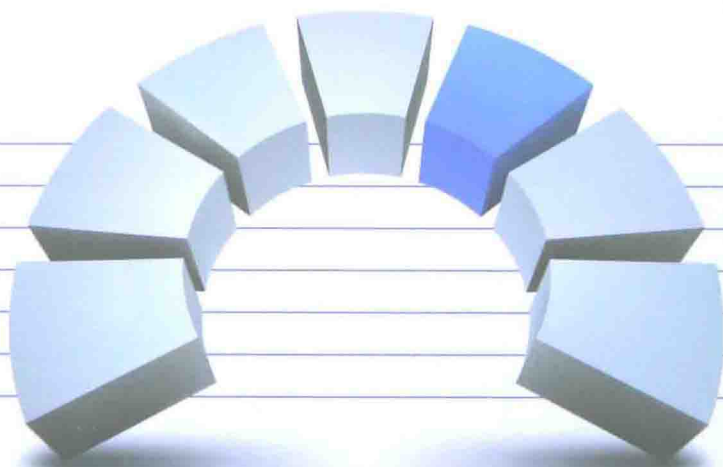


嵌入式

Qt

实战教程

深圳信盈达电子有限公司 陈志发 王苑增◎编 著



中国工信出版集团



电子工业出版社
PUBLISHING HOUSE OF ELECTRONICS INDUSTRY
<http://www.phei.com.cn>

信盈达技术创新系列图书

嵌入式 Qt 实战教程

深圳信盈达电子有限公司 陈志发 王苑增 编著

电子工业出版社

Publishing House of Electronics Industry

北京·BEIJING

内 容 简 介

本书是配合 Qt GUI 程序设计的指导教材，独立于任何 Qt 编程教科书。主要内容有常用 GUI 介绍、Qt 开发工具的安装及其优点、Qt 入门、窗口的布局、信号和槽机制、常见对话框的使用、主窗口（QMain-Window 类）的使用、菜单和工具条的实现、鼠标和键盘事件、Qt 进程与线程、网络编程、互斥锁等。最后还详细讲解了 Qt 的一些应用小项目，并列出了全部源代码，如图片浏览器、MP3 播放器、网络视频播放器、门禁机项目。

通过这些内容，本书涵盖了 Qt 编程的基础知识，并且讲解清晰易懂。通过详细的示例，从零开始、由浅入深、循序渐进地指导初学者掌握 Qt GUI 编程，培养实际分析问题和编程的能力，提高读者和学生的综合素质。

本书适合作为高等院校计算机专业的 Qt 编程教科书，也可为广大专业编程人员提供参考。

读者可登录华信教育资源网（www.hxedu.com.cn）免费注册后下载本书相关源代码。

未经许可，不得以任何方式复制或抄袭本书之部分或全部内容。

版权所有，侵权必究。

图书在版编目（CIP）数据

嵌入式 Qt 实战教程/深圳信盈达电子有限公司，陈志发，王苑增等著. —北京：电子工业出版社，2015.4

（信盈达技术创新系列图书）

ISBN 978-7-121-23027-1

I. ① 嵌… II. ① 深… ② 陈… ③ 王… ④ 软件工具—程序设计—教材 IV. ① TP311.56

中国版本图书馆 CIP 数据核字（2014）第 080408 号

责任编辑：田宏峰

印 刷：北京京师印务有限公司

装 订：北京京师印务有限公司

出版发行：电子工业出版社

北京市海淀区万寿路 173 信箱 邮编 100036

开 本：787×1 092 1/16 印张：14.5 字数：368 千字

版 次：2015 年 4 月第 1 版

印 次：2015 年 4 月第 1 次印刷

印 数：3 000 册 定价：49.00 元

凡所购买电子工业出版社图书有缺损问题，请向购买书店调换。若书店售缺，请与本社发行部联系，联系及邮购电话：（010）88254888。

质量投诉请发邮件至 zlt@phei.com.cn，盗版侵权举报请发邮件至 dbqq@phei.com.cn。

服务热线：（010）88258888。

近年来，Linux 操作系统的使用越来越广泛，尤其是嵌入式领域。Linux 系统离不开图形应用程序。诺基亚收购的奇趣公司为开发人员提供了一整套跨平台的开发工具包 Qt，我们可以用它很好地开发 Linux 下的图形用户界面程序，而且 Qt 界面程序也可以顺利地移植到其他系统，甚至是嵌入式操作系统上，并且用 Qt 开发出的图形用户界面（GUI）程序非常美观，其美观程度甚至可以达到艺术级。由于 Qt 跨平台的特性，我们可以实现一次编写代码，到处编译，就可以使用。如果客户要求应用程序能同时运行于不同的平台时，Qt 的这一优势就非常重要了。总之，Qt 具有很大的使用价值和发展潜力。

Qt 官方提供了大量的文档，但这些文档由于太详细、太全面，而内容太多，我们通常没有时间详尽地阅读它们，但为了掌握相关知识又不得不查阅这些文档。这些帮助文档以参考性的内容为主，其中的示例程序虽然非常有用，但仅仅通过这些有限的示例程序显然难以正确使用 Qt，所以需要一本通俗易懂的介绍 Qt 的图书。

本书是一本系统地介绍 Qt 图形界面编程知识的书籍，涵盖了 Qt4 程序开发中常用的类和 API 函数，足够广大开发人员参考使用。

国内关于讲解 Qt 图形界面编程知识的书籍比较少，国外相关著作虽然全面系统，但讲解风格却依照西方的思维方式，过于专业和深奥，这与中国人的思维方式有些不同，所以我们在阅读那些书难免会感到不习惯，影响学习效果。因此，国内读者需要一本按照国人容易接受的方式去讲解 Qt 编程知识的参考书。本书正是按照这种思维方式来编写的，其讲解清晰易懂，内容科学合理，有助于初学者迅速掌握知识体系和精髓。在相同的学习时间下，可以收到更好的学习效果。

本书适合广大的计算机编程人员学习，只要读者具备一定的 C++ 语言编程基础即可。即使没进行过图形用户界面（GUI）程序设计的人，阅读本书也会迅速入门。本书知识点的顺序和示例程序都是精心安排的，清晰地表达出了重点，其内容涵盖面广，语言简练而清晰，能让读者迅速掌握 Qt 基础知识和精髓，适合读者自学。

本书首先介绍了常用 GUI，Qt 开发工具的安装及其优点；然后讲述了 Qt 入门，窗口的布局，信号和槽机制，常见对话框的使用，主窗口的使用，菜单和工具条的实现，鼠标和键盘事件，Qt 进程与线程，网络编程，互斥锁等常见内容；最后还详细讲解了 Qt 的一些应用小项目，并列出了全部的源代码，如图片浏览器、MP3 播放器、网络视频播放器、门禁机项目等，让读者很容易上手，初学者和专业开发人员都能从中获益。

本书的出版，离不开深圳信盈达电子有限公司所有同事们的支持和帮助，尤其是牛乐乐和陈志发、黄英涌等工程师，在此向他们表示衷心的感谢。另外，感谢我的父母亲人和朋友，是他们给予我精神上的支持和鼓励。感谢电子工业出版社，是他们认真专业的审核，让本书由粗糙的初稿变成了精美的图书。

由于时间仓促，编者水平有限，书中难免有不恰当的地方，希望广大读者批评指正，联系邮箱：niusdw@163.com，欢迎来信交流。

陈志发
2015年3月

第 1 章 嵌入式 GUI 简介与 Qt 概述	1
1.1 常用 GUI 介绍	1
1.1.1 MiniGUI	2
1.1.2 MicroWindows	2
1.1.3 DinX	2
1.1.4 OpenGUI	3
1.1.5 Qt/Embedded	3
1.2 Qt 简史	4
1.3 Qt 程序设计简介	6
1.4 Qt 版本介绍	6
1.5 怎样学习 Qt	7
1.5.1 从 C++ 到 Qt 的跨越	7
1.5.2 C 语言和其他程序设计的关系	7
1.6 Qt 开发工具的优点	8
1.7 Qt/Embedded 介绍	9
1.7.1 架构	9
1.7.2 Qt 的开发环境	11
1.7.3 Qt 的支撑工具	11
1.8 Qt 的安装	11
1.8.1 Linux 下 Qt 的安装	12
1.8.2 Windows 下 Qt 的安装配置	23
第 2 章 Qt 编程基础	26
2.1 程序设计语言介绍	26
2.2 C++ 语法基础知识	28
2.3 C++ 高级应用——虚函数	34
第 3 章 Qt 模块与 Qt 功能类	37
3.1 Qt 常用模块	37
3.2 Qt 部件与组类说明	37
3.3 Qt 基于图形模式部件类	38
3.3.1 窗口与窗口相关部件	39
3.3.2 布局管理	40
第 4 章 Qt 入门	42
4.1 Hello Qt	42

4.2	建立连接	43
4.3	窗口的布局	44
4.4	编译	46
4.5	Qt Creator 编写多窗口程序	53
第5章	对话框	60
5.1	深入信号和槽	60
5.2	派生对话框类	64
5.3	Qt 中常见的对话框及应用举例	68
5.4	Qt Creator 登录对话框应用举例	71
5.5	快速设计对话框	76
5.6	能够改变的对话框	80
第6章	创建主窗口	83
6.1	继承 QMainWindow 类	83
6.2	创建菜单和工具条	86
6.3	创建状态条	90
6.4	实现文件菜单	92
6.5	使用对话框	98
6.6	存储设置	103
6.7	多文档	104
6.8	一个创建主窗口及各种控件的示例	107
第7章	鼠标和键盘事件	112
第8章	Qt 线程及网络编程	118
8.1	早期的 IPC 方法——信号与管道	118
8.1.1	信号机制	118
8.1.2	管道	119
8.2	基于平台的线程 QThread	120
8.2.1	线程的创建	120
8.2.2	线程同步	126
8.2.3	线程互斥锁	128
8.3	Qt 中的网络编程	131
第9章	XML 和数据库	136
9.1	XML	136
9.1.1	DOM	136
9.1.2	SAX	146
9.1.3	XML 流	150
9.2	数据库	153
9.2.1	连接到数据库	154
9.2.2	执行 SQL 语句	159
9.2.3	使用 SQL 模型类	163

第 10 章 Qt 应用举例	172
10.1 Qt 图片浏览器	172
10.2 MP3 音乐播放器	180
10.3 视频监控	190
10.3.1 什么是 V4L2	190
10.3.2 常用的结构体	190
10.3.3 视频采集的基本流程	192
10.4 Qt 门禁机项目	210
10.4.1 可视门禁项目的说明	210
10.4.2 项目代码	210
参考文献	221

嵌入式 GUI 简介 与 Qt 概述

1.1 常用 GUI 介绍

所谓 GUI (Graphics User Interface), 就是指图形用户界面。图形用户界面的广泛流行是当今计算机技术的重大成就之一, 它极大地方便了非专业用户的使用, 人们不再需要死记硬背大量的命令, 而可以通过窗口、菜单方便地操作。它的主要特征有四点: WIMP。其中, W (Windows) 指窗口, 是用户或系统的一个工作区域, 一个屏幕上可以有多个窗口; I (Icons) 指图图标, 是形象化的图形标志, 易于人们隐喻和理解; M (Menu) 指菜单, 可供用户选择的功能提示; P (Pointing Devices) 指鼠标等, 便于用户直接对屏幕对象进行操作。GUI 采用了不少 Desktop 桌面办公的隐喻, 让使用者共享一个直观的界面框架。由于人们熟悉办公桌的情况, 因而对计算机显示的图标的含义容易理解, 诸如文件夹、收件箱、画笔、工作簿、钥匙及时钟等。过去的界面不仅需要记忆大量命令, 而且需要指定操作对象的位置, 如行号、空格数、X 及 Y 的坐标等。采用 GUI 后, 用户可直接对屏幕上的对象进行操作, 如拖动、删除、插入, 以及放大和旋转等。用户执行操作后, 屏幕能立即给出反馈信息或结果, 称为所见即所得 (What You See Is What You Get, WYSIWYG)。用视、点 (鼠标) 代替了记、击 (键盘), 给用户带来了方便。通常所见的 GUI 都是位于 PC 上的, 但是在 PC 上 GUI 并不适合嵌入式系统。嵌入式设备有严格的资源要求 (如十分有限的存储空间)。同时嵌入式系统经常有一些特殊的要求, 而普通的 PC 上的图形窗口系统是不能满足这些要求的, 比如特殊的外观效果, 控制提供给用户的函数, 提高装载速度, 特殊的低层图形或输入设备, 因此可见嵌入式系统必定要有自己的 GUI。

综上所述, 嵌入式 GUI 就是在嵌入式系统中为特定的硬件设备或环境而设计的图形用户界面系统, 所以嵌入式 GUI 不但要具有以上有关 GUI 的特征, 而且在实际应用中, 嵌入式系统对它来说还有如下的基本要求:

- 体积小;
- 运行时耗用系统资源少;
- 上层接口与硬件无关, 高度可移植;
- 高可靠性;
- 在某些应用场合应具备实时性。

UNIX 环境下的图形视窗标准为 X Window System, Linux 是类 UNIX 系统, 所以顶层运

行的 GUI 系统是兼容 X 标准的 XFree86 系统。X 标准大致可以划分为 X Server、Graphic Library（底层绘图函数库）、Toolkits、Window Manager 等几大部分，其好处是具有可扩展性、可移植性等优点，但对于嵌入式系统而言无疑太过庞大、累赘、低效。目前流行的嵌入式 GUI 与 X 思路不同，这些 GUI 一般不局限于 X 标准，更强调系统的空间和效率。

下面我们就目前市场上常用的嵌入式 GUI 做一个简单的介绍。

✎ 1.1.1 MiniGUI

由北京飞漫软件技术有限公司开发的 MiniGUI (<http://www.minigui.org>)，是国内为数不多的几大国际知名自由软件之一。MiniGUI 是面向实时嵌入式系统的轻量级图形用户界面支持系统，1999 年初遵循 GPL 条款发布第一个版本以来，已广泛应用于手持信息终端、机顶盒、工业控制系统及工业仪表、彩票机、金融终端等产品和领域。目前，MiniGUI 已成为跨操作系统的图形用户界面支持系统，可在 Linux/ μ CLinux、eCOS、 μ C/OS-II、VxWorks、等操作系统上运行；已验证的硬件平台包括 Intel x86、ARM（ARM7/AMR9/StrongARM/xScale）、PowerPC、MIPS、M68K（DragonBall/ColdFire）等。

MiniGUI 的良好体系结构及优化的图形接口，可确保最快的图形绘制速度。在设计之初，就充分考虑到了实时嵌入式系统的特点，针对多窗口环境下的图形绘制开展了大量的研究及开发，优化了 MiniGUI 的图形绘制性能及资源占有。MiniGUI 在大量实际系统中的应用，尤其在工业控制系统的应用，证明 MiniGUI 具有非常好的性能。

✎ 1.1.2 MicroWindows

MicroWindows (<http://microwindows.censoft.com>) 是一个开放源码的项目，目前由美国 Century Software 公司主持开发。该项目的开发一度非常活跃，国内也有人参与了其中的开发，并编写了 GB2312 等字符集的支持。但在 Qt/Embedded 发布以来，该项目变得不太活跃了，并长时间停留在 0.89Pre7 版本。可以说，以开放源码形式发展的 MicroWindows 项目，基本停滞。

MicroWindows 是一个典型的基于客户端/服务器体系结构的 GUI 系统，基本分为三层：最底层是面向图形输出和键盘、鼠标或触摸屏的驱动程序；中间层提供底层硬件的抽象接口，并进行窗口管理；最高层分别提供兼容于 X Window 和 Windows CE（Win32 子集）的 API。该项目的主要特色在于提供了类似 X 的客户端/服务器体系结构，并提供了相对完善的图形功能，包括一些高级的功能，如 Alpha 混合、三维支持、True Type 字体支持等。MicroWindows 采用 MPL（mozilla public license）条款发布。

✎ 1.1.3 DinX

DinX 非常适合在很小的系统上运行多窗口程序，它简单、轻巧并且快速。DinX 并不是 X，它使用 Linux 核心的 framebuffer 视频驱动，采用 Client/Server 模式，为此，系统提供了两个界面：`/dev/dinxsvr` 和 `/dev/dinxwin`。

一个服务器程序连接到 `/dev/dinxsvr`，并决定来自各程序窗口的 request 各占有视屏的各个部分，它也负责给各窗口发送像鼠标移动这样的事件消息。Clnet 程序连接到 `/dev/dinxwin`，与 Server 进行消息通信等。Server 进程还负责处理事件、窗口管理、调色板配置等功

能。DinX 是一个实验性的窗口系统，它处在发展阶段中，还存在一些缺陷和问题。DinX 的 license 属于 MPL，也可以转化为 GPL。这样，DinX 核心模块可以集成到 Linux 中，DinX 库可以链接到其他的 GPL 程序中。

1.1.4 OpenGUI

OpenGUI (<http://www.tutok.sk/fastgl/>) 在 Linux 系统上存在已经很长时间了。最初的名字叫作 FastGL，只支持 256 色的线性显存模式，但目前也支持其他显示模式，并且支持多种操作系统平台，比如 MS - DOS、QNX 和 Linux 等，不过目前只支持 x86 硬件平台。OpenGUI 也分为三层：最低层是由汇编编写的快速图形引擎；中间层提供了图形绘制 API，包括线条、矩形、圆弧等，并且兼容于 Borland 的 BGI API；最高层用 C ++ 编写，提供了完整的 GUI 对象集。

OpenGUI 采用 LGPL 条款发布，OpenGUI 比较适合于基于 x86 平台的实时系统，可移植性稍差，目前的发展也基本停滞。

1.1.5 Qt/Embedded

Qt 是 Nokia 公司 (Trolltech) 的一个 C ++ 图形库，可以跨平台使用，与 Windows 下的 MFC 在同一个层次，但比 MFC 更好用。主要特点是比 MFC 要庞大，很多 Qt 是可以移植的。Qt 最初主要是为跨平台的软件开发者提供统一的、精美的图形用户编程接口。Qt 在 LinuxX86 上通过 Qt lib 访问 Linux 下的 x - server 与 x - lib，x - server 与底层硬件相连。Qt 在 Windows 上通过 Qt lib 访问 win api。

Qt 的嵌入式版本叫作 Qt/Embedded，不需要 x - server 与 x - lib 的支持，可以直接写 framebuffer，所以很节省内存。

表 1.1 归纳了 Qt/Embedded 的一些优缺点。

表 1.1 Qt/Embedded 的优缺点分析

优点	以开发包形式提供	包括了图形设计器、Makefile 制作工具、字体国际化工具、Qt 的 C ++ 类库等
	跨平台	支持 Microsoft Windows 95/98/2000、Microsoft Windows NT、MacOS X、Linux、Solaris、HP - UX、Tru64 (Digital UNIX)、Irix、FreeBSD、BSD/OS、SCO、AIX 等众多平台
	类库支持跨平台	Qt 类库封装了适应不同操作系统的访问细节，这正是 Qt 的魅力所在
	模块化	可以任意裁减
缺点	结构也过于复杂臃肿，很难进行底层的扩充、定制和移植	例如： (1) 尽管 Qt/Embedded 声称，它最小可以裁剪到几百 KB，但这时的 Qt/Embedded 库已经基本失去了使用价值； (2) 它提供的控件集沿用了 PC 风格，并不太适合许多手持设备的操作要求； (3) Qt/Embedded 的底层图形引擎只能采用 framebuffer，只是针对高端嵌入式图形领域的应用而设计的； (4) 由于该库的代码追求面面俱到，以增加它对多种硬件设备的支持，造成了其底层代码比较凌乱、各种补丁较多的问题

表 1.2 总结了常见 GUI 的参数比较。

表 1.2 常见 GUI 参数比较

参 数 \ 名 称	MiniGUI	OpenGUI	Qt/Embedded
API（完备性）	Win32（很完备）	私有（很完备）	Qt（C++）（很完备）
函数库的典型大小	300 KB	300 KB	600 KB
移植性	很好	只支持 x86 平台	较好
授权条款	LGPL	LGPL	QPL/GPL
系统消耗	小	最小	最大
操作系统支持	Linux	Linux、DOS、QNX	Linux

1.2 Qt 简史

Qt 工具包最初是在 1995 年 5 月变为公众可用的。它最初由 Haavard Nord（Trolltech 的 CEO）和 Eirik Chambe - Eng（Trolltech 的总裁）开发。Haavard 和 Eirik 是在位于挪威特隆赫姆的挪威科技学院相识的，他们都在那里获得了计算机科学的硕士学位。

Haavard 对 C++ 图形用户界面开发的兴趣始于 1988 年，当时他被一家瑞典公司委托设计并且开发一个 C++ 图形用户界面工具包。

几年后，在 1990 年夏天，Haavard 和 Eirik 因为一个超声波图像方面的 C++ 数据库应用程序在一起工作。这个系统要求能够在 UNIX、Macintosh 和 Windows 上都能运行。那个夏天中的一天，Haavard 和 Eirik 出去散步享受阳光。当他们坐在公园的一个长椅上，Haavard 说：“我们需要一个面向对象的显示系统。”由此引发的讨论奠定了他们即将创建的面向对象的多平台图形用户界面工具包的智力基础。

Haavard 于 1991 年开始和 Eirik 合作设计、编写最终成为 Qt 的这些类。随后的一年，Eirik 提出了“信号和槽”的设想——一个简单并且有效的强大的图形用户界面编程范例。Haavard 实践了这个想法，并且建立了一个手写代码实现。到 1993 年，Haavard 和 Eirik 当时已经开发出了 Qt 的第一个图形核心并且能够实现他们自己的窗口部件。同年末，Haavard 提议他们一起开展并且创建“世界上最好的 C++ 图形用户界面工具包”的业务。

1994 年成为两个程序员不幸的一年，他们没有客户，没有资金，只有一个未完成的产品，却希望闯入一个稳定的市场。非常幸运的是他们的妻子都有工作并且愿意支持他们的丈夫。在这两年 Haavard 和 Eirik 认为需要继续开发他们的产品并且开始盈利。

他们选择“Q”作为类的前缀，是因为该字母在 Haavard 的 Emacs 字体中看起来非常漂亮。“t”代表“工具包（toolkit）”，是从“Xt”，X 工具包中获得的灵感。公司于 1994 年 3 月 4 日创立，最初名为“Quasar Technologies”，后更名为“Troll Tech”，现在改为“Trolltech”。

1995 年 5 月，通过 Haavard 大学时的一个教授的联系，挪威 Metis 公司与他们签订了一份基于 Qt 进行软件开发的合同。大约同一时间，Trolltech 雇佣了 Arnt Gulbrandsen，正是他设计并实现了一套有独创性的文档系统，并且对 Qt 的代码也做出了贡献。

1995 年 5 月 20 日，Qt 0.90 被上传到 sunsite.unc.edu。6 天后，在 comp.os.linux-an-

announce 上发布, 这是 Qt 的第一次公开发布。Qt 可以被同时用于 Windows 和 UNIX 开发, 它在两种平台上提供了相同的应用程序编程接口。从第一天起, Qt 就提供两种许可版本: 一种是进行商业开发所需的商业许可版本, 另一种是进行开源开发的自由软件版本。Metis 的合同确保了 Trolltech 的发展, 在 10 个月之内没有人购买 Qt 的商业许可。

1996 年, European Space Agency 购买了 10 份 Qt 商业许可, 成了第 2 个 Qt 客户。凭着坚定的信念, Eirik 和 Haavard 又雇佣了另外一个开发人员。Qt 0.97 在 5 月底发布, 并且在 1999 年 9 月 24 日, Qt1.0 面世。在这一年底, Qt 已经发展到了 1.1 的版本, 共有来自 8 个不同国家的客户, 他们购买了 18 份商业许可。这一年, 在 Matthias Ettrich 的带领下, 创立了 KDE 项目。

Qt1.2 在 1997 年 4 月发布。Matthias Ettrich 利用 Qt 建立 KDE 的决定使得 Qt 成为在 Linux 环境下开发 C++ 图形用户界面的实际标准。Qt1.3 在 1997 年 9 月发布。

Matthias Ettrich 本人也在 1998 年离开学术界加入 TrollTech, 当年 9 月最后一个 1 系列的版本 1.40 发布。1999 年 6 月 Qt2.0 发布。Qt2.0 有很多架构上的改变, 比它的前几个版本功能更为强大, 更为成熟。它还具备了 40 个新类和 Unicode 支持。Qt2 有了一个新的开源许可, Q 公共许可 (Q Public License, QPL), 它遵循了开源定义。在 1999 年 8 月, Qt 赢得了 LinuxWorld 的最佳库/工具奖。大约在这个时候, Trolltech Pty Ltd (澳大利亚) 建立了。

Trolltech 在 2000 年发布了 Qt/Embedded。它被设计为可以运行在嵌入式 Linux 设备上, 并且提供了它自己的窗口系统作为 X11 的轻型替代品。现在 Qt/Embedded 和 Qt/X11 除了提供商业许可之外, 还可以提供被广泛使用的 GNU 通用许可 (GNU General Public License, GPL)。到 2000 年年底, Trolltech 已经创建了 Trolltech Inc. (美国), 并发布了第一个 Qtopia 版本, 一个手持设备环境。Qt/Embedded 在 2001 年和 2002 年两次获得 LinuxWorld “Best Embedded Linux Solution” (最佳嵌入式 Linux 解决方案) 奖。

Qt3.0 在 2001 年发布。现在 Qt 可以用于 Windows、UNIX、Linux、嵌入式 Linux 和 MacOS X。Qt3.0 提供了 42 个新类和超过 50 万行的代码。Qt3.0 在 2002 年赢得了 Software Development Tools 的 “Jolt Productivity Award (震撼生产力奖)”。

2005 年夏, Qt4.0 发布, 它大约有 500 个类和 9 000 多个函数, Qt4 比以往的任何一个版本都要全面和丰富, 并且它已经裂变成多个函数库, 从而使开发人员可以根据需要只链接所需要的 Qt 部分。相对于以前的所有 Qt 版本, Qt4 的进步是巨大的, 它彻底地对高效易用的模板容器、高级的模型/视图功能、快速而灵活的二维绘图框架和强大的统一字符编码标准的文本查看和编辑类进行了大量改进, 更不必说对那些贯穿整个 Qt 类中的成千上万个小的改良了。现如今, Qt4 具有如此广泛的特性, 以至于 Qt 已经超越了作为图形用户界面工具包的界限, 逐渐成长为一个成熟的应用程序开发框架。Qt4 也是第一个能够在其所有可支持的平台上既可用于商业开发又可用于开源开发的 Qt 版本。

同样在 2005 年, 奇趣公司在北京开设了一家办事处, 以便为中国及其销售区域内的用户提供服务和培训, 并且为 Qt/Embedded Linux 和 Qtopia 提供技术支持。

通过获取一些非官方的语言绑定件 (Language Binding), 非 C++ 程序员也已早就开始使用 Qt, 特别是用于 Python 程序员的 PyQt 语言绑定件。2007 年, 公司发布了用于 C#程序员的非官方语言绑定件 Qyoto。同年, Qt Jambi 投放市场, 它是一个官方支持的 Java 版 Qt 应用程序编程接口。

自奇趣公司 (现已被 Nokia 收购) 诞生以来, Qt 的声望经久不衰, 而且至今仍然保持

高涨。取得这样的成绩不但说明了 Qt 的质量，而且也说明了人们都喜欢使用它。在过去的 10 年中，Qt 已经从一个只被少数专业人士所熟悉的“秘密”产品，发展到了如今遍及全世界拥有数以千计的客户和数以万计的开源开发人员的产品。

1.3 Qt 程序设计简介

Qt 是 Trolltech 公司开发的给予标准框架的图形应用程序，它是一种高效与跨平台的应用程序的解决方案，Qt 支持的平台有微软操作系统、苹果 OS 和 Linux 操作系统，并支持了大部分商业的 UNIX 操作系统和 Linux 嵌入式操作系统。

Qt 提供给应用程序开发者大部分的功能，来完成建立适合的、高效率的图形面程序与后台执行的应用程序，它提供的是一种面向对象的可扩展性能和真正基于组件的编程模式。建议读者在学习这本书之前最好首先熟悉官方所提供的白皮书。

Qt 的第一次商业版本发行是在 1996 年，发展到现在，Qt 已经提供了成千上万的应用程序，包括 Google 地图、Photoshop 的一些元素和 Skype 软件，同时还支持 Linux 桌面系统 KDE 的开发环境，KDE 是 Linux 发行版最广泛的一种图形作业界面。

Qt 支持以下的平台：Windows 98、WindowsNT 4.0、Windows ME、Windows 2000、WindowsXP、UNIX/X11 - Linux、SunSolaris、HP - UX、HPTru64UNIX、IBMAIX、SGIIRIX、MacOSX、MacOSX10.3 +，以及其他的一些版本的 UNIX。在 EmbeddedLinux - Linux 嵌入式平台上需要使用 Framebuffer 帧缓冲的支持。

1.4 Qt 版本介绍

在 Qt 发行版本中将要涉及两个版本：Qt 商业版本和 Qt 开源版本。

1. Qt 商业版本

Qt 商业版本是设计商业软件的开发环境，这些商业软件使用了传统的商业来发布，它包含了一些更新的功能、技术上的支持和大量的解决方案，开发了使用于行业的一些特定的组件，有一些特殊的功能只在商业用户中使用。

2. Qt 开源版本

Qt 开源版本是用来开发开源的软件，它提供了一些免费的支持，并遵循 QPL 协议。

开放源代码是免费的软件，不牵涉用户的某些权益。任何人都有使用开源软件和参与它的修改的机会，这就意味着其他的人同样可获得你开发的代码。

开源软件应该较多地谈论软件自由，而不是金钱。在自由软件基金会的许可背景下，讲究软件自由，而不是不计成本地去使用软件。

Trolltech 公司提供了 Qt 的开源版本，让大部分人可以使用开源的 Qt 来开发自由的开源软件，所有在 Qt 开源版本下开发的软件都需要遵守 GNU 的公共许可。

自从 Qt 开源以来已经成功地完成了很多优秀的项目，如 KDE。全世界有相当部分的程序员在使用 Qt 的开源版本，它们没有被告知要缴纳使用费用。在 Qt4X 以上的版本，Qt 开

源主要面向的 UNIX/X11、MacOS 及 Windows 用户。

开源版本的 Qt 可以在原 TrollTech 公司的网站上进行下载。当然程序员在开发程序的时候也可以使用 Qt 的商业版本,但这样就需要购买一个商业的许可,才可以将你的软件作为商业的软件进行发布。

这样的运作模式是很优秀的,开源软件不断地使用最小的成本增加了软件的功能,商业软件符合了商业软件的发行模式。许多大型的公司因需要保护软件的许可权益,所以愿意购买原 TrollTech 的商业许可权,使得原 TrollTech 公司保证了盈利的来源。

如果你选择了 Qt 的开源版本,GNU 的权益也就强加给了你,也就是说你要遵守自由的理念,将你开发的开源代码分享给其他的朋友。总之,在 GNU 的协议下你需要将软件的源代码共享到底。

1.5 怎样学习 Qt

1.5.1 从 C++ 到 Qt 的跨越

C++ 程序语言是一个成功的典范,虽然它有很多不足的地方(如快速开发,对程序员质量的要求很高)。但在 Qt 当中不需要计较这么多,Qt 的出色表现让 C++ 语言蓬荪生辉。

在 Qt 中我们不需要关心底层的类怎样工作,而着重于图形的开发、美工的修饰、功能的扩展及接口的形成。当链接数据库的时候,C++ 语言会提供接口,SQL 的提供商也会开发这方面的包,以满足各种程序设计条件下的需要。

当试图链接一个服务器的时候,程序员需要对于网络协议和套接字的构造有充分的经验,这些在 Qt 中都被涉及,并且很好地提供了丰富的类。

从 C 到 C++ 是机制的升华,而从 C++ 到 Qt,则是在内容和方便性上的变革。如果选择 C++ 开发一个大型的图形程序,就需要对 XWindow 和显卡硬件有所了解,编写大量的代码,还需要掌握很多知识,如各种显卡参数。而 Qt 从上到下都为用户做了这些,用户只需要负责对它调用就能完成工作,用户只要活动鼠标,就可以构造出美观的窗口。

1.5.2 C 语言和其他程序设计的关系

C 语言是一种通用的程序设计语言,它包含了紧凑的表达式、丰富的运算符数据类型,以及用户自定义的数据结构。C 语言功能丰富,可扩展性很强。本书的重点是讲解 Qt 的编程,但有必要涉及部分 C 语言的内容。这是因为 Linux 内核是用 C 语言写成的,C++ 也是由 C 语言继承和扩展而来,Qt 则是一组 C++ 的工具包。

在 C 语言中使用的一种特殊的数据结构,即结构体,例如:

```
struct stu
{
    unsigned long number;
    char * name;
    char * sex;
};
```

C++ 对 C 进行了扩展，例如：

```
struct stu_info
{
    unsigned long number;
    char * name;
    char * sex;
public :
    void set_name();
    void get_number();
};
```

定义类的模式：

```
class stu_info
{
    unsigned long number;
    char * name;
    char * sex;
public :
    void set_name();
    void get_number();
}
```

类与结构体十分相似，上面的例子展示了从结构体到类的演变过程。在面向对象设计中，可以将结构体看作类的一种特殊变体。本书的主要内容是 Qt 类的操作，这就要求读者要有一定的 C++ 的基础。Qt 是建立在 C++ 基础之上的，要学好它就必须掌握 C++。这样用户才能很顺利地去研究 Qt 的例子，分析各种类的功能，熟练地使用它们。使用 Qt 编写程序时，主要面向的对象就是类的操作。虽然大部分类的属性可以在属性编辑框中操作，但是有些信号与槽是必须要个人自定义设计的，程序往往要完成不同的操作，而 Qt 这个工具在适用更大范围的操作的时候会显得力不从心。

大量的槽和信号的编写可以帮助用户完成很多事情。最好的学习方式还是阅读源代码，在国外有很多的软件都是使用 Qt 编写的，它们都有源码包，可以帮助用户熟悉 Qt 工具。本书也是由大量的实例构成的，也是为了让大家能够在学习中侧重于应用，分析并了解源代码的作用。

1.6 Qt 开发工具的优点

1. 面向对象

良好的封装机制使得 Qt 的模块化程度非常高，可重用性较好，非常有利于用户的开发。Qt 提供了一种称为信号/槽的安全机制来替代回调，这使得各个部件之间的协同工作变得十分简单，另外 Qt 还提供了信号/槽的编译器，具有可视化的编辑功能。

2. 丰富的 API

Qt 有多达 250 个以上的 C++ 类，还有基于模板的集合、串行、文件、I/O 设备、目录管理和日期/时间类，甚至还包括正则表达式的处理功能。

3. 大量的开发文档

Qt 提供了丰富的开发文档，有利于方便地查找有用的信息，Qt 在许多类当中提供了相似的接口支持，以便于使用 Qt 的官方文档编写程序。

XML 是可扩展标识语言（Extensible Markup Language）的简写，目前推荐遵循的是 W3C 组织于 2000 年 10 月 6 日发布的 XML1.0 版本，具体请参考（www.w3.org/TR/2000/REC-XML-20001006）。Qt 还包含了大量的关于 XML 的接口。

1.7 Qt/Embedded 介绍

1.7.1 架构

Qt/Embedded 以原始 Qt 为基础，并做了许多出色的调整以适用于嵌入式环境。Qt/Embedded 通过 Qt API 与 Linux I/O 设施直接交互，成为嵌入式 Linux 端口。同 Qt/X11 相比，Qt/Embedded 很省内存，因为它不需要一个 X 服务器或者 Xlib 库，它在底层抛弃了 Xlib，采用 framebuffer（帧缓冲）作为底层图形接口；同时，将外部输入设备抽象为 keyboard 和 mouse 输入事件。Qt/Embedded 的应用程序可以直接写内核缓冲帧，这避免开发者使用烦琐的 Xlib/Server 系统。图 1.1 比较了 Qt/Embedded 与 Qt/X11 的架构区别。使用单一的 API 进行跨平台的编程可以有很多好处，提供嵌入式设备和桌面计算机环境下应用的公司可以培训开发人员使用同一套工具开发包，这有利于开发人员之间共享开发经验与知识，也使得管理人员在分配开发人员到项目中的时候增加灵活性。更进一步来说，针对某个平台而开发的应用和组件也可以销售到 Qt 支持的其他平台上，从而以低廉的成本扩大产品的市场。



图 1.1 Qt/Embedded 与 Qt/X11 的 Linux 版本的比较

1. 窗口系统

一个 Qt/Embedded 窗口系统包含了一个或多个进程，其中的一个进程可作为服务进程，该服务进程会分配客户显示区域，以及产生鼠标和键盘事件，还能够给运行起来的客户应用程序提供输入方法和一个用户接口。该服务进程其实就是一个有某些额外权限的客户进程，任何程序都可以在命令行上加上“-qws”的选项来把它作为一个服务进程运行。

客户端与服务器之间的通信是使用共享内存的方法实现的，通信量应该保持最小，例如客户进程直接访问帧缓冲来完成全部的绘制操作，而不会通过服务器，客户程序需要负责绘制它们自己的标题栏和其他式样。这就是 Qt/Embedded 库内部层次分明的处理过程。客户