

嵌入式 Linux 操作系统

第五讲 嵌入式图形界面编程

杨延军

yangyj.ee@gmail.com

北京大学

2016 年

主要内容

- 1 图形界面简介
- 2 QT 编程环境简介
- 3 交叉编译 Qt 简介

编写图形界面程序

图形界面的好处

- 操作直观，用户容易上手

图形界面程序设计的不同之处

- 绘制图形界面，标准库
- 处理用户鼠标和键盘事件

常见的图形界面库

- Xlib
- QT , KDE 的基础
- Gtk , Gnome 的基础
- wxWindow
- tcl/tk
- 嵌入式的图形环境 : QTE , DirectFB , miniGUI , Tiny-X

Framebuffer 简介

- 相当于显示缓冲区
- 直接对内存操作，效率更高
- Linux 内核有对 FrameBufffer 的直接支持
- 大部分嵌入式图形接口都是直接对 FrameBuffer 操作

QTE 简介

- QT 的嵌入式版本，为 QT 的子集。
- 跨平台性好，支持多种 CPU
- 基于 C++，结构合理，效率较高
- 类库完整，界面美观
- 接口友好，编程容易上手
- 包含 qvfb 作为调试界面，开发方便

miniGUI 简介

- 中国人的优秀开源项目
- 接口和 win32 类似，Windows 程序员容易上手
- 对中文的支持好，包含输入法
- 轻量级，专为嵌入式系统设计

GTK 简介

- The GIMP(GNU Image Manipulation Program) Toolkit
- 包含三个部分
 - GDK——Xlib 的低层封装库
 - GTK——面向对象的类库，包含很多成熟控件
 - Glib——与 GUI 无关的常用函数库
- Glade 界面辅助设计工具

主要内容

1

图形界面简介

2

QT 编程环境简介

3

交叉编译 Qt 简介

QT 编程环境

- 桌面环境
- 仿真环境
 - qvfb：虚拟的 FrameBuffer
 - 对程序做界面调整和调试
- 嵌入式环境
 - 交叉编译并下载到目标板
- Qt Quick：qml 编程环境

QT 版本

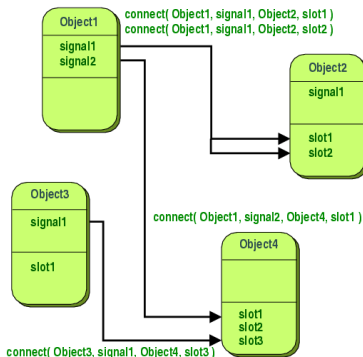
- Qt 是所有 Qt 软件版本的总称，例如 qt-everywhere-opensource-4.6.3
- Qt 软件分为商用版和开源版，开源版的名称中一般带有 `opensource` 或者 `free` 单词
- Qt 软件包含面向不同平台的发行版，如 win/x11/mac/wince 等
- Qt 的嵌入式版商业上曾经被称为 Qt/Embedded，Qtopia Core 和 Qt Embedded
- Qt 的 win/x11/mac/embedded 版本最后融合到了一起，被称为 Qt Everywhere。

Qt Everywhere

- Qt 不仅是一个 GUI 库，已经是完整的跨平台解决方案
 - Qt Network，网络通信类库
 - Qt OpenGL，3D 图形引擎
 - Qt SQL 数据库引擎
 - Qt Quick
 - ... many more
- Qt Linguist 多语言支持
- Qt Resource 资源管理

信号与槽的概念

- 信号与槽可以灵活的连接
- 严格的数据类型检验
- 对 C++ 进行扩展，无缝的支持信号与槽的机制



第一个 Qt 程序

```
#include <QApplication>
#include <QPushButton>

int main(int argc, char *argv[])
{
    QApplication app(argc, argv);

    QPushButton hello( QObject::tr("Hello World!"), 0 );
    hello.resize( 100, 30 );

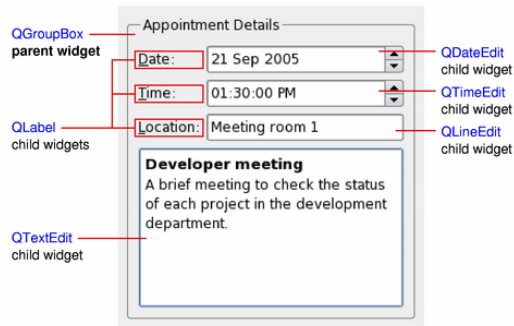
    hello.show();
    return app.exec();
}
```

编译：

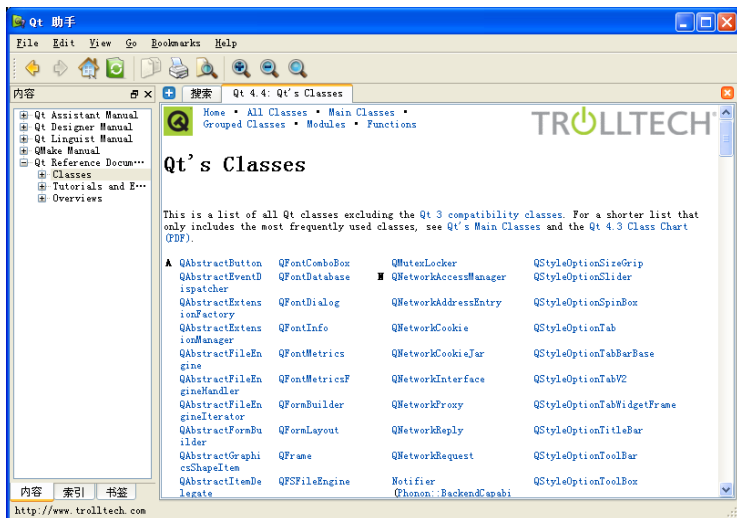
```
qmake -project ——生成 pro 文件
qmake      —— 生成 Makefile
make       —— 生成可执行文件
```

Widget

- 图形程序的基本元素
- 可以用来显示内容和接受输入
- 基类为 QWidget

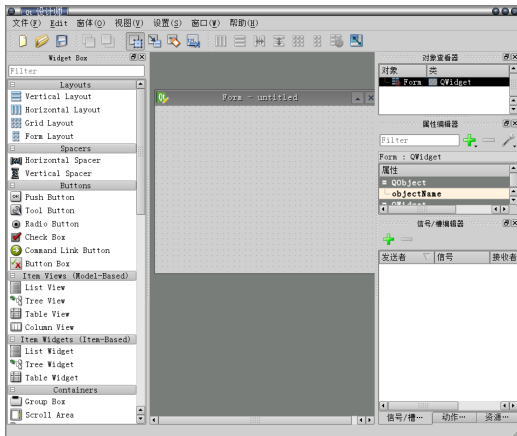


从 assistant 获取帮助



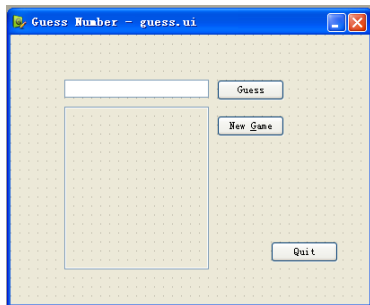
使用 Qt Designer

- 图形化的界面设计工具
- 界面被保存为 ui 文件，可以通过 uic 工具编译为标准 C 代码



QT Designer 示例

- 三个按钮
Pushbutton
- 一个输入框 QLineEdit
- 一个文本框 Label



Ui 文件

```
<ui version="4.0" >
  <author></author>
  <comment></comment>
  <exportmacro></exportmacro>
  <class>GuessNum</class>
  <widget class="QWidget" name="GuessNum" >
    <property name="geometry" >
      <rect>
        <x>0</x>
        <y>0</y>
        <width>400</width>
        <height>300</height>
      </rect>
    </property>
    <property name="windowTitle" >
      <string>Guess Number</string>
    </property>
    .....
```

Ui.h 文件

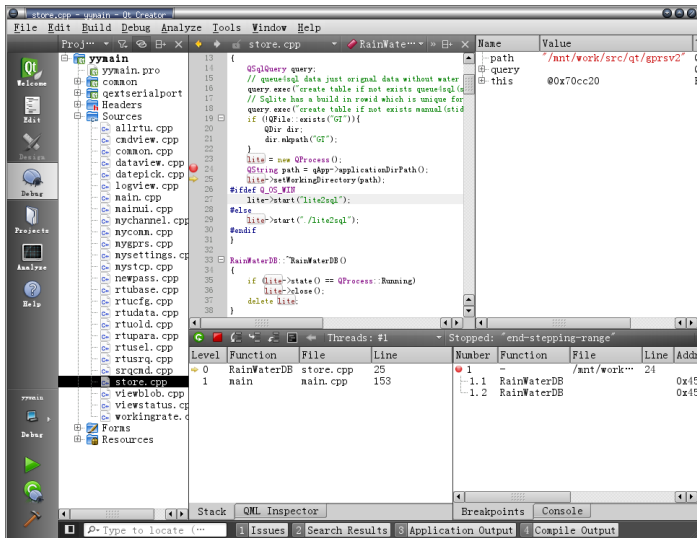
```
class Ui_GuessNum
{
public:
    QPushButton *quit;
    QLabel *label;
    QPushButton *guess;
    QPushButton *newgame;
    QLineEdit *num;

    void setupUi(QWidget *GuessNum)
    {
        GuessNum->setObjectName(QString::fromUtf8("GuessNum"));
        GuessNum->resize(QSize(400, 300).expandedTo(GuessNum->minimumSizeHint()));
        quit = new QPushButton(GuessNum);
        .....
```

需要注意的问题

- ui 文件的使用方法
 - 直接使用 ui 类
 - 继承 ui.h 文件中的类
 - 多重继承
- 自定义信号和槽
- 类定义要使用单独的头文件
- 源代码要放到单独目录中

集成开发环境



主要内容

1

图形界面简介

2

QT 编程环境简介

3

交叉编译 Qt 简介

交叉编译环境

- 把光盘中提供的工具解压到 `/usr/local` 目录

组件

- `qmake` `moc` 等工具及配置文件
- 交叉编译所需要的库文件

编译过程

- 编译相关的依赖软件，如 `tslib` 和 `libjpeg`
- 配置编译 Qt 自身代码

交叉编译应用程序

- 仍然使用 qmake 生成 Makefile
 - 使用新工具链中的 qmake
 - .pro 文件没有变化

在开发板环境中运行

- 可能需要设置如下环境变量

```
export TSLIB_TSDEVICE=/dev/input/event1
export QWS_MOUSE_PROTO="tslib:/dev/input/event1"
export QWS_DISPLAY=Transformed:Rot90:0
```

其他可能涉及的问题

- Qt 的资源文件
- Qt 网络编程
- 将数据文件（资源）编译到可执行文件中
- Qt 程序的国际化
- Qt Quick

参考资料

- 《C++ GUI Programming with QT 4》
- assistant
- 网络资源