

基于 Tilcon 的嵌入式系统人机界面开发技术

冯文飞

中国电子科技集团公司第二十七研究所, 郑州 450047

摘要: 为克服嵌入式系统中图形界面开发难度高、工作量大的困难, 文章采用支持多平台的图形用户界面开发环境 Tilcon, 本文介绍 Tilcon 的结构、开发过程及在嵌入式应用系统中的设计方法, 采用该方法可简化用户编程, 提高开发效率。

关键词: 嵌入式实时操作系统; 图形界面; VxWorks; Tilcon; WindML

中图分类号: TP316.2

文献标识码: A

Design of Graphic User Interface of Embeded System Based on Tilcon FENG Wen-Fei

(The 27th Research Institute of China Electronics Technology
Group Corporation, Zhengzhou 450047, China)

Abstract: To ease the hardness and to reduce the amount of work in graphical interface development in embedded system, this paper introduces the graphical user interface development environment tilcon, which supports multiple-platform, giving its details of the structure, development processes and design method in embedded application system. Using this method proposed can simplify the programming process and enhance the development efficiency.

Key words: Embeded RTOS; Graphical Interface; VxWorks; Tilcon; WindML

1 引言

VxWorks 是风河公司开发的一套具有可裁剪的微内核、高性能的实时操作系统^[1]。它以其良好的可靠性和卓越的实时性被广泛应用于军事、航空、航天等领域。

Tilcon 是风河公司提供的一款跨平台的人机界面开发环境, 可实现在嵌入式系统下的图形界面开发能力。用户利用它可快速、便捷的完成原型构建并对其进行测试, 高效率的开发出应用系统的用户图形界面^[2]。

开发宿主端, Tilcon 可安装集成于 VxWorks6.x 的开发平台 WorkBench 上; 而在目标设备端, Tilcon 内核在 VxWorks 下的应用基于风河的 WindML。

WindML(Wind Media Library)是 VxWorks 嵌入式系统的多媒体应用程序, 它提供最基本的图形、视频、音频技术和一个标准的驱动程序框架, 同时

提供了二维图形, 如点、线、圆、矩形、多边形等的绘制函数和国际化的字体支持^[3]。

2 Tilcon 的组成与结构

Tilcon 开发环境主要包括以下三个部分: 可视化图形界面设计工具——图形编辑器 IDS, Tilcon 的嵌入式图形引擎(Embedded Vector Engines)和一组与平台无关的 API 函数。其中, 嵌入式的图形引擎是其核心部分。Tilcon 的组成和布局示意图如图 1 所示。

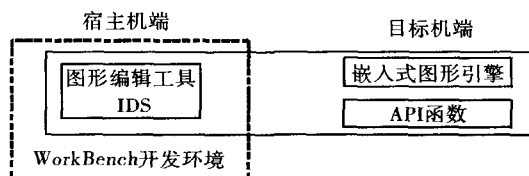


图 1 Tilcon 的组成和布局示意图

IDS 为用户提供了丰富的控件, 用户只需通过拖动控件、设置控件的显示和运行属性, 就可以组建自己所需的应用图形界面; 用户在完成图形界

面设计后,可将其存储为一个数据资源文件(文件扩展名为 twd),无需生成应用程序代码。

Tilcon 的嵌入式图形引擎是一个小型的、事件驱动的图形处理内核,是一个与用户的应用程序相互独立的任务或进程,它负责接受用户事件驱动,处理所有的屏幕显示和用户输入事件。

API 函数用来提供用户程序和 Tilcon 嵌入式图形引擎之间的通信功能,支持用户动态的创建对象,以及改变显示对象属性的控制功能。即用户应用程序通过调用相应的 API 函数实现控制图形界面显示相关数据信息的功能。

3 VxWorks 下应用 Tilcon 进行图形界面开发的过程

用户应用 Tilcon 完成图形界面的设计结果是一个文件扩展名为 twd 数据资源文件,为了要在目标设备上实现图形界面显示,这个生成的数据资源文件当然需要放置在目标机的存储设备中。

VxWorks 操作系统下,完成用户应用程序开发后,需要与 Tilcon 的嵌入式图形引擎、API 一起编译,连接为最终产品文件,一般为 VxWorks 映像文件。这个 VxWorks 映像文件需与 Tilcon 图形资源文件放置在用户应用程序指定的路径下,用户应用程序和图形界面的就可通过互相通信,实现目标设备的显控功能。

VxWorks 操作系统下,应用程序和图形界面的开发过程如图 2 所示。

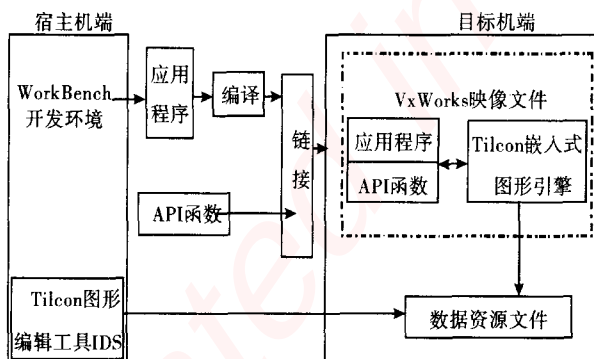


图 2 WorkBench 平台下应用 Tilcon 开发的过程和产品结构示意图

3.1 开发宿主机端的应用

1) 开发环境安装及配置

以应用系统开发所用的 Tilcon 5.7 和 WorkBench 3.1 / VxWorks 6.7 环境为例,描述 VxWorks

操作系统下应用 Tilcon 进行图形界面开发的过程。

安装 WorkBench 3.1 开发环境后,继续安装 Tilcon 就会在 WorkBench 环境中单独生成一个菜单选项,将 Tilcon 集成在 WorkBench 环境中。以后的图形界面编辑环境就可通过点击该菜单项进入。

Tilcon 提供了图形引擎定制工具 Scalable^[3],利用该工具用户可以灵活选择工程应用中必需的组件,链接生成较小的 Tilcon 内核,减少 Tilcon 应用程序对系统空间的需求。Tilcon 定制的主要过程包括:启动配置工具 Scalable,选择 VxWorks 6.7 系统平台和 PENTIUM 4 处理器后,将链接相关组件并生成相应的 Tilcon 目标库文件 tlncore.o 和 tlnapi.o。

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, POSIX Clocks, POSIX Message Queues, POSIX Semaphores, POSIX Timers 选择包含进 VxWorks 工程中,编译生成 VxWorks 映像。另外,用户可以使用 WorkBench 工程管理工具把 WindML 和 Tilcon 连接到 VxWorks 映像里。对于 Tilcon 图形界面的开发,至少需要将 WindML 的相关组件 2D graphics 和 complete 2D library 选择包含进 VxWorks 工程中,并将 Tilcon 内核目标文件 tlncore.o 和 tlnapi.o 连接到工程中^[4]。

2) 图形界面设计及数据资源文件生成

Tilcon 的 IDS 使用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 设计工具)构建图形界面时,使用鼠标以“拖-放”的方式从工具栏添加用户界面对象,并点击鼠标右键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(例如颜色、标签、边框、材质和字体等)。通过对象的属性项还可以指定对象的行为,包括大多数功能、事件响应方式,初始状态,文本换行方式,显示焦点,帮助对话框,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 数据资源文件。

此数据资源文件包括一系列的标签、坐标和 API 码率,并不产生代码,因此保证了数据资源文件的独立性,从而使得图形界面的变化只影响数据资源文件的改变,数据资源文件与操作系统无关,不需要重新编译,极大地提高了开发效率,并支持在不同平台间的移植。

用户还可以在开发环境下,测试整个图形界

面的运行效果, Tilcon 可以使用指定的模拟数据在测试状态下驱动各个对象。

3) 图形界面应用程序开发

使用 IDS 完成图形界面设计后, 便可开展相应的图形界面应用程序编写工作, 基于 Tilcon 的图形界面应用程序执行流程如图 3 所示。

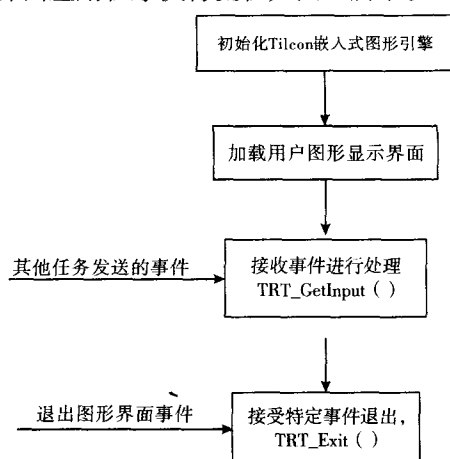


图 3 基于 Tilcon 的图形界面应用程序执行流程图

在图形界面程序启动之前, 用户应用程序必须先初始化 Tilcon 图形引擎内核, 设定操作系统类型 `Os_Env`、对象空间名 `AppName (object space)`、Licence 文件路径等参数, 并使用 `TRT_Start` 函数启动 Tilcon 图形引擎。

在执行完必要的初始化操作之后, 应用程序就可以使用 Tilcon API 函数命令图形引擎装载并显示用户界面窗口或者动态创建用户界面。首先指定窗口资源文件所在的路径, 然后通过 `TRT_WindowLoad` 把图形界面资源文件加载到图形引擎中, 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。

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。运行时, 用户应用程序完成 Tilcon 图形引擎初始化, 窗口加载显示后, 进入主循环等待事件的产生。在主循环中, `TRT_GetInput` 函数等待直到图形引擎报告一个事件, 应用程序定义了窗口事件处理函数, 当图形引擎发出事件通知时窗口事件处理函数被调用。用户可以配置 `TRT_GetInput` 函数来接收来自其他进程的消息, 或者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, 用户程序会中止循环并调用 `TRT_Exit` 函数通知图形引擎结束程序。

3.2 目标设备端的应用

1) 图形显示所需文件配置

将 Tilcon 的 license 文件 (`licensev.til`) 拷贝到目标机文件系统的根目录下。

将 `*.twd` 图形数据资源文件和应用程序 `VxWorks` 映像文件放置在目标机应用程序指定的文件系统相应路径下。

2) 图形显示工作原理应用程序加载启动后, 按图 2 所示的工作流程执行图形界面应用程序。包含在 `Vxworks` 映像文件内的 Tilcon 的嵌入式图形引擎具备自己的绘制能力, 所有的显示对象都是 Tilcon 对象, 与操作系统无关。嵌入式图形引擎负责读取数据资源文件, 绘制出相应的图形界面; 并通过事件接收函数进行相应处理, 实现图形界面的显示和刷新控制。

4 图形界面中汉字的支持和应用

Tilcon 的嵌入式引擎默认支持英文字体, 并且提供一个字体引擎支持用户添加新的字体, 要在 Tilcon 应用中使用中文, 首先需要在 WindML 配置中包含 UNICOD 的中文字库。WindML 提供对点阵字库和矢量字库两种字库的支持, 如果想要使用 Windows 系统下现有的 TrueType 汉字库, 需要用户自行开发 GB 库到 UNICOD 库的转换函数, 在使用汉字时, 调用该函数进行数值转换。

完成 WindML 配置后, 在 Tilcon 的 IDS 设计环境菜单中有一个 `FontManager` 选项可以定义字体, 需定义 `FontType (字体类型)` 和 `FontFaceName (字体名称)`, 如 `FontType` 定义为 28, `FontFaceName` 定义为 `KaiTi` (必需是 WindML 中已包含的字体)。定义好后, 目录 `Tilcon` 下的 `fonts.dat` 文件中会增加一项: `0028 KaiTi`; 在应用程序中进行注册后就可以使用该字体的汉字了。

5 结束语

Tilcon 以其独特的嵌入式图形引擎和图形数据文件资源相结合模式, 并采用事件驱动方式, 解决了 `VxWorks`, `Linux` 等嵌入式操作系统下图形界面开发的难题。将软件设计人员从大量、繁琐的

嵌入式系统下图形开发编码工作中解放出来,使其专著于应用程序设计。简化了用户编程,提高了项目开发效率,并将图形界面设计者、美工人员引入产品开发过程,使他们可以和程序员非常方便的协同工作。

本文较为详细的介绍了 Tilcon 的结构、特点,以及它在 VxWorks 操作系统下的应用。基于 Tilcon 的嵌入式系统下图形界面设计的思想值得我们借鉴,希望能给有兴趣的读者带来一些帮助,为相关领域的设计开发提供一种思路。

参考文献

- [1] WindRiver Systems Inc. VxWorks Kernel Programmer's Guide 6.7 [M]. USA, 2010.
- [2] Tilcon Software Ltd. Tilcon Programmer's Guide [M]. CA, 2005.
- [3] WindRiver Systems Inc. WindML SDK Programmer's Guide 3.0 [M]. USA, 2002.
- [4] 周启平, 张扬. VxWorks 下设备驱动程序及 BSP 开发指南 [M]. 北京: 中国电力出版社, 2004.

(上接第 35 页)

(6) 多个点目标模拟。多个点目标模拟就是在天线波束内出现了多个目标, 每个目标的速度和位置各不相同。多个点目标模拟的方法是: 先产生各个目标的回波数据, 然后在距离上将多个目标叠加, 合成一个等效的单点目标输出。在此只讨论两个目标的合成。两个目标的合成分 2 种情况: 有重叠和不重叠。如图 5 和图 6 所示。

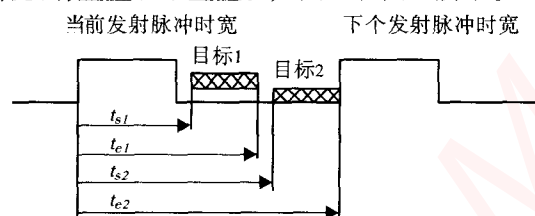


图 6 目标不重叠

图 5、图 6 中 t_{s1}, t_{e1} 为目标 1 的输出起始时间和输出结束时间, t_{s2}, t_{e2} 目标 2 的起始输出时间和终止输出时间。

目标数据合成时, 先由 t_{s1}, t_{e1}, t_{s2} 和 t_{e2} 间的关系确定是有重叠或不重叠。有重叠时, 确定各个目标重叠区的起始位置和结束位置, 并将重叠区内 I, Q 两路数据分别相加, 其他数据不变; 不重叠时, 确定无重叠区内码元的个数并补相同 0 值, 其他数据不变。最后, 调整合成后等效的单目标起始输出时间和终止输出时间。

多目标模拟带来的主要问题是计算量的成倍增加, 影响到模拟器能否实时产生回波数据。短序列码的码短, 数据叠加的计算量小, 但它的 PRT 也小, 在 PRT 时间内不一定能将多目标合成。长序列码 PRT 大, 但其码长, 数据叠加的计算量大, 也不一定在 PRT 内能将多目标合成。所以, 在多目标合成时, 短序列码和长序列码的实时性都有严重影响。要想增加模拟目标的数量, 只能靠增大模拟器的计算能力, 如采用多个 DSP。

5 结束语

为了配合信号处理机的研发, 研制了该雷达信号模拟器。雷达信号模拟器电路结构采用 DSP + FPGA 的模式, 用计算机实现计算生成各种杂波数据文件, 在应用中下载, 能适应不同雷达仿真要求, 满足了雷达信号处理机的调试需求。

参考文献

- [1] 王国玉, 肖顺平, 汪连栋. 电子系统建模仿真与评估 [M]. 长沙: 国防科技大学出版社, 1999.
- [2] 来浩, 蔡兴雨, 付学斌. 雷达信号模拟器的分析与设计 [J]. 火控雷达技术, 2008, (2): 77-80.
- [3] 杨帆, 蚁强, 王永伦. 雷达回波模拟器的设计与实现 [J]. 现代电子技术, 2006, (13): 53-55.

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](#)
68. [机车车辆的 MVB CAN 总线网关设计](#)
69. [智能变电站冗余网络中 IEEE1588 协议的应用](#)

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](#)
45. [基于 Tilcon 的 VxWorks 图形界面开发技术](#)
46. [嵌入式图形系统 Tilcon 及应用研究](#)
47. [基于 VxWorks 的数据采集与重演软件的图形界面的设计与实现](#)
48. [基于嵌入式的 Tilcon 用户图形界面设计与开发](#)
49. [基于 Tilcon 的交互式多页面的设计](#)

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 定制嵌入式 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](#)
29. [基于 PowerPC 的多屏系统设计](#)
30. [基于 PowerPC 的嵌入式 SMP 系统设计](#)

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](#)
36. [面向 OpenCL 架构的 GPGPU 量化性能模型](#)
37. [基于 OpenCL 的图像积分图算法优化研究](#)
38. [基于 OpenCL 的均值平移算法在多个众核平台的性能优化研究](#)
39. [基于 OpenCL 的异构系统并行编程](#)

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](#)

FPGA / CPLD:

1. [一种基于并行处理器的快速车道线检测系统及 FPGA 实现](#)
- 2.