

本 科 生 毕 业 论 文（设计）



题目 面向环境数据的mashup应用开发

姓名与学号 胡雪峰 3062211056

指导教师 陈华钧

年级与专业 06软件工程

所在学院 计算机科学与技术学院

A Dissertation Submitted to Zhejiang University for the Degree of Bachelor of Engineering



TITLE: Environment data oriented mashup application development

Author: Xuefeng Hu

Supervisor: Huajun Chen

Subject: Mashup

College: College of Computer Science

Submitted Date: 2010.5.30

浙江大学本科生毕业论文（设计）诚信承诺书

1.本人郑重地承诺所呈交的毕业论文（设计），是在指导教师的指导下严格按照学校和学院有关规定完成的。

2.本人在毕业论文（设计）中引用他人的观点和参考资料均加以注释和说明。

3. 本人承诺在毕业论文（设计）选题和研究内容过程中没有抄袭他人研究成果和伪造相关数据等行为。

4. 在毕业论文（设计）中对侵犯任何方面知识产权的行为，由本人承担相应的法律责任。

毕业论文（设计）作者签名：

2010 年 5 月 30 日

摘要

当今社会，互联网对于现实生活的影响越来越大。人们希望能随时随地访问到网络的同时，还希望关于自己的动态信息能够在网络上被别人访问到。手机，掌上电脑等可以访问网络的移动设备正好可以满足人们的这一需求。然而，在移动设备上开发专门的应用的效率不高，开发者需要花费大量的时间和精力去实现很多基本模块。根据网络中流行的Mashup技术，我们提出了一种移动设备上的快速开发解决方案，即移动设备上的Mashup应用开发。Mashup技术可以让我们很容易的把移动设备上的数据源与网络中的数据源做混搭，从而开发出新的应用。而且，为了更加丰富移动设备Mashup应用开发的应用场景，我们用传感器接收本地环境数据，把它作为本地数据源。我们把本地环境数据与网络中的数据源Mashup起来，开发了一个面向环境数据的应用。该应用可以让人们实时的在地图上共享到其他位置的环境信息。在本文中将介绍这个应用的具体实现。

关键词　环境数据，传感器，Mashup，移动设备

Abstract

Nowadays the internet has an increasing influence to the real life. People wish to access the internet anytime and anywhere, on the other hand, wish the dynamic information about themselves can be accessed by others. Mobile devices such as Cellular phone and PDA can meet the demand. However, it is not efficient to develop new customed application. Developers need to cost a lot of time and energy to implement many base modules. Based on the popular mashup application, we propose a solution for rapid development on mobile devices, which is so called mashup application development on mobile devices. Mashup allows us to easily combine data source on mobile devices and data source in the internet together to make up new applications. In addition, to make more scenarios of mobile devices mashup application development, we collect local environment data by sensors and make the data into a local data source. Then we mashup the local environment data and the data source from the internet to develop an environment data oriented application. This application allows people to share real-time environment information from other positions on the map. In the paper we will introduce the concrete implementation of the application.

**Keywords**Sensor, Mashup, Mobile Device, Environment Data

目录

[摘要 I](#_Toc264028312)

[Abstract II](#_Toc264028313)

[目录 III](#_Toc264028314)

[第1章 绪论 1](#_Toc264028315)

[1.1 课题背景 1](#_Toc264028316)

[1.2 国内外现状 2](#_Toc264028317)

[1.3 课题目标 2](#_Toc264028318)

[1.4 本文结构 3](#_Toc264028319)

[1.5 本章小结 3](#_Toc264028320)

[第2章 相关工作 4](#_Toc264028321)

[2.1 uSD卡 4](#_Toc264028322)

[2.2 Scratchbox 4](#_Toc264028323)

[2.3 本章小结 5](#_Toc264028324)

[第3章 体系结构 6](#_Toc264028325)

[3.1 整体架构 6](#_Toc264028326)

[3.1.1 系统框架概述 6](#_Toc264028327)

[3.1.2 系统流程 7](#_Toc264028328)

[3.2 环境数据模块 9](#_Toc264028329)

[3.2.1 环境数据模块框架 9](#_Toc264028330)

[3.2.2 通信模块 10](#_Toc264028331)

[3.3 Map API调用模块 11](#_Toc264028332)

[3.4 本章小结 11](#_Toc264028333)

[第4章 实现 12](#_Toc264028334)

[4.1 传感器节点 12](#_Toc264028335)

[4.1.1 传感器节点初始化 12](#_Toc264028336)

[4.1.2 数据包格式 13](#_Toc264028337)

[4.1.3 传感器节点数据发送 14](#_Toc264028338)

[4.2 通信与数据处理 14](#_Toc264028339)

[4.2.1 uSD卡的初始化 15](#_Toc264028340)

[4.2.2 数据处理 15](#_Toc264028341)

[4.3 Google Map API的调用 16](#_Toc264028342)

[4.3.1 嵌入的Google Map页面 16](#_Toc264028343)

[4.3.2 Map类 17](#_Toc264028344)

[4.4 Mashup实现 17](#_Toc264028345)

[4.4.1 UI设计 17](#_Toc264028346)

[4.4.2 功能实现 18](#_Toc264028347)

[4.5 本章小结 18](#_Toc264028348)

[第5章 应用场景 19](#_Toc264028349)

[5.1 应用场景介绍 19](#_Toc264028350)

[5.1.1 地图显示环境数据 19](#_Toc264028351)

[5.1.2 示例程序 19](#_Toc264028352)

[5.1.3 演示程序的条件 20](#_Toc264028353)

[5.2 本章小结 20](#_Toc264028354)

[第6章 总结 21](#_Toc264028355)

[6.1 讨论 21](#_Toc264028356)

[6.2 未来的工作 21](#_Toc264028357)

[参考文献 22](#_Toc264028358)

[致谢 23](#_Toc264028359)

绪论

课题背景

当今社会，互联网对于现实生活的影响越来越大。人们希望能随时随地访问到网络的同时，还希望关于自己的动态信息能够在网络上被别人访问到。手机，掌上电脑等与人们生活密切相关的移动设备在这方面的需求与日俱增。当然，传统的集成方法可以把移动设备与网络数据集成起来开发应用。但是，因为移动设备，特别是嵌入式传感器设备，都是在不同的软件和硬件平台上开发的，没有一个通用的通信标准协议，所以用传统的集成方法来开发新的应用是非常耗费时间和精力的。开发者必须重新实现大量基本的功能和特定的用户接口，因此开发的成本是相当高的。在理想的情况下，开发者应该可以只是做做类似于搭积木的工作，把已经做好的模块重构一下，就可以快速构建应用程序了。

一个现今流行的Web 2.0领域的技术-mashup，就是用来简化开发者开发工作的。Mashup是当今网络上新出现的一种网络现象。它是从多个分散的站点获取数据源，组合成新网络应用的一种应用模式。Mashup一般使用源应用的API接口，或者是一些rss输出作为数据源，用什么技术则没有什么限制。Mashup在极客群体和互联网玩家之中获得了极大的欢迎。因为它不需要很高的编程技能，只需要熟悉API和网络服务的工作方式就能进行开发，所以很快成为一个流行的网络现象。很多公司，例如yahoo，google，都为此提供开放接口，以吸引这个群体。使用mashup，开发将会变得越来越容易，开发者利用现有的网络数据源，就可以开发很酷的应用，例如哪里有停车空位的地图，又或者开发一个地图标记出通讯录上的联系人。

另外，近些年中，移动设备的功能越来越强大，很多移动设备上的系统已经开始运行PC机的应用程序了。以诺基亚N810平台为例，它拥有128M RAM，256M闪存和一个主频为400MHZ的OMAP2420微处理器，并且内置了Wifi模块。诺基亚N810平台上安装的是基于Linux的Maemo操作系统，并且安装了很多基于Linux的应用程序，其中包括了Mozilla浏览器。这样的配置让用户可以在诺基亚N810上用浏览器访问网页时体验到PC机上网时的感觉。并且，越来越多的高级编程语言，如Qt，GTK，Python等，都逐渐开始支持移动设备开发。这使得在移动设备上开发新的应用也逐渐像在PC机上开发那样，可以使用强大的高级语言库的支持。

基于以上情况，本课题提出了一种简单快速开发移动设备和网络的集成的方案，就是一种mashup的实现。在这种方案中，将在移动设备的本地调用网络中的网络服务API，并且与移动设备本地数据混搭，构建出新的mashup应用程序。本课题采用了Crossbow Imote2传感器来采集环境数据，把传感器数据作为移动设备的本地数据。由于传感器节点本身无法访问网络，因此本课题采用在诺基亚N810平台上插入一张装有Zigbee通信模块的uSD卡，利用uSD卡上的Zigbee通信模块与传感器节点通信，在诺基亚N810上收集传感器节点采集到的数据。然后，在诺基亚N810上使用Qt编程，把收集到的数据与Google Map API做混搭做成一个应用。

* 1. 国内外现状

国内外有不少项目小组都在做基于环境数据的mashup应用开发的工作，不过侧重点都有所不同。Dominique Guinard 和Vlad Trifa 在Towards the Web of Things: Web Mashups for Embedded Devices[1]一文中提出了一种基于嵌入式设备的mashup应用开发解决方案，侧重于获取电器设备中的能量传感器数据，用于开发能量监控之类的应用。Andreas Brodt 和Daniela Nicklas 在The TELAR Mobile Mashup Platform for Nokia Internet Tablets[2]一文中提出了一个诺基亚N810平台上的面向GPS传感器数据的mashup开发平台。

课题目标

本课题的目标是做传感器网络与网络服务在移动设备上的Mashup应用开发的一个尝试。这是一个探索性的过程，所以本课题选择了一个简单的应用场景，仅仅是把传感器节点作为REST资源，与电子地图做Mashup，在地图上显示传感器节点所在位置的环境数据。这样做的目的在于展示基于环境数据的Mashup应用开发的一种可能的方式。传感器节点本身不带有网络访问模块，例如Crossbow Imote2传感器，本身只带有Zigbee通信模块。这种传感器节点本身的限制使得以往使用传感器数据的应用开发都需要程序员来重新实现专门的与传感器通信的协议，这样重复的工作是十分浪费时间的。而本课题中，利用装有Zigbee模块的移动平台与传感器通信，采集到的数据可以通过Qt编程处理，并且通过调用网络中的网络服务API来实现mashup开发。这种开发方式大大的提高了利用传感器节点作为应用开发的效率，对于传感器网络的Mashup开发是一种十分有意义的探索。

本文结构

第2章介绍了一些与本课题相关的技术背景。

第3章构介绍了本课题中整体系统的体系结构，并且利用系统框架图和流程图介绍了各个模块的框架，以及各个模块之间的联系。

第4章分模块对整个系统的各个实现部分进行了具体介绍。

第5章介绍了由传感器网络服务与google map混搭而成的一个示例应用程序，介绍了程序的功能，展示了程序的界面。

第6章对整个课题进行了总结，讨论了课题技术的优点以及目前工作中的不足之处，总结了本人在项目中的工作，并且提出了未来的工作和改进的想法。

* 1. 本章小结

本章介绍了课题的背景，课题目标以及国内外相关课题的现状。通过以上内容可以看出，本课题研究的内容有自己侧重的方向，对于传感器网络的mashup开发是一种十分有意义的探索。

相关工作

uSD卡

本课题中使用uSD卡上的Zigbee模块使传感器节点和移动设备的通信。uSD是universal Sensor Data（通用传感器数据）的缩写。uSD卡是一张看起来像存储卡一样的外部硬件设备。它可以让移动设备通过Zigbee无线连接方式与外部无线传感器通信。为了简化基于uSD卡的开发，有一套基于QT C++的开发工具包已经被设计和开发出来了。此外，还有一个外部传感器数据记录器和数据库被构建出来，用来获取，存储和共享外部传感器数据，并且使设备服务可以根据外部传感器搜集到的数据进行更新。目前基于uSD的网络数据服务支持XML格式和JSON格式的数据。uSD卡支持的平台有Symbian和Maemo。本课题采用的移动设备是诺基亚N810，运行的是Maemo平台。

uSD卡主要由三部分组成，分别是存储介质 Nand Flash，射频芯片 Zigbee和控制模块 FPGA。其中Nand Flash主要负责存储用户信息和文件系统，Zigbee用于实现无线通信，FPGA 用于协调主机与Nand Flash和ZigBee之间的通信。

Scratchbox

本课题中的主要开发平台是诺基亚N810平台，操作系统是Maemo5.0。 Scratchbox就是Maemo SDK的交叉编译环境。Scratchbox实际上是一个完整的GNU/Linux系统，它模拟了部分Maemo平台的终端环境，并且可以运行一些开发工具。Scratchbox下可以选择两个target：DIABLO\_ARMEL和DIABLO\_X86，分别对应ARM平台和X86平台。X86平台是开发的环境，编译出来的代码可以运行在本机的调试环境中。在一个本机的终端开启一个例如Xephyr这样的 X server，然后在Scratchbox中运行一个Hildon Desktop，就可以在X server上显示一个模拟的Maemo桌面。这时，开发的图形界面软件就可以通过run-standalone.sh脚本在Scratchbox中调试运行了。ARM平台是交叉编译环境，编译出来的代码只能运行在ARM架构的嵌入式移动设备中，例如诺基亚N810平台。

因此，在Scratchbox中对Maemo开发与其他嵌入式开发环境相比，是一件相对较容易的事。因为Scratchbox的X86平台和ARM平台上的环境和工具基本上是一样的。以本课题中的开发为例，本课程在N810平台上的开发语言是Qt。在配置开发环境的时候，需要做的工作就是安装Scratchbox，登录Scratchbox，切换到X86平台，在源上下载安装Qt开发环境，再切换到ARM平台，重新安装一次Qt开发环境。而之后的开发过程基本上就是，在X86平台下编程，调试，完成之后切换到ARM平台下，编译之后把产生的二进制可执行文件拷贝到诺基亚N810平台上执行。基本上，真机上执行的效果比Scratchbox中模拟的效果还要更加高效。

本章小结

本章介绍了本项目中用到的关键技术和工具。uSD卡用来实现传感器与移动设备的通信，Scratchbox用来做开发与交叉编译。

体系结构

整体架构

系统框架概述

整个系统是以移动设备为中心的。传感器节点通过传感器板上的各种传感器采集各种环境数据（温度传感器采集温度数据，湿度传感器采集湿度数据，光强度传感器采集光强度数据），并按照指定的数据包格式在指定的无线信道上发送；在移动设备上，通过uSD卡上的Zigbee模块，在Zigbee信道上接收传感器发送的数据包，并在移动设备上对数据包进行解包，并把相关数据储存到指定的数据结构中；移动设备本身需要通过无线网络连接到互联网中，这样，就可以在移动设备上直接通过API访问网络中网络服务了。最后就是一个mashup的过程。在移动设备上把传感器数据（包括内部传感器，如GPS等，和外部传感器，如图3-1中的传感器节点）与网络服务数据经过整合，就可以混搭出一个新的应用了。总体系统框架见图3-1。

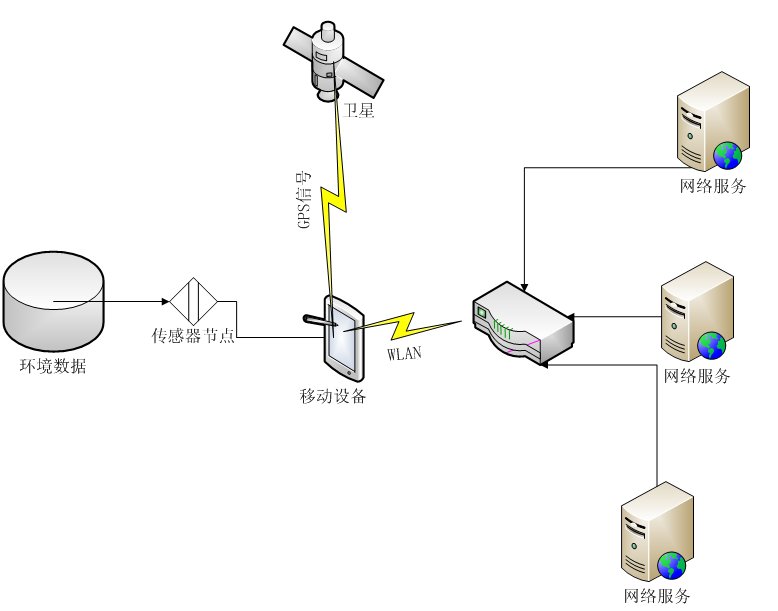


图 3-1 总体系统框架图

系统流程

图3-2所示的是系统工作的流程图。系统流程分为传感器节点数据通信流程与移动设备调用网络服务API流程。传感器节点数据通信流程开始后，传感器节点需要完成采集环境数据，打包环境数据与发送环境数据3个环节的工作。传感器节点只负责在指定信道发送数据，没有交互功能。在另一头，移动设备上通过初始化uSD卡的通信模块，设置一个与传感器节点发送的数据包在同一个信道的监听器，接收传感器节点发送的数据包。当uSD卡通信模块开始接收数据包后，会把接收到的二进制数据包发送到数据处理模块，通过数据处理模块的处理再返回给程序使用。调用网络服务API流程开始后，程序将获取一个网络连接对象，并且发送请求来调用网络服务API。在获得响应之后，系统就可以成功的初始化网络服务对象了。最后，在移动设备上把转换后的移动设备数据与网络服务数据呈现在同一个用户界面下，完成mashup工作。

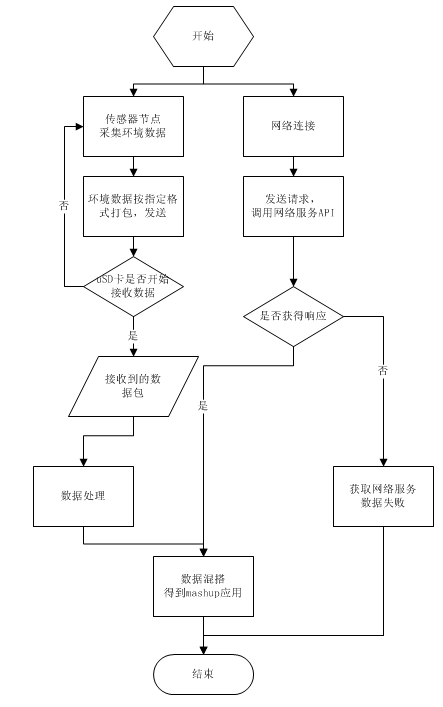


图 3-2 系统流程图

环境数据模块

环境数据模块框架

传感器模块由传感器数据采集模块，通信模块和数据处理模块组成。其中，数据采集模块由传感器内部芯片硬件实现，所以本文重点介绍通信模块和数据处理模块。

环境数据模块总体框架如图3-3所示。传感器节点通过板载的传感器模块，接收环境中的温度数据，湿度数据和光照强度数据。按照指定的数据包格式打包之后，传感器节点在指定的信道以一定的时间间隔发送环境数据。在移动设备端，通过uSD卡上的Zigbee模块，在Zigbee信道上接收数据，即可以接收到传感器节点发出的数据包。因为传输过程中，数据包是以字节流的格式传输的，所以在移动设备端无法直接使用接收到的数据包。数据包需要发送到数据处理模块进行处理。在数据处理模块中，把数据包作为输入，到指定的XML文件中进行查询，从而解析得到数据包中每一段的数值和意义，并且把解析结果返回给移动设备端。这样，可以直接调用的环境数据包就可以在移动设备端使用了。

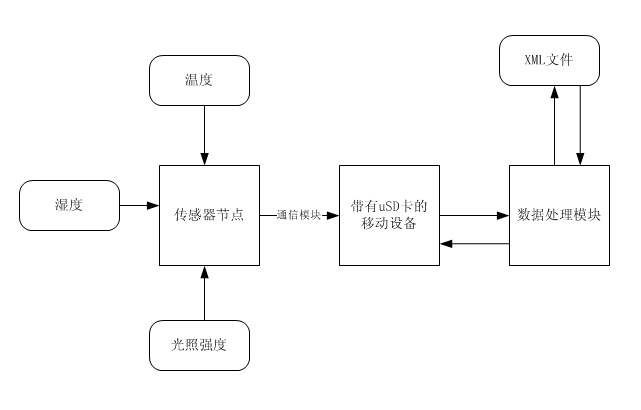
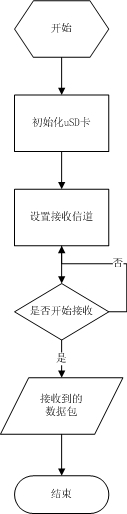
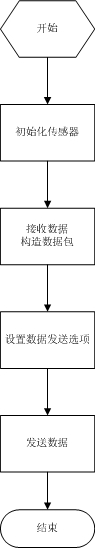


图 3-3 环境数据模块总体框架图

通信模块

通信模块是指传感器节点与移动设备之间通信的模块。传感器节点在通信模块中需要设置数据发送的选项，比如数据发送的信道，数据发送的时间间隔，发送的功率等。并且，在通信模块还需要定义发送的数据包格式，以便在移动设备端使用接收到的数据。在移动设备端，需要使用uSD卡上的zigbee模块来与传感器节点通信。所以，在通信之前需要在移动设备端初始化uSD卡，并且设置接收数据的信道。在这些工作之后，就可以在移动设备端接收到来自传感器的数据包了。通信模块流程图如图3-4所示。



图（a） 图（b）

图 3-4 通信模块流程图,

图（a）为传感器发送数据流程图，图（b）为uSD卡接收数据流程图

Map API调用模块

Map API调用使用了两个层次的调用。一个是表现层的调用，即在一个html页面中嵌入Javascript代码，通过Javascript调用Map API，对地图的数据展示方面起作用。另一个是在类中调用Map API。这种调用不是为了数据的展示，而是通过交互获得程序内部业务逻辑所需要的数据。一切关于数据展示的方法都是放在html页面中，通过Javascript函数实现。程序中需要设计到页面展示的方法不需要在自己的类中实现相应的方法，而只需要调用Javascript函数即可。图3-5是Map API调用模块的框架图。

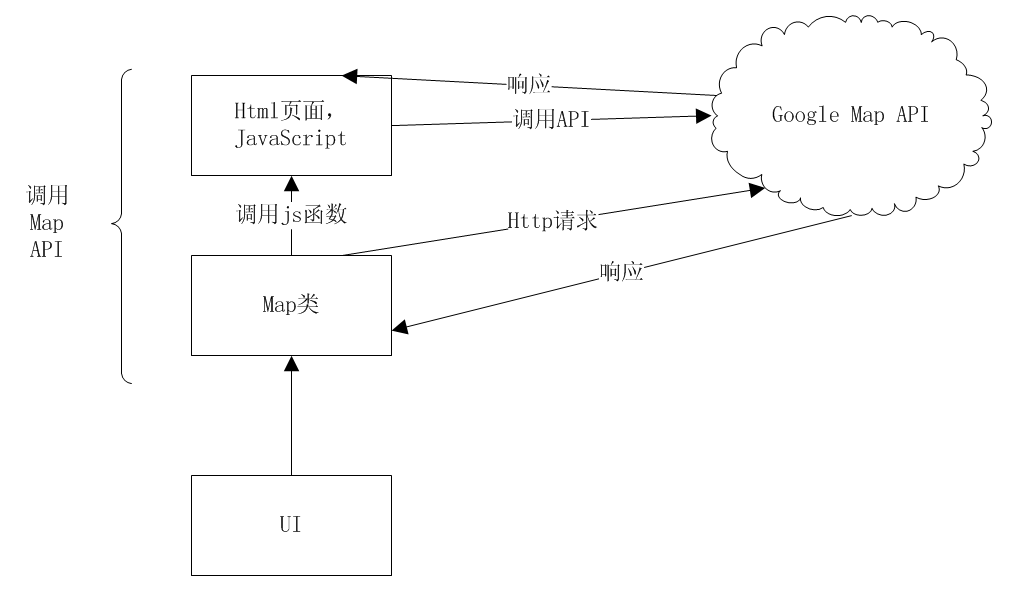


图 3-5 Map API调用模块框架图

* 1. 本章小结

本节首先介绍了系统的总体框架以及系统处理的流程，然后分模块介绍了环境数据模块和Map API调用模块的结构。通过本节的介绍，让大家对本项目的系统体系结构有一个全面的了解。这也是本人在项目设计中所完成的工作。

实现

传感器节点

本节介绍了通过imote2开发套件API对传感器节点编程，使传感器按照指定数据包格式以指定的时间间隔来发送数据包的过程。Imote2开发套件是基于C#的一套用于传感器节点开发的API，开发环境需要visual studio和micro framework2.0。

传感器节点初始化

传感器节点的初始化需要设置传感器的id，数据包的id，数据包的长度，节点的id，计数器，间隔时间，无线信道和功率等。此外，还需要创建硬件的对象来驱动硬件工作。如图4-1是Its400类的类图，对应传感器节点的功能。通过创建Its400类的实例，就可以通过实例的属性来获取传感器节点采集到的环境数据了。



图 4-1 Its400类的类图

数据包格式

如4.1.1节所示，数据包是一个14字节长度的数组。在传感器采集数据之后，通过调用传感器对象的get方法来获得相应的数据，并且把数据简单地以序列化的方式存储在数据包中发送。表4-1展示了数据包的具体格式，其中整型的数据均使用两个字节在数据包中储存。数据包的第1个字节是传感器的id，这是传感器的标识，用于识别不同的传感器。数据包第2个字节是数据包的id，它是数据包的标识，用于识别不同的数据包类型。数据包第3和第4字节是节点的id，它是传感器节点的标识，用于识别不同的传感器节点。数据包第5和第6字节是计数器，它是数据包的序列号，在发送过程中，每发送一个数据包计数器的值就会加1。数据包第7和第8字节是TI温度传感器上采集到的温度值，第9和第10个字节是Sensirion温度传感器上采集到的温度值，第11和第12个字节是湿度传感器上采集到的湿度值，第13和第14字节是光照传感器上采集到的光照强度值。

表 4-1 传感器数据包格式

|  |  |
| --- | --- |
| 序号 | 内容 |
| 0 | 传感器id |
| 1 | 数据包id |
| 2 | 节点id |
| 3 |
| 4 | 计数器 |
| 5 |
| 6 | TI传感器温度 |
| 7 |
| 8 | Sensirion传感器温度 |
| 9 |
| 10 | 湿度 |
| 11 |
| 12 | 光照强度 |
| 13 |

传感器节点数据发送

传感器节点数据发送使用TOSRadio类的send方法来实现。在调用send方法之前，可以使用TOSRadio类的setRadioOption方法来设置不同的选项。传感器节点的无线通信模块是Zigbee信号，将通过下一节中的uSD卡的Zigbee模块在诺基亚N810平台上接收。

通信与数据处理

本节首先在4.2.1小节介绍了通过uSD卡在诺基亚N810平台上接收传感器节点发送的无线信号的实现。然后，又在4.2.2小节介绍了使用基于XCL（XML-based Converter Library）的数据处理过程。

uSD卡的初始化

uSD卡的初始化用以下几个步骤完成：

1. 构建 CUSDCardManager 对象,并调用CUSDCardManager::init()方法对卡管理器进行初始化。
2. 调用CUSDCardManager::NumberOfUSDCards()方法得到uSD卡的数量。
3. 调用CUSDCardManager::GetUSDCardByIndex(int aIndex)方法，通过uSD卡的index来得到卡的引用。
4. 调用CUSDCardManager::GetZigbeeReceivingChannel()方法来得到接收Zigbee信号的信道，返回得到信道实例的引用。
5. 创建一个ChannelObserver类的实例，并调用CUSDChannel::SetObserver(CUSDChannelObserver \*aChannelObserver)方法，把这个ChannelObserver实例设置为ZigbeeReceivingChannel的观察者。

经过初始化之后，uSD卡开始接收数据操作可以通过调用CZigbeeReceivingChannel::StartReceiving()方法实现，停止接收数据操作可以通过调用CZigbeeReceivingChannel::StopReceiving()方法实现。

数据处理

数据处理的目的是把传感器传输过来的数据包转化成程序可用的数据格式。在这个过程中，需要一个XML文件来标识数据包中的各位数据各表示什么意义。这里用到了一个叫XCL(XML-based Converter Library)的函数库。使用数据包数据和数据包长度作为参数构建一个converter类的对象，并且调用converter::autoLoadBestXML()方法，加载XML文件，把数据包数据与XML文件匹配。此时，就可以通过调用converter::getDataFromStream()方法，获得返回的一个map类型对象。该map类型对象的键是数据包的id，值是包中各种数值的一个list。

这里调用的XML文件是一种用来描述传感器数据包结构的的文档，被称为CSDL（Context Source Description Language）。Converter对象需要一个CSDL文档来描述数据包的格式，从而解析数据包，并返回数据包中的实际有用的数据。下面是按照传感器节点发送的数据包的格式而定制的CSDL文件的片段。

下段文档中，condition标签表示该标签内的内容用于判断是否采用该CSDL文件作为数据包描述语言。Var标签定义了一个名为”@con”的变量，变量的值由pos标签和size标签决定从数据包中抽取的数据的位置和大小，在此例中，取”@con”变量的值为数据包中第0个位置起的一个字节的值。Expr标签表示了判断的条件。此例中，判断条件是”@con==0x16”，即如果数据包中的第一个字节的数据值为0x16的话，就使用这个CSDL文件来解析这个数据包。

|  |
| --- |
| <condition>  <var name="@con">  <pos>0</pos>  <size>1</size>  </var>  <expr>  <param1>@con</param1>  <oper>==</oper>  <param2>0x16</param2>  </expr>  </condition> |

Google Map API的调用

本节介绍了使用Qt调用Google Map API编程的过程。4.3.1小节介绍了在html文档内使用一段Javascript代码，调用Google Map API来展示地图。4.3.2小节介绍了实现的一个基于QWebView的类，通过实现类的方法来调用Google Map API在地图上定位，并且通过4.3.1小节中的页面来展示自定义的信息。

嵌入的Google Map页面

一个html文档被用来在应用程序窗口中展示Google Map。文档中没有html元素的代码，只有Javascript函数来调用Google Map API。下面代码展示了一个Javascript的函数initialize()。Initialize()方法中创建了一个GMap2的对象，并且给了地图一个初始化的中心，即初始化了一次地图。Initialize()方法被放在html页面的body的onload方法中，即打开该页面就执行一次initialize()方法。

在嵌入的html文档中，除了initialize()方法外，还有一个重要的方法，就是Open(x,y,temp，hum，light)方法。其中参数x,y是传感器的地图坐标，temp，hum，light分别是温度，湿度，光照强度数据。在Open方法中，把地图中心设置成了(x,y)坐标点上，并且在(x,y)点上添加了一个overlay，并把一个标记点加到overlay上。然后，再把temp，hun和light参数添加到标记点打开的气泡信息框中展示。

Map类

为了在Qt程序中调用Google Map API，创建了一个Map类。Map类继承自QWebView类，可以在框体内显示web页面。在Map类中实现了通过地址查询地图坐标的方法，代码如下。这个方法中通过发送请求到google map服务器来获得local地址的地图坐标。请求中包含了3个参数，第一个参数local是传入的参数，指定查询的地名；第二个参数是输出的类型，这里我们使用csv格式作为输出类型；第三个参数是google map API上申请的唯一的密钥。请求由QNetworkRequest(QString)方法发送，结果被QNetworkAccessManager对象用get方法获得。

在Map类中对Google Map的操作是通过对嵌入式页面中的Javascript函数的调用实现的。如果要在Map上加载一个坐标，Map类是调用了页面中的Javascript函数open(x,y)而实现的。

另外，通过Qt信号-槽的绑定机制把Google Map请求的响应信号与坐标位置获取槽连接，把刷新地图信号与加载坐标信息槽连接，在Map类中有效的搭建了一个反应机制。每当发送一个Google Map请求后，一个QNetworkAccessManager对象通过get方法等待服务器响应；在响应到达之后，就会触发坐标位置获取槽，从收到的响应数据中取出第3位和第4位数据，就是位置坐标的x坐标和y坐标。另外，每当类中函数发出刷新地图信号时，都会触发加载坐标信息槽，此时，获取到的位置坐标才会通过调用页面中的open方法来更改地图的当前中心位置。

Mashup实现

本节中介绍了传感器数据与Google Map API在实际代码中混搭的实现方式。

UI设计

UI设计部分采用Qt Creator来设计。主窗口是一个Form窗口，窗口主体部分是一个QWebView控件，该控件可以在程序中嵌入网页。窗口左侧是三个PushButton控件。界面如图4-1所示。

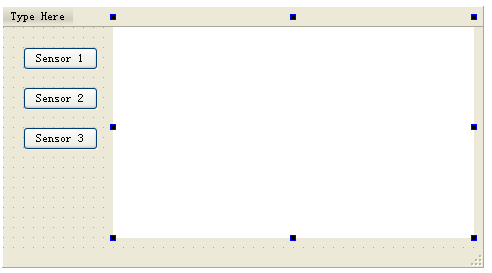


图 4-1 UI设计图

功能实现

程序中用一个继承了QWebView类的Map类来替代界面中的QWebView空间，用来加载调用Google Map API的html文档，这部分用来在界面上展示地图信息。左侧的三个PushButton用来调用3个传感器的信息。当点击一个PushButton的时候，将会在按钮响应中调用uSDCard中的GetInfo方法，GetInfo方法又调用数据接收与处理方法，获得与该PushButton对应的传感器所发送的数据包。然后调用Map类的方法，把位置信息在http请求中发送到Google Map服务器，从响应的数据包中获取该传感器的位置信息所对应的地图坐标，然后通过Map类加载数据的方法调用页面的JavaScript函数，刷新地图上的信息，在传感器所在位置显示当地的环境数据。

本章小结

本章介绍了各个模块具体的实现，对于每一个模块的实现都做了详细的介绍。这也是本人在项目实现过程中所完成的工作。

应用场景

应用场景介绍

地图显示环境数据

演示程序创建的一个应用场景就是在地图上显示指定的传感器当前的环境数据。因为无法取到GPS位置信息，所以演示程序中根据传感器数据包中的数据，为每个传感器模拟了一个位置信息。当传感器节点开始发送数据后，在移动设备端会根据接收到的数据包信息来获取传感器节点的位置信息，并通过调用Google map API来获取地图上的坐标。然后再把该传感器节点的数据显示在地图上对应的坐标上。目前该演示程序支持展示3个传感器节点的信息，可以通过界面上的选择按钮来设置地图的焦点到指定的传感器节点。

示例程序

示例程序中，提供了3个按钮，点击后分别响应得到对应的传感器节点最新的数据。如图5-1所示，点击Sensor1按钮，主窗口中的地图焦点会转移到Sensor1所在的地点，并且会在Sensor1所在位置设置一个标记。然后，在标记的位置弹出一个气泡窗口，并且显示1号传感器节点最新发送的环境数据。

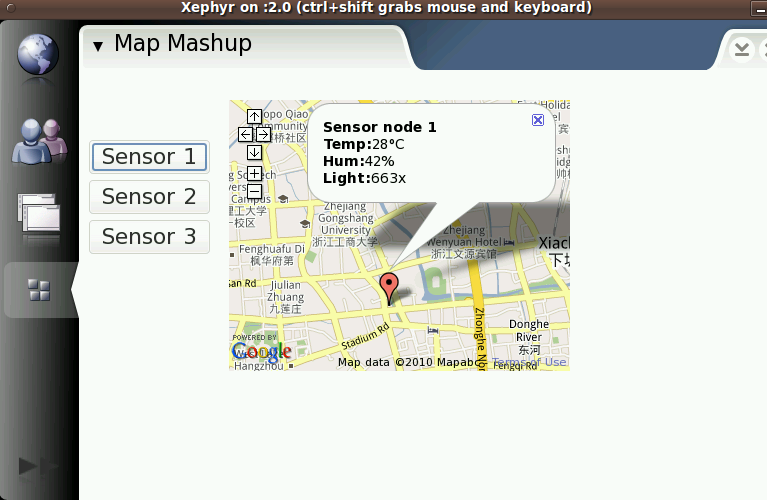


图 5-1 示例程序截图

演示程序的条件

因为演示程序需要设备访问Google Map API来获取地图信息，所以移动设备需要能够访问网络。并且，网络的质量对于地图刷新的速度影响较大，所以，对诺基亚N810平台来说，设备需要一个信号较强，且能够连接到网络的wifi接入点。

本章小结

本章介绍了一个实际的应用场景示例程序。通过示例程序可以展示出面向环境数据的mashup应用开发的基本模式。

总结

讨论

在本文中，尝试了一种新型的mashup应用开发模式，即在移动设备上使用环境数据与网络服务开发mashup应用。通过这种新模式的尝试，可以看到很多应用开发的可能性。生活中随处可见的嵌入式设备都可以是一个传感器。通过移动设备把传感器数据与网络服务结合的mashup开发有着十分广阔的发展前景。

在本项目中，本人参与并基本完成了指定的工作，包括以下几点：

1. 传感器编程，涉及到Imote2传感器API的调用。
2. Maemo开发环境的搭建，涉及到Linux下Scratchbox的安装配置，Maemo开发环境的安装，Qt的安装。
3. uSD卡与传感器的数据通信，涉及uSD SDK API的调用，uSD卡的初始化以及设置。
4. 数据包解析，涉及到XCL库的调用，以及自定义CSDL文件。
5. 网络服务API调用和开发一个mashup示例程序，涉及到Qt Gui开发。

未来的工作

面向环境数据的mashup应用开发课题到此暂时告一段落。对于这个课题来说，本文所做的工作是十分有限的，未来还有很多可以完成的工作。未来在这个课题上的继续探索的工作大致可以分为两个方向。一个方向就是在现在的基础上，继续获取更多的网络服务，并且加入到现有的mashup体系中来。相信更多的网络服务的加入，会使得围绕环境数据的mashup开发创造出更多有用的应用场景，也可以使mashup开发的应用更加丰满。另一个方向就是把传感器数据通过移动设备传输到网络中，通过网络中的服务器把传感器数据以网络服务的方式发布。这样，世界各地的传感器真正的构成了一个“传感器网络”，通过调用这个网络服务，可以在一个应用中调用整个传感器网络中的资源。完成这一步之后，面向环境数据的mashup应用开发才真正的和普通网络服务的mashup应用开发没有区别了。

参考文献

1. Dominique Guinard，Vlad Trifa. Towards the Web of Things: Web Mashups for Embedded Devices. Integror.net. 2009.
2. Andreas Brodt, Daniela Nicklas. The TELAR mobile mashup platform for Nokia internet tablets. ACM. 2008.
3. Timo Koskela, Nonna Kostamo, Otso Kassinen, Juuso Ohtonen, and Mika Ylianttila. Towards Context-Aware Mobile Web 2.0 Service Architecture. 2007.
4. Jasmin Blanchette, Mark Summerfield. C++ GUI Programming with Qt 4. Prentice Hall. 2006.
5. <http://usdcard.org/>
6. <http://wiki.forum.nokia.com/index.php/Wiki_Home>
7. [http://www.geonames.org/export/wikipedia-webservice.html#](http://www.geonames.org/export/wikipedia-webservice.html)
8. Robert Martignoni, Katarina Stanoevska-Slabeva. Mobile Web 2.0. 20th Bled eConference. 2007.
9. Sami Mäkeläinen and Timo Alakoski. Fixed-Mobile Hybrid Mashups: Applying the REST Principles to Mobile-Specific Resources. 2008.
10. E. Michael Maximilien. Mobile Mashups: Thoughts, Directions, and Challenges. IEEE Computer Society  Washington, DC, USA. 2008.

致谢

过去的几个月做毕设的时间是我人生中一段重要的旅程。在此毕设完成之际，我要感谢我的导师陈华钧老师对我这段时间的悉心指导，感谢学长彭志鹏在毕设工作上对我的帮助，并且对我的毕设提了很多十分有帮助的建议，还要感谢父母一路以来对我的陪伴和鼓励。

**本科生毕业论文（设计）任务书**

**一、题目：**

**二、指导教师对毕业论文（设计）的进度安排及任务要求：**

2009.11-2010.1 资料查阅以及学习

2010.3.1-2010.3.15 Imote2 Builder SDK 环境的搭建和安装，

uSD SDK环境的搭建和安装，

阅读Imote2 Builder SDK和uSD SDK文档，学习API

2010.3.16-2010.3.31 IPR2400开发板开发及测试

2010.4.1-2010.4.15 uSD card开发及测试

2010.4.18-2010.4.24 Mashup应用场景创建以及Demo应用的开发

2010.4.26-2010.5.5 测试，性能及用户体验优化

**起讫日期 200 年 月 日 至 200 年 月 日**

**指导教师**（**签名） 职称**

**三、系或研究所审核意见:**

**负责人**（**签名）**

**年 月 日**

**毕 业 论 文（设计） 考 核**

**一、指导教师对毕业论文（设计）的评语：**

**对相关技术进行了研究和尝试，基本完成了规定的任务。**

**指导教师(签名）**

**年 月 日**

**二、答辩小组对毕业论文（设计）的答辩评语及总评成绩：**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **成绩比例** | **实习报告**  **占（10%）** | **开题报告**  **占（20%）** | **外文翻译**  **占（10%）** | **毕业论文（设计）质量及答辩**  **占（60%）** | **总评成绩** |
| **分**  **值** |  |  |  |  |  |

**答辩小组负责人（签名）**

**年 月 日**