

# 基于 VxWorks平台的 SCTP协议软件设计实现

林利,马宏,朱宇航

(信息工程大学 国家数字交换系统工程技术研究中心,河南 郑州 450002)

**摘要:**由于 VxWorks平台在电信领域的巨大市场和 SCTP协议在 IP网传输电信业务数据的优异性能,研究基于 VxWorks平台的 SCTP协议具有重要的意义。通过对 SCTP协议基本概念和高级特性的简单介绍及它与传统传输协议主要区别的简单对比,凸显了 SCTP协议的优越性能。提出了 SCTP协议的软件设计方案。重点设计了协议的初始化机制、数据发送机制和数据接收机制,重点优化了协议的实时处理能力。通过工程实践表明,即使在不稳定网络等恶劣条件下, SCTP协议能够更高效,更可靠地完成数据的传输。证明了 SCTP协议作为下一代传输协议具有广阔的前景。

**关键词:**流控制传输协议 (SCTP);多宿;多流;偶联

**中图分类号:** TN915.04      **文献标识码:** A      **文章编号:** 1671-0673(2010)05-0595-05

## SCTP Protocol Software Design Implementation Based on VxWorks

LIN Li, MA Hong, ZHU Yu-hang

(National Digital Switching System Engineering & Technical Research Center,  
Information Engineering University, Zhengzhou 450002, China)

**Abstract:** VxWorks platform has a huge market in the field of telecommunications and the SCTP protocol exhibits outstanding performance in IP network transmission of telecommunication traffic, so it is very important to research SCTP protocol based on VxWorks platform. This paper introduces the basic concept and advanced features of the SCTP protocol, and makes a brief contrast with the traditional transfer protocol, thus showing the superior performance of the SCTP protocol. SCTP protocol software design is also proposed with focus on the design of the protocol initialization mechanism, data transmission mechanism and the data receiving system, this optimizes real-time processing capability of the protocol especially. The engineering application shows that even in unstable networks and other adverse conditions, SCTP protocol can be more efficient and more reliable in data transmission, and SCTP protocol has broad prospects as a next-generation transport protocol.

**Key words:** stream control transmission protocol (SCTP); multi-home; multi-stream; association

因特网和电信网曾经是两个独立的网络,它们提供不同的业务。因特网的爆炸式发展和人们对电信业务的更多需求,驱动着两大网络的融合。IP技术的飞速发展,促使网络朝着以IP为基础的分组网方向融合发展。随着VoIP和3G无线网络的兴起,通信业界迫切需要传统的七号信令可以承载在IP网上。IETF为此定义了SIGTRAN体系结构。针对目前IP网络中传输层TCP/UDP的缺点,在SIGTRAN中新增了SCTP协议用于保障信令传送的可靠性,为IP网提供可靠的数据传输。

新生成的SCTP集合了TCP、UDP的优点,解决TCP对头阻塞、Dos攻击等问题<sup>[1]</sup>,另外新增了许多功

能,如多流、多宿、面向消息传输、Cookie安全机制、动态地址重配置、部分可靠传输、支持移动 IP等等。不但满足了在分组网上传输信令的要求,而且成为了通用的传输层协议,能提供更好的多媒体传输服务,移动环境下端到端的高效服务<sup>[2]</sup>。

## 1 SCTP协议

### 1.1 基本概念

**偶联** SCTP偶联是 2个 SCTP端点间的一个对应关系,它包括了 2个 SCTP端点,以及验证标签和传输顺序号码等信息在内的协议状态信息。一个偶联可以用传输地址来唯一识别,传输地址是用多个网络层地址,传输层协议和传输层端口号定义的<sup>[1]</sup>。

**多宿** 指建立偶联的端点可以支持多个 IP地址。每个 IP地址都可以用于偶联的通信。应用层规定其中一个 IP是主地址,其它 IP是备用地址。

**多流** 流是偶联中的单向逻辑通道,是消息序列。多流就是一个偶联中可以有多条流同时进行数据传输。各个流相互独立。流内数据可以顺序传输,递交,也可无序传输,递交,这取决于应用层对流的性质的规定。

### 1.2 协议主要特性

SCTP协议有 5大特性。这 5大特性使 SCTP的传输性能相比 TCP、UDP等协议的传输性能有了很大的提高。

**多宿性** 相比 TCP的单一地址的连接方式,SCTP支持多 IP地址的偶联方式<sup>[3]</sup>。即建立偶联的端点可以使用多个 IP地址进行通信。若 SCTP只允许使用其中一条路径来传输数据,这条路径称为主路径,其它路径称为备选路径。只有在主路径失效的情况下,才会使用备选路径传输数据<sup>[2]</sup>。SCTP的多宿特点极大地提高了 SCTP会话的潜在存活能力。

**多流性** SCTP在偶联通信中支持多流机制。“流”在 TCP中指一系列的字节,而在 SCTP中是指发送到上层协议的用户消息。SCTP用户在建立偶联时,可以规定偶联支持的流的数目。偶联可以通过每个流独立地传递消息序列。且消息中包含的传输处理标记(flags)可控制消息的分段和组装。流标识和流序号用于确保在给定流中消息可按顺序发送,称为部分有序传递。阻塞的流不会影响同一偶联中的其它流。

**初始化保护** SCTP和 TCP对连接的初始化都通过报文握手完成。TCP采用 3次握手。通信如图 1所示。

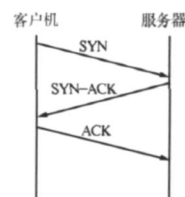


图 1 SCTP 3次握手示例

SCTP初始化机制采用 4次握手。客户机用 init报文发起连接。服务器响应 init-ack报文,其中包含一个“状态 cookie”,cookie包括建立连接所需的全部信息,服务器不必再保存任何信息或分配任何资源。这样可以防止 TCP中出现的 SYN-flooding等 DOS攻击。客户机然后使用 cookie-echo报文响应,此报文包含从服务器收到的状态 cookie。服务器收到报文后才分配资源,向客户机发送 cookie-ack响应。SCTP通信如图 2所示。虽然 SCTP比 TCP在会话建立的过程中多了一次确认的过程,但是在第 1次交互过后,SCTP的确认包就可以负载有效数据信息,实际上它比 TCP更早开始终端之间的信息传输。通过 SCTP 4次握手中的 cookie机制,使得服务器资源得到合理利用,提供更加可靠的服务。

**消息分帧** SCTP在数据传输中提供了消息分帧功能,对于面向消息的协议非常有益。当一端执行写操作时,可确保对等端读出的数据大小与此相同。与此不同,TCP是按照字节流的方式进行操作,没有消息分帧功能,每次收端和发端处理的数据大小可能不同。因此,在传输电信网信令数据时,SCTP相比 TCP的效果更好。

**平滑关闭** SCTP与 TCP删除连接的不同之处在于 TCP连接的删除是半关闭的。在 TCP中,若通信一方关闭本地的连接,另一方仍可继续接收数据。由于应用程序很少使用这种半关闭状态,SCTP协议规定通信双方若有一方发送结束通话的信令消息后,发送方停止向对方发送消息,只将缓存中剩余的消息发送出去;接收方也

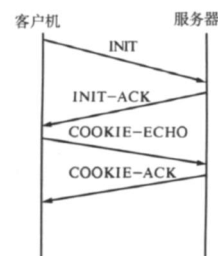


图 2 SCTP 4次握手示例

停止从对方接收数据,仅接收缓存剩下的数据。

## 1.3 与传统传输协议的比较

SCTP和 TCP都为保证用户的所有消息成功到达对端端点提供了必要的可靠性,同时也提供了消息的有序递交。在今天的网络中,拥塞控制是任何一个传输协议的关键因素。目前 SCTP和 TCP都提供相同的拥塞控制机制——慢增快减拥塞窗口管理,这确保了它们在今后网络的共同应用中的公平性。同时,SCTP在连接建立,无序传输,消息边界,选择证实和关闭等方面与 TCP不同。SCTP在这些方面都有相应的改进。

SCTP与 UDP都是面向消息的协议,不同之处是UDP不提供可靠传输,不提供顺序控制和连接确认,而SCTP在确认方式下提供无差错无重复的可靠传输。除此之外,SCTP是一种单播协议,不支持 IP组播和广播,而 UDP支持 IP组播和广播。SCTP与 TCP、UDP的特性比较总结如表 1所示。

表 1 协议特性比较

协议特性	SCTP	TCP	UDP
是否面向连接			×
可靠数据传输			×
拥塞控制和避免			×
消息边界保留		×	
消息捆绑			×
支持多宿主机制		×	×
支持多流机制		×	×
数据的无序递交		×	
COOKIE安全验证机制		×	×
可达性检测		×	×

## 2 SCTP协议实现

### 2.1 VxWorks平台实现 SCTP协议的背景

Vxworks操作系统是美国 WindRiver公司开发的一种嵌入式实时操作系统,具有良好的可靠性和卓越的实时性,被广泛地应用在电信、军事和航天等高精度技术领域中。VxWorks网络协议栈是 VxWorks内部的实时 TCP/IP协议栈。目前 VxWorks网络协议栈并不包含 SCTP协议。

SCTP被称为“超级 TCP”。是因为它比 TCP具有更好的传输性能。当电信网的信令数据在 IP网传输时,首先必须保证传输的可靠性和高效性。除了 SCTP外,UDP、TCP等传输协议都无法保证。因为UDP的无连接通信机制无法保证信令的准确传输;TCP面向数据流的通信也降低了信令传输的实时性。而 SCTP同时支持无连接通信和面向连接的通信,且 SCTP是将每个消息作为一个处理单元,而不是将字节流作为处理单元。因此,SCTP协议才是实现信令在 IP网上传输的理想协议。

目前由于商用软件的价格非常昂贵,且一个致力于在不同的 BSD上提供免费网络协议栈的组织 KAME,实现了 BSD内核级的 SCTP协议栈。因此,根据开源的 SCTP协议代码实现 VxWorks系统下的 SCTP通信是非常经济和可行的。

### 2.2 协议架构实现

基于 VxWorks平台的 SCTP协议软件设计主要分为 6个模块。它们分别是应用控制模块、流引擎控制模块、发送控制模块、传输控制模块、接收控制模块和网络适配模块。下面简单介绍各模块的主要功能。协议架构如图 3所示。



图 3 SCTP协议功能示意图

**应用控制模块** 负责与应用层进程通信,向上层发送接收到数据和向下转发应用层的指令。

**流引擎控制模块** 负责 SCTP协议中各个功能消息的封装,如 NIT消息、SHUTDOWN消息和 ABORT消息等。针对 SCTP协议中的“流”相关功能进行控制和处理,如设置流 ID,流序号等。

**发送控制模块** 控制通信过程中发送窗口值的大小,通过心跳消息监控路径状态,以及实现流量控制的机制等等。

**传输控制模块** 控制用户回调函数和协议功能等等。根据上层用户的指令控制偶联状态参数。

**接收控制模块** 通过读取报头信息判断数据的正误和接收方式。通过控制发送 sack消息来控制数据的确认机制。

**网络适配模块** 通过加密算法往数据写入密钥,完成数据的传输层封装,给相应数据设置定时器等功能。

## 2.3 协议初始化实现

当 VxWorks 系统要支持 SCTP 协议通信时,首先需要进行协议的初始化。下面简单介绍 SCTP 协议的初始化过程。

SCTP 协议的初始化主要有 3 个部分。第 1 部分就是初始化 SCTP 库环境,它将初始化 SCTP 协议的文件描述符池,定时器列表,通信套接字和发送缓冲区等;第 2 部分就是注册应用实例。SCTP 协议在通信过程中为相应的偶联建立一个实例来维持通信;第 3 部分是初始化偶联结构体。它包括申请新偶联,初始化路径管理结构体和 SCTP 实例控制结构体。此外,当客户端发起偶联请求后,协议将初始化重传管理结构体、流量管理结构体、接收管理结构体和流引擎控制结构体等 4 个用于偶联通信的结构体。协议初始化结构如图 4 所示。



图 4 SCTP 协议初始化示意图

## 2.4 数据发送实现

当 SCTP 协议初始化完成之后,一旦上层用户送来指令要求发送数据,协议首先要查看 SCTP 库环境参数等是否已经建立。然后在流引擎模块划分一片缓存空间,将上层数据存入该缓冲区。接着判断偶联对端的缓存空间是否符合传输要求,OBPA 数据是否超出拥塞窗口值。接着是为数据添加不同的报头。共有 sack, data 和 ctrl 3 种报头格式。之后还需要向数据写入端口号,验证标签和流的参数等。完成以上工作后将数据保存到发送缓冲区,准备发送数据。同时还将数据保存到消息重发队列,若数据发送失败可在此对数据进行重发。最后根据 MTU 值对数据进行捆绑,调用 send 函数发送消息。发送流程如图 5 所示。



图 5 数据发送过程示意图

## 2.5 数据接收实现

当 SCTP 协议发现新消息时,由分布控制层判断数据长度等接收条件,然后将新数据存入接收缓存。接着先是读取数据的端口号和 IP 地址信息等,根据本地的偶联管理表信息判断数据是否正确,以及数据对应哪个偶联实例。然后还要判断数据有无填充字节。完成以上工作后将数据存储到等待队列,等待上层接收。当上层命令准备接收数据时再将数据存入通知队列。上层从通知队列读取数据。此外,上层完成数据接收之后,接收控制模块将发送控制指令,发送对应的 sack 确认信息。

## 3 SCTP 协议实现的优化

本次 SCTP 协议的软件实现是基于开源免费的 SCTP 网络协议栈,在针对具体应用的时候需要做一些优化,以提高程序的执行效率和可读性。

### 3.1 接收机制的优化

开源协议栈将 SCTP 协议接收过程定义为 3 个步骤,首先将数据存入流控制模块的接收缓冲中,定义一个指针指向这个缓存。之后将该指针指向等待队列,最后该指针又指向通知队列,由上层用户读取该指针指向的数据。这样做虽然提高了协议的安全性,但是过多的控制命令既影响了程序的实时性,又增加了程序的复杂性。因此,在实现过程中可以将 3 个步骤并为一步完成,如图 7 所示。通过改进,协议只有一个统一的接收队列,新数据进入队列后直接读取相应流信息,调用上层原语通知上层接收数据。

### 3.2 发送机制的优化

定义静态的 SCTP 协议全局变量和发送缓存区。因为目前实现的 BSD 内核级的 SCTP 协议栈都采用动态的全局变量和发送缓存。这影响了数据处理的速度,对实时性能产生不利影响。

可对每个数据流的参数进行设置。针对具体数据定制相应流功能。比如传输流媒体数据时可以去掉分帧功能,针对消息的类型设置是否需要重传,这样可以减少重传对有效带宽的影响。

验证机制的改进。开源协议栈规定对多个数据块进行一次确认。为了提高协议实时处理性能,可以针对具体应用环境将多数据块一次确认改成每数据块一次确认。通过修改定时器参数可实现。

### 3.3 流量控制机制的优化

SCTP协议规定通信是一直在主路径上传输数据。只有在主路径出现故障无法工作时才切换到备选路径。这虽然保障了数据传输的稳定性,但却没有考虑实际网络状况,容易降低数据传输效率。因为主路径的网络状况不一定一直都是最优的。因此,在实现协议时利用心跳消息查询路径状况信息和RTO值,与主路径的RTO值对比,若小于主路径的RTO值则切换到新的路径作为数据传输通道。只有选取网络延时最小的路径作为当前传输路径才能达到最优传输效率。另一种优化策略是将原来的主备路径传输方式改成多路径同时传输的传输方式。各个传输路径的传输控制相互独立,有自己独立的发送缓存和独立的流量控制机制。这种策略将大大提高数据传输的效率和协议的实时性。

3.4 协议栈的裁减

开源的SCTP协议栈功能全面,支持windows等多种系统平台。在VxWorks系统下实现SCTP通信时就可以根据需要裁减部分无用代码,包括Windows环境变量和一些宏定义语句,协议中数据处理的优化等,这将大大简化协议栈代码,提高程序执行效率。

### 4 结束语

SCTP协议通过选择性反馈,无序递交,多流通道和多宿性等协议优化,使其在网络数据,尤其在流媒体数据的传输上体现出优异的性能。本文通过分析SCTP协议主要特性和简单比较传统传输协议可以看出,SCTP在可靠性上优于UDP,因为SCTP支持面向连接的通信方式;SCTP在传输效率上优于TCP,因为SCTP是面向消息的通信方式,并支持有选择的按序传输。弥补了TCP严格有序和面向字节流处理机制的不足。同时SCTP具有多宿和多流等新特性,使得数据信息的传输变得更为高效和可靠。最后还针对协议的实时处理能力进行了优化。相信在不久的将来SCTP会得到更大的运用,取代基于TCP/UDP传输协议的网络数据传输方案,提供一个稳定、高效的传输层支持,促进新业务的发展和应用。

#### 参考文献:

- [1] 中华人民共和国信息产业部. YD/T 1194 - 2002流控制传送协议(SCTP) [S]. 2002
- [2] RFC 2960. Stream Control Transmission Protocol[S]. USA: IETF, 2000.
- [3] 王璐. 流控制传输协议高级特性介绍 [J]. 技术交流, 2008, 2(1): 01.
- [4] 杜恒. SCTP流媒体传输性能分析研究 [J]. 计算机与信息技术, 2007, 36(1): 3-4
- [5] 郭云霞. 流控制传输协议的研究及NS2仿真实现 [M]. 南京:南京邮电大学, 2004.

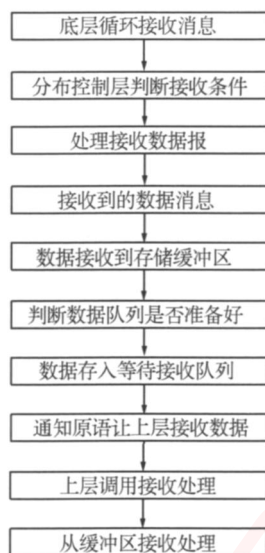


图6 数据接收流程



图7 改进的数据接收流程

# 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. [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](#)

## 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](#)
53. [基于 AT91RM9200 的嵌入式 Linux 系统的移植与实现](#)

## 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](#)
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 I/O 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](#)

## 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](#)

## 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 COM Express 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 CPU/GPU 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](#)
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.