

基于 WINCE.NET 的多种抄表模式的设计与实现

王 辉, 马维华

(南京航空航天大学信息科学与技术学院, 南京 210016)

摘要: 介绍在 WINCE.NET 平台下基于 UART 的串行通信手持设备的编程方法, 设计出 RS232 抄表、红外抄表、射频抄表等多种抄表模式, 并对串口 PDD 层驱动程序进行移植和修改, 同时为系统注册表添加串口驱动。实验结果证明了多种抄表模式的有效性。

关键词: EVC++语言; 串口编程; 串口驱动

Design and Implementation of Multiple Meter-reading Modes Based on WINCE.NET

WANG Hui, MA Wei-hua

(College of Information Science and Technology, Nanjing University of Aeronautics & Astronautics, Nanjing 210016)

【Abstract】 A programming method about UART serial communication handset equipment based on WINCE.NET platform is introduced. Some multiple meter-reading modes such as RS232, infrared and RF are also designed. The drivers for PDD layer in series are transplanted and modified, and the serial driver is added into the system registry. Experimental results show these meter-reading modes are effective.

【Key words】 EVC++; serial programming; serial driver

随着人们生活质量的提高, 传统的入室抄表方式逐步被自动抄表系统(Automatic Meter Reading System, AMRS)所取代。该系统在小区设置集中器, 通过无线射频或 485 有线方式定期采集住户的水、电、气表的表量信息, 同时抄表员也会定期读记录表量显示数, 并带回数据中心进行统计、汇总、计费处理。为提高抄表员的工作效率, 本文设计了一种具有采集、记录且可进行简单数据处理功能的手持设备, 供抄表员在抄表现场使用, 同时着重讨论手持设备与小区集中器之间多种通信模式的设计与实现。

1 抄表模式

抄表员使用何种抄表模式实际上等同于在手持抄表设备与集中器间使用何种通信方式进行数据交互^[1]。本文中的手持抄表设备端采用嵌入式实时操作系统 WINCE.NET 版本, 具有丰富的通信功能。因此, 双方通信的瓶颈仅存于集中器端。从成本、通信数据量、通信速度和集中器端拥有的硬件资源等方面考虑, 本文采用简单可靠的串行通信方式作为其数据交互方式, 同时采用 RS232 模块、红外模块、射频模块等 3 个串行通信模块作为其外部通信接口。抄表模式如图 1 所示。

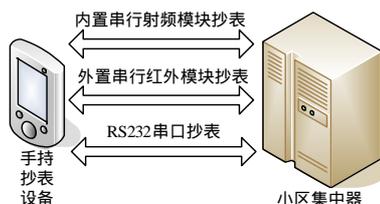


图 1 抄表模式

射频模块采用华奥通公司的 HAC-UM 双向无线数传模块。其较低的功耗、较小的模块体积和大于 300 m 的可靠传输距离使其完全可以内置在手持抄表设备里。将射频模块的 UART 串行接口与手持设备 CPU 的 UART 接口连接, 通过应

用程序控制射频模块数据的收发, 达到远距离抄表的目的。

通常 PDA 中都会内置红外模块用来进行红外数据传输, 但这种红外模块支持的是最高速率达 4 Mb/s 的红外收发器, 同时上层对红外的应用也引入了 IrDA 协议栈的支持, 因此, 想要与集中器中单片机控制的低速红外模块进行通信, 在硬件和软件上都是不可能的。本文设计了一种与集中器具有相同调制方式和载波频率的串口红外硬件模块, 并将其外置, 在应用软件上对串口直接编程, 从而实现手持设备与单片机红外模块间的通信。

RS232 串口抄表模式是常用的 9 针串口通信, 硬件上采用 3 线制接法, 即地、接收数据、发送数据 3 个引脚相连, 软件上对该串口进行编程, 也可满足对集中器中表量数据的读写要求。

由此可见, 以上 3 种抄表模式尽管在硬件通信方式上有所不同, 但它们都是需要连接在手持设备 UART 口上的串行设备, 因此, 从软件上看, 多种抄表模式的实现实际就是应用程序对串口进行不同操作的编程。

2 UART 串行通信

该手持抄表设备采用 Intel Xscale 系列的 PXA255 处理器, 它具有全功能串口 (FFUART)、蓝牙串口 (BTUART)、标准串口 (STUART)、硬件串口 (HWUART) 等 4 种 UART 接口^[2], 其中, 蓝牙串口和具有红外功能的标准串口被 WINCE 操作系统占用, 与蓝牙协议栈和 IrDA 红外协议栈配合使用, 实现蓝牙和红外传输功能。因此, 供用户应用程序使用的只有全功能串口 (FFUART) 和硬件串口 (HWUART)。

2.1 UART 串行通信框架

作者简介: 王 辉(1983 -), 男, 硕士研究生, 主研方向: 嵌入式系统; 马维华, 教授

收稿日期: 2008-02-05 **E-mail:** flyingeagle24@163.com

在 WINCE.NET 分层模型构架下实现多抄表模式的串行通信体系结构如图 2 所示。

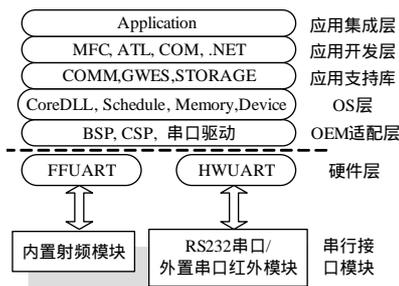


图 2 串行通信体系结构

在硬件上 RS232 串口和红外模块共享一个 UART 接口，具体使用哪种功能可通过跳线在两者间进行切换，同时在软件上对波特率进行相应调整。射频模块选择内置方式且独享全功能串口。从整个体系结构上看，软件层上只有应用集成层和 OEM 适配层需要开发串口应用软件并修改串口驱动。对于其他层次的组件实体，操作系统和应用开发环境都给予一定程度的支持。

2.2 串行通信应用开发环境

开发工具所提供的应用开发环境是上层应用程序与操作系统间的桥梁。由于手持设备的硬件资源有限，因此要求操作系统和应用软件尽量减少对系统资源的消耗，同时还要保证较高的执行效率。目前比较流行的 WINCE 平台应用程序开发工具有 EVC++ 和 VS.NET。两者相关性能的比较如图 3 所示。

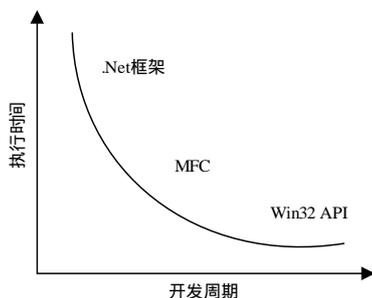


图 3 开发工具的性能比较

VS.NET 是基于 .NET 框架的开发软件，虽然其开发周期短，但在执行效率和资源节省程度以及对串口硬件的访问速度等方面都不如 EVC++。因此，本文采用 EVC++4.0 作为开发工具，并结合 MFC 和 Win32 API 两种技术进行串口应用程序的开发。

3 串行通信功能实现

要想在 WINCE.NET 平台上实现多串口的通信功能，除了编写相应的串口应用程序外，还要完成对串口驱动的移植和修改，使应用程序能通过操作系统成功访问底层串口硬件。

3.1 串口通信编程

在 WINCE 平台下对串口硬件的访问不是直接对端口进行操作，而是把串口当作文件来处理。通过使用 Win32 API 提供的一些文件 I/O 函数如 CreateFile(), ReadFile(), WriteFile(), CloseHandle() 等来打开、读写和关闭串口^[3]。

为达到程序复用的目的，本文运用一个通用串口类 Class CCESerial 来封装这些基本的串口操作，通过输入不同的参数 COM 索引值建立不同的串口实体，利用实体完成对底层串口

FFUART 和 HWUART 的访问。由于 WINCE 不支持重叠 I/O 操作，因此可分别用单独的线程去读写串口，通过多线程来模拟重叠操作^[4]。串口读数据的速度通常较慢，读到的数据还要进行处理，因此，需要创建一个线程专门负责数据处理。

以上这些线程创建工作都要在打开串口操作后进行，具体分为以下几个步骤：(1)调用 CreateFile() 函数，打开串口；(2)填充 DCB 结构体，配置串口属性；(3)填充 COMMTIMEOUTS 结构体，设置端口的读/写超时时间；(4)创建读/写事件；(5)创建读线程；(6)创建写线程；(7)创建读数据处理线程；(8)结束返回，等待系统消息响应。此时，系统创建了读线程、写线程以及读数据处理线程。这 3 个线程的处理流程如图 4 所示。

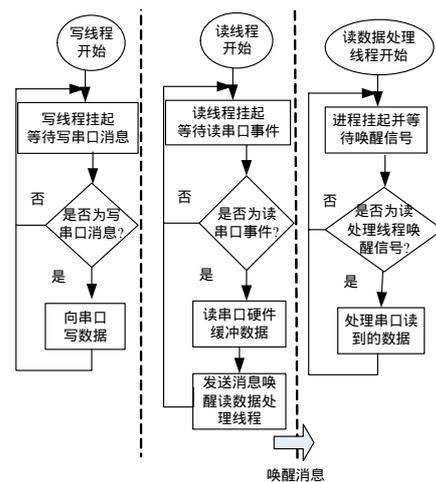


图 4 多线程处理流程

在写线程中，由于数据是主动发送的，因此在向串口写数据时可以采取适当的延时间隔来避免写进硬件缓冲且准备发送的数据被下一次写数据冲掉。另外，读线程和读数据处理线程间需要一个消息机制来进行线程同步。在读线程中，当串口硬件缓冲区中读到数据后，马上发消息唤醒另一个线程对其进行处理，自己继续等待下一个串口事件。

3.2 串口驱动修改

在标准的 WINCE.NET 平台注册表中并未加载相应的串口驱动。同时笔者所使用硬件平台的 OEM 中也未添加对 HWUART 串口中断的支持。因此，应用程序要想使用 HWUART 串口就必须先加载它的串口驱动。

3.2.1 串口驱动的中断处理

串口驱动是内建分层的流接口驱动程序，在系统结构上分为模型设备驱动(MDD)和平台相关驱动(PDD)两层^[5]。MDD层包含了给定类型所有驱动的通用代码；PDD层由特定硬件平台的代码组成。MDD层调用PDD层的函数来访问硬件信息。PDD层程序一般由OEM厂商或用户自己来编写。

WINCE.NET把中断处理分为两部分：中断服务例行程序(ISR)和中断服务线程(IST)。ISR与硬件进行交互，是基本的中断处理程序，常驻在OEM适配层，因此，代码短小快速^[6]。对于串口而言，其ISR实现过程为：(1)当串口硬件产生中断时由IRQ跳转到OEMInterruptHandler()进行中断处理；(2)读中断控制寄存器中的ICIP寄存器值，判断发生中断的设备类型；(3)屏蔽该设备中断；(4)返回该设备中断所对应的系统中断号 SysIntr。

串口的中断服务线程是在串口驱动 MDD 层实现的，而

所有对中断的处理工作都是在该线程中进行的。

IST 是个独立线程, 创建后被挂起, 并一直等待串口事件。当内核收到 ISR 返回的系统中断号后, 设置与该中断号绑定的串口事件, 以唤醒 IST 中断服务线程。具体工作步骤为: (1)读取该 UART 串口的内部寄存器 IIR(中断识别寄存器)值; (2)判断 UART 设备中断类型; (3)根据类型不同进入相应的中断处理程序; (4)解除中断屏蔽。

3.2.2 串口驱动修改

由于UART串口是个符合 16550 标准规范的常用外部串行接口, 因此其与平台无关的驱动程序MDD层已经相当成熟, 可以移植过来直接使用。需要修改的仅是PDD层中与串口中断相关的部分, 同时还要将串口驱动添加到系统注册表中^[5]。又因为OEM厂商已经添加了FFUART的串口驱动, 所以现在只需将HWUART串口驱动添加到系统中即可。

在 OEMInterruptHandler() 中断处理函数中添加对 HWUART 串口中断的响应, 并返回相应的系统中断号:

```
if(ipreg_copy & INTC_HWUART)
//ipreg_copy:中断挂起寄存器值
{
// 如果是 HWUART 硬件中断
INTC_HWUART_INT_DIS(v_pICReg->icmr);
//屏蔽该设备硬件中断
return SYSINTR_SERIAL2;
// 返回系统中断号给内核, OEM 硬件厂商提供值为 16
}
```

在 PDD 层代码中, GetSerialObject()返回的 HWOBJ 结构体起到了连接 MDD 层抽象操作和 PDD 层具体实现的作用, 在这个函数中同样要添加对 HWUART 串口硬件实体的支持代码:

```
HWOBJ SerObj2 =
{
THREAD_AT_INIT, // 串口初始化的时候启动 IST 线程
SYSINTR_SERIAL2, // HWUART 口系统中中断号
(PHW_VTBL) & XSC1_SerVTbl
// 具体硬件操作函数指针的集合, 包括初始化、打开、关闭、
//接收、发送、设置 Baudrate 等一系列操作
};
...
```

(上接第 254 页)

参考文献

- [1] 洪 玫. 人力资源信息化管理[M]. 北京: 中国发展出版社, 2006.
- [2] 张继焦. 价值链管理[M]. 北京: 中国物价出版社, 2001.

(上接第 266 页)

广域分布式文件管理系统中的存储空间、统一命名空间以及文件共享与授权等关键问题。下一步工作将研究数据的可靠传输、分布式数据一致性以及性能优化等问题。

参考文献

- [1] Adrew S T, Marten V S. Distributed Systems Principles and

```
if (4 == DeviceArrayIndex)
// 注册表中 HWUART 的设备索引号, 用于区分同类设备
return(&SerObj2); // 返回一个 HWUART 串口硬件对象
对串口驱动 PDD 层中断相关代码的修改需要注意的是:
不管是硬件中断号还是系统中断号都不能重复或有误, 否则
会引起系统冲突。
```

注册表中有关串口驱动添加部分的主要代码如下:

```
IF BSP_NOSERIAL !
[HKEY_LOCAL_MACHINE\Drivers\BuiltIn\Serial]
// HWUART 口驱动在注册表中键值的位置
"Dll"="xsc1bd_serial.Dll"//注册表在系统中加载的串口驱动名称
"ToBase"=dword:a7800000 // HWUART 口 GPIO 物理地址映
//射到虚拟地址空间的 Cached 地址, 驱动程序通过它来访问
//HWUART 相关的寄存器
...
"DeviceArrayIndex"=dword:4// 多个串口对象存在时, 标识
HWUART 的设备索引号
"Prefix"="COM" //应用程序使用的设备描述符前缀
"Order"=dword:0 // 驱动加载时的顺序
```

4 结束语

本文在 WINCE.NET 串口通信框架的基础上, 使用射频、红外等模块来添加串口驱动程序, 同时编写应用层串口程序, 实现了手持抄表设备的红外抄表、射频抄表、RS232 抄表等多种抄表模式。今后的工作将不再采用二次开发模式, 而是完全自主研发抄表应用程序, 以提高整个系统的稳定性, 降低产品成本。

参考文献

- [1] 卢建宁, 杜广永. 一种基于 WINCE 的抄表设计方案[J]. 电测与仪表, 2003, 40(4): 38-40.
- [2] Intel LTD. PXA255 Processor Developer's Manual[Z]. 2004.
- [3] Boling D. Programming Microsoft Windows CE[Z]. 2001.
- [4] 汪 兵, 李存斌. EVC 高级编程及其应用开发[M]. 北京: 中国水利水电出版社, 2005.
- [5] Microsoft LTD. Microsoft Windows CE.NET 4.2[Z]. 2002.
- [6] 王日宏. 基于 WINCE 的串口外设接口驱动程序实现[J]. 微电子学与计算机, 2004, 21(10): 124-125.

- [3] 薛华成. 管理信息系统 [M]. 3 版. 北京: 清华大学出版社, 2002.
- [4] 顾培亮. 系统分析与协调[M]. 天津: 天津大学出版社, 1998.
- [5] 王元放. 基于层次分析法的信息化项目评估模型[J]. 计算机工程, 2007, 33(8): 68-70.

Paradigms[Z]. 2002.

- [2] Barkes J, Barrios M, Cougard F. A Parallel File System[Z]. 1998.
- [3] 徐志伟, 冯百明, 李 伟. 网格计算技术[M]. 北京: 电子工业出版社, 2004.