

虚拟串口软件程序结构以及接口

- 1 背景.....2
- 2 虚拟串口软件整体框架.....3
- 3 虚拟串口设备驱动程序（ Ethcomm.sys ）5
- 4 虚拟串口网络服务程序（ Service.sys ）7
 - 4.1 服务程序与虚拟串口驱动程序的接口7
 - 4.2 服务程序流程框.....8

1 背景

在现场总线普及之前，非常多的仪表或者设备都是通过 RS232 或者 RS485 通讯接口来和计算机进行通讯的，这些设备目前广泛存在于工业以及民用领域中，如何改造这些设备，以使其适应于现代的分布式集中管理体系呢？解决这一问题自然想到是使用“RS232/RS485< ->现场总线”的桥接设备，但是这仅仅只是考虑了设备硬件的接入，而设备原有的基于串口通讯的管理软件却肯定不能适用新的现场总线。如何将这些管理软件也平滑地移植到新的管理系统中呢？

这个问题的答案就是需要使用一个“软件桥接器”，如图 1 所示：

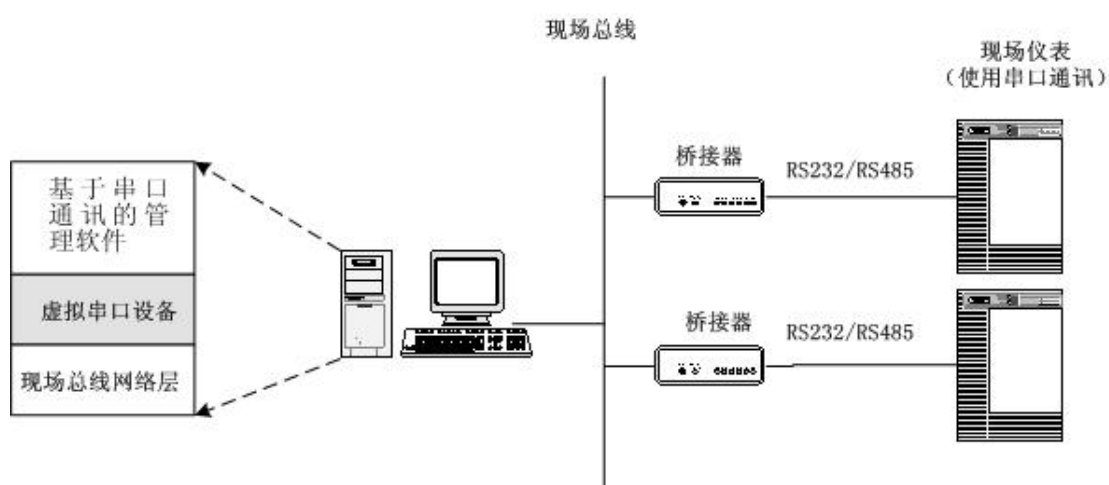


图1 基于串口通讯的仪表设备平滑移植到现场总线的方案

这个“桥接器”位于基于串口通讯的管理软件与现场总线网络驱动层之间，它实际上是一个“虚拟串口驱动程序”，负责管理软件和现场总线之间的信息转换工作，在作用上非常类似于硬件意义上的桥接器。

虚拟串口驱动程序向基于串口通讯的管理软件提供一个虚拟出来的串口设备（从 COM3~COM255 均可），管理软件对这个虚拟串口的操作和真实串口一模一样，从它的角度来看，这个虚拟串口和真实串口没有任何区别，虚拟串口获得管理软件的信息以后，将其转换成现场总线的网络帧，发往桥接器，桥接器再经过一次逆向转换，将网络帧转化为串口信息，发往现场设备。反之从现场设备发往管理软件的信息同样也需要经过这两次转换。

这样通过虚拟串口和桥接器的作用，管理软件和现场仪表都感知不到中间存在有现场总线。完全实现了原有的基于串口通讯的仪表设备平滑移植到现场总线的体系中来。

在下面几个章节中，我们采用以太网作为现场总线的例子，详细讨论了虚拟串口设备的

功能以及它的程序架构和实现方法。

2 虚拟串口软件整体框架

如图 2 所示。虚拟串口软件分为三个独立程序：

1. **虚拟串口驱动程序 (Ethcomm.sys)**：这是一个标准的 WDM 的设备驱动程序，它的作用是将用户发出的串口调用进行转换，然后传递给应用程序层的 Service 程序，并从 Service 程序收取以太网上的信息，将其转换后送给他串口应用程序。该驱动程序的系统 IO 调用接口以及读写控制完全仿造标准的串口驱动程序制作，所以从用户角度看来，虚拟串口的使用与普通串口没有任何区别。
2. **网络服务程序 (Service.exe)**：该程序是一个 WINDOW NT 的服务程序，它的任务主要是起到虚拟串口驱动程序和网络之间的联系。为什么要在应用层完成网络功能，而不直接在驱动程序层调用 TCP/IP 栈？主要是因为应用程序层对 TCP/IP 栈的调用较为安全和灵活，并且也易于调试，系统采取此种结构也会很稳定。而直接在内核调用 TCP/IP 栈，在某些流量比较极端的情况下，会有系统崩溃的隐患。
3. **安装程序(setup.exe)**：该程序负责为系统添加和删除虚拟串口设备，当系统中的虚拟串口设备被删光后，该程序还负责清除配置表和已安装的驱动程序。

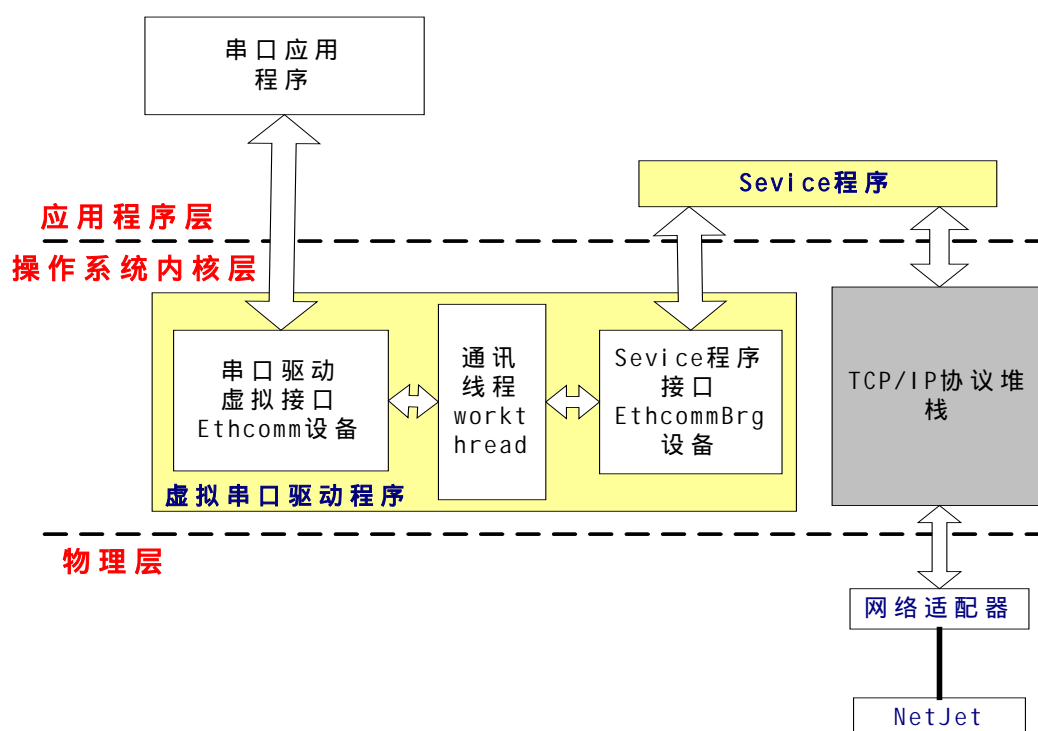


图2 虚拟串口设备的系统整体架构

本软件使用面对对象的方法进行设计，安装完成后，便可以添加任意多个虚拟串口，虚拟串口的最大个数只取决于系统资源是否充裕。

另外本软件安装完成后，桌面上没有任何应用程序运行，用户的使用虚拟串口的感觉和普通串口没有任何区别。

3 虚拟串口设备驱动程序 (Ethcomm.sys)

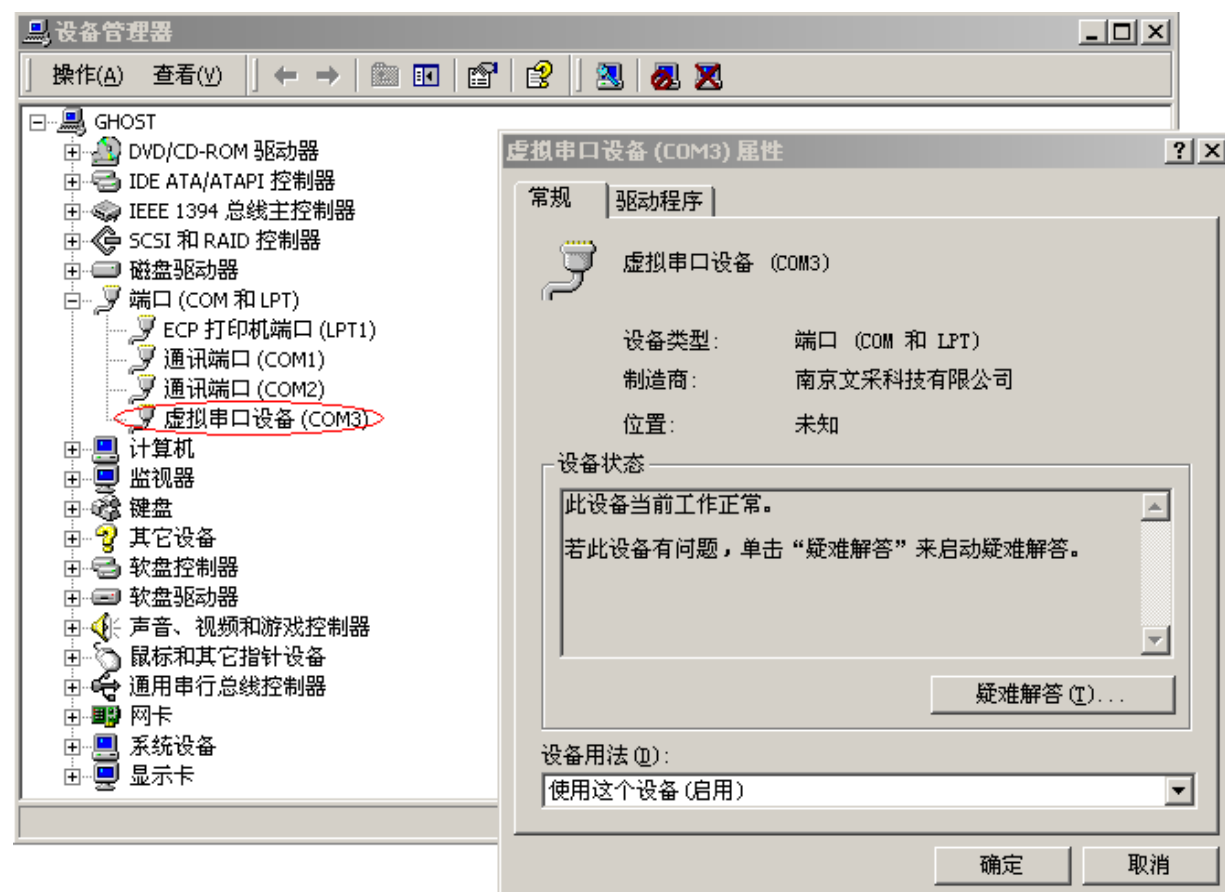


图3 虚拟串口设备驱动程序

本驱动程序是符合标准的 WDM 规范，安装后设备管理器中会出现“虚拟串口设备”这个条目。

如图 1 所示，本驱动程序由两个独立的设备构成：

一个设备称之为“Ethcomm”设备，它面向串口用户程序，为用户提供标准而完整的串口设备接口，使用户在使用虚拟串口时和普通串口没有任何区别。

另外一个设备称之为“EthcommBrdg”设备或者简称为 BRDG 设备，它面向服务程序，是 Ethcomm 设备与服务程序之间的桥梁。使驱动程序能够和 NetJet 顺利通讯。

为了保证驱动程序的稳定性和可靠性，防止系统死锁或阻塞，提高程序性能。两个设备之间的数据交换使用一个内核线程 workthread 来完成。当设备需要相互传递数据时，它们并不直接通讯，而是将数据缓存起来，扔给 workthread 进行传递。

Ethcomm 设备与串口用户程序的接口符合标准串口调用协议，这些接口在 Windows SDK 的帮助文件中均可看到。

BRDG 设备与服务程序之间的接口定义可以参看下一节的描述

4 虚拟串口网络服务程序（Service.sys）

本程序是一个 windows NT 的服务程序，安装后可在在控制面板/管理工具/服务 里看到该服务程序的条目。

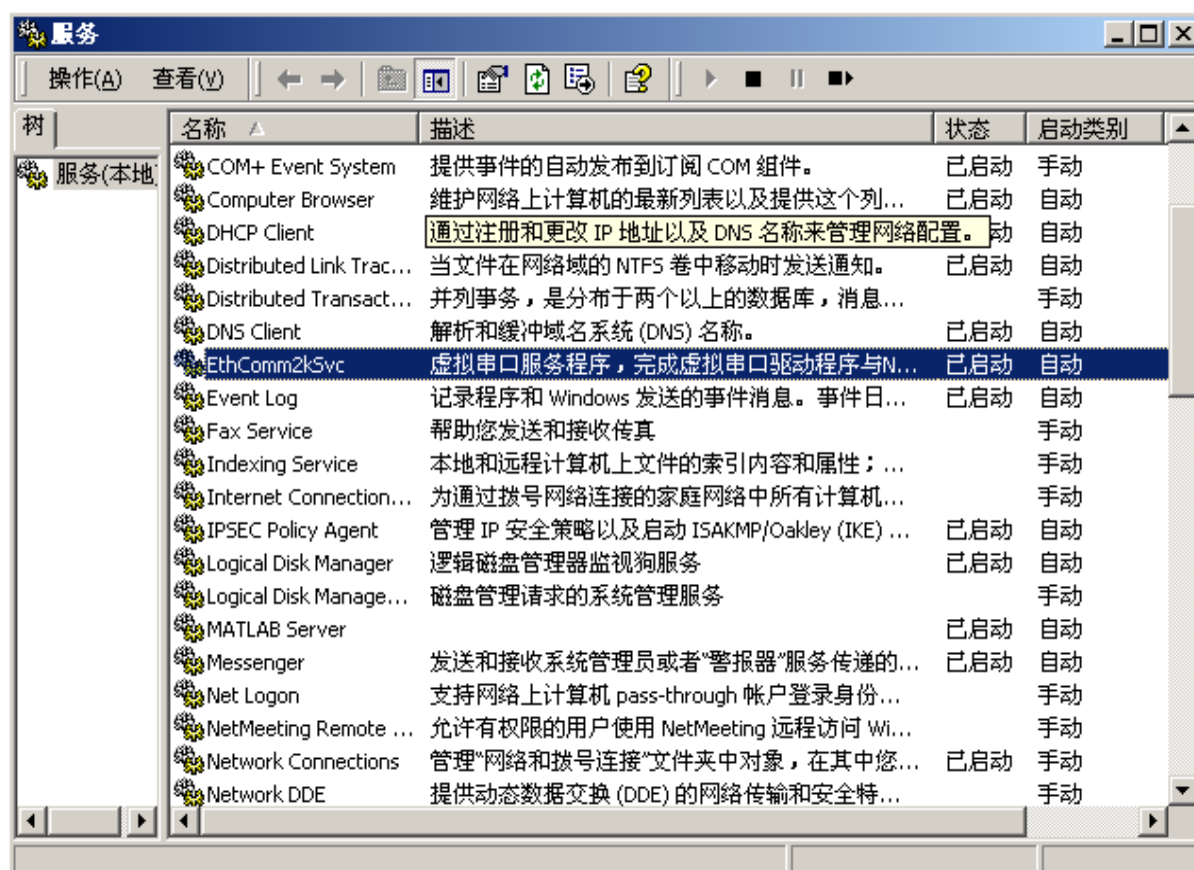


图4 网络服务程序

本服务程序主要功能是完成虚拟串口驱动程序与 NetJet 之间的通讯转换

4.1 服务程序与虚拟串口驱动程序的接口

服务程序从 Ethcomm.sys 驱动程序的 EthcommBridge 设备中获取用户对串口的指令和数据信息，并且将 NetJet 的数据和控制信号传给驱动程序

Ethcomm.sys 驱动程序的 EthcommBridge 设备名一般为“COMxBRDG”，其中 x 为串口号，例如 COM6 对应的 EthcommBridge 设备名就为 COM6BRDG

服务程序使用 IO control 命令（也就是说调用 DeviceIoControl 函数）来控制 EthcommBridge 设备，下面是这些控制接口

- 打开设备
- 检查驱动程序版本
- 获取用户的对串口的控制命令
- 向驱动程序报告虚拟串口的状态
- 获得串口波特率
- 获得串口位线控制
- ReadFile：获得用户发给串口的数据
- Writefile：向驱动程序写入从网上获得的 NetJet 数据

4.2 服务程序流程框

服务程序初始化部分流程如下，该部分程序初始化好服务线程的运行环境后就退居幕后，所有实际功能均由服务线程完成。在该初始化流程里，最重要的步骤是读取注册表配置信息，为每个虚拟串口创建一个服务对象（CEthCommSvc）实例，并启动对应的独立线程（每个虚拟串口对应一个）。

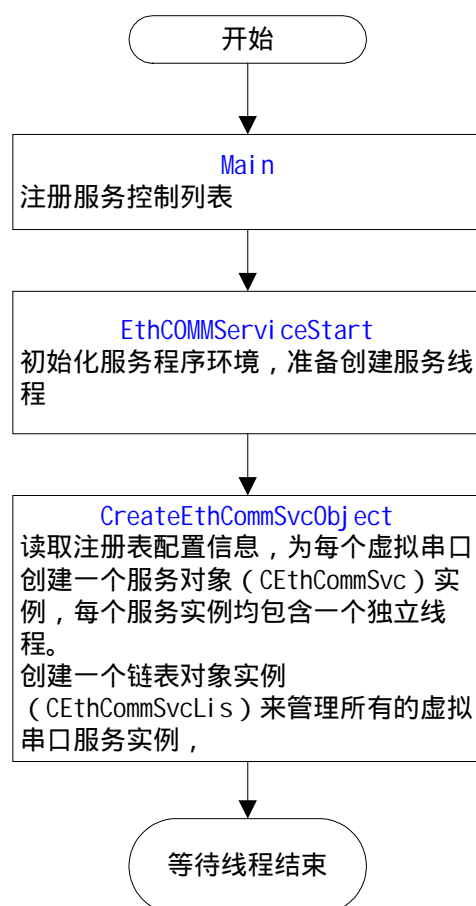
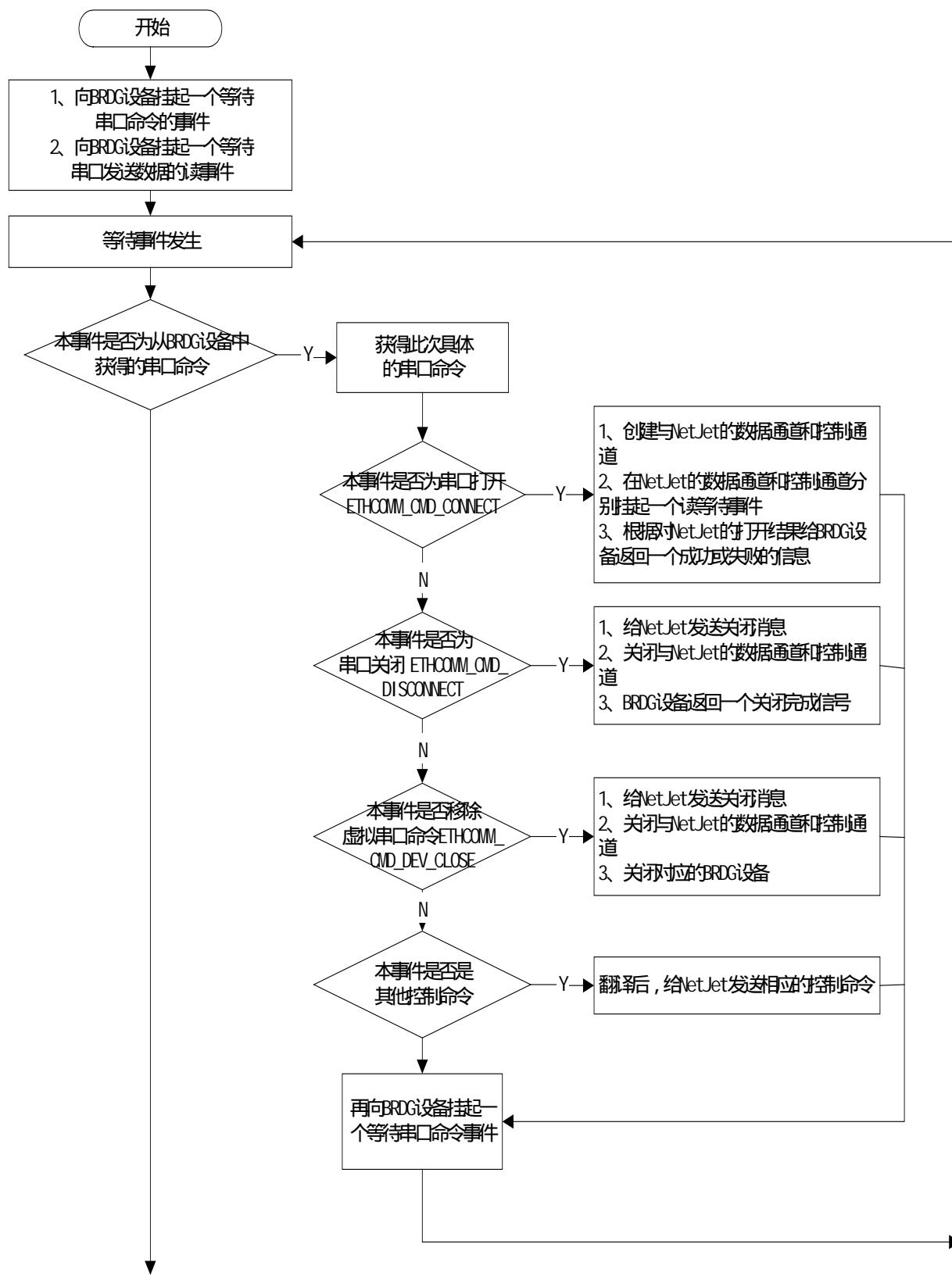


图5 服务程序的初始化流程图

服务程序中的线程是完成与 NetJet 以及驱动程序具体通讯工作的实体，该线程主体结构为事件驱动型，该线程注册了如下 5 个事件：

1. EVENT_BRDG_COMMAND：用户对虚拟串口有控制命令发生时，该事件产生
2. EVENT_BRDG_READ：用户对虚拟串口发送数据时，该事件产生
3. EVENT_DPORT_READ：NetJet 向服务程序发送数据时，该事件产生
4. EVENT_DPORT_WRITE：服务程序向 NetJet 发送数据完成时，该事件产生
5. EVENT_CPORT_READ：NetJet 向服务程序反馈虚拟串口状态时，该事件产生。



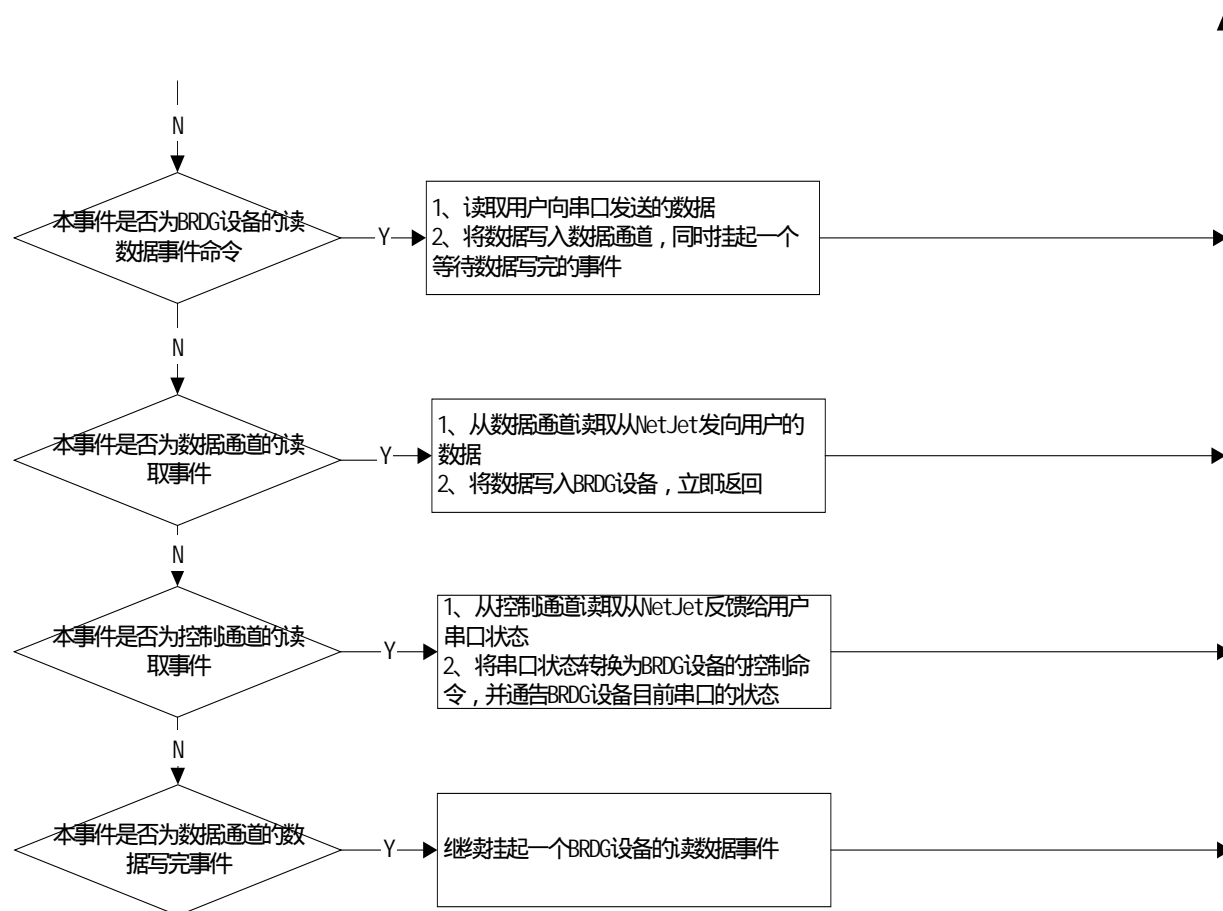


图6 服务线程流程框图