

远程监控系统中基于 J2ME 手机软件的设计与实现

孙 媛, 石江宏

(厦门大学通信工程系, 厦门 361005)

摘要: 简要介绍了手机软件 J2ME 开发平台和开发工具, 分析了手机远程监控的系统结构和功能实现, 着重介绍了远程监控系统中基于 J2ME 手机客户端软件的编写, 包括用户界面设计、网络连接、云台控制、图片显示几个部分的设计与实现, 并给出了部分程序流程图及运行演示。

关键词: J2ME; 远程监控系统; 手机监控

Design of handset software based on the J2ME in long-distance monitor system and its implementation

SUN Yuan, SHI Jiang-hong

(Communication Engineering Department of Xiamen University, Xiamen 361005, China)

Abstract: This article briefly introduces the J2ME development platform and development kit for handset software, analyzes the structure and the function of handset long-distance monitor system, emphatically introduces the handset client software programming based on J2ME in long-distance monitor system, including the design and realizing of user interface, the network connection, camera control, and the picture displaying. This article also presents some flow chart diagrams and shows the result of program.

Key words: J2ME; long-distance monitor system; handset monitor

0 引言

计算机网络、多媒体和数据通信技术的高速发展,使得远程监控的实现成为可能。目前,以嵌入式技术为依托,以网络、通信技术为平台的远程监控系统已经越来越成熟,延伸到政府、酒店、城市交通、医院、工厂、城市治安等各个领域,远程监控技术已经成为监控研究领域中的热点。

目前有三种远程监控系统:基于 WEB 的远程监控;基于嵌入式 Internet 的远程监控;基于无线网络的手机移动监控。随着移动通信向宽带化、数据化、多应用化发展,移动通信和网络互联技术融合在一起,形成移动互联网。在移动互连网技术的推动下,手机互连网和多媒体业务也逐渐普及,手机已经不再只是传统意义上单纯的通信工具。手机应用程序的开发使手机发挥作用的领域扩展到人们生活的各个领域,带来了巨大的商业价值。手机移动监控技术作为手机服务的一个新领域,在现在以至未来一

段时间里都将是远程监控技术中的研究热点。

本文就手机在远程监控中的应用提出一种设计方法和实现。首先介绍了软件开发使用的 J2ME 平台相关知识,在概述监控系统总体实现流程的基础上,主要说明了手机客户端软件部分的设计和实现。

1 手机软件 J2ME 开发平台与开发工具

1.1 J2ME 开发平台

J2ME: Sun Microsystems 将 J2ME 定义为:“一种以广泛的消费性产品为目标的高度优化的 Java 运行环境,包括寻呼机,移动电话,可视电话,数字机顶盒和汽车导航系统”。

使用 J2ME 平台开发手机软件有以下优点:

平台的兼容性,通过虚拟机屏蔽了各种设备的差异性,支持手机种类多;本身带有的安全性设计确

收稿日期: 2007-04-28

作者简介: 孙媛(1982-),女,硕士研究生,主要研究方向为无线通信及远程监控系统中手机客户端软件的开发研究。

保第三方应用程序的可靠性;支持应用程序动态下载和升级;提供丰富实用的类库和方法,大大提高了程序开发速度。

1.2 J2ME 开发工具

(1) J2ME Wireless Toolkit

J2ME Wireless Toolkit 是 Sun 公司官方发布的无线开发工具包。这一工具包的设计目的是为了帮助开发人员简化 J2ME 的开发过程。该工具箱包含了完整的生成工具、实用程序以及设备仿真器。

(2) J2ME 集成开发工具

目前进行 J2ME 程序开发主要有以下几种集成开发工具: Netbeans IDE+ Mobility Pack, JBuilder+ MobileSet, Sun One Studio, Eclipse+ EclipseME。本文软件设计采用 NetBeans 的 J2ME 集成开发工具。

NetBeans 是由 Sun 发起的一套完全以 Java 撰写而成,并且开放源代码的免费开发平台。NetBeans 安装、使用都非常简单。可以在 NetBeans 网站上下载最新版本的 NetBeans5. 5, 还有提供 JDK 捆绑安装的版本。下载到默认安装 IDE 环境时, NetBeans 已支持 J2EE、J2SE 的开发, 要支持 J2ME 开发, 则需要安装 Mobility Pack 插件, 这个插件功能非常强大, 可以方便管理项目、打包和签名、混淆、部署。Netbeans IDE+ Mobility Pack 还可以实现可视化应用流程开发。

2 客户端软件的设计与实现

2.1 远程监视系统总体结构

远程监控系统总体框图如图 1 所示。远程监控系统实现的基本流程: 首先客户端通过 GPRS 网络连接上服务器, 服务器控制摄像头采集监视现场图像数据, 然后取回摄像头采集的图像数据, 编码成 JPEG 图像, 并存储在服务器上。在此期间, 服务器控制定时更新存储的图片, 直到客户端断开连接。手机与服务器建立连接后, 定时重新读取服务器上存储的 JPEG 图片, 同时手机还可以通过向服务器发送控制命令, 由服务器控制云台摄像机的转动, 光圈、焦距调节。

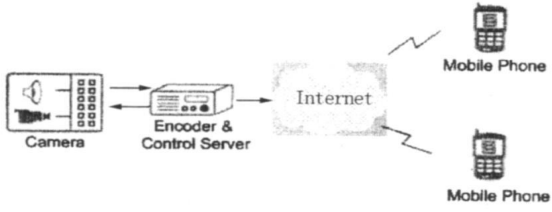


图 1 远程监控系统总体框图

摄像头主要完成图像数据的采集功能。服务器主要完成控制摄像头、对图像数据进行 JPEG 编码并存储, 并且与用户交互的功能。手机作为监控系统的控制终端, 完成连接服务器, 通过服务器控制摄像头的移动, 以及读取服务器上的监视图片并显示的功能。下面就手机上的客户端软件的设计与实现做详细说明。

2.2 软件总体流程图

本文提出的客户端软件的设计主要包括用户界面设计、网络连接部分、控制部分以及图片显示等部分。用户界面设计使用 MIDlet 类, 主要包括登陆界面、等待连接界面、错误提示函数、退出函数; 网络连接部分主要是创建 HttpURLConnection 网络连接线程, 包括登入连接、控制信息的发送、错误处理; 控制部分主要是利用相应的服务器接口参数以及对手机按键行为的处理产生控制命令。图片显示部分使用了定时器任务函数, 完成定时读取图片, 软件总体流程图如图 2 所示。

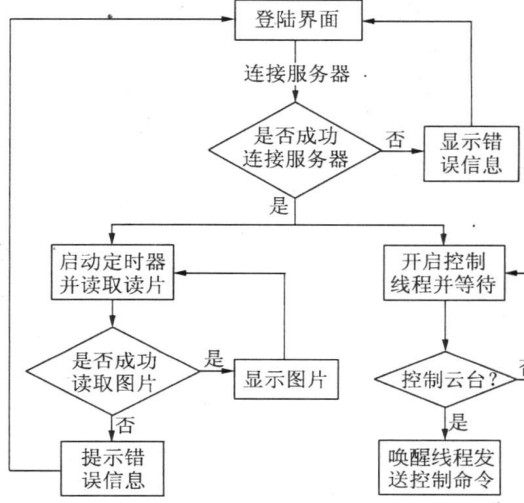


图 2 软件总体流程图

2.3 MIDlet 类以及用户界面设计

J2ME 的 MIDP (Mobile Information Device Profile) 包含的系统构架和 Java 类库, 为小型、资源受限的移动信息设备 (MID) 创建了一个开放的应用程序环境。

MIDP 应用程序中的基本执行单元叫做“MIDlet”, 它必须继承自 javax. microedition. midlet. MIDlet 类并实现 javax. . microedition. midlet. MIDlet 类中包含的三个抽象方法: startApp(), pauseApp(), destroyApp()。在 J2ME 手机软件开发中, 程序必须至少包含一个 MIDlet 类, 通常初始化用户界面的程序应该放在 startApp() 中执行。

MIDP 应用程序用户界面 API 包含于 javax. microedition. lcdui 包中, 分为高级用户界面和低级用户界面。高级用户界面由类 Screen 实现, 低级用户界面类由类 Canvas 和 Graphics 实现。高级用户界面类包括 Alert, List, TextBox 以及 Form 类。其中 Form 类缺省没有任何用户界面的组件, 可以通过添加 Item 类的子类向 Form 中添加组件, 展示各种不同类型组成的用户界面, 这些类型包括本文、图像、可编辑的数据域和文本域等等。可以使用通过在 Form 类中添加 TextField 及 Gauge 类来实现登陆界面、等待连接界面。

2.4 网络连接及线程 HttpConn 部分

J2ME 的连接及网络 API 位于包 javax. microedition. io 中。这些 API 接口包括 HttpURLConnection, SocketConnection, ServerSocketConnection, HttpsConnection 等, 这些接口是在通用连接框架上的扩展实现。Connection 类是最基本的通用连接类型, 只定义了 close() 一个方法用于关闭连接, 而创建连接的操作使用 Connector 类的 open() 方法实现。

本文设计示例中使用 HTTP 协议与服务器进行网络通信。HttpConn 连接线程需要完成两项工作:

(1) 连接服务器, 如果成功, 则启动定时器定时读取服务器上存储的图片。

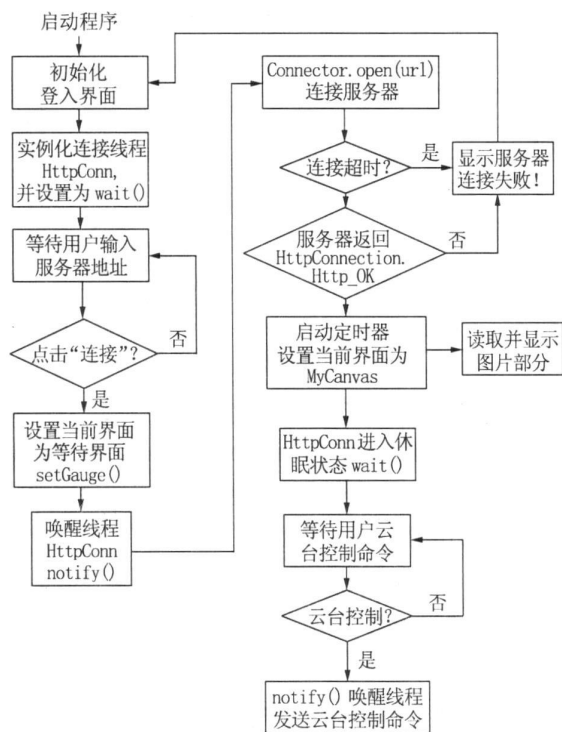


图3 网络连接及 HttpConn 线程部分流程图

首先在运行程序并初始化用户界面时, 实例化 HttpConn 线程, 并设置线程处于休眠状态。通常连接服务器时要花费一些等待时间, 先设置当前界面为连接等待界面, 然后使用 notify() 唤醒线程, 同时读取用户输入的 URL 地址, 与服务器连接。如果连接超时或者没有收到服务器返回 HttpURLConnection.HTTP_OK 响应, 则向用户报告错误, 否则启动定时器, 开始读图片并设置当前界面为 MyCanvas, 显示图片。

(2) 等待用户控制云台摄像机, 向服务器发送云台控制命令。

HttpConn 线程成功连接服务器, 启动定时器 timer, 显示图片后进入休眠状态, 直到手机有控制云台命令时, 再次用 notify() 唤醒 HttpConn 线程, 向服务器传送云台控制命令, 如图 3 所示。

2.5 云台控制部分设计与实现

手机用户使用 HTTP 协议向服务器发送云台控制命令, 服务器响应向客户端并且控制云台摄像头进行相应的操作, 云台控制接口参数如表 1 所示。

表 1 云台控制接口参数设计

Key	Value
command	Up_Start, 向上
	Up_Stop
	Down_Start, 向下
	Down_Stop
	Left_Start, 向左
	Left_Stop
	Right_Start, 向右
	Right_Stop
	ZoomTele_Start, 放大图像
	ZoomTele_Stop
	ZoomWide_Start, 缩小图像, 加大视野
	ZoomWide_Stop

(说明: Up_Start 是向上开始命令, Up_Stop 为向上停止命令, 当接受到命令后, 摄像头视角向上移动, 直到收到向上停止命令或限制位置, 其他类似。)

用户要控制控制云台摄像头的转动和图像的放大缩小时, 可以通过手机上对应的不同按键实现。以控制向上为例, 手机按键按下时发送向上开始命令, 按键弹起时发送向上停止命令, 其他命令类似。

MIDP 规范中低级事件的处理都定义在类 Canvas 中, 主要包括处理设备定义的按键、指针以及显示类事件。当用户在设备上按下按键并放开的时候, 产生 keyPressed() 和 keyReleased() 事件, 并将按键代码(keyCode) 传给事件处理程序 keyPressed() 和 keyReleased()。keyPressed() 和 keyReleased() 的具体处理方法需要开发者自己实现。本文设计示例在 keyPressed() 实现控制开始命令, 在 keyReleased() 实

现控制停止命令, 根据按键字母和按键动作的不同, 产生相应的控制命令参数。云台控制部分流程如图 4 所示。

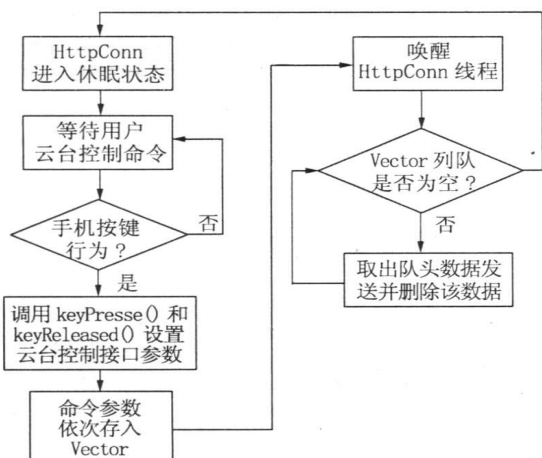


图 4 云台控制部分流程图

考虑到实际网络速度的影响, 为了保证控制开始、停止命令能按照顺序达到服务器, 使用 Vector 类, 此类实现了一个动态增长的对象数组, 使用 vector.addElement() 把获得的控制命令参数依次存入 Vector, 然后唤醒线程, 依次发送控制命令。

```
while(! vector.isEmpty()) {
    //判断 vector 是否为空, 保证存储的控制命令参数全部发送;
    String cont= (String) vector.firstElement();
    //每次取出队头数据;
    sendHttpPost( cont ); //发送云台控制命令参数;
    vector.removeElementAt( 0 );
    //发送后移除队头数据;
```

2.6 定时刷新图片和显示部分设计与实现

定时从服务器读取图片显示部分, 使用 Timer 与 TimerTask 类完成这部分的功能。Timer 类是一个定时器, 可以设定成特定时间或者特定时间周期产生信号, TimerTask 类代表一个可以被定时器进行时间控制的任务。Timer 一旦与某个 TimerTask 产生关联, 就会在产生信号的同时, 连带执行 TimerTask 所定义的工作。本文设计示例中编写 mytask 类继承 TimerTask 类, 并在 run() 方法中实现从服务器读取图片并调用图片显示函数的功能。通过 Timer 的 schedule 方法设定图片刷新的时间周期, 并将它与 mytask 关联, 一旦 timer 在设定的时间周期产生信号, mytask 的 run() 方法就被执行。设计使用 Timer 与 TimerTask 类来完成定时刷新图片的功能, 不需要

为图片读取和显示另外再开启一个线程, 减小了程序的复杂度。

图像绘制使用低级用户界面的 Canvas 类来实现, 编写 MyCanvas 继承于 Canvas 类, 通过向 paint() 方法中传递一个 Graphics 对象, 调用 Graphics 对象 g 的 drawImage 绘制图形。

以下是程序运行的效果图, 图 5 是在 WTK 手机模拟器上运行显示的结果, 采用的模拟器是 Default-ColorPhone. 图 6 是程序在 NOKIA6600 上运行显示的效果。



图 5 WTK 手机模拟器上运行显示的结果



图 6 NOKIA 6600 上运行显示效果图

3 结束语

本文主要介绍了远程监控系统中基于 J2ME 手机客户端软件的一种设计与实现, 示例程序在模拟器和实机上都成功实现了手机对云台摄像头的控制以及定时刷新显示监视点图片的功能。随着无线通信技术的发展, 远程图片监控已经不能满足人们对远程监控系统的要求。3G 移动通信系统走向实用, 为视频文件的传输提供了足够的网络带宽, 为远程视频监控的实现提供了更加强有力的支持, 使用手机作为远程视频监控系统的客户端已经成为远程监控中的研究热点。要实现手机的远程视频监控, 做到实时监视视频播放, 最关键的就是要求手机对流媒体播放的支持, 这当然也可以通过编写第三方软件来实现。

参考文献:

- [1] [美]里格斯, 等. J2ME 无线设备程序设计[M]. 肖炜, 郭晓刚, 译. 北京: 电子工业出版社, 2006.
- [2] Leopold Lee. J2ME 手机编程基础[M]. 清华大学出版社, 2003.
- [3] 张小玮. J2ME 无线平台应用开发[M]. 北京: 清华大学出版社, 2004.
- [4] 王森. Java 手机/PDA 程序设计入门[M]. 北京: 电子工业出版社, 2004.
- [5] 詹建飞. J2ME 开发精解[M]. 北京: 电子工业出版社, 2006.

责任编辑: 么丽苹