

嵌入式网络系统设计

【博客藏经阁丛书】

——基于Atmel ARM7系列

焦海波 刘健康 编著



北京航空航天大学出版社

嵌入式网络系统设计

【博客藏经阁丛书】

——基于Atmel ARM7系列

焦海波 刘健康 编著

TP360.21/28D

2008



北京航空航天大学出版社

内 容 简 介

正如书名所描述的那样,本书将引导读者从零开始,一步一步地设计完成一个具备实际应用价值的嵌入式网络系统。章节安排如下:第1章详细阐述硬件平台的设计;第2章指导读者搭建一个完整的开发、调试环境;第3章引领读者构建一个最基本的嵌入式系统;第4章讲解 $\mu\text{C}/\text{OS-II}$ 的移植和使用;第5章从理论层面讲解嵌入式TCP/IP协议栈LwIP的实现细节;第6~8章讲解如何把LwIP移植到本书在第1章完成的硬件平台上;最后两章讲解如何根据客户需求重构软件系统。

本书注重个人实际开发经验的总结,尽力避免纯理论性描述带给读者的枯燥乏味感。书中涉及的知识点凡是能够运用实例进行描述的,笔者均会提供一个设计实例,讲述将围绕实例进行;而且随书光盘还提供了大量、完整的设计实例和实现代码,具备较高的可操作性。

本书不会想当然地认为读者已经具备了某项知识,尽力做到对技术细节的描述没有遗漏,即使确实因篇幅和选题所限无法展开讲述的,也要告诉读者在哪里能够找到相关资料。

本书围绕一个实际的目标系统进行讲述,针对性很强,对工程技术人员具备较高的参考价值。

本书可作为嵌入式系统开发初学者的入门书籍,也可作为AT91SAM系列开发人员的工程参考书。

图书在版编目(CIP)数据

嵌入式网络系统设计:基于Atmel ARM7系列/焦海波,
刘健康编著. —北京:北京航空航天大学出版社,2008.4
ISBN 978-7-81124-262-1

I. 嵌… II. ①焦…②刘… III. 微型计算机—系统设计
IV. TP360.21

中国版本图书馆CIP数据核字(2008)第029525号

©2008,北京航空航天大学出版社,版权所有。

未经本书出版者书面许可,任何单位和个人不得以任何形式或手段复制或传播本书及其所附光盘中的内容。侵权必究。

嵌入式网络系统设计——基于Atmel ARM7系列

焦海波 刘健康 编著
责任编辑 杨 昕 苏向鹏

*

北京航空航天大学出版社出版发行

北京市海淀区学院路37号(100083) 发行部电话:(010)82317024 传真:(010)82328026

http://www.buaapress.com.cn E-mail:bhpress@263.net

涿州市新华印刷有限公司印装 各地书店经销

*

开本:787×960 1/16 印张:31 字数:694千字

2008年4月第1版 2008年4月第1次印刷 印数:5 000册

ISBN 978-7-81124-262-1 定价:49.00元(含光盘1张)

前言

记不清是受哪一部儿童科幻作品的影响,从很小的时候,就对能够控制机器的软件产生了浓厚的兴趣,一直梦想着自己也能够编写出这种神奇的东西。大学毕业后,虽然为了谋生,我一直从事数据库软件的开发,但这个梦想却一直未曾放弃。

大约是在 2003 年 11 月,笔者有幸参与了一款基于 ARM7 处理器的消费类电子产品的研发项目,从那时开始迈出了实现梦想的第一步。由于在这之前一直从事数据库软件的开发,没有任何嵌入式软件的开发经验,因此从书中获取相关的知识成为了最佳选择。对于还没有入门的嵌入式软件开发人员来说,太需要一个经典的、类似“Hello, World!”那样的程序引领我们进入嵌入式开发的殿堂了。但很不幸的是——我并没有找到这样的书。在所能找到的关于 ARM7 的书中,要么是罗列很多种不同厂家的芯片,然后逐一进行介绍;要么是针对某一个厂家的 ARM7 系列产品进行介绍;另外一些直接介绍 ARM7 的核,而其内容却来自官方发布的数据手册。必须承认,也有少量的讲解开发实例的书,但是这样的实例对于一个门外汉来说显得太高深莫测。回想笔者当时所处的困境,可以借用这样一段话来描述:“你是否有过这样的经历,有一天你兴致勃勃地买来一堆菜谱想学厨艺,翻开之后却发现自己根本没见过那些材料的名字,也不知道什么叫文火什么叫武火,什么叫上浆什么叫勾芡。而菜谱里根本没告诉你!你扔掉菜谱,垂头丧气,从此对厨艺失去兴趣。”(这段话来自于渊编写的《自己动手写操作系统》的自序,它曾经引起我强烈的共鸣。)

笔者当时的心情就是这样,类似异常、内核模式、中断、向量表、控制总线等诸如此类的名词、概念对我来说显得太抽象,太难以理解,我甚至怀疑自己是不是很不够聪明。为了证明自己,我疯狂地翻阅书本,强迫自己长时间地埋头于阅读之中,可结果还是不知道从哪里开始!最终,我放弃了这些一上来就把自己打懵的教科书,开始静下心来调整自己的学习思路。我决定从最简单的程序开始,仔细研读并尝试修改别人的例子代码,从实践中理解、掌握枯燥的理论知识。

我的第一个程序仍然是那个经典的“Hello, World!”,这个程序让我熟悉了嵌入式软件开发工具 ADS。记得当第一声问候成功地出现在 AXD(ADS 提供的调试工具)的控制台时,我很激动,我知道——起跑线已经被甩在了身后。虽然只迈出了一小步,但我已经找对了方向。接下来,继续根据别人提供的例子代码,用最简短的几句话实现了开发板上的流水灯程序,这个程序让我明白了最基本的底层驱动的开发方法。当看到板子上的 LED 指示灯亮起来时,我备受鼓舞,看来技术的基本原理是相通的,只要有信心,有扎实的编程功底,上层应用软件开发与底层软件开发没有什么不同。如果能够充分利用自身的知识储备,结合现有的芯片资料,编

前言

写底层驱动没有什么大不了的。就这样,我步步为营、积少成多,逐渐掌握了嵌入式软件开发的完整流程。到了这个时候,当我重新拿起那些教科书时,我发现能够看进去了,而且它们能够为我提供帮助了。

回顾我的学习历程,真正浪费大部分时间的正是很多 ARM 开发类的图书没有提及或只是粗略提及的内容。纵览各大 ARM 开发论坛,网友提出的大部分问题也大都集中于此,像 ARM 如何启动、如何处理中断、如何重映射、如何调试、如何下载、 $\mu\text{C}/\text{OS}-\text{II}$ 如何移植等,内容虽然个别书籍有所涉及,但是讲得并不详细,看了之后仍然给人一种云里雾里的感觉,不知如何着手。

大约是在 2006 年 4 月,我的业余时间逐渐多了起来,突然间有了写书的想法。我想通过一本书完整地再现自己的学习历程,并系统地总结已经掌握的知识,为其他想要从事 ARM 开发的朋友提供一些帮助。创作的过程是艰辛的,当我试图用浅显易懂的语言向读者阐明某一个知识点时,才发现理解与表达之间的距离是那么地遥远。为了避免泛泛的理论阐述给读者带来的疲劳感,每一个知识点我都提供了完整的例子代码和图示。这些例子代码都经过了我自己的验证,能够在目标板上稳定、可靠地运行,读者可放心使用。

本书围绕一个实际的目标系统进行。硬件平台以 Atmel 的 AT91SAM7x256(也简称 7x256)为基础进行搭建。AT91SAM7x256 是一个集成度很高的 ARM7 芯片,2006 年 4 月左右开始上市销售。这个芯片集成了很多时下流行的外设及外围接口,非常适合做工业级的产品研发。本书之所以选择这款芯片作为目标硬件平台,也正是充分考虑了它拥有的广阔应用前景。

本书的章节内容安排如下:

- 第 1 章指导读者设计完成硬件开发平台。这个平台包括两部分:一部分是基于 AT91SAM7x256 芯片的网络系统开发板,后面章节提供的所有例子程序都将在这个板子上执行;另一部分是仿真器,它提供程序的仿真、调试功能。对硬件设计不感兴趣的朋友可直接从第 2 章开始。
- 第 2 章指导读者建立开发环境,包括安装 ADS1.2 和安装设置仿真器两部分内容。
- 第 3 章指导读者构建一个最基本的嵌入式系统。本章将从那个经典的“Hello, World!”程序开始,由简到繁,完整地剖析了 ARM 程序的基本构造。包括以下几个方面内容:ADS 的基本使用方法;调试、仿真及下载程序;底层驱动的基本构造及编写方法;程序执行时的地址空间与 RO 选项的关系;系统的时钟选择及设置;ARM 的中断处理及内核模式;ARM 的启动过程分析;中断嵌套与快速中断;编写自己的 Flash 下载工具。
- 第 4 章将指导读者如何使用 RTOS 改进第 3 章刚刚建立的嵌入式系统。内容分为两部分:第 1 部分讲解如何将 $\mu\text{C}/\text{OS}-\text{II}$ 移植到 AT91SAM7x256 上;第 2 部分将通过几个实例讲解 $\mu\text{C}/\text{OS}-\text{II}$ 的内部通信及同步机制。
- 第 5 章从理论层面讲解嵌入式 TCP/IP 协议栈 LwIP 的实现细节。
- 第 6 章讲解如何利用 $\mu\text{C}/\text{OS}-\text{II}$ 提供的内部通信及同步机制实现 LwIP 的操作系统模拟层。

- 第7、8两章讲解如何使用LwIP,包括4方面内容:初始化LwIP;编写底层网络驱动;ping测试;简单WEB服务器测试。
- 第9章讲解如何根据客户需求重构软件系统,以改善系统的可维护性、可扩展性及可重用性。这一章仍然围绕一个具体的实例展开。笔者通过一个假定的客户需求向读者展示了系统重构的完整过程。包括以下内容:如何改善系统的可维护性;如何通过调整系统架构以改善系统的可扩展性及可重用性;编码实现新的系统架构,包括建立SWI系统调用机制。
- 第10章仍然属于软件重构的一部分,只不过由于它属于系统某一个模块的重构,所以笔者单独安排了一章进行专门阐述。这一章讲解的内容来自于第9章笔者假定的那个客户需求。讲解围绕着如何实现动态网页服务展开。包括以下内容:动态网页服务的设计思路;源码剖析;利用LwIP直接接口(即最底层的LwIP内部函数)实现DNS客户端。

附录A提供了开发板的原理图,有兴趣的读者朋友可以参考使用。

本书使用的硬件平台由刘健康设计、制作,第1章内容亦由他撰写,对硬件平台感兴趣的朋友可通过如下方式与他交流。

MSN及邮箱地址:lj004@163.com

除第1章之外的其他内容由笔者亲自撰写,对软件系统感兴趣的朋友可通过如下方式与我交流。

MSN及邮箱地址:marsstory99@hotmail.com

GOOGLE论坛:<http://groups.google.com/group/marsstory>

个人BLOG:<http://marsstory.blogcn.com>

本书提供的所有源码可通过随书光盘获得,如有不慎丢失光盘的朋友可通过MSN或邮件向笔者索取。

阅读本书只需读者具备一定的C语言及汇编语言基础即可。本书是假定读者不具备ARM汇编语言的基础,因此,凡是书中涉及的汇编代码笔者均进行了详细说明。

由于笔者水平有限,书中难免存在错误与不妥之处,欢迎读者朋友不吝赐教!

在本书的写作过程中,魏博、张海伦、屈明慧、吴成有、王先富、姜亦文、艾磊、李宗军、潘浩强、刘海朋、朱民、王超给了我和刘健康很大支持,在此表示感谢!

最后,感谢北航出版社嵌入式系统事业部的胡晓柏主任,是他的支持与信任才使本书与读者顺利见面。

焦海波

2008年4月

于济南

目 录

第 1 章 搭建自己的硬件平台

1.1 目标系统简介	1
1.2 芯片选型	1
1.3 主芯片功能特性介绍	2
1.3.1 Atmel AT91SAM7x256 芯片	2
1.3.2 Realtek RTL8201BL 物理层(PHY)芯片	6
1.4 电路设计	8
1.4.1 电源及时钟电路	8
1.4.2 接口电路	9
1.4.3 以太网电路	11
1.5 自己制作 Wiggler 仿真器	12
1.5.1 Wiggler 仿真器简介	12
1.5.2 Wiggler 仿真器电路	13
1.5.3 Wiggler 仿真器驱动软件	14

第 2 章 建立开发环境

2.1 安装 ADS1.2	17
2.2 安装设置仿真器	18

第 3 章 构建一个最基本的嵌入式系统

3.1 Hello, embedded world!	19
3.2 第一个驱动程序	25
3.2.1 点亮开发板	25
3.2.2 脱离仿真环境	32
3.2.3 代码剖析——驱动程序的基本构造	41
3.2.4 RO 与程序执行时的地址空间	48

目 录

3.3 使用高速时钟 PLLCK——让 LED 飞起来	53
3.3.1 Atmel 的头文件及时钟选择代码	53
3.3.2 时钟发生器与电源管理控制器	58
3.3.3 时钟设置代码剖析	62
3.4 精确控制 LED 的闪烁频率——使用定时器	66
3.4.1 编写第五个程序	67
3.4.2 映像文件、入口点与 Startup 段	71
3.4.3 中断处理	76
3.4.4 处理器模式	88
3.4.5 异 常	92
3.4.6 代码剖析	96
3.5 中断嵌套与快速中断	112
3.5.1 普通中断的嵌套	112
3.5.2 快速中断与嵌套	125
3.6 自己编写下载工具	128

第 4 章 系统的改进——使用 RTOS

4.1 RTOS 简介	132
4.2 μ C/OS-II——一个开源的 RTOS 简介	136
4.3 移植 μ C/OS-II 到 AT91SAM7x256	137
4.3.1 下载 μ C/OS-II 及移植范例	137
4.3.2 移植方法	142
4.3.3 准备工作——建立基本的 ADS 工程	144
4.3.4 OS_CPU.H	147
4.3.5 OS_CPU_C.C 的移植	159
4.3.6 OS_CPU_A.S	179
4.3.7 编写测试代码	197
4.3.8 编译器移植与 μ C/OS-II 配置文件	200
4.3.9 使用 AXD 验证 μ C/OS-II	207
4.4 μ C/OS-II 的通信及同步机制	223
4.4.1 实现第一个要求——使用信号量	223
4.4.2 实现第二个要求——使用互斥型信号量	228
4.4.3 实现第三个要求	231
4.5 总 结	247

第5章 进阶——嵌入式 TCP/IP 协议栈 LwIP

5.1 协议层	248
5.2 LwIP 概览	249
5.3 进程模型	249
5.4 操作系统模拟层	250
5.5 缓冲与内存管理	251
5.5.1 包缓冲区(Packet buffers)——pbufs	251
5.5.2 内存管理	252
5.6 网络接口	253
5.7 IP 处理	254
5.7.1 接收数据包	254
5.7.2 发送数据包	254
5.7.3 转发数据包	255
5.7.4 ICMP 处理	255
5.8 UDP 处理	256
5.9 TCP 处理	257
5.9.1 概 览	257
5.9.2 数据结构	258
5.9.3 序号计算	261
5.9.4 排队与发送数据	261
5.9.5 接收报文段	262
5.9.6 接受新的连接	263
5.9.7 快速重发	263
5.9.8 定时器	263
5.9.9 RTT 估算	264
5.9.10 阻塞控制	264
5.10 协议栈接口	264
5.11 应用程序接口	265
5.11.1 基本概念	265
5.11.2 API 实现	266
5.12 API 参考手册	267
5.12.1 数据类型	267
5.12.2 缓冲区函数	267

5.12.3 网络连接函数	270
5.13 实现 BSD Socket 库	276
5.13.1 socket 表示方法	276
5.13.2 分配 socket	276
5.13.3 连接设置	277
5.13.4 发送和接收数据	279
5.14 LwIP 应用实例	283
5.14.1 使用 LwIP API	283
5.14.2 使用协议栈直接接口	285

第 6 章 移植 LwIP——实现操作系统模拟层

6.1 获得 LwIP 源码	288
6.2 建立一个最基本的工程	288
6.3 把 LwIP 加入工程	288
6.4 官方文档——操作系统模拟层移植说明	290
6.5 准备工作——建立文件、定义数据类型及其他	293
6.6 信号量操作函数	297
6.6.1 sys_new_sem()函数	297
6.6.2 sys_sem_signal()函数	297
6.6.3 sys_sem_free()函数	298
6.6.4 sys_arch_sem_wait()函数	298
6.6.5 编译	301
6.7 邮箱操作函数	304
6.7.1 准备工作——设计思路及数据类型定义	304
6.7.2 sys_mbox_new()函数	309
6.7.3 sys_mbox_free()函数	311
6.7.4 sys_mbox_post()函数	311
6.7.5 sys_arch_mbox_fetch()函数	312
6.8 实现 sys_thread_new()函数	313
6.9 实现 sys_arch_timeouts()函数	316
6.10 实现临界保护函数	318
6.11 扫尾——结束操作系统模拟层的编写	319

第7章 以太网 MAC(EMAC)编程手册

7.1 概 览	323
7.2 结构框图	323
7.3 功能描述	324
7.3.1 接收缓冲区	325
7.3.2 发送缓冲区	328
7.3.3 EMAC 发送模块	330
7.3.4 暂停帧(Pause Frame)支持	331
7.3.5 EMAC 接收模块	331
7.3.6 EMAC 地址检查模块	332
7.3.7 广播地址	333
7.3.8 哈希寻址	333
7.3.9 复制所有帧(或混杂模式)	334
7.3.10 类型 ID 检查	334
7.3.11 VLAN 支持	334
7.3.12 PHY 管理	335
7.3.13 MII 与 RMII 接口	335
7.4 操作说明	336
7.4.1 初始化配置	336
7.4.2 初始化接收缓冲区数组	336
7.4.3 初始化发送缓冲区数组	337
7.4.4 地址匹配	337
7.4.5 中 断	338
7.4.6 发送帧	338
7.4.7 接收帧	338
7.5 用户接口——寄存器说明	339
7.5.1 网络控制寄存器	340
7.5.2 网络配置寄存器	342
7.5.3 网络状态寄存器	343
7.5.4 发送状态寄存器	344
7.5.5 接收缓冲区队列指针寄存器	344
7.5.6 发送缓冲区队列指针寄存器	345
7.5.7 接收状态寄存器	346

目 录

7.5.8	中断状态寄存器	346
7.5.9	中断使能寄存器	348
7.5.10	中断禁止寄存器	348
7.5.11	中断掩码寄存器	349
7.5.12	PHY 管理寄存器	350
7.5.13	暂停时间寄存器	351
7.5.14	低位哈希寄存器	351
7.5.15	高位哈希寄存器	351
7.5.16	低位特殊地址寄存器 1~4	352
7.5.17	高位特殊地址寄存器 1~4	352
7.5.18	类型 ID 寄存器	353
7.5.19	用户输入/输出寄存器	353
7.5.20	统计寄存器	354
 第 8 章 LwIP 应用之旅——初始化、网络驱动及使用		
8.1	rawapi.txt——LwIP 初始化说明	355
8.2	准备工作——建立 LwIP 入口文件	357
8.3	__ilvInitLwIP() 函数	358
8.4	__ilvSetLwIP() 函数	359
8.5	ethernetif_init()——初始化底层网络接口函数	364
8.5.1	ethernetif_init()——底层网络接口的入口函数	365
8.5.2	low_level_output()——链路层发送函数	366
8.5.3	low_level_init()——网卡初始化函数	368
8.5.4	EMACInit()——网卡初始化工作的实际完成者	370
8.5.5	ethernetif_input()——实现接收线程	397
8.5.6	low_level_input()——读取完整的一帧数据	399
8.5.7	GetInputPacketLen() 函数——获得帧长	401
8.5.8	EMACReadPacket()——搬运,从接收缓冲区到 pbuf	405
8.5.9	EMACSendPacket() 函数——发送一帧数据	411
8.5.10	编译 ethernetif.c 及 lib_emac.c 文件	415
8.6	ping——结束 LwIP 的移植	418
8.6.1	编译、链接整个工程	418
8.6.2	遗漏的工作——把 LwIP 整合到系统中	421
8.6.3	测试 LwIP——ping 及简单的 WEB 服务器	425

第9章 重构——继续改进系统

9.1 改善软件系统的可维护性	428
9.1.1 提高源码的可读性	428
9.1.2 添加必要的注释语句	430
9.1.3 重新调整程序文件及代码的组织结构	431
9.2 改善软件系统的可扩展性、可重用性	432
9.2.1 重新架构网络系统	432
9.2.2 编码实现新的系统架构	435
9.2.3 小 结	443
9.3 SWI 系统调用	444
9.3.1 工作机制	444
9.3.2 编码实现 SWI 系统调用	447

第10章 实现动态网页服务

10.1 设计思路	451
10.2 源码剖析	454
10.3 实现 DNS 解析	460
附录 A 开发板原理图	471
附录 B ARM 指令集编码	472
附录 C 程序状态寄存器位域说明	474
附录 D AT91SAM7x256 内嵌外设用户接口列表	475
附录 E PIO 控制器复用引脚说明	479
参考文献	481

第 1 章

搭建自己的硬件平台

1.1 目标系统简介

一个完整的嵌入式系统大体上可以分成软件和硬件 2 个部分。软件是灵魂,硬件是躯体。硬件是整个系统运行的基础,没有一整套完善可靠的硬件,再优秀的软件也将没有载体,发挥不出应有的功能。因此,第一步先要设计一个适合嵌入式网络系统的硬件电路。在本章中,首先分析系统的需求,根据需求确定处理器和相应的外设部件,最后一步一步完成整个电路的设计。

在当前的嵌入式处理器市场上,ARM7 在性能、功耗、价格等方面都有很大的优势,采用 ARM7 核的处理器型号众多,应用广泛,于是本书的设计也采用 ARM7 内核。

下面将要实现的这个系统具有网络通信能力,很显然它应该有一个以太网接口,为方便本书由浅入深的讲述,还包括其他一些基本的输入输出接口,例如按键、LED 指示灯等。

1.2 芯片选型

综合各方面的因素,包括性能、开发工具支持、成本等,最终选用了 Atmel 用于工业控制网络领域的 AT91SAM7x 系列 ARM 处理器 AT91SAM7x256 来作主处理器。它基于 ARM7TDMI 内核,内含高性能的 32 位 RISC 处理器,性能足以满足通常的网络应用,并且支持它的开发工具选择范围广泛,支持 JTAG 调试,开发和调试成本低廉。

在确定处理器之后,就可以确定相关的外部设备,如 Flash、RAM、串口等。在本系统中,AT91SAM7x256 芯片集成了 Flash 和 RAM,不需要再外扩。集成了一个以太网 MAC,只需扩展一片标准的 PHY 芯片就可以实现以太网接口,PHY 芯片选用了常见的 RTL8021BL。为充分利用 AT91SAM7x256 的集成外设,使用一片 MAX3232 实现 RS232 接口的电平转换,TJA1050T 实现 CAN 接口的电平转换,另外扩充一片 SPI 接口 Flash 芯片 AT45DB041 和 I²C 接口 E²PROM 芯片 AT24C04。

1.3 主芯片功能特性介绍

这一节主要是熟悉即将用到的2个主要芯片:AT91SAM7x256、RTL8201BL。对它们的内部结构和功能特点有个大致的了解,至于其内部某个设备的配置和用法,将在后面相关章节中用到时再作详细介绍。

1.3.1 Atmel AT91SAM7x256 芯片

AT91SAM7x256可以算是本书这个设计中外形体积最大的一个芯片了,有100个引脚,它不光外形大,内部包含的东西更是不少,在一块芯片上集成了ARM7TDMI处理器,片内256 KB Flash和64 KB SRAM,还包括USART、SPI、CAN控制器、Ethernet网口、定时器/计数器、实时定时器和模数转换器在内的一系列外围设备。AT91SAM7x系列中还包括AT91SAM7x128和AT91SAM7x512两款芯片,它们的区别在于内部集成Flash和SRAM数量不同,AT91SAM7x128拥有128KB的Flash和32KB的SRAM,AT91SAM7x512拥有512KB的Flash和128KB的SRAM。

片内Flash存储器可以通过JTAG-ICE接口编程,也可以通过并行接口对其进行编程。内置锁定位和安全位可以保护固件防止其被误覆盖并能保持其内容的机密性。

AT91SAM7x系列处理器可以为很多嵌入式控制应用提供灵活、成本优化的方案,特别是在一些要用到的以太网、CAN总线和Zigbee无线网络通信领域。

该芯片的功能特点如下:

- 内部集成ARM7TDMI ARM Thumb处理器。
 - 高性能32位RISC架构;
 - 高密度16位指令集;
 - 性能/功耗(MIPS/Watt)的领先者;
 - 支持嵌入式ICE内电路仿真以及调试通信接口。
- 片内高速Flash。
 - 512KB(AT91SAM7x512),分为2个Banks,每个Bank有1024页,每页256字节;
 - 256KB(AT91SAM7x256),共1024页,每页256字节;
 - 128KB(AT91SAM7x128),共512页,每页256字节;
 - 最差情况下仍保持30MHz的单时钟周期存取速度;
 - 预读取缓冲器使Thumb指令最优化运行以达到最高速度;
 - 每页编程时间为6ms,包括页自动擦除,全片擦除时间为15ms;
 - 10000次擦写次数,10年数据保持能力,扇区锁定功能,Flash安全锁定位;
 - 适合量产的快速Flash编程接口。

- 片内高速 SRAM, 可以在最高时钟速度下进行单时钟周期访问操作。
 - 128 KB(AT91SAM7x512);
 - 64 KB(AT91SAM7x256);
 - 32 KB(AT91SAM7x128)。
- 存储器控制器 (MC)。
 - 嵌入式 Flash 控制器, 中止状态和未对齐检测。
- 复位控制器(RSTC)。
 - 基于上电复位单元和经过工厂标定的低功耗掉电检测;
 - 提供外部复位信号整形和复位源状态。
- 时钟发生器(CKGR)。
 - 低功耗 RC 振荡器, 3~20 MHz 片上振荡器和一个 PLL。
- 电源管理控制器(PMC)。
 - 具有电源优化功能, 包括慢速时钟模式(低至 500 Hz)和空闲模式;
 - 4 个可编程外部时钟信号。
- 高级中断控制器 (AIC)。
 - 可分别单独屏蔽的具有 8 个优先级的向量中断源;
 - 2 个外部中断源和一个快速中断源, 可以防止伪中断。
- 调试单元(DBGU)。
 - 两线 UART, 支持调试通信接口中断, 可编程禁止 ICE 访问。
- 周期性间隔定时器(PIT)。
 - 20 位可编程计数器以及 12 位间隔计数器。
- 时间窗看门狗定时器(WDT)。
 - 12 位受预设值保护的可编程计数器;
 - 向系统提供复位或中断信号;
 - 当处理器处于调试模式或空闲状态时可以停止计数器。
- 实时定时器(RTT)。
 - 32 位自由运行的具有报警功能的 32 位计数器;
 - 以内部 RC 振荡器为时钟源。
- 两个并行输入/输出控制器(PIO)。
 - 62 个可编程的复用 I/O, 每个 I/O 最多可以支持两个外设功能;
 - 每个 I/O 口的电平变化都可以产生中断;
 - 可以独立编程为开漏、使能上拉电阻和同步输出。
- 13 个外围数据 DMA 控制器(PDC)通道。
- 一个 USB 2.0 全速(12 Mb/s)设备口。
 - 具有片上收发器, 1352 字节的可配置集成 FIFO。

第1章 搭建自己的硬件平台

- 一个 10/100 Base-T (Mb/s) 的以太网口。
 - 媒体独立接口 (MII) 或简化的媒体独立接口 (RMII);
 - 集成 28 字节的 FIFO 和专门用于收发数据的 DMA 通道。
- 一个兼容 CAN 2.0A 和 CAN 2.0B 的 CAN 控制器。
 - 8 个完全可编程消息对象邮箱, 16 位时间标志计数器。
- 一个同步串行控制器 (SSC)。
 - 每个接收器和发送器都有独立的时钟和帧同步信号;
 - 支持 I²S 接口, 支持分时复用;
 - 支持 32 位数据传输的高速连续数据流功能。
- 两个通用同步/异步收发器 (USART)。
 - 独立的波特率发生器, IrDA 红外调制/解调;
 - 支持 ISO7816 T0/T1 智能卡, 硬件握手信号以及 RS485;
 - USART1 口支持全功能的调制解调器信号。
- 两个主/从串行外设接口 (SPI)。
 - 8 位到 16 位可编程数据长度, 4 个外围设备片选线。
- 一个 3 通道 16 位定时器/计数器 (TC)。
 - 3 个外部时钟输入端, 每个通道有 2 个通用 I/O 引脚;
 - 倍速 PWM 发生器, 捕捉/波形模式, 递增/递减计数功能。
- 一个 4 通道的 16 位 PWM 控制器 (PWMC)。
- 一个两线接口 (TWI)。
 - 只支持主机模式, 支持所有的两线 Atmel E²PROM。
- 一个 8 通道 10 位模数转化器, 其中 4 个通道与数字 I/O 接口复用。
- SAM-BA Boot 助手。
 - 默认 Boot 编程方式;
 - 与 SAM-BA 图形用户界面的接口。
- IEEE 1149.1 JTAG 边界扫描支持所有数字引脚。
- 容许 5 V 输入/输出, 包括 4 个高达 16 mA 的大电流驱动 I/O。
- 电源提供。
 - 片上 1.8 V 稳压器, 可以为内核及外部组件提供高达 100 mA 的电流;
 - 3.3 V 的 VDDIO 提供 I/O 线电源, 独立的 3.3 V 的 VDDFLASH 提供 Flash 电源;
 - 具有掉电检测的 1.8 V 的 VDDCORE 提供内核电源。
- 全静态操作: 在 1.65 V 和 85℃ 的极限条件下仍能保持 55 MHz。
- 封装为 100 引脚的 LQFP。

AT91SAM7x512/256/128 的内部结构和包含的外设如图 1.3.1 所示。