文章编号: 1671- 251X(2009) 05- 0067- 03

基于 Modbus 协议的监控系统的改进方案

郑先锋, 毛景魁, 张开拓

(河南机电高等专科学校, 河南 新乡 453002)

摘要: 文章分析了目前工业控制中广泛使用的基于 RS485 总线 Modbus 协议的监控系统的拓扑结构、协议规范。针对监控系统升级中遇到的问题, 在不改变系统架构的基础上, 提出了一种新的软件改进方案, 即通过重新编写系统底层通信程序, 实现 Modbus 协议的数据通信。具体应用说明了该方案的可行性。

关键词: 监控系统; 串行通信; RS485; Modbus 协议

中图分类号: **TD76** 文献标识码: **B**

0 引言

工业控制已从单机控制走向集中监控、集散控制, 由此应运而生了众多的控制方案或策略, Modbus 协议就是其中之一。Modbus 通信协议简单、易于使用和扩展。结合 RS485 总线可远距离传输的优点, 工业控制中广泛使用基于 RS485 总线的 Modbus 协议监控系统。这些监控系统通常采用专用面板作为操作、显示的人机接口, 需要采用工业控制计算机作为上位机的控制系统。然而由于先前的计算机中没有专用的 RS485 接口, 因此, 多采用专门的信息转发器或者 PCI 板卡实现上位机和监控设备的通信, 其拓扑结构如图 1 所示。

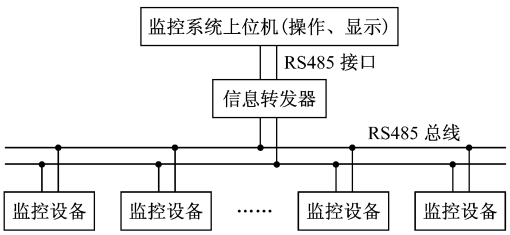


图 1 单机监控系统拓扑结构图

随着现代科学技术的进步, 以工控机作为监控中心的监控系统以其海量的存储能力、灵活的监控功能, 正逐步取代原有的小型面板式监控系统。但由于通信速度的提高和数据量的加大, 图 1 所示的拓扑结构中的信息转发器不但会造成数据通信的“瓶颈效应”, 而且对系统的通信可靠性、稳定性和通信速度的提高都有很大的影响。因为一旦信息转发器出现故障, 挂接在该转发器下的所有监控设备的信息无法传输到监控系统上位机中, 造成整个系统的瘫痪。

由于目前工业控制计算机的通信接口可以根据用户需要定制, 很多采用 RS485 总线结构的控制系统逐步撤除了信息转发器, 将工业控制计算机通过本身的 RS485 接口直接连接到 RS485 总线上, 作为控制系统的核心控制单元, 如图 2 所示。

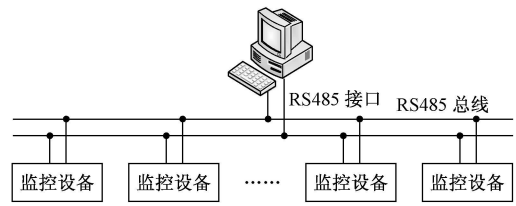


图 2 改进后的监控系统拓扑结构图

图 2 所示的拓扑结构不但可以消除通信的“瓶颈效应”, 还可以提高通信的可靠性, 使系统不会因为某个节点或者某点的物理线路连接出现问题而造成整个系统瘫痪。但在实际的实现过程中, 笔者发现通信仍很难实现。通过仔细的调试和分析观察, 发现主要是因为工控机的通信时序与 Modbus 规范的时序要求不一致造成的。为此, 笔者结合工控机的硬件结构和 Modbus 协议特点, 采用基于 Windows API 函数的软件方案对系统进行改进设计, 重新编写系统的底层通信协议, 撤除系统中原有的信息转发器, 增强了系统的可靠性和灵活性。

1 Modbus 协议

Modbus 协议是 OSI 模型第七层上的应用层报文传输协议, 它可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, 是一个请求/应

收稿日期: 2009- 01- 21

作者简介: 郑先锋(1972-), 男, 河南宁陵人, 硕士, 副教授, 中国电工技术学会高级会员, 电线电缆专委会委员, 现主要从事计算机测控与电气绝缘测试技术方面的教学与研究工作。E-mail: zhengxianf313@163.com

答协议,并且提供功能码规定的服务。

控制器通信使用主- 从技术,即仅主设备能初始化传输(查询),其它设备(从设备)根据主设备查询提供的数据作出相应反应。主设备可单独和从设备通信,也能以广播方式和所有从设备通信。如果单独通信,从设备返回一消息作为回应,如果是以广播方式查询,则不作任何回应。Modbus 协议建立了主设备查询的格式:设备(或广播)地址、功能代码、所有要发送的数据、错误检测域[2]。从设备回应消息也由 Modbus 协议构成,包括确认要行动的域、任何要返回的数据和错误检测域。如果在消息接收过程中发生错误,或从设备不能执行其命令,从设备将建立一错误消息并把它作为回应发送出去。

在其它网络上,控制器使用对等技术通信,故任何控制都能初始和其它控制器的通信。这样,在单独的通信过程中,控制器既可作为主设备也可作为从设备。提供的多个内部通道可允许同时发生传输进程。尽管网络通信方法是“对等”在消息位,Modbus 协议仍提供了主从原则。如果 1 个控制器发送 1 条消息,它只是作为主设备,并期望从从设备得到回应。同样,当控制器接收到 1 条消息,它将建立 1 个从设备回应格式并返回给发送的控制器。

Modbus 协议定义了一个与基础通信层无关的简单协议数据单元(PDU),特定总线或网络上的 Modbus 协议映射能在应用数据单元(ADU)上引入一些附加域,发起 Modbus 事务处理的客户端构造 Modbus PDU,然后添加附加域以构成适当的通信 PDU。Modbus 协议 ADU 数据组成如图 3 所示。

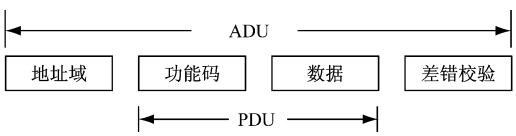


图 3 Modbus 协议 ADU 数据组成图

控制器能设置为 2 种传输模式(ASCII 码或 RTU)中的任何一种,并可在标准的 Modbus 网络上通信。当控制器在 Modbus 网络上以 RTU(远程终端单元)模式通信时,消息的每 8 Bit 字节包含 2 个4 Bit 的十六进制字符。这种模式的优点:在同样的波特率下,可比 ASCII 码模式传输更多的数据。发送至少要以 3.5 个字符时间的停顿间隔开始。传输的第一个域是设备地址,可以使用的字符为十六进制的 0 …9,A …F。网络设备不断侦测网络总线,包括停顿间隔时间在内。当接收到第一个域(地址域),每个设备都进行解码以判断是否是发给自己的。在最后一个传输字符之后,一个至少 3.5 个字符间的停顿标定了消息的结束。RTU 报文帧格式如图 4所示。

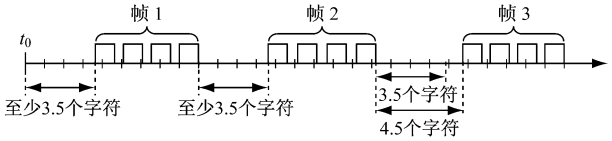


图 4 RTU 报文帧格式图

整个报文帧需以连续的字符流发送。如果2 个字符之间的空闲间隔大于 1.5 个字符时间,则报文被认为是不完整的应该被节点丢弃。如果一个新报文在小于 3.5 个字符时间内接着前一个消息开始,接收的设备将认为它是前一个消息的延续。这将导致一个错误,因为在最后 CRC 域的值是不正确的。

2 通信的同步

要保证监控系统通信的正常进行,需要对通信进行同步设置。这是因为信息发送者对信息的发送是随机的,而信息接收者不知道何时接收到信号,因此,需要有一种机制对两者进行同步处理,即信息发送者发送一标志,信息接收者对该标志进行判断后,开始接收随后发送来的数据信息。

目前,普遍使用的同步方式是字符同步,即信息发送者在发送数据时,先发送特定字符,该字符不能被用于其它功能。另一种方式是位同步,即设定某一位为同步信号,这种方式在采用单片机的控制系统中,由于其简单易行,不占用通信数据资源,被广泛运用,本文所讨论的 Modbus 协议采用的就是位同步。

一般来说,工业控制计算机至少具有 2 个以上的串行通信接口,1 个 RS485 接口。因此,采用带有 RS485 接口的工业控制计算机完全可以采用位同步的方式实现串行通信。工业控制计算机串行通信的校验位设置分为以下 5 种:无、奇校验、偶校验、Mark 和 Space。如果设置为 Mark 时,则串行数据的第九位一直为高电平;设置为 Space 时,则串行数据的第九位一直为低电平。因此,必需在发送 1 帧数据之前,将串口的校验位设置为 Mark,当发送完地址域的数据之后,快速地关闭串口,重新设置为 Space 后,将剩余的数据域数据发送完毕。因此,由于需不断地改变串口的设置,对于底层串口的控制可以采用 MSCom 控件或者 Windows API 函数。

考虑到该监控系统 Modbus 协议采用的是位同步,笔者采用 Windows API 函数的方式和 Visual C++ 6.0 重新编写监控系统底层通信类 CSerialEx

类, 并在初始化完毕后, 启动辅助数据接收线程等待接收数据, 具体实现方案如下文所述。

3　改进方案的实现

使用 Visual C+ + 6.0 可方便地实现 Modbus 协议的组帧和数据解析。在使用 CreateFile 函数打开串行端口后, 调用 GetCommState 函数得到该端口的配置, 根据需要更改波特率、数据位、停止位等, 并通过更改 BCB 的 EvtChar 字段来设置触发信号事件的字符, 然后调用 SetCommState 函数将端口设置为需要的配置。

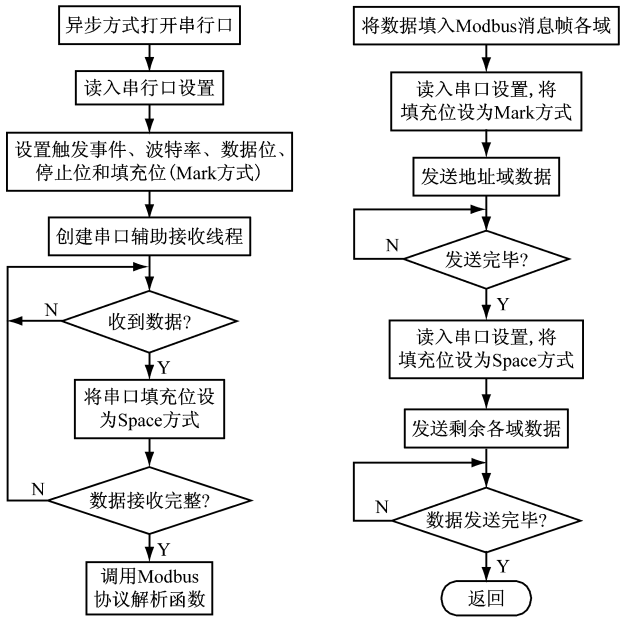
成功打开串口并配置后, 创建一个辅助线程, 该线程根据所设定的触发事件, 不断监视串口, 一旦满足条件, 则向主窗口发送 1 条接收到数据的消息。主窗口消息处理函数调用数据解析函数, 对接收到的数据进行校验、解包, 并根据解析得到的数据进行相应的处理。

数据的发送是采用向文件写数据的方法实现的。在 Windows 操作系统中, 对硬件的控制采用的是文件的方式, 即 Widows 认为所有的硬件都是文件, 对该硬件的操作也就是对于与之关联的文件进行操作。考虑到串行口发送数据比较缓慢, 因此, 采用异步方式打开串口关联的文件, 即程序只负责将数据写入串行口的数据缓冲区中, 由操作系统下的底层 API 函数负责将该数据按照设定的串行口配置发送出去。接收和发送 Modbus 消息帧的程序流程如图 5 所示。

为了满足同步的要求, 在发送地址时需要设置每帧的第九位, 将它和数据区分开来。本文采用 Mark 和 Space 填充的方式进行设置, 在发送每帧之前, 首先使用“1 200, m, 8, 1”配置串行口(Mark 方式), 待地址域发送完毕后, 使用 SetCommState 函数将串行口配置为“1 200, s, 8, 1”(Space 方式)。

采用 SetCommState 函数在发送每帧数据时, 不断更改串口配置, 地址域和后续各域的间隔不会很长, 符合 Modbus 协议规范。

图 6 为采用 Windows API 函数编写的 Modbus 通信程序发送的 Modbus 消息帧波形。由于在发送地址域的数据后立即更改了串行口设置, 可以将地址域和其它各数据域之间的时间设置为最短, 使之符合 Modbus 协议的时序标准。由图 6 可以看出, 由于不需要执行其它任务, 地址域与其它数据域的时间间隔很短, 消息帧的时序正确、数据连续、波形稳定。



(a) 接收 Modbus 消息帧的程序流程图　　(b) 发送 Modbus 消息帧的程序流程图

图 5　接收和发送 Modbus 消息帧的程序流程图

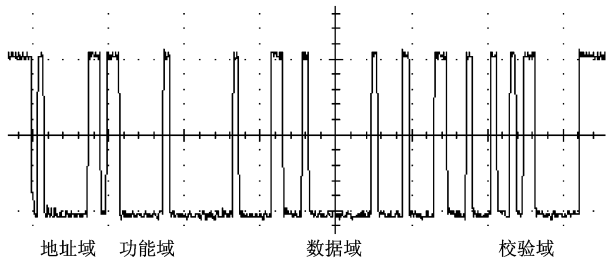


图 6　采用 Windows API 函数编写的 Modbus 通信程序发送的 Modbus 消息帧的波形

4　结语

通过采用 Windows API 函数编写 Modbus 协议消息帧的收发程序, 解决了原有单机监控系统不能直接和工控机通信的问题, 省去了中间的信息转发器, 不但节省了成本, 而且提高了通信的可靠性。按照上述方案编制的程序已经应用于某火灾监控系统的 PC 机程序中。实际运行表明, 性能可靠, 达到了设计的要求。

参考文献:

[1] 陈柏金. 通过串行口访问 Modbus 现场控制网络[J]. 微计算机信息, 2003, 19(1): 52~ 54.

[2] 王成多, 方祥武, 素　莲. 基于 Modbus- RTU 协议的现场总线局域网在智能电器中的应用[J]. 电气制造, 2007(12): 72~ 75.

[3] 卢文俊. 基于 Modbus 协议的控制器远程监控系统[J]. 电力自动化设计, 2003(6): 54~ 56.