

文章编号:1671-251X(2014)04-0118-03 DOI:10.13272/j.issn.1671-251x.2014.04.030
方崇全. 基于 Modbus RTU 协议的 ProConOS I/O 驱动设计[J]. 工矿自动化, 2014, 40(4): 118-120.

基于 Modbus RTU 协议的 ProConOS I/O 驱动设计

方崇全

(中煤科工集团重庆研究院有限公司, 重庆 400039)

摘要:介绍了 Modbus RTU 通信协议的特点,提出了一种基于 Modbus RTU 协议的可编程逻辑控制内核软件 ProConOS 的 I/O 驱动设计方案,并进行了测试与分析。测试和应用结果表明,该方案能有效实现支持 Modbus RTU 协议的不同控制设备的互连互通、工业组网和集中控制,为 Modbus RTU 协议在嵌入式系统中的运用提供了依据。

关键词:ProConOS; I/O 驱动; Modbus RTU 协议

中图分类号:TD655 文献标志码:B 网络出版时间:
网络出版地址:

Design of I/O driver for ProConOS based on Modbus RTU protocol

FANG Chongquan

(CCTEG Chongqing Research Institute, Chongqing 400039, China)

Abstract: The paper introduced characteristics of Modbus RTU communication protocol, proposed a design scheme of I/O driver for programmable logic control kernel software ProConOS based on Modbus RTU protocol, and conducted on test and analysis. The test and application results show that the scheme can effectively realize interconnection, industrial networking and centralized control of different control equipments which support the Modbus RTU protocol, and provides basis for application of the Modbus RTU protocol in embedded system.

Key words: ProConOS; I/O driver; Modbus RTU protocol

0 引言

ProConOS 是 PLC 的实时操作系统,已在全球工业自动化控制领域内得到大量应用。然而 ProConOS 没有提供基于 Modbus RTU 协议的 I/O 驱动,这在工业现场自动化控制使用中很不方便。为提高工业现场控制设备数据传输的标准性和开放性,实现不同厂商生产的控制设备的互连互通和集中控制,本文将 Modbus 协议应用到基于 ARM 的微控制器上,通过定制 WinCE 操作系统,设计了基于 Modbus RTU 协议的 ProConOS I/O 驱动,成功实现了基于 ProConOS I/O 驱动方式的监控设备与多个串行链路 I/O 设备的 Modbus RTU 通信。

1 Modbus RTU 协议

串行链路上的 Modbus 协议有 RTU 和 ASCII 2 种传输模式。RTU 模式信息帧的 8 bit 数据包括 2 个 4 bit 十六进制字符。相对于 ASCII 模式,RTU 模式表达相同的信息需要较少的位数,且在相同通信速率下具有更大的数据流量。因此,通常情况下,一般工业智能仪表仪器都是采用 RTU 模式的 Modbus 规约^[1-4]。

2 ProConOS 简介

ProConOS 作为可编程逻辑控制内核软件,是一种高效的针对复杂控制运用程序的 PLC 运行时

收稿日期:2013-06-13;修回日期:2013-12-09;责任编辑:盛男。

作者简介:方崇全(1978—),男,四川广安人,工程师,硕士,主要研究方向为嵌入式系统、可编程自动化控制器、煤矿自动化,E-mail:fcqmky@163.com。

系统,包含全部 IEC 61131 功能。它是在实时多任务操作系统上开发的,因此保证了其执行效率。ProConOS 结合实时多任务操作系统,为通用工业 PC、VME (VersaModule Eurocard) 总线系统及 PLC 等的测量、控制、调节、可视化等自动化任务提供了可能。

3 I/O 驱动设计

在多任务环境下,ProConOS 每个任务先读输入,执行程序,再写输出。通过指定 I/O 驱动,实现输入和 ProConOS 内部 I/O 区的数据交互,同时定义 I/O 驱动描述字 PlcIODef 结构体来关联内部 I/O 区和具体的 ProConOS 任务。ProConOS 任务调用驱动程序流程如图 1 所示。

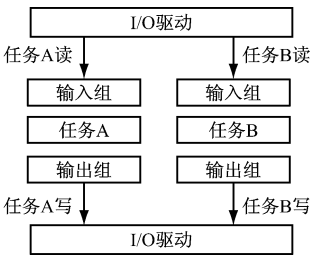


图 1 ProConOS 任务调用驱动程序流程

ProConOS 执行任务程序前读取输入和执行任务程序后写输出时,会通过一个函数入口点 (iodrv 结构体)调用内部函数,定制的 I/O 驱动被调用。

3.1 总体设计

WinCE 平台基于 ProConOS 的 Modbus RTU 协议 I/O 驱动支持的功能码见表 1。

表 1 Modbus RTU 协议 I/O 驱动支持的功能码

功能码	功能描述	数据类型	数据位数
0x02	读离散量输入	物理离散量输入	1 bit
0x01	读线圈		
0x05	写单个线圈	内部比特/物理线圈	1 bit
0x0F	写多个线圈		
0x04	读输入寄存器	物理输入寄存器	16 bit
0x03	读保持寄存器		
0x06	写单个寄存器	内部寄存器/物理	16 bit
0x10	写多个寄存器	输出寄存器	
0x17	读/写多个寄存器		

为了减少 I/O 驱动的通信等待时间,专门设计通信线程。

(1) 每个 PLC 任务有自己独立的通信线程,这是为了使通信的优先级可调,即通信线程的优先级可随任务的优先级不同而不同。

(2) 各 COM 口的通信不互相干扰,由不同的线程负责。

主线程 (PLC 任务线程) 的读写函数和各通信线程之间用一个数据缓冲区交换数据。数据缓冲区中存放着所有输入输出模块的数据,如从设备地址、功能码、起始地址、寄存器个数以及需要发送或接收到的数据等。

基于 Modbus RTU 协议的 I/O 驱动总体设计如图 2 所示。在 DriverRead() 中,首先判断上次通信是否出错,如果没有出错,则将上次通信收到的数据复制到 ProConOS 的 I/O 内存中。然后判断调用 DriverRead() 的任务是否只关联了输入模块。

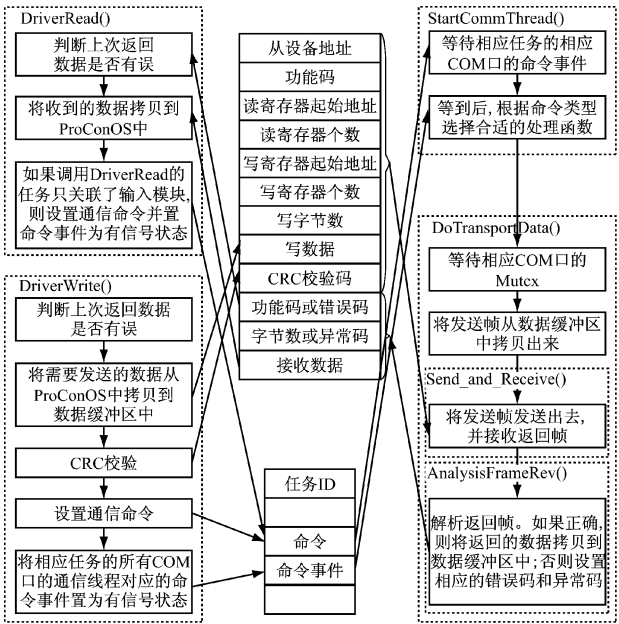


图 2 基于 Modbus RTU 协议的 I/O 驱动总体设计

3.2 串口管理

在 WinCE 下串行端口属于可以打开、关闭、读写流文件方式操作的常规流设备接口。

3.2.1 打开和关闭串口

CreateFile 用来打开或创建串口。

```
hPort = CreateFile (TEXT ( " COM3:"),
GENERIC_READ|GENERIC_WRITE,0, NULL,
OPEN_EXISTING,0, NULL);
```

"COM3:"为串口 3 设备文件名,由于串口不能共享,第 3 个参数必须置为 0,对串口而言,最后一个参数必须为 NULL,由于 WinCE 不支持 Windows 下常用的串行通信重叠 I/O 方式,倒数第 2 个参数置 0。该函数返回已开启的串口句柄或 INVALID_HANDLE_VALUE。

CloseHandle(hPort)用于关闭串口操作。

3.2.2 配置串口

在打开通信设备句柄后,常常需要对串口进行一些初始化配置工作。这需要通过一个 DCB(Data Center Bridging)结构来进行,DCB 结构包含了诸如波特率、数据位数、奇偶校验和停止位数等信息。

一般用 CreateFile 打开串口后,可以调用 GetCommState 函数来获取串口的初始配置。要修改串口的配置,应该先修改 DCB 结构,然后再调用 SetCommState 函数设置串口。

3.2.3 超时设置

在用 ReadFile 和 WriteFile 读写串行口时,需要考虑超时问题。超时设置的作用是在指定时间内没有读入或发送指定数量的字符,ReadFile 或 WriteFile 的操作仍然会结束。要查询当前的超时设置应调用 GetCommTimeouts 函数,该函数会填充一个 COMMTIMEOUTS 结构。调用 SetCommTimeouts 可以用某一个 COMMTIMEOUTS 结构的内容来设置超时。

4 协议测试

以监控主站作为 Modbus 协议测试设备。监控主站以低功耗、外围接口完备的 AT91SAM9263 处理器作为控制核心,选用 WinCE 嵌入式实时操作系统,在其上运行集成了基于 Modbus RTU 协议 I/O 驱动的内核软件 ProConOS。Modbus RTU 协议测试系统网络结构如图 3 所示。

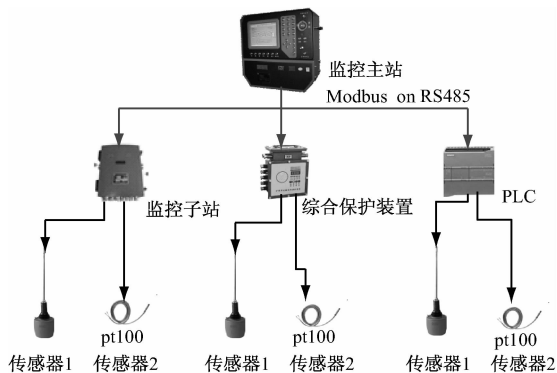


图3 Modbus RTU 协议测试系统网络结构

监控主站 PLC 程序开发与编译环境为 MULTIPROG。结合串口调试工具 Commix1.4,分析 Modbus RTU 请求帧和响应帧。

(1) Modbus RTU 请求帧:02 17 9C 52 00 00 9C 40 00 01 02 00 01 66 B2。其中,“02”为 Modbus 从设备地址,“17”为功能码(读写多个寄存器),“9C 52”为要读取的寄存器起始地址,“00 00”为要读取的寄存器个数,“9C 40”为要写入的寄存器起始地

址,“00 01”为要写入的寄存器个数,“02”为要写入的字节数,“00 01”为要写入的寄存器值,“66 B2”为 CRC 校验码。

(2) Modbus RTU 响应帧:02 17 00 2F F0。其中,“02”为 Modbus 从设备地址,“17”为功能码,“00”为读取的字节数,“2F F0”为 CRC 校验码。

整个测试系统通信正常并稳定运行 1 个月以上,证明基于 WinCE 平台的 ProConOS 可编程内核软件成功实现了 Modbus RTU 通信。

该基于 Modbus RTU 协议的 I/O 驱动已经在工业现场得到运用,工业现场数据采集子界面如图 4 所示。



图4 工业现场数据采集子界面

5 结语

分析了 Modbus RTU 通信协议特点,提出在实时操作系统 WinCE 下,通过可编程内核软件 ProConOS 来实现工业现场与串行链路 I/O 设备的 Modbus RTU 通信,将基于 Modbus RTU 协议的 I/O 驱动成功集成到了可编程内核软件 ProConOS 中。测试结果和现场运用表明,在 WinCE 下基于 ProConOS I/O 驱动方式实现工业现场 Modbus RTU 通信可行。

参考文献:

- [1] 王海波. 基于 CP341 的 Modbus RTU 通信的设计与实现[J]. 工矿自动化, 2009, 35(8): 89-91.
- [2] 张桂, 金国强, 李辉. 基于 ARM 平台 Modbus RTU 协议的研究与实现[J]. 电力科学与工程, 2011, 27(1): 23-25.
- [3] 王力. Modbus RTU 从站通信协议在嵌入式系统中的实现[J]. 石油化工自动化, 2011, 47(5): 46-48.
- [4] 赵小兵, 周雪峰. 煤矿安全监控系统中 RS485 总线的抗干扰设计[J]. 工矿自动化, 2013, 39(2): 83-86.