

基于 Tilcon 的交互式多页面的设计

The Design of Conversational MultiPage Graphical Interfaces Based on Tilcon

(中国舰船研究设计中心) 侯小鹏 谭显春 秦克

HOU Xiao-peng TAN Xian-chun QIN Ke

摘要: 在嵌入式应用开发过程中,有时需要友好的用户界面来完成一些人机交互操作。针对交互式多界面开发困难的难题,提出了基于 Tilcons 下进行交互式多页面开发的方法。并对 Tilcon 的结构、特点做了简要的描述,结合作者的实际工作经验,给出了使用 Tilcon 进行交互式多用户图形界面开发的具体实现,并对实现过程中所涉及到的技术细节进行了探讨和详细说明。以实际工作程序中部分重点函数说明了这种方法的可行性。

关键词: 嵌入式操作系统; Tilcon; 人机交互

中图分类号: TP311.11 **文献标识码:** A

Abstract: In the development of embedded systems, we need some user-friendly interfaces to carry out some man-machine conversation operations. In view of the difficulty about graphical interface development, bring forward the method of enpoldering Conversational MultiPage Graphical Interfaces. Introduced the Tilcon's principle of work, Gives and explicates an example of Conversational MultiPage Graphical Interfaces combined with author's experience, accordingly explain this method is feasible.

Key words: Embedded systems; Tilcon; Man-machine conversation operations

1 引言

随着嵌入式系统的发展,很多领域的嵌入式应用(如导航显控系统、监视终端、多媒体终端、监控系统)的人机交互界面也对图形用户界面的需求不断提高。良好的人机交互界面不但要美观易懂而且要操作简单,从而提高用户的使用效率。与一般系统的图形用户界面相比较,嵌入式系统下的图形用户界面要求体积小、占用资源少、高性能、高可靠性、可配置等特点。本文根据嵌入式系统的特点,利用 Tilcon 图形界面开发工具设计的交互式多页面程序具有可靠稳定的运行性能以及简便快捷的设计方式,在工程实际中应用广泛。

2 Tilcon 简介

Tilcon 是一个支持多种操作系统平台的图形用户界面开发工具,用户运用它能够快速、方便地开发出令人满意的图形界面应用程序,用户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之上。

Tilcon 的开发环境主要由三部分组成:Tilcon 嵌入式引擎(EMBEDDED ENGINE)、与平台无关的 API 接口(API)和可视化的界面设计环境(INTERFACE BUILDER)。其体系结构如图所示:Tilcon 的关键部分是嵌入式引擎。嵌入式引擎是一个小巧的、以事件为驱动方式的内核。它与用户的应用程序相独立,负责处理用户所有的事件。API 是 Tilcon 提供的功能函数调用接口,它为用户的应用程序与嵌入式引擎建立了一种通讯机制。它支持动态地创建对象,提供对对象属性所有的控制,同时保持使用上简单、方便。可视化的界面设计环境如下图所示,它本身也是使用 Tilcon 自己的功能函数开发的,它为用户提供了丰富的控件,用户只需要拖动控件、设置控件的显示和运行属性的方式组建自

己的图形界面,用户的应用图形界面完成后,存储为数据资源文件。在目标机环境中,嵌入式引擎负责读取数据资源文件(即在可视化的界面设计环境中生成的*.twd 文件),绘制出相应的图形,同时还可以通过 TCP/IP 与其他目标机的嵌入式引擎相联系。

Tilcon 与其他的图形界面设计工具相比,突出的特点是在使用独特的二进制图形引擎的开发方法,使用它的开发过程和最终产品的系统结构如下所示。用户使用工具(Tool,在 Tilcon 中为 INTERFACE BUILDER)拖拽控件防止到主程序窗口内,并设置各控件的显示和运行属性。用户开发的应用图形界面完成后,存储为数据资源文件(即 RESOURCE FILES)。用户的应用程序(APP SOURCE)与 Tilcon 提供的图形引擎、API 编译链接在一起最终形成产品。数据资源文件的独立使得图形界面的变化只影响数据资源文件的改变,并不需要应用程序重新标以链接,极大提高了开发效率。

3 人机交互的开发

我们所开发的系统目标是使使用者通过控制面板上键盘的操作,达到在屏幕上调用相应的界面的目的。

由于 Tilcon 多种平台的支持,因此可以使用 VC++ 开发,也可以基于嵌入式操作系统如 VxWorks 进行开发,本系统程序在两种平台下均能正确地运行。以 VxWorks 为操作系统,硬件平台为:奔腾 3 中央处理器、4G 电子盘、彩色显示器、标准键盘等。

为了使人机交互操作简洁、方便,采用单独按键控制方式。根据画面上操作命令的提示,按下相应的按键,调用相应的页面,然后可以进行其他的各种命令操作。为了准确地获得被按下按键的键值,传统的方式是采用多线程技术,即产生两个线程,一个是键盘扫描进程,负责不断地读取键盘输入,例如可以循环使用 read(STD_IN, &c, 1) 得到标准输入设备上输入的一个字符。获得键值后用进程间通讯的方式通知另一个进程:键盘消息处理进程,在键盘消息处理程序中实现状态页面的切换,具体的代码设

计上还要考虑到进程间同步等问题。在 Tilcon 下则对键值的捕获更为简单, 我们通过使用 Tilcon 提供的键盘 Notification 来解决人机交互, 即在主消息循环中加入 TRT_keyboard 消息的响应就可以达到, 此消息的捕获是在主消息循环中用 TRT_GetInput() 函数实现, 获得的相对应的数据保存在 Recdata.data 结构里。具体添加消息以及响应消息的方法如下:

在 Tilcon biulder 下面, 设计出几个自己想要的窗口及控件, 然后在每个界面中添加键盘消息。在窗口上单击鼠标右键进入属性对话框, 选择 Action 页面, 在 Transmit Keys 功能面板上鼠标左键单击 Alphabetic characters 按钮, 则在下面对应的列表框 Keylist 中相对应的按键被激活, 这样就可以响应键盘消息了。

将设计好的并生成的界面资源 twd 文件分别装载到电子盘下, 系统启动后会相应地调用这些资源。

4 Tilcon 下多页面动态加载的实现

对于多页面的动态切换, 普遍地有两种方法可以使用, 方法一将所有的窗口资源在程序开始时便装载到内存, 除主窗口显示外, 其他的窗口采用隐藏方式, 根据程序的需要来显示以及隐藏不同的窗口。此方法的优点是响应速度快, 页面上控件的状态也可以一直保持, 缺点就是耗用大量的内存, 方法二采用动态加载的方法, 即在一开始只显示主页面, 其他窗口根据需要再进行加载。

本系统采用方法二, 以键盘的按键“Q”来加载界面资源“myTil1”, 以按键“W”加载界面资源 myTil2”。在两种界面下, 通过使用“Q”“W”按键来切换不同的界面。其主程序流程图如图1所示。键盘处理程序流程图如图2所示。

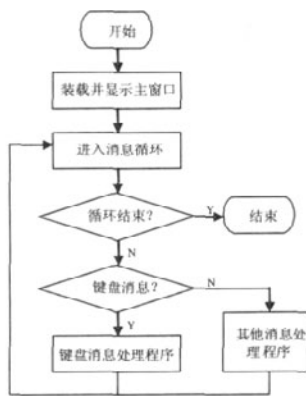


图1 主程序流程图

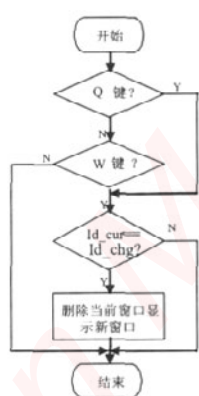


图2 键盘处理程序流程图

程序中涉及到的关键代码如下:

```
int main()
{ Initialapp(); Winfunc( myTil1", "mytil1"); MainEventLoop();
  TRT_WindowDelete(TRT_cid, rec_data.window_ID); TRT_Exit
  (TRT_cid); Return 0; }
```

main()函数是程序的入口函数, 在 main()函数中分别调用系统初始化函数 Initialapp(), 作用是初始化系统变量; 窗口加载与显示子程序 Winfunc(); 主消息循环函数 MainEventLoop()函数, 一旦消息循环结束, 则调用窗口删除程序删掉当前窗口, 退出工作环境。

WinFunc()函数功能为加载并显示窗口。以窗口资源文件名及窗口 ID 为参数。

```
long WinFunc(char *winFilename, char * winId)
{ ...; if(TRT_WindowLoad(TRT_cid, winFilename)) return -1;
  if(TRT_WindowDisplay(TRT_cid, winId)) return -1;
```

```
return 1; }
```

键盘消息的捕获是在 MainEventLoop()函数中, 此函数采用 while()循环以实现窗口消息的连续获取。循环的结束以退出进程为触发。在 MainEventLoop()函数中, 使用 TRT_GetInput()函数获取当前的事件, 并将事件信息存储在 rec_data 数据结构中, 当 TRT_GetInput()返回值为 0 时代表有消息事件发生。当判断出当前消息是键盘消息时, 则调用键盘消息处理函数 ProcesskeyboardMessage()来处理。

```
long MainEventLoop()
{ ...; while (true) { c =TRT_GetInput (NULL, 0, NULL, 0,
  &rec_data, TRT_BLOCK);
  switch(c) { case 0: if(rec_data.code == TRT_keyboard)
    ProcesskeyboardMessage(); ... } }
```

键盘消息处理函数是获取到当前键盘消息中按键的 ASCII 码值, 根据这个值来选择删除与加载相应的窗口, 其中 81 为“q”的 ASCII 码值, 113 则为“Q”; 同理 87 与 119 代表“w”与“W”。

```
void ProcesskeyboardMessage()
{ switch (rec_data.state) case 81: case 113: ReloadWin
  ( MyTil1", "Mytil1"); Break;
  Case 87: Case 119: reloadWin( MyTil2", "Mytil2"); break; ...
  default: break; }
```

ReloadWin()函数根据当前保存在 rec_data 结构中的当前窗口 ID 来与即将切换的窗口的 ID 进行比较, 若 ID 是一致, 则不切换, 只有 ID 不一致时才进行删除当前窗口, 加载并显示新窗口。

```
long ReloadWin(char *winFilename, char * winId)
{ ...; if (strcmp (rec_data.window_ID, winId) != 0) TRT_win-
  dowDelete (TRT_cid, rec_data.window_ID); WinFunc (winFile,
  winId); ... return 1; }
```

5 结束语

本文作者创新点: 使用所见所得的 Tilcon 进行界面设计, 摆脱 WindML 下复杂的函数的调用。创建的界面美观, 可操作性强, 使用窗口重新加载方法节约了系统内存, 降低了系统任务级的开销, 使得在嵌入式这种高集成系统中系统更能发挥高的性能。在 VxWorks 系统中的实际使用表明, 使用 Tilcon 进行交互式多页面的方法是可行的。

参考文献

- [1]Tilcon SoftWare White Paper [Z]. Canada: Tilcon Software Ltd.
- [2]Tilcon User Guide [Z]. Canada: Tilcon Software Ltd.
- [3]Tilcon Evaluation Guide [Z]. Tilcon Software Ltd.
- [4]廖容, 马中, 肖成俊. Tilcon 在 Vxworks 操作系统中的应用. 舰船电子工程. 2007, 27(2)
- [5]曲宏松, 张叶, 曹立华, 耿爱辉, 陈淘. Vxworks 下多页图形界面的设计. 微计算机信息. 2006, 10-2, 192

作者简介: 侯小鹏(1977-), 男(汉族), 硕士研究生毕业于西北工业大学, 工作于中国舰船研究设计中心, 主要从事软件系统开发。

Biography: HOU Xiao-peng (1977 -), male (Han nationality), China Ship Development and Design Center, Research area: major in the software research.

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](#)

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](#)

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](#)
29. [基于 PowerPC 的多屏系统设计](#)

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](#)

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.