

# 基于 PCI 总线的 GPS 授时卡设计

刘军良<sup>1,2</sup> 胡永辉<sup>1</sup> 侯雷<sup>1,2</sup>

(1 中国科学院国家授时中心, 陕西 临潼 710600)

(2 中国科学院研究生院, 北京 100039)

**摘要:** 为了提高计算机本地时钟的精度, 介绍了一种以协调世界时修正计算机本地时钟的设备——基于 PCI 总线的 GPS 授时卡。采用了基于 PCI 总线的设计方案, 阐述了 PCI 总线的驱动程序的设计。最后采集分析观测数据, 结果表明授时卡对计算机的授时精度小于 1ms。

**关键词:** 协调世界时; 总线; PCI; GPS; WDM;

## Design of GPS Timing Card Based on PCI Bus

Liu Junliang<sup>1,2</sup> Hu Yonghui<sup>1</sup> Hou Lei<sup>1,2</sup>

(1 National Time Server Center, the Chinese Academy of Sciences, Lintong, Shaanxi 710600, China)

(2 The Graduate School of the Chinese Academy of Sciences, Beijing 100039, China)

**Abstract:** In order to improve the accuracy of the computer system clock, the device - GPS timing card based on PCI Bus is introduced that can Proofread the local clock of a computer by Coordinated Universal Time. PCI bus design scheme was used and the design of PCI bus drivers was described. Finally, observing data was sampled and analyzed and the results of the analysis show that the timing precision of the timing card on a computer is less than 1 ms.

**Keywords:** Coordinated Universal Time; Bus; PCI; GPS; WDM;

随着计算机和计算机网络在计量测试, 通信和电子电力等领域地广泛应用, 对计算机处理数据时的时间准确度要求越来越高, 计算机的本地时钟的准确程度直接关系到应用系统的性能。而没有经过修正的计算机的本机时钟是由计算机主板上的时钟芯片提供的, 由于该时钟芯片的稳定度差, 准确度低, 所以计算机长期工作会使得计算机本地时钟与 UTC (协调世界时) 时间产生较大偏差。为了使计算机的本地时钟达到更高的精度, 本文介绍了一种授时设备——基于 PCI 总线的 GPS 授时卡。通过实验分析, 表明 PCI 授时卡 (基于 PCI 总线的 GPS 授时卡简称为 PCI 授时卡) 的授时精度小于 1ms。

## 1 概述

GPS 是 Global Positioning System 的英文缩写, 即全球定位系统。该系统向具有适当接收设备

的全球范围内的用户提供精确、连续的三维位置和速度信息<sup>[2]</sup>。GPS 还广播一种形式的协调世界时 (UTC)<sup>[2]</sup>。所以 PCI 授时卡采用了 GPS 信号作为时钟源。

PCI 是 Peripheral Component Interconnect 的英文缩写, 即外围部件互联。PCI 总线是一种具有多路地址线 and 数据线的高性能的 32/64 位总线<sup>[1]</sup>。它工作于 33MHz 的时钟频率, 支持 32 位的数据传输, 最高传输速率可达 132MB/s。传统的 ISA 总线的最大传输速率只有 8MB/s, 并且目前绝大多数 PC 机主板提供 PCI 插槽而已不再提供 ISA 总线插槽。由于以上问题, 本文介绍的 PCI 授时卡采用了基于 PCI 总线设计方案。

PCI 授时卡主要功能包括提取和处理由 GPS 接收模块发送来的时间信息, 并在 PCI 授时卡上维持一个高精度的本地时间, 最后通过 PCI 总线传递给计算机。本文主要介绍 PCI 授时卡的硬件系统设计和 PCI 总线驱动程序设计。

## 2 硬件系统设计

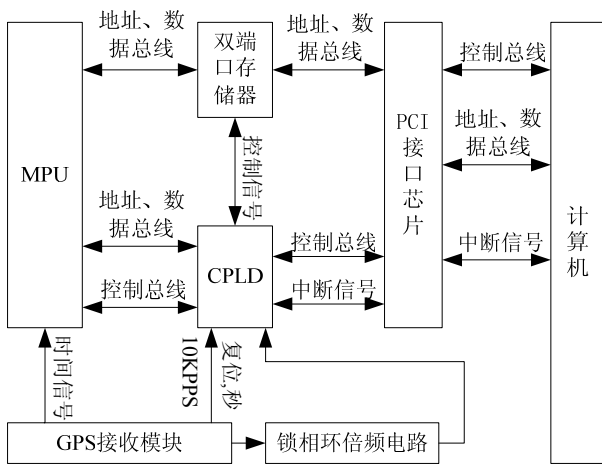


图 1 GPS 授时卡硬件组成框图

如图 1 所示，GPS 授时卡的硬件主要是由微处理单元 MPU、可编程器件 CPLD、PCI 接口芯片、GPS 接收模块和锁相环倍频电路组成。

GPS 接收模块采用了 ROCKWELL 公司的 OEM 模块，工作于 C/A 码，该 OEM 模块具有集成度高、体积小、功耗小、工作稳定等特点。该模块输出时间信号通过串口送至 MPU 进行时间信息处理，而它产生的 10KPPS（PPS 为每秒脉冲数）信号进入锁相环倍频电路。

锁相环倍频电路负责将 10KPPS 信号与 GPS 秒脉冲信号的上升沿不断的对齐，并且倍频输出 20MPPS 信号，该信号作为 CPLD 的时钟源。

MPU 主要完成对板卡的控制和对时间信息的处理。当 MPU 确认可以接收时间信息时，将由 MPU 的中断子程序解调来自于 GPS 接收模块的数据，接着将解调后的 UTC 时间转换为北京时间，最后向双端口存储器发送时间数据。以使计算机在任意时刻都能读取正确的时间。

MPU 同时还负责将时间数据缓存区的数据与 CPLD 产生的时间进行比较、判别。实现时间数据的纠错、保护。

CPLD 由锁相环电路提供的 20MPPS 信号进行分频设计，产生毫秒脉冲。由于锁相环电路输出的 20MPPS 信号与秒脉冲信号具有严格的相位关系，所以板上向主机提供的毫秒计数是由 MPU 的计数器通过读取 CPLD 的毫秒脉冲数得到的。

另外，GPS 接收模块送入的秒起点（秒脉冲）信号送入 MPU 和 CPLD 作为 GPS 授时卡的秒中断信

号。该秒中断作为产生各种频率信号的复位信号同步设在 CPLD 内的各分频链，以使产生的各频率信号与秒起点相关。

PCI 接口芯片主要是负责 MPU 和计算机之间的数据通信。由于 PCI 总线协议比较复杂，本文的设计使用了 PLX 公司生产的专用 PCI 接口芯片 PCI9052，这提高了设备的可靠性，也使得设计更加的容易，大大缩短了开发周期。PCI9052 芯片为 GPS 授时卡提供了一个高性能的 PCI 总线目标（从）接口。通过编程，PCI9052 可以直接连接到复用或者非复用的 8 位，16 位或者 32 位本地总线<sup>[1]</sup>。

PCI9052 的一个主要结构特点是包含一个 ISA 逻辑接口，它提供了 ISA 到 PCI 的平滑转换。它支持 8 位和 16 位宽的 ISA 从设备，同时也支持内存映射和 I/O 映射。因此这个逻辑接口使得 ISA 设计向 PCI 设计转换更加容易。本文的设计便是使用 PCI9052 的 8 位宽的 ISA 接口模式，图 3 展示了 PCI9052 的 ISA 模式的信号接口示意图。

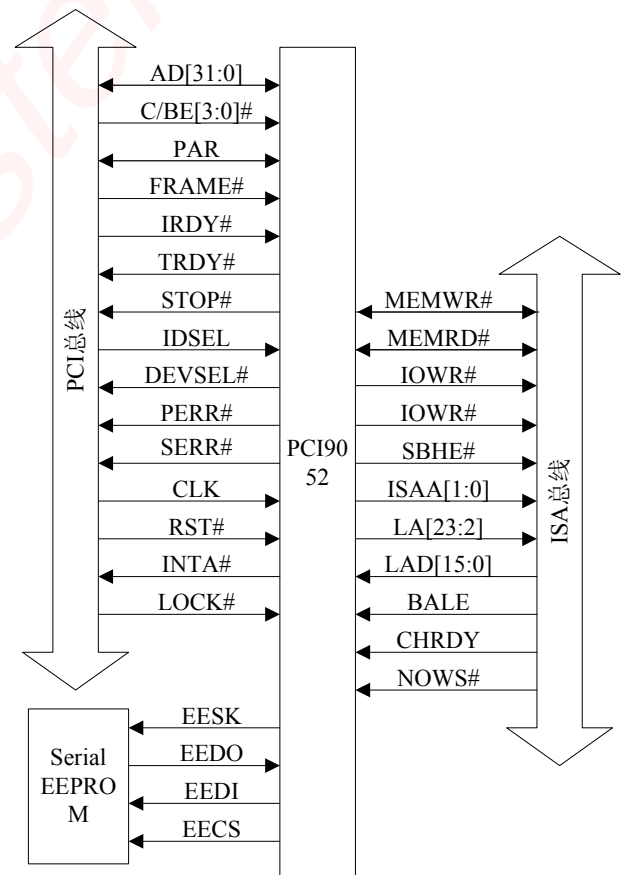


图 2 PCI9052 的 ISA 接口模式的信号接口示意图

如图 2 所示，左下角的 Serial EEPROM 芯片用于配置 PCI9052 芯片内部的配置寄存器。该寄存器

的内容直接关系到驱动程序的编写，应该小心配置。

在与 ISA 总线相连接的信号中，MEMWR#和 MEMRD#信号是内存映射方式下的内存读写信号，而 IOWR#和 IORW#信号是 I/O 映射方式下的读写信号，本文的设计使用的是内存映射方式。

在 8 位宽 ISA 模式中，LA[23: 2]和 ISAA[1: 0]共同组成地址线，其中 ISAA[1: 0]为地址的底两位。

LAD[8: 0]为设计中的数据线。

BALE 为地址锁存信号。

CHRDY 为图三中 PCI9052 右边通道选通信号。

NOW#为减少等待时间信号。

### 3 驱动程序设计

本文的设计是在 Windows XP 环境下实现 GPS 授时卡的 WDM 设备驱动程序的。WDM 驱动程序是指遵从 Windows 驱动程序模型 (WDM, Windows Driver Model) 的设备驱动程序<sup>[3]</sup>。WDM 包含了对 Windows 电源管理、即插即用和 WMI 的支持，而且，绝大多数即插即用驱动程序都遵从 WDM 规范<sup>[3]</sup>。该驱动程序的开发工具使用的是微软公司提供的 Windows DDK (Driver Development Kit)。下面主要从 WDM 设备驱动程序的初始化例程，驱动程序提取资源，读写数据 and 应用程序接口几个方面说明 GPS 授时卡的驱动程序设计。

#### 3.1 初始化例程

初始化例程通常被命名为 DriverEntry，它填充一些数据结构，完成驱动程序的全局初始化工作。它是 GPS 授时卡驱动程序的入口点。

```
extern "C"
NTSTATUS DriverEntry(IN PDRIVER_OBJECT
    DriverObject,
    IN PUNICODE_STRING RegistryPath)
{
    NTSTATUS status = STATUS_SUCCESS;
    ...
    DriverObject->DriverExtension->AddDevice
= GPSAddDevice;
    DriverObject->DriverStartIo=GPSStartI;
    DriverObject->DriverUnload=GPSUnload;
    DriverObject->MajorFunction[IRP_MJ_CREAT
E]= GPSCreate;
```

```
    DriverObject->MajorFunction[IRP_MJ_CLOSE]
= GPSClose;
    DriverObject->MajorFunction[IRP_MJ_PNP]
= GPSPnp;
    DriverObject->MajorFunction[IRP_MJ_POWER]
= GPSPower;
    DriverObject->MajorFunction[IRP_MJ_READ]
= GPSRead;
    DriverObject->MajorFunction[IRP_MJ_WRITE]
= GPSWrite;
    DriverObject->MajorFunction[IRP_MJ_DEVIC
E_CONTROL]=GPSDeviceControl;
    DriverObject->MajorFunction[IRP_MJ_SYSTE
M_CONTROL]=GPSSystemControl;
    ...
    return status;
}
```

该例程的成员变量 DriverObject 称作驱动程序对象，代表了系统中的一个驱动程序，这里代表 GPS 授时卡的驱动程序。Windows 操作系统 I/O 管理器从该驱动程序对象获得所有分发例程（一个设备驱动程序提供的主要函数）的地址。由上面的代码可以直接看出 DriverObject 的结构成员指向相应的分发例程或函数。

GPSAddDevice 例程是 GPS 授时卡驱动程序的增加设备例程。每当 Windows 操作系统的即插即用管理器检测到 GPS 授时卡时，它通过增加设备例程向驱动程序发送一个通知。在这个例程中创建了代表 GPS 授时卡的设备对象 (device object)。

GPSUnload 例程是 GPS 授时卡驱动程序的卸载例程，它用于卸载驱动程序用到的任何系统资源。I/O 管理器会在卸载驱动程序前调用该例程。

GPSStartIo 例程是 GPS 授时卡的启动 I/O 例程。它主要是对来自 Windows 应用程序的 I/O 请求进行排队。

GPSCreate、GPSClose、GPSRead、GPSWrite 和 GPSDeviceControl 分发例程分别用于处理来自 Windows 应用程序的打开、关闭、读、写和 IOCTL 请求。

GPSPnp、GPSPower 和 GPSSystemControl 例程分别于 GPS 授时卡的即插即用、电源管理和 I/O 请求的传递有关。

#### 3.2 提取资源以及读写数据

GPS 授时卡驱动程序提取的资源是存放在 PCI9052 寄存器中的资源信息。当计算机启动时, EEPROM 首先会将这些信息配置到 PCI9052 芯片中, 进入 Windows 操作系统后, 操作系统的 Pnp 管理器会自动检测到 GPS 授时卡, 并提取该卡的 I/O 资源需求。驱动程序要做的事情就是接收 Pnp 管理器的消息, 并将该消息中有用的资源信息存放到驱动程序的全局变量中去。

由于 GPS 授时卡使用了内存映射方式, 从 GPS 授时卡传来的年、月、日、时、分、秒和毫秒信息作为了计算机的内存资源。在驱动程序中从内存资源中读写时间信息, 必须获取内存范围基址, 然后根据预先定义的地址偏移量求得该时间信息存放的内存映射地址。使用内存访问函数 READ\_REGISTER\_UCHAR 按字节从内存映射地址读取时间信息, 它将时间信息复制到 I/O 请求包指向的数据缓冲区以便应用程序读取。

### 3.3 应用程序接口

GPS 授时卡需要 Windows 应用程序访问, 而访问该卡的应用程序需要先知道设备采用的命名方案。WDM 引入了一个新的设备命名方案, 该方案依靠一个设备接口 (device interface) 的概念, 它基本上是软件如何访问硬件的一个说明, 一个设备接口被一个 128 位的 GUID 唯一标识<sup>[5]</sup>。Windows 操作系统自带了生成 GUID 的工具 GUIDGEN。

在编写 Windows 应用程序时, 首先根据在驱动程序中定义的 GUID, 使用 SetupDiEnumDeviceInterfaces 和 SetupDiGetDeviceInterfaceDetail 函数分别枚举设备接口和获得 GPS 授时卡的设备接口信息, 接着根据获得的设备接口信息使用 Windows 的 API 函数 CreateFile 向驱动程序发送 IRP\_MJ\_CREATE 打开设备请求, 驱动程序会调用与之相应的 GPSCreate 分发例程, 并打开设备。

应用程序打开 GPS 授时卡后, 可以使用 Windows 的 API 函数 DeviceIoControl 函数读取时间信息。DeviceIoControl 函数的第二的参数是在驱动程序中定义的 IOCTL 请求码, 即 IOCTL\_GPS\_GET\_TIME。当应用程序调用 DeviceIoControl 函数时, 应用程序向驱动程序发送 IRP\_MJ\_DEVICE\_CONTROL 请求, 驱动程序则调用 GPSDeviceControl 分发例程并执行与 IOCTL\_GPS\_GET\_TIME 的操作,

即驱动程序将使用 READ\_REGISTER\_UCHAR 读取时间信息。

在应用程序退出时, 使用 Windows 的 API 函数 CloseHandle 关闭 GPS 授时卡。

## 4 测试结果分析

本文设计的测试平台主要包括 IBM 6824 微型计算机, Window XP SP2 专业版操作系统, 安捷伦 53131A 计数器, GPS 授时卡, 时间基准源, Matlab6.5。其中时间基准源 (为一台 GPS 接收机) 的授时精度小于 100 纳秒。

本文设计的测试方法如下:

首先编写了一个并口驱动程序, 它主要功能是从计算机并口输出数据。可以编写 Windows 应用程序先通过 GPS 授时卡驱动程序读取秒信号, 然后根据该秒信号通过并口驱动程序实现每秒种向并口的 2 管脚输出秒信号。将该并口秒脉冲输出接入安捷伦 53131A 计数器的通道一, 将来自时间基准源的秒输出信号接入安捷伦 53131A 计数器的通道二。安捷伦 53131A 计数器提供串口输出, 通过串口将两个秒信号的差发送到计算机。图 3 为测试平台的连接框图。

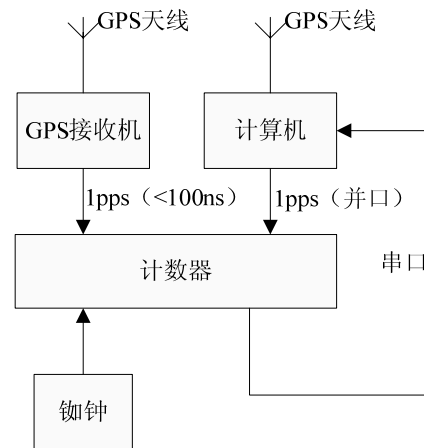


图 3 测试平台连接框图

经过近 16 个小时连续采样, 其原始采样数据如图 4 所示。显然, 部分采样点的秒信号误差很大, 这是由于在实际测量中数据采样过程受到外部环境和计算机状态的不稳定性的影响, 会产生异常值。

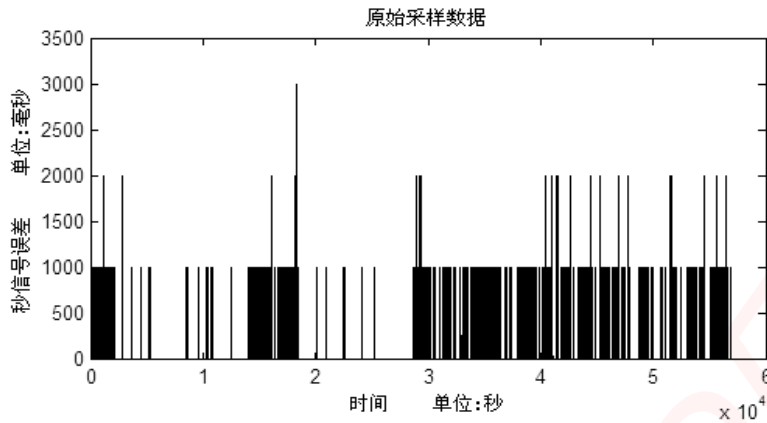


图 4 原始采样数据

本文设计通过使用拉布斯准则 ( $3\sigma$  准则) 剔除异常值, 其数据如图 5 所示。其均值为 0.80155 毫秒, 均方差为 0.14585 毫秒。除了 4738 点异常值, 其数据如图 5 所示。其均值

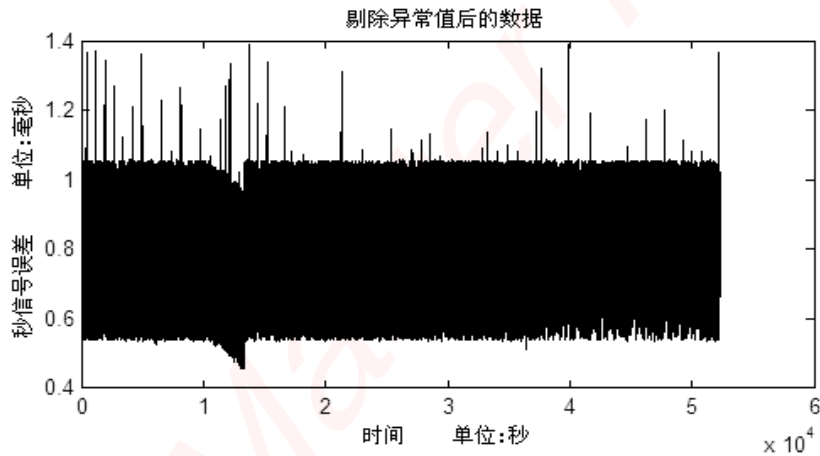


图 5 剔除异常值后的数据

通过观察图 5 可发现秒信号误差中存在固定偏差, 可以通过 GPS 授时卡将其消除。将固定偏差设为 0.80155 毫秒, 并从秒信号误差去处该固定偏差, 则数据如图六所示。

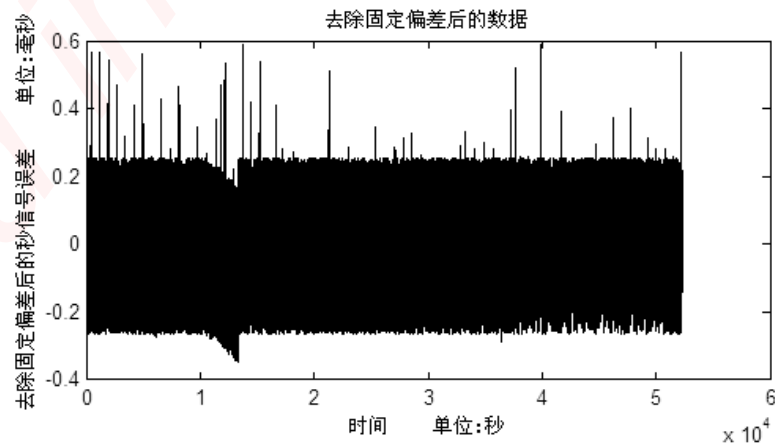


图 6 去除固定偏差后的数据



如图 6 所示,所有的秒信号误差小于 1 毫秒。

## 6 结束语

根据本文介绍的设计方法研制的 GPS 授时卡已经过相关用户地使用证明,该卡集成度高,灵活性强,安装方便,授时精度高,能够满足众多客户的需求。

### 参考文献:

- [1] 李贵山,陈金鹏.PCI 局部总线及其应用[M].西安:西安电子科技大学出版社[M], 2003.
- [2] ElliottD.Kaplan,Christoper J.Hegarty 著,寇艳红译.GPS 原理与应用[M],第二版.北京:电子工业出版社,2007.
- [3] Mark E.Russinovich,David A.Solomom 著,潘爱民译.深入解析 Windows 操作系统[M],第四版.北京:电子工业出版社,2007.
- [4] PLX Tecnology,Inc.PCI9052 Data Book, Version2.2001.
- [5] WalterOney.Programming The Microsoft Win-dows Driver Model[M].Microsoft Press.1999.

### 作者简介:

**刘军良:** 1981 年出生,现就读于中国科学院研究生院,硕士研究生。主要研究方向是无线通信领域的智能化仪器仪表。

**胡永辉:** 1952 年出生,首席研究员,博士生导师,国际天文学学会会员。主要研究方向是多模多通道共用接收机。

**侯 雷:** 1980 年出生,助理研究员,博士研究生。主要研究方向是无线通信和计算机领域的授时、定时技术。

# 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](#)

## 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](#)

## 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](#)

## 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\[可实现读写的源码\]](#)

## 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](#)

## 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](#)

## 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](#)