RT Embedded http://www.kontronn.com

基于工控级 AT91RM9200开发板的U-Boot移植分析

何剑锋[1],刘思颂[1],李 燕[2]

(1.成都理工大学 核技术与自动化工程学院,成都 610059; 2.南昌市公路管理局,南昌 330007)

摘要: Boot Loader是操作系统和硬件的枢纽,负责初始化硬件和引导加载操作系统内核。介绍采用 U-Boot (Universal Boot Loader)构建 ARM-Linux的引导加载程序 Boot Loader,详细阐述 U-Boot1.1.4代码体系结构和工作流程,并具体分析 U-Boot在 AT91RM9200开发板上的移植方法和过程。

关键词: Boot Loader; U-Boot; AT91RM9200; ARM-Linux;移植

中图分类号: TP316 **文献标识码**: B **文章编号**: 1000-3932(2007)03-0038-04

1 引 言

在专用的嵌入式开发板上运行操作系统必须利用 Boot Loader来引导加载其内核和系统程序,通过 Boot Loader这段引导程序,可以初始化硬件设备、建立内存空间的映射图,从而将系统的软硬件很好地衔接起来,以便为调用操作系统内核准备好正确的环境[1]。

U-Boot是当前比较流行、功能强大、最具弹性的 Boot Loader,不仅可支持多种体系结构的处理器,如 PowerPC、ARM (ARM7, ARM9, StrongARM, XScale)、MIPS(4Kc, 5Kc)、x86、Motorola (MC6800)和 ColdFire等系列,还支持 Linux、Solaris、VxWorks、LynxOS、QNX、NetBSD、RTEMS、ARTOS等多种目标操作系统,同时提供了完备的命令体系。因此对 U-Boot的研究与应用有着深远的意义。由于 Boot Loader代码与 CPU的体系结构、使用的操作系统、具体嵌入式板级设备的配置等因素密切相关,本文针对工业控制级 AT91RM9200开发板和 U-Boot对嵌入式 Linux的完善支持,具体阐述 U-Boot的软件结构框架、运行机理,实践了 Boot Loader(U-Boot1.1.4)移植到开发板上并可靠运行,为 ARM9微控制器的后续移植嵌入式 Liunx操作系统与系统软件开发创造良好条件[2]。①

2 U-Boot的源代码结构

U-Boot支持多种开发板配置,且 U-Boot的源码也相当复杂,因此在移植 U-Boot之前需要结合目标板的硬件资源及其 U-Boot的源码体系结构进行分析和设计才能使嵌入式系统的软硬件很好地配合工作。

2.1 开发板的硬件资源

开发板是基于 AT91RM9200处理器为核心的开发平台,其主要硬件资源如下:

(1) AT91RM9200处理器为 ARM920T内核 16/32 bit RISC CPU,在 180 MHz时运行速度高达 200 MIPS,内带 16 MB的数据 Cache, 16 KB指令 Cache,全功能的虚拟内存管理单元 MMU,系统时钟采用 18 MHz晶振。

(2) 16 MB的并行 Flash(E28F128J3A150)、32 MB的 SDRAM、带有 IDE/CF卡、SD卡、SMC卡接口。

(3) 10 M/100 M以太网接口(DM9161E)。

(4) 2个全速的 USB2.0主接口和一个从接口。

(5) SPI接口; 2通道 UART串行接口作为调试通道; JTAG调试接口。

(6) TFT/STN LCD接口。

2.2 U-Boot的源码目录结构[3]

U-Boot采用了高度模块化的编程方式,文中以 U-Boot1.1.4最新版本来分析 U-Boot代码结构并根据 AT91RM9200目标板硬件资源列出与移植相关的几个重要目录和文件。移植相关的源码结构如图 1所示。

(1) board:该目录下是与一些特定开发板相关的初始化和操作代码,主要是 Makefile、SDRAM、Flash、U-Boot lds等都和具体开发板的硬件和地址分配有关。

(2) common:此目录存放与处理器体系结构无关的公用代码,U-Boot的一些公共命令的实现,如内存大小探测、故障检测等,其中与内核引导相关的是 cmd_Boot c和 cmd_Bootm. c。

① 收稿日期: 2007-05-08(修改稿)

RT Embedded http://www.kontronn.com

(3) CPU:存放 CPU相关的代码,其子目录都是以 U-Boot所支持的 CPU为名,每个特定的子目录中都包括 cpu. c和 interrupt c, start S。其中 cpu. c初始化 CPU、设置指令 Cache和数据 Cache等; interrupt c设置系统的各种中断和异常; start S是 U-Boot启动时设置系统堆栈、工作方式和内存时序等。

(4) drivers:各种通用设备驱动程序,如网卡、支持 CFI的 Flash、串口和 USB总线等就存放在该目录下。

(5) fs:包含支持文件系统的文件, U-Boot现在支持 cramfs、ext2、fdos、jffs2和 registerfs。

(6) include:包含各种硬件平台相应的头文件,系统的配置文件和对文件系统支持的文件。其中的 include/congfigs/at91rm9200dk. h定义了所有和 AT91RM9200相关资源的配置参数。

(7) net:与网络功能有关的代码,如 BOOTP、TFTP、RARP等协议和 NFS文件系统的实现。

(8) lib_arm:包含与 ARM体系结构相关的接口代码。

(9) post:上电自检的文件目录。

(10) tools:用于创建 S-Record、U-Boot images和 BIN文件格式的工具。

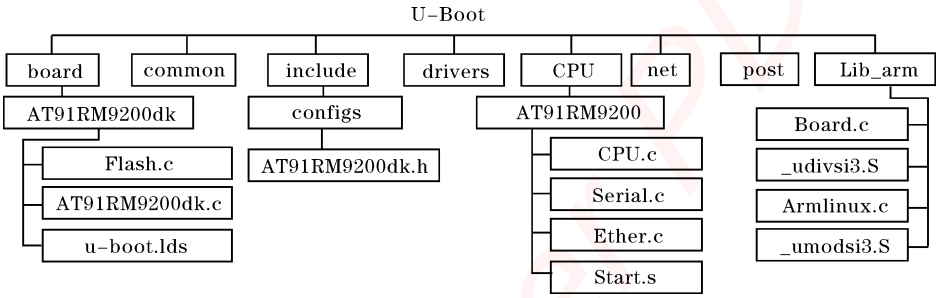

图 1 U-Boot(AT91RM9200)的源码结构

3 U-Boot运行流程

U-Boot对嵌入式 Linux的完善支持可使其成为一种通用的 Linux Boot Loader,并与硬件有高度紧密的依存关系,当 Boot Loader被下载到 Flash的 0x0地址处,作为上电后执行的第一部分代码需要完成两个主要任务[4]: ①初始化板载硬件设备并建立内存空间的映射图; ②把操作系统的内核装载到 SDRAM里合适的位置上。

U-Boot启动流程如图 2所示,系统上电或复位后,立即跳转到 Flash的最低地址 0x0处执行 start S,该文件是 AT91RM9200开发板的入口代码[5]。Flash的开始 1 K空间为中断矢量表,将表中第一个 32位字传给 SP,第二个 32位字(_start的地址)传给 PC,完成复位中断跳转。该复位中断跳转到 start S中的 _start处,然后就是对 CPU的初始化操作,并确定是否需要对整个 U-Boot代码重定位,最终从 Flash中跳转到定位好的内存位置执行,这样第一阶段的使命已完成。第二阶段的起始地址是由第一阶段代码中指定的,被复制到 SDRAM中后,就从第一阶段跳转到这个入口地址开始执行第二阶段的代码,通过下面的语句跳转到 C代码执行:

```
ldr pc, _start_armBoot
_start_armBoot: . word start_armBoot
```

主要包括 /lib_arm/board. c文件中的函数 start_armBoot()和 /common/main. c文件中的 main_loop()函数。其任务是继续进行硬件的初始化, BSS段的设置,堆栈的初始化工作。

图 2 U-BOOT的启动流程

要针对开发板硬件资源配置和 CPU的复位地

RT Embedded http://www.kontronn.com

址修改 U-Boot的源码并进行移植工作,还须对存储器的地址空间分布进行了解,存储器的存储空间在目录 /include/configs/AT91RM9200dk. h 中定义。Flash存储器映射如表 1所示。

表 1 Flash存储器分布情况

	开始地址	结束地址
Boot映像文件	0X10000000	0X10005FFF
自由空间	0X10006000	0X1000FFFF
U-Boot压缩映像	0X10010000	0X1001FFFF
U-Boot环境变量	0X10020000	0X1003FFFF
自由空间	0X10040000	0X1005FFFF
内核映像	0X10060000	0X101FFFFF
文件系统映像	0X10200000	0X11000000

4 移植 U-Boot到 AT91RM9200开发板

4.1 建立 U-Boot交叉开发环境

在嵌入式系统开发中,通常利用交叉编译工具在 PC机上编译产生目标平台的目标代码,在开发板上运行这些目标代码,我们称这种编译方式为“交叉编译 ”。在本开发板中使用官方网站下载的编译工具 cross-2. 9. 3. tar. bz2,解压开发环境就建立在 /usr/local/2. 95. 3/bin 下面,包括 binutils二进制工具、gcc交叉编译器、glibc库。如以 arm-linux为前缀的编译工具。

4.2 U-Boot的系统移植

在 U-Boot系统移植需要修改其部分配置文件和底层接口文件才可以在目标板上运行起来。首先根据目标板实际情况对 U-Boot 的 /board/AT91RM9200dk目录中进行修改或添加部分源代码[4,6]。

(1)因 Flash存储器种类繁多,本目标板的移植必须根据 StrataFlash(E28F128J3A150)数据手册重新编写 Flash设备控制程序 Flash. c,包括打印 Flash扇区分配信息、初始化、确定 Flash大小(16M)、读/写 Flash、删除 Flash、并分配扇区和保护 Flash的前四个扇区(主要存放 U-Boot映像代码)等。

(2)添加 memsetup. S汇编文件,主要是初始化时钟、SMC控制器和 SDRAM控制器等操作。

(3)添加网络芯片 DM9161E的设备控制程序 dm9161e. c和 dm9161e. h,该程序完成的功能是配置 AT91RM9200的 MDI接口以实现对 PHY进行控制。

(4)对上面修改或添加的源代码文件编译后,在相应的目录中 Makefile文件应做如下修改:

OBJS := at91rm9200dk. o at45. o dm9161e. o flash. o

SOBJS := memsetup. o

(5) AT91RM9200. c,设置各种总线时钟,打开数据 Cache和指令 Cache,并设置相关内存参数。

(6) U-Boot lds,内核可执行文件由许多链接在一起的对象文件组成,U-Boot将各个对象文件通过 U-Boot lds文件被链接并装入到内存中特定的偏移量处。主要作如下修改(其它基本不作改动):

```
. text:
{
  CPU/arm920t/start o(. text)
 * (. text)
}
```

(7) config mk,用于设置程序链接的起始地址,修改后 TEXT BASE = 0X21F10000。

其次,根据目标板的资源配置,内容包括 CPU、系统时钟、RAM、Flash等配置信息以及内存映射相关参数修改目录 /include/configs 的头文件 AT91RM9200dk. h,其头文件还定义了 U-Boot的一些环境变量和内核启动参数。U-Boot1. 1. 4版本对 AT91RM9200处理器提供良好的支持,因此对于目录 /CPU/at920t/AT91RM9200中的源码基本不做修改。

最后,在 U-Boot1. 1. 4的 Makefile文件中加入如下代码:

```
at91rm9200dk_config: unconfig
@. /mkconfig $ (@: _config = ) arm arm920t at91rm9200dk AT91RM9200;
```

对 AT91RM9200处理器进行配置。

4.3 U-Boot的编译与测试

在装有 Redhat9. 0 Linux操作系统的 PC机上编译 U-Boot。在编译工作进行之前,须检查 U-Boot\board\AT91RM920dk\config mk中 TEXT_BASE的值,因为在测试阶段的代码是在 RAM中运行的,而最后需要把这些代码固化到 Flash中,因此 U-Boot需要自己从 Flash转移到 RAM中运行,这是重定向的目的所在。然后键入 make命令编译 U-Boot源码生成 U-Boot bin的二进制文件,编译时先指定 Makefile文件中的 LINUX_INCLUDE_DIR和 CROSS_COMPILE,最后通过 JTAG接口将 U-Boot bin下载到开发板的 Flash中。

U-Boot启动后可在其命令模式下做进一步的测试,如利用 setenv命令设置系统启动的环境变量,通过交叉网线连接主机和目标机,利用 tftpboot命令从主机下载 Linux内核映像文件等。如果运行正常即

RT Embedded http://www.kontronn.com

可在目标板上开始 ARM Linux系统的调试和系统应用程序的开发。

5 小 结

综上所述,U-Boot是功能强大完备的大型 Boot Loader,相当于一个小型的操作系统。由于 U-Boot可支持多种体系结构的处理器和目标操作系统,使得许多嵌入式系统开发者对 U-Boot的研究和应用产生了浓厚的兴趣,许多知识点值得深入研究。同时,实现一个好的 U-Boot是一个庞大复杂的过程,直接关系到操作系统移植的稳定性和可靠性。笔者移植的 U-Boot能稳定地运行在 AT91RM9200目标板上,并在此基础上加载了 Linux内核和文件系统,为后续的应用开发奠定了坚实的基础。

参考文献:

[1] 孙天泽,袁文菊,等. 嵌入式设计及 Linux驱动开发指南——基于 ARM9处理器[M]. 北京:电子工业出版社, 2005.

[2] 赵铁峰,王 凯,王为民,等. 基于 ARM微处理器的智能控制器[J]. 化工自动化及仪表, 2005, 32(1): 79-80.

[3] Atmel Corporation. AT91RM9200DK U-Boot Flash Programming Solutions[EB/OL]. (2003)[2007-05-08]. http://www.Atmel.com.

[4] 朱继杭,杨世武. 基于 AT91RM9200的 U-Boot移植方法[J]. 仪器仪表用户, 2005, 12(6): 121-122.

[5] 马建华,李兰兰. 基于 M5272C3开发板的 U-Boot分析与应用[J]. 微处理机, 2006, 6(3): 74-77.

[6] DENK W. The DENX U-Boot and Linux Guide(DULG) for TQM8xxL[EB/OL]. (2005-02)[2007-05-08]. http://www.denx.de/twiki//bin/view/DULG/Manual

Porting Analysis of U-Boot Based on Industry Controlling AT91RM9200 Board

HE Jian-feng[1], LIU Si-song[1], LI Yan[2]

(1. *College of Nuclear Technology & Automation Engineering, Chengdu University of Technology; Chengdu* 610059, *China;* 2. *Authority Road of Nanchang, Nanchang* 330007, *China*)

Abstract: A BootLoader for ARM-Linux system with the U-Boot(Universal Boot Loader) is developed. The source code structure and the working stream of U-Boot1.1.4 are expatiated. The porting method and process in an embedded system board based on the AT91RM9200 CPU are analyzed.

Key words: BootLoader; U-Boot; AT91RM9200; ARM-Linux; porting

Created in Master PDF Editor

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

Creator Master PDF Editor

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. 基于磁盘异或引擎的 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

Create by Master PDF Editor

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

Creator Master PDF Editor

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

Created with Master PDF Editor

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 CC++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

Created by Master PDF Editor

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

Created with Master PDF Editor

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

Creator Master PDF Editor

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μC-OSⅡ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

Created with a-PDF PDF Editor

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 NiosⅡ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

Created with Master PDF Editor

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

Creator in Master PDF Editor

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.

Created in Master PDF Editor