

基于 STM32F107VC 的 IEEE 1588 精密时钟同步分析与实现

张小强, 谢鹏程

(华南理工大学 机械与汽车工程学院, 广东 广州 510640)

摘要:随着网络化与分布式系统的应用,对系统各节点间的时钟同步精度要求越来越高,尤其在分布式运动控制中,高精度时钟同步更是一切应用的基础。针对此问题,首先分析了 IEEE 1588 时钟同步的基本原理,并从理论上深入研究了影响时钟同步的关键因素,经过综合分析对比各实现方案的优劣,创新性提出了具有较高性价比的基于 STM32F107VC 的 IEEE 1588 实现方案,通过捕获硬件时间戳和校正频率漂移,在自定义协议中实现了低于 300 ns 的高精度时钟同步。

关键词:IEEE 1588; STM32F107VC; 时钟同步; 时间戳

中图分类号: TN911.7-34

文献标识码: A

文章编号: 1004-373X(2011)13-0172-03

Analysis and Implementation of IEEE 1588 Precision Clock Synchronization Based on STM32F107VC

ZHANG Xiao-qiang, XIE Peng-cheng

(South China University of Technology, Guangzhou 510640, China)

Abstract: A higher precision of clock synchronization is needed with the development of network and distributed system, and especially the high-precision clock synchronization is the foundation of all applications in a distributed motion control system. The basic principle of IEEE 1588 standard is introduced, and the key factors which affect the clock synchronization are studied theoretically. Through comparison of various methods, the best cost-effective implementation based on STM32F107VC microprocessor is decided, and the high precision clock synchronization below 300ns is realized by capturing hardware time stamp and adjusting frequency drift.

Keywords: IEEE 1588; STM32F107VC; clock synchronization; time stamp

0 引言

在分布式系统应用^[1]中,由于各节点物理上的分散性,要保证系统整体的高精度协调运动,时钟同步一直是个关键问题,现有的解决方案中,IEEE 1588 精密时间协议(PTP)^[2]较为突出,但诸多实现方案中,无论是 FPGA 定制^[3]或纯软件实现^[4]或其他方式^[5],均存在成本较高或精度不够的问题,性价比不高,本文通过深入研究其多种实现方案的优劣,提出了现今具有较高性价比的基于 STM32F107VC 的 IEEE 1588 实现方案,极好的满足了分布式系统的应用,尤其在中高端工业控制的分布式应用中,具有较高的市场价值和工程意义。

1 IEEE 1588 的同步原理

该协议以 UDP/IP 为载体进行消息传递,整个系

统的各个节点,按发送或接收时间信息的角色,可分为主节点和从节点。协议通过在主从节点之间交换 PTP 消息来实现精确时钟同步,其基本原理^[6]如图 1 所示。

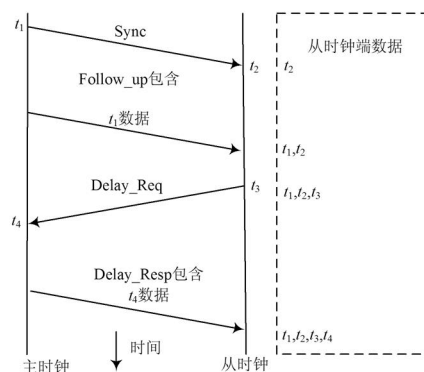


图 1 同步原理

该同步过程分为两个阶段:时钟偏移(Offset)测量阶段和传输延迟(Delay)测量阶段^[7]。

(1) 时钟偏移测量,用于修正主从时钟之间时钟偏移,如图 1 所示:主时钟按定义时间间隔周期性发送

Sync 报文到从时钟,从时钟记录报文到达时间 t_2 ,随后主时钟发送 Follow_up 报文,该报文与之前 Sync 报文相关联,并包含之前 Sync 报文的精确发送时间 t_1 ,由时间关系知:

$$\text{Offset} + t_1 + \text{Master To Slave Delay} = t_2 \quad (1)$$

如不考虑主时钟到从时钟的传输延迟 (Master To Slave Delay),则有:

$$\text{Offset} = t_2 - t_1 \quad (2)$$

依此修正从时钟时间,两时钟同步。

(2) 传输延迟测量,用于计算从时钟到主时钟的报文传输延迟,提高修正精度,如图 1 所示,从时钟按随机时间间隔向主时钟发送 Delay_req 报文,为有效减少网络负荷,该时间间隔大于 Sync 报文发送间隔,从时钟记录报文发送时间 t_3 ,主时钟接收到 Delay_req 报文后,封装 Delay_resp 报文,该报文是对 Delay_req 的响应,并包含 Delay_req 的精确接收时间 t_4 ,依时间关系有:

$$\text{Slave To Master Delay} + t_3 = t_4 + \text{Offset} \quad (3)$$

此时,从时钟包含 t_1, t_2, t_3, t_4 ,假设主从时钟间 (Master To Slave Delay) 和从主时钟间 (Slave To Master Delay) 传输延迟相等,均为 Delay,则有^[8]:

$$\begin{aligned} \text{Delay} &= \frac{(t_2 - t_1) + (t_4 - t_3)}{2} \\ \text{Offset} &= \frac{(t_2 - t_1) - (t_4 - t_3)}{2} \end{aligned} \quad (4)$$

2 影响时钟同步精度因素分析

由式(4)可得:

$$\begin{aligned} t_2 - t_1 &= \text{Delay} + \text{Offset} \\ \text{Offset} &= \frac{(t_2 - t_1) - (t_4 - t_3)}{2} \end{aligned} \quad (5)$$

因时钟同步精度由时钟偏移体现,由式(5)知,IEEE 1588 报文的精确发送和接收时间的测量精度,将直接影响最终的同步精度;同时因时钟同步分阶段完成,在时钟偏移测量阶段,当前周期的 t_3, t_4 并未取得,只能使用前一周期近似值,为更精确地消除传输延迟影响,实际算法中采用式(6),以前一周期的平均延迟代替前一周期地单向延迟估算值 $t_4 - t_3$ 。

$$\begin{aligned} \text{Delay} &= \frac{(t_2 - t_1) + (t_4 - t_3)}{2} \\ \text{Offset} &= (t_2 - t_1) - \text{Delay} \end{aligned} \quad (6)$$

如式(6)所示,因每次在时钟偏移阶段计算时钟偏移量用到的传输延迟值都是前一同步周期地过去状态,故当前传输延迟量越接近前一周期传输延迟量,即传输延迟越确定,则时钟偏移测量越准确,而同步精度越高。

根据 UDP/IP 协议,报文传递需要逐层进行,要减小传输延迟的变化,必须使报文时间戳捕获点尽量接近

物理层^[9],以减少报文层层封装和操作系统多任务处理带来的传输延迟不确定。

3 IEEE 1588 实施方案分析

IEEE 1588 时钟同步的实现方式,常有以下四种:

(1) 利用 FPGA 实现时钟同步功能^[10],此方案同步功能需要 FPGA 执行,其特点在于 FPGA 输出窥探信号,检测物理层 (PHY) 芯片及链路层 (MAC) 芯片之间的 MII 总线的操作情况,以便记录同步协议数据包的封包时间,而封包处理及时间戳的捕获必须由 FPGA 负责,其同步精度取决于 FPGA 对时间戳的处理及物理层 (PHY) 芯片之间的时间延迟。

(2) 利用可支持 IEEE 1588 的微控制器执行时钟同步功能,目前市场上已有不少微控制器支持 IEEE 1588 标准,如 Intel 的 IXP435,意法半导体 (ST) 的 STM32F107VC 等,该时钟同步的精度主要取决于微控制器的设计及以太网物理层 (PHY) 芯片之间的时间延迟。

(3) 利用可支持 IEEE 1588 的以太网物理层芯片执行时钟同步功能,通过改用支持 IEEE 1588 的高精度物理层芯片,如美国国家半导体的 DP83640 芯片,由于利用芯片内部的线路记录封包时间,是目前市场上精度最高的解决方案,但特殊的物理层芯片增加了设计的成本。

(4) 纯软件实现,利用软件系统时间模拟时间戳,虽成本较低,但精度不够。

在分布式系统的应用中,方案(1)~(3)均可实现 200 ns 的高同步精度,但综合考虑性价比,方案(2)最优,微处理器包括所有成本,在现有的支持 IEEE 1588 时钟同步的微处理器中,意法半导体的 STM32F107VC 价格实惠,性能优越,能够实现低成本的网络化和精确的时钟同步,为本系统采用方案。

4 IEEE 1588 功能的实现

4.1 系统时钟初始化

IEEE 1588 协议规定,系统需 64 位格式的基准时间来记录当前时间,分为两个 32 位通道,在 STM32F107VC 中,ETH_PTPTSHR 和 ETH_PTPSLR 这两个寄存器分别保存时间信息的高 32 位和低 32 位,要能使 IEEE 1588 功能,必须初始化系统时间,核心代码如下:

```
// 使能时间戳功能
ETH_PTPTimeStamCmd(ENABLe);
// 写系统时间到时间戳更新寄存器
ETH_SetPTPTimeStamUpdate(0, 0, 0);
```

4.2 时间戳捕获实现

在 PTP 协议的实现中,精确的时间戳是实现高精度时钟同步的基础,在 STM32F107VC 中,当以太网帧的 SFD 从 MII 输出的时候,就会记录下时间戳,但 MAC 并不通过处理帧本身的方式来识别 PTP 帧,而是通过帧描述符里的控制位来控制,记录下的时间戳会与帧的处理状态一起,存放在相应的描述符里,从而自动地把时间戳和帧的处理状态联系起来。

下面就发送时间戳的捕获说明这一机制的实现,在 STM32F107VC 中,网络数据包的发送接收都由 DMA 控制,DMA 描述符是整个控制过程的关键,每个描述符结构体包含四个 32 位字。当描述符使用链式结构时,发送 DMA 描述符结构^[11]如图 2 所示。

OWN	控制	TTSE	保留	控制	保留	TTSE	状态	TDES0
保留		缓冲区 2 B 计算			保留		缓冲区 1 B 计算	TDES1
缓冲区 1 地址[31:0]/时间戳低位[31:0]								TDES2
下一描述符地址/时间戳低位[31:0]								TDES3

图 2 发送描述符

发送数据时,首先使能 IEEE 1588,设置 TDES0 的位 TTSE 为“1”,当一帧传输完成后,DMA 描述符的 TDES2 和 TDES3 会记录时间戳并覆盖 TDES2 和 TDES3 中原

有的缓存地址和下一描述符地址,因此在内存中复制一份当前发送描述符,使 DMA 传输时找到正确的地址是整个驱动实现的关键,核心代码如下:

```
// 原始 DMA 发送描述符
ETH_DMADESCTypeDef DMA TxDescr Tab[ ETH_TX-
BUFNB];
// 备份 DMA 发送描述符
ETH_DMADESCTypeDef DMAPTPTx Descr Tab[ ETH_
TXBUFNB];
```

初始化 DMA 发送描述符时,将 DMA TxDescr Tab 的 TDES2 及 TDES3 中数据复制到 DMAPTPTx Descr Tab 里,在传输过程中,当时间戳标志位 TTSS 置 1 标示记录下发送时间戳后,再从 DMAPTPTx Descr Tab 中恢复缓存地址和下一描述符地址,从而为 DMA 传输指向正确的地址,成功进行下一次传输。

5 实验设计与分析

实验采用两块 STM32F107VC 开发板和自实现的 IEEE 1588 代码,开发板直接相连,STM32F107VC 与计算机通过 RS 232 相连,由串口输出同步信息。

当同步周期为 2 s 时,对比是否采用 Follow_up 报文用于同步的同步效果;当同步周期为 3 s 时,采用 Follow_up 报文用于同步的同步效果,时钟偏移值(Offset)如表 1 所示,不含第 1 列偏移值数据的折线图如图 3 所示。

表 1 时钟偏移值 ns

序号	1	2	3	4	5	6	7	8	9	10	11
T=2 s 有 Follow_up	- 82 553	- 54	- 97	- 54	- 97	- 97	- 97	- 140	- 140	- 183	- 140
T=2 s 无 Follow_up	- 1 141 325	- 1 430	- 312	- 269	- 2 290	- 2 204	- 2 204	- 2 118	- 2 204	- 2 161	- 2 075
T=3 s 有 Follow_up	- 79 073	- 570	- 570	- 527	- 527	- 527	- 527	- 570	- 1 516	- 1 473	- 1 559

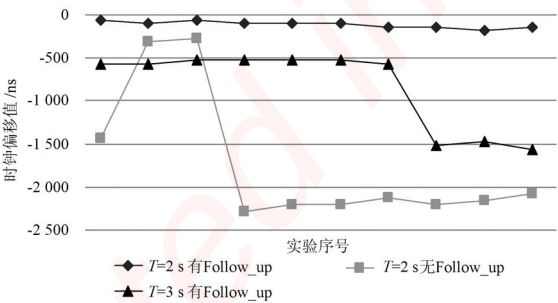


图 3 实验数据对比

图 3 中纵坐标为时钟偏移值,单位为 ns,横坐标为依时间进行的实验序号。对比分析图 3 中实验数据,当同步周期为 2 s 时,采用 Follow_up 报文进行时钟同步的精度明显高于仅用 Sync 报文同步的精度,其关键原因在于,Sync 报文仅提供自身发送的估计时间,而非准确的硬件时间戳,报文封装及传递带来了传输延迟的变

化,故折线的拐变大,而 Follow_up 中含 Sync 报文发送的精确硬件时间戳,消除了软件带来的传输延迟变化。

对比同步周期分别为 2 s 和 3 s,采用 Follow_up 报文用于时钟同步的精度可知,同步周期越短,同步精度越高,但周期减小时,会增大网络负担,故一般取同步周期为 2 s。

6 结 语

本文通过对 IEEE 1588 的深入分析,针对分布式应用中的时钟同步关键问题,创新性提出了性价比较高的解决方案,通过利用 STM32F107VC 的硬件 IEEE 1588 功能,实现了完整的 PTP 协议栈及小于 300 ns 时钟同步,极好地满足了工业控制中分布式应用,具有较高的市场价值与理论及工程意义。

4.3 软件调试

软件调试以子程序为单位逐个进行,最后结合硬件实时调试。

子程序调试包括:DS1302 的计时和读写程序、显示程序;SPCE061A 读写程序;DS18B20 读出温度子程序、温度转换命令子程序;计算温度子程序、显示数据^[9]。

4.4 测试结果分析

测试结果见表 1,表 2。

表 1 温度测试

序号	室温表 /	测试数据 /
1	22	22.4
2	21	21.3
3	25	25.2
4	27	27.3

表 2 功耗测试

要求功耗 / μW	静态功耗 / μW	工作功耗 / mW
3	3.0	5.6

由于采用了 DS1302 作为计时器使用,其计时精度相对来说比较高。用制作的成品与万年历计时比较,基本没有误差。

由于 DS18B20 的误差指标在 0.5 以内,在一般场合完全适用。

功耗测试结果表明,静态功耗达到了设计的要求。

作者简介:陈世夏 男,1957 年出生,副教授,教员。主要从事电子技术教学与研究。

(上接第 174 页)

参 考 文 献

- [1] 翟诺,赵琳,申敏. 工业以太网在工业控制中应用的探讨[J]. 信息技术,2009(6):159-162.
- [2] IEEE Std. 1588-2002, IEEE standard for a precision clock synchronization protocol for networked measurement and control systems [S]. 2002.
- [3] 谢型果. IEEE 1588 时钟同步报文硬件标记研究与实现[D]. 武汉:华中科技大学,2008.
- [4] 尤志坚. IEEE 1588 协议分析与实现[D]. 上海:华东师范大学,2009.
- [5] 吴敏凉,石旭刚,张胜,等. 基于 IEEE 1588 的同步以太网实现方式[J]. 单片机与嵌入式系统应用,2010(1):38-40.

5 结 论

本系统由于采用了凌阳 SPCE061A 单片机和 DS18B20 等低功耗器件,经过测试,系统静态功耗低于给定的最小值,且实现了对温度值的显示和处理,并实现了对温度的控制,时间显示准确。系统的性价比较高,有较好的推广应用价值。

参 考 文 献

- [1] 李朝青. 单片机原理及接口技术[M]. 3 版. 北京:北京航空航天大学出版社,2005.
- [2] 沙占友. 智能化集成温度传感器原理与应用[M]. 北京:机械工业出版社,2002.
- [3] 张洪润,傅瑾新. 传感器应用电路 200 例[M]. 北京:北京航空航天大学出版社,2006.
- [4] 吴金戌. 8051 单片机实践与应用[M]. 北京:清华大学出版社,2002.
- [5] 全国大学生电子设计竞赛组委会. 全国大学生电子设计竞赛获奖作品汇编[M]. 北京:北京理工大学出版社,2006.
- [6] 翟志华,王留奎. 嵌入式脉象采集仪电路设计[J]. 现代电子技术,2010,33(10):39-42.
- [7] 蒋焕文. 电子测量[M]. 北京:高等教育出版社,2005.
- [8] 郑润娜,胡建民,侯丽娟. 基于 TLC5620 信号发生器的设计与实现[J]. 电子测量技术,2009(6):102-104.
- [9] 王振辉. 基于 XML 的 Web 数据交换系统设计与实现[J]. 现代电子技术,2010,33(10):99-100.
- [10] 董军堂,宋永东,李建新. 基于 SPCE061A 的智能温度计设计[J]. 现代电子技术,2010,33(19):141-143.

- [6] 方强. PTP 技术分析[J]. 无线电工程,2010,40(2):61-62.
- [7] 王海燕. IEEE 1588 精确授时协议通用实现方案[J]. 电脑知识与技术,2009(29):8131-8132.
- [8] 王晓冬,阚德涛,张志武. 以太网的时钟同步技术[J]. 电子工程师,2008,34(9):47-50.
- [9] 方菁. 基于 FPGA 的电力系统时钟同步技术设计[J]. 武汉职业技术学院学报,2010,9(1):75-78.
- [10] 孔令彬,文赫胜,陈向文. IEEE 1588 精密时钟同步关键技术研究[J]. 计算机测量与控制,2010,18(7):1585-1586,1598.
- [11] 意法半导体(中国)投资有限公司. STM32F107xx reference manual. pdf [EB/OL]. [2010-01-10]. <http://www.st.com/stoneline/products/literature/ds/15274.pdf>.

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](#)
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](#)

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 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](#)

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 技术研究与实现](#)
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.