基于 NAND FLASH 的嵌入式系统启动速度的研究

杨　博，李　波

（西北工业大学 电子信息学院，陕西 西安　710072）

摘要：以 K9F1208U0M 为例，介绍了 NAND FLASH 的结构及原理并实现了一套基于 ARM9 处理器（S3C2410）的嵌入式 Linux 系统平台；加电后系统启动耗时过长（约为 60s），不能满足系统设计要求；通过分析嵌入式系统的结构和启动流程以及对 NAND FLASH 读写等管脚信号波形的捕捉和测试，发现嵌入式系统启动速度和 NAND FLASH 容量有着密切的联系；在此基础上进行了相关测试并加以论证，从而定位原因并提出解决方法，实验结果表明系统启动速度提升了约 2 倍，达到了系统设计要求。

关键词：嵌入式系统；S3C2410；启动；NAND FLASH；坏块

Research on the Booting Speed of Embedded System Based on NAND FLASH

Yang Bo, Li Bo

(School of Electronics and Information, Northwestern Polytechnical University, Shaanxi Xi'an　710072, China)

Abstract: Taking K9F1208U0M as an example, the structure and principles of NAND FLASH is presented, then a suit of Embedded Linux platform based on the ARM9 processor (S3C2410) is implemented. After powered on, the initialization time of embedded system is about 60s, which can not satisfy the system designing requirements because of its long time consuming. Through analyzing the structure and booting process of embedded system, capturing the related waveform about the key pin signals of NAND FLASH such as reading and writing, the close relationship between capacity of NAND FLASH and the booting speed of embedded system is found. After analysis of some tests, we locate the reasons for slowbooting and propose the corresponding solution. Experiment results show that the booting speed of system has been increased by up to 2 times, which can meet the system design targets.

Key words: embedded system; S3C2410; boot; NAND FLASH; bad block

0　引言

以 32 位的 ARM 处理器、开源的 Linux 操作系统为基础的嵌入式系统凭借其高性能、低功耗等优势被广泛应用于各种消费电子产品中，高度集成性和便携性已成为嵌入式系统的发展趋势。嵌入式系统中以闪存技术为代表的数据存储问题已开始受到人们越来越多的关注。

1　NAND FLASH 简介

1.1　闪存技术

闪存（Flash Memory）技术是 20 世纪 80 年代末逐渐发展起来的一种新型存储技术，按块擦除、按位编程，具备高速擦除、低功耗、高密度、可重复编程及高可靠性等优点。其中 NAND FLASH 和 NOR FLASH 是当前市场中两种主要的非易失闪存技术，NAND FLASH 具有存储容量大、成本低、擦写速度快等优点，主要用于大容量数据的存储，已在嵌入式系统的数据存储中被广泛使用。

1.2　NAND FLASH 存储结构和功能引脚

以 K9F1208U0M[1] 为例，将存储空间划分为 4 个页面（Plane），每个页面包含 1024 个块，共计 4096 个块（block），每个块由 32 个页（page）构成，页存储容量为 528Byte。

K9F1208U0M 芯片的功能引脚（指令集）如表 1 所示。

表 1　K9F1208U0M 功能引脚

引脚名称	有效电平	引脚描述
IO0～IO7(15)	高/低	数据输入/输出
CLE	高	命令锁存使能
ALE	高	地址锁存使能
nCE	低	片选使能
nRE	低	读使能
nWE	低	写使能
nWP	低	写保护
R/nB	高/低	准备好/忙

1.3　NAND FLASH 的读操作

NAND FLASH 的读写都以页为单位，擦除则以块为单位。通过 8 位 I/O 端口完成主设备对内存储器的访问，有效地节省了引脚的数量。另外，通过 CLE 信号和 ALE 信号线实现 I/O 口上的指令和地址的复用。NAND FLASH 页面读取操作流程如图 1 所示。

2　嵌入式系统平台

2.1　嵌入式系统平台的硬件环境

使用 Samsung 公司生产 S3C2410 芯片[2] 作为处理单元，选取容量为 64Mbyte，型号为 K9F1208U0M 的 NAND FLASH 芯片作为系统的存储介质。S3C2410 与周围接口、存储模块等一起构成嵌入式系统的硬件环境[3]。图 2 是嵌入式系统平台硬件的设计框图，图 3 是 S3C2410 与 NAND FLASH

收稿日期：2010－01－25；　**修回日期**：2010－03－02。

基金项目：国家自然科学基金项目（60872045，60572144，60702057）；教育部新世纪优秀人才支持计划（CET－06－0876）；西北工业大学基础研究基金（W018105）；西北工业大学科技创新基金（2008KJ01005）；西北工业大学青年科技创新基金（W016207）共同资助。

作者简介：杨　博（1986－），男，陕西省西安市人，硕士研究生，主要从事嵌入式系统，无线视频传输等方向的研究。

李　波，教授，博士研究生导师，主要从事无线 AD Hoc 网络，嵌入式系统等方向的研究。

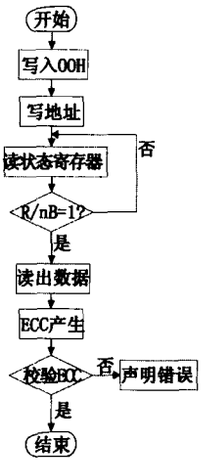

图 1　NAND FLASH 页读流程

的管脚信号连接图。

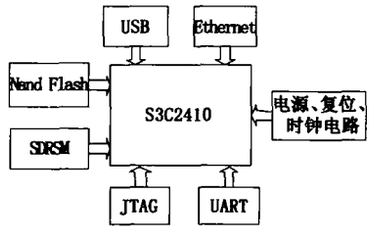

图 2　嵌入式系统的硬件结构

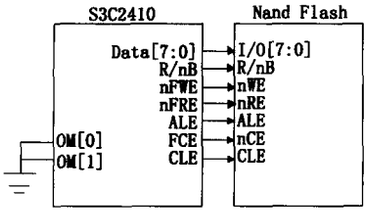

图 3　S3C2410 与 NAND FLASH 的连接

2.2　嵌入式系统平台的软件环境

本文选取 Linux 作为嵌入式操作系统[4]，版本为 Ubundu 8.04，vivi 版本为 0.1.4，内核版本为 2.6.28，文件系统为 Cramfs。

NAND FLASH 的烧写分区配置分别为 vivi：1～8 扇区，内核：13～128 扇区，文件系统：129～2048 扇区，其中 1 扇区 = 16KB。

3　基于 NAND FLASH 的系统启动问题

3.1　问题描述

为了支持 Linux 系统下的 NAND FLASH 启动，S3C2410 配置了一个叫作 Steppingstone 的内部 SRAM 缓冲器。当 S3C2410 设置成从 NAND FLASH 启动时，系统加电后，NAND FLASH 控制器会自动将其前 4K 数据搬移到 4K 内部 RAM 空间中，并把内部的 4K RAM 映射到 0x00000000 开始的地址空间。CPU 从内部 RAM 的 0x00000000 位置启动，然后将内核（存储空间分配约 2MB）从 NAND FLASH 拷贝到内存中运行。实验中启动过程花费约 60s，耗时过长，无法满足系统设计要求。

3.2　原因分析

基于对嵌入式系统启动流程分析的基础上，发现 NAND FLASH 与系统的启动存在密切的联系。我们在完成对 NAND FLASH 部分重要管脚信号的测试后发现：就本系统使用的 64MB 的 NAND FLASH 来说，存在理论公式：基于 NAND FLASH 的系统启动时间 T= NAND FLASH 的访问操作时间 T_1+ NAND FLASH 坏块检测时间 T_2。以下给出这两部分时间的深入分析：

3.2.1　NAND FLASH 的访问操作

NAND FLASH 的访问操作耗时 T_1=命令输入时间 +页面读取操作时间。由于命令输入时间所占比例很小，在性能估算中将其忽略，因此 NAND FLASH 的访问操作耗时主要取决于页面读取时间。

3.2.1.1　I/O 寻址

由于地址在 I/O [7：0] 上传递，因此必须采用移位的方式进行 I/O 寻址。对于 64MB 的 NAND FLASH，I/O 寻址分 4 个 cycle[1]]（周期）进行：在第一周期，传递列地址（column address），即 A [7：0]，A [8]（halfpage pointer）是由操作指令决定在上半页还是下半页进行读操作；在第二周期，通过移位将 A [16：9] 传到 I/O [7：0] 上；在第三周期，将 A [24：17] 放到 I/O [7：0] 上；最后，将 A [25] 放到 I/O 上。

3.2.1.2　页面读取

页面读取操作主要完成把数据从 NFDATA 寄存器中读出至缓冲区 buffer 中。以 64MB 的 NAND FLASH K9F1208U0M 为例，实现从 NFDATA 寄存器中读出数据的代码如下：

```
/*定义 NAND FLASH 的基地址*/
#define NF_BASE        0x4e000000
/*定义 NAND FLASH 相关寄存器地址*/
#define NFCONF      __REGi (NF_BASE + 0x0)
#define NFCMD       __REGb (NF_BASE + 0x4)
#define NFADDR      __REGb (NF_BASE + 0x8)
#define NFDATA      __REGb (NF_BASE + 0xc)
#define NFSTAT      __REGb (NF_BASE + 0x10)
#define BUSY 1
/*延时等待直到 NAND FLASH 为空闲状态*/
inline void wait_idle (void)
  {
     int i;
 while (! (NFSTAT & BUSY))
 for (i=0; i<10; i++);
  }
#define NAND_SECTOR_SIZE    512                # define NAND_BLOCK_MASK     (NAND_SECTOR_SIZE-1)
/*NAND FLASH 页读操作*/
int nand_read_ll (unsigned char *buf, unsigned long start_addr, int size)
{
    int i, j;
    if ( (start_addr & NAND_BLOCK_MASK) | | (size & NAND_BLOCK_MASK))
{
          return-1;
```

```
}
 /*芯片 Enable*/
 NFCONF &= ~0x800;
 for (i=0; i<10; i++);
 for (i=start_addr; i<(start_addr + size);)
 {
   /*读命令字 00H，执行读操作*/
    NFCMD = 0;
   /*四步寻址*/
    NFADDR = i & 0xff;
    NFADDR = (i >> 9) & 0xff;
    NFADDR = (i >> 17) & 0xff;
    NFADDR = (i >> 25) & 0xff;
   /*读状态寄存器*/
    wait_idle ();
   /*循环一次读 NFDATA 寄存器 512 次，读出 512 个字节*/
    for (j=0; j<NAND_SECTOR_SIZE; j++, i++)
    {
        *buf = (NFDATA & 0xff);
        buf++;
    }
 }
 /*芯片 Disable*/
  NFCONF |= 0x800;
   return 0;
}
```

3.2.2 NAND FLASH 的坏块检测

完成内核的拷贝后，系统会检测 NAND FLASH 芯片中每个块的可用性。由于 NAND FLASH 的生产工艺和技术等原因，不能保证 NAND FLASH 的内存排列在其生命周期中始终保持可靠的性能，因此在 NAND FLASH 的生产及使用过程中避免不了会产生坏块。

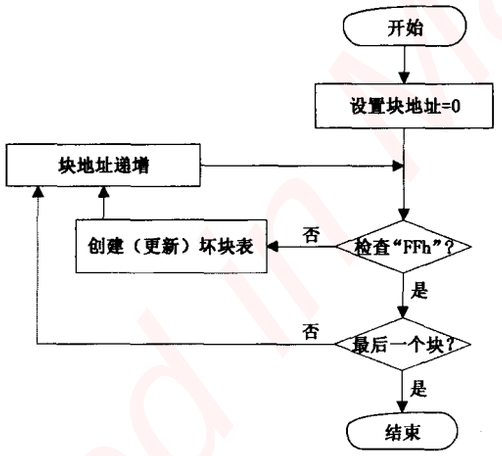

图 4 坏块检测流程图

坏块检测[1]时需检测每个块的 OOB（Out Of Band）区域的第 6 个字节（即列地址 517 字节）是不是 0xff。若不是，则表明这是坏块并标记为“不可用”，进而将其地址添加到创建好的坏块表（Bad Block Table）中。除了 OOB 区域的第 6 个字节外，通常至少把 OOB 前 3 个字节用来存放 NAND FLASH 的 ECC 校验码。坏块检测流程如图 4 所示。

3.3 解决方法

通过以上分析，要提高嵌入式系统的启动速度可以从减少 NAND FLASH 的访问操作时间和坏块检测时间两方面入手，使用容量较小的 NAND FLASH 可以同时从这两个方面减少额外的时间开销。

相比于 64M 的 NAND FLASH，容量减半为 32M 的 NAND FLASH 仅包含 2048 个块，块地址最高位只到 24 位，因此 I/O 寻址仅需 3 个周期即可完成，即每读取 512 个字节，将减少一个 NAND FLASH 写地址周期的时间开销：t=2ms+200μs=2.2ms[1]，其中 2ms 为擦除时间，200μs 为数据写入时间。对于本实验系统，需完成 Linux 系统内核约 2MB 的数据拷贝，这将节省 24tMb/521B 约 4000t 的时间开销。

其次，由于每读取 512 字节数据节省了一个 NAND FLASH 的写地址周期时间开销 t，使得状态寄存器 NFSTAT 处于“忙”的时间减少 t，因而再次避免了由于读取 2MB 数据而进入延时子函数 wait_idle() 所带来的额外时间开销，约 4000t。

基于以上两点，容量为 32M 的 NAND FLASH 访问操作耗时减少了约 8000t，即 8000×2.2ms=1.76s 的额外时间开销，因此有效地提升了数据从 NFDATA 读出至内存的速度。

再次，由于生产工艺水平有限，容量为 64M 的 NAND FLASH 存在的坏块数平均约为 70 个，容量为 32M 的 NAND FLASH 的坏块分布也随之减少，平均约 35 个。因而对于 32M 的 NAND FLASH 而言，花费在检测坏块、标记坏块并建立或更新坏块表的时间开销 T_2 将减少约为 64M 的一半。

通过以上的分析论证，当 NAND FLASH 容量由 64M 减半为 32M 后，系统启动时间减少为：$T' = T_1 - 18 + T_2/2$（秒）。因此可以采取使用较小容量的 NAND FLASH 的方法来解决此问题。

由于容量为 32M 的 NAND FLASH 已满足本系统中数据存储的需要，通过查阅 K9F5608U0D[5] NAND FLASH 的数据手册并与 64M 的 K9F1208U0M 进行对比，两者的管脚封装唯一区别是 pin6，K9F1208U0M 的 pin6 为 GND 信号，而 K9F5608U0D 的 pin6 为 N. C（无连接）。因此在无须改动原先设计好的电路原理图的前提下，可将 K9F1208U0M NAND FLASH 芯片直接更换为 K9F5608U0D NAND FLASH 芯片。

我们针对本嵌入式系统，分别使用 K9F1208U0M（64M）和 K9F5608U0D（32M）进行相应的模块耗时对比测试，实验结果如表 2 所示。测试结果表明当 NAND FLASH 的容量减半为 32M 时，系统启动耗时降至约 22s（已能满足系统设计要求），与 K9F1208U0D 相比，启动速度提升了 2 倍左右，性能得到了明显提升。

表 2 模块耗时对比测试结果

NAND FLASH 容量(M)	坏块检测耗时 T_2(s)	访问操作耗时 T_1(s)
64	23	37
32	12	20

4 结论

本文基于 S3C2410 的嵌入式 Linux 系统平台，介绍了 NAND FLASH 的工作原理并分析了嵌入式系统的启动过程，

变化趋势。从指标的数值结果可以看出，在相同的负载 ρ，平均数据帧数随空闲窗口长度的增大而增大；在相同的空闲窗口长度，系统负载越大平均数据帧数越大。这是因为空闲窗口越长，空闲期内到达的数据帧越多；负载越大，到达的数据帧越多，从而平均数据帧数越大。

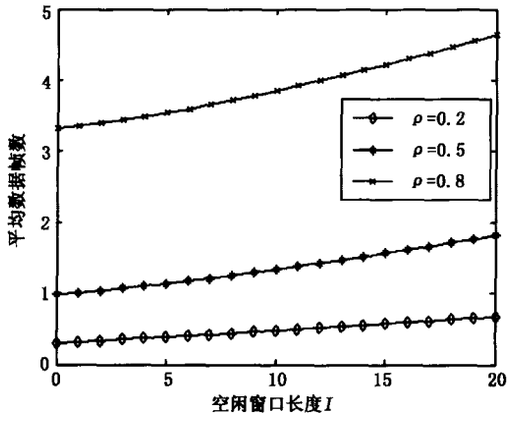

图 3　平均数据帧数 vs 空闲窗口长度

图 4 表示在不同负载下，系统切换率 β 随空闲窗口长度的变化趋势。在相同的负载 ρ，系统切换率随空闲窗口长度的增大而减小，这是因为空闲窗口越长，系统处于空闲模式的时间越长，空闲模式与清醒模式之间的切换次数越少，从而系统切换率越小。在相同的空闲窗口长度，$\rho=0.5$ 时系统切换率最大，$\rho=0.8$ 时系统切换率最小，这是因为负载较大时，系统处于清醒状态的时间越长，从而空闲模式与清醒模式之间的切换次数越少。

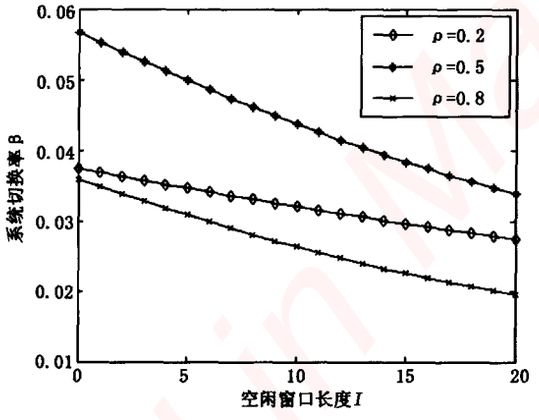

图 4　系统切换率 vs 空闲窗口长度

图 5 表示在不同负载下，能量节省率 θ 随空闲窗口长度的变化趋势。在相同的负载 ρ，能量节省率随空闲窗口长度的增大而增大，这是因为空闲窗口越长，系统处于空闲模式的时间越长，从而能量节省的越多，但这也导致数据帧的平均响应时间增大。在相同的空闲窗口长度，系统负载越大能量节省率越小，这是因为负载越大，系统处于空闲模式的几率越小，系统可能始终处于清醒状态或工作状态，从而使得能量节省越小。

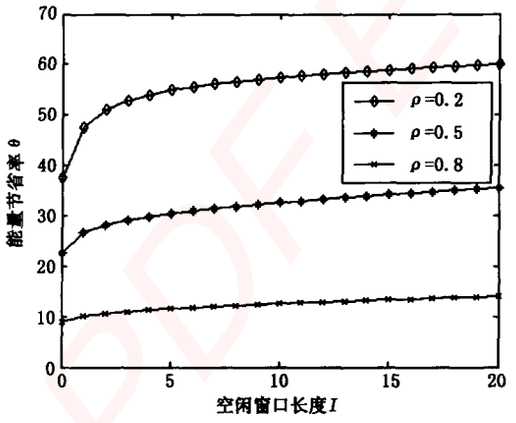

图 5　能量节省率 vs 空闲窗口长度

5　结论

本文首先分析了 IEEE 802.16e 中空闲模式机制，为更有效的研究性能指标与系统配置参数之间的关系，将缓冲区建模为带启动期和关闭期的多重休假 Geom/G/1 排队。应用嵌入 Markov 链方法，推导出性能指标的均值。根据排队指标得到空闲模式的性能指标。最后通过数值例子，进一步说明性能指标与空闲窗口长度之间关系。

参考文献：

[1] IEEE 802. 16e—2005, Part 16: Air Interface for Fixed and Mobile Broadband Wireless Access Systems—Amendment for Physical and Medium Access Control Layers for Combined Fixed and Mobile Operation in Licensed Bands [M]. 2006.

[2] Beomjoon Kim, Jaesung Park, Yong—Hoon Choi. Power Saving Mechanisms of IEEE 802. 16e: Sleep Mode vs. Idle Mode [M]. G. Min. ISPA 2006, LNCS 4331. Berlin Heidelberg: Springer—Verlag, 2006: 332-340.

[3] Xiao Y. Energy Saving Mchanism in the IEEE 802. 16e Wireless MAN [J]. IEEE Communication Letters, 2005, 9 (7): 595-597.

[4] 陈　利，纪澍琴，杨海波. IEEE 802. 16 中空闲模式节能方案分析 [J]. 计算机工程，2009，35 (20)：106-108.

[5] 田乃硕，徐秀丽，马占友. 离散时间排队论 [M]. 北京：科学出版社，2008.

[6] 唐少先，陈　曦. 增强型移动自主网络 MAC 层节能机制 [J]. 计算机测量与控制，2007，15 (11)：1645-1647.

(上接第 1871 页)

在此基础上对系统的启动速度慢的问题进行了论证、分析。通过多次实验和深入分析，我们发现 NAND FLASH 的启动速度与其存储容量有直接关系，因此系统设计时在满足存储容量要求的前提下，应尽可能选用小容量的 NAND FLASH 以尽可能缩短系统的启动延迟。

参考文献：

[1] Samsung. K9F1208U0M FLASH MEMORY Datasheet [Z]. 2003.

[2] Samsung. S3C2410A Datasheet [Z]. 2004.

[3] 孙秋野，孙　凯，冯　建. ARM 嵌入式系统开发典型模块 [M]. 北京：人民邮电出版社，2006.

[4] 俞　辉. 嵌入式 Linux 程序设计案例与实验教程 [M]. 北京：机械工业出版社，2009.

[5] Samsung. K9F5608U0D FLASH MEMORY Datasheet [Z]. 2005.

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
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
86. MVB1 类设备控制器的 FPGA 设计
87. MVB 接口彩色液晶显示诊断单元的显示应用软件设计
88. IPv6 新型套接字的网络编程剖析
89. 基于规则的 IPv4 源程序到 IPv6 源程序的移植方法
90. MVB 网络接口单元的 SOC 解决方案
91. 基于 IPSec 协议的 IPv6 安全研究
92. 具有 VME 总线的车载安全计算机 MVB 通信板卡
93. SD 卡的传输协议和读写程序
94. 基于 SCTP 的 TLS 应用
95. 基于 IPv6 的静态路由实验设计
96. 基于 MVB 的地铁列车司机显示系统研究
97. 基于参数优化批处理的 TLS 协议
98. SSD 数据结构与算法综述
99. 大容量 NAND Flash 文件系统中的地址映射算法研究
100. 基于 MVB 总线的动车组门控系统的设计与仿真研究

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. 基于 VxWorks 系统的 MiniGUI 图形界面开发
60. VxWorks BSP 开发中的 PCI 配置方法
61. VxWorks 在 S3C2410 上的 BSP 设计
62. VxWorks 操作系统中 PCI 总线驱动程序的设计与实现
63. VxWorks 概述
64. 基于 AT91RM9200 的 VxWorks END 网络驱动开发
65. 基于 EBD9200 的 VxWorks BSP 设计和实现
66. 基于 VxWorks 的 BSP 技术分析
67. ARM LPC2210 的 VxWorks BSP 源码
68. 基于 LPC2210 的 VxWorks BSP 移植
69. 基于 VxWorks 平台的 SCTP 协议软件设计实现
70. VxWorks 快速启动的实现方法[上电到应用程序 1 秒]

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
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
53. 基于 AT91RM9200 的嵌入式 Linux 系统的移植与实现
54. 路由信息协议在 Linux 平台上的实现
55. Linux 下 IPv6 高级路由器的实现
56. 基于 Android 平台的嵌入式视频监控系统设计

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
33. 基于 Windows CE 下大像素图像分块显示算法的研究
34. 基于 Windows CE 的数控软件开发与实现
35. NAND FLASH 在 WINCENET 系统中的应用设计

PowerPC：

WeChat ID: kontronn

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
31. 基于 MPC850 的多功能通信管理器
32. 基于 MPC8640D 处理系统的技术研究
33. 基于双核 MPC8641D 处理器的计算机模块设计
34. 基于 MPC8641D 处理器的对称多处理技术研究

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
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
36. UBoot 在 AT91RM9200 上的移植简析
37. 基于工控级 AT91RM9200 开发板的 UBoot 移植分析
38. 基于 ARM11 和 Zigbee 的人员定位防丢器

Hardware：

WeChat ID: kontronn

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
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

41. EFI-Tiano 环境下的 AES 算法应用模型
42. EFI 及其安全性研究
43. 基于 UEFI Shell 的 PreOS Application 的开发与研究
44. UEFI Bootkit 模型与分析

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. 用 IPv6 编程接口实现有连接通信的方法

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
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. FPGA 的 VHDL 设计策略
16. 用 FPGA 实现串口通信的设计
17. GPIB 接口的 FPGA 实现
18. 一种基于 FPGA 的 FFT 阵列处理器