

基于 Tilcon 的 VxWorks 图形界面开发技术

张士福

(船舶重工集团公司 723 所,扬州 225001)

摘要:针对嵌入式系统下图形界面开发困难的问题,介绍了一种理想的图形界面开发工具 Tilcon,分析了该工具的工作原理和主要特点,提出了基于 Tilcon 的 VxWorks 图形界面开发方法,给出了一个图形界面应用实例,对基于 Tilcon 的程序开发方法和程序运行流程进行了分析和说明。

关键词:实时操作系统;嵌入式系统;图形界面

中图分类号: TP316.2

文献标识码: A

文章编号: CN32-1413(2011)04-0025-04

Development Technology of VxWorks Graphical Interfaces

Based on Tilcon

ZHANG Shi-fu

(The 723 Institute of CSIC, Yangzhou 225001, China)

Abstract: In view of the problem that the development of graphical interface is difficult in embedded system, this paper introduces the Tilcon as an ideal development tool of graphical interface, analyzes the operating principle and main characteristics of the tool, puts forward the method to develop the VxWorks graphical interface based on Tilcon, presents an application example of graphical interface, analyzes and illuminates the program development method and program operational flow based on Tilcon.

Key words: real time operating system; embedded system; graphical interface

0 引言

VxWorks 是由美国风河公司(Wind River Systems Inc.)开发的一套微内核、高可靠、可裁减的嵌入式实时操作系统,它以高效的实时任务调度、中断管理等优点成为航空、航天、医疗、通信等领域首选的操作系统。VxWorks 开发之初并没有提出很好的、完整的图形显示方案,随着该操作系统应用领域的不断扩展,风河公司改善了图形产品开发中存在的不足,提出了基于嵌入式系统的显示技术风河多媒体库(WindML)。WindML 为 VxWorks 操作系统提供对基本图形、图像和音频的支持,是开发标准用户设备驱动程序的平台,它为用户提供了进行图形界面开发的应用程序接口(API)。利用 WindML 提供的 API 进行图形界面开发对开发人员要求特

别高,需要编写大量的代码,存在开发周期长、界面程序可移植性差等特点。针对嵌入式图形界面开发面临的困难,出现了一些第 3 方的图形开发支持软件,典型的有 Tilcon、Zinc 等,用户利用这些软件工具可以实现“拖-放”式的图形界面设计,把用户从大量编码的图形界面开发方式中解放出来,大大简化了嵌入式系统的图形界面开发过程。

本文重点探讨 Wind River VxWorks 5.5 下利用 Tilcon 进行窗口图形界面开发的方法。

1 Tilcon 的组成及原理

1.1 Tilcon 简介

Tilcon 是加拿大 Tilcon 软件公司推出的多平台图形用户界面(GUI)开发工具,其界面开发包

(IDS) 集成开发环境能够在 VxWorks、Linux、WinCE、QNX 等嵌入式实时操作系统下设计出健壮且交互性极强的“人-机-环”应用软件。

Tilcon 集成了大量的成熟控件,用户无需调用 C 语言 `uglLine()` 等绘图函数以画线填充的方式来完成 GUI 设计,极大地简化了用户开发过程,所开发的程序不需要修改代码就可以从一个操作系统移植到另一个操作系统上。

Tilcon 可以与 Tornado 之间形成无缝链接,图 1 描绘出两者之间的交叉关系。该工具具有功能齐全、界面美观、使用方便的特点,开发的图形应用程序具有可裁剪性强、运行效率高、显示效果好等优点。目前,Tilcon 已广泛应用于航空航天、装甲车、潜艇等应用领域,得到了用户的一致好评,成为 Wind River 开发环境首推的图形界面开发工具,在嵌入式领域有着广泛的应用前景。

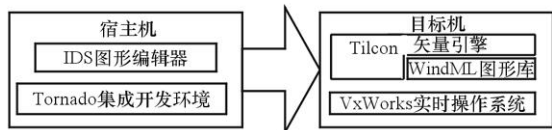


图 1 Tilcon 与 VxWorks 的集成开发环境 Tornado 的关系

1.2 Tilcon 组成

Tilcon 图形界面开发系统主要由三部分组件组成,即嵌入式图形界面引擎 (EVE)、应用程序接口 API 和可视化界面设计工具,其组成及工作原理如图 2 所示。

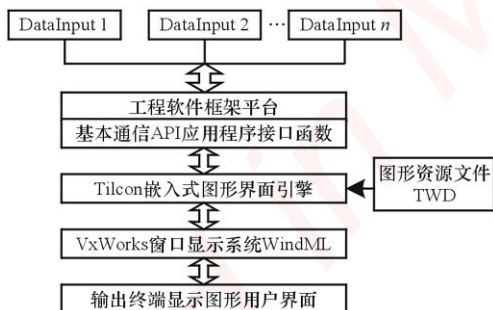


图 2 Tilcon 组成及工作原理

嵌入式图形界面引擎是一个事件驱动的微内核,独立于用户应用程序,其主要任务是驱动、管理所有的用户界面显示及人机交互事件。用户应用程序仅需使用内核提供的实时接口函数,这种图形交互界面和用户程序相互独立的结构提高了用户界面程序的可修改性、可移植性和可维护性。

应用程序接口是一组便于理解、使用的接口函

数,独立于嵌入式平台,为用户应用程序提供操作访问图形界面引擎的接口,支持控件对象的动态创建、对象属性修改等操作。

可视化界面设计工具为用户提供便捷的界面设计环境,用户只需对控件对象面板的控件进行简单的“拖-放”式操作就可以构建复杂的应用图形界面,所有控件的显示风格和功能可以通过对象属性页进行方便的修改。

1.3 Tilcon 工作原理

首先,用户使用 Tilcon 提供的可视化界面设计工具构建图形界面,其以二进制形式保存到与平台无关的 TWD 文件中,用户程序使用 Tilcon API 函数命令图形引擎装载 TWD 文件,并控制图形引擎完成图形界面的显示。当一个 GUI 事件产生时,图形引擎接收到事件后,根据事件类型决定直接处理或通知用户程序处理该事件。另一方面,当用户程序发送 API 命令给 Tilcon 图形引擎的命令队列,用户程序不需要等待这些异步的命令处理完毕就可以直接返回到自己的主要任务中,图形引擎将负责完成这些命令的解释处理。

2 基于 Tilcon 的图形界面开发方法

利用 Tilcon 进行图形界面开发应用时,需对 WindML 与 Tilcon 进行相应的配置、编译和连接,并完成 VxWorks 操作系统配置。基于 Tilcon 的 VxWorks 图形界面及应用程序开发的主要流程如图 3 所示。

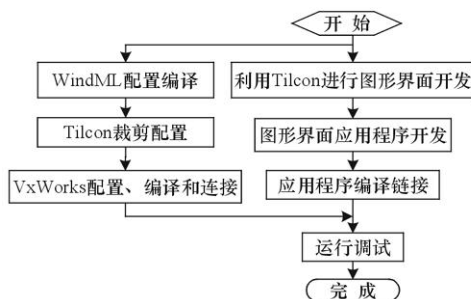


图 3 基于 Tilcon 的 VxWorks 图形界面及应用程序开发流程

2.1 WindML 配置和编译

WindML 是 Wind River 公司推出的多媒体支持库,是 Tilcon 在 VxWorks 系统上工作运行的基础。在使用 Tilcon 之前必须先配置编译 WindML。

进入 Tornado 集成开发环境后,依照 Tools/WindML/Configure 步骤打开相应配置对话框,结

合嵌入式系统的具体硬件构成,按图 4 流程配置对应选项,处理完成后点击“Clean”清除已有编译文件,再单击“Build”按钮重新生成所需 OBJ 文件。若这些操作还不能满足使用要求,还可以用命令行方式配置并编译 WindML 媒体库,如果没有错误提示信息,则会在 Tornado 项目管理区 VxWorks 标签页中出现 WindML 组件。为检验 WindML 媒体库配置、编译是否成功,按如下路径关系 \$(WIND_BASE)\target\src\ugl\example 定位到 ugl demo.c 文件,并在 Tornado 环境下建立 Downloadable 工程,经编译下载运行后,如果能在目标机上出现欢迎界面,则表示 WindML 操作成功。

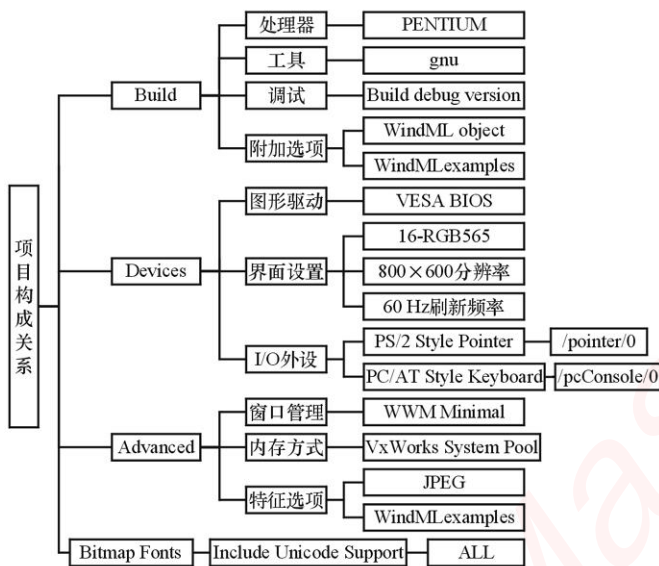


图 4 WindML 配置编译流程

2.2 Tilcon 裁减和配置

Tilcon 提供了图形引擎定制工具 Scalable,利用该工具用户可以灵活选择工程应用中必需的组件,除去不必要的组件,并链接生成较小的 Tilcon 内核,减小 Tilcon 应用程序对系统控件的需求。

Tilcon 定制的主要过程包括:首先将 C:\Tilcon\TSP\ VXWORKS-x86\ target\ lib 下的 obj-PENTIUMgnuTugcl 文件夹复制到 C:\Tornado2.2\ target\ lib 路径下。然后启动配置工具 Scalable,选择 VxWorks 版本信息和处理器类型,并在控件选项中选择所需要的控件,工具会自动生成相应的 Makefile,点击创建按钮后将链接 obj-PENTIUMgnuTugcl 文件夹中的组件生成支持 IDS 应用程序模型的目标静态库文件 tlncore.o 和 tlnapi.o。

2.3 VxWorks 配置和编译

配置和编译好 WindML 和 Tilcon 后,根据工程应用需要在 VxWorks 工程中对 VxWorks 进行配置,需将 C++ Components、WindML 媒体库组件下的 2D Graphics 和 Complete 2D Library 子项以及 POSIX Components 组件下的 POSIX Clocks、POSIX Message Queues、POSIX Semaphores 和 POSIX Timers 子项包含进 VxWorks 工程中,编译生成 VxWorks 镜像。

然后,配置 Downloadable 工程编译环境。在 C/C++ compiler 标签页,具体选项如下: - g - mcpu = pentium - march = Pentium - nostdlib - fno - builtin - fno - defer - pop - I - IC:/Tornado2.2/target/h - IC:/tilcon/include - DCPU = PENTIUM - DTOOL_FAMILY = gnu - DTOOL = gnu - DCC_TRT_VXWORKS,这其中务必去掉 - ansic 选项,因为 VxWorks 可能会包含某些特殊字符信息,如果按 - ansic 语法进行编译就会出现错误提示信息。另外,在 Macros 标签页下的 PRJ_LIBS 宏选项中,将目标静态库文件 tlncore.o 和 tlnapi.o 连接到工程中。

最后,在 Bootable 工程项目 usrAppInit.c 文件中指定 Tilcon 所使用的版本序列号,即:putenv("Tilcon_5_4=/tgtsvr")。

2.4 图形界面设计与编辑

Tilcon 窗口的开发主要通过 Tilcon 图形编辑器来实现,Tilcon 图形编辑器是一组集成的设计工具,用来辅助用户完成图形界面的设计。在使用 Tilcon 图形编辑器构建图形界面的屏幕显示时,以“拖-放”的方式从工具栏添加用户界面对象,并打开属性框编辑对象的属性,通过属性项的设定调整对象的所有外观效果、行为、初始状态、文本方式、显示焦点、菜单、鼠标指针和快捷键等。编辑完成之后,设计的图形界面保存为 TWD 文件,用户可以点击测试按键来检验整个图形界面的运行效果。另外,Tilcon 可以使用用户指定的模拟数据在测试状态下驱动各个对象。

2.5 图形界面应用程序开发

使用 Tilcon IDS 设计好图形界面后,便可开展相应的图形界面应用程序编写工作,基于 Tilcon 的应用程序执行流程如图 5 所示。

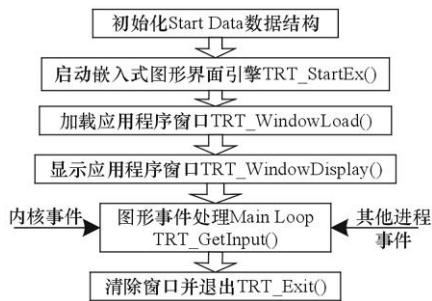


图5 基于 Tilcon 的应用程序执行流程

2.5.1 初始化内核

Tilcon 图形引擎是一个小型的事件驱动图形处理内核,它的任务是更新屏幕显示。它是一个与用户程序相互独立的进程,负责处理所有的屏幕显示和用户输入事件。在图形界面程序启动之前,用户程序必须先初始化 Tilcon 图形引擎内核,设定操作系统类型 Os_Env、对象空间名 AppName、Licence 文件路径等参数。使用 TRT_Start 函数启动一个 Tilcon 图形引擎并与此引擎进行通信。正确完成初始化后,应用程序可以显示用户设计的界面,与用户进行交互。

2.5.2 加载资源文件

执行完必要的初始化操作后,用户程序就可以使用 Tilcon API 函数命令图形引擎装载并显示用户界面窗口或动态创建用户界面。首先指定窗口资源文件所在的路径,然后通过调用 TRT_WindowLoad 函数把图形界面资源文件加载到图形引擎中,这样用户程序就可以通过 Tilcon API 和 Tilcon 图形引擎进行通信来控制图形界面的显示和刷新。

2.5.3 窗口显示管理

窗口管理主要指对已加载到图形引擎中的窗口进行显示、隐藏、删除等操作管理。Tilcon 提供了一系列对窗口管理的 API 函数,调用 TRT_WindowDisplay 函数可实现把用户定义的窗口显示到屏幕上的操作。当不再需要某个窗口的显示时,可通过调用 TRT_WindowHide 函数把窗口隐藏起来,或调用 TRT_WindowDelete 函数直接把窗口删除,删除窗口的同时,所有窗口上的控件也会被一起清除。

2.5.4 窗口事件处理

基于 Tilcon 图形界面用户程序是通过一个主事件循环来处理相应窗口事件,实现人机交互功能。在主循环中,TRT_GetInput 函数等待直到图形引擎报告一个事件,调用窗口事件处理函数进行处理。用户可以配置 TRT_GetInput 函数接收来自其

它进程的消息,或设置为等待某种事件信号量的到来,这使得用户程序可以和其它进程进行通信。当某个特定的事件出现时,用户程序中中止循环并调用 TRT_Exit 函数通知图形引擎结束进程。

2.5.5 应用示例

用 Tilcon 图形开发工具在 VxWorks 操作系统下,结合 Tornado 工具开发的一个示波器面板如图6所示。模拟测试时,Tilcon 画图对象 Charts 画正弦曲线,对参数进行设置,可看到不同的测试效果。

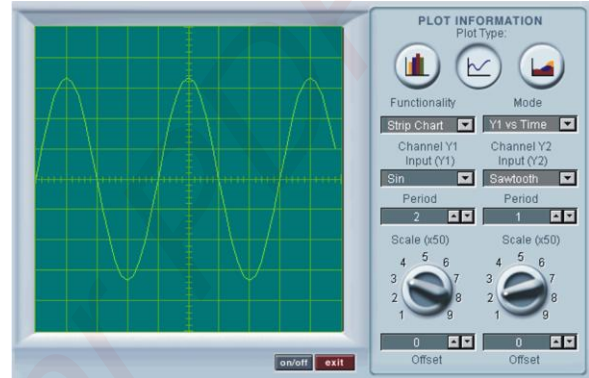


图6 示波器面板设计

3 结束语

Tilcon 作为 Wind River 新支持的革命性图形界面开发工具,为用户提供了功能齐全的控件,拥有出色的高效率、模块化和可裁剪交互面板设计能力,其顶层应用界面与底层功能驱动代码相互独立,因此具备很强的跨平台移植能力,将开发人员从繁琐的嵌入式系统界面开发中脱离出来,正被广泛地应用到嵌入式实时操作系统领域。本文介绍了 Tilcon 图形界面开发工具,探讨了利用该工具设计用户界面及开发应用程序的过程。目前,Tilcon 已在某舰指控系统软件开发中得到应用,工程应用表明,Tilcon 在功能、性能方面都能满足基于 VxWorks 的实时系统图形界面开发要求,是理想的嵌入式系统图形界面开发解决方案。

参考文献

- [1] Wind River Systems Inc. VxWorks Programmer's Guide 5.5 [M]. USA, Alameda: Wind River Systems Inc, 2002.
- [2] Tilcon Software Ltd. Tilcon Programmer's Guide [M]. CA, Ottawa: Tilcon Software Ltd, 2005.
- [3] Wind River Systems Inc. WindML SDK Programmer's Guide 3.0 [M]. USA, Alameda: Wind River Systems Inc, 2002.

嵌入式资源免费下载

总线协议:

1. [基于 PCIe 驱动程序的数据传输卡 DMA 传输](#)
2. [基于 PCIe 总线协议的设备驱动开发](#)
3. [CANopen 协议介绍](#)
4. [基于 PXI 总线 RS422 数据通信卡 WDM 驱动程序设计](#)
5. [FPGA 实现 PCIe 总线 DMA 设计](#)
6. [PCI Express 协议实现与验证](#)
7. [VPX 总线技术及其实现](#)
8. [基于 Xilinx FPGA 的 PCIE 接口实现](#)
9. [基于 PCI 总线的 GPS 授时卡设计](#)
10. [基于 CPCI 标准的 6U 信号处理平台的设计](#)
11. [USB30 电路保护](#)
12. [USB30 协议分析与框架设计](#)
13. [USB 30 中的 CRC 校验原理及实现](#)
14. [基于 CPLD 的 UART 设计](#)
15. [IPMI 在 VPX 系统中的应用与设计](#)
16. [基于 CPCI 总线的 PMC 载板设计](#)
17. [基于 VPX 总线的工件台运动控制系统研究与开发](#)
18. [PCI Express 流控机制的研究与实现](#)
19. [UART16C554 的设计](#)
20. [基于 VPX 的高性能计算机设计](#)
21. [基于 CAN 总线技术的嵌入式网关设计](#)
22. [Visual C 串行通讯控件使用方法与技巧的研究](#)
23. [IEEE1588 精密时钟同步关键技术研究](#)
24. [GPS 信号发生器射频模块的一种实现方案](#)
25. [基于 CPCI 接口的视频采集卡的设计](#)
26. [基于 VPX 的 3U 信号处理平台的设计](#)
27. [基于 PCI Express 总线 1394b 网络传输系统 WDM 驱动设计](#)
28. [AT89C52 单片机与 ARINC429 航空总线接口设计](#)
29. [基于 CPCI 总线多 DSP 系统的高速主机接口设计](#)
30. [总线协议中的 CRC 及其在 SATA 通信技术中的应用](#)
31. [基于 FPGA 的 SATA 硬盘加解密控制器设计](#)
32. [Modbus 协议在串口通讯中的研究及应用](#)
33. [高可用的磁盘阵列 Cache 的设计和实现](#)
34. [RAID 阵列中高速 Cache 管理的优化](#)

35. [一种新的基于 RAID 的 CACHE 技术研究与实现](#)
36. [基于 PCIE-104 总线的高速数据接口设计](#)
37. [基于 VPX 标准的 RapidIO 交换和 Flash 存储模块设计](#)
38. [北斗卫星系统在海洋工程中的应用](#)
39. [北斗卫星系统在远洋船舶上应用的研究](#)
40. [基于 CPCI 总线的红外实时信号处理系统](#)
41. [硬件实现 RAID 与软件实现 RAID 的比较](#)
42. [基于 PCI Express 总线系统的热插拔设计](#)
43. [基于 RAID5 的磁盘阵列 Cache 的研究与实现](#)
44. [基于 PCI 总线的 MPEG2 码流播放卡驱动程序开发](#)
45. [基于磁盘阵列引擎的 RAID5 小写性能优化](#)
46. [基于 IEEE1588 的时钟同步技术研究](#)
47. [基于 Davinci 平台的 SD 卡读写优化](#)
48. [基于 PCI 总线的图像处理及传输系统的设计](#)
49. [串口和以太网通信技术在油液在线监测系统中的应用](#)
50. [USB30 数据传输协议分析及实现](#)
51. [IEEE 1588 协议在工业以太网中的实现](#)
52. [基于 USB30 的设备自定义请求实现方法](#)
53. [IEEE1588 协议在网络测控系统中的应用](#)
54. [USB30 物理层中弹性缓冲的设计与实现](#)
55. [USB30 的高速信息传输瓶颈研究](#)
56. [基于 IPv6 的 UDP 通信的实现](#)
57. [一种基于 IPv6 的流媒体传送方案研究与实现](#)
58. [基于 IPv4-IPv6 双栈的 MODBUS-TCP 协议实现](#)
59. [RS485CAN 网关设计与实现](#)
60. [MVB 周期信息的实时调度](#)
61. [RS485 和 PROFINET 网关设计](#)
62. [基于 IPv6 的 Socket 通信的实现](#)
63. [MVB 网络重复器的设计](#)
64. [一种新型 MVB 通信板的探究](#)
65. [具有 MVB 接口的输入输出设备的分析](#)
66. [基于 STM32 的 GSM 模块综合应用](#)
67. [基于 ARM7 的 MVB CAN 网关设计](#)

VxWorks:

1. [基于 VxWorks 的多任务程序设计](#)
2. [基于 VxWorks 的数据采集存储装置设计](#)
3. [Flash 文件系统分析及其在 VxWorks 中的实现](#)

4. [VxWorks 多任务编程中的异常研究](#)
5. [VxWorks 应用技巧两例](#)
6. [一种基于 VxWorks 的飞行仿真实时管理系统](#)
7. [在 VxWorks 系统中使用 TrueType 字库](#)
8. [基于 FreeType 的 VxWorks 中文显示方案](#)
9. [基于 Tilcon 的 VxWorks 简单动画开发](#)
10. [基于 Tilcon 的某武器显控系统界面设计](#)
11. [基于 Tilcon 的综合导航信息处理装置界面设计](#)
12. [VxWorks 的内存配置和管理](#)
13. [基于 VxWorks 系统的 PCI 配置与应用](#)
14. [基于 MPC8270 的 VxWorks BSP 的移植](#)
15. [Bootrom 功能改进经验谈](#)
16. [基于 VxWorks 嵌入式系统的中文平台研究与实现](#)
17. [VxBus 的 A429 接口驱动](#)
18. [基于 VxBus 和 MPC8569E 千兆网驱动开发和实现](#)
19. [一种基于 vxBus 的 PPC 与 FPGA 高速互联的驱动设计方法](#)
20. [基于 VxBus 的设备驱动开发](#)
21. [基于 VxBus 的驱动程序架构分析](#)
22. [基于 VxBus 的高速数据采集卡驱动程序开发](#)
23. [Vxworks 下的冗余 CAN 通讯模块设计](#)
24. [WindML 工业平台下开发 S1d13506 驱动及显示功能的实现](#)
25. [WindML 中 Mesa 的应用](#)
26. [VxWorks 下图形用户界面开发中双缓冲技术应用](#)
27. [VxWorks 上的一种 GUI 系统的设计与实现](#)
28. [VxWorks 环境下 socket 的实现](#)
29. [VxWorks 的 WindML 图形界面程序的框架分析](#)
30. [VxWorks 实时操作系统及其在 PC104 下以太网编程的应用](#)
31. [实时操作系统任务调度策略的研究与设计](#)
32. [军事指挥系统中 VxWorks 下汉字显示技术](#)
33. [基于 VxWorks 实时控制系统中文交互界面开发平台](#)
34. [基于 VxWorks 操作系统的 WindML 图形操控界面实现方法](#)
35. [基于 GPU FPGA 芯片原型的 VxWorks 下驱动软件开发](#)
36. [VxWorks 下的多串口卡设计](#)
37. [VxWorks 内存管理机制的研究](#)
38. [T9 输入法在 Tilcon 下的实现](#)
39. [基于 VxWorks 的 WindML 图形界面开发方法](#)
40. [基于 Tilcon 的 IO 控制板可视化测试软件的设计和实现](#)
41. [基于 VxWorks 的通信服务器实时多任务软件设计](#)
42. [基于 VXWORKS 的 RS485MVB 网关的设计与实现](#)
43. [实时操作系统 VxWorks 在微机保护中的应用](#)
44. [基于 VxWorks 的多任务程序设计及通信管理](#)

Linux:

1. [Linux 程序设计第三版及源代码](#)
2. [NAND FLASH 文件系统的设计与实现](#)
3. [多通道串行通信设备的 Linux 驱动程序实现](#)
4. [Zsh 开发指南-数组](#)
5. [常用 GDB 命令中文速览](#)
6. [嵌入式 C 进阶之道](#)
7. [Linux 串口编程实例](#)
8. [基于 Yocto Project 的嵌入式应用设计](#)
9. [Android 应用的反编译](#)
10. [基于 Android 行为的加密应用系统研究](#)
11. [嵌入式 Linux 系统移植步步通](#)
12. [嵌入式 C++ 语言精华文章集锦](#)
13. [基于 Linux 的高性能服务器端的设计与研究](#)
14. [S3C6410 移植 Android 内核](#)
15. [Android 开发指南中文版](#)
16. [图解 Linux 操作系统架构设计与实现原理（第二版）](#)
17. [如何在 Ubuntu 和 Linux Mint 下轻松升级 Linux 内核](#)
18. [Android 简单 mp3 播放器源码](#)
19. [嵌入式 Linux 系统实时性的研究](#)
20. [Android 嵌入式系统架构及内核浅析](#)
21. [基于嵌入式 Linux 操作系统内核实时性的改进方法研究](#)
22. [Linux TCP IP 协议详解](#)
23. [Linux 桌面环境下内存去重技术的研究与实现](#)
24. [掌握 Android 7.0 新增特性 Quick Settings](#)
25. [Android 应用逆向分析方法研究](#)
26. [Android 操作系统的课程教学](#)
27. [Android 智能手机操作系统的研究](#)
28. [Android 英文朗读功能的实现](#)
29. [基于 Yocto 订制嵌入式 Linux 发行版](#)
30. [基于嵌入式 Linux 的网络设备驱动设计与实现](#)
31. [如何高效学习嵌入式](#)
32. [基于 Android 平台的 GPS 定位系统的设计与实现](#)
33. [LINUX ARM 下的 USB 驱动开发](#)
34. [Linux 下基于 I2C 协议的 RTC 驱动开发](#)
35. [嵌入式下 Linux 系统设备驱动程序的开发](#)
36. [基于嵌入式 Linux 的 SD 卡驱动程序的设计与实现](#)
37. [Linux 系统中进程调度策略](#)

38. [嵌入式 Linux 实时性方法](#)
39. [基于实时 Linux 计算机联锁系统实时性分析与改进](#)
40. [基于嵌入式 Linux 下的 USB30 驱动程序开发方法研究](#)
41. [Android 手机应用开发之音乐资源播放器](#)
42. [Linux 下以太网的 IPv6 隧道技术的实现](#)
43. [Research and design of mobile learning platform based on Android](#)
44. [基于 linux 和 Qt 的串口通信调试器调的设计及应用](#)
45. [在 Linux 平台上基于 QT 的动态图像采集系统的设计](#)
46. [基于 Android 平台的医护查房系统的研究与设计](#)
47. [基于 Android 平台的软件自动化监控工具的设计开发](#)
48. [基于 Android 的视频软硬解码及渲染的对比研究与实现](#)
49. [基于 Android 移动设备的加速度传感器技术研究](#)
50. [基于 Android 系统振动测试仪研究](#)
51. [基于缓存竞争优化的 Linux 进程调度策略](#)

Windows CE:

1. [Windows CE.NET 下 YAFFS 文件系统 NAND Flash 驱动程序设计](#)
2. [Windows CE 的 CAN 总线驱动程序设计](#)
3. [基于 Windows CE.NET 的 ADC 驱动程序实现与应用的研究](#)
4. [基于 Windows CE.NET 平台的串行通信实现](#)
5. [基于 Windows CE.NET 下的 GPRS 模块的研究与开发](#)
6. [win2k 下 NTFS 分区用 ntldr 加载进 dos 源代码](#)
7. [Windows 下的 USB 设备驱动程序开发](#)
8. [WinCE 的大容量程控数据传输解决方案设计](#)
9. [WinCE6.0 安装开发详解](#)
10. [DOS 下仿 Windows 的自带计算器程序 C 源码](#)
11. [G726 局域网语音通话程序和源代码](#)
12. [WinCE 主板加载第三方驱动程序的方法](#)
13. [WinCE 下的注册表编辑程序和源代码](#)
14. [WinCE 串口通信源代码](#)
15. [WINCE 的 SD 卡程序\[可实现读写的源码\]](#)
16. [基于 WinCE 的 BootLoader 研究](#)
17. [Windows CE 环境下无线网卡的自动安装](#)
18. [基于 Windows CE 的可视电话的研究与实现](#)
19. [基于 WinCE 的嵌入式图像采集系统设计](#)
20. [基于 ARM 与 WinCE 的掌纹鉴别系统](#)
21. [DCOM 协议在网络冗余环境下的应用](#)
22. [Windows XP Embedded 在变电站通信管理机中的应用](#)

23. [XPE 在多功能显控台上的开发与应用](#)
24. [基于 Windows XP Embedded 的 LKJ2000 仿真系统设计与实现](#)
25. [虚拟仪器的 Windows XP Embedded 操作系统开发](#)
26. [基于 EVC 的嵌入式导航电子地图设计](#)
27. [基于 XPEmbedded 的警务区 SMS 指挥平台的设计与实现](#)
28. [基于 XPE 的数字残币兑换工具开发](#)

PowerPC:

1. [Freescale MPC8536 开发板原理图](#)
2. [基于 MPC8548E 的固件设计](#)
3. [基于 MPC8548E 的嵌入式数据处理系统设计](#)
4. [基于 PowerPC 嵌入式网络通信平台的实现](#)
5. [PowerPC 在车辆显控系统中的应用](#)
6. [基于 PowerPC 的单板计算机的设计](#)
7. [用 PowerPC860 实现 FPGA 配置](#)
8. [基于 MPC8247 嵌入式电力交换系统的设计与实现](#)
9. [基于设备树的 MPC8247 嵌入式 Linux 系统开发](#)
10. [基于 MPC8313E 嵌入式系统 UBoot 的移植](#)
11. [基于 PowerPC 处理器 SMP 系统的 UBoot 移植](#)
12. [基于 PowerPC 双核处理器嵌入式系统 UBoot 移植](#)
13. [基于 PowerPC 的雷达通用处理机设计](#)
14. [PowerPC 平台引导加载程序的移植](#)
15. [基于 PowerPC 嵌入式内核的多串口通信扩展设计](#)
16. [基于 PowerPC 的多网口系统抗干扰设计](#)
17. [基于 MPC860T 与 VxWorks 的图形界面设计](#)
18. [基于 MPC8260 处理器的 PPMC 系统](#)
19. [基于 PowerPC 的控制器研究与设计](#)
20. [基于 PowerPC 的模拟量输入接口扩展](#)
21. [基于 PowerPC 的车载通信系统设计](#)
22. [基于 PowerPC 的嵌入式系统中通用 IO 口的扩展方法](#)
23. [基于 PowerPC440GP 型微控制器的嵌入式系统设计与研究](#)
24. [基于双 PowerPC 7447A 处理器的嵌入式系统硬件设计](#)
25. [基于 PowerPC603e 通用处理模块的设计与实现](#)
26. [嵌入式微机 MPC555 驻留片内监控器的开发与实现](#)
27. [基于 PowerPC 和 DSP 的电能质量在线监测装置的研制](#)
28. [基于 PowerPC 架构多核处理器嵌入式系统硬件设计](#)

ARM:

1. [基于 DiskOnChip 2000 的驱动程序设计及应用](#)
2. [基于 ARM 体系的 PC-104 总线设计](#)
3. [基于 ARM 的嵌入式系统中断处理机制研究](#)
4. [设计 ARM 的中断处理](#)
5. [基于 ARM 的数据采集系统并行总线的驱动设计](#)
6. [S3C2410 下的 TFT LCD 驱动源码](#)
7. [STM32 SD 卡移植 FATFS 文件系统源码](#)
8. [STM32 ADC 多通道源码](#)
9. [ARM Linux 在 EP7312 上的移植](#)
10. [ARM 经典 300 问](#)
11. [基于 S5PV210 的频谱监测设备嵌入式系统设计与实现](#)
12. [Uboot 中 start.S 源码的指令级的详尽解析](#)
13. [基于 ARM9 的嵌入式 Zigbee 网关设计与实现](#)
14. [基于 S3C6410 处理器的嵌入式 Linux 系统移植](#)
15. [CortexA8 平台的 \$\mu\$ C-OS II 及 LwIP 协议栈的移植与实现](#)
16. [基于 ARM 的嵌入式 Linux 无线网卡设备驱动设计](#)
17. [ARM S3C2440 Linux ADC 驱动](#)
18. [ARM S3C2440 Linux 触摸屏驱动](#)
19. [Linux 和 Cortex-A8 的视频处理及数字微波传输系统设计](#)
20. [Nand Flash 启动模式下的 Uboot 移植](#)
21. [基于 ARM 处理器的 UART 设计](#)
22. [ARM CortexM3 处理器故障的分析与处理](#)
23. [ARM 微处理器启动和调试浅析](#)
24. [基于 ARM 系统下映像文件的执行与中断运行机制的实现](#)
25. [中断调用方式的 ARM 二次开发接口设计](#)
26. [ARM11 嵌入式系统 Linux 下 LCD 的驱动设计](#)
27. [Uboot 在 S3C2440 上的移植](#)
28. [基于 ARM11 的嵌入式无线视频终端的设计](#)
29. [基于 S3C6410 的 Uboot 分析与移植](#)
30. [基于 ARM 嵌入式系统的高保真无损音乐播放器设计](#)
31. [UBoot 在 Mini6410 上的移植](#)
32. [基于 ARM11 的嵌入式 Linux NAND FLASH 模拟 U 盘挂载分析与实现](#)
33. [基于 ARM11 的电源完整性分析](#)
34. [基于 ARM S3C6410 的 uboot 分析与移植](#)
35. [基于 S5PC100 移动视频监控终端的设计与实现](#)

Hardware:

1. [DSP 电源的典型设计](#)
2. [高频脉冲电源设计](#)
3. [电源的综合保护设计](#)
4. [任意波形电源的设计](#)
5. [高速 PCB 信号完整性分析及应用](#)
6. [DM642 高速图像采集系统的电磁干扰设计](#)
7. [使用 COMExpress Nano 工控板实现 IP 调度设备](#)
8. [基于 COM Express 架构的数据记录仪的设计与实现](#)
9. [基于 COM Express 的信号系统逻辑运算单元设计](#)
10. [基于 COM Express 的回波预处理模块设计](#)
11. [基于 X86 平台的简单多任务内核的分析与实现](#)
12. [基于 UEFI Shell 的 PreOS Application 的开发与研究](#)
13. [基于 UEFI 固件的恶意代码防范技术研究](#)
14. [MIPS 架构计算机平台的支持固件研究](#)
15. [基于 UEFI 固件的攻击验证技术研究](#)
16. [基于 UEFI 的 Application 和 Driver 的分析与开发](#)
17. [基于 UEFI 的可信 BIOS 研究与实现](#)
18. [基于 UEFI 的国产计算机平台 BIOS 研究](#)
19. [基于 UEFI 的安全模块设计分析](#)
20. [基于 FPGA Nios II 的等精度频率计设计](#)
21. [基于 FPGA 的 SOPC 设计](#)
22. [基于 SOPC 基本信号产生器的设计与实现](#)
23. [基于龙芯平台的 PMON 研究与开发](#)
24. [基于 X86 平台的嵌入式 BIOS 可配置设计](#)
25. [基于龙芯 2F 架构的 PMON 分析与优化](#)
26. [CPU 与 GPU 之间接口电路的设计与实现](#)
27. [基于龙芯 1A 平台的 PMON 源码编译和启动分析](#)
28. [基于 PC104 工控机的嵌入式直流监控装置的设计](#)
29. [GPGPU 技术研究与发展](#)
30. [GPU 实现的高速 FIR 数字滤波算法](#)
31. [一种基于 CPUGPU 异构计算的混合编程模型](#)
32. [面向 OpenCL 模型的 GPU 性能优化](#)
33. [基于 GPU 的 FDTD 算法](#)
34. [基于 GPU 的瑕疵检测](#)
35. [基于 GPU 通用计算的分析与研究](#)

Programming:

1. [计算机软件基础数据结构 - 算法](#)
2. [高级数据结构对算法的优化](#)
3. [零基础学算法](#)
4. [Linux 环境下基于 TCP 的 Socket 编程浅析](#)
5. [Linux 环境下基于 UDP 的 socket 编程浅析](#)
6. [基于 Socket 的网络编程技术及其实现](#)
7. [数据结构考题 - 第 1 章 绪论](#)
8. [数据结构考题 - 第 2 章 线性表](#)
9. [数据结构考题 - 第 2 章 线性表 - 答案](#)