基于 Windows CE．NET 平台的串行通信实现

吕秀平　李小民*　杨　毅[1]

(军械工程学院光学与电子工程系,石家庄 050003; 武汉士官学校[1],武汉 430000)

摘　要　讨论了基于 Windows CE. NET平台的串行通信应用开发,分析了 Windows CE. NET下串行通信的特点。介绍了在 Embedded Visual C + + 4. 2开发环境下如何利用串行通信 API函数,结合串口事件、多线程等技术手段,编写出高质量的通信程序。提出了程序设计的基本思路,并给出了关键程序的代码。该程序在某基于 WINCE的嵌入式系统中运行良好,完全满足系统通信要求。

关键词　嵌入式系统　Windows CE. NET　串行通信　多线程

中图法分类号　TP311. 1;　**文献标识码**　A

Windows CE. NET是微软公司向嵌入式领域推出的一款紧凑、高效的 32位嵌入式操作系统,具有多任务、实时性、模块化及可伸缩性、强大的通信和多媒体能力等特点[1],在移动计算、工业控制和信息家电等诸多领域都有广泛的应用。

通讯能力是基于 Windows CE. NET设备的一个关键性特征。在嵌入式系统开发过程中,常常需要面临与外围设备通信的问题,对此 Windows CE. NET提供了广泛的硬件支持和通讯方式,对于用电缆、调制解调器或者红外传输介质连接的设备提供了串行通信支持。尽管在计算机外围通讯中已经有了比串口更快、更稳定的接口,但是在基于 Windows CE. NET的嵌入式设备中,串行通讯仍然是最基本、最常用的通讯方式,被绝大部分 Windows CE. NET设备所支持。例如,目前流行的基于 Windows CE. NET的手持设备一般都具有导航和电话功能,而 GPS和 GSM/GPRS模块都是外置串口的终端设备。因此,在很多基于 Windows CE. NET的嵌入式系统开发中,串行通信的实现是一个重要环节。

2006年 9月 7日收到

第一作者简介:吕秀平 (1979—),男,江西萍乡人,信号与信息处理硕士研究生,研究方向:嵌入式检测与故障诊断。E - mail: LvXiuPing2005@Yahoo. com. cn。

*通信作者简介:李小民 (1968—),男,河北人,教授,博士生导师,研究方向:自主导航与控制技术。

1　Windows CE. NET下的串行通信

基于 Windows CE. NET的操作系统平台可以在集成开发环境 Platform Builder 4. 2中根据需要进行定制。若要实现串口通信,需要平台提供对串行通信的支持;若不支持,可在 Platform Builder中定制一个平台,并将 Catalog/ Core OS/ Core OS Service/ Serial Port Support特性加入到平台,然后添加设备驱动。

Windows CE. NET系统中的设备驱动分为两种类型:本地设备驱动和流设备驱动。本地设备驱动程序由原始设备制造商创建,而流设备驱动程序由第三方生产商提供,用来支持添加到系统中的设备。串行端口是流接口设备,其驱动属于流设备驱动[2]。Windows CE. NET提供了通用串口驱动程序 COM16550,可以直接从 Platform Builder中将其加入自定义的平台;如果不满足需要,则要自己开发串口驱动。驱动开发完成后,使用 CEC编辑工具将驱动文件生成一个“. CEC”文件,将其添加到 Catalog中,然后加入平台。要实现正确的串口通信,用户还必须在“. REG”文件中正确设置注册表键值,尤其是中断和基地址,必须与嵌入式板上所给出的串口中断和基地址一致,否则将无法正常通信。

尽管 Windows CE. NET支持标准的桌面 Windows串行通信函数,其串行通信的实现与一般桌面

Windows系统还是有一些区别[3]:

(1)Windows CE. NET不支持桌面 Windows下常用的串行通信重叠 I/O方式 (OVERLAPPED),不能使用异步 I/O方式来操作串口设备;

(2)Windows CE. NET是一个基于 Unicode的操作系统,其 API函数都是 Unicode格式的,因而提供给 CreateFile等 API函数的字符串参数必须使用 Unicode字符串;

(3)Windows CE. NET下的设备名称书写为"XXXn:"的形式 (n为 0～9的整数),串口设备名必须写为"COMn:",最后的冒号不能省略。

2 程序实现

无论是 Win32平台还是 Windows CE平台,Visual C++都是一个强大的开发工具。而 Embedded Visual C++ (EVC)也是 Windows CE. NET平台上应用程序开发的主流工具。EVC支持 MFC类库的子集,包含有许多 Windows CE. NET专用的标准开发工具,可以给开发者提供最强大的支持,迅速开发出各种应用程序,也使 Win32平台上的 VC程序员可以很容易地迁移到 Windows CE平台上。下面介绍在 EVC4.2下如何利用串行通信 API函数实现串口通信。

2.1 Windows CE. NET串行通信 API函数简介[4]

Windows CE. NET的 API是 Win32 API的一个子集,支持近 1 500个 Win32 API,其串口通信 API函数和其他大多数 Windows的通信函数相同。这些函数用于打开、关闭和操作串行端口,发送和接收数据,管理连接。主要包括:(1)打开串口。在使用串口之前,首先应使用 CreateFile函数打开串口,返回一个串口句柄;(2)串口配置。打开串口后用 SetCommState函数进行端口配置,包括波特率和缓冲区设置,流控制机制和数据格式等;(3)读写串口。通过 ReadFile和 WriteFile函数来读写串口,读和写的超时时间由 SetCommTimeouts函数设置;(4)设置串口事件。用 SetCommMask函数设定要查找的一个或多个串口事件,然后调用等待函数 WaitCommEvent监视串口事件;(5)关闭串口。在使用通信资源结束后,应调用 CloseHandle函数来关闭串口句柄,释放资源。

2.2 程序设计思路

虽然 Windows CE. NET不支持重叠 I/O操作,但可以采用多线程来执行类似重叠操作。可以创建读和写线程,当主线程正忙时,即可运行单独的线程来处理 I/O同步操作。进行串口通信处理的不同线程之间需要协调运行,也就是实现同步,采用的方法是使用事件对象。事件对象允许一个线程减缓它的运行,并且等待直到指定事件发生,即线程处于阻塞状态,此时线程只消耗少量的电能和 CPU处理能力,可以使应用程序对 CPU的利用率大大提高,加快信息处理速度,从而提高通信程序的实时性和增大数据吞吐量。因此,在串口通信中可以设置通信事件,方法是:先用 SetCommMask函数设定要查找的一个或多个串口事件,然后调用等待函数 WaitCommEvent,指定导致这个函数返回的串口事件。利用 WaitCommEvent函数阻塞线程,直到指定的串口事件发生。当 WaitCommEvent函数返回后,处理相应的 I/O操作,如循环调用 ReadFile函数,读回所有接收到的字符等。

综上所述,在 Windows CE. NET下实现串行通信可以采用多线程技术。使用用户界面线程作为主线程来进行数据和消息处理,以及提供友好的用户界面;而使用后台线程来进行串口读写操作。其中,读线程用来监视串口状态,依靠串口事件驱动,查看有无数据到达、通信有无错误等。当数据到达时,及时读取并通过回调函数指针将数据送回主线程;写线程由主线程发送的线程消息触发,向串口写入数据。

2.3 关键代码的实现

这里只给出关键步骤的代码。

(1)打开串口并初始化

```
m_hComm = CreateFile(L"COM1:", GENERIC_READ | GENERIC_WRITE, 0, NULL, OPEN_EXISTING, 0, NULL);
if (m_hComm == INVALID_HANDLE_VALUE)
{ TRACE(_T("串口打开失败"));
return FALSE; }
//得到打开串口的当前属性参数,修改后再重新设置串口。
DCB commParam;
GetCommState(m_hComm, &commParam);
……//此处设置 DCB属性参数,略。
SetCommState(m_hComm, &commParam)
//配置超时值。
COMMTIMEOUTS CommTimeOuts;
GetCommTimeouts(m_hComm, &CommTimeOuts);
……//此处设置具体各项超时值,略。
```

```
SetCommTimeouts( m_hComm, &CommTimeOuts )
//创建读串口线程。
m_hReadThread = CreateThread (NULL, 0, ReadThread, this, 0,
&m_dwReadThreadID);
//创建写串口线程。
m_hWriteThread = CreateThread (NULL, 0, WriteThread, this, 0,
&m_dwWriteThreadID);
```

需要说明的是:在 CreateFile函数的参数中,共享参数必须设置为 0,表示独占方式,安全参数和模板文件参数必须设置成 NULL,而参数 dwFlagsAndAttributes也只能设置为 0,因为 Windows CE. NET不支持重叠 I/O[4]。串口的配置不需要设置很多参数,默认的配置大部分是不需要修改的,一般改动就是波特率、数据位、奇偶校验等几项。超时值是需要改动的,设置超时是一个关键,如果设置太小可能丢失数据。在一个嵌入式设备中,串口通信通常只是其事务处理的一部分,所以必须考虑程序执行的效率问题。当串口初始化成功后,就使其处于等待通信事件的状态,减少消耗的 CPU时间,提高程序运行效率。

(2)读串口线程

```
DWORD ReadThread (LPVOID lparam)
{   HANDLE hPort = *(HANDLE*) lparam;
    DWORD evtMask;
    //清空串口。
    PurgeComm ( hPort, PURGE_RXCLEAR | PURGE_TX-
    CLEAR );
    //指定端口监测的事件集。
    SetCommMask (hPort, EV_RXCHAR | EV_CTS | EV_
    DSR );
    while (TRUE)
    {   if (WaitCommEvent (hPort, &evtMask, 0))
        {   //重新设置端口监测的事件集。
            SetCommMask (hPort, EV_RXCHAR | EV_CTS
            | EV_DSR );
            //表示串口收到字符。
            if (evtMask & EV_RXCHAR)
            {   //用 ReadFile函数读取串口,而后通过回调
            函数指针将数据送回主线程,在此略。}
            }
            //如果收到读线程退出信号,则退出线程。
            if (WaitForSingleObject (m_hReadCloseEvent,
            500) == WAIT_OBJECT_0)
            {   break;   }
    }
}
```

(3)写串口线程

```
DWORD WriteThread ( LPVOID lparam)
{   HANDLE hPort = *(HANDLE*) lparam;
    MSG msg;
    while (TRUE)
    {   //如果捕捉到线程消息。
        if (PeekMessage (&msg, 0, 0, 0, PM_REMOVE))
        {   if (msg.hwnd != 0)
            {   TranslateMessage(&msg);
                DispatchMessage(&msg);
                continue;   }
            //如果收到写串口消息。
            if (msg.message == CM_THREADCOMMWRITE)
            {   //用 WriteFile函数写串口,在此略。   }
    }
    //如果收到写线程退出信号,则退出线程。
    if (WaitForSingleObject (m_hWriteCloseEvent, 500) == WAIT_
OBJECT_0)
    {   break;   }
    }
}
```

在整个应用程序结束时或不再使用串口设备时,应将串口设备关闭,包括取消事件监视,清除发送/接收缓冲区和关闭设备句柄。

3 结论

本文讨论了基于 Windows CE. NET平台的串行通信应用开发,提出了程序设计的基本思路,并给出了关键程序的代码。程序在 Embedded Visual C++ 4.2下编译通过,在基于 Windows CE. NET的嵌入式单板机和单片机构成的主从分布式数据采集与控制系统中运行良好。在实际应用中,我们可以将这些串行通信函数和成员变量加到一个已有的 CWnd类或其派生类中来实现串行通信,也可设计一个新的串行通信类来包含这些成员函数和成员变量。

参 考 文 献

1 周毓林, 宁 杨, 陆贵强, 等. Windows CE. net内核定制及应用开发. 北京: 电子工业出版社, 2005

2 傅 曦, 齐 宇. 嵌入式 Windows CE开发技巧与实例. 北京: 化学工业出版社, 2004

3 Muench C. Windows CE权威指南. 精英科技,译. 北京: 中国电力出版社, 2002

4 汪 兵, 李存斌, 陈 鹏. EVC高级编程及其应用开发. 北京: 中国水利水电出版社, 2005

(下转第 1758页)

程序中,其维护仅仅只需要修改中间件的配置文件。同时系统也充分考虑到了数据源的扩展问题,对于中间件没有的数据库类型,用户可以自行开发包装器,然后只需要在中间件的配置文件中增加相应的类型即可。

2 结束语

本文对现有的异构数据库集成技术进行了分析,并结合实际运用中遇到的问题,设计了异构数据库集成中间件的设计方案。该中间件解决方案具有一定的可行性和实用性,能够胜任异构数据库的集成工作。

参 考 文 献

1 潘大四. 基于 XML的异构数据库数据集成的关键技术研究. 中国优秀硕博论文数据库; 2004; 11

2 邱万彬. 数据库之 XML中间件设计与应用. 中国优秀硕博论文数据库, 2005; 04

3 杜军威,隋树林. 基于 XML的异构数据库模式转换的实现. 青岛科技大学学报, 2005; 26(2): 158—161

4 张 雷. 基于 XML的异构数据集成研究. 北华大学学报, 2005; 6(4): 382—384

5 何 慧, 孙 芙, 李 遂,异构数据库数据类型转换模型. 计算机技术与工程设计学报, 2005; 26(9): 2461—1463

Heterogeneous Database Integration Middleware Design

CHEN Yong, JIANG Ze-jun*

(Software Institute, Northwestern Polytechnical University, Xi'an 700046, P. R. China)

[Abstract] This article first has carried on the analysis to the existing heterogeneous database, then proposed heterogeneous data integration middleware design proposal, and elaborated emphatically [this middleware] each module detailed design content, and design concept in. Finally this article has carried on the applied analysis and the summary to this design proposal

[Key words] heterogeneous database data integration middleware

(上接 1754页)

Realization of Serial Communication Based on Windows CE. NET

LU Xiu-ping, LI Xiao-min, YANG Yi[1]

(Department of Optics and Electronics Engineering, Ordnance Engineering College, Shijiazhuang 050003 P. R. China, Ordnance Engineering School[1], Wuhan 430000 P. R. China)

[Abstract] The development of serial communication based on Windows CE. NET is discussed, and analysed its characteristic. It is introduced how to write high-quality serial communication program in Embedded Visual C++ 4.2 with API functions. The way of program design is put forward, and the code of key program is also provided. The program runs well in one embedded system based on WINCE, and meets the request of the system's communication completely.

[Key words] embedded system Windows CE. NET serial communication multithread