

## 基于 VxWorks 嵌入式系统的中文平台研究与实现

杨远宾, 刘 炯

(中国航天科工集团第二研究院 706 所, 北京 100854)

**摘要:**介绍了 VxWorks 嵌入式操作系统的中文平台的原理和组件式实现方法,概述了点阵字体,矢量字体的基本知识、设计方法和输出过程,给出了中文输入法的解决方案和中文平台在 VxWorks 系统中的位置关系,使得在 VxWorks 操作系统上可以自由的使用中文。

**关键词:** VxWorks; 点阵字体; 矢量字体; TrueType; FreeType; 中文输入法

**中图分类号:** 文献标识码: A 文章编号: 1000-7024 (2006) 20-3890-04

### Design and implementation of Chinese platform based on VxWorks embedded system

YANG Yuan-bin, LIU Jiong

(Institute 706, Second Academy of China Aerospace Science and Industry Corporation, Beijing 100854, China)

**Abstract:** The theory and implementation of Chinese platform based on component, the basic knowledge, design and output method of the raster and vector font are introduced, and the solution of Chinese inputting, position of Chinese platform in VxWorks embedded system are pointed out, which make it possibly to use Chinese in VxWorks OS freely.

**Key words:** VxWorks; raster font; vector font; TrueType; FreeType; Chinese inputting

### 0 引言

VxWorks 操作系统是美国 WindRiver 公司于 1983 年设计开发的一种嵌入式实时操作系统(RTOS),是嵌入式开发平台的关键组成部分。它以其良好的可靠性和卓越的实时性被广泛地应用在通信、军事、航空、航天等高精尖技术及实时性要求极高的领域中。

在国内, VxWorks 的应用群体不断扩大,应用范围不断拓展,用户的需求也在不断提高,程序设计已经不再局限于无界面,无交互的方式。人们越来越需要 VxWorks 系统开发的应用程序能具有友好的人机交互能力,尤其是具有直观而灵活的中文平台,体现 Windows 平台的风格。然而, VxWorks 系统只支持西文,若开发的应用程序人机界面较多,这种全西文的交互方式,对广大用户来说,显然不便于文件的传输和信息的交换。只要在 VxWorks 操作系统中加入中文平台组件,就可以实现点阵汉字(也可称作位图汉字或者光栅汉字,以下简称点阵汉字)、矢量汉字的输出以及汉字的输入,可以解决 VxWorks 只支持西文的局限性,更好的提高人机交互的友好度,对 VxWorks 系统应用的推广也会起着重要的作用。

本文专门就 VxWorks 系统如何支持中文输入输出的问题进行了讨论,分 3 大模块,即点阵汉字模块,矢量汉字模块和输入法模块,各模块既有联系又有较强的独立性,文章前部分采用各个模块独立阐述的方式,先介绍 VxWorks 系统点阵汉

字和矢量汉字,在此基础上,进一步阐述了中文输入法的解决方案,最后说明了各个模块在 VxWorks 系统中的位置关系。

### 1 点阵汉字的实现原理

点阵汉字处理需要有图形库的支持, VxWorks 系统一般用 WindML 组件来做图形支持, WindML 是 Wind River Multimedia Library 的简称,其中包含 UGL(universal graphics library)。该组件提供了向上的抽象接口和向下的与具体硬件相关的设备驱动。主要包括输出模式设置、标准输入输出控制、点线面作图等函数。 VxWorks 系统的点阵汉字实现是利用 WindML 的图形输出功能来完成的,因此,这里介绍的点阵汉字原理也可以理解为 WindML 的点阵汉字实现原理。本节先说明了 WindML 输出汉字的流程,然后分述点阵汉字的存储方法和存储结构。

#### 1.1 WindML 点阵汉字的输出过程

按 VxWorks 西文处理的方法,设计一种类似西文字库格式的 C 语言文件,在这种文件里可以存储用双字节的汉字编码、汉字的点阵信息及字体的控制信息等,这种文件就叫做 WindML 的字库文件。 WindML 有很多字库文件,安装 WindML 后,就可以在 WIND\_BASE/target/src/ugl/fonts/bmf 目录下找到,原始状态下都是西文字库文件。因此,要在 VxWorks 系统中输出汉字就得把西文字库文件修改成为中文字库文件。这些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交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，其过程如图 1 所示。

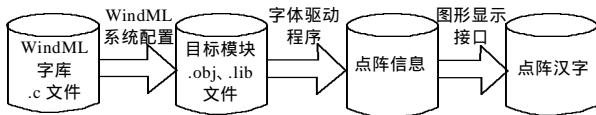


图 1 点阵汉字显示过程

## 1.2 点阵字库的存储结构

前面提到，WindML 的点阵字库是一些 C 语言文件。这节课里将对它们的结构做概要介绍。这些文件内部都是固定的结构，可分 3 部分来描述：分别是点阵描述结构，点阵信息描述表总表和点阵信息结构。其中有两个重要的概念，一个是字模结构，WindML 为每一种字体定义了一个字模结构，它规定了每个字符所拥有的空间大小，相邻两个字符的间距；一个是字符结构，定义了每个字符的实际大小和它的字体属性，它是以字模为标准的，不得超过字模的大小，否则，超出部分将会被裁剪掉。

研究 WindML 字体显示驱动程序的执行过程，可以更好的理解点阵字库的存储结构。字体驱动程序在字库中寻找汉字的过程是这样的，驱动程序根据字体名称，如宋体、楷体等先找到一个点阵描述结构，然后找到一个点阵信息描述表的总表，最后找到具体的点阵信息。按照这个执行顺序来说明字库文件的 3 部分结构。

第 1 部分：点阵描述结构，用来描述点阵汉字的字符结构和字模结构参数的结构体，其包含的参数有像素大小、粗细度、倾斜度、间隔大小、字符集、字符名字、字符家族名字等，特别定义了一个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到点阵信息总表。

第 2 部分：点阵信息表总表，是一个存放指向点阵信息表的指针的动态数组，其元素个数取决于汉字的点阵信息表的个数。

第 3 部分：点阵信息表，它以数组的形式表示，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一个混合表里做成一个大表，西文字符被扩展成双字节存储。点阵汉字字库的详细结构在文章 ([http://www.2lic.com/new\\_info/news?num=VxWorks](http://www.2lic.com/new_info/news?num=VxWorks))中也有描述，这里不再累赘。

VxWorks 系统点阵汉字实现技术目前已经比较成熟，其原理是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，这个过程可以用 Windows 编程来实现。

## 2 TrueType 字库输出原理

### 2.1 矢量字体基础

与点阵字体截然不同的是矢量字体中每一个字形是通过数学曲线来描述的，它包含了字形边界上的关键点，连线的导数信息等，字体的渲染发动机通过读取这些数学矢量，然后进行一定的数学运算来进行渲染。这类字体的好处是字体可以无限放大、缩小而不产生失真。VxWorks 系统的矢量汉字主要用在 2D 图形显示系统、3D 图形显示系统，打印输出，GIS 软件系统等方面，即可以提高输出速度，又可以得到高质量的汉

字输出，真可谓所见即所得。

矢量字体有许多种类，这里只研究其中的一种——由 TrueType 字库产生的字体，这里简称成 TrueType 字体，也是 Windows 常用的字体，它是一种基于轮廓技术的字体，字体信息是用直线段，二次贝塞尔曲线来描述的，这使得它们比其它矢量的字体更容易处理，保证了屏幕与打印输出的一致性；同时，可以随意缩放、旋转而不必担心会出现锯齿，这也是矢量字体相对于点阵字体无可比拟的优越性所在。

### 2.2 实现方法研究

TrueType 字体信息存储在 TrueType 字体文件中，这些文件就是 TrueType 字库，VxWorks 下实现 TrueType 字体的汉字输出目前有两种方法：

第 1 种方法是从最底层开发，需要弄清楚 TrueType 字库文件的结构，从字库文件中取出汉字的轮廓信息，利用系统绘图函数进行绘制。TrueType 字库文件格式比较复杂(具体可以参看相关书籍，这里只做核心分析)，它的文件结构是由树形的表组织起来的，除了字库文件的首部信息外还有许多用来记录字体信息的描述表，其中有 3 个重要的描述表，分别是 Cmap 表、Local 表、Glyph 表。

Cmap 表：系统内码(如 gb2312 码、unicode 码等)到字模索引号的映射表；

Local 表：描述每个字的轮廓信息在“Glyph”表中的偏移；

Glyph 表：存放轮廓信息，包括数据信息和指令信息。

TrueType 字库文件的存储结构采用两级映射机制，如图 2 所示。

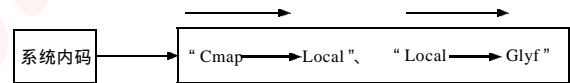


图 2 TrueType 字库两级映射机制

从 TrueType 字库文件中查询字体信息的具体过程，如图 3 所示。

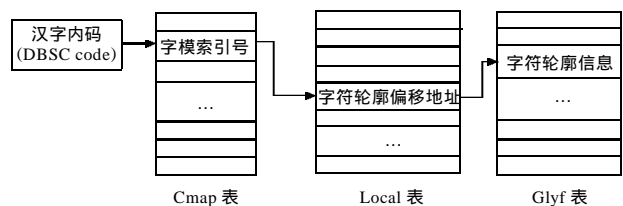


图 3 TrueType 字库的字符轮廓信息查询

第 2 种方法是利用 FreeType 这个开放源代码的字体引擎，不用关心 TrueType 字库文件格式，使用 FreeType 的应用编程接口(API)就可以完成大部分工作。而且 FreeType 软件可以编译运行在 Windows、Linux、VxWorks、WinCE 等多种平台下。因此，采用这种方法来解决矢量汉字输出问题比较理想。

### 2.3 FreeType 的使用

FreeType 提供了许多 API 函数，在其文档中做了详细的说明，关于 FreeType 的详细介绍可以从网站(<http://www.freetype.org/index.htm>)上直接下载。这里给出用 FreeType 的 API 绘制字符的流程，如图 4 所示，实际编程中采用这种流程实现是可行的。

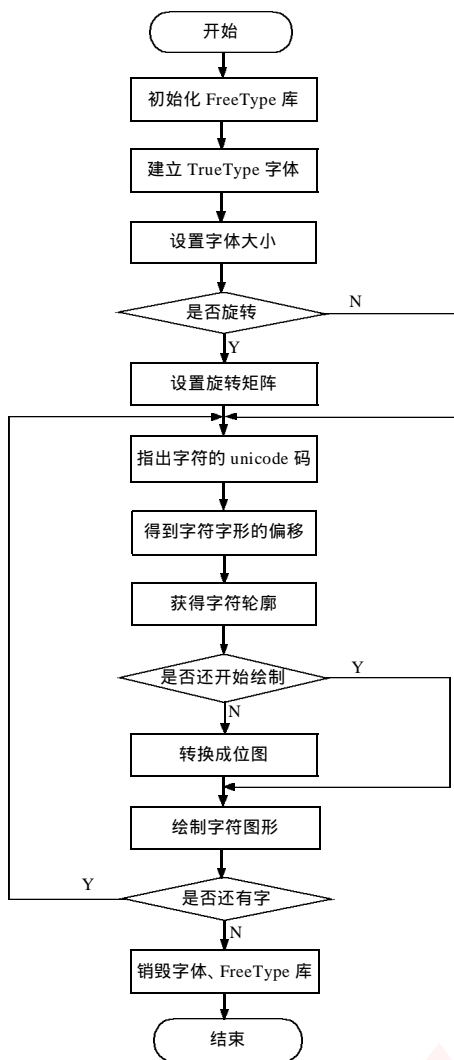


图 4 矢量汉字或字符的输出流程

每一个输出的字符或汉字都须经过这个基本过程,当然,实际应用中会有许多细节的问题需要解决,比这里讲得要复杂。事实证明,采用FreeType技术输出矢量汉字,开发周期短,输出速度快,字体质量好。

### 3 中文输入法实现简介

#### 3.1 输入法原理研究

在 VxWorks 系统中有了点阵汉字和矢量汉字显示的功能,显然为中文输入法的实现奠定了基础。这里首先对中文输入法的原理做一个简单的介绍。

纵观国内,中文输入法种类繁多,各有千秋,技术较为成熟。随着电脑的普及和时间的推移,一种种输入法不断出现,据说中文输入法申请专利的就有 500 多种。现在,经常还会有各种输入法诞生。

输入法种类虽多,但万变不离其宗,其原理都是利用信息处理技术,将一个“字”编进几个英文或数字代码(一般是 4 个),这样敲打这几个代码时,就可以打出该字来。输入法的不同体现在编码方式上,每一种输入法都有其自己的编码特点。简单的说,设计输入法实际上是在根据汉字的某些规律设计一种编码方式,即字母(或者数字)到汉字的一种映射方式。例

如:全拼就是用几个字母去对应若干个汉字,双拼就是用两个字母去对应若干个汉字,显然,词组输入就是用一串字母去对应一串汉字。笔画输入法,智能输入法等都是这个原理,就连当今最先进的声音识别输入法,手写输入法也类似,只不过声音识别输入法是音频编码到汉字的对应,手写输入法是一种图像到汉字的对应关系。不同的是,各种输入法,根据它们的智能化水平的高低,其实现难度就不一样了。不过,一般的输入法其工作过程是这样的,首先获取若干个字母组合,根据这种预先定义的映射关系从某个存储汉字的文件里去查找对应的汉字,然后输出给后台。需要说明的是,这个存储汉字的文件也可叫汉字码字库。

这里说的汉字码字库不同于点阵汉字和矢量汉字的字库,两者有不同的存储方法、不同的结构和不同的用途,但都包含了常用的汉字编码,是中文输入输出的根源。

制作中文输入法的关键技术在于汉字字库的存储方法,也就是如何合理的组织存储几千个甚至几万个汉字以方便查找。目前,汉字的存储方法有许多,常见的可分为两大类:

第 1 类是数字存储,用几个阿拉伯数字代替一个汉字,把这些数字按照一定的次序排列存储,根据每组阿拉伯数字的排列位置可以计算出该汉字的国标码,利用系统机制输出汉字。这种方法存储空间小,查询速度快,对程序员要求高,最适合嵌入式、移动设备、PDA 等。

第 2 类是字母与汉字混合存储,这种方式类似数据库里的表的形式,表的第 1 个域存字母组合,第 2 个域存汉字串。这类方法存储的汉字字库一目了然,简单易用,编程很方便,但需要较大的存储空间。一般适合 Windows 系统,现在某些嵌入式系统也常采用这种存储方法。

当然,对于大多数智能输入法来说,其存储方法不会这么单一,也不会这么简单。但基本原理是这样的,在实际应用中往往要动态生成临时的字库。

#### 3.2 VxWorks 中文输入法实现过程

在 VxWorks 的某些应用系统中,有时需要利用中文来进行通信和交流,这时设计一个中文输入法显得尤为重要。中文输入法的实现需要图形的支持和汉字输出的机制,点阵汉字和矢量汉字的显示为中文输入法提供了输出基础。图形显示机制仍然采用 WindML 里的 UGL 图形库。如要求更高级的图形界面,可以使用 Zinc 设计器来设计,Zinc 图形库是对 WindML 图形库的封装,采用面向对象的设计思想,提供类似 Windows 界面的图形控件。

输入法的实现可分 3 步完成:定义一种或多种字母(数字)到汉字的映射机制;根据这种机制建立汉字字库;编程实现。为了简明扼要的说明中文输入法的实现方法,并具实用性,这里给出了详细的程序设计流程图,如图 5 所示。

最后,衡量一个输入法好坏的标准是看其是否方便、易学、快速、普及,字库是否容易扩展,而它们之间又是相互约束的,因此,要设计出一个优秀的通用的输入法仍然是人们所追求和期望的。

### 4 应用层次关系

VxWorks 系统中中文平台的主体位于 VxWorks 操作系统的

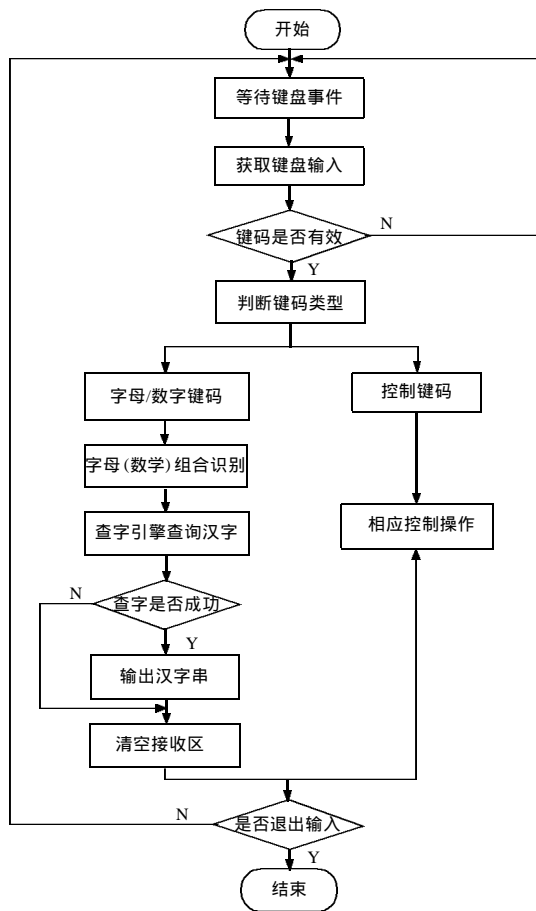


图5 中文输入法实现流程

硬件依赖层和应用层之间,与2D图形库并列存在,如图6所示。图中同时给出VxWorks系统的图形显示机制,包括输入输出的硬件层,硬件依赖层和硬件抽象层。中文平台在实际应用中是以组件的方式提供给用户,组件式开发是当今嵌入式系统中很流行,很有价值的开发方式,在这里开发的中文平台组件,不仅符合VxWorks系统高度可剪裁的特性,而且这些组件只需要稍加修改就可以移植到大多数操作系统平台上运行。组件式开发的基本知识可以参考相关书籍,这里不做详细介绍。

## 5 结束语

在VxWorks系统中进行复杂的可视化程序设计时,若需

(上接第3889页)

的硬件结构以及操作数寻址方式等信息自动构造不同的汇编程序的设计和实现方法。测试证明该方法,对不同的输入信息,能够正确汇编出不同的目标代码。该方法使汇编语言的开发过程变得简单、有效,可用于实际的汇编程序构造。

## 参考文献:

[1] Barry B Brey. The intel microprocessors architecture, programming and interfacing[M].BJ:Higher Education Press, 2001.  
 [2] Barry B Brey. Intel 微处理器——从 8086 到 Pentium 系列体系结构、编程与接口技术(影印版)[M].北京:高等教育出版社, 2001.

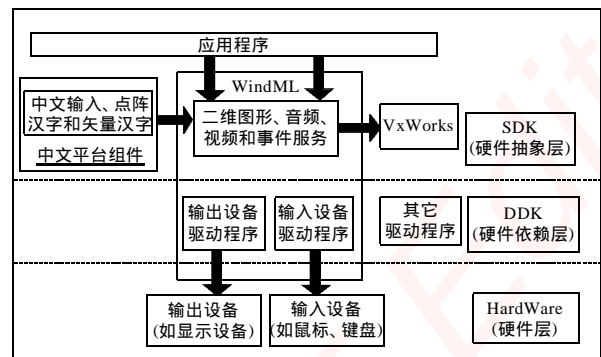


图6 VxWorks系统图形显示机制及中文平台的位置

要输出汉字,系统中一般采用点阵汉字与矢量汉字并存的方式。菜单、标题栏、对话框等汉字输出采用点阵汉字技术;在GIS系统中或打印输出等方面采用矢量汉字技术;在进行实时的中文信息传输时,需要使用中文输入法。目前,这些技术已经在某型号指控软件系统中成功实现。

## 参考文献:

[1] Wind River Systems Inc. VxWorks Programmer's Guide 5.5 edition 1[Z].2000.  
 [2] Wind River Systems Inc. WindML Programmer's Guide 2.0 edition[Z].2000.  
 [3] 孔祥营,柏桂枝.嵌入式实时操作系统VxWorks及其开发平台Tornado[M].北京:中国电力出版社, 2001.  
 [4] 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.  
 [5] Rick Grehan, Robert Moote. Real-time programming [M]. Arrangement with Addison Wesley Longman Inc, 2001.  
 [6] 袁渊. 基于嵌入式VxWorks系统的图形界面开发 [EB/OL]. <http://www.eepw.com.cn/show.aspx?id=81&id=34>.  
 [7] TrueType specifications[EB/OL]. <http://www.microsoft.com/typograph/specs/default.htm>.  
 [8] 李宏,潘玉奇.手写体汉字识别中小波分形特征的研究[J].计算机工程与设计, 2005,26(5):1217-1219.  
 [9] 周克兰,张玉华.汉字数码输入法功能的分析与研究[EB/OL]. <http://www.yywzw.com/jt/srh/hlg01-80.htm>.

[3] 仇玉章,孙力娟,洪龙,等. 32位微型计算机原理与接口技术[M].北京:清华大学出版社, 2000.  
 [4] 胡元义,邓亚,玲胡英. 编译原理实践教程[M].西安:西安电子科技大学出版社, 2002.  
 [5] 白中英,邝坚. 计算机组成原理[M].第3版.北京:科学出版社, 2000.  
 [6] 潘雪增,平玲娣. 计算机组成与设计[M].杭州:浙江大学出版社, 2004.  
 [7] 温敬和. 语法制导翻译在汇编程序自动构造中的应用[J].计算机工程, 2005,31(12): 75-77.  
 [8] Loudon K C. 编译原理及实践[M].北京:机械工业出版社, 2000.

# 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](#)

## 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](#)

## 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](#)

## 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](#)

## 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](#)

## 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](#)

## 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](#)