以太网时钟同步协议的研究

汪志水[1]　范林刚[2]

（1、713 厂，江西 九江 332100　2、武汉大学电子信息学院，湖北 武汉 430000）

摘　要：针对现代工业应用对时钟同步精度的需求，分析了目前在工业以太网应用中的时钟同步协议（NTP 协议）的同步原理、时间戳精度、以及应用场合和各自特点；从不同的角度对现有的时钟同步协议进行了归纳总结和进一步的分析。比较了 IEEE1588 标准的精确时钟同步协议(PTP 协议)与其它同步协议的异同，重点介绍了在实现过程中为取得更高精度可以采用的具体措施。

关键词：时钟同步；时间戳；NTP 协议；IEEE 1588；PTP 协议

引言

现代测试与测量技术飞速的发展，使其对数据传输与处理综合要求的逐步提高，于是以太网以其优秀的传输性能开始被广泛采用。人们开始在测试与测量系统中直接接入以太网，然后使用 GPIB、VXI 或者 PXI 总线连接仪器，达到向远程地点传输数据或者从远程地点接收命令的目的。因此，许多应用系统都是建立在分布式网络环境之上，此时，如果没有一个统一的、准确的时钟，这些应用很难正常的协调工作和运行。现场总线国际标准（IEC）在 IEC 6185025 12. 6. 6. 1 和 12. 6. 6. 2 部分中对时间同步精确度定义了 5 个级别 IECclasses T1~T5，分别是：① IEC Class T1: 1 ms; ② IEC Class T2: 0. 1 ms; ③ IEC Class T3: ±25μs; ④ IEC Class T4: ±4μs; ⑤ IEC Class T5: ±1μs。

目前，工业应用的网络时钟同步系统主要是应用网络时间协议(NTP,Network Time Protocol)进行同步。受协议实现原理的制约，同步精度大多在 IECclasses T1（毫秒级）以下，很难满足分布式控制系统的要求，高性能的时钟同步控制器的研究很有现实意义。本文通过比较 IEEE1588 标准的精密时间协议 (PTP, Precision Time Protocol) 与其它同步协议的异同，重点介绍了为实现高精度同步，PTP 协议在实现过程中可以采用的一些措施。

1 时间同步方法的分类

按实现方式来看，时钟同步有硬时钟同步、软时钟同步和混合时钟同步 3 种，他们各自的精度不同、成本不一，有自己不可替代的应用价值。

1.1 硬时钟同步

世界上大多数国家的时间都是基于地球自转和天文历法的协调世界时 UTC(Universal Time Coordinated)为基准时间。硬件时间同步就是利用一定的硬件设备（GPS 接受机、高精度外部时钟、无线电广播接受机、UTC 接收机等）接收 UTC 进行时间同步或利用 UTC 的时间的固定差值借助高精度外部时钟实现局部时钟同步。

硬时钟同步的优点是可获得很高的时间精度（一般可达到 10^{-9}—10^{-6} 秒），缺点是成本高、安装和操作不方便、系统的自适应性较差。适合小规模、要求高精度同步时间的系统。

1.2 软时钟同步

软时钟同步是采用时钟同步算法，实现网络各个结点的时钟之间的同步，一般不需要对结点的硬件时钟进行操作。但是由于网络传输延时及其不确定性、结点间的同步偏差容易积累，导致误差较大，精度不高（一般可达到 10^{-6}—10^{-3} 秒），此种方法的优点是构成的系统方便灵活、成本低廉；缺点是工作量大，精度不高。

1.3 混合时钟同步

将软时钟同步和硬时钟同步结合起来，发挥各自优点。将大规模分布式系统的结点网络分成多个网段，在网段间使用硬时钟同步，在各个网段使用软时钟同步，只在网段相交处使用硬件设备减少了硬件开销；将广域网划分为网段，大大降低了传输延时，增加了传输延时的可预测性，非常有利于同步精度。从而实现系统方便灵活、成本低廉的同时确保了较高精度（介于软时钟同步和硬时钟同步精度之间）。适用于大规模的分布式系统。

2 几种常见的同步协议

2.1 网络时间协议(NTP , Network Time Protocol)

RFC 1305 [MIL92]完整的定义了网络时间协议(NTP)，它是典型的软时钟同步协议。NTP 基于无连接的 IP 协议和 UDP，能够在复杂的 Internet 环境中提供精确的时间服务。采用客户机/服务器模式，客户机以传统的客户机/服务器方式周期性地向服务器请求时间信息，服务器接受到请求后对其进行响应来达到同步。

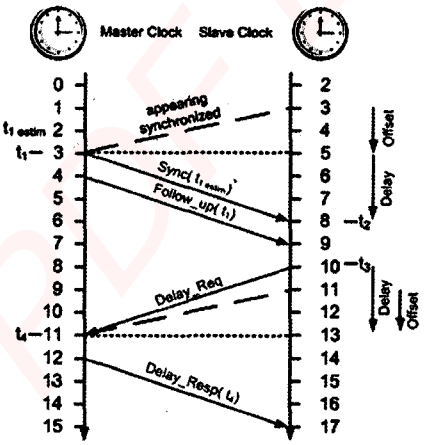

图 1 PTP 同步原理图

NTP 协议可以使计算机对其服务器或时钟源(如石英钟，GPS 等)进行同步，由于网络传输延时和计算机处理、协议封装(解封)开销等不可具体量化的时延，从而造成 NTP 精度无法进一步提高，精度一般在毫秒级，随网络结构和网络的负荷变化。

2.2 简单网络时间协议(SNTP, Simple Network Time Protocol)

目前以太网中时钟同步的主要协议。当不须要实现 RFC 1305 所描述的 NTP 完全功能的情况下，可以使用 SNTP，为 NTP 的简化模型，主要用来同步因特网中的计算机时钟。其同步机理与 NTP 基本相同，在此不再赘述。SNTP 有单播方式(点对点) 和广播方式(点对多点) 两种操作模式，也能在 IP 多播方式下操作。它的时间精度依赖于客户端和服务端网络的情况。工作站或服务器一般来说 1~10ms 的精度，广域网中的 PC 一般是 100ms 的精度，在局域网内部则可以达到 0.5~2ms 的精度。

SNTP 是针对广泛分散在互联网上的各个独立的系统或是网络条件相对不稳定的场合，它通过复杂的协议保证了协议的兼容性，并可以使用层次式的时间分布模型、冗余服务器和多条网络路径等措施来获得时间同步高可靠性。因此适用于条件复杂、稳定性较差以及无线、蓝牙等网络环境。

2.3 基于 IEEE1588 的精密时间协议 (PTP, Precision Time Protocol)

IEEE1588 标准定义了一种精确时间协议 (PTP, Precision Time Protocol)，用于对以太网或其他分布式系统进行时钟同步。它采用软硬件配合的方式，最高可以实现纳秒量级的超高精度的时钟同步，高于其他网络时间同步协议的精度。

2.3.1 校时原理

IEEE 1588 同步过程分为偏移测量阶段和延迟测量 2 个阶段。偏移测量阶段用来修正主、从属时钟的时间差，延时测量用来测量主从结点间的传输延时。

如图 1 所示，在该偏移测量过程中，主时钟周期性(缺省为 1 次/2s)发出一个包含预测发送时间($t_{estimate}$)的同步数据报 (Sync)，从时钟收到 Sync 后记录收到 Sync 的本地时间 t_2。主时钟在 Sync 信息发出后发出一个 Follow- Up 信息，该数据报包含了一个记载了 Sync 信息的真实发出时间 t_1 的时间戳。这样做能使报文传输和时间测量分

Break;

}

3 结束语

随着电脑技术的迅速发展，USB 设备得到了广泛的应用，然而也带来了严重的安全问题，若能够及时对 USB 设备进行管理控制，可以更大程度的消除安全隐患。本文设计的 USB 端口监控程序，实现了对 USB 设备的实时监控，具有一定的使用价值。

参考文献

[1]陈贤敏.USB 接口监控系统的研究与实现.
[2]张敏.USB 接口存取控制系统研究与设计.
[3]王黎.计算机数据安全管理系统的研究与实现.

开进行,相互不影响。

在该延时测量过程中，为了测量主从结点间的网络传输延时，IEEE 1588 定义了一个延迟请求数据报 (Delay Request Packet, Delay-Req)。从时钟不定时的发送给主时钟延时测量请求数据包 Delay- Req,然后监测记录下发送出去的本地确切时间 t_3,主时钟收到后在延迟响应数据报 (Delay Request Packet, Delay- Resp) 加入接收到 Delay- Req 的时间 t_4,并发送给从属时钟,从属时钟就可以非常准确地计算出网络延时。与偏移测量阶段不同的是,延迟测量阶段的延迟请求数据报是随机发出的,并没有时间限制。PTP 协议的假设主从时钟间的网络对称,即相互发送握手信号的过程中网络传输延时相等(都为 T_{Delay}),所以由于

由偏移测量过程得 $t_2 - t_1 = T_{Delay} + \Delta$ ①

由延迟测量过程得 $t_4 - t_3 = T_{Delay} - \Delta$ ②

由方程①②求得

传输延时 $T_{Delay} = \frac{1}{2}[(t_2 - t_1) + (t_4 - t_3)]$ ③

时间偏差 $\Delta = \frac{1}{2}[(t_2 - t_1) - (t_4 - t_3)]$ ④

则从属时钟新的时间 $T_{new} = T_{old} + \Delta$ ⑤

从时钟根据上述计算结果调整本地时间。

2.4 几种同步协议的比较

表 1 几种同步协议的比较

	PTP	NTP	GPS
软/硬件实现协议	任意方式	软件实现协议	混合实现协议
精度	微秒级	毫秒级	微妙（单节点） 毫秒（网络式系统）
IEC 等级	T5	T1	T3（单节点） T1（网络式系统）
工作模式	主从模式	全体平等	客户机/服务器
延迟校正	有	有	无
协议安全性	无	有	无
管理方式	自控	管理员设定	持续进行
占用网络和计算资源	少量	中等	中等
同步周期	默认 2 秒/次	变化的	1 秒/次
应用环境	分布式网络系统	条件复杂、稳定性较差的网络环境(Internet)	分布式网络系统

NTP 与 PTP 都是通过结点间的握手信号进行时钟同步的协议，时间戳的提取和发送直接关系到同步精度。TPT 针对 NTP 同步机制的缺陷做出了相应的调整,将发送同步报文的时间戳与同步报文分开发送,减少了数据处理延时,大大提高了同步精度(至少可达到 IEC Class T3),相对其它协议而言,有着绝对的优势。另外 TPT 并没有对实现形式进行定义,在实现方式上可以采用软时钟、硬时钟和混合时钟的多种实现形式,使系统的实现变得更加灵活。

3 PTP 协议在应用中的几点建议：

通过分析 PTP 协议的同步机制,我们了解了在同步过程中对同步精度影响最大的一些因素,针对这些影响可以采用相应的措施予以克服,从而达到更高精度。

3.1 最佳主时钟算法

主时钟源给整个系统提供标准时钟,所以主时钟的性能也在很大程度上决定了同步精度。BMC 算法在待同步网络内通过一定的算法选择工作状态最佳的结点时钟作为主时钟。在 PTP 网络中,可以用方差来表示时钟的性能。PTP 方差的大小就代表了时钟性能的好坏,方差的值越小,时钟间隔的总的波动范围就越小,表明时钟性能越好。如果条件允许,还可以同时在网络上提供多个备用时钟,根据某种特定算法,如果主时钟状况不佳,这些备用时钟中会有一个出来充当主时钟。

3.2 尽量保证需要同步的主从机间的网络结构对称

由协议的同步原理可知,在延迟测量阶段中,传输延迟测的是从节点到主结点的,而在结果运算过程中应该是用主节点到从节点的传输延迟参与的,只有二者近似相等时才能保证同步精度,保证网络结构对称能在一定程度上促进传输延迟相等。

3.3 从主结点到从节点间的传输延迟务必要等于从结点到主节点的传输延迟

如果不相等，应该采用相应的算法加以补偿。如第 3.2 点所述原因,在既定的非对称网络结构中,如果两种传输延迟不相等,又不能改变网络结构本身,可以采用一定的算法加以修正。

3.4 在两次延迟测量之间的时间,应该尽量保证传输延迟没有发生变化

因为偏移测量是每两秒一次,延迟测量是从节点不定时的请求进行,所以在两次延迟测量间最好是传输延迟没有发生变化。在网络比较稳定时,一般都能够保证;不稳定时,应该加大延时测量频率或者采用其它措施进行处理。

3.5 在子网之间(通过交换机、路由等)必须使用带有边界时钟的连接设备

由于网络传输延时的不确定性,该协议的同步精度很大程度上还取决于网络结构和网络负载，点对点的结构能提供更高的精度。但是由于网桥、交换机和路由器的传输延迟,当主从结点间通过它们时将引入很大的时间不稳定。为克服多个 PTP 子网的延迟较大,同步精度降低的问题,IEEE1588 引入边界时钟,使子网间的同步也成为点对点的同步关系。通过稳定的传输延时,来消除一定的网络延迟,使多个 PTP 子网互联的网络同步精度提高。

3.6 实现形式(软时钟/硬时钟)

PTP 协议只规定了同步机制,对实现形式并没有说明,用户可以根据自己的需求,采用软时钟,硬时钟或者混合时钟实现同步,通过前述的实现机制的比较,结合具体应用环境,可以选择合适的实现方式。在分布式网络系统中,由于协议栈处理也导致的很大的延迟波动,建议采用混合时钟同步。具体可以参考以下方式:将时间戳的生成和提取放在更接近传输介质的物理层,通过相应的硬件电路实现时间戳的处理,再由上层协议算法实现时钟同步(混合时钟同步)。

3.6 其它建议

对于实现过程中仍然存在的时间误差,可以借助一些数学算法对数据进行处理。例如:由于数据在以太网中的传输不可避免的会有偶然因素导致的传输延迟,可以对连续两次收到的同步数据包进行处理以后(如:求平均)只进行一次时钟校正,这样假如原来出现一次偶然的错误数据(假设与正常数据相差了 Δ),那么经过求平均处理以后,误差减少了一半($\Delta/2$)。对于某些精度要求苛刻的系统,这种差异是巨大的。

以上措施,都是根据 PTP 协议的实现原理中的一些可能导致同步误差的地方,提出的相应的处理办法,能够有效的提高同步的精度。通过改进,在稳定的分布式系统中甚至能达到 ns 级同步,这样就特别适合于对时钟同步精度要求苛刻的系统,对于绝大多数工业以太网的运用都不成问题。

4 结语

PTP 协议可以实现微妙级同步并且有着很大的提高空间,相对与传统的时钟同步协议,PTP 协议以其高精度、高稳定性受到越来越多的关注,并开始被广泛应用在各种分布式测试网络中,势必逐渐成为为网络同步市场的主流。在实现的过程中,如果能够按照本文建议的措施进行设计,那么同步结果将更加令人满意,甚至能达到 ns 级,对于绝大多数系统都足以胜任。

参考文献

[1]王燕山等，“以太网时间同步技术研究与应用”，测量和控制技术，2007 年第 26 卷,第 4 期，2-6 页。

[2]David L.Mills, “ Internet Time Synchronization: The Network Time Protocol ”, IEEE TRANSACTIONS ON COMMUNICATIONS, VOL.39,NO. 10,OCTOBER 1991. pp. 1482-1493.

[3]唐惠玲,刘学军，“基于以太网控制系统的实时性分析”，铁道通信信号，第 40 卷第 8 期，2004 年 8 月，13-14 页

[4]Simon, G. et al, “Sensor Network-Based Countersniper System”. The Second ACM Conference on Embedded Networked Sensor Systems (SenSys), November 2004. pp. 111-116

[5]Kopetz, H., and Ochsenreiter, W, Clock Synchron-ization in Distributed Real-Time Systems. IEEE Transactions on Computers, C-36 (8), August 1987. p. 933-939.

[6]Kopetz, H., and Schwabl, W, “ Global time in distributed real-time systems ”. Technical Report, Technische Universitat Wien, 1989. pp. 15-89

[7]Mainwaring, A., Polastre, J., Szewczyk, R., Culler, D., and Anderson, J, “Wireless Sensor Networks for Habitat”,1998, pp. 122-125

[8]Mills, D. L, “ Internet Time Synchronization: The Network Time Protocol ”. IEEE Transactions on Communications COM 39 no. 10, October 1991, pp. 1482-1493.

[9]Schwiebert, L., Gupta, S., and Weinmann, J. “ Research Challenges in Wireless Networks of Biomedical Sensors ”. SIGMOBILE 2001, July 2001, pp. 151-165.

[10]Mark Andrews,Bernd Altmeier etc, http://www.ntp.org/

[11]Dirk S. Mohl, http://www.ieee1588.com/.

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
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

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

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
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

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

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μC-OSⅡ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 NiosⅡ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

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.