

基于 VxWorks 的数据采集 与重演软件的图形界面的设计与实现

杨 蔚

北京机械设备研究所 (北京 100854)

摘 要 通过介绍嵌入式实时操作系统 VxWorks 的图形组件 WindML 以及应用其开发图形用户界面 (GUI) 的框架流程,重点阐述了某数据采集设备中采集与重演软件人机界面的需求和使用 WindML 对其进行设计与实现的过程。

关键词 VxWorks WindML 应用 图形用户界面

The Development of the GUI of a Data Acquisition and Reacting Software Based on VxWorks

YANG Wei

Beijing Institute of Mechanical Equipment, Beijing 100854

Abstract Through introducing the WindML, the graphical component of embedded real-time operation system VxWorks, and the framework of applying it to develop Graphical User Interface (GUI), the paper emphatically expatiates the requirements of the man-machine interface of acquisition and reacting software in a data acquisition device and the process of design and realization of it using WindML.

Key words VxWorks WindML Application GUI

1 引 言

嵌入式设备以及运行于其上的嵌入式实时操作系统 RTOS 被广泛用于消费电子、航空航天、数字图像处理各个领域。嵌入式设备具有体积小、功耗低、集成度高、系统成本低等独特优势。某型控制采集组合是一个嵌入式设备,采用嵌入式实时操作系统 VxWorks 作为其操作系统。VxWorks 是一种高可靠、微内核、可裁剪的实时嵌入式操作系统,以其高效的实时任务调度中断管理等优点,成为航天、

医疗、通信等领域首选的操作系统^[1]。为了尽量减少嵌入到实际系统中运行内核的大小,一般的嵌入式系统不提供图形界面。但为了提高系统的易用性,VxWorks 提供了一个图形开发组件 WindML,利用它可以更快、更简单地开发资源消耗更少的图形界面^[2]。数据采集与重演软件是这型控制采集设备中的一个重要软件,需要大量与用户的交互操作,故需要一个人性化的图形用户界面 (GUI) 来高效地实现其完整功能。本文详细阐述了用 VxWorks 的图形开发组件 WindML 进行交互式图形用户界面开

发的相关技术和流程并阐述了在开发采集与重演软件的图形界面的具体实践过程中的具体问题及其实际的解决方案。

3 实时操作系统 VxWorks 及其图形组件 WindML 概述

2.1 实时操作系统 VxWorks 简介

VxWorks 操作系统是 20 世纪 80 年代由美国 Wind River 公司专门为实时嵌入式系统设计开发的一种 32 位操作系统。良好的持续发展能力、高性能的内核以及以 Tornado 为代表的友好的用户开发环境使其在嵌入式实时操作系统 RTOS 领域逐渐占据一席之地^[3]。VxWorks 具备很多其他操作系统没有或达不到的以下 6 个特点：

- 1 实时性强；
- 2 支持多任务；
- 3 体积小,具有可伸缩、可剪裁和高可靠性；
- 4 支持网络通信、串口通信；
- 5 汇编 + 标准 C 的编程模式,支持 C++ ,兼容 POSIX 标准；
- 6 内核和定制任务可以分开编译、动态下载,支持用户自定义启动任务。

2.2 WindML 的概述和使用

2.2.1 WindML 概述

WindML (Wind Media Library) 是由 WindRiver 公司提供的一种媒体组件库,是 VxWorks 库的一部分,以支持基于嵌入式系统的多媒体应用程序,为各种操作系统提供基本的图形、视频和音频技术,并可以开发基于这些操作系统的设备驱动^[4]。其提供一套与硬件无关的逻辑 API 给应用层调用,而其底层提供不同硬件架构的驱动,使程序具有很好的移植性和可扩展性。其中包含的组件直接控制显示硬件,主要提供显示模式设置、标准输入输出控制、基本图元的绘制、简单窗口管理等模块,其接口是一套 C 语言函数库,是 VxWorks 的图形用户界面 (GUI) 的基础。

2.2.2 WindML 的使用

在第一次使用 WindML 前必须要配置 WindML。WindML 的配置有两种方法:一是通过 Tornado 下的 WindML 配置工具,二是通过命令行方式。一般都是使用前者。如果使用的是 WindML 配

置工具,可以直接使用配置工具中的编译选项将配置好的 WindML 编译、链接生成目标文件即可。如果没有配置工具,则要使用 WindML 自带的 Makefile 文件进行编译和链接来生成目标文件^[4]。编译和链接好 WindML 后,就可以使用 Tornado 工程管理工具把 WindML 加载进 VxWorks 映像里,加载时可以根据自身的需要进行相应的选择。一般对于图形界面的开发,只需将 2D graphics 和 complete 2D library 添加进 VxWorks 即可。这样就可以在应用程序中使用 WindML 提供的 API 函数来开发图形显示程序了。

3 采集与重演软件简介及其对图形用户界面的需求

3.1 采集与重演软件简介

采集与重演软件运行在控制采集组合的采集计算机中,主要完成控制总线报文信息的捕获、验证、记录、过程数据重演,对总线上所挂接设备中数据的网络下载,以及本地数据下载等功能。其采集功能主要包括采集以太网上所有符合系统通讯协议的信息并能快速保存至电子盘中,重演功能主要包括按时间顺序显示系统各设备在以太网上传输的报文或执行的动作,将以太网上采集数据的重演结果以中文形式显示在人机界面上,数据下载功能主要包括通过以太网将设备存储的数据下载到本地或外部存储介质、通过本地 USB 接口将保存在本地的网络数据和设备数据导出。

3.2 软件对图形用户界面的主要需求

采集与重演软件的图形用户界面与用户的交互的途径是包括采集页面、重演页面、网络下载页面及本地下载页面在内的数个页面窗口。整个界面软件的功能包括与后台软件的数据通信功能和显示功能。这里主要讨论对显示功能需求,并以操作和显示的方便性和人性化为重点。

3.2.1 各页面的公共需求

各个页面都需要能显示后台软件传来的当前系统时间,将后台软件在实现各自功能时产生的报警信息和故障信息在一定的位处显示出来。

3.2.2 各自页面的主要需求

数据采集页面主要显示后台软件所进行的数据采集的状态是进行中还是停止,为了显示的人性化,

要求在数据采集进行的过程中实现进度条的滚动。

重演界面需要将后台软件以一定格式传输给界面软件的各设备在以太网上传输的报文或者执行的动作以表格形式直观地显示给用户,表格横坐标为时间,纵坐标为各不同设备。此外,在界面输入方面需要用户可以点击按钮控制重演的开始和结束、选择需要进行重演的设备,以及输入相应的重演起始时间和结束时间。还有一页的显示内容是有限的,后台的数据条数往往超过单页显示条数的限制,所以需要能进行上下左右的翻页显示。

网络下载和本地下载页面的操作和显示比较类似,都需要将后台软件数据下载的状态显示给用户,辅与采集界面类似的进度条,还有需要用户控制下载的开始和结束、输入下载的起始时间和结束时间。所不同的仅是本地下载在选择设备时多了一个“网络数据”的选择按钮,用以将已经下载的网络数据下载至本地。

3.2.3 整体上的其它重要需求

根据软件总体要求,首先在数据采集、数据重演以及数据下载正在进行的过程中,点击执行其它功能的按钮是被禁止的,必须等当前过程结束才能进行其他功能的操作,所以需要一个小提示页面显示操作提示信息。第二,在进入本地下载和网络下载界面时都需要一个小页面输入用户名和密码进行确认,此页面在操作习惯上须类似 Windows 的相应输入框。

4 人机交互图形用户界面的开

4.1 WindML 图形用户界面的框架

使用任何工具进行图形用户界面的开发几乎都有一个框架可以遵循,使我们能够尽量减少编程的错误和编程工作的重复性,从而提高编程的效率。利用 WindML 开发图形界面也不例外。一般情况下,使用 WindML 进行图形开发设计有着如下的总体框架:

- 1 对程序进行初始化。
- 2 进行包括显示、字体、指针、事件处理、声音、框架等的设备驱动装载。
- 3 初始化包括主窗口设置、时间路由创建等其它信息。

4 创建图形环境变量,初始化颜色并分配预定义的 RGB 值,创建指针、画背景等。

5 利用一个无限循环构成时间处理主机。这一步是程序关键所在,所有的交互(鼠标、键盘、视图等)过程都由其产生或完成。循环体只有在接收到特定的命令时才会跳出,执行下面的内容。

6 完成对所有资源的释放并退出。少释放了任何资源都将导致内存泄漏,最终导致不可预料的严重后果^[10]。

以上步骤是总体设计框架,较为抽象笼统,在具体编程时应根据实际要求灵活运用。

4.2 重演软件人机交互界面的整体开发

用 WindML 绘制界面由于缺少类似 VB、VC 等高级软件中能直接利用的现成控件,所以所有界面元素的实现都需要将基本图形如直线、矩形、文字等以坐标为核心进行有机组合以实现所需的显示效果。所以在进行了 WindML 程序框架中所必须的一些初始化工作之后,在正式开始页面主体编程之前,必须对所画的页面有一个整体规划,将各个界面元素如按钮、时间输入框等的坐标进行预先设计和排列,之后再在调试中根据需求变化及实际效果进行调整。在开发的过程中采取先开发主功能页面窗口(即数据采集、重演和两个数据下载窗口),然后再开发辅助功能窗口(即提示信息窗口和用户名密码验证窗口)。

整个程序用一个独立的主循环接收后台软件的数据,根据接收数据的约定预设标志位的不同分别用对应窗口进行显示,而用户的操作和输入的数据由各相应窗口自身的程序发送给后台软件进行处理。程序的流程为通过一个置于重演软件执行入口处的起始函数首先发起一个创建窗口的任务,在其中对字体、驱动、颜色初始化完毕之后,创建各个页面窗口并对这些窗口的见面元素如按钮、边框等进行初始化,紧接着发起一个数据接收任务,此任务执行的就是上面接收数据的主循环函数,此时整个界面程序开始工作,在不断接收后台数据的同时不断刷新显示,并将用户在界面上的操作包括点击选择和输入的数据发送给后台程序处理。软件的各项窗口之间关系以及任务与窗口之间关系如图 1 所示。

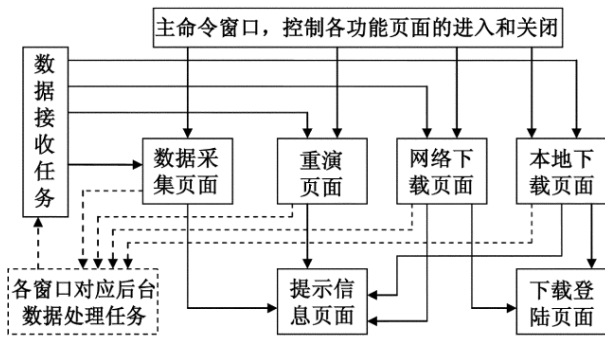


图1 窗口间及窗口与任务间关系框图

5 界面设计中的几项关键技术及显示技巧

5.1 列表显示功能

数据重演窗口的最重要功能就是以表格形式将各设备在以太网上传输的报文或者执行的动作列出来显示给用户,并还能进行左右上下的翻页显示的操作,这也可以说是整个界面程序的一个主要难点之一。在头文件中将所接收的报文消息定义在一个

结构体数组 NetReport []内,每一个数组元素对应一条报文消息,给每个元素定义的主要属性有时戳 timestamp、设备号 devIndex、报文内容字符串数组 rpt []、显示行号 line、显示列号 cols。表格线条的坐标首位两点的坐标在头文件中以二维数组 tablecols [] [4]和 tableline [] [4]的方式定义好,如表格第三条列线的坐标为 tablecols [2] [0],第二条行线坐标 tableline [1] [0]。报文显示的位置均以表格坐标为参考点。页面显示时首先绘制好表格线条并根据按钮的选择将设备名显示在表头横栏中,设备名也是一个结构体数组,也有设备号 devIndex 和列号 cols 属性,每选择一个设备,所选设备的 cols 加 1。接着将当前接收到的报文的设备号与已经选中的设备的设备号进行循环比较,符合某个设备后将列号赋为对应设备名的列号从而确定报文显示的列位置,同时将一个屏幕中所有 21 条报文按其时戳顺序行号循环加 1 从而确定各自的行位置,最后按照行列位置以对应的表格坐标为参考在对应的设备名底下对应时间的位置上。

采集与重演			
时间 设备	控制设备1	控制设备2	控制与维护计算机
20101020_13:20:30:10021		配置信息	
20101020_13:20:30:12011	模式选择		
20101020_13:20:30:90120		模式反馈	
20101020_13:20:31:10010			配置信息查询
20101020_13:20:31:13012		配置信息	
20101020_13:20:33:12450	选择3#对象		
20101020_13:20:33:90125		3# 对象选择成功	
20101020_13:20:37:53142	3# 对象供电控制		
20101020_13:20:37:90154		供电 控制故障	
20101020_13:20:40:70120	3#对象供电控制		
20101020_13:20:40:90120		供电 控制正常	
原始数据区			
显示报警信息			
<div style="display: flex; justify-content: space-around;"> 数据采集 退出重演 网络下载 本地下载 </div>			

FK1	FK2
FK3	FK4
FK5	FK6
FK7	FK8
WK1	WK2
WK3	WK4
FX	FX
JZ1	JZ2

重演起始时间设置:

重演结束时间设置:

上翻页

前翻 ← → 后翻

下翻页

开始重演

图2 界面列表显示示意图

实现翻页的重点在于左右翻页。表格除时间戳外只有四列,而设备有 16 个,最多能占到四页。需定义一个变量 `offset`,以右翻页为例,每按一次右翻页按钮 `offset` 加 1,说明右翻了一页。屏幕有四列则表格列坐标等间隔有 4 个,此时以第六个设备为例应显示在第二列的位置,则其所有报文的纵坐标为 `table[1 + offset][0]`。若未翻页则 `offset` 为 0,第二列正好对应第二个设备。左翻页方法类似,需将 `offset` 减 1。界面列表显示的示意如图 2 所示。

5.2 实现滚动条动画显示的功能

在数据采集和两个数据下载界面,当正在进行数据采集或者数据下载时,屏幕上应该有相应的提示,而比较人性化的提示当然是类似 Windows 中的进度滚动条。用 WindML 中实现这样的进度滚动条时第一个问题是间隔一定时间重画当前屏幕。定时采用看门狗来实现,并通过在看门狗所执行函数体中再次加入看门狗启动函数 `wdStart` (来实现每隔一秒循环启动看门狗。看门狗的函数体通过消息队列向数据接收主循环发送窗口刷新消息,主循环根据所接收数据的标志位判断当前需要刷新的是数据采集还是数据下载窗口,然后调用 `winRectInvalidate`

(函数重画窗口。将整个滚动条框宽为 400 像素,分为 10 段每段 40 像素,滚动条宽 40 像素从左至右在框内移动,移动 10 段后返回第一段的位置,周而复始。变量 `dispX`, `dispY` 为滚动条起始坐标即第一段起始坐标,另外使用一个 `count` 变量进行计数。窗口每次刷新重画时, `dispX` 加 40 即一个滚动条宽度, `count` 加 1,当 `count` 加到 11 时, `dispX` 重新赋值为起始值, `count` 重新赋为 1。这样,就实现了进度条从左至右的循环滚动。

5.3 在屏幕上修改输入数据并实时显示的技巧

WindML 中没有高级软件中类似输入框的控件,要实现用户数据输入并实时显示在屏幕上需从底层图形绘制着手用一定技巧来实现。以数据下载窗口的时间输入为例,其格式为年/月/日,其中年为 4 位,月和日都为两位,初始显示 2000/01/01。

基本的显示方法为以键盘事件提取输入键值,在窗口刷新时在原字符位置画一个全黑矩形抹掉原字符后画上新字符。

而根据软件的具体要求,要能对日期进行逐位修改,这在实现了基本输入显示的基础上还需要一些细节方法的扩展。定义一个日期结构体,其中有两个重要元素:日期字符串 `dateTime[]` 以及当前操作的日期字符串的位数 `pos`。初始输入默认从字符

串第一位开始, `dateTime[]` 初值“2000/01/01”, `pos` 初值为 0,每输入一个字符,将此字符赋给 `dateTime[pos]` 然后 `pos` 加 1,那么下一次输入的字符将自动赋给字符串的下一位。另外还应实现按键盘左右键时,字符串的修改位置能跟着左右移动,且在字符串下方画出一道横线也随之移动以向用户指示用户修改的是字符串哪一位。以右键为例,在右键按下的事件的响应函数中使 `pos` 加 1,当遇到“/”时 `pos` 再次加 1 跳过,然后在对应字符的下方画出一条长为一个字符宽度,宽为数个像素的矩形标识当前所修改的数据位,矩形的横坐标每次随右键按下增加一个字符的宽度,同时将前一个矩形条用画一个相同大小全黑矩形条的方法抹掉,这样每按一次右键,矩形条就会向右移到下一个字符下方,恰好这个字符处于可修改状态。当 `pos` 值到了字符串末位时通过条件判断让其归零,这样修改位置就可循环移动。

5.4 实现人性化显示的用户名及密码输入功能的技巧

实现用户名及密码输入窗口以上面所述的数据输入显示及修改为基础,但是本着与用户交互尽量人性化的目标,要实现类似高级软件的效果则还需要一些添加一些细节。主要是除使用左右键定位所需修改的已输入字符外,还需要能在已输入字符串中的任何两位之间插入若干字符或者从任意两位之间开始删除若干字符,也就是说所输入字符串长度不像日期字符串那样固定,而是动态的。

以在用户名输入框中删除字符为例,用户名字符串数组为 `loginname[]`, `pos` 定义为操作的字符位。字符串在判断鼠标点击在用户名输入框坐标范围之内后,将 `pos` 定位于已输入字符串数组的末尾,同时在屏幕上所显示的已输入字符串末字符右端绘制一个竖型矩形条,此时用修改上面日期的方法可实现在左键时竖型矩形条以一个字符宽度为单位向左移动,同时 `pos` 值相应的同步改变,确保竖型矩形条一直位于 `loginname[pos]` 对应的字符右端,此时按删除键,在其按键事件响应函数中将 `pos` 减一,并使用 `strncpy` 函数将以 `loginname[pos + 1]` 开始的被删除位以后的字符串拼接在此时的 `loginname[pos]` 字符串之后形成一个新的 `loginname` 字符串,然后重画在输入框中。这样就成功地删除了原字符串中间的某一位,并且恰好 `pos` 位于被删除这一位的前面一位,竖型矩形条也位于被删除这一位的前面一位的右端,可以继续删除或者其他操作。

6 结束语

在后 PC 时代,嵌入式技术将深入到各行各业以及生活和工作的方方面面中去,而嵌入式人机界面的应用前景也将更为广阔^[5]。本文所设计的基于 VxWorks 的采集与重演软件人机交互图形用户界面是某型号发射装置的重要组成部分,具有较好的软件可维护性、代码可重用性、功能扩展性,以一些实用化的技巧尽量做到了人性化即操作直观、方便、简洁,经过与后台程序的联调,较好地支持了整个系统功能的实现。

参考文献

- [1] 皇甫祯,陈怀民,段晓军,吴成富. VxWorks 嵌入式图形界面设计[J]. 测控技术,2007,27(4):77~79.
- [2] 赵漫菲,王光辉. 基于 VxWorks 的 WindML 图形界面程序的框架分析[J]. 计算机工程与设计,2006,27(6):522~540.
- [3] 袁渊,曾文火. 基于嵌入式操作系统 VxWorks 的图形界面开发[J]. 华东船舶工业学院学报(自然科学版),2002,4(17-2):40~44.
- [4] 余彬,靳力,曾翔. WindML 显示技术的优势和实现[J]. 舰船电子工程,2005,33(4):72~74.
- [5] 徐芳,程少华,何亦征等. 基于 VxWorks 的无线接收器的人机界面设计与实现[J]. 微型电脑应用,2005,21(7):39~61.
- [6] 王珊珊,秦开宇,谈秀娟. VxWork 下交互式用户界面开发[J]. 现代电子技术,2007,40(4):58~60.
- [7] 曲宏松,张叶,曹立华等. VxWorks 下多页图形界面的设计[J]. 微计算机信息(嵌入式与 SOC),2006,22(10~2):192~194.
- [8] 郭琳,付梦印,肖炬. 基于 VxWorks 的导航系统人及交互界面的开发[J]. 微计算机信息(嵌入式与 SOC),2006,22(10~2):50~52.
- [9] 梁勇,孟桥. 嵌入式操作系统 VxWorks 中的多媒体组件 WindML2.0[J]. 现代计算机,2002,10(8):88~91.

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.