第32卷 第14期 计 算 机 工 程 2006年7月
Vol.32 № 14 Computer Engineering July 2006
·开发研究与设计技术· 文章编号：1000—3428(2006)14—0261—02 文献标识码：A 中图分类号：TN914

基于 Java 技术的纸币识别接收器通信控制实现

贺正方[1]，金 瓯[1,2]，贺建飚[1,2]，彭浩明[2]
(1. 中南大学信息科学与工程学院，长沙 410083；2. 湖南金融货币识别与自助服务平台工程技术研究中心，长沙 410004)

摘 要：论述了纸币器的通信控制原理及其通信协议，介绍了 Javax.comm 类库的组成和功能，结合 Java 串口通信技术实现了在 Windows 和 Linux 下对纸币器的跨平台通信控制。结合 Java 强大的网络功能，将更容易实现对设备的远程控制。
关键词：Java；纸币识别接收器；串口通信；通信控制

Realization of Communication Control for Bill Acceptor Based on Java Technology

HE Zhengfang[1], JIN Ou[1,2], HE Jianbiao[1,2], PENG Haoming[2]
(1. School of Information Science and Engineering, Central South University, Changsha 410083;
2. Hunan Engineering Center for Currency Recognition and Self-service, Changsha 410004)

【Abstract】 In this paper, communication control principle and protocol for bill acceptor are discussed. Constitutes and functions of Javax.comm API are introduced. With Java serial communication technology, communication control for bill acceptor is realized on different OS platform—Windows and Linux. Combining with Java's powerful network function, it is more easy to remotely control the equipments.
【Key words】 Java; Bill acceptor; Serial communication; Communication control

纸币识别接收器(简称纸币器)作为自助服务系统中实现现金交易的必要设备，其主要作用是识别和接收自助系统所指定国别的不同面额纸币，并将接收状态通过其串口发送给PC机(上位机)，实现自助系统的现金交易功能。以往主要是通过C或其他高级语言来实现纸币器控制，这种实现方式只能针对某一特定开发平台(比如 Windows 或 Linux)，系统可移植性不强，一旦出现跨平台的情况，控制程序需重新编写。而Java语言的平台无关性可较好解决控制程序移植难题；其面向对象的特征极大地缩短了开发周期，降低了开发难度，同时提供了良好的可扩展性。

本文根据课题的实际需要，给出了一种基于Java技术实现微机与纸币识别接收器之间的通信方案，并实现微机对纸币识别接收器的跨平台控制。

1 纸币识别接收器

1.1 概述

纸币识别接收器根据其功能可分为机械系统、电气系统、软件系统。而其控制部分主要由MCU、外扩存储模块、外部接口模块、AD/DA模块、通信模块等构成。其中MCU使用性价比高的 MCS-51 系列 8 位单片机作为控制器，片内含EEROM、256B RAM，多个计数器与中断源；外扩存储模块包括RAM、闪存，可为程序运算提供足够的存储空间，同时可存储纸币特征数据库以及记录设备工作状态等；通信模块负责接收和应答上位机的各种指令，接口采用RS-232标准。

1.2 通信协议

纸币器采用RS-232串口与主机进行通信，主机可发送复位、使能、压钞、退钞等控制命令，也可查询纸币器所处的状态。采用异步通信方式，串口的波特率选用9 600bps，每个字符的数据帧格式定义为：8 个数据位，无奇偶校验，1个数据停止位。

根据需求，自定义的信息帧格式有以下两种：

(1)命令帧、应答帧一般格式

如表1所示，一般命令帧和应答帧长度为4个字节：其中第1个字节为帧头，固定为10H，表示帧发送或接收开始；第2个字节表示帧长度，其值为从帧头到校验码的字节个数，取值可以为0H-FFH；第3个字节代表上位机发送的控制指令或者下位机应答的状态码，表示本帧的信息含义；最后一个字节表示校验和。

表1 命令帧、应答帧一般格式

帧头	帧长度	命令或应答的信息	校验和
10H	1字节	1字节	1字节

(2)特殊应答帧格式

如表2所示，特殊应答帧特指设备暂存纸币后回送给上位机的指令。与一般格式信息帧不同的是，该帧共有5个字节，除第4字节为纸币识别接收器收到的钱币面额外，其他部分与一般的帧相同。

表2 特殊应答帧

帧头	帧长度	应答信息	钱币面额	校验和
10H	1字节	1字节	1字节	1字节

2 Java 串口通信技术

Java语言本身并不能直接通过串行通信编程来操纵硬件设备，它必须借助于本地方法 (Java Native Interface，JNI)技术。JNI技术是Java平台的强大特征之一，它可以使在Java

基金项目：国家“863”计划基金资助项目(2003AA1Z2190)；国家“十五”科技攻关计划基金资助项目(2003BA104C)
作者简介：贺正方(1979—)，男，硕士生，主研方向：网络，数据库与信息处理；金 瓯，教授、国务院有突出贡献中青年专家；贺建飚，副教授；彭浩明，高工
收稿日期：2005-10-30 **E-mail**：he_zf@126.com

虚拟机内部运行的 Java 代码能够与用其他编程语言(例如 C、C++和汇编语言)编写的应用程序和库进行互操作。由 SUN 公司推出的 Javax.comm 类库已经做好了这部分工作，只需调用 Javax.comm API，就可以编写出 Java 串行通信程序。

Javax.comm 类库提供的接口包括：CommDriver 接口表示可装载的设备驱动接口，但它不可以由应用级程序运行。CommPortOwnershipListener 接口产生各种通信口所拥有的事件。打开一个端口时，PORT_OWNED 事件将被传递；关闭一个端口时，PORT_UNOWNED 事件将被传递。Serial PortEventListener 接口产生串行口事件；ParallelPortEvent Listener 接口产生并行口事件。

Javax.comm 类库主要提供了 6 个类，其主要功能如下：

•CommPort Identifier 和 CommPort 属于上层类，用于管理对通信端口的拥有和存取权限。

•类 CommPortIdentifier 用于通信端口管理。该类是控制对通信端口访问的核心类，提供的功能有：(1)在驱动程序帮助下发现有效的通信端口；(2)为输入输出操作打开通信端口；(3)决定端口所有权；(4)处理端口所有权的竞争问题；(5)管理表示端口所有权状态改变的事件。

•类 CommPort 表示通信端口，它是一个描述底层系统定义有效通信端口的抽象类，它提供了控制不同种类型通信端口输入输出的公共方法。SerialPort 和 ParallelPort 都是它的子类。它的方法用于基本通信端口的低级控制：getPortIdentifier()方法来产生一个有效端口列表；从列表中选择一个端口并调用 open()方法创建一个 CommPort 对象；最后，把 CommPort 对象投射到一个像 SerialPort 或 ParallelPort 的物理通信设备类上。当一个通信端口已被标识并打开，就可在低级类 SerialPort 和 ParallelPort 中调用相应方法，打开输入输出流读写数据。端口不使用时，调用 close()方法关闭，否则别的应用程序不能使用该端口。如果使用，会抛出一个 IllegalState Exception 异常。

•类 SerialPort 和 ParallelPort 提供对物理通信端口的操作接口，属于底层类。类 SerialPort 表示 RS-232 串行通信端口，对作用于底层系统的串行通信端口的低级接口，定义了串行通口所需的基本功能的方法。

•类 SerialPortEvent 表示串口事件，类 ParallelPortEvent 表示并口事件。它们属于驱动层类，提供底层类与操作系统之间的接口，不直接提供给程序员使用。

另外，还有 3 个处理异常的类：若驱动器找不到指定端口抛出 NoSuchPortException 异常；若驱动器不承认指定端口时抛出 UnsupportedCommOperationException 异常；若指定端口正被使用，抛出 PortInUseException 异常。

3 纸币识别接收器通信控制实现

3.1 控制软件设计

根据设计需求，本软件基于 Java 平台，可以实现纸币器对纸币的识别和接收，并可将相应数据传送给 PC 机。纸币器控制的实质就是：判断状态，并依据不同状态发送不同指令，纸币器依据指令产生相应的动作，从而实现对其有效的通信控制。

如图 1 所示，纸币器主要有禁用、使能、接收、暂存、压钞、退钞、售卖等 7 种状态。纸币器上电或接收复位指令后，处于禁用状态，此时不能接收任何纸币。当接收到使能指令后，纸币器变为使能状态，即处于准备接收纸币的工作状态。当用户从入钞口输入纸币时，传感器感应后产生的电信号驱动电机，纸币器吞入并识别纸币，并将此时状态改为接收态。如果识别纸币为假，则变为退钞态并退出假币，纸币器恢复为使能态；如果识别为真币，则进入暂存态，若接着发送压钞指令则将纸币压入钞箱，纸币器状态变为售卖。只有当纸币器接收到应答指令，才将其状态恢复为禁用，否则不改变。随后可开始新一轮接收纸币的过程。依据以上状态转换过程，可得到如图 2 所示的控制流程。

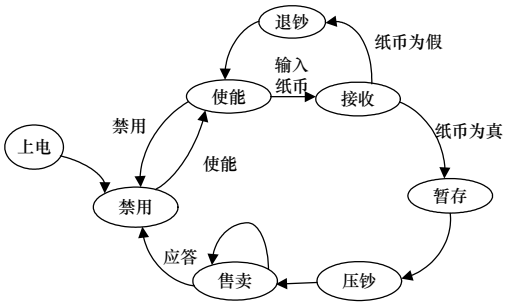

图 1 命令状态转换

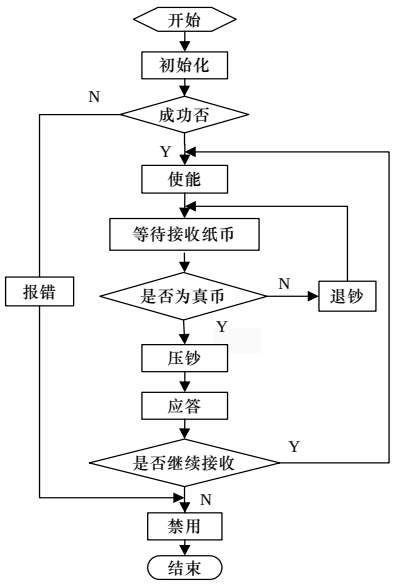

图 2 控制流程

3.2 控制软件实现

控制软件主要包括两个模块：串口通信模块和纸币器控制模块。根据需要设计了 SerialBean 和 zbjControl 两个核心类，分别用来处理串口通信和纸币器控制。

(1)类 SerialBean 通过对一些 Javax.comm API 的封装，实现了对串口操作细节的隐藏，提供对串口基本操作的方法。其主要的类方法有串口的打开 Open()、通信参数设定 setProperties()、关闭 Close()和写串口 Write()。程序在生成一个 SerialBean 的实例后，通过调用其 Open 方法打开参数中指定的端口。打开所选定串口失败时会抛出 PortInUseException 异常，设定通信参数失败时会抛出 UnsupportedComm OperationException 异常，设定端口事件监听器失败时 TooManyListenersException 异常。代码示例如下：

```
public class SerialBean implements SerialPortEvent Listener
{…
//打开串口的 OPEN 方法
public void open(String Port_name) throws Exception{
//将 String 型端口号转换成 CommPortIdentifier 类数据
portId = CommPortIdentifier.getPortIdentifier(Port_name);
//打开所选定的串口
sPort = (SerialPort)portId.open("SerialBean ", 30000);
//设定端口事件监听器
sPort.addEventListener(this);
//设定基本通信参数
```

(下转第 265 页)