001	0010	0100	0001|0
			最高有效位	最低有效位

则帧的 CRC16 为：4112

示例：

下面是一个用 C 语言函数进行 CRC 生成的示例。将所有的可能 CRC 值都预先装入两个数组中，伴随着函数对报文缓存区的处理来简单地索引这些数组。一个数组含有 16 位 CRC 字段高位字节的所有可能的 256 个 CRC 值，另一个数组含有低位字节的所有可能的 CRC 值。

这种索引 CRC 的方式比对报文缓存区的每个新字符都计算新的 CRC 值的方法更快捷。

注：此函数内部执行高/低 CRC 字节的交换。此函数返回的 CRC 值是已经交换了的 CRC 值。

因此，从该函数返回的 CRC 值可以直接地放置于报文中，用于发送。

函数带有两个参数：

unsigned char * puchMsg：含有生成 CRC 所使用的二进制数据的报文缓存区指针。

unsigned short usDataLen;报文缓存区中的字节数。

B.2.2 CRC 生成函数

```
unsigned short CRC16 (puchMsg,usDataLen)   /* 函数以 unsigned short 类型返回 CRC   */
unsigned char *puchMsg;                    /* 用于计算 CRC 的报文                  */
unsigned short usDataLen;                  /* 报文中的字节数量                     */
{

    unsigned char uchCRCHi=0xFF;           /* CRC 的高字节初始化                   */
    unsigned char uchCRCLo=0xFF;           /* CRC 的低字节初始化                   */
    unsigned uIndex;                       /* CRC 查询表索引                       */
    while (usDataLen--)                    /* 遍历报文缓存区                       */
{
    uIndex=uchCRCLo ^ *puchMsg++;          /* 计算 CRC                             */
    uchCRCLo=uchCRCHi ^ auchCRCHi[uIndex];
    uchCRCHi=auchCRCLo[uIndex];
}
    return (uchCRCHi << 8 | uchCRCLo);
}
```

B.2.3 高位字节表

```
/* 高位字节的 CRC 值表 */
static unsigned char auchCRCHi[]={
0x00,0xC1,0x81,0x40,0x01,0xC0,0x80,0x41,0x01,0xC0,0x80,0x41,0x00,0xC1,0x81,
0x40,0x01,0xC0,0x80,0x41,0x00,0xC1,0x81,0x40,0x00,0xC1,0x81,0x40,0x01,0xC0,
0x80,0x41,0x01,0xC0,0x80,0x41,0x00,0xC1,0x81,0x40,0x00,0xC1,0x81,0x40,0x01,
0xC0,0x80,0x41,0x00,0xC1,0x81,0x40,0x01,0xC0,0x80,0x41,0x01,0xC0,0x80,0x41,
0x00,0xC1,0x81,0x40,0x01,0xC0,0x80,0x41,0x00,0xC1,0x81,0x40,0x00,0xC1,0x81,
0x40,0x01,0xC0,0x80,0x41,0x00,0xC1,0x81,0x40,0x01,0xC0,0x80,0x41,0x01,0xC0,
0x80,0x41,0x00,0xC1,0x81,0x40,0x00,0xC1,0x81,0x40,0x01,0xC0,0x80,0x41,0x01,
0xC0,0x80,0x41,0x00,0xC1,0x81,0x40,0x01,0xC0,0x80,0x41,0x00,0xC1,0x81,0x40,
0x00,0xC1,0x81,0x40,0x01,0xC0,0x80,0x41,0x01,0xC0,0x80,0x41,0x00,0xC1,0x81,
0x40,0x00,0xC1,0x81,0x40,0x01,0xC0,0x80,0x41,0x00,0xC1,0x81,0x40,0x01,0xC0,
0x80,0x41,0x01,0xC0,0x80,0x41,0x00,0xC1,0x81,0x40,0x00,0xC1,0x81,0x40,0x01,
0xC0,0x80,0x41,0x01,0xC0,0x80,0x41,0x00,0xC1,0x81,0x40,0x01,0xC0,0x80,0x41,
0x00,0xC1,0x81,0x40,0x00,0xC1,0x81,0x40,0x01,0xC0,0x80,0x41,0x00,0xC1,0x81,
0x40,0x01,0xC0,0x80,0x41,0x01,0xC0,0x80,0x41,0x00,0xC1,0x81,0x40,0x01,0xC0,
0x80,0x41,0x00,0xC1,0x81,0x40,0x00,0xC1,0x81,0x40,0x01,0xC0,0x80,0x41,0x01,
0xC0,0x80,0x41,0x00,0xC1,0x81,0x40,0x00,0xC1,0x81,0x40,0x01,0xC0,0x80,0x41,
0x00,0xC1,0x81,0x40,0x01,0xC0,0x80,0x41,0x01,0xC0,0x80,0x41,0x00,0xC1,0x81,
0x40
};
```

B.2.4 低位字节表

```
/* 低位字节的 CRC 值表 */
static unsigned char auchCRCLo[]={
```

```
    0x00,0xC0,0xC1,0x01,0xC3,0x03,0x02,0xC2,0xC6,0x06,0x07,0xC7,0x05,0xC5,0xC4,
    0x04,0xCC,0x0C,0x0D,0xCD,0x0F,0xCF,0xCE,0x0E,0x0A,0xCA,0xCB,0x0B,0xC9,0x09,
    0x08,0xC8,0xD8,0x18,0x19,0xD9,0x1B,0xDB,0xDA,0x1A,0x1E,0xDE,0xDF,0x1F,0xDD,
    0x1D,0x1C,0xDC,0x14,0xD4,0xD5,0x15,0xD7,0x17,0x16,0xD6,0xD2,0x12,0x13,0xD3,
    0x11,0xD1,0xD0,0x10,0xF0,0x30,0x31,0xF1,0x33,0xF3,0xF2,0x32,0x36,0xF6,0xF7,
    0x37,0xF5,0x35,0x34,0xF4,0x3C,0xFC,0xFD,0x3D,0xFF,0x3F,0x3E,0xFE,0xFA,0x3A,
    0x3B,0xFB,0x39,0xF9,0xF8,0x38,0x28,0xE8,0xE9,0x29,0xEB,0x2B,0x2A,0xEA,0xEE,
    0x2E,0x2F,0xEF,0x2D,0xED,0xEC,0x2C,0xE4,0x24,0x25,0xE5,0x27,0xE7,0xE6,0x26,
    0x22,0xE2,0xE3,0x23,0xE1,0x21,0x20,0xE0,0xA0,0x60,0x61,0xA1,0x63,0xA3,0xA2,
    0x62,0x66,0xA6,0xA7,0x67,0xA5,0x65,0x64,0xA4,0x6C,0xAC,0xAD,0x6D,0xAF,0x6F,
    0x6E,0xAE,0xAA,0x6A,0x6B,0xAB,0x69,0xA9,0xA8,0x68,0x78,0xB8,0xB9,0x79,0xBB,
    0x7B,0x7A,0xBA,0xBE,0x7E,0x7F,0xBF,0x7D,0xBD,0xBC,0x7C,0xB4,0x74,0x75,0xB5,
    0x77,0xB7,0xB6,0x76,0x72,0xB2,0xB3,0x73,0xB1,0x71,0x70,0xB0,0x50,0x90,0x91,
    0x51,0x93,0x53,0x52,0x92,0x96,0x56,0x57,0x97,0x55,0x95,0x94,0x54,0x9C,0x5C,
    0x5D,0x9D,0x5F,0x9F,0x9E,0x5E,0x5A,0x9A,0x9B,0x5B,0x99,0x59,0x58,0x98,0x88,
    0x48,0x49,0x89,0x4B,0x8B,0x8A,0x4A,0x4E,0x8E,0x8F,0x4F,0x8D,0x4D,0x4C,0x8C,
    0x44,0x84,0x85,0x45,0x87,0x47,0x46,0x86,0x82,0x42,0x43,0x83,0x41,0x81,0x80,
    0x40
};
```

参 考 文 献

[1] ANSI/ TIA/ EIA-232-E-1997 Interface Between Data Terminal Equipment and Data Circuit-Terminating Equipment Employing Serial Binary Data Interchange.

[2] ANSI/ TIA/ EIA-485-A-1998 Electrical Characteristics of Generators and Receivers for Use in Balanced Digital Multipoint Systems.

[3] AWG(American Wire Gauge)("AWG"(美国线规)是在美国及其他国家中使用的表示线径的标准方法,参见 1993 年 McGraw-Hill 出版的第 13 期电气工程师标准手册 D. G. Fink and H. W. Beaty).

[4] Modbus. 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2部分:Modbus协议在串行链路上的
实现指南
GB/T 19582.2—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 2.5 字数 66 千字
2008年5月第一版 2008年5月第一次印刷
*
书号:155066·1-31329 定价 28.00 元

GB/T 19582.2-2008