

基于 EBD9200的 VxWorks BSP设计和实现

李霞, 刘宏立

(湖南大学 电气与信息工程学院, 湖南 长沙 410082)

[摘要] 操作系统的移植是嵌入式开发的基础。介绍了嵌入式实时操作系统 VxWorks的 BSP概念,分析了 VxWorks的启动顺序和系统存储布局,然后简述了英贝德 EBD9200开发板的组成和硬件结构,在此基础上给出了 EBD9200开发板的 BSP设计和实现步骤,包括软硬件初始化和驱动程序设计以及引导模块和工程映像的下载。最后阐述了在调试 BootROM时采用的小技巧,为方便操作系统的移植提供了参考。

[关键词] VxWorks; 移植; 板级支持包 (BSP)

[中图分类号] TP316.2

[文献标识码] A

VxWorks是美国 WindRiver公司专门为实时嵌入式系统设计开发的操作系统软件,为程序员提供了高效的实时任务调度、中断管理、实时的系统资源以及实时的任务间通信。VxWorks采用模块化设计,把所有特定硬件功能都集成在板级支持包 (Board Support Package, BSP)的库中,为所有硬件功能板提供相同的软件界面。EBD9200是深圳英贝特公司的一款基于 AT91RM9200处理器,ARM920T内核的评估板。本文主要研究如何将 VxWorks操作系统移植到 EBD9200硬件平台上,即 BSP的设计和实现。

1 BSP概念

BSP通常是针对具体的硬件平台,用户所编写的启动代码和部分设备驱动程序的集合,其目的是为 VxWorks提供硬件环境的接口。

通电后,BSP负责初始化硬件,支持 VxWorks和硬件驱动通信,使硬件相关 (Hardware-dependence)和独立于硬件 (Hardware-independence)的部分在 VxWorks系统中很好地结合。

2 系统启动顺序

2.1 VxWorks Image

VxWorks映像分为在 ROM中运行和在 RAM中运行,两者的区别在于 sysInit()函数的调用,在 RAM运行的 VxWorks中该函数初始化 RAM。两种类型的启动顺序如图 1所示。

2.1.1 ROM中运行的 VxWorks

VxWorks在 ROM中运行,即写入 ROM的 VxWorks是非压缩的,系统直接跳到 ROM的首地址运行 VxWorks。VxWorks在 ROM中运行主要是为了节省 RAM空间,以便应用程序有更大的空间运行。只把 VxWorks Image的 data段复制到 RAM的 LOCAL_LOW_ADRS, text段留在 ROM并在 ROM中执行。

2.1.2 RAM中运行的 VxWorks

VxWorks在 RAM中运行,即写入 ROM的 Bootrom或 VxWorks Image是压缩的,需要先解压复制所有的 text和 data到 RAM的 LOCAL_LOW_ADRS中。sysInit()主要是初始化 RAM用的;在 usrInit()之前不能压缩,即 romInit()和 romStart()不能压缩。

2.2 BootROM Image

BootROM Image是最少的系统初始化,主要用于启动装载 VxWorks Image. 与 VxWorks Image的区别是 BootROM 调用 bootConfig.c,而 VxWorks调用 usrConfig.c.

2.3 存储布局

对于所有 ARM处理器, VxWorks存储布局是一样的,如图 2 示。图 2 展示的地址是相对于它的起始内存而言的,在包含 INCLUDE_MEMORY_CONFIG配置下,内存起始地址宏定义为 LOCAL_MEM_LOCAL_ADRS 初期的堆栈和系统映像地址是在 BSP的 config.h里面配置的。

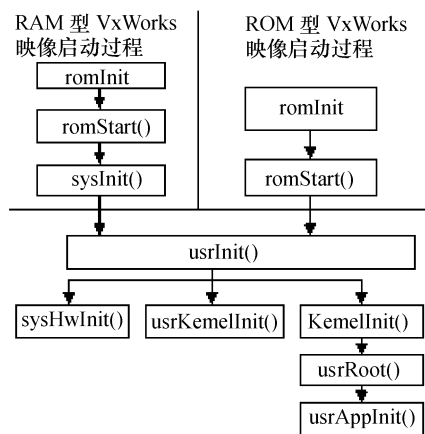


图 1 VxWorks映像启动类型

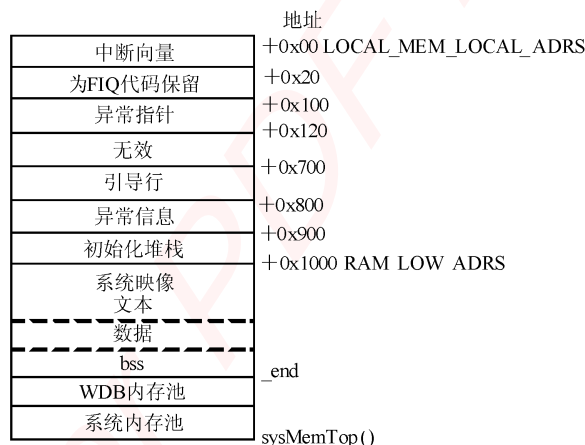


图 2 VxWorks系统存储布局

3 VxWorks在 EBD9200上的 BSP设计和实现

3.1 EBD9200开发板硬件综述

开发板采用 AT91RM9200处理器,AT91RM9200处理器是 ARM920T内核的 16/32位 RISC CPU. 其外围电路包含了 SDRAM、FLASH、存储卡接口、USB 接口、LCD 接口和网络接口。系统控制软件保存于 FLASH 中,运算速度可达 180 MIPS,已经达到低端 DSP的水平。

AT91RM9200微处理器在高性能和低功耗特性方面提供最佳性能。它具有 5级整数流水线,提供 1.1 MIPS/MHz的哈佛结构,支持数据 Cache和指令 Cache,全性能的 MMU支持 VxWorks等多种主流嵌入式操作系统,最小功能如图 3 所示。

3.2 BSP设计

BSP模板的选择是由 CPU 确定的, target/config/integrator920t是专门为基于 ARM920T内核的处理器设计的 BSP模板,我们就是在这个模板的基础上进行 BSP的设计,包括软硬件的初始化和主要设备驱动程序。

3.2.1 主要的软硬件初始化

(1) Flash: 我们采用的是 16MB 的 E28F128J3150 Nor Flash,使用片选 CS0,所以对 EB_LCSA 寄存器中 CS0A 位置 1,并根据 Flash 的位宽、访问时序等信息配置相应的 SMC_CSR0寄存器。以上配置是在 romInit.s 中完成。另外在 config.h 中配置宏定义 ROM_BASE_ADRS (0x1000000), ROM_SIZE (0x0100000)等,这些宏定义要与 Mikefile 中的定义一致。

(2) SDRAM: 我们采用 2片 HY57V281620HCT - 6组成 32M 的 SDRAM,所以对 EB_LCSA 寄存器中 CS1A 位置 1,并根据 SDRAM 的访问时序、行列地址和刷新频率等信息配置 SDRAMC 的寄存器,这些配置是在 romInit.s 中完成。另外在 config.h 中配置宏定义 RAM_LOW_ADRS (0x20001000), RAM_HIGH_ADRS (0x21800000), LOCAL_MEM_SIZE (0x01a00000), LOCAL_MEM_LOCAL_ADRS (0x20000000)等。其中, RAM_LOW_ADRS 和 RAM_HIGH_ADRS 的定义要与 Mikefile 中的定义一致。

(3) CLOCK: 开发板主晶振为 18.432MHz,为了达到 180MHz 的 CPU 主时钟频率,需要设置 CKGR_PLLAR 寄存器将主晶振时钟先除 4 再倍 39 倍频以得到 179.712MHz 的主时钟频率。另外,为了向 SDRAM 和 UART 等外设提供时钟,需要设置 PMC_MCKR 寄存器,将主时钟频率除 3 得到 59.904MHz 的外设时钟频率。这些配置在 romInit.s 中完成。

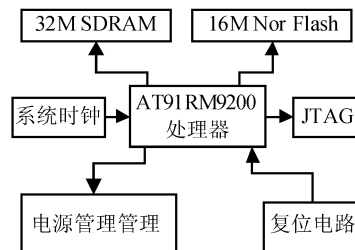


图 3 EBD9200开发板最小功能图

(4) AIC (Advance Interrupt Controller): EBD9200提供 32个独立的中断源,因为 VxWorks不提供 FIQ,故保留中断源 0。在 VxWorks系统中中断号与中断向量是通过 NUM_TO_MEC (inNum)转换的,中断服务程序的安装则是由 inConnect()函数实现。在内核启动之前要关闭所有的外设的中断并且不能安装中断服务程序。

(5) ST(System Timer)和 TC(Timer Controller): VxWorks要求提供两个独立的时钟用于系统时钟和辅助时钟,系统时钟用于实现 VxWorks系统中的时间片,辅助时钟用于提供高速优先级的周期性中断。在 AT91RM9200中提供系统时钟 ST和两套时钟控制器 TC,我们采用 ST(VxWorks系统时钟)和第零套时钟控制器的 TC0(辅助时钟)。

在 sysHwInit()函数中执行 ST和 TC0的硬件初始化,并在 sysHwInit2()函数中调用 inConnect()函数连接时钟中断。

(6) DBGU和 UART:前者只能工作在 UART模式,后者还可以工作在 IrDA, ISO7816, RS485和 Modem模式。EBD9200提供四套 UART控制器,我们使用 UART1。对 DBGU和 UART1的配置参数可通过相应寄存器进行设置,这些参数包括:PDA的外设 A与外设 B的配置、PDB的外设 A与外设 B的配置、波特率、数据位数、停止位数、奇偶校验。

(7) EMAC(Ethernet MAC): EMAC控制器实现了物理层与逻辑层之间的 OSI参考模型 MAC子层的物理功能,它能在 MII(Media-Independence Interface)和 RMII(Reduced Media-Independence Interface)两种模式下发送数据,模式的选择通过配置 ETH_CFG寄存器相应位来实现。

3.2.2 主要的设备驱动程序设计

(1) 串行口驱动 at91uart.c:

数据结构如下:

```
LOCAL SD_DRV_FUNCS AT91UartDrvFuncs =
```

```
{
    AT91UartBctl,
    AT91UartTxStartup,
    AT91UartCallbackInstall,
    AT91UartPollInput,
    AT91UartPollOutput,
};
```

AT91UartBctl函数主要用作设备控制,主要包括波特率的读/写和硬件属性的设置,比如奇偶校验、数据长度、硬件流控等。

AT91UartTxStartup函数用于启动发送器,若采用中断方式发送,该函数负责使能中断并调用中断发送字符。

(2) 网络驱动 at91emacend.c:

AT91RM9200自带有以太网媒体访问控制器(EMAC),物理层接口采用的是 DM9161E,根据 PHY自动协商结果进行工作模式配置、以太网报文的发送和接收、中断的使能和禁止、组播地址的添加和删除等功能。

网络驱动主要为上层应用提供系统接口函数和串行口中断处理子程序。

系统接口函数结构如下:

```
LOCAL NET_FUNCS at91endFuncTable =
```

```
{
    (STATUS(*) (END_OBJ*)) at91EndStart,
    (STATUS(*) (END_OBJ*)) at91EndStop,
    (STATUS(*) (END_OBJ*)) at91EndUnload,
    (int(*) (END_OBJ*, int, caddr_t)) at91EndBctl,
    (STATUS(*) (END_OBJ*, M_BLK_D)) at91EndSend,
    (STATUS(*) (END_OBJ*, char*)) at91EndMCastAdd,
    (STATUS(*) (END_OBJ*, char*)) at91EndMCastDel,
    (STATUS(*) (END_OBJ*, MULTI_TABLE*)) at91EndMCastGet,
    (STATUS(*) (END_OBJ*, M_BLK_D)) at91EndPollSend,
    (STATUS(*) (END_OBJ*, M_BLK_D)) at91EndPollRcv,
    endEtherAddressForm,
    endEtherPacketDataGet,
    endEtherPacketAddrGet
};
```

3.3 BSP的实现

3.3.1 引导模块及工程映象下载

引导模块完成系统软硬件初始化并加载 VxWorks版本,系统上电时系统自动跳转到 0x0处执行,引导模块配置时钟,SDRAM控制器等硬件环境后,将 BootROM 内容拷贝到 SDRAM 的 RAM_LOW_ADRS (0x20001000),并跳转到 SDRAM 中继续进行软硬件初始化,为 VxWorks 下载做准备。在加载过程中,若发生错误能重新引导。

工程映象下载是指在 BootROM 完成系统初始化以后,从网络或者 Flash 上下载一个更大的映象,可以选择从以太网和 Flash 下载工程映象,我们选择网络下载,BootROM 启动 Ftp Client,以指定的用户和口令(由 config.h 里 DEFAULT_BOOTLINE 定义),登录到 Tomado 自带的 Ftp Server 查找相应的 VxWorks 映象,并通过 Ftp 将工程映象文件复制到 RAM_HIGH_ADRS (0x21800000) 开始的内存空间,下载完毕后程序自动跳转到 0x21800000 并开始执行 VxWorks 映象。

3.3.2 调试技巧

对 Bootrom 进行调试时,若是基于 Flash 进行调试,步骤如下:从内部启动 ROM 启动,通过串口的 xmodem 协议下载 Loader bin 至 SRAM 中运行,Loader bin 通过串口的 kexmit 协议下载需要烧写的 Bootrom 至内存并烧写到 Flash。这个过程不仅复杂而且通过串口下载程序速度较慢,因此我们采用内存调试模式:在 config.h 和 Makefile 中将 ROM_BASE_ADRS 的地址修改到 SDRAM 空间;在 rom Init.s 中通过宏定义 RAM_SM 屏蔽 FLASH, SDRAM, CLOCK 初始化部分;在下载 Bootrom 到 SDRAM 之前,通过仿真器的脚本对 AT91RM9200 的主时钟、外部时钟、EB 控制器、SMC 控制器、SDRAM 控制器进行配置。

4 结 语

VxWorks 具有很好的可靠性、实时性和可裁减性,适合工业控制、通信等对实时性要求很高的领域。将 VxWorks 成功移植到基于工业级 ARM9 处理器 AT91RM9200 的 EBD9200 开发板上,将为以后应用程序的开发提供强大的系统软件支撑;同时,在熟悉 ARM 体系结构和 VxWorks BSP 设计流程的基础上,我们可以将这款优秀的实时操作系统移植到任何一种 ARM 处理器上。

[参 考 文 献]

- [1] 周启平. VxWorks 下设备驱动及 BSP 开发指南 [M]. 北京:中国电力出版社,2004. 37—38
- [2] [美] Wind River. VxWorks BSP User Guide [M]. Tomado 2.2, 2002
- [3] 罗国庆. VxWorks 与嵌入式软件开发 [M]. 北京:机械工业出版社,2003. 229—240
- [4] 张忠,樊留群. VxWorks 在 S3C2410 上的 BSP 设计 [J]. 微型电脑应用,2005,21(10): 16—19.
- [5] ATMAL. Data sheet of ARM920T-based Microcontroller AT91RM9200 [EB/OL]. <http://www.atmel.com>, 2006

Design and realization of VxWorks based on EBD9200

LIXia, LU Hong-li

(School of Electrical and Information Engineering, Hunan University, Changsha 410082, China)

Abstract: Porting operation system is the basis of embedded-system's development. The conception of the embedded real-time operating system VxWorks BSP (Board Support Package), the VxWorks start-up flow and the memory layout is introduced firstly. Then the hardware members of the board EBD9200, and the design and step to realize the BSP for EBD9200 is presented, including the initialization of hardware and software, the design of several important drivers, and the download of the bootrom and the project image. Lastly, a tip is recommended to debug the bootrom and to be referred by operation system's porting.

Key words: VxWorks; Porting; Board Support Package (BSP)

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.