

基于 Windows CE 下大像素图像分块显示算法的研究

朱亚奇 刘维亭*

(江苏科技大学电子信息学院, 镇江 21200)

摘要 Windows CE是一种典型的嵌入式系统,由于其配置的硬件资源的瓶颈,内存资源比一般的 PC相对紧张。所以在安装 Windows CE的设备下显示大像素图像文件比在一般的 PC(Windows 2000系统)上更加困难。介绍在 Windows CE环境下如何利用分块显示法,对大像素图像进行分块显示。提出首先在 PC机上利用网格分块法将一幅大像素图像文件 BMP, JPEG图像文件分割压缩成多块像素量小的 BMP, JPEG图像文件,对每个小图像文件进行编号,存放到 Windows CE PDA上;然后在 Windows CE PDA上利用网格检索的方法进行检索所需要的像素块的索引号;最后利用多线程的方法在显示区分块显示,实现大像素图像在 Windows CE下的显示。

关键字 Windows CE 嵌入式系统 JPEG 大像素图像 网格分块法 网格检索 多线程
中图分类号 TP391.74; 文献标志码 A

在很多工程应用中,往往涉及到对大幅图像的显示问题,然而对图像的操作非常消耗计算机内存资源^[1-4]。在 Windows环境中最重要的图像格式就是位图(Bitmap),另一种就是 JPEG有损压缩图。位图在 Windows中是采用 RGB颜色表示法,就是采用 R, G, B颜色分量来表示数字图像颜色值,每一个颜色值用一个 8位的无符号(BYTE)整数来表示,3个颜色值则占用 3个字节 24位的存储空间。对于一幅大图,如一幅 10 240 ×10 240像素的真彩色位图,采用的 RGB颜色表示法占用 10 240 ×10 240 × 3 = 300 Mb,要对这幅图像在硬件资源有限的 PC机中显示是比较困难的。而 JPEG一个复杂有损压缩图像文件,对其进行压缩的目的主要为了节省存储空间,便于网络传输。对 JPEG图像进行处理,必须先通过译码器对其进行解压,在计算机内存中解压还原成原始图像数据^[5,6]。无论对是 BMP格式,还

是对 JPEG格式图像的显示都将消耗大量计算机内存。

Windows CE是微软公司嵌入式、移动计算平台的基础,它是一个开放的、可升级的 32位嵌入式操作系统,它是精简的 Windows 95, Windows CE的图形用户界面相当出色。Windows CE具备 Windows的一般特性,但装置 Windows CE的设备,一般多是 CPU资源,内存空间相当有限的嵌入式设备,在资源有限的 Windows CE嵌入式设备中,要是采用一般的方法直接调用 Windows CE提供的 API函数显示类似 10 240 ×10 240像素的 24位数字图像,在目前来说是实现不了的。本文提出一种改进的方法,采用对大图像分块显示法,实现大图像在 Windows CE环境显示。

1 JPEG图像压缩与解压缩

1.1 JPEG压缩的基本原理

JPEG一个复杂图像的压缩技术。这里,采用了 JPEG基准模式对图像进行压缩,在基准模式中,熵编码采用哈夫曼编码方法^[2],图 1是 JPEG压缩与解压缩过程的图解。

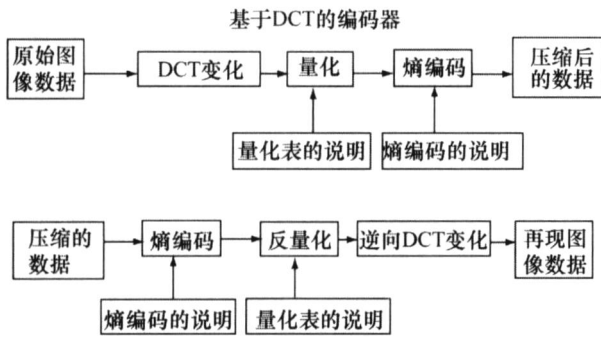


图 1 JPEG压缩与解压缩过程的图解

1.2 BMP图像的灰度化

由 RGB空间向 YCbCr空间转换的公式如公式组 (1)。

$$\begin{aligned} Y &= 0.2990R + 0.5870G + 0.1140B, \\ Cb &= -0.1687R - 0.3313G + 0.5000B + 128 \\ Cr &= 0.5000R - 0.4187G - 0.0813B + 128 \end{aligned} \quad (1)$$

1.3 利用 DCT对空间频率的变换

离散余弦变换 (DCT)实现将一组光强数据转换成频率数据。在压缩时,将源图像数据分成 8×8 像素构成的像块的集合。经过零偏置将每一取样值从 $0 \sim 255$ 转为 $-128 \sim +127$,再做 DCT处理。解压缩是正向变换的反过程。DCT和 DCT分别由公式 (2)和公式 (3)实现。

$$F(u, v) = \frac{1}{4} C(u) C(v) \sum_{x=0}^7 \sum_{y=0}^7 f(x, y) \times \cos\left(\frac{(2x+1)u}{16}\right) \cos\left(\frac{(2y+1)v}{16}\right) \quad (2)$$

$$f(x, y) = \frac{1}{4} \sum_{u=0}^7 \sum_{v=0}^7 C(u) C(v) F(u, v) \times \cos\left(\frac{(2x+1)u}{16}\right) \cos\left(\frac{(2y+1)v}{16}\right) \quad (3)$$

$f(x, y)$ 为原始图像数据空间的函数, x 为像素所处的行, y 为像素所处的列, u 和 v 对应频域空间的行和列。上式中 $C(u), C(v) = \frac{1}{2} (u, v = 0);$

$C(u), C(v) = 1$ (其他)。

1.4 量化

对于前面得到的 8×8 个空间频率振幅值按比例缩小,并取其最接近的整数值的处理过程称为量化。JJPEG标准中推荐的量化表。一共有 64个元

素,按照从上到下,从左到右的顺序与 DCT变换后的幅值对应,每一元素记为 $Q(u, v)$, u 和 v 对应行和列,量化公式为式 (4),解码时反量化公式为式 (5)。

$$F^Q(u, v) = \text{IntegerRound} \frac{F(u, v)}{Q(u, v)} \quad (4)$$

$$F^Q(u, v) = F^Q(u, v) \times Q(u, v) \quad (5)$$

$F^Q(u, v)$ 代表量化后的幅值, $F^Q(u, v)$ 代表反量化后的幅值。IntegerRound就是进行四舍五入。

1.5 熵编码

在 JPEG图像压缩中,熵编码采用哈夫曼编码。哈夫曼编码基于不同符号的概率分布,对出现次数较多的符号赋予较短的代码,出现次数较少的符号赋予较长的代码。

2 分块显示的基本原理

Windows CE是一个图形用户界面相当出色的操作系统,采用窗口化的图形显示界面。对于一般的 Windows CE应用程序,除了系统提供的标题栏,菜单栏,工具栏之外还有一个客户区,它的大小可以通过 API函数 GetClientRect()来获得。客户区是用户与应用程序进行交互的区域。将图像信息显示给用户也是在这个区域进行。本文把客户区中固定用来显示图象的区域称为图像显示区。大图像所要显示图像的面积,肯定比图像显示区大得多,也就是说图像在图像显示区,只能显示它其中的一部分像素,其它部分是没有显示出来的。如果只显示其中的一部分像素而将整个文件调入计算机内存,虽然 Windows CE有虚拟内存机制,但是对于一幅 10240×10240 的图像,即使用完所有物理内存和虚拟内存肯定还是不够的。本文采用一种分块显示的方法对大像素图像进行显示,其原理如图 2所示。

首先对图像文件分块,将大像素文件分成若干个小像素块的图像文件,索引编号然后再进行存储。然后根据需要进行检索,检索所需要的目标图像块的文件,调入计算机内存,最终采用多线程将各个小像素图像文件分别显示出来。

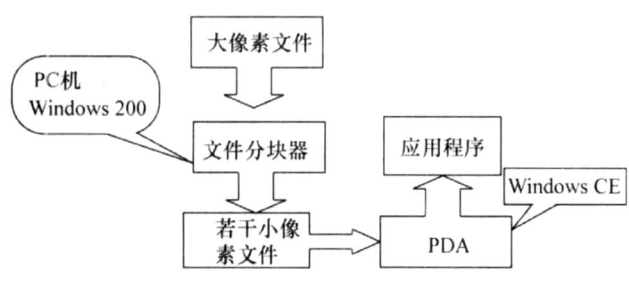


图 2 分块显示原理图

2.1 图像显示区域的网格分块

图像显示区是一块整的区域,如果整个显示区域只显示一个小图像文件,这样分割法既便于图像文件分块器的分割,也便于显示,采用一般的单线程对小像素文件进行 JPEG 解压,显示就可以了(如果为 BMP 图可以直接用 Windows CE API 显示)。但是由于分块后,每个小像素图像比原始图像,增加了多余边界区域。这些边界区域显示在屏幕的边缘,影响视觉效果。为消除图像显示区单一划分所带来的负面影响,本文对图像显示区划分进行了改进。把显示区域进行划分,平均划分成 4 个大小相等的区域, , , 如下图 2 所示。这样四个区域的像素块可以移动替换,处于单一划分情况下的边界区域,可以通过移动替换来消除负面影响。每个区域可以采用多线程单独分别显示每个区域所要显示的目标像素块。

图 3 应用程序图像显示区分块图

2.2 文件网格分块

把原始大像素图像 $(X \times Y)$, 划分若干个成与 , , 面积大小完全吻合的小像素图像并进行编号存储,假设整个图像显示区的宽度为 W (单位为像素),高度为 H 。分成四个小区域后,每个小区域宽度为 $W/2$ ($/$ 为求商运算),高度为 $H/2$ 。对大像素图像横向以 $W/2$ 为单位,纵向以 $H/2$ 为单位进行网格划分,若横向剩余不足整 $W/2$ 的部分对齐补足为 $W/2$,

纵向剩余不足整 $H/2$ 的部分对齐补足为 $H/2$ 。分成

$$M = X / \frac{W}{2} + x,$$

$$N = Y / \frac{H}{2} + y_0,$$

其中

$$x = \begin{cases} 0 (X \bmod \frac{W}{2} = 0), \\ 1 (\text{其他}). \end{cases} \quad y = \begin{cases} 0 (Y \bmod \frac{H}{2} = 0), \\ 1 (\text{其他}). \end{cases}$$

其分块情况如下图 4 所示。图中灰色部分为补足的区域。其中索引编号 $index$ 与网格坐标 (x, y) 的关系为 $index = x + y \times M$ ($x \geq 0, y \geq 0$), 如果索引号为 -1 表示该像素块为空。

$(0, 0)$ A_0	$(1, 0)$ A_1	...	$(M-1, 0)$ A_{M-1}
$(1, 0)$ A_M	$(1, 1)$ A_{M+1}	...	
		(x, y) $A_{x+y \times M}$	
			$(M-1, N-1)$ $A_{M \times N-1}$

图 4 分块网格图

将大像素文件,按照图 4 所示的分块,对每块图像信息进行存储成若干个小像素文件,文件名为其索引编号。任意四个彼此相邻(横向,纵向,斜向相邻)的像素块,组成一个像素信息组。

3 分块显示算法

在图像显示区域能显示给用为任意相邻的四幅小像素组成的一组图像信息,如何有效检索所需要的图像文件编号是算法的关键。本文采取一种中心区域显示法来检索,组织所需的图像编号。

3.1 中心区域显示法

中心区域:图像显示区域四分后,以图像显示

区域的中心点为中心,延着图像显示区域的对角线扩展的宽度等于 $W/2$,高度为 $H/2$ 的矩形区域,即为显示中心区域。同理,每一个像素信息组也具有其中心区域。

3.2 中心区域显示法

当需要显示的目标像素点,在某一图像组的中心区域时,就将该组图像信息显示在图像显示区域内。其主要就是利用网格检索算法检索出所需像素信息组左上,右上,左下,右下的网格坐标,具体描述如下。

3.3 网格检索算法

设 (x, y) 为所需要显示的目标像素点,首先该像素点所在像素块的网格坐标 $P(p_x, p_y)$ (6)式所示。然后根据其在图像块的位置来判断其所在的像素信息组中心点的网格坐标 $C(c_x, c_y)$, (7)式所示。最后根据像素信息组中心点的网格坐标来确定该组各个元素的网格坐标号 $P_0(x_0, y_0), P_1(x_1, y_1), P_2(x_2, y_2), P_3(x_3, y_3)$, (8)式、(9)式、(10)式, (11)式所示。

$$p_x = x / \frac{W}{2} \quad (6)$$

$$p_y = y / \frac{H}{2}$$

$$c_x = p_x + x / \frac{2p_x + 1}{2} \frac{W}{2} \quad (7)$$

$$c_y = p_y + y / \frac{2p_y + 1}{2} \frac{H}{2}$$

其中该像素组左上角的像素块的网格坐标 $P_0(x_0, y_0)$:

$$x_0 = c_x - 1 \quad (8)$$

$$y_0 = c_y - 1$$

右上角像素块的网格坐标 $P_1(x_1, y_1)$:

$$x_1 = c_x \quad (9)$$

$$y_1 = c_y - 1$$

左下角像素块的网格坐标 $P_2(x_2, y_2)$:

$$x_2 = c_x - 1 \quad (10)$$

$$y_2 = c_y$$

右下角像素块的网格坐标 $P_3(x_3, y_3)$:

$$x_3 = c_x \quad (11)$$

$$y_3 = c_y$$

最后利用

$$index = \begin{matrix} x + y \times M & (x < 0 & y < 0) \\ - 1 & (x < 0 & y < 0) \end{matrix}$$

将网格坐标转换文件索引号,如果索引号为 - 1表示该像素块为空。通过多线程分块显示在 , , 分区上。

4 结束语

在 Windows CE下显示大图像应用在很多工程领域,如在数字测绘的嵌入式导航地图中,经常利用一些像素非常大的航拍图作为电子底图以便进行野外作业^[5]。本文提出了分块索引显示法,在 PC机 Windows 2000下使用 Visual C++ 6.0编写文件分块器对 10 240 ×10 240大像素航拍图进行分块,在 Windows CE 5.0的操作系统,处理器速度为 520 MHz,内存为 128 Mb的 PDA,使用 EVC4.0编写测试程序进行了测试。实验结果及内存消耗情况如下表 1所示。

表 1 实验结果与内存消耗

	能否显示	设备内存	内存消耗情况	显示质量	边界区域
图像显示区单一划分	否	128 MB	几乎消耗所以内存
图像显示区单一划分	能	128 MB	218 kb	很好	不能显示
图像显示区四分	能	128 MB	308 kb	良好	能

参 考 文 献

- 1 Wu R Y, Tsai W H . A new one-pass parallel thinning algorithm for binary images. Pattern Recognition Letters, 1992; 13: 715—729
- 2 <http://www.jpeg.org/>
- 3 Kay D C. 图形图像文件格式大全北京:学苑出版社,
- 4 周长发精通 Visual C++图像处理北京:电子工业出版社,2007
- 5 张为舟. 嵌入式电子地图开发方法的研究. 华中师范大学硕士学位论文,2006
- 6 陈国军,叶飞跃. 大幅图像的分块细化加速算法. 计算机工程应用,2001; 23: 101—102

Model and Realization of Weapon SoS Combat Based on Multiagent

CAI Yan-xi, ZHANG Zhuo, SUN Yan

(Ordnance Engineering College, Shijiazhuang 050003, P. R. China)

[**Abstract**] Weapon SoS combat simulation basing on MAS is the frontier of military science. Currently, MAS models with practical values are few in the military. A MAS model of weapon SoS combat is introduced based on a weapon SoS combat simulation project which has been finished by now. The inside structure of single agent in MAS is analyzed, and the software realization is also shown.

[**Key words**] weapons SoS SoS combat MAS agent

(上接第 4173页)

Research of Blocking Display Large Pixels Pictures Algorithm Based on Windows CE

ZHU Ya-qi LU Wei-ting

(School of Electronics and Information Jiangsu University of Science and Technology, Zhenjiang 21200, P. R. China)

[**Abstract**] Windows CE is a typical embedded OS, because of the limitation of its hardware resource RAM of the embedded device is more deficient than traditional personal computer. So displaying a picture with large pixels is harder to embedded device (Windows CE OS) than PC (Windows 2000). This essay mainly introduces how to use Separate Blocked method to display a large pixels picture on Windows CE. First, it uses Grid Separate Blocked method to cut a large pixels picture (in BMP or JPEG format) into small pieces in PC. These small pieces are independent pixel blocks. Each saved in BMP or JPEG format. Each of small pictures has its index code and then copies all of these small pictures to Windows CE PDA. Second, in the PDA, it uses a Grid Searching method in order to search the index codes of the pixel blocks which are needed. And then, it displays each block with multithread method.

[**Key words**] Windows CE Embedded OS JPEG large pixels picture grid separate blocked method grid searching method multithread

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](#)
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](#)
73. [基于 CANopen 的列车通信网络实现研究](#)
74. [基于 SJA1000 的 CAN 总线智能控制系统设计](#)
75. [基于 CANopen 的运动控制单元的设计](#)
76. [基于 STM32F107VC 的 IEEE 1588 精密时钟同步分析与实现](#)

77. [分布式控制系统精确时钟同步技术](#)
78. [基于 IEEE 1588 的时钟同步技术在分布式系统中应用](#)
79. [基于 SJA1000 的 CAN 总线通讯模块的实现](#)
80. [嵌入式设备的精确时钟同步技术的研究与实现](#)
81. [基于 SJA1000 的 CAN 网桥设计](#)
82. [基于 CAN 总线分布式温室监控系统的设计与实现](#)
83. [基于 DSP 的 CANopen 通讯协议的实现](#)
84. [基于 PCI9656 控制芯片的高速网卡 DMA 设计](#)
85. [基于以太网及串口的数据采集模块设计](#)

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](#)
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](#)
54. [VxWorks 环境下内存文件系统的应用](#)
55. [VxWorks 下的多重定时器设计](#)
56. [Freescale 的 MPC8641D 的 VxWorks BSP](#)
57. [VxWorks 实验五\[时间片轮转调度\]](#)
58. [解决 VmWare 下下载大型工程.out 出现 WTX Error 0x100de 的问题](#)
- 59.

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](#)
52. [Linux 基于 W83697 和 W83977 的 UART 串口驱动开发文档](#)

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](#)
29. [Windows CENET 下 ADC 驱动开发设计](#)

30. [Windows CE 下 USB 设备流驱动开发与设计](#)
31. [Windows 驱动程序设计](#)
32. [基于 Windows CE 的 GPS 应用](#)

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](#)
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](#)
14. [数据结构考题 - 第 4 章 串](#)
15. [数据结构考题 - 第 4 章 串答案](#)
- 16.

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. [可编程 FIR 滤波器的 FPGA 实现](#)
7. [基于 FPGA 的 AES 加密算法的高速实现](#)
8. [基于 FPGA 的精确时钟同步方法](#)
9. [应用分布式算法在 FPGA 平台实现 FIR 低通滤波器](#)
10. [流水线技术在用 FPGA 实现高速 DSP 运算中的应用](#)
11. [基于 FPGA 的 CAN 总线通信节点设计](#)
12. [基于 FPGA 的高速时钟数据恢复电路的实现](#)

RT Embedded <http://www.kontronn.com>

13. [基于 FPGA 的高阶高速 FIR 滤波器设计与实现](#)
14. [基于 FPGA 高效实现 FIR 滤波器的研究](#)
- 15.