



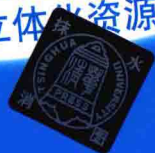
高职高专**立体化教材**计算机系列

嵌入式 Linux 基础与实训教程

龙 华 谢剑明 主 编



赠送电子课件及
其他立体化资源

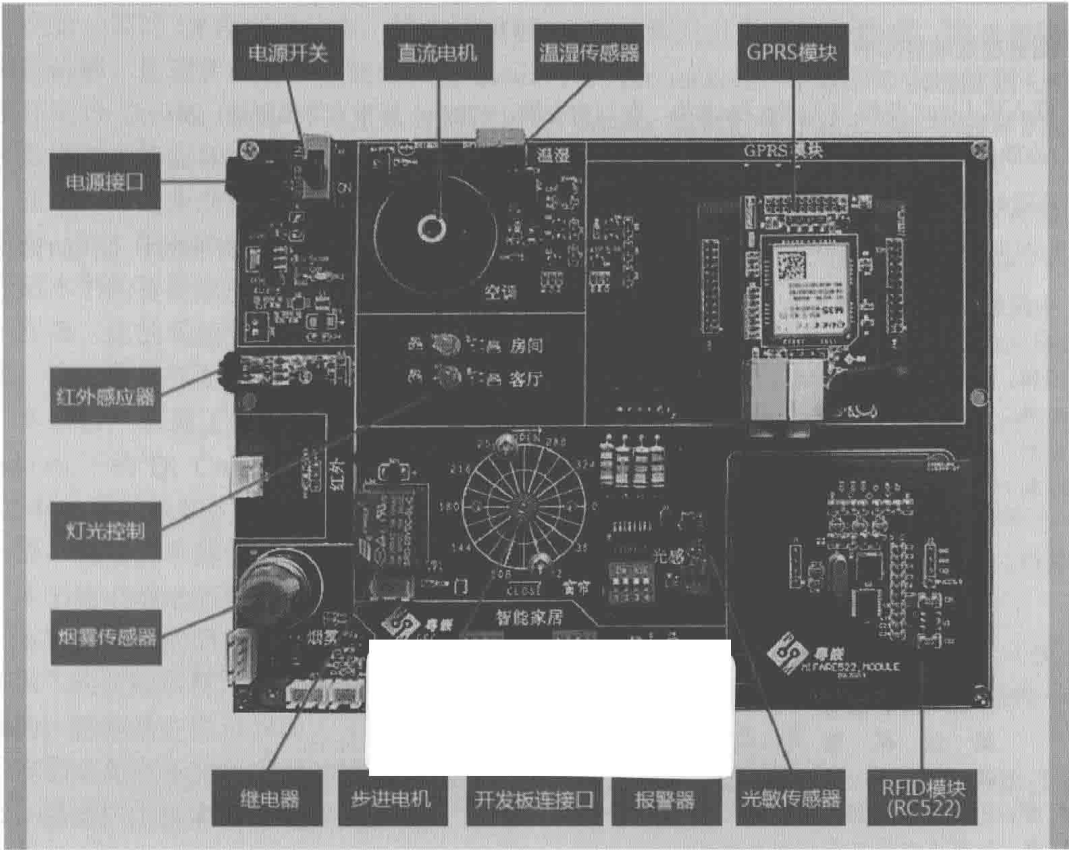


清华大学出版社

高职高专立体化教材 计算机系列

嵌入式 Linux 基础与实训教程

龙 华 谢剑明 主编



清华大学出版社
北 京

内 容 简 介

本书以完成一个智能家居项目为主线,讲述了实训各个阶段所需要的知识。全书分为三篇,第一篇讲述了项目的界面部分,用 Qt 程序设计,并以一个小项目媒体播放器的制作来加深理解。第二篇讲述了项目的驱动程序部分,即基于 ARM 学习板及扩展电路板的驱动程序的编写。第三篇讲述了系统的集成,即嵌入式 Qt 界面控制硬件,完成一个简易的智能家居项目。

本书适合作为大专院校嵌入式专业、物联网专业学生的实训教材。此外,有志从事嵌入式 Linux 开发的人员可以把本书作为入门级读物。

本书封面贴有清华大学出版社防伪标签,无标签者不得销售。
版权所有,侵权必究。侵权举报电话:010-62782989 13701121933

图书在版编目(CIP)数据

嵌入式 Linux 基础与实训教程/龙华,谢剑明主编. —北京:清华大学出版社,2015

(高职高专立体化教材 计算机系列)

ISBN 978-7-302-41314-1

I. ①嵌… II. ①龙… ②谢… III. ①Linux 操作系统—程序设计—高等职业教育—教材 IV. ①TP316.89

中国版本图书馆 CIP 数据核字(2015)第 195618 号

责任编辑:陈冬梅 宋延清

封面设计:刘孝琼

责任校对:周剑云

责任印制:杨 艳

出版发行:清华大学出版社

网 址: <http://www.tup.com.cn>, <http://www.wqbook.com>

地 址:北京清华大学学研大厦 A 座 邮 编:100084

社 总 机:010-62770175 邮 购:010-62786544

投稿与读者服务:010-62776969, c-service@tup.tsinghua.edu.cn

质 量 反 馈:010-62772015, zhiliang@tup.tsinghua.edu.cn

课 件 下 载: <http://www.tup.com.cn>, 010-62791865

印 刷 者:北京富博印刷有限公司

装 订 者:北京市密云县京文制本装订厂

经 销:全国新华书店

开 本:185mm×260mm 印 张:16.75 插 页:1 字 数:409 千字

版 次:2015 年 9 月第 1 版

印 次:2015 年 9 月第 1 次印刷

印 数:1~3000

定 价:32.00 元

前 言

Linux 操作系统作为源码开放的自由软件,经过 20 多年的发展与壮大,越来越受到 IT 界的认可,运行于 ARM 平台的嵌入式 Linux 在智能家居、智能汽车等智能化管理系统中的运用占据了很大的比重。嵌入式 Linux 的学习和开发相对于纯粹的程序设计或是电路设计而言,门槛更高一点,因为知识面要求更广一点,既要精通软件,硬件知识也要懂一点。横跨软硬件的嵌入式 Linux 驱动程序开发是学生最感困难的地方,但可视化的 Qt 程序设计,是相对容易一些的。本书通过一个通俗、简易的智能家居项目,先讲述如何以 Qt 创建界面,然后讲述驱动程序的编写,最后集成为可视化的界面控制硬件,从而减轻读者的学习压力,引发读者的学习兴趣。

本书的软件开发是基于 Windows 下编辑、Linux 下编译的模式,注重讲解实验的过程,力求从看到的实验现象去分析背后的原理。

在第一篇的 Qt 程序设计中,第 1 章讲解 Qt 的起源及开发环境的搭建;第 2 章通过几个例题讲解,让读者从 C++ 基础知识及 Linux 中的 gcc makefile 等基础知识过渡到 Qt 集成开发环境 Qt Creator 的使用,并重点分析使用 Qt Creator 编辑的程序在 Linux 中编译的过程;第 3 章详细讲解媒体播放器的制作,加深对 Qt 程序设计技能的掌握。

在第二篇的驱动程序开发中,第 4 章讲解软件开发环境的搭建,并详细介绍实验的硬件平台;第 5 章讲解 Linux 内核对驱动程序的支撑,并介绍一些常见设备的驱动程序;第 6 章介绍本书的智能家居项目所需的 8 个驱动程序。

在第三篇的系统集成中,主要完成的工作是在硬件平台上把图形界面与驱动程序连接起来,实现图形界面与硬件的交互。

本书由广东理工职业学院的龙华和深圳技师学院的谢剑明共同编写,书中的代码在 Windows 下的 Qt Creator、SourceInsight 以及 Redhat Enterprise 5 虚拟机中编写,并且在 GEC2440 学习板和扩展板上通过测试。第一篇和第二篇是相对独立的,读者可以单独学习第一篇,也可以单独学习第二篇。

本书随书赠送的资源可从清华大学出版社的网站下载。

本书所讲解的简易智能家居项目曾经是作者所带的 12 级嵌入式专业的“嵌入式技术综合实训”的课程项目,在此感谢这些学生,他们给了作者很多灵感。学生的想法是特殊的、新颖的,特别是在图形界面方面的设计,比老师的更容易扩展。

本书难免有不完美之处和错误的地方,写作方法也因构思创新而有需要进一步改良的地方,恳请广大读者提出宝贵的意见。

龙 华

目 录

第一篇 Qt 程序设计

第 1 章 Qt 编程基础及开发环境的搭建...1

- 1.1 Qt 的起源.....1
- 1.2 Linux 系统下的图形界面.....3
 - 1.2.1 Linux 本身包括内核和 Shell.....3
 - 1.2.2 关于 X11.....4
 - 1.2.3 窗口管理器(综合桌面环境).....5
- 1.3 Qt 的特点及主要构成.....5
 - 1.3.1 Qt 的特点.....5
 - 1.3.2 Qt 开发环境的主要构成.....7
- 1.4 Qt 开发环境的搭建.....8
 - 1.4.1 安装 Qt Creator.....8
 - 1.4.2 安装虚拟机.....11
 - 1.4.3 在虚拟机中安装 Linux 操作系统.....15
- 1.5 使用 Qt 开发环境创建第一个应用程序.....28
 - 1.5.1 关于可视化编程.....28
 - 1.5.2 Qt Creator 常用菜单的功能介绍.....28
 - 1.5.3 举例.....29

第 2 章 Qt 编程基础.....36

- 2.1 C++基础.....36
- 2.2 Qt 程序编译过程分析.....43
 - 2.2.1 Qt Creator 中应用程序的编译过程.....43
 - 2.2.2 Qt 的元对象系统.....48
 - 2.2.3 Qt 资源文件.....49
- 2.3 Qt 控件编程.....52
 - 2.3.1 Qt 的帮助系统.....53
 - 2.3.2 Qt Creator 界面中的控件.....54
 - 2.3.3 常用控件举例.....57
- 2.4 Qt 布局管理.....60
- 2.5 信号与槽.....61
- 2.6 媒体播放器的制作.....62
 - 2.6.1 制作界面.....62
 - 2.6.2 编写控件的事件代码.....64

第 3 章 智能家居项目 Qt 界面设计.....75

- 3.1 界面跳转.....75
- 3.2 智能家居多级子项目界面设计.....81
 - 3.2.1 智能家居项目需求分析.....81
 - 3.2.2 代码实现.....82

第二篇 驱动程序开发

第 4 章 搭建开发环境.....93

- 4.1 软件开发环境的搭建.....93
 - 4.1.1 安装 VMware Workstation 软件.....93
 - 4.1.2 建立虚拟电脑.....94
 - 4.1.3 Ubuntu 12.04 的安装.....102
 - 4.1.4 Ubuntu 12.04 系统的配置.....106
 - 4.1.5 安装使用 Source Insight.....113

- 4.1.6 交叉环境的搭建.....127
- 4.2 硬件环境介绍.....129
 - 4.2.1 S3C2440 芯片介绍.....129
 - 4.2.2 S3C2440 开发板主要硬件的说明.....133

第 5 章 驱动程序开发的内核基础.....144

- 5.1 嵌入式 Linux 操作系统的移植过程概述.....144

5.1.1	Linux 操作系统及其特点	144
5.1.2	嵌入式 Linux 操作系统的特点	144
5.1.3	嵌入式 Linux 系统的移植步骤	145
5.2	基于 GEC2440 开发板的内核移植	146
5.2.1	Linux-2.6.30 内核移植	146
5.2.2	驱动模块的动态编译与静态编译	153
5.3	常见的嵌入式内核驱动编译与移植	161
5.3.1	LCD 驱动移植	161
5.3.2	触摸屏驱动移植	166
5.3.3	USB 驱动移植	170
5.3.4	网卡驱动移植	174

第 6 章 智能家居项目驱动程序实验

6.1	房间灯驱动程序	177
6.1.1	实验目的	177
6.1.2	实验原理	178
6.1.3	实验步骤	178
6.2	窗帘模块驱动程序	186
6.2.1	光敏电阻驱动程序	186
6.2.2	步进电机的驱动程序	194
6.3	温湿度控制模块驱动程序	203
6.3.1	温湿度传感器 DHT11 驱动程序	203
6.3.2	直流电机驱动程序	211
6.4	烟雾及红外检测模块驱动程序	215
6.4.1	烟雾传感器 MQ-2 驱动程序	215
6.4.2	红外模块驱动程序	222
6.4.3	蜂鸣器驱动程序	228

第三篇 系统集成

第 7 章 Qt 图形下载到学习板

7.1	带 QtE4 的文件系统分析	239
7.2	对 QtE 应用程序的编译	240

第 8 章 智能家居项目集成

8.1	把测试程序加入到 Qt 程序中	241
8.1.1	灯光控制界面	241

8.1.2	窗帘控制界面	244
8.1.3	温湿度控制界面	248
8.1.4	烟雾及红外检测界面	253
8.2	驱动程序的统一加载	257

附录 制作根文件系统

参考文献

第一篇 Qt 程序设计

第 1 章 Qt 编程基础及开发环境的搭建

主要内容

- Qt 简介
- Qt 开发环境的搭建
- 使用 Qt 开发环境创建第一个应用程序

1.1 Qt 的起源

Qt 使用“一次编写，随处编译”的方式，为开发跨平台的图形用户界面提供了一个完整的 C++ 应用程序开发框架。Qt 框架首度为公众见识，是在 1995 年 5 月，它最初由 Haavard Nord(奇趣科技公司的 CEO)和 Eirik Chambe-Eng(公司总裁)开发而成。Haavard 和 Eirik 在位于挪威特隆赫姆的挪威科技学院相识，在那里，他们都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，一家瑞典公司委托他开发一套 C++ 图形用户界面框架。几年后，在 1990 的夏天，Haavard 和 Eirik 因为一个超声波图像方面的 C++ 数据库应用程序而在一起工作，该系统需要一个在 Unix、Macintosh 和 Windows 上都能运行的图形用户界面。在那个夏日中的某一天，Haavard 和 Eirik 一起出去散步，享受阳光，当他们坐在公园的一条长椅上时，Haavard 说：“我们需要一个面向对象的显示系统。”由此引发的讨论，为即将创建的面向对象的跨平台的图形用户界面框架奠定了基础。

1991 年，Haavard 和 Eirik 开始一起合作设计、编写最终成为 Qt 的那些类。在随后的一年中，Eirik 提出了“信号与槽”的设想，那是一个简单并且有效的强大的图形用户界面编程规范，而现在，它已经可以被多个工具包实现。Haavard 实践了这一想法，并且据此创建了一个特定代码的实现系统。到 1993 年，Haavard 和 Eirik 已经开发出了 Qt 的第一套图形内核程序，并且能够利用它实现一些窗口部件。同年末，为了创建“世界上最好的 C++ 图形界面框架”，Haavard 提议一起进军商业领域。

1994 年却成为这两位年轻程序员遗憾的一年。他们没有客户，没有资金，只有一个未完成的产品。但是，它们希望能够闯进一个稳定的市场。幸运的是，他们的妻子都有工作，并且愿意为他们的丈夫提供支持，在这两年里，Haavard 和 Eirik 都认为，他们需要继续开发产品，并且从中赚得收益。

之所以选择字母 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。随后添加的字母 t 代表“工具包(toolkit)”。这是从 Xt(一个 X 工具包)的命名方式中获得的灵感。公司于 1994 年 3 月 4 日成立，最初的名字是 Quasar Technologies，随后更

名为 Troll Tech, 而公司今天的名字则是 Trolltech。

1995 年 4 月, 通过 Haavard 就读过的大学的一位教授的联系, 挪威的 Metis 公司与他们签订了一份基于 Qt 进行软件开发的合同。大约在同一时间, 公司雇用了 Arnt Gulbrandsen。在公司工作的 6 年时间里, 他设计并实现了一套独具特色的文档系统, 并且对 Qt 的代码也做了不少的贡献。

1995 年 5 月 20 日, Qt 0.90 被上传到 sunsite.unc.edu, 6 天后, 在 comp.os.linux.announce 上发布。这是 Qt 的第一个公开发行人版本, Qt 既可以用于 Windows 上的程序开发, 又可以用于 Unix 上的程序开发, 而且在这两种平台上, 都提供了相同的应用程序编程接口。从第一天起, Qt 就提供了两个版本的软件许可协议: 一个是商业开发所需的商业许可协议版, 另一个则是适用于开源开发的自由软件许可协议版。Metis 的合同确保了公司的发展, 然而, 在随后长达 10 个月的时间内, 再没有人购买 Qt 的商业许可协议。

1996 年 3 月, 欧洲航天局(European Space Agency)购买了 10 份 Qt 的商业许可协议, 它成了第二位 Qt 客户。凭着坚定的信念, Eirik 和 Haavard 又雇用了另外一名开发人员, Qt 0.97 在同年 5 月底正式发布, 随后, 在 1996 年 9 月 24 日, Qt 1.0 正式面世。到了这一年的年底, Qt 的版本已经发展到了 1.1, 共有来自 8 个不同国家的客户购买了 18 份 Qt 的商业许可协议。也就是在这一年, 在 Matthias Ettrich 的带领下, 创立了 KDE 项目。

Qt 1.2 于 1997 年 4 月发布。Matthias Ettrich 利用 Qt 建立 KDE 的决定, 使 Qt 成为 Linux 环境下开发 C++ 图形用户界面的事实标准。Qt 1.3 于 1997 年 9 月发布。

Matthias 在 1998 年加入公司, 并且在当年 9 月, 发布了 Qt1 系列的最后一个版本, 即 Qt 1.40。

1999 年 6 月, Qt 2.0 发布, 该版本拥有一个新的开源许可协议, 即 Q 公共许可协议(Q Public License, QPL), 它与开源的定义一致。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)成立了。

2000 年, 公司发布了 Qt/Embedded Linux, 它用于 Linux 嵌入式设备。

Qt/Embedded Linux 提供了自己的窗口系统, 并且可以作为 X11 的轻量级替代产品。Qt/X11 和 Qt/Embedded Linux 除了提供商业许可协议外, 还提供广为使用的 GNU 通用公共许可协议(General Public License, GPL)。2000 年底, 成立了 Trolltech Inc.(美国), 并发布了 Qtopia 的第一版, 它是一个用于移动电话和掌上电脑(PDA)的环境平台。Qt/Embedded Linux 在 2001 年和 2002 年两次获得了 LinuxWorld 的 Best Embedded Linux Solution 奖, Qtopia Phone 也在 2004 获得了同样的荣誉。

2001 年, Qt 3.0 发布。此时, Qt 已经可用于 Windows、Mac OS X、Unix 和 Linux(桌面和嵌入式)平台。Qt3 提供了 42 个新类和超过 500,000 行的代码。Qt3 是自 Qt2 以来前进历程中最为重要的一步, 它主要在诸多方面进行了众多改良, 包括本地化和统一字符编码标准的支持、全新的文本查看和编辑窗口部件, 以及一个类似于 Perl 正则表达式的类等。2002 年, Qt3 赢得了 Software Development Times 的 Jolt Productivity Award 奖项。

2005 年夏, Qt 4.0 发布, 它大约有 500 个类和 9000 多个函数, Qt 比以往的任何一个版本都要全面和丰富, 并且它已经形成多个函数库, 从而使开发人员可以根据自己的需要, 只连接所需要的 Qt 部分。相对于以前的所有 Qt 版本, Qt4 的进步是巨大的, 它不仅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,而且还有一些贯穿于整个 Qt 类中的成千上万个小的改良。这时候,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,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。

2008 年,奇趣科技公司被诺基亚公司收购,Qt 也因此成为诺基亚旗下的编程语言工具。2012 年,Qt 被 Digia(总部位于芬兰的 IT 业务供应商)收购。

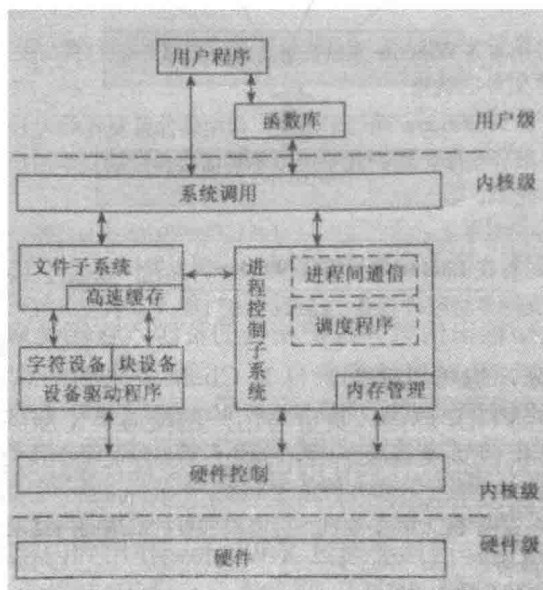
Qt 作为一个著名的跨平台程序开发框架,拥有直观、强大的 API(应用程序接口),在现实世界中拥有广泛的用途。例如,Adobe PhotoShop Album 就是用 Qt 编写的面向大众市场的 Windows 应用程序的一个例子;市场中很多功能完善的软件系统,如三维动画工具、数字电影处理工具、自动化电路设计系统(用于芯片设计)、油气资源勘探系统、金融服务系统以及医学成像系统等,都可以基于 Qt 构建而成;很多的 ATM 柜员机应用程序也是基于 Qt 开发的。基于 Qt 编写的 Windows 应用软件产品,在不使用 Native API 前提下,口令经过重新编译,就可以轻松地在 Mac OS X 和 Linux 世界中开拓出新的市场。

1.2 Linux 系统下的图形界面

Linux 图形界面的结构层次如下:Linux 本身 → X 服务器(通过 X 协议通信) → 窗口管理器(综合桌面环境)。

1.2.1 Linux 本身包括内核和 Shell

内核是系统的核心,主要由 5 个子系统组成:进程调度、内存管理、虚拟文件系统、网络接口、进程间通信。Linux 内核的整体结构如下图所示。



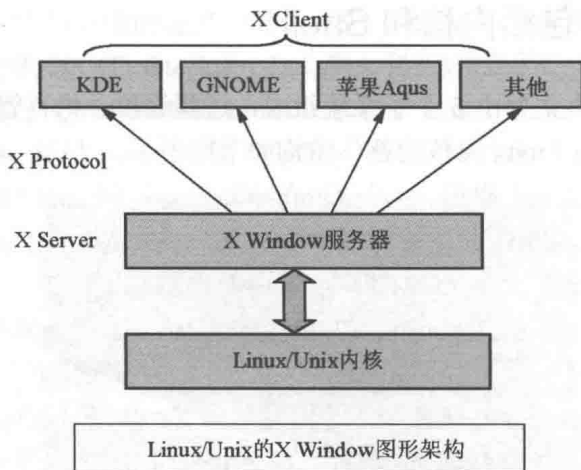
在 Linux 虚拟机中, Linux 内核源代码位于 `/usr/src/linux` 目录下, Linux 操作系统的主要目录如下。

- `/include`: 该子目录包含了建立内核代码时所需的大部分包含文件, 这个模块利用其他模块重建内核。
- `/init`: 该子目录包含了内核的初始化代码, 这是内核工作开始的起点。
- `/arch`: 该子目录包含了所有硬件结构特定的内核代码, 如 i386、alpha。
- `/drivers`: 该子目录包含了内核中所有的设备驱动程序, 如 SCSI 设备。
- `/fs`: 该子目录包含了所有的文件系统的代码, 如 ext2、vfat 等。
- `/net`: 该子目录包含了内核的联网代码。
- `/mm`: 该子目录包含了所有内存管理。
- `/ipc`: 该子目录包含了进程间通信代码。
- `/kernel`: 该子目录包含了主内核代码。

Shell 是系统的内核界面, 提供了用户与内核进行交互操作的接口。Shell 指的是一种程序, 负责解释用户命令输入, 通知内核执行相应的系统服务子程序。Shell 脚本程序是一种解释型程序, 类似于批处理命令, 但比其更丰富, 主要用于系统管理和服务器配置。

1.2.2 关于 X11

X11 是 X Window 系统的第 11 版, 现在几乎所有的操作系统都能支持和使用 X11, 包括当今知名的桌面环境——GNOME 和 KDE, 它们都是以 X Window 系统为基础构建的。X11 是用户使用的桌面图形与内核交互的中间桥梁。它的结构分为三层, 如下图所示。



X 服务器主要处理输入/输出信息并维护相关的资源, 它接受来自键盘、鼠标的操作并将它交给 X Client, 反过来, 也可以接受来自 X Client 发来的信息。最外层的 X Client(如 KDE GNOME)则提供完整的 GUI 界面, 负责与用户的交互。X 服务器通过 X Protocol 与 X Client 沟通。因为 X Client 可以是各种各样的, 所以, 尽管各操作系统都采用了 X Window, 但外观的 GUI 有很大的不同。

在桌面上开发应用系统时, 必然要通过 X Window 作用到内核上去, 为了屏蔽 X 协议层实现的细节, Xlib 库就被开发出来了。

Xlib 是一种 X Window System 协议的客户端,用 C 语言编写,其功能是与 X Server 沟通。因为 Xlib 是底层库,所以在应用程序开发过程中很少直接调用 Xlib,而是在 Xlib 库的基础上发展出更高层的函数库,比如 Qt SDK 库。因此,目前嵌入式 ARM 平台上的项目,很多都是使用 Qt 来控制硬件的。

1.2.3 窗口管理器(综合桌面环境)

窗口管理器(Window Manager)是单独的客户程序,它是系统的一部分,用来控制屏幕上每个应用视窗的外观和操作方式(如最大化、最小化)。在 X Window 系统中,提供了多种窗口管理器,用户可以随时更换它们,而不需要改变系统其他任何部分,也不需要重新编译和改变其他应用程序。

桌面环境为 Linux/Unix 操作系统提供一个更加完善的界面以及大量各类整合工具和应用程序,它提供的典型功能有:文件管理、播放功能、菜单和对话框、配置功能。最常见的桌面系统是 KDE 和 GNOME。

KDE 的全称是 K Desktop Environment,是一个由德国人 Matthias Ettrich 发起,起源于 1996 年 10 月的软件项目,目的是建立一个基于 X Window 标准、适用于 Unix/Linux 的完整易用的桌面环境。作为一个完整的桌面环境,KDE 不仅包含有文件管理器、窗口管理器、帮助系统、邮件处理程序、新闻组阅读程序等,还有不计其数的工具及图形界面配置系统。

KDE 一经推出,即受到了 Linux 用户的欢迎,在 GNOME 出现之前,KDE 几乎成为各 Linux 发行套件使用的标准桌面环境。

由于 KDE 是基于 Trolltech 公司的 Qt SDK 开发的,而 Qt 不是自由软件,使用它开发商业软件,是必须向 Trolltech 公司支付版权费用的,鉴于此,以 Miguel De Icazq 为首的 200 名程序员决定从零开始,开发一个图形桌面环境,即 GNOME(Gnu Network Object Modeling Environment),GNOME 项目得到了 Red Hat 公司的大力支持,经过开发人员 14 个月的努力,GNOME 最终得以完成,它是以真正自由的 GTK 作为底层开发库实现 KDE 功能的。

1.3 Qt 的特点及主要构成

1.3.1 Qt 的特点

Qt 作为现代跨平台应用程序开发框架,具有与一般 GUI 开发包不同的特征,这使得它的应用十分广泛。Qt 具有下列特点。

1. 支持构件化编程

Qt 通过模块化设计和良好的封装,提高了软件构件和模块的可重用性。相关联的组件之间不需要知道彼此的内容,通过一种“信号/槽”的机制与外界交互。“信号/槽”是一种安全可靠的方式,它实现了回调,并支持对象间无须知道彼此信息的情况下的协同工作。这种特性使得 Qt 适用于构件编程。

2. 跨平台

Qt 解决了开发人员在处理不同系统时的本地化问题。Qt 通过提供一系列的类,在为开发人员提供功能的同时,屏蔽了不同操作系统上的实现细节,使得开发人员进行诸如文件处理、日期时间处理等方面更加便利。

3. 国际化

Qt 支持应用程序本地化,在程序界面中出现的文本可以通过翻译表被翻译成各国语言。此外,Qt 支持 Unicode 字符集,同时,它也能运行从右到左及从上到下型的语言,像阿拉伯语(Arabic)和希伯来语(Hebrew)。

4. 完整的功能组件

Qt 为各种专业应用提供了丰富的类库,共计 370 个,主要包括界面控件类、数据库类、日期时间类、拖放操作类、事件类、通用窗口类、图形图像类、打印类、系统帮助类、布局管理类、设备相关类、输入输出与网络类、窗口相关类等。这些类组成了 Qt 编程的基本模块——组件,Qt 含有用来实现各种专业应用所需要的各种组件。

5. 高性能

Qt 是一个全面的应用程序开发框架,包括广泛的特征、性能与工具,可以开发高性能、跨平台的客户端及服务器端的应用程序。Qt 已由成千上万商业与开源应用程序开发人员在多个操作系统与编译器上进行了全面的测试,奠定了高性能与资源性应用程序的基础。

6. 自定义外观

Qt 支持主题,通过 Qt 开发的应用软件都能在 Motif Style、Windows Style 等外观主题之间切换,不管是在 X Window 环境下还是在 Microsoft Windows 环境下。

7. 图形支持

Qt 通过 QPainter 和 QGraphicsView API 对高级 2D 图形提供了支持。此外,Qt 通过其 OpenGL 模块支持 3D 图形。

QPainter 提供了一个全面的 2D 绘图框架。除了渲染多边形、绘图路径等功能外,还支持平滑处理、渐变和 Alpha 混合。

Qt 开发人员可以使用 OpenGL 在 GUI 应用程序中绘制 3D 图形。Qt 提供了单独的 QtOpenGL 模块,可将 QtOpenGL 图像与本地窗口系统集成在一起。

8. 强大的软件开发工具包

Qt 的成功不仅归功于它的跨平台特性、丰富的功能组件、优异的图形图像处理功能及先进的程序设计理念,一个不可忽视的特征就是 Qt 提供的强大的软件开发工具包(Qt SDK)。Qt SDK 包括了 Qt 库、Qt Creator IDE 和 Qt 工具,Qt SDK 同样也支持 Linux/X11、Windows、Mac 三种主流桌面操作系统。主要的开发工具如下。

(1) Qt Designer: 一个功能强大的 GUI 布局和窗口构造器,能够在所有支持平台上以本地化的视图外观和认知快速开发高性能的用户界面。

(2) Qt Linguist: 一套用来消除国际化工作流程中所带来的障碍的工具。使用 Qt Linguist, 开发小组可把应用程序的翻译转换外包给非技术性的翻译人员, 从而可增加精确度, 大大加快本地化处理过程。

(3) Qt Assistant: 一个完全可自定义、重新分配的文件或文档浏览器。

1.3.2 Qt 开发环境的主要构成

截至目前, Qt 已经由一个简单的图形工具包演变成为具有事实标准意义的应用程序开发框架。成长为一个综合性的软件开发环境, 它的主要构成简要介绍如下。

1. GCC

GCC 是 GNU C Compiler 的缩写(GNU 是类似 Unix 的操作系统, 是由一系列应用程序、系统库和开发工具构成的软件集合, 包括用于资源分配和硬件管理的内核), 是 Linux 系统下程序的编译器。GCC 最初只是一个 C 语言编译器, 随着众多自由开发者的加入和 GCC 自身的发展, 如今的 GCC 已经是一个包括 C、C++、Ada、Object C 和 Java 等众多语言的编译器, 所以, GCC 也由原来的 GNU C Compiler 变为 GNU Compiler Collection。也就是 GNU 编译家族的意思。当然, 如今的 GCC 借助于它的特性, 具有了在一个平台下编译另一个平台代码的交叉编译器功能。

2. MinGW

MinGW 是 Windows 平台下的 GCC 移植版, 是可以自由使用及发布的 Windows 特定头文件和使用 GNU 工具集导入库的集合, 允许在 GNU/Linux 和 Windows 平台生成本地的 Windows 程序, 而不需要第三方 C 运行时库。MinGW 本身也存在多个分支, 可以在 Linux 中编译能在 Windows 中执行的二进制代码。

3. Qt Creator

Qt Creator 是 Qt 的集成开发环境, 能够跨平台运行, 支持的系统包括 Linux(32 位及 64 位)、Mac OS X, 以及 Windows。包括项目生成向导、高级的 C++ 代码编辑器、调试器、图形设计器及类的工具, 集成了 Qt Designer、Qt Assistant、Qt Linguist、图形化的 GDB 调试前端, 集成了 qmake 构建工具等。开发人员利用 Qt 这个应用程序框架, 能更快及轻易地完成开发任务。

4. Qt Designer

Qt Designer 是一个功能强大的 GUI 和窗体构造器, 能够在所有支持平台上以本地化的视图外观与认知, 快速开发高性能的用户界面, 是 Qt 用来设计应用程序图形界面的工具。

5. Qt Assistant

Qt Assistant 又称为 Qt 助手, 是一个可以完全自定义、能重新组织的帮助文件或文档浏览器, 它能与基于 Qt 的应用程序一起运行。开发人员使用它能加快文档的处理过程。

6. Qt Library

Qt Library 是一个拥有超过 400 个 C++ 类，同时类的数量还在不断扩展的类库。它封闭了用于端到端应用程序开发所需要的所有基础结构，包括成熟的对象模型的优秀的 Qt 应用程序接口，以及内容丰富的集合类，具有图形用户界面编程、布局设计、数据库编程、网络编程、XML、国际化、OpenGL 等支持功能。

7. Qt Linguist

Qt Linguist 又称为 Qt 语言专家，是用来消除国际化程序设计流程中障碍的工具。借助于这个工具，开发人员可将应用程序的翻译转换外包给非技术性翻译人员，从而可增加精确度，大大加快软件的本地化速度。

8. Qt dev-tools

Qt dev-tools 包含了 Qt Assistant 及 Qt Linguist 等工具，因此无须单独安装这两个工具。

9. Qt doc

Qt doc 是帮助文档，包含 Qt 中各个类库的详细说明以及丰富的例子程序，可以使用 Qt Assistant 工具来打开阅读。

10. Qt Qtconfig

Qt Qtconfig 是配置 Qt 环境的一个对话框，一般默认就行了，很少有必要去更改。

11. Qt/Embedded

Qt/Embedded 是一个完整的包含 GUI 和基于 Linux 的嵌入式平台开发工具。

12. QWT

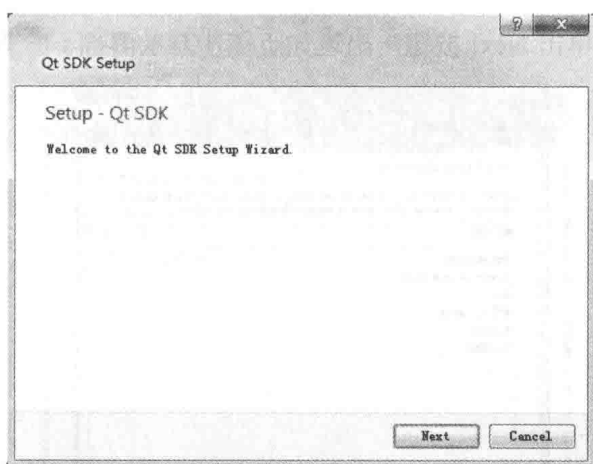
QWT 全称是 Qt Widgets for Technical Applications，是一个基于 LGPL 版权协议的第三方类库，可生成各种统计图。它为具有技术专业背景的程序提供 GUI 组件和一组实用类，以基于 2D 方式的窗体部件来显示数据，输出方式可以是 Curves(曲线)、Slider(滚动条)、Dials(圆盘)、Compasses(仪表盘)等。该工具库基于 Qt 开发，也继承了 Qt 的跨平台特性。

1.4 Qt 开发环境的搭建

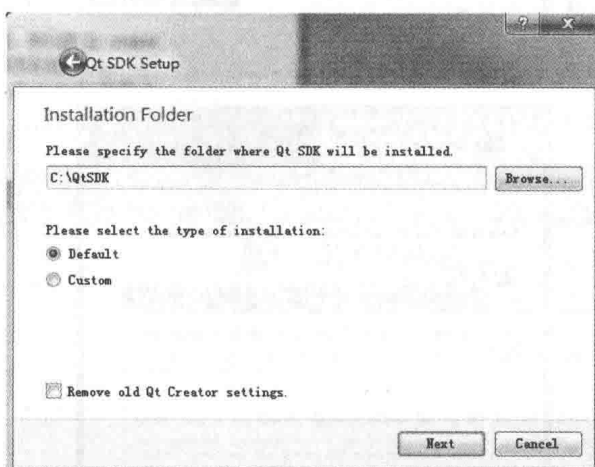
本书讲解 Qt 开发时，遵循了在 Windows 下编辑、在 Linux 下编译的步骤。所以，把 Qt Creator 安装在 Windows 操作系统中，把 qt4.7.2 编译包安装在虚拟机的 Linux 操作系统下，把 Qt 工程目录设置为 Linux 的共享目录。

1.4.1 安装 Qt Creator

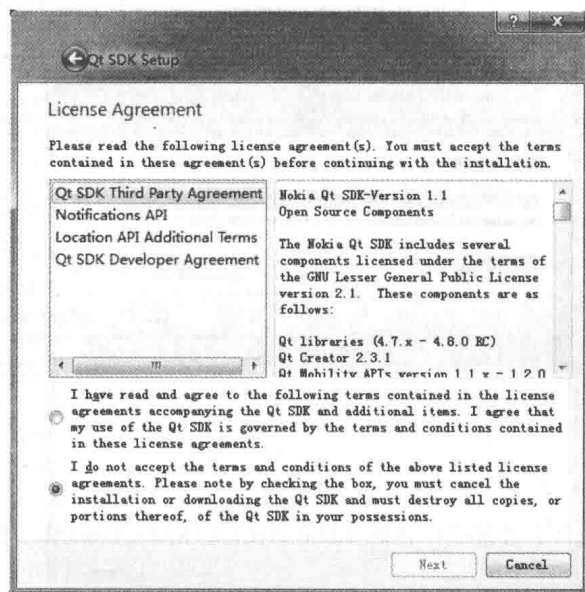
可以在百度中输入“Qt Creator”，然后下载，也可以从随书赠送的资源中找到安装包 Qt_SDK_Win_offline_v1_1_4_en.exe，双击该文件，弹出的安装设置对话框如下图所示。



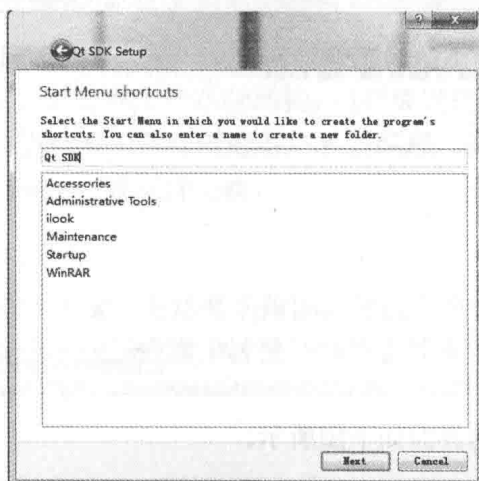
单击 Next 按钮，出现的界面如下图所示。



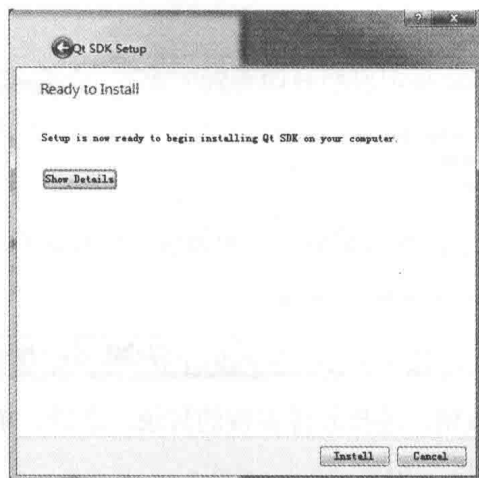
可以通过单击 Browse 按钮，选择软件安装的目录，之后，单击 Next 按钮，出现如下图所示的界面。



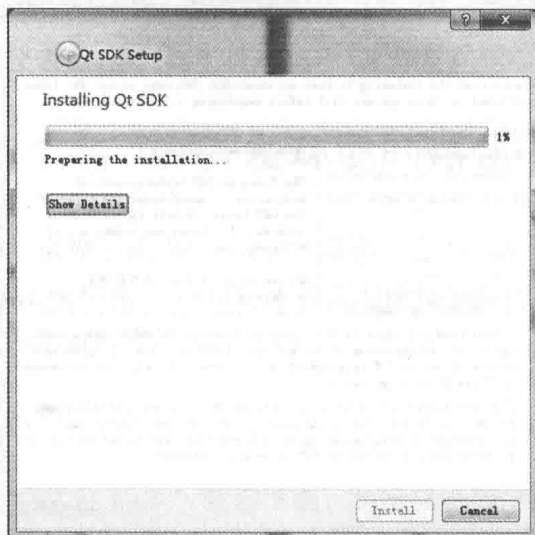
选中单选按钮后，单击 Next 按钮，出现如下图所示的界面。



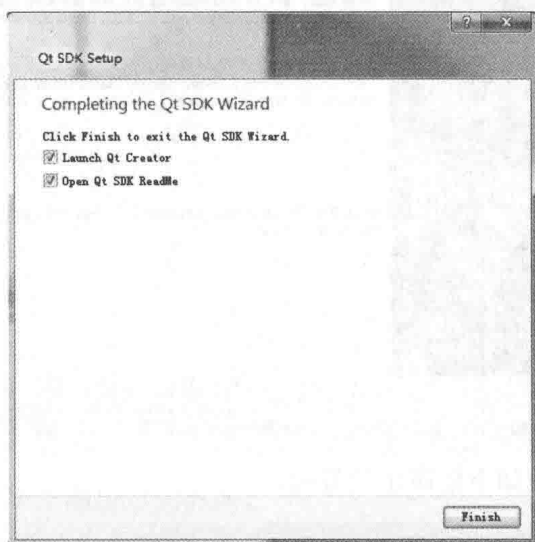
单击 Next 按钮，出现如下图所示的界面。



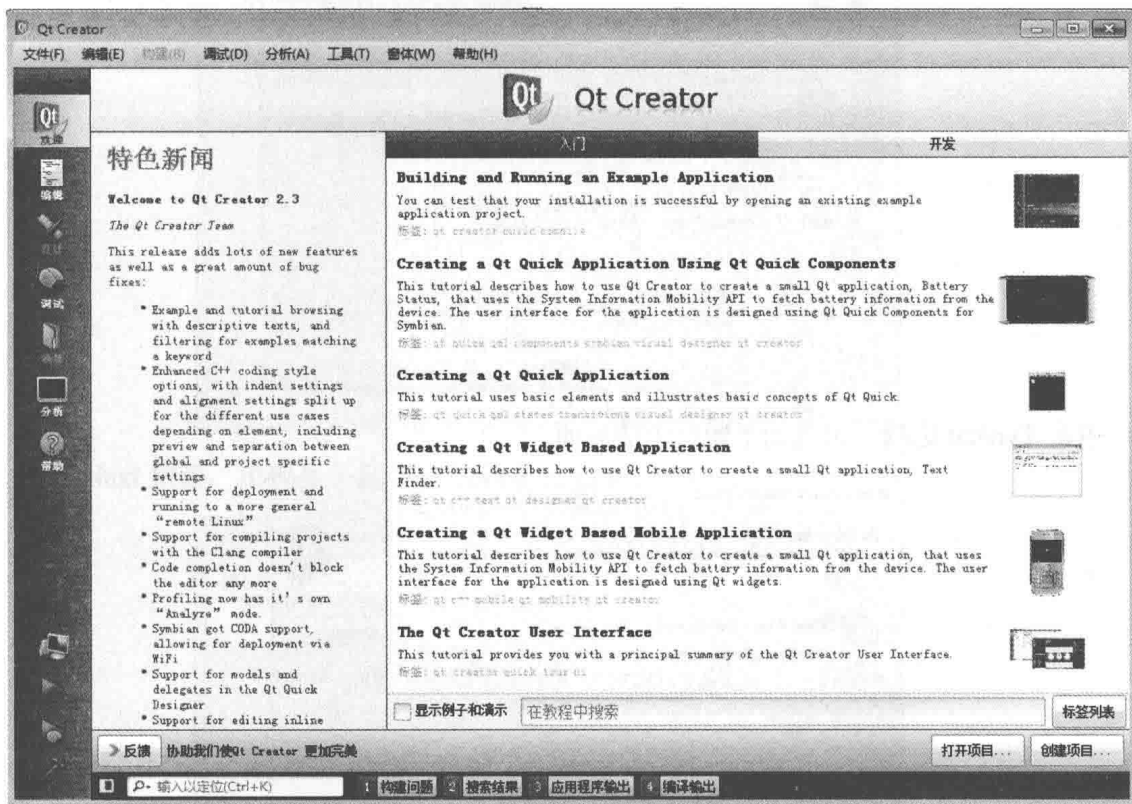
单击 Install 按钮，出现如下图所示的界面。



安装结束后，出现如下图所示的界面。



单击 Finish 按钮，即可安装成功，下图是 Qt Creator 软件的主界面。



1.4.2 安装虚拟机

点击光盘目录下的 VMware-workstation-9.0.1.exe，出现如下图所示的对话框。