

基于 VxWorks 的 BSP 技术分析

The Analysis of BSP Based on VxWorks

西安电子科技大学 隋霞 许录平
Sui, Xia

摘要 结合嵌入式操作系统设计的主要特点 针对 4 目标机上 VxWorks 的定制运用基于 BSP 技术解决了移植过程中底层硬件与上层软件无关性的问题。描述了 BSP 的概念和启动流程 介绍了 BSP 设计过程 并结合实际工程给出了 4 目标机下 VxWorks 的定制及应用程序加载方法。

关键词 嵌入式操作系统 ; VxWorks 板级支持包 映像 ronald ;
中文图分类号 : TP316.9 文献标识码 : A

Abstract: Combining with the features of the embedded operating system design, this paper analyzes the BSP technology based on VxWorks, which solves the problem of the independence of the hardware and software in the process of transplantation. It describes the concept and start-up flow of the BSP, introduces the design process of the BSP, and gives the customization and application program loading method of VxWorks on the 4 target machine.

Keywords: embedded operating system, VxWorks board support package, image, ronald

VxWorks 是由美国 Motorola 公司推出的一种强实时嵌入式实时操作系统 (RT) 它采用微内核结构 具有高可靠性、实时性、可裁减性等特点 而且支持多种 CPU。VxWorks 的一个重要特点是它的可移植性 应用程序编码在很大程度上与目标板的硬件和结构无关。它把所有特定的硬件功能都集成在一个称之为板级支持包 (BSP) 的库中。目前为止 Win公司提供的基于某一 C 评估板的 BSP 超过 20 个 当用户开发自己目标板的 BSP 时 比较简单的方法是从所提供的 BSP 中最接近的来修改。这里提到的 BSP 的设计就是指这种板级的对依赖于硬件的源代码做必要的修改、增加或删除 以满足特定的硬件环境。

1 的概念

BSP (Board Support Package) 是介于底层硬件和上层软件之间的底层软件开发包 这源于嵌入式操作系统与硬件无关的设计思想。如今的各种嵌入式系统 都采用了分层的设计方法。它将系统中与硬件直接相关的一层软件独立出来 称之为板级支持包 (BSP) 的主要功能为屏蔽硬件 提供操作系统的驱动及硬件驱动 具体功能包括 :

(目标板硬件初始化 主要是 C 的初始化 为整个系统提供底层硬件支持 ;

(为操作系统提供设备驱动程序和系统中断服务程序 ;

(定制操作系统的功能 为软件系统提供一个实时多任务的运行环境 ;

(初始化操作系统 为操作系统的正常运行做好准备 ;

2 文件组成及功能

BSP 文件中包括源代码、头文件、Makefile、派生文件以及二进制驱动程序模块。在 VxWorks 中 驱动程序分为通用性和 BSP 专用性两类。在开发 BSP 时 既要为 BSP 专用的驱动程序提供支持 还要将通用的设备驱动程序集成在一起。因此 编写的设备驱动程序 可根据具体情况放在 BSP 中的合适位置。

BSP 文件在 VxWorks 和 target 的 Works/target 文件夹里。其中 c、conf 文件夹里的文件是所有 BSP 的通用文件 ,bsp 文件夹里的文件是用户自己定制的 ,BSP 文件经过编译、链接 并在 makefile 等文件的控制下 原程序最后将生成镜像。BSP 的组成及主要文件功能如下。

(1) Target 目录下的文件包括 boot, boot, boot, boot 等 它们是所有 BSP 共享的 主要是缺省定义了所有 VxWorks 的配置以及 kernel 的初始化控制代码 所以不是特别需要不要更改里面的任何文件。

(2) Target 目录下的文件包括 makefile, boot, boot, boot, boot 等。其中 ronald 中包含特定 CPU 的所有包含文件和定义。

VxWorks支持多种处理器平台 用户可以针对不同的目标系统 通过修改这些文件来完成 BSP 的 VxWorks 在不同平台上的移植。

3 Vxworks映像分类及其启动过程

3.1 VxWorks映像及其分类

VxWorks在启动过程中涉及到两个映像文件：bootrom映像和 VxWorks映像。这两个映像文件都要使用 BSP 代码。bootrom文件是从 ROM 引导 VxWorks 的引导目标模块 用于对目标板进行初始化 加载 VxWorks 操作系统映像。VxWorks image 包含 VxWorks OS 最终运行于目标板上。VxWorks 映像按加载类型可分为 可下载映像和可引导映像 按运行方式分为在 ROM 和在 RAM 两种。

可下载映像 (Loadable Image) 它实际包括两部分 一是 VxWorks 二是 boot ROM 两部分是独立创建的。可引导映像 (Bootable Image) 将引导程序和 VxWorks 融为一体的映像 一般为最终产品 包括不驻留 ROM 映像和驻留 ROM 映像两种类型。

VxWorks 在 ROM 中运行 只要把 VxWorks image 的 data 复制到 RAM 的 RAM-LOW-ADRS 部分 留在 ROM 并在 ROM 中运行 VxWorks 可以节省 RAM 空间 有利于应用程序有更大的空间运行 但是运行速度相对较慢。VxWorks 在 RAM 中运行 写入 ROM 的 boot 或 VxWorks image 是压缩的 需先解压并复制所有的 text 或 data 到 RAM 的 RAM-LOW-ADRS 在 RAM 中运行。

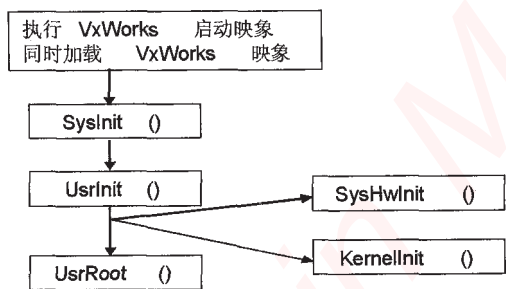


图 1 VxWorks 可下载映像启动顺序

3.2 VxWorks 映像的启动流程

VxWorks 应用系统的启动顺序 一般是先加载 BootRom; BootRom 再加载 VxWorks 应用 并跳转到 VxWorks 的入口 SysInit 执行。SysInit 复位 CPU 执行最少而必要的硬件初始化 转而调用 UserInit 完成内核启动之前的初始化。UserInit 调用子程序激活 VxWorks 内核。首先子程序 SysHwInit 将硬件置于初始的安全状态 然后子程序 KernelInit 激活 VxWorks 内核 生成多任务环境 并产生一个任务来安装驱动程序以及创建设备、初始化 VxWorks、调用应用程序启动代码 而用来生成这个任务的子程序是 UserRoot (可见 通用的设备驱动程序应该位于 UserRoot() 中 而 BSP 用的设备驱动程序则位于子程序 SysH-

WInit)。整个启动过程完成后 系统进入多任务环境运行。VxWorks 再加载映像的启动顺序如图 所示。

基于 PC486 的 BSP 设计过程及应用程序加载

4.1 BSP 设计过程

BSP 开发处于整个嵌入式开发的前期 是后面系统上应用程序能够正常运行的保证。大体上 , BSP 的设计与开发可以分为以下几个步骤 :

(1)建立开发环境。安装 VxWorks 集成开发环境 Tornado 采用宿主机目标机交叉开发。

(2)选择合适的 BSP 模板。一个合适相近的 BSP 模板可以大大减少开发周期。

(3)bootrom 的运行。修改、添加 WIN 内核激活前的初始化代码。

(4)VxWorks 的配置。修改或添加所需要组件及驱动程序 激活内核 实现基本的操作系统功能。

(5)测试与验证。正确实现 bootrom 及 VxWorks 的下载 根据不同需要 进行修改与测试。

结合实际中的工程 针对单串口的 pc486 目标机 , 在串口通信方式下 如何生成 BSP 及 VxWorks 系统的启动流程做一简单介绍。VxWorks 主机工具与目标机交互关系如图 所示。

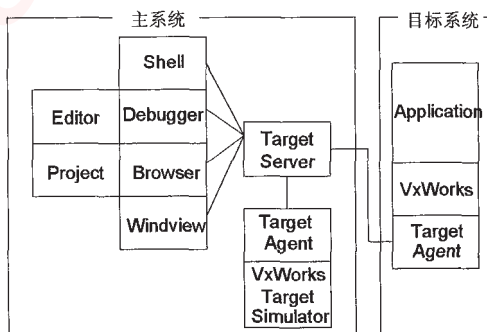


图 2 VxWorks 主机工具与目标机交互关系

建立所需目录。

在 c:\Tornado\target\config 中建立子目录 all486, sr1486 拷贝 c:\Tornado\target\config\all 中的全部文件到子目录 all486, 拷贝 c:\Tornado\target\config\pc486 中的全部文件到子目录 sr1486 以便修改且不影响其他人的使用。

修改 makefile

修改或添加以下语句 :

TARGET_DIR =sr1486, TARGET_DIR 的 bsp 目录名 将其指向要修改的 bsp 目录。

CONFIG_ALL=.\all486 将缺省 ALL 的文件复制到 all486 目录 在 all486 目录下做自己的修改。

正确理解四个地址的含义 如下 :

ROM_BASE_ADRS: rom 的起始地址。

ROM_TEXT_ADRS: rom 映像的起始地址。

ROM_SIZE: ro的大小。

RAM_HIGH_ADRS: VxWork应用在存储器低端定位开始的地址。

上述地址根据目标机实际的硬件情况进行设置，在 config.h, makefile 和 makefile 文件中都要定义且大小必须要保持一致。Makefile 的上述地址不以 0x 开头与 config 有所区别。

(修改 config.h 这里 config 修改包括：

```
#define DEFAULT_BOOT_LINE \  
"tsfs (0,0)host:vxWorks h =192.168.0.66 e =  
192.168.0.98 u=target"
```

```
...  
#if TRUE  
#define INCLUDE_PC_CONSOLE  
#endif  
...  
#undef WDB_TTY_DEV_NAME  
#define WDB_TTY_DEV_NAME "/tyCo/0"  
#undef CONSOLE_TTY  
#define CONSOLE_TTY NONE  
#undef WDB_TTY_CHANNEL  
#define WDB_TTY_CHANNEL 0  
#undef WDB_COMM_TYPE  
#define WDB_COMM_TYPE WDB_COMM_SE-  
RIAL  
#undef WDB_TTY_BAUD  
#define WDB_TTY_BAUD 38400  
#define INCLUDE_TSFS_BOOT  
#undef INCLUDE_WDB_TSFS  
#define INCLUDE_WDB_TSFS  
...  
同时屏蔽掉与网络相关的组件。  
(生成 bootrom 和 VxWork 映像。在生成 bootrom
```

时，可在 Tornado 的集成环境下 Build 菜单中选择 Build Boot ROM 来创建指定类型的 Boot ROM。在生成 VxWork 映像时，可在 Tornado 的集成环境下 Build 菜单中选择 standard BSP Build 来生成 VxWork 映像。此外，也可以在命令行环境下利用 Makefile 创建各种映像类型。

根据 bootConfig 的说明，我们的在 config.h 中的设置使能了虚拟控制台，所以 bootrom 启动后我们在 Tornado 开发环境中，通过选中 configure target server 的 target server property 的 console and redirection 启动虚拟控制台。这样在虚拟控制台可以看到启动信息，且按下 " @ 之后开始加载 VxWorks image

4 基于 VxWork 的应用程序的加载

应用程序运行于 VxWork 操作系统上。对于已存在的应用程序，VxWork 要通过两种方式添加。一类是可下载的应用程序，一类是可启动的应用程序。

一个可下载的应用程序包含一个或多个目标对象模块。这些模块可以下载和动态链接到 VxWork 并从 shell 或调试器中启动。即允许目标模块装载到一个运行的系统中。通过创建一个可下载的工程，构建该工程，使用适当的映像启动目标。下载工程所产生的部分链接和变换的 .o 文件。

一个可启动的应用程序包含一个链接到 VxWorks 映像的应用程序。目标机启动时，可启动的应用程序开始运行。通过创建一个可启动的工程，添加应用程序文件，然后编辑 VxWork 初始化文件 usrAppInit 为应用程序的初始化和启动工程添加调用。

结束语

VxWork 以其良好的性能和持续发展能力在各种高精尖技术及实时性要求极高的领域得到了越来越广泛的应用。同时希望能将其移植到更多的硬件平台上。本文对基于 VxWork 的 BSP 进行了分析，并在实际工程中得到了实现。但是针对不通的 CPU 板，BSP 的实现不同，需根据具体的硬件板进行具体设计，这是嵌入式设计的难点所在。

本文作者创新点，在运用 BSP 技术解决 VxWorks 移植的过程中，提出了串口通信以及其相应的应用程序的加载方法。

参考文献：

- [1] WindRiver System Inc. VxWorks BSP Developer's Guide 5.5 [M]. 2002.
- [2] 学龙 嵌入式 VxWorks 系统开发与应用 [M]. 京：人民邮电出版社，2003.
- [3] 智育 . VxWorks 程序开发实践 [M]. 京：人民邮电出版社，2004.
- [4] WindRiver System Inc. Tornado User's Guide 2.2 [M]. 2003.
- [5] 建恒 嵌入式系统应用研究及实例 [J]. 计算机信息，2004, 6: 65-66

作者简介：隋霞，女，汉族，1981年生，西安电子科技大学硕士研究生。主要研究嵌入式开发。Email: suixia16@163.com。许录平，男，汉族，1961年生，西安电子科技大学教授、博士生导师。主要研究方向：图象分析和视频处理、信号检测与目标探测跟踪技术、信息隐藏与信息安全技术。

Biography: Sui Xia, female, the Han nationality, born in 1981, graduate student of Xidian University, majored in embedded OS. Xu, Luping, male, the Han nationality, born in 1961, professor of Xidian University, research on image analysis and video processing, signal detecting and target tracking technology, information hiding and information safety technology.

(71007 西安电子科技大学电子工程学院) 隋霞 许录平

(School of Electronic Engineering, Xidian University, Xi-an 710071, China) Sui, Xia Xu, Luping

通讯地址：(71007 西安电子科技大学 13 信箱) 隋霞

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](#)
70. [CAN 总线的浅析 CANopen 协议](#)
71. [基于 CANopen 协议实现多电机系统实时控制](#)
72. [以太网时钟同步协议的研究](#)
73. [基于 CANopen 的列车通信网络实现研究](#)
74. [基于 SJA1000 的 CAN 总线智能控制系统设计](#)
75. [基于 CANopen 的运动控制单元的设计](#)
76. [基于 STM32F107VC 的 IEEE 1588 精密时钟同步分析与实现](#)

77. [分布式控制系统精确时钟同步技术](#)
78. [基于 IEEE 1588 的时钟同步技术在分布式系统中应用](#)
79. [基于 SJA1000 的 CAN 总线通讯模块的实现](#)
80. [嵌入式设备的精确时钟同步技术的研究与实现](#)
81. [基于 SJA1000 的 CAN 网桥设计](#)
82. [基于 CAN 总线分布式温室监控系统的设计与实现](#)
83. [基于 DSP 的 CANopen 通讯协议的实现](#)
84. [基于 PCI9656 控制芯片的高速网卡 DMA 设计](#)
85. [基于以太网及串口的数据采集模块设计](#)

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](#)
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](#)
50. [基于 Tilcon 的嵌入式系统人机界面开发技术](#)
51. [基于 Tilcon 的指控系统多任务人机交互软件设计](#)
52. [基于 Tilcon 航海标绘台界面设计](#)
53. [基于 Tornado 和 Tilcon 的嵌入式 GIS 图形编辑软件的开发](#)
54. [VxWorks 环境下内存文件系统的应用](#)
55. [VxWorks 下的多重定时器设计](#)
56. [Freescale 的 MPC8641D 的 VxWorks BSP](#)
57. [VxWorks 实验五\[时间片轮转调度\]](#)
58. [解决 VmWare 下下载大型工程.out 出现 WTX Error 0x100de 的问题](#)
- 59.

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](#)
52. [Linux 基于 W83697 和 W83977 的 UART 串口驱动开发文档](#)

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](#)
29. [Windows CENET 下 ADC 驱动开发设计](#)

30. [Windows CE 下 USB 设备流驱动开发与设计](#)
31. [Windows 驱动程序设计](#)
32. [基于 Windows CE 的 GPS 应用](#)
33. [基于 Windows CE 下大像素图像分块显示算法的研究](#)
34. [基于 Windows CE 的数控软件开发与实现](#)

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](#)
40. [嵌入式系统中热备份双机切换技术研究](#)

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](#)
10. [基于小波变换与偏微分方程的图像分解及边缘检测](#)
11. [基于图像能量的布匹瑕疵检测方法](#)
12. [基于 OpenCL 的拉普拉斯图像增强算法优化研究](#)
13. [异构平台上基于 OpenCL 的 FFT 实现与优化](#)
14. [数据结构考题 - 第 4 章 串](#)
15. [数据结构考题 - 第 4 章 串答案](#)
- 16.

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. [基于 FPGA 和 DSP 的 DBF 实现](#)
3. [高速浮点运算单元的 FPGA 实现](#)
4. [DLMS 算法的脉动阵结构设计及 FPGA 实现](#)
5. [一种基于 FPGA 的 3DES 加密算法实现](#)
6. [可编程 FIR 滤波器的 FPGA 实现](#)
7. [基于 FPGA 的 AES 加密算法的高速实现](#)
8. [基于 FPGA 的精确时钟同步方法](#)
9. [应用分布式算法在 FPGA 平台实现 FIR 低通滤波器](#)
10. [流水线技术在用 FPGA 实现高速 DSP 运算中的应用](#)
11. [基于 FPGA 的 CAN 总线通信节点设计](#)

RT Embedded <http://www.kontronn.com>

12. [基于 FPGA 的高速时钟数据恢复电路的实现](#)
13. [基于 FPGA 的高阶高速 FIR 滤波器设计与实现](#)
14. [基于 FPGA 高效实现 FIR 滤波器的研究](#)
- 15.