

U-Boot 在 Mini6410 上的移植

Porting of U-Boot to Mini6410

宋耀华¹ 赵阳² 龚素文¹

(1 九江职业技术学院 电气工程学院,江西 九江 332007;2 尚扬电子科技有限公司,江苏 扬州 225000)

摘要

在深入分析 U-Boot 的基础上,对 Mini6410 开发板上的 u-boot-2011.06 移植步骤进行了详细说明,结合当前系统的软硬件环境,实现了 U-Boot 的开机 logo,这对于其它嵌入式 Bootloader 的移植也具有一定的参考价值。

关键词:U-Boot,S3C6410,嵌入式,移植

Abstract

Based on u-boot depth analysis,particularly explained the porting process about u-boot-2011.06 on Mini6410,combining with the system hardware and software features,realizes boot logo of u-boot,it has reference value for other embedded system Bootloader porting.

Keywords:U-Boot,S3C6410,embedded,porting

随着嵌入式技术的普及,专门为嵌入式产品开发的操作系统也层出不穷,如 Android,WindowsCE,Pocket PC,Vx-Works,Linux 等。因此,如何引导嵌入式操作系统成为越来越多讨论的热点——这就是引导与加载内核镜像代码 boot-loader 的讨论范畴。

嵌入式系统常用的 Boot Loader 有 ARM-Boot,RedBoot,U-Boot 等。U-Boot(全称 Universal Boot Loader)是当前比较流行的遵循 GPL 条件的开放源码项目。U-Boot 具有源码公开的特点,开发人员可根据自身需要进行裁减;支持多种处理器和嵌入式操作系统内核。本文针对 S3C6410 处理器,将 u-boot-2011.06 移植到该平台上。

1 开发环境的建立

1.1 宿主机-目标板开发模式

由于嵌入式 Linux 的开发板资源有限,不可能在开发板上运行开发和调试,通常采用“宿主机+目标板(开发板)”的形式。宿主机和目标板一般通过串口、USB 或网线连接,如图 1 所示。

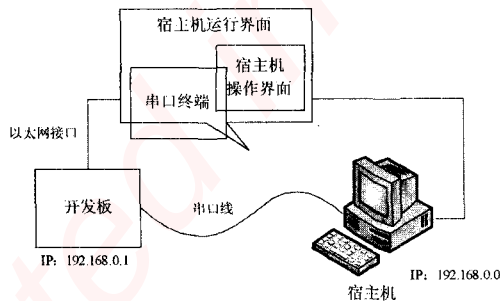


图 1 宿主机-目标板开发模式

大部分嵌入式系统在宿主机(PC)大多都采用 Telnet 或 Minicom 实现与目标板的通信,本系统采用 Minicom。Minicom 是 Linux 下一个类似于 Windows 超级终端的友好串口通信程序。在终端输入 bash#minicom-s 进入 Minicom 设置画面,设置串口波特率、有效数据位、停止位以及奇偶校验位分别为 115200,8 bit,1 位停止位、无奇偶校验位以及无流控制等。

1.2 建立交叉编译环境

开发环境:

宿主机系统 Windows XP SP3 系统;

虚拟机采用的是 VMware Workstation ACE 版本;

虚拟机内采用的 Linux 系统是 UBUNTU10.10 版本;

交叉编译工具(动手制作)。

交叉编译是在一个架构下编译另外一个架构的目标文件。

要从 <http://ymorin.is-a-geek.org/download/crosstool-ng/> 处下载 crosstool-ng,用 crosstool-ng 建立 arm-linux 交叉工具链,使用的交叉编译器和交叉调试器分别是 gcc-4.3.2 和 gdb-6.8,Linux 内核为 linux-2.6.33。该交叉编译工具经过验证,成功编译 U-Boot、Linux Kernel、BusyBox 等。

在编译好交叉编译工具链后,关键是要在环境变量的 PATH 中添加编译工具的路径(也就是 arm-* -linux-* -gcc 所在的路径),在编译时系统才找得到命令。

在 Ubuntu 下的修改方法:

```
vi ~/.profile
```

并在最后加上:

```
PATH="<交叉编译工具的路径>:$PATH"
```

2 U-Boot

2.1 U-Boot 启动分析

U-Boot 启动过程如图 2 所示,分为两阶段。依赖于 CPU

体系结构的代码,比如设备初始化代码等,通常都放在第一阶段中,通常用汇编语言来实现,以达到短小精悍的目的。而第二阶段则通常用 C 语言来实现,这样可以实现给复杂的功能,而且代码会具有更好的可读性和可移植性。主要包括对 Flash 设备的初始化,内存分配,硬件设备进行初始化等任务,完成后,进去命令循环,接受用户从串口输入的命令

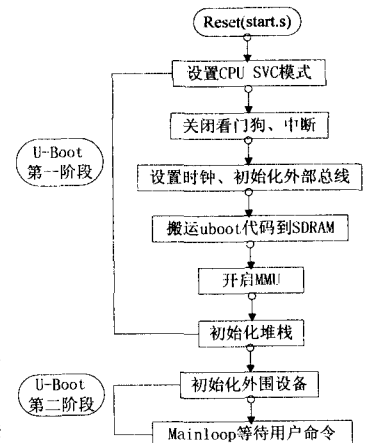


图 2 U-Boot 启动流程图

令,然后进行相应的工作。

2.2 U-Boot 的移植

第一步:下载 U-Boot 代码

在 <ftp://ftp.denx.de/pub/u-boot/> 网站下载得到最新的 U-Boot。下载后把它解压,然后得到 u-boot-2011.06 的文件夹。

第二步:建立自己的开发板资源

1)在 u-boot-2011.06 目录下 Makefile 文件中增加开发板配置信息 mini6410_config。

```
mini6410_config : unconfig
```

```
@$(MKCONFIG) mini6410 arm arm1176 mini6410 adward s3c64xx
```

2)修改编译器 CROSS_COMPILE 路径为自己的编译器所在的路径。

3)在/board 子目录中建立自己的开发板 Mini6410 目录。

将/u-boot-2011.06/board/samsung/smdk6400/所有文件复制到/u-boot-2011.06/board/adward/mini6410 下并将 smdk6400 全部更名为 mini6410。

4)修改自己的开发板 Mini6410 目录下的 Makefile 文件,不然编译时会出错。

```
root@localhost mini6410# vi Makefile
```

```
COBJS := mini6410.o flash.o
```

5)修改 u-boot-2011.06 目录下的 boards.cfg 文件,不然编译时会出错。

```
mini6410 arm arm1176 mini6410 samsung s3c64xx
```

6)修改 u-boot-2011.06/board/samsung/mini6410/low-level_init.S 文件,去除有关 smdk6400 的 LED 相关配置信息。

到此为止,U-Boot 对自己的 Mini6410 开发板还没有任何用处,以上的移植只是搭建了一个 Mini6410 开发板 U-Boot 的框架,要使其功能实现,还要根据 Mini6410 开发板的具体资源情况来对 U-Boot 源码进行修改。

第三步:芯片级移植

这一步也是最关键的一步,它使 U-Boot 可以支持芯片 S3C6410 正常启动工作,但没有增加任何附加功能。主要参阅开发板的硬件说明和芯片手册,主要完成 GPIO、PLL 的设置,应注意 GPBCON 的设置,不要让蜂鸣器响。

1)修改 board_init() 函数。

2)修改 checkboard() 函数

3)屏蔽函数 ulong board_flash_get_legacy()。

第四步:完成外围设备驱动(串口、DM9000 网卡、MMC、LCD 等)

我们下载的 U-Boot 的代码中实际是有 DM9000 的网口驱动的,位于 driver/net/dm9000x.c,DM9000 的驱动没有太大的问题,修改了一点可能出现问题的地方。

在 u-boot-2011.06/drivers/net/dm9000x.c 文件中修改函数 DM9000_ior()

```
DM9000_ior(int reg)
```

```
{
...
    rxbyte = DM9000_inb(DM9000_DATA) & 0x03;
    u8 temp;
    temp=DM9000_ior(DM9000_MRRH);
    temp=DM9000_ior(DM9000_MRRL);
    #endif
...
}
```

第五步:更新配置头文件,在以下文件中将外围硬件参照参阅开发板的硬件说明和芯片手册做相应的修改。

```
arch\arm\include\asm\arch-s3c64xx\s3c64x0.h
```

```
include\common.h
```

```
u-boot-2011.06\drivers\net\dm9000.h
```

```
include\configs\mini6410.h
```

第六步:编译 U-Boot。

```
Make mini6410_config
```

```
Make
```

第七步:下载编译好的 U-Boot.bin 文件。

```
U-Boot 2011.06 (Jan 04 2012 - 20:37:58) for MINI6410
```

```
CPU: S3C6410@533MHz
      Fclk = 533MHz, Hclk = 133MHz, Pclk = 66MHz (ASYNC Mode)
Board: MINI6410
DRAM: 128 MB
Flash: 0 Bytes
NAND: 256 MB
*** Warning - bad CRC, using default environment

In: serial
Out: serial
Err: serial
Net: No ethernet found.
Hit any key to stop autoboot: 0
```

图 3 U-Boot 运行结果

3 结束语

本文研究了 U-Boot 在 S3C6410 系统上的移植,通过调试,实现板级硬件的初始化与正确引导 Linux 系统功能(如图 3 所示),为后续的系统驱动程序开发奠定了基础,这种 U-Boot 的移植开发具有通用性,对嵌入式开发人员具有一定的参考价值。

参考文献

- [1]杨明极,张明辉.U-Boot 的分析及其在 DM6446 上的移植[J].计算机应用,2011,30(4):34-37
- [2]王志平,杨国武,李晓瑜.SST39VF040 NorFlash 芯片的软件仿真实现[J].微型计算机系统,2011,32(12):2511-2514.
- [3]程伟.支持 Yaffs2 文件系统的 U-Boot 的实现[J].计算机工程与设计,2012,33(3):936-940
- [4]蔡利平,任家富,童锐,等.基于 ARM 的 Nand Flash 启动分析与移植[J].计算机工程与设计,2011,32(12):2511-2514
- [5]Samsung Electronics S3C6410X USR'S MANUAL (REV 1.20) [K].Korea: Samsung Electronics,2009:1-1371

[收稿日期:2013.4.3]

(上接第 25 页)

3)该系统以简明的规则库机制对 ECG 数据质量进行评估,能够在线实时进行。

本文的后续工作将会重点放在 ECG 质量评估规则库的扩展方面,计划增加更多规则,并尝试规则组合推理评估。

参考文献

- [1]Gelge Svoboda.心电图仪设计综述[J].电子技术应用,2010(11):

6-10

- [2]S J Redmond, Y Xie, D Chang, J Basilakis and N H Lovell, Electrocardiogram signal quality measures for unsupervised telehealth environments, Physiological Measurement, vol. 33, 2012, pp.1517-1533
- [3]Health Level Seven International, <http://www.hl7.org/>. 2013
- [4]李刚.疯狂 Android 讲义[M].北京:电子工业出版社,2011

[收稿日期:2013.5.27]

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](#)

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](#)

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 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](#)
46. [基于 Android 平台的医护查房系统的研究与设计](#)

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](#)
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](#)

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](#)

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](#)

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](#)

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](#)