

Qt/Embedded 应用程序自行设计系统的设计与实现

The Design and the Implementation of QT/Embedded Application Self - Designer System

吴巍荪 邓中亮 (北京邮电大学 电子工程学院 北京 100876)

摘要: 人们对手持终端设备应用的个性化需求越来越高,仅通过程序员编程实现手持机应用程序的开发方式逐渐不能满足人们快速变化的定制需求。针对当前手持终端流行的开发工具 Qt/Embedded,本文提出了一种自行设计系统的解决方案。使用这种方案,用户可以在 Windows 操作系统上,通过鼠标操作和一些简单的数据输入进行个性化应用设计,并将自己设计的应用程序运行在 Linux 环境下的 Qt/Embedded 环境手持终端。最后本文给出了具体的开发实例。

关键词: Qt/Embedded 手持终端设备 应用设计

1 引言

随着嵌入式终端软硬件的发展,高端嵌入式系统被越来越广泛的应用到人们的生活中。Qt/Embedded^[1]作为手持终端中被广泛使用的界面程序开发工具集,也逐渐受到应用程序开发人员的青睐。可是随着用户个性化需求的不断增长,只通过程序员来设计和完成软件应用已经逐渐不能满足用户的需求。如何让用户能够更专注于应用逻辑的设计而又不需要编程,甚至如何引导用户根据自己的想法和需要去设计手持终端的应用仍是人们很关注的问题。

如今,个人电脑用户使用最广泛、最友好的操作系统是 Windows,而基于 Qt/Embedded 开发的应用软件需要的交叉编译环境大都基于 Linux。如果我们能够设计一种基于 Windows 系统的应用设计器,用户只需通过点击鼠标和一些简单的键盘输入就能生成手持机端的应用,并在 Qt/Embedded 环境下运行,那么就会更加方便地实现用户个性化需求了。

本文旨在介绍这样一种在 Windows 下进行用户应用设计,在手持机端 Qt 程序自行解析用户设计的应用并执行的一套系统的解决方案,同时介绍一种针对

基本界面和数据库操作的设计实现用例。

2 系统框架设计

2.1 总体设计

由于当前基于 Qt/Embedded 开发的嵌入式手持终端应用程序所需的交叉编译环境都不是基于 Windows 系列操作系统的,因此,如果在 PC 端的 Windows 操作系统下直接生成可执行的手持端 Qt/Embedded 应用程序,就需要开发 Windows 操作系统下支持该手持终端硬件的交叉编译器。这种编译器的开发难度很大,开发时间周期又长,即使开发出来,也只能针对某一种手持终端硬件设计应用,缺乏通用性。所以,如果需要通过简单可靠地方法来实现用户手持终端 Qt/Embedded 程序的辅助设计,并使设计具有通用性和易移植性,就必须跨过操作系统和硬件平台的差异,在各个操作系统和硬件平台的上层找到共通点来进行设计。

本系统采用文本配置文件的方式来进行应用的信息存储,即用户在 PC 端通过生成器生成描述应用的文本配置文件,然后将此配置文件传输到手持终端上,手

持终端上基于 Qt/Embedded 编写的解析器在运行时会将此文件读取到系统中并解析运行。系统应用总体流程如图 1 所示。

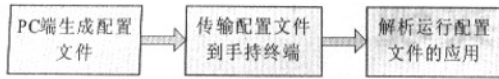


图 1 系统应用总体流程图

根据用户的使用流程,系统应主要包含以下三部分:

- 配置文件
- 解析器
- 生成器

2.2 配置文件

虽然系统应尽可能保持通用性,可凡是应用软件的产品必将有其适用的产品对象,用户所需要进行的各种操作也就可以被归纳和总结。对于手持终端的 Qt/Embedded 应用程序来说,一般的需求都会包括界面的控件状态改变和数据传递,有些应用还会包括嵌入式数据库的操作、外部 IO 的读写等。

归纳总结而来的用户操作通过语言描述被保存到配置文件中。

描述语言的格式可以采用只有机器才能读懂的二进制方式,也可以采用用户和机器都能识别的文本方式。当然,文本的方式会比二进制的的方式浪费存储空间,但一般配置文件都只有几 k,不会太大,而且文本又便于修改调试,所以这里采用文本方式的描述语言。

配置文件的具体描述一定要涵盖用户所需的所有可能操作(即上面总结的用户操作),而且操作所描述的应用必须能被手持终端的 Qt/Embedded 解析器识别并执行(即 Qt/Embedded 的开发库或者外部的其他嵌入 Qt/Embedded 的程序库支持的用户设计的操作)。

2.3 解析器

手持终端的解析器主要实现从配置文件读取配置信息,并将其翻译成 Qt/Embedded 的类、函数的表示形式的功能。解析器的部分设计模块划分可以如图 2 所示。

解析器包含的“配置文件读取”模块,用于在启动程序时一次性从指定路径中读取配置文件到内存中,

尽量避免反复对文件的读写操作。

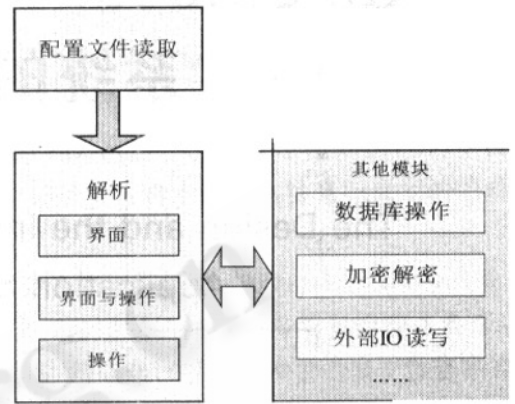


图 2 解析器设计模块图

解析模块分为三部分:界面、界面与操作、操作。界面解析是把配置文件中对于界面控件外观的描述和 Qt/Embedded 的界面控件类相关联;操作解析是把配置文件中对于用户应用操作的逻辑描述和 Qt/Embedded 的相应的类和函数进行对应(有时应用逻辑会用到 Qt 以外的库或者模块也在这部分进行对应);界面与操作解析是把配置文件中表述的界面上的某个操作动作与用户设定的某个操作进行相关联,即用 Qt 的“connect”函数把 Qt 中的“信号”和“槽”连接起来。

很多用户应用仅使用 Qt/Embedded 提供的库是无法实现的,比如我们的手持终端存在嵌入式数据库,要进行数据库的增删查改操作,就需要在程序中嵌入数据库提供的编程的 API,将其封装成模块供解析模块在需要时调用,再比如应用数据要进行加密处理,也可以将加密解密的部分封装成模块,这样的模块可以根据用户的应用来移植添加。

2.4 生成器

用户的配置文件格式定好后,需要在 PC 端设计生成器软件来引导用户进行配置文件的生成。

如图 3 所示,生成器的模块完全对应配置文件的格式,主要包含如下功能:

- 通过界面设计器的鼠标拖拽设定控件在窗体的位置及大小信息的记录
- 通过事件设计器设计用户使用手持终端点击按钮时执行各个事件的记录
- 窗体文件的保存与打开
- 通过合成器将已经生成的多个窗体信息文件

合成最终的配置文件

对于有数据库应用的系统,还应该加入数据库管理系统来对数据库进行管理。

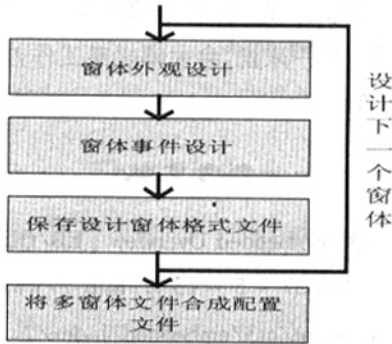


图 3 生成器的应用流程

3 系统实现

基于以上的系统设计,下面介绍一种手持终端带数据库操作的应用系统。用户可以在 PC 机端的生成器上实现手持机应用的界面设计、操作设计、数据库新建及管理、配置文件生成等操作,以及生成手持终端解析用到的配置文件和数据库。用户将配置文件和数据库传送到手持终端,可以在手持终端直接解析运行用户定制的逻辑。

3.1 配置文件方案

本系统为用户设计了包括 Pushbutton、Label、Table、Lineedit、Textedit、Combobox、Radiobutton、Checkbox、Groupbox、Spinbox、Listbox 和 Picbox 12 种界面控件。

归纳总结而来的用户需求操作包括:查询数据库并显示、插入到数据库、修改数据库内容、删除数据库内容、打开新的窗体、关闭当前的窗体、表项上移、表项下移、清空控件数据、删除某一表项、空间数据传递、弹出警告、逻辑 if else、控件显隐设定、控件可用否设定、控件数值加法、控件数值减法、单/复选框选中设置等用户操作。

配置文件采用文本模式。采用“\n”回车符分割各个配置语句。总体上配置文件分为 4 部分:系统信息、界面控件信息、界面控件引起事件信息和事件描述信息,每部分都有开始和结束的标识符。

“系统信息”占据配置文件的第一行,记录的信息

内容包括数据库中的数据是否采用加密、加密的密钥等系统应用信息。这里为了安全考虑,文件中保存的密钥也是经过加密的(加密算法采用 AES)。

“界面控件信息”保存的是各个用户设计的控件的属性信息和窗体信息。因为用户的设计可能是多窗体的,即可能是对一个主窗体进行操作来打开新的窗体,所以这部分的控件属性都有一项来描述自己归属的窗体。窗体的信息包括窗体的 ID、窗体的标题和窗体的父窗体。控件的属性各有不同,举个 Pushbutton 控件的例子:

格式:父窗体 ID 类型 ID 位置及大小(左 顶 宽 高) 显隐 显示名称 能用否

举例: DID1 Pushbutton BUTTON1 66 66 66 66 Y 确定 N

“界面控件引起事件信息”保存的是引起事件的控件 ID 和引起的事件的编号,及其所归属的窗体,即在某个窗体的某个控件引起了什么事件。这里的“什么事件”只保存编号,具体内容在后面的部分进行描述。

“事件描述信息”就是描述控件引发事件后要进行的处理。这里为每个窗体预制了 10 个事件,每个事件可以是上面需求操作归纳的 18 种操作的任意组合。这部分中每个事件也都有自己的开始和结束标识符。

3.2 解析器的实现

根据上述的解析器的设计思路和设计的配置文件格式,我们在 XScale PXA270 的手持硬件终端上的 Linux 操作系统中实现了 Qt/Embedded 应用程序的解析。实现的过程分为以下几步:

- 编译移植 Qt/Embedded 环境以及 Qtopia 图形界面环境到手持机终端^[2]
- 交叉编译 SQLite^[3]嵌入式数据库
- 基于以上的 Qt/Embedded 和 SQLite 库编写解析器程序

实现后将配置文件导入手持终端指定目录位置,运行解析器即可运行用户设计的应用。

3.3 生成器的实现

根据上面描述的设计思路和配置文件格式,在 Windows XP 操作系统环境下用 Visual C++ 2003 编写生成器程序。

生成器采用微软 MFC 的 MDI 多视实现,每个文档保存着一个应用窗体的控件信息以及事件信息的记

录,并保存为临时文件。菜单和快捷工具栏中的“配置文件合成器”按钮用于打开“配置文件合成器”,来合成最终的应用配置文件。生成器程序的主界面实现效果如图 4。

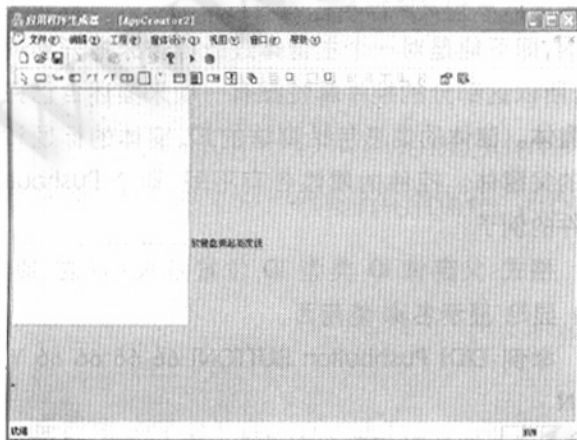


图 4 生成器效果图

4 结束语

本文提出了一种 Qt/Embedded 界面应用程序设计器的系统解决方案,并实现了包含基本控件和数据库应用的简单设计器。这种解决方案对于需要经常改动界面应用的用户来说,会大幅提高工作效率,具有很高的实用价值。

参考文献

- [s. n.]. Qt/Embedded Overview [EB/OL]. (2006 - 10 - 27). [2007 - 5 - 13]. <http://www.trolltech.com/products/qt/qt3/embedded/index/?searchterm=Embedded>.
- 倪继利,Qt 及 Linux 操作系统窗口设计[M],北京:电子工业出版社,2006.
- [s. n.]. About SQLite [EB/OL]. (2007 - 04 - 25).