

军事指挥系统中 VxWorks 下汉字显示技术

万海东,刘 丹,赵 伟,夏毅军

(南京电子技术研究所 江苏 南京 210013)

摘 要: VxWorks 作为一个高性能、可裁剪的嵌入式实时操作系统,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、航天等技术领域。为实现现代军事指挥系统中的中文显示界面,详细分析了 VxWorks 中 WindML 的显示和字体驱动,并根据现有的西文字库设计出点阵汉字字库文件,随后设计出 VxWorks 下与汉字字库对应的接口。并最终将以上方法具体应用于 VxWorks 程序设计中,使军事指挥系统中战场信息获得了中文显示,最后给出相应源代码作为示例。

关键词: 军事指挥系统;嵌入式操作系统;VxWorks;WindML;汉字点阵

中图分类号: TP311

文献标识码: A

文章编号: 1004-373X(2009)13-047-03

Display Technology of Chinese Characters Based on VxWorks in Military Command System

WAN Haidong, LIU Dan, ZHAO Wei, XIA Yijun

(Nanjing Research Institute of Electronics Technology, Nanjing, 210013, China)

Abstract: VxWorks as a kind of embedded real-time operating system with high performance and the function of being cut out, are widely applied in areas of communication, military, aviation, astronautics by its reliability and remarkable real-time. For the purpose of constructing a Chinese character display interface in modern military command system, the display of WindML and font drive in VxWorks are provided. Chinese character lattice font document is designed based on the existing west character font in the paper. And the interface with the Chinese character font is designed in VxWorks. All above the method are applied in the VxWorks programming, so the battlefield information can be displayed in Chinese in military command system. The corresponding source code is given to being demonstration.

Keywords: military command system; embedded operating system; VxWorks; WindML; Chinese character lattice

由于对实时性和可靠性的要求,现代军事指挥系统越来越广泛地采用嵌入式实时操作系统 VxWorks。WindRiver 公司的媒体库 WindML (Wind Media Library) 提供了基于 VxWorks 的图形解决方案,其 UGL 组件提供了显示模式设置、标准输入/输出控制和点、线、面作图等函数,但没有直接提供完备的汉字显示解决方案,不支持汉字显示。针对实际指挥系统中人机交互界面的需求,全西文的交互方式显然不便于文件的传输和信息的交换。本文通过对汉字编码和 WindML 显示机制的分析,给出了 VxWorks 嵌入式平台下汉字显示的实施方案。

1 VxWorks 在军事指挥系统中的应用

所谓嵌入式系统,是指内置到火控、战车、飞机、导弹以及卫星和舰船导航等各类武器系统和作战平台上的计算机系统。随着信息技术的迅猛发展,尤其是计算机技术和微电子技术的快速发展,嵌入式系统在军事上得到了广泛应用。嵌入式计算机系统,最早出现在 20 世纪 60 年代武器控制中,后来用于军事指挥控制和通信系统,所以军事国防历来就是嵌入式系统的一个重

要应用领域。现在,各种武器控制(火炮控制、导弹控制、智能炸弹制导引爆装置)、坦克、舰艇、轰炸机等,以及陆海空各种军用电子装备、现代化的军事指挥系统等都可以看到嵌入式系统的影子。

在实时嵌入式操作系统中,VxWorks 以其高可靠性和高实时性得到了广泛的应用。VxWorks 是美国 Wind River System 公司(风河公司,简称 WRS 公司)推出的一个实时操作系统。VxWorks 是一个运行在目标机上的高性能、可裁剪的嵌入式实时操作系统,具有专门为实时嵌入式系统设计开发的操作系统内核,它包括了进程管理、存储管理、设备管理、文件系统管理、网络协议及系统应用等几部分。

VxWorks 在军事指挥系统中应用时,由于指挥系统对人机交互能力的要求较高,需要对武器参数、操作进程以及作战指令等战场信息进行显示,因此迫切需要完成 VxWorks 下的中文显示开发。军事指挥系统中所显示战场信息如图 1 所示。

2 VxWorks 下汉字显示方案

2.1 WindML 基本功能

WindML 是由 WindRiver 公司提供的媒体库,其

U GL 组件提供了显示模式设置、标准输入/输出控制和点、线、面作图等函数,以及对芯片、输入/输出设备、音频/视频、帧缓冲器等开放的应用程序接口。

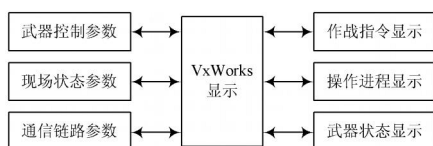


图1 军事指挥系统中战场信息显示

WindML 的主要功能有二维图形 API、事件服务、区域和视窗管理、多媒体 API 和资源管理。其中,二维图形 API 是最常用的部分,包括基本绘图操作(如画线、矩形、椭圆、多边形、点等),选择字体输出文本、位图管理、光标管理、批量绘图操作、图形上下文、色彩管理和双缓冲等。事件服务程序用来处理输入设备的输入请求。它会把键盘、鼠标等输入数据转化为事件并加入应用队列;区域和视窗管理可以在界面上定义一个区域或多线程之间共享的窗口,以进行绘图操作;多媒体 API 支持 NTSC, PAL, SECAM 等视频制式下 DSP 或混频器两种设备的音频输出,也支持 JPEG 图形格式;资源管理器是指常规的 WindML 资源(例如设备和事件队列)、内存管理和驱动器注册等资源的建立、控制和删除。

2.2 WindML 字体驱动体系结构

WindML 字体驱动在 WindML 2D API 上提供了一个通用的抽象层,允许应用程序通过一个通用 2D 字体 API 库使用不同的字体引擎。通用 2D 字体 API 是提供给应用程序使用字体 API 的公共接口。字体驱动不同于 VxWorks 中的其他驱动,它不允许执行或使用任何图形驱动接口以下的程序,也就是说它不直接访问硬件,只访问系统所使用的图形驱动或间接通过 U GL 图形驱动结构来访问硬件。这就允许字体驱动可以运行在任何图形驱动之上,与字体引擎硬件无关。WindML 字体驱动体系结构如图 2 所示。

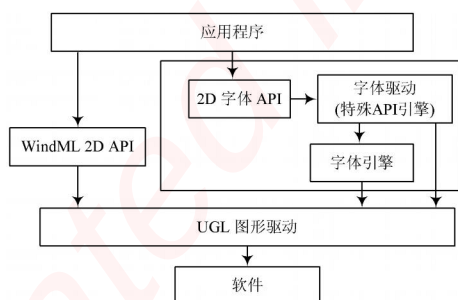


图2 WindML 字体驱动体系结构

2.3 点阵汉字输出过程

点阵汉字的显示是利用 WindML 的图形输出功能来实现的。设计开发中利用 WindML 对位图字符显示和双字节编码的支持,按照 WindML 位图字体文件的格式

完成汉字点阵的转换存储,并使用 WindML 的双字节显示函数实现点阵获取、点阵显示,将汉字显示出来。

WindML 有很多字库文件,原始安装状态下都是西文字库文件,要在 VxWorks 系统中显示中文汉字,解决方案是将西文字库文件修改成为中文字库文件,即设计一种类似西文字库格式的 C 语言文件,文件中存储双字节的汉字编码、点阵信息及字体的控制信息等。字库文件经过系统配置、编译后生成点阵字库文件目标代码,点阵驱动程序从目标代码中获取点阵信息传递给应用程序编程接口,应用程序调用图形显示函数输出汉字。WindML 下汉字显示处理过程如图 3 所示。

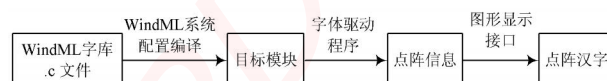


图3 汉字显示处理过程

2.4 点阵汉字存储结构

国家标准对汉字库结构做了统一规定:字库分为 94 个区,每个区有 94 个汉字,每个汉字有确定的区和位编码,用两个字节表示,第一个字节表示汉字所在的区号,第二个字节表示位号,即所谓的区位码。每个汉字在字库中以点阵字模形式存储,一般采用 16 × 16 点阵、24 × 24 点阵,每个点用二进制位(0 或 1)表示,对应应在屏幕上显示出来就是相应的汉字。每个汉字都有惟一的区位码,通过区位码计算出汉字在字库中的位置,从而确定汉字的字模。

点阵汉字字库 C 语言文件有固定的结构,分为三个部分:点阵属性表、点阵信息描述总表和点阵信息表。

点阵属性表 用来描述点阵汉字的字符结构和字模结构参数的结构体,包含参数有像素大小、字符名称、字符家族名称等,特别定义了指向点阵信息总表的数组指针,字体驱动程序利用这个指针寻找点阵信息总表。

点阵信息描述总表 它是一个存放指向点阵信息表指针的动态数组,其元素个数取决于汉字点阵信息表的个数。

点阵信息表 点阵信息表以数组的形式表示,主要用来存储汉字的 GB2312 码(西文是 ASCII 码)和详细点阵信息。可以是一个表也可以是若干个表。把汉字按 GB2312 编码的页分类,构建多个汉字分表,也可以把中西文装在混合表里做成一个大表,西文字符被扩展成双字节存储。

3 点阵汉字字库实例

自己开发点阵存储、点阵获取、点阵显示,改造 WindML 的双字节或单字节显示函数,使其能判断汉字码,一旦判断出汉字码,则使用自己开发的点阵获取、

点阵显示等把汉字显示出来。该方法对点阵的操作更加灵活,适合非标准的点阵算法,或者当开发者已有成熟的点阵操作方法时,把该方法绑接到 WindML 上。

不必理会 WindML 双字节编码机制,这里需要的是一个汉字的点阵字库,这个点阵字库不必按照 WindML 的双字节编码机制进行编码,里面所存储的只是汉字的点阵信息和区位码信息,这个字库可以通过从 Windows 中导出;由于不选用 WindML 的双字节编码机制,所以对于 VxWorks 中关于如何控制打点以及显示点阵字体驱动,并不需要具体了解;更重要的是拥有更大的灵活性,只需把自己所写的汉字显示函数绑定到 WindML 上面即可。因此这里必须要求自己写点阵的存储、点阵显示、汉字显示函数。做起来比较简单,只要求了解汉字编码的基本知识,即汉字双字节编码中,高、低字节的最高位都是为‘1’,而西文字符编码中最高位为‘0’,利用这点来判断所要显示的字符是汉字还是英文。

在设计开发过程中,在现有的西文字库的基础上进行修改,添加相应的字体参数和汉字的点阵信息,形成点阵汉字字库文件 uflhz12.c, uflhz16.c, uflhz24.c, 以分别存储 12, 16 和 24 号宋体的字库信息。这个过程用 Windows 编程来实现。

以下是 16 号宋体字库文件中的点阵信息表:

```
U_GL LOCAL const unsigned char page0Data[] =
{
    /* ASCII[032] */
    0x00,          /* * page */
    0x20,          /* * index */
    0x00,          /* * size(MSB) */
    20,           /* * size(LSB) */
    8,            /* * width */
    16,           /* * height */
    16,           /* * ascent */
    0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,
    0x00,0x00,0x00,0x00,0x00,0x00,
    ...
    /* End of Page */
    0,0,0,0
};
```

点阵信息表后定义了点阵信息描述总表:

```
U_GL LOCAL const unsigned char *const pageArray[] =
{
    page0Data,
    page1Data,
    page2Data,
    ...
    page93Data,
    page94Data,
    U_GL NULL
};
```

作者简介 万海东 男,1978 年出生,江苏常州人,工程师,工程硕士。研究方向为系统工程。
刘 丹 女,1979 年出生,湖北武汉人,工程师,硕士。研究方向为系统工程。
赵 伟 男,1982 年出生,江苏盐城人,助理工程师,硕士。研究方向为系统工程。

该指针指向点阵信息表的首地址,并被点阵描述结构直接调用。

字库文件的最后定义了点阵属性表,该表规定了整个点阵字库的属性以及字模和间距,并定义点阵指针。

```
const U_GL BMF_FONT_DESC uglBMFont_Song_16 =
{
    { 16,16}, /* pixelSize */
    {U_GL_FONT_BOLD OFF,U_GL_FONT_BOLD OFF}, /* weight */
    U_GL_FONT_UPRIGHT,
    U_GL_FONT_PROPORTIONAL,
    U_GL_FONT_ISO_8859_1, /* charset */
    'Song 16x16 Dot', /* face name */
    'Song' /* family name */
},
0,
16,
0,
16,
pageArray
};
```

4 结 语

在 VxWorks 嵌入式平台下采用点阵汉字显示技术,使得战场信息、态势、指令等获得了中文显示。最终使设计开发的军事指挥系统不仅具有良好的可靠性和卓越的实时性,同时具有直观而方便的中文平台以及友好的交互能力,大大提升了整体作战效能。

参 考 文 献

- [1] WindRiver System, Inc. . VxWorks Programmer's Guide[Z].
- [2] WindRiver System, Inc. . WindML Programmer's Guide[Z].
- [3] 刘尚军,张志兵. ARM 嵌入式技术原理与应用[M]. 北京:北京航空航天大学出版社,2007.
- [4] 张海堂. Visual C++ 6.0 编程指南[M]. 北京:航空工业出版社,1999.
- [5] 王学龙. 嵌入式 VxWorks 系统开发与应用[M]. 北京:人民邮电出版社,2003.
- [6] 梁勇,孟桥. 嵌入式操作系统 VxWorks 中的多媒体组件 WindML 2.0[J]. 现代计算机,2002(10):88-91.
- [7] 李德元. 嵌入式计算机技术及其应用[J]. 计算机工程,1998,24(10):7-11.
- [8] 孔详营,柏桂枝. 嵌入式实时操作系统 VxWorks 及其开发环境 Tornado[M]. 北京:中国电力出版社,2002.
- [9] 陈智育,温彦军,陈琪. VxWorks 程序开发实践[M]. 北京:人民邮电出版社,2004.
- [10] 谭浩强. C 程序设计[M]. 北京:清华大学出版社,2001.

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](#)

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](#)

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](#)

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](#)

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](#)

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](#)

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](#)

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](#)