

基于 Windows CE 的无线通信系统的设计与实现

Design and Implementation of Wireless Communication System Based on Windows CE

朱金秀 吴 淑 王 恒 顾春静 管薇薇

(河海大学计算机及信息工程学院(常州)江苏常州 213022)

摘 要: 针对港口生产现场实时数据传输的需求,文章介绍了基于 Windows CE 操作系统的手持终端的通信系统设计与实现。本文在研究 Windows CE 系统特征的基础上,设计了基于 Windows CE 的无线通信流程,重点阐述了服务器和客户端等关键模块的实现方法。该系统实际运用在江阴苏南港口,很好地适应了港口的业务需求。

关键词: Windows CE Java 手持终端 通信模式 通信协议

1 引言

根据江阴苏南港集装箱码头的发展形势、理货业务的发展需求以及码头集装箱吞吐量的不断增长的情况,通过对苏南港口的业务流程进行深入调查,利用 GPRS 无线数据网络进行数据实时采集,结合码头的应用现状和今后的业务发展的需求,进行了本次系统方案设计。

本文实现一个基于 Windows CE 操作系统的手持终端的通信,该系统是在 JDK1.3 环境下使用 JAVA 语言实现,采用 C/S(客户机/服务器)模式,通过公网将数据传输到监控终端,从而改善了原用户端复杂的手工操作状况,使数据能得到实时采集。

2 系统总体方案设计

2.1 系统方案论证

2.1.1 Windows CE 系统特性

Microsoft Windows CE 是一个多平台、多任务、多线程、可裁减的 32 位嵌入式实时性的操作系统,在内存管理方面要比其它 Windows 操作系统更节约物理内存和虚拟地址空间,并且它具有强大的通信能力和出色的用户界面。

鉴于 Windows CE 操作系统的优越特性,并且符合系统设计要求,在设计中也需要充分利用这些特性,所以选择在此操作系统中实现本次设计。

2.1.2 Java 语言在嵌入式系统 Windows CE 中的应用

Windows CE 操作系统不能直接支持 Java 语言,需要借助于一个 Java 虚拟机,这里采用的是一个可靠的、可裁剪的 J9 虚拟机。该虚拟机提供了对 Java 本地方法(JNI)的支持和可靠的递增式垃圾回收机制。该虚拟机支持扩展的 Java 实时标准和 TCP/IP 协议,并且支持动态类库加载。更为重要的是,该虚拟机能够从文件加载事先编译好的软件与程序,从而减少了应用程序的启动时间,提高了整个应用程序的速度。因此,可以成功地将 Java 编写的程序打包后移植到 Windows CE 操作系下运行。

对于 Java 语言的编程环境而言,JDK1.3 与所有操作系统对接时,所有接口都提供相同的信息,能与 WinCE 兼容,并且在功能实现中也能达到设计要求,最主要的是港口采用的得逻辑手持终端所支持的 J9 虚拟机是建立在 JDK1.3 基础之上的,因此选择使用 JDK1.3 编程环境。

2.2 系统总体实现方案

本设计中,各监控点使用 GPRS 透明数据传输终端,通过移动 GPRS 网络与监控中心相连,并采用的是 APN 接入点方式。这里采用 GPRS 网络主要是由于它具有覆盖范围广、数据传输率高、传输容量大,并且 GPRS 技术适用于连续的、突发性的和频繁的数据传输,能满足手持移动终端众多时系统对突发性数据传

输进行处理的要求。各数据采集点使用移动通信公司统一的 STK 卡 ,同时监控中心对各点进行登记 ,保存相关资料以便识别和维护处理。各信息采集点运行监控系统软件 ,支持 24 小时实时在线 ,实现信息点 24 小时传送采集的信息数据。

APN 具有很好的安全性能。基于源和目的 IP 地址 ,通过互联网建立 IPSec 加密隧道 ,专用 APN 由移动公司配置 ,监控终端的 SIM 卡号和 APN 绑定 ,用于 APN 专网的 SIM 卡仅开通该专用 APN ,限制使用其他 APN ,普通用户不得申请该 APN ,也进入不了专用 APN。因此 ,它兼备公网的便捷和专用网络的安全 ,即使系统通信的实时性和安全性得到有效保障。

系统总体实现方案如图 1 :

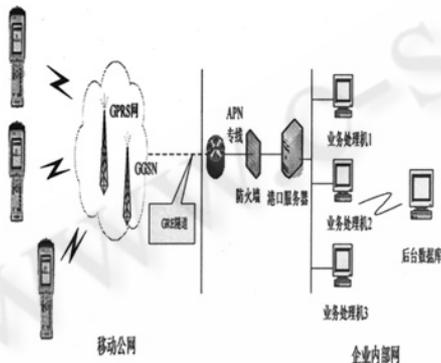


图 1 无线通信系统总体实现方案

根据表中内容可以看出 ,C/S 模式更加符合系统要求 ,因此 ,本设计选择使用 C/S 模式。

3.2 通信协议格式

Windows CE 操作系统支持 TCP/IP 协议 ,则将消息数据封装成 IP 数据报格式发出。为了实现从数据流中方便提取数据、保证数据安全可靠地传输 ,自定义了数据通信格式。自定义数据格式是以“ # ”号开头和结尾的 ,并且在消息主体中每个单项的信息之间是用“ * ”号隔开。按照自定义数据通信格式 ,可以获得发送端 IP 地址 ,以确保服务器端能正确地把终端请求的响应信息发送到相应的终端 ;另外 ,还能从消息头中得知所安排的任务和具体信息 ,保证任务正确操作 ,实现了所需终端与服务器端的双向通信功能。具体通信协议格式如图 2 所示。



图 2 通信协议的格式

3 系统的通信流程设计

3.1 通信模式论证

目前 ,有两种常见的通信模式 :一种是客户机/服务器(C/S)模式 ;另一种是浏览器和服务(B/S)模式。

针对本系统设计而言 ,对安全可靠、实时性要求高 ,并且使用 GPRS 网络通信 ,网络流量要尽可能小 ,以减少通信费用。为此 ,对 C/S 模式和 B/S 模式作了比较 ,结果如下表 1 :

表 1 C/S 模式和 B/S 模式比较

技术指标	C/S 模式	B/S 模式
交互性	强	较弱
安全性	高	低
网络流量	小	大
通信速度	快	慢

3.3 通信过程流程分析

如图 1 所示 ,系统采用的是客户/服务器通信模式(C/S) ,服务器端利用 ServerSocket 创建侦听特定端口的 Socket ,Socket 用于创建手持终端和服务端之间的通信 Socket。为了实现通信 ,服务器端要开启侦听服务、发送和接收服务。具体如图 3 ,演示了服务器与手持终端利用 ServerSocket 和 Socket 来通信的过程。

实际应用中会有多个手持终端对服务器进行访问 ,因此要考虑到避免冲突问题。本设计采用的是线程之间通过执行 sleep()方法 ,让线程处于睡眠状态 ,这样就能保证循环接收消息 ,以解决线程之间的冲突问题。

4 系统具体实现

本系统程序实现过程中 ,服务器端通过 ListenImpl()类实现服务器侦听服务 ,SendImpl()和 ReceiveImpl()

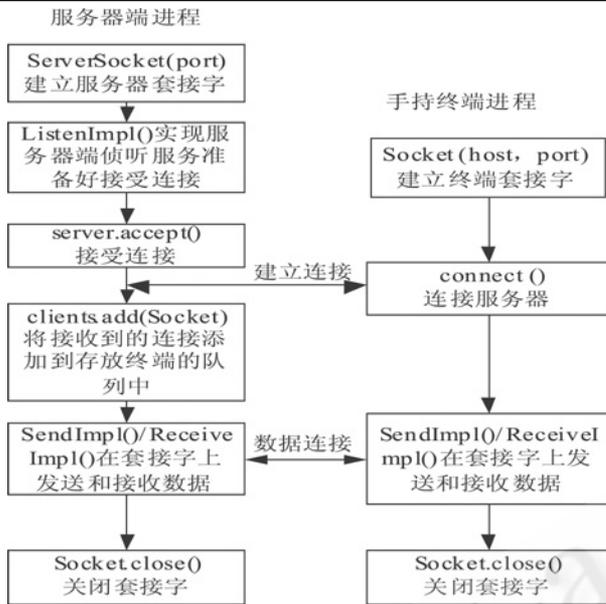


图 3 服务器与手持终端通信

()类实现数据的发送和接收。手持终端同样要用 SendImpl()和 ReceiveImpl()类实现数据发送和接收。

4.1 服务器端的程序实现

服务器端程序流程图如图 4 所示。服务器可以接受多个合法手持终端的连接请求,与手持终端数据传递采用字符串形式,按固定格式传输。



程序实现关键代码

如下：
服务器的初始化程序为：

```
private void initServer( ){
try{
receiveMessages = new
Vector( );
//创建存放接收消息数组
对象
sendMessages = new Vec-
tor( );
//创建存放发送
```

图 4 服务器程序实现流程图

消息数组对象

```
sender = new SendImpl( ); //生成 SendImpl 类的对象
receiver = new ReceiveImpl( );
//生成 ReceiveImpl 类的对象
```

```
listener = new ListenImpl( );
clients = new Vector( );
serverSocket = new ServerSocket( 8000 );
//生成服务器端口
listener.setListenSocket( clients );
//实现对客户端的侦听
listener.setServerSocket( serverSocket ) /
/设置服务器套接字
sender.setSockets( clients );
receiver.setSockets( clients );
sender.setMessages( sendMessages );
receiver.setMessages( receiveMessages );
//设置消息对象
}catch( BindException b ){
System.out.println( " Server 已经启动 " );
System.exit( 0 ); //结束程序
}
```

4.2 客户端的程序实现

客户端的程序与服务器的程序实现基本类似,都要包括以上流程图中介绍的功能,唯一不同的是需要增加一个与服务器的连接过程,也就需要构造一个连接方法。因此阐述服务器和客户端的连接时的关键代码如下：

```
private boolean connect( ){
try {
System.out.println( " 连接到服务器 " );
connect = new Socket( " 服务器 IP 地址 " , 8000 );
//建立 Socket 连接
String
address = InetAddress.getLocalHost( ).toString( );
//用于获取地址
int port = connect.getPort( ); //用于获取端口号
int ii = address.indexOf( "/" );
adrthis = address.substring( ii + 1 );
System.out.println( " 此终端的地址为 : " + adrthis );
System.out.println( " 此终端的端口为 : " + port );
in = connect.getInputStream( ); //输入数据
out = connect.getOutputStream( ); //输出数据
return true ;
```

```

} catch( Exception e ) { //抛出异常
    e.printStackTrace( );
System.out.println(“ 不能连接到服务器 ”);
    return false ;
}
}

```

通信连接建立了以后就可以对其线程进行处理,当线程工作完成时,则可以关闭服务器,即 serversocket.close()。

4.3 系统总体实现效果

下面列举了两个系统在运行过程中所获得的用户图形界面,以表明设计所得到的效果。



图5 用户登录界面 图6 理货人员卸船作业界面

图5是理货人员装卸船登录界面,只有输入信息与数据库中的匹配才能成功登录,保证了安全性。图6是接收到作业信息的界面,在此界面中可将装卸的箱信息反馈给数据库,保证数据的实时采集。

C/S模式只需传输输入数据,以图5为例,即只需传输员工登录时需要输入以对其身份的认证数据,而B/S结构则要传输实现这个界面所有内容的信息量,所以通过两种模式传输该界面时,测试结果如表2。

表2 测试结果对照表

输入信息 (字节)	传输模式	通信流量 (字节)	传输时间 (ms)
16	C/S	74	0.64
	B/S	973	8.46
32	C/S	90	0.78
	B/S	989	8.6
64	C/S	122	1.06
	B/S	1021	8.88

从数据可知,完成同样的任务,C/S所需要传输的字节数远少于B/S模式,即C/S模式网络流量小、传输速率快。因此,C/S模式能够保证系统的安全可靠性和实时性,达到了港口需求,并且流量小能为企业节省网络通信费用。

5 结束语

以上介绍无线通信系统的设计与实现,主要是在手持通信设备上的应用。此程序设计的手持通信设备已经应用到了江阴苏南港集装箱码头,并且使得码头工作人员的工作效率得到了很大的提高,也使集装箱信息能实时得到采集。随着现在各个码头集装箱码头吞吐量不断增大,这种手持通信系统将会在其他码头得到推广。

参考文献

- 周晓光、周晓雄. 多功能物流手持终端集群通信方式的设计与实现[J]. 电子技术与应用, 2007, (2): 98-99, 106.
- 杨光、张子凡、王瑞慧、欧阳俊. 基于GPRS的APN网络通讯技术在环境监控系统中的应用[J]. 环境监测管理与技术, 2006, 18(2): 8-10.
- 刘峰、韩春燕、林许. 基于嵌入式TCPIP的远程GPRS控制终端的设计与实现[J]. 小型微型计算机系统, 2006, 26(6): 1069-1071.
- Chris Muench. Windows CE权威指南[M]. 精英科技, 北京: 中国电力出版社, 2001.
- Shang-Liang Chen, Ting-Hao Lin. Windows CE-based Intelligent Remote Monitoring System[M]. Institute of Manufacturing Engineering, National Cheng-Kung University, Taiwan. 2007.
- BOLING D. Programming Microsoft Windows CE. NET, Third Edition, Microsoft Press, 2003.
- 孙卫琴. Java面向对象编程[M]. 北京: 电子工业出版社, 2006. 7.
- 孙卫琴. Java网络编程精解[M]. 北京: 电子工业出版社, 2007. 3.