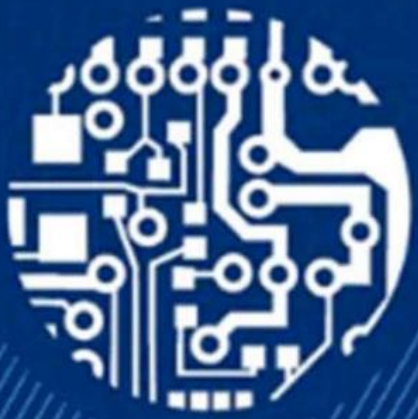
高等院校电子信息类规划教材

Modbus 总线技术与应用系统

Modbus Technology and Application System

主　编　方虎生
副主编　张海涛　芮　挺　杨成松
何宇宸　罗吉庆　朱经纬
张详坡　周建钊　刘　晴



北京邮电大学出版社
www.buptpress.com

Modbus 总线技术与应用系统

Modbus Technology and Application System

内 容 简 介

本书从工程实际应用出发，紧密围绕系统设计与开发需求，系统地介绍了 Modbus 总线技术、51 内核微控制器、硬件设计、嵌入式软件设计、VB 应用软件设计等应用技术。

Modbus 涉及的知识点多、内容广，本书以案例带动知识点开展学习，注重培养读者的理论基础、设计基础和解决实际问题的能力。

本书总结了作者团队在科学研究、装备研发、系统集成领域的最新成果。本书的内容选择合理、结构清楚、图文并茂、面向实际应用。本书适合作为本科生的教学用书，也可作为研究生、工程人员的培训教材或相关科研人员的参考书。

图书在版编目（CIP）数据

Modbus 总线技术与应用系统 / 方虎生主编. -- 北京：北京邮电大学出版社，2023. 10

ISBN 978-7-5635-6994-6

Ⅰ. ①M… Ⅱ. ①方… Ⅲ. ①工业自动控制-通信协议 Ⅳ. ①TP273 ②TN915. 04

中国国家版本馆 CIP 数据核字（2023）第 147692 号

策划编辑：姚 顺 刘纳新 **责任编辑**：满志文 **责任校对**：张会良 **封面设计**：七星博纳

出版发行：北京邮电大学出版社

社 址：北京市海淀区西土城路 10 号

邮政编码：100876

发 行 部：电话：010-62282185 传真：010-62283578

E-mail：publish@bupt. edu. cn

经 销：各地新华书店

印 刷：

开 本：787 mm×1 092 mm 1/16

印 张：9

字 数：155 千字

版 次：2023 年 10 月第 1 版

印 次：2023 年 10 月第 1 次印刷

ISBN 978-7-5635-6994-6 定价：39. 00 元

前　言

随着微处理器技术、通信技术、网络技术及自动控制技术的不断发展，现场总线技术在各领域的应用越来越广泛，Modbus已经成为通信协议的业界标准，是控制设备之间常用的互联方式。Modbus总线与其他总线一样，具有标准、开放的特点，同时可以支持多种电气接口，如RS-232、RS-485等，可以在各种介质上传送，如双绞线、光纤、无线等。Modbus总线协议的帧格式简明、易用，工程设计人员能够快速构建应用系统。

本书围绕总线技术及其应用，在介绍Modbus技术规范的基础上，拓展工程技术应用，涵盖了现场总线技术、微控制器技术、电子技术、VB程序设计技术等。微控制器是总线工程应用的程序运行载体，8051内核的微控制器广泛应用在控制领域，众多厂家基于该内核生产了种类繁多、功能丰富的IC器件。8051内核的微控制器的开发工具便捷，易于工程师设计、编程和应用。面向Modbus嵌入式应用需求，工程师通常需要根据实际设备的空间、布局、接口的特点，开展个性化的软硬件设计，硬件设计主要包括模拟电路、数字电路，在微控制器芯片具体型号确定后，依据数据手册开展硬件设计：首先需要确定微控制器引脚的功能和物理信号接入；其次选择合适的外围器件；再次开展原理图和PCB图的设计；最后制版焊接与测试。微控制器软件设计，通常使用Keil系列开发工具，8051内核相对简单，各功能模块定义清晰，工程师在熟悉具体芯片结构配置特点基础上，就可以开展底层驱动和应用层程序设计，开发出适合特定应用的程序并测试评估。VB开发工具是高效率的开发工具，功能强大，使用Windows内部的广泛应用程序接口（API）函数，使用动态链接库（DLL）、对象的链接与嵌入（OLE）、开放式数据连接（ODBC）等技术，可以快速开发功能丰富、界面美观的应用系统，这使得硬件工程师、嵌入式软件工程师，能够快速利用VB构建上位机的应用系统或者测试系统，高效完成系统评估与验证。

本书主要内容包括：第1章Modbus总线协议，主要介绍Modbus的协议规范和传输标准；第2章AT89S51微控制器，主要介绍51内核的基本结构，功能特点和开发方法；第3章面向Modbus应用的硬件设计，从器件选型、性能参数、原理图设计，阐述硬件设计的过程；第4章面向Modbus应用的控制器软件设计，以典型控制器设计为例，利用Keil开发工具，根据协议规范，开展软件设计；第5章基于VB的串口通信设计，利用串

行通信控件，设计串行通信程序；第 6 章上位机 Modbus 应用软件设计，围绕应用需求，设计软件功能，基于 VB 设计应用系统软件。

本书在编写时考虑到 Modbus 涉及的知识点多、内容广等特点，以案例带动知识点开展学习，注重培养读者解决实际问题的能力。内容选择合理、结构清楚、图文并茂、面向应用，适合作为本科生的教学用书，也可作为工程人员的培训教材或相关科研人员的参考书。

本书总结了团队在设备研发、系统集成领域的最新成果，同时在编写过程中参考了大量书籍、文献及手册资料，在此向各位相关作者表示诚挚的感谢。同时，由于编者水平有限，书中难免有不恰当之处，敬请读者批评指正。

编　者

目　录

第1章
Modbus总线协议

1.1 现场总线技术

现场总线技术是指把单个分散的控制设备变成网络节点，以现场总线为纽带，把它们连接成可以相互沟通信息、共同完成自控任务的网络系统与控制系统。现场总线技术的运用大大减少了系统连线的数量以及布线和连接的难度，完成了低层现场设备间以及与外界设备的信息交换，打破了信息孤岛。基于同一种总线技术的不同厂家生产的现场总线设备可以直接连接在相应的现场总线上，成为网络的一部分，极大地方便了测控系统在工程中的应用、维护与扩充。常用的现场总线主要包括基金会现场总线、LonWorks现场总线、PROFIBUS总线、CAN总线、WorldFIP、DeviceNet、ControlNet、M-bus、Modbus、LIN等。现场总线的应用已经从工业现场覆盖到机器人控制、过程自动化、武器装备、智能交通系统(ITS)等各领域。

1.2 Modbus总线

Modbus协议是应用于电子控制器上的一种通用语言，通过此协议，控制器互相之间、控制器经由网络(例如以太网)和其他设备之间可以通信。目前，它已经成为一项通用工业标准，按照ISO/OSI参考模型规范，Modbus协议模型规定应用层协议和串行链路层协议，Modbus协议模型如图1-1所示。

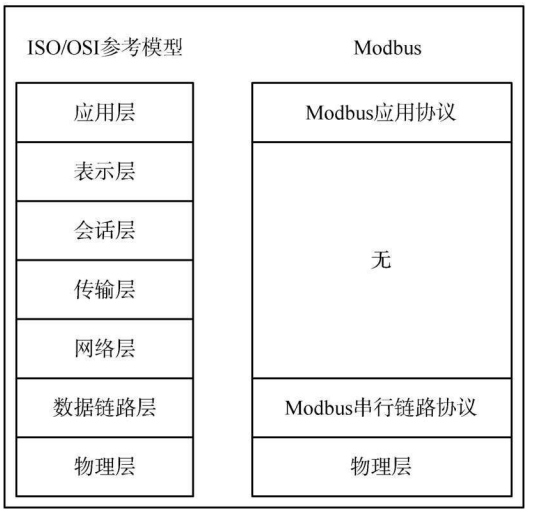


图 1-1　Modbus 协议模型

Modbus 通信采取主从方式通信，即由主设备发起查询消息指令，从设备接收并处理后，回应消息指令给主设备。Modbus 主从设备通信模式如图 1-2 所示。

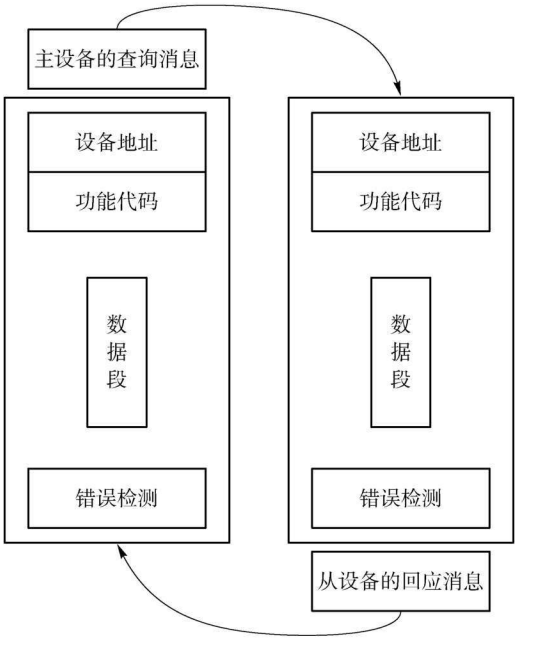


图 1-2　Modbus 主从设备通信模式

Modbus 是一款只定义了应用层和数据链路层协议的总线。由于其并未涉及物理层，所以 Modbus 网络理论上可以通过各种方式进行组网，

比如 RS485、ZigBee、光纤等。此协议只是告诉每一个在总线上的设备该如何与其他设备进行通信。通常一个信息帧格包含以下信息，如表 1-1 所示。

表 1-1 Modbus 信息帧

起始位	设备地址	功能代码	数据	差错校验	结束符

Modbus 具有以下几个特点：

(1) 免费而且标准。这是一款完全免费的总线协议，用户使用该总线不会受到任何的限制，这就方便了商业开发。

(2) 支持多种电气接口。没有物理层的定义，也就方便了 Modbus 的组网，只要是能够构成网络，任何方式都可以使用 Modbus 总线。

(3) 帧格式紧凑、简单。帧的格式非常简单，用户可以根据自己需要，设置功能代码，完成各种功能的设置。

1.3 RS485 接口

RS485 接口的出现主要是随着 20 世纪 80 年代单片机技术在智能仪表方面的快速发展而发展起来的。最初传感器的数据是以模拟信号输出的，后来发展出 RS232 接口，这种接口可以实现点对点的通信方式，但不能实现联网功能。随后出现的 RS485 解决了这个问题。现在 RS485 在工业中有着大量的应用，尤其配合 Modbus 协议，进行组网控制，更是工业控制中不可缺少的一部分。RS485 采用差分信号进行传输，只要差分信号端口的电压差超过 200 mV，芯片即可识别判断。正因这种传输方式，RS485 才能传输超过 1 km，并且可以较好地抑制传输中出现的共模干扰。在逻辑表示上，RS485 采用负逻辑，即用 +2 V ~ +6 V 表示“0”，−6 V ~ −2 V 表示“1”。

RS485 有两线制和四线制两种接线。其中四线制是全双工通信方式，又称为 RS422 通信。需要注意的是，四线制接线方式只适合用作点对点传输。两线制是半双工通信方式，通常使用这种接线方式来构建 Modbus 网络，在同一总线上最多可以挂接 32 个节点。

RS485 网络拓扑支持终端电阻匹配的总线型结构，不支持环形或星形网络拓扑。在构建 RS485 的网络时，应注意如下几点：

(1) 引出线尽量短。通常使用一条带屏蔽的双绞线作为总线，节点通过从总线上引出两条线将设备挂载至总线上的方式进行组网。从总线到

每个节点的引出线长度应尽量短，以便使引出线中的反射信号对总线信号的影响最低。这种由于反射信号造成的影响在短距离、低速的情况下并不明显。但随着通信距离的延长或通信速率的提高，各支路末端反射后的信号与原信号叠加情况变得严重，造成信号质量下降，导致整个系统出错。

(2) 注意阻抗的连续性。出现阻抗不连续的情况通常包括：总线采用了多种电缆，或收发器集中在某一段安装，再或者是分支线过长。而信号遇到阻抗不连续的点时会发生信号反射，所以我们应该避免以上情况出现。

(3) 注意终端负载电阻。虽然根据 RS485 说明需要进行电阻匹配，但在实际使用时，短距离、低速率传输都无须匹配电阻。只有在长距离、高速率传输情况下，我们才需要依据信号衰减程度进行计算，加上匹配负载。

常用的终端匹配的方法是通过在 RS485 总线两端接上 200 Ω 电阻进行匹配，因为大多数双绞线阻抗在 100～200 Ω 之间。这种匹配方法简单而有效，但消耗功率较大，不适合对功率有严格要求的场合。另外一种比较省电的匹配方式是 RC 匹配。利用电容隔断直流成分可以节省大部分功率。但电容 C 的取值是个难点，需要在功耗和匹配质量间进行折中。还有一种“伪匹配”，即通过并联二极管的方法进行匹配。由于匹配是为了消除反射信号对原信号的影响，而二极管的钳位作用恰恰可以迅速削弱总线中的反射信号，使得原信号质量得到改善，因而也可以达到“匹配”的作用。并且此种“匹配”节能效果显著，基于工程装备所需的总线长度较短，负载数量较少，因而无须匹配。

1.4 Modbus 传输模式

Modbus 有两种传输模式：ASCII 模式和 RTU 模式。要在标准的 Modbus 网络上进行通信，需要将控制器设置为其中一种。并且互相通信的设备需要配置成相同的模式以及相同的串口通信参数。

1.4.1 ASCII 模式

当设备在 Modbus 网络上以 ASCII 模式通信时，在消息中的每个字节都作为一个 ASCII 字符发送。这种发送方式的优点是不容易出错，因为即使相邻两个字符的发送中间有长达 1 s 的时间间隔也不会发生出错现象。在 ASCII 模式下，典型的帧格式如表 1-2 所示。

表 1-2 ASCII 模式下的标准帧

起始位	设备地址	功能代码	数据	LRC 校验	结束符
1 个字符	2 个字符	2 个字符	n 个字符	2 个字符	2 个字符

由表 1-2 可得，一个典型的 ASCII 帧以 ASCII 码 3AH(即冒号)开始，以 ASCII 码 0DH(回车)和 0AH(换行)结束。所以，网络上的 Modbus 控制器不断监测网络总线上的冒号字符，当接收到冒号时，控制器开始解码地址域，以判断该帧是否是发来本设备的。如果是本设备帧，也是一直接收，直到回车换行符截止。校验无误后，再解析其他域的代码。在 ASCII 模式下，用于校验的是 LRC(纵向冗长检测)校验码。

1.4.2 RTU 模式

当控制器以 RTU 模式在 Modbus 网络上通信时，在消息中的每个 8 bit 字节包含两个 4 bit 的十六进制字符。很明显，在相同波特率下，采用 RTU 方式传输数据可以比采用 ASCII 模式传输更多的数据。典型的 RTU 帧如表 1-3 所示。

表 1-3 RTU 模式下的标准帧

起始位	设备地址	功能代码	数据	CRC 校验	结束符
T1-T2-T3-T4	8 bit	8 bit	$n$bit	16 bit	T1-T2-T3-T4

使用 RTU 模式，一帧消息的发送至少要以 3.5 个字符时间的停顿开始。不同于 ASCII 帧，RTU 帧的第一个域为地址域，而没有冒号这一用于表示起始位的数据域。处于工作状态的控制器会不断地侦测总线，其不仅会侦测总线上的数据域，还会侦测停顿的间隔时间。当地址域接收到后，控制器将进行解码以判断该消息是否是发往自己的。在结束最后一个字符传输之后，至少 3.5 个字符时间的停顿来标定该消息的结束。此时一个新的消息便可立即开始传输。

在该模式下，每一帧消息都必须连续传输，传输过程中不能出现停顿。如果在一帧数据完成传输之前，停顿时间超过了 3.5 个字符，那么接收设备将认定这一帧消息为不完整消息，删除之前所接收内容，并且会认为下一个字节是一个新消息的地址域，开始接收保存。同样地，如果一个新消息在小于 3.5 个字符时间内开始传输，接收控制器将认为它是前一消息的一部分，这很可能导致该消息的 CRC 校验产生错误，从而造成数据传输错误。

Modbus RTU 和 ASCII 均采用了校验，分别是 CRC 校验和 LRC 校验，这两种不同的传输模式开始标记和结束标记如表 1-4 所示。

表 1-4　Modbus RTU/ASCII 协议

序号	协议	开始标记	结束标记	校验方式
1	Modbus RTU	无	无	16 位 CRC
2	Modbus ASCII	:	CR(回车)，LF(换行)	LRC

1.5　校验方式

标准的 Modbus 串行网络采用两种错误检测方法。奇偶校验对每个字符都可用，LRC 或 CRC 校验则用于每一帧的检测。这些校验码都是由主设备在发送前通过计算产生的，而从设备则是在接收完数据后，再通过相应计算，检测校验码是否正确。

1.5.1　奇偶校验

奇偶校验是根据被传输的一组二进制代码的数位中“1”的个数是奇数或偶数来进行校验。采用奇数的称为奇校验，反之，称为偶校验。通常 Modbus 采用的是串口进行传输，对于串口而言，用户可以配置控制器是奇校验，也可以是偶校验，亦或是无校验。

如果指定了奇偶校验，将会有 9 位传输数据，“1”的位数将算到每个字符的位数中。例如 RTU 字符帧中包含以下 8 个数据位：10100101.如果采用偶校验，帧的奇偶校验位将是 0。如果使用了奇校验，帧的奇偶校验位将是 1。

如果不采用校验，传输时就只有 8 位数据，也不进行校验检测。

通常以上都由串口硬件自动完成，只需进行设置即可，并且在有其他校验方式时，一般都不采用奇偶校验。

1.5.2　LRC 检测

纵向冗余校验(Longitudinal Redundancy Check，LRC)是通信常用的一种校验形式，也称为 LRC 校验或纵向校验。它是一种从纵向通道上的特定位串产生校验位的错误检测方法。

LRC 校验在 ASCII 模式中使用，在 ASCII 消息包括了一个基于 LRC 方法的错误检测域。LRC 域检测了消息域中的地址域、功能域和数据域。

LRC 域是一个包含一个 8 位二进制值的字节。LRC 值由主设备计算得到，并将其附到消息帧的指定位置。接收设备在接收完一帧消息后，立即开始计算 LRC，LRC 校验采取将消息中的字节连续累加，丢弃了进位。LRC 占用 Modbus ASCII 帧 1 个字节长度，由传输设备计算 LRC 值，并和接收到消息中的 LRC 值进行比较，如果两值不等，说明帧数据有错误，需要另作处理。

LRC 为一个 8 位域，那么每个会导致值大于 255 新的相加只是简单地将域的值在“零”回绕。从 FF(11111111)十六进制中减去域的最终值，产生 1 的补码(二进制的反码)加 1 产生二进制补码。由此可得函数如下：

```
Unsigned char calculateLRC(unsigned char*  auchMsg,unsigned short usDataLen)
{Unsigned char uchLRC=0;
While(usDataLen--)
uchLRC+=*  auchMsg++;
Return((unsigned char)(-((char)uchLRC)));}
```

1.5.3 CRC 检测

循环冗余码校验的英文名称为 Cyclical Redundancy Check，简称 CRC，它是利用除法及余数的原理来作为错误侦测(Error Detecting)的。

CRC 校验码都是利用事先约定的多项式 G(*X*)计算得到的。本书主要采用 CRC16 作为校验码，其约定多项式 $G(X)=X_{16}+X_{15}+X_2+1$，对应代码记为 8005。主机根据该多项式，对需要发送的内容进行计算，即可求得一个 16 位的 CRC16 校验的数据帧。接收端在收完整个数据后，对整个数据帧进行同样的计算，包括 CRC 校验域，如果结果为 0，则表明数据正确，不为 0 则表明数据帧在传输过程中出现错误。

1.6 Modbus 功能码

1.6.1 常见的 Modbus 功能码

Modbus 具有许多的功能码，不同的功能码具有不同的功能，表 1-5 为常见的几种功能码以及其功能，主要涉及线圈、离散输入、保持、输入四种寄存器。

表 1-5　Modbus 常见功能码

功能码	功能
01	读取逻辑线圈状态
02	读取开关输入状态
03	读取保持寄存器内容
04	读取输入寄存器内容
05	强置单线圈通断
06	写保持寄存器
07	读取 8 个内部线圈的状态
15	强置多个线圈
16	写多个寄存器

(1)线圈寄存器:实际上就类比于开关量,每个 bit 都对应一个信号的开关状态。对应上面的功能码就是:0x01、0x05、0x0f。

(2) 离散输入寄存器:也是每个 bit 表示一个开关量,而它的开关量只能读取输入的开关信号,是不能够写入的。对应上面的功能码就是 0x02。

(3) 保持寄存器:这个寄存器单位不再是 bit 而是两个 byte,也就是可以存放具体的数据量,并且是可读写的。不仅仅可以读也可以写,而且也可以写多个或者单个。所以对应的功能码有三个:0x03、0x06、0x10。

(4) 输入寄存器:与保持寄存器相似,但是只支持读而不能写。一个寄存器也是占据两个 byte 的空间。对应的功能码:0x04。

1.6.2　读保持寄存器(0x03)

实现读取模拟量输入通道和数字量输入通道数据,指令格式如下:

[:][设备地址][功能码 03][起始寄存器地址高 8 位][低 8 位][读取的寄存器数高 8 位][低 8 位][LRC][CR][LF]

读保持寄存器指令内容如表 1-6 所示。

表 1-6　读保持寄存器指令内容

内容	占用字节个数	取值范围
前缀	1B	:
设备地址	2B	1～247
功能码	2B	0x03
起始寄存器地址	4B	
读取的寄存器数量	4B	N,1～5;(起始寄存器+N) 这个和必须小于等于 5
LRC	2B	
后缀	2B	CRLF