

μC/OS - II 在 LPC2210 上的移植研究

尚宇, 鄧琦

(西安工业大学, 陕西 西安 710032)

摘要: 嵌入式系统已在各个领域得到广泛应用。在嵌入式应用中, 只有把 CPU 嵌入到系统中, 同时又把操作系统嵌入进去, 才是真正意义上的嵌入式应用^[1]。嵌入式实时操作系统目前应用越来越广泛重要。因此嵌入式实时操作系统在目标处理器平台上的移植就成为嵌入式软件开发的基础。文中研究了嵌入式实时操作系统 μC/OS - II, 并将其成功移植到 PHILIPS 公司的 ARM 微控制器 LPC2210 上, 实现了嵌入式开发平台的移植及实现。

关键词: ARM; 嵌入式; μC/OS - II; 实时操作系统; 移植; LPC2210

中图分类号: TP316.2

文献标识码: A

文章编号: 1673 - 629X(2007)02 - 0103 - 03

Research of Porting RTOS μC/OS - II to LPC2210

SHANG Yu, ZHI Qi

(Xi'an Technological University, Xi'an 710032, China)

Abstract: Embedded system has been widely applied in various fields. In an embedded application, only the CPU embedded, while also operating system embedded is a real sense of an embedded application. An embedded real-time operating system is increasingly widespread application. Therefore which in the target processor platform transplant has become very important. It shows μC/OS - II and its successful transplant to Philips company ARM LPC2210 and achieves this platforms successful transplant and realized.

Key words: ARM; embedded system; μC/OS - II; RTOS; porting; LPC2210

1 实时操作系统 μC/OS - II

μC/OS - II 是一个源码公开、可移植、可固化、可裁剪及占先式的实时多任务操作系统^[1], 它运行优先级最高的就绪任务, 其绝大部分源码是用 C 语言编写, 小部分用汇编语言代码已被压缩到最低限度, 使之可供不同构架的微处理器使用。它最多可管理 64 个任务, 用户可使用 56 个任务。至今, 其已在超过 40 种不同构架的微处理器上运行^[2]。

μC/OS - II 的软件体系结构^[1]如图 1 所示。应用程序软件是用户根据需求编写与应用相关代码定制合适的内核服务功能, 实现对 μC/OS - II 的裁剪; 与处理器无关代码就是操作系统的内核, μC/OS - II 内核提供所有的系统服务, 这部分代码完全公开, 文中采用 μC/OS - II v2.52 版本, 内核将应用程序与底层硬件有机结合成一个实时系统; 与处理器相关代码可看作是内核与硬件之间的中间层, 它实现同一内核于不同硬件体系的目标, 处理器不同, 这部分代码也不同, 由用户自行编写。所谓移植就是编码使一个实时内核能

在其他微处理器或微控制器上运行即编写与处理器相关代码和进行一些 μC/OS - II 的设置。

2 μC/OS - II 在 LPC2210 上的移植

LPC2210 是 PHILIPS 公司开发的基于支持实时仿真和嵌入式跟踪的 16/32 位 ARM7TDMI CPU 微控制器^[3]。具有标准 32 位 ARM 指令集和 16 位 THUMB 指令集, 16k 静态 RAM, 144 管脚封装, 片内 BOOT 装载程序实现 ISP 和 IAP, 9 个外部中断, 向量中断控制器可配置优先级和向量地址等, 具有低成本、低功耗、高性能等优点^[3]。

开发工具选用 ARM 公司提供的 ADS1.2 集成开发环境。移植工作主要集中在与处理器相关的 3 个文件^[4]: OS_CPU.H, OS_CPU.C, OS_CPU.A.S。其中 OS_CPU.H 文件主要包含与编译器相关的数据类型定义、堆栈类型定义、两个宏定义和几个函数说明。OS_CPU.C 包含与移植有关 6 个 C 函数, 为 OSTaskStkInit(), OSTaskCreateHook(), OSTaskDelHook(), OSTaskSwHook(), OSTaskStatHook(), OSTimeTickHook()。除第一个任务堆栈初始化函数外的 5 个函数为钩子函数, 需要声明, 没有实际内容, 由系统函

数调用,以使用户能在操作系统中加入自己需要的功能。OS_CPU_A.S 中包含与移植有关的 4 个汇编语言函数,为 OSStartHighRdy(), OSCtxSw(), OSIntCtxSw(), OSTickISR()。

2.1 OS_CPU.H 文件移植

μC/OS-II 不使用 C 语言中 short, int 和 long 等数据类型定义,因为它们与处理器类型相关,隐含不可移植性,因此必须对其进行移植。H 文件中主要包含与编译器相关数据类型定义、堆栈类型定义、两个宏定义和几个函数说明。数据类型定义代码如下:

```
typedef unsigned char BOOLEAN; /* 布尔变量 */
typedef unsigned char INT8U; /* 无符号 8 位整型变量 */
typedef signed char INT8S; /* 有符号 8 位整型变量 */
```

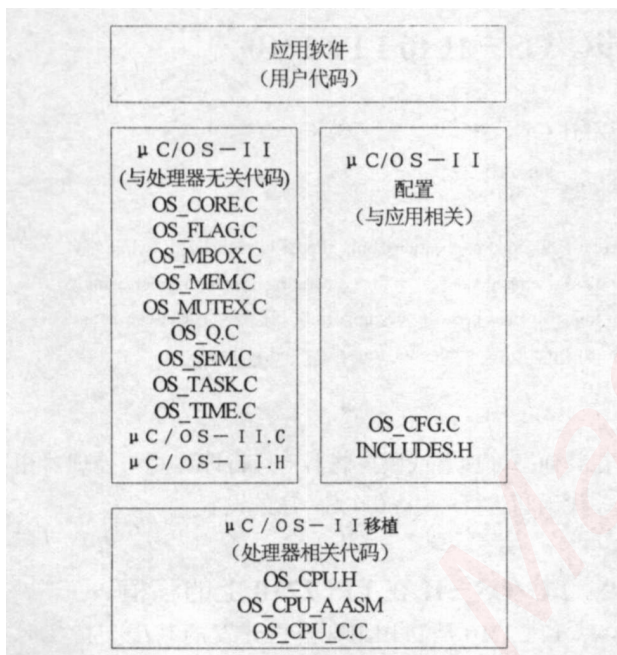


图 1 μC/OS-II 软件体系结构

```
typedef unsigned short INT16U; /* 无符号 16 位整型变量 */
typedef signed short INT16S; /* 有符号 16 位整型变量 */
typedef unsigned int INT32U; /* 无符号 32 位整型变量 */
typedef signed int INT32S; /* 有符号 32 位整型变量 */
typedef float FP32; /* 单精度浮点数(32 位长度) */
typedef double FP64; /* 双精度浮点数(64 位长度) */
typedef INT32U OS_STK; /* 堆栈是 32 位宽度 */
```

另外,带 T 变量的 ARM7 处理器内核有两个指令集,用户任务可使用两种处理器模式即用户模式和系统模式,本移植使用软中断指令 SWI 作为底层接口,使用不同功能号区分不同函数。在 ADS 中,用关键字 .swi 声明一个不存在的函数,则调用该函数就在调用的地方插入一条 SWI 指令并可指定功能号。SWI 函数部分代码如下:

```
swi(0x00) void OS_TASK_SW(void); /* 任务级任务切换函数 */
swi(0x01) void OSStartHighRdy(void); /* 运行优先级最高的任务 */
swi(0x02) void OS_ENTER_CRITICAL(void); /* 关中断 */
swi(0x03) void OS_EXIT_CRITICAL(void); /* 开中断 */
swi(0x42) void OSISRBegin(void); /* 中断开始处理 */
swi(0x43) int OSISRNeedSwap(void); /* 判断中断是否需要切换 */
swi(0x80) void ChangeToSYSMode(void); /* 任务切换到系统模式 */
swi(0x81) void ChangeToUSRMode(void); /* 任务切换到用户模式 */
```

由于 ADSv1.2 的 C 语言编译器只支持满递减堆栈,所以只能定义堆栈增长方向从上向下,即 #define OS_STK_GROWTH 1。

2.2 OS_CPU.C.C 文件移植

每个任务要有自己的堆栈空间,以便进行任务切换时能将当时处理器现场保存到任务堆栈空间中,在下次执行时再恢复出来。因此必先确定任务堆栈结构,而任务堆栈结构与 CPU 体系结构、编译器密切相关。在 ARM 体系下,处理器现场通常指{PC, LR, R0 - R12, CPSR, SPSR}。本移植堆栈结构如图 2 所示,任务堆栈由高到低依次将保存 PC, LR, R12, R11, R10 .. R1, R0, CPSR 等。每个任务有独立的 OSEnterSum, 在任务切换时保存和恢复各自 OSEnterSum 值。各个任务开关中断的状态可不同,这样实现了开关中断的嵌套。

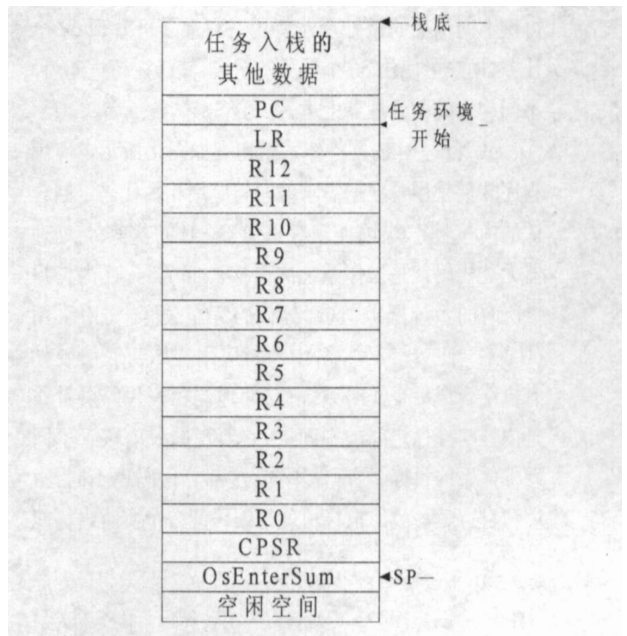


图 2 任务堆栈结构图

堆栈任务初始化代码即函数 `OSTaskStkInit()` ,该函数用于系统创建用户任务时,建立并初始化任务堆栈,其跟处理器硬件体系密切相关,它将所有的寄存器压栈,返回新的堆栈栈顶,并将它们保存在该任务的任务控制块 `OS_TCB` 中,最终使初始化后的堆栈跟刚发生过一次中断一样。这样,系统无需对调度程序作特殊处理即可直接对新任务进行调度。由于在 `LPC2210` 中堆栈是按 32 位数据类型来进行操作,所以堆栈数据类型 `OS_STK` 声明为 32 位无符号整数。`OSTaskStkInit()` 函数关键代码如下:

```
*stk = (OS_STK) task; /* pc */
* - - stk = (OS_STK) task; /* lr */
* - - stk = 0; /* r12...r1 同理 */
* - - stk = (unsigned int) pdata; /* r0,第一个参数使用 R0 传递 */
* - - stk = (USER_USING_MODE|0x00); /* spsr,允许 IRQ, FIQ 中断 */
```

另外有几个系统规定的 hook 函数:`OSTaskGreateHook()` ,`OSTaskDelHook()` ,`OSTaskSwHook()` ,`OSTaskStatHook()` ,`OSTimeTickHook()` 。本移植将它们都设置为空函数。

2.3 OS_CPU_A.S 文件移植

此文件包括四个函数都涉及对寄存器的处理,跟处理器有关,由于不同处理器有不同寄存器,所以操作系统在这个文件里给用户留下四个函数接口,以使用户根据所选处理器编写相应的汇编程序以完成固定功能。分别为:多任务启动函数中调用的 `OSStartHighRdy()` ,任务切换函数 `OSCtxSw()` ,中断任务切换函数 `OSIntCtxSw()` 和时钟节拍服务函数 `OSTickISR()` 。

多任务启动函数中调用 `OSStartHighRdy()` 函数,`μC/OS-II` 启动多任务环境的函数是 `OSStart()` ,用户在调用该函数之前,必须建立一个或多个任务。`OSStart()` 最终调用函数 `OSStartHighRdy()` 运行优先级最高的任务。

`OSCtxSw()` 函数是由任务级任务切换函数 `OS_TASK_SW()` 调用,实现从低优先级向高优先级切换。文中任务级切换是通过 `SWI` 软中断来实现。该中断完成:保存任务的环境变量;将当前任务的堆栈指针 `SP` 存入任务控制块 `OS_TCB` 中;载入就绪最高优先级任务的堆栈指针 `SP`;恢复就绪最高优先级任务的环境变量;中断返回。软中断的异常处理程序挂接到 `CPU` 内核,通过修改启动代码里的异常向量表实现。中断服务程序与 `CPU` 芯片相关,中断源的挂接通过增加汇编接口的支持来实现,即在文件 `IRQ.S` 的适当位置添加如下的程序代码:`Z.Handler HANDLER Z.Excep-`

`tion` ,其中 `Z` 替换为所需要的字符串,举例代码如下:

```
IRQ_Handler HANDLER IRQ_Exception ; IRQ 中断
```

`OSIntCtxSw()` 是中断级任务切换函数,在中断退出时由函数 `OSIntExit()` 调用。在中断服务程序中,当发现有高优先级任务等待时钟信号到来时,在中断退出后并不返回被中断任务,而是直接调度就绪的最高优先级任务执行,从而尽快响应高优先级任务,保证系统实时性。`OSIntCtxSw()` 函数基本原理与任务级切换相同,只是由于进入中断时已保存了被中断任务的环境变量,就不进行类似操作,只需要对堆栈指针进行调整。中断任务切换函数 `OSIntCtxSw()` 部分代码如下:

```
LDR R1, =OSTCBCur;保存当前任务堆栈指针到当前任务的 TCB
LDR R1, [R1]
STR SP, [R1]
MOV SP, R4 ;设置堆栈指针
LDR R3, =OsEnterSum ;恢复新任务的 OsEnterSum
MSR SPS_cxsf, R5 ;恢复 CPSR
LDMFD SP!, {R0 - R12, LR, PC}^ ;运行新任务
```

系统时钟节拍中断函数 `OSTickISR()` 是一个周期性中断,为内核提供时钟节拍。其周期决定了内核所能给应用系统提供的最小时间间隔服务。本移植中,`IRQ` 是受 `μC/OS-II` 管理的中断,对 `FIQ` 不处理。时钟节拍的移植可通过汇编宏完成。

以上完成了 `μC/OS-II` 在 `LPC2210` 上的移植工作。为了测试移植代码的运行情况,建立两个 `μC/OS-II` 的任务,一个任务用于检测按键输入(本测试中为 `P0.14` 口输入),称之为按键检测任务;另一个任务用于控制蜂鸣器,称之为蜂鸣器控制任务。蜂鸣器控制任务平时处于等待状态,当按键检测任务检测到有效按键输入时,立即唤醒蜂鸣器控制任务。`LPC2210` 利用外扩的 16MB 的 `FLASH` 存储数据及程序。利用 `ADSV1.2` 及 `AXD` 进行 `JTAG` 仿真调试。测试结果正确,脱机运行稳定,因此本移植成功。

3 结论

`μC/OS-II` 是一个很有优势的嵌入式实时操作系统,已在多种结构微处理器上成功移植应用,尤其在基于 `ARM` 体系的嵌入式系统领域有着更广泛的应用前景,所以研究 `μC/OS-II` 在 `ARM` 微处理器上的移植应用,能更深入地了解实时操作系统原理,加速基于 `ARM` 微处理器的应用开发,为更进一步应用打下基础。文中成功实现了 `μC/OS-II` 在 `LPC2210` 上的移植,并通过简单的多任务例程得到验证。

采用2.1节给出的属性约简算法,得到结果如表2所示,带*的为核值。测试用例由15个减少到12个。

表2 属性约简后的结果

U'	U	a	b	d	e	f
1	1	0	0	1	*1	1
2	2,3	0	*1	*0	*0	1
3	4,7	1	1	0	0	1
4	5,6	*1	1	1	*0	1
5	8	1	*0	*1	1	1
6	9	*1	0	1	*0	1
7	10	0	*0	0	0	0
8	11	*0	0	1	*0	0
9	12	*0	1	*1	0	0
10	13	0	1	0	*1	0
11	14	1	*1	1	*1	0
12	15	1	0	*0	1	0

根据2.2节提出的新的属性值约简算法对表2的数据进一步进行值约简,得到表3。

表3 最终的测试数据表

U''	a	b	d	e	f
1	~	0	1	1	1
2	~	1	0	0	1
3	1	~	~	0	1
4	~	0	~	~	0
5	0	~	1	~	0
6	~	~	~	1	0

通过对三个数据表的分析可以发现,如果只进行到属性约简,输入数据量降低36%;对表2继续进行值约简处理得到表3,输入数据量减少了84%。

表4根据文献[4]中值约简算法得到,可以看出,输入数据量比表3多了50%。因此,采用改进的值约简算法能很好地提高测试用例优化效果。

(上接第105页)

参考文献:

[1] Labrosse J J. $\mu C/OS-II$ —源码公开的实时嵌入式操作系统[M]. 邵贝贝译. 北京:中国电力出版社,2001.
 [2] 周立功. ARM与嵌入式系统基础教程[M]. 广州:广州周立功单片机发展有限公司,2004.

表4 文献[4]算法值约简结果

U''	a	b	d	e	f
1	~	0	1	1	1
2	~	1	0	0	1
3	1	~	~	0	1
4	~	0	1	~	1
5	~	0	~	~	0
6	0	~	~	0	0
7	0	~	1	~	0
8	~	~	~	1	0
9	~	1	~	1	0
10	~	~	0	~	0

4 结论

采用粗糙集优化软件测试用例,是粗糙集在软件测试上一个试探性的应用和研究。通过实例可以看出,把Rough集用于软件测试用例优化上是可行的,而采用合适的Rough集约简算法能提高优化过程的效率,有效地降低测试数据量。

不过,为了突出说明,选用的例子比较特殊。今后的工作将以连续性和区间性的数据展开。粗糙集在处理这类问题上也有了较为成熟的研究。

参考文献:

[1] 徐中伟,吴芳美. 软件测试质量的度量[J]. 计算机工程与应用,2002,38(21):100-102.
 [2] 杨涛,李龙澍. Rough集在软件测试用例选择中的应用[J]. 微机发展,2005,15(2):12-14.
 [3] 刘清. Rough集及Rough集推理[M]. 北京:科学出版社,2001:57-60.
 [4] 胡斐,张峰筠,刘少辉. 一种基于Rough集的属性值约简算法[J]. 计算机工程与应用,2003(31):48-51.
 [5] 朱红. 基于Rough Set的属性及属性值简约的一种算法[J]. 湘潭大学自然科学学报,2002,24(3):36-39.
 [6] 杨振峰,郭景峰,常峰. 一种基于粗糙集的值约简算法[J]. 计算机工程,2003,29(9):96-97.
 [7] 顾军华,周艳聪,宋洁,等. 一种新的求解属性值约简算法[J]. 南开大学学报,2003,36(4):38-42.

[3] PHILIPS公司.LPC2210 User Manual[EB/OL]. 2004.
<http://www.zlgmcu.com>.
 [4] 田泽. 嵌入式系统开发与应用实验教程[M]. 北京:北京航空航天大学出版社,2004.
 [5] 桑楠. 嵌入式系统原理与应用开发技术[M]. 北京:北京航空航天大学出版社,2002.

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](#)

邀请注册码



关注论坛公众号

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](#)

邀请注册码



关注论坛公众号

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](#)
86. [MVB1 类设备控制器的 FPGA 设计](#)
87. [MVB 接口彩色液晶显示诊断单元的显示应用软件设计](#)
88. [IPv6 新型套接字的网络编程剖析](#)
89. [基于规则的 IPv4 源程序到 IPv6 源程序的移植方法](#)
90. [MVB 网络接口单元的 SOC 解决方案](#)
91. [基于 IPSec 协议的 IPv6 安全研究](#)
92. [具有 VME 总线的车载安全计算机 MVB 通信板卡](#)
93. [SD 卡的传输协议和读写程序](#)
94. [基于 SCTP 的 TLS 应用](#)
95. [基于 IPv6 的静态路由实验设计](#)
96. [基于 MVB 的地铁列车司机显示系统研究](#)
97. [基于参数优化批处理的 TLS 协议](#)
98. [SSD 数据结构与算法综述](#)
99. [大容量 NAND Flash 文件系统中的地址映射算法研究](#)
100. [基于 MVB 总线的动车组门控系统的设计与仿真研究](#)
101. [多功能车辆总线 MVB 控制](#)
102. [基于 LabVIEW 的 MVB 和 WTB 帧解码方法](#)

邀请注册码



关注论坛公众号

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](#)

邀请注册码



关注论坛公众号

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. [基于 VxWorks 系统的 MiniGUI 图形界面开发](#)
60. [VxWorks BSP 开发中的 PCI 配置方法](#)
61. [VxWorks 在 S3C2410 上的 BSP 设计](#)
62. [VxWorks 操作系统中 PCI 总线驱动程序的设计与实现](#)
63. [VxWorks 概述](#)
64. [基于 AT91RM9200 的 VxWorks END 网络驱动开发](#)
65. [基于 EBD9200 的 VxWorks BSP 设计和实现](#)
66. [基于 VxWorks 的 BSP 技术分析](#)
67. [ARM LPC2210 的 VxWorks BSP 源码](#)
68. [基于 LPC2210 的 VxWorks BSP 移植](#)
69. [基于 VxWorks 平台的 SCTP 协议软件设计实现](#)
70. [VxWorks 快速启动的实现方法\[上电到应用程序 1 秒\]](#)

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](#)

邀请注册码



关注论坛公众号

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](#)
53. [基于 AT91RM9200 的嵌入式 Linux 系统的移植与实现](#)
54. [路由信息协议在 Linux 平台上的实现](#)
55. [Linux 下 IPv6 高级路由器的实现](#)
56. [基于 Android 平台的嵌入式视频监控系统设计](#)

邀请注册码



关注论坛公众号

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](#)
35. [NAND FLASH 在 WINCENET 系统中的应用设计](#)

邀请注册码



关注论坛公众号

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](#)
31. [基于 MPC850 的多功能通信管理器](#)
32. [基于 MPC8640D 处理系统的技术研究](#)
33. [基于双核 MPC8641D 处理器的计算机模块设计](#)
34. [基于 MPC8641D 处理器的对称多处理技术研究](#)

邀请注册码



关注论坛公众号

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](#)
36. [UBoot 在 AT91RM9200 上的移植简析](#)
37. [基于工控级 AT91RM9200 开发板的 UBoot 移植分析](#)
38. [基于 ARM11 和 Zigbee 的人员定位防丢器](#)

邀请注册码



关注论坛公众号

39. [基于 NAND FLASH 的嵌入式系统启动速度的研究](#)
40. [μ COS II 在 ARM7 上的移植](#)
41. [基于 ARM11 的嵌入式视频采集系统设计](#)

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 技术研究与展望](#)
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](#)

邀请注册码



关注论坛公众号

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](#)
41. [EFI-Tiano 环境下的 AES 算法应用模型](#)
42. [EFI 及其安全性研究](#)
43. [基于 UEFI Shell 的 PreOS Application 的开发与研究](#)
44. [UEFI Bootkit 模型与分析](#)
45. [UEFI 计算机系统快速调试方法的实现](#)
46. [基于 EFI 系统的多文件系统解决方案](#)
47. [基于 UEFI 的可信 Tiano 设计与研究](#)

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. [用 IPv6 编程接口实现有连接通信的方法](#)
17. [一种战棋游戏的 AI 算法设计与实现浅析](#)
18. [基于 TLS 协议的 ECC 扩展研究](#)

邀请注册码



关注论坛公众号

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](#)
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. [FPGA 的 VHDL 设计策略](#)
16. [用 FPGA 实现串口通信的设计](#)
17. [GPIB 接口的 FPGA 实现](#)
18. [一种基于 FPGA 的 FFT 阵列处理器](#)
19. [基于 FPGA 的 FFT 信号处理器的硬件实现](#)
20. [CPLD 在 CAN 通讯卡中的应用](#)
21. [用 CPLD 实现同步串口与异步串口的转换](#)

邀请注册码



关注论坛公众号