

嵌入式 Linux 下的 Qt 通讯簿程序设计

随着嵌入式系统的发展,在 Linux 下使用 Qt 开发嵌入式应用程序是嵌入式图形用户界面 (GUI) 系统应用研究的重点。通过一个通讯簿程序的设计,给出了基于 Qt 的嵌入式 Linux 程序开发的关键技术,包括 QtDesigner、信号与槽机制的使用、程序主要功能代码的实现以及程序开发的整体过程。

Linux 操作系统具有开放的源码、高效稳定的内核、良好的开发环境以及支持多种硬件平台的特点,且具有可配置性和模块化的特点; Qt 是一个面向对象的支持多操作系统平台的应用程序开发框架,它采用了一种独有的 Signal2Slot (信号与槽) 的对象间通信机制,以代替传统的不安全的回调技术。Qt 的自由版开放了源代码,因此开发人员在 Linux 下使用 Qt 进行嵌入式系统开发是非常理想的选择。

从目前国内嵌入式系统 (Embedded system) 的开发情况来看,在 Linux 下使用 Qt 来开发嵌入式 GUI 程序还不是很多,且相关中文资料也较少。

究其原因,主要是由于 Qt 历史较短,其比较稳定的版本是在 2002 年之后才逐渐被国内工程人员使用,虽然功能强大,但中文文档匮乏,延缓了在国内的普及。另外一个主要原因是虽然嵌入式技术随着半导体技术的进步得到长足的发展,但较为成熟的嵌入式操作系统及支持嵌入式操作系统的平台是在近几年才逐步发展起来,而 Qt 需要在操作系统的支持下才可使用。文献 [2] ~ [4] 从不同的方面阐述了 Linux 下使用 Qt 进行嵌入式开发的一般过程和特点,但没有对其关键技术及程序开发实现过程进行具体的说明。为此,笔者通过通讯簿程序的设计,从工程文件的建立到可执行程序的生成,详细给出了其中关键技术的实现及关键代码的运用。

1 基于 Qt 的嵌入式 Linux 通讯簿程序设计

在嵌入式开发之前,首先要搭建好开发环境。正确设置编译器、Qt 及 Qt E 环境变量对后续程序编译过程至关重要。本设计使用的嵌入式处理器为三星公司的基于 ARM 920 T 的 S3C2410X 芯片,最高工作频率 203 MHz. 操作系统为 REDHAT L INUX 910 , 内核版本 214118. 串口通信程序为 MINICOM , 使用 Troll Tech 公司发布的自由版: qt2x1122. 3. 2 , qt2embedded22. 3. 10 , 使用 arm2linux2gcc231411 编译器。

通讯簿程序主要记录联系人的姓名、电话、邮件地址及地址信息,具有新建、编辑、删除、查询联系人信息等功能。

1.1 生成工程文件

通常一个应用程序对应一个工程。工程文件的作用是管理当前项目中所有的文件及其关系,可以使用 progen 工具来生成程序的工程文件:

```
progen -t appl t -o AddressBook. pro
```

生成的 AddressBook1pro 工程文件并不完整，还需后续添加工程所包含的头文件、源文件等信息。

1.2 使用 QtDesigner 设计程序窗体

QtDesigner 是 Qt 的一个可视化的程序界面设计工具，使用它来设计程序的界面非常方便，从而开发者可以专注于程序功能的实现。通讯簿主窗口包括联系人信息查询、新建、编辑及删除功能部件。查询功能使用 2 个 TextLable, 1 个 LineEdit, 1 个 ComboBox 及 1 个 PushButton 部件，其中 ComboBox 可以设置 NAME, Telephone, Email, Address 4 种查询方式。新建联系人可以单独弹出新建窗口来实现，使用 4 个 TextLable 及 4 个 LineEdit 部件。编辑和删除联系人由 1 个 ListView 和 2 个 PushBut ton 部件构成，其中 ListView 包括 1 个 List 和 4 个 Column (包括 Name, Telephone, Email, Address)。

在 Qt 中，“事件驱动”通过 Signal2Slot (信号和槽) 机制实现。Signal 主要是各可视化组件产生的事件，这些事件通常由 Qt 控件预定义，如按钮单击事件 clicked 等。Slot 是对某个事件的处理方法，它也属于类的成员函数，其编写和普通成员函数类似。当需要处理某组件的 Signal 事件并要求把定义好的 Slot 作为响应事件的方法时，只要在实现文件中使用 connect 函数把 Signal 和指定的 Slot 连接起来即可。当对象改变其状态时，信号被发送，对象不关心有没有其它对象接收到这信号，可以将信号和槽通过 connect 函数任意相连。信号与槽的连接原理如图 1 所示。

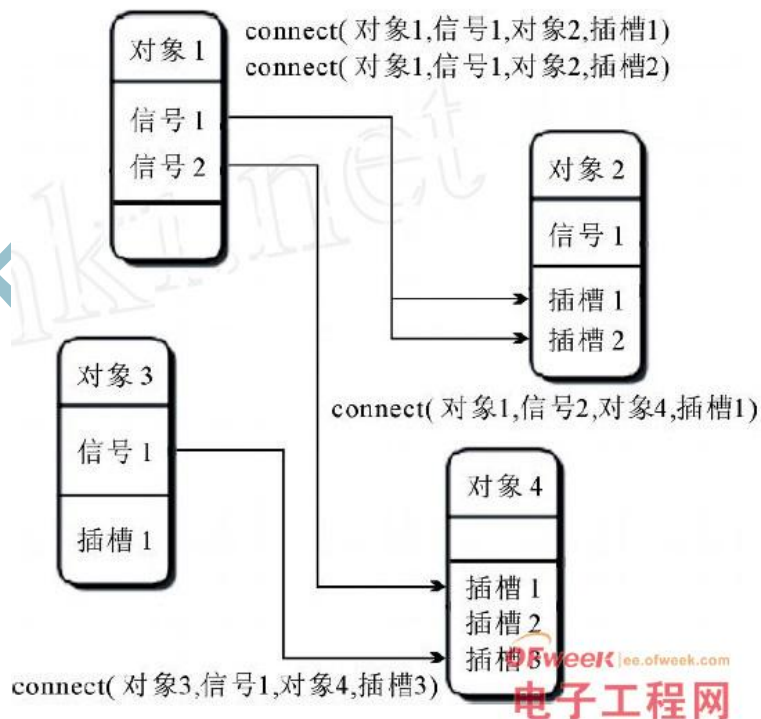


图 1 信号与槽连接抽象图

以下是通讯簿程序主窗口 5 个按钮的 Signal2Slot 的 connect 代码:

```
connect (add, SIGNAL (clicked ( ) ) ,this, SLOT(addSlot) ) ) ;  
connect (edit, SIGNAL (clicked ( ) ) ,this, SLOT(editSlot) ) ) ;  
connect (del, SIGNAL (clicked ( ) ) ,this, SLOT(delSlot) ) ) ;  
connect (exit, SIGNAL (clicked ( ) ) ,this, SLOT(exitSlot) ) ) ;  
  
connect ( search , SIGNAL (clicked ( ) ) ) , this , SLOT  
( searchSlot) ) ) ;
```

窗口(如图 2 及图 3 所示)设计完成后,将通讯簿主窗口及新建联系人窗口分别保存为 AddressBook. ui 和 New. ui 界面文件。

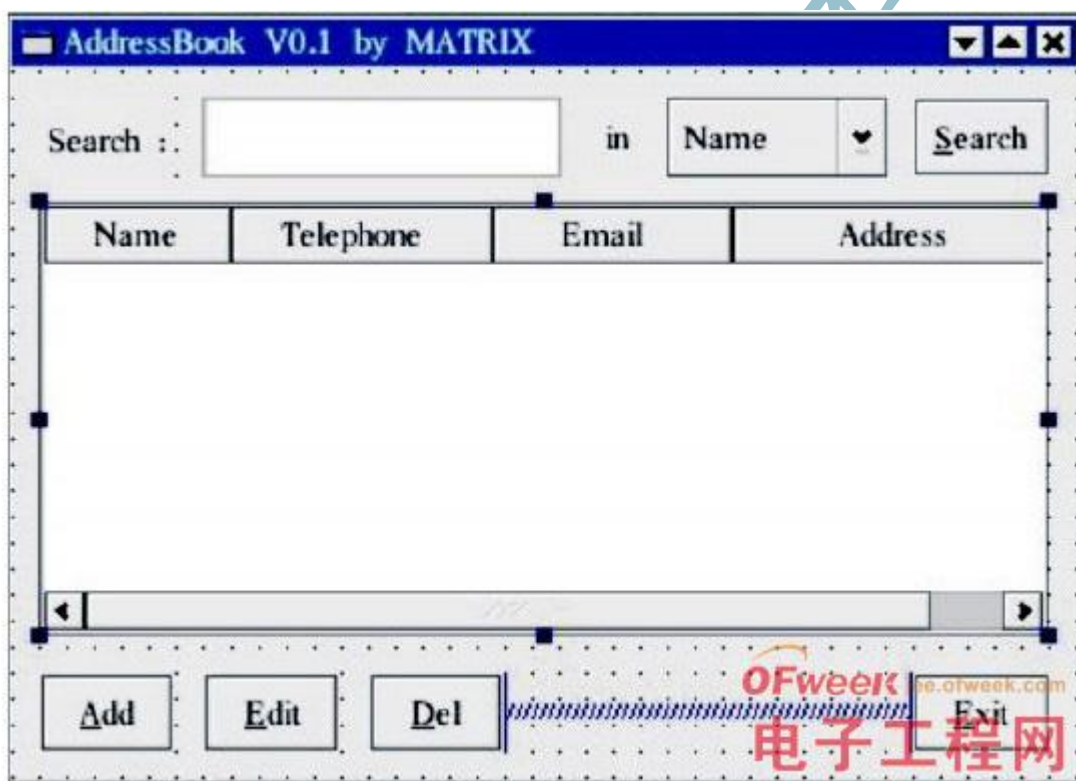


图 2 通讯簿主窗口

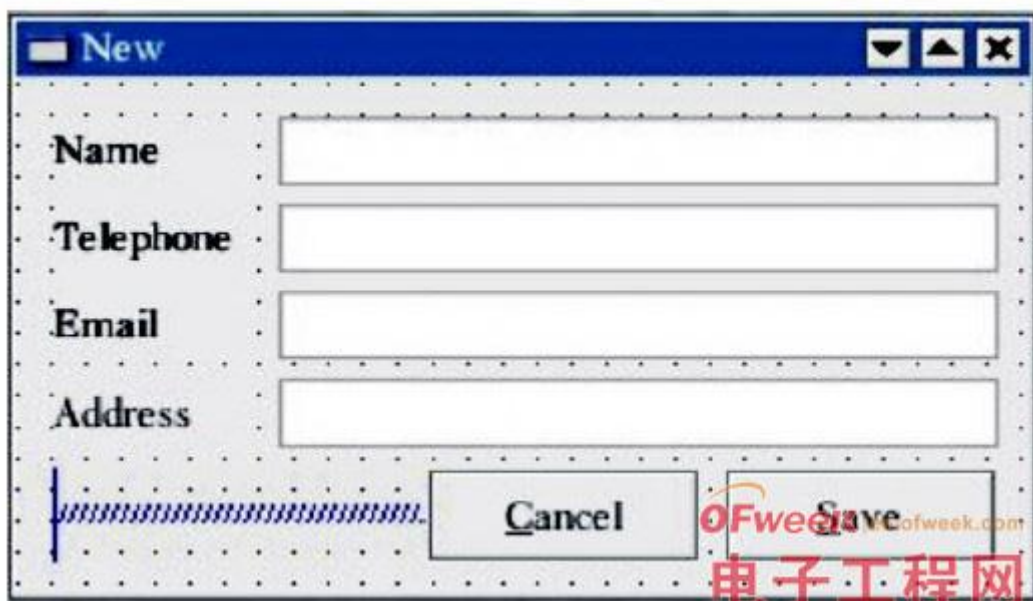


图3 新建联系人窗口

1.3 生成窗体类的头文件和源文件

使用 Qt 提供的 uic 工具通过界面文件 AddressBook.ui 来生成 AddressBook.h 文件和 AddressBook.cpp 文件（同样，使用 uic 工具通过界面文件 New.ui 来生成 New.h 和 New.cpp 文件）：

```
uic -o AddressBook.h AddressBook.ui
uic -o AddressBook.cpp -impl AddressBook.h
AddressBook.ui
```

通讯簿要实现新建、编辑、删除、查询联系人信息等功能还需要在 AddressBook1.cpp 文件里添加相应的功能实现代码。

为实现新建联系人功能，在文件 AddressBook1.cpp 中定义一个 AddressBook 类的成员函数：

```
void AddressBook::addSlot ()
```

新建联系人信息包括姓名、电话、邮件及地址，在函数中定义一个字符串类型函数：

```
QString lSQL St ring
```

并给 lSQL St ring 赋值：

```
lSQLSt ring = " Create Table " + Const : : DBTBL _ADDRESS + " (" +
```

```
Const::DBCOL_NAME + ", " + Const::DBCOL_TEL EPHONE + ", " +  
Const::DBCOL_EMAIL + ", " + Const::DBCOL_ADDRESS + ", " + "  
";
```

编辑联系人信息时，定义成员函数：

```
void AddressBook::edit Slot ()
```

分别在姓名、电话、邮件及地址栏中输入信息：

```
ent ry -> setName (lRows[0 ]) ; // 在第 1 行 0 列输入 name  
ent ry -> set TelePhone (lRows[1 ]) ; // 在第 1 行 1 列输入 telephone  
ent ry -> set Email (lRows[2 ]) ; // 在第 1 行 2 列输入 email  
ent ry -> setAddress (lRows[3 ]) ; // 在第 1 行 3 列输入 address
```

删除联系人信息时，定义成员函数：

```
void AddressBook::delSlot ()
```

选择要删除的联系人，从数据库删除该联系人信息：

```
st ring lSQLSt ring ;  
lSQLSt ring = " Delete f rom " + Const::DBTBL_ADDRESS + " where  
" +  
Const::DBCOL_NAME + " = " + aName ;
```

查询联系人时，定义成员函数：

```
void AddressBook::searchSlot ()
```

在查询框中输入联系人的姓名、电话、邮件及地址信息之一，选择相应的查询方式：

```
st ring lSQLSt ring = " Select " + Const::DBCOL_NAME + ", "  
+ Const::DBCOL_TEL EPHONE + ", " + Const::DBCOL_EMAIL + ", "  
+ Const::DBCOL_ADDRESS + " f rom " + Const::DBTBL_ADDRESS  
+ " where " + Const::DBCOL_NAME + " = " + aName ;
```

受篇幅限制, AddressBook1.h, New1.h, New1.cpp, SQL1.cpp, SQL1.h 代码不再给出。

1.4 编写主函数

每一个 Qt 应用程序都包含一个主函数 main1.cpp。主函数是应用程序执行的入口点。以下是 AddressBook 的主函数文件 main1.cpp 的关键代码:

```
int main (int argc , char * argv)
{
    QApplication AddressBook ( argc , argv ) ;// 创建 QApplication 对象, 管理整个程序资源

    AddressBook * AddressBook = new AddressBook (log) ;// 创建程序主窗口

    app . setMainWidget ( mainWin ) ;// 设置 mainWin 部件为程序的主部件

    mainWin - > show ( ) ;// 使 mainWin 部件可视

    return AddressBook . exec ( ) ;// 把程序控制权交还给 Qt
}
```

1.5 编辑工程文件

代码编写完成后, 需要把文件 AddressBook1.cpp, AddressBook1.h, New1.h, New1.cpp, SQL1.h 及 SQL1.cpp 添加到工程文件 AddressBook1.pro 中:

```
HEADERS = AddressBook. h New. h SQL. h
```

```
SOURCES = AddressBook. cpp New. cpp SQL. cpp main. cpp
```

1.6 生成可执行文件

编译器是根据 Makefile 文件内容来进行编译的, 所以需要生成一个 Makefile 文件。Qt 提供的 qmake 工具可以从一个工程文件 (.pro 文件) 中产生 Makefile 文件, 然后再使用 make 命令对整个工程进行编译链接:

生成的二进制文件 AddressBook 就是可以直接运行的可执行文件。程序运行结果如下面图示, 图 4 是按照姓名 (Name) 查询联系人 Matrix 的信息, 图 5 是编辑新联系人 Matrix 的信息。



图 4 按姓名查询联系人



图 5 编辑新联系人信息

程序编译成功后可先在 Virtual framebuffer (虚拟缓冲帧) 中运行, 确证无误后再交叉编译在开发板上运行。交叉编译时要确保 \$QTEDIR/lib 下有需要在开发板上运行的库文件。

2 结论

所设计的通讯簿程序具有对联系人的姓名、电话、邮件地址及通讯地址等信息的记录、新建、编辑、删除、查询等功能, 且可按照姓名、电话、邮件地址及通讯地址 4 种查询方式来检索联系人。程序的设计很好地体现了 Qt 的信号与槽机制, 并且使用 QtDesigner 可方便地进行可视化程序界面设计, 通过使用 Qt 中丰富的类函数, 利用其面向对象和模块化的特征, 避免了繁琐的 X 编程。通讯

簿程序的设计对在 Linux 下使用 Qt 开发其他嵌入式 GUI 程序具有一定的指导意义。

OFweek 电子工程网