基于 Tilcon 的 IO 控制板可视化测试软件的设计和实现

孔梁萍

（北京计算机技术及应用研究所，北京　100854）

摘　要：针对嵌入式系统下图形界面开发困难的问题，以一个通用的 IO 控制板测试软件的开发为例，介绍了一种嵌入式下图形开发工具 Tilcon，深入分析和研究了开发中涉及的 BSP 和 WindML 图形库组件的配置以及 Tilcon 工具的裁减和配置，对基于 Tilcon 的程序开发方法和流程进行了分析和说明。工程应用实践表明，软件的可靠性和实时性得以保证，操作方便有较高的实用价值。

关键词：Vxworks；WindML；BSP；Tilcon；图形界面

中图分类号：TP311.52　**文献标识码：**A　**文章编号：**1007-9599 (2012) 14-0159-05

一、引言

VxWorks 是由美国风河系统公司开发的高性能嵌入式实时操作系统之一，它以其优秀的可靠性、实时性及内核的可裁减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。WindML 是 Wind River 公司提供的基于 VxWorks 的多媒体的支持库，为各种体系结构的计算机提供基本图形媒体库，但是 WindML 功能单一，代码繁琐，效率较低，开发高质量的图形界面相当困难。为满足复杂图形界面用户需求，目前风河公司推荐使用 Tilcon 工具，它同步支持最新版本的 Tornado/VxWorks 以及 WindML 多媒体库。

Tilcon 是一个支持多种操作系统平台的图形用户界面的开发工具，它是目前最先进的 Vxworks 下实时操作系统图形开发工具。Tilcon 本身已经集成大量成熟控件，用户可以像 Windows 下 VC 可视化编程一样用拖动控件的方式构造自己的图形应用，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，彻底解决了困扰嵌入式实时图形领域应用的难题，具有极高的可靠性和可维护性。

本文以一块通用的 IO 控制板的可视化测试软件的开发和实现过程为例，结合上述各个软件工具的开发特点，针对开发过程中各个工具使用的关键和难点，提出一种基于嵌入式图形软件开发的方案，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，实现嵌入式实时操作系统下图形的开发。

二、硬件系统设计

IO 控制板的作用是进行数据的控制，其外部连接 16 路开关输入和 16 路输出来控制设备状态的显示。内部通过桥接芯片 9052 进行 PCI 总线与局部总线之间的转换，并与主机通信。开关量的通信和控制主要是采用 FPGA EP1C3T144I 来实现的。接收到主机的开关量控制信号后，经过 EP1C3T144I 的译码、光耦 ACPL247 隔离、继电器驱动，输出电源 24V 或者 24VGND；同时，开关量输入输出模块接收到开关量输入，经过光耦 ACPL247 隔离、信号整形后，送入 FPGA，再以中断方式发送给主机。

IO 控制板采用多个中断复用的方式，当 IO 开关量任意一个输入产生中断时，均向主机产生中断。主机读取中断状态寄存器来判断中断源，读取后中断状态寄存器自动清 0，硬件原理图如图 1 所示：

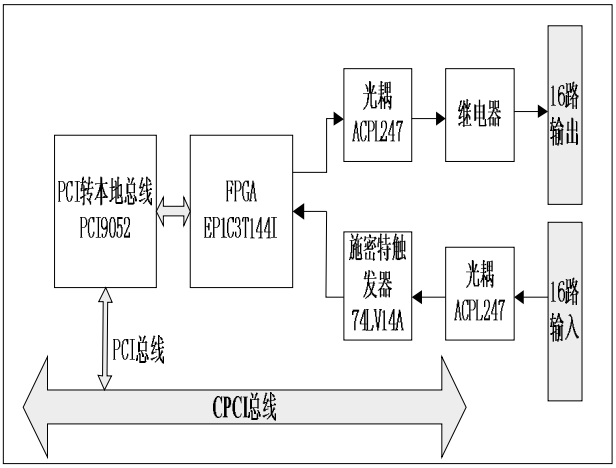

图 1　IO 控制板的原理框图

三、软件系统设计

软件设计包含六个部分，主要涉及 VxWorks 下 BSP 系统支持包的配置、WindML 的配置和编译、Tilcon 的裁减与配置、以及驱动程序的设计、人机交互演示程序的设计，软件各个部分的关系如图 2 所示。

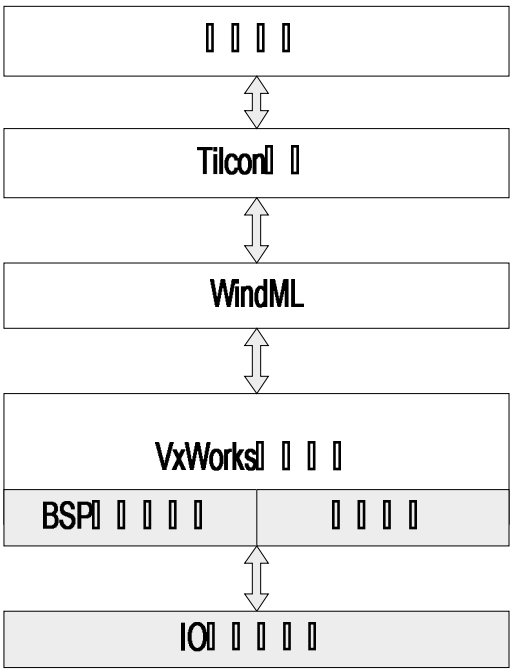

图 2　层次关系

为了实现应用软件的可视化，本系统采用 Tilcon 工具进行图形界面的设计。Tilcon 在 VxWorks 下的应用是基于风河公司的 WindML 的，所以，必须对 WindML 和 Tilcon 进行必要的裁减和配置，接下来将对各个部分作具体的介绍。

（一）系统支持软件的设计

板级支持包(BSP)软件是 VxWork 操作系统与便件目标板的桥梁和屏障，它对上屏蔽目标板的硬件细节，为 VxWorks 操作系统的应用程序提供了一个统一的接口。对 BSP 的配置首先要分析主机系统以及目标板的架构和组成，因为系统主机采用 855GME 芯片组构建主处理器板，主处理器板提供的各种端口都是标准端口，所以选用 Tornado2.2 下 PENTIUM4 配置作为模板进行开发，该 BSP 可以支持主处理器板的大多数标准设备。在 pcPentium4 标准 BSP 的基础上，通过修改对应的选项为主处理器板做支持。

IO 控制板支持 16 路独立的 IO 控制通道，可控制 16 个 IO 开关量输入和 16 个 IO 开关量输出，支持查询的方式获取 IO 输入状态，当 IO 输入状态方式变化时，支持以中断的方式获取 IO 输入状态的变化。因此在驱动程序设计上按照字符设备驱动模式来开发，支持标准的 creat、open、read、write、ioctl、close、delete 函数访问。当基于 Tilcon 进行界面设计时，只需要将 Tilcon 所对应的按钮或文本框等消息响应函数与驱动接口函数进行连接就可以了。

（二）WindML 的配置和编译

用 Tilcon 完成应用程序图形用户界面开发的前提条件就是对 WindML 进行相应项目配置和系统编译并将其加载到嵌入式实时操作系统 VxWorks 内核，经此处理后方可在 Tornado 集成开发环境中进行源代码编程与 API 函数调用。WindML 的配置是指用户根据目标机配置情况及应用需求对 WindML 的处理器类型、图形设备、输出设备、字体、音频设备等选项进行配置。完成配置后，需对所配置的 WindML 进行编译并生成目标文件，具体步骤如下：

1. 进入 Tornado 集成开发环境后，选择 Tools/WindML/Configure 步骤，打开 WindML Configuration 配置对话框，结合主处理器和 IO 控制板硬件构成对 WindML 进行配置。具体配置参见下表：

表 1　WindML 配置表

配置项	配置子项	配置内容或配置子项	配置内容
Build	Processor	PENTIUM4	
	Tool	gnu	
	Debug	Build debug version	
	Additional builds	Build WindML object	
		Build WindML examples	
Devices	Graphics Driver	VESA BIOS	
	Display Type	Generic Monitor	
	Graphics Settings	Color Format	16-RGB565
		Resolution	1024x768
		Refresh Rate	60
	Pointer	PS/2 Style Pointer	/pointer/0
	Keyboard	PC/AT Style	/pcConsole/0
Advanced	Advanced Features	JPEG	
		Double Buffer	
	Memory Options	VxWorks System Pool	
	Window Manager	WWM	
Bitmap Fonts	Unicode	Include Unicode Support	
	Fonts	All	

2.按上表所示流程来配置对应选项，根据实际图形用户界面

开发需要选择添加 WindML components 组件。选择添加 WindML devices 下的 WindML input device 下的 PS2 keyboard 和 PS2 mouse 和 graphics device 以及 select 2D layer link method 下的 complete 2D library。

为检验 WindML 媒体库是否配置、编译成功，按如下路径关系$(WIND_BASE)\target\src\ugl\example 定位到 ugldemo.c 文件，并在 Tornado 环境下建立 Downloadable 工程经编译下载运行后，如果能在目标机上出现欢迎界面则表示 WindML 操作成功。

（三）Tilcon 的裁减与配置

首先将 Tilcon 安装包提供的 licensev.til 文件拷贝到目标机 C 盘根目录下，将 licensem.til 文件拷贝到主机 Tilocn 安装目录下。其次，在 Image 镜像文件中加载几种组件支持，包括组件 C++Components 下的所有子项和 POSIX Components 组件下 Clocks、Message Queues、Semaphores 和 Timers 子项。再次，将 C：\tilcon\TSP\VXWORKS-x86\target\lib 目录下的 objPENTIUMgnuTugcl 文件夹拷贝到 C：\Tornado2.2\target\lib 目录下，执行“create”。最后，运行 Tilcon 安装目录下的 scalable.exe 可执行文件生成支持 IDS 应用程序模型的 tlncore.o 和 tlnapi.o 这两个目标静态库文件。

配置完 Tilcon 后就可以在 Tornado 的 Bootable 工程项目中配置编译环境了，在 C/C++ Compiler 标签页中添加-DPRJ_BUILD -DCC_TRT_VXWORKS -IC：/tilcon/Include/，因为 VxWorks 可能会包含某些特殊字符信息，如果按-ansic 语法进行编译检测就会出现错误提示信息，所以去掉-ansic 选项。在 Micros 标签页下选择 EXTRA_MODULES 宏选项，在其中添加 C：/Tornado2.2/target/lib/tlnapi.o 和 C：/Tornado2.2/target/lib/tlncore.o 以便找到目标静态库文件。

完成以上所有 Tilcon 与 Tornardo 集成开发环境属性配置后就可以开始根据驱动提供的接口函数结合 Tilcon 所提供的 API 接口函数进行 IO 控制板的可视化测试软件的设计了。

四、典型应用程序的设计

根据武器显控系统对 IO 控制板的具体要求，可视化测试软件可以划分为以下几个功能模块：初始化模块、数据通信模块、人机交互显示模块。初始化模块主要完成 Tilcon 和驱动的初始化，如对中文字体的注册以及主窗体和驱动的加载等功能。数据通信模块主要是将各种 TRT 事件与 IO 控制板数据的输入和输出控制连接起来。人机交互显示模块就是通过响应 TRT 事件从界面上设置或接收并显示 IO 控制板数据通信的结果。软件设计流程图 3 如下：

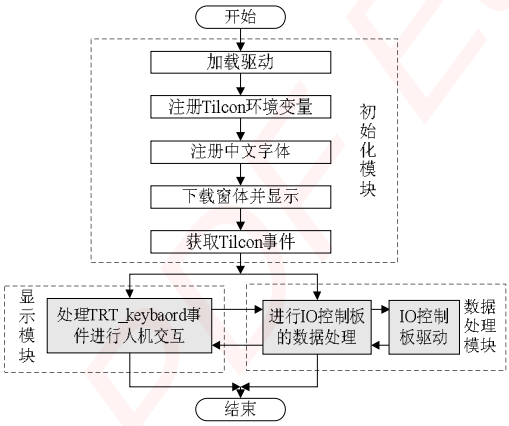

图 3　软件设计流程图

程序设计时应该包括一个主事件循环来控制用户界面。首先使用 TRT_Start 函数启动一个 Tilcon 图形引擎，这样 VxWorks 与 Tilcon 之间就建立了连接，在完成了图形用户界面窗口初始化操作并进行加载显示之后就可以与此引擎进行通信。这时，程序创建两个任务，其中一个任务进入主循环等待事件的产生，另一个任务处理接收和发送的 IO 控制板的数据。当有按钮或文本框等事件产生时，Tilcon 传送一个描述此事件的数据结构 Recdata.data 给用户程序，分析此数据结构，然后根据界面的设计确定此事件对应的是 IO 控制板数据的输入还是输出或是对中断数据的接收等控制动作。Tilcon 的图形引擎把 GUI 函数调用和用户程序隔离开来，它是一个独立的进程，这种结构在保证用户程序实时性的同时提高了系统的可靠性，用户程序不需要等待命令是否被处理完毕或图形显示是否被更新，就可以直接返回到主任务中。

程度设计时还需要注意以下两点：

1.应当在 Bootable 工程项目 usrAppInit.c 文件中的用户初始化函数 usrAppInit 中定义系统盘名称挂载硬盘，如：usrAtaConfig(1，0，"/ata0")表示本系统电子盘挂在 1 控制器上的第 0 个设备，ata0 是对系统第一个分区的命名。

2.在 Bootable 工程项目 usrAppInit.c 文件中指定 Tilcon 所使用的版本序列号，即：putenv("Tilcon_5_4=/ata0")，本系统中 ata0 就是系统盘 C，设置这个环境变量就是使其指向注册文件地址。

程序的基本框架和主要代码如下：

```
/*必须包含的 tilcon 的头文件*/
#include <tilcon/TRTConst.h>
#include <tilcon/TRTAPI.h>
#include <tilcon/TRT_API.h>
#define OS_TYPE TRT_VXWORKS/*定义操作系统*/
pid_t   TRT_cid；/*ID 资源*/
TRT_ReceiveData  rec_data；
#define MAIN_WINDOW_FILE      "IOTestDemo.twd"/*图像界面工程名称*/
#define MAIN_WINDOW_ID        "ID_IOTest"/*软件主窗口 ID*/
char   main_window_id[TRT_MAX_ID_LENGTH]；
void IOTest()
{        /*加载驱动*/
TRT_StartData StartData；
TRT_FontRegister fontregisterprops；
long taskMask；
/*设置 Tilcon 环境变量*/
putenv("Tilcon_5_4=/ata0")；
StartData.Os_Env = OS_TYPE；
StartData.Display   = NULL；
StartData.IPAddr    = NULL；
StartData.AppName       = strdup("TRTD_CRSE1")；
StartData.Userprog        = "TRTDCRSE1"；
StartData.Flags      = FALSE；
errorcode =TRT_StartEx ( mask，   &StartData)；
TRT_cid = StartData.TRT_CID；
/*注册中文字体*/
taskMask = TRT_FONTREGISTER_MASK_SET
||TRT_FONTREGISTER_MASK_FACENAME；
fontregisterprops.fontType = 33；
fontregisterprops.fontStyle = 0；
fontregisterprops.nameStr = "song"；/*Unicode*/
TRT_RegisterFont(TRT_cid ， taskMask ， (void *)(&fontregisterprops))；
/* 设置系统刷新频率为 52*50ms */
errorcode   = TRT_TimerHintEnable(TRT_cid，2)；
/*导入主窗口*/
errorcode = TRT_WindowLoad(TRT_cid ， MAIN_WINDOW_FILE )；
/*获取窗口 ID*/
errorcode = TRT_GetWindowID(TRT_cid，main_window_id)；
/*显示窗体*/
errorcode = TRT_WindowDisplay(TRT_cid ， MAIN_WINDOW_ID)；
while(ContinueLooping)
{/*等待消息通知*/
c = TRT_GetInput(NULL， 0， NULL， 0， &rec_data， TRT_BLOCK)；
switch(c)
{
case 0：
{
switch(rec_data.code)
{
case TRT_keyboard：
ProcesskeyboardMessage()；/*处理按键消息*/
break；
case TRT_button：
ProcessbuttonMessage()；/*处理按钮消息*/
TRT_SetValues(TRT_cid ， "ID_P1" ， TRT_ATT_VALUE ， Rev_buffer，NULL)；     TRT_TextUpdateNumeric(TRT_cid ， "ID_P2"，TRT_ATT_TEXT，Rev_buffer)；
break；
case TRT_radio_button：
ProcessradiobuttoMessage()；
break；
}
}
break；
```

（下转第 176 页）

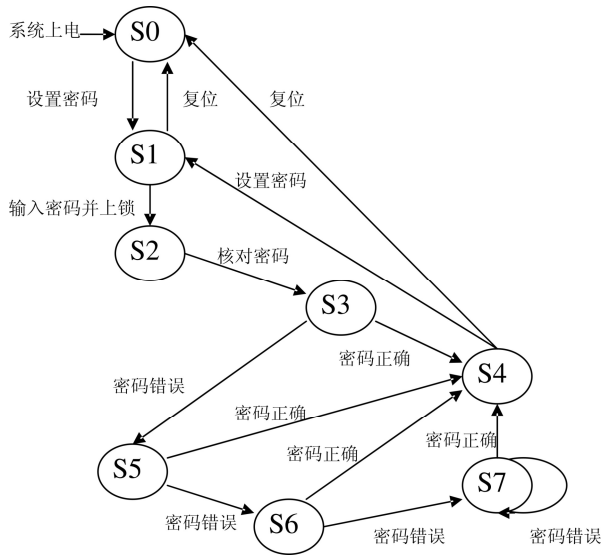

图 1

五、结束语

本设计采用状态机的描述方法实现密码锁的逻辑功能，基于状态机的编程方法使程序结构清晰，在 Quarters II 软件平台下通过编译、锁定引脚、配置，系统的仿真结果和设计要求一致，能够完成控制任务。由于 FPGA 的可重复编程性，可在此基础上还通过修改源程序和相应外围电路对密码锁的功能进行升级和扩展，该密码锁的设计和开发方法灵活、高效，具有一定的实际应用价值。

参考文献：

[1]焦素敏.EDA 应用技术[M].北京:清华大学出版社,2005

[2]赵立民.可编程逻辑器件与数字系统设计[M].北京:机械工业出版社,2003

[3]陈丽华,何颜平.基于 VHDL 的数字密码锁设计[J].国外电子测量技术,2008,(4)

[作者简介]侯静（1983-），女，陕西省西安市，助教，研究生，研究方向：电子信息技术。

（上接第 162 页）

```
default：
break；
}
taskDelay(10)；
}
TRT_WindowDelete (TRT_cid，  rec_data.ID )；
TRT_Exit (TRT_cid)；
return；
}
```

软件运行结果如下图所示：

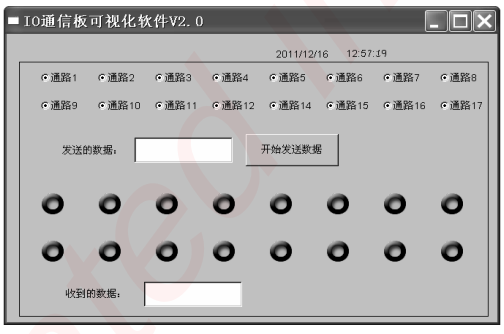

图 4　软件运行效果图

五、结束语

IO 控制板可视化测试软件采用 VxWorks 操作系统，以 Tilcon 设计用户操控界面，使用时直接在图形界面上对 IO 控制板进行控制和操作，实现了数据的同步显示，此软件的设计流程和架构可以作为 Tilcon 开发的通用模板使用。经测试表明，运行效果和图形质量以及可操作性都较单独使用 WindML 有较大的提高，这样的设计既缩短了软件设计周期，又提高了系统的可靠性和可维护性，可视化测试软件界面美观操作方便，具有良好的可视化效果和很高的实用价值。

参考文献：

[1]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

[2]廖容,马忠,肖成俊.Tilcon 在 VxWorks 操作系统中的[J].舰船电子工程,2007,2:124-126

[3]姜飞,王屹华,崔晓宇.VxWorks 下 Tilcon 嵌入式图形界面设计与实现[J].工业控制计算机,2008,3:29-33

[4]Tilcon 公司 Tilcon User Guide 2005

[5]W indMl3.0 Programers' Guide

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

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

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

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

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

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

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

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