

# 高速串口的软件设计模式研究

姚蓓窈, 向东游, 张华栋

(北京机电工程研究所, 北京 100074)

**摘要:** 高速串口具有较大数据吞吐量, 在耗时可观的任务环境中, 极易产生因读取串口缓存不及时而导致数据丢失的现象; 针对该缺陷, 提出了一种适用于高速串口的上位机软件设计模式, 采用 Windows API 和重叠 I/O 方法搭建串口数据的读写操作; 该模式运用多线程技术, 对其数据一致性问题进行了论证; 对于使用串口传输 JPEG 图像压缩码流的特殊应用场合, 进行了优化设计; 对使用波特率为 2.5M 的高速串口传输尺寸为 640 × 480、帧频为 12Hz、压缩比为 12.5 的图像测试表明, 提出的设计模式满足工程应用需求。

**关键词:** 高速串口; 设计模式

## Software Design Model of High Speed Serial Communication

Yao Beiyao, Xiang Dongyou, Zhang Huadong

(Beijing Research Institute of Mechanical and Electrical Engineering, Beijing 100074, China)

**Abstract:** The high-speed serial communication works with a mass of data streams. The buffer overflows in result of reading delayed in situation of time-limited tasks. The paper proposed a software design model to solve the problem. The design model used Windows API and overlapped functions to implement reading and writing operation. Multithread technique was used and synchronization was discussed. For the application of transfer image streams compressed by JPEG with serial communication, the design model was optimized. The test of transferring image data with size 640 × 480, frame frequency 12Hz, compression ratio 12.5 proves that the proposed design model meets the needs of engineering application.

**Keywords:** high-speed serial communication; design model

## 0 引言

在工业测控应用领域, PC 机与下位机主要采用以 RS232、RS422 为电气标准的串行接口方式。由于串行接口方式的需求依然存在, 目前出现了两种普遍应用的串行接口卡。一类是多串口接口卡, 这类接口卡一般采用跳线方式来兼容 RS232 和 RS422 两种标准; 另一类则是 USB 转串口接口卡, 这类接口卡与 PC 机采用 USB 接口方式因而具有热拔插特性, 但在 PC 机中被映射为串口设备, 这一特点使得上位机程序可完全按照串口设备进行设计。

随着测控领域对数据传输带宽的要求不断增加, 串行接口卡呈现出高码率的特点, 其波特率甚至高达 8M。本文针对高码率串口的特点, 提出一种上位机软件设计模式, 解决复杂任务环境下由于码率高而引起的接收缓冲区溢出问题。针对使用串口传输图像数据这一具有特殊要求的应用场合, 对该设计模式进行了改进, 测试结果表明该设计模式具有普遍的应用价值。

## 1 MSComm 控件的应用局限性

MSComm 控件是微软采用 ActiveX 技术设计的一种应用非常普遍的串口控件, 该控件采用事件方式通知应用程序串口设备已接收到一定数量的数据。对于 PC 机配备的标准串口收发器而言, 其波特率一般不超过 256 000, 亦即每秒数据吞吐量小于 32 kB。MSComm 控件的接口函数允许设置的最大接收缓冲区为 32 kB (MSComm 提供的设置接收缓冲区的参数为

signed short 型), 表明在传输带宽完全被占用且波特率为 256 000 的条件下, 应用程序读取串口的最大可允许延时不超过 1 s, 否则会造成接收缓冲区溢出而丢失数据<sup>[1]</sup>。

对于小规模的应用程序而言, 1 s 的延时要求很容易满足, 一般不会出现接收缓冲区溢出的情况。对于高速串口, 其波特率一般为 2 M 至 4 M, 亦即每秒数据吞吐量为 0.25 MB 至 0.5 MB, 对于 32 kB 的接收缓冲区而言, 可允许的最大读取延时为 64 ms 至 128 ms, 对于波特率为 8 M 的串口卡而言, 最大读取延时仅为 32 ms。

在采用 MSComm 控件进行高速串口的上位机软件设计时, 一般在主窗口中响应串口事件, 不具备太大的灵活性<sup>[2]</sup>。主程序中一些具有可观耗时的任务或主窗口的屏幕刷新均有可能导致读取延时超过 128 ms。此外, 在该台计算机中若同时运行着另一 CPU 使用率较高的应用软件, 也可能导致响应串口事件不及时。因此, MSComm 控件在高速串口的上位机软件设计中具有很大局限性。

## 2 基于 Windows API 的串口软件设计

微软 Windows 平台将 PC 机的所有外围设备均映射为文件, 因而对其读写等操作均与读写硬盘上的实际文件相同。对文件的读写操作, Windows API 提供了非重叠 I/O 和重叠 I/O 两种方式。调用一个读或写文件的 API 函数时, 在重叠 I/O 方式下, 无论读或写操作是否完成, 该 API 函数立即返回; 在非重叠 I/O 方式下, 直到读或写操作完成时, 该 API 函数才返回。在多线程应用程序中, 重叠 I/O 方式具有更高的效率, 在读写操作不能立即返回时, 读写操作会自动转入后台运行<sup>[3]</sup>。

在采用 Windows API 进行串口上位机软件设计时, 主要

收稿日期: 2014-03-05; 修回日期: 2014-04-10。

作者简介: 姚蓓窈(1989-), 女, 陕西人, 硕士在读, 主要从事通信与遥测技术方向的研究。

涉及的 API 函数为 ReadFile、WaitForSingleObject、GetOverlappedResult 和 ClearCommError。其中, ReadFile 函数用于读取串口, WaitForSingleObject 用于等待重叠 I/O 事件被激活, GetOverlappedResult 用于获取重叠 I/O 执行结果, ClearCommError 则用于清除各种串口错误同时返回串口状态。基本逻辑流程如图 1 所示。

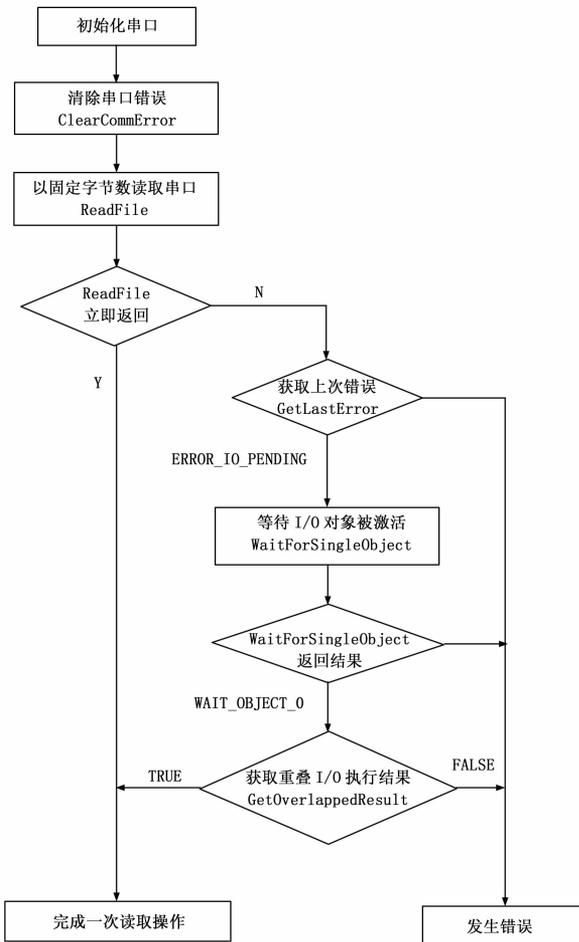


图 1 重叠 I/O 方式读取串口逻辑流程

采用重叠 I/O 方式读取串口缓冲区时, 在读取到的数据还未达到由用户设定的数量要求时, Wait ForSingleObject 将会一直等待, 若在子线程中则该线程会处于挂起状态。为防止陷入长期等待, WaitForSingleObject 允许设置超时返回。写串口操作与该逻辑流程基本一致, 这里不详述<sup>[4]</sup>。

### 3 多线程的设计模式

#### 3.1 多线程设计模型

一般而言, 在读取串口数据后, 需要对这些数据进行分包、解算、存储等操作。在高速串口上位机软件设计中, 若将读取串口数据和数据处理任务安排在一个子线程中执行, 当数据处理任务复杂繁琐、耗时可观时, 则极易出现读取串口不及时而导致数据丢失的现象。因此, 在高速串口应用软件设计中, 必须将读取串口数据和数据处理分别采用两个子线程实现。

如图 2 所示为一种可行的多线程设计模型。在该模型中, 在内存开辟了 A 片与 B 片两块缓存区, 读取串口数据子线程

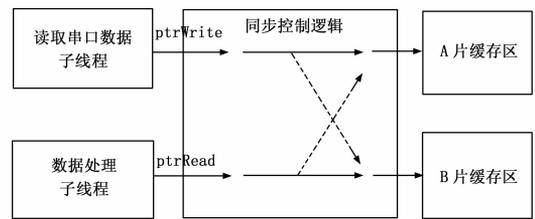


图 2 多线程设计模型

通过写指针 ptrWrite 将从串口读取的数据存入缓存区, 数据处理子线程则通过读指针 ptrRead 从缓存区读取串口数据进行处理。为保证数据的一致性, 读指针 ptrRead 与写指针 ptrWrite 不能同时指向同一片缓存区。同时, 在读指针 ptrRead 所指向的缓存区串口数据被处理完成后应交换读指针 ptrRead 与写指针 ptrWrite 所指向的缓存区。

在一些特殊应用场合如采用高速串口传输图像数据, 如若设计为当写指针 ptrWrite 指向的缓存区被写满后才交换读写指针, 则会出现图像数据流传输不均匀的缺陷, 在视觉上则表现为图像场景快慢不均。因此, 采用当数据处理子线程完成数据处理任务后立即交换读写指针的设计方案更具普适性。

若当 A 片与 B 片缓存区均设置为 5 MB, 在传输带宽被完全占用的条件下, 对于波特率为 4 M 的高速串口, 每秒产生 0.5 MB 的数据, 此时数据处理子线程读取缓存区的最大延时可达 10 s。

#### 3.2 同步控制逻辑

在图 2 中, 交换读指针 ptrRead 与写指针 ptrWrite 指向的缓存区涉及到两个子线程的相关变量, 因此必须考虑这些变量的一致性问题, 亦即线程的同步问题。Windows 为线程同步提供了关键区、互斥锁、信号量和事件对象等方式。

如图 3 所示为采用关键区和同步事件对象实现的同步控制逻辑流程, 其中 EXCHANGE\_ENABLE 为读写指针是否可交换标志。对于数据处理子线程, 该线程一直处于挂起状态直到同步事件对象被激活, 激活后利用 ptrRead 指针读取缓存区数据进行后续处理, 之后复位同步事件对象并置 EXCHANGE\_ENABLE 为 TRUE 状态。对于读取串口子线程, 首先检查

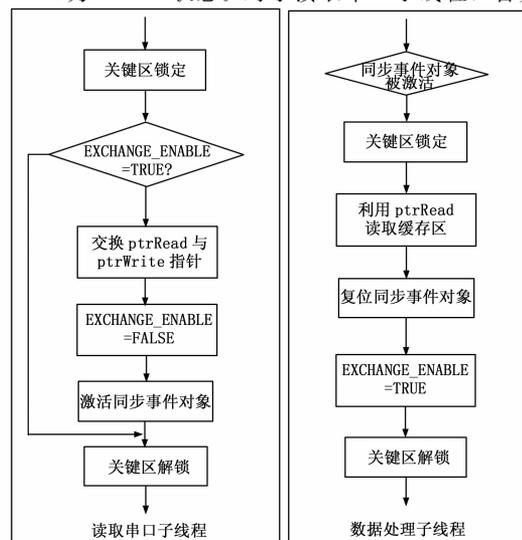


图 3 同步控制逻辑流程

EXCHANGE\_ENABLE 标志，若可交换则交换读写指针指向的缓存区，之后置 EXCHANGE\_ENABLE 为 FALSE 状态并激活同步事件对象。此外，为保证逻辑流程的数据一致性，两个线程中与同步控制相关的流程必须放在关键区中执行<sup>[5]</sup>。

#### 4 模式的编码实现与测试

##### 4.1 模式的编码实现

本模式以一个 MFC 规则动态链接库的方式实现，可以被其他应用程序链接调用，提供的外部接口函数如表 1 所示。

表 1 导出函数列表

导出函数	功能
CTRInitialCommSrv	初始化串口
CTRSetCommSettings	涉及串口参数
CTROpenComm	打开串口
CTRStartRdCommThrd	启动串口读取线程
CTRGetRdCommThrdState	获取串口读取线程状态
CTRStopRdCommThrd	停止串口读取线程
CTRCloseCommSrv	关闭串口
CTRDeleteSyncParams	删除读写同步参数
CTRGetCommData	获取串口数据
CTRWrComm	写串口(发送数据)

其中，CTRInitialCommSrv 用于初始化串口服务，设置该串口是否采用重叠 I/O 方式，并返回一个指向目的串口的句柄，CTRSetCommSettings 利用该句柄设置波特率、数据位、停止位、校验方式等参数，CTROpenComm 利用该句柄打开串口开始数据收发工作。CTRStartRdCommThrd 启动串口数据接收线程，CTRGetCommData 读取已接收到的数据。CTRWrComm 则通过串口向外发送数据。

##### 4.2 测试及效果评估

在测试环境中，下位机通过高速串口传输 JPEG 标准的红外图像压缩码流，图像尺寸为 640 \* 480，压缩比为 12.5，高速串口波特率为 2.5M，传输带宽几乎全部被占用。读取串口子线程读取图像压缩码流，数据处理子线程对压缩码流进行解压并显示图像，单幅图像解压耗时约为 40 ms。如图 4 所示为

解压出的红外灰度图像。

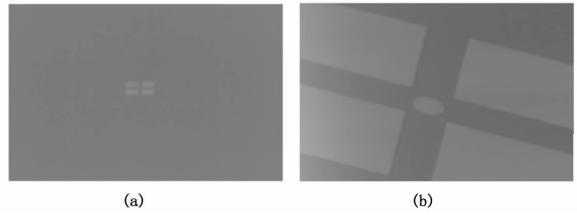


图 4 传输红外灰度图像压缩码流及解压测试

在测试过程中，在每秒解压图像达 12 帧的状态下，未出现数据丢失的情况，图像场景匀速。测试结果表明，提出的多线程设计模式解决了高速串口上位机软件设计中因串口缓冲区溢出导致的数据丢失问题，提出的同步控制逻辑解决了串口数据接收与耗时可观的后续数据处理任务之间的同步问题。

#### 5 结语

考虑到串口传输带宽的占用率、后续数据处理任务的耗时程度，串口的上位机软件设计方法应根据实际应用环境来选择。在低速串口和后续数据处理任务耗时较少的情况下，采用微软的 MSComm 控件能减少代码量、简化设计；对于后续数据处理任务耗时较多的情况下，无论高速串口还是低速串口，应采用多线程设计方式，既能提高数据处理的实时性，又能降低读取不及时而导致串口缓冲区溢出的风险。

##### 参考文献：

[1] 王中训, 徐超. 基于 VC++6.0 的多串口通信方法 [J]. 计算机应用, 2008, (6): 254-256.  
 [2] 赵晓辉, 陈艳萍, 张科英. 基于 VC++ 环境下串口通信的研究与实现 [J]. 电子设计工程, 2010, (1): 39-40.  
 [3] 石海杰, 常虹. 基于 VC 的多线程串口通信程序设计 [J]. PLC & FA, 2009, (9): 65-66.  
 [4] 申晓宇, 赵毅强. 多线程串口类在实时数据采集系统中的应用 [J]. 计算机时代, 2010, (1): 28-30.  
 [5] 杨旭东, 蔡敬坤. 一种通用串口线程在 C++ Builder 中的实现 [J]. 计算机测量与控制, 2011, (7): 1687-1689.

\*\*\*\*\*

(上接第 2317 页)

管理器、while 循环以及对硬件系统的初始化，可成功利用配置文件实现测试系统设置。软件构架采用 2.1 节的结构，其中测试处理层是整个系统的核心，它的实现是基于配置文件的形式。系统的人机交互界面，主要分为信息记录、参数显示、指令控制三部分，软件用户界面良好，有方便易学的人机对话窗口。

该软件缩短了设计人员的开发时间，实现了同型号类型的通用和不同型号之间的重用，具备高度的通用性，延长了软件的生命周期<sup>[5-8]</sup>。

#### 4 结束语

本文设计了一种基于配置文件的供配电测试软件的设计方法，实现复杂测试需求的灵活选配，解决了不同型号之间软件的通用问题，在某航天器的测试应用中取得了良好效果。该方法可为复杂的航天器供配电测试软件设计提供了一种通用架构。目前已在多个重要型号中应用，具有较高的应用价值。该方法将在航天器研制中发挥更大作用。

##### 参考文献：

[1] 王庆成. 航天器电测技术 [M]. 北京: 中国科学技术出版社, 2007.  
 [2] 王建军, 马楠, 李立. 卫星星座地面供配电测试系统研制 [J]. 航天器环境工程, 2010, 27 (3): 346-350.  
 [3] 赵琳, 翟正军, 周健. 基于配置文件的测试软件结构一体化设计方法 [J]. 测控技术, 2011, 30 (5): 94-98.  
 [4] 阎梅芝, 李立, 章雷, 等. 环境减灾-1A、1B 卫星供配电测试系统设计 [J]. 航天器工程, 2010, 19 (2): 121-127.  
 [5] 刘兆存, 范玮佳. 软件过程中可复用需求分析 [J]. 重庆理工大学学报, 2012, 26 (1): 53-60.  
 [6] 安幼林, 杨锁昌, 黄考利. 基于功能接口的网络化 ATS 软件平台研究 [J]. 计算机测量与控制, 2007, 15 (6): 810-812.  
 [7] 杨占龙, 陈航, 杨虎. 通用鱼雷模拟器适应性软件开发 [J]. 计算机工程, 2011, 37 (24): 290-292.  
 [8] 刘金宁, 孟晨, 崔少辉, 等. 基于配置文件的虚拟仪器测试软件设计与实现 [J]. 电力自动化设备, 2005, 25 (1): 50-53.