

基于ARM 的 嵌入式系统开发 与实例

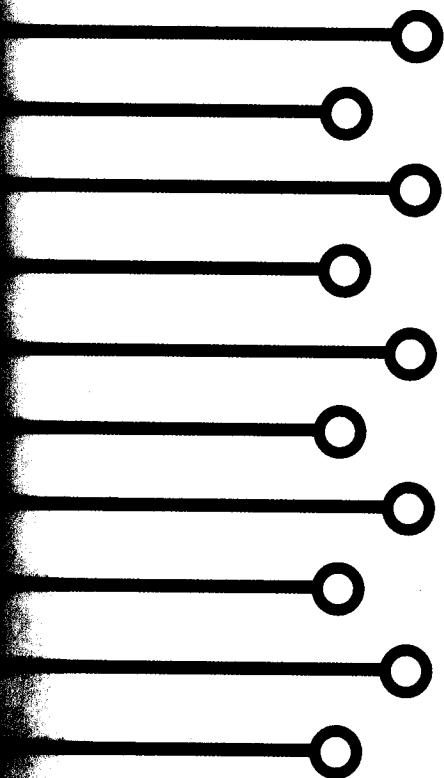
陈艳华 侯安华 刘盼盼 编著

TP332/125

2008

基于ARM的 嵌入式系统开发 与实例

陈艳华 侯安华 刘盼盼 编著



人民邮电出版社

北京

图书在版编目 (CIP) 数据

基于 ARM 的嵌入式系统开发与实例 / 陈艳华, 侯安华, 刘盼盼编著. —北京: 人民邮电出版社, 2008. 2
ISBN 978-7-115-17279-2

I. 基… II. ①陈…②侯…③刘… III. 微处理器, ARM—系统设计 IV. TP332

中国版本图书馆 CIP 数据核字 (2007) 第 186954 号

内 容 提 要

本书以嵌入式系统中应用非常广泛的 ARM 体系结构处理器为基础, 用带有三星公司 S3C44B0X 处理器的嵌入式开发板, 配合各种软件系统和工具, 来讲述嵌入式系统开发技术。第 1 章~第 3 章主要介绍嵌入式开发前常用平台和工具以及 ARM 体系结构和 SkyEye 模拟器原理等。第 4 章~第 22 章全部是工程案例, 具有极高的参考价值, 内容包括嵌入式开发中涉及的所有典型接口设计以及两种嵌入式操作系统 (μ C/OS 和 μ Clinux) 在嵌入式系统中的应用。

本书偏重于实例运用, 着重介绍了如何以 C 语言为工具进行嵌入式系统软件的开发, 所有源代码都是分段讲解, 层次分明且讲解透彻细致。

本书适合于高等院校自动化、通信、计算机及其相关专业高年级学生学习使用, 也适合从事 ARM 嵌入式开发人员和科技人员参考。

基于 ARM 的嵌入式系统开发与实例

- ◆ 编 著 陈艳华 侯安华 刘盼盼
责任编辑 刘 浩
- ◆ 人民邮电出版社出版发行 北京市崇文区夕照寺街 14 号
邮编 100061 电子函件 315@ptpress.com.cn
网址 <http://www.ptpress.com.cn>
北京鸿佳印刷厂印刷
新华书店总店北京发行所经销
- ◆ 开本: 787×1092 1/16
印张: 25.75
字数: 626 千字
印数: 1—5 000 册
- 2008 年 2 月第 1 版
2008 年 2 月北京第 1 次印刷

ISBN 978-7-115-17279-2/TP

定价: 45.00 元

读者服务热线: (010)67132692 印装质量热线: (010)67129223
反盗版热线: (010)67171154

前 言

嵌入式系统就是用于完成特有功能的计算机软硬件系统。嵌入式系统以应用为中心,以芯片技术、计算机技术和总线通信技术为基础。嵌入式系统是继 IT 网络技术之后,又一个新的技术发展方向。IEEE 对嵌入式系统的定义为:嵌入式系统是“用于控制、监视或者辅助操作机器和设备的装置”。嵌入式产品方便了人们的生活,嵌入式技术已经成为计算机应用领域一个新的增长点,嵌入式系统的开发越来越受到 IT 业界的重视,诸如实时控制、管理、移动计算、数据处理等各种自动化处理任务,移植嵌入式操作系统,开发设备驱动已经成为嵌入式开发领域的热点。

嵌入式微处理器是嵌入式系统的核心,与 PC 的处理器相比,嵌入式微处理器具有体积小、重量轻、成本低、可靠性高的优点。目前主要的嵌入式处理器类型有 ARM、MIPS、Power PC、6800、386EX、SC-400 系列等,其中 ARM 的应用范围最广泛。基于 ARM (Advanced RISC Machines) 内核的处理器以其诸多优异性能而成为各类产品中选用较多的处理器之一,以 32 位处理器作为高性能嵌入式系统开发的核心已经是嵌入式技术发展的必然趋势。

ARM 是 ARM 公司设计开发的一种嵌入式处理器的设计方案和体系结构。ARM 公司是全球领先的 16/32 位嵌入式微处理器知识产权设计供应商。ARM 内核最大的优势在于高速度、低功耗,ARM 内核创新之处还在于采用了 16 位 Thumb 指令集。16 位 Thumb 指令级是 ARM 指令集的一个子集,它是由 32 位指令通过代码压缩得到的,这种经过压缩的代码大大提高了系统运行效率。到目前为止,ARM 微处理器及技术的应用几乎深入到包括工业控制领域、无线通信领域、网络应用、消费类电子产品、成像和安全产品等各个领域,并会在将来取得更加广泛的应用。

S3C44B0X 微处理器是 Samsung 公司专为便携式设备提供的高性能和高性价比的微控制器解决方案,使用 32 位的低功耗 RISC 内核 ARM7TDMI,支持新型总线结构 SAMBA II (SAMSUNG ARM CPU embedded Microcontroller Bus Architecture)。同时,S3C44BOX 在 ARM7TDMI 处理器核的基础上,扩展了一系列通用外围器件,使系统成本及外围器件数口降至最低,并且集成了 LCD 控制器、UART 控制器、定时器、ADC 控制器、RTC 日历时钟控制器等,对于熟练掌握 ARM 系统开发,是一款很好的芯片。这也是本书选用它的原因。

本书共 22 章,基本涵盖了 S3C44B0X 中所有典型接口设计。从第 4 章开始,均以实例

形式编排，先讲述理论、结构和功能，然后讲述开发过程。各章节内容如下：

第 1 章介绍嵌入式开发前的准备工作，对于初入嵌入式开发的读者有一定帮助，这章目的是让读者熟悉嵌入式开发中的一些常用平台和工具。

第 2 章介绍 ARM 体系结构，包括了 ARM 处理器存储器、工作模式、工作状态，ARM 处理器内部寄存器和异常，ARM 指令集 S3C44B0X 处理器的引脚信号等，最后是对 BootLoader 的系列分析。本章对全面把握 ARM 体系结构具有很大帮助。

第 3 章主要介绍 SkyEye 模拟器的原理和如何利用 SkyEye 模拟器进行程序开发，包括 SkyEye 模拟器的原理与意义、SkyEye 模拟器的安装与使用。

第 4 章~第 14 章分别对 S3C44B0X 内部各模块的结构、功能和寄存器等进行了详细的介绍，并且在最后对这些模块进行了编程上的程序介绍。

第 15 章详细讲述 RTL8019AS 网卡的功能结构和寄存器，以及与 S3C44B0X 连接的方式。

第 16 章介绍 μ C/OS-II 实时操作系统的基本原理、工作机制， μ C/OS-II 在 ARM 处理器 S3C44B0 上的移植过程，实例过程是在移植好的 μ C/OS-II 源码的基础之上编写了一个多任务通信的典型例子，读者通过详读本章，能够初步掌握 μ C/OS-II 操作系统的应用。

第 17 章~第 19 章，这 3 章都是介绍基于 μ Clinux 的移植和应用。第 17 章介绍的是 μ Clinux 在 S3C44B0X 处理器上的移植过程，第 18 章和第 19 章介绍的是 μ Clinux 上驱动机制以及驱动的编写。

第 20 章介绍两种嵌入式中常用的图形界面开发工具：MiniGUI 和 QT，并以几个实例展现了如何去使用这两个工具的基本功能。

第 21 章以系统的观点介绍一个完整的 GPS 导航终端嵌入式系统开发过程，本章主要是从系统流程，对各部分进行了介绍。

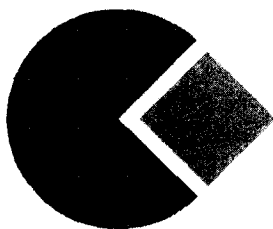
第 22 章介绍嵌入式上 TCP/IP 协议层设计过程，该实例采用 MICROCHIP 公司的一个小型以太网控制器 ENC28J60，在 S3C44B0X 上用软件模拟 SPI 时序，进行网络数据通信。实现了一个 Web 服务器。

本书代码可在 <http://www.ptpress.com.cn/download> 处下载。先注册用户，再通过书号或书名查询。

本书在编写的过程中得到王翠翠、仇亚飞、刘广兴、孙干、侯焕磊、王莹莹、张辰威、柳军旺、孙永全、张伟、寇团团、江孝林、陈运来、王跃、吴叶伟等很多人的技术支持和帮助，在此，衷心地谢谢他们。

本书编写时间匆促，编者水平有限，书中难免会出现错误，请读者不吝批评指正（电子函件：book_better@sina.com）。

编 者
2008 年 1 月



目 录

第 1 章 准备工作	1
1.1 建立开发平台	1
1.1.1 硬件平台	1
1.1.2 软件平台	3
1.1.3 Windows 开发平台	13
1.2 程序的调试	18
1.2.1 JTAG 调试技术介绍	18
1.2.2 调试例子	19
本章小结	22
第 2 章 ARM 处理器结构与指令集	23
2.1 ARM 处理器结构	23
2.1.1 ARM 处理器介绍	23
2.1.2 ARM 存储器	24
2.1.3 ARM 工作模式和状态	26
2.1.4 ARM 内部寄存器	27
2.1.5 ARM 异常	31
2.2 ARM 处理器指令系统	36
2.2.1 ARM 处理器寻址方式	36
2.2.2 ARM 指令集	37
2.2.3 Thumb 指令	59
2.3 S3C44B0X 处理器结构	60
2.3.1 S3C44B0X 结构介绍	60
2.3.2 S3C44B0X 片上资源	60
2.3.3 内部结构及引脚信号定义	62
2.4 S3C44B0X 汇编初始化程序	64

2.4.1	BootLoader 介绍	65
2.4.2	初始化代码	65
2.4.3	调试与运行	74
	本章小结	76
第 3 章	SkyEye 模拟器	77
3.1	SkyEye 模拟器介绍	77
3.1.1	SkyEye 简介	77
3.1.2	SkyEye 模拟器结构	78
3.1.3	SkyEye 的意义	78
3.2	SkyEye 模拟器的安装与使用	79
3.2.1	SkyEye 模拟器的安装	79
3.2.2	SkyEye 模拟器的使用	80
3.2.3	编写应用程序	82
3.2.4	文件系统	83
3.2.5	运行结果	83
	本章小结	84
第 4 章	Flash ROM 存储器接口设计	85
4.1	S3C44B0X 存储器控制器	85
4.1.1	概述	85
4.1.2	功能描述	86
4.1.3	特殊寄存器	90
4.2	Flash Rom 原理分析	96
4.2.1	Flash 器件介绍	96
4.2.2	Flash 读写操作	97
4.2.3	SST39VF160 芯片介绍	97
4.2.4	SST39VF160 芯片操作	98
4.3	接口电路与程序设计	102
4.3.1	电路连接	102
4.3.2	硬件和寄存器设置	103
4.3.3	程序的编写	103
4.3.4	调试与运行结果	106
	本章小结	107
第 5 章	RAM 存储器接口设计	108
5.1	SDRAM 芯片介绍	108
5.1.1	SDRAM 介绍	108
5.1.2	HY57V641620HG 的结构	109

5.2 接口电路与程序设计	109
5.2.1 电路连接	110
5.2.2 硬件和寄存器设置	110
5.2.3 程序的编写	111
5.2.4 调试与运行结果	112
本章小结	112
第6章 PWM 定时器与中断	113
6.1 S3C44B0X 中断机制分析	113
6.1.1 中断控制器	113
6.1.2 中断源与中断模式	116
6.1.3 中断优先级	117
6.1.4 其他特殊寄存器	119
6.2 PWM 定时器	121
6.2.1 定时器结构概述	122
6.2.2 定时器操作	123
6.2.3 死区产生器	125
6.2.4 DMA 请求模式	125
6.2.5 特殊寄存器	126
6.3 中断与定时器的应用	129
6.3.1 寄存器设置	129
6.3.2 程序的编写	130
6.3.3 调试与运行结果	131
本章小结	131
第7章 看门狗定时器	132
7.1 S3C44B0X 看门狗定时器	132
7.1.1 看门狗定时器概述	132
7.1.2 看门狗定时器操作	132
7.1.3 特殊寄存器	133
7.2 看门狗定时器应用	134
7.2.1 寄存器设置	134
7.2.2 程序的编写	135
7.2.3 调试与运行结果	136
本章小结	136
第8章 通用 I/O 键盘接口设计	137
8.1 S3C44B0X 通用 I/O 端口	137
8.1.1 I/O 端口概述	137

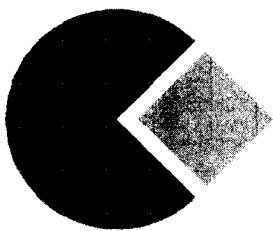
8.1.2 I/O 端口控制寄存器	140
8.2 键盘工作原理	143
8.2.1 键盘结构	143
8.2.2 按键消抖	143
8.2.3 键盘扫描	143
8.3 接口电路与程序设计	144
8.3.1 电路连接	144
8.3.2 寄存器设置	144
8.3.3 程序的编写	145
8.3.4 调试与运行结果	147
本章小结	147
第 9 章 串行通信接口设计	148
9.1 串行通信	148
9.1.1 异步串行通信原理	148
9.1.2 RS-232C 串口规范	149
9.2 S3C44B0X 异步串行通信	151
9.2.1 异步串行通信特性概述	151
9.2.2 UART 的操作	151
9.2.3 特殊寄存器	155
9.3 接口电路与程序设计	159
9.3.1 电路连接	160
9.3.2 寄存器设置	160
9.3.3 程序的编写	161
9.3.4 调试与运行结果	162
本章小结	162
第 10 章 I²C 总线接口设计	163
10.1 S3C44B0X 的 I ² C 总线模块	163
10.1.1 I ² C 总线介绍	163
10.1.2 I ² C 总线工作原理	164
10.1.3 I ² C 总线信号和时序	164
10.1.4 特殊寄存器	168
10.1.5 EEPROM 器件	170
10.2 接口电路与程序设计	172
10.2.1 电路连接	172
10.2.2 寄存器设置	172
10.2.3 程序的编写	173
10.2.4 调试与运行结果	175

本章小结	175
第 11 章 实时日历时钟 RTC	176
11.1 实时日历时钟 RTC	176
11.1.1 实时日历时钟 RTC 概述	176
11.1.2 实时日历时钟 RTC 功能	177
11.1.3 特殊寄存器	178
11.2 RTC 日历时钟的应用	181
11.2.1 寄存器设置	182
11.2.2 程序的编写	182
11.2.3 调试与运行结果	184
本章小结	184
第 12 章 A/D 转换器	185
12.1 A/D 转换器原理	185
12.1.1 A/D 转换器的类别	185
12.1.2 A/D 转换器的参数	186
12.2 S3C44B0X 的 A/D 转换器	187
12.2.1 S3C44B0X 的 A/D 转换器概述	187
12.2.2 特殊寄存器	189
12.3 A/D 转换器的应用	190
12.3.1 电路连接	190
12.3.2 硬件和寄存器设置	190
12.3.3 程序的编写	190
12.3.4 调试与运行结果	191
本章小结	192
第 13 章 DMA 控制器	193
13.1 DMA 控制器	193
13.1.1 DMA 控制器结构	193
13.1.2 DMA 控制器操作	195
13.1.3 特殊寄存器	198
13.2 DMA 数据传输的应用	203
13.2.1 寄存器	203
13.2.2 程序的编写	203
13.2.3 调试与运行结果	205
本章小结	206
第 14 章 LCD 接口设计	207
14.1 LCD 控制器	207

14.1.1	LCD 控制器介绍	207
14.1.2	LCD 控制器的结构	208
14.1.3	LCD 控制器操作	208
14.1.4	特殊寄存器	212
14.2	接口电路与程序设计	218
14.2.1	电路连接	218
14.2.2	寄存器设置	219
14.2.3	程序的编写	220
14.2.4	调试与运行结果	223
	本章小结	223
第 15 章	网卡接口设计	224
15.1	RTL8019AS 以太网控制器	224
15.1.1	RTL8019AS 介绍	224
15.1.2	NE2000 兼容的寄存器	225
15.1.3	RTL8019AS 操作分析	229
15.2	接口电路与程序设计	230
15.2.1	电路连接	230
15.2.2	寄存器	231
15.2.3	程序的编写	233
15.2.4	调试与运行结果	238
	本章小结	238
第 16 章	μC/OS-II 移植与应用	239
16.1	μ C/OS-II 实时操作系统	239
16.1.1	实时操作系统概念	239
16.1.2	μ C/OS-II 的文件结构	240
16.1.3	μ C/OS-II 的任务与中断	241
16.2	μ C/OS-II 的移植	242
16.2.1	移植条件和内容分析	242
16.2.2	μ C/OS-II 移植源文件	244
16.3	μ C/OS-II 应用程序设计	251
16.3.1	配置 OS_CFG.H 文件	252
16.3.2	任务函数的编写	252
16.3.3	调试与运行结果	256
	本章小结	256
第 17 章	μCLinux 移植	257
17.1	Linux 操作系统	257

17.1.1 Linux 介绍	257
17.1.2 Linux 内核	258
17.2 μ CLinux 操作系统	261
17.2.1 μ CLinux 介绍	261
17.2.2 μ CLinux 文件结构	262
17.3 μ CLinux 移植过程	263
17.3.1 寄存器配置和文件修改	263
17.3.2 编译过程	266
17.3.4 下载与运行结果	272
本章小结	272
第 18 章 μCLinux 驱动程序设计	273
18.1 Linux 驱动程序基本知识	273
18.1.1 Linux 驱动程序	273
18.1.2 字符设备驱动	278
18.1.3 添加驱动到内核	285
18.2 I ² C 驱动程序设计	285
18.2.1 I ² C 驱动程序的编写	285
18.2.2 测试程序的编写	291
18.2.3 模块的安装	292
18.2.4 调试与运行结果	293
本章小结	293
第 19 章 μCLinux 网卡驱动程序设计	294
19.1 Linux 网络协议层	294
19.1.1 网络层次总体结构	294
19.1.2 驱动程序分析	296
19.2 网卡驱动程序设计	307
19.2.1 RTL8019AS 驱动的编写	307
19.2.2 调试与运行结果	312
本章小结	312
第 20 章 嵌入式图形用户界面工具	313
20.1 显示驱动接口	313
20.1.1 framebuffer 驱动接口	313
20.1.2 qvfb 虚拟驱动接口	314
20.2 MiniGUI 图形用户界面工具	316
20.2.1 MiniGUI 介绍	316
20.2.2 MiniGUI 使用基础	320

20.2.3	MiniGUI 对话框、控件、菜单与绘图	323
20.3	Qt embeded 图形界面工具	333
20.3.1	Qt embeded 介绍	333
20.3.2	Qt embeded 使用基础	335
20.4	Qt Designer 图形界面开发	339
20.4.1	Qt Designer 介绍	339
20.4.2	Qt Designer 的使用	339
20.4.3	添加源代码	342
20.4.4	调试与运行结果	346
	本章小结	346
第 21 章	GPS 导航终端系统设计	347
21.1	GPS 全球定位系统	347
21.1.1	GPS 定位原理	347
21.1.2	NMEA-0183 标准	349
21.1.3	MapInfo 电子地图格式	350
21.2	GSM 网络系统	351
21.2.1	GSM 网络简介	351
21.2.2	SMS 短信息服务	351
21.3	GPS 导航系统设计过程	354
21.3.1	系统分析与规划	354
21.3.2	功能模块设计	357
21.3.3	电子地图设计	367
21.3.4	BootLoader 设计	371
21.3.5	μClinux 操作系统	371
	本章小结	375
第 22 章	微型嵌入式 Web 服务器的设计	376
22.1	网络通信技术	376
22.1.1	以太网简述	377
22.1.2	TCP/IP 协议	378
22.2	系统硬件结构设计	385
22.2.1	S3C44B0X 微控制器	385
22.2.2	网络控制器 ENC28J0 及其接口电路	386
22.3	软件程序结构设计	389
22.3.1	主程序流程	389
22.3.2	子程序的设计和实现	389
	本章小结	398
	参考资料	399



第 1 章 准备工作

在嵌入式开发的过程中开发人员应熟练使用相关的工具。例如在 Linux 环境下,需要开发人员掌握 Linux 的一些基本命令能熟练使用编译工具。本章作为嵌入式开发前的准备章节,是掌握嵌入式开发的起点,要掌握如下内容:

.....

硬件开发板资源 ■

Linux 和 Windows 开发平台中工具的使用 ■

JTAG 调试技术 ■

1.1 建立开发平台

开发平台包括硬件平台和软件平台;硬件平台主要是指开发板,也叫做目标机;软件平台主要是指开发的操作系统以及一些开发工具等。这些都是嵌入式开发的基础,下面将分别进行介绍。

1.1.1 硬件平台

本书所介绍的嵌入式开发都是基于三星 S3C44B0X ARM 处理器的硬件开发板的,因此有必要对此开发板进行介绍。开发板资源如图 1-1 所示。

开发板上的资源主要包括以下几个方面:

- S3C44B0X 处理器 (ARM7TDMI 内核), 其外部时钟频率为 10MHz, 内部时钟频率经 PLL 倍频最高可至 66MHz, 推荐工作频率为 64MHz;
- 8MB 的 16bit SDRAM (HY57V641620) 存储器;
- 2MB 的 16 bit Flash (SST39VF160) 存储器;
- 2 个 14 针 UART 接口 (经 MAX3232 转换);
- 符合 USB1.1 规范的 USB 设备控制器 PDIUSB12;
- RTL8019AS 全双工以太网控制器;

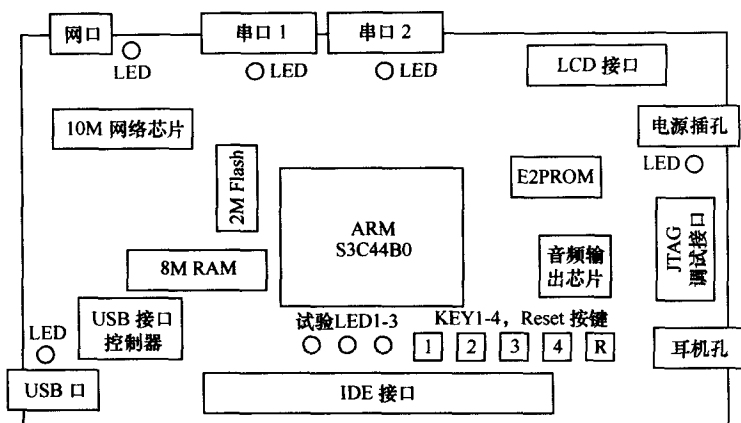


图 1-1 开发板示意图

- CS4334 音频输出芯片；
- 14 针 JTAG 调试接口；
- 3V 锂电池实时时钟保持系统；
- 5 个按键（其中 1 个是 RESET 键）；
- 1 片 24C02 E²PROM 存储芯片；
- 1 个蜂鸣器；
- 电源指示 LED，网络接口指示 LED，UART 接口指示 LED；
- 4 通道 PWM；
- 8 个外部中断口。

其中 CPU 内置资源如下：

- 2.5V 供电的 ARM7TDMI 内核，带 8KB 的高速缓冲器；
- 专用外部存储器（具备 FP/EDO/SDRAM/ROM 控制器和片选逻辑）；
- LCD 控制器（最大可支持 256 色 STN，专用 DMA 控制器）；
- 2 通道通用 DMA、2 通道桥 DMA，并具有外部请求引脚；
- 2 通道 UART（带有握手协议支持 IrDA1.0，具有 16-Byte FIFO）和 1 通道 SIO；
- 1 通道多主 IIC 总线控制器；
- 1 通道 IIS 总线控制器；
- 5 个 PWM 定时器和 1 通道内部定时器；
- 看门狗定时器；
- 71 个通用 I/O 口；
- 8 通道外部中断输入；
- 功耗控制具有普通、慢速、空闲和停止 4 种模式；
- 8 通道 10 位 ADC 输入；
- 带日历功能的实时时钟（RTC）；
- 带 PLL 的片内时钟发生器。

开发板内存配置为：NOR Flash（SST39VF160）接在 nGCS0，地址空间为 0x00000~0xFFFFF，宽度为 16 位，共计 0x200000 字节；SDRAM（HY57V641620）接在 nGCS6

上, 容量大小为 8MB, 地址范围是: 0xC000000~0xC7FFFFFF。内存分配如图 1-2 所示。

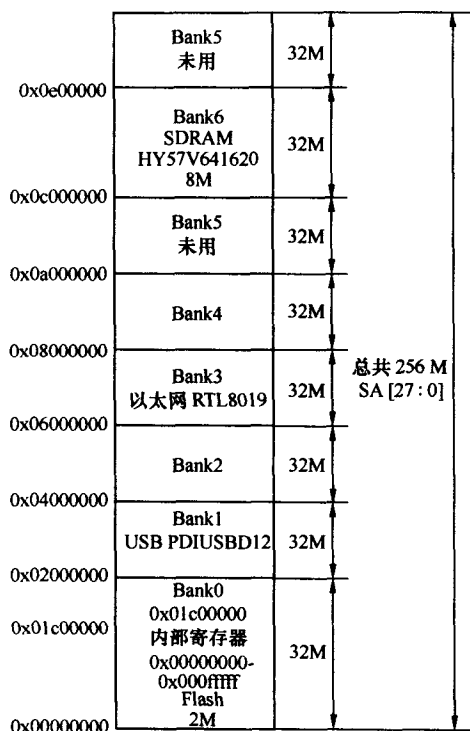


图 1-2 内存分配图

1.1.2 软件平台

Linux 软件开发平台主要指的是交叉编译环境、开发工具, 包括 minicom 串口仿真工具、TFTP 网络下载、MiniGui 图形开发工具和 Emacs 编辑器等。以下主要介绍 GNU 开发工具、交叉编译环境和 Emacc 编辑器等。

1. GNU 开发工具

(1) GNU 与 GCC 简介

GNU 是自由软件基金会的董事长 Richard M. Stallman 于 1984 年发起的, 至今已经有 20 多年的历史了。GNU 有自己的版权声明 GPL, 该版权声明规定, 用户在获得自由软件之后, 可以自己使用和修改, 并可以重新发布传播。但是在传播 GNU 软件时, 必须让下一个用户也有权利获得源代码, 并必须告之这一点。这样就保证了 GNU 永远是免费的、公开的。

GCC 的全名是“GNU 编译器合集”(GNU Compiler Collection), 属于 GNU 替代 UNIX 方案中的原始计划之一。Richard Stallman 成立 GNU 以及姊妹机构自由软件基金会 (Free Software Foundation), 其宗旨是开创一种可自由扩散的 UNIX 仿制版, 不受专有软件授权的拘束。第一版 GCC 在 1987 年发布, GCC 3.0 版于 2001 年问世。GCC 是通用型编译器, 支持的程序设计语言包括 C、C++、Java、Fortran、Pascal、Objective-C 以及 Ada。GCC。可支持的处理器平台包括英特尔 Pentium 的 x86, AMD 的 Opteron, Sun 的 Sparc, 惠普的 PA-RISC, IBM 的 Powe, 英特尔

的 Itanium、MIPS 和 ARM，日立的 SuperH 以及摩托罗拉的 68000 系列等处理器。其优点向来是可携性（portability）与跨平台支持（cross-platform support），而不是速度。

（2）GCC 工具的安装

可以在 GCC 官方网站 <http://gcc.gnu.org/> 下载 GCC 工具。可供下载的文件一般有两种形式：gcc-3.4.0.tar.gz 和 gcc-3.4.0.tar.bz2，这两种文件只是压缩格式不一样，内容完全一致，下载其中一种即可，安装步骤如下所述。

① 根据压缩格式，选择下面相应的一种方式解包：

```
#tar xzvf gcc-3.4.0.tar.gz
```

或者

```
#bzip2 gcc-3.4.0.tar.bz2 | tar xvf -
```

新生成的 gcc-3.4.0 这个目录被称为源目录，用 \$(src) 表示它。以后在出现 \${gcc} 的地方，应该用真实的路径来替换它。用 pwd 命令可以查看当前路径。在 \${src}/INSTALL 目录下有详细的 GCC 安装说明，可用浏览器打开 index.html 阅读。

② 建立目标目录，目标目录用 \$(obj) 表示，是用来存放编译结果的地方。GCC 建议编译后的文件不要放在源目录 \$(src) 中，最好单独存放在另外一个目录中。

③ 配置，目的是决定将 GCC 编译器安装到什么地方 \$(des)，支持什么语言并指定其他的一些选项等。其中，\$(des) 不能与 \$(obj) 或 \$(src) 目录相同。配置是通过执行 \$(src) 下的 configure 命令来完成的，其格式为：

```
$(src) ./configure --prefix=$(des) [其他选项]
```

例如，如果想将 GCC 3.4.0 安装到 /usr/local/gcc-3.4.0 目录下，用 \$(des) 表示这个路径，执行如下命令：

```
$(src) ./gcc-3.4.0/configure --prefix=/usr/local/gcc-3.4.0 --enable-threads=posix --disable-checking --enable-long-long --host=i386-redhat-linux --with-system-zlib --enable-languages=c,c++,java
```

表示将 GCC 安装在 /usr/local/gcc-3.4.0 目录下，支持 C/C++ 和 Java 语言，其他选项表示的意思参见 GCC 提供的帮助说明。

④ 编译，使用 make 命令，这个过程需要等待一段时间。

⑤ 安装，执行下面的命令将编译好的库文件等拷贝到 \$(des) 目录中，再键入下列命令即可完成安装。

```
$(src) make install
```

（3）GCC 工具的使用

GCC/g++ 在执行编译工作的时候，共需要 4 步：

- 预处理，生成 .i 文件；
- 预处理后的文件不转换成汇编语言，生成 .s 文件；
- 由汇编语言变为目标代码（机器代码）生成 .o 文件；
- 连接目标代码，生成可执行程序。

虽然 GCC 是 C 语言的编译器，但使用 GCC 将 C 语言源代码文件生成可执行文件的过程不仅仅是编译的过程，还要经历上述 4 个相互关联的步骤。GCC 首先调用 cpp 进行预处理，在预处理过程中，对源代码文件中的文件包含语句（include）和预编译语句（如宏定义 define 等）进