

基于嵌入式的 Tilcon 用户图形界面设计与开发

韩改宁¹, 李永锋²

(1. 咸阳师范学院 陕西 咸阳 712000; 2. 咸阳 202 研究所 陕西 咸阳 712000)

摘要:针对嵌入式应用领域下的实时高质量图形开发困难的问题,引入一种嵌入式下图形开发工具 Tilcon,并对 Tilcon 的程序开发方法和流程进行深入分析和研究。在此结合 VxWorks 系统,利用 Tornado 开发环境,WindML 图形库组件层次结构的配置,在此基础上设计开发智能水表的图形实例,实现嵌入式系统下图形设计与开发。该设计是基于三方嵌入式软件进行联合开发,解决了嵌入式 VxWorks 下的图形显示与实时刷新困难的问题,再结合相应的硬件,具有很大工程使用价值。

关键词:嵌入式系统; Vxworks; Tornado; Tilcon; 图形界面设计

中图分类号: TP311

文献标识码: A

文章编号: 1004-373X(2009)16-051-03

Design and Development of Embedded Graphical User Interface Based on Tilcon

HAN Gaining¹, LI Yongfeng²

(1. Xianyang Normal University, Xianyang, 712000, China;

2. 202 Institute of Mechanical & Electrical Engineering, Xianyang, 712000, China)

Abstract: According to developing difficulty of real-time and superior quality graphics interface in embedding application domain, Tilcon which is an embedded graphics developing tool is introduced, the method and process of the program development about Tilcon are analysed and researched. Combining VxWorks operating system, making use of Tornado environment and configuring WindML, the article designs a graphics example on intelligent water, and achieves development of embedded graphics. The design is developed based on three softwares union, and solves the problem of real-time graphics display and update based on Tilcon in VxWorks. It has important engineering practical value according to suitable hardware.

Keywords: embedded system; Vxworks; Tornado; Tilcon; graphical interface design

Tilcon 实时开发环境是一种多平台用户界面开发工具,用来开发健壮、高度交互性的用户界面,用户运用它能够快速、方便地开发出令人满意的图形界面应用程序,所开发的程序不需要修改代码就可以从一个操作系统移植到另一个操作系统之上^[1]。

VxWorks 是高性能嵌入式实时操作系统。它以良好的可靠性和卓越的实时性被广泛地应用在通信、军事、航空、航天等高精尖技术及实时性要求极高的领域^[2]。为满足复杂图形界面用户需求,基于 VxWorks 嵌入式操作系统可以使用 Tilcon 图形开发工具,在 Tornado 环境下配置图形开发库 WindML,而 Tilcon 集成大量成熟控件,无需调用低级的图形函数去画线填充,用拖动控件的方式构造自己的图形应用,极大地简化了开发过程。

基于上述各个软件工具的开发特点,提出一种基于嵌入式图形开发的方案,该开发方法适用于嵌入式系统

WindRiver VxWorks,利用 Tilcon 与 Tornado 集成开发环境之间的无缝联接,实现了嵌入式实时操作系统下图形的开发。

1 VxWork, Tornado 及 Tilcon 之间的层次关系

Tornado 就是为开发 VxWorks 应用系统提供的集成开发环境, Tornado 中包含的工程管理软件,可以将用户自己的代码与 VxWorks 的核心有效地组合起来,可以按用户的需要裁剪配置 VxWorks 内核; VxSim 原型仿真器可以让程序员不用目标机的情况下,直接开发系统原型,做出系统评估;功能强大的 CrossWind 调试器可以提供任务级和系统级的调试模式,可以进行多目标机的联调;优化分析工具可以帮助程序员从多种方式真正地观察、跟踪系统的运行,排除错误,优化性能。

VxWorks 在此指的是主机上的 Tornado 集成开发环境编译生成的目标机上可执行代码 IMAGE,即是可裁剪的 VxWorks 操作系统镜像文件^[3]。它与 DOS 的 *.EXE 文件类似,可被静态或动态链接,在开发阶段可动态地下载目标文件,并与其他目标文件动态链接;

在成品阶段是采用静态链接的,产品上电后立即与其他目标文件链接。

Tornado 是一套强大的图形化嵌入式集成开发环境,它的开发系统包括三个高度集成的部分:一是,运行在主机和目标机上的功能强大的交叉开发工具和实用程序;二是,运行在目标机上的高性能、可裁剪的实时操作系统 VxWorks;三是,连接主机和目标机的多种通信方式,如串行口、以太网、仿真器或其他通信手段相联系^[4]。

Tilcon 实时开发环境是一种多平台用户界面开发解决方案,可以用来开发健壮的、高度交互性的用户界面。Tilcon 开发系统包括三个基本组成部分:Tilcon 图形引擎,平台无关的 API 和一个可视化图形界面设计工具——Tilcon 图形编辑器^[5]。

在 Tornado 开发环境下,首先根据自己的应用和目标机的需要,对 VxWorks 模块进行配置,包括:C++ 模块、图形模块、目标机硬件模块、操作系统模块等;其次选择 VxWorks 镜像类型,包括 RAM,ROM,Rom-Compress 等;再次构造 VxWorks 镜像,最后引导 Vx-Works 镜像,生成目标加上所需的 VxWorks 镜像文件。在 Tilcon 开发环境下实现图形界面的设计,包括拖拉所需的控件及各个控件的驱动代码,然后生成图形库文件;再根据 Tornado 与 Tilcon 之间的无缝连接关系,将 Tilcon 生成的图形库文件加载到 Tornado 环境中,从而实现嵌入式下的图形开发。它们的层次关系如图 1 所示。

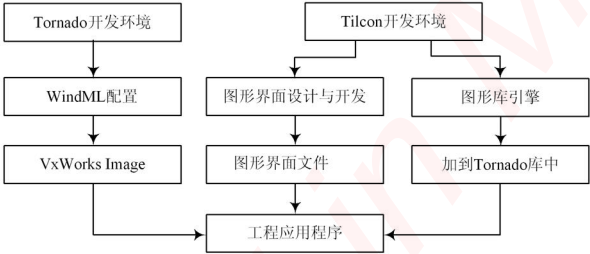


图 1 层次关系图

2 嵌入式图形开发步骤

2.1 Tilcon 系统环境配置

在进行图形设计之前,将嵌入式系统开发环境配置好,首先要在 VxWorks 实时操作系统下正确运行 Tilcon 矢量图形引擎,必须在 Image 镜像文件中加载几种组件支持,即:组件 C++ Components 所有子项、WindML 媒体库组件^[6]。其次,IDS 应用程序模型需要 tlncore.o 和 tlnapi.o 目标静态库文件支持,它们可以通过 Tilcon 附带提供的 Scalable.exe 可执行文件来生成,在运行 Scalable.exe 之前还需选择好处理器类型以

及 VxWorks 版本信息并保证 Tilcon 能准确定位到目标机文件信息。再次,在 Tornado 的 Downloadable 工程项目中配置编译环境 C/ C++ Compile 和 Micros。最后,在此工程项目中添加应用程序。

2.2 Tilcon 运行机理

首先调用 TRT_Start 应用程序接口函数,此时打开 VxWorks 与 Tilcon 之间的信息通道,然后调用 TRT_WindowLoad 函数开始 Tilcon 图形界面窗口的初始化操作,并加载和显示窗口文件 ID;随后调用 TRT_WindowDisplay 函数,显示刚才加载的图形窗口及其相关的子窗口和对话框;接着进入主框架循环区域,同时 TRT_GetInput 函数处于等待状态并实时接收来自于 Tilcon 引擎所传递的界面操作事件,将其通过条件语句结构分发给相应的处理子函数,即用户的具体事件处理函数。如果接收到终止信息则调用 TRT_WindowDelete 清除图形用户界面用于释放内存,最后调用 TRT_Exit 函数完成退出操作。具体操作流程如图 2 所示。

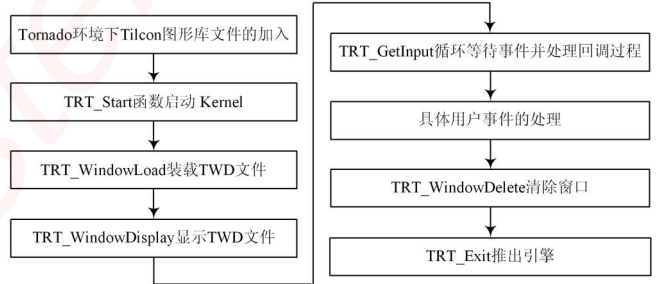


图 2 嵌入式开发流程图

3 示例开发

一个用 Tilcon 工具开发的智能水表软件界面开发实例。利用 Tilcon 在嵌入式 VxWorks 系统下,结合 Tornado 工具一起开发的图形软件,按照嵌入式开发流程图的机理实现嵌入式系统下的智能水表图形界面开发,基本框架程序如下,界面图形如图 3 所示。

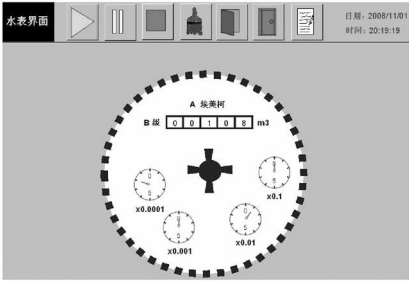


图 3 智能水表界面设计

该实例创建了一个水表窗口,有四个表盘和表盘控制操作按钮及水表数据的显示等控件。该软件与智能水表电路板可以实现真正意义上的嵌入式图形开发和

控制,具有实际应用价值。

Tilcon 用户程序的基本框架及其主要代码如下:

```
# include < time. h >
# include < stdlib. h >
# include < stdio. h >
# include < string. h >
# include < tilcon/ TRTConst. h >
# include < tilcon/ TRTAPI. h >
# include < tilcon/ TRT API. h >
# define CC TRT VXWORKS
// 主文件名称
# define main window file " ICWATER. TWD "
# define main window id " ICWATER ID "
# define APPNAME " ICWATER "
# define USERPROG " ICWATER "
// ID 资源
Pid t TRT cid; // Identifies Tilcon Runtime
TRT ReceiveData rec_data; // receive data
void ExitBox(char * ErrorMessage)
{
    TRT MessageBox ( TRT cid, NULL, APPNAME, ErrorMessage, NULL, "OK", NULL,
        TRT MessageBox Font Helvetica, 14,
        TRT MessageBox Exclamation| TRT MessageBox Wait,
        0, NULL, NULL);
    TRT Exit (TRT cid);
    exit(0);
}
// 主程序
int main(int argc, char argv[])
{
    long errorcode = 0;
    int c;
    long running;
    int ContinueLooping = TRUE;
    TRT StartData StartData;
    // 设定系统环境
    #ifdef CC TRT VXWORKS
    StartData. OsEnv = TRT VXWORKS;
        // The operating system this app is running under.
    #endif
    StartData. Display = NULL;
    StartData. IPAddr = NULL;
    StartData. Userprog = APPNAME;
    StartData. AppName = USERPROG;
    StartData. Flags = NULL; // For now set to NULL
    running = TRT Start ( &StartData);
    // 启动 Tilcon 图形引擎
    errorcode = TRT Start ( &StartData);
    TRT cid = StartData. TRT CID;
    if(errorcode)
    {
        printf("Cannot Start Runtime\n");
        exit(0);
    }
    // 加强所有 API 命令同步
    errorcode = TRT Debug(TRT cid, 3);
    if(errorcode) ExitBox("Cannot run TRT Debug");
    // Setup timer hints for 1/10th second
    errorcode = TRT TimerHintEnable(TRT cid, 2);
        // 2 x50 ms//
    if(errorcode) ExitBox("Cannot start timerhints");
    // 装载 Main Window
    errorcode = TRT WindowLoad (TRT cid, main window
file);
    if(errorcode) ExitBox("Cannot load the Main Window! File
NOT found");
```

```
// 获取 the Window ID
errorcode = TRT GetWindowID (TRT cid, main window
id);
if(errorcode) ExitBox("Cannot Get Main Window ID");
// 显示 Main Window
errorcode = TRT WindowDisplay (TRT cid, main window
id);
if(errorcode) ExitBox("Cannot Display Main Window");
// 主循环, Wait for communication from Engine, Act on notification
while(ContinueLooping)
{
    // Wait for a notification
    c = TRT GetInput (NULL, 0, NULL, 0, &rec_data,
TRT BLOCK);
    switch(c)
    {
        // Received a standard notification
        case 0:
        {
            // what type of object is notifying us
            switch (rec_data.code)
            {
                case TRT window:
                {
                    if (rec_data.state == TRT window quit)
                    {
                        ContinueLooping = FALSE;
                    }
                    break;
                }
                case TRT timer hint:
                {
                    break;
                }
                case TRT button:
                {
                    break;
                }
                default: break;
            }
        }
        break;
    }
    // A Callback function was called inside TRT GetInput
    case 1:
    {
        break;
    }
    case - 1:
    {
        break;
    }
    default: break;
}
}
TRT WindowDelete (TRT cid, main window id); //
Close Window
TRT Exit (TRT cid); // Notify Runtime to exit return(0);
}
```

使用 Tilcon 设计好图形界面并编写完用户程序之后,就可以进行编译并把程序和 Tilcon API 链接起来,然后整个系统就可以运行。

4 结 语

Tilcon 作为嵌入式图形开发工具,利用 Tilcon 提供的 Tilcon 图形引擎、平台无关的 API 和一个可视化图形界面设计工具——Tilcon 图形编辑器实现界面开发与设计。这里介绍了基于 Tilcon 的嵌入式操作系统图形界面开发中的工具,探讨了利用 Tilcon 设界面和开发应用程序的过程。通过示例说明, Tilcon 在功能性能方面都能满足基于 VxWorks 的实时系统图形界面

(下转第 62 页)

```
XmlUser xu = new XmlUser( Server. MapPath("users.
xml"));
// 用户凭证验证
if(xu. Authenticate(username. Text ,password. Text))
{
    FormsAuthentication. RedirectFromLoginPage ( user-
name. Text ,false) ;
    Response. Redirect (" ~ \ Admin\ admin. aspx" ) ;
}
else
    Response. Write ("用户名或密码不正确 !") ;
```

单击提交按钮后,Forms 身份验证的过程如下^[10]:

通过 xu. Authenticate 方法进行身份验证成功后,ASP. NET 调用 RedirectFromLoginPage 方法生成身份验证票证;然后调用 FormsAuthentication. Encrypt 方法,将身份验证票证加密为字符串,并将该字符串以 LoginCheck 命名,存放于 Cookie 中。

生成身份验证的 Cookie 后,将该 Cookie 加入到 Response. Cookies 中,发送到浏览器端。

最后通过 RedirectFromLoginPage 方法调用 FormsAuthentication. GetRedirectUrl 方法,以获取到用户原先请求的页面,重定向到此页面。

4 结 语

ASP. NET 作为. Net Framework 的重要组成部分,已成为 Web 应用程序的主流开发工具。在 Internet 中 Web 应用程序的安全是不可忽视的问题,IIS 和 ASP. NET 提供的安全配置系统可大大简化和方便 Web 应用

程序的安全设置。基于 XML 的 Forms 身份验证是利用 XML 文件,结合 ASP. NET 安全配置系统中的 Forms 身份验证,它可以方便地实现 Web 应用程序的安全设置。当然,在实际应用中,还需要进一步加强 XML 文件中用户凭证的安全性以及用户凭证传输的安全性。

参 考 文 献

- [1] [美]拜尔. 开发更安全的 ASP. NET 2. 0 实用程序[M]. 华中宇,田亮君,陈文,译. 北京:人民邮电出版社,2008.
- [2] 刘志成,陈承欢. 基于 ASP. NET 的安全登录系统的实现[J]. 电脑学习,2006(6):44-45.
- [3] 吴观福. 基于 ASP. NET 窗体身份验证构建安全的 Web 应用程序[J]. 电脑学习,2008(3):48-49.
- [4] Jesse Liberty. Programming ASP. NET 中文版[M]. 北京:电子工业出版社,2008.
- [5] [美]活尔瑟. ASP. NET 2. 0 揭秘[M]. 谭振林,译. 北京:人民邮电出版社,2007.
- [6] 李万宝. ASP. NET 技术详解与应用实例[M]. 北京:机械工业出版社,2005.
- [7] [美]贝尔利纳索. ASP. NET 2. 0 网站开发全程解析[M]. 2 版. 杨剑,译. 北京:清华大学出版社,2008.
- [8] [美]Dan Wahlin. 基于 XML 的 ASP. NET 开发[M]. 王玉良,译. 北京:清华大学出版社,2005.
- [9] 肖奕. ASP. NET 2. 0 XML 高级编程[M]. 3 版. 北京:清华大学出版社,2006.
- [10] 邵良杉,刘好增,马海军. ASP. NET(C#) 实践教程[M]. 北京:清华大学出版社,2007.

作者简介 马海峰 男,1979 年出生,山东威海人,硕士研究生,讲师。研究方向为软件工程。

秦桂英 女,1978 年出生,江苏如皋人,硕士研究生,讲师。研究方向为计算机应用。

(上接第 53 页)

开发要求,是理想的嵌入式系统图形界面快速开发解决方案。

参 考 文 献

- [1] 张加林,李松,汪高武,等. 基于 Tilcon 的 VxWorks 图形界面开发技术[J]. 指挥控制与仿真,2007,29(5):100-103.
- [2] 廖容,马忠,肖成俊. Tilcon 在 VxWorks 操作系统中的应用[J]. 舰船电子工程,2007(2):124-126.
- [3] 姜飞,王屹华,崔晓宇. VxWorks 下 Tilcon 嵌入式图形用户界面设计与实现[J]. 工业控制计算机,2008(3):29-33.
- [4] 李阳,黄浩华,刘晓亮. 嵌入式图形系统 Tilcon 及应用研究

[J]. 计算机与数字工程,2008(2):110-112.

- [5] 林建民. 嵌入式操作系统发展趋势[J]. 计算机工程,2001(10):1-4.
- [6] 钱华锋,雷航. 面向对象嵌入式 GUI 研究和模式应用[J]. 计算机应用,2004,24(4):10-12.
- [7] Tilcon Software Ltd. Tilcon Programmer's Guide [Z]. CA,2005.
- [8] WindRiver Systems Inc. WindML SDK Programmer's Guide3. 0[Z]. USA,2002.
- [9] WindRiver Systems Inc. VxWorks Programmer's Guide5. 5 [Z]. USA,2002.
- [10] Tilcon Technology & Architecture Guide Manual [EB/OL]. <http://www.tilcon.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](#)

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](#)

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](#)

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](#)

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](#)

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](#)

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.