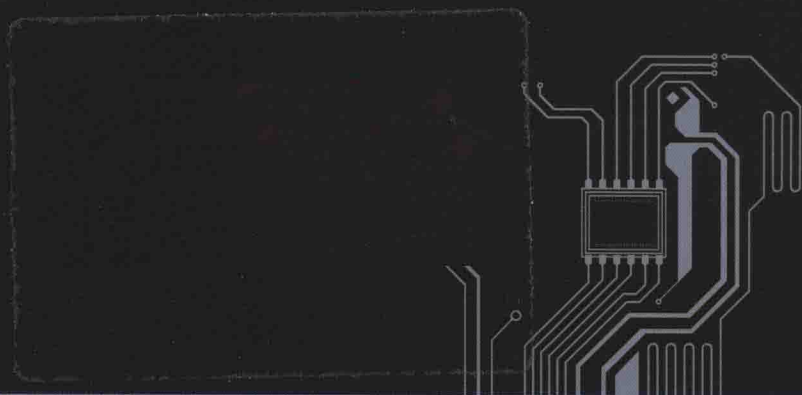




嵌入式Linux系统 开发入门宝典

—— 基于ARM Cortex-A8处理器

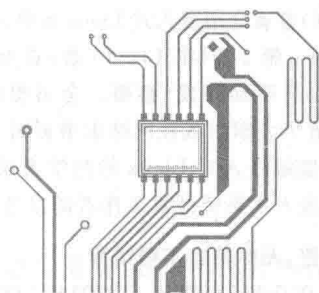
◎ 李建祥 编著



- 嵌入式系统开发入门必备图书，详细介绍Bootloader和Linux内核移植
- 循序渐进，重点突出；例程丰富，解释翔实；资源共享，超值服务
- 注重实践和调试，讲究细节和方法

清华大学出版社





嵌入式Linux系统 开发入门宝典

—— 基于ARM Cortex-A8处理器

◎ 李建祥 编著

清华大学出版社
北京

内 容 简 介

本书从嵌入式系统开发的基础知识开始讲起,全面介绍嵌入式开发过程中的方方面面。内容涵盖宿主 Linux 操作系统的安装设置以及常用工具的使用、配置,嵌入式编程基础知识(包括基于 Cortex-A8 架构开发环境的制作、配置和使用,ARM 处理器的常用汇编编程及其 ATPCS 规则,Makefile 规则,嵌入式 C 编程等),常用 IC 部件工作原理及其编程(俗称裸机编程),U-Boot、Linux 内核的分析、配置和移植,根文件系统的制作,基于 Linux 系统的驱动架构分析、驱动程序开发和移植。

全书共分 3 篇:第 1 篇(第 1~4 章)着重介绍嵌入式 Linux 系统开发前的一些准备知识;第 2 篇(第 5~12 章)着重讲解硬件部件的使用与编程;第 3 篇(第 13~16 章)着重讲解基于 Cortex-A8 处理器的嵌入式 Linux 系统开发中的系统分析、移植以及驱动开发、移植。全书提供了大量的应用实例,并且均在天嵌 TQ210 开发板上调试通过,读者可在清华大学出版社网站本书页面下载。

本书由浅入深、循序渐进,适合刚接触嵌入式 Linux 的初学者学习,同时可作为高等院校嵌入式相关专业本科、研究生教材,亦可作为广大嵌入式系统开发工作者的参考书。

本书封面贴有清华大学出版社防伪标签,无标签者不得销售。

版权所有,侵权必究。侵权举报电话:010-62782989 13701121933

图书在版编目(CIP)数据

嵌入式 Linux 系统开发入门宝典:基于 ARM Cortex-A8 处理器/李建祥编著.--北京:清华大学出版社,2016

(电子设计与嵌入式开发实践丛书)

ISBN 978-7-302-42471-0

I. ①嵌… II. ①李… III. ①Linux 操作系统—程序设计 IV. ①TP316.89

中国版本图书馆 CIP 数据核字(2015)第 310882 号

责任编辑:刘 星

封面设计:刘 键

责任校对:李建庄

责任印制:何 芊

出版发行:清华大学出版社

网 址: <http://www.tup.com.cn>, <http://www.wqbook.com>

地 址:北京清华大学学研大厦 A 座

邮 编:100084

社总机:010-62770175

邮 购:010-62786544

投稿与读者服务:010-62776969, c-service@tup.tsinghua.edu.cn

质量反馈:010-62772015, zhiliang@tup.tsinghua.edu.cn

课件下载: <http://www.tup.com.cn>, 010-62795954

印装者:三河市春园印刷有限公司

经 销:全国新华书店

开 本:185mm×260mm 印 张:23

字 数:568 千字

版 次:2016 年 4 月第 1 版

印 次:2016 年 4 月第 1 次印刷

印 数:1~2500

定 价:59.00 元

产品编号:062013-01

前言

一、为什么要写本书

随着芯片制造工艺的不断改进与提升,如今的芯片不仅体积越来越小,而且功能也越来越丰富,速度也成倍提升。比较典型的芯片如大家耳熟能详的 ARM 系列,从我们过去比较熟悉的 ARM7、ARM9 时代,发展到如今的 Cortex 系列,从单核时代升华到多核时代。硬件在飞速发展的同时,为之带来的是电子产品更新换代迅速、应用领域越来越广阔、知识更新也越来越快。

嵌入式 Linux 系统开发是嵌入式领域中非常热门的专业,需求大,但是对嵌入式 Linux 的入门很难,很多人不知道从何入手,迷失了方向,甚至半途而废。还有很多初学者,比如在校学生,对嵌入式的学习认识匮乏,常常为买什么开发板、买到开发板从何入手而发愁。追根究底,很多人是因为对嵌入式开发不了解,或者说缺乏相关的技术帮助资料。

鉴于上述种种原因,作者对如今嵌入式开发市场做了一些调查,发现与 ARM 相关的很多资料都还是基于 ARM7、ARM9 而写的,可 ARM11、Cortex 等资料较少,没有做到与时俱进,或者有一些但又太过专业,不适合初学者。对 ARM 家族的处理器的做了一些比较,从 ARM11 往前,属于 ARM 的一个时代,而从 Cortex 开始,可以说是 ARM 公司产品中的一个新亮点,未来基于它的产品会越来越多,而且 Cortex 还兼容前面的 ARM 架构。看来 Cortex 系列应该是未来的一个趋势,因此作者选择基于 Cortex 家族中的 A8,结合自己学习、工作的经历,循序渐进、由浅入深地讲解嵌入式 Linux 系统开发的方方面面,最终完成此书,期望能帮助读者加快嵌入式 Linux 系统开发的入门,并且对嵌入式 Linux 学习产生浓厚兴趣。

二、内容特色

与同类书籍相比,本书有如下特色。

(1) 循序渐进,由浅入深

本书以 TQ210(基于 Cortex-A8 架构、S5PV210 处理器)开发板为例,从开发环境的安

Foreword

装、配置,ARM 基本指令、Linux 常用命令的使用以及嵌入式 C 语言等相关基础知识开始,在读者掌握了基础知识后,结合硬件原理图,逐个分析硬件部件的工作原理以及编程方法,最后带领读者一步步进入嵌入式 Linux 系统开发的殿堂,学习诸如 U-Boot、Linux 内核的移植,根文件系统的制作,驱动开发与移植等各种技术。

(2) 例程丰富,解释翔实

古人云:“熟读唐诗三百首,不会做诗也会吟。”本书基于 S5PV210 开发板(TQ210),编写了丰富的实例源代码,并且每一个实例源代码都在 S5PV210 开发板上调试通过。每个代码后面都附有详细的分析注解,帮助读者理解掌握,进而加深对相关理论知识的理解。除此之外,一些编程思想、经验技巧亦可为读者提供借鉴。

(3) 资源共享,超值服务

书中用到的所有软件工具、程序源代码、文档学习资料,以及所有基于 S5PV210 开发板的裸机程序、U-Boot 代码、Linux 内核代码、根文件系统等学习资源,读者都可从清华大学出版社网站本书页面下载,并可以直接使用与测试。

另外,作者为此书开通了专用的网站 <http://www.qinfenwang.com>,读者可以直接与我们交流,共同学习和提高。另外,在国内比较知名的技术交流网站都有作者的博客(ID:js_gary),比如电子工程世界 EEWORLD、CSDN、电子技术设计 EDN China 等,读者可以通过博客与作者零距离接触。

(4) 传承经典,突出前沿

本书详细探讨了基于 Cortex-A8 架构的嵌入式 Linux 系统开发的始末,对 Cortex-A8 架构处理器的操作顺序、通用 GPIO 接口、内存管理器、中断机制等做了详细的讲解。书中配备了大量新颖的图片,以便提升读者的兴趣,加深对理论的理解。

三、内容结构

本书按照嵌入式 Linux 初学者的学习过程,从简单到复杂,从基本工具使用到系统的开发进行讲解,全书分 3 篇,共 16 章。

第 1 篇(第 1~4 章)为嵌入式 Linux 系统开发环境搭建篇,主要讲解以下内容:

- 第 1 章介绍嵌入式系统的概念、特点、发展历史,重点介绍了 Cortex-A8 的 ARM 架构以及 S5PV210 的处理器。
- 第 2 章讲解嵌入式 Linux 开发环境的搭建,包括在宿主机上安装、配置 Linux 操作系统(Ubuntu),交叉编译工具链制作等。
- 第 3 章介绍在嵌入式开发过程中经常使用的一些开发工具和 Linux 系统常用的命令,比如代码阅读和编辑工具、终端仿真工具等。
- 第 4 章介绍 GNU ARM 常用汇编指令、Makefile 的基本语法以及交叉编译工具的选项,ARM 基本指令集相关知识。本章可作为阅读本书的参考手册。

第 2 篇(第 5~12 章)为 Cortex-A8 嵌入式系统基本裸机编程篇。本篇基于 S5PV210 的数据手册介绍硬件部件的原理与使用方法,然后介绍怎样编写程序(即裸机程序)来操控它们。书中介绍了常用硬件部件的使用技巧,这是上层应用开发人员所不具备的技能。通过读/写各个硬件部件的寄存器来操控硬件,读者可以深刻体会到“软件”和“硬件”是怎么配

合工作的。另外,本篇也是第3篇的基础。

第3篇(第13~16章)为嵌入式Linux系统移植篇,具体内容如下:

- 第13章分析U-Boot代码的结构,最后详细讲解将U-Boot移植到S5PV210开发板上的方法。
- 第14章分析Linux内核代码的结构,以及内核启动过程,最后详细讲解移植内核到S5PV210的过程。
- 第15章介绍嵌入式Linux文件系统的目录结构,移植Linux常用命令工具集Busybox,建立各个目录和配置文件,最后编译制作文件系统映像文件。
- 第16章为驱动移植篇,先总体介绍嵌入式内核中驱动的编写、移植方法,然后重点介绍怎么在S5PV210平台上移植相关功能模块的驱动程序。

四、读者对象

- 对嵌入式Linux开发感兴趣的读者;
- 电子信息工程、计算机科学与技术相关专业的本科生、研究生;
- 相关工程技术人员。

五、致谢

本书主要由李建祥编写,同时还有王锋、卞曙旺、瞿苏、史瑞东等参与了本书的编写工作。另外,在本书编写过程中,得到了广州天嵌科技有限公司的大力支持和帮助,他们提供的高质量的开发板和技术资料,使得本书的写作有了很好的硬件平台,事半功倍,在此一并表示感谢。

感谢我的家人,在本书写作过程中给了我强大的精神支持和鼓励,使我能够坚持写完本书。

本书从写作到出版,曾得到刘其明教授的指导,并对书本中的关键章节提出了宝贵意见,在此表示感谢。另外,特别感谢清华大学出版社的工作人员,在本书的资料整理及校对过程中所付出的辛勤劳动。

限于作者的水平和经验,加之时间比较仓促,疏漏或者错误之处在所难免,敬请读者批评指正。有兴趣的朋友可发送邮件到js_gary@163.com,与作者交流。

作 者

2016年2月

目录

第一篇 工欲善其事,必先利其器

第 1 章 嵌入式系统概述	3
1.1 嵌入式系统基础知识	3
1.1.1 嵌入式系统简介	3
1.1.2 嵌入式系统的特点	3
1.1.3 嵌入式系统的发展历史	4
1.1.4 嵌入式系统的组成	5
1.1.5 嵌入式操作系统简介	6
1.1.6 嵌入式系统开发概述	6
1.2 基于 ARM 架构的 S5PV210 处理器	7
1.2.1 ARM 微处理器概述	7
1.2.2 ARM 流水线技术的发展	8
1.2.3 ARM Cortex-A8 处理器介绍	9
1.2.4 ARM Cortex-A8 寄存器组介绍	11
1.2.5 SAMSUNG S5PV210 处理器介绍	13
1.3 本章小结	15
第 2 章 嵌入式 Linux 开发环境搭建	16
2.1 交叉开发模式	16
2.1.1 嵌入式交叉开发模式介绍	16
2.1.2 硬件需求	17
2.2 软件环境搭建与配置	18
2.2.1 宿主机 Linux 操作系统的安装	18
2.2.2 配置宿主 Linux 操作系统	25
2.2.3 在宿主机上安装、配置开发环境	28
2.2.4 制作交叉编译工具链	29

Contents

2.3 本章小结	33
第3章 常用开发工具和命令的使用	34
3.1 Windows 环境下的工具介绍	34
3.1.1 代码阅读、编辑工具 Source Insight	34
3.1.2 文件传输工具 CuteFTP	39
3.1.3 终端仿真工具 SecureCRT	40
3.2 Linux 环境下的工具介绍	43
3.2.1 代码阅读、编辑工具	43
3.2.2 终端访问工具	43
3.3 嵌入式 Linux 系统常用命令介绍	43
3.3.1 编辑命令 vi(vim)	43
3.3.2 常用 13 个命令介绍	45
3.3.3 SD 卡烧写命令 df、dd	49
3.4 本章小结	50
第4章 嵌入式编程基础知识	51
4.1 GNU ARM 常用汇编指令介绍	51
4.1.1 相对跳转指令 b 和 bl	51
4.1.2 数据传送指令 mov 和地址读取伪指令 ldr	52
4.1.3 内存访问指令 ldr、str、ldm、stm	52
4.1.4 加减指令 add、sub	54
4.1.5 程序状态寄存器访问指令 msr、mrs	54
4.1.6 其他伪指令	54
4.2 ARM-Thumb 子程序调用(ATPCS)规则介绍	55
4.2.1 寄存器使用规则	55
4.2.2 数据栈使用规则	56
4.2.3 参数传递规则	56
4.3 ARM 交叉工具链介绍	56
4.3.1 编译工具 arm-linux-gcc	57
4.3.2 链接工具 arm-linux-ld	62
4.3.3 对象生成工具 arm-linux-objcopy	63
4.3.4 反汇编工具 arm-linux-objdump	64
4.4 Makefile 简介	65
4.4.1 基本规则	66
4.4.2 make 是如何工作的	66
4.4.3 变量的用法	66
4.4.4 常用函数介绍	68
4.5 本章小结	70

第二篇 千里之行,始于足下

第 5 章 基于 Cortex-A8 的 S5PV210 启动流程介绍	73
5.1 S5PV210 启动流程概述	73
5.1.1 外部启动介质介绍	73
5.1.2 iROM 启动的优势	73
5.2 S5PV210 上电初始化及内存空间分布	74
5.2.1 启动流程	74
5.2.2 空间分布	75
5.2.3 SD 卡引导块分配情况介绍	77
5.2.4 iROM 中的时钟配置	77
5.3 S5PV210 上的程序烧写介绍	78
5.3.1 程序烧写概述	78
5.3.2 制作 BL1 头信息	78
5.3.3 烧写 SD 卡	78
5.3.4 制作 Shell 脚本	79
5.4 本章小结	82
第 6 章 通用输入/输出接口 GPIO	83
6.1 GPIO 控制技术概述	83
6.1.1 GPIO 的介绍	83
6.1.2 S5PV210 的 GPIO 寄存器	83
6.1.3 实验用到的寄存器详解	85
6.2 S5PV210 的 GPIO 应用实例	86
6.2.1 实验介绍	86
6.2.2 程序设计与代码详解	87
6.3 本章小结	91
第 7 章 通用异步收发器 UART	92
7.1 UART 的原理及 S5PV210 的 UART 介绍	92
7.1.1 UART 通信的基本原理	92
7.1.2 S5PV210 的 UART 介绍	93
7.2 S5PV210 的 UART 应用实例	97
7.2.1 实验介绍	97
7.2.2 程序设计与代码详解	97
7.2.3 实例测试	100
7.3 本章小结	101

第 8 章 中断体系结构	102
8.1 S5PV210 中断体系结构	102
8.1.1 中断体系结构概述	102
8.1.2 中断控制寄存器介绍	105
8.2 S5PV210 的中断应用实例	107
8.2.1 实验介绍	107
8.2.2 程序设计与代码详解	107
8.2.3 实例测试	111
8.3 本章小结	111
第 9 章 系统时钟和定时器	112
9.1 S5PV210 的时钟体系结构	112
9.1.1 S5PV210 的时钟域和时钟源	112
9.1.2 S5PV210 的时钟应用和配置流程	113
9.1.3 S5PV210 时钟控制寄存器介绍	115
9.2 S5PV210 PWM 定时器	117
9.2.1 S5PV210 PWM 定时器概述	117
9.2.2 S5PV210 定时器	119
9.2.3 PWM 定时器的寄存器介绍	120
9.3 S5PV210 时钟和定时器应用实例	122
9.3.1 实验介绍	122
9.3.2 程序设计与代码详解	123
9.3.3 实验测试	132
9.4 本章小结	133
第 10 章 S5PV210 存储控制器	134
10.1 S5PV210 存储控制器介绍	134
10.1.1 存储控制器概述	134
10.1.2 DRAM 存储控制器	134
10.1.3 与外设的接线方式	135
10.1.4 DDR2 SDRAM 概述	136
10.1.5 S5PV210 DDR2 初始化顺序	140
10.1.6 存储控制器的寄存器介绍	141
10.2 存储控制器应用实例	147
10.2.1 实验介绍	147
10.2.2 程序设计与代码详解	147
10.2.3 实验测试	151
10.3 本章小结	151

第 11 章 S5PV210 Nand Flash 控制器	152
11.1 Nor Flash 与 Nand Flash 介绍	152
11.1.1 Flash 闪存	152
11.1.2 Nor Flash 与 Nand Flash 比较	152
11.1.3 Nand Flash 的物理结构	154
11.1.4 Nand Flash 的访问方法	155
11.2 S5PV210 Nand Flash 控制器介绍	158
11.2.1 Nand Flash 控制器的特性	158
11.2.2 Nand Flash 的模块图	159
11.2.3 Nand Flash 的引脚配置	159
11.2.4 Nand Flash 存储控制器配置	160
11.2.5 Nand Flash 寄存器介绍	160
11.2.6 Nand Flash 控制器的 ECC 校验方法	165
11.3 Nand Flash 控制器应用实例	166
11.3.1 实验介绍	166
11.3.2 程序设计与代码详解	166
11.3.3 实例测试	170
11.4 本章小结	170
第 12 章 LCD 控制器	171
12.1 LCD 介绍	171
12.1.1 LCD 的分类	171
12.1.2 LCD 的接口	172
12.2 S5PV210 LCD 控制器	173
12.2.1 S5PV210 LCD 控制器概述	173
12.2.2 S5PV210 LCD 控制器主要特性介绍	174
12.2.3 S5PV210 LCD 控制器功能介绍	175
12.2.4 S5PV210 TFT LCD 的操作	177
12.2.5 S5PV210 LCD 控制器编程方法介绍	181
12.2.6 S5PV210 LCD 控制器主要寄存器介绍	181
12.3 LCD 控制器应用实例	189
12.3.1 实验介绍	189
12.3.2 程序设计与代码详解	189
12.3.3 实例测试	194
12.4 本章小结	195

第三篇 欲穷千里目,更上一层楼

第 13 章 移植 U-Boot	199
13.1 Bootloader 介绍	199

13.1.1	Bootloader 概述	199
13.1.2	Bootloader 的结构和启动方式	201
13.1.3	Bootloader 操作模式和安装位置	203
13.1.4	如何编写 Bootloader	204
13.1.5	U-Boot 与内核之间的交互	205
13.2	移植 U-Boot 到 S5PV210 开发板	208
13.2.1	U-Boot 简介	208
13.2.2	U-Boot 源码结构	209
13.2.3	U-Boot 配置、编译与 SPL 介绍	212
13.2.4	U-Boot 启动过程源码分析	232
13.2.5	U-Boot 下的驱动移植	252
13.2.6	添加启动菜单	279
13.3	本章小结	281
第 14 章	Linux 内核移植和根文件系统制作	282
14.1	Linux 内核概述	282
14.1.1	Linux 内核发展及其版本特点	282
14.1.2	Linux 内核源码获取	283
14.1.3	内核源码结构及 Makefile 分析	285
14.1.4	Linux 内核的 Kconfig 介绍	291
14.1.5	Linux 内核配置选项	296
14.2	Linux 内核移植	299
14.2.1	Linux 内核启动过程分析	299
14.2.2	修改内核支持 S5PV210 平台	301
14.3	本章小结	304
第 15 章	构建 Linux 根文件系统	305
15.1	Linux 文件系统概述	305
15.1.1	文件系统概述	305
15.1.2	Linux 根文件系统目录结构	306
15.1.3	文件系统工作原理	310
15.2	移植 Busybox	310
15.2.1	Busybox 介绍	310
15.2.2	Busybox 的目录结构	311
15.2.3	内核 init 进程及用户程序启动过程	311
15.2.4	配置/编译/安装 Busybox	316
15.2.5	构建根文件系统	320
15.3	本章小结	325

第 16 章 驱动相关移植	326
16.1 Linux 驱动程序概述	326
16.1.1 驱动程序、内核和应用程序之间的关系	326
16.1.2 驱动程序分类	327
16.1.3 驱动程序开发步骤	327
16.1.4 驱动程序的加载和卸载	328
16.2 网卡驱动移植	328
16.2.1 DM9000 网卡特性	328
16.2.2 DM9000 驱动移植	329
16.3 Nand 驱动移植	332
16.3.1 S5PV210 平台 Nand 驱动移植	332
16.3.2 8 位硬件 ECC 和 Nand 启动	338
16.4 LCD 驱动移植	342
16.4.1 LCD 驱动概述	342
16.4.2 LCD 驱动移植	342
16.5 其他驱动移植	344
16.5.1 支持 SD 卡驱动	344
16.5.2 LED 子系统驱动移植	345
16.5.3 支持 RTC 驱动	346
16.5.4 支持 1-wire 单总线驱动	347
16.6 本章小结	348
参考文献	349

第一篇

工欲善其事,必先利其器

嵌入式系统已成为当前较为热门的技术,所以越来越多的人开始学习嵌入式系统开发,而在做开发前,必须要做一些准备工作,磨刀不误砍柴工。

嵌入式系统概述

本章学习目标

- 了解嵌入式系统概念、特点及发展历史；
- 了解 ARM 处理器的发展历程；
- 了解常用的嵌入式操作系统。

1.1 嵌入式系统基础知识

1.1.1 嵌入式系统简介

从 20 世纪 70 年代单片机的出现到各式各样的嵌入式微处理器、微控制器的大规模应用,嵌入式系统已经有了 40 多年的发展历史。如今嵌入式系统已经应用到科研、工业设计、军事以及人们日常生活的方方面面。表 1-1 列举了嵌入式系统应用的部分领域。

表 1-1 嵌入式系统应用领域举例

领 域	举 例
工业控制	能源系统、汽车电子、工控设备、智能仪表等
信息家电	通信设备、智能家居、智能玩具、家用电器等
交通管理	车辆导航、信息监测、安全监控等
航空航天	飞行设备、卫星等
国防军事	军用电子设备等
环境工程与自然	水源和空气质量监测、地震监测等

根据电子和电气工程师协会(IEEE)的定义,嵌入式系统为控制、监视或辅助设备、机器或用于工厂运作的设备。这是从嵌入式系统的应用领域来定义的。

另外国内普遍认同的嵌入式系统定义是:以应用为中心,以计算机技术为基础,软硬件可裁剪,适应应用系统对功能、可靠性、成本、体积、功耗等严格要求的专用计算机系统。

1.1.2 嵌入式系统的特点

从上面的定义,可以看出嵌入式系统有如下几个重要特征。

1. 软、硬件可裁剪,量身定制

日常用的个人计算机,同一个操作系统适用于所有的 CPU,反之亦然。而嵌入式系统为了实现低成本、高性能,通常软、硬件的种类繁多、功能各异、系统不具通用性,通常都是一套硬件配一套操作系统,比较有针对性。这些特征就决定了嵌入式系统在设计时需要精心设计、量身定做、去除冗余。

2. 体积小、低成本、低功耗、高可靠性、高稳定性

由于嵌入式系统应用领域广泛,这也就决定了它要适应不同的环境,比如高温、寒冷、长时间不间断运作等,所以在嵌入式系统软、硬件设计时就需要格外谨慎,使其具有低功耗、高可靠性、高稳定性等性能。

另外对嵌入式产品的外形体积、成本也有很高的要求,以使其可以镶嵌到主体设备之中,或者方便人们随身携带等。

3. 实时性、交互性强

很多嵌入式系统都有实时性的要求,在特定的空间或时间内,及时作出处理,比如温度监控系统等。这就要求其软件要固态存储,以提高速度,而且对软件代码的质量和可靠性也有较高要求。

除了实时性,嵌入式系统在很多时候需要人机交互,比如人们用键盘、鼠标操控嵌入式系统,这就要求嵌入式系统在设计时必须考虑其灵活方便性。

4. 对开发环境、开发人员的要求高

开发不同的嵌入式系统需要不同的开发环境,通常称之为交叉开发环境。比如做 ARM 嵌入式开发,就要有适合 ARM 架构的编译环境,做 MIPS 开发,就要有适合 MIPS 架构的编译环境。

另外,嵌入式系统不是一门独立的学科,它是将计算机技术、半导体技术、电子技术以及各行各业的应用结合于一体的产物。这就要求开发人员必须是复合型人才,对这些技术都要有所了解才行。

1.1.3 嵌入式系统的发展历史

在过去几十年的发展中,嵌入式系统主要经历了以下几个发展阶段。

1. SCM(Single Chip Microcomputer)阶段

SCM 中文名称是单片微型计算机,简称单片机。这一阶段系统的特点是芯片结构和功能都比较单一、存储容量较小、速率较低,几乎没有人机交互接口。嵌入式系统也只是一些可编程控制器形式的嵌入式系统,这类系统大部分应用于一些专业性强的工业控制系统中,通常都没有操作系统的支持,软件都通过汇编编写。这一阶段的单片机有 Intel 公司的 8048,也是最早的单片机,1976 年 Motorola 也推出了 68HC05 单片机,之后还有很多厂家也陆续研发出了自己公司的单片机产品。

随着大规模集成电路的出现和发展,以及通用计算机的性能不断提升,单片机式的嵌入式系统也随之发展了起来,这就是下一代嵌入式系统。

2. MCU(Micro Controller Unit)阶段

MCU 即微控制器阶段,这一阶段的嵌入式系统是以嵌入式 CPU 为主、以简单操作系统为核心,其中比较著名的有 Ready System 公司的 VRTX 等。其主要特点是:采用占先