

## 基于 ARM11 的嵌入式 Linux NAND FLASH 模拟 U 盘挂载分析与实现

彭玉峰, 赵越, 张淑丽, 金龙, 林思宏  
(河南师范大学 物理与信息工程学院, 河南 新乡 453007)

**摘要:**介绍了嵌入式 Linux 操作系统下基于三星微处理器 S3C6410 的 NAND FLASH 模拟 U 盘的原理与实现方法。操作系统采用 Linux 2.6.28 版本,平台为飞凌 OK6410-A 开发板。采用的方案是通过添加一个 512 MB 的 NAND FLASH 分区空间,配置 Linux 系统 USB Gadget 功能,实现划分出的 512 MB 的 NAND FLASH 空间以 U 盘存储系统与 PC 机通信。该方案的文献目前在国内外同等操作系统版本和平台上并无先例。通过上述方案成功地实现了 S3C6410 开发板划分出 512 MB 的 NAND FLASH 空间以 U 盘形式挂载到了 PC 机上,实现与 PC 机的信息交换。

**关键词:**ARM11; 模拟 U 盘; Linux; S3C6410

### Analysis and implementation of ARM11-based mount for NAND Flash simulation U-disk under embedded Linux

PENG Yu-feng, ZHAO Yue, ZHANG Shu-li, JIN Long, LIN Si-Hong  
(College of Physics and Information Engineering, Henan Normal University, Xinxiang 453007, China)

**Abstract:** The principle and implementation method of the NAND Flash simulation U-disk based on SAMSUNG micro-processor s3c6410 under the embedded Linux is introduced. The version Linux 2.6.28 is taken as the operating system, and OK6410 of Fei-Ling company as the platform. The scheme is to add a NAND Flash divided space of 512 MB and to configure the USB Gadget function of Linux system to achieve NAND Flash space of the divided 512 MB for communicating with PC through U-disk memory system. Up to now, this scheme has not appeared in the equal version of operating system and the same platform in the world. The scheme successfully accomplished the mount of NAND Flash space of the divided 512 MB as U-disk type with PC to realize the information exchange with PC.

**Keywords:** ARM11; simulation U-disk; Linux; S3C6410

## 0 引言

现阶段嵌入式产品作为 U 盘挂载到 PC 机上在各类电子产品中被越来越多的应用, Linux 操作系统在电子产品中的应用也越来越广泛,但是 Linux 中模拟 U 盘挂载到 PC 机中,与 PC 机上通用 Windows 的通信还未很好的达到实现。本文基于 ARM11 体系结构的处理器,实现了 Linux 系统下 NAND FLASH 模拟 U 盘挂载到 PC 机上与 Windows 进行通信。并在飞凌的 OK6410-A 板上得到验证,实现了该功能。

目前国内嵌入式产品多用 ARM9<sup>[1]</sup> 体系结构,

ARM11 则用于高端电子产品, NAND FLASH 模拟 U 盘的挂载技术虽然在电子产品中广泛应用,但是对于新起的 Linux 操作系统的高端电子产品,这项技术还鲜少以文献的形式得以系统阐述,国内长春大学张伟等人在这方面做过研究,但是他们是基于 ARM9 体系,本文基于 ARM9<sup>[2]</sup> 在 ARM11 体系结构的处理器上做了新的研究和应用。

现阶段有 3 类方式可以实现模拟 U 盘的功能,虚拟 vfat.ing 模拟 U 盘挂载, SD 卡模拟 U 盘挂载, NAND FLASH 模拟 U 盘挂载。NAND FLASH 由于其大容量和高性价比,往往更加流行。本文是通过 NAND FLASH 实现模拟 U 盘的挂载的。

## 1 NAND FLASH 模拟 U 盘原理分析

嵌入式产品模拟 U 盘与 PC 通信是现代嵌入式产品所广泛应用的一种技术,因此在实现 Linux 下 NAND FLASH 模拟 U 盘之前先要对 Linux 下 NAND

FLASH 模拟 U 盘的原理进行分析。Linux 下对模拟 U 盘存储设备的支持就是 Linux-USB Gadget 驱动的加载,该驱动框架实现了 USB 协议定义的设备端的软件功能。

## 1.1 Linux-Gadget 框架

Linux 系统中 Gadget 驱动层序框架分为 3 层: Gadget 驱动层;Gadget API 层;UDC 层。Gadget 驱动层实现实现不同的 USB 功能,例如“USB 网卡”、“U 盘”、“打印机”等。Gadget API 层,为 Gadget 提供的一套标准 API,由底层 USB 设备控制器(USB Device Controller, UDC)实现这套 API,Gadget 驱动通过调用这些 API 实现与 UDC 通信。UDC 是片上系统(SoC)的一部分,不同的 UDC 需要不同的驱动,同样的 UDC 不同板子代码也不一样,这一层为平台相关层,直接访问访问硬件,并向上层提供与硬件相关操作的回调函数。Gadget 驱动框架原理图如图 1 所示。

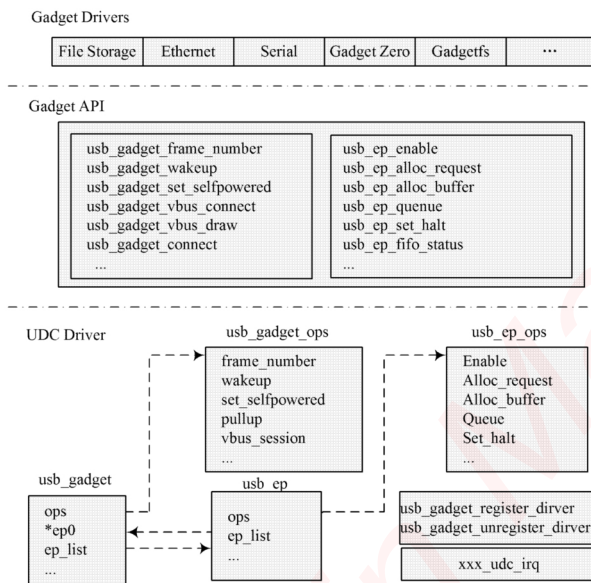


图 1 Gadget 驱动框架原理图

UDC 层中 `usb_gadget` 和 `usb_ep` 为 2 个描述结构体,分别描述 UDC 和端点,Gadget API 通过这两个结构实现对 UDC 的控制。`usb_gadget_ops` 和 `usb_ep_ops` 为函数集结构体,其主要功能是实现与硬件的交互并实现返回给上层 Gadget API 所需信息的功能。`usb_gadget_register_driver` 和 `usb_gadget_unregister_driver` 为 Gadget 框架提供的标准 API,是驱动层序的注册函数,由 UDC 层提供,之所以不是通过 Gadget API 层实现的原因是,用于将 Gadget 驱动程序绑定到 UDC。`xxx_udc_irq` 是 UDC 设备的中断处理函数,处理设备及其端点的中断请求。

Gadget API 层为 Gadget 定义的一组标准 API 接口函数,并向上提供编程接口,这一层的存在实现了将

下层 UDC 驱动和上层 Gadget 驱动程序的隔离,使得 Linux 系统中编写 USB 设备侧驱动程序时能够把功能的实现和底层通信分离。

Gadget 驱动程序使用 `usb_gadget_driver` 结构体描述,实现将 Gadget 驱动与下层设备控制器相关联,并开启设备功能。

## 1.2 Linux NAND FLASH 分区原理

NAND FLASH 作为存储设备,存储着嵌入式产品从开机到结束的所有代码,所以做好存储设备的分区是至关重要的问题,LINUX 操作系统下,分区如图 2 所示。

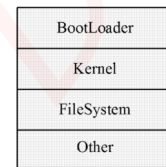


图 2 LINUX 操作系统软件分区

Linux2.6 内核中,在文件 `arch/arm/plat-s3c/include/plat/Partition.h` 中定义了分区信息如下:

```
struct mtd_partition s3c_partition_info[] = {
    {
        .name = "Bootloader,"
        .offset = 0,
        .size = (1 * SZ_1M),
        // .mask_flags = MTD_CAP_NANDFLASH,
    },
    {
        .name = "Kernel,"
        .offset = MTDPART_OFS_APPEND,
        .size = (8 * SZ_1M),
    },
    ...
};
```

其中“. name”为分区的名称,“. offset”为分区起始地址,“. size”为分区大小。这个结构体包涵了 NAND FLASH 的所有分区信息。

## 2 NAND FLASH 模拟 U 盘实践方法

### 2.1 修改内核

由于 `struct mtd_partition s3c_partition_info[]` 定义了分区信息,所以首先要修改或者添加分区信息,使得模拟得到的 U 盘空间大小符合实际需求。

在 `arch/arm/plat-s3c/include/plat/Partition.h` 的 `struct mtd_partition s3c_partition_info[]` 中添加分区信息如下:

```
{
    .name = "U_Storage,"
    .offset = MTDPART_OFS_APPEND,
```

```
.size = 512 * (SZ_1MS),  
}
```

其中: . name 为分区名称, 定义为 U\_Storage, . offset 为起始地址, 这个值需要结合该模块前面的模块的大小和起始地址算出; . size 为定义的分区大小, 为 512 MB。

添加分区后重新编译, 可在文件系统的/etc 下找到相应盘符, 如: mtdblock4。

## 2.2 配置内核

内核中实现 NAND FLASH 模拟 U 盘挂载的驱动是 USB 中的 Gadget 实现的, 所以在实现这项功能时, 首先在编译内核的时候, 配置上 Gadget 功能。具体操作如下:

在内核源代码的根目录下执行 # make menuconfig, 这时将会看到内核的菜单选项, 选择:

```
device driver->  
[*]usb support->  
<*>usb gadget support  
<*>usb peripheral controller
```

将 usb support 功能选中, 选择编译该功能进内核, 并在 usb support 功能子目录中选择中 usb gadget support 和 usb peripheral controller 功能, 选择编译进内核, 做完这些操作后编译内核。

在 shell 窗口命令行中输入:

```
# make zImage  
# make modules
```

在 driver/usb/gadget 目录下会生成 g\_file\_storage.ko 和 s3c\_udc\_otg.ko。讲这两个模块复制到目标板的文件系统中。

注意: 当 usb gadget support 功能非作为模块编译进内核时, 编译过模块后需要更新内核。而当 usb gadget support 功能作为模块编译进内核时, (选择栏应填“<M>usb gadget support”, 其中“M”代表模块), 在编译生成模块后, 除了要加载 g\_file\_storage.ko 还要加载 s3c\_udc\_otg.ko。

## 2.3 加载模块

在目标板的 shell 命令窗口中执行:

```
# insmod s3c_udc_otg.ko  
# insmod g_file_storage.ko file=/dev/mtdblock4 stall  
=0 removable=1
```

这个时候将目标板连接到 PC 机即可看到被模拟的 U 盘盘符, 接着就可以像操作正常 U 盘一样对该 U 盘进行操作。

注意: 这个时候模拟 U 盘已经存在, 但是在目标板中还看不到该 U 盘盘符, 需要执行:

```
# mkdir /mnt/U_Storage  
# mount -t vfat /dev/mtdblock4/U_Storage
```

然后, 才能在目标板的/mnt/U\_Storage 下操作这个 U 盘。

## 3 总结与展望

到此已经阐述并实现了模拟 U 盘的实现原理, 其实践结果如图 3 所示。

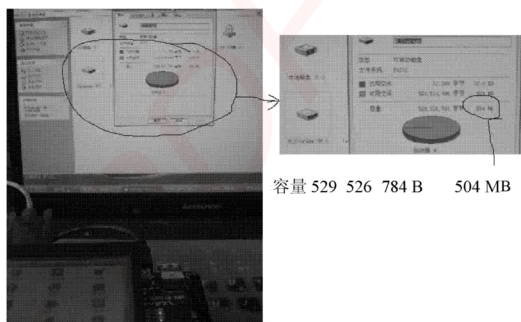


图 3 模拟 U 盘挂载属性图

从图中看出属性中 U\_Storage 的容量为 504 MB 并不是 512 MB, 这是因为厂家生产存储介质时, 容量是 10 进制, 比如 1 GB=1 000 MB, 但是电脑却是采用二进制, 1 GB=2<sup>10</sup> MB, 1 MB=2<sup>10</sup> KB, 这样计算, 电脑显示容量大小要比标注的容量小些。在 PC 机上在 U\_Storage 中新建一个名为“zy”的文件夹, 然后再在开发板上新建一个名为“haha”的文件夹, 同时在 Windows 下和板子中查看, 会出现如图 4 所示信息。

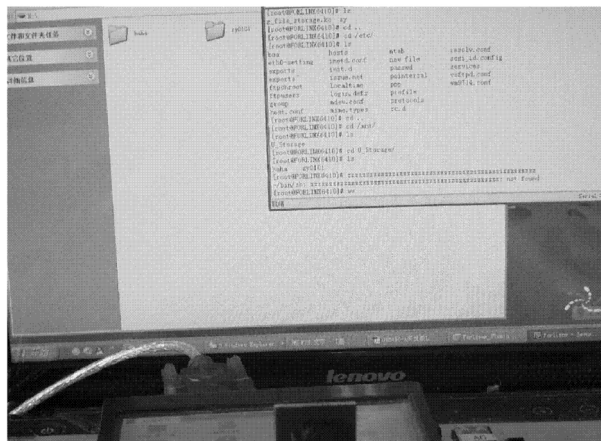


图 4 模拟 U 盘与 PC 交换信息

图 4 中 H 盘为 PC 机给 U\_Storage 分配的盘符, 上层界面为 PC 串口与开发板交互的打印信息该信息显示的是开发板的内容。从图中可知, H 盘中存在“haha”和“zy”两个目录; 开发板与 PC 交换界面中跳到 /mnt/U\_Storage/目录下, 用“ls”命令查得, 开发板交换

(下转第 7 页)

低电平的占空比为 31/32。特意留出的 1/32 脉冲用来对电池进行阻抗测试,以确保装到充电器里的不是碱性电池。当检测到电池电压出现 2 mV 的电压降时,快冲持续。如果没有检测到电压降,快冲就会持续,直到快冲定时器计时完毕或者检测到过热、过压等故障条件(包括阻抗故障)。当快充因为检测到电池电压降或者快冲定时器计时完毕而结束后,DS2712 进入定时充满模式。定时充满模式以 1/8 的占空比进行充电,持续时间为程序设定的快冲定时器计时长度的二分之一。定时充满完成后,充电器进入维持模式,此时以 1/64 的占空比进行涓流充电,充电器会一直停留在维持模式直到电池被取走。

图 5 所示的充电器由大功率 USB 接口供电,对一节容量为 2 100 mA·h 的镍氢电池进行充电,完成快充需要 2 h 多一点,而完全充满电池则需要大约 3 h。

## 5 结 语

本文设计的 USB 接口镍氢电池充电器经硬件验证,实现了对一节镍氢电池的快速智能充电功能,充电过程稳定可靠。本文设计的充电器进一步还可以改进为对多节镍氢电池同时充电。USB 接口作为一种经济实用的电源,可以为大多数小型消费类电子产品进行电

池充电。为了与 USB 2.0 接口标准完全兼容,连接在 USB 接口的负载必须能跟主机进行双向通信。负载还要具有包括低功耗模式在内的电源管理功能,并能由主机决定大功率输出的条件。对于大多数主机而言,部分兼容的负载系统也可以与 USB 接口相互配合、正常工作,但有时会出现意想不到的情况。这就需要对 USB 接口特性和负载特性都具有深入的理解,从而权衡接口兼容性和负载复杂性。

## 参 考 文 献

- [1] 张葵葵. 镍氢电池智能充电控制系统设计[D]. 长沙:国防科学技术大学,2004.
- [2] 杨幼明,胡珊玲. 镍氢电池性能的影响因素试验[J]. 电源技术,2000,5(6):36-39.
- [3] 杨忠敏. 车用镍氢电池及其应用[J]. 汽车电器,2004(12):1-3.
- [4] 赵同贺. 开关电源设计技术与应用实例[M]. 北京:人民邮电出版社,2007.
- [5] 王志强. 开关电源设计[M]. 北京:电子工业出版社,2006.
- [6] 张占松,蔡宣三. 开关电源原理与设计[M]. 北京:电子工业出版社,2000.
- [7] 潘天红,赵德安,刘星桥. 电动自行车智能充电器的研制[J]. 仪表技术与传感器,2004(8):55-57.

作者简介: 屈宝鹏 男,陕西佳县人,讲师。主要从事半导体材料及集成电路方面的科研与教学工作。

张喜凤 女,内蒙古萨拉齐人,硕士,讲师。现主要从事半导体材料及集成电路方面的科研与教学工作。

许 燕 女,河北邢台人,硕士,助教。现主要从事半导体材料及集成电路方面的科研与教学工作。

(上接第 3 页)

目录中存在“haha”和“zy”两个目录,证明 U 盘和 PC 交互信息成功。NAND FLASH 模拟 U 盘挂载,使得目标板充分利用了 U 盘的便捷功能,使得携带、操作、移植更为方便。Linux 下基于 ARM11 体系结构模拟 U 盘的实现使得高端电子产品模拟 U 盘技术得到广泛应用。经实验验证该设计是成功的,成功解决了 ARM11 体系结构下目标板作为 U 盘挂载到 PC 机的问题,具有实际意义。

## 参 考 文 献

- [1] 缙新科,滕永. 基于 ARM9 的嵌入式 Linux 系统分析与移植[J]. 甘肃科学学报,2011,23(12):106-109.

作者简介: 彭玉峰 男,1963 年出生,河南夏邑人,博士,教授。主要从事通信技术、嵌入式技术与集成电路等方面的研究工作。

赵 越 女,1987 年出生,河南新乡人,硕士研究生。研究方向为嵌入式系统与通信技术。

- [2] ZHANG Wei. The embedded data acquisition storage systems based on S3C2410 [C]// 2009 4th IEEE Conference on Industrial Electronics and Applications. Xi'an, China; ICIEA, 2009: 1812-1814.
- [3] 曾强. 基于 FT245 的 Linux USB 驱动的设计[J]. 光电技术应用,2011,26(2):69-73.
- [4] RUBINI Alessandro. Linux device drivers [M]. 2nd ed. USA: O' Reilly, 2001: 101-160.
- [5] 宋宝华. Linux 设备驱动开发详解[M]. 2 版. 北京:人民邮电出版社,2010.
- [6] PETERSEN Richard. Linux and Unix shell pocket programmer [M]. [S. l.]: Surfing Turtle Press, 2008.
- [7] 毛玲,阎学文. 嵌入式 Linux 下用 AT91RM9200 实现对 U 盘的操作[J]. 电子科技,2008(11):35-38.

# 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](#)

## 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](#)

## 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](#)
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](#)

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](#)

## 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](#)

## 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](#)

## 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](#)

## 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](#)