

嵌入式 Linux 平台下地理信息系统^①

马志强, 张德兴, 杨 磊

(总参谋部通信训练基地, 宣化 075100)

摘要: 在军事行动、抢险救灾等特殊活动中, 需要使用定制的便携地理信息系统。讨论了嵌入式 Linux 下, 以 QT、MAPCAP、MapInfo 等软件为开发工具, 构建地理信息系统的步骤和方法。经测试, 系统运行良好, 整个设计思路对快速开发嵌入式地理信息系统有一定参考价值。

关键词: Linux 系统; 地理信息系统; 电子地图; 最短路径算法

Geographic Information System Based on Embedded Linux System

MA Zhi-Qiang, ZHANG De-Xing, YANG Lei

(Communication Training Base, PLA General Staff Headquarters, Xuanhua 075100, China)

Abstract: In military action, emergency rescue and disaster relief and other special activities need to use custom portable geographic information system(GIS). This article discusses the steps and methods to construct GIS in embedded Linux environment, taking QT, MAPCAP, MapInfo and other softwares as development tools. In testing, the system works well. The whole design ideas have certain reference value to rapidly develop embedded GIS.

Key words: Linux system, geographic information system, electronic map, shortest path algorithm

1 引言

地理信息系统 (Geographic Information System, 简称 GIS) 作为获取、存储、分析和管理地理空间数据的重要工具、技术和学科, 近年来得到了广泛关注和迅猛发展, 并广泛应用于资源调查、国土管理、城市规划、交通运输、军事公安、水利电力等领域。嵌入式 GIS 是新一代地理信息系统发展方向之一, 它是运行在嵌入式计算机系统上高度浓缩、高度精简的 GIS 软件系统。GIS 空间数据量非常大, 所需存储空间也很大, 这与嵌入式设备有限的系统资源形成矛盾, 因此针对嵌入式设备的特点并结合 GIS 应用程序的需求设计嵌入式 GIS 平台是非常必要的。本文重点讨论了在嵌入式开发平台的基础上, 制作电子地图、提取地图数据以及控制地图显示的方法, 并对导航中非常重要的最短路径算法进行了介绍。

2 嵌入式开发平台

2.1 硬件平台

系统的硬件平台采用核心板、底板相结合的方式设计实现, 主要功能模块如图 1 所示。处理器采用 Marvell 公司的 PXA310。它是专门为手持设备、GPS 定位系统、无线手持和其他消费类电子设备而设计的, 采用 90 纳米工艺, 运行频率高达 624MHz, 内嵌视频加速器, 支持对 H.264、MPEG-4、H.263、RealVideo 以及 Microsoft WMV9 的编码和解码; 支持 CIF 分辨率的 3.5G 视频通话, 可以使用户更加流畅的观看以上格式的高码率、高分辨率视频文件, Marvell 电源管理技术在设备处于待机状态或播放音频、视频时都能帮助系统节省电力, 以达到延长电池使用时间的目的。

GPS 模块采用瑞士 u-blox 公司 LEA-6S, 它具有 50 个接收通道, 可同步追踪 GPS 及伽利略导航卫星信号, 提供 UART、USB、DDC 等多种接口, 冷启动 29

① 收稿时间:2011-04-13;收到修改稿时间:2011-05-05

秒,热启动1秒,在 Autonomous 模式下定位精度2.5米,在 SBAS 模式下为2米。

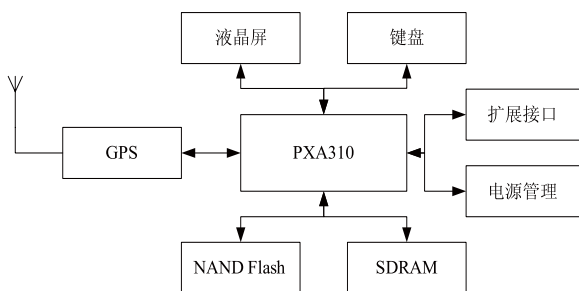


图1 硬件平台框图

液晶屏采用带触摸屏的 OLED; 键盘为 3×4 矩阵键盘; 扩展接口接 USB、串口、Micro SD, 主要是在导入地图和调试时使用。为满足便携设备对电源的需要, 尽可能延长工作时间, 电源管理电路采用锂电池供电模式, 充电电路选用 MAX1551 芯片, 电源管理模块采用 MAX1586 电源管理芯片。

2.2 软件环境

2.2.1 操作系统

主流的嵌入式系统有 Linux、Windows CE、Windows Mobile、uClinux、uCOS-II、VxWorks 等十几种。其中 Linux 的开发和研究是目前嵌入式操作系统领域的热点之一。其源代码开放, 内核小巧灵活、稳定高效、易于裁剪, 这使 Linux 很适合嵌入式系统的应用。开发人员可以根据实际需要自行定制 Linux 操作系统内核, 使之更为精简实用。Linux 内核的最新版本是今年3月推出的 2.6.38, 可以在网上下载得到。本文采用的是 2.6.21 版本 Linux 操作系统。

2.2.2 开发环境

在嵌入式 Linux 下比较常用的软件开发工具有 QT/Embedded 和 MiniGUI。MiniGUI 的开发始于 1998 年底, 经过飞漫软件多年的精心打造, 已经成为性能优良、功能丰富的跨操作系统嵌入式图形用户界面支持系统。从 2.0.4 开始, MiniGUI 已经成为商业化软件, 目前版本为 3.0。

Qt/Embedded 是著名的挪威的 Trolltech (奇趣科技) 于 2001 年推出的面向嵌入式系统的 Qt 版本, 是一个面向嵌入式系统的类库与应用框架, 2008 年 6 月 Trolltech 被诺基亚收购, 加速了其跨平台开发战略。QT/Embedded 类库完全采用 C++ 封装, 兼容同版本的

Qt-X11。因为 Qt 是 KDE 等项目使用的 GUI 支持库, 所以其可移植性非常好, 许多基于 Qt 的 X Window 程序可以非常方便地移植到嵌入式系统。基于 Qt/Embedded 的 Qtopia 是 Trolltech 公司为采用嵌入式 Linux 操作系统的电子设备开发的应用平台, 是一个著名的开放源代码项目。Qtopia 包含完整的应用层、灵活的用户界面、窗口操作系统、应用程序启动程序以及开发框架。在 QT4.2 以后的版本中, QT/Embedded 被整合进来, 形成统一的 QT。从 QT4.5 开始, QT 遵循 LGPL 协议开放源码。作为一款轻量级软件, 占有资源少是 MiniGUI 的一大亮点, 但是其功能没有 QT 丰富, 且代码设计没有 QT 简单方便, 因此我们选择 QT4 作为开发软件^[1-3]。

3 电子地图的制作

GIS 软件很多, 国外的有 AutoCAD Map3d、Arc GIS、MapInfo 等, 国内的有 MapGIS、SuperMap 等。下面我们借助多种工具软件, 定制自己的电子地图。

3.1 抓取网络地图

在实验中, 我们采用 MAPCAP 对一些网络电子地图进行抓取。用户只需要设置抓图区域的大小以及整幅地图的宽度和高度, 其余工作 (截屏、拖图、拼接) 由程序自动完成。得到的地图可以保存为 TIFF 格式的栅格图像文件。

3.2 选择定位点, 标定地标经纬度

在地图上标定一些地标的经纬度, 选择的定位点越多, 其精确度越高。

3.3 生成等高线

从 <http://srtm.csi.cgiar.org/> 下载 SRTM 90 高程数据, 使用 Global Mapper 生成 shapefile 格式的等高线。

3.4 生成 MapInfo 地图

MapInfo 软件中打开由抓取的网络地图生成的 tiff 文件。打开后选择注册地图。利用 2.2 中得到的标定将地图标定。建立道路信息表、节点信息表、显示控制信息表 (表 1-表 3)。

表 1 中, point1、point2 用来保存道路两端的连结点 ID, cost 保存道路通行因子, 用于计算路由, type 定义道路类型, 0 城镇街道, 1 乡村柏油路, 2 乡村土路, 3 县道, 4 省道, 5 国道, 6 高速。表 2 中, connection 保存连入此点的道路数, road_point 保存相邻的道路、节点信息, 结构为: 道路 ID: 节点 ID; 道路 ID: 节点 ID。

表 3 中, level 保存显示控制数据, 用来设置在多大的比例尺级别下可显示。

表 1 道路信息表

字段名	数据类型
id	integer
name	character
length	integer
point1	integer
Point2	integer
cost	integer
type	integer

表 2 道路节点信息表

字段名	数据类型
id	integer
connection	character
road_point	integer

表 3 显示控制信息表

字段名	数据类型
id	integer
name	character
level	integer

在图层控制中依次选择上述三个表, 设置可编辑。用绘图工具根据网络地图描绘道路、添加道路节点, 并在相应的表中添加相关信息。

4 地图数据的提取

4.1 导出 MIF 文件

MapInfo 的数据格式主要分为 TAB 和 MIF 两种^[4]。TAB 格式是 MapInfo 唯一的数据存储格式, 但由于商用原因一直未公开其格式。MIF 格式是 MapInfo 提供的一种与外界交换数据的机制, 其空间数据以 ASCII 方式保存。当以 MIF 格式存储地图数据时, 每个表的数据都是用两个文件来保存, 一个是.MIF 文件, 用来保存空间对象的几何数据, 一个是.MID 文件, 用来保存与几何数据相对应的属性数据。我们主要研究 MIF 格式的地图数据, 在 MapInfo 中分别选择道路表和节点表导出.MIF 文件。

4.2 通过 MITAB 库读取 MIF 文件数据

MITAB^[5]是一个开放源码的 C++库, 目前最新版本为 2008 年 4 月发布的 1.7.0。MITAB 功能强大, 通

过调用这个函数库, 不仅可以读取 MapInfo 的.TAB 和.MIF/MID 格式文件, 也可写入即实现地图的编辑。文献[6]中对采用 MITAB 的 C++接口函数读取地图数据进行的探讨, 但是由于 MITAB 的开发者对它的 C++接口函数并没有说明, 也缺少实例, 使用中有一定难度。为了方便调用, MITAB 还提供了一个简便的 C 语言的函数接口, 并提供了详细说明和示例。如用 mitab_c_get_feature_count() 获取地图上对象个数; 用 mitab_c_get_type() 获取对象类型, 点、线、多边形或者区域; 用 mitab_c_get_color() 获取画笔颜色; mitab_c_get_vertex_x() 和 mitab_c_get_vertex_y() 获取每个顶点的球面坐标等等。这里, 选用 C 接口来调用 MITAB 的库函数获取 MapInfo 地图数据。下面给出了读取地图数据的示例代码。

```
//打开地图数据文件
mitab_handle handle= mitab_c_open("./xh01.MIF");
int featureID[100];
//获得下一个有效代码编号
featureID[1] = mitab_c_next_feature_id(handle,-1);
mitab_feature feature[MAX_OBJECTS];
//读出一个对象
feature[1]=mitab_c_read_feature(handle,featureID[1]);
//获得矢量对象类型, 例如返回值 8 表示为矩形
int type = mitab_c_get_type(feature[1]);
mitab_c_close (handle) //关闭文件
```

5 地图的显示控制

在获取了地图的数据后, 需要在嵌入式 Linux 下实现地图的绘制, 也就是说要将电子地图的要素一一显示出来, 同时要把 GPS 模块传输过来的经纬度数据与地图结合起来, 在地图上实现定位。这部分功能由 QT4 开发实现, QT4 中有现成的类可以显示点、线、多边形、椭圆、矩形、文本, 显然, MapInfo 的地图要素是可以用上述的类来绘出的。每个地图要素要绘在哪里, 则由要素中的经纬度来决定, 借助高斯-克吕格地图投影公式把读取出来的经纬度坐标转换成大地的平面坐标。

QT4 中的 2D 绘图部分称为 Arthur 绘图系统。它由三个类支撑整个框架: QPainter、QPainterDevice 和 QPainterEngine。QPainter 用来执行具体的绘图相关操

作,如画点、线,填充、交换、alpha通道等。QPainterDevice是QPainter用来绘图的绘图设备,QT中有几种预定义的绘图设备,如QWidget、QPixmap、QPrinter等,它们都由QPainterDevice继承。QPainterEngine类提供了不同类型设备的接口,QPainterEngine对程序员不透明,由QPainter、QPainterDevice类与其进行交互。从QT4.2开始,Graphics View框架取代了QCanvas,Graphics View框架使用了MVC模式,适合对大量2D图元的管理,Graphics View框架中,场景(scene)存储了图形数据,它通过视图(view)以多种表现形式,每个图元(item)可以单独进行控制。利用Graphics View框架可以有效实现地图的放大、缩小,可以任意拖动地图快速漫游,根据比例尺动态切换图层显示状态,在地图上标注文字等等。图2是以张家口市宣化区为例显示的最大放大比例时地图的局部效果图。

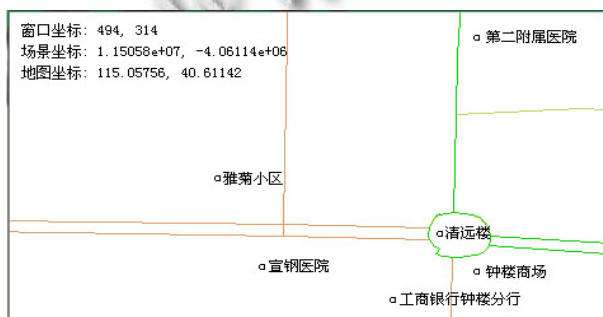


图2 电子地图

需要说明的是,QT3中的2D绘图采用的是QCanvas模块。QT4为了能兼容QT3,把QCanvas模块移到Q3Support库并重新命名,如QCanvas变为Q3Canvas,QCanvasEllipse变为Q3CanvasEllipse等等。对于QT3的老用户,尽管可以在QT4中沿用QCanvas,但是Trolltech并不建议这么做^[7]。

6 最短路径算法

最短路径问题是图论研究中的一个经典问题,也是地理信息系统的研究焦点。用于解决最短路径问题的算法被称为“最短路径算法”。最短路径算法有多种,如Dijkstra算法、Floyd算法、A*算法、深度优先搜索算法、动态规划法、蚁群算法等等。以上各种算法针对不

同的网络特征、应用需求及具体的软硬件环境,在空间复杂度、时间复杂度及易操作性等方面各具特色。

Dijkstra算法是由G. w. Dijkstra于1959年提出的一个适用于所有弧的权均为非负的最短路算法,也是目前公认的求解最短路问题的最经典的算法。Dijkstra算法的优点是适合于计算两点间的最短路径问题,只要路权为正,总能找到两点间的最短路径。其缺点是算法的搜索速度受图的存储结构的影响,图的存储空间越小搜索速度越快,反之,图的存储空间大,则会降低搜索速度。

Floyd算法又称为插点法,是一种用于寻找给定的加权图中顶点间最短路径的算法。其核心思想就是通过一个图的权值矩阵求出它的每两点间的最短路径矩阵。Floyd算法简单有效,代码编写简单,可以算出任意两个节点之间的最短距离。由于三重循环结构紧凑,对于稠密图,效率要高于执行 $|V|$ 次Dijkstra算法。但是时间复杂度比较高,不适合计算大量数据。由于在嵌入式设备上数据量不大,并且Floyd算法相对简单,所以我们在实验平台上实现了Floyd算法。

7 结语

文章针对嵌入式Linux平台下地理信息系统的构建进行了研究,重点剖析了电子地图的设计、地图数据的读取、地图的显示控制以及最短路径算法等关键技术,并在实验平台上得到了验证,可以实现移动设备自主导航定位。在后续研究中,将继续增加地图显示部分的功能,进一步优化最短路径算法。

参考文献

- 1 Jasmin Blanchette, Mark Summerfield. C++ GUI QT4编程. 电子工业出版社,2010,4.
- 2 成洁,卢紫毅.Linux窗口程序设计-Qt4精彩实例分析.清华大学出版社,2008,11.
- 3 苗忠良,宛斌.Qtopia编程之道.清华大学出版社,2009,1.
- 4 <http://www.mapinfo.com.cn/location/integration.html>.
- 5 <http://mitab.maptools.org/>.
- 6 翟艳磊,何祥宇,夏玉杰.嵌入式电子地图设计中MapInfo文件的读取与显示.现代计算机,2010,2:122-125.
- 7 <http://doc.trolltech.com/4.6/porting4.html#qcanvas>.