

## 基于 S3C6410 的 Uboot 分析与移植

冯林琳 耿恒山

河北工业大学计算机科学与软件学院 天津 300401 )

**摘要** 分析 Uboot 新版源代码结构及其启动流程 给出基于 S3C6410 芯片开发板上 Uboot 的详细移植方案 实现从 NAND Flash 中启动 Uboot。通过对 DM9000A 网卡的移植 完成 Uboot 的网络通信功能。对类似平台的 Uboot 移植有一定的参考价值。

**关键词** BootLoader S3C6410 Uboot

中图分类号 TP368 文献标识码 A doi 10.3969/j.issn.1006-2475.2013.01.034

### Uboot Analysis and Transplantation Based on S3C6410

FENG Lin-lin , GENG Heng-shan

School of Computer Science and Engineering , Hebei University of Technology , Tianjin 300401 , China )

**Abstract** This paper analyzes the structure and the starting process of the new edition Uboot , gives a detailed transplant program based on S3C6410 development board , to achieve Uboot start from NAND Flash. Through the transplant of the net card DM9000A , Uboot network communication function is implemented. The paper has certain reference value for similar platform Uboot transplant.

**Key words** BootLoader S3C6410 Uboot

## 1 移植环境和 BootLoader 介绍

### 1.1 移植 Uboot 的软硬件环境

本文采用的 S3C6410 芯片是基于 ARM11 架构的 16/32 位 RISC 微处理器。它采用 ARMv6 指令集 , 支持从 SD Card、NOR Flash、NAND Flash 和 One-NAND Flash 中引导系统 , 是一个低成本、低功耗、高性能的应用处理器解决方案。开发板主要硬件资源见表 1。

表 1 开发板主要硬件资源

名称	型号	说明
CPU	S3C6410	主频为 400 ~ 677MHz
DDR RAM	K4X1G163PC	64M × 16bit × 2 片
NAND Flash	K9F2G08U0C	256M × 8Bit SLC
网卡	DM9000A	10 ~ 100Mbit 自适应

本文使用 Kubuntu 12.04 作为编译系统 , 采用 arm-linux-gcc-4.7.0 工具链作为编译器。开发板与目标机通过串口和网口链接 , 并使用 Jlink V8 仿真器进

行调试和下载。

### 1.2 BootLoader 介绍

BootLoader 是嵌入式设备上电后执行的第一段程序 , 负责初始化硬件设备、建立内存空间映射图 , 引导嵌入式操作系统内核 , 其功能同 X86 体系结构中 BIOS 基本相同。BootLoader 通常使用 SD Card、NOR Flash 和 NAND Flash 启动。目前 , 常用的 BootLoader 主要为 vivi 和 Uboot , 其中 Uboot 又占据了绝大部分份额。

## 2 Uboot 结构与启动流程分析

### 2.1 Uboot 结构分析

Uboot 为德国 DENX 小组开发的嵌入式通用 BootLoader。它支持 ARM、MIPS、XScale 等多种体系结构 , 并支持引导 Linux、VxWorks、NETBSD 等多种操作系统。本文采用 Uboot-2012-4 版本。

Uboot 共有 18 个目录 , 按层次和调用关系如图 1 所示。

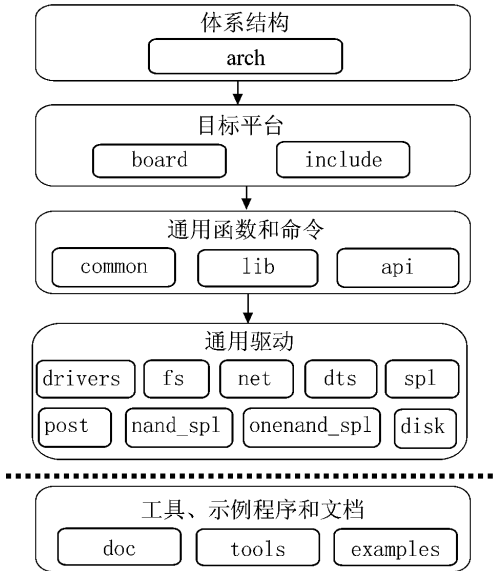


图1 UBoot 源代码层次和调用关系

Uboot 新版本相对之前的旧版本,层次结构和调用关系更加清晰,对于移植新平台非常有利,降低了移植的难度和时间。

## 2.2 Uboot 启动流程分析

Uboot 的启动分为两个阶段。第一阶段主要为汇编代码,涉及 `cpu/arm1176/start.s`, `board/开发板目录/lowlevel_init.S` 和 `arch/arm/lib/board.c` 中的 `board_init_f` 函数。完成如下工作:

- 1( 设置 CPU 为 SVC 模式,关中断,关看门狗,关 MMU );
- 2( 设置时钟,初始化各硬件控制器 );
- 3( 设置堆栈 );
- 4( 复制 重定位 Uboot 到内存中 );
- 5( 跳转到第二阶段代码入口。

第一阶段流程图如图 2 所示。

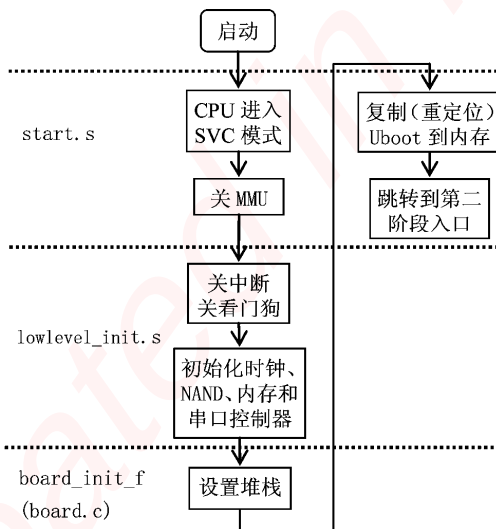


图2 第一阶段流程图

因为在第一阶段已经设置好堆栈,所以从第二阶

段开始使用 C 语言实现。在 `arch/arm/lib/board.c` 的 `board_init_r` 函数中 `uboot` 继续初始化各硬件资源,如串口、网卡、NAND Flash 等。然后跳转至死循环 `common/main.c` 中的 `main_loop` 函数,启动 Linux 内核或等待用户输入命令。

## 3 Uboot 移植

针对本文采用的开发板,使用与其硬件资源相近并且 Uboot 已经支持的 SMDK6400 开发板实现移植。本开发板没有配置 NOR Flash,因此要实现从 NAND Flash 中启动 UBoot。

### 3.1 创建工程目录

1( 在 `board\samsung` 下,新建文件夹 S3C6410。将 SMDK6400 目录下所有文件拷贝到 S3C6410 中,重命名 `smdk6400.c` 和 `smdk6400_nand_spl.c` 为 `s3c6410.c` 和 `s3c6410_nand_spl.c`,修改 Makefile 中 `COBJS-y = smdk6400.o` 为 `COBJS-y = s3c6410.o`。

2( 在 `nand_spl\board\samsung` 下,新建文件夹 `s3c6410`,拷贝 `smdk6400` 目录下所有文件到 `s3c6410` 中。重命名 `smdk6400_nand_spl.c` 为 `s3c6410_nand_spl.c`,修改 makefile 中 `smdk6400` 字段为 `s3c6410`。

3( 在 `include\configs` 目录下,拷贝 `smdk6400.h` 并重命名为 `s3c6410.h`。

4( 修改根目录下的 makefile,依据 `smdk6400` 添加 `s3c6410` 编译规则。

### 3.2 实现 NAND Flash 启动 Uboot

ARM 体系结构的微处理器上电后从地址 `0x0` 开始执行第一条代码。随着时代的发展,NAND Flash 凭借容量大、速度快、价格低等优点逐渐取代 NOR Flash。许多嵌入式平台上已经不再配备 NOR Flash。

NOR Flash 可以按地址读取,NAND Flash 只能按页读取,因此不能直接从 NAND Flash 中启动 Uboot。当 S3C6410 芯片配置成从 NAND Flash 启动模式时,系统上电后通过 NAND Flash 控制器自动读取 NAND Flash 中前 4k 内容到 S3C6410 芯片内部的 stepping stone 中,并将其映射到地址 `0x0` 开始执行。所以,如果 S3C6410 要从 NAND Flash 中启动,就必须在程序前 4k 代码中将完整的 Uboot 从 NAND Flash 中复制到内存,然后跳转到内存中相应继续执行,完成启动过程。

从 NAND Flash 启动的 Uboot 由两部分组成:前 4k `uboot_nand_spl` 文件和正常 Uboot 文件。它们顺序相接,中间没有空隙。宏 `CONFIG_NAND_SPL` 是 `uboot_nand_spl` 的编译开关。当编译 `uboot_nand_spl` 时,第一阶段将由 `s3c6410_nand_spl.c` 中 `board_init_f` 函数替换 `board.c` 中的 `board_init_f` 函数。此处的 `board_init_f` 不设置堆栈直接跳转回 `start.S`,然后再从

start.s 跳转至 nand\_spl\nand\_boot.c 中的 nand\_boot 函数。nand\_boot 函数负责将 NAND Flash 中 4k 之后正常的 Uboot 复制到内存中,并跳转到内存中相应位置继续执行。

nand\_boot 函数用 C 语言实现,但第一阶段的 board\_init\_f 没有设置堆栈,所以这里使用的堆栈应该是 s3c6410 内部的 IRAM 地址为 0x0c00\_0000 ~ 0x0fff\_ffff,实际使用只有 4k。)

将 start.s 中的

```
stack_setup :  
mov sp, #4
```

修改为 :

```
stack_setup :  
#ifndef CONFIG_NAND_SPL  
mov sp, #4  
#endif
```

此时 sp 指向 S3C6410 内部的 IRAM。

### 3.3 添加 DM9000A 网卡

在 uboot-2012.4 版本中,已经实现对 DM9000A 网卡的支持,在此只需正确配置网卡信息,即可完成网卡驱动的移植。

开发板的所有硬件资源配置信息在 include/configs/s3c6410.h 中。本开发板使用的 DM9000A 网卡连接至 S3c6410 芯片的 Bank0,地址为 0x18000300,DM9000A。修改 s3c6410.h,添加以下 DM9000A 网卡配置信息 :

```
#define CONFIG_DRIVER_DM9000  
#define CONFIG_DM9000_BASE 0x18000300  
#define CONFIG_DM9000_NO_SROM  
#define DM9000_IOCONFIG_DM9000_BASE  
#define DM9000_DATA (CONFIG_DM9000_BASE + 4 )  
#define CONFIG_NET_MULTI
```

删除以下 cs8900 网卡配置信息 :

```
#define CONFIG_CS8900  
#define CONFIG_CS8900_BASE 0x18800300  
#define CONFIG_CS8900_BUS16
```

本文使用的开发板为 256MB 内存,需修改内存配置信息 :

将#define PHYS\_SDRAM\_1\_SIZE0x08000000 改为#define PHYS\_SDRAM\_1\_SIZE0x10000000 将#define CONFIG\_SYS\_MEMTEST\_END (CONFIG\_SYS\_SDRAM\_BASE + 0x7e00000) 改为#define CONFIG\_SYS\_MEMTEST\_END (CONFIG\_SYS\_SDRAM\_BASE + 0xfe00000)。

### 3.4 编译并烧写到 NAND Flash

在 Uboot 根目录下执行 make s3c6410\_config,生成编译配置文件,再执行 make 编译 Uboot 镜像。根目录生成的 u-boot.bin 镜像可在内存和 NOR Flash 中启

动 u-boot-nand.bin 镜像可在 NAND Flash 中启动。

NAND Flash 无法直接使用 Jlink 烧写,本文通过 Uboot 自身功能实现 NAND Flash 的烧写。

使用 Jlink 把 u-boot.bin 和 u-boot-nand.bin 下载到开发板内存中 0x50000000 和 0x50100000 处,运行 u-boot.bin。此时 Uboot 由内存中启动。执行 Uboot 的 nand erase 0x0 64000 擦除 NAND Flash 最前面 400kB 空间,执行 nand write 0x50100000 0x0 64000 将 u-boot-nand.bin 烧写到 NAND Flash。复位开发板实现从 NAND Flash 中启动 Uboot。

## 4 结束语

BootLoader 是嵌入式开发极其重要的一环,本文在分析 Uboot 原理后,结合最新 Uboot 源代码和编译环境,在 S3C6410 开发板上移植成功,并实现了 NAND Flash 启动和网络通信等功能。本方法为 S3C6410 后续开发奠定了基础,并对其它开发板的 Uboot 移植工作提供了一定的参考价值。

参考文献 :

- [1] 田泽. 嵌入式系统开发与应用 [M]. 北京: 北京航空航天大学出版社, 2005: 134-349.
- [2] 陈文智, 王总辉. 嵌入式系统原理与设计 [M]. 北京: 清华大学出版社, 2011: 53-178.
- [3] 陈贇. 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, 2011: 127-183.
- [4] Sumsuang Electronics. S3C6410x User's Manual [Z]. Sumsuang Electronics, 2008: 60-245.
- [5] Sumsuang Electronics. K9F2G08UXA Datasheet [Z]. Sumsuang Electronics, 2007: 2-44.
- [6] Davicom Semiconductor. DM9000A Datasheet [Z]. Davicom Semiconductor, 2006: 6-52.
- [7] Sumsuang Electronics. S3C6410x Application Note Internal ROM Booting [Z]. Sumsuang Electronics, 2008: 5-19.
- [8] 广州友善之臂计算机科技有限公司. Tiny6410 硬件说明手册 [Z]. 广州友善之臂计算机科技有限公司, 2011: 5-26.
- [9] [美] Daniel W Levis. 嵌入式软件基础—C 语言与汇编的整合 [M]. 陈宗斌译. 北京: 高等教育出版社, 2005: 182-194.
- [10] 刘凯. ARM 嵌入式接口技术应用 [M]. 北京: 清华大学出版社, 2009: 78-157.
- [11] 刘峰. ARM 汇编语言 [M]. 成都: 电子科技大学出版社, 2010: 10-63.
- [12] DENX. U-Boot Source Code [DB/OL]. <http://www.denx.de/wiki/U-Boot/SourceCode>, 2012-08-15.
- [13] DENX. The DENX U-Boot and Linux Guide [DULG for canyonlands] [DB/OL]. <http://www.denx.de/wiki/DULG/Manual>, 2012-08-15.
- [14] DENX. The Universal Boot Loader ("Das U-Boot") [DB/OL]. <http://www.denx.de/wiki/U-Bootdoc/Presentation>, 2012-08-15.

# 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](#)

## 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](#)

## 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](#)

## 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](#)

## 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](#)

## 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](#)

## 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](#)

## 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](#)